i.MX 8M Plus Applications Processor Reference Manual

Document Number: IMX8MPRM
Rev. 3, 08/2024

Contents

Section number **Title** **Page**

Chapter 1
Introduction

1.1 Product Overview 9

1.2 Target Applications 9

1.3 Acronyms and Abbreviations 9

1.4 Architectural Overview 12

Chapter 2
Memory Map

2.1 Memory system overview 23

2.2 Cortex-A53 Memory Map 24

2.3 Cortex-M7 Memory Map 26

2.4 DMA memory maps 29

2.5 AIPS Memory Maps 30

2.6 DAP Memory Map 36

2.7 Audio Processor Memory Map 38

2.8 HDMI_TX Subsystem Memory Map 38

Chapter 3
Security

3.1 System Security 41

3.2 Resource Domain Controller (RDC) 44

Chapter 4
Arm Platform and Debug

4.1 Arm Cortex A53 Platform (A53) 91

4.2 Arm Cortex M7 Platform (CM7) 98

4.3 Messaging Unit (MU) 100

4.4 Semaphore (SEMA4) 144

4.5 On-Chip RAM Memory Controller (OCRAM) 162

4.6 Network Interconnect Bus System (NIC) 165

4.7 AHB to IP Bridge (AIPSTZ)..... 166
4.8 Shared Peripheral Bus Arbiter (SPBA)..... 189
4.9 TrustZone Address Space Controller (TZASC)..... 202
4.10 System Debug..... 204
4.11 System Counter (SYS_CTR)..... 208

Chapter 5
Clocks and Power Management

5.1 Clock Control Module (CCM)..... 229
5.2 General Power Controller (GPC)..... 570
5.3 Crystal Oscillator (XTALOSC)..... 724
5.4 Thermal Monitoring Unit (TMU)..... 728

Chapter 6
SNVS, Reset, Fuse, and Boot

6.1 System Boot..... 747
6.2 Fusemap..... 814
6.3 On-Chip OTP Controller (OCOTP_CTRL)..... 830
6.4 Secure Non-Volatile Storage (SNVS)..... 875
6.5 System Reset Controller (SRC)..... 904
6.6 Watchdog Timer (WDOG)..... 976

Chapter 7
Interrupts and DMA

7.1 Interrupts and DMA Events..... 995
7.2 Smart Direct Memory Access Controller (SDMA)..... 1016
7.3 Enhanced Direct Memory Access (eDMA)..... 1263
7.4 Interrupt Request Steering (IRQ_STEER)..... 1325

Chapter 8
Chip IO and Pinmux

8.1 External Signals and Pin Multiplexing..... 1335
8.2 IOMUX Controller (IOMUXC)..... 1358
8.3 General Purpose Input/Output (GPIO)..... 1985

Chapter 9
External Memory

9.1 External Memory Overview 2005
9.2 DDR Controller (DDRC) 2007
9.3 DDR BLK_CTRL 2181
9.4 DDR PHY (DDR_PHY) 2183
9.5 AHB-to-APBH Bridge with DMA (APBH-Bridge-DMA) 2194
9.6 62BIT Correcting ECC Accelerator (BCH) 2234
9.7 General Purpose Media Interface (GPMI) 2297

Chapter 10
Mass Storage

10.1 Enhanced Configurable SPI (ECSPI) 2355
10.2 FlexSPI Controller (FlexSPI) 2385
10.3 Ultra Secured Digital Host Controller (uSDHC) 2518

Chapter 11
Connectivity

11.1 HSIO BLK_CTRL 2677
11.2 Universal Serial Bus Controller (USB) 2699
11.3 Universal Serial Bus PHY (USB_PHY) 3014
11.4 PCI Express (PCIe) 3019
11.5 PCI Express PHY (PCIe_PHY) 3395
11.6 Ethernet MAC (ENET) 3787
11.7 Ethernet Quality Of Service (ENET_QOS) 3990
11.8 FlexCAN 5018

Chapter 12
Timers

12.1 General Purpose Timer (GPT) 5163
12.2 Pulse Width Modulation (PWM) 5183

Chapter 13
Display, Imaging, and Camera

13.1 Display, Imaging, and Camera Overview....5195
13.2 MEDIA BLK_CTRL....5203
13.3 LCD Interface (LCDIF)....5275
13.4 Image Sensing Interface (ISI)....5306
13.5 MIPI CSI Host Controller (MIPI_CSI)....5389
13.6 MIPI DSI Host Controller (MIPI_DSI)....5438
13.7 MIPI D-PHY (MIPI_DPHY)....5511
13.8 LVDS Display Bridge (LDB)....5535
13.9 HDMI TX Controller....5538
13.10 HDMI TX PHY....5821
13.11 HDMI TX BLK_CTRL....5868
13.12 HDMI TX Parallel Audio Interface (HTX_PAI)....5899
13.13 HDMI TX Parallel Video Interface (HTX_PVI)....5912
13.14 Image Signal Processor (ISP)....5936
13.15 DeWarp....5942
13.16 Machine Learning Neural Processing Unit (NPU)....5948

Chapter 14
Audio

14.1 Audio Overview....5985
14.2 AUDIO BLK_CTRL....5989
14.3 PDM Microphone Interface (MICFIL)....6020
14.4 Synchronous Audio Interface (SAI)....6066
14.5 Asynchronous Sample Rate Converter (ASRC)....6120
14.6 Enhanced Audio Return Channel (eARC)....6206
14.7 Audio DSP (HiFi 4 DSP)....6275

Chapter 15
Graphics Processing Unit (GPU)

15.1 GPU Overview....6283
15.2 2D Graphics Processing Unit (GPU2D)....6284

15.3 3D Graphics Processing Unit (GPU3D)..... 6298

Chapter 16
Video Processing Unit (VPU)

16.1 VPU G1 Decoder..... 6307

16.2 VPU G2 Decoder..... 6476

16.3 VPU VC8000E Encoder..... 6712

16.4 VPU BLK_CTRL..... 7331

Chapter 17
Low Speed Communication and Interconnects

17.1 I2C Controller (I2C)..... 7343

17.2 Universal Asynchronous Receiver/Transmitter (UART)..... 7367

Chapter 1
Introduction

1.1 Product Overview

This chapter introduces the architecture of the i.MX 8M Plus Applications Processor.

The i.MX 8M Plus family is a set of NXP products focused on machine learning applications, combining state-of-art multimedia features with high-performance processing optimized for low-power consumption.

The i.MX 8M Plus Applications Processor relies on a powerful fully coherent core complex based on a quad Cortex-A53 cluster, a Cortex-M7 coprocessor, audio digital signal processor, machine learning and graphics accelerators.

The i.MX 8M Plus provides additional computing resources and peripherals:

- Advanced security modules for secure boot, cipher acceleration and DRM support
- A wide range of audio interfaces
- Large set of peripherals that are commonly used in consumer/industrial markets including USB , PCIe, Ethernet, and CAN

1.2 Target Applications

The i.MX 8M Plus Media Applications Processor targets applications on:

- Smart Homes, Buildings and Cities
- Machine Learning and Industrial Automation
- Consumer and Pro Audio/Voice Systems

1.3 Acronyms and Abbreviations

The table below contains acronyms and abbreviations used in this document.

Acronyms and Abbreviated Terms

Term	Meaning
ADC	Analog-to-Digital Converter
AHB	Advanced High-performance Bus
AIPS	Arm IP Bus
ALU	Arithmetic Logic Unit
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
ASRC	Asynchronous Sample Rate Converter
AXI	Advanced eXtensible Interface
CA/CM	Arm Cortex-A/Cortex-M
CAAM	Cryptographic Acceleration and Assurance Module
CA53	Arm Cortex A53 Core
CAN	Controller Area Network
CM7	Arm Cortex M7 Core
CPU	Central Processing Unit
CSI	CMOS Sensor Interface
CSU	Central Security Unit
CTI	Cross Trigger Interface
D-cache	Data cache
DAP	Debug Access Port
DDR	Double data rate
DMA	Direct memory access
DPLL	Digital phase-locked loop
DRAM	Dynamic random access memory
ECC	Error correcting codes
ECSPI	Enhanced Configurable SPI
LPSPI	Low-power SPI
EDMA	Enhanced Direct Memory Access
EIM	External Interface Module
ENET	Ethernet
EPIT	Enhanced Periodic Interrupt Timer
EPROM	Erasable Programmable Read-Only Memory
ETF	Embedded Trace FIFO
ETM	Embedded Trace Macrocell
FIFO	First-In-First-Out
GIC	General Interrupt Controller
GPC	General Power Controller
GPIO	General-Purpose I/O
GPR	General-Purpose Register
GPS	Global Positioning System

Table continues on the next page...

Term	Meaning
GPT	General-Purpose Timer
GPU	Graphics Processing Unit
GPV	Global Programmers View
HAB	High-Assurance Boot
I-cache	Instruction cache
I2C or I^2C	Inter-Integrated Circuit
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IOMUX	Input-Output Multiplexer
IP	Intellectual Property
IrDA	Infrared Data Association
ISP	Image Signal Processor
JTAG	Joint Test Action Group (a serial bus protocol usually used for test purposes)
ELCDIF	Liquid Crystal Display Interface
LDO	Low-Dropout
LIFO	Last-In-First-Out
LRU	Least-Recently Used
LSB	Least-Significant Byte
LUT	Look-Up Table
LVDS	Low Voltage Differential Signaling
MAC	Medium Access Control
MCM	Miscellaneous Control Module
ML	Machine Learning
MMC	Multimedia Card
MSB	Most-Significant Byte
MT/s	Mega Transfers per second
NPU	Neural Processing Unit
OCRAM	On-Chip Random-Access Memory
OCOTP	On-Chip One-Time Programmable Controller
PCI	Peripheral Component Interconnect
PCIe	PCI express
PGC	Power Gating Controller
PIC	Programmable Interrupt Controller
PMU	Power Management Unit
POR	Power-On Reset
PSRAM	Pseudo-Static Random Access Memory
PWM	Pulse Width Modulation
PXP	Pixel Pipeline
QoS	Quality of Service
R2D	Radians to Degrees
RISC	Reduced Instruction Set Computing

Table continues on the next page...

Term	Meaning
ROM	Read-Only Memory
RTOS	Real-Time Operating System
Rx	Receive
SAI	Synchronous Audio Interface
SD	Secure Digital
SDIO	Secure Digital Input/Output
SDLC	Synchronous Data Link Control
SDMA	Smart DMA
SIM	Subscriber Identification Module
SNVS	Secure Non-Volatile Storage
SoC	System-on-Chip
SPBA	Shared Peripheral Bus Arbiter
SPDIF	Sony Philips Digital Interface
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SRC	System Reset Controller
TFT	Thin-Film Transistor
TPIU	Trace Port Interface Unit
TSGEN	Time Stamp Generator
Tx	Transmit
TZASC	TrustZone Address Space Controller
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
USDHC	Ultra Secured Digital Host Controller
WDOG	Watchdog
WLAN	Wireless Local Area Network
WXGA	Wide Extended Graphics Array

1.4 Architectural Overview

This section contains the i.MX 8M Plus architectural overview.

1.4.1 Block Diagram

The high-level block diagram is shown in the figure below.

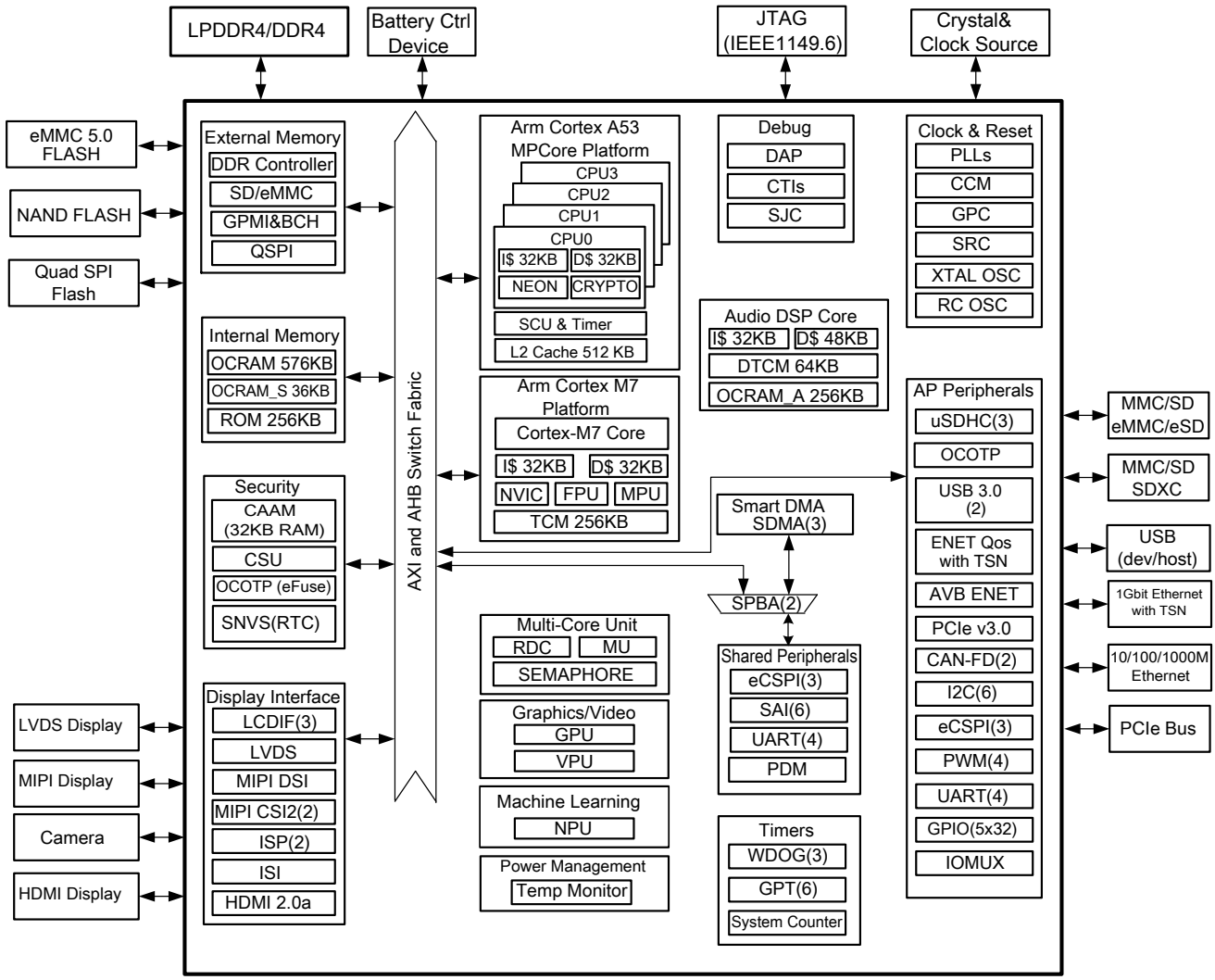


Figure 1-1. Block Diagram

1.4.2 Features

1.4.2.1 Arm Cortex-A53 MPCore™ Platform

The i.MX 8M Plus Applications Processor is based on the Arm Cortex-A53 MPCore™ Platform, which has the following features:

- Quad symmetric Cortex-A53 processors, including:
  - 32 KB L1 Instruction Cache
  - 32 KB L1 Data Cache

    - Media Processing Engine (MPE) with NEON technology supporting the Advanced Single Instruction Multiple Data architecture
    - Floating Point Unit (FPU) with support of the VFPv4-D16 architecture
- Support of 64-bit Armv8-A architecture
- 512 KB unified L2 cache

1.4.2.2 Arm Cortex-M7 Platform

The Cortex-M7 Core Platform includes the following:

- 32 KB L1 Instruction Cache
- 32 KB L1 Data Cache
- 256 KB TCM

Available customer applications include:

- Low power standby mode
- IoT device control
- ML applications

1.4.2.3 System Bus and Interconnect

System bus and interconnect include the following:

- Network interconnect (NoC) AXI arbiter
- Quality of service controller (QoSC) to configure priorities and limits of AXI transcations
- Performance monitor (PERFMON) to monitor AXI bus activity
- Debug monitor (DBGMON) to record AXI transactions preceding a system reset

1.4.2.4 Clocking and Resets

Clocking and resets include:

- Clock control module (CCM) provides centralized clock generation and control
    - Simplified clock tree structure
    - Unified clock programming model for each clock root
    - Multicore awareness for resource domains
- System reset controller (SRC) provides reset generation and distribution

1.4.2.5 Interrupts and DMA

Interrupts and DMA include:

- 160 shared peripheral interrupts routed to Cortex-A53 Global Interrupt Controller (GIC) and Cortex-M7 nested vector interrupt controller (NVIC) for flexible interrupt handling
- Three Smart direct memory access (SDMA) engines. Although these three engines are identical to each other, they are integrated into the processor to serve different peripherals.
  - SDMA-1 is a general-purpose DMA engine which can be used by low speed peripherals including UART, SPI and also others peripherals.
  - SDMA-2 and SMDA-3 is used for audio interface, including SAI-1/2/3/5/6/7, SPDIF and PDM audio input.

1.4.2.6 On-Chip Memory

The on-chip memory system consists of the following:

- Boot ROM (256KB)
- On-Chip RAM - OCRAM (576KB)
- Audio Processor System RAM - OCRAM_A (256KB)
- On-Chip RAM for State Retention - OCRAM_S (36KB)

1.4.2.7 External Memory Interface

The external memory interfaces supported on this chip include:

- 32-bit DRAM Interface:
  - LPDDR4-4000
  - DDR4-3200
- 8-bit NAND FLASH, including support for Raw MLC/SLC devices, BCH ECC up to 62-bit, and ONFi3.2 compliance (clock rates up to 100 MHz and data rates up to 200 MB/sec)
- eMMC 5.1 FLASH (2 interfaces)
- SPI NOR FLASH (3 interfaces)
- FlexSPI FLASH with support for XIP (for example for use in a low-power mode when DRAM is not accessible) and support for either one Octal SPI, or parallel read mode of two identical Quad SPI FLASH devices

1.4.2.8 Timers

The timers on this chip include:

- One local generic timer integrated into each Cortex-A53 CPU
- Global system counter with timer bus interface to Cortex-A53 MPCore generic timers
- One local system timer (SysTick) integrated into the Cortex-M7 CPU
- Six general purpose timer (GPT) modules
- Three watchdog timer (WDOG) modules
- Four pulse width modulation (PWM) modules

1.4.2.9 Graphics Processing Unit (GPU)

The chip incorporates the following Graphics Processing Unit (GPU) features:

- One GPU for 2D and composition acceleration
  - Supports multi-source composition
  - Supports one-pass filter
  - Supports tile format
- One GPU for 3D processing
  - Two Shader Execution Units
  - Supports OpenGL ES 1.1, 2.0, 3.0, 3.1
  - Supports OpenCL 3.0
  - Supports OpenVG 1.1
  - Supports OpenGL 4.0
  - Supports EGL 1.5
  - Supports Vulkan 1.1
  - Supports tile format

1.4.2.10 Video Processing Unit (VPU)

The chip incorporates the following Video Processing Unit (VPU) features:

- Video Decode:
  - 1080p60 HEVC/H.265 Main, Main 10 (up to level 5.1) (VPU G2)
  - 1080p60 VP9 Profile 0, 2 (VPU G2)
  - 1080p60 VP8 (VPU G1)
  - 1080p60 AVC/H.264 Baseline, Main, High decoder (VPU G1)

- Video Encode:
  - 1080p60 AVC/H.264 encoder
  - 1080p60 HEVC/H.265 encoder
- TrustZone support

1.4.2.11 Machine Learning: NPU (Neural Processing Unit)

- 2.3 TOP/s Neural Network performance available for user applications
  - Speech recognition (e.g., Deep Speech 2)
  - Image recognition (e.g., ResNet-50)
  - Object detection (e.g., MobileNet-SSD)

1.4.2.12 Display Interfaces

The chip has the following display support:

- Three LCDIF Display Controllers:
  - One LCDIF drives MIPI DSI
  - One LCDIF drives LVDS Tx
  - One LCDIF drives HDMI Tx
  - Support up to 1920x1200p60 display per LCDIF if no more than 2 instances used simultaneously, or 2x 1080p60 + 1x 4kp30 on HDMI if all 3 instances used simultaneously.
  - Supports 8-bit / 16-bit / 18-bit / 24-bit / 32-bit pixel depth
  - Supports one layer
- MIPI Interface:
  - One 4-lane MIPI DSI interface
  - Two 4-lane MIPI CSI interfaces
- One 4-lane or 8-lane LVDS interface
- ISI (Image Sensor Interface):
  - The ISI is a simple camera interface that supports image processing and transfer via a bus master interface for up to 2 cameras
- Two Camera ISP (Image Signal Processor):
  - When one camera is used, supports up to 12MP@30fps or 4kp45
  - When two cameras are used, each supports up to 1080p80
- HDMI 2.0a
  - HDMI 2.0a Tx supporting one display
    - Resolutions of: 740x480p60, 720x480p60, 1280x720p60, 1920x1080p60, 1920x1080p120, 3840x2160p30
  - Audio support

- 32 channel audio output support
- 1 S/PDIF audio eARC input support

1.4.2.13 Audio

Audio include the following:

- Audio DSP
- S/PDIF Input and Output, including a Raw Capture input mode
- Six external SAI (synchronous audio interface) modules supporting I2S, AC97, TDM, codec/DSP and DSD interfaces, comprising one SAI with 8 TX and 8 RX lanes, one SAI with 4 TX and 4 RX lanes, two SAI with 2 TX and 2 RX lanes, and two SAI with 1 TX and 1 RX lanes.
- PDM Microphone Interface module which supports up to 8-microphones (4 lanes)
- Asynchronous Sample Rate Converter (ASRC) module which supports:
  - Processing of up to 32 audio channels
  - 4 context groups
  - 8 kHz to 384 kHz sample rate
  - 1/16 to 8x sample rate conversion ratio

1.4.2.14 General Connectivity Interfaces

The chip contains a rich set of general connectivity interfaces, including:

- One PCI Express (PCIe):
  - Single lane supporting PCIe Gen 3
  - Dual mode operation to function as root complex or endpoint
  - Integrated PHY interface
  - Supports L1 low power substate
- Two USB 3.0 Type C controllers with integrated PHY interface
  - Backwards compatibility with USB 2.0
  - Spread spectrum clock support
- Three Ultra Secure Digital Host Controller (uSDHC) interfaces
  - MMC 5.1 compliance with HS400 DDR signaling to support up to 400 MB/sec
  - SD/SDIO 3.01 compliance with 200 MHZ SDR signaling to support up to 100 MB/sec
  - Support for SDXC (extended capacity)
- Two Ethernet controllers, capable of simultaneous operation

    - One Gigabit Ethernet controller with support for EEE, Ethernet AVB and IEEE1588
    - One Gigabit Ethernet controller with support for TSN, EEE, Ethernet AVB and IEEE1588
- Two controller area network (FlexCAN) modules, each optionally supporting flexible datarate (FD)
- Four universal asynchronous receiver/transmitter (UART) modules
- Six I2C modules
- Three SPI modules

1.4.2.15 Security

Security functions are enabled and accelerated by the following hardware:

- RDC – Resource Domain Controller:
    - Supports 4 domains and up to 8 regions
- Arm TrustZone including the TZ architecture:
    - Arm Cortex-A53 MPCore TrustZone support
- On-chip RAM (OCRAM) secure region protection using OCRAM controller
- High Assurance Boot (HAB)
- Cryptographic Acceleration and Assurance Module (CAAM)
    - Support Widevine and PlayReady content protection
    - Public Key Cryptography (PKHA) with RSA and Elliptic Curve (ECC) algorithms
    - Real-time integrity checker (RTIC)
    - DRM support for RSA, AES, 3DES, DES
    - True random number generation (RNG)
    - Manufacturing protection support
- Secure Non-Volatile Storage (SNVS), including
    - Secure Real Time Clock (SRTC)
- Secure JTAG Controller (SJC)

1.4.2.16 Multicore Support

Multicore support contains:

- Resource domain controller (RDC) to support isolation and safe sharing of system resources
- Messaging unit (MU)

- Hardware Semaphore (SEMA4)
- Shared bus topology

1.4.2.17 GPIO and Pin Multiplexing

- General-purpose input/output (GPIO) modules with interrupt capability
- Input/output multiplexing controller (IOMUXC) to provide centralized pad control

1.4.2.18 Power Management

The power management unit consists of:

- Temperature sensor with programmable trip points
- Flexible power domain partitioning with internal power switches to support efficient power management

1.4.2.19 System Debug

The system debug features are:

- Arm CoreSight debug and trace architecture
- Embedded Trace FIFO (ETF) with 4 KB internal storage to provide trace buffering
- Unified trace capability for Quad Cortex-A53 and Cortex-M7 CPUs
- Cross Triggering Interface (CTI)
- Support for 5-pin (JTAG) debug interfaces

1.4.3 Primary Boot Options

The i.MX 8M Plus supports the following boot devices:

- NAND FLASH (including SLC and MLC)
- SDIO / MMC / SDXC
- eSD 3.0/eMMC 5.1 (fast boot)
- SPI (serial FLASH)
- USB Serial Download

The Quad-A53 core on i.MX 8M Plus is enabled during boot as the primary core to handle the entire secure boot flow. The chip will always boot from the A53 core first, the M7 core will be held in reset during the A53 boot and won't run until it is enabled by the A53 core. The image for the M7 core will be loaded into memory and authenticated by the A53 core.

1.4.4 Endianness Support

This chip supports Little Endian mode only.

Chapter 2
Memory Map

2.1 Memory system overview

2.1.1 On-chip L1, L2 caches, TCM

Cortex-A53 MPcore Platform

- Level 1 Cache (4x per Cortex-A53 Core)
  - Instruction (32 KB)
  - Data (32 KB)
- Level 2 Cache, shared by the four Cortex-A53 cores:
  - Unified instruction and data (512 KB)

Cortex M7 Platform

- Cache
  - Instruction (32 KB)
  - Data (32 KB)
- Tightly-Coupled-Memory
  - DTCM (128KB)
  - ITCM (128KB)

2.1.2 On-chip memories

- Boot ROM (256 KB)
- On-Chip RAM - OCRAM (576 KB)
- Audio Processor System RAM - OCRAM_A (256KB)
- On-Chip RAM for State Retention - OCRAM_S (36 KB)

2.1.3 External L3 memories

The chip supports external memories, via the following memory interfaces / controllers:

- 32-bit DRAM Interface:
  - LPDDR4-4000 and DDR4-3200
- 8-bit NAND FLASH, including support for Raw MLC/SLC devices, BCH ECC up to 62-bit, and ONFI 3.2 compliance (clock rates up to 100 MHz and data rates up to 200 MB/sec)
- eMMC 5.1 FLASH
- SPI NOR FLASH
- FlexSPI FLASH with support for XIP (for example for use in a low-power mode when DRAM is not accessible) and support for one Octal SPI, or parallel read mode of two identical Quad SPI FLASH devices.

2.2 Cortex-A53 Memory Map

Start Address	End Address	Region	Size	Description
1_0000_0000	2_3FFF_FFFF	DDR Address	5120MB	DDR Memory (All modules except M7)
4000_0000	FFFF_FFFF	DDR Address	3072MB	DDR Memory (All modules)
3F00_0000	3FFF_FFFF	Reserved	16MB	Reserved
3E00_0000	3EFF_FFFF	Reserved	16MB	Reserved
3DC0_0000	3DFF_FFFF	DDRC	4MB	DDR PHY (Broadcast)
3D80_0000	3DBF_FFFF	DDRC	4MB	DDR PERF_MON
3D40_0000	3D7F_FFFF	DDRC	4MB	DDR CTL
3D10_0000	3D3F_FFFF	Reserved	3MB	Reserved
3D00_0000	3D0F_FFFF	DDRC	1MB	DDR BLK_CTRL
3C00_0000	3CFF_FFFF	DDRC	16MB	DDR PHY
3B00_0000	3BFF_FFFF	Audio DSP	16MB	Audio DSP. See Audio Processor Memory Map.
3A00_0000	3AFF_FFFF	Reserved	16MB	Reserved
3890_0000	39FF_FFFF	Reserved	23MB	Reserved
3880_0000	388F_FFFF	GIC	1MB	GIC REG
3870_0000	387F_FFFF	Reserved	1MB	Reserved
3850_0000	386F_FFFF	NPU	2MB	NPU
3834_0000	384F_FFFF	VPU	2MB	VPU
3833_0000	383F_FFFF		2MB	VPU BLK_CTRL
3832_0000	3842_FFFF		2MB	VPU VC8000E Encoder
3831_0000	3841_FFFF		2MB	VPU G2 Decoder
3830_0000	3830_FFFF		2MB	VPU G1 Decoder
3820_0000	382F_FFFF	USB2 REG	1MB	USB2 REG

Table continues on the next page...

Start Address	End Address	Region	Size	Description
3810_0000	381F_FFFF	USB1 REG	1MB	USB1 REG
3801_0000	380F_FFFF	Reserved	960KB	Reserved
3800_8000	3800_FFFF	GPU REG	32KB	GPU2D REG
3800_0000	3800_7FFF	GPU REG	32KB	GPU3D REG
3600_0000	37FF_FFFF	QSPI RX Buffers	32MB	Reserved
3400_0000	35FF_FFFF		32MB	QSPI1 RX Buffer
33C0_0000	33FF_FFFF	PCIe REG	4MB	Reserved
3380_0000	33BF_FFFF		4MB	PCIe1 REG
3310_0000	337F_FFFF	Reserved	7MB	Reserved
3301_0000	330F_FFFF	Reserved	960KB	Reserved
3300_8000	3300_FFFF	QSPI TX Buffers	32KB	QSPI1 TX Buffer
3300_0000	3300_7FFF	APBH DMA	32KB	APBH DMA
32C0_0000	32FF_FFFF	Periph (AIPS)	4MB	AIPS-4. See IP listing on separate map.
3290_0000	32BF_FFFF	Reserved	3MB	Reserved
3280_0000	328F_FFFF	Reserved	1MB	Reserved
3270_0000	327F_FFFF	Reserved	1MB	Reserved
3260_0000	326F_FFFF	Reserved	1MB	Reserved
3250_0000	325F_FFFF	Reserved	1MB	Reserved
3240_0000	324F_FFFF	Reserved	1MB	Reserved
3230_0000	323F_FFFF	Reserved	1MB	Reserved
3220_0000	322F_FFFF	Reserved	1MB	Reserved
3210_0000	321F_FFFF	Reserved	1MB	Reserved
3200_0000	320F_FFFF	Reserved	1MB	Reserved
3100_0000	31FF_FFFF	Reserved	16MB	Reserved
30C0_0000	30FF_FFFF	Periph (AIPS)	4MB	AIPS-5. See IP listing on separate map.
3080_0000	30BF_FFFF		4MB	AIPS-3. See IP listing on separate map.
3040_0000	307F_FFFF		4MB	AIPS-2. See IP listing on separate map.
3000_0000	303F_FFFF		4MB	AIPS-1. See IP listing on separate map.
2900_0000	2FFF_FFFF	Reserved	112MB	Reserved
2800_0000	28FF_FFFF	A53 / DAP	16MB	A53 / DAP
2000_0000	27FF_FFFF	Reserved	128MB	Reserved
1800_0000	1FFF_FFFF	PCIe-1	128MB	PCIe-1
0800_0000	17FF_FFFF	QSPI	256MB	QSPI
0400_0000	07FF_FFFF	Reserved	64MB	Reserved
0100_0000	03FF_FFFF	Reserved	48MB	Reserved
00C0_0000	00FF_FFFF	Reserved	4MB	Reserved
00B0_0000	00BF_FFFF	Reserved	1MB	Reserved
00A0_0000	00AF_FFFF	OCRAM	1MB	Reserved
0099_0000	009F_FFFF		448KB	Reserved
0090_0000	0098_FFFF		576KB	OCRAM
0084_0000	008F_FFFF	Reserved	768KB	Reserved

Table continues on the next page...

Start Address	End Address	Region	Size	Description
0082_0000	0083_FFFF	TCM	128KB	Reserved
0080_0000	0081_FFFF		128KB	DTCM
007E_0000	007F_FFFF		128KB	ITCM
007C_0000	007D_FFFF		128KB	Reserved
0070_0000	007B_FFFF	Reserved	768KB	Reserved
0060_0000	006F_FFFF	Reserved	1MB	Reserved
0050_0000	005F_FFFF	Reserved	1MB	Reserved
0040_0000	004F_FFFF	Reserved	1MB	Reserved
0020_0000	003F_FFFF	Reserved	2MB	Reserved
0019_0000	001F_FFFF	Reserved	448KB	Reserved
0018_9000	0018_FFFF	OCRAM_S	28KB	Reserved
0018_0000	0018_8FFF		36KB	OCRAM_S
0011_0000	0017_FFFF	Reserved	448KB	Reserved
0010_8000	0010_FFFF	CAAM	32KB	Reserved
0010_0000	0010_7FFF		32KB	CAAM (32K secure RAM)
0004_0000	000F_FFFF	Reserved	768KB	Reserved
0003_F000	0003_FFFF	Boot ROM	4KB	Boot ROM - Protected 4KB area
0000_0000	0003_EFFF		252KB	Boot ROM

2.3 Cortex-M7 Memory Map

Start Address	End Address	Region	Size	Description
E010_0000	FFFF_FFFF	Reserved	511MB	Reserved
E000_0000	E00F_FFFF	CM7 PPB	1MB	CM7 PPB
D800_0000	DFFF_FFFF	Reserved	128MB	Reserved
D000_0000	D7FF_FFFF	PCIe-1	128MB	PCIe-1
C000_0000	CFFF_FFFF	FLASH	256MB	QSPI
4000_0000	BFFF_FFFF	DDR Address	2048MB	DDR Memory
3F00_0000	3FFF_FFFF	Reserved	16MB	Reserved
3E00_0000	3EFF_FFFF	Reserved	16MB	Reserved
3DC0_0000	3DFF_FFFF	DDRC	4MB	DDR PHY (Broadcast)
3D80_0000	3DBF_FFFF	DDRC	4MB	DDR PERF_MON
3D40_0000	3D7F_FFFF	DDRC - CTL	4MB	DDR - CTL
3D10_0000	3D3F_FFFF	Reserved	3MB	Reserved
3D00_0000	3D0F_FFFF	DDRC	1MB	DDR BLK_CTRL
3C00_0000	3CFF_FFFF	DDRC	16MB	DDR PHY
3B00_0000	3BFF_FFFF	Audio DSP	16MB	Audio DSP. See Audio Processor Memory Map.

Table continues on the next page...

Start Address	End Address	Region	Size	Description
3A00_0000	3AFF_FFFF	Reserved	16MB	Reserved
3890_0000	39FF_FFFF	Reserved	23MB	Reserved
3880_0000	388F_FFFF	GIC	1MB	GIC. Please refer to GIC for details on the limitations for CM7 accessing GIC.
3870_0000	387F_FFFF	Reserved	1MB	Reserved
3850_0000	386F_FFFF	NPU	2MB	NPU
3834_0000	384F_FFFF	VPU	2MB	VPU
3833_0000	383F_FFFF		2MB	VPU BLK_CTRL
3832_0000	3842_FFFF		2MB	VPU VC8000E Encoder
3831_0000	3841_FFFF		2MB	VPU G2 Decoder
3830_0000	3830_FFFF		2MB	VPU G1 Decoder
3820_0000	382F_FFFF	USB2 REG	1MB	USB2 REG
3810_0000	381F_FFFF	USB1 REG	1MB	USB1 REG
3801_0000	380F_FFFF	Reserved	960KB	Reserved
3800_8000	3800_FFFF	GPU REG	32KB	GPU2D REG
3800_0000	3800_7FFF	GPU REG	32KB	GPU3D REG
3600_0000	37FF_FFFF	QSPI RX Buffers	32MB	Reserved
3400_0000	35FF_FFFF		32MB	QSPI1 RX Buffer
33C0_0000	33FF_FFFF	PCIe REG	4MB	Reserved
3380_0000	33BF_FFFF		4MB	PCIe1 REG
3310_0000	337F_FFFF	Reserved	7MB	Reserved
3301_0000	330F_FFFF	Reserved	960KB	Reserved
3300_8000	3300_FFFF	QSPI TX Buffers	32KB	QSPI1 TX Buffer
3300_0000	3300_7FFF	APBH DMA	32KB	APBH DMA
32C0_0000	32FF_FFFF	Periph (AIPS)	4MB	AIPS-4. See IP listing on separate map.
3290_0000	32BF_FFFF	Reserved	3MB	Reserved
3280_0000	328F_FFFF	Reserved	1MB	Reserved
3270_0000	327F_FFFF	Reserved	1MB	Reserved
3260_0000	326F_FFFF	Reserved	1MB	Reserved
3250_0000	325F_FFFF	Reserved	1MB	Reserved
3240_0000	324F_FFFF	Reserved	1MB	Reserved
3230_0000	323F_FFFF	Reserved	1MB	Reserved
3220_0000	322F_FFFF	Reserved	1MB	Reserved
3210_0000	321F_FFFF	Reserved	1MB	Reserved
3200_0000	320F_FFFF	Reserved	1MB	Reserved
3100_0000	31FF_FFFF	Reserved	16MB	Reserved
30C0_0000	30FF_FFFF	Periph (AIPS)	4MB	AIPS-5. See IP listing on separate map.
3080_0000	30BF_FFFF		4MB	AIPS-3. See IP listing on separate map.
3040_0000	307F_FFFF		4MB	AIPS-2. See IP listing on separate map.
3000_0000	303F_FFFF		4MB	AIPS-1. See IP listing on separate map.
2900_0000	2FFF_FFFF	Reserved	112MB	Reserved

Table continues on the next page...

Start Address	End Address	Region	Size	Description
2800_0000	28FF_FFFF	A53/ DAP	16MB	A53/ DAP
2400_0000	27FF_FFFF	Reserved	64MB	Reserved
2200_0000	23FF_FFFF	Reserved	32MB	Reserved
2100_0000	21FF_FFFF	Reserved	16MB	Reserved
2040_0000	20FF_FFFF	Reserved	12MB	Reserved
2030_0000	203F_FFFF	CM7 ALIAS SYSTEM	1MB	Reserved
2029_0000	202F_FFFF		448KB	Reserved
2020_0000	2028_FFFF		576KB	OCRAM
2019_0000	201F_FFFF		448KB	Reserved
2018_9000	2018_FFFF		28KB	Reserved
2018_0000	2018_8FFF		36KB	OCRAM_S
2011_0000	2017_FFFF		448KB	Reserved
2010_8000	2010_FFFF		32KB	Reserved
2010_0000	2010_7FFF		32KB	CAAM (32K secure RAM)
2006_0000	200F_FFFF	Reserved	640KB	Reserved
2002_0000	2005_FFFF	Reserved	256KB	Reserved
2000_0000	2001_FFFF	DTCM	128KB	DTCM
1FFE_0000	1FFF_FFFF	Reserved	128KB	Reserved
1000_0000	1FFD_FFFF	CM7 ALIAS CODE	262016KB	DDR Code alias
0800_0000	0FFF_FFFF		128MB	QSPI Code alias
0400_0000	07FF_FFFF		64MB	Reserved
0100_0000	03FF_FFFF	Reserved	48MB	Reserved
00C0_0000	00FF_FFFF	Reserved	4MB	Reserved
00B0_0000	00BF_FFFF	Reserved	1MB	Reserved
00A0_0000	00AF_FFFF	OCRAM	1MB	Reserved
0099_0000	009F_FFFF		448KB	Reserved
0090_0000	0098_FFFF		576KB	OCRAM
0082_8000	008F_FFFF	Reserved	864KB	Reserved
0082_0000	0082_7FFF	Reserved	32KB	Reserved
0080_0000	0081_FFFF	Reserved	128KB	Reserved
007E_0000	007F_FFFF	Reserved	128KB	Reserved
007D_8000	007D_FFFF	Reserved	32KB	Reserved
0070_0000	007D_7FFF	Reserved	864KB	Reserved
0060_0000	006F_FFFF	Reserved	1MB	Reserved
0050_0000	005F_FFFF	Reserved	1MB	Reserved
0040_0000	004F_FFFF	Reserved	1MB	Reserved
0020_0000	003F_FFFF	Reserved	2MB	Reserved
0019_0000	001F_FFFF	Reserved	448KB	Reserved
0018_9000	0018_FFFF	OCRAM_S	28KB	Reserved
0018_0000	0018_8FFF		36KB	OCRAM_S
0011_0000	0017_FFFF	Reserved	448KB	Reserved

Table continues on the next page...

Start Address	End Address	Region	Size	Description
0010_8000	0010_FFFF	CAAM	32KB	Reserved
0010_0000	0010_7FFF		32KB	CAAM (32K secure RAM)
0002_0000	000F_FFFF	Reserved	896KB	Reserved
0000_0000	0001_FFFF	ITCM	128KB	ITCM

2.4 DMA memory maps

The Smart DMA memory maps are defined in the following tables.

Table 2-1. SDMA1 Peripheral Memory Map

Address	Peripheral
0xF000	SPBA
0xE000	Reserved for SDMA internal registers
0xD000	CAN_FD2
0xC000	CAN_FD1
0xB000	Reserved
0xA000	Reserved
0x9000	UART2
0x8000	UART3
0x7000	Reserved for SDMA internal registers
0x6000	UART1
0x5000	Reserved
0x4000	eCSPI3
0x3000	eCSPI2
0x2000	eCSPI1
0x1000	Reserved
0x0000	Reserved for SDMA internal memory

Table 2-2. SDMA2/3 Peripheral Memory Map

Address	Peripheral
0xF000	SPBA
0xE000	Reserved for SDMA internal registers
0xD000	Reserved
0xC000	AUDIO XCVR RX (eARC)
0xB000	HDMI TX AUDLNK MST
0xA000	PDM (MICFIL)
0x9000	ASRC

Table continues on the next page...

Table 2-2. SDMA2/3 Peripheral Memory Map (continued)

Address	Peripheral
0x8000	SAI7
0x7000	Reserved for SDMA internal registers
0x6000	SAI6
0x5000	SAI5
0x4000	Reserved
0x3000	SAI3
0x2000	SAI2
0x1000	SAI1
0x0000	Reserved for SDMA internal memory

2.5 AIPS Memory Maps

Table 2-3. AIPS1 Memory Map

Start Address	End Address	Region	NIC Port	Size
303F_0000	303F_FFFF	AIPS-1 (s_b_0)	Reserved	64KB
303E_0000	303E_FFFF		CSU	64KB
303D_0000	303D_FFFF		RDC	64KB
303C_0000	303C_FFFF		SEMAPHORE2	64KB
303B_0000	303B_FFFF		SEMAPHORE1	64KB
303A_0000	303A_FFFF		GPC	64KB
3039_0000	3039_FFFF		SRC	64KB
3038_0000	3038_FFFF		CCM	64KB
3037_0000	3037_FFFF		SNVS_HP	64KB
3036_0000	3036_FFFF		ANA_PLL	64KB
3035_0000	3035_FFFF		OCOTP_CTRL	64KB
3034_0000	3034_FFFF		IOMUXC_GPR	64KB
3033_0000	3033_FFFF		IOMUXC	64KB
3032_0000	3032_FFFF		Reserved	64KB
3031_0000	3031_FFFF		Reserved	64KB
3030_0000	3030_FFFF		Reserved	64KB
302F_0000	302F_FFFF		GPT3	64KB
302E_0000	302E_FFFF		GPT2	64KB
302D_0000	302D_FFFF		GPT1	64KB
302C_0000	302C_FFFF		Reserved	64KB
302B_0000	302B_FFFF		Reserved	64KB
302A_0000	302A_FFFF		WDOG3	64KB

Table continues on the next page...

Table 2-3. AIPS1 Memory Map (continued)

Start Address	End Address	Region	NIC Port	Size
3029_0000	3029_FFFF		WDOG2	64KB
3028_0000	3028_FFFF		WDOG1	64KB
3027_0000	3027_FFFF		ANA_OSC	64KB
3026_0000	3026_FFFF		TMU (ANA_TSENSOR)	64KB
3025_0000	3025_FFFF		Reserved	64KB
3024_0000	3024_FFFF		GPIO5	64KB
3023_0000	3023_FFFF		GPIO4	64KB
3022_0000	3022_FFFF		GPIO3	64KB
3021_0000	3021_FFFF		GPIO2	64KB
3020_0000	3020_FFFF		GPIO1	64KB
301F_0000	301F_FFFF		AIPS1_Configuration	64KB
3010_0000	301E_FFFF	AIPS-1 Glob. Module Enable	Reserved	960KB
3000_0000	300F_FFFF		Reserved	1024KB

Table 2-4. AIPS2 Memory Map

Start Address	End Address	Region	NIC Port	Size
307F_0000	307F_FFFF	AIPS-2 (s_b_1)	QoSC	64KB
307E_0000	307E_FFFF		Reserved	64KB
307D_0000	307D_FFFF		PERFMON2	64KB
307C_0000	307C_FFFF		PERFMON1	64KB
307B_0000	307B_FFFF		Reserved	64KB
307A_0000	307A_FFFF		Reserved	64KB
3079_0000	3079_FFFF		Reserved	64KB
3078_0000	3078_FFFF		Reserved	64KB
3077_0000	3077_FFFF		Reserved	64KB
3076_0000	3076_FFFF		Reserved	64KB
3075_0000	3075_FFFF		Reserved	64KB
3074_0000	3074_FFFF		Reserved	64KB
3073_0000	3073_FFFF		Reserved	64KB
3072_0000	3072_FFFF		Reserved	64KB
3071_0000	3071_FFFF		Reserved	64KB
3070_0000	3070_FFFF		GPT4	64KB
306F_0000	306F_FFFF		GPT5	64KB
306E_0000	306E_FFFF		GPT6	64KB
306D_0000	306D_FFFF		Reserved	64KB
306C_0000	306C_FFFF		System_Counter_CTRL	64KB
306B_0000	306B_FFFF		System_Counter_CMP	64KB
306A_0000	306A_FFFF		System_Counter_RD	64KB

Table continues on the next page...

Table 2-4. AIPS2 Memory Map (continued)

Start Address	End Address	Region	NIC Port	Size
3069_0000	3069_FFFF		PWM4	64KB
3068_0000	3068_FFFF		PWM3	64KB
3067_0000	3067_FFFF		PWM2	64KB
3066_0000	3066_FFFF		PWM1	64KB
3065_0000	3065_FFFF		Reserved	64KB
3064_0000	3064_FFFF		Reserved	64KB
3063_0000	3063_FFFF		Reserved	64KB
3062_0000	3062_FFFF		Reserved	64KB
3061_0000	3061_FFFF		Reserved	64KB
3060_0000	3060_FFFF		Reserved	64KB
305F_0000	305F_FFFF		AIPS2_configuration	64KB
3050_0000	305E_FFFF	AIPS-2 Glob. Module Enable	Reserved	960KB
3040_0000	304F_FFFF		Reserved	1024KB

Table 2-5. AIPS3 Memory Map

Start Address	End Address	Region	NIC Port	Size
30BF_0000	30BF_FFFF	AIPS-3 (s_b_2)	ENET2_TSN	64KB
30BE_0000	30BE_FFFF		ENET1	64KB
30BD_0000	30BD_FFFF		SDMA1	64KB
30BC_0000	30BC_FFFF		Reserved	64KB
30BB_0000	30BB_FFFF		QSPI	64KB
30BA_0000	30BA_FFFF		Reserved	64KB
30B9_0000	30B9_FFFF		Reserved	64KB
30B8_0000	30B8_FFFF		Reserved	64KB
30B7_0000	30B7_FFFF		Reserved	64KB
30B6_0000	30B6_FFFF		uSDHC3	64KB
30B5_0000	30B5_FFFF		uSDHC2	64KB
30B4_0000	30B4_FFFF		uSDHC1	64KB
30B3_0000	30B3_FFFF		Reserved	64KB
30B2_0000	30B2_FFFF		Reserved	64KB
30B1_0000	30B1_FFFF		Reserved	64KB
30B0_0000	30B0_FFFF		Reserved	64KB
30AF_0000	30AF_FFFF		Reserved	64KB
30AE_0000	30AE_FFFF		I2C6	64KB
30AD_0000	30AD_FFFF		I2C5	64KB
30AC_0000	30AC_FFFF		SEMAPHORE_HS	64KB
30AB_0000	30AB_FFFF		MU_1_B (A53, M7)	64KB
30AA_0000	30AA_FFFF		MU_1_A (A53,M7)	64KB

Table continues on the next page...

Table 2-5. AIPS3 Memory Map (continued)

Start Address	End Address	Region	NIC Port	Size
30A9_0000	30A9_FFFF		Reserved	64KB
30A8_0000	30A8_FFFF		IRQ_STEER (Audio Processor)	64KB
30A7_0000	30A7_FFFF		Reserved	64KB
30A6_0000	30A6_FFFF		UART4	64KB
30A5_0000	30A5_FFFF		I2C4	64KB
30A4_0000	30A4_FFFF		I2C3	64KB
30A3_0000	30A3_FFFF		I2C2	64KB
30A2_0000	30A2_FFFF		I2C1	64KB
30A1_0000	30A1_FFFF		Reserved	64KB
30A0_0000	30A0_FFFF		Reserved	64KB
309F_0000	309F_FFFF		AIPS3_Configuration	64KB
3094_0000	309E_FFFF	AIPS-3 Glob. Module Enable 1	Reserved	704KB
3090_0000	3093_FFFF		CAAM	256KB
308F_0000	308F_FFFF	AIPS-3 (s_b_2, via SPBA) Glob. Module Enable 0	SPBA1	64KB
308E_0000	308E_FFFF		Reserved	64KB
308D_0000	308D_FFFF		FlexCAN2	64KB
308C_0000	308C_FFFF		FlexCAN1	64KB
308B_0000	308B_FFFF		Reserved	64KB
308A_0000	308A_FFFF		Reserved	64KB
3089_0000	3089_FFFF		UART2	64KB
3088_0000	3088_FFFF		UART3	64KB
3087_0000	3087_FFFF		Reserved	64KB
3086_0000	3086_FFFF		UART1	64KB
3085_0000	3085_FFFF		Reserved	64KB
3084_0000	3084_FFFF		eCSPI3	64KB
3083_0000	3083_FFFF		eCSPI2	64KB
3082_0000	3082_FFFF		eCSPI1	64KB
3081_0000	3081_FFFF		Reserved	64KB
3080_0000	3080_FFFF		Reserved	64KB

Table 2-6. AIPS4 Memory Map

Start Address	End Address	Region	NIC Port	Size
32FF_0000	32FF_FFFF	AIPS-4 (s_h_10)	Reserved	64KB
32FE_0000	32FE_FFFF		Reserved	64KB
32FD_0000	32FD_FFFF		HDMI_TX	64KB
32FC_0000	32FC_FFFF			64KB

Table continues on the next page...

Table 2-6. AIPS4 Memory Map (continued)

Start Address	End Address	Region	NIC Port	Size
			See HDMI_TX Subsystem Memory Map for more information.	
32FB_0000	32FB_FFFF		Reserved	64KB
32FA_0000	32FA_FFFF		Reserved	64KB
32F9_0000	32F9_FFFF		Reserved	64KB
32F8_0000	32F8_FFFF		TZASC	64KB
32F7_0000	32F7_FFFF		Reserved	64KB
32F6_0000	32F6_FFFF		Reserved	64KB
32F5_0000	32F5_FFFF		Reserved	64KB
32F4_0000	32F4_FFFF		Reserved	64KB
32F3_0000	32F3_FFFF		Reserved	64KB
32F2_0000	32F2_FFFF		Reserved	64KB
32F1_0000	32F1_FFFF		HSIO BLK_CTL	64KB
32F0_0000	32F0_FFFF		PCIE_PHY1	64KB
32EF_0000	32EF_FFFF		Reserved	64KB
32EE_0000	32EE_FFFF		Reserved	64KB
32ED_0000	32ED_FFFF		Reserved	64KB
32EC_0000	32EC_FFFF		MEDIA BLK_CTL	64KB
32EB_0000	32EB_FFFF		Reserved	64KB
32EA_0000	32EA_FFFF		Reserved	64KB
32E9_0000	32E9_FFFF		LCDIF2	64KB
32E8_0000	32E8_FFFF		LCDIF1	64KB
32E7_0000	32E7_FFFF		Reserved	64KB
32E6_0000	32E6_FFFF		MIPI_DSI1	64KB
32E5_0000	32E5_FFFF		MIPI_CSI2	64KB
32E4_0000	32E4_FFFF		MIPI_CSI1	64KB
32E3_0000	32E3_FFFF		IPS Dewarp	64KB
32E2_0000	32E2_FFFF		ISP2	64KB
32E1_0000	32E1_FFFF		ISP1	64KB
32E0_0000	32E0_FFFF		ISI	64KB
32DF_0000	32DF_FFFF		AIPS4_configuration	64KB
32D0_0000	32DE_FFFF	AIPS-4 Glob. Module Enable	Reserved	960KB
32C0_0000	32CF_FFFF		Reserved	1024KB

Table 2-7. AIPS5 Memory Map

Start Address	End Address	Region	NIC Port	Size
30FF_0000	30FF_FFFF	AIPS-5	Reserved	64KB

Table continues on the next page...

Table 2-7. AIPS5 Memory Map (continued)

Start Address	End Address	Region	NIC Port	Size
30FE_0000	30FE_FFFF		Reserved	64KB
30FD_0000	30FD_FFFF		Reserved	64KB
30FC_0000	30FC_FFFF		Reserved	64KB
30FB_0000	30FB_FFFF		Reserved	64KB
30FA_0000	30FA_FFFF		Reserved	64KB
30F9_0000	30F9_FFFF		Reserved	64KB
30F8_0000	30F8_FFFF		Reserved	64KB
30F7_0000	30F7_FFFF		Reserved	64KB
30F6_0000	30F6_FFFF		Reserved	64KB
30F5_0000	30F5_FFFF		Reserved	64KB
30F4_0000	30F4_FFFF		Reserved	64KB
30F3_0000	30F3_FFFF		Reserved	64KB
30F2_0000	30F2_FFFF		Reserved	64KB
30F1_0000	30F1_FFFF		Reserved	64KB
30F0_0000	30F0_FFFF		Reserved	64KB
30EF_0000	30EF_FFFF		Reserved	64KB
30EE_0000	30EE_FFFF		Reserved	64KB
30ED_0000	30ED_FFFF		Reserved	64KB
30EC_0000	30EC_FFFF		Reserved	64KB
30EB_0000	30EB_FFFF		Reserved	64KB
30EA_0000	30EA_FFFF		Reserved	64KB
30E9_0000	30E9_FFFF		MU_3_B (M7, Audio Processor)	64KB
30E8_0000	30E8_FFFF		MU_3_A (M7, Audio Processor)	64KB
30E7_0000	30E7_FFFF		MU_2_B (A53, Audio Processor	64KB
30E6_0000	30E6_FFFF		MU_2_A (A53, Audio Processor)	64KB
30E5_0000	30E5_FFFF		eDMA Channels[31:16] at 4KB/ch	64KB
30E4_0000	30E4_FFFF		eDMA Channels[15:0] at 4KB/ch	64KB
30E3_0000	30E3_FFFF		eDMA Management Page	64KB
30E2_0000	30E2_FFFF		AUDIO BLK_CTRL	64KB
30E1_0000	30E1_FFFF		SDMA2	64KB
30E0_0000	30E0_FFFF		SDMA3	64KB
30DF_0000	30DF_FFFF		AIPS5_Configuration	64KB
30D0_0000	30DE_FFFF	AIPS-5 Glob. Module Enable 1	Reserved	960KB

Table continues on the next page...

Table 2-7. AIPS5 Memory Map (continued)

Start Address	End Address	Region	NIC Port	Size
30CF_0000	30CF_FFFF	AIPS-5 (via SPBA) Glob. Module Enable 0	SPBA2	64KB
30CE_0000	30CE_FFFF		Reserved	64KB
30CD_0000	30CD_FFFF		Reserved	64KB
30CC_0000	30CC_FFFF		AUDIO XCVR RX (eARC)	64KB
30CB_0000	30CB_FFFF		HDMI TX AUDLNK MSTR	64KB
30CA_0000	30CA_FFFF		PDM (MICFIL)	64KB
30C9_0000	30C9_FFFF		ASRC	64KB
30C8_0000	30C8_FFFF		SAI7	64KB
30C7_0000	30C7_FFFF		Reserved	64KB
30C6_0000	30C6_FFFF		SAI6	64KB
30C5_0000	30C5_FFFF		SAI5	64KB
30C4_0000	30C4_FFFF		Reserved	64KB
30C3_0000	30C3_FFFF		SAI3	64KB
30C2_0000	30C2_FFFF		SAI2	64KB
30C1_0000	30C1_FFFF		SAI1	64KB
30C0_0000	30C0_FFFF		Reserved	64KB

2.6 DAP Memory Map

Table 2-8. DAP Memory Map Table

Start Address	End Address	Size	Allocation
28C0_A000	28FF_FFFF	4056KB	Reserved
28C0_9000	28C0_9FFF	4KB	HUGO_CXCTI1
28C0_8000	28C0_8FFF	4KB	HUGO_CXCTI0
28C0_7000	28C0_7FFF	4KB	CXTPIU
28C0_6000	28C0_6FFF	4KB	CXTMC_ETR
28C0_5000	28C0_5FFF	4KB	ATB_REPLICATOR
28C0_4000	28C0_4FFF	4KB	CXTMC_ETB
28C0_3000	28C0_3FFF	4KB	HUGO_ATB_FUNNEL
28C0_2000	28C0_2FFF	4KB	CXTSGEN_READ
28C0_1000	28C0_1FFF	4KB	CXTSGEN_CTRL
28C0_0000	28C0_0FFF	4KB	HUGO ROM Table
28B5_0000	28BF_FFFF	704KB	Reserved
28B4_0000	28B4_FFFF	64KB	Reserved

Table continues on the next page...

Table 2-8. DAP Memory Map Table (continued)

Start Address	End Address	Size	Allocation
28B3_0000	28B3_FFFF	64KB	Reserved
28B2_0000	28B2_FFFF	64KB	Reserved
28B1_0000	28B1_FFFF	64KB	Reserved
28A5_0000	28B0_FFFF	768KB	Reserved
28A4_0000	28A4_FFFF	64KB	Reserved
28A3_0000	28A3_FFFF	64KB	Reserved
28A2_0000	28A2_FFFF	64KB	Reserved
28A1_0000	28A1_FFFF	64KB	Reserved
2895_0000	28A0_FFFF	768KB	Reserved
2894_0000	2894_FFFF	64KB	Reserved
2893_0000	2893_FFFF	64KB	Reserved
2892_0000	2892_FFFF	64KB	Reserved
2891_0000	2891_FFFF	64KB	Reserved
2885_0000	2890_FFFF	768KB	Reserved
2884_0000	2884_FFFF	64KB	Reserved
2883_0000	2883_FFFF	64KB	Reserved
2882_0000	2882_FFFF	64KB	Reserved
2881_0000	2881_FFFF	64KB	Reserved
2880_0000	2880_FFFF	64KB	Audio Processor Debug
2875_0000	287F_FFFF	704KB	Reserved
2874_0000	2874_FFFF	64KB	MP4-CPU3 Trace
2873_0000	2873_FFFF	64KB	MP4-CPU3 PMU
2872_0000	2872_FFFF	64KB	MP4-CPU3 CTI
2871_0000	2871_FFFF	64KB	MP4-CPU3 Debug
2865_0000	2870_FFFF	768KB	Reserved
2864_0000	2864_FFFF	64KB	MP4-CPU2 Trace
2863_0000	2863_FFFF	64KB	MP4-CPU2 PMU
2862_0000	2862_FFFF	64KB	MP4-CPU2 CTI
2861_0000	2861_FFFF	64KB	MP4-CPU2 Debug
2855_0000	2860_FFFF	768KB	Reserved
2854_0000	2854_FFFF	64KB	MP4-CPU1 Trace
2853_0000	2853_FFFF	64KB	MP4-CPU1 PMU
2852_0000	2852_FFFF	64KB	MP4-CPU1 CTI
2851_0000	2851_FFFF	64KB	MP4-CPU1 Debug
2845_0000	2850_FFFF	768KB	Reserved
2844_0000	2844_FFFF	64KB	MP4-CPU0 Trace
2843_0000	2843_FFFF	64KB	MP4-CPU0 PMU
2842_0000	2842_FFFF	64KB	MP4-CPU0 CTI
2841_0000	2841_FFFF	64KB	MP4-CPU0 Debug

Table continues on the next page...

Table 2-8. DAP Memory Map Table (continued)

Start Address	End Address	Size	Allocation
2840_0000	2840_FFFF	64KB	MP4 ROM Table
2801_0000	283F_FFFF	4032KB	Reserved
2800_0000	2800_FFFF	64KB	DAP ROM Table

2.7 Audio Processor Memory Map

This section includes the Audio Processor Memory Map portion of the System Memory Map.

Table 2-9. Audio Processor Memory Map

Start Address	End Address	Size	Allocation
3B77_0000	3BFF_FFFF	8768KB	Reserved
3B74_0000	3B76_FFFF	192KB	Reserved
3B70_0000	3B73_FFFF	256KB	Audio Processor System RAM (OCRAM_A)
3B6F_8800	3B6F_FFFF	30KB	Reserved
3B6F_8000	3B6F_87FF	2KB	Audio Processor Local Memory (Instr Ram0)
3B6F_0000	3B6F_7FFF	32KB	Audio Processor Local Memory (Data Ram1)
3B6E_8000	3B6E_FFFF	32KB	Audio Processor Local Memory (Data Ram0)
3B00_0000	3B6E_7FFF	7072KB	Reserved

2.8 HDMI_TX Subsystem Memory Map

The HDMI_TX subsystem address space is configured as shown in the table below.

Table 2-10. HDMI_TX Subsystem Memory Map

Start Address	End Address	Region	Size	Description
32FD_8000	32FD_FFFF	HDMI TX Controller & PHY	32KB	0xFF00-0xFFFF - HDMI TX PHY 0x8000-0xFE31 - HDMI TX Controller
32FD_4000	32FD_7FFF	Reserved	16KB	Reserved
32FD_3000	32FD_3FFF	HDMI TX TRNG	4KB	HDMI TX TRNG
32FD_2000	32FD_2FFF	Reserved	4KB	Reserved
32FD_1000	32FD_1FFF	Reserved	4KB	Reserved
32FD_0000	32FD_0FFF	Reserved	4KB	Reserved

Table continues on the next page...

Table 2-10. HDMI_TX Subsystem Memory Map (continued)

Start Address	End Address	Region	Size	Description
32FC_F000	32FC_FFFF	Reserved	4KB	Reserved
32FC_E000	32FC_EFFF	Reserved	4KB	Reserved
32FC_D000	32FC_DFFF	Reserved	4KB	Reserved
32FC_C000	32FC_CFFF	Reserved	4KB	Reserved
32FC_7000	32FC_BFFF	Reserved	20KB	Reserved
32FC_6000	32FC_6FFF	LCDIF3	4KB	-
32FC_5000	32FC_5FFF	Reserved	4KB	Reserved
32FC_4800	32FC_4FFF	BLK_CTRL	4KB	HDMI TX Parallel Audio Interface (HTX_PAI)
32FC_4000	32FC_47FF		4KB	HDMI TX Parallel Video Interface (HTX_PVI)
32FC_3000	32FC_3FFF	Reserved	4KB	Reserved
32FC_2000	32FC_2FFF	IRQ_STEER	4KB	-
32FC_1000	32FC_1FFF	Reserved	4KB	Reserved
32FC_0000	32FC_0FFF	BLK_CTRL	4KB	HDMI TX BLK_CTRL

Chapter 3
Security

3.1 System Security

3.1.1 Overview

The security system modules are described in the sections below.

NOTE

This chapter provides a brief overview of the security features that are available for this chip. Please refer to the chip's Security Reference Manual for additional details on the security features.

3.1.2 Central Security Unit (CSU)

The chip uses the CSU to manage security for all masters/slaves that don’t directly support security like the CPU core.

3.1.3 Cryptographic Acceleration and Assurance Module (CAAM)

CAAM is the chip's cryptographic acceleration and assurance module, which serves as NXP's cryptographic acceleration and offloading hardware to create a modular and scalable acceleration and assurance engine. It also implements block encryption algorithms, stream cipher algorithms, hashing algorithms, public key algorithms, run-time integrity checking, a secure memory controller, and a hardware random number generator.

CAAM supports the following key features:

- PKHA block to support Public Key Cryptography with RSA 4096 and ECC algorithms

- 3 Job Rings
- RTIC (real time integrity checking)
- AES, DES, 3DES support
- Widevine cipher text stealing (AES-CBC-CTS mode)
- PlayReady content protection

In order to provide better video content protection, CAAM also supports the domain based resource protection.

3.1.4 Secure Non-Volatile Storage (SNVS)

Secure Non-Volatile Storage (SNVS) is a companion logic block to the Cryptographic Acceleration and Assurance Module (CAAM). This block is the chip's central reporting point for security-relevant events, such as the success or failure of boot software validation and the detection of potential security compromises. This security event information allows the SNVS to determine whether the chip is in a trustworthy state.

3.1.5 On-Chip OTP Controller (OCOTP_CTRL)

The OCOPT_CTRL is used to control the 8K bit OTP in the chip. It supports:

- Register interface to allow SW to read, override or program the fuse.
- Flexible permission control to the fuse, including read-protect, override-protect, program-protect, and lock.
- Programming sequence to allow single bit to be programmed individually, and also allow the non-programmed bit to be programmed later.

For the 8K bits fuse, following bits are reserved for various IP requirements:

- Fuse control to disable each of the four Arm Cortex A53 cores.
- Fuse control to disable individual IP modules such as MIPI, ENET, USB, etc.

3.1.6 Resource Domain Controller (RDC)

The chip uses domain based resource control architecture for the memory/peripheral resource sharing and isolation between the Quad-core Cortex A53 platform, Cortex M7 core, and other bus masters. RDC supports flexible configuration on IP’s access permission, each individual IP can, for example, be configured as A-core only or M-core only. RDC also supports multiple regions with flexible access permission control for each memory space. The memory region access control is defined in the table below. The security and exclusive support for DRAM will be done by the TZASC and the DRAM controller itself.

NOTE

The RDC and TrustZone features are completely independent of each other but will be used together.

Table 3-1. Memory Region Access Control

Memory Space	Regions Supported	Granularity	Security Support	Exclusive Support
DEBUG (DAP)	4 regions	4K Bytes	Enable	-
FlexSPI	8 regions	4K Bytes	Enable	Disable
OCRAM	5 regions	128 Bytes	Disable	Enable
OCRAM_S	5 regions	128 Bytes	Disable	Enable
Cortex-M7 TCM	5 regions	128 Bytes	Enable	Disable
GIC	4 regions	4K Bytes	Enable	-
GPUMIX	8 regions	4K Bytes	Enable	-
DRAM	8 regions	4K Bytes	Enabled by TZASC	Enabled by NoC
DDRC (DRAM registers)	5 regions	4K Bytes	Enable	Disable
HSIOMIX (PCIe and USB)	8 regions	4K Bytes	Enable	-
VPUMIX	4 regions	4K Bytes	Enable	Disable
MLMIX (NPU)	4 regions	4K Bytes	Enable	-
Audio Processor TCM	4 regions	4K Bytes	Enable	Disable
OCRAM_A	5 regions	128 Bytes	Disable	Disable

NOTE

The Security Support column in the table indicates if the Trustzone Secure-World attribute is supported for the respective memory space.

3.1.7 TrustZone

TrustZone security architecture is supported in the chip. For on-chip RAM, both OCRAM/OCRAM_S have TrustZone access control support through its TZ wrapper logic. One region with configurable address range of the OCRAM/OCRAM_S can be set to TZ access only. DRAM has a dedicated TZASC block that can support up to 16 configurable memory regions.

3.2 Resource Domain Controller (RDC)

3.2.1 Overview

The Resource Domain Controller (RDC) provides robust support for the isolation of destination memory mapped locations such as peripherals and memory to a single cluster, a bus master, or set of clusters and bus masters.

3.2.1.1 Block Diagram

Many of today's processors have multiple clusters for increased performance and flexibility. In some cases, the clusters serve different functions (for example, user level applications versus real time machine control) and in such cases the software for each cluster may be developed by different providers.

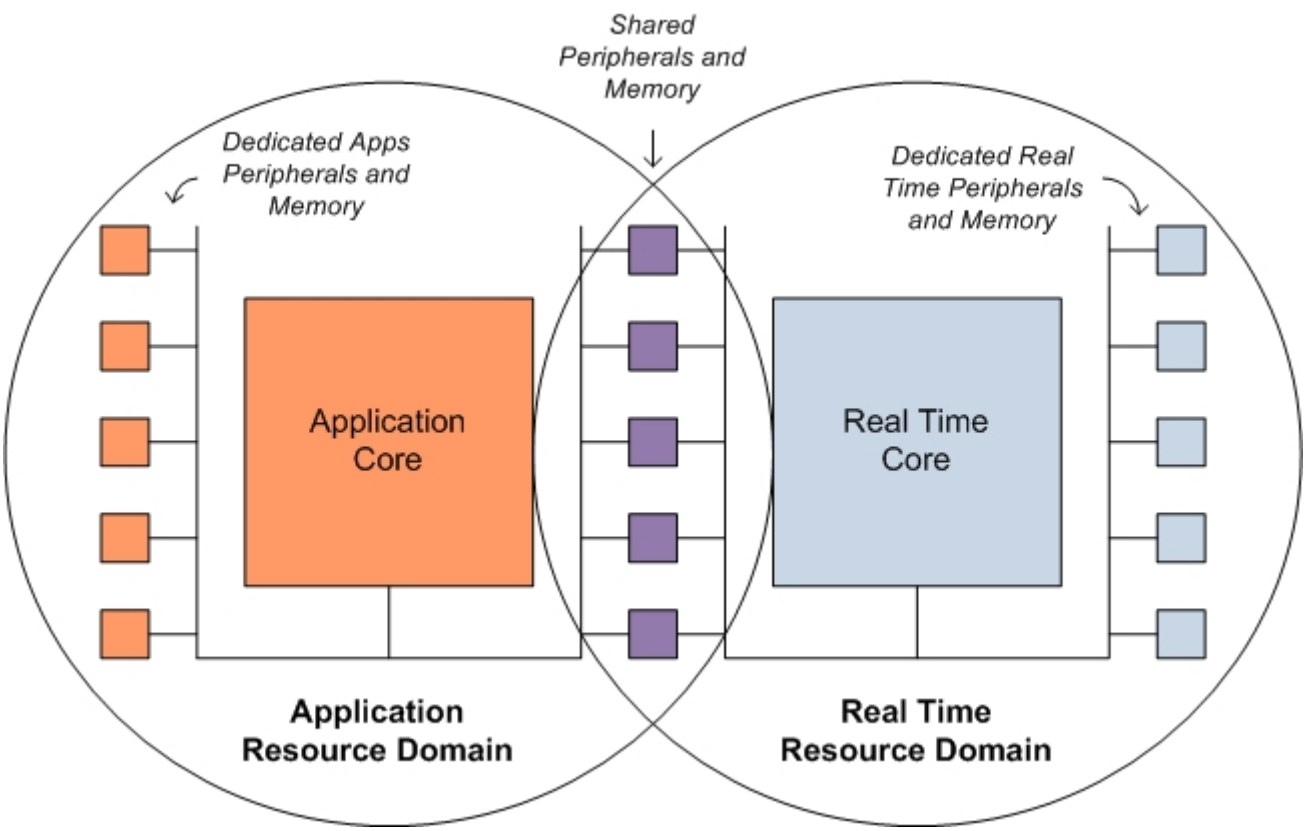


Figure 3-1. Dedicated and Shared Peripherals

For efficiency reasons the code on the clusters may share chip resources such as peripherals and memory. The sharing of chip resources between the somewhat independent processing domains allows for the opportunity of data collisions where information stored in peripherals or memory by a process on one cluster is overwritten by software running on another cluster. Without careful collaboration between the two operating systems inadvertent malfunction or degradation in performance may result.

The RDC provides a mechanism to allow boot time configuration code to establish resource domains by assigning clusters, bus masters, peripherals and memory regions to domain identifiers. Once configured, bus transactions are monitored to restrict accesses initiated by clusters and bus masters to their respective peripherals and memory.

For shared peripherals, the RDC provides a semaphore-based locking mechanism to provide for temporary exclusivity while the domain software uses the peripheral. Once the software of one domain has finished the task and finished with the peripheral then it may release the semaphore making the peripheral available to the other domain.

3.2.1.2 Features

Resource domain subsystem has the following features:

- Assignment of clusters, bus masters, peripherals, and memory regions to a resource domain
- Fixed memory resolution of 128 Bytes for small address spaces and 4 KB for large address spaces
- Four resource domain identifiers (DIDs)
- Memory read/write access controls for each resource domain and region
- Optional semaphore-based, hardware-enforced exclusive access of shared peripherals to a resource domain
- Prioritized access permissions for overlapping memory regions
- Automatic restoration of resource domain access permissions to memory regions in the power-down domain

3.2.2 Functional Description

The RDC is the central location for creation of isolated resource domains and for the enablement of semaphore-based access also known as “safe sharing”. Configuration software assigns one of resource domain identifiers to each cluster and bus master, and allocates each memory region and peripheral to one or more resource domains.

Memory Read or Write access privileges for each resource domain are declared for each memory region. In addition, the software configuration determines which shared peripherals (those peripherals allocated to more than one domain) require safe sharing by setting the semaphore-required configuration for each peripheral.

The RDC configuration information is sent to the fabric ports, memories gaskets, semaphore controller, and peripherals to control access based on domain assignments. The fabric uses the domain identifier associated with each port to include this information along with the bus transaction. When the slave gasket encounters a bus transaction, it makes a comparison of the transaction domain ID to the RDC-provided list of allowed domains. If the transaction domain ID is on the list, then access may be permitted.

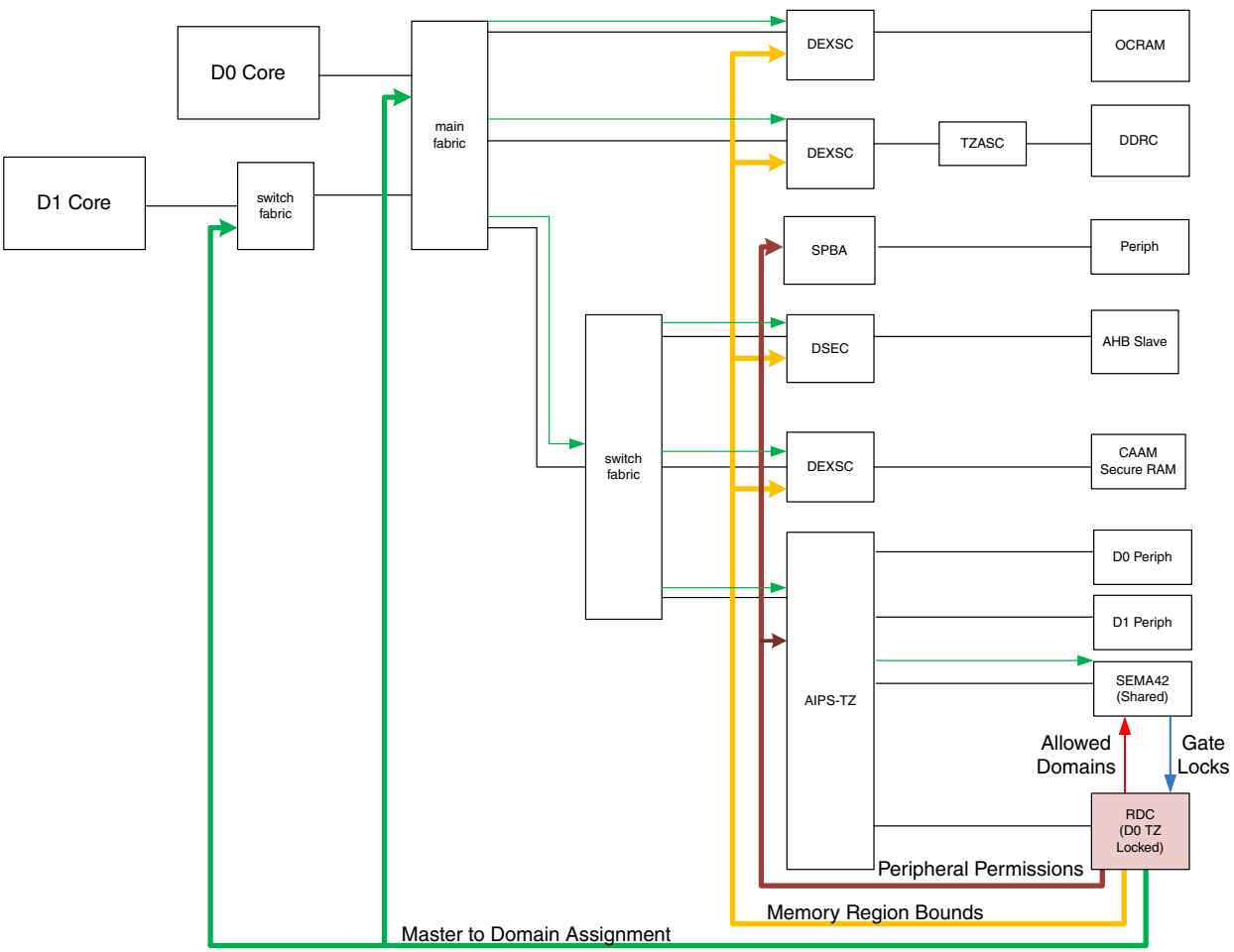


Figure 3-2. Example RDC Connections

For shared peripherals, RDC permits more than one domain access to a single peripheral. RDC also provides three ways to control synchronized use of shared peripherals. These methods include hardware-enforced synchronization, software-based semaphores, or no synchronization. The latter may be suitable for well-tuned multi-cluster operating systems that handle synchronization in the cluster platform, for instance.

For hardware-enforced synchronization, also known as "safe sharing", ownership of the peripheral must be claimed in the semaphore controller before access is allowed to the shared peripheral. Each shared peripheral has a corresponding Peripheral Domain Access Permissions (PDAP) register. When the Semaphore Required (SREQ) bit in a PDAP register is set, a master cannot access this shared peripheral until obtaining a semaphore. During the time that the domain has the semaphore in possession, its bus masters have exclusive access to the peripheral.

When the semaphore is released, then no domain masters have access until the semaphore is obtained again. When the SREQ bit is set, RDC module does not allow masters to obtain semaphores of peripherals to which the domain is not allocated; the master must have designated access in the D-bits of the corresponding PDAP register (for example, D1R bit set for Domain 1 access of the shared peripheral). There is a one-to-one mapping between the semaphore controller gate and the resource domain controller peripheral. The mapping of PDAP registers and peripherals can be found in the Peripheral Map section of the RDC chapter.

3.2.2.1 Domain ID

The RDC provides for an isolation of domain resources by use of an identifier called the Domain ID (DID). A cluster and its resources including memory, bus masters, and peripherals are all associated with a single DID. When software or a DMA attempts to access a peripheral or memory, the corresponding bus transaction includes the DID along with the other bus control information such as Read, Write, and privilege mode.

3.2.2.2 Resource Assignment

The RDC allows assignment of peripherals and memories to one or more domains while each bus master or cluster is placed in one of several domains. The masters are assigned a domain in the Master Domain Assignment (MDA) register. A peripheral is given R/W access permissions to each domain in the PDAP register. Memory regions are bound by address space in start and end registers, i.e. MRSA and MREA. Each memory region is assigned one or more allowed domains and R/W permissions in the Memory Region Control (MRC) register. Memory regions must be enabled before the permissions are active. Otherwise the permissions are not restricted.

The RDC itself should be isolated to ensure that only a trustworthy resource manager can configure the RDC registers. This process may either be present initially during secure boot, or during the runtime in the secure world, for example. If the operating system does not support a runtime trusted execution, then during the secure boot process the RDC configuration can be locked to prevent further modification after the operating systems are running.

NOTE

The CCM supports multi-cluster awareness based on resource domain assignments programmed into the RDC. Refer to the CCM chapter regarding the relationship between cluster resource domains and their respective CCM resources. Failing to follow the proper sequence when updating the resource domain assignments of the cluster can result in clocks being inadvertently gated.

3.2.2.3 Safe Sharing

For shared peripherals, the RDC can be configured to require a domain to obtain a semaphore lock before access to the peripheral is allowed. This feature helps prevent collisions from processes on separate clusters that may want to use the same peripheral at the same time. The RDC sends a list of eligible domains to the semaphore module for each gate/peripheral. The eligible domains are those that are set in the peripheral domain access permissions (PDAP) registers. There is a one-to-one correspondence between semaphore gates and peripherals so each gate in the semaphore block represents a peripheral. The RDC receives semaphore locks from the hardware semaphore module (SEMA42). A semaphore lock is acquired when a cluster or bus master from a given domain requests a lock for a particular gate. The semaphore module compares the request's domain ID against the list of eligible domain IDs. If the domain ID is on the list and the lock is available then the lock is set and a signal is sent back to the RDC module indicating a lock has been acquired for a particular gate and a domain ID the lock belongs to. The RDC then restricts access to the corresponding peripheral to only transactions originating from the domain that has the lock. Another domain, though on the shared list to access the peripheral, must then wait until the lock is released before acquiring the lock and gaining access to the peripheral. To enable this feature of hardware enforcement for the semaphore locks, the SREQ bit is set in the RDC resource register.

If the SREQ is set, then when a process determines it needs a shared peripheral, it must first lock the resource in the semaphore module. Once the resource is locked, the semaphore module sends a signal to the RDC indicating the domain has access to the resource. The RDC will then set the access permissions to allow that domain access to the peripheral.

For a domain to acquire a lock on a peripheral, the domain must have been assigned to the peripheral in the RDC Peripheral Domain Access Permissions register (PDAP). The semaphore module only allows safe-sharing locks for those domains that are assigned to the peripheral. The semaphore module does not consider the access type (Read or Write) when allowing domains to acquire locks.

The SEMA42 module implements hardware-enforced semaphores as an IPS-mapped slave peripheral device. The feature set includes:

- Module definition supporting 64 hardware-enforced gates in a multi-processor configuration, where up to 15 processors can be supported; cp*X* is meant to represent cluster processor *X*
  - Gates appear as an *n*-entry byte-size array with read and write accesses (*n* = 64).
    - Processors lock gates by writing "Master_index" to the appropriate gate and must read back the gate value to verify the lock operation was successful. The Master_index value for the processors can be found in the Master Index Allocation table, which can be found in the AIPSTZ block. Also note that after locking, the gate register contains the master_id value of the locking processor (in the field GTFSM), and also the value of the locking domain (in the field LDOM).
    - Once locked, the gate is unlocked by a write of zeroes from the locking processor.
    - The number of implemented gates is specified by a hardware configuration define.
  - Each hardware gate appears as a 16-state, 4-bit state machine.
    - 16-state implementation

      if gate = 0x0, then state = unlocked
      if gate = 0x1, then state = locked by processor (master_index) 0
      if gate = 0x2, then state = locked by processor (master_index) 1
      …
      if gate = 0xF, then state = locked by processor (master_index) 14

- Uses the logical bus master number (master_index) as a reference attribute plus the specified data patterns to validate all write operations.
- Once locked, the gate can (and must) be unlocked by a write of zeroes from the locking processor.

- Secure reset mechanisms are supported to clear the contents of individual gates, as well as a clear_all capability.

- Memory-mapped IPS slave peripheral platform module
  - Interface to the IPS bus for programming-model accesses

3.2.2.4 Resource Domain Control and Security Considerations

Conceptually, the RDC configuration is independent of the processor privilege mode and security domain. It is intended to allow for isolation between cluster processing environments to prevent collisions and increase reliability. Access between resource domains is mutually exclusive and each domain should be in control of its own privilege modes and access rights.

However, it is important to realize multi-cluster processors may have a multiple resource domains but only one overarching security domain. Chip security controls reside in one resource domain. In this configuration, a domain can affect at least one level of access privileges in the other domain. This may be acceptable but clarity and care is needed to ensure expected functionality.

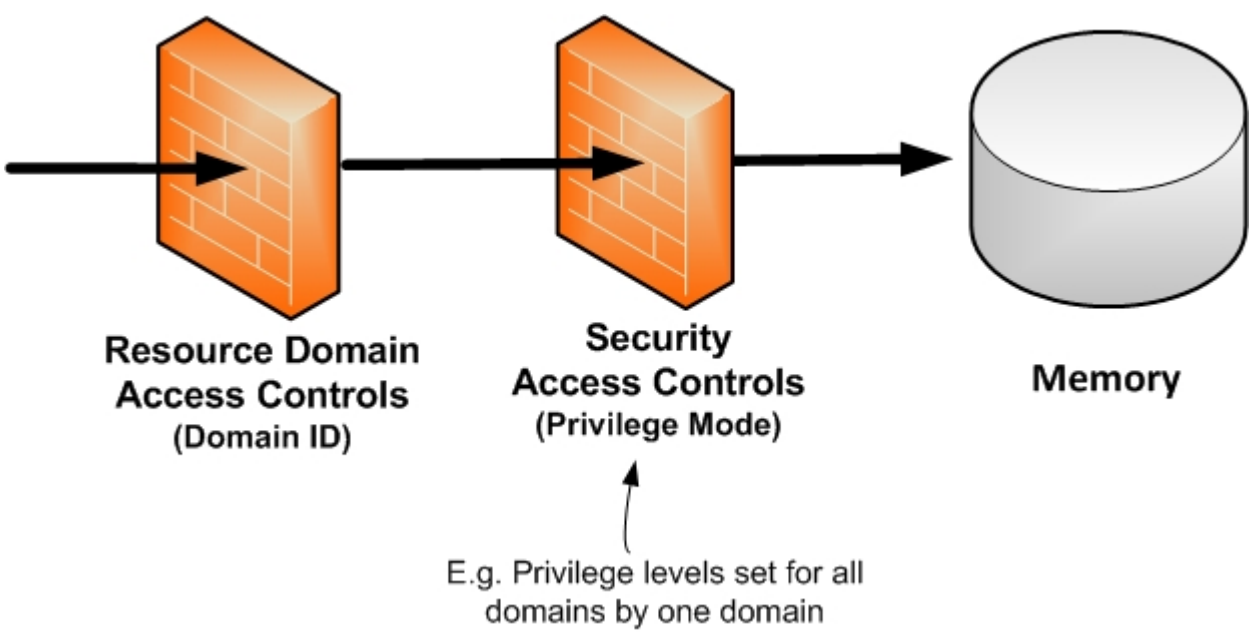


Figure 3-3. Access Control to Memory

Therefore, access to the security controls should be restricted to the most trustworthy operating mode of the cluster and privilege levels should be coordinated to ensure that shared peripherals and memory regions are accessible by both clusters. For instance, if a memory region is designated for secure accesses then all domain masters that share that region must have secure privileges.

3.2.2.5 Modes of Operation

The RDC provides access controls to the resource domain subsystem. When the device is in a low power mode then some memory regions in the subsystem may be powered off. RDC responds to the impacted memory regions by automatically reconfiguring the memory regions once power returns and blocking access to those memory regions until the reconfiguration process is complete.

3.2.2.5.1 Low Power Modes

The RDC loads configuration information for memory regions (MRSA, MRSE, MRC) into access control mechanisms (gaskets) at the memory interface. The location of this configuration information may reside inside power domains that lose power during sleep modes for energy savings. To restore configuration information upon return from sleep mode, the RDC receives a global power control signal indicating power is restored. The RDC then automatically reconfigures the memory regions with the configuration information.

During reconfiguration, access is blocked to the previously powered down memories. When the RDC completes reconfiguration it issues an interrupt and allows access to the memory regions. Only the powered down regions are blocked during the reconfiguration. Memory regions in the "always-on" power domain (still powered during sleep mode) remain available according to the programmed access rights. If no memory regions were enabled then the powered down regions are available immediately when power is restored.

The figure below shows the Global Power Control signal which RDC uses to invalidate the configuration upon deassertion and to restore the configuration when re-asserted. The configuration is valid and bus transactions allowed once the memory regions have been restored.

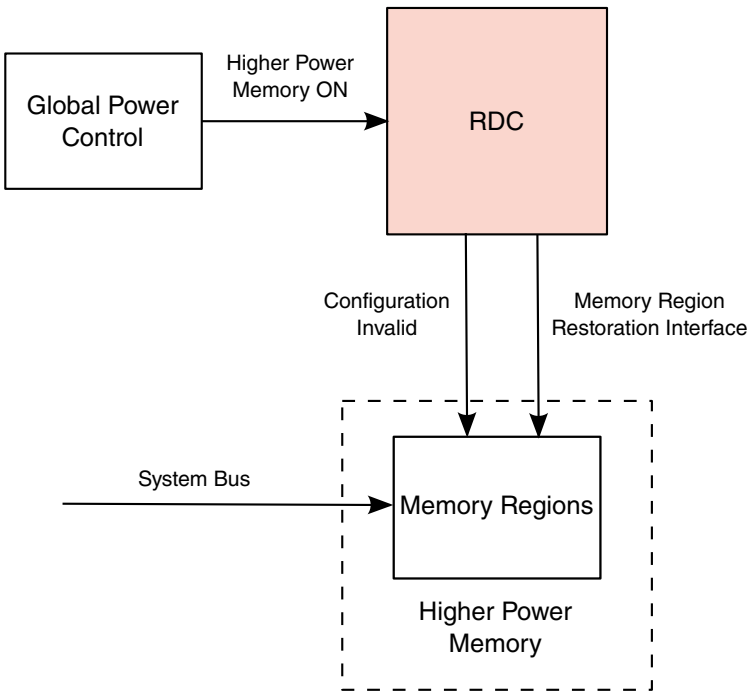


Figure 3-4. Memory Restoration Signaling

3.2.3 External Signals

RDC has no external signals pinned out.

3.2.4 Programming Interface

This section provides product specific details describing the mapping of resources - peripherals, bus masters, and memory regions - to corresponding resource domain controls RDC registers.

The RDC and RDC_SEMA42 register maps are combined in this chapter. The base address for the one RDC map and two SEMA42 maps are each separated by 4KB. While there are two SEMA42 submodules and therefore two sets of SEMA42 registers, this chapter describes one. Please refer to the peripheral memory map for the base addresses of the RDC and SEMA42 modules.

3.2.4.1 Master Assignment Registers

Table 3-2. Master Assignment Mapping

Master	RDC MDA Register
Arm Cortex A53 Platform	RDC_MDA0
Arm Cortex M7 Platform	RDC_MDA1
PCIE_CTRL1	RDC_MDA2
SDMA3(p)	RDC_MDA3
SDMA3(b)	RDC_MDA4
LCDIF1	RDC_MDA5
ISI	RDC_MDA6
NPU	RDC_MDA7
Coresight	RDC_MDA8
DAP	RDC_MDA9
CAAM	RDC_MDA10
SDMA1(p)	RDC_MDA11
SDMA1(b)	RDC_MDA12
APBHDMA	RDC_MDA13
RAWNAND	RDC_MDA14
uSDHC1	RDC_MDA15
uSDHC2	RDC_MDA16
uSDHC3	RDC_MDA17
AUDIO PROCESSOR	RDC_MDA18
USB1	RDC_MDA19
USB2	RDC_MDA20
Reserved	RDC_MDA21
ENET1_TX	RDC_MDA22
ENET1_RX	RDC_MDA23
SDMA2(p), SDMA2(b), SDMA2 to SPBA2	RDC_MDA24
SDMA3 to SPBA2	RDC_MDA25
SDMA1 to SPBA1	RDC_MDA26
LCDIF2	RDC_MDA27
HDMI_TX	RDC_MDA28
ENET2 (QoS TSN)	RDC_MDA29
GPU3D	RDC_MDA30
GPU2D	RDC_MDA31
VPU G1	RDC_MDA32
VPU G2	RDC_MDA33
VPU VC8000E	RDC_MDA34
AUDIO_EDMA	RDC_MDA35
ISP1	RDC_MDA36
ISP2	RDC_MDA37
DEWARP	RDC_MDA38
GIC500	RDC_MDA39

3.2.4.2 Peripheral Mapping

Each peripheral has a corresponding resource domain assignment register in the RDC and semaphore lock register in the RDC_SEMA42 module. The following table shows allocation of the RDC_PDAP and RDC_SEMAPHORE*x*_GATE registers for peripheral resource domain assignment.

NOTE

Access control of the RDC registers can be programmed using the respective PDAP register. The default setting of the PDAP register for the RDC allows access from all domains. Use caution when restricting access of the RDC registers to avoid conditions where access to the RDC registers is needed but no master is assigned to a domain with access rights to the RDC.

Table 3-3. RDC Peripheral Mapping

Peripheral	RDC PDAP register	RDC_SEMA42 block/gate register
GPIO1	RDC_PDAP00	SEMA42 B1 / G0
GPIO2	RDC_PDAP01	SEMA42 B1 / G1
GPIO3	RDC_PDAP02	SEMA42 B1 / G2
GPIO4	RDC_PDAP03	SEMA42 B1 / G3
GPIO5	RDC_PDAP04	SEMA42 B1 / G4
MU_2_A (A53, Audio Processor)	RDC_PDAP05	SEMA42 B1 / G5
ANA_TSENSOR	RDC_PDAP06	SEMA42 B1 / G6
ANA_OSC	RDC_PDAP07	SEMA42 B1 / G7
WDOG1	RDC_PDAP08	SEMA42 B1 / G8
WDOG2	RDC_PDAP09	SEMA42 B1 / G9
WDOG3	RDC_PDAP10	SEMA42 B1 / G10
OCRAM	RDC_PDAP11	SEMA42 B1 / G11
OCRAM_S	RDC_PDAP12	SEMA42 B1 / G12
GPT1	RDC_PDAP13	SEMA42 B1 / G13
GPT2	RDC_PDAP14	SEMA42 B1 / G14
GPT3	RDC_PDAP15	SEMA42 B1 / G15
MU_2_B (A53, Audio Processor)	RDC_PDAP16	SEMA42 B1 / G16
Reserved	RDC_PDAP17	SEMA42 B1 / G17
MU_3_A (M7, Audio Processor)	RDC_PDAP18	SEMA42 B1 / G18
IOMUXC	RDC_PDAP19	SEMA42 B1 / G19
IOMUXC_GPR	RDC_PDAP20	SEMA42 B1 / G20

Table continues on the next page...

Table 3-3. RDC Peripheral Mapping (continued)

Peripheral	RDC PDAP register	RDC_SEMA42 block/gate register
OCOTP_CTRL	RDC_PDAP21	SEMA42 B1 / G21
ANA_PLL	RDC_PDAP22	SEMA42 B1 / G22
SNVS_HP	RDC_PDAP23	SEMA42 B1 / G23
CCM	RDC_PDAP24	SEMA42 B1 / G24
SRC	RDC_PDAP25	SEMA42 B1 / G25
GPC	RDC_PDAP26	SEMA42 B1 / G26
SEMAPHORE1	RDC_PDAP27	SEMA42 B1 / G27
SEMAPHORE2	RDC_PDAP28	SEMA42 B1 / G28
RDC	RDC_PDAP29	SEMA42 B1 / G29
CSU	RDC_PDAP30	SEMA42 B1 / G30
MU_3_B (M7, Audio Processor)	RDC_PDAP31	SEMA42 B1 / G31
ISI	RDC_PDAP32	SEMA42 B1 / G32
ISP1	RDC_PDAP33	SEMA42 B1 / G33
ISP2	RDC_PDAP34	SEMA42 B1 / G34
IPS Dewarp	RDC_PDAP35	SEMA42 B1 / G35
MIPI_CSI1	RDC_PDAP36	SEMA42 B1 / G36
HSIOMIX BLK_CTL	RDC_PDAP37	SEMA42 B1 / G37
PWM1	RDC_PDAP38	SEMA42 B1 / G38
PWM2	RDC_PDAP39	SEMA42 B1 / G39
PWM3	RDC_PDAP40	SEMA42 B1 / G40
PWM4	RDC_PDAP41	SEMA42 B1 / G41
System_Counter_RD	RDC_PDAP42	SEMA42 B1 / G42
System_Counter_CMP	RDC_PDAP43	SEMA42 B1 / G43
System_Counter_CTRL	RDC_PDAP44	SEMA42 B1 / G44
I2C5	RDC_PDAP45	SEMA42 B1 / G45
GPT6	RDC_PDAP46	SEMA42 B1 / G46
GPT5	RDC_PDAP47	SEMA42 B1 / G47
GPT4	RDC_PDAP48	SEMA42 B1 / G48
MIPI_CSI2	RDC_PDAP49	SEMA42 B1 / G49
MIPI_DSI1	RDC_PDAP50	SEMA42 B1 / G50
MEDIAMIX BLK_CTL	RDC_PDAP51	SEMA42 B1 / G51
LCDIF1	RDC_PDAP52	SEMA42 B1 / G52
eDMA Management Page	RDC_PDAP53	SEMA42 B1 / G53
eDMA Channels[15:0] at 4KB/ch	RDC_PDAP54	SEMA42 B1 / G54
eDMA Channels[31:16] at 4KB/ch	RDC_PDAP55	SEMA42 B1 / G55
TZASC	RDC_PDAP56	SEMA42 B1 / G56
I2C6	RDC_PDAP57	SEMA42 B1 / G57
CAAM	RDC_PDAP58	SEMA42 B1 / G58
LCDIF2	RDC_PDAP59	SEMA42 B1 / G59

Table continues on the next page...

Table 3-3. RDC Peripheral Mapping (continued)

Peripheral	RDC PDAP register	RDC_SEMA42 block/gate register
PERFMON1	RDC_PDAP60	SEMA42 B1 / G60
PERFMON2	RDC_PDAP61	SEMA42 B1 / G61
NOC BLK_CTL	RDC_PDAP62	SEMA42 B1 / G62
QoSC	RDC_PDAP63	SEMA42 B1 / G63
LVDS1	RDC_PDAP64	SEMA42 B2 / G0
LVDS2	RDC_PDAP65	SEMA42 B2 / G1
I2C1	RDC_PDAP66	SEMA42 B2 / G2
I2C2	RDC_PDAP67	SEMA42 B2 / G3
I2C3	RDC_PDAP68	SEMA42 B2 / G4
I2C4	RDC_PDAP69	SEMA42 B2 / G5
UART4	RDC_PDAP70	SEMA42 B2 / G6
HDMI_TX	RDC_PDAP71	SEMA42 B2 / G7
IRQ_STEER (Audio Processor)	RDC_PDAP72	SEMA42 B2 / G8
SDMA2	RDC_PDAP73	SEMA42 B2 / G9
MU_1_A (A53,M7)	RDC_PDAP74	SEMA42 B2 / G10
MU_1_B (A53, M7)	RDC_PDAP75	SEMA42 B2 / G11
SEMAPHORE_HS	RDC_PDAP76	SEMA42 B2 / G12
Reserved	RDC_PDAP77	SEMA42 B2 / G13
SAI1	RDC_PDAP78	SEMA42 B2 / G14
SAI2	RDC_PDAP79	SEMA42 B2 / G15
SAI3	RDC_PDAP80	SEMA42 B2 / G16
CAN_FD1	RDC_PDAP81	SEMA42 B2 / G17
SAI5	RDC_PDAP82	SEMA42 B2 / G18
SAI6	RDC_PDAP83	SEMA42 B2 / G19
uSDHC1	RDC_PDAP84	SEMA42 B2 / G20
uSDHC2	RDC_PDAP85	SEMA42 B2 / G21
uSDHC3	RDC_PDAP86	SEMA42 B2 / G22
PCIE_PHY1	RDC_PDAP87	SEMA42 B2 / G23
HDMI TX AUDLNK MSTR	RDC_PDAP88	SEMA42 B2 / G24
CAN_FD2	RDC_PDAP89	SEMA42 B2 / G25
SPBA2	RDC_PDAP90	SEMA42 B2 / G26
QSPI	RDC_PDAP91	SEMA42 B2 / G27
AUDIO BLK_CTRL	RDC_PDAP92	SEMA42 B2 / G28
SDMA1	RDC_PDAP93	SEMA42 B2 / G29
ENET1	RDC_PDAP94	SEMA42 B2 / G30
ENET2_TSN	RDC_PDAP95	SEMA42 B2 / G31
Reserved	RDC_PDAP96	SEMA42 B2 / G32
ASRC	RDC_PDAP97	SEMA42 B2 / G33
eCSPI1	RDC_PDAP98	SEMA42 B2 / G34

Table continues on the next page...

Table 3-3. RDC Peripheral Mapping (continued)

Peripheral	RDC PDAP register	RDC_SEMA42 block/gate register
eCSPI2	RDC_PDAP99	SEMA42 B2 / G35
eCSPI3	RDC_PDAP100	SEMA42 B2 / G36
SAI7	RDC_PDAP101	SEMA42 B2 / G37
UART1	RDC_PDAP102	SEMA42 B2 / G38
Reserved	RDC_PDAP103	SEMA42 B2 / G39
UART3	RDC_PDAP104	SEMA42 B2 / G40
UART2	RDC_PDAP105	SEMA42 B2 / G41
PDM (MICFIL)	RDC_PDAP106	SEMA42 B2 / G42
AUDIO XCVR RX (eARC)	RDC_PDAP107	SEMA42 B2 / G43
Reserved	RDC_PDAP108	SEMA42 B2 / G44
SDMA3	RDC_PDAP109	SEMA42 B2 / G45
Reserved	RDC_PDAP110	SEMA42 B2 / G46
SPBA1	RDC_PDAP111	SEMA42 B2 / G47
Reserved	RDC_PDAP112	SEMA42 B2 / G48

3.2.4.3 Memory Region Map

The number of memories with domain isolation support varies per device. The number of memory regions for a particular memory and the size of those regions varies per memory gasket. Each region of memory has a set of registers to define the boundaries of the region based on start and end addresses, a control register to set the domain access permissions and enable the region, and a status register to determin if access was denied to a region.

For this device, refer to the table below to determine the memories with domain support, the number of regions for each memory, the region resolution, the identifying numbers for the sets of memory region registers, and the addresses of the RDC registers to access the sets of Memory Region registers.

Table 3-4. Memory Region Mapping

Memory/Port	# of Regions	Region Resolution	Memory Region Register Set Number (e.g. MRSA, MREA, MRC, MRVS)	Register Address Range	Needs right-shift by 1-bit[1]
DEBUG (DAP)	4	4 KB	0 - 3	0x800 - 0x83C	Yes
FLEXSPI	8	4 KB	4 - 11	0x840 - 0x8BC	No
OCRAM	5	128 B	12 - 16	0x8C0 - 0x90C	Yes
OCRAM_S	5	128 B	17 - 21	0x910 - 0x95C	Yes

Table continues on the next page...

Table 3-4. Memory Region Mapping (continued)

Memory/Port	# of Regions	Region Resolution	Memory Region Register Set Number (e.g. MRSA, MREA, MRC, MRVS)	Register Address Range	Needs right-shift by 1-bit[1]
M7 TCM	5	128 B	22 - 26	0x960 - 0x9AC	No
GIC	4	4 KB	27 - 30	0x9B0 - 0x9EC	Yes
GPU	8	4 KB	31 - 38	0x9F0 - 0xA6C	Yes
DRAM	8	4 KB	39 - 46	0xA70 - 0xAEC	No
DDRC (REG)	5	4 KB	47 - 51	0xAF0 - 0xB3C	Yes
PCIe1, USB1, and USB2	8	4 KB	52 - 59	0xB40 - 0xBBC	No
VPU	4	4 KB	60 - 63	0xBC0 - 0xBFC	Yes
NPU	4	4 KB	64 - 67	0xC00 - 0xC3C	Yes
AUDIO PROCESSOR	4	4 KB	68 - 71	0xC40 - 0xC7C	Yes
OCRAM_A	5	128 B	72 - 76	0xC80 - 0xCCC	Yes

1. Whether or not the actual start/end address needs right-shift by 1-bit, then filled into MRSA and MREA.

3.2.5 RDC Memory Map/Register Definition

RDC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0000	Version Information (RDC_VIR)	32	R	04D7_2284h	3.2.5.1/72
303D_0024	Status (RDC_STAT)	32	R/W	0000_0100h	3.2.5.2/73
303D_0028	Interrupt and Control (RDC_INTCTRL)	32	R/W	0000_0000h	3.2.5.3/74
303D_002C	Interrupt Status (RDC_INTSTAT)	32	R/W	See section	3.2.5.4/74
303D_0200	Master Domain Assignment (RDC_MDA0)	32	R/W	0000_0000h	3.2.5.5/75
303D_0204	Master Domain Assignment (RDC_MDA1)	32	R/W	0000_0000h	3.2.5.5/75
303D_0208	Master Domain Assignment (RDC_MDA2)	32	R/W	0000_0000h	3.2.5.5/75
303D_020C	Master Domain Assignment (RDC_MDA3)	32	R/W	0000_0000h	3.2.5.5/75
303D_0210	Master Domain Assignment (RDC_MDA4)	32	R/W	0000_0000h	3.2.5.5/75
303D_0214	Master Domain Assignment (RDC_MDA5)	32	R/W	0000_0000h	3.2.5.5/75
303D_0218	Master Domain Assignment (RDC_MDA6)	32	R/W	0000_0000h	3.2.5.5/75
303D_021C	Master Domain Assignment (RDC_MDA7)	32	R/W	0000_0000h	3.2.5.5/75
303D_0220	Master Domain Assignment (RDC_MDA8)	32	R/W	0000_0000h	3.2.5.5/75
303D_0224	Master Domain Assignment (RDC_MDA9)	32	R/W	0000_0000h	3.2.5.5/75
303D_0228	Master Domain Assignment (RDC_MDA10)	32	R/W	0000_0000h	3.2.5.5/75

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_022C	Master Domain Assignment (RDC_MDA11)	32	R/W	0000_0000h	3.2.5.5/75
303D_0230	Master Domain Assignment (RDC_MDA12)	32	R/W	0000_0000h	3.2.5.5/75
303D_0234	Master Domain Assignment (RDC_MDA13)	32	R/W	0000_0000h	3.2.5.5/75
303D_0238	Master Domain Assignment (RDC_MDA14)	32	R/W	0000_0000h	3.2.5.5/75
303D_023C	Master Domain Assignment (RDC_MDA15)	32	R/W	0000_0000h	3.2.5.5/75
303D_0240	Master Domain Assignment (RDC_MDA16)	32	R/W	0000_0000h	3.2.5.5/75
303D_0244	Master Domain Assignment (RDC_MDA17)	32	R/W	0000_0000h	3.2.5.5/75
303D_0248	Master Domain Assignment (RDC_MDA18)	32	R/W	0000_0000h	3.2.5.5/75
303D_024C	Master Domain Assignment (RDC_MDA19)	32	R/W	0000_0000h	3.2.5.5/75
303D_0250	Master Domain Assignment (RDC_MDA20)	32	R/W	0000_0000h	3.2.5.5/75
303D_0254	Master Domain Assignment (RDC_MDA21)	32	R/W	0000_0000h	3.2.5.5/75
303D_0258	Master Domain Assignment (RDC_MDA22)	32	R/W	0000_0000h	3.2.5.5/75
303D_025C	Master Domain Assignment (RDC_MDA23)	32	R/W	0000_0000h	3.2.5.5/75
303D_0260	Master Domain Assignment (RDC_MDA24)	32	R/W	0000_0000h	3.2.5.5/75
303D_0264	Master Domain Assignment (RDC_MDA25)	32	R/W	0000_0000h	3.2.5.5/75
303D_0268	Master Domain Assignment (RDC_MDA26)	32	R/W	0000_0000h	3.2.5.5/75
303D_026C	Master Domain Assignment (RDC_MDA27)	32	R/W	0000_0000h	3.2.5.5/75
303D_0270	Master Domain Assignment (RDC_MDA28)	32	R/W	0000_0000h	3.2.5.5/75
303D_0274	Master Domain Assignment (RDC_MDA29)	32	R/W	0000_0000h	3.2.5.5/75
303D_0278	Master Domain Assignment (RDC_MDA30)	32	R/W	0000_0000h	3.2.5.5/75
303D_027C	Master Domain Assignment (RDC_MDA31)	32	R/W	0000_0000h	3.2.5.5/75
303D_0280	Master Domain Assignment (RDC_MDA32)	32	R/W	0000_0000h	3.2.5.5/75
303D_0284	Master Domain Assignment (RDC_MDA33)	32	R/W	0000_0000h	3.2.5.5/75
303D_0288	Master Domain Assignment (RDC_MDA34)	32	R/W	0000_0000h	3.2.5.5/75
303D_028C	Master Domain Assignment (RDC_MDA35)	32	R/W	0000_0000h	3.2.5.5/75
303D_0290	Master Domain Assignment (RDC_MDA36)	32	R/W	0000_0000h	3.2.5.5/75
303D_0294	Master Domain Assignment (RDC_MDA37)	32	R/W	0000_0000h	3.2.5.5/75
303D_0298	Master Domain Assignment (RDC_MDA38)	32	R/W	0000_0000h	3.2.5.5/75
303D_029C	Master Domain Assignment (RDC_MDA39)	32	R/W	0000_0000h	3.2.5.5/75
303D_0400	Peripheral Domain Access Permissions (RDC_PDAP0)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0404	Peripheral Domain Access Permissions (RDC_PDAP1)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0408	Peripheral Domain Access Permissions (RDC_PDAP2)	32	R/W	0000_00FFh	3.2.5.6/76
303D_040C	Peripheral Domain Access Permissions (RDC_PDAP3)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0410	Peripheral Domain Access Permissions (RDC_PDAP4)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0414	Peripheral Domain Access Permissions (RDC_PDAP5)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0418	Peripheral Domain Access Permissions (RDC_PDAP6)	32	R/W	0000_00FFh	3.2.5.6/76
303D_041C	Peripheral Domain Access Permissions (RDC_PDAP7)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0420	Peripheral Domain Access Permissions (RDC_PDAP8)	32	R/W	0000_00FFh	3.2.5.6/76

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0424	Peripheral Domain Access Permissions (RDC_PDAP9)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0428	Peripheral Domain Access Permissions (RDC_PDAP10)	32	R/W	0000_00FFh	3.2.5.6/76
303D_042C	Peripheral Domain Access Permissions (RDC_PDAP11)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0430	Peripheral Domain Access Permissions (RDC_PDAP12)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0434	Peripheral Domain Access Permissions (RDC_PDAP13)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0438	Peripheral Domain Access Permissions (RDC_PDAP14)	32	R/W	0000_00FFh	3.2.5.6/76
303D_043C	Peripheral Domain Access Permissions (RDC_PDAP15)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0440	Peripheral Domain Access Permissions (RDC_PDAP16)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0444	Peripheral Domain Access Permissions (RDC_PDAP17)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0448	Peripheral Domain Access Permissions (RDC_PDAP18)	32	R/W	0000_00FFh	3.2.5.6/76
303D_044C	Peripheral Domain Access Permissions (RDC_PDAP19)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0450	Peripheral Domain Access Permissions (RDC_PDAP20)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0454	Peripheral Domain Access Permissions (RDC_PDAP21)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0458	Peripheral Domain Access Permissions (RDC_PDAP22)	32	R/W	0000_00FFh	3.2.5.6/76
303D_045C	Peripheral Domain Access Permissions (RDC_PDAP23)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0460	Peripheral Domain Access Permissions (RDC_PDAP24)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0464	Peripheral Domain Access Permissions (RDC_PDAP25)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0468	Peripheral Domain Access Permissions (RDC_PDAP26)	32	R/W	0000_00FFh	3.2.5.6/76
303D_046C	Peripheral Domain Access Permissions (RDC_PDAP27)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0470	Peripheral Domain Access Permissions (RDC_PDAP28)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0474	Peripheral Domain Access Permissions (RDC_PDAP29)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0478	Peripheral Domain Access Permissions (RDC_PDAP30)	32	R/W	0000_00FFh	3.2.5.6/76
303D_047C	Peripheral Domain Access Permissions (RDC_PDAP31)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0480	Peripheral Domain Access Permissions (RDC_PDAP32)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0484	Peripheral Domain Access Permissions (RDC_PDAP33)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0488	Peripheral Domain Access Permissions (RDC_PDAP34)	32	R/W	0000_00FFh	3.2.5.6/76
303D_048C	Peripheral Domain Access Permissions (RDC_PDAP35)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0490	Peripheral Domain Access Permissions (RDC_PDAP36)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0494	Peripheral Domain Access Permissions (RDC_PDAP37)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0498	Peripheral Domain Access Permissions (RDC_PDAP38)	32	R/W	0000_00FFh	3.2.5.6/76
303D_049C	Peripheral Domain Access Permissions (RDC_PDAP39)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04A0	Peripheral Domain Access Permissions (RDC_PDAP40)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04A4	Peripheral Domain Access Permissions (RDC_PDAP41)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04A8	Peripheral Domain Access Permissions (RDC_PDAP42)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04AC	Peripheral Domain Access Permissions (RDC_PDAP43)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04B0	Peripheral Domain Access Permissions (RDC_PDAP44)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04B4	Peripheral Domain Access Permissions (RDC_PDAP45)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04B8	Peripheral Domain Access Permissions (RDC_PDAP46)	32	R/W	0000_00FFh	3.2.5.6/76

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_04BC	Peripheral Domain Access Permissions (RDC_PDAP47)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04C0	Peripheral Domain Access Permissions (RDC_PDAP48)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04C4	Peripheral Domain Access Permissions (RDC_PDAP49)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04C8	Peripheral Domain Access Permissions (RDC_PDAP50)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04CC	Peripheral Domain Access Permissions (RDC_PDAP51)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04D0	Peripheral Domain Access Permissions (RDC_PDAP52)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04D4	Peripheral Domain Access Permissions (RDC_PDAP53)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04D8	Peripheral Domain Access Permissions (RDC_PDAP54)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04DC	Peripheral Domain Access Permissions (RDC_PDAP55)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04E0	Peripheral Domain Access Permissions (RDC_PDAP56)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04E4	Peripheral Domain Access Permissions (RDC_PDAP57)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04E8	Peripheral Domain Access Permissions (RDC_PDAP58)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04EC	Peripheral Domain Access Permissions (RDC_PDAP59)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04F0	Peripheral Domain Access Permissions (RDC_PDAP60)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04F4	Peripheral Domain Access Permissions (RDC_PDAP61)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04F8	Peripheral Domain Access Permissions (RDC_PDAP62)	32	R/W	0000_00FFh	3.2.5.6/76
303D_04FC	Peripheral Domain Access Permissions (RDC_PDAP63)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0500	Peripheral Domain Access Permissions (RDC_PDAP64)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0504	Peripheral Domain Access Permissions (RDC_PDAP65)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0508	Peripheral Domain Access Permissions (RDC_PDAP66)	32	R/W	0000_00FFh	3.2.5.6/76
303D_050C	Peripheral Domain Access Permissions (RDC_PDAP67)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0510	Peripheral Domain Access Permissions (RDC_PDAP68)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0514	Peripheral Domain Access Permissions (RDC_PDAP69)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0518	Peripheral Domain Access Permissions (RDC_PDAP70)	32	R/W	0000_00FFh	3.2.5.6/76
303D_051C	Peripheral Domain Access Permissions (RDC_PDAP71)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0520	Peripheral Domain Access Permissions (RDC_PDAP72)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0524	Peripheral Domain Access Permissions (RDC_PDAP73)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0528	Peripheral Domain Access Permissions (RDC_PDAP74)	32	R/W	0000_00FFh	3.2.5.6/76
303D_052C	Peripheral Domain Access Permissions (RDC_PDAP75)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0530	Peripheral Domain Access Permissions (RDC_PDAP76)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0534	Peripheral Domain Access Permissions (RDC_PDAP77)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0538	Peripheral Domain Access Permissions (RDC_PDAP78)	32	R/W	0000_00FFh	3.2.5.6/76
303D_053C	Peripheral Domain Access Permissions (RDC_PDAP79)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0540	Peripheral Domain Access Permissions (RDC_PDAP80)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0544	Peripheral Domain Access Permissions (RDC_PDAP81)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0548	Peripheral Domain Access Permissions (RDC_PDAP82)	32	R/W	0000_00FFh	3.2.5.6/76
303D_054C	Peripheral Domain Access Permissions (RDC_PDAP83)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0550	Peripheral Domain Access Permissions (RDC_PDAP84)	32	R/W	0000_00FFh	3.2.5.6/76

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0554	Peripheral Domain Access Permissions (RDC_PDAP85)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0558	Peripheral Domain Access Permissions (RDC_PDAP86)	32	R/W	0000_00FFh	3.2.5.6/76
303D_055C	Peripheral Domain Access Permissions (RDC_PDAP87)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0560	Peripheral Domain Access Permissions (RDC_PDAP88)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0564	Peripheral Domain Access Permissions (RDC_PDAP89)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0568	Peripheral Domain Access Permissions (RDC_PDAP90)	32	R/W	0000_00FFh	3.2.5.6/76
303D_056C	Peripheral Domain Access Permissions (RDC_PDAP91)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0570	Peripheral Domain Access Permissions (RDC_PDAP92)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0574	Peripheral Domain Access Permissions (RDC_PDAP93)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0578	Peripheral Domain Access Permissions (RDC_PDAP94)	32	R/W	0000_00FFh	3.2.5.6/76
303D_057C	Peripheral Domain Access Permissions (RDC_PDAP95)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0580	Peripheral Domain Access Permissions (RDC_PDAP96)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0584	Peripheral Domain Access Permissions (RDC_PDAP97)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0588	Peripheral Domain Access Permissions (RDC_PDAP98)	32	R/W	0000_00FFh	3.2.5.6/76
303D_058C	Peripheral Domain Access Permissions (RDC_PDAP99)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0590	Peripheral Domain Access Permissions (RDC_PDAP100)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0594	Peripheral Domain Access Permissions (RDC_PDAP101)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0598	Peripheral Domain Access Permissions (RDC_PDAP102)	32	R/W	0000_00FFh	3.2.5.6/76
303D_059C	Peripheral Domain Access Permissions (RDC_PDAP103)	32	R/W	0000_00FFh	3.2.5.6/76
303D_05A0	Peripheral Domain Access Permissions (RDC_PDAP104)	32	R/W	0000_00FFh	3.2.5.6/76
303D_05A4	Peripheral Domain Access Permissions (RDC_PDAP105)	32	R/W	0000_00FFh	3.2.5.6/76
303D_05A8	Peripheral Domain Access Permissions (RDC_PDAP106)	32	R/W	0000_00FFh	3.2.5.6/76
303D_05AC	Peripheral Domain Access Permissions (RDC_PDAP107)	32	R/W	0000_00FFh	3.2.5.6/76
303D_05B0	Peripheral Domain Access Permissions (RDC_PDAP108)	32	R/W	0000_00FFh	3.2.5.6/76
303D_05B4	Peripheral Domain Access Permissions (RDC_PDAP109)	32	R/W	0000_00FFh	3.2.5.6/76
303D_05B8	Peripheral Domain Access Permissions (RDC_PDAP110)	32	R/W	0000_00FFh	3.2.5.6/76
303D_05BC	Peripheral Domain Access Permissions (RDC_PDAP111)	32	R/W	0000_00FFh	3.2.5.6/76
303D_0800	Memory Region Start Address (RDC_MRSA0)	32	R/W	Undefined	3.2.5.7/77
303D_0804	Memory Region End Address (RDC_MREA0)	32	R/W	Undefined	3.2.5.8/79
303D_0808	Memory Region Control (RDC_MRC0)	32	R/W	0000_00FFh	3.2.5.9/80
303D_080C	Memory Region Violation Status (RDC_MRVS0)	32	R/W	0000_0000h	3.2.5.10/81
303D_0810	Memory Region Start Address (RDC_MRSA1)	32	R/W	Undefined	3.2.5.7/77
303D_0814	Memory Region End Address (RDC_MREA1)	32	R/W	Undefined	3.2.5.8/79
303D_0818	Memory Region Control (RDC_MRC1)	32	R/W	0000_00FFh	3.2.5.9/80
303D_081C	Memory Region Violation Status (RDC_MRVS1)	32	R/W	0000_0000h	3.2.5.10/81
303D_0820	Memory Region Start Address (RDC_MRSA2)	32	R/W	Undefined	3.2.5.7/77
303D_0824	Memory Region End Address (RDC_MREA2)	32	R/W	Undefined	3.2.5.8/79
303D_0828	Memory Region Control (RDC_MRC2)	32	R/W	0000_00FFh	3.2.5.9/80

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_082C	Memory Region Violation Status (RDC_MRVS2)	32	R/W	0000_0000h	3.2.5.10/81
303D_0830	Memory Region Start Address (RDC_MRSA3)	32	R/W	Undefined	3.2.5.7/77
303D_0834	Memory Region End Address (RDC_MREA3)	32	R/W	Undefined	3.2.5.8/79
303D_0838	Memory Region Control (RDC_MRC3)	32	R/W	0000_00FFh	3.2.5.9/80
303D_083C	Memory Region Violation Status (RDC_MRVS3)	32	R/W	0000_0000h	3.2.5.10/81
303D_0840	Memory Region Start Address (RDC_MRSA4)	32	R/W	Undefined	3.2.5.7/77
303D_0844	Memory Region End Address (RDC_MREA4)	32	R/W	Undefined	3.2.5.8/79
303D_0848	Memory Region Control (RDC_MRC4)	32	R/W	0000_00FFh	3.2.5.9/80
303D_084C	Memory Region Violation Status (RDC_MRVS4)	32	R/W	0000_0000h	3.2.5.10/81
303D_0850	Memory Region Start Address (RDC_MRSA5)	32	R/W	Undefined	3.2.5.7/77
303D_0854	Memory Region End Address (RDC_MREA5)	32	R/W	Undefined	3.2.5.8/79
303D_0858	Memory Region Control (RDC_MRC5)	32	R/W	0000_00FFh	3.2.5.9/80
303D_085C	Memory Region Violation Status (RDC_MRVS5)	32	R/W	0000_0000h	3.2.5.10/81
303D_0860	Memory Region Start Address (RDC_MRSA6)	32	R/W	Undefined	3.2.5.7/77
303D_0864	Memory Region End Address (RDC_MREA6)	32	R/W	Undefined	3.2.5.8/79
303D_0868	Memory Region Control (RDC_MRC6)	32	R/W	0000_00FFh	3.2.5.9/80
303D_086C	Memory Region Violation Status (RDC_MRVS6)	32	R/W	0000_0000h	3.2.5.10/81
303D_0870	Memory Region Start Address (RDC_MRSA7)	32	R/W	Undefined	3.2.5.7/77
303D_0874	Memory Region End Address (RDC_MREA7)	32	R/W	Undefined	3.2.5.8/79
303D_0878	Memory Region Control (RDC_MRC7)	32	R/W	0000_00FFh	3.2.5.9/80
303D_087C	Memory Region Violation Status (RDC_MRVS7)	32	R/W	0000_0000h	3.2.5.10/81
303D_0880	Memory Region Start Address (RDC_MRSA8)	32	R/W	Undefined	3.2.5.7/77
303D_0884	Memory Region End Address (RDC_MREA8)	32	R/W	Undefined	3.2.5.8/79
303D_0888	Memory Region Control (RDC_MRC8)	32	R/W	0000_00FFh	3.2.5.9/80
303D_088C	Memory Region Violation Status (RDC_MRVS8)	32	R/W	0000_0000h	3.2.5.10/81
303D_0890	Memory Region Start Address (RDC_MRSA9)	32	R/W	Undefined	3.2.5.7/77
303D_0894	Memory Region End Address (RDC_MREA9)	32	R/W	Undefined	3.2.5.8/79
303D_0898	Memory Region Control (RDC_MRC9)	32	R/W	0000_00FFh	3.2.5.9/80
303D_089C	Memory Region Violation Status (RDC_MRVS9)	32	R/W	0000_0000h	3.2.5.10/81
303D_08A0	Memory Region Start Address (RDC_MRSA10)	32	R/W	Undefined	3.2.5.7/77
303D_08A4	Memory Region End Address (RDC_MREA10)	32	R/W	Undefined	3.2.5.8/79
303D_08A8	Memory Region Control (RDC_MRC10)	32	R/W	0000_00FFh	3.2.5.9/80
303D_08AC	Memory Region Violation Status (RDC_MRVS10)	32	R/W	0000_0000h	3.2.5.10/81
303D_08B0	Memory Region Start Address (RDC_MRSA11)	32	R/W	Undefined	3.2.5.7/77
303D_08B4	Memory Region End Address (RDC_MREA11)	32	R/W	Undefined	3.2.5.8/79
303D_08B8	Memory Region Control (RDC_MRC11)	32	R/W	0000_00FFh	3.2.5.9/80
303D_08BC	Memory Region Violation Status (RDC_MRVS11)	32	R/W	0000_0000h	3.2.5.10/81
303D_08C0	Memory Region Start Address (RDC_MRSA12)	32	R/W	Undefined	3.2.5.7/77

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_08C4	Memory Region End Address (RDC_MREA12)	32	R/W	Undefined	3.2.5.8/79
303D_08C8	Memory Region Control (RDC_MRC12)	32	R/W	0000_00FFh	3.2.5.9/80
303D_08CC	Memory Region Violation Status (RDC_MRVS12)	32	R/W	0000_0000h	3.2.5.10/81
303D_08D0	Memory Region Start Address (RDC_MRSA13)	32	R/W	Undefined	3.2.5.7/77
303D_08D4	Memory Region End Address (RDC_MREA13)	32	R/W	Undefined	3.2.5.8/79
303D_08D8	Memory Region Control (RDC_MRC13)	32	R/W	0000_00FFh	3.2.5.9/80
303D_08DC	Memory Region Violation Status (RDC_MRVS13)	32	R/W	0000_0000h	3.2.5.10/81
303D_08E0	Memory Region Start Address (RDC_MRSA14)	32	R/W	Undefined	3.2.5.7/77
303D_08E4	Memory Region End Address (RDC_MREA14)	32	R/W	Undefined	3.2.5.8/79
303D_08E8	Memory Region Control (RDC_MRC14)	32	R/W	0000_00FFh	3.2.5.9/80
303D_08EC	Memory Region Violation Status (RDC_MRVS14)	32	R/W	0000_0000h	3.2.5.10/81
303D_08F0	Memory Region Start Address (RDC_MRSA15)	32	R/W	Undefined	3.2.5.7/77
303D_08F4	Memory Region End Address (RDC_MREA15)	32	R/W	Undefined	3.2.5.8/79
303D_08F8	Memory Region Control (RDC_MRC15)	32	R/W	0000_00FFh	3.2.5.9/80
303D_08FC	Memory Region Violation Status (RDC_MRVS15)	32	R/W	0000_0000h	3.2.5.10/81
303D_0900	Memory Region Start Address (RDC_MRSA16)	32	R/W	Undefined	3.2.5.7/77
303D_0904	Memory Region End Address (RDC_MREA16)	32	R/W	Undefined	3.2.5.8/79
303D_0908	Memory Region Control (RDC_MRC16)	32	R/W	0000_00FFh	3.2.5.9/80
303D_090C	Memory Region Violation Status (RDC_MRVS16)	32	R/W	0000_0000h	3.2.5.10/81
303D_0910	Memory Region Start Address (RDC_MRSA17)	32	R/W	Undefined	3.2.5.7/77
303D_0914	Memory Region End Address (RDC_MREA17)	32	R/W	Undefined	3.2.5.8/79
303D_0918	Memory Region Control (RDC_MRC17)	32	R/W	0000_00FFh	3.2.5.9/80
303D_091C	Memory Region Violation Status (RDC_MRVS17)	32	R/W	0000_0000h	3.2.5.10/81
303D_0920	Memory Region Start Address (RDC_MRSA18)	32	R/W	Undefined	3.2.5.7/77
303D_0924	Memory Region End Address (RDC_MREA18)	32	R/W	Undefined	3.2.5.8/79
303D_0928	Memory Region Control (RDC_MRC18)	32	R/W	0000_00FFh	3.2.5.9/80
303D_092C	Memory Region Violation Status (RDC_MRVS18)	32	R/W	0000_0000h	3.2.5.10/81
303D_0930	Memory Region Start Address (RDC_MRSA19)	32	R/W	Undefined	3.2.5.7/77
303D_0934	Memory Region End Address (RDC_MREA19)	32	R/W	Undefined	3.2.5.8/79
303D_0938	Memory Region Control (RDC_MRC19)	32	R/W	0000_00FFh	3.2.5.9/80
303D_093C	Memory Region Violation Status (RDC_MRVS19)	32	R/W	0000_0000h	3.2.5.10/81
303D_0940	Memory Region Start Address (RDC_MRSA20)	32	R/W	Undefined	3.2.5.7/77
303D_0944	Memory Region End Address (RDC_MREA20)	32	R/W	Undefined	3.2.5.8/79
303D_0948	Memory Region Control (RDC_MRC20)	32	R/W	0000_00FFh	3.2.5.9/80
303D_094C	Memory Region Violation Status (RDC_MRVS20)	32	R/W	0000_0000h	3.2.5.10/81
303D_0950	Memory Region Start Address (RDC_MRSA21)	32	R/W	Undefined	3.2.5.7/77
303D_0954	Memory Region End Address (RDC_MREA21)	32	R/W	Undefined	3.2.5.8/79
303D_0958	Memory Region Control (RDC_MRC21)	32	R/W	0000_00FFh	3.2.5.9/80

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_095C	Memory Region Violation Status (RDC_MRVS21)	32	R/W	0000_0000h	3.2.5.10/81
303D_0960	Memory Region Start Address (RDC_MRSA22)	32	R/W	Undefined	3.2.5.7/77
303D_0964	Memory Region End Address (RDC_MREA22)	32	R/W	Undefined	3.2.5.8/79
303D_0968	Memory Region Control (RDC_MRC22)	32	R/W	0000_00FFh	3.2.5.9/80
303D_096C	Memory Region Violation Status (RDC_MRVS22)	32	R/W	0000_0000h	3.2.5.10/81
303D_0970	Memory Region Start Address (RDC_MRSA23)	32	R/W	Undefined	3.2.5.7/77
303D_0974	Memory Region End Address (RDC_MREA23)	32	R/W	Undefined	3.2.5.8/79
303D_0978	Memory Region Control (RDC_MRC23)	32	R/W	0000_00FFh	3.2.5.9/80
303D_097C	Memory Region Violation Status (RDC_MRVS23)	32	R/W	0000_0000h	3.2.5.10/81
303D_0980	Memory Region Start Address (RDC_MRSA24)	32	R/W	Undefined	3.2.5.7/77
303D_0984	Memory Region End Address (RDC_MREA24)	32	R/W	Undefined	3.2.5.8/79
303D_0988	Memory Region Control (RDC_MRC24)	32	R/W	0000_00FFh	3.2.5.9/80
303D_098C	Memory Region Violation Status (RDC_MRVS24)	32	R/W	0000_0000h	3.2.5.10/81
303D_0990	Memory Region Start Address (RDC_MRSA25)	32	R/W	Undefined	3.2.5.7/77
303D_0994	Memory Region End Address (RDC_MREA25)	32	R/W	Undefined	3.2.5.8/79
303D_0998	Memory Region Control (RDC_MRC25)	32	R/W	0000_00FFh	3.2.5.9/80
303D_099C	Memory Region Violation Status (RDC_MRVS25)	32	R/W	0000_0000h	3.2.5.10/81
303D_09A0	Memory Region Start Address (RDC_MRSA26)	32	R/W	Undefined	3.2.5.7/77
303D_09A4	Memory Region End Address (RDC_MREA26)	32	R/W	Undefined	3.2.5.8/79
303D_09A8	Memory Region Control (RDC_MRC26)	32	R/W	0000_00FFh	3.2.5.9/80
303D_09AC	Memory Region Violation Status (RDC_MRVS26)	32	R/W	0000_0000h	3.2.5.10/81
303D_09B0	Memory Region Start Address (RDC_MRSA27)	32	R/W	Undefined	3.2.5.7/77
303D_09B4	Memory Region End Address (RDC_MREA27)	32	R/W	Undefined	3.2.5.8/79
303D_09B8	Memory Region Control (RDC_MRC27)	32	R/W	0000_00FFh	3.2.5.9/80
303D_09BC	Memory Region Violation Status (RDC_MRVS27)	32	R/W	0000_0000h	3.2.5.10/81
303D_09C0	Memory Region Start Address (RDC_MRSA28)	32	R/W	Undefined	3.2.5.7/77
303D_09C4	Memory Region End Address (RDC_MREA28)	32	R/W	Undefined	3.2.5.8/79
303D_09C8	Memory Region Control (RDC_MRC28)	32	R/W	0000_00FFh	3.2.5.9/80
303D_09CC	Memory Region Violation Status (RDC_MRVS28)	32	R/W	0000_0000h	3.2.5.10/81
303D_09D0	Memory Region Start Address (RDC_MRSA29)	32	R/W	Undefined	3.2.5.7/77
303D_09D4	Memory Region End Address (RDC_MREA29)	32	R/W	Undefined	3.2.5.8/79
303D_09D8	Memory Region Control (RDC_MRC29)	32	R/W	0000_00FFh	3.2.5.9/80
303D_09DC	Memory Region Violation Status (RDC_MRVS29)	32	R/W	0000_0000h	3.2.5.10/81
303D_09E0	Memory Region Start Address (RDC_MRSA30)	32	R/W	Undefined	3.2.5.7/77
303D_09E4	Memory Region End Address (RDC_MREA30)	32	R/W	Undefined	3.2.5.8/79
303D_09E8	Memory Region Control (RDC_MRC30)	32	R/W	0000_00FFh	3.2.5.9/80
303D_09EC	Memory Region Violation Status (RDC_MRVS30)	32	R/W	0000_0000h	3.2.5.10/81
303D_09F0	Memory Region Start Address (RDC_MRSA31)	32	R/W	Undefined	3.2.5.7/77

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_09F4	Memory Region End Address (RDC_MREA31)	32	R/W	Undefined	3.2.5.8/79
303D_09F8	Memory Region Control (RDC_MRC31)	32	R/W	0000_00FFh	3.2.5.9/80
303D_09FC	Memory Region Violation Status (RDC_MRVS31)	32	R/W	0000_0000h	3.2.5.10/81
303D_0A00	Memory Region Start Address (RDC_MRSA32)	32	R/W	Undefined	3.2.5.7/77
303D_0A04	Memory Region End Address (RDC_MREA32)	32	R/W	Undefined	3.2.5.8/79
303D_0A08	Memory Region Control (RDC_MRC32)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0A0C	Memory Region Violation Status (RDC_MRVS32)	32	R/W	0000_0000h	3.2.5.10/81
303D_0A10	Memory Region Start Address (RDC_MRSA33)	32	R/W	Undefined	3.2.5.7/77
303D_0A14	Memory Region End Address (RDC_MREA33)	32	R/W	Undefined	3.2.5.8/79
303D_0A18	Memory Region Control (RDC_MRC33)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0A1C	Memory Region Violation Status (RDC_MRVS33)	32	R/W	0000_0000h	3.2.5.10/81
303D_0A20	Memory Region Start Address (RDC_MRSA34)	32	R/W	Undefined	3.2.5.7/77
303D_0A24	Memory Region End Address (RDC_MREA34)	32	R/W	Undefined	3.2.5.8/79
303D_0A28	Memory Region Control (RDC_MRC34)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0A2C	Memory Region Violation Status (RDC_MRVS34)	32	R/W	0000_0000h	3.2.5.10/81
303D_0A30	Memory Region Start Address (RDC_MRSA35)	32	R/W	Undefined	3.2.5.7/77
303D_0A34	Memory Region End Address (RDC_MREA35)	32	R/W	Undefined	3.2.5.8/79
303D_0A38	Memory Region Control (RDC_MRC35)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0A3C	Memory Region Violation Status (RDC_MRVS35)	32	R/W	0000_0000h	3.2.5.10/81
303D_0A40	Memory Region Start Address (RDC_MRSA36)	32	R/W	Undefined	3.2.5.7/77
303D_0A44	Memory Region End Address (RDC_MREA36)	32	R/W	Undefined	3.2.5.8/79
303D_0A48	Memory Region Control (RDC_MRC36)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0A4C	Memory Region Violation Status (RDC_MRVS36)	32	R/W	0000_0000h	3.2.5.10/81
303D_0A50	Memory Region Start Address (RDC_MRSA37)	32	R/W	Undefined	3.2.5.7/77
303D_0A54	Memory Region End Address (RDC_MREA37)	32	R/W	Undefined	3.2.5.8/79
303D_0A58	Memory Region Control (RDC_MRC37)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0A5C	Memory Region Violation Status (RDC_MRVS37)	32	R/W	0000_0000h	3.2.5.10/81
303D_0A60	Memory Region Start Address (RDC_MRSA38)	32	R/W	Undefined	3.2.5.7/77
303D_0A64	Memory Region End Address (RDC_MREA38)	32	R/W	Undefined	3.2.5.8/79
303D_0A68	Memory Region Control (RDC_MRC38)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0A6C	Memory Region Violation Status (RDC_MRVS38)	32	R/W	0000_0000h	3.2.5.10/81
303D_0A70	Memory Region Start Address (RDC_MRSA39)	32	R/W	Undefined	3.2.5.7/77
303D_0A74	Memory Region End Address (RDC_MREA39)	32	R/W	Undefined	3.2.5.8/79
303D_0A78	Memory Region Control (RDC_MRC39)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0A7C	Memory Region Violation Status (RDC_MRVS39)	32	R/W	0000_0000h	3.2.5.10/81
303D_0A80	Memory Region Start Address (RDC_MRSA40)	32	R/W	Undefined	3.2.5.7/77
303D_0A84	Memory Region End Address (RDC_MREA40)	32	R/W	Undefined	3.2.5.8/79
303D_0A88	Memory Region Control (RDC_MRC40)	32	R/W	0000_00FFh	3.2.5.9/80

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0A8C	Memory Region Violation Status (RDC_MRVS40)	32	R/W	0000_0000h	3.2.5.10/81
303D_0A90	Memory Region Start Address (RDC_MRSA41)	32	R/W	Undefined	3.2.5.7/77
303D_0A94	Memory Region End Address (RDC_MREA41)	32	R/W	Undefined	3.2.5.8/79
303D_0A98	Memory Region Control (RDC_MRC41)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0A9C	Memory Region Violation Status (RDC_MRVS41)	32	R/W	0000_0000h	3.2.5.10/81
303D_0AA0	Memory Region Start Address (RDC_MRSA42)	32	R/W	Undefined	3.2.5.7/77
303D_0AA4	Memory Region End Address (RDC_MREA42)	32	R/W	Undefined	3.2.5.8/79
303D_0AA8	Memory Region Control (RDC_MRC42)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0AAC	Memory Region Violation Status (RDC_MRVS42)	32	R/W	0000_0000h	3.2.5.10/81
303D_0AB0	Memory Region Start Address (RDC_MRSA43)	32	R/W	Undefined	3.2.5.7/77
303D_0AB4	Memory Region End Address (RDC_MREA43)	32	R/W	Undefined	3.2.5.8/79
303D_0AB8	Memory Region Control (RDC_MRC43)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0ABC	Memory Region Violation Status (RDC_MRVS43)	32	R/W	0000_0000h	3.2.5.10/81
303D_0AC0	Memory Region Start Address (RDC_MRSA44)	32	R/W	Undefined	3.2.5.7/77
303D_0AC4	Memory Region End Address (RDC_MREA44)	32	R/W	Undefined	3.2.5.8/79
303D_0AC8	Memory Region Control (RDC_MRC44)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0ACC	Memory Region Violation Status (RDC_MRVS44)	32	R/W	0000_0000h	3.2.5.10/81
303D_0AD0	Memory Region Start Address (RDC_MRSA45)	32	R/W	Undefined	3.2.5.7/77
303D_0AD4	Memory Region End Address (RDC_MREA45)	32	R/W	Undefined	3.2.5.8/79
303D_0AD8	Memory Region Control (RDC_MRC45)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0ADC	Memory Region Violation Status (RDC_MRVS45)	32	R/W	0000_0000h	3.2.5.10/81
303D_0AE0	Memory Region Start Address (RDC_MRSA46)	32	R/W	Undefined	3.2.5.7/77
303D_0AE4	Memory Region End Address (RDC_MREA46)	32	R/W	Undefined	3.2.5.8/79
303D_0AE8	Memory Region Control (RDC_MRC46)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0AEC	Memory Region Violation Status (RDC_MRVS46)	32	R/W	0000_0000h	3.2.5.10/81
303D_0AF0	Memory Region Start Address (RDC_MRSA47)	32	R/W	Undefined	3.2.5.7/77
303D_0AF4	Memory Region End Address (RDC_MREA47)	32	R/W	Undefined	3.2.5.8/79
303D_0AF8	Memory Region Control (RDC_MRC47)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0AFC	Memory Region Violation Status (RDC_MRVS47)	32	R/W	0000_0000h	3.2.5.10/81
303D_0B00	Memory Region Start Address (RDC_MRSA48)	32	R/W	Undefined	3.2.5.7/77
303D_0B04	Memory Region End Address (RDC_MREA48)	32	R/W	Undefined	3.2.5.8/79
303D_0B08	Memory Region Control (RDC_MRC48)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0B0C	Memory Region Violation Status (RDC_MRVS48)	32	R/W	0000_0000h	3.2.5.10/81
303D_0B10	Memory Region Start Address (RDC_MRSA49)	32	R/W	Undefined	3.2.5.7/77
303D_0B14	Memory Region End Address (RDC_MREA49)	32	R/W	Undefined	3.2.5.8/79
303D_0B18	Memory Region Control (RDC_MRC49)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0B1C	Memory Region Violation Status (RDC_MRVS49)	32	R/W	0000_0000h	3.2.5.10/81
303D_0B20	Memory Region Start Address (RDC_MRSA50)	32	R/W	Undefined	3.2.5.7/77

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0B24	Memory Region End Address (RDC_MREA50)	32	R/W	Undefined	3.2.5.8/79
303D_0B28	Memory Region Control (RDC_MRC50)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0B2C	Memory Region Violation Status (RDC_MRVS50)	32	R/W	0000_0000h	3.2.5.10/81
303D_0B30	Memory Region Start Address (RDC_MRSA51)	32	R/W	Undefined	3.2.5.7/77
303D_0B34	Memory Region End Address (RDC_MREA51)	32	R/W	Undefined	3.2.5.8/79
303D_0B38	Memory Region Control (RDC_MRC51)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0B3C	Memory Region Violation Status (RDC_MRVS51)	32	R/W	0000_0000h	3.2.5.10/81
303D_0B40	Memory Region Start Address (RDC_MRSA52)	32	R/W	Undefined	3.2.5.7/77
303D_0B44	Memory Region End Address (RDC_MREA52)	32	R/W	Undefined	3.2.5.8/79
303D_0B48	Memory Region Control (RDC_MRC52)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0B4C	Memory Region Violation Status (RDC_MRVS52)	32	R/W	0000_0000h	3.2.5.10/81
303D_0B50	Memory Region Start Address (RDC_MRSA53)	32	R/W	Undefined	3.2.5.7/77
303D_0B54	Memory Region End Address (RDC_MREA53)	32	R/W	Undefined	3.2.5.8/79
303D_0B58	Memory Region Control (RDC_MRC53)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0B5C	Memory Region Violation Status (RDC_MRVS53)	32	R/W	0000_0000h	3.2.5.10/81
303D_0B60	Memory Region Start Address (RDC_MRSA54)	32	R/W	Undefined	3.2.5.7/77
303D_0B64	Memory Region End Address (RDC_MREA54)	32	R/W	Undefined	3.2.5.8/79
303D_0B68	Memory Region Control (RDC_MRC54)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0B6C	Memory Region Violation Status (RDC_MRVS54)	32	R/W	0000_0000h	3.2.5.10/81
303D_0B70	Memory Region Start Address (RDC_MRSA55)	32	R/W	Undefined	3.2.5.7/77
303D_0B74	Memory Region End Address (RDC_MREA55)	32	R/W	Undefined	3.2.5.8/79
303D_0B78	Memory Region Control (RDC_MRC55)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0B7C	Memory Region Violation Status (RDC_MRVS55)	32	R/W	0000_0000h	3.2.5.10/81
303D_0B80	Memory Region Start Address (RDC_MRSA56)	32	R/W	Undefined	3.2.5.7/77
303D_0B84	Memory Region End Address (RDC_MREA56)	32	R/W	Undefined	3.2.5.8/79
303D_0B88	Memory Region Control (RDC_MRC56)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0B8C	Memory Region Violation Status (RDC_MRVS56)	32	R/W	0000_0000h	3.2.5.10/81
303D_0B90	Memory Region Start Address (RDC_MRSA57)	32	R/W	Undefined	3.2.5.7/77
303D_0B94	Memory Region End Address (RDC_MREA57)	32	R/W	Undefined	3.2.5.8/79
303D_0B98	Memory Region Control (RDC_MRC57)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0B9C	Memory Region Violation Status (RDC_MRVS57)	32	R/W	0000_0000h	3.2.5.10/81
303D_0BA0	Memory Region Start Address (RDC_MRSA58)	32	R/W	Undefined	3.2.5.7/77
303D_0BA4	Memory Region End Address (RDC_MREA58)	32	R/W	Undefined	3.2.5.8/79
303D_0BA8	Memory Region Control (RDC_MRC58)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0BAC	Memory Region Violation Status (RDC_MRVS58)	32	R/W	0000_0000h	3.2.5.10/81
303D_0BB0	Memory Region Start Address (RDC_MRSA59)	32	R/W	Undefined	3.2.5.7/77
303D_0BB4	Memory Region End Address (RDC_MREA59)	32	R/W	Undefined	3.2.5.8/79
303D_0BB8	Memory Region Control (RDC_MRC59)	32	R/W	0000_00FFh	3.2.5.9/80

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0BBC	Memory Region Violation Status (RDC_MRVS59)	32	R/W	0000_0000h	3.2.5.10/81
303D_0BC0	Memory Region Start Address (RDC_MRSA60)	32	R/W	Undefined	3.2.5.7/77
303D_0BC4	Memory Region End Address (RDC_MREA60)	32	R/W	Undefined	3.2.5.8/79
303D_0BC8	Memory Region Control (RDC_MRC60)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0BCC	Memory Region Violation Status (RDC_MRVS60)	32	R/W	0000_0000h	3.2.5.10/81
303D_0BD0	Memory Region Start Address (RDC_MRSA61)	32	R/W	Undefined	3.2.5.7/77
303D_0BD4	Memory Region End Address (RDC_MREA61)	32	R/W	Undefined	3.2.5.8/79
303D_0BD8	Memory Region Control (RDC_MRC61)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0BDC	Memory Region Violation Status (RDC_MRVS61)	32	R/W	0000_0000h	3.2.5.10/81
303D_0BE0	Memory Region Start Address (RDC_MRSA62)	32	R/W	Undefined	3.2.5.7/77
303D_0BE4	Memory Region End Address (RDC_MREA62)	32	R/W	Undefined	3.2.5.8/79
303D_0BE8	Memory Region Control (RDC_MRC62)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0BEC	Memory Region Violation Status (RDC_MRVS62)	32	R/W	0000_0000h	3.2.5.10/81
303D_0BF0	Memory Region Start Address (RDC_MRSA63)	32	R/W	Undefined	3.2.5.7/77
303D_0BF4	Memory Region End Address (RDC_MREA63)	32	R/W	Undefined	3.2.5.8/79
303D_0BF8	Memory Region Control (RDC_MRC63)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0BFC	Memory Region Violation Status (RDC_MRVS63)	32	R/W	0000_0000h	3.2.5.10/81
303D_0C00	Memory Region Start Address (RDC_MRSA64)	32	R/W	Undefined	3.2.5.7/77
303D_0C04	Memory Region End Address (RDC_MREA64)	32	R/W	Undefined	3.2.5.8/79
303D_0C08	Memory Region Control (RDC_MRC64)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0C0C	Memory Region Violation Status (RDC_MRVS64)	32	R/W	0000_0000h	3.2.5.10/81
303D_0C10	Memory Region Start Address (RDC_MRSA65)	32	R/W	Undefined	3.2.5.7/77
303D_0C14	Memory Region End Address (RDC_MREA65)	32	R/W	Undefined	3.2.5.8/79
303D_0C18	Memory Region Control (RDC_MRC65)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0C1C	Memory Region Violation Status (RDC_MRVS65)	32	R/W	0000_0000h	3.2.5.10/81
303D_0C20	Memory Region Start Address (RDC_MRSA66)	32	R/W	Undefined	3.2.5.7/77
303D_0C24	Memory Region End Address (RDC_MREA66)	32	R/W	Undefined	3.2.5.8/79
303D_0C28	Memory Region Control (RDC_MRC66)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0C2C	Memory Region Violation Status (RDC_MRVS66)	32	R/W	0000_0000h	3.2.5.10/81
303D_0C30	Memory Region Start Address (RDC_MRSA67)	32	R/W	Undefined	3.2.5.7/77
303D_0C34	Memory Region End Address (RDC_MREA67)	32	R/W	Undefined	3.2.5.8/79
303D_0C38	Memory Region Control (RDC_MRC67)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0C3C	Memory Region Violation Status (RDC_MRVS67)	32	R/W	0000_0000h	3.2.5.10/81
303D_0C40	Memory Region Start Address (RDC_MRSA68)	32	R/W	Undefined	3.2.5.7/77
303D_0C44	Memory Region End Address (RDC_MREA68)	32	R/W	Undefined	3.2.5.8/79
303D_0C48	Memory Region Control (RDC_MRC68)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0C4C	Memory Region Violation Status (RDC_MRVS68)	32	R/W	0000_0000h	3.2.5.10/81
303D_0C50	Memory Region Start Address (RDC_MRSA69)	32	R/W	Undefined	3.2.5.7/77

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0C54	Memory Region End Address (RDC_MREA69)	32	R/W	Undefined	3.2.5.8/79
303D_0C58	Memory Region Control (RDC_MRC69)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0C5C	Memory Region Violation Status (RDC_MRVS69)	32	R/W	0000_0000h	3.2.5.10/81
303D_0C60	Memory Region Start Address (RDC_MRSA70)	32	R/W	Undefined	3.2.5.7/77
303D_0C64	Memory Region End Address (RDC_MREA70)	32	R/W	Undefined	3.2.5.8/79
303D_0C68	Memory Region Control (RDC_MRC70)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0C6C	Memory Region Violation Status (RDC_MRVS70)	32	R/W	0000_0000h	3.2.5.10/81
303D_0C70	Memory Region Start Address (RDC_MRSA71)	32	R/W	Undefined	3.2.5.7/77
303D_0C74	Memory Region End Address (RDC_MREA71)	32	R/W	Undefined	3.2.5.8/79
303D_0C78	Memory Region Control (RDC_MRC71)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0C7C	Memory Region Violation Status (RDC_MRVS71)	32	R/W	0000_0000h	3.2.5.10/81
303D_0C80	Memory Region Start Address (RDC_MRSA72)	32	R/W	Undefined	3.2.5.7/77
303D_0C84	Memory Region End Address (RDC_MREA72)	32	R/W	Undefined	3.2.5.8/79
303D_0C88	Memory Region Control (RDC_MRC72)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0C8C	Memory Region Violation Status (RDC_MRVS72)	32	R/W	0000_0000h	3.2.5.10/81
303D_0C90	Memory Region Start Address (RDC_MRSA73)	32	R/W	Undefined	3.2.5.7/77
303D_0C94	Memory Region End Address (RDC_MREA73)	32	R/W	Undefined	3.2.5.8/79
303D_0C98	Memory Region Control (RDC_MRC73)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0C9C	Memory Region Violation Status (RDC_MRVS73)	32	R/W	0000_0000h	3.2.5.10/81
303D_0CA0	Memory Region Start Address (RDC_MRSA74)	32	R/W	Undefined	3.2.5.7/77
303D_0CA4	Memory Region End Address (RDC_MREA74)	32	R/W	Undefined	3.2.5.8/79
303D_0CA8	Memory Region Control (RDC_MRC74)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0CAC	Memory Region Violation Status (RDC_MRVS74)	32	R/W	0000_0000h	3.2.5.10/81
303D_0CB0	Memory Region Start Address (RDC_MRSA75)	32	R/W	Undefined	3.2.5.7/77
303D_0CB4	Memory Region End Address (RDC_MREA75)	32	R/W	Undefined	3.2.5.8/79
303D_0CB8	Memory Region Control (RDC_MRC75)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0CBC	Memory Region Violation Status (RDC_MRVS75)	32	R/W	0000_0000h	3.2.5.10/81
303D_0CC0	Memory Region Start Address (RDC_MRSA76)	32	R/W	Undefined	3.2.5.7/77
303D_0CC4	Memory Region End Address (RDC_MREA76)	32	R/W	Undefined	3.2.5.8/79
303D_0CC8	Memory Region Control (RDC_MRC76)	32	R/W	0000_00FFh	3.2.5.9/80
303D_0CCC	Memory Region Violation Status (RDC_MRVS76)	32	R/W	0000_0000h	3.2.5.10/81

3.2.5.1 Version Information (RDC_VIR)

The VIR provides version information including the number of domains, number of master slots, number of peripheral slots, and number of memory regions.

Address: 303D_0000h base + 0h offset = 303D_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				NRGN								NPER								NMSTR								NDID			
W																																
Reset	0	0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	0	0	1	0	0	0	1	0	1	0	0	0	0	1	0	0

RDC_VIR field descriptions

Field	Description
31–28 Reserved	This field is reserved.
27–20 NRGN	Number of Memory Regions Indicates the number of memory regions in this instance of the RDC.
19–12 NPER	Number of Peripherals Indicates the number of peripherals that can be isolated or safe-shared.
11–4 NMSTR	Number of Masters Indicates the number of masters supported by this instance of RDC.
NDID	Number of Domains Indicates the number of domain IDs supported by this instance of the RDC. For example, value '0010' means the actual number of domains is 2.

3.2.5.2 Status (RDC_STAT)

Address: 303D_0000h base + 24h offset = 303D_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							PDS	Reserved				DID			
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

RDC_STAT field descriptions

Field	Description
31–9 Reserved	This field is reserved.
8 PDS	Power Domain Status Indicates if the "Power Down" memory regions are powered and available. Power Down memory regions are only those memory regions susceptible to power outage for power savings are unavailable if this is zero. "Always-On" memory regions remain available. Always On memory regions are those regions that are not powered down unless the entire SoC is powered down. This signal remains low until all access controls have been restored to the domain. 0 Power Down Domain is OFF 1 Power Down Domain is ON
7–4 Reserved	This field is reserved.
DID	Domain ID The Domain ID of the core or bus master that is reading this. The value is different for requests from different domains.

3.2.5.3 Interrupt and Control (RDC_INTCTRL)

Address: 303D_0000h base + 28h offset = 303D_0028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															RCI_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDC_INTCTRL field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 RCI_EN	Restoration Complete Interrupt Interrupt generated when the RDC has completed restoring state to a recently re-powered memory regions.

Table continues on the next page...

RDC_INTCTRL field descriptions (continued)

Field	Description
	0 Interrupt Disabled 1 Interrupt Enabled

3.2.5.4 Interrupt Status (RDC_INTSTAT)

Indication of Interrupt Pending for State Restoration

Address: 303D_0000h base + 2Ch offset = 303D_002Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															INT w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDC_INTSTAT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 INT	Interrupt Status Indicates state of interrupt signal for state restoration. This is that status of the interrupt enabled in RDC_INTCTRL. Write one to interrupt status to clear it. 0 No Interrupt Pending 1 Interrupt Pending

3.2.5.5 Master Domain Assignment (RDC_MDA*n*)

Address: 303D_0000h base + 200h offset + (4d × i), where i=0d to 39d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LCK	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DID
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDC_MDA*n* field descriptions

Field	Description
31 LCK	Assignment Lock 0 Not Locked 1 Locked
30–2 Reserved	This field is reserved.
DID	Domain ID Indicates the domain to which the Master is assigned 00 Master assigned to Processing Domain 0 01 Master assigned to Processing Domain 1 10 Master assigned to Processing Domain 2 11 Master assigned to Processing Domain 3

3.2.5.6 Peripheral Domain Access Permissions (RDC_PDAP*n*)

Address: 303D_0000h base + 400h offset + (4d × i), where i=0d to 111d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LCK	SREQ	Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								D3R	D3W	D2R	D2W	D1R	D1W	D0R	D0W
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

RDC_PDAP*n* field descriptions

Field	Description
31 LCK	Peripheral Permissions Lock When set prevents further modification of the Peripheral Domain Access Permissions (sticky bit until reset) 0 Not Locked 1 Locked
30 SREQ	Semaphore Required When set the hardware semaphore state enforces the semaphore lock. If a domain has access permissions and a semaphore has locked a shared peripheral then only the domain holding the semaphore signal can access this peripheral.

Table continues on the next page...

RDC_PDAP*n* field descriptions (continued)

Field	Description
	0 Semaphores have no effect 1 Semaphores are enforced
29–8 Reserved	This field is reserved.
7 D3R	Domain 3 Read Access 0 No Read Access 1 Read Access Allowed
6 D3W	Domain 3 Write Access 0 No Write Access 1 Write Access Allowed
5 D2R	Domain 2 Read Access 0 No Read Access 1 Read Access Allowed
4 D2W	Domain 2 Write Access 0 No Write Access 1 Write Access Allowed
3 D1R	Domain 1 Read Access 0 No Read Access 1 Read Access Allowed
2 D1W	Domain 1 Write Access 0 No Write Access 1 Write Access Allowed
1 D0R	Domain 0 Read Access 0 No Read Access 1 Read Access Allowed
0 D0W	Domain 0 Write Access 0 No Write Access 1 Write Access Allowed

3.2.5.7 Memory Region Start Address (RDC_MRSA*n*)

NOTE

The DDR space is 33-bit width. The RDC memory region registers are 32-bit width. The RDC configuration is the most significant bits in the DDR address space (32:1). To set the start address for this configuration, the MRSA value should be shifted 1-bit and added to the DDR base address. The example below illustrates how to calculate the proper start and end address value. Please refer to the Memory Map for the actual DDR base address.

```
Start Address:      0x5000_0000
End Address:        0xAE00_0000
DDR Base Address:   0x4000_0000

Calculating MRSA value:

0x5000_0000 - 0x4000_0000 = 0x1000_0000   // Desired Start - DDR Base
0x1000_0000 / 2 = 0x800_0000              // Right-shift 1 bit

MRSA Value:        0x800_0000

Calculating MREA value:

0xAE00_0000 - 0x4000_0000 = 0x6E00_0000   // Desired End - DDR Base
0x6E00_0000 / 2 = 0x3700_0000             // Right-shift 1 bit

MREA Value:        0x3700_0000
```

Figure 3-5. Calculating Address Value Example

Address: 303D_0000h base + 800h offset + (16d × i), where i=0d to 76d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SADR																Reserved															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

RDC_MRSA*n* field descriptions

Field	Description
31–7 SADR	Start address for memory region

Table continues on the next page...

RDC_MRSA*n* field descriptions (continued)

Field	Description
	Lower bound (inclusive) modulo the defined granularity byte size of a region. The region size (granularity) is defined for each Memory/Port in the Memory Region Map section. Region boundaries are aligned to the minimum possible region size for the Memory/Port.
Reserved	This field is reserved.

3.2.5.8 Memory Region End Address (RDC_MREA*n*)

NOTE

The DDR space is 33-bit width. The RDC memory region registers are 32-bit width. The RDC configuration is the most significant bits in the DDR address space (32:1). To set the start address for this configuration, the MRSA value should be shifted 1-bit and added to the DDR base address. The example below illustrates how to calculate the proper start and end address value. Please refer to the Memory Map for the actual DDR base address.

```
Start Address:      0x5000_0000
End Address:        0xAE00_0000
DDR Base Address:   0x4000_0000

Calculating MRSA value:

0x5000_0000 - 0x4000_0000 = 0x1000_0000   // Desired Start - DDR Base
0x1000_0000 / 2 = 0x800_0000              // Right-shift 1 bit

MRSA Value:        0x800_0000

Calculating MREA value:

0xAE00_0000 - 0x4000_0000 = 0x6E00_0000   // Desired End - DDR Base
0x6E00_0000 / 2 = 0x3700_0000             // Right-shift 1 bit

MREA Value:        0x3700_0000
```

Figure 3-6. Calculating Address Value Example

Address: 303D_0000h base + 804h offset + (16d × i), where i=0d to 76d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EADR																Reserved															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

RDC_MREA*n* field descriptions

Field	Description
31–7 EADR	Upper bound for memory region Upper bound (exclusive) modulo the defined granularity byte size of a region. The region size (granularity) is defined for each Memory/Port in the Memory Region Map section. Region boundaries are aligned to the minimum possible region size for the Memory/Port.
Reserved	This field is reserved.

3.2.5.9 Memory Region Control (RDC_MRC*n*)

Address: 303D_0000h base + 808h offset + (16d × i), where i=0d to 76d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LCK	ENA	Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								D3R	D3W	D2R	D2W	D1R	D1W	D0R	D0W
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

RDC_MRC*n* field descriptions

Field	Description
31 LCK	Region Lock Locks all region fields from further modification except ENA, which can be set but not reset after LCK is set. LCK is a sticky bit. 0 No Lock. All fields in this register may be modified. 1 Locked. No fields in this register may be modified except ENA, which may be set but not cleared.
30 ENA	Region Enable Activates the memory region. If the region is not activated then the permissions and address boundaries have not affect and the region will be fully accessible. 0 Memory region is not defined or restricted. 1 Memory boundaries, domain permissions and controls are in effect.
29–8 Reserved	This field is reserved.
7 D3R	Domain 3 Read Access to Region 0 Processing Domain 3 does not have Read access to the memory region 1 Processing Domain 3 has Read access to the memory region
6 D3W	Domain 3 Write Access to Region

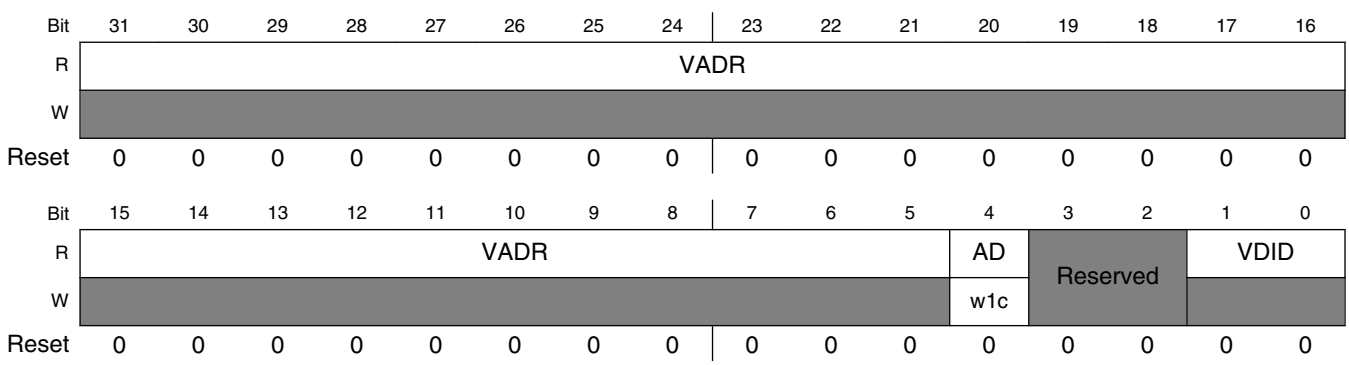
Table continues on the next page...

RDC_MRC*n* field descriptions (continued)

Field	Description
	0 Processing Domain 3 does not have Write access to the memory region 1 Processing Domain 3 has Read access to the memory region
5 D2R	Domain 2 Read Access to Region 0 Processing Domain 2 does not have Read access to the memory region 1 Processing Domain 2 has Read access to the memory region
4 D2W	Domain 2 Write Access to Region 0 Processing Domain 2 does not have Write access to the memory region 1 Processing Domain 2 has Write access to the memory region
3 D1R	Domain 1 Read Access to Region 0 Processing Domain 1 does not have Read access to the memory region 1 Processing Domain 1 has Read access to the memory region
2 D1W	Domain 1 Write Access to Region 0 Processing Domain 1 does not have Write access to the memory region 1 Processing Domain 1 has Write access to the memory region
1 D0R	Domain 0 Read Access to Region 0 Processing Domain 0 does not have Read access to the memory region 1 Processing Domain 0 has Read access to the memory region
0 D0W	Domain 0 Write Access to Region 0 Processing Domain 0 does not have Write access to the memory region 1 Processing Domain 0 has Write access to the memory region

3.2.5.10 Memory Region Violation Status (RDC_MRVS*n*)

Address: 303D_0000h base + 80Ch offset + (16d × i), where i=0d to 76d



RDC_MRVS*n* field descriptions

Field	Description
31–5 VADR	Violating Address

Table continues on the next page...

RDC_MRVS*n* field descriptions (continued)

Field	Description
	The address of the denied access. The first access violation is captured. Subsequent violations are ignored until the status register is cleared. Contents are cleared upon reading the register. Clearing of contents occurs only when the status is read by the memory region's associated domain ID (s).
4 AD	Access Denied Access to a memory region denied. This bit is cleared when this bit is written by one of the allowed domains.
3–2 Reserved	This field is reserved.
VDID	Violating Domain ID The domain ID of the denied access. The first access violation is captured. Subsequent violations are ignored until the status register is cleared. Contents are cleared upon reading the register. 00 Processing Domain 0 01 Processing Domain 1 10 Processing Domain 2 11 Processing Domain 3

3.2.6 RDC SEMA42 Memory Map/Register Definition

Only Supervisor Mode accesses are allowed on these registers. User accesses generate an error termination.

RDC_SEMAPHORE memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303B_0000	Gate Register (RDC_SEMAPHORE1_GATE0)	8	R/W	00h	3.2.6.1/86
303B_0001	Gate Register (RDC_SEMAPHORE1_GATE1)	8	R/W	00h	3.2.6.1/86
303B_0002	Gate Register (RDC_SEMAPHORE1_GATE2)	8	R/W	00h	3.2.6.1/86
303B_0003	Gate Register (RDC_SEMAPHORE1_GATE3)	8	R/W	00h	3.2.6.1/86
303B_0004	Gate Register (RDC_SEMAPHORE1_GATE4)	8	R/W	00h	3.2.6.1/86
303B_0005	Gate Register (RDC_SEMAPHORE1_GATE5)	8	R/W	00h	3.2.6.1/86
303B_0006	Gate Register (RDC_SEMAPHORE1_GATE6)	8	R/W	00h	3.2.6.1/86
303B_0007	Gate Register (RDC_SEMAPHORE1_GATE7)	8	R/W	00h	3.2.6.1/86
303B_0008	Gate Register (RDC_SEMAPHORE1_GATE8)	8	R/W	00h	3.2.6.1/86
303B_0009	Gate Register (RDC_SEMAPHORE1_GATE9)	8	R/W	00h	3.2.6.1/86
303B_000A	Gate Register (RDC_SEMAPHORE1_GATE10)	8	R/W	00h	3.2.6.1/86
303B_000B	Gate Register (RDC_SEMAPHORE1_GATE11)	8	R/W	00h	3.2.6.1/86

Table continues on the next page...

RDC_SEMAPHORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303B_000C	Gate Register (RDC_SEMAPHORE1_GATE12)	8	R/W	00h	3.2.6.1/86
303B_000D	Gate Register (RDC_SEMAPHORE1_GATE13)	8	R/W	00h	3.2.6.1/86
303B_000E	Gate Register (RDC_SEMAPHORE1_GATE14)	8	R/W	00h	3.2.6.1/86
303B_000F	Gate Register (RDC_SEMAPHORE1_GATE15)	8	R/W	00h	3.2.6.1/86
303B_0010	Gate Register (RDC_SEMAPHORE1_GATE16)	8	R/W	00h	3.2.6.1/86
303B_0011	Gate Register (RDC_SEMAPHORE1_GATE17)	8	R/W	00h	3.2.6.1/86
303B_0012	Gate Register (RDC_SEMAPHORE1_GATE18)	8	R/W	00h	3.2.6.1/86
303B_0013	Gate Register (RDC_SEMAPHORE1_GATE19)	8	R/W	00h	3.2.6.1/86
303B_0014	Gate Register (RDC_SEMAPHORE1_GATE20)	8	R/W	00h	3.2.6.1/86
303B_0015	Gate Register (RDC_SEMAPHORE1_GATE21)	8	R/W	00h	3.2.6.1/86
303B_0016	Gate Register (RDC_SEMAPHORE1_GATE22)	8	R/W	00h	3.2.6.1/86
303B_0017	Gate Register (RDC_SEMAPHORE1_GATE23)	8	R/W	00h	3.2.6.1/86
303B_0018	Gate Register (RDC_SEMAPHORE1_GATE24)	8	R/W	00h	3.2.6.1/86
303B_0019	Gate Register (RDC_SEMAPHORE1_GATE25)	8	R/W	00h	3.2.6.1/86
303B_001A	Gate Register (RDC_SEMAPHORE1_GATE26)	8	R/W	00h	3.2.6.1/86
303B_001B	Gate Register (RDC_SEMAPHORE1_GATE27)	8	R/W	00h	3.2.6.1/86
303B_001C	Gate Register (RDC_SEMAPHORE1_GATE28)	8	R/W	00h	3.2.6.1/86
303B_001D	Gate Register (RDC_SEMAPHORE1_GATE29)	8	R/W	00h	3.2.6.1/86
303B_001E	Gate Register (RDC_SEMAPHORE1_GATE30)	8	R/W	00h	3.2.6.1/86
303B_001F	Gate Register (RDC_SEMAPHORE1_GATE31)	8	R/W	00h	3.2.6.1/86
303B_0020	Gate Register (RDC_SEMAPHORE1_GATE32)	8	R/W	00h	3.2.6.1/86
303B_0021	Gate Register (RDC_SEMAPHORE1_GATE33)	8	R/W	00h	3.2.6.1/86
303B_0022	Gate Register (RDC_SEMAPHORE1_GATE34)	8	R/W	00h	3.2.6.1/86
303B_0023	Gate Register (RDC_SEMAPHORE1_GATE35)	8	R/W	00h	3.2.6.1/86
303B_0024	Gate Register (RDC_SEMAPHORE1_GATE36)	8	R/W	00h	3.2.6.1/86
303B_0025	Gate Register (RDC_SEMAPHORE1_GATE37)	8	R/W	00h	3.2.6.1/86
303B_0026	Gate Register (RDC_SEMAPHORE1_GATE38)	8	R/W	00h	3.2.6.1/86
303B_0027	Gate Register (RDC_SEMAPHORE1_GATE39)	8	R/W	00h	3.2.6.1/86
303B_0028	Gate Register (RDC_SEMAPHORE1_GATE40)	8	R/W	00h	3.2.6.1/86
303B_0029	Gate Register (RDC_SEMAPHORE1_GATE41)	8	R/W	00h	3.2.6.1/86
303B_002A	Gate Register (RDC_SEMAPHORE1_GATE42)	8	R/W	00h	3.2.6.1/86
303B_002B	Gate Register (RDC_SEMAPHORE1_GATE43)	8	R/W	00h	3.2.6.1/86
303B_002C	Gate Register (RDC_SEMAPHORE1_GATE44)	8	R/W	00h	3.2.6.1/86
303B_002D	Gate Register (RDC_SEMAPHORE1_GATE45)	8	R/W	00h	3.2.6.1/86
303B_002E	Gate Register (RDC_SEMAPHORE1_GATE46)	8	R/W	00h	3.2.6.1/86
303B_002F	Gate Register (RDC_SEMAPHORE1_GATE47)	8	R/W	00h	3.2.6.1/86
303B_0030	Gate Register (RDC_SEMAPHORE1_GATE48)	8	R/W	00h	3.2.6.1/86
303B_0031	Gate Register (RDC_SEMAPHORE1_GATE49)	8	R/W	00h	3.2.6.1/86

Table continues on the next page...

RDC_SEMAPHORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303B_0032	Gate Register (RDC_SEMAPHORE1_GATE50)	8	R/W	00h	3.2.6.1/86
303B_0033	Gate Register (RDC_SEMAPHORE1_GATE51)	8	R/W	00h	3.2.6.1/86
303B_0034	Gate Register (RDC_SEMAPHORE1_GATE52)	8	R/W	00h	3.2.6.1/86
303B_0035	Gate Register (RDC_SEMAPHORE1_GATE53)	8	R/W	00h	3.2.6.1/86
303B_0036	Gate Register (RDC_SEMAPHORE1_GATE54)	8	R/W	00h	3.2.6.1/86
303B_0037	Gate Register (RDC_SEMAPHORE1_GATE55)	8	R/W	00h	3.2.6.1/86
303B_0038	Gate Register (RDC_SEMAPHORE1_GATE56)	8	R/W	00h	3.2.6.1/86
303B_0039	Gate Register (RDC_SEMAPHORE1_GATE57)	8	R/W	00h	3.2.6.1/86
303B_003A	Gate Register (RDC_SEMAPHORE1_GATE58)	8	R/W	00h	3.2.6.1/86
303B_003B	Gate Register (RDC_SEMAPHORE1_GATE59)	8	R/W	00h	3.2.6.1/86
303B_003C	Gate Register (RDC_SEMAPHORE1_GATE60)	8	R/W	00h	3.2.6.1/86
303B_003D	Gate Register (RDC_SEMAPHORE1_GATE61)	8	R/W	00h	3.2.6.1/86
303B_003E	Gate Register (RDC_SEMAPHORE1_GATE62)	8	R/W	00h	3.2.6.1/86
303B_003F	Gate Register (RDC_SEMAPHORE1_GATE63)	8	R/W	00h	3.2.6.1/86
303B_0042	Reset Gate Write (RDC_SEMAPHORE1_RSTGT_W)	16	R/W	0000h	3.2.6.2/87
303B_0042	Reset Gate Read (RDC_SEMAPHORE1_RSTGT_R)	16	R/W	0000h	3.2.6.3/88
303C_0000	Gate Register (RDC_SEMAPHORE2_GATE0)	8	R/W	00h	3.2.6.1/86
303C_0001	Gate Register (RDC_SEMAPHORE2_GATE1)	8	R/W	00h	3.2.6.1/86
303C_0002	Gate Register (RDC_SEMAPHORE2_GATE2)	8	R/W	00h	3.2.6.1/86
303C_0003	Gate Register (RDC_SEMAPHORE2_GATE3)	8	R/W	00h	3.2.6.1/86
303C_0004	Gate Register (RDC_SEMAPHORE2_GATE4)	8	R/W	00h	3.2.6.1/86
303C_0005	Gate Register (RDC_SEMAPHORE2_GATE5)	8	R/W	00h	3.2.6.1/86
303C_0006	Gate Register (RDC_SEMAPHORE2_GATE6)	8	R/W	00h	3.2.6.1/86
303C_0007	Gate Register (RDC_SEMAPHORE2_GATE7)	8	R/W	00h	3.2.6.1/86
303C_0008	Gate Register (RDC_SEMAPHORE2_GATE8)	8	R/W	00h	3.2.6.1/86
303C_0009	Gate Register (RDC_SEMAPHORE2_GATE9)	8	R/W	00h	3.2.6.1/86
303C_000A	Gate Register (RDC_SEMAPHORE2_GATE10)	8	R/W	00h	3.2.6.1/86
303C_000B	Gate Register (RDC_SEMAPHORE2_GATE11)	8	R/W	00h	3.2.6.1/86
303C_000C	Gate Register (RDC_SEMAPHORE2_GATE12)	8	R/W	00h	3.2.6.1/86
303C_000D	Gate Register (RDC_SEMAPHORE2_GATE13)	8	R/W	00h	3.2.6.1/86
303C_000E	Gate Register (RDC_SEMAPHORE2_GATE14)	8	R/W	00h	3.2.6.1/86
303C_000F	Gate Register (RDC_SEMAPHORE2_GATE15)	8	R/W	00h	3.2.6.1/86
303C_0010	Gate Register (RDC_SEMAPHORE2_GATE16)	8	R/W	00h	3.2.6.1/86
303C_0011	Gate Register (RDC_SEMAPHORE2_GATE17)	8	R/W	00h	3.2.6.1/86
303C_0012	Gate Register (RDC_SEMAPHORE2_GATE18)	8	R/W	00h	3.2.6.1/86
303C_0013	Gate Register (RDC_SEMAPHORE2_GATE19)	8	R/W	00h	3.2.6.1/86
303C_0014	Gate Register (RDC_SEMAPHORE2_GATE20)	8	R/W	00h	3.2.6.1/86
303C_0015	Gate Register (RDC_SEMAPHORE2_GATE21)	8	R/W	00h	3.2.6.1/86

Table continues on the next page...

RDC_SEMAPHORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303C_0016	Gate Register (RDC_SEMAPHORE2_GATE22)	8	R/W	00h	3.2.6.1/86
303C_0017	Gate Register (RDC_SEMAPHORE2_GATE23)	8	R/W	00h	3.2.6.1/86
303C_0018	Gate Register (RDC_SEMAPHORE2_GATE24)	8	R/W	00h	3.2.6.1/86
303C_0019	Gate Register (RDC_SEMAPHORE2_GATE25)	8	R/W	00h	3.2.6.1/86
303C_001A	Gate Register (RDC_SEMAPHORE2_GATE26)	8	R/W	00h	3.2.6.1/86
303C_001B	Gate Register (RDC_SEMAPHORE2_GATE27)	8	R/W	00h	3.2.6.1/86
303C_001C	Gate Register (RDC_SEMAPHORE2_GATE28)	8	R/W	00h	3.2.6.1/86
303C_001D	Gate Register (RDC_SEMAPHORE2_GATE29)	8	R/W	00h	3.2.6.1/86
303C_001E	Gate Register (RDC_SEMAPHORE2_GATE30)	8	R/W	00h	3.2.6.1/86
303C_001F	Gate Register (RDC_SEMAPHORE2_GATE31)	8	R/W	00h	3.2.6.1/86
303C_0020	Gate Register (RDC_SEMAPHORE2_GATE32)	8	R/W	00h	3.2.6.1/86
303C_0021	Gate Register (RDC_SEMAPHORE2_GATE33)	8	R/W	00h	3.2.6.1/86
303C_0022	Gate Register (RDC_SEMAPHORE2_GATE34)	8	R/W	00h	3.2.6.1/86
303C_0023	Gate Register (RDC_SEMAPHORE2_GATE35)	8	R/W	00h	3.2.6.1/86
303C_0024	Gate Register (RDC_SEMAPHORE2_GATE36)	8	R/W	00h	3.2.6.1/86
303C_0025	Gate Register (RDC_SEMAPHORE2_GATE37)	8	R/W	00h	3.2.6.1/86
303C_0026	Gate Register (RDC_SEMAPHORE2_GATE38)	8	R/W	00h	3.2.6.1/86
303C_0027	Gate Register (RDC_SEMAPHORE2_GATE39)	8	R/W	00h	3.2.6.1/86
303C_0028	Gate Register (RDC_SEMAPHORE2_GATE40)	8	R/W	00h	3.2.6.1/86
303C_0029	Gate Register (RDC_SEMAPHORE2_GATE41)	8	R/W	00h	3.2.6.1/86
303C_002A	Gate Register (RDC_SEMAPHORE2_GATE42)	8	R/W	00h	3.2.6.1/86
303C_002B	Gate Register (RDC_SEMAPHORE2_GATE43)	8	R/W	00h	3.2.6.1/86
303C_002C	Gate Register (RDC_SEMAPHORE2_GATE44)	8	R/W	00h	3.2.6.1/86
303C_002D	Gate Register (RDC_SEMAPHORE2_GATE45)	8	R/W	00h	3.2.6.1/86
303C_002E	Gate Register (RDC_SEMAPHORE2_GATE46)	8	R/W	00h	3.2.6.1/86
303C_002F	Gate Register (RDC_SEMAPHORE2_GATE47)	8	R/W	00h	3.2.6.1/86
303C_0030	Gate Register (RDC_SEMAPHORE2_GATE48)	8	R/W	00h	3.2.6.1/86
303C_0031	Gate Register (RDC_SEMAPHORE2_GATE49)	8	R/W	00h	3.2.6.1/86
303C_0032	Gate Register (RDC_SEMAPHORE2_GATE50)	8	R/W	00h	3.2.6.1/86
303C_0033	Gate Register (RDC_SEMAPHORE2_GATE51)	8	R/W	00h	3.2.6.1/86
303C_0034	Gate Register (RDC_SEMAPHORE2_GATE52)	8	R/W	00h	3.2.6.1/86
303C_0035	Gate Register (RDC_SEMAPHORE2_GATE53)	8	R/W	00h	3.2.6.1/86
303C_0036	Gate Register (RDC_SEMAPHORE2_GATE54)	8	R/W	00h	3.2.6.1/86
303C_0037	Gate Register (RDC_SEMAPHORE2_GATE55)	8	R/W	00h	3.2.6.1/86
303C_0038	Gate Register (RDC_SEMAPHORE2_GATE56)	8	R/W	00h	3.2.6.1/86
303C_0039	Gate Register (RDC_SEMAPHORE2_GATE57)	8	R/W	00h	3.2.6.1/86
303C_003A	Gate Register (RDC_SEMAPHORE2_GATE58)	8	R/W	00h	3.2.6.1/86
303C_003B	Gate Register (RDC_SEMAPHORE2_GATE59)	8	R/W	00h	3.2.6.1/86

Table continues on the next page...

RDC_SEMAPHORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303C_003C	Gate Register (RDC_SEMAPHORE2_GATE60)	8	R/W	00h	3.2.6.1/86
303C_003D	Gate Register (RDC_SEMAPHORE2_GATE61)	8	R/W	00h	3.2.6.1/86
303C_003E	Gate Register (RDC_SEMAPHORE2_GATE62)	8	R/W	00h	3.2.6.1/86
303C_003F	Gate Register (RDC_SEMAPHORE2_GATE63)	8	R/W	00h	3.2.6.1/86
303C_0042	Reset Gate Write (RDC_SEMAPHORE2_RSTGT_W)	16	R/W	0000h	3.2.6.2/87
303C_0042	Reset Gate Read (RDC_SEMAPHORE2_RSTGT_R)	16	R/W	0000h	3.2.6.3/88

3.2.6.1 Gate Register (RDC_SEMAPHORE*x*_GATE*n*)

Each semaphore gate is implemented in a 4-bit finite state machine, right-justified in a byte data structure. The hardware uses the logical bus master number (master_index) in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. Attempted writes with a data value that is neither the unlock value nor the appropriate lock value (master_index + 1) are simply treated as "no operation" and do not affect any gate state. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes. Processor dex values can be found in AIPSTZ Memory Map/Register Definition.

Address: Base address + 0h offset + (1d × i), where i=0d to 63d

Bit	7	6	5	4	3	2	1	0
Read	0		LDOM		GTFSM			
Write								
Reset	0	0	0	0	0	0	0	0

RDC_SEMAPHORE*x*_GATE*n* field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 LDOM	Read-only bits. They indicate which domain had currently locked the gate. 00 The gate is locked by domain 0. (True if the field GTFSM does not equal to 0000.) 01 The gate has been locked by domain 1. 10 The gate has been locked by domain 2. 11 The gate has been locked by domain 3.

Table continues on the next page...

RDC_SEMAPHORE*x*_GATE*n* field descriptions (continued)

Field	Description
GTFSM	Gate Finite State Machine. The state of the gate reflects the last processor that locked it, which can be useful during system debug. The hardware gate is maintained in a 16-state implementation, defined as: 0000 The gate is unlocked (free). 0001 The gate has been locked by processor with master_index = 0. 0010 The gate has been locked by processor with master_index = 1. 0011 The gate has been locked by processor with master_index = 2. 0100 The gate has been locked by processor with master_index = 3. 0101 The gate has been locked by processor with master_index = 4. 0110 The gate has been locked by processor with master_index = 5. 0111 The gate has been locked by processor with master_index = 6. 1000 The gate has been locked by processor with master_index = 7. 1001 The gate has been locked by processor with master_index = 8. 1010 The gate has been locked by processor with master_index = 9. 1011 The gate has been locked by processor with master_index = 10. 1100 The gate has been locked by processor with master_index = 11. 1101 The gate has been locked by processor with master_index = 12. 1110 The gate has been locked by processor with master_index = 13. 1111 The gate has been locked by processor with master_index = 14.

3.2.6.2 Reset Gate Write (RDC_SEMAPHORE*x*_RSTGT_W)

Although the intent of the hardware gate implementation specifies a protocol where the locking processor must unlock the gate, it is recognized that system operation may require a reset function to re-initialize the state of any gate(s) without requiring a system-level reset.

To support this special gate reset requirement, the RDC Semaphores module implements a "secure" reset mechanism that allows a hardware gate (or all the gates) to be initialized by following a specific dual-write access pattern. Using a technique similar to that required for the servicing of a software watchdog timer, the secure gate reset requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the specified gate(s). The required access pattern is:

1. A processor performs a 16-bit write to the RDC_SEMA42RSTGT memory location. The least significant byte (RDC_SEMA42RSTGT[RSTGDP]) must be 0xE2; the most significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the RDC_SEMA42RSTGT location. For this write, the lower byte (RDC_SEMA42RSTGT[RSTGDP]) is the logical complement of the first data pattern (0x1D) and the upper byte (RDC_SEMA42RSTGT[RSTGTN]) specifies the

gate(s) to be reset. This gate field can specify a single gate be cleared, or else that all gates are to be cleared. If the same processor writes incorrect data on the second access or another processor performs the second write access, the special gate reset sequence is aborted and no error signal will be asserted.

3. Reads of the RDC_SEMA42RSTGT location return information on the 2-bit state machine (RDC_SEMA42RSTGT[RSTGSM]) that implements this function, the bus master performing the reset (RDC_SEMA42RSTGT[RSTGMS]), and the gate number(s) last cleared (RDC_SEMA42RSTGT[RSTGTN]). Reads of the RDC_SEMA42RSTGT register do not affect the secure reset finite state machine in any manner.

Address: Base address + 42h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	RSTGTN								0							
Write									RSTGDP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDC_SEMAPHOREx_RSTGT_W field descriptions

Field	Description
15–8 RSTGTN	Reset Gate Number. This 8-bit field specifies the specific hardware gate to be reset. This field is updated by the second write. This field contains the hexidecimal value of the gate number. If RSTGTN < 64, then reset the single gate defined by RSTGTN, else reset all the gates.
RSTGDP	Reset Gate Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the gate reset mechanism. For the first write, RSTGDP = 0xE2 while the second write requires RSTGDP = 0x1D.

3.2.6.3 Reset Gate Read (RDC_SEMAPHOREx_RSTGT_R)

Although the intent of the hardware gate implementation specifies a protocol where the locking processor must unlock the gate, it is recognized that system operation may require a reset function to re-initialize the state of any gate(s) without requiring a system-level reset.

To support this special gate reset requirement, the RDC Semaphores module implements a "secure" reset mechanism that allows a hardware gate (or all the gates) to be initialized by following a specific dual-write access pattern. Using a technique similar to that required for the servicing of a software watchdog timer, the secure gate reset requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the specified gate(s). The required access pattern is:

1. A processor performs a 16-bit write to the RDC_SEMA42RSTGT memory location. The least significant byte (RDC_SEMA42RSTGT[RSTGDP]) must be 0xE2; the most significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the RDC_SEMA42RSTGT location. For this write, the lower byte (RDC_SEMA42RSTGT[RSTGDP]) is the logical complement of the first data pattern (0x1D) and the upper byte (RDC_SEMA42RSTGT[RSTGTN]) specifies the gate(s) to be reset. This gate field can specify a single gate be cleared, or else that all gates are to be cleared. If the same processor writes incorrect data on the second access or another processor performs the second write access, the special gate reset sequence is aborted and no error signal will be asserted.
3. Reads of the RDC_SEMA42RSTGT location return information on the 2-bit state machine (RDC_SEMA42RSTGT[RSTGSM]) that implements this function, the bus master performing the reset (RDC_SEMA42RSTGT[RSTGMS]), and the gate number(s) last cleared (RDC_SEMA42RSTGT[RSTGTN]). Reads of the RDC_SEMA42RSTGT register do not affect the secure reset finite state machine in any manner.

Address: Base address + 42h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	RSTGTN								0		RSTGSM		RSTGMS			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDC_SEMAPHOREx_RSTGT_R field descriptions

Field	Description
15–8 RSTGTN	Reset Gate Number. This 8-bit field specifies the specific hardware gate to be reset. This field is updated by the second write. This field contains the hexidecimal value of the gate number. If RSTGTN < 64, then reset the single gate defined by RSTGTN, else reset all the gates.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 RSTGSM	Reset Gate Finite State Machine. Reads of the RDC_SEMA42RSTGT register return the encoded state machine value. Note the RSTGSM = 10 state is valid for only a single machine cycle, so it is impossible for a read to return this value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as: 00 Idle, waiting for the first data pattern write. 01 Waiting for the second data pattern write. 10 The 2-write sequence has completed. Generate the specified gate reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. The "01" state persists for only one clock cycle. Software will never be able to observe this state. 11 This state encoding is never used and therefore reserved.

Table continues on the next page...

RDC_SEMAPHORE*x*_RSTGT_R field descriptions (continued)

Field	Description
RSTGMS	Reset Gate Bus Master. This 4-bit read-only field records the logical number of the bus master performing the gate reset function. The reset function requires that the two consecutive writes to this register must be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device, and the logical processor number is SoC-specific.

Chapter 4
Arm Platform and Debug

4.1 Arm Cortex A53 Platform (A53)

4.1.1 Overview

The Cortex-A53 cluster is a mid-range, low-power processor that implements the ARMv8-A architecture. The Cortex-A53 cluster has four cores, each with an L1 memory system and a single shared L2 cache that has a set of additional functions, which are included in a single APR region.

The core supports debug through real-time trace via ETM, and static debug via JTAG. The core platform supports static debug through the debug logic to SoC. This includes the capability of real-time trace via ARM's CoreSight ETM modules. The CTI and CTM modules allow cross-triggering of internal and external trigger sources.

4.1.1.1 Block Diagram

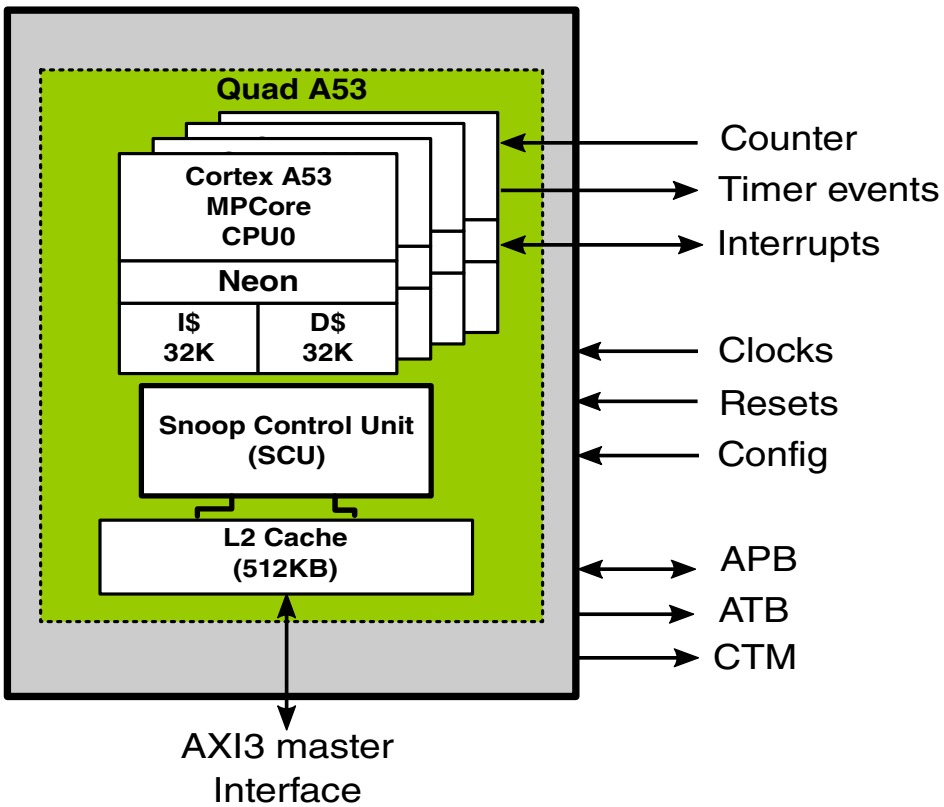


Figure 4-1. Cortex-A53 Block Diagram

4.1.1.2 Features

- 4x cores
- L1 instruction cache 32K
- L1 data cache 32K with SECDED
- Advanced SIMD (NEON) per core
- Crypto extension per core
- AMBA AXI3 interface
- No ACP is included
- utilizes the ARMv8 debug map
- 512KB of L2 Cache
  - 512KB with Single-bit Error Correct and Double-bit Error Detect (SECDED)
  - Input latency 2 cycles
  - Output latency 2 cycles
  - SCU-L2 cache protection

The configuration of the bus, cores and memory are detailed in the sections below. The core revision is CORTEXA53-r0p4-52rel0.

4.1.2 Functional Description

4.1.2.1 Platform sub-blocks

The sections below discuss the high-level overview of the ARM® Cortex®-A53 Platform components.

4.1.2.1.1 ARM Cortex-A53 MPCore Processor

The information presented in this section focuses on the design aspects of the ARM Cortex-A53 MPCore in the AP subsystem. The Cortex-A53 processor is a mid-range, low-power processor that implements the ARMv8-A architecture. The A53 complex has four cores, each with an L1 memory system and a single shared L2 cache.

The ARM Cortex-A53 MPCore processor consists of:

- Tightly-coupled L2 cache and an integrated Snoop Control Unit (SCU), connecting the four cores within the cluster providing cluster memory coherency, and a configurable coherent external interface supporting AMBA4 bus architecture.
- The Cortex-A53 implements the ARM Generic Interrupt Controller v3 (GICv3) architecture.

4.1.2.1.2 Advanced SIMD (Neon)

The Cortex-A53 processor supports the Advanced SIMD and Scalar Floating-point instructions in the A64 instruction set, and the Advanced SIMD and VFP instructions in the A32 and T32 instruction sets.

The ARMv8 architecture eliminates the concept of version numbers for Advanced SIMD and Floating-point in the AArch64 execution state.

4.1.2.1.3 Generic Interrupt Controller (GIC)

The Generic Interrupt Controller (GIC) supports and manages interrupts in the cluster. The GIC provides registers for managing:

- Interrupt sources
- Interrupt behavior
- Interrupt routing to one or more cores

The Cortex-A53 processor implements the GIC CPU interface as described in the Generic Interrupt Controller (GICv4) architecture. This interfaces with an external GICv3 or GICv4 interrupt distributor component within the system. The GICv4 architecture supports:

- Two security states
- Interrupt virtualization
- Software-generated Interrupts (SGIs)
- Message Based Interrupts
- System register access
- Memory-mapped register access
- Interrupt masking and prioritization
- Cluster environments
- Wake-up events in power management environments

The GIC includes interrupt grouping functionality that supports:

- Signaling interrupt groups to the target core using either the IRQ or the FIQ exception request, based on software configuration
- A unified scheme for handling the priority of Group 0 and Group 1 interrupts.

NOTE

GIC-500 supports limited backwards compatibility with GICv2. When GIC-500 is in backwards compatibility mode, only Cortex-A53 processors can access registers in GIC region, while Cortex-M7 and other masters in the system, including the Debug Access Port (DAP), can not. Cortex-A53 processors, Cortex-M7 and DAP can access GIC region when GIC-500 is not in backwards compatibility mode.

4.1.2.1.4 L1 Cache

The L1 instruction memory system has the following features:

- 32KB Instruction Cache
  - Instruction side cache line length of 64 bytes
  - 2-way set associative L1 Instruction cache
  - 128-bit read interface to the L2 memory system
- 32KB Data Cache
  - Data side cache line length of 64 bytes
  - 4-way set associative L1 Data cache
  - 256-bit write interface to the L2 memory system
  - 128-bit read interface to the L2 memory system
  - Read buffer that services the Data Cache Unit (DCU), the Instruction Fetch Unit (IFU) and the TLB
  - 64-bit read path from the data L1 memory system to the datapath

- 128-bit write path from the datapath to the L1 memory system
- Support for three outstanding data cache misses
- Merging store buffer capability. This handles writes to:
  - Device memory
  - Normal Cacheable memory
  - Normal Non-cacheable memory
  - Data side prefetch engine

4.1.2.1.5 L2 Cache

The L2 cache consists of an integrated Snoop Control Unit (SCU), connecting four cores within the A53 cluster. The SCU also has duplicate copies of the L1 Data cache tags for coherency support. The L2 memory system interfaces to the external memory system through an AMBA 128-bit bus. The tightly-coupled L2 cache includes the following features:

- 1MB shared cache size
- AMBA AXI3 interface
- Fixed line length of 64 bytes
- Physically indexed and tagged cache
- 16-way set-associative cache structure
- No ACP support
- ECC/parity support

4.1.2.2 Power

There are several power states supported by the A53 Core Complex. Supported primary states are listed below:

- **Run** – At least one of 4 cores is running. The other cores may either be running, clock gated, or powered down.
- **L2 only coherent** – The L2 is powered up and servicing snoops. The cores are stopped (either powered down or clock gated). In this state the cache is retained coherent to the system.
- **L2 only non-coherent** – Both cores are stopped and powered down, the logic in the L2 controller are retaining the arrays only. In this state the L2 is not coherent, and as a result, the other AP cores must also be stopped.
- **Cluster Off** – The L2 has been flushed from the L2 only coherent state, using the HW flush mechanism (no core required). Then the cluster is fully powered off including the L2 arrays.

The power supply for the cluster is separated into two regions. The first is the control domain and second is for the remainder of the cluster (core and cluster domains). These regions are listed below:

- **Control domain** – The AON control domain contains the controls for powering up and down the rest of the core. The control domain is always powered on first and powered off last.
- **Core domains** – The core domains contains the whole core. The supply for the core domains is the same as the cluster domain, but require separate power down switches.
- **Cluster domain** – The cluster domain contains the rest of the cluster outside of the cores, which includes the shared logic of the L2 memory. The supply for the cluster power domain is the same as the core domains, but does not require power switches as it's shut down externally.

Isolation cells are required between each of these domains and is controlled by the signals on the control domain. These cells are necessary to force the output signal to be isolated to null values when the local power shuts down.

4.1.2.2.1 Power States

4.1.2.2.1.1 Cluster power up sequence

This chapter describes Cortex A53 Cluster power Up Sequence where the SCU is controlling it. WFI wake-up events is received by the SCU through the steer controller sitting in the DBLOG subsystem.

1. For each core in the Cluster nCPUPORESET is asserted LOW
2. Then the nL2RESET is asserted LOW and L2RSTDISABLE is hold LOW
3. The switches are forced open during voltage ramp-up
4. SCU turns on power regulator supplying Cortex-A53 cluster
5. SCU waits for power being stabilized
6. Switches are controlled by SCU through low and high fan-out chains
7. SCU reloads repair configuration of memories used for L2 Cache
8. SCU sets BIAS value according to the targeted operating point of the cluster
9. SCU locks the PLL at targeted frequency
10. SCU performs SSI-ACE power-up sequence
11. SCU disables isolation cells of the cluster
12. By SCU, ACINACTM is asserted to 0 which indicated that snoop interface is active and participating to coherency

13. SCU starts the lead core. Other cores are started upon request

4.1.2.2.1.2 Individual core power up sequence

The Power up sequence is controlled by SCU and the following are the steps:

1. SCU shall check internally for identifying which core shall be woken-up.
2. SCU shall check nCPUPORESET of the core is LOW
3. SCU turns on power domain of the core
   a. Turn on the logic
   b. Turn on the memories
4. SCU reloads repair configuration of memories used for caches
5. SCU sets BIAS value to according to operating point of the cluster
6. SCU disables isolation cells of the core
7. SCU sets reset nCPUPORESET HIGH of the core
8. CPU starts-up. It has to set the CPUECTLR.SMPEN bit to 1 to enable snooping into the core
9. SCU controls the assertion of DBGPWRDUP to HIGH to allow external debug access to the core
10. If required use software to restore the state of the core as it was prior to power-down

4.1.2.3 Clocking

The A53 platform is provided a main processor clock that supplies the component clocks to the cluster components, including CoreSight debug components. The maximum frequency targets are specified in the chip datasheet.

The cores are intended to support up to 1.8 GHz dependent on forward bias within the operating temperature range. Please see the datasheet for more information. The clocks are described in the table below:

Table 4-1. A53 Clocks

Clock Signal	Clock Name	Frequency	Description
Main Clock	CLKIN	Target	Main input clock.
APB Clock	PCLKENDBG	CLKIN/4	APB clock controls the timing on the debug port. Fixed frequency ratio to main frequency.
AXI3 Bus Clock	ACLKENM	CLKIN/2	AXI3 Interface bus clock.
ATB Clock	ATCLKEN	CLKIN/4	ATB clock provides the clocks or outgoing trace. This clock needs to be reasonably high to enable sending samples out, but also needs to be the same as CNTCLKEN. Link to same signal as CNTCLKEN (1:4 core frequency fixed ratio).

Table continues on the next page...

Table 4-1. A53 Clocks (continued)

Input Counter Clock	CNTCLKEN	CLKIN/4	Input counter clock. Timing is identical to debug port. The frequency is a fixed ratio (1:4) with the core clock. The same signal may be used for both inputs.

4.2 Arm Cortex M7 Platform (CM7)

4.2.1 Overview

The Cortex-M7 platform features a single Arm®Cortex®-M7 processor in this chip. The Arm®Cortex®-M7 processor is a highly efficient, high-performance, embedded processor that features low interrupt latency, low-cost debug, and has backwards compatibility with existing Cortex-M profile processors. The processor has an in-order super-scalar pipeline by which many instructions can be dual-issued, including load/load and load/store instruction pairs because of multiple memory interfaces.

Memory interfaces that the processor supports include:
- Tightly-Coupled Memory (TCM)
- Harvard instruction and data caches and AXI master (AXIM) interface

The Arm Cortex-M7 Platform supports the following:
- 32 KB L1 Instruction Cache
- 32 KB L1 Data Cache
- Floating Point Unit (FPU) with support for the FPv5 architecture
- Internal Trace (TRC)

The number of IRQs supported for this chip is 160. In addition, it supports various components composing the Arm CoreSight debug/Trace system, such as ETM and CTI.

NOTE

This chip supports up to 16 interrupt priority levels, i.e. it implements bits [7:4] of each NVIC Interrupt Priority Register.

4.2.1.1 Block Diagram

A block diagram for the Cortex-M7 is given below:

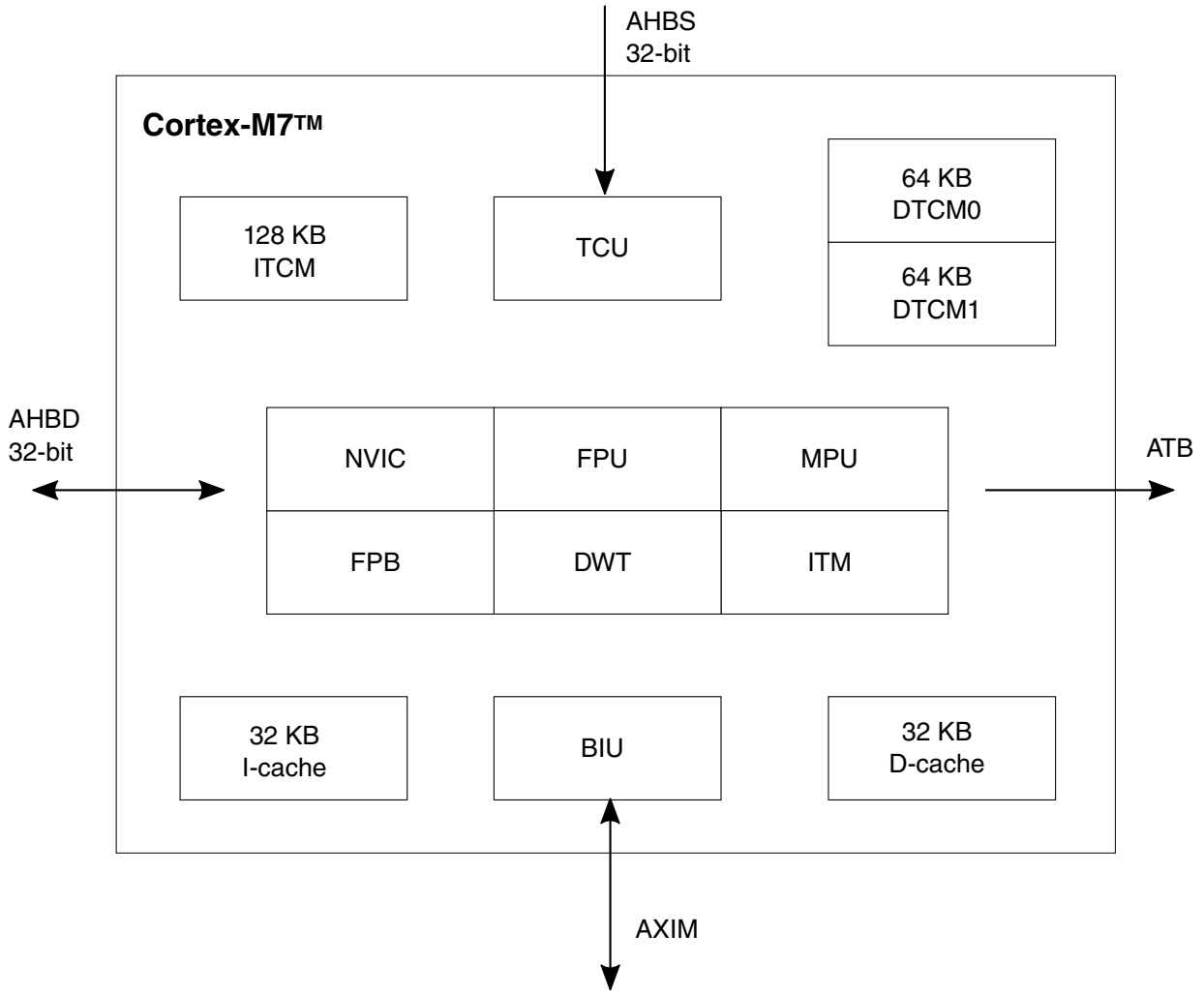


Figure 4-2. Cortex-M7 Platform Block Diagram

4.2.1.2 Features

The Cortex-M7 platform in this chip has the following features.

- 1× Cortex M7 processor: architecture Arm v7E-M
- floating point unit (FPU) FPv5 including single and double precision
- bus interface 64-bit AMBA AXI3, 32-bit AMBA AHB slave port, AMBA AHB debug interface for CoreSight components
- 32 KB instruction / 32 KB data L1 caches
- 128 KB ITCM / 128 KB DTCM
- 16 regions of MPU
- non-maskable (NMI) + 160 IRQs
- 16 interrupt priority levels (4-bits)
- sleep (WAIT) and deep sleep (STOP) power modes
- full debug with 8 breakpoints and 4 watchpoints

4.2.2 Functional Description

4.2.2.1 Clocks

Table 4-2. Arm Cortex-M7 Clocks

Clock Name	Description
ARM_M7_CLK_ROOT	M7 core clock

4.3 Messaging Unit (MU)

4.3.1 Overview

The Messaging Unit (MU) module enables two processors within the SoC to communicate and coordinate by passing messages (e.g. data, status and control) through the MU interface. The MU also provides the ability for one processor to signal the other processor using interrupts.

Because the MU manages the messaging between processors, the MU uses different clocks (from each side of the different peripheral buses). Therefore, the MU must synchronize the accesses from one side to the other. The MU accomplishes synchronization using two sets of matching registers (Processor A-facing, Processor B-facing).

4.3.1.1 Block Diagram

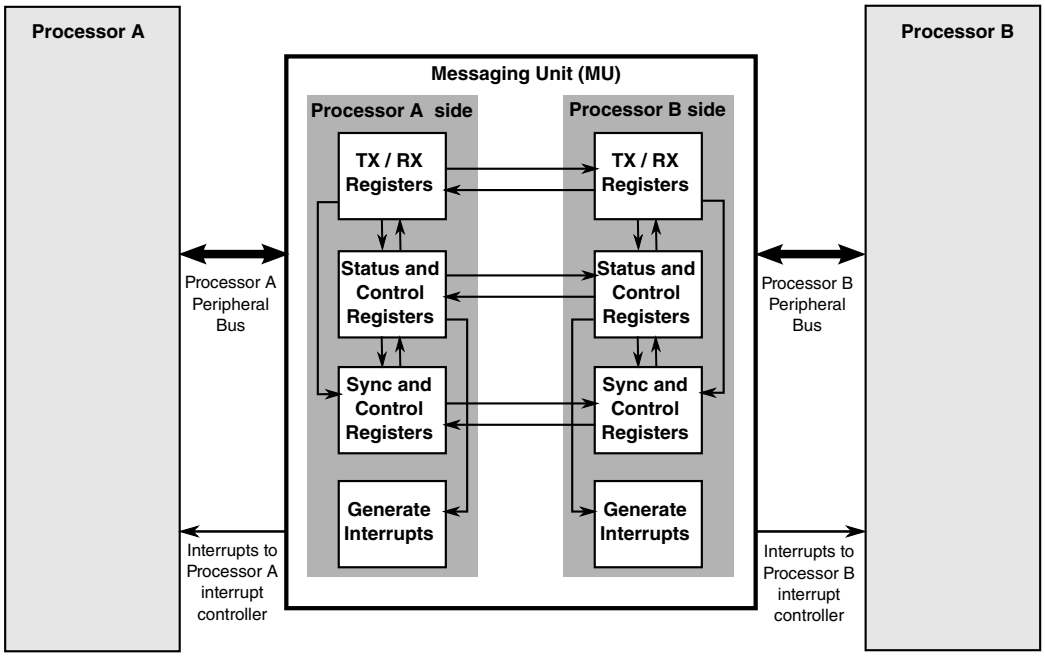


Figure 4-3. MU Block Diagram

4.3.1.2 Features

The MU includes the following features:

- Messaging control by interrupts or by polling
- Symmetrical processor interfaces with each side supporting the following:
  - Four general-purpose interrupt requests reflected to the other side
  - Three general-purpose flags reflected to the other side [1]
  - Four receive registers with maskable interrupt
  - Four transmit registers with maskable interrupt
- The above interrupts can also be used for waking up the other processor from low-power mode.

NOTE

MU chapter references to Processor A correspond to Core Complex 0 and references to Processor B correspond to Core Complex 1 in this device.

1. For example, MUA (Processor A) can set the flag to MUB (Processor B), and MUB(Processor A) can set the flag to MUA(Processor A).

4.3.2 Functional Description

Table 4-3. Major Features of the MU

Major Feature	Description
Interprocessor Interrupts	• The MU has 12 interrupt sources on each side (Processor A-side, Processor B-side) that are used for signaling the other processor. The interrupts can be used for notification of RX/TX events and general-purpose signaling between the processors.
MU Reset	• The Processor A can issue a reset to the entire MU, using a control bit (MUR) in the Processor A Control Register (ACR). • The MUR bit is a self-clearing bit.
Status and Control Communications between Core Complexes	• The MU provides a way for the two core complexes to communicate using the status and control registers present on both the Processor B and Processor A sides of the MU. • The status register of one MU side reflects the status of the other MU side. • The control register is used for control operations, such as enabling an interrupt and sending an interrupt to the other processor.
Synchronized Message Transfers between Core Complexes	• The transfer of data messages between core complexes uses transmit empty and receive full flags provided on both sides of the MU. • The update of these transmit and receive flags is accomplished using a synchronization mechanism. There is inherent latency between updating the flag on one side and reflecting its status on other side. For more about latency, see Event Update Timing
Accessing Shared Memory Directly and Avoiding Collisions	• For sending data or messages from one MU-side to the other MU-side, the MU provides 4 transmit registers and 4 receive registers on each side of the MU. • The Processor A or Processor B can access shared memory resources of the SoC directly. However, to avoid simultaneous access to shared memory by both core complexes, the MU provides a method (to prevent simultaneous access) using interrupts and transmit-receive registers for both processors. See Messaging Examples for more information.
Support for Different Clocks in the Two Core Complexes	• The heart of the MU module is the event control mechanism, which synchronizes the access of one MU-side to the other MU-side, because these two MU-sides can operate using different clocks. • Formulated event update latency.
Memory-Mapped Registers	• The MU is connected as a peripheral under the Peripheral bus on both sides—on the Processor A-side, the Processor A Peripheral Bus, and on the Processor B-side, the Processor B Peripheral Bus.

4.3.2.1 Operating Modes

The MU operates in one of two modes: Run or low power.

4.3.2.2 Low Power Modes

This section describes the low power operating modes of the MU module.

4.3.2.2.1 Low Power Clocks and Synchronization

The Processor B and the Processor A clocks operate at different frequencies and from different sources. The MU design does not assume any frequency relationship between the Processor A and the Processor B clocks. Be aware, however, that the frequency relationship affects the MU's throughput performance.

The data buffers and control logic of each MU-side operate with its corresponding clock.

4.3.2.2.2 Processor Low Power Modes

The Processors have four power modes:

- Run
- WAIT
- STOP
- SUSPEND, can also be called DSM (Deep Sleep Mode)

The Processor can be awakened from a low-power mode by any enabled Processor side MU interrupt, as reflected in the xSR "status" register (RF0–3, TE0–3, GIP0–3 bits are set) and enabled in the xCR control register. Using these bits, the Processor can actively control when to wake the other Processor.

While the Processor is in STOP mode (such that the xSR register bits cannot be updated with events), special logic drives the enabled MU interrupts directly from the other Processor-side (instead of from the xSR register).

While the Processor is in STOP mode, the asynchronous MU interrupt will be asserted to wake the Processor:

- If any transmit data register of the transmitter Processor-side is full, because of a write to it (transmit data register); that is, its "empty" bit in the xSR register is cleared while its corresponding receive interrupt is enabled on the receiver Processor-side.
- If any receive data register of the receiver Processor-side is empty, because of a read on the other Processor -side; that is, its "full" bit in the xSR register is cleared while its corresponding transmit interrupt is enabled on the transmitter Processor-side.
- If any general purpose interrupt bit is set in the xCR register on the other Processor-side while the corresponding interrupt is enabled on the Processor-side.

The logic enables the other Processor to operate independently while the Processor is in any power mode (including STOP). However, the Processor power mode change protocol should be handled with care regarding:

- The interrupts that are enabled on the Processor-side
- The events that could be triggered by the other Processor-side
- The compatibility with the other Processor protocol of entering STOP mode

If the Processor is in STOP mode and an event on the other Processor is triggered, the EP bit (in the xSR register) will remain high until the Processor wakes up.

Before entering STOP mode, the Processor programmer should verify that the EP bit (in the xSR register) is cleared. This check is needed to ensure that all pending updates from the Processor, including the power mode change when STOP or WAIT is executed, will be updated in the xSR register.

- If the other Processor is in STOP mode or DSM mode, the EP bit (in the xSR register) may be stuck high; in this case, the Processor need not check the EP bit before entering STOP mode.

4.3.2.3 Processor A Side Memory-Mapping

The messaging, control, and status registers of the Processor A-side for the MU are mapped to the Processor A memory as a regular peripheral. The Peripheral bus data is 32 bits wide inside the MU module.

4.3.2.4 Processor B Side Memory-Mapping

The messaging, control, and status registers of the Processor B-side for the MU are mapped to the Processor B memory as a regular peripheral. The Peripheral bus data is 32 bits wide inside the MU module.

4.3.2.5 MU Messaging

The MU provides 32-bit status and control registers to the Processor B and Processor A sides for control operations (such as interrupts and reset), and for status checking of the other MU-side.

For messaging, the MU has four, 32-bit write-only transmit registers and four, 32-bit read-only receive registers on the Processor B and Processor A-sides. These registers are used for sending messages to each other. These messages can also be controlled using the 3 general purpose flags provided in the control and status registers of either MU-side.

4.3.2.5.1 Programmer Model

The messaging logic is used in conjunction with external memory. There are various messaging methods, which can be used to implement a messaging protocol. Some of these messages can mean “a message of N words has just been written, starting at offset X in the memory,” or “a previous data block that was sent has just been read.” Having the

messaging logic independent from the memory array does not restrict users to a predefined hardware protocol. On the other hand, the software needed to manage the messaging is short and straightforward.

Most of the messaging mechanisms are symmetric; they are duplicated and are available on both the Processor B-side and the Processor A-side. The messaging mechanisms are:

- Four, 32-bit write-only transmit registers, which are each reflected in four, read-only receive registers in the other processor's side. Users can use these registers to transfer 32-bit word messages or frame information of messages written to the shared memory (number of words, initial address, and message type code).
- A write to a transmit register on the transmitter side clears a "transmitter empty" bit in the Status Register on the transmitter side, and sets a "receiver full" bit in the Status Register on the receiver side. The setting of the bit at the receiver side can optionally trigger an interrupt at the receiver side (maskable receive interrupt).
- A read of one of the receive registers at the receiver side clears the "receiver full" bit in the Status Register at the receiver side, and sets the "transmitter empty" bit in the Status Register on the transmitter side. The setting of the "transmitter empty" bit can optionally trigger an interrupt at the transmitter side (maskable transmit interrupt).
- Four general purpose flags are reflected in the Status Register on the receiver side
- A read/write access to any reserved location and a write to a read-only register on the Processor A-side of the MU will generate a module transfer error acknowledge to the Processor A.
- A read/write access to any reserved location and write to a read-only register on the Processor B-side of the MU will generate a module transfer error acknowledge to the Processor B.

4.3.2.5.2 Messaging Examples

The following are messaging examples:

- **Passing short messages:** Transmit register(s) can be used to pass short messages from one to four words in length. For example, when a four-word message is desired, only one of the registers needs to have its corresponding interrupt enable bit set at the receiver side; the message's first three words are written to the registers whose interrupt is masked, and the fourth word is written to the other register (which triggers an interrupt at the receiver side).
- **Passing frame information:** Transmit registers can be used to pass frame information for long messages written to the shared system memory. Such frame information normally includes a start address, number of words, and perhaps a message type code.

- **Passing event notices and requests:** Events and requests that do not include data words can be signaled from the Processor B to the Processor A using the general interrupts, such as acknowledging that a long message was read from the shared system memory.
- **Passing fixed length data:** Formatted data with a fixed length can be written in predetermined locations in the shared memory. A processor can use a general interrupt (Processor A or Processor B) to signal the other processor that the data is ready.
- **Passing announcements:** The three flags can be used by a processor to announce its current program state or other billboard messages to the other processor.

Figure 4-4 shows the MU registers schematic.

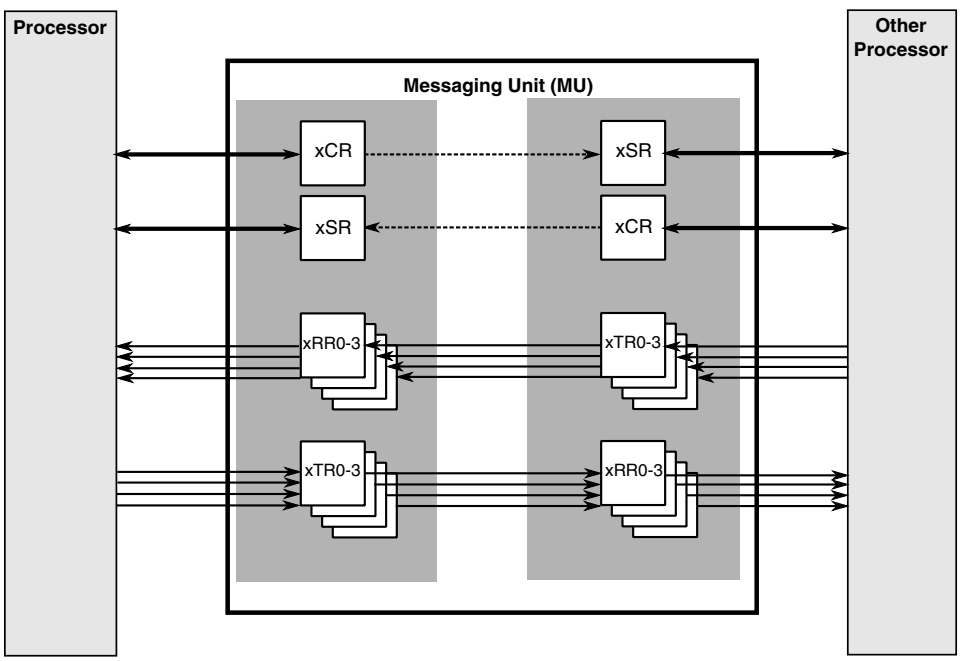


Figure 4-4. MU Registers

4.3.2.6 Event Update Timing

Each processor's MU messaging side (Processor B or Processor A) has a hardware mechanism to send "event update requests" to the other processor's side. An "event" is considered when any information change should be reflected at the Status Register of the receiving processor. The event update latency is the delay between the event being ready at one processor and the resulting update at the Status Register of the other processor.

- The minimum event latency is “1 clock of the sending side” + “2 1/2 clocks of the receiving side”. The minimum case is if there is no event pending when the new event occurs. See clocking chapter for this device for more information about clock sources.
- The maximum event latency is “6 clocks of the sending side” + “6 1/2 clocks of the receiving side.” The maximum case is if the event occurred just after a previous event was sent to the other side. The event update latency will vary between the above-mentioned minimum and maximum latencies, depending on the time at which the subsequent event is triggered.

4.3.2.7 Reset

The MU has two sources of reset, and each reset has a different function from the MU or system perspective.

- One asynchronous system that is connected to both sides of the MU interface.
- One programmable hardware reset (MUR bit) in the ACR register (on the Processor A-side).

Table 4-4. MU programmable resets

Reset	Description
Processor A MU reset	• Processor A MU Reset bit (MUR) of the ACR register • The MUR reset affects the messaging section on both the Processor A and the Processor B sides. The MUR reset causes all control and status registers to return to their default values and all internal states to be cleared. • It is up to the Processor A software to decide whether to use the MUR reset or not. • The instruction immediately following assertion of the MUR bit should not write to MU registers. Such a write may be overwritten by the reset sequence and the register will remain with the reset value. Wait at least one instruction (after assertion of the MUR bit) before attempting a write to MU registers.

After issuing MUR bit reset events, the Processor A programmer can verify that the reset sequence on the Processor B-side has ended, by checking the RS bit in the ASR register.

NOTE

MUR bit assertion is a delicate operation because it affects the other side’s registers asynchronously. MUR bit assertion may cause unpredictable behavior if, for example, the Processor B is concurrently testing an MU register bit (TE bit in Processor B SR register). Before asserting the MUR bit, it should be verified that the Processor B is not presently engaged in an MU signalling activity.

4.3.2.8 Interrupts

The MU controls the Processor B interrupt requests to the Processor A, and the Processor A interrupt requests to the Processor B. This section describes all the interrupts that the module generates.

4.3.2.8.1 Interrupts to the Processors

There are 12 interrupt sources from the MU to the Processors:

- Four receive interrupts for each of the receive registers: Asserted when the Processor receive full bits (xSR[RFn]) are set and the xCR[RIEn] bits enabled
- Four transmit interrupts for each of the transmit registers: Asserted when the Processor transmit empty bits (xSR[TEn]) are set and the xCR[TIEn] bits enabled
- Four general purpose interrupts: Asserted when the xSR[GIPn] bits are set and the xCR[GIEn] bits enabled

All the interrupts are maskable in the Processor Control Register (xCR). The MU does not assume any internal priority of these interrupts. Multiple interrupts (for example, Receive 0 and Receive 1 interrupts or any of the transmit and general purpose interrupts) can be asserted at one time. The priority of these interrupts should be resolved by the interrupt controller at the chip level.

The General Purpose Interrupt Pending bits (GIP0, GIP1, GIP2 and GIP3) should be cleared by the software (as part of the interrupt service routine) to de-assert the request to the interrupt controller.

4.3.2.8.2 General Purpose Interrupt Clearing Sequence

When a Processor writes to the general interrupt bit (GIR), the write event is synchronized to the other Processor clock to set the general interrupt request pending bit (GIP). When the GIP bit is set, and if the general purpose interrupt is enabled on the receiving Processor side (GIE bit is set), then the transmitting Processor general purpose interrupt is issued to the receiving Processor. The receiving Processor clears this interrupt by writing a “1” on the GIP bit. The interrupt is de-asserted as soon as the GIP bit is written. The write event of the GIP bit is synchronized to the other Processor clock. The synchronized signal clears the GIR bit. The software should not write the GIR bit again until the GIR bit is cleared.

4.3.2.9 Interrupt Messaging Protocols

4.3.2.9.1 Messaging Protocols using Interrupts

The example below describes a four-word messaging sequence sent by the Processor to the other Processor.

In this example, the first, second, and third receive interrupts are disabled, and the fourth receive interrupt is enabled. We write registers sequentially for n = 0, 1, 2, 3. For n = 0, 1, 2, the interrupts are disabled, therefore no interrupt will go to the other core complex (although interrupt conditions occur). For n = 3, the interrupt is enabled, and the last Receive Interrupt request is generated.

1. Write Sequence
   - The Processor writes the message information sequentially to its Transmit Registers 0, 1, 2.
   - When the write to the Transmit Register 3 occurs, the RF3 bit of the xSR is set after synchronization, and it immediately trigger the Receive 3 interrupt to the other Processor.
2. Read Sequence
   - The other Processor receives the Receive 3 interrupt and starts reading the message transferred from the receive registers.
   - After Receive Register 3 is read, the interrupt bit is cleared.

Figure 4-5 shows the programmer's model of a messaging protocol using transmit and receive registers. Use Table 4-5 and Figure 4-5 to understand the generalized protocol sequence.

Table 4-5. Interrupt Messaging Protocol (Generalized)

Sequence	Action	Description
1	Processor A Data write	A data write to the ATRn register by Processor A is immediately reflected in the Processor B BRRn register.
2	Clear Tx Empty bit and Set Rx Full bit	The data write to the ATRn register • Clears the transmitter empty bit (TEn) in the Processor A Transmit Status Register • Sets the receiver full bit (RFn) in the Processor B Receive Status Register
3	Generate Receive Interrupt request	The setting of the receiver full bit (RFn) in the Receive Status Register generates a Receive Interrupt request to Processor B.
4	Processor B Data read	After receiving the Receive Interrupt request, Processor B performs a data read of the BRRn register.
5	Clear Rx Full bit and Set Tx Empty bit	Reading the data out of the BRRn register • Clears the receiver full bit (RFn) in the Processor B Receive Status Register • Sets the transmitter empty bit (TEn) in the Processor A Transmit Status Register
6	Generate Transmit Interrupt request	The setting of the transmitter empty bit (TEn) in the Transmit Status Register generates a Transmit Interrupt request to Processor A.

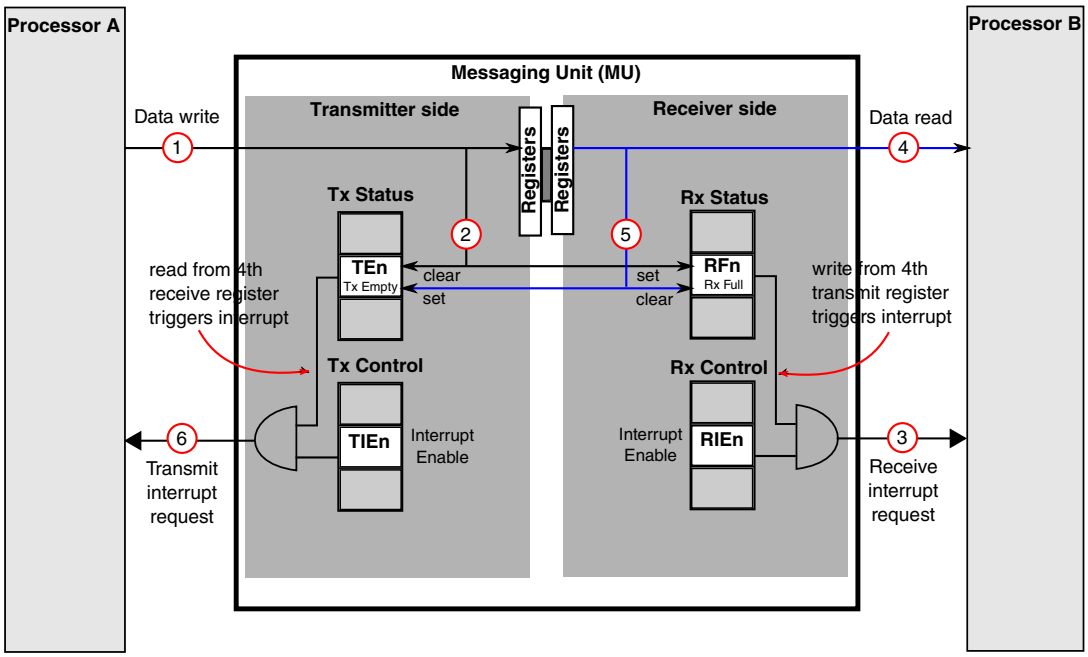


Figure 4-5. Messaging Model Using Transmit and Receive Registers

NOTE

The Transmit registers can be used to pass frame information on long messages written to the shared memory. Such frame information would typically include an initial address, number of words, and perhaps a message type code.

The messaging hardware can be used by software to implement messaging protocols for a wide array of message types. Full support is given for both interrupt and polling management schemes.

4.3.2.9.2 Messaging Protocols using Event Interrupts

Events and requests that do not include data words can be signaled from the Processor B to the Processor A using the two general interrupts.

Formatted data with a fixed length can be written in predetermined locations in the shared memory. A processor can use a general purpose interrupt to signal the other processor that the data is ready.

The three flags can be used by a processor to announce to the other processor the program state it is currently in, or to announce similar messages.

Table 4-6 and Figure 4-6 describe the event sequence when the Processor triggers an interrupt.

Table 4-6. Interrupt Messaging Protocol (Generalized)

Sequence	Action	Description
1	Processor A sets General Interrupt request bit	Processor A sets its associated General Interrupt request bit (GIRn = 1) in the control register (ACR).
2	General Interrupt Request Pending status bit is set	The General Interrupt Request Pending status bit (GIPn) in the status register (BSR) is set to "1"
3	General Interrupt request to Processor B is generated	Setting the GIPn bit generates the General Interrupt request to Processor B (Interrupt Request Enable bit, GIEn, must be set for Processor B)
4	Processor B reads status register	The Processor B reads the GIPn bit in the BSR register.
5	Processor B services the interrupt	-
6	Processor B sets GIPn bit to clear interrupt	The Processor B writes "1" to the corresponding GIPn bit to clear the interrupt
7	GIRn bit is cleared	Setting the GIPn bit to "1" clears the General Interrupt request bit (GIRn) in the Processor A control register (ACR).

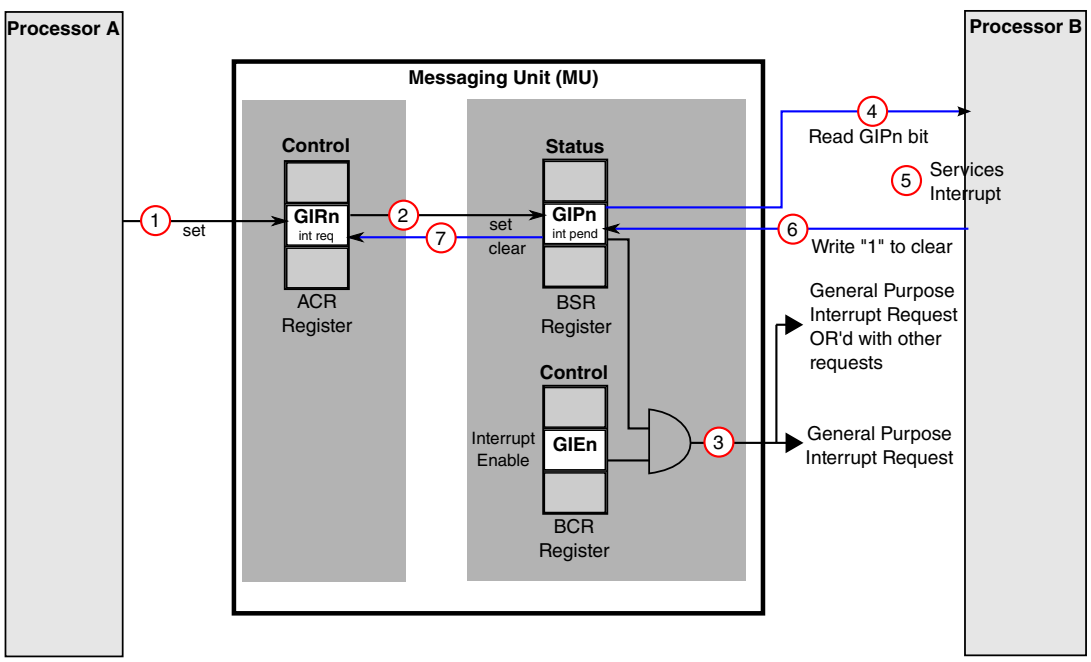


Figure 4-6. Messaging Model Using a General Purpose Interrupt

4.3.2.10 Exclusive Access to Shared Memory

The MU can be used to signal one processor about its current access to the shared memory, so that the data is not overwritten by the other processor during the exclusive memory access period.

The following tables describe the signaling protocol that the Processor A uses to inform the Processor B about its current access (write) to the shared memory, assuming that the set of bits and registers (GIR0 bit, BRR0 register, BTR0 register, GIR0 bit, ARR0 register, ATR0 register) are reserved to support exclusive access to the shared memory protocol.

Table 4-7. How the Processor A Performs an Exclusive Access to Shared Memory

Sequence	Action	Description
1	Processor A sends GIRn request to Processor B using Processor A control register	When the Processor A wants to perform an exclusive access to the shared memory, the Processor A sends an GIR0 request to the Processor B.
2	Processor A sends an exclusive-access request using a transmit data register (ATRn)	The Processor A will send an exclusive-access request (command, location, and length of target access) to Processor B using a selected transmit data register (ATR0).
3	Processor A waits for a dedicated interrupt from Processor B	The Processor A waits for a dedicated interrupt (as an acknowledgement) triggered by the Processor B before proceeding.
4	Processor A accesses shared memory	After receiving a dedicated interrupt from the Processor B, Processor A proceeds.

Table 4-8. How the Processor B Scans for Transaction Information

Sequence	Action	Description
1	Processor B receives an interrupt from a receive data register (BRRn)	-
2	Processor B reads the receive data register (BRRn)	-
3	Processor B scans the receive data register contents	For transaction information (whether Processor A has requested an exclusive-access)

Table 4-9. How the Processor B Accepts Exclusive Access by Processor A

Sequence	Action	Description
1	Processor B triggers a dedicated interrupt	Processor B acknowledges the Processor A request by triggering a dedicated interrupt (ack) to the Processor A.
2	Processor B sends a code message to Processor A	Along with the acknowledge interrupt, the Processor B sends a code message to the Processor A through the selected transmit register (BTRn). The message informs the Processor A that it can exclusively access the shared memory.

Table 4-10. How the Processor B Rejects Exclusive Access by Processor A

Sequence	Action	Description
1	Processor B ignores Processor A request for exclusive access	If the Processor B does not want to give go-ahead permission to the Processor A, Processor B ignores the exclusive access request.

4.3.2.11 Packet Data Transfers

The following example describes the packet transfer sequence between the Processor B and Processor A subsystems:

Table 4-11. Packet Data Transfer Sequence

Action	Sequence	Description
Processor B requests DMA	1	The Processor B sends a DMA request to initiate the packet data transfer
DMA data transfer	2	DMA acknowledges.
	3	DMA starts transferring data from the specified Processor B location to the specified shared memory
	4	DMA interrupts the Processor B to signal that the packet transfer has finished.
Processor B informs Processor A that data is in shared memory	5	Using an MU Processor B-side transmit register, the Processor B sends a packet information message to the Processor A to inform the Processor A of the arrival of new packet data that is stored in shared memory . The message contains the command, location, and length of packet data information.
Processor A receives interrupt	6	The Processor A receives an interrupt (assuming its corresponding Processor A MU-side receive interrupt is enabled), and the pending processing task becomes active and processes packet data from memory.
Processor A reads data, writes data	7	The Processor A reads or processes packet data from shared memory.
	8	The Processor A writes the result from packet processing to a separate buffer.
Processor A informs Processor B that transfer is finished	9	After the processing of the packet data finishes, the Processor A informs the Processor B (using the MU Processor A-side transmit register, ATRn).
Processor A sends interrupt to Processor B (request for more data)	10	The Processor B receives the next interrupt from the Processor A, in which the Processor A requests more packet data.

4.3.3 External Signals

There are no Messaging Unit signals directly available at the chip boundary of the SoC.

4.3.4 Initialization

The MU is used for two processors to communicate by passing messages. Below are examples for passing messages:

1. Interrupt mode
    - Processor A:

- write ATR0;
- ACR.TIEn[0] = 1;
- wait for transmit register empty interrupt;

- Processor B:
  - BCR.RIEn[0] = 1;
  - wait for receive register full interrupt;
  - read BRR0;

2. Polling mode
   - Processor A:
     - write ATR0;
     - while(ASR.TEn[0] != 1) ;
   - Processor B:
     - while(BSR.RFn[0] != 1) ;
     - read BRR0;

See Application Information for details regarding software restrictions.

4.3.5 Application Information

This section describes certain software restrictions when accessing the MU.

4.3.5.1 General Restrictions

This section lists the restrictions that apply to both the sides (Processor A, Processor B) of the MU.

4.3.5.1.1 Write-After-Write to a Transmit Register

A write to a transmit register signals the receiver side that data is ready for retrieval.

- Writing to the transmit register again without verifying that the data was retrieved is prohibited, because the transmitter side has no way of knowing the exact time that the receiver will attempt to retrieve the data.
- Before attempting to write the transmit register again, the transmitter side should wait for a “Transmitter Empty” interrupt, or should poll the “Transmitter Empty” bit in the Status Register.
- Failure to follow this restriction may result in the wrong data being read on the receiver side of the MU.

4.3.5.1.2 Read-After-Read from a Receive Register

A read of a receive register signals the transmitter side that data can be written to that register. In the same way, the receiver processor should not read a receive register before receiving a “Receiver Full” interrupt or polling the “Receiver Full” bit in the Status Register.

- Reading the receive register again without verifying that the data was written is prohibited, because the receiver side has no way of knowing the exact time that the transmitter will attempt to write the data.
- Before attempting to read the receive register again, the receiver side should wait for a “Receiver Full” interrupt, or should poll the “Receiver Full” bit in the Status Register.
- Failure to follow this restriction may result in the wrong data being written on the transmitter side of the MU.

4.3.5.2 Processor Restrictions

This section lists the restrictions that apply each side of the processor in the MU.

4.3.5.2.1 Before Entering Low Power Mode

Before entering Low Power mode, the Processor should verify that the Processor Event Pending (EP) bit in the Status Register is cleared.

- If the Event Pending bit (EP) is still set to “1”, then the Processor should wait and poll the EP bit until it is cleared, before executing the low power mode instruction.
- Note that if the other Processor is in Low Power mode, the EP bit may be stuck high. In this case, the other Processor clock must be turned ON to get the EP bit cleared before the Processor can enter Low Power mode.
- To discover which power mode the other Processor is in, the Processor can check the Power Mode bits in the xSR register.

4.3.5.2.2 Before Setting a General Interrupt Request Bit (GIR0–3)

Before setting a General Interrupt Request bit (GIR0–3), verify the GIRn bit is cleared, which means that a general interrupt is not pending. Generally, setting the GIRn bit while the bit is set to “1” will be ignored, but in some cases it may issue a second interrupt. This restriction is meant to prevent this indeterministic behavior.

4.3.5.2.3 Reset Bit Restrictions

The reset bit (MUR, BHR) restrictions are:

- Before asserting the MUR bit in the ACR register, verify that the Processor B-side is not engaged in some MU activity.
- Do not write to an MU register in the instruction immediately after the assertion of the MUR bit in the ACR register, because the written data can be overridden by the reset value.

4.3.6 MU Processor A-side Memory Map/Register Definition

4.3.6.1 MUA register descriptions

This section contains the detailed register descriptions for the Processor A-side MU registers.

4.3.6.1.1 MUA memory map

MUA base address: 30AA_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Processor A Transmit Register 0 (ATR0)	32	RW	0000_0000
4	Processor A Transmit Register 1 (ATR1)	32	RW	0000_0000
8	Processor A Transmit Register 2 (ATR2)	32	RW	0000_0000
C	Processor A Transmit Register 3 (ATR3)	32	RW	0000_0000
10	Processor A Receive Register 0 (ARR0)	32	RO	0000_0000
14	Processor A Receive Register 1 (ARR1)	32	RO	0000_0000
18	Processor A Receive Register 2 (ARR2)	32	RO	0000_0000
1C	Processor A Receive Register 3 (ARR3)	32	RO	0000_0000
20	Processor A Status Register (ASR)	32	RW	00F0_0080
24	Processor A Control Register (ACR)	32	RW	0000_0000

4.3.6.1.2 Processor A Transmit Register 0 (ATR0)

Use Processor A Transmit Register 0 (ATR0, 32-bit, write-only) to transmit a message or data to the Processor B.

- Users can only write to the ATR0 register when the TE0 bit in ASR register is set to "1".
- Reading the ATR0 register returns all zeros.

4.3.6.1.2.1 Offset

Register	Offset
ATR0	0h

4.3.6.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ATR0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATR0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.6.1.2.3 Fields

Field	Description
31-0 ATR0	ATR0 Processor A Transmit Register 0. (Write-only) • Data written to the ATR0 register is reflected on the Processor B-side in the Processor B Receive Register 0 (BRR0). The ATR0 and BRR0 registers are not double-buffered—a write to the ATR0 register overrides the data readable at the BRR0 register. • A write to the transmit register clears a "transmitter empty" bit (TE0) in the Processor A Status Register (ASR) on the transmitter side, and sets a "receiver full" bit (RF0) in the Processor B Status Register (BSR) on the receiver side (optionally triggering an interrupt 0 on the Processor B-side). • Any write to the ATR0 register will update all status information.

4.3.6.1.3 Processor A Transmit Register 1 (ATR1)

Use Processor A Transmit Register 1 (ATR1, 32-bit, write-only) to transmit a message or data to the Processor B.

- Users can only write to the ATR1 register when the TE1 bit in ASR register is set to "1".
- Reading the ATR1 register returns all zeros.

4.3.6.1.3.1 Offset

Register	Offset
ATR1	4h

4.3.6.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ATR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.6.1.3.3 Fields

Field	Description
31-0 ATR1	ATR1 Processor A Transmit Register 1. (Write-only) • Data written to the ATR1 register is reflected on the Processor B-side in the Processor B Receive Register 1 (BRR1). The ATR1 and BRR1 registers are not double-buffered—a write to the ATR1 register overrides the data readable at the BRR1 register. • A write to the transmit register clears a "transmitter empty" bit (TE1) in the Processor A Status Register (ASR) on the transmitter side, and sets a "receiver full" bit (RF1) in the Processor B Status Register (BSR) on the receiver side (optionally triggering an interrupt 1 on the Processor B-side). • Any write to the ATR1 register will update all status information.

4.3.6.1.4 Processor A Transmit Register 2 (ATR2)

Use Processor A Transmit Register 2 (ATR2, 32-bit, write-only) to transmit a message or data to the Processor B.

- Users can only write to the ATR2 register when the TE2 bit in ASR register is set to "1".
- Reading the ATR2 register returns all zeros.

4.3.6.1.4.1 Offset

Register	Offset
ATR2	8h

4.3.6.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ATR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.6.1.4.3 Fields

Field	Description
31-0 ATR2	ATR2 Processor A Transmit Register 2. (Write-only) • Data written to the ATR2 register is reflected on the Processor B-side in the Processor B Receive Register 2 (BRR2). The ATR2 and BRR2 registers are not double-buffered—a write to the ATR2 register overrides the data readable at the BRR2 register. • A write to the transmit register clears a "transmitter empty" bit (TE2) in the Processor A Status Register (ASR) on the transmitter side, and sets a "receiver full" bit (RF2) in the Processor B Status Register (BSR) on the receiver side (optionally triggering an interrupt 2 on the Processor B-side). • Any write to the ATR2 register will update all status information.

4.3.6.1.5 Processor A Transmit Register 3 (ATR3)

Use Processor A Transmit Register 3 (ATR3, 32-bit, write-only) to transmit a message or data to the Processor B.

- Users can only write to the ATR3 register when the TE3 bit in ASR register is set to "1".
- Reading the ATR3 register returns all zeros.

4.3.6.1.5.1 Offset

Register	Offset
ATR3	Ch

4.3.6.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ATR3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATR3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.6.1.5.3 Fields

Field	Description
31-0 ATR3	ATR3 Processor A Transmit Register 3. (Write-only) • Data written to the ATR3 register is reflected on the Processor B-side in the Processor B Receive Register 3 (BRR3). The ATR3 and BRR3 registers are not double-buffered—a write to the ATR3 register overrides the data readable at the BRR3 register. • A write to the transmit register clears a "transmitter empty" bit (TE3) in the Processor A Status Register (ASR) on the transmitter side, and sets a "receiver full" bit (RF3) in the Processor B Status Register (BSR) on the receiver side (optionally triggering an interrupt 3 on the Processor B-side). • Any write to the ATR3 register will update all status information.

4.3.6.1.6 Processor A Receive Register 0 (ARR0)

Use Processor A Receive Register 0 (ARR0, 32-bit, read-only) to receive a message or data from the Processor B.

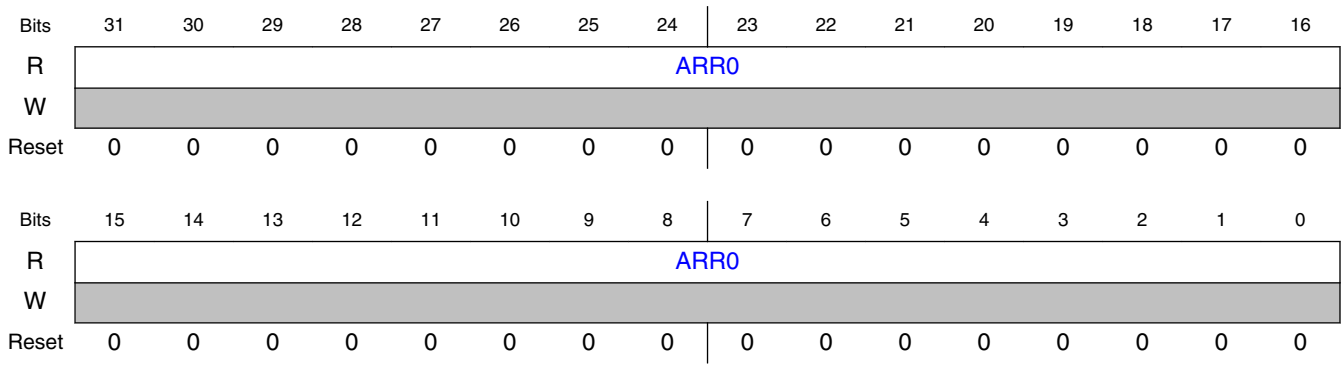
- Data written to the BTR0 register is immediately reflected in the ARR0 register.

- Users can only read the ARR0 register when the RF0 bit in the ASR register is set to "1".
- Writing to the ARR0 register generates an error response to the Processor A.

4.3.6.1.6.1 Offset

Register	Offset
ARR0	10h

4.3.6.1.6.2 Diagram



4.3.6.1.6.3 Fields

Field	Description
31-0 ARR0	ARR0 Processor A Receive Register 0. (Read-only) • Reflects the data written to Processor B Transmit Register 0 (BTR0). • Reading the ARR0 register clears the "receiver full" bit (RF0) in the Processor A Status Register (ASR) on the receiver side, and sets the "transmitter empty" bit (TE0) in the Processor B Status Register on the transmitter side (optionally triggering a transmit interrupt 0 on the Processor B-side). • Any read of the ARR0 register will update all status information.

4.3.6.1.7 Processor A Receive Register 1 (ARR1)

Use Processor A Receive Register 1 (ARR1, 32-bit, read-only) to receive a message or data from the Processor B.

- Data written to the BTR1 register is immediately reflected in the ARR1 register.

- Users can only read the ARR1 register when the RF1 bit in the ASR register is set to "1".
- Writing to the ARR1 register generates an error response to the Processor A.

4.3.6.1.7.1 Offset

Register	Offset
ARR1	14h

4.3.6.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ARR1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ARR1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.6.1.7.3 Fields

Field	Description
31-0 ARR1	ARR1 Processor A Receive Register 1. (Read-only) • Reflects the data written to Processor B Transmit Register 1 (BTR1). • Reading the ARR1 register clears the "receiver full" bit (RF1) in the Processor A Status Register (ASR) on the receiver side, and sets the "transmitter empty" bit (TE1) in the Processor B Status Register on the transmitter side (optionally triggering a transmit interrupt 1 on the Processor B-side). • Any read of the ARR1 register will update all status information.

4.3.6.1.8 Processor A Receive Register 2 (ARR2)

Use Processor A Receive Register 2 (ARR2, 32-bit, read-only) to receive a message or data from the Processor B.

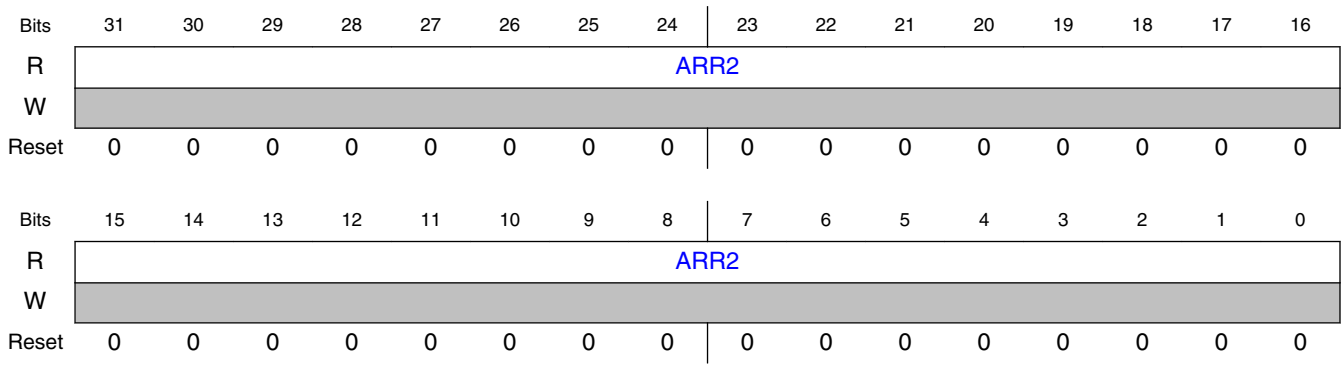
- Data written to the BTR2 register is immediately reflected in the ARR2 register.

- Users can only read the ARR2 register when the RF2 bit in the ASR register is set to "1".
- Writing to the ARR2 register generates an error response to the Processor A.

4.3.6.1.8.1 Offset

Register	Offset
ARR2	18h

4.3.6.1.8.2 Diagram



4.3.6.1.8.3 Fields

Field	Description
31-0 ARR2	ARR2 Processor A Receive Register 2. (Read-only) • Reflects the data written to Processor B Transmit Register 1 (BTR2). • Reading the ARR2 register clears the "receiver full" bit (RF2) in the Processor A Status Register (ASR) on the receiver side, and sets the "transmitter empty" bit (TE2) in the Processor B Status Register on the transmitter side (optionally triggering a transmit interrupt 2 on the Processor B-side). • Any read of the ARR2 register will update all status information.

4.3.6.1.9 Processor A Receive Register 3 (ARR3)

Use Processor A Receive Register 3 (ARR3, 32-bit, read-only) to receive a message or data from the Processor B.

- Data written to the BTR3 register is immediately reflected in the ARR3 register.

- Users can only read the ARR3 register when the RF3 bit in the ASR register is set to "1".
- Writing to the ARR3 register generates an error response to the Processor A.

4.3.6.1.9.1 Offset

Register	Offset
ARR3	1Ch

4.3.6.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ARR3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ARR3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.6.1.9.3 Fields

Field	Description
31-0 ARR3	ARR3 Processor A Receive Register 3. (Read-only) • Reflects the data written to Processor B Transmit Register 3 (BTR3). • Reading the ARR3 register clears the "receiver full" bit (RF3) in the Processor A Status Register (ASR) on the receiver side, and sets the "transmitter empty" bit (TE3) in the Processor B Status Register on the transmitter side (optionally triggering a transmit interrupt 3 on the Processor B-side). • Any read of the ARR3 register will update all status information.

4.3.6.1.10 Processor A Status Register (ASR)

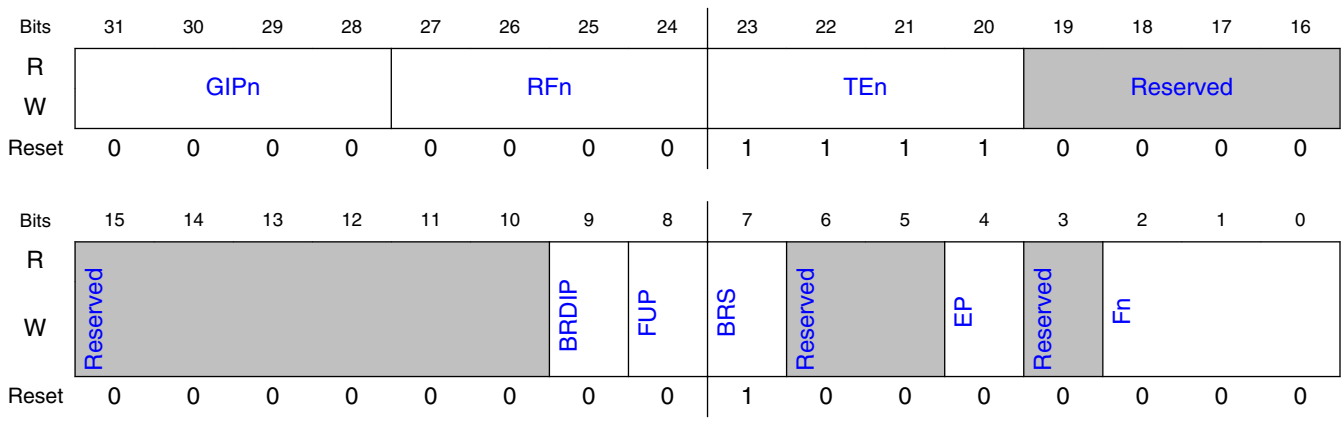
Use the Processor A Status Register (ASR, 32-bit, read-write) to show interrupt status from the Processor B, general purpose flags, and to set dual function control-status bits.

- Some dual-purpose bits are set by the MU logic, and cleared by the Processor A-side programmer
- Other dual-purpose bits are set by the Processor A-side programmer, and cleared by the MU logic.

4.3.6.1.10.1 Offset

Register	Offset
ASR	20h

4.3.6.1.10.2 Diagram



4.3.6.1.10.3 Fields

Field	Description
31-28 GIPn	GIPn For n = {0, 1, 2, 3} Processor A General Interrupt Request n Pending. (Read-Write) • GIPn bit signals the Processor A that the GIRn bit in the BCR register on the Processor B-side was set from "0" to "1". If the GIEn bit in the ACR register is set to "1", a General Interrupt n request is issued. • The GIPn bit is cleared by writing it back as "1". Writing "0", or writing "1" when the GIPn bit is cleared is ignored. Use this feature in the interrupt routine, where the GIPn bit is cleared in order to de-assert the interrupt request source at the interrupt controller. The proper bit clearing sequence is: clear an Processor A register, set the desired bit in it (Processor A register), and write it to the ASR register, thus clearing the GIPn bit. • GIPn bit is cleared when the MU is reset. 0000 - Processor A general purpose interrupt n is not pending. (default) 0001 - Processor A general purpose interrupt n is pending.
27-24 RFn	RFn For n = {0, 1, 2, 3} Processor A Receive Register n Full. (Read-only)

Table continues on the next page...

Field	Description
	• The RFn bit is set to "1" when the BTRn register is written on the Processor B-side. • After the RFn bit is set to "1", the RFn bit signals the Processor A-side that new data is ready to be read by the Processor A in the ARRn register, and a Receive n interrupt is issued on the Processor A-side (if the RIEn bit in the ACR register has been set to "1"). • RFn bit is cleared when the ARRn register is read, and when the MU is reset. 0000 - ARRn register is not full (default). 0001 - ARRn register has received data from BTRn register and is ready to be read by the Processor A.
23-20 TEn	TEn For n = {0, 1, 2, 3} Processor A Transmit Register n Empty. (Read-only) • The TEn bit is set to "1" after the BRRn register is read on the Processor B-side. • After the TEn bit is set to "1", the TEn bit signals the Processor A-side that the ATRn register is ready to be written on the Processor A-side, and a Transmit n interrupt is issued on the Processor A-side (if the TEn bit in the ACR register is set to "1"). • TEn bit is cleared after the ATRn register is written on the Processor A-side. • TEn bit is set to "1" when the MU is reset. 0000 - ATRn register is not empty. 0001 - ATRn register is empty (default).
19-10 —	- Reserved
9 BRDIP	BRDIP Processor B Reset De-asserted Interrupt Pending. (Read-Write) • BRDIP bit signals the Processor A-side that the Processor B-side has come out of reset. • BRDIP bit is set to "1" after the MU Processor B-side comes out of reset, after synchronization. The interrupt generated by a Processor B-side reset de-assertion is ORed with the Processor A general purpose interrupt 3. The Processor A general purpose interrupt 3 is issued when the Processor B-side comes out of reset (if the interrupt is enabled). • To clear the BRDIP bit, write "1", which also clears general purpose interrupt 3. • When Processor A-side of MU comes out of reset BRDIP bit has value "0"(default).Then Processor A sees the status of Processor B-side and if Processor B-side has come out of reset then BRDIP bit goes high. This takes 5-6 clock cycles. If the BRDIP bit is read now it shows as high although its reset value was "0". 0 - The Processor A general purpose interrupt 3, because of a Processor B-side reset de-assertion, is cleared (default). 1 - The Processor B-side is out of reset.
8 FUP	FUP Processor A Flags Update Pending. (Read-only) • FUP bit is set to "1" when the Processor A-side sends a Flags Update request to the Processor B-side. • A Flags Update request is generated when the ABF[2:0] bits of the ACR register change. No flag update changes are allowed while the FUP bit is set to "1". Any write to the ABF[2:0] bits, while the FUP bit is set to "1", will not generate a Flags Update event, and the ABF[2:0] bits will stay unchanged. • FUP bit is cleared when this Flags Update request is internally acknowledged (that the flag is updated) from the MU Processor B-side, and during MU reset. 0 - No flags updated, initiated by the Processor A, in progress (default) 1 - Processor A initiated flags update, processing
7	BRS

Table continues on the next page...

Field	Description
BRS	Processor B-side Reset State. (Read-only) • BRS bit indicates if the Processor B-side of the MU is in a reset state or not. • If the BRS bit is set to "1", then the Processor B-side of the MU is still in the reset state. • If the BRS bit is cleared, then the Processor B-side of the MU are out of reset. • The BRS bit is set to "1" during: a Processor B system reset, or an MU reset (caused by setting the MUR bit at the ACR register). • The BRS bit is cleared when the reset sequence on the Processor B-side of the MU ends. After issuing any of the reset events mentioned previously, verify that the BRS bit is cleared before starting any accesses. • When Processor A side of MU comes out of reset BRS bit has value "1"(default).Then Processor A sees the status of Processor B-side and if Processor B-side has come out of reset then BRS bit goes low. This takes 5-6 clock cycles. If the BRS bit is read now it shows as low although its reset value was "1" . 0 - The Processor B-side of the MU is not in reset. 1 - The Processor B-side of the MU is in reset.
6-5 —	- Reserved
4 EP	EP Processor A-Side Event Pending. (Read-only) • EP bit is set to "1" when the Processor A-side mechanism sends an event update request to the Processor B-side. • EP bit is cleared when the event update acknowledge is received. An "event" is any hardware message that is reflected in the BSR register on the Processor B-side (for example, "transmit register 0 written"). During normal operations, users do not have to deal with the state of the EP bit because the event update mechanism works automatically. • To ensure events have been posted to Processor B before entering STOP mode, verify that the EP bit is cleared. If EP bit is set to "1", wait and continue to poll it (EP bit) before entering STOP mode. • Reading the ASR register (to check the EP bit) should be the last access to the MU that should be performed before entering STOP or WAIT modes; otherwise, the EP bit may be set by subsequent additional actions. • The EP bit is cleared when the MU resets. 0 - The Processor A-side event is not pending (default). 1 - The Processor A-side event is pending.
3 —	- Reserved
2-0 Fn	Fn For n = {0, 1, 2} Processor A-Side Flag n. (Read-only) • Fn bit is the Processor A-side flag that reflects the values written to the BAFn bit in the Processor B control register. • Every time that the BAFn bit is written, the BAFn bit write event updates the Fn bit after the event update latency, which is measured in terms of the number of clocks of the Processor B and the Processor A. 000 - BAFn bit in BCR register is written 0 (default). 001 - BAFn bit in BCR register is written 1.

4.3.6.1.11 Processor A Control Register (ACR)

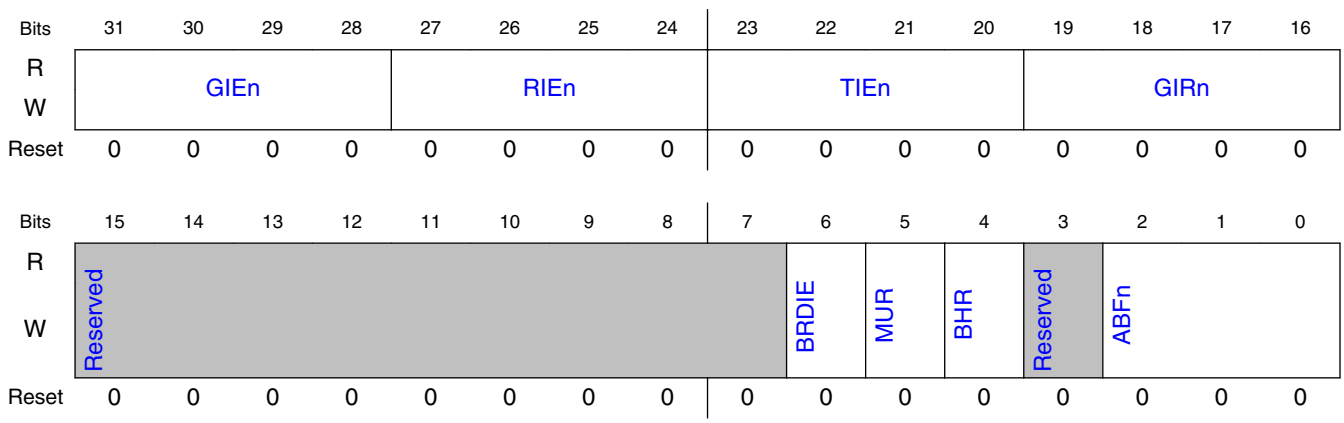
Use the Processor A Control Register (ACR, 32-bit, read-write) to enable the MU interrupts on the Processor A-side, and trigger events and interrupts on the Processor B-side (general purpose interrupt, flag update).

For the fields GIEn, RIEn, TIEn and GIRn, n=0 corresponds to the high order bit and n=3 corresponds to the low order bit.

4.3.6.1.11.1 Offset

Register	Offset
ACR	24h

4.3.6.1.11.2 Diagram



4.3.6.1.11.3 Fields

Field	Description
31-28 GIEn	GIEn For n = {0, 1, 2, 3} Processor A General Purpose Interrupt Enable n. (Read-Write) When GIEn=0 corresponds to the high order bit and GIE3 corresponds to the low order bit. • GIEn bit enables Processor A General Interrupt n. • If GIEn bit is set to "1" (enabled), then a General Interrupt n request is issued when the GIPn bit in the ASR register is set to "1". • If GIEn is cleared (disabled), then the value of the GIPn bit is ignored and no General Interrupt n request will be issued. • GIEn bit is cleared when the MU resets. 0000 - Disables Processor A General Interrupt n. (default)

Table continues on the next page...

Field	Description
	0001 - Enables Processor A General Interrupt n.
27-24 RIEn	RIEn For n = {0, 1, 2, 3} Processor A Receive Interrupt Enable n. (Read-Write) • RIEn bit enables Processor A Receive Interrupt n. • If RIEn bit is set to "1" (enabled), then an Processor A Receive Interrupt n request is issued when the RFn bit in the ASR register is set to "1". • If RIEn bit is cleared (disabled), then the value of the RFn bit is ignored and no Processor A Receive Interrupt n request will be issued. • RIEn bit is cleared when the MU resets. 0000 - Disables Processor A Receive Interrupt n. (default) 0001 - Enables Processor A Receive Interrupt n.
23-20 TIEn	TIEn For n = {0, 1, 2, 3} Processor A Transmit Interrupt Enable n. (Read-Write) • TIEn bit enables Processor A Transmit Interrupt n. • If TIEn bit is set to "1" (enabled), then an Processor A Transmit Interrupt n request is issued when the TEn bit in the ASR register is set to "1". • If TIEn bit is cleared (disabled), then the value of the TEn bit is ignored and no Processor A Transmit Interrupt n request will be issued. • TIEn bit is cleared when the MU resets. 0000 - Disables Processor A Transmit Interrupt n. (default) 0001 - Enables Processor A Transmit Interrupt n.
19-16 GIRn	GIRn For n = {0, 1, 2, 3} Processor A General Purpose Interrupt Request n. (Read-Write) • Writing "1" to the GIRn bit sets the GIPn bit in the BSR register on the Processor B-side. If the GIEn bit in the BCR register is set to "1" on the Processor B-side, a General Purpose Interrupt n request is triggered. • The GIRn bit is cleared if the GIPn bit (in the BSR register on the Processor B-side) is cleared by writing it (GIPn bit) as "1", thereby signalling the Processor A that the interrupt was accepted (cleared by the software). The GIPn bit cannot be written as "0" on the Processor A-side. • To ensure proper operations, verify that the GIRn bit is cleared (meaning that there is no pending interrupt) before setting it (GIRn bit). • GIRn bit is cleared when the MU resets. 0000 - Processor A General Interrupt n is not requested to the Processor B (default). 0001 - Processor A General Interrupt n is requested to the Processor B.
15-7 —	- Reserved
6 BRDIE	BRDIE Processor B Reset De-assertion Interrupt Enable. (Read-Write) • BRDIE bit enables Processor A General Interrupt 3. • If BRDIE bit is set to "1", then General Interrupt 3 request is issued to the Processor A when the BRDIP bit in the ASR register is set to "1". • If BRDIE is cleared, then the value of the BRDIP bit is ignored and no General Interrupt 3 request will be issued. • The BRDIE bit is cleared when the MU resets. 0 - Disables the Processor A General Purpose Interrupt 3 request due to the Processor B reset de-assertion to the Processor A. Processor B reset deassertion causes Processor B and MU-Processor B side to come out of reset thus setting BRDIP bit to "1".

Table continues on the next page...

Field	Description
	1 - Enables Processor A General Purpose Interrupt 3 request due to the Processor B reset de-assertion to the Processor A.
5 MUR	MUR Processor A MU Reset. • Setting MUR bit to "1" resets both the Processor B and the Processor A sides of the MU module, forcing all control and status registers to return to their default values (except the BHR bit in the ACR register and BHRM bit in BCR register), and all internal states to be cleared. • Before setting the MUR bit to "1", it is advisable to interrupt the Processor B, because setting the MUR bit may affect the ongoing Processor B program. • After setting the MUR bit, monitor the value of the BRS bit in the ASR register to know when the reset sequence on the Processor B-side has ended. • MUR bit can only be written as "1". • MUR bit is always read as "0". • MUR bit is cleared during the MU reset sequence. 0 - N/A. Self clearing bit (default). 1 - Asserts the Processor A MU reset.
4 BHR	BHR Processor B Hardware Reset. (Read-Write) • BHR bit asserts and de-asserts the hardware reset of the Processor B. • Set BHR bit to "1" to start a hardware reset of the Processor B. • Clear the BHR bit to de-assert the Processor B hardware reset input. • Assert the BHR bit for a minimum of 3 clock cycles of network clock (sampling clock in SRC) clock. The BRS bit in MU_ASR register (b[7]) indicates the state of the Processor B. As soon as the Processor B goes into Reset (BRS bit is set to "1"), the BHR bit can be de-asserted. • Strobe-setting the BHR bit will not cause an internal MU reset but will be routed outside MU to Processor B domain reset logic • After clearing the BHR bit, monitor the value of the BRS bit at the ASR to know when the Processor B reset sequence has ended. • The BHR reset issued by the Processor A to the Processor B is maskable by the Processor B (according to the settings of the BHRM bit in the BCR register). If the BHRM bit (in the BCR register) is set to "1", then the BHR reset is masked; if the BHRM bit (in the BCR register) is cleared (default), then the BHR reset is enabled. • The BHR bit does not return to the reset value during the software (MUR) reset. 0 - De-assert Hardware reset to the Processor B. (default) 1 - Assert Hardware reset to the Processor B.
3 —	- Reserved
2-0 ABFn	ABFn For n = {0, 1, 2} Processor A to Processor B Flag n. (Read-Write) • ABFn bit is a read-write flag that is reflected in Fn bit in the BSR register on the Processor B-side. • ABFn bit is cleared when the MU resets. 000 - N/A. Self clearing bit (default). 001 - Asserts the Processor A MU reset.

4.3.7 MU Processor B-side Memory Map/Register Definition

4.3.7.1 MUB register descriptions

This section contains the detailed register descriptions for the Processor B-side MU registers.

4.3.7.1.1 MUB memory map

MUB base address: 30AB_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Processor B Transmit Register 0 (BTR0)	32	RW	0000_0000
4	Processor B Transmit Register 1 (BTR1)	32	RW	0000_0000
8	Processor B Transmit Register 2 (BTR2)	32	RW	0000_0000
C	Processor B Transmit Register 3 (BTR3)	32	RW	0000_0000
10	Processor B Receive Register 0 (BRR0)	32	RO	0000_0000
14	Processor B Receive Register 1 (BRR1)	32	RO	0000_0000
18	Processor B Receive Register 2 (BRR2)	32	RO	0000_0000
1C	Processor B Receive Register 3 (BRR3)	32	RO	0000_0000
20	Processor B Status Register (BSR)	32	RW	00F0_0080
24	Processor B Control Register (BCR)	32	RW	0000_0000

4.3.7.1.2 Processor B Transmit Register 0 (BTR0)

Use Processor B Transmit Register 0 (BTR0, 32-bit, write-only) to transmit a message or data to the Processor A.

- The BTR0 register can only be written when the TE0 bit in BSR register is set to "1".
- Reading the BTR0 register returns all zeros.

4.3.7.1.2.1 Offset

Register	Offset
BTR0	0h

4.3.7.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BTR0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BTR0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.7.1.2.3 Fields

Field	Description
31-0 BTR0	BTR0 Processor B Transmit Register 0. (Write-only) • Data written to the BTR0 register is reflected on the Processor A-side in the Processor A Receive Register 0 (ARR0). The BTR0 and ARR0 registers are not double-buffered—a write to the BTR0 register overrides the data readable at the ARR0 register. • A write to the transmit register clears a "transmitter empty" bit (TE0) in the Processor B Status Register (BSR) on the transmitter side, and sets a "receiver full" bit (RF0) in the Processor A Status Register (ASR) on the receiver side (optionally triggering an interrupt 0 on the Processor A-side). • Any write to the BTR0 register will update all status information.

4.3.7.1.3 Processor B Transmit Register 1 (BTR1)

Use Processor B Transmit Register 1 (BTR1, 32-bit, write-only) to transmit a message or data to the Processor A.

- The BTR1 register can only be written when the TE1 bit in BSR register is set to "1".
- Reading the BTR1 register returns all zeros.

4.3.7.1.3.1 Offset

Register	Offset
BTR1	4h

4.3.7.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BTR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BTR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.7.1.3.3 Fields

Field	Description
31-0 BTR1	BTR1 Processor B Transmit Register 1. (Write-only) • Data written to the BTR1 register is reflected on the Processor A-side in the Processor A Receive Register 1 (ARR1). The BTR1 and ARR1 registers are not double-buffered—a write to the BTR1 register overrides the data readable at the ARR1 register. • A write to the transmit register clears a "transmitter empty" bit (TE1) in the Processor B Status Register (BSR) on the transmitter side, and sets a "receiver full" bit (RF1) in the Processor A Status Register (ASR) on the receiver side (optionally triggering an interrupt 1 on the Processor A-side). • Any write to the BTR1 register will update all status information.

4.3.7.1.4 Processor B Transmit Register 2 (BTR2)

Use Processor B Transmit Register 2 (BTR2, 32-bit, write-only) to transmit a message or data to the Processor A.

- The BTR2 register can only be written when the TE2 bit in BSR register is set to "1".
- Reading the BTR2 register returns all zeros.

4.3.7.1.4.1 Offset

Register	Offset
BTR2	8h

4.3.7.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BTR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BTR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.7.1.4.3 Fields

Field	Description
31-0 BTR2	BTR2 Processor B Transmit Register 2. (Write-only) • Data written to the BTR2 register is reflected on the Processor A-side in the Processor A Receive Register 2 (ARR2). The BTR2 and ARR2 registers are not double-buffered—a write to the BTR2 register overrides the data readable at the ARR2 register. • A write to the transmit register clears a "transmitter empty" bit (TE2) in the Processor B Status Register (BSR) on the transmitter side, and sets a "receiver full" bit (RF2) in the Processor A Status Register (ASR) on the receiver side (optionally triggering an interrupt 2 on the Processor A-side). • Any write to the BTR2 register will update all status information.

4.3.7.1.5 Processor B Transmit Register 3 (BTR3)

Use Processor B Transmit Register 3 (BTR3, 32-bit, write-only) to transmit a message or data to the Processor A.

- The BTR3 register can only be written when the TE3 bit in BSR register is set to "1".
- Reading the BTR3 register returns all zeros.

4.3.7.1.5.1 Offset

Register	Offset
BTR3	Ch

4.3.7.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BTR3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BTR3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.7.1.5.3 Fields

Field	Description
31-0 BTR3	BTR3 Processor B Transmit Register 3. (Write-only) • Data written to the BTR3 register is reflected on the Processor A-side in the Processor A Receive Register 3 (ARR3). The BTR3 and ARR3 registers are not double-buffered—a write to the BTR3 register overrides the data readable at the ARR3 register. • A write to the transmit register clears a "transmitter empty" bit (TE3) in the Processor B Status Register (BSR) on the transmitter side, and sets a "receiver full" bit (RF3) in the Processor A Status Register (ASR) on the receiver side (optionally triggering an interrupt 3 on the Processor A-side). • Any write to the BTR3 register will update all status information.

4.3.7.1.6 Processor B Receive Register 0 (BRR0)

Use Processor B Receive Register 0 (BRR0, 32-bit, read-only) to receive a message or data from the Processor A.

- Data written to the ATR0 register is immediately reflected in the BRR0 register.
- The BRR0 register can only be read when the RF0 bit in the BSR register is set to "1".
- Writing to the BRR0 register generates an error response to the Processor B.

4.3.7.1.6.1 Offset

Register	Offset
BRR0	10h

4.3.7.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BRR0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BRR0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.7.1.6.3 Fields

Field	Description
31-0 BRR0	BRR0 Processor B Receive Register 0. (Read-only) • Reflects the data written to Processor A Transmit Register 0 (ATR0). • Reading the BRR0 register clears the "receiver full" bit (RF0) in the Processor B Status Register (BSR) on the receiver side, and sets the "transmitter empty" bit (TE0) in the Processor A Status Register on the transmitter side (optionally triggering a transmit interrupt 0 on the Processor A-side). • Any read of the BRR0 register will update all status information.

4.3.7.1.7 Processor B Receive Register 1 (BRR1)

Use Processor B Receive Register 1 (BRR1, 32-bit, read-only) to receive a message or data from the Processor A.

- Data written to the ATR1 register is immediately reflected in the BRR1 register.
- The BRR1 register can only be read when the RF1 bit in the BSR register is set to "1".
- Writing to the BRR1 register generates an error response to the Processor B.

4.3.7.1.7.1 Offset

Register	Offset
BRR1	14h

4.3.7.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BRR1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BRR1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.7.1.7.3 Fields

Field	Description
31-0 BRR1	BRR1 Processor B Receive Register 1. (Read-only) • Reflects the data written to Processor A Transmit Register 1 (ATR1). • Reading the BRR1 register clears the "receiver full" bit (RF1) in the Processor B Status Register (BSR) on the receiver side, and sets the "transmitter empty" bit (TE1) in the Processor A Status Register on the transmitter side (optionally triggering a transmit interrupt 1 on the Processor A-side). • Any read of the BRR1 register will update all status information.

4.3.7.1.8 Processor B Receive Register 2 (BRR2)

Use Processor B Receive Register 2 (BRR2, 32-bit, read-only) to receive a message or data from the Processor A.

- Data written to the ATR2 register is immediately reflected in the BRR2 register.
- The BRR2 register can only be read when the RF2 bit in the BSR register is set to "1".
- Writing to the BRR2 register generates an error response to the Processor B.

4.3.7.1.8.1 Offset

Register	Offset
BRR2	18h

4.3.7.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BRR2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BRR2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.7.1.8.3 Fields

Field	Description
31-0 BRR2	BRR2 Processor B Receive Register 2. (Read-only) • Reflects the data written to Processor A Transmit Register 1 (ATR2). • Reading the BRR2 register clears the "receiver full" bit (RF2) in the Processor B Status Register (BSR) on the receiver side, and sets the "transmitter empty" bit (TE2) in the Processor A Status Register on the transmitter side (optionally triggering a transmit interrupt 2 on the Processor A-side). • Any read of the BRR2 register will update all status information.

4.3.7.1.9 Processor B Receive Register 3 (BRR3)

Use Processor B Receive Register 3 (BRR3, 32-bit, read-only) to receive a message or data from the Processor A.

- Data written to the ATR3 register is immediately reflected in the BRR3 register.
- The BRR3 register can only be read when the RF3 bit in the BSR register is set to "1".
- Writing to the BRR3 register generates an error response to the Processor B.

4.3.7.1.9.1 Offset

Register	Offset
BRR3	1Ch

4.3.7.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BRR3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BRR3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.7.1.9.3 Fields

Field	Description
31-0 BRR3	BRR3 Processor B Receive Register 3. (Read-only) • Reflects the data written to Processor A Transmit Register 3 (ATR3). • Reading the BRR3 register clears the "receiver full" bit (RF3) in the Processor B Status Register (BSR) on the receiver side, and sets the "transmitter empty" bit (TE3) in the Processor A Status Register on the transmitter side (optionally triggering a transmit interrupt 3 on the Processor A-side). • Any read of the BRR3 register will update all status information.

4.3.7.1.10 Processor B Status Register (BSR)

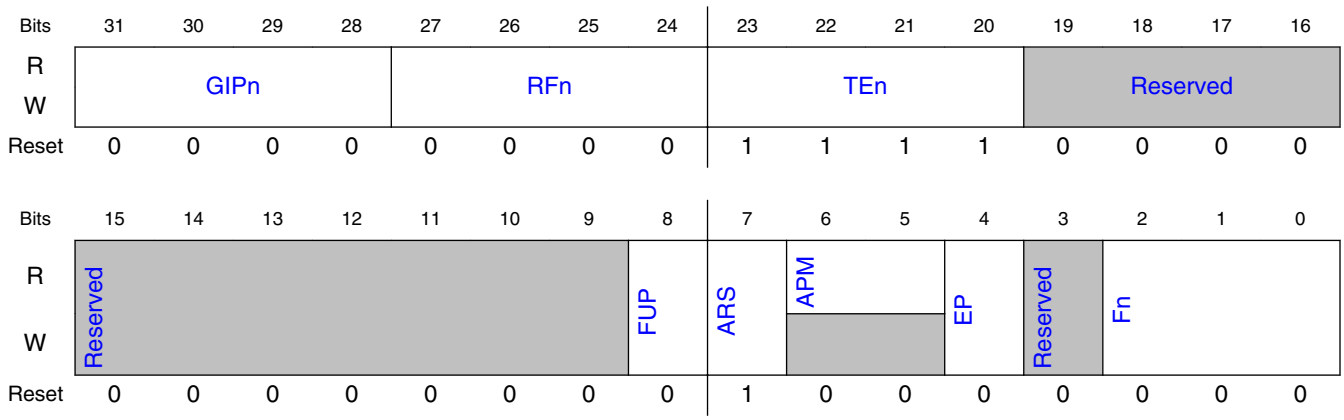
Use the Processor B Status Register (BSR, 32-bit, read-write) to show interrupt status from the Processor B, general purpose flags, the Processor A power mode, and to set dual function control-status bits.

- Dual-purpose bits are set by the Processor B-side programmer, and cleared by the MU logic.

4.3.7.1.10.1 Offset

Register	Offset
BSR	20h

4.3.7.1.10.2 Diagram



4.3.7.1.10.3 Fields

Field	Description
31-28 GIPn	GIPn For n = {0, 1, 2, 3} Processor B General Interrupt Request n Pending. (Read-Write) • GIPn bit signals the Processor B that the GIRn bit in the ACR register on the Processor A-side was set from "0" to "1". If the GIEn bit in the BCR register is set to "1", a General Interrupt n request is issued. • The GIPn bit is cleared by writing it back as "1". Writing "0", or writing "1" when the GIPn bit is cleared is ignored. Use this feature in the interrupt routine, where the GIPn bit is cleared in order to de-assert the interrupt request source at the interrupt controller. • GIPn bit is cleared when the MU is reset. 0000 - Processor B general purpose interrupt n is not pending. (default) 0001 - Processor B general purpose interrupt n is pending.
27-24 RFn	RFn For n = {0, 1, 2, 3} Processor B Receive Register n Full. (Read-only) • RFn bit signals to the Processor B-side that new data was written by the Processor A to the ATRn register, and is ready to be read by the Processor B in the BRRn register. • The RFn bit is set to "1" when the ATRn register is written on the Processor A-side. • After the RFn bit is set to "1", the RFn bit signals the Processor B-side that new data is ready to be read by the Processor B in the BRRn register, and a Receive n interrupt is issued on the Processor A-side (if the RIEn bit in the BCR register has been set to "1"). • RFn bit is cleared when the BRRn register is read, and when the MU is reset. 0000 - BRRn register is not full (default). 0001 - BRRn register has received data from ATRn register and is ready to be read by the Processor B.
23-20 TEn	TEn For n = {0, 1, 2, 3} Processor B Transmit Register n Empty. (Read-only) • When TEn = "1", it signals to the Processor B-side that the BTRn register is ready to be written on the Processor B-side. • The TEn bit is set to "1" after the ARRn register is read on the Processor A-side. • Setting TEn bit will issue a transmit n interrupt on the Processor B-side (if the TIEn bit in the BCR register is set to "1".

Table continues on the next page...

Field	Description
	• TEn bit is cleared after the BTRn register is written on the Processor B-side. • TEn bit is set to "1" when the MU is reset. 0000 - BTRn register is not empty. 0001 - BTRn register is empty (default).
19-9 —	Reserved
8 FUP	FUP Processor B Flags Update Pending. (Read-only) • FUP bit is set to "1" when the Processor B-side sends a Flags Update request to the Processor A-side. • A Flags Update request is generated when the BAF[2:0] bits of the BCR register change. No flag update changes are allowed while the FUP bit is set to "1". Any write to the BAF[2:0] bits, while the FUP bit is set to "1", will not generate a Flags Update event, and the BAF[2:0] bits will stay unchanged. • FUP bit is cleared when this Flags Update request is internally acknowledged (that the flag is updated) from the MU Processor A-side, and during MU reset. 0 - No flags updated, initiated by the Processor B, in progress (default) 1 - Processor B initiated flags update, processing
7 ARS	ARS Processor A Reset State. (Read-only) • ARS bit indicates if the Processor A-side of the MU is in a reset state or not. • If the ARS bit is set to "1", then the Processor A-side of the MU is still in the reset state. • If the ARS bit is cleared, then both the Processor A and the Processor A-side of the MU are out of reset. • The ARS bit is set to "1" during: a Processor A system reset, or an MU reset (caused by setting the MUR bit at the BCR register). • The ARS bit is cleared when the reset sequence on the Processor A-side of the MU ends. After issuing any of the three reset events mentioned previously, it should be verified that the ARS bit is cleared before starting any accesses. • When Processor B side of MU comes out of reset ARS bit has value "1"(default).Then Processor B sees the status of Processor A side and if Processor A has come out of reset then ARS bit goes low. This takes 5-6 clock cycles. If the ARS bit is read now it is low although its reset value was "1" . 0 - The Processor A or the Processor A-side of the MU is not in reset. 1 - The Processor A or the Processor A-side of the MU is in reset.
6-5 APM	APM Processor A Power Mode. (Read-only) • APM[1:0] bits indicate the Processor A power mode. 00 - The System is in Run Mode. 01 - The System is in WAIT Mode. 10 - Reserved. 11 - The System is in STOP Mode.
4 EP	EP Processor B-Side Event Pending. (Read-only) • EP bit is set to "1" when the Processor B-side mechanism sends an event update request to the Processor A-side.

Table continues on the next page...

Field	Description
	• EP bit is cleared when the event update acknowledge is received. An "event" is any hardware message that is reflected in the ASR register on the Processor A-side (for example, "transmit register 0 written"). During normal operations, the state of the EP bit does not have to be manually updated because the event update mechanism works automatically. • To ensure events have been posted to Processor A before entering STOP mode, it should be verified that the EP bit is cleared. If EP bit is set to "1", continue to poll it (EP bit) before entering STOP mode. • Reading the BSR register (to check the EP bit) should be the last access to the MU that should be performed before entering STOP mode; otherwise, the EP bit may be set by subsequent additional actions. • Due to Processor B pipeline effects, three NOP operations (or their timing equivalent) should be given after an instruction that sets an event before the EP bit can reflect this event. • The EP bit is cleared when the MU resets. 0 - The Processor B-side event is not pending (default). 1 - The Processor B-side event is pending.
3 —	Reserved
2-0 Fn	Fn For n = {0, 1, 2} Processor B-Side Flag n. (Read-only) • Fn bit is the Processor B-side flag that reflects the values written to the ABFn bit in the Processor A control register. • Every time that the ABFn bit is written, the ABFn bit write event updates the Fn bit after the event update latency, which is measured in terms of the number of clocks of the Processor A and the Processor B. 000 - ABFn bit in ACR register is written 0 (default). 001 - ABFn bit in ACR register is written 1.

4.3.7.1.11 Processor B Control Register (BCR)

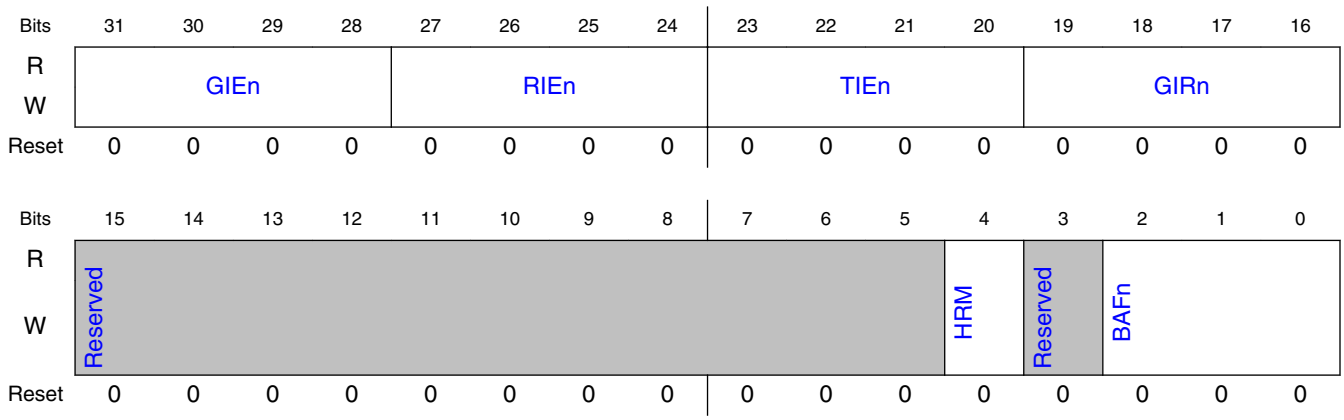
Use the Processor B Control Register (BCR, 32-bit, read-write) to enable the MU interrupts on the Processor B-side, and trigger events and interrupts on the Processor A-side (wake from STOP, hardware reset, flag update).

For the fields GIEn, RIEn, TIEn and GIRn, n=0 corresponds to the high order bit and n=3 corresponds to the low order bit.

4.3.7.1.11.1 Offset

Register	Offset
BCR	24h

4.3.7.1.11.2 Diagram



4.3.7.1.11.3 Fields

Field	Description
31-28 GIEn	GIEn For n = {0, 1, 2, 3} Processor B General Purpose Interrupt Enable n. (Read-Write) • GIEn bit enables Processor B General Interrupt n. • If GIEn bit is set to "1" (enabled), then a General Interrupt n request is issued when the GIPn bit in the BSR register is set to "1". • If GIEn is cleared (disabled), then the value of the GIPn bit is ignored and no General Interrupt n request will be issued. • GIEn bit is cleared when the MU resets. 0000 - Disables Processor B General Interrupt n. (default) 0001 - Enables Processor B General Interrupt n.
27-24 RIEn	RIEn For n = {0, 1, 2, 3} Processor B Receive Interrupt Enable n. (Read-Write) • RIEn bit enables Processor B Receive Interrupt n. • If RIEn bit is set to "1" (enabled), then an Processor B Receive Interrupt n request is issued when the RFn bit in the BSR register is set to "1". • If RIEn bit is cleared (disabled), then the value of the RFn bit is ignored and no Processor B Receive Interrupt n request will be issued. • RIEn bit is cleared when the MU resets. 0000 - Disables Processor B Receive Interrupt n. (default) 0001 - Enables Processor B Receive Interrupt n.
23-20 TIEn	TIEn For n = {0, 1, 2, 3} Processor B Transmit Interrupt Enable n. (Read-Write) • TIEn bit enables Processor B Transmit Interrupt n. • If TIEn bit is set to "1" (enabled), then an Processor B Transmit Interrupt n request is issued when the TEn bit in the BSR register is set to "1". • If TIEn bit is cleared (disabled), then the value of the TEn bit is ignored and no Processor B Transmit Interrupt n request will be issued. • TIEn bit is cleared when the MU resets. 0000 - Disables Processor B Transmit Interrupt n. (default)

Table continues on the next page...

Field	Description
	0001 - Enables Processor B Transmit Interrupt n.
19-16 GIRn	GIRn For n = {0, 1, 2, 3} Processor B General Purpose Interrupt Request n. (Read-Write) • Writing "1" to the GIRn bit sets the GIPn bit in the ASR register on the Processor A-side. If the GIEn bit in the ACR register is set to "1" on the Processor A-side, a General Purpose Interrupt n request is triggered. • The GIRn bit is cleared if the GIPn bit (in the ASR register on the Processor A-side) is cleared by writing it (GIPn bit) as "1", thereby signalling the Processor B that the interrupt was accepted (cleared by the software). The GIPn bit cannot be written as "0" on the Processor B-side. • To ensure proper operations, it must be verified that the GIRn bit is cleared (meaning that there is no pending interrupt) before setting it (GIRn bit). • GIRn bit is cleared when the MU resets. 0000 - Processor B General Interrupt n is not requested to the Processor A (default). 0001 - Processor B General Interrupt n is requested to the Processor A.
15-5 —	Reserved
4 HRM	HRM Processor B Hardware Reset Mask. (Read-Write) • The Processor A can give a hardware reset to the Processor B by setting the BHR bit in the ACR Register to "1". • When the HRM bit is set to "1" by the Processor B, the BHR reset issued by the Processor A is masked (disabled by the Processor B). • When the HRM bit is cleared, the BHR reset issued by the Processor A to the Processor B is not masked (enabled by the Processor B). 0 - BHR bit in ACR is not masked, enables the hardware reset to the Processor B (default after hardware reset). 1 - BHR bit in ACR is masked, disables the hardware reset request to the Processor B.
3 —	Reserved
2-0 BAFn	BAFn For n = {0, 1, 2} Processor B to Processor A Flag n. (Read-Write) • BAFn bit is a read-write flag that is reflected in Fn bit in the ASR register on the Processor A-side. • BAFn bit is cleared when the MU resets. 000 - Clears the Fn bit in the ASR register. 001 - Sets the Fn bit in the ASR register.

4.4 Semaphore (SEMA4)

4.4.1 Overview

The Semaphores (SEMA4) module provides a platform IPS slave device which implements 16 hardware-enforced gates.

4.4.1.1 Block Diagram

A simplified block diagram of the Semaphores module is shown in Figure 4-7. This is a dual-processor configuration, where cp0 is core processor 0 and cp1 is core processor 1. In the diagram, the register blocks named gate0, gate1,…, gate 15 include the finite state machines (FSM) implementing the semaphore gates plus the interrupt notification logic.

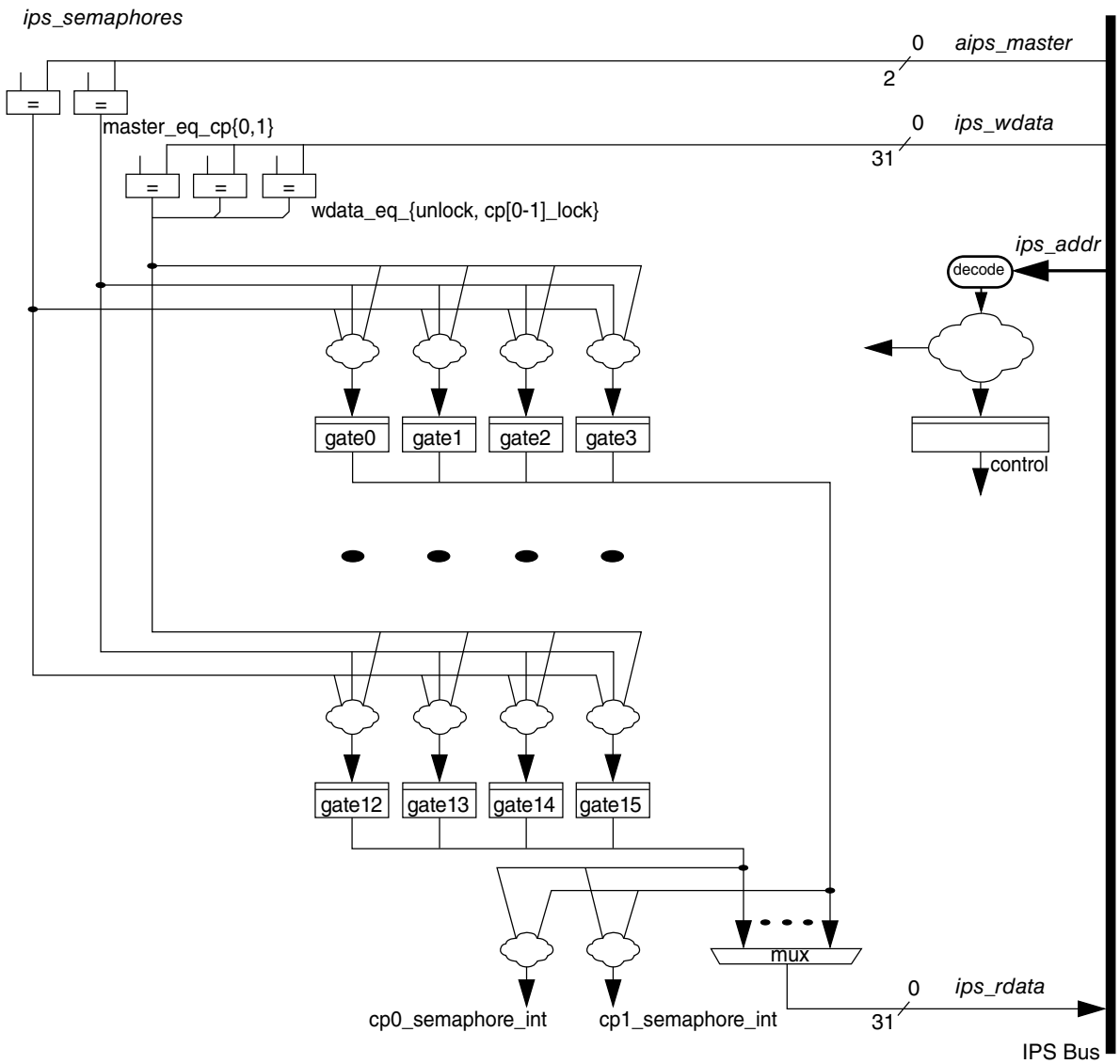


Figure 4-7. Semaphores Block Diagram

4.4.1.2 Features

The Semaphores module implements hardware-enforced semaphores as an IPS-mapped slave peripheral device. The feature set includes:

- Support for 16 hardware-enforced gates in a dual-processor configuration
  - Each hardware gate appears as a 3-state, 2-bit state machine, with all 16 gates mapped as a byte-size array
    - Processors lock gates by writing "processor_number+1" to the appropriate gate and must read back the gate value to verify the lock operation was successful.

      3-state implementation:

      if gate = 0b00, then state = unlocked
      if gate = 0b01, then state = locked by processor 0
      if gate = 0b10, then state = locked by processor 1
    - Uses the bus master number/ID as a reference attribute plus the specified data patterns to validate all write operations
    - Once locked, the gate can (and must) be unlocked by a write of zeroes from the locking processor
  - Optional interrupt notification after a failed lock write provides a mechanism to indicate when the gate is unlocked
  - Secure reset mechanisms are supported to clear the contents of individual gates or notification logic, as well as a clear_all capability
- Memory-mapped IPS slave peripheral platform module
  - Interface to the IPS bus for programming-model accesses
  - Two outputs (one per processor) for interrupt notification of failed lock writes

4.4.2 Functional Description

In this section, the functional operation of the Semaphores module, specifically the state machines of the SEMA4_GATEn and SEMA4_CPnNTF registers are detailed.

4.4.2.1 Modes of Operation

The Semaphores module does not support any special modes of operation. As a slave peripheral memory-mapped device located on the platform's IPS slave bus, it responds based strictly on the memory addresses of the connected bus. The IPS bus is used to access the Semaphores' programming model.

4.4.2.2 SEMA4_GATEn Operation

Recall each of the SEMA4_GATEn registers implements a 2-bit, 3-state machine. The state transitions for each gate are shown in the following figure.

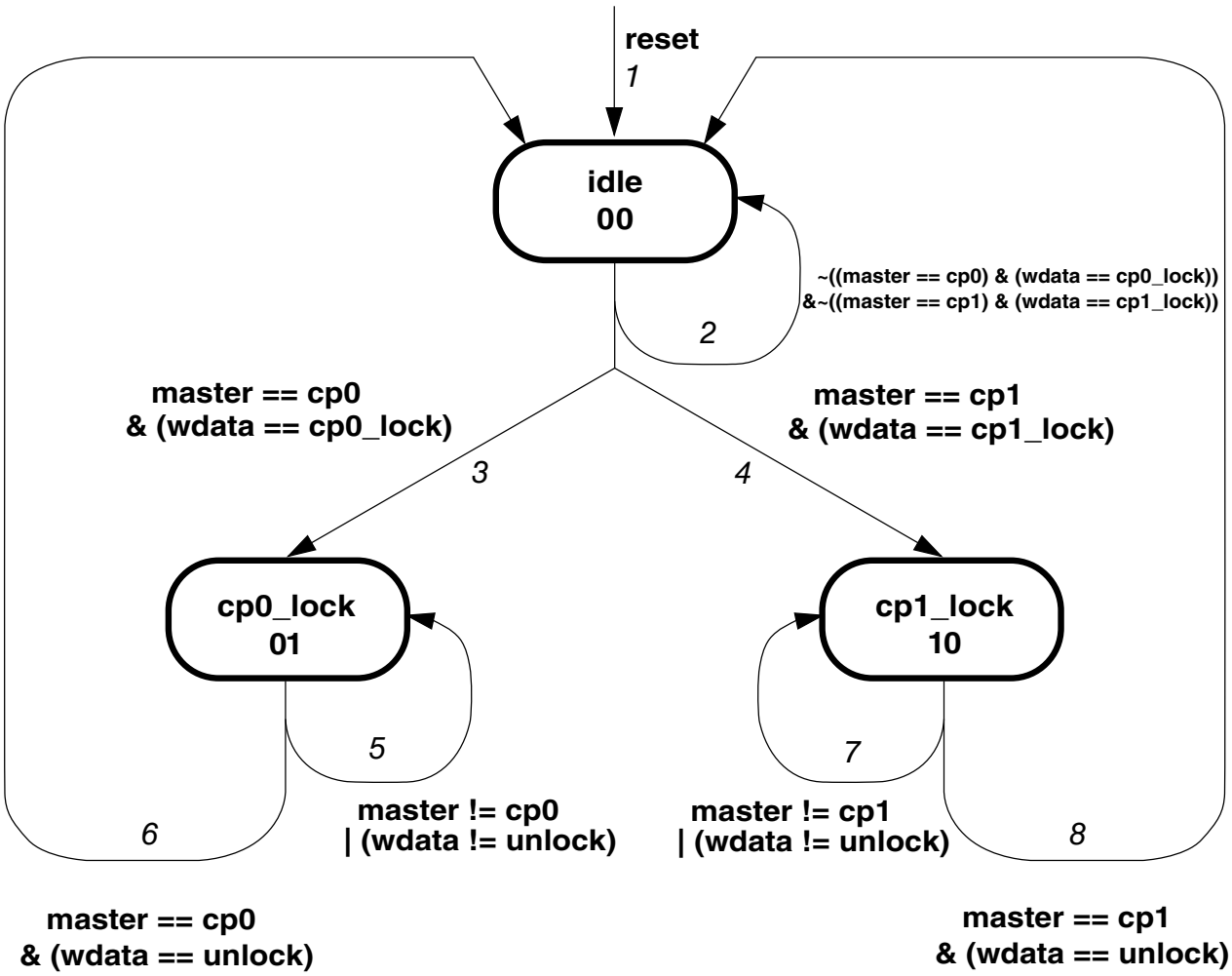


Figure 4-8. SEMA4_GATEn State Machine

The bus master number/ID is used to identify core processor 0 (cp0) or core processor 1 (cp1).

The state transitions for SEMA4_GATEn are defined in the following table.

Table 4-12. SEMA4_GATEn State Transitions

Current State	Next State	Transition	Description
–	idle	1	Any reset, whether a system reset or an individual gate reset, unconditionally forces the gate into the idle state.

Table continues on the next page...

Table 4-12. SEMA4_GATEn State Transitions (continued)

Current State	Next State	Transition	Description
idle	idle	2	Unless a write of the appropriate lock value from the corresponding processor occurs, the gate remains in the idle state.
idle	cp0_lock	3	When a write of the "cp0_lock" data value is initiated by processor 0, the gate transitions into the cp0_lock state.
idle	cp1_lock	4	When a write of the "cp1_lock" value is initiated by processor 1, the gate transitions into the cp1_lock state.
cp0_lock	cp0_lock	5	Once in this state, the gate remains here if any attempted write is not from cp0 with the unlock data value.
cp0_lock	idle	6	The gate returns to the idle (unlocked) state once a write from cp0 with the unlock data value occurs.
cp1_lock	cp1_lock	7	Once in this state, the gate remains here if any attempted write is not from cp1 with the unlock data value.
cp1_lock	idle	8	The gate returns to the idle (unlocked) state once a write from cp1 with the unlock data value occurs.

4.4.2.3 SEMA4_CPnNTF Operation

The failed lock write notification interrupt request is implemented in a 3-bit, 5-state machine which records failed lock attempts and transitions based on gate locking and unlocking. Two specific states are encoded and program-visible as SEMA4_CP0NTF[GNn] and SEMA4_CP1NTF[GNn]. See the following figure.

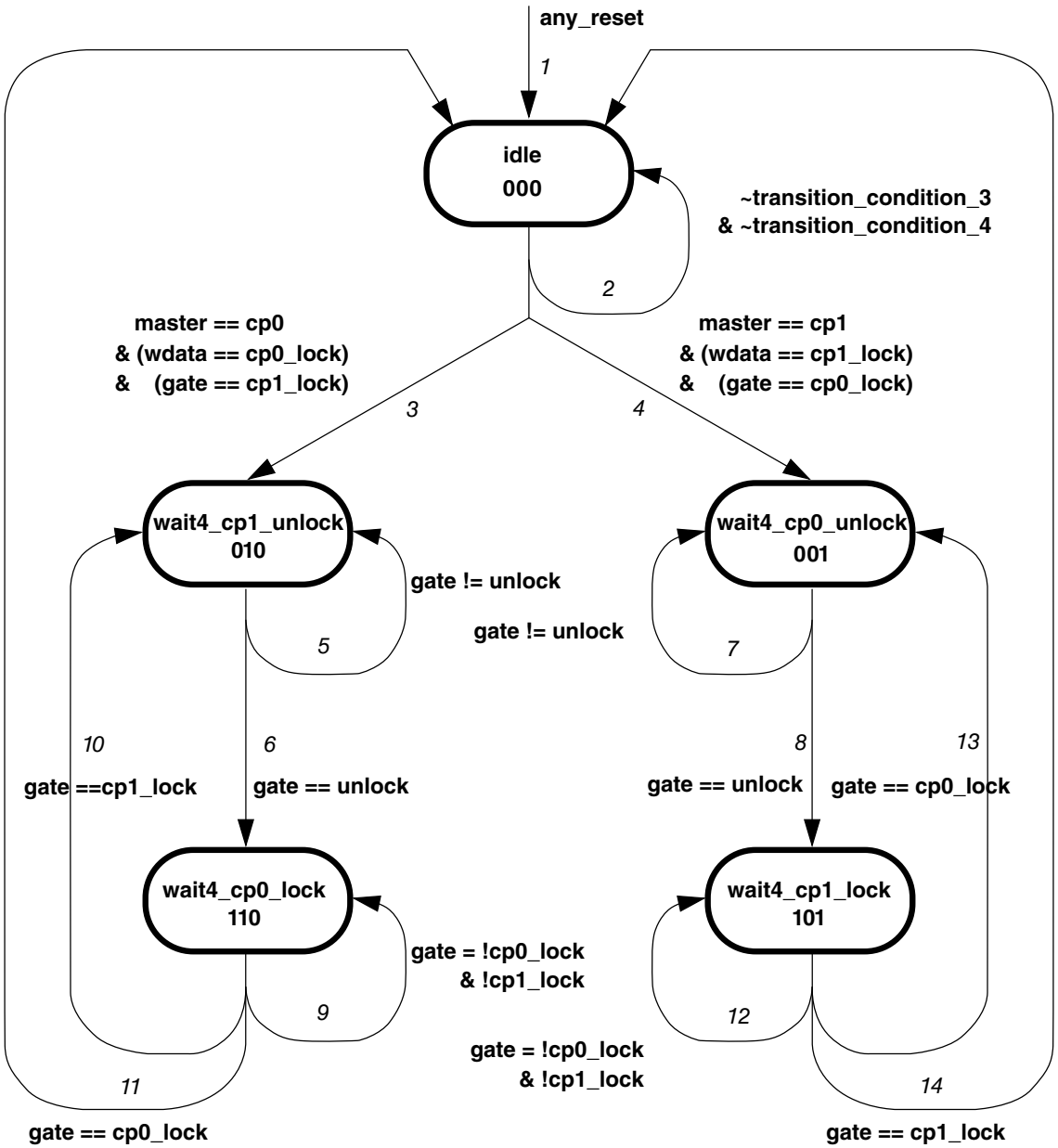


Figure 4-9. IRQ Notification State Machine

The state transitions of the IRQ notification function are defined in the following table. Specific states of this machine are program-visible as the SEMA4_CPnNTF registers. In particular, two states are program-visible:

```
if state = wait4_cp0_lock (0b110) // generate cp0_semaphore_int if properly enabled
then SEMA4_CP0NTF[GNn] = 1;
else SEMA4_CP0NTF[GNn] = 0

if state = wait4_cp1_lock (0b101) // generate cp1_semaphore_int if properly enabled
then SEMA4_CP1NTF[GNn] = 1;
else SEMA4_CP1NTF[GNn] = 0
```

Table 4-13. IRQ Notification State Transitions

Current State	Next State	Transition	Description
–	idle	1	Any reset, including a system reset or an individual notification or secure gate reset, unconditionally forces the machine into the idle state.
idle	idle	2	Unless a write of the appropriate lock value from the corresponding processor to an already-locked gate occurs, the machine remains in the idle state.
idle	wait4_cp1_unlock	3	When a write of the "cp0_lock" data value is initiated by processor 0 but the gate is already locked by cp1, the machine transitions into this state, where it waits for cp1 to unlock the gate.
idle	wait4_cp0_unlock	4	When a write of the "cp1_lock" data value is initiated by processor 1but the gate is already locked by cp0, the machine transitions into this state, where it waits for cp0 to unlock the gate
wait4_cp1_unlock	wait4_cp1_unlock	5	Once in this state, the machine remains here until the gate is unlocked.
wait4_cp1_unlock	wait4_cp0_lock	6	From this state, the machine transitions into the next state, waiting for cp0 to lock the gate, once it has been unlocked.
wait4_cp0_unlock	wait4_cp0_unlock	7	Once in this state, the machine remains here until the gate is unlocked.
wait4_cp0_unlock	wait4_cp1_lock	8	From this state, the machine transitions into the next state, waiting for cp1 to lock the gate, once it has been unlocked.
wait4_cp0_lock	wait4_cp0_lock	9	In this state, the machine generates the notification interrupt (if properly-enabled) and remains here until the gate is locked by processor 0 or the gate is again locked by processor 1.
wait4_cp0_lock	wait4_cp1_unlock	10	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is again locked by processor 1. With this transition, the notification interrupt request is negated.
wait4_cp0_lock	idle	11	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is finally locked by processor 0. With this transition, the notification interrupt request is negated.
wait4_cp1_lock	wait4_cp1_lock	12	In this state, the machine generates the notification interrupt (if properly-enabled) and remains here until the gate is locked by processor 1or the gate is again locked by processor 0.
wait4_cp1_lock	wait4_cp0_unlock	13	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is again locked by processor 0. With this transition, the notification interrupt request is negated.
wait4_cp1_lock	idle	14	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is finally locked by processor 1. With this transition, the notification interrupt request is negated.

The Semaphores module generates two interrupt request output signals, one per processor, combining the SEMA4_CPnINE and SEMA4_CPnNTF registers, where the boolean equations are:

```
cp0_semaphore_int
    =    SEMA4_CP0INE[INE0]    &    SEMA4_CP0NTF[GN0]
    |    SEMA4_CP0INE[INE1]    &    SEMA4_CP0NTF[GN1]
    |    SEMA4_CP0INE[INE2]    &    SEMA4_CP0NTF[GN2]
    ...
    |    SEMA4_CP0INE[INE15]    &    SEMA4_CP0NTF[GN15]
cp1_semaphore_int
    =    SEMA4_CP1INE[INE0]    &    SEMA4_CP1NTF[GN0]
    |    SEMA4_CP1INE[INE1]    &    SEMA4_CP1NTF[GN1]
    |    SEMA4_CP1INE[INE2]    &    SEMA4_CP1NTF[GN2]
    ...
    |    SEMA4_CP1INE[INE15]    &    SEMA4_CP1NTF[GN15]
```

4.4.3 External Signal Description

The Semaphores module does not include any external interfaces.

4.4.4 Initialization Information

The reset state of the Semaphores module allows it to begin operation without the need for any further initialization. All the internal state machines are cleared by any reset event, allowing the module to immediately begin operation.

4.4.5 Application Information

In an operational multi-core system, most interactions involving the Semaphores module involves reads and writes to the SEMA4_GATEn registers for implementation of the hardware-enforced software gate functions. Typical code segments for gate functions perform the following operations:

- To lock (close) a gate
    - The processor performs a byte write of "logical_processor_number + 1" to gate[i]
    - The processor reads back gate[i] and checks for a value of "logical_processor_number + 1"

If the compare indicates the expected value, then the gate is locked; proceed with the protected code segment. If the compare does not indicate the expected value, the lock operation failed; repeat the process beginning with byte write to gate[i] in spin-wait loop, or proceed with another execution path and wait for failed lock interrupt notification.

A simple C-language example of a `gateLock` function is shown in the following figure.

```
#define UNLOCK      0
#define CP0_LOCK    1
#define CP1_LOCK    2

void gateLock (n)
int   n;                                        /* gate number to lock */
{
   int i;
   int current_value;
   int locked_value;

   i = processor_number();  /* obtain logical CPU number */

   if (i == 0)
      locked_value = CP0_LOCK;
   else
      locked_value = CP1_LOCK;

   /* read the current value of the gate and wait until the state == UNLOCK */
   do {
        current_value = gate[n];
   } while (current_value != UNLOCK);

   /* the current value of the gate == UNLOCK. attempt to lock the gate for this
        processor. spin-wait in this loop until gate ownership is obtained */
   do {
      gate[n] = locked_value;  /* write gate with processor_number + 1 */
      current_value = gate[n]; /* read gate to verify ownership was obtained */
   } while (current_value != locked_value);
```

Figure 4-10. Sample gateLock Function

- To unlock (open) a gate
    - After completing the protected code segment, the locking processor performs a byte write of zeroes to gate[i], opening (unlocking) the gate

A few comments on the logical CPU number are appropriate. In this example, a reference to `processor_number()` is used to retrieve this hardware configuration value. Typically, the logical processor numbers are defined by a hardwired input vector to the individual cores. The exact method for accessing the logical processor number varies by architecture.

If the optional failed lock IRQ notification mechanisms are used, then accesses to the related registers (SEMA4_CPnINE, SEMA4_ CPnNTF) are required. Note that there is no required negation of the failed lock write notification interrupt, as the request is automatically cleared by the Semaphores module once the gate has been successfully locked by the "failing" processor.

Finally, in the event a system state requires a software-controlled reset of a gate or IRQ notification register(s), accesses to the secure reset control registers (SEMA4_RSTGT, SEMA4_RSTNTF) are required. For these situations, it is recommended that the

appropriate IRQ notification enable (SEMA4_CPnINE) bits should be disabled *before* initiating the secure reset 2-write sequence (write RSTGT and write RSTNTF), to avoid any race conditions involving spurious notification interrupt requests.

4.4.6 Memory map and register definition

The Semaphores module provides an IPS programming model mapped to an on-platform 16 KB space. The description here specifies a dual-core configuration with 16 semaphore gates. All the register names are prefixed with "SEMA4" as an abbreviation for the full module name.

The programming model is referenced using 8-, 16- and 32-bit accesses. Reads can use any reference size, while writes are generally restricted to the size of the register. Exceptions to the write size restrictions are detailed in the individual register descriptions. Attempted references using inappropriate access sizes, to undefined (reserved) addresses, or with a non-supported access type (for example, a write to a read-only register) generate an IPS error termination.

The 16 KB Semaphores programming model map is shown in the following table.

SEMA4 memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30AC_0000	Semaphores Gate n Register (SEMA4_Gate0)	8	R/W	00h	4.4.6.1/154
30AC_0001	Semaphores Gate n Register (SEMA4_Gate1)	8	R/W	00h	4.4.6.1/154
30AC_0002	Semaphores Gate n Register (SEMA4_Gate2)	8	R/W	00h	4.4.6.1/154
30AC_0003	Semaphores Gate n Register (SEMA4_Gate3)	8	R/W	00h	4.4.6.1/154
30AC_0004	Semaphores Gate n Register (SEMA4_Gate4)	8	R/W	00h	4.4.6.1/154
30AC_0005	Semaphores Gate n Register (SEMA4_Gate5)	8	R/W	00h	4.4.6.1/154
30AC_0006	Semaphores Gate n Register (SEMA4_Gate6)	8	R/W	00h	4.4.6.1/154
30AC_0007	Semaphores Gate n Register (SEMA4_Gate7)	8	R/W	00h	4.4.6.1/154
30AC_0008	Semaphores Gate n Register (SEMA4_Gate8)	8	R/W	00h	4.4.6.1/154
30AC_0009	Semaphores Gate n Register (SEMA4_Gate9)	8	R/W	00h	4.4.6.1/154
30AC_000A	Semaphores Gate n Register (SEMA4_Gate10)	8	R/W	00h	4.4.6.1/154
30AC_000B	Semaphores Gate n Register (SEMA4_Gate11)	8	R/W	00h	4.4.6.1/154
30AC_000C	Semaphores Gate n Register (SEMA4_Gate12)	8	R/W	00h	4.4.6.1/154
30AC_000D	Semaphores Gate n Register (SEMA4_Gate13)	8	R/W	00h	4.4.6.1/154
30AC_000E	Semaphores Gate n Register (SEMA4_Gate14)	8	R/W	00h	4.4.6.1/154
30AC_000F	Semaphores Gate n Register (SEMA4_Gate15)	8	R/W	00h	4.4.6.1/154

Table continues on the next page...

SEMA4 memory map (continued)

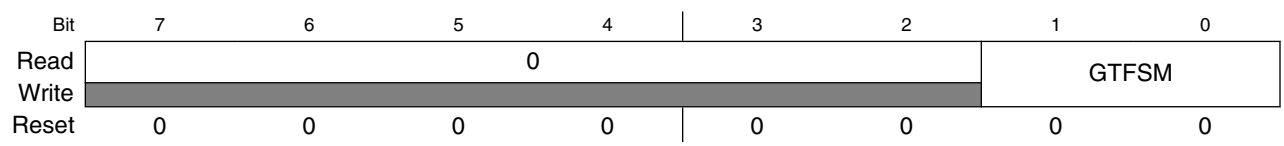
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30AC_0040	Semaphores Processor n IRQ Notification Enable (SEMA4_CP0INE)	16	R/W	0000h	4.4.6.2/155
30AC_0048	Semaphores Processor n IRQ Notification Enable (SEMA4_CP1INE)	16	R/W	0000h	4.4.6.2/155
30AC_0080	Semaphores Processor n IRQ Notification (SEMA4_CP0NTF)	16	R	0000h	4.4.6.3/157
30AC_0088	Semaphores Processor n IRQ Notification (SEMA4_CP1NTF)	16	R	0000h	4.4.6.3/157
30AC_0100	Semaphores (Secure) Reset Gate n (SEMA4_RSTGT)	16	R/W	0000h	4.4.6.4/159
30AC_0104	Semaphores (Secure) Reset IRQ Notification (SEMA4_RSTNTF)	16	R/W	0000h	4.4.6.5/161

4.4.6.1 Semaphores Gate n Register (SEMA4_Gate*n*)

Each semaphore gate is implemented in a 2-bit finite state machine (FSM), right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + 0h offset + (1d × i), where i=0d to 15d



SEMA4_Gate*n* field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine.

Table continues on the next page...

SEMA4_Gate*n* field descriptions (continued)

Field	Description
	Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.2 Semaphores Processor n IRQ Notification Enable (SEMA4_CP*n*INE)

The application of a hardware semaphore module provides an opportunity for implementation of helpful system-level features. An example is an optional mechanism to generate a processor interrupt after a failed lock attempt. Recall traditional software gate functions execute a spin-wait loop in an effort to obtain and lock the referenced gate. With this module, the processor that fails in the lock attempt could continue with other tasks and allow a properly-enabled notification interrupt to return its execution to the original lock function.

The optional notification interrupt function consists of two registers for each processor: an interrupt notification enable register (SEMA4_CPnINE) and the interrupt request register (SEMA4_CPnNTF). To support implementations with more than 16 gates, these registers can be referenced with aligned 16- or 32-bit accesses. For the SEMA4_CPnINE registers, unimplemented bits read as zeroes, and writes are ignored.

Address: 30AC_0000h base + 40h offset + (8d × i), where i=0d to 1d

Bit	15	14	13	12	11	10	9	8
Read Write	INE8	INE9	INE10	INE11	INE12	INE13	INE14	INE15
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	INE0	INE1	INE2	INE3	INE4	INE5	INE6	INE7
Reset	0	0	0	0	0	0	0	0

SEMA4_CP*n*INE field descriptions

Field	Description
15 INE8	Interrupt Request Notification Enable 8. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 8.

Table continues on the next page...

SEMA4_CP*n*INE field descriptions (continued)

Field	Description
	0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
14 INE9	Interrupt Request Notification Enable 9. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 9. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
13 INE10	Interrupt Request Notification Enable 10. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 10. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
12 INE11	Interrupt Request Notification Enable 11. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 11. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
11 INE12	Interrupt Request Notification Enable 12. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 12. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
10 INE13	Interrupt Request Notification Enable 13. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 13. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
9 INE14	Interrupt Request Notification Enable 14. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 14. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
8 INE15	Interrupt Request Notification Enable 15. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 15. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
7 INE0	Interrupt Request Notification Enable 0. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 0. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
6 INE1	Interrupt Request Notification Enable 1. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 1. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
5 INE2	Interrupt Request Notification Enable 2. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 2.

Table continues on the next page...

SEMA4_CP*n*INE field descriptions (continued)

Field	Description
	0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
4 INE3	Interrupt Request Notification Enable 3. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 3. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
3 INE4	Interrupt Request Notification Enable 4. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 4. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
2 INE5	Interrupt Request Notification Enable 5. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 5. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
1 INE6	Interrupt Request Notification Enable 6. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 6. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
0 INE7	Interrupt Request Notification Enable 7. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 7. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.

4.4.6.3 Semaphores Processor n IRQ Notification (SEMA4_CP*n*NTF)

The Semaphores module optionally allows the processor that fails in the lock attempt to continue with other tasks and allow a properly-enabled notification interrupt to return its execution to the original lock function rather than simply execute in a spin-wait loop.

The optional notification interrupt mechanism consists of two registers for each processor: an interrupt notification enable register (SEMA4_CPnINE) and the read-only notification interrupt request register (SEMA4_CPnNTF). To support implementations with more than 16 gates, these registers can be referenced with aligned 16- or 32-bit accesses. For the SEMA4_CPnNTF registers, unimplemented bits read as zeroes.

The notification interrupt is generated via a unique finite state machine (FSM), one per hardware gate. This machine operates in the following manner:

1. When an attempted lock fails, the FSM enters a first state where it waits until the gate is unlocked.

2. Once unlocked, the FSM enters a second state where it generates an interrupt request to the “failed lock” processor.
3. When the “failed lock” processor succeeds in locking the gate, the IRQ is automatically cleared and the FSM returns to the idle state. However, if the other processor again locks the gate, the FSM returns to the first state, clears the interrupt request, and then waits for the gate to be unlocked (again).

The notification interrupt request is implemented in a 3-bit, 5-state machine, where two specific states are encoded and program-visible as SEMA4_CP0NTF[GNn] and SEMA4_CP1NTF[GNn].

Address: 30AC_0000h base + 80h offset + (8d × i), where i=0d to 1d

Bit	15	14	13	12	11	10	9	8
Read	GN8	GN9	GN10	GN11	GN12	GN13	GN14	GN15
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	GN0	GN1	GN2	GN3	GN4	GN5	GN6	GN7
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_CP*n*NTF field descriptions

Field	Description
15 GN8	Gate 8 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 8. For more details, see SEMA4_CPnNTF Operation .
14 GN9	Gate 9 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 9. For more details, see SEMA4_CPnNTF Operation .
13 GN10	Gate 10 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 10. For more details, see SEMA4_CPnNTF Operation .
12 GN11	Gate 11 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 11. For more details, see SEMA4_CPnNTF Operation .
11 GN12	Gate 12 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 12. For more details, see SEMA4_CPnNTF Operation .
10 GN13	Gate 13 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 13. For more details, see SEMA4_CPnNTF Operation .
9 GN14	Gate 14 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 14. For more details, see SEMA4_CPnNTF Operation .
8 GN15	Gate 15 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 15. For more details, see SEMA4_CPnNTF Operation .
7 GN0	Gate 0 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 0. For more details, see SEMA4_CPnNTF Operation .
6 GN1	Gate 1 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 1. For more details, see SEMA4_CPnNTF Operation .
5 GN2	Gate 2 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 2. For more details, see SEMA4_CPnNTF Operation .

Table continues on the next page...

SEMA4_CP*n*NTF field descriptions (continued)

Field	Description
4 GN3	Gate 3 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 3. For more details, see SEMA4_CPnNTF Operation .
3 GN4	Gate 4 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 4. For more details, see SEMA4_CPnNTF Operation .
2 GN5	Gate 5 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 5. For more details, see SEMA4_CPnNTF Operation .
1 GN6	Gate 6 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 6. For more details, see SEMA4_CPnNTF Operation .
0 GN7	Gate 7 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 7. For more details, see SEMA4_CPnNTF Operation .

4.4.6.4 Semaphores (Secure) Reset Gate n (SEMA4_RSTGT)

Although the intent of the hardware gate implementation specifies a protocol where the locking processor must unlock the gate, it is recognized that system operation may require a reset function to re-initialize the state of any gate(s) without requiring a system-level reset.

To support this special gate reset requirement, the Semaphores module implements a "secure" reset mechanism which allows a hardware gate (or all the gates) to be initialized by following a specific dual-write access pattern. Using a technique similar to that required for the servicing of a software watchdog timer, the secure gate reset requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the specified gate(s). The required access pattern is:

1. A processor performs a 16-bit write to the SEMA4_RSTGT memory location. The least significant byte (SEMA4_RSTGT[RSTGDP]) must be 0xe2; the most significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the SEMA4_RSTGT location. For this write, the lower byte (SEMA4_RSTGT[RSTGDP]) is the logical complement of the first data pattern (0x1d) and the upper byte (SEMA4_RSTGT[RSTGTN]) specifies the gate(s) to be reset. This gate field can specify a single gate be cleared, or that all gates are cleared.
3. Reads of the SEMA4_RSTGT location return information on the 2-bit state machine (SEMA4_RSTGT[RSTGSM]) which implements this function, the bus master performing the reset (SEMA4_RSTGT[RSTGMS]) and the gate number(s) last cleared (SEMA4_RSTGT[RSTGTN]). Reads of the SEMA4_RSTGT register do not affect the secure reset finite state machine in any manner.

Address: 30AC_0000h base + 100h offset = 30AC_0100h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	RSTGTN								RSTGSM_RSTGMS_RSTGDP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SEMA4_RSTGT field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>15–8
RSTGTN</td><td>Reset Gate Number. This 8-bit field specifies the specific hardware gate to be reset. This field is updated by the second write.

This field contains the hexidecimal value of the gate number. If RSTGTN < 64, then reset the single gate defined by RSTGTN, else reset all the gates. The corresponding secure IRQ notification state machine(s) are also reset.</td></tr>
<tr><td>RSTGSM_
RSTGMS_
RSTGDP</td><td>NOTE: This field contains sub-fields that vary depending on whether it is being read or written. Sub-fields indicated as having read access are valid only for read operations. Sub-fields indicated as having write access are valid only for write operations. Bit numbering in the descriptions begins with the most significant bit numbered 0. See the following table for details.
<table>
<tr><th>Access</th><th>Sub-Field</th><th>Description</th></tr>
<tr><td rowspan="4">Read-Only</td><td>7-6
Reserved</td><td>Reserved. Always reads 0.</td></tr>
<tr><td>5-4
RSTGSM</td><td>Reset Gate Finite State Machine. Reads of the SEMA4_RSTGT register return the encoded state machine value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as:
00 Idle, waiting for the first data pattern write.
01 Waiting for the second data pattern write.
10 The 2-write sequence has completed. Generate the specified gate reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. Note that the RSTGSM = 0b10 state is valid for only a single machine cycle, so it is impossible for a read to return this value
11 This state encoding is never used and therefore reserved.</td></tr>
<tr><td>3
Reserved</td><td>Reserved. Always reads 0.</td></tr>
<tr><td>2-0
RSTGMS</td><td>Reset Gate Bus Master. This 3-bit read-only field records the logical number of the bus master performing the gate reset function. The reset function requires that the two consecutive writes to this register be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device and the logical processor number is SoC-specific. See the chip-specific information section.</td></tr>
<tr><td>Write-Only</td><td>7-0
RSTGDP</td><td>Reset Gate Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the gate reset mechanism. For the first write, RSTGDP = 0xe2 while the second write requires RSTGDP = 0x1d.</td></tr>
</table>
</td></tr>
</table>

4.4.6.5 Semaphores (Secure) Reset IRQ Notification (SEMA4_RSTNTF)

As with the case of the secure reset function of the hardware gates, it is recognized that system operation may require a reset function to re-initialize the state of the IRQ notification logic without requiring a system-level reset.

To support this special notification reset requirement, the Semaphores module implements a "secure" reset mechanism which allows an IRQ notification (or all the notifications) to be initialized by following a specific dual-write access pattern. When successful, the specified IRQ notification state machine(s) are reset. Using a technique similar to that required for the servicing of a software watchdog timer, the secure reset mechanism requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the IRQ notification(s). The required access pattern is:

1. A processor performs a 16-bit write to the SEMA4_RSTNTF memory location. The least significant byte (SEMA4_RSTNTF[RSTNDP]) must be 0x47; the most significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the SEMA4_RSTNTF location. For this write, the lower byte (SEMA4_RSTNTF[RSTNDP]) is the logical complement of the first data pattern (0xb8) and the upper byte (SEMA4_RSTNTF[RSTNTN]) specifies the notification(s) to be reset. This field can specify a single notification be cleared, or that all notifications are cleared.
3. Reads of the SEMA4_RSTNTF location return information on the 2-bit state machine (SEMA4_RSTNTF[RSTNSM]) which implements this function, the bus master performing the reset (SEMA4_RSTNTF[RSTNMS]) and the notification number(s) last cleared (SEMA4_RSTNTF[RSTNTN]). Reads of the SEMA4_RSTNTF register do not affect the secure reset finite state machine in any manner.

Address: 30AC_0000h base + 104h offset = 30AC_0104h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	RSTNTN								RSTNSM_RSTNMS_RSTNDP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SEMA4_RSTNTF field descriptions

Field	Description
15–8 RSTNTN	Reset Notification Number. This 8-bit field specifies the specific IRQ notification state machine to be reset. This field is updated by the second write. This field contains the hexidecimal value of the gate number. If RSTNTN < 64, then reset the single IRQ notification machine defined by RSTNTN, else reset all the notifications.

Table continues on the next page...

SEMA4_RSTNTF field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>RSTNSM_
RSTNMS_
RSTNDP</td><td>
NOTE: This field contains sub-fields that vary depending on whether it is being read or written. Sub-fields indicated as having read access are valid only for read operations. Sub-fields indicated as having write access are valid only for write operations. Bit numbering in the descriptions begins with the most significant bit numbered 0. See the following table for details.
<table>
<tr><th>Access</th><th>Sub-Field</th><th>Description</th></tr>
<tr><td rowspan="4">Read-Only</td><td>7-6
Reserved</td><td>Reserved. Always reads 0.</td></tr>
<tr><td>5-4
RSTNSM</td><td>Reset Notification Finite State Machine. Reads of the SEMA4_RSTNTF register return the encoded state machine value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as:
00 Idle, waiting for the first data pattern write.
01 Waiting for the second data pattern write.
10 The 2-write sequence has completed. Generate the specified notification reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. Note the RSTNSM = 10 state is valid for only a single machine cycle, so it is impossible for a read to return this value.
11 This state encoding is never used and therefore reserved..</td></tr>
<tr><td>3
Reserved</td><td>Reserved. Always reads 0.</td></tr>
<tr><td>2-0
RSTNMS</td><td>Reset Notification Bus Master. This 3-bit read-only field records the logical number of the bus master performing the notification reset function. The reset function requires that the two consecutive writes to this register be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device and the logical processor number is SoC-specific. See the chip-specific information section.</td></tr>
<tr><td>Write-Only</td><td>7-0
RSTNDP</td><td>Reset Notification Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the notification reset mechanism. For the first write, RSTNDP = 0x47 while the second write requires RSTNDP = 0xb8.</td></tr>
</table>
</td></tr>
</table>

4.5 On-Chip RAM Memory Controller (OCRAM)

4.5.1 Overview

The OCRAM block is implemented as an slave module on the 64-bit system AXI bus. Designed as a simple on-chip memory controller, it supports only one AXI port with memory banks. For the AXI port, the read and write transactions are handled by two

independent modules. As it is possible to have simultaneous read and write request from the AXI bus, each memory bank has an arbiter with round-robin scheme. After arbitration, the granted read or write access command can then be issued to the memory cell through a read/write MUX.

The memory banks are organized with the lower 2 bits of the address which is the AXI bus address and is 64 bits aligned interleaved. This allows a read access and a write access to be processed at the same time if they are targeted to different memory banks.

Various options are provided for adding a pipeline in a read/write access, in order to ensure flexible timing control at both high and low frequencies.

4.5.1.1 Block diagram

The OCRAM block diagram is shown in the figure below.

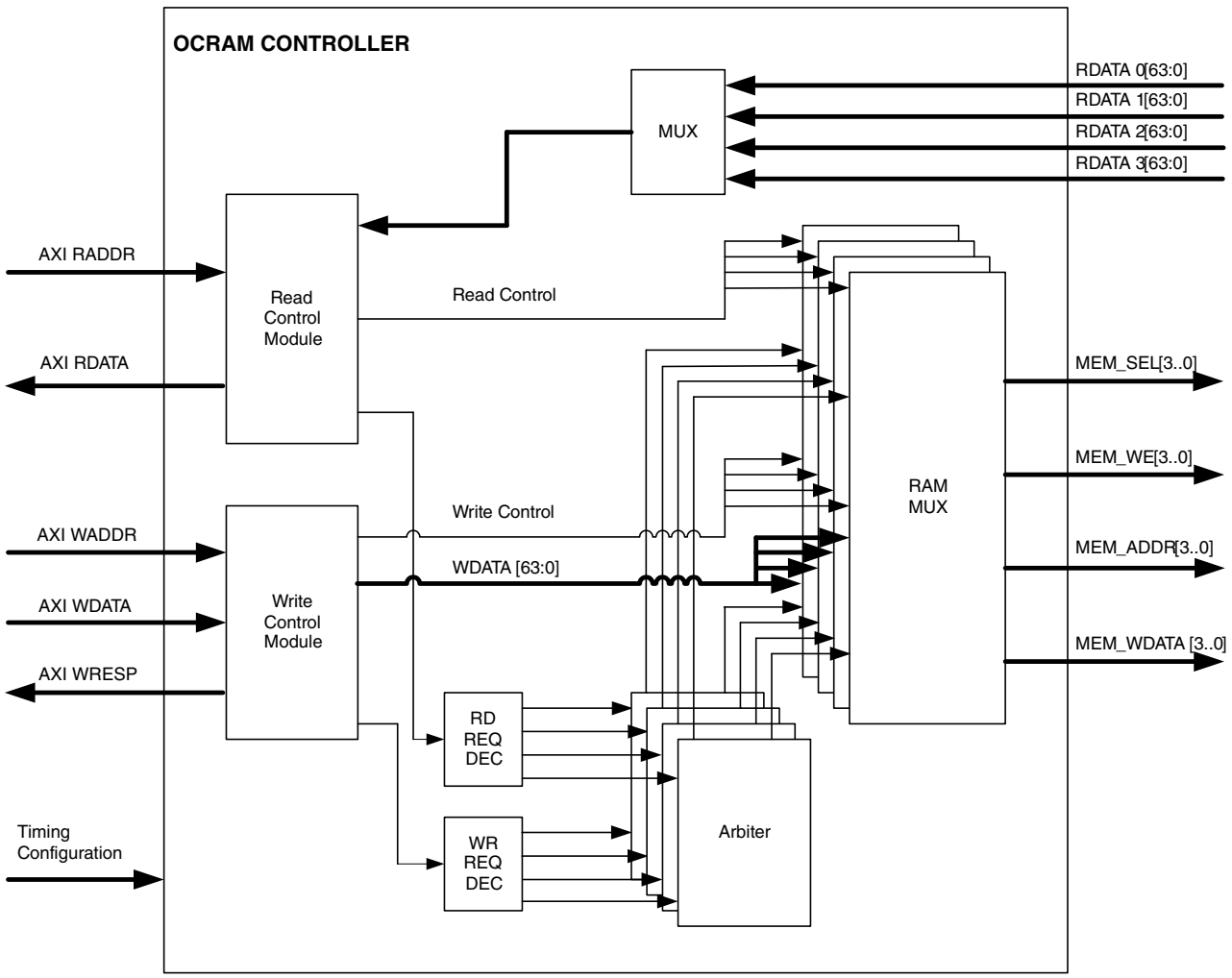


Figure 4-11. OCRAM Block Diagram

4.5.2 Functional Description

4.5.2.1 Read/Write Arbitration

The detailed rules used in arbitration are as follows:

- If there is no granted read or write in the last cycle, and there is only a read request or a write request, the request will be granted.
- If there is no granted read or write in the last cycle, and there are both read or write requests coming in at the same time, the read request will be granted first.

- If a granted read/write transaction has just finished, the write/read request will have the higher priority in the next cycle.
- If the first read/write access request in a transaction is granted, all the data transfer in this burst will be finished before the next arbitration begins, that is, the round-robin arbitration mechanism is based on AXI transaction, not data access.

4.5.3 Programmable Registers

There are no programmable registers in this block. However, OCRAM configurable bits can be found in the IOMUX Controller (IOMUXC) general purpose registers as below.

- TrustZone bits: IOMUXC_GPR_GPR10 and IOMUXC_GPR_GPR11

4.6 Network Interconnect Bus System (NIC)

4.6.1 Overview

This section provides an overview of the NIC-301 (Network Inter-Connect) AXI arbiter IP.on chip Network Interconnection Bus System. The Bus System is composed of two kinds of IP

The NIC-301 (by Arm Ltd.) is a configurable AXI arbiter between several masters and slaves. The NIC-301 IP is designed so that many configuration options are selected at the hardware design stage, determined by SoC characteristics and needs, while several other configuration options are software-controlled.

- NIC-301 (Network Inter-Connect) AXI arbiter IP

  The NIC-301 (by Arm Ltd.) is a configurable AXI arbiter between several masters and slaves. The NIC-301 IP is designed so that many configuration options are selected at the hardware design stage, determined by SoC characteristics and needs, while several other configuration options are software-controlled.
- NoC (Network on Chip) fabric IP

  The NoC (by Arteris Ltd.) is a configurable high efficiency and performance fabric IP. Similar to the NIC-301 IP the majority design options are configured during hardware design stage. Mean while those application specific configurations are required to be configured by software.

NOTE

The NIC-301 and NoC default settings are configured by NXP's board support package (BSP), and in most cases should not be

modified by the customer. The default settings have gone through exhaustive testing during the validation of the part, and have proven to work well for the part's intended target applications. Changes to the default settings may result in a degradation in system performance.

4.7 AHB to IP Bridge (AIPSTZ)

4.7.1 Overview

This section provides an overview of the AHB to IP Bridge (AIPSTZ). This particular peripheral is designed as the bridge between AHB bus and peripherals with the lower bandwidth IP Slave (IPS) buses.

4.7.1.1 Features

The following list summarizes the key features of the bridge:

- The bridge supports the IPS slave bus signals.
- The bridge supports 8-, 16-, and 32-bit IPS peripherals. (Accesses larger than the size of a peripheral are not supported, except to 32-bit memory.)
- The bridge supports a pair of IPS accesses for 64-bit and certain misaligned AHB transfers to 32-bit memory in 64-bit platforms.
- The bridge directly supports up to 32 64-Kbyte external IPS peripherals, and 2 global external IPS peripheral spaces. The bridge occupies 1 MBytes of total address space.
- The bridge provides configurable per-block and per-master access protections. Access permissions are based on bus master (e.g. DMA or core) privilege levels and resource domain. More details on the protection features and configuration can be found in the Security Reference Manual
- Peripheral read transactions require a minimum of 2 hclk clocks, and unbuffered write transactions require a minimum of 3 hclk clocks.
- The bridge uses one single asynchronous reset and one global clock.

4.7.2 Clocks

The following table describes the clock sources for AIPSTZ. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 4-14. AIPSTZ Clocks

Clock name	Clock Root	Description
hclk	ahb_clk_root	Module clock

4.7.3 Functional Description

The AIPS bridge serves as a protocol translator between the AHB system bus and the IP bus.

Support is provided for generating a pair of 32-bit IP bus accesses when targeted by a 64-bit system bus access, or a misaligned access which crosses a 32-bit boundary. No other bus-sizing access support is provided.

The AHB to IP bridge is the interface between the AHB and on-chip IPS peripherals, which are sub-blocks containing readable/writable control and status registers.

The AHB master reads and writes these registers through the AIPSTZ. The bridge generates block enables, the block address, transfer attributes, byte enables and write data as inputs to the IPS peripherals. The bridge captures read data from the IPS interface and drives it on the AHB.

Each bridge that connects to the IPS (or peripherals) are referred as AIPS. The chip has several separate AIPS modules, and peripherals are grouped and assigned under each AIPS block. The list of peripherals are indicated as n-1, ... and n-*x* for AIPS-1, ... and AIPS-*x* respectively.

AIPS occupies a 1-Mbyte portion of the address space. The register maps of the IPS peripherals are located on 64-Kbyte boundaries. Each IPS peripheral is allocated one 64-Kbyte block of the memory map, and is activated by one of the block enables from the bridge. Up to thirty-two 64-Kbyte external IPS peripherals may be implemented, occupying contiguous blocks of 64-Kbytes. Two global external IPS block enables are available for the remaining address space to allow for customization and expansion of addressed peripheral devices. In addition, a single "non-global" block enable is also asserted whenever any of the thirty-two non-global block enables is asserted.

The bridge is responsible for indicating to IPS peripherals if an access is in supervisor or user mode. It may block user mode accesses to certain IPS peripherals or it may allow the individual IPS peripherals to determine if user mode accesses are allowed. In addition, peripherals may be designated as write-protected.

The bridge supports the notion of "trusted" masters for security purposes. Masters may be individually designated as trusted for reads, trusted for writes, or trusted for both reads and writes, as well as being forced to look as though all accesses from a master are in user-mode privilege level. Refer to AIPSTZ Memory Map/Register Definition for more information.

The AIPSTZ prevents access to a peripheral if the transaction originated from a source from a resource domain that has been explicitly omitted. Resource domains are assigned in the RDC submodule. Please refer to the RDC chapter for programming details.

All peripheral devices are expected to only require aligned accesses equal to or smaller in size than the peripheral size. An exception to this rule is supported for 32-bit peripherals to allow memory to be placed on the IPS.

4.7.4 Access Protections

The AIPSTZ bridge provides programmable access protections for both masters and peripherals. It allows the privilege level of a master to be overridden, forcing it to user-mode privilege, and allows masters to be designated as trusted or untrusted.

Peripherals may require supervisor privilege level for access, may restrict access to a trusted master only, and may be write-protected. IP bus peripherals are subject to access control policies set in both CSU registers and AIPSTZ registers. An access is blocked if it is denied by either policy.

Masters and peripherals are assigned to one or more resource domains in the RDC submodule (see the RDC chapter for details). Depending on RDC programming, masters transactions through the AIPSTZ may or may not be allowed access to peripherals in different resource domains.

4.7.5 Access Support

Aligned 64-bit accesses, aligned and misaligned word and half word accesses, as well as byte accesses are supported for 32-bit peripherals. Misaligned accesses are supported to allow memory to be placed on the IPS.

Peripheral registers must not be misaligned, although no explicit checking is performed by the AIPS bridge. The bridge will perform two IPS transfers for 64-bit accesses, word accesses with byte offsets of 1, 2, or 3, and for half word accesses with a byte offset of 3. All other accesses will be performed with a single IPS transfer.

Only aligned half word and byte accesses are supported for 16-bit peripherals. All other accesses types are unsupported, and results of such accesses are undefined. They are not terminated with an error response.

Only byte accesses are supported for 8-bit peripherals. All other accesses types are unsupported, and results of such accesses are undefined. They are not terminated with an error response.

4.7.6 Initialization Information

The AIPS bridge should be programmed before use.

The following registers should be initialized: The Master Privilege Registers (AIPSTZ_MPRs), the Peripheral Access Control registers (AIPSTZ_PACRs), and the Off-platform Peripheral Access Control registers (AIPSTZ_OPACRs) described in AIPSTZ Memory Map/Register Definition.

4.7.6.1 Security Block

The AIPSTZ contains a security block that is connected to each off-platform peripheral. This block filters accesses based on write/read, non-secure, and supervisor signals.

Each peripheral can be individually configured to allow or deny each of the following transactions as described in the table below:

Table 4-15. Peripheral Access Configuration options

Config Bit	Write	Non-Secure	Supervisor	Meaning
0	0	0	0	Secure User Read
1	0	0	1	Secure Supervisor Read
2	0	1	0	Non-Secure User Read
3	0	1	1	Non-Secure Supervisor Read
4	1	0	0	Secure User Write
5	1	0	1	Secure Supervisor Write
6	1	1	0	Non-Secure User Write
7	1	1	1	Non-Secure Supervisor Write

Each peripheral has a security configuration (sec_config_X) input for determining whether to allow or deny a given access type. These are 8-bit vectors, with each bit corresponding to one of the transactions above as listed in the Config Bit column of Table 4-15. If the bit is asserted (1'b1), the transaction is allowed. If the bit is negated (1'b0), the transaction is not allowed.

For example, if peripheral 0 is configured as follows:

sec_config_0 [7:0] = 8'b0011_0011

This peripheral can only be accessed by secure transactions. Bits 0, 1, 4, and 5 are asserted and these bits refer to the four types of secure transactions. If an insecure transaction is attempted to this peripheral, it will result in an error.

Eight bits per peripheral across an entire system can result in a large number of configuration bits that must be assigned and controlled, most likely in a series of registers in another block. To reduce the number of register bits required predefined sets of security profiles can be defined and encapsulated in an external security translation block. The table below describes one set of security profiles that has been proposed for use with the AIPSTZ.

Table 4-16. Security Levels

CSU_SEC_LEVEL	Non-Secure User	Non-Secure Supervisor	Secure User	Secure Supervisor
0	RD+WR	RD+WR	RD+WR	RD+WR
1	NOT ALLOWED	RD+WR	RD+WR	RD+WR
2	Read Only	Read Only	RD+WR	RD+WR
3	NOT ALLOWED	Read Only	RD+WR	RD+WR
4	NOT ALLOWED	NOT ALLOWED	RD+WR	RD+WR
5	NOT ALLOWED	NOT ALLOWED	NOT ALLOWED	RD+WR
6	NOT ALLOWED	NOT ALLOWED	Read Only	Read Only
7	NOT ALLOWED	NOT ALLOWED	NOT ALLOWED	NOT ALLOWED

Information regarding CSU is provided in the Security Reference Manual. Contact your NXP representative for information about obtaining this document.

A 3-bit input, 8-bit output translation block can be used such that only three register bits are required to set the security profile and the translation block will drive the correct 8-bit configuration vector. Each peripheral connected to the AIPSTZ would require this translation block. The top level AIPSTZ has this three bit input line `csu_sec_level[2:0]' corresponding to each peripheral X.

4.7.7 Off-Platform Peripherals Index

The off-platform peripherals index allocation is shown in the following table.

Table 4-17. External IPS Peripherals

OPAC ID	AIPSTZ1	AIPSTZ2	AIPSTZ3	AIPSTZ4	AIPSTZ5
OPAC31	Reserved	QoSC	ENET2_TSN	Reserved	Reserved
OPAC30	CSU	Reserved	ENET1	NOC BLK_CTL	Reserved
OPAC29	RDC	PERFMON2	SDMA	HDMI_TX	Reserved
OPAC28	SEMAPHORE2	PERFMON1	Reserved		Reserved
OPAC27	SEMAPHORE1	Reserved	QSPI	Reserved	Reserved
OPAC26	GPC	Reserved	Reserved	Reserved	Reserved
OPAC25	SRC	Reserved	Reserved	Reserved	Reserved
OPAC24	CCM	Reserved	Reserved	TZASC	Reserved
OPAC23	SNVS_HP	Reserved	Reserved	Reserved	Reserved
OPAC22	ANA_PLL	Reserved	uSDHC3	Reserved	Reserved
OPAC21	OCOTP_CTRL	Reserved	uSDHC2	Reserved	Reserved
OPAC20	IOMUXC_GPR	Reserved	uSDHC1	Reserved	Reserved
OPAC19	IOMUXC	Reserved	Reserved	Reserved	Reserved
OPAC18	Reserved	Reserved	Reserved	Reserved	Reserved
OPAC17	Reserved	Reserved	Reserved	HSIOMIX BLK_CTL	Reserved
OPAC16	Reserved	GPT4	Reserved	PCIE_PHY1	Reserved
OPAC15	GPT3	GPT5	Reserved	Reserved	Reserved
OPAC14	GPT2	GPT6	I2C6	Reserved	Reserved
OPAC13	GPT1	Reserved	I2C5	Reserved	Reserved
OPAC12	OCRAM_S	System Counter_CTRL	SEMAPHORE_HS	MEDIAMIX BLK_CTL	Reserved
OPAC11	OCRAM	System Counter_CMP	MU_1_B (A53, M7)	LVDS2	Reserved
OPAC10	WDOG3	System Counter_RD	MU_1_A (A53,M7)	LVDS1	Reserved
OPAC9	WDOG2	PWM4	Reserved	LCDIF2	MU_3_B (M7, Audio Processor)
OPAC8	WDOG1	PWM3	IRQ_STEER (Audio Processor)	LCDIF1	MU_3_A (M7, Audio Processor)
OPAC7	ANA_OSC	PWM2	Reserved	Reserved	MU_2_B (A53, Audio Processor
OPAC6	ANA_TSENSOR	PWM1	UART4	MIPI_DSI1	MU_2_A (A53, Audio Processor)
OPAC5	Reserved	Reserved	I2C4	MIPI_CSI2	eDMA Channels[31:16] at 4KB/ch

Table continues on the next page...

Table 4-17. External IPS Peripherals (continued)

OPAC ID	AIPSTZ1	AIPSTZ2	AIPSTZ3	AIPSTZ4	AIPSTZ5
OPAC4	GPIO5	Reserved	I2C3	MIPI_CSI1	eDMA Channels[15:0] at 4KB/ch
OPAC3	GPIO4	Reserved	I2C2	IPS Dewarp	eDMA Management Page
OPAC2	GPIO3	Reserved	I2C1	ISP2	AUDIO BLK_CTRL
OPAC1	GPIO2	Reserved	Reserved	ISP1	SDMA2
OPAC0	GPIO1	Reserved	Reserved	ISI	SDMA3
Global external IPS peripheral					
OPAC ID	**AIPSTZ1**	**AIPSTZ2**	**AIPSTZ3**	**AIPSTZ4**	**AIPSTZ5**
OPAC33	Reserved	Reserved	CAAM	Reserved	Reserved
OPAC32	Reserved	Reserved	SPBA1: CAN_FD2, CAN_FD1, UART2, UART3, UART1, eCSPI3, eCSPI2, eCSPI1.	Reserved	SPBA2: AUDIO XCVR RX (eARC), HDMI TX AUDLNK MSTR, PDM (MICFIL), ASRC, SAI7, SAI6, SAI5, SAI3, SAI2, SAI1.

4.7.8 AIPSTZ Memory Map/Register Definition

The memory map for the AIPS SW-visible registers is shown in the table below.

The MPROT and OPACR fields are 4 bits in width. Some bits may be reserved depending on device.

AIPSTZ memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3000_0000	Master Priviledge Registers (AIPSTZ1_MPR)	32	R/W	7700_0000h	4.7.8.1/174
3000_0040	Off-Platform Peripheral Access Control Registers (AIPSTZ1_OPACR)	32	R/W	4444_4444h	4.7.8.2/176
3000_0044	Off-Platform Peripheral Access Control Registers (AIPSTZ1_OPACR1)	32	R/W	4444_4444h	4.7.8.3/179
3000_0048	Off-Platform Peripheral Access Control Registers (AIPSTZ1_OPACR2)	32	R/W	4444_4444h	4.7.8.4/182
3000_004C	Off-Platform Peripheral Access Control Registers (AIPSTZ1_OPACR3)	32	R/W	4444_4444h	4.7.8.5/185
3000_0050	Off-Platform Peripheral Access Control Registers (AIPSTZ1_OPACR4)	32	R/W	4444_4444h	4.7.8.6/188

AIPSTZ memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3040_0000	Master Priviledge Registers (AIPSTZ2_MPR)	32	R/W	7700_0000h	4.7.8.1/174
3040_0040	Off-Platform Peripheral Access Control Registers (AIPSTZ2_OPACR)	32	R/W	4444_4444h	4.7.8.2/176
3040_0044	Off-Platform Peripheral Access Control Registers (AIPSTZ2_OPACR1)	32	R/W	4444_4444h	4.7.8.3/179
3040_0048	Off-Platform Peripheral Access Control Registers (AIPSTZ2_OPACR2)	32	R/W	4444_4444h	4.7.8.4/182
3040_004C	Off-Platform Peripheral Access Control Registers (AIPSTZ2_OPACR3)	32	R/W	4444_4444h	4.7.8.5/185
3040_0050	Off-Platform Peripheral Access Control Registers (AIPSTZ2_OPACR4)	32	R/W	4444_4444h	4.7.8.6/188
3080_0000	Master Priviledge Registers (AIPSTZ3_MPR)	32	R/W	7700_0000h	4.7.8.1/174
3080_0040	Off-Platform Peripheral Access Control Registers (AIPSTZ3_OPACR)	32	R/W	4444_4444h	4.7.8.2/176
3080_0044	Off-Platform Peripheral Access Control Registers (AIPSTZ3_OPACR1)	32	R/W	4444_4444h	4.7.8.3/179
3080_0048	Off-Platform Peripheral Access Control Registers (AIPSTZ3_OPACR2)	32	R/W	4444_4444h	4.7.8.4/182
3080_004C	Off-Platform Peripheral Access Control Registers (AIPSTZ3_OPACR3)	32	R/W	4444_4444h	4.7.8.5/185
3080_0050	Off-Platform Peripheral Access Control Registers (AIPSTZ3_OPACR4)	32	R/W	4444_4444h	4.7.8.6/188
30C0_0000	Master Priviledge Registers (AIPSTZ5_MPR)	32	R/W	7700_0000h	4.7.8.1/174
30C0_0040	Off-Platform Peripheral Access Control Registers (AIPSTZ5_OPACR)	32	R/W	4444_4444h	4.7.8.2/176
30C0_0044	Off-Platform Peripheral Access Control Registers (AIPSTZ5_OPACR1)	32	R/W	4444_4444h	4.7.8.3/179
30C0_0048	Off-Platform Peripheral Access Control Registers (AIPSTZ5_OPACR2)	32	R/W	4444_4444h	4.7.8.4/182
30C0_004C	Off-Platform Peripheral Access Control Registers (AIPSTZ5_OPACR3)	32	R/W	4444_4444h	4.7.8.5/185
30C0_0050	Off-Platform Peripheral Access Control Registers (AIPSTZ5_OPACR4)	32	R/W	4444_4444h	4.7.8.6/188
32C0_0000	Master Priviledge Registers (AIPSTZ4_MPR)	32	R/W	7700_0000h	4.7.8.1/174
32C0_0040	Off-Platform Peripheral Access Control Registers (AIPSTZ4_OPACR)	32	R/W	4444_4444h	4.7.8.2/176
32C0_0044	Off-Platform Peripheral Access Control Registers (AIPSTZ4_OPACR1)	32	R/W	4444_4444h	4.7.8.3/179
32C0_0048	Off-Platform Peripheral Access Control Registers (AIPSTZ4_OPACR2)	32	R/W	4444_4444h	4.7.8.4/182
32C0_004C	Off-Platform Peripheral Access Control Registers (AIPSTZ4_OPACR3)	32	R/W	4444_4444h	4.7.8.5/185
32C0_0050	Off-Platform Peripheral Access Control Registers (AIPSTZ4_OPACR4)	32	R/W	4444_4444h	4.7.8.6/188

4.7.8.1 Master Priviledge Registers (AIPSTZx_MPR)

Each AIPSTZ_MPR specifies 16 4-bit fields defining the access privilege level associated with a bus master in the platform, as well as specifying whether write accesses from this master are bufferable shown in Table 4-18

The registers provide one field per bus master, where field 15 corresponds to master 15, field 14 to master 14,... field 0 to master 0 (typically the processor core). The master index allocation is shown in the table below.

Table 4-18. MPROT Field

Bit	Field	Description
3	MBW	**Master Buffer Writes** - This bit determines whether the AIPSTZ is enabled to buffer writes from this master.
2	MTR	**Master Trusted for Reads** - This bit determines whether the master is trusted for read accesses.
1	MTW	**Master Trusted for Writes** - This bit determines whether the master is trusted for write accesses.
0	MPL	**Master Privilege Level** - This bit determines how the privilege level of the master is determined.

NOTE

The reset value is set to 0000_0000_7700_0000, which makes master 0 and master 1 (Arm CORE) the trusted masters. Trusted software can change the settings after reset.

Table 4-19. Master Index Allocation

Master Index	Master Name	Comments
Master 0	All masters excluding Arm core	Share the same number allocation.
Master 1	Arm A53 core	
Master 3	SDMA	
Master 5	Arm M7 core	
Master 6	HIFI4 Audio Processor	

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MPROT0				MPROT1				MPROT2				MPROT3				Reserved				MPROT5				Reserved				Reserved			
Reset	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AIPSTZx_MPR field descriptions

Field	Description
31–28 MPROT0	Master 0 Priviledge, Buffer, Read, Write Control xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses. x1xx **MTR1** — This master is trusted for read accesses. 0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered
27–24 MPROT1	Master 1 Priviledge, Buffer, Read, Write Control xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses. x1xx **MTR1** — This master is trusted for read accesses. 0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered
23–20 MPROT2	Master 2 Priviledge, Buffer, Read, Write Control xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses. x1xx **MTR1** — This master is trusted for read accesses. 0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered
19–16 MPROT3	Master 3 Priviledge, Buffer, Read, Write Control. xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses. x1xx **MTR1** — This master is trusted for read accesses. 0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered

Table continues on the next page...

AIPSTZx_MPR field descriptions (continued)

Field	Description
15–12 -	This field is reserved. Reserved
11–8 MPROT5	Master 5 Priviledge, Buffer, Read, Write Control. xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses. x1xx **MTR1** — This master is trusted for read accesses. 0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered
7–4 -	This field is reserved. Reserved
-	This field is reserved. Reserved

4.7.8.2 Off-Platform Peripheral Access Control Registers (AIPSTZx_OPACR)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 4-20

Table 4-20. OPAC Field

Bit	Field	Description
3	BW	**Buffer Writes** - This bit determines whether write accesses to this peripheral are allowed to be buffered.[1]
2	SP	**Supervisor Protect** - This bit determines whether the peripheral requires supervisor privilege level for access.
1	WP	**Write Protect** - This bit determines whether the peripheral allows write accesses.
0	TP	**Trusted Protect** - This bit determines whether the peripheral allows accesses from an untrusted master.

1. Buffered writes are not available for AIPSTZ. This bit should be set to '0'.

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC0				OPAC1				OPAC2				OPAC3				OPAC4				OPAC5				OPAC6				OPAC7			
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZx_OPACR field descriptions

Field	Description
31–28 OPAC0	Off-platform Peripheral Access Control 0 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC1	Off-platform Peripheral Access Control 1 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
23–20 OPAC2	Off-platform Peripheral Access Control 2 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
19–16 OPAC3	Off-platform Peripheral Access Control 3

Table continues on the next page...

AIPSTZx_OPACR field descriptions (continued)

Field	Description
	xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
15–12 OPAC4	Off-platform Peripheral Access Control 4 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
11–8 OPAC5	Off-platform Peripheral Access Control 5 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
7–4 OPAC6	Off-platform Peripheral Access Control 6 xxx0 **TP0** — Accesses from an untrusted master are allowed.

Table continues on the next page...

AIPSTZx_OPACR field descriptions (continued)

Field	Description
	xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
OPAC7	Off-platform Peripheral Access Control 7 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

4.7.8.3 Off-Platform Peripheral Access Control Registers (AIPSTZx_OPACR1)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 4-20

Address: Base address + 44h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC8				OPAC9				OPAC10				OPAC11				OPAC12				OPAC13				OPAC14				OPAC15			
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZx_OPACR1 field descriptions

Field	Description
31–28 OPAC8	Off-platform Peripheral Access Control 8 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC9	Off-platform Peripheral Access Control 9 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
23–20 OPAC10	Off-platform Peripheral Access Control 10 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
19–16 OPAC11	Off-platform Peripheral Access Control 11

Table continues on the next page...

AIPSTZx_OPACR1 field descriptions (continued)

Field	Description
	xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
15–12 OPAC12	Off-platform Peripheral Access Control 12 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
11–8 OPAC13	Off-platform Peripheral Access Control 13 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
7–4 OPAC14	Off-platform Peripheral Access Control 14 xxx0 **TP0** — Accesses from an untrusted master are allowed.

Table continues on the next page...

AIPSTZx_OPACR1 field descriptions (continued)

Field	Description
	xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
OPAC15	Off-platform Peripheral Access Control 15 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

4.7.8.4 Off-Platform Peripheral Access Control Registers (AIPSTZx_OPACR2)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 4-20

Address: Base address + 48h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC16				OPAC17				OPAC18				OPAC19				OPAC20				OPAC21				OPAC22				OPAC23			
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZx_OPACR2 field descriptions

Field	Description
31–28 OPAC16	Off-platform Peripheral Access Control 16 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC17	Off-platform Peripheral Access Control 17 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
23–20 OPAC18	Off-platform Peripheral Access Control 18 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
19–16 OPAC19	Off-platform Peripheral Access Control 19

Table continues on the next page...

AIPSTZx_OPACR2 field descriptions (continued)

Field	Description
	xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
15–12 OPAC20	Off-platform Peripheral Access Control 20 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
11–8 OPAC21	Off-platform Peripheral Access Control 21 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
7–4 OPAC22	Off-platform Peripheral Access Control 22 xxx0 **TP0** — Accesses from an untrusted master are allowed.

Table continues on the next page...

AIPSTZx_OPACR2 field descriptions (continued)

Field	Description
	xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
OPAC23	Off-platform Peripheral Access Control 23 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

4.7.8.5 Off-Platform Peripheral Access Control Registers (AIPSTZx_OPACR3)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 4-20

Address: Base address + 4Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC24				OPAC25				OPAC26				OPAC27				OPAC28				OPAC29				OPAC30				OPAC31			
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZx_OPACR3 field descriptions

Field	Description
31–28 OPAC24	Off-platform Peripheral Access Control 24 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC25	Off-platform Peripheral Access Control 25 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
23–20 OPAC26	Off-platform Peripheral Access Control 26 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
19–16 OPAC27	Off-platform Peripheral Access Control 27

Table continues on the next page...

AIPSTZx_OPACR3 field descriptions (continued)

Field	Description
	xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
15–12 OPAC28	Off-platform Peripheral Access Control 28 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
11–8 OPAC29	Off-platform Peripheral Access Control 29 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
7–4 OPAC30	Off-platform Peripheral Access Control 30 xxx0 **TP0** — Accesses from an untrusted master are allowed.

Table continues on the next page...

AIPSTZx_OPACR3 field descriptions (continued)

Field	Description
	xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
OPAC31	Off-platform Peripheral Access Control 31 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

4.7.8.6 Off-Platform Peripheral Access Control Registers (AIPSTZx_OPACR4)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 4-20

Address: Base address + 50h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC32				OPAC33				Reserved																							
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZx_OPACR4 field descriptions

Field	Description
31–28 OPAC32	Off-platform Peripheral Access Control 32 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC33	Off-platform Peripheral Access Control 33 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
-	This field is reserved. Reserved

4.8 Shared Peripheral Bus Arbiter (SPBA)

4.8.1 Overview

The Shared Peripheral Bus Arbiter (SPBA) is a three-to-one IP Bus interface arbiter. Three masters arbitrate for shared peripheral access through the SPBA.

The SPBA has three primary functions:

- The IP Bus Line switches a master to one peripheral

- The Masters arbiter arbitrates between the three masters to solve concurrent access or restricted access to peripherals
- The Control Registers and Ownership Control includes a set of registers which are reachable through software and permit the access scheme to be defined for each peripheral (Resource Ownership and Access Control). It generates signals for the external steering logic of interrupts and DMA signals.

4.8.1.1 Block Diagram

The figure below shows the SPBA block diagram

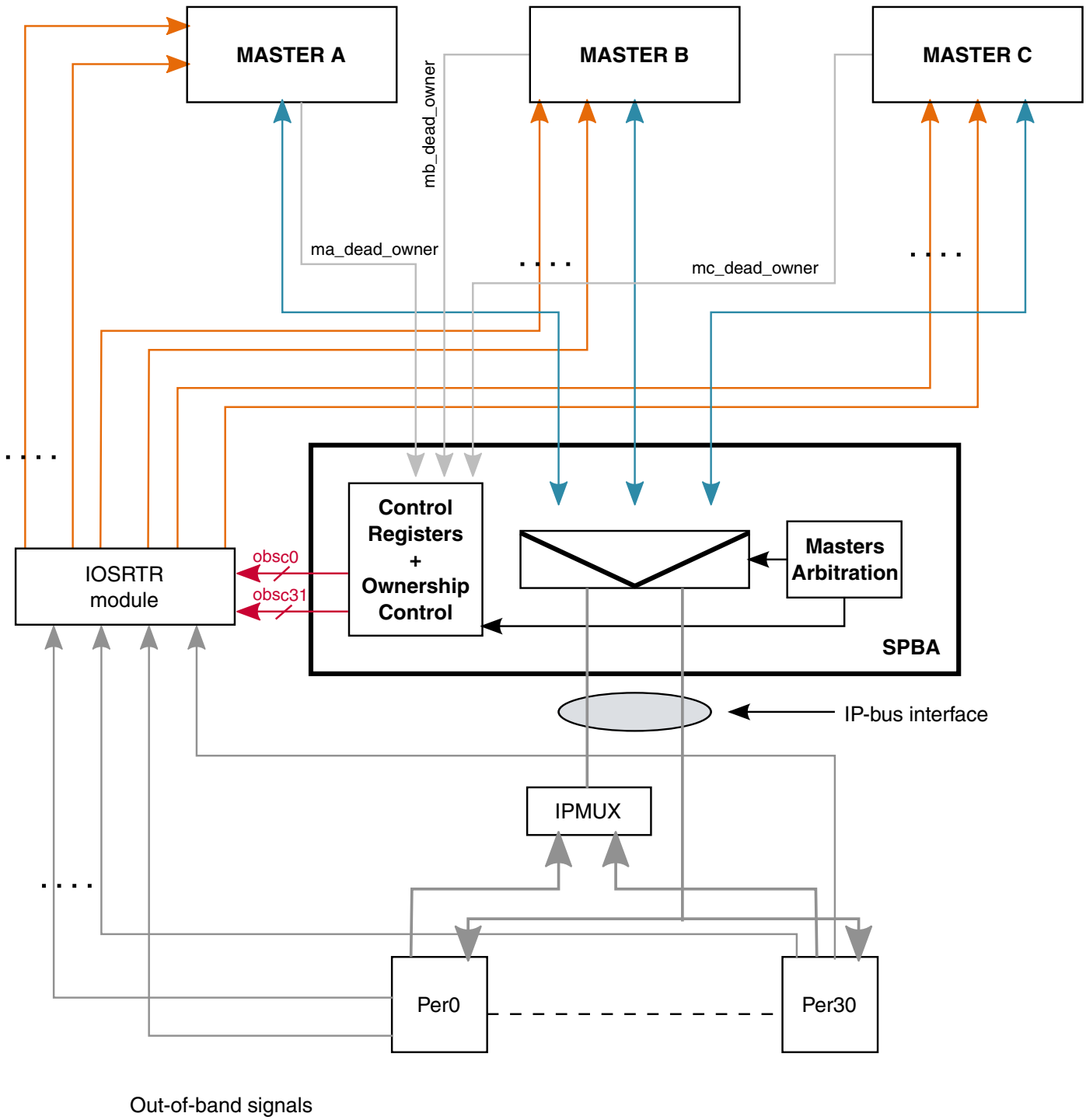


Figure 4-12. SPBA Block Diagram

4.8.1.2 Features

The SPBA includes the following features:

- Three IP Bus masters arbitration: Master A, B and C
- Support for DMA masters
- 32-bit data
- Supports up to 32 shared peripherals, each consuming 64 kilobytes of address space
- SPBA can be considered the 33rd peripheral, used for resource ownership and access control of the 31 peripherals
- Provides 31 sets of out of band steering control (OBSC) signals to the off-block steering logic
- Operating frequency up to 67 MHz
- Clocks: ipg_clk, ipg_clk_s

4.8.2 Functional description

4.8.2.1 Modes of operation

SPBA behavior is transparent when accessing a peripheral, though it has these distinct modes of operation.

Reset/Abort

The SPBA has a hardware reset which initializes all registers, arbitration and peripherals rights registers (PRRs).

An abort signal input is provided allowing each master to abort its current access and release ownership (in case of master reset sequence).

Functional

Once a master request is granted, its IP Bus signals are steered to the requested peripheral.

Standby

No clock needed. The SPBA needs clocks only during access to the PRRs, arbitration, and abort phases. It generates two clock enable signals indicating when the clocks must be provided.

Configuration

During this phase, a master accesses the SPBA PRRs. The SPBA memory-mapped registers are seen as a shared peripheral.

4.8.2.2 Masters arbitration

The arbitration mechanism determines which port will control the master port, based on a simple round-robin arbitration scheme.

There are several use cases to consider.

- Only one master request per access. The master is switched to the shared peripheral bus, without arbitration. Figure 4-13 shows the MB request on the global module enable signal, served without wait state.
- If two masters simultaneously access SPBA, the last granted master is held off using the <master>_ips_xfr_wait output signal (default value is high). When the master is granted sips_xfr_wait, shared IP Bus peripheral is connected to <master>_ips_xfr_wait outputs.
- If three masters simultaneously access SPBA, then the last two granted masters are held off using <master>_ips_xfr_wait. Figure 4-14 shows a case in which the last two accesses granted are MA and MB. The requests are used even if they are in the same cycle.
- If after reset, at the first multiple access, no master has been granted, the priority is static: Master A (MA), Master B (MB) and last Master C (MC) port.
- No master request. No master switch to shared peripherals.

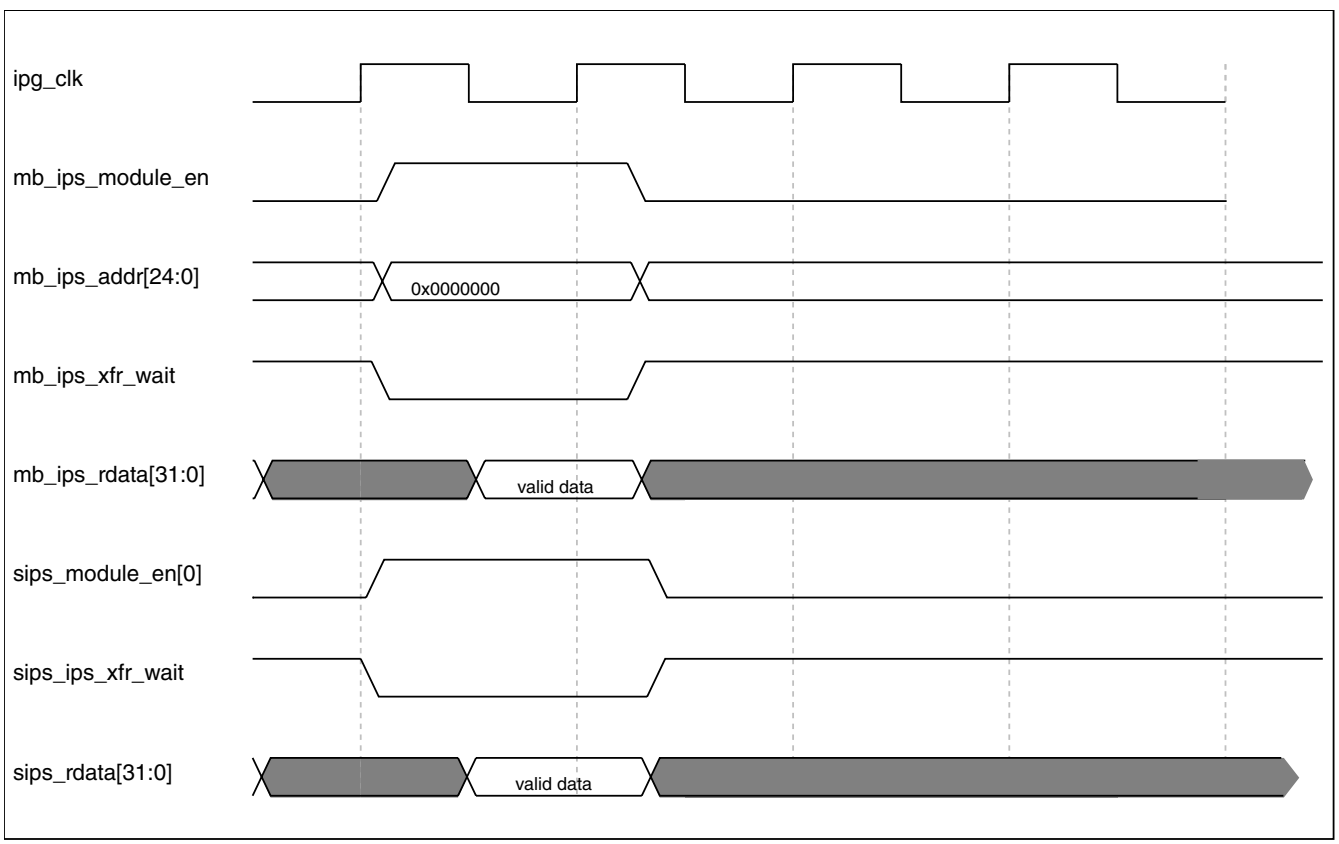


Figure 4-13. Example of one master request, no SPBA arbitration

The following figure assumes MA and MB have been the last two masters granted in the previous transfers (MA then MB).

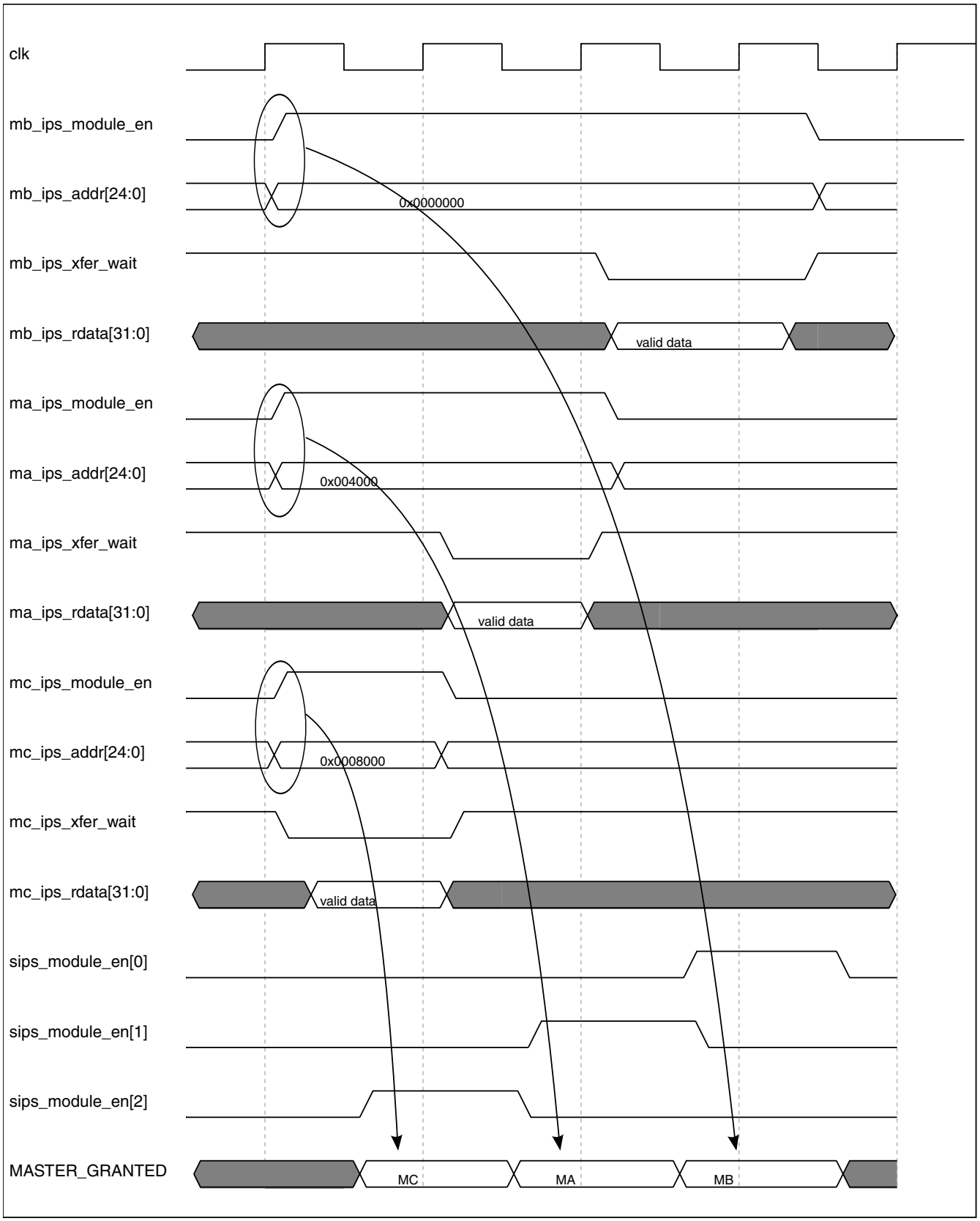


Figure 4-14. Example of three master requests: Masters already granted are "waited";

4.8.2.3 Resource ownership control

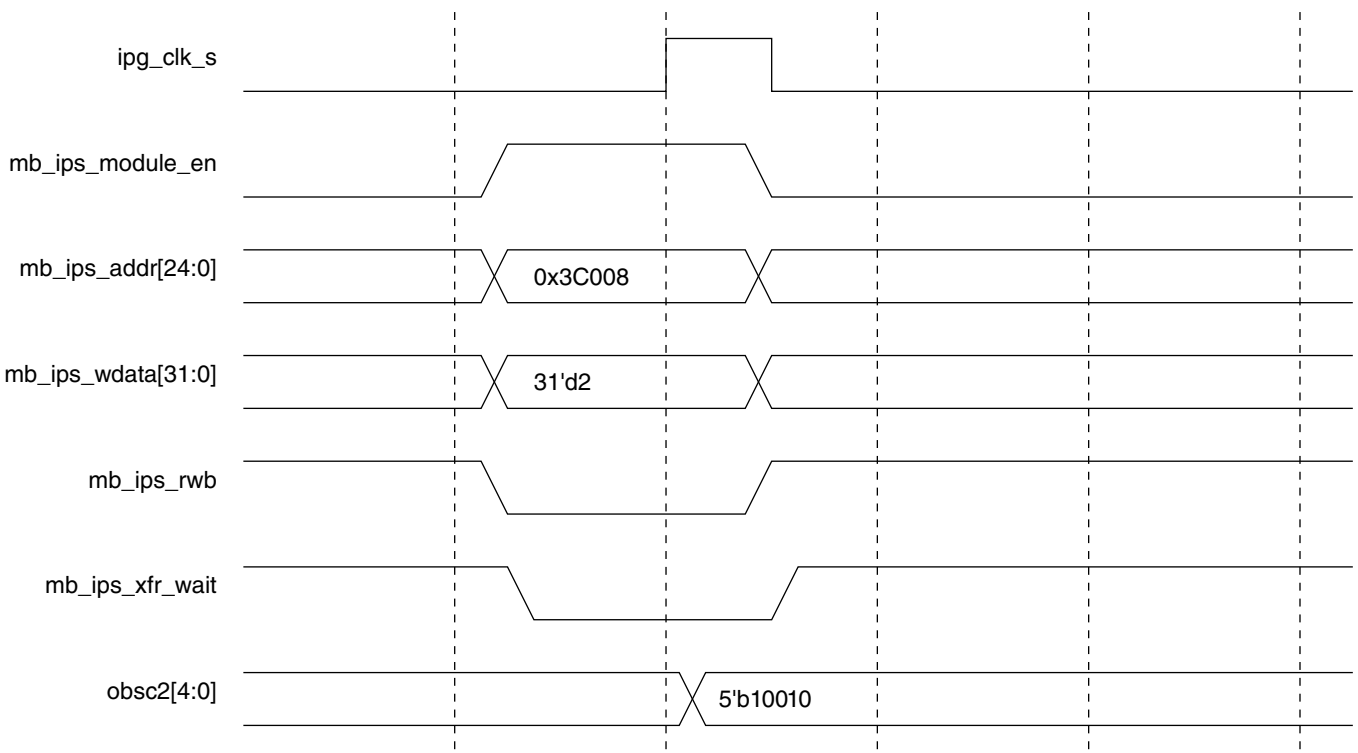
The resource ownership control regulates access to the shared peripherals and determines the steering of out-of-band signals.

4.8.2.3.1 Access control

4.8.2.3.1.1 Peripheral access

The peripheral access (resource access) of the requesting master is given by the corresponding RAR bit of the Peripheral Right Register. It determines if the master has access privilege to the resource.

Any attempt at access made by a requesting master whose access privilege bit is not set (in the PRR) is terminated with a bus error (**<master>_ips_xfr_err** is asserted by SPBA logic). The master that owns the resource can lock the peripheral for itself and/or grant other masters access to the peripheral by setting the appropriate bit(s) in the RAR field.



Master B is taking ownership of peripheral 2 by writing 3'b010 in the SPBA peripheral 2 right register (rarfield)
This ownership can be checked on obsc2 output as roi2[1:0] = 2'b10 and rar2[2:0] = 3'b010
(obsc[4:0] = {roi2[1], roi2[0], rar2[2], rar2[1], rar2[0]})

Figure 4-15. Example of one master B gaining ownership of peripheral 2

4.8.2.3.1.2 Peripheral Right Register access

The ROI bits of the Peripheral Right Register (PRR) determine which master is allowed to make write access to PRR. The identification of the requesting master is compared to the ROI bits of the PRR to determine if the master has ownership of the corresponding register.

Any attempted write access to a PRR already owned by another master will be ignored.

4.8.2.3.2 Owner election

When the peripheral is not owned by any master (ROI="00", after coming out of reset for instance), the first master to perform successfully a write to the RAR bits of the PRR is granted ownership of the peripheral and its associated PRR.

After writing to the PRR (RAR bit(s)), the master must read it back to make sure that it was granted ownership. If the RMO field is 2'b11, then the ownership claim is successful. If RMO is 2'b10, another master claimed ownership before this master was able to complete its write. This resolves the case in which two or more masters attempt to write the PRR at the same time; only the first master will be granted ownership. However all masters must read the PRR to determine if this case occurred, and if so, whether they were the first master which was granted ownership.

NOTE

A master that has been granted ownership of the PRR does not automatically have the right access to the peripheral; it must still set its own RAR bits in the PRR to access the peripheral.

4.8.2.3.3 Ending ownership

Ownership may be voluntarily ended by the owning master, or automatically upon assertion of a master-specific dead_owner signal.

The former is appropriate for software-controlled yielding of ownership. The latter is appropriate for automatic yielding of ownership when the owner has gone into reset.

When a master is reset, it clears the ROI bits of the PRRs owned by the corresponding master. When the owner is dead (in reset), all peripherals previously owned by that master must be changed to the un-owned state.

NOTE

It is the programmer's responsibility to make sure the peripherals are placed in an appropriate state before ending ownership.

4.8.2.3.3.1 Software Controlled Ownership Ending

The ROI bits will be automatically cleared when the master that owns the PRR access right clears (write) the RAR bits (Table 2).

It will then end the ownership of the PRR.

4.8.2.3.4 The Un-owned State

During the time when the peripheral is un-owned (i.e the ROI field contains all 0's), all masters have full access to it (RAR bits can then be modified by a master if ROI[1:0] = 2'b0).

In such cases it is necessary for software to ensure any necessary coherency in the resource, there is no hardware protection.

4.8.2.4 Clocks

The table found here describes the clock sources for SPBA.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 4-21. SPBA Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

4.8.3 SPBA Memory Map/Register Definition

The SPBA control registers (Peripheral Right Registers) are mapped as a virtual shared peripheral.

SPBA can support up to 32 shared peripherals. Each of them has its own Peripheral Right Register (PRR) accessible within the SPBA memory-mapped registers, and consists of the Requesting Master Owner, the Resource Owner ID and the Resource Access Right fields.

SPBA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
308F_0000	Peripheral Rights Register (SPBA1_PRR0)	32	R/W	0000_0007h	4.8.3.1/200
308F_0004	Peripheral Rights Register (SPBA1_PRR1)	32	R/W	0000_0007h	4.8.3.1/200
308F_0008	Peripheral Rights Register (SPBA1_PRR2)	32	R/W	0000_0007h	4.8.3.1/200
308F_000C	Peripheral Rights Register (SPBA1_PRR3)	32	R/W	0000_0007h	4.8.3.1/200
308F_0010	Peripheral Rights Register (SPBA1_PRR4)	32	R/W	0000_0007h	4.8.3.1/200
308F_0014	Peripheral Rights Register (SPBA1_PRR5)	32	R/W	0000_0007h	4.8.3.1/200
308F_0018	Peripheral Rights Register (SPBA1_PRR6)	32	R/W	0000_0007h	4.8.3.1/200
308F_001C	Peripheral Rights Register (SPBA1_PRR7)	32	R/W	0000_0007h	4.8.3.1/200
308F_0020	Peripheral Rights Register (SPBA1_PRR8)	32	R/W	0000_0007h	4.8.3.1/200
308F_0024	Peripheral Rights Register (SPBA1_PRR9)	32	R/W	0000_0007h	4.8.3.1/200
308F_0028	Peripheral Rights Register (SPBA1_PRR10)	32	R/W	0000_0007h	4.8.3.1/200
308F_002C	Peripheral Rights Register (SPBA1_PRR11)	32	R/W	0000_0007h	4.8.3.1/200
308F_0030	Peripheral Rights Register (SPBA1_PRR12)	32	R/W	0000_0007h	4.8.3.1/200
308F_0034	Peripheral Rights Register (SPBA1_PRR13)	32	R/W	0000_0007h	4.8.3.1/200
308F_0038	Peripheral Rights Register (SPBA1_PRR14)	32	R/W	0000_0007h	4.8.3.1/200
308F_003C	Peripheral Rights Register (SPBA1_PRR15)	32	R/W	0000_0007h	4.8.3.1/200
308F_0040	Peripheral Rights Register (SPBA1_PRR16)	32	R/W	0000_0007h	4.8.3.1/200
308F_0044	Peripheral Rights Register (SPBA1_PRR17)	32	R/W	0000_0007h	4.8.3.1/200
308F_0048	Peripheral Rights Register (SPBA1_PRR18)	32	R/W	0000_0007h	4.8.3.1/200
308F_004C	Peripheral Rights Register (SPBA1_PRR19)	32	R/W	0000_0007h	4.8.3.1/200
308F_0050	Peripheral Rights Register (SPBA1_PRR20)	32	R/W	0000_0007h	4.8.3.1/200
308F_0054	Peripheral Rights Register (SPBA1_PRR21)	32	R/W	0000_0007h	4.8.3.1/200
308F_0058	Peripheral Rights Register (SPBA1_PRR22)	32	R/W	0000_0007h	4.8.3.1/200
308F_005C	Peripheral Rights Register (SPBA1_PRR23)	32	R/W	0000_0007h	4.8.3.1/200
308F_0060	Peripheral Rights Register (SPBA1_PRR24)	32	R/W	0000_0007h	4.8.3.1/200

Table continues on the next page...

SPBA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
308F_0064	Peripheral Rights Register (SPBA1_PRR25)	32	R/W	0000_0007h	4.8.3.1/200
308F_0068	Peripheral Rights Register (SPBA1_PRR26)	32	R/W	0000_0007h	4.8.3.1/200
308F_006C	Peripheral Rights Register (SPBA1_PRR27)	32	R/W	0000_0007h	4.8.3.1/200
308F_0070	Peripheral Rights Register (SPBA1_PRR28)	32	R/W	0000_0007h	4.8.3.1/200
308F_0074	Peripheral Rights Register (SPBA1_PRR29)	32	R/W	0000_0007h	4.8.3.1/200
308F_0078	Peripheral Rights Register (SPBA1_PRR30)	32	R/W	0000_0007h	4.8.3.1/200
308F_007C	Peripheral Rights Register (SPBA1_PRR31)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0000	Peripheral Rights Register (SPBA2_PRR0)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0004	Peripheral Rights Register (SPBA2_PRR1)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0008	Peripheral Rights Register (SPBA2_PRR2)	32	R/W	0000_0007h	4.8.3.1/200
30CF_000C	Peripheral Rights Register (SPBA2_PRR3)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0010	Peripheral Rights Register (SPBA2_PRR4)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0014	Peripheral Rights Register (SPBA2_PRR5)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0018	Peripheral Rights Register (SPBA2_PRR6)	32	R/W	0000_0007h	4.8.3.1/200
30CF_001C	Peripheral Rights Register (SPBA2_PRR7)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0020	Peripheral Rights Register (SPBA2_PRR8)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0024	Peripheral Rights Register (SPBA2_PRR9)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0028	Peripheral Rights Register (SPBA2_PRR10)	32	R/W	0000_0007h	4.8.3.1/200
30CF_002C	Peripheral Rights Register (SPBA2_PRR11)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0030	Peripheral Rights Register (SPBA2_PRR12)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0034	Peripheral Rights Register (SPBA2_PRR13)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0038	Peripheral Rights Register (SPBA2_PRR14)	32	R/W	0000_0007h	4.8.3.1/200
30CF_003C	Peripheral Rights Register (SPBA2_PRR15)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0040	Peripheral Rights Register (SPBA2_PRR16)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0044	Peripheral Rights Register (SPBA2_PRR17)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0048	Peripheral Rights Register (SPBA2_PRR18)	32	R/W	0000_0007h	4.8.3.1/200
30CF_004C	Peripheral Rights Register (SPBA2_PRR19)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0050	Peripheral Rights Register (SPBA2_PRR20)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0054	Peripheral Rights Register (SPBA2_PRR21)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0058	Peripheral Rights Register (SPBA2_PRR22)	32	R/W	0000_0007h	4.8.3.1/200
30CF_005C	Peripheral Rights Register (SPBA2_PRR23)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0060	Peripheral Rights Register (SPBA2_PRR24)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0064	Peripheral Rights Register (SPBA2_PRR25)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0068	Peripheral Rights Register (SPBA2_PRR26)	32	R/W	0000_0007h	4.8.3.1/200
30CF_006C	Peripheral Rights Register (SPBA2_PRR27)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0070	Peripheral Rights Register (SPBA2_PRR28)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0074	Peripheral Rights Register (SPBA2_PRR29)	32	R/W	0000_0007h	4.8.3.1/200
30CF_0078	Peripheral Rights Register (SPBA2_PRR30)	32	R/W	0000_0007h	4.8.3.1/200

Table continues on the next page...

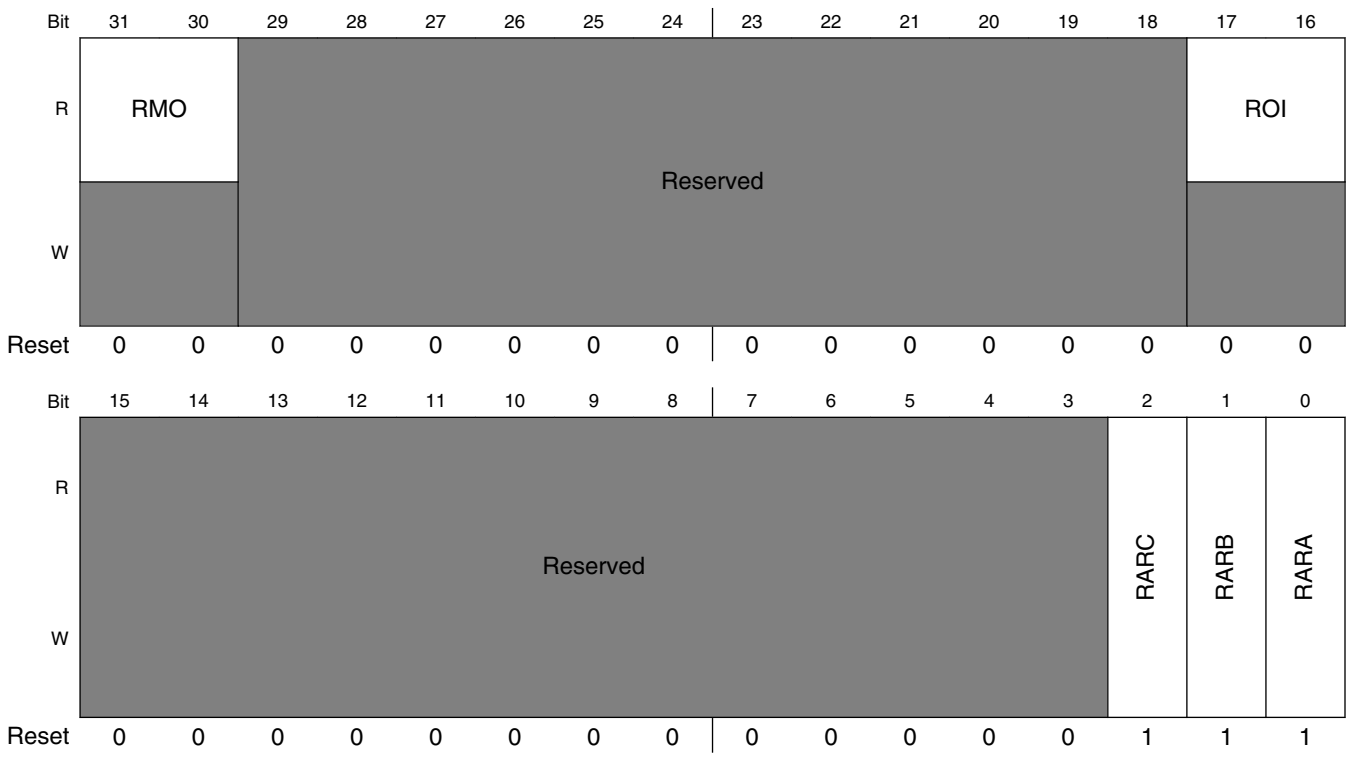
SPBA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30CF_007C	Peripheral Rights Register (SPBA2_PRR31)	32	R/W	0000_0007h	4.8.3.1/200

4.8.3.1 Peripheral Rights Register (SPBA*x*_PRR*n*)

This register controls master ownership and access for a peripheral.

Address: Base address + 0h offset + (4d × i), where i=0d to 31d



SPBA*x*_PRR*n* field descriptions

Field	Description
31–30 RMO	Requesting Master Owner. This 2-bit register field indicates if the corresponding resource is owned by the requesting master or not. This register is reset to 2'b0 if ROI = 2'b0. 00 **UNOWNED** — The resource is unowned. 01 Reserved. 10 **ANOTHER_MASTER** — The resource is owned by another master. 11 **REQUESTING_MASTER** — The resource is owned by the requesting master.
29–18 -	This field is reserved. Reserved

Table continues on the next page...

SPBA*x*_PRR*n* field descriptions (continued)

Field	Description
17–16 ROI	Resource Owner ID. This field indicates which master (one at a time) can access to the PRR for rights modification. This is a read-only register. After reset, ROI bits are cleared ("00" -> un-owned resource). A master performing a write access to the an un-owned PRR will get its ID automatically written into ROI, while modifying RARx bits. It can then read back the RMO, RAR, ROI bits to make sure RMO returns the right value, ROI bits contain its ID and RARx bits are correctly asserted. Then no other master (whom ID is different from the one stored in ROI) will be able to modify RAR fields. Owner master of a peripheral can assert its dead_owner signal, or write 1'b0 in the RARx to release the ownership (ROI[1:0] reset to 2'b0). 00 **UNOWNED** — Unowned resource. 01 **MASTER_A** — The resource is owned by master A port. 10 **MASTER_B** — The resource is owned by master B port. 11 **MASTER_C** — The resource is owned by master C port.
15–3 -	This field is reserved. Reserved
2 RARC	Resource Access Right. Control and Status bit for master C. This field indicates whether master C can access the peripheral. From 0 up to 3 masters can have permission to access a resource (all the master can be granted on a peripheral, but only one access at a time will be granted by SPBA). 0 **PROHIBITED** — Access to peripheral is not allowed. 1 **ALLOWED** — Access to peripheral is granted.
1 RARB	Resource Access Right. Control and Status bit for master B. This field indicates whether master B can access the peripheral. From 0 up to 3 masters can have permission to access a resource (all the master can be granted on a peripheral, but only one access at a time will be granted by SPBA). 0 **PROHIBITED** — Access to peripheral is not allowed. 1 **ALLOWED** — Access to peripheral is granted.
0 RARA	Resource Access Right. Control and Status bit for master A. This field indicates whether master A can access the peripheral. From 0 up to 3 masters can have permission to access a resource (all the master can be granted on a peripheral, but only one access at a time will be granted by SPBA). 0 **PROHIBITED** — Access to peripheral is not allowed. 1 **ALLOWED** — Access to peripheral is granted.

4.9 TrustZone Address Space Controller (TZASC)

4.9.1 Overview

The TrustZone Address Space Controller (TZASC) protects security-sensitive SW and data in a trusted execution environment against potentially compromised SW running on the platform.

4.9.1.1 Block Diagram

The TZASC block diagram is shown in figure below.

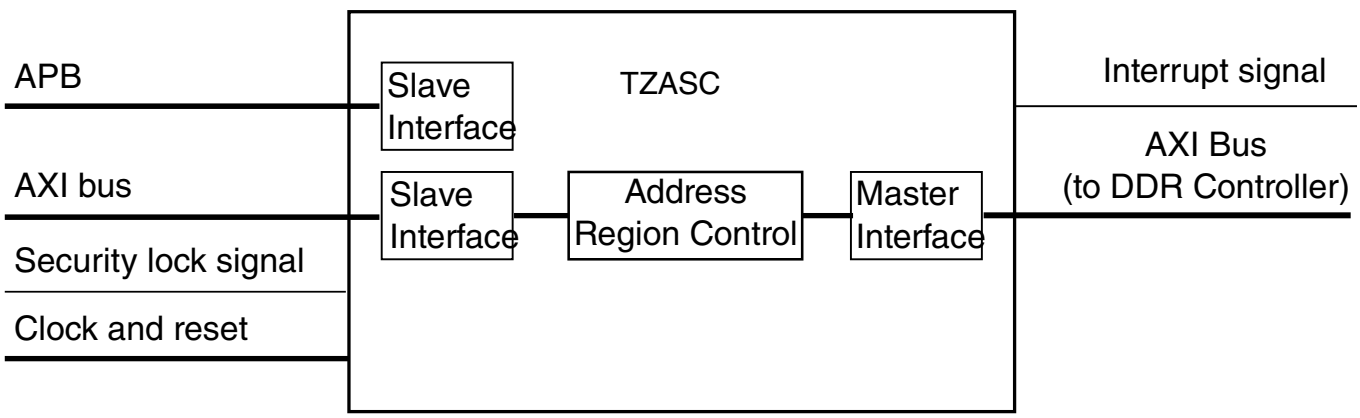


Figure 4-16. TZASC Block Diagram

4.9.1.2 Features

By default, the TZASC module is bypassed. The TZASC_ENABLE fuse should be blown in order for the TZASC to be taken out of bypass mode, and start to perform security checks on AXI accesses to DRAM memory. If this fuse is not blown, it is not possible to create protected regions in DDR.

Enabling the TZASC may have an impact on memory performance. The exact value cannot be stated since it can vary depending on the specific application software.

The TZASC is an IP by Arm ("CoreLink™ TrustZone Address Space Controller TZC-380"), designed to provide configurable protection over program (SW) memory space.

The main features of TZASC are:

- Supports 16 independent address regions
- Access controls are independently programmable for each address region
- Sensitive registers may be locked
- Host interrupt may be programmed to signal attempted access control violations
- AXI master/slave interfaces for transactions
- APB slave interface for configuration and status reporting

NOTE

In this device it is necessary to set TZASC_ID_SWAP_BYPASS in IOMUXC_GPR10[1] as ID SWAP function can't work as expected.

4.9.2 Functional Description

4.9.2.1 Address Mapping in various memory mapping modes

The TZASC region base address starts at the beginning of DDR memory space (0x40000000) instead of the beginning of memory map (0x00000000). In this case the addresses configured in TZASC controller will be 1GB (0x40000000) offset and does not match the local addresses.

For example, setting region_setup_low_x=0xBE000000 maps DDR_ADDR=0xFE000000, the same behavior is observed with fail_address_x registers.

Memory "aliasing" implications on TZASC settings - in systems which does not utilize the maximal supported DDR space the controller is designed for, the whole DDR memory map becomes "aliased" (replicated) by the size of the physical memory used. In such cases, the TZASC must be configured to protect all aliased regions as well (i.e. effectively reducing the number of available TZASC regions, since all aliased regions must be handled, for each "real" space needing protection).

For complete details on TZASC functionality and the programming model, see the Arm document, “CoreLink™ TrustZone Address Space Controller TZC-380 Technical Reference Manual, (Rev r0p1 or newer)”, available at **http://infocenter.arm.com**.

4.9.2.2 Clocks

The table found here describes the clock sources for TZASC.

Table 4-22. TZASC Clocks

Clock name	Clock Root	Description
aclk	ccm_clk_root	Module clock

4.10 System Debug

4.10.1 Debug

4.10.1.1 Debug Architecture

The chapter describes the debug architecture of the chip.

4.10.1.2 Debug System Features

The chip debug is based on Arm's CoreSight platform, with support for Quad-core A53 platform and Cortex-M7 core. The key features of the debug system include:

- Support 5-pins(JTAG) interface.
- Support both non-intrusive and halt-mode trace/debug options.
- MDM-AP registers for debugger to control mutli-core halt/resume cores.
- Trace Memory Controller (TMC) is used to enable capturing trace.
- 4KB in SOC trace block.
- ETR is used to allow routing trace data to system memory.
- Support Arm real time trace interface (TPIU) 16-bit @133MHz.
- Support cross trigger between Quad Cortex-A53 and Cortex-M7.

4.10.1.3 System level debug architecture

The debug architecture is shown in the following figure:

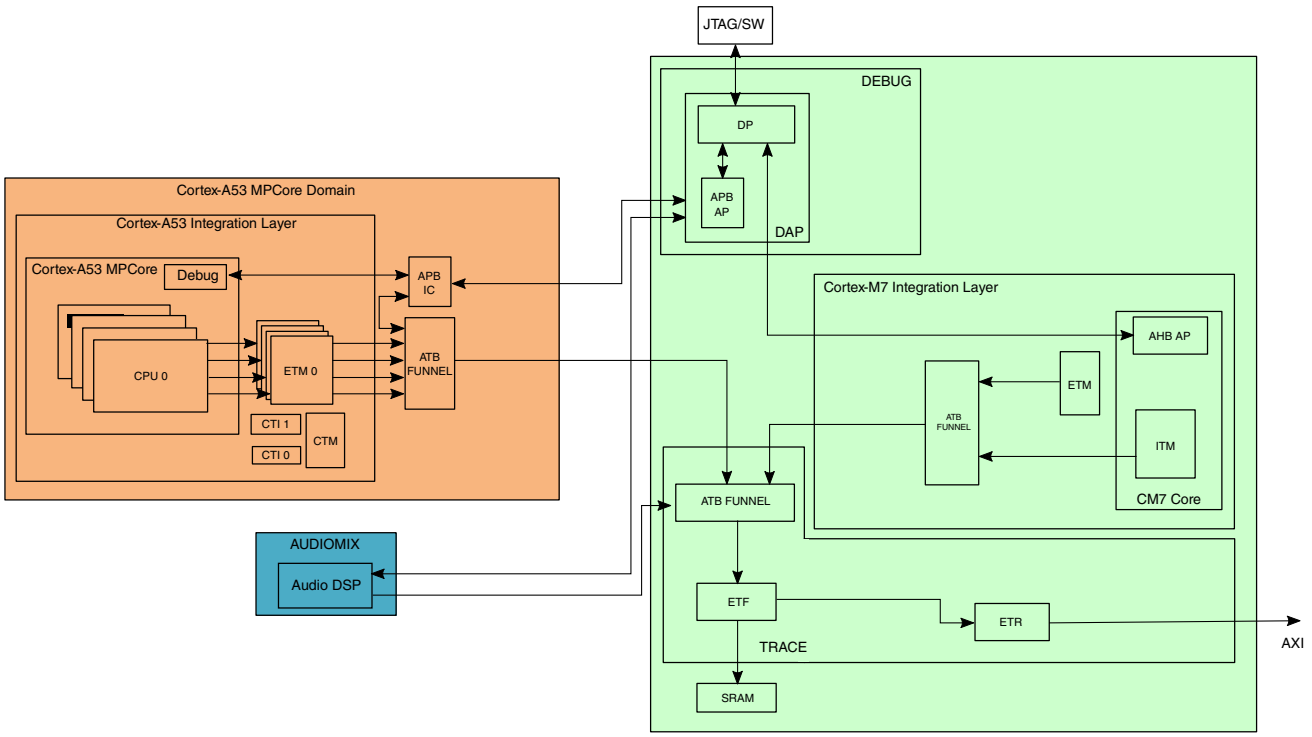


Figure 4-17. Debug Architecture Diagram

4.10.1.4 Functional description

This section gives a brief overview of the modules that are implemented within the Cortex-M/Cortex-A Core Platform. The debug blocks are part of the overall CoreSight platform debug system, which include the ETR, CTM, CTI, ATB replicator, APB address decode, TPIU and DAP. The CoreSight™ compatible Embedded Trace Macrocell (ETM) enables traces of program flow to be collected, compressed, and fed into the trace infrastructure. The Cross Trigger Interface (CTI) is included to provide a common programming model for use by the debug tools, control the trigger sources, and interface to the Cross Trigger Matrix (CTM). The debug is controlled via an ARM Debug Access Port (DAP).

4.10.1.4.1 Embedded Trace Router (ETR)

The Embedded Trace Router (ETR) is a CoreSight trace component which buffers trace data before dumping that data into DRAM through the AXI bus (via AXI master port). The output of the funnel is the ETR, and has the ability to buffer a certain amount before writing the data to DRAM.

4.10.1.4.2 Embedded Trace Macrocell (ETM)

The Embedded Trace Macrocell (ETM) is a CoreSight component. The ETM trace unit is a module that performs real-time instruction flow tracing based on the Embedded Trace Macrocell (ETM) architecture ETMv4.

4.10.1.4.3 Cross Trigger Matrix (CTM)

The CoreSight CTI channel signals from all the cores are combined using a Cross Trigger Matrix (CTM) block so that a single cross trigger channel interface is presented in the Cortex processor. This module can combine four internal channel interfaces corresponding to each core along with one external channel interface.

In the Cortex processor CTM, the external channel output is driven by the OR output of all internal channel outputs. Each internal channel input is driven by the OR output of internal channel outputs of all other CTIs in addition to the external channel input.

4.10.1.4.4 Cross Trigger Interface (CTI)

The Cortex processor has a single external cross trigger channel interface. This external interface is connected to the CoreSight Cross Trigger Interface (CTI) interface corresponding to each core through a Cross Trigger Matrix (CTM). A number of Embedded Cross Trigger (ECT) trigger inputs and trigger outputs are connected between debug components in the Cortex-A processor and CoreSight CTI blocks.

The CTI enables the debug logic, ETM trace unit, and performance monitoring, to interact with each other and with other CoreSight components. This is called cross triggering. For example, you configure the CTI to generate an interrupt when the ETM trace unit trigger event occurs.

4.10.1.4.5 AMBA Trace Bus Interface (ATB)

The AMBA Trace Bus (ATB) CoreSight debug components are part of the Trace stream block. The ATB components consist of the ATB Funnel and ATB Replicator. The ATB Funnel is used to merge several streams together to produce a single output to the ATB Replicator. The ATB Replicator enables two trace sinks to be wired together and receive ATB trace data from the same trace source.

4.10.1.4.6 Advanced Peripheral Bus Interface (APB)

The APB asynchronous interface / bridge connects several debug components. Each component has a single separate APB interface from the APB-IC (w/ ROM), which hooks up to a separate APB asynchronous component. The APB interface connects the core debug functions to the DAP.

4.10.1.4.7 Instrumentation Trace Macrocell (ITM)

The Instrumentation Trace Macrocell (ITM) is a application-driven trace source that supports printf style debugging to trace operating system and application events. The ITM generates diagnostic system information as packets. Multiple sources can generate packets. If multiple sources generate packets at the same time, the ITM arbitrates the order in which packets are output. These sources in decreasing order of priority are:

- **Software trace.** Software can write directly to ITM stimulus registers to generate packets.
- **Hardware trace.** The DWT generates these packets, and the ITM outputs them.
- **Time stamping.** Timestamps are generated relative to packets. The ITM contains a 21-bit counter to generate the timestamp. The Cortex-M processor clock or the bitclock rate of the Serial Wire Viewer (SWV) output clocks the counter.

4.10.1.4.8 Data Watchpoint and Trace (DWT)

The Data Watchpoint and Trace (DWT) debug unit provides watchpoints, data tracing, and system profiling for the processor.

4.10.1.4.9 Debug Access Port (DAP)

The Debug Access Port (DAP) is a physical port that connects to external debug tools. The DAP is part of the standardized ARM Debug Interface, and provides a bridge between a JTAG pin interface and on-chip memory mapped peripherals via the DAP bus. Transactions generated by the DAP are referred to as External Debugger Accesses.

4.10.1.5 JTAG topology

There is only one JTAG on the chip, and two JTAG modes are supported. Select via the JTAG_MOD pin.

- Debug mode: JTAG_MOD == 0, DAP is the only TAP controller in the daisy chain. SJC will be attached to JTAG-AP of DAP.
- Test mode: JTAG_MOD == 1, SJC is the only TAP controller in the daisy chain. 1149.1-compliant, and support 1149.6 AC coupled test.

When the JTAG interface is in Debug Mode, it can be operating in standard 5-pin JTAG interface. cJTAG/SWD interface is not supported by this chip.

4.11 System Counter (SYS_CTR)

4.11.1 Overview

The System Counter (SYS_CTR) is a programmable system counter, which provides a shared time base to multiple processors. It is intended for applications where the counter is always powered on, and supports multiple unrelated clocks.

4.11.1.1 Block diagram

The block diagram of the System Counter is shown below.

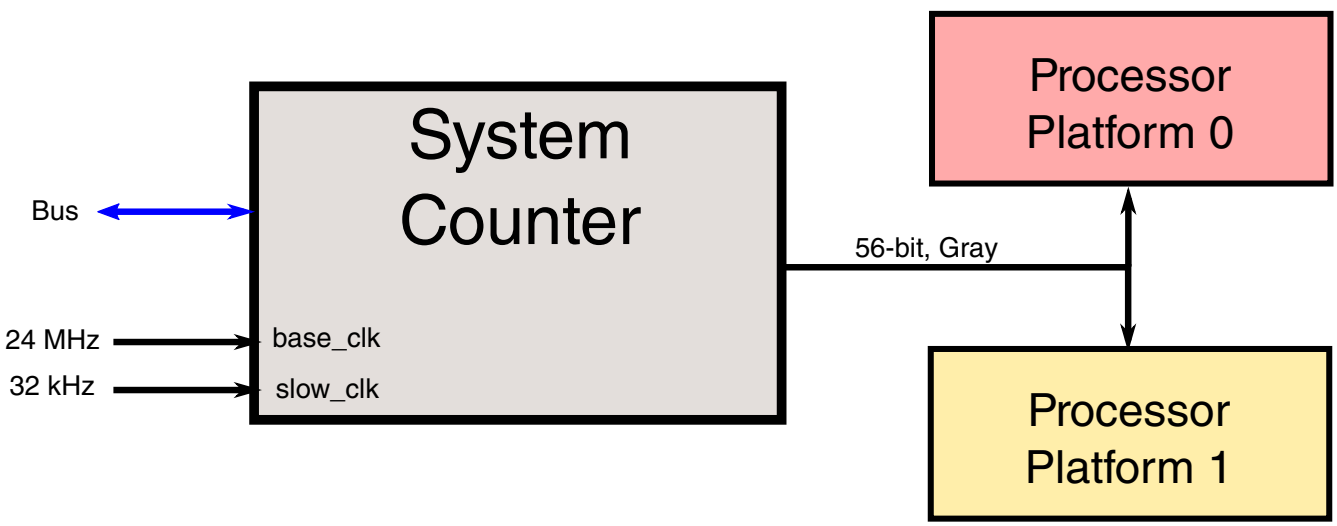


Figure 4-18. Block diagram

4.11.1.2 Features

- Two counter clock frequencies
  - Base clock for normal operation
  - Alternate clock for low power operation
- 56-bit counter width
- Gray coded counter output for distribution to the processor timers
- 2 Compare Frames

- Each compare frame contains a 64-bit compare value
- Each compare frame contains programmable interrupt generation

4.11.2 Functional description

The System Counter inputs two counter clock sources and outputs a gray coded counter value and interrupt signals (one per compare frame) to the platform's interrupt controller.

4.11.2.1 Operation

After reset, the System Counter is disabled with count value reset to zero and base frequency selected. Once the counter is enabled, it will increment the appropriate value on each rising edge of the selected clock. Because the System Counter is handling a 56-bit count value across multiple clock domains, synchronization is necessary. The System Counter provides synchronization mechanisms between the various clock domains.

When the system switches the counter's clock source, there is a short pause while the clock multiplexer is handling the clock transition. In order to maintain an accurate count value, the clock control logic employs two offset counters; one for the base-to-slow transtion and one for the slow-to-base transition. These offset counters only operate during the clock transition time to compensate for the idled source clock. Both counters run off of the base clock. The transition offset values are added to the system count value at the appropriate time when the counter's clock is restored.

NOTE

Both base clock and alternate clock must be running when changing frequencies.

4.11.2.2 Clocks

The System Counter clocks are shown in the table below.

Table 4-23. Clocks

Clock	Description
ipg_clk	Peripheral Clock
ipg_clk_s	Gated peripheral clock for register transactions
base_clk	Base Clock. This clock is used during normal operation. It is internally divided by 3 before use.
slow_clk	Slow Clock. This clock is used during low power mode. It is internally divided by 64 before use.

Table continues on the next page...

Table 4-23. Clocks (continued)

Clock	Description
ctr_clk	Counter Clock. The ctr_clk is generated by selecting either the base_clk or slow_clk. Because the compare frame must generate an interrupt when either clock is selected, most of the compare frame logic resides in this clock domain.

4.11.3 Programming model

The System Counter module contians three programmer's model sections. Each sections has an indepentent module enable to allow individual secure access control.

Table 4-24. Programming model

Programming model	Description
CNTControlBase	This model controls the counter frequency, enable, debug, and count value. This model also contains the frequency modes table which supplies control and status for the frequency of the System Counter.
CNTReadBase	This model allows the user to read the count value.
CNTCompareBase	This model controls the compare value frequency, enable, interrupt mask, and interrupt status for each compare frame.

4.11.4 Memory map/register definition

This section describes the registers and data structures in the SYS_CTR module.

4.11.4.1 SYS_CTR_CONTROL register descriptions

SYS_CTR_CONTROL Hardware Register Format Summary

4.11.4.1.1 SYS_CTR_CONTROL memory map

SYS_CTR_CONTROL base address: 0h

Offset	Register	Width (In bits)	Access	Reset value
0h	Counter Control Register (CNTCR)	32	RW	0000_0000h
4h	Counter Status Register (CNTSR)	32	RO	0000_0100h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8h	Counter Count Value Low Register (CNTCV0)	32	RW	0000_0000h
Ch	Counter Count Value High Register (CNTCV1)	32	RW	0000_0000h
20h	Frequency Modes Table 0 Register (CNTFID0)	32	RO	007A_1200h
24h	Frequency Modes Table 1 Register (CNTFID1)	32	RO	0000_0200h
28h	Frequency Modes Table 2 Register (CNTFID2)	32	RO	0000_0000h
FD0h	Counter ID Register (CNTID0)	32	RO	0000_0000h

4.11.4.1.2 Counter Control Register (CNTCR)

4.11.4.1.2.1 Offset

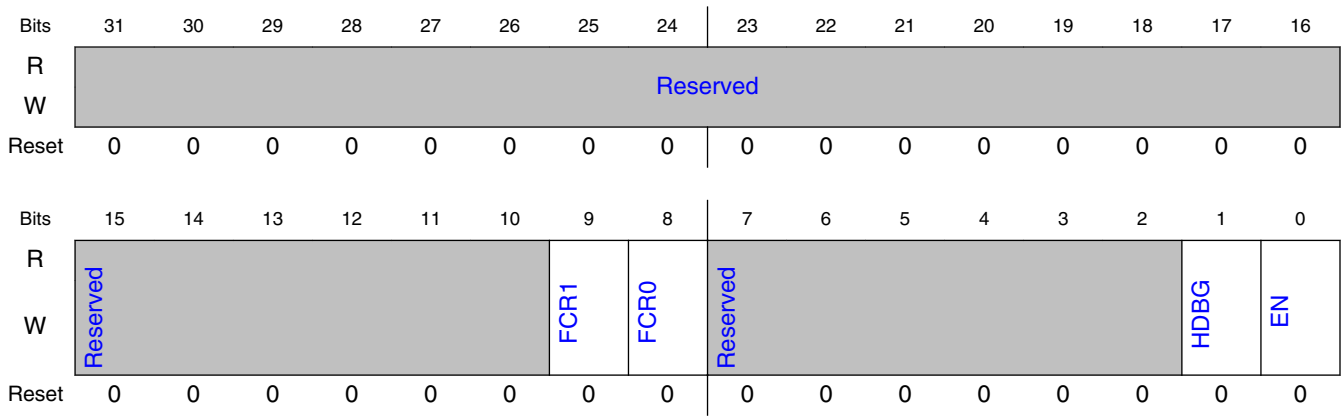
Register	Offset
CNTCR	0h

4.11.4.1.2.2 Function

The 32-bit Counter Control Register defines the basic operating configuration of the System Counter.

The System Counter operates using a fixed base frequency. However, the counter can increment at a lower, alternate frequency than the base frequency, using a correspondingly larger increment. For example, it can increment by 15625 and run at a frequency of 1/15625 of the base frequency. The two frequencies available are indentified in the frequency modes table. These two frequencies are the base frequency (table entry 0) and the lower, alternate frequency (table entry 1). Setting the FCR1 bit selects the alternate frequency. Setting the FCR0 bit selects the base frequency. Setting or clearing both FCR0 and FCR1 will have no effect; the freqeuncy will not change.

4.11.4.1.2.3 Diagram



4.11.4.1.2.4 Fields

Field	Function
31-10 —	- Reserved
9 FCR1	FCR1 Frequency Change Request, ID 1 0b - No change. 1b - Select frequency modes table entry 1, the base frequency.
8 FCR0	FCR0 Frequency Change Request, ID 0 0b - No change. 1b - Select frequency modes table entry 0, the base frequency.
7-2 —	- Reserved
1 HDBG	HDBG Enable Debug 0b - The assertion of the debug input is ignored. 1b - The assertion of the debug input causes the System Counter to halt.
0 EN	EN Enable Counting 0b - Counter disabled 1b - Counter enabled

4.11.4.1.3 Counter Status Register (CNTSR)

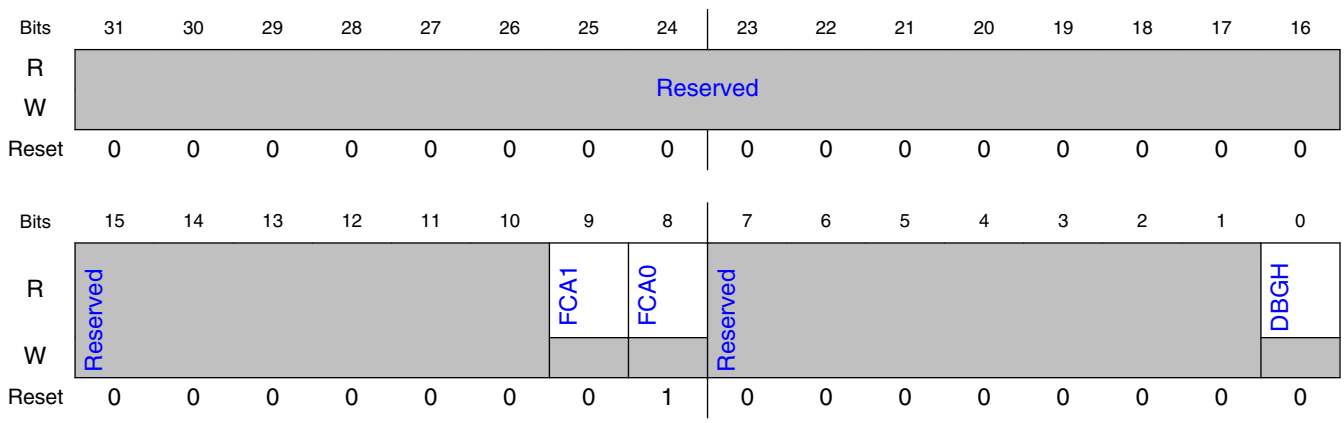
4.11.4.1.3.1 Offset

Register	Offset
CNTSR	4h

4.11.4.1.3.2 Function

The system counter status register provides information concerning the clock frequency and debug state.

4.11.4.1.3.3 Diagram



4.11.4.1.3.4 Fields

Field	Function
31-10 —	- Reserved
9 FCA1	FCA1 Frequency Change Acknowledge, ID 1 0b - Base frequency is not selected. 1b - Base frequency is selected.
8 FCA0	FCA0 Frequency Change Acknowledge, ID 0 0b - Base frequency is not selected. 1b - Base frequency is selected.
7-1 —	- Reserved
0	DBGH

Field	Function
DBGH	Debug Halt 0b - Counter is not halted by debug. 1b - Counter is halted by debug.

4.11.4.1.4 Counter Count Value Low Register (CNTCV0)

4.11.4.1.4.1 Offset

Register	Offset
CNTCV0	8h

4.11.4.1.4.2 Function

The Counter Count Value Low register indicates the current count value bits 31-0.

NOTE

Writes to the CNTCV registers must be performed while operating on the base frequency only. Writes to these registers while running on the alternate frequency may have unpredictable results.

4.11.4.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CNTCV0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CNTCV0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.11.4.1.4.4 Fields

Field	Function
31-0 CNTCV0	CNTCV0 Counter Count Value bits [31:0]

4.11.4.1.5 Counter Count Value High Register (CNTCV1)

4.11.4.1.5.1 Offset

Register	Offset
CNTCV1	Ch

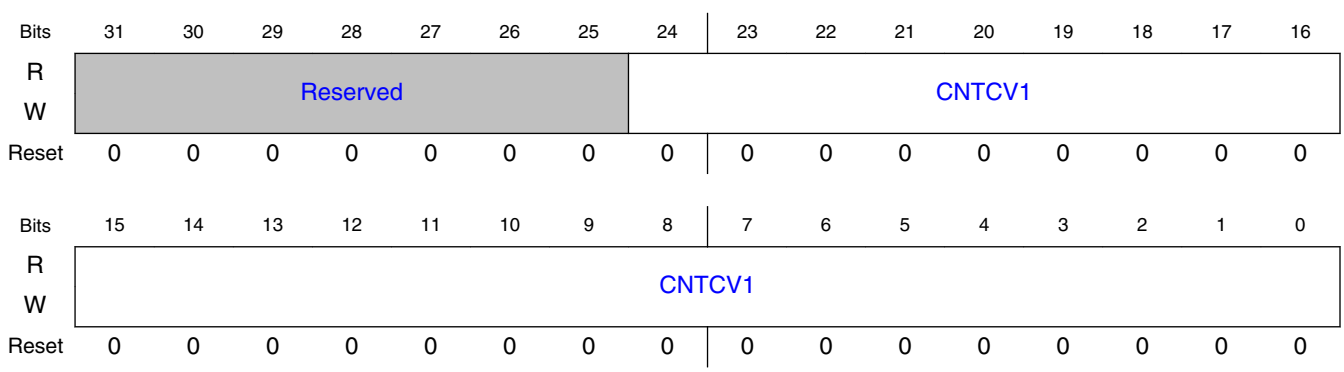
4.11.4.1.5.2 Function

The Counter Count Value High register indicates the current count value bits 63-32.

NOTE

Writes to the CNTCV registers must be performed while operating on the base frequency only. Writes to these registers while running on the alternate frequency may have unpredictable results.

4.11.4.1.5.3 Diagram



4.11.4.1.5.4 Fields

Field	Function
31-25 —	- Reserved
24-0 CNTCV1	CNTCV1 Counter Count Value bits [55:32]

4.11.4.1.6 Frequency Modes Table 0 Register (CNTFID0)

4.11.4.1.6.1 Offset

Register	Offset
CNTFID0	20h

4.11.4.1.6.2 Function

The Counter Frequency ID registers is the frequency modes table starting at offset 0x020.

Table entries are 32-bits, and each entry specifies a system counter update frequency, in Hz. The first entry in the table specifies the base frequency of the system counter. To ensure overall counter accuracy is maintained, any subsequent entries in the table are exact divisors of the base frequency.

When the system timer is operating at a lower frequency than the base frequency, the increment applied at each counter update is given by: increment = (base_frequency) / (selected_frequency) A 32-bit word of zero value marks the end of the table. That is, the word of memory immediately after the last entry in the table is zero.

4.11.4.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CNTFID0															
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNTFID0															
W																
Reset	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0

4.11.4.1.6.4 Fields

Field	Function
31-0 CNTFID0	CNTFID0 Base Frequency (24 MHz /3 = 8 MHz)

4.11.4.1.7 Frequency Modes Table 1 Register (CNTFID1)

4.11.4.1.7.1 Offset

Register	Offset
CNTFID1	24h

4.11.4.1.7.2 Function

The Counter Frequency ID registers is the frequency modes table starting at offset 0x020.

Table entries are 32-bits, and each entry specifies a system counter update frequency, in Hz. The first entry in the table specifies the base frequency of the system counter. To ensure overall counter accuracy is maintained, any subsequent entries in the table are exact divisors of the base frequency.

When the system timer is operating at a lower frequency than the base frequency, the increment applied at each counter update is given by: increment = (base_frequency) / (selected_frequency) A 32-bit word of zero value marks the end of the table. That is, the word of memory immediately after the last entry in the table is zero.

4.11.4.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CNTFID1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNTFID1															
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

4.11.4.1.7.4 Fields

Field	Function
31-0 CNTFID1	CNTFID1 Alternate Frequency (32 kHz /64 = 512 Hz)

4.11.4.1.8 Frequency Modes Table 2 Register (CNTFID2)

4.11.4.1.8.1 Offset

Register	Offset
CNTFID2	28h

4.11.4.1.8.2 Function

The Counter Frequency ID registers is the frequency modes table starting at offset 0x020.

Table entries are 32-bits, and each entry specifies a system counter update frequency, in Hz. The first entry in the table specifies the base frequency of the system counter. To ensure overall counter accuracy is maintained, any subsequent entries in the table are exact divisors of the base frequency.

When the system timer is operating at a lower frequency than the base frequency, the increment applied at each counter update is given by: increment = (base_frequency) / (selected_frequency) A 32-bit word of zero value marks the end of the table. That is, the word of memory immediately after the last entry in the table is zero.

4.11.4.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CNTFID2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNTFID2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.11.4.1.8.4 Fields

Field	Function
31-0 CNTFID2	CNTFID2 End Marker

4.11.4.1.9 Counter ID Register (CNTID0)

4.11.4.1.9.1 Offset

Register	Offset
CNTID0	FD0h

4.11.4.1.9.2 Function

The Counter ID register indicates the architecture version 0.

4.11.4.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CNTID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNTID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.11.4.1.9.4 Fields

Field	Function
31-0 CNTID	CNTID Counter Identification. Counter ID 0.

4.11.4.2 SYS_CTR_READ register descriptions

The read frame registers are all read-only. These registers read the same values as the control frame registers for the count value and counter ID. They are processed via a separate mechanism from the control frame to allow nonsecure, user mode access.

4.11.4.2.1 SYS_CTR_READ memory map

SYS_CTR_READ base address: 0h

Offset	Register	Width (In bits)	Access	Reset value
0h	Counter Count Value Low Register (CNTCV0)	32	RO	0000_0000h
4h	Counter Count Value High Register (CNTCV1)	32	RO	0000_0000h
FD0h	Counter ID Register (CNTID0)	32	RO	0000_0000h

4.11.4.2.2 Counter Count Value Low Register (CNTCV0)

4.11.4.2.2.1 Offset

Register	Offset
CNTCV0	0h

4.11.4.2.2.2 Function

The Counter Count Value Low register indicates the current count value bits 31-0.

4.11.4.2.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CNTCV0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNTCV0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.11.4.2.2.4 Fields

Field	Function
31-0 CNTCV0	CNTCV0 Counter Count Value bits [31:0]

4.11.4.2.3 Counter Count Value High Register (CNTCV1)

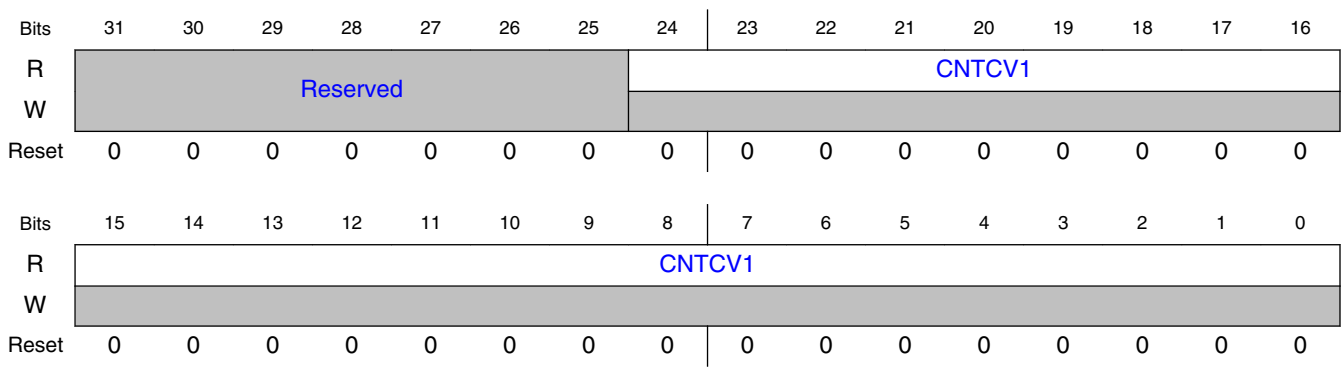
4.11.4.2.3.1 Offset

Register	Offset
CNTCV1	4h

4.11.4.2.3.2 Function

The Counter Count Value High register indicates the current count value bits 63-32.

4.11.4.2.3.3 Diagram



4.11.4.2.3.4 Fields

Field	Function
31-25 —	- Reserved
24-0 CNTCV1	CNTCV1 Counter Count Value bits [55:32]. Bits[63:56] are always zero.

4.11.4.2.4 Counter ID Register (CNTID0)

4.11.4.2.4.1 Offset

Register	Offset
CNTID0	FD0h

4.11.4.2.4.2 Function

The Counter ID register indicates the architecture version 0.

4.11.4.2.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CNTID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNTID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.11.4.2.4.4 Fields

Field	Function
31-0 CNTID	CNTID Counter Identification. Counter ID 0.

4.11.4.3 SYS_CTR_COMPARE register descriptions

Each compare frame consists of a 256 byte region. Each compare frame has its own compare value and control register. Each compare frame is capable of generating one maskable interrupt.

4.11.4.3.1 SYS_CTR_COMPARE memory map

SYS_CTR_COMPARE base address: 0h

Offset	Register	Width (In bits)	Access	Reset value
20h	Compare Count Value Low Register (CMPCVL0)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
24h	Compare Count Value High Register (CMPCVH0)	32	RO	0000_0000h
2Ch	Compare Control Register (CMPCR0)	32	RW	0000_0000h
120h	Compare Count Value Low Register (CMPCVL1)	32	RO	0000_0000h
124h	Compare Count Value High Register (CMPCVH1)	32	RO	0000_0000h
12Ch	Compare Control Register (CMPCR1)	32	RW	0000_0000h
FD0h	Counter ID Register (CNTID0)	32	RO	0000_0000h

4.11.4.3.2 Compare Count Value Low Register (CMPCVL0 - CMPCVL1)

4.11.4.3.2.1 Offset

Register	Offset
CMPCVL0	20h
CMPCVL1	120h

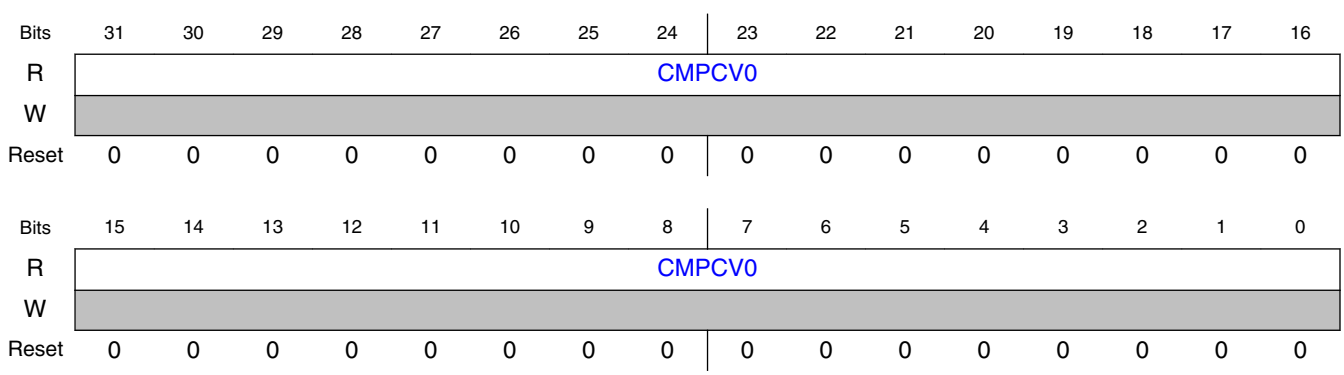
4.11.4.3.2.2 Function

The Compare Count Value Low register indicates the current count value bits 31-0.

NOTE

Writes to the CMPCV registers must be performed while operating on the base frequency only. Writes to these registers while running on the alternate frequency may have unpredictable results.

4.11.4.3.2.3 Diagram



4.11.4.3.2.4 Fields

Field	Function
31-0 CMPCV0	CMPCV0 Compare Count Value bits [31:0]

4.11.4.3.3 Compare Count Value High Register (CMPCVH0 - CMPCVH1)

4.11.4.3.3.1 Offset

Register	Offset
CMPCVH0	24h
CMPCVH1	124h

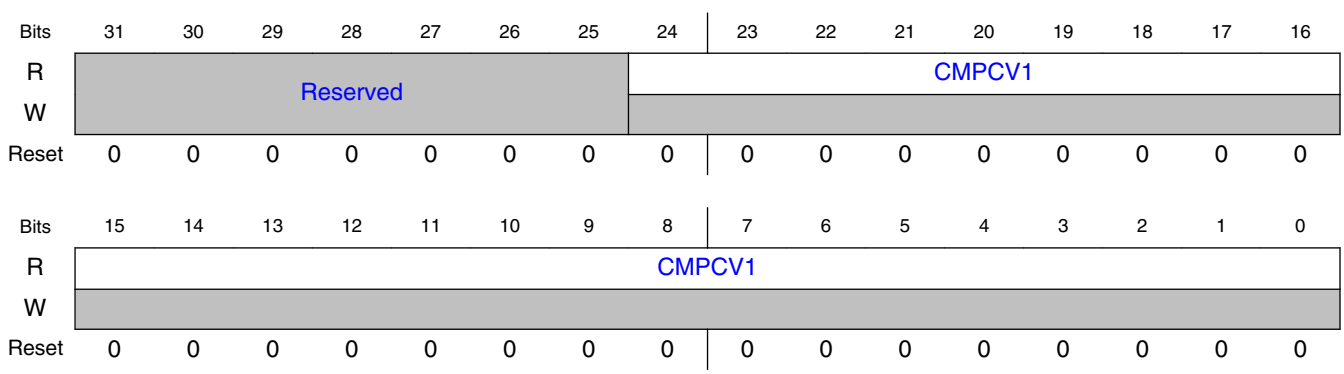
4.11.4.3.3.2 Function

The Compare Count Value High register indicates the current count value bits 63-32.

NOTE

Writes to the CMPCV registers must be performed while operating on the base frequency only. Writes to these registers while running on the alternate frequency may have unpredictable results.

4.11.4.3.3.3 Diagram



4.11.4.3.3.4 Fields

Field	Function
31-25 —	- Reserved
24-0 CMPCV1	CMPCV1 Compare Count Value bits [55:32]. Bits[63:56] are always zero.

4.11.4.3.4 Compare Control Register (CMPCR0 - CMPCR1)

4.11.4.3.4.1 Offset

Register	Offset
CMPCR0	2Ch
CMPCR1	12Ch

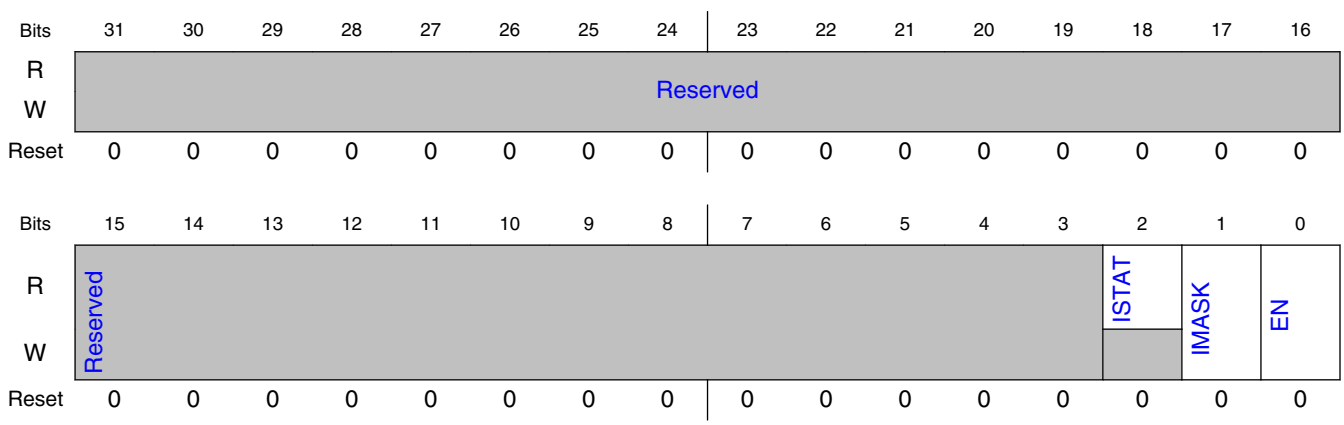
4.11.4.3.4.2 Function

The compare control register provides control and status of the compare function. When enabled, the ISTAT bit indicates whether the counter value is greater than or equal to the value in the compare value register (CMPCV). The ISTAT equation is:

ISTAT = (CNTCV >= CMPCV)

ISTAT takes no account of the value of the IMASK bit. If ISTAT is set to 1 and IMASK is 0, then the interrupt request is asserted. Clearing the enable bit (EN=0) will clear the status bit (ISTAT=0) and will negate the interrupt output signal.

4.11.4.3.4.3 Diagram



4.11.4.3.4.4 Fields

Field	Function
31-3 —	- Reserved
2 ISTAT	ISTAT Compare (interrupt) status 0b - Counter value is less than the compare value or compare is disabled. 1b - Counter value is greater than or equal to the compare value and compare is enabled.
1 IMASK	IMASK Interrupt request mask 0b - Interrupt output signal is not masked. 1b - Interrupt output signal is masked.
0 EN	EN Enable the compare function 0b - Compare disabled 1b - Compare enabled

4.11.4.3.5 Counter ID Register (CNTID0)

4.11.4.3.5.1 Offset

Register	Offset
CNTID0	FD0h

4.11.4.3.5.2 Function

The Counter ID register indicates the architecture version 0.

4.11.4.3.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CNTID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNTID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.11.4.3.5.4 Fields

Field	Function
31-0 CNTID	CNTID Counter Identification. Counter ID 0.

Chapter 5
Clocks and Power Management

5.1 Clock Control Module (CCM)

5.1.1 Overview

Clock Control Module (CCM) manages the on-chip module clocks. CCM receives clocks from PLLs and oscillators and creates clocks for on-chip peripherals through a set of multiplexers, dividers and gates. When entering or exiting a low power mode, CCM automatically turns on and off PLLs and peripheral clocks.

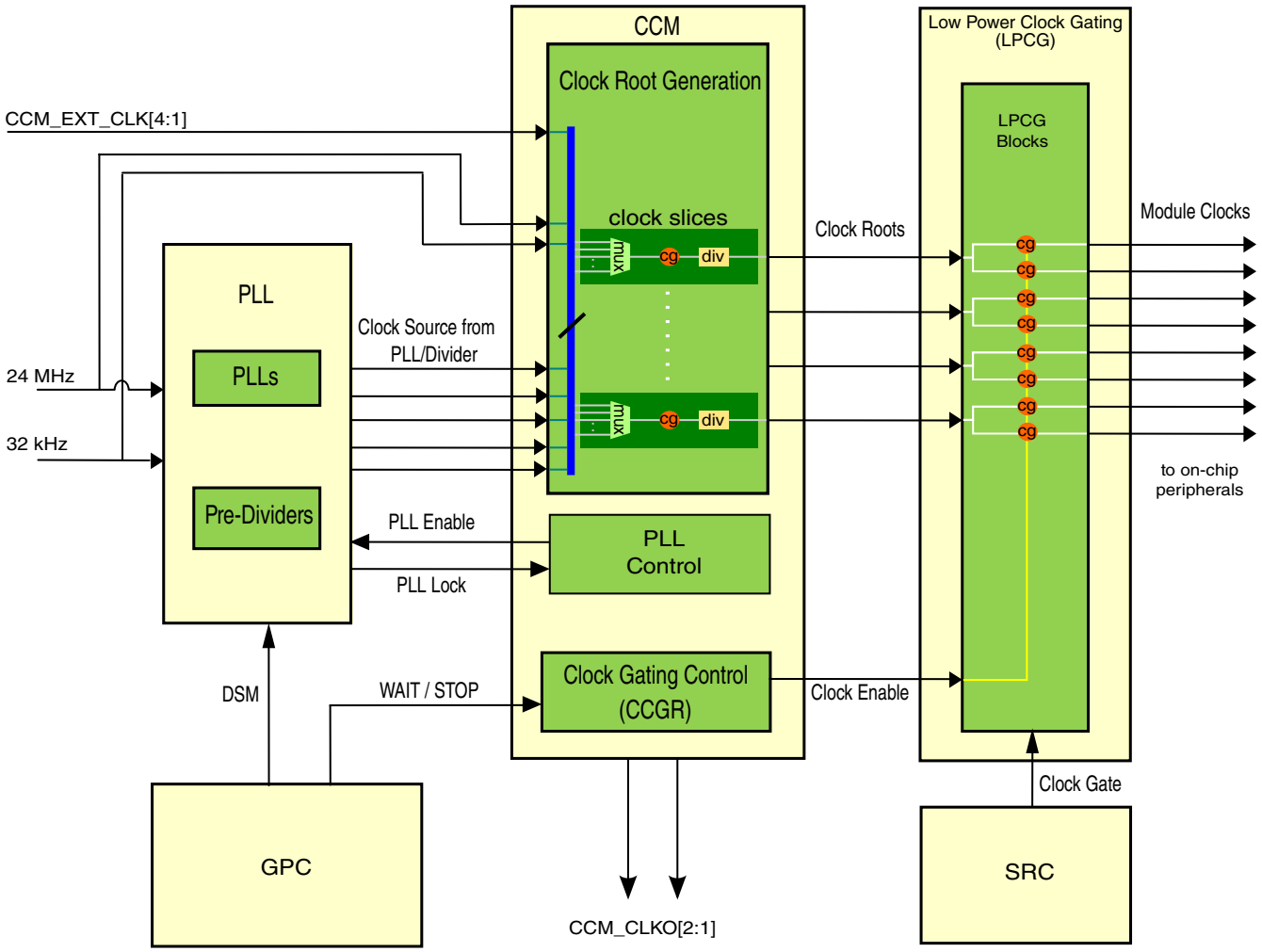


Figure 5-1. CCM Block Diagram

5.1.2 Clock Root Selects

The table below details the clock root slices and clock source selection inputs for each clock slice.

NOTE

The value of all clock root slice registers (CCM_TARGET_ROOTn) are zero after Power On Reset (POR) with the exception of DRAM_CLK_ROOT_SEL. Please see the System Boot chapter for ROM reset values and default frequency settings for the clock root slices.

Table 5-1. Clock Root Table

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
0 (Core)	ARM_A53_CLK_ROOT	0x8000	1000	000 - 24M_REF_CLK 001 - ARM_PLL_CLK 100 - SYSTEM_PLL1_CLK 101 - SYSTEM_PLL1_DIV2 011 - SYSTEM_PLL2_CLK 010 - SYSTEM_PLL2_DIV2 111 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL1_CLK
1 (Core)	ARM_M7_CLK_ROOT	0x8080	800	000 - 24M_REF_CLK 011 - VPU_PLL_CLK 100 - SYSTEM_PLL1_CLK 010 - SYSTEM_PLL2_DIV4 001 - SYSTEM_PLL2_DIV5 111 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL_CLK
2 (Core)	ML_CLK_ROOT	0x8100	1000	000 - 24M_REF_CLK 001 - GPU_PLL_CLK 010 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 011 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 111 - AUDIO_PLL2_CLK 110 - VIDEO_PLL_CLK
3 (Core)	GPU3D_CORE_CLK_ROOT	0x8180	1000	000 - 24M_REF_CLK 001 - GPU_PLL_CLK 010 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 011 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 111 - AUDIO_PLL2_CLK 110 - VIDEO_PLL_CLK
4 (Core)	GPU3D_SHADER_CLK_ROOT	0x8200	1000	000 - 24M_REF_CLK 001 - GPU_PLL_CLK 010 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				011 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 111 - AUDIO_PLL2_CLK 110 - VIDEO_PLL_CLK
5 (Core)	GPU2D_CLK_ROOT	0x8280	1000	000 - 24M_REF_CLK 001 - GPU_PLL_CLK 010 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 011 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 111 - AUDIO_PLL2_CLK 110 - VIDEO_PLL_CLK
6 (Core)	AUDIO_AXI_CLK_ROOT	0x8300	800	000 - 24M_REF_CLK 001 - GPU_PLL_CLK 010 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 011 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 111 - AUDIO_PLL2_CLK 110 - VIDEO_PLL_CLK
7 (Core)	HSIO_AXI_CLK_ROOT	0x8380	500	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_CLK 001 - SYSTEM_PLL2_DIV2 100 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL2_DIV10 111 - AUDIO_PLL2_CLK 101 - EXT_CLK_2 110 - EXT_CLK_4
8 (Core)	MEDIA_ISP_CLK_ROOT	0x8400	500	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_CLK 111 - SYSTEM_PLL2_DIV2 011 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL2_CLK 110 - EXT_CLK_1
16 (Bus)	MAIN_AXI_CLK_ROOT	0x8800	400	000 - 24M_REF_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				010 - SYSTEM_PLL1_CLK 111 - SYSTEM_PLL1_DIV8 100 - SYSTEM_PLL2_CLK 001 - SYSTEM_PLL2_DIV3 011 - SYSTEM_PLL2_DIV4 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL_CLK
17 (Bus)	ENET_AXI_CLK_ROOT	0x8880	266	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_CLK 001 - SYSTEM_PLL1_DIV3 011 - SYSTEM_PLL2_DIV4 100 - SYSTEM_PLL2_DIV5 111 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL_CLK
18 (Bus)	NAND_USDHC_BUS_CLK_ROOT	0x8900	266	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_CLK 001 - SYSTEM_PLL1_DIV3 100 - SYSTEM_PLL1_DIV6 110 - SYSTEM_PLL2_DIV4 011 - SYSTEM_PLL2_DIV5 101 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL1_CLK
19 (Bus)	VPU_BUS_CLK_ROOT	0x8980	800	000 - 24M_REF_CLK 010 - VPU_PLL_CLK 001 - SYSTEM_PLL1_CLK 111 - SYSTEM_PLL1_DIV8 101 - SYSTEM_PLL2_CLK 110 - SYSTEM_PLL2_DIV5 100 - SYSTEM_PLL3_CLK 011 - AUDIO_PLL2_CLK
20 (Bus)	MEDIA_AXI_CLK_ROOT	0x8A00	500	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_CLK 111 - SYSTEM_PLL2_DIV2 011 - SYSTEM_PLL3_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				101 - AUDIO_PLL2_CLK 110 - EXT_CLK_1
21 (Bus)	MEDIA_APB_CLK_ROOT	0x8A80	200	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_CLK 111 - SYSTEM_PLL1_DIV6 100 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_DIV8 011 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL2_CLK 110 - EXT_CLK_1
22 (Bus)	HDMI_APB_CLK_ROOT	0x8B00	200	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_CLK 111 - SYSTEM_PLL1_DIV6 100 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_DIV8 011 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL2_CLK 110 - EXT_CLK_1
23 (Bus)	HDMI_AXI_CLK_ROOT	0x8B80	500	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_CLK 111 - SYSTEM_PLL2_DIV2 011 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL2_CLK 110 - EXT_CLK_1
24 (Bus)	GPU_AXI_CLK_ROOT	0x8C00	800	000 - 24M_REF_CLK 010 - GPU_PLL_CLK 001 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 011 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 111 - AUDIO_PLL2_CLK 110 - VIDEO_PLL_CLK
25 (Bus)	GPU_AHB_CLK_ROOT	0x8C80	400	000 - 24M_REF_CLK 010 - GPU_PLL_CLK 001 - SYSTEM_PLL1_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				100 - SYSTEM_PLL2_CLK 011 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 111 - AUDIO_PLL2_CLK 110 - VIDEO_PLL_CLK
26 (Bus)	NOC_CLK_ROOT	0x8D00	1000	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_CLK 100 - SYSTEM_PLL2_DIV2 010 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 111 - AUDIO_PLL2_CLK 110 - VIDEO_PLL_CLK
27 (Bus)	NOC_IO_CLK_ROOT	0x8D80	800	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_CLK 100 - SYSTEM_PLL2_DIV2 010 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 111 - AUDIO_PLL2_CLK 110 - VIDEO_PLL_CLK
28 (Bus)	ML_AXI_CLK_ROOT	0x8E00	800	000 - 24M_REF_CLK 010 - GPU_PLL_CLK 001 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 011 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 111 - AUDIO_PLL2_CLK 110 - VIDEO_PLL_CLK
29 (Bus)	ML_AHB_CLK_ROOT	0x8E80	400	000 - 24M_REF_CLK 010 - GPU_PLL_CLK 001 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 011 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 111 - AUDIO_PLL2_CLK 110 - VIDEO_PLL_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
32 (Bus)	AHB_CLK_ROOT	0x9000	133	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL1_DIV2 001 - SYSTEM_PLL1_DIV6 100 - SYSTEM_PLL2_DIV8 101 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL1_CLK 111 - VIDEO_PLL_CLK
33 (Bus)	IPG_CLK_ROOT	0x9080	133	xxx - AHB_CLK_ROOT
34 (Bus)	AUDIO_AHB_CLK_ROOT	0x9100	400	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_CLK 001 - SYSTEM_PLL2_DIV2 100 - SYSTEM_PLL2_DIV6 101 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL1_CLK 111 - VIDEO_PLL_CLK
38 (Bus)	MEDIA_DISP2_CLK_ROOT	0x9300	160	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_CLK 101 - SYSTEM_PLL2_CLK 110 - SYSTEM_PLL3_CLK 011 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 001 - VIDEO_PLL_CLK 111 - EXT_CLK_4
48	DRAM_CLK_ROOT_SEL	0x9800	800	000 - DRAM_PLL_CLK 001 - DRAM_ALT_CLK_ROOT
49	ARM_A53_CLK_ROOT_SEL	0x9880	1200	000 - ARM_PLL_CLK 100 - ARM_A53_CLK_ROOT
64 (Peripheral)	DRAM_ALT_CLK_ROOT	0xA000	800	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_CLK 111 - SYSTEM_PLL1_DIV3 010 - SYSTEM_PLL1_DIV8 100 - SYSTEM_PLL2_CLK 011 - SYSTEM_PLL2_DIV2 101 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL1_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
65 (Peripheral)	DRAM_APB_CLK_ROOT	0xA080	200	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL1_DIV5 010 - SYSTEM_PLL1_DIV20 110 - SYSTEM_PLL2_DIV4 001 - SYSTEM_PLL2_DIV5 101 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL2_CLK
66 (Peripheral)	VPU_G1_CLK_ROOT	0xA100	800	000 - 24M_REF_CLK 001 - VPU_PLL_CLK 010 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL1_DIV8 011 - SYSTEM_PLL2_CLK 101 - SYSTEM_PLL2_DIV8 110 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL1_CLK
67 (Peripheral)	VPU_G2_CLK_ROOT	0xA180	700	000 - 24M_REF_CLK 001 - VPU_PLL_CLK 010 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL1_DIV8 011 - SYSTEM_PLL2_CLK 101 - SYSTEM_PLL2_DIV8 110 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL1_CLK
68 (Peripheral)	CAN1_CLK_ROOT	0xA200	80	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL1_DIV5 010 - SYSTEM_PLL1_DIV20 110 - SYSTEM_PLL2_DIV4 001 - SYSTEM_PLL2_DIV5 101 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL2_CLK
69 (Peripheral)	CAN2_CLK_ROOT	0xA280	80	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL1_DIV5 010 - SYSTEM_PLL1_DIV20

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				110 - SYSTEM_PLL2_DIV4 001 - SYSTEM_PLL2_DIV5 101 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL2_CLK
70 (Peripheral)	MEMREPAIR_CLK_ROOT	0xA300	50	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_DIV5 111 - SYSTEM_PLL1_DIV6 010 - SYSTEM_PLL2_DIV20 011 - SYSTEM_PLL3_CLK 100 - AUDIO_PLL1_CLK 110 - AUDIO_PLL2_CLK 101 - VIDEO_PLL_CLK
71 (Peripheral)	PCIE_PHY_CLK_ROOT	0xA380	250	000 - 24M_REF_CLK 111 - SYSTEM_PLL1_DIV2 010 - SYSTEM_PLL2_DIV2 001 - SYSTEM_PLL2_DIV10 011 - EXT_CLK_1 100 - EXT_CLK_2 101 - EXT_CLK_3 110 - EXT_CLK_4
72 (Peripheral)	PCIE_AUX_CLK_ROOT	0xA400	10	000 - 24M_REF_CLK 111 - SYSTEM_PLL1_DIV4 110 - SYSTEM_PLL1_DIV5 101 - SYSTEM_PLL1_DIV10 001 - SYSTEM_PLL2_DIV5 100 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL2_DIV20 011 - SYSTEM_PLL3_CLK
73 (Peripheral)	I2C5_CLK_ROOT	0xA480	66	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_DIV5 111 - SYSTEM_PLL1_DIV6 010 - SYSTEM_PLL2_DIV20 011 - SYSTEM_PLL3_CLK 100 - AUDIO_PLL1_CLK 110 - AUDIO_PLL2_CLK 101 - VIDEO_PLL_CLK
74 (Peripheral)	I2C6_CLK_ROOT	0xA500	66	000 - 24M_REF_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				001 - SYSTEM_PLL1_DIV5 111 - SYSTEM_PLL1_DIV6 010 - SYSTEM_PLL2_DIV20 011 - SYSTEM_PLL3_CLK 100 - AUDIO_PLL1_CLK 110 - AUDIO_PLL2_CLK 101 - VIDEO_PLL_CLK
75 (Peripheral)	SAI1_CLK_ROOT	0xA580	66	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_DIV6 001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL_CLK 110 - EXT_CLK_1 111 - EXT_CLK_2 101 - Reserved
76 (Peripheral)	SAI2_CLK_ROOT	0xA600	66	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_DIV6 001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL_CLK 110 - EXT_CLK_2 111 - EXT_CLK_3 101 - Reserved
77 (Peripheral)	SAI3_CLK_ROOT	0xA680	66	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_DIV6 001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL_CLK 110 - EXT_CLK_3 111 - EXT_CLK_4 101 - Reserved
79 (Peripheral)	SAI5_CLK_ROOT	0xA780	66	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_DIV6 001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL_CLK 110 - EXT_CLK_2

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				111 - EXT_CLK_3 101 - Reserved
80 (Peripheral)	SAI6_CLK_ROOT	0xA800	66	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_DIV6 001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL_CLK 110 - EXT_CLK_3 111 - EXT_CLK_4 101 - Reserved
81 (Peripheral)	ENET_QOS_CLK_ROOT	0xA880	125	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_DIV5 001 - SYSTEM_PLL2_DIV8 011 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL2_DIV20 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL_CLK 111 - EXT_CLK_4
82 (Peripheral)	ENET_QOS_TIMER_CLK_ROOT	0xA900	200	000 - 24M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - AUDIO_PLL1_CLK 111 - VIDEO_PLL_CLK 011 - EXT_CLK_1 100 - EXT_CLK_2 101 - EXT_CLK_3 110 - EXT_CLK_4
83 (Peripheral)	ENET_REF_CLK_ROOT	0xA980	125	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_DIV5 001 - SYSTEM_PLL2_DIV8 011 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL2_DIV20 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL_CLK 111 - EXT_CLK_4
84 (Peripheral)	ENET_TIMER_CLK_ROOT	0xAA00	125	000 - 24M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - AUDIO_PLL1_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				111 - VIDEO_PLL_CLK 011 - EXT_CLK_1 100 - EXT_CLK_2 101 - EXT_CLK_3 110 - EXT_CLK_4
85 (Peripheral)	ENET_PHY_REF_CLK_ROOT	0xAA80	125	000 - 24M_REF_CLK 100 - SYSTEM_PLL2_DIV2 011 - SYSTEM_PLL2_DIV5 010 - SYSTEM_PLL2_DIV8 001 - SYSTEM_PLL2_DIV20 101 - AUDIO_PLL1_CLK 111 - AUDIO_PLL2_CLK 110 - VIDEO_PLL_CLK
86 (Peripheral)	NAND_CLK_ROOT	0xAB00	500	000 - 24M_REF_CLK 011 - SYSTEM_PLL1_DIV2 001 - SYSTEM_PLL2_DIV2 110 - SYSTEM_PLL2_DIV4 101 - SYSTEM_PLL3_CLK 010 - AUDIO_PLL1_CLK 100 - AUDIO_PLL2_CLK 111 - VIDEO_PLL_CLK
87 (Peripheral)	QSPI_CLK_ROOT	0xAB80	400	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_DIV2 101 - SYSTEM_PLL1_DIV3 111 - SYSTEM_PLL1_DIV8 011 - SYSTEM_PLL2_DIV2 010 - SYSTEM_PLL2_DIV3 110 - SYSTEM_PLL3_CLK 100 - AUDIO_PLL2_CLK
88 (Peripheral)	USDHC1_CLK_ROOT	0xAC00	400	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_CLK 001 - SYSTEM_PLL1_DIV2 101 - SYSTEM_PLL1_DIV3 111 - SYSTEM_PLL1_DIV8 011 - SYSTEM_PLL2_DIV2 100 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL2_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
89 (Peripheral)	USDHC2_CLK_ROOT	0xAC80	400	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_CLK 001 - SYSTEM_PLL1_DIV2 101 - SYSTEM_PLL1_DIV3 111 - SYSTEM_PLL1_DIV8 011 - SYSTEM_PLL2_DIV2 100 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL2_CLK
90 (Peripheral)	I2C1_CLK_ROOT	0xAD00	66	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_DIV5 111 - SYSTEM_PLL1_DIV6 010 - SYSTEM_PLL2_DIV20 011 - SYSTEM_PLL3_CLK 100 - AUDIO_PLL1_CLK 110 - AUDIO_PLL2_CLK 101 - VIDEO_PLL_CLK
91 (Peripheral)	I2C2_CLK_ROOT	0xAD80	66	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_DIV5 111 - SYSTEM_PLL1_DIV6 010 - SYSTEM_PLL2_DIV20 011 - SYSTEM_PLL3_CLK 100 - AUDIO_PLL1_CLK 110 - AUDIO_PLL2_CLK 101 - VIDEO_PLL_CLK
92 (Peripheral)	I2C3_CLK_ROOT	0xAE00	66	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_DIV5 111 - SYSTEM_PLL1_DIV6 010 - SYSTEM_PLL2_DIV20 011 - SYSTEM_PLL3_CLK 100 - AUDIO_PLL1_CLK 110 - AUDIO_PLL2_CLK 101 - VIDEO_PLL_CLK
93 (Peripheral)	I2C4_CLK_ROOT	0xAE80	66	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_DIV5 111 - SYSTEM_PLL1_DIV6 010 - SYSTEM_PLL2_DIV20

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				011 - SYSTEM_PLL3_CLK 100 - AUDIO_PLL1_CLK 110 - AUDIO_PLL2_CLK 101 - VIDEO_PLL_CLK
94 (Peripheral)	UART1_CLK_ROOT	0xAF00	80	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_DIV10 010 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL2_CLK 101 - EXT_CLK_2 110 - EXT_CLK_4
95 (Peripheral)	UART2_CLK_ROOT	0xAF80	80	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_DIV10 010 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL2_CLK 101 - EXT_CLK_2 110 - EXT_CLK_3
96 (Peripheral)	UART3_CLK_ROOT	0xB000	80	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_DIV10 010 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL2_CLK 101 - EXT_CLK_2 110 - EXT_CLK_4
97 (Peripheral)	UART4_CLK_ROOT	0xB080	80	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_DIV10 010 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL2_CLK 101 - EXT_CLK_2 110 - EXT_CLK_3
100 (Peripheral)	GIC_CLK_ROOT	0xB200	500	000 - 24M_REF_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				100 - SYSTEM_PLL1_CLK 010 - SYSTEM_PLL1_DIV20 101 - SYSTEM_PLL2_DIV2 001 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL2_DIV10 111 - AUDIO_PLL2_CLK 110 - EXT_CLK_4
101 (Peripheral)	ECSPI1_CLK_ROOT	0xB280	80	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL1_DIV5 010 - SYSTEM_PLL1_DIV20 110 - SYSTEM_PLL2_DIV4 001 - SYSTEM_PLL2_DIV5 101 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL2_CLK
102 (Peripheral)	ECSPI2_CLK_ROOT	0xB300	80	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL1_DIV5 010 - SYSTEM_PLL1_DIV20 110 - SYSTEM_PLL2_DIV4 001 - SYSTEM_PLL2_DIV5 101 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL2_CLK
103 (Peripheral)	PWM1_CLK_ROOT	0xB380	66	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_DIV5 110 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL3_CLK 111 - VIDEO_PLL_CLK 101 - EXT_CLK_1
104 (Peripheral)	PWM2_CLK_ROOT	0xB400	66	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_DIV5 110 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL3_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				111 - VIDEO_PLL_CLK 101 - EXT_CLK_1
105 (Peripheral)	PWM3_CLK_ROOT	0xB480	66	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_DIV5 110 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL3_CLK 111 - VIDEO_PLL_CLK 101 - EXT_CLK_2
106 (Peripheral)	PWM4_CLK_ROOT	0xB500	66	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_DIV5 110 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL3_CLK 111 - VIDEO_PLL_CLK 101 - EXT_CLK_2
107 (Peripheral)	GPT1_CLK_ROOT	0xB580	100	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_DIV2 101 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_DIV10 110 - AUDIO_PLL1_CLK 100 - VIDEO_PLL_CLK 111 - EXT_CLK_1
108 (Peripheral)	GPT2_CLK_ROOT	0xB600	100	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_DIV2 101 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_DIV10 110 - AUDIO_PLL1_CLK 100 - VIDEO_PLL_CLK 111 - EXT_CLK_2
109 (Peripheral)	GPT3_CLK_ROOT	0xB680	100	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_DIV2 101 - SYSTEM_PLL1_DIV10

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				011 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_DIV10 110 - AUDIO_PLL1_CLK 100 - VIDEO_PLL_CLK 111 - EXT_CLK_3
110 (Peripheral)	GPT4_CLK_ROOT	0xB700	100	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_DIV2 101 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_DIV10 110 - AUDIO_PLL1_CLK 100 - VIDEO_PLL_CLK 111 - EXT_CLK_1
111 (Peripheral)	GPT5_CLK_ROOT	0xB780	100	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_DIV2 101 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_DIV10 110 - AUDIO_PLL1_CLK 100 - VIDEO_PLL_CLK 111 - EXT_CLK_2
112 (Peripheral)	GPT6_CLK_ROOT	0xB800	100	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_DIV2 101 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_DIV20 001 - SYSTEM_PLL2_DIV10 110 - AUDIO_PLL1_CLK 100 - VIDEO_PLL_CLK 111 - EXT_CLK_3
113 (Peripheral)	TRACE_CLK_ROOT	0xB880	133	000 - 24M_REF_CLK 011 - VPU_PLL_CLK 010 - SYSTEM_PLL1_DIV5 001 - SYSTEM_PLL1_DIV6 100 - SYSTEM_PLL2_DIV8 101 - SYSTEM_PLL3_CLK 110 - EXT_CLK_1 111 - EXT_CLK_3

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
114 (Peripheral)	WDOG_CLK_ROOT	0xB900	66	000 - 24M_REF_CLK 011 - VPU_PLL_CLK 010 - SYSTEM_PLL1_DIV5 001 - SYSTEM_PLL1_DIV6 110 - SYSTEM_PLL1_DIV10 111 - SYSTEM_PLL2_DIV6 100 - SYSTEM_PLL2_DIV8 101 - SYSTEM_PLL3_CLK
115 (Peripheral)	WRCLK_CLK_ROOT	0xB980	40	000 - 24M_REF_CLK 010 - VPU_PLL_CLK 101 - SYSTEM_PLL1_DIV3 111 - SYSTEM_PLL1_DIV8 001 - SYSTEM_PLL1_DIV20 110 - SYSTEM_PLL2_DIV2 100 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL3_CLK
116 (Peripheral)	IPP_DO_CLKO1	0xBA00	266	000 - 24M_REF_CLK 110 - VPU_PLL_CLK 001 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL1_DIV4 010 - SYSTEM_PLL1_DIV6 111 - SYSTEM_PLL1_DIV10 101 - SYSTEM_PLL2_DIV2 100 - AUDIO_PLL2_CLK
117 (Peripheral)	IPP_DO_CLKO2	0xBA80	266	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_DIV2 001 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL2_DIV6 100 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL_CLK 111 - 32K_REF_CLK
118 (Peripheral)	HDMI_FDCC_TST_CLK_ROOT	0xBB00	600	000 - 24M_REF_CLK 011 - SYSTEM_PLL1_CLK 001 - SYSTEM_PLL1_DIV3 100 - SYSTEM_PLL2_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				010 - SYSTEM_PLL2_DIV4 101 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL2_CLK 111 - VIDEO_PLL_CLK
120 (Peripheral)	HDMI_REF_266M_CLK_ROOT	0xBC00	266	000 - 24M_REF_CLK 001 - SYSTEM_PLL1_DIV2 100 - SYSTEM_PLL1_DIV3 011 - SYSTEM_PLL2_DIV3 101 - SYSTEM_PLL2_DIV5 010 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL1_CLK 111 - VIDEO_PLL_CLK
121 (Peripheral)	USDHC3_CLK_ROOT	0xBC80	400	000 - 24M_REF_CLK 010 - SYSTEM_PLL1_CLK 001 - SYSTEM_PLL1_DIV2 101 - SYSTEM_PLL1_DIV3 111 - SYSTEM_PLL1_DIV8 011 - SYSTEM_PLL2_DIV2 100 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL2_CLK
122 (Peripheral)	MEDIA_CAM1_PIX_CLK_ROOT	0xBD00	500	000 - 24M_REF_CLK 011 - SYSTEM_PLL1_CLK 001 - SYSTEM_PLL1_DIV3 100 - SYSTEM_PLL2_CLK 010 - SYSTEM_PLL2_DIV4 101 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL2_CLK 111 - VIDEO_PLL_CLK
123 (Peripheral)	MEDIA_MIPI_PHY1_REF_CLK_ROOT	0xBD80	125	000 - 24M_REF_CLK 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 001 - SYSTEM_PLL2_DIV3 010 - SYSTEM_PLL2_DIV10 110 - AUDIO_PLL2_CLK 111 - VIDEO_PLL_CLK 101 - EXT_CLK_2

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
124 (Peripheral)	MEDIA_DISP1_PIX_CLK_ROOT	0xBE00	250	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_CLK 101 - SYSTEM_PLL2_CLK 110 - SYSTEM_PLL3_CLK 011 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 001 - VIDEO_PLL_CLK 111 - EXT_CLK_4
125 (Peripheral)	MEDIA_CAM2_PIX_CLK_ROOT	0xBE80	266	000 - 24M_REF_CLK 011 - SYSTEM_PLL1_CLK 001 - SYSTEM_PLL1_DIV3 100 - SYSTEM_PLL2_CLK 010 - SYSTEM_PLL2_DIV4 101 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL2_CLK 111 - VIDEO_PLL_CLK
126 (Peripheral)	MEDIA_LDB_CLK_ROOT	0xBF00	560	000 - 24M_REF_CLK 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 001 - SYSTEM_PLL2_DIV3 010 - SYSTEM_PLL2_DIV10 110 - AUDIO_PLL2_CLK 111 - VIDEO_PLL_CLK 101 - EXT_CLK_2
130 (Peripheral)	MEDIA_MIPI_TEST_BYTE_CLK	0xC100	200	000 - 24M_REF_CLK 111 - SYSTEM_PLL1_DIV4 110 - SYSTEM_PLL1_DIV5 101 - SYSTEM_PLL1_DIV10 001 - SYSTEM_PLL2_DIV5 100 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL2_DIV20 011 - SYSTEM_PLL3_CLK
131 (Peripheral)	ECSPI3_CLK_ROOT	0xC180	80	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL1_DIV5 010 - SYSTEM_PLL1_DIV20

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Slice Index *n*	Clock Root	Offset	Max Freq (MHz)	Source Select (CCM_TARGET_ROOTn[MUX])
				110 - SYSTEM_PLL2_DIV4 001 - SYSTEM_PLL2_DIV5 101 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL2_CLK
132 (Peripheral)	PDM_CLK_ROOT	0xC200	200	000 - 24M_REF_CLK 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 001 - SYSTEM_PLL2_DIV10 101 - SYSTEM_PLL3_CLK 010 - AUDIO_PLL1_CLK 111 - AUDIO_PLL2_CLK 110 - EXT_CLK_3
133 (Peripheral)	VPU_VC8000E_CLK_ROOT	0xC280	800	000 - 24M_REF_CLK 001 - VPU_PLL_CLK 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_CLK 101 - SYSTEM_PLL2_DIV8 110 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL1_CLK 100 - AUDIO_PLL2_CLK
134 (Peripheral)	SAI7_CLK_ROOT	0xC300	66	000 - 24M_REF_CLK 100 - SYSTEM_PLL1_DIV6 001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL_CLK 110 - EXT_CLK_3 111 - EXT_CLK_4 101 - Reserved

5.1.3 Clock Tree

The figure below illustrates the clock sources from the PLLs.

NOTE

Some clock gates illustrated below are symbolic of distributed clock gates (multiple) and not a sole clock gate. These clock slices typically source multiple IP (e.g. bus clocks).

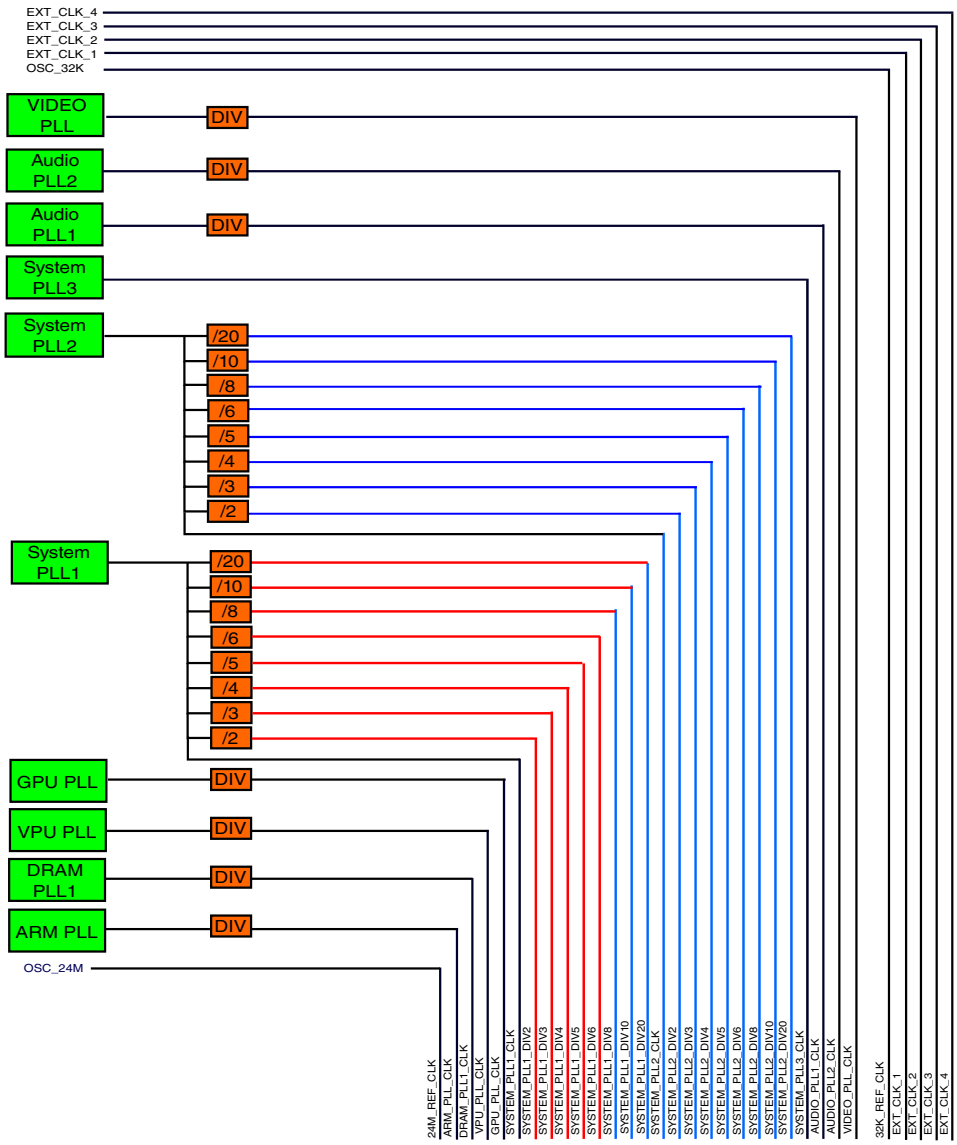
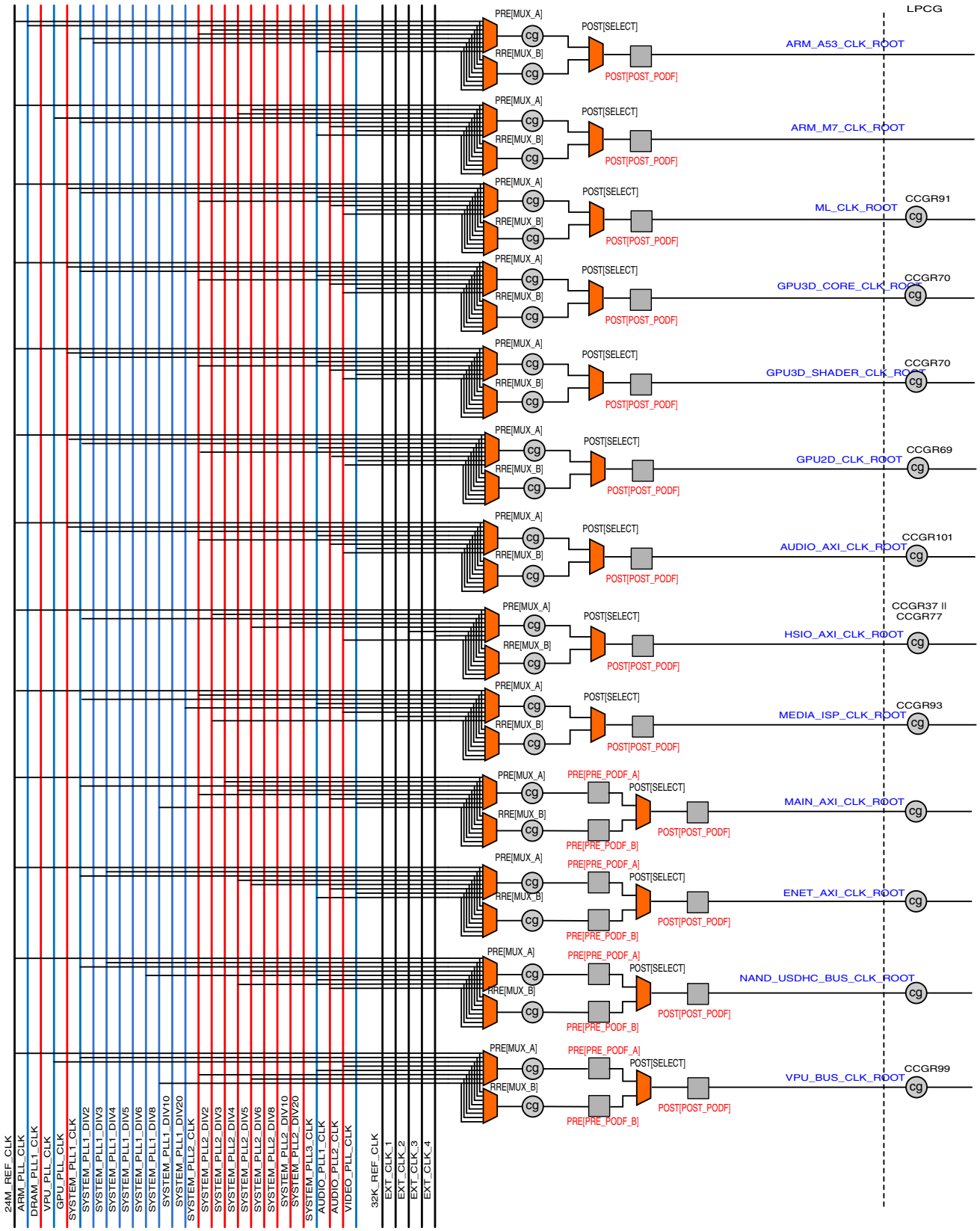
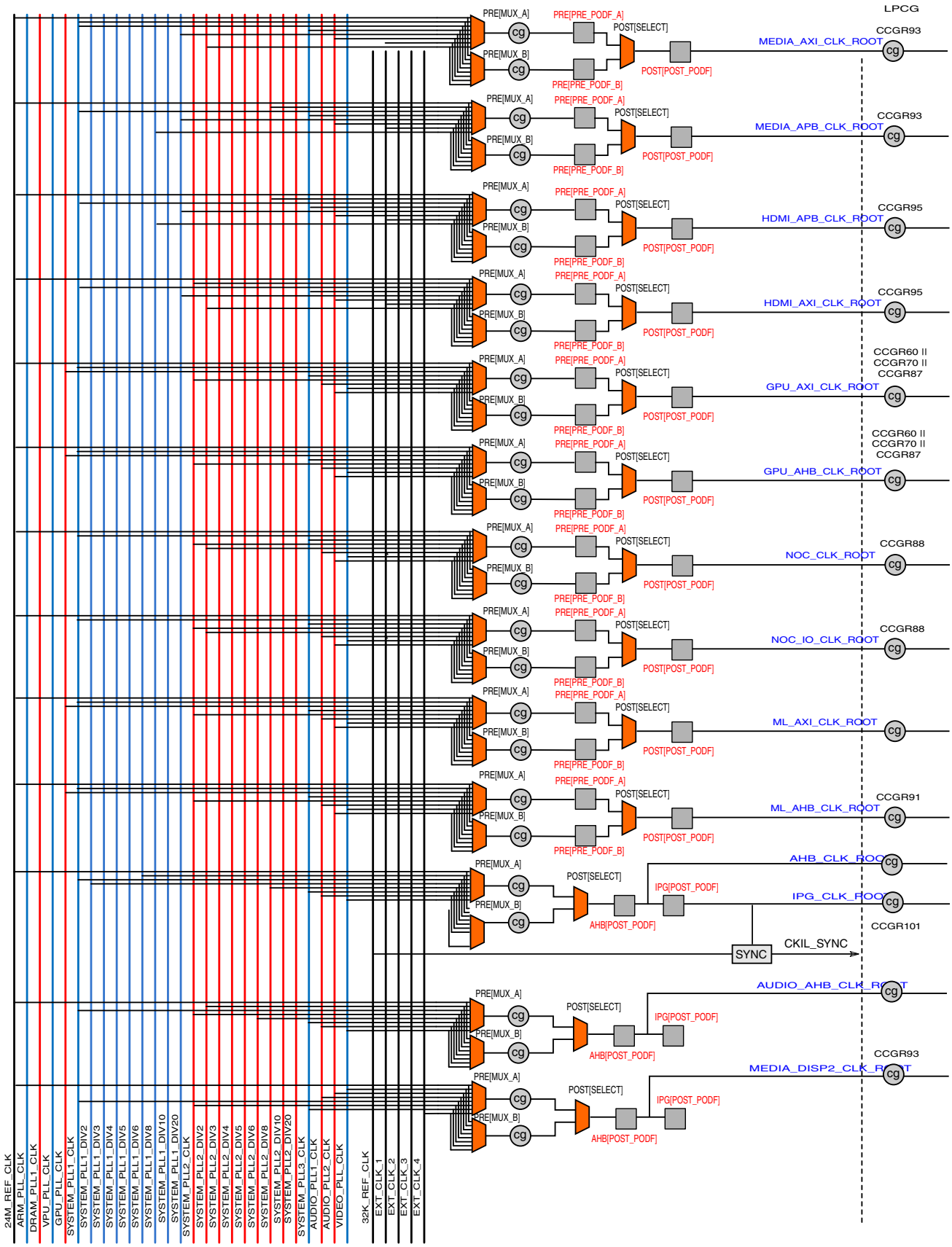
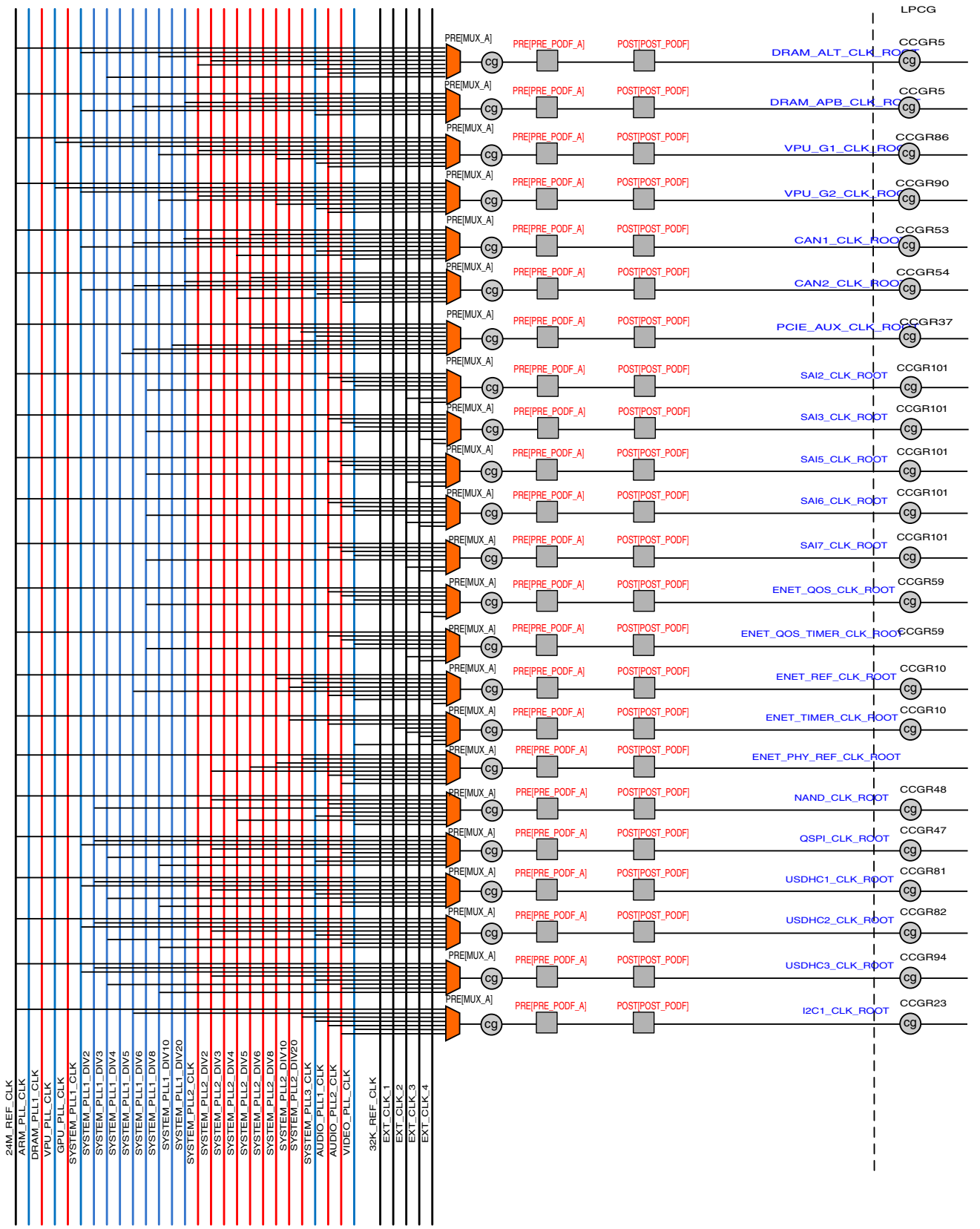


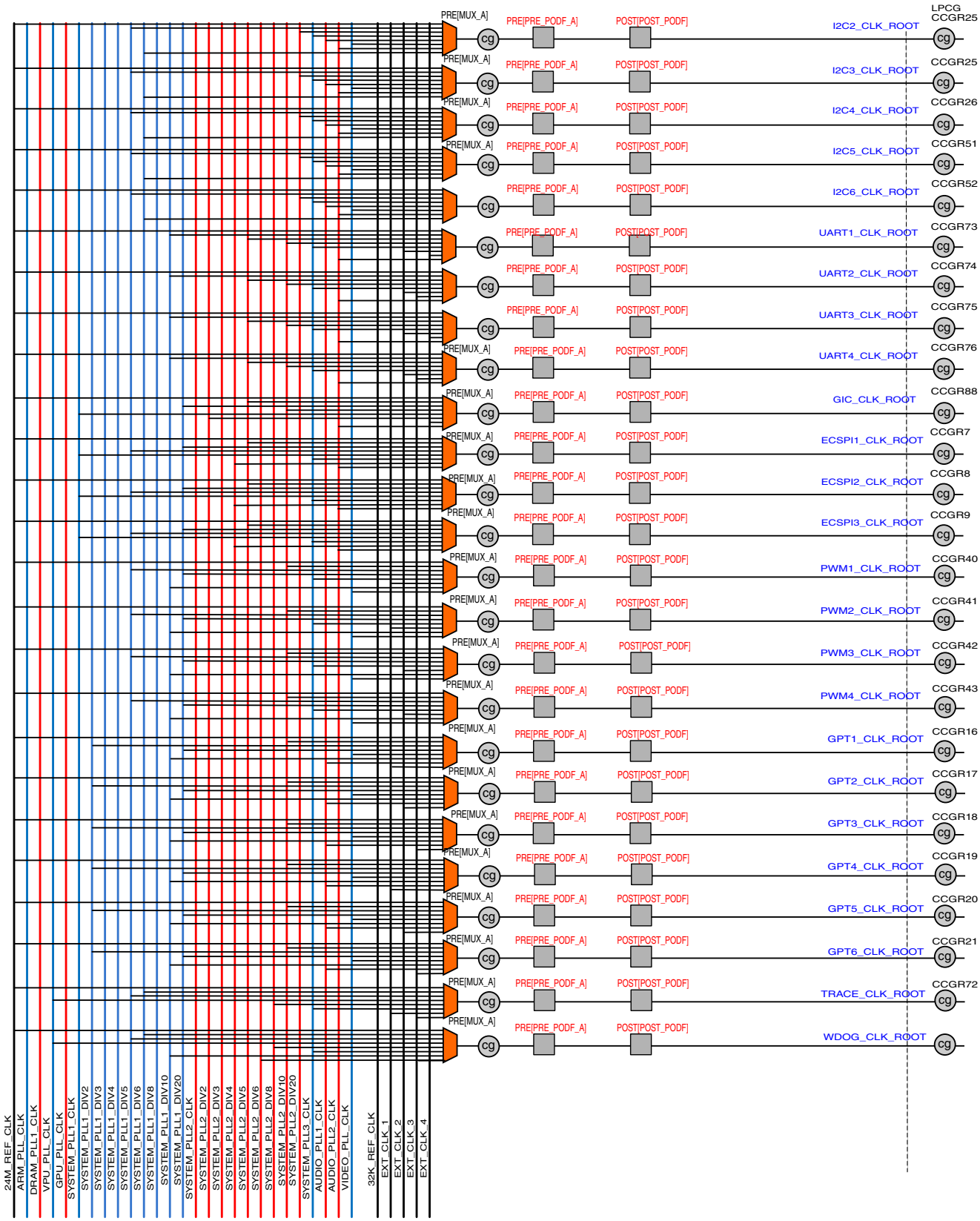
Figure 5-2. CCM input clock sources

The figure below illustrates the clock slices of CCM and clock root generation.


LPCG
PRE[MUX_A]
RRE[MUX_B]
cg
POST[SELECT]
POST[POST_PODF]
PRE[PRE_PODF_A]
PRE[PRE_PODF_B]
ARM_A53_CLK_ROOT
ARM_M7_CLK_ROOT
ML_CLK_ROOT
CCGR91
GPU3D_CORE_CLK_ROOT
CCGR70
GPU3D_SHADER_CLK_ROOT
CCGR70
GPU2D_CLK_ROOT
CCGR69
AUDIO_AXI_CLK_ROOT
CCGR101
HSIO_AXI_CLK_ROOT
CCGR37 II
CCGR77
MEDIA_ISP_CLK_ROOT
CCGR93
MAIN_AXI_CLK_ROOT
ENET_AXI_CLK_ROOT
NAND_USDHC_BUS_CLK_ROOT
VPU_BUS_CLK_ROOT
CCGR99
24M_REF_CLK
ARM_PLL_CLK
DRAM_PLL1_CLK
VPU_PLL_CLK
GPU_PLL_CLK
SYSTEM_PLL1_CLK
SYSTEM_PLL1_DIV2
SYSTEM_PLL1_DIV3
SYSTEM_PLL1_DIV4
SYSTEM_PLL1_DIV5
SYSTEM_PLL1_DIV6
SYSTEM_PLL1_DIV8
SYSTEM_PLL1_DIV10
SYSTEM_PLL1_DIV20
SYSTEM_PLL2_CLK
SYSTEM_PLL2_DIV2
SYSTEM_PLL2_DIV3
SYSTEM_PLL2_DIV4
SYSTEM_PLL2_DIV5
SYSTEM_PLL2_DIV6
SYSTEM_PLL2_DIV8
SYSTEM_PLL2_DIV10
SYSTEM_PLL2_DIV20
SYSTEM_PLL3_CLK
AUDIO_PLL1_CLK
AUDIO_PLL2_CLK
VIDEO_PLL_CLK
32K_REF_CLK
EXT_CLK_1
EXT_CLK_2
EXT_CLK_3
EXT_CLK_4


LPCG
CCGR93
MEDIA_AXI_CLK_ROOT
MEDIA_APB_CLK_ROOT
CCGR95
HDMI_APB_CLK_ROOT
HDMI_AXI_CLK_ROOT
CCGR60 II
CCGR70 II
CCGR87
GPU_AXI_CLK_ROOT
GPU_AHB_CLK_ROOT
CCGR88
NOC_CLK_ROOT
NOC_IO_CLK_ROOT
ML_AXI_CLK_ROOT
CCGR91
ML_AHB_CLK_ROOT
AHB_CLK_ROOT
IPG_CLK_ROOT
CCGR101
CKIL_SYNC
SYNC
AUDIO_AHB_CLK_ROOT
MEDIA_DISP2_CLK_ROOT
PRE[MUX_A]
PRE[MUX_B]
PRE[PRE_PODF_A]
PRE[PRE_PODF_B]
POST[SELECT]
POST[POST_PODF]
IPG[POST_PODF]
AHB[POST_PODF]
cg
24M_REF_CLK
ARM_PLL_CLK
DRAM_PLL1_CLK
VPU_PLL_CLK
GPU_PLL_CLK
SYSTEM_PLL1_CLK
SYSTEM_PLL1_DIV2
SYSTEM_PLL1_DIV3
SYSTEM_PLL1_DIV4
SYSTEM_PLL1_DIV5
SYSTEM_PLL1_DIV6
SYSTEM_PLL1_DIV8
SYSTEM_PLL1_DIV10
SYSTEM_PLL1_DIV20
SYSTEM_PLL2_CLK
SYSTEM_PLL2_DIV2
SYSTEM_PLL2_DIV3
SYSTEM_PLL2_DIV4
SYSTEM_PLL2_DIV5
SYSTEM_PLL2_DIV6
SYSTEM_PLL2_DIV8
SYSTEM_PLL2_DIV10
SYSTEM_PLL2_DIV20
SYSTEM_PLL3_CLK
AUDIO_PLL1_CLK
AUDIO_PLL2_CLK
VIDEO_PLL_CLK
32K_REF_CLK
EXT_CLK_1
EXT_CLK_2
EXT_CLK_3
EXT_CLK_4


LPCG
24M_REF_CLK
ARM_PLL_CLK
DRAM_PLL1_CLK
VPU_PLL_CLK
GPU_PLL_CLK
SYSTEM_PLL1_CLK
SYSTEM_PLL1_DIV2
SYSTEM_PLL1_DIV3
SYSTEM_PLL1_DIV4
SYSTEM_PLL1_DIV5
SYSTEM_PLL1_DIV6
SYSTEM_PLL1_DIV8
SYSTEM_PLL1_DIV10
SYSTEM_PLL1_DIV20
SYSTEM_PLL2_CLK
SYSTEM_PLL2_DIV2
SYSTEM_PLL2_DIV3
SYSTEM_PLL2_DIV4
SYSTEM_PLL2_DIV5
SYSTEM_PLL2_DIV6
SYSTEM_PLL2_DIV8
SYSTEM_PLL2_DIV10
SYSTEM_PLL2_DIV20
SYSTEM_PLL3_CLK
AUDIO_PLL1_CLK
AUDIO_PLL2_CLK
VIDEO_PLL_CLK
32K_REF_CLK
EXT_CLK_1
EXT_CLK_2
EXT_CLK_3
EXT_CLK_4
PRE[MUX_A]
PRE[PRE_PODF_A]
POST[POST_PODF]
cg
DRAM_ALT_CLK_ROOT
CCGR5
DRAM_APB_CLK_ROOT
CCGR5
VPU_G1_CLK_ROOT
CCGR86
VPU_G2_CLK_ROOT
CCGR90
CAN1_CLK_ROOT
CCGR53
CAN2_CLK_ROOT
CCGR54
PCIE_AUX_CLK_ROOT
CCGR37
SAI2_CLK_ROOT
CCGR101
SAI3_CLK_ROOT
CCGR101
SAI5_CLK_ROOT
CCGR101
SAI6_CLK_ROOT
CCGR101
SAI7_CLK_ROOT
CCGR101
ENET_QOS_CLK_ROOT
CCGR59
ENET_QOS_TIMER_CLK_ROOT
CCGR59
ENET_REF_CLK_ROOT
CCGR10
ENET_TIMER_CLK_ROOT
CCGR10
ENET_PHY_REF_CLK_ROOT
NAND_CLK_ROOT
CCGR48
QSPI_CLK_ROOT
CCGR47
USDHC1_CLK_ROOT
CCGR81
USDHC2_CLK_ROOT
CCGR82
USDHC3_CLK_ROOT
CCGR94
I2C1_CLK_ROOT
CCGR23


24M_REF_CLK
ARM_PLL_CLK
DRAM_PLL1_CLK
VPU_PLL_CLK
GPU_PLL_CLK
SYSTEM_PLL1_CLK
SYSTEM_PLL1_DIV2
SYSTEM_PLL1_DIV3
SYSTEM_PLL1_DIV4
SYSTEM_PLL1_DIV5
SYSTEM_PLL1_DIV6
SYSTEM_PLL1_DIV8
SYSTEM_PLL1_DIV10
SYSTEM_PLL1_DIV20
SYSTEM_PLL2_CLK
SYSTEM_PLL2_DIV2
SYSTEM_PLL2_DIV3
SYSTEM_PLL2_DIV4
SYSTEM_PLL2_DIV5
SYSTEM_PLL2_DIV6
SYSTEM_PLL2_DIV8
SYSTEM_PLL2_DIV10
SYSTEM_PLL2_DIV20
SYSTEM_PLL3_CLK
AUDIO_PLL1_CLK
AUDIO_PLL2_CLK
VIDEO_PLL_CLK
32K_REF_CLK
EXT_CLK_1
EXT_CLK_2
EXT_CLK_3
EXT_CLK_4
PRE[MUX_A]
cg
PRE[PRE_PODF_A]
POST[POST_PODF]
LPCG
I2C2_CLK_ROOT
CCGR25
I2C3_CLK_ROOT
CCGR25
I2C4_CLK_ROOT
CCGR26
I2C5_CLK_ROOT
CCGR51
I2C6_CLK_ROOT
CCGR52
UART1_CLK_ROOT
CCGR73
UART2_CLK_ROOT
CCGR74
UART3_CLK_ROOT
CCGR75
UART4_CLK_ROOT
CCGR76
GIC_CLK_ROOT
CCGR88
ECSPI1_CLK_ROOT
CCGR7
ECSPI2_CLK_ROOT
CCGR8
ECSPI3_CLK_ROOT
CCGR9
PWM1_CLK_ROOT
CCGR40
PWM2_CLK_ROOT
CCGR41
PWM3_CLK_ROOT
CCGR42
PWM4_CLK_ROOT
CCGR43
GPT1_CLK_ROOT
CCGR16
GPT2_CLK_ROOT
CCGR17
GPT3_CLK_ROOT
CCGR18
GPT4_CLK_ROOT
CCGR19
GPT5_CLK_ROOT
CCGR20
GPT6_CLK_ROOT
CCGR21
TRACE_CLK_ROOT
CCGR72
WDOG_CLK_ROOT

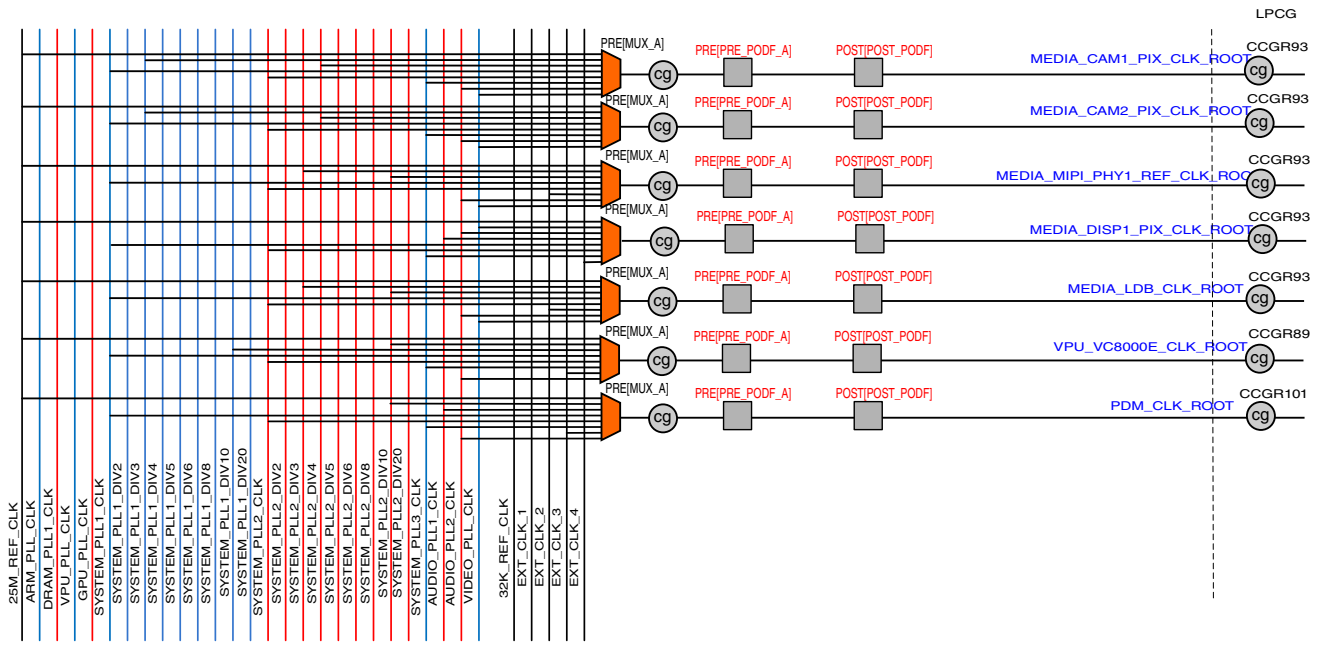


Figure 5-3. CCM Clock Tree Root Slices

5.1.4 System Clocks

The table below shows the CCM output module clock connectivity and gating.

Table 5-2. System Clocks

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
AIPSTZ	aips_tz1.hclk	AHB_CLK_ROOT	ccm_clk_enable_ipmux1 (CCGR28)
	aips_tz2.hclk	AHB_CLK_ROOT	ccm_clk_enable_ipmux2 (CCGR29)
	aips_tz3.hclk	AHB_CLK_ROOT	ccm_clk_enable_ipmux3 (CCGR30)
APBH (RAWNAND)	rawnand.u_bch_input_apb_clk	NAND_USDHC_BUS_CLK_ROOT	ccm_clk_enable_rawnand (CCGR48)
	rawnand.u_gpmi_input_apb_clk	NAND_USDHC_BUS_CLK_ROOT	ccm_clk_enable_rawnand (CCGR48)
APBHDMA	apbhdma.hclk	NAND_USDHC_BUS_CLK_ROOT	ccm_clk_enable_rawnand (CCGR48)
	apbhdma_sec.mst_hclk	NAND_USDHC_BUS_CLK_ROOT	ccm_clk_enable_rawnand (CCGR48)
ARM Platform	coresight.DBGCLK	MAIN_AXI_CLK_ROOT	ccm_clk_enable_trace (CCGR72)
	coresight.traceclkin	TRACE_CLK_ROOT	ccm_clk_enable_trace (CCGR72)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
	dap.dapclk_2_2	AHB_CLK_ROOT	ccm_clk_enable_debug (CCGR4)
	m7.CLKIN	ARM_M7_CLK_ROOT	ccm_clk_enable_m7
	m7.FCLK	ARM_M7_CLK_ROOT	ccm_clk_enable_m7
	m7.ATCLK	ARM_M7_CLK_ROOT	ccm_clk_enable_debug (CCGR4)
	m7.DAPCLK	AHB_CLK_ROOT	ccm_clk_enable_debug (CCGR4)
	m7_sec.mst_hclk	ARM_M7_CLK_ROOT	ccm_clk_enable_m7
ASRC (AUDIOMIX)	asrc_ipg_clk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	asrc_ipg_gatedclk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
BCH (RAWNAND)	rawnand.u_gpmi_bch_input_bch_clk	NAND_USDHC_BUS_CLK_ROOT	ccm_clk_enable_rawnand (CCGR48)
CAAM	caam.aclk	AHB_CLK_ROOT	-
	caam.ipg_clk	IPG_CLK_ROOT	-
	caam.ipg_clk_s	IPG_CLK_ROOT	-
CCM (CCMSRCGPCMIX)	ccmsrcgpcmix.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_mrpr (CCGR100)
DDR (DDRMIX)	ddrmix.dram_alt_clk_root	DRAM_ALT_CLK_ROOT	ccm_clk_enable_dram1 (CCGR5)
	ddrmix.dram_apb_clk_root	DRAM_APB_CLK_ROOT	ccm_clk_enable_dram1 (CCGR5)
eARC (AUDIOMIX)	earc_ipg_earc_clk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	earc_phy_ai_clk	SYS_OSC_CLK	ccm_clk_enable_audio (CCGR101)
	earc_phy_audio_ss_clk	SAI_PLL_DIV2.CLK_OUT	ccm_clk_enable_audio (CCGR101)
ECSPI	ecspi1.ipg_clk_per	ECSPI1_CLK_ROOT	ccm_clk_enable_ecspi1 (CCGR7)
	ecspi1.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_ecspi1 (CCGR7)
	ecspi1.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_ecspi1 (CCGR7)
	ecspi2.ipg_clk_per	ECSPI2_CLK_ROOT	ccm_clk_enable_ecspi2 (CCGR8)
	ecspi2.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_ecspi2 (CCGR8)
	ecspi2.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_ecspi2 (CCGR8)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
	ecspi3.ipg_clk_per	ECSPI3_CLK_ROOT	ccm_clk_enable_ecspi3 (CCGR9)
	ecspi3.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_ecspi3 (CCGR9)
	ecspi3.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_ecspi3 (CCGR9)
eDMA (AUDIOMIX)	edma_ipg_clk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	asrc_ipg_gatedclk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
ENET_QOS	enet_qos.aclk_i	ENET_AXI_CLK_ROOT	ccm_clk_enable_enet_qos (CCGR59)
	enet_qos.clk_csr_i	ENET_AXI_CLK_ROOT	ccm_clk_enable_enet_qos (CCGR59)
	enet_qos.clk_ptp_ref_i	ENET_QOS_TIMER_CLK_ROOT	ccm_clk_enable_enet_qos (CCGR59)
	enet_qos_mem.mem_clk	ENET_AXI_CLK_ROOT	ccm_clk_enable_enet_qos (CCGR59)
	enet_qos_mem.clk_ptp_ref_i	ENET_QOS_TIMER_CLK_ROOT	ccm_clk_enable_enet_qos (CCGR59)
ENETn	enet1.ipg_clk	ENET_AXI_CLK_ROOT	ccm_clk_enable_enet1 (CCGR10)
	enet1.ipg_clk_mac0	ENET_AXI_CLK_ROOT	ccm_clk_enable_enet1 (CCGR10)
	enet1.ipg_clk_mac0_s	ENET_AXI_CLK_ROOT	ccm_clk_enable_enet1 (CCGR10)
	enet1.ipg_clk_s	ENET_AXI_CLK_ROOT	ccm_clk_enable_enet1 (CCGR10)
	enet1.ipp_ind_mac0_txclk	ENET_REF_CLK_ROOT	ccm_clk_enable_enet1 (CCGR10)
	enet1.ipg_clk_time	ENET_TIMER_CLK_ROOT	ccm_clk_enable_enet1 (CCGR10)
	enet1_mem.mac0_rxmem_clk	ENET_AXI_CLK_ROOT	ccm_clk_enable_enet1 (CCGR10)
	enet1_mem.mac0_txmem_clk	ENET_AXI_CLK_ROOT	ccm_clk_enable_enet1 (CCGR10)
FlexCAN	can1.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_can1 (CCGR53)
	can1.ipg_clk_chi	IPG_CLK_ROOT	ccm_clk_enable_can1 (CCGR53)
	can1.ipg_clk_pe	CAN1_PECLK_MUXED	ccm_clk_enable_can1 (CCGR53)
	can1.ipg_clk_pe_nogate	CAN1_PECLK_MUXED	ccm_clk_enable_can1 (CCGR53)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
	can1.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_can1 (CCGR53)
	can1.ipt_clk	ANAMIX_OSC_24M_CLK	ccm_clk_enable_can1 (CCGR53)
	can1.osc_clk	ANAMIX_OSC_24M_CLK	ccm_clk_enable_can1 (CCGR53)
	can1_mem.can_hclk	IPG_CLK_ROOT	ccm_clk_enable_can1 (CCGR53)
	can2.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_can2 (CCGR54)
	can2.ipg_clk_chi	IPG_CLK_ROOT	ccm_clk_enable_can2 (CCGR54)
	can2.ipg_clk_pe	CAN2_PECLK_MUXED	ccm_clk_enable_can2 (CCGR54)
	can2.ipg_clk_pe_nogate	CAN2_PECLK_MUXED	ccm_clk_enable_can2 (CCGR54)
	can2.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_can2 (CCGR54)
	can2.ipt_clk	ANAMIX_OSC_24M_CLK	ccm_clk_enable_can2 (CCGR54)
	can2.osc_clk	ANAMIX_OSC_24M_CLK	ccm_clk_enable_can2 (CCGR54)
	can2_mem.can_hclk	IPG_CLK_ROOT	ccm_clk_enable_can2 (CCGR54)
FlexSPI	flexspi.hclk	AHB_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)
	flexspi.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)
	flexspi.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)
	flexspi.ipg_clk_sfck	QSPI_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)
	flexspi_mem.AHBRX_CLKA	AHB_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)
	flexspi_mem.AHBRX_CLKB	AHB_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)
	flexspi_mem.IPTX_CLKA	AHB_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)
	flexspi_mem.IPTX_CLKB	AHB_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)
	flexspi_mem.IPRX_CLKA	AHB_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)
	flexspi_mem.IPRX_CLKB	AHB_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
	flexspi_mem.LUT_CLKA	AHB_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)
	flexspi_mem.LUT_CLKB	AHB_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)
	flexspi_sec.mst_hclk	AHB_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)
	flexspi_sec.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_flexspi (CCGR47)
GPC (CCMSRCGPCMIX)	ccmsrcgpcmix.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_mrpr (CCGR100)
	ccmsrcgpcmix.ckil_clk	CKIL_SYNC_CLK_ROOT	ccm_clk_enable_mrpr (CCGR100)
GPIO	gpio1.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_gpio1 (CCGR11)
	gpio2.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_gpio2 (CCGR12)
	gpio3.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_gpio3 (CCGR13)
	gpio4.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_gpio4 (CCGR14)
	gpio5.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_gpio5 (CCGR15)
GPMI (RAWNAND)	rawnand.u_gpmi_bch_input_gpmi_io_clk	NAND_CLK_ROOT	ccm_clk_enable_rawnand (CCGR48)
GPT	gpt1.ipg_clk	GPT1_CLK_ROOT	ccm_clk_enable_gpt1 (CCGR16)
	gpt1.ipg_clk_highfreq	GPT1_CLK_ROOT	ccm_clk_enable_gpt1 (CCGR16)
	gpt1.ipg_clk_s	GPT1_CLK_ROOT	ccm_clk_enable_gpt1 (CCGR16)
	gpt1.ipg_clk_24m	ANAMIX_OSC_24M_CLK	ccm_clk_enable_gpt1 (CCGR16)
	gpt2.ipg_clk	GPT2_CLK_ROOT	ccm_clk_enable_gpt2 (CCGR17)
	gpt2.ipg_clk_highfreq	GPT2_CLK_ROOT	ccm_clk_enable_gpt2 (CCGR17)
	gpt2.ipg_clk_s	GPT2_CLK_ROOT	ccm_clk_enable_gpt2 (CCGR17)
	gpt2.ipg_clk_24m	ANAMIX_OSC_24M_CLK	ccm_clk_enable_gpt2 (CCGR17)
	gpt3.ipg_clk	GPT3_CLK_ROOT	ccm_clk_enable_gpt3 (CCGR18)
	gpt3.ipg_clk_highfreq	GPT3_CLK_ROOT	ccm_clk_enable_gpt3 (CCGR18)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
	gpt3.ipg_clk_s	GPT3_CLK_ROOT	ccm_clk_enable_gpt3 (CCGR18)
	gpt3.ipg_clk_24m	ANAMIX_OSC_24M_CLK	ccm_clk_enable_gpt3 (CCGR18)
	gpt4.ipg_clk	GPT4_CLK_ROOT	ccm_clk_enable_gpt4 (CCGR19)
	gpt4.ipg_clk_highfreq	GPT4_CLK_ROOT	ccm_clk_enable_gpt4 (CCGR19)
	gpt4.ipg_clk_s	GPT4_CLK_ROOT	ccm_clk_enable_gpt4 (CCGR19)
	gpt4.ipg_clk_24m	ANAMIX_OSC_24M_CLK	ccm_clk_enable_gpt4 (CCGR19)
	gpt5.ipg_clk	GPT5_CLK_ROOT	ccm_clk_enable_gpt5 (CCGR20)
	gpt5.ipg_clk_highfreq	GPT5_CLK_ROOT	ccm_clk_enable_gpt5 (CCGR20)
	gpt5.ipg_clk_s	GPT5_CLK_ROOT	ccm_clk_enable_gpt5 (CCGR20)
	gpt5.ipg_clk_24m	ANAMIX_OSC_24M_CLK	ccm_clk_enable_gpt5 (CCGR20)
	gpt6.ipg_clk	GPT6_CLK_ROOT	ccm_clk_enable_gpt6 (CCGR21)
	gpt6.ipg_clk_highfreq	GPT6_CLK_ROOT	ccm_clk_enable_gpt6 (CCGR21)
	gpt6.ipg_clk_s	GPT6_CLK_ROOT	ccm_clk_enable_gpt6 (CCGR21)
	gpt6.ipg_clk_24m	ANAMIX_OSC_24M_CLK	ccm_clk_enable_gpt6 (CCGR21)
GPU 520L (GPUMIX)	gpumix.ccm_gpu_ahb_clk_root	GPU_AHB_CLK_ROOT	ccm_clk_enable_gpu_all ccm_clk_enable_gpu2d (CCGR69) ccm_clk_enable_gpu (CCGR87)
	gpumix.ccm_gpu_axi_clk_root	GPU_AXI_CLK_ROOT	ccm_clk_enable_gpu_all ccm_clk_enable_gpu2d (CCGR69) ccm_clk_enable_gpu (CCGR87)
	gpumix.ccm_gpu_g2d2x_clk_root	GPU2D_CLK_ROOT	ccm_clk_enable_gpu2d (CCGR69)
GPU 7000UL (GPUMIX)	gpumix.ccm_gpu_ahb_clk_root	GPU_AHB_CLK_ROOT	ccm_clk_enable_gpu_all ccm_clk_enable_gpu3d (CCGR70)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
			ccm_clk_enable_gpu (CCGR87)
	gpumix.ccm_gpu_axi_clk_root	GPU_AXI_CLK_ROOT	ccm_clk_enable_gpu_all ccm_clk_enable_gpu3d (CCGR70) ccm_clk_enable_gpu (CCGR87)
	gpumix.ccm_gpu_g3d2x_clk_root	GPU3D_CORE_CLK_ROOT	ccm_clk_enable_gpu3d (CCGR70)
	gpumix.ccm_gpu_g3dsh_clk_root	GPU3D_SHADER_CLK_ROOT	ccm_clk_enable_gpu3d (CCGR70)
HDMI TRNG Controller (HDMIMIX)	clk	HDMI_APB_CLK	ccm_clk_enable_hdmi (CCGR95)
HDMI TX Controller (HDMIMIX)	iapb_hpi_clk	HDMI_APB_CLK	ccm_clk_enable_hdmi (CCGR95)
	iapbclk	HDMI_APB_CLK	ccm_clk_enable_hdmi (CCGR95)
	iesmclk	HDMI_HTX_REF266M_CLK	ccm_clk_enable_hdmi (CCGR95)
HDMI TX BLK CTRL (HDMIMIX)	apb_clk	HDMI_APB_CLK	ccm_clk_enable_hdmi (CCGR95)
	apb_clkg	HDMI_APB_CLK	ccm_clk_enable_hdmi (CCGR95)
HIFI 4 DSP (AUDIOMIX)	hifi4cluster_hifi_core__CLK	AUDIO_HIFI4_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	hifi4cluster_hifi_dbg_apbs__pclkm	AUDIO_HIFI4_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
HTX PAI (HDMIMIX)	apb_clk	HDMI_APB_CLK	ccm_clk_enable_hdmi (CCGR95)
	apb_gatedclk	HDMI_APB_CLK	ccm_clk_enable_hdmi (CCGR95)
HTX PVI (HDMIMIX)	apb_clk	HDMI_APB_CLK	ccm_clk_enable_hdmi (CCGR95)
	apb_clkg	HDMI_APB_CLK	ccm_clk_enable_hdmi (CCGR95)
I2Cn	i2c1.ipg_clk_patref	I2C1_CLK_ROOT	ccm_clk_enable_i2c1 (CCGR23)
	i2c1.ipg_clk_s	I2C1_CLK_ROOT	ccm_clk_enable_i2c1 (CCGR23)
	i2c2.ipg_clk_patref	I2C2_CLK_ROOT	ccm_clk_enable_i2c2 (CCGR24)
	i2c2.ipg_clk_s	I2C2_CLK_ROOT	ccm_clk_enable_i2c2 (CCGR24)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
	i2c3.ipg_clk_patref	I2C3_CLK_ROOT	ccm_clk_enable_i2c3 (CCGR25)
	i2c3.ipg_clk_s	I2C3_CLK_ROOT	ccm_clk_enable_i2c3 (CCGR25)
	i2c4.ipg_clk_patref	I2C4_CLK_ROOT	ccm_clk_enable_i2c4 (CCGR26)
	i2c4.ipg_clk_s	I2C4_CLK_ROOT	ccm_clk_enable_i2c4 (CCGR26)
	i2c5.ipg_clk_patref	I2C5_CLK_ROOT	ccm_clk_enable_i2c5 (CCGR51)
	i2c5.ipg_clk_s	I2C5_CLK_ROOT	ccm_clk_enable_i2c5 (CCGR51)
	i2c6.ipg_clk_patref	I2C6_CLK_ROOT	ccm_clk_enable_i2c6 (CCGR52)
	i2c6.ipg_clk_s	I2C6_CLK_ROOT	ccm_clk_enable_i2c6 (CCGR52)
IOMUXC	iomuxc.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_iomux (CCGR27)
	iomuxc_gpr.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_iomux (CCGR27)
IRQ_STEER	irq_steer.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_irq_steer (CCGR63)
IRQ_STEER (HDMIMIX)	ipg_clk	HDMI_APB_CLK	ccm_clk_enable_hdmi (CCGR95)
ISI (MEDIAMIX)	apb_clk	MEDIA_APB_CLK_ROOT	ccm_clk_enable_media (CCGR93)
	ipg_proc_clk[0]	MEDIA_AXI_CLK_ROOT	ccm_clk_enable_media (CCGR93)
	ipg_proc_clk[1]	MEDIA_AXI_CLK_ROOT	ccm_clk_enable_media (CCGR93)
	ipg_proc_ungated_clk	MEDIA_AXI_CLK_ROOT	ccm_clk_enable_media (CCGR93)
	axi_clk	MEDIA_AXI_CLK_ROOT	ccm_clk_enable_media (CCGR93)
ISP (MEDIAMIX)	mediamix.ccm_media_mix_isp_cor_clk_root	MEDIA_ISP_CLK_ROOT	ccm_clk_enable_media (CCGR93)
LCDIF (MEDIAMIX)	apb_clk	MEDIA_AXI_CLK_ROOT	ccm_clk_enable_media (CCGR93)
	pixel_clk	MEDIA_DISP1_PIX_CLK_ROOT	ccm_clk_enable_media (CCGR93)
	pixel_clk	MEDIA_DISP2_PIX_CLK_ROOT	ccm_clk_enable_media (CCGR93)
LCDIF (HDMIMIX)	apb_clk	HDMI_APB_CLK	ccm_clk_enable_hdmi (CCGR95)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
	b_clk	HDMI_B_CLK	ccm_clk_enable_hdmi (CCGR95)
LDB (MEDIAMIX)	mediamix.ccm_media_ldb_serial_clk	MEDIA_LDB_CLK_ROOT	ccm_clk_enable_media (CCGR93)
MICFIL (AUDIOMIX)	pdm_ipg_clk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_s	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_app_nonstop	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_app	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_inp	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dc	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_hb	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_fir	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_app_fifo	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_vad[0]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_vad_dec[0]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_vad_dec16[0]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_vad_ndec[0]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_vad_zcd[0]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_vad_noise[0]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec[0]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec16[0]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_fifo[0]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec[1]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec16[1]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
	pdm_ipg_clk_fifo[1]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec[2]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec16[2]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_fifo[2]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec[3]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec16[3]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_fifo[3]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec[4]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec16[4]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_fifo[4]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec[5]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec16[5]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_fifo[5]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec[6]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec16[6]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_fifo[6]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec[7]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_dec16[7]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_fifo[7]	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_decfil	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	pdm_ipg_clk_decfil_free	AUDIO_PDM_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
MIPI_DSI (MEDIAMIX)	mipi_dsi_CLKREF	MEDIA_MIPI_DSI_REF_CLK_ROOT	ccm_clk_enable_media (CCGR93)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
	mipi_dsi_I_PCLK	MEDIA_APB_CLK_ROOT	ccm_clk_enable_media (CCGR93)
MU	mu1.ipg_clk_dsp	IPG_CLK_ROOT	ccm_clk_enable_mu (CCGR33)
	mu1.ipg_clk_mcu	IPG_CLK_ROOT	ccm_clk_enable_mu (CCGR33)
	mu1.ipg_clk_s_dsp	IPG_CLK_ROOT	ccm_clk_enable_mu (CCGR33)
	mu1.ipg_clk_s_mcu	IPG_CLK_ROOT	ccm_clk_enable_mu (CCGR33)
MU (AUDIOMIX)	mu2_ipg_clk_dsp	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	mu2_ipg_clk_s_dsp	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	mu2_ipg_clk_mcu	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	mu2_ipg_clk_s_mcu	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	mu3_ipg_clk_dsp	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	mu3_ipg_clk_s_dsp	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	mu3_ipg_clk_mcu	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	mu3_ipg_clk_s_mcu	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
NPU	mlmix.ccm_npu_ahb_clk_root	ML_AHB_CLK_ROOT	ccm_clk_enable_npu (CCGR91)
	mlmix.ccm_npu_axi_clk_root	ML_AXI_CLK_ROOT	ccm_clk_enable_npu (CCGR91)
	mlmix.ccm_npu_npu2x_clk_root	ML_CLK_ROOT	ccm_clk_enable_npu (CCGR91)
OCOTP_CTRL	ocotp.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_ocotp (CCGR34)
	ocotp.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_ocotp (CCGR34)
OCRAM	ocram_mem.clk	MAIN_AXI_CLK_ROOT	ccm_clk_enable_ocram (CCGR35)
	ocram_s_mem.clk	AHB_CLK_ROOT	ccm_clk_enable_ocram_s (CCGR36)
OCRAM (AUDIOMIX)	ocram_a_clk	AUDIO_HIFI4_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
PCIe (HSIOMIX)	module_clk	HSIOMIX_BUS_CLK_ROOT	ccm_clk_enable_hsio (CCGR92)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
	ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_hsio (CCGR92)
	aux_clk	PCIE_AUX_CLK_ROOT	ccm_clk_enable_pcie (CCGR37)
PWM	pwm1.ipg_clk	PWM1_CLK_ROOT	ccm_clk_enable_pwm1 (CCGR40)
	pwm1.ipg_clk_highfreq	PWM1_CLK_ROOT	ccm_clk_enable_pwm1 (CCGR40)
	pwm1.ipg_clk_s	PWM1_CLK_ROOT	ccm_clk_enable_pwm1 (CCGR40)
	pwm2.ipg_clk	PWM2_CLK_ROOT	ccm_clk_enable_pwm2 (CCGR41)
	pwm2.ipg_clk_highfreq	PWM2_CLK_ROOT	ccm_clk_enable_pwm2 (CCGR41)
	pwm2.ipg_clk_s	PWM2_CLK_ROOT	ccm_clk_enable_pwm2 (CCGR41)
	pwm3.ipg_clk	PWM3_CLK_ROOT	ccm_clk_enable_pwm3 (CCGR42)
	pwm3.ipg_clk_highfreq	PWM3_CLK_ROOT	ccm_clk_enable_pwm3 (CCGR42)
	pwm3.ipg_clk_s	PWM3_CLK_ROOT	ccm_clk_enable_pwm3 (CCGR42)
	pwm4.ipg_clk	PWM4_CLK_ROOT	ccm_clk_enable_pwm4 (CCGR43)
	pwm4.ipg_clk_highfreq	PWM4_CLK_ROOT	ccm_clk_enable_pwm4 (CCGR43)
	pwm4.ipg_clk_s	PWM4_CLK_ROOT	ccm_clk_enable_pwm4 (CCGR43)
RDC	rdc.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_rdc (CCGR49)
	rdc.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_rdc (CCGR49)
SAIn (AUDIOMIX)	sai1_ipg_clk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai1_ipg_clk_s	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai1_sai_mclk_1	SAI__MCLK_MUX.SAI1_MCLK_1_ROOT	ccm_clk_enable_audio (CCGR101)
	sai1_sai_mclk_2	SAI__MCLK_MUX.SAI1_MCLK_2_ROOT	ccm_clk_enable_audio (CCGR101)
	sai1_sai_mclk_3	SAI_PLL_DIV2.CLK_OUT	ccm_clk_enable_audio (CCGR101)
	sai1_ipt_clk_sai_bclk	CCM_SAI1_CLK_ROOT	ccm_clk_enable_audio (CCGR101)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
	sai2_ipg_clk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai2_ipg_clk_s	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai2_sai_mclk_1	SAI__MCLK_MUX.SAI2_MCLK_1_ROOT	ccm_clk_enable_audio (CCGR101)
	sai2_sai_mclk_2	SAI__MCLK_MUX.SAI2_MCLK_2_ROOT	ccm_clk_enable_audio (CCGR101)
	sai2_sai_mclk_3	SAI_PLL_DIV2.CLK_OUT	ccm_clk_enable_audio (CCGR101)
	sai2_ipt_clk_sai_bclk	CCM_SAI2_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai3_ipg_clk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai3_ipg_clk_s	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai3_sai_mclk_1	SAI__MCLK_MUX.SAI3_MCLK_1_ROOT	ccm_clk_enable_audio (CCGR101)
	sai3_sai_mclk_2	SAI__MCLK_MUX.SAI3_MCLK_2_ROOT	ccm_clk_enable_audio (CCGR101)
	sai3_sai_mclk_3	SAI_PLL_DIV2.CLK_OUT	ccm_clk_enable_audio (CCGR101)
	sai3_ipt_clk_sai_bclk	CCM_SAI3_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai5_ipg_clk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai5_ipg_clk_s	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai5_sai_mclk_1	SAI__MCLK_MUX.SAI5_MCLK_1_ROOT	ccm_clk_enable_audio (CCGR101)
	sai5_sai_mclk_2	SAI__MCLK_MUX.SAI5_MCLK_2_ROOT	ccm_clk_enable_audio (CCGR101)
	sai5_sai_mclk_3	SAI_PLL_DIV2.CLK_OUT	ccm_clk_enable_audio (CCGR101)
	sai5_ipt_clk_sai_bclk	CCM_SAI5_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai6_ipg_clk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai6_ipg_clk_s	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai6_sai_mclk_1	SAI__MCLK_MUX.SAI6_MCLK_1_ROOT	ccm_clk_enable_audio (CCGR101)
	sai6_sai_mclk_2	SAI__MCLK_MUX.SAI6_MCLK_2_ROOT	ccm_clk_enable_audio (CCGR101)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
	sai6_sai_mclk_3	SAI_PLL_DIV2.CLK_OUT	ccm_clk_enable_audio (CCGR101)
	sai7_ipt_clk_sai_bclk	CCM_SAI6_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai7_ipg_clk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai7_ipg_clk_s	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sai7_sai_mclk_1	SAI__MCLK_MUX.SAI7_MCLK_1_ROOT	ccm_clk_enable_audio (CCGR101)
	sai7_sai_mclk_2	SAI__MCLK_MUX.SAI7_MCLK_2_ROOT	ccm_clk_enable_audio (CCGR101)
	sai7_sai_mclk_3	SAI_PLL_DIV2.CLK_OUT	ccm_clk_enable_audio (CCGR101)
	sai7_ipt_clk_sai_bclk	CCM_SAI7_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
SDMA	sdma1.ips_hostctrl_clk	IPG_CLK_ROOT	ccm_clk_enable_sdma1 (CCGR58)
	sdma1.sdma_core_clk	IPG_CLK_ROOT	ccm_clk_enable_sdma1 (CCGR58)
	sdma1.sdma_ap_ahb_clk	AHB_CLK_ROOT	ccm_clk_enable_sdma1 (CCGR58)
SDMA (AUDIOMIX)	spba2_ipg_clk_s	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	sdma2_sdma_core_clk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
SEMA4	sema1.clk	IPG_CLK_ROOT	ccm_clk_enable_sema1 (CCGR61)
	sema2.clk	IPG_CLK_ROOT	ccm_clk_enable_sema2 (CCGR62)
SJC	sjc.tck		
SNVS	snvs_hp_wrapper.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_snvs (CCGR71)
	snvs_hp_wrapper.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_snvs (CCGR71)
SPBA	spba1.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_ipmux3 (CCGR30)
SPBA (AUDIOMIX)	spba2_ipg_clk_s	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
	spba2_ipg_clk	AUDIO_AHB_CLK_ROOT	ccm_clk_enable_audio (CCGR101)
SRC (CCMSRCGPCMIX)	ccmsrcgpcmix.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_mrpr (CCGR100)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
SYS_CTR	sctr.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_sctr (CCGR57)
	sctr.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_sctr (CCGR57)
	sctr.scan_clk	ANAMIX_OSC_24M_CLK	ccm_clk_enable_sctr (CCGR57)
	sctr.sys_ctr_base_clk	ANAMIX_OSC_24M_CLK	ccm_clk_enable_sctr (CCGR57)
UART	uart1.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_uart1 (CCGR73)
	uart1.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_uart1 (CCGR73)
	uart1.ipg_perclk	UART1_CLK_ROOT	ccm_clk_enable_uart1 (CCGR73)
	uart2.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_uart2 (CCGR74)
	uart2.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_uart2 (CCGR74)
	uart2.ipg_perclk	UART2_CLK_ROOT	ccm_clk_enable_uart2 (CCGR74)
	uart3.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_uart3 (CCGR75)
	uart3.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_uart3 (CCGR75)
	uart3.ipg_perclk	UART3_CLK_ROOT	ccm_clk_enable_uart3 (CCGR75)
	uart4.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_uart4 (CCGR76)
	uart4.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_uart4 (CCGR76)
	uart4.ipg_perclk	UART4_CLK_ROOT	ccm_clk_enable_uart4 (CCGR76)
USB (HSIOMIX)	module_clk	HSIOMIX_BUS_CLK_ROOT	ccm_clk_enable_hsio (CCGR92)
	ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_hsio (CCGR92)
	ckil_sync	CKIL_SYNC_CLK_ROOT	ccm_clk_enable_usb (CCGR77)
USDHC	usdhc1.hclk	NAND_USDHC_BUS_CLK_ROOT	ccm_clk_enable_usdhc1 (CCGR81)
	usdhc1.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_usdhc1 (CCGR81)
	usdhc1.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_usdhc1 (CCGR81)

Table continues on the next page...

Table 5-2. System Clocks (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable
	usdhc1.ipg_clk_perclk	USDHC1_CLK_ROOT	ccm_clk_enable_usdhc1 (CCGR81)
	usdhc2.hclk	NAND_USDHC_BUS_CLK_ROOT	ccm_clk_enable_usdhc2 (CCGR82)
	usdhc2.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_usdhc2 (CCGR82)
	usdhc2.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_usdhc2 (CCGR82)
	usdhc2.ipg_clk_perclk	USDHC2_CLK_ROOT	ccm_clk_enable_usdhc2 (CCGR82)
	usdhc3.hclk	NAND_USDHC_BUS_CLK_ROOT	ccm_clk_enable_usdhc3 (CCGR94)
	usdhc3.ipg_clk	IPG_CLK_ROOT	ccm_clk_enable_usdhc3 (CCGR94)
	usdhc3.ipg_clk_s	IPG_CLK_ROOT	ccm_clk_enable_usdhc3 (CCGR94)
	usdhc3.ipg_clk_perclk	USDHC3_CLK_ROOT	ccm_clk_enable_usdhc3 (CCGR94)
VPU G1 (VPUMIX)	vpumix.g1_core_clk	VPU_G1_CLK_ROOT	ccm_clk_enable_vpu_g1 (CCGR86)
VPU G2 (VPUMIX)	vpumix.g2_core_clk	VPU_G2_CLK_ROOT	ccm_clk_enable_vpu_g2 (CCGR90)
VPU VC8000E (VPUMIX)	vpumix.main_clk	VPU_BUS_CLK_ROOT	ccm_clk_enable_vpu (CCGR99)
	vpumix.vc8000e_core_clk	VPU_VC8000E_CLK_ROOT	ccm_clk_enable_vpu_vc8ke (CCGR89)
WDOG	wdog1.ipg_clk	WDOG_CLK_ROOT	ccm_clk_enable_wdog1 (CCGR83)
	wdog1.ipg_clk_s	WDOG_CLK_ROOT	ccm_clk_enable_wdog1 (CCGR83)
	wdog2.ipg_clk	WDOG_CLK_ROOT	ccm_clk_enable_wdog2 (CCGR84)
	wdog2.ipg_clk_s	WDOG_CLK_ROOT	ccm_clk_enable_wdog2 (CCGR84)
	wdog3.ipg_clk	WDOG_CLK_ROOT	ccm_clk_enable_wdog3 (CCGR85)
	wdog3.ipg_clk_s	WDOG_CLK_ROOT	ccm_clk_enable_wdog3 (CCGR85)

5.1.5 Functional Description

CCM receives clocks from PLLs and oscillators. Then the clock root generation logic inside CCM will generate various clock roots required for core, bus, and peripheral blocks. These clock roots will be delivered to each module through LPCG, which contains the clock gating logic for each clock.

The following sections further describe the functional details of CCM.

5.1.5.1 Input Clocks

The table below describes the input clock sources that supply the 8: 1 selector muxes to the clock slices in the clock root generator.

Control Register (CCM_PLL_CTRLn)	Input Clock	Frequency (MHz)	Description
12	ARM_PLL_CLK	1000	Arm PLL
13	GPU_PLL_CLK	1000	GPU PLL clock output
14	VPU_PLL_CLK	800	VPU PLL clock output
15	DRAM_PLL1_CLK	1067	DDR PLL
16	SYSTEM_PLL1_CLK	800	System PLL1 output clock
17	SYSTEM_PLL1_DIV2	400	System PLL1 divided 2 clock output
18	SYSTEM_PLL1_DIV3	266	System PLL1 divided 3 clock output
19	SYSTEM_PLL1_DIV4	200	System PLL1 divided 4 clock output
20	SYSTEM_PLL1_DIV5	160	System PLL1 divided 5 clock output
21	SYSTEM_PLL1_DIV6	133	System PLL1 divided 6 clock output
22	SYSTEM_PLL1_DIV8	100	System PLL1 divided 8 clock output
23	SYSTEM_PLL1_DIV10	80	System PLL1 divided 10 clock output
24	SYSTEM_PLL1_DIV20	40	System PLL1 divided 20 clock output
25	SYSTEM_PLL2_CLK	1000	System PLL2 output clock
26	SYSTEM_PLL2_DIV2	500	System PLL2 divided 2 clock output
27	SYSTEM_PLL2_DIV3	333	System PLL2 divided 3 clock output
28	SYSTEM_PLL2_DIV4	250	System PLL2 divided 4 clock output
29	SYSTEM_PLL2_DIV5	200	System PLL2 divided 5 clock output
30	SYSTEM_PLL2_DIV6	166	System PLL2 divided 6 clock output
31	SYSTEM_PLL2_DIV8	125	System PLL2 divided 8 clock output
32	SYSTEM_PLL2_DIV10	100	System PLL2 divided 10 clock output
33	SYSTEM_PLL2_DIV20	50	System PLL2 divided 20 clock output
34	SYSTEM_PLL3_CLK	1000	System PLL3 output clock
35	AUDIO_PLL1_CLK	650	Audio PLL1 clock output

Table continues on the next page...

Control Register (CCM_PLL_CTRLn)	Input Clock	Frequency (MHz)	Description
36	AUDIO_PLL2_CLK	650	Audio PLL2 clock output
37	VIDEO_PLL_CLK	1190	Video PLL clock output
external source	32K_REF_CLK	0.032	32K oscillator clock output
external source	24M_REF_CLK	24	24M oscillator clock output
external source	EXT_CLK_1	133	Clock input from external IO
external source	EXT_CLK_2	133	Clock input from external IO
external source	EXT_CLK_3	133	Clock input from external IO
external source	EXT_CLK_4	133	Clock input from external IO

5.1.5.2 CKIL Synchronizer

CCM provides a synchronized version of the 32K clock called CKIL Synchronizer (CKIL_SYNC). The CKIL_SYNC clock is generated and synchronized by the IPG_CLK_ROOT when IPG_CLK_ROOT is active.

When the system enters low-power mode that requires the shutdown of IPG_CLK_ROOT, the CKIL Synchonrizer will be bypassed and fed directly from the XTALOSC 32K source.

NOTE

CKIL_SYNC needs to be configured the same as the PLL for ipg_clk

5.1.5.3 Clock Components

The details of the CCM clock components are detailed below.

5.1.5.3.1 Clock Divider

The synchronized clock dividers (PRE_PODF and POST_PODF) can be used to perform integer division on the source clock frequency. The divider guarantees a clean clock signal on its output during the change of the divide factor. Dividers perform a 1/(N+1) divide. A glitch can cause a sync-divider to go into an unrecoverable state, therefore a clean clock signal must be provided to a sync-divider.

5.1.5.3.2 Clock Switching Multiplexer

The 2-to-1 clock switching multiplexers can guarantee a clean clock signal when switching between 2 clock sources. Both clock inputs must be active when switching. If there is any glitch in the selected clock source, then it may be transferred to the output clock while the de-selected source is blocked. Glitches may cause an unpredictable state, and will recover in 2~3 cycles.

The clock switching multiplexers shutdown the current active clock source before switching. After receiving an acknowledgement that the current clock source has shutdown, the multiplexer turns on the selected clock source. As the acknowledgement of turning on the new clock source is received, the clock source switch finishes.

5.1.5.3.3 Clock Gate

A clock gate cell is used to gate clocks. When gated, the clock will stop at 0. A clock gate cell can accept glitches on its clock inputs. If the gate cell is off, it will block or absorb glitches. If the gate cell is on, it may pass glitches to its output. A gate cell needs 2~3 clock cycles to change states. The state during this period is either on or off and will recover in 2~3 cycles, but cannot be predicted which state it is in during this period.

5.1.5.3.4 8 to 1 Multiplexer

The 8-to-1 multiplexer is a combinational multiplexer that can switch anytime. The multiplexer output does not guarantee a clean clock signal. The output clock path from the multiplexer must be clock gated before changing the multiplexer selection. This will insure a clean clock source change.

5.1.5.4 Clock Slices

There are several types of clock generation slices in CCM. The slices are categorized as Core, Bus, Peripheral (IP), and DRAM.

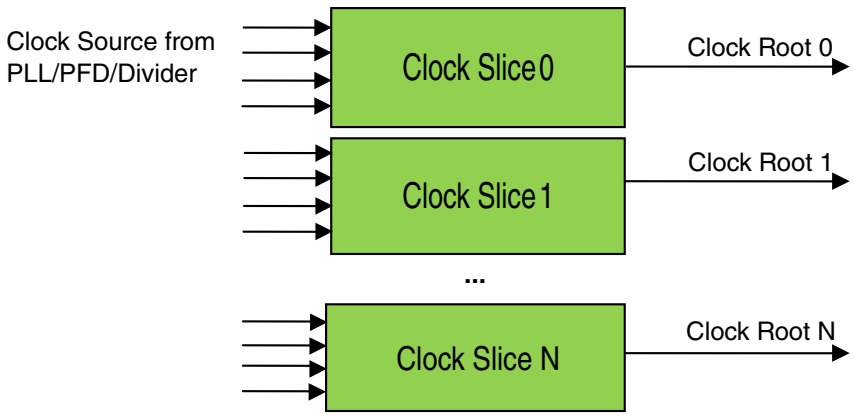


Figure 5-4. Clock Slices

The following figure illustrates the CCM clock components that a clock slice can comprise of, and the associated register controls. Not all clock slice types will comprise of all the components provided in the figure below. Please refer to the following sections to identify the components included in particular clock slice type. The slice shown below is comprised of a post divider and a clock switching multiplexer with 2 input sources. Each input source has a pre-divider, a clock gate and a clock multiplexer inside.

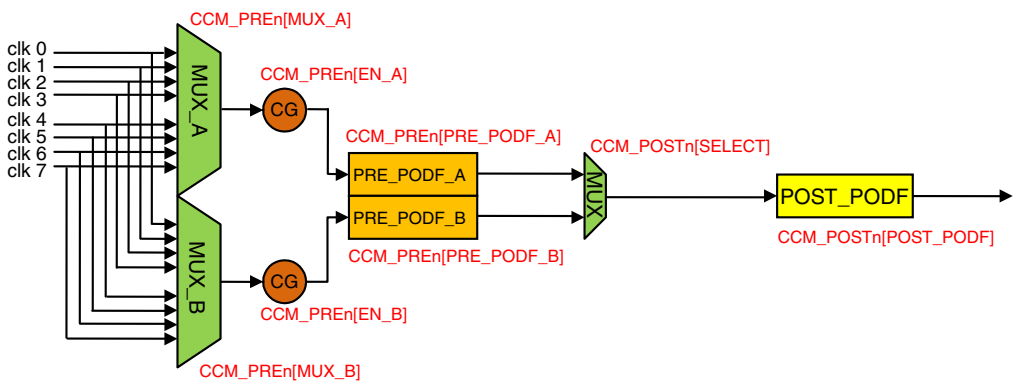


Figure 5-5. Clock Slice Components

5.1.5.4.1 Core clock slice

Core clock slices are designed for high speed, non-stop clock generation, typically for an Arm core. A core clock slice is comprised of a post divider and a 2-to-1 clock switching multiplexer with 2 input sources. Each input source has a clock gate and a 8-to-1 clock multiplexer. To run at high frequency, the post divider is only 3 bits, which is half the bit width of other slices. The two 8-to-1 multiplexers are not glitch-less, so switching them should only be done when they are clock gated to prevent propagation glitches.

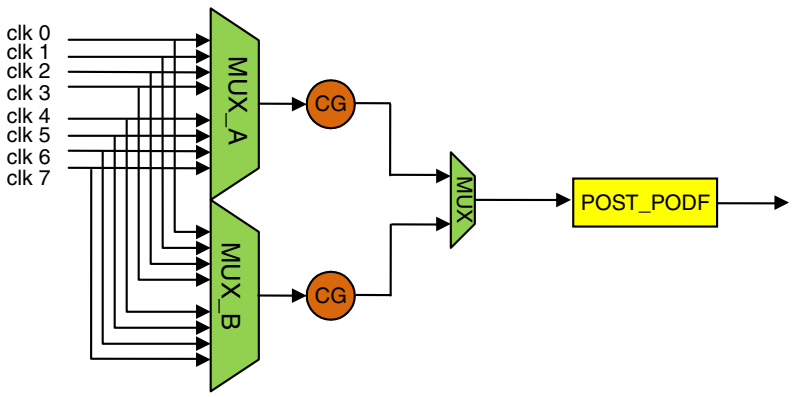


Figure 5-6. Core clock slice

5.1.5.4.2 Bus clock slice

Bus clock slices are comprised of a post divider and a clock switching multiplexer with 2 input sources. Each input source has a pre-divider, a clock gate and a clock multiplexer inside. The pre-divider is three bits and provides a maximum division factor of 8. The post-divider is six bits. The two 8-to-1 multiplexers are not glitch-less, so switching should only be done when they are clock gated to prevent propagation glitches. The eight clock selections are the same for MUX_A and MUX_B.

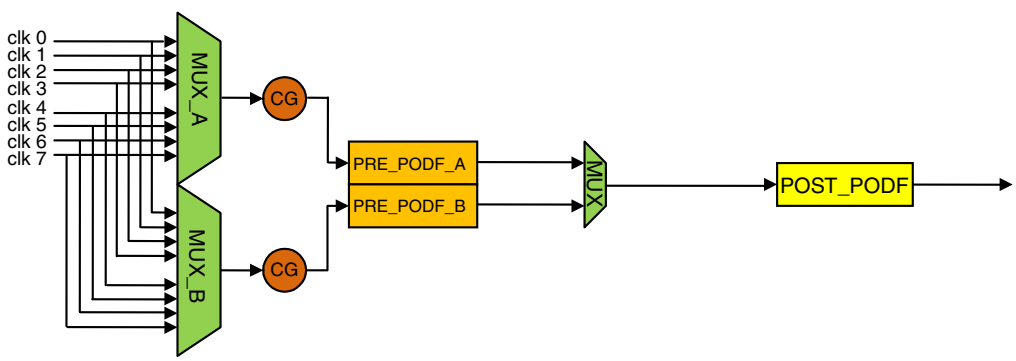


Figure 5-7. Bus clock slice

5.1.5.4.3 Peripheral clock slice

The peripheral (IP) clock slices are comprised of a clock multiplexer, clock gate, pre-divider and post-divider. The pre-divider is three bit and can divide down by a factor of 8. The post-divider is six bit . The 8-to-1 multiplexer is not glitch-less, so switching should only be done when they are clock gated to prevent propagation glitches.

Peripheral clock slices must be stopped to change the clock source.

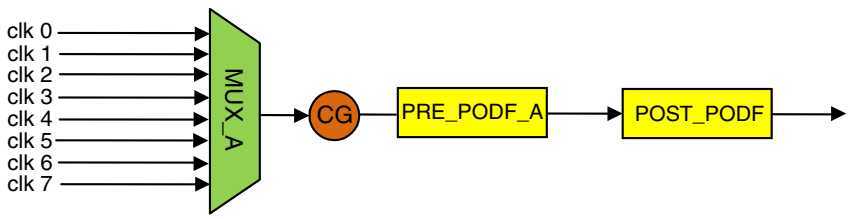


Figure 5-8. Peripheral clock slice

5.1.5.4.4 SSCG and Fractional PLLs

The Spread Spectrum Clock Generator (SSCG) and Fractional-N PLLs are detailed below.

The fractional PLL consists of a phase-frequency detector (PFD), charge pump, voltage controller oscillator (VCO), a 6-bit pre divider, 10-bit main divider, a 3-bit scaler, a delta-sigma modulator (DSM) and an automatic frequency control (AFC).

The figure below shows the Fractional PLL block diagram.

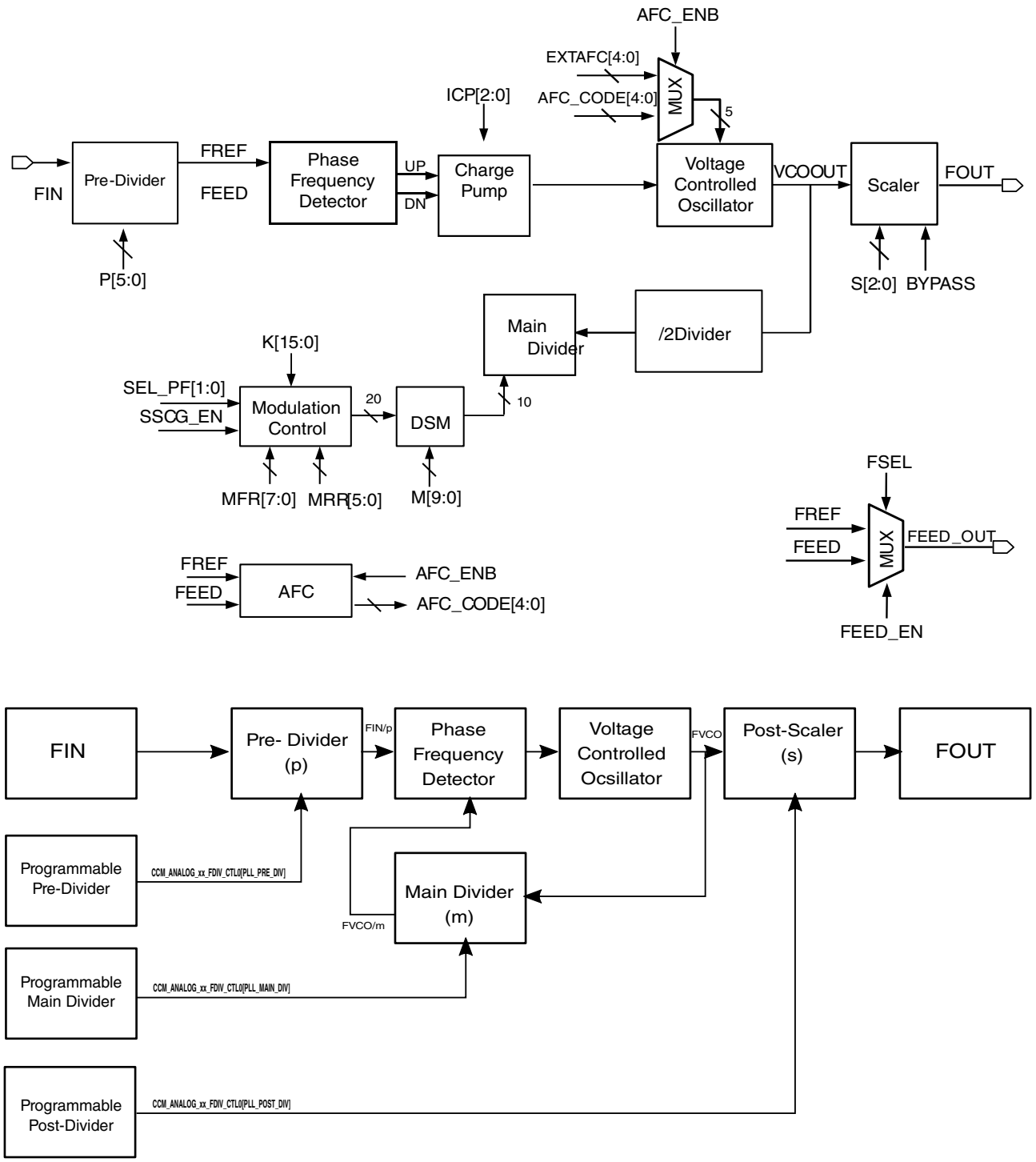


Figure 5-9. Fractional PLL Block Diagram

Formula for Fraction PLLOUT:

- FOUT=((m + k/65536) × FIN) / (p × 2^s)
- Where, 1 ≤ p ≤ 63, 64 ≤ m ≤ 1023, 0 ≤ s ≤ 6, -32768 ≤ k ≤ 32767

- p, m, and s are unsigned integers. k is a two's complement integer.
- Where, FOUT is the output frequency, FIN is the input frequency, and p,m,s and k are division values for pre-divider, main divider, scaler and DSM respectively

When SSCG_EN = 1, the spread spectrum mode is enabled. The associated formulas and values are provided below:

- Modulation frequency (MF) = F_{FIN} / p / mfr / (2^5) [Hz]
- Modulation rate (MR) = mfr × mrr / m / (2^6) × 100 [%]
- Where, 0 ≤ mfr ≤ 255, 1 ≤ mrr ≤ 63, 0 ≤ mrr × mfr ≤ 512

The ARM PLL, GPU PLL, VPU PLL, DRAM PLL, Audio PLL1/2, and Video PLL1 are fractional PLLs. The frequency on these can be tuned to be very accurate to meet audio and video interface requirements.

5.1.5.5 Clock gate control

CCM can perform automatic clock shutdown of on-chip peripherals according to system low power mode. Each logic domain can respectively declare its dependency level on each clock. If a clock is detected that is not dependant on any domain, it will be shutdown to save power.

Clock generation inside the CCM creates a clock root for on-chip peripherals. Before the clock root goes into peripherals via low power clock gating cells. By controlling these LPCGs, CCM can manage on-chip peripheral clocks.

Clock gate controls use active clock gating in LPCG, which means an active clock root is required. The clock generation module only performs multiplexing, gating and dividing on clock sources. Therefore, the clock root from the generation module will stop when the corresponding clock source stops.

The ENABLE bit must be set for the clock root that LPCG is actively gating.

5.1.5.6 Clock source control

The CCM can perform automatic PLL shutdown of CCM_ANALOG according to system low power mode. Each logic domain can respectively declare its dependency level on each clock. If a PLL is not dependent on any domain, it will be shutdown to save power.

PLLs in CCM_ANALOG can also be controlled by the CCM via setting the override bits in the respective PLL's control register.

For PLLs, there are controls on every PLL. PLLs are the source for the PFDs and dividers. For any clock that is enabled, its source must be left on, otherwise, the behavior is undefined. For a shutdown clock source, if it is set as dependent by writing to the clock source control registers (CCM_PLL_CTRLn[SETTING0/1/2/3]), the controlling logic will turn on the source immediately, while the setting goes into a shadow register. After the clock source is ready, the setting will be accepted by the source control logic, and copied from the shadow register to the setting register. Handshake with the PLL happens if the change of the setting causes a PLL to start up or shutdown. The time cost is determined by the PLL that is under control. Software can poll the setting field for a new setting value.

5.1.5.7 Access control

CCM can implement its own access control based on domain to provide more precise control on shared resources. Access controls are implemented on clock root generation, clock gate control, and clock source control. Access control logic does not impact read access, but blocks unauthentic write access.

Access controls on clock root generation are independent between every clock root. A sticky authentic fail flag is set when a domain writes to a register and authentication fails. The access control logic contains a whitelist and a semaphore. By default, each clock root's access control logic is disabled after power-on reset. Software can enable access control anytime after reset.

NOTE

Once access logic is enabled, it cannot be disabled until the next power-on reset.

Table 5-3. Whitelist

Enabled	Write access will be authenticated before being performed.
Disabled	every access on protected item will be performed.

NOTE

Only domains that are on the whitelist can perform write access to this clock root when access control is enabled.

Table 5-4. Semaphore

Enabled	A domain must obtain the semaphore’s ownership before its write access can be authenticated. Only a domain on the whitelist can obtain the ownership and the ownership will last until the domain explicitly releases the ownership. Semaphore obtain will fail if it is already fetched by some other domain.
Disabled	Authentic check will check only on whitelist.

NOTE

Semaphore is intended to help software keep the clock root from unexpectedly changing.

Access control of clock gate and clock source control is performed in a simple operation. Every domain can only write on the bits for it's own setting. Any write to irrespective domain will be ignored.

5.1.5.8 System level considerations

Clock shutdown strategy

Any clock shutdown should first shutdown the LPCG, then the PLL. If the LPCG is configured not to shutdown, the clock root for the LPCG should not stop either. Violating this rule leads the system into an unpredictable state.

Core clock root frequency

If the core clock is set lower than one third of the IPG clock, SRC needs to generate a longer reset signal to match the requirement from the Arm core. This typically happens when the Arm core runs at some divided value of the XTAL 24M while IPG clock is supplied by the PLL.

DRAM clock

The DRAM PHYM clock needs a clock frequency faster than 400MHz.

USB CLOCK

The USB clock may not be a reliable clock source in some applications. This clock may stop when USB cable is disconnected.

5.1.6 Programming Guide

5.1.6.1 Set, Clear, and Toggle register features

Every register of the CCM has set, clear, and toggle features.

The set feature for a given register is located at:
Base Address of the register + 0x04

The clear feature for a given register is located at:
Base Address of the regsister + 0x08

The toggle feature is located at:
Base Address of the register + 0x0c

Read from all 4 locations to get the current register value. Writing to the base register bits sets the register to the write value. Writing 1 to the set register bits sets them to 1, while writing 0 has no effect. Writing 1 to the clear register bits clears them to 0, while writing 0 has no effect. Writing 1 to the toggle register bits sets them to invert the value, while writing 0 has no effect.

5.1.6.2 PLL Interface

CCM can control PLLs inside CCM_ANALOG when entering or leaving low-power mode. Software must set the PLL override inside CCM_ANALOG before entering low-power mode after power-on reset (POR).

There are four levels of low-power modes in a logic domain:
- Not needed
- Needed in RUN
- Needed in RUN and WAIT
- Needed in RUN, WAIT, and STOP

CCM only takes action while domain status are switching between STOP (DEEP SLEEP mode is considered the same as STOP). There are 4 domains that can be assigned. Any CPU platform can be assigned to any domain by RDC. If a domain is empty, the domain is considered as STOP.

Each domain can declare its dependency to CCM. The use of any clock, without declaring it in its own domain, is not permitted. A domain declares its dependency on a clock by writing the dependency level. Settings against behavior in low-power mode are as follows:

Table 5-5. Domain Dependency

Domain Level	Run	Wait	Stop / Deep sleep
0			
1	Required		
2	Required	Required	
3	Required	Required	Required

Table 5-6. CCGR Program Interface

CCGR		Domain3	Domain2	Domain1	Domain0

Table continues on the next page...

Table 5-6. CCGR Program Interface (continued)

	31-16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				Level1	Level0			Level1	Level0			Level1	Level0			Level1	Level0

Each domain can change only bits assigned to control access. Any irrelevant write to it will be ignored. For example, Domain 0 can only write to bits [1:0]. Any bits written to, other than bits [1:0], will be ignored. Other domains can read all of the other domain settings. The default value for domain 0 is 2, and will enter STOP mode after shutting down. When the default value of the other domain setting is 0, it will not be required.

When setting clock source, the settings will not take effect immediately. The setting will enter the shadow register first. If a PLL shutdown or new setting enters the shadow register to declare dependency on the PLL, the PLL will turn on immediately. When the PLL is ready, the setting in shadow register will be updated to the new setting. During this period, the pending bit will be set and cleared. Then CCM will send the PLL control signal as a shadow register and inform GPC the PLL status according to the setting register. In other cases, the setting will be updated from the shadow register immediately. Clock sources have dependency on each other.

NOTE

Do not shutdown the parent clock when the required child clock is active. Attempting to do so will lead to unpredicable and unrecoverable behavior. It is recommended to shutdown the parent clock and child clock together.

5.1.6.3 CCGR Interface

Before a clock root goes to on–chip peripherals, the clock root is distributed through low power clock gates (LPCG). These LPCG are implemented to automatically perform clock shutdown when a domain enters and leaves a low-power state.

There are four levels of low-power modes in a logic domain:

- Not needed
- Needed in RUN
- Needed in RUN and WAIT
- Needed in RUN, WAIT, and STOP

CCM only takes action while domain status are switching between STOP (DEEP SLEEP mode is considered the same as STOP). There are 4 domains that can be assigned. Any CPU platform can be assigned to any domain by RDC. If a domain is empty, the domain is considered as STOP.

Each domain can declare its dependency to CCM. The use of any clock, without declaring it in its own domain, is not permitted. A domain declares its dependency on a clock by writing the dependency level. Settings against behavior in low-power mode are as follows:

Table 5-7. Domain Dependency

Domain Level	RUN	WAIT	STOP/ DEEP SLEEP
0			
1	Required		
2	Required	Required	
3	Required	Required	Required

Table 5-8. CCGR Program Interface

CCGR		Domain3				Domain2				Domain1				Domain0			
	31-16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				Level1	Level0			Level1	Level0			Level1	Level0			Level1	Level0

Each domain can change only bits assigned to control access. Any irrelevant write to it will be ignored. For example, Domain 0 can only write to bits [1:0]. Any bits written to, other than bits [1:0], will be ignored. Other domains can read all of the other domain settings. The default value for domain 0 is 2, and will enter STOP mode after shutting down. When the default value of the other domain setting is 0, it will not be required.

The table below lists the CCM Clock Gating Register (CCGR) and associated offset for each LPCG enable.

NOTE

Not all CCGRs are mapped.

NOTE

Sec_debug clock gating (CCGR60) must be active in low power mode. DO NOT gate this clock in low power mode to guarantee the low power mode functions such as stop WDOG counting.

Table 5-9. CCGR Mapping Table

Gating Register	LPCG Enable	Offset
CCM_CCGR0	DVFS	0x4000
CCM_CCGR1	Anamix	0x4010
CCM_CCGR2	CPU	0x4020

Table continues on the next page...

Table 5-9. CCGR Mapping Table (continued)

Gating Register	LPCG Enable	Offset
CCM_CCGR3	CSU	0x4030
CCM_CCGR4	Debug	0x4040
CCM_CCGR5	DRAM1	0x4050
CCM_CCGR6	Reserved	0x4060
CCM_CCGR7	ECSPI1	0x4070
CCM_CCGR8	ECSPI2	0x4080
CCM_CCGR9	ECSPI3	0x4090
CCM_CCGR10	ENET1	0x40A0
CCM_CCGR11	GPIO1	0x40B0
CCM_CCGR12	GPIO2	0x40C0
CCM_CCGR13	GPIO3	0x40D0
CCM_CCGR14	GPIO4	0x40E0
CCM_CCGR15	GPIO5	0x40F0
CCM_CCGR16	GPT1	0x4100
CCM_CCGR17	GPT2	0x4110
CCM_CCGR18	GPT3	0x4120
CCM_CCGR19	GPT4	0x4130
CCM_CCGR20	GPT5	0x4140
CCM_CCGR21	GPT6	0x4150
CCM_CCGR22	HS	0x4160
CCM_CCGR23	I2C1	0x4170
CCM_CCGR24	I2C2	0x4180
CCM_CCGR25	I2C3	0x4190
CCM_CCGR26	I2C4	0x41A0
CCM_CCGR27	IOMUX	0x41B0
CCM_CCGR28	IPMUX1	0x41C0
CCM_CCGR29	IPMUX2	0x41D0
CCM_CCGR30	IPMUX3	0x41E0
CCM_CCGR31	Reserved	0x41F0
CCM_CCGR32	Reserved	0x4200
CCM_CCGR33	MU	0x4210
CCM_CCGR34	OCOTP	0x4220
CCM_CCGR35	OCRAM	0x4230
CCM_CCGR36	OCRAM_s	0x4240
CCM_CCGR37	PCIE	0x4250
CCM_CCGR38	PERFMON1	0x4260
CCM_CCGR39	PERFMON2	0x4270
CCM_CCGR40	PWM1	0x4280
CCM_CCGR41	PWM2	0x4290

Table continues on the next page...

Table 5-9. CCGR Mapping Table (continued)

Gating Register	LPCG Enable	Offset
CCM_CCGR42	PWM3	0x42A0
CCM_CCGR43	PWM4	0x42B0
CCM_CCGR44	QoS	0x42C0
CCM_CCGR45	Reserved	0x42D0
CCM_CCGR46	QoS_ENET	0x42E0
CCM_CCGR47	FLEXSPI	0x42F0
CCM_CCGR48	RAWNAND	0x4300
CCM_CCGR49	RDC	0x4310
CCM_CCGR50	ROM	0x4320
CCM_CCGR51	I2C5	0x4330
CCM_CCGR52	I2C6	0x4340
CCM_CCGR53	CAN1	0x4350
CCM_CCGR54	CAN2	0x4360
CCM_CCGR55	Reserved	0x4370
CCM_CCGR56	Reserved	0x4380
CCM_CCGR57	SCTR	0x4390
CCM_CCGR58	SDMA1	0x43A0
CCM_CCGR59	ENET_QoS	0x43B0
CCM_CCGR60	SEC_DEBUG	0x43C0
CCM_CCGR61	SEMA1	0x43D0
CCM_CCGR62	SEMA2	0x43E0
CCM_CCGR63	IRQ_STEER	0x43F0
CCM_CCGR64	SIM_ENET	0x4400
CCM_CCGR65	SIM_m	0x4410
CCM_CCGR66	SIM_main	0x4420
CCM_CCGR67	SIM_s	0x4430
CCM_CCGR68	SIM_wakeup	0x4440
CCM_CCGR69	GPU2D	0x4450
CCM_CCGR70	GPU3D	0x4460
CCM_CCGR71	SNVS	0x4470
CCM_CCGR72	TRACE	0x4480
CCM_CCGR73	UART1	0x4490
CCM_CCGR74	UART2	0x44A0
CCM_CCGR75	UART3	0x44B0
CCM_CCGR76	UART4	0x44C0
CCM_CCGR77	USB	0x44D0
CCM_CCGR78	Reserved	0x44E0
CCM_CCGR79	USB_PHY	0x44F0
CCM_CCGR80	Reserved	0x4500

Table continues on the next page...

Table 5-9. CCGR Mapping Table (continued)

Gating Register	LPCG Enable	Offset
CCM_CCGR81	USDHC1	0x4510
CCM_CCGR82	USDHC2	0x4520
CCM_CCGR83	WDOG1	0x4530
CCM_CCGR84	WDOG2	0x4540
CCM_CCGR85	WDOG3	0x4550
CCM_CCGR86	VPU_G1	0x4560
CCM_CCGR87	GPU	0x4570
CCM_CCGR88	NOC_WRAPPER	0x4580
CCM_CCGR89	VPU_VC8KE	0x4590
CCM_CCGR90	VPU_G2	0x45A0
CCM_CCGR91	NPU	0x45B0
CCM_CCGR92	HSIO	0x45C0
CCM_CCGR93	MEDIA	0x45D0
CCM_CCGR94	USDHC3	0x45E0
CCM_CCGR95	HDMI	0x45F0
CCM_CCGR96	XTAL	0x4600
CCM_CCGR97	PLL	0x4610
CCM_CCGR98	TSENSOR	0x4620
CCM_CCGR99	VPU	0x4630
CCM_CCGR100	MRPR	0x4640
CCM_CCGR101	AUDIO	0x4650
CCM_CCGR102	Reserved	0x4660

5.1.6.4 Target Interface

The Target Interface is optimized to simplify software operation. Using this interface, all clock roots are in the same program model with the same register bit field mapping. The software does not handle the details of the clock slice and clock slice types. Software writes the desired settings to the register, and the internal hardware logic generates a required sequence to achieve the desired settings.

The Target Interface requires the software to provide whether a clock is active, the clock source number to be selected, pre-divide value, and post-divide value. If a clock slice does not support a setting, that setting is simply ignored, and will not effect the supported fields.

Freq = (clock source freq)/(pre_div+1)/(post_div+1)

The internal logic sequence of the Target Interface guarantees a clean clock on output without frequency overshoot. A requirement of the Target Interface's software is that the target clock source is active.

The Target Interface sequence begins by opening all clocks, applying highest divider value, switching to the new clock source, then decreasing divider value to the target frequency. If Shutdown is requested, it will be performed last.

The clock output is always active when using the Target Interface. For intermediate frequency requests, the Target Interface choses the lowest frequency source to avoid frequency overshoot on the Peripheral clock slices. For Core and Bus clock slices, the clock switching multiplexer is used to guarantee smooth clock switching.

A write operation on a target interface completes once the output clock is running at the desired setting. Software polling is not necessary to determine clock stability.

STEP	STATE	OPERATION
0	SMART_IDLE	Idle state, no write operation is pending
1	SMART_WAIT_READY	State occurs when a write access is received, wait for every field to be ready
2	SMART_APPLY_GATE1	Open all branches, all gates inside clock slices
3	SMART_WAIT_GATE1	Wait for gate applied
4	SMART_APPLY_PODF1	apply post divider and post divider for if new value generate slower clock
5	SMART_WAIT_PODF1	Wait for divider accept new value
6	SMART_APPLY_GATE2	shutdown spare branch if exist, else shutdown working branch
7	SMART_WAIT_GATE2	Wait for shutdown operation complete
8	SMART_APPLY_MUX	Change multiplexer to new source on spare branch if exist, else switch working one
9	SMART_APPLY_GATE3	open gates on all branches
10	SMART_WAIT_GATE3	Wait for gates opened
11	SMART_APPLY_SWITCH	Switch clock switching multiplexer if there is one
12	SMART_WAIT_SWITCH	Wait for clock switching multiplexer switch
13	SMART_APPLY_PODF2	apply post divider and post divider for if new value generate faster clock
14	SMART_WAIT_PODF2	Wait for divider accept new value
15	SMART_APPLY_GATE4	apply clock gate setting, shutdown spare one if there is
16	SMART_WAIT_GATE4	Wait for gate applied
17	SMART_APPLY_AUTO	apply auto and auto divider
18	SMART_DONE	Finish, wait for bus operation complete

5.1.6.5 Normal Interface

Normal interface is more controllable than target interface. And also provides protections against dangerous operations.

Normal interface provides safe sequences to handle each clock component, divider, gate, and multiplexer. But it is software that needs to guarantee the order and relationship between updating components.

Writing to this interface will complete immediately, and internal logic will continue to apply written values to clock components. A busy flag will be asserted during application.

Field access rules:

1. Only one field can be modified at a time
2. No field can be modified when any field is pending
3. Do not violate change condition in following table

FIELD	CHANGE CONDITION	FINISH CONDITION
Auto		immediate
Auto-divider		immediate
Bypass	Gatea active and gateb active	Bypass switch complete
Post-divider		New divider value applied
Gate B	Bypass disable	New gate value active
Pre-divider B	Bypass disable and gateb not gated	New divider value applied
MUX B	Bypass disable and gateb gating	immediate
Gate A	Bypass	New gate value active
Pre-divider A	Bypass and gatea not gated	New divider value applied
MUX A	Bypass and gatea gating	immediate

- Error will be reported if access rules violated.
- Unsafe or ambiguous access will be ignored.

If a write access iss blocked by normal interface, the write operation will be ignored. And a sticky bit “violate” will be set. The bit will last until software clears it explicitly. The violate bit is 4 bits inside CCM, each for a logic domain. Each domain can read and clear the bit for itself, the bits for other domain is neither visible nor clearable.

5.1.7 CCM Memory Map/Register Definition

The Memory Map below represents the full array for CCM.

NOTE

Not all mapped Clock Slices and CCGRs are tied to functional components.

Please see the following for the functional mapping tables and information:

- CCM_PLL_CTRL - Input Clocks
- CCM_TARGET_ROOT - Clock Root Selects
- CCM_CCGR - CCGR Interface

CCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_0000	General Purpose Register (CCM_GPR0)	32	R/W	0000_0000h	5.1.7.1/444
3038_0004	General Purpose Register (CCM_GPR0_SET)	32	R/W	0000_0000h	5.1.7.1/444
3038_0008	General Purpose Register (CCM_GPR0_CLR)	32	R/W	0000_0000h	5.1.7.1/444
3038_000C	General Purpose Register (CCM_GPR0_TOG)	32	R/W	0000_0000h	5.1.7.1/444
3038_0800	CCM PLL Control Register (CCM_PLL_CTRL0)	32	R/W	0000_0002h	5.1.7.2/445
3038_0804	CCM PLL Control Register (CCM_PLL_CTRL0_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0808	CCM PLL Control Register (CCM_PLL_CTRL0_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_080C	CCM PLL Control Register (CCM_PLL_CTRL0_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0810	CCM PLL Control Register (CCM_PLL_CTRL1)	32	R/W	0000_0002h	5.1.7.2/445
3038_0814	CCM PLL Control Register (CCM_PLL_CTRL1_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0818	CCM PLL Control Register (CCM_PLL_CTRL1_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_081C	CCM PLL Control Register (CCM_PLL_CTRL1_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0820	CCM PLL Control Register (CCM_PLL_CTRL2)	32	R/W	0000_0002h	5.1.7.2/445
3038_0824	CCM PLL Control Register (CCM_PLL_CTRL2_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0828	CCM PLL Control Register (CCM_PLL_CTRL2_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_082C	CCM PLL Control Register (CCM_PLL_CTRL2_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0830	CCM PLL Control Register (CCM_PLL_CTRL3)	32	R/W	0000_0002h	5.1.7.2/445
3038_0834	CCM PLL Control Register (CCM_PLL_CTRL3_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0838	CCM PLL Control Register (CCM_PLL_CTRL3_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_083C	CCM PLL Control Register (CCM_PLL_CTRL3_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0840	CCM PLL Control Register (CCM_PLL_CTRL4)	32	R/W	0000_0002h	5.1.7.2/445
3038_0844	CCM PLL Control Register (CCM_PLL_CTRL4_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0848	CCM PLL Control Register (CCM_PLL_CTRL4_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_084C	CCM PLL Control Register (CCM_PLL_CTRL4_TOG)	32	R/W	0000_0002h	5.1.7.5/451

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_0850	CCM PLL Control Register (CCM_PLL_CTRL5)	32	R/W	0000_0002h	5.1.7.2/445
3038_0854	CCM PLL Control Register (CCM_PLL_CTRL5_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0858	CCM PLL Control Register (CCM_PLL_CTRL5_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_085C	CCM PLL Control Register (CCM_PLL_CTRL5_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0860	CCM PLL Control Register (CCM_PLL_CTRL6)	32	R/W	0000_0002h	5.1.7.2/445
3038_0864	CCM PLL Control Register (CCM_PLL_CTRL6_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0868	CCM PLL Control Register (CCM_PLL_CTRL6_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_086C	CCM PLL Control Register (CCM_PLL_CTRL6_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0870	CCM PLL Control Register (CCM_PLL_CTRL7)	32	R/W	0000_0002h	5.1.7.2/445
3038_0874	CCM PLL Control Register (CCM_PLL_CTRL7_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0878	CCM PLL Control Register (CCM_PLL_CTRL7_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_087C	CCM PLL Control Register (CCM_PLL_CTRL7_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0880	CCM PLL Control Register (CCM_PLL_CTRL8)	32	R/W	0000_0002h	5.1.7.2/445
3038_0884	CCM PLL Control Register (CCM_PLL_CTRL8_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0888	CCM PLL Control Register (CCM_PLL_CTRL8_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_088C	CCM PLL Control Register (CCM_PLL_CTRL8_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0890	CCM PLL Control Register (CCM_PLL_CTRL9)	32	R/W	0000_0002h	5.1.7.2/445
3038_0894	CCM PLL Control Register (CCM_PLL_CTRL9_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0898	CCM PLL Control Register (CCM_PLL_CTRL9_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_089C	CCM PLL Control Register (CCM_PLL_CTRL9_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_08A0	CCM PLL Control Register (CCM_PLL_CTRL10)	32	R/W	0000_0002h	5.1.7.2/445
3038_08A4	CCM PLL Control Register (CCM_PLL_CTRL10_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_08A8	CCM PLL Control Register (CCM_PLL_CTRL10_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_08AC	CCM PLL Control Register (CCM_PLL_CTRL10_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_08B0	CCM PLL Control Register (CCM_PLL_CTRL11)	32	R/W	0000_0002h	5.1.7.2/445
3038_08B4	CCM PLL Control Register (CCM_PLL_CTRL11_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_08B8	CCM PLL Control Register (CCM_PLL_CTRL11_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_08BC	CCM PLL Control Register (CCM_PLL_CTRL11_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_08C0	CCM PLL Control Register (CCM_PLL_CTRL12)	32	R/W	0000_0002h	5.1.7.2/445
3038_08C4	CCM PLL Control Register (CCM_PLL_CTRL12_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_08C8	CCM PLL Control Register (CCM_PLL_CTRL12_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_08CC	CCM PLL Control Register (CCM_PLL_CTRL12_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_08D0	CCM PLL Control Register (CCM_PLL_CTRL13)	32	R/W	0000_0002h	5.1.7.2/445
3038_08D4	CCM PLL Control Register (CCM_PLL_CTRL13_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_08D8	CCM PLL Control Register (CCM_PLL_CTRL13_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_08DC	CCM PLL Control Register (CCM_PLL_CTRL13_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_08E0	CCM PLL Control Register (CCM_PLL_CTRL14)	32	R/W	0000_0002h	5.1.7.2/445
3038_08E4	CCM PLL Control Register (CCM_PLL_CTRL14_SET)	32	R/W	0000_0002h	5.1.7.3/447

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_08E8	CCM PLL Control Register (CCM_PLL_CTRL14_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_08EC	CCM PLL Control Register (CCM_PLL_CTRL14_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_08F0	CCM PLL Control Register (CCM_PLL_CTRL15)	32	R/W	0000_0002h	5.1.7.2/445
3038_08F4	CCM PLL Control Register (CCM_PLL_CTRL15_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_08F8	CCM PLL Control Register (CCM_PLL_CTRL15_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_08FC	CCM PLL Control Register (CCM_PLL_CTRL15_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0900	CCM PLL Control Register (CCM_PLL_CTRL16)	32	R/W	0000_0002h	5.1.7.2/445
3038_0904	CCM PLL Control Register (CCM_PLL_CTRL16_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0908	CCM PLL Control Register (CCM_PLL_CTRL16_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_090C	CCM PLL Control Register (CCM_PLL_CTRL16_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0910	CCM PLL Control Register (CCM_PLL_CTRL17)	32	R/W	0000_0002h	5.1.7.2/445
3038_0914	CCM PLL Control Register (CCM_PLL_CTRL17_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0918	CCM PLL Control Register (CCM_PLL_CTRL17_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_091C	CCM PLL Control Register (CCM_PLL_CTRL17_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0920	CCM PLL Control Register (CCM_PLL_CTRL18)	32	R/W	0000_0002h	5.1.7.2/445
3038_0924	CCM PLL Control Register (CCM_PLL_CTRL18_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0928	CCM PLL Control Register (CCM_PLL_CTRL18_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_092C	CCM PLL Control Register (CCM_PLL_CTRL18_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0930	CCM PLL Control Register (CCM_PLL_CTRL19)	32	R/W	0000_0002h	5.1.7.2/445
3038_0934	CCM PLL Control Register (CCM_PLL_CTRL19_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0938	CCM PLL Control Register (CCM_PLL_CTRL19_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_093C	CCM PLL Control Register (CCM_PLL_CTRL19_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0940	CCM PLL Control Register (CCM_PLL_CTRL20)	32	R/W	0000_0002h	5.1.7.2/445
3038_0944	CCM PLL Control Register (CCM_PLL_CTRL20_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0948	CCM PLL Control Register (CCM_PLL_CTRL20_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_094C	CCM PLL Control Register (CCM_PLL_CTRL20_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0950	CCM PLL Control Register (CCM_PLL_CTRL21)	32	R/W	0000_0002h	5.1.7.2/445
3038_0954	CCM PLL Control Register (CCM_PLL_CTRL21_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0958	CCM PLL Control Register (CCM_PLL_CTRL21_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_095C	CCM PLL Control Register (CCM_PLL_CTRL21_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0960	CCM PLL Control Register (CCM_PLL_CTRL22)	32	R/W	0000_0002h	5.1.7.2/445
3038_0964	CCM PLL Control Register (CCM_PLL_CTRL22_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0968	CCM PLL Control Register (CCM_PLL_CTRL22_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_096C	CCM PLL Control Register (CCM_PLL_CTRL22_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0970	CCM PLL Control Register (CCM_PLL_CTRL23)	32	R/W	0000_0002h	5.1.7.2/445
3038_0974	CCM PLL Control Register (CCM_PLL_CTRL23_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0978	CCM PLL Control Register (CCM_PLL_CTRL23_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_097C	CCM PLL Control Register (CCM_PLL_CTRL23_TOG)	32	R/W	0000_0002h	5.1.7.5/451

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_0980	CCM PLL Control Register (CCM_PLL_CTRL24)	32	R/W	0000_0002h	5.1.7.2/445
3038_0984	CCM PLL Control Register (CCM_PLL_CTRL24_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0988	CCM PLL Control Register (CCM_PLL_CTRL24_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_098C	CCM PLL Control Register (CCM_PLL_CTRL24_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0990	CCM PLL Control Register (CCM_PLL_CTRL25)	32	R/W	0000_0002h	5.1.7.2/445
3038_0994	CCM PLL Control Register (CCM_PLL_CTRL25_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0998	CCM PLL Control Register (CCM_PLL_CTRL25_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_099C	CCM PLL Control Register (CCM_PLL_CTRL25_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_09A0	CCM PLL Control Register (CCM_PLL_CTRL26)	32	R/W	0000_0002h	5.1.7.2/445
3038_09A4	CCM PLL Control Register (CCM_PLL_CTRL26_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_09A8	CCM PLL Control Register (CCM_PLL_CTRL26_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_09AC	CCM PLL Control Register (CCM_PLL_CTRL26_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_09B0	CCM PLL Control Register (CCM_PLL_CTRL27)	32	R/W	0000_0002h	5.1.7.2/445
3038_09B4	CCM PLL Control Register (CCM_PLL_CTRL27_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_09B8	CCM PLL Control Register (CCM_PLL_CTRL27_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_09BC	CCM PLL Control Register (CCM_PLL_CTRL27_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_09C0	CCM PLL Control Register (CCM_PLL_CTRL28)	32	R/W	0000_0002h	5.1.7.2/445
3038_09C4	CCM PLL Control Register (CCM_PLL_CTRL28_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_09C8	CCM PLL Control Register (CCM_PLL_CTRL28_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_09CC	CCM PLL Control Register (CCM_PLL_CTRL28_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_09D0	CCM PLL Control Register (CCM_PLL_CTRL29)	32	R/W	0000_0002h	5.1.7.2/445
3038_09D4	CCM PLL Control Register (CCM_PLL_CTRL29_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_09D8	CCM PLL Control Register (CCM_PLL_CTRL29_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_09DC	CCM PLL Control Register (CCM_PLL_CTRL29_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_09E0	CCM PLL Control Register (CCM_PLL_CTRL30)	32	R/W	0000_0002h	5.1.7.2/445
3038_09E4	CCM PLL Control Register (CCM_PLL_CTRL30_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_09E8	CCM PLL Control Register (CCM_PLL_CTRL30_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_09EC	CCM PLL Control Register (CCM_PLL_CTRL30_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_09F0	CCM PLL Control Register (CCM_PLL_CTRL31)	32	R/W	0000_0002h	5.1.7.2/445
3038_09F4	CCM PLL Control Register (CCM_PLL_CTRL31_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_09F8	CCM PLL Control Register (CCM_PLL_CTRL31_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_09FC	CCM PLL Control Register (CCM_PLL_CTRL31_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0A00	CCM PLL Control Register (CCM_PLL_CTRL32)	32	R/W	0000_0002h	5.1.7.2/445
3038_0A04	CCM PLL Control Register (CCM_PLL_CTRL32_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0A08	CCM PLL Control Register (CCM_PLL_CTRL32_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_0A0C	CCM PLL Control Register (CCM_PLL_CTRL32_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0A10	CCM PLL Control Register (CCM_PLL_CTRL33)	32	R/W	0000_0002h	5.1.7.2/445
3038_0A14	CCM PLL Control Register (CCM_PLL_CTRL33_SET)	32	R/W	0000_0002h	5.1.7.3/447

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_0A18	CCM PLL Control Register (CCM_PLL_CTRL33_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_0A1C	CCM PLL Control Register (CCM_PLL_CTRL33_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0A20	CCM PLL Control Register (CCM_PLL_CTRL34)	32	R/W	0000_0002h	5.1.7.2/445
3038_0A24	CCM PLL Control Register (CCM_PLL_CTRL34_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0A28	CCM PLL Control Register (CCM_PLL_CTRL34_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_0A2C	CCM PLL Control Register (CCM_PLL_CTRL34_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0A30	CCM PLL Control Register (CCM_PLL_CTRL35)	32	R/W	0000_0002h	5.1.7.2/445
3038_0A34	CCM PLL Control Register (CCM_PLL_CTRL35_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0A38	CCM PLL Control Register (CCM_PLL_CTRL35_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_0A3C	CCM PLL Control Register (CCM_PLL_CTRL35_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0A40	CCM PLL Control Register (CCM_PLL_CTRL36)	32	R/W	0000_0002h	5.1.7.2/445
3038_0A44	CCM PLL Control Register (CCM_PLL_CTRL36_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0A48	CCM PLL Control Register (CCM_PLL_CTRL36_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_0A4C	CCM PLL Control Register (CCM_PLL_CTRL36_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0A50	CCM PLL Control Register (CCM_PLL_CTRL37)	32	R/W	0000_0002h	5.1.7.2/445
3038_0A54	CCM PLL Control Register (CCM_PLL_CTRL37_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0A58	CCM PLL Control Register (CCM_PLL_CTRL37_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_0A5C	CCM PLL Control Register (CCM_PLL_CTRL37_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_0A60	CCM PLL Control Register (CCM_PLL_CTRL38)	32	R/W	0000_0002h	5.1.7.2/445
3038_0A64	CCM PLL Control Register (CCM_PLL_CTRL38_SET)	32	R/W	0000_0002h	5.1.7.3/447
3038_0A68	CCM PLL Control Register (CCM_PLL_CTRL38_CLR)	32	R/W	0000_0002h	5.1.7.4/449
3038_0A6C	CCM PLL Control Register (CCM_PLL_CTRL38_TOG)	32	R/W	0000_0002h	5.1.7.5/451
3038_4000	CCM Clock Gating Register (CCM_CCGR0)	32	R/W	0000_0002h	5.1.7.6/452
3038_4004	CCM Clock Gating Register (CCM_CCGR0_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4008	CCM Clock Gating Register (CCM_CCGR0_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_400C	CCM Clock Gating Register (CCM_CCGR0_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4010	CCM Clock Gating Register (CCM_CCGR1)	32	R/W	0000_0002h	5.1.7.6/452
3038_4014	CCM Clock Gating Register (CCM_CCGR1_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4018	CCM Clock Gating Register (CCM_CCGR1_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_401C	CCM Clock Gating Register (CCM_CCGR1_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4020	CCM Clock Gating Register (CCM_CCGR2)	32	R/W	0000_0002h	5.1.7.6/452
3038_4024	CCM Clock Gating Register (CCM_CCGR2_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4028	CCM Clock Gating Register (CCM_CCGR2_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_402C	CCM Clock Gating Register (CCM_CCGR2_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4030	CCM Clock Gating Register (CCM_CCGR3)	32	R/W	0000_0002h	5.1.7.6/452
3038_4034	CCM Clock Gating Register (CCM_CCGR3_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4038	CCM Clock Gating Register (CCM_CCGR3_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_403C	CCM Clock Gating Register (CCM_CCGR3_TOG)	32	R/W	0000_0002h	5.1.7.9/459

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4040	CCM Clock Gating Register (CCM_CCGR4)	32	R/W	0000_0002h	5.1.7.6/452
3038_4044	CCM Clock Gating Register (CCM_CCGR4_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4048	CCM Clock Gating Register (CCM_CCGR4_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_404C	CCM Clock Gating Register (CCM_CCGR4_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4050	CCM Clock Gating Register (CCM_CCGR5)	32	R/W	0000_0002h	5.1.7.6/452
3038_4054	CCM Clock Gating Register (CCM_CCGR5_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4058	CCM Clock Gating Register (CCM_CCGR5_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_405C	CCM Clock Gating Register (CCM_CCGR5_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4060	CCM Clock Gating Register (CCM_CCGR6)	32	R/W	0000_0002h	5.1.7.6/452
3038_4064	CCM Clock Gating Register (CCM_CCGR6_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4068	CCM Clock Gating Register (CCM_CCGR6_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_406C	CCM Clock Gating Register (CCM_CCGR6_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4070	CCM Clock Gating Register (CCM_CCGR7)	32	R/W	0000_0002h	5.1.7.6/452
3038_4074	CCM Clock Gating Register (CCM_CCGR7_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4078	CCM Clock Gating Register (CCM_CCGR7_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_407C	CCM Clock Gating Register (CCM_CCGR7_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4080	CCM Clock Gating Register (CCM_CCGR8)	32	R/W	0000_0002h	5.1.7.6/452
3038_4084	CCM Clock Gating Register (CCM_CCGR8_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4088	CCM Clock Gating Register (CCM_CCGR8_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_408C	CCM Clock Gating Register (CCM_CCGR8_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4090	CCM Clock Gating Register (CCM_CCGR9)	32	R/W	0000_0002h	5.1.7.6/452
3038_4094	CCM Clock Gating Register (CCM_CCGR9_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4098	CCM Clock Gating Register (CCM_CCGR9_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_409C	CCM Clock Gating Register (CCM_CCGR9_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_40A0	CCM Clock Gating Register (CCM_CCGR10)	32	R/W	0000_0002h	5.1.7.6/452
3038_40A4	CCM Clock Gating Register (CCM_CCGR10_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_40A8	CCM Clock Gating Register (CCM_CCGR10_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_40AC	CCM Clock Gating Register (CCM_CCGR10_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_40B0	CCM Clock Gating Register (CCM_CCGR11)	32	R/W	0000_0002h	5.1.7.6/452
3038_40B4	CCM Clock Gating Register (CCM_CCGR11_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_40B8	CCM Clock Gating Register (CCM_CCGR11_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_40BC	CCM Clock Gating Register (CCM_CCGR11_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_40C0	CCM Clock Gating Register (CCM_CCGR12)	32	R/W	0000_0002h	5.1.7.6/452
3038_40C4	CCM Clock Gating Register (CCM_CCGR12_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_40C8	CCM Clock Gating Register (CCM_CCGR12_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_40CC	CCM Clock Gating Register (CCM_CCGR12_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_40D0	CCM Clock Gating Register (CCM_CCGR13)	32	R/W	0000_0002h	5.1.7.6/452
3038_40D4	CCM Clock Gating Register (CCM_CCGR13_SET)	32	R/W	0000_0002h	5.1.7.7/455

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_40D8	CCM Clock Gating Register (CCM_CCGR13_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_40DC	CCM Clock Gating Register (CCM_CCGR13_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_40E0	CCM Clock Gating Register (CCM_CCGR14)	32	R/W	0000_0002h	5.1.7.6/452
3038_40E4	CCM Clock Gating Register (CCM_CCGR14_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_40E8	CCM Clock Gating Register (CCM_CCGR14_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_40EC	CCM Clock Gating Register (CCM_CCGR14_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_40F0	CCM Clock Gating Register (CCM_CCGR15)	32	R/W	0000_0002h	5.1.7.6/452
3038_40F4	CCM Clock Gating Register (CCM_CCGR15_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_40F8	CCM Clock Gating Register (CCM_CCGR15_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_40FC	CCM Clock Gating Register (CCM_CCGR15_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4100	CCM Clock Gating Register (CCM_CCGR16)	32	R/W	0000_0002h	5.1.7.6/452
3038_4104	CCM Clock Gating Register (CCM_CCGR16_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4108	CCM Clock Gating Register (CCM_CCGR16_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_410C	CCM Clock Gating Register (CCM_CCGR16_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4110	CCM Clock Gating Register (CCM_CCGR17)	32	R/W	0000_0002h	5.1.7.6/452
3038_4114	CCM Clock Gating Register (CCM_CCGR17_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4118	CCM Clock Gating Register (CCM_CCGR17_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_411C	CCM Clock Gating Register (CCM_CCGR17_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4120	CCM Clock Gating Register (CCM_CCGR18)	32	R/W	0000_0002h	5.1.7.6/452
3038_4124	CCM Clock Gating Register (CCM_CCGR18_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4128	CCM Clock Gating Register (CCM_CCGR18_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_412C	CCM Clock Gating Register (CCM_CCGR18_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4130	CCM Clock Gating Register (CCM_CCGR19)	32	R/W	0000_0002h	5.1.7.6/452
3038_4134	CCM Clock Gating Register (CCM_CCGR19_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4138	CCM Clock Gating Register (CCM_CCGR19_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_413C	CCM Clock Gating Register (CCM_CCGR19_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4140	CCM Clock Gating Register (CCM_CCGR20)	32	R/W	0000_0002h	5.1.7.6/452
3038_4144	CCM Clock Gating Register (CCM_CCGR20_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4148	CCM Clock Gating Register (CCM_CCGR20_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_414C	CCM Clock Gating Register (CCM_CCGR20_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4150	CCM Clock Gating Register (CCM_CCGR21)	32	R/W	0000_0002h	5.1.7.6/452
3038_4154	CCM Clock Gating Register (CCM_CCGR21_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4158	CCM Clock Gating Register (CCM_CCGR21_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_415C	CCM Clock Gating Register (CCM_CCGR21_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4160	CCM Clock Gating Register (CCM_CCGR22)	32	R/W	0000_0002h	5.1.7.6/452
3038_4164	CCM Clock Gating Register (CCM_CCGR22_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4168	CCM Clock Gating Register (CCM_CCGR22_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_416C	CCM Clock Gating Register (CCM_CCGR22_TOG)	32	R/W	0000_0002h	5.1.7.9/459

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4170	CCM Clock Gating Register (CCM_CCGR23)	32	R/W	0000_0002h	5.1.7.6/452
3038_4174	CCM Clock Gating Register (CCM_CCGR23_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4178	CCM Clock Gating Register (CCM_CCGR23_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_417C	CCM Clock Gating Register (CCM_CCGR23_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4180	CCM Clock Gating Register (CCM_CCGR24)	32	R/W	0000_0002h	5.1.7.6/452
3038_4184	CCM Clock Gating Register (CCM_CCGR24_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4188	CCM Clock Gating Register (CCM_CCGR24_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_418C	CCM Clock Gating Register (CCM_CCGR24_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4190	CCM Clock Gating Register (CCM_CCGR25)	32	R/W	0000_0002h	5.1.7.6/452
3038_4194	CCM Clock Gating Register (CCM_CCGR25_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4198	CCM Clock Gating Register (CCM_CCGR25_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_419C	CCM Clock Gating Register (CCM_CCGR25_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_41A0	CCM Clock Gating Register (CCM_CCGR26)	32	R/W	0000_0002h	5.1.7.6/452
3038_41A4	CCM Clock Gating Register (CCM_CCGR26_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_41A8	CCM Clock Gating Register (CCM_CCGR26_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_41AC	CCM Clock Gating Register (CCM_CCGR26_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_41B0	CCM Clock Gating Register (CCM_CCGR27)	32	R/W	0000_0002h	5.1.7.6/452
3038_41B4	CCM Clock Gating Register (CCM_CCGR27_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_41B8	CCM Clock Gating Register (CCM_CCGR27_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_41BC	CCM Clock Gating Register (CCM_CCGR27_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_41C0	CCM Clock Gating Register (CCM_CCGR28)	32	R/W	0000_0002h	5.1.7.6/452
3038_41C4	CCM Clock Gating Register (CCM_CCGR28_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_41C8	CCM Clock Gating Register (CCM_CCGR28_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_41CC	CCM Clock Gating Register (CCM_CCGR28_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_41D0	CCM Clock Gating Register (CCM_CCGR29)	32	R/W	0000_0002h	5.1.7.6/452
3038_41D4	CCM Clock Gating Register (CCM_CCGR29_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_41D8	CCM Clock Gating Register (CCM_CCGR29_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_41DC	CCM Clock Gating Register (CCM_CCGR29_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_41E0	CCM Clock Gating Register (CCM_CCGR30)	32	R/W	0000_0002h	5.1.7.6/452
3038_41E4	CCM Clock Gating Register (CCM_CCGR30_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_41E8	CCM Clock Gating Register (CCM_CCGR30_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_41EC	CCM Clock Gating Register (CCM_CCGR30_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_41F0	CCM Clock Gating Register (CCM_CCGR31)	32	R/W	0000_0002h	5.1.7.6/452
3038_41F4	CCM Clock Gating Register (CCM_CCGR31_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_41F8	CCM Clock Gating Register (CCM_CCGR31_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_41FC	CCM Clock Gating Register (CCM_CCGR31_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4200	CCM Clock Gating Register (CCM_CCGR32)	32	R/W	0000_0002h	5.1.7.6/452
3038_4204	CCM Clock Gating Register (CCM_CCGR32_SET)	32	R/W	0000_0002h	5.1.7.7/455

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4208	CCM Clock Gating Register (CCM_CCGR32_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_420C	CCM Clock Gating Register (CCM_CCGR32_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4210	CCM Clock Gating Register (CCM_CCGR33)	32	R/W	0000_0002h	5.1.7.6/452
3038_4214	CCM Clock Gating Register (CCM_CCGR33_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4218	CCM Clock Gating Register (CCM_CCGR33_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_421C	CCM Clock Gating Register (CCM_CCGR33_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4220	CCM Clock Gating Register (CCM_CCGR34)	32	R/W	0000_0002h	5.1.7.6/452
3038_4224	CCM Clock Gating Register (CCM_CCGR34_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4228	CCM Clock Gating Register (CCM_CCGR34_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_422C	CCM Clock Gating Register (CCM_CCGR34_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4230	CCM Clock Gating Register (CCM_CCGR35)	32	R/W	0000_0002h	5.1.7.6/452
3038_4234	CCM Clock Gating Register (CCM_CCGR35_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4238	CCM Clock Gating Register (CCM_CCGR35_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_423C	CCM Clock Gating Register (CCM_CCGR35_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4240	CCM Clock Gating Register (CCM_CCGR36)	32	R/W	0000_0002h	5.1.7.6/452
3038_4244	CCM Clock Gating Register (CCM_CCGR36_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4248	CCM Clock Gating Register (CCM_CCGR36_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_424C	CCM Clock Gating Register (CCM_CCGR36_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4250	CCM Clock Gating Register (CCM_CCGR37)	32	R/W	0000_0002h	5.1.7.6/452
3038_4254	CCM Clock Gating Register (CCM_CCGR37_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4258	CCM Clock Gating Register (CCM_CCGR37_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_425C	CCM Clock Gating Register (CCM_CCGR37_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4260	CCM Clock Gating Register (CCM_CCGR38)	32	R/W	0000_0002h	5.1.7.6/452
3038_4264	CCM Clock Gating Register (CCM_CCGR38_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4268	CCM Clock Gating Register (CCM_CCGR38_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_426C	CCM Clock Gating Register (CCM_CCGR38_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4270	CCM Clock Gating Register (CCM_CCGR39)	32	R/W	0000_0002h	5.1.7.6/452
3038_4274	CCM Clock Gating Register (CCM_CCGR39_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4278	CCM Clock Gating Register (CCM_CCGR39_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_427C	CCM Clock Gating Register (CCM_CCGR39_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4280	CCM Clock Gating Register (CCM_CCGR40)	32	R/W	0000_0002h	5.1.7.6/452
3038_4284	CCM Clock Gating Register (CCM_CCGR40_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4288	CCM Clock Gating Register (CCM_CCGR40_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_428C	CCM Clock Gating Register (CCM_CCGR40_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4290	CCM Clock Gating Register (CCM_CCGR41)	32	R/W	0000_0002h	5.1.7.6/452
3038_4294	CCM Clock Gating Register (CCM_CCGR41_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4298	CCM Clock Gating Register (CCM_CCGR41_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_429C	CCM Clock Gating Register (CCM_CCGR41_TOG)	32	R/W	0000_0002h	5.1.7.9/459

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_42A0	CCM Clock Gating Register (CCM_CCGR42)	32	R/W	0000_0002h	5.1.7.6/452
3038_42A4	CCM Clock Gating Register (CCM_CCGR42_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_42A8	CCM Clock Gating Register (CCM_CCGR42_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_42AC	CCM Clock Gating Register (CCM_CCGR42_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_42B0	CCM Clock Gating Register (CCM_CCGR43)	32	R/W	0000_0002h	5.1.7.6/452
3038_42B4	CCM Clock Gating Register (CCM_CCGR43_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_42B8	CCM Clock Gating Register (CCM_CCGR43_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_42BC	CCM Clock Gating Register (CCM_CCGR43_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_42C0	CCM Clock Gating Register (CCM_CCGR44)	32	R/W	0000_0002h	5.1.7.6/452
3038_42C4	CCM Clock Gating Register (CCM_CCGR44_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_42C8	CCM Clock Gating Register (CCM_CCGR44_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_42CC	CCM Clock Gating Register (CCM_CCGR44_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_42D0	CCM Clock Gating Register (CCM_CCGR45)	32	R/W	0000_0002h	5.1.7.6/452
3038_42D4	CCM Clock Gating Register (CCM_CCGR45_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_42D8	CCM Clock Gating Register (CCM_CCGR45_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_42DC	CCM Clock Gating Register (CCM_CCGR45_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_42E0	CCM Clock Gating Register (CCM_CCGR46)	32	R/W	0000_0002h	5.1.7.6/452
3038_42E4	CCM Clock Gating Register (CCM_CCGR46_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_42E8	CCM Clock Gating Register (CCM_CCGR46_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_42EC	CCM Clock Gating Register (CCM_CCGR46_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_42F0	CCM Clock Gating Register (CCM_CCGR47)	32	R/W	0000_0002h	5.1.7.6/452
3038_42F4	CCM Clock Gating Register (CCM_CCGR47_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_42F8	CCM Clock Gating Register (CCM_CCGR47_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_42FC	CCM Clock Gating Register (CCM_CCGR47_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4300	CCM Clock Gating Register (CCM_CCGR48)	32	R/W	0000_0002h	5.1.7.6/452
3038_4304	CCM Clock Gating Register (CCM_CCGR48_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4308	CCM Clock Gating Register (CCM_CCGR48_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_430C	CCM Clock Gating Register (CCM_CCGR48_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4310	CCM Clock Gating Register (CCM_CCGR49)	32	R/W	0000_0002h	5.1.7.6/452
3038_4314	CCM Clock Gating Register (CCM_CCGR49_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4318	CCM Clock Gating Register (CCM_CCGR49_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_431C	CCM Clock Gating Register (CCM_CCGR49_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4320	CCM Clock Gating Register (CCM_CCGR50)	32	R/W	0000_0002h	5.1.7.6/452
3038_4324	CCM Clock Gating Register (CCM_CCGR50_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4328	CCM Clock Gating Register (CCM_CCGR50_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_432C	CCM Clock Gating Register (CCM_CCGR50_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4330	CCM Clock Gating Register (CCM_CCGR51)	32	R/W	0000_0002h	5.1.7.6/452
3038_4334	CCM Clock Gating Register (CCM_CCGR51_SET)	32	R/W	0000_0002h	5.1.7.7/455

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4338	CCM Clock Gating Register (CCM_CCGR51_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_433C	CCM Clock Gating Register (CCM_CCGR51_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4340	CCM Clock Gating Register (CCM_CCGR52)	32	R/W	0000_0002h	5.1.7.6/452
3038_4344	CCM Clock Gating Register (CCM_CCGR52_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4348	CCM Clock Gating Register (CCM_CCGR52_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_434C	CCM Clock Gating Register (CCM_CCGR52_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4350	CCM Clock Gating Register (CCM_CCGR53)	32	R/W	0000_0002h	5.1.7.6/452
3038_4354	CCM Clock Gating Register (CCM_CCGR53_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4358	CCM Clock Gating Register (CCM_CCGR53_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_435C	CCM Clock Gating Register (CCM_CCGR53_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4360	CCM Clock Gating Register (CCM_CCGR54)	32	R/W	0000_0002h	5.1.7.6/452
3038_4364	CCM Clock Gating Register (CCM_CCGR54_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4368	CCM Clock Gating Register (CCM_CCGR54_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_436C	CCM Clock Gating Register (CCM_CCGR54_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4370	CCM Clock Gating Register (CCM_CCGR55)	32	R/W	0000_0002h	5.1.7.6/452
3038_4374	CCM Clock Gating Register (CCM_CCGR55_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4378	CCM Clock Gating Register (CCM_CCGR55_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_437C	CCM Clock Gating Register (CCM_CCGR55_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4380	CCM Clock Gating Register (CCM_CCGR56)	32	R/W	0000_0002h	5.1.7.6/452
3038_4384	CCM Clock Gating Register (CCM_CCGR56_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4388	CCM Clock Gating Register (CCM_CCGR56_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_438C	CCM Clock Gating Register (CCM_CCGR56_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4390	CCM Clock Gating Register (CCM_CCGR57)	32	R/W	0000_0002h	5.1.7.6/452
3038_4394	CCM Clock Gating Register (CCM_CCGR57_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4398	CCM Clock Gating Register (CCM_CCGR57_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_439C	CCM Clock Gating Register (CCM_CCGR57_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_43A0	CCM Clock Gating Register (CCM_CCGR58)	32	R/W	0000_0002h	5.1.7.6/452
3038_43A4	CCM Clock Gating Register (CCM_CCGR58_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_43A8	CCM Clock Gating Register (CCM_CCGR58_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_43AC	CCM Clock Gating Register (CCM_CCGR58_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_43B0	CCM Clock Gating Register (CCM_CCGR59)	32	R/W	0000_0002h	5.1.7.6/452
3038_43B4	CCM Clock Gating Register (CCM_CCGR59_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_43B8	CCM Clock Gating Register (CCM_CCGR59_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_43BC	CCM Clock Gating Register (CCM_CCGR59_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_43C0	CCM Clock Gating Register (CCM_CCGR60)	32	R/W	0000_0002h	5.1.7.6/452
3038_43C4	CCM Clock Gating Register (CCM_CCGR60_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_43C8	CCM Clock Gating Register (CCM_CCGR60_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_43CC	CCM Clock Gating Register (CCM_CCGR60_TOG)	32	R/W	0000_0002h	5.1.7.9/459

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_43D0	CCM Clock Gating Register (CCM_CCGR61)	32	R/W	0000_0002h	5.1.7.6/452
3038_43D4	CCM Clock Gating Register (CCM_CCGR61_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_43D8	CCM Clock Gating Register (CCM_CCGR61_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_43DC	CCM Clock Gating Register (CCM_CCGR61_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_43E0	CCM Clock Gating Register (CCM_CCGR62)	32	R/W	0000_0002h	5.1.7.6/452
3038_43E4	CCM Clock Gating Register (CCM_CCGR62_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_43E8	CCM Clock Gating Register (CCM_CCGR62_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_43EC	CCM Clock Gating Register (CCM_CCGR62_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_43F0	CCM Clock Gating Register (CCM_CCGR63)	32	R/W	0000_0002h	5.1.7.6/452
3038_43F4	CCM Clock Gating Register (CCM_CCGR63_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_43F8	CCM Clock Gating Register (CCM_CCGR63_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_43FC	CCM Clock Gating Register (CCM_CCGR63_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4400	CCM Clock Gating Register (CCM_CCGR64)	32	R/W	0000_0002h	5.1.7.6/452
3038_4404	CCM Clock Gating Register (CCM_CCGR64_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4408	CCM Clock Gating Register (CCM_CCGR64_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_440C	CCM Clock Gating Register (CCM_CCGR64_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4410	CCM Clock Gating Register (CCM_CCGR65)	32	R/W	0000_0002h	5.1.7.6/452
3038_4414	CCM Clock Gating Register (CCM_CCGR65_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4418	CCM Clock Gating Register (CCM_CCGR65_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_441C	CCM Clock Gating Register (CCM_CCGR65_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4420	CCM Clock Gating Register (CCM_CCGR66)	32	R/W	0000_0002h	5.1.7.6/452
3038_4424	CCM Clock Gating Register (CCM_CCGR66_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4428	CCM Clock Gating Register (CCM_CCGR66_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_442C	CCM Clock Gating Register (CCM_CCGR66_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4430	CCM Clock Gating Register (CCM_CCGR67)	32	R/W	0000_0002h	5.1.7.6/452
3038_4434	CCM Clock Gating Register (CCM_CCGR67_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4438	CCM Clock Gating Register (CCM_CCGR67_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_443C	CCM Clock Gating Register (CCM_CCGR67_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4440	CCM Clock Gating Register (CCM_CCGR68)	32	R/W	0000_0002h	5.1.7.6/452
3038_4444	CCM Clock Gating Register (CCM_CCGR68_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4448	CCM Clock Gating Register (CCM_CCGR68_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_444C	CCM Clock Gating Register (CCM_CCGR68_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4450	CCM Clock Gating Register (CCM_CCGR69)	32	R/W	0000_0002h	5.1.7.6/452
3038_4454	CCM Clock Gating Register (CCM_CCGR69_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4458	CCM Clock Gating Register (CCM_CCGR69_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_445C	CCM Clock Gating Register (CCM_CCGR69_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4460	CCM Clock Gating Register (CCM_CCGR70)	32	R/W	0000_0002h	5.1.7.6/452
3038_4464	CCM Clock Gating Register (CCM_CCGR70_SET)	32	R/W	0000_0002h	5.1.7.7/455

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4468	CCM Clock Gating Register (CCM_CCGR70_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_446C	CCM Clock Gating Register (CCM_CCGR70_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4470	CCM Clock Gating Register (CCM_CCGR71)	32	R/W	0000_0002h	5.1.7.6/452
3038_4474	CCM Clock Gating Register (CCM_CCGR71_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4478	CCM Clock Gating Register (CCM_CCGR71_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_447C	CCM Clock Gating Register (CCM_CCGR71_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4480	CCM Clock Gating Register (CCM_CCGR72)	32	R/W	0000_0002h	5.1.7.6/452
3038_4484	CCM Clock Gating Register (CCM_CCGR72_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4488	CCM Clock Gating Register (CCM_CCGR72_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_448C	CCM Clock Gating Register (CCM_CCGR72_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4490	CCM Clock Gating Register (CCM_CCGR73)	32	R/W	0000_0002h	5.1.7.6/452
3038_4494	CCM Clock Gating Register (CCM_CCGR73_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4498	CCM Clock Gating Register (CCM_CCGR73_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_449C	CCM Clock Gating Register (CCM_CCGR73_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_44A0	CCM Clock Gating Register (CCM_CCGR74)	32	R/W	0000_0002h	5.1.7.6/452
3038_44A4	CCM Clock Gating Register (CCM_CCGR74_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_44A8	CCM Clock Gating Register (CCM_CCGR74_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_44AC	CCM Clock Gating Register (CCM_CCGR74_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_44B0	CCM Clock Gating Register (CCM_CCGR75)	32	R/W	0000_0002h	5.1.7.6/452
3038_44B4	CCM Clock Gating Register (CCM_CCGR75_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_44B8	CCM Clock Gating Register (CCM_CCGR75_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_44BC	CCM Clock Gating Register (CCM_CCGR75_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_44C0	CCM Clock Gating Register (CCM_CCGR76)	32	R/W	0000_0002h	5.1.7.6/452
3038_44C4	CCM Clock Gating Register (CCM_CCGR76_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_44C8	CCM Clock Gating Register (CCM_CCGR76_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_44CC	CCM Clock Gating Register (CCM_CCGR76_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_44D0	CCM Clock Gating Register (CCM_CCGR77)	32	R/W	0000_0002h	5.1.7.6/452
3038_44D4	CCM Clock Gating Register (CCM_CCGR77_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_44D8	CCM Clock Gating Register (CCM_CCGR77_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_44DC	CCM Clock Gating Register (CCM_CCGR77_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_44E0	CCM Clock Gating Register (CCM_CCGR78)	32	R/W	0000_0002h	5.1.7.6/452
3038_44E4	CCM Clock Gating Register (CCM_CCGR78_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_44E8	CCM Clock Gating Register (CCM_CCGR78_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_44EC	CCM Clock Gating Register (CCM_CCGR78_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_44F0	CCM Clock Gating Register (CCM_CCGR79)	32	R/W	0000_0002h	5.1.7.6/452
3038_44F4	CCM Clock Gating Register (CCM_CCGR79_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_44F8	CCM Clock Gating Register (CCM_CCGR79_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_44FC	CCM Clock Gating Register (CCM_CCGR79_TOG)	32	R/W	0000_0002h	5.1.7.9/459

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4500	CCM Clock Gating Register (CCM_CCGR80)	32	R/W	0000_0002h	5.1.7.6/452
3038_4504	CCM Clock Gating Register (CCM_CCGR80_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4508	CCM Clock Gating Register (CCM_CCGR80_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_450C	CCM Clock Gating Register (CCM_CCGR80_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4510	CCM Clock Gating Register (CCM_CCGR81)	32	R/W	0000_0002h	5.1.7.6/452
3038_4514	CCM Clock Gating Register (CCM_CCGR81_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4518	CCM Clock Gating Register (CCM_CCGR81_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_451C	CCM Clock Gating Register (CCM_CCGR81_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4520	CCM Clock Gating Register (CCM_CCGR82)	32	R/W	0000_0002h	5.1.7.6/452
3038_4524	CCM Clock Gating Register (CCM_CCGR82_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4528	CCM Clock Gating Register (CCM_CCGR82_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_452C	CCM Clock Gating Register (CCM_CCGR82_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4530	CCM Clock Gating Register (CCM_CCGR83)	32	R/W	0000_0002h	5.1.7.6/452
3038_4534	CCM Clock Gating Register (CCM_CCGR83_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4538	CCM Clock Gating Register (CCM_CCGR83_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_453C	CCM Clock Gating Register (CCM_CCGR83_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4540	CCM Clock Gating Register (CCM_CCGR84)	32	R/W	0000_0002h	5.1.7.6/452
3038_4544	CCM Clock Gating Register (CCM_CCGR84_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4548	CCM Clock Gating Register (CCM_CCGR84_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_454C	CCM Clock Gating Register (CCM_CCGR84_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4550	CCM Clock Gating Register (CCM_CCGR85)	32	R/W	0000_0002h	5.1.7.6/452
3038_4554	CCM Clock Gating Register (CCM_CCGR85_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4558	CCM Clock Gating Register (CCM_CCGR85_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_455C	CCM Clock Gating Register (CCM_CCGR85_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4560	CCM Clock Gating Register (CCM_CCGR86)	32	R/W	0000_0002h	5.1.7.6/452
3038_4564	CCM Clock Gating Register (CCM_CCGR86_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4568	CCM Clock Gating Register (CCM_CCGR86_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_456C	CCM Clock Gating Register (CCM_CCGR86_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4570	CCM Clock Gating Register (CCM_CCGR87)	32	R/W	0000_0002h	5.1.7.6/452
3038_4574	CCM Clock Gating Register (CCM_CCGR87_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4578	CCM Clock Gating Register (CCM_CCGR87_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_457C	CCM Clock Gating Register (CCM_CCGR87_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4580	CCM Clock Gating Register (CCM_CCGR88)	32	R/W	0000_0002h	5.1.7.6/452
3038_4584	CCM Clock Gating Register (CCM_CCGR88_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4588	CCM Clock Gating Register (CCM_CCGR88_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_458C	CCM Clock Gating Register (CCM_CCGR88_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4590	CCM Clock Gating Register (CCM_CCGR89)	32	R/W	0000_0002h	5.1.7.6/452
3038_4594	CCM Clock Gating Register (CCM_CCGR89_SET)	32	R/W	0000_0002h	5.1.7.7/455

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4598	CCM Clock Gating Register (CCM_CCGR89_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_459C	CCM Clock Gating Register (CCM_CCGR89_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_45A0	CCM Clock Gating Register (CCM_CCGR90)	32	R/W	0000_0002h	5.1.7.6/452
3038_45A4	CCM Clock Gating Register (CCM_CCGR90_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_45A8	CCM Clock Gating Register (CCM_CCGR90_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_45AC	CCM Clock Gating Register (CCM_CCGR90_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_45B0	CCM Clock Gating Register (CCM_CCGR91)	32	R/W	0000_0002h	5.1.7.6/452
3038_45B4	CCM Clock Gating Register (CCM_CCGR91_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_45B8	CCM Clock Gating Register (CCM_CCGR91_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_45BC	CCM Clock Gating Register (CCM_CCGR91_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_45C0	CCM Clock Gating Register (CCM_CCGR92)	32	R/W	0000_0002h	5.1.7.6/452
3038_45C4	CCM Clock Gating Register (CCM_CCGR92_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_45C8	CCM Clock Gating Register (CCM_CCGR92_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_45CC	CCM Clock Gating Register (CCM_CCGR92_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_45D0	CCM Clock Gating Register (CCM_CCGR93)	32	R/W	0000_0002h	5.1.7.6/452
3038_45D4	CCM Clock Gating Register (CCM_CCGR93_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_45D8	CCM Clock Gating Register (CCM_CCGR93_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_45DC	CCM Clock Gating Register (CCM_CCGR93_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_45E0	CCM Clock Gating Register (CCM_CCGR94)	32	R/W	0000_0002h	5.1.7.6/452
3038_45E4	CCM Clock Gating Register (CCM_CCGR94_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_45E8	CCM Clock Gating Register (CCM_CCGR94_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_45EC	CCM Clock Gating Register (CCM_CCGR94_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_45F0	CCM Clock Gating Register (CCM_CCGR95)	32	R/W	0000_0002h	5.1.7.6/452
3038_45F4	CCM Clock Gating Register (CCM_CCGR95_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_45F8	CCM Clock Gating Register (CCM_CCGR95_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_45FC	CCM Clock Gating Register (CCM_CCGR95_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4600	CCM Clock Gating Register (CCM_CCGR96)	32	R/W	0000_0002h	5.1.7.6/452
3038_4604	CCM Clock Gating Register (CCM_CCGR96_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4608	CCM Clock Gating Register (CCM_CCGR96_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_460C	CCM Clock Gating Register (CCM_CCGR96_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4610	CCM Clock Gating Register (CCM_CCGR97)	32	R/W	0000_0002h	5.1.7.6/452
3038_4614	CCM Clock Gating Register (CCM_CCGR97_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4618	CCM Clock Gating Register (CCM_CCGR97_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_461C	CCM Clock Gating Register (CCM_CCGR97_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4620	CCM Clock Gating Register (CCM_CCGR98)	32	R/W	0000_0002h	5.1.7.6/452
3038_4624	CCM Clock Gating Register (CCM_CCGR98_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4628	CCM Clock Gating Register (CCM_CCGR98_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_462C	CCM Clock Gating Register (CCM_CCGR98_TOG)	32	R/W	0000_0002h	5.1.7.9/459

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4630	CCM Clock Gating Register (CCM_CCGR99)	32	R/W	0000_0002h	5.1.7.6/452
3038_4634	CCM Clock Gating Register (CCM_CCGR99_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4638	CCM Clock Gating Register (CCM_CCGR99_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_463C	CCM Clock Gating Register (CCM_CCGR99_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4640	CCM Clock Gating Register (CCM_CCGR100)	32	R/W	0000_0002h	5.1.7.6/452
3038_4644	CCM Clock Gating Register (CCM_CCGR100_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4648	CCM Clock Gating Register (CCM_CCGR100_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_464C	CCM Clock Gating Register (CCM_CCGR100_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4650	CCM Clock Gating Register (CCM_CCGR101)	32	R/W	0000_0002h	5.1.7.6/452
3038_4654	CCM Clock Gating Register (CCM_CCGR101_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4658	CCM Clock Gating Register (CCM_CCGR101_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_465C	CCM Clock Gating Register (CCM_CCGR101_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4660	CCM Clock Gating Register (CCM_CCGR102)	32	R/W	0000_0002h	5.1.7.6/452
3038_4664	CCM Clock Gating Register (CCM_CCGR102_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4668	CCM Clock Gating Register (CCM_CCGR102_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_466C	CCM Clock Gating Register (CCM_CCGR102_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4670	CCM Clock Gating Register (CCM_CCGR103)	32	R/W	0000_0002h	5.1.7.6/452
3038_4674	CCM Clock Gating Register (CCM_CCGR103_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4678	CCM Clock Gating Register (CCM_CCGR103_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_467C	CCM Clock Gating Register (CCM_CCGR103_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4680	CCM Clock Gating Register (CCM_CCGR104)	32	R/W	0000_0002h	5.1.7.6/452
3038_4684	CCM Clock Gating Register (CCM_CCGR104_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4688	CCM Clock Gating Register (CCM_CCGR104_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_468C	CCM Clock Gating Register (CCM_CCGR104_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4690	CCM Clock Gating Register (CCM_CCGR105)	32	R/W	0000_0002h	5.1.7.6/452
3038_4694	CCM Clock Gating Register (CCM_CCGR105_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4698	CCM Clock Gating Register (CCM_CCGR105_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_469C	CCM Clock Gating Register (CCM_CCGR105_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_46A0	CCM Clock Gating Register (CCM_CCGR106)	32	R/W	0000_0002h	5.1.7.6/452
3038_46A4	CCM Clock Gating Register (CCM_CCGR106_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_46A8	CCM Clock Gating Register (CCM_CCGR106_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_46AC	CCM Clock Gating Register (CCM_CCGR106_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_46B0	CCM Clock Gating Register (CCM_CCGR107)	32	R/W	0000_0002h	5.1.7.6/452
3038_46B4	CCM Clock Gating Register (CCM_CCGR107_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_46B8	CCM Clock Gating Register (CCM_CCGR107_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_46BC	CCM Clock Gating Register (CCM_CCGR107_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_46C0	CCM Clock Gating Register (CCM_CCGR108)	32	R/W	0000_0002h	5.1.7.6/452
3038_46C4	CCM Clock Gating Register (CCM_CCGR108_SET)	32	R/W	0000_0002h	5.1.7.7/455

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_46C8	CCM Clock Gating Register (CCM_CCGR108_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_46CC	CCM Clock Gating Register (CCM_CCGR108_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_46D0	CCM Clock Gating Register (CCM_CCGR109)	32	R/W	0000_0002h	5.1.7.6/452
3038_46D4	CCM Clock Gating Register (CCM_CCGR109_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_46D8	CCM Clock Gating Register (CCM_CCGR109_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_46DC	CCM Clock Gating Register (CCM_CCGR109_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_46E0	CCM Clock Gating Register (CCM_CCGR110)	32	R/W	0000_0002h	5.1.7.6/452
3038_46E4	CCM Clock Gating Register (CCM_CCGR110_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_46E8	CCM Clock Gating Register (CCM_CCGR110_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_46EC	CCM Clock Gating Register (CCM_CCGR110_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_46F0	CCM Clock Gating Register (CCM_CCGR111)	32	R/W	0000_0002h	5.1.7.6/452
3038_46F4	CCM Clock Gating Register (CCM_CCGR111_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_46F8	CCM Clock Gating Register (CCM_CCGR111_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_46FC	CCM Clock Gating Register (CCM_CCGR111_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4700	CCM Clock Gating Register (CCM_CCGR112)	32	R/W	0000_0002h	5.1.7.6/452
3038_4704	CCM Clock Gating Register (CCM_CCGR112_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4708	CCM Clock Gating Register (CCM_CCGR112_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_470C	CCM Clock Gating Register (CCM_CCGR112_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4710	CCM Clock Gating Register (CCM_CCGR113)	32	R/W	0000_0002h	5.1.7.6/452
3038_4714	CCM Clock Gating Register (CCM_CCGR113_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4718	CCM Clock Gating Register (CCM_CCGR113_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_471C	CCM Clock Gating Register (CCM_CCGR113_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4720	CCM Clock Gating Register (CCM_CCGR114)	32	R/W	0000_0002h	5.1.7.6/452
3038_4724	CCM Clock Gating Register (CCM_CCGR114_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4728	CCM Clock Gating Register (CCM_CCGR114_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_472C	CCM Clock Gating Register (CCM_CCGR114_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4730	CCM Clock Gating Register (CCM_CCGR115)	32	R/W	0000_0002h	5.1.7.6/452
3038_4734	CCM Clock Gating Register (CCM_CCGR115_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4738	CCM Clock Gating Register (CCM_CCGR115_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_473C	CCM Clock Gating Register (CCM_CCGR115_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4740	CCM Clock Gating Register (CCM_CCGR116)	32	R/W	0000_0002h	5.1.7.6/452
3038_4744	CCM Clock Gating Register (CCM_CCGR116_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4748	CCM Clock Gating Register (CCM_CCGR116_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_474C	CCM Clock Gating Register (CCM_CCGR116_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4750	CCM Clock Gating Register (CCM_CCGR117)	32	R/W	0000_0002h	5.1.7.6/452
3038_4754	CCM Clock Gating Register (CCM_CCGR117_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4758	CCM Clock Gating Register (CCM_CCGR117_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_475C	CCM Clock Gating Register (CCM_CCGR117_TOG)	32	R/W	0000_0002h	5.1.7.9/459

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4760	CCM Clock Gating Register (CCM_CCGR118)	32	R/W	0000_0002h	5.1.7.6/452
3038_4764	CCM Clock Gating Register (CCM_CCGR118_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4768	CCM Clock Gating Register (CCM_CCGR118_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_476C	CCM Clock Gating Register (CCM_CCGR118_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4770	CCM Clock Gating Register (CCM_CCGR119)	32	R/W	0000_0002h	5.1.7.6/452
3038_4774	CCM Clock Gating Register (CCM_CCGR119_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4778	CCM Clock Gating Register (CCM_CCGR119_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_477C	CCM Clock Gating Register (CCM_CCGR119_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4780	CCM Clock Gating Register (CCM_CCGR120)	32	R/W	0000_0002h	5.1.7.6/452
3038_4784	CCM Clock Gating Register (CCM_CCGR120_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4788	CCM Clock Gating Register (CCM_CCGR120_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_478C	CCM Clock Gating Register (CCM_CCGR120_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4790	CCM Clock Gating Register (CCM_CCGR121)	32	R/W	0000_0002h	5.1.7.6/452
3038_4794	CCM Clock Gating Register (CCM_CCGR121_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4798	CCM Clock Gating Register (CCM_CCGR121_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_479C	CCM Clock Gating Register (CCM_CCGR121_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_47A0	CCM Clock Gating Register (CCM_CCGR122)	32	R/W	0000_0002h	5.1.7.6/452
3038_47A4	CCM Clock Gating Register (CCM_CCGR122_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_47A8	CCM Clock Gating Register (CCM_CCGR122_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_47AC	CCM Clock Gating Register (CCM_CCGR122_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_47B0	CCM Clock Gating Register (CCM_CCGR123)	32	R/W	0000_0002h	5.1.7.6/452
3038_47B4	CCM Clock Gating Register (CCM_CCGR123_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_47B8	CCM Clock Gating Register (CCM_CCGR123_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_47BC	CCM Clock Gating Register (CCM_CCGR123_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_47C0	CCM Clock Gating Register (CCM_CCGR124)	32	R/W	0000_0002h	5.1.7.6/452
3038_47C4	CCM Clock Gating Register (CCM_CCGR124_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_47C8	CCM Clock Gating Register (CCM_CCGR124_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_47CC	CCM Clock Gating Register (CCM_CCGR124_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_47D0	CCM Clock Gating Register (CCM_CCGR125)	32	R/W	0000_0002h	5.1.7.6/452
3038_47D4	CCM Clock Gating Register (CCM_CCGR125_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_47D8	CCM Clock Gating Register (CCM_CCGR125_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_47DC	CCM Clock Gating Register (CCM_CCGR125_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_47E0	CCM Clock Gating Register (CCM_CCGR126)	32	R/W	0000_0002h	5.1.7.6/452
3038_47E4	CCM Clock Gating Register (CCM_CCGR126_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_47E8	CCM Clock Gating Register (CCM_CCGR126_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_47EC	CCM Clock Gating Register (CCM_CCGR126_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_47F0	CCM Clock Gating Register (CCM_CCGR127)	32	R/W	0000_0002h	5.1.7.6/452
3038_47F4	CCM Clock Gating Register (CCM_CCGR127_SET)	32	R/W	0000_0002h	5.1.7.7/455

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_47F8	CCM Clock Gating Register (CCM_CCGR127_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_47FC	CCM Clock Gating Register (CCM_CCGR127_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4800	CCM Clock Gating Register (CCM_CCGR128)	32	R/W	0000_0002h	5.1.7.6/452
3038_4804	CCM Clock Gating Register (CCM_CCGR128_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4808	CCM Clock Gating Register (CCM_CCGR128_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_480C	CCM Clock Gating Register (CCM_CCGR128_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4810	CCM Clock Gating Register (CCM_CCGR129)	32	R/W	0000_0002h	5.1.7.6/452
3038_4814	CCM Clock Gating Register (CCM_CCGR129_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4818	CCM Clock Gating Register (CCM_CCGR129_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_481C	CCM Clock Gating Register (CCM_CCGR129_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4820	CCM Clock Gating Register (CCM_CCGR130)	32	R/W	0000_0002h	5.1.7.6/452
3038_4824	CCM Clock Gating Register (CCM_CCGR130_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4828	CCM Clock Gating Register (CCM_CCGR130_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_482C	CCM Clock Gating Register (CCM_CCGR130_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4830	CCM Clock Gating Register (CCM_CCGR131)	32	R/W	0000_0002h	5.1.7.6/452
3038_4834	CCM Clock Gating Register (CCM_CCGR131_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4838	CCM Clock Gating Register (CCM_CCGR131_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_483C	CCM Clock Gating Register (CCM_CCGR131_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4840	CCM Clock Gating Register (CCM_CCGR132)	32	R/W	0000_0002h	5.1.7.6/452
3038_4844	CCM Clock Gating Register (CCM_CCGR132_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4848	CCM Clock Gating Register (CCM_CCGR132_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_484C	CCM Clock Gating Register (CCM_CCGR132_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4850	CCM Clock Gating Register (CCM_CCGR133)	32	R/W	0000_0002h	5.1.7.6/452
3038_4854	CCM Clock Gating Register (CCM_CCGR133_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4858	CCM Clock Gating Register (CCM_CCGR133_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_485C	CCM Clock Gating Register (CCM_CCGR133_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4860	CCM Clock Gating Register (CCM_CCGR134)	32	R/W	0000_0002h	5.1.7.6/452
3038_4864	CCM Clock Gating Register (CCM_CCGR134_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4868	CCM Clock Gating Register (CCM_CCGR134_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_486C	CCM Clock Gating Register (CCM_CCGR134_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4870	CCM Clock Gating Register (CCM_CCGR135)	32	R/W	0000_0002h	5.1.7.6/452
3038_4874	CCM Clock Gating Register (CCM_CCGR135_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4878	CCM Clock Gating Register (CCM_CCGR135_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_487C	CCM Clock Gating Register (CCM_CCGR135_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4880	CCM Clock Gating Register (CCM_CCGR136)	32	R/W	0000_0002h	5.1.7.6/452
3038_4884	CCM Clock Gating Register (CCM_CCGR136_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4888	CCM Clock Gating Register (CCM_CCGR136_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_488C	CCM Clock Gating Register (CCM_CCGR136_TOG)	32	R/W	0000_0002h	5.1.7.9/459

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4890	CCM Clock Gating Register (CCM_CCGR137)	32	R/W	0000_0002h	5.1.7.6/452
3038_4894	CCM Clock Gating Register (CCM_CCGR137_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4898	CCM Clock Gating Register (CCM_CCGR137_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_489C	CCM Clock Gating Register (CCM_CCGR137_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_48A0	CCM Clock Gating Register (CCM_CCGR138)	32	R/W	0000_0002h	5.1.7.6/452
3038_48A4	CCM Clock Gating Register (CCM_CCGR138_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_48A8	CCM Clock Gating Register (CCM_CCGR138_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_48AC	CCM Clock Gating Register (CCM_CCGR138_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_48B0	CCM Clock Gating Register (CCM_CCGR139)	32	R/W	0000_0002h	5.1.7.6/452
3038_48B4	CCM Clock Gating Register (CCM_CCGR139_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_48B8	CCM Clock Gating Register (CCM_CCGR139_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_48BC	CCM Clock Gating Register (CCM_CCGR139_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_48C0	CCM Clock Gating Register (CCM_CCGR140)	32	R/W	0000_0002h	5.1.7.6/452
3038_48C4	CCM Clock Gating Register (CCM_CCGR140_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_48C8	CCM Clock Gating Register (CCM_CCGR140_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_48CC	CCM Clock Gating Register (CCM_CCGR140_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_48D0	CCM Clock Gating Register (CCM_CCGR141)	32	R/W	0000_0002h	5.1.7.6/452
3038_48D4	CCM Clock Gating Register (CCM_CCGR141_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_48D8	CCM Clock Gating Register (CCM_CCGR141_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_48DC	CCM Clock Gating Register (CCM_CCGR141_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_48E0	CCM Clock Gating Register (CCM_CCGR142)	32	R/W	0000_0002h	5.1.7.6/452
3038_48E4	CCM Clock Gating Register (CCM_CCGR142_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_48E8	CCM Clock Gating Register (CCM_CCGR142_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_48EC	CCM Clock Gating Register (CCM_CCGR142_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_48F0	CCM Clock Gating Register (CCM_CCGR143)	32	R/W	0000_0002h	5.1.7.6/452
3038_48F4	CCM Clock Gating Register (CCM_CCGR143_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_48F8	CCM Clock Gating Register (CCM_CCGR143_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_48FC	CCM Clock Gating Register (CCM_CCGR143_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4900	CCM Clock Gating Register (CCM_CCGR144)	32	R/W	0000_0002h	5.1.7.6/452
3038_4904	CCM Clock Gating Register (CCM_CCGR144_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4908	CCM Clock Gating Register (CCM_CCGR144_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_490C	CCM Clock Gating Register (CCM_CCGR144_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4910	CCM Clock Gating Register (CCM_CCGR145)	32	R/W	0000_0002h	5.1.7.6/452
3038_4914	CCM Clock Gating Register (CCM_CCGR145_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4918	CCM Clock Gating Register (CCM_CCGR145_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_491C	CCM Clock Gating Register (CCM_CCGR145_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4920	CCM Clock Gating Register (CCM_CCGR146)	32	R/W	0000_0002h	5.1.7.6/452
3038_4924	CCM Clock Gating Register (CCM_CCGR146_SET)	32	R/W	0000_0002h	5.1.7.7/455

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4928	CCM Clock Gating Register (CCM_CCGR146_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_492C	CCM Clock Gating Register (CCM_CCGR146_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4930	CCM Clock Gating Register (CCM_CCGR147)	32	R/W	0000_0002h	5.1.7.6/452
3038_4934	CCM Clock Gating Register (CCM_CCGR147_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4938	CCM Clock Gating Register (CCM_CCGR147_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_493C	CCM Clock Gating Register (CCM_CCGR147_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4940	CCM Clock Gating Register (CCM_CCGR148)	32	R/W	0000_0002h	5.1.7.6/452
3038_4944	CCM Clock Gating Register (CCM_CCGR148_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4948	CCM Clock Gating Register (CCM_CCGR148_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_494C	CCM Clock Gating Register (CCM_CCGR148_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4950	CCM Clock Gating Register (CCM_CCGR149)	32	R/W	0000_0002h	5.1.7.6/452
3038_4954	CCM Clock Gating Register (CCM_CCGR149_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4958	CCM Clock Gating Register (CCM_CCGR149_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_495C	CCM Clock Gating Register (CCM_CCGR149_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4960	CCM Clock Gating Register (CCM_CCGR150)	32	R/W	0000_0002h	5.1.7.6/452
3038_4964	CCM Clock Gating Register (CCM_CCGR150_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4968	CCM Clock Gating Register (CCM_CCGR150_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_496C	CCM Clock Gating Register (CCM_CCGR150_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4970	CCM Clock Gating Register (CCM_CCGR151)	32	R/W	0000_0002h	5.1.7.6/452
3038_4974	CCM Clock Gating Register (CCM_CCGR151_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4978	CCM Clock Gating Register (CCM_CCGR151_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_497C	CCM Clock Gating Register (CCM_CCGR151_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4980	CCM Clock Gating Register (CCM_CCGR152)	32	R/W	0000_0002h	5.1.7.6/452
3038_4984	CCM Clock Gating Register (CCM_CCGR152_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4988	CCM Clock Gating Register (CCM_CCGR152_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_498C	CCM Clock Gating Register (CCM_CCGR152_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4990	CCM Clock Gating Register (CCM_CCGR153)	32	R/W	0000_0002h	5.1.7.6/452
3038_4994	CCM Clock Gating Register (CCM_CCGR153_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4998	CCM Clock Gating Register (CCM_CCGR153_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_499C	CCM Clock Gating Register (CCM_CCGR153_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_49A0	CCM Clock Gating Register (CCM_CCGR154)	32	R/W	0000_0002h	5.1.7.6/452
3038_49A4	CCM Clock Gating Register (CCM_CCGR154_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_49A8	CCM Clock Gating Register (CCM_CCGR154_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_49AC	CCM Clock Gating Register (CCM_CCGR154_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_49B0	CCM Clock Gating Register (CCM_CCGR155)	32	R/W	0000_0002h	5.1.7.6/452
3038_49B4	CCM Clock Gating Register (CCM_CCGR155_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_49B8	CCM Clock Gating Register (CCM_CCGR155_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_49BC	CCM Clock Gating Register (CCM_CCGR155_TOG)	32	R/W	0000_0002h	5.1.7.9/459

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_49C0	CCM Clock Gating Register (CCM_CCGR156)	32	R/W	0000_0002h	5.1.7.6/452
3038_49C4	CCM Clock Gating Register (CCM_CCGR156_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_49C8	CCM Clock Gating Register (CCM_CCGR156_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_49CC	CCM Clock Gating Register (CCM_CCGR156_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_49D0	CCM Clock Gating Register (CCM_CCGR157)	32	R/W	0000_0002h	5.1.7.6/452
3038_49D4	CCM Clock Gating Register (CCM_CCGR157_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_49D8	CCM Clock Gating Register (CCM_CCGR157_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_49DC	CCM Clock Gating Register (CCM_CCGR157_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_49E0	CCM Clock Gating Register (CCM_CCGR158)	32	R/W	0000_0002h	5.1.7.6/452
3038_49E4	CCM Clock Gating Register (CCM_CCGR158_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_49E8	CCM Clock Gating Register (CCM_CCGR158_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_49EC	CCM Clock Gating Register (CCM_CCGR158_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_49F0	CCM Clock Gating Register (CCM_CCGR159)	32	R/W	0000_0002h	5.1.7.6/452
3038_49F4	CCM Clock Gating Register (CCM_CCGR159_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_49F8	CCM Clock Gating Register (CCM_CCGR159_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_49FC	CCM Clock Gating Register (CCM_CCGR159_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4A00	CCM Clock Gating Register (CCM_CCGR160)	32	R/W	0000_0002h	5.1.7.6/452
3038_4A04	CCM Clock Gating Register (CCM_CCGR160_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4A08	CCM Clock Gating Register (CCM_CCGR160_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4A0C	CCM Clock Gating Register (CCM_CCGR160_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4A10	CCM Clock Gating Register (CCM_CCGR161)	32	R/W	0000_0002h	5.1.7.6/452
3038_4A14	CCM Clock Gating Register (CCM_CCGR161_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4A18	CCM Clock Gating Register (CCM_CCGR161_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4A1C	CCM Clock Gating Register (CCM_CCGR161_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4A20	CCM Clock Gating Register (CCM_CCGR162)	32	R/W	0000_0002h	5.1.7.6/452
3038_4A24	CCM Clock Gating Register (CCM_CCGR162_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4A28	CCM Clock Gating Register (CCM_CCGR162_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4A2C	CCM Clock Gating Register (CCM_CCGR162_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4A30	CCM Clock Gating Register (CCM_CCGR163)	32	R/W	0000_0002h	5.1.7.6/452
3038_4A34	CCM Clock Gating Register (CCM_CCGR163_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4A38	CCM Clock Gating Register (CCM_CCGR163_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4A3C	CCM Clock Gating Register (CCM_CCGR163_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4A40	CCM Clock Gating Register (CCM_CCGR164)	32	R/W	0000_0002h	5.1.7.6/452
3038_4A44	CCM Clock Gating Register (CCM_CCGR164_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4A48	CCM Clock Gating Register (CCM_CCGR164_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4A4C	CCM Clock Gating Register (CCM_CCGR164_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4A50	CCM Clock Gating Register (CCM_CCGR165)	32	R/W	0000_0002h	5.1.7.6/452
3038_4A54	CCM Clock Gating Register (CCM_CCGR165_SET)	32	R/W	0000_0002h	5.1.7.7/455

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4A58	CCM Clock Gating Register (CCM_CCGR165_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4A5C	CCM Clock Gating Register (CCM_CCGR165_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4A60	CCM Clock Gating Register (CCM_CCGR166)	32	R/W	0000_0002h	5.1.7.6/452
3038_4A64	CCM Clock Gating Register (CCM_CCGR166_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4A68	CCM Clock Gating Register (CCM_CCGR166_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4A6C	CCM Clock Gating Register (CCM_CCGR166_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4A70	CCM Clock Gating Register (CCM_CCGR167)	32	R/W	0000_0002h	5.1.7.6/452
3038_4A74	CCM Clock Gating Register (CCM_CCGR167_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4A78	CCM Clock Gating Register (CCM_CCGR167_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4A7C	CCM Clock Gating Register (CCM_CCGR167_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4A80	CCM Clock Gating Register (CCM_CCGR168)	32	R/W	0000_0002h	5.1.7.6/452
3038_4A84	CCM Clock Gating Register (CCM_CCGR168_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4A88	CCM Clock Gating Register (CCM_CCGR168_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4A8C	CCM Clock Gating Register (CCM_CCGR168_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4A90	CCM Clock Gating Register (CCM_CCGR169)	32	R/W	0000_0002h	5.1.7.6/452
3038_4A94	CCM Clock Gating Register (CCM_CCGR169_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4A98	CCM Clock Gating Register (CCM_CCGR169_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4A9C	CCM Clock Gating Register (CCM_CCGR169_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4AA0	CCM Clock Gating Register (CCM_CCGR170)	32	R/W	0000_0002h	5.1.7.6/452
3038_4AA4	CCM Clock Gating Register (CCM_CCGR170_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4AA8	CCM Clock Gating Register (CCM_CCGR170_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4AAC	CCM Clock Gating Register (CCM_CCGR170_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4AB0	CCM Clock Gating Register (CCM_CCGR171)	32	R/W	0000_0002h	5.1.7.6/452
3038_4AB4	CCM Clock Gating Register (CCM_CCGR171_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4AB8	CCM Clock Gating Register (CCM_CCGR171_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4ABC	CCM Clock Gating Register (CCM_CCGR171_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4AC0	CCM Clock Gating Register (CCM_CCGR172)	32	R/W	0000_0002h	5.1.7.6/452
3038_4AC4	CCM Clock Gating Register (CCM_CCGR172_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4AC8	CCM Clock Gating Register (CCM_CCGR172_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4ACC	CCM Clock Gating Register (CCM_CCGR172_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4AD0	CCM Clock Gating Register (CCM_CCGR173)	32	R/W	0000_0002h	5.1.7.6/452
3038_4AD4	CCM Clock Gating Register (CCM_CCGR173_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4AD8	CCM Clock Gating Register (CCM_CCGR173_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4ADC	CCM Clock Gating Register (CCM_CCGR173_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4AE0	CCM Clock Gating Register (CCM_CCGR174)	32	R/W	0000_0002h	5.1.7.6/452
3038_4AE4	CCM Clock Gating Register (CCM_CCGR174_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4AE8	CCM Clock Gating Register (CCM_CCGR174_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4AEC	CCM Clock Gating Register (CCM_CCGR174_TOG)	32	R/W	0000_0002h	5.1.7.9/459

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4AF0	CCM Clock Gating Register (CCM_CCGR175)	32	R/W	0000_0002h	5.1.7.6/452
3038_4AF4	CCM Clock Gating Register (CCM_CCGR175_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4AF8	CCM Clock Gating Register (CCM_CCGR175_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4AFC	CCM Clock Gating Register (CCM_CCGR175_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4B00	CCM Clock Gating Register (CCM_CCGR176)	32	R/W	0000_0002h	5.1.7.6/452
3038_4B04	CCM Clock Gating Register (CCM_CCGR176_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4B08	CCM Clock Gating Register (CCM_CCGR176_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4B0C	CCM Clock Gating Register (CCM_CCGR176_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4B10	CCM Clock Gating Register (CCM_CCGR177)	32	R/W	0000_0002h	5.1.7.6/452
3038_4B14	CCM Clock Gating Register (CCM_CCGR177_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4B18	CCM Clock Gating Register (CCM_CCGR177_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4B1C	CCM Clock Gating Register (CCM_CCGR177_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4B20	CCM Clock Gating Register (CCM_CCGR178)	32	R/W	0000_0002h	5.1.7.6/452
3038_4B24	CCM Clock Gating Register (CCM_CCGR178_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4B28	CCM Clock Gating Register (CCM_CCGR178_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4B2C	CCM Clock Gating Register (CCM_CCGR178_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4B30	CCM Clock Gating Register (CCM_CCGR179)	32	R/W	0000_0002h	5.1.7.6/452
3038_4B34	CCM Clock Gating Register (CCM_CCGR179_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4B38	CCM Clock Gating Register (CCM_CCGR179_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4B3C	CCM Clock Gating Register (CCM_CCGR179_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4B40	CCM Clock Gating Register (CCM_CCGR180)	32	R/W	0000_0002h	5.1.7.6/452
3038_4B44	CCM Clock Gating Register (CCM_CCGR180_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4B48	CCM Clock Gating Register (CCM_CCGR180_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4B4C	CCM Clock Gating Register (CCM_CCGR180_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4B50	CCM Clock Gating Register (CCM_CCGR181)	32	R/W	0000_0002h	5.1.7.6/452
3038_4B54	CCM Clock Gating Register (CCM_CCGR181_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4B58	CCM Clock Gating Register (CCM_CCGR181_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4B5C	CCM Clock Gating Register (CCM_CCGR181_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4B60	CCM Clock Gating Register (CCM_CCGR182)	32	R/W	0000_0002h	5.1.7.6/452
3038_4B64	CCM Clock Gating Register (CCM_CCGR182_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4B68	CCM Clock Gating Register (CCM_CCGR182_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4B6C	CCM Clock Gating Register (CCM_CCGR182_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4B70	CCM Clock Gating Register (CCM_CCGR183)	32	R/W	0000_0002h	5.1.7.6/452
3038_4B74	CCM Clock Gating Register (CCM_CCGR183_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4B78	CCM Clock Gating Register (CCM_CCGR183_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4B7C	CCM Clock Gating Register (CCM_CCGR183_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4B80	CCM Clock Gating Register (CCM_CCGR184)	32	R/W	0000_0002h	5.1.7.6/452
3038_4B84	CCM Clock Gating Register (CCM_CCGR184_SET)	32	R/W	0000_0002h	5.1.7.7/455

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4B88	CCM Clock Gating Register (CCM_CCGR184_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4B8C	CCM Clock Gating Register (CCM_CCGR184_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4B90	CCM Clock Gating Register (CCM_CCGR185)	32	R/W	0000_0002h	5.1.7.6/452
3038_4B94	CCM Clock Gating Register (CCM_CCGR185_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4B98	CCM Clock Gating Register (CCM_CCGR185_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4B9C	CCM Clock Gating Register (CCM_CCGR185_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4BA0	CCM Clock Gating Register (CCM_CCGR186)	32	R/W	0000_0002h	5.1.7.6/452
3038_4BA4	CCM Clock Gating Register (CCM_CCGR186_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4BA8	CCM Clock Gating Register (CCM_CCGR186_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4BAC	CCM Clock Gating Register (CCM_CCGR186_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4BB0	CCM Clock Gating Register (CCM_CCGR187)	32	R/W	0000_0002h	5.1.7.6/452
3038_4BB4	CCM Clock Gating Register (CCM_CCGR187_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4BB8	CCM Clock Gating Register (CCM_CCGR187_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4BBC	CCM Clock Gating Register (CCM_CCGR187_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4BC0	CCM Clock Gating Register (CCM_CCGR188)	32	R/W	0000_0002h	5.1.7.6/452
3038_4BC4	CCM Clock Gating Register (CCM_CCGR188_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4BC8	CCM Clock Gating Register (CCM_CCGR188_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4BCC	CCM Clock Gating Register (CCM_CCGR188_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4BD0	CCM Clock Gating Register (CCM_CCGR189)	32	R/W	0000_0002h	5.1.7.6/452
3038_4BD4	CCM Clock Gating Register (CCM_CCGR189_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4BD8	CCM Clock Gating Register (CCM_CCGR189_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4BDC	CCM Clock Gating Register (CCM_CCGR189_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4BE0	CCM Clock Gating Register (CCM_CCGR190)	32	R/W	0000_0002h	5.1.7.6/452
3038_4BE4	CCM Clock Gating Register (CCM_CCGR190_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4BE8	CCM Clock Gating Register (CCM_CCGR190_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4BEC	CCM Clock Gating Register (CCM_CCGR190_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_4BF0	CCM Clock Gating Register (CCM_CCGR191)	32	R/W	0000_0002h	5.1.7.6/452
3038_4BF4	CCM Clock Gating Register (CCM_CCGR191_SET)	32	R/W	0000_0002h	5.1.7.7/455
3038_4BF8	CCM Clock Gating Register (CCM_CCGR191_CLR)	32	R/W	0000_0002h	5.1.7.8/457
3038_4BFC	CCM Clock Gating Register (CCM_CCGR191_TOG)	32	R/W	0000_0002h	5.1.7.9/459
3038_8000	Target Register (CCM_TARGET_ROOT0)	32	R/W	1000_0000h	5.1.7.10/461
3038_8004	Target Register (CCM_TARGET_ROOT0_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8008	Target Register (CCM_TARGET_ROOT0_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_800C	Target Register (CCM_TARGET_ROOT0_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8010	Miscellaneous Register (CCM_MISC0)	32	R/W	0000_0000h	5.1.7.14/469
3038_8014	Miscellaneous Register (CCM_MISC_ROOT0_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8018	Miscellaneous Register (CCM_MISC_ROOT0_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_801C	Miscellaneous Register (CCM_MISC_ROOT0_TOG)	32	R/W	0000_0000h	5.1.7.17/472

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8020	Post Divider Register (CCM_POST0)	32	R/W	0000_0000h	5.1.7.18/473
3038_8024	Post Divider Register (CCM_POST_ROOT0_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8028	Post Divider Register (CCM_POST_ROOT0_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_802C	Post Divider Register (CCM_POST_ROOT0_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8030	Pre Divider Register (CCM_PRE0)	32	R/W	1000_0000h	5.1.7.22/485
3038_8034	Pre Divider Register (CCM_PRE_ROOT0_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8038	Pre Divider Register (CCM_PRE_ROOT0_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_803C	Pre Divider Register (CCM_PRE_ROOT0_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8070	Access Control Register (CCM_ACCESS_CTRL0)	32	R/W	0000_0000h	5.1.7.26/497
3038_8074	Access Control Register (CCM_ACCESS_CTRL_ROOT0_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8078	Access Control Register (CCM_ACCESS_CTRL_ROOT0_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_807C	Access Control Register (CCM_ACCESS_CTRL_ROOT0_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8080	Target Register (CCM_TARGET_ROOT1)	32	R/W	1000_0000h	5.1.7.10/461
3038_8084	Target Register (CCM_TARGET_ROOT1_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8088	Target Register (CCM_TARGET_ROOT1_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_808C	Target Register (CCM_TARGET_ROOT1_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8090	Miscellaneous Register (CCM_MISC1)	32	R/W	0000_0000h	5.1.7.14/469
3038_8094	Miscellaneous Register (CCM_MISC_ROOT1_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8098	Miscellaneous Register (CCM_MISC_ROOT1_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_809C	Miscellaneous Register (CCM_MISC_ROOT1_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_80A0	Post Divider Register (CCM_POST1)	32	R/W	0000_0000h	5.1.7.18/473
3038_80A4	Post Divider Register (CCM_POST_ROOT1_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_80A8	Post Divider Register (CCM_POST_ROOT1_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_80AC	Post Divider Register (CCM_POST_ROOT1_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_80B0	Pre Divider Register (CCM_PRE1)	32	R/W	1000_0000h	5.1.7.22/485
3038_80B4	Pre Divider Register (CCM_PRE_ROOT1_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_80B8	Pre Divider Register (CCM_PRE_ROOT1_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_80BC	Pre Divider Register (CCM_PRE_ROOT1_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_80F0	Access Control Register (CCM_ACCESS_CTRL1)	32	R/W	0000_0000h	5.1.7.26/497
3038_80F4	Access Control Register (CCM_ACCESS_CTRL_ROOT1_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_80F8	Access Control Register (CCM_ACCESS_CTRL_ROOT1_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_80FC	Access Control Register (CCM_ACCESS_CTRL_ROOT1_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8100	Target Register (CCM_TARGET_ROOT2)	32	R/W	1000_0000h	5.1.7.10/461
3038_8104	Target Register (CCM_TARGET_ROOT2_SET)	32	R/W	0000_0000h	5.1.7.11/463

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8108	Target Register (CCM_TARGET_ROOT2_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_810C	Target Register (CCM_TARGET_ROOT2_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8110	Miscellaneous Register (CCM_MISC2)	32	R/W	0000_0000h	5.1.7.14/469
3038_8114	Miscellaneous Register (CCM_MISC_ROOT2_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8118	Miscellaneous Register (CCM_MISC_ROOT2_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_811C	Miscellaneous Register (CCM_MISC_ROOT2_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8120	Post Divider Register (CCM_POST2)	32	R/W	0000_0000h	5.1.7.18/473
3038_8124	Post Divider Register (CCM_POST_ROOT2_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8128	Post Divider Register (CCM_POST_ROOT2_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_812C	Post Divider Register (CCM_POST_ROOT2_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8130	Pre Divider Register (CCM_PRE2)	32	R/W	1000_0000h	5.1.7.22/485
3038_8134	Pre Divider Register (CCM_PRE_ROOT2_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8138	Pre Divider Register (CCM_PRE_ROOT2_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_813C	Pre Divider Register (CCM_PRE_ROOT2_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8170	Access Control Register (CCM_ACCESS_CTRL2)	32	R/W	0000_0000h	5.1.7.26/497
3038_8174	Access Control Register (CCM_ACCESS_CTRL_ROOT2_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8178	Access Control Register (CCM_ACCESS_CTRL_ROOT2_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_817C	Access Control Register (CCM_ACCESS_CTRL_ROOT2_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8180	Target Register (CCM_TARGET_ROOT3)	32	R/W	1000_0000h	5.1.7.10/461
3038_8184	Target Register (CCM_TARGET_ROOT3_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8188	Target Register (CCM_TARGET_ROOT3_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_818C	Target Register (CCM_TARGET_ROOT3_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8190	Miscellaneous Register (CCM_MISC3)	32	R/W	0000_0000h	5.1.7.14/469
3038_8194	Miscellaneous Register (CCM_MISC_ROOT3_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8198	Miscellaneous Register (CCM_MISC_ROOT3_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_819C	Miscellaneous Register (CCM_MISC_ROOT3_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_81A0	Post Divider Register (CCM_POST3)	32	R/W	0000_0000h	5.1.7.18/473
3038_81A4	Post Divider Register (CCM_POST_ROOT3_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_81A8	Post Divider Register (CCM_POST_ROOT3_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_81AC	Post Divider Register (CCM_POST_ROOT3_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_81B0	Pre Divider Register (CCM_PRE3)	32	R/W	1000_0000h	5.1.7.22/485
3038_81B4	Pre Divider Register (CCM_PRE_ROOT3_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_81B8	Pre Divider Register (CCM_PRE_ROOT3_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_81BC	Pre Divider Register (CCM_PRE_ROOT3_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_81F0	Access Control Register (CCM_ACCESS_CTRL3)	32	R/W	0000_0000h	5.1.7.26/497

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_81F4	Access Control Register (CCM_ACCESS_CTRL_ROOT3_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_81F8	Access Control Register (CCM_ACCESS_CTRL_ROOT3_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_81FC	Access Control Register (CCM_ACCESS_CTRL_ROOT3_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8200	Target Register (CCM_TARGET_ROOT4)	32	R/W	1000_0000h	5.1.7.10/461
3038_8204	Target Register (CCM_TARGET_ROOT4_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8208	Target Register (CCM_TARGET_ROOT4_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_820C	Target Register (CCM_TARGET_ROOT4_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8210	Miscellaneous Register (CCM_MISC4)	32	R/W	0000_0000h	5.1.7.14/469
3038_8214	Miscellaneous Register (CCM_MISC_ROOT4_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8218	Miscellaneous Register (CCM_MISC_ROOT4_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_821C	Miscellaneous Register (CCM_MISC_ROOT4_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8220	Post Divider Register (CCM_POST4)	32	R/W	0000_0000h	5.1.7.18/473
3038_8224	Post Divider Register (CCM_POST_ROOT4_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8228	Post Divider Register (CCM_POST_ROOT4_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_822C	Post Divider Register (CCM_POST_ROOT4_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8230	Pre Divider Register (CCM_PRE4)	32	R/W	1000_0000h	5.1.7.22/485
3038_8234	Pre Divider Register (CCM_PRE_ROOT4_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8238	Pre Divider Register (CCM_PRE_ROOT4_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_823C	Pre Divider Register (CCM_PRE_ROOT4_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8270	Access Control Register (CCM_ACCESS_CTRL4)	32	R/W	0000_0000h	5.1.7.26/497
3038_8274	Access Control Register (CCM_ACCESS_CTRL_ROOT4_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8278	Access Control Register (CCM_ACCESS_CTRL_ROOT4_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_827C	Access Control Register (CCM_ACCESS_CTRL_ROOT4_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8280	Target Register (CCM_TARGET_ROOT5)	32	R/W	1000_0000h	5.1.7.10/461
3038_8284	Target Register (CCM_TARGET_ROOT5_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8288	Target Register (CCM_TARGET_ROOT5_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_828C	Target Register (CCM_TARGET_ROOT5_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8290	Miscellaneous Register (CCM_MISC5)	32	R/W	0000_0000h	5.1.7.14/469
3038_8294	Miscellaneous Register (CCM_MISC_ROOT5_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8298	Miscellaneous Register (CCM_MISC_ROOT5_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_829C	Miscellaneous Register (CCM_MISC_ROOT5_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_82A0	Post Divider Register (CCM_POST5)	32	R/W	0000_0000h	5.1.7.18/473
3038_82A4	Post Divider Register (CCM_POST_ROOT5_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_82A8	Post Divider Register (CCM_POST_ROOT5_CLR)	32	R/W	0000_0000h	5.1.7.20/479

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_82AC	Post Divider Register (CCM_POST_ROOT5_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_82B0	Pre Divider Register (CCM_PRE5)	32	R/W	1000_0000h	5.1.7.22/485
3038_82B4	Pre Divider Register (CCM_PRE_ROOT5_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_82B8	Pre Divider Register (CCM_PRE_ROOT5_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_82BC	Pre Divider Register (CCM_PRE_ROOT5_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_82F0	Access Control Register (CCM_ACCESS_CTRL5)	32	R/W	0000_0000h	5.1.7.26/497
3038_82F4	Access Control Register (CCM_ACCESS_CTRL_ROOT5_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_82F8	Access Control Register (CCM_ACCESS_CTRL_ROOT5_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_82FC	Access Control Register (CCM_ACCESS_CTRL_ROOT5_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8300	Target Register (CCM_TARGET_ROOT6)	32	R/W	1000_0000h	5.1.7.10/461
3038_8304	Target Register (CCM_TARGET_ROOT6_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8308	Target Register (CCM_TARGET_ROOT6_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_830C	Target Register (CCM_TARGET_ROOT6_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8310	Miscellaneous Register (CCM_MISC6)	32	R/W	0000_0000h	5.1.7.14/469
3038_8314	Miscellaneous Register (CCM_MISC_ROOT6_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8318	Miscellaneous Register (CCM_MISC_ROOT6_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_831C	Miscellaneous Register (CCM_MISC_ROOT6_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8320	Post Divider Register (CCM_POST6)	32	R/W	0000_0000h	5.1.7.18/473
3038_8324	Post Divider Register (CCM_POST_ROOT6_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8328	Post Divider Register (CCM_POST_ROOT6_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_832C	Post Divider Register (CCM_POST_ROOT6_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8330	Pre Divider Register (CCM_PRE6)	32	R/W	1000_0000h	5.1.7.22/485
3038_8334	Pre Divider Register (CCM_PRE_ROOT6_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8338	Pre Divider Register (CCM_PRE_ROOT6_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_833C	Pre Divider Register (CCM_PRE_ROOT6_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8370	Access Control Register (CCM_ACCESS_CTRL6)	32	R/W	0000_0000h	5.1.7.26/497
3038_8374	Access Control Register (CCM_ACCESS_CTRL_ROOT6_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8378	Access Control Register (CCM_ACCESS_CTRL_ROOT6_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_837C	Access Control Register (CCM_ACCESS_CTRL_ROOT6_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8380	Target Register (CCM_TARGET_ROOT7)	32	R/W	1000_0000h	5.1.7.10/461
3038_8384	Target Register (CCM_TARGET_ROOT7_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8388	Target Register (CCM_TARGET_ROOT7_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_838C	Target Register (CCM_TARGET_ROOT7_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8390	Miscellaneous Register (CCM_MISC7)	32	R/W	0000_0000h	5.1.7.14/469

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8394	Miscellaneous Register (CCM_MISC_ROOT7_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8398	Miscellaneous Register (CCM_MISC_ROOT7_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_839C	Miscellaneous Register (CCM_MISC_ROOT7_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_83A0	Post Divider Register (CCM_POST7)	32	R/W	0000_0000h	5.1.7.18/473
3038_83A4	Post Divider Register (CCM_POST_ROOT7_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_83A8	Post Divider Register (CCM_POST_ROOT7_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_83AC	Post Divider Register (CCM_POST_ROOT7_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_83B0	Pre Divider Register (CCM_PRE7)	32	R/W	1000_0000h	5.1.7.22/485
3038_83B4	Pre Divider Register (CCM_PRE_ROOT7_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_83B8	Pre Divider Register (CCM_PRE_ROOT7_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_83BC	Pre Divider Register (CCM_PRE_ROOT7_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_83F0	Access Control Register (CCM_ACCESS_CTRL7)	32	R/W	0000_0000h	5.1.7.26/497
3038_83F4	Access Control Register (CCM_ACCESS_CTRL_ROOT7_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_83F8	Access Control Register (CCM_ACCESS_CTRL_ROOT7_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_83FC	Access Control Register (CCM_ACCESS_CTRL_ROOT7_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8400	Target Register (CCM_TARGET_ROOT8)	32	R/W	1000_0000h	5.1.7.10/461
3038_8404	Target Register (CCM_TARGET_ROOT8_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8408	Target Register (CCM_TARGET_ROOT8_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_840C	Target Register (CCM_TARGET_ROOT8_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8410	Miscellaneous Register (CCM_MISC8)	32	R/W	0000_0000h	5.1.7.14/469
3038_8414	Miscellaneous Register (CCM_MISC_ROOT8_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8418	Miscellaneous Register (CCM_MISC_ROOT8_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_841C	Miscellaneous Register (CCM_MISC_ROOT8_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8420	Post Divider Register (CCM_POST8)	32	R/W	0000_0000h	5.1.7.18/473
3038_8424	Post Divider Register (CCM_POST_ROOT8_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8428	Post Divider Register (CCM_POST_ROOT8_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_842C	Post Divider Register (CCM_POST_ROOT8_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8430	Pre Divider Register (CCM_PRE8)	32	R/W	1000_0000h	5.1.7.22/485
3038_8434	Pre Divider Register (CCM_PRE_ROOT8_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8438	Pre Divider Register (CCM_PRE_ROOT8_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_843C	Pre Divider Register (CCM_PRE_ROOT8_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8470	Access Control Register (CCM_ACCESS_CTRL8)	32	R/W	0000_0000h	5.1.7.26/497
3038_8474	Access Control Register (CCM_ACCESS_CTRL_ROOT8_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8478	Access Control Register (CCM_ACCESS_CTRL_ROOT8_CLR)	32	R/W	0000_0000h	5.1.7.28/502

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_847C	Access Control Register (CCM_ACCESS_CTRL_ROOT8_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8480	Target Register (CCM_TARGET_ROOT9)	32	R/W	1000_0000h	5.1.7.10/461
3038_8484	Target Register (CCM_TARGET_ROOT9_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8488	Target Register (CCM_TARGET_ROOT9_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_848C	Target Register (CCM_TARGET_ROOT9_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8490	Miscellaneous Register (CCM_MISC9)	32	R/W	0000_0000h	5.1.7.14/469
3038_8494	Miscellaneous Register (CCM_MISC_ROOT9_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8498	Miscellaneous Register (CCM_MISC_ROOT9_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_849C	Miscellaneous Register (CCM_MISC_ROOT9_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_84A0	Post Divider Register (CCM_POST9)	32	R/W	0000_0000h	5.1.7.18/473
3038_84A4	Post Divider Register (CCM_POST_ROOT9_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_84A8	Post Divider Register (CCM_POST_ROOT9_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_84AC	Post Divider Register (CCM_POST_ROOT9_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_84B0	Pre Divider Register (CCM_PRE9)	32	R/W	1000_0000h	5.1.7.22/485
3038_84B4	Pre Divider Register (CCM_PRE_ROOT9_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_84B8	Pre Divider Register (CCM_PRE_ROOT9_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_84BC	Pre Divider Register (CCM_PRE_ROOT9_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_84F0	Access Control Register (CCM_ACCESS_CTRL9)	32	R/W	0000_0000h	5.1.7.26/497
3038_84F4	Access Control Register (CCM_ACCESS_CTRL_ROOT9_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_84F8	Access Control Register (CCM_ACCESS_CTRL_ROOT9_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_84FC	Access Control Register (CCM_ACCESS_CTRL_ROOT9_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8500	Target Register (CCM_TARGET_ROOT10)	32	R/W	1000_0000h	5.1.7.10/461
3038_8504	Target Register (CCM_TARGET_ROOT10_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8508	Target Register (CCM_TARGET_ROOT10_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_850C	Target Register (CCM_TARGET_ROOT10_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8510	Miscellaneous Register (CCM_MISC10)	32	R/W	0000_0000h	5.1.7.14/469
3038_8514	Miscellaneous Register (CCM_MISC_ROOT10_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8518	Miscellaneous Register (CCM_MISC_ROOT10_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_851C	Miscellaneous Register (CCM_MISC_ROOT10_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8520	Post Divider Register (CCM_POST10)	32	R/W	0000_0000h	5.1.7.18/473
3038_8524	Post Divider Register (CCM_POST_ROOT10_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8528	Post Divider Register (CCM_POST_ROOT10_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_852C	Post Divider Register (CCM_POST_ROOT10_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8530	Pre Divider Register (CCM_PRE10)	32	R/W	1000_0000h	5.1.7.22/485
3038_8534	Pre Divider Register (CCM_PRE_ROOT10_SET)	32	R/W	0000_0000h	5.1.7.23/488

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8538	Pre Divider Register (CCM_PRE_ROOT10_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_853C	Pre Divider Register (CCM_PRE_ROOT10_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8570	Access Control Register (CCM_ACCESS_CTRL10)	32	R/W	0000_0000h	5.1.7.26/497
3038_8574	Access Control Register (CCM_ACCESS_CTRL_ROOT10_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8578	Access Control Register (CCM_ACCESS_CTRL_ROOT10_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_857C	Access Control Register (CCM_ACCESS_CTRL_ROOT10_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8580	Target Register (CCM_TARGET_ROOT11)	32	R/W	1000_0000h	5.1.7.10/461
3038_8584	Target Register (CCM_TARGET_ROOT11_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8588	Target Register (CCM_TARGET_ROOT11_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_858C	Target Register (CCM_TARGET_ROOT11_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8590	Miscellaneous Register (CCM_MISC11)	32	R/W	0000_0000h	5.1.7.14/469
3038_8594	Miscellaneous Register (CCM_MISC_ROOT11_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8598	Miscellaneous Register (CCM_MISC_ROOT11_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_859C	Miscellaneous Register (CCM_MISC_ROOT11_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_85A0	Post Divider Register (CCM_POST11)	32	R/W	0000_0000h	5.1.7.18/473
3038_85A4	Post Divider Register (CCM_POST_ROOT11_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_85A8	Post Divider Register (CCM_POST_ROOT11_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_85AC	Post Divider Register (CCM_POST_ROOT11_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_85B0	Pre Divider Register (CCM_PRE11)	32	R/W	1000_0000h	5.1.7.22/485
3038_85B4	Pre Divider Register (CCM_PRE_ROOT11_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_85B8	Pre Divider Register (CCM_PRE_ROOT11_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_85BC	Pre Divider Register (CCM_PRE_ROOT11_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_85F0	Access Control Register (CCM_ACCESS_CTRL11)	32	R/W	0000_0000h	5.1.7.26/497
3038_85F4	Access Control Register (CCM_ACCESS_CTRL_ROOT11_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_85F8	Access Control Register (CCM_ACCESS_CTRL_ROOT11_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_85FC	Access Control Register (CCM_ACCESS_CTRL_ROOT11_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8600	Target Register (CCM_TARGET_ROOT12)	32	R/W	1000_0000h	5.1.7.10/461
3038_8604	Target Register (CCM_TARGET_ROOT12_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8608	Target Register (CCM_TARGET_ROOT12_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_860C	Target Register (CCM_TARGET_ROOT12_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8610	Miscellaneous Register (CCM_MISC12)	32	R/W	0000_0000h	5.1.7.14/469
3038_8614	Miscellaneous Register (CCM_MISC_ROOT12_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8618	Miscellaneous Register (CCM_MISC_ROOT12_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_861C	Miscellaneous Register (CCM_MISC_ROOT12_TOG)	32	R/W	0000_0000h	5.1.7.17/472

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8620	Post Divider Register (CCM_POST12)	32	R/W	0000_0000h	5.1.7.18/473
3038_8624	Post Divider Register (CCM_POST_ROOT12_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8628	Post Divider Register (CCM_POST_ROOT12_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_862C	Post Divider Register (CCM_POST_ROOT12_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8630	Pre Divider Register (CCM_PRE12)	32	R/W	1000_0000h	5.1.7.22/485
3038_8634	Pre Divider Register (CCM_PRE_ROOT12_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8638	Pre Divider Register (CCM_PRE_ROOT12_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_863C	Pre Divider Register (CCM_PRE_ROOT12_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8670	Access Control Register (CCM_ACCESS_CTRL12)	32	R/W	0000_0000h	5.1.7.26/497
3038_8674	Access Control Register (CCM_ACCESS_CTRL_ROOT12_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8678	Access Control Register (CCM_ACCESS_CTRL_ROOT12_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_867C	Access Control Register (CCM_ACCESS_CTRL_ROOT12_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8680	Target Register (CCM_TARGET_ROOT13)	32	R/W	1000_0000h	5.1.7.10/461
3038_8684	Target Register (CCM_TARGET_ROOT13_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8688	Target Register (CCM_TARGET_ROOT13_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_868C	Target Register (CCM_TARGET_ROOT13_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8690	Miscellaneous Register (CCM_MISC13)	32	R/W	0000_0000h	5.1.7.14/469
3038_8694	Miscellaneous Register (CCM_MISC_ROOT13_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8698	Miscellaneous Register (CCM_MISC_ROOT13_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_869C	Miscellaneous Register (CCM_MISC_ROOT13_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_86A0	Post Divider Register (CCM_POST13)	32	R/W	0000_0000h	5.1.7.18/473
3038_86A4	Post Divider Register (CCM_POST_ROOT13_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_86A8	Post Divider Register (CCM_POST_ROOT13_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_86AC	Post Divider Register (CCM_POST_ROOT13_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_86B0	Pre Divider Register (CCM_PRE13)	32	R/W	1000_0000h	5.1.7.22/485
3038_86B4	Pre Divider Register (CCM_PRE_ROOT13_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_86B8	Pre Divider Register (CCM_PRE_ROOT13_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_86BC	Pre Divider Register (CCM_PRE_ROOT13_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_86F0	Access Control Register (CCM_ACCESS_CTRL13)	32	R/W	0000_0000h	5.1.7.26/497
3038_86F4	Access Control Register (CCM_ACCESS_CTRL_ROOT13_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_86F8	Access Control Register (CCM_ACCESS_CTRL_ROOT13_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_86FC	Access Control Register (CCM_ACCESS_CTRL_ROOT13_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8700	Target Register (CCM_TARGET_ROOT14)	32	R/W	1000_0000h	5.1.7.10/461
3038_8704	Target Register (CCM_TARGET_ROOT14_SET)	32	R/W	0000_0000h	5.1.7.11/463

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8708	Target Register (CCM_TARGET_ROOT14_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_870C	Target Register (CCM_TARGET_ROOT14_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8710	Miscellaneous Register (CCM_MISC14)	32	R/W	0000_0000h	5.1.7.14/469
3038_8714	Miscellaneous Register (CCM_MISC_ROOT14_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8718	Miscellaneous Register (CCM_MISC_ROOT14_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_871C	Miscellaneous Register (CCM_MISC_ROOT14_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8720	Post Divider Register (CCM_POST14)	32	R/W	0000_0000h	5.1.7.18/473
3038_8724	Post Divider Register (CCM_POST_ROOT14_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8728	Post Divider Register (CCM_POST_ROOT14_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_872C	Post Divider Register (CCM_POST_ROOT14_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8730	Pre Divider Register (CCM_PRE14)	32	R/W	1000_0000h	5.1.7.22/485
3038_8734	Pre Divider Register (CCM_PRE_ROOT14_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8738	Pre Divider Register (CCM_PRE_ROOT14_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_873C	Pre Divider Register (CCM_PRE_ROOT14_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8770	Access Control Register (CCM_ACCESS_CTRL14)	32	R/W	0000_0000h	5.1.7.26/497
3038_8774	Access Control Register (CCM_ACCESS_CTRL_ROOT14_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8778	Access Control Register (CCM_ACCESS_CTRL_ROOT14_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_877C	Access Control Register (CCM_ACCESS_CTRL_ROOT14_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8780	Target Register (CCM_TARGET_ROOT15)	32	R/W	1000_0000h	5.1.7.10/461
3038_8784	Target Register (CCM_TARGET_ROOT15_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8788	Target Register (CCM_TARGET_ROOT15_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_878C	Target Register (CCM_TARGET_ROOT15_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8790	Miscellaneous Register (CCM_MISC15)	32	R/W	0000_0000h	5.1.7.14/469
3038_8794	Miscellaneous Register (CCM_MISC_ROOT15_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8798	Miscellaneous Register (CCM_MISC_ROOT15_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_879C	Miscellaneous Register (CCM_MISC_ROOT15_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_87A0	Post Divider Register (CCM_POST15)	32	R/W	0000_0000h	5.1.7.18/473
3038_87A4	Post Divider Register (CCM_POST_ROOT15_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_87A8	Post Divider Register (CCM_POST_ROOT15_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_87AC	Post Divider Register (CCM_POST_ROOT15_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_87B0	Pre Divider Register (CCM_PRE15)	32	R/W	1000_0000h	5.1.7.22/485
3038_87B4	Pre Divider Register (CCM_PRE_ROOT15_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_87B8	Pre Divider Register (CCM_PRE_ROOT15_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_87BC	Pre Divider Register (CCM_PRE_ROOT15_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_87F0	Access Control Register (CCM_ACCESS_CTRL15)	32	R/W	0000_0000h	5.1.7.26/497

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_87F4	Access Control Register (CCM_ACCESS_CTRL_ROOT15_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_87F8	Access Control Register (CCM_ACCESS_CTRL_ROOT15_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_87FC	Access Control Register (CCM_ACCESS_CTRL_ROOT15_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8800	Target Register (CCM_TARGET_ROOT16)	32	R/W	1000_0000h	5.1.7.10/461
3038_8804	Target Register (CCM_TARGET_ROOT16_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8808	Target Register (CCM_TARGET_ROOT16_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_880C	Target Register (CCM_TARGET_ROOT16_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8810	Miscellaneous Register (CCM_MISC16)	32	R/W	0000_0000h	5.1.7.14/469
3038_8814	Miscellaneous Register (CCM_MISC_ROOT16_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8818	Miscellaneous Register (CCM_MISC_ROOT16_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_881C	Miscellaneous Register (CCM_MISC_ROOT16_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8820	Post Divider Register (CCM_POST16)	32	R/W	0000_0000h	5.1.7.18/473
3038_8824	Post Divider Register (CCM_POST_ROOT16_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8828	Post Divider Register (CCM_POST_ROOT16_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_882C	Post Divider Register (CCM_POST_ROOT16_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8830	Pre Divider Register (CCM_PRE16)	32	R/W	1000_0000h	5.1.7.22/485
3038_8834	Pre Divider Register (CCM_PRE_ROOT16_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8838	Pre Divider Register (CCM_PRE_ROOT16_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_883C	Pre Divider Register (CCM_PRE_ROOT16_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8870	Access Control Register (CCM_ACCESS_CTRL16)	32	R/W	0000_0000h	5.1.7.26/497
3038_8874	Access Control Register (CCM_ACCESS_CTRL_ROOT16_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8878	Access Control Register (CCM_ACCESS_CTRL_ROOT16_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_887C	Access Control Register (CCM_ACCESS_CTRL_ROOT16_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8880	Target Register (CCM_TARGET_ROOT17)	32	R/W	1000_0000h	5.1.7.10/461
3038_8884	Target Register (CCM_TARGET_ROOT17_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8888	Target Register (CCM_TARGET_ROOT17_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_888C	Target Register (CCM_TARGET_ROOT17_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8890	Miscellaneous Register (CCM_MISC17)	32	R/W	0000_0000h	5.1.7.14/469
3038_8894	Miscellaneous Register (CCM_MISC_ROOT17_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8898	Miscellaneous Register (CCM_MISC_ROOT17_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_889C	Miscellaneous Register (CCM_MISC_ROOT17_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_88A0	Post Divider Register (CCM_POST17)	32	R/W	0000_0000h	5.1.7.18/473
3038_88A4	Post Divider Register (CCM_POST_ROOT17_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_88A8	Post Divider Register (CCM_POST_ROOT17_CLR)	32	R/W	0000_0000h	5.1.7.20/479

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_88AC	Post Divider Register (CCM_POST_ROOT17_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_88B0	Pre Divider Register (CCM_PRE17)	32	R/W	1000_0000h	5.1.7.22/485
3038_88B4	Pre Divider Register (CCM_PRE_ROOT17_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_88B8	Pre Divider Register (CCM_PRE_ROOT17_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_88BC	Pre Divider Register (CCM_PRE_ROOT17_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_88F0	Access Control Register (CCM_ACCESS_CTRL17)	32	R/W	0000_0000h	5.1.7.26/497
3038_88F4	Access Control Register (CCM_ACCESS_CTRL_ROOT17_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_88F8	Access Control Register (CCM_ACCESS_CTRL_ROOT17_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_88FC	Access Control Register (CCM_ACCESS_CTRL_ROOT17_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8900	Target Register (CCM_TARGET_ROOT18)	32	R/W	1000_0000h	5.1.7.10/461
3038_8904	Target Register (CCM_TARGET_ROOT18_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8908	Target Register (CCM_TARGET_ROOT18_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_890C	Target Register (CCM_TARGET_ROOT18_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8910	Miscellaneous Register (CCM_MISC18)	32	R/W	0000_0000h	5.1.7.14/469
3038_8914	Miscellaneous Register (CCM_MISC_ROOT18_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8918	Miscellaneous Register (CCM_MISC_ROOT18_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_891C	Miscellaneous Register (CCM_MISC_ROOT18_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8920	Post Divider Register (CCM_POST18)	32	R/W	0000_0000h	5.1.7.18/473
3038_8924	Post Divider Register (CCM_POST_ROOT18_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8928	Post Divider Register (CCM_POST_ROOT18_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_892C	Post Divider Register (CCM_POST_ROOT18_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8930	Pre Divider Register (CCM_PRE18)	32	R/W	1000_0000h	5.1.7.22/485
3038_8934	Pre Divider Register (CCM_PRE_ROOT18_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8938	Pre Divider Register (CCM_PRE_ROOT18_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_893C	Pre Divider Register (CCM_PRE_ROOT18_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8970	Access Control Register (CCM_ACCESS_CTRL18)	32	R/W	0000_0000h	5.1.7.26/497
3038_8974	Access Control Register (CCM_ACCESS_CTRL_ROOT18_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8978	Access Control Register (CCM_ACCESS_CTRL_ROOT18_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_897C	Access Control Register (CCM_ACCESS_CTRL_ROOT18_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8980	Target Register (CCM_TARGET_ROOT19)	32	R/W	1000_0000h	5.1.7.10/461
3038_8984	Target Register (CCM_TARGET_ROOT19_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8988	Target Register (CCM_TARGET_ROOT19_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_898C	Target Register (CCM_TARGET_ROOT19_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8990	Miscellaneous Register (CCM_MISC19)	32	R/W	0000_0000h	5.1.7.14/469

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8994	Miscellaneous Register (CCM_MISC_ROOT19_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8998	Miscellaneous Register (CCM_MISC_ROOT19_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_899C	Miscellaneous Register (CCM_MISC_ROOT19_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_89A0	Post Divider Register (CCM_POST19)	32	R/W	0000_0000h	5.1.7.18/473
3038_89A4	Post Divider Register (CCM_POST_ROOT19_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_89A8	Post Divider Register (CCM_POST_ROOT19_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_89AC	Post Divider Register (CCM_POST_ROOT19_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_89B0	Pre Divider Register (CCM_PRE19)	32	R/W	1000_0000h	5.1.7.22/485
3038_89B4	Pre Divider Register (CCM_PRE_ROOT19_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_89B8	Pre Divider Register (CCM_PRE_ROOT19_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_89BC	Pre Divider Register (CCM_PRE_ROOT19_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_89F0	Access Control Register (CCM_ACCESS_CTRL19)	32	R/W	0000_0000h	5.1.7.26/497
3038_89F4	Access Control Register (CCM_ACCESS_CTRL_ROOT19_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_89F8	Access Control Register (CCM_ACCESS_CTRL_ROOT19_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_89FC	Access Control Register (CCM_ACCESS_CTRL_ROOT19_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8A00	Target Register (CCM_TARGET_ROOT20)	32	R/W	1000_0000h	5.1.7.10/461
3038_8A04	Target Register (CCM_TARGET_ROOT20_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8A08	Target Register (CCM_TARGET_ROOT20_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_8A0C	Target Register (CCM_TARGET_ROOT20_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8A10	Miscellaneous Register (CCM_MISC20)	32	R/W	0000_0000h	5.1.7.14/469
3038_8A14	Miscellaneous Register (CCM_MISC_ROOT20_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8A18	Miscellaneous Register (CCM_MISC_ROOT20_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_8A1C	Miscellaneous Register (CCM_MISC_ROOT20_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8A20	Post Divider Register (CCM_POST20)	32	R/W	0000_0000h	5.1.7.18/473
3038_8A24	Post Divider Register (CCM_POST_ROOT20_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8A28	Post Divider Register (CCM_POST_ROOT20_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_8A2C	Post Divider Register (CCM_POST_ROOT20_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8A30	Pre Divider Register (CCM_PRE20)	32	R/W	1000_0000h	5.1.7.22/485
3038_8A34	Pre Divider Register (CCM_PRE_ROOT20_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8A38	Pre Divider Register (CCM_PRE_ROOT20_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_8A3C	Pre Divider Register (CCM_PRE_ROOT20_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8A70	Access Control Register (CCM_ACCESS_CTRL20)	32	R/W	0000_0000h	5.1.7.26/497
3038_8A74	Access Control Register (CCM_ACCESS_CTRL_ROOT20_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8A78	Access Control Register (CCM_ACCESS_CTRL_ROOT20_CLR)	32	R/W	0000_0000h	5.1.7.28/502

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8A7C	Access Control Register (CCM_ACCESS_CTRL_ROOT20_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8A80	Target Register (CCM_TARGET_ROOT21)	32	R/W	1000_0000h	5.1.7.10/461
3038_8A84	Target Register (CCM_TARGET_ROOT21_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8A88	Target Register (CCM_TARGET_ROOT21_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_8A8C	Target Register (CCM_TARGET_ROOT21_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8A90	Miscellaneous Register (CCM_MISC21)	32	R/W	0000_0000h	5.1.7.14/469
3038_8A94	Miscellaneous Register (CCM_MISC_ROOT21_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8A98	Miscellaneous Register (CCM_MISC_ROOT21_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_8A9C	Miscellaneous Register (CCM_MISC_ROOT21_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8AA0	Post Divider Register (CCM_POST21)	32	R/W	0000_0000h	5.1.7.18/473
3038_8AA4	Post Divider Register (CCM_POST_ROOT21_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8AA8	Post Divider Register (CCM_POST_ROOT21_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_8AAC	Post Divider Register (CCM_POST_ROOT21_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8AB0	Pre Divider Register (CCM_PRE21)	32	R/W	1000_0000h	5.1.7.22/485
3038_8AB4	Pre Divider Register (CCM_PRE_ROOT21_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8AB8	Pre Divider Register (CCM_PRE_ROOT21_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_8ABC	Pre Divider Register (CCM_PRE_ROOT21_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8AF0	Access Control Register (CCM_ACCESS_CTRL21)	32	R/W	0000_0000h	5.1.7.26/497
3038_8AF4	Access Control Register (CCM_ACCESS_CTRL_ROOT21_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8AF8	Access Control Register (CCM_ACCESS_CTRL_ROOT21_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_8AFC	Access Control Register (CCM_ACCESS_CTRL_ROOT21_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8B00	Target Register (CCM_TARGET_ROOT22)	32	R/W	1000_0000h	5.1.7.10/461
3038_8B04	Target Register (CCM_TARGET_ROOT22_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8B08	Target Register (CCM_TARGET_ROOT22_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_8B0C	Target Register (CCM_TARGET_ROOT22_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8B10	Miscellaneous Register (CCM_MISC22)	32	R/W	0000_0000h	5.1.7.14/469
3038_8B14	Miscellaneous Register (CCM_MISC_ROOT22_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8B18	Miscellaneous Register (CCM_MISC_ROOT22_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_8B1C	Miscellaneous Register (CCM_MISC_ROOT22_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8B20	Post Divider Register (CCM_POST22)	32	R/W	0000_0000h	5.1.7.18/473
3038_8B24	Post Divider Register (CCM_POST_ROOT22_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8B28	Post Divider Register (CCM_POST_ROOT22_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_8B2C	Post Divider Register (CCM_POST_ROOT22_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8B30	Pre Divider Register (CCM_PRE22)	32	R/W	1000_0000h	5.1.7.22/485
3038_8B34	Pre Divider Register (CCM_PRE_ROOT22_SET)	32	R/W	0000_0000h	5.1.7.23/488

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8B38	Pre Divider Register (CCM_PRE_ROOT22_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_8B3C	Pre Divider Register (CCM_PRE_ROOT22_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8B70	Access Control Register (CCM_ACCESS_CTRL22)	32	R/W	0000_0000h	5.1.7.26/497
3038_8B74	Access Control Register (CCM_ACCESS_CTRL_ROOT22_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8B78	Access Control Register (CCM_ACCESS_CTRL_ROOT22_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_8B7C	Access Control Register (CCM_ACCESS_CTRL_ROOT22_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8B80	Target Register (CCM_TARGET_ROOT23)	32	R/W	1000_0000h	5.1.7.10/461
3038_8B84	Target Register (CCM_TARGET_ROOT23_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8B88	Target Register (CCM_TARGET_ROOT23_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_8B8C	Target Register (CCM_TARGET_ROOT23_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8B90	Miscellaneous Register (CCM_MISC23)	32	R/W	0000_0000h	5.1.7.14/469
3038_8B94	Miscellaneous Register (CCM_MISC_ROOT23_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8B98	Miscellaneous Register (CCM_MISC_ROOT23_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_8B9C	Miscellaneous Register (CCM_MISC_ROOT23_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8BA0	Post Divider Register (CCM_POST23)	32	R/W	0000_0000h	5.1.7.18/473
3038_8BA4	Post Divider Register (CCM_POST_ROOT23_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8BA8	Post Divider Register (CCM_POST_ROOT23_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_8BAC	Post Divider Register (CCM_POST_ROOT23_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8BB0	Pre Divider Register (CCM_PRE23)	32	R/W	1000_0000h	5.1.7.22/485
3038_8BB4	Pre Divider Register (CCM_PRE_ROOT23_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8BB8	Pre Divider Register (CCM_PRE_ROOT23_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_8BBC	Pre Divider Register (CCM_PRE_ROOT23_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8BF0	Access Control Register (CCM_ACCESS_CTRL23)	32	R/W	0000_0000h	5.1.7.26/497
3038_8BF4	Access Control Register (CCM_ACCESS_CTRL_ROOT23_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8BF8	Access Control Register (CCM_ACCESS_CTRL_ROOT23_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_8BFC	Access Control Register (CCM_ACCESS_CTRL_ROOT23_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8C00	Target Register (CCM_TARGET_ROOT24)	32	R/W	1000_0000h	5.1.7.10/461
3038_8C04	Target Register (CCM_TARGET_ROOT24_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8C08	Target Register (CCM_TARGET_ROOT24_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_8C0C	Target Register (CCM_TARGET_ROOT24_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8C10	Miscellaneous Register (CCM_MISC24)	32	R/W	0000_0000h	5.1.7.14/469
3038_8C14	Miscellaneous Register (CCM_MISC_ROOT24_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8C18	Miscellaneous Register (CCM_MISC_ROOT24_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_8C1C	Miscellaneous Register (CCM_MISC_ROOT24_TOG)	32	R/W	0000_0000h	5.1.7.17/472

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8C20	Post Divider Register (CCM_POST24)	32	R/W	0000_0000h	5.1.7.18/473
3038_8C24	Post Divider Register (CCM_POST_ROOT24_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8C28	Post Divider Register (CCM_POST_ROOT24_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_8C2C	Post Divider Register (CCM_POST_ROOT24_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8C30	Pre Divider Register (CCM_PRE24)	32	R/W	1000_0000h	5.1.7.22/485
3038_8C34	Pre Divider Register (CCM_PRE_ROOT24_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8C38	Pre Divider Register (CCM_PRE_ROOT24_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_8C3C	Pre Divider Register (CCM_PRE_ROOT24_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8C70	Access Control Register (CCM_ACCESS_CTRL24)	32	R/W	0000_0000h	5.1.7.26/497
3038_8C74	Access Control Register (CCM_ACCESS_CTRL_ROOT24_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8C78	Access Control Register (CCM_ACCESS_CTRL_ROOT24_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_8C7C	Access Control Register (CCM_ACCESS_CTRL_ROOT24_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8C80	Target Register (CCM_TARGET_ROOT25)	32	R/W	1000_0000h	5.1.7.10/461
3038_8C84	Target Register (CCM_TARGET_ROOT25_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8C88	Target Register (CCM_TARGET_ROOT25_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_8C8C	Target Register (CCM_TARGET_ROOT25_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8C90	Miscellaneous Register (CCM_MISC25)	32	R/W	0000_0000h	5.1.7.14/469
3038_8C94	Miscellaneous Register (CCM_MISC_ROOT25_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8C98	Miscellaneous Register (CCM_MISC_ROOT25_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_8C9C	Miscellaneous Register (CCM_MISC_ROOT25_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8CA0	Post Divider Register (CCM_POST25)	32	R/W	0000_0000h	5.1.7.18/473
3038_8CA4	Post Divider Register (CCM_POST_ROOT25_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8CA8	Post Divider Register (CCM_POST_ROOT25_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_8CAC	Post Divider Register (CCM_POST_ROOT25_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8CB0	Pre Divider Register (CCM_PRE25)	32	R/W	1000_0000h	5.1.7.22/485
3038_8CB4	Pre Divider Register (CCM_PRE_ROOT25_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8CB8	Pre Divider Register (CCM_PRE_ROOT25_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_8CBC	Pre Divider Register (CCM_PRE_ROOT25_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8CF0	Access Control Register (CCM_ACCESS_CTRL25)	32	R/W	0000_0000h	5.1.7.26/497
3038_8CF4	Access Control Register (CCM_ACCESS_CTRL_ROOT25_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8CF8	Access Control Register (CCM_ACCESS_CTRL_ROOT25_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_8CFC	Access Control Register (CCM_ACCESS_CTRL_ROOT25_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8D00	Target Register (CCM_TARGET_ROOT26)	32	R/W	1000_0000h	5.1.7.10/461
3038_8D04	Target Register (CCM_TARGET_ROOT26_SET)	32	R/W	0000_0000h	5.1.7.11/463

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8D08	Target Register (CCM_TARGET_ROOT26_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_8D0C	Target Register (CCM_TARGET_ROOT26_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8D10	Miscellaneous Register (CCM_MISC26)	32	R/W	0000_0000h	5.1.7.14/469
3038_8D14	Miscellaneous Register (CCM_MISC_ROOT26_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8D18	Miscellaneous Register (CCM_MISC_ROOT26_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_8D1C	Miscellaneous Register (CCM_MISC_ROOT26_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8D20	Post Divider Register (CCM_POST26)	32	R/W	0000_0000h	5.1.7.18/473
3038_8D24	Post Divider Register (CCM_POST_ROOT26_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8D28	Post Divider Register (CCM_POST_ROOT26_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_8D2C	Post Divider Register (CCM_POST_ROOT26_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8D30	Pre Divider Register (CCM_PRE26)	32	R/W	1000_0000h	5.1.7.22/485
3038_8D34	Pre Divider Register (CCM_PRE_ROOT26_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8D38	Pre Divider Register (CCM_PRE_ROOT26_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_8D3C	Pre Divider Register (CCM_PRE_ROOT26_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8D70	Access Control Register (CCM_ACCESS_CTRL26)	32	R/W	0000_0000h	5.1.7.26/497
3038_8D74	Access Control Register (CCM_ACCESS_CTRL_ROOT26_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8D78	Access Control Register (CCM_ACCESS_CTRL_ROOT26_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_8D7C	Access Control Register (CCM_ACCESS_CTRL_ROOT26_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8D80	Target Register (CCM_TARGET_ROOT27)	32	R/W	1000_0000h	5.1.7.10/461
3038_8D84	Target Register (CCM_TARGET_ROOT27_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8D88	Target Register (CCM_TARGET_ROOT27_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_8D8C	Target Register (CCM_TARGET_ROOT27_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8D90	Miscellaneous Register (CCM_MISC27)	32	R/W	0000_0000h	5.1.7.14/469
3038_8D94	Miscellaneous Register (CCM_MISC_ROOT27_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8D98	Miscellaneous Register (CCM_MISC_ROOT27_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_8D9C	Miscellaneous Register (CCM_MISC_ROOT27_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8DA0	Post Divider Register (CCM_POST27)	32	R/W	0000_0000h	5.1.7.18/473
3038_8DA4	Post Divider Register (CCM_POST_ROOT27_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8DA8	Post Divider Register (CCM_POST_ROOT27_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_8DAC	Post Divider Register (CCM_POST_ROOT27_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8DB0	Pre Divider Register (CCM_PRE27)	32	R/W	1000_0000h	5.1.7.22/485
3038_8DB4	Pre Divider Register (CCM_PRE_ROOT27_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8DB8	Pre Divider Register (CCM_PRE_ROOT27_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_8DBC	Pre Divider Register (CCM_PRE_ROOT27_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8DF0	Access Control Register (CCM_ACCESS_CTRL27)	32	R/W	0000_0000h	5.1.7.26/497

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8DF4	Access Control Register (CCM_ACCESS_CTRL_ROOT27_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8DF8	Access Control Register (CCM_ACCESS_CTRL_ROOT27_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_8DFC	Access Control Register (CCM_ACCESS_CTRL_ROOT27_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8E00	Target Register (CCM_TARGET_ROOT28)	32	R/W	1000_0000h	5.1.7.10/461
3038_8E04	Target Register (CCM_TARGET_ROOT28_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8E08	Target Register (CCM_TARGET_ROOT28_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_8E0C	Target Register (CCM_TARGET_ROOT28_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8E10	Miscellaneous Register (CCM_MISC28)	32	R/W	0000_0000h	5.1.7.14/469
3038_8E14	Miscellaneous Register (CCM_MISC_ROOT28_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8E18	Miscellaneous Register (CCM_MISC_ROOT28_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_8E1C	Miscellaneous Register (CCM_MISC_ROOT28_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8E20	Post Divider Register (CCM_POST28)	32	R/W	0000_0000h	5.1.7.18/473
3038_8E24	Post Divider Register (CCM_POST_ROOT28_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8E28	Post Divider Register (CCM_POST_ROOT28_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_8E2C	Post Divider Register (CCM_POST_ROOT28_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8E30	Pre Divider Register (CCM_PRE28)	32	R/W	1000_0000h	5.1.7.22/485
3038_8E34	Pre Divider Register (CCM_PRE_ROOT28_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8E38	Pre Divider Register (CCM_PRE_ROOT28_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_8E3C	Pre Divider Register (CCM_PRE_ROOT28_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8E70	Access Control Register (CCM_ACCESS_CTRL28)	32	R/W	0000_0000h	5.1.7.26/497
3038_8E74	Access Control Register (CCM_ACCESS_CTRL_ROOT28_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8E78	Access Control Register (CCM_ACCESS_CTRL_ROOT28_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_8E7C	Access Control Register (CCM_ACCESS_CTRL_ROOT28_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8E80	Target Register (CCM_TARGET_ROOT29)	32	R/W	1000_0000h	5.1.7.10/461
3038_8E84	Target Register (CCM_TARGET_ROOT29_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8E88	Target Register (CCM_TARGET_ROOT29_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_8E8C	Target Register (CCM_TARGET_ROOT29_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8E90	Miscellaneous Register (CCM_MISC29)	32	R/W	0000_0000h	5.1.7.14/469
3038_8E94	Miscellaneous Register (CCM_MISC_ROOT29_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8E98	Miscellaneous Register (CCM_MISC_ROOT29_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_8E9C	Miscellaneous Register (CCM_MISC_ROOT29_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8EA0	Post Divider Register (CCM_POST29)	32	R/W	0000_0000h	5.1.7.18/473
3038_8EA4	Post Divider Register (CCM_POST_ROOT29_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8EA8	Post Divider Register (CCM_POST_ROOT29_CLR)	32	R/W	0000_0000h	5.1.7.20/479

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8EAC	Post Divider Register (CCM_POST_ROOT29_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8EB0	Pre Divider Register (CCM_PRE29)	32	R/W	1000_0000h	5.1.7.22/485
3038_8EB4	Pre Divider Register (CCM_PRE_ROOT29_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8EB8	Pre Divider Register (CCM_PRE_ROOT29_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_8EBC	Pre Divider Register (CCM_PRE_ROOT29_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8EF0	Access Control Register (CCM_ACCESS_CTRL29)	32	R/W	0000_0000h	5.1.7.26/497
3038_8EF4	Access Control Register (CCM_ACCESS_CTRL_ROOT29_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8EF8	Access Control Register (CCM_ACCESS_CTRL_ROOT29_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_8EFC	Access Control Register (CCM_ACCESS_CTRL_ROOT29_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8F00	Target Register (CCM_TARGET_ROOT30)	32	R/W	1000_0000h	5.1.7.10/461
3038_8F04	Target Register (CCM_TARGET_ROOT30_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8F08	Target Register (CCM_TARGET_ROOT30_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_8F0C	Target Register (CCM_TARGET_ROOT30_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8F10	Miscellaneous Register (CCM_MISC30)	32	R/W	0000_0000h	5.1.7.14/469
3038_8F14	Miscellaneous Register (CCM_MISC_ROOT30_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8F18	Miscellaneous Register (CCM_MISC_ROOT30_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_8F1C	Miscellaneous Register (CCM_MISC_ROOT30_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8F20	Post Divider Register (CCM_POST30)	32	R/W	0000_0000h	5.1.7.18/473
3038_8F24	Post Divider Register (CCM_POST_ROOT30_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8F28	Post Divider Register (CCM_POST_ROOT30_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_8F2C	Post Divider Register (CCM_POST_ROOT30_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8F30	Pre Divider Register (CCM_PRE30)	32	R/W	1000_0000h	5.1.7.22/485
3038_8F34	Pre Divider Register (CCM_PRE_ROOT30_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8F38	Pre Divider Register (CCM_PRE_ROOT30_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_8F3C	Pre Divider Register (CCM_PRE_ROOT30_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8F70	Access Control Register (CCM_ACCESS_CTRL30)	32	R/W	0000_0000h	5.1.7.26/497
3038_8F74	Access Control Register (CCM_ACCESS_CTRL_ROOT30_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8F78	Access Control Register (CCM_ACCESS_CTRL_ROOT30_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_8F7C	Access Control Register (CCM_ACCESS_CTRL_ROOT30_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_8F80	Target Register (CCM_TARGET_ROOT31)	32	R/W	1000_0000h	5.1.7.10/461
3038_8F84	Target Register (CCM_TARGET_ROOT31_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_8F88	Target Register (CCM_TARGET_ROOT31_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_8F8C	Target Register (CCM_TARGET_ROOT31_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_8F90	Miscellaneous Register (CCM_MISC31)	32	R/W	0000_0000h	5.1.7.14/469

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8F94	Miscellaneous Register (CCM_MISC_ROOT31_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_8F98	Miscellaneous Register (CCM_MISC_ROOT31_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_8F9C	Miscellaneous Register (CCM_MISC_ROOT31_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_8FA0	Post Divider Register (CCM_POST31)	32	R/W	0000_0000h	5.1.7.18/473
3038_8FA4	Post Divider Register (CCM_POST_ROOT31_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_8FA8	Post Divider Register (CCM_POST_ROOT31_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_8FAC	Post Divider Register (CCM_POST_ROOT31_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_8FB0	Pre Divider Register (CCM_PRE31)	32	R/W	1000_0000h	5.1.7.22/485
3038_8FB4	Pre Divider Register (CCM_PRE_ROOT31_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_8FB8	Pre Divider Register (CCM_PRE_ROOT31_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_8FBC	Pre Divider Register (CCM_PRE_ROOT31_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_8FF0	Access Control Register (CCM_ACCESS_CTRL31)	32	R/W	0000_0000h	5.1.7.26/497
3038_8FF4	Access Control Register (CCM_ACCESS_CTRL_ROOT31_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_8FF8	Access Control Register (CCM_ACCESS_CTRL_ROOT31_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_8FFC	Access Control Register (CCM_ACCESS_CTRL_ROOT31_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9000	Target Register (CCM_TARGET_ROOT32)	32	R/W	1000_0000h	5.1.7.10/461
3038_9004	Target Register (CCM_TARGET_ROOT32_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9008	Target Register (CCM_TARGET_ROOT32_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_900C	Target Register (CCM_TARGET_ROOT32_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9010	Miscellaneous Register (CCM_MISC32)	32	R/W	0000_0000h	5.1.7.14/469
3038_9014	Miscellaneous Register (CCM_MISC_ROOT32_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9018	Miscellaneous Register (CCM_MISC_ROOT32_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_901C	Miscellaneous Register (CCM_MISC_ROOT32_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9020	Post Divider Register (CCM_POST32)	32	R/W	0000_0000h	5.1.7.18/473
3038_9024	Post Divider Register (CCM_POST_ROOT32_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9028	Post Divider Register (CCM_POST_ROOT32_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_902C	Post Divider Register (CCM_POST_ROOT32_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9030	Pre Divider Register (CCM_PRE32)	32	R/W	1000_0000h	5.1.7.22/485
3038_9034	Pre Divider Register (CCM_PRE_ROOT32_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9038	Pre Divider Register (CCM_PRE_ROOT32_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_903C	Pre Divider Register (CCM_PRE_ROOT32_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9070	Access Control Register (CCM_ACCESS_CTRL32)	32	R/W	0000_0000h	5.1.7.26/497
3038_9074	Access Control Register (CCM_ACCESS_CTRL_ROOT32_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9078	Access Control Register (CCM_ACCESS_CTRL_ROOT32_CLR)	32	R/W	0000_0000h	5.1.7.28/502

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_907C	Access Control Register (CCM_ACCESS_CTRL_ROOT32_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9080	Target Register (CCM_TARGET_ROOT33)	32	R/W	1000_0000h	5.1.7.10/461
3038_9084	Target Register (CCM_TARGET_ROOT33_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9088	Target Register (CCM_TARGET_ROOT33_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_908C	Target Register (CCM_TARGET_ROOT33_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9090	Miscellaneous Register (CCM_MISC33)	32	R/W	0000_0000h	5.1.7.14/469
3038_9094	Miscellaneous Register (CCM_MISC_ROOT33_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9098	Miscellaneous Register (CCM_MISC_ROOT33_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_909C	Miscellaneous Register (CCM_MISC_ROOT33_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_90A0	Post Divider Register (CCM_POST33)	32	R/W	0000_0000h	5.1.7.18/473
3038_90A4	Post Divider Register (CCM_POST_ROOT33_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_90A8	Post Divider Register (CCM_POST_ROOT33_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_90AC	Post Divider Register (CCM_POST_ROOT33_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_90B0	Pre Divider Register (CCM_PRE33)	32	R/W	1000_0000h	5.1.7.22/485
3038_90B4	Pre Divider Register (CCM_PRE_ROOT33_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_90B8	Pre Divider Register (CCM_PRE_ROOT33_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_90BC	Pre Divider Register (CCM_PRE_ROOT33_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_90F0	Access Control Register (CCM_ACCESS_CTRL33)	32	R/W	0000_0000h	5.1.7.26/497
3038_90F4	Access Control Register (CCM_ACCESS_CTRL_ROOT33_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_90F8	Access Control Register (CCM_ACCESS_CTRL_ROOT33_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_90FC	Access Control Register (CCM_ACCESS_CTRL_ROOT33_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9100	Target Register (CCM_TARGET_ROOT34)	32	R/W	1000_0000h	5.1.7.10/461
3038_9104	Target Register (CCM_TARGET_ROOT34_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9108	Target Register (CCM_TARGET_ROOT34_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_910C	Target Register (CCM_TARGET_ROOT34_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9110	Miscellaneous Register (CCM_MISC34)	32	R/W	0000_0000h	5.1.7.14/469
3038_9114	Miscellaneous Register (CCM_MISC_ROOT34_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9118	Miscellaneous Register (CCM_MISC_ROOT34_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_911C	Miscellaneous Register (CCM_MISC_ROOT34_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9120	Post Divider Register (CCM_POST34)	32	R/W	0000_0000h	5.1.7.18/473
3038_9124	Post Divider Register (CCM_POST_ROOT34_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9128	Post Divider Register (CCM_POST_ROOT34_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_912C	Post Divider Register (CCM_POST_ROOT34_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9130	Pre Divider Register (CCM_PRE34)	32	R/W	1000_0000h	5.1.7.22/485
3038_9134	Pre Divider Register (CCM_PRE_ROOT34_SET)	32	R/W	0000_0000h	5.1.7.23/488

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9138	Pre Divider Register (CCM_PRE_ROOT34_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_913C	Pre Divider Register (CCM_PRE_ROOT34_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9170	Access Control Register (CCM_ACCESS_CTRL34)	32	R/W	0000_0000h	5.1.7.26/497
3038_9174	Access Control Register (CCM_ACCESS_CTRL_ROOT34_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9178	Access Control Register (CCM_ACCESS_CTRL_ROOT34_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_917C	Access Control Register (CCM_ACCESS_CTRL_ROOT34_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9180	Target Register (CCM_TARGET_ROOT35)	32	R/W	1000_0000h	5.1.7.10/461
3038_9184	Target Register (CCM_TARGET_ROOT35_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9188	Target Register (CCM_TARGET_ROOT35_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_918C	Target Register (CCM_TARGET_ROOT35_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9190	Miscellaneous Register (CCM_MISC35)	32	R/W	0000_0000h	5.1.7.14/469
3038_9194	Miscellaneous Register (CCM_MISC_ROOT35_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9198	Miscellaneous Register (CCM_MISC_ROOT35_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_919C	Miscellaneous Register (CCM_MISC_ROOT35_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_91A0	Post Divider Register (CCM_POST35)	32	R/W	0000_0000h	5.1.7.18/473
3038_91A4	Post Divider Register (CCM_POST_ROOT35_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_91A8	Post Divider Register (CCM_POST_ROOT35_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_91AC	Post Divider Register (CCM_POST_ROOT35_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_91B0	Pre Divider Register (CCM_PRE35)	32	R/W	1000_0000h	5.1.7.22/485
3038_91B4	Pre Divider Register (CCM_PRE_ROOT35_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_91B8	Pre Divider Register (CCM_PRE_ROOT35_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_91BC	Pre Divider Register (CCM_PRE_ROOT35_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_91F0	Access Control Register (CCM_ACCESS_CTRL35)	32	R/W	0000_0000h	5.1.7.26/497
3038_91F4	Access Control Register (CCM_ACCESS_CTRL_ROOT35_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_91F8	Access Control Register (CCM_ACCESS_CTRL_ROOT35_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_91FC	Access Control Register (CCM_ACCESS_CTRL_ROOT35_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9200	Target Register (CCM_TARGET_ROOT36)	32	R/W	1000_0000h	5.1.7.10/461
3038_9204	Target Register (CCM_TARGET_ROOT36_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9208	Target Register (CCM_TARGET_ROOT36_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_920C	Target Register (CCM_TARGET_ROOT36_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9210	Miscellaneous Register (CCM_MISC36)	32	R/W	0000_0000h	5.1.7.14/469
3038_9214	Miscellaneous Register (CCM_MISC_ROOT36_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9218	Miscellaneous Register (CCM_MISC_ROOT36_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_921C	Miscellaneous Register (CCM_MISC_ROOT36_TOG)	32	R/W	0000_0000h	5.1.7.17/472

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9220	Post Divider Register (CCM_POST36)	32	R/W	0000_0000h	5.1.7.18/473
3038_9224	Post Divider Register (CCM_POST_ROOT36_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9228	Post Divider Register (CCM_POST_ROOT36_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_922C	Post Divider Register (CCM_POST_ROOT36_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9230	Pre Divider Register (CCM_PRE36)	32	R/W	1000_0000h	5.1.7.22/485
3038_9234	Pre Divider Register (CCM_PRE_ROOT36_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9238	Pre Divider Register (CCM_PRE_ROOT36_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_923C	Pre Divider Register (CCM_PRE_ROOT36_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9270	Access Control Register (CCM_ACCESS_CTRL36)	32	R/W	0000_0000h	5.1.7.26/497
3038_9274	Access Control Register (CCM_ACCESS_CTRL_ROOT36_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9278	Access Control Register (CCM_ACCESS_CTRL_ROOT36_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_927C	Access Control Register (CCM_ACCESS_CTRL_ROOT36_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9280	Target Register (CCM_TARGET_ROOT37)	32	R/W	1000_0000h	5.1.7.10/461
3038_9284	Target Register (CCM_TARGET_ROOT37_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9288	Target Register (CCM_TARGET_ROOT37_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_928C	Target Register (CCM_TARGET_ROOT37_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9290	Miscellaneous Register (CCM_MISC37)	32	R/W	0000_0000h	5.1.7.14/469
3038_9294	Miscellaneous Register (CCM_MISC_ROOT37_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9298	Miscellaneous Register (CCM_MISC_ROOT37_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_929C	Miscellaneous Register (CCM_MISC_ROOT37_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_92A0	Post Divider Register (CCM_POST37)	32	R/W	0000_0000h	5.1.7.18/473
3038_92A4	Post Divider Register (CCM_POST_ROOT37_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_92A8	Post Divider Register (CCM_POST_ROOT37_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_92AC	Post Divider Register (CCM_POST_ROOT37_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_92B0	Pre Divider Register (CCM_PRE37)	32	R/W	1000_0000h	5.1.7.22/485
3038_92B4	Pre Divider Register (CCM_PRE_ROOT37_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_92B8	Pre Divider Register (CCM_PRE_ROOT37_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_92BC	Pre Divider Register (CCM_PRE_ROOT37_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_92F0	Access Control Register (CCM_ACCESS_CTRL37)	32	R/W	0000_0000h	5.1.7.26/497
3038_92F4	Access Control Register (CCM_ACCESS_CTRL_ROOT37_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_92F8	Access Control Register (CCM_ACCESS_CTRL_ROOT37_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_92FC	Access Control Register (CCM_ACCESS_CTRL_ROOT37_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9300	Target Register (CCM_TARGET_ROOT38)	32	R/W	1000_0000h	5.1.7.10/461
3038_9304	Target Register (CCM_TARGET_ROOT38_SET)	32	R/W	0000_0000h	5.1.7.11/463

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9308	Target Register (CCM_TARGET_ROOT38_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_930C	Target Register (CCM_TARGET_ROOT38_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9310	Miscellaneous Register (CCM_MISC38)	32	R/W	0000_0000h	5.1.7.14/469
3038_9314	Miscellaneous Register (CCM_MISC_ROOT38_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9318	Miscellaneous Register (CCM_MISC_ROOT38_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_931C	Miscellaneous Register (CCM_MISC_ROOT38_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9320	Post Divider Register (CCM_POST38)	32	R/W	0000_0000h	5.1.7.18/473
3038_9324	Post Divider Register (CCM_POST_ROOT38_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9328	Post Divider Register (CCM_POST_ROOT38_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_932C	Post Divider Register (CCM_POST_ROOT38_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9330	Pre Divider Register (CCM_PRE38)	32	R/W	1000_0000h	5.1.7.22/485
3038_9334	Pre Divider Register (CCM_PRE_ROOT38_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9338	Pre Divider Register (CCM_PRE_ROOT38_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_933C	Pre Divider Register (CCM_PRE_ROOT38_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9370	Access Control Register (CCM_ACCESS_CTRL38)	32	R/W	0000_0000h	5.1.7.26/497
3038_9374	Access Control Register (CCM_ACCESS_CTRL_ROOT38_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9378	Access Control Register (CCM_ACCESS_CTRL_ROOT38_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_937C	Access Control Register (CCM_ACCESS_CTRL_ROOT38_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9380	Target Register (CCM_TARGET_ROOT39)	32	R/W	1000_0000h	5.1.7.10/461
3038_9384	Target Register (CCM_TARGET_ROOT39_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9388	Target Register (CCM_TARGET_ROOT39_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_938C	Target Register (CCM_TARGET_ROOT39_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9390	Miscellaneous Register (CCM_MISC39)	32	R/W	0000_0000h	5.1.7.14/469
3038_9394	Miscellaneous Register (CCM_MISC_ROOT39_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9398	Miscellaneous Register (CCM_MISC_ROOT39_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_939C	Miscellaneous Register (CCM_MISC_ROOT39_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_93A0	Post Divider Register (CCM_POST39)	32	R/W	0000_0000h	5.1.7.18/473
3038_93A4	Post Divider Register (CCM_POST_ROOT39_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_93A8	Post Divider Register (CCM_POST_ROOT39_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_93AC	Post Divider Register (CCM_POST_ROOT39_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_93B0	Pre Divider Register (CCM_PRE39)	32	R/W	1000_0000h	5.1.7.22/485
3038_93B4	Pre Divider Register (CCM_PRE_ROOT39_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_93B8	Pre Divider Register (CCM_PRE_ROOT39_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_93BC	Pre Divider Register (CCM_PRE_ROOT39_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_93F0	Access Control Register (CCM_ACCESS_CTRL39)	32	R/W	0000_0000h	5.1.7.26/497

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_93F4	Access Control Register (CCM_ACCESS_CTRL_ROOT39_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_93F8	Access Control Register (CCM_ACCESS_CTRL_ROOT39_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_93FC	Access Control Register (CCM_ACCESS_CTRL_ROOT39_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9400	Target Register (CCM_TARGET_ROOT40)	32	R/W	1000_0000h	5.1.7.10/461
3038_9404	Target Register (CCM_TARGET_ROOT40_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9408	Target Register (CCM_TARGET_ROOT40_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_940C	Target Register (CCM_TARGET_ROOT40_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9410	Miscellaneous Register (CCM_MISC40)	32	R/W	0000_0000h	5.1.7.14/469
3038_9414	Miscellaneous Register (CCM_MISC_ROOT40_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9418	Miscellaneous Register (CCM_MISC_ROOT40_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_941C	Miscellaneous Register (CCM_MISC_ROOT40_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9420	Post Divider Register (CCM_POST40)	32	R/W	0000_0000h	5.1.7.18/473
3038_9424	Post Divider Register (CCM_POST_ROOT40_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9428	Post Divider Register (CCM_POST_ROOT40_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_942C	Post Divider Register (CCM_POST_ROOT40_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9430	Pre Divider Register (CCM_PRE40)	32	R/W	1000_0000h	5.1.7.22/485
3038_9434	Pre Divider Register (CCM_PRE_ROOT40_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9438	Pre Divider Register (CCM_PRE_ROOT40_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_943C	Pre Divider Register (CCM_PRE_ROOT40_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9470	Access Control Register (CCM_ACCESS_CTRL40)	32	R/W	0000_0000h	5.1.7.26/497
3038_9474	Access Control Register (CCM_ACCESS_CTRL_ROOT40_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9478	Access Control Register (CCM_ACCESS_CTRL_ROOT40_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_947C	Access Control Register (CCM_ACCESS_CTRL_ROOT40_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9480	Target Register (CCM_TARGET_ROOT41)	32	R/W	1000_0000h	5.1.7.10/461
3038_9484	Target Register (CCM_TARGET_ROOT41_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9488	Target Register (CCM_TARGET_ROOT41_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_948C	Target Register (CCM_TARGET_ROOT41_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9490	Miscellaneous Register (CCM_MISC41)	32	R/W	0000_0000h	5.1.7.14/469
3038_9494	Miscellaneous Register (CCM_MISC_ROOT41_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9498	Miscellaneous Register (CCM_MISC_ROOT41_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_949C	Miscellaneous Register (CCM_MISC_ROOT41_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_94A0	Post Divider Register (CCM_POST41)	32	R/W	0000_0000h	5.1.7.18/473
3038_94A4	Post Divider Register (CCM_POST_ROOT41_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_94A8	Post Divider Register (CCM_POST_ROOT41_CLR)	32	R/W	0000_0000h	5.1.7.20/479

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_94AC	Post Divider Register (CCM_POST_ROOT41_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_94B0	Pre Divider Register (CCM_PRE41)	32	R/W	1000_0000h	5.1.7.22/485
3038_94B4	Pre Divider Register (CCM_PRE_ROOT41_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_94B8	Pre Divider Register (CCM_PRE_ROOT41_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_94BC	Pre Divider Register (CCM_PRE_ROOT41_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_94F0	Access Control Register (CCM_ACCESS_CTRL41)	32	R/W	0000_0000h	5.1.7.26/497
3038_94F4	Access Control Register (CCM_ACCESS_CTRL_ROOT41_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_94F8	Access Control Register (CCM_ACCESS_CTRL_ROOT41_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_94FC	Access Control Register (CCM_ACCESS_CTRL_ROOT41_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9500	Target Register (CCM_TARGET_ROOT42)	32	R/W	1000_0000h	5.1.7.10/461
3038_9504	Target Register (CCM_TARGET_ROOT42_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9508	Target Register (CCM_TARGET_ROOT42_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_950C	Target Register (CCM_TARGET_ROOT42_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9510	Miscellaneous Register (CCM_MISC42)	32	R/W	0000_0000h	5.1.7.14/469
3038_9514	Miscellaneous Register (CCM_MISC_ROOT42_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9518	Miscellaneous Register (CCM_MISC_ROOT42_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_951C	Miscellaneous Register (CCM_MISC_ROOT42_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9520	Post Divider Register (CCM_POST42)	32	R/W	0000_0000h	5.1.7.18/473
3038_9524	Post Divider Register (CCM_POST_ROOT42_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9528	Post Divider Register (CCM_POST_ROOT42_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_952C	Post Divider Register (CCM_POST_ROOT42_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9530	Pre Divider Register (CCM_PRE42)	32	R/W	1000_0000h	5.1.7.22/485
3038_9534	Pre Divider Register (CCM_PRE_ROOT42_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9538	Pre Divider Register (CCM_PRE_ROOT42_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_953C	Pre Divider Register (CCM_PRE_ROOT42_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9570	Access Control Register (CCM_ACCESS_CTRL42)	32	R/W	0000_0000h	5.1.7.26/497
3038_9574	Access Control Register (CCM_ACCESS_CTRL_ROOT42_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9578	Access Control Register (CCM_ACCESS_CTRL_ROOT42_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_957C	Access Control Register (CCM_ACCESS_CTRL_ROOT42_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9580	Target Register (CCM_TARGET_ROOT43)	32	R/W	1000_0000h	5.1.7.10/461
3038_9584	Target Register (CCM_TARGET_ROOT43_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9588	Target Register (CCM_TARGET_ROOT43_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_958C	Target Register (CCM_TARGET_ROOT43_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9590	Miscellaneous Register (CCM_MISC43)	32	R/W	0000_0000h	5.1.7.14/469

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9594	Miscellaneous Register (CCM_MISC_ROOT43_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9598	Miscellaneous Register (CCM_MISC_ROOT43_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_959C	Miscellaneous Register (CCM_MISC_ROOT43_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_95A0	Post Divider Register (CCM_POST43)	32	R/W	0000_0000h	5.1.7.18/473
3038_95A4	Post Divider Register (CCM_POST_ROOT43_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_95A8	Post Divider Register (CCM_POST_ROOT43_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_95AC	Post Divider Register (CCM_POST_ROOT43_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_95B0	Pre Divider Register (CCM_PRE43)	32	R/W	1000_0000h	5.1.7.22/485
3038_95B4	Pre Divider Register (CCM_PRE_ROOT43_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_95B8	Pre Divider Register (CCM_PRE_ROOT43_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_95BC	Pre Divider Register (CCM_PRE_ROOT43_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_95F0	Access Control Register (CCM_ACCESS_CTRL43)	32	R/W	0000_0000h	5.1.7.26/497
3038_95F4	Access Control Register (CCM_ACCESS_CTRL_ROOT43_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_95F8	Access Control Register (CCM_ACCESS_CTRL_ROOT43_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_95FC	Access Control Register (CCM_ACCESS_CTRL_ROOT43_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9600	Target Register (CCM_TARGET_ROOT44)	32	R/W	1000_0000h	5.1.7.10/461
3038_9604	Target Register (CCM_TARGET_ROOT44_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9608	Target Register (CCM_TARGET_ROOT44_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_960C	Target Register (CCM_TARGET_ROOT44_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9610	Miscellaneous Register (CCM_MISC44)	32	R/W	0000_0000h	5.1.7.14/469
3038_9614	Miscellaneous Register (CCM_MISC_ROOT44_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9618	Miscellaneous Register (CCM_MISC_ROOT44_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_961C	Miscellaneous Register (CCM_MISC_ROOT44_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9620	Post Divider Register (CCM_POST44)	32	R/W	0000_0000h	5.1.7.18/473
3038_9624	Post Divider Register (CCM_POST_ROOT44_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9628	Post Divider Register (CCM_POST_ROOT44_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_962C	Post Divider Register (CCM_POST_ROOT44_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9630	Pre Divider Register (CCM_PRE44)	32	R/W	1000_0000h	5.1.7.22/485
3038_9634	Pre Divider Register (CCM_PRE_ROOT44_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9638	Pre Divider Register (CCM_PRE_ROOT44_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_963C	Pre Divider Register (CCM_PRE_ROOT44_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9670	Access Control Register (CCM_ACCESS_CTRL44)	32	R/W	0000_0000h	5.1.7.26/497
3038_9674	Access Control Register (CCM_ACCESS_CTRL_ROOT44_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9678	Access Control Register (CCM_ACCESS_CTRL_ROOT44_CLR)	32	R/W	0000_0000h	5.1.7.28/502

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_967C	Access Control Register (CCM_ACCESS_CTRL_ROOT44_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9680	Target Register (CCM_TARGET_ROOT45)	32	R/W	1000_0000h	5.1.7.10/461
3038_9684	Target Register (CCM_TARGET_ROOT45_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9688	Target Register (CCM_TARGET_ROOT45_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_968C	Target Register (CCM_TARGET_ROOT45_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9690	Miscellaneous Register (CCM_MISC45)	32	R/W	0000_0000h	5.1.7.14/469
3038_9694	Miscellaneous Register (CCM_MISC_ROOT45_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9698	Miscellaneous Register (CCM_MISC_ROOT45_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_969C	Miscellaneous Register (CCM_MISC_ROOT45_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_96A0	Post Divider Register (CCM_POST45)	32	R/W	0000_0000h	5.1.7.18/473
3038_96A4	Post Divider Register (CCM_POST_ROOT45_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_96A8	Post Divider Register (CCM_POST_ROOT45_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_96AC	Post Divider Register (CCM_POST_ROOT45_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_96B0	Pre Divider Register (CCM_PRE45)	32	R/W	1000_0000h	5.1.7.22/485
3038_96B4	Pre Divider Register (CCM_PRE_ROOT45_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_96B8	Pre Divider Register (CCM_PRE_ROOT45_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_96BC	Pre Divider Register (CCM_PRE_ROOT45_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_96F0	Access Control Register (CCM_ACCESS_CTRL45)	32	R/W	0000_0000h	5.1.7.26/497
3038_96F4	Access Control Register (CCM_ACCESS_CTRL_ROOT45_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_96F8	Access Control Register (CCM_ACCESS_CTRL_ROOT45_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_96FC	Access Control Register (CCM_ACCESS_CTRL_ROOT45_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9700	Target Register (CCM_TARGET_ROOT46)	32	R/W	1000_0000h	5.1.7.10/461
3038_9704	Target Register (CCM_TARGET_ROOT46_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9708	Target Register (CCM_TARGET_ROOT46_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_970C	Target Register (CCM_TARGET_ROOT46_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9710	Miscellaneous Register (CCM_MISC46)	32	R/W	0000_0000h	5.1.7.14/469
3038_9714	Miscellaneous Register (CCM_MISC_ROOT46_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9718	Miscellaneous Register (CCM_MISC_ROOT46_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_971C	Miscellaneous Register (CCM_MISC_ROOT46_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9720	Post Divider Register (CCM_POST46)	32	R/W	0000_0000h	5.1.7.18/473
3038_9724	Post Divider Register (CCM_POST_ROOT46_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9728	Post Divider Register (CCM_POST_ROOT46_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_972C	Post Divider Register (CCM_POST_ROOT46_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9730	Pre Divider Register (CCM_PRE46)	32	R/W	1000_0000h	5.1.7.22/485
3038_9734	Pre Divider Register (CCM_PRE_ROOT46_SET)	32	R/W	0000_0000h	5.1.7.23/488

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9738	Pre Divider Register (CCM_PRE_ROOT46_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_973C	Pre Divider Register (CCM_PRE_ROOT46_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9770	Access Control Register (CCM_ACCESS_CTRL46)	32	R/W	0000_0000h	5.1.7.26/497
3038_9774	Access Control Register (CCM_ACCESS_CTRL_ROOT46_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9778	Access Control Register (CCM_ACCESS_CTRL_ROOT46_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_977C	Access Control Register (CCM_ACCESS_CTRL_ROOT46_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9780	Target Register (CCM_TARGET_ROOT47)	32	R/W	1000_0000h	5.1.7.10/461
3038_9784	Target Register (CCM_TARGET_ROOT47_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9788	Target Register (CCM_TARGET_ROOT47_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_978C	Target Register (CCM_TARGET_ROOT47_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9790	Miscellaneous Register (CCM_MISC47)	32	R/W	0000_0000h	5.1.7.14/469
3038_9794	Miscellaneous Register (CCM_MISC_ROOT47_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9798	Miscellaneous Register (CCM_MISC_ROOT47_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_979C	Miscellaneous Register (CCM_MISC_ROOT47_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_97A0	Post Divider Register (CCM_POST47)	32	R/W	0000_0000h	5.1.7.18/473
3038_97A4	Post Divider Register (CCM_POST_ROOT47_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_97A8	Post Divider Register (CCM_POST_ROOT47_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_97AC	Post Divider Register (CCM_POST_ROOT47_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_97B0	Pre Divider Register (CCM_PRE47)	32	R/W	1000_0000h	5.1.7.22/485
3038_97B4	Pre Divider Register (CCM_PRE_ROOT47_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_97B8	Pre Divider Register (CCM_PRE_ROOT47_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_97BC	Pre Divider Register (CCM_PRE_ROOT47_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_97F0	Access Control Register (CCM_ACCESS_CTRL47)	32	R/W	0000_0000h	5.1.7.26/497
3038_97F4	Access Control Register (CCM_ACCESS_CTRL_ROOT47_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_97F8	Access Control Register (CCM_ACCESS_CTRL_ROOT47_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_97FC	Access Control Register (CCM_ACCESS_CTRL_ROOT47_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9800	Target Register (CCM_TARGET_ROOT48)	32	R/W	1000_0000h	5.1.7.10/461
3038_9804	Target Register (CCM_TARGET_ROOT48_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9808	Target Register (CCM_TARGET_ROOT48_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_980C	Target Register (CCM_TARGET_ROOT48_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9810	Miscellaneous Register (CCM_MISC48)	32	R/W	0000_0000h	5.1.7.14/469
3038_9814	Miscellaneous Register (CCM_MISC_ROOT48_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9818	Miscellaneous Register (CCM_MISC_ROOT48_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_981C	Miscellaneous Register (CCM_MISC_ROOT48_TOG)	32	R/W	0000_0000h	5.1.7.17/472

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9820	Post Divider Register (CCM_POST48)	32	R/W	0000_0000h	5.1.7.18/473
3038_9824	Post Divider Register (CCM_POST_ROOT48_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9828	Post Divider Register (CCM_POST_ROOT48_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_982C	Post Divider Register (CCM_POST_ROOT48_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9830	Pre Divider Register (CCM_PRE48)	32	R/W	1000_0000h	5.1.7.22/485
3038_9834	Pre Divider Register (CCM_PRE_ROOT48_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9838	Pre Divider Register (CCM_PRE_ROOT48_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_983C	Pre Divider Register (CCM_PRE_ROOT48_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9870	Access Control Register (CCM_ACCESS_CTRL48)	32	R/W	0000_0000h	5.1.7.26/497
3038_9874	Access Control Register (CCM_ACCESS_CTRL_ROOT48_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9878	Access Control Register (CCM_ACCESS_CTRL_ROOT48_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_987C	Access Control Register (CCM_ACCESS_CTRL_ROOT48_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9880	Target Register (CCM_TARGET_ROOT49)	32	R/W	1000_0000h	5.1.7.10/461
3038_9884	Target Register (CCM_TARGET_ROOT49_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9888	Target Register (CCM_TARGET_ROOT49_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_988C	Target Register (CCM_TARGET_ROOT49_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9890	Miscellaneous Register (CCM_MISC49)	32	R/W	0000_0000h	5.1.7.14/469
3038_9894	Miscellaneous Register (CCM_MISC_ROOT49_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9898	Miscellaneous Register (CCM_MISC_ROOT49_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_989C	Miscellaneous Register (CCM_MISC_ROOT49_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_98A0	Post Divider Register (CCM_POST49)	32	R/W	0000_0000h	5.1.7.18/473
3038_98A4	Post Divider Register (CCM_POST_ROOT49_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_98A8	Post Divider Register (CCM_POST_ROOT49_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_98AC	Post Divider Register (CCM_POST_ROOT49_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_98B0	Pre Divider Register (CCM_PRE49)	32	R/W	1000_0000h	5.1.7.22/485
3038_98B4	Pre Divider Register (CCM_PRE_ROOT49_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_98B8	Pre Divider Register (CCM_PRE_ROOT49_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_98BC	Pre Divider Register (CCM_PRE_ROOT49_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_98F0	Access Control Register (CCM_ACCESS_CTRL49)	32	R/W	0000_0000h	5.1.7.26/497
3038_98F4	Access Control Register (CCM_ACCESS_CTRL_ROOT49_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_98F8	Access Control Register (CCM_ACCESS_CTRL_ROOT49_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_98FC	Access Control Register (CCM_ACCESS_CTRL_ROOT49_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9900	Target Register (CCM_TARGET_ROOT50)	32	R/W	1000_0000h	5.1.7.10/461
3038_9904	Target Register (CCM_TARGET_ROOT50_SET)	32	R/W	0000_0000h	5.1.7.11/463

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9908	Target Register (CCM_TARGET_ROOT50_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_990C	Target Register (CCM_TARGET_ROOT50_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9910	Miscellaneous Register (CCM_MISC50)	32	R/W	0000_0000h	5.1.7.14/469
3038_9914	Miscellaneous Register (CCM_MISC_ROOT50_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9918	Miscellaneous Register (CCM_MISC_ROOT50_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_991C	Miscellaneous Register (CCM_MISC_ROOT50_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9920	Post Divider Register (CCM_POST50)	32	R/W	0000_0000h	5.1.7.18/473
3038_9924	Post Divider Register (CCM_POST_ROOT50_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9928	Post Divider Register (CCM_POST_ROOT50_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_992C	Post Divider Register (CCM_POST_ROOT50_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9930	Pre Divider Register (CCM_PRE50)	32	R/W	1000_0000h	5.1.7.22/485
3038_9934	Pre Divider Register (CCM_PRE_ROOT50_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9938	Pre Divider Register (CCM_PRE_ROOT50_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_993C	Pre Divider Register (CCM_PRE_ROOT50_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9970	Access Control Register (CCM_ACCESS_CTRL50)	32	R/W	0000_0000h	5.1.7.26/497
3038_9974	Access Control Register (CCM_ACCESS_CTRL_ROOT50_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9978	Access Control Register (CCM_ACCESS_CTRL_ROOT50_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_997C	Access Control Register (CCM_ACCESS_CTRL_ROOT50_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9980	Target Register (CCM_TARGET_ROOT51)	32	R/W	1000_0000h	5.1.7.10/461
3038_9984	Target Register (CCM_TARGET_ROOT51_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9988	Target Register (CCM_TARGET_ROOT51_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_998C	Target Register (CCM_TARGET_ROOT51_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9990	Miscellaneous Register (CCM_MISC51)	32	R/W	0000_0000h	5.1.7.14/469
3038_9994	Miscellaneous Register (CCM_MISC_ROOT51_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9998	Miscellaneous Register (CCM_MISC_ROOT51_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_999C	Miscellaneous Register (CCM_MISC_ROOT51_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_99A0	Post Divider Register (CCM_POST51)	32	R/W	0000_0000h	5.1.7.18/473
3038_99A4	Post Divider Register (CCM_POST_ROOT51_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_99A8	Post Divider Register (CCM_POST_ROOT51_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_99AC	Post Divider Register (CCM_POST_ROOT51_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_99B0	Pre Divider Register (CCM_PRE51)	32	R/W	1000_0000h	5.1.7.22/485
3038_99B4	Pre Divider Register (CCM_PRE_ROOT51_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_99B8	Pre Divider Register (CCM_PRE_ROOT51_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_99BC	Pre Divider Register (CCM_PRE_ROOT51_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_99F0	Access Control Register (CCM_ACCESS_CTRL51)	32	R/W	0000_0000h	5.1.7.26/497

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_99F4	Access Control Register (CCM_ACCESS_CTRL_ROOT51_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_99F8	Access Control Register (CCM_ACCESS_CTRL_ROOT51_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_99FC	Access Control Register (CCM_ACCESS_CTRL_ROOT51_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9A00	Target Register (CCM_TARGET_ROOT52)	32	R/W	1000_0000h	5.1.7.10/461
3038_9A04	Target Register (CCM_TARGET_ROOT52_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9A08	Target Register (CCM_TARGET_ROOT52_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_9A0C	Target Register (CCM_TARGET_ROOT52_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9A10	Miscellaneous Register (CCM_MISC52)	32	R/W	0000_0000h	5.1.7.14/469
3038_9A14	Miscellaneous Register (CCM_MISC_ROOT52_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9A18	Miscellaneous Register (CCM_MISC_ROOT52_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_9A1C	Miscellaneous Register (CCM_MISC_ROOT52_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9A20	Post Divider Register (CCM_POST52)	32	R/W	0000_0000h	5.1.7.18/473
3038_9A24	Post Divider Register (CCM_POST_ROOT52_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9A28	Post Divider Register (CCM_POST_ROOT52_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_9A2C	Post Divider Register (CCM_POST_ROOT52_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9A30	Pre Divider Register (CCM_PRE52)	32	R/W	1000_0000h	5.1.7.22/485
3038_9A34	Pre Divider Register (CCM_PRE_ROOT52_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9A38	Pre Divider Register (CCM_PRE_ROOT52_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_9A3C	Pre Divider Register (CCM_PRE_ROOT52_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9A70	Access Control Register (CCM_ACCESS_CTRL52)	32	R/W	0000_0000h	5.1.7.26/497
3038_9A74	Access Control Register (CCM_ACCESS_CTRL_ROOT52_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9A78	Access Control Register (CCM_ACCESS_CTRL_ROOT52_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_9A7C	Access Control Register (CCM_ACCESS_CTRL_ROOT52_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9A80	Target Register (CCM_TARGET_ROOT53)	32	R/W	1000_0000h	5.1.7.10/461
3038_9A84	Target Register (CCM_TARGET_ROOT53_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9A88	Target Register (CCM_TARGET_ROOT53_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_9A8C	Target Register (CCM_TARGET_ROOT53_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9A90	Miscellaneous Register (CCM_MISC53)	32	R/W	0000_0000h	5.1.7.14/469
3038_9A94	Miscellaneous Register (CCM_MISC_ROOT53_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9A98	Miscellaneous Register (CCM_MISC_ROOT53_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_9A9C	Miscellaneous Register (CCM_MISC_ROOT53_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9AA0	Post Divider Register (CCM_POST53)	32	R/W	0000_0000h	5.1.7.18/473
3038_9AA4	Post Divider Register (CCM_POST_ROOT53_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9AA8	Post Divider Register (CCM_POST_ROOT53_CLR)	32	R/W	0000_0000h	5.1.7.20/479

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9AAC	Post Divider Register (CCM_POST_ROOT53_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9AB0	Pre Divider Register (CCM_PRE53)	32	R/W	1000_0000h	5.1.7.22/485
3038_9AB4	Pre Divider Register (CCM_PRE_ROOT53_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9AB8	Pre Divider Register (CCM_PRE_ROOT53_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_9ABC	Pre Divider Register (CCM_PRE_ROOT53_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9AF0	Access Control Register (CCM_ACCESS_CTRL53)	32	R/W	0000_0000h	5.1.7.26/497
3038_9AF4	Access Control Register (CCM_ACCESS_CTRL_ROOT53_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9AF8	Access Control Register (CCM_ACCESS_CTRL_ROOT53_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_9AFC	Access Control Register (CCM_ACCESS_CTRL_ROOT53_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9B00	Target Register (CCM_TARGET_ROOT54)	32	R/W	1000_0000h	5.1.7.10/461
3038_9B04	Target Register (CCM_TARGET_ROOT54_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9B08	Target Register (CCM_TARGET_ROOT54_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_9B0C	Target Register (CCM_TARGET_ROOT54_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9B10	Miscellaneous Register (CCM_MISC54)	32	R/W	0000_0000h	5.1.7.14/469
3038_9B14	Miscellaneous Register (CCM_MISC_ROOT54_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9B18	Miscellaneous Register (CCM_MISC_ROOT54_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_9B1C	Miscellaneous Register (CCM_MISC_ROOT54_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9B20	Post Divider Register (CCM_POST54)	32	R/W	0000_0000h	5.1.7.18/473
3038_9B24	Post Divider Register (CCM_POST_ROOT54_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9B28	Post Divider Register (CCM_POST_ROOT54_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_9B2C	Post Divider Register (CCM_POST_ROOT54_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9B30	Pre Divider Register (CCM_PRE54)	32	R/W	1000_0000h	5.1.7.22/485
3038_9B34	Pre Divider Register (CCM_PRE_ROOT54_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9B38	Pre Divider Register (CCM_PRE_ROOT54_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_9B3C	Pre Divider Register (CCM_PRE_ROOT54_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9B70	Access Control Register (CCM_ACCESS_CTRL54)	32	R/W	0000_0000h	5.1.7.26/497
3038_9B74	Access Control Register (CCM_ACCESS_CTRL_ROOT54_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9B78	Access Control Register (CCM_ACCESS_CTRL_ROOT54_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_9B7C	Access Control Register (CCM_ACCESS_CTRL_ROOT54_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9B80	Target Register (CCM_TARGET_ROOT55)	32	R/W	1000_0000h	5.1.7.10/461
3038_9B84	Target Register (CCM_TARGET_ROOT55_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9B88	Target Register (CCM_TARGET_ROOT55_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_9B8C	Target Register (CCM_TARGET_ROOT55_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9B90	Miscellaneous Register (CCM_MISC55)	32	R/W	0000_0000h	5.1.7.14/469

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9B94	Miscellaneous Register (CCM_MISC_ROOT55_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9B98	Miscellaneous Register (CCM_MISC_ROOT55_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_9B9C	Miscellaneous Register (CCM_MISC_ROOT55_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9BA0	Post Divider Register (CCM_POST55)	32	R/W	0000_0000h	5.1.7.18/473
3038_9BA4	Post Divider Register (CCM_POST_ROOT55_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9BA8	Post Divider Register (CCM_POST_ROOT55_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_9BAC	Post Divider Register (CCM_POST_ROOT55_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9BB0	Pre Divider Register (CCM_PRE55)	32	R/W	1000_0000h	5.1.7.22/485
3038_9BB4	Pre Divider Register (CCM_PRE_ROOT55_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9BB8	Pre Divider Register (CCM_PRE_ROOT55_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_9BBC	Pre Divider Register (CCM_PRE_ROOT55_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9BF0	Access Control Register (CCM_ACCESS_CTRL55)	32	R/W	0000_0000h	5.1.7.26/497
3038_9BF4	Access Control Register (CCM_ACCESS_CTRL_ROOT55_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9BF8	Access Control Register (CCM_ACCESS_CTRL_ROOT55_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_9BFC	Access Control Register (CCM_ACCESS_CTRL_ROOT55_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9C00	Target Register (CCM_TARGET_ROOT56)	32	R/W	1000_0000h	5.1.7.10/461
3038_9C04	Target Register (CCM_TARGET_ROOT56_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9C08	Target Register (CCM_TARGET_ROOT56_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_9C0C	Target Register (CCM_TARGET_ROOT56_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9C10	Miscellaneous Register (CCM_MISC56)	32	R/W	0000_0000h	5.1.7.14/469
3038_9C14	Miscellaneous Register (CCM_MISC_ROOT56_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9C18	Miscellaneous Register (CCM_MISC_ROOT56_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_9C1C	Miscellaneous Register (CCM_MISC_ROOT56_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9C20	Post Divider Register (CCM_POST56)	32	R/W	0000_0000h	5.1.7.18/473
3038_9C24	Post Divider Register (CCM_POST_ROOT56_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9C28	Post Divider Register (CCM_POST_ROOT56_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_9C2C	Post Divider Register (CCM_POST_ROOT56_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9C30	Pre Divider Register (CCM_PRE56)	32	R/W	1000_0000h	5.1.7.22/485
3038_9C34	Pre Divider Register (CCM_PRE_ROOT56_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9C38	Pre Divider Register (CCM_PRE_ROOT56_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_9C3C	Pre Divider Register (CCM_PRE_ROOT56_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9C70	Access Control Register (CCM_ACCESS_CTRL56)	32	R/W	0000_0000h	5.1.7.26/497
3038_9C74	Access Control Register (CCM_ACCESS_CTRL_ROOT56_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9C78	Access Control Register (CCM_ACCESS_CTRL_ROOT56_CLR)	32	R/W	0000_0000h	5.1.7.28/502

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9C7C	Access Control Register (CCM_ACCESS_CTRL_ROOT56_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9C80	Target Register (CCM_TARGET_ROOT57)	32	R/W	1000_0000h	5.1.7.10/461
3038_9C84	Target Register (CCM_TARGET_ROOT57_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9C88	Target Register (CCM_TARGET_ROOT57_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_9C8C	Target Register (CCM_TARGET_ROOT57_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9C90	Miscellaneous Register (CCM_MISC57)	32	R/W	0000_0000h	5.1.7.14/469
3038_9C94	Miscellaneous Register (CCM_MISC_ROOT57_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9C98	Miscellaneous Register (CCM_MISC_ROOT57_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_9C9C	Miscellaneous Register (CCM_MISC_ROOT57_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9CA0	Post Divider Register (CCM_POST57)	32	R/W	0000_0000h	5.1.7.18/473
3038_9CA4	Post Divider Register (CCM_POST_ROOT57_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9CA8	Post Divider Register (CCM_POST_ROOT57_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_9CAC	Post Divider Register (CCM_POST_ROOT57_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9CB0	Pre Divider Register (CCM_PRE57)	32	R/W	1000_0000h	5.1.7.22/485
3038_9CB4	Pre Divider Register (CCM_PRE_ROOT57_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9CB8	Pre Divider Register (CCM_PRE_ROOT57_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_9CBC	Pre Divider Register (CCM_PRE_ROOT57_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9CF0	Access Control Register (CCM_ACCESS_CTRL57)	32	R/W	0000_0000h	5.1.7.26/497
3038_9CF4	Access Control Register (CCM_ACCESS_CTRL_ROOT57_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9CF8	Access Control Register (CCM_ACCESS_CTRL_ROOT57_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_9CFC	Access Control Register (CCM_ACCESS_CTRL_ROOT57_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9D00	Target Register (CCM_TARGET_ROOT58)	32	R/W	1000_0000h	5.1.7.10/461
3038_9D04	Target Register (CCM_TARGET_ROOT58_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9D08	Target Register (CCM_TARGET_ROOT58_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_9D0C	Target Register (CCM_TARGET_ROOT58_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9D10	Miscellaneous Register (CCM_MISC58)	32	R/W	0000_0000h	5.1.7.14/469
3038_9D14	Miscellaneous Register (CCM_MISC_ROOT58_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9D18	Miscellaneous Register (CCM_MISC_ROOT58_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_9D1C	Miscellaneous Register (CCM_MISC_ROOT58_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9D20	Post Divider Register (CCM_POST58)	32	R/W	0000_0000h	5.1.7.18/473
3038_9D24	Post Divider Register (CCM_POST_ROOT58_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9D28	Post Divider Register (CCM_POST_ROOT58_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_9D2C	Post Divider Register (CCM_POST_ROOT58_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9D30	Pre Divider Register (CCM_PRE58)	32	R/W	1000_0000h	5.1.7.22/485
3038_9D34	Pre Divider Register (CCM_PRE_ROOT58_SET)	32	R/W	0000_0000h	5.1.7.23/488

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9D38	Pre Divider Register (CCM_PRE_ROOT58_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_9D3C	Pre Divider Register (CCM_PRE_ROOT58_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9D70	Access Control Register (CCM_ACCESS_CTRL58)	32	R/W	0000_0000h	5.1.7.26/497
3038_9D74	Access Control Register (CCM_ACCESS_CTRL_ROOT58_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9D78	Access Control Register (CCM_ACCESS_CTRL_ROOT58_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_9D7C	Access Control Register (CCM_ACCESS_CTRL_ROOT58_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9D80	Target Register (CCM_TARGET_ROOT59)	32	R/W	1000_0000h	5.1.7.10/461
3038_9D84	Target Register (CCM_TARGET_ROOT59_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9D88	Target Register (CCM_TARGET_ROOT59_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_9D8C	Target Register (CCM_TARGET_ROOT59_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9D90	Miscellaneous Register (CCM_MISC59)	32	R/W	0000_0000h	5.1.7.14/469
3038_9D94	Miscellaneous Register (CCM_MISC_ROOT59_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9D98	Miscellaneous Register (CCM_MISC_ROOT59_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_9D9C	Miscellaneous Register (CCM_MISC_ROOT59_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9DA0	Post Divider Register (CCM_POST59)	32	R/W	0000_0000h	5.1.7.18/473
3038_9DA4	Post Divider Register (CCM_POST_ROOT59_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9DA8	Post Divider Register (CCM_POST_ROOT59_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_9DAC	Post Divider Register (CCM_POST_ROOT59_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9DB0	Pre Divider Register (CCM_PRE59)	32	R/W	1000_0000h	5.1.7.22/485
3038_9DB4	Pre Divider Register (CCM_PRE_ROOT59_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9DB8	Pre Divider Register (CCM_PRE_ROOT59_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_9DBC	Pre Divider Register (CCM_PRE_ROOT59_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9DF0	Access Control Register (CCM_ACCESS_CTRL59)	32	R/W	0000_0000h	5.1.7.26/497
3038_9DF4	Access Control Register (CCM_ACCESS_CTRL_ROOT59_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9DF8	Access Control Register (CCM_ACCESS_CTRL_ROOT59_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_9DFC	Access Control Register (CCM_ACCESS_CTRL_ROOT59_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9E00	Target Register (CCM_TARGET_ROOT60)	32	R/W	1000_0000h	5.1.7.10/461
3038_9E04	Target Register (CCM_TARGET_ROOT60_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9E08	Target Register (CCM_TARGET_ROOT60_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_9E0C	Target Register (CCM_TARGET_ROOT60_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9E10	Miscellaneous Register (CCM_MISC60)	32	R/W	0000_0000h	5.1.7.14/469
3038_9E14	Miscellaneous Register (CCM_MISC_ROOT60_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9E18	Miscellaneous Register (CCM_MISC_ROOT60_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_9E1C	Miscellaneous Register (CCM_MISC_ROOT60_TOG)	32	R/W	0000_0000h	5.1.7.17/472

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9E20	Post Divider Register (CCM_POST60)	32	R/W	0000_0000h	5.1.7.18/473
3038_9E24	Post Divider Register (CCM_POST_ROOT60_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9E28	Post Divider Register (CCM_POST_ROOT60_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_9E2C	Post Divider Register (CCM_POST_ROOT60_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9E30	Pre Divider Register (CCM_PRE60)	32	R/W	1000_0000h	5.1.7.22/485
3038_9E34	Pre Divider Register (CCM_PRE_ROOT60_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9E38	Pre Divider Register (CCM_PRE_ROOT60_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_9E3C	Pre Divider Register (CCM_PRE_ROOT60_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9E70	Access Control Register (CCM_ACCESS_CTRL60)	32	R/W	0000_0000h	5.1.7.26/497
3038_9E74	Access Control Register (CCM_ACCESS_CTRL_ROOT60_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9E78	Access Control Register (CCM_ACCESS_CTRL_ROOT60_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_9E7C	Access Control Register (CCM_ACCESS_CTRL_ROOT60_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9E80	Target Register (CCM_TARGET_ROOT61)	32	R/W	1000_0000h	5.1.7.10/461
3038_9E84	Target Register (CCM_TARGET_ROOT61_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9E88	Target Register (CCM_TARGET_ROOT61_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_9E8C	Target Register (CCM_TARGET_ROOT61_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9E90	Miscellaneous Register (CCM_MISC61)	32	R/W	0000_0000h	5.1.7.14/469
3038_9E94	Miscellaneous Register (CCM_MISC_ROOT61_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9E98	Miscellaneous Register (CCM_MISC_ROOT61_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_9E9C	Miscellaneous Register (CCM_MISC_ROOT61_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9EA0	Post Divider Register (CCM_POST61)	32	R/W	0000_0000h	5.1.7.18/473
3038_9EA4	Post Divider Register (CCM_POST_ROOT61_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9EA8	Post Divider Register (CCM_POST_ROOT61_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_9EAC	Post Divider Register (CCM_POST_ROOT61_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9EB0	Pre Divider Register (CCM_PRE61)	32	R/W	1000_0000h	5.1.7.22/485
3038_9EB4	Pre Divider Register (CCM_PRE_ROOT61_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9EB8	Pre Divider Register (CCM_PRE_ROOT61_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_9EBC	Pre Divider Register (CCM_PRE_ROOT61_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9EF0	Access Control Register (CCM_ACCESS_CTRL61)	32	R/W	0000_0000h	5.1.7.26/497
3038_9EF4	Access Control Register (CCM_ACCESS_CTRL_ROOT61_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9EF8	Access Control Register (CCM_ACCESS_CTRL_ROOT61_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_9EFC	Access Control Register (CCM_ACCESS_CTRL_ROOT61_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9F00	Target Register (CCM_TARGET_ROOT62)	32	R/W	1000_0000h	5.1.7.10/461
3038_9F04	Target Register (CCM_TARGET_ROOT62_SET)	32	R/W	0000_0000h	5.1.7.11/463

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9F08	Target Register (CCM_TARGET_ROOT62_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_9F0C	Target Register (CCM_TARGET_ROOT62_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9F10	Miscellaneous Register (CCM_MISC62)	32	R/W	0000_0000h	5.1.7.14/469
3038_9F14	Miscellaneous Register (CCM_MISC_ROOT62_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9F18	Miscellaneous Register (CCM_MISC_ROOT62_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_9F1C	Miscellaneous Register (CCM_MISC_ROOT62_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9F20	Post Divider Register (CCM_POST62)	32	R/W	0000_0000h	5.1.7.18/473
3038_9F24	Post Divider Register (CCM_POST_ROOT62_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9F28	Post Divider Register (CCM_POST_ROOT62_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_9F2C	Post Divider Register (CCM_POST_ROOT62_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9F30	Pre Divider Register (CCM_PRE62)	32	R/W	1000_0000h	5.1.7.22/485
3038_9F34	Pre Divider Register (CCM_PRE_ROOT62_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9F38	Pre Divider Register (CCM_PRE_ROOT62_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_9F3C	Pre Divider Register (CCM_PRE_ROOT62_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9F70	Access Control Register (CCM_ACCESS_CTRL62)	32	R/W	0000_0000h	5.1.7.26/497
3038_9F74	Access Control Register (CCM_ACCESS_CTRL_ROOT62_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9F78	Access Control Register (CCM_ACCESS_CTRL_ROOT62_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_9F7C	Access Control Register (CCM_ACCESS_CTRL_ROOT62_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_9F80	Target Register (CCM_TARGET_ROOT63)	32	R/W	1000_0000h	5.1.7.10/461
3038_9F84	Target Register (CCM_TARGET_ROOT63_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_9F88	Target Register (CCM_TARGET_ROOT63_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_9F8C	Target Register (CCM_TARGET_ROOT63_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_9F90	Miscellaneous Register (CCM_MISC63)	32	R/W	0000_0000h	5.1.7.14/469
3038_9F94	Miscellaneous Register (CCM_MISC_ROOT63_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_9F98	Miscellaneous Register (CCM_MISC_ROOT63_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_9F9C	Miscellaneous Register (CCM_MISC_ROOT63_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_9FA0	Post Divider Register (CCM_POST63)	32	R/W	0000_0000h	5.1.7.18/473
3038_9FA4	Post Divider Register (CCM_POST_ROOT63_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_9FA8	Post Divider Register (CCM_POST_ROOT63_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_9FAC	Post Divider Register (CCM_POST_ROOT63_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_9FB0	Pre Divider Register (CCM_PRE63)	32	R/W	1000_0000h	5.1.7.22/485
3038_9FB4	Pre Divider Register (CCM_PRE_ROOT63_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_9FB8	Pre Divider Register (CCM_PRE_ROOT63_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_9FBC	Pre Divider Register (CCM_PRE_ROOT63_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_9FF0	Access Control Register (CCM_ACCESS_CTRL63)	32	R/W	0000_0000h	5.1.7.26/497

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9FF4	Access Control Register (CCM_ACCESS_CTRL_ROOT63_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_9FF8	Access Control Register (CCM_ACCESS_CTRL_ROOT63_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_9FFC	Access Control Register (CCM_ACCESS_CTRL_ROOT63_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A000	Target Register (CCM_TARGET_ROOT64)	32	R/W	1000_0000h	5.1.7.10/461
3038_A004	Target Register (CCM_TARGET_ROOT64_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A008	Target Register (CCM_TARGET_ROOT64_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A00C	Target Register (CCM_TARGET_ROOT64_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A010	Miscellaneous Register (CCM_MISC64)	32	R/W	0000_0000h	5.1.7.14/469
3038_A014	Miscellaneous Register (CCM_MISC_ROOT64_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A018	Miscellaneous Register (CCM_MISC_ROOT64_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A01C	Miscellaneous Register (CCM_MISC_ROOT64_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A020	Post Divider Register (CCM_POST64)	32	R/W	0000_0000h	5.1.7.18/473
3038_A024	Post Divider Register (CCM_POST_ROOT64_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A028	Post Divider Register (CCM_POST_ROOT64_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A02C	Post Divider Register (CCM_POST_ROOT64_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A030	Pre Divider Register (CCM_PRE64)	32	R/W	1000_0000h	5.1.7.22/485
3038_A034	Pre Divider Register (CCM_PRE_ROOT64_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A038	Pre Divider Register (CCM_PRE_ROOT64_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A03C	Pre Divider Register (CCM_PRE_ROOT64_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A070	Access Control Register (CCM_ACCESS_CTRL64)	32	R/W	0000_0000h	5.1.7.26/497
3038_A074	Access Control Register (CCM_ACCESS_CTRL_ROOT64_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A078	Access Control Register (CCM_ACCESS_CTRL_ROOT64_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A07C	Access Control Register (CCM_ACCESS_CTRL_ROOT64_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A080	Target Register (CCM_TARGET_ROOT65)	32	R/W	1000_0000h	5.1.7.10/461
3038_A084	Target Register (CCM_TARGET_ROOT65_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A088	Target Register (CCM_TARGET_ROOT65_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A08C	Target Register (CCM_TARGET_ROOT65_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A090	Miscellaneous Register (CCM_MISC65)	32	R/W	0000_0000h	5.1.7.14/469
3038_A094	Miscellaneous Register (CCM_MISC_ROOT65_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A098	Miscellaneous Register (CCM_MISC_ROOT65_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A09C	Miscellaneous Register (CCM_MISC_ROOT65_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A0A0	Post Divider Register (CCM_POST65)	32	R/W	0000_0000h	5.1.7.18/473
3038_A0A4	Post Divider Register (CCM_POST_ROOT65_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A0A8	Post Divider Register (CCM_POST_ROOT65_CLR)	32	R/W	0000_0000h	5.1.7.20/479

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A0AC	Post Divider Register (CCM_POST_ROOT65_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A0B0	Pre Divider Register (CCM_PRE65)	32	R/W	1000_0000h	5.1.7.22/485
3038_A0B4	Pre Divider Register (CCM_PRE_ROOT65_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A0B8	Pre Divider Register (CCM_PRE_ROOT65_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A0BC	Pre Divider Register (CCM_PRE_ROOT65_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A0F0	Access Control Register (CCM_ACCESS_CTRL65)	32	R/W	0000_0000h	5.1.7.26/497
3038_A0F4	Access Control Register (CCM_ACCESS_CTRL_ROOT65_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A0F8	Access Control Register (CCM_ACCESS_CTRL_ROOT65_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A0FC	Access Control Register (CCM_ACCESS_CTRL_ROOT65_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A100	Target Register (CCM_TARGET_ROOT66)	32	R/W	1000_0000h	5.1.7.10/461
3038_A104	Target Register (CCM_TARGET_ROOT66_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A108	Target Register (CCM_TARGET_ROOT66_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A10C	Target Register (CCM_TARGET_ROOT66_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A110	Miscellaneous Register (CCM_MISC66)	32	R/W	0000_0000h	5.1.7.14/469
3038_A114	Miscellaneous Register (CCM_MISC_ROOT66_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A118	Miscellaneous Register (CCM_MISC_ROOT66_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A11C	Miscellaneous Register (CCM_MISC_ROOT66_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A120	Post Divider Register (CCM_POST66)	32	R/W	0000_0000h	5.1.7.18/473
3038_A124	Post Divider Register (CCM_POST_ROOT66_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A128	Post Divider Register (CCM_POST_ROOT66_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A12C	Post Divider Register (CCM_POST_ROOT66_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A130	Pre Divider Register (CCM_PRE66)	32	R/W	1000_0000h	5.1.7.22/485
3038_A134	Pre Divider Register (CCM_PRE_ROOT66_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A138	Pre Divider Register (CCM_PRE_ROOT66_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A13C	Pre Divider Register (CCM_PRE_ROOT66_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A170	Access Control Register (CCM_ACCESS_CTRL66)	32	R/W	0000_0000h	5.1.7.26/497
3038_A174	Access Control Register (CCM_ACCESS_CTRL_ROOT66_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A178	Access Control Register (CCM_ACCESS_CTRL_ROOT66_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A17C	Access Control Register (CCM_ACCESS_CTRL_ROOT66_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A180	Target Register (CCM_TARGET_ROOT67)	32	R/W	1000_0000h	5.1.7.10/461
3038_A184	Target Register (CCM_TARGET_ROOT67_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A188	Target Register (CCM_TARGET_ROOT67_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A18C	Target Register (CCM_TARGET_ROOT67_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A190	Miscellaneous Register (CCM_MISC67)	32	R/W	0000_0000h	5.1.7.14/469

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A194	Miscellaneous Register (CCM_MISC_ROOT67_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A198	Miscellaneous Register (CCM_MISC_ROOT67_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A19C	Miscellaneous Register (CCM_MISC_ROOT67_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A1A0	Post Divider Register (CCM_POST67)	32	R/W	0000_0000h	5.1.7.18/473
3038_A1A4	Post Divider Register (CCM_POST_ROOT67_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A1A8	Post Divider Register (CCM_POST_ROOT67_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A1AC	Post Divider Register (CCM_POST_ROOT67_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A1B0	Pre Divider Register (CCM_PRE67)	32	R/W	1000_0000h	5.1.7.22/485
3038_A1B4	Pre Divider Register (CCM_PRE_ROOT67_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A1B8	Pre Divider Register (CCM_PRE_ROOT67_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A1BC	Pre Divider Register (CCM_PRE_ROOT67_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A1F0	Access Control Register (CCM_ACCESS_CTRL67)	32	R/W	0000_0000h	5.1.7.26/497
3038_A1F4	Access Control Register (CCM_ACCESS_CTRL_ROOT67_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A1F8	Access Control Register (CCM_ACCESS_CTRL_ROOT67_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A1FC	Access Control Register (CCM_ACCESS_CTRL_ROOT67_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A200	Target Register (CCM_TARGET_ROOT68)	32	R/W	1000_0000h	5.1.7.10/461
3038_A204	Target Register (CCM_TARGET_ROOT68_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A208	Target Register (CCM_TARGET_ROOT68_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A20C	Target Register (CCM_TARGET_ROOT68_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A210	Miscellaneous Register (CCM_MISC68)	32	R/W	0000_0000h	5.1.7.14/469
3038_A214	Miscellaneous Register (CCM_MISC_ROOT68_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A218	Miscellaneous Register (CCM_MISC_ROOT68_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A21C	Miscellaneous Register (CCM_MISC_ROOT68_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A220	Post Divider Register (CCM_POST68)	32	R/W	0000_0000h	5.1.7.18/473
3038_A224	Post Divider Register (CCM_POST_ROOT68_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A228	Post Divider Register (CCM_POST_ROOT68_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A22C	Post Divider Register (CCM_POST_ROOT68_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A230	Pre Divider Register (CCM_PRE68)	32	R/W	1000_0000h	5.1.7.22/485
3038_A234	Pre Divider Register (CCM_PRE_ROOT68_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A238	Pre Divider Register (CCM_PRE_ROOT68_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A23C	Pre Divider Register (CCM_PRE_ROOT68_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A270	Access Control Register (CCM_ACCESS_CTRL68)	32	R/W	0000_0000h	5.1.7.26/497
3038_A274	Access Control Register (CCM_ACCESS_CTRL_ROOT68_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A278	Access Control Register (CCM_ACCESS_CTRL_ROOT68_CLR)	32	R/W	0000_0000h	5.1.7.28/502

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A27C	Access Control Register (CCM_ACCESS_CTRL_ROOT68_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A280	Target Register (CCM_TARGET_ROOT69)	32	R/W	1000_0000h	5.1.7.10/461
3038_A284	Target Register (CCM_TARGET_ROOT69_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A288	Target Register (CCM_TARGET_ROOT69_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A28C	Target Register (CCM_TARGET_ROOT69_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A290	Miscellaneous Register (CCM_MISC69)	32	R/W	0000_0000h	5.1.7.14/469
3038_A294	Miscellaneous Register (CCM_MISC_ROOT69_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A298	Miscellaneous Register (CCM_MISC_ROOT69_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A29C	Miscellaneous Register (CCM_MISC_ROOT69_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A2A0	Post Divider Register (CCM_POST69)	32	R/W	0000_0000h	5.1.7.18/473
3038_A2A4	Post Divider Register (CCM_POST_ROOT69_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A2A8	Post Divider Register (CCM_POST_ROOT69_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A2AC	Post Divider Register (CCM_POST_ROOT69_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A2B0	Pre Divider Register (CCM_PRE69)	32	R/W	1000_0000h	5.1.7.22/485
3038_A2B4	Pre Divider Register (CCM_PRE_ROOT69_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A2B8	Pre Divider Register (CCM_PRE_ROOT69_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A2BC	Pre Divider Register (CCM_PRE_ROOT69_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A2F0	Access Control Register (CCM_ACCESS_CTRL69)	32	R/W	0000_0000h	5.1.7.26/497
3038_A2F4	Access Control Register (CCM_ACCESS_CTRL_ROOT69_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A2F8	Access Control Register (CCM_ACCESS_CTRL_ROOT69_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A2FC	Access Control Register (CCM_ACCESS_CTRL_ROOT69_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A300	Target Register (CCM_TARGET_ROOT70)	32	R/W	1000_0000h	5.1.7.10/461
3038_A304	Target Register (CCM_TARGET_ROOT70_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A308	Target Register (CCM_TARGET_ROOT70_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A30C	Target Register (CCM_TARGET_ROOT70_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A310	Miscellaneous Register (CCM_MISC70)	32	R/W	0000_0000h	5.1.7.14/469
3038_A314	Miscellaneous Register (CCM_MISC_ROOT70_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A318	Miscellaneous Register (CCM_MISC_ROOT70_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A31C	Miscellaneous Register (CCM_MISC_ROOT70_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A320	Post Divider Register (CCM_POST70)	32	R/W	0000_0000h	5.1.7.18/473
3038_A324	Post Divider Register (CCM_POST_ROOT70_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A328	Post Divider Register (CCM_POST_ROOT70_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A32C	Post Divider Register (CCM_POST_ROOT70_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A330	Pre Divider Register (CCM_PRE70)	32	R/W	1000_0000h	5.1.7.22/485
3038_A334	Pre Divider Register (CCM_PRE_ROOT70_SET)	32	R/W	0000_0000h	5.1.7.23/488

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A338	Pre Divider Register (CCM_PRE_ROOT70_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A33C	Pre Divider Register (CCM_PRE_ROOT70_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A370	Access Control Register (CCM_ACCESS_CTRL70)	32	R/W	0000_0000h	5.1.7.26/497
3038_A374	Access Control Register (CCM_ACCESS_CTRL_ROOT70_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A378	Access Control Register (CCM_ACCESS_CTRL_ROOT70_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A37C	Access Control Register (CCM_ACCESS_CTRL_ROOT70_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A380	Target Register (CCM_TARGET_ROOT71)	32	R/W	1000_0000h	5.1.7.10/461
3038_A384	Target Register (CCM_TARGET_ROOT71_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A388	Target Register (CCM_TARGET_ROOT71_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A38C	Target Register (CCM_TARGET_ROOT71_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A390	Miscellaneous Register (CCM_MISC71)	32	R/W	0000_0000h	5.1.7.14/469
3038_A394	Miscellaneous Register (CCM_MISC_ROOT71_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A398	Miscellaneous Register (CCM_MISC_ROOT71_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A39C	Miscellaneous Register (CCM_MISC_ROOT71_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A3A0	Post Divider Register (CCM_POST71)	32	R/W	0000_0000h	5.1.7.18/473
3038_A3A4	Post Divider Register (CCM_POST_ROOT71_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A3A8	Post Divider Register (CCM_POST_ROOT71_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A3AC	Post Divider Register (CCM_POST_ROOT71_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A3B0	Pre Divider Register (CCM_PRE71)	32	R/W	1000_0000h	5.1.7.22/485
3038_A3B4	Pre Divider Register (CCM_PRE_ROOT71_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A3B8	Pre Divider Register (CCM_PRE_ROOT71_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A3BC	Pre Divider Register (CCM_PRE_ROOT71_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A3F0	Access Control Register (CCM_ACCESS_CTRL71)	32	R/W	0000_0000h	5.1.7.26/497
3038_A3F4	Access Control Register (CCM_ACCESS_CTRL_ROOT71_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A3F8	Access Control Register (CCM_ACCESS_CTRL_ROOT71_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A3FC	Access Control Register (CCM_ACCESS_CTRL_ROOT71_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A400	Target Register (CCM_TARGET_ROOT72)	32	R/W	1000_0000h	5.1.7.10/461
3038_A404	Target Register (CCM_TARGET_ROOT72_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A408	Target Register (CCM_TARGET_ROOT72_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A40C	Target Register (CCM_TARGET_ROOT72_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A410	Miscellaneous Register (CCM_MISC72)	32	R/W	0000_0000h	5.1.7.14/469
3038_A414	Miscellaneous Register (CCM_MISC_ROOT72_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A418	Miscellaneous Register (CCM_MISC_ROOT72_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A41C	Miscellaneous Register (CCM_MISC_ROOT72_TOG)	32	R/W	0000_0000h	5.1.7.17/472

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A420	Post Divider Register (CCM_POST72)	32	R/W	0000_0000h	5.1.7.18/473
3038_A424	Post Divider Register (CCM_POST_ROOT72_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A428	Post Divider Register (CCM_POST_ROOT72_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A42C	Post Divider Register (CCM_POST_ROOT72_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A430	Pre Divider Register (CCM_PRE72)	32	R/W	1000_0000h	5.1.7.22/485
3038_A434	Pre Divider Register (CCM_PRE_ROOT72_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A438	Pre Divider Register (CCM_PRE_ROOT72_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A43C	Pre Divider Register (CCM_PRE_ROOT72_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A470	Access Control Register (CCM_ACCESS_CTRL72)	32	R/W	0000_0000h	5.1.7.26/497
3038_A474	Access Control Register (CCM_ACCESS_CTRL_ROOT72_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A478	Access Control Register (CCM_ACCESS_CTRL_ROOT72_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A47C	Access Control Register (CCM_ACCESS_CTRL_ROOT72_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A480	Target Register (CCM_TARGET_ROOT73)	32	R/W	1000_0000h	5.1.7.10/461
3038_A484	Target Register (CCM_TARGET_ROOT73_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A488	Target Register (CCM_TARGET_ROOT73_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A48C	Target Register (CCM_TARGET_ROOT73_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A490	Miscellaneous Register (CCM_MISC73)	32	R/W	0000_0000h	5.1.7.14/469
3038_A494	Miscellaneous Register (CCM_MISC_ROOT73_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A498	Miscellaneous Register (CCM_MISC_ROOT73_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A49C	Miscellaneous Register (CCM_MISC_ROOT73_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A4A0	Post Divider Register (CCM_POST73)	32	R/W	0000_0000h	5.1.7.18/473
3038_A4A4	Post Divider Register (CCM_POST_ROOT73_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A4A8	Post Divider Register (CCM_POST_ROOT73_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A4AC	Post Divider Register (CCM_POST_ROOT73_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A4B0	Pre Divider Register (CCM_PRE73)	32	R/W	1000_0000h	5.1.7.22/485
3038_A4B4	Pre Divider Register (CCM_PRE_ROOT73_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A4B8	Pre Divider Register (CCM_PRE_ROOT73_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A4BC	Pre Divider Register (CCM_PRE_ROOT73_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A4F0	Access Control Register (CCM_ACCESS_CTRL73)	32	R/W	0000_0000h	5.1.7.26/497
3038_A4F4	Access Control Register (CCM_ACCESS_CTRL_ROOT73_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A4F8	Access Control Register (CCM_ACCESS_CTRL_ROOT73_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A4FC	Access Control Register (CCM_ACCESS_CTRL_ROOT73_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A500	Target Register (CCM_TARGET_ROOT74)	32	R/W	1000_0000h	5.1.7.10/461
3038_A504	Target Register (CCM_TARGET_ROOT74_SET)	32	R/W	0000_0000h	5.1.7.11/463

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A508	Target Register (CCM_TARGET_ROOT74_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A50C	Target Register (CCM_TARGET_ROOT74_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A510	Miscellaneous Register (CCM_MISC74)	32	R/W	0000_0000h	5.1.7.14/469
3038_A514	Miscellaneous Register (CCM_MISC_ROOT74_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A518	Miscellaneous Register (CCM_MISC_ROOT74_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A51C	Miscellaneous Register (CCM_MISC_ROOT74_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A520	Post Divider Register (CCM_POST74)	32	R/W	0000_0000h	5.1.7.18/473
3038_A524	Post Divider Register (CCM_POST_ROOT74_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A528	Post Divider Register (CCM_POST_ROOT74_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A52C	Post Divider Register (CCM_POST_ROOT74_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A530	Pre Divider Register (CCM_PRE74)	32	R/W	1000_0000h	5.1.7.22/485
3038_A534	Pre Divider Register (CCM_PRE_ROOT74_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A538	Pre Divider Register (CCM_PRE_ROOT74_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A53C	Pre Divider Register (CCM_PRE_ROOT74_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A570	Access Control Register (CCM_ACCESS_CTRL74)	32	R/W	0000_0000h	5.1.7.26/497
3038_A574	Access Control Register (CCM_ACCESS_CTRL_ROOT74_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A578	Access Control Register (CCM_ACCESS_CTRL_ROOT74_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A57C	Access Control Register (CCM_ACCESS_CTRL_ROOT74_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A580	Target Register (CCM_TARGET_ROOT75)	32	R/W	1000_0000h	5.1.7.10/461
3038_A584	Target Register (CCM_TARGET_ROOT75_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A588	Target Register (CCM_TARGET_ROOT75_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A58C	Target Register (CCM_TARGET_ROOT75_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A590	Miscellaneous Register (CCM_MISC75)	32	R/W	0000_0000h	5.1.7.14/469
3038_A594	Miscellaneous Register (CCM_MISC_ROOT75_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A598	Miscellaneous Register (CCM_MISC_ROOT75_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A59C	Miscellaneous Register (CCM_MISC_ROOT75_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A5A0	Post Divider Register (CCM_POST75)	32	R/W	0000_0000h	5.1.7.18/473
3038_A5A4	Post Divider Register (CCM_POST_ROOT75_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A5A8	Post Divider Register (CCM_POST_ROOT75_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A5AC	Post Divider Register (CCM_POST_ROOT75_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A5B0	Pre Divider Register (CCM_PRE75)	32	R/W	1000_0000h	5.1.7.22/485
3038_A5B4	Pre Divider Register (CCM_PRE_ROOT75_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A5B8	Pre Divider Register (CCM_PRE_ROOT75_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A5BC	Pre Divider Register (CCM_PRE_ROOT75_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A5F0	Access Control Register (CCM_ACCESS_CTRL75)	32	R/W	0000_0000h	5.1.7.26/497

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A5F4	Access Control Register (CCM_ACCESS_CTRL_ROOT75_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A5F8	Access Control Register (CCM_ACCESS_CTRL_ROOT75_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A5FC	Access Control Register (CCM_ACCESS_CTRL_ROOT75_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A600	Target Register (CCM_TARGET_ROOT76)	32	R/W	1000_0000h	5.1.7.10/461
3038_A604	Target Register (CCM_TARGET_ROOT76_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A608	Target Register (CCM_TARGET_ROOT76_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A60C	Target Register (CCM_TARGET_ROOT76_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A610	Miscellaneous Register (CCM_MISC76)	32	R/W	0000_0000h	5.1.7.14/469
3038_A614	Miscellaneous Register (CCM_MISC_ROOT76_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A618	Miscellaneous Register (CCM_MISC_ROOT76_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A61C	Miscellaneous Register (CCM_MISC_ROOT76_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A620	Post Divider Register (CCM_POST76)	32	R/W	0000_0000h	5.1.7.18/473
3038_A624	Post Divider Register (CCM_POST_ROOT76_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A628	Post Divider Register (CCM_POST_ROOT76_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A62C	Post Divider Register (CCM_POST_ROOT76_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A630	Pre Divider Register (CCM_PRE76)	32	R/W	1000_0000h	5.1.7.22/485
3038_A634	Pre Divider Register (CCM_PRE_ROOT76_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A638	Pre Divider Register (CCM_PRE_ROOT76_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A63C	Pre Divider Register (CCM_PRE_ROOT76_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A670	Access Control Register (CCM_ACCESS_CTRL76)	32	R/W	0000_0000h	5.1.7.26/497
3038_A674	Access Control Register (CCM_ACCESS_CTRL_ROOT76_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A678	Access Control Register (CCM_ACCESS_CTRL_ROOT76_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A67C	Access Control Register (CCM_ACCESS_CTRL_ROOT76_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A680	Target Register (CCM_TARGET_ROOT77)	32	R/W	1000_0000h	5.1.7.10/461
3038_A684	Target Register (CCM_TARGET_ROOT77_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A688	Target Register (CCM_TARGET_ROOT77_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A68C	Target Register (CCM_TARGET_ROOT77_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A690	Miscellaneous Register (CCM_MISC77)	32	R/W	0000_0000h	5.1.7.14/469
3038_A694	Miscellaneous Register (CCM_MISC_ROOT77_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A698	Miscellaneous Register (CCM_MISC_ROOT77_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A69C	Miscellaneous Register (CCM_MISC_ROOT77_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A6A0	Post Divider Register (CCM_POST77)	32	R/W	0000_0000h	5.1.7.18/473
3038_A6A4	Post Divider Register (CCM_POST_ROOT77_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A6A8	Post Divider Register (CCM_POST_ROOT77_CLR)	32	R/W	0000_0000h	5.1.7.20/479

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A6AC	Post Divider Register (CCM_POST_ROOT77_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A6B0	Pre Divider Register (CCM_PRE77)	32	R/W	1000_0000h	5.1.7.22/485
3038_A6B4	Pre Divider Register (CCM_PRE_ROOT77_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A6B8	Pre Divider Register (CCM_PRE_ROOT77_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A6BC	Pre Divider Register (CCM_PRE_ROOT77_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A6F0	Access Control Register (CCM_ACCESS_CTRL77)	32	R/W	0000_0000h	5.1.7.26/497
3038_A6F4	Access Control Register (CCM_ACCESS_CTRL_ROOT77_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A6F8	Access Control Register (CCM_ACCESS_CTRL_ROOT77_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A6FC	Access Control Register (CCM_ACCESS_CTRL_ROOT77_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A700	Target Register (CCM_TARGET_ROOT78)	32	R/W	1000_0000h	5.1.7.10/461
3038_A704	Target Register (CCM_TARGET_ROOT78_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A708	Target Register (CCM_TARGET_ROOT78_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A70C	Target Register (CCM_TARGET_ROOT78_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A710	Miscellaneous Register (CCM_MISC78)	32	R/W	0000_0000h	5.1.7.14/469
3038_A714	Miscellaneous Register (CCM_MISC_ROOT78_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A718	Miscellaneous Register (CCM_MISC_ROOT78_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A71C	Miscellaneous Register (CCM_MISC_ROOT78_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A720	Post Divider Register (CCM_POST78)	32	R/W	0000_0000h	5.1.7.18/473
3038_A724	Post Divider Register (CCM_POST_ROOT78_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A728	Post Divider Register (CCM_POST_ROOT78_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A72C	Post Divider Register (CCM_POST_ROOT78_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A730	Pre Divider Register (CCM_PRE78)	32	R/W	1000_0000h	5.1.7.22/485
3038_A734	Pre Divider Register (CCM_PRE_ROOT78_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A738	Pre Divider Register (CCM_PRE_ROOT78_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A73C	Pre Divider Register (CCM_PRE_ROOT78_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A770	Access Control Register (CCM_ACCESS_CTRL78)	32	R/W	0000_0000h	5.1.7.26/497
3038_A774	Access Control Register (CCM_ACCESS_CTRL_ROOT78_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A778	Access Control Register (CCM_ACCESS_CTRL_ROOT78_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A77C	Access Control Register (CCM_ACCESS_CTRL_ROOT78_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A780	Target Register (CCM_TARGET_ROOT79)	32	R/W	1000_0000h	5.1.7.10/461
3038_A784	Target Register (CCM_TARGET_ROOT79_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A788	Target Register (CCM_TARGET_ROOT79_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A78C	Target Register (CCM_TARGET_ROOT79_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A790	Miscellaneous Register (CCM_MISC79)	32	R/W	0000_0000h	5.1.7.14/469

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A794	Miscellaneous Register (CCM_MISC_ROOT79_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A798	Miscellaneous Register (CCM_MISC_ROOT79_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A79C	Miscellaneous Register (CCM_MISC_ROOT79_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A7A0	Post Divider Register (CCM_POST79)	32	R/W	0000_0000h	5.1.7.18/473
3038_A7A4	Post Divider Register (CCM_POST_ROOT79_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A7A8	Post Divider Register (CCM_POST_ROOT79_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A7AC	Post Divider Register (CCM_POST_ROOT79_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A7B0	Pre Divider Register (CCM_PRE79)	32	R/W	1000_0000h	5.1.7.22/485
3038_A7B4	Pre Divider Register (CCM_PRE_ROOT79_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A7B8	Pre Divider Register (CCM_PRE_ROOT79_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A7BC	Pre Divider Register (CCM_PRE_ROOT79_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A7F0	Access Control Register (CCM_ACCESS_CTRL79)	32	R/W	0000_0000h	5.1.7.26/497
3038_A7F4	Access Control Register (CCM_ACCESS_CTRL_ROOT79_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A7F8	Access Control Register (CCM_ACCESS_CTRL_ROOT79_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A7FC	Access Control Register (CCM_ACCESS_CTRL_ROOT79_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A800	Target Register (CCM_TARGET_ROOT80)	32	R/W	1000_0000h	5.1.7.10/461
3038_A804	Target Register (CCM_TARGET_ROOT80_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A808	Target Register (CCM_TARGET_ROOT80_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A80C	Target Register (CCM_TARGET_ROOT80_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A810	Miscellaneous Register (CCM_MISC80)	32	R/W	0000_0000h	5.1.7.14/469
3038_A814	Miscellaneous Register (CCM_MISC_ROOT80_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A818	Miscellaneous Register (CCM_MISC_ROOT80_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A81C	Miscellaneous Register (CCM_MISC_ROOT80_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A820	Post Divider Register (CCM_POST80)	32	R/W	0000_0000h	5.1.7.18/473
3038_A824	Post Divider Register (CCM_POST_ROOT80_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A828	Post Divider Register (CCM_POST_ROOT80_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A82C	Post Divider Register (CCM_POST_ROOT80_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A830	Pre Divider Register (CCM_PRE80)	32	R/W	1000_0000h	5.1.7.22/485
3038_A834	Pre Divider Register (CCM_PRE_ROOT80_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A838	Pre Divider Register (CCM_PRE_ROOT80_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A83C	Pre Divider Register (CCM_PRE_ROOT80_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A870	Access Control Register (CCM_ACCESS_CTRL80)	32	R/W	0000_0000h	5.1.7.26/497
3038_A874	Access Control Register (CCM_ACCESS_CTRL_ROOT80_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A878	Access Control Register (CCM_ACCESS_CTRL_ROOT80_CLR)	32	R/W	0000_0000h	5.1.7.28/502

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A87C	Access Control Register (CCM_ACCESS_CTRL_ROOT80_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A880	Target Register (CCM_TARGET_ROOT81)	32	R/W	1000_0000h	5.1.7.10/461
3038_A884	Target Register (CCM_TARGET_ROOT81_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A888	Target Register (CCM_TARGET_ROOT81_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A88C	Target Register (CCM_TARGET_ROOT81_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A890	Miscellaneous Register (CCM_MISC81)	32	R/W	0000_0000h	5.1.7.14/469
3038_A894	Miscellaneous Register (CCM_MISC_ROOT81_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A898	Miscellaneous Register (CCM_MISC_ROOT81_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A89C	Miscellaneous Register (CCM_MISC_ROOT81_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A8A0	Post Divider Register (CCM_POST81)	32	R/W	0000_0000h	5.1.7.18/473
3038_A8A4	Post Divider Register (CCM_POST_ROOT81_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A8A8	Post Divider Register (CCM_POST_ROOT81_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A8AC	Post Divider Register (CCM_POST_ROOT81_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A8B0	Pre Divider Register (CCM_PRE81)	32	R/W	1000_0000h	5.1.7.22/485
3038_A8B4	Pre Divider Register (CCM_PRE_ROOT81_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A8B8	Pre Divider Register (CCM_PRE_ROOT81_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A8BC	Pre Divider Register (CCM_PRE_ROOT81_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A8F0	Access Control Register (CCM_ACCESS_CTRL81)	32	R/W	0000_0000h	5.1.7.26/497
3038_A8F4	Access Control Register (CCM_ACCESS_CTRL_ROOT81_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A8F8	Access Control Register (CCM_ACCESS_CTRL_ROOT81_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A8FC	Access Control Register (CCM_ACCESS_CTRL_ROOT81_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A900	Target Register (CCM_TARGET_ROOT82)	32	R/W	1000_0000h	5.1.7.10/461
3038_A904	Target Register (CCM_TARGET_ROOT82_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A908	Target Register (CCM_TARGET_ROOT82_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A90C	Target Register (CCM_TARGET_ROOT82_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A910	Miscellaneous Register (CCM_MISC82)	32	R/W	0000_0000h	5.1.7.14/469
3038_A914	Miscellaneous Register (CCM_MISC_ROOT82_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A918	Miscellaneous Register (CCM_MISC_ROOT82_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A91C	Miscellaneous Register (CCM_MISC_ROOT82_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A920	Post Divider Register (CCM_POST82)	32	R/W	0000_0000h	5.1.7.18/473
3038_A924	Post Divider Register (CCM_POST_ROOT82_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A928	Post Divider Register (CCM_POST_ROOT82_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A92C	Post Divider Register (CCM_POST_ROOT82_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A930	Pre Divider Register (CCM_PRE82)	32	R/W	1000_0000h	5.1.7.22/485
3038_A934	Pre Divider Register (CCM_PRE_ROOT82_SET)	32	R/W	0000_0000h	5.1.7.23/488

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A938	Pre Divider Register (CCM_PRE_ROOT82_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A93C	Pre Divider Register (CCM_PRE_ROOT82_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A970	Access Control Register (CCM_ACCESS_CTRL82)	32	R/W	0000_0000h	5.1.7.26/497
3038_A974	Access Control Register (CCM_ACCESS_CTRL_ROOT82_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A978	Access Control Register (CCM_ACCESS_CTRL_ROOT82_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A97C	Access Control Register (CCM_ACCESS_CTRL_ROOT82_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_A980	Target Register (CCM_TARGET_ROOT83)	32	R/W	1000_0000h	5.1.7.10/461
3038_A984	Target Register (CCM_TARGET_ROOT83_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_A988	Target Register (CCM_TARGET_ROOT83_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_A98C	Target Register (CCM_TARGET_ROOT83_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_A990	Miscellaneous Register (CCM_MISC83)	32	R/W	0000_0000h	5.1.7.14/469
3038_A994	Miscellaneous Register (CCM_MISC_ROOT83_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_A998	Miscellaneous Register (CCM_MISC_ROOT83_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_A99C	Miscellaneous Register (CCM_MISC_ROOT83_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_A9A0	Post Divider Register (CCM_POST83)	32	R/W	0000_0000h	5.1.7.18/473
3038_A9A4	Post Divider Register (CCM_POST_ROOT83_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_A9A8	Post Divider Register (CCM_POST_ROOT83_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_A9AC	Post Divider Register (CCM_POST_ROOT83_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_A9B0	Pre Divider Register (CCM_PRE83)	32	R/W	1000_0000h	5.1.7.22/485
3038_A9B4	Pre Divider Register (CCM_PRE_ROOT83_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_A9B8	Pre Divider Register (CCM_PRE_ROOT83_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_A9BC	Pre Divider Register (CCM_PRE_ROOT83_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_A9F0	Access Control Register (CCM_ACCESS_CTRL83)	32	R/W	0000_0000h	5.1.7.26/497
3038_A9F4	Access Control Register (CCM_ACCESS_CTRL_ROOT83_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_A9F8	Access Control Register (CCM_ACCESS_CTRL_ROOT83_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_A9FC	Access Control Register (CCM_ACCESS_CTRL_ROOT83_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_AA00	Target Register (CCM_TARGET_ROOT84)	32	R/W	1000_0000h	5.1.7.10/461
3038_AA04	Target Register (CCM_TARGET_ROOT84_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_AA08	Target Register (CCM_TARGET_ROOT84_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_AA0C	Target Register (CCM_TARGET_ROOT84_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_AA10	Miscellaneous Register (CCM_MISC84)	32	R/W	0000_0000h	5.1.7.14/469
3038_AA14	Miscellaneous Register (CCM_MISC_ROOT84_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_AA18	Miscellaneous Register (CCM_MISC_ROOT84_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_AA1C	Miscellaneous Register (CCM_MISC_ROOT84_TOG)	32	R/W	0000_0000h	5.1.7.17/472

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AA20	Post Divider Register (CCM_POST84)	32	R/W	0000_0000h	5.1.7.18/473
3038_AA24	Post Divider Register (CCM_POST_ROOT84_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_AA28	Post Divider Register (CCM_POST_ROOT84_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_AA2C	Post Divider Register (CCM_POST_ROOT84_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_AA30	Pre Divider Register (CCM_PRE84)	32	R/W	1000_0000h	5.1.7.22/485
3038_AA34	Pre Divider Register (CCM_PRE_ROOT84_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_AA38	Pre Divider Register (CCM_PRE_ROOT84_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_AA3C	Pre Divider Register (CCM_PRE_ROOT84_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_AA70	Access Control Register (CCM_ACCESS_CTRL84)	32	R/W	0000_0000h	5.1.7.26/497
3038_AA74	Access Control Register (CCM_ACCESS_CTRL_ROOT84_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_AA78	Access Control Register (CCM_ACCESS_CTRL_ROOT84_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_AA7C	Access Control Register (CCM_ACCESS_CTRL_ROOT84_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_AA80	Target Register (CCM_TARGET_ROOT85)	32	R/W	1000_0000h	5.1.7.10/461
3038_AA84	Target Register (CCM_TARGET_ROOT85_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_AA88	Target Register (CCM_TARGET_ROOT85_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_AA8C	Target Register (CCM_TARGET_ROOT85_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_AA90	Miscellaneous Register (CCM_MISC85)	32	R/W	0000_0000h	5.1.7.14/469
3038_AA94	Miscellaneous Register (CCM_MISC_ROOT85_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_AA98	Miscellaneous Register (CCM_MISC_ROOT85_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_AA9C	Miscellaneous Register (CCM_MISC_ROOT85_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_AAA0	Post Divider Register (CCM_POST85)	32	R/W	0000_0000h	5.1.7.18/473
3038_AAA4	Post Divider Register (CCM_POST_ROOT85_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_AAA8	Post Divider Register (CCM_POST_ROOT85_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_AAAC	Post Divider Register (CCM_POST_ROOT85_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_AAB0	Pre Divider Register (CCM_PRE85)	32	R/W	1000_0000h	5.1.7.22/485
3038_AAB4	Pre Divider Register (CCM_PRE_ROOT85_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_AAB8	Pre Divider Register (CCM_PRE_ROOT85_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_AABC	Pre Divider Register (CCM_PRE_ROOT85_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_AAF0	Access Control Register (CCM_ACCESS_CTRL85)	32	R/W	0000_0000h	5.1.7.26/497
3038_AAF4	Access Control Register (CCM_ACCESS_CTRL_ROOT85_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_AAF8	Access Control Register (CCM_ACCESS_CTRL_ROOT85_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_AAFC	Access Control Register (CCM_ACCESS_CTRL_ROOT85_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_AB00	Target Register (CCM_TARGET_ROOT86)	32	R/W	1000_0000h	5.1.7.10/461
3038_AB04	Target Register (CCM_TARGET_ROOT86_SET)	32	R/W	0000_0000h	5.1.7.11/463

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AB08	Target Register (CCM_TARGET_ROOT86_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_AB0C	Target Register (CCM_TARGET_ROOT86_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_AB10	Miscellaneous Register (CCM_MISC86)	32	R/W	0000_0000h	5.1.7.14/469
3038_AB14	Miscellaneous Register (CCM_MISC_ROOT86_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_AB18	Miscellaneous Register (CCM_MISC_ROOT86_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_AB1C	Miscellaneous Register (CCM_MISC_ROOT86_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_AB20	Post Divider Register (CCM_POST86)	32	R/W	0000_0000h	5.1.7.18/473
3038_AB24	Post Divider Register (CCM_POST_ROOT86_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_AB28	Post Divider Register (CCM_POST_ROOT86_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_AB2C	Post Divider Register (CCM_POST_ROOT86_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_AB30	Pre Divider Register (CCM_PRE86)	32	R/W	1000_0000h	5.1.7.22/485
3038_AB34	Pre Divider Register (CCM_PRE_ROOT86_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_AB38	Pre Divider Register (CCM_PRE_ROOT86_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_AB3C	Pre Divider Register (CCM_PRE_ROOT86_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_AB70	Access Control Register (CCM_ACCESS_CTRL86)	32	R/W	0000_0000h	5.1.7.26/497
3038_AB74	Access Control Register (CCM_ACCESS_CTRL_ROOT86_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_AB78	Access Control Register (CCM_ACCESS_CTRL_ROOT86_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_AB7C	Access Control Register (CCM_ACCESS_CTRL_ROOT86_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_AB80	Target Register (CCM_TARGET_ROOT87)	32	R/W	1000_0000h	5.1.7.10/461
3038_AB84	Target Register (CCM_TARGET_ROOT87_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_AB88	Target Register (CCM_TARGET_ROOT87_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_AB8C	Target Register (CCM_TARGET_ROOT87_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_AB90	Miscellaneous Register (CCM_MISC87)	32	R/W	0000_0000h	5.1.7.14/469
3038_AB94	Miscellaneous Register (CCM_MISC_ROOT87_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_AB98	Miscellaneous Register (CCM_MISC_ROOT87_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_AB9C	Miscellaneous Register (CCM_MISC_ROOT87_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_ABA0	Post Divider Register (CCM_POST87)	32	R/W	0000_0000h	5.1.7.18/473
3038_ABA4	Post Divider Register (CCM_POST_ROOT87_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_ABA8	Post Divider Register (CCM_POST_ROOT87_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_ABAC	Post Divider Register (CCM_POST_ROOT87_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_ABB0	Pre Divider Register (CCM_PRE87)	32	R/W	1000_0000h	5.1.7.22/485
3038_ABB4	Pre Divider Register (CCM_PRE_ROOT87_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_ABB8	Pre Divider Register (CCM_PRE_ROOT87_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_ABBC	Pre Divider Register (CCM_PRE_ROOT87_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_ABF0	Access Control Register (CCM_ACCESS_CTRL87)	32	R/W	0000_0000h	5.1.7.26/497

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_ABF4	Access Control Register (CCM_ACCESS_CTRL_ROOT87_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_ABF8	Access Control Register (CCM_ACCESS_CTRL_ROOT87_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_ABFC	Access Control Register (CCM_ACCESS_CTRL_ROOT87_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_AC00	Target Register (CCM_TARGET_ROOT88)	32	R/W	1000_0000h	5.1.7.10/461
3038_AC04	Target Register (CCM_TARGET_ROOT88_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_AC08	Target Register (CCM_TARGET_ROOT88_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_AC0C	Target Register (CCM_TARGET_ROOT88_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_AC10	Miscellaneous Register (CCM_MISC88)	32	R/W	0000_0000h	5.1.7.14/469
3038_AC14	Miscellaneous Register (CCM_MISC_ROOT88_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_AC18	Miscellaneous Register (CCM_MISC_ROOT88_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_AC1C	Miscellaneous Register (CCM_MISC_ROOT88_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_AC20	Post Divider Register (CCM_POST88)	32	R/W	0000_0000h	5.1.7.18/473
3038_AC24	Post Divider Register (CCM_POST_ROOT88_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_AC28	Post Divider Register (CCM_POST_ROOT88_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_AC2C	Post Divider Register (CCM_POST_ROOT88_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_AC30	Pre Divider Register (CCM_PRE88)	32	R/W	1000_0000h	5.1.7.22/485
3038_AC34	Pre Divider Register (CCM_PRE_ROOT88_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_AC38	Pre Divider Register (CCM_PRE_ROOT88_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_AC3C	Pre Divider Register (CCM_PRE_ROOT88_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_AC70	Access Control Register (CCM_ACCESS_CTRL88)	32	R/W	0000_0000h	5.1.7.26/497
3038_AC74	Access Control Register (CCM_ACCESS_CTRL_ROOT88_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_AC78	Access Control Register (CCM_ACCESS_CTRL_ROOT88_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_AC7C	Access Control Register (CCM_ACCESS_CTRL_ROOT88_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_AC80	Target Register (CCM_TARGET_ROOT89)	32	R/W	1000_0000h	5.1.7.10/461
3038_AC84	Target Register (CCM_TARGET_ROOT89_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_AC88	Target Register (CCM_TARGET_ROOT89_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_AC8C	Target Register (CCM_TARGET_ROOT89_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_AC90	Miscellaneous Register (CCM_MISC89)	32	R/W	0000_0000h	5.1.7.14/469
3038_AC94	Miscellaneous Register (CCM_MISC_ROOT89_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_AC98	Miscellaneous Register (CCM_MISC_ROOT89_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_AC9C	Miscellaneous Register (CCM_MISC_ROOT89_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_ACA0	Post Divider Register (CCM_POST89)	32	R/W	0000_0000h	5.1.7.18/473
3038_ACA4	Post Divider Register (CCM_POST_ROOT89_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_ACA8	Post Divider Register (CCM_POST_ROOT89_CLR)	32	R/W	0000_0000h	5.1.7.20/479

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_ACAC	Post Divider Register (CCM_POST_ROOT89_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_ACB0	Pre Divider Register (CCM_PRE89)	32	R/W	1000_0000h	5.1.7.22/485
3038_ACB4	Pre Divider Register (CCM_PRE_ROOT89_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_ACB8	Pre Divider Register (CCM_PRE_ROOT89_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_ACBC	Pre Divider Register (CCM_PRE_ROOT89_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_ACF0	Access Control Register (CCM_ACCESS_CTRL89)	32	R/W	0000_0000h	5.1.7.26/497
3038_ACF4	Access Control Register (CCM_ACCESS_CTRL_ROOT89_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_ACF8	Access Control Register (CCM_ACCESS_CTRL_ROOT89_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_ACFC	Access Control Register (CCM_ACCESS_CTRL_ROOT89_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_AD00	Target Register (CCM_TARGET_ROOT90)	32	R/W	1000_0000h	5.1.7.10/461
3038_AD04	Target Register (CCM_TARGET_ROOT90_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_AD08	Target Register (CCM_TARGET_ROOT90_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_AD0C	Target Register (CCM_TARGET_ROOT90_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_AD10	Miscellaneous Register (CCM_MISC90)	32	R/W	0000_0000h	5.1.7.14/469
3038_AD14	Miscellaneous Register (CCM_MISC_ROOT90_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_AD18	Miscellaneous Register (CCM_MISC_ROOT90_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_AD1C	Miscellaneous Register (CCM_MISC_ROOT90_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_AD20	Post Divider Register (CCM_POST90)	32	R/W	0000_0000h	5.1.7.18/473
3038_AD24	Post Divider Register (CCM_POST_ROOT90_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_AD28	Post Divider Register (CCM_POST_ROOT90_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_AD2C	Post Divider Register (CCM_POST_ROOT90_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_AD30	Pre Divider Register (CCM_PRE90)	32	R/W	1000_0000h	5.1.7.22/485
3038_AD34	Pre Divider Register (CCM_PRE_ROOT90_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_AD38	Pre Divider Register (CCM_PRE_ROOT90_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_AD3C	Pre Divider Register (CCM_PRE_ROOT90_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_AD70	Access Control Register (CCM_ACCESS_CTRL90)	32	R/W	0000_0000h	5.1.7.26/497
3038_AD74	Access Control Register (CCM_ACCESS_CTRL_ROOT90_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_AD78	Access Control Register (CCM_ACCESS_CTRL_ROOT90_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_AD7C	Access Control Register (CCM_ACCESS_CTRL_ROOT90_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_AD80	Target Register (CCM_TARGET_ROOT91)	32	R/W	1000_0000h	5.1.7.10/461
3038_AD84	Target Register (CCM_TARGET_ROOT91_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_AD88	Target Register (CCM_TARGET_ROOT91_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_AD8C	Target Register (CCM_TARGET_ROOT91_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_AD90	Miscellaneous Register (CCM_MISC91)	32	R/W	0000_0000h	5.1.7.14/469

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AD94	Miscellaneous Register (CCM_MISC_ROOT91_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_AD98	Miscellaneous Register (CCM_MISC_ROOT91_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_AD9C	Miscellaneous Register (CCM_MISC_ROOT91_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_ADA0	Post Divider Register (CCM_POST91)	32	R/W	0000_0000h	5.1.7.18/473
3038_ADA4	Post Divider Register (CCM_POST_ROOT91_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_ADA8	Post Divider Register (CCM_POST_ROOT91_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_ADAC	Post Divider Register (CCM_POST_ROOT91_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_ADB0	Pre Divider Register (CCM_PRE91)	32	R/W	1000_0000h	5.1.7.22/485
3038_ADB4	Pre Divider Register (CCM_PRE_ROOT91_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_ADB8	Pre Divider Register (CCM_PRE_ROOT91_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_ADBC	Pre Divider Register (CCM_PRE_ROOT91_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_ADF0	Access Control Register (CCM_ACCESS_CTRL91)	32	R/W	0000_0000h	5.1.7.26/497
3038_ADF4	Access Control Register (CCM_ACCESS_CTRL_ROOT91_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_ADF8	Access Control Register (CCM_ACCESS_CTRL_ROOT91_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_ADFC	Access Control Register (CCM_ACCESS_CTRL_ROOT91_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_AE00	Target Register (CCM_TARGET_ROOT92)	32	R/W	1000_0000h	5.1.7.10/461
3038_AE04	Target Register (CCM_TARGET_ROOT92_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_AE08	Target Register (CCM_TARGET_ROOT92_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_AE0C	Target Register (CCM_TARGET_ROOT92_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_AE10	Miscellaneous Register (CCM_MISC92)	32	R/W	0000_0000h	5.1.7.14/469
3038_AE14	Miscellaneous Register (CCM_MISC_ROOT92_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_AE18	Miscellaneous Register (CCM_MISC_ROOT92_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_AE1C	Miscellaneous Register (CCM_MISC_ROOT92_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_AE20	Post Divider Register (CCM_POST92)	32	R/W	0000_0000h	5.1.7.18/473
3038_AE24	Post Divider Register (CCM_POST_ROOT92_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_AE28	Post Divider Register (CCM_POST_ROOT92_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_AE2C	Post Divider Register (CCM_POST_ROOT92_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_AE30	Pre Divider Register (CCM_PRE92)	32	R/W	1000_0000h	5.1.7.22/485
3038_AE34	Pre Divider Register (CCM_PRE_ROOT92_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_AE38	Pre Divider Register (CCM_PRE_ROOT92_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_AE3C	Pre Divider Register (CCM_PRE_ROOT92_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_AE70	Access Control Register (CCM_ACCESS_CTRL92)	32	R/W	0000_0000h	5.1.7.26/497
3038_AE74	Access Control Register (CCM_ACCESS_CTRL_ROOT92_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_AE78	Access Control Register (CCM_ACCESS_CTRL_ROOT92_CLR)	32	R/W	0000_0000h	5.1.7.28/502

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AE7C	Access Control Register (CCM_ACCESS_CTRL_ROOT92_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_AE80	Target Register (CCM_TARGET_ROOT93)	32	R/W	1000_0000h	5.1.7.10/461
3038_AE84	Target Register (CCM_TARGET_ROOT93_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_AE88	Target Register (CCM_TARGET_ROOT93_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_AE8C	Target Register (CCM_TARGET_ROOT93_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_AE90	Miscellaneous Register (CCM_MISC93)	32	R/W	0000_0000h	5.1.7.14/469
3038_AE94	Miscellaneous Register (CCM_MISC_ROOT93_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_AE98	Miscellaneous Register (CCM_MISC_ROOT93_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_AE9C	Miscellaneous Register (CCM_MISC_ROOT93_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_AEA0	Post Divider Register (CCM_POST93)	32	R/W	0000_0000h	5.1.7.18/473
3038_AEA4	Post Divider Register (CCM_POST_ROOT93_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_AEA8	Post Divider Register (CCM_POST_ROOT93_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_AEAC	Post Divider Register (CCM_POST_ROOT93_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_AEB0	Pre Divider Register (CCM_PRE93)	32	R/W	1000_0000h	5.1.7.22/485
3038_AEB4	Pre Divider Register (CCM_PRE_ROOT93_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_AEB8	Pre Divider Register (CCM_PRE_ROOT93_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_AEBC	Pre Divider Register (CCM_PRE_ROOT93_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_AEF0	Access Control Register (CCM_ACCESS_CTRL93)	32	R/W	0000_0000h	5.1.7.26/497
3038_AEF4	Access Control Register (CCM_ACCESS_CTRL_ROOT93_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_AEF8	Access Control Register (CCM_ACCESS_CTRL_ROOT93_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_AEFC	Access Control Register (CCM_ACCESS_CTRL_ROOT93_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_AF00	Target Register (CCM_TARGET_ROOT94)	32	R/W	1000_0000h	5.1.7.10/461
3038_AF04	Target Register (CCM_TARGET_ROOT94_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_AF08	Target Register (CCM_TARGET_ROOT94_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_AF0C	Target Register (CCM_TARGET_ROOT94_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_AF10	Miscellaneous Register (CCM_MISC94)	32	R/W	0000_0000h	5.1.7.14/469
3038_AF14	Miscellaneous Register (CCM_MISC_ROOT94_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_AF18	Miscellaneous Register (CCM_MISC_ROOT94_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_AF1C	Miscellaneous Register (CCM_MISC_ROOT94_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_AF20	Post Divider Register (CCM_POST94)	32	R/W	0000_0000h	5.1.7.18/473
3038_AF24	Post Divider Register (CCM_POST_ROOT94_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_AF28	Post Divider Register (CCM_POST_ROOT94_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_AF2C	Post Divider Register (CCM_POST_ROOT94_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_AF30	Pre Divider Register (CCM_PRE94)	32	R/W	1000_0000h	5.1.7.22/485
3038_AF34	Pre Divider Register (CCM_PRE_ROOT94_SET)	32	R/W	0000_0000h	5.1.7.23/488

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AF38	Pre Divider Register (CCM_PRE_ROOT94_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_AF3C	Pre Divider Register (CCM_PRE_ROOT94_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_AF70	Access Control Register (CCM_ACCESS_CTRL94)	32	R/W	0000_0000h	5.1.7.26/497
3038_AF74	Access Control Register (CCM_ACCESS_CTRL_ROOT94_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_AF78	Access Control Register (CCM_ACCESS_CTRL_ROOT94_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_AF7C	Access Control Register (CCM_ACCESS_CTRL_ROOT94_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_AF80	Target Register (CCM_TARGET_ROOT95)	32	R/W	1000_0000h	5.1.7.10/461
3038_AF84	Target Register (CCM_TARGET_ROOT95_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_AF88	Target Register (CCM_TARGET_ROOT95_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_AF8C	Target Register (CCM_TARGET_ROOT95_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_AF90	Miscellaneous Register (CCM_MISC95)	32	R/W	0000_0000h	5.1.7.14/469
3038_AF94	Miscellaneous Register (CCM_MISC_ROOT95_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_AF98	Miscellaneous Register (CCM_MISC_ROOT95_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_AF9C	Miscellaneous Register (CCM_MISC_ROOT95_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_AFA0	Post Divider Register (CCM_POST95)	32	R/W	0000_0000h	5.1.7.18/473
3038_AFA4	Post Divider Register (CCM_POST_ROOT95_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_AFA8	Post Divider Register (CCM_POST_ROOT95_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_AFAC	Post Divider Register (CCM_POST_ROOT95_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_AFB0	Pre Divider Register (CCM_PRE95)	32	R/W	1000_0000h	5.1.7.22/485
3038_AFB4	Pre Divider Register (CCM_PRE_ROOT95_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_AFB8	Pre Divider Register (CCM_PRE_ROOT95_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_AFBC	Pre Divider Register (CCM_PRE_ROOT95_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_AFF0	Access Control Register (CCM_ACCESS_CTRL95)	32	R/W	0000_0000h	5.1.7.26/497
3038_AFF4	Access Control Register (CCM_ACCESS_CTRL_ROOT95_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_AFF8	Access Control Register (CCM_ACCESS_CTRL_ROOT95_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_AFFC	Access Control Register (CCM_ACCESS_CTRL_ROOT95_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B000	Target Register (CCM_TARGET_ROOT96)	32	R/W	1000_0000h	5.1.7.10/461
3038_B004	Target Register (CCM_TARGET_ROOT96_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B008	Target Register (CCM_TARGET_ROOT96_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B00C	Target Register (CCM_TARGET_ROOT96_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B010	Miscellaneous Register (CCM_MISC96)	32	R/W	0000_0000h	5.1.7.14/469
3038_B014	Miscellaneous Register (CCM_MISC_ROOT96_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B018	Miscellaneous Register (CCM_MISC_ROOT96_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B01C	Miscellaneous Register (CCM_MISC_ROOT96_TOG)	32	R/W	0000_0000h	5.1.7.17/472

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B020	Post Divider Register (CCM_POST96)	32	R/W	0000_0000h	5.1.7.18/473
3038_B024	Post Divider Register (CCM_POST_ROOT96_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B028	Post Divider Register (CCM_POST_ROOT96_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B02C	Post Divider Register (CCM_POST_ROOT96_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B030	Pre Divider Register (CCM_PRE96)	32	R/W	1000_0000h	5.1.7.22/485
3038_B034	Pre Divider Register (CCM_PRE_ROOT96_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B038	Pre Divider Register (CCM_PRE_ROOT96_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B03C	Pre Divider Register (CCM_PRE_ROOT96_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B070	Access Control Register (CCM_ACCESS_CTRL96)	32	R/W	0000_0000h	5.1.7.26/497
3038_B074	Access Control Register (CCM_ACCESS_CTRL_ROOT96_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B078	Access Control Register (CCM_ACCESS_CTRL_ROOT96_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B07C	Access Control Register (CCM_ACCESS_CTRL_ROOT96_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B080	Target Register (CCM_TARGET_ROOT97)	32	R/W	1000_0000h	5.1.7.10/461
3038_B084	Target Register (CCM_TARGET_ROOT97_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B088	Target Register (CCM_TARGET_ROOT97_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B08C	Target Register (CCM_TARGET_ROOT97_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B090	Miscellaneous Register (CCM_MISC97)	32	R/W	0000_0000h	5.1.7.14/469
3038_B094	Miscellaneous Register (CCM_MISC_ROOT97_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B098	Miscellaneous Register (CCM_MISC_ROOT97_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B09C	Miscellaneous Register (CCM_MISC_ROOT97_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B0A0	Post Divider Register (CCM_POST97)	32	R/W	0000_0000h	5.1.7.18/473
3038_B0A4	Post Divider Register (CCM_POST_ROOT97_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B0A8	Post Divider Register (CCM_POST_ROOT97_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B0AC	Post Divider Register (CCM_POST_ROOT97_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B0B0	Pre Divider Register (CCM_PRE97)	32	R/W	1000_0000h	5.1.7.22/485
3038_B0B4	Pre Divider Register (CCM_PRE_ROOT97_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B0B8	Pre Divider Register (CCM_PRE_ROOT97_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B0BC	Pre Divider Register (CCM_PRE_ROOT97_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B0F0	Access Control Register (CCM_ACCESS_CTRL97)	32	R/W	0000_0000h	5.1.7.26/497
3038_B0F4	Access Control Register (CCM_ACCESS_CTRL_ROOT97_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B0F8	Access Control Register (CCM_ACCESS_CTRL_ROOT97_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B0FC	Access Control Register (CCM_ACCESS_CTRL_ROOT97_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B100	Target Register (CCM_TARGET_ROOT98)	32	R/W	1000_0000h	5.1.7.10/461
3038_B104	Target Register (CCM_TARGET_ROOT98_SET)	32	R/W	0000_0000h	5.1.7.11/463

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B108	Target Register (CCM_TARGET_ROOT98_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B10C	Target Register (CCM_TARGET_ROOT98_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B110	Miscellaneous Register (CCM_MISC98)	32	R/W	0000_0000h	5.1.7.14/469
3038_B114	Miscellaneous Register (CCM_MISC_ROOT98_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B118	Miscellaneous Register (CCM_MISC_ROOT98_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B11C	Miscellaneous Register (CCM_MISC_ROOT98_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B120	Post Divider Register (CCM_POST98)	32	R/W	0000_0000h	5.1.7.18/473
3038_B124	Post Divider Register (CCM_POST_ROOT98_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B128	Post Divider Register (CCM_POST_ROOT98_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B12C	Post Divider Register (CCM_POST_ROOT98_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B130	Pre Divider Register (CCM_PRE98)	32	R/W	1000_0000h	5.1.7.22/485
3038_B134	Pre Divider Register (CCM_PRE_ROOT98_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B138	Pre Divider Register (CCM_PRE_ROOT98_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B13C	Pre Divider Register (CCM_PRE_ROOT98_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B170	Access Control Register (CCM_ACCESS_CTRL98)	32	R/W	0000_0000h	5.1.7.26/497
3038_B174	Access Control Register (CCM_ACCESS_CTRL_ROOT98_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B178	Access Control Register (CCM_ACCESS_CTRL_ROOT98_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B17C	Access Control Register (CCM_ACCESS_CTRL_ROOT98_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B180	Target Register (CCM_TARGET_ROOT99)	32	R/W	1000_0000h	5.1.7.10/461
3038_B184	Target Register (CCM_TARGET_ROOT99_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B188	Target Register (CCM_TARGET_ROOT99_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B18C	Target Register (CCM_TARGET_ROOT99_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B190	Miscellaneous Register (CCM_MISC99)	32	R/W	0000_0000h	5.1.7.14/469
3038_B194	Miscellaneous Register (CCM_MISC_ROOT99_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B198	Miscellaneous Register (CCM_MISC_ROOT99_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B19C	Miscellaneous Register (CCM_MISC_ROOT99_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B1A0	Post Divider Register (CCM_POST99)	32	R/W	0000_0000h	5.1.7.18/473
3038_B1A4	Post Divider Register (CCM_POST_ROOT99_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B1A8	Post Divider Register (CCM_POST_ROOT99_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B1AC	Post Divider Register (CCM_POST_ROOT99_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B1B0	Pre Divider Register (CCM_PRE99)	32	R/W	1000_0000h	5.1.7.22/485
3038_B1B4	Pre Divider Register (CCM_PRE_ROOT99_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B1B8	Pre Divider Register (CCM_PRE_ROOT99_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B1BC	Pre Divider Register (CCM_PRE_ROOT99_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B1F0	Access Control Register (CCM_ACCESS_CTRL99)	32	R/W	0000_0000h	5.1.7.26/497

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B1F4	Access Control Register (CCM_ACCESS_CTRL_ROOT99_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B1F8	Access Control Register (CCM_ACCESS_CTRL_ROOT99_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B1FC	Access Control Register (CCM_ACCESS_CTRL_ROOT99_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B200	Target Register (CCM_TARGET_ROOT100)	32	R/W	1000_0000h	5.1.7.10/461
3038_B204	Target Register (CCM_TARGET_ROOT100_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B208	Target Register (CCM_TARGET_ROOT100_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B20C	Target Register (CCM_TARGET_ROOT100_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B210	Miscellaneous Register (CCM_MISC100)	32	R/W	0000_0000h	5.1.7.14/469
3038_B214	Miscellaneous Register (CCM_MISC_ROOT100_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B218	Miscellaneous Register (CCM_MISC_ROOT100_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B21C	Miscellaneous Register (CCM_MISC_ROOT100_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B220	Post Divider Register (CCM_POST100)	32	R/W	0000_0000h	5.1.7.18/473
3038_B224	Post Divider Register (CCM_POST_ROOT100_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B228	Post Divider Register (CCM_POST_ROOT100_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B22C	Post Divider Register (CCM_POST_ROOT100_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B230	Pre Divider Register (CCM_PRE100)	32	R/W	1000_0000h	5.1.7.22/485
3038_B234	Pre Divider Register (CCM_PRE_ROOT100_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B238	Pre Divider Register (CCM_PRE_ROOT100_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B23C	Pre Divider Register (CCM_PRE_ROOT100_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B270	Access Control Register (CCM_ACCESS_CTRL100)	32	R/W	0000_0000h	5.1.7.26/497
3038_B274	Access Control Register (CCM_ACCESS_CTRL_ROOT100_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B278	Access Control Register (CCM_ACCESS_CTRL_ROOT100_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B27C	Access Control Register (CCM_ACCESS_CTRL_ROOT100_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B280	Target Register (CCM_TARGET_ROOT101)	32	R/W	1000_0000h	5.1.7.10/461
3038_B284	Target Register (CCM_TARGET_ROOT101_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B288	Target Register (CCM_TARGET_ROOT101_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B28C	Target Register (CCM_TARGET_ROOT101_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B290	Miscellaneous Register (CCM_MISC101)	32	R/W	0000_0000h	5.1.7.14/469
3038_B294	Miscellaneous Register (CCM_MISC_ROOT101_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B298	Miscellaneous Register (CCM_MISC_ROOT101_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B29C	Miscellaneous Register (CCM_MISC_ROOT101_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B2A0	Post Divider Register (CCM_POST101)	32	R/W	0000_0000h	5.1.7.18/473
3038_B2A4	Post Divider Register (CCM_POST_ROOT101_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B2A8	Post Divider Register (CCM_POST_ROOT101_CLR)	32	R/W	0000_0000h	5.1.7.20/479

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B2AC	Post Divider Register (CCM_POST_ROOT101_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B2B0	Pre Divider Register (CCM_PRE101)	32	R/W	1000_0000h	5.1.7.22/485
3038_B2B4	Pre Divider Register (CCM_PRE_ROOT101_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B2B8	Pre Divider Register (CCM_PRE_ROOT101_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B2BC	Pre Divider Register (CCM_PRE_ROOT101_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B2F0	Access Control Register (CCM_ACCESS_CTRL101)	32	R/W	0000_0000h	5.1.7.26/497
3038_B2F4	Access Control Register (CCM_ACCESS_CTRL_ROOT101_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B2F8	Access Control Register (CCM_ACCESS_CTRL_ROOT101_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B2FC	Access Control Register (CCM_ACCESS_CTRL_ROOT101_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B300	Target Register (CCM_TARGET_ROOT102)	32	R/W	1000_0000h	5.1.7.10/461
3038_B304	Target Register (CCM_TARGET_ROOT102_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B308	Target Register (CCM_TARGET_ROOT102_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B30C	Target Register (CCM_TARGET_ROOT102_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B310	Miscellaneous Register (CCM_MISC102)	32	R/W	0000_0000h	5.1.7.14/469
3038_B314	Miscellaneous Register (CCM_MISC_ROOT102_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B318	Miscellaneous Register (CCM_MISC_ROOT102_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B31C	Miscellaneous Register (CCM_MISC_ROOT102_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B320	Post Divider Register (CCM_POST102)	32	R/W	0000_0000h	5.1.7.18/473
3038_B324	Post Divider Register (CCM_POST_ROOT102_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B328	Post Divider Register (CCM_POST_ROOT102_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B32C	Post Divider Register (CCM_POST_ROOT102_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B330	Pre Divider Register (CCM_PRE102)	32	R/W	1000_0000h	5.1.7.22/485
3038_B334	Pre Divider Register (CCM_PRE_ROOT102_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B338	Pre Divider Register (CCM_PRE_ROOT102_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B33C	Pre Divider Register (CCM_PRE_ROOT102_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B370	Access Control Register (CCM_ACCESS_CTRL102)	32	R/W	0000_0000h	5.1.7.26/497
3038_B374	Access Control Register (CCM_ACCESS_CTRL_ROOT102_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B378	Access Control Register (CCM_ACCESS_CTRL_ROOT102_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B37C	Access Control Register (CCM_ACCESS_CTRL_ROOT102_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B380	Target Register (CCM_TARGET_ROOT103)	32	R/W	1000_0000h	5.1.7.10/461
3038_B384	Target Register (CCM_TARGET_ROOT103_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B388	Target Register (CCM_TARGET_ROOT103_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B38C	Target Register (CCM_TARGET_ROOT103_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B390	Miscellaneous Register (CCM_MISC103)	32	R/W	0000_0000h	5.1.7.14/469

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B394	Miscellaneous Register (CCM_MISC_ROOT103_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B398	Miscellaneous Register (CCM_MISC_ROOT103_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B39C	Miscellaneous Register (CCM_MISC_ROOT103_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B3A0	Post Divider Register (CCM_POST103)	32	R/W	0000_0000h	5.1.7.18/473
3038_B3A4	Post Divider Register (CCM_POST_ROOT103_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B3A8	Post Divider Register (CCM_POST_ROOT103_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B3AC	Post Divider Register (CCM_POST_ROOT103_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B3B0	Pre Divider Register (CCM_PRE103)	32	R/W	1000_0000h	5.1.7.22/485
3038_B3B4	Pre Divider Register (CCM_PRE_ROOT103_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B3B8	Pre Divider Register (CCM_PRE_ROOT103_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B3BC	Pre Divider Register (CCM_PRE_ROOT103_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B3F0	Access Control Register (CCM_ACCESS_CTRL103)	32	R/W	0000_0000h	5.1.7.26/497
3038_B3F4	Access Control Register (CCM_ACCESS_CTRL_ROOT103_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B3F8	Access Control Register (CCM_ACCESS_CTRL_ROOT103_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B3FC	Access Control Register (CCM_ACCESS_CTRL_ROOT103_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B400	Target Register (CCM_TARGET_ROOT104)	32	R/W	1000_0000h	5.1.7.10/461
3038_B404	Target Register (CCM_TARGET_ROOT104_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B408	Target Register (CCM_TARGET_ROOT104_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B40C	Target Register (CCM_TARGET_ROOT104_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B410	Miscellaneous Register (CCM_MISC104)	32	R/W	0000_0000h	5.1.7.14/469
3038_B414	Miscellaneous Register (CCM_MISC_ROOT104_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B418	Miscellaneous Register (CCM_MISC_ROOT104_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B41C	Miscellaneous Register (CCM_MISC_ROOT104_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B420	Post Divider Register (CCM_POST104)	32	R/W	0000_0000h	5.1.7.18/473
3038_B424	Post Divider Register (CCM_POST_ROOT104_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B428	Post Divider Register (CCM_POST_ROOT104_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B42C	Post Divider Register (CCM_POST_ROOT104_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B430	Pre Divider Register (CCM_PRE104)	32	R/W	1000_0000h	5.1.7.22/485
3038_B434	Pre Divider Register (CCM_PRE_ROOT104_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B438	Pre Divider Register (CCM_PRE_ROOT104_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B43C	Pre Divider Register (CCM_PRE_ROOT104_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B470	Access Control Register (CCM_ACCESS_CTRL104)	32	R/W	0000_0000h	5.1.7.26/497
3038_B474	Access Control Register (CCM_ACCESS_CTRL_ROOT104_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B478	Access Control Register (CCM_ACCESS_CTRL_ROOT104_CLR)	32	R/W	0000_0000h	5.1.7.28/502

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B47C	Access Control Register (CCM_ACCESS_CTRL_ROOT104_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B480	Target Register (CCM_TARGET_ROOT105)	32	R/W	1000_0000h	5.1.7.10/461
3038_B484	Target Register (CCM_TARGET_ROOT105_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B488	Target Register (CCM_TARGET_ROOT105_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B48C	Target Register (CCM_TARGET_ROOT105_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B490	Miscellaneous Register (CCM_MISC105)	32	R/W	0000_0000h	5.1.7.14/469
3038_B494	Miscellaneous Register (CCM_MISC_ROOT105_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B498	Miscellaneous Register (CCM_MISC_ROOT105_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B49C	Miscellaneous Register (CCM_MISC_ROOT105_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B4A0	Post Divider Register (CCM_POST105)	32	R/W	0000_0000h	5.1.7.18/473
3038_B4A4	Post Divider Register (CCM_POST_ROOT105_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B4A8	Post Divider Register (CCM_POST_ROOT105_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B4AC	Post Divider Register (CCM_POST_ROOT105_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B4B0	Pre Divider Register (CCM_PRE105)	32	R/W	1000_0000h	5.1.7.22/485
3038_B4B4	Pre Divider Register (CCM_PRE_ROOT105_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B4B8	Pre Divider Register (CCM_PRE_ROOT105_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B4BC	Pre Divider Register (CCM_PRE_ROOT105_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B4F0	Access Control Register (CCM_ACCESS_CTRL105)	32	R/W	0000_0000h	5.1.7.26/497
3038_B4F4	Access Control Register (CCM_ACCESS_CTRL_ROOT105_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B4F8	Access Control Register (CCM_ACCESS_CTRL_ROOT105_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B4FC	Access Control Register (CCM_ACCESS_CTRL_ROOT105_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B500	Target Register (CCM_TARGET_ROOT106)	32	R/W	1000_0000h	5.1.7.10/461
3038_B504	Target Register (CCM_TARGET_ROOT106_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B508	Target Register (CCM_TARGET_ROOT106_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B50C	Target Register (CCM_TARGET_ROOT106_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B510	Miscellaneous Register (CCM_MISC106)	32	R/W	0000_0000h	5.1.7.14/469
3038_B514	Miscellaneous Register (CCM_MISC_ROOT106_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B518	Miscellaneous Register (CCM_MISC_ROOT106_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B51C	Miscellaneous Register (CCM_MISC_ROOT106_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B520	Post Divider Register (CCM_POST106)	32	R/W	0000_0000h	5.1.7.18/473
3038_B524	Post Divider Register (CCM_POST_ROOT106_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B528	Post Divider Register (CCM_POST_ROOT106_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B52C	Post Divider Register (CCM_POST_ROOT106_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B530	Pre Divider Register (CCM_PRE106)	32	R/W	1000_0000h	5.1.7.22/485
3038_B534	Pre Divider Register (CCM_PRE_ROOT106_SET)	32	R/W	0000_0000h	5.1.7.23/488

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B538	Pre Divider Register (CCM_PRE_ROOT106_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B53C	Pre Divider Register (CCM_PRE_ROOT106_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B570	Access Control Register (CCM_ACCESS_CTRL106)	32	R/W	0000_0000h	5.1.7.26/497
3038_B574	Access Control Register (CCM_ACCESS_CTRL_ROOT106_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B578	Access Control Register (CCM_ACCESS_CTRL_ROOT106_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B57C	Access Control Register (CCM_ACCESS_CTRL_ROOT106_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B580	Target Register (CCM_TARGET_ROOT107)	32	R/W	1000_0000h	5.1.7.10/461
3038_B584	Target Register (CCM_TARGET_ROOT107_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B588	Target Register (CCM_TARGET_ROOT107_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B58C	Target Register (CCM_TARGET_ROOT107_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B590	Miscellaneous Register (CCM_MISC107)	32	R/W	0000_0000h	5.1.7.14/469
3038_B594	Miscellaneous Register (CCM_MISC_ROOT107_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B598	Miscellaneous Register (CCM_MISC_ROOT107_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B59C	Miscellaneous Register (CCM_MISC_ROOT107_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B5A0	Post Divider Register (CCM_POST107)	32	R/W	0000_0000h	5.1.7.18/473
3038_B5A4	Post Divider Register (CCM_POST_ROOT107_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B5A8	Post Divider Register (CCM_POST_ROOT107_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B5AC	Post Divider Register (CCM_POST_ROOT107_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B5B0	Pre Divider Register (CCM_PRE107)	32	R/W	1000_0000h	5.1.7.22/485
3038_B5B4	Pre Divider Register (CCM_PRE_ROOT107_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B5B8	Pre Divider Register (CCM_PRE_ROOT107_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B5BC	Pre Divider Register (CCM_PRE_ROOT107_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B5F0	Access Control Register (CCM_ACCESS_CTRL107)	32	R/W	0000_0000h	5.1.7.26/497
3038_B5F4	Access Control Register (CCM_ACCESS_CTRL_ROOT107_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B5F8	Access Control Register (CCM_ACCESS_CTRL_ROOT107_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B5FC	Access Control Register (CCM_ACCESS_CTRL_ROOT107_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B600	Target Register (CCM_TARGET_ROOT108)	32	R/W	1000_0000h	5.1.7.10/461
3038_B604	Target Register (CCM_TARGET_ROOT108_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B608	Target Register (CCM_TARGET_ROOT108_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B60C	Target Register (CCM_TARGET_ROOT108_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B610	Miscellaneous Register (CCM_MISC108)	32	R/W	0000_0000h	5.1.7.14/469
3038_B614	Miscellaneous Register (CCM_MISC_ROOT108_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B618	Miscellaneous Register (CCM_MISC_ROOT108_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B61C	Miscellaneous Register (CCM_MISC_ROOT108_TOG)	32	R/W	0000_0000h	5.1.7.17/472

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B620	Post Divider Register (CCM_POST108)	32	R/W	0000_0000h	5.1.7.18/473
3038_B624	Post Divider Register (CCM_POST_ROOT108_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B628	Post Divider Register (CCM_POST_ROOT108_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B62C	Post Divider Register (CCM_POST_ROOT108_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B630	Pre Divider Register (CCM_PRE108)	32	R/W	1000_0000h	5.1.7.22/485
3038_B634	Pre Divider Register (CCM_PRE_ROOT108_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B638	Pre Divider Register (CCM_PRE_ROOT108_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B63C	Pre Divider Register (CCM_PRE_ROOT108_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B670	Access Control Register (CCM_ACCESS_CTRL108)	32	R/W	0000_0000h	5.1.7.26/497
3038_B674	Access Control Register (CCM_ACCESS_CTRL_ROOT108_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B678	Access Control Register (CCM_ACCESS_CTRL_ROOT108_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B67C	Access Control Register (CCM_ACCESS_CTRL_ROOT108_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B680	Target Register (CCM_TARGET_ROOT109)	32	R/W	1000_0000h	5.1.7.10/461
3038_B684	Target Register (CCM_TARGET_ROOT109_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B688	Target Register (CCM_TARGET_ROOT109_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B68C	Target Register (CCM_TARGET_ROOT109_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B690	Miscellaneous Register (CCM_MISC109)	32	R/W	0000_0000h	5.1.7.14/469
3038_B694	Miscellaneous Register (CCM_MISC_ROOT109_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B698	Miscellaneous Register (CCM_MISC_ROOT109_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B69C	Miscellaneous Register (CCM_MISC_ROOT109_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B6A0	Post Divider Register (CCM_POST109)	32	R/W	0000_0000h	5.1.7.18/473
3038_B6A4	Post Divider Register (CCM_POST_ROOT109_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B6A8	Post Divider Register (CCM_POST_ROOT109_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B6AC	Post Divider Register (CCM_POST_ROOT109_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B6B0	Pre Divider Register (CCM_PRE109)	32	R/W	1000_0000h	5.1.7.22/485
3038_B6B4	Pre Divider Register (CCM_PRE_ROOT109_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B6B8	Pre Divider Register (CCM_PRE_ROOT109_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B6BC	Pre Divider Register (CCM_PRE_ROOT109_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B6F0	Access Control Register (CCM_ACCESS_CTRL109)	32	R/W	0000_0000h	5.1.7.26/497
3038_B6F4	Access Control Register (CCM_ACCESS_CTRL_ROOT109_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B6F8	Access Control Register (CCM_ACCESS_CTRL_ROOT109_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B6FC	Access Control Register (CCM_ACCESS_CTRL_ROOT109_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B700	Target Register (CCM_TARGET_ROOT110)	32	R/W	1000_0000h	5.1.7.10/461
3038_B704	Target Register (CCM_TARGET_ROOT110_SET)	32	R/W	0000_0000h	5.1.7.11/463

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B708	Target Register (CCM_TARGET_ROOT110_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B70C	Target Register (CCM_TARGET_ROOT110_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B710	Miscellaneous Register (CCM_MISC110)	32	R/W	0000_0000h	5.1.7.14/469
3038_B714	Miscellaneous Register (CCM_MISC_ROOT110_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B718	Miscellaneous Register (CCM_MISC_ROOT110_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B71C	Miscellaneous Register (CCM_MISC_ROOT110_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B720	Post Divider Register (CCM_POST110)	32	R/W	0000_0000h	5.1.7.18/473
3038_B724	Post Divider Register (CCM_POST_ROOT110_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B728	Post Divider Register (CCM_POST_ROOT110_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B72C	Post Divider Register (CCM_POST_ROOT110_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B730	Pre Divider Register (CCM_PRE110)	32	R/W	1000_0000h	5.1.7.22/485
3038_B734	Pre Divider Register (CCM_PRE_ROOT110_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B738	Pre Divider Register (CCM_PRE_ROOT110_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B73C	Pre Divider Register (CCM_PRE_ROOT110_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B770	Access Control Register (CCM_ACCESS_CTRL110)	32	R/W	0000_0000h	5.1.7.26/497
3038_B774	Access Control Register (CCM_ACCESS_CTRL_ROOT110_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B778	Access Control Register (CCM_ACCESS_CTRL_ROOT110_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B77C	Access Control Register (CCM_ACCESS_CTRL_ROOT110_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B780	Target Register (CCM_TARGET_ROOT111)	32	R/W	1000_0000h	5.1.7.10/461
3038_B784	Target Register (CCM_TARGET_ROOT111_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B788	Target Register (CCM_TARGET_ROOT111_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B78C	Target Register (CCM_TARGET_ROOT111_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B790	Miscellaneous Register (CCM_MISC111)	32	R/W	0000_0000h	5.1.7.14/469
3038_B794	Miscellaneous Register (CCM_MISC_ROOT111_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B798	Miscellaneous Register (CCM_MISC_ROOT111_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B79C	Miscellaneous Register (CCM_MISC_ROOT111_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B7A0	Post Divider Register (CCM_POST111)	32	R/W	0000_0000h	5.1.7.18/473
3038_B7A4	Post Divider Register (CCM_POST_ROOT111_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B7A8	Post Divider Register (CCM_POST_ROOT111_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B7AC	Post Divider Register (CCM_POST_ROOT111_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B7B0	Pre Divider Register (CCM_PRE111)	32	R/W	1000_0000h	5.1.7.22/485
3038_B7B4	Pre Divider Register (CCM_PRE_ROOT111_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B7B8	Pre Divider Register (CCM_PRE_ROOT111_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B7BC	Pre Divider Register (CCM_PRE_ROOT111_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B7F0	Access Control Register (CCM_ACCESS_CTRL111)	32	R/W	0000_0000h	5.1.7.26/497

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B7F4	Access Control Register (CCM_ACCESS_CTRL_ROOT111_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B7F8	Access Control Register (CCM_ACCESS_CTRL_ROOT111_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B7FC	Access Control Register (CCM_ACCESS_CTRL_ROOT111_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B800	Target Register (CCM_TARGET_ROOT112)	32	R/W	1000_0000h	5.1.7.10/461
3038_B804	Target Register (CCM_TARGET_ROOT112_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B808	Target Register (CCM_TARGET_ROOT112_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B80C	Target Register (CCM_TARGET_ROOT112_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B810	Miscellaneous Register (CCM_MISC112)	32	R/W	0000_0000h	5.1.7.14/469
3038_B814	Miscellaneous Register (CCM_MISC_ROOT112_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B818	Miscellaneous Register (CCM_MISC_ROOT112_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B81C	Miscellaneous Register (CCM_MISC_ROOT112_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B820	Post Divider Register (CCM_POST112)	32	R/W	0000_0000h	5.1.7.18/473
3038_B824	Post Divider Register (CCM_POST_ROOT112_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B828	Post Divider Register (CCM_POST_ROOT112_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B82C	Post Divider Register (CCM_POST_ROOT112_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B830	Pre Divider Register (CCM_PRE112)	32	R/W	1000_0000h	5.1.7.22/485
3038_B834	Pre Divider Register (CCM_PRE_ROOT112_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B838	Pre Divider Register (CCM_PRE_ROOT112_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B83C	Pre Divider Register (CCM_PRE_ROOT112_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B870	Access Control Register (CCM_ACCESS_CTRL112)	32	R/W	0000_0000h	5.1.7.26/497
3038_B874	Access Control Register (CCM_ACCESS_CTRL_ROOT112_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B878	Access Control Register (CCM_ACCESS_CTRL_ROOT112_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B87C	Access Control Register (CCM_ACCESS_CTRL_ROOT112_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B880	Target Register (CCM_TARGET_ROOT113)	32	R/W	1000_0000h	5.1.7.10/461
3038_B884	Target Register (CCM_TARGET_ROOT113_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B888	Target Register (CCM_TARGET_ROOT113_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B88C	Target Register (CCM_TARGET_ROOT113_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B890	Miscellaneous Register (CCM_MISC113)	32	R/W	0000_0000h	5.1.7.14/469
3038_B894	Miscellaneous Register (CCM_MISC_ROOT113_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B898	Miscellaneous Register (CCM_MISC_ROOT113_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B89C	Miscellaneous Register (CCM_MISC_ROOT113_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B8A0	Post Divider Register (CCM_POST113)	32	R/W	0000_0000h	5.1.7.18/473
3038_B8A4	Post Divider Register (CCM_POST_ROOT113_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B8A8	Post Divider Register (CCM_POST_ROOT113_CLR)	32	R/W	0000_0000h	5.1.7.20/479

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B8AC	Post Divider Register (CCM_POST_ROOT113_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B8B0	Pre Divider Register (CCM_PRE113)	32	R/W	1000_0000h	5.1.7.22/485
3038_B8B4	Pre Divider Register (CCM_PRE_ROOT113_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B8B8	Pre Divider Register (CCM_PRE_ROOT113_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B8BC	Pre Divider Register (CCM_PRE_ROOT113_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B8F0	Access Control Register (CCM_ACCESS_CTRL113)	32	R/W	0000_0000h	5.1.7.26/497
3038_B8F4	Access Control Register (CCM_ACCESS_CTRL_ROOT113_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B8F8	Access Control Register (CCM_ACCESS_CTRL_ROOT113_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B8FC	Access Control Register (CCM_ACCESS_CTRL_ROOT113_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B900	Target Register (CCM_TARGET_ROOT114)	32	R/W	1000_0000h	5.1.7.10/461
3038_B904	Target Register (CCM_TARGET_ROOT114_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B908	Target Register (CCM_TARGET_ROOT114_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B90C	Target Register (CCM_TARGET_ROOT114_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B910	Miscellaneous Register (CCM_MISC114)	32	R/W	0000_0000h	5.1.7.14/469
3038_B914	Miscellaneous Register (CCM_MISC_ROOT114_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B918	Miscellaneous Register (CCM_MISC_ROOT114_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B91C	Miscellaneous Register (CCM_MISC_ROOT114_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B920	Post Divider Register (CCM_POST114)	32	R/W	0000_0000h	5.1.7.18/473
3038_B924	Post Divider Register (CCM_POST_ROOT114_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B928	Post Divider Register (CCM_POST_ROOT114_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B92C	Post Divider Register (CCM_POST_ROOT114_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B930	Pre Divider Register (CCM_PRE114)	32	R/W	1000_0000h	5.1.7.22/485
3038_B934	Pre Divider Register (CCM_PRE_ROOT114_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B938	Pre Divider Register (CCM_PRE_ROOT114_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B93C	Pre Divider Register (CCM_PRE_ROOT114_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B970	Access Control Register (CCM_ACCESS_CTRL114)	32	R/W	0000_0000h	5.1.7.26/497
3038_B974	Access Control Register (CCM_ACCESS_CTRL_ROOT114_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B978	Access Control Register (CCM_ACCESS_CTRL_ROOT114_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B97C	Access Control Register (CCM_ACCESS_CTRL_ROOT114_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_B980	Target Register (CCM_TARGET_ROOT115)	32	R/W	1000_0000h	5.1.7.10/461
3038_B984	Target Register (CCM_TARGET_ROOT115_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_B988	Target Register (CCM_TARGET_ROOT115_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_B98C	Target Register (CCM_TARGET_ROOT115_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_B990	Miscellaneous Register (CCM_MISC115)	32	R/W	0000_0000h	5.1.7.14/469

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B994	Miscellaneous Register (CCM_MISC_ROOT115_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_B998	Miscellaneous Register (CCM_MISC_ROOT115_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_B99C	Miscellaneous Register (CCM_MISC_ROOT115_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_B9A0	Post Divider Register (CCM_POST115)	32	R/W	0000_0000h	5.1.7.18/473
3038_B9A4	Post Divider Register (CCM_POST_ROOT115_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_B9A8	Post Divider Register (CCM_POST_ROOT115_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_B9AC	Post Divider Register (CCM_POST_ROOT115_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_B9B0	Pre Divider Register (CCM_PRE115)	32	R/W	1000_0000h	5.1.7.22/485
3038_B9B4	Pre Divider Register (CCM_PRE_ROOT115_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_B9B8	Pre Divider Register (CCM_PRE_ROOT115_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_B9BC	Pre Divider Register (CCM_PRE_ROOT115_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_B9F0	Access Control Register (CCM_ACCESS_CTRL115)	32	R/W	0000_0000h	5.1.7.26/497
3038_B9F4	Access Control Register (CCM_ACCESS_CTRL_ROOT115_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_B9F8	Access Control Register (CCM_ACCESS_CTRL_ROOT115_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_B9FC	Access Control Register (CCM_ACCESS_CTRL_ROOT115_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_BA00	Target Register (CCM_TARGET_ROOT116)	32	R/W	1000_0000h	5.1.7.10/461
3038_BA04	Target Register (CCM_TARGET_ROOT116_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_BA08	Target Register (CCM_TARGET_ROOT116_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_BA0C	Target Register (CCM_TARGET_ROOT116_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_BA10	Miscellaneous Register (CCM_MISC116)	32	R/W	0000_0000h	5.1.7.14/469
3038_BA14	Miscellaneous Register (CCM_MISC_ROOT116_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_BA18	Miscellaneous Register (CCM_MISC_ROOT116_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_BA1C	Miscellaneous Register (CCM_MISC_ROOT116_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_BA20	Post Divider Register (CCM_POST116)	32	R/W	0000_0000h	5.1.7.18/473
3038_BA24	Post Divider Register (CCM_POST_ROOT116_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_BA28	Post Divider Register (CCM_POST_ROOT116_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_BA2C	Post Divider Register (CCM_POST_ROOT116_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_BA30	Pre Divider Register (CCM_PRE116)	32	R/W	1000_0000h	5.1.7.22/485
3038_BA34	Pre Divider Register (CCM_PRE_ROOT116_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_BA38	Pre Divider Register (CCM_PRE_ROOT116_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_BA3C	Pre Divider Register (CCM_PRE_ROOT116_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_BA70	Access Control Register (CCM_ACCESS_CTRL116)	32	R/W	0000_0000h	5.1.7.26/497
3038_BA74	Access Control Register (CCM_ACCESS_CTRL_ROOT116_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_BA78	Access Control Register (CCM_ACCESS_CTRL_ROOT116_CLR)	32	R/W	0000_0000h	5.1.7.28/502

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BA7C	Access Control Register (CCM_ACCESS_CTRL_ROOT116_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_BA80	Target Register (CCM_TARGET_ROOT117)	32	R/W	1000_0000h	5.1.7.10/461
3038_BA84	Target Register (CCM_TARGET_ROOT117_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_BA88	Target Register (CCM_TARGET_ROOT117_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_BA8C	Target Register (CCM_TARGET_ROOT117_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_BA90	Miscellaneous Register (CCM_MISC117)	32	R/W	0000_0000h	5.1.7.14/469
3038_BA94	Miscellaneous Register (CCM_MISC_ROOT117_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_BA98	Miscellaneous Register (CCM_MISC_ROOT117_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_BA9C	Miscellaneous Register (CCM_MISC_ROOT117_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_BAA0	Post Divider Register (CCM_POST117)	32	R/W	0000_0000h	5.1.7.18/473
3038_BAA4	Post Divider Register (CCM_POST_ROOT117_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_BAA8	Post Divider Register (CCM_POST_ROOT117_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_BAAC	Post Divider Register (CCM_POST_ROOT117_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_BAB0	Pre Divider Register (CCM_PRE117)	32	R/W	1000_0000h	5.1.7.22/485
3038_BAB4	Pre Divider Register (CCM_PRE_ROOT117_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_BAB8	Pre Divider Register (CCM_PRE_ROOT117_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_BABC	Pre Divider Register (CCM_PRE_ROOT117_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_BAF0	Access Control Register (CCM_ACCESS_CTRL117)	32	R/W	0000_0000h	5.1.7.26/497
3038_BAF4	Access Control Register (CCM_ACCESS_CTRL_ROOT117_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_BAF8	Access Control Register (CCM_ACCESS_CTRL_ROOT117_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_BAFC	Access Control Register (CCM_ACCESS_CTRL_ROOT117_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_BB00	Target Register (CCM_TARGET_ROOT118)	32	R/W	1000_0000h	5.1.7.10/461
3038_BB04	Target Register (CCM_TARGET_ROOT118_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_BB08	Target Register (CCM_TARGET_ROOT118_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_BB0C	Target Register (CCM_TARGET_ROOT118_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_BB10	Miscellaneous Register (CCM_MISC118)	32	R/W	0000_0000h	5.1.7.14/469
3038_BB14	Miscellaneous Register (CCM_MISC_ROOT118_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_BB18	Miscellaneous Register (CCM_MISC_ROOT118_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_BB1C	Miscellaneous Register (CCM_MISC_ROOT118_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_BB20	Post Divider Register (CCM_POST118)	32	R/W	0000_0000h	5.1.7.18/473
3038_BB24	Post Divider Register (CCM_POST_ROOT118_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_BB28	Post Divider Register (CCM_POST_ROOT118_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_BB2C	Post Divider Register (CCM_POST_ROOT118_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_BB30	Pre Divider Register (CCM_PRE118)	32	R/W	1000_0000h	5.1.7.22/485
3038_BB34	Pre Divider Register (CCM_PRE_ROOT118_SET)	32	R/W	0000_0000h	5.1.7.23/488

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BB38	Pre Divider Register (CCM_PRE_ROOT118_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_BB3C	Pre Divider Register (CCM_PRE_ROOT118_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_BB70	Access Control Register (CCM_ACCESS_CTRL118)	32	R/W	0000_0000h	5.1.7.26/497
3038_BB74	Access Control Register (CCM_ACCESS_CTRL_ROOT118_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_BB78	Access Control Register (CCM_ACCESS_CTRL_ROOT118_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_BB7C	Access Control Register (CCM_ACCESS_CTRL_ROOT118_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_BB80	Target Register (CCM_TARGET_ROOT119)	32	R/W	1000_0000h	5.1.7.10/461
3038_BB84	Target Register (CCM_TARGET_ROOT119_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_BB88	Target Register (CCM_TARGET_ROOT119_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_BB8C	Target Register (CCM_TARGET_ROOT119_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_BB90	Miscellaneous Register (CCM_MISC119)	32	R/W	0000_0000h	5.1.7.14/469
3038_BB94	Miscellaneous Register (CCM_MISC_ROOT119_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_BB98	Miscellaneous Register (CCM_MISC_ROOT119_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_BB9C	Miscellaneous Register (CCM_MISC_ROOT119_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_BBA0	Post Divider Register (CCM_POST119)	32	R/W	0000_0000h	5.1.7.18/473
3038_BBA4	Post Divider Register (CCM_POST_ROOT119_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_BBA8	Post Divider Register (CCM_POST_ROOT119_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_BBAC	Post Divider Register (CCM_POST_ROOT119_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_BBB0	Pre Divider Register (CCM_PRE119)	32	R/W	1000_0000h	5.1.7.22/485
3038_BBB4	Pre Divider Register (CCM_PRE_ROOT119_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_BBB8	Pre Divider Register (CCM_PRE_ROOT119_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_BBBC	Pre Divider Register (CCM_PRE_ROOT119_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_BBF0	Access Control Register (CCM_ACCESS_CTRL119)	32	R/W	0000_0000h	5.1.7.26/497
3038_BBF4	Access Control Register (CCM_ACCESS_CTRL_ROOT119_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_BBF8	Access Control Register (CCM_ACCESS_CTRL_ROOT119_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_BBFC	Access Control Register (CCM_ACCESS_CTRL_ROOT119_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_BC00	Target Register (CCM_TARGET_ROOT120)	32	R/W	1000_0000h	5.1.7.10/461
3038_BC04	Target Register (CCM_TARGET_ROOT120_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_BC08	Target Register (CCM_TARGET_ROOT120_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_BC0C	Target Register (CCM_TARGET_ROOT120_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_BC10	Miscellaneous Register (CCM_MISC120)	32	R/W	0000_0000h	5.1.7.14/469
3038_BC14	Miscellaneous Register (CCM_MISC_ROOT120_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_BC18	Miscellaneous Register (CCM_MISC_ROOT120_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_BC1C	Miscellaneous Register (CCM_MISC_ROOT120_TOG)	32	R/W	0000_0000h	5.1.7.17/472

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BC20	Post Divider Register (CCM_POST120)	32	R/W	0000_0000h	5.1.7.18/473
3038_BC24	Post Divider Register (CCM_POST_ROOT120_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_BC28	Post Divider Register (CCM_POST_ROOT120_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_BC2C	Post Divider Register (CCM_POST_ROOT120_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_BC30	Pre Divider Register (CCM_PRE120)	32	R/W	1000_0000h	5.1.7.22/485
3038_BC34	Pre Divider Register (CCM_PRE_ROOT120_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_BC38	Pre Divider Register (CCM_PRE_ROOT120_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_BC3C	Pre Divider Register (CCM_PRE_ROOT120_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_BC70	Access Control Register (CCM_ACCESS_CTRL120)	32	R/W	0000_0000h	5.1.7.26/497
3038_BC74	Access Control Register (CCM_ACCESS_CTRL_ROOT120_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_BC78	Access Control Register (CCM_ACCESS_CTRL_ROOT120_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_BC7C	Access Control Register (CCM_ACCESS_CTRL_ROOT120_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_BC80	Target Register (CCM_TARGET_ROOT121)	32	R/W	1000_0000h	5.1.7.10/461
3038_BC84	Target Register (CCM_TARGET_ROOT121_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_BC88	Target Register (CCM_TARGET_ROOT121_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_BC8C	Target Register (CCM_TARGET_ROOT121_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_BC90	Miscellaneous Register (CCM_MISC121)	32	R/W	0000_0000h	5.1.7.14/469
3038_BC94	Miscellaneous Register (CCM_MISC_ROOT121_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_BC98	Miscellaneous Register (CCM_MISC_ROOT121_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_BC9C	Miscellaneous Register (CCM_MISC_ROOT121_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_BCA0	Post Divider Register (CCM_POST121)	32	R/W	0000_0000h	5.1.7.18/473
3038_BCA4	Post Divider Register (CCM_POST_ROOT121_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_BCA8	Post Divider Register (CCM_POST_ROOT121_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_BCAC	Post Divider Register (CCM_POST_ROOT121_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_BCB0	Pre Divider Register (CCM_PRE121)	32	R/W	1000_0000h	5.1.7.22/485
3038_BCB4	Pre Divider Register (CCM_PRE_ROOT121_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_BCB8	Pre Divider Register (CCM_PRE_ROOT121_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_BCBC	Pre Divider Register (CCM_PRE_ROOT121_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_BCF0	Access Control Register (CCM_ACCESS_CTRL121)	32	R/W	0000_0000h	5.1.7.26/497
3038_BCF4	Access Control Register (CCM_ACCESS_CTRL_ROOT121_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_BCF8	Access Control Register (CCM_ACCESS_CTRL_ROOT121_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_BCFC	Access Control Register (CCM_ACCESS_CTRL_ROOT121_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_BD00	Target Register (CCM_TARGET_ROOT122)	32	R/W	1000_0000h	5.1.7.10/461
3038_BD04	Target Register (CCM_TARGET_ROOT122_SET)	32	R/W	0000_0000h	5.1.7.11/463

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BD08	Target Register (CCM_TARGET_ROOT122_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_BD0C	Target Register (CCM_TARGET_ROOT122_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_BD10	Miscellaneous Register (CCM_MISC122)	32	R/W	0000_0000h	5.1.7.14/469
3038_BD14	Miscellaneous Register (CCM_MISC_ROOT122_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_BD18	Miscellaneous Register (CCM_MISC_ROOT122_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_BD1C	Miscellaneous Register (CCM_MISC_ROOT122_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_BD20	Post Divider Register (CCM_POST122)	32	R/W	0000_0000h	5.1.7.18/473
3038_BD24	Post Divider Register (CCM_POST_ROOT122_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_BD28	Post Divider Register (CCM_POST_ROOT122_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_BD2C	Post Divider Register (CCM_POST_ROOT122_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_BD30	Pre Divider Register (CCM_PRE122)	32	R/W	1000_0000h	5.1.7.22/485
3038_BD34	Pre Divider Register (CCM_PRE_ROOT122_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_BD38	Pre Divider Register (CCM_PRE_ROOT122_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_BD3C	Pre Divider Register (CCM_PRE_ROOT122_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_BD70	Access Control Register (CCM_ACCESS_CTRL122)	32	R/W	0000_0000h	5.1.7.26/497
3038_BD74	Access Control Register (CCM_ACCESS_CTRL_ROOT122_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_BD78	Access Control Register (CCM_ACCESS_CTRL_ROOT122_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_BD7C	Access Control Register (CCM_ACCESS_CTRL_ROOT122_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_BD80	Target Register (CCM_TARGET_ROOT123)	32	R/W	1000_0000h	5.1.7.10/461
3038_BD84	Target Register (CCM_TARGET_ROOT123_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_BD88	Target Register (CCM_TARGET_ROOT123_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_BD8C	Target Register (CCM_TARGET_ROOT123_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_BD90	Miscellaneous Register (CCM_MISC123)	32	R/W	0000_0000h	5.1.7.14/469
3038_BD94	Miscellaneous Register (CCM_MISC_ROOT123_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_BD98	Miscellaneous Register (CCM_MISC_ROOT123_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_BD9C	Miscellaneous Register (CCM_MISC_ROOT123_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_BDA0	Post Divider Register (CCM_POST123)	32	R/W	0000_0000h	5.1.7.18/473
3038_BDA4	Post Divider Register (CCM_POST_ROOT123_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_BDA8	Post Divider Register (CCM_POST_ROOT123_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_BDAC	Post Divider Register (CCM_POST_ROOT123_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_BDB0	Pre Divider Register (CCM_PRE123)	32	R/W	1000_0000h	5.1.7.22/485
3038_BDB4	Pre Divider Register (CCM_PRE_ROOT123_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_BDB8	Pre Divider Register (CCM_PRE_ROOT123_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_BDBC	Pre Divider Register (CCM_PRE_ROOT123_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_BDF0	Access Control Register (CCM_ACCESS_CTRL123)	32	R/W	0000_0000h	5.1.7.26/497

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BDF4	Access Control Register (CCM_ACCESS_CTRL_ROOT123_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_BDF8	Access Control Register (CCM_ACCESS_CTRL_ROOT123_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_BDFC	Access Control Register (CCM_ACCESS_CTRL_ROOT123_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_BE00	Target Register (CCM_TARGET_ROOT124)	32	R/W	1000_0000h	5.1.7.10/461
3038_BE04	Target Register (CCM_TARGET_ROOT124_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_BE08	Target Register (CCM_TARGET_ROOT124_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_BE0C	Target Register (CCM_TARGET_ROOT124_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_BE10	Miscellaneous Register (CCM_MISC124)	32	R/W	0000_0000h	5.1.7.14/469
3038_BE14	Miscellaneous Register (CCM_MISC_ROOT124_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_BE18	Miscellaneous Register (CCM_MISC_ROOT124_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_BE1C	Miscellaneous Register (CCM_MISC_ROOT124_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_BE20	Post Divider Register (CCM_POST124)	32	R/W	0000_0000h	5.1.7.18/473
3038_BE24	Post Divider Register (CCM_POST_ROOT124_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_BE28	Post Divider Register (CCM_POST_ROOT124_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_BE2C	Post Divider Register (CCM_POST_ROOT124_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_BE30	Pre Divider Register (CCM_PRE124)	32	R/W	1000_0000h	5.1.7.22/485
3038_BE34	Pre Divider Register (CCM_PRE_ROOT124_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_BE38	Pre Divider Register (CCM_PRE_ROOT124_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_BE3C	Pre Divider Register (CCM_PRE_ROOT124_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_BE70	Access Control Register (CCM_ACCESS_CTRL124)	32	R/W	0000_0000h	5.1.7.26/497
3038_BE74	Access Control Register (CCM_ACCESS_CTRL_ROOT124_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_BE78	Access Control Register (CCM_ACCESS_CTRL_ROOT124_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_BE7C	Access Control Register (CCM_ACCESS_CTRL_ROOT124_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_BE80	Target Register (CCM_TARGET_ROOT125)	32	R/W	1000_0000h	5.1.7.10/461
3038_BE84	Target Register (CCM_TARGET_ROOT125_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_BE88	Target Register (CCM_TARGET_ROOT125_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_BE8C	Target Register (CCM_TARGET_ROOT125_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_BE90	Miscellaneous Register (CCM_MISC125)	32	R/W	0000_0000h	5.1.7.14/469
3038_BE94	Miscellaneous Register (CCM_MISC_ROOT125_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_BE98	Miscellaneous Register (CCM_MISC_ROOT125_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_BE9C	Miscellaneous Register (CCM_MISC_ROOT125_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_BEA0	Post Divider Register (CCM_POST125)	32	R/W	0000_0000h	5.1.7.18/473
3038_BEA4	Post Divider Register (CCM_POST_ROOT125_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_BEA8	Post Divider Register (CCM_POST_ROOT125_CLR)	32	R/W	0000_0000h	5.1.7.20/479

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BEAC	Post Divider Register (CCM_POST_ROOT125_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_BEB0	Pre Divider Register (CCM_PRE125)	32	R/W	1000_0000h	5.1.7.22/485
3038_BEB4	Pre Divider Register (CCM_PRE_ROOT125_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_BEB8	Pre Divider Register (CCM_PRE_ROOT125_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_BEBC	Pre Divider Register (CCM_PRE_ROOT125_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_BEF0	Access Control Register (CCM_ACCESS_CTRL125)	32	R/W	0000_0000h	5.1.7.26/497
3038_BEF4	Access Control Register (CCM_ACCESS_CTRL_ROOT125_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_BEF8	Access Control Register (CCM_ACCESS_CTRL_ROOT125_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_BEFC	Access Control Register (CCM_ACCESS_CTRL_ROOT125_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_BF00	Target Register (CCM_TARGET_ROOT126)	32	R/W	1000_0000h	5.1.7.10/461
3038_BF04	Target Register (CCM_TARGET_ROOT126_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_BF08	Target Register (CCM_TARGET_ROOT126_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_BF0C	Target Register (CCM_TARGET_ROOT126_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_BF10	Miscellaneous Register (CCM_MISC126)	32	R/W	0000_0000h	5.1.7.14/469
3038_BF14	Miscellaneous Register (CCM_MISC_ROOT126_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_BF18	Miscellaneous Register (CCM_MISC_ROOT126_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_BF1C	Miscellaneous Register (CCM_MISC_ROOT126_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_BF20	Post Divider Register (CCM_POST126)	32	R/W	0000_0000h	5.1.7.18/473
3038_BF24	Post Divider Register (CCM_POST_ROOT126_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_BF28	Post Divider Register (CCM_POST_ROOT126_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_BF2C	Post Divider Register (CCM_POST_ROOT126_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_BF30	Pre Divider Register (CCM_PRE126)	32	R/W	1000_0000h	5.1.7.22/485
3038_BF34	Pre Divider Register (CCM_PRE_ROOT126_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_BF38	Pre Divider Register (CCM_PRE_ROOT126_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_BF3C	Pre Divider Register (CCM_PRE_ROOT126_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_BF70	Access Control Register (CCM_ACCESS_CTRL126)	32	R/W	0000_0000h	5.1.7.26/497
3038_BF74	Access Control Register (CCM_ACCESS_CTRL_ROOT126_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_BF78	Access Control Register (CCM_ACCESS_CTRL_ROOT126_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_BF7C	Access Control Register (CCM_ACCESS_CTRL_ROOT126_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_BF80	Target Register (CCM_TARGET_ROOT127)	32	R/W	1000_0000h	5.1.7.10/461
3038_BF84	Target Register (CCM_TARGET_ROOT127_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_BF88	Target Register (CCM_TARGET_ROOT127_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_BF8C	Target Register (CCM_TARGET_ROOT127_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_BF90	Miscellaneous Register (CCM_MISC127)	32	R/W	0000_0000h	5.1.7.14/469

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BF94	Miscellaneous Register (CCM_MISC_ROOT127_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_BF98	Miscellaneous Register (CCM_MISC_ROOT127_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_BF9C	Miscellaneous Register (CCM_MISC_ROOT127_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_BFA0	Post Divider Register (CCM_POST127)	32	R/W	0000_0000h	5.1.7.18/473
3038_BFA4	Post Divider Register (CCM_POST_ROOT127_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_BFA8	Post Divider Register (CCM_POST_ROOT127_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_BFAC	Post Divider Register (CCM_POST_ROOT127_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_BFB0	Pre Divider Register (CCM_PRE127)	32	R/W	1000_0000h	5.1.7.22/485
3038_BFB4	Pre Divider Register (CCM_PRE_ROOT127_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_BFB8	Pre Divider Register (CCM_PRE_ROOT127_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_BFBC	Pre Divider Register (CCM_PRE_ROOT127_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_BFF0	Access Control Register (CCM_ACCESS_CTRL127)	32	R/W	0000_0000h	5.1.7.26/497
3038_BFF4	Access Control Register (CCM_ACCESS_CTRL_ROOT127_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_BFF8	Access Control Register (CCM_ACCESS_CTRL_ROOT127_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_BFFC	Access Control Register (CCM_ACCESS_CTRL_ROOT127_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C000	Target Register (CCM_TARGET_ROOT128)	32	R/W	1000_0000h	5.1.7.10/461
3038_C004	Target Register (CCM_TARGET_ROOT128_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_C008	Target Register (CCM_TARGET_ROOT128_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C00C	Target Register (CCM_TARGET_ROOT128_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C010	Miscellaneous Register (CCM_MISC128)	32	R/W	0000_0000h	5.1.7.14/469
3038_C014	Miscellaneous Register (CCM_MISC_ROOT128_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C018	Miscellaneous Register (CCM_MISC_ROOT128_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C01C	Miscellaneous Register (CCM_MISC_ROOT128_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_C020	Post Divider Register (CCM_POST128)	32	R/W	0000_0000h	5.1.7.18/473
3038_C024	Post Divider Register (CCM_POST_ROOT128_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C028	Post Divider Register (CCM_POST_ROOT128_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_C02C	Post Divider Register (CCM_POST_ROOT128_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C030	Pre Divider Register (CCM_PRE128)	32	R/W	1000_0000h	5.1.7.22/485
3038_C034	Pre Divider Register (CCM_PRE_ROOT128_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_C038	Pre Divider Register (CCM_PRE_ROOT128_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C03C	Pre Divider Register (CCM_PRE_ROOT128_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C070	Access Control Register (CCM_ACCESS_CTRL128)	32	R/W	0000_0000h	5.1.7.26/497
3038_C074	Access Control Register (CCM_ACCESS_CTRL_ROOT128_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C078	Access Control Register (CCM_ACCESS_CTRL_ROOT128_CLR)	32	R/W	0000_0000h	5.1.7.28/502

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C07C	Access Control Register (CCM_ACCESS_CTRL_ROOT128_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C080	Target Register (CCM_TARGET_ROOT129)	32	R/W	1000_0000h	5.1.7.10/461
3038_C084	Target Register (CCM_TARGET_ROOT129_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_C088	Target Register (CCM_TARGET_ROOT129_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C08C	Target Register (CCM_TARGET_ROOT129_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C090	Miscellaneous Register (CCM_MISC129)	32	R/W	0000_0000h	5.1.7.14/469
3038_C094	Miscellaneous Register (CCM_MISC_ROOT129_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C098	Miscellaneous Register (CCM_MISC_ROOT129_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C09C	Miscellaneous Register (CCM_MISC_ROOT129_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_C0A0	Post Divider Register (CCM_POST129)	32	R/W	0000_0000h	5.1.7.18/473
3038_C0A4	Post Divider Register (CCM_POST_ROOT129_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C0A8	Post Divider Register (CCM_POST_ROOT129_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_C0AC	Post Divider Register (CCM_POST_ROOT129_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C0B0	Pre Divider Register (CCM_PRE129)	32	R/W	1000_0000h	5.1.7.22/485
3038_C0B4	Pre Divider Register (CCM_PRE_ROOT129_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_C0B8	Pre Divider Register (CCM_PRE_ROOT129_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C0BC	Pre Divider Register (CCM_PRE_ROOT129_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C0F0	Access Control Register (CCM_ACCESS_CTRL129)	32	R/W	0000_0000h	5.1.7.26/497
3038_C0F4	Access Control Register (CCM_ACCESS_CTRL_ROOT129_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C0F8	Access Control Register (CCM_ACCESS_CTRL_ROOT129_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_C0FC	Access Control Register (CCM_ACCESS_CTRL_ROOT129_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C100	Target Register (CCM_TARGET_ROOT130)	32	R/W	1000_0000h	5.1.7.10/461
3038_C104	Target Register (CCM_TARGET_ROOT130_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_C108	Target Register (CCM_TARGET_ROOT130_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C10C	Target Register (CCM_TARGET_ROOT130_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C110	Miscellaneous Register (CCM_MISC130)	32	R/W	0000_0000h	5.1.7.14/469
3038_C114	Miscellaneous Register (CCM_MISC_ROOT130_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C118	Miscellaneous Register (CCM_MISC_ROOT130_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C11C	Miscellaneous Register (CCM_MISC_ROOT130_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_C120	Post Divider Register (CCM_POST130)	32	R/W	0000_0000h	5.1.7.18/473
3038_C124	Post Divider Register (CCM_POST_ROOT130_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C128	Post Divider Register (CCM_POST_ROOT130_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_C12C	Post Divider Register (CCM_POST_ROOT130_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C130	Pre Divider Register (CCM_PRE130)	32	R/W	1000_0000h	5.1.7.22/485
3038_C134	Pre Divider Register (CCM_PRE_ROOT130_SET)	32	R/W	0000_0000h	5.1.7.23/488

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C138	Pre Divider Register (CCM_PRE_ROOT130_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C13C	Pre Divider Register (CCM_PRE_ROOT130_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C170	Access Control Register (CCM_ACCESS_CTRL130)	32	R/W	0000_0000h	5.1.7.26/497
3038_C174	Access Control Register (CCM_ACCESS_CTRL_ROOT130_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C178	Access Control Register (CCM_ACCESS_CTRL_ROOT130_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_C17C	Access Control Register (CCM_ACCESS_CTRL_ROOT130_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C180	Target Register (CCM_TARGET_ROOT131)	32	R/W	1000_0000h	5.1.7.10/461
3038_C184	Target Register (CCM_TARGET_ROOT131_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_C188	Target Register (CCM_TARGET_ROOT131_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C18C	Target Register (CCM_TARGET_ROOT131_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C190	Miscellaneous Register (CCM_MISC131)	32	R/W	0000_0000h	5.1.7.14/469
3038_C194	Miscellaneous Register (CCM_MISC_ROOT131_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C198	Miscellaneous Register (CCM_MISC_ROOT131_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C19C	Miscellaneous Register (CCM_MISC_ROOT131_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_C1A0	Post Divider Register (CCM_POST131)	32	R/W	0000_0000h	5.1.7.18/473
3038_C1A4	Post Divider Register (CCM_POST_ROOT131_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C1A8	Post Divider Register (CCM_POST_ROOT131_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_C1AC	Post Divider Register (CCM_POST_ROOT131_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C1B0	Pre Divider Register (CCM_PRE131)	32	R/W	1000_0000h	5.1.7.22/485
3038_C1B4	Pre Divider Register (CCM_PRE_ROOT131_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_C1B8	Pre Divider Register (CCM_PRE_ROOT131_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C1BC	Pre Divider Register (CCM_PRE_ROOT131_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C1F0	Access Control Register (CCM_ACCESS_CTRL131)	32	R/W	0000_0000h	5.1.7.26/497
3038_C1F4	Access Control Register (CCM_ACCESS_CTRL_ROOT131_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C1F8	Access Control Register (CCM_ACCESS_CTRL_ROOT131_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_C1FC	Access Control Register (CCM_ACCESS_CTRL_ROOT131_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C200	Target Register (CCM_TARGET_ROOT132)	32	R/W	1000_0000h	5.1.7.10/461
3038_C204	Target Register (CCM_TARGET_ROOT132_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_C208	Target Register (CCM_TARGET_ROOT132_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C20C	Target Register (CCM_TARGET_ROOT132_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C210	Miscellaneous Register (CCM_MISC132)	32	R/W	0000_0000h	5.1.7.14/469
3038_C214	Miscellaneous Register (CCM_MISC_ROOT132_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C218	Miscellaneous Register (CCM_MISC_ROOT132_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C21C	Miscellaneous Register (CCM_MISC_ROOT132_TOG)	32	R/W	0000_0000h	5.1.7.17/472

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C220	Post Divider Register (CCM_POST132)	32	R/W	0000_0000h	5.1.7.18/473
3038_C224	Post Divider Register (CCM_POST_ROOT132_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C228	Post Divider Register (CCM_POST_ROOT132_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_C22C	Post Divider Register (CCM_POST_ROOT132_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C230	Pre Divider Register (CCM_PRE132)	32	R/W	1000_0000h	5.1.7.22/485
3038_C234	Pre Divider Register (CCM_PRE_ROOT132_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_C238	Pre Divider Register (CCM_PRE_ROOT132_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C23C	Pre Divider Register (CCM_PRE_ROOT132_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C270	Access Control Register (CCM_ACCESS_CTRL132)	32	R/W	0000_0000h	5.1.7.26/497
3038_C274	Access Control Register (CCM_ACCESS_CTRL_ROOT132_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C278	Access Control Register (CCM_ACCESS_CTRL_ROOT132_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_C27C	Access Control Register (CCM_ACCESS_CTRL_ROOT132_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C280	Target Register (CCM_TARGET_ROOT133)	32	R/W	1000_0000h	5.1.7.10/461
3038_C284	Target Register (CCM_TARGET_ROOT133_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_C288	Target Register (CCM_TARGET_ROOT133_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C28C	Target Register (CCM_TARGET_ROOT133_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C290	Miscellaneous Register (CCM_MISC133)	32	R/W	0000_0000h	5.1.7.14/469
3038_C294	Miscellaneous Register (CCM_MISC_ROOT133_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C298	Miscellaneous Register (CCM_MISC_ROOT133_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C29C	Miscellaneous Register (CCM_MISC_ROOT133_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_C2A0	Post Divider Register (CCM_POST133)	32	R/W	0000_0000h	5.1.7.18/473
3038_C2A4	Post Divider Register (CCM_POST_ROOT133_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C2A8	Post Divider Register (CCM_POST_ROOT133_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_C2AC	Post Divider Register (CCM_POST_ROOT133_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C2B0	Pre Divider Register (CCM_PRE133)	32	R/W	1000_0000h	5.1.7.22/485
3038_C2B4	Pre Divider Register (CCM_PRE_ROOT133_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_C2B8	Pre Divider Register (CCM_PRE_ROOT133_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C2BC	Pre Divider Register (CCM_PRE_ROOT133_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C2F0	Access Control Register (CCM_ACCESS_CTRL133)	32	R/W	0000_0000h	5.1.7.26/497
3038_C2F4	Access Control Register (CCM_ACCESS_CTRL_ROOT133_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C2F8	Access Control Register (CCM_ACCESS_CTRL_ROOT133_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_C2FC	Access Control Register (CCM_ACCESS_CTRL_ROOT133_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C300	Target Register (CCM_TARGET_ROOT134)	32	R/W	1000_0000h	5.1.7.10/461
3038_C304	Target Register (CCM_TARGET_ROOT134_SET)	32	R/W	0000_0000h	5.1.7.11/463

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C308	Target Register (CCM_TARGET_ROOT134_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C30C	Target Register (CCM_TARGET_ROOT134_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C310	Miscellaneous Register (CCM_MISC134)	32	R/W	0000_0000h	5.1.7.14/469
3038_C314	Miscellaneous Register (CCM_MISC_ROOT134_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C318	Miscellaneous Register (CCM_MISC_ROOT134_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C31C	Miscellaneous Register (CCM_MISC_ROOT134_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_C320	Post Divider Register (CCM_POST134)	32	R/W	0000_0000h	5.1.7.18/473
3038_C324	Post Divider Register (CCM_POST_ROOT134_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C328	Post Divider Register (CCM_POST_ROOT134_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_C32C	Post Divider Register (CCM_POST_ROOT134_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C330	Pre Divider Register (CCM_PRE134)	32	R/W	1000_0000h	5.1.7.22/485
3038_C334	Pre Divider Register (CCM_PRE_ROOT134_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_C338	Pre Divider Register (CCM_PRE_ROOT134_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C33C	Pre Divider Register (CCM_PRE_ROOT134_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C370	Access Control Register (CCM_ACCESS_CTRL134)	32	R/W	0000_0000h	5.1.7.26/497
3038_C374	Access Control Register (CCM_ACCESS_CTRL_ROOT134_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C378	Access Control Register (CCM_ACCESS_CTRL_ROOT134_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_C37C	Access Control Register (CCM_ACCESS_CTRL_ROOT134_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C380	Target Register (CCM_TARGET_ROOT135)	32	R/W	1000_0000h	5.1.7.10/461
3038_C384	Target Register (CCM_TARGET_ROOT135_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_C388	Target Register (CCM_TARGET_ROOT135_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C38C	Target Register (CCM_TARGET_ROOT135_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C390	Miscellaneous Register (CCM_MISC135)	32	R/W	0000_0000h	5.1.7.14/469
3038_C394	Miscellaneous Register (CCM_MISC_ROOT135_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C398	Miscellaneous Register (CCM_MISC_ROOT135_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C39C	Miscellaneous Register (CCM_MISC_ROOT135_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_C3A0	Post Divider Register (CCM_POST135)	32	R/W	0000_0000h	5.1.7.18/473
3038_C3A4	Post Divider Register (CCM_POST_ROOT135_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C3A8	Post Divider Register (CCM_POST_ROOT135_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_C3AC	Post Divider Register (CCM_POST_ROOT135_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C3B0	Pre Divider Register (CCM_PRE135)	32	R/W	1000_0000h	5.1.7.22/485
3038_C3B4	Pre Divider Register (CCM_PRE_ROOT135_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_C3B8	Pre Divider Register (CCM_PRE_ROOT135_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C3BC	Pre Divider Register (CCM_PRE_ROOT135_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C3F0	Access Control Register (CCM_ACCESS_CTRL135)	32	R/W	0000_0000h	5.1.7.26/497

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C3F4	Access Control Register (CCM_ACCESS_CTRL_ROOT135_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C3F8	Access Control Register (CCM_ACCESS_CTRL_ROOT135_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_C3FC	Access Control Register (CCM_ACCESS_CTRL_ROOT135_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C400	Target Register (CCM_TARGET_ROOT136)	32	R/W	1000_0000h	5.1.7.10/461
3038_C404	Target Register (CCM_TARGET_ROOT136_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_C408	Target Register (CCM_TARGET_ROOT136_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C40C	Target Register (CCM_TARGET_ROOT136_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C410	Miscellaneous Register (CCM_MISC136)	32	R/W	0000_0000h	5.1.7.14/469
3038_C414	Miscellaneous Register (CCM_MISC_ROOT136_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C418	Miscellaneous Register (CCM_MISC_ROOT136_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C41C	Miscellaneous Register (CCM_MISC_ROOT136_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_C420	Post Divider Register (CCM_POST136)	32	R/W	0000_0000h	5.1.7.18/473
3038_C424	Post Divider Register (CCM_POST_ROOT136_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C428	Post Divider Register (CCM_POST_ROOT136_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_C42C	Post Divider Register (CCM_POST_ROOT136_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C430	Pre Divider Register (CCM_PRE136)	32	R/W	1000_0000h	5.1.7.22/485
3038_C434	Pre Divider Register (CCM_PRE_ROOT136_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_C438	Pre Divider Register (CCM_PRE_ROOT136_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C43C	Pre Divider Register (CCM_PRE_ROOT136_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C470	Access Control Register (CCM_ACCESS_CTRL136)	32	R/W	0000_0000h	5.1.7.26/497
3038_C474	Access Control Register (CCM_ACCESS_CTRL_ROOT136_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C478	Access Control Register (CCM_ACCESS_CTRL_ROOT136_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_C47C	Access Control Register (CCM_ACCESS_CTRL_ROOT136_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C480	Target Register (CCM_TARGET_ROOT137)	32	R/W	1000_0000h	5.1.7.10/461
3038_C484	Target Register (CCM_TARGET_ROOT137_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_C488	Target Register (CCM_TARGET_ROOT137_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C48C	Target Register (CCM_TARGET_ROOT137_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C490	Miscellaneous Register (CCM_MISC137)	32	R/W	0000_0000h	5.1.7.14/469
3038_C494	Miscellaneous Register (CCM_MISC_ROOT137_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C498	Miscellaneous Register (CCM_MISC_ROOT137_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C49C	Miscellaneous Register (CCM_MISC_ROOT137_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_C4A0	Post Divider Register (CCM_POST137)	32	R/W	0000_0000h	5.1.7.18/473
3038_C4A4	Post Divider Register (CCM_POST_ROOT137_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C4A8	Post Divider Register (CCM_POST_ROOT137_CLR)	32	R/W	0000_0000h	5.1.7.20/479

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C4AC	Post Divider Register (CCM_POST_ROOT137_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C4B0	Pre Divider Register (CCM_PRE137)	32	R/W	1000_0000h	5.1.7.22/485
3038_C4B4	Pre Divider Register (CCM_PRE_ROOT137_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_C4B8	Pre Divider Register (CCM_PRE_ROOT137_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C4BC	Pre Divider Register (CCM_PRE_ROOT137_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C4F0	Access Control Register (CCM_ACCESS_CTRL137)	32	R/W	0000_0000h	5.1.7.26/497
3038_C4F4	Access Control Register (CCM_ACCESS_CTRL_ROOT137_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C4F8	Access Control Register (CCM_ACCESS_CTRL_ROOT137_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_C4FC	Access Control Register (CCM_ACCESS_CTRL_ROOT137_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C500	Target Register (CCM_TARGET_ROOT138)	32	R/W	1000_0000h	5.1.7.10/461
3038_C504	Target Register (CCM_TARGET_ROOT138_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_C508	Target Register (CCM_TARGET_ROOT138_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C50C	Target Register (CCM_TARGET_ROOT138_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C510	Miscellaneous Register (CCM_MISC138)	32	R/W	0000_0000h	5.1.7.14/469
3038_C514	Miscellaneous Register (CCM_MISC_ROOT138_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C518	Miscellaneous Register (CCM_MISC_ROOT138_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C51C	Miscellaneous Register (CCM_MISC_ROOT138_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_C520	Post Divider Register (CCM_POST138)	32	R/W	0000_0000h	5.1.7.18/473
3038_C524	Post Divider Register (CCM_POST_ROOT138_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C528	Post Divider Register (CCM_POST_ROOT138_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_C52C	Post Divider Register (CCM_POST_ROOT138_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C530	Pre Divider Register (CCM_PRE138)	32	R/W	1000_0000h	5.1.7.22/485
3038_C534	Pre Divider Register (CCM_PRE_ROOT138_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_C538	Pre Divider Register (CCM_PRE_ROOT138_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C53C	Pre Divider Register (CCM_PRE_ROOT138_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C570	Access Control Register (CCM_ACCESS_CTRL138)	32	R/W	0000_0000h	5.1.7.26/497
3038_C574	Access Control Register (CCM_ACCESS_CTRL_ROOT138_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C578	Access Control Register (CCM_ACCESS_CTRL_ROOT138_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_C57C	Access Control Register (CCM_ACCESS_CTRL_ROOT138_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C580	Target Register (CCM_TARGET_ROOT139)	32	R/W	1000_0000h	5.1.7.10/461
3038_C584	Target Register (CCM_TARGET_ROOT139_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_C588	Target Register (CCM_TARGET_ROOT139_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C58C	Target Register (CCM_TARGET_ROOT139_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C590	Miscellaneous Register (CCM_MISC139)	32	R/W	0000_0000h	5.1.7.14/469

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C594	Miscellaneous Register (CCM_MISC_ROOT139_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C598	Miscellaneous Register (CCM_MISC_ROOT139_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C59C	Miscellaneous Register (CCM_MISC_ROOT139_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_C5A0	Post Divider Register (CCM_POST139)	32	R/W	0000_0000h	5.1.7.18/473
3038_C5A4	Post Divider Register (CCM_POST_ROOT139_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C5A8	Post Divider Register (CCM_POST_ROOT139_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_C5AC	Post Divider Register (CCM_POST_ROOT139_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C5B0	Pre Divider Register (CCM_PRE139)	32	R/W	1000_0000h	5.1.7.22/485
3038_C5B4	Pre Divider Register (CCM_PRE_ROOT139_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_C5B8	Pre Divider Register (CCM_PRE_ROOT139_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C5BC	Pre Divider Register (CCM_PRE_ROOT139_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C5F0	Access Control Register (CCM_ACCESS_CTRL139)	32	R/W	0000_0000h	5.1.7.26/497
3038_C5F4	Access Control Register (CCM_ACCESS_CTRL_ROOT139_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C5F8	Access Control Register (CCM_ACCESS_CTRL_ROOT139_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_C5FC	Access Control Register (CCM_ACCESS_CTRL_ROOT139_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C600	Target Register (CCM_TARGET_ROOT140)	32	R/W	1000_0000h	5.1.7.10/461
3038_C604	Target Register (CCM_TARGET_ROOT140_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_C608	Target Register (CCM_TARGET_ROOT140_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C60C	Target Register (CCM_TARGET_ROOT140_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C610	Miscellaneous Register (CCM_MISC140)	32	R/W	0000_0000h	5.1.7.14/469
3038_C614	Miscellaneous Register (CCM_MISC_ROOT140_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C618	Miscellaneous Register (CCM_MISC_ROOT140_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C61C	Miscellaneous Register (CCM_MISC_ROOT140_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_C620	Post Divider Register (CCM_POST140)	32	R/W	0000_0000h	5.1.7.18/473
3038_C624	Post Divider Register (CCM_POST_ROOT140_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C628	Post Divider Register (CCM_POST_ROOT140_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_C62C	Post Divider Register (CCM_POST_ROOT140_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C630	Pre Divider Register (CCM_PRE140)	32	R/W	1000_0000h	5.1.7.22/485
3038_C634	Pre Divider Register (CCM_PRE_ROOT140_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_C638	Pre Divider Register (CCM_PRE_ROOT140_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C63C	Pre Divider Register (CCM_PRE_ROOT140_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C670	Access Control Register (CCM_ACCESS_CTRL140)	32	R/W	0000_0000h	5.1.7.26/497
3038_C674	Access Control Register (CCM_ACCESS_CTRL_ROOT140_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C678	Access Control Register (CCM_ACCESS_CTRL_ROOT140_CLR)	32	R/W	0000_0000h	5.1.7.28/502

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C67C	Access Control Register (CCM_ACCESS_CTRL_ROOT140_TOG)	32	R/W	0000_0000h	5.1.7.29/505
3038_C680	Target Register (CCM_TARGET_ROOT141)	32	R/W	1000_0000h	5.1.7.10/461
3038_C684	Target Register (CCM_TARGET_ROOT141_SET)	32	R/W	0000_0000h	5.1.7.11/463
3038_C688	Target Register (CCM_TARGET_ROOT141_CLR)	32	R/W	0000_0000h	5.1.7.12/465
3038_C68C	Target Register (CCM_TARGET_ROOT141_TOG)	32	R/W	0000_0000h	5.1.7.13/467
3038_C690	Miscellaneous Register (CCM_MISC141)	32	R/W	0000_0000h	5.1.7.14/469
3038_C694	Miscellaneous Register (CCM_MISC_ROOT141_SET)	32	R/W	0000_0000h	5.1.7.15/470
3038_C698	Miscellaneous Register (CCM_MISC_ROOT141_CLR)	32	R/W	0000_0000h	5.1.7.16/471
3038_C69C	Miscellaneous Register (CCM_MISC_ROOT141_TOG)	32	R/W	0000_0000h	5.1.7.17/472
3038_C6A0	Post Divider Register (CCM_POST141)	32	R/W	0000_0000h	5.1.7.18/473
3038_C6A4	Post Divider Register (CCM_POST_ROOT141_SET)	32	R/W	0000_0000h	5.1.7.19/476
3038_C6A8	Post Divider Register (CCM_POST_ROOT141_CLR)	32	R/W	0000_0000h	5.1.7.20/479
3038_C6AC	Post Divider Register (CCM_POST_ROOT141_TOG)	32	R/W	0000_0000h	5.1.7.21/482
3038_C6B0	Pre Divider Register (CCM_PRE141)	32	R/W	1000_0000h	5.1.7.22/485
3038_C6B4	Pre Divider Register (CCM_PRE_ROOT141_SET)	32	R/W	0000_0000h	5.1.7.23/488
3038_C6B8	Pre Divider Register (CCM_PRE_ROOT141_CLR)	32	R/W	0000_0000h	5.1.7.24/491
3038_C6BC	Pre Divider Register (CCM_PRE_ROOT141_TOG)	32	R/W	0000_0000h	5.1.7.25/494
3038_C6F0	Access Control Register (CCM_ACCESS_CTRL141)	32	R/W	0000_0000h	5.1.7.26/497
3038_C6F4	Access Control Register (CCM_ACCESS_CTRL_ROOT141_SET)	32	R/W	0000_0000h	5.1.7.27/499
3038_C6F8	Access Control Register (CCM_ACCESS_CTRL_ROOT141_CLR)	32	R/W	0000_0000h	5.1.7.28/502
3038_C6FC	Access Control Register (CCM_ACCESS_CTRL_ROOT141_TOG)	32	R/W	0000_0000h	5.1.7.29/505

5.1.7.1 General Purpose Register (CCM_GPR0*n*)

GPR0

Address: 3038_0000h base + 0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GP0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_GPR0*n* field descriptions

Field	Description
GP0	Timeout cycle count of ipg_clk, when perform read and write.

5.1.7.2 CCM PLL Control Register (CCM_PLL_CTRL*n*)

See Input Clocks for PLL control mapping.

NOTE

For the SoC to correctly power up after entering DSM, CCM_PLL_CTRLx must not be set to 0x0 or 0x3 for any domain in use.

Address: 3038_0000h base + 800h offset + (16d × i), where i=0d to 38d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	SETTING3		Reserved	Reserved	SETTING2		Reserved	Reserved	SETTING1		Reserved	Reserved	SETTING0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

CCM_PLL_CTRL*n* field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed

Table continues on the next page...

CCM_PLL_CTRL*n* field descriptions (continued)

Field	Description
	01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.3 CCM PLL Control Register (CCM_PLL_CTRL*n*_SET)

See Input Clocks for PLL control mapping.

NOTE

For the SoC to correctly power up after entering DSM, CCM_PLL_CTRLx must not be set to 0x0 or 0x3 for any domain in use.

Address: 3038_0000h base + 804h offset + (16d × i), where i=0d to 38d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	SETTING3		Reserved	Reserved	SETTING2		Reserved	Reserved	SETTING1		Reserved	Reserved	SETTING0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

CCM_PLL_CTRL*n*_SET field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed

Table continues on the next page...

CCM_PLL_CTRL*n*_SET field descriptions (continued)

Field	Description
	01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.4 CCM PLL Control Register (CCM_PLL_CTRL*n*_CLR)

See Input Clocks for PLL control mapping.

NOTE

For the SoC to correctly power up after entering DSM, CCM_PLL_CTRLx must not be set to 0x0 or 0x3 for any domain in use.

Address: 3038_0000h base + 808h offset + (16d × i), where i=0d to 38d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	SETTING3		Reserved	Reserved	SETTING2		Reserved	Reserved	SETTING1		Reserved	Reserved	SETTING0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

CCM_PLL_CTRL*n*_CLR field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed

Table continues on the next page...

CCM_PLL_CTRL*n*_CLR field descriptions (continued)

Field	Description
	01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.5 CCM PLL Control Register (CCM_PLL_CTRL*n*_TOG)

See Input Clocks for PLL control mapping.

NOTE

For the SoC to correctly power up after entering DSM, CCM_PLL_CTRLx must not be set to 0x0 or 0x3 for any domain in use.

Address: 3038_0000h base + 80Ch offset + (16d × i), where i=0d to 38d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	SETTING3		Reserved	Reserved	SETTING2		Reserved	Reserved	SETTING1		Reserved	Reserved	SETTING0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

CCM_PLL_CTRL*n*_TOG field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed

Table continues on the next page...

CCM_PLL_CTRL*n*_TOG field descriptions (continued)

Field	Description
	01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.6 CCM Clock Gating Register (CCM_CCGR*n*)

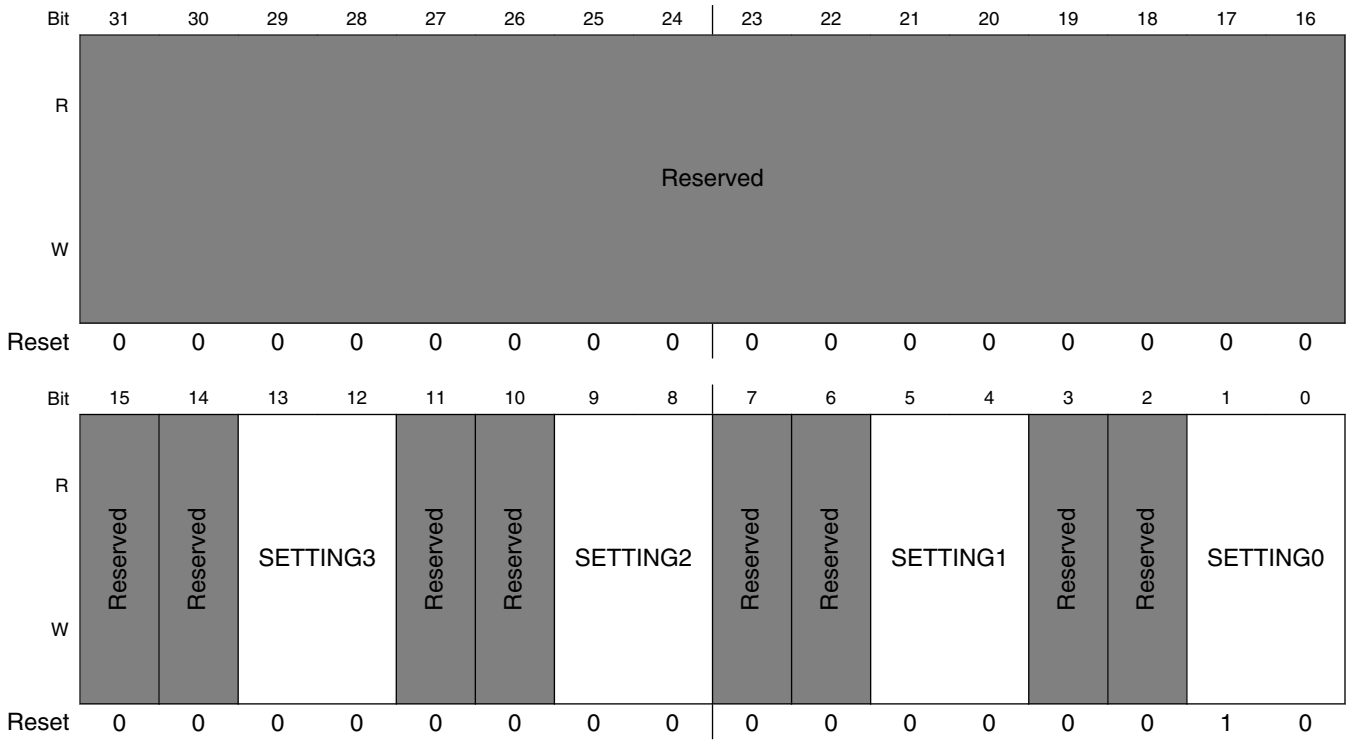
NOTE

Not all CCGRs are mapped. See CCGR Interface for CCGR mapping and clock gating information.

NOTE

Sec_debug clock gating (CCGR60) must be active in low power mode. DO NOT gate this clock in low power mode to guarantee the low power mode functions such as stop WDOG counting.

Address: 3038_0000h base + 4000h offset + (16d × i), where i=0d to 191d



CCM_CCGR*n* field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
11 -	This field is reserved. Reserved

Table continues on the next page...

CCM_CCGR*n* field descriptions (continued)

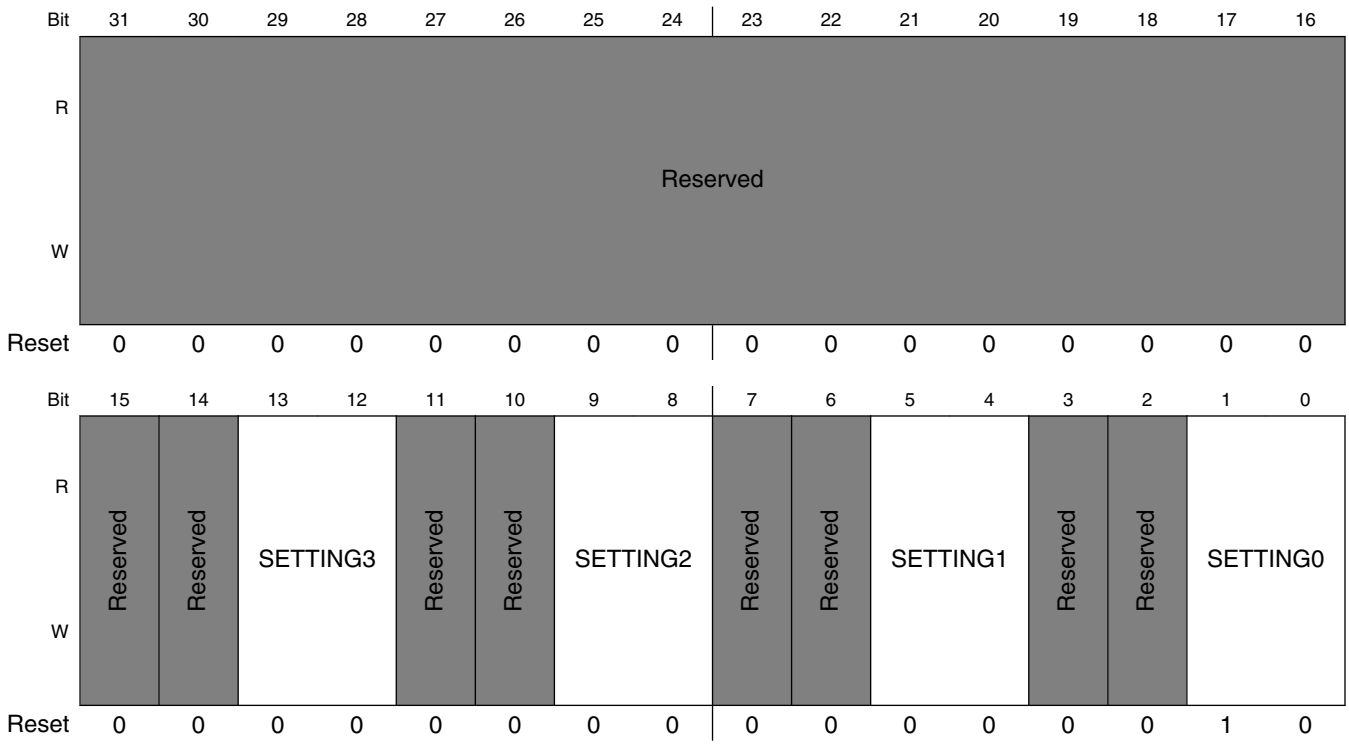
Field	Description
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.7 CCM Clock Gating Register (CCM_CCGR*n*_SET)

NOTE

Not all CCGRs are mapped. See CCGR Interface for CCGR mapping and clock gating information.

Address: 3038_0000h base + 4004h offset + (16d × i), where i=0d to 191d



CCM_CCGR*n*_SET field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

Table continues on the next page...

CCM_CCGR*n*_SET field descriptions (continued)

Field	Description
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.8 CCM Clock Gating Register (CCM_CCGR*n*_CLR)

NOTE

Not all CCGRs are mapped. See CCGR Interface for CCGR mapping and clock gating information.

Address: 3038_0000h base + 4008h offset + (16d × i), where i=0d to 191d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	SETTING3		Reserved	Reserved	SETTING2		Reserved	Reserved	SETTING1		Reserved	Reserved	SETTING0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

CCM_CCGR*n*_CLR field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

Table continues on the next page...

CCM_CCGR*n*_CLR field descriptions (continued)

Field	Description
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.9 CCM Clock Gating Register (CCM_CCGR*n*_TOG)

NOTE

Not all CCGRs are mapped. See CCGR Interface for CCGR mapping and clock gating information.

Address: 3038_0000h base + 400Ch offset + (16d × i), where i=0d to 191d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	SETTING3		Reserved	Reserved	SETTING2		Reserved	Reserved	SETTING1		Reserved	Reserved	SETTING0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

CCM_CCGR*n*_TOG field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

Table continues on the next page...

CCM_CCGR*n*_TOG field descriptions (continued)

Field	Description
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

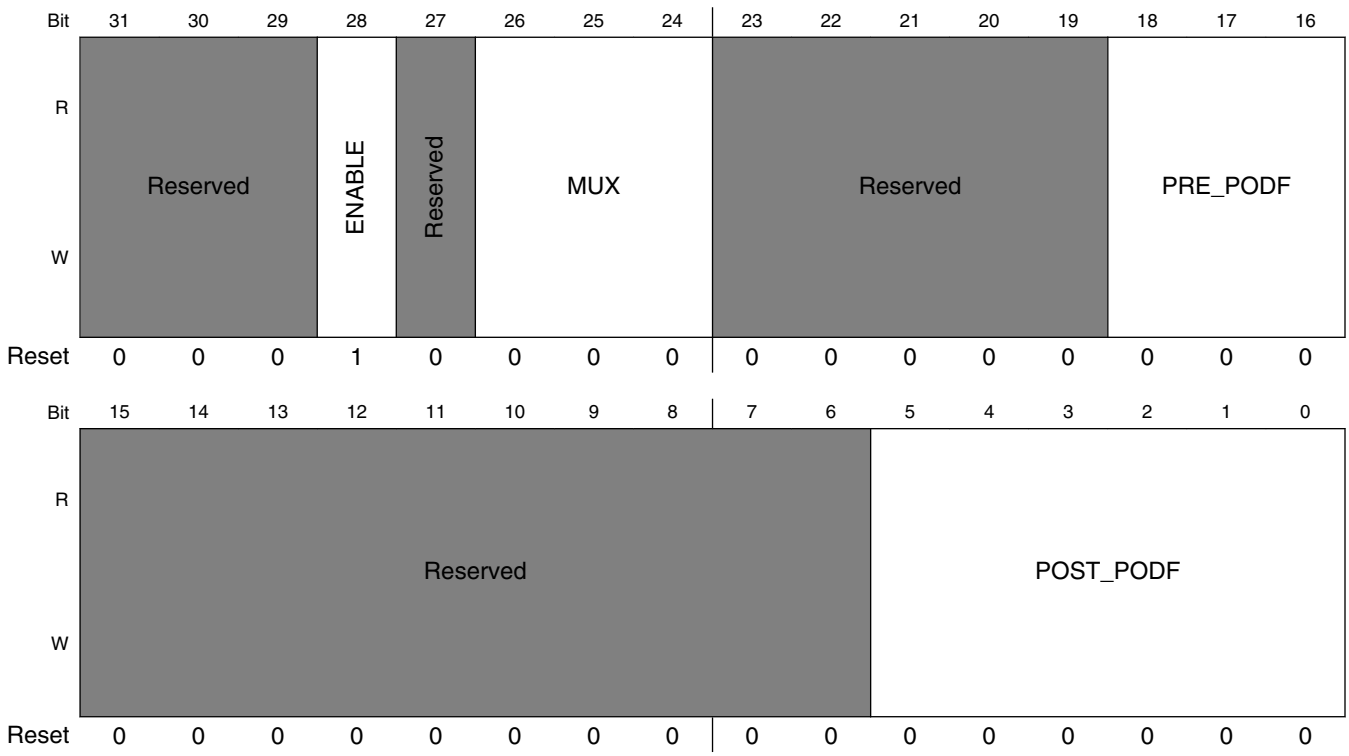
5.1.7.10 Target Register (CCM_TARGET_ROOT*n*)

See Target Interface for more information.

NOTE

See Clock Root Selects for clock root offsets and muxing information.

Address: 3038_0000h base + 8000h offset + (128d × i), where i=0d to 141d



CCM_TARGET_ROOT*n* field descriptions

Field	Description
31–29 -	This field is reserved. Reserved
28 ENABLE	Enable this clock 0 clock root is OFF 1 clock root is ON
27 -	This field is reserved. Reserved
26–24 MUX	Selection of clock sources Please see Clock Root Selects for clock root offsets and muxing information. This field is 1 bit long for DRAM and CORE

Table continues on the next page...

CCM_TARGET_ROOT*n* field descriptions (continued)

Field	Description
23–19 -	This field is reserved. Reserved
18–16 PRE_PODF	Pre divider divide the number Divider value is n+1 This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15–6 -	This field is reserved. Reserved
POST_PODF	Post divider divide number Divider value is n + 1. For CORE, this field is 3 bit long. For Peripheral (IP), this field is 1 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

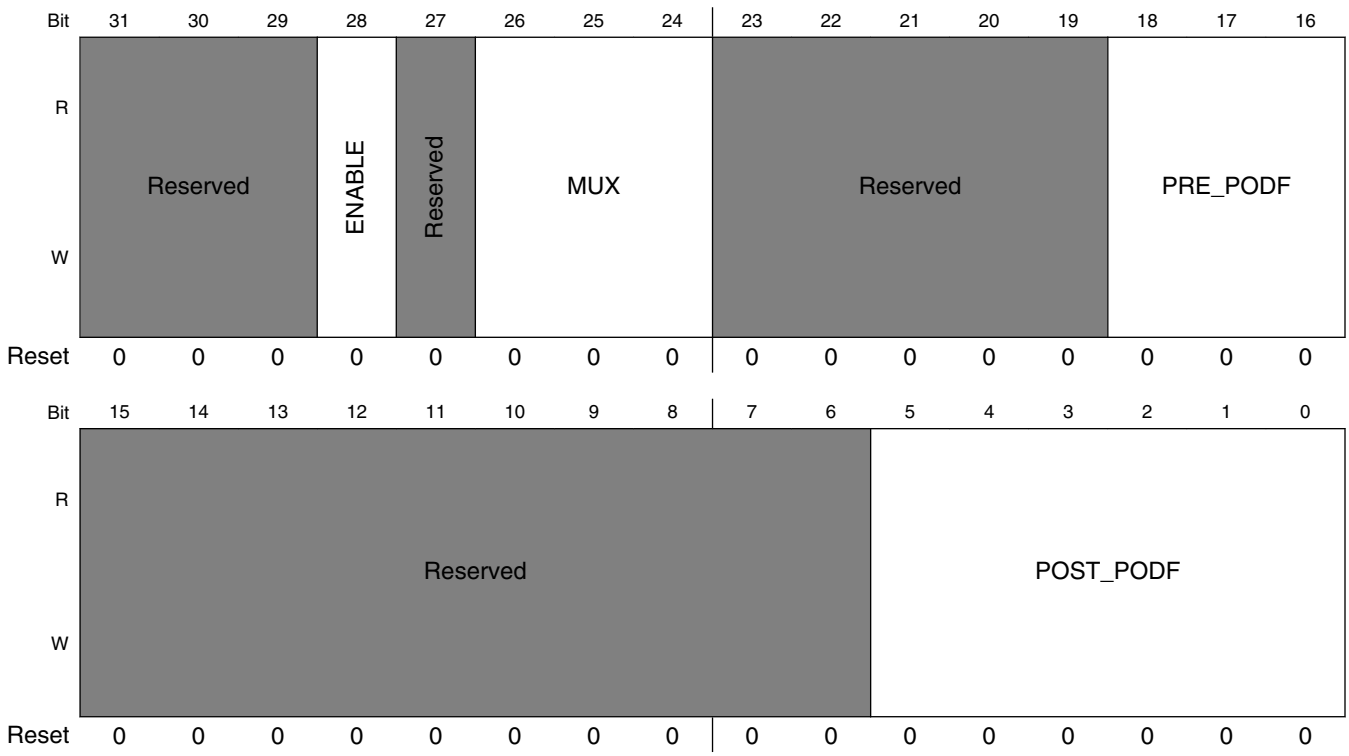
5.1.7.11 Target Register (CCM_TARGET_ROOT*n*_SET)

See Target Interface for more information.

NOTE

See Clock Root Selects for clock root offsets and muxing information.

Address: 3038_0000h base + 8004h offset + (128d × i), where i=0d to 141d



CCM_TARGET_ROOT*n*_SET field descriptions

Field	Description
31–29 -	This field is reserved. Reserved
28 ENABLE	Enable this clock 0 clock root is OFF 1 clock root is ON
27 -	This field is reserved. Reserved
26–24 MUX	Selection of clock sources Please see Clock Root Selects for clock root offsets and muxing information. This field is 1 bit long for DRAM and CORE

Table continues on the next page...

CCM_TARGET_ROOT*n*_SET field descriptions (continued)

Field	Description
23–19 -	This field is reserved. Reserved
18–16 PRE_PODF	Pre divider divide the number Divider value is n+1 This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15–6 -	This field is reserved. Reserved
POST_PODF	Post divider divide the number Divider value is n + 1. For CORE, this field is 3 bit long. For Peripheral (IP), this field is 1 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

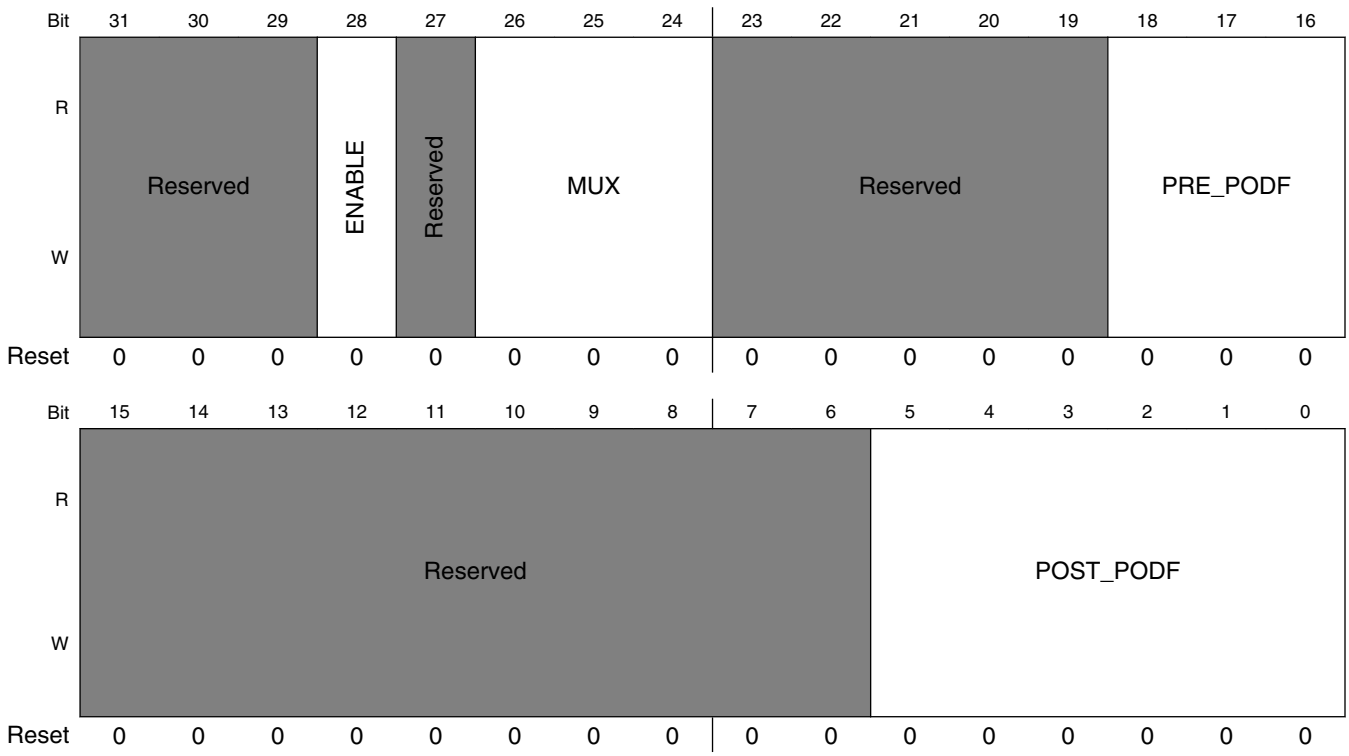
5.1.7.12 Target Register (CCM_TARGET_ROOT*n*_CLR)

See Target Interface for more information.

NOTE

See Clock Root Selects for clock root offsets and muxing information.

Address: 3038_0000h base + 8008h offset + (128d × i), where i=0d to 141d



CCM_TARGET_ROOT*n*_CLR field descriptions

Field	Description
31–29 -	This field is reserved. Reserved
28 ENABLE	Enable this clock 0 clock root is OFF 1 clock root is ON
27 -	This field is reserved. Reserved
26–24 MUX	Selection of clock sources Please see Clock Root Selects for clock root offsets and muxing information. This field is 1 bit long for DRAM and CORE

Table continues on the next page...

CCM_TARGET_ROOT*n*_CLR field descriptions (continued)

Field	Description
23–19 -	This field is reserved. Reserved
18–16 PRE_PODF	Pre divider divide the number Divider value is n+1 This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15–6 -	This field is reserved. Reserved
POST_PODF	Post divider divide the number Divider value is n + 1. For CORE, this field is 3 bit long. For Peripheral (IP), this field is 1 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

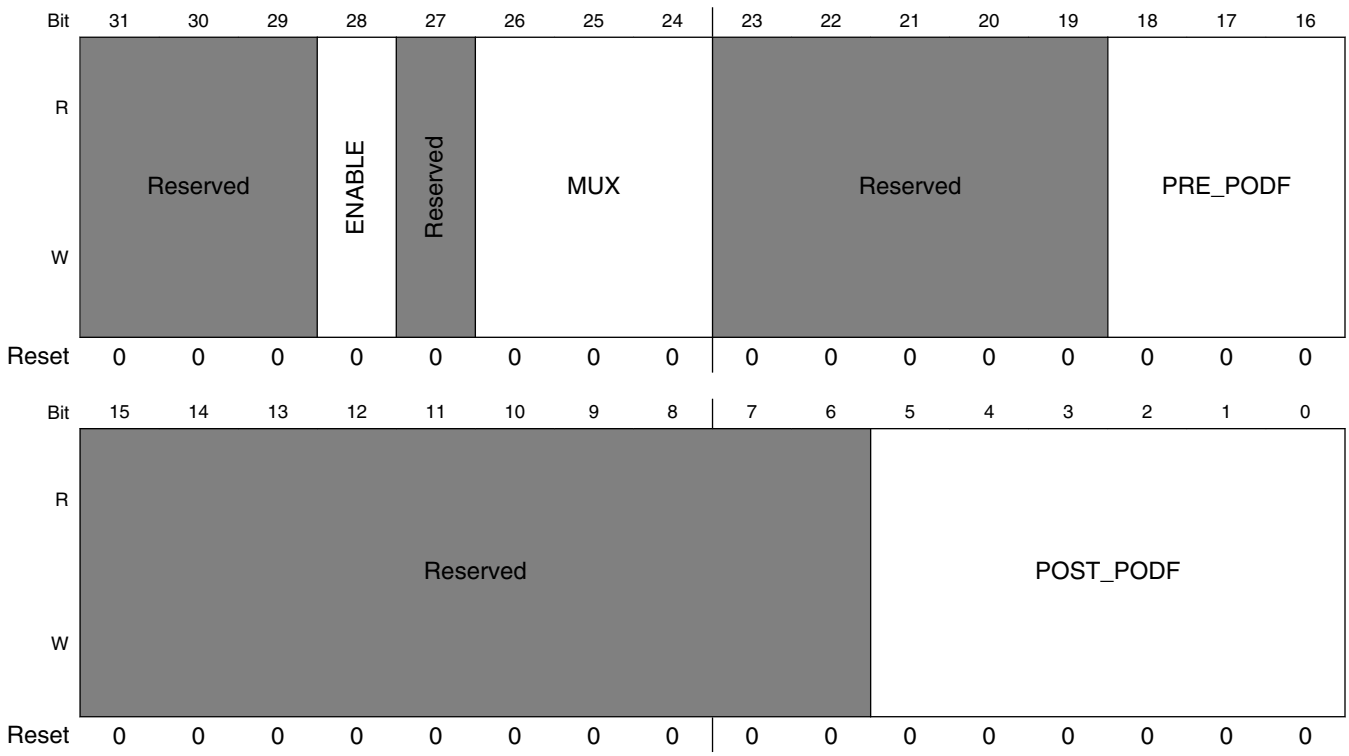
5.1.7.13 Target Register (CCM_TARGET_ROOT*n*_TOG)

See Target Interface for more information.

NOTE

See Clock Root Selects for clock root offsets and muxing information.

Address: 3038_0000h base + 800Ch offset + (128d × i), where i=0d to 141d



CCM_TARGET_ROOT*n*_TOG field descriptions

Field	Description
31–29 -	This field is reserved. Reserved
28 ENABLE	Enable this clock 0 clock root is OFF 1 clock root is ON
27 -	This field is reserved. Reserved
26–24 MUX	Selection of clock sources Please see Clock Root Selects for clock root offsets and muxing information. This field is 1 bit long for DRAM and CORE

Table continues on the next page...

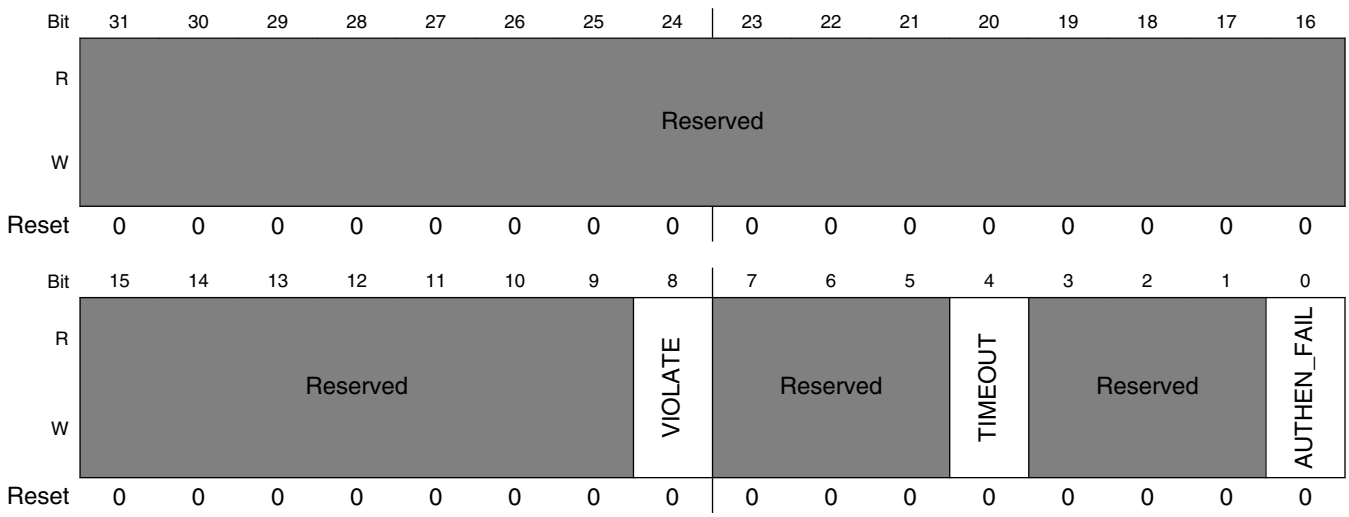
CCM_TARGET_ROOT*n*_TOG field descriptions (continued)

Field	Description
23–19 -	This field is reserved. Reserved
18–16 PRE_PODF	Pre divide divide number Divider value is n+1 This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15–6 -	This field is reserved. Reserved
POST_PODF	Post divider divide number Divider value is n + 1. For CORE, this field is 3 bit long. For Peripheral (IP), this field is 1 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

5.1.7.14 Miscellaneous Register (CCM_MISC*n*)

MISC

Address: 3038_0000h base + 8010h offset + (128d × i), where i=0d to 141d



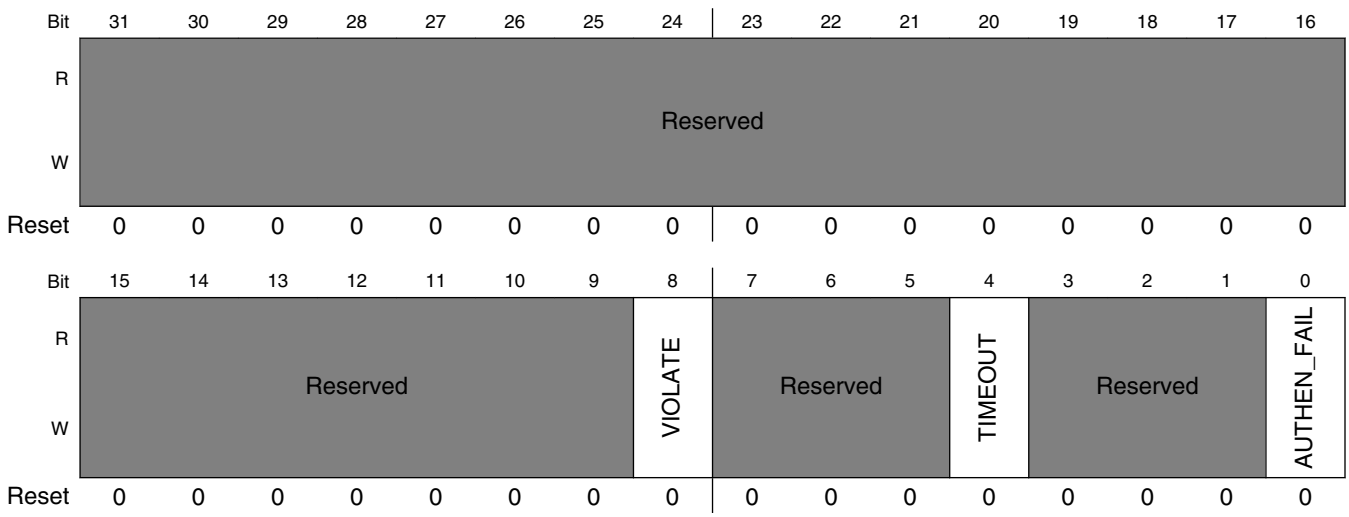
CCM_MISC*n* field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 VIOLATE	This sticky bit reflects access violation in normal interface of this clock. This bit has internal 4 bits, one for each domain. Violation from other domain is not visible or clearable. This file is cleared to 0 while write 1.
7–5 -	This field is reserved. Reserved
4 TIMEOUT	This sticky bit reflects time out happened during accessing this clock. This bit has internal 4 bits, one for each domain. Timeout from other domain is not visible or clearable. This file is cleared to 0 while write 1.
3–1 -	This field is reserved. Reserved
0 AUTHEN_FAIL	This sticky bit reflects access restricted by access control of this clock. This bit has internal 4 bits, one for each domain. Authentic fail from other domain is not visible or clearable. This file is cleared to 0 while write 1

5.1.7.15 Miscellaneous Register (CCM_MISC_ROOT*n*_SET)

Misc

Address: 3038_0000h base + 8014h offset + (128d × i), where i=0d to 141d



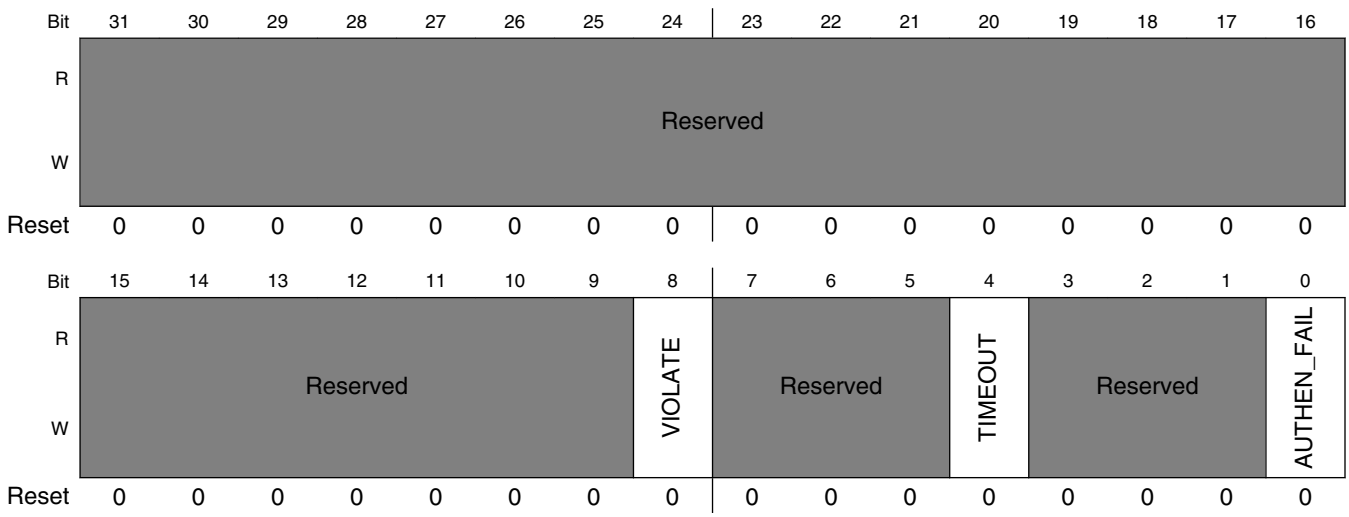
CCM_MISC_ROOT*n*_SET field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 VIOLATE	This sticky bit reflects access violation in normal interface of this clock. This bit has internal 4 bits, one for each domain. Violation from other domain is not visible or clearable. This file is cleared to 0 while write 1.
7–5 -	This field is reserved. Reserved
4 TIMEOUT	This sticky bit reflects time out happened during accessing this clock. This bit has internal 4 bits, one for each domain. Timeout from other domain is not visible or clearable. This file is cleared to 0 while write 1.
3–1 -	This field is reserved. Reserved
0 AUTHEN_FAIL	This sticky bit reflects access restricted by access control of this clock. This bit has internal 4 bits, one for each domain. Authentic fail from other domain is not visible or clearable. This file is cleared to 0 while write 1

5.1.7.16 Miscellaneous Register (CCM_MISC_ROOT*n*_CLR)

MISC

Address: 3038_0000h base + 8018h offset + (128d × i), where i=0d to 141d



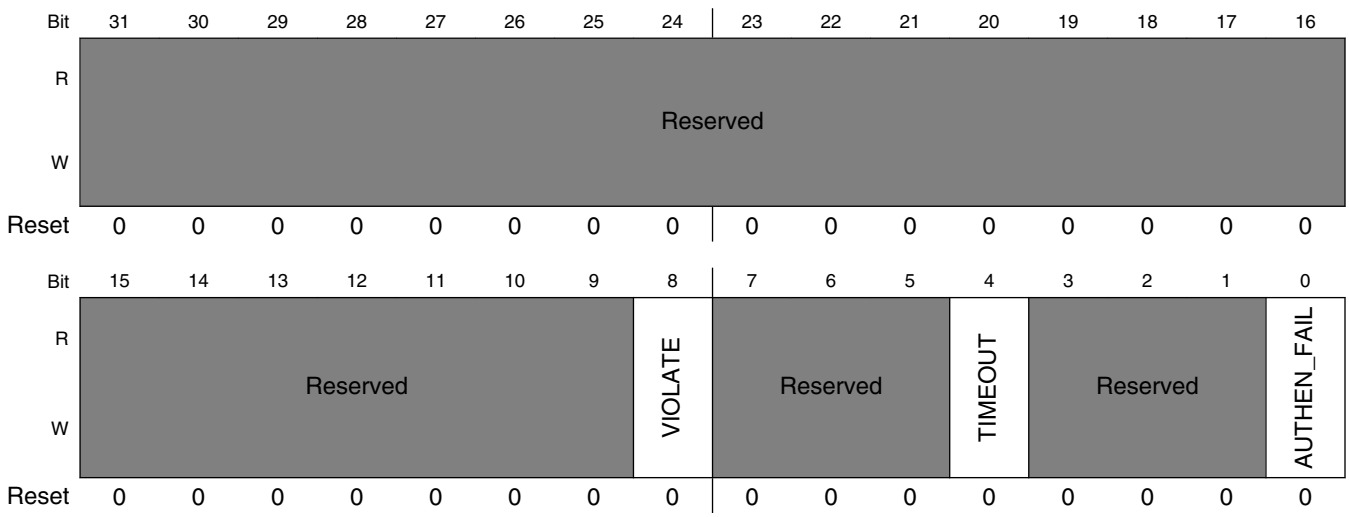
CCM_MISC_ROOT*n*_CLR field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 VIOLATE	This sticky bit reflects access violation in normal interface of this clock. This bit has internal 4 bits, one for each domain. Violation from other domain is not visible or clearable. This file is cleared to 0 while write 1.
7–5 -	This field is reserved. Reserved
4 TIMEOUT	This sticky bit reflects time out happened during accessing this clock. This bit has internal 4 bits, one for each domain. Timeout from other domain is not visible or clearable. This file is cleared to 0 while write 1.
3–1 -	This field is reserved. Reserved
0 AUTHEN_FAIL	This sticky bit reflects access restricted by access control of this clock. This bit has internal 4 bits, one for each domain. Authentic fail from other domain is not visible or clearable. This file is cleared to 0 while write 1

5.1.7.17 Miscellaneous Register (CCM_MISC_ROOT*n*_TOG)

MISC

Address: 3038_0000h base + 801Ch offset + (128d × i), where i=0d to 141d



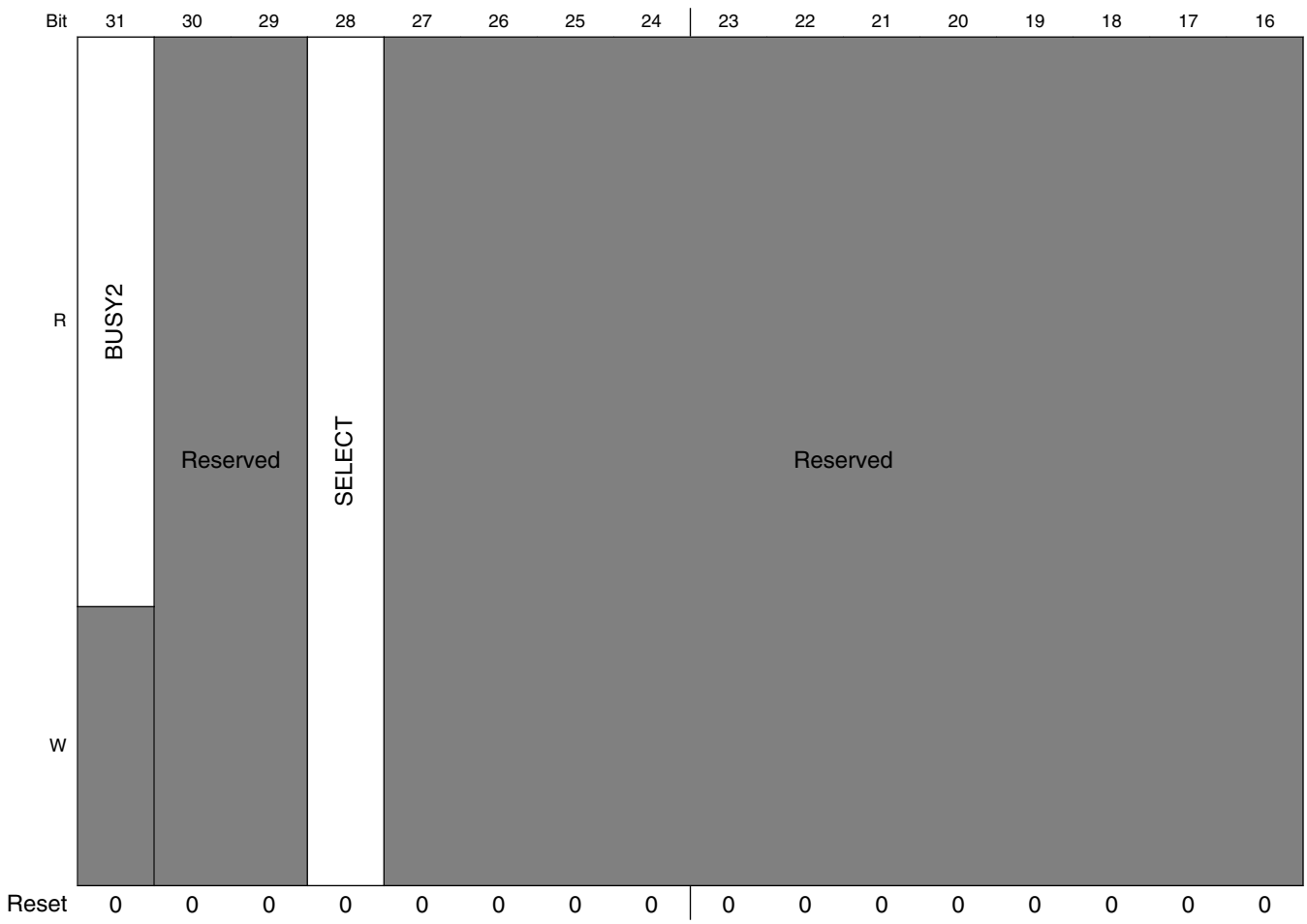
CCM_MISC_ROOT*n*_TOG field descriptions

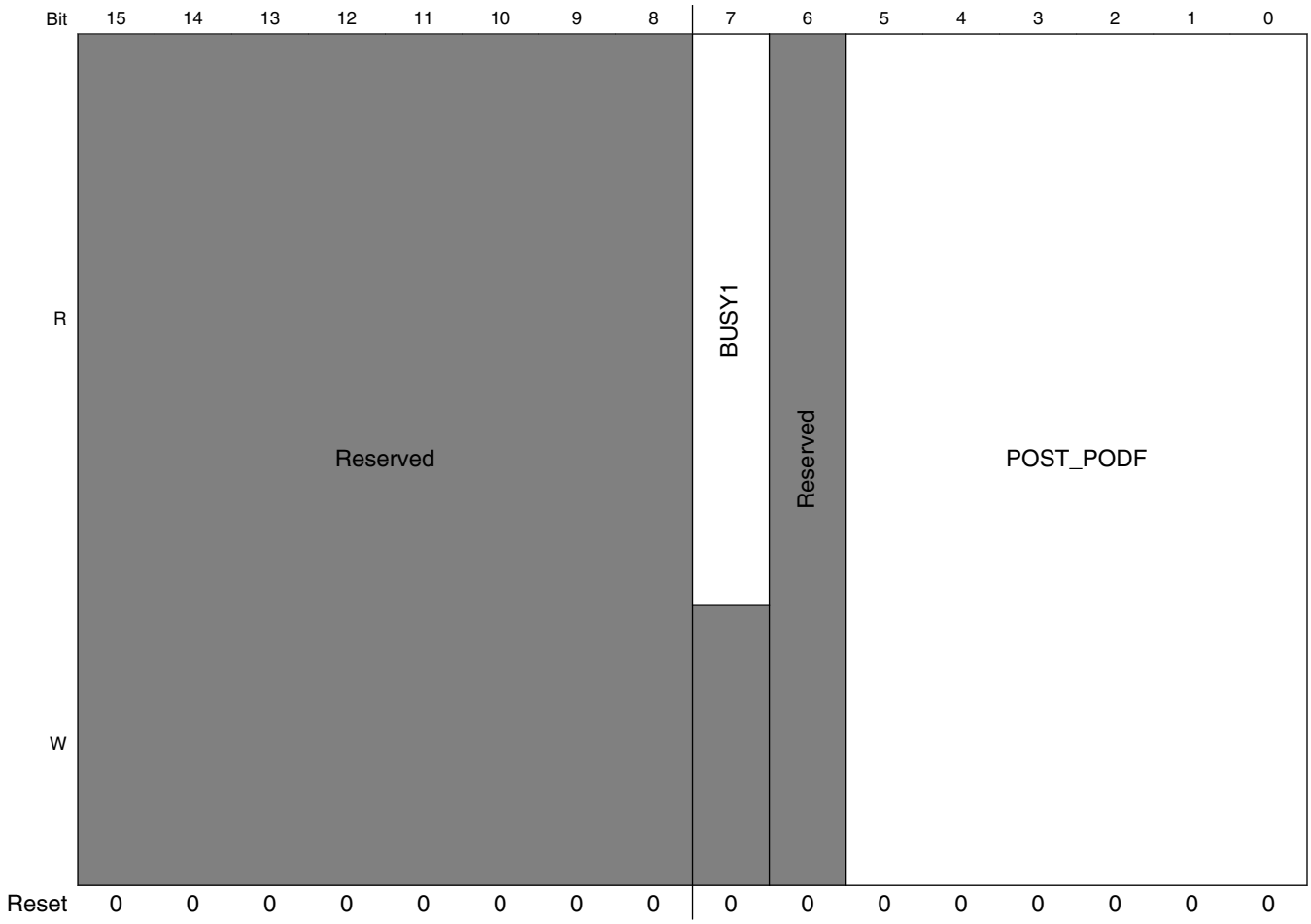
Field	Description
31–9 -	This field is reserved. Reserved
8 VIOLATE	This sticky bit reflects access violation in normal interface of this clock. This bit has internal 4 bits, one for each domain. Violation from other domain is not visible or clearable. This file is cleared to 0 while write 1.
7–5 -	This field is reserved. Reserved
4 TIMEOUT	This sticky bit reflects time out happened during accessing this clock. This bit has internal 4 bits, one for each domain. Timeout from other domain is not visible or clearable. This file is cleared to 0 while write 1.
3–1 -	This field is reserved. Reserved
0 AUTHEN_FAIL	This sticky bit reflects access restricted by access control of this clock. This bit has internal 4 bits, one for each domain. Authentic fail from other domain is not visible or clearable. This file is cleared to 0 while write 1

5.1.7.18 Post Divider Register (CCM_POST*n*)

Post Register

Address: 3038_0000h base + 8020h offset + (128d × i), where i=0d to 141d





CCM_POST*n* field descriptions

Field	Description
31 BUSY2	Clock switching multiplexer is applying new setting
30–29 -	This field is reserved. Reserved
28 SELECT	Selection of post clock branches This field is not applicable to Peripheral (IP) Clock Slice, see Peripheral clock slice Peripheral clock slice 0 select branch A 1 select branch B
27–8 -	This field is reserved. Reserved
7 BUSY1	Post divider is applying new set value
6 -	This field is reserved. Reserved
POST_PODF	Post divider divide the number Divider value is n + 1

Table continues on the next page...

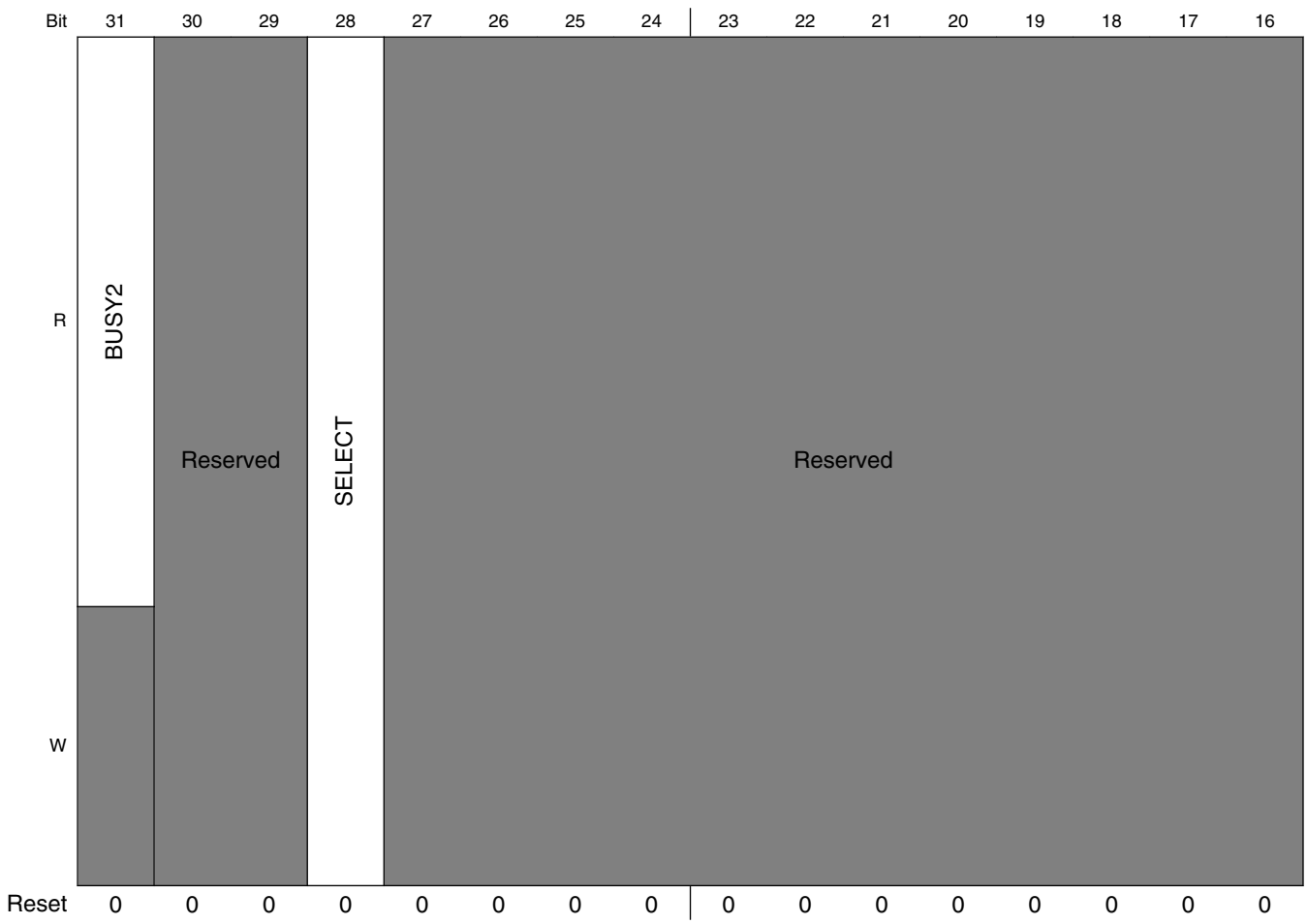
CCM_POST*n* field descriptions (continued)

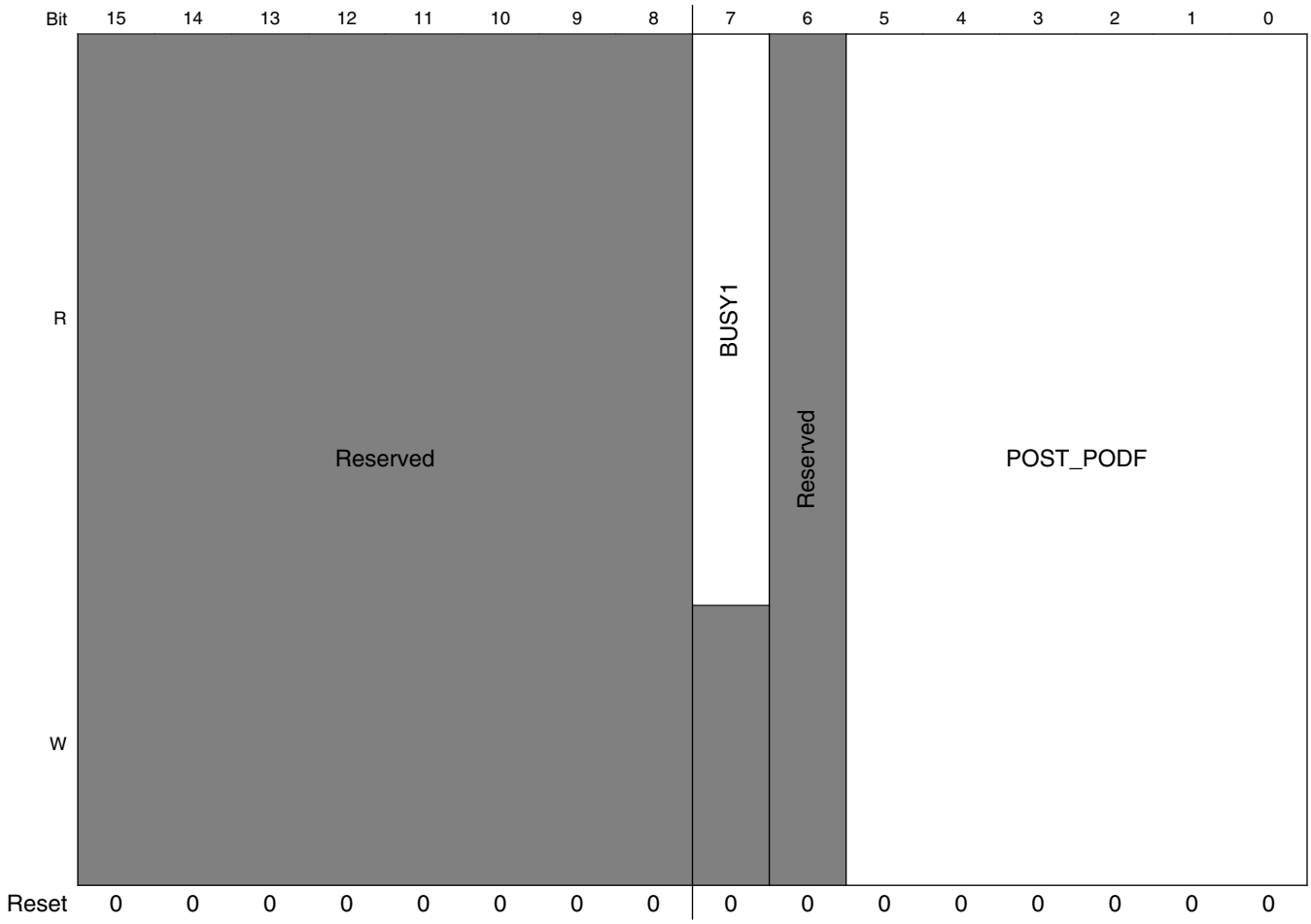
Field	Description
	For CORE, this field is 3 bit long. For Peripheral (IP), this field is 2 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

5.1.7.19 Post Divider Register (CCM_POST_ROOT*n*_SET)

Post Divider Register

Address: 3038_0000h base + 8024h offset + (128d × i), where i=0d to 141d





CCM_POST_ROOT*n*_SET field descriptions

Field	Description
31 BUSY2	Clock switching multiplexer is applying new setting
30–29 -	This field is reserved. Reserved
28 SELECT	Selection of post clock branches This field is not applicable to Peripheral (IP) Clock Slice, see Peripheral clock slice Peripheral clock slice 0 select branch A 1 select branch B
27–8 -	This field is reserved. Reserved
7 BUSY1	Post divider is applying new set value
6 -	This field is reserved. Reserved
POST_PODF	Post divider divide number Divider value is n + 1

Table continues on the next page...

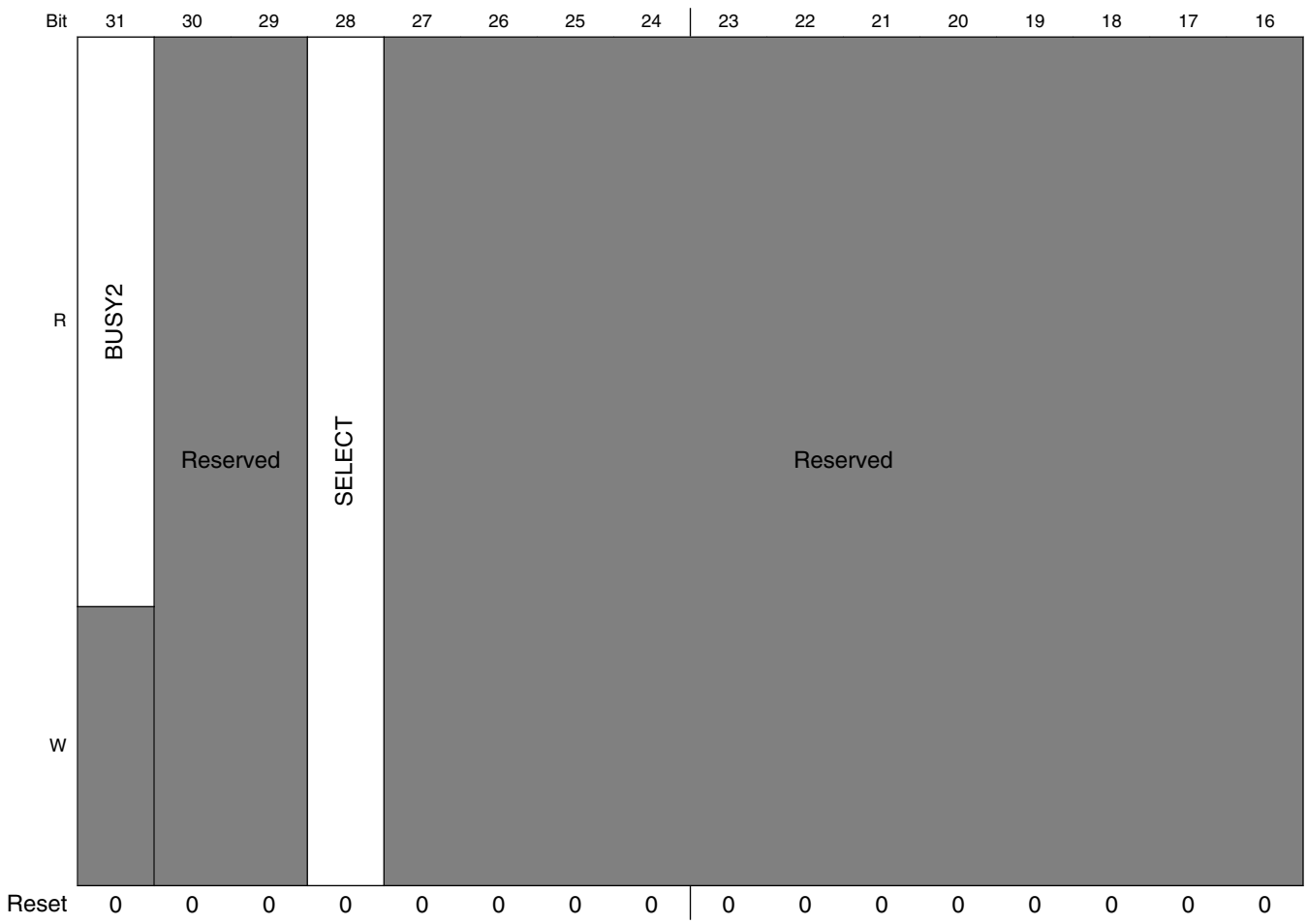
CCM_POST_ROOT*n*_SET field descriptions (continued)

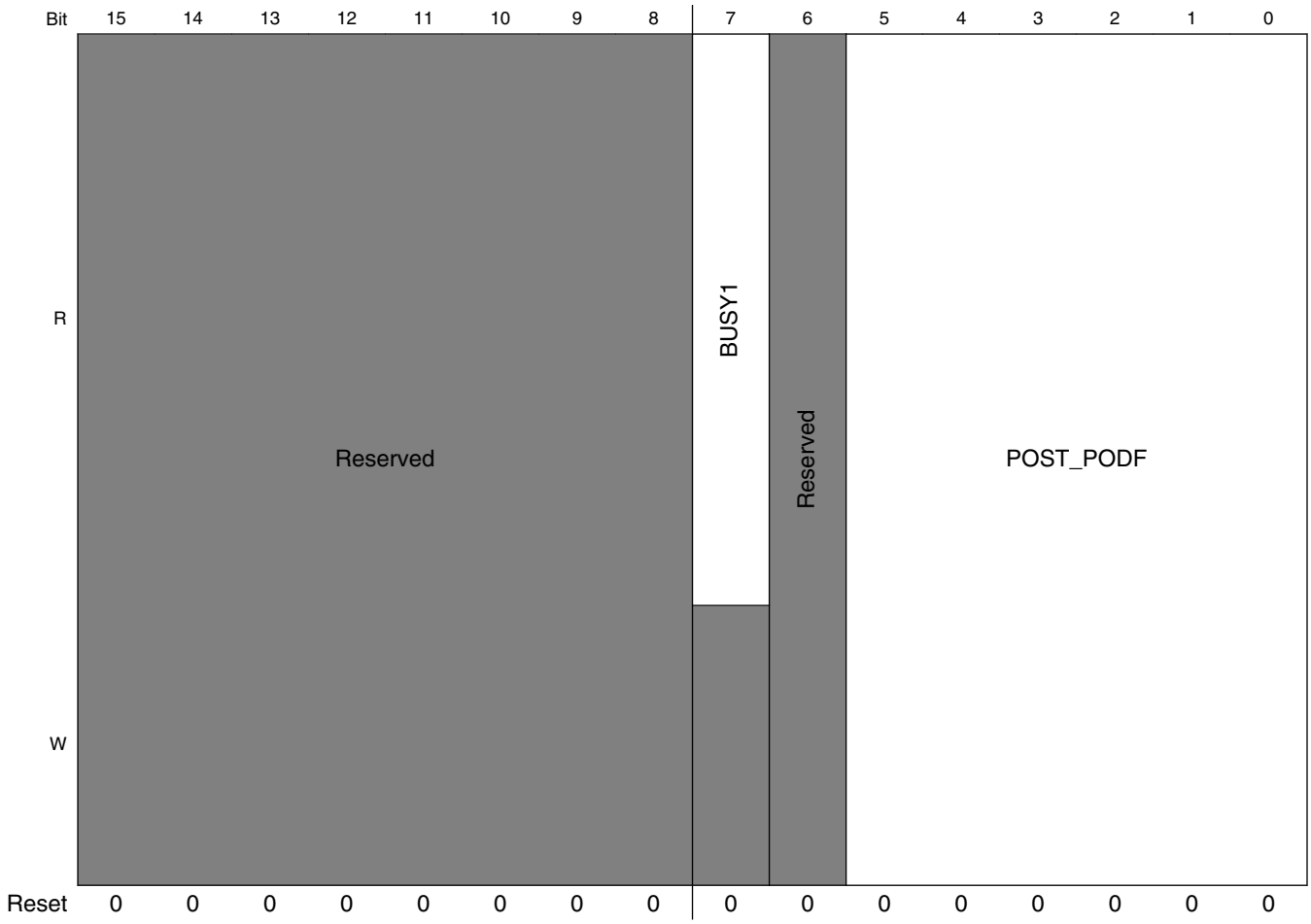
Field	Description
	For CORE, this field is 3 bit long. For Peripheral (IP), this field is 2 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

5.1.7.20 Post Divider Register (CCM_POST_ROOT*n*_CLR)

Post Root Register

Address: 3038_0000h base + 8028h offset + (128d × i), where i=0d to 141d





CCM_POST_ROOT*n*_CLR field descriptions

Field	Description
31 BUSY2	Clock switching multiplexer is applying new setting
30–29 -	This field is reserved. Reserved
28 SELECT	Selection of post clock branches This field is not applicable to Peripheral (IP) Clock Slice, see Peripheral clock slice Peripheral clock slice 0 select branch A 1 select branch B
27–8 -	This field is reserved. Reserved
7 BUSY1	Post divider is applying new set value
6 -	This field is reserved. Reserved
POST_PODF	Post divider divide the number Divider value is n + 1

Table continues on the next page...

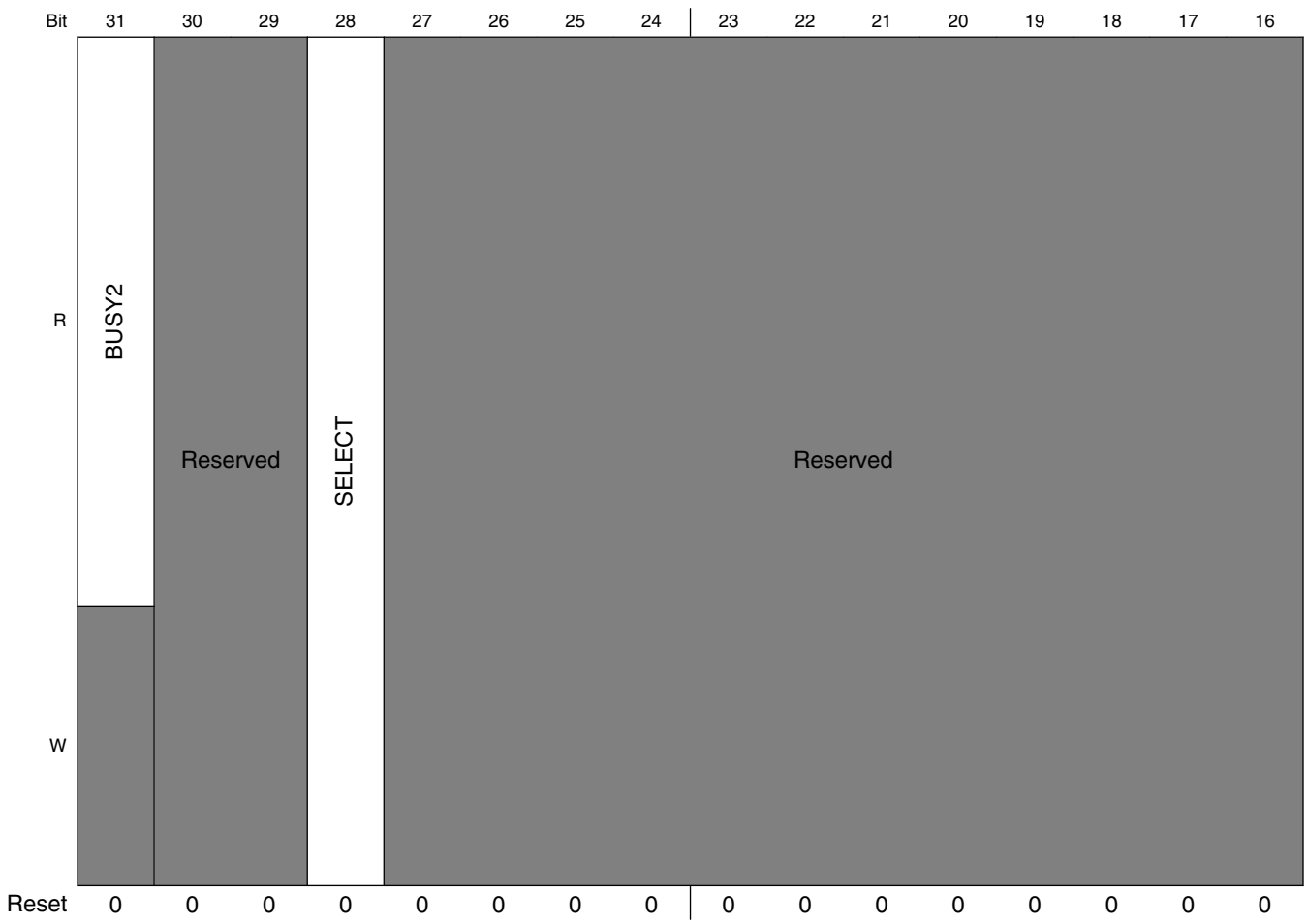
CCM_POST_ROOT*n*_CLR field descriptions (continued)

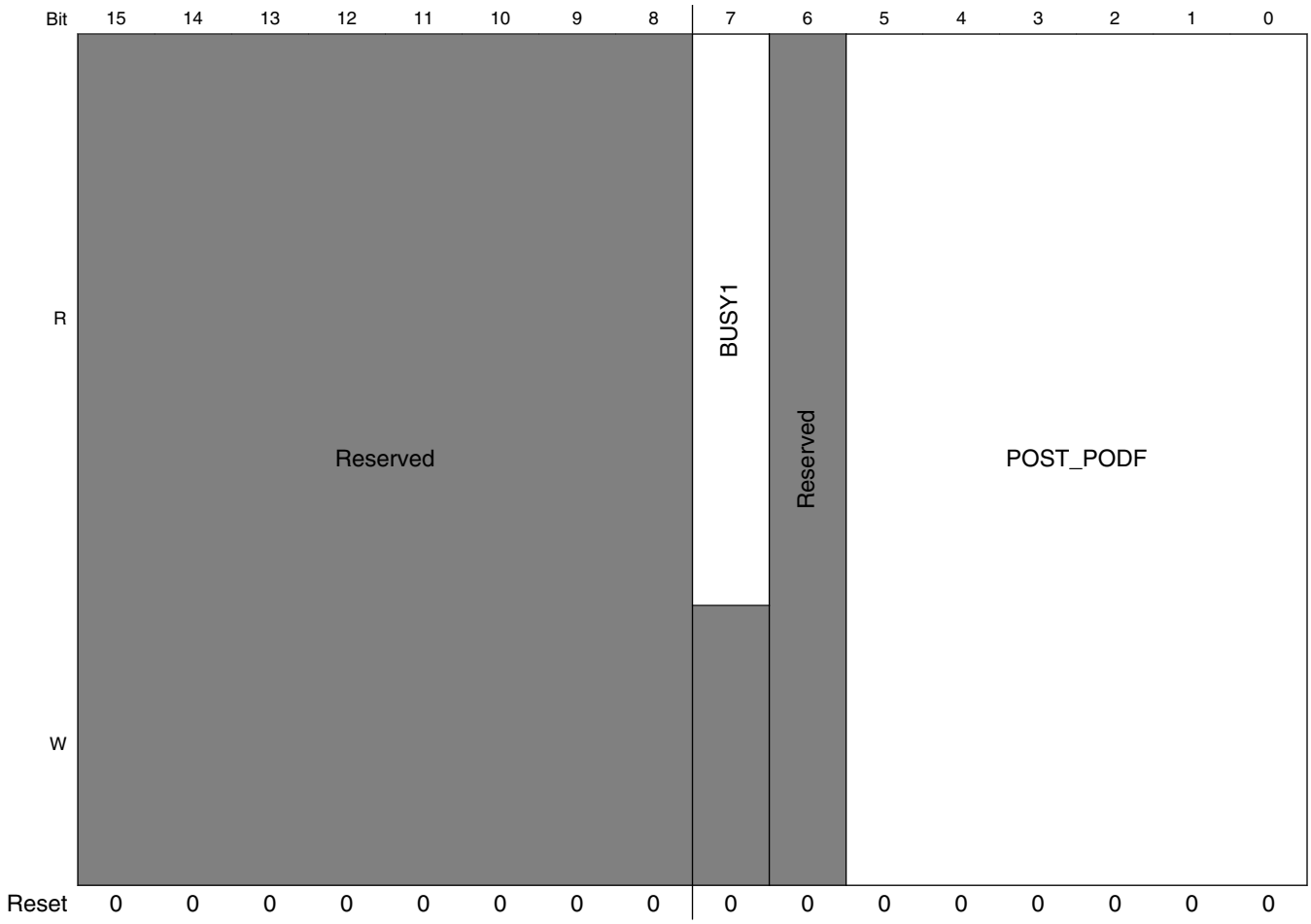
Field	Description
	For CORE, this field is 3 bit long. For Peripheral (IP), this field is 2 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

5.1.7.21 Post Divider Register (CCM_POST_ROOT*n*_TOG)

Post Root Register

Address: 3038_0000h base + 802Ch offset + (128d × i), where i=0d to 141d





CCM_POST_ROOT*n*_TOG field descriptions

Field	Description
31 BUSY2	Clock switching multiplexer is applying new setting
30–29 -	This field is reserved. Reserved
28 SELECT	Selection of post clock branches This field is not applicable to Peripheral (IP) Clock Slice, see Peripheral clock slice Peripheral clock slice 0 select branch A 1 select branch B
27–8 -	This field is reserved. Reserved
7 BUSY1	Post divider is applying new set value
6 -	This field is reserved. Reserved
POST_PODF	Post divider divide number Divider value is n + 1

Table continues on the next page...

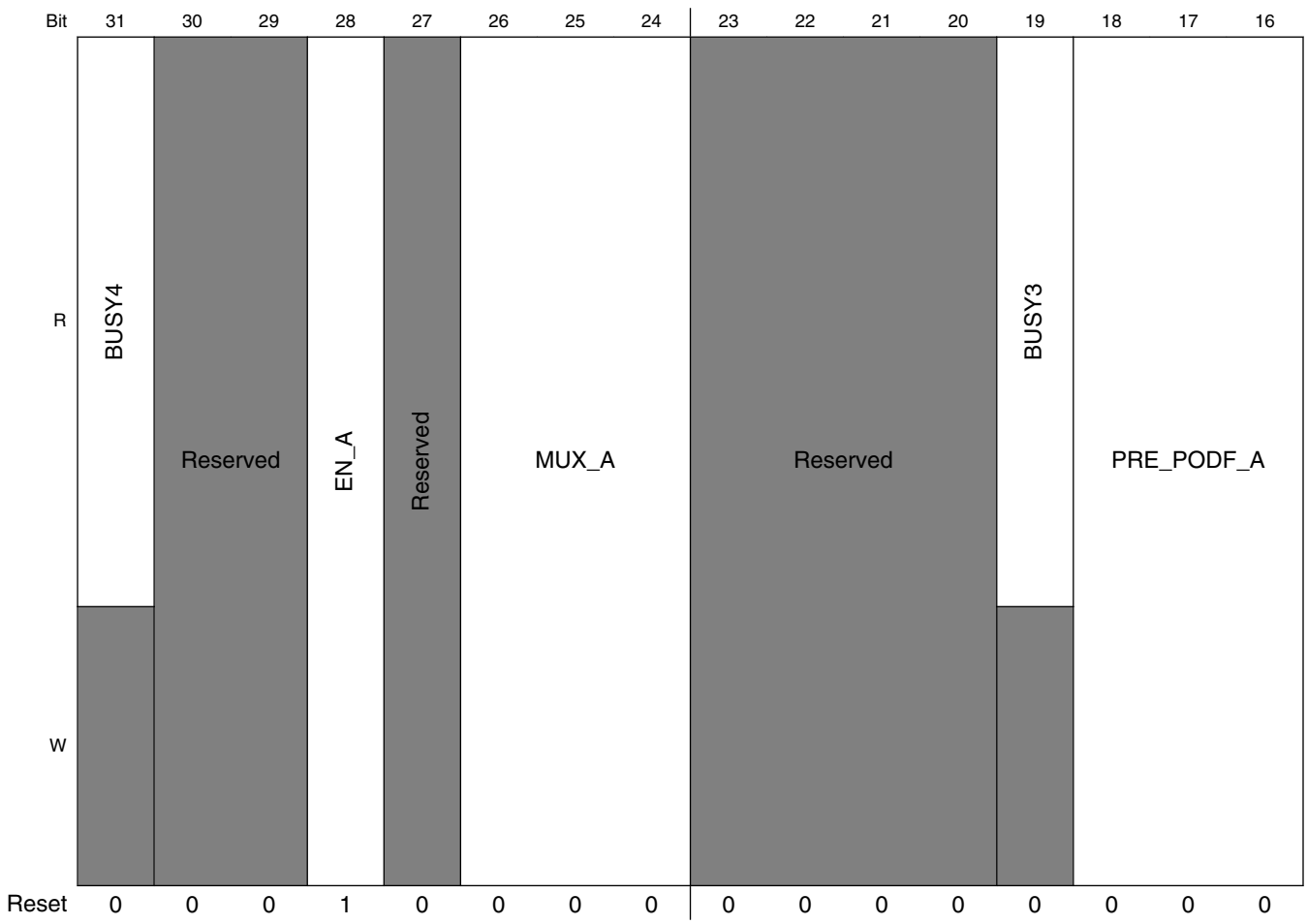
CCM_POST_ROOT*n*_TOG field descriptions (continued)

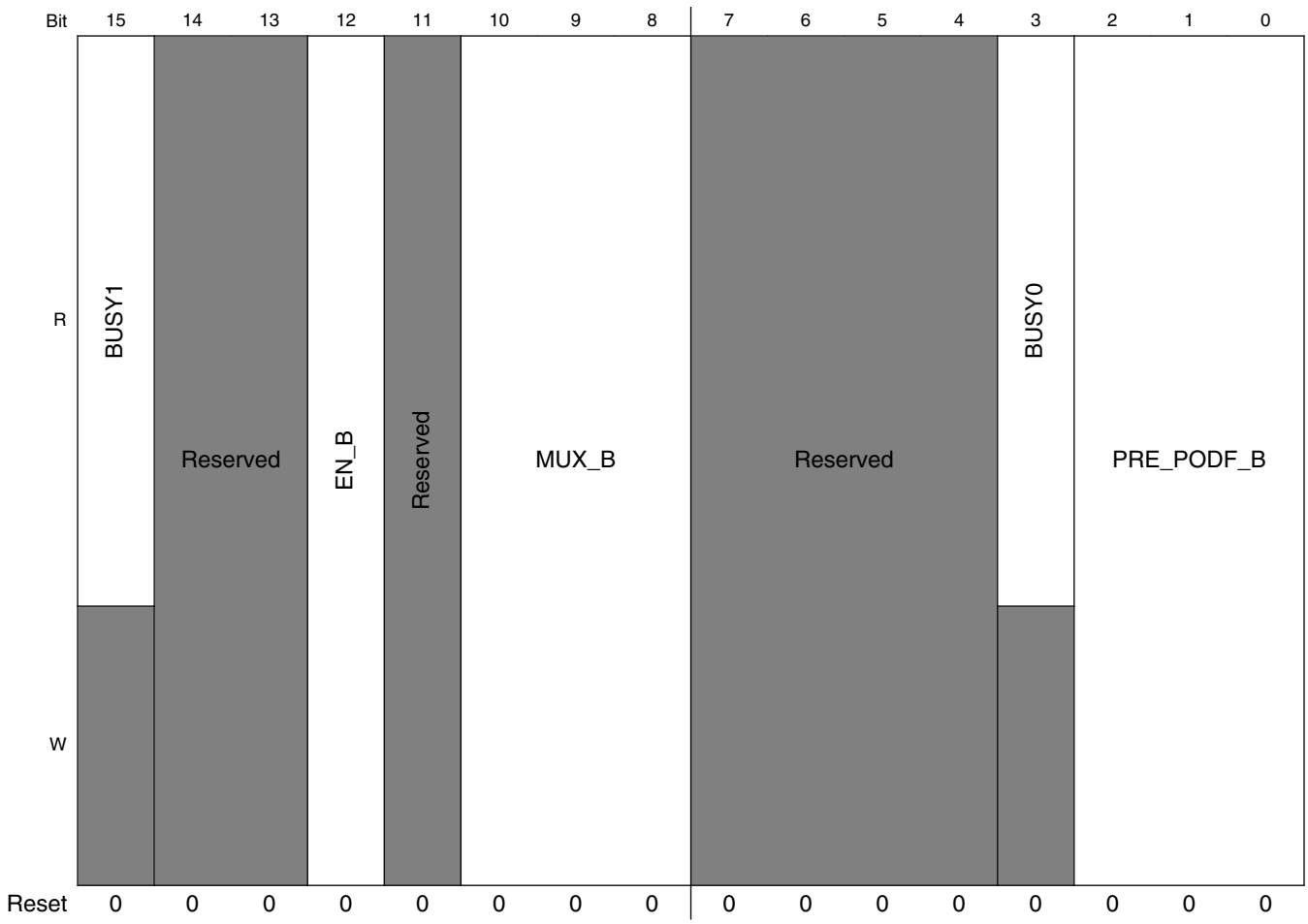
Field	Description
	For CORE, this field is 3 bit long. For Peripheral (IP), this field is 2 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

5.1.7.22 Pre Divider Register (CCM_PRE*n*)

Pre Register

Address: 3038_0000h base + 8030h offset + (128d × i), where i=0d to 141d





CCM_PRE*n* field descriptions

Field	Description
31 BUSY4	EN_A field is applied to field This field applies to DRAM and DRAM_PHYM
30–29 -	This field is reserved. Reserved
28 EN_A	Branch A clock gate control This field applies to DRAM and DRAM_PHYM 0 Clock shutdown 1 clock ON
27 -	This field is reserved. Reserved
26–24 MUX_A	Selection control of multiplexer of branch A This field applies to DRAM and DRAM_PHYM
23–20 -	This field is reserved. Reserved
19 BUSY3	Pre divider value for branch A is applied This field applies to DRAM and DRAM_PHYM

Table continues on the next page...

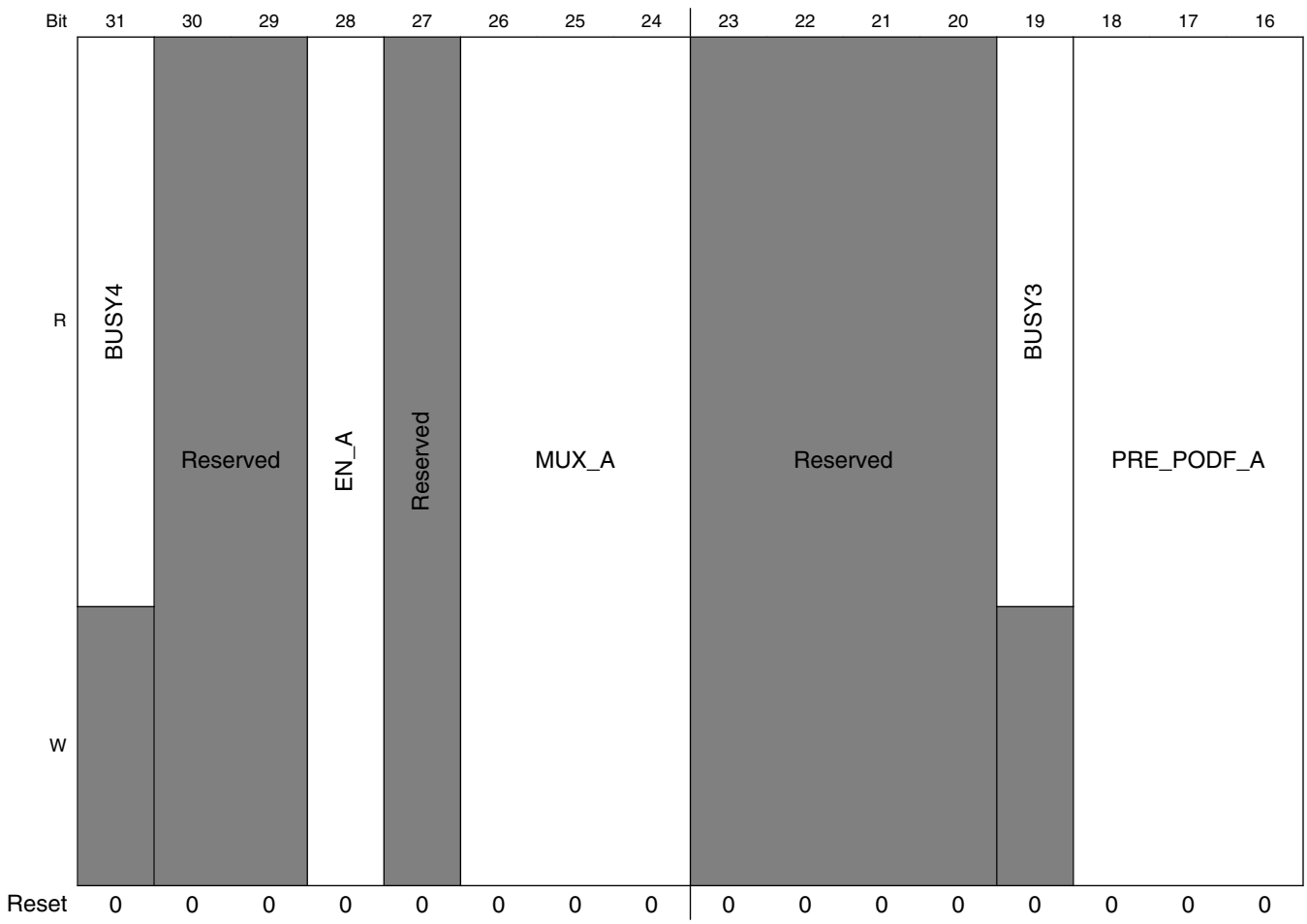
CCM_PRE*n* field descriptions (continued)

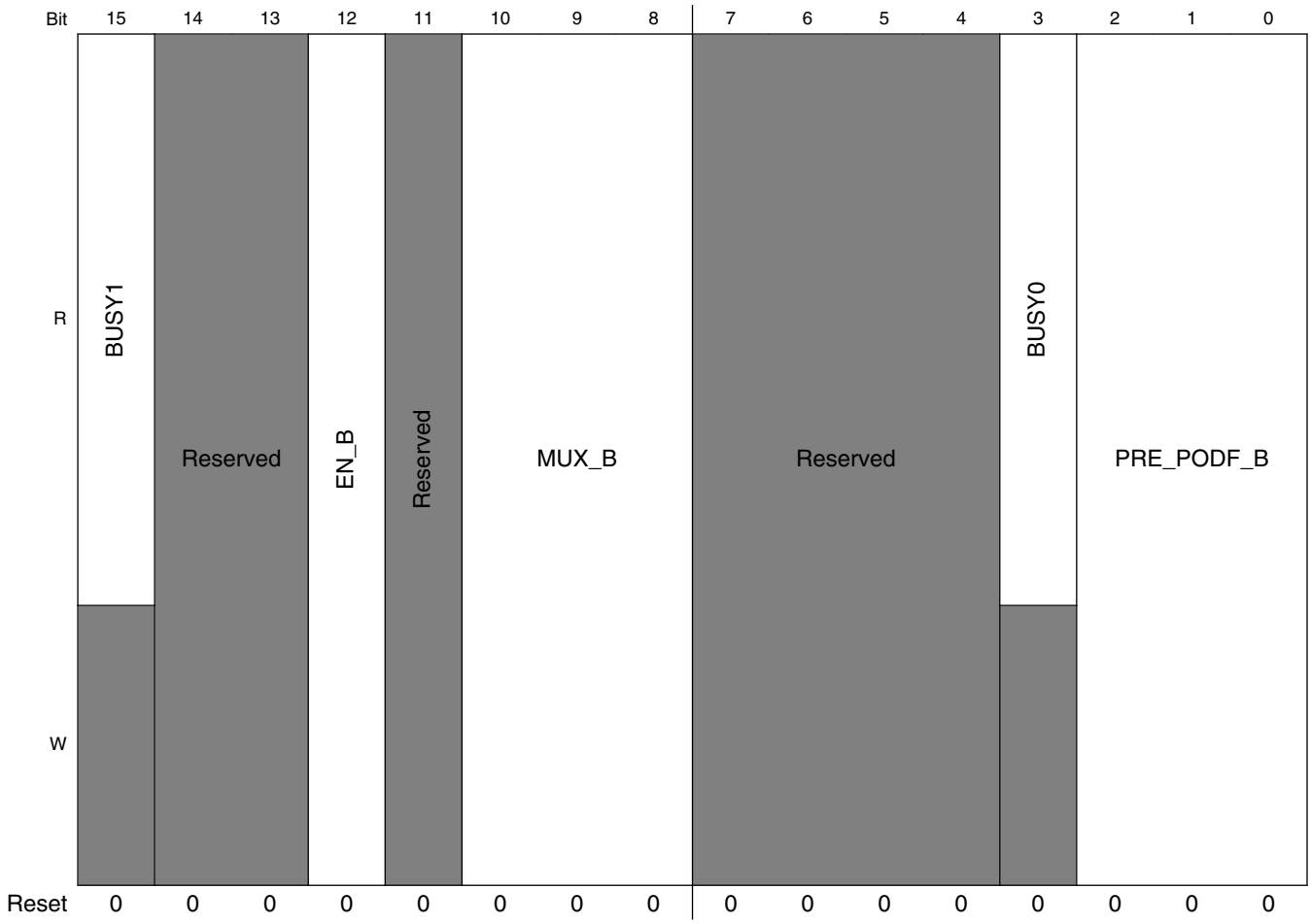
Field	Description
18–16 PRE_PODF_A	Pre divider divide number for branch A Divider value is n + 1. This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15 BUSY1	EN_B is applied to field This field does not apply for CORE, IP,DRAM, DRAM_PHYM
14–13 -	This field is reserved. Reserved
12 EN_B	Branch B clock gate control This field does not apply for CORE, IP,DRAM, DRAM_PHYM 0 Clock shutdown 1 Clock ON
11 -	This field is reserved. Reserved
10–8 MUX_B	Selection control of multiplexer of branch B This field does not apply for CORE, IP,DRAM, DRAM_PHYM
7–4 -	This field is reserved. Reserved
3 BUSY0	Pre divider value for branch a is applying field does not apply for CORE, IP,DRAM, DRAM_PHYM
PRE_PODF_B	Pre divider divide number for branch B Divider value is n + 1. This field does not apply for CORE, IP, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8

5.1.7.23 Pre Divider Register (CCM_PRE_ROOT*n*_SET)

Pre Divider Register

Address: 3038_0000h base + 8034h offset + (128d × i), where i=0d to 141d





CCM_PRE_ROOT*n*_SET field descriptions

Field	Description
31 BUSY4	EN_A field is applied to field This field applies to DRAM and DRAM_PHYM
30–29 -	This field is reserved. Reserved
28 EN_A	Branch A clock gate control This field applies to DRAM and DRAM_PHYM 0 Clock shutdown 1 clock ON
27 -	This field is reserved. Reserved
26–24 MUX_A	Selection control of multiplexer of branch A This field applies to DRAM and DRAM_PHYM
23–20 -	This field is reserved. Reserved
19 BUSY3	Pre divider value for branch A is applied This field applies to DRAM and DRAM_PHYM

Table continues on the next page...

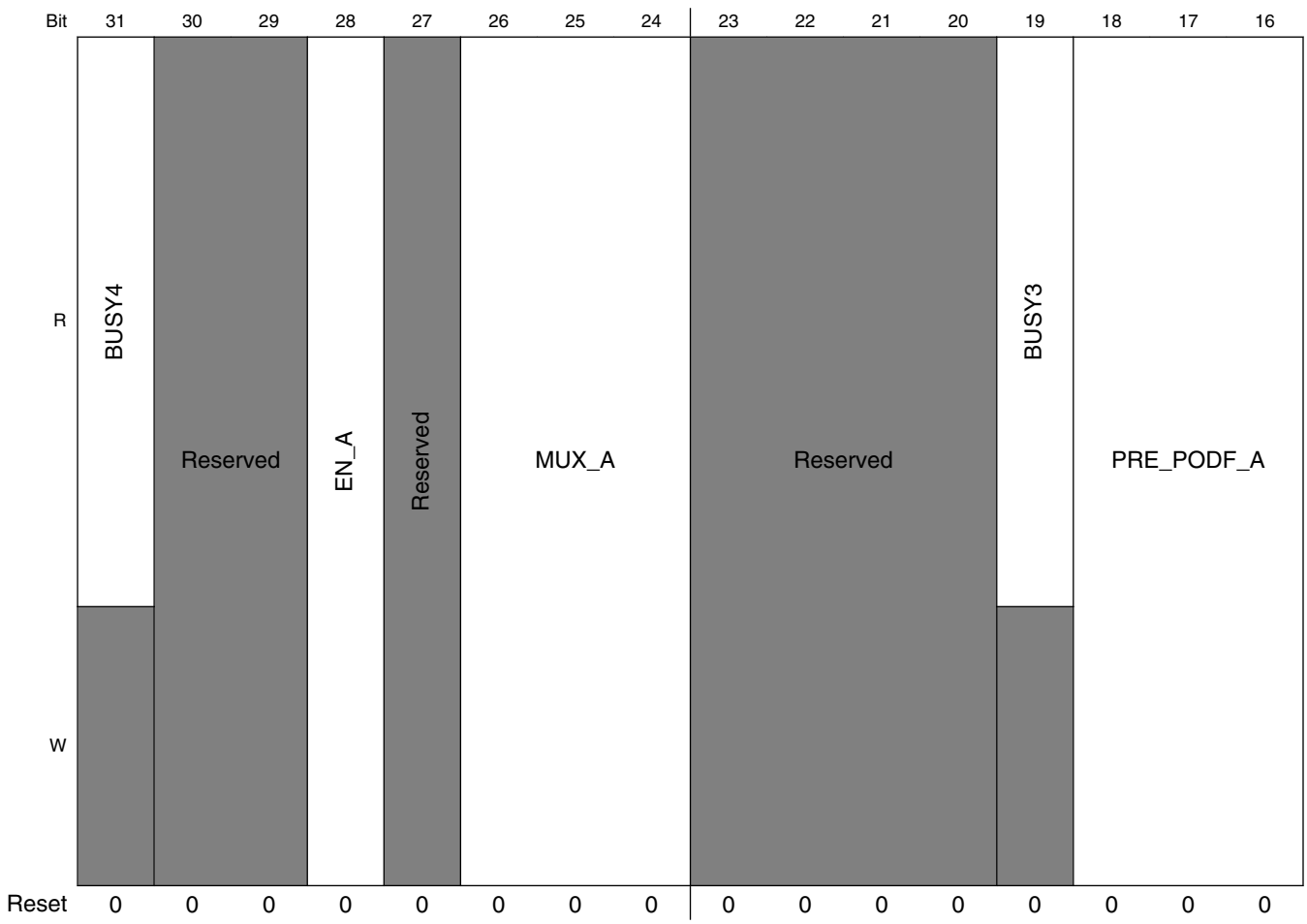
CCM_PRE_ROOT*n*_SET field descriptions (continued)

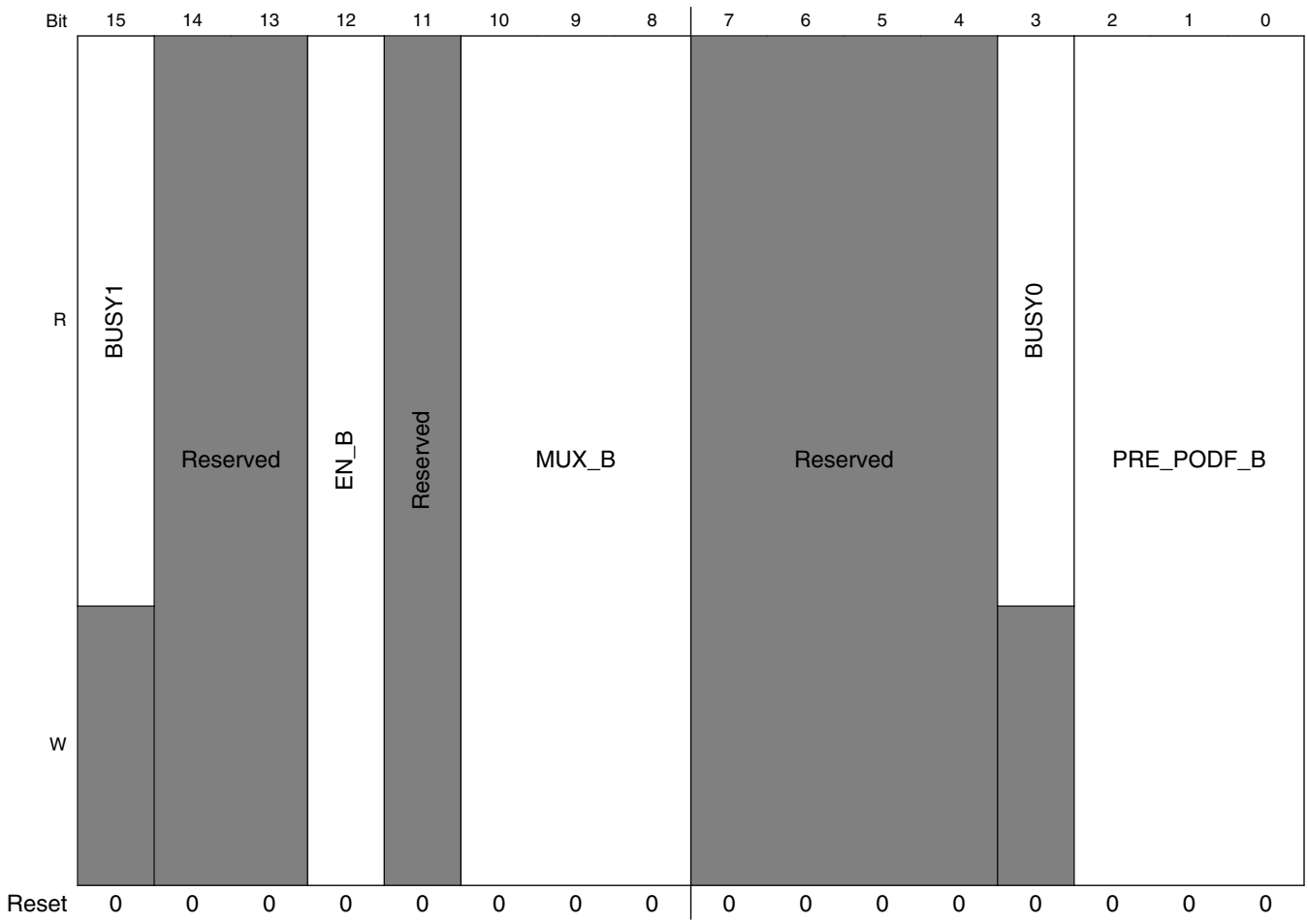
Field	Description
18–16 PRE_PODF_A	Pre divider divide number for branch A Divider value is n + 1. This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15 BUSY1	EN_B is applied to field This field does not apply for CORE, IP,DRAM, DRAM_PHYM
14–13 -	This field is reserved. Reserved
12 EN_B	Branch B clock gate control This field does not apply for CORE, IP,DRAM, DRAM_PHYM 0 Clock shutdown 1 Clock ON
11 -	This field is reserved. Reserved
10–8 MUX_B	Selection control of multiplexer of branch B This field does not apply for CORE, IP,DRAM, DRAM_PHYM
7–4 -	This field is reserved. Reserved
3 BUSY0	Pre divider value for branch A is applying field does not apply for CORE, IP,DRAM, DRAM_PHYM
PRE_PODF_B	Pre divider divide number for branch B Divider value is n + 1. This field does not apply for CORE, IP, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8

5.1.7.24 Pre Divider Register (CCM_PRE_ROOT*n*_CLR)

Pree Root Register

Address: 3038_0000h base + 8038h offset + (128d × i), where i=0d to 141d





CCM_PRE_ROOT*n*_CLR field descriptions

Field	Description
31 BUSY4	EN_A field is applied to field This field applies to DRAM and DRAM_PHYM
30–29 -	This field is reserved. Reserved
28 EN_A	Branch A clock gate control This field applies to DRAM and DRAM_PHYM 0 Clock shutdown 1 clock ON
27 -	This field is reserved. Reserved
26–24 MUX_A	Selection control of multiplexer of branch A This field applies to DRAM and DRAM_PHYM
23–20 -	This field is reserved. Reserved
19 BUSY3	Pre divider value for branch A is applied This field applies to DRAM and DRAM_PHYM

Table continues on the next page...

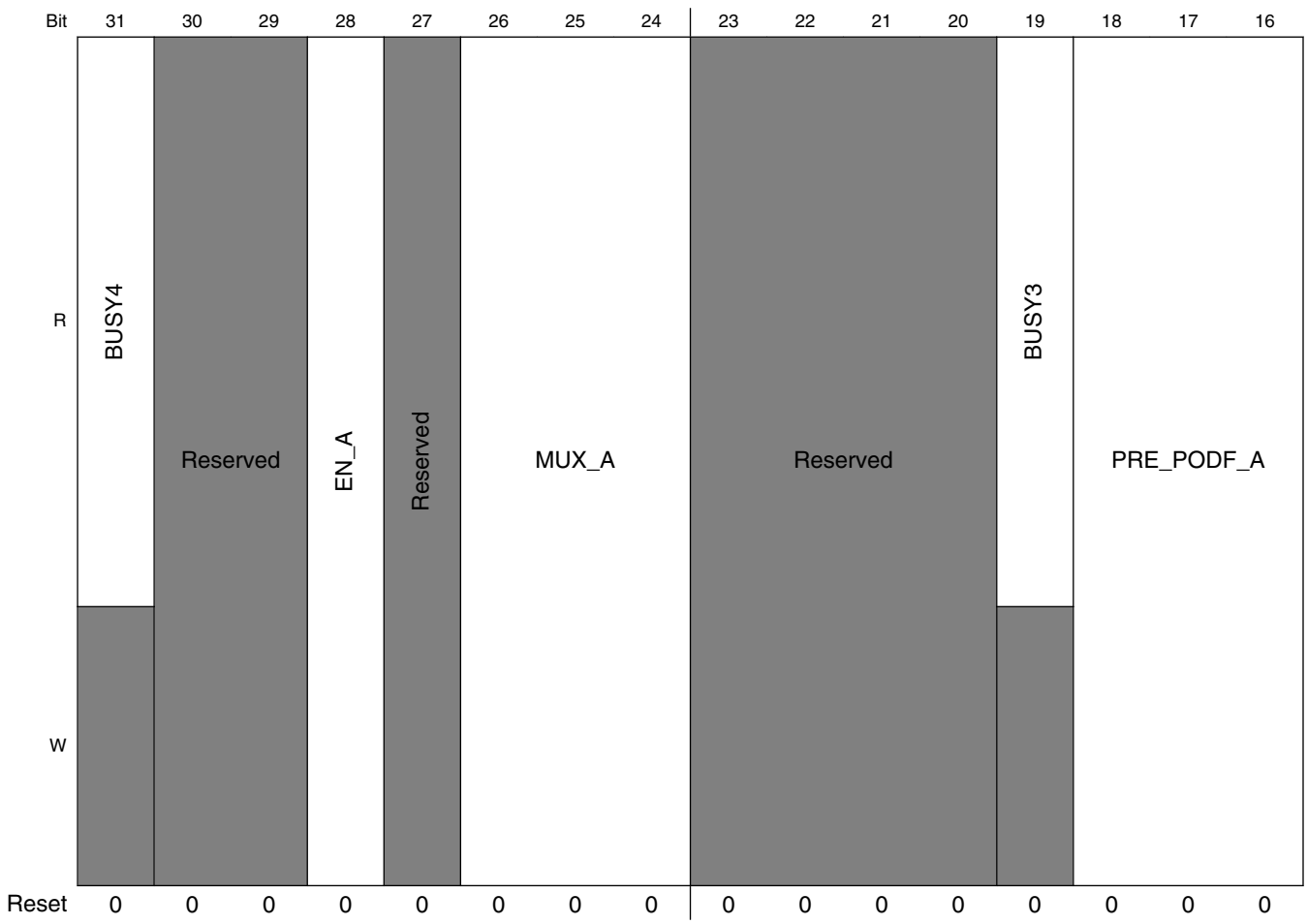
CCM_PRE_ROOT*n*_CLR field descriptions (continued)

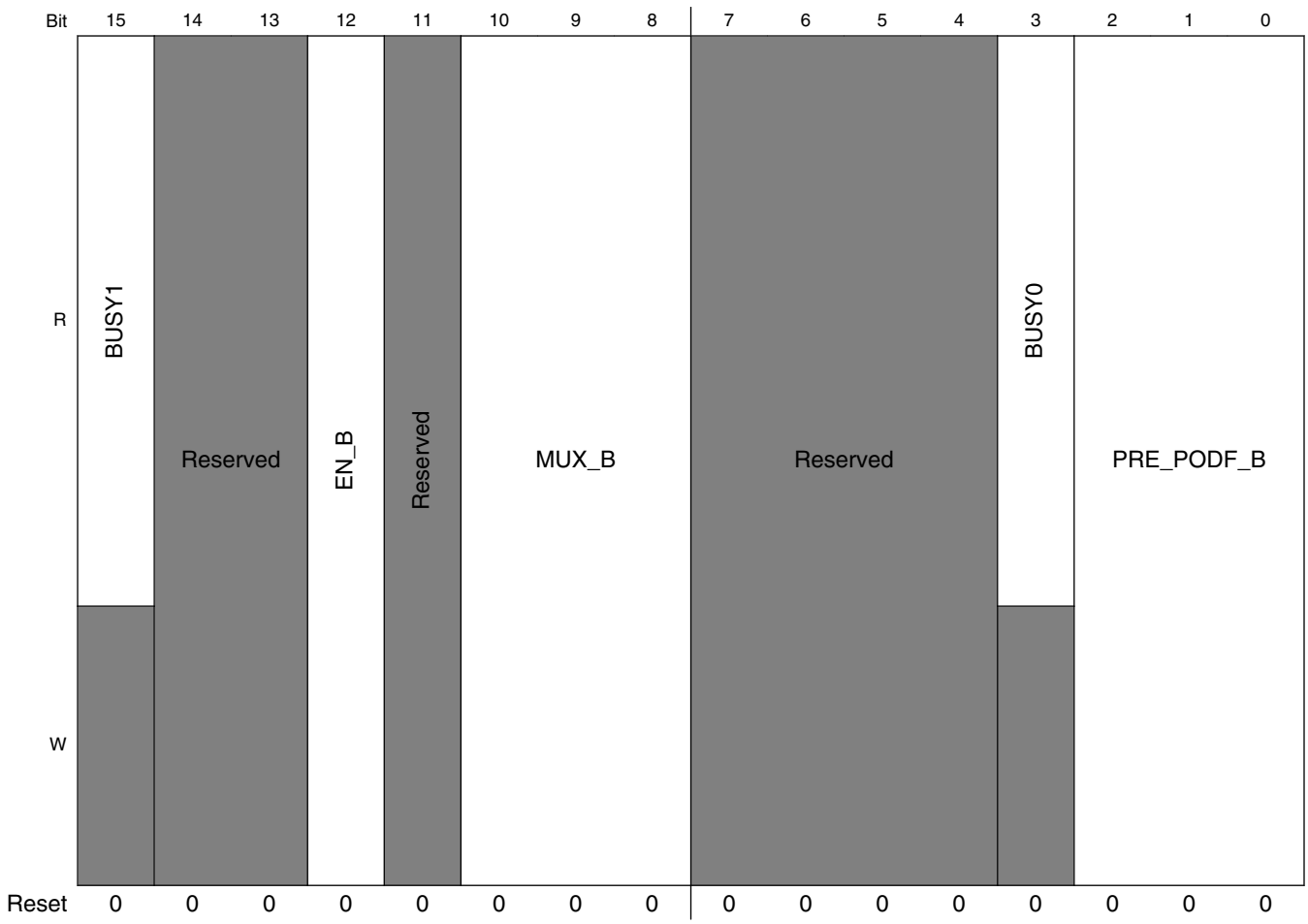
Field	Description
18–16 PRE_PODF_A	Pre divider divide number for branch A Divider value is n + 1. This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15 BUSY1	EN_B is applied to field This field does not apply for CORE, IP,DRAM, DRAM_PHYM
14–13 -	This field is reserved. Reserved
12 EN_B	Branch B clock gate control This field does not apply for CORE, IP,DRAM, DRAM_PHYM 0 Clock shutdown 1 Clock ON
11 -	This field is reserved. Reserved
10–8 MUX_B	Selection control of multiplexer of branch B This field does not apply for CORE, IP,DRAM, DRAM_PHYM
7–4 -	This field is reserved. Reserved
3 BUSY0	Pre divider value for branch A is applied This field does not apply for CORE, IP,DRAM, DRAM_PHYM
PRE_PODF_B	Pre divider divide number for branch B Divider value is n + 1. This field does not apply for CORE, IP, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8

5.1.7.25 Pre Divider Register (CCM_PRE_ROOT*n*_TOG)

Pre Root Register

Address: 3038_0000h base + 803Ch offset + (128d × i), where i=0d to 141d





CCM_PRE_ROOT*n*_TOG field descriptions

Field	Description
31 BUSY4	EN_A field is applied to field This field applies to DRAM and DRAM_PHYM
30–29 -	This field is reserved. Reserved
28 EN_A	Branch A clock gate control This field applies to DRAM and DRAM_PHYM 0 Clock shutdown 1 clock ON
27 -	This field is reserved. Reserved
26–24 MUX_A	Selection control of multiplexer of branch A This field applies to DRAM and DRAM_PHYM
23–20 -	This field is reserved. Reserved
19 BUSY3	Pre divider value for branch A is applied This field applies to DRAM and DRAM_PHYM

Table continues on the next page...

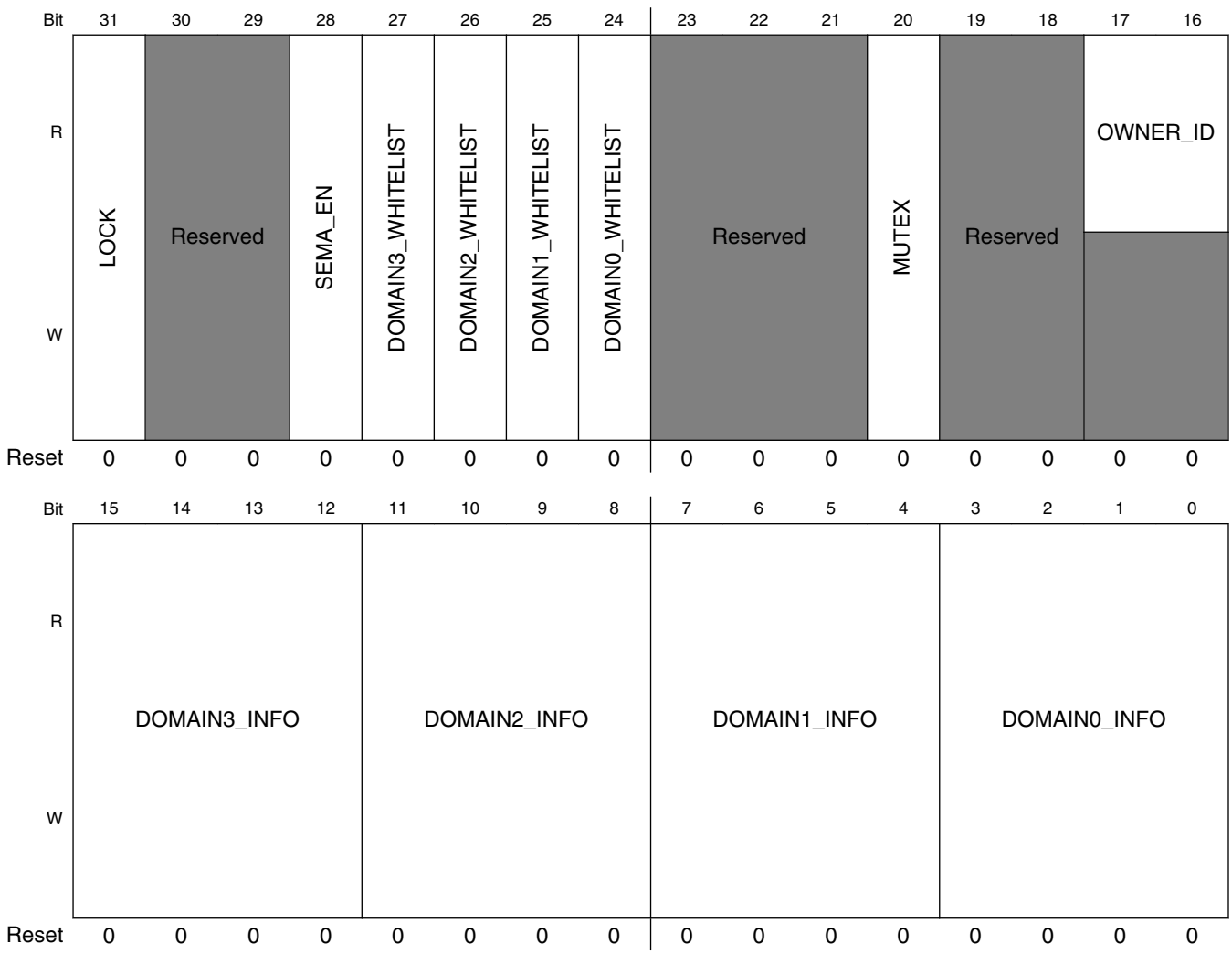
CCM_PRE_ROOT*n*_TOG field descriptions (continued)

Field	Description
18–16 PRE_PODF_A	Pre divider divide number for branch A Divider value is n + 1. This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15 BUSY1	EN_B is applied to field This field does not apply for CORE, IP,DRAM, DRAM_PHYM
14–13 -	This field is reserved. Reserved
12 EN_B	Branch B clock gate control This field does not apply for CORE, IP,DRAM, DRAM_PHYM 0 Clock shutdown 1 Clock ON
11 -	This field is reserved. Reserved
10–8 MUX_B	Selection control of multiplexer of branch B This field does not apply for CORE, IP,DRAM, DRAM_PHYM
7–4 -	This field is reserved. Reserved
3 BUSY0	Pre divider value for branch a is applied field does not apply for CORE, IP,DRAM, DRAM_PHYM
PRE_PODF_B	Pre divider divide number for branch B Divider value is n + 1. This field does not apply for CORE, IP, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8

5.1.7.26 Access Control Register (CCM_ACCESS_CTRL*n*)

Access Control Register

Address: 3038_0000h base + 8070h offset + (128d × i), where i=0d to 141d



CCM_ACCESS_CTRL*n* field descriptions

Field	Description
31 LOCK	Lock this clock root to use access control This bit can be set to 1 by software, and can be cleared only by system reset. 0 Access control inactive 1 Access control active
30–29 -	This field is reserved. Reserved

Table continues on the next page...

CCM_ACCESS_CTRL*n* field descriptions (continued)

Field	Description
28 SEMA_EN	Enable internal semaphore This field cannot be changed when lock bit is 1 0 Disable 1 Enable
27 DOMAIN3_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
26 DOMAIN2_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
25 DOMAIN1_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
24 DOMAIN0_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
23–21 -	This field is reserved. Reserved
20 MUTEX	Semaphore to control access 0 Semaphore is free to take 1 Semaphore is taken Write 0 Release semaphore Write 1 Acquire semaphore
19–18 -	This field is reserved. Reserved
17–16 OWNER_ID	Current domain that owns semaphore This field is meaningless when MUTEX is 0 0 domaino 1 domain1 2 domain2 3 domain3
15–12 DOMAIN3_INFO	Information from domain 3 to pass to others This field can only be changed by domain 3
11–8 DOMAIN2_INFO	Information from domain 2 to pass to others This field can only be changed by domain 2

Table continues on the next page...

CCM_ACCESS_CTRL*n* field descriptions (continued)

Field	Description
7–4 DOMAIN1_INFO	Information from domain 1 to pass to others This field can only be changed by domain 1
DOMAIN0_INFO	Information from domain 0 to pass to others This field can only be changed by domain 0

5.1.7.27 Access Control Register (CCM_ACCESS_CTRL_ROOT*n*_SET)

Access Control Register

Address: 3038_0000h base + 8074h offset + (128d × i), where i=0d to 141d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	Reserved		SEMA_EN	DOMAIN3_WHITELIST	DOMAIN2_WHITELIST	DOMAIN1_WHITELIST	DOMAIN0_WHITELIST	Reserved			MUTEX	Reserved		OWNER_ID	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DOMAIN3_INFO				DOMAIN2_INFO				DOMAIN1_INFO				DOMAIN0_INFO			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_ACCESS_CTRL_ROOT*n*_SET field descriptions

Field	Description
31 LOCK	Lock this clock root to use access control This bit can be set to 1 by software, and can be cleared only by system reset. 0 Access control inactive 1 Access control active
30–29 -	This field is reserved. Reserved
28 SEMA_EN	Enable internal semaphore This field cannot be changed when lock bit is 1 0 Disable 1 Enable
27 DOMAIN3_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
26 DOMAIN2_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
25 DOMAIN1_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
24 DOMAIN0_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
23–21 -	This field is reserved. Reserved
20 MUTEX	Semaphore to control access 0 Semaphore is free to take 1 Semaphore is taken Write 0 Release semaphore Write 1 Acquire semaphore
19–18 -	This field is reserved. Reserved
17–16 OWNER_ID	Current domain that owns semaphore This field is meaningless when MUTEX is 0 0 domaino 1 domain1

Table continues on the next page...

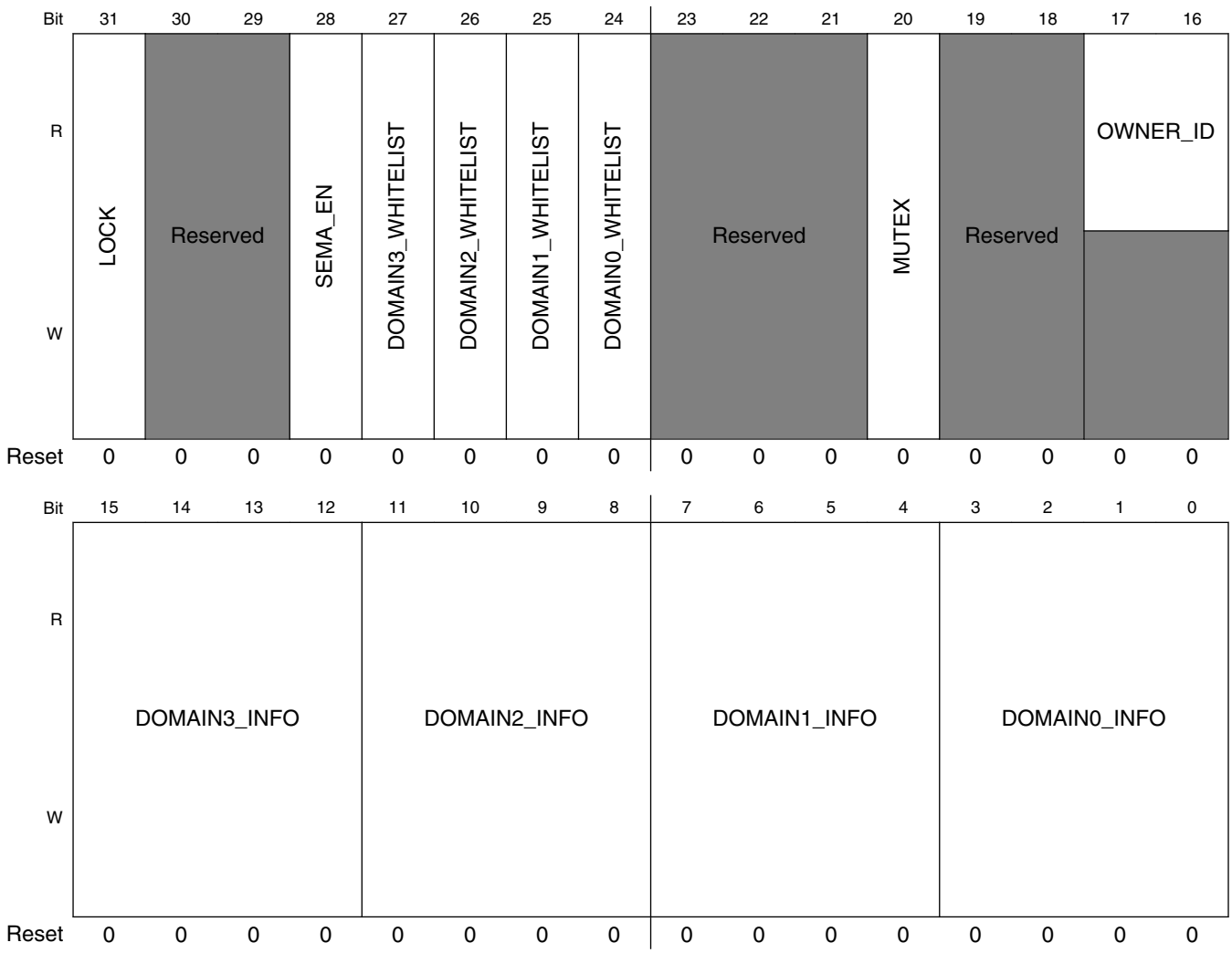
CCM_ACCESS_CTRL_ROOT*n*_SET field descriptions (continued)

Field	Description
	2 domain2 3 domain3
15–12 DOMAIN3_INFO	Information from domain 3 to pass to others This field can only be changed by domain 3
11–8 DOMAIN2_INFO	Information from domain 2 to pass to others This field can only be changed by domain 2
7–4 DOMAIN1_INFO	Information from domain 1 to pass to others This field can only be changed by domain 1
DOMAIN0_INFO	Information from domain 0 to pass to others This field can only be changed by domain 0

5.1.7.28 Access Control Register (CCM_ACCESS_CTRL_ROOT*n*_CLR)

Access Control Register

Address: 3038_0000h base + 8078h offset + (128d × i), where i=0d to 141d



CCM_ACCESS_CTRL_ROOT*n*_CLR field descriptions

Field	Description
31 LOCK	Lock this clock root to use access control This bit can be set to 1 by software, and can be cleared only by system reset. 0 Access control inactive 1 Access control active

Table continues on the next page...

CCM_ACCESS_CTRL_ROOT*n*_CLR field descriptions (continued)

Field	Description
30–29 -	This field is reserved. Reserved
28 SEMA_EN	Enable internal semaphore This field cannot be changed when lock bit is 1 0 Disable 1 Enable
27 DOMAIN3_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
26 DOMAIN2_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
25 DOMAIN1_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
24 DOMAIN0_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
23–21 -	This field is reserved. Reserved
20 MUTEX	Semaphore to control access 0 Semaphore is free to take 1 Semaphore is taken Write 0 Release semaphore Write 1 Acquire semaphore
19–18 -	This field is reserved. Reserved
17–16 OWNER_ID	Current domain that owns semaphore This field is meaningless when MUTEX is 0 0 domaino 1 domain1 2 domain2 3 domain3
15–12 DOMAIN3_INFO	Information from domain 3 to pass to others This field can only be changed by domain 3

Table continues on the next page...

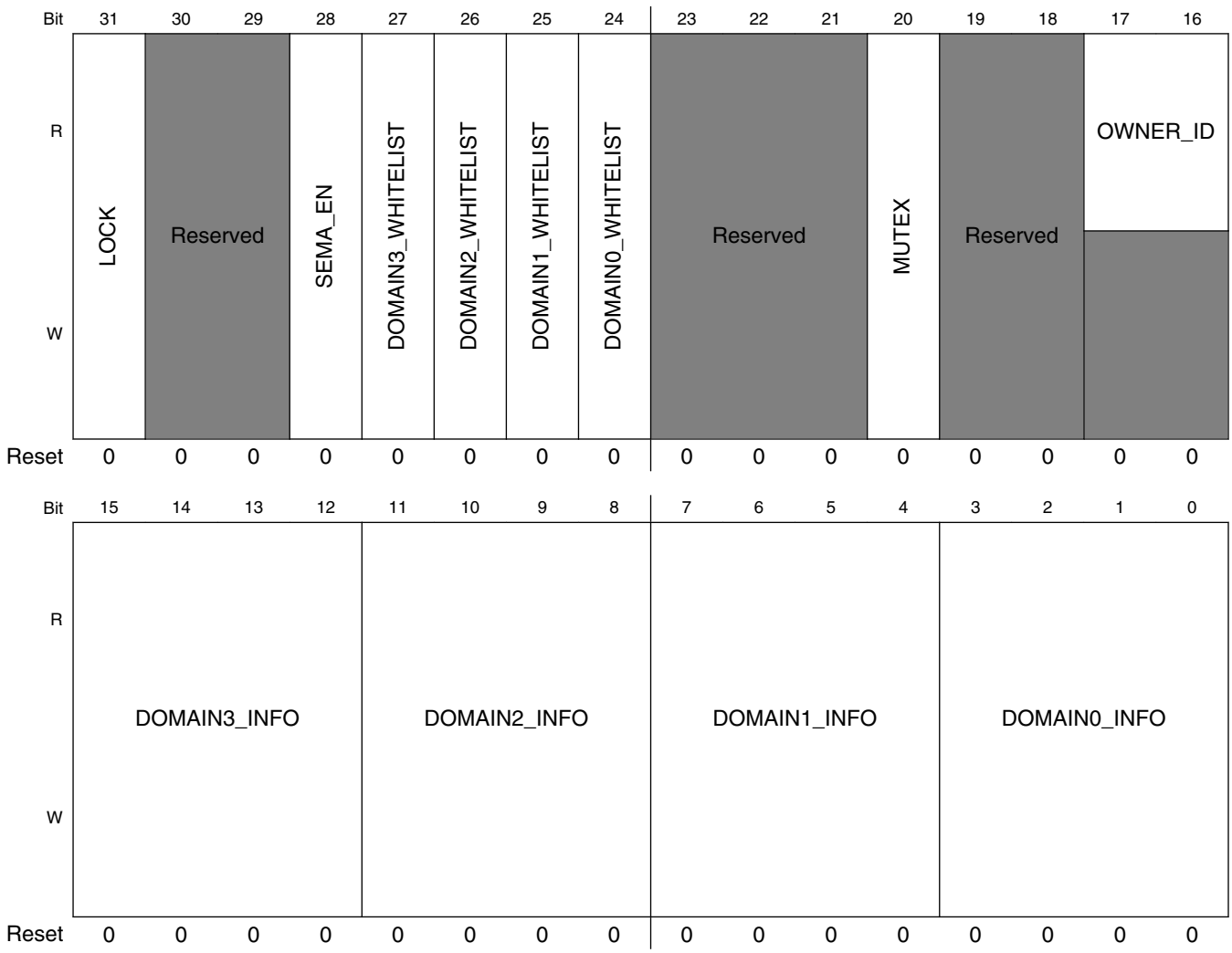
CCM_ACCESS_CTRL_ROOT*n*_CLR field descriptions (continued)

Field	Description
11–8 DOMAIN2_INFO	Information from domain 2 to pass to others This field can only be changed by domain 2
7–4 DOMAIN1_INFO	Information from domain 1 to pass to others This field can only be changed by domain 1
DOMAIN0_INFO	Information from domain 0 to pass to others This field can only be changed by domain 0

5.1.7.29 Access Control Register (CCM_ACCESS_CTRL_ROOT*n*_TOG)

Access Control Register

Address: 3038_0000h base + 807Ch offset + (128d × i), where i=0d to 141d



CCM_ACCESS_CTRL_ROOT*n*_TOG field descriptions

Field	Description
31 LOCK	Lock this clock root to use access control This bit can be set to 1 by software, and can be cleared only by system reset. 0 Access control inactive 1 Access control active

Table continues on the next page...

CCM_ACCESS_CTRL_ROOT*n*_TOG field descriptions (continued)

Field	Description
30–29 -	This field is reserved. Reserved
28 SEMA_EN	Enable internal semaphore This field cannot be changed when lock bit is 1 0 Disable 1 Enable
27 DOMAIN3_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
26 DOMAIN2_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
25 DOMAIN1_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
24 DOMAIN0_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
23–21 -	This field is reserved. Reserved
20 MUTEX	Semaphore to control access 0 Semaphore is free to take 1 Semaphore is taken Write 0 Release semaphore Write 1 Acquire semaphore
19–18 -	This field is reserved. Reserved
17–16 OWNER_ID	Current domain that owns semaphore This field is meaningless when MUTEX is 0 0 domaino 1 domain1 2 domain2 3 domain3
15–12 DOMAIN3_INFO	Information from domain 3 to pass to others This field can only be changed by domain 3

Table continues on the next page...

CCM_ACCESS_CTRL_ROOT*n*_TOG field descriptions (continued)

Field	Description
11–8 DOMAIN2_INFO	Information from domain 2 to pass to others This field can only be changed by domain 2
7–4 DOMAIN1_INFO	Information from domain 1 to pass to others This field can only be changed by domain 1
DOMAIN0_INFO	Information from domain 0 to pass to others This field can only be changed by domain 0

5.1.8 CCM Analog Memory Map/Register Definition

CCM_ANALOG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3036_0000	AUDIO PLL1 General Function Control Register (CCM_ANALOG_AUDIO_PLL1_GEN_CTRL)	32	R/W	0000_2010h	5.1.8.1/510
3036_0004	AUDIO PLL1 Divide and Fraction Data Control 0 Register (CCM_ANALOG_AUDIO_PLL1_FDIV_CTL0)	32	R/W	0014_5032h	5.1.8.2/512
3036_0008	AUDIO PLL1 Divide and Fraction Data Control 1 Register (CCM_ANALOG_AUDIO_PLL1_FDIV_CTL1)	32	R/W	0000_0000h	5.1.8.3/513
3036_000C	AUDIO PLL1 PLL SSCG Control Register (CCM_ANALOG_AUDIO_PLL1_SSCG_CTRL)	32	R/W	0000_0000h	5.1.8.4/513
3036_0010	AUDIO PLL1 PLL Monitoring Control Register (CCM_ANALOG_AUDIO_PLL1_MNIT_CTRL)	32	R/W	0010_0103h	5.1.8.5/515
3036_0014	AUDIO PLL2 General Function Control Register (CCM_ANALOG_AUDIO_PLL2_GEN_CTRL)	32	R/W	0000_2010h	5.1.8.6/517
3036_0018	AUDIO PLL2 Divide and Fraction Data Control 0 Register (CCM_ANALOG_AUDIO_PLL2_FDIV_CTL0)	32	R/W	0014_5032h	5.1.8.7/519
3036_001C	AUDIO PLL2 Divide and Fraction Data Control 1 Register (CCM_ANALOG_AUDIO_PLL2_FDIV_CTL1)	32	R/W	0000_0000h	5.1.8.8/520
3036_0020	AUDIO PLL2 PLL SSCG Control Register (CCM_ANALOG_AUDIO_PLL2_SSCG_CTRL)	32	R/W	0000_0000h	5.1.8.9/520
3036_0024	AUDIO PLL2 PLL Monitoring Control Register (CCM_ANALOG_AUDIO_PLL2_MNIT_CTRL)	32	R/W	0010_0103h	5.1.8.10/522
3036_0028	VIDEO PLL1 General Function Control Register (CCM_ANALOG_VIDEO_PLL1_GEN_CTRL)	32	R/W	0000_2010h	5.1.8.11/524
3036_002C	VIDEO PLL1 Divide and Fraction Data Control 0 Register (CCM_ANALOG_VIDEO_PLL1_FDIV_CTL0)	32	R/W	0014_5032h	5.1.8.12/526
3036_0030	VIDEO PLL1 Divide and Fraction Data Control 1 Register (CCM_ANALOG_VIDEO_PLL1_FDIV_CTL1)	32	R/W	0000_0000h	5.1.8.13/527
3036_0034	VIDEO PLL1 PLL SSCG Control Register (CCM_ANALOG_VIDEO_PLL1_SSCG_CTRL)	32	R/W	0000_0000h	5.1.8.14/527

Table continues on the next page...

CCM_ANALOG memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3036_0038	VIDEO PLL1 PLL Monitoring Control Register (CCM_ANALOG_VIDEO_PLL1_MNIT_CTRL)	32	R/W	0010_0103h	5.1.8.15/529
3036_0050	DRAM PLL General Function Control Register (CCM_ANALOG_DRAM_PLL_GEN_CTRL)	32	R/W	0000_2010h	5.1.8.16/531
3036_0054	DRAM PLL Divide and Fraction Data Control 0 Register (CCM_ANALOG_DRAM_PLL_FDIV_CTL0)	32	R/W	0012_C032h	5.1.8.17/533
3036_0058	DRAM PLL Divide and Fraction Data Control 1 Register (CCM_ANALOG_DRAM_PLL_FDIV_CTL1)	32	R/W	0000_0000h	5.1.8.18/534
3036_005C	DRAM PLL PLL SSCG Control Register (CCM_ANALOG_DRAM_PLL_SSCG_CTRL)	32	R/W	0000_0000h	5.1.8.19/534
3036_0060	DRAM PLL PLL Monitoring Control Register (CCM_ANALOG_DRAM_PLL_MNIT_CTRL)	32	R/W	0010_0103h	5.1.8.20/536
3036_0064	GPU PLL General Function Control Register (CCM_ANALOG_GPU_PLL_GEN_CTRL)	32	R/W	0000_0810h	5.1.8.21/538
3036_0068	GPU PLL Divide and Fraction Data Control 0 Register (CCM_ANALOG_GPU_PLL_FDIV_CTL0)	32	R/W	000C_8031h	5.1.8.22/540
3036_006C	PLL Lock Detector Control Register (CCM_ANALOG_GPU_PLL_LOCKD_CTRL)	32	R/W	0010_003Fh	5.1.8.23/541
3036_0070	PLL Monitoring Control Register (CCM_ANALOG_GPU_PLL_MNIT_CTRL)	32	R/W	0028_0081h	5.1.8.24/542
3036_0074	VPU PLL General Function Control Register (CCM_ANALOG_VPU_PLL_GEN_CTRL)	32	R/W	0000_0810h	5.1.8.25/544
3036_0078	VPU PLL Divide and Fraction Data Control 0 Register (CCM_ANALOG_VPU_PLL_FDIV_CTL0)	32	R/W	0012_C032h	5.1.8.26/546
3036_007C	PLL Lock Detector Control Register (CCM_ANALOG_VPU_PLL_LOCKD_CTRL)	32	R/W	0010_003Fh	5.1.8.23/541
3036_0080	PLL Monitoring Control Register (CCM_ANALOG_VPU_PLL_MNIT_CTRL)	32	R/W	0028_0081h	5.1.8.24/542
3036_0084	ARM PLL General Function Control Register (CCM_ANALOG_ARM_PLL_GEN_CTRL)	32	R/W	0000_0810h	5.1.8.27/548
3036_0088	ARM PLL Divide and Fraction Data Control 0 Register (CCM_ANALOG_ARM_PLL_FDIV_CTL0)	32	R/W	000F_A031h	5.1.8.28/550
3036_008C	PLL Lock Detector Control Register (CCM_ANALOG_ARM_PLL_LOCKD_CTRL)	32	R/W	0010_003Fh	5.1.8.23/541
3036_0090	PLL Monitoring Control Register (CCM_ANALOG_ARM_PLL_MNIT_CTRL)	32	R/W	0028_0081h	5.1.8.24/542
3036_0094	SYS PLL1 General Function Control Register (CCM_ANALOG_SYS_PLL1_GEN_CTRL)	32	R/W	0AAA_A810h	5.1.8.29/552
3036_0098	SYS PLL1 Divide and Fraction Data Control 0 Register (CCM_ANALOG_SYS_PLL1_FDIV_CTL0)	32	R/W	0019_0032h	5.1.8.30/555
3036_009C	PLL Lock Detector Control Register (CCM_ANALOG_SYS_PLL1_LOCKD_CTRL)	32	R/W	0010_003Fh	5.1.8.23/541
3036_0100	PLL Monitoring Control Register (CCM_ANALOG_SYS_PLL1_MNIT_CTRL)	32	R/W	0028_0081h	5.1.8.31/556

Table continues on the next page...

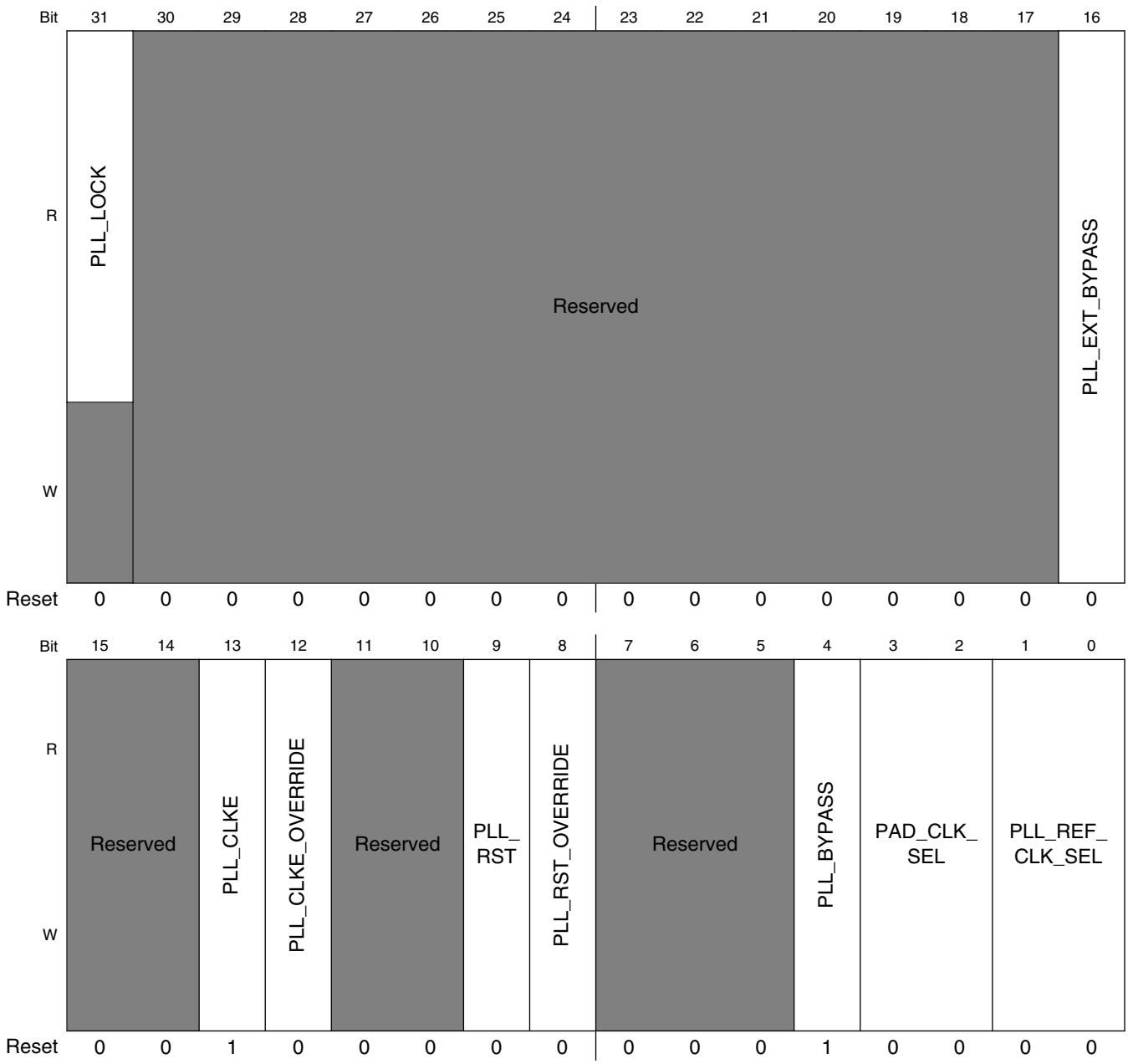
CCM_ANALOG memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3036_0104	SYS PLL2 General Function Control Register (CCM_ANALOG_SYS_PLL2_GEN_CTRL)	32	R/W	0AAA_A810h	5.1.8.32/558
3036_0108	SYS PLL2 Divide and Fraction Data Control 0 Register (CCM_ANALOG_SYS_PLL2_FDIV_CTL0)	32	R/W	000F_A031h	5.1.8.33/561
3036_010C	PLL Lock Detector Control Register (CCM_ANALOG_SYS_PLL2_LOCKD_CTRL)	32	R/W	0010_003Fh	5.1.8.34/562
3036_0110	PLL Monitoring Control Register (CCM_ANALOG_SYS_PLL2_MNIT_CTRL)	32	R/W	0028_0081h	5.1.8.31/556
3036_0114	SYS PLL3 General Function Control Register (CCM_ANALOG_SYS_PLL3_GEN_CTRL)	32	R/W	0000_0810h	5.1.8.35/564
3036_0118	SYS PLL3 Divide and Fraction Data Control 0 Register (CCM_ANALOG_SYS_PLL3_FDIV_CTL0)	32	R/W	000F_A031h	5.1.8.36/566
3036_011C	PLL Lock Detector Control Register (CCM_ANALOG_SYS_PLL3_LOCKD_CTRL)	32	R/W	0010_003Fh	5.1.8.34/562
3036_0120	PLL Monitoring Control Register (CCM_ANALOG_SYS_PLL3_MNIT_CTRL)	32	R/W	0028_0081h	5.1.8.31/556
3036_0124	Osc Misc Configuration Register (CCM_ANALOG_OSC_MISC_CFG)	32	R/W	0000_0000h	5.1.8.37/567
3036_0128	PLL Clock Output for Test Enable and Select Register (CCM_ANALOG_ANAMIX_PLL_MNIT_CTL)	32	R/W	0000_0000h	5.1.8.38/568
3036_0800	DIGPROG Register (CCM_ANALOG_DIGPROG)	32	R	0082_4010h	5.1.8.39/570

5.1.8.1 AUDIO PLL1 General Function Control Register (CCM_ANALOG_AUDIO_PLL1_GEN_CTRL)

AUDIO PLL1 General Function Control Register

Address: 3036_0000h base + 0h offset = 3036_0000h



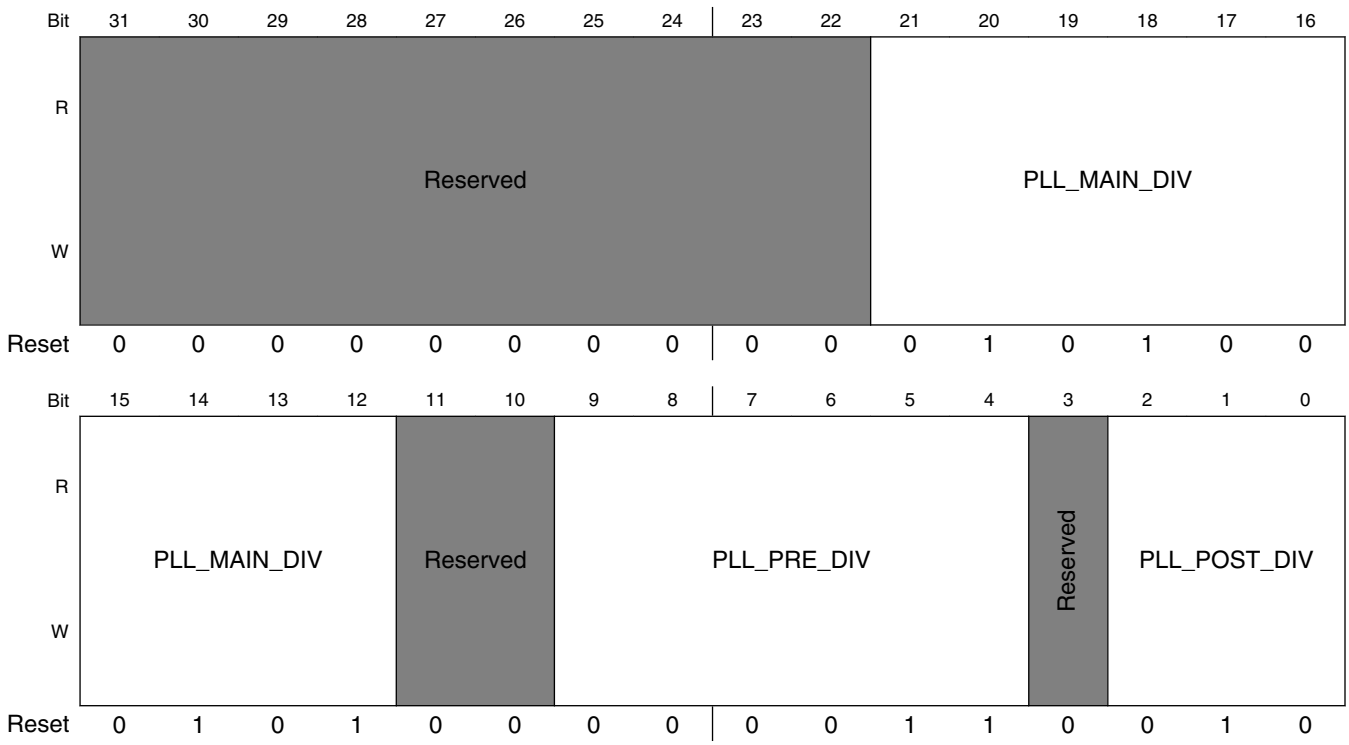
CCM_ANALOG_AUDIO_PLL1_GEN_CTRL field descriptions

Field	Description
31 PLL_LOCK	PLL lock signal
30–17 -	This field is reserved. Reserved
16 PLL_EXT_ BYPASS	PLL analog block bypass, clock output traces to PLL source
15–14 -	This field is reserved. Reserved
13 PLL_CLKE	PLL output clock clock gating enable
12 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM
11–10 -	This field is reserved. Reserved
9 PLL_RST	PLL reset (active low)
8 PLL_RST_ OVERRIDE	PLL reset overrided by CCM
7–5 -	This field is reserved. Reserved
4 PLL_BYPASS	PLL output clock bypass
3–2 PAD_CLK_SEL	PAD clock select PAD_CLK is an alternate input reference clock for the PLL. The clock source selection for PAD_CLK is defined below. 00 CLKIN1 XOR CLKIN2 01 CLKIN2 10 CLKIN1 11 Reserved
PLL_REF_CLK_ SEL	PLL reference clock select 00 24M_REF_CLK 01 PAD_CLK 10 Reserved 11 Reserved

5.1.8.2 AUDIO PLL1 Divide and Fraction Data Control 0 Register (CCM_ANALOG_AUDIO_PLL1_FDIV_CTL0)

AUDIO PLL1 Divide and Fraction Data Control 0 Register

Address: 3036_0000h base + 4h offset = 3036_0004h



CCM_ANALOG_AUDIO_PLL1_FDIV_CTL0 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–12 PLL_MAIN_DIV	Value of the main-divider
11–10 -	This field is reserved. Reserved
9–4 PLL_PRE_DIV	Value of the pre-divider
3 -	This field is reserved. Reserved
PLL_POST_DIV	Value of the post-divider

5.1.8.3 AUDIO PLL1 Divide and Fraction Data Control 1 Register (CCM_ANALOG_AUDIO_PLL1_FDIV_CTL1)

AUDIO PLL1 Divide and Fraction Data Control 1 Register

Address: 3036_0000h base + 8h offset = 3036_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																PLL_DSM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_ANALOG_AUDIO_PLL1_FDIV_CTL1 field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
PLL_DSM	Value of the DSM

5.1.8.4 AUDIO PLL1 PLL SSCG Control Register (CCM_ANALOG_AUDIO_PLL1_SSCG_CTRL)

AUDIO PLL1 PLL SSCG Control Register

Address: 3036_0000h base + Ch offset = 3036_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SSCG_EN	Reserved											PLL_MFREQ_CTL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_MFREQ_CTL				Reserved		PLL_MRAT_CTL						Reserved		SEL_PF	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

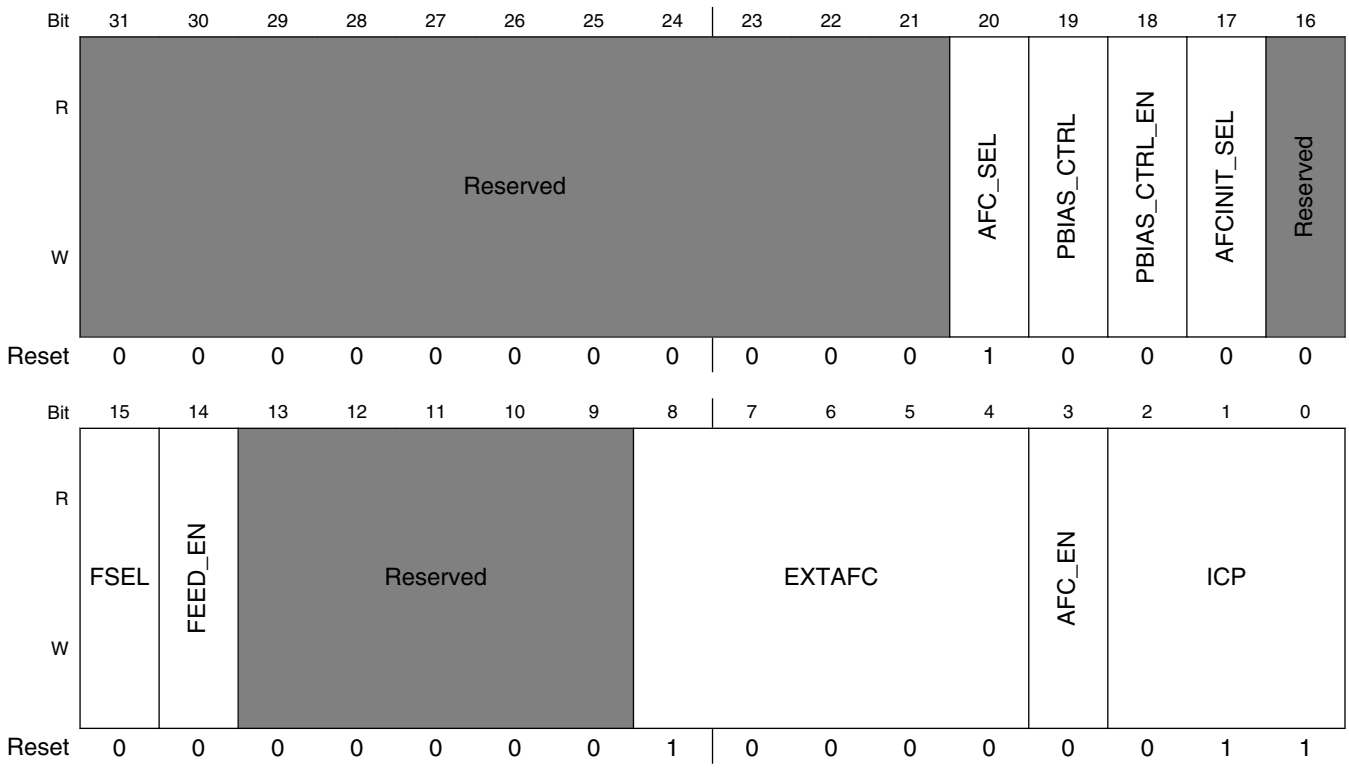
CCM_ANALOG_AUDIO_PLL1_SSCG_CTRL field descriptions

Field	Description
31 SSCG_EN	SSCG Enable 1 Enable Spread Spectrum Mode 0 Disable Spread Spectrum Mode
30–20 -	This field is reserved. Reserved
19–12 PLL_MFREQ_CTL	Value of modulation frequency control Modulation Frequency, MF is determined by the following equation: MF = F_{FIN}/p/mfr/(2^5) Hz F_{FIN} is the PLL input clock frequency, mfr is the decimal value for PLL_MFREQ_CTL[7:0], and p is the decimal value for PLL_PRE_DIV.
11–10 -	This field is reserved. Reserved
9–4 PLL_MRAT_CTL	Value of modulation rate control Modulation rate (pk-pk), MR, is determined by the following equation: MR = mfr x mrr /m /(2^6) x 100 [%] mfr is the decimal value of PLL_MFREQ_CTL, mrr is the decimal value for PLL_MRAT_CTL[5:0], and m is the decimal of PLL_MAIN_DIV.
3–2 -	This field is reserved. Reserved
SEL_PF	Value of modulation method control 00 Down spread 01 Up spread 1x Center spread

5.1.8.5 AUDIO PLL1 PLL Monitoring Control Register (CCM_ANALOG_AUDIO_PLL1_MNIT_CTRL)

AUDIO PLL1 PLL Monitoring Control Register

Address: 3036_0000h base + 10h offset = 3036_0010h



CCM_ANALOG_AUDIO_PLL1_MNIT_CTRL field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 AFC_SEL	AFC Mode select
19 PBIAS_CTRL	PBIAS pull-down initial voltage control pin 0 0.50*VDD 1 0.67*VDD
18 PBIAS_CTRL_EN	PBIAS voltage pull-down enable pin
17 AFCINIT_SEL	AFC initial delay select pin

Table continues on the next page...

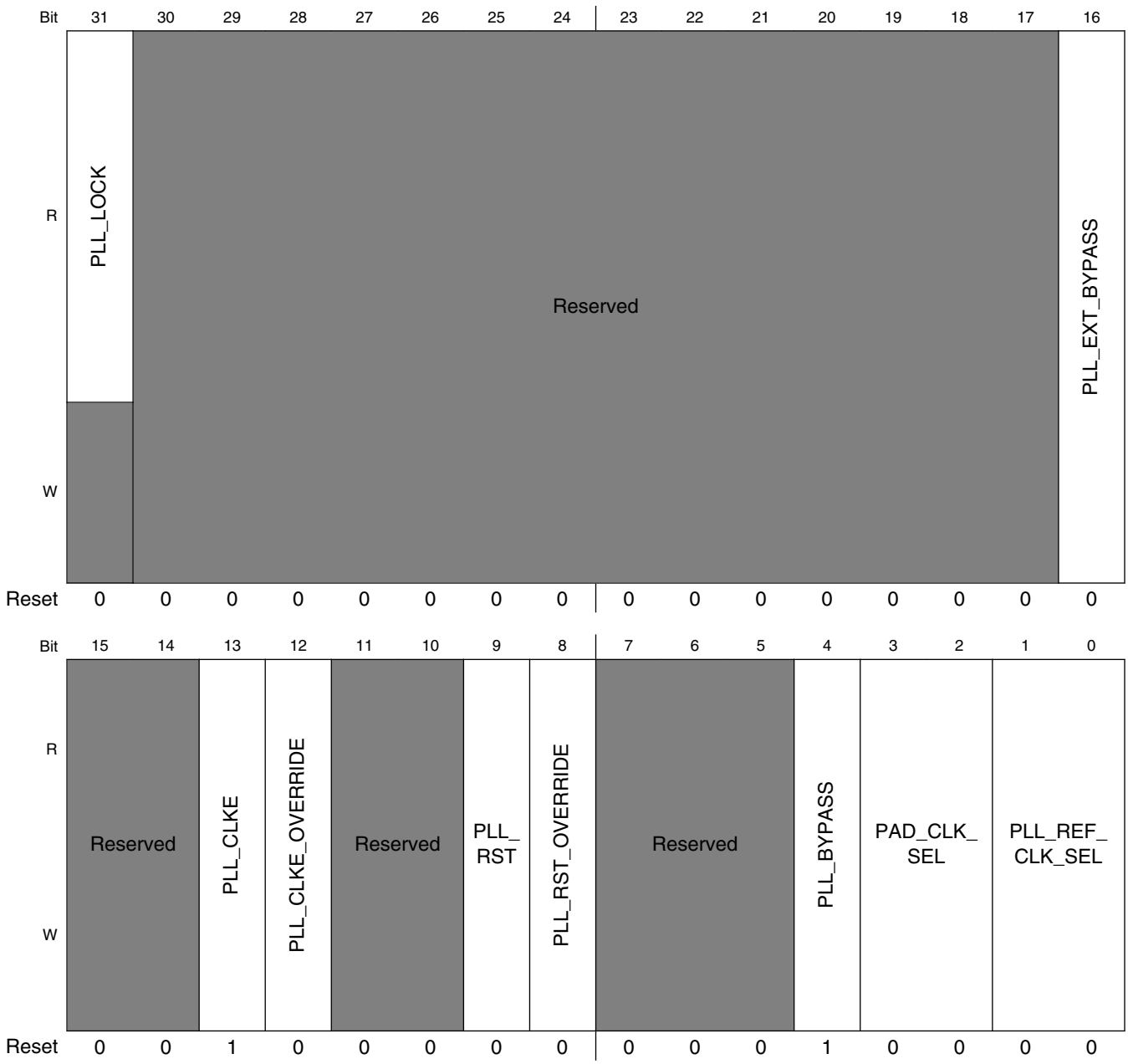
CCM_ANALOG_AUDIO_PLL1_MNIT_CTRL field descriptions (continued)

Field	Description
	0 nominal delay 1 nominal delay * 2
16 -	This field is reserved. Reserved
15 FSEL	Monitoring frequency select pin 0 FEED_OUT = FREF 1 FEED_OUT = FEED
14 FEED_EN	FEED_OUT enable pin
13–9 -	This field is reserved. Reserved
8–4 EXTAFC	Monitoring pin. If AFC_ENB=1, AFC is disabled and VCO is calibrated manually by EXTAFC[4:0] for the test of VCO range
3 AFC_EN	If AFC_ENB=0, AFC is enabled and VCO is calibrated automatically.(AFC_ENB=0 and EXTAFC=0 are mandatory) If AFC_ENB=1, AFC is disabled and VCO is calibrated manually by EXTAFC[4:0] for the test of VCO range
ICP	Controls the charge-pump current

5.1.8.6 AUDIO PLL2 General Function Control Register (CCM_ANALOG_AUDIO_PLL2_GEN_CTRL)

AUDIO PLL2 General Function Control Register

Address: 3036_0000h base + 14h offset = 3036_0014h



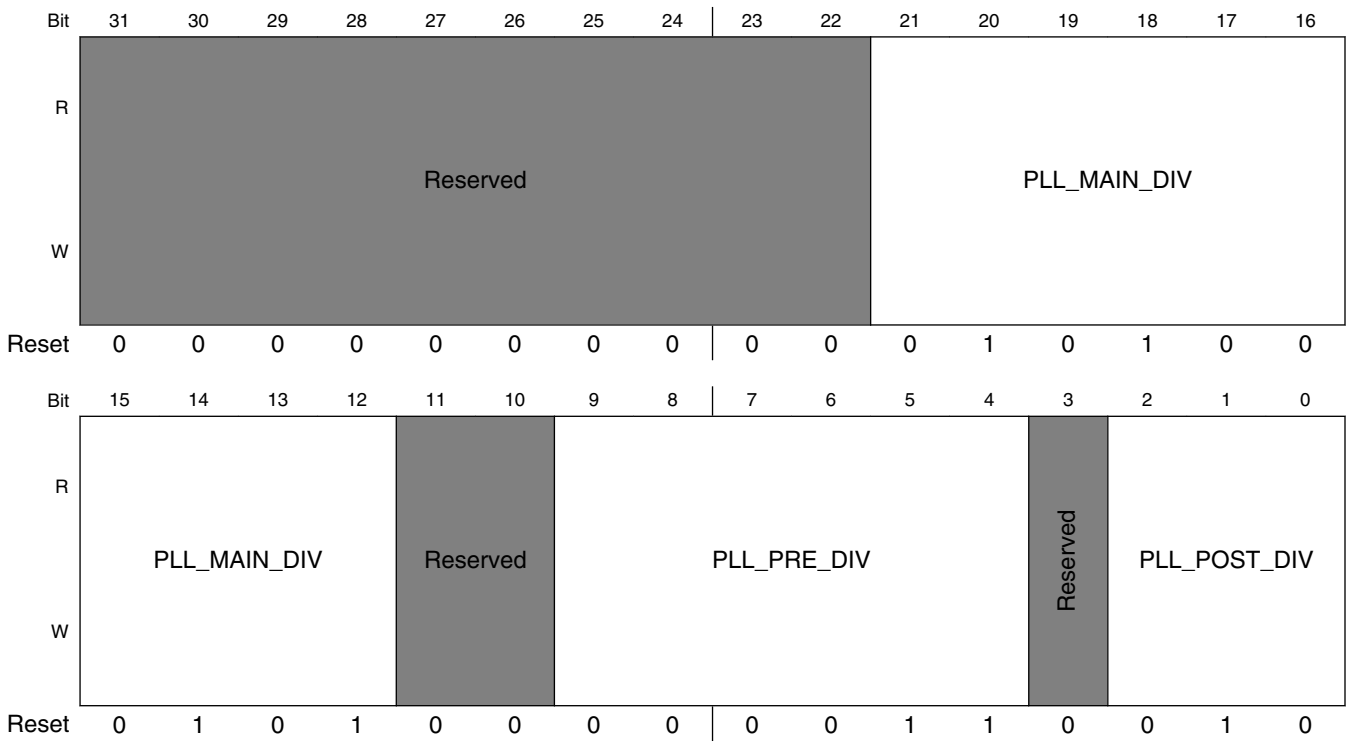
CCM_ANALOG_AUDIO_PLL2_GEN_CTRL field descriptions

Field	Description
31 PLL_LOCK	PLL lock signal
30–17 -	This field is reserved. Reserved
16 PLL_EXT_ BYPASS	PLL analog block bypass, clock output traces to PLL source
15–14 -	This field is reserved. Reserved
13 PLL_CLKE	PLL output clock clock gating enable
12 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM
11–10 -	This field is reserved. Reserved
9 PLL_RST	PLL reset (active low)
8 PLL_RST_ OVERRIDE	PLL reset overrided by CCM
7–5 -	This field is reserved. Reserved
4 PLL_BYPASS	PLL output clock bypass
3–2 PAD_CLK_SEL	PAD clock select PAD_CLK is an alternate input reference clock for the PLL. The clock source selection for PAD_CLK is defined below. 00 CLKIN1 XOR CLKIN2 01 CLKIN2 10 CLKIN1 11 Reserved
PLL_REF_CLK_ SEL	PLL reference clock select 00 24M_REF_CLK 01 PAD_CLK 10 Reserved 11 Reserved

5.1.8.7 AUDIO PLL2 Divide and Fraction Data Control 0 Register (CCM_ANALOG_AUDIO_PLL2_FDIV_CTL0)

AUDIO PLL2 Divide and Fraction Data Control 0 Register

Address: 3036_0000h base + 18h offset = 3036_0018h



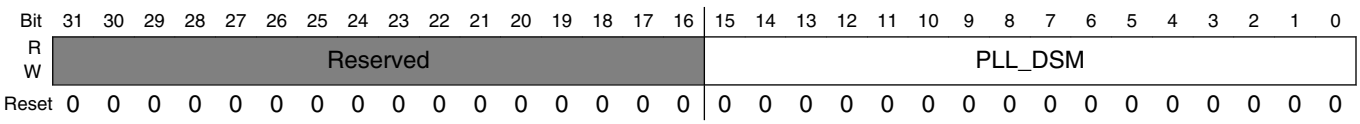
CCM_ANALOG_AUDIO_PLL2_FDIV_CTL0 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–12 PLL_MAIN_DIV	Value of the main-divider
11–10 -	This field is reserved. Reserved
9–4 PLL_PRE_DIV	Value of the pre-divider
3 -	This field is reserved. Reserved
PLL_POST_DIV	Value of the post-divider

5.1.8.8 AUDIO PLL2 Divide and Fraction Data Control 1 Register (CCM_ANALOG_AUDIO_PLL2_FDIV_CTL1)

AUDIO PLL2 Divide and Fraction Data Control 1 Register

Address: 3036_0000h base + 1Ch offset = 3036_001Ch



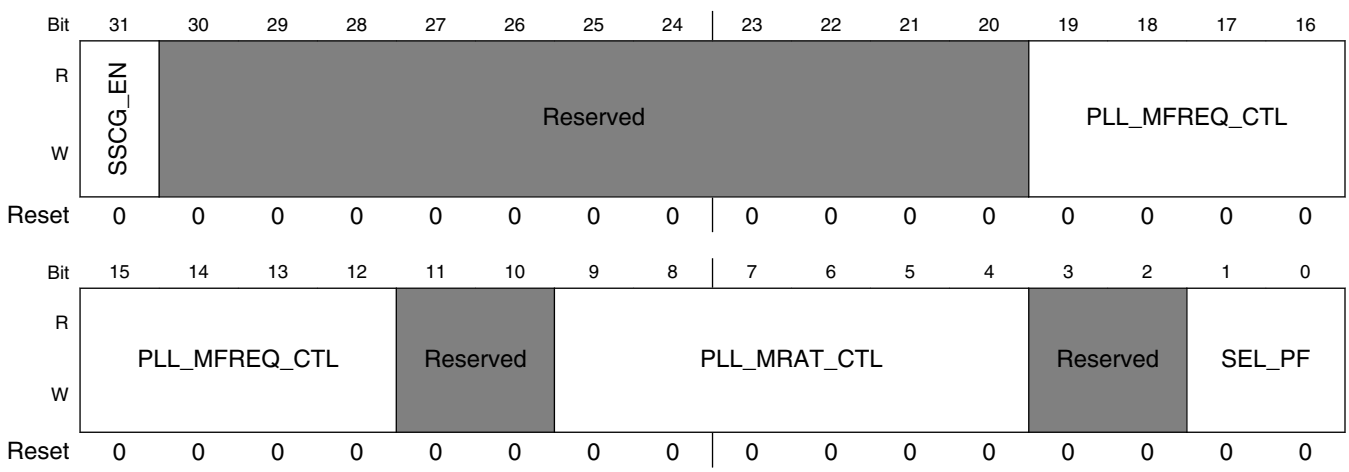
CCM_ANALOG_AUDIO_PLL2_FDIV_CTL1 field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
PLL_DSM	Value of the DSM

5.1.8.9 AUDIO PLL2 PLL SSCG Control Register (CCM_ANALOG_AUDIO_PLL2_SSCG_CTRL)

AUDIO PLL2 PLL SSCG Control Register

Address: 3036_0000h base + 20h offset = 3036_0020h



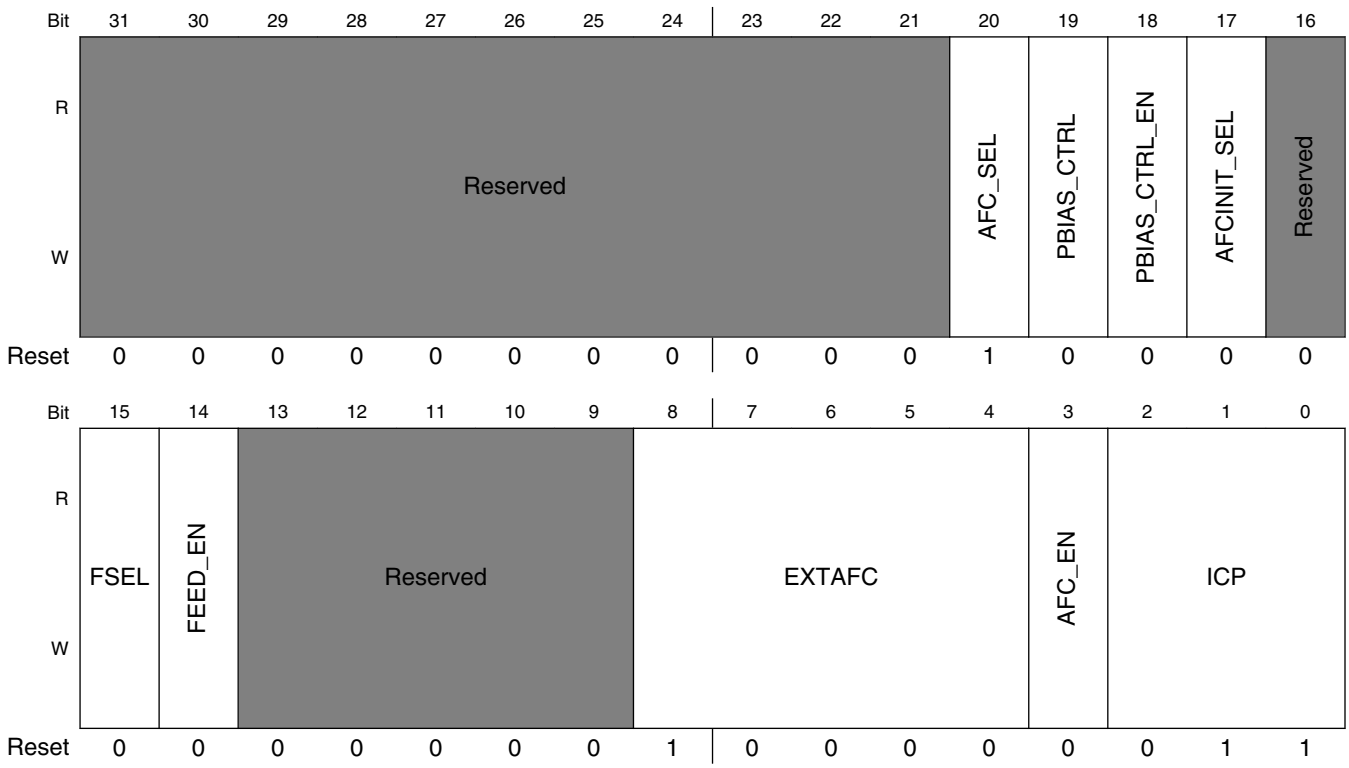
CCM_ANALOG_AUDIO_PLL2_SSCG_CTRL field descriptions

Field	Description
31 SSCG_EN	SSCG Enable 1 Enable Spread Spectrum Mode 0 Disable Spread Spectrum Mode
30–20 -	This field is reserved. Reserved
19–12 PLL_MFREQ_CTL	Value of modulation frequency control Modulation Frequency, MF is determined by the following equation: MF = F_{FIN}/p/mfr/(2^5) Hz F_{FIN} is the PLL input clock frequency, mfr is the decimal value for PLL_MFREQ_CTL[7:0], and p is the decimal value for PLL_PRE_DIV.
11–10 -	This field is reserved. Reserved
9–4 PLL_MRAT_CTL	Value of modulation rate control Modulation rate (pk-pk), MR, is determined by the following equation: MR = mfr x mrr /m /(2^6) x 100 [%] mfr is the decimal value of PLL_MFREQ_CTL, mrr is the decimal value for PLL_MRAT_CTL[5:0], and m is the decimal of PLL_MAIN_DIV.
3–2 -	This field is reserved. Reserved
SEL_PF	Value of modulation method control 00 Down spread 01 Up spread 1x Center spread

5.1.8.10 AUDIO PLL2 PLL Monitoring Control Register (CCM_ANALOG_AUDIO_PLL2_MNIT_CTRL)

AUDIO PLL2 PLL Monitoring Control Register

Address: 3036_0000h base + 24h offset = 3036_0024h



CCM_ANALOG_AUDIO_PLL2_MNIT_CTRL field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 AFC_SEL	AFC Mode select
19 PBIAS_CTRL	PBIAS pull-down initial voltage control pin 0 0.50*VDD 1 0.67*VDD
18 PBIAS_CTRL_EN	PBIAS voltage pull-down enable pin
17 AFCINIT_SEL	AFC initial delay select pin

Table continues on the next page...

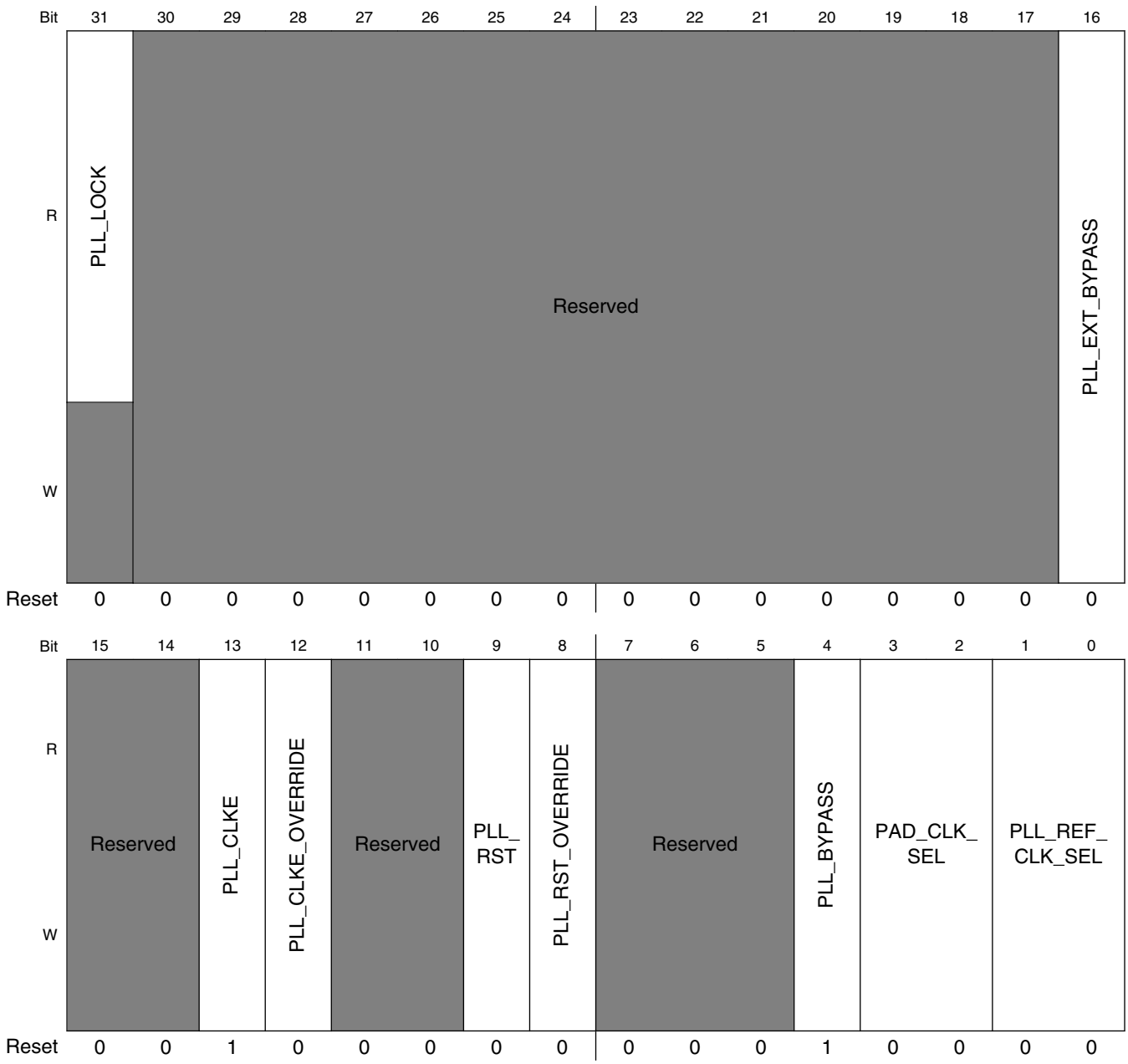
CCM_ANALOG_AUDIO_PLL2_MNIT_CTRL field descriptions (continued)

Field	Description
	0 nominal delay 1 nominal delay * 2
16 -	This field is reserved. Reserved
15 FSEL	Monitoring frequency select pin 0 FEED_OUT = FREF 1 FEED_OUT = FEED
14 FEED_EN	FEED_OUT enable pin
13–9 -	This field is reserved. Reserved
8–4 EXTAFC	Monitoring pin. If AFC_ENB=1, AFC is disabled and VCO is calibrated manually by EXTAFC[4:0] for the test of VCO range
3 AFC_EN	If AFC_ENB=0, AFC is enabled and VCO is calibrated automatically.(AFC_ENB=0 and EXTAFC=0 are mandatory) If AFC_ENB=1, AFC is disabled and VCO is calibrated manually by EXTAFC[4:0] for the test of VCO range
ICP	Controls the charge-pump current

5.1.8.11 VIDEO PLL1 General Function Control Register (CCM_ANALOG_VIDEO_PLL1_GEN_CTRL)

VIDEO PLL1 General Function Control Register

Address: 3036_0000h base + 28h offset = 3036_0028h



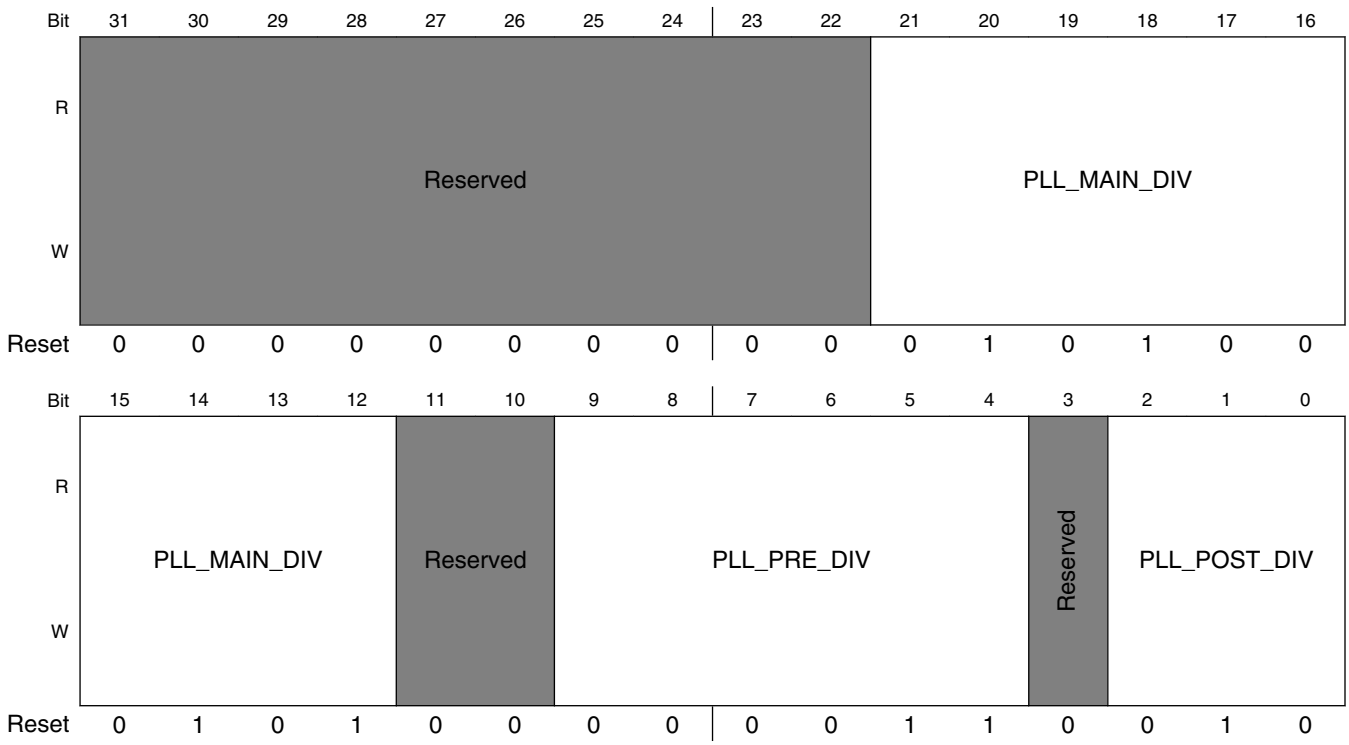
CCM_ANALOG_VIDEO_PLL1_GEN_CTRL field descriptions

Field	Description
31 PLL_LOCK	PLL lock signal
30–17 -	This field is reserved. Reserved
16 PLL_EXT_ BYPASS	PLL analog block bypass, clock output traces to PLL source
15–14 -	This field is reserved. Reserved
13 PLL_CLKE	PLL output clock clock gating enable
12 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM
11–10 -	This field is reserved. Reserved
9 PLL_RST	PLL reset (active low)
8 PLL_RST_ OVERRIDE	PLL reset overrided by CCM
7–5 -	This field is reserved. Reserved
4 PLL_BYPASS	PLL output clock bypass
3–2 PAD_CLK_SEL	PAD clock select PAD_CLK is an alternate input reference clock for the PLL. The clock source selection for PAD_CLK is defined below. 00 CLKIN1 XOR CLKIN2 01 CLKIN2 10 CLKIN1 11 Reserved
PLL_REF_CLK_ SEL	PLL reference clock select 00 24M_REF_CLK 01 PAD_CLK 10 Reserved 11 Reserved

5.1.8.12 VIDEO PLL1 Divide and Fraction Data Control 0 Register (CCM_ANALOG_VIDEO_PLL1_FDIV_CTL0)

VIDEO PLL1 Divide and Fraction Data Control 0 Register

Address: 3036_0000h base + 2Ch offset = 3036_002Ch



CCM_ANALOG_VIDEO_PLL1_FDIV_CTL0 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–12 PLL_MAIN_DIV	Value of the main-divider
11–10 -	This field is reserved. Reserved
9–4 PLL_PRE_DIV	Value of the pre-divider
3 -	This field is reserved. Reserved
PLL_POST_DIV	Value of the post-divider

5.1.8.13 VIDEO PLL1 Divide and Fraction Data Control 1 Register (CCM_ANALOG_VIDEO_PLL1_FDIV_CTL1)

VIDEO PLL1 Divide and Fraction Data Control 1 Register

Address: 3036_0000h base + 30h offset = 3036_0030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																PLL_DSM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_ANALOG_VIDEO_PLL1_FDIV_CTL1 field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
PLL_DSM	Value of the DSM

5.1.8.14 VIDEO PLL1 PLL SSCG Control Register (CCM_ANALOG_VIDEO_PLL1_SSCG_CTRL)

VIDEO PLL1 PLL SSCG Control Register

Address: 3036_0000h base + 34h offset = 3036_0034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SSCG_EN	Reserved											PLL_MFREQ_CTL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_MFREQ_CTL				Reserved		PLL_MRAT_CTL						Reserved		SEL_PF	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

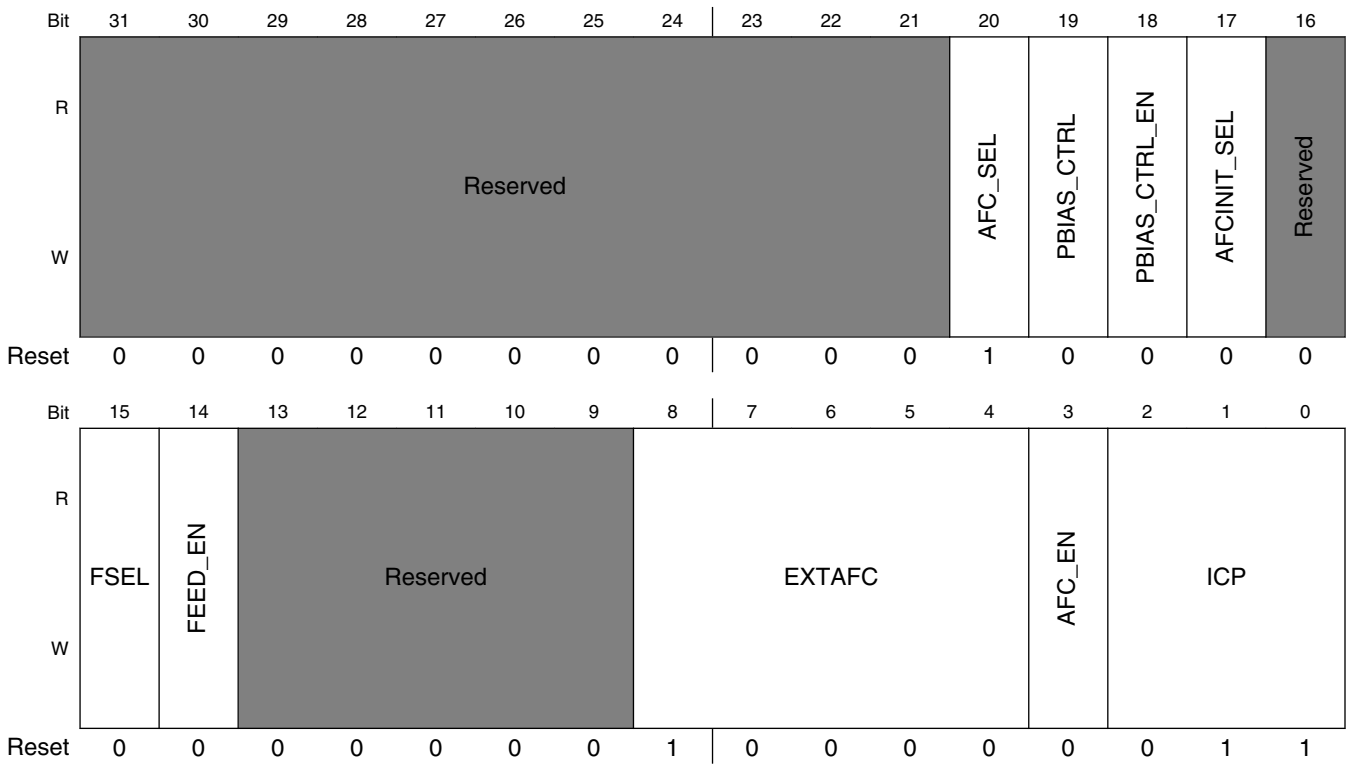
CCM_ANALOG_VIDEO_PLL1_SSCG_CTRL field descriptions

Field	Description
31 SSCG_EN	SSCG Enable 1 Enable Spread Spectrum Mode 0 Disable Spread Spectrum Mode
30–20 -	This field is reserved. Reserved
19–12 PLL_MFREQ_CTL	Value of modulation frequency control Modulation Frequency, MF is determined by the following equation: MF = F_{FIN}/p/mfr/(2^5) Hz F_{FIN} is the PLL input clock frequency, mfr is the decimal value for PLL_MFREQ_CTL[7:0], and p is the decimal value for PLL_PRE_DIV.
11–10 -	This field is reserved. Reserved
9–4 PLL_MRAT_CTL	Value of modulation rate control Modulation rate (pk-pk), MR, is determined by the following equation: MR = mfr x mrr /m /(2^6) x 100 [%] mfr is the decimal value of PLL_MFREQ_CTL, mrr is the decimal value for PLL_MRAT_CTL[5:0], and m is the decimal of PLL_MAIN_DIV.
3–2 -	This field is reserved. Reserved
SEL_PF	Value of modulation method control 00 Down spread 01 Up spread 1x Center spread

5.1.8.15 VIDEO PLL1 PLL Monitoring Control Register (CCM_ANALOG_VIDEO_PLL1_MNIT_CTRL)

VIDEO PLL1 PLL Monitoring Control Register

Address: 3036_0000h base + 38h offset = 3036_0038h



CCM_ANALOG_VIDEO_PLL1_MNIT_CTRL field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 AFC_SEL	AFC Mode select
19 PBIAS_CTRL	PBIAS pull-down initial voltage control pin 0 0.50*VDD 1 0.67*VDD
18 PBIAS_CTRL_EN	PBIAS voltage pull-down enable pin
17 AFCINIT_SEL	AFC initial delay select pin

Table continues on the next page...

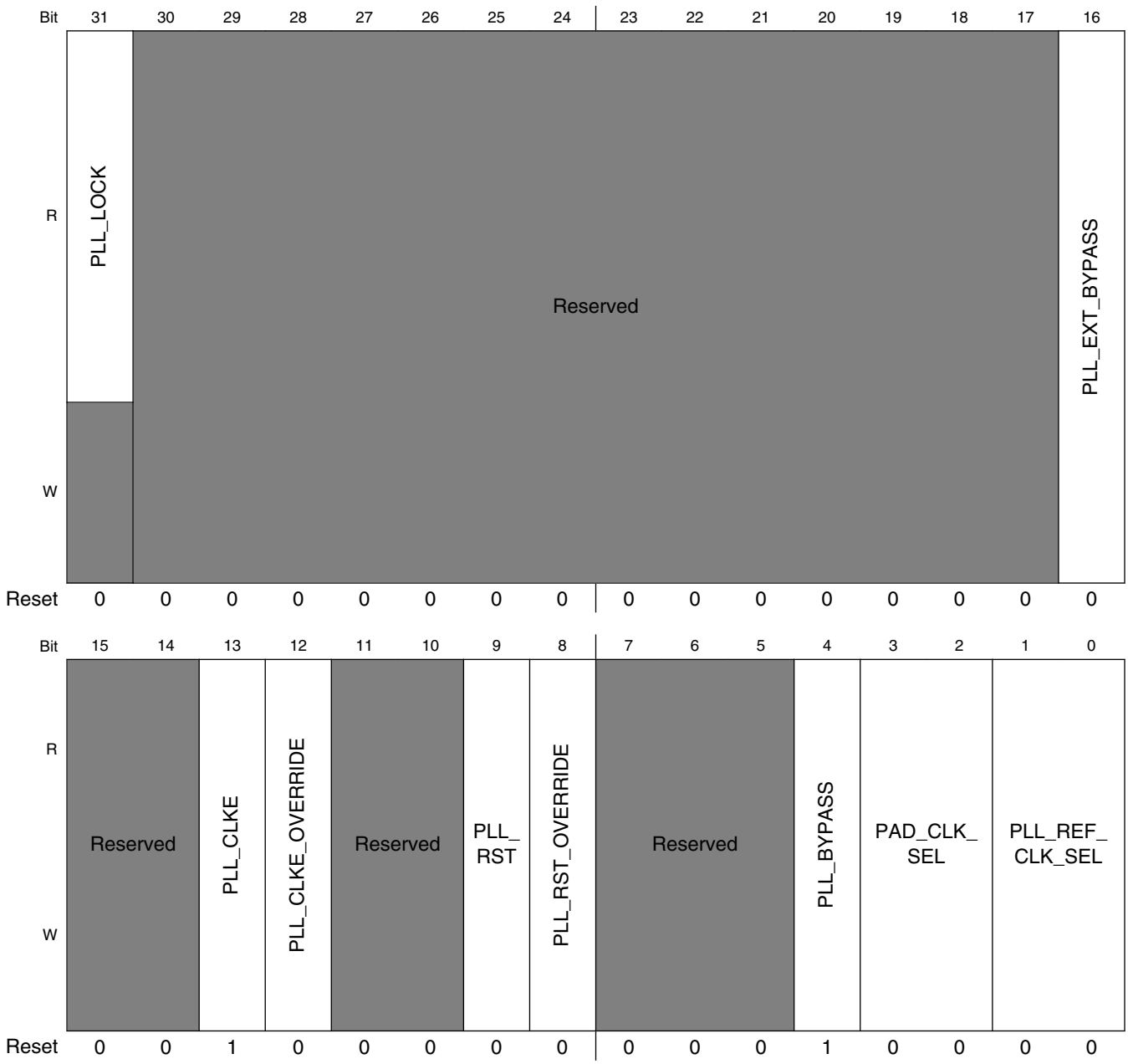
CCM_ANALOG_VIDEO_PLL1_MNIT_CTRL field descriptions (continued)

Field	Description
	0 nominal delay 1 nominal delay * 2
16 -	This field is reserved. Reserved
15 FSEL	Monitoring frequency select pin 0 FEED_OUT = FREF 1 FEED_OUT = FEED
14 FEED_EN	FEED_OUT enable pin
13–9 -	This field is reserved. Reserved
8–4 EXTAFC	Monitoring pin. If AFC_ENB=1, AFC is disabled and VCO is calibrated manually by EXTAFC[4:0] for the test of VCO range
3 AFC_EN	If AFC_ENB=0, AFC is enabled and VCO is calibrated automatically.(AFC_ENB=0 and EXTAFC=0 are mandatory) If AFC_ENB=1, AFC is disabled and VCO is calibrated manually by EXTAFC[4:0] for the test of VCO range
ICP	Controls the charge-pump current

5.1.8.16 DRAM PLL General Function Control Register (CCM_ANALOG_DRAM_PLL_GEN_CTRL)

DRAM PLL General Function Control Register

Address: 3036_0000h base + 50h offset = 3036_0050h



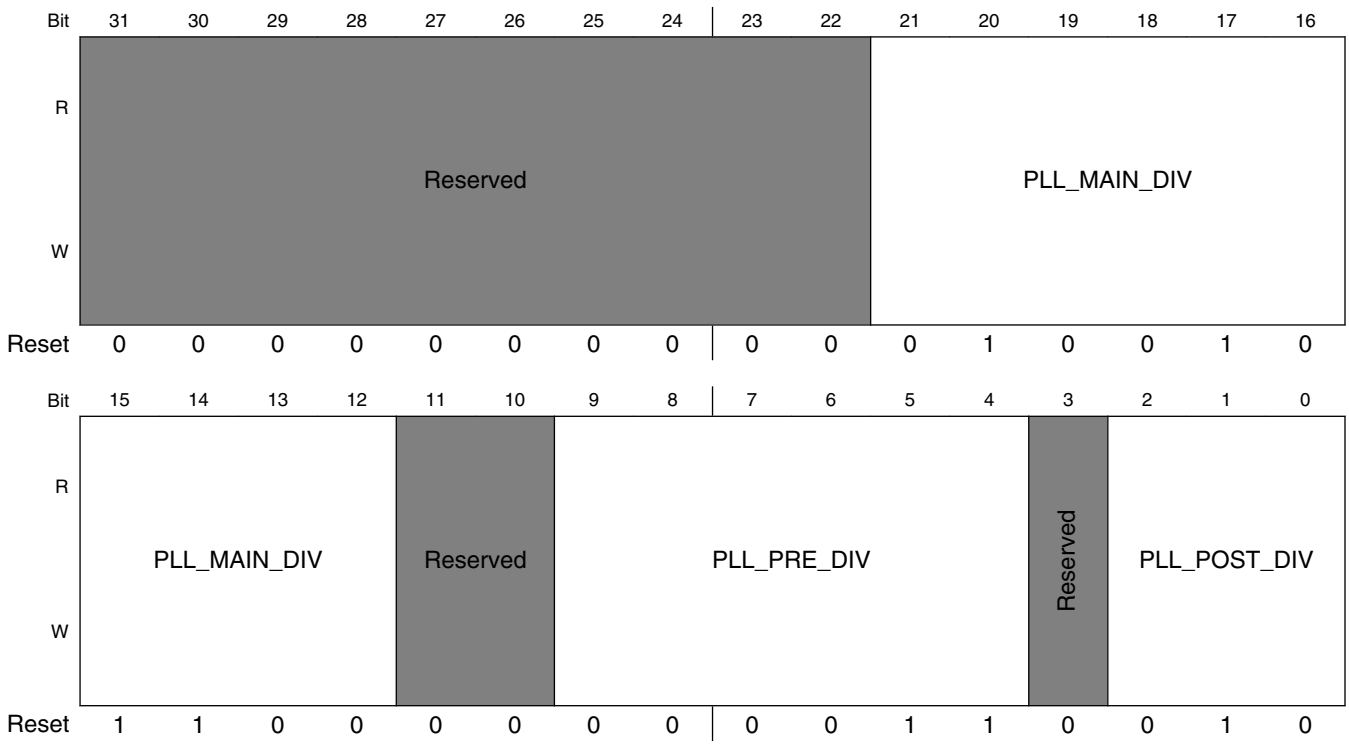
CCM_ANALOG_DRAM_PLL_GEN_CTRL field descriptions

Field	Description
31 PLL_LOCK	PLL lock signal
30–17 -	This field is reserved. Reserved
16 PLL_EXT_ BYPASS	PLL analog block bypass, clock output traces to PLL source
15–14 -	This field is reserved. Reserved
13 PLL_CLKE	PLL output clock clock gating enable
12 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM
11–10 -	This field is reserved. Reserved
9 PLL_RST	PLL reset (active low)
8 PLL_RST_ OVERRIDE	PLL reset overrided by CCM
7–5 -	This field is reserved. Reserved
4 PLL_BYPASS	PLL output clock bypass
3–2 PAD_CLK_SEL	PAD clock select PAD_CLK is an alternate input reference clock for the PLL. The clock source selection for PAD_CLK is defined below. 00 CLKIN1 XOR CLKIN2 01 CLKIN2 10 CLKIN1 11 Reserved
PLL_REF_CLK_ SEL	PLL reference clock select 00 24M_REF_CLK 01 PAD_CLK 10 Reserved 11 Reserved

5.1.8.17 DRAM PLL Divide and Fraction Data Control 0 Register (CCM_ANALOG_DRAM_PLL_FDIV_CTL0)

DRAM PLL Divide and Fraction Data Control 0 Register

Address: 3036_0000h base + 54h offset = 3036_0054h



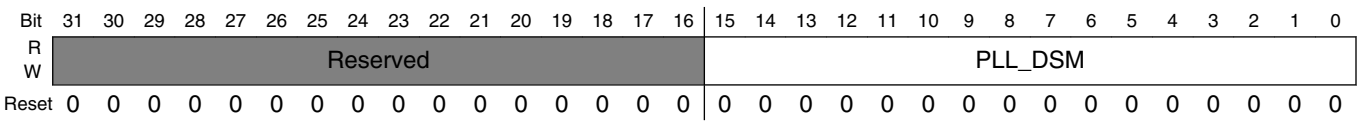
CCM_ANALOG_DRAM_PLL_FDIV_CTL0 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–12 PLL_MAIN_DIV	Value of the main-divider
11–10 -	This field is reserved. Reserved
9–4 PLL_PRE_DIV	Value of the pre-divider
3 -	This field is reserved. Reserved
PLL_POST_DIV	Value of the post-divider

5.1.8.18 DRAM PLL Divide and Fraction Data Control 1 Register (CCM_ANALOG_DRAM_PLL_FDIV_CTL1)

DRAM PLL Divide and Fraction Data Control 1 Register

Address: 3036_0000h base + 58h offset = 3036_0058h



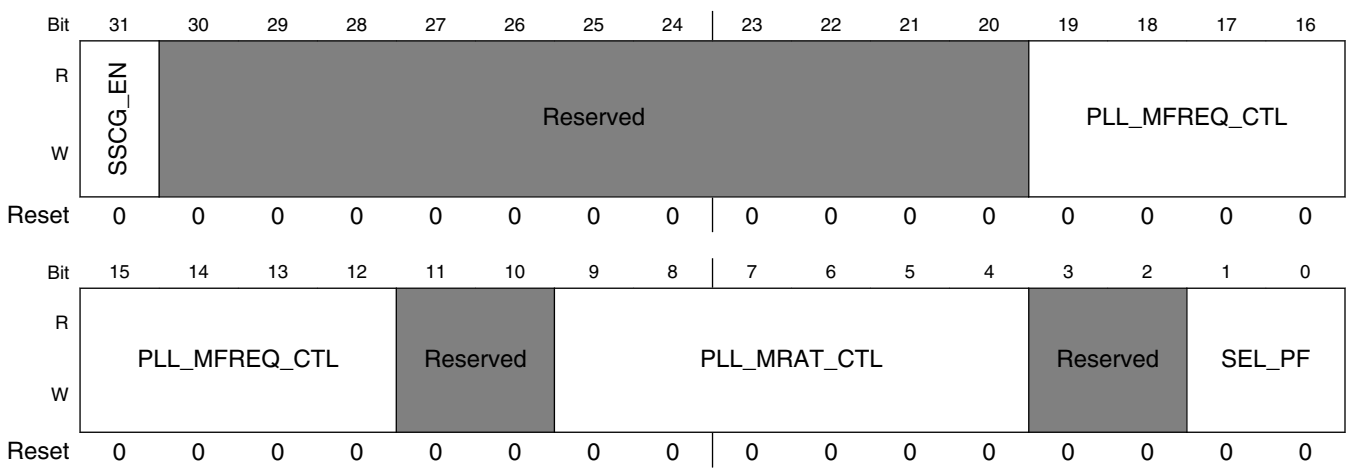
CCM_ANALOG_DRAM_PLL_FDIV_CTL1 field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
PLL_DSM	Value of the DSM

5.1.8.19 DRAM PLL PLL SSCG Control Register (CCM_ANALOG_DRAM_PLL_SSCG_CTRL)

DRAM PLL PLL SSCG Control Register

Address: 3036_0000h base + 5Ch offset = 3036_005Ch



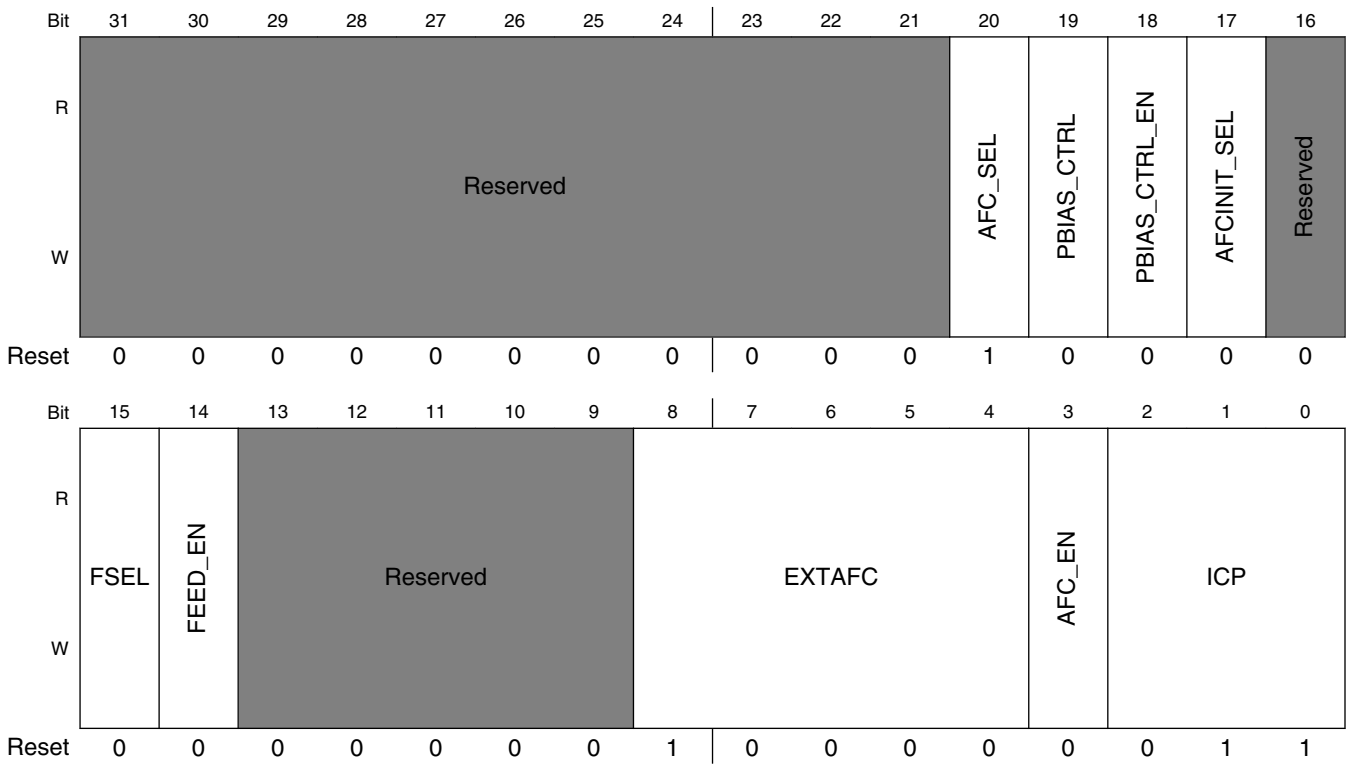
CCM_ANALOG_DRAM_PLL_SSCG_CTRL field descriptions

Field	Description
31 SSCG_EN	SSCG Enable 1 Enable Spread Spectrum Mode 0 Disable Spread Spectrum Mode
30–20 -	This field is reserved. Reserved
19–12 PLL_MFREQ_CTL	Value of modulation frequency control Modulation Frequency, MF is determined by the following equation: MF = F_{FIN}/p/mfr/(2^5) Hz F_{FIN} is the PLL input clock frequency, mfr is the decimal value for PLL_MFREQ_CTL[7:0], and p is the decimal value for PLL_PRE_DIV.
11–10 -	This field is reserved. Reserved
9–4 PLL_MRAT_CTL	Value of modulation rate control Modulation rate (pk-pk), MR, is determined by the following equation: MR = mfr × mrr / m / f(2^6) × 100 [%] mfr is the decimal value of PLL_MFREQ_CTL, mrr is the decimal value for PLL_MRAT_CTL[5:0], and m is the decimal of PLL_MAIN_DIV.
3–2 -	This field is reserved. Reserved
SEL_PF	Value of modulation method control 00 Down spread 01 Up spread 1x Center spread

5.1.8.20 DRAM PLL PLL Monitoring Control Register (CCM_ANALOG_DRAM_PLL_MNIT_CTRL)

DRAM PLL PLL Monitoring Control Register

Address: 3036_0000h base + 60h offset = 3036_0060h



CCM_ANALOG_DRAM_PLL_MNIT_CTRL field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 AFC_SEL	AFC Mode select
19 PBIAS_CTRL	PBIAS pull-down initial voltage control pin 0 0.50 × VDD 1 0.67 × VDD
18 PBIAS_CTRL_EN	PBIAS voltage pull-down enable pin
17 AFCINIT_SEL	AFC initial delay select pin

Table continues on the next page...

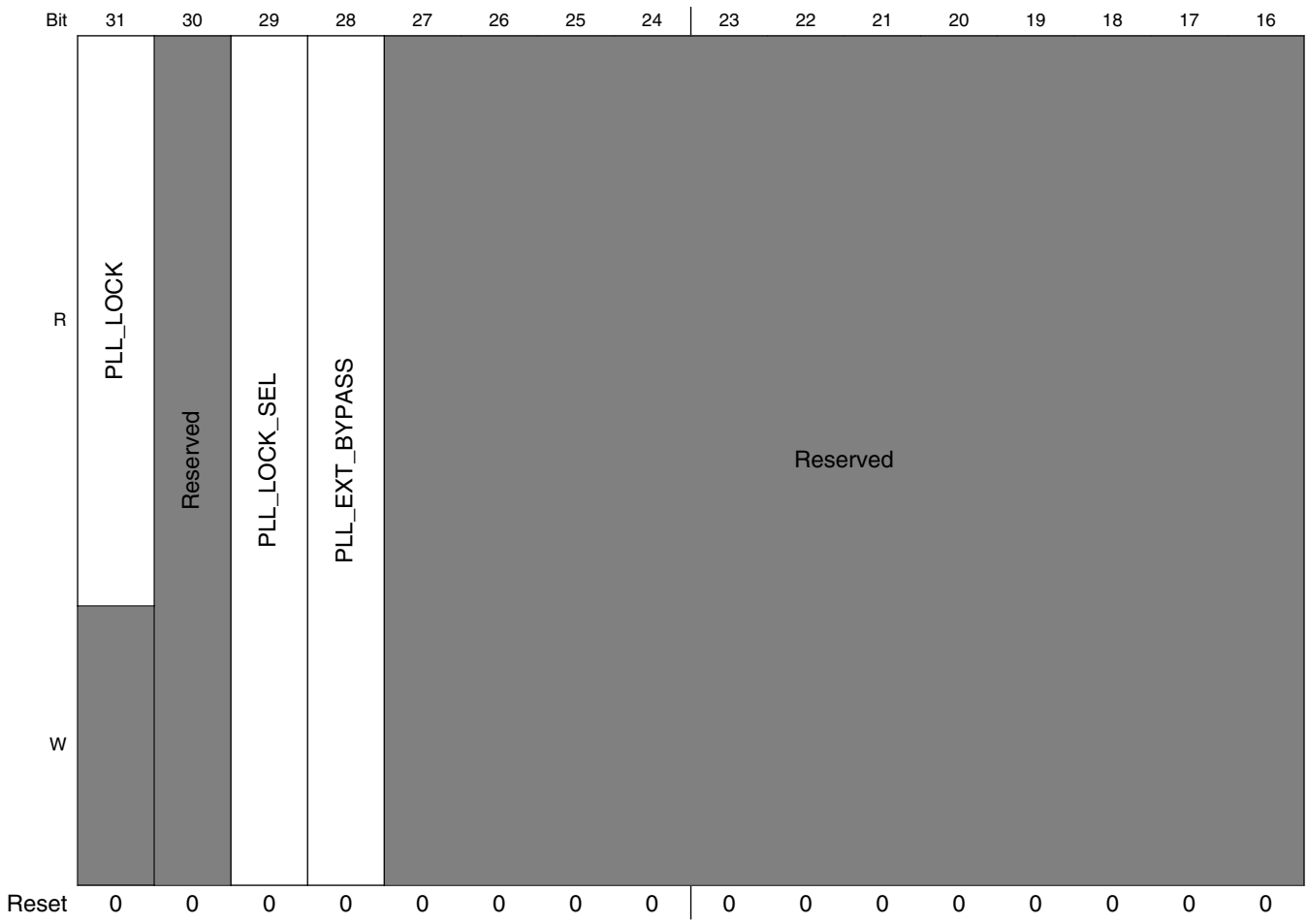
CCM_ANALOG_DRAM_PLL_MNIT_CTRL field descriptions (continued)

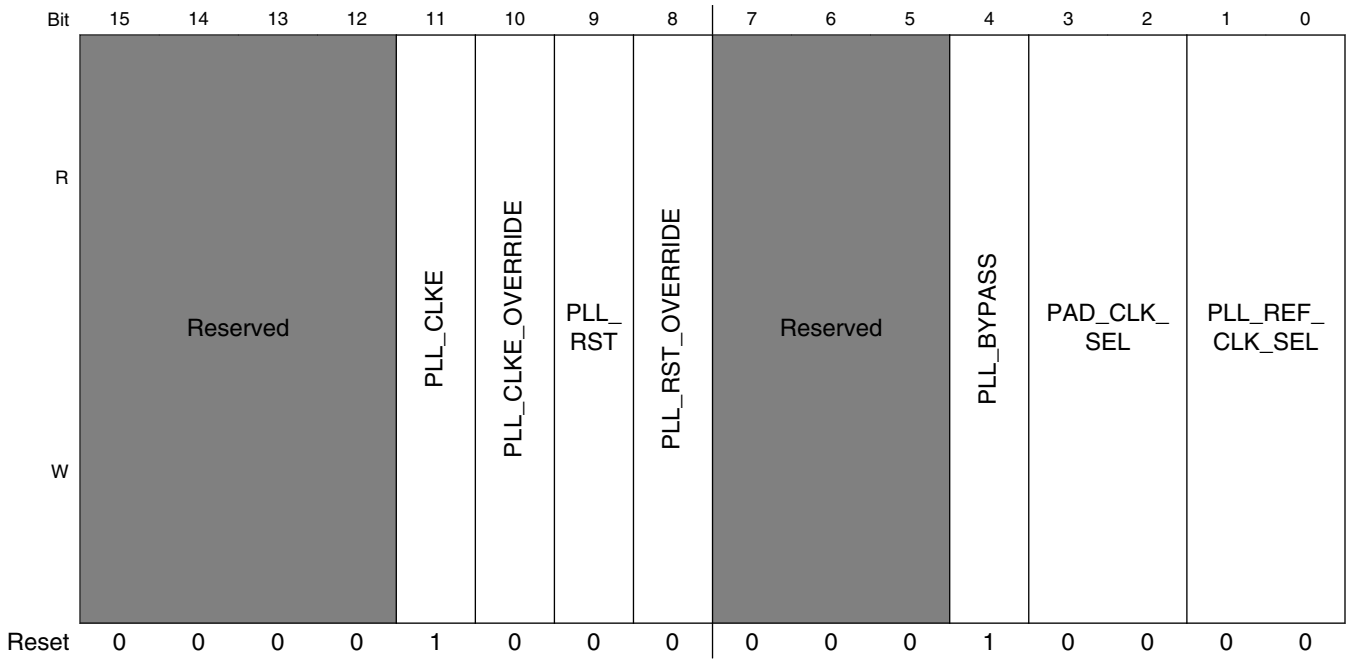
Field	Description
	0 nominal delay 1 nominal delay × 2
16 -	This field is reserved. Reserved
15 FSEL	Monitoring frequency select pin 0 FEED_OUT = FREF 1 FEED_OUT = FEED
14 FEED_EN	FEED_OUT enable pin
13–9 -	This field is reserved. Reserved
8–4 EXTAFC	Monitoring pin. If AFC_ENB=1, AFC is disabled and VCO is calibrated manually by EXTAFC[4:0] for the test of VCO range
3 AFC_EN	If AFC_ENB=0, AFC is enabled and VCO is calibrated automatically.(AFC_ENB=0 and EXTAFC=0 are mandatory) If AFC_ENB=1, AFC is disabled and VCO is calibrated manually by EXTAFC[4:0] for the test of VCO range
ICP	Controls the charge-pump current

5.1.8.21 GPU PLL General Function Control Register (CCM_ANALOG_GPU_PLL_GEN_CTRL)

GPU PLL General Function Control Register

Address: 3036_0000h base + 64h offset = 3036_0064h





CCM_ANALOG_GPU_PLL_GEN_CTRL field descriptions

Field	Description
31 PLL_LOCK	PLL lock signal
30 -	This field is reserved. Reserved
29 PLL_LOCK_SEL	PLL lock select 0 Using PLL maximum lock time 1 Using PLL output lock
28 PLL_EXT_BYPASS	PLL analog block bypass, clock output traces to PLL source
27–12 -	This field is reserved. Reserved
11 PLL_CLKE	PLL output clock clock gating enable
10 PLL_CLKE_OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM
9 PLL_RST	PLL reset (active low)
8 PLL_RST_OVERRIDE	PLL reset overrided by CCM
7–5 -	This field is reserved. Reserved
4 PLL_BYPASS	PLL output clock bypass

Table continues on the next page...

CCM_ANALOG_GPU_PLL_GEN_CTRL field descriptions (continued)

Field	Description
3–2 PAD_CLK_SEL	PAD clock select PAD_CLK is an alternate input reference clock for the PLL. The clock source selection for PAD_CLK is defined below. 00 CLKIN1 XOR CLKIN2 01 CLKIN2 10 CLKIN1 11 Reserved
PLL_REF_CLK_SEL	PLL reference clock select 00 24M_REF_CLK 01 PAD_CLK 10 Reserved 11 Reserved

5.1.8.22 GPU PLL Divide and Fraction Data Control 0 Register (CCM_ANALOG_GPU_PLL_FDIV_CTL0)

GPU PLL Divide and Fraction Data Control 0 Register

Address: 3036_0000h base + 68h offset = 3036_0068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved										PLL_MAIN_DIV					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_MAIN_DIV				Reserved		PLL_PRE_DIV						Reserved	PLL_POST_DIV		
Reset	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1

CCM_ANALOG_GPU_PLL_FDIV_CTL0 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–12 PLL_MAIN_DIV	Value of the main-divider
11–10 -	This field is reserved. Reserved
9–4 PLL_PRE_DIV	Value of the pre-divider
3 -	This field is reserved. Reserved
PLL_POST_DIV	Value of the post-divider

5.1.8.23 PLL Lock Detector Control Register (CCM_ANALOG_*n*LOCKD_CTRL)

PLL Lock Detector Control Register

Address: 3036_0000h base + 6Ch offset + (16d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										LOCK_CON_DLY		LOCK_CON_OUT		LOCK_CON_IN	
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

CCM_ANALOG_*n*LOCKD_CTRL field descriptions

Field	Description
31–6 -	This field is reserved. Reserved
5–4 LOCK_CON_DLY	Lock detector setting of the detection resolution

Table continues on the next page...

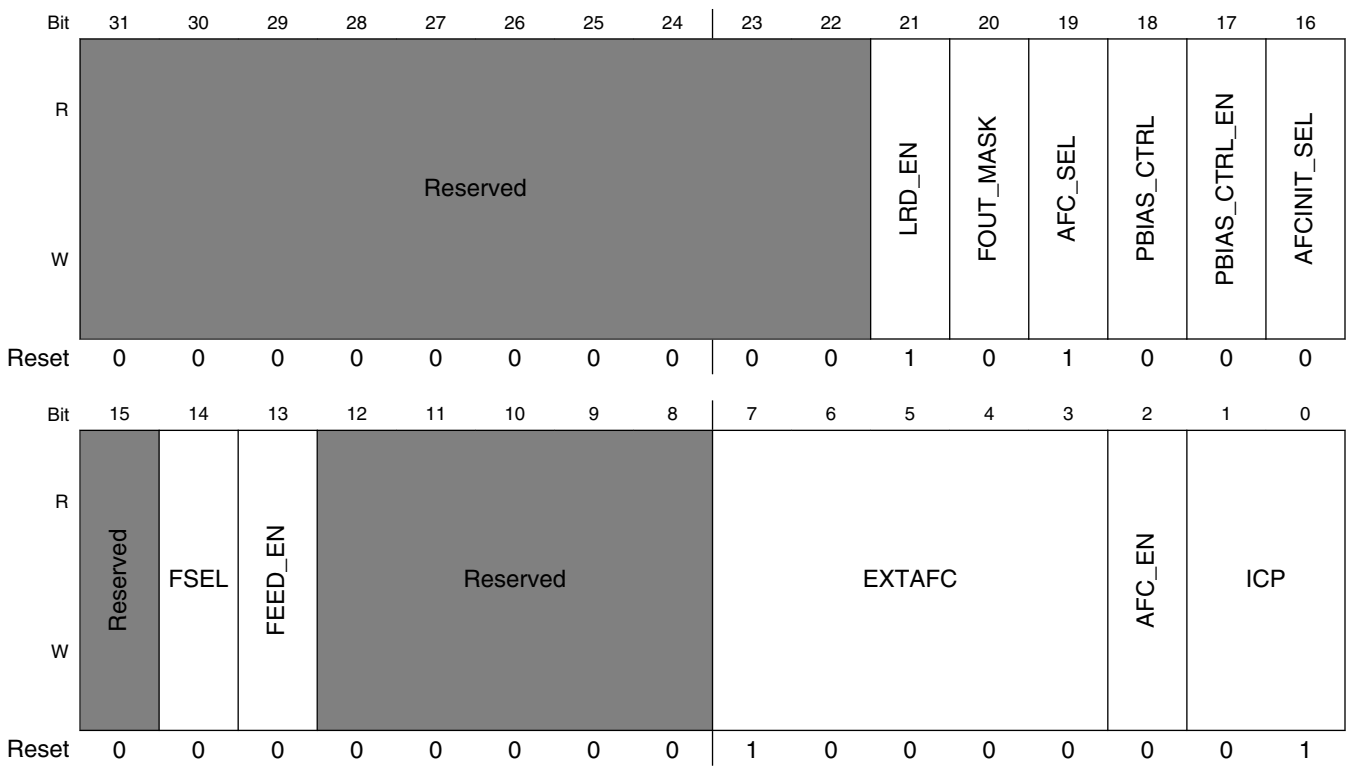
CCM_ANALOG_*n*LOCKD_CTRL field descriptions (continued)

Field	Description
3–2 LOCK_CON_ OUT	Lock detector setting of the output margin
LOCK_CON_IN	Lock detector setting of the input margin

5.1.8.24 PLL Monitoring Control Register (CCM_ANALOG_*n*MNIT_CTRL)

PLL Monitoring Control Register

Address: 3036_0000h base + 70h offset + (16d × i), where i=0d to 2d



CCM_ANALOG_*n*MNIT_CTRL field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21 LRD_EN	Monitoring pin. AFC operation mode select pin
20 FOUT_MASK	Scaler's re-initialization time control pin[3]

Table continues on the next page...

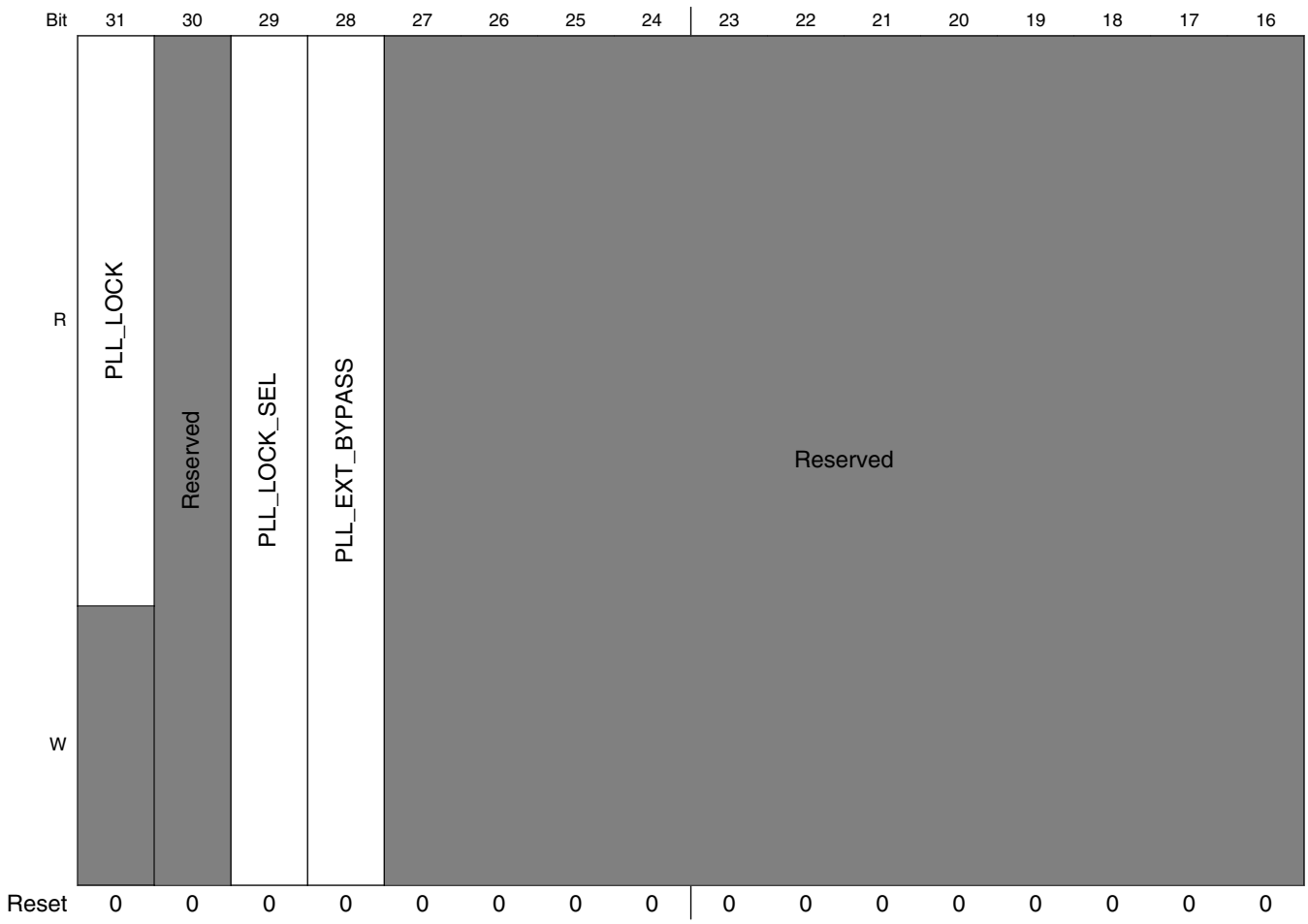
CCM_ANALOG_*n*MNIT_CTRL field descriptions (continued)

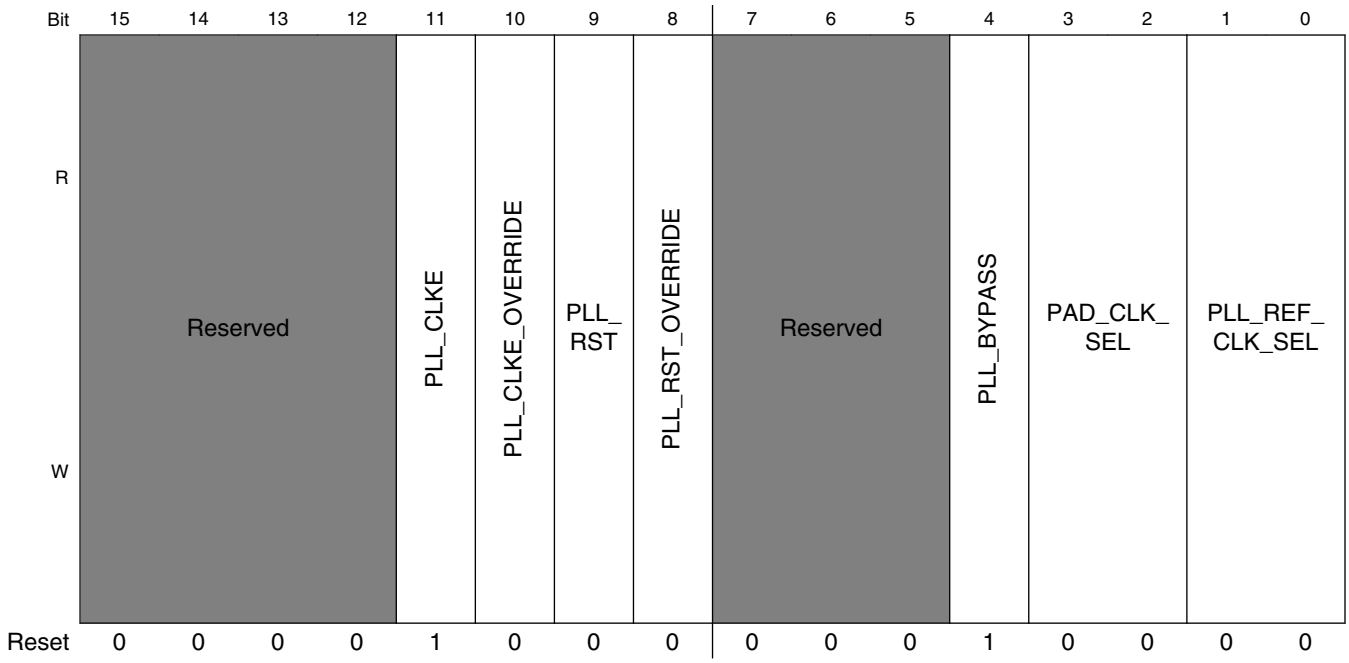
Field	Description
19 AFC_SEL	AFC Mode select
18 PBIAS_CTRL	PBIAS pull-down initial voltage control pin 0 0.50 × VDD 1 0.67 × VDD
17 PBIAS_CTRL_EN	PBIAS voltage pull-down enable pin
16 AFCINIT_SEL	AFC initial delay select pin 0 nominal delay 1 nominal delay × 2
15 -	This field is reserved. Reserved
14 FSEL	Monitoring frequency select pin 0 FEED_OUT = FREF 1 FEED_OUT = FEED
13 FEED_EN	FEED_OUT enable pin
12–8 -	This field is reserved. Reserved
7–3 EXTAFC	Monitoring pin. If AFC_ENB=1, AFC is disabled and VCO is calibrated manually by EXTAFC[4:0] for the test of VCO range
2 AFC_EN	If AFC_ENB=0, AFC is enabled and VCO is calibrated automatically.(AFC_ENB=0 and EXTAFC=0 are mandatory) If AFC_ENB=1, AFC is disabled and VCO is calibrated manually by EXTAFC[4:0] for the test of VCO range
ICP	Controls the charge-pump current

5.1.8.25 VPU PLL General Function Control Register (CCM_ANALOG_VPU_PLL_GEN_CTRL)

VPU PLL General Function Control Register

Address: 3036_0000h base + 74h offset = 3036_0074h





CCM_ANALOG_VPU_PLL_GEN_CTRL field descriptions

Field	Description
31 PLL_LOCK	PLL lock signal
30 -	This field is reserved. Reserved
29 PLL_LOCK_SEL	PLL lock select 0 Using PLL maximum lock time 1 Using PLL output lock
28 PLL_EXT_BYPASS	PLL analog block bypass, clock output traces to PLL source
27–12 -	This field is reserved. Reserved
11 PLL_CLKE	PLL output clock clock gating enable
10 PLL_CLKE_OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM
9 PLL_RST	PLL reset (active low)
8 PLL_RST_OVERRIDE	PLL reset overrided by CCM
7–5 -	This field is reserved. Reserved
4 PLL_BYPASS	PLL output clock bypass

Table continues on the next page...

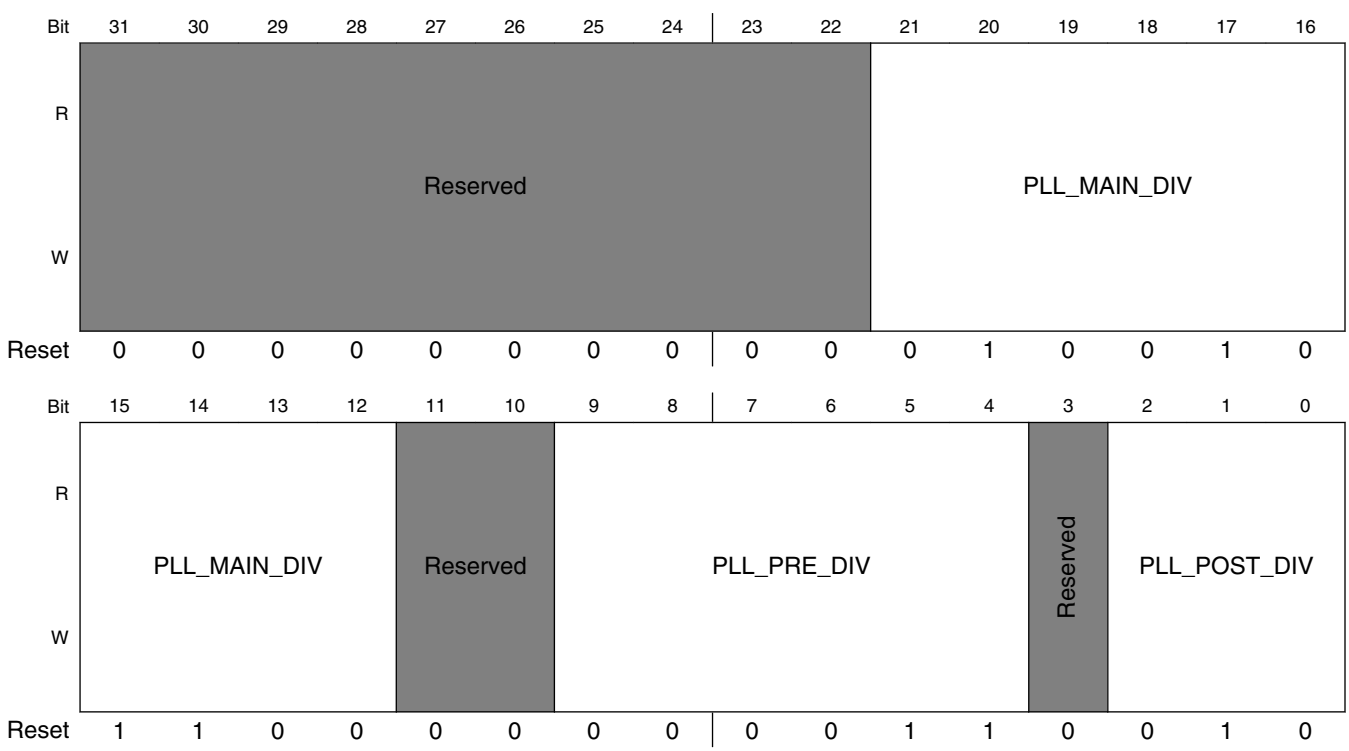
CCM_ANALOG_VPU_PLL_GEN_CTRL field descriptions (continued)

Field	Description
3–2 PAD_CLK_SEL	PAD clock select PAD_CLK is an alternate input reference clock for the PLL. The clock source selection for PAD_CLK is defined below. 00 CLKIN1 XOR CLKIN2 01 CLKIN2 10 CLKIN1 11 Reserved
PLL_REF_CLK_SEL	PLL reference clock select 00 24M_REF_CLK 01 PAD_CLK 10 Reserved 11 Reserved

5.1.8.26 VPU PLL Divide and Fraction Data Control 0 Register (CCM_ANALOG_VPU_PLL_FDIV_CTL0)

VPU PLL Divide and Fraction Data Control 0 Register

Address: 3036_0000h base + 78h offset = 3036_0078h



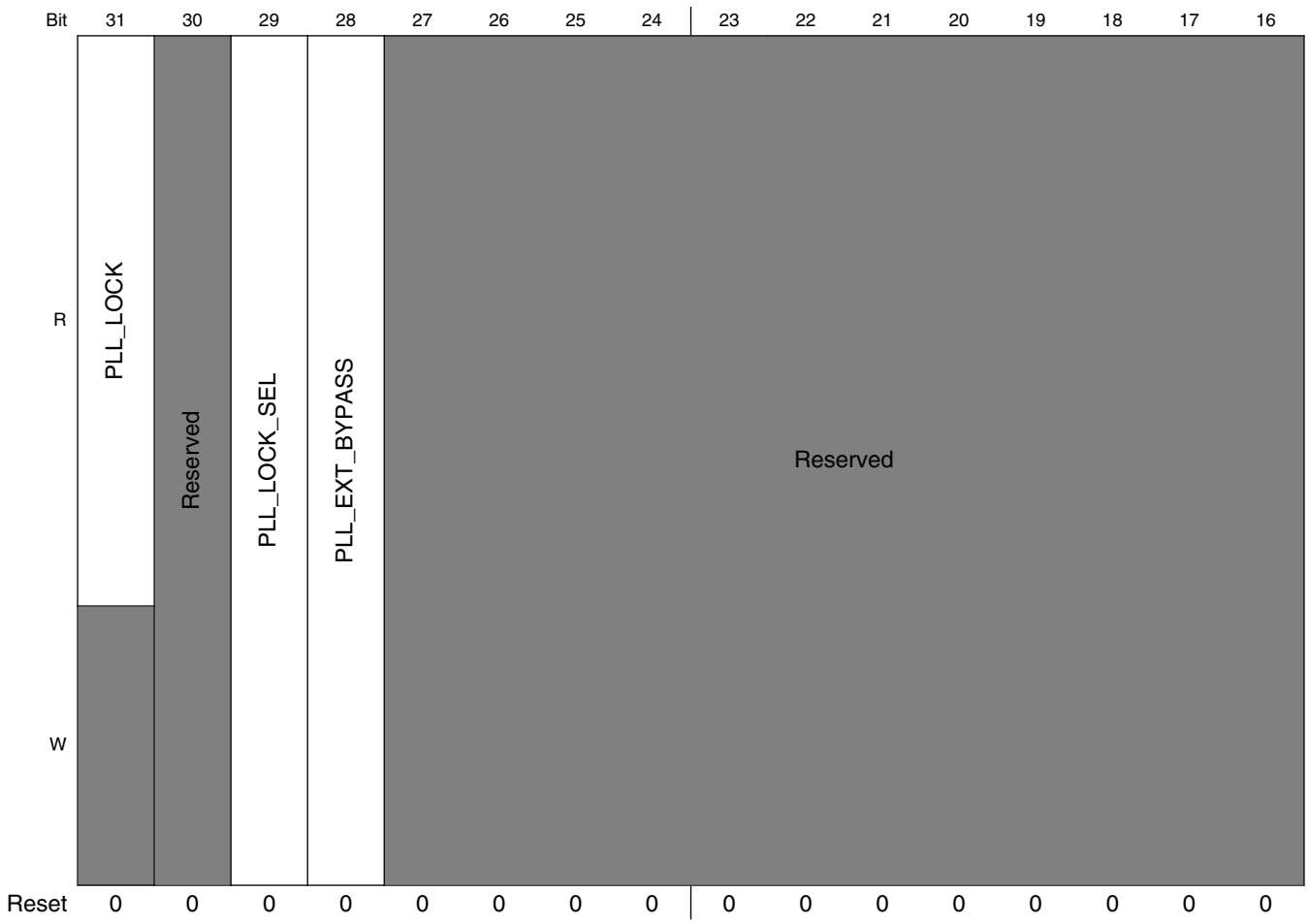
CCM_ANALOG_VPU_PLL_FDIV_CTL0 field descriptions

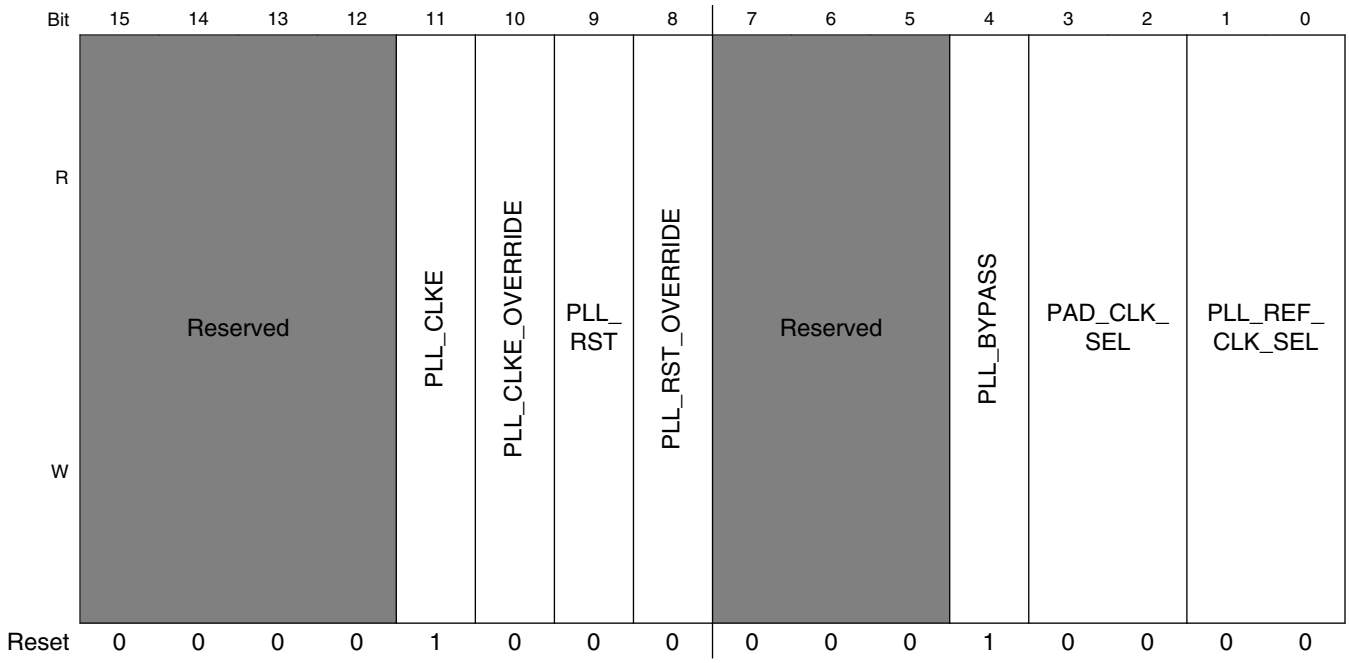
Field	Description
31–22 -	This field is reserved. Reserved
21–12 PLL_MAIN_DIV	Value of the main-divider
11–10 -	This field is reserved. Reserved
9–4 PLL_PRE_DIV	Value of the pre-divider
3 -	This field is reserved. Reserved
PLL_POST_DIV	Value of the post-divider

5.1.8.27 ARM PLL General Function Control Register (CCM_ANALOG_ARM_PLL_GEN_CTRL)

ARM PLL General Function Control Register

Address: 3036_0000h base + 84h offset = 3036_0084h





CCM_ANALOG_ARM_PLL_GEN_CTRL field descriptions

Field	Description
31 PLL_LOCK	PLL lock signal
30 -	This field is reserved. Reserved
29 PLL_LOCK_SEL	PLL lock select 0 Using PLL maximum lock time 1 Using PLL output lock
28 PLL_EXT_BYPASS	PLL analog block bypass, clock output traces to PLL source
27–12 -	This field is reserved. Reserved
11 PLL_CLKE	PLL output clock clock gating enable
10 PLL_CLKE_OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM
9 PLL_RST	PLL reset (active low)
8 PLL_RST_OVERRIDE	PLL reset overrided by CCM
7–5 -	This field is reserved. Reserved
4 PLL_BYPASS	PLL output clock bypass

Table continues on the next page...

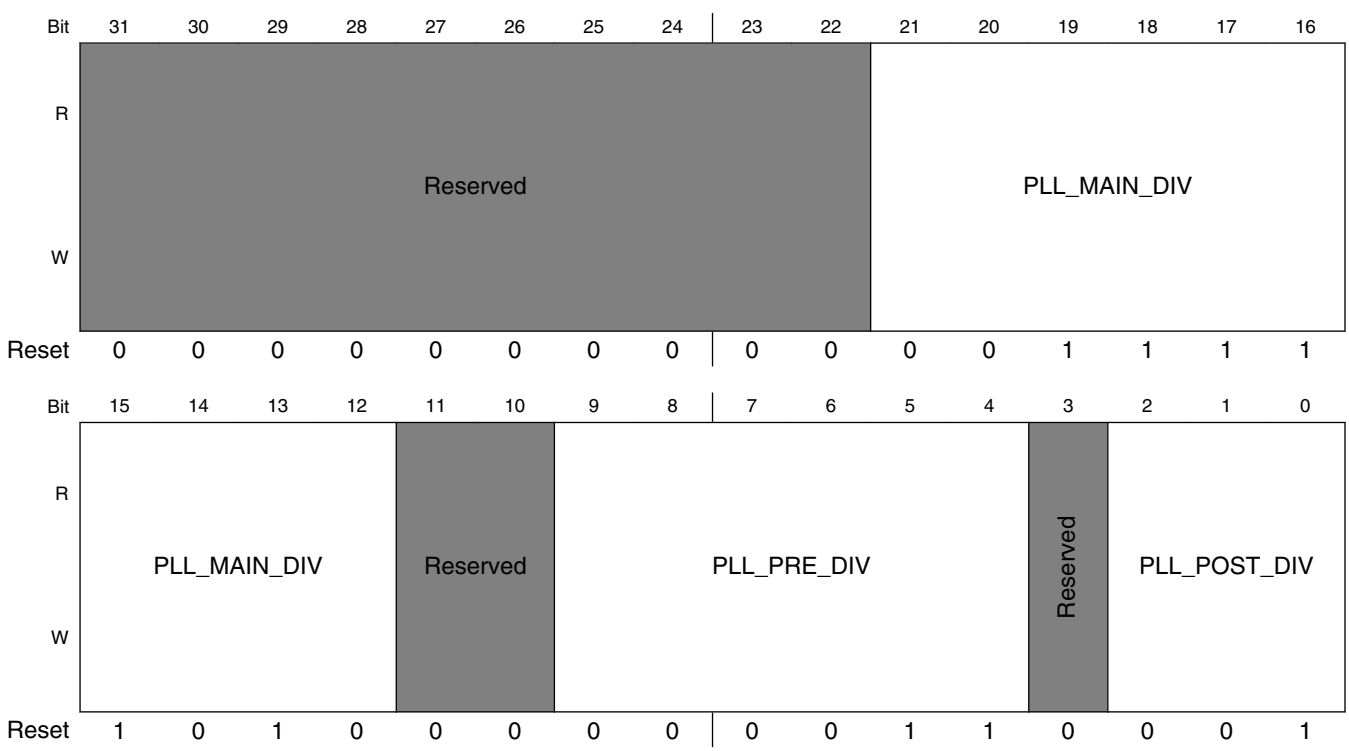
CCM_ANALOG_ARM_PLL_GEN_CTRL field descriptions (continued)

Field	Description
3–2 PAD_CLK_SEL	PAD clock select PAD_CLK is an alternate input reference clock for the PLL. The clock source selection for PAD_CLK is defined below. 00 CLKIN1 XOR CLKIN2 01 CLKIN2 10 CLKIN1 11 Reserved
PLL_REF_CLK_SEL	PLL reference clock select 00 24M_REF_CLK 01 PAD_CLK 10 Reserved 11 Reserved

5.1.8.28 ARM PLL Divide and Fraction Data Control 0 Register (CCM_ANALOG_ARM_PLL_FDIV_CTL0)

ARM PLL Divide and Fraction Data Control 0 Register

Address: 3036_0000h base + 88h offset = 3036_0088h



CCM_ANALOG_ARM_PLL_FDIV_CTL0 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–12 PLL_MAIN_DIV	Value of the main-divider
11–10 -	This field is reserved. Reserved
9–4 PLL_PRE_DIV	Value of the pre-divider
3 -	This field is reserved. Reserved
PLL_POST_DIV	Value of the post-divider

5.1.8.29 SYS PLL1 General Function Control Register (CCM_ANALOG_SYS_PLL1_GEN_CTRL)

SYS PLL1 General Function Control Register

Address: 3036_0000h base + 94h offset = 3036_0094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PLL_LOCK	Reserved	PLL_LOCK_SEL	PLL_EXT_BYPASS	PLL_DIV20_CLKE	PLL_DIV20_CLKE_OVERRIDE	PLL_DIV10_CLKE	PLL_DIV10_CLKE_OVERRIDE	PLL_DIV8_CLKE	PLL_DIV8_CLKE_OVERRIDE	PLL_DIV6_CLKE	PLL_DIV6_CLKE_OVERRIDE	PLL_DIV5_CLKE	PLL_DIV5_CLKE_OVERRIDE	PLL_DIV4_CLKE	PLL_DIV4_CLKE_OVERRIDE
W																
Reset	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_DIV3_CLKE	PLL_DIV3_CLKE_OVERRIDE	PLL_DIV2_CLKE	PLL_DIV2_CLKE_OVERRIDE	PLL_CLKE	PLL_CLKE_OVERRIDE	PLL_RST	PLL_RST_OVERRIDE	Reserved			PLL_BYPASS	PAD_CLK_SEL		PLL_REF_CLK_SEL	
Reset	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0

CCM_ANALOG_SYS_PLL1_GEN_CTRL field descriptions

Field	Description
31 PLL_LOCK	PLL lock signal
30 -	This field is reserved. Reserved
29 PLL_LOCK_SEL	PLL lock select 0 Using PLL maximum lock time 1 Using PLL output lock
28 PLL_EXT_BYPASS	PLL analog block bypass, clock output traces to PLL source
27 PLL_DIV20_CLKE	PLL clock divided by 20 output gating enable
26 PLL_DIV20_CLKE_OVERRIDE	PLL clock divided by 20 output gating enable overrided by CCM
25 PLL_DIV10_CLKE	PLL clock divided by 10 output gating enable
24 PLL_DIV10_CLKE_OVERRIDE	PLL clock divided by 10 output gating enable overrided by CCM
23 PLL_DIV8_CLKE	PLL clock divided by 8 output gating enable

Table continues on the next page...

CCM_ANALOG_SYS_PLL1_GEN_CTRL field descriptions (continued)

Field	Description
22 PLL_DIV8_ CLKE_ OVERRIDE	PLL clock divided by 8 output gating enable overrided by CCM
21 PLL_DIV6_CLKE	PLL clock divided by 6 output gating enable
20 PLL_DIV6_ CLKE_ OVERRIDE	PLL clock divided by 6 output gating enable overrided by CCM
19 PLL_DIV5_CLKE	PLL clock divided by 5 output gating enable
18 PLL_DIV5_ CLKE_ OVERRIDE	PLL clock divided by 5 output gating enable overrided by CCM
17 PLL_DIV4_CLKE	PLL clock divided by 4 output gating enable
16 PLL_DIV4_ CLKE_ OVERRIDE	PLL clock divided by 4 output gating enable overrided by CCM
15 PLL_DIV3_CLKE	PLL clock divided by 3 output gating enable
14 PLL_DIV3_ CLKE_ OVERRIDE	PLL clock divided by 3 output gating enable overrided by CCM
13 PLL_DIV2_CLKE	PLL clock divided by 2 output gating enable
12 PLL_DIV2_ CLKE_ OVERRIDE	PLL clock divided by 2 output gating enable overrided by CCM
11 PLL_CLKE	PLL output clock clock gating enable
10 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM
9 PLL_RST	PLL reset (active low)
8 PLL_RST_ OVERRIDE	PLL reset overrided by CCM
7–5 -	This field is reserved. Reserved
4 PLL_BYPASS	PLL output clock bypass

Table continues on the next page...

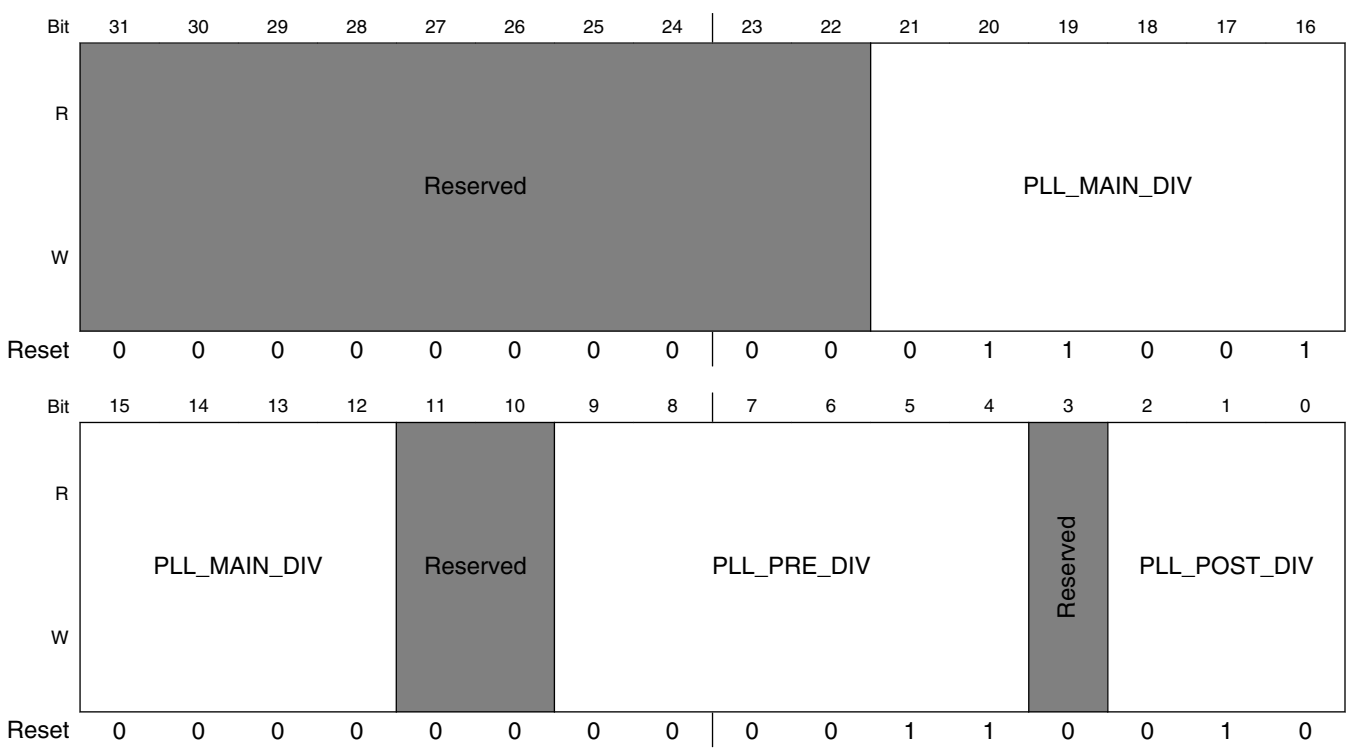
CCM_ANALOG_SYS_PLL1_GEN_CTRL field descriptions (continued)

Field	Description
3–2 PAD_CLK_SEL	PAD clock select PAD_CLK is an alternate input reference clock for the PLL. The clock source selection for PAD_CLK is defined below. 00 CLKIN1 XOR CLKIN2 01 CLKIN2 10 CLKIN1 11 Reserved
PLL_REF_CLK_SEL	PLL reference clock select 00 24M_REF_CLK 01 PAD_CLK 10 Reserved 11 Reserved

5.1.8.30 SYS PLL1 Divide and Fraction Data Control 0 Register (CCM_ANALOG_SYS_PLL1_FDIV_CTL0)

SYS PLL1 Divide and Fraction Data Control 0 Register

Address: 3036_0000h base + 98h offset = 3036_0098h



CCM_ANALOG_SYS_PLL1_FDIV_CTL0 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–12 PLL_MAIN_DIV	Value of the main-divider
11–10 -	This field is reserved. Reserved
9–4 PLL_PRE_DIV	Value of the pre-divider
3 -	This field is reserved. Reserved
PLL_POST_DIV	Value of the post-divider

5.1.8.31 PLL Monitoring Control Register (CCM_ANALOG_*n*MNIT_CTRL)

PLL Monitoring Control Register

Address: 3036_0000h base + 100h offset + (16d × i), where i=0d to 2d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved										LRD_EN	FOUT_MASK	AFC_SEL	PBIAS_CTRL	PBIAS_CTRL_EN	AFCINIT_SEL
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	FSEL	FEED_EN	Reserved					EXTAFC					AFC_EN	ICP	
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

CCM_ANALOG_*n*MNIT_CTRL field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21 LRD_EN	Monitoring pin. AFC operation mode select pin
20 FOUT_MASK	Scaler's re-initialization time control pin[3]
19 AFC_SEL	AFC Mode select
18 PBIAS_CTRL	PBIAS pull-down initial voltage control pin 0 0.50 × VDD 1 0.67 × VDD
17 PBIAS_CTRL_EN	PBIAS voltage pull-down enable pin
16 AFCINIT_SEL	AFC initial delay select pin 0 nominal delay 1 nominal delay × 2
15 -	This field is reserved. Reserved
14 FSEL	Monitoring frequency select pin 0 FEED_OUT = FREF 1 FEED_OUT = FEED
13 FEED_EN	FEED_OUT enable pin
12–8 -	This field is reserved. Reserved
7–3 EXTAFC	Monitoring pin. If AFC_ENB=1, AFC is disabled and VCO is calibrated manually by EXTAFC[4:0] for the test of VCO range
2 AFC_EN	If AFC_ENB=0, AFC is enabled and VCO is calibrated automatically.(AFC_ENB=0 and EXTAFC=0 are mandatory) If AFC_ENB=1, AFC is disabled and VCO is calibrated manually by EXTAFC[4:0] for the test of VCO range
ICP	Controls the charge-pump current

5.1.8.32 SYS PLL2 General Function Control Register (CCM_ANALOG_SYS_PLL2_GEN_CTRL)

SYS PLL2 General Function Control Register

Address: 3036_0000h base + 104h offset = 3036_0104h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PLL_LOCK	Reserved	PLL_LOCK_SEL	PLL_EXT_BYPASS	PLL_DIV20_CLKE	PLL_DIV20_CLKE_OVERRIDE	PLL_DIV10_CLKE	PLL_DIV10_CLKE_OVERRIDE	PLL_DIV8_CLKE	PLL_DIV8_CLKE_OVERRIDE	PLL_DIV6_CLKE	PLL_DIV6_CLKE_OVERRIDE	PLL_DIV5_CLKE	PLL_DIV5_CLKE_OVERRIDE	PLL_DIV4_CLKE	PLL_DIV4_CLKE_OVERRIDE
W																
Reset	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PLL_DIV3_CLKE	PLL_DIV3_CLKE_OVERRIDE	PLL_DIV2_CLKE	PLL_DIV2_CLKE_OVERRIDE	PLL_CLKE	PLL_CLKE_OVERRIDE	PLL_RST	PLL_RST_OVERRIDE	Reserved			PLL_BYPASS	PAD_CLK_SEL		PLL_REF_CLK_SEL	
W																
Reset	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0

CCM_ANALOG_SYS_PLL2_GEN_CTRL field descriptions

Field	Description
31 PLL_LOCK	PLL lock signal
30 -	This field is reserved. Reserved
29 PLL_LOCK_SEL	PLL lock select 0 Using PLL maximum lock time 1 Using PLL output lock
28 PLL_EXT_BYPASS	PLL analog block bypass, clock output traces to PLL source
27 PLL_DIV20_CLKE	PLL clock divided by 20 output gating enable
26 PLL_DIV20_CLKE_OVERRIDE	PLL clock divided by 20 output gating enable overrided by CCM
25 PLL_DIV10_CLKE	PLL clock divided by 10 output gating enable
24 PLL_DIV10_CLKE_OVERRIDE	PLL clock divided by 10 output gating enable overrided by CCM
23 PLL_DIV8_CLKE	PLL clock divided by 8 output gating enable

Table continues on the next page...

CCM_ANALOG_SYS_PLL2_GEN_CTRL field descriptions (continued)

Field	Description
22 PLL_DIV8_ CLKE_ OVERRIDE	PLL clock divided by 8 output gating enable overrided by CCM
21 PLL_DIV6_CLKE	PLL clock divided by 6 output gating enable
20 PLL_DIV6_ CLKE_ OVERRIDE	PLL clock divided by 6 output gating enable overrided by CCM
19 PLL_DIV5_CLKE	PLL clock divided by 5 output gating enable
18 PLL_DIV5_ CLKE_ OVERRIDE	PLL clock divided by 5 output gating enable overrided by CCM
17 PLL_DIV4_CLKE	PLL clock divided by 4 output gating enable
16 PLL_DIV4_ CLKE_ OVERRIDE	PLL clock divided by 4 output gating enable overrided by CCM
15 PLL_DIV3_CLKE	PLL clock divided by 3 output gating enable
14 PLL_DIV3_ CLKE_ OVERRIDE	PLL clock divided by 3 output gating enable overrided by CCM
13 PLL_DIV2_CLKE	PLL clock divided by 2 output gating enable
12 PLL_DIV2_ CLKE_ OVERRIDE	PLL clock divided by 2 output gating enable overrided by CCM
11 PLL_CLKE	PLL output clock clock gating enable
10 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM
9 PLL_RST	PLL reset (active low)
8 PLL_RST_ OVERRIDE	PLL reset overrided by CCM
7–5 -	This field is reserved. Reserved
4 PLL_BYPASS	PLL output clock bypass

Table continues on the next page...

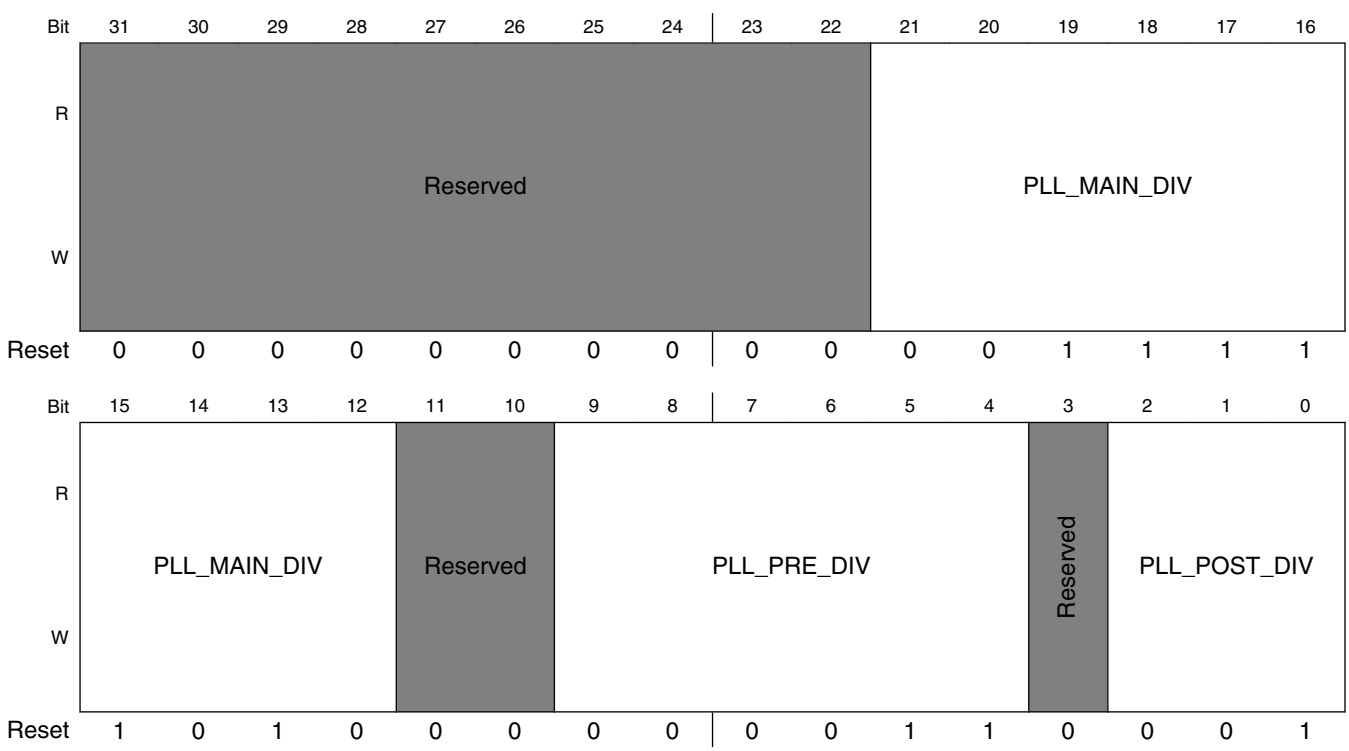
CCM_ANALOG_SYS_PLL2_GEN_CTRL field descriptions (continued)

Field	Description
3–2 PAD_CLK_SEL	PAD clock select PAD_CLK is an alternate input reference clock for the PLL. The clock source selection for PAD_CLK is defined below. 00 CLKIN1 XOR CLKIN2 01 CLKIN2 10 CLKIN1 11 Reserved
PLL_REF_CLK_SEL	PLL reference clock select 00 24M_REF_CLK 01 PAD_CLK 10 Reserved 11 Reserved

5.1.8.33 SYS PLL2 Divide and Fraction Data Control 0 Register (CCM_ANALOG_SYS_PLL2_FDIV_CTL0)

SYS PLL2 Divide and Fraction Data Control 0 Register

Address: 3036_0000h base + 108h offset = 3036_0108h



CCM_ANALOG_SYS_PLL2_FDIV_CTL0 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–12 PLL_MAIN_DIV	Value of the main-divider
11–10 -	This field is reserved. Reserved
9–4 PLL_PRE_DIV	Value of the pre-divider
3 -	This field is reserved. Reserved
PLL_POST_DIV	Value of the post-divider

5.1.8.34 PLL Lock Detector Control Register (CCM_ANALOG_*n*LOCKD_CTRL)

PLL Lock Detector Control Register

Address: 3036_0000h base + 10Ch offset + (16d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										LOCK_CON_DLY		LOCK_CON_OUT		LOCK_CON_IN	
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

CCM_ANALOG_*n*LOCKD_CTRL field descriptions

Field	Description
31–6 -	This field is reserved. Reserved
5–4 LOCK_CON_DLY	Lock detector setting of the detection resolution

Table continues on the next page...

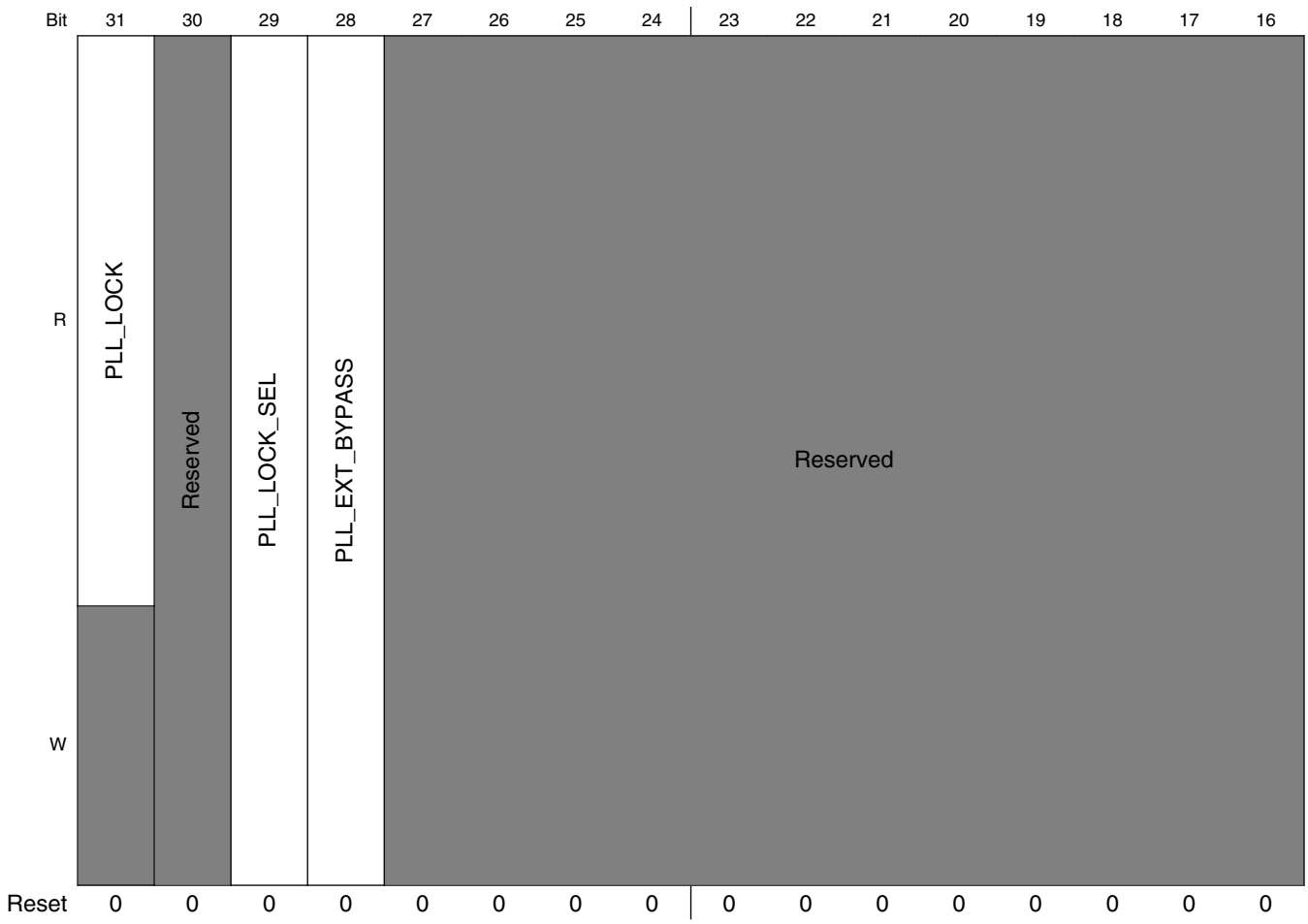
CCM_ANALOG_*n*LOCKD_CTRL field descriptions (continued)

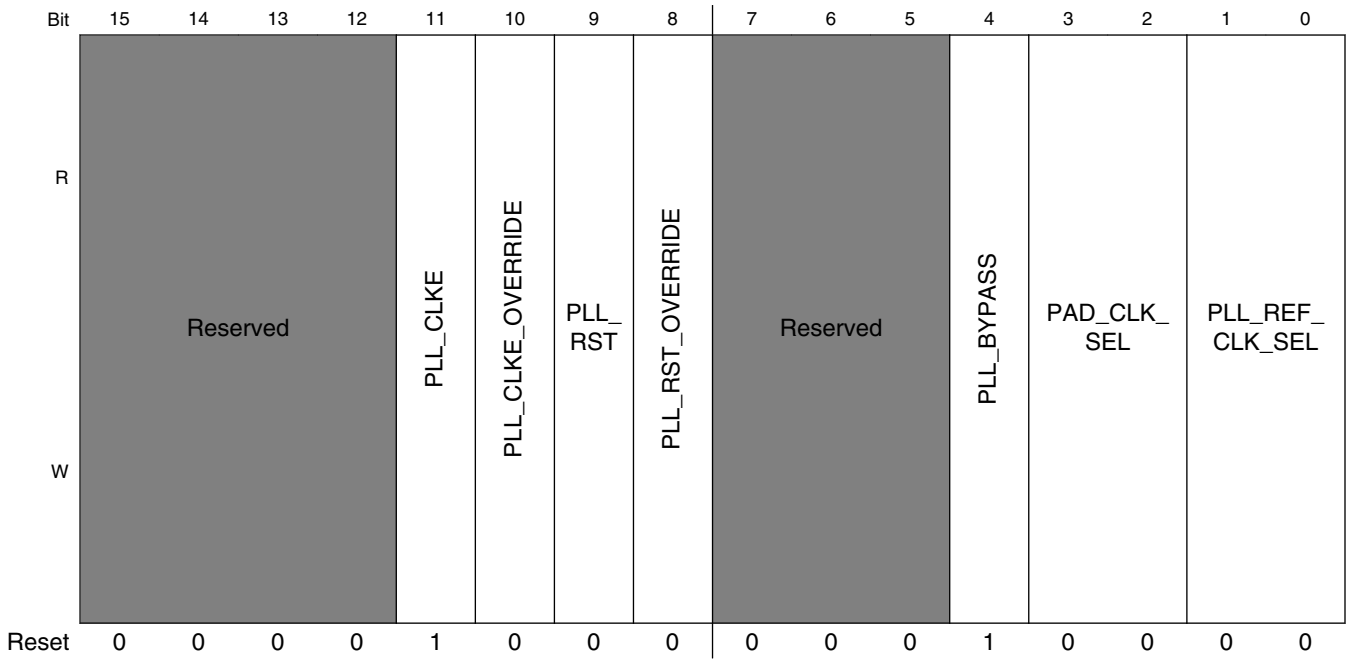
Field	Description
3–2 LOCK_CON_ OUT	Lock detector setting of the output margin
LOCK_CON_IN	Lock detector setting of the input margin

5.1.8.35 SYS PLL3 General Function Control Register (CCM_ANALOG_SYS_PLL3_GEN_CTRL)

SYS PLL3 General Function Control Register

Address: 3036_0000h base + 114h offset = 3036_0114h





CCM_ANALOG_SYS_PLL3_GEN_CTRL field descriptions

Field	Description
31 PLL_LOCK	PLL lock signal
30 -	This field is reserved. Reserved
29 PLL_LOCK_SEL	PLL lock select 0 Using PLL maximum lock time 1 Using PLL output lock
28 PLL_EXT_BYPASS	PLL analog block bypass, clock output traces to PLL source
27–12 -	This field is reserved. Reserved
11 PLL_CLKE	PLL output clock clock gating enable
10 PLL_CLKE_OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM
9 PLL_RST	PLL reset (active low)
8 PLL_RST_OVERRIDE	PLL reset overrided by CCM
7–5 -	This field is reserved. Reserved
4 PLL_BYPASS	PLL output clock bypass

Table continues on the next page...

CCM_ANALOG_SYS_PLL3_GEN_CTRL field descriptions (continued)

Field	Description
3–2 PAD_CLK_SEL	PAD clock select PAD_CLK is an alternate input reference clock for the PLL. The clock source selection for PAD_CLK is defined below. 00 CLKIN1 XOR CLKIN2 01 CLKIN2 10 CLKIN1 11 Reserved
PLL_REF_CLK_SEL	PLL reference clock select 00 24M_REF_CLK 01 PAD_CLK 10 Reserved 11 Reserved

5.1.8.36 SYS PLL3 Divide and Fraction Data Control 0 Register (CCM_ANALOG_SYS_PLL3_FDIV_CTL0)

SYS PLL3 Divide and Fraction Data Control 0 Register

Address: 3036_0000h base + 118h offset = 3036_0118h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved										PLL_MAIN_DIV					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_MAIN_DIV				Reserved		PLL_PRE_DIV						Reserved	PLL_POST_DIV		
Reset	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1

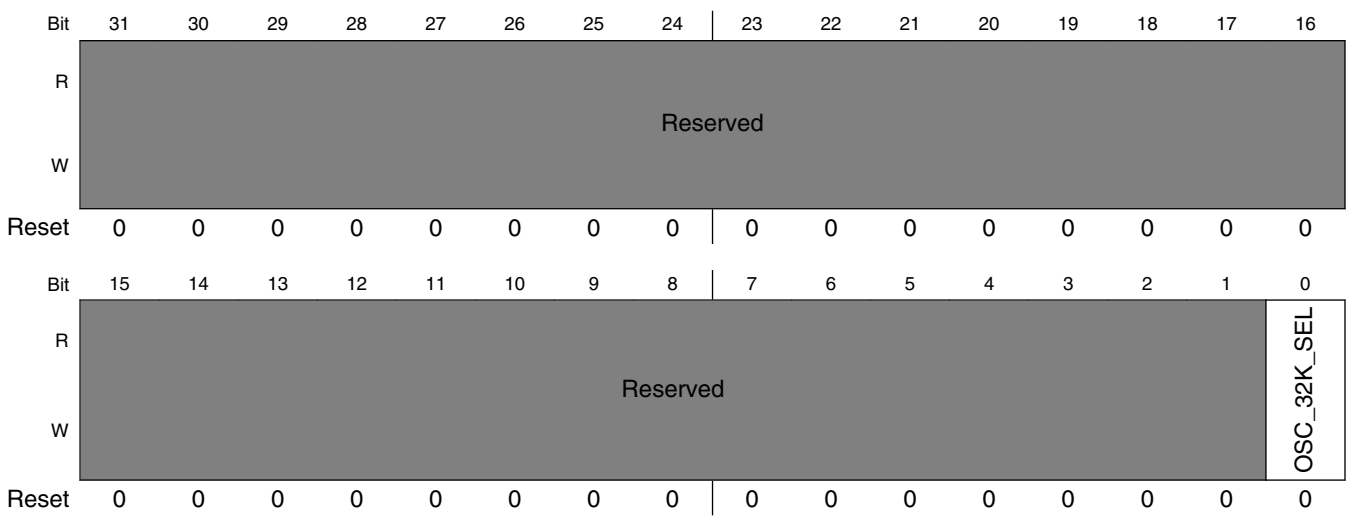
CCM_ANALOG_SYS_PLL3_FDIV_CTL0 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–12 PLL_MAIN_DIV	Value of the main-divider
11–10 -	This field is reserved. Reserved
9–4 PLL_PRE_DIV	Value of the pre-divider
3 -	This field is reserved. Reserved
PLL_POST_DIV	Value of the post-divider

5.1.8.37 Osc Misc Configuration Register (CCM_ANALOG_OSC_MISC_CFG)

Osc Misc Register

Address: 3036_0000h base + 124h offset = 3036_0124h



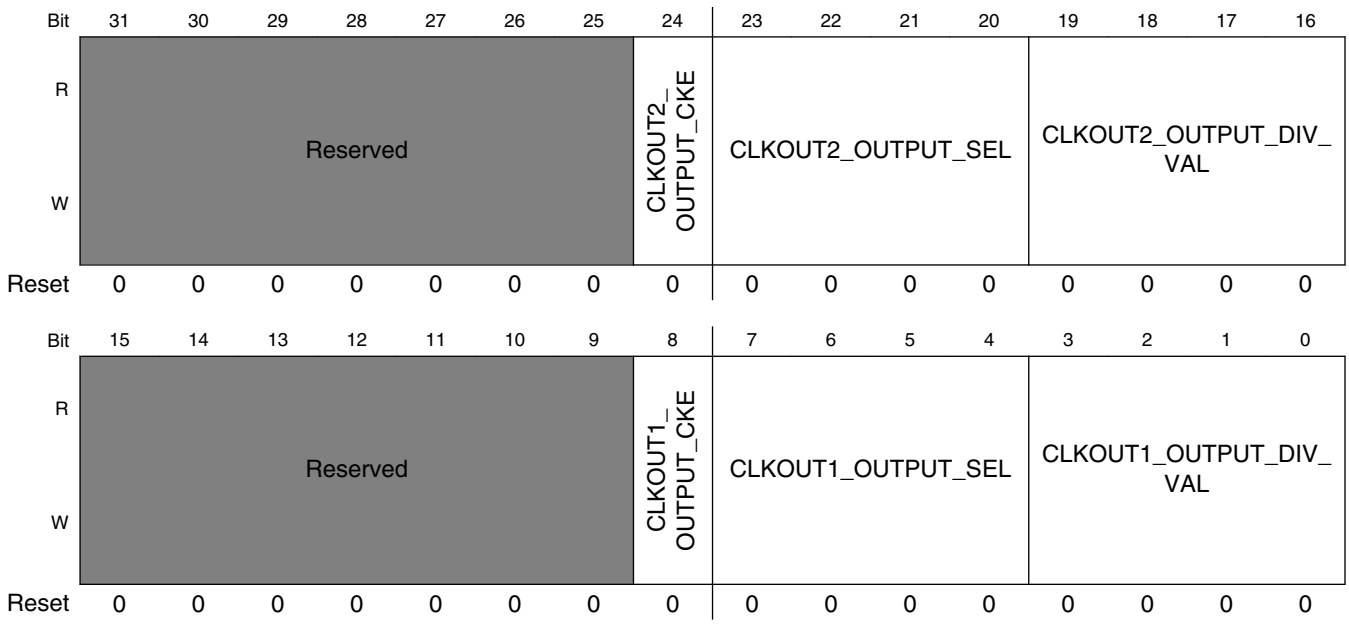
CCM_ANALOG_OSC_MISC_CFG field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 OSC_32K_SEL	32KHz OSC input select 0 Divided by 24M clock 1 32K Oscillator

5.1.8.38 PLL Clock Output for Test Enable and Select Register (CCM_ANALOG_ANAMIX_PLL_MNIT_CTL)

PLL Clock Output for Test Enable and Select Register

Address: 3036_0000h base + 128h offset = 3036_0128h



CCM_ANALOG_ANAMIX_PLL_MNIT_CTL field descriptions

Field	Description
31–25 -	This field is reserved. Reserved
24 CLKOUT2_ OUTPUT_CKE	CLKOUT2 Monitor output enable
23–20 CLKOUT2_ OUTPUT_SEL	CLKOUT2 Monitor output clock select 4'b0000 : audio_pll1_clk 4'b0001 : audio_pll2_clk 4'b0010 : video_pll1_clk 4'b0011 : hsio_pll_clk 4'b0100 : misc_mnit_clk 4'b0101 : gpu_pll_clk 4'b0110 : vpu_pll_clk 4'b0111 : arm_pll_clk 4'b1000 : system_pll1_clk

Table continues on the next page...

CCM_ANALOG_ANAMIX_PLL_MNIT_CTL field descriptions (continued)

Field	Description
	4'b1001 : system_pll2_clk 4'b1010 : system_pll3_clk 4'b1011 : CLKIN1 4'b1100 : CLKIN2 4'b1101 : sysosc_24m_clk 4'b1110 : sai_pll_clk 4'b1111 : osc_32k_clk
19–16 CLKOUT2_ OUTPUT_DIV_ VAL	CLKOUT2 output divide value
15–9 -	This field is reserved. Reserved
8 CLKOUT1_ OUTPUT_CKE	CLKOUT1 Monitor output enable
7–4 CLKOUT1_ OUTPUT_SEL	CLKOUT1 Monitor output clock select 4'b0000 : audio_pll1_clk 4'b0001 : audio_pll2_clk 4'b0010 : video_pll1_clk 4'b0011 : hsio_pll_clk 4'b0100 : misc_mnit_clk 4'b0101 : gpu_pll_clk 4'b0110 : vpu_pll_clk 4'b0111 : arm_pll_clk 4'b1000 : system_pll1_clk 4'b1001 : system_pll2_clk 4'b1010 : system_pll3_clk 4'b1011 : CLKIN1 4'b1100 : CLKIN2 4'b1101 : sysosc_24m_clk 4'b1110 : sai_pll_clk 4'b1111 : osc_32k_clk
CLKOUT1_ OUTPUT_DIV_ VAL	CLKOUT1 output divide value

5.1.8.39 DIGPROG Register (CCM_ANALOG_DIGPROG)

DIGPROG Register

Address: 3036_0000h base + 800h offset = 3036_0800h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								DIGPROG_MAJOR_UPPER								DIGPROG_MAJOR_LOWER								DIGPROG_MINOR							
W																																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0

CCM_ANALOG_DIGPROG field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 DIGPROG_ MAJOR_UPPER	Bit[7:4] is 0x8, stands for “i.MX8” Bit[3:0] is 0x2, stands for ”M”
15–8 DIGPROG_ MAJOR_LOWER	Bit[7:4] is 0x4, stands for “Quad” Bit[3:0] is 0x3, stands for “Plus”
DIGPROG_ MINOR	Bit[7:4] is the base layer revision, Bit[3:0] is the metal layer revision 0x10 stands for Tapeout 1.0

5.2 General Power Controller (GPC)

5.2.1 Overview

The General Power Controller (GPC) module controls the following functions:

- Provide low power mode control for A53 and M7 platform
- Provide Power domain management all Arm and SOC power domain
- Provide domain control mechanism based on A53 and M7 CPU domain
- Provide handshake with CCM for clock management in low power mode
- Provide handshake with SRC for power down and power up sequence
- Provide handshake with Analog for Deep Sleep Mode control

5.2.2 Features

The General Power Controller (GPC) module controls the following functions:

- Support programmable feature for WAIT/STOP/DSM low power mode
- Support time slot based power domain control
- Support flexible sleep and wakeup condition
- Support domain control for multi CPU platforms system
- All register accessed by IP bus
- Interface for the following IPs:
  - CCM – clock controller module
  - SRC – system reset controller
  - ANALOG – miscellaneous analog control

5.2.3 Block Diagram

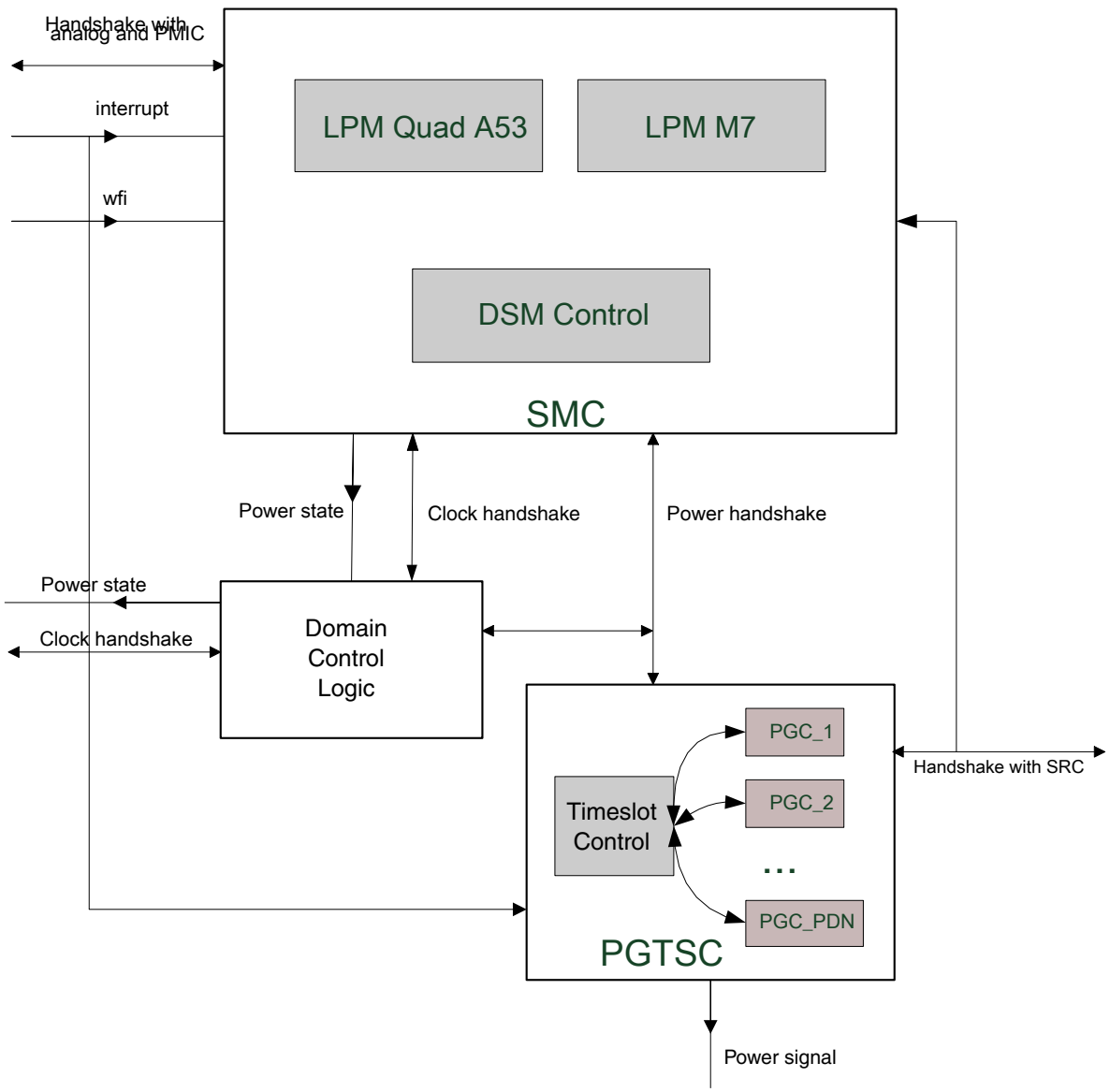


Figure 5-10. GPC Block Diagram

The GPC module contains two sub-modules: System Mode Controller (SMC) and Power Gating Time Slot Control (PGTSC):

- GPC Top: the top level GPC. It also includes the top memory map and registers, domain control information, and memory low power control.
- System Mode Controller (SMC):
  - The SMC supports two low power modes (LPM), WAIT and STOP. Each LPM corresponds to one mode for A53 platform and one mode for M7 platform.

- SMC controls the power sequence in Deep Sleep Mode (DSM)
- SMC support the power up and power down of A53 core0/core1/core2/core3 by IRQ/WFI signals without LPM triggered
- SMC can translate the LPM request for A53 and M7 platform to power up and power down request to PGTSC.
- Power Gating Time Slot Control (PGTSC):
  - The Power Gating Controller (PGC) is a power management component that controls the power-down and power-up sequencing of individual subsystems. For subsystems to be completely powered down in low power modes, a specific sequence of power control signals must be followed. The sequence timing is programmable using the PGC control registers.
  - There are 27 PGCs in the chip, all of them can be power down/up with a software trigger and all of them can be mapped to 27 timing slots and power-up and power down by request from SMC.

5.2.4 Functional Description

5.2.4.1 RUN mode

This is the normal/functional operating mode. In this mode, the CPU runs in its normal operational mode.

5.2.4.2 Low power mode

There are two CPU platforms (each of them represents a CPU domain): Quad core Cortex A53 platform and Cortex M7 platform. Each platform supports two low power modes: WAIT mode and STOP mode.

5.2.4.2.1 WAIT mode

In this low power mode:

- LPCG can be defined to be shut off or not in wait mode for each CPU domain
- PLL can be defined to be shut off or not in wait mode for each CPU domain

NOTE

The PLLs will only been closed in non-fast wake-up mode, relevant bit are

GPC_SLPCR[EN_A53_FASTWUP_WAIT_MODE] and
GPC_SLPCR[EN_M7_FASTWUP_WAIT_MODE]

- CPU clock can be defined been shut off or not in wait for each CPU platform. (GPC_LPCR_A53_BSC[CPU_CLK_ON_LPM] and GPC_LPCR_M7[CPU_CLK_ON_LPM])
- Power of different power domain can be defined be shut off or not in wait mode for each platform domain
- Some peripherals may go to wait mode along with A53 or M7 platform.

5.2.4.2.2 STOP mode

In this low power mode:

- LPCG can be defined been shut off or not in stop mode for each CPU domain
- PLL can be defined been shut off or not in stop mode for each CPU domain

NOTE

The PLLs will only been closed in non-fast wake-up mode, relevant bit are
GPC_SLPCR[EN_A53_FASTWUP_STOP_MODE] and
GPC_SLPCR[EN_M7_FASTWUP_STOP_MODE]

- CPU clock can be defined been shut off or not in stop for each CPU (GPC_LPCR_A53_BSC[CPU_CLK_ON_LPM] and GPC_LPCR_M7[CPU_CLK_ON_LPM])
- Power of different power domain can be defined be shut off or not in stop mode for each platform domain
- Some peripherals may go to stop mode along with A53 or M7 platform.

5.2.4.3 Deep Sleep Mode

The Deep Sleep Mode (DSM) is a system low power mode.

In this mode:

- On-chip OSC can be defined to be shut off or not in DSM (GPC_SLPCR[SBYOS])
- PMIC can be defined to be stand-by mode or not in DSM (GPC_SLPCR[VSTBY])
- Regulator can be defined to be BYPASS mode or not in DSM (GPC_SLPCR[RBC_EN])
- Memory can be defined to go to retention mode or not in DSM(GPC_MLPCR[MEMLP_CTL_DIS])
- A53 platform power (VDD_ARM) can be defined to be shut off or not in DSM.

NOTE

CCM configuration must make sure close all PLLs before system goes to DSM

NOTE

For the SoC to correctly power up after entering DSM, CCM_PLL_CTRLx must not be set to 0x0 or 0x3 for any domain in use.

5.2.4.4 LPM Sleep Process

CPU platform will go to WAIT/STOP under the following conditions:

- LPM registers (GPC_LPCR_A53_BSC[LPM0], GPC_LPCR_A53_BSC[LPM1], GPC_LPCR_A53_BSC2[LPM2], GPC_LPCR_A53_BSC2[LPM3], GPC_LPCR_M7[LPM]) are set to WAIT or STOP.

  NOTE

  Since A53 platform has four cores, the so each core has its own LPM register: GPC_LPCR_A53_BSC[LPM0] and GPC_LPCR_A53_BSC[LPM1], GPC_LPCR_A53_BSC2[LPM2] and GPC_LPCR_A53_BSC2[LPM3]. The unified LPM of A53 will be generated with the lower LPM of the cores.

- Asserting the WFI signal will trigger CPU sleep process. There are five WFIs that come from A53 platform: WFI_core0, WFI_core1, WFI_core2, WFI_core3, and WFI_scu. The A53 platform will go to LPM when all WFIs are asserted. If the GPC_LPCR_A53_AD[EN_C0_WFI_PDN] or GPC_LPCR_A53_AD[EN_C1_WFI_PDN] or GPC_LPCR_A53_AD[EN_C2_WFI_PDN] or GPC_LPCR_A53_AD[EN_C3_WFI_PDN] bit is set, the LPM trigger condition will be a little different. See Power control for A53 Platform for more information. Only one WFI comes from M7 platform. The M7 platform will go to LPM when WFI_M7 asserted.

  NOTE

  WFI condition can be masked by register bits GPC_LPCR_A53_BSC[MASK_n_WFI] and GPC_LPCR_M7[MASK_M7_WFI]

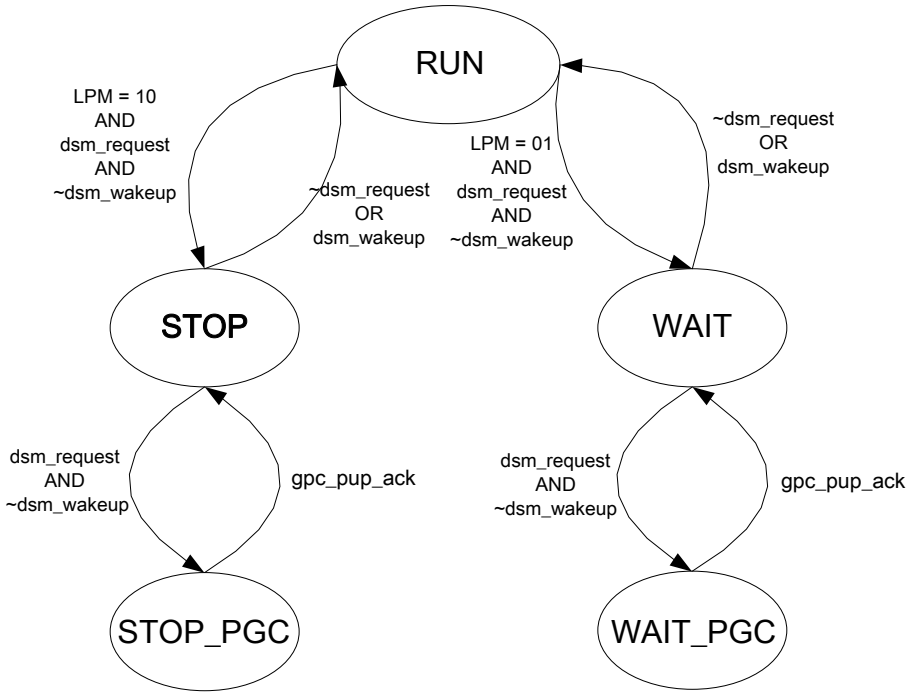


Figure 5-11. LPM transition inside CPU platform

System will go to DSM under the following conditions:

- Both A53 and M7 are STOP mode.
- Both GPC_SLPCR[EN_A53_FASTWUP_STOP_MODE] and GPC_SLPCR[EN_M7_FASTWUP_STOP_MODE] are not set.
- GPC_SLPCR[EN_DSM] is set.

NOTE

If GPC_LPCR_M7[MASK_DSM_TRIGGER] is set, the system will go to DSM when A53 goes STOP mode and GPC_SLPCR[EN_A53_FASTWUP_STOP_MODE] is not set. If GPC_LPCR_A53_BSC[MASK_DSM_TRIGGER] is set, the system will go to DSM when M7 goes to STOP mode and GPC_SLPCR[EN_M7_FASTWUP_STOP_MODE] not set. GPC_LPCR_M7[MASK_DSM_TRIGGER] and GPC_LPCR_A53_BSC[MASK_DSM_TRIGGER] cannot be set at the same time.

5.2.4.5 LPM Wake Up Process

DSM, STOP, WAIT mode will be woken up by interrupts:

- The CPU platforms share the same IRQ sources in this chip. Software can use GPC_IMRn_CORE0_A53, GPC_IMRn_CORE1_A53, GPC_IMRn_CORE2_A53, GPC_IMRn_CORE3_A53, and GPC_IMRn_M7 to separate the 128 bits IRQ sources to A53 core0, core1, core2, and core3, and M7 platform.
- The A53 core0, core1, core2, and core3 IRQ can also be from GIC source (defined by GPC_LPCR_A53_BSC[IRQ_SRC_C3], GPC_LPCR_A53_BSC[IRQ_SRC_C2], GPC_LPCR_A53_BSC[IRQ_SRC_C1], and GPC_LPCR_A53_BSC[IRQ_SRC_C0]) and if it is chosen from GIC source the GPC_IMRn_x_A53 will lose its function. See Power control for A53 Platform for more information.
- Interrupts for both A53 and M7 will cause the system wake up from DSM, A53 interrupt will wake up A53 from LPM, M7 interrupt will wake up M7 from LPM.

5.2.5 Power Gating Controller (PGC) Overview

The Power Gating Controller (PGC) is a power management component that controls the power-down and power-up sequencing of individual subsystems. For subsystems to be completely powered down in low power modes, a specific sequence of power control signals must be followed. The sequence timing is programmable using the PGC control registers.

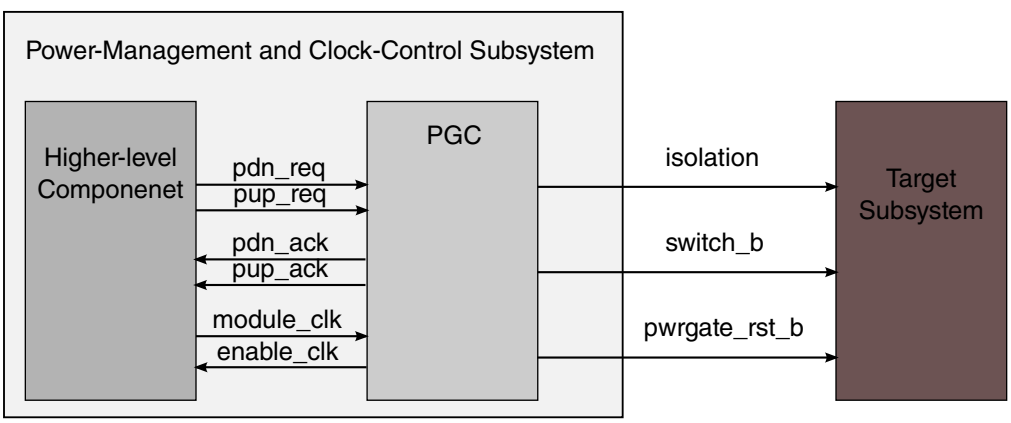


Figure 5-12. Power Gating Controller (PGC)

5.2.5.1 PGC power domains

The following table lists the PGCs in the chip and the corresponding power domain. There are three types of PGC - CPU, MIX, and PU. The power for the PUs (Power Units) can be controlled by the GPC.

PGC type	Power domain
PGC_C0	core0 of ca53
PGC_C1	core1 of ca53
PGC_C2	core0 of ca53
PGC_C3	core0 of ca53
PGC_SCU	SCU/L2 cache RAM of A53
PGC_SUPERMIX	supermix
PGC_SS_NOC_WRAPPER	ss_noc_wrapper
PGC_MIPI_PHY1	mipi_phy1
PGC_PCIE_PHY	pcie_phy
PGC_USB1_PHY	usb1_phy
PGC_USB2_PHY	usb2_phy
PGC_MLMIX	mlmix
PGC_AUDIOMIX	audiomix switchable
PGC_GPU_2D	gpu_2d
PGC_GPU_SHARE_LOGIC	gpu_share_logic
PGC_VPUMIX_SHARE_LOGIC	vpumix_share_logic
PGC_GPU3D	gpu3d
PGC_MEDIMIX	medimix default switchable
PGC_VPU_G1	vpu_g1
PGC_VPU_G2	vpu_g2
PGC_VPU_VC8K	VPU_VC8K
PGC_HDMIMIX	hdmimix
PGC_HDMI_PHY	hdmi_phy
PGC_MIPI_PHY2	mipi_phy2
PGC_HSIOMIX	hsiomix switchable
PGC_MEDIA_ISP_DWP	media_isp_dwp
PGC_DDRMIX	ddrmix

5.2.5.2 Trigger to PGC: Hardware and Software Requests

All PGCs can be power up/down by hardware or software request.

The LPM controller for each platform can generate hardware power down or power up request. All hardware requests will be mapped to “timeslot controller” before they goes to relevant PGC (see “Time slot control for PGCs” for more information).

The CPU can also generate software power up or power down request to relevant PGCs. The software trigger will not be mapped to timeslot control. If there are PGCs in software PDN/PUP sequence the request from LPM will be masked. The software trigger will also be failed if the timeslot control is in “busy” state.

All power up/down request to PGCs will be mapped to domain control module (see “Domain control for PGCs ”).

PGC_C0, PGC_C1, PGC_C2, and PGC_C3 can be triggered by its own “WFI/IRQ” without LPM trigger and time slot. See Power control for A53 Platform for more information.

5.2.5.3 Time slot control mechanism for PGCs

GPC uses a time slot controller to control the PGC sub-systems, such as the PGC in CPU0, CPU1,CPU2, CPU3, MIX, PCIE PHY, etc. We use it for below reasons when system wakes from low power mode.

1. Support flexible power down/up sequence for different sub-system
2. Sub-system can be power up in different slot to avoid large ramping up current in case they start ramping up at the same time

There are a total of 27 time slots used in the chip, one or more PGCs are used for power up or power down in each of these slots. The time slot controller will sample power up/down requests at slot0. If there are power up/down requests from SMC, it will scan from slot0 to slot26. When the scan process comes to one slot which has a defined power up or power down for one or more PGCs (defined by “SLTn_CFG”), the power up/down sequence of relevant PGCs will happen in that slot. Otherwise, the relevant slot will be skipped.

The next slot will not begin until the all PGCs finish their power up or power down process in current time slot. When all the 27 slots are finished, slot controller will jump to IDLE state and monitor new request from SMC.

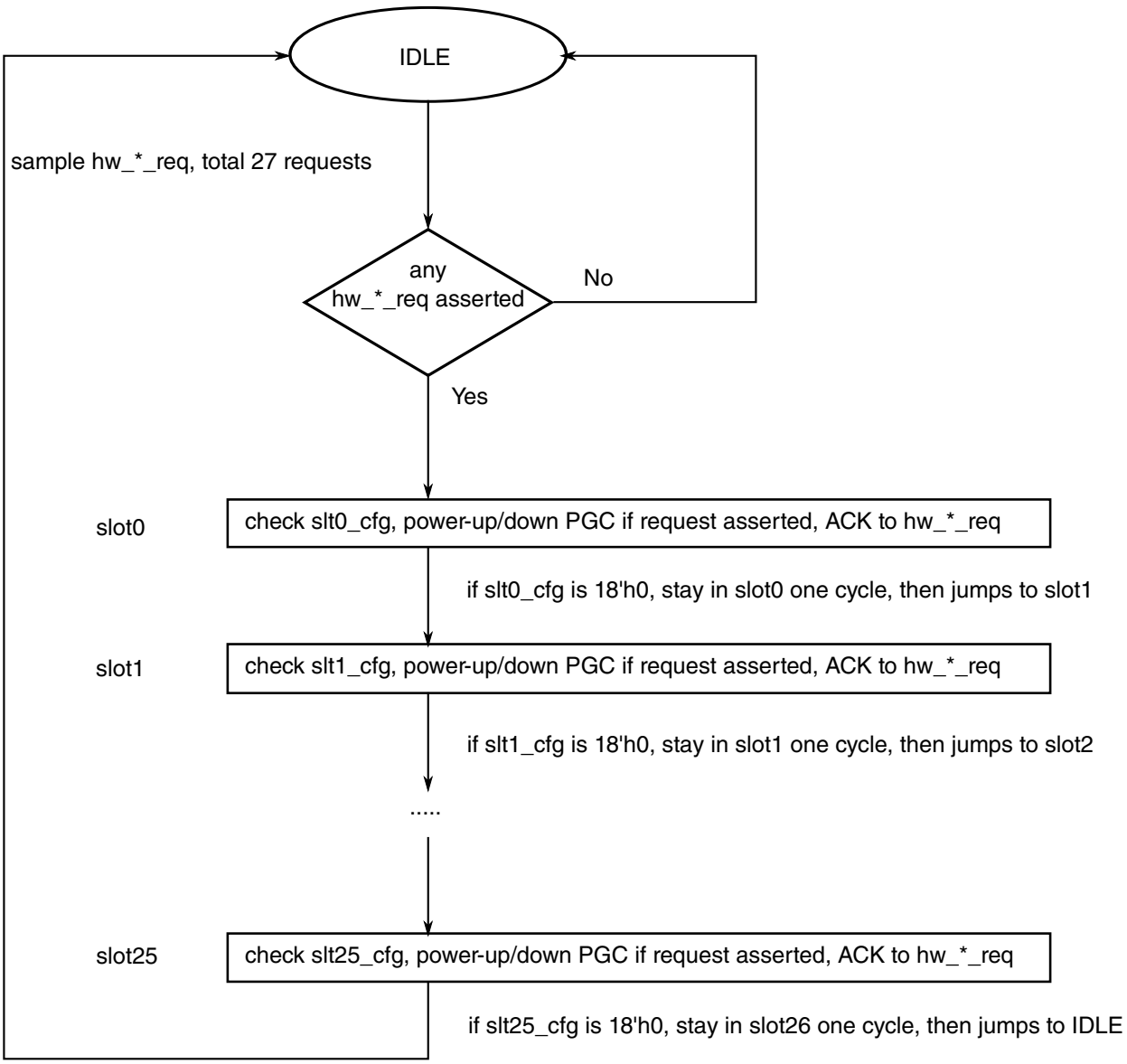


Figure 5-13. Slot controller processing flow

NOTE

PGC_SCU should be “always-on” to PGC_C0 PGC_C1, PGC_C2, and PGC_C3. This means PGC_SCU should be power up earlier than PGC_C0/PGC_C1/PGC_C2/PGC_C3 and should be power down later than PGC_C0/PGC_C1/ PGC_C2/PGC_C3 (see example code 1 and 2). If we arrange A53 Cx/A53 SCU power down/up in same slot, special setting is required (see example code 2).

NOTE

When the system enters/exists ALL_OFF or L2_RETENTION mode, PGC_MF should be power up earlier than PGC_C0/ PGC_C1/PGC_C2/PGC_C3/PGC_SCU. We can arrange MIX PGC power up in earlier slot than A53 Cx/SCU power up slot (See example code 1 and 2).

NOTE

SCU power down should not be enabled in wait mode.

5.2.5.4 Handshake between LPM controller and time slot controller

The figure below shows an example of A53 “into” LPM sequence. We want to power up A53 SCU and A53 core0 in time slot0 /slot1/slot2 respectively. Request from low power mode controller will be mapped to relevant PGCs according to the rules listed above. We choose acknowledge from PGC_core0 as the acknowledge for A53 LPM power down request (relevant register bits are defined in “PGC_ACK_SEL_A53”). The A53 LPM will regard the three PGCs as a virtual big PGC and it will cancel all power up request when it receive the acknowledge signals from PGC_core0.

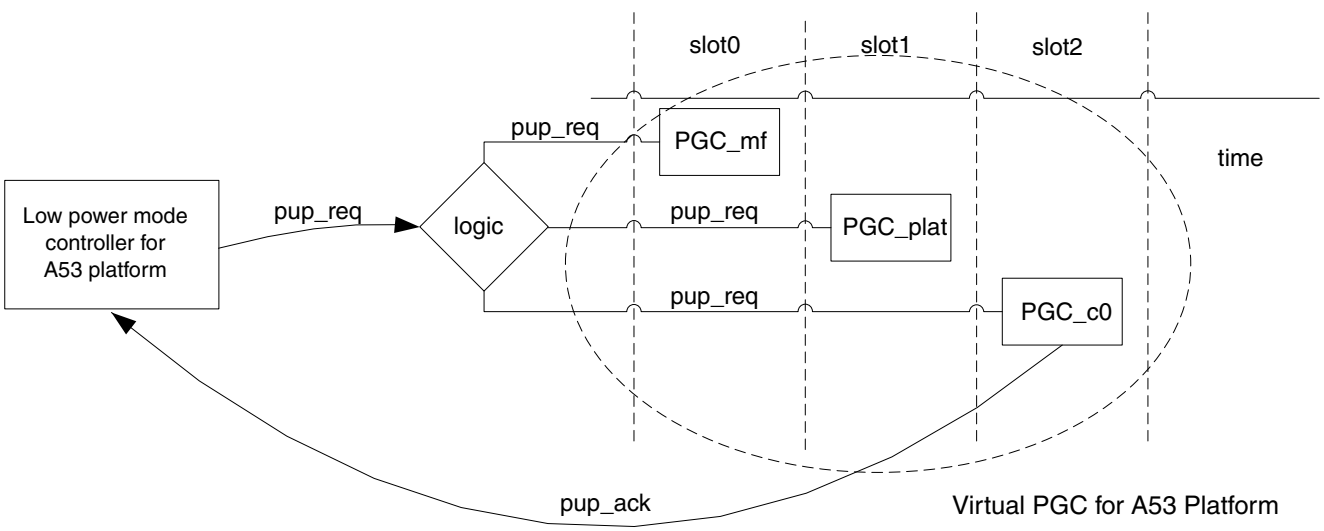


Figure 5-14. A53 into LPM sequence

NOTE

If a PGC is mapped to two CPU domain (refer to “Domain control for PGCs ”for more information), it cannot be selected as the power down acknowledge for both of the CPU platform. “PGC_ACK_SEL_A53”, "PGC_ACK_SEL_A53_PU", and “PGC_ACK_SEL_M7” are should be chosen for the last PGC in power up or power down sequence in the time slot. If there is

no PGC be power up/power down with LPM sequence, the “dummy” acknowledge should be selected. Only one PGC should be selected for power down or power up acknowledge for one CPU platform.

5.2.6 Power control for A53 Platform

5.2.6.1 A53 Platform power domains and power modes

There are six power domains inside SEC dual core Cortex A53 platform: Core0, Core1, Core2, Core3, SCU, and L2 RAM. There are seven power states in A53 platform:

Power State	PDCPU0	PDCPU1	PDCPU2	PDCPU3	PDPLAT	PDL2	VDD_ARM
ALL_ON	ON	ON	ON	ON	ON	ON	ON
THREE_CPU_ON	3 CPUs are ON, 1 CPU is OFF				ON	ON	ON
TWO_CPU_ON	2 CPUs are ON, 2 CPUs are OFF				ON	ON	ON
ONE_CPU_ON	1 CPU is ON, 3 CPUs are OFF				ON	ON	ON
ALL_CPU_OFF	OFF	OFF	OFF	OFF	ON	ON	ON
L2_RETENTION	OFF	OFF	OFF	OFF	OFF	RET	ON
VDD_ARM_OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON

NOTE

In all seven power states, “ALL_ON”, “THREE_CPU_ON", "TWO_CPU_ON", and "ONE_CPU_ON" can exist in all RUN, WAIT or STOP mode of A53 platform.
"ALL_CPU_OFF","L2_RETENTION" and "VDD_ARM_OFF" can only exist in WAIT or STOP mode of A53 platform.

5.2.6.2 Power down process for the A53 Platform

5.2.6.2.1 Power down of Core0, Core1, Core2, and Core3 in the A53 Platform

The power of core0, core1, core2, and core3 can be shut off along with the LPM process, as show in the following figure:

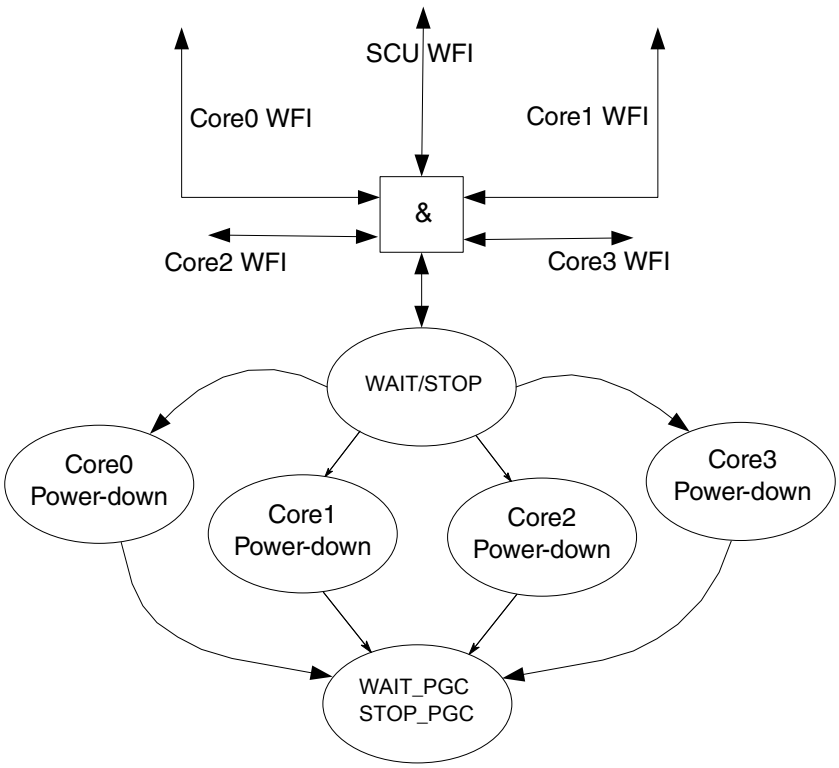


Figure 5-15. Power down of Core0, Core1, Core2, and Core3

WFIs from A53 platform will trigger the A53 platform LPM and the power of core0, core1, core2, or core3 will be shut off when “lpcr_a53_ad.en_c0_pdn”, “lpcr_a53_ad.en_c1_pdn”, “lpcr_a53_ad.en_c2_pdn”, or “lpcr_a53_ad.en_c3_pdn” enabled in this process. This mode should be used when core0 is used as the leading core of A53 platform.

The power of core0, core1, core2, and core3 can also be shut off in RUN mode: in this mode “LPCR_A53_AD.en_c0_wfi_pdn”, “LPCR_A53_AD.en_c1_wfi_pdn”, “LPCR_A53_AD.en_c2_wfi_pdn”, and “LPCR_A53_AD.en_c3_wfi_pdn” should be set and the condition to trigger A53 LPM will be some different:

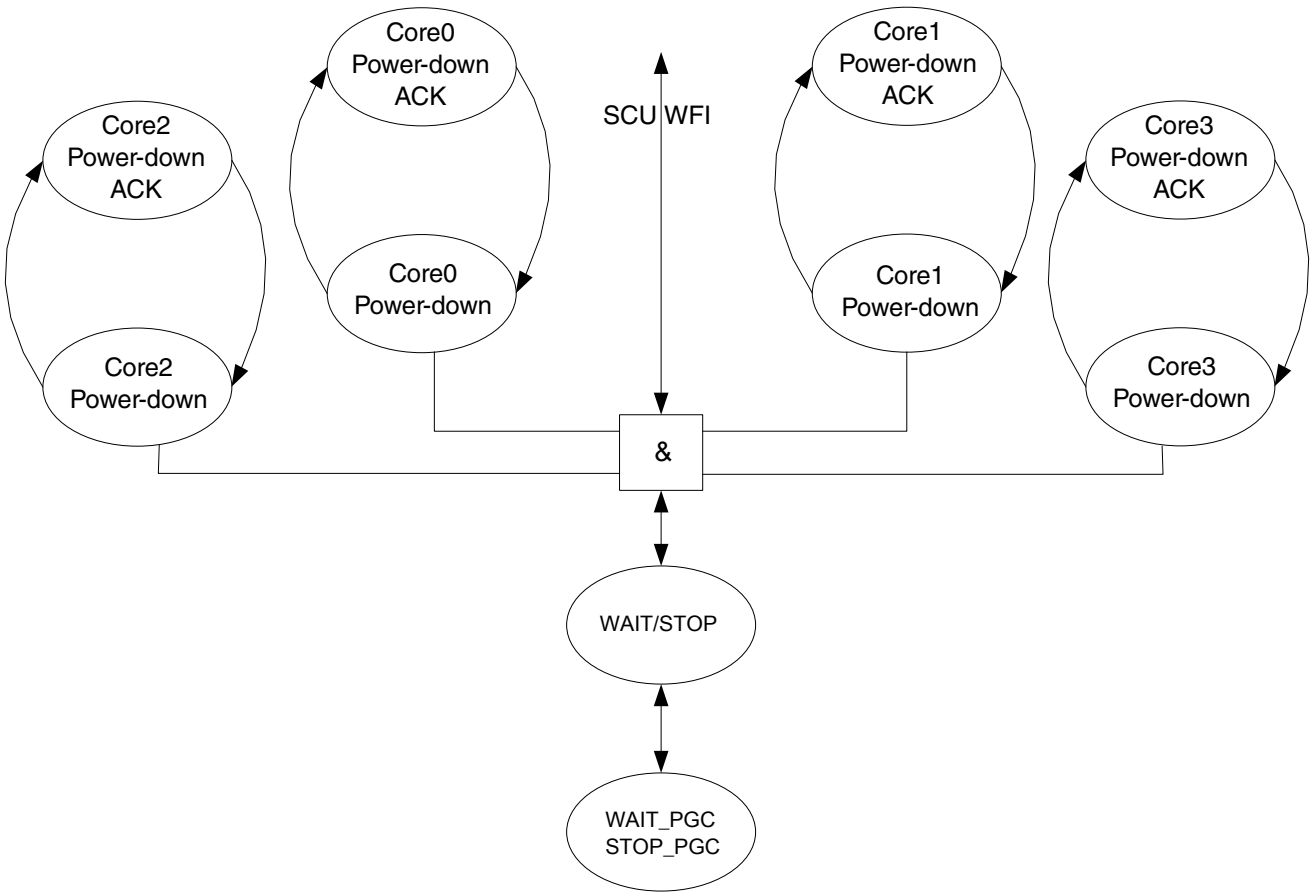


Figure 5-16. Power down of Core0, Core1, Core2, and Core3 in RUN mode

In this mode, core0/core1/core2/core3 power down process should not be disturbed by IRQs.

5.2.6.2.2 Power down of SCU and L2 Cache RAM

Power domain SCU and L2 cache RAM is controlled by one PGC – “PGC_PLAT” and they can only be power down in LPM process (when “LPCR_A53_AD.en_plat_pdn” is set). There is another bit “LPCR_A53_AD.l2pge” which will decide if L2 cache RAM need to be in retention mode when SCU domain is power down.

5.2.6.3 Power up process for the A53 Platform

- The core0, core1, core2, and core3 can be powered up with the exit of A53 LPM. The relevant bits are “LPCR_A53_AD.en_c0_pup”, “LPCR_A53_AD.en_c1_pup”, “LPCR_A53_AD.en_c2_pup”, and “LPCR_A53_AD.en_c3_pup”.
- The core0, core1, core2, and core3 can also be powered up by interrupt signal in RUN mode.

- The relevant bit are “LPCR_A53_AD.en_c0_irq_pup”, “LPCR_A53_AD.en_c1_irq_pup”, “LPCR_A53_AD.en_c2_irq_pup”, and “LPCR_A53_AD.en_c3_irq_pup”.
- The interrupt signal can be chosen from GIC or directly from IRQ with mask in GPCv2.
- The relevant select bits are “LPCR_A53_BSC.irq_src_c0”, “LPCR_A53_BSC.irq_src_c1”, “LPCR_A53_BSC.irq_src_c2”, “LPCR_A53_BSC.irq_src_c3”, and “LPCR_A53_BSC.irq_src_a53_wup”. (LPCR_A53_BSC[30],LPCR_A53_BSC[23:22],LPCR_A53_BSC.)

{LPCR_A53_BSC[30],LPCR_A53_BSC[23:22],LPCR_A53_BSC[29:28]}	Usage	Restriction
5'b00000	Use IRQ trigger A53 LPM and use IRQ to power up core0 to core3	None
5'b01111	Use GIC trigger A53 LPM and GIC to power up core0 to core3	SCU cannot power down in LPM, CPU clock cannot stop in LPM
5'b11111	Use IRQ trigger A53 LPM and GIC to power up core0 to core3	SCU cannot power down in LPM

As show in the table above, core0/core1/core2/core3 can only be power up by its own interrupt in RUN mode of A53 platform.

There are three combination of {LPCR_A53_BSC[30],LPCR_A53_BSC[23:22],LPCR_A53_BSC[29:28]}:

1. In the first case, “IMRn_CORE0_A53, IMRn_CORE1_A53, IMRn_CORE2_A53, IMRn_CORE3_A53“ are used to separate the 128 bits interrupts for core0, core1, core2, and core3 of A53 platform and also used as the interrupt mask for A53 LPM.
2. In the second case, “IMRn_CORE0_A53, IMRn_CORE1_A53, IMRn_CORE2_A53, IMRn_CORE3_A53” are not used, GIC setting are used to separate interrupts for core0, core1, core2, and core3 of A53 platform and also used as the interrupt mask for A53 LPM.
3. In the third case, “IMRn_CORE0_A53, IMRn_CORE1_A53, IMRn_CORE2_A53, IMRn_CORE3_A53” is used as the mask for interrupt for A53 LPM, GIC setting are used to separate interrupts for core0, core1, core2, and core3.

5.2.7 Power control for the M7 Platform

M7 LPM is same as A53 LPM. The M7 platform does not have its own power domain. There is a virtual PGC reserved for the M7. The virtual PGC will do nothing else except generating an acknowledge signal and this signal can be chosen as the acknowledge signal for the M7 platform.

5.2.8 Domain control for PGCs

The following rules are used for PGC power up/down with domain mapping control:

1. For PGCs inside CPU platform (since M7 doesn't have its own power domain, only PGCs in A53 platform are referred): for hardware trigger (including both power up and power down) only the LPM request from its own platform will take effect; for software trigger (including both power up and power down) (the relevant register bit are "CPU_PGC_SW_PUP_REQ" and "CPU_PGC_SW_PDN_REQ"), only the software running in its own platform will take effect.
2. For PGCs outside CPU platform(MIX and PU PGCs), register bits "PGC_CPU_MAPPING" will map MIX and PU PGCs to A53 or M7 CPU domain:
   - One PGC can be mapped to one or both of A53 and M7 domain;
   - For hardware power up request, if a PGC is mapped to any CPU domain, the PGC will be powered up when the corresponding CPU platform sends out its power up request.
   - For software power up, if a PGC is mapped any CPU platform, the PGC can be powered up by software running in corresponding CPU.(the relevant register bits are "MIX_PGC_SW_PUP_REQ" and "PU_PGC_SW_PUP_REQ")
   - For hardware power down, if a power domain is mapped to only one CPU domain, the relevant PGC can be powered down when the corresponding CPU platform sends out its power down request; If a power domain is mapped to both two CPU domain, when one CPU platform sends out its hardware power down request, the relevant PGC will power down with any of the following condition satisfied: the other CPU already in LPM; the other CPU want to power down relevant PGC (the relevant register are "A53_MIX_PDN_FLG, M7_MIX_PDN_FLG, A53_PU_PDN_FLG, M7_PU_PDN_FLG")
   - For software power down, if a power domain is mapped to only one CPU domain, the PGC can be powered down by software running in corresponding CPU. (the relevant register bits are "MIX_PGC_SW_PUP_REQ" and "PU_PGC_SW_PUP_REQ"). If a power domain is mapped to both two CPU domain, when one CPU platform sends out its software power down request, the relevant PGC will power down with any of the following condition satisfied: the other CPU already in LPM; the other CPU want to power down relevant PGC (the relevant register are "A53_MIX_PDN_FLG, M7_MIX_PDN_FLG, A53_PU_PDN_FLG, M7_PU_PDN_FLG")
   - The access of "PGC_CPU_MAPPING" is controlled by domain information from RDC

5.2.9 Example Code

Below are code examples for entering specific power scenarios

5.2.9.1 Example Code 1

```
//Arm enters into ALL_OFF(STOP) mode and enable DSM :
//after "wfi", MIX/C0/C1/C2/C3 power down in SLOT0, SCU power down in SLOT1 when
//after "GPT1_INT" arrived, MIX power up in SLOT2, SCU power up in SLOT3, C0 power up in
SLOT4
//IMRx_CORE0_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x30, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x34, 0xFF7FFFFF); //[23] : GPT1 used as wakeup source
reg32_write(GPC_IPS_BASE_ADDR + 0x38, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x3C, 0xFFFFFFFF);
//IMRx_CORE1_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x40, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x44, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x48, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x4C, 0xFFFFFFFF);
//IMRx_CORE2_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x1C0, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1C4, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1C8, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1CC, 0xFFFFFFFF);
//IMRx_CORE3_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x1D0, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1D4, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1D8, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1DC, 0xFFFFFFFF);
//LPCR_A53_BSC
reg32_write(GPC_IPS_BASE_ADDR, 0x0000000A) ;
//[30],[23:22],[29:28] : A53_CO/A53_C1/A53_C2/A53_C3/LPM wakeup from external INT
//[14] : CLOCK OFF during STOP mode
//[3:0] : STOP mode
//LPCR_A53_BSC2
reg32_write(GPC_IPS_BASE_ADDR + 0x108, 0x0000000A) ;
//[3:0] : STOP mode
//LPCR_A53_AD
reg32_write(GPC_IPS_BASE_ADDR + 0x4, 0x0A0A0A1A) ;
//[16] : 1(ALL_OFF mode); 0(L2 retention mode)
//[27]/[25]/[11]/[9]: A53_C0/A53_C1/A53_C2/A53_C3 power up with A53 LPM PUP REQ
//[4] : A53_SCU power down with A53 LPM PDN REQ
//[19]/[17]/[3]/[1] : A53_C0/A53_C1/A53_C2/A53_C3 powr down with A53 LPM PDN REQ
//LPCR_M4
reg32_write(GPC_IPS_BASE_ADDR + 0x8, 0x80000000) ;
//[31] : DSM ingore to check M4 low power state
//[1:0] : M4 LPM run mode
//SLPCR
reg32_write(GPC_IPS_BASE_ADDR + 0x14,0xe000ffA7) ;
//[31] : enable DSM
//[30] : eneable regulator bypass
//[5:3] : wait 64 ckil clock cycles
//[2] : enable PMIC standby
//[1] : enable OSC power down
//[0] : bypass PMIC ready handshake
//PGC_ACK_SEL_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x24,0x00010004) ;
//[2] : A53_SCU PGC as LPM power down ack
//[16] : A53_C0 PGC as LPM power up ack
//SLT_CFG0
```

```
reg32_write(GPC_IPS_BASE_ADDR + 0xB0,0x00000055) ;
//[6]/[4]/[2]/[0] : A53_C0/A53_C1/ A53_C2/A53_C3 power down in SLOT0
//SLT_CFG0_PU
reg32_write(GPC_IPS_BASE_ADDR + 0x200,0x0000001) ;
//[0] : power down in SLOT0
//SLT_CFG1
reg32_write(GPC_IPS_BASE_ADDR + 0xB4,0x00000100) ;
//[8] : A53_SCU power down in SLOT1
//SLT_CFG2_PU
reg32_write(GPC_IPS_BASE_ADDR + 0x208,0x00000002) ;
//[1] : power up in SLOT2
//SLT_CFG3
reg32_write(GPC_IPS_BASE_ADDR + 0xBC,0x00000200) ;
//[9] : A53_SCU power up in SLOT3
//SLT_CFG4
reg32_write(GPC_IPS_BASE_ADDR + 0xC0,0x00000002) ;
//[1] : A53_C0 power up in SLOT4, A53_C1/ A53_C2/A53_C3 not power up
//PGC_CPU_MAPPING
reg32_write(GPC_IPS_BASE_ADDR + 0xEC,0x00000001) ;
//[0] : MIX PGC mapping to A53 LPM
//A53 PGC
reg32_write(GPC_IPS_BASE_ADDR +0x800, reg32_read(GPC_IPS_BASE_ADDR +0x800) | 0x00000001) ;
// enable A53_C0 PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x840, reg32_read(GPC_IPS_BASE_ADDR +0x840) | 0x00000001) ;
// enable A53_C1 PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x880, reg32_read(GPC_IPS_BASE_ADDR +0x880) | 0x00000001) ;
// enable A53_C2 PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x8C0, reg32_read(GPC_IPS_BASE_ADDR +0x8C0) | 0x00000001) ;
// enable A53_C3 PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x900, reg32_read(GPC_IPS_BASE_ADDR +0x900) | 0x00000001) ;
// enable A53_SCU PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x910, (0x59 lt:lt: 10) | 0x5B | (0x51 lt:lt: 20) ) ;
// change nL2retn/mempwr/dftram to meet SCU power up timing
// PGC
reg32_write(GPC_IPS_BASE_ADDR +0xA00, reg32_read(GPC_IPS_BASE_ADDR +0xA00) | 0x00000001) ;
// enable MIX PGC power down
```

5.2.9.2 Example Code 2

```
//Arm enters into L2_RETENTION(STOP) mode and enable DSM :
//after "wfi", MIX/C0/C1/C2/C3/SCU power down in SLOT0
//after "GPT1_INT" arrived, MIX power up in SLOT1, SCU/C0 power up in SLOT2
//IMRx_CORE0_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x30, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x34, 0xFF7FFFFF);
//[23] : GPT1 used as wakeup source
reg32_write(GPC_IPS_BASE_ADDR + 0x38, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x3C, 0xFFFFFFFF);
//IMRx_CORE1_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x40, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x44, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x48, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x4C, 0xFFFFFFFF);
//IMRx_CORE2_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x1C0, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1C4, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1C8, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1CC, 0xFFFFFFFF);
//IMRx_CORE3_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x1D0, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1D4, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1D8, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1DC, 0xFFFFFFFF);
```

```
//LPCR_A53_BSC
reg32_write(GPC_IPS_BASE_ADDR, 0x0000000A) ;
//[30],[23:22],[29:28] : A53_C0/A53_C1/A53_C2/A53_C3/LPM wakeup from external INT
//[14] : CLOCK OFF during STOP mode
//[3:0] : STOP mode
//LPCR_A53_BSC2
reg32_write(GPC_IPS_BASE_ADDR + 0x108, 0x0000000A) ;
//[3:0] : STOP mode
//LPCR_A53_AD
reg32_write(GPC_IPS_BASE_ADDR + 0x4, 0x0A0A0A1A) ;
//[16] : 1(ALL_OFF mode); 0(L2 retention mode)
//[27]/[25]/[11]/[9]: A53_C0/A53_C1/A53_C2/A53_C3 power up with A53 LPM PUP REQ
//[4] : A53_SCU power down with A53 LPM PDN REQ
//[19]/[17]/[3]/[1] : A53_C0/A53_C1/A53_C2/A53_C3 powr down with A53 LPM PDN REQ
//LPCR_M4
reg32_write(GPC_IPS_BASE_ADDR + 0x8, 0x80000000) ;
//[31] : DSM ingore to check M4 low power state
//[1:0] : M4 LPM run mode
//SLPCR
reg32_write(GPC_IPS_BASE_ADDR + 0x14,0xe000ffA7) ;
//[31] : enable DSM
//[30] : eneable regulator bypass
//[5:3] : wait 64 ckil clock cycles
//[2] : enable PMIC standby
//[1] : enable OSC power down
//[0] : bypass PMIC ready handshake
//PGC_ACK_SEL_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x24,0x00010004) ;
//[2] : A53_SCU PGC as LPM power down ack
//[16] : A53_C0 PGC as LPM power up ack
//SLT_CFG0
reg32_write(GPC_IPS_BASE_ADDR + 0xB0,0x00000155) ;
//[6]/[4]/[2]/[0] : A53_C0/A53_C1/ A53_C2/A53_C3 power down in SLOT0
//[8] : A53_SCU power down in SLOT0
//SLT_CFG0_PU
reg32_write(GPC_IPS_BASE_ADDR + 0x200,0x0000001) ;
//[0] : power down in SLOT0
//A53_Cx/SCU are power down in same slot. Special setting is required( see below PGC setting
#A )
//SLT_CFG1_PU
reg32_write(GPC_IPS_BASE_ADDR + 0x204,0x00000002) ;
//[1] : power up in SLOT1
//SLT_CFG2
reg32_write(GPC_IPS_BASE_ADDR + 0xB8,0x00000202) ;
//[9] : A53_SCU power up in SLOT1
//[1] : A53_C0 power up in SLOT1, A53_C1/ A53_C2/ A53_C3 not power up
//A53_Cx/SCU are power up in same slot. Special setting is required( see below PGC setting
#B )
//PGC_CPU_MAPPING
reg32_write(GPC_IPS_BASE_ADDR + 0xEC,0x00000001) ;
//[0] : MIX PGC mapping to A53 LPM
//Special PGC setting #A for C0/C1/C2/C3/SCU power down in same slot(SCU should be always ON
comparing to C0/C1/C2/C3)
reg32_write(GPC_IPS_BASE_ADDR +0x808, (reg32_read(GPC_IPS_BASE_ADDR +0x808) & 0xFFFFC0C0) |
0x0801 ) ;
// set C0.ISO2SW = 8 ; C0.ISO = 1;
reg32_write(GPC_IPS_BASE_ADDR +0x848, (reg32_read(GPC_IPS_BASE_ADDR +0x848) & 0xFFFFC0C0) |
0x0801 ) ;
// set C1.ISO2SW = 8 ; C1.ISO = 1;
reg32_write(GPC_IPS_BASE_ADDR +0x888, (reg32_read(GPC_IPS_BASE_ADDR +0x888) & 0xFFFFC0C0) |
0x0801 ) ;
// set C2.ISO2SW = 8 ; C2.ISO = 1;
reg32_write(GPC_IPS_BASE_ADDR +0x8C8, (reg32_read(GPC_IPS_BASE_ADDR +0x8C8) & 0xFFFFC0C0) |
0x0801 ) ;
// set C3.ISO2SW = 8 ; C3.ISO = 1;
reg32_write(GPC_IPS_BASE_ADDR +0x908, (reg32_read(GPC_IPS_BASE_ADDR +0x908) & 0xFFFFC0C0) |
0x1001 ) ;
// set SCU.ISO2SW = 16 ; SCU.ISO = 1;
//Special PGC setting #B for A53_Cx/SCU power up in same slot(SCU should be always ON
comparing to C0/C1/C2/C3)
```

```
reg32_write(GPC_IPS_BASE_ADDR +0x804, (reg32_read(GPC_IPS_BASE_ADDR +0x804) & 0xFF800040) |
0x11 | (0x20 lt;lt; 7) ) ;
// set C0.SW = 0x11 ; C0.SW2ISO = 0x20 ;
reg32_write(GPC_IPS_BASE_ADDR +0x844, (reg32_read(GPC_IPS_BASE_ADDR +0x844) & 0xFF800040) |
0x11 | (0x20 lt;lt; 7) ) ;
// set C1.SW = 0x11 ; C1.SW2ISO = 0x20 ;
reg32_write(GPC_IPS_BASE_ADDR +0x884, (reg32_read(GPC_IPS_BASE_ADDR +0x884) & 0xFF800040) |
0x11 | (0x20 lt;lt; 7) ) ;
// set C2.SW = 0x11 ; C2.SW2ISO = 0x20 ;
reg32_write(GPC_IPS_BASE_ADDR +0x8C4, (reg32_read(GPC_IPS_BASE_ADDR +0x8C4) & 0xFF800040) |
0x11 | (0x20 lt;lt; 7) ) ;
// set C3.SW = 0x11 ; C3.SW2ISO = 0x20 ;
reg32_write(GPC_IPS_BASE_ADDR +0x904, (reg32_read(GPC_IPS_BASE_ADDR +0x904) & 0xFF800040) |
0x1 | (0x0f lt;lt; 7) ) ;
// set SCU.SW = 0x1 ; SCU.SW2ISO = 0x0f ;
//A53 PGC
reg32_write(GPC_IPS_BASE_ADDR +0x800, reg32_read(GPC_IPS_BASE_ADDR +0x800) | 0x00000001) ;
// enable A53_C0 PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x840, reg32_read(GPC_IPS_BASE_ADDR +0x840) | 0x00000001) ;
// enable A53_C1 PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x880, reg32_read(GPC_IPS_BASE_ADDR +0x880) | 0x00000001) ;
// enable A53_C2 PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x8C0, reg32_read(GPC_IPS_BASE_ADDR +0x8C0) | 0x00000001) ;
// enable A53_C3 PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x900, reg32_read(GPC_IPS_BASE_ADDR +0x900) | 0x00000001) ;
// enable A53_SCU PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x910, (0x59 lt;lt; 10) | 0x5B | (0x51 lt;lt; 20) ) ;
// change nL2retn/mempwr/dftram to meet SCU power up timing
// PGC
reg32_write(GPC_IPS_BASE_ADDR +0xA00, reg32_read(GPC_IPS_BASE_ADDR +0xA00) | 0x00000001) ;
// enable MIX PGC power down
```

5.2.9.3 Example Code 3

```
//A53/M4 both enters into low power mode. A53/M4 are in different master domain.
//MIX are mapping to both A53 and M4
//after A53/M4 enters into low power mode, MIX will be also power down.A53/M4 enters into
low power mode any time
//either A53 or M4 exists from low power mode, MIX will be also power up
//IMRx_CORE0_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x30, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x34, 0xFF7FFFFF);
//[23] : GPT1 used as Arm wakeup source
reg32_write(GPC_IPS_BASE_ADDR + 0x38, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x3C, 0xFFFFFFFF);
//IMRx_CORE1_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x40, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x44, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x48, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x4C, 0xFFFFFFFF);
//IMRx_CORE2_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x1C0, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1C4, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1C8, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1CC, 0xFFFFFFFF);
//IMRx_CORE3_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x1D0, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1D4, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1D8, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x1DC, 0xFFFFFFFF);
//IMRx_M4
reg32_write(GPC_IPS_BASE_ADDR + 0x50, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x54, 0xFFBFFFFF);
```

```
//[22] : GPT2 used as M4 wakeup source
reg32_write(GPC_IPS_BASE_ADDR + 0x58, 0xFFFFFFFF);
reg32_write(GPC_IPS_BASE_ADDR + 0x5C, 0xFFFFFFFF);
//LPCR_A53_BSC
reg32_write(GPC_IPS_BASE_ADDR, 0x0000000A) ;
//[30],[23:22],[29:28] : A53_C0/A53_C1/A53_C2/A53_C3/LPM wakeup from external INT
//[14] : CLOCK OFF during STOP mode
//[3:0] : STOP mode
//LPCR_A53_BSC2
reg32_write(GPC_IPS_BASE_ADDR + 0x108, 0x0000000A) ;
//[3:0] : STOP mode
//LPCR_A53_AD
reg32_write(GPC_IPS_BASE_ADDR + 0x4, 0x0A0A0A1A) ;
//[16] : 1(ALL_OFF mode); 0(L2 retention mode)
//[27]/[25]/[11]/[9]: A53_C0/A53_C1/A53_C2/A53_C3 power up with A53 LPM PUP REQ
//[4] : A53_SCU power down with A53 LPM PDN REQ
//[19]/[17]/[3]/[1] : A53_C0/A53_C1/A53_C2/A53_C3 powr down with A53 LPM PDN REQ
//LPCR_M4
reg32_write(GPC_IPS_BASE_ADDR + 0x8, 0x00003FFE) ;
//[31] : 0(check M4 low power state to enter into DSM)
//[14] : 0(M4 clock OFF during low power mode)
//[3:2] : enable M4 virtual PGC power up/down with M4 LPM
//[1:0] : M4 LPM STOP mode
//SLPCR
reg32_write(GPC_IPS_BASE_ADDR + 0x14,0xe000ffA7) ;
//[31] : enable DSM
//[30] : eneable regulator bypass
//[5:3] : wait 64 ckil clock cycles
//[2] : enable PMIC standby
//[1] : enable OSC power down
//[0] : bypass PMIC ready handshake
//PGC_ACK_SEL_A53
reg32_write(GPC_IPS_BASE_ADDR + 0x24,0x00010004) ;
//[2] : A53_SCU PGC as LPM power down ack
//[16] : A53_C0 PGC as LPM power up ack
//PGC_ACK_SEL_M4
reg32_write(GPC_IPS_BASE_ADDR + 0x28,0x00010001) ;
//[0] : M4 virtual PGC as M4 LPM power down ack
//[16] : M4 virtual PGC as M4 LPM power up ack
//GPC_MISC
reg32_write(GPC_IPS_BASE_ADDR + 0x2C,reg32_read(GPC_IPS_BASE_ADDR + 0x2C) | 0x100) ;
//[8] : not mask M4 power down request to M4 virtual PGC
//SLT_CFG0
reg32_write(GPC_IPS_BASE_ADDR + 0xB0,0x00000155) ;
//[6]/[4]/[2]/[0] : A53_C0/A53_C1/ A53_C2/A53_C3 power down in SLOT0
//[8] : A53_SCU power down in SLOT0
//SLT_CFG0_PU
reg32_write(GPC_IPS_BASE_ADDR + 0x200,0x0001001) ;
//[0] : power down in SLOT0
//[12] : M4 virtual PGC power down in SLOT0
//A53_Cx/SCU are power down in same slot. Special setting is required( see below PGC setting
#A )
//SLT_CFG1_PU
reg32_write(GPC_IPS_BASE_ADDR + 0x204,0x00002002) ;
//[2] : power up in SLOT1
//[13] : M4 virtual PGC power up in SLOT1
//SLT_CFG2
reg32_write(GPC_IPS_BASE_ADDR + 0xB8,0x00000202) ;
//[9] : A53_SCU power up in SLOT1
//[1] : A53_C0 power up in SLOT1, A53_C1/ A53_C2/ A53_C3 not power up
//A53_Cx/SCU are power up in same slot. Special setting is required( see below PGC setting
#B )
//PGC_CPU_MAPPING
reg32_write(GPC_IPS_BASE_ADDR + 0xEC,0x00010001) ;
//[0] : MIX PGC mapping to A53 LPM
//[16] : MIX PGC mapping to M4 LPM
//Special PGC setting #A for C0/C1/SCU power down in same slot(SCU should be always ON
comparing to C0/C1/C2/C3)
reg32_write(GPC_IPS_BASE_ADDR +0x808, (reg32_read(GPC_IPS_BASE_ADDR +0x808) & 0xFFFFC0C0) |
0x0801 ) ;
```

```
// set C0.ISO2SW = 8 ; C0.ISO = 1;
reg32_write(GPC_IPS_BASE_ADDR +0x848, (reg32_read(GPC_IPS_BASE_ADDR +0x848) & 0xFFFFC0C0) |
0x0801 ) ;
// set C1.ISO2SW = 8 ; C1.ISO = 1;
reg32_write(GPC_IPS_BASE_ADDR +0x848, (reg32_read(GPC_IPS_BASE_ADDR +0x848) & 0xFFFFC0C0) |
0x0801 ) ;
// set C2.ISO2SW = 8 ; C2.ISO = 1;
reg32_write(GPC_IPS_BASE_ADDR +0x888, (reg32_read(GPC_IPS_BASE_ADDR +0x888) & 0xFFFFC0C0) |
0x0801 ) ;
// set C3.ISO2SW = 8 ; C3.ISO = 1;
reg32_write(GPC_IPS_BASE_ADDR +0x908, (reg32_read(GPC_IPS_BASE_ADDR +0x908) & 0xFFFFC0C0) |
0x1001 ) ;
// set SCU.ISO2SW = 16 ; SCU.ISO = 1;
//Special PGC setting #B for A53_Cx/SCU power up in same slot(SCU should be always ON
comparing to C0/C1/C2/C3)
reg32_write(GPC_IPS_BASE_ADDR +0x804, (reg32_read(GPC_IPS_BASE_ADDR +0x804) & 0xFF800040) |
0x10 | (0x1f lt:lt: 7) ) ;
// set C0.SW = 0x10 ; C0.SW2ISO = 0x1f ;
reg32_write(GPC_IPS_BASE_ADDR +0x844, (reg32_read(GPC_IPS_BASE_ADDR +0x844) & 0xFF800040) |
0x10 | (0x1f lt:lt: 7) ) ;
// set C1.SW = 0x10 ; C1.SW2ISO = 0x1f ;
reg32_write(GPC_IPS_BASE_ADDR +0x844, (reg32_read(GPC_IPS_BASE_ADDR +0x844) & 0xFF800040) |
0x10 | (0x1f lt:lt: 7) ) ;
// set C2.SW = 0x10 ; C2.SW2ISO = 0x1f ;
reg32_write(GPC_IPS_BASE_ADDR +0x884, (reg32_read(GPC_IPS_BASE_ADDR +0x884) & 0xFF800040) |
0x10 | (0x1f lt:lt: 7) ) ;
// set C3.SW = 0x10 ; C3.SW2ISO = 0x1f ;
reg32_write(GPC_IPS_BASE_ADDR +0x904, (reg32_read(GPC_IPS_BASE_ADDR +0x904) & 0xFF800040) |
0x1 | (0x0f lt:lt: 7) ) ;
// set SCU.SW = 0x1 ; SCU.SW2ISO = 0x0f ;
//A53 PGC
reg32_write(GPC_IPS_BASE_ADDR +0x800, reg32_read(GPC_IPS_BASE_ADDR +0x800) | 0x00000001) ;
// enable A53_C0 PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x840, reg32_read(GPC_IPS_BASE_ADDR +0x840) | 0x00000001) ;
// enable A53_C1 PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x840, reg32_read(GPC_IPS_BASE_ADDR +0x840) | 0x00000001) ;
// enable A53_C2 PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x880, reg32_read(GPC_IPS_BASE_ADDR +0x880) | 0x00000001) ;
// enable A53_C3 PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x900, reg32_read(GPC_IPS_BASE_ADDR +0x900) | 0x00000001) ;
// enable A53_SCU PGC power down
reg32_write(GPC_IPS_BASE_ADDR +0x910, (0x59 lt:lt: 10) | 0x5B | (0x51 lt:lt: 20) ) ;
// change nL2retn/mempwr/dftram to meet SCU power up timing
// PGC
reg32_write(GPC_IPS_BASE_ADDR +0xA00, reg32_read(GPC_IPS_BASE_ADDR +0xA00) | 0x00000001) ;
// enable MIX PGC power down
```

5.2.9.4 Example Code 4

```
// software power up/down PCIE PHY
//power up PCIE PHY
reg32_write ( GPC_IPS_BASE_ADDR + 0xEC, reg32_read(GPC_IPS_BASE_ADDR + 0x0EC) | 0x8 );
//map PCIE PGC to A53
reg32_write ( GPC_IPS_BASE_ADDR + 0xF8 , reg32_read(GPC_IPS_BASE_ADDR + 0xF8) | 0x2 );
//trigger sw Power up Request
while( read(GPC_IPS_BASE_ADDR + 0xF8) & 0x2 );
//wait software power up request self clear
//power down PCIE PHY
reg32_write ( GPC_IPS_BASE_ADDR + 0xEC, reg32_read(GPC_IPS_BASE_ADDR + 0x0EC) | 0x8 );
//map PCIE PGC to A53
reg32_write ( GPC_IPS_BASE_ADDR + 0xC40, reg32_read(GPC_IPS_BASE_ADDR + 0xC40) | 0x1 );
// enable PCIE PGC power down
reg32_write ( GPC_IPS_BASE_ADDR + 0x104 , reg32_read(GPC_IPS_BASE_ADDR + 0x104) | 0x2 );
```

```
//trigger sw Power Down Request
while( read(GPC_IPS_BASE_ADDR + 0x104) & 0x2 );
//wait software power down request self clear
```

5.2.10 GPC Memory Map/Register Definition

Detailed descriptions of each register can be found below.

The total GPC memory map is 4KB

Table 5-10. Memory Regions

Address Range(offset)	Region
0x000 - 0x3FF	GPC configuration register
0x400 - 0x7FF	Reserved
0x800 - 0x9FF	CPU and SCU type PGC register base address
0xA000 - 0xAFF	MIX type PGC register base address
0xB00 - 0xFFF	PU type PGC register base address

Each PGC (CPU type, MIX type, PU type) will occupy 64 Bytes address space, the specific base address of each PGC are listed as below.

- 0x800 ~ 0x83F : PGC for A53 core0
- 0x840 ~ 0x87F: PGC for A53 core1
- 0x880 ~ 0x8BF: PGC for A53 core2
- 0x8C0 ~ 0x8FF: PGC for A53 core3
- 0x900 ~ 0x93F: PGC for A53 SCU
- 0xA00 ~ 0xA3F: Reserved
- 0xA40 ~ 0xA7F: PGC for NOC mix
- 0xB00 ~ 0xB3F: PGC for MIPI PHY1
- 0xB40 ~ 0xB7F: PGC for PCIE PHY
- 0xB80 ~ 0xBBF: PGC for USB1 PHY
- 0xBC0 ~ 0xBFF: PGC for USB2 PHY
- 0xC00 ~ 0xC3F: PGC for MLMIX
- 0xC40 ~ 0xC7F: PGC for AUDIOMIX
- 0xC80 ~ 0xCBF: PGC for GPU2D
- 0xCC0 ~ 0xCFF: PGC for GPU Share Logic
- 0xD00 ~ 0xD3F: PGC for VPUMIX Share Logic
- 0xD40 ~ 0xD7F: PGC for GPU3D
- 0xD80 ~ 0xDBF: PGC for MEDIMIX
- 0xDC0 ~ 0xDFF: PGC for VPU G1
- 0xE00 ~ 0xE3F: PGC for VPU G2

- 0xE40 ~ 0xE7F: PGC for VPU VC8000E
- 0xE80 ~ 0xEBF: PGC for HDMIMIX
- 0xEC0 ~ 0xEFF: PGC for HDMI PHY
- 0xF00 ~ 0xF3F: PGC for MIPI PHY2
- 0xF40 ~ 0xF7F: PGC for HSIOMIX
- 0xF80 ~ 0xFBF: PGC for Media ISP DWP
- 0xFC0 ~ 0xFFF: PGC for DDRMIX

For more specific information about PGC register definition, please see the register definition for each PGC.

GPC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0000	Basic Low power control register of A53 platform (GPC_LPCR_A53_BSC)	32	R/W	0000_3FF0h	5.2.10.1/602
303A_0004	Advanced Low power control register of A53 platform (GPC_LPCR_A53_AD)	32	R/W	0000_0020h	5.2.10.2/605
303A_0008	Low power control register of CPU1 (GPC_LPCR_M7)	32	R/W	0000_3FF0h	5.2.10.3/607
303A_0014	System low power control register (GPC_SLPCR)	32	R/W	E000_FF82h	5.2.10.4/609
303A_0018	MASTER LPM Handshake (GPC_MST_CPU_MAPPING)	32	R/W	0000_00FFh	5.2.10.5/612
303A_0020	Memory low power control register (GPC_MLPCR)	32	R/W	0101_0100h	5.2.10.6/613
303A_0024	PGC acknowledge signal selection of A53 platform (GPC_PGC_ACK_SEL_A53)	32	R/W	8000_8000h	5.2.10.7/614
303A_0028	PGC acknowledge signal selection of M7 platform (GPC_PGC_ACK_SEL_M7)	32	R/W	8000_8000h	5.2.10.8/616
303A_002C	GPC Miscellaneous register (GPC_MISC)	32	R/W	0000_0021h	5.2.10.9/617
303A_0030	IRQ masking register 1 of A53 core0 (GPC_IMR1_CORE0_A53)	32	R/W	0000_0000h	5.2.10.10/ 619
303A_0034	IRQ masking register 2 of A53 core0 (GPC_IMR2_CORE0_A53)	32	R/W	0000_0000h	5.2.10.11/ 619
303A_0038	IRQ masking register 3 of A53 core0 (GPC_IMR3_CORE0_A53)	32	R/W	0000_0000h	5.2.10.12/ 619
303A_003C	IRQ masking register 4 of A53 core0 (GPC_IMR4_CORE0_A53)	32	R/W	0000_0000h	5.2.10.13/ 620
303A_0040	IRQ masking register 5 of A53 core0 (GPC_IMR5_CORE0_A53)	32	R/W	0000_0000h	5.2.10.14/ 620
303A_0044	IRQ masking register 1 of A53 core1 (GPC_IMR1_CORE1_A53)	32	R/W	0000_0000h	5.2.10.15/ 621
303A_0048	IRQ masking register 2 of A53 core1 (GPC_IMR2_CORE1_A53)	32	R/W	0000_0000h	5.2.10.16/ 621
303A_004C	IRQ masking register 3 of A53 core1 (GPC_IMR3_CORE1_A53)	32	R/W	0000_0000h	5.2.10.17/ 622
303A_0050	IRQ masking register 4 of A53 core1 (GPC_IMR4_CORE1_A53)	32	R/W	0000_0000h	5.2.10.18/ 622

Table continues on the next page...

GPC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0054	IRQ masking register 5 of A53 core1 (GPC_IMR5_CORE1_A53)	32	R/W	0000_0000h	5.2.10.19/ 622
303A_0058	IRQ masking register 1 of M7 (GPC_IMR1_M7)	32	R/W	0000_0000h	5.2.10.20/ 623
303A_005C	IRQ masking register 2 of M7 (GPC_IMR2_M7)	32	R/W	0000_0000h	5.2.10.21/ 623
303A_0060	IRQ masking register 3 of M7 (GPC_IMR3_M7)	32	R/W	0000_0000h	5.2.10.22/ 624
303A_0064	IRQ masking register 4 of M7 (GPC_IMR4_M7)	32	R/W	0000_0000h	5.2.10.23/ 624
303A_0068	IRQ masking register 5 of M7 (GPC_IMR5_M7)	32	R/W	0000_0000h	5.2.10.24/ 624
303A_0080	IRQ status register 1 of A53 (GPC_ISR1_A53)	32	R	0000_0000h	5.2.10.25/ 625
303A_0084	IRQ status register 2 of A53 (GPC_ISR2_A53)	32	R	0000_0000h	5.2.10.26/ 625
303A_0088	IRQ status register 3 of A53 (GPC_ISR3_A53)	32	R	0000_0000h	5.2.10.27/ 626
303A_008C	IRQ status register 4 of A53 (GPC_ISR4_A53)	32	R	0000_0000h	5.2.10.28/ 626
303A_0090	IRQ status register 5 of A53 (GPC_ISR5_A53)	32	R	0000_0000h	5.2.10.29/ 626
303A_0094	IRQ status register 1 of M7 (GPC_ISR1_M7)	32	R	0000_0000h	5.2.10.30/ 627
303A_0098	IRQ status register 2 of M7 (GPC_ISR2_M7)	32	R	0000_0000h	5.2.10.31/ 627
303A_009C	IRQ status register 3 of M7 (GPC_ISR3_M7)	32	R	0000_0000h	5.2.10.32/ 627
303A_00A0	IRQ status register 4 of M7 (GPC_ISR4_M7)	32	R	0000_0000h	5.2.10.33/ 628
303A_00A4	IRQ status register 5 of M7 (GPC_ISR5_M7)	32	R	0000_0000h	5.2.10.34/ 628
303A_00D0	CPU PGC software power up trigger (GPC_CPU_PGC_SW_PUP_REQ)	32	R/W	0000_0000h	5.2.10.35/ 629
303A_00D4	MIX PGC software power up trigger (GPC_MIX_PGC_SW_PUP_REQ)	32	R/W	0000_0000h	5.2.10.36/ 630
303A_00D8	PU PGC software up trigger (GPC_PU_PGC_SW_PUP_REQ)	32	R/W	0000_0000h	5.2.10.37/ 630
303A_00DC	CPU PGC software down trigger (GPC_CPU_PGC_SW_PDN_REQ)	32	R/W	0000_0000h	5.2.10.38/ 632
303A_00E0	MIX PGC software power down trigger (GPC_MIX_PGC_SW_PDN_REQ)	32	R/W	0000_0000h	5.2.10.39/ 633
303A_00E4	PU PGC software down trigger (GPC_PU_PGC_SW_PDN_REQ)	32	R/W	0000_0000h	5.2.10.40/ 634

Table continues on the next page...

GPC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_00EC	Advanced Low power control register of CM4 platform (GPC_GPC_LPCR_CM4_AD)	32	R	0000_0000h	5.2.10.41/ 637
303A_0108	CPU PGC software up trigger status1 (GPC_CPU_PGC_PUP_STATUS1)	32	R	0000_0000h	5.2.10.42/ 638
303A_010C	A53 MIX software up trigger status register (GPC_A53_MIX_PGC_PUP_STATUS0)	32	R	0000_0000h	5.2.10.43/ 640
303A_0110	A53 MIX software up trigger status register (GPC_A53_MIX_PGC_PUP_STATUS1)	32	R	0000_0000h	5.2.10.43/ 640
303A_0114	A53 MIX software up trigger status register (GPC_A53_MIX_PGC_PUP_STATUS2)	32	R	0000_0000h	5.2.10.43/ 640
303A_0118	M7 MIX PGC software up trigger status register (GPC_M7_MIX_PGC_PUP_STATUS0)	32	R	0000_0000h	5.2.10.44/ 641
303A_011C	M7 MIX PGC software up trigger status register (GPC_M7_MIX_PGC_PUP_STATUS1)	32	R	0000_0000h	5.2.10.44/ 641
303A_0120	M7 MIX PGC software up trigger status register (GPC_M7_MIX_PGC_PUP_STATUS2)	32	R	0000_0000h	5.2.10.44/ 641
303A_0124	A53 PU software up trigger status register (GPC_A53_PU_PGC_PUP_STATUS0)	32	R	0000_0000h	5.2.10.45/ 643
303A_0128	A53 PU software up trigger status register (GPC_A53_PU_PGC_PUP_STATUS1)	32	R	0000_0000h	5.2.10.45/ 643
303A_012C	A53 PU software up trigger status register (GPC_A53_PU_PGC_PUP_STATUS2)	32	R	0000_0000h	5.2.10.45/ 643
303A_0130	M7 PU PGC software up trigger status register (GPC_M7_PU_PGC_PUP_STATUS0)	32	R	0000_0000h	5.2.10.46/ 647
303A_0134	M7 PU PGC software up trigger status register (GPC_M7_PU_PGC_PUP_STATUS1)	32	R	0000_0000h	5.2.10.46/ 647
303A_0138	M7 PU PGC software up trigger status register (GPC_M7_PU_PGC_PUP_STATUS2)	32	R	0000_0000h	5.2.10.46/ 647
303A_013C	CPU PGC software dn trigger status1 (GPC_CPU_PGC_PDN_STATUS1)	32	R	0000_0000h	5.2.10.47/ 651
303A_0140	A53 MIX software down trigger status register (GPC_A53_MIX_PGC_PDN_STATUS0)	32	R	0000_0000h	5.2.10.48/ 653
303A_0144	A53 MIX software down trigger status register (GPC_A53_MIX_PGC_PDN_STATUS1)	32	R	0000_0000h	5.2.10.48/ 653
303A_0148	A53 MIX software down trigger status register (GPC_A53_MIX_PGC_PDN_STATUS2)	32	R	0000_0000h	5.2.10.48/ 653
303A_014C	M7 MIX PGC software power down trigger status register (GPC_M7_MIX_PGC_PDN_STATUS0)	32	R	0000_0000h	5.2.10.49/ 654
303A_0150	M7 MIX PGC software power down trigger status register (GPC_M7_MIX_PGC_PDN_STATUS1)	32	R	0000_0000h	5.2.10.49/ 654
303A_0154	M7 MIX PGC software power down trigger status register (GPC_M7_MIX_PGC_PDN_STATUS2)	32	R	0000_0000h	5.2.10.49/ 654
303A_0158	A53 PU PGC software down trigger status (GPC_A53_PU_PGC_PDN_STATUS0)	32	R	0000_0000h	5.2.10.50/ 656

Table continues on the next page...

GPC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_015C	A53 PU PGC software down trigger status (GPC_A53_PU_PGC_PDN_STATUS1)	32	R	0000_0000h	5.2.10.50/ 656
303A_0160	A53 PU PGC software down trigger status (GPC_A53_PU_PGC_PDN_STATUS2)	32	R	0000_0000h	5.2.10.50/ 656
303A_0164	M7 PU PGC software down trigger status (GPC_M7_PU_PGC_PDN_STATUS0)	32	R	0000_0000h	5.2.10.51/ 660
303A_0168	M7 PU PGC software down trigger status (GPC_M7_PU_PGC_PDN_STATUS1)	32	R	0000_0000h	5.2.10.51/ 660
303A_016C	M7 PU PGC software down trigger status (GPC_M7_PU_PGC_PDN_STATUS2)	32	R	0000_0000h	5.2.10.51/ 660
303A_0170	A53 MIX PDN FLG (GPC_A53_MIX_PDN_FLG)	32	R/W	0000_0000h	5.2.10.52/ 664
303A_0174	A53 PU PDN FLG (GPC_A53_PU_PDN_FLG)	32	R/W	0000_0000h	5.2.10.53/ 665
303A_0178	M7 MIX PDN FLG (GPC_M7_MIX_PDN_FLG)	32	R/W	0000_0000h	5.2.10.54/ 666
303A_017C	M7 PU PDN FLG (GPC_M7_PU_PDN_FLG)	32	R/W	0000_0000h	5.2.10.55/ 667
303A_0180	Basic Low power control register of A53 platform (GPC_LPCR_A53_BSC2)	32	R/W	0000_0000h	5.2.10.56/ 667
303A_0190	Power handshake register (GPC_PU_PWRHSK)	32	R/W	0000_FFFFh	5.2.10.57/ 669
303A_0194	IRQ masking register 1 of A53 core2 (GPC_IMR1_CORE2_A53)	32	R/W	0000_0000h	5.2.10.58/ 673
303A_0198	IRQ masking register 2 of A53 core2 (GPC_IMR2_CORE2_A53)	32	R/W	0000_0000h	5.2.10.59/ 673
303A_019C	IRQ masking register 3 of A53 core2 (GPC_IMR3_CORE2_A53)	32	R/W	0000_0000h	5.2.10.60/ 674
303A_01A0	IRQ masking register 4 of A53 core2 (GPC_IMR4_CORE2_A53)	32	R/W	0000_0000h	5.2.10.61/ 674
303A_01A4	IRQ masking register 5 of A53 core2 (GPC_IMR5_CORE2_A53)	32	R/W	0000_0000h	5.2.10.62/ 675
303A_01A8	IRQ masking register 1 of A53 core3 (GPC_IMR1_CORE3_A53)	32	R/W	0000_0000h	5.2.10.63/ 675
303A_01AC	IRQ masking register 2 of A53 core3 (GPC_IMR2_CORE3_A53)	32	R/W	0000_0000h	5.2.10.64/ 676
303A_01B0	IRQ masking register 3 of A53 core3 (GPC_IMR3_CORE3_A53)	32	R/W	0000_0000h	5.2.10.65/ 676
303A_01B4	IRQ masking register 4 of A53 core3 (GPC_IMR4_CORE3_A53)	32	R/W	0000_0000h	5.2.10.66/ 677
303A_01B8	IRQ masking register 5 of A53 core3 (GPC_IMR5_CORE3_A53)	32	R/W	0000_0000h	5.2.10.67/ 677
303A_01BC	PGC acknowledge signal selection of A53 platform for PUs (GPC_ACK_SEL_A53_PU)	32	R/W	0000_0000h	5.2.10.68/ 678

Table continues on the next page...

GPC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_01C0	PGC acknowledge signal selection of A53 platform for PUs (GPC_ACK_SEL_A53_PU1)	32	R/W	0000_0000h	5.2.10.69/ 681
303A_01C4	PGC acknowledge signal selection of M7 platform for PUs (GPC_ACK_SEL_M7_PU)	32	R/W	0000_0000h	5.2.10.70/ 682
303A_01C8	PGC acknowledge signal selection of M7 platform for PUs (GPC_ACK_SEL_M7_PU1)	32	R/W	0000_0000h	5.2.10.71/ 685
303A_01CC	PGC CPU A53 mapping (GPC_PGC_CPU_A53_MAPPING)	32	R/W	0000_0000h	5.2.10.72/ 687
303A_01D0	PGC CPU M7 mapping (GPC_PGC_CPU_M7_MAPPING)	32	R/W	0000_0000h	5.2.10.73/ 689
303A_0200	Slot configure register for CPUs (GPC_SLT0_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0204	Slot configure register for CPUs (GPC_SLT1_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0208	Slot configure register for CPUs (GPC_SLT2_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_020C	Slot configure register for CPUs (GPC_SLT3_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0210	Slot configure register for CPUs (GPC_SLT4_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0214	Slot configure register for CPUs (GPC_SLT5_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0218	Slot configure register for CPUs (GPC_SLT6_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_021C	Slot configure register for CPUs (GPC_SLT7_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0220	Slot configure register for CPUs (GPC_SLT8_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0224	Slot configure register for CPUs (GPC_SLT9_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0228	Slot configure register for CPUs (GPC_SLT10_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_022C	Slot configure register for CPUs (GPC_SLT11_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0230	Slot configure register for CPUs (GPC_SLT12_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0234	Slot configure register for CPUs (GPC_SLT13_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0238	Slot configure register for CPUs (GPC_SLT14_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_023C	Slot configure register for CPUs (GPC_SLT15_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0240	Slot configure register for CPUs (GPC_SLT16_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692

Table continues on the next page...

GPC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0244	Slot configure register for CPUs (GPC_SLT17_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0248	Slot configure register for CPUs (GPC_SLT18_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_024C	Slot configure register for CPUs (GPC_SLT19_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0250	Slot configure register for CPUs (GPC_SLT20_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0254	Slot configure register for CPUs (GPC_SLT21_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0258	Slot configure register for CPUs (GPC_SLT22_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_025C	Slot configure register for CPUs (GPC_SLT23_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0260	Slot configure register for CPUs (GPC_SLT24_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0264	Slot configure register for CPUs (GPC_SLT25_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0268	Slot configure register for CPUs (GPC_SLT26_CFG)	32	R/W	0000_0000h	5.2.10.74/ 692
303A_0280	Slot configure register for PGC PUs (GPC_SLT0_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_0284	Extended slot configure register for PGC PUs (GPC_SLT0_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_0288	Slot configure register for PGC PUs (GPC_SLT1_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_028C	Extended slot configure register for PGC PUs (GPC_SLT1_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_0290	Slot configure register for PGC PUs (GPC_SLT2_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_0294	Extended slot configure register for PGC PUs (GPC_SLT2_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_0298	Slot configure register for PGC PUs (GPC_SLT3_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_029C	Extended slot configure register for PGC PUs (GPC_SLT3_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_02A0	Slot configure register for PGC PUs (GPC_SLT4_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_02A4	Extended slot configure register for PGC PUs (GPC_SLT4_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_02A8	Slot configure register for PGC PUs (GPC_SLT5_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_02AC	Extended slot configure register for PGC PUs (GPC_SLT5_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698

Table continues on the next page...

GPC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_02B0	Slot configure register for PGC PUs (GPC_SLT6_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_02B4	Extended slot configure register for PGC PUs (GPC_SLT6_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_02B8	Slot configure register for PGC PUs (GPC_SLT7_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_02BC	Extended slot configure register for PGC PUs (GPC_SLT7_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_02C0	Slot configure register for PGC PUs (GPC_SLT8_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_02C4	Extended slot configure register for PGC PUs (GPC_SLT8_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_02C8	Slot configure register for PGC PUs (GPC_SLT9_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_02CC	Extended slot configure register for PGC PUs (GPC_SLT9_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_02D0	Slot configure register for PGC PUs (GPC_SLT10_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_02D4	Extended slot configure register for PGC PUs (GPC_SLT10_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_02D8	Slot configure register for PGC PUs (GPC_SLT11_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_02DC	Extended slot configure register for PGC PUs (GPC_SLT11_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_02E0	Slot configure register for PGC PUs (GPC_SLT12_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_02E4	Extended slot configure register for PGC PUs (GPC_SLT12_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_02E8	Slot configure register for PGC PUs (GPC_SLT13_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_02EC	Extended slot configure register for PGC PUs (GPC_SLT13_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_02F0	Slot configure register for PGC PUs (GPC_SLT14_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_02F4	Extended slot configure register for PGC PUs (GPC_SLT14_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_02F8	Slot configure register for PGC PUs (GPC_SLT15_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_02FC	Extended slot configure register for PGC PUs (GPC_SLT15_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_0300	Slot configure register for PGC PUs (GPC_SLT16_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_0304	Extended slot configure register for PGC PUs (GPC_SLT16_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698

Table continues on the next page...

GPC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0308	Slot configure register for PGC PUs (GPC_SLT17_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_030C	Extended slot configure register for PGC PUs (GPC_SLT17_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_0310	Slot configure register for PGC PUs (GPC_SLT18_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_0314	Extended slot configure register for PGC PUs (GPC_SLT18_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_0318	Slot configure register for PGC PUs (GPC_SLT19_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_031C	Extended slot configure register for PGC PUs (GPC_SLT19_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_0320	Slot configure register for PGC PUs (GPC_SLT20_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_0324	Extended slot configure register for PGC PUs (GPC_SLT20_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_0328	Slot configure register for PGC PUs (GPC_SLT21_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_032C	Extended slot configure register for PGC PUs (GPC_SLT21_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_0330	Slot configure register for PGC PUs (GPC_SLT22_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_0334	Extended slot configure register for PGC PUs (GPC_SLT22_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_0338	Slot configure register for PGC PUs (GPC_SLT23_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_033C	Extended slot configure register for PGC PUs (GPC_SLT23_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_0340	Slot configure register for PGC PUs (GPC_SLT24_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_0344	Extended slot configure register for PGC PUs (GPC_SLT24_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_0348	Slot configure register for PGC PUs (GPC_SLT25_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_034C	Extended slot configure register for PGC PUs (GPC_SLT25_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698
303A_0350	Slot configure register for PGC PUs (GPC_SLT26_CFG_PU)	32	R/W	0000_0000h	5.2.10.75/ 694
303A_0354	Extended slot configure register for PGC PUs (GPC_SLT26_CFG_PU1)	32	R/W	0000_0000h	5.2.10.76/ 698

5.2.10.1 Basic Low power control register of A53 platform (GPC_LPCR_A53_BSC)

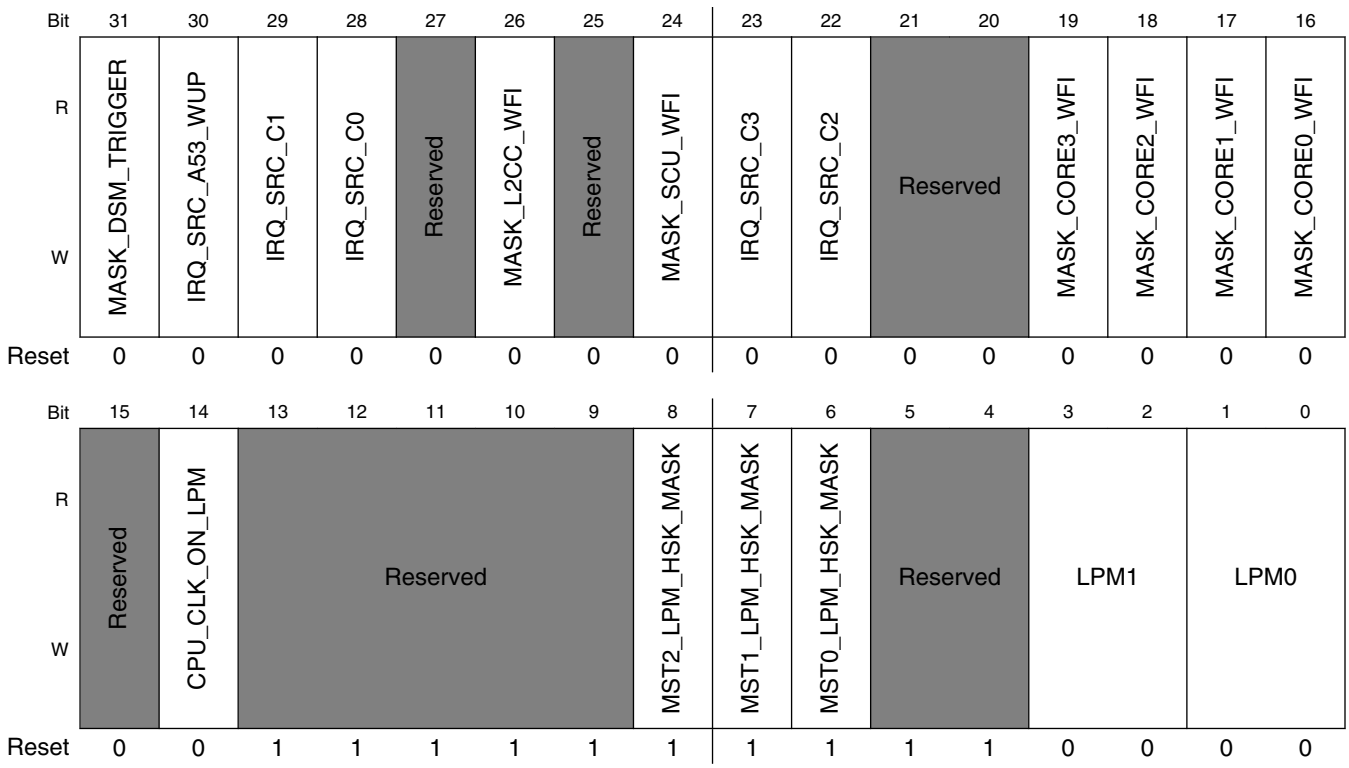
NOTE

LPCR_A53_BSC[CPU_CLK_ON_LPM] should be set 1'b1 when using A53 low power debug feature

NOTE

Always set LPM1/LPM0 with same value

Address: 303A_0000h base + 0h offset = 303A_0000h



GPC_LPCR_A53_BSC field descriptions

Field	Description
31 MASK_DSM_TRIGGER	DSM Trigger Mask 0 DSM trigger of A53 platform will not be masked 1 DSM trigger of A53 platform will be masked
30 IRQ_SRC_A53_WUP	LPCR_A53_BSC[IRQ_SRC_C0], LPCR_A53_BSC[IRQ_SRC_C1], LPCR_A53_BSC[IRQ_SRC_C2], LPCR_A53_BSC[IRQ_SRC_C3], and LPCR_A53_BSC[IRQ_SRC_A53_WUP] work together to decide the wake up source for A53 LPM and core0/core1/core2/core3 power. See "Power up process for A53 platform" for more specific information.

Table continues on the next page...

GPC_LPCR_A53_BSC field descriptions (continued)

Field	Description
	0 LPM wakeup source be “OR” result of LPCR_A53_BSC[IRQ_SRC_C0]/ LPCR_A53_BSC[IRQ_SRC_C1]/LPCR_A53_BSC[IRQ_SRC_C2]/LPCR_A53_BSC[IRQ_SRC_C3] setting 1 LPM wakeup source from external INT[127:0], masked by IMR0
29 IRQ_SRC_C1	LPCR_A53_BSC[IRQ_SRC_C0], LPCR_A53_BSC[IRQ_SRC_C1], LPCR_A53_BSC[IRQ_SRC_C2], LPCR_A53_BSC[IRQ_SRC_C3], and LPCR_A53_BSC[IRQ_SRC_A53_WUP] work together to decide the wake up source for A53 LPM and core0/core1/core2/core3 power. See “Power up process for A53 platform” for more specific information. 0 core1 wakeup source from external INT[127:0], masked by IMR1 refer to “Power up process for A53 platform” for more specific information 1 core1 wakeup source from GIC(nFIQ[1]/nIRQ[1]), SCU should not be power down during low power mode when this bit is set to 1’b1
28 IRQ_SRC_C0	LPCR_A53_BSC[IRQ_SRC_C0], LPCR_A53_BSC[IRQ_SRC_C1], LPCR_A53_BSC[IRQ_SRC_C2], LPCR_A53_BSC[IRQ_SRC_C3], and LPCR_A53_BSC[IRQ_SRC_A53_WUP] work together to decide the wake up source for A53 LPM and core0/core1/core2/core3 power. See “Power up process for A53 platform” for more specific information. 0 core0 wakeup source from external INT[127:0], masked by IMR0 refer to “Power up process for A53 platform” for more specific information 1 core0 wakeup source from GIC(nFIQ[0]/nIRQ[0]), SCU should not be power down during low power mode when this bit is set to 1’b1
27 -	This field is reserved. Reserved
26 MASK_L2CC_ WFI	L2 cache controller Wait For Interrupt Mask Register 0 WFI for L2 cache controller is not masked 1 WFI for L2 cache controller is masked
25 -	This field is reserved. Reserved
24 MASK_SCU_WFI	SCU Wait For Interrupt Mask Register 0 WFI for SCU is not masked 1 WFI for SCU is masked
23 IRQ_SRC_C3	LPCR_A53_BSC[IRQ_SRC_C0], LPCR_A53_BSC[IRQ_SRC_C1], LPCR_A53_BSC[IRQ_SRC_C2], LPCR_A53_BSC[IRQ_SRC_C3], and LPCR_A53_BSC[IRQ_SRC_A53_WUP] work together to decide the wake up source for A53 LPM and core0/core1/core2/core3 power. 0 core3 wakeup source from external INT[127:0], masked by IMR1. See Power Up Process for A53 Platform for more specific information. 1 core3 wakeup source from external GIC(nFIQ[1]/nIRQ[1]), SCU should not be powered down during low power mode when this bit is set to 1'b1.
22 IRQ_SRC_C2	LPCR_A53_BSC[IRQ_SRC_C0], LPCR_A53_BSC[IRQ_SRC_C1], LPCR_A53_BSC[IRQ_SRC_C2], LPCR_A53_BSC[IRQ_SRC_C3], and LPCR_A53_BSC[IRQ_SRC_A53_WUP] work together to decide the wake up source for A53 LPM and core0/core1/core2/core3 power. 0 core2 wakeup source from external INT[127:0], masked by IMR1. See Power Up Process for A53 Platform for more specific information. 1 core2 wakeup source from external GIC(nFIQ[1]/nIRQ[1]), SCU should not be powered down during low power mode when this bit is set to 1'b1.

Table continues on the next page...

GPC_LPCR_A53_BSC field descriptions (continued)

Field	Description
21–20 -	This field is reserved.
19 MASK_CORE3_WFI	CORE3 Wait For Interrupt Mask 0 WFI for CORE3 is not masked 1 WFI for CORE3 is masked
18 MASK_CORE2_WFI	CORE2 Wait For Interrupt Mask 0 WFI for CORE2 is not masked 1 WFI for CORE2 is masked
17 MASK_CORE1_WFI	CORE1 Wait For Interrupt Mask 0 WFI for CORE1 is not masked 1 WFI for CORE1 is masked
16 MASK_CORE0_WFI	CORE0 Wait For Interrupt Mask 0 WFI for CORE0 is not masked 1 WFI for CORE0 is masked
15 -	This field is reserved. Reserved
14 CPU_CLK_ON_LPM	Define if A53 clocks will be disabled on wait/stop mode. 0 A53 clock disabled on wait/stop mode 1 A53 clock enabled on wait/stop mode
13–9 -	This field is reserved. Reserved
8 MST2_LPM_HSK_MASK	MASTER2 LPM handshake mask MASTER2(supermix2noc ADB) will handshake with GPC in LPM, follow this when you want this master power off. This bit should use together with MST_CPU_MAPPING[2] If you want power of supermix2noc ADB, use this setting: LPCR_A53_BSC[8]=0; MST_CPU_MAPPING[2]=1 Otherwise use: LPCR_A53_BSC[8]=1; MST_CPU_MAPPING[2]=0 0 enable MASTER2 LPM handshake, wait ACK from MASTER2 1 disable MASTER2 LPM handshake, mask ACK from MASTER2
7 MST1_LPM_HSK_MASK	MASTER1 LPM handshake mask MASTER1(supermix2noc ADB) will handshake with GPC in LPM, follow this when you want this master power off. This bit should use together with MST_CPU_MAPPING[1] If you want power of supermix2noc ADB, use this setting: LPCR_A53_BSC[7]=0; MST_CPU_MAPPING[1]=1 Otherwise use: LPCR_A53_BSC[7]=1; MST_CPU_MAPPING[1]=0 0 enable MASTER1 LPM handshake, wait ACK from MASTER1 1 disable MASTER1 LPM handshake, mask ACK from MASTER1
6 MST0_LPM_HSK_MASK	MASTER0 LPM handshake mask

Table continues on the next page...

GPC_LPCR_A53_BSC field descriptions (continued)

Field	Description
	MASTER0(SCU) will handshake with GPC in LPM, follow this when you want this master power off. This bit should be used together with MST_CPU_MAPPING[0]. If you want power of SCU, use this setting: LPCR_A53_BSC[6]=0; MST_CPU_MAPPING[0]=1 Otherwise use: LPCR_A53_BSC[6]=1; MST_CPU_MAPPING[0]=0 0 enable MASTER0 LPM handshake, wait ACK from MASTER0 1 disable MASTER0 LPM handshake, mask ACK from MASTER0
5–4 -	This field is reserved. Reserved
3–2 LPM1	CORE1 Setting the low power mode that system will enter on next assertion of dsm_request signal. 00 Remain in RUN mode 01 Transfer to WAIT mode 10 Transfer to STOP mode 11 Reserved
LPM0	CORE0 Setting the low power mode that system will enter on next assertion of dsm_request signal. 00 Remain in RUN mode 01 Transfer to WAIT mode 10 Transfer to STOP mode 11 Reserved

5.2.10.2 Advanced Low power control register of A53 platform (GPC_LPCR_A53_AD)

Address: 303A_0000h base + 4h offset = 303A_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L2PGE	Reserved			EN_C3_PUP	EN_C3_IRQ_PUP	EN_C2_PUP	EN_C2_IRQ_PUP	Reserved				EN_C3_PDN	EN_C3_WFI_PDN	EN_C2_PDN	EN_C2_WFI_PDN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				EN_C1_PUP	EN_C1_IRQ_PUP	EN_C0_PUP	EN_C0_IRQ_PUP	Reserved		EN_L2_WFI_PDN	EN_PLAT_PDN	EN_C1_PDN	EN_C1_WFI_PDN	EN_C0_PDN	EN_C0_WFI_PDN
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

GPC_LPCR_A53_AD field descriptions

Field	Description
31 L2PGE	0 L2 cache RAM will power down with SCU power domain in A53 platform (used for ALL_OFF mode) 1 L2 cache RAM will not power down with SCU power domain in A53 platform (used for ALL_OFF mode)
30–28 -	This field is reserved. Reserved
27 EN_C3_PUP	0 CORE3 will not power up with lower power mode request 1 CORE3 will power up with low power mode request (only used wake up from CPU_OFF)
26 EN_C3_IRQ_PUP	0 CORE3 will not power up with IRQ request 1 CORE3 will power up with IRQ request
25 EN_C2_PUP	0 CORE2 will not power up with lower power mode request 1 CORE2 will power up with low power mode request (only used wake up from CPU_OFF)
24 EN_C2_IRQ_PUP	0 CORE2 will not power up with IRQ request 1 CORE2 will power up with IRQ request
23–20 -	This field is reserved. Reserved
19 EN_C3_PDN	0 CORE3 will not be power down with low power mode request 1 CORE3 will be power down with low power mode request
18 EN_C3_WFI_PDN	0 CORE3 will not be power down with WFI request 1 CORE3 will be power down with WFI request
17 EN_C2_PDN	0 CORE2 will not be power down with low power mode request 1 CORE2 will be power down with low power mode request
16 EN_C2_WFI_PDN	0 CORE2 will not be power down with WFI request 1 CORE2 will be power down with WFI request
15–12 -	This field is reserved. Reserved
11 EN_C1_PUP	0 CORE1 will not power up with low power mode request (only used wake up from CPU01_OFF mode) 1 CORE1 will power up with low power mode request
10 EN_C1_IRQ_PUP	0 CORE1 will not power up with IRQ request 1 CORE1 will power up with IRQ request
9 EN_C0_PUP	(only used wake up from CPU01_OFF mode) 0 CORE0 will not power up with low power mode request 1 CORE0 will power up with low power mode request
8 EN_C0_IRQ_PUP	0 CORE0 will not power up with IRQ request 1 CORE0 will power up with IRQ request
7–6 -	This field is reserved. Reserved
5 EN_L2_WFI_PDN	**NOTE:** Before reset, L2 WFI is 1 and make GPC generate an error DSM request. This bit is used to mask the L2 WFI before reset. After reset, L2 WFI change to 0, and functions are OK, SW must clear this bit at the beginning of code.

Table continues on the next page...

GPC_LPCR_A53_AD field descriptions (continued)

Field	Description
	0 SCU and L2 will be power down with WFI request 1 SCU and L2 will not be power down with WFI request (default)
4 EN_PLAT_PDN	0 SCU and L2 cache RAM will not be power down with low power mode request 1 SCU and L2 cache RAM will be power down with low power mode request
3 EN_C1_PDN	0 CORE1 will not be power down with low power mode request 1 CORE1 will be power down with low power mode request
2 EN_C1_WFI_PDN	0 CORE1 will not be power down with WFI request 1 CORE1 will be power down with WFI request
1 EN_C0_PDN	0 CORE0 will not be power down with low power mode request 1 CORE0 will be power down with low power mode request
0 EN_C0_WFI_PDN	0 CORE0 will not be power down with WFI request 1 CORE0 will be power down with WFI request

5.2.10.3 Low power control register of CPU1 (GPC_LPCR_M7)

Address: 303A_0000h base + 8h offset = 303A_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MASK_DSM_TRIGGER	Reserved														MASK_M7_WFI
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	CPU_CLK_ON_LPM	Reserved										EN_M7_PUP	EN_M7_PDN	LPM0	
Reset	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0

GPC_LPCR_M7 field descriptions

Field	Description
31 MASK_DSM_TRIGGER	M7 WFI Mask 0 DSM trigger of M7 platform will not be masked 1 DSM trigger of M7 platform will be masked
30–17 -	This field is reserved. Reserved
16 MASK_M7_WFI	M7 WFI Mask 0 WFI for M7 is not masked 1 WFI for M7 is masked
15 -	This field is reserved. Reserved
14 CPU_CLK_ON_LPM	Define if M7 clocks will be disabled on wait/stop mode. 0 M7 clock disabled on wait/stop mode. 1 M7 clock enabled on wait/stop mode.
13–4 -	This field is reserved. Reserved
3 EN_M7_PUP	Enable M7 virtual PGC power up with LPM enter
2 EN_M7_PDN	Enable M7 virtual PGC power down with LPM enter
LPM0	Setting the low power mode that system will enter on next assertion of dsm_request signal. 00 Remain in RUN mode 01 Transfer to WAIT mode 10 Transfer to STOP mode 11 Reserved

5.2.10.4 System low power control register (GPC_SLPCR)

NOTE

SLPCR[VSTBY] must be set to 1'b1 if SLPCR[RBC_EN] is set to 1'b1; SLPCR[SBYOS] must be set to 1'b1 if SLPCR[VSTBY] is set to 1'b1.

Address: 303A_0000h base + 14h offset = 303A_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EN_DSM	RBC_EN	REG_BYPASS_COUNT						DISABLE_A53_IS_DSM	Reserved			EN_M7_FASTWUP_STOP_MODE	EN_M7_FASTWUP_WAIT_MODE	EN_A53_FASTWUP_STOP_MODE	EN_A53_FASTWUP_WAIT_MODE
W																
Reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OSCCNT								COSC_EN	COSC_PWRDOWN	STBY_COUNT			VSTBY	SBYOS	BYPASS_PMIC_READY
W																
Reset	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0

GPC_SLPCR field descriptions

Field	Description
31 EN_DSM	DSM enable 0 DSM disabled 1 DSM enabled
30 RBC_EN	Enable for REG_BYPASS_COUNTER. If enabled, REG_BYPASS signal will be asserted after REG_BYPASS_COUNT clocks of CKIL, after standby voltage is requested. If standby voltage is not requested REG_BYPASS won't be asserted, even if counter is enabled. 0 REG_BYPASS_COUNTER disabled 1 REG_BYPASS_COUNTER enabled
29–24 REG_BYPASS_COUNT	Counter for REG_BYPASS signal assertion after standby voltage request by PMIC_STBY_REQ. 000000 no delay

Table continues on the next page...

GPC_SLPCR field descriptions (continued)

Field	Description
	000001 1 CKIL clock period delay 111111 63 CKIL clock period delay
23 DISABLE_A53_IS_DSM	0 Enable A53 isolation signal in DSM 1 Disable A53 isolation signal in DSM
22–20 -	This field is reserved. Reserved
19 EN_M7_FASTWUP_STOP_MODE	Enable M7 fast wake up stop mode, relevant PLLs will not be closed in this mode.
18 EN_M7_FASTWUP_WAIT_MODE	Enable M7 fast wake up wait mode, relevant PLLs will not be closed in this mode.
17 EN_A53_FASTWUP_STOP_MODE	Enable A53 fast wake up stop mode, relevant PLLs will not be closed in this mode.
16 EN_A53_FASTWUP_WAIT_MODE	Enable A53 fast wake up wait mode, relevant PLLs will not be closed in this mode.
15–8 OSCCNT	Oscillator ready counter value. These bits define value of 32KHz counter, that serve as counter for oscillator lock time. This is used for oscillator lock time. Current estimation is ~5ms. This counter will be used in sequence out of DSM and if sbyos bit was defined. GPC will wait the “OSCCNT” number of cycles before it notify CCM to open the relevant PLLs. 00000000 count 1 ckil 11111111 count 256 ckils
7 COSC_EN	On-chip oscillator enable bit - this bit value is reflected on the output cosc_en. The system will start with on-chip oscillator enabled to supply source for the PLLs. Software can change this bit if a transition to the bypass PLL clocks was performed for all the PLLs. In cases that this bit is changed from ‘0’ to ‘1’ then GPC will enable the on-chip oscillator and after counting oscnt ckil clock cycles before it notify CCM to open the relevant PLLs . The cosc_en bit should be changed only when on-chip oscillator is not chosen as the clock source. 0 Disable on-chip oscillator 1 Enable on-chip oscillator
6 COSC_PWRDOWN	In run mode, software can manually control powering down of on chip oscillator, i.e. generating ‘1’ on cosc_pwrdown signal. If software manually powered down the on chip oscillator, then sbyos functionality for on-chip oscillator will be bypassed. The manual closing of on-chip oscillator should be performed only in case the reference oscillator is not the source of all the clocks generation. 0 On-chip oscillator will not be powered down, i.e. cosc_pwrdown = 0 1 On-chip oscillator will be powered down, i.e. cosc_pwrdown = 1
5–3 STBY_COUNT	Standby counter definition. These two bits define, in the case of stop exit (if VSTBY bit was set), the amount of time GPC will wait between PMIC_STBY_REQ negation and the check of assertion of PMIC_READY.

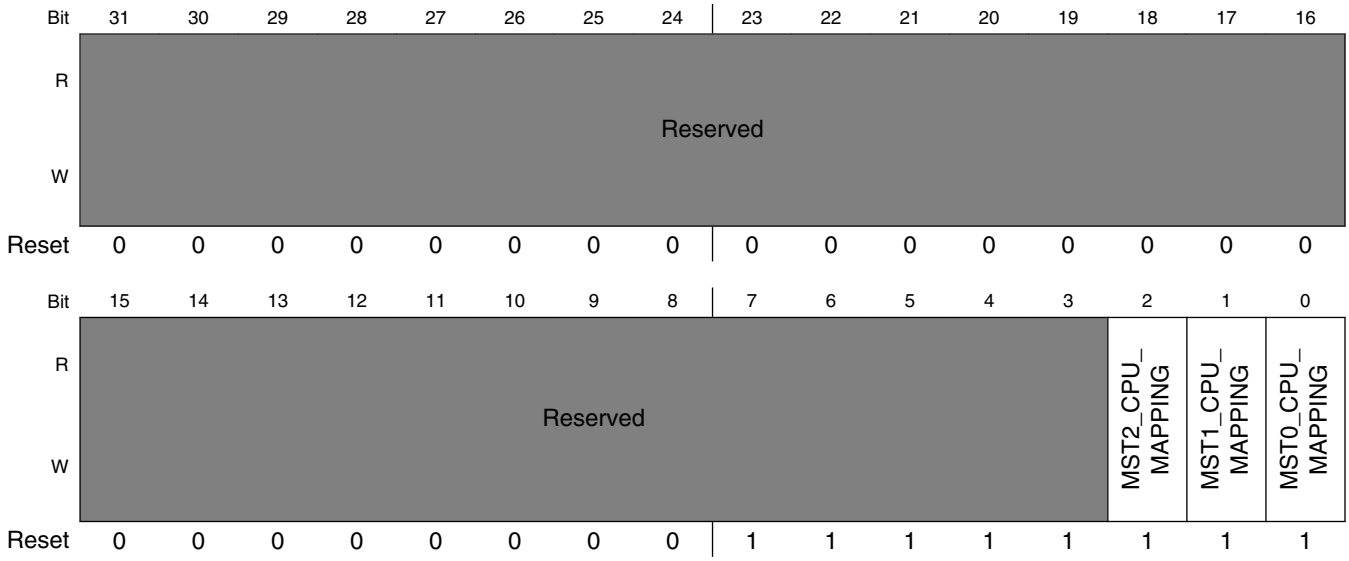
Table continues on the next page...

GPC_SLPCR field descriptions (continued)

Field	Description
	000 GPC will wait 4 ckil clock cycles 001 GPC will wait 8 ckil clock cycles 010 GPC will wait 16 ckil clock cycles 011 GPC will wait 32 ckil clock cycles 100 GPC will wait 64 ckil clock cycles 101 GPC will wait 128 ckil clock cycles 110 GPC will wait 256 ckil clock cycles 111 GPC will wait 512 ckil clock cycles
2 VSTBY	Voltage standby request bit. This bit defines if PMIC_STBY_REQ pin, which notifies external power management IC to move from functional voltage to standby voltage, will be asserted in stop mode. 0 Voltage will not be changed to standby voltage after next entrance to stop mode. (PMIC_STBY_REQ will remain negated - '0') 1 Voltage will be changed to standby voltage after next entrance to stop mode.
1 SBYOS	Standby clock oscillator bit. This bit defines if cosc_pwrdown, which power down the on chip oscillator, will be asserted in DSM. 0 On chip oscillator will not be powered down, after next entrance to DSM. 1 On chip oscillator will be powered down, after next entrance to DSM. When returning from DSM, external oscillator will be enabled again, on chip oscillator will return to oscillator mode , and after oscnt count GPC will continue with the exit from DSM process.
0 BYPASS_PMIC_READY	By asserting this bit GPC will bypass waiting for PMIC_READY signal when coming out of DSM. This should be used for PMIC's that don't support the PMIC_READY signal. 0 Don't bypass the PMIC_READY signal - GPC will wait for its assertion during exit of low power mode if standby voltage was enabled 1 Bypass the PMIC_READY signal - GPC will not wait for its assertion during exit of low power mode if standby voltage was enabled

5.2.10.5 MASTER LPM Handshake (GPC_MST_CPU_MAPPING)

Address: 303A_0000h base + 18h offset = 303A_0018h

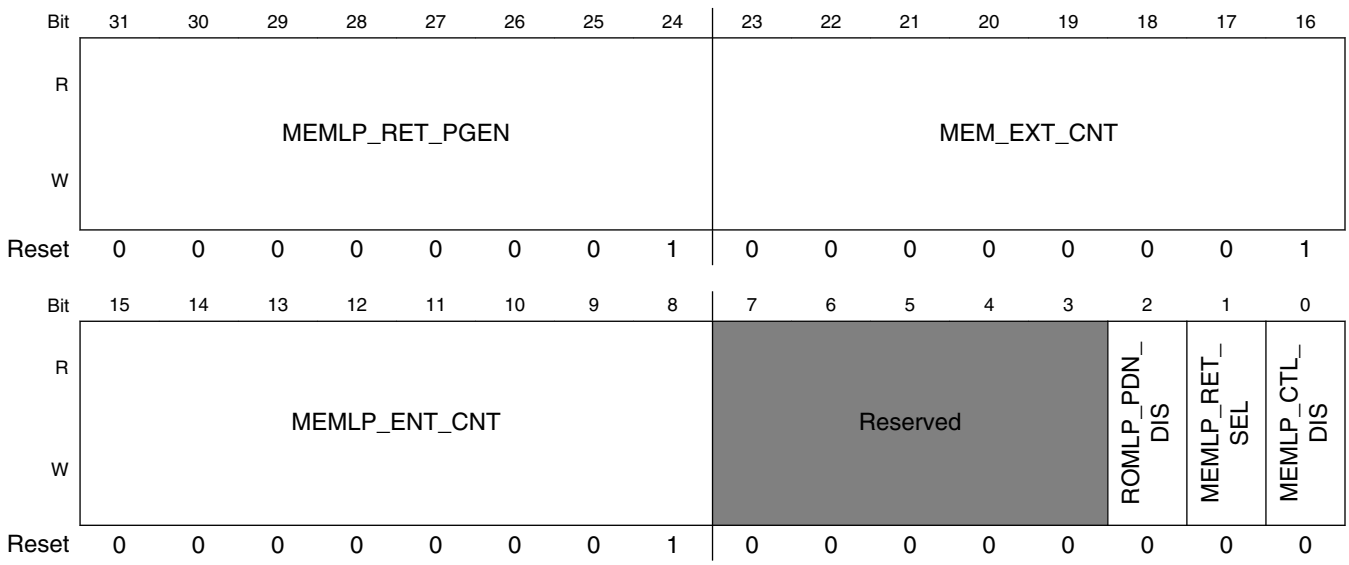


GPC_MST_CPU_MAPPING field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
2 MST2_CPU_MAPPING	MASTER2 CPU Mapping noc2supermix ADB LPM handshake mask. This bit should be used together with LPCR_A53_BSC[8]. If you want power of SCU, use this setting: LPCR_A53_BSC[8]=0; MST_CPU_MAPPING[2]=1 Otherwise, use this: LPCR_A53_BSC[8]=1; MST_CPU_MAPPING[2]=0 0 GPC will not send out power off requirement 1 GPC will send out power off requirement
1 MST1_CPU_MAPPING	MASTER1 CPU Mapping Supermix2noc ADB LPM handshake mask. This bit should be used together with LPCR_A53_BSC[7]. If you want power of SCU, use this setting: LPCR_A53_BSC[7]=0; MST_CPU_MAPPING[1]=1 Otherwise, use this: LPCR_A53_BSC[7]=1; MST_CPU_MAPPING[1]=0 0 GPC will not send out power off requirement 1 GPC will send out power off requirement
0 MST0_CPU_MAPPING	MASTER0 CPU Mapping SCU LPM handshake mask. This bit should be used together with LPCR_A7_BSC[6]. If you want power of SCU, use this setting: LPCR_A7_BSC[6]=0; MST_CPU_MAPPING[0]=1 Otherwise, use this: LPCR_A7_BSC[6]=1; MST_CPU_MAPPING[0]=0 0 GPC will not send out power off requirement 1 GPC will send out power off requirement

5.2.10.6 Memory low power control register (GPC_MLPCR)

Address: 303A_0000h base + 20h offset = 303A_0020h



GPC_MLPCR field descriptions

Field	Description
31–24 MEMLP_RET_PGEN	Delay counter for “retnx” and “pgen”
23–16 MEM_EXT_CNT	Delay counter to start existing from memory low power
15–8 MEMLP_ENT_CNT	Delay counter to make sure all clock off after pll_dis_req is issued by smc
7–3 -	This field is reserved. Reserved
2 ROMLP_PDN_DIS	ROM shut down control 0 Enable ROM shut down control(should also enable RAM low power control); 1 Disable ROM shut down control
1 MEMLP_RET_SEL	Retention select 0 retention mode 2 1 retention mode 1
0 MEMLP_CTL_DIS	RAM low-power control 0 Enable RAM low power control 1 Disable RAM low power control

5.2.10.7 PGC acknowledge signal selection of A53 platform (GPC_PGC_ACK_SEL_A53)

The register can only be accessed by A53 platform

Address: 303A_0000h base + 24h offset = 303A_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	A53_PGC_PUP_ACK	A53_PGC_PDN_ACK	Reserved													
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		NOC_PGC_PUP_ACK	NOC_PGC_PDN_ACK	Reserved		A53_PLAT_PGC_PUP_ACK	A53_PLAT_PGC_PDN_ACK	A53_C3_PGC_PUP_ACK	A53_C3_PGC_PDN_ACK	A53_C2_PGC_PUP_ACK	A53_C2_PGC_PDN_ACK	A53_C1_PGC_PUP_ACK	A53_C1_PGC_PDN_ACK	A53_C0_PGC_PUP_ACK	A53_C0_PGC_PDN_ACK
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_PGC_ACK_SEL_A53 field descriptions

Field	Description
31 A53_PGC_PUP_ACK	Select power up acknowledge signal of A53 (dummy) PGC as the power up acknowledge for A53 LPM.
30 A53_PGC_PDN_ACK	Select power down acknowledge signal of A53 (dummy) PGC as the power down acknowledge for A53 LPM.
29–14 -	This field is reserved. Reserved
13 NOC_PGC_PUP_ACK	Select power down acknowledge signal of NOC PGC as the power up acknowledge for A53 LPM.
12 NOC_PGC_PDN_ACK	Select power down acknowledge signal of NOC PGC as the power down acknowledge for A53 LPM.
11–10 -	This field is reserved. Reserved
9 A53_PLAT_PGC_PUP_ACK	Select power down acknowledge signal of A53 PLATFORM PGC as the power up acknowledge for A53 LPM.

Table continues on the next page...

GPC_PGC_ACK_SEL_A53 field descriptions (continued)

Field	Description
8 A53_PLAT_ PGC_PDN_ACK	Select power down acknowledge signal of A53 PLATFORM PGC as the power down acknowledge for A53 LPM.
7 A53_C3_PGC_ PUP_ACK	Select power down acknowledge signal of A53 CORE3 PGC as the power up acknowledge for A53 LPM.
6 A53_C3_PGC_ PDN_ACK	Select power down acknowledge signal of A53 CORE3 PGC as the power down acknowledge for A53 LPM.
5 A53_C2_PGC_ PUP_ACK	Select power down acknowledge signal of A53 CORE2 PGC as the power up acknowledge for A53 LPM.
4 A53_C2_PGC_ PDN_ACK	Select power down acknowledge signal of A53 CORE2 PGC as the power down acknowledge for A53 LPM.
3 A53_C1_PGC_ PUP_ACK	Select power down acknowledge signal of A53 CORE1 PGC as the power up acknowledge for A53 LPM.
2 A53_C1_PGC_ PDN_ACK	Select power down acknowledge signal of A53 CORE1 PGC as the power down acknowledge for A53 LPM.
1 A53_C0_PGC_ PUP_ACK	Select power down acknowledge signal of A53 CORE0 PGC as the power up acknowledge for A53 LPM.
0 A53_C0_PGC_ PDN_ACK	Select power down acknowledge signal of A53 CORE0 PGC as the power down acknowledge for A53 LPM.

5.2.10.8 PGC acknowledge signal selection of M7 platform (GPC_PGC_ACK_SEL_M7)

This register can only be accessed by the M7 platform.

NOTE

"dummy" PGC cannot be mapped to time slot control. "virtual" PGC can be mapped to time slot control. When virtual PGC is used, below setting is required - GPC_MISC[M7_PDN_REQ_MASK] should be set to 1'b1 and arrange virtual GPC in same slot with MIX. power/up slot (See example code 3). MIX PGC may possibly power down later than A53 platform power down when virtual PGC is used.

Address: 303A_0000h base + 28h offset = 303A_0028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	M7_DUMMY_PGC_PUP_ACK	M7_DUMMY_PGC_PDN_ACK	Reserved													
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		NOC_PGC_PUP_ACK	NOC_PGC_PDN_ACK	Reserved										M7_VIRTUAL_PGC_PUP_ACK	M7_VIRTUAL_PGC_PDN_ACK
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_PGC_ACK_SEL_M7 field descriptions

Field	Description
31 M7_DUMMY_PGC_PUP_ACK	Select power up acknowledge signal of M7 (dummy) PGC as the power up acknowledge for M7 LPM.
30 M7_DUMMY_PGC_PDN_ACK	Select power down acknowledge signal of M7 (dummy) PGC as the power down acknowledge for M7 LPM.

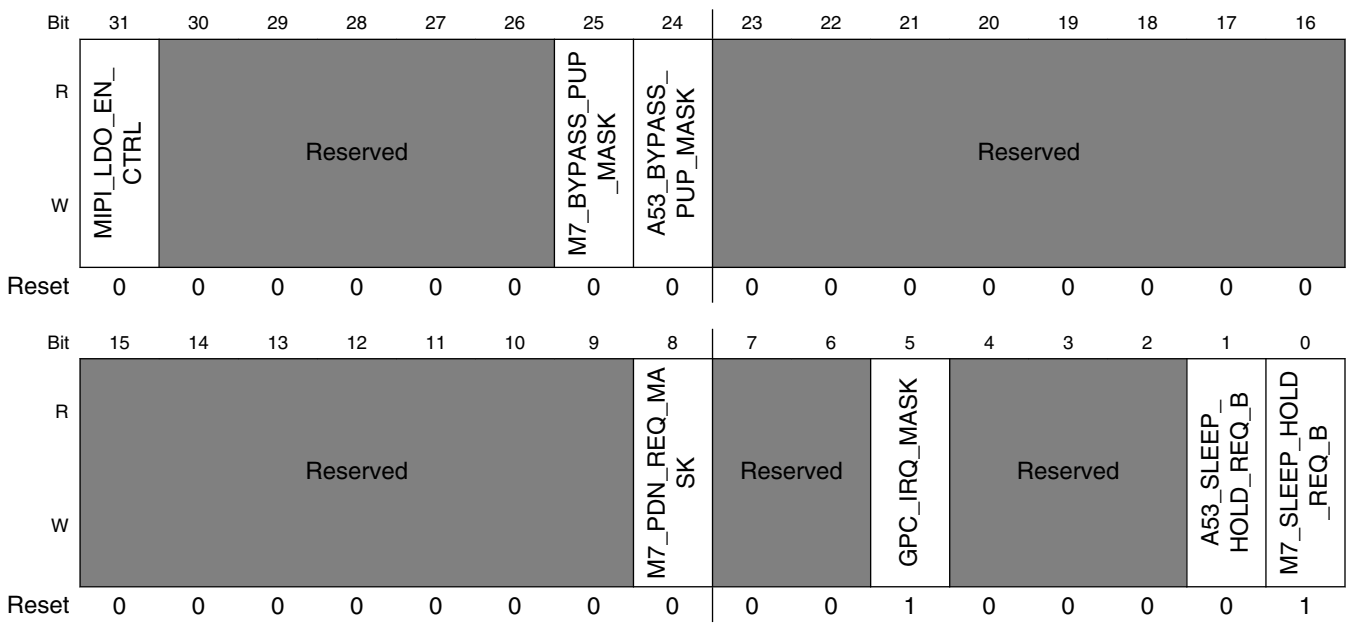
Table continues on the next page...

GPC_PGC_ACK_SEL_M7 field descriptions (continued)

Field	Description
29–14 -	This field is reserved. Reserved
13 NOC_PGC_ PUP_ACK	Select power down acknowledge signal of NOC PGC as the power up acknowledge for M7 LPM.
12 NOC_PGC_ PDN_ACK	Select power down acknowledge signal of NOC PGC as the power down acknowledge for M7 LPM.
11–2 -	This field is reserved. Reserved
1 M7_VIRTUAL_ PGC_PUP_ACK	Select power up acknowledge signal of M7 virtual PGC as the power up acknowledge for M7 LPM. M7 virtual PGC only acknowledge power up request in the end of current slot time
0 M7_VIRTUAL_ PGC_PDN_ACK	Select power down acknowledge signal of M7 virtual PGC as the power down acknowledge for M7 LPM. M7 virtual PGC only acknowledge power down request in the end of current slot time

5.2.10.9 GPC Miscellaneous register (GPC_MISC)

Address: 303A_0000h base + 2Ch offset = 303A_002Ch



GPC_MISC field descriptions

Field	Description
31 MIPI_LDO_EN_CTRL	MIPI LDO enable control. Default is 1'b1.
30–26 -	This field is reserved. Reserved
25 M7_BYPASS_PUP_MASK	
24 A53_BYPASS_PUP_MASK	
23–9 -	This field is reserved. Reserved
8 M7_PDN_REQ_MASK	M7 power-down mask 0 M7 power down request to virtual M7 PGC will be masked. 1 M7 power down request to virtual M7 PGC will not be masked. Set this bit to 1’b1 when M7 virtual PGC is used.
7–6 -	This field is reserved. Reserved
5 GPC_IRQ_MASK	GPC interrupt/event masking 0 Not masked 1 Interrupt / event is masked
4–2 -	This field is reserved. Reserved
1 A53_SLEEP_HOLD_REQ_B	A53 sleep hold 0 Hold A53 platform in sleep mode. This bit is a software control bit to A53 platform. 1 Don’t hold A53 platform in sleep mode.
0 M7_SLEEP_HOLD_REQ_B	M7 sleep hold 0 Hold M7 platform in sleep mode. This bit is a software control bit to M7 platform. 1 Don’t hold M7 platform in sleep mode.

5.2.10.10 IRQ masking register 1 of A53 core0 (GPC_IMR1_CORE0_A53)

The five IMRn_CORE0_A53 (n = 1,2,3,4,5) registers are used as interrupt mask for A53 core0.

Address: 303A_0000h base + 30h offset = 303A_0030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR1_CORE0_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR1_CORE0_A53 field descriptions

Field	Description
IMR1_CORE0_A53	A53 core0 IRQ[31:0] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.11 IRQ masking register 2 of A53 core0 (GPC_IMR2_CORE0_A53)

Address: 303A_0000h base + 34h offset = 303A_0034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR2_CORE0_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR2_CORE0_A53 field descriptions

Field	Description
IMR2_CORE0_A53	A53 core0 IRQ[63:32] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.12 IRQ masking register 3 of A53 core0 (GPC_IMR3_CORE0_A53)

Address: 303A_0000h base + 38h offset = 303A_0038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR3_CORE0_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR3_CORE0_A53 field descriptions

Field	Description
IMR3_CORE0_A53	A53 core0 IRQ[95:64] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.13 IRQ masking register 4 of A53 core0 (GPC_IMR4_CORE0_A53)

Address: 303A_0000h base + 3Ch offset = 303A_003Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR4_CORE0_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR4_CORE0_A53 field descriptions

Field	Description
IMR4_CORE0_A53	A53 core0 IRQ[127:96] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.14 IRQ masking register 5 of A53 core0 (GPC_IMR5_CORE0_A53)

Address: 303A_0000h base + 40h offset = 303A_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR5_CORE0_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR5_CORE0_A53 field descriptions

Field	Description
IMR5_CORE0_A53	A53 core0 IRQ[159:128] masking bits:

GPC_IMR5_CORE0_A53 field descriptions (continued)

Field	Description
	0 IRQ not masked 1 IRQ masked

5.2.10.15 IRQ masking register 1 of A53 core1 (GPC_IMR1_CORE1_A53)

The five IMRn_CORE1_A53 (n = 1,2,3,4,5) registers are used as interrupt mask for A53 core1.

Address: 303A_0000h base + 44h offset = 303A_0044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR1_CORE1_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR1_CORE1_A53 field descriptions

Field	Description
IMR1_CORE1_A53	A53 core1 IRQ[31:0] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.16 IRQ masking register 2 of A53 core1 (GPC_IMR2_CORE1_A53)

Address: 303A_0000h base + 48h offset = 303A_0048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR2_CORE1_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR2_CORE1_A53 field descriptions

Field	Description
IMR2_CORE1_A53	A53 core1 IRQ[63:32] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.17 IRQ masking register 3 of A53 core1 (GPC_IMR3_CORE1_A53)

Address: 303A_0000h base + 4Ch offset = 303A_004Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR3_CORE1_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR3_CORE1_A53 field descriptions

Field	Description
IMR3_CORE1_A53	A53 core1 IRQ[95:64] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.18 IRQ masking register 4 of A53 core1 (GPC_IMR4_CORE1_A53)

Address: 303A_0000h base + 50h offset = 303A_0050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR4_CORE1_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR4_CORE1_A53 field descriptions

Field	Description
IMR4_CORE1_A53	A53 core1 IRQ[127:96] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.19 IRQ masking register 5 of A53 core1 (GPC_IMR5_CORE1_A53)

Address: 303A_0000h base + 54h offset = 303A_0054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR5_CORE1_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR5_CORE1_A53 field descriptions

Field	Description
IMR5_CORE1_A53	A53 core1 IRQ[159:128] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.20 IRQ masking register 1 of M7 (GPC_IMR1_M7)

The five IMRn_M7 (n = 1,2,3,4,5) registers are used as interrupt mask for M7.

Address: 303A_0000h base + 58h offset = 303A_0058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR1_M7																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR1_M7 field descriptions

Field	Description
IMR1_M7	M7 IRQ[31:0] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.21 IRQ masking register 2 of M7 (GPC_IMR2_M7)

Address: 303A_0000h base + 5Ch offset = 303A_005Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR2_M7																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR2_M7 field descriptions

Field	Description
IMR2_M7	M7 IRQ[63:32] masking bits:

GPC_IMR2_M7 field descriptions (continued)

Field	Description
	0 IRQ not masked 1 IRQ masked

5.2.10.22 IRQ masking register 3 of M7 (GPC_IMR3_M7)

Address: 303A_0000h base + 60h offset = 303A_0060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR3_M7																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR3_M7 field descriptions

Field	Description
IMR3_M7	M7 IRQ[95:64] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.23 IRQ masking register 4 of M7 (GPC_IMR4_M7)

Address: 303A_0000h base + 64h offset = 303A_0064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR4_M7																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR4_M7 field descriptions

Field	Description
IMR4_M7	M7 IRQ[127:96] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.24 IRQ masking register 5 of M7 (GPC_IMR5_M7)

Address: 303A_0000h base + 68h offset = 303A_0068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR5_M7																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR5_M7 field descriptions

Field	Description
IMR5_M7	M7 IRQ[159:128] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.25 IRQ status register 1 of A53 (GPC_ISR1_A53)

The five ISRn_A53 (n = 1,2,3,4,5) registers, all of them are read only register

Address: 303A_0000h base + 80h offset = 303A_0080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR1_A53																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR1_A53 field descriptions

Field	Description
ISR1_A53	A53 IRQ[31:0] status

5.2.10.26 IRQ status register 2 of A53 (GPC_ISR2_A53)

Address: 303A_0000h base + 84h offset = 303A_0084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR2_A53																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR2_A53 field descriptions

Field	Description
ISR2_A53	A53 IRQ[63:32] status

5.2.10.27 IRQ status register 3 of A53 (GPC_ISR3_A53)

Address: 303A_0000h base + 88h offset = 303A_0088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR3_A53																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR3_A53 field descriptions

Field	Description
ISR3_A53	A53 IRQ[95:64] status

5.2.10.28 IRQ status register 4 of A53 (GPC_ISR4_A53)

Address: 303A_0000h base + 8Ch offset = 303A_008Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR4_A53																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR4_A53 field descriptions

Field	Description
ISR4_A53	A53 IRQ[127:96] status

5.2.10.29 IRQ status register 5 of A53 (GPC_ISR5_A53)

Address: 303A_0000h base + 90h offset = 303A_0090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR5_A53																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR5_A53 field descriptions

Field	Description
ISR5_A53	A53 IRQ[159:128] status

5.2.10.30 IRQ status register 1 of M7 (GPC_ISR1_M7)

The five ISRn_M7 (n = 1,2,3,4,5) registers, all of them are read only register

Address: 303A_0000h base + 94h offset = 303A_0094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR1_M7																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR1_M7 field descriptions

Field	Description
ISR1_M7	M7 IRQ[31:0] status

5.2.10.31 IRQ status register 2 of M7 (GPC_ISR2_M7)

Address: 303A_0000h base + 98h offset = 303A_0098h

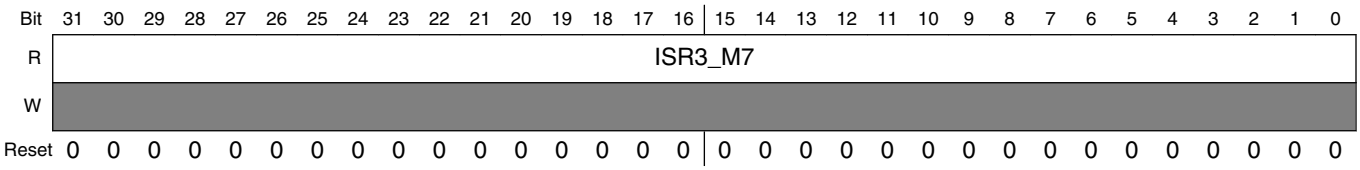
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR2_M7																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR2_M7 field descriptions

Field	Description
ISR2_M7	M7 IRQ[63:32] status

5.2.10.32 IRQ status register 3 of M7 (GPC_ISR3_M7)

Address: 303A_0000h base + 9Ch offset = 303A_009Ch

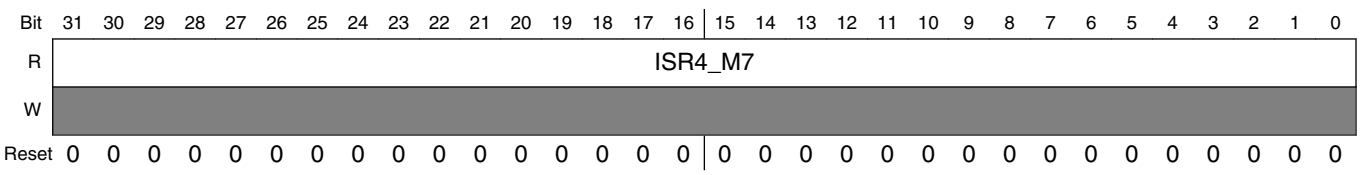


GPC_ISR3_M7 field descriptions

Field	Description
ISR3_M7	M7 IRQ[95:64] status

5.2.10.33 IRQ status register 4 of M7 (GPC_ISR4_M7)

Address: 303A_0000h base + A0h offset = 303A_00A0h

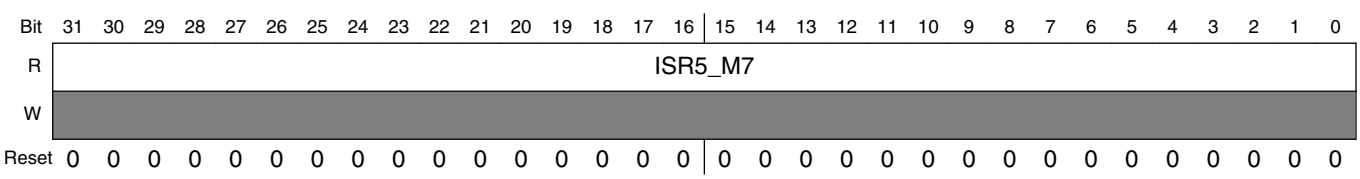


GPC_ISR4_M7 field descriptions

Field	Description
ISR4_M7	M7 IRQ[127:96] status

5.2.10.34 IRQ status register 5 of M7 (GPC_ISR5_M7)

Address: 303A_0000h base + A4h offset = 303A_00A4h

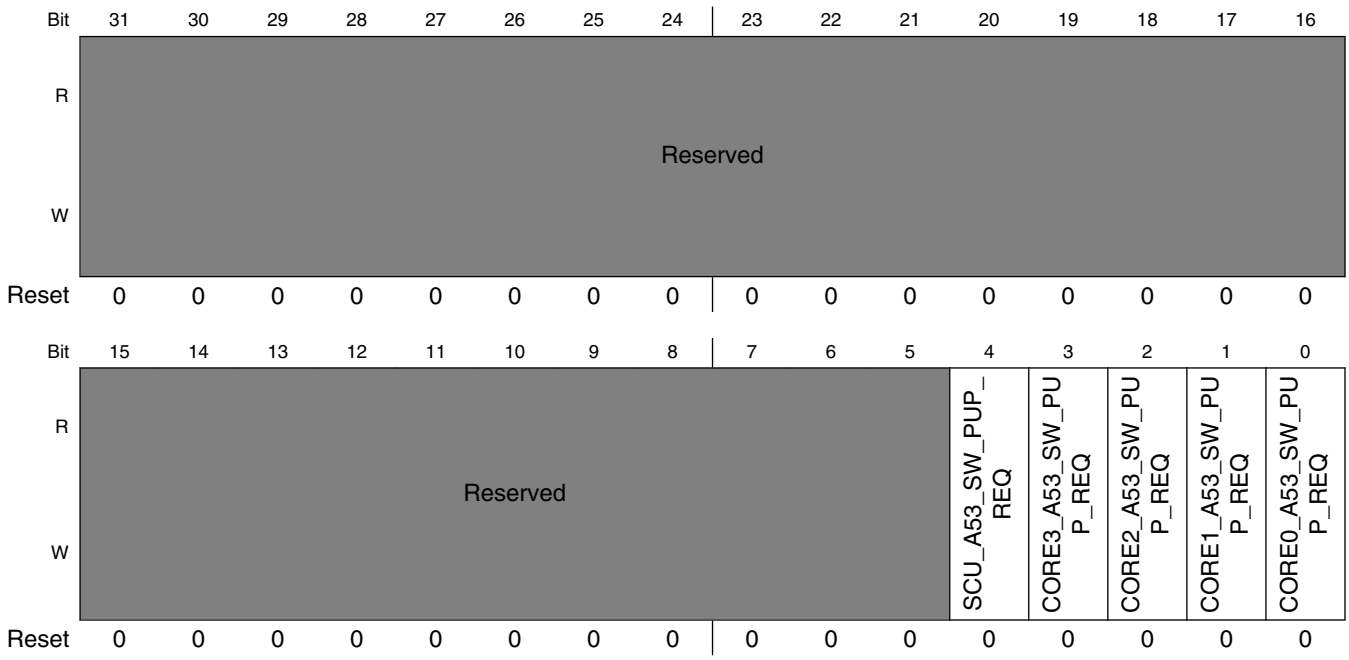


GPC_ISR5_M7 field descriptions

Field	Description
ISR5_M7	M7 IRQ[159:128] status

5.2.10.35 CPU PGC software power up trigger (GPC_CPU_PGC_SW_PUP_REQ)

Address: 303A_0000h base + D0h offset = 303A_00D0h

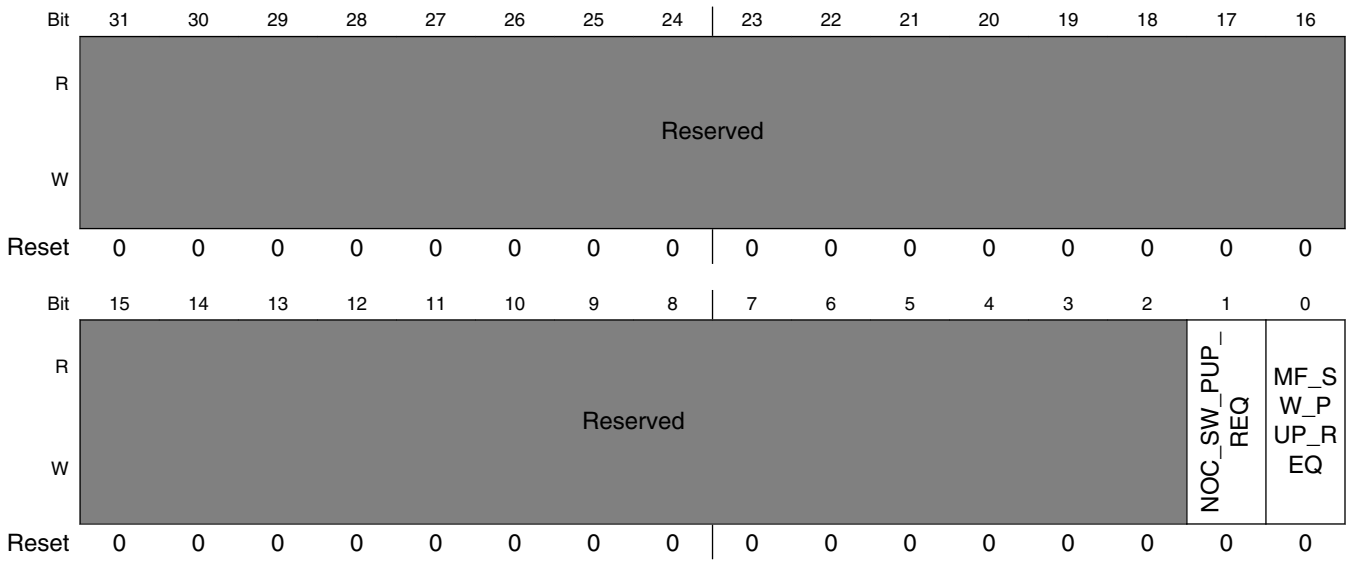


GPC_CPU_PGC_SW_PUP_REQ field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SCU_A53_SW_PUP_REQ	Software power up trigger for SCU A53 PGC
3 CORE3_A53_SW_PUP_REQ	Software power up trigger for Core3 A53 PGC
2 CORE2_A53_SW_PUP_REQ	Software power up trigger for Core2 A53
1 CORE1_A53_SW_PUP_REQ	Software power up trigger for Core1 A53 PGC
0 CORE0_A53_SW_PUP_REQ	Software power up trigger for Core0 A53 PGC

5.2.10.36 MIX PGC software power up trigger (GPC_MIX_PGC_SW_PUP_REQ)

Address: 303A_0000h base + D4h offset = 303A_00D4h

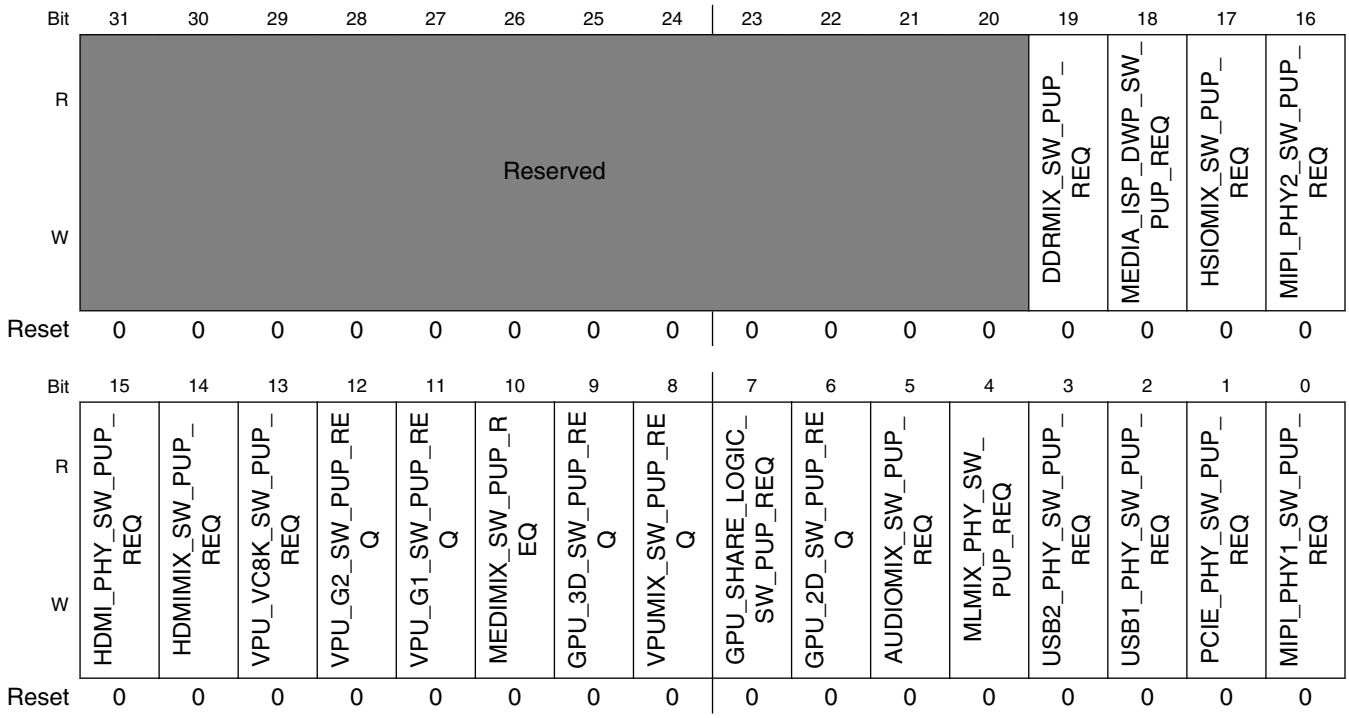


GPC_MIX_PGC_SW_PUP_REQ field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
1 NOC_SW_PUP_REQ	Software power up trigger for NOC PGC
0 MF_SW_PUP_REQ	Software power up trigger for MIX PGC

5.2.10.37 PU PGC software up trigger (GPC_PU_PGC_SW_PUP_REQ)

Address: 303A_0000h base + D8h offset = 303A_00D8h



GPC_PU_PGC_SW_PUP_REQ field descriptions

Field	Description
31–20 -	This field is reserved. Reserved
19 DDRMIX_SW_PUP_REQ	Software power up trigger for DDRMIX
18 MEDIA_ISP_DWP_SW_PUP_REQ	Software power up trigger for MEDIA_ISP_DWP
17 HSIOMIX_SW_PUP_REQ	Software power up trigger for HSIOMIX
16 MIPI_PHY2_SW_PUP_REQ	Software power up trigger for MIPI_PHY2
15 HDMI_PHY_SW_PUP_REQ	Software power up trigger for HDMI_PHY
14 HDMIMIX_SW_PUP_REQ	Software power up trigger for HDMIMIX
13 VPU_VC8K_SW_PUP_REQ	Software power up trigger for VPU_VC8K

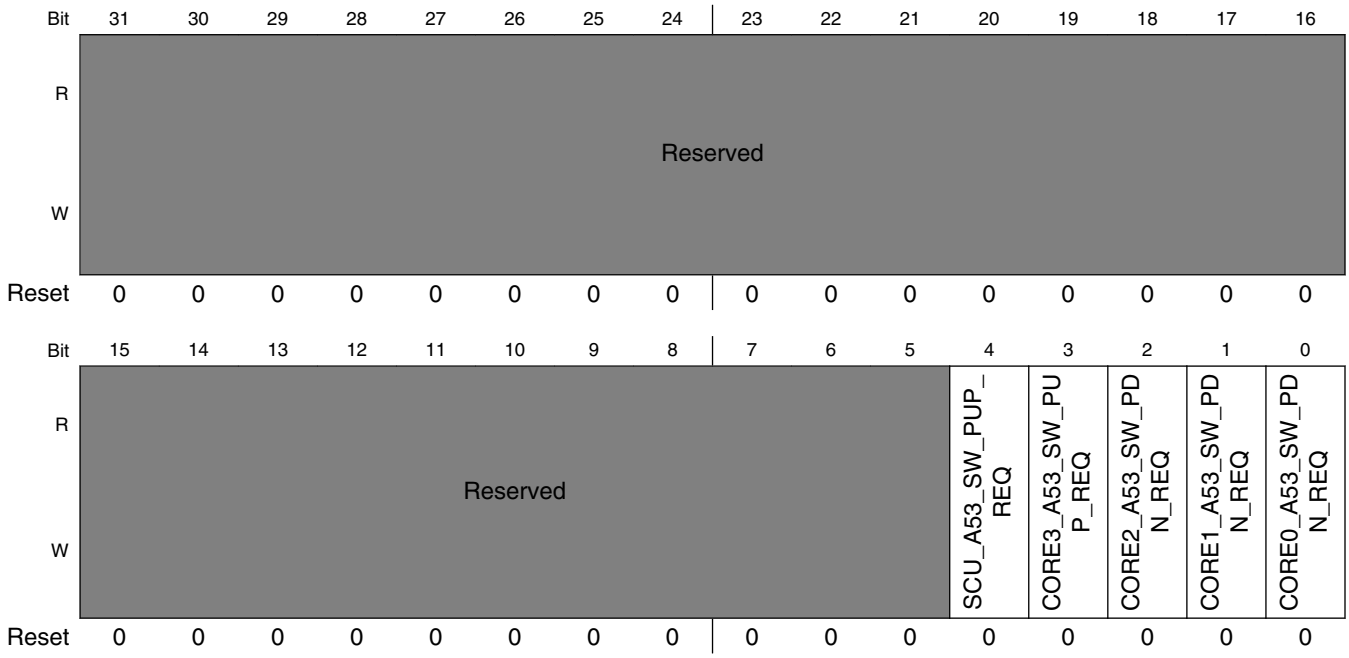
Table continues on the next page...

GPC_PU_PGC_SW_PUP_REQ field descriptions (continued)

Field	Description
12 VPU_G2_SW_PUP_REQ	Software power up trigger for VPU_G2
11 VPU_G1_SW_PUP_REQ	Software power up trigger for VPU_G1
10 MEDIMIX_SW_PUP_REQ	Software power up trigger for MEDIMIX
9 GPU_3D_SW_PUP_REQ	Software power up trigger for GPU_3D
8 VPUMIX_SW_PUP_REQ	Software power up trigger for VPUMIX
7 GPU_SHARE_LOGIC_SW_PUP_REQ	Software power up trigger for GPU_SHARE_LOGIC
6 GPU_2D_SW_PUP_REQ	Software power up trigger for GPU_2D
5 AUDIOMIX_SW_PUP_REQ	Software power up trigger for AUDIOMIX
4 MLMIX_PHY_SW_PUP_REQ	Software power up trigger for MLMIX
3 USB2_PHY_SW_PUP_REQ	Software power up trigger for USB2_PHY
2 USB1_PHY_SW_PUP_REQ	Software power up trigger for USB1_PHY
1 PCIE_PHY_SW_PUP_REQ	Software power up trigger for PCIE_PHY
0 MIPI_PHY1_SW_PUP_REQ	Software power up trigger for MIPI_PHY1

5.2.10.38 CPU PGC software down trigger (GPC_CPU_PGC_SW_PDN_REQ)

Address: 303A_0000h base + DCh offset = 303A_00DCh

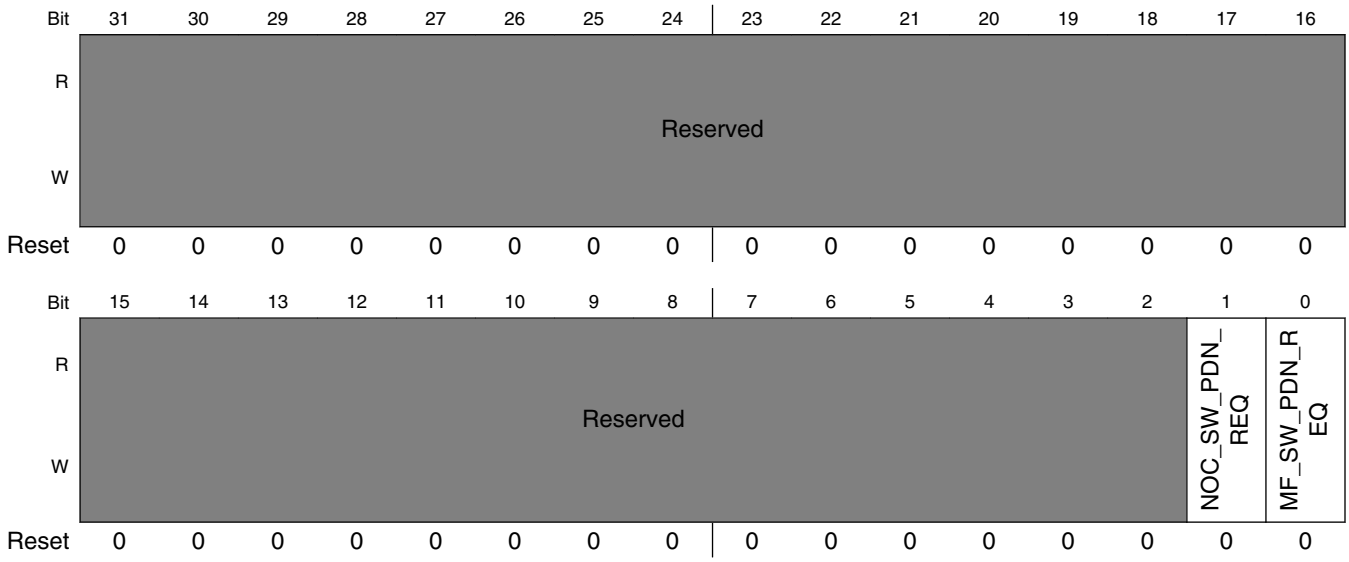


GPC_CPU_PGC_SW_PDN_REQ field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SCU_A53_SW_PUP_REQ	Software power up trigger for SCU A53 PGC
3 CORE3_A53_SW_PUP_REQ	Software power up trigger for Core3 A53 PGC
2 CORE2_A53_SW_PDN_REQ	Software power down trigger for Core2 A53 PGC
1 CORE1_A53_SW_PDN_REQ	Software power down trigger for Core1 A53 PGC
0 CORE0_A53_SW_PDN_REQ	Software power down trigger for Core0 A53 PGC

5.2.10.39 MIX PGC software power down trigger (GPC_MIX_PGC_SW_PDN_REQ)

Address: 303A_0000h base + E0h offset = 303A_00E0h

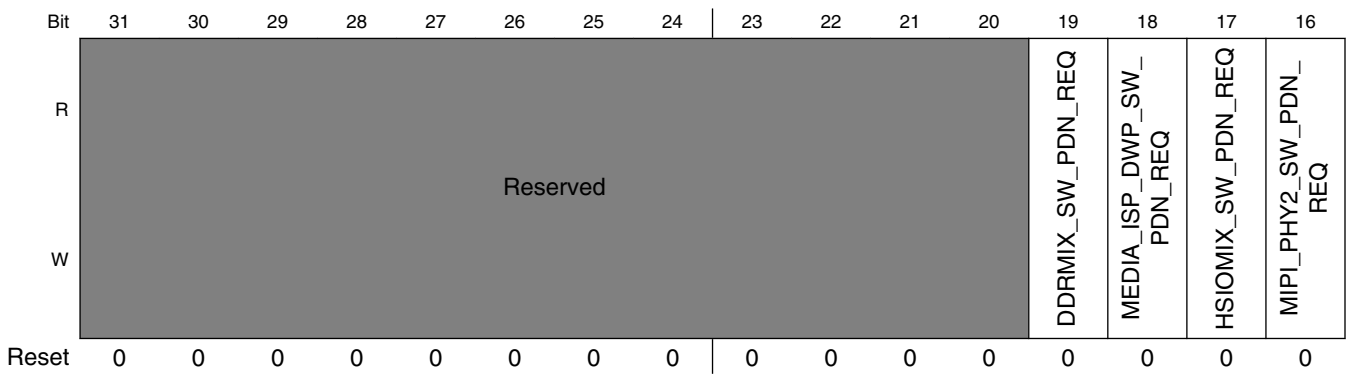


GPC_MIX_PGC_SW_PDN_REQ field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
1 NOC_SW_PDN_REQ	Software power down trigger for NOC PGC
0 MF_SW_PDN_REQ	Software power down trigger for MIX PGC

5.2.10.40 PU PGC software down trigger (GPC_PU_PGC_SW_PDN_REQ)

Address: 303A_0000h base + E4h offset = 303A_00E4h



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HDMI_PHY_SW_PDN_REQ	HDMIMIX_SW_PDN_REQ	VPU_VC8K_SW_PDN_REQ	_SW_PDN_REQ	VPU_G1_SW_PDN_REQ	MEDIMIX_SW_PDN_REQ	GPU_3D_SW_PDN_REQ	VPUMIX_SHARE_LOGIC_SW_PDN_REQ	GPU_SHARE_LOGIC_SW_PDN_REQ	GPU_2D_SW_PDN_REQ	AUDIOMIX_SW_PDN_REQ	MLMIX_SW_PDN_REQ	USB2_PHY_SW_PDN_REQ	USB1_PHY_SW_PDN_REQ	PCIE_PHY_SW_PDN_REQ	MIPI_PHY1_SW_PDN_REQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_PU_PGC_SW_PDN_REQ field descriptions

Field	Description
31–20 -	This field is reserved. Reserved
19 DDRMIX_SW_PDN_REQ	Software power down trigger for DDRMIX
18 MEDIA_ISP_DWP_SW_PDN_REQ	Software power down trigger for MEDIA_ISP_DWP
17 HSIOMIX_SW_PDN_REQ	Software power down trigger for HSIOMIX
16 MIPI_PHY2_SW_PDN_REQ	Software power down trigger for MIPI_PHY2
15 HDMI_PHY_SW_PDN_REQ	Software power down trigger for HDMI_PHY
14 HDMIMIX_SW_PDN_REQ	Software power down trigger for HDMIMIX
13 VPU_VC8K_SW_PDN_REQ	Software power down trigger for VPU_VC8K
12 _SW_PDN_REQ	Software power down trigger for VPU_G2
11 VPU_G1_SW_PDN_REQ	Software power down trigger for VPU_G1
10 MEDIMIX_SW_PDN_REQ	Software power down trigger for MEDIMIX
9 GPU_3D_SW_PDN_REQ	Software power down trigger for GPU_3D

Table continues on the next page...

GPC_PU_PGC_SW_PDN_REQ field descriptions (continued)

Field	Description
8 VPUMIX_SHARE_LOGIC_SW_PDN_REQ	Software power down trigger for VPUMIX_SHARE_LOGIC
7 GPU_SHARE_LOGIC_SW_PDN_REQ	Software power down trigger for GPU_SHARE_LOGIC
6 GPU_2D_SW_PDN_REQ	Software power down trigger for GPU_2D
5 AUDIOMIX_SW_PDN_REQ	Software power down trigger for AUDIOMIX
4 MLMIX_SW_PDN_REQ	Software power down trigger for MLMIX
3 USB2_PHY_SW_PDN_REQ	Software power down trigger for USB2_PHY
2 USB1_PHY_SW_PDN_REQ	Software power down trigger for USB1_PHY
1 PCIE_PHY_SW_PDN_REQ	Software power down trigger for PCIE_PHY
0 MIPI_PHY1_SW_PDN_REQ	Software power down trigger for MIPI_PHY1

5.2.10.41 Advanced Low power control register of CM4 platform (GPC_GPC_LPCR_CM4_AD)

Address: 303A_0000h base + ECh offset = 303A_00ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	m4_sleep_hold_req_b	cm4_sleep_hold_ack_b_sync1	cm4_gate_hclk_sync1	cm4_sleep_deep_sync1	cm4_sleeping_sync1	cm4_lookup_sync1	cm4_is_halted_sync1	src_cm4_plat_rst_b	src_cm4_core_rst_b	Reserved						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_GPC_LPCR_CM4_AD field descriptions

Field	Description
31 m4_sleep_hold_req_b	cm4 sleep hold requested
30 cm4_sleep_hold_ack_b_sync1	cm4 sleep hold acknowledged
29 cm4_gate_hclk_sync1	cm4 hclk gated
28 cm4_sleep_deep_sync1	cm4 in deep sleep now
27 cm4_sleeping_sync1	cm4 sleeping now
26 cm4_lookup_sync1	cm4 is lookup
25 cm4_is_halted_sync1	cm4 is halted

Table continues on the next page...

GPC_GPC_LPCR_CM4_AD field descriptions (continued)

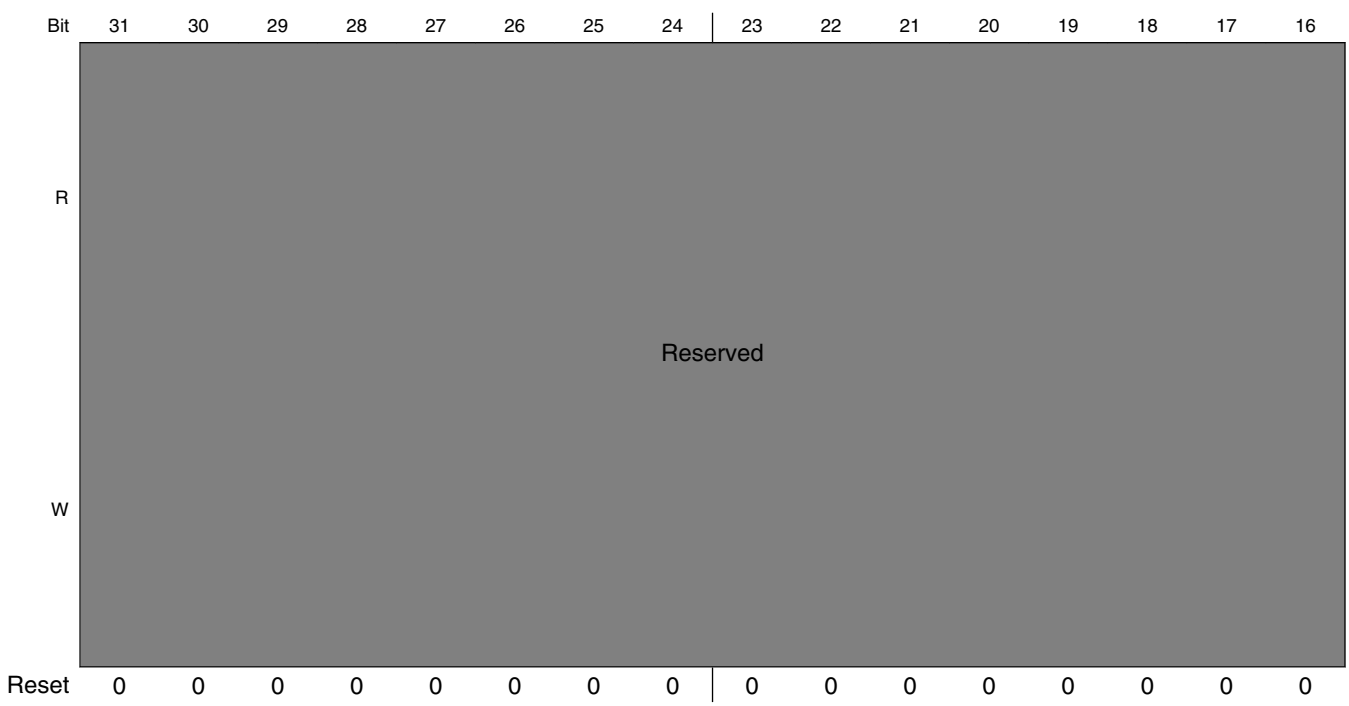
Field	Description
24 src_cm4_plat_rst_b	cm4_plat rst_b assert
23 src_cm4_core_rst_b	cm4_core rst_b assert
-	This field is reserved. Reserved

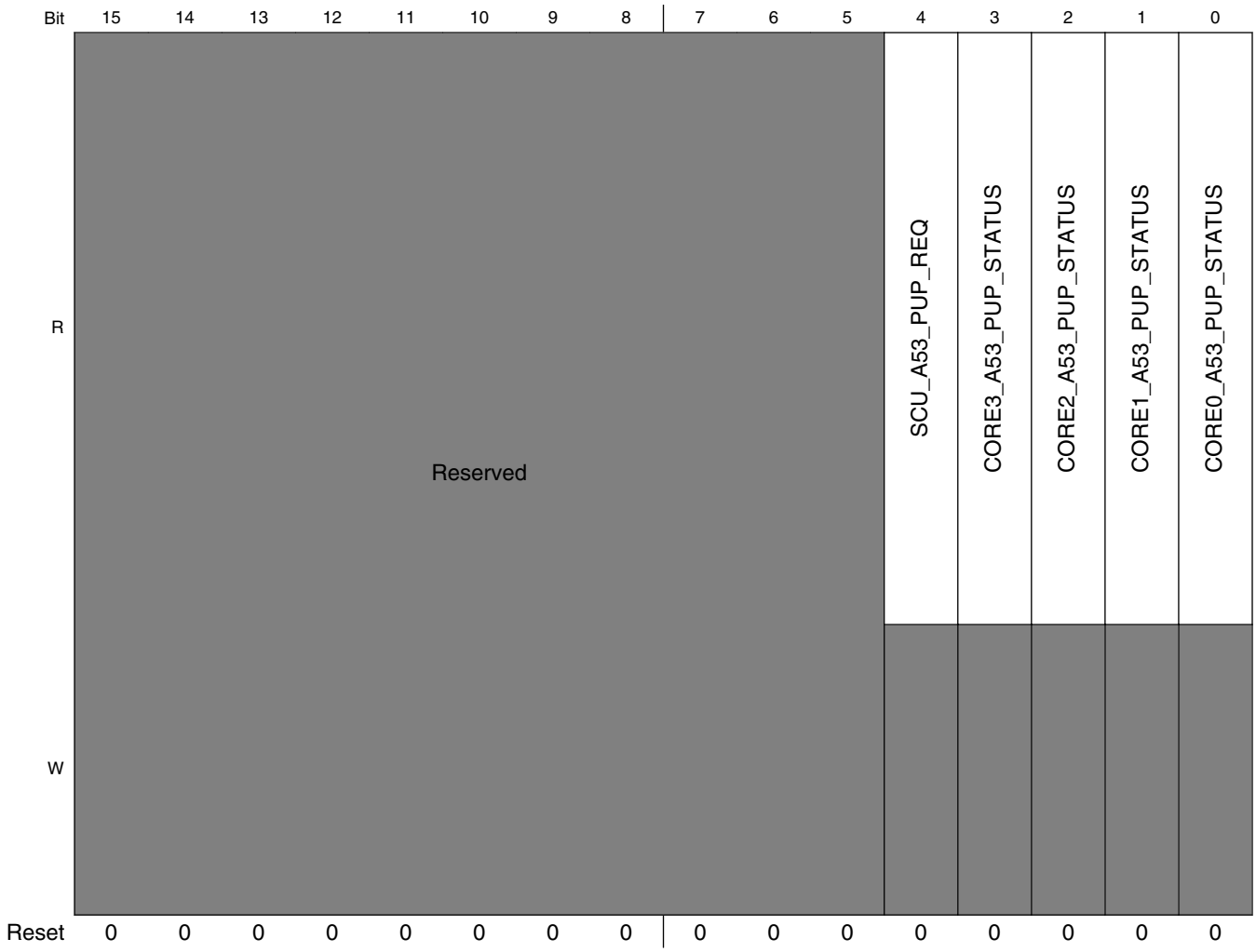
5.2.10.42 CPU PGC software up trigger status1 (GPC_CPU_PGC_PUP_STATUS1)

CPU_PGC_PUP_STATUS1 is a read only register, represents the results for power up software trigger for CPU type PGCs.

The field description is show in table below, the value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power down process. The relevant bit will be cleared after a success operation of power up software trigger for CPU type PGCs.

Address: 303A_0000h base + 108h offset = 303A_0108h





GPC_CPU_PGC_PUP_STATUS1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SCU_A53_PUP_REQ	
3 CORE3_A53_PUP_STATUS	
2 CORE2_A53_PUP_STATUS	
1 CORE1_A53_PUP_STATUS	
0 CORE0_A53_PUP_STATUS	

5.2.10.43 A53 MIX software up trigger status register (GPC_A53_MIX_PGC_PUP_STATUS*n*)

A53_MIX_PGC_PUP_STATUSn (n = 0,1,2) are a read only register, represents the results for power up software trigger from A53 platform to MIX type PGCs.

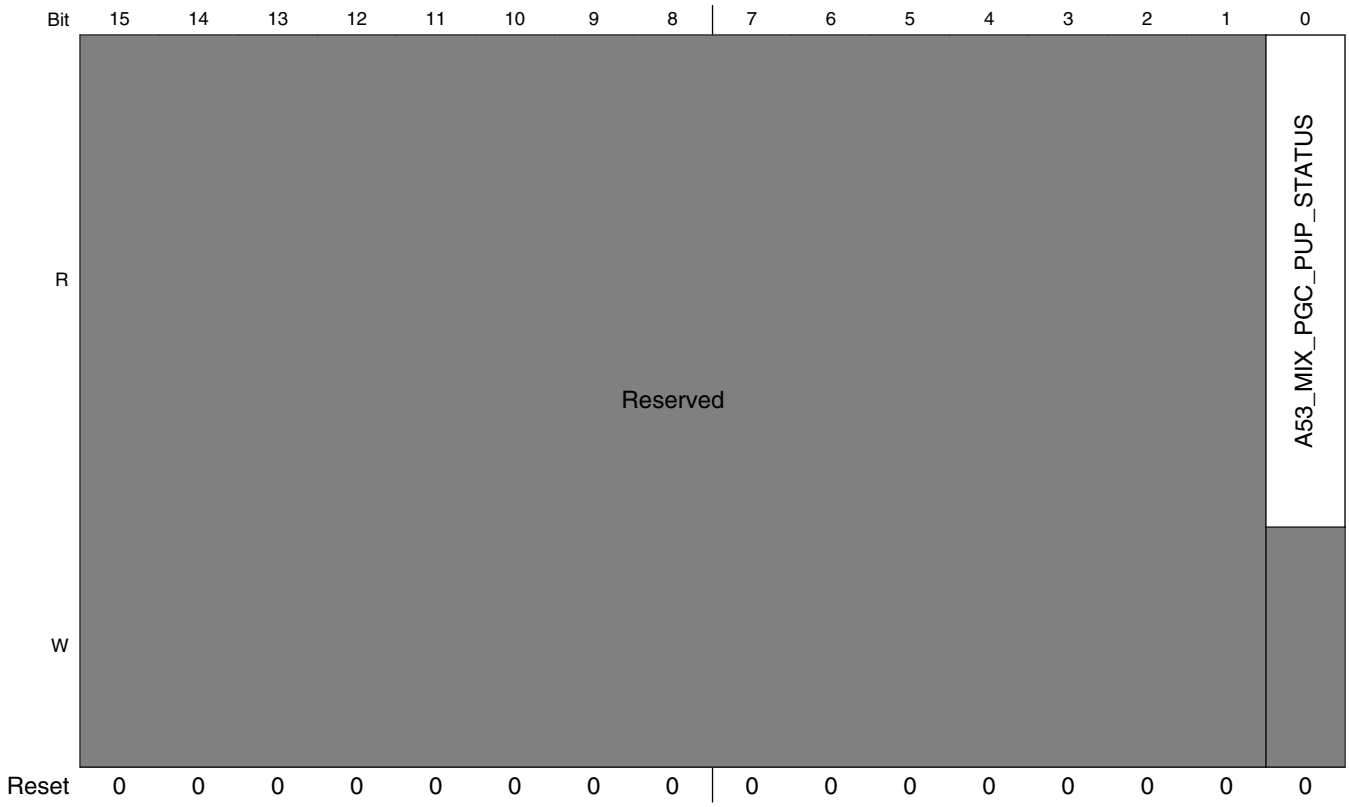
A53_MIX_PGC_PUP_STATUS0: value of “1’b1” represent the software power up trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power up software trigger for MIX type PGCs.

A53_MIX_PGC_PUP_STATUS1: value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power up process. The relevant bit will be cleared after a success operation of power up software trigger for MIX type PGCs.

A53_MIX_PGC_PUP_STATUS2: value of “1’b1” represent the software power up trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power up software trigger for MIX type PGCs.

Address: 303A_0000h base + 10Ch offset + (4d × i), where i=0d to 2d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



GPC_A53_MIX_PGC_PUP_STATUS*n* field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 A53_MIX_PGC_PUP_STATUS	

5.2.10.44 M7 MIX PGC software up trigger status register (GPC_M7_MIX_PGC_PUP_STATUS*n*)

M7_MIX_PGC_PUP_STATUSn (n = 0,1,2) are a read only register, represents the results for power up software trigger from M7 platform to MIX type PGCs.

M7_MIX_PGC_PUP_STATUS0: value of “1’b1” represent the software power up trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power up software trigger for MIX type PGCs.

M7_MIX_PGC_PUP_STATUS1: value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power up process. The relevant bit will be cleared after a success operation of power up software trigger for MIX type PGCs.

M7_MIX_PGC_PUP_STATUS2: value of “1’b1” represent the software power up trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power up software trigger for MIX type PGCs.

Address: 303A_0000h base + 118h offset + (4d × i), where i=0d to 2d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															M7_MIX_PGC_PUP_STATUS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_M7_MIX_PGC_PUP_STATUS*n* field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 M7_MIX_PGC_ PUP_STATUS	

5.2.10.45 A53 PU software up trigger status register (GPC_A53_PU_PGC_PUP_STATUS*n*)

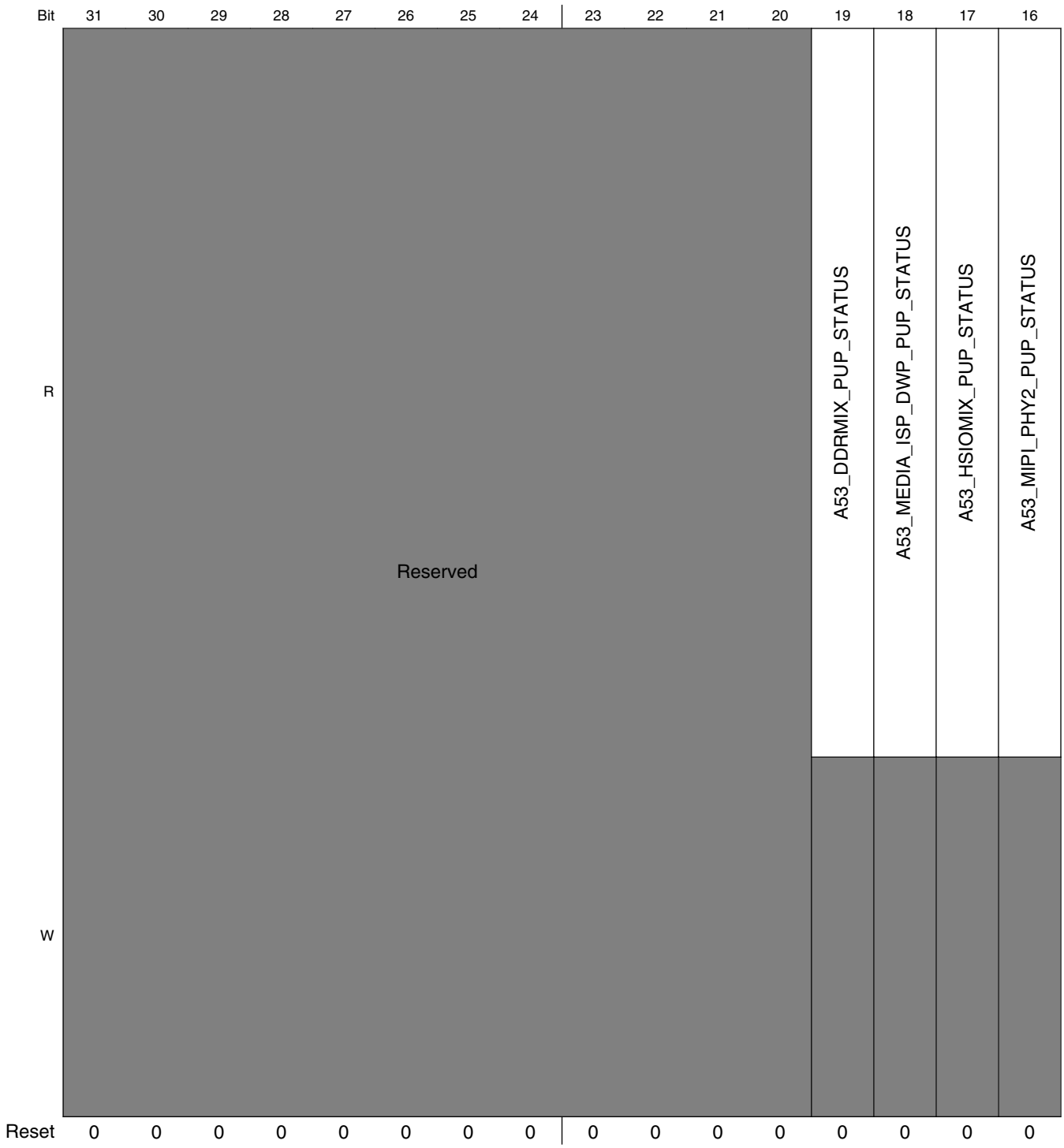
A53_PU_PGC_PUP_STATUSn (n = 0,1,2) are a read only register, represents the results for power up software trigger from A53 platform to PU type PGCs.

A53_PU_PGC_PUP_STATUS0: value of “1’b1” represent the software power up trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power up software trigger for PU type PGCs.

A53_PU_PGC_PUP_STATUS1: value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power up process. The relevant bit will be cleared after a success operation of power up software trigger for PU type PGCs.

A53_PU_PGC_PUP_STATUS2: value of “1’b1” represent the software power up trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power up software trigger for PU type PGCs.

Address: 303A_0000h base + 124h offset + (4d × i), where i=0d to 2d



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	A53_HDMI_PHY_PUP_STATUS	A53_HDMIMIX_PUP_STATUS	A53_VPU_VC8K_PUP_STATUS	A53_VPU_G2_PUP_STATUS	A53_VPU_G1_PUP_STATUS	A53_MEDIMIX_PUP_STATUS	A53_GPU_3D_PUP_STATUS	A53_VPUMIX_SHARE_LOGIC_PUP_STATUS	A53_GPU_SHARE_LOGIC_PUP_STATUS	A53_GPU_2D_PUP_STATUS	A53_AUDIOMIX_PUP_STATUS	A53_MLMIX_PUP_STATUS	A53_USB2_PHY_PUP_STATUS	A53_USB1_PHY_PUP_STATUS	A53_PCIE_PHY_PUP_STATUS	A53_MIPI_PHY1_PUP_STATUS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_A53_PU_PGC_PUP_STATUS*n* field descriptions

Field	Description
31–20 -	This field is reserved. Reserved

Table continues on the next page...

GPC_A53_PU_PGC_PUP_STATUS*n* field descriptions (continued)

Field	Description
19 A53_DDRMIX_ PUP_STATUS	
18 A53_MEDIA_ ISP_DWP_PUP_ STATUS	
17 A53_HSIOMIX_ PUP_STATUS	
16 A53_MIPI_ PHY2_PUP_ STATUS	
15 A53_HDMI_ PHY_PUP_ STATUS	
14 A53_HDMIMIX_ PUP_STATUS	
13 A53_VPU_VC8K _PUP_STATUS	
12 A53_VPU_G2_P UP_STATUS	
11 A53_VPU_G1_P UP_STATUS	
10 A53_MEDIMIX_P UP_STATUS	
9 A53_GPU_3D_P UP_STATUS	
8 A53_VPUMIX_S HARE_LOGIC_P UP_STATUS	
7 A53_GPU_SHAR E_LOGIC_PUP_ STATUS	
6 A53_GPU_2D_ PUP_STATUS	

Table continues on the next page...

GPC_A53_PU_PGC_PUP_STATUS*n* field descriptions (continued)

Field	Description
5 A53_AUDIOMIX_ PUP_STATUS	
4 A53_MLMIX_ PUP_STATUS	
3 A53_USB2_PHY _PUP_STATUS	
2 A53_USB1_PHY _PUP_STATUS	
1 A53_PCIE_PHY_ PUP_STATUS	
0 A53_MIPI_PHY1 _PUP_STATUS	

5.2.10.46 M7 PU PGC software up trigger status register (GPC_M7_PU_PGC_PUP_STATUS*n*)

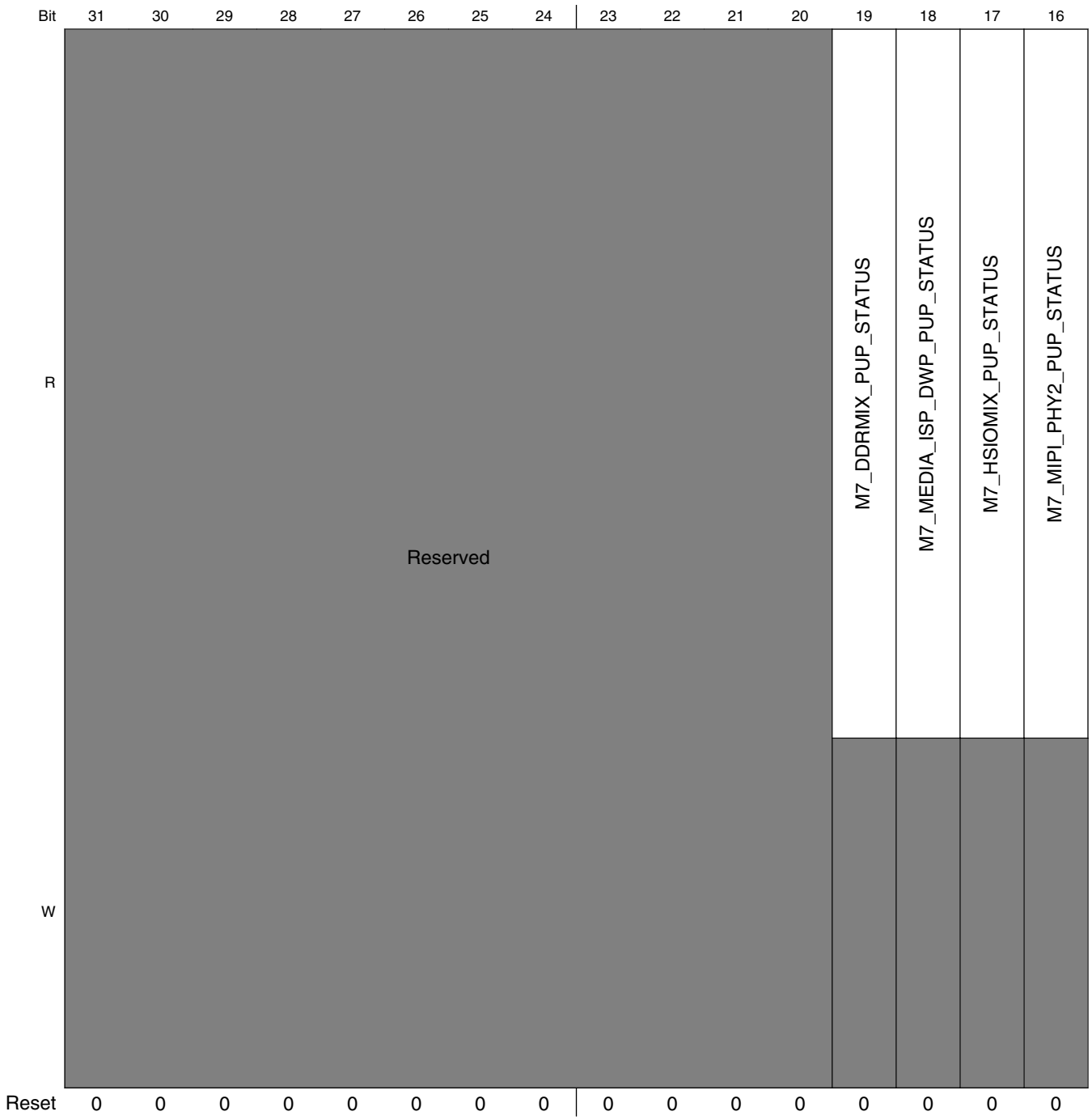
M7_PU_PGC_PUP_STATUSn (n = 0,1,2) are a read only register, represents the results for power up software trigger from M7 platform to PU type PGCs.

M7_PU_PGC_PUP_STATUS0: value of “1’b1” represent the software power up trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power up software trigger for PU type PGCs.

M7_PU_PGC_PUP_STATUS1: value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power up process. The relevant bit will be cleared after a success operation of power up software trigger for PU type PGCs.

M7_PU_PGC_PUP_STATUS2: value of “1’b1” represent the software power up trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power up software trigger for PU type PGCs.

Address: 303A_0000h base + 130h offset + (4d × i), where i=0d to 2d



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M7_HDMI_PHY_PUP_STATUS	M7_HDMIMIX_PUP_STATUS	M7_VPU_VC8K_PUP_STATUS	M7_VPU_G2_PUP_STATUS	M7_VPU_G1_PUP_STATUS	M7_MEDIMIX_PUP_STATUS	M7_GPU3D_PUP_STATUS	M7_VPUMIX_SHARE_LOGIC_PUP_STATUS	M7_GPU_SHARE_LOGIC_PUP_STATUS	M7_GPU2D_PUP_STATUS	M7_AUDIOMIX_PUP_STATUS	M7_MLMIX_PUP_STATUS	M7_USB2_PHY_PUP_STATUS	M7_USB1_PHY_PUP_STATUS	M7_PCIE_PHY_PUP_STATUS	M7_MIPI_PHY1_PUP_STATUS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_M7_PU_PGC_PUP_STATUS*n* field descriptions

Field	Description
31–20 -	This field is reserved. Reserved
19 M7_DDRMIX_ PUP_STATUS	

Table continues on the next page...

GPC_M7_PU_PGC_PUP_STATUS*n* field descriptions (continued)

Field	Description
18 M7_MEDIA_ISP_DWP_PUP_STATUS	
17 M7_HSIOMIX_PUP_STATUS	
16 M7_MIPI_PHY2_PUP_STATUS	
15 M7_HDMI_PHY_PUP_STATUS	
14 M7_HDMIMIX_PUP_STATUS	
13 M7_VPU_VC8K_PUP_STATUS	
12 M7_VPU_G2_PUP_STATUS	
11 M7_VPU_G1_PUP_STATUS	
10 M7_MEDIMIX_PUP_STATUS	
9 M7_GPU3D_PUP_STATUS	
8 M7_VPUMIX_SHARE_LOGIC_PUP_STATUS	
7 M7_GPU_SHARE_LOGIC_PUP_STATUS	
6 M7_GPU2D_PUP_STATUS	
5 M7_AUDIOMIX_PUP_STATUS	
4 M7_MLMIX_PUP_STATUS	

Table continues on the next page...

GPC_M7_PU_PGC_PUP_STATUS*n* field descriptions (continued)

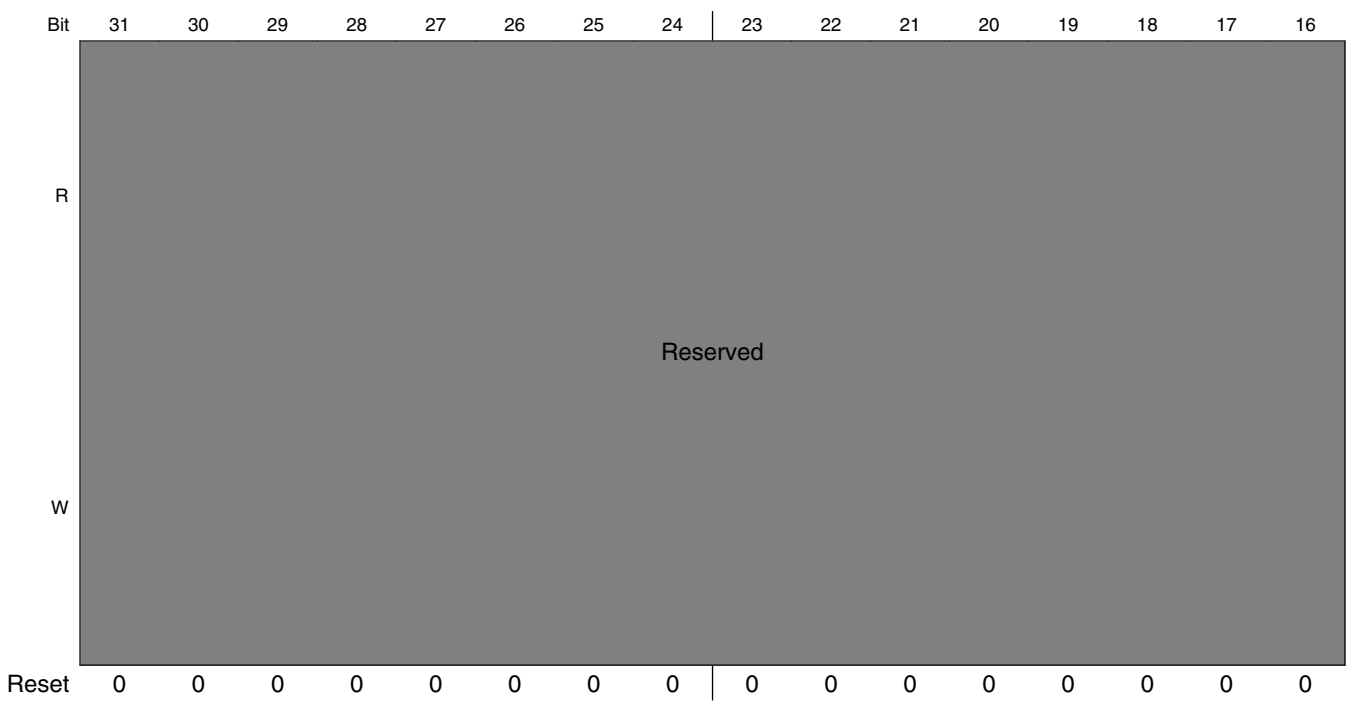
Field	Description
3 M7_USB2_PHY_PUP_STATUS	
2 M7_USB1_PHY_PUP_STATUS	
1 M7_PCIE_PHY_PUP_STATUS	
0 M7_MIPI_PHY1_PUP_STATUS	

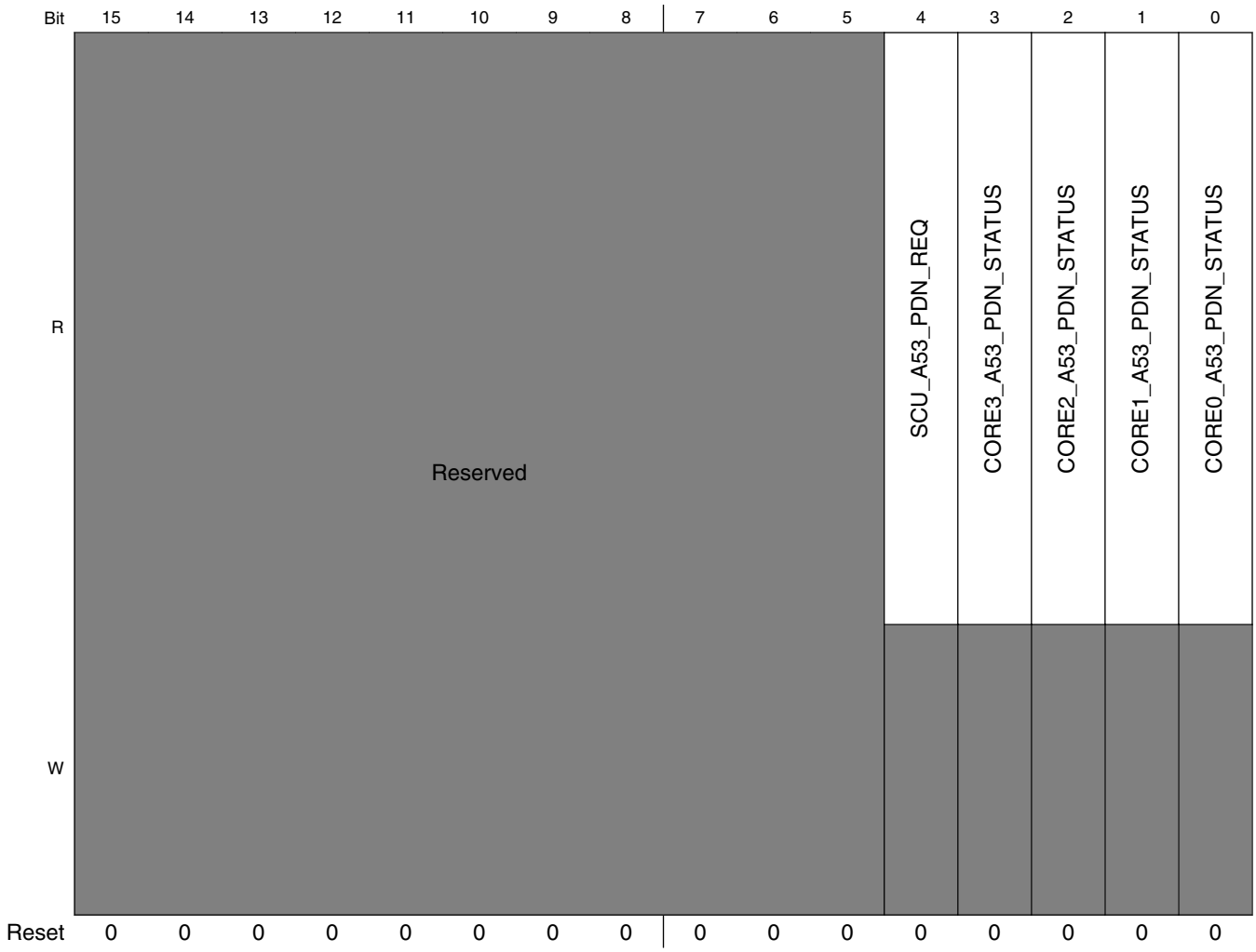
5.2.10.47 CPU PGC software dn trigger status1 (GPC_CPU_PGC_PDN_STATUS1)

CPU_PGC_PDN_STATUS1 is a read only register, represents the results for power DN software trigger for CPU type PGCs.

The field description is show in table below, the value of “1’b1” represent the software power DN trigger failed because the relevant PGC is in a power down process. The relevant bit will be cleared after a success operation of power DN software trigger for CPU type PGCs.

Address: 303A_0000h base + 13Ch offset = 303A_013Ch





GPC_CPU_PGC_PDN_STATUS1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SCU_A53_PDN_REQ	
3 CORE3_A53_PDN_STATUS	
2 CORE2_A53_PDN_STATUS	
1 CORE1_A53_PDN_STATUS	
0 CORE0_A53_PDN_STATUS	

5.2.10.48 A53 MIX software down trigger status register (GPC_A53_MIX_PGC_PDN_STATUS*n*)

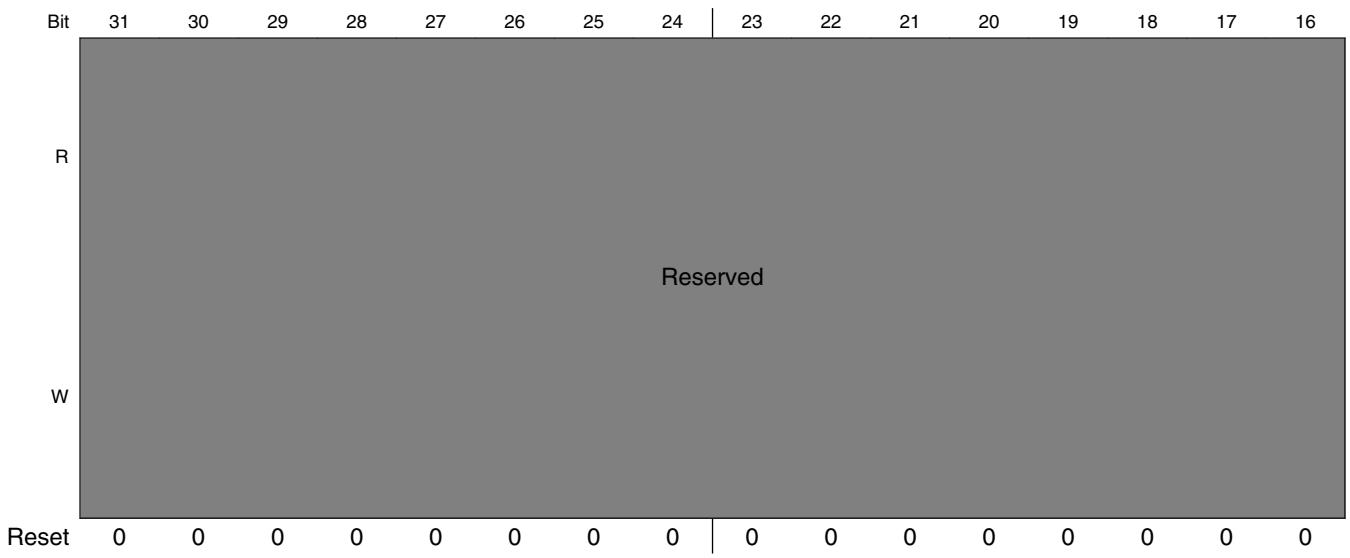
A53_MIX_PGC_PDN_STATUSn (n = 0,1,2) are a read only register, represents the results for power down software trigger from A53 platform to MIX type PGCs.

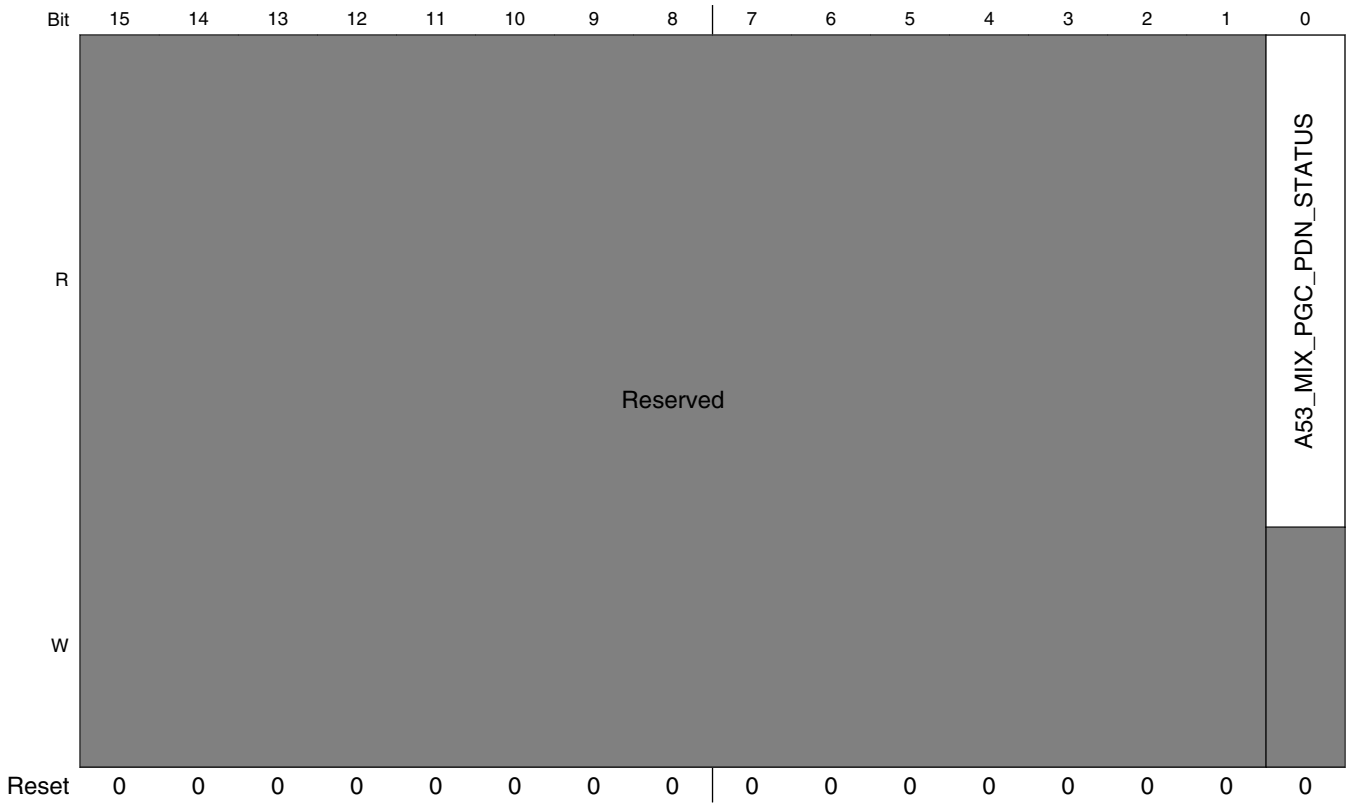
A53_MIX_PGC_PDN_STATUS0: value of “1’b1” represent the software power up trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power down software trigger for MIX type PGCs.

A53_MIX_PGC_PDN_STATUS1: value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power up process. The relevant bit will be cleared after a success operation of power down software trigger for MIX type PGCs.

A53_MIX_PGC_PDN_STATUS2: value of “1’b1” represent the software power up trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power down software trigger for MIX type PGCs.

Address: 303A_0000h base + 140h offset + (4d × i), where i=0d to 2d





GPC_A53_MIX_PGC_PDN_STATUS*n* field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 A53_MIX_PGC_ PDN_STATUS	

5.2.10.49 M7 MIX PGC software power down trigger status register (GPC_M7_MIX_PGC_PDN_STATUS*n*)

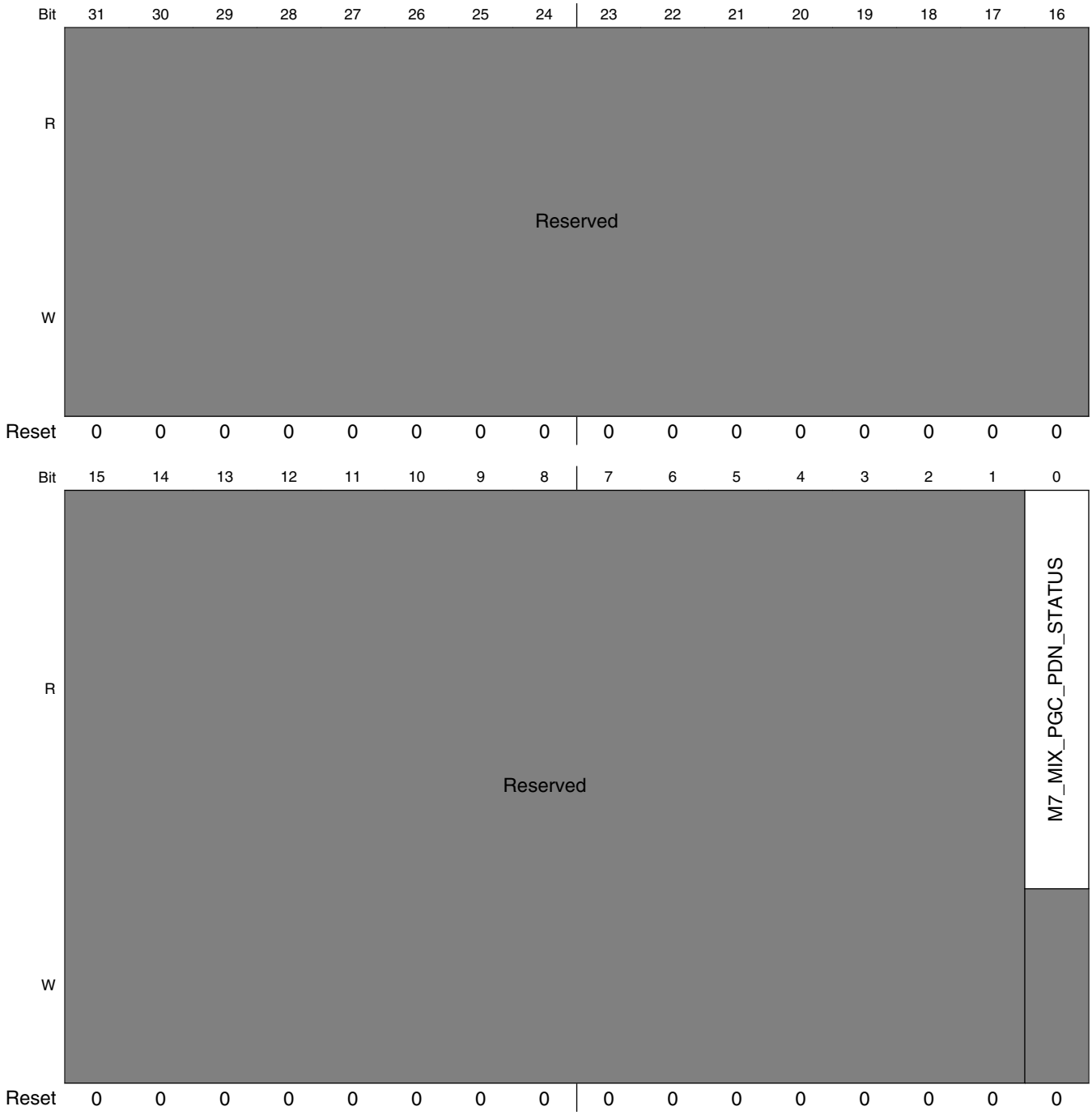
M7_MIX_PGC_PDN_STATUSn (n = 0,1,2) are a read only register, represents the results for power down software trigger from M7 platform to MIX type PGCs.

M7_MIX_PGC_PDN_STATUS0: value of “1’b1” represent the software power up trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power down software trigger for MIX type PGCs.

M7_MIX_PGC_PDN_STATUS1: value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power up process. The relevant bit will be cleared after a success operation of power down software trigger for MIX type PGCs.

M7_MIX_PGC_PDN_STATUS2: value of “1’b1” represent the software power up trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power down software trigger for MIX type PGCs.

Address: 303A_0000h base + 14Ch offset + (4d × i), where i=0d to 2d



GPC_M7_MIX_PGC_PDN_STATUS*n* field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 M7_MIX_PGC_ PDN_STATUS	

5.2.10.50 A53 PU PGC software down trigger status (GPC_A53_PU_PGC_PDN_STATUS*n*)

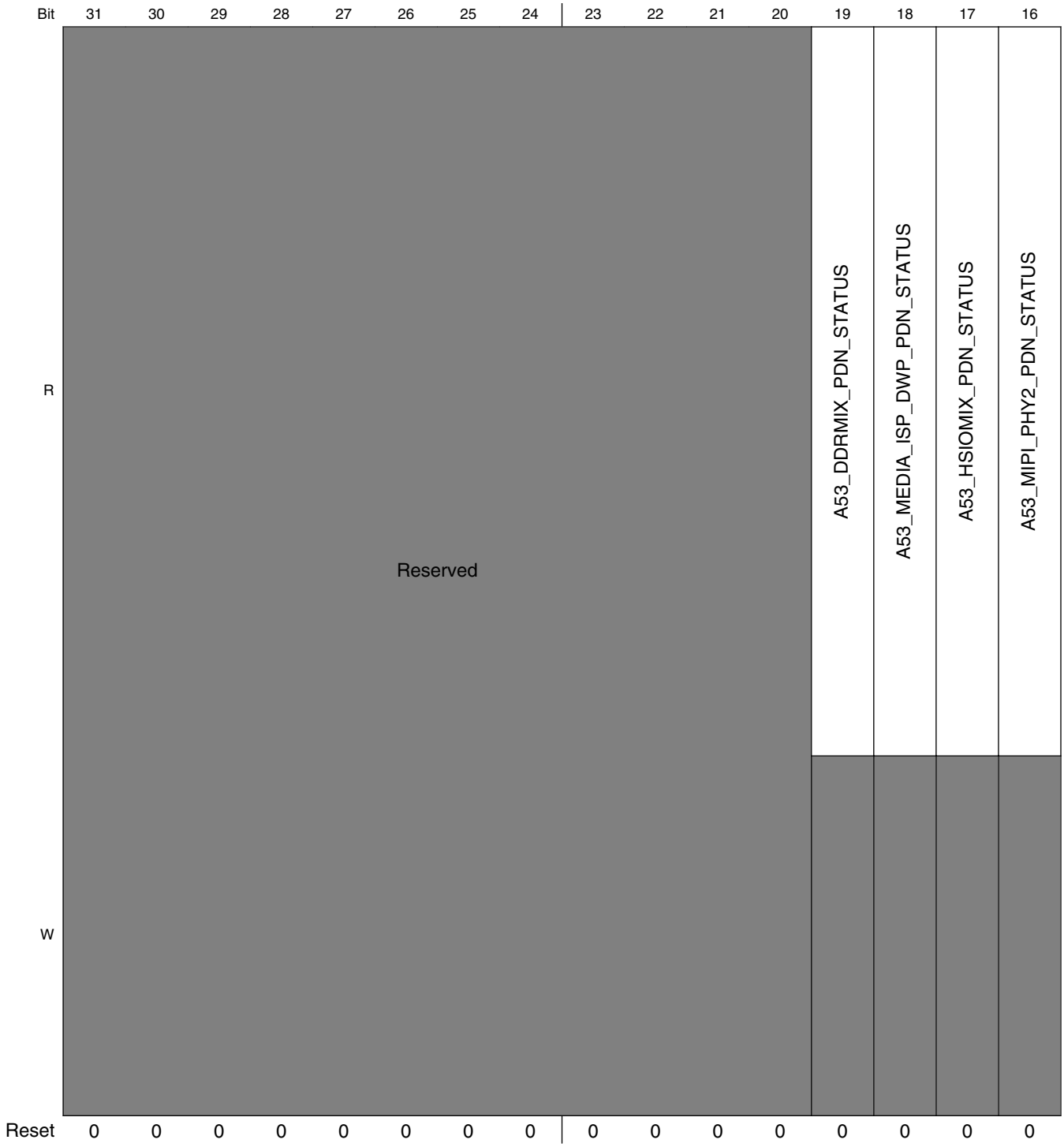
A53_PU_PGC_PDN_STATUSn (n = 0,1,2) are a read only register, represents the results for power DN software trigger from A53 platform to PU type PGCs.

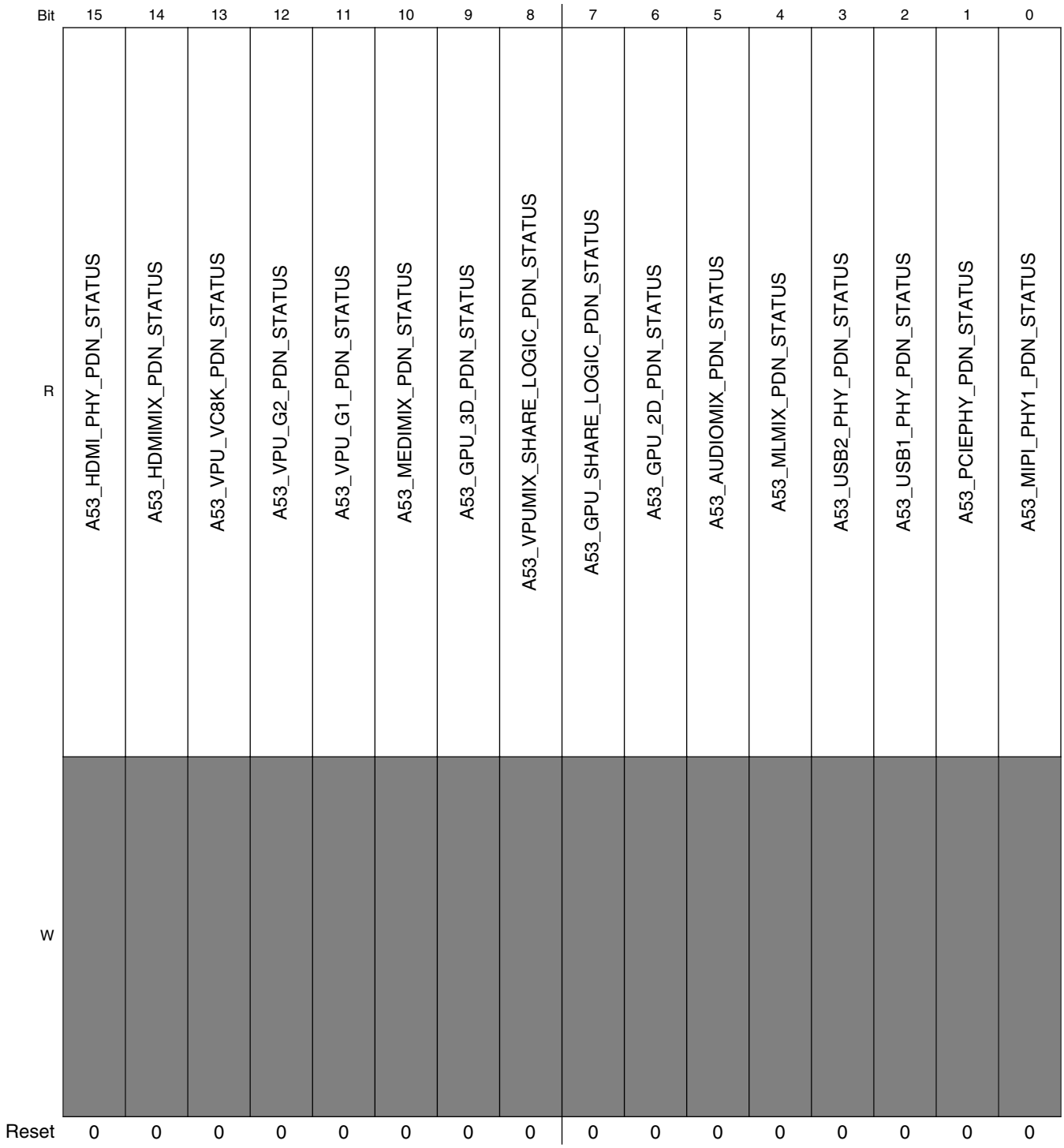
A53_PU_PGC_PDN_STATUS0: value of “1’b1” represent the software power DN trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power DN software trigger for PU type PGCs.

A53_PU_PGC_PDN_STATUS1: value of “1’b1” represent the software power DN trigger failed because the relevant PGC is in a power DN process. The relevant bit will be cleared after a success operation of power DN software trigger for PU type PGCs.

A53_PU_PGC_PDN_STATUS2: value of “1’b1” represent the software power DN trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power DN software trigger for PU type PGCs.

Address: 303A_0000h base + 158h offset + (4d × i), where i=0d to 2d





GPC_A53_PU_PGC_PDN_STATUS*n* field descriptions

Field	Description
31–20 -	This field is reserved. Reserved

Table continues on the next page...

GPC_A53_PU_PGC_PDN_STATUS*n* field descriptions (continued)

Field	Description
19 A53_DDRMIX_ PDN_STATUS	
18 A53_MEDIA_ ISP_DWP_PDN_ STATUS	
17 A53_HSIOMIX_ PDN_STATUS	
16 A53_MIPI_ PHY2_PDN_ STATUS	
15 A53_HDMI_ PHY_PDN_ STATUS	
14 A53_HDMIMIX_ PDN_STATUS	
13 A53_VPU_VC8K _PDN_STATUS	
12 A53_VPU_G2_P DN_STATUS	
11 A53_VPU_G1_P DN_STATUS	
10 A53_MEDIMIX_P DN_STATUS	
9 A53_GPU_3D_P DN_STATUS	
8 A53_VPUMIX_S HARE_LOGIC_P DN_STATUS	
7 A53_GPU_SHAR E_LOGIC_PDN_ STATUS	
6 A53_GPU_2D_ PDN_STATUS	

Table continues on the next page...

GPC_A53_PU_PGC_PDN_STATUS*n* field descriptions (continued)

Field	Description
5 A53_AUDIOMIX_ PDN_STATUS	
4 A53_MLMIX_ PDN_STATUS	
3 A53_USB2_PHY _PDN_STATUS	
2 A53_USB1_PHY _PDN_STATUS	
1 A53_PCIEPHY_ PDN_STATUS	
0 A53_MIPI_PHY1 _PDN_STATUS	

5.2.10.51 M7 PU PGC software down trigger status (GPC_M7_PU_PGC_PDN_STATUS*n*)

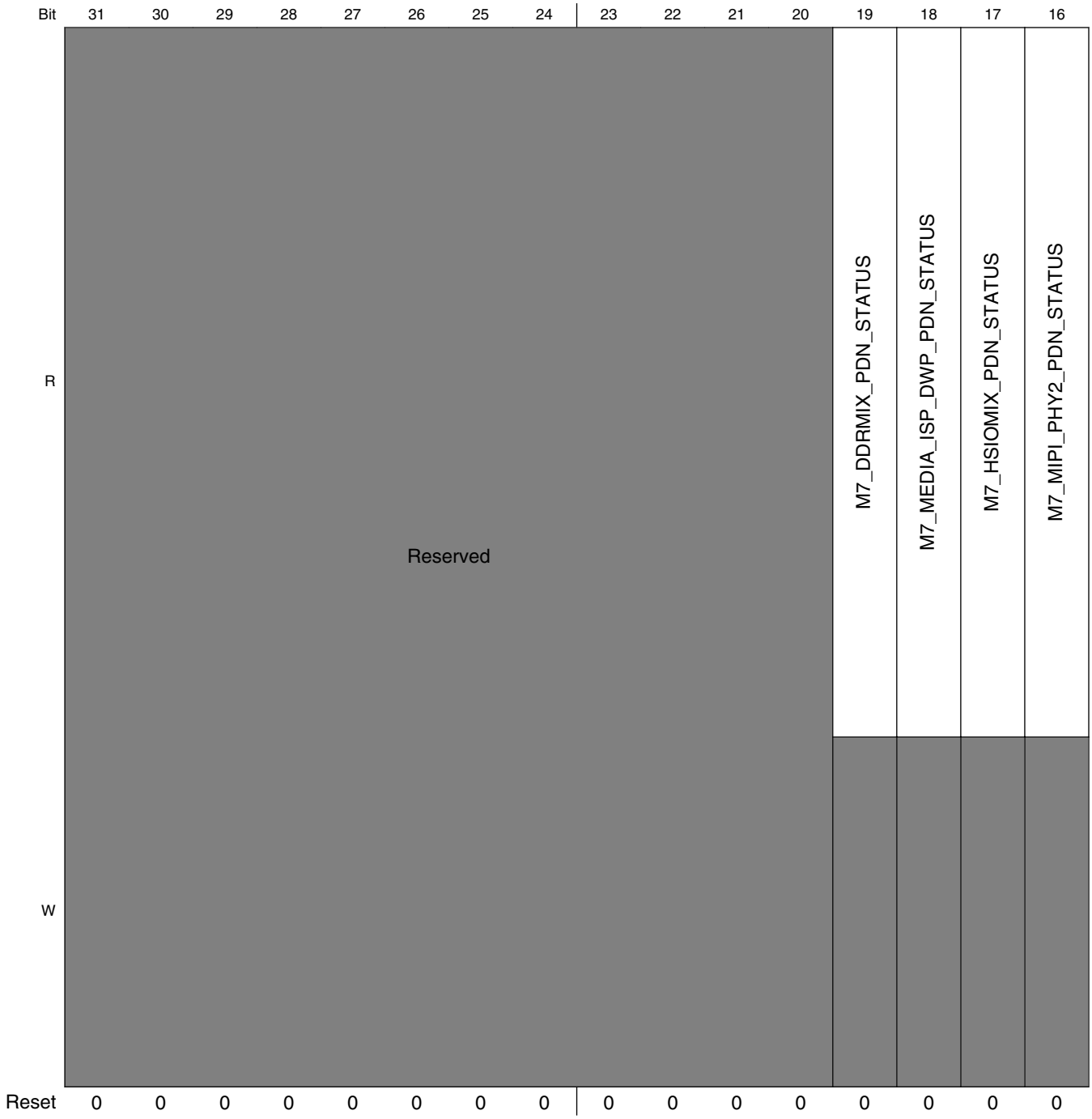
M7_PU_PGC_PDN_STATUSn (n = 0,1,2) are a read only register, represents the results for power DN software trigger from M7 platform to PU type PGCs.

M7_PU_PGC_PDN_STATUS0: value of “1’b1” represent the software power DN trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power DN software trigger for PU type PGCs.

M7_PU_PGC_PDN_STATUS1: value of “1’b1” represent the software power DN trigger failed because the relevant PGC is in a power DN process. The relevant bit will be cleared after a success operation of power DN software trigger for PU type PGCs.

M7_PU_PGC_PDN_STATUS2: value of “1’b1” represent the software power DN trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power DN software trigger for PU type PGCs.

Address: 303A_0000h base + 164h offset + (4d × i), where i=0d to 2d



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M7_HDMI_PHY_PDN_STATUS	M7_HDMIMIX_PDN_STATUS	M7_VPU_VC8K_PDN_STATUS	M7_VPU_G2_PDN_STATUS	M7_VPU_G1_PDN_STATUS	M7_MEDIMIX_PDN_STATUS	M7_GPU3D_PDN_STATUS	M7_VPUMIX_SHARE_LOGIC_PDN_STATUS	M7_GPU_SHARE_LOGIC_PDN_STATUS	M7_GPU_2D_PDN_STATUS	M7_AUDIOMIX_PDN_STATUS	M7_MLMIX_PDN_STATUS	M7_USB2_PHY_PDN_STATUS	M7_USB1_PHY_PDN_STATUS	M7_PCIE_PHY_PDN_STATUS	M7_MIPI_PHY1_PDN_STATUS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_M7_PU_PGC_PDN_STATUS*n* field descriptions

Field	Description
31–20 -	This field is reserved. Reserved
19 M7_DDRMIX_ PDN_STATUS	

Table continues on the next page...

GPC_M7_PU_PGC_PDN_STATUS*n* field descriptions (continued)

Field	Description
18 M7_MEDIA_ISP_DWP_PDN_STATUS	
17 M7_HSIOMIX_PDN_STATUS	
16 M7_MIPI_PHY2_PDN_STATUS	
15 M7_HDMI_PHY_PDN_STATUS	
14 M7_HDMIMIX_PDN_STATUS	
13 M7_VPU_VC8K_PDN_STATUS	
12 M7_VPU_G2_PDN_STATUS	
11 M7_VPU_G1_PDN_STATUS	
10 M7_MEDIMIX_PDN_STATUS	
9 M7_GPU3D_PDN_STATUS	
8 M7_VPUMIX_SHARE_LOGIC_PDN_STATUS	
7 M7_GPU_SHARE_LOGIC_PDN_STATUS	
6 M7_GPU_2D_PDN_STATUS	
5 M7_AUDIOMIX_PDN_STATUS	
4 M7_MLMIX_PDN_STATUS	

Table continues on the next page...

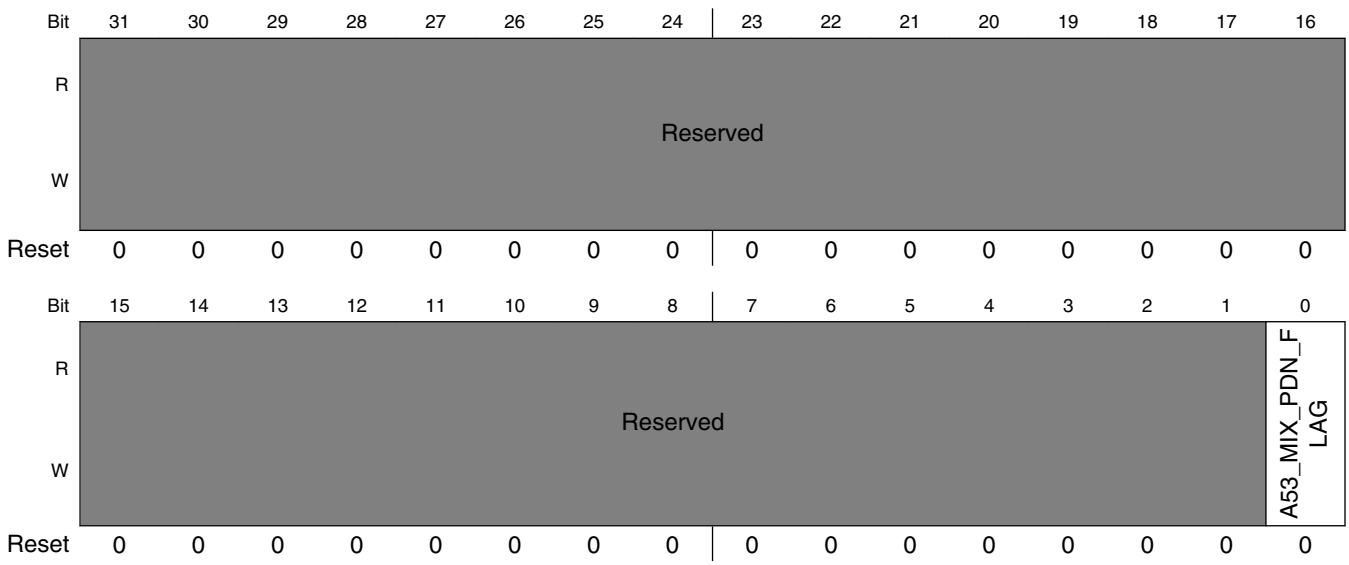
GPC_M7_PU_PGC_PDN_STATUS*n* field descriptions (continued)

Field	Description
3 M7_USB2_PHY_PDN_STATUS	
2 M7_USB1_PHY_PDN_STATUS	
1 M7_PCIE_PHY_PDN_STATUS	
0 M7_MIPI_PHY1_PDN_STATUS	

5.2.10.52 A53 MIX PDN FLG (GPC_A53_MIX_PDN_FLG)

This is flag bit relevant domain control, represents A53 CPU platform wants to power down MIX PGC. The register can only be accessed by A53 platform.

Address: 303A_0000h base + 170h offset = 303A_0170h



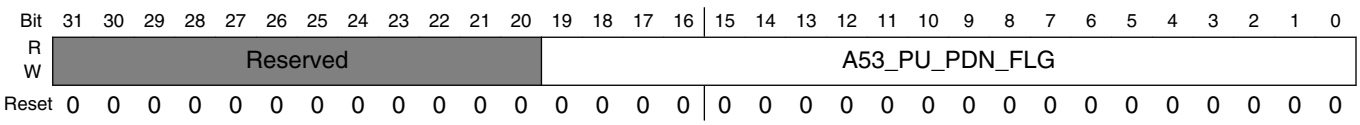
GPC_A53_MIX_PDN_FLG field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 A53_MIX_PDN_FLAG	A53 MIX power-down flag

5.2.10.53 A53 PU PDN FLG (GPC_A53_PU_PDN_FLG)

The register field is show in the table below. The 1’b1 represents A53 CPU platform wants to power down certain PU PGC. The register is a read only register. The register bits will be set when corresponding A53 software power down trigger happens and will be clear when corresponding A53 software power up trigger happens.

Address: 303A_0000h base + 174h offset = 303A_0174h



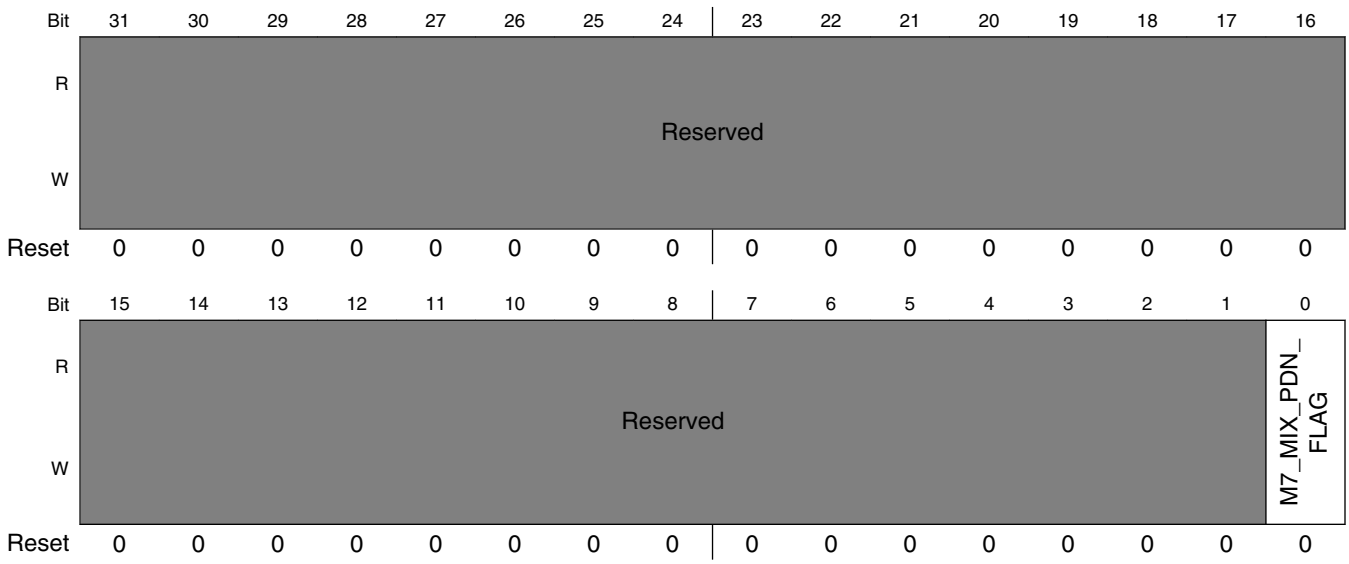
GPC_A53_PU_PDN_FLG field descriptions

Field	Description
31–20 -	This field is reserved. Reserved
A53_PU_PDN_FLG	A53 PGC power-down flag

5.2.10.54 M7 MIX PDN FLG (GPC_M7_MIX_PDN_FLG)

This is flag bit relevant domain control, represents M7 CPU platform wants to power down MIX PGC. The register can only be accessed by M7 platform.

Address: 303A_0000h base + 178h offset = 303A_0178h



GPC_M7_MIX_PDN_FLG field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 M7_MIX_PDN_FLAG	M7_MIX power-down flag

5.2.10.55 M7 PU PDN FLG (GPC_M7_PU_PDN_FLG)

The register field is show in the table below. The 1'b1 represents M7 CPU platform wants to power down certain PU PGC. The register is a read only register. The register bits will be set when corresponding M7 software power down trigger happens and will be clear when corresponding M7 software power up trigger happens.

Address: 303A_0000h base + 17Ch offset = 303A_017Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												M7_PU_PDN_FLG																				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_M7_PU_PDN_FLG field descriptions

Field	Description
31–20 -	This field is reserved. Reserved
M7_PU_PDN_ FLG	M7 power-down flag

5.2.10.56 Basic Low power control register of A53 platform (GPC_LPCR_A53_BSC2)

Address: 303A_0000h base + 180h offset = 303A_0180h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												LPM3		LPM2	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_LPCR_A53_BSC2 field descriptions

Field	Description
31–4 -	This field is reserved. Reserved
3–2 LPM3	CORE3 Setting the low power mode that system will enter on next assertion of dsm_request signal. 00 Remain in RUN mode 01 Transfer to WAIT mode

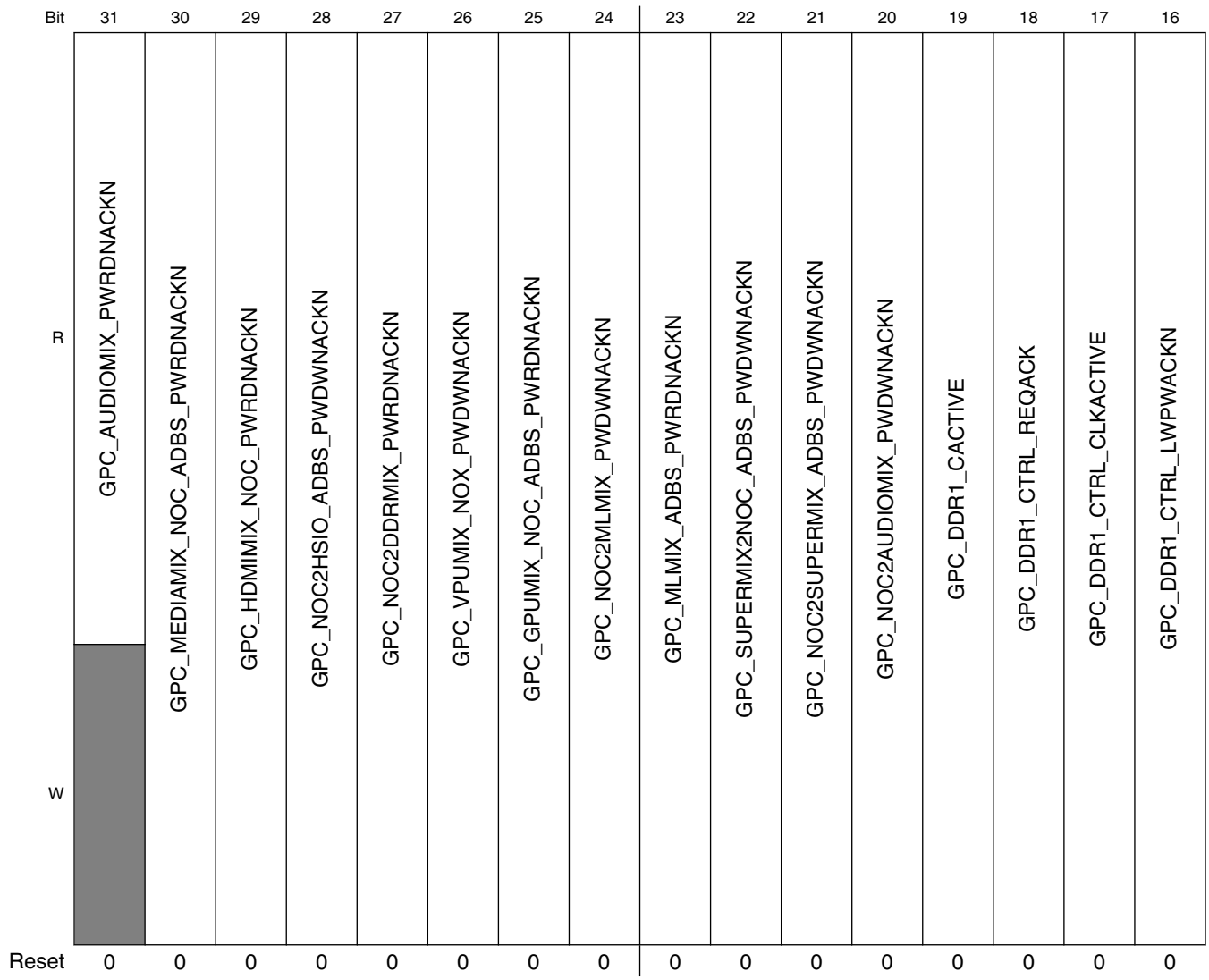
Table continues on the next page...

GPC_LPCR_A53_BSC2 field descriptions (continued)

Field	Description
	10 Transfer to STOP mode 11 Reserved
LPM2	CORE2 Setting the low power mode that system will enter on next assertion of dsm_request signal. 00 Remain in RUN mode 01 Transfer to WAIT mode 10 Transfer to STOP mode 11 Reserved

5.2.10.57 Power handshake register (GPC_PU_PWRHSK)

Address: 303A_0000h base + 190h offset = 303A_0190h



Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GPC_AUDIOMIX_PWRDNACKN	GPC_MEDIAMIX_NOC_ADBS_PWRDNACKN	GPC_HDMIMIX_NOC_PWRDNACKN	GPC_NOC2HSIO_ADBS_PWDWNACKN	GPC_NOC2DDRMIX_PWRDNACKN	GPC_VPUMIX_NOX_PWDWNACKN	GPC_GPUMIX_NOC_ADBS_PWRDNACKN	GPC_NOC2MLMIX_PWDWNACKN	GPC_MLMIX_ADBS_PWRDNACKN	GPC_SUPERMIX2NOC_ADBS_PWDWNACKN	GPC_NOC2SUPERMIX_ADBS_PWDWNACKN	GPC_NOC2AUDIOMIX_PWDWNACKN	GPC_DDR1_CACTIVE	GPC_DDR1_CTRL_REQACK	GPC_DDR1_CTRL_CLKACTIVE	GPC_DDR1_CTRL_LWPWACKN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPC_AUDIOMIX_NOC_PWRDNREQN	GPC_MEDIAMIX_NOC_ADBS_PWRDNREQN	GPC_HDMIMIX_NOC_PWRDNREQN	GPC_NOC2HSIO_ADBS_PWRDNREQN	GPC_NOC2DDRMIX_PWRDNREQN	GPC_VPUMIX_NOC_PWRDNREQN	GPC_GPUMIX_NOC_ADBS_PWRDNREQN	GPC_NOC2MLMIX_PWRDNREQN	GPC_MLMIX_ADBS_PWRDNREQN	GPC_SUPERMIX2NOC_PWRDNREQN	GPC_NOC2SUPERMIX_PWRDNREQN	GPC_NOC2AUDIOMIX_PWRDNREQN	Reserved		GPC_DDR1_AXI_CSYSREQ	GPC_DDR1_CORE_CSYSREQ
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

GPC_PU_PWRHSK field descriptions

Field	Description
31 GPC_AUDIOMIX_ PWRDNACKN	Audiomix noc power down ackn. Read 0 before power down audiomix switchable domain or main noc domain
30 GPC_MEDIAMIX_ NOC_ADBS_ PWRDNACKN	Mediamix noc and adbs power down ackn. Read 0 before power down mediamix default domain or main noc domain
29 GPC_HDMIMIX_ NOC_PWRDNACKN	Hdmimix noc power down ackn. Read 0 before power down hdmimix switchable domain or main noc domain
28 GPC_NOC2HSIO_ ADBS_ PWDWNACKN	Main noc 2 hsio and adbs power down ackn. Read 0 before power down main noc or hsio switchable domain
27 GPC_ NOC2DDRMIX_ PWRDNACKN	Main noc 2 ddrmix power down ackn. Read 0 before power down main noc or ddrmix switchable domain
26 GPC_VPUMIX_ NOX_PWDWNACKN	Vpumix noc power down ackn. Read 0 before power down vpumix default share logic domain or main noc domain
25 GPC_GPUMIX_ NOC_ADBS_ PWRDNACKN	gpumix noc and adbs power down ackn. Read 0 before power down gpumix default share logic domain or main noc domain

Table continues on the next page...

GPC_PU_PWRHSK field descriptions (continued)

Field	Description
24 GPC_NOC2MLMIX_ PWDWNACKN	Main noc 2 mlmix power down ackn. Read 0 before power down mlmix switchable domain or main noc domain
23 GPC_MLMIX_ ADBS_ PWRDNACKN	Mlmix adbs power down ackn. Read 0 before power down mlmix switchable domain or main noc domain
22 GPC_ SUPERMIX2NOC_ ADBS_ PWDWNACKN	Supermix 2 noc adbs power down ackn. Read 0 before power down main noc domain. It can be covered by hardware handshake follow.
21 GPC_ NOC2SUPERMIX_ ADBS_ PWDWNACKN	Main noc 2 Supermix adbs power down ackn. Read 0 before power down main noc domain. It can be covered by hardware handshake follow.
20 GPC_ NOC2AUDIOMIX_ PWDWNACKN	Main noc 2 audiomix power down ackn. Read 0 before power down main noc or audiomix switchable domain
19 GPC_DDR1_ CACTIVE	DDR1 AXI Clock Active
18 GPC_DDR1_CTRL_ REQACK	DDR1 AXI Low-Power Request ack
17 GPC_DDR1_CTRL_ CLKACTIVE	DDR1 controller Hardware Low-Power Clock active
16 GPC_DDR1_CTRL_ LWPWACKN	DDR1 controller Hardware Low_Power ack
15 GPC_AUDIOMIX_ NOC_PWRDNREQN	Audiomix noc power down request. Write 0 and wait according ackn before power down audiomix switchable domain or main noc domain
14 GPC_MEDIAMIX_ NOC_ADBS_ PWRDNREQN	Mediamix noc and adbs power down request. Write 0 and wait according ackn before power down mediamix default domain or main noc domain
13 GPC_HDMIMIX_ NOC_PWRDNREQN	Hdmimix noc power down request. Write 0 and wait according ackn before power down hdmimix switchable domain or main noc domain
12 GPC_NOC2HSIO_ ADBS_ PWRDNREQN	Main noc 2 hsio and adbs power down request. Write 0 and wait according ackn before power down main noc or hsio switchable domain

Table continues on the next page...

GPC_PU_PWRHSK field descriptions (continued)

Field	Description
11 GPC_ NOC2DDRMIX_ PWRDNREQN	Main noc 2 ddrmix power down request. Write 0 and wait according ackn before power down main noc or ddrmix switchable domain
10 GPC_VPUMIX_ NOC_PWRDNREQN	Vpumix noc power down request. Write 0 and wait according ackn before power down vpumix default share logic domain or main noc domain
9 GPC_GPUMIX_ NOC_ADBS_ PWRDNREQN	gpumix noc and adbs power down request. Write 0 and wait according ackn before power down gpumix default share logic domain or main noc domain
8 GPC_NOC2MLMIX_ PWRDNREQN	Main noc 2 mlmix power down request. Write 0 and wait according ackn before power down mlmix switchable domain or main noc domain
7 GPC_MLMIX_ ADBS_ PWRDNREQN	Mlmix adbs power down request. Write 0 and wait according ackn before power down mlmix switchable domain or main noc domain
6 GPC_ SUPERMIX2NOC_ PWRDNREQN	Supermix 2 noc adbs power down request. Write 0 and wait according ackn before power down main noc domain. It can be covered by hardware handshake follow.
5 GPC_ NOC2SUPERMIX_ PWRDNREQN	DISPMIX ADB400 power down request. Active 0
4 GPC_ NOC2AUDIOMIX_ PWRDNREQN	Main noc 2 audiomix power down request. Write 0 and wait according ackn before power down main noc or audiomix switchable domain
3–2 -	This field is reserved. Reserved
1 GPC_DDR1_AXI_ CSYSREQ	DDR1 AXI Low-Power Request
0 GPC_DDR1_CORE_ CSYSREQ	DDR1 controller Hardware Low-Power Request

5.2.10.58 IRQ masking register 1 of A53 core2 (GPC_IMR1_CORE2_A53)

The five IMRn_CORE2_A53 (n = 1,2,3,4,5) registers are used as interrupt mask for A53 core2.

Address: 303A_0000h base + 194h offset = 303A_0194h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR1_CORE2_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR1_CORE2_A53 field descriptions

Field	Description
IMR1_CORE2_A53	A53 core2 IRQ[31:0] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.59 IRQ masking register 2 of A53 core2 (GPC_IMR2_CORE2_A53)

Address: 303A_0000h base + 198h offset = 303A_0198h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR2_CORE2_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR2_CORE2_A53 field descriptions

Field	Description
IMR2_CORE2_A53	A53 core2 IRQ[63:32] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.60 IRQ masking register 3 of A53 core2 (GPC_IMR3_CORE2_A53)

Address: 303A_0000h base + 19Ch offset = 303A_019Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR3_CORE2_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR3_CORE2_A53 field descriptions

Field	Description
IMR3_CORE2_A53	A53 core2 IRQ[95:64] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.61 IRQ masking register 4 of A53 core2 (GPC_IMR4_CORE2_A53)

Address: 303A_0000h base + 1A0h offset = 303A_01A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR4_CORE2_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR4_CORE2_A53 field descriptions

Field	Description
IMR4_CORE2_A53	A53 core2 IRQ[127:96] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.62 IRQ masking register 5 of A53 core2 (GPC_IMR5_CORE2_A53)

Address: 303A_0000h base + 1A4h offset = 303A_01A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR5_CORE2_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR5_CORE2_A53 field descriptions

Field	Description
IMR5_CORE2_A53	A53 core2 IRQ[159:128] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.63 IRQ masking register 1 of A53 core3 (GPC_IMR1_CORE3_A53)

The five IMRn_CORE2_A53 (n = 1,2,3,4,5) registers are used as interrupt mask for A53 core3.

Address: 303A_0000h base + 1A8h offset = 303A_01A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR1_CORE3_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR1_CORE3_A53 field descriptions

Field	Description
IMR1_CORE3_A53	A53 core3 IRQ[31:0] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.64 IRQ masking register 2 of A53 core3 (GPC_IMR2_CORE3_A53)

Address: 303A_0000h base + 1ACh offset = 303A_01ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR2_CORE3_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR2_CORE3_A53 field descriptions

Field	Description
IMR2_CORE3_A53	A53 core3 IRQ[63:32] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.65 IRQ masking register 3 of A53 core3 (GPC_IMR3_CORE3_A53)

Address: 303A_0000h base + 1B0h offset = 303A_01B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR3_CORE3_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR3_CORE3_A53 field descriptions

Field	Description
IMR3_CORE3_A53	A53 core3 IRQ[95:64] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.66 IRQ masking register 4 of A53 core3 (GPC_IMR4_CORE3_A53)

Address: 303A_0000h base + 1B4h offset = 303A_01B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR4_CORE3_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR4_CORE3_A53 field descriptions

Field	Description
IMR4_CORE3_A53	A53 core3 IRQ[127:96] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.67 IRQ masking register 5 of A53 core3 (GPC_IMR5_CORE3_A53)

Address: 303A_0000h base + 1B8h offset = 303A_01B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IM5_CORE3_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR5_CORE3_A53 field descriptions

Field	Description
IM5_CORE3_A53	A53 core3 IRQ[159:128] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.68 PGC acknowledge signal selection of A53 platform for PUs (GPC_ACK_SEL_A53_PU)

Address: 303A_0000h base + 1BCh offset = 303A_01BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HDMI_PHY_PGC_PUP_ACK	HDMI_PHY_PGC_PDN_ACK	HDMIMIX_PGC_PUP_ACK	HDMIMIX_PGC_PDN_ACK	VPU_VC8K_PGC_PUP_ACK	VPU_VC8K_PGC_PDN_ACK	VPU_G2_PGC_PUP_ACK	VPU_G2_PGC_PDN_ACK	VPU_G1_PGC_PUP_ACK	VPU_G1_PGC_PDN_ACK	MEDIMIX_PGC_PUP_ACK	MEDIMIX_PGC_PDN_ACK	GPU3D_PGC_PUP_ACK	GPU3D_PGC_PDN_ACK	VPUMIX_SHARE_LOGIC_PGC_PUP_ACK	VPUMIX_SHARE_LOGIC_PGC_PDN_ACK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPU_SHARE_LOGIC_PGC_PUP_ACK	GPU_SHARE_LOGIC_PGC_PDN_ACK	GPU_2D_PGC_PUP_ACK	GPU_2D_PGC_PDN_ACK	AUDIOMIX_PGC_PUP_ACK	AUDIOMIX_PGC_PDN_ACK	MLMIX_PGC_PUP_ACK	MLMIX_PGC_PDN_ACK	USB2_PHY_PGC_PUP_ACK	USB2_PHY_PGC_PDN_ACK	USB1_PHY_PGC_PUP_ACK	USB1_PHY_PGC_PDN_ACK	PCIE_PHY_PGC_PUP_ACK	PCIE_PHY_PGC_PDN_ACK	MIPI_PHY1_PGC_PUP_ACK	MIPI_PHY1_PGC_PDN_ACK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ACK_SEL_A53_PU field descriptions

Field	Description
31 HDMI_PHY_PGC_PUP_ACK	Select power down acknowledge signal of HDMI_PHY PGC as the power up acknowledge for A53 LPM
30 HDMI_PHY_PGC_PDN_ACK	Select power down acknowledge signal of HDMI_PHY PGC as the power down acknowledge for A53 LPM
29 HDMIMIX_PGC_PUP_ACK	Select power down acknowledge signal of HDMIMIX PGC as the power up acknowledge for A53 LPM
28 HDMIMIX_PGC_PDN_ACK	Select power down acknowledge signal of HDMIMIX PGC as the power down acknowledge for A53 LPM
27 VPU_VC8K_PGC_PUP_ACK	Select power down acknowledge signal of VPU_VC8K PGC as the power up acknowledge for A53 LPM
26 VPU_VC8K_PGC_PDN_ACK	Select power down acknowledge signal of VPU_VC8K PGC as the power down acknowledge for A53 LPM

Table continues on the next page...

GPC_ACK_SEL_A53_PU field descriptions (continued)

Field	Description
25 VPU_G2_PGC_PUP_ACK	Select power down acknowledge signal of VPU_G2 PGC as the power up acknowledge for A53 LPM
24 VPU_G2_PGC_PDN_ACK	Select power down acknowledge signal of VPU_G2 PGC as the power down acknowledge for A53 LPM
23 VPU_G1_PGC_PUP_ACK	Select power down acknowledge signal of VPU_G1 PGC as the power up acknowledge for A53 LPM
22 VPU_G1_PGC_PDN_ACK	Select power down acknowledge signal of VPU_G1 PGC as the power down acknowledge for A53 LPM
21 MEDIMIX_PGC_PUP_ACK	Select power down acknowledge signal of MEDIMIX PGC as the power up acknowledge for A53 LPM
20 MEDIMIX_PGC_PDN_ACK	Select power down acknowledge signal of MEDIMIX PGC as the power down acknowledge for A53 LPM
19 GPU3D_PGC_PUP_ACK	Select power down acknowledge signal of GPU3D PGC as the power up acknowledge for A53 LPM
18 GPU3D_PGC_PDN_ACK	Select power down acknowledge signal of GPU3D PGC as the power down acknowledge for A53 LPM
17 VPUMIX_SHARE_LOGIC_PGC_PUP_ACK	Select power down acknowledge signal of VPUMIX_SHARE_LOGIC PGC as the power up acknowledge for A53 LPM
16 VPUMIX_SHARE_LOGIC_PGC_PDN_ACK	Select power down acknowledge signal of VPUMIX_SHARE_LOGIC PGC as the power down acknowledge for A53 LPM
15 GPU_SHARE_LOGIC_PGC_PUP_ACK	Select power down acknowledge signal of GPU_SHARE_LOGIC PGC as the power up acknowledge for A53 LPM
14 GPU_SHARE_LOGIC_PGC_PDN_ACK	Select power down acknowledge signal of GPU_SHARE_LOGIC PGC as the power down acknowledge for A53 LPM
13 GPU_2D_PGC_PUP_ACK	Select power down acknowledge signal of GPU_2D PGC as the power up acknowledge for A53 LPM
12 GPU_2D_PGC_PDN_ACK	Select power down acknowledge signal of GPU_2D PGC as the power down acknowledge for A53 LPM
11 AUDIOMIX_PGC_PUP_ACK	Select power down acknowledge signal of AUDIOMIX PGC as the power up acknowledge for A53 LPM

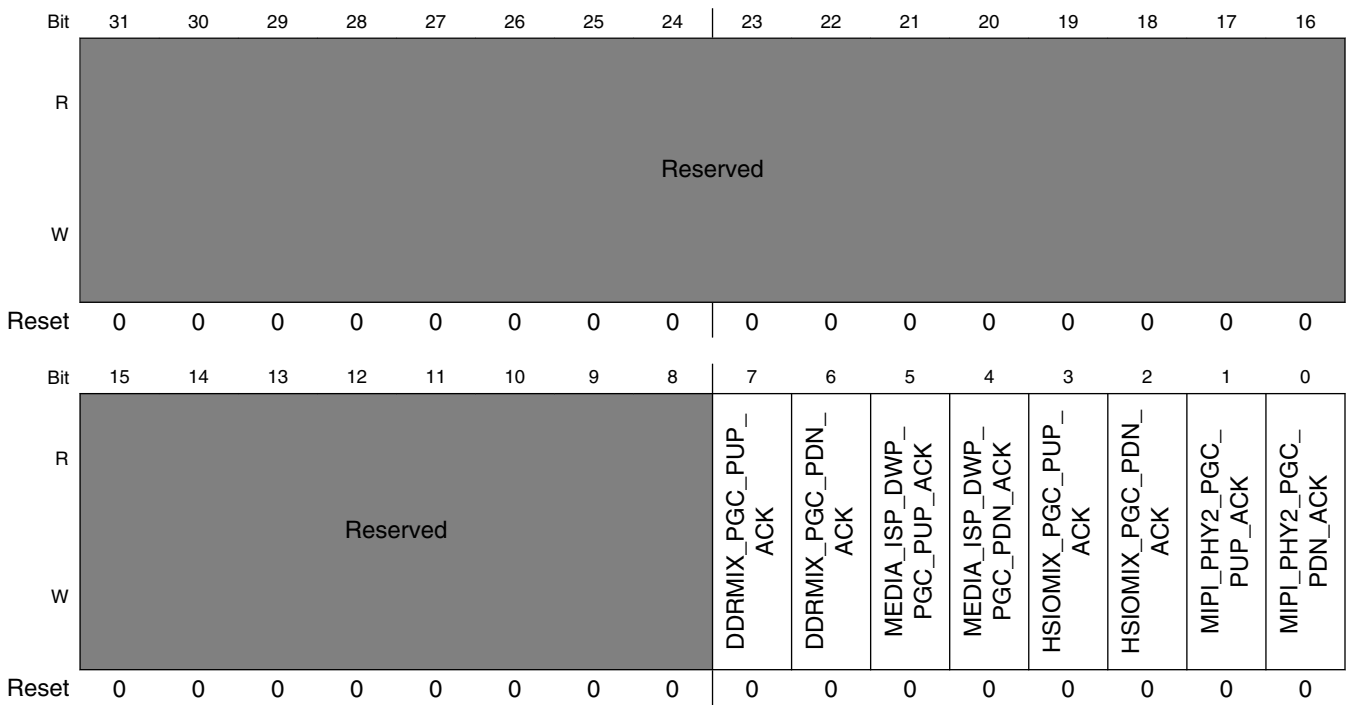
Table continues on the next page...

GPC_ACK_SEL_A53_PU field descriptions (continued)

Field	Description
10 AUDIOMIX_ PGC_PDN_ACK	Select power down acknowledge signal of AUDIOMIX PGC as the power down acknowledge for A53 LPM
9 MLMIX_PGC_ PUP_ACK	Select power down acknowledge signal of MLMIX PGC as the power up acknowledge for A53 LPM
8 MLMIX_PGC_ PDN_ACK	Select power down acknowledge signal of MLMIX PGC as the power down acknowledge for A53 LPM
7 USB2_PHY_ PGC_PUP_ACK	Select power down acknowledge signal of USB2_PHY PGC as the power up acknowledge for A53 LPM
6 USB2_PHY_ PGC_PDN_ACK	Select power down acknowledge signal of USB2_PHY PGC as the power down acknowledge for A53 LPM
5 USB1_PHY_ PGC_PUP_ACK	Select power down acknowledge signal of USB1_PHY PGC as the power up acknowledge for A53 LPM
4 USB1_PHY_ PGC_PDN_ACK	Select power down acknowledge signal of USB1_PHY PGC as the power down acknowledge for A53 LPM
3 PCIE_PHY_ PGC_PUP_ACK	Select power down acknowledge signal of PCIE_PHY PGC as the power up acknowledge for A53 LPM
2 PCIE_PHY_ PGC_PDN_ACK	Select power down acknowledge signal of PCIE_PHY PGC as the power down acknowledge for A53 LPM
1 MIPI_PHY1_ PGC_PUP_ACK	Select power down acknowledge signal of MIPI_PHY1 PGC as the power up acknowledge for A53 LPM
0 MIPI_PHY1_ PGC_PDN_ACK	Select power down acknowledge signal of MIPI_PHY1 PGC as the power down acknowledge for A53 LPM

5.2.10.69 PGC acknowledge signal selection of A53 platform for PUs (GPC_ACK_SEL_A53_PU1)

Address: 303A_0000h base + 1C0h offset = 303A_01C0h



GPC_ACK_SEL_A53_PU1 field descriptions

Field	Description
31–8 -	This field is reserved. Reserved
7 DDRMIX_PGC_PUP_ACK	Select power down acknowledge signal of DDRMIX PGC as the power up acknowledge for A53 LPM
6 DDRMIX_PGC_PDN_ACK	Select power down acknowledge signal of DDRMIX PGC as the power down acknowledge for A53 LPM
5 MEDIA_ISP_DWP_PGC_PUP_ACK	Select power down acknowledge signal of MEDIA_ISP_DWP PGC as the power up acknowledge for A53 LPM
4 MEDIA_ISP_DWP_PGC_PDN_ACK	Select power down acknowledge signal of MEDIA_ISP_DWP PGC as the power down acknowledge for A53 LPM

Table continues on the next page...

GPC_ACK_SEL_A53_PU1 field descriptions (continued)

Field	Description
3 HSIOMIX_PGC_PUP_ACK	Select power down acknowledge signal of HSIOMIX PGC as the power up acknowledge for A53 LPM
2 HSIOMIX_PGC_PDN_ACK	Select power down acknowledge signal of HSIOMIX PGC as the power down acknowledge for A53 LPM
1 MIPI_PHY2_PGC_PUP_ACK	Select power down acknowledge signal of MIPI_PHY2 PGC as the power up acknowledge for A53 LPM
0 MIPI_PHY2_PGC_PDN_ACK	Select power down acknowledge signal of MIPI_PHY2 PGC as the power down acknowledge for A53 LPM

5.2.10.70 PGC acknowledge signal selection of M7 platform for PUs (GPC_ACK_SEL_M7_PU)

Address: 303A_0000h base + 1C4h offset = 303A_01C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HDMI_PHY_PGC_PUP_ACK	HDMI_PHY_PGC_PDN_ACK	HDMIMIX_PGC_PUP_ACK	HDMIMIX_PGC_PDN_ACK	VPU_VC8K_PGC_PUP_ACK	VPU_VC8K_PGC_PDN_ACK	VPU_G2_PGC_PUP_ACK	VPU_G2_PGC_PDN_ACK	VPU_G1_PGC_PUP_ACK	VPU_G1_PGC_PDN_ACK	MEDIMIX_PGC_PUP_ACK	MEDIMIX_PGC_PDN_ACK	GPU3D_PGC_PUP_ACK	GPU3D_PGC_PDN_ACK	VPUMIX_SHARE_LOGIC_PGC_PUP_ACK	VPUMIX_SHARE_LOGIC_PGC_PDN_ACK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPU_SHARE_LOGIC_PGC_PUP_ACK	GPU_SHARE_LOGIC_PGC_PDN_ACK	GPU_2D_PGC_PUP_ACK	GPU_2D_PGC_PDN_ACK	AUDIOMIX_PGC_PUP_ACK	AUDIOMIX_PGC_PDN_ACK	MLMIX_PGC_PUP_ACK	MLMIX_PGC_PDN_ACK	USB2_PHY_PGC_PUP_ACK	USB2_PHY_PGC_PDN_ACK	USB1_PHY_PGC_PUP_ACK	USB1_PHY_PGC_PDN_ACK	PCIE_PHY_PGC_PUP_ACK	PCIE_PHY_PGC_PDN_ACK	MIPI_PHY1_PGC_PUP_ACK	MIPI_PHY1_PGC_PDN_ACK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ACK_SEL_M7_PU field descriptions

Field	Description
31 HDMI_PHY_ PGC_PUP_ACK	Select power down acknowledge signal of HDMI_PHY PGC as the power up acknowledge for M7 LPM
30 HDMI_PHY_ PGC_PDN_ACK	Select power down acknowledge signal of HDMI_PHY PGC as the power down acknowledge for M7 LPM
29 HDMIMIX_PGC_ PUP_ACK	Select power down acknowledge signal of HDMIMIX PGC as the power up acknowledge for M7 LPM
28 HDMIMIX_PGC_ PDN_ACK	Select power down acknowledge signal of HDMIMIX PGC as the power down acknowledge for M7 LPM
27 VPU_VC8K_ PGC_PUP_ACK	Select power down acknowledge signal of VPU_VC8K PGC as the power up acknowledge for M7 LPM
26 VPU_VC8K_ PGC_PDN_ACK	Select power down acknowledge signal of VPU_VC8K PGC as the power down acknowledge for M7 LPM
25 VPU_G2_PGC_ PUP_ACK	Select power down acknowledge signal of VPU_G2 PGC as the power up acknowledge for M7 LPM
24 VPU_G2_PGC_ PDN_ACK	Select power down acknowledge signal of VPU_G2 PGC as the power down acknowledge for M7 LPM
23 VPU_G1_PGC_ PUP_ACK	Select power down acknowledge signal of VPU_G1 PGC as the power up acknowledge for M7 LPM
22 VPU_G1_PGC_ PDN_ACK	Select power down acknowledge signal of VPU_G1 PGC as the power down acknowledge for M7 LPM
21 MEDIMIX_PGC_ PUP_ACK	Select power down acknowledge signal of MEDIMIX PGC as the power up acknowledge for M7 LPM
20 MEDIMIX_PGC_ PDN_ACK	Select power down acknowledge signal of MEDIMIX PGC as the power down acknowledge for M7 LPM
19 GPU3D_PGC_ PUP_ACK	Select power down acknowledge signal of GPU3D PGC as the power up acknowledge for M7 LPM
18 GPU3D_PGC_ PDN_ACK	Select power down acknowledge signal of GPU3D PGC as the power down acknowledge for M7 LPM
17 VPUMIX_ SHARE_LOGIC_ PGC_PUP_ACK	Select power down acknowledge signal of VPUMIX_SHARE_LOGIC PGC as the power up acknowledge for M7 LPM

Table continues on the next page...

GPC_ACK_SEL_M7_PU field descriptions (continued)

Field	Description
16 VPUMIX_ SHARE_LOGIC_ PGC_PDN_ACK	Select power down acknowledge signal of VPUMIX_SHARE_LOGIC PGC as the power down acknowledge for M7 LPM
15 GPU_SHARE_ LOGIC_PGC_ PUP_ACK	Select power down acknowledge signal of GPU_SHARE_LOGIC PGC as the power up acknowledge for M7 LPM
14 GPU_SHARE_ LOGIC_PGC_ PDN_ACK	Select power down acknowledge signal of GPU_SHARE_LOGIC PGC as the power down acknowledge for M7 LPM
13 GPU_2D_PGC_ PUP_ACK	Select power down acknowledge signal of GPU_2D PGC as the power up acknowledge for M7 LPM
12 GPU_2D_PGC_ PDN_ACK	Select power down acknowledge signal of GPU_2D PGC as the power down acknowledge for M7 LPM
11 AUDIOMIX_ PGC_PUP_ACK	Select power down acknowledge signal of AUDIOMIX PGC as the power up acknowledge for M7 LPM
10 AUDIOMIX_ PGC_PDN_ACK	Select power down acknowledge signal of AUDIOMIX PGC as the power down acknowledge for M7 LPM
9 MLMIX_PGC_ PUP_ACK	Select power down acknowledge signal of MLMIX PGC as the power up acknowledge for M7 LPM
8 MLMIX_PGC_ PDN_ACK	Select power down acknowledge signal of MLMIX PGC as the power down acknowledge for M7 LPM
7 USB2_PHY_ PGC_PUP_ACK	Select power down acknowledge signal of USB2_PHY PGC as the power up acknowledge for A53 LPM
6 USB2_PHY_ PGC_PDN_ACK	Select power down acknowledge signal of USB2_PHY PGC as the power down acknowledge for M7 LPM
5 USB1_PHY_ PGC_PUP_ACK	Select power down acknowledge signal of USB1_PHY PGC as the power up acknowledge for M7 LPM
4 USB1_PHY_ PGC_PDN_ACK	Select power down acknowledge signal of USB1_PHY PGC as the power down acknowledge for M7 LPM
3 PCIE_PHY_ PGC_PUP_ACK	Select power down acknowledge signal of PCIE_PHY PGC as the power up acknowledge for M7 LPM
2 PCIE_PHY_ PGC_PDN_ACK	Select power down acknowledge signal of PCIE_PHY PGC as the power down acknowledge for M7 LPM

Table continues on the next page...

GPC_ACK_SEL_M7_PU field descriptions (continued)

Field	Description
1 MIPI_PHY1_ PGC_PUP_ACK	Select power down acknowledge signal of MIPI_PHY1 PGC as the power up acknowledge for M7 LPM
0 MIPI_PHY1_ PGC_PDN_ACK	Select power down acknowledge signal of MIPI_PHY1 PGC as the power down acknowledge for M7 LPM

5.2.10.71 PGC acknowledge signal selection of M7 platform for PUs (GPC_ACK_SEL_M7_PU1)

Address: 303A_0000h base + 1C8h offset = 303A_01C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								DDRMIX_PGC_PUP_ACK	DDRMIX_PGC_PDN_ACK	MEDIA_ISP_DWP_PGC_PUP_ACK	MEDIA_ISP_DWP_PGC_PDN_ACK	HSIOMIX_PGC_PUP_ACK	HSIOMIX_PGC_PDN_ACK	MIPI_PHY2_PGC_PUP_ACK	MIPI_PHY2_PGC_PDN_ACK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ACK_SEL_M7_PU1 field descriptions

Field	Description
31–8 -	This field is reserved. Reserved
7 DDRMIX_PGC_ PUP_ACK	Select power down acknowledge signal of DDRMIX PGC as the power up acknowledge for M7 LPM

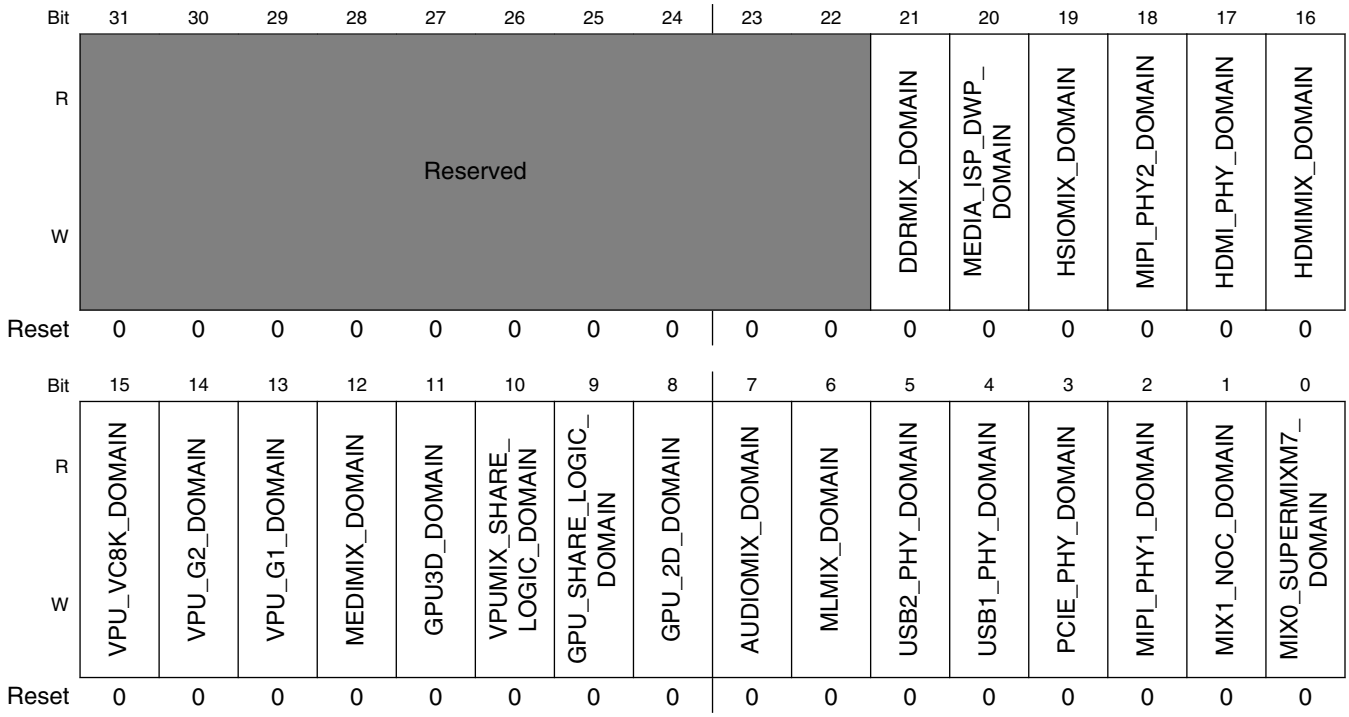
Table continues on the next page...

GPC_ACK_SEL_M7_PU1 field descriptions (continued)

Field	Description
6 DDRMIX_PGC_ PDN_ACK	Select power down acknowledge signal of DDRMIX PGC as the power down acknowledge for M7 LPM
5 MEDIA_ISP_ DWP_PGC_ PUP_ACK	Select power down acknowledge signal of MEDIA_ISP_DWP PGC as the power up acknowledge for M7 LPM
4 MEDIA_ISP_ DWP_PGC_ PDN_ACK	Select power down acknowledge signal of MEDIA_ISP_DWP PGC as the power down acknowledge for M7 LPM
3 HSIOMIX_PGC_ PUP_ACK	Select power down acknowledge signal of HSIOMIX PGC as the power up acknowledge for M7 LPM
2 HSIOMIX_PGC_ PDN_ACK	Select power down acknowledge signal of HSIOMIX PGC as the power down acknowledge for M7 LPM
1 MIPI_PHY2_ PGC_PUP_ACK	Select power down acknowledge signal of MIPI_PHY2 PGC as the power up acknowledge for M7 LPM
0 MIPI_PHY2_ PGC_PDN_ACK	Select power down acknowledge signal of MIPI_PHY2 PGC as the power down acknowledge for M7 LPM

5.2.10.72 PGC CPU A53 mapping (GPC_PGC_CPU_A53_MAPPING)

Address: 303A_0000h base + 1CCh offset = 303A_01CCh



GPC_PGC_CPU_A53_MAPPING field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21 DDRMIX_ DOMAIN	DDR mapping 0 Don’t map DDR to A53 domain 1 Map DDR to A53 domain
20 MEDIA_ISP_ DWP_DOMAIN	MEDIA_ISP_DWP_DOMAIN mapping 0 Don’t map MEDIA_ISP_DWP_DOMAIN to A53 domain 1 Map DDR to MEDIA_ISP_DWP_DOMAIN domain
19 HSIOMIX_ DOMAIN	HSIOMIX mapping 0 Don’t map HSIOMIX to A53 domain 1 Map HSIOMIX to A53 domain
18 MIPI_PHY2_ DOMAIN	MIPI PHY2 mapping 0 Don’t map MIPI PHY2 to A53 domain 1 Map MIPI PHY2 to A53 domain
17 HDMI_PHY_ DOMAIN	HDMI PHY mapping

Table continues on the next page...

GPC_PGC_CPU_A53_MAPPING field descriptions (continued)

Field	Description
	0 Don't map HDMI PHY to A53 domain 1 Map HDMI PHY to A53 domain
16 HDMIMIX_ DOMAIN	HDMI mapping 0 Don't map HDMI to A53 domain 1 Map HDMI to A53 domain
15 VPU_VC8K_ DOMAIN	VPU_VC8K mapping 0 Don't map VPU_VC8K to A53 domain 1 Map VPU_VC8K to A53 domain
14 VPU_G2_ DOMAIN	VPU_G2 mapping 0 Don't map VPU_G1 to A53 domain 1 Map VPU_G1 to A53 domain
13 VPU_G1_ DOMAIN	VPU_G1 mapping 0 Don't map VPU_G1 to A53 domain 1 Map VPU_G1 to A53 domain
12 MEDIMIX_ DOMAIN	MEDIMIX mapping 0 Don't map MEDIMIX to A53 domain 1 Map MEDIMIX to A53 domain
11 GPU3D_DOMAIN	GPU3D mapping 0 Don't map GPU2D to A53 domain 1 Map GPU2D to A53 domain
10 VPUMIX_ SHARE_LOGIC_ DOMAIN	VPUMIX Share Logic mapping 0 Don't map VPUMIX Share Logic to A53 domain 1 Map VPUMIX Share Logic to A53 domain
9 GPU_SHARE_ LOGIC_DOMAIN	GPU_SHARE_LOGIC mapping 0 Don't map GPU Share Logic to A53 domain 1 Map GPU Share Logic to A53 domain
8 GPU_2D_ DOMAIN	GPU2D mapping 0 Don't map GPU2D to A53 domain 1 Map GPU2D to A53 domain
7 AUDIOMIX_ DOMAIN	AUDIOMIX mapping 0 Don't map AUDIOMIX to A53 domain 1 Map AUDIOMIX to A53 domain
6 MLMIX_DOMAIN	MLMIX mapping 0 Don't map MLMIX to A53 domain 1 Map MLMIX to A53 domain
5 USB2_PHY_ DOMAIN	USB2_PHY mapping

Table continues on the next page...

GPC_PGC_CPU_A53_MAPPING field descriptions (continued)

Field	Description
	0 Don't map USB2_PHY to A53 domain 1 Map USB2_PHY to A53 domain
4 USB1_PHY_ DOMAIN	USB1_PHY mapping 0 Don't map USB1_PHY to A53 domain 1 Map USB1_PHY to A53 domain
3 PCIE_PHY_ DOMAIN	PCIE_PHY mapping 0 Don't map PCIE_PHY to A53 domain 1 Map PCIE_PHY to A53 domain
2 MIPI_PHY1_ DOMAIN	MIPI_PHY1 mapping 0 Don't map MIPI_PHY1 to A53 domain 1 Map MIPI_PHY1 to A53 domain
1 MIX1_NOC_ DOMAIN	MIX1 (NOC) mapping 0 Don't map NOC to A53 domain 1 Map NOC to A53 domain
0 MIX0_ SUPERMIXM7_ DOMAIN	MIX0 (SUPERMIXM7) mapping 0 Don't map M7 to A53 domain 1 Map M7 to A53 domain

5.2.10.73 PGC CPU M7 mapping (GPC_PGC_CPU_M7_MAPPING)

Address: 303A_0000h base + 1D0h offset = 303A_01D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved										DDRMIX_DOMAIN	MEDIA_ISP_DWP_DOMAIN	HSIOMIX_DOMAIN	MIPI_PHY2_DOMAIN	HDMI_PHY_DOMAIN	HDMIMIX_DOMAIN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VPU_VC8K_DOMAIN	VPU_G2_DOMAIN	VPU_G1_DOMAIN	MEDIMIX_DOMAIN	GPU3D_DOMAIN	VPUMIX_SHARE_LOGIC_DOMAIN	GPU_SHARE_LOGIC_DOMAIN	GPU_2D_DOMAIN	AUDIOMIX_DOMAIN	MLMIX_DOMAIN	USB2_PHY_DOMAIN	USB1_PHY_DOMAIN	PCIE_PHY_DOMAIN	MIPI_PHY1_DOMAIN	MIX1_NOC_DOMAIN	MIX0_SUPERMIXM7_DOMAIN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_PGC_CPU_M7_MAPPING field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21 DDRMIX_ DOMAIN	DDR mapping 0 Don’t map DDR to M7 domain 1 Map DDR to M7 domain
20 MEDIA_ISP_ DWP_DOMAIN	MEDIA_ISP_DWP_DOMAIN mapping 0 Don’t map MEDIA_ISP_DWP_DOMAIN to M7 domain 1 Map MEDIA_ISP_DWP_DOMAIN to M7 domain
19 HSIOMIX_ DOMAIN	HSIOMIX mapping 0 Don’t map HSIOMIX to M7 domain 1 Map HSIOMIX to M7 domain
18 MIPI_PHY2_ DOMAIN	MIPI PHY2 mapping 0 Don’t map MIPI PHY2 to M7 domain 1 Map MIPI PHY2 to M7 domain
17 HDMI_PHY_ DOMAIN	HDMI PHY mapping 0 Don’t map HDMI PHY to M7 domain 1 Map HDMI PHY to M7 domain
16 HDMIMIX_ DOMAIN	HDMI mapping 0 Don’t map HDMI to M7 domain 1 Map HDMI to M7 domain
15 VPU_VC8K_ DOMAIN	VPU_VC8K mapping 0 Don’t map VPU_VC8K to M7 domain 1 Map VPU_VC8K to M7 domain
14 VPU_G2_ DOMAIN	VPU_G2 mapping 0 Don’t map VPU_G1 to M7 domain 1 Map VPU_G1 to M7 domain
13 VPU_G1_ DOMAIN	VPU_G1 mapping 0 Don’t map VPU_G1 to M7 domain 1 Map VPU_G1 to M7 domain
12 MEDIMIX_ DOMAIN	MEDIMIX mapping 0 Don’t map MEDIMIX to M7 domain 1 Map MEDIMIX to M7 domain
11 GPU3D_DOMAIN	GPU3D mapping 0 Don’t map GPU2D to M7 domain 1 Map GPU2D to M7 domain

Table continues on the next page...

GPC_PGC_CPU_M7_MAPPING field descriptions (continued)

Field	Description
10 VPUMIX_ SHARE_LOGIC_ DOMAIN	VPUMIX Share Logic mapping 0 Don't map VPUMIX Share Logic to M7 domain 1 Map VPUMIX Share Logic to M7 domain
9 GPU_SHARE_ LOGIC_DOMAIN	GPU_SHARE_LOGIC mapping 0 Don't map GPU Share Logic to M7 domain 1 Map GPU Share Logic to M7 domain
8 GPU_2D_ DOMAIN	GPU2D mapping 0 Don't map GPU2D to M7 domain 1 Map GPU2D to M7 domain
7 AUDIOMIX_ DOMAIN	AUDIOMIX mapping 0 Don't map AUDIOMIX to M7 domain 1 Map AUDIOMIX to M7 domain
6 MLMIX_DOMAIN	MLMIX mapping 0 Don't map MLMIX to M7 domain 1 Map MLMIX to M7 domain
5 USB2_PHY_ DOMAIN	USB2_PHY mapping 0 Don't map USB2_PHY to M7 domain 1 Map USB2_PHY to M7 domain
4 USB1_PHY_ DOMAIN	USB1_PHY mapping 0 Don't map USB1_PHY to M7 domain 1 Map USB1_PHY to M7 domain
3 PCIE_PHY_ DOMAIN	PCIE_PHY mapping 0 Don't map PCIE_PHY to M7 domain 1 Map PCIE_PHY to M7 domain
2 MIPI_PHY1_ DOMAIN	MIPI_PHY1 mapping 0 Don't map MIPI_PHY1 to M7 domain 1 Map MIPI_PHY1 to M7 domain
1 MIX1_NOC_ DOMAIN	MIX1 (NOC) mapping 0 Don't map MIX1_NOC to M7 domain 1 Map MIX1_NOC to M7 domain
0 MIX0_ SUPERMIXM7_ DOMAIN	MIX0 (SUPERMIXM7) mapping 0 Don't map MIX0_SUPERMIXM7 to M7 domain 1 Map MIX0_SUPERMIXM7 to M7 domain

5.2.10.74 Slot configure register for CPUs (GPC_SLT*n*_CFG)

There are 27 slots in each SLTn_CFG(n = 0~26) that define the power up or power down behavior of one or more A53 core, NOC, or SCU PGC in each slot. This array contains slots 0 to 14, see Memory Map for slots 15 to 19.

In each “SLTn_cfg”, 2 bits (slt_cfg[1:0])are reserved for each PGC:

- 2’b01 (slot controller will power down relevant PGC in corresponding slot if hardware power down request asserted)
- 2’b10 (slot controller will power up relevant PGC in corresponding slot if hardware power up request asserted)
- 2’b00 or 2’b11 (not power down or power up behavior in relevant slot)

The specific bits assignment for each PGC is shown in the table below.

	PGCx	PGCx-1	..	PGC2	PGC1	PGC0
SLT0_CFG	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]
SLT1_CFG	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]
:	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]
SLTn_CFG	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]

Address: 303A_0000h base + 200h offset + (4d × i), where i=0d to 26d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		NOC_PUP_SLOT_CONTROL	NOC_PDN_SLOT_CONTROL	Reserved		SCU_PUP_SLOT_CONTROL	SCU_PDN_SLOT_CONTROL	CORE3_A53_PUP_SLOT_CONTROL	CORE3_A53_PDN_SLOT_CONTROL	CORE2_A53_PUP_SLOT_CONTROL	CORE2_A53_PDN_SLOT_CONTROL	CORE1_A53_PUP_SLOT_CONTROL	CORE1_A53_PDN_SLOT_CONTROL	CORE0_A53_PUP_SLOT_CONTROL	CORE0_A53_PDN_SLOT_CONTROL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_SLT*n*_CFG field descriptions

Field	Description
31–14 -	This field is reserved.
13 NOC_PUP_SLOT_CONTROL	NOC Power-up slot control
12 NOC_PDN_SLOT_CONTROL	NOC Power-down slot control
11–10 -	This field is reserved.
9 SCU_PUP_SLOT_CONTROL	SCU Power-up slot control
8 SCU_PDN_SLOT_CONTROL	SCU Power-down slot control
7 CORE3_A53_PUP_SLOT_CONTROL	CORE3 A53 Power-up slot control
6 CORE3_A53_PDN_SLOT_CONTROL	CORE3 A53 Power-down slot control
5 CORE2_A53_PUP_SLOT_CONTROL	CORE2 A53 Power-up slot control
4 CORE2_A53_PDN_SLOT_CONTROL	CORE2 A53 Power-down slot control
3 CORE1_A53_PUP_SLOT_CONTROL	CORE1 A53 Power-up slot control
2 CORE1_A53_PDN_SLOT_CONTROL	CORE1 A53 Power-down slot control
1 CORE0_A53_PUP_SLOT_CONTROL	CORE0 A53 Power-up slot control

Table continues on the next page...

GPC_SLT*n*_CFG field descriptions (continued)

Field	Description
0 CORE0_A53_ PDN_SLOT_ CONTROL	CORE0 A53 Power-down slot control

5.2.10.75 Slot configure register for PGC PUs (GPC_SLT*n*_CFG_PU)

There are 27 slots in each SLTn_CFG_PU (n = 0~26) that define the power up or power down behavior of PU PGC in each slot. See PGC power domains section for list of PGC PUs.

Address: 303A_0000h base + 280h offset + (8d × i), where i=0d to 26d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HDMI_PHY_PUP_SLOT_CONTROL	HDMI_PHY_PDN_SLOT_CONTROL	HDMIMIX_PUP_SLOT_CONTROL	HDMIMIX_PDN_SLOT_CONTROL	VPU_VC8K_PUP_SLOT_CONTROL	VPU_VC8K_PDN_SLOT_CONTROL	VPU_G2_PUP_SLOT_CONTROL	VPU_G2_PDN_SLOT_CONTROL	VPU_G1_PUP_SLOT_CONTROL	VPU_G1_PDN_SLOT_CONTROL	MEDIMIX_PUP_SLOT_CONTROL	MEDIMIX_PDN_SLOT_CONTROL	GPU3D_PUP_SLOT_CONTROL	GPU3D_PDN_SLOT_CONTROL	VPUMIX_SHARE_LOGIC_PUP_SLOT_CONTROL	VPUMIX_SHARE_LOGIC_PDN_SLOT_CONTROL
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPU_SHARE_LOGIC_PUP_SLOT_CONTROL	GPU_SHARE_LOGIC_PDN_SLOT_CONTROL	GPU_2D_PUP_SLOT_CONTROL	GPU_2D_PDN_SLOT_CONTROL	AUDIOMIX_PUP_SLOT_CONTROL	AUDIOMIX_PDN_SLOT_CONTROL	MLMIX_PUP_SLOT_CONTROL	MLMIX_PDN_SLOT_CONTROL	USB2_PHY_PUP_SLOT_CONTROL	USB2_PHY_PDN_SLOT_CONTROL	USB1_PHY_PUP_SLOT_CONTROL	USB1_PHY_PDN_SLOT_CONTROL	PCIE_PHY_PUP_SLOT_CONTROL	PCIE_PHY_PDN_SLOT_CONTROL	MIPI_PHY1_PUP_SLOT_CONTROL	MIPI_PHY1_PDN_SLOT_CONTROL
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_SLT*n*_CFG_PU field descriptions

Field	Description
31 HDMI_PHY_ PUP_SLOT_ CONTROL	HDMI_PHY power up slot control
30 HDMI_PHY_ PDN_SLOT_ CONTROL	HDMI_PHY power down slot control
29 HDMIMIX_PUP_ SLOT_ CONTROL	HDMIMIX power up slot control
28 HDMIMIX_PDN_ SLOT_ CONTROL	HDMIMIX power down slot control
27 VPU_VC8K_ PUP_SLOT_ CONTROL	VPU_VC8K power up slot control
26 VPU_VC8K_ PDN_SLOT_ CONTROL	VPU_VC8K power down slot control
25 VPU_G2_PUP_ SLOT_ CONTROL	VPU_G2 power up slot control
24 VPU_G2_PDN_ SLOT_ CONTROL	VPU_G2 power down slot control
23 VPU_G1_PUP_ SLOT_ CONTROL	VPU_G1 power up slot control
22 VPU_G1_PDN_ SLOT_ CONTROL	VPU_G1 power down slot control
21 MEDIMIX_PUP_ SLOT_ CONTROL	MEDIMIX power up slot control
20 MEDIMIX_PDN_ SLOT_ CONTROL	MEDIMIX power down slot control

Table continues on the next page...

GPC_SLT*n*_CFG_PU field descriptions (continued)

Field	Description
19 GPU3D_PUP_SLOT_CONTROL	GPU3D power up slot control
18 GPU3D_PDN_SLOT_CONTROL	GPU3D power down slot control
17 VPUMIX_SHARE_LOGIC_PUP_SLOT_CONTROL	VPUMIX_SHARE_LOGIC power up slot control
16 VPUMIX_SHARE_LOGIC_PDN_SLOT_CONTROL	VPUMIX_SHARE_LOGIC power down slot control
15 GPU_SHARE_LOGIC_PUP_SLOT_CONTROL	GPU_SHARE_LOGIC power up slot control
14 GPU_SHARE_LOGIC_PDN_SLOT_CONTROL	GPU_SHARE_LOGIC power down slot control
13 GPU_2D_PUP_SLOT_CONTROL	GPU_2D power up slot control
12 GPU_2D_PDN_SLOT_CONTROL	GPU_2D power down slot control
11 AUDIOMIX_PUP_SLOT_CONTROL	AUDIOMIX power up slot control
10 AUDIOMIX_PDN_SLOT_CONTROL	AUDIOMIX power down slot control
9 MLMIX_PUP_SLOT_CONTROL	MLMIX power up slot control

Table continues on the next page...

GPC_SLT*n*_CFG_PU field descriptions (continued)

Field	Description
8 MLMIX_PDN_ SLOT_ CONTROL	MLMIX power down slot control
7 USB2_PHY_ PUP_SLOT_ CONTROL	USB2_PHY power up slot control
6 USB2_PHY_ PDN_SLOT_ CONTROL	USB2_PHY power down slot control
5 USB1_PHY_ PUP_SLOT_ CONTROL	USB1_PHY power up slot control
4 USB1_PHY_ PDN_SLOT_ CONTROL	USB1_PHY power down slot control
3 PCIE_PHY_ PUP_SLOT_ CONTROL	PCIE_PHY power up slot control
2 PCIE_PHY_ PDN_SLOT_ CONTROL	PCIE_PHY power down slot control
1 MIPI_PHY1_ PUP_SLOT_ CONTROL	MIPI_PHY1 power up slot control
0 MIPI_PHY1_ PDN_SLOT_ CONTROL	MIPI_PHY1 power down slot control

5.2.10.76 Extended slot configure register for PGC PUs (GPC_SLT*n*_CFG_PU1)

Address: 303A_0000h base + 284h offset + (8d × i), where i=0d to 26d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						M7_PUP_SLOT_CONTROL	M7_PDN_SLOT_CONTROL	DDRMIX_PUP_SLOT_CONTROL	DDRMIX_PDN_SLOT_CONTROL	MEDIA_ISP_DWP_PUP_SLOT_CONTROL	MEDIA_ISP_DWP_PDN_SLOT_CONTROL	HSIOMIX_PUP_SLOT_CONTROL	HSIOMIX_PDN_SLOT_CONTROL	MIPI_PHY2_PUP_SLOT_CONTROL	MIPI_PHY2_PDN_SLOT_CONTROL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_SLT*n*_CFG_PU1 field descriptions

Field	Description
31–10 -	This field is reserved. Reserved
9 M7_PUP_SLOT_CONTROL	M7 power up slot control
8 M7_PDN_SLOT_CONTROL	M7 power down slot control
7 DDRMIX_PUP_SLOT_CONTROL	DDRMIX power up slot control
6 DDRMIX_PDN_SLOT_CONTROL	DDRMIX power down slot control

Table continues on the next page...

GPC_SLT*n*_CFG_PU1 field descriptions (continued)

Field	Description
5 MEDIA_ISP_ DWP_PUP_ SLOT_ CONTROL	MEDIA_ISP_DWP power up slot control
4 MEDIA_ISP_ DWP_PDN_ SLOT_ CONTROL	MEDIA_ISP_DWP power down slot control
3 HSIOMIX_PUP_ SLOT_ CONTROL	HSIOMIX power up slot control
2 HSIOMIX_PDN_ SLOT_ CONTROL	HSIOMIX power down slot control
1 MIPI_PHY2_ PUP_SLOT_ CONTROL	MIPI_PHY2 power up slot control
0 MIPI_PHY2_ PDN_SLOT_ CONTROL	MIPI_PHY2 power down slot control

5.2.11 GPC PGC Memory Map/Register Definition

There are numerous PGC inside GPCv2, with 4 different types: CPU/SCU/MIX/PU. Each PGC type has 4 different control words PGC_CTRL, PGC_PUPSCR, PGC_PDNSCR, and PGC_SR. Different PGC types may have different field definition in these four registers. There is another extra control word PGC_AUXSW which has different field definition for SCU type PGC.

The total GPC memory map is 4KB

Table 5-11. Memory Regions

Address Range(offset)	Region
0x000 - 0x3FF	GPC configuration register
0x400 - 0x7FF	Reserved
0x800 - 0x9FF	CPU and SCU type PGC register base address

Table continues on the next page...

Table 5-11. Memory Regions (continued)

Address Range(offset)	Region
0xA00 - 0xAFF	MIX type PGC register base address
0xB00 - 0xFFF	PU type PGC register base address

Each PGC (CPU type, MIX type, PU type) will occupy 64 Bytes address space, the specific base address of each PGC are listed as below.

- 0x800 ~ 0x83F: PGC for A53 core0
- 0x840 ~ 0x87F: PGC for A53 core1
- 0x880 ~ 0x8BF: PGC for A53 core2
- 0x8C0 ~ 0x8FF: PGC for A53 core3
- 0x900 ~ 0x93F: PGC for A53 SCU
- 0xA40 ~ 0xA7F: PGC for NOC mix
- 0xB00 ~ 0xB3F: PGC for MIPI PHY1 (PU0)
- 0xB40 ~ 0xB7F: PGC for PCIE PHY (PU1)
- 0xB80 ~ 0xBBF: PGC for USB1_PHY (PU2)
- 0xBC0 ~ 0xBFF: PGC for USB2_PHY (PU3)
- 0xC00 ~ 0xC3F: PGC for ML (PU4)
- 0xC40 ~ 0xC7F: PGC for AUDIO (PU5)
- 0xC80 ~ 0xCBF: PGC for GPU2D (PU6)
- 0xCC0 ~ 0xCFF: PGC for GPU Shared Logic (PU7)
- 0xD00 ~ 0xD3F: PGC for VPU Shared Logic (PU8)
- 0xD40 ~ 0xD7F: PGC for GPU3D (PU9)
- 0xD80 ~ 0xDBF: PGC for Medi (PU10)
- 0xDC0 ~ 0xDFF: PGC for VPU_G1 (PU11)
- 0xE00 ~ 0xE3F: PGC for VPU_G2 (PU12)
- 0xE40 ~ 0xE7F: PGC for VPU VC8000E (PU13)
- 0xE80 ~ 0xEBF: PGC for HDMI (PU14)
- 0xEC0 ~ 0xEFF: PGC for HDMI PHY (PU15)
- 0xF00 ~ 0xF3F: PGC for MIPI PHY2 (PU16)
- 0xF40 ~ 0xF7F: PGC for HSIO (PU17)
- 0xF80 ~ 0xFBF: PGC for ISP DWP (PU18)
- 0xFC0 ~ 0xFFF: PGC for DDR (PU19)

GPC_PGC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0800	GPC PGC Control Register for PGC CPUs (GPC_PGC_A53CORE0_CTRL)	32	R/W	0604_0202h	5.2.11.1/706

Table continues on the next page...

GPC_PGC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0804	GPC PGC Up Sequence Control Register (GPC_PGC_A53CORE0_PUPSCR)	32	R/W	0009_97C1h	5.2.11.2/708
303A_0808	GPC PGC Down Sequence Control Register (GPC_PGC_A53CORE0_PDNSCR)	32	R/W	2100_0801h	5.2.11.3/709
303A_080C	GPC PGC Status Register (GPC_PGC_A53CORE0_SR)	32	R/W	0000_1000h	5.2.11.4/710
303A_0840	GPC PGC Control Register for PGC CPUs (GPC_PGC_A53CORE1_CTRL)	32	R/W	0604_0202h	5.2.11.1/706
303A_0844	GPC PGC Up Sequence Control Register (GPC_PGC_A53CORE1_PUPSCR)	32	R/W	0009_97C1h	5.2.11.2/708
303A_0848	GPC PGC Down Sequence Control Register (GPC_PGC_A53CORE1_PDNSCR)	32	R/W	2100_0801h	5.2.11.3/709
303A_084C	GPC PGC Status Register (GPC_PGC_A53CORE1_SR)	32	R/W	0000_1000h	5.2.11.4/710
303A_0880	GPC PGC Control Register for PGC CPUs (GPC_PGC_A53CORE2_CTRL)	32	R/W	0604_0202h	5.2.11.1/706
303A_0884	GPC PGC Up Sequence Control Register (GPC_PGC_A53CORE2_PUPSCR)	32	R/W	0009_97C1h	5.2.11.2/708
303A_0888	GPC PGC Down Sequence Control Register (GPC_PGC_A53CORE2_PDNSCR)	32	R/W	2100_0801h	5.2.11.3/709
303A_088C	GPC PGC Status Register (GPC_PGC_A53CORE2_SR)	32	R/W	0000_1000h	5.2.11.4/710
303A_08C0	GPC PGC Control Register for PGC CPUs (GPC_PGC_A53CORE3_CTRL)	32	R/W	0604_0202h	5.2.11.1/706
303A_08C4	GPC PGC Up Sequence Control Register (GPC_PGC_A53CORE3_PUPSCR)	32	R/W	0009_97C1h	5.2.11.2/708
303A_08C8	GPC PGC Down Sequence Control Register (GPC_PGC_A53CORE3_PDNSCR)	32	R/W	2100_0801h	5.2.11.3/709
303A_08CC	GPC PGC Status Register (GPC_PGC_A53CORE3_SR)	32	R/W	0000_1000h	5.2.11.4/710
303A_0900	GPC PGC Control Register for PGC CPUs (GPC_PGC_A53SCU_CTRL)	32	R/W	0604_0202h	5.2.11.1/706
303A_0904	GPC PGC Up Sequence Control Register (GPC_PGC_A53SCU_PUPSCR)	32	R/W	0009_97C1h	5.2.11.2/708
303A_0908	GPC PGC Down Sequence Control Register (GPC_PGC_A53SCU_PDNSCR)	32	R/W	2100_0801h	5.2.11.3/709
303A_090C	GPC PGC Status Register (GPC_PGC_A53SCU_SR)	32	R/W	0000_1000h	5.2.11.4/710
303A_0A40	GPC PGC Control Register for PGC MIX. (GPC_PGC_NOC_MIX_CTRL)	32	R/W	0604_0202h	5.2.11.5/713
303A_0A44	GPC PGC Up Sequence Control Register (GPC_PGC_NOC_MIX_PUPSCR)	32	R/W	0009_97C1h	5.2.11.6/714
303A_0A48	GPC PGC Down Sequence Control Register (GPC_PGC_NOC_MIX_PDNSCR)	32	R/W	2100_0801h	5.2.11.7/715
303A_0A4C	GPC PGC Status Register (GPC_PGC_NOC_MIX_SR)	32	R/W	0000_1000h	5.2.11.8/716
303A_0B00	GPC PGC Control Register for PGC PUs (GPC_PGC_PU0_CTRL)	32	R/W	0604_0202h	5.2.11.9/719

Table continues on the next page...

GPC_PGC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0B04	GPC PGC Up Sequence Control Register (GPC_PGC_PU0_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0B08	GPC PGC Down Sequence Control Register (GPC_PGC_PU0_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0B0C	GPC PGC Status Register (GPC_PGC_PU0_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0B40	GPC PGC Control Register for PGC PUs (GPC_PGC_PU1_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0B44	GPC PGC Up Sequence Control Register (GPC_PGC_PU1_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0B48	GPC PGC Down Sequence Control Register (GPC_PGC_PU1_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0B4C	GPC PGC Status Register (GPC_PGC_PU1_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0B80	GPC PGC Control Register for PGC PUs (GPC_PGC_PU2_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0B84	GPC PGC Up Sequence Control Register (GPC_PGC_PU2_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0B88	GPC PGC Down Sequence Control Register (GPC_PGC_PU2_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0B8C	GPC PGC Status Register (GPC_PGC_PU2_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0BC0	GPC PGC Control Register for PGC PUs (GPC_PGC_PU3_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0BC4	GPC PGC Up Sequence Control Register (GPC_PGC_PU3_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0BC8	GPC PGC Down Sequence Control Register (GPC_PGC_PU3_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0BCC	GPC PGC Status Register (GPC_PGC_PU3_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0C00	GPC PGC Control Register for PGC PUs (GPC_PGC_PU4_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0C04	GPC PGC Up Sequence Control Register (GPC_PGC_PU4_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0C08	GPC PGC Down Sequence Control Register (GPC_PGC_PU4_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0C0C	GPC PGC Status Register (GPC_PGC_PU4_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0C40	GPC PGC Control Register for PGC PUs (GPC_PGC_PU5_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0C44	GPC PGC Up Sequence Control Register (GPC_PGC_PU5_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0C48	GPC PGC Down Sequence Control Register (GPC_PGC_PU5_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721

Table continues on the next page...

GPC_PGC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0C4C	GPC PGC Status Register (GPC_PGC_PU5_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0C80	GPC PGC Control Register for PGC PUs (GPC_PGC_PU6_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0C84	GPC PGC Up Sequence Control Register (GPC_PGC_PU6_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0C88	GPC PGC Down Sequence Control Register (GPC_PGC_PU6_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0C8C	GPC PGC Status Register (GPC_PGC_PU6_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0CC0	GPC PGC Control Register for PGC PUs (GPC_PGC_PU7_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0CC4	GPC PGC Up Sequence Control Register (GPC_PGC_PU7_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0CC8	GPC PGC Down Sequence Control Register (GPC_PGC_PU7_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0CCC	GPC PGC Status Register (GPC_PGC_PU7_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0D00	GPC PGC Control Register for PGC PUs (GPC_PGC_PU8_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0D04	GPC PGC Up Sequence Control Register (GPC_PGC_PU8_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0D08	GPC PGC Down Sequence Control Register (GPC_PGC_PU8_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0D0C	GPC PGC Status Register (GPC_PGC_PU8_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0D40	GPC PGC Control Register for PGC PUs (GPC_PGC_PU9_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0D44	GPC PGC Up Sequence Control Register (GPC_PGC_PU9_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0D48	GPC PGC Down Sequence Control Register (GPC_PGC_PU9_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0D4C	GPC PGC Status Register (GPC_PGC_PU9_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0D80	GPC PGC Control Register for PGC PUs (GPC_PGC_PU10_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0D84	GPC PGC Up Sequence Control Register (GPC_PGC_PU10_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0D88	GPC PGC Down Sequence Control Register (GPC_PGC_PU10_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0D8C	GPC PGC Status Register (GPC_PGC_PU10_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0DC0	GPC PGC Control Register for PGC PUs (GPC_PGC_PU11_CTRL)	32	R/W	0604_0202h	5.2.11.9/719

Table continues on the next page...

GPC_PGC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0DC4	GPC PGC Up Sequence Control Register (GPC_PGC_PU11_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0DC8	GPC PGC Down Sequence Control Register (GPC_PGC_PU11_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0DCC	GPC PGC Status Register (GPC_PGC_PU11_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0E00	GPC PGC Control Register for PGC PUs (GPC_PGC_PU12_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0E04	GPC PGC Up Sequence Control Register (GPC_PGC_PU12_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0E08	GPC PGC Down Sequence Control Register (GPC_PGC_PU12_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0E0C	GPC PGC Status Register (GPC_PGC_PU12_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0E40	GPC PGC Control Register for PGC PUs (GPC_PGC_PU13_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0E44	GPC PGC Up Sequence Control Register (GPC_PGC_PU13_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0E48	GPC PGC Down Sequence Control Register (GPC_PGC_PU13_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0E4C	GPC PGC Status Register (GPC_PGC_PU13_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0E80	GPC PGC Control Register for PGC PUs (GPC_PGC_PU14_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0E84	GPC PGC Up Sequence Control Register (GPC_PGC_PU14_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0E88	GPC PGC Down Sequence Control Register (GPC_PGC_PU14_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0E8C	GPC PGC Status Register (GPC_PGC_PU14_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0EC0	GPC PGC Control Register for PGC PUs (GPC_PGC_PU15_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0EC4	GPC PGC Up Sequence Control Register (GPC_PGC_PU15_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0EC8	GPC PGC Down Sequence Control Register (GPC_PGC_PU15_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0ECC	GPC PGC Status Register (GPC_PGC_PU15_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0F00	GPC PGC Control Register for PGC PUs (GPC_PGC_PU16_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0F04	GPC PGC Up Sequence Control Register (GPC_PGC_PU16_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0F08	GPC PGC Down Sequence Control Register (GPC_PGC_PU16_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721

Table continues on the next page...

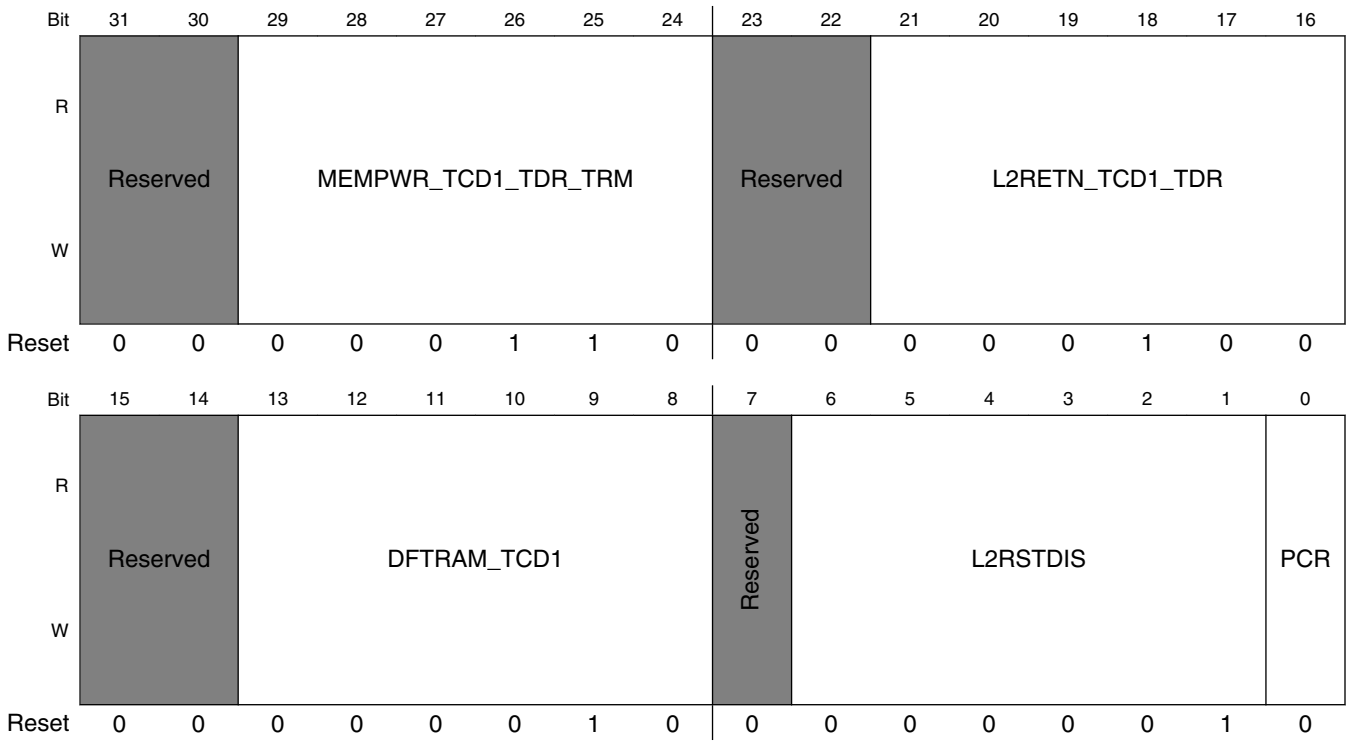
GPC_PGC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0F0C	GPC PGC Status Register (GPC_PGC_PU16_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0F40	GPC PGC Control Register for PGC PUs (GPC_PGC_PU17_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0F44	GPC PGC Up Sequence Control Register (GPC_PGC_PU17_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0F48	GPC PGC Down Sequence Control Register (GPC_PGC_PU17_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0F4C	GPC PGC Status Register (GPC_PGC_PU17_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0F80	GPC PGC Control Register for PGC PUs (GPC_PGC_PU18_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0F84	GPC PGC Up Sequence Control Register (GPC_PGC_PU18_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0F88	GPC PGC Down Sequence Control Register (GPC_PGC_PU18_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0F8C	GPC PGC Status Register (GPC_PGC_PU18_SR)	32	R/W	0000_1000h	5.2.11.12/ 722
303A_0FC0	GPC PGC Control Register for PGC PUs (GPC_PGC_PU19_CTRL)	32	R/W	0604_0202h	5.2.11.9/719
303A_0FC4	GPC PGC Up Sequence Control Register (GPC_PGC_PU19_PUPSCR)	32	R/W	0009_97C1h	5.2.11.10/ 720
303A_0FC8	GPC PGC Down Sequence Control Register (GPC_PGC_PU19_PDNSCR)	32	R/W	2100_0801h	5.2.11.11/ 721
303A_0FCC	GPC PGC Status Register (GPC_PGC_PU19_SR)	32	R/W	0000_1000h	5.2.11.12/ 722

5.2.11.1 GPC PGC Control Register for PGC CPUs (GPC_PGC_*n*CTRL)

GPC PGC Control Register for the PGC CPUs. See the PGC Memory Map for the assignments.

Address: 303A_0000h base + 800h offset + (64d × i), where i=0d to 4d



GPC_PGC_*n*CTRL field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29–24 MEMPWR_ TCD1_TDR_ TRM	After scu pdn_req, count this value to assert A53 mempwr to 1'b1 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
23–22 -	This field is reserved. Reserved
21–16 L2RETN_TCD1_ TDR	After scu pdn_req, count this value to assert A53 l2retn to 1'b0 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
15–14 -	This field is reserved. Reserved

Table continues on the next page...

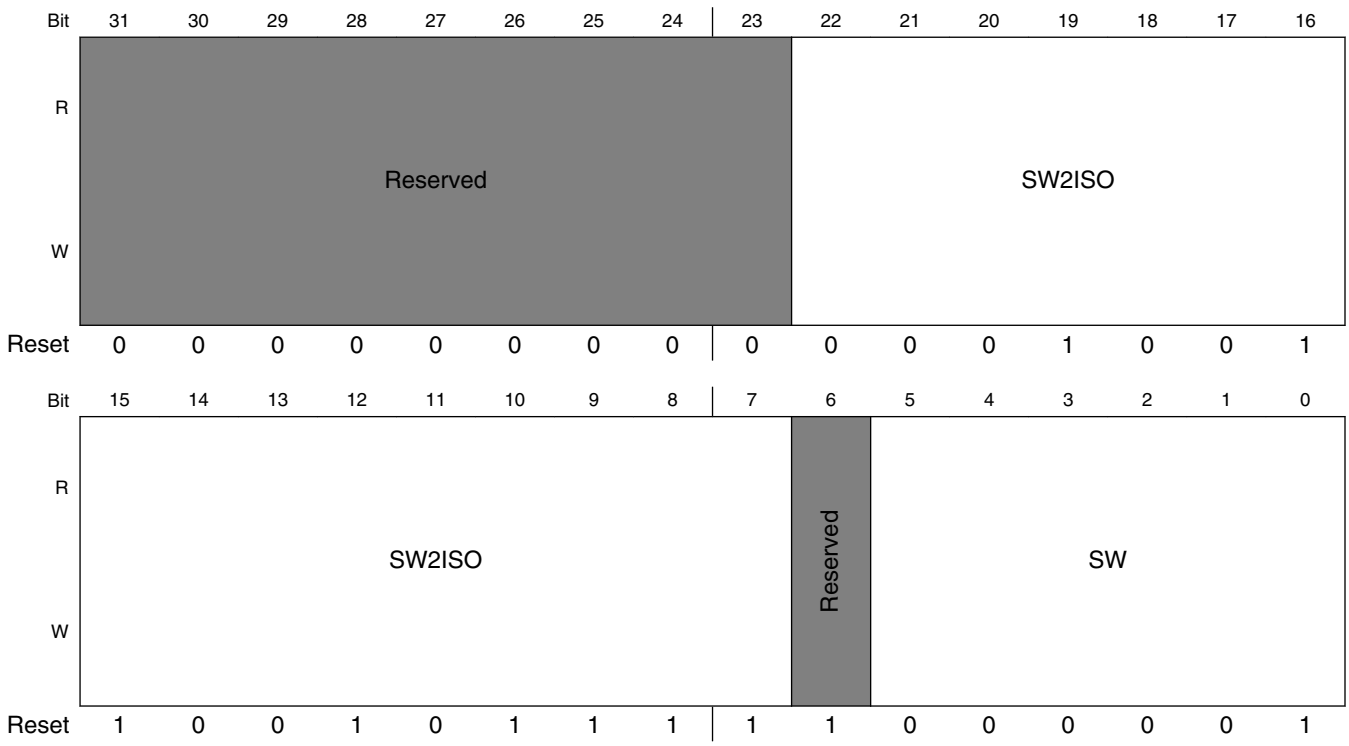
GPC_PGC_*n*CTRL field descriptions (continued)

Field	Description
13–8 DFTRAM_TCD1	After scu pdn_req, count this value to assert A53 dftram to 1'b1 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
7 -	This field is reserved. Reserved
6–1 L2RSTDIS	After scu pdn_req, count this value to assert A53 l2rstdis to 1'b1, it will be clear automatically once any of A53 core0/core1/core2/core3 is wakeup **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
0 PCR	Power Control **NOTE:** PCR must not change from power-down request (pdn_req) assertion until the target subsystem is completely powered up. 0 Do not switch off power even if pdn_req is asserted. 1 Switch off power when pdn_req is asserted.

5.2.11.2 GPC PGC Up Sequence Control Register (GPC_PGC_*n*PUPSCR)

GPC PGC Up Sequence Control Register

Address: 303A_0000h base + 804h offset + (64d × i), where i=0d to 4d



GPC_PGC_*n*PUPSCR field descriptions

Field	Description
31–23 -	This field is reserved. Reserved
22–7 SW2ISO	After asserting switch_b, the PGC waits a number of clocks equal to the value of SW2ISO before negating isolation.
6 -	This field is reserved. Reserved
SW	After a power-up request (pup_req assertion), the PGC waits a number of clocks equal to the value of SW before asserting switch_b **NOTE:** SW must not be programmed to zero.

5.2.11.3 GPC PGC Down Sequence Control Register (GPC_PGC_*n*PDNSCR)

GPC PGC Down Sequence Control Register

Address: 303A_0000h base + 808h offset + (64d × i), where i=0d to 4d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		ISO2SW						Reserved		ISO					
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

GPC_PGC_*n*PDNSCR field descriptions

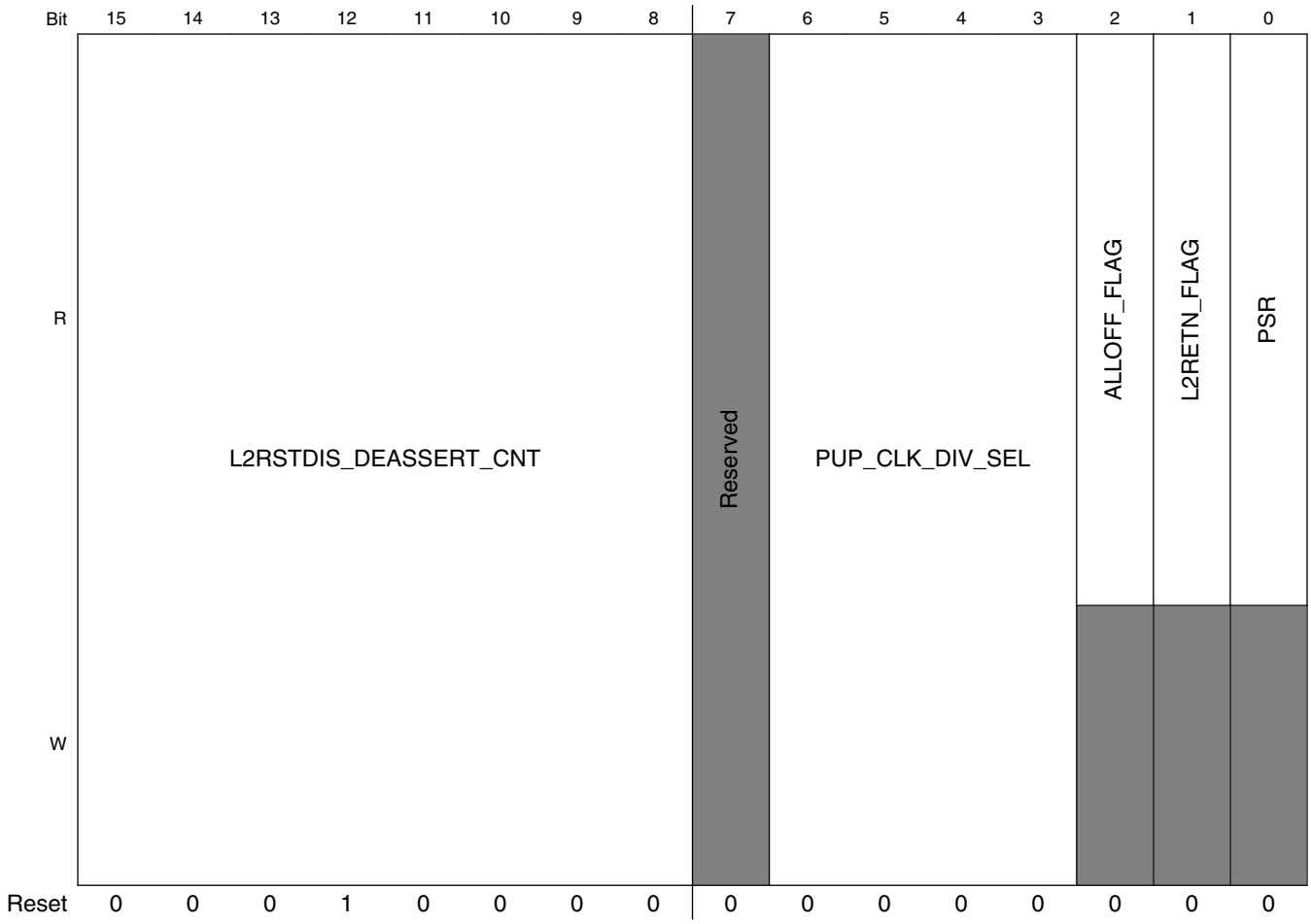
Field	Description
31–14 -	This field is reserved. Reserved
13–8 ISO2SW	After asserting isolation(by pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO2SW before negating switch_b **NOTE:** ISO2SW must not be programmed to zero.
7–6 -	This field is reserved. Reserved
ISO	After a power-down request (pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO before asserting isolation **NOTE:** ISO must not be programmed to zero.

5.2.11.4 GPC PGC Status Register (GPC_PGC_*n*SR)

GPC PGC Status Register

Address: 303A_0000h base + 80Ch offset + (64d × i), where i=0d to 4d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved														L2RSTDIS_DEASSERT_CNT	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



GPC_PGC_*n*SR field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17–8 L2RSTDIS_ DEASSERT_ CNT	Count this value to de-assert L2RSTDISABLE to LOW after CPU0 or CPU1 power up **NOTE:** This value can't be programmed to zero (This register control only for SCU Type PGC)
7 -	This field is reserved. Reserved
6–3 PUP_CLK_DIV_ SEL	Clock divider select for the clock of power up counter(count_clk is 32KHz for CPU/SCU type PGC, ipg_clk(66MHz) for MIX/PU Type PGC) 0000 1 0001 1/2 count_clk 0010 1/4 count_clk 0011 1/8 count_clk 0100 1/16 count_clk 0101 1/32 count_clk 0110 1/64 count_clk 0111 1/128 count_clk

Table continues on the next page...

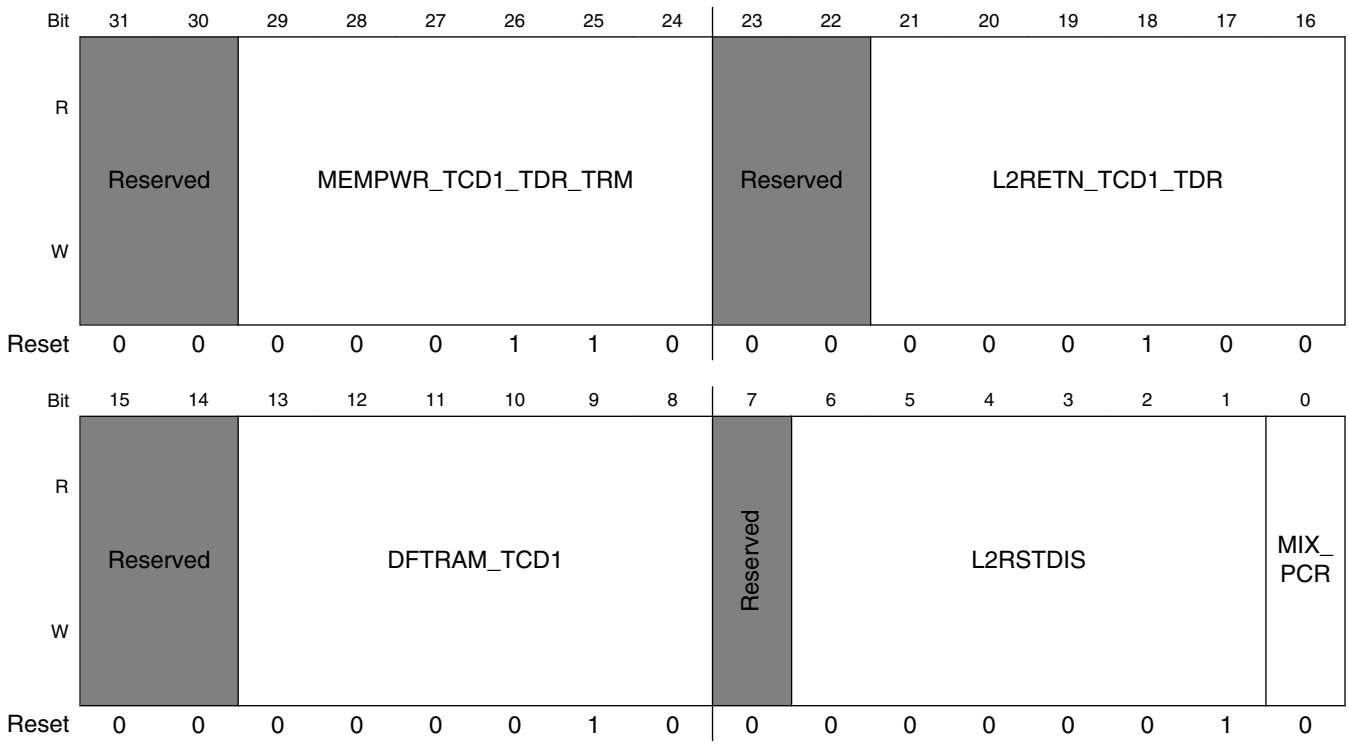
GPC_PGC_*n*SR field descriptions (continued)

Field	Description
	1000 1/256 count_clk 1001 1/512 count_clk 1010 1/1024 count_clk 1011 1/2056 count_clk 1100 1/4096 count_clk 1101 1/8192 count_clk 1110 1/16384 count_clk 1111 1/32768 count_clk
2 ALLOFF_FLAG	All-off flag. **NOTE:** Software should write “1” to clear this flag after A53 is wakeup from ALL_OFF mode, otherwise, it will always keep to 1 (This register control only for SCU Type PGC) 0 A53 is not wakeup from ALL_OFF mode. 1 A53 is wakeup from ALL_OFF mode.
1 L2RETN_FLAG	L2 Retention Flag **NOTE:** Software should write “1” to clear this flag after A53 is wakeup from L2 retention mode, otherwise it will always keep to 1 (This register control only for SCU Type PGC) 0 A53 is not wakeup from L2 retention mode. 1 A53 is wakeup from L2 retention mode.
0 PSR	Power status. When in functional (or software-controlled debug) mode, PGC hardware sets PSR as soon as any of the power control output changes its state to one. Write one to clear this bit. Software should clear this bit after power up; otherwise, PSR continues to reflect the power status of the initial power down. 0 The target subsystem was not powered down for the previous power-down request. 1 The target subsystem was powered down for the previous power-down request.

5.2.11.5 GPC PGC Control Register for PGC MIX. (GPC_PGC_NOC_MIX_CTRL)

GPC PGC Control Register.

Address: 303A_0000h base + A40h offset = 303A_0A40h



GPC_PGC_NOC_MIX_CTRL field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29–24 MEMPWR_TCD1_TDR_TRM	After scu pdn_req, count this value to assert A53 mempwr to 1'b1 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
23–22 -	This field is reserved. Reserved
21–16 L2RETN_TCD1_TDR	After scu pdn_req, count this value to assert A53 l2retn to 1'b0 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
15–14 -	This field is reserved. Reserved

Table continues on the next page...

GPC_PGC_NOC_MIX_CTRL field descriptions (continued)

Field	Description
13–8 DFTRAM_TCD1	After scu pdn_req, count this value to assert A53 dftram to 1'b1 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
7 -	This field is reserved. Reserved
6–1 L2RSTDIS	After scu pdn_req, count this value to assert A53 l2rstdis to 1'b1, it will be clear automatically once any of A53 core0/core1/core2/core3 is wakeup **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
0 MIX_PCR	Power Control **NOTE:** PCR must not change from power-down request (pdn_req) assertion until the target subsystem is completely powered up. 0 Do not switch off power even if pdn_req is asserted. 1 Switch off power when pdn_req is asserted.

5.2.11.6 GPC PGC Up Sequence Control Register (GPC_PGC_NOC_MIX_PUPSCR)

GPC PGC Up Sequence Control Register

Address: 303A_0000h base + A44h offset = 303A_0A44h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									SW2ISO						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW2ISO									PUP_WAIT_SCALL_OUT	Reserved					
Reset	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0	1

GPC_PGC_NOC_MIX_PUPSCR field descriptions

Field	Description
31–23 -	This field is reserved. Reserved
22–7 SW2ISO	After asserting switch_b, the PGC waits a number of clocks equal to the value of SW2ISO before negating isolation.
6 PUP_WAIT_ SCALL_OUT	After SCALL asserting to 1'b0, wait handshake signal SCALL_OUT to return to 1'b0 (This register control only for MIX Type PGC)
-	This field is reserved. Reserved

5.2.11.7 GPC PGC Down Sequence Control Register (GPC_PGC_NOC_MIX_PDNSCR)

GPC PGC Down Sequence Control Register

Address: 303A_0000h base + A48h offset = 303A_0A48h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		ISO2SW						Reserved		ISO					
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

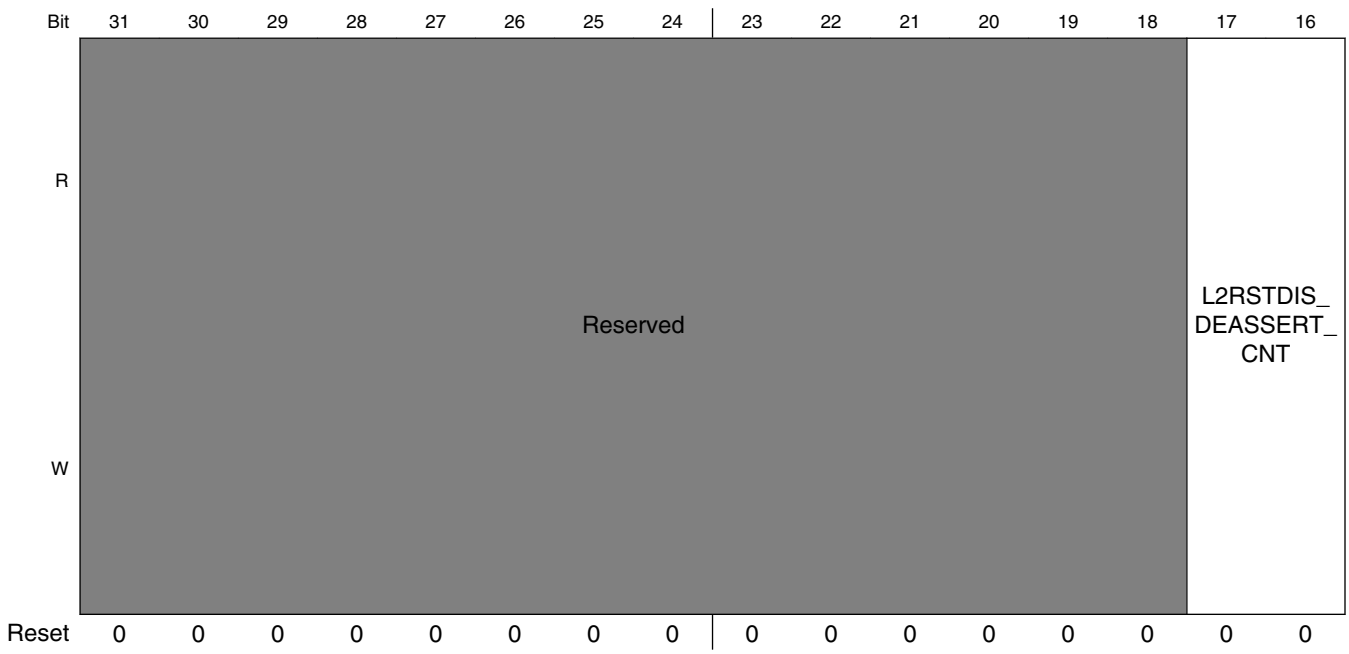
GPC_PGC_NOC_MIX_PDNSCR field descriptions

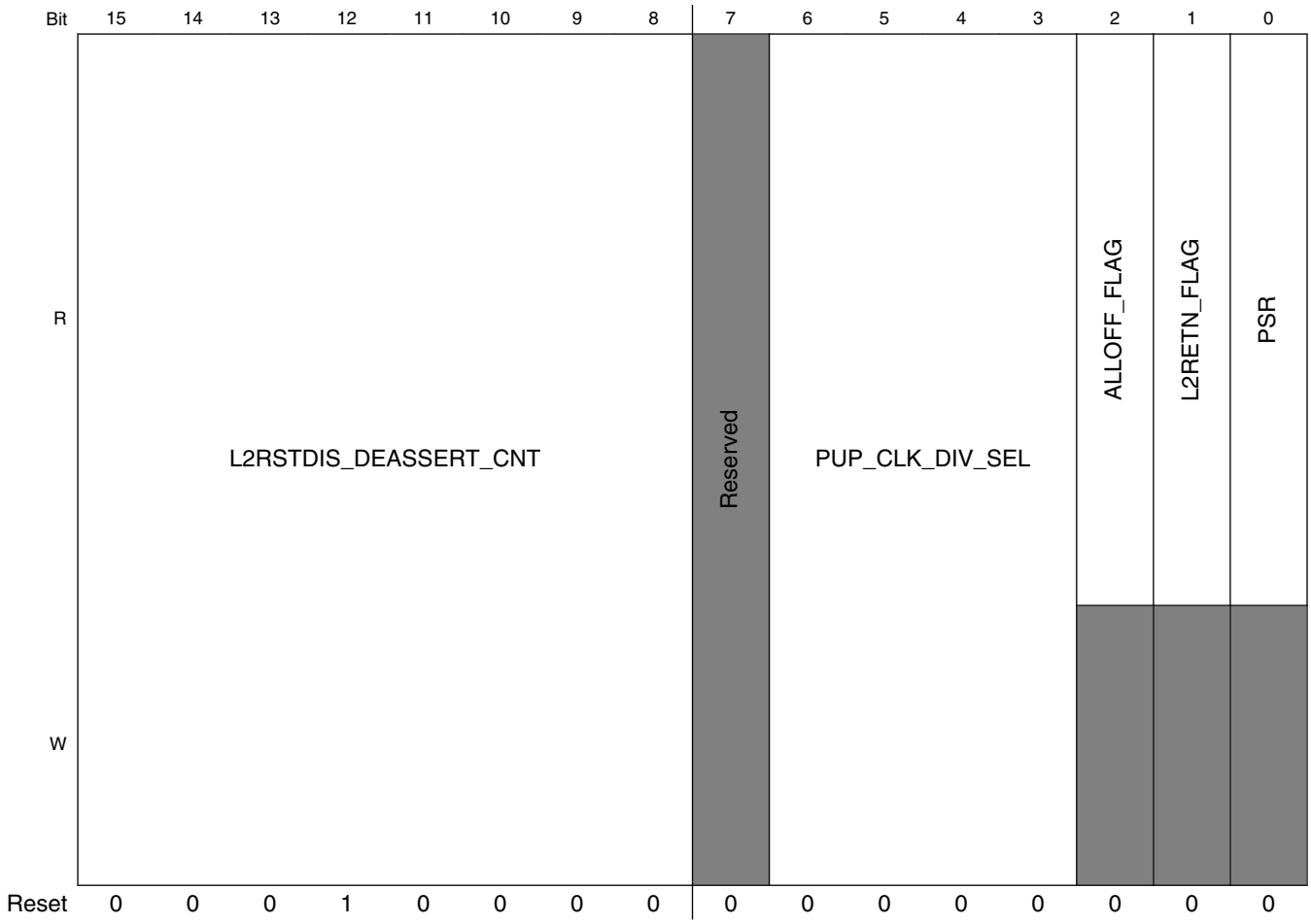
Field	Description
31–14 -	This field is reserved. Reserved
13–8 ISO2SW	After asserting isolation(by pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO2SW before negating switch_b **NOTE:** ISO2SW must not be programmed to zero.
7–6 -	This field is reserved. Reserved
ISO	After a power-down request (pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO before asserting isolation **NOTE:** ISO must not be programmed to zero.

5.2.11.8 GPC PGC Status Register (GPC_PGC_NOC_MIX_SR)

GPC PGC Status Register

Address: 303A_0000h base + A4Ch offset = 303A_0A4Ch





GPC_PGC_NOC_MIX_SR field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17–8 L2RSTDIS_ DEASSERT_ CNT	Count this value to de-assert L2RSTDISABLE to LOW after CPU0 or CPU1 power up **NOTE:** This value can't be programmed to zero (This register control only for SCU Type PGC)
7 -	This field is reserved. Reserved
6–3 PUP_CLK_DIV_ SEL	Clock divider select for the clock of power up counter(count_clk is 32KHz for CPU/SCU type PGC, ipg_clk(66MHz) for MIX/PU Type PGC) 0000 1 0001 1/2 count_clk 0010 1/4 count_clk 0011 1/8 count_clk 0100 1/16 count_clk 0101 1/32 count_clk 0110 1/64 count_clk 0111 1/128 count_clk

Table continues on the next page...

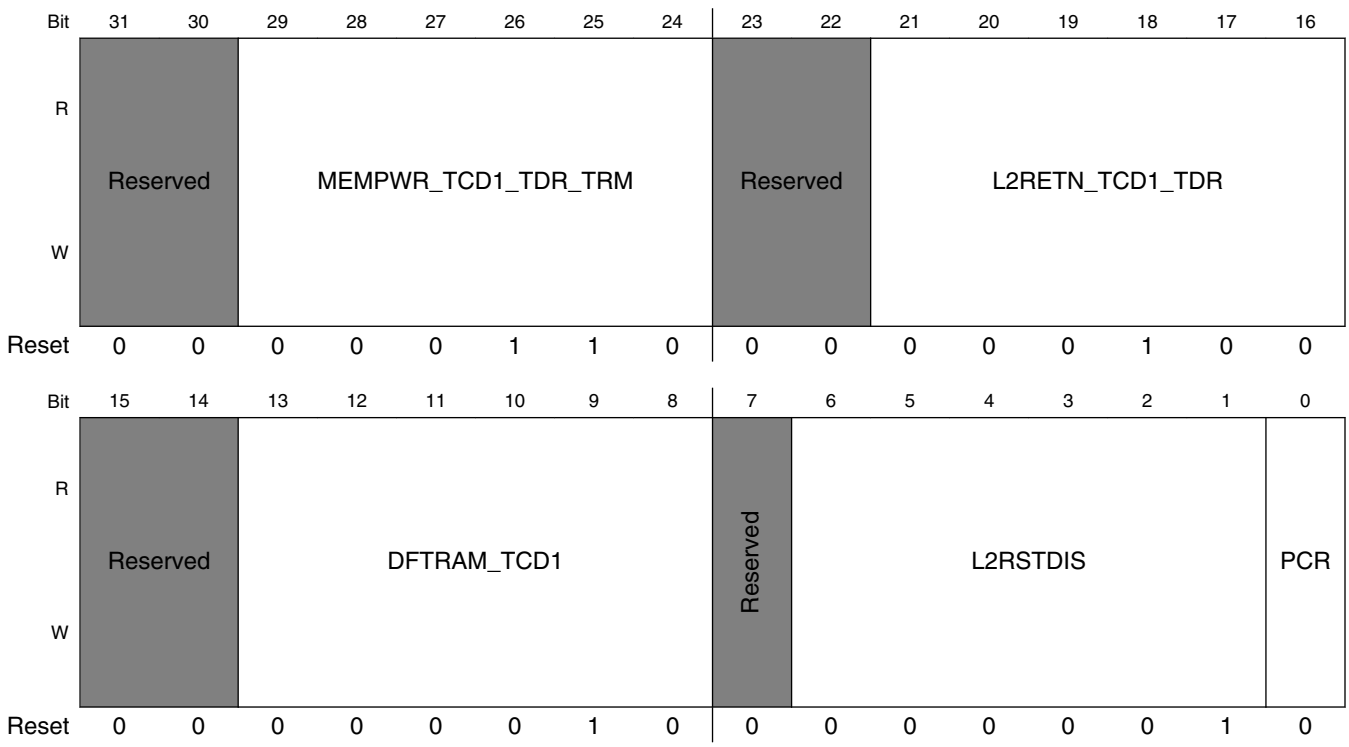
GPC_PGC_NOC_MIX_SR field descriptions (continued)

Field	Description
	1000 1/256 count_clk 1001 1/512 count_clk 1010 1/1024 count_clk 1011 1/2056 count_clk 1100 1/4096 count_clk 1101 1/8192 count_clk 1110 1/16384 count_clk 1111 1/32768 count_clk
2 ALLOFF_FLAG	All-off flag. **NOTE:** Software should write “1” to clear this flag after A53 is wakeup from ALL_OFF mode, otherwise, it will always keep to 1 (This register control only for SCU Type PGC) 0 A53 is not wakeup from ALL_OFF mode. 1 A53 is wakeup from ALL_OFF mode.
1 L2RETN_FLAG	L2 Retention Flag **NOTE:** Software should write “1” to clear this flag after A53 is wakeup from L2 retention mode, otherwise it will always keep to 1 (This register control only for SCU Type PGC) 0 A53 is not wakeup from L2 retention mode. 1 A53 is wakeup from L2 retention mode.
0 PSR	Power status. When in functional (or software-controlled debug) mode, PGC hardware sets PSR as soon as any of the power control output changes its state to one. Write one to clear this bit. Software should clear this bit after power up; otherwise, PSR continues to reflect the power status of the initial power down. 0 The target subsystem was not powered down for the previous power-down request. 1 The target subsystem was powered down for the previous power-down request.

5.2.11.9 GPC PGC Control Register for PGC PUs (GPC_PGC_*n*CTRL)

GPC PGC Control Register for the PUs. See the PGC Memory Map for the assignments.

Address: 303A_0000h base + B00h offset + (64d × i), where i=0d to 19d



GPC_PGC_*n*CTRL field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29–24 MEMPWR_ TCD1_TDR_ TRM	After scu pdn_req, count this value to assert A53 mempwr to 1'b1 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
23–22 -	This field is reserved. Reserved
21–16 L2RETN_TCD1_ TDR	After scu pdn_req, count this value to assert A53 l2retn to 1'b0 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
15–14 -	This field is reserved. Reserved
13–8 DFTRAM_TCD1	After scu pdn_req, count this value to assert A53 dftram to 1'b1

Table continues on the next page...

GPC_PGC_*n*CTRL field descriptions (continued)

Field	Description
	NOTE: Can't be programmed to zero (This register control only for SCU Type PGC)
7 -	This field is reserved. Reserved
6–1 L2RSTDIS	After scu pdn_req, count this value to assert A53 l2rstdis to 1'b1, it will be clear automatically once any of A53 core0/core1/core2/core3 is wakeup **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
0 PCR	Power Control **NOTE:** PCR must not change from power-down request (pdn_req) assertion until the target subsystem is completely powered up. 0 Do not switch off power even if pdn_req is asserted. 1 Switch off power when pdn_req is asserted.

5.2.11.10 GPC PGC Up Sequence Control Register (GPC_PGC_*n*PUPSCR)

GPC PGC Up Sequence Control Register

Address: 303A_0000h base + B04h offset + (64d × i), where i=0d to 19d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									SW2ISO						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW2ISO								Reserved	SW						
Reset	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0	1

GPC_PGC_*n*PUPSCR field descriptions

Field	Description
31–23 -	This field is reserved. Reserved
22–7 SW2ISO	After asserting switch_b, the PGC waits a number of clocks equal to the value of SW2ISO before negating isolation.
6 -	This field is reserved. Reserved
SW	After a power-up request (pup_req assertion), the PGC waits a number of clocks equal to the value of SW before asserting switch_b **NOTE:** SW must not be programmed to zero.

5.2.11.11 GPC PGC Down Sequence Control Register (GPC_PGC_*n*PDNSCR)

GPC PGC Down Sequence Control Register

Address: 303A_0000h base + B08h offset + (64d × i), where i=0d to 19d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		ISO2SW						Reserved		ISO					
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

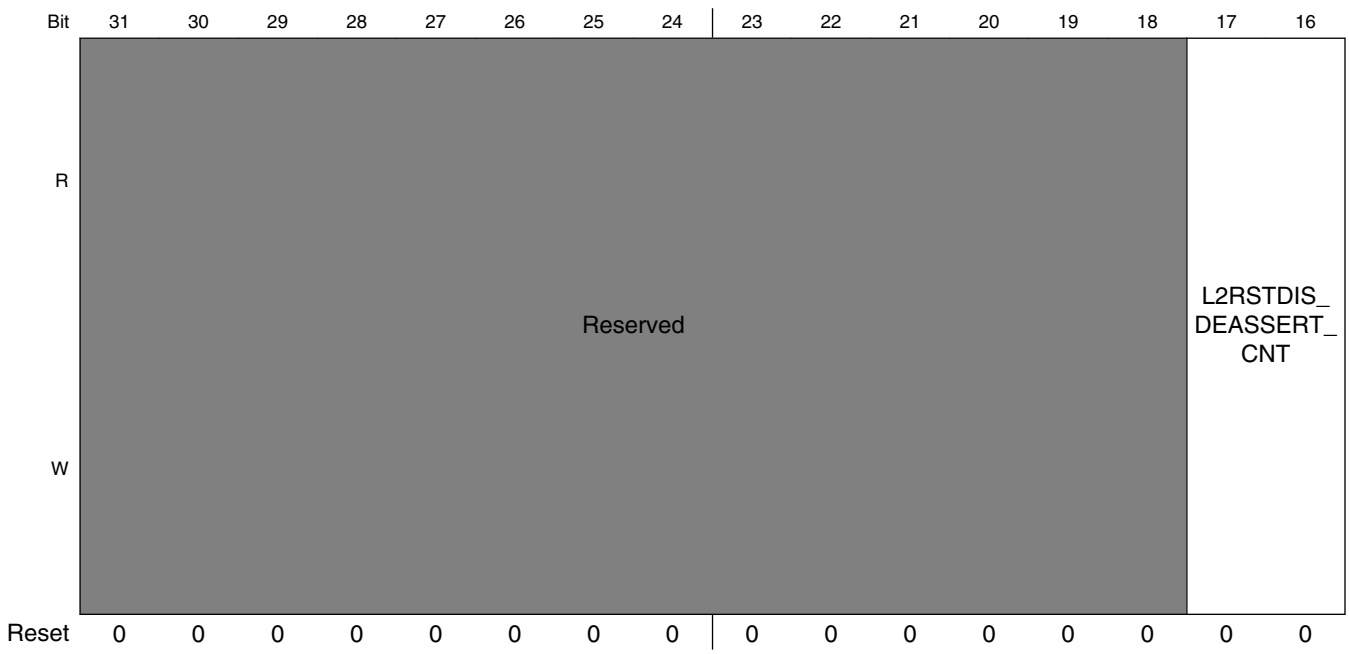
GPC_PGC_*n*PDNSCR field descriptions

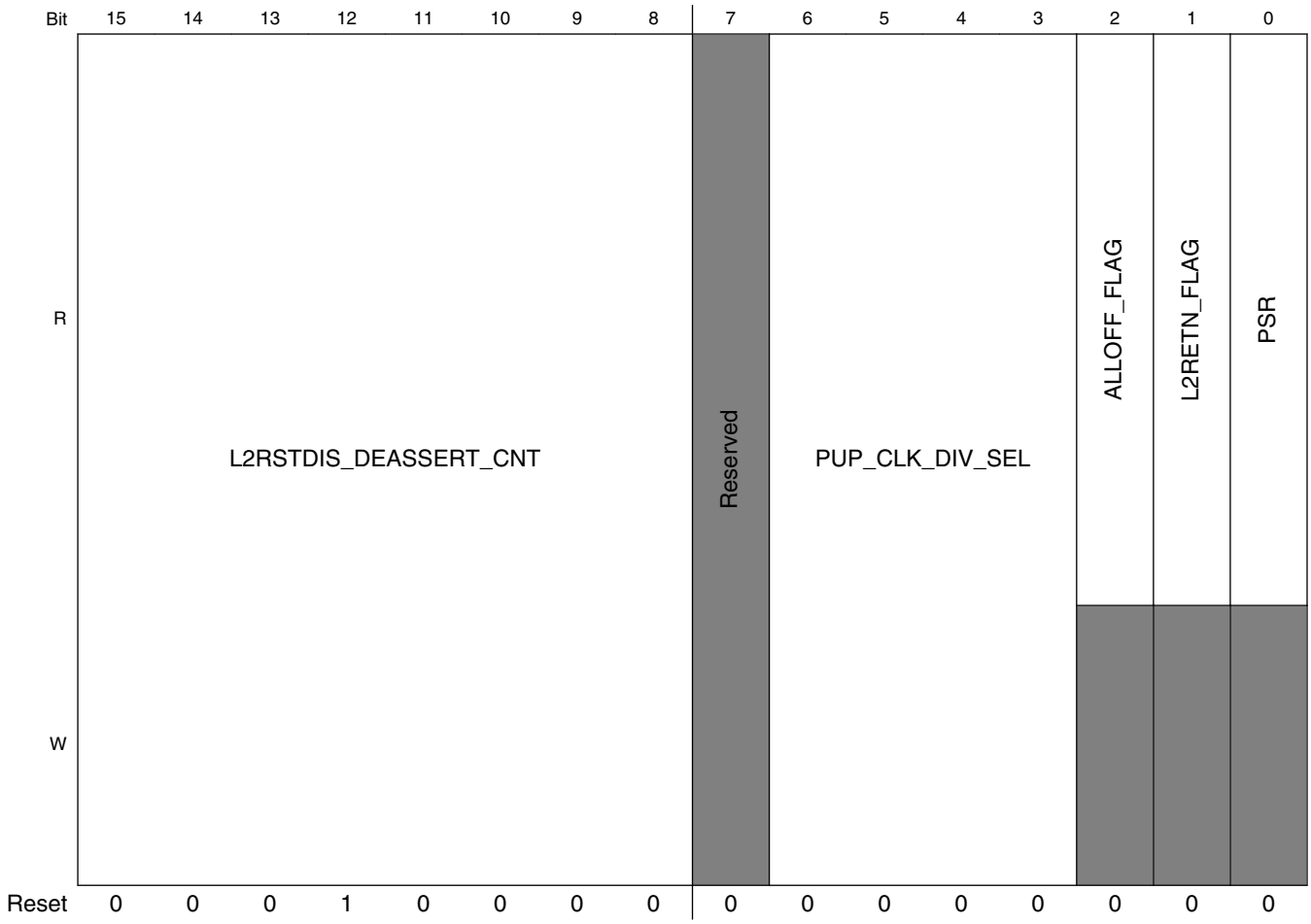
Field	Description
31–14 -	This field is reserved. Reserved
13–8 ISO2SW	After asserting isolation(by pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO2SW before negating switch_b **NOTE:** ISO2SW must not be programmed to zero.
7–6 -	This field is reserved. Reserved
ISO	After a power-down request (pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO before asserting isolation **NOTE:** ISO must not be programmed to zero.

5.2.11.12 GPC PGC Status Register (GPC_PGC_*n*SR)

GPC PGC Status Register

Address: 303A_0000h base + B0Ch offset + (64d × i), where i=0d to 19d





GPC_PGC_*n*SR field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17–8 L2RSTDIS_ DEASSERT_ CNT	Count this value to de-assert L2RSTDISABLE to LOW after CPU0 or CPU1 power up **NOTE:** This value can't be programmed to zero (This register control only for SCU Type PGC)
7 -	This field is reserved. Reserved
6–3 PUP_CLK_DIV_ SEL	Clock divider select for the clock of power up counter(count_clk is 32KHz for CPU/SCU type PGC, ipg_clk(66MHz) for MIX/PU Type PGC) 0000 1 0001 1/2 count_clk 0010 1/4 count_clk 0011 1/8 count_clk 0100 1/16 count_clk 0101 1/32 count_clk 0110 1/64 count_clk 0111 1/128 count_clk

Table continues on the next page...

GPC_PGC_*n*SR field descriptions (continued)

Field	Description
	1000 1/256 count_clk 1001 1/512 count_clk 1010 1/1024 count_clk 1011 1/2056 count_clk 1100 1/4096 count_clk 1101 1/8192 count_clk 1110 1/16384 count_clk 1111 1/32768 count_clk
2 ALLOFF_FLAG	All-off flag. **NOTE:** Software should write “1” to clear this flag after A53 is wakeup from ALL_OFF mode, otherwise, it will always keep to 1 (This register control only for SCU Type PGC) 0 A53 is not wakeup from ALL_OFF mode. 1 A53 is wakeup from ALL_OFF mode.
1 L2RETN_FLAG	L2 Retention Flag **NOTE:** Software should write “1” to clear this flag after A53 is wakeup from L2 retention mode, otherwise it will always keep to 1 (This register control only for SCU Type PGC) 0 A53 is not wakeup from L2 retention mode. 1 A53 is wakeup from L2 retention mode.
0 PSR	Power status. When in functional (or software-controlled debug) mode, PGC hardware sets PSR as soon as any of the power control output changes its state to one. Write one to clear this bit. Software should clear this bit after power up; otherwise, PSR continues to reflect the power status of the initial power down. 0 The target subsystem was not powered down for the previous power-down request. 1 The target subsystem was powered down for the previous power-down request.

5.3 Crystal Oscillator (XTALOSC)

5.3.1 Overview

The chip has two XTAL modules, 24MHz XTAL module and 32KHz XTAL module. The 24MHz XTAL module is instantiated from the XTAL IP, which includes:

- 24MHz crystal oscillator to generate reference clock
- Digital control logics for the XTAL

The 32KHz XTAL module uses a different IP and it is used as the clock source for the RTC, located in the SNVS.

5.3.1.1 Block Diagram

The figure below shows the OSC IP integration diagram:

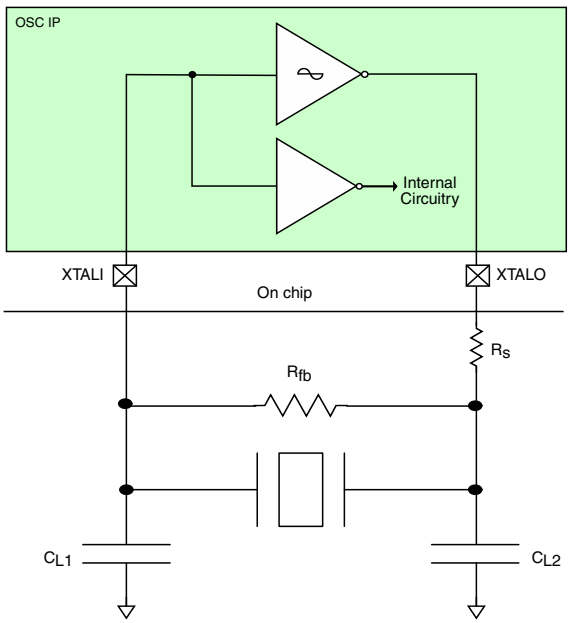


Figure 5-17. OSC IP Integration Diagram

5.3.2 Functional Description

The 24MHz oscillator will be used as the primary clock source for the PLLs to generate the clock for CPU, BUS, and high-speed interfaces. For all PLLs, the 24MHz clock from the oscillator can be used as the PLL reference clock directly.

The OSC IP used by the 24MHz XTAL module has three modes, Internal clock generation mode, External clock receive mode and Retention mode.

During internal clock generation mode, a suitable quartz crystal is connected between PADI and PADO to generate the clock signal at the CK pin.

During external clock generation mode, the cell acts like a buffer, reflecting the PADI signal at CK.

The RTO (retention enable) signal retains the previous state of all the core input control signals. Logic at the RTO signal enables the retention operation.

Each XTAL module supports the following modes through register configuration:

- Normal oscillator mode - In normal mode, the XTAL IP generates stable square wave based on the crystal oscillator input.
- Bypass mode - In bypass mode, an external clock can be input through the XTAL pad.

5.3.3 Memory Map and Register Definition

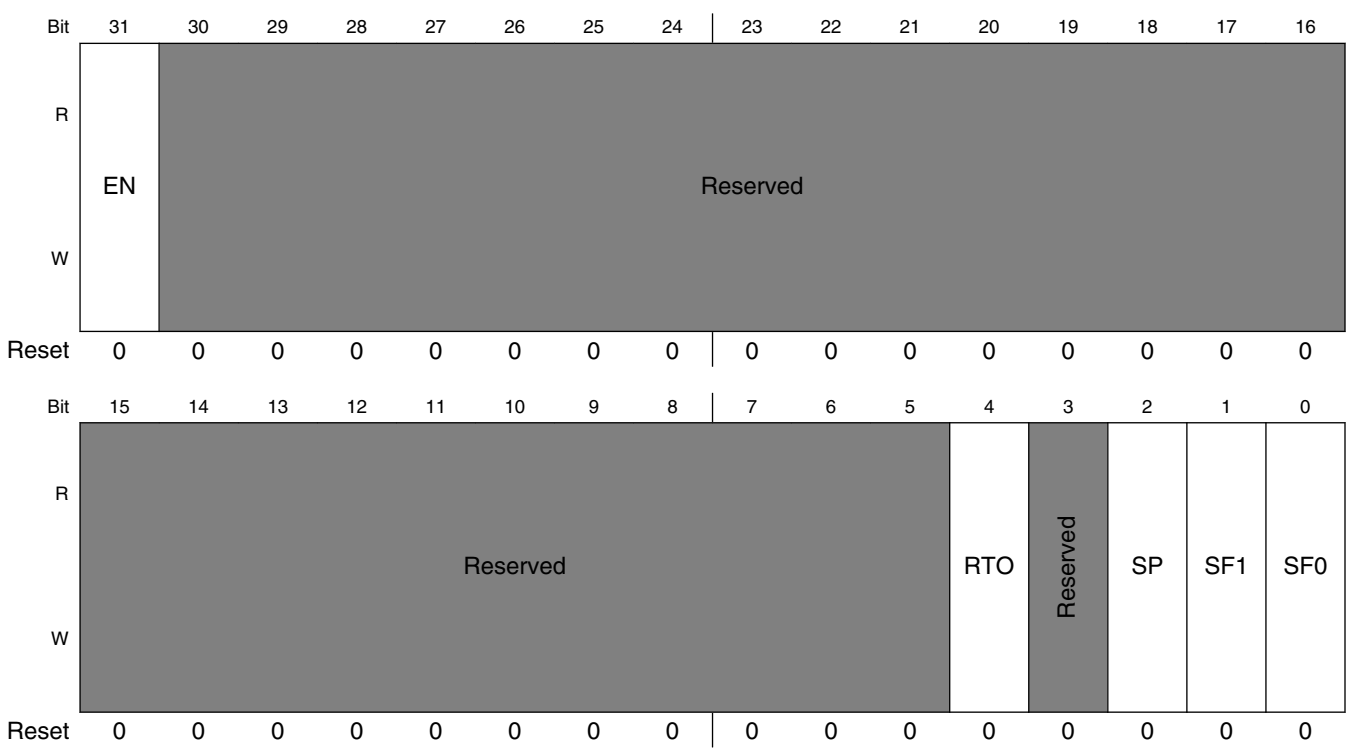
XTALOSC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3027_0000	OSC Normal Clock Generation Control Register0 (XTALOSC_SYS_OSCNML_CTL0)	32	R/W	0000_0000h	5.3.3.1/726
3027_0004	OSC Normal Clock Generation Control Register1 (XTALOSC_SYS_OSCNML_CTL1)	32	R/W	0000_0000h	5.3.3.2/727

5.3.3.1 OSC Normal Clock Generation Control Register0 (XTALOSC_SYS_OSCNML_CTL0)

OSC Normal Clock Generation Control Register0

Address: 3027_0000h base + 0h offset = 3027_0000h



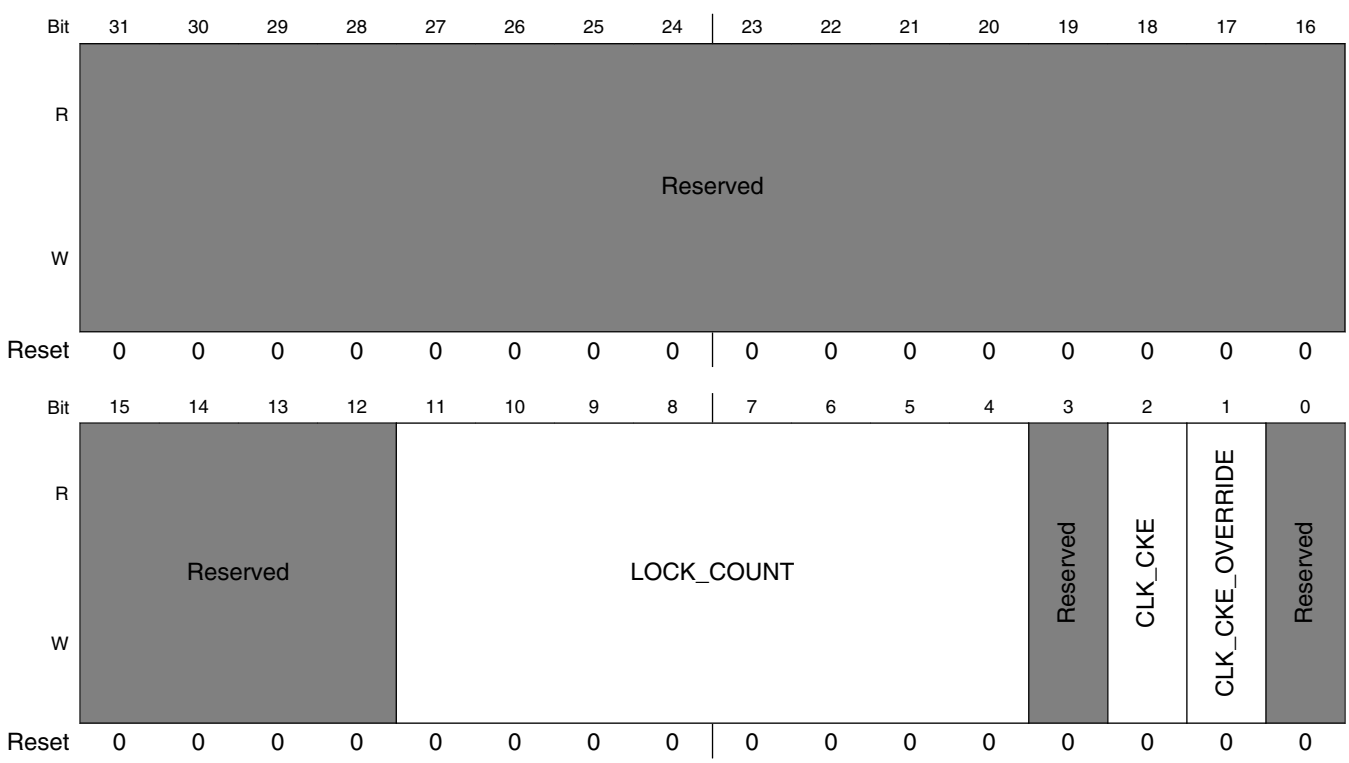
XTALOSC_SYS_OSCNML_CTL0 field descriptions

Field	Description
31 EN	Enable Oscillator
30–5 -	This field is reserved. Reserved
4 RTO	Retention Enable
3 -	This field is reserved. Reserved
2 SP	Select Power
1 SF1	Select Frequency1
0 SF0	Select Frequency0

5.3.3.2 OSC Normal Clock Generation Control Register1 (XTALOSC_SYS_OSCNML_CTL1)

OSC Normal Clock Generation Control Register1

Address: 3027_0000h base + 4h offset = 3027_0004h



XTALOSC_SYS_OSCNML_CTL1 field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11–4 LOCK_COUNT	Lock Signal Gen Counter
3 -	This field is reserved. Reserved
2 CLK_CKE	Oscillator Clock Gating Enable
1 CLK_CKE_ OVERRIDE	Oscillator Clock Gating Enable Override
0 -	This field is reserved. Reserved

5.4 Thermal Monitoring Unit (TMU)

5.4.1 Overview

The Thermal Monitoring Unit (TMU) monitors and reports the temperature from one or more remote temperature measurement sites located on chip.

5.4.1.1 Features

The temperature management unit features:

- Temperature measurement range -40-105°C.
- Monitoring
  - Single-, or multi-site monitoring
  - Out-of-range indication
  - High/low temperature range monitoring
  - Immediate and average temperature monitoring
  - Average temperature monitoring programmable low-pass filtering
  - Programmable monitoring thresholds for normal and critical
- Reporting
  - Immediate and average temperature reporting

5.4.2 Functional description

The TMU has access to two temperature measurement sites located on the chip. The main probe is located inside of the ANAMIX, while the remote probe is located near the ARM core. The TMU monitors these sites and can signal an alarm if a programmed threshold is ever exceeded. The upper and lower temperature range is continuously captured. A set of reporting registers allow for reading the current temperature at monitored sites.

5.4.3 Application Information

5.4.3.1 Re-enabling the TMU with different threshold setting

If the TMU was enabled with a specified threshold setting, the sequence to re-enable the TMU with a different threshold setting is provided below.

In the following steps, the TMU is being re-enabled with a new High Temperature Average threshold value:

1. Disable the TMU by setting TMU_TER[EN] = 0
2. Disable the threshold if it was set before, by setting TMU_TMHTATR[ENx] = 0
3. Set the new threshold value by programming TMU_TMHTATR[TEMPx]
4. Enable the TMU by setting TMU_TER[EN] = 1
5. A delay of at least 5us is needed here to reset the TMU internal states of the last run. Otherwise, the old values might still be used, leading to unexpected results.
6. Enable the threshold by setting TMU_TMHTATR[ENx] = 1

The above sequence also applies for setting new High Temperature Immediate and High Temperature Average Critical threshold values. Please program the appropriate register (TMU_TMHTITR or TMU_TMHTACTR) in steps 2 and 3.

5.4.4 TMU Memory Map/Register Definition

TMU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3026_0000	TMU Enable register (TMU_TER)	32	R/W	4000_0001h	5.4.4.1/731
3026_0004	TMU Probe Select register (TMU_TPS)	32	R/W	C000_0000h	5.4.4.2/732

Table continues on the next page...

TMU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3026_0008	TMU Interrupt Enable register (TMU_TIER)	32	R/W	0000_0000h	5.4.4.3/733
3026_000C	TMU Interrupt Detect register (TMU_TIDR)	32	R/W	0000_0000h	5.4.4.4/734
3026_0010	TMU Monitor High Temperature Immediate Threshold register (TMU_TMHTITR)	32	R/W	0000_0000h	5.4.4.5/735
3026_0014	TMU Monitor High Temperature Average threshold register (TMU_TMHTATR)	32	R/W	0000_0000h	5.4.4.6/736
3026_0018	TMU Monitor High Temperature Average Critical Threshold register (TMU_TMHTACTR)	32	R/W	0000_0000h	5.4.4.7/737
3026_001C	TMU Sensor Calibration register (TMU_TSCR)	32	R	0000_0000h	5.4.4.8/738
3026_0020	TMU Report Immediate Temperature Site register n (TMU_TRITSR)	32	R	0000_0000h	5.4.4.9/739
3026_0024	TMU Report Average Temperature Site register n (TMU_TRATSR)	32	R	0000_0000h	5.4.4.10/740
3026_0028	TMU AS Register (TMU_TASR)	32	R/W	0000_0000h	5.4.4.11/741
3026_002C	TMU TMC Register (TMU_TTMC)	32	R/W	0000_0000h	5.4.4.12/742
3026_0030	TMU CALIV0 Register (TMU_TCALIV0)	32	R/W	0000_0000h	5.4.4.13/742
3026_0034	TMU CALIV1 Register (TMU_TCALIV1)	32	R/W	0000_0000h	5.4.4.14/743
3026_0038	TMU CALIV_M40 Register (TMU_TCALIV_M40)	32	R/W	0000_0000h	5.4.4.15/744
3026_003C	TMU TRIM register (TMU_TRIM)	32	R/W	0000_0080h	5.4.4.16/745

5.4.4.1 TMU Enable register (TMU_TER)

Address: 3026_0000h base + 0h offset = 3026_0000h

TMU_TER field descriptions

Field	Description
31 EN	Enable the temperature sensor 0 Disable 1 Enable
30 ADC_PD	ADC power down controll bit This bit needs to be 0 to enable TMU 0 normal operating mode 1 power down mode
29–2 RESERVED	This field is reserved.
ALPF	Average low pass filter setting. The average temperature is calculated as: ALPF x Current_Temp + (1 - ALPF) x Average_Temp. If no previous (average) temperature is valid, current temperature is used. For proper operation, this field should only change when monitoring is disabled. 00 1.0 01 0.5 10 0.25 11 0.125

5.4.4.2 TMU Probe Select register (TMU_TPS)

Address: 3026_0000h base + 4h offset = 3026_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PROBE_SEL		RESERVED													
W																
Reset	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

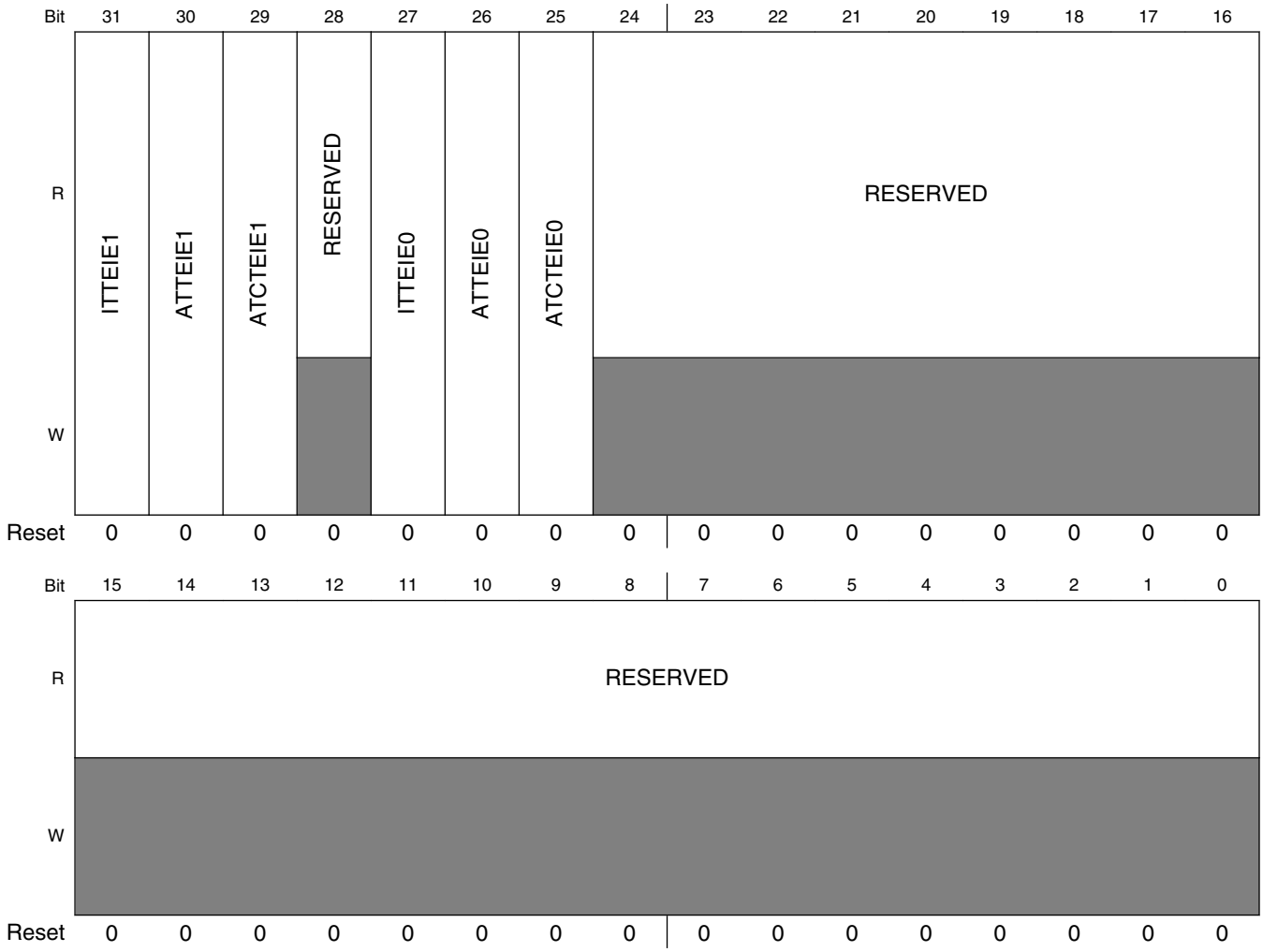
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RESERVED															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMU_TPS field descriptions

Field	Description
31–30 PROBE_SEL	Probe selection bit This field should only change when TER.EN=1'b0. 00 select the main probe only 01 select the remote probe(near A53) only 1x select both 2 probes
RESERVED	This field is reserved.

5.4.4.3 TMU Interrupt Enable register (TMU_TIER)

Address: 3026_0000h base + 8h offset = 3026_0008h



TMU_TIER field descriptions

Field	Description
31 ITTEIE1	Immediate temperature threshold exceeded interrupt enable of probe1. 0 Disabled. 1 Interrupt enabled. Generate an interrupt if TIDR[ITTE] is set.
30 ATTEIE1	Average temperature threshold exceeded interrupt enable of probe1. 0 Disabled. 1 Interrupt enabled. Generate an interrupt if TIDR[ATTE] is set.
29 ATCTEIE1	Average temperature critical threshold exceeded interrupt enable of probe1. 0 Disabled. 1 Interrupt enabled. Generate an interrupt if TIDR[ATCTE] is set.

Table continues on the next page...

TMU_TIER field descriptions (continued)

Field	Description
28 RESERVED	This field is reserved.
27 ITTEIE0	Immediate temperature threshold exceeded interrupt enable of probe0. 0 Disabled. 1 Interrupt enabled. Generate an interrupt if TIDR[ITTE] is set.
26 ATTEIE0	Average temperature threshold exceeded interrupt enable of probe0. 0 Disabled. 1 Interrupt enabled. Generate an interrupt if TIDR[ATTE] is set.
25 ATCTEIE0	Average temperature critical threshold exceeded interrupt enable of probe0. 0 Disabled. 1 Interrupt enabled. Generate an interrupt if TIDR[ATCTE] is set.
RESERVED	This field is reserved.

5.4.4.4 TMU Interrupt Detect register (TMU_TIDR)

Address: 3026_0000h base + Ch offset = 3026_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ITTE1	ATTE1	ATCTE1	RESERVED	ITTE0	ATTE0	ATCTE0	RESERVED								
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RESERVED															
W	w1c															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMU_TIDR field descriptions

Field	Description
31 ITTE1	Immediate temperature threshold exceeded of probe1. 0 No threshold exceeded. 1 Immediate temperature threshold, as defined by TMHTITR, has been exceeded. This includes an out-of-range measured temperature above 125 degree C.
30 ATTE1	Average temperature threshold exceeded of probe1. 0 No threshold exceeded. 1 Average temperature threshold, as defined by TMHTATR, has been exceeded.
29 ATCTE1	Average temperature critical threshold exceeded of probe1. 0 No threshold exceeded. 1 Average temperature critical threshold, as defined by TMHTACTR, has been exceeded.
28 RESERVED	This field is reserved.
27 ITTE0	Immediate temperature threshold exceeded of probe0. 0 No threshold exceeded. 1 Immediate temperature threshold, as defined by TMHTITR, has been exceeded. This includes an out-of-range measured temperature above 125 degree C.
26 ATTE0	Average temperature threshold exceeded of probe0. 0 No threshold exceeded. 1 Average temperature threshold, as defined by TMHTATR, has been exceeded.
25 ATCTE0	Average temperature critical threshold exceeded of probe0. 0 No threshold exceeded. 1 Average temperature critical threshold, as defined by TMHTACTR, has been exceeded.
RESERVED	This field is reserved.

5.4.4.5 TMU Monitor High Temperature Immediate Threshold register (TMU_TMHTITR)

Address: 3026_0000h base + 10h offset = 3026_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EN1	EN0	RESERVED						TEMP1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RESERVED								TEMP0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMU_TMHTITR field descriptions

Field	Description
31 EN1	Enable threshold for probe1. 0 Disabled. 1 Threshold enabled.
30 EN0	Enable threshold for probe0. 0 Disabled. 1 Threshold enabled.
29–24 RESERVED	This field is reserved.
23–16 TEMP1	High temperature immediate threshold value. Determines the current upper temperature threshold when EN=1. -40~125 degree C Sensor range. bit23 is sign bit. Other values are Reserved
15–8 RESERVED	This field is reserved.
TEMP0	High temperature immediate threshold value. Determines the current upper temperature threshold when EN=1. -40~125 degree C Sensor range. bit7 is sign bit. Other values are Reserved

5.4.4.6 TMU Monitor High Temperature Average threshold register (TMU_TMHTATR)

Address: 3026_0000h base + 14h offset = 3026_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EN1	EN0	RESERVED						TEMP1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RESERVED								TEMP0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMU_TMHTATR field descriptions

Field	Description
31 EN1	Enable threshold for probe1.

Table continues on the next page...

TMU_TMHTATR field descriptions (continued)

Field	Description
	0 Disabled. 1 Threshold enabled.
30 EN0	Enable threshold for probe0. 0 Disabled. 1 Threshold enabled.
29–24 RESERVED	This field is reserved.
23–16 TEMP1	High temperature average threshold value. Determines the average upper temperature threshold when EN=1. -40-125 degree C Sensor range. bit23 is sign bit. Other values are Reserved
15–8 RESERVED	This field is reserved.
TEMP0	High temperature average threshold value. Determines the average upper temperature threshold when EN=1. -40-125 degree C Sensor range. bit7 is sign bit. Other values are Reserved

5.4.4.7 TMU Monitor High Temperature Average Critical Threshold register (TMU_TMHTACTR)

Address: 3026_0000h base + 18h offset = 3026_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EN1	EN0	RESERVED						TEMP1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RESERVED								TEMP0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMU_TMHTACTR field descriptions

Field	Description
31 EN1	Enable threshold for probe1. 0 Disabled. 1 Threshold enabled.

Table continues on the next page...

TMU_TMHTACTR field descriptions (continued)

Field	Description
30 EN0	Enable threshold for probe0. 0 Disabled. 1 Threshold enabled.
29–24 RESERVED	This field is reserved.
23–16 TEMP1	High temperature average critical threshold value for probe1. Determines the average upper critical temperature threshold when EN=1. -40-125 degree C Sensor range. bit23 is sign bit. other values reserved
15–8 RESERVED	This field is reserved.
TEMP0	High temperature average critical threshold value for probe0. Determines the average upper critical temperature threshold when EN=1. -40-125 degree C Sensor range. bit7 is sign bit. other values reserved

5.4.4.8 TMU Sensor Calibration register (TMU_TSCR)

Address: 3026_0000h base + 1Ch offset = 3026_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	V1	V0	RESERVED		SNSR1											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RESERVED				SNSR0											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMU_TSCR field descriptions

Field	Description
31 V1	Measured temperature ready of probe1 0 Not ready. First measurement still pending. 1 Ready. Extra 1us delay is needed to read the first [SNSR1] value after this bit is set.
30 V0	Measured temperature ready of probe0

Table continues on the next page...

TMU_TSCR field descriptions (continued)

Field	Description
	0 Not ready. First measurement still pending. 1 Ready. Extra 1us delay is needed to read the first [SNSR0] value after this bit is set.
29–28 RESERVED	This field is reserved.
27–16 SNSR1	Raw sensor value of probe1
15–12 RESERVED	This field is reserved.
SNSR0	Raw sensor value of probe0

5.4.4.9 TMU Report Immediate Temperature Site register n (TMU_TRITSR)

Address: 3026_0000h base + 20h offset = 3026_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	V1	V0	RESERVED						TEMP1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RESERVED								TEMP0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMU_TRITSR field descriptions

Field	Description
31 V1	Measured temperature ready of probe1 0 Not ready. First measurement still pending. 1 Ready. Extra 1us delay is needed to read the first [TEMP1] value after this bit is set.
30 V0	Measured temperature ready of probe0 0 Not ready. First measurement still pending. 1 Ready. Extra 1us delay is needed to read the first [TEMP0] value after this bit is set.
29–24 RESERVED	This field is reserved.
23–16 TEMP1	Last calibratied temperature of probe1 reading at site when V=1.TRITSR[23] is sign bit, 1 means negative, 0 means positive.
15–8 RESERVED	This field is reserved.
TEMP0	Last calibratied temperature of probe0 reading at site when V=1.TRITSR[7] is sign bit, 1 means negative, 0 means positive.

5.4.4.10 TMU Report Average Temperature Site register n (TMU_TRATSR)

Address: 3026_0000h base + 24h offset = 3026_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	V1	V0	RESERVED						TEMP1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

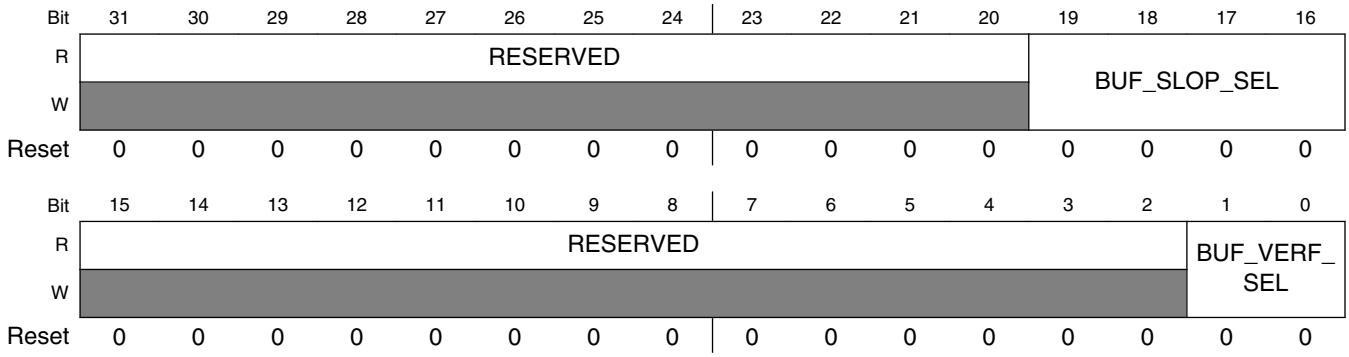
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RESERVED								TEMP0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMU_TRATSR field descriptions

Field	Description
31 V1	Measured temperature ready of probe1 0 Not ready. First measurement still pending. 1 Ready. Extra 1us delay is needed to read the first [TEMP1] value after this bit is set.
30 V0	Measured temperature ready of probe0 0 Not ready. First measurement still pending. 1 Ready. Extra 1us delay is needed to read the first [TEMP0] value after this bit is set.
29–24 RESERVED	This field is reserved.
23–16 TEMP1	Average temperature of probe1 reading at site when V=1. TRATSR[23] is sign bit, 1 means negative, 0 means positive.
15–8 RESERVED	This field is reserved.
TEMP0	Average temperature of probe0 reading at site when V=1. TRATSR[7] is sign bit, 1 means nagetive, 0 means positive.

5.4.4.11 TMU AS Register (TMU_TASR)

Address: 3026_0000h base + 28h offset = 3026_0028h

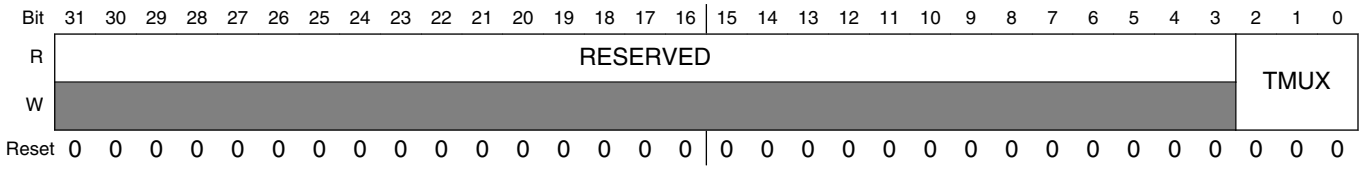


TMU_TASR field descriptions

Field	Description
31–20 RESERVED	This field is reserved.
19–16 BUF_SLOP_SEL	Amplifier gain setting bits. 4'b0000 6.34 mV 4'b0001 6.485 mV 4'b0010 6.63 mV 4'b0011 6.775 mV 4'b0100 6.92 mV 4'b0101 7.065 mV 4'b0110 7.21 mV 4'b0111 7.355 mV 4'b1000 7.5 mV 4'b1001 7.645 mV 4'b1010 7.79 mV 4'b1011 7.935 mV 4'b1100 8.08 mV 4'b1101 8.225 mV 4'b1110 8.37 mV 4'b1111 8.515 mV
15–2 RESERVED	This field is reserved.
BUF_VERF_SEL	Reference voltage setting bits for the amplifier in the Positive-TC generator block.

5.4.4.12 TMU TMC Register (TMU_TTMC)

Address: 3026_0000h base + 2Ch offset = 3026_002Ch

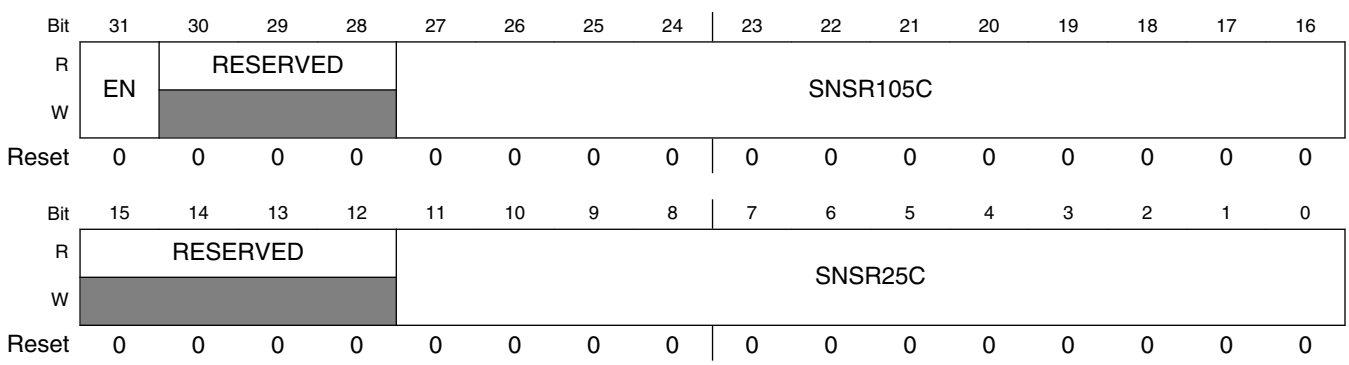


TMU_TTMC field descriptions

Field	Description
31–3 RESERVED	This field is reserved.
TMUX	Test-mux address setting bits. Only used for debugging purpose. 3'b000 ADC Function Test 3'b001 PTAT Voltage Monitoring 3'b010 Reference-top Voltage Monitoring 3'b011 Reference Voltage (for Amp.) Monitoring 3'b100 N/A 3'b101 VBE Voltage Monitoring 3'b110 N/A (Default) 3'b111 N/A

5.4.4.13 TMU CALIV0 Register (TMU_TCALIV0)

Address: 3026_0000h base + 30h offset = 3026_0030h



TMU_TCALIV0 field descriptions

Field	Description
31 EN	Enable 105C calibration value used in 1p calibration. If enable, 105C sensor value take the priority. Used for probe0

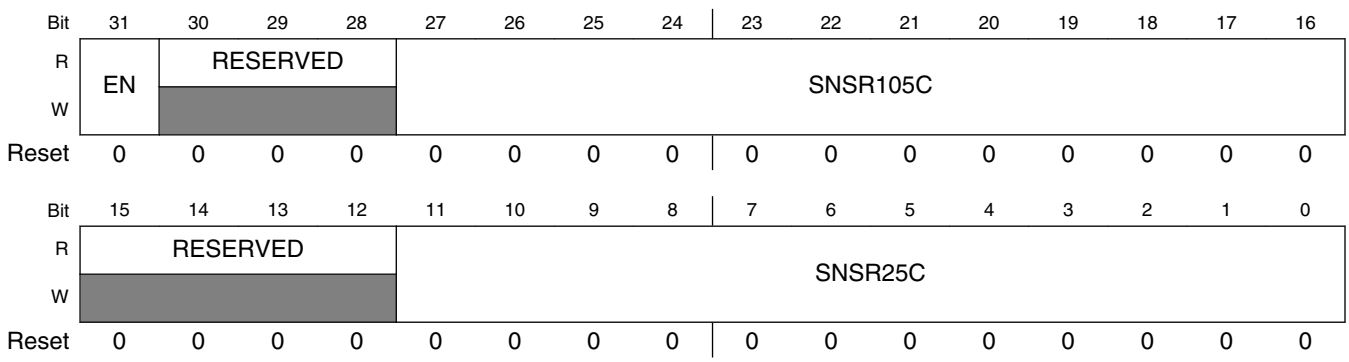
Table continues on the next page...

TMU_TCALIV0 field descriptions (continued)

Field	Description
	0 Disable. 1 Enable.
30–28 RESERVED	This field is reserved.
27–16 SNSR105C	105C sensor value of probe0 read from FUSE, system should program it before enable TMU if enable 1P calibration.
15–12 RESERVED	This field is reserved.
SNSR25C	25C sensor value of probe0 read from FUSE(default 1p value), system should program it before enable TMU if enable 1P calibration.

5.4.4.14 TMU CALIV1 Register (TMU_TCALIV1)

Address: 3026_0000h base + 34h offset = 3026_0034h



TMU_TCALIV1 field descriptions

Field	Description
31 EN	Enable 105C calibration value used in 1p calibration. If enable, 105C sensor value take the priority. Used for probe1 0 Disable. 1 Enable.
30–28 RESERVED	This field is reserved.
27–16 SNSR105C	105C sensor value of probe1 read from FUSE, system should program it before enable TMU if enable 1P calibration.
15–12 RESERVED	This field is reserved.
SNSR25C	25C sensor value of probe1 read from FUSE(default 1p value), system should program it before enable TMU if enable 1P calibration.

5.4.4.15 TMU CALIV_M40 Register (TMU_TCALIV_M40)

Address: 3026_0000h base + 38h offset = 3026_0038h

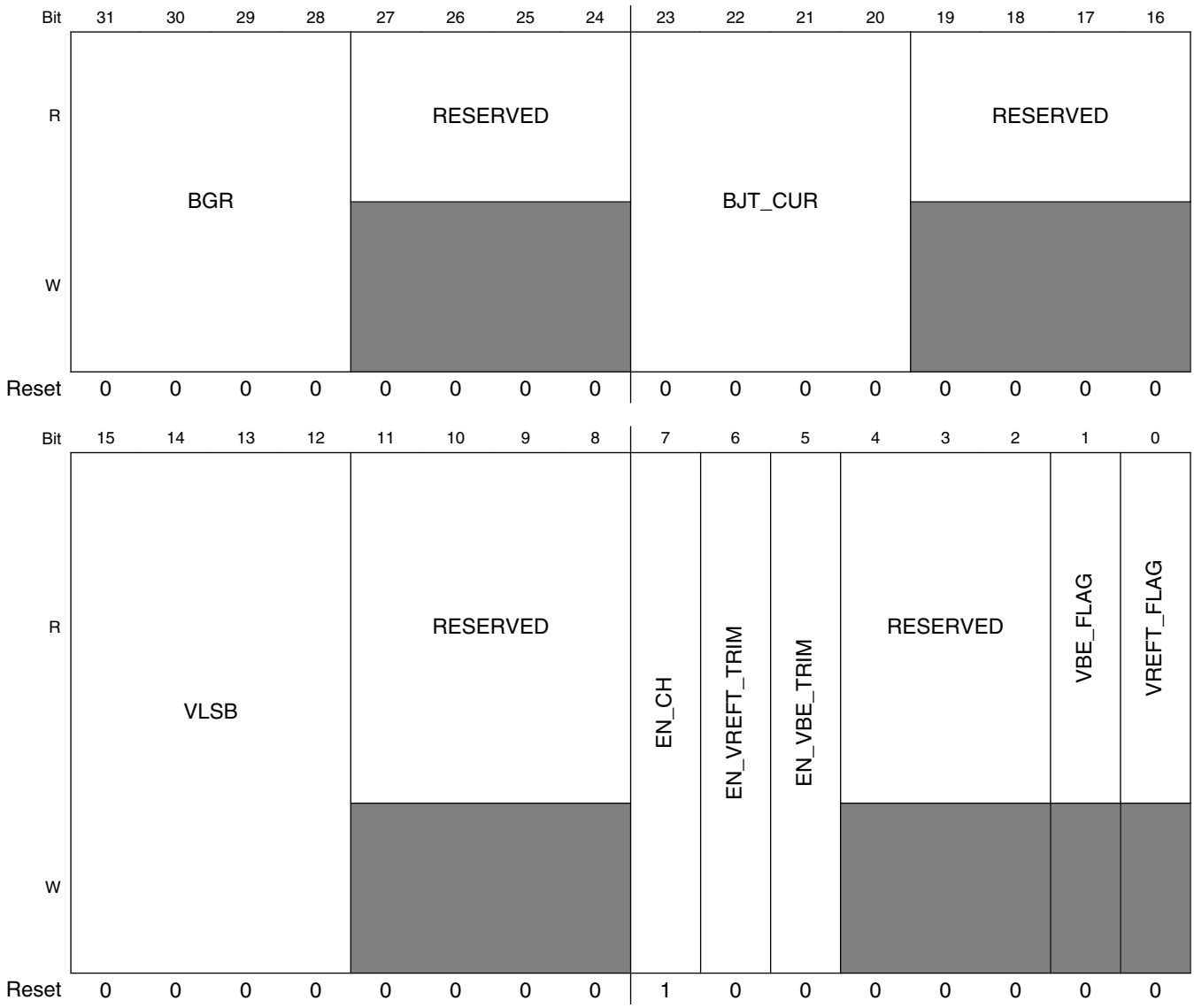
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RESERVED				SNSR_M40C_1												RESERVED				SNSR_M40C_0											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMU_TCALIV_M40 field descriptions

Field	Description
31–28 RESERVED	This field is reserved.
27–16 SNSR_M40C_1	m40C sensor value of probe1 read from FUSE, used for software 3p calibration if needed
15–12 RESERVED	This field is reserved.
SNSR_M40C_0	m40C sensor value of probe0 read from FUSE, used for software 3p calibration if needed

5.4.4.16 TMU TRIM register (TMU_TRIM)

Address: 3026_0000h base + 3Ch offset = 3026_003Ch



TMU_TRIM field descriptions

Field	Description
31–28 BGR	Curvature trimming bits for voltage change in BGR
27–24 RESERVED	This field is reserved.
23–20 BJT_CUR	Current trimming ports for BJT
19–16 RESERVED	This field is reserved.

Table continues on the next page...

TMU_TRIM field descriptions (continued)

Field	Description
15–12 VLSB	LSB voltage of DAC trimming
11–8 RESERVED	This field is reserved.
7 EN_CH	Offset calibration enable
6 EN_VREFT_ TRIM	Verf calibration enable
5 EN_VBE_TRIM	Vbe calibration enable
4–2 RESERVED	This field is reserved.
1 VBE_FLAG	Flag of the Vbe trim
0 VREFT_FLAG	Flag of the Vref trim

Chapter 6
SNVS, Reset, Fuse, and Boot

6.1 System Boot

6.1.1 Overview

The boot process begins at the Power-On Reset (POR) where the hardware reset logic forces the Arm core to begin the execution starting from the on-chip boot ROM.

The boot ROM code uses the state of the internal register BOOT_MODE[3:0] as well as the state of various eFUSEs settings to determine the boot flow behavior of the device.

The main features of the ROM include:

- Support for booting from various boot devices
- Serial downloader support (USB)
- High Assurance Boot

The boot ROM supports these boot devices:

- Serial NOR Flash via FlexSPI
- Serial NAND Flash via FlexSPI
- NAND flash
- SD/eMMC
- Serial (SPI) NOR

The boot ROM code also allows to download the programs to be run on the device. The example is a provisioning program that can make further use of the serial connection to provide a boot device with a new image. Typically, the provisioning program is downloaded to the internal RAM and allows to program the boot devices, such as the SD/MMC flash. The ROM serial downloader uses a high-speed USB in a non-stream mode connection.

A key feature of the boot ROM is the ability to perform a secure boot, also known as a High-Assurance Boot (HAB). This is supported by the HAB security library which is a subcomponent of the ROM code. The HAB uses a combination of hardware and software together with the Public Key Infrastructure (PKI) protocol to protect the system from executing unauthorized programs. Before the HAB allows the user image to execute, the image must be signed. The signing process is done during the image build process by the private key holder and the signatures are then included as a part of the final program image. If configured to do so, the ROM verifies the signatures using the public keys included in the program image. A secure boot with HAB can be performed on all boot devices supported on the chip in addition to the serial downloader. The HAB library in the boot ROM also provides the API functions, allowing the additional boot chain components (bootloaders) to extend the secure boot chain. The out-of-fab setting for the SEC_CONFIG is the open configuration, in which the ROM/HAB performs the image authentication, but all authentication errors are ignored and the image is still allowed to execute.

6.1.2 Boot modes

During boot, the core's behavior is defined by the boot mode pin settings, as described in Boot mode pin settings.

6.1.2.1 Boot mode pin settings

The device supports 16 boot modes but only several are supported on the chip and the others are reserved for future use. The boot mode is selected based on the binary value stored in the internal BOOT_MODE[3:0] register.

The BOOT_MODE[3:0] is initialized by sampling the BOOT_MODE[3:0] inputs on the rising edge of the POR_B. After these inputs are sampled, their subsequent state does not affect the contents of the BOOT_MODE[3:0] internal register. The state of the internal BOOT_MODE[3:0] register may be read from the IPP_BOOT_MODE[3:0] field of the SRC Boot Mode Register (SRC_SBMR2). The available boot modes are: Boot From Fuses, serial boot via USB, and Internal Boot. See this table for settings:

Table 6-1. Boot MODE pin settings

Boot Device Select	BOOT_MODE [3]	BOOT_MODE [2]	BOOT_MODE [1]	BOOT_MODE [0]
Boot from internal fuses	0	0	0	0
USB Serial Download	0	0	0	1
USDHC3 (eMMC)	0	0	1	0

Table continues on the next page...

Table 6-1. Boot MODE pin settings (continued)

Boot Device Select	BOOT_MODE [3]	BOOT_MODE [2]	BOOT_MODE [1]	BOOT_MODE [0]
USDHC2 (SD)	0	0	1	1
NAND 8-bit single device, 256 pages	0	1	0	0
NAND 8-bit single device, 512 pages	0	1	0	1
FlexSPI 3B Read	0	1	1	0
FlexSPI Hyperflash 3.3V	0	1	1	1
eCSPI Boot	1	0	0	0
Reserved	1	0	0	1
FLEXSPI Serial NAND - 2k page	1	0	1	0
FLEXSPI Serial NAND - 4k page	1	0	1	1

6.1.2.2 High-level boot sequence

The figure found here show the high-level boot ROM code flow.

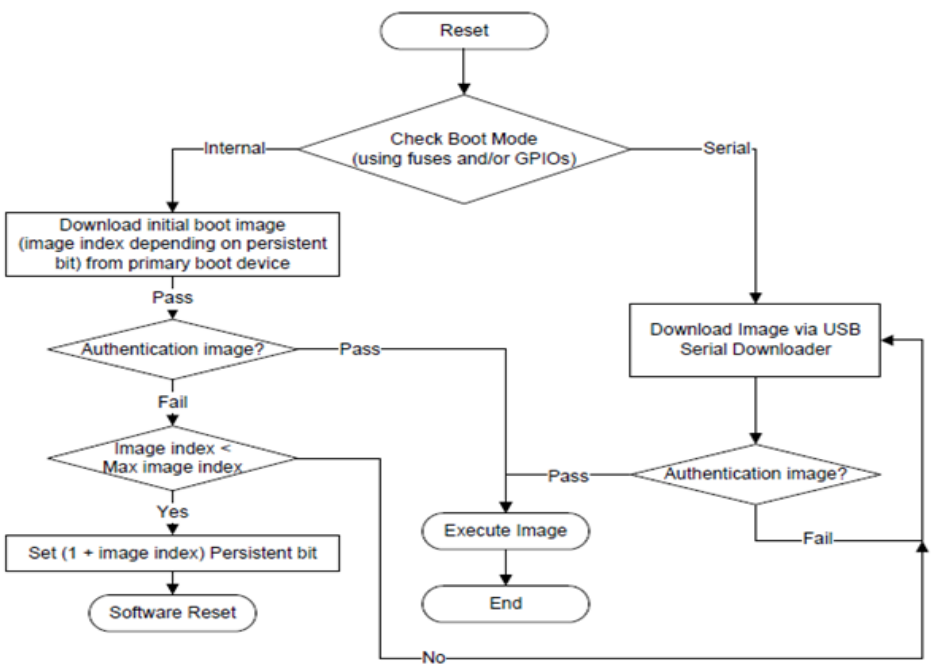


Figure 6-1. Boot flow

6.1.2.3 Boot From Fuses mode (BOOT_MODE[3:0] = 0000b)

A value of 0000b in the BOOT_MODE[3:0] register selects the Boot From Fuses mode.

This mode is similar to the Internal Boot mode described in Internal Boot mode with one difference. In this mode, the ROM code retrieves boot mode settings from the boot mode fuses.The boot ROM code uses the boot eFUSE settings only. This mode also supports a secure boot using HAB.

If set to Boot From Fuses or FORCE_BT_FROM_FUSE=1, the boot flow is controlled by the BT_FUSE_SEL eFUSE value. If BT_FUSE_SEL = 0, indicating that the boot device (for example, flash, SD/MMC) was not programmed yet, the boot flow jumps directly to the Serial Downloader. If BT_FUSE_SEL = 1, the normal boot flow is followed, where the ROM attempts to boot from the selected boot device.

The first time a board is used, the default eFUSEs may be configured incorrectly for the hardware on the platform. In such case, the Boot ROM code may try to boot from a device that does not exist. This may cause an electrical/logic violation on some pads. Using the Boot From Fuses mode addresses this problem.

Setting the BT_FUSE_SEL=0 forces the ROM code to jump directly to the Serial Downloader. This allows a bootloader to be downloaded which can then provision the boot device with a program image and blow the BT_FUSE_SEL and the other boot configuration eFUSEs. After the reset, the boot ROM code determines that the BT_FUSE_SEL is blown (BT_FUSE_SEL = 1) and the ROM code performs an internal boot according to the new eFUSE settings. This allows the user to set BOOT_MODE[3:0]=0000b on a production device and burn the fuses on the same device (by forcing the entry to the Serial Downloader), without changing the value of the BOOT_MODE[3:0] or the pullups/pulldowns on the BOOT_MODE[3:0] pins.

6.1.2.4 Internal Boot mode

Any of the values from Table 6-8 in the BOOT_MODE[3:0] register selects the Internal Boot mode. In this mode, the processor continues to execute the boot code from the internal boot ROM.

The boot code performs the hardware initialization, loads the program image from the chosen boot device, performs the image validation using the HAB library (see Boot security settings), and then jumps to an address derived from the program image. If an error occurs during the internal boot, the boot code jumps to the Serial Downloader (see Serial Downloader).

When set to the Internal Boot, the boot flow may be controlled by a combination of eFUSE settings and Boot Mode pins. The fuse (BT_FUSE_SEL) will not impact the behavior of ROM in Internal Boot Mode.

6.1.2.5 Boot security settings

The internal boot modes use one of three security configurations.

- Closed: This level is intended for use with shipping-secure products. All HAB functions are executed and the security hardware is initialized (the Security Controller or SNVS enters the Secure state), the DCD is processed if present, and the program image is authenticated by the HAB before its execution. All detected errors are logged, and the boot flow is aborted with the control being passed to the serial downloader. At this level, the execution does not leave the internal ROM unless the target executable image is authenticated.
- Open: This level is intended for use in non-secure products or during the development phases of a secure product. All HAB functions are executed as for a closed device. The security hardware is initialized (except for the SNVS which is left in the Non-Secure state), the DCD is processed if present, and the program image is authenticated by the HAB before its execution. All detected errors are logged, but have no influence on the boot flow which continues as if the errors did not occur. This configuration is useful for a secure product development because the program image runs even if the authentication data is missing or incorrect, and the error log can be examined to determine the cause of the authentication failure.
- Field Return: This level is intended for the parts returned from the shipped products.

NOTE

Refer to the chip's Security Reference Manual for more details on security configuration.

6.1.3 Device configuration

This section describes the external inputs that control the behavior of the Boot ROM code.

This includes the boot device selection (SD, MMC, and so on), boot device configuration (SD bus width, speed, and so on), and other. In general, the source for this configuration should come from the eFUSEs embedded inside the chip. However, certain configuration parameters can be sourced from the BOOT_MODE[3:0] pins (with Internal Boot mode), allowing further flexibility during the development process.

6.1.3.1 Boot eFUSE descriptions

This table is a comprehensive list of the configuration parameters that the ROM uses.

Table 6-2. Boot eFUSE descriptions

Fuse Address	Configuration	Definition	Shipped value	Settings[1]
BT_FUSE_SEL 0x470[28]	OEM	In the Boot From Fuse mode (BOOT_MODE[3:0] = 0000b), the BT_FUSE_SEL fuse indicates whether the bit configuration eFuses are programmed.	0	If Boot From Fuse mode (BOOT_MODE[3:0] = 0000b) or FORCE_BT_FROM_FUSE=1 : 0—The BOOT configuration eFuses are not programmed yet. The boot flow jumps to the serial downloader. 1—The BOOT configuration eFuses are programmed. The regular boot flow is performed.
SDP_DISABLE 0x480[21]	OEM	Serial download disable bit	0	0 - Enable 1 - Disable
L1 I-Cache DISABLE 0x480[12]	OEM	L1 I Cache disable bit used by the boot during the entire execution.	0	0—L1 I Cache is enabled by the ROM during the boot. 1—L1 I Cache is disabled by the ROM during the boot.
BT_FREQ 0x480[9]	OEM	Boot frequency selection	0	0—Arm—1000 MHz 1—Arm—500 MHz
LPB_BOOT 0x480[15:14]	OEM	USB Low-Power Boot	00	0x—LPB Disable 10—Divide by 2 11—Divide by 4
BT_LPB_POLARITY 0x480[13]	OEM	USB Low-Power Boot GPIO polarity	0	0—Low on the GPIO pad indicates the lowpower condition. 1—High on the GPIO pad indicates the low-power condition.
WDOG_ENABLE 0x480[10]	OEM	Watchdog reset counter enable	0	0—The watchdog reset counter is disabled during the serial downloader.

Table continues on the next page...

Table 6-2. Boot eFUSE descriptions (continued)

				1—The watchdog reset counter is enabled during the serial downloader.
MMC_DLL_DLY[6:0] 0x490[14:8]	OEM	uSDHC Delay Line settings	0000000	uSDHC Delay Line settings
USDHC_PAD_SETTINGS 0x490[31:24] NAND_PAD_SETTINGS 0x4A0[31:24]	OEM	Override values for the SD/MMC and NAND boot modes	00000000	Override the following IO PAD settings: [1:0] Driver Strength [2] Slew Rate [3] Hysteresis [4] Pull/Keeper select [6:5] Pull up/down config [7] Reserved.

1. 0 = intact fuse and 1= blown fuse

6.1.4 Device initialization

This section describes the details of the ROM and provides the initialization details.

This includes details on:

- The ROM memory map
- The RAM memory map
- On-chip blocks that the ROM must use or change the POR register default values
- Clock initialization
- Enabling the MMU/L2 cache
- Exception handling and interrupt handling

6.1.4.1 Internal ROM/RAM memory map

These figures show the iROM memory map:

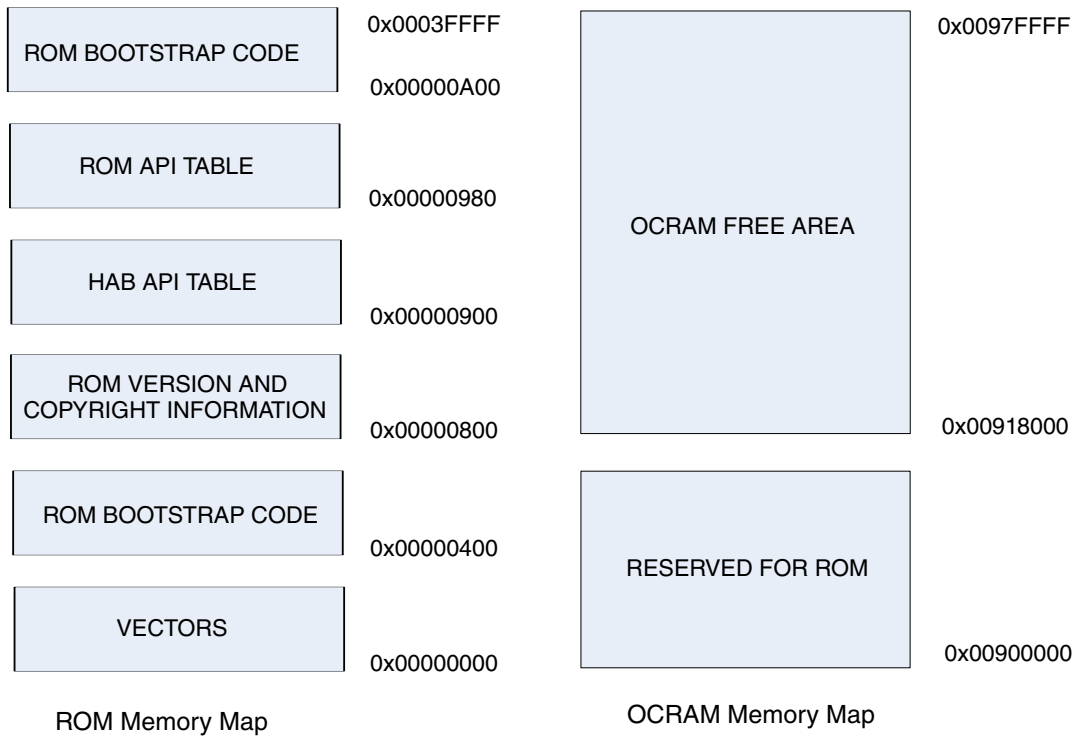


Figure 6-2. Internal ROM and RAM memory map

NOTE

If no ROM/HAB APIs are being used, the entire OCRAM region can be used freely after the boot.

6.1.4.2 Boot block activation

The boot ROM affects a number of different hardware blocks which are activated and play a vital role in the boot flow.

The ROM configures and uses the following blocks (listed in an alphabetical order) during the boot process. Note that the blocks actually used depend on the boot mode and the boot device selection:

- APBH—the DMA engine to drive the GPMI module
- BCH—62-bit error correction hardware engine with the AXI bus master and a private connection to the GPMI
- CCM—Clock Control Module
- ECSPI—Enhanced Configurable Serial Peripheral Interface
- FlexSPI—Flexible SPI Interface which supports serial NOR devices
- GPMI—NAND controller pin interface
- OCOTP_CTRL—On-Chip OTP Controller; the OCOTP contains the eFUSEs

- IOMUXC—I/O Multiplexer Control which allows the GPIO use to override the eFUSE boot settings;
- IOMUXC GPR—I/O Multiplexer Control General-Purpose Registers
- CAAM—Cryptographic Acceleration and Assurance Module
- SNVS—Secure Non-Volatile Storage
- SRC—System Reset Controller
- USB—used for the serial download of a boot device provisioning program
- USDHC—Ultra-Secure Digital Host Controller
- WDOG-1—Watchdog timer

6.1.4.3 Clocks at boot time

The following table shows the PLLs and clock source settings in ROM code.

Table 6-3. PLL setting by ROM

PLL name	Frequency	Comment
ARM_PLL	1000 MHz	-
SYS_PLL1	800 MHz	-
SYS_PLL2	1000 MHz	-

NOTE

All other PLLs are in the default status.

NOTE

Refer to Low-power boot for low-power boot frequencies.

Table 6-4. Clock root setting by ROM

Clock Name	Frequency (MHz)	Source	Enable
ARM_A53_ROOT	1000	ARM_PLL_CLK	Yes
ARM_M7_CLK_ROOT	200	SYSTEM_PLL2_200M_CLK	No
AHB_CLK_ROOT	133	SYSTEM_PLL1_133M_CLK	Yes
MAIN_AXI_CLK_ROOT	400	SYSTEM_PLL1_800M_CLK	Yes
NAND_CLK_ROOT	500	SYSTEM_PLL2_500M_CLK	Enabled by driver
NAND_USDHC_BUS_CLK_ROOT	266	SYSTEM_PLL1_266M_CLK	Enabled by driver
USB_BUS_CLK_ROOT		SYSTEM_PLL2_500M_CLK	Enabled by driver
NOC_CLK_ROOT	400	SYSTEM_PLL1_800M_CLK	Yes
USDHC1_CLK_ROOT	200	SYSTEM_PLL1_400M_CLK	Enabled by driver
USDHC2_CLK_ROOT	200	SYSTEM_PLL1_400M_CLK	Enabled by driver
USDHC3_CLK_ROOT	200	SYSTEM_PLL1_400M_CLK	Enabled by driver

Table continues on the next page...

Table 6-4. Clock root setting by ROM (continued)

Clock Name	Frequency (MHz)	Source	Enable
USB_PHY_REF_CLK_ROOT		SYSTEM_PLL1_100M_CLK	Enabled by driver
ECSPI1_CLK_ROOT	50	SYSTEM_PLL2_200M_CLK	No
ECSPI2_CLK_ROOT	50	SYSTEM_PLL2_200M_CLK	No
ECSPI3_CLK_ROOT	50	SYSTEM_PLL2_200M_CLK	No
WRCLK_CLK_ROOT		SYSTEM_PLL1_40M_CLK	No

NOTE

All other clock roots are in the default status.

Table 6-5. NAND_CLK_ROOT setting

NAND data rate	NAND_CLK_ROOT source	Frequency
Async/Legacy NAND	SYSTEM_PLL1_400M_CLK	25 MHz
Sync 40M	SYSTEM_PLL1_400M_CLK	40 MHz
Toggle/Sync 66M	SYSTEM_PLL1_400M_CLK	66 MHz
Toggle 80M	SYSTEM_PLL1_400M_CLK	80 MHz
Sync 100M	SYSTEM_PLL1_400M_CLK	100 MHz
Toggle/Sync 133M	SYSTEM_PLL1_400M_CLK	133 MHz
Sync 160M	SYSTEM_PLL1_400M_CLK	133 MHz
Toggle/Sync 200M	SYSTEM_PLL1_400M_CLK	200 MHz

NOTE

The NAND_CLK_ROOT source depends on the NAND data rate.

The ROM code disables the clocks listed in the following table, except for the boot devices listed in the "Enabled for boot device" column below.

Table 6-6. CCGR setting by ROM

Gating Register	LPCG Enable	Enabled for boot device
CCM_CCGR0	DVFS	
CCM_CCGR1	Anamix	Yes
CCM_CCGR2	CPU	Yes
CCM_CCGR3	CSU	Yes
CCM_CCGR4	Debug	Yes
CCM_CCGR5	DRAM1	
CCM_CCGR6	Reserved	Yes
CCM_CCGR7	ECSPI1	Yes
CCM_CCGR8	ECSPI2	Yes

Table continues on the next page...

Table 6-6. CCGR setting by ROM (continued)

Gating Register	LPCG Enable	Enabled for boot device
CCM_CCGR9	ECSPI3	Yes
CCM_CCGR10	ENET1	Yes
CCM_CCGR11	GPIO1	Yes
CCM_CCGR12	GPIO2	Yes
CCM_CCGR13	GPIO3	Yes
CCM_CCGR14	GPIO4	Yes
CCM_CCGR15	GPIO5	Yes
CCM_CCGR16	GPT1	Yes
CCM_CCGR17	GPT2	Yes
CCM_CCGR18	GPT3	
CCM_CCGR19	GPT4	
CCM_CCGR20	GPT5	
CCM_CCGR21	GPT6	
CCM_CCGR22	HS	
CCM_CCGR23	I2C1	
CCM_CCGR24	I2C2	
CCM_CCGR25	I2C3	
CCM_CCGR26	I2C4	
CCM_CCGR27	IOMUX	Yes
CCM_CCGR28	IPMUX1	Yes
CCM_CCGR29	IPMUX2	Yes
CCM_CCGR30	IPMUX3	Yes
CCM_CCGR31	Reserved	Yes
CCM_CCGR32	Reserved	Yes
CCM_CCGR33	MU	
CCM_CCGR34	OCOTP	Yes
CCM_CCGR35	OCRAM	Yes
CCM_CCGR36	OCRAM_s	Yes
CCM_CCGR37	PCIE	
CCM_CCGR38	PERFMON1	
CCM_CCGR39	PERFMON2	
CCM_CCGR40	PWM1	
CCM_CCGR41	PWM2	
CCM_CCGR42	PWM3	
CCM_CCGR43	PWM4	
CCM_CCGR44	QoS	Yes
CCM_CCGR45	Reserved	Yes
CCM_CCGR46	QoS_ENET	Yes
CCM_CCGR47	FLEXSPI	Yes

Table continues on the next page...

Table 6-6. CCGR setting by ROM (continued)

Gating Register	LPCG Enable	Enabled for boot device
CCM_CCGR48	RAWNAND	Yes
CCM_CCGR49	RDC	
CCM_CCGR50	ROM	Yes
CCM_CCGR51	I2C5	
CCM_CCGR52	I2C6	
CCM_CCGR53	CAN1	Yes
CCM_CCGR54	CAN2	Yes
CCM_CCGR55	Reserved	Yes
CCM_CCGR56	Reserved	Yes
CCM_CCGR57	SCTR	
CCM_CCGR58	SDMA1	
CCM_CCGR59	ENET_QoS	
CCM_CCGR60	SEC_DEBUG	
CCM_CCGR61	SEMA1	
CCM_CCGR62	SEMA2	
CCM_CCGR63	IRQ_STEER	Yes
CCM_CCGR64	SIM_ENET	Yes
CCM_CCGR65	SIM_m	Yes
CCM_CCGR66	SIM_main	Yes
CCM_CCGR67	SIM_s	Yes
CCM_CCGR68	SIM_wakeup	Yes
CCM_CCGR69	GPU2D	Yes
CCM_CCGR70	GPU3D	Yes
CCM_CCGR71	SNVS	Yes
CCM_CCGR72	TRACE	Yes
CCM_CCGR73	UART1	
CCM_CCGR74	UART2	
CCM_CCGR75	UART3	
CCM_CCGR76	UART4	
CCM_CCGR77	USB	Yes
CCM_CCGR78	Reserved	Yes
CCM_CCGR79	USB_PHY	
CCM_CCGR80	Reserved	Yes
CCM_CCGR81	USDHC1	
CCM_CCGR82	USDHC2	
CCM_CCGR83	WDOG1	Yes
CCM_CCGR84	WDOG2	
CCM_CCGR85	WDOG3	
CCM_CCGR86	VPU_G1	

Table continues on the next page...

Table 6-6. CCGR setting by ROM (continued)

Gating Register	LPCG Enable	Enabled for boot device
CCM_CCGR87	GPU	
CCM_CCGR88	NOC_WRAPPER	Yes
CCM_CCGR89	VPU_ENCODER	
CCM_CCGR90	VPU_G2	
CCM_CCGR91	NPU	Yes
CCM_CCGR92	HSIO	Yes
CCM_CCGR93	MEDIA	Yes
CCM_CCGR94	USDHC3	Yes
CCM_CCGR95	HDMI	Yes
CCM_CCGR96	XTAL	Yes
CCM_CCGR97	PLL	Yes
CCM_CCGR98	TSENSOR	Yes
CCM_CCGR99	VPU	
CCM_CCGR100	MRPR	Yes
CCM_CCGR101	AUDIO	Yes
CCM_CCGR102	Reserved	Yes

6.1.4.4 Enabling MMU and caches

The boot ROM includes a feature that enables the Memory Management Unit (MMU) and the caches to improve the boot speed.

The L1 instruction cache is enabled at the very beginning, unless the ICACHE_DISABLE fuse is blown. The MMU is always enabled and ROM enables it at the very beginning, after enabling the L1 ICACHE. The L1 data cache is enabled during image authentication and will be disabled after the image authentication is completed. The fuse, DCACHE_DIS, is used to control L1 DCACHE enable/disable. By default (fuse not programmed), ROM enables the L1 DCACHE.

6.1.4.5 Exception handling

The exception vectors located at the start of the ROM are used to map all the Arm exceptions (except the reset exception) to a duplicate exception vector table in the internal RAM.

During the boot phase of CPU0, the RAM vectors point to the serial downloader in the ROM.

After the boot, the program image can overwrite the vectors as required. The code shown below is used to map the ROM exception vector table to the duplicate exception vector table in the RAM.

When an exception occurs, ROM does not move into USB serial download mode. Instead, ROM will set a flag (EXCEPTION_OCCURED) in the SRC GPR, and then reset the chip.

Mapping ROM Exception Vector Table

```
rom_vectors
B       startup                     ; offset is 0x000
ALIGN   0x80
B       default_exception_handler   ; offset is 0x080
ALIGN   0x80
B       default_exception_handler   ; offset is 0x100
ALIGN   0x80
B       default_exception_handler   ; offset is 0x180
ALIGN   0x80
B       sync_exception_handler      ; offset is 0x200
ALIGN   0x80
B       default_exception_handler   ; offset is 0x280
ALIGN   0x80
B       default_exception_handler   ; offset is 0x300
ALIGN   0x80
B       default_exception_handler   ; offset is 0x380
```

6.1.4.6 Interrupt handling during boot

No special interrupt-handling routines are required during the boot process. The interrupts are disabled during the boot ROM execution and may be enabled in a later boot stage.

6.1.4.7 Persistent bits

Some modes of the boot ROM require the registers that keep their values after a warm reset. The SRC General-Purpose registers are used for this purpose.

See this table for persistent bits list and description:

Table 6-7. Persistent bits

Bit name	Bit location	Description
-	SRC_GPR2[31:0]	Holds the FlexSPI NOR auto probe persistent content.
PERSIST_EXCEPTION_OCCURED	SRC_GPR10[17]	Exception cause reset

6.1.5 Boot devices (internal boot)

The chip supports these boot flash devices:

- Serial NOR flash via FlexSPI interface
- Serial NAND flash via FlexSPI, with options for 2k page and 4k page
- Raw NAND (MLC and SLC), and Toggle-mode NAND flash through GPMI-2 interface, located at CS0. Page sizes of 2 KB, 4 KB, and 8 KB. The bus widths of 8-bit with 2 through 62-bit BCH hardware ECC (Error Correction) are supported.
- SD/MMC/eSD/SDXC/eMMC via USDHC interface, supporting high capacity cards.

The selection of the external boot device type is controlled by BOOT_MODE[3:0]. See the table below for more details:

Table 6-8. Boot device selection

BOOT_MODE[3:0]	Boot device
0010	eMMC boot - by default, boots from USDHC3 port. Can be overridden by fuses settings.
0011	SD boot - by default, boots from USDHC2 port. Can be overridden by fuses settings.
0100	NAND 8-bit single device - 256 pages in block
0101	NAND 8-bit single device - 512 pages in block
0110	FlexSPI - 3B Read
0111	FlexSPI - Hyperflash 3.3V
1000	eCSPI Boot
1001	Reserved
1010	FLEXSPI Serial NAND - 2k page
1011	FLEXSPI Serial NAND - 4k page

6.1.5.1 Serial NAND Flash Boot over FlexSPI

The boot ROM supports a number of Serial NAND Flash devices from different vendors. The Embedded Error Correction and Control (ECC) module in SPI NAND devices are used to detect and correct the errors.

6.1.5.1.1 NAND Flash Boot Flow and Boot Control Blocks (BCB)

There are two BCB data structures:

- FCB
- DBBT

As part of the NAND media initialization, the ROM driver uses safe NAND timings to search for a Firmware Configuration Block (FCB) that contains the optimum NAND timings, page address of Discovered Bad Block Table (DBBT) Search Area and start page address of primary and secondary firmware.

The hardware ECC level to use is embedded inside FCB block. The FCB data structure is also protected using ECC. Driver reads raw 2112 bytes of first sector and runs through software ECC engine that determines whether FCB data is valid or not.

If the FCB is found, the optimum NAND timings are loaded for further reads. If the ECC fails, or the fingerprints do not match, the Block Search state machine increments page number to Search Stride number of pages to read for the next BCB until SearchCount pages have been read.

If search fails to find a valid FCB, the NAND driver responds with an error and the boot ROM enters into serial download mode.

The FCB contains the page address of DBBT Search Area, and the page address for primary and secondary boot images. DBBT is searched in DBBT Search Area just like how FCB is searched. After the FCB is read, the DBBT is loaded, and the primary or secondary boot image is loaded using starting page address from FCB.

Figure 6-3 shows the state diagram of FCB search.

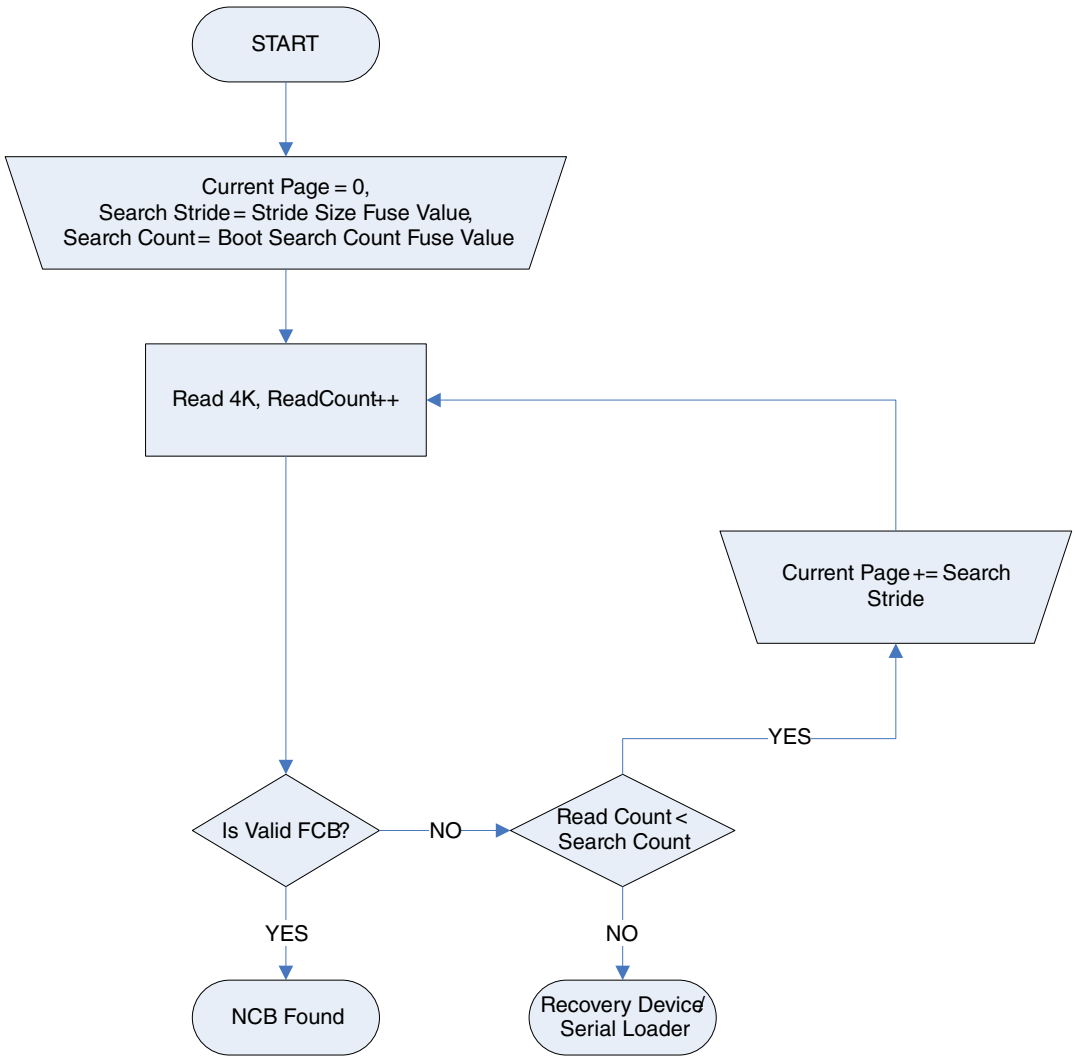


Figure 6-3. FCB Search Flow

Once FCB is found, the boot ROM searches for the Discovered Bad Blocks Table (DBBT). If DBBT Search Area is 0 in FCB, then ROM assumes that there are no bad blocks on NAND device boot area. See Figure 6-4 for the DBBT search flow.

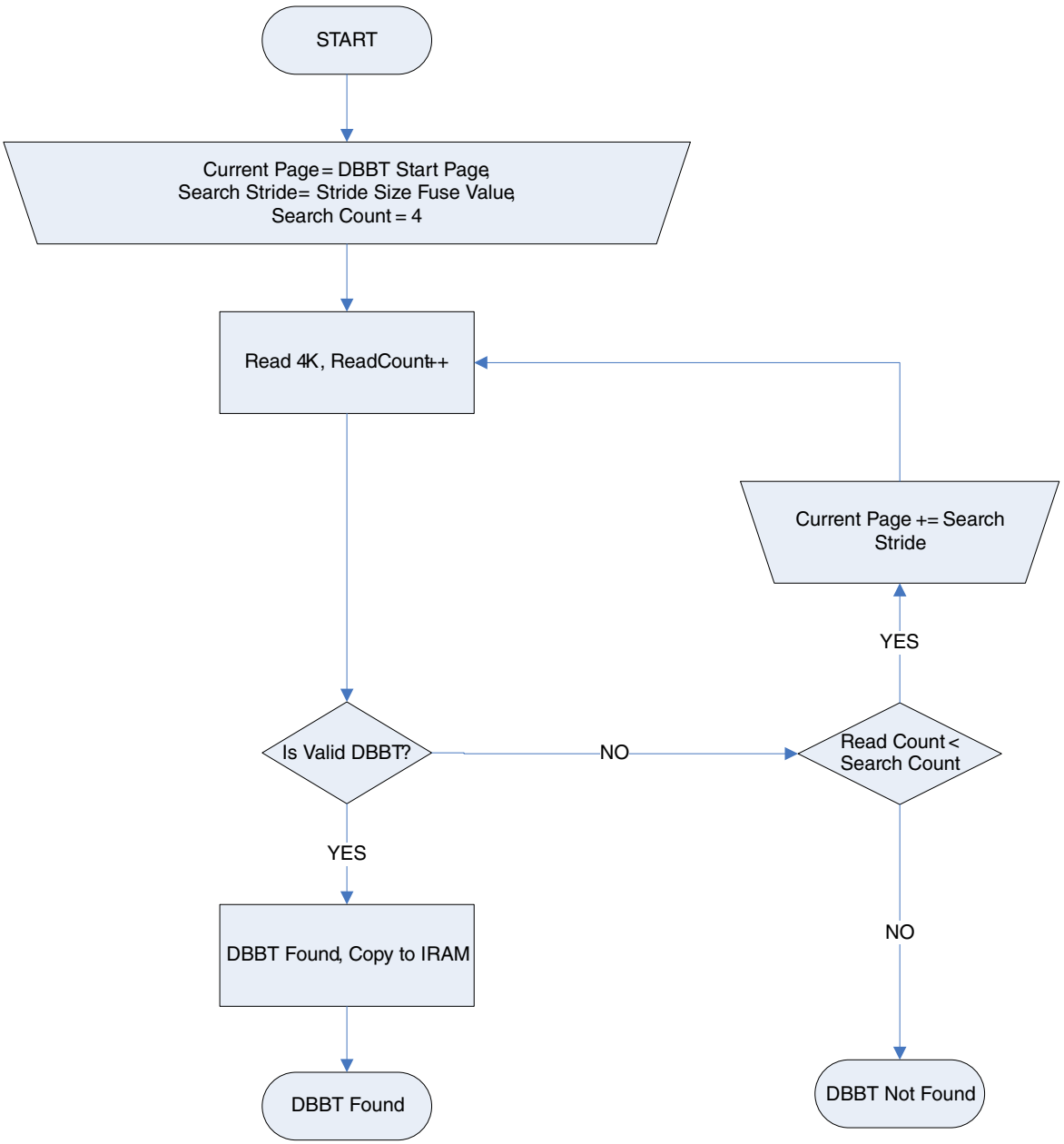


Figure 6-4. DBBT Search Flow

6.1.5.1.2 Firmware Configuration Block

The FCB is at the first page in the first good block. The FCB should be present at each search stride of the search area.

The search area contains copies of the FCB at each stride distance, in case the first Serial NAND block becomes corrupted, the ROM will find its copy in the next Serial NAND block. The search area should span over at least two Serial NAND blocks. The location information for DBBT search area and images are all specified in the FCB. Table 25-9 show the Flash Control Block Structure.

Table 6-9. Flash Control Block Structure

<table>
<tr><th>Name</th><th>Offset</th><th>Size Bytes</th><th>Description</th></tr>
<tr><td>crcChecksum</td><td>0x000</td><td>4</td><td>Checksum</td></tr>
<tr><td>fingerprint</td><td>0x004</td><td>4</td><td>0x4E46_4342
ASCII: “NFCB”</td></tr>
<tr><td>version</td><td>0x008</td><td>4</td><td>0x0000_0001</td></tr>
<tr><td>DBBTSearchStartPage</td><td>0x00C</td><td>4</td><td>Start Page address for bad block table search area</td></tr>
<tr><td>searchStride</td><td>0x010</td><td>2</td><td>Search stride for DBBT and FCB search. Not used by ROM Max value is 8.</td></tr>
<tr><td>searchCount</td><td>0x012</td><td>2</td><td>Copies of DBBT and FCB.
Not used by ROM, max value is 8.</td></tr>
<tr><td>firmwareCopies</td><td>0x014</td><td>4</td><td>Firmware copies
Valid range 1-8.</td></tr>
<tr><td>Reserved</td><td>0x018</td><td>40</td><td>Reserved for future use
Must be set to 0.</td></tr>
<tr><td>firmwareInfoTable</td><td>0x40</td><td>64</td><td>This table consists of (up to 8 entries):
<table>
<tr><th>Field</th><th>Size</th><th>Description</th></tr>
<tr><td>StartPage</td><td>4</td><td>Start page of this firmware</td></tr>
<tr><td>pageCount</td><td>4</td><td>Pages in this firmware</td></tr>
</table></td></tr>
<tr><td>Reserved</td><td>0x080</td><td>128</td><td>Reserved
Must be set to 0</td></tr>
<tr><td>spiNandConfigBlock</td><td>0x100</td><td>512</td><td>Serial NAND configuration block over FlexSPI</td></tr>
<tr><td>Reserved</td><td>0x300</td><td>256</td><td>Must be set to 0</td></tr>
</table>

NOTE

1. The “crcChecksum” is calculated with an MPEG2 variant of CRC-32. See Table 6-10 for more details.
2. The “crcChecksum” calculation starts from fingerprint to the end of FCB, 1020 bytes in total.
3. the “spiNandConfigBlock” is FlexSPI NAND configuration block which consists of common FlexSPI memory configuration block and Serial NAND specified configuration parameters. See Serial Flash boot through SPI for more details.

Table 6-10. CRC-32 variant algorithm

Property	Description
Width	32 bits
Polynomial	0x04C11BD7
Init Value	0xFFFFFFFF
Reflect in	False
Reflect Out	False
XOR Out	0x00000000

6.1.5.1.3 Discovered Bad Blocks Table (DBBT)

Table 6-11. DBBT Structure

Name	Offset	Size in Bytes	Decription
crcChecksum	0x000	4	Checksum
Fingerprint	0x004	4	32-bit word with a value of 0x4442_4254, in asci “DBBT”
Version	0x008	4	32-bit version number, this version of DBBT is 0x00000001
-	0x00C	4	Reserved
badBlockNumber	0x010	4	Number of bad blocks
Reserved	0x014	12	Must be filled with 0x00s
Bad Block entries	0x020-0x41F	256	Bad Block entries, only the first “badBlockNumber” entries are valid, the remaining entries must be filled with 0x00s

NOTE

1. Maximum bad block number is 256.

2. crcChecksum is calculated with the same algorithm as the one in FCB, from Fingerprint to the end of DBBT, 1052 bytes in total.

6.1.5.1.4 Bad block handling in ROM

During the firmware boot, at the block boundary, the Bad Block table is searched for a match to the next block.

If no match is found, the next block can be loaded. If a match is found, the block must be skipped and the next block checked.

If the Bad Block table start page is null, check the manufactory made Bad Block marker. The location of the Bad Block maker is at the first three or last three pages in every block of the NAND flash. The NAND manufacturers normally use one byte in the spare area of certain pages within a block to mark that a block is bad or not. A value of 0xFF means good block, non-FF means bad block.

To preserve the BI (bad block information), the flash updater or gang programmer applications must swap the Bad Block Information (BI) data to byte 0 of the metadata area for every page before programming the NAND flash. When the ROM loads the firmware, it copies back the value at metadata[0] to the BI offset in the page data. This figure shows how the factory bad block marker is preserved:

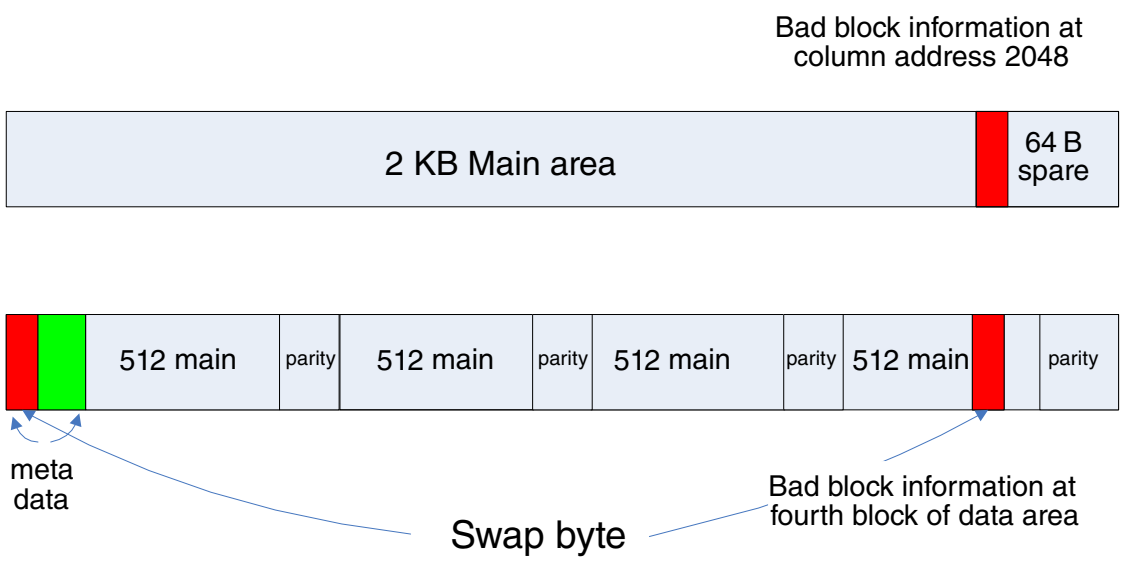


Figure 6-5. Factory bad block marker preservation

In the FCB structure, there are two elements (m_u32BadBlockMarkerByte and m_u32BadBlockMarkerStartBit) to indicate the byte and bit place in the page data that the manufacturer marked the bad block marker.

6.1.5.2 Serial NOR Flash Boot via FlexSPI

6.1.5.2.1 Serial NOR eFUSE Configuration

Table 6-12. Fuse definition for Serial NOR over FlexSPI

Fuse Config	Config	Definitions	Shipped Value	Settings
BOOT_MODE[3:1]	OEM	Boot device selection	0	011 - Boot from FlexSPI
BOOT_MODE[0]	OEM	FlexSPI interface selection	0	0 - Flash with 3B READ 1 - FlexSPI Hyperflash 3.3V
470[1:0]	OEM	FLEXSPI_AUTO_PROBE_TYPE FLEXSPI Flash AutoProbe Type	0	0 – QuadSPI NOR 1 – MXIC Octal 2 – Micron Octal 3 – Adesto Octal
470[2]	OEM	FLEXSPI_AUTO_PROBE_EN FLEXSPI Flash AutoProbe Enable	0	0 – Disabled 1 – Enabled
0x470[4:3] (Main)	OEM	OVERRIDE_FLEXSPI_BT_SEL_VAL FlexSPI Boot Selection Override Value	0	00 - Flex SPI (Hyperflash 1.8V) 01 - Flex SPI (Flash with 4B READ(0x13) default supported) 10 - Default Octal mode (Micron) 11 – Default Octal Mode (Mixc)
0x470[5] (Main)	OEM	OVERRIDE_FLEXSPI_BT_SEL Override FlexSPI boot selection	0	0 - Do not override 1 - Override
0x480[2:0]	OEM	FLEXSPI_FEQ_SEL FLEXSPI Flash Frequency	0	0 - 100 MHz 1 - 133 MHz 2 - 166 MHz 3 - 200 MHz 4 - 80 MHz 5 - 20 MHz Others – Reserved
480[6:3]	OEM	FLEXSPI_DUMMY_CYCLE_SEL FLEXSPI Flash Dummy Cycle	0	0 – Dummy cycles is auto-probed Others – Actual dummy cycles for Read command

Table continues on the next page...

Table 6-12. Fuse definition for Serial NOR over FlexSPI (continued)

Fuse Config	Config	Definitions	Shipped Value	Settings
480[19:18]	OEM	FLEXSPI_HOLD_TIME_SEL Hold time before read from device	0	0 – 500us 1 – 1ms 2 – 3ms 3 – 10ms

NOTE

If the xSPI FLASH Auto Probe feature is enabled, the following is the logic how this feature works with other fuse combinations:

- Flash Type - If Flash type is 0, the "xSPI FLASH Auto Probe Type" takes effect for the Flash type selection. If Flash Type is greater than or equal to 1, the "Flash Type" Fuse is used for Flash type selection, ROM will issue specific command to probe the presence of Serial NOR FLASH.
- xSPI FLASH Frequency - This field is used for specifying the Flash working frequency.

6.1.5.2.2 FlexSPI Serial NOR Flash Boot Operation

The Boot ROM attempts to boot from Serial NOR flash if the BOOT_MODE[3:1] fuses are programmed to 0b011 as shown in the Serial NOR eFUSE Configuration table, then the ROM will initialize FlexSPI interface as selected in the BOOT_MODE[0] settings of the FlexSPI eFUSE configuration. FlexSPI interface initialization is a two-step process.

The ROM expects the 512-byte FlexSPI NOR configuration parameters as explained in next section to be present at offset 0x400 in Serial NOR flash. The ROM reads these configuration parameters using the read command configured in the LUT of the FlexSPI interface with Serial clock operating at 30 MHz.

In the second step, ROM configures FlexSPI1 interface with the parameters provided in configuration block read from Serial NOR flash and starts the boot procedure. Refer to Table 25-14 for details regarding FlexSPI configuration parameters and to the FlexSPI NOR boot flow chart for detailed boot flow chart of FlexSPI NOR.

Both booting an XIP and non XIP image are supported from Serial NOR Flash. For XIP boot, the image has to be built for FlexSPI address space and for non XIP the image can be built to execute from Internal RAM.

6.1.5.2.3 FlexSPI NOR boot flow chart

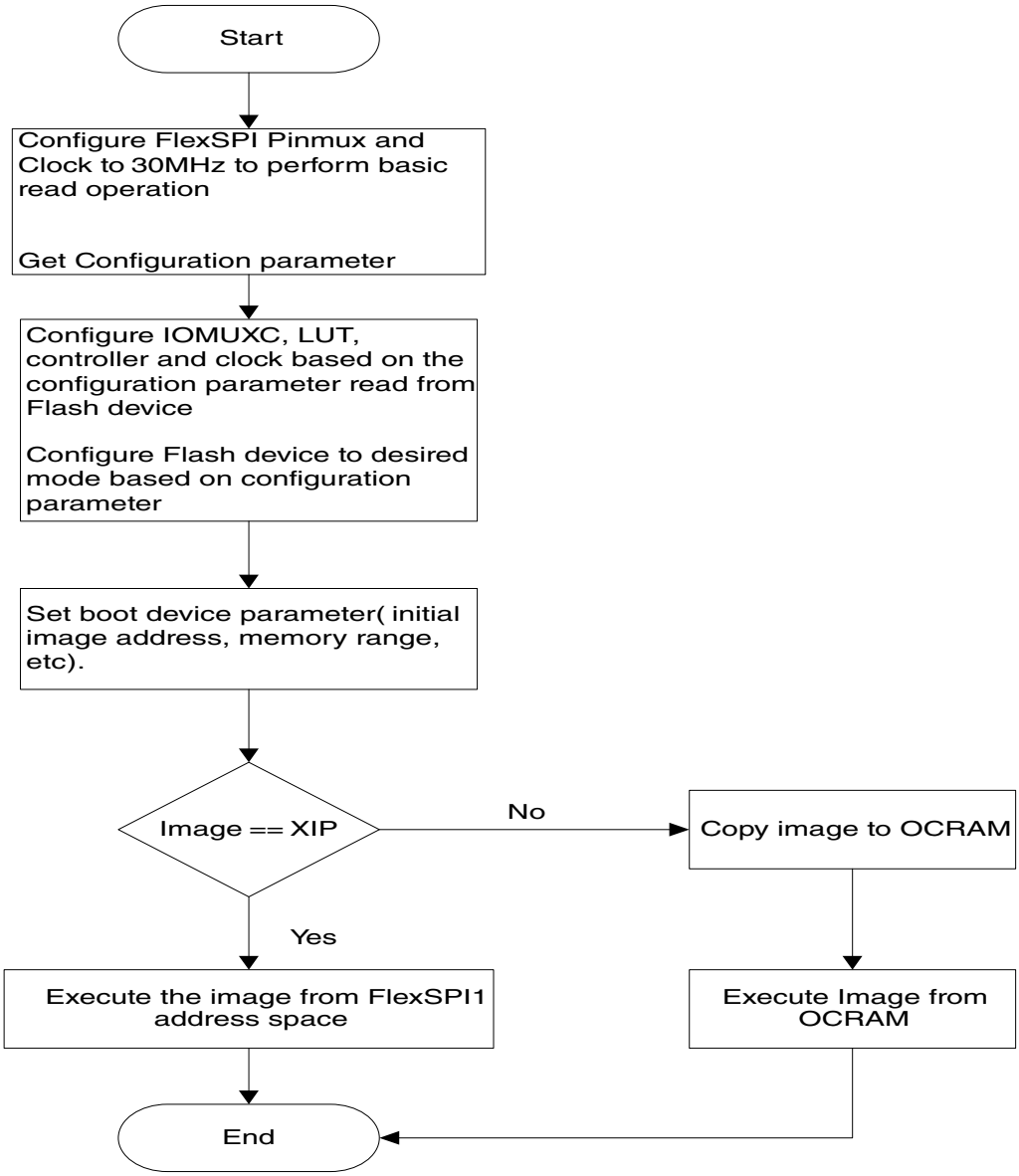


Figure 6-6. FlexSPI NOR boot flow

6.1.5.3 Serial NOR and NAND configuration based on FlexSPI interface

The ROM SW supports Serial NOR and Serial NAND based on FlexSPI module, using a 448-bytes common FlexSPI configuration block and several specified parameters for Serial NOR and Serial NAND respectively. See below sections for more details.

6.1.5.3.1 FlexSPI Configuration Block

FlexSPI Configuration block consists of parameters regarding specific Flash devices including read command sequence, quad mode enablement sequence (optional), etc.

Table 6-13. FlexSPI Configuration block

Name	Offset	Size(bytes)	Description
Tag	0x000	4	0x42464346, ascii:" FCFB"
Version	0x004	4	0x56010000 [07:00] bugfix = 0 [15:08] minor = 0 [23:16] major = 1 [31:24] ascii 'V'
-	0x008	4	Reserved
readSampleClkSrc	0x00C	1	0 – internal loopback 1 – loopback from DQS pad 3 – Flash provided DQS
dataHoldTime	0x00D	1	Serial Flash CS Hold Time Recommend default value is 0x03
dataSetupTime	0x00E	1	Serial Flash CS setup time Recommended default value is 0x03
columnAdressWidth	0x00F	1	3 – For HyperFlash 12/13 – For Serial NAND, see datasheet to find correct value 0 – Other devices
deviceModeCfgEnable	0x010	1	Device Mode Configuration Enable feature 0 – Disabled 1 – Enabled
-	0x011	3	Reserved
deviceModeSeq	0x014	4	Sequence parameter for device mode configuration
deviceModeArg	0x018	4	Device Mode argument, effective only when deviceModeCfgEnable = 1
configCmdEnable	0x01C	1	Config Command Enable feature 0 – Disabled 1 – Enabled
-	0x01D	3	Reserved
configCmdSeqs	0x020	16	Sequences for Config Command, allow 4 separate

Table continues on the next page...

Table 6-13. FlexSPI Configuration block (continued)

Name	Offset	Size(bytes)	Description
			configuration command sequences.
cfgCmdArgs	0x030	16	Arguments for each separate configuration command sequence.
controllerMiscOption	0x040	4	Bit0 – differential clock enable Bit1 – CK2 enable, must set to 0 in this silicon Bit2 – ParallelModeEnable, must set to 0 for this silicon Bit3 – wordAddressableEnable Bit4 – Safe Configuration Frequency enable set to 1 for the devices that support DDR Read instructions Bit5 – Pad Setting Override Enable Bit6 – DDR Mode Enable, set to 1 for device supports DDR read command
deviceType	0x044	1	1 – Serial NOR 2 – Serial NAND
sflashPadType	0x045	1	1 – Single pad 2 – Dual pads 4 – Quad pads 8 – Octal pads
serialClkFreq	0x046	1	Chip specific value, for this silicon 1 – 30 MHz 2 – 50 MHz 3 – 60 MHz 4 – 75 MHz 5 – 80 MHz 6 – 100 MHz 7 – 133 MHz 8 – 166 MHz Other value: 30 MHz
lutCustomSeqEnable	0x047	1	0 – Use pre-defined LUT sequence index and number 1 - Use LUT sequence parameters provided in this block

Table continues on the next page...

Table 6-13. FlexSPI Configuration block (continued)

Name	Offset	Size(bytes)	Description
-	0x048	8	Reserved
sflashA1Size	0x050	4	For SPI NOR, need to fill with actual size For SPI NAND, need to fill with actual size * 2
sflashA2Size	0x054	4	The same as above
sflashB1Size	0x058	4	The same as above
sflashB2Size	0x05C	4	The same as above
csPadSettingOverride	0x060	4	Set to 0 if it is not supported
sclkPadSettingOverride	0x064	4	Set to 0 if it is not supported
dataPadSettingOverride	0x068	4	Set to 0 if it is not supported
dqsPadSettingOverride	0x06C	4	Set to 0 if it is not supported
timeoutInMs	0x070	0	Maximum wait time during read busy status 0 – Disabled timeout checking feature Other value – Timeout if the wait time exceeds this value.
commandInterval	0x074	4	Unit: ns Currently, it is used for SPI NAND only at high frequency
dataValidTime	0x078	4	Time from clock edge to data valid edge, unit ns. This field is used when the FlexSPI Root clock is less than 100 MHz and the read sample clock source is device provided DQS signal without CK2 support. [31:16] data valid time for DLLB in terms of 0.1 ns [15:0] data valid time for DLLA in terms of 0.1 ns
busyOffset	0x07C	2	busy bit offset, valid range : 0-31
busyBitPolarity	0x07E	2	0 – busy bit is 1 if device is busy 1 – busy bit is 0 if device is busy
lookupTable	0x080	256	Lookup table
lutCustomSeq	0x180	48	Customized LUT sequence, see below table for details.
	0x1B0	16	Reserved for future use

Note:

1. To customize the LUT sequence for some specific device, users need to enable “lutCustomSeqEnable” and fill in corresponding “lutCustomSeq” field specified by command index below.
2. For Serial (SPI) NOR, the pre-defined LUT index is as follows:

Table 6-14. LUT sequence definition for Serial NOR

Command Index	Name	Index in lookup table	Description
0	Read	0	Read command Sequence
1	ReadStatus	1	Read Status command
2	WriteEnable	3	Write Enable command sequence
3	EraseSector	5	Erase Sector Command
4	PageProgram	9	Page Program Command
5	ChipErase	11	Full Chip Erase
6	Dummy	15	Dummy Command as needed
	Reserved	2,4,6,7,8,10,12,13,14	All reserved indexes can be freely used for other purpose

6.1.5.3.2 Serial NOR configuration block (512 bytes)

Table 6-15. Serial NOR configuration block

Name	Offset	Size (Bytes)	Description
memCfg	0	448	The common memory configuration block, see FlexSPI configuration block for more details
pageSize	0x1C0	4	Page size in terms of bytes, not used by ROM
sectorSize	0x1C4	4	Sector size in terms of bytes, not used by ROM
ipCmdSerialClkFreq	0x1C8	4	Chip specific value, not used by ROM For Ultra 0 – No change, keep current serial clock unchanged 1 – 30 MHz 2 – 50 MHz 3 – 60 MHz 4 – 75 MHz 5 – 80 MHz

Table continues on the next page...

Table 6-15. Serial NOR configuration block (continued)

Name	Offset	Size (Bytes)	Description
			6 – 100 MHz 7 – 133 MHz 8 – 166 MHz
Reserved	0x1CC	52	Reserved for future use

6.1.5.3.3 Serial NAND configuration block(512 bytes)

Table 6-16. Serial NAND configuration block

Name	Offset	Size (Bytes)	Description
memCfg	0	448	The common memory configuration block, see FlexSPI configuration block for more details
pageDataSize	0x1C0	4	Page size in terms of bytes, usually, it is 2048 or 4096
pageTotalSize	0x1C4	4	It equals to 2 ^ width of column adddress
pagesPerBlock	0x1C8	4	Pages in one block
bypassReadStatus	0x1CC	1	0 – Read Status Register 1 – Bypass Read status register
bypassEccRead	0x1CD	1	0 – Perform ECC read 1 – Bypass ECC read
hasMultiPlanes	0x1CE	1	0 – Only 1 plane 1 – Has two planes
skippOddBlocks	0x1CF	1	0 – Read Odd blocks 1 – Skip Odd blocks
eccCheckCustomEnable	0x1D0	1	0 – Use the common ECC check command and ECC related masks 1 - Use ECC check related masks provided in this configuration block
ipCmdSerialClkFreq	0x1D1	1	Chip specific value, not used by ROM 0 – No change, keep current serial clock unchanged 1 – 30 MHz 2 – 50 MHz 3 – 60 MHz 4 – 75 MHz

Table continues on the next page...

Table 6-16. Serial NAND configuration block (continued)

Name	Offset	Size (Bytes)	Description
			5 – 80 MHz 6 – 100 MHz 7 – 133 MHz 8 – 166 MHz
readPageTimeUs	0x1D2	2	Wait time during page read, this field will take effect on if the bypassReadStatus is set to 1
eccStatusMask	0x1D4	4	ECC Status Mask
eccFailureMask	0x1D8	4	ECC Check Failure mask
blocksPerDevice	0x1DC	4	Blocks in a Serial NAND
Reserved	0x1ED	32	Reserved for future use

Below is an example of Serial NAND configuration block for Winbond W25N01GVZEIG:

```
const flexspi_nand_config_t kSerialNandCfgBlk =
{
    .memConfig =
    {
        .tag = FLEXSPI_CFG_BLK_TAG,
        .version = FLEXSPI_CFG_BLK_VERSION,
        .readSampleClkSrc = kFlexSPIReadSampleClk_LoopbackInternally,
        .dataHoldTime = 3,
        .dataSetupTime = 3,
        .columnAddressWidth = 12,
        .deviceModeCfgEnable = 1,
        .deviceModeSeq = { 1, 2 },
        .deviceType = kFlexSpiDeviceType_SerialNAND,
        .sflashPadType = kSerialFlash_4Pads,
        .serialClkFreq = kFlexSpiSerialClk_50MHz,
        .lutCustomSeqEnable = 0,
        .sflashA1Size = 128 * 1024 * 1024U * 2, // Flash size = 2 * actual data size
(exclude spare space)
        .lookupTable =
        {
            // Read cache 4 I/0
            [4 * NOR_CMD_LUT_SEQ_IDX_READ] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD, 0xEB,
CADDR_SDR, FLEXSPI_4PAD, 0x10),
            [4 * NOR_CMD_LUT_SEQ_IDX_READ + 1] = FLEXSPI_LUT_SEQ(DUMMY_SDR, FLEXSPI_4PAD,
0x04, READ_SDR, FLEXSPI_4PAD, 0x80),
            // Clear Status1 flag
            [4 * 2] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD, 0x1F, CMD_SDR, FLEXSPI_1PAD,
0xA0),
            [4 * 2 + 1] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD, 0x00, STOP, FLEXSPI_1PAD,
0x00),
            // Read Page
            [4 * NAND_CMD_LUT_SEQ_IDX_READPAGE] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD,
0x13, RADDR_SDR, FLEXSPI_1PAD, 0x18),
            // Read Status
            [4 * NAND_CMD_LUT_SEQ_IDX_READSTATUS] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD,
0x0F, CMD_SDR, FLEXSPI_1PAD, 0xC0),
            [4 * NAND_CMD_LUT_SEQ_IDX_READSTATUS + 1] = FLEXSPI_LUT_SEQ(READ_SDR,
FLEXSPI_1PAD, 0x01, STOP, FLEXSPI_1PAD, 0),

            // Write Enable
```

```
            [4 * NAND_CMD_LUT_SEQ_IDX_WRITEENABLE] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD,
0x06, STOP, FLEXSPI_1PAD, 0),

            // Page Program Load 4x
            [4 * NAND_CMD_LUT_SEQ_IDX_PROGRAMLOAD] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD,
0x32, CADDR_SDR, FLEXSPI_1PAD, 0x10),
            [4 * NAND_CMD_LUT_SEQ_IDX_PROGRAMLOAD + 1] = FLEXSPI_LUT_SEQ(WRITE_SDR,
FLEXSPI_4PAD, 0x40, STOP, FLEXSPI_1PAD, 0),
            // Page Program Execute
            [4 * NAND_CMD_LUT_SEQ_IDX_PROGRAMEXECUTE] = FLEXSPI_LUT_SEQ(CMD_SDR,
FLEXSPI_1PAD, 0x10, RADDR_SDR, FLEXSPI_1PAD, 0x18),
            // Erase Sector
            [4 * NAND_CMD_LUT_SEQ_IDX_ERASEBLOCK] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD,
0xD8, RADDR_SDR, FLEXSPI_1PAD, 0x18),
            // Read ECC status
            [4 * NAND_CMD_LUT_SEQ_IDX_READECCSTAT] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD,
0x0F, CMD_SDR, FLEXSPI_1PAD, 0xC0),
            [4 * NAND_CMD_LUT_SEQ_IDX_READECCSTAT + 1] = FLEXSPI_LUT_SEQ(READ_SDR,
FLEXSPI_1PAD, 0x01, STOP, FLEXSPI_1PAD, 0),
        },
    },
    .pageDataSize = 2048,
    .pageTotalSize = 4096,
    .pagesPerBlock = 64,
};
```

6.1.5.4 NAND flash

The boot ROM supports a number of MLC/SLC NAND flash devices from different vendors and LBA NAND flash devices. The Error Correction and Control (ECC) subblock (BCH) is used to detect the errors.

6.1.5.4.1 NAND eFUSE configuration

The boot ROM determines the configuration of the external NAND flash by parameters, either provided by the eFUSE, or sampled on the BOOT_MODE[3:0] pins during boot. See Table 6-17 for parameters details.

Table 6-17. NAND boot eFUSE descriptions

Fuse	Config	Definition	Shipped value	Settings
4A0[15]	OEM	BT_TOGGLE_MODE	0	0—raw NAND 1—toggle mode NAND
470[7:6]	OEM	OVERRIDE_NAND_PG_PER_BLK_VAL Pages in block	0	RAWNAND: 00 - 32 pages 01 - 64 pages 10 - 128 pages 11 - 32 pages FlexSPI NAND: 00 - 64 pages 01 - 128 pages 10 - 256 pages 11 - 32 pages

Table continues on the next page...

Table 6-17. NAND boot eFUSE descriptions (continued)

Fuse	Config	Definition	Shipped value	Settings
4B0[11:10]	OEM	Row address cycles	00	00—3 01—2 10—4 11—5
4A0[12:9]	OEM	Toggle mode 33 MHz preamble delay, read latency	000	0000—16 GPMICLK cycles 0001—1 GPMICLK cycles 0010—2 GPMICLK cycles 0011—3 GPMICLK cycles 0100—4 GPMICLK cycles 0101—5 GPMICLK cycles 0110—6 GPMICLK cycles 0111—7 GPMICLK cycles 1000—8 GPMICLK cycles 1001—9 GPMICLK cycles 1010—10 GPMICLK cycles 1011—11 GPMICLK cycles 1100—12 GPMICLK cycles 1101—13 GPMICLK cycles 1110—14 GPMICLK cycles 1111—15 GPMICLK cycles
4A0[14:13]	OEM	Boot search count	00	00—2 01—2 10—4 11—8
0x4B0[7]	OEM	Override pad settings	0	Override the NAND pad settings 0—use the default values 1—use the PAD_SETTINGS value
0x4A0[31:24]	OEM	PAD_SETTINGS[7:0]	0	NAND pad settings value
0x4B0[15:12]	OEM	READ_RETRY_SEQ_ID[3:0]	0000	0000—don't use the ROM embedded read-retry sequence 0001—use Micron 20 nm read-retry sequence 0010—use Toshiba A19nm read-retry sequence 0011—use Toshiba 19nm read-retry sequence 0100—use SanDisk 19nm read-retry sequence

Table 6-17. NAND boot eFUSE descriptions

Fuse	Config	Definition	Shipped value	Settings
				0101—use SanDisk 1ynm read-retry sequence 0110 to 1111—reserved

6.1.5.4.2 NAND flash boot flow and Boot Control Blocks (BCB)

There are two BCB data structures:

- FCB
- DBBT

As a part of the NAND media initialization, the ROM driver uses safe NAND timings to search for the Firmware Configuration Block (FCB) that contains the optimum NAND timings, the page address of the Discovered Bad Block Table (DBBT) Search Area, and the start page address of the primary and secondary firmware.

The hardware ECC level to use is embedded inside the FCB block. The FCB data structure is also protected using the ECC. The driver reads raw 2112 bytes of the first sector and runs through the software ECC engine that determines whether the FCB data is valid or not.

If the FCB is found, the optimum NAND timings are loaded for further reads. If the ECC fails, or the fingerprints do not match, the Block Search state machine increments the page number to the Search Stride number of pages to read for the next BCB until the SearchCount pages have been read.

If the search fails to find a valid FCB, the NAND driver responds with an error and the boot ROM enters the serial download mode.

The FCB contains the page address of the DBBT Search Area, and the page address for primary and secondary boot images. The DBBT is searched in the DBBT Search Area, just like the FCB is searched. After the FCB is read, the DBBT is loaded, and the primary or secondary boot image is loaded using the starting page address from the FCB.

This figure shows the state diagram of the FCB search:

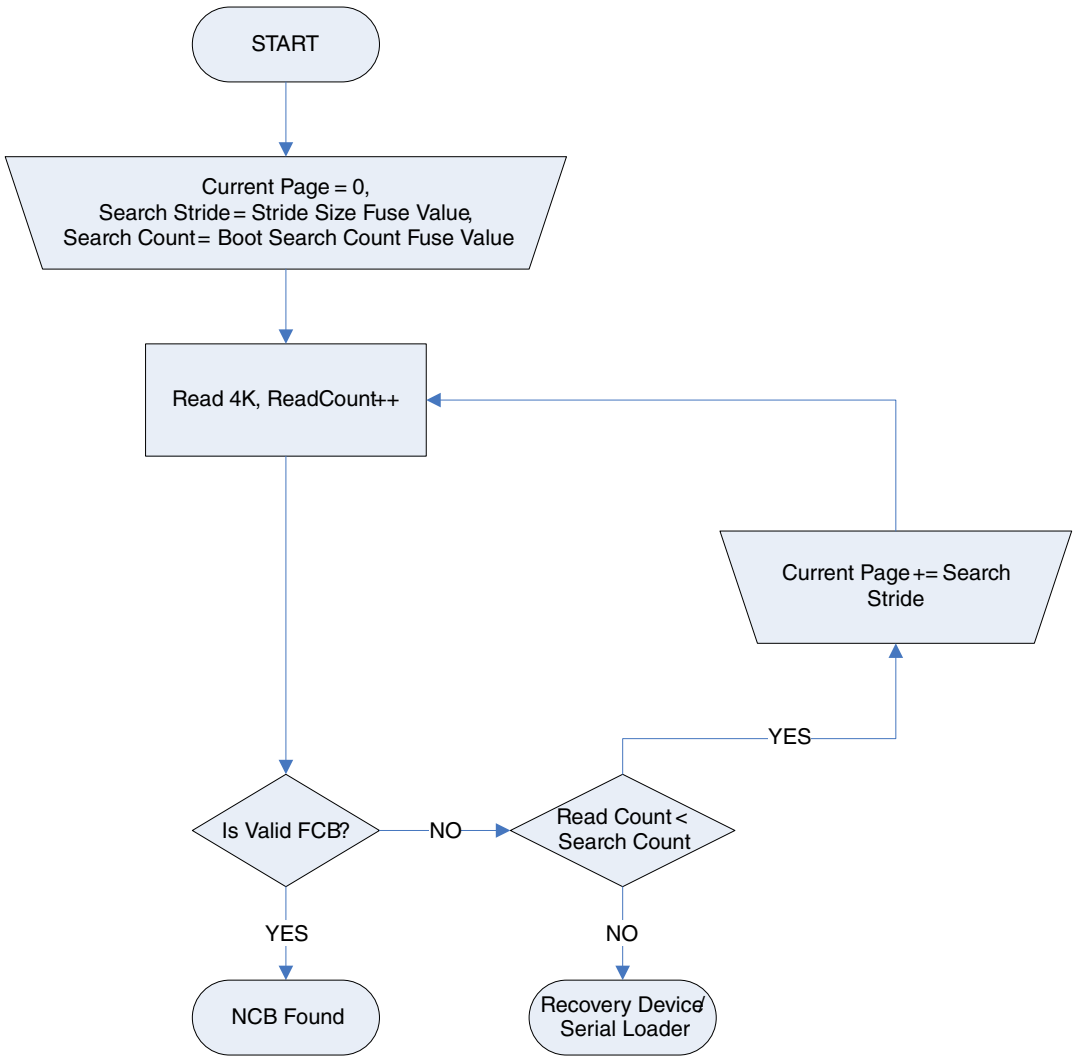


Figure 6-7. FCB search flow

When the FCB is found, the boot ROM searches for the Discovered Bad Blocks Table (DBBT). If the DBBT Search Area is 0 in the FCB, the ROM assumes that there are no bad blocks on the NAND device boot area. See this figure for the DBBT search flow:

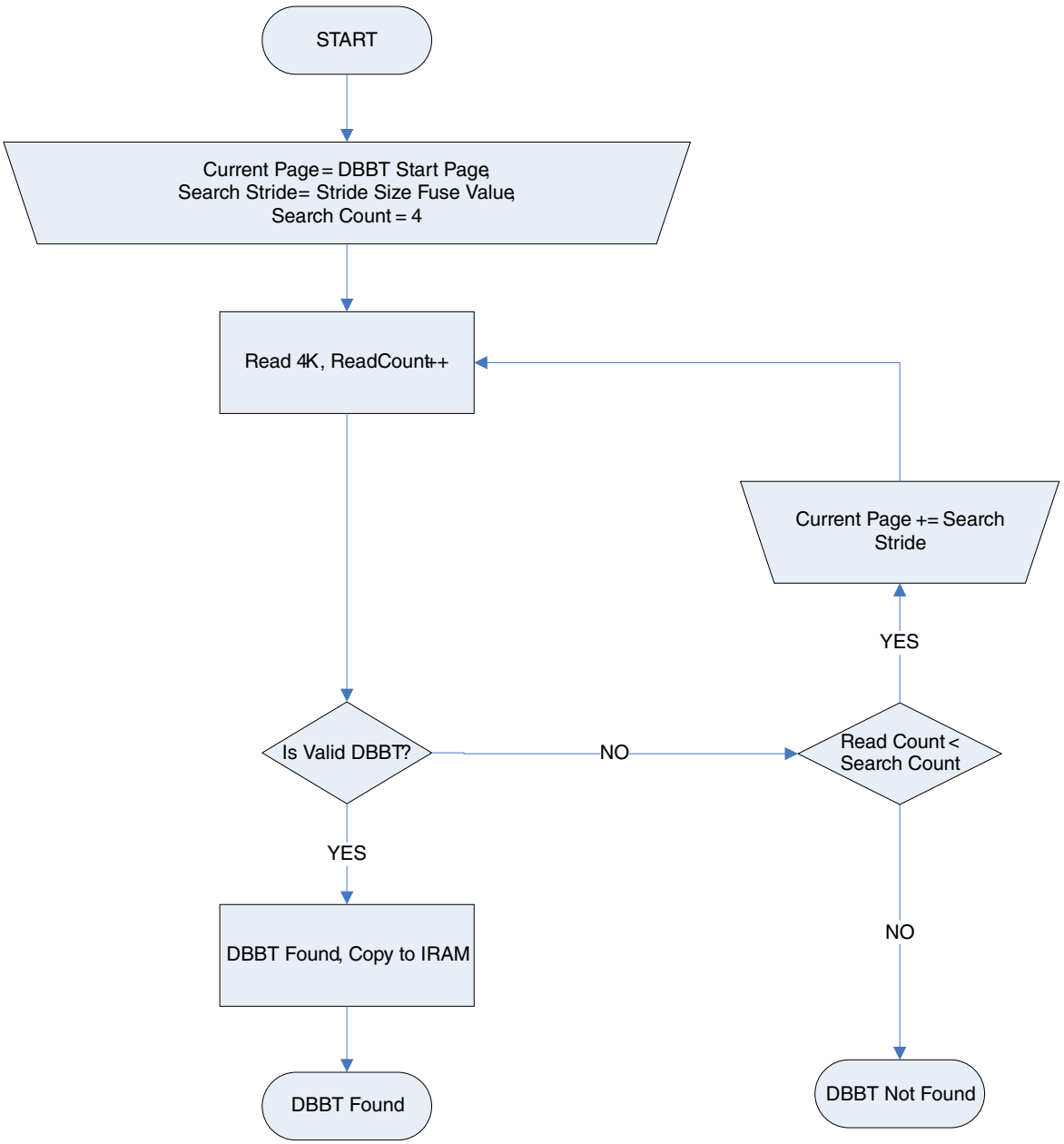


Figure 6-8. DBBT search flow

6.1.5.4.3 Firmware configuration block

The FCB is the first sector in the first good block. The FCB must be present at each search stride of the search area.

The search area contains copies of the FCB at each stride distance, so, in case the first NAND block becomes corrupted, the ROM finds its copy in the next NAND block. The search area must span over at least two NAND blocks. The location information for the DBBT search area, FW1, and FW2 are all specified in the FCB. This table shows the flash control block structure:

Table 6-18. Flash control block structure

Name	Start byte	Size in bytes	Description
Reserved	0	4	Reserved for Fingerprint #1(Checksum)
FingerPrint	4	4	32-bit word with a value of 0x20424346, in ascii "FCB"
Version	8	4	32-bit version number; this version of FCB is 0x00000001
m_NANDTiming	12	8	8 B of data for eight NAND timing parameters from the NAND datasheet. The eight parameters are: m_NandTiming[0]=data_setup, m_NandTiming[1]=data_hold, m_NandTiming[2]=address_setup, m_NandTiming[3]=dsample_time, m_NandTiming[4]=nand_timing_state, m_NandTiming[5]=REA, m_NandTiming[6]=RLOH, m_NandTiming[7]=RHOH. The ROM only uses the first four parameters, but the FCB provides space for other four parameters to be used by the bootloader or other applications.
PageDataSize	20	4	The number of bytes of data in a page. Typically, this is 2048 bytes for 2112 bytes page size or 4096 bytes for 4314/4224 bytes page size or 8192 for 8568 bytes page size.
TotalPageSize	24	4	The total number of bytes in a page. Typically, 2112 for 2-KB page or 4224 or 4314 for 4-KB page or 8568 for 8-KB page.
SectorsPerBlock	28	4	The number of pages per block. Typically 64 or 128 or depending on the NAND device type.
NumberOfNANDs	32	4	Not used by ROM
TotalInternalDie	36	4	Not used by ROM
CellType	40	4	Not used by ROM
EccBlockNEccType	44	4	Value from 0 to is used to set the BCH Error Corrrection level 0, 2, 4, .. or 62 for Block BN of ECC page, used in configuring the BCH62 page layout registers.
EccBlock0Size	48	4	Size of block B0 used in configuring the BCH62 page-layout registers.

Table continues on the next page...

Table 6-18. Flash control block structure (continued)

Name	Start byte	Size in bytes	Description
EccBlockNSize	52	4	Size of block BN used in configuring the BCH62 page-layout registers.
EccBlock0EccType	56	4	Value from 0 to used to set the BCH Error Corrrection level 0, 2, 4, .. or 62 for Block BN of ECC page, used in configuring the BCH62 page layout registers.
MetadataBytes	60	4	Size of metadata bytes used in configuring the BCH62 page-layout registers.
NumEccBlocksPerPage	64	4	Number of the ECC blocks BN not including B0. This value is used in configuring the BCH62 page-layout registers.
EccBlockNEccLevelSDK	68	4	Not used by ROM
EccBlock0SizeSDK	72	4	Not used by ROM
EccBlockNSizeSDK	76	4	Not used by ROM
EccBlock0EccLevelSDK	80	4	Not used by ROM
NumEccBlocksPerPageSDK	84	4	Not used by ROM
MetadataBytesSDK	88	4	Not used by ROM
EraseThreshold	92	4	Not used by ROM
Firmware1_startingPage	104	4	Page number address where the first copy of bootable firmware is located.
Firmware2_startingPage	108	4	Page number address where the second copy of bootable firmware is located.
PagesInFirmware1	112	4	Size of the first copy of firmware in pages.
PagesInFirmware2	116	4	Size of the second copy of firmware in pages.
DBBTSearchAreaStartAddress	120	4	Page address for the bad block table search area.
BadBlockMarkerByte	124	4	This is an input offset in the BCH page for the ROM to swap with the first byte of metadata after reading a page using the BCH62. The ROM supports the restoration of manufacturer-marked bad block markers in the page and this offset is the bad block marker offset location.
BadBlockMarkerStartBit	128	4	This is an input bit offset in the BadBlockMarkerByte for the ROM to use when swapping eight bits with the first byte of metadata.
BBMarkerPhysicalOffset	132	4	This is the offset where the manufacturer leaves the bad block marker on a page.
BCHType	136	4	0 for BCH20 and 1 for BCH62. The chip is backwards compatible to BCH20 and this field tells the ROM to use the BCH20 or BCH62 block.
TMTiming2_ReadLatency	140	4	Toggle mode NAND timing parameter read latency, the ROM uses this value to configure the timing2 register of the GPMI.

Table continues on the next page...

Table 6-18. Flash control block structure (continued)

Name	Start byte	Size in bytes	Description
TMTiming2_PreambleDelay	144	4	Toggle mode NAND timing parameter Preamble Delay. The ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_CEDelay	148	4	Toggle mode NAND timing parameter CE Delay. The ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_PostambleDelay	152	4	Toggle mode NAND timing parameter Postamble Delay. The ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_CmdAddPause	156	4	Toggle mode NAND timing parameter Cmd Add Pause. The ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_DataPause	160	4	Toggle mode NAND timing parameter Data Pause. The ROM uses this value to configure the timing2 register of the GPMI.
TMSpeed	164	4	This is the toggle mode speed for the ROM to configure the gpmi clock. 0 for 33 MHz, 1 for 40 MHz, and 2 for 66 MHz.
TMTiming1_BusyTimeout	168	4	Toggle mode NAND timing parameter Busy Timeout. The ROM uses this value to configure the timing1 register of the GPMI.
DISBBM	172	4	If 0, the ROM swaps the BadBlockMarkerByte with metadata[0] after reading a page using the BCH62. If the value is 1, the ROM does not swap.
BBMark_spare_offset	176	4	The offset in the metadata place which stores the data in the bad block marker place.
Onfi_sync_enable	180	4	Enable the Onfi nand sync mode support.
Onfi_sync_speed	184	4	Speed for the Onfi nand sync mode: 0 - 24 MHz, 1 - 33 MHz, 2 - 40 MHz, 3 - 50 MHz, 4 - 66 MHz, 5 - 80 MHz, 6 - 100 MHz, 7 - 133 MHz, 8 - 160 MHz, 9 - 200 MHz
Onfi_syncNANDData	188	28	The parameters for the Onfi nand sync mode timing. They are read latency, ce_delay, preamble_delay, postamble_delay, cmdadd_pause, data_pause, and busy_timeout.
DISBB_Search	216	4	Disable the bad block search function when reading the firmware, only using DBBT.

The FCB data structure is protected using a 62-bit ECC. The layout of the FCB page is illustrated in this figure:

Figure 6-9. Layout of the FCB page

The detailed parameters of the FCB pages are listed in this table:

Table 6-19. Parameters setting for FCB page

Parameter	Value
TotalPageSize	2048+64=2112
MetadataBytes	32
EccBlock0Size	128
EccBlock0EccType	31
BCHType	0
EccBlockNSize	128
EccBlockNEccType	31
NumEccBlocksPerPage	7

To reduce the disturbances caused by a neighboring cell in the FCB page in the NAND chip, a randomizer is enabled when reading the FCB page. BCH ECC has a Randomizer module that is interfaced through the GPMI APBHDMA chain. The Randomizer can generate random data based on BCH ECC encoded/decoded data. It can be employed to reduce the disturbances caused by a neighboring cell in the NAND chip, thus reducing bit errors. The randomizer is used to reduce the bit errors in the FCB. Ensure that the randomizer is enabled when burning the FCB pages in the NAND flash. To control the randomizer for the pages (except for FCB), a new field called Randomizer_Enable is added into the FCB structure. If the Randomizer_Enable field is set to 0, the randomizer is disabled. Reading the pages (except for FCB) being set to a non-zero value enables the randomizer. For detailed randomizer information, see Randomizer.

6.1.5.4.4 Discovered Bad Block Table (DBBT)

See this table for the DBBT format:

Table 6-20. DBBT structure

Name	Start byte	Size in bytes	Description
Reserved	0	4	-
FingerPrint	4	4	32-bit word with a value of 0x44424254,in ascii "DBBT"
Version	8	4	32-bit version number; this version of DBBT is 0x00000001
Reserved	12	4	-
DBBT_NUM_OF_PAGES	16	4	Size of the DBBT in pages
Reserved	20	4*PageSize-20	-
Reserved	4*PageSize	4	-
Number of Entries	4*PageSize + 4	4	Number of bad blocks
Bad Block Number 1	4*PageSize + 8	4	First bad block number
Bad Block Number 2	4*PageSize + 12	4	Second bad block number
...	...	...	...
Bad Block Number n	4*PageSize + 4*(n+1)	4	The n^{th} bad block

6.1.5.4.5 Bad block handling in ROM

During the firmware boot, at the block boundary, the Bad Block table is searched for a match to the next block.

If no match is found, the next block can be loaded. If a match is found, the block must be skipped and the next block checked.

If the Bad Block table start page is null, check the manufactory made Bad Block marker. The location of the Bad Block maker is at the first three or last three pages in every block of the NAND flash. The NAND manufacturers normally use one byte in the spare area of certain pages within a block to mark that a block is bad or not. A value of 0xFF means good block, non-FF means bad block.

To preserve the BI (bad block information), the flash updater or gang programmer applications must swap the Bad Block Information (BI) data to byte 0 of the metadata area for every page before programming the NAND flash. When the ROM loads the firmware, it copies back the value at metadata[0] to the BI offset in the page data. This figure shows how the factory bad block marker is preserved:

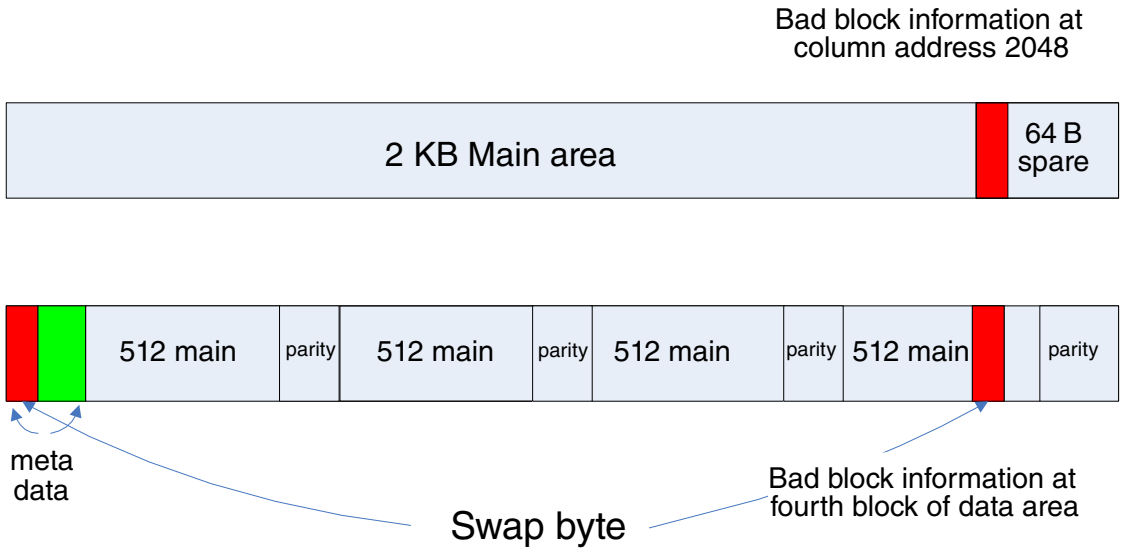


Figure 6-10. Factory bad block marker preservation

In the FCB structure, there are two elements (m_u32BadBlockMarkerByte and m_u32BadBlockMarkerStartBit) to indicate the byte and bit place in the page data that the manufacturer marked the bad block marker.

6.1.5.4.6 Toggle mode DDR NAND boot

If the BT_TOGGLEMODE efuse is blown, the ROM does the following to boot from the Samsung's toggle mode DDR NAND.

6.1.5.4.6.1 GPMI and BCH clocks configuration

The ROM sets the clock source and the dividers in the CCM registers.

If the 0x4A0[15] is set (toggle mode), the GPMI/BCH CLK source is PLL2PFD4, and running at 66 MHz, otherwise the GPMI/ BCH CLK souce is PLL3, running at 24 MHz. The ROM sets the default values to timing0, timing1, and timing2 gpmi registers for 24 MHz clock speed. It uses the 0x4A0[12:9] fuse to configure the GPMI timing2 register parameters preamble delay and read latency. The default value for these parameters is 2 when the fuses are not blown.

The default timing parameter values used by the ROM for the toggle-mode device are:

- Timing0.ADDRESS_SETUP = 5
- Timing0.DATA_SETUP = 10
- Timing0.DATA_HOLD = 10
- Timing1.DEVICE_BUSY_TIMEOUT = 0 x 500

- Timing2.READ_LATENCY = 0x4A0[12:9] fuse selection, if blown, otherwise 2
- Timing2.CE_DELAY = 2
- Timing2.PREAMBLE_DELAY = 0x4A0[12:9] fuse selection, if blown, otherwise 2
- Timing2.POSTAMBLE_DELAY = 3
- Timing2.CMDADD_PAUSE = 4
- Timing2.DATA_PAUSE = 6

The default timing parameters can be overriden by the TMTiming2_ReadLatency, TMTiming2_PreambleDelay, TMTiming2_CEDelay, TMTiming2_PostambleDelay, TMTiming2_CmdAddPause, and TMTiming2_DataPause parameters of the FCB.

6.1.5.4.6.2 Setup DMA for DDR transfers

In the DMA descriptors, the GPMI is configured to read the page data at a double data rate, the word length is set to 16, and the transfer count to a half of the page size.

6.1.5.4.6.3 Reconfigure timing and speed using values in FCB

After reading the FCB page with the GPMI set to default timings and a speed of 33 MHz, the ROM reconfigures the CCM dividers to run the gpmi/bch clks to a desired speed specified in the FCB for the rest of the boot process. The GPMI timing registers are also reconfigured to the values specified in the FCB.

The GPMI speed can be configured using the FCB parameter TMSpeed:

- 0—25 MHz
- 1—33 MHz
- 2—40 MHz
- 3—50 MHz
- 4—66 MHz
- 5—80 MHz
- 6—100 MHz
- 7—133 MHz
- 8—133 MHz
- 9—200 MHz

The GPMI timing0 register fields data_setup, data_hold, and address_setup are set to the values specified for the data_setup and data_hold and address_setup in the FCB member m_NANDTiming.

The GPMI timing1.DEVICE_BUSY_TIMEOUT is set to the value specified in the FCB member TMTiming1_BusyTimeout.

The GPMI timing2 register values are set using the FCB members TMTiming2.READ_LATENCY, CE_DELAY, PREAMBLE_DELAY, POSTAMBLE_DELAY, CMDADD_PAUSE, and DATA_PAUSE.

6.1.5.4.7 Typical NAND page organization

6.1.5.4.7.1 BCH ECC page organization

The first data block is called block 0 and the rest of the blocks are called block N. A separate ECC level scan is used for block 0 and block N.

The metadata bytes must be located at the beginning of a page, starting at byte 0, followed by the data block 0, the ECC bytes for data block 0, the block 1 and its ECC bytes, and so on, up until the N data blocks. The ECC level for the block 0 can be different from the ECC level for the rest of the blocks.

For the NAND boot with page-size restrictions and the data block size restricted to 512 B, only few combinations of the ECC for block 0 and block N are possible.

This figure shows the valid layout for 2112-byte sized page.

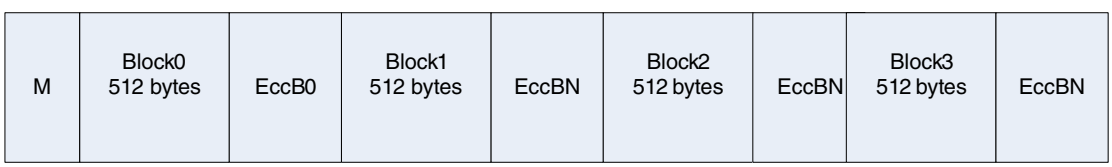


Figure 6-11. Valid layout for 2112-byte sized page

The example below is for 13 bits of parity (GF13). The number of ECC bits required for a data block is calculated using the (ECC_Correction_Level * 13) bits.

In the above layout, the ECC size for EccB0 and EccBN must be selected to not exceed a total page size of 2112 bytes. The EccB0 and EccBN can be one of the 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20 bits on the ECC correction level. The total bytes are:

[M + (data_block_size x 4) + ([EccB0 + (EccBN x 3)] x 13) / 8] <= 2112;

M = metadata bytes and data_block_size is 512.

There are four data blocks of 512 bytes each in a page of 2-KB page sized NAND. The values of EccB0 and EccBN must be such that the above calculation does not result in a value greater than 2112 bytes.

M	Block0 512 bytes	EccB0	Block1 512 bytes	EccBN	Block2 512 bytes	EccBN	Block3 512 bytes	EccBN
	Block4 512 bytes	EccBN	Block5 512 bytes	EccBN	Block6 512 bytes	EccBN	Block7 512 bytes	EccBN

Figure 6-12. Valid layout for 4-KB sized page

Different NAND manufacturers have different sizes for a 4-KB page; 4314 bytes is typical.

[M + (data_block_size x 8) + ([EccB0 + (EccBN x 7)] x 13) / 8] <= 4314;

M= metadata bytes and data_block_size is 512.

There are eight data blocks of 512 bytes each in a page of a 4-KB page sized NAND. The values of the EccB0 and EccBN must be such that the above calculation does not result in a value greater than the size of a page in a 4-KB page NAND.

6.1.5.4.7.2 Metadata

The number of bytes used for the metadata is specified in the FCB. The metadata for the BCH encoded pages is placed at the beginning of a page. The ROM only cares about the first byte of metadata to swap it with a bad block marker byte in the page data after each page read; it is important to have at least one byte for the metadata bytes field in the FCB data structure.

6.1.5.4.8 IOMUX configuration for NAND

The following table shows the RawNAND IOMUX pin configuration.

Table 6-21. NAND IOMUX pin configuration

Signal	Pad name
NAND_ALE	NAND_ALE.alt0
NAND_CE0_B	NAND_CE0_B.alt0
NAND_CE1_B	NAND_CE1_B.alt0
NAND_CE2_B	NAND_CE2_B.alt0
NAND_CE3_B	NAND_CE3_B.alt0
NAND_CLE	NAND_CLE.alt0
NAND_DATA00	NAND_DATA00.alt0

Table continues on the next page...

Table 6-21. NAND IOMUX pin configuration (continued)

Signal	Pad name
NAND_DATA01	NAND_DATA01.alt0
NAND_DATA02	NAND_DATA02.alt0
NAND_DATA03	NAND_DATA03.alt0
NAND_DATA04	NAND_DATA04.alt0
NAND_DATA05	NAND_DATA05.alt0
NAND_DATA06	NAND_DATA06.alt0
NAND_DATA07	NAND_DATA07.alt0
NAND_DQS	NAND_DQS.alt0
NAND_RE_B	NAND_RE_B.alt0
NAND_READY_B	NAND_READY_B.alt0
NAND_WE_B	NAND_WE_B.alt0
NAND_WP_B	NAND_WP_B.alt0

6.1.5.5 Expansion device

The ROM supports booting from the MMC/eMMC and SD/eSD compliant devices.

6.1.5.5.1 Expansion device eFUSE configuration

The SD/MMC/eSD/eMMC/SDXC boot can be performed using either USDHC ports, based on the setting of the BOOT_MODE[3:0].

All USDHC ports support the fast boot. See this table for details:

Table 6-22. USDHC boot eFUSE descriptions

Fuse	Config	Definition	Shipped value	Settings
0x470[15:12]	OEM	BOOT_MODE_FUSES Boot device selection	0	0010 - USDHC3 (eMMC boot only, SD3 8-bit) 0011 - USDHC2 (SD boot only, SD2)
490[6]	OEM	Fast boot support	000	0 - Normal boot 1 - Fast boot
490[5:4]	OEM	Bus width	000	00 - 8-bit 01 - 4-bit 10 - 8-bit DDR (MMC 4.4) 11 - 4-bit DDR (MMC 4.4)
490[3:2]	OEM	Speed mode		SD speed selection 00 - Normal/SDR12 01 - High/SDR25

Table continues on the next page...

Table 6-22. USDHC boot eFUSE descriptions (continued)

Fuse	Config	Definition	Shipped value	Settings
				10 - SDR50 11 - SDR104 MMC speed selection 00 - Normal 01 - High
490[1]	OEM	USDHC IO Voltage Selection		USDHC1 IO VOLTAGE SELECTION (for Normal boot mode) 0 - 3.3 V 1 - 1.8 V
470[10:9]	OEM	OVERRIDE_USDHC_BT_SEL_VAL	00	00 - USDHC1 SD 01 - USDHC1 eMMC 10 - USDHC2 eMMC 11 - USDHC3 SD
490[7]	OEM	SD power cycle enable/ eMMC reset enable	0	SD power cycle/eMMC reset 0 - Disabled 1 - Enabled
0x490[14:8]	OEM	SD/MMC DLL DLY config	0	Delay target for USDHC DLL, it is applied to the slave mode target delay or overrides the mode target delay, depending on the DLL override fuse bit value.
0x490[15]	OEM	USDHC DLL override enabled	0	0 - No override 1 - Override
0x490[16]	OEM	USDHC DLL enabled	0	0 - Disable the DLL for SD/eMMC 1 - Enable the DLL for SD/eMMC
0x490[18]	OEM	USDHC_IOMUX_SION_BIT_ENABLE	0	0 - Disable 1 - Enable
0x490[21]	OEM	Issue pre-idle command enabled (for eMMC4.4)	0	0 - Enable 1 - Disable
0x490[23]	OEM	Disable SDMMC manufacture mode	0	0 - Enable 1 - Disable
0X490[31:24]	OEM	USDHC pad setting override	0	Override pad settings
0x4A0[0]	OEM	Fast boot acknowledge enable	0	0 - Boot Ack disabled 1 - Boot Ack enabled
0x4A0[2]	OEM	uSDHC power-off polarity selection	0	0 - Low 1 - High
0x4A0[3]	OEM	uSDHC power cycle delay selection	0	0 - 5 ms 1 - 2.5 ms
0x4A0[5:4]	OEM	uSDHC power cycle interval	0	00 - 20 ms 01 - 10 ms

Table 6-22. USDHC boot eFUSE descriptions

Fuse	Config	Definition	Shipped value	Settings
				10 - 5 ms 11 - 2.5 ms

The boot code supports these standards:

- MMCv4.4 or less
- eMMCv5.1 or less
- SDv2.0 or less
- eSDv2.10 rev-0.9, with or without FAST_BOOT
- SDXCv3.0

The MMC/SD/eSD/SDXC/eMMC can be connected to any of the USDHC blocks and can be booted by copying 4 KB of data from the MMC/SD/eSD/eMMC device to the internal RAM. After checking the Image Vector Table header value (0xD1) from program image, the ROM code performs a DCD check. After a successful DCD extraction, the ROM code extracts from the Boot Data Structure the destination pointer and length of image to be copied to the RAM device from where the code execution occurs.

The maximum image size to load into the SD/MMC boot is 32 MB. This is due to a limited number of uSDHC ADMA Buffer Descriptors allocated by the ROM.

NOTE

The initial 4 KB of the program image must contain the IVT, DCD, and the Boot Data structures.

Table 6-23. SD/MMC frequencies

	SD	MMC	MMC (DDR mode)
Identification (KHz)	347.22		
Normal-speed mode (MHz)	25	20	25
High-speed mode (MHz)	50	40	50
UHSI SDR50 (MHz)	100		
UHSI SDR104 (MHz)	200		

NOTE

The boot ROM code reads the application image length and the application destination pointer from the image.

6.1.5.5.2 MMC and eMMC boot

This table provides the MMC and eMMC boot details.

Table 6-24. MMC and eMMC boot details

Normal boot mode	During the initialization (normal boot mode), the MMC frequency is set to 347.22 KHz. When the MMC card enters the identification portion of the initialization, the voltage validation is performed, and the ROM boot code checks the high-voltage settings and the card capacity. The ROM boot code supports both the high-capacity and low-capacity MMC/eMMC cards. After the initialization phase is complete, the ROM boot code switches to a higher frequency (20 MHz in the normal boot mode or 40 MHz in the high-speed mode). The eMMC is also interfaced via the USDHC and follows the same flow as the MMC. The boot partition can be selected for an MMC4.x card after the card initialization is complete. The ROM code reads the BOOT_PARTITION_ENABLE field in the Ext_CSD[179] to get the boot partition to be set. If there is no boot partition mentioned in the BOOT_PARTITION_ENABLE field or the user partition was mentioned, the ROM boots from the user partition.
eMMC4.3 or eMMC4.4 device supporting special boot mode	If using an eMMC4.3 or eMMC4.4 device that supports the special boot mode, it can be initiated by pulling the CMD line low. If the BOOT ACK is enabled, the eMMC4.3/eMMC4.4 device sends the BOOT ACK via the DATA lines and the ROM can read the BOOT ACK [S010E] to identify the eMMC4.3/eMMC4.4 device. If the BOOT ACK is enabled, the ROM waits 50 ms to get the BOOT ACK. If BOOT ACK is disabled ROM waits 1 second for data. If the BOOT ACK or data was received, the eMMC4.3/eMMC4.4 is booted in the "boot mode", otherwise the eMMC4.3/eMMC4.4 boots as a normal MMC card from the selected boot partition. This boot mode can be selected by the 0x490[6] (fast boot) fuse. The BOOT ACK is selected by the 0x4A0[0] fuse.
eMMC4.4 device	If using the eMMC4.4 device, the Double Data Rate (DDR) mode can be used. This mode can be selected by the 0x490[5:4] (bus width) fuse.

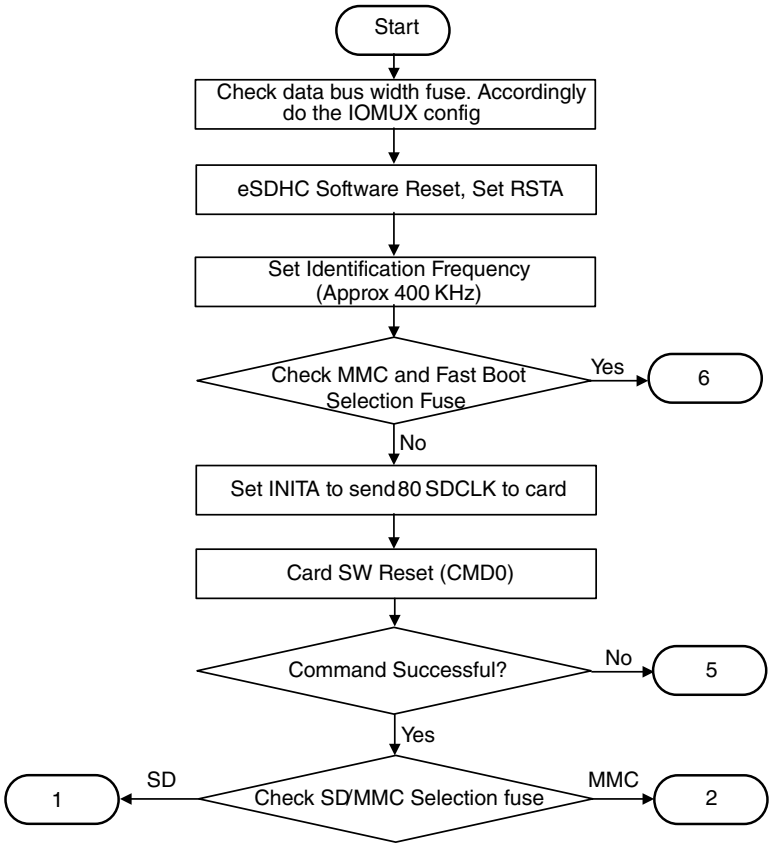


Figure 6-13. Expansion device boot flow (1 of 6)

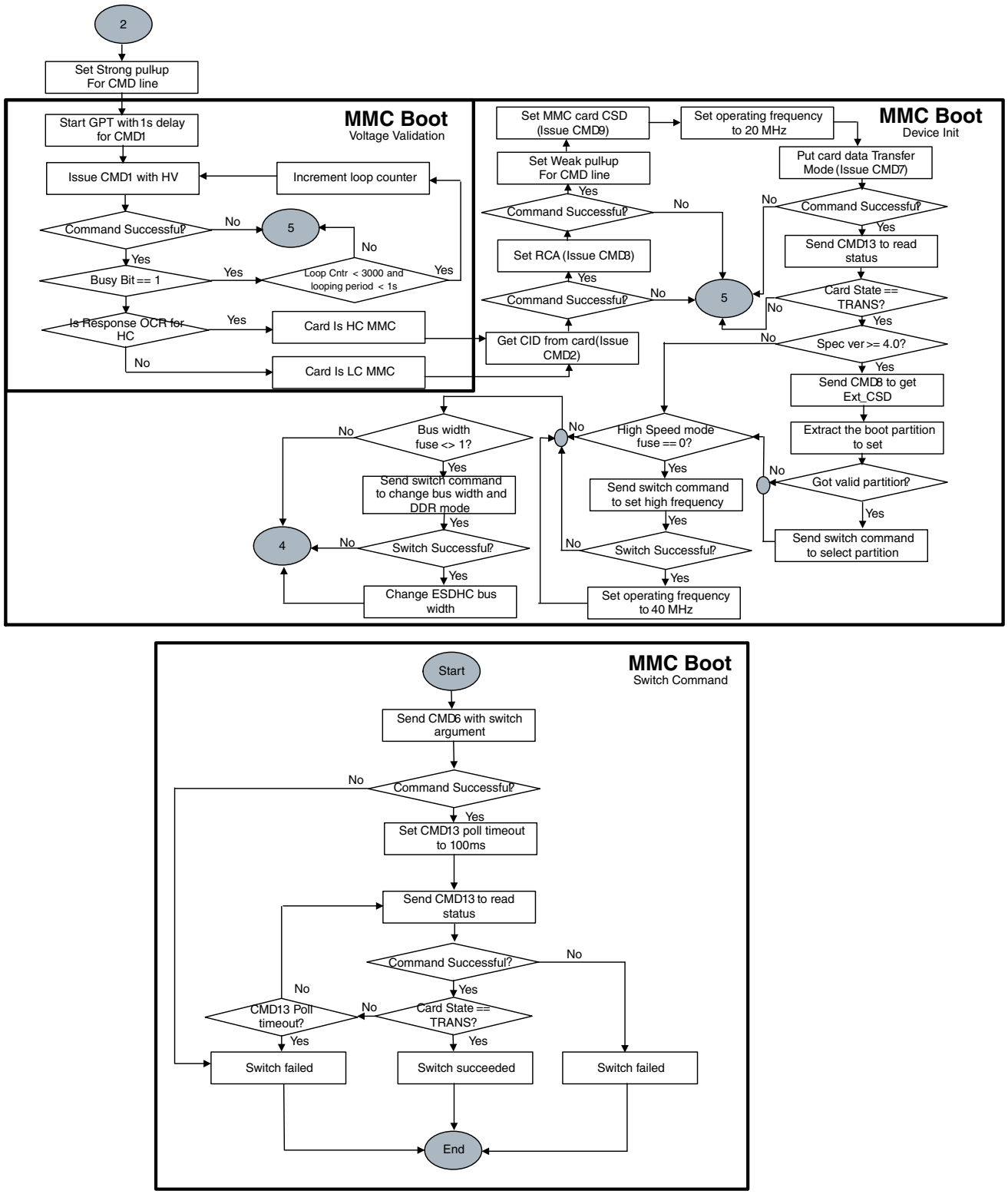


Figure 6-14. Expansion device (MMC) boot flow (2 of 6)

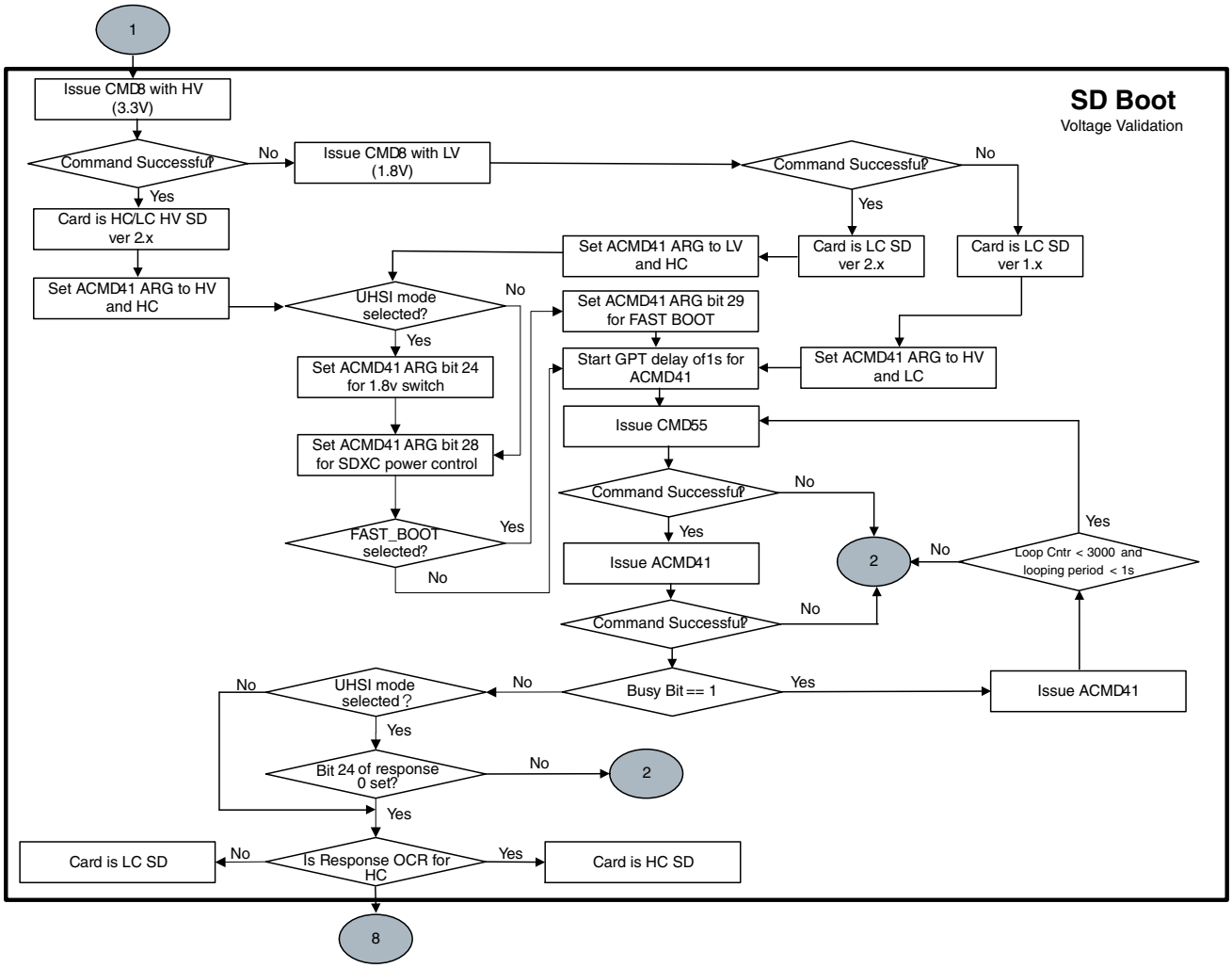


Figure 6-15. Expansion device (SD/eSD/SDXC) boot flow (3 of 6) part 1

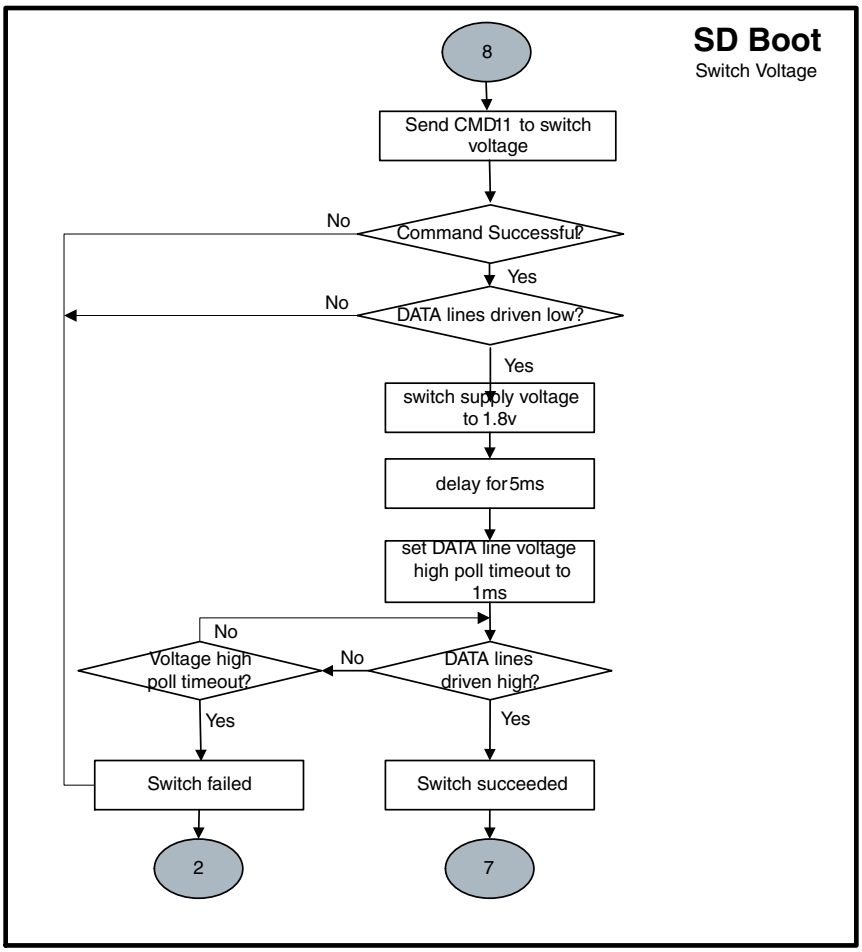


Figure 6-16. Expansion device (SD/eSD/SDXC) boot flow (3 of 6) part 2

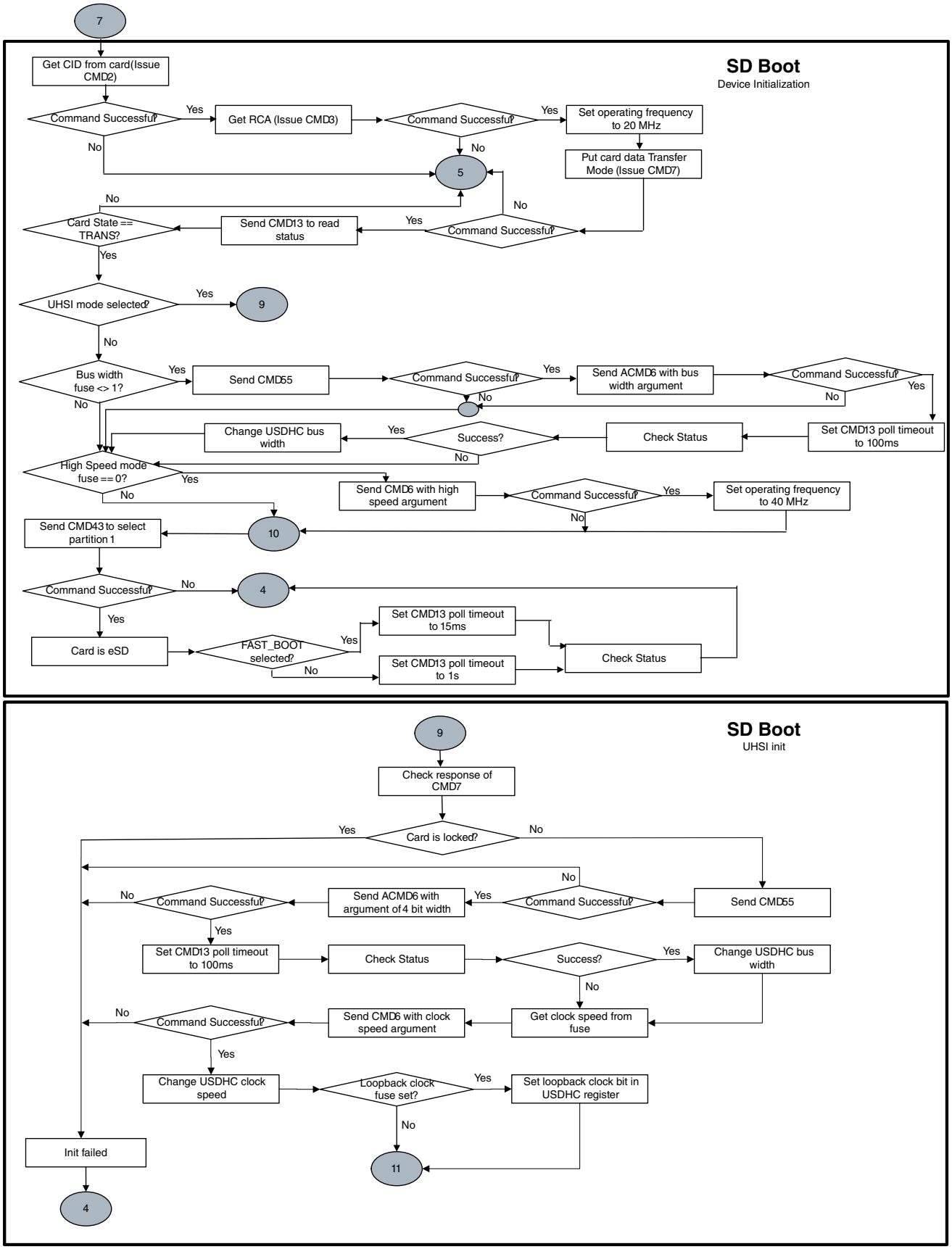


Figure 6-17. Expansion device (MMCSD/eSD/SDXC) boot flow (4 of 6)

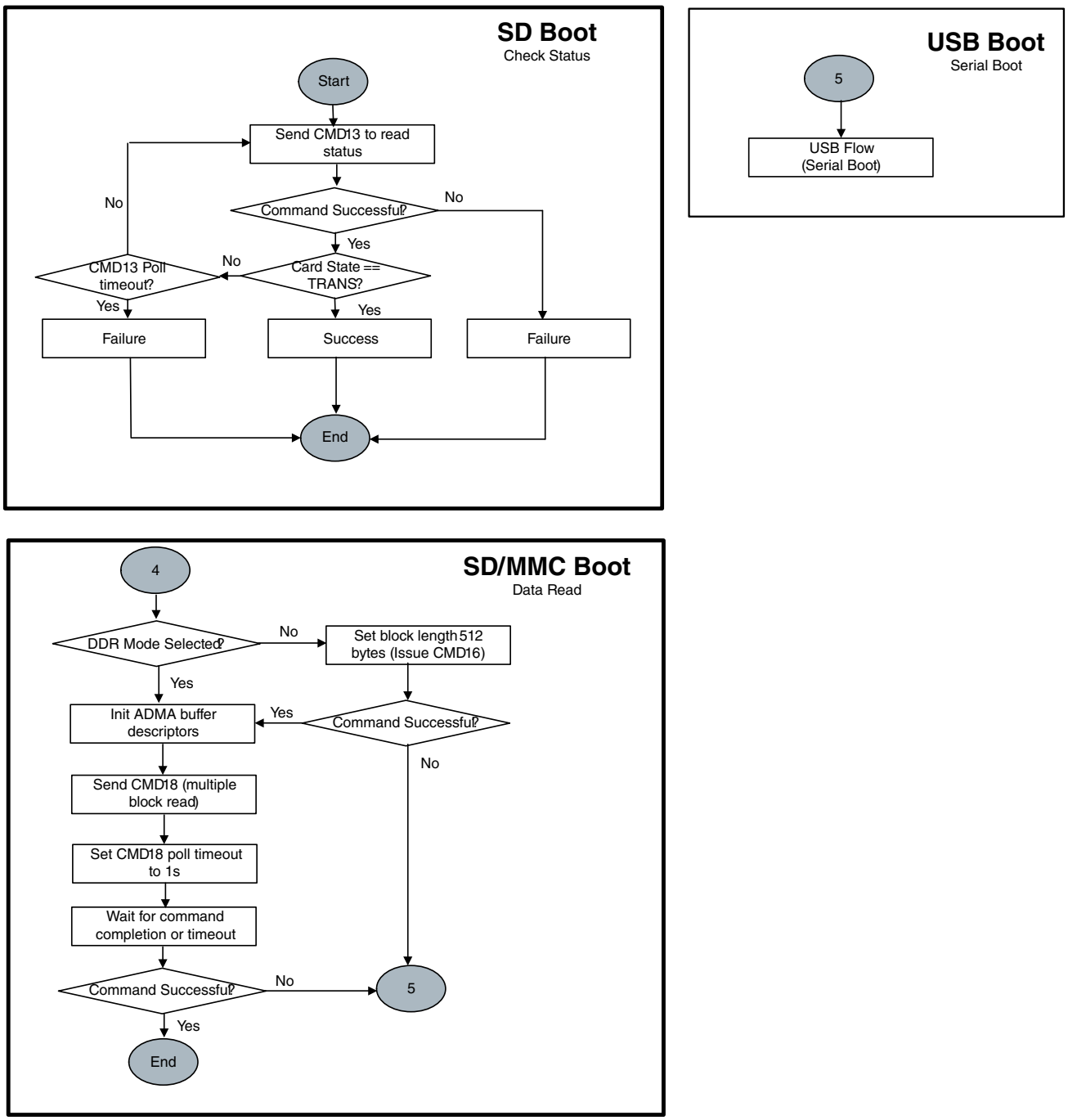


Figure 6-18. Expansion device (SD/eSD) boot flow (5 of 6)

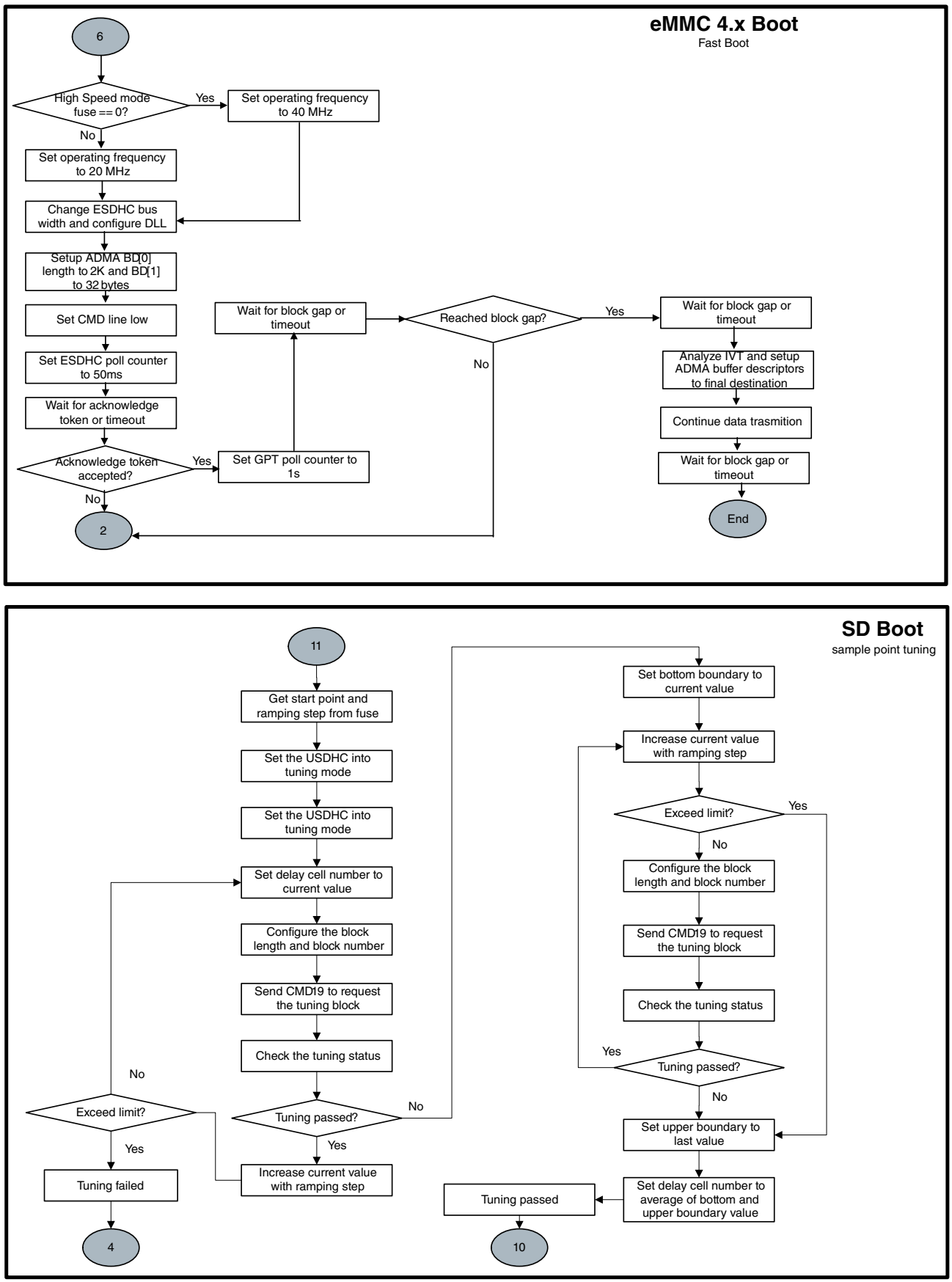


Figure 6-19. Expansion device boot flow (6 of 6)

6.1.5.5.3 SD, eSD, and SDXC

After the normal boot mode initialization begins, the SD/eSD/SDXC frequency is set to 347.22 kHz. During the identification phase, the SD/eSD/SDXC card voltage validation is performed. During the voltage validation, the boot code first checks with the high-voltage settings; if that fails, it checks with the low-voltage settings.

The capacity of the card is also checked. The boot code supports the high-capacity and low-capacity SD/eSD/SDXC cards after the voltage validation card initialization is done.

During the card initialization, the ROM boot code attempts to set the boot partition for all SD, eSD, and SDXC devices. If this fails, the boot code assumes that the card is a normal SD or SDXC card. If it does not fail, the boot code assumes it is an eSD card. After the initialization phase is over, the boot code switches to a higher frequency (25 MHz in the normal-speed mode or 50 MHz in the high-speed mode).

For the UHSI cards, the clock speed fuses can be set to SDR50 or SDR104 on USDHC1 ports. This enables the voltage switch process to set the signaling voltage to 1.8 V during the voltage validation. The bus width is fixed at a 4-bit width and a sampling point tuning process is needed to calibrate the number of the delay cells. The SD clock speed can be selected by the 0x490[3:2].

The UHSI calibration start value (MMC_DLL_DLY[6:0]) and the step value can be set to optimize the sample point tuning process.

If the SD Power Cycle Enable eFuse is 1, the ROM sets the SD_RST pad low, waits for 5 ms, and then sets the SD_RST pad high. If the SD_RST pad is connected to the SD power supply enable logic on board, it enables the power cycle of the SD card. This may be crucial in case the SD logic is in the 1.8 V states and must be reset to the 3.3 V states.

6.1.5.5.4 IOMUX configuration for SD/MMC

Table 6-25. SD/MMC IOMUX pin configuration

Signal	USDHC1	USDHC2	USDHC3
CD_B	GPIO1_IO06.alt5	SD2_CD_B.alt0	I2C2_SCL.alt2
CLK	SD1_CLK.alt0	SD2_CLK.alt0	NAND_WE_B.alt2
CMD	SD1_CMD.alt0	SD2_CMD.alt0	NAND_WP_B.alt2
DATA0	SD1_DATA0.alt0	SD2_DATA0.alt0	NAND_DATA04.alt2
DATA1	SD1_DATA1.alt0	SD2_DATA1.alt0	NAND_DATA05.alt2
DATA2	SD1_DATA2.alt0	SD2_DATA2.alt0	NAND_DATA06.alt2
DATA3	SD1_DATA3.alt0	SD2_DATA3.alt0	NAND_DATA07.alt2
DATA4	SD1_DATA4.alt0	-	NAND_RE_B.alt2
DATA5	SD1_DATA5.alt0	-	NAND_CE2_B.alt2

Table continues on the next page...

Table 6-25. SD/MMC IOMUX pin configuration (continued)

Signal	USDHC1	USDHC2	USDHC3
DATA6	SD1_DATA6.alt0	-	NAND_CE3_B.alt2
DATA7	SD1_DATA7.alt0	-	NAND_CLE.alt2
RESET_B	SD1_RESET_B.alt0	SD2_RESET_B.alt0	GPIO1_IO09.alt4
VSELECT	GPIO1_IO03.alt1	GPIO1_IO04.alt1	GPIO1_IO11.alt4

6.1.5.6 Serial NOR through SPI

The chip supports booting from serial memory devices, such as EEPROM and serial flash, using the SPI.

These ports are available for serial boot: eCSPI (eCSPI1, eCSPI2, eCSPI3) interfaces.

6.1.5.6.1 Serial(SPI) NOR eFUSE configuration

The boot ROM code determines the type of device using the following parameters, either provided by the eFUSE settings or sampled on the BOOT_MODE[3:0] pins, during boot.

See this table for details:

Table 6-26. Serial(SPI) NOR boot eFUSE descriptions

Fuse	Config	Definition	Shipped value	Settings
480[31:29]	OEM	ECSPI port selection	000	000 - eCSPI1 001 - eCSPI2 010 - eCSPI3
480[28]	OEM	SPI addressing	0	0 - 3 B (24-bit) 1 - 2 B (16-bit)
0x480[25]	OEM	Recovery boot enable	0	0 – Disabled 1 – Enabled
480[27:26]	OEM	CS selection (SPI only)	00	00 – CS#0

The ECPSI-1/ECPSI-2/ECPSI-3 block can be used as a boot device using the ECSPI interface for the serial(SPI) NOR boot. The SPI interface is configured to operate at 12.5 MHz for 3-byte addressing devices and at 3.125 MHz for 2-byte addressing devices.

The boot ROM copies 4 KB of data from the serial ROM device to the internal RAM. After checking the Image Vector Table header value (0xD1) from the program image, the ROM code performs a DCD check. After a successful DCD extraction, the ROM code extracts the destination pointer and length of image from the Boot Data Structure to be copied to the RAM device from where the code execution occurs.

NOTE

The Initial 4 KB of program image must contain the IVT, DCD, and the Boot Data Structures.

6.1.5.6.2 ECSPI boot

The Enhanced Configurable SPI (ECSPI) interface is configured in the master mode and the EEPROM device is connected to the ECSPI interface as a slave.

The boot ROM code copies 4 KB of data from the EEPROM device to the internal RAM. If the DCD verification is successful, the ROM code copies the initial 4 KB of data, as well as the rest of the image extracted from the application image, directly to the application destination. The ECSPI can read data from the EEPROM using 2- or 3-byte addressing. Its burst length is 32 B.

When using the SPI as a boot device, the chip supports booting from both the serial EEPROM and serial flash devices. The boot code determines which device is being used by reading the appropriate eFUSE/I/O values at the boot (see Table 6-26 for details).

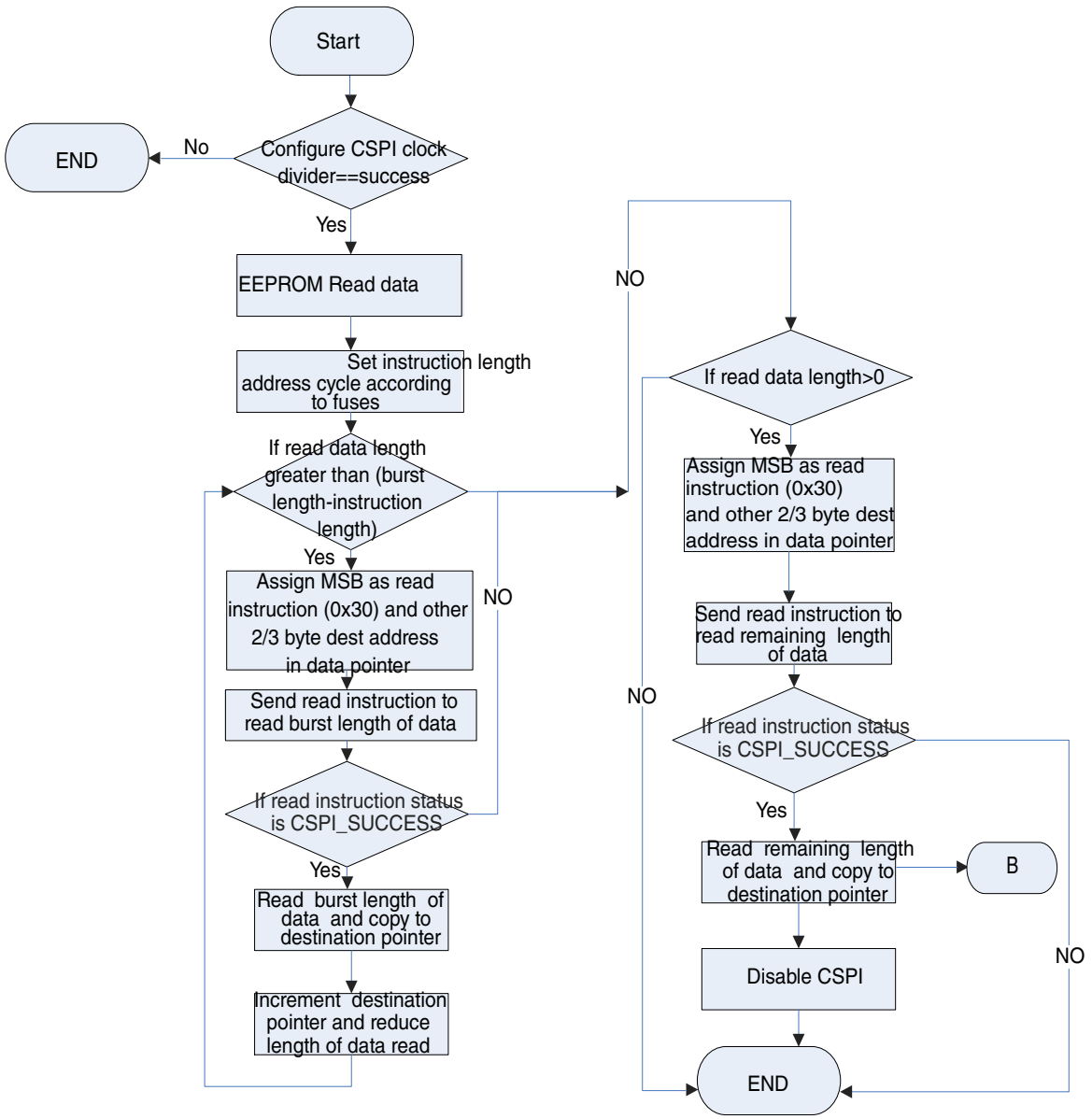


Figure 6-20. CSPI flow chart

6.1.5.6.2.1 ECSPI IOMUX pin configuration

The contacts assigned to the signals used by the CSPI blocks are shown in this table:

Table 6-27. ECSPI IOMUX pin configuration

Signal	eCSPI1	eCSPI2	eCSPI3
MISO	ECSPI1_MISO.alt0	ECSPI2_MISO.alt0	UART2_RXD.alt1

Table continues on the next page...

Table 6-27. ECSPI IOMUX pin configuration (continued)

MOSI	ECSPI1_MOSI.alt0	ECSPI2_MOSI.alt0	UART1_TXD.alt1
SCLK	ECSPI1_SCLK.alt0	ECSPI2_SCLK.alt0	UART1_RXD.alt1
SS0	ECSPI1_SS0.alt0	ECSPI2_SS0.alt0	UART2_TXD.alt1

6.1.6 Boot image

After the initial image (4KB) is loaded, ROM will check the image header's validity. The header must meet the following requirements:

- Reserved1 and Reserved2 of IVT must be 0.
- Device Configuration Data (DCD) of IVT must be 0.
- TAG of IVT header must be 0xD120004x
- SPL image's entry, boot_data, self, CSF in IVT need in IRAM_FREE_SPACE or FlexSPI space
- The address of the image's entry, boot data and CSF (if CSF is not NULL) must be out of IVT header range
- Image's boot_data must be in initial 4k image
- Image target address should be 4 bytes aligned, and the image need in IRAM_FREE_SPACE range or FlexSPI space. No plugin support.
- The first 4KB of IRAM_FREE_SPACE is used as an initial image buffer. If image target address is not equal to the base address of IRAM_FREE_SPACE, it should be out of the first 4KB range of IRAM_FREE_SPACE.

NOTE

The addresses referred in IVT header, boot data or CSF is from the boot core's view (A53 core).

6.1.6.1 Primary Image boot

Boot from the device selected by the boot configuration pin or eFuse (Primary Device). The Primary Image (the image at the beginning of the boot device with the image offset) will be selected.

6.1.6.2 Secondary Image Boot

If booting from the device selected by the boot configuration pin or eFuse (Primary Device) fails, ROM will try to boot from the Secondary Image, the offset is relative to the beginning of the boot device (which is provisioned in the fuse IMG_CNTN_SET1_OFFSET).

The fuse IMG_CNTN_SET1_OFFSET (0x490[22:19]) is defined as follows:

- Secondary boot is disabled if fuse value is bigger than 10, n = fuse value bigger than 10.
- n == 0: Offset = 4MB
- n == 2: Offset = 1MB
- Others & n <= 10 : Offset = 1MB*2^n
- For FlexSPI NOR boot, the valid values are: 0, 1, 2, 3, 4, 5, 6, and 7.

6.1.6.3 Primary image offset and IVT offset

ROM supports one single image for all boot devices. The image offset and IVT offset for the supported boot devices is provided below.

Table 6-28. Primary image offset and IVT offset details

	Primary Image Offset	IVT Offset
SD	32KB (for GPT)	0
EMMC	0, if the image is in boot partition and 32K if it is in user partition	0
NAND	0	0
FlexSPI NOR	4KB	0
SPI	0	0
FlexSPI NAND	0	0

NOTE

- Primary Image Offset - The image offset on device against to the beginning of boot device.
- IVT Offset - The IVT header offset in boot image against to the beginning of boot image.

6.1.6.4 Typical image placement in boot device

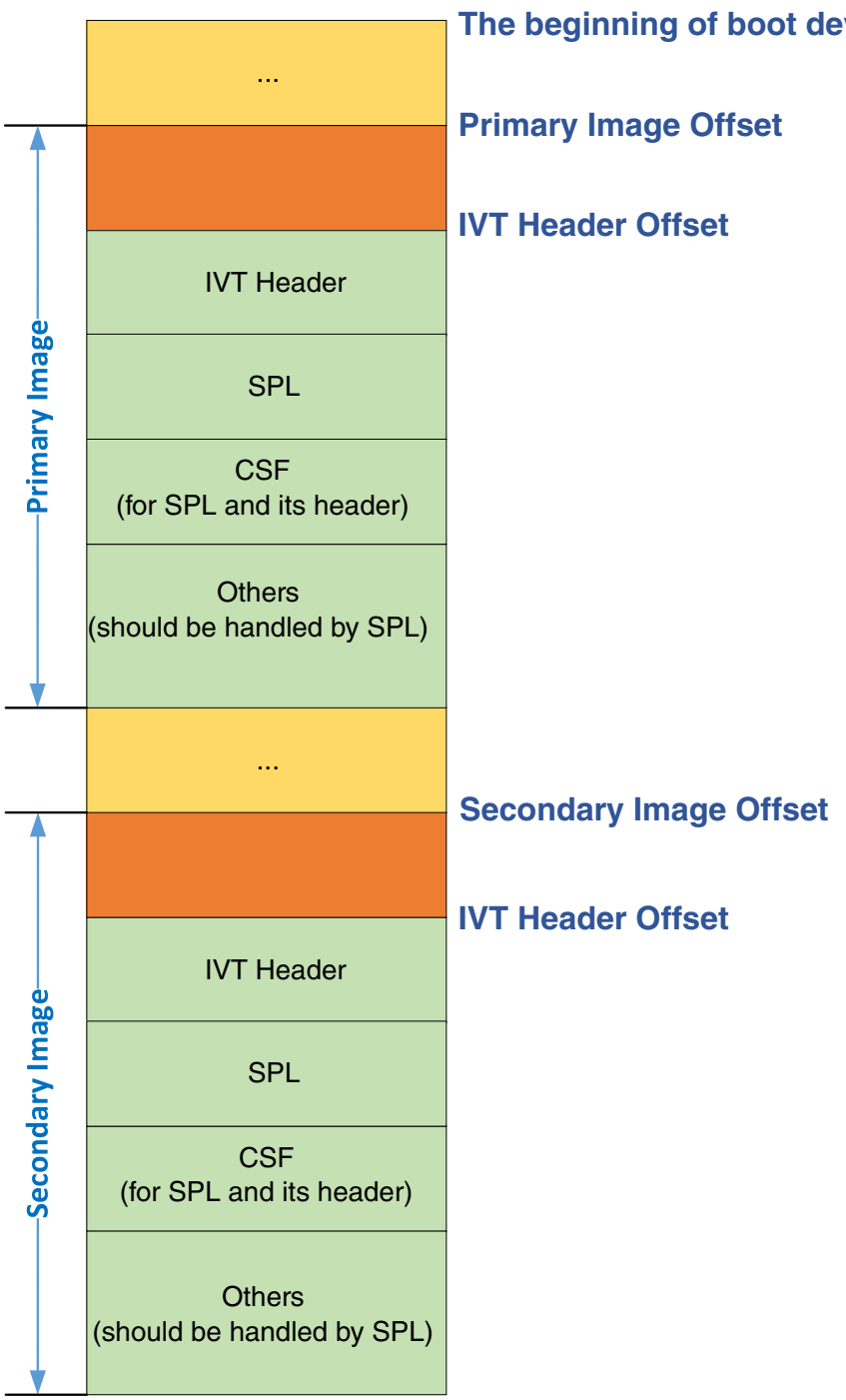


Figure 6-21. Typical image placement

- For the primary image offset and IVT offset details, please refer to Primary image offset and IVT offset.
- The secondary image offset is specified by fuses, please refer to the fuse map chapter.

6.1.7 USB boot

ROM supports both of the USB ports on the chip for boot purposes. Only one port can be selected as the boot connection. The port that is active first wins the selection.

USB boot in ROM code works as a HID device. ROM takes USB as a normal boot device. The difference from the SD card here is that the USB is a 'stream' device, which is the same as eMMC fast boot. ROM must 'read' it one by one byte, and it can not specify the offset for the reading. This means that the image for USB boot must be continuous.

USB Serial Download boot can be disabled by fuse.

The USB HID device VID/PID and strings are listed in the following table:

Table 6-29. VID/PID and strings for USB HID device

Descriptor	Value
VID	0x1FC9
PID	0x0146
bcdDevice	0x0001
String Descriptor0	bLength : 0x04 (4 bytes) bDescriptorType : 0x03 (String Descriptor) Language ID[0] : 0x0409 (English - United States)
String Descriptor1	bLength : 0x3A (58 bytes) bDescriptorType : 0x03 (String Descriptor) Language 0x0409 : "NXP SemiConductor Inc "
String Descriptor2	bLength : 0x1E (30 bytes) bDescriptorType : 0x03 (String Descriptor) Language 0x0409 : "SE Blank 865"
String Descriptor4	bLength : 0x22 (34 bytes) bDescriptorType : 0x03 (String Descriptor) Language 0x0409 : *This value depends on chip uuid.*

6.1.8 Low-power boot

The ROM supports the low-power boot. If the LPB_BOOT fuses are blown, the chip checks if there is a low-power condition via the pad. If there is a low-power boot condition, the ROM applies division factors on the Arm, DDR, AXI, and AHB root

clocks based on the LPB_BOOT fuses value (see the table below). The polarity of the low-power boot condition on the pad is set by the BT_LPB_POLARITY fuse (see the following figure).

Table 6-30. Low-power boot frequencies

LPB_BOOT	Boot Frequencies=0	Boot Frequencies=1
00	ARM_A53_CLK_ROOT= 1000 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 400 MHz	ARM_A53_CLK_ROOT= 500 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 200 MHz
01	ARM_A53_CLK_ROOT= 1000 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 400 MHz	ARM_A53_CLK_ROOT= 500 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 200 MHz
10	ARM_A53_CLK_ROOT= 500 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 200 MHz	ARM_A53_CLK_ROOT= 250 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 100 MHz
11	ARM_A53_CLK_ROOT= 250 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 100 MHz	ARM_A53_CLK_ROOT= 125 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 50 MHz

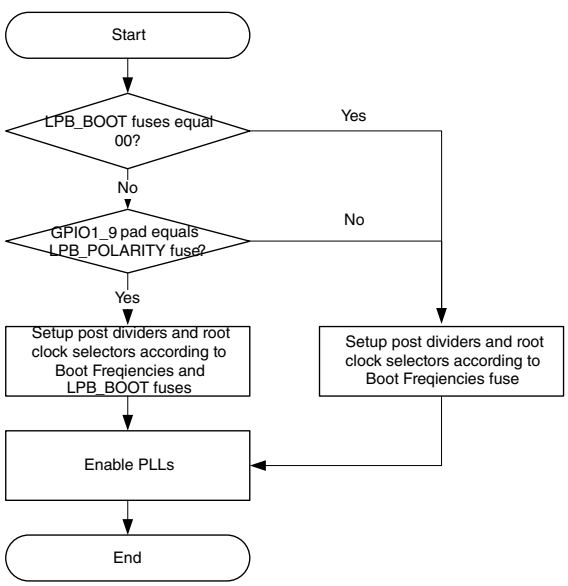


Figure 6-22. Low-power boot flow

6.1.9 SD/MMC manufacture mode

When the internal boot and recover boot (if enabled) failed, the boot goes to the SD/MMC manufacture mode before the serial download mode. In the manufacture mode, one bit bus width is used despite of the fuse setting.

In the manufacture mode, the SD or MMC card will be scanned on the uSDHC2. If a card is detected and a valid boot image is found in the card, the boot image is loaded and executed. Pad of SD1_CD is used to detect whether a card is inserted or not.

By default, the SD/MMC manufacture mode is enabled. Blow the fuse of the RECOVERY_SDMMC_BOOT_DIS to disable it.

NOTE

A secondary boot is not supported in the SD/MMC manufacture mode.

6.1.9.1 Using manufacture mode / serial download mode with eMMC

Manufacture mode is intended to allow a system to boot from a SD/MMC card on a board with unprogrammed boot media or to upgrade the image on a boot device. For manufacture mode, the boot ROM assumes if there is a SD/MMC card present indicated by the uSDHC card detect (CD) signal pulled low, then there is a valid image on the card.

If an unprogrammed eMMC device is connected to the uSDHC port(s) on which manufacture mode is supported and the CD signal is low, the ROM will attempt manufacturing mode with the eMMC device, which does not contain a valid image. The ROM loads the invalid image, and this will cause the ROM code to fail to enter serial download mode, resulting in a reset of the system by the ROM. To enter serial download mode in this case, the CD pin should be pulled up so the ROM does not detect a eMMC device present and will bypass the manufacture mode to enter serial download mode.

6.1.10 High-Assurance Boot (HAB)

The High Assurance Boot (HAB) component of the ROM protects against the potential threat of attackers modifying the areas of code or data in the programmable memory to make it behave in an incorrect manner. The HAB also prevents the attempts to gain access to features which must not be available.

The integration of the HAB feature with the ROM code ensures that the chip does not enter an operational state if the existing hardware security blocks detected a condition that may be a security threat or if the areas of memory deemed to be important were modified. The HAB uses the RSA digital signatures to enforce these policies.

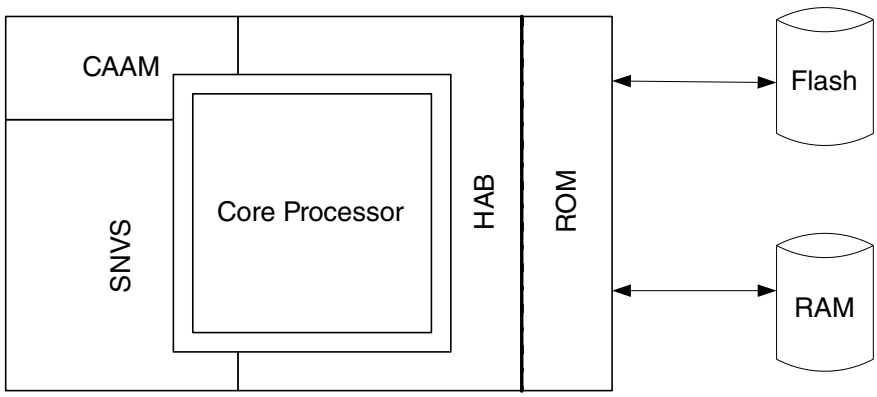


Figure 6-23. Secure boot components

The figure above illustrates the components used during a secure boot using HAB. The HAB interfaces with the SNVS to make sure that the system security state is as expected. The HAB also uses the CAAM hardware block to accelerate the SHA-256 message digest operations performed during the signature verifications and AES-128 operations for the encrypted boot operations. The HAB also includes a software implementation of SHA-256 for cases where a hardware accelerator can't be used. The RSA key sizes supported are 1024, 2048, 3072, and 4096 bits. The RSA signature verification operations are performed by a software implementation contained in the HAB library. The main features supported by the HAB are:

- X.509 public key certificate support
- CMS signature format support
- Proprietary encrypted boot support. Note that the encrypted boot depends on the CAAM hardware module. When the CAAM is disabled (when the EXPORT_CONTROL fuse is blown), the encrypted boot is not available.

NOTE

NXP provides the reference Code Signing Tool (CST) for key generation, certificate generation, and code signing for use with the HAB library. The CST can be found by searching for "IMX_CST_TOOL_NEW" at http://www.nxp.com.

NOTE

For further details on using the secure boot feature using HAB, refer to *Secure Boot on i.MX Series (**AN4581**)*.

6.1.10.1 HAB API vector table addresses

For devices that perform a secure boot, the HAB library may be called by the boot stages that execute after the ROM code.

NOTE

For additional information on the secure boot including the HAB API, refer to *HABv4 RVT Guidelines and Recommendations (AN12263)*.

6.1.11 Boot information for software

To address the requirement that the boot image may need to get the basic boot information when getting out of the boot process, the boot software information (Boot_SW_Info) is provisioned by the ROM.

The software must read the ROM address 0x9e8 to get the base address of the Boot_SW_Info data structure, and parse the Boot_SW_Info content to get the boot information.

Table 6-31. Boot_SW_Info structure

Offset	Byte3	Byte2	Byte1	Byte0
0x0	Reserved	Boot Device Type	Boot Device Instance	Reserved
0x4	Arm core frequency (in Hz)			
0x8	AXI bus frequency (in Hz)			
0xC	DDR frequency (in Hz)			
0x10	GPT1 input clock frequency (in Hz)			
0x14	Reserved			
0x18				
0x1C				

NOTE

The boot ROM sets the GPT1 in a free-running mode with a 32-kHz input clock.

Boot device type mapping:

- 0x1 - SD card or eSD chip

- 0x2 - MMC card or eMMC chip
- 0x3 - NAND chip
- 0x4 - FLEXSPI NOR
- 0x6 - ECSPI EEPROM
- 0x8 - FLEXSPI NAND

Boot device instance: The instance index of the boot device, starting from 0.

6.2 Fusemap

6.2.1 Boot Fusemap

The boot mode utilizes 4 dedicated BOOT_MODE (BOOT_MODE[3:0]) pins. The following section details the various modes and selection of the required boot devices.

Table 6-32. Boot Device Select

Boot Device Select	BOOT CODE	BOOT_MODE [3]	BOOT_MODE [2]	BOOT_MODE [1]	BOOT_MODE [0]
Boot from internal fuses	0x00	0	0	0	0
USB Serial Download	0x01	0	0	0	1
USDHC3 (eMMC boot only, SD3 8-bit)	0x02	0	0	1	0
USDHC2 (SD boot only, SD2)	0x03	0	0	1	1
NAND 8-bit single device, 256 pages	0x04	0	1	0	0
NAND 8-bit single device, 512 pages	0x05	0	1	0	1
FlexSPI 3B Read	0x06	0	1	1	0
FlexSPI Hyperflash 3.3V	0x07	0	1	1	1
eCSPI Boot	0x08	1	0	0	0
FLEXSPI Serial NAND 2k page	0x0A	1	0	1	0
FLEXSPI Serial NAND 4k page	0x0B	1	0	1	1

NOTE

Fuses marked as “Reserved” are reserved for NXP internal (and future) use only. Customers should not attempt to burn these, as the IC behavior may be unpredictable. The reserved fuses can be read as either 0 or 1.

Table 6-33. Boot Fusemap

Addr	7	6	5	4	3	2	1	0
0x470[7:0]	OVERRIDE_NAND_PG_PER_BLK_VAL		OVERRIDE_FLEXSPI_BT_SEL	OVERRIDE_FLEXSPI_BT_SEL_VAL		FLEXSPI_AUTO_PROBE_EN	FLEXSPI_AUTO_PROBE_TYPE	

Table continues on the next page...

Table 6-33. Boot Fusemap (continued)

Addr	7	6	5	4	3	2	1	0
	RAWNAND: 00 - 32 pages 01 - 64 pages 10 - 128 pages 11 - 32 pages FlexSPI NAND: 00 - 64 pages 01 - 128 pages 10 - 256 pages 11 - 32 pages		0 - Do not override 1 - Override	00 - FlexSPI (Hyperflash 1.8V) 01 - FlexSPI (Flash with 4B READ (1x13 default supported)) 10 - Default Octal mode (Micron) 11 - Default Octal mode (Macronix)		0 - Disable 1 - Enable	00 - QuadSPI NOR 01 - MacronixOctal 10 - MicronOctal 11 - AdestoOctal	
0x470[15:8]	BOOT_MODE_FUSES Boot Rom will retrieve boot mode from these fuses instead of BOOT_MODE pins if: (BOOT_MODE_PINS=0 or FORCE_BT_FROM_FUSE = 1) and BT_FUSE_SEL=1				OVERRIDE_USDHC_BT_SEL 0 - Do not override 1 - Override	OVERRIDE_USDHC_BT_SEL_VAL 00 - uSDHC1 SD 01 - uSDHC1 eMMC 10 - uSDHC2 eMMC 11 - uSDHC3 SD		OVERRIDE_NAND_PG_PER_BLK 0 - Do not override 1 - Override
0x480[7:0]	Reserved	FLEXSPI_DUMMY_CYCLE_SEL 0 – Dummy cycle is auto-probed Other values – Actual dummy cycles for Read command				FLEXSPI_FEQ_SEL 000 - 100 MHz 001 - 133 MHz 010 - 166 MHz 011 - 200 MHz 100 - 80 MHz 101 - 20 MHz		
0x480[15:8]	BT_LPB (Core/DDR/Bus) '00'/'01' - LPB Disable '10' - Div by 2 '11' - Div by 4		BT_LPB_POLARITY (GPIO polarity)	ICACHE_DIS L1 I-Cache DISABLE 0 - L1 I Cache is enabled by the ROM during the boot. 1 - L1 I Cache is disabled by the ROM during the boot.	Reserved	WDOG_EN '0' - Disabled '1' - Enabled	BT_FREQ_SEL (Arm/DDR) 0 - 800 / 800 MHz 1 - 400 / 400 MHz	DCACHE_DIS Disable L1 and L2 D-Cache
0x480[23:16]	NOC_ID_REMAP_BYPASS	Reserved	SDP_DISABLE Disable USB serial download	FORCE_BT_FROM_FUSE	FLEXSPI_HOLD_TIME_SEL 00 - 500us 01 - 1ms 10 - 3ms 11 - 10ms		WDOG_TIMEOUT_SELECT 00 - 2.0s 01 - 1.5s 10 - 1.0s 11 - 0.5s	

Table continues on the next page...

Table 6-33. Boot Fusemap (continued)

Addr	7	6	5	4	3	2	1	0
				Boot from programmed fuses, not Boot Mode Pins				
0x480[31:24]	ECSPI_PORT_SEL 000 - eCSPI1 001 - eCSPI2 010 - eCSPI3			ECSPI_ADDR_SEL 0 - 3-bytes (24-bit) 1 - 2-bytes (16-bit)	ECSPI_CS_SEL (SPI only) 00 - CS#0 (default) 01 - CS#1 10 - CS#2 11 - CS#3		RECOVER_ECSPI_BOOT_EN '0' - Disabled '1' - Enabled	DCACHE_BYPASS_DIS
0x490[7:0]	USDHC_PWR_EN 0 - No power cycle 1 - Enabled	EMMC_FAST_BT 0 - Regular 1 - Fast Boot	SDMMC_BUS_WIDTH 00 - 8-bit 01 - 4-bit 10 - 8-bit DDR (MMC 4.4) 11 - 4-bit DDR (MMC 4.4)		SD_SPEED: 00 - Normal/SDR12 01 - High/SDR25 10 - SDR50 11 - SDR104 EMMC_SPEED: 00 - Normal 01 - High		USDHC_VOL_SEL For Normal Boot Mode IO Voltage 0 - 3.3V 1 - 1.8V	USDHC_MFG_VOL_SEL For Mfg Mode IO Voltage 0 - 3.3V 1 - 1.8V
0x490[15:8]	USDHC_DLL_SEL 0 - DLL Slave Mode 1 - DLL Override Mode	SDMMC_DLL_DLY[6:0] Delay target for USDHC DLL, it is applied to slave mode target delay or override mode target delay depends on DLL Override fuse bit value.						
0x490[23:16]	RECOVERY_SDMMC_BOOT_DIS 0 - Enable 1 - Disable	IMG_CNTN_SET1_OFFSET				USDHC_PAD_SION_EN 0 - Disable 1 - Enable	Reserved	USDHC_DLL_EN 0 - Disable DLL for SD/eMMC 1 - Enable DLL for SD/eMMC
0x490[31:24]	USDHC_OVRD_PAD_SETTING_LOW8 [7:0]							
0x4A0[7:0]	SD_CALI_STEP '00' - 1		USDHC_PWR_INTERVAL 00 - 20ms 01 - 10ms 10 - 5ms 11 - 2.5	USDHC_PWR_DELAY 0 - 5ms 1 - 2.5ms	USDHC_PWR_POLARITY 0 - Low 1 - High		USDHC_OVRD_PAD_SETTING_UP1	EMMC_FAST_BT_ACK 0 - Boot Ack Disabled 1 - Boot Ack Enabled
0x4A0[15:8]	BT_TOGGLE_MODE	NAND_FCB_SERCH_COUNT	NAND_TG_PREAMBLE_RD_LATENCY					NAND_RST_TIME

Table continues on the next page...

Table 6-33. Boot Fusemap (continued)

Addr	7	6	5	4	3	2	1	0
		00 - 2 01 - 2 10 - 4 11 - 8		(Toggle Mode 33MHz Preamble Delay, Read Latency) '000' - 16 GPMICLK cycles. '001' - 1 GPMICLK cycles. '010' - 2 GPMICLK cycles. '011' - 3 GPMICLK cycles. '100' - 4 GPMICLK cycles. '101' - 5 GPMICLK cycles. '110' - 6 GPMICLK cycles. '111' - 7 GPMICLK cycles. '1111'- 15 GPMICLK cycles.				
0x4A0[23:16]	Reserved							
0x4A0[31:24]	NAND_OVERRIDE_PAD_SETTING[7:0]							
0x4B0[7:0]	Reserved	NAND_GPMI_DDR_DLL_VAL (GPMI Read DDR DLL Target Value) 0000 - 7 0001 - 1 0111 - 0 1111 - 15				Reserved	NAND_CS_NUM (Nand Number Of Devices) 00 - 1 01 - 2 10 - 4 11 - Reserved	
0x4B0[15:8]	NAND_READ_RETRY_SEQ_ID[3:0][1] 0000 - Do not use read retry (RR) sequence embedded in ROM 0001 - Micron 20nm RR sequence 0010 - Kioxia A19nm RR sequence 0011 - Kioxia 19nm RR sequence 0100 - SanDisk 19nm RR sequence 0101 - SanDisk 1ynmRR sequence 0110 - SK Hynix 20nm A Die RR sequence 0111 - SK Hynix 26nm RR sequence Others - Reserved				NAND_ROW_ADDR_BYTES 00 - 3 01 - 2 10 - 4 11 - 5		Reserved	
0x4B0[31:16]	Reserved							

1. Testing has been performed on select memory devices. Please contact your NXP representative for more details.

6.2.2 Lock Fusemap

Table 6-34 describes the functions of various lock fuses.

NOTE

Definitions for fuse settings are as follows:

- Unlock - The controlled field can be read, sensed, burned, or overridden in the corresponding OCOTP shadow register.
- Lock - The controlled field cannot be read, burned, or overridden.
- Override Protect (OP) - The controlled field cannot overwrite the corresponding OCOTP shadow register content. The controlled field can sense or burn fuses, and read the corresponding OCOTP shadow register.
- Write Protect (WP) - The controlled field cannot burn a fuse. The controlled field can be sensed (fuse), read or overridden in the corresponding OCOTP shadow register.
- OP + WP - The controlled field cannot burn (fuses) or overwrite the corresponding OCOTP shadow register. The controlled field can only sense (fuses) or read the corresponding OCOTP shadow register.

Table 6-34. Lock Fuses

Addr	7	6	5	4	3	2	1	0
0x400[7:0]	Reserved							
0x400[15:8]	MAC_ADDR_LOCK 1x - OP x1 - WP		USB_ID_LOCK 1x - OP x1 - WP		Reserved			
0x400[23:16]	GP2_LOCK 1x - OP x1 - WP		GP1_LOCK 1x - OP x1 - WP		Reserved			
0x400[31:24]	Reserved							
0x410[7:0]	Reserved		GP4_LOCK 1x – OP x1 – WP		Reserved			
0x410[15:8]	GP8_LOCK 1x – OP x1 – WP		GP7_LOCK 1x – OP x1 – WP		GP6_LOCK 1x – OP x1 – WP		GP5_LOCK 00 – Unlock 10 – OP 01 – WP 11 - WP + OP	
0x410[23:16]	Reserved						GP9_LOCK 1x – OP x1 – WP	
0x410[31:24]	Reserved							

NOTE

The GP5_LOCK eFuse locks the upper 64 bits of the 128-bit UNIQUE_ID (UID) eFuse field. The users must program the GP5_LOCK eFuse to b'11 (WP+OP), thereby locking the upper 64-bit UID eFuse field.

For devices shipped with date codes 2435 and later, the GP5_LOCK has been programmed to b'11 by NXP in production. Only devices with date codes prior to 2435 should be programmed to update the GP5_LOCK eFuse to b'11.

For additional information, contact your NXP Account Manager or Field Representative.

6.2.3 Fusemap Descriptions Table

NOTE

For security related fuses, please refer to the chip's Security Reference Manual.

Table 6-35. Fusemap Descriptions

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
0x400[11:0]	Reserved	12	Reserved	Reserved	Reserved
0x400[13:12]	USB_ID_LOCK	2	Lock for USB_PID and USB_VID fuses.	00 - Unlock 10 - OP 01 - WP 11 - OP + WP	OCOTP
0x400[15:14]	MAC_ADDR_LOCK	2	Lock for MAC_ADDR fuses.	00 - Unlock 10 - OP 01 - WP 11 - OP + WP	OCOTP
0x400[19:16]	Reserved	4	Reserved	Reserved	Reserved
0x400[21:20]	GP1_LOCK	2	Lock for GP1 fuses.	00 - Unlock 10 - OP 01 - WP 11 - OP + WP	OCOTP
0x400[23:22]	GP2_LOCK	2	Lock for GP2 fuses.	00 - Unlock 10 - OP 01 - WP	OCOTP

Table continues on the next page...

Table 6-35. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
				11 - OP + WP	
0x400[31:24]	Reserved	8	Reserved	Reserved	Reserved
0x410[3:0]	Reserved	4	Reserved	Reserved	Reserved
0x410[5:4]	GP4_LOCK	2	Lock for GP4 fuses.	00 - Unlock 10 - OP 01 - WP 11 - OP + WP	OCOTP
0x410[7:6]	Reserved	4	Reserved	Reserved	Reserved
0x410[9:8]	GP5_LOCK[1]	2	Lock for upper 64 bits of 128-bit UNIQUE_ID (UID) efuse field	00 - Unlock 10 - OP 01 - WP 11 - WP + OP	OCOTP
0x410[11:10]	GP6_LOCK	2	Lock for GP6 fuses.	00 - Unlock 10 - OP 01 - WP 11 - OP + WP	OCOTP
0x410[13:12]	GP7_LOCK	2	Lock for GP7 fuses.	00 - Unlock 10 - OP 01 - WP 11 - OP + WP	OCOTP
0x410[15:14]	GP8_LOCK	2	Lock for GP8 fuses.	00 - Unlock 10 - OP 01 - WP 11 - OP + WP	OCOTP
0x410[17:16]	GP9_LOCK	2	Lock for GP9 fuses.	00 - Unlock 10 - OP 01 - WP 11 - OP + WP	OCOTP
0x410[31:18]	Reserved	14	Reserved	Reserved	Reserved
0x420[10:0]	UNIQUE_ID[42:0]	43	Unique ID	-	SJC, SW
0x430[15:11]	UNIQUE_ID[47:43]	5	Unique ID	-	SJC, SW
0x430[23:16]	UNIQUE_ID[55:48]	8	Unique ID	-	SJC, SW
0x430[31:24]	UNIQUE_ID[63:56]	8	Unique ID	-	SJC, SW
0x440[7:0]	Reserved	8	Reserved	Reserved	Reserved

Table continues on the next page...

Table 6-35. Fusemap Descriptions (continued)

<table>
<tr><th>Fuse Address</th><th>Fuses Name</th><th>Number of Fuses</th><th>Fuses Function</th><th>Setting</th><th>Used by</th></tr>
<tr><td>0x440[13:8]</td><td>SPEED_GRADING[5:0]</td><td>6</td><td>Burned by tester program, for indicating IC core speed. (Hot burn may not be used).</td><td><table>
<tr><th>SPEED_GRADING[5:4]</th><th>SPEED_GRADING[3:0]</th><th>MHz</th><th>P/N Code</th></tr>
<tr><td>xx</td><td>0000</td><td>2300</td><td>23</td></tr>
<tr><td>xx</td><td>0001</td><td>2200</td><td>22</td></tr>
<tr><td>xx</td><td>0010</td><td>2100</td><td>21</td></tr>
<tr><td>xx</td><td>0011</td><td>2000</td><td>20</td></tr>
<tr><td>xx</td><td>0100</td><td>1900</td><td>19</td></tr>
<tr><td>xx</td><td>0101</td><td>1800</td><td>18</td></tr>
<tr><td>xx</td><td>0110</td><td>1700</td><td>17</td></tr>
<tr><td>xx</td><td>0111</td><td>1600</td><td>16</td></tr>
<tr><td>xx</td><td>1000</td><td>1500</td><td>15</td></tr>
<tr><td>xx</td><td>1001</td><td>1400</td><td>14</td></tr>
<tr><td>xx</td><td>1010</td><td>1300</td><td>13</td></tr>
<tr><td>xx</td><td>1011</td><td>1200</td><td>12</td></tr>
<tr><td>xx</td><td>1100</td><td>1100</td><td>11</td></tr>
<tr><td>xx</td><td>1101</td><td>1000</td><td>10</td></tr>
<tr><td>xx</td><td>1110</td><td>900</td><td>09</td></tr>
<tr><td>xx</td><td>1111</td><td>800</td><td>08</td></tr>
</table></td><td>PROD / SW</td></tr>
<tr><td>0x440[15:14]</td><td>Reserved</td><td>2</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0x440[16]</td><td>Reserved</td><td>1</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0x440[17]</td><td>Reserved</td><td>1</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0x440[19:18]</td><td>Reserved</td><td>2</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0x440[20]</td><td>Reserved</td><td>1</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0x440[21]</td><td>M7_DISABLE</td><td>1</td><td>Disable M7 Core.</td><td>0 - enabled
1 - disabled</td><td>M7</td></tr>
<tr><td>0x440[23:22]</td><td>Reserved</td><td>2</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0x440[24]</td><td>VPU_G1_DISABLE</td><td>1</td><td>Disable G1 Decoder in VPU</td><td>0 - enabled
1 - disabled</td><td>VPU</td></tr>
<tr><td>0x440[25]</td><td>VPU_G2_DISABLE</td><td>1</td><td>Disable G2 Decoder in VPU</td><td>0 - enabled
1 - disabled</td><td>VPU</td></tr>
<tr><td>0x440[27:26]</td><td>Reserved</td><td>2</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0x440[28]</td><td>CAN_DISABLE</td><td>1</td><td>Disable CAN functionality</td><td>0 - enabled</td><td>CAN</td></tr>
</table>

Table continues on the next page...

Table 6-35. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
				1 - disabled (Disables both CAN and CAN-FD functionality)	
0x440[29]	CAN_FD_DISABLE	1	Disable CAN-FD functionality	0 - enabled 1 - disabled (Disables CAN-FD, but CAN function is operational if CAN_DISABLE fuse is enabled)	CAN
0x440[30]	VPU_VC8000E_DISABLE	1	Disable VC8000E Encoder in VPU	0 - enabled 1 - disabled	VPU
0x440[31]	Reserved	1	Reserved	Reserved	Reserved
0x450[0]	IMG_ISP1_DISABLE	1	Disable IMAGING_ISP1 IP	0 - enabled 1 - disabled	ISP
0x450[1]	IMG_ISP2_DISABLE	1	Disable IMAGING_ISP2 IP	0 - enabled 1 - disabled	ISP
0x450[2]	IMG_DEWARP_DISABLE	1	Disable IMAGING_DEWARP IP	0 - enabled 1 - disabled	DEWARP
0x450[3]	NPU_DISABLE	1	Disable NPU IP	0 - enabled 1 - disabled	NPU
0x450[4]	AUDIO_PROCESSOR_DISABLE	1	Disable HiFi4 DSP IP	0 - enabled 1 - disabled	AUDIO
0x450[5]	ASRC_DISABLE	1	Disable Audio ASRC IP	0 - enabled 1 - disabled	ASRC
0x450[6]	GPU2D_DISABLE	1	Disable GPU 2D IP	0 - enabled 1 - disabled	GPU
0x450[7]	GPU3D_DISABLE	1	Disable GPU 3D IP.	0 - enabled 1 - disabled	GPU
0x450[8]	USB1_DISABLE	1	Disable USB1 Controller IP.	0 - enabled 1 - disabled	USB
0x450[9]	USB2_DISABLE	1	Disable USB2 Controller IP.	0 - enabled 1 - disabled	USB
0x450[10]	Reserved	1	Reserved	Reserved	Reserved
0x450[11]	PCIE1_DISABLE	1	Disable PCIe-1 IP.	0 - enabled 1 - disabled	PCIE
0x450[12]	Reserved	1	Reserved	Reserved	Reserved
0x450[13]	ENET1_DISABLE	1	Disable ENET-1 IP.	0 - enabled 1 - disabled	ENET
0x450[14]	ENET2_DISABLE	1	Disable ENET-2 IP.	0 - enabled 1 - disabled	ENET

Table continues on the next page...

Table 6-35. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
0x450[15]	MIPI_CSI1_DISABLE	1	Disable MIPI CSI-1 IP.	0 - enabled 1 - disabled	MIPI CSI
0x450[16]	MIPI_CSI2_DISABLE	1	Disable MIPI CSI-2 IP.	0 - enabled 1 - disabled	MIPI CSI
0x450[17]	MIPI_DSI1_DISABLE	1	Disable MIPI DSI-1 IP.	0 - enabled 1 - disabled	MIPI DSI
0x450[18]	Reserved	1	Reserved	Reserved	Reserved
0x450[19]	LVDS1_DISABLE	1	Disable LVDS-1 IP	0 - enabled 1 - disabled	LVDS
0x450[20]	LVDS2_DISABLE	1	Disable LVDS-2 IP	0 - enabled 1 - disabled	LVDS
0x450[21]	Reserved	1	Reserved	Reserved	Reserved
0x450[23:22]	Reserved	2	Reserved	Reserved	Reserved
0x450[24]	Reserved	1	Reserved	Reserved	Reserved
0x450[28:25]	Reserved	4	Reserved	Reserved	Reserved
0x450[29]	Reserved	1	Reserved	Reserved	Reserved
0x450[30]	EARC_RX_DISABLE	1	Disable eARC-RX IP	0 - enabled 1 - disabled	EARC
0x450[31]	Reserved	1	Reserved	Reserved	Reserved
0x460[6:0]	Reserved	7	Reserved	Reserved	Reserved
0x460[31:7]	GP4	25	General Purpose fuse register #4	-	SW
0x470[1:0]	FLEXSPI_AUTO_PROBE_TYPE	2	FLEXSPI Flash AutoProbe Type	00 - QuadSPI NOR 01 - MacronixOctal 10 - MicronOctal 11 - AdestoOctal	SW(ROM)
0x470[2]	FLEXSPI_AUTO_PROBE_EN	1	FLEXSPI Flash AutoProbe Enable	0 - Disable 1 - Enable	SW(ROM)
0x470[4:3]	OVERRIDE_FLEXSPI_BT_SEL_VAL	2	FlexSPI Boot Selection Override Value	00 - FlexSPI (Hyperflash 1.8V) 01 - FlexSPI (Flash with 4B READ(1x13) default supported) 10 - Default Octal mode (Micron) 11 - Default Octal mode (Macronix)	SW(ROM)
0x470[5]	OVERRIDE_FLEXSPI_BT_SEL	1	Override FlexSPI Boot Selection	0 - Do not override 1 - Override	SW(ROM)

Table continues on the next page...

Table 6-35. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
0x470[7:6]	OVERRIDE_NAND_PG_PER_BLK_VAL	2	Pages in block override value	RAWNAND: • 00 - 32 pages • 01 - 64 pages • 10 - 128 pages • 11 - 32 pages FlexSPI NAND: • 00 - 64 pages • 01 - 128 pages • 10 - 256 pages • 11 - 32 pages	SW(ROM)
0x470[8]	OVERRIDE_NAND_PG_PER_BLK	1	Override pages in block	0 - Do not override 1 - Override	SW(ROM)
0x470[10:9]	OVERRIDE_USDHC_BT_SEL_VAL	2	USDHC boot Select override value	00 - uSDHC1 SD 01 - uSDHC1 eMMC 10 - uSDHC2 eMMC 11 - uSDHC3 SD	SW(ROM)
0x470[11]	OVERRIDE_USDHC_BT_SEL	1	Override USDHC boot selection	0 - Do not override 1 - Override	SW(ROM)
0x470[15:12]	BOOT_MODE_FUSES	4	Boot Rom will retrieve boot mode from these fuses instead of BOOT_MODE pins if: (BOOT_MODE_PINS=0 or FORCE_BT_FROM_FUSE = 1) and BT_FUSE_SEL=1	Please refer to Table 6-32	SW(ROM)
0x470[24:16]	Reserved	9	Reserved	Reserved	Reserved
0x470[27:25]	Reserved	3	Reserved	Reserved	Reserved
0x470[28]	BT_FUSE_SEL	1	In the Boot From Fuse mode (BOOT_MODE[3:0] = 0000b), the BT_FUSE_SEL fuse indicates whether the bit configuration eFuses are programmed.	If Boot From Fuse mode (BOOT_MODE[3:0] = 0000b) or FORCE_BT_FROM_FUSE=1 : 0—The BOOT configuration eFuses are not programmed yet. The boot flow jumps to the serial downloader. 1—The BOOT configuration eFuses are programmed. The regular boot flow is performed.	SRC SW(ROM)
0x470[29]	FORCE_COLD_BOOT(SBMR)	1	Force cold boot when core comes out of reset. Reflected in SBMR register of SRC.	Fuse Function: 0 – Default behavior, allowing a fast recovery from low power modes. That is, the ROM is allowed to jump	SRC SW(ROM)

Table continues on the next page...

Table 6-35. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
				to the address previously programmed in the SRC persistent register. 1 – Fast recovery path in the ROM is not allowed and a cold boot is always performed. Customers wanting a higher level of security should burn this fuse.	
0x470[31:30]	Reserved	2	Reserved	Reserved	Reserved
0x480[2:0]	FLEXSPI_FEQ_SEL	3	FLEXSPI Flash Frequency	000 - 100 MHz 001 - 133 MHz 010 - 166 MHz 011 - 200 MHz 100 - 80 MHz 101 - 20 MHz	SW(ROM)
0x480[6:3]	FLEXSPI_DUMMY_CYCLE_SEL	4	FLEXSPI Flash Dummy Cycle	0 – Dummy cycle is auto-probed Other values – Actual dummy cycles for Read command	SW(ROM)
0x480[7]	Reserved	1	Reserved	Reserved	Reserved
0x480[8]	DCACHE_DIS	1	Disable L1 and L2 Dcache		SW(ROM)
0x480[9]	BT_FREQ_SEL	1	Boot frequency selection	BT_FREQ_SEL (Arm/DDR) 0 - 800 / 800 MHz 1 - 400 / 400 MHz	SW(ROM)
0x480[10]	WDOG_EN	1	Watchdog reset counter enable	0 - Disabled 1 - Enabled	SW(ROM)
0x480[11]	Reserved	1	Reserved	Reserved	Reserved
0x480[12]	ICACHE_DIS	1	L1 I-Cache DISABLE	0 - L1 I Cache is enabled by the ROM during the boot. 1 - L1 I Cache is disabled by the ROM during the boot.	SW(ROM)
0x480[13]	BT_LPB_POLARITY	1	USB Low-Power Boot GPIO polarity	0 - Low on the GPIO pad indicates the low power condition. 1 - High on the GPIO pad indicates the low power condition.	SW(ROM)
0x480[14:15]	BT_LPB	2	USB Low-Power Boot	0x - LPB Disable 10 - Div by 2	SW(ROM)

Table continues on the next page...

Table 6-35. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
				11 - Div by 4	
0x480[17:16]	WDOG_TIMEOUT_SELECT	2	Watchdog timeout select	00 - 2.0s 01 - 1.5s 10 - 1.0s 11 - 0.5s	SW(ROM)
0x480[19:18]	FLEXSPI_HOLD_TIME_SEL	2	Hold time before read from device	00 - 500us 01 - 1ms 10 - 3ms 11 - 10ms	SW(ROM)
0x480[20]	FORCE_BT_FROM_FUSE	1	Boot from programmed fuses, not Boot Mode Pins.	1 - boot flow is controlled by the BT_FUSE_SEL fuse value	SW(ROM)
0x480[21]	SDP_DISABLE	1	Disable USB serial download		SW(ROM)
0x480[22]	Reserved	1	Reserved	Reserved	Reserved
0x480[23]	NOC_ID_REMAP_BYPASS	1	NOC_ID_REMAP_BYPASS		SW(ROM)
0x480[24]	DCACHE_BYPASS_DIS	1	DCACHE_BYPASS_DIS		SW(ROM)
0x480[25]	RECOVER_ECSPI_BOOT_EN	1	OEM Recovery boot enable	0 - Disabled 1 - Enabled	SW(ROM)
0x480[27:26]	ECSPI_CS_SEL	2	CS selection (SPI only)	00 - CS#0 (default) 01 - CS#1 10 - CS#2 11 - CS#3	SW(ROM)
0x480[28]	ECSPI_ADDR_SEL	1	SPI addressing	0 - 3-bytes (24-bit) 1 - 2-bytes (16-bit)	SW(ROM)
0x480[31:29]	ECSPI_PORT_SEL	3	ECSPI port selection	000 - eCSPI1 001 - eCSPI2 010 - eCSPI3	SW(ROM)
0x490[0]	USDHC_MFG_VOL_SEL	1	For Mfg Mode IO Voltage	0 - 3.3V 1 - 1.8V	SW(ROM)
0x490[1]	USDHC_VOL_SEL	1	For Normal Boot Mode IO Voltage	0 - 3.3V 1 - 1.8V	SW(ROM)
0x490[3:2]	SD_SPEED	2	Speed mode	SD_SPEED: • 00 - Normal/SDR12 • 01 - High/SDR25 • 10 - SDR50 • 11 - SDR104 EMMC_SPEED:	SW(ROM)

Table continues on the next page...

Table 6-35. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
				• 00 - Normal • 01 - High	
0x490[5:4]	SDMMC_BUS_WIDTH	2	SDMMC Bus width	00 - 8-bit 01 - 4-bit 10 - 8-bit DDR (MMC 4.4) 11 - 4-bit DDR (MMC 4.4)	SW(ROM)
0x490[6]	EMMC_FAST_BT	1	Fast boot support	0 - Regular 1 - Fast Boot	SW(ROM)
0x490[7]	USDHC_PWR_EN	1	SD power cycle enable/ eMMC reset enable	0 - No power cycle 1 - Enabled	SW(ROM)
0x490[14:8]	SDMMC_DLL_DLY[6:0]	7	SD/MMC DLL DLY config	Delay target for USDHC DLL. It is applied to slave mode target delay or override mode target delay. Depends on DLL Override fuse bit value.	SW(ROM)
0x490[15]	USDHC_DLL_SEL	1	USDHC DLL override enable	0 - DLL Slave Mode 1 - DLL Override Mode	SW(ROM)
0x490[16]	USDHC_DLL_EN	1	USDHC DLL enable	0 - Disable DLL for SD/ eMMC 1 - Enable DLL for SD/ eMMC	SW(ROM)
0x490[17]	Reserved	1	Reserved	Reserved	SW(ROM)
0x490[18]	USDHC_PAD_SION_EN	1	USDHC IOMUX SION bit enable	0 - Disable 1 - Enable	SW(ROM)
0x490[22:19]	IMG_CNTN_SET1_OFFSET	4	Secondary Image Boot Offset	• Secondary boot is disabled if fuse value is bigger than 10, n = fuse value bigger than 10. • n == 0: Offset = 4MB • n == 2: Offset = 1MB • Others & n <= 10 : Offset = 1MB*2^n • For FlexSPI NOR boot, the valid values are: 0, 1, 2, 3, 4, 5, 6, and 7.	SW(ROM)
0x490[23]	RECOVERY_SDMMC_BOOT_DIS	1	Disable SDMMC manufacture mode	0 - Enable 1 - Disable	SW(ROM)
0x490[31:24]	USDHC_OVRD_PAD_SETTING_LOW8[7:0]	8	USDHC pad setting override	Override pad settings	SW(ROM)

Table continues on the next page...

Table 6-35. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
0x4A0[0]	EMMC_FAST_BT_ACK	1	Fast boot acknowledge enable	0 - Boot Ack Disabled 1 - Boot Ack Enabled	SW(ROM)
0x4A0[1]	USDHC_OVRD_PAD_SETTING_UP1	1	USDHC_OVRD_PAD_SETTING_UP1		SW(ROM)
0x4A0[2]	USDHC_PWR_POLARITY	1	USDHC power-off polarity selection	0 - Low 1 - High	SW(ROM)
0x4A0[3]	USDHC_PWR_DELAY	1	USDHC power cycle delay selection	0 - 5ms 1 - 2.5ms	SW(ROM)
0x4A0[5:4]	USDHC_PWR_INTERVAL	2	USDHC power cycle interval	00 - 20ms 01 - 10ms 10 - 5ms 11 - 2.5ms	SW(ROM)
0x4A0[7:6]	SD_CALI_STEP	2	SD_CALI_STEP	00 - 1 Others - N/A	SW(ROM)
0x4A0[8]	NAND_RST_TIME	1	NAND reset time		SW(ROM)
0x4A0[12:9]	NAND_TG_PREAMBLE_RD_LATENCY	4	Toggle Mode 33MHz Preamble Delay, Read Latency.	000 - 16 GPMICLK cycles. 001 - 1 GPMICLK cycles. 010 - 2 GPMICLK cycles. 011 - 3 GPMICLK cycles. 100 - 4 GPMICLK cycles. 101 - 5 GPMICLK cycles. 110 - 6 GPMICLK cycles. 111 - 7 GPMICLK cycles. 1111 - 15 GPMICLK cycles.	SW(ROM)
0x4A0[14:13]	NAND_FCB_SERCH_COUNT	2	Boot search count	00 - 2 01 - 2 10 - 4 11 - 8	SW(ROM)
0x4A0[15]	BT_TOGGLE_MODE	1	Boot toggle mode	0 - raw NAND 1 - toggle mode NAND	SW(ROM)
0x4A0[23:16]	Reserved	8	Reserved	Reserved	SW(ROM)
0x4A0[31:24]	NAND_OVERRIDE_PAD_SETTING[7:0]	8	NAND_OVERRIDE_PAD_SETTING[7:0]	NAND pad settings value	SW(ROM)
0x4B0[1:0]	NAND_CS_NUM	2	Nand Number Of Devices	00 - 1 01 - 2 10 - 4 11 - Reserved	SW(ROM)

Table continues on the next page...

Table 6-35. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
0x4B0[2]	Reserved	1	Reserved	Reserved	SW(ROM)
0x4B0[6:3]	NAND_GPMI_DDR_DLL_VAL	4	GPMI Read DDR DLL Target Value	0000 - 7 0001 - 1 0111 - 0 1111 - 15	SW(ROM)
0x4B0[9:7]	Reserved	3	Reserved	Reserved	SW(ROM)
0x4B0[11:10]	NAND_ROW_ADDR_BYTES	2	Row address cycles	00 - 3 01 - 2 10 - 4 11 - 5	SW(ROM)
0x4B0[15:12]	NAND_READ_RETRY_SEQ_ID[3:0]	4	NAND_READ_RETRY_SEQ_ID[2]	0000 - don't use read retry(RR) sequence embedded in ROM 0001 - Micron 20nm RR sequence 0010 - Kioxia A19nm RR sequence 0011 - Kioxia 19nm RR sequence 0100 - SanDisk 19nm RR sequence 0101 - SanDisk 1ynmRR sequence 0110 - SK Hynix 20nm A Die RR sequence 0111 - SK Hynix 26nm RR sequence Others - Reserved	SW(ROM)
0x4B0[31:16]	Reserved	16	Reserved	Reserved	Reserved
0x4C0 - 0x610	Reserved	704	Reserved	Reserved	Reserved
0x620[15:0]	USB_VID[31:0]	16	USB VID	-	SW
0x620[31:16]	USB_PID[31:0]	16	USB PID	-	SW
0x630[31:0]	Reserved	32	Reserved	Reserved	Reserved
0x640[15:0]	MAC_0_ADDR[47:0]	48	Reserved for customers/SW	-	SW
0x650[31:16]	MAC_1_ADDR[47:0]	48	Reserved for customers/SW	-	SW
0x670[31:0]	Reserved	32	Reserved	Reserved	>Reserved
0x680 - 0x6F0	Reserved	256	Reserved	Reserved	Reserved

Table continues on the next page...

Table 6-35. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
0x700 - 0x770	Reserved	256	Reserved	Reserved	Reserved
0x780 - 0x790	GP1[63:0]	64	General Purpose fuse register #1	-	SW
0x7A0 - 0x7B0	GP2[63:0]	64	General Purpose fuse register #2	-	SW
0x7C0[31:0]	Reserved	32	Reserved	Reserved	Reserved
0x7D0 - 0x7E0	Reserved	64	Reserved	Reserved	Reserved
0x7F0 - 0xDF0	Reserved	3104	Reserved	Reserved	Reserved
0xE00 - 0xE10	UNIQUE_ID[127:64]	64	Unique ID	-	SJC, SW
0xE20 - 0xE30	Reserved	64	Reserved	Reserved	Reserved
0xE40 - 0xE70	GP6[127:0]	128	General Purpose fuse register #6	-	PROD / SW
0xE80 - 0xEB0	GP7[127:0]	128	General Purpose fuse register #7	-	PROD / SW
0xEC0 - 0xEF0	GP8[127:0]	128	General Purpose fuse register #8	-	PROD / SW
0xF00 - 0xF30	GP9[127:0]	128	General Purpose fuse register #9	-	PROD / SW
0xF40 - 0x1BF0	Reserved	-	Reserved	Reserved	Reserved

1. The GP5_LOCK eFuse locks the upper 64 bits of the 128-bit UNIQUE_ID (UID) eFuse field. The users must program the GP5_LOCK eFuse to b'11 (WP+OP), thereby locking the upper 64-bit UID eFuse field.

   For devices shipped with date codes 2435 and later, the GP5_LOCK has been programmed to b'11 by NXP in production. Only devices with date codes prior to 2435 should be programmed to update the GP5_LOCK eFuse to b'11.

   For additional information, contact your NXP Account Manager or Field Representative.
2. Testing has been performed on select memory devices. Please contact your NXP representative for more details.

6.3 On-Chip OTP Controller (OCOTP_CTRL)

6.3.1 Overview

This section contains information describing the On-Chip OTP controller (OCOTP_CTRL) along with details about the block functionality and implementation.

In this document, the words "eFuse" and "OTP" are interchangeable. OCOTP refers to the hardware block itself.

6.3.1.1 Block Diagram

The system level block diagram for OCOTP is provided below.

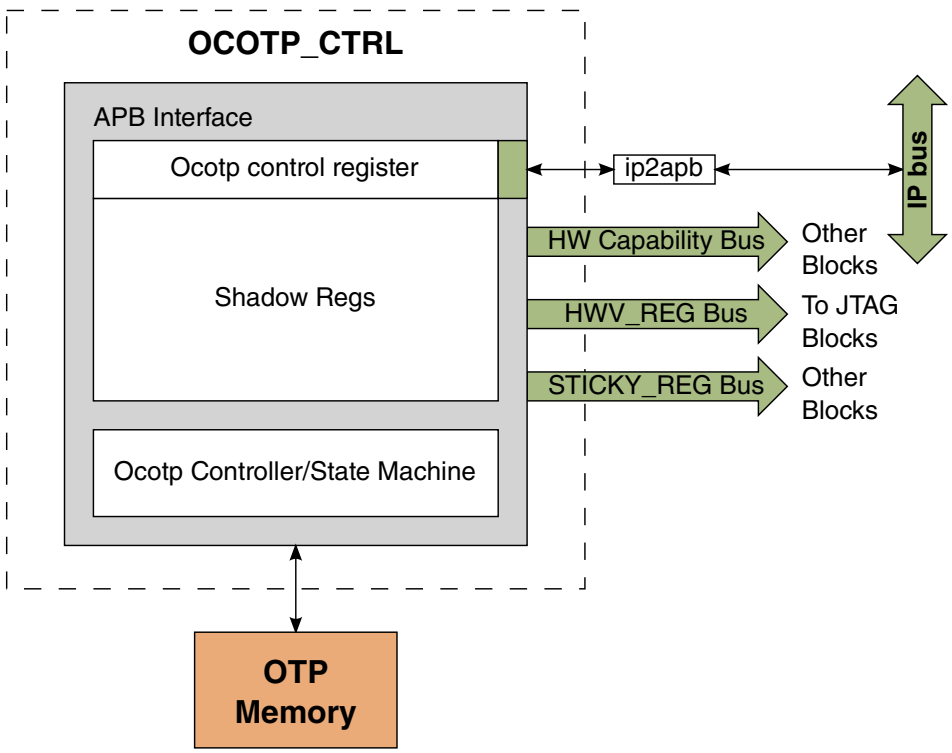


Figure 6-24. OCOTP System Level Block Diagram

6.3.1.2 Features

The OCOTP provides the following features :

- 32-bit word restricted program and read to
- Loading and housing of eFuse content into shadow registers
- Memory-mapped (restricted) access to shadow registers
- Provides program-protect and read-protect of eFuse
- Provides override and read protection of shadow register

6.3.2 Functional Description

6.3.2.1 Operations

The IP bus interface of the OCOTP has the following functions.

- Configure control registers for programming and reading all the fuse words
- Override and read shadow registers

OCOTP configuration for programs and reads are performed on 32-bit words for software (SW) convenience. For writes, the 32-bit word reflects the "write-mask". Bit fields with 0 will not be programmed and bit fields with 1 will be programmed.

6.3.2.1.1 Shadow Register Reload

All fuse words in efuse box are shadowed. This means that at reset, fuse values are read and then copied into memory-mapped shadow registers. If fuses are subsequently programmed, the shadow registers should be reloaded to keep them coherent with the fuse bank arrays.

The "reload shadows" feature allows the user to force a reload of the shadow registers without having to reset the device. To force a reload, complete the following steps:

1. Set the TIMING[STROBE_READ] field value appropriately.
2. Set the TIMING[RELAX] field value appropriately.
3. Check that CTRL[BUSY] and CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write, read, or reload must be completed before a new access can be requested.
4. Set the CTRL[RELOAD_SHADOWS] bit. OCOTP will read all the fuses one by one and put it into corresponding shadow register.
5. Wait for CTRL[BUSY] and CTRL[RELOAD_SHADOWS] to be cleared by the controller.

The controller will automatically clear the CTRL[RELOAD_SHADOWS] bit after the successful completion of the operation.

6.3.2.1.2 Fuse and Shadow Register Read

All shadow registers are always readable through the IPS bus except some secret key regions. When their corresponding fuse lock bits are set, the shadow registers also become read locked. After read locking, reading from these registers will return 0xBADABADA.

In addition, CTRL[ERROR] will be set. It must be cleared by software before any new write, read, or reload access can be issued. Subsequent reads to unlocked shadow registers will still work successfully.

To read fuse words directly from the fusebox instead of the shadow registers, complete the following steps:

1. Program TIMING[STROBE_READ] and TIMING[RELAX] fields with timing values to match the current frequency of the ipg_clk. OTP read will work at maximum bus frequencies as long as the TIMING parameters are set correctly.
2. Check that CTRL[BUSY] and CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write, read, or reload must be completed before a read access can be requested.
3. Write the requested fuse address to CTRL[ADDR].
4. Set READ_CTRL[READ_FUSE] to start the read operation.
5. Wait for the controller to clear CTRL[BUSY]. A read request to a protected or locked region will result in no OTP access and no setting of CTRL[BUSY]. In addition, CTRL[ERROR] will be set. It must be cleared by software before any new access can be issued.
6. Read READ_FUSE_DATA to get fuse word value. READ_FUSE_DATA will be 0xBADABADA when CTRL[ERROR] is set.

6.3.2.1.3 Fuse and Shadow Register Writes

Shadow register bits can be overridden by software until the corresponding fuse lock bit for the region is set. When the lock shadow bit is set, the shadow registers for that lock region become write locked. The LOCKn register also has no overwrite or fuse lock bits, but it is always read-only.

In order to avoid "rogue" code performing erroneous writes to OTP, a special unlocking sequence is required for writes to the fuse banks. To program fuse bank complete the following steps:

1. Program the following fields with timing values to match the frequency of ipg_clk:
   - TIMING[STROBE_PROG]
   - TIMING[RELAX]

OTP writes will work at maximum bus frequencies as long as the TIMING parameters are set correctly.

2. Check that CTRL[BUSY] and CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write, read, or reload must be completed before a write access can be requested.
3. Write the requested address to CTRL[ADDR] and program the unlock code into CTRL[WR_UNLOCK]. The unlock code must be programmed for each write access. The unlock code is documented in the register description. Both the unlock code and address can be written in the same operation.
4. Write the data to the DATA register. This will automatically set CTRL[BUSY] and clear CTRL[WR_UNLOCK]. Bit fields with 1's will result in that OTP bit being programmed. Bit fields with 0's will be ignored. At the same time that the write is accepted, the controller makes an internal copy of CTRL[ADDR] which cannot be updated until the next write sequence is initiated. This copy guarantees that erroneous writes to CTRL[ADDR] will not affect an active write operation. During the write operation, DATA cannot be modified.
5. Wait for the controller to clear CTRL[BUSY]. A write request to a protected or locked region will result in no OTP access and no setting of CTRL[BUSY]. In addition CTRL[ERROR] will be set. It must be cleared by software before any new write access can be issued.

It should be noted that write latencies to OTP are numbers of 10 micro-seconds per word. The write latency is based on the number of "1" bits being written. For example, to program half the fuse bits in one 32-bt word requires 10 us x 16.

6.3.2.1.4 Error Conditions

The CTRL[ERROR] bit will be set under the following conditions:

- A write is performed to a shadow register during a shadow reload (CTRL[RELOAD_SHADOWS] is set). In addition, the contents of the shadow register shall not be updated.
- A write is performed to a shadow register which has been locked.
- A read is performed to a shadow register which has been read locked.
- A program is performed to a fuse word which has been locked.
- A read is performed to a fuse word which has been read locked.
- Overlapping access - a read, write, or reload is attempted while a read, write, or reload is already in progress.

6.3.2.1.5 Write Postamble

Due to internal electrical characteristics of the OTP during writes, all OTP operations (direct reads, shadow reloads, or other writes) following a write must wait at least 2 µs after CTRL[BUSY] clears. The delay guarantees programming voltages on-chip reach a steady state when exiting a write sequence.

A recommended software sequence to meet the postamble requirements is as follows:

1. Issue the write and poll for CTRL[BUSY].
2. After CTRL[BUSY] is clear, wait 2 µs.
3. Perform the next OTP operation.

6.3.2.2 Fuse Shadow Memory Footprint

The OTP memory footprint is shown in the following figure. The registers are grouped by lock region. Their names correspond to the fusemap names.

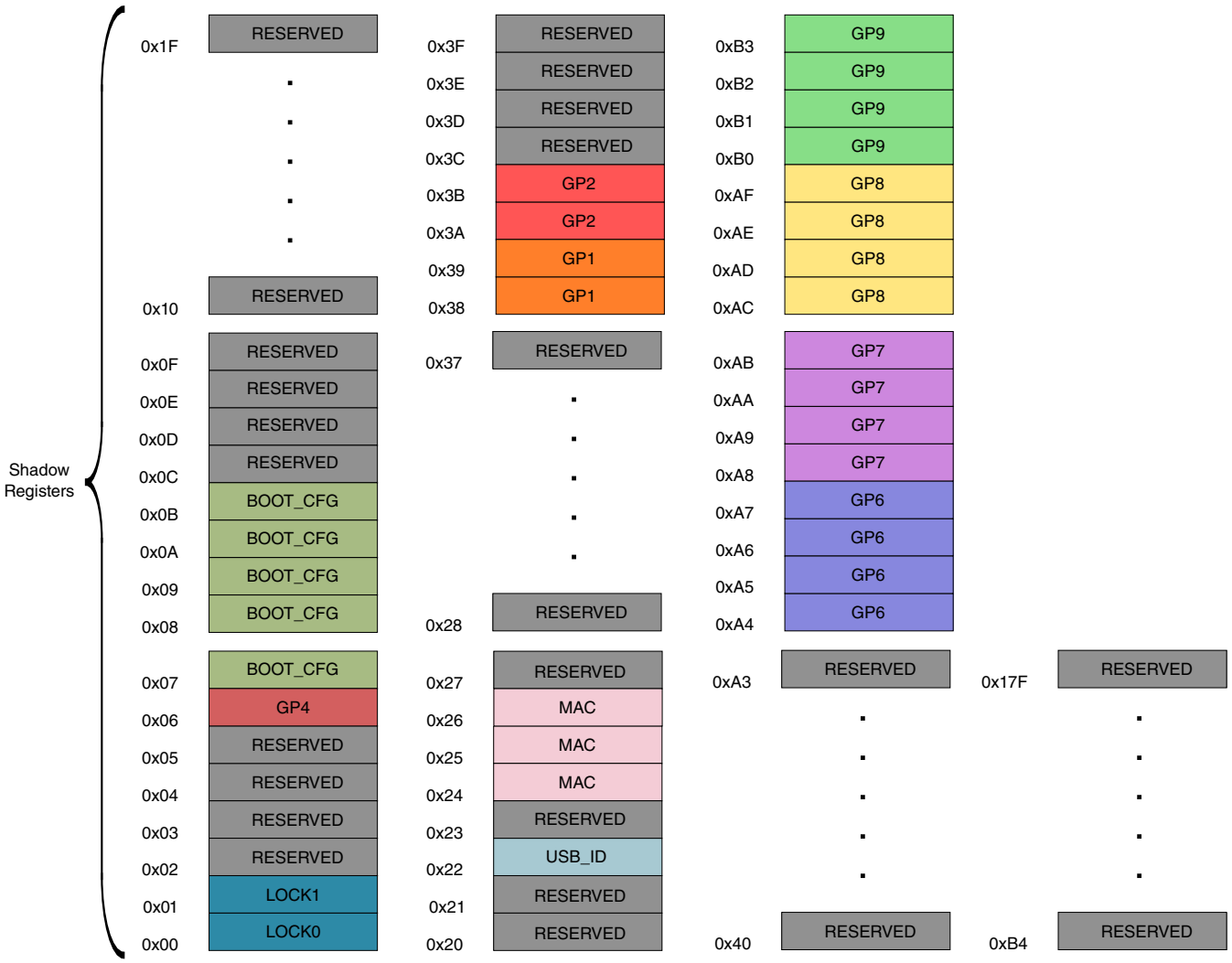


Figure 6-25. OTP Memory Footprint

6.3.2.3 OTP Read/Write Timing Parameters

The timing fields contained in the TIMING register that specify counter limit values, which are used to time how long the state machine remains in the various states, as well as specify the STROBE signal timing.

The timing parameters are specified in ipg_clk cycles. Since the ipg_clk frequency can be set to a range of values, these parameters must be adjusted with the clock to yield the appropriate delay.

The TIMING[RELAX] field specifies how long to remain in states to meet setup and hold timing requirement in fuse spec. This parameter should be set by the following equation:

tRELAX = tHP_PG = (TIMING[RELAX]+1)/ipg_frequency > 16.2ns

TIMING[RELAX] field is used to create other setup and hold timing delays in addition to tHP_PG. For all timing to be met, this is the max delay that must be programmed.

Except for setup and hold timing delay, there are 2 timing parameters for STROBE signal pulse width in program and read.

The TIMING[STROBE_PROG] field specifies the period of the STROBE signal for fuse writes and is given in units of ipg_clk cycles. This value should be specified so that the requirement for the time when the STROBE signal is asserted high is met: 9000ns < tPGM < 11000ns is met. Even though a range is given for tPGM, it is advised in **[eFUSE]** to program for a value of 10000ns. Therefore, this field should be set according to the following equation:

tPGM = ((TIMING[STROBE_PROG]+1)- 2*(TIMING[RELAX]+1))/ipg_frequency = 10000ns.

The TIMING[STROBE_READ] field specifies the period of the STROBE signal for fuse reads and is given in units of ipg_clk cycles. This field should be set according to the following equation:

tRD = ((TIMING[STROBE_READ]+1)- 2*(TIMING[RELAX]+1))/ipg_frequency > 36ns.

The figure below illustrates the relationship between the STROBE signal in programming and reading mode, as well as the timing PIO register fields that affect it. The implementation uses one counter to generate the STROBE waveform within one period and a second counter counts the number of cycles to create for programming the designated word.

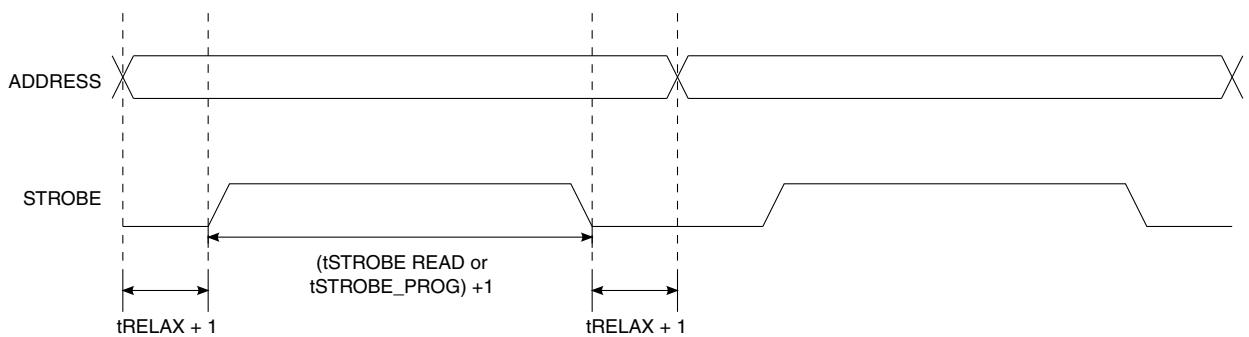


Figure 6-26. STROBE Signal Creation and Timing

6.3.2.4 Resets

The OCOTP is always active. The shadow registers automatically load the appropriate OTP contents after reset is deasserted. During this load-time CTRL[BUSY] is set.

6.3.2.5 Clocks

The table found here describes the clock sources for OCOTP. Please see the chip-specific clocking section for clock setting, configuration and gating information.

Table 6-36. OCOTP Clocks

Clock name	Description
ipg_clk	Peripheral clock
ipg_clk_s	Peripheral access clock

6.3.3 External Signals

There are no external signals pinned out for OCOTP.

6.3.4 Fuse Map

See the Fusemap chapter of this reference manual for more information.

6.3.5 Memory Map/Register Definition

The OCOTP Memory Map/Register Definition can be found here.

NOTE

Writing or reading an unimplemented register address in the OCOTP Controller will not send a bus error and read data will be 0.

6.3.5.1 register descriptions

6.3.5.1.1 OCOTP memory map

OCOTP base address: 3035_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	OTP Controller Control Register (HW_OCOTP_CTRL)	32	RW	0000_0000h
4h	OTP Controller Control Register (HW_OCOTP_CTRL_SET)	32	RW	0000_0000h
8h	OTP Controller Control Register (HW_OCOTP_CTRL_CLR)	32	RW	0000_0000h
Ch	OTP Controller Control Register (HW_OCOTP_CTRL_TOG)	32	RW	0000_0000h
10h	OTP Controller Timing Register (HW_OCOTP_TIMING)	32	RW	0148_1299h
20h	OTP Controller Write Data Register (HW_OCOTP_DATA)	32	RW	0000_0000h
30h	OTP Controller Write Data Register (HW_OCOTP_READ_CTRL)	32	RW	0000_0000h
40h	OTP Controller Read Data Register (HW_OCOTP_READ_FUSE_DATA)	32	RW	0000_0000h
90h	OTP Controller Version Register (HW_OCOTP_VERSION)	32	RO	0400_0000h
400h	Value of OTP Bank0 Word0 (Lock controls) (HW_OCOTP_LOCK0)	32	RO	0000_0000h
410h	Value of OTP Bank0 Word1 (Lock controls) (HW_OCOTP_LOCK1)	32	RO	FFFF_FFFFh
460h	Value of OTP Bank1 Word2 (HW_OCOTP_GP4)	32	RW	0000_0000h
470h	Value of OTP Bank1 Word3 (Boot Configuration Info.) (HW_OCOTP_BOOT_CFG0)	32	RW	0000_0000h
480h	Value of OTP Bank2 Word0 (Boot Configuration Info.) (HW_OCOTP_BOOT_CFG1)	32	RW	0000_0000h
490h	Value of OTP Bank2 Word1 (Boot Configuration Info.) (HW_OCOTP_BOOT_CFG2)	32	RW	0000_0000h
4A0h	Value of OTP Bank2 Word2 (Boot Configuration Info.) (HW_OCOTP_BOOT_CFG3)	32	RW	0000_0000h
4B0h	Value of OTP Bank2 Word3 (BOOT Configuration Info.) (HW_OCOTP_BOOT_CFG4)	32	RW	0000_0000h
620h	Value of OTP Bank8 Word2 (USB ID info) (HW_OCOTP_USB_ID)	32	RW	0000_0000h
640h	Value of OTP Bank9 Word0 (MAC Address) (HW_OCOTP_MAC_ADDR0)	32	RW	0000_0000h
650h	Value of OTP Bank9 Word1 (MAC Address) (HW_OCOTP_MAC_ADDR1)	32	RW	0000_0000h
660h	Value of OTP Bank9 Word2 (MAC Address) (HW_OCOTP_MAC_ADDR2)	32	RW	0000_0000h
780h	Value of OTP Bank14 Word0 () (HW_OCOTP_GP1_0)	32	RW	0000_0000h
790h	Value of OTP Bank14 Word1 () (HW_OCOTP_GP1_1)	32	RW	0000_0000h
7A0h	Value of OTP Bank14 Word2 () (HW_OCOTP_GP2_0)	32	RW	0000_0000h
7B0h	Value of OTP Bank14 Word3 () (HW_OCOTP_GP2_1)	32	RW	0000_0000h
E40h	Value of OTP Bank41 Word0 () (HW_OCOTP_GP6_0)	32	RW	0000_0000h
E50h	Value of OTP Bank41 Word1 () (HW_OCOTP_GP6_1)	32	RW	0000_0000h
E60h	Value of OTP Bank41 Word2 () (HW_OCOTP_GP6_2)	32	RW	0000_0000h
E70h	Value of OTP Bank41 Word3 () (HW_OCOTP_GP6_3)	32	RW	0000_0000h
E80h	Value of OTP Bank42 Word0 () (HW_OCOTP_GP7_0)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
E90h	Value of OTP Bank42 Word1 () (HW_OCOTP_GP7_1)	32	RW	0000_0000h
EA0h	Value of OTP Bank42 Word2 () (HW_OCOTP_GP7_2)	32	RW	0000_0000h
EB0h	Value of OTP Bank42 Word3 () (HW_OCOTP_GP7_3)	32	RW	0000_0000h
EC0h	Value of OTP Bank43 Word0 () (HW_OCOTP_GP8_0)	32	RW	0000_0000h
ED0h	Value of OTP Bank43 Word1 () (HW_OCOTP_GP8_1)	32	RW	0000_0000h
EE0h	Value of OTP Bank43 Word2 () (HW_OCOTP_GP8_2)	32	RW	0000_0000h
EF0h	Value of OTP Bank43 Word3 () (HW_OCOTP_GP8_3)	32	RW	0000_0000h
F00h	Value of OTP Bank44 Word0 () (HW_OCOTP_GP9_0)	32	RW	0000_0000h
F10h	Value of OTP Bank44 Word1 () (HW_OCOTP_GP9_1)	32	RW	0000_0000h
F20h	Value of OTP Bank44 Word2 () (HW_OCOTP_GP9_2)	32	RW	0000_0000h
F30h	Value of OTP Bank44 Word3 () (HW_OCOTP_GP9_3)	32	RW	0000_0000h

6.3.5.1.2 OTP Controller Control Register (HW_OCOTP_CTRL)

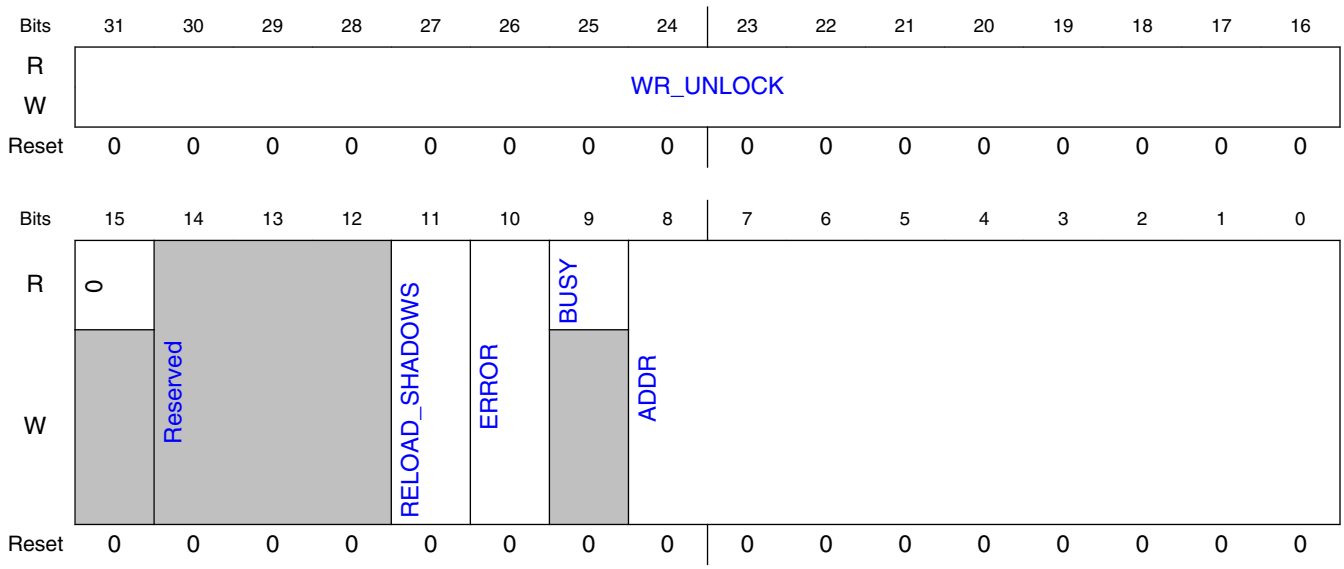
6.3.5.1.2.1 Offset

Register	Offset	Description
HW_OCOTP_CTRL	0h	OTP Controller Control Register
HW_OCOTP_CTRL_SET	4h	Writing a 1 to a bit in this register sets the corresponding bit in HW_OCOTP_CTRL
HW_OCOTP_CTRL_CLR	8h	Writing a 1 to a bit in this register clears the corresponding bit in HW_OCOTP_CTRL
HW_OCOTP_CTRL_TOG	Ch	Writing a 1 to a bit in this register toggles the corresponding bit in HW_OCOTP_CTRL

6.3.5.1.2.2 Function

The OCOTP Control and Status Register provides the necessary software interface for performing read and write operations to the On-Chip OTP (One-Time Programmable ROM). The control fields such as WR_UNLOCK, ADDR and BUSY/ERROR may be used in conjuction with the HW_OCOTP_DATA register to perform write operations. Read operations to the On-Chip OTP are involving ADDR, BUSY/ERROR bit field and HW_OCOTP_READ_CTRL register. Read value is saved in HW_OCOTP_READ_FUSE_DATA register.

6.3.5.1.2.3 Diagram



6.3.5.1.2.4 Fields

Field	Function
31-16 WR_UNLOCK	Write 0x3E77 to enable OTP write accesses. NOTE: This register must be unlocked on a write-by-write basis (a write is initiated when HW_OCOTP_DATA is written), so the UNLOCK bitfield must contain the correct key value during all writes to HW_OCOTP_DATA, otherwise a write shall not be initiated. This field is automatically cleared after a successful write completion (clearing of BUSY).
15 —	Reserved
14-12 —	Reserved
11 RELOAD_SHADOWS	Set to force re-loading the shadow registers (HW/SW capability and LOCK). This operation will automatically set BUSY. Once the shadow registers have been re-loaded, BUSY and RELOAD_SHADOWS are automatically cleared by the controller.
10 ERROR	Set by the controller when an access to a locked region(OTP or shadow register) is requested. Must be cleared before any further access can be performed. This bit can only be set by the controller. This bit is also set if the Pin interface is active and software requests an access to the OTP. In this instance, the ERROR bit cannot be cleared until the Pin interface access has completed. Reset this bit by writing a one to the SCT clear address space and not by a general write.
9 BUSY	OTP controller status bit. When active, no new write access or read access to OTP(including RELOAD_SHADOWS) can be performed. Cleared by controller when access complete. After reset (or after setting RELOAD_SHADOWS), this bit is set by the controller until the HW/SW and LOCK registers are successfully copied, after which time it is automatically cleared by the controller.
8-0 ADDR	OTP write and read access address register.

6.3.5.1.3 OTP Controller Timing Register (HW_OCOTP_TIMING)

6.3.5.1.3.1 Offset

Register	Offset
HW_OCOTP_TIMING	10h

6.3.5.1.3.2 Function

This register specifies timing parameters for programming and reading the OCOTP fuse array.

6.3.5.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				WAIT						STROBE_READ					
W																
Reset	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RELAX				STROBE_PROG											
W																
Reset	0	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1

6.3.5.1.3.4 Fields

Field	Function
31-28 —	Reserved
27-22 WAIT	This count value specifies time interval between auto read and write access in one time program. It is given in number of ipg_clk periods.
21-16 STROBE_READ	This count value specifies the strobe period in one time read OTP. Trd = ((STROBE_READ+1)-2*(RELAX+1)) /ipg_clk_freq. It is given in number of ipg_clk periods.
15-12 RELAX	This count value specifies the time to add to all default timing parameters other than the Tpgm and Trd. It is given in number of ipg_clk periods.
11-0 STROBE_PROG	This count value specifies the strobe period in one time write OTP. Tpgm = ((STROBE_PROG+1)-2*(RELAX+1)) /ipg_clk_freq. It is given in number of ipg_clk periods.

6.3.5.1.4 OTP Controller Write Data Register (HW_OCOTP_DATA)

6.3.5.1.4.1 Offset

Register	Offset
HW_OCOTP_DATA	20h

6.3.5.1.4.2 Function

This register is used in conjuction with HW_OCOTP_CTRL to perform one-time writes to the OTP. Please see the "Software Write Sequence" section for operating details.

6.3.5.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.4.4 Fields

Field	Function
31-0 DATA	Used to initiate a write to OTP. Please see the "Software Write Sequence" section for operating details.

6.3.5.1.5 OTP Controller Write Data Register (HW_OCOTP_READ_CTRL)

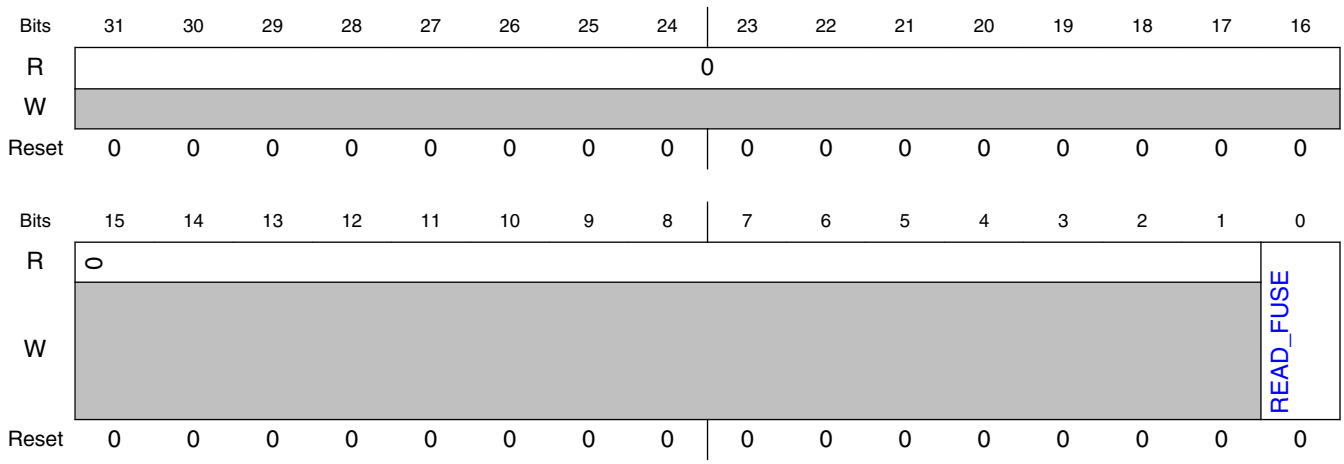
6.3.5.1.5.1 Offset

Register	Offset
HW_OCOTP_READ_CTRL	30h

6.3.5.1.5.2 Function

This register is used in conjuction with HW_OCOTP_CTRL to perform one time read to the OTP. Please see the "Software read Sequence" section for operating details.

6.3.5.1.5.3 Diagram



6.3.5.1.5.4 Fields

Field	Function
31-1 —	Reserved
0 READ_FUSE	Used to initiate a read to OTP. Please see the "Software read Sequence" section for operating details.

6.3.5.1.6 OTP Controller Read Data Register (HW_OCOTP_READ_FUSE_DATA)

6.3.5.1.6.1 Offset

Register	Offset
HW_OCOTP_READ_FUSE_DATA	40h

6.3.5.1.6.2 Function

The data read from OTP

6.3.5.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.6.4 Fields

Field	Function
31-0 DATA	The data read from OTP

6.3.5.1.7 OTP Controller Version Register (HW_OCOTP_VERSION)

6.3.5.1.7.1 Offset

Register	Offset
HW_OCOTP_VERSION	90h

6.3.5.1.7.2 Function

This register indicates the RTL version in use.

6.3.5.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STEP															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.7.4 Fields

Field	Function
31-24 MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23-16 MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
15-0 STEP	Fixed read-only value reflecting the stepping of the RTL version.

6.3.5.1.8 Value of OTP Bank0 Word0 (Lock controls) (HW_OCOTP_LOCK0)

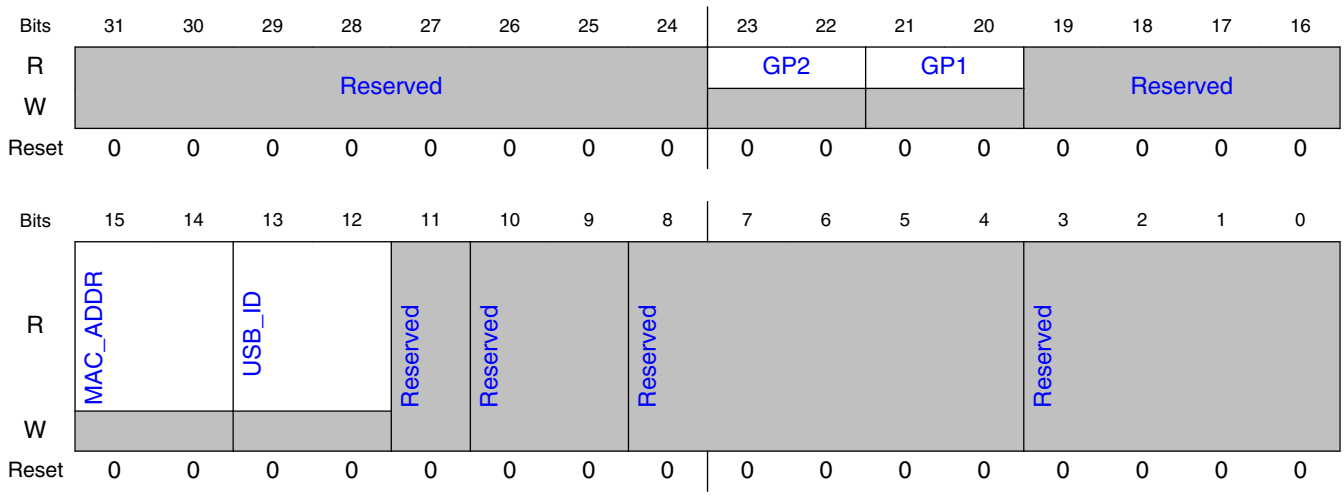
6.3.5.1.8.1 Offset

Register	Offset
HW_OCOTP_LOCK0	400h

6.3.5.1.8.2 Function

Shadowed memory mapped access to OTP Bank 0, word 0.

6.3.5.1.8.3 Diagram



6.3.5.1.8.4 Fields

Field	Function
31-24 —	Reserved
23-22 GP2	Status of shadow register and OTP write lock for gp2 region. 00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register) 01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)
21-20 GP1	Status of shadow register and OTP write lock for gp1 region. 00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register) 01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)
19-16 —	Reserved
15-14 MAC_ADDR	Status of shadow register and OTP write lock for mac_addr region. 00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register)

Table continues on the next page...

Field	Function
	10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register) 01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)
13-12 USB_ID	Status of shadow register and OTP write lock for usb_id region. 00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register) 01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)
11 —	Reserved
10-9 —	Reserved
8-4 —	Reserved
3-0 —	Reserved

6.3.5.1.9 Value of OTP Bank0 Word1 (Lock controls) (HW_OCOTP_LOCK1)

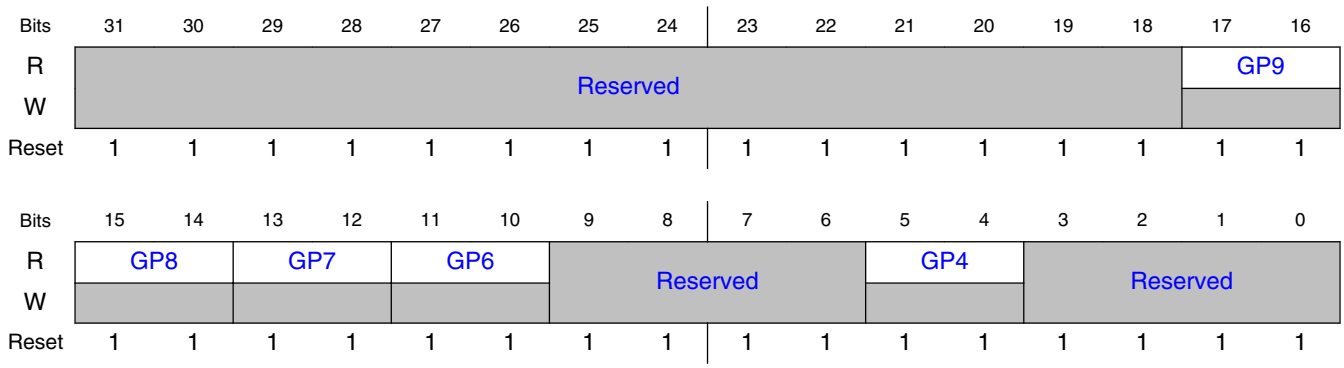
6.3.5.1.9.1 Offset

Register	Offset
HW_OCOTP_LOCK1	410h

6.3.5.1.9.2 Function

Shadowed memory mapped access to OTP Bank 0, word 1.

6.3.5.1.9.3 Diagram



6.3.5.1.9.4 Fields

Field	Function
31-18 —	Reserved
17-16 GP9	Status of shadow register and OTP write lock for gp9 region. 00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register) 01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)
15-14 GP8	Status of shadow register and OTP write lock for gp8 region. 00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register) 01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)
13-12 GP7	Status of shadow register and OTP write lock for gp7 region. 00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register) 01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)

Table continues on the next page...

Field	Function
11-10 GP6	Status of shadow register and OTP write lock for gp6 region. 00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register) 01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)
9-6 —	Reserved
5-4 GP4	Status of shadow register and OTP write lock for gp8 region. 00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register) 01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)
3-0 —	Reserved

6.3.5.1.10 Value of OTP Bank1 Word2 (HW_OCOTP_GP4)

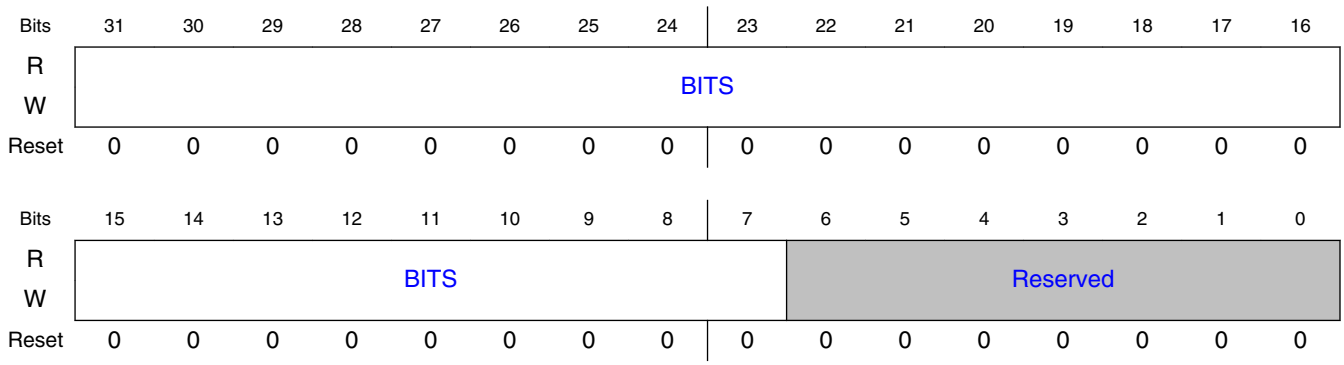
6.3.5.1.10.1 Offset

Register	Offset
HW_OCOTP_GP4	460h

6.3.5.1.10.2 Function

Shadowed memory mapped access to OTP Bank 1, word 2.

6.3.5.1.10.3 Diagram



6.3.5.1.10.4 Fields

Field	Function
31-7 BITS	Reflects value of OTP Bank 1, word 2, bits [31:7].
6-0 —	Reserved

6.3.5.1.11 Value of OTP Bank1 Word3 (Boot Configuration Info.) (HW_OCOTP_BOOT_CFG0)

6.3.5.1.11.1 Offset

Register	Offset
HW_OCOTP_BOOT_CFG0	470h

6.3.5.1.11.2 Function

Shadowed memory mapped access to OTP Bank 1, word 3.

6.3.5.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.11.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 1, word 3. These bits become read-only after the HW_OCOTP_LOCK_BOOT_CFG[1] bit is set.

6.3.5.1.12 Value of OTP Bank2 Word0 (Boot Configuration Info.) (HW_OCOTP_BOOT_CFG1)

6.3.5.1.12.1 Offset

Register	Offset
HW_OCOTP_BOOT_CFG1	480h

6.3.5.1.12.2 Function

Shadowed memory mapped access to OTP bank 2, word 0.

6.3.5.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.12.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP bank 2, word 0. These bits become read-only after the HW_OCOTP_LOCK_BOOT_CFG[1] bit is set.

6.3.5.1.13 Value of OTP Bank2 Word1 (Boot Configuration Info.) (HW_OCOTP_BOOT_CFG2)

6.3.5.1.13.1 Offset

Register	Offset
HW_OCOTP_BOOT_CFG2	490h

6.3.5.1.13.2 Function

Shadowed memory mapped access to OTP bank 2, word 1.

6.3.5.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.13.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP bank 2, word 1. These bits become read-only after the HW_OCOTP_LOCK_BOOT_CFG[1] bit is set.

6.3.5.1.14 Value of OTP Bank2 Word2 (Boot Configuration Info.) (HW_OCOTP_BOOT_CFG3)

6.3.5.1.14.1 Offset

Register	Offset
HW_OCOTP_BOOT_CFG3	4A0h

6.3.5.1.14.2 Function

Shadowed memory mapped access to OTP bank 2, word 2.

6.3.5.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.14.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP bank 2, word 2. These bits become read-only after the HW_OCOTP_LOCK_BOOT_CFG[1] bit is set.

6.3.5.1.15 Value of OTP Bank2 Word3 (BOOT Configuration Info.) (HW_OCOTP_BOOT_CFG4)

6.3.5.1.15.1 Offset

Register	Offset
HW_OCOTP_BOOT_CFG4	4B0h

6.3.5.1.15.2 Function

Shadowed memory mapped access to OTP bank 2, word 3.

6.3.5.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.15.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP bank 2, word 3. These bits become read-only after the HW_OCOTP_LOCK_BOOT_CFG[1] bit is set.

6.3.5.1.16 Value of OTP Bank8 Word2 (USB ID info) (HW_OCOTP_USB_ID)

6.3.5.1.16.1 Offset

Register	Offset
HW_OCOTP_USB_ID	620h

6.3.5.1.16.2 Function

Shadowed memory mapped access to OTP Bank 8, word 2.

6.3.5.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.16.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 8, word 2.

6.3.5.1.17 Value of OTP Bank9 Word0 (MAC Address) (HW_OCOTP_MAC_ADDR0)

6.3.5.1.17.1 Offset

Register	Offset
HW_OCOTP_MAC_ADDR0	640h

6.3.5.1.17.2 Function

Shadowed memory mapped access to OTP Bank 9, word 0.

6.3.5.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.17.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 9, word 0.

6.3.5.1.18 Value of OTP Bank9 Word1 (MAC Address) (HW_OCOTP_MAC_ADDR1)

6.3.5.1.18.1 Offset

Register	Offset
HW_OCOTP_MAC_ADDR1	650h

6.3.5.1.18.2 Function

Shadowed memory mapped access to OTP Bank 9, word 1.

6.3.5.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.18.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 9, word 1.

6.3.5.1.19 Value of OTP Bank9 Word2 (MAC Address) (HW_OCOTP_MAC_ADDR2)

6.3.5.1.19.1 Offset

Register	Offset
HW_OCOTP_MAC_ADDR2	660h

6.3.5.1.19.2 Function

Shadowed memory mapped access to OTP Bank 9, word 2.

6.3.5.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.19.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 9, word 2.

6.3.5.1.20 Value of OTP Bank14 Word0 () (HW_OCOTP_GP1_0)

6.3.5.1.20.1 Offset

Register	Offset
HW_OCOTP_GP1_0	780h

6.3.5.1.20.2 Function

Shadowed memory mapped access to OTP Bank 14, word 0.

6.3.5.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.20.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 14, word 0.

6.3.5.1.21 Value of OTP Bank14 Word1 () (HW_OCOTP_GP1_1)

6.3.5.1.21.1 Offset

Register	Offset
HW_OCOTP_GP1_1	790h

6.3.5.1.21.2 Function

Shadowed memory mapped access to OTP Bank 14, word 1.

6.3.5.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.21.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 14, word 1.

6.3.5.1.22 Value of OTP Bank14 Word2 () (HW_OCOTP_GP2_0)

6.3.5.1.22.1 Offset

Register	Offset
HW_OCOTP_GP2_0	7A0h

6.3.5.1.22.2 Function

Shadowed memory mapped access to OTP Bank 14, word 2.

6.3.5.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.22.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 14, word 2.

6.3.5.1.23 Value of OTP Bank14 Word3 () (HW_OCOTP_GP2_1)

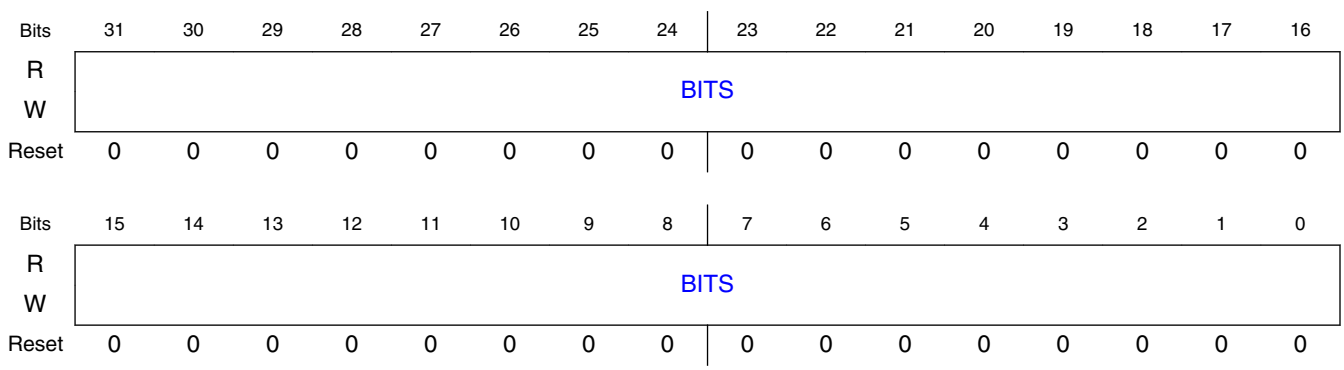
6.3.5.1.23.1 Offset

Register	Offset
HW_OCOTP_GP2_1	7B0h

6.3.5.1.23.2 Function

Shadowed memory mapped access to OTP Bank 14, word 3.

6.3.5.1.23.3 Diagram



6.3.5.1.23.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 14, word 3.

6.3.5.1.24 Value of OTP Bank41 Word0 () (HW_OCOTP_GP6_0)

6.3.5.1.24.1 Offset

Register	Offset
HW_OCOTP_GP6_0	E40h

6.3.5.1.24.2 Function

Shadowed memory mapped access to OTP Bank 41, word 0.

6.3.5.1.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.24.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 41, word 0.

6.3.5.1.25 Value of OTP Bank41 Word1 () (HW_OCOTP_GP6_1)

6.3.5.1.25.1 Offset

Register	Offset
HW_OCOTP_GP6_1	E50h

6.3.5.1.25.2 Function

Shadowed memory mapped access to OTP Bank 41, word 1.

6.3.5.1.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.25.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 41, word 1.

6.3.5.1.26 Value of OTP Bank41 Word2 () (HW_OCOTP_GP6_2)

6.3.5.1.26.1 Offset

Register	Offset
HW_OCOTP_GP6_2	E60h

6.3.5.1.26.2 Function

Shadowed memory mapped access to OTP Bank 41, word 2.

6.3.5.1.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.26.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 41, word 2.

6.3.5.1.27 Value of OTP Bank41 Word3 () (HW_OCOTP_GP6_3)

6.3.5.1.27.1 Offset

Register	Offset
HW_OCOTP_GP6_3	E70h

6.3.5.1.27.2 Function

Shadowed memory mapped access to OTP Bank 41, word 3.

6.3.5.1.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.27.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 41, word 3.

6.3.5.1.28 Value of OTP Bank42 Word0 () (HW_OCOTP_GP7_0)

6.3.5.1.28.1 Offset

Register	Offset
HW_OCOTP_GP7_0	E80h

6.3.5.1.28.2 Function

Shadowed memory mapped access to OTP Bank 42, word 0.

6.3.5.1.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.28.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 42, word 0.

6.3.5.1.29 Value of OTP Bank42 Word1 () (HW_OCOTP_GP7_1)

6.3.5.1.29.1 Offset

Register	Offset
HW_OCOTP_GP7_1	E90h

6.3.5.1.29.2 Function

Shadowed memory mapped access to OTP Bank 42, word 1.

6.3.5.1.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.29.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 42, word 1.

6.3.5.1.30 Value of OTP Bank42 Word2 () (HW_OCOTP_GP7_2)

6.3.5.1.30.1 Offset

Register	Offset
HW_OCOTP_GP7_2	EA0h

6.3.5.1.30.2 Function

Shadowed memory mapped access to OTP Bank 42, word 2.

6.3.5.1.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.30.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 42, word 2.

6.3.5.1.31 Value of OTP Bank42 Word3 () (HW_OCOTP_GP7_3)

6.3.5.1.31.1 Offset

Register	Offset
HW_OCOTP_GP7_3	EB0h

6.3.5.1.31.2 Function

Shadowed memory mapped access to OTP Bank 42, word 3.

6.3.5.1.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.31.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 42, word 3.

6.3.5.1.32 Value of OTP Bank43 Word0 () (HW_OCOTP_GP8_0)

6.3.5.1.32.1 Offset

Register	Offset
HW_OCOTP_GP8_0	EC0h

6.3.5.1.32.2 Function

Shadowed memory mapped access to OTP Bank 43, word 0.

6.3.5.1.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.32.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 43, word 0.

6.3.5.1.33 Value of OTP Bank43 Word1 () (HW_OCOTP_GP8_1)

6.3.5.1.33.1 Offset

Register	Offset
HW_OCOTP_GP8_1	ED0h

6.3.5.1.33.2 Function

Shadowed memory mapped access to OTP Bank 43, word 1.

6.3.5.1.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.33.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 43, word 1.

6.3.5.1.34 Value of OTP Bank43 Word2 () (HW_OCOTP_GP8_2)

6.3.5.1.34.1 Offset

Register	Offset
HW_OCOTP_GP8_2	EE0h

6.3.5.1.34.2 Function

Shadowed memory mapped access to OTP Bank 43, word 2.

6.3.5.1.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.34.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 43, word 2.

6.3.5.1.35 Value of OTP Bank43 Word3 () (HW_OCOTP_GP8_3)

6.3.5.1.35.1 Offset

Register	Offset
HW_OCOTP_GP8_3	EF0h

6.3.5.1.35.2 Function

Shadowed memory mapped access to OTP Bank 43, word 3.

6.3.5.1.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.35.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 43, word 3.

6.3.5.1.36 Value of OTP Bank44 Word0 () (HW_OCOTP_GP9_0)

6.3.5.1.36.1 Offset

Register	Offset
HW_OCOTP_GP9_0	F00h

6.3.5.1.36.2 Function

Shadowed memory mapped access to OTP Bank 44, word 0.

6.3.5.1.36.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.36.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 44, word 0.

6.3.5.1.37 Value of OTP Bank44 Word1 () (HW_OCOTP_GP9_1)

6.3.5.1.37.1 Offset

Register	Offset
HW_OCOTP_GP9_1	F10h

6.3.5.1.37.2 Function

Shadowed memory mapped access to OTP Bank 44, word 1.

6.3.5.1.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.37.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 44, word 1.

6.3.5.1.38 Value of OTP Bank44 Word2 () (HW_OCOTP_GP9_2)

6.3.5.1.38.1 Offset

Register	Offset
HW_OCOTP_GP9_2	F20h

6.3.5.1.38.2 Function

Shadowed memory mapped access to OTP Bank 44, word 2.

6.3.5.1.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.38.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 44, word 2.

6.3.5.1.39 Value of OTP Bank44 Word3 () (HW_OCOTP_GP9_3)

6.3.5.1.39.1 Offset

Register	Offset
HW_OCOTP_GP9_3	F30h

6.3.5.1.39.2 Function

Shadowed memory mapped access to OTP Bank 44, word 3.

6.3.5.1.39.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.5.1.39.4 Fields

Field	Function
31-0 BITS	Reflects value of OTP Bank 44, word 3.

6.4 Secure Non-Volatile Storage (SNVS)

6.4.1 Overview

The Secure Non-Volatile Storage (SNVS) is a companion module to the CAAM module.

The SNVS non-security functionality is described in this document, but the SNVS security functionality is described only in the Security Reference Manual.

6.4.1.1 Block diagram

The following figure illustrates the structure of SNVS.

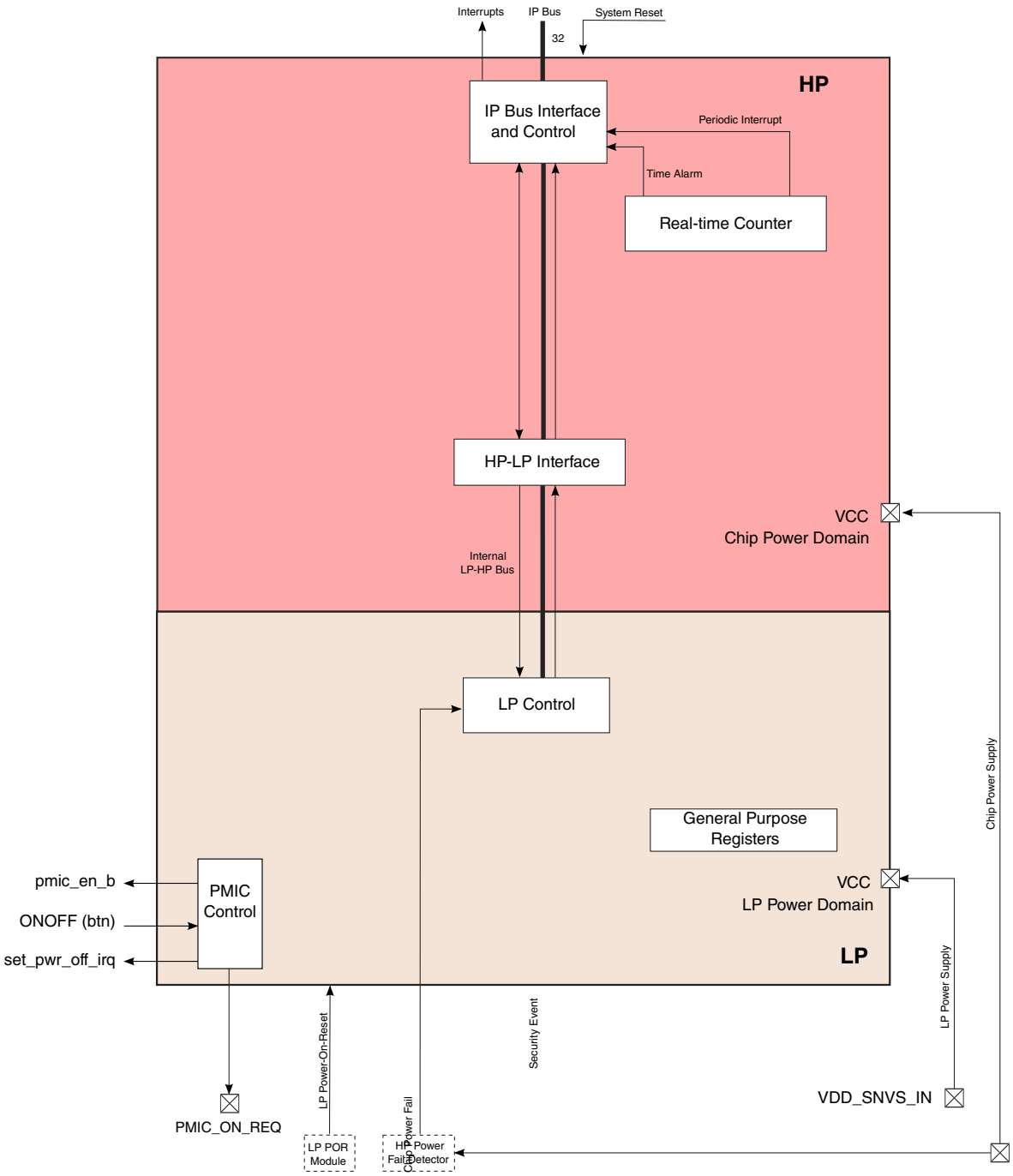


Figure 6-27. SNVS Block Diagram

6.4.1.2 Features

The following table summarizes the features of SNVS:

Table 6-37. SNVS feature list

Feature	Description	Links for Further Information
Real time counter (RTC)	• The RTC is driven by a dedicated clock, which is off when the system power is down. • Programmable time alarm interrupt	SNVS_HP Real Time Counter
General-purpose register	• The general-purpose register is available to software to store 128 bits of data. • The general-purpose register is zeroized when a security violation is detected. • If the SNVS_LP power input is connected to an uninterrupted power supply (see SNVS power domains), the general-purpose register value is retained even if the main chip is powered down.	Using the General-Purpose Register
Register access restrictions	• Some registers/values can be written only once per boot cycle.	privileged and non-privileged registers
Wakeup from power off	• Input signal from off chip requests SNVS_LP to power on the main SoC (Assuming that the SNVS_LP power input is connected to an uninterrupted power supply (see SNVS power domains). • Hardware debounces the input signal using software-specified signal bounce characteristics	LP Wake-Up Interrupt Enable

6.4.2 SNVS functional description

SNVS implements several non-security features that involve software interaction:

- reading or writing the Realtime Counter (RTC) (This is a non-privileged operation.) - software can also instruct SNVS to load the current SRTC value into the RTC
- reading or writing the General Purpose Register (GPR) (Note that there may be a significant delay when reading or writing registers in the LP section if the LP clock is different from the HP clock.)

The following sections describe in more detail the operation of SNVS.

6.4.2.1 SNVS power domains

In some versions of SNVS (including this version), the LP (Low Power) section is implemented in an independent power domain from the HP (High Power) section, and most other logic on the chip. Throughout the SNVS documentation whenever mention is made of "always-on" logic, this assumes a version of SNVS that implements an independent power domain for the LP section, and that the power for this section is supplied by an uninterrupted power supply. The purpose for the independent power

domain is so that data can be retained and certain logic can remain functional even when the main chip logic is powered down. But this is possible only if the LP domain remains powered via an uninterrupted power supply when the main chip power domain is powered off. Usually this uninterrupted power supply would be a battery, with possibly some power management logic to power the LP section from main power (and perhaps recharge the battery) when main power is on, and switch to battery power when the main power is off. In versions of SNVS with an independent LP power domain the LP section can be electrically isolated from the rest of the chip logic to ensure that its logic does not get corrupted when the main chip is powered down. If the battery runs down or is removed, an LP POR will occur when the LP section next powers up. Note that some OEMs may choose to connect LP power to HP/main chip power and dispense with a battery. In that case the SNVS will operate the same as an SNVS without an independent LP power domain. No state will be retained in the LP section when the chip is powered down, and an LP POR will occur whenever there is an HP POR.

6.4.2.2 Runtime Procedures

SNVS implements a number of features that are intended to be accessed by software at runtime (as opposed to accessed at boot time). These features include:

- Real Time Clock (see SNVS_HP Real Time Counter)
- General Purpose Register (see Using the General-Purpose Register)

Procedures for using these features are described in the following sections.

6.4.2.2.1 Using SNVS Timer Facilities

SNVS incorporates timer facilities that can optionally generate an interrupt at a specified time. As described in the following sections, SNVS_HP incorporates a Real Time Counter that is available for general use, and SNVS_LP incorporates a Secure Real Time Counter intended for security applications.

6.4.2.2.1.1 SNVS_HP Real Time Counter

SNVS_HP implements a real time counter that can be read or written by any application; it has no privileged software access restrictions. When the chip is powered down the RTC is not active and it is reset at chip POR. The RTC can be used to generate a functional interrupt request either at a specific time, or at a specific frequency, or both. To generate an interrupt request at a specific time HPTA_EN is set to 0, the desired time is written to HPTA_MS and HPTA_LS and then HPTA_EN is set to 1. HPTA_EN, HPTA_MS and HPTA_LS can be written by any software that has access to SNVS registers; there are no

privileged access restrictions. The counter can be synchronized to the SNVS_LP SRTC by writing to the HP_TS bit of SNVS_HP Control Register. This is particularly useful if the SNVS_LP is powered from an uninterrupted power source because the RTC can then be set from a chip-internal time source.

6.4.2.2.1.2 RTC/SRTC control bits setting

All SNVS registers are programmed from the register bus, consequently any software-initiated changes are synchronized with the IP clock. Several registers can also change synchronously with the RTC/SRTC clock after they are programmed. To avoid IP clock and RTC/SRTC clock synchronization issues, the following values can be changed only when the corresponding function is disabled.

Table 6-38. RTC/SRTC synchronized values list

Function	Value/register	Control bit setting
HP section		
HP Real Time Counter	HPRTCMR and HPRTCLR Registers	RTC_EN = 0 : HPRTCMR/HPRTCLR **can** be programmed RTC_EN = 1 : HPRTCMR/HPRTCLR **cannot** be programmed
HP Time Alarm	HPTAMR and HPTALR Registers	HPTA_EN = 0 : HPTAMR/HPTALR **can** be programmed HPTA_EN = 1 : HPTAMR/HPTALR **cannot** be programmed
LP section		
LP Secure Real Time Counter	LPRTCMR and LPRTCLR Registers	SRTC_ENV = 0 : LPRTCMR/LPRTCLR **can** be programmed SRTC_ENV = 1 : LPRTCMR/LPRTCLR **cannot** be programmed
LP Time Alarm	LPTAR Register	LPTA_EN = 0 : LPTAR **can** be programmed LPTA_EN = 1 : LPTAR **cannot** be programmed

Use the following steps to program synchronized values:

1. Check the enable bit value. If set, clear it.

2. Verify that the enable bit is cleared. There are two reasons to verify the enable bit's setting:

- Enable bit clearing does not happen immediately; it takes three IP clock cycles and two RTC/SRTC clock cycles to change the enable bit's value.
- If the enable bit is locked for programming, it cannot be cleared.

3. Program the desired value.

4. Set the enable bit; it takes three IP clock cycles and two RTC/SRTC clock cycles for the bit to set.

NOTE

Incrementing the value programmed into RTC/SRTC registers by two compensates for the two RTC/SRTC clock cycle delay that is required to enable the counter.

6.4.2.2.1.3 Reading RTC and SRTC values

Software should follow the following procedure to ensure that it has read correct data from the RTC (HPRTCMR and HPRTCLR) and SRTC (LPSRTCMR and LPSRTCLR) registers:

- Read the most-significant half and the least-significant half of the RTC/SRTC and then read both halves again. If the values read are the same both times, the value is correct.
- If the two consecutive pairs of reads yield different results, perform two more reads.

The worst case scenario may require three sessions of two consecutive pairs of reads. There are several reasons that the values may be incorrectly read initially:

- Synchronization issues between the RTC/SRTC clock and the system clock
- Since the counter continues to increment, there may be a carry from the least-significant 32-bits to the most-significant bits in between reading the two halves of the counter

6.4.2.2.2 Using Other SNVS Registers

The sections below describe how to use the General Purpose Register.

6.4.2.2.2.1 Using the General-Purpose Register

SNVS implements a 128-bit general-purpose register allows software to store a small amount of data. To maintain backward compatibility with versions of SNVS that implement only a 32-bit general purpose register, the most-significant word of the general purpose register is aliased to the original legacy address, and to maintain backward compatibility with versions of snvs_module_name that implement a 128-bit general purpose register, the most-significant half of the general purpose register is

aliased to the previous legacy address address. The data in the GPR will be retained during system power-down mode as long as the SNVS_LP remains powered by an uninterrupted power source.

6.4.2.3 Clocks

The SNVS has the following clock sources:

- System peripheral clock input. This clock is used by the SNVS's internal logic, for example, the Security State Machine. This clock can be gated outside of the module when the SNVS indicates that it is not in use.
- HP RTC clock. This clock is used by SNVS_HP real-time counter. This clock does not need to be synchronous with other clocks.

6.4.2.4 Reset

The table below shows the different resets associated with this module.

Table 6-39. Reset summary

Reset	Source	Characteristics	Internally resets
HP hard	ipg_hard_async_reset_b	active-low, asynchronous	All SNVS_HP and SNVS_LP registers and flops.
LP Power On Reset (POR)	lp_por_b	active-low, asynchronous	All SNVS_LP registers and flops.
LP software reset	software	active-high, synchronous, one cycle	All SNVS_LP registers and flops. The LP software reset can be asserted if not disabled.

6.4.3 External Signals

Table 6-40. External Signals

Signal	Description	Direction
PMIC_ON_REQ	PMIC ON Request signal	I

6.4.4 Initialization of SNVS

SNVS is implemented in two sections (HP and LP) that both must be initialized by software. If the SNVS_LP is powered by an uninterrupted power source that is separate from main SoC power, then SNVS can operate in either of two modes, depending upon whether the main SoC power is on or off.

- During main SoC power-down SNVS_HP is powered-down, but SNVS_LP is powered from the backup power supply and is electrically isolated from the rest of the chip. In this mode SNVS_LP keeps its registers' values but the LP registers cannot be read or written.
- During main SoC power-up the isolation of SNVS_LP is disabled and both SNVS_HP and SNVS_LP are powered from the main SoC power. Both LP and HP registers can be read and written (locks and privilege modes permitting). Signals between the SNVS_HP and SNVS_LP sections are enabled and all SNVS functions are operational.

Since the HP and LP sections reside in different power domains, the POR for the two sections can occur at different times. If the SNVS_LP section remains powered by an uninterrupted power source when the main SoC power is off, SNVS_LP is initialized rarely, typically once when the device is first powered on and again whenever the battery is replaced. During main SoC power-up the isolation of SNVS_LP is disabled and both SNVS_HP and SNVS_LP are powered from the main SoC power. Signals between the SNVS_HP and SNVS_LP sections are enabled and all SNVS functions are operational. The SNVS_HP section is powered from the main SoC power, so it must be initialized after the device is powered on. If the SNVS_LP section is powered from the main SoC power rather than from an uninterrupted power source, the SNVS_LP section must also be initialized at SoC POR.

- Initializing the LP section
  - The following steps should be completed to properly initialize the SNVS LP section (required only on LP POR, i.e. when the battery is replaced):
- Initializing the HP section
  - The following steps should be completed to properly initialize the SNVS HP section (required on HP POR, i.e. SoC POR):
    - Perform normal boot to put the SNVS into a functional state (Non-secure, Trusted, Secure) (see HP Command Register, SSM_ST bitfield).
    - Program SNVS general functions/configurations (see HP Control Register).

6.4.5 Memory Map and register definition

This section contains detailed register descriptions for the SNVS registers. Each description includes a standard register diagram and register table. The register table provides detailed descriptions of the register bit and field functions, in bit order.

SNVS registers consist of two types:

- Privileged read/write accessible
- Non-privileged read/write accessible

Privileged read/write accessible registers can only be accessed for read/write by privileged software. Unauthorized write accesses are ignored, and unauthorized read accesses return zero. Non-privileged software can access privileged access registers when the non-privileged software access enable bit is set in the SNVS_HP Command Register.

- Non-Secure
- Trusted
- Secure

Non-privileged read/write accessible registers are read/write accessible by any software.

The LP register values are set only on LP POR and are unaffected by System (HP) POR. The HP registers are set only on System POR and are unaffected by LP POR.

The following table shows the SNVS main memory map.

NOTE

For more information on security-related bitfields, see the Security Reference Manual.

6.4.5.1 SNVS register descriptions

6.4.5.1.1 SNVS memory map

SNVS base address: 3037_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	SNVS_HP Command Register (HPCOMR)	32	RW	0000_0000
8	SNVS_HP Control Register (HPCR)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
14	SNVS_HP Status Register (HPSR)	32	RW	8000_0000
24	SNVS_HP Real Time Counter MSB Register (HPRTCMR)	32	RW	0000_0000
28	SNVS_HP Real Time Counter LSB Register (HPRTCLR)	32	RW	0000_0000
2C	SNVS_HP Time Alarm MSB Register (HPTAMR)	32	RW	0000_0000
30	SNVS_HP Time Alarm LSB Register (HPTALR)	32	RW	0000_0000
34	SNVS_LP Lock Register (LPLR)	32	RW	0000_0000
38	SNVS_LP Control Register (LPCR)	32	RW	0000_0020
4C	SNVS_LP Status Register (LPSR)	32	RW	0000_0008
68	SNVS_LP General Purpose Register 0 (legacy alias) (LPGPR0_legacy_alias)	32	RW	0000_0000
90 - 9C	SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0 - LPGPR3)	32	RW	0000_0000
BF8	SNVS_HP Version ID Register 1 (HPVIDR1)	32	RO	003E_0103
BFC	SNVS_HP Version ID Register 2 (HPVIDR2)	32	RO	0600_0300

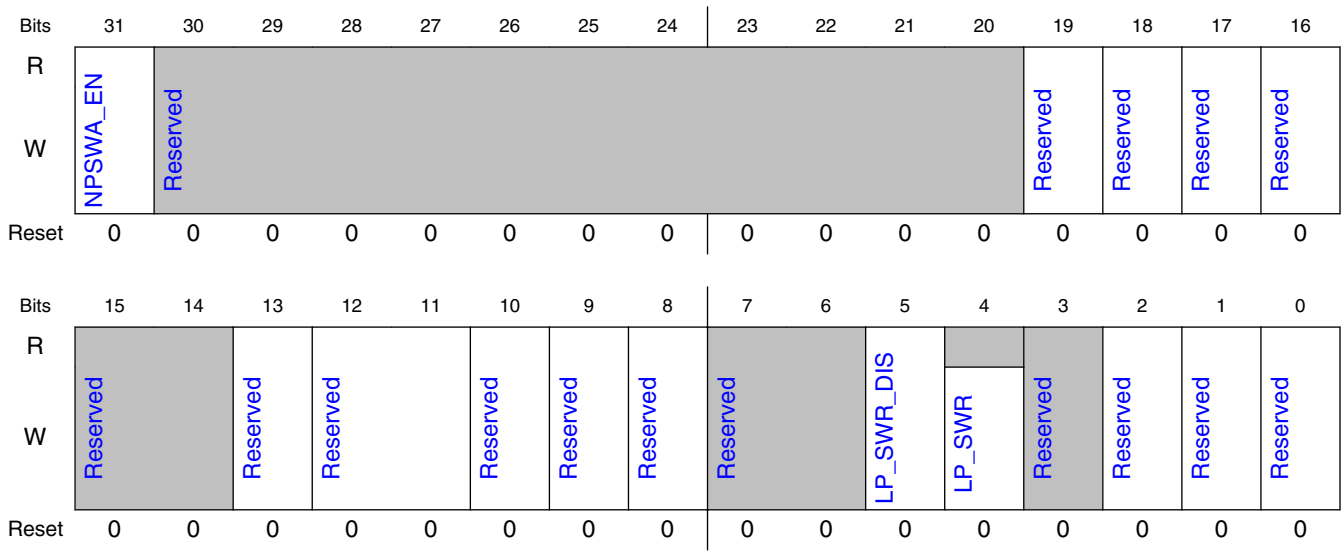
6.4.5.1.2 SNVS_HP Command Register (HPCOMR)

The SNVS_HP Command Register contains the command, configuration, and control bits for the SNVS block. This is a privileged write register.

6.4.5.1.2.1 Offset

Register	Offset
HPCOMR	4h

6.4.5.1.2.2 Diagram



6.4.5.1.2.3 Fields

Field	Description
31 NPSWA_EN	Non-Privileged Software Access Enable When set, allows non-privileged software to access all SNVS registers, including those that are privileged software read/write access only. 0 Only privileged software can access privileged registers 1 Any software can access privileged registers
30-20 —	Reserved
19 —	Reserved
18 —	Reserved
17 —	Reserved
16 —	Reserved
15-14 —	Reserved
13 —	Reserved
12-11 —	Reserved

Table continues on the next page...

Field	Description
10 —	Reserved
9 —	Reserved
8 —	Reserved
7-6 —	Reserved
5 LP_SWR_DIS	LP Software Reset Disable When set, disables the LP software reset. Once set, this bit can only be reset by the system reset. 0 - LP software reset is enabled 1 - LP software reset is disabled
4 LP_SWR	LP Software Reset. When set to 1, the registers in the SNVS_LP section are reset. 0 - No Action 1 - Reset LP section
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 —	Reserved

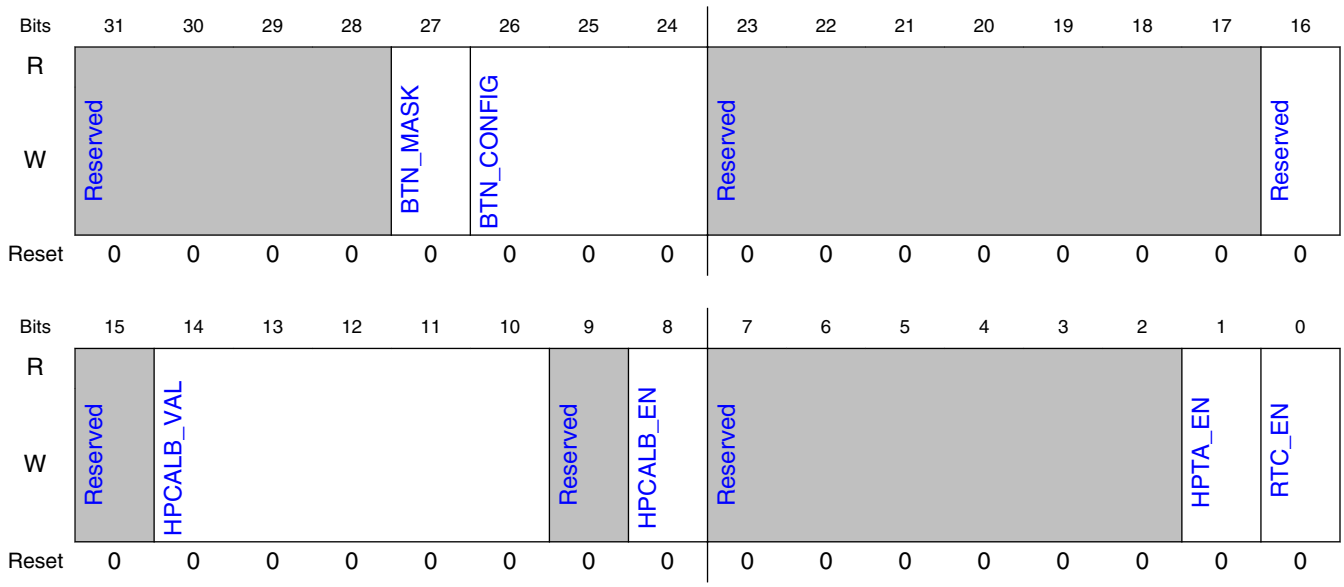
6.4.5.1.3 SNVS_HP Control Register (HPCR)

The SNVS_HP Control Register contains various control bits of the HP section of SNVS. This is *not* a privileged write register.

6.4.5.1.3.1 Offset

Register	Offset
HPCR	8h

6.4.5.1.3.2 Diagram



6.4.5.1.3.3 Fields

Field	Description
31-28 —	Reserved
27 BTN_MASK	Button Interrupt Mask This bit is used to mask the button (BTN) interrupt request. 0: Interrupt disabled 1: Interrupt enabled
26-24 BTN_CONFIG	Button Configuration This field is used to configure which feature of the button (BTN) input signal constitutes "active". 000: Button signal is active high 001: Button signal is active low 010: Button signal is active on the falling edge 011: Button signal is active on the rising edge 100: Button signal is active on any edge All other patterns are Reserved
23-17 —	Reserved
16 —	Reserved
15 —	Reserved

Table continues on the next page...

Field	Description
14-10 HPCALB_VAL	HP Calibration Value Defines signed calibration value for the HP Real Time Counter. This field can be programmed only when RTC Calibration is disabled (HPCALB_EN is not set). This is a 5-bit 2's complement value, hence the allowable calibration values are in the range from -16 to +15 counts per 32768 ticks of the counter. 00000 - +0 counts per each 32768 ticks of the counter 00001 - +1 counts per each 32768 ticks of the counter 00010 - +2 counts per each 32768 ticks of the counter 01111 - +15 counts per each 32768 ticks of the counter 10000 - -16 counts per each 32768 ticks of the counter 10001 - -15 counts per each 32768 ticks of the counter 11110 - -2 counts per each 32768 ticks of the counter 11111 - -1 counts per each 32768 ticks of the counter
9 —	Reserved
8 HPCALB_EN	HP Real Time Counter Calibration Enabled Indicates that the time calibration mechanism is enabled. 0 - HP Timer calibration disabled 1 - HP Timer calibration enabled
7-2 —	Reserved
1 HPTA_EN	HP Time Alarm Enable When set, the time alarm interrupt is generated if the value in the HP Time Alarm Registers is equal to the value of the HP Real Time Counter. This bit syncs with the 32KHz clock. It won't update with the bus clock. 0 - HP Time Alarm Interrupt is disabled 1 - HP Time Alarm Interrupt is enabled
0 RTC_EN	HP Real Time Counter Enable This bit syncs with the 32KHz clock. It won't update with the bus clock. 0 - RTC is disabled 1 - RTC is enabled

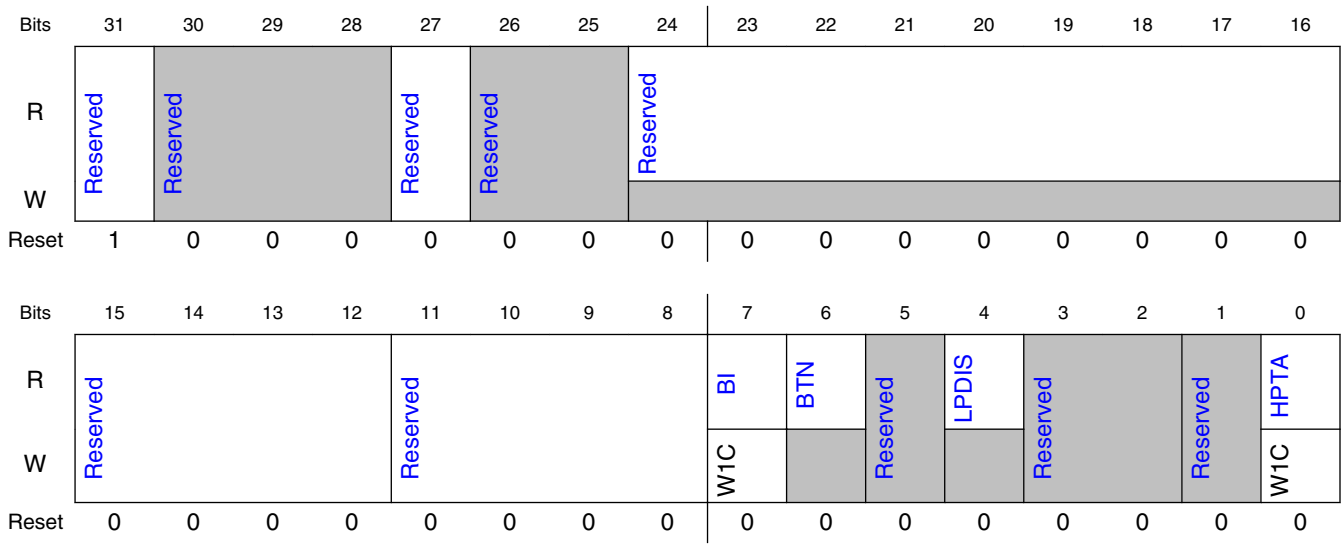
6.4.5.1.4 SNVS_HP Status Register (HPSR)

The HP Status Register reflects the internal state of the SNVS. This is *not* a privileged write register.

6.4.5.1.4.1 Offset

Register	Offset
HPSR	14h

6.4.5.1.4.2 Diagram



6.4.5.1.4.3 Fields

Field	Description
31 —	Reserved
30-28 —	Reserved
27 —	Reserved
26-25 —	Reserved
24-16 —	Reserved
15-12 —	Reserved
11-8 —	Reserved
7	Button Interrupt

Table continues on the next page...

Field	Description
BI	Signal ipi_snvs_btn_int_b was asserted.
6 BTN	Button Value of the BTN input. This is the external button used for PMIC control. 0: BTN not pressed 1: BTN pressed
5 —	Reserved
4 LPDIS	Low Power Disable If 1, the low power section has been disabled by means of an input signal to SNVS.
3-2 —	Reserved
1 —	Reserved
0 HPTA	HP Time Alarm Indicates that the HP Time Alarm has occurred since this bit was last cleared. 0 - No time alarm interrupt occurred. 1 - A time alarm interrupt occurred.

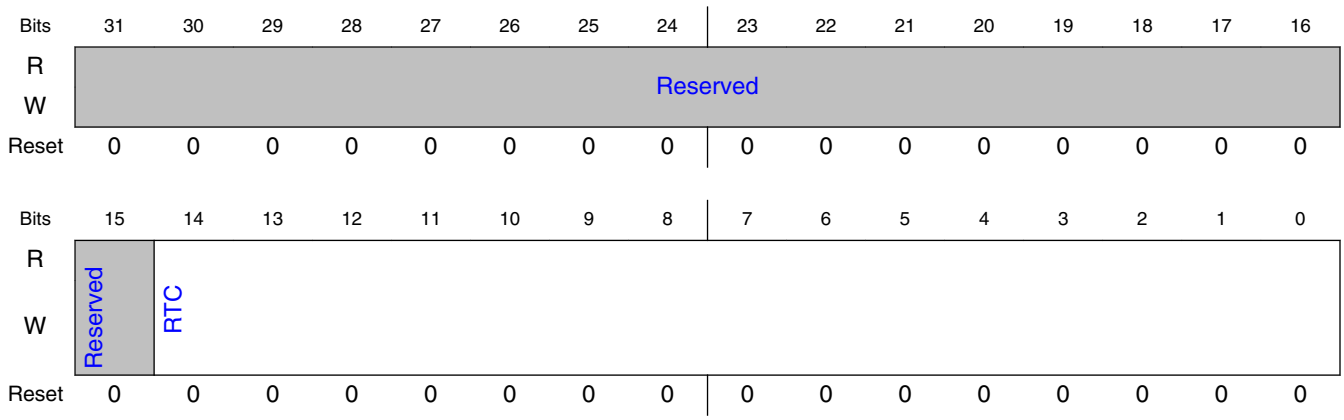
6.4.5.1.5 SNVS_HP Real Time Counter MSB Register (HPRTCMR)

The SNVS_HP Real Time Counter MSB register contains the 15 most-significant bits of the HP Real Time Counter. This is *not* a privileged write register.

6.4.5.1.5.1 Offset

Register	Offset
HPRTCMR	24h

6.4.5.1.5.2 Diagram



6.4.5.1.5.3 Fields

Field	Description
31-15 —	Reserved
14-0 RTC	HP Real Time Counter The most-significant 15 bits of the RTC. This register can be programmed only when RTC is not active (RTC_EN bit is not set).

6.4.5.1.6 SNVS_HP Real Time Counter LSB Register (HPRTCLR)

The SNVS_HP Real Time Counter LSB register contains the 32 least-significant bits of the HP real time counter. This is *not* a privileged write register.

6.4.5.1.6.1 Offset

Register	Offset
HPRTCLR	28h

6.4.5.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RTC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RTC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4.5.1.6.3 Fields

Field	Description
31-0 RTC	HP Real Time Counter least-significant 32 bits. This register can be programmed only when RTC is not active (RTC_EN bit is not set).

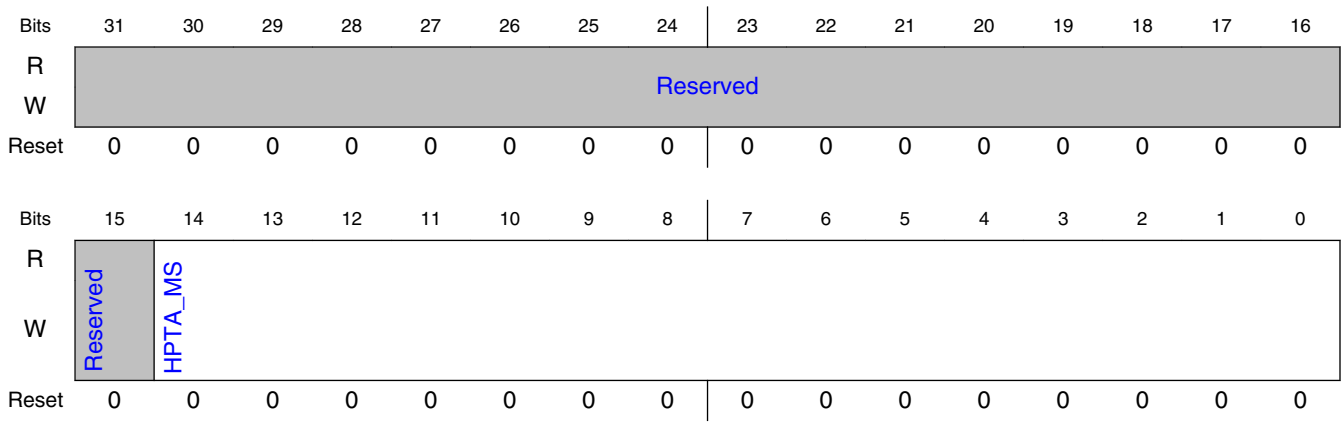
6.4.5.1.7 SNVS_HP Time Alarm MSB Register (HPTAMR)

The SNVS_HP Time Alarm MSB register contains the most-significant bits of the SNVS_HP Time Alarm value. This is *not* a privileged write register.

6.4.5.1.7.1 Offset

Register	Offset
HPTAMR	2Ch

6.4.5.1.7.2 Diagram



6.4.5.1.7.3 Fields

Field	Description
31-15 —	Reserved
14-0 HPTA_MS	HP Time Alarm, ms HP Time Alarm, most-significant 15 bits. This register can be programmed only when HP time alarm is disabled (HPTA_EN bit is not set).

6.4.5.1.8 SNVS_HP Time Alarm LSB Register (HPTALR)

The SNVS_HP Time Alarm LSB register contains the 32 least-significant bits of the SNVS_HP Time Alarm value. This is *not* a privileged write register.

6.4.5.1.8.1 Offset

Register	Offset
HPTALR	30h

6.4.5.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HPTA_LS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HPTA_LS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4.5.1.8.3 Fields

Field	Description
31-0 HPTA_LS	HP Time Alarm, ls HP Time Alarm, 32 least-significant bits. This register can be programmed only when HP time alarm is disabled (HPTA_EN bit is not set).

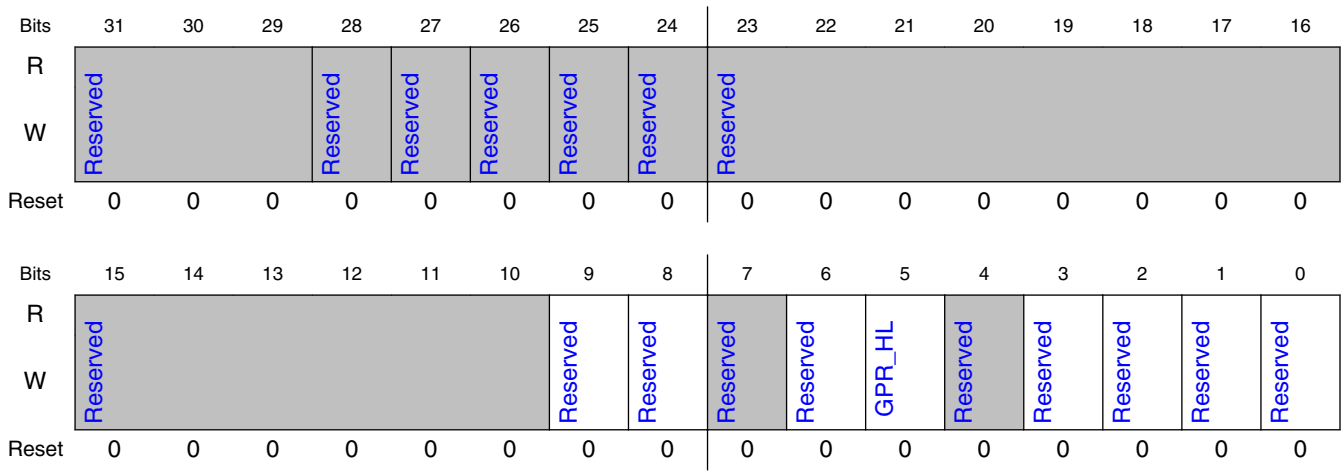
6.4.5.1.9 SNVS_LP Lock Register (LPLR)

The SNVS_LP Lock Register contains lock bits for the SNVS_LP registers. This is a privileged write register.

6.4.5.1.9.1 Offset

Register	Offset
LPLR	34h

6.4.5.1.9.2 Diagram



6.4.5.1.9.3 Fields

Field	Description
31-29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25 —	Reserved
24 —	Reserved
23-10 —	Reserved
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 —	Reserved
5	General Purpose Register Hard Lock

Table continues on the next page...

Field	Description
GPR_HL	When set, prevents any writes to the GPR. Once set, this bit can only be reset by the LP POR. 0 - Write access is allowed. 1 - Write access is not allowed.
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 —	Reserved

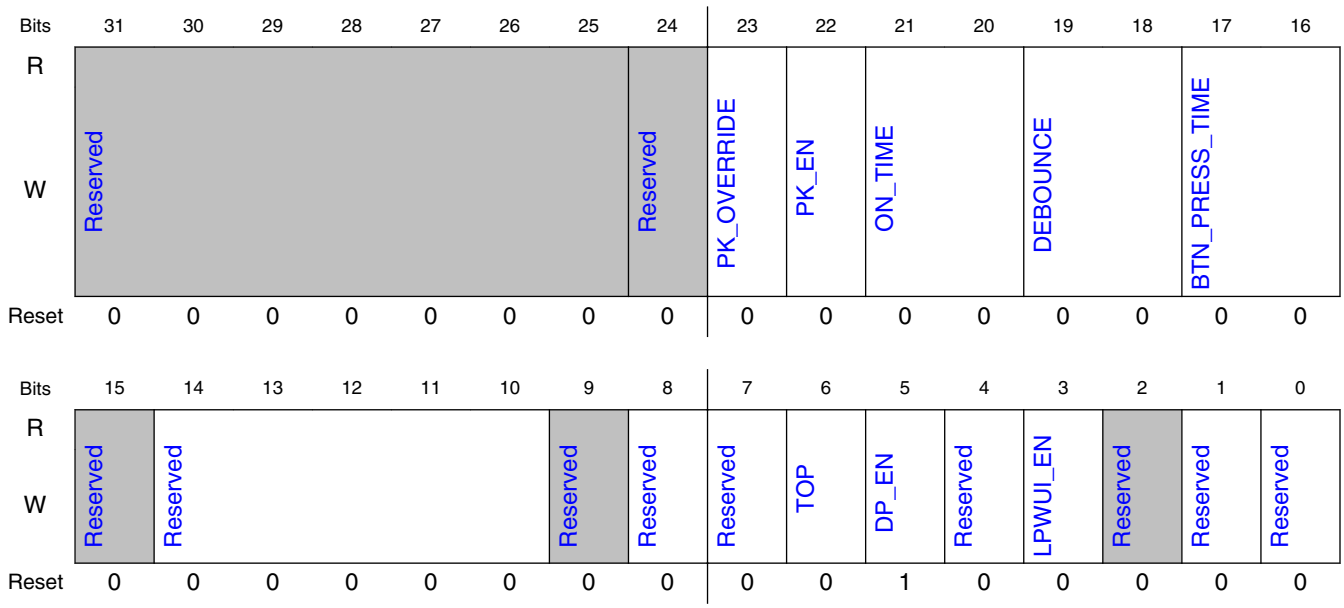
6.4.5.1.10 SNVS_LP Control Register (LPCR)

The SNVS_LP Control Register contains various control bits of the LP section of SNVS. This is a privileged write register.

6.4.5.1.10.1 Offset

Register	Offset
LPCR	38h

6.4.5.1.10.2 Diagram



6.4.5.1.10.3 Fields

Field	Description
31-25 —	Reserved
24 —	Reserved
23 PK_OVERRIDE	PMIC On Request Override The value written to PK_OVERRIDE will be asserted on output signal snvs_lp_pk_override. That signal is used to override the IOMUX control for the PMIC I/O pad.
22 PK_EN	PMIC On Request Enable The value written to PK_EN will be asserted on output signal snvs_lp_pk_en. That signal is used to turn off the pullup/pulldown circuitry in the PMIC I/O pad.
21-20 ON_TIME	ON_TIME Config The ON_TIME field is used to configure the period of time after BTN is asserted before pmic_en_b is asserted to turn on the SoC power. 00: 500msec off->on transition time 01: 50msec off->on transition time 10: 100msec off->on transition time 11: 0msec off->on transition time
19-18 DEBOUNCE	Debounce Time Config This field configures the amount of debounce time for the BTN input signal. 00: 50msec debounce 01: 100msec debounce

Table continues on the next page...

Field	Description
	10: 500msec debounce 11: 0msec debounce
17-16 BTN_PRESS_TIME	Button Press Time Out config This field configures the button press time out values for the PMIC Logic. 00 : 5 secs 01 : 10 secs 10 : 15 secs 11 : long press disabled (pmic_en_b will not be asserted regardlessof how long BTN is asserted)
15 —	Reserved
14-10 —	Reserved
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 TOP	Turn off System Power Asserting this bit causes a signal to be sent to the Power Management IC to turn off the system power. This bit will clear once power is off. This bit is only valid when the Dumb PMIC is enabled. 0 - Leave system power on. 1 - Turn off system power.
5 DP_EN	Dumb PMIC Enabled When set, software can control the system power. When cleared, the system requires a Smart PMIC to automatically turn power off. 0 - Smart PMIC enabled. 1 - Dumb PMIC enabled.
4 —	Reserved
3 LPWUI_EN	LP Wake-Up Interrupt Enable This interrupt line should be connected to the external pin and is intended to inform the external chip about an SNVS_LP event (MC rollover, SRTC rollover, or time alarm). This wake-up signal can be asserted only when the chip (HP section) is powered down, and the LP section is isolated. 0 LP wake-up interrupt is disabled. 1 LP wake-up interrupt is enabled.
2 —	Reserved
1 —	Reserved
0	Reserved

Field	Description
—	

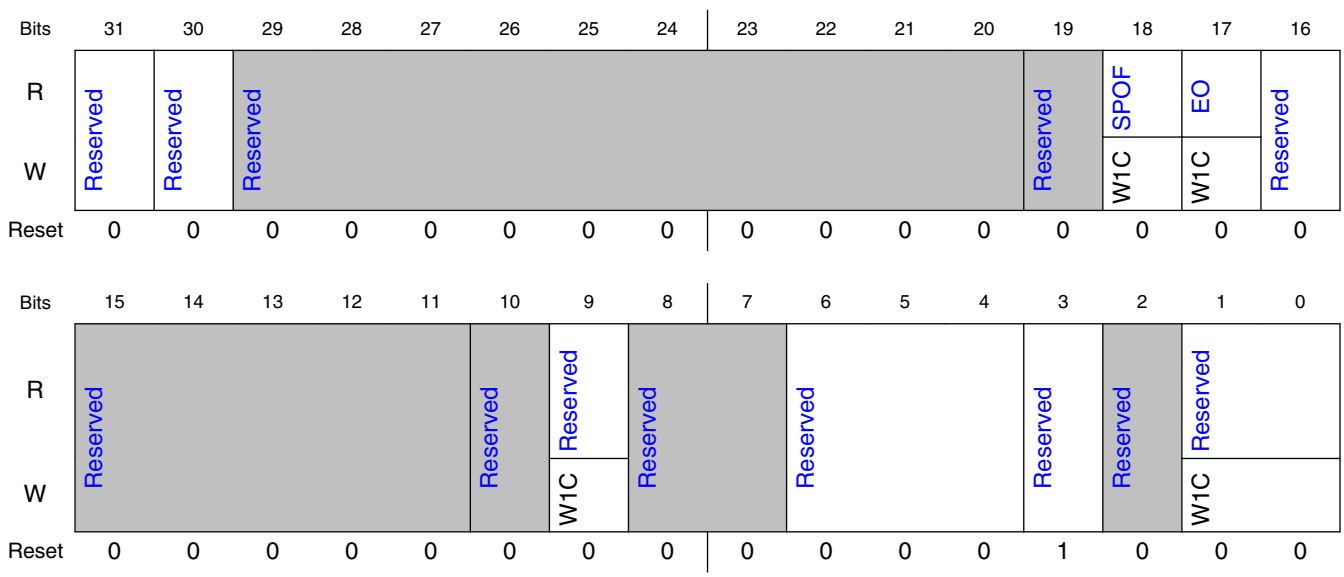
6.4.5.1.11 SNVS_LP Status Register (LPSR)

The SNVS_LP Status Register reflects the internal state and behavior of the SNVS_LP. This is a privileged write register.

6.4.5.1.11.1 Offset

Register	Offset
LPSR	4Ch

6.4.5.1.11.2 Diagram



6.4.5.1.11.3 Fields

Field	Description
31 —	Reserved
30	Reserved

Table continues on the next page...

Field	Description
—	
29-20 —	Reserved
19 —	Reserved
18 SPOF	Set Power Off The SPO bit is set when the power button is pressed longer than the configured debounce time. Writing to the SPO bit will clear the set_pwr_off_irq interrupt. 0 - Set Power Off was not detected. 1 - Set Power Off was detected.
17 EO	Emergency Off This bit is set when a power off is requested. 0 - Emergency off was not detected. 1 - Emergency off was detected.
16 —	Reserved
15-11 —	Reserved
10 —	Reserved
9 —	Reserved
8-7 —	Reserved
6-4 —	Reserved
3 —	Reserved
2 —	Reserved
1-0 —	Reserved

6.4.5.1.12 SNVS_LP General Purpose Register 0 (legacy alias) (LPGPR0_legacy_alias)

See register SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0 - LPGPR3).

6.4.5.1.12.1 Offset

Register	Offset
LPGPR0_legacy_alias	68h

6.4.5.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4.5.1.12.3 Fields

Field	Description
31-0 GPR	General Purpose Register When GPR_SL or GPR_HL bit is set, the register cannot be programmed.

6.4.5.1.13 SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0 - LPGPR3)

The SNVS_LP General Purpose Register is a 128-bit read/write register located in SNVS_LP, which can be used by any application for retaining data during an SoC power-down mode. This is a privileged read/write register. The full GPR register is accessed as 4 32-bit registers located in successive word addresses starting at offset 90h. For backward compatibility with earlier versions of SNVS, LPGPR0 is also aliased at its original offset of 68h. New software should access the GPR register at the preferred offset of 90h.

6.4.5.1.13.1 Offset

Register	Offset
LPGPR0	90h
LPGPR1	94h
LPGPR2	98h
LPGPR3	9Ch

6.4.5.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4.5.1.13.3 Fields

Field	Description
31-0 GPR	General Purpose Register When GPR_SL or GPR_HL bit is set, the register cannot be programmed.

6.4.5.1.14 SNVS_HP Version ID Register 1 (HPVIDR1)

The SNVS_HP Version ID Register 1 is a non-privileged read-only register that contains the current version of the SNVS. The version consists of a module ID, a major version number, and a minor version number.

6.4.5.1.14.1 Offset

Register	Offset
HPVIDR1	BF8h

6.4.5.1.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IP_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR_REV								MINOR_REV							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1

6.4.5.1.14.3 Fields

Field	Description
31-16 IP_ID	Block ID SNVS block ID
15-8 MAJOR_REV	Major Version Number SNVS block major version number
7-0 MINOR_REV	Minor Version Number SNVS block minor version number

6.4.5.1.15 SNVS_HP Version ID Register 2 (HPVIDR2)

The SNVS_HP Version ID Register 2 is a non-privileged read-only register that indicates the current version of the SNVS. Version ID register 2 consists of the following fields: integration options, ECO revision, and configuration options.

6.4.5.1.15.1 Offset

Register	Offset
HPVIDR2	BFCh

6.4.5.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IP_ERA								Reserved							
W																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ECO_REV								Reserved							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

6.4.5.1.15.3 Fields

Field	Description
31-24 IP_ERA	IP Era 00h - Era 1 or 2 03h - Era 3 04h - Era 4 05h - Era 5 06h - Era 6
23-16 —	Reserved
15-8 ECO_REV	ECO Revision SNVS ECO Revision The engineering change order revision number for this release of SNVS.
7-0 —	Reserved

6.5 System Reset Controller (SRC)

6.5.1 Overview

The System Reset Controller (SRC) is responsible for the generation of all the system reset signals and boot argument latching.

6.5.1.1 Features

The SRC includes the following features.

- Receives and handles the resets from all the reset sources
- Resets the appropriate domains based upon the resets sources and the nature of the reset
- Latches the SRC_BOOT_MODE pins and common configuration signals from the internal fuse

6.5.2 Functional Description

The reset controller determines the source and the type of reset, such as POR, COLD, and performs the necessary reset qualification and stretching sequences. Based on the type of reset, the reset logic generates the reset sequence for the entire IC. Whenever the chip is powered on, the reset is issued through SRC_ONOFF signal and the entire chip is reset.

6.5.2.1 Reset and Power-up Flow

The chip presumes the following reset and power-up flow:

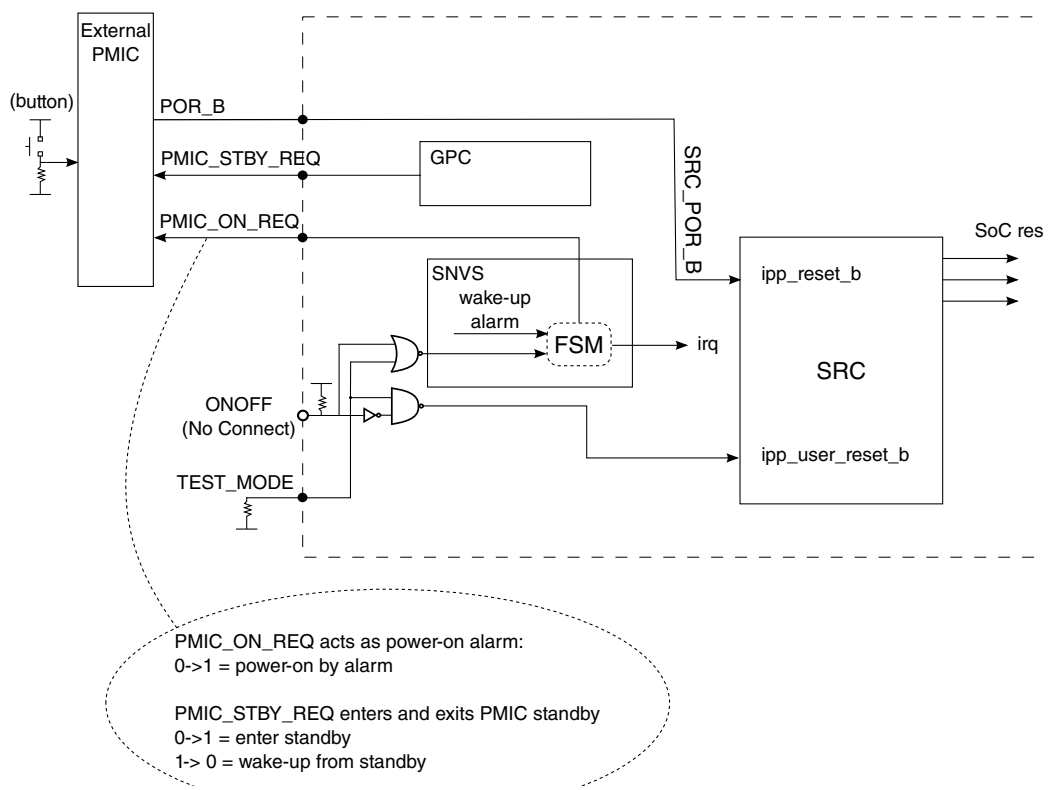


Figure 6-28. Chip reset scheme under external PMIC control

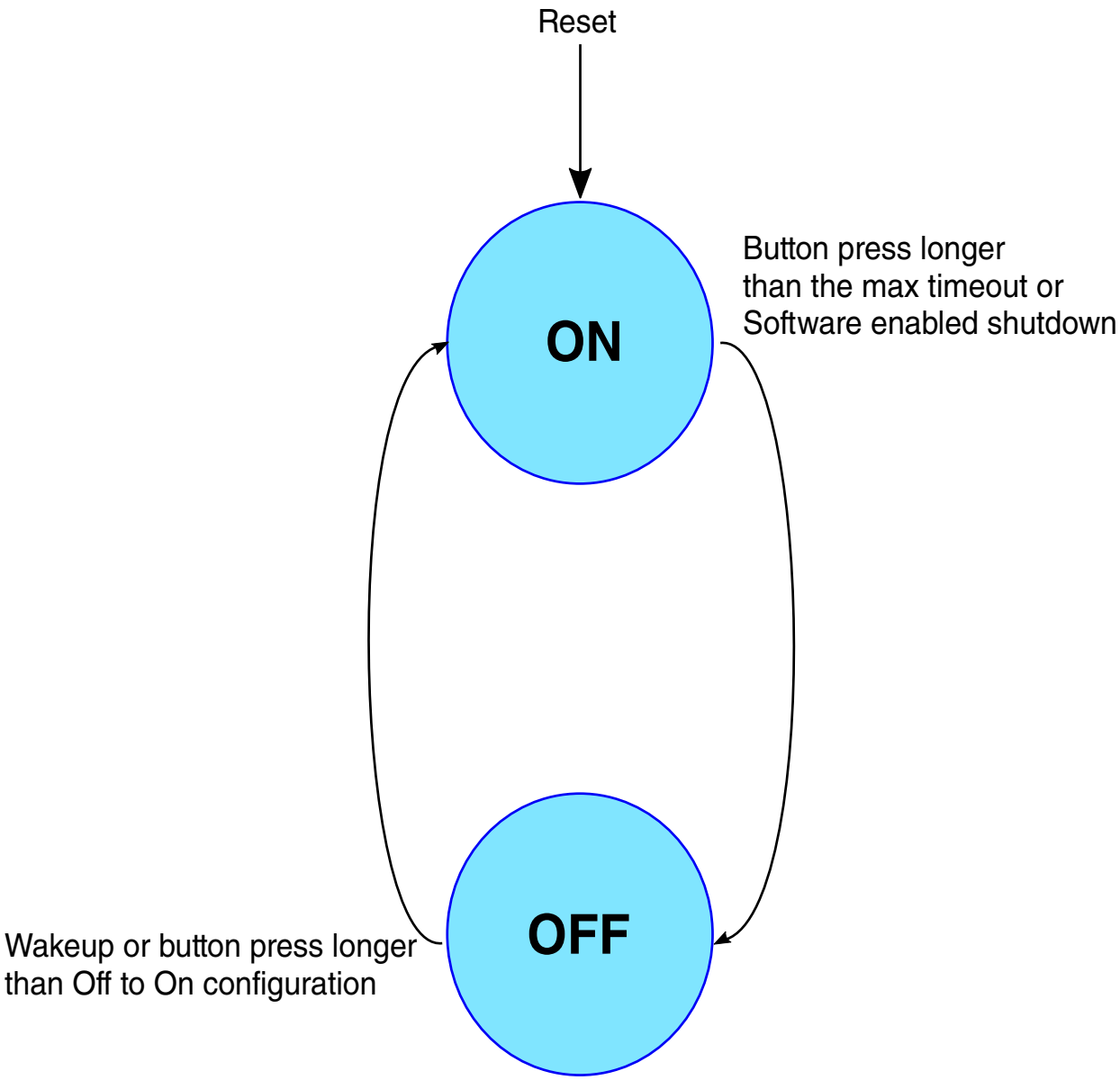


Figure 6-29. Chip on/off state flow diagram

6.5.2.2 Finite-State Machine (FSM)

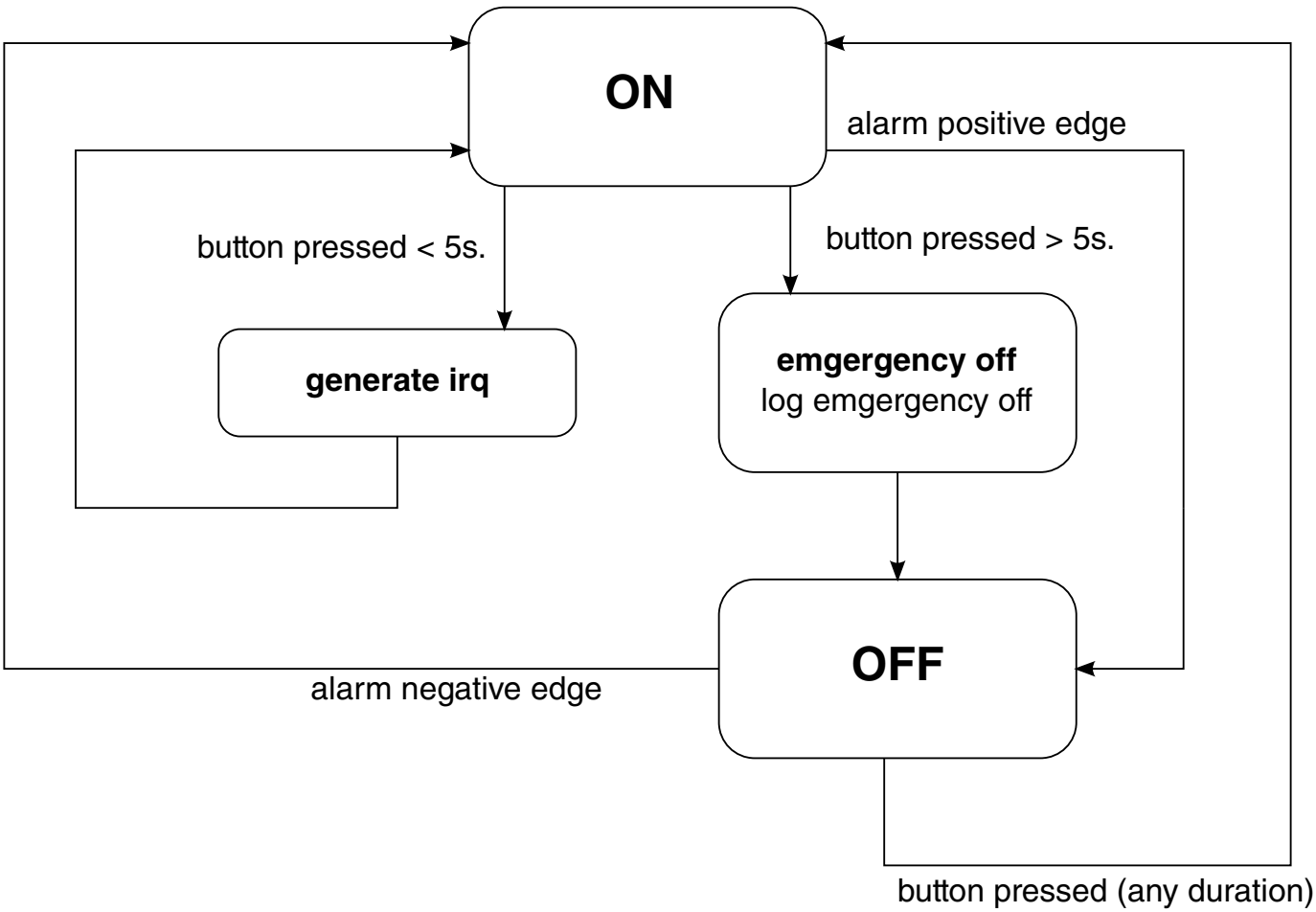


Figure 6-30. FSM

6.5.2.3 Power mode transitions

Table 6-41. Power mode transitions

Power mode	Configuration with external PMIC	Configuration with internal PMIC
ON, first time	1. SoC power supply is connected to SNVS. 2. When button is pressed, PMIC powers on.	1. SoC power supply is connected to SNVS. 2. When button is pressed, 'state' goes ON, PMIC_ON_REQ goes '1'. 3. External regulator is enabled.
Normal ON to OFF, by button	1. Button is pressed for a short duration on the external PMIC. 2. Interrupt request (irq) is sent to SoC from external PMIC. 3. SoC is programming PMIC for power off when standby is asserted.	1. SoC button is pressed for a short duration of 50ms/100ms/500ms, which can be selected by configuring SNVS_LP register “SNVS_REG_LPCR” bit[19:18]. 2. Interrupt request (irq) is sent to SoC from FSM.

Table continues on the next page...

Table 6-41. Power mode transitions (continued)

Power mode	Configuration with external PMIC	Configuration with internal PMIC
	4. In CCM STOP mode, Standby is asserted, PMIC gates SoC supplies.	3. If SNVS_REG_LPCR bit[6]), is configured to be 1, then after about 16~18 cycles (32K clock), PMIC_ON_REQ will be 0. 5. External regulator goes OFF.
Emergency ON to OFF, by button	1. Button is pressed for an extended time on the external PMIC. 2. PMIC is powering off.	1. Button is pressed for longer than 5 seconds on the SoC. 2. FSM validates button pressed for 5 seconds. 3. Emergency power off is logged, PMIC_ON_REQ goes '0', alarm_mask goes '1'. 4. External regulator goes OFF.
OFF to ON, by button	1. Button is pressed on the external PMIC. 2. PMIC powers ON.	1. Button is pressed on the SoC. 2. PMIC_ON_REQ goes '1', alarm_mask goes '0'. 3. External regulator powers ON.
OFF to ON, by timer alarm	1. Timer alarm in SNVS is programmed by software before SoC goes OFF. 2. SoC enters OFF mode. 3. Upon timer limit, wake up alarm goes '0'. PMIC_ON_REQ goes '1'. 4. PMIC receives assertion of PMIC_ON_REQ and wakes up.	1. Timer alarm in SNVS is programmed by software before SoC goes OFF. 2. SoC enters OFF mode. 3. Upon timer limit, wake up alarm goes '0'. PMIC_ON_REQ goes '1'. 4. External regulator is enabled by PMIC_ON_REQ = 1.

6.5.2.4 Reset Control

This section details the reset control of this device.

6.5.2.4.1 Reset inputs and outputs

The reset control logic receives reset requests from all potential reset sources. All the immediate sources of reset are directly passed to the reset stretching block, whereas the resets requiring qualification are passed on to the reset qualification logic before they are sent to the reset stretching block.

All reset inputs and outputs are described in the following figure:

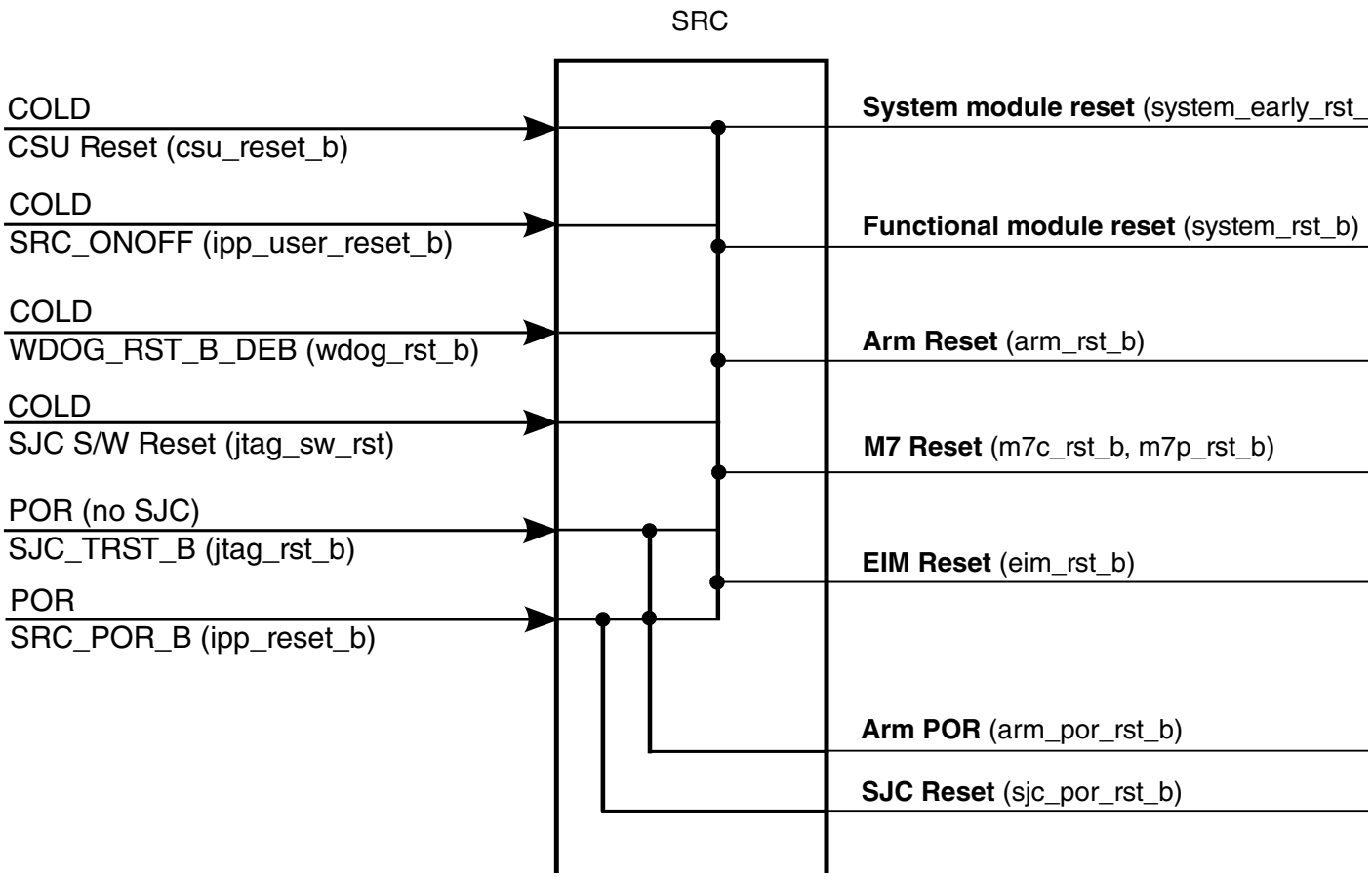


Figure 6-31. SRC inputs and outputs

The reset types and modules they affect are shown in Table 6-42. As there is no chip POR, the POR_B is used to reset the entire chip including test logic and JTAG modules.

NOTE

All resets are expected to be active low except jtag_sw_rst.

Table 6-42. SRC reset functionality

SoC Modules	POR	COLD
System modules (PLLs, fuses, etc)	yes	yes
Functional modules	yes	yes
Arm	yes	yes
Arm SoC	yes	yes
M7 Core	yes	yes
M7 Platform	yes	yes
Arm POR	yes	no
Arm debug	yes	no
SJC	yes	no
SRTC	yes	no

The reset priorities are POR (strongest) and COLD (weakest). If a stronger reset is asserted during the sequence of a weaker reset, then the weaker sequence will be overridden, and the stronger reset sequence will commence. There is no priority within a reset type (POR, etc). If a reset is asserted during the reset sequence of the same type, the reset sequence will be interrupted and restarted.

The following lists the functionality of each of these reset outputs:

- system_early_rst_b - Resets the system modules that need to start first as CCM, OCOTP_CTRL, FUSEBOX, etc.
- system_rst_b - Resets functional modules
- arm_rst_b - Resets Arm module (on regular system reset)
- arm_por_rst_b - Resets Arm POR input
- arm_soc_rst_b - Reset for Arm SOC
- m7c_rst_b - Reset for M7 core
- m7p_rst_b - Reset for M7 platform
- arm_dbg_rst_b - Reset debug logic of Arm
- test_logic_rst_b - Reset test logic (IOMUXC, DAP)
- sjc_por_rst_b - Reset to SJC
- srtc_rst_b - Resets SRTC

NOTE

It is assumed that each reset source will deassert after its assertion, either due to reset generated to the system from SRC, or by negation of the reset source (if it came from an external source to the chip). In the latter case, the reset source is assumed to be held for at least 2 XTALI clocks so it can be sampled by SRC.

6.5.2.4.2 Reset Handling

6.5.2.4.2.1 POR (SRC_POR_B)

SRC_POR_B is an external reset signal of the SRC module. When the chip is powered up, the reset signal is passed through the POR_B pin indicating power-up sequence. The SRC resets the entire chip including the JTAG (SJC) module. All SRC registers will be reset during the POR sequence.

As soon as SRC_POR_B occurs, all resets are asserted and the entire chip is reset by SRC. The SRC_POR_B is stretched for 2 XTALI cycles and the stretching sequence takes place after 2 XTALI clocks of POR_B pin deassertion.

The srtc_rst_b signal is deasserted together with SRC_POR_B signal. The output are also deasserted after the stretching of SRC_POR_B has deasserted.

The sjc_por_rst_b signal is deasserted together with SRC_POR_B signal. The output is also deasserted after the stretching of SRC_POR_B has deasserted.

After the above resets deassert, system_early_rst_b reset is deasserted after 2 XTALI clocks. The system_early_rst_b is used for the CCM and PLL-IPs to start generating PLL clock ouputs and the system root clocks.

When the system root clocks are ready, the CCM will assert system_clk_ready signal. This signal is generated during the start sequence in the CCM and it involves the preparation of the PLLs to generate clock roots for functional operation.

SRC then enables OCOTP_CTRL and fusebox clocks, so that fuses can be loaded to OCOTP_CTRL.

- SRC will prepare the boot information
- After 8 ipg cycles, resets to all modules will be de-asserted
- After 8 ipg cycles, system clocks will be enabled (en_system_clk).

6.5.2.4.2.2 COLD RESET

The sequence is similar to SRC_POR_B except the memory repair operation is not performed.

After the reset source deasserts, system_early_rst_b reset is deasserted after at least 2 XTALI clocks. The system_early_rst_b is used for the CCM and PLL-IPs to start generating PLL clock outputs and the system root clocks.

After the system root clocks are ready, the CCM will assert system_clk_ready signal. This signal is generated during the start sequence in the CCM and it involves the preparation of the PLLs to generate clock roots for functional operation. See CCM for more information.

After system_clk_ready arrives at the SRC, it will enable OCOTP_CTRL and fusebox clocks, so that fuses can be loaded to OCOTP_CTRL. OCOTP_CTRL will notify with iim_ready_flag when the fusebox loading finishes.

- SRC will prepare the boot information
- After 8 ipg cycles resets to all modules will be deasserted
- After 8 ipg cycles, system clocks will be enabled (en_system_clk).

6.5.2.5 Parallel Reset Requests

SRC will follow the following rules in the case of parallel reset requests:

1. The order of strength of resets is POR - strongest, COLD - weakest
2. If a stronger reset is asserted during weaker reset sequence, then the stronger reset will take over and the stronger reset process will commence. The following cases fall into this category:
   - POR reset request in the middle of cold reset process - the cold will be stopped and the POR sequence will start.
3. If a weaker reset is asserted during stronger reset sequence, then the stronger reset sequence will continue without interference. If at the end of the stronger reset process the weaker request is still asserted then the weaker sequence will commence. The following cases fall into this category:
   - COLD reset requests in the middle of POR reset process - the POR process will continue without interference.
4. If a similar reset request is asserted during the process of reset handling, then the process of reset handling will start over (with the same process). The following cases fall into this category:
   - POR reset request in the middle of POR reset process - the POR process will start over.
   - COLD reset request in the middle of COLD reset process - the COLD process will start over.

6.5.2.6 Boot Mode Control

6.5.2.6.1 BOOT_MODE Pin Latching

The exact boot sequence is controlled by the values of the BOOT_MODE pins on this device.

The value of the BOOT_MODE pins will be latched on de-assertion of POR reset. After latching, the values of the BOOT_MODE pins are used to determine the booting options of the core as described in the SRC_SBMRx registers.

The boot mode general purpose bits can be provided to the SRC from either e-fuses or GPIO signals. The gpio_bt_sel e-fuse defines the source to be used to derive the boot information. When gpio_bt_sel is set, e-fuses are used. When cleared, GPIO signals are used.

The boot information is provided in SRC_SBMR1 and SRC_SBMR2 registers. The figure below shows the selection of boot mode information.

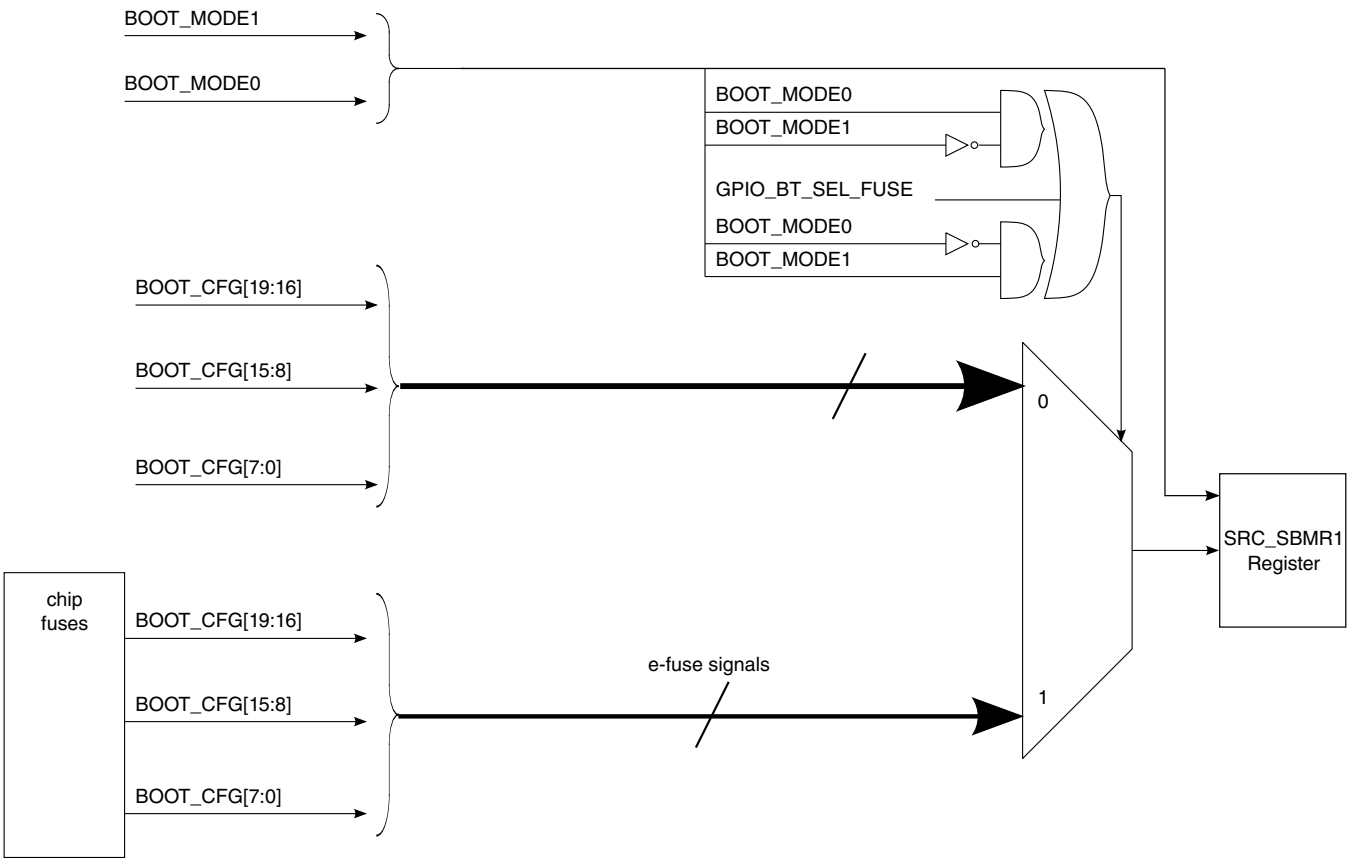


Figure 6-32. Boot mode information

6.5.3 External Signals

The following table describes the external signals of SRC.

6.5.4 Initialization

6.5.4.1 Power-On Reset and power sequencing

The SRC module generates an internal POR_B signal that is logically AND'ed with any externally applied SRC_POR_B signal. The internal POR_B signal will be held low until all of the following conditions are met:

- 4ms after the external power supply VDDHIGH_IN is valid
- 1ms after the VDD_SOC_CAP supply is valid

The 4ms and 1ms delays are derived from counting the 32 kHz RTC clock cycles; the accuracy depends on the accuracy of the RTC.

6.5.4.1.1 External POR using SRC_POR_B

If the external SRC_POR_B signal is used to control the processor POR, SRC_POR_B must remain low (asserted) until the VDD_ARM_CAP and VDD_SOC_CAP supplies are stable.

6.5.4.1.2 Internal POR

If the external SRC_POR_B signal is not used (always held high or left unconnected), the processor defaults to the internal POR function (PMU controls generation of the POR based on the power supplies).

If the internal POR function is used, the following power supply requirements must be met:

- VDD_ARM_IN and VDD_SOC_IN may be supplied from the same source, or
- VDD_SOC_IN can be supplied before VDD_ARM_IN with a maximum delay of 1 ms.

6.5.5 SRC Memory Map/Register Definition

SRC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3039_0000	SRC Reset Control Register (SRC_SCR)	32	R/W	0000_00A0h	6.5.5.1/916
3039_0004	A53 Reset Control Register (SRC_A53RCR0)	32	R/W	000A_0000h	6.5.5.2/918
3039_0008	A53 Reset Control Register (SRC_A53RCR1)	32	R/W	0000_0001h	6.5.5.3/923
3039_000C	M7 Reset Control Register (SRC_M7RCR)	32	R/W	0000_00A8h	6.5.5.4/926
3039_0018	SUPERMIX Reset Control Register (SRC_SUPERMIX_RCR)	32	R/W	0000_0000h	6.5.5.5/928
3039_001C	AUDIOMIX Reset Control Register (SRC_AUDIOMIX_RCR)	32	R/W	0000_0000h	6.5.5.6/930
3039_0020	USB PHY1 Reset Control Register (SRC_USBPHY1_RCR)	32	R/W	0000_0000h	6.5.5.7/931
3039_0024	USB PHY2 Reset Control Register (SRC_USBPHY2_RCR)	32	R/W	0000_0000h	6.5.5.8/933
3039_0028	MLMIX Reset Control Register (SRC_MLMIX_RCR)	32	R/W	0000_0000h	6.5.5.9/934
3039_002C	PCIE PHY Reset Control Register (SRC_PCIEPHY_RCR)	32	R/W	0000_000Ah	6.5.5.10/936
3039_0030	HDMI Reset Control Register (SRC_HDMI_RCR)	32	R/W	0000_0000h	6.5.5.11/939
3039_0034	MEDIAMIX Reset Control Register (SRC_MEDIA_RCR)	32	R/W	0000_0000h	6.5.5.12/940
3039_0038	GPU2D Reset Control Register (SRC_GPU2D_RCR)	32	R/W	0000_0000h	6.5.5.13/942

Table continues on the next page...

SRC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3039_003C	GPU3D Reset Control Register (SRC_GPU3D_RCR)	32	R/W	0000_0000h	6.5.5.14/943
3039_0040	GPU Reset Control Register (SRC_GPU_RCR)	32	R/W	0000_0000h	6.5.5.15/945
3039_0044	VPU Reset Control Register (SRC_VPU_RCR)	32	R/W	0000_0000h	6.5.5.16/946
3039_0048	VPU G1 Reset Control Register (SRC_VPU_G1_RCR)	32	R/W	0000_0000h	6.5.5.17/948
3039_004C	VPU G2 Reset Control Register (SRC_VPU_G2_RCR)	32	R/W	0000_0000h	6.5.5.18/949
3039_0050	VPU VC8000E Reset Control Register (SRC_VPUVC8KE_RCR)	32	R/W	0000_0000h	6.5.5.19/951
3039_0054	NOC Wrapper Reset Control Register (SRC_NOC_RCR)	32	R/W	0000_0000h	6.5.5.20/952
3039_0058	SRC Boot Mode Register 1 (SRC_SBMR1)	32	R	0000_0000h	6.5.5.21/954
3039_005C	SRC Reset Status Register (SRC_SRSR)	32	R/W	0000_0001h	6.5.5.22/954
3039_0068	SRC Interrupt Status Register (SRC_SISR)	32	R/W	0000_0000h	6.5.5.23/957
3039_006C	SRC Interrupt Mask Register (SRC_SIMR)	32	R/W	0000_03FFh	6.5.5.24/959
3039_0070	SRC Boot Mode Register 2 (SRC_SBMR2)	32	R	0000_0000h	6.5.5.25/961
3039_0074	SRC General Purpose Register 1 (SRC_GPR1)	32	R/W	0000_0000h	6.5.5.26/963
3039_0078	SRC General Purpose Register 2 (SRC_GPR2)	32	R/W	0000_0000h	6.5.5.27/963
3039_007C	SRC General Purpose Register 3 (SRC_GPR3)	32	R/W	0000_0000h	6.5.5.28/963
3039_0080	SRC General Purpose Register 4 (SRC_GPR4)	32	R/W	0000_0000h	6.5.5.29/964
3039_0084	SRC General Purpose Register 5 (SRC_GPR5)	32	R/W	0000_0000h	6.5.5.30/964
3039_0088	SRC General Purpose Register 6 (SRC_GPR6)	32	R/W	0000_0000h	6.5.5.31/965
3039_008C	SRC General Purpose Register 7 (SRC_GPR7)	32	R/W	0000_0000h	6.5.5.32/965
3039_0090	SRC General Purpose Register 8 (SRC_GPR8)	32	R/W	0000_0000h	6.5.5.33/965
3039_0094	SRC General Purpose Register 9 (SRC_GPR9)	32	R/W	0000_0000h	6.5.5.34/966
3039_0098	SRC General Purpose Register 10 (SRC_GPR10)	32	R/W	0000_0000h	6.5.5.35/966
3039_1000	SRC DDR Controller Reset Control Register (SRC_DDRC_RCR)	32	R/W	0000_000Fh	6.5.5.36/967
3039_1008	HDMIPHY Reset Control Register (SRC_HDMIPHY_RCR)	32	R/W	0000_0000h	6.5.5.37/969
3039_100C	MIPI PHY1 Reset Control Register (SRC_MIPIPHY1_RCR)	32	R/W	0000_0000h	6.5.5.38/970
3039_1010	MIPI PHY2 Reset Control Register (SRC_MIPIPHY2_RCR)	32	R/W	0000_0000h	6.5.5.39/972
3039_1014	HSIO Reset Control Register (SRC_HSIO_RCR)	32	R/W	0000_0000h	6.5.5.40/973
3039_1018	MEDIAMIX ISP and Dewarp Reset Control Register (SRC_MEDIAISPDWP_RCR)	32	R/W	0000_0000h	6.5.5.41/975

6.5.5.1 SRC Reset Control Register (SRC_SCR)

The reset control register (SCR), contains bits that control operation of the reset controller.

Address: 3039_0000h base + 0h offset = 3039_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DOM_EN	LOCK	Reserved		DOMAIN3	DOMAIN2	DOMAIN1	DOMAIN0	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								MASK_TEMPSENSE_RESET				Reserved			
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

SRC_SCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register

Table continues on the next page...

SRC_SCR field descriptions (continued)

Field	Description
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–8 -	This field is reserved. Reserved
7–4 MASK_ TEMPSENSE_ RESET	Mask tempsense_reset source. If these 4 bits are coded from A to 5 then, the tempsense_reset input to SRC will be masked and the tempsense_reset will not create a reset to the chip. 0101 tempsense_reset is masked 1010 tempsense_reset is not masked
-	This field is reserved. Reserved

6.5.5.2 A53 Reset Control Register (SRC_A53RCR0)

The A53 Reset Control Register (A53RCR), contains bits that control the A53 reset generation.

Address: 3039_0000h base + 4h offset = 3039_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DOM_EN	LOCK	Reserved		DOMAIN3	DOMAIN2	DOMAIN1	DOMAIN0	Reserved		A53_L2RESET	A53_SOC_DBG_RESET	MASK_WDOG1_RST			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	A53_ETM_RESET3	A53_ETM_RESET2	A53_ETM_RESET1	A53_ETM_RESET0	A53_DBG_RESET3	A53_DBG_RESET2	A53_DBG_RESET1	A53_DBG_RESET0	A53_CORE_RESET3	A53_CORE_RESET2	A53_CORE_RESET1	A53_CORE_RESET0	A53_CORE_POR_RESET3	A53_CORE_POR_RESET2	A53_CORE_POR_RESET1	A53_CORE_POR_RESET0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_A53RCR0 field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–22 -	This field is reserved. Reserved
21 A53_L2RESET	Software reset for A53 Snoop Control Unit (SCU). **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert SCU reset 1 assert SCU reset
20 A53_SOC_DBG_RESET	Software reset for system level debug reset. It initializes the shared Debug APB, the CTI, and the CTM. It also causes: • A53_dbgreset[3:0] and A53_etmreset[3:0] to be asserted • debug logic in the processor power domain and in the debug power domain to be reset

Table continues on the next page...

SRC_A53RCR0 field descriptions (continued)

Field	Description
	NOTE: This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert system level debug reset 1 assert system level debug reset
19–16 MASK_WDOG1_RST	Mask wdog1_rst_b source. If these 4 bits are coded from A to 5 then, the wdog1_rst_b input to SRC will be masked and the wdog1_rst_b will not create a reset to the chip. **NOTE:** During the time the WDOG event is masked using SRC logic, it is likely that the WDOG Reset Status Register (WRSR) bit 1 (which indicates a WDOG timeout event) will get asserted. software / OS developer must prepare for this case. Re-enabling the WDOG is possible, by un-masking it in SRC, though it must be preceded by servicing the WDOG. However, for the case that the event has been asserted, the status bit (WRSR bit-1) will remain asserted, regardless of servicing the WDOG module. (Hardware reset is the only way to cause the de-assertion of that bit). Any other code will be coded to 1010 i.e. wdog1_rst_b is not masked 0101 wdog1_rst_b is masked 1010 wdog1_rst_b is not masked
15 A53_ETM_RESET3	Software reset for core3 ETM only. **NOTE:** This is a self clearing bit. Once it is set to 1, the rest process will begin, and once it finished, this bit will be self-cleared. 0 do not assert core3 ETM reset 1 assert core3 ETM reset
14 A53_ETM_RESET2	Software reset for core2 ETM only. **NOTE:** This is a self clearing bit. Once it is set to 1, the rest process will begin, and once it finished, this bit will be self-cleared. 0 do not assert core2 ETM reset 1 assert core2 ETM reset
13 A53_ETM_RESET1	Software reset for core1 ETM only. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core1 ETM reset 1 assert core1 ETM reset
12 A53_ETM_RESET0	Software reset for core0 ETM only. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core0 ETM reset 1 assert core0 ETM reset
11 A53_DBG_RESET3	Software reset for core3 debug only. It initialize the debug, and breakpoint and watchpoint logic in the core3 processor power domain. It also reset the debug logic for core1 processor, which is in the debug power domain.

Table continues on the next page...

SRC_A53RCR0 field descriptions (continued)

Field	Description
	NOTE: This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core3 debug reset 1 assert core3 debug reset
10 A53_DBG_ RESET2	Software reset for core2 debug only. It initialize the debug, and breakpoint and watchpoint logic in the core2 processor power domain. It also reset the debug logic for core1 processor, which is in the debug power domain. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core2 debug reset 1 assert core2 debug reset
9 A53_DBG_ RESET1	Software reset for core1 debug only. It initialize the debug, and breakpoint and watchpoint logic in the core1 processor power domain. It also reset the debug logic for core1 processor, which is in the debug power domain. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core1 debug reset 1 assert core1 debug reset
8 A53_DBG_ RESET0	Software reset for core0 debug only. It initialize the debug, and breakpoint and watchpoint logic in the core1 processor power domain. It also reset the debug logic for core1 processor, which is in the debug power domain. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core0 debug reset 1 assert core0 debug reset
7 A53_CORE_ RESET3	Software reset for core3 only. It initializes the processor logic in the core1 processor power domains, not including the debug, breakpoint and watchpoint logic. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core3 reset 1 assert core3 reset
6 A53_CORE_ RESET2	Software reset for core2 only. It initializes the processor logic in the core1 processor power domains, not including the debug, breakpoint and watchpoint logic. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core2 reset 1 assert core2 reset

Table continues on the next page...

SRC_A53RCR0 field descriptions (continued)

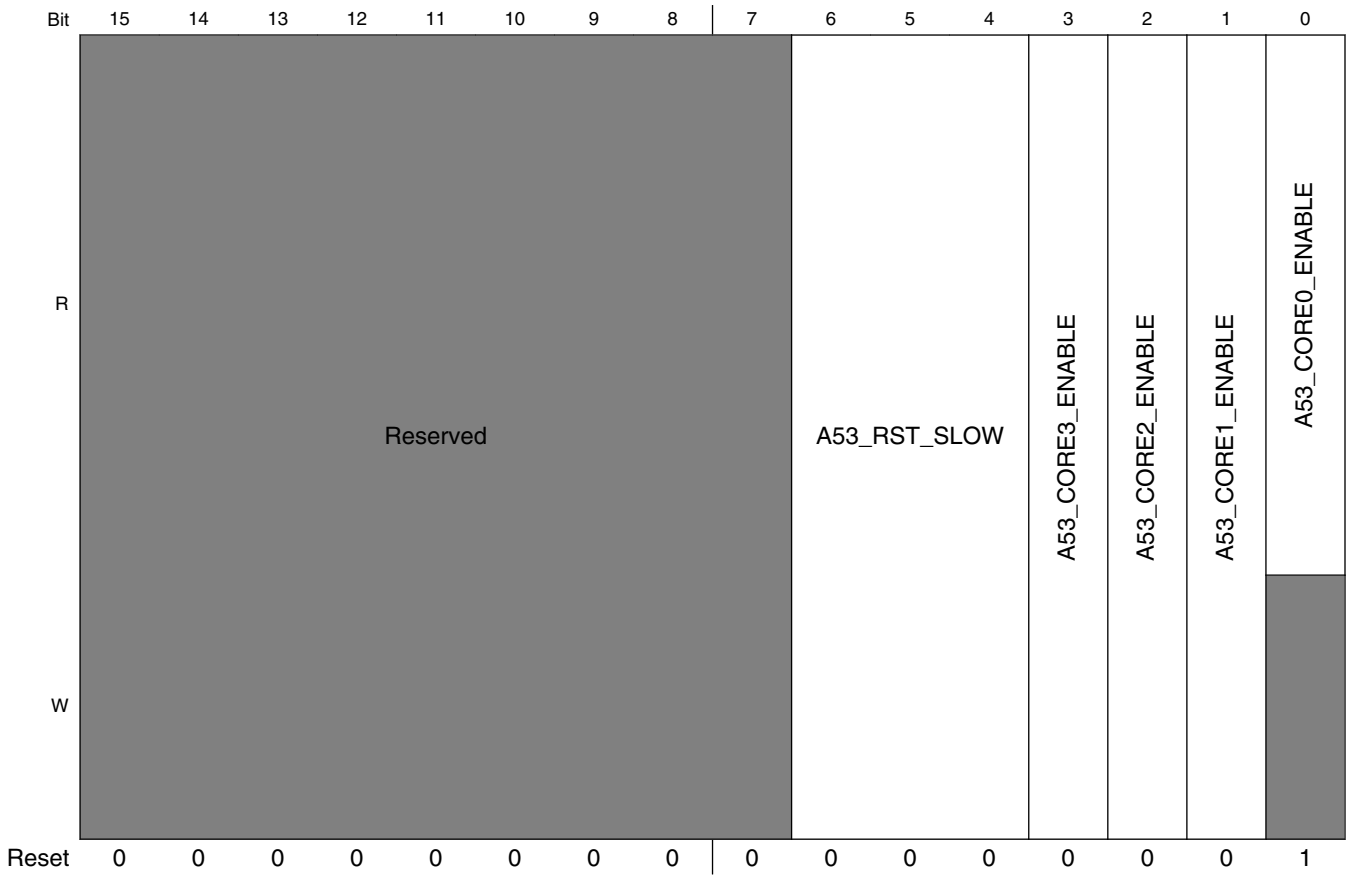
Field	Description
5 A53_CORE_ RESET1	Software reset for core1 only. It initializes the processor logic in the core1 processor power domains, not including the debug, breakpoint and watchpoint logic. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core1 reset 1 assert core1 reset
4 A53_CORE_ RESET0	Software reset for core0 only. It initializes the processor logic in the core0 processor power domains, not including the debug, breakpoint and watchpoint logic. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core0 reset 1 assert core0 reset
3 A53_CORE_ POR_RESET3	POR reset for A53 core3 only. It initializes all the core1 processor logic, including CPU Debug, and breakpoint and watchpoint logic in the core3 processor power domains **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core3 reset 1 assert core3 reset
2 A53_CORE_ POR_RESET2	POR reset for A53 core2 only. It initializes all the core1 processor logic, including CPU Debug, and breakpoint and watchpoint logic in the core2 processor power domains **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core2 reset 1 assert core2 reset
1 A53_CORE_ POR_RESET1	POR reset for A53 core1 only. It initializes all the core1 processor logic, including CPU Debug, and breakpoint and watchpoint logic in the core1 processor power domains **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core1 reset 1 assert core1 reset
0 A53_CORE_ POR_RESET0	POR reset for A53 core0 only. It initializes all the core0 processor logic, including CPU Debug, and breakpoint and watchpoint logic in the core0 processor power domains **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core0 reset 1 assert core0 reset

6.5.5.3 A53 Reset Control Register (SRC_A53RCR1)

The A53 Reset Control Register (A53RCR), contains bits that control the A53 reset generation.

Address: 3039_0000h base + 8h offset = 3039_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DOM_EN	LOCK	Reserved		DOMAIN3	DOMAIN2	DOMAIN1	DOMAIN0	Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



SRC_A53RCR1 field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register

Table continues on the next page...

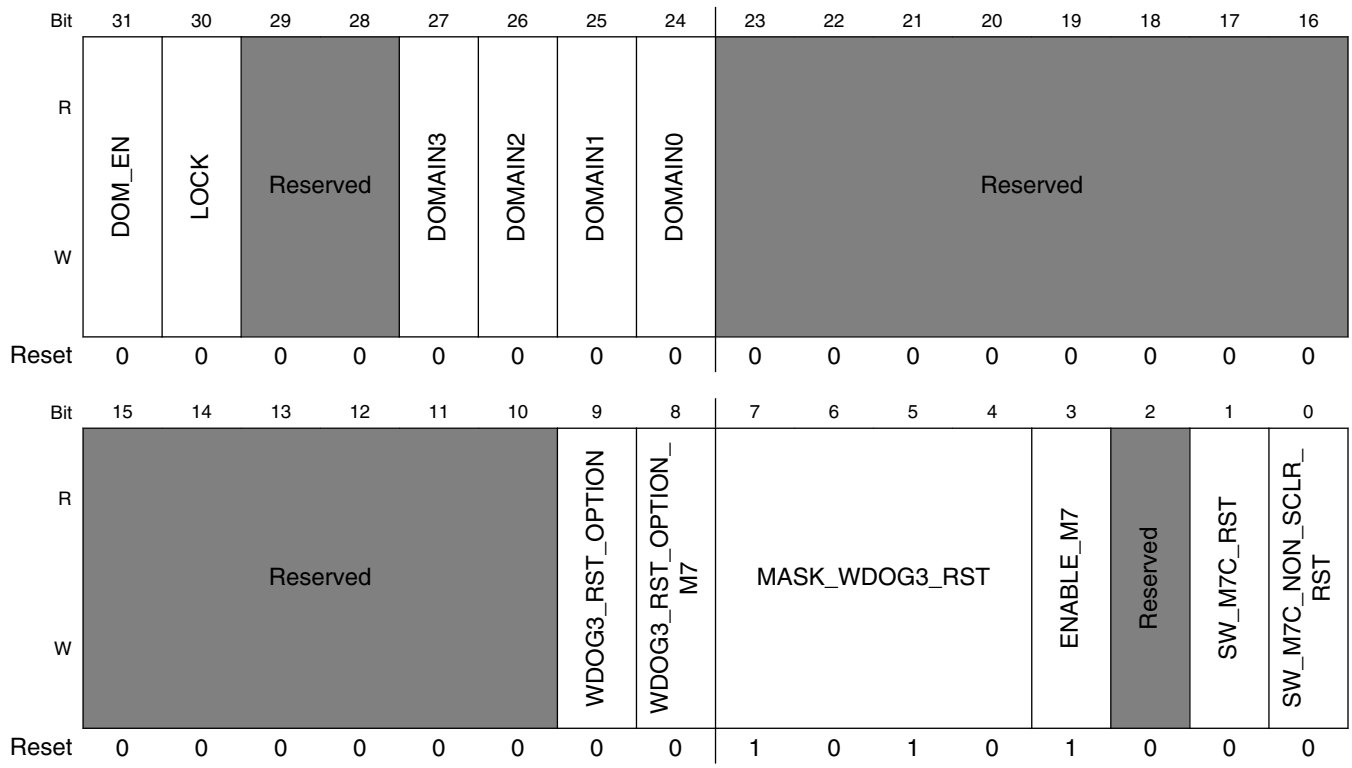
SRC_A53RCR1 field descriptions (continued)

Field	Description
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–7 -	This field is reserved. Reserved
6–4 A53_RST_SLOW	A53_RST_SLOW
3 A53_CORE3_ ENABLE	core 3 enable 0 core3 is disabled 1 core3 is enabled
2 A53_CORE2_ ENABLE	core 2 enable 0 core2 is disabled 1 core2 is enabled
1 A53_CORE1_ ENABLE	core 1 enable 0 core1 is disabled 1 core1 is enabled
0 A53_CORE0_ ENABLE	Always 1, can't be changed.

6.5.5.4 M7 Reset Control Register (SRC_M7RCR)

The M7 Reset Control Register (M7RCR), contains bits that control the M7 reset generation.

Address: 3039_0000h base + Ch offset = 3039_000Ch



SRC_M7RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified

Table continues on the next page...

SRC_M7RCR field descriptions (continued)

Field	Description
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–10 -	This field is reserved. Reserved
9 WDOG3_RST_ OPTION	Wdog3_rst_b option 0 Wdog3_rst_b asserts M7 reset 1 Wdog3_rst_b asserts global reset
8 WDOG3_RST_ OPTION_M7	Wdog3_rst_b option for M7. This bit is only effective when wdog3_rst_option is set to 1. 0 wdgo3_rst_b Reset M7 core only 1 Reset both M7 core and platform
7–4 MASK_WDOG3_ RST	Mask wdog3_rst_b source. If these 4 bits are coded from A to 5 then, the wdog3_rst_b input to SRC will be masked and the wdog3_rst_b will not create a reset to the chip. **NOTE:** During the time the WDOG3 event is masked using SRC logic, it is likely that the WDOG3 Reset Status Register (WRSR) bit 1 (which indicates a WDOG3 timeout event) will get asserted. Software / OS developer must prepare for this case. Re-enabling the WDOG3 is possible, by un-masking it in SRC, though it must be preceded by servicing the WDOG3. However, for the case that the event has been asserted, the status bit (WRSR bit-1) will remain asserted, regardless of servicing the WDOG3 module. (Hardware reset is the only way to cause the de-assertion of that bit). Any other code will be coded to 1010 i.e. wdog3_rst_b is not masked 0101 wdog3_rst_b is masked 1010 wdog3_rst_b is not masked
3 ENABLE_M7	Enable M7 0 M7 is disabled 1 M7 is enabled
2 -	This field is reserved. Reserved

Table continues on the next page...

SRC_M7RCR field descriptions (continued)

Field	Description
1 SW_M7C_RST	Self-clearing SW reset for M7 core **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. Software can determine that the reset has finished once this bit is cleared. Software can also configure SRC to generate interrupt once the software has finished. Please refer to SRC_SISR register for details. 0 do not assert M7 core reset 1 assert M7 core reset
0 SW_M7C_NON_SCLR_RST	Non-self-clearing SW reset for M7 core 0 do not assert M7 core reset 1 assert M7 core reset

6.5.5.5 SUPERMIX Reset Control Register (SRC_SUPERMIX_RCR)

The SUPERMIX Reset Control Register contains bits that control the SUPERMIX reset generation.

Address: 3039_0000h base + 18h offset = 3039_0018h

SRC_SUPERMIX_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set.

Table continues on the next page...

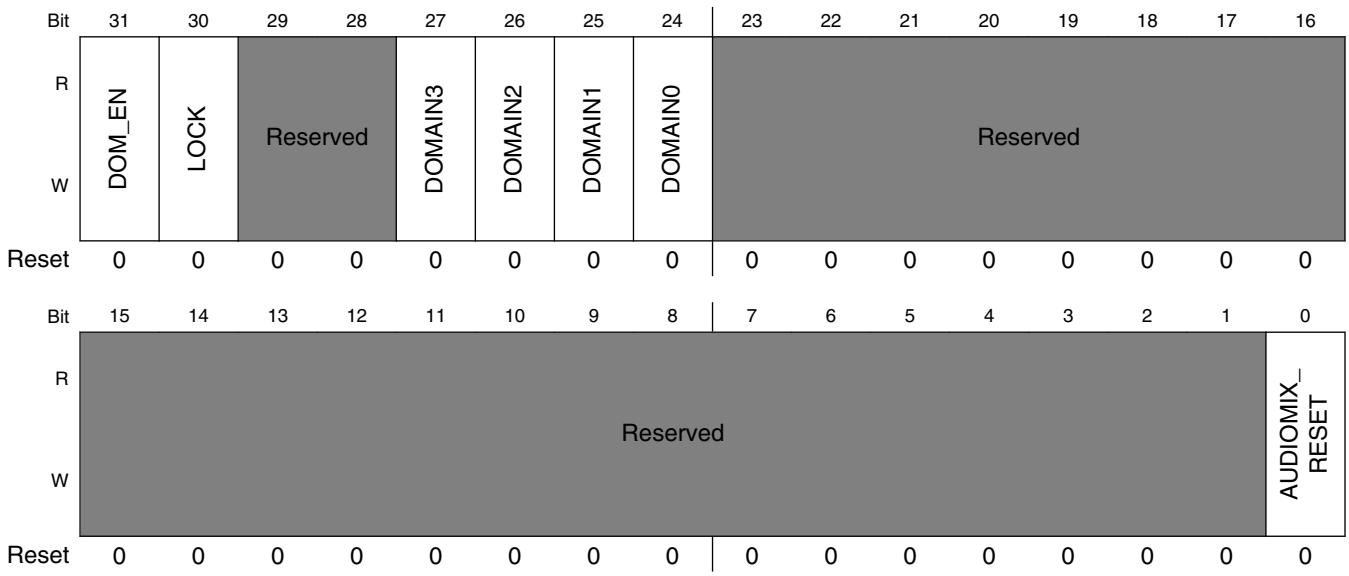
SRC_SUPERMIX_RCR field descriptions (continued)

Field	Description
	0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 SUPERMIX_ RESET	Self-clearing SW reset for SUPERMIX 0 Do not assert SUPERMIX reset 1 Assert SUPERMIX reset

6.5.5.6 AUDIOMIX Reset Control Register (SRC_AUDIOMIX_RCR)

The AUDIOMIX Reset Control Register contains bits that control the AUDIOMIX reset generation.

Address: 3039_0000h base + 1Ch offset = 3039_001Ch



SRC_AUDIOMIX_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register

Table continues on the next page...

SRC_AUDIOMIX_RCR field descriptions (continued)

Field	Description
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 AUDIOMIX_ RESET	Self-clearing SW reset for AUDIOMIX 0 Do not assert AUDIOMIX reset 1 Assert AUDIOMIX reset

6.5.5.7 USB PHY1 Reset Control Register (SRC_USBPHY1_RCR)

The USB PHY1 Reset Control Register (SRC_IP_RCR2), contains bits that control the USB PHY1 reset generation.

Address: 3039_0000h base + 20h offset = 3039_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DOM_EN	LOCK	Reserved		DOMAIN3	DOMAIN2	DOMAIN1	DOMAIN0	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															USB1_PHY_RESET
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

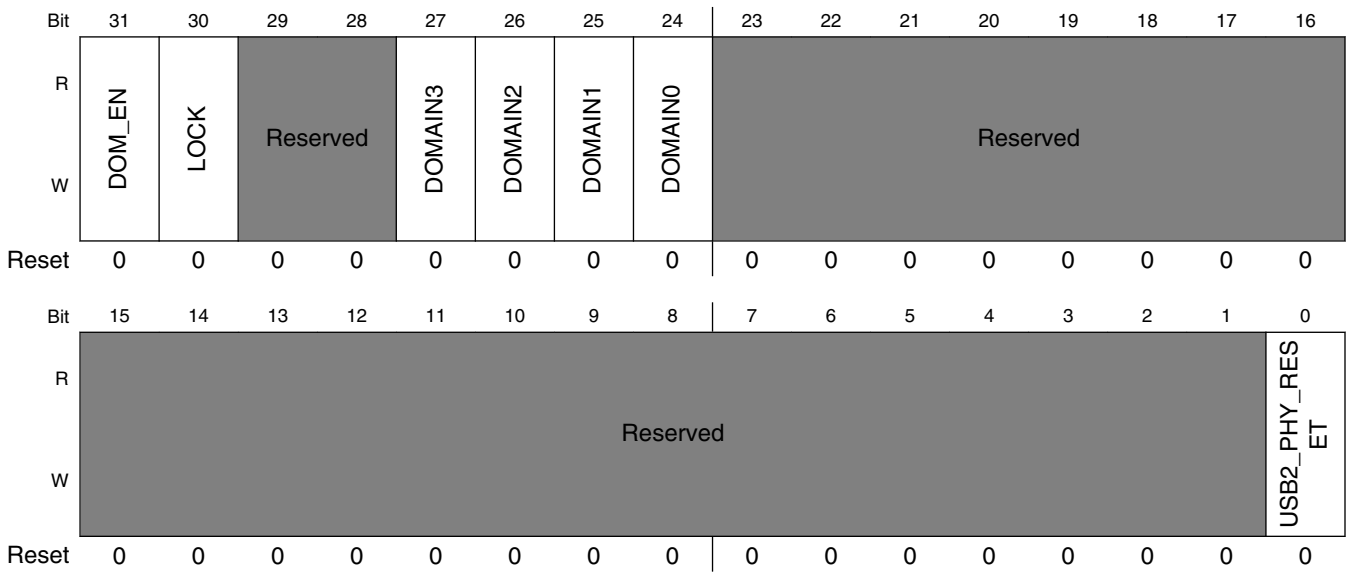
SRC_USBPHY1_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 USB1_PHY_RESET	Self-clearing SW reset for USB 1 PHY 0 Don't reset USB 1 PHY 1 Reset USB 1 PHY

6.5.5.8 USB PHY2 Reset Control Register (SRC_USBPHY2_RCR)

The USB PHY2 Reset Control Register (SRC_IP_RCR2), contains bits that control the USB PHY2 reset generation.

Address: 3039_0000h base + 24h offset = 3039_0024h



SRC_USBPHY2_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1.

Table continues on the next page...

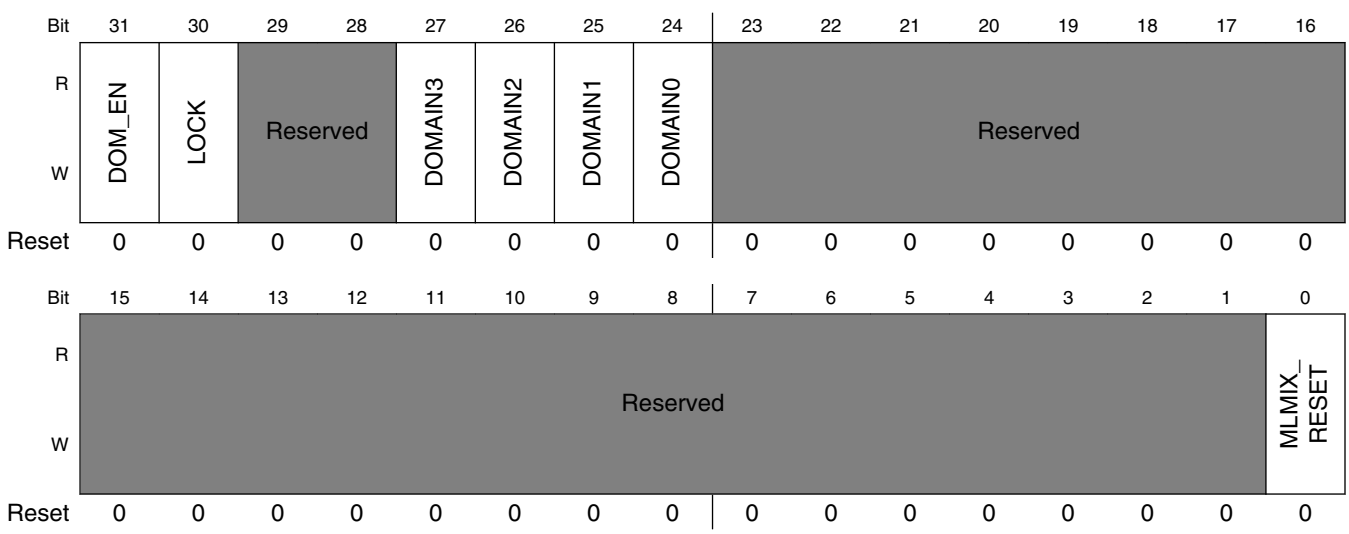
SRC_USBPHY2_RCR field descriptions (continued)

Field	Description
	0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 USB2_PHY_RESET	0 Don't reset USB 2 PHY 1 Reset USB 2 PHY

6.5.5.9 MLMIX Reset Control Register (SRC_MLMIX_RCR)

The MLMIX Reset Control Register contains bits that control the MLMIX reset generation.

Address: 3039_0000h base + 28h offset = 3039_0028h



SRC_MLMIX_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 MLMIX_RESET	Self-clearing SW reset for MLMIX 0 Do not assert MLMIX reset 1 Assert MLMIX reset

6.5.5.10 PCIE PHY Reset Control Register (SRC_PCIEPHY_RCR)

The PCIE PHY Control Register (SRC_PCIEPHY_RCR), contains bits that control the PCIE PHY reset generation.

Address: 3039_0000h base + 2Ch offset = 3039_002Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DOM_EN	LOCK	Reserved		DOMAIN3	DOMAIN2	DOMAIN1	DOMAIN0	Reserved							PCIE_CTRL_APP_XFER_PENDING
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCIE_CTRL_APP_UNLOCK_MSG	PCIE_CTRL_SYS_INT	Reserved	PCIE_CTRL_CFG_L1_AUX	PCIE_CTRL_APPS_TURNOFF	PCIE_CTRL_APPS_PME	PCIE_CTRL_APPS_EXIT	PCIE_CTRL_APPS_ENTER	PCIE_CTRL_APPS_READY	PCIE_CTRL_APPS_EN	PCIE_CTRL_APPS_RST	PCIE_CTRL_APPS_CLK_REQ	PCIEPHY_PERST	PCIEPHY_BTNRST	Reserved	PCIE_PHY_POWER_ON_RESET
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

SRC_PCIEPHY_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified

Table continues on the next page...

SRC_PCIEPHY_RCR field descriptions (continued)

Field	Description
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–17 -	This field is reserved. Reserved
16 PCIE_CTRL_ APP_XFER_ PENDING	PCIE_CTRL_APP_XFER_PENDING
15 PCIE_CTRL_ APP_UNLOCK_ MSG	PCIE_CTRL_APP_UNLOCK_MSG
14 PCIE_CTRL_ SYS_INT	PCIE_CTRL_SYS_INT
13 -	This field is reserved.
12 PCIE_CTRL_ CFG_L1_AUX	Pcie_ctrl_cfg_l1_aux_clk_switch_core_clk_gate_en
11 PCIE_CTRL_ APPS_ TURNOFF	Pcie_ctrl_apps_pm_xmt_turnoff
10 PCIE_CTRL_ APPS_PME	Pcie_ctrl_apps_pm_xmt_pme
9 PCIE_CTRL_ APPS_EXIT	Pcie_ctrl_app_req_exit_l1
8 PCIE_CTRL_ APPS_ENTER	Pcie_ctrl_app_req_entr_l1

Table continues on the next page...

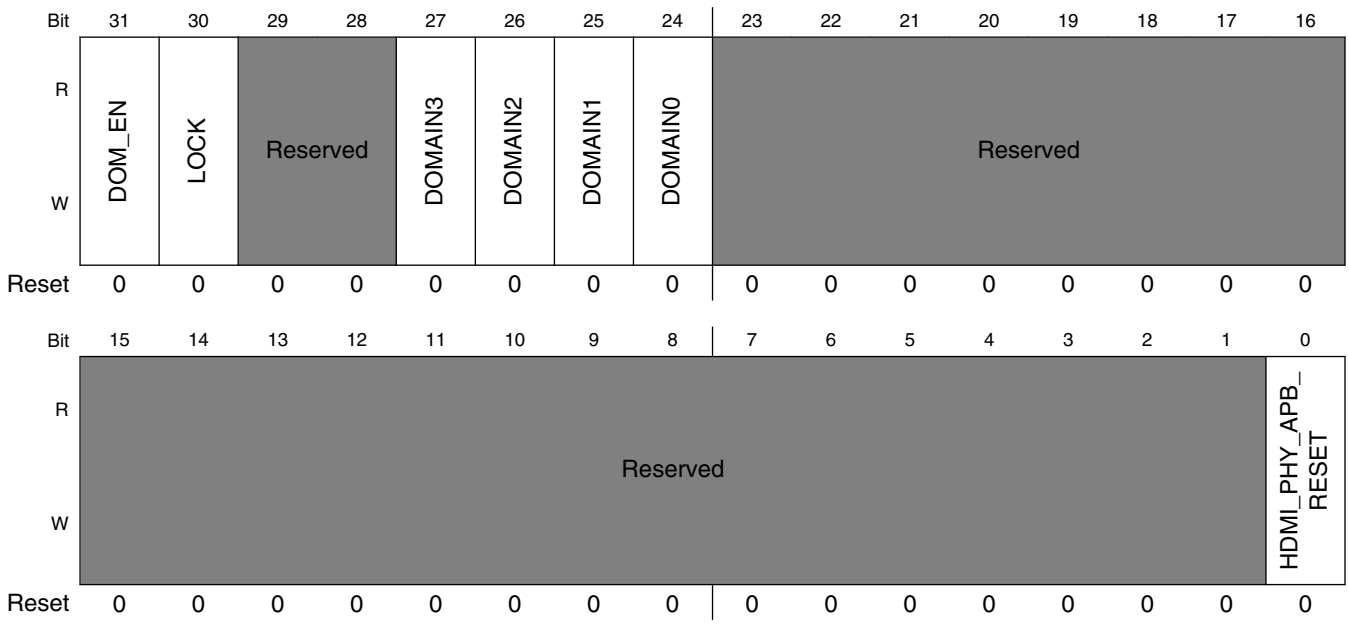
SRC_PCIEPHY_RCR field descriptions (continued)

Field	Description
7 PCIE_CTRL_ APPS_READY	Pcie_ctrl_app_ready_entr_l23
6 PCIE_CTRL_ APPS_EN	Pcie_ctrl_app_ltssm_enable
5 PCIE_CTRL_ APPS_RST	Pcie_ctrl_app_init_rst
4 PCIE_CTRL_ APPS_CLK_REQ	Pcie_ctrl_app_clk_req_n
3 PCIEPHY_ PERST	Pciephy_perst
2 PCIEPHY_BTNR ST	PCIE PHY button
1 -	This field is reserved. Reserved
0 PCIE_PHY_ POWER_ON_ RESET	PCIE_PHY_POWER_ON_RESET

6.5.5.11 HDMI Reset Control Register (SRC_HDMI_RCR)

The HDMI Control Register contains bits that control the HDMI reset generation.

Address: 3039_0000h base + 30h offset = 3039_0030h



SRC_HDMI_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register

Table continues on the next page...

SRC_HDMI_RCR field descriptions (continued)

Field	Description
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 HDMI_PHY_ APB_RESET	Active 1

6.5.5.12 MEDIAMIX Reset Control Register (SRC_MEDIA_RCR)

The MEDIAMIX Control Register contains bits that control the MEDIAMIX (expect ISP/Dewrap) reset generation.

Address: 3039_0000h base + 34h offset = 3039_0034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DOM_EN	LOCK	Reserved		DOMAIN3	DOMAIN2	DOMAIN1	DOMAIN0	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

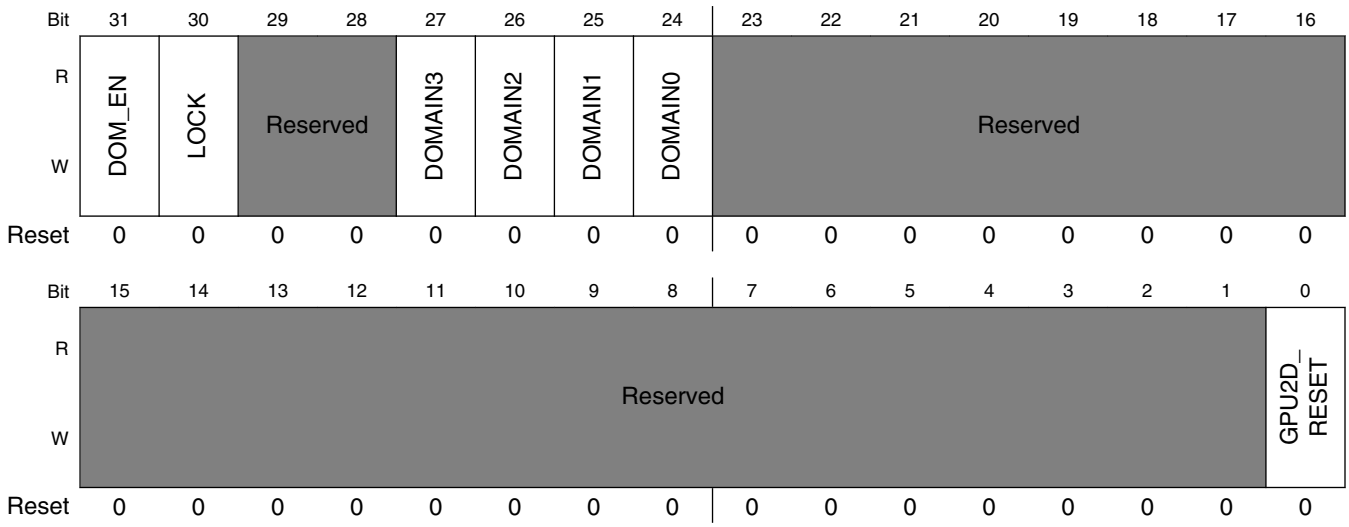
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															MEDIAMIX_ RESET
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_MEDIA_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 MEDIAMIX_ RESET	MEDIAMIX reset for MEDIAMIX, expect ISP/Dewrap 0 Don't reset MEDIAMIX 1 Reset MEDIAMIX

6.5.5.13 GPU2D Reset Control Register (SRC_GPU2D_RCR)

Address: 3039_0000h base + 38h offset = 3039_0038h



SRC_GPU2D_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register

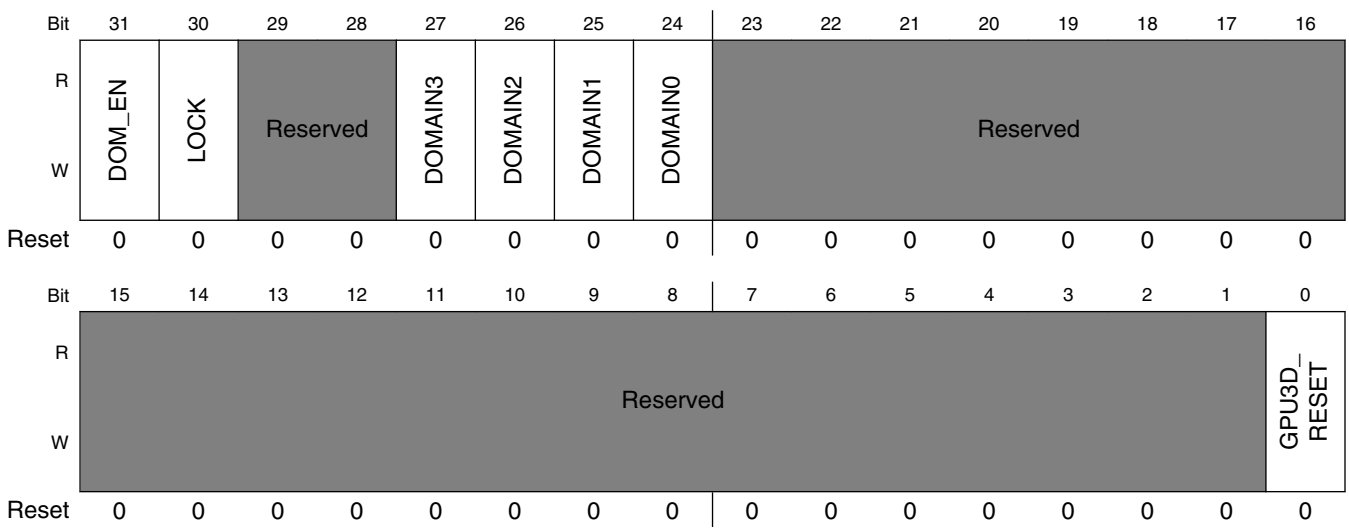
Table continues on the next page...

SRC_GPU2D_RCR field descriptions (continued)

Field	Description
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 GPU2D_RESET	GPU shared logic reset active 1

6.5.5.14 GPU3D Reset Control Register (SRC_GPU3D_RCR)

Address: 3039_0000h base + 3Ch offset = 3039_003Ch



SRC_GPU3D_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.

Table continues on the next page...

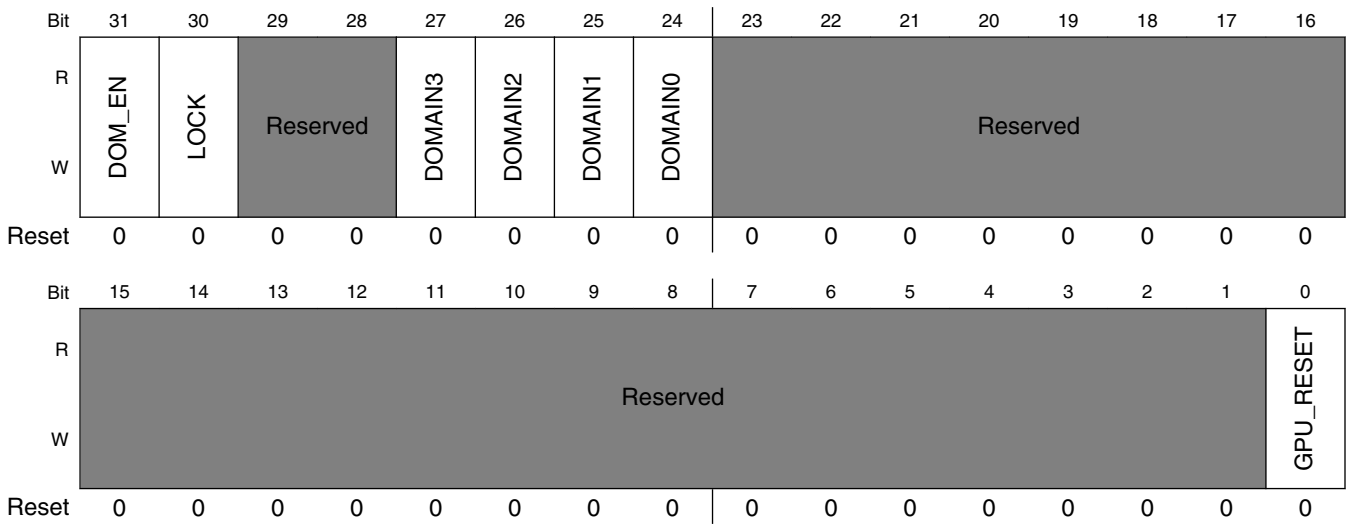
SRC_GPU3D_RCR field descriptions (continued)

Field	Description
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 GPU3D_RESET	GPU3D reset active 1

6.5.5.15 GPU Reset Control Register (SRC_GPU_RCR)

The GPU Control Register contains bits that control the GPU reset generation.

Address: 3039_0000h base + 40h offset = 3039_0040h



SRC_GPU_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1.

Table continues on the next page...

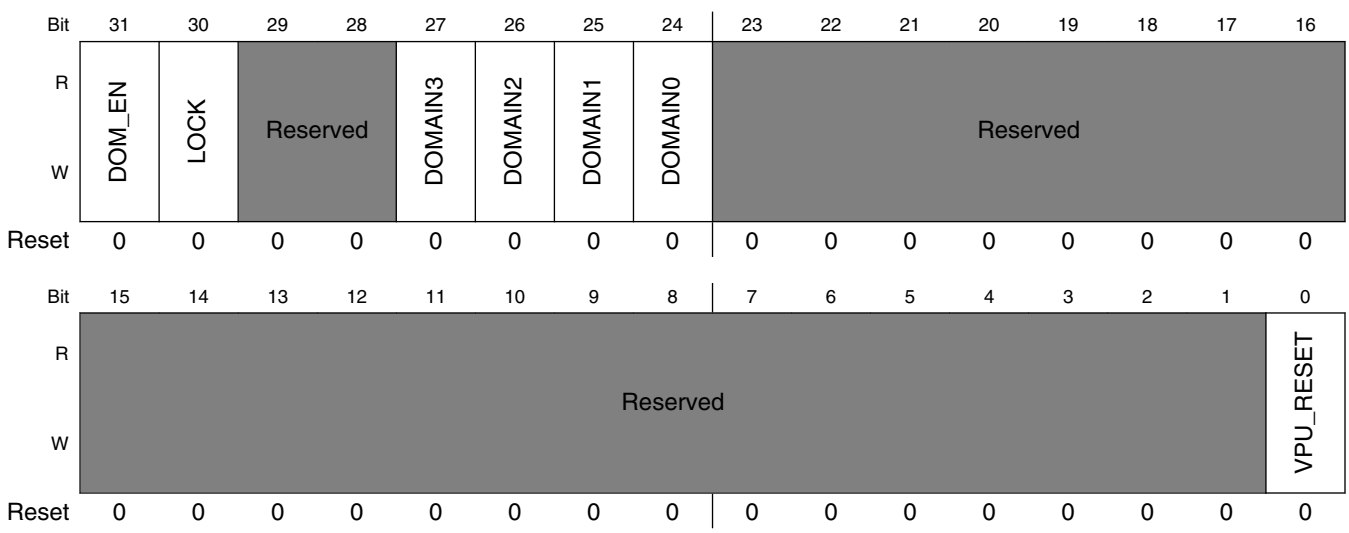
SRC_GPU_RCR field descriptions (continued)

Field	Description
	0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 GPU_RESET	GPU shared logic reset

6.5.5.16 VPU Reset Control Register (SRC_VPU_RCR)

The VPU Control Register contains bits that control the VPU reset generation.

Address: 3039_0000h base + 44h offset = 3039_0044h



SRC_VPU_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set.

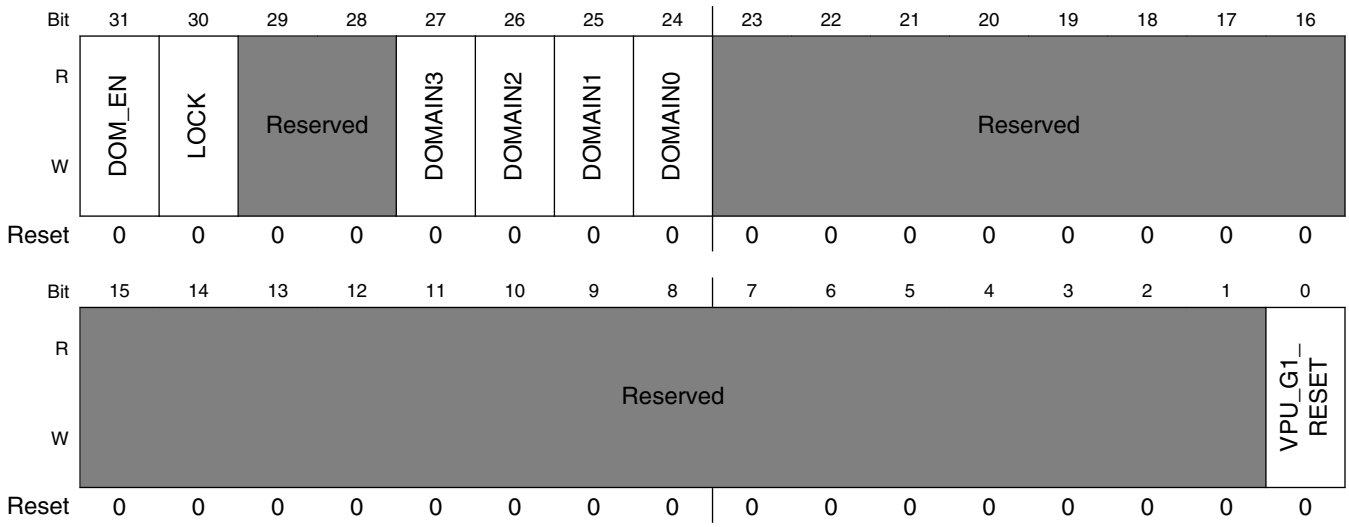
Table continues on the next page...

SRC_VPU_RCR field descriptions (continued)

Field	Description
	0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 VPU_RESET	VPU shared logic reset

6.5.5.17 VPU G1 Reset Control Register (SRC_VPU_G1_RCR)

Address: 3039_0000h base + 48h offset = 3039_0048h



SRC_VPU_G1_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register

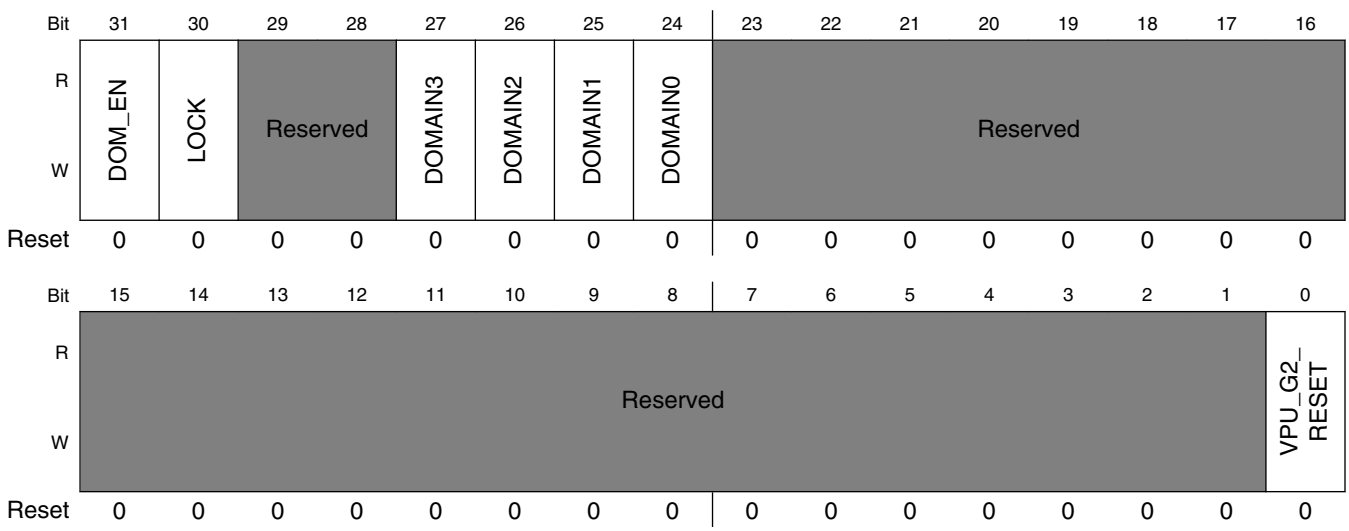
Table continues on the next page...

SRC_VPU_G1_RCR field descriptions (continued)

Field	Description
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 VPU_G1_RESET	VPU G1 reset active 1

6.5.5.18 VPU G2 Reset Control Register (SRC_VPU_G2_RCR)

Address: 3039_0000h base + 4Ch offset = 3039_004Ch



SRC_VPU_G2_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.

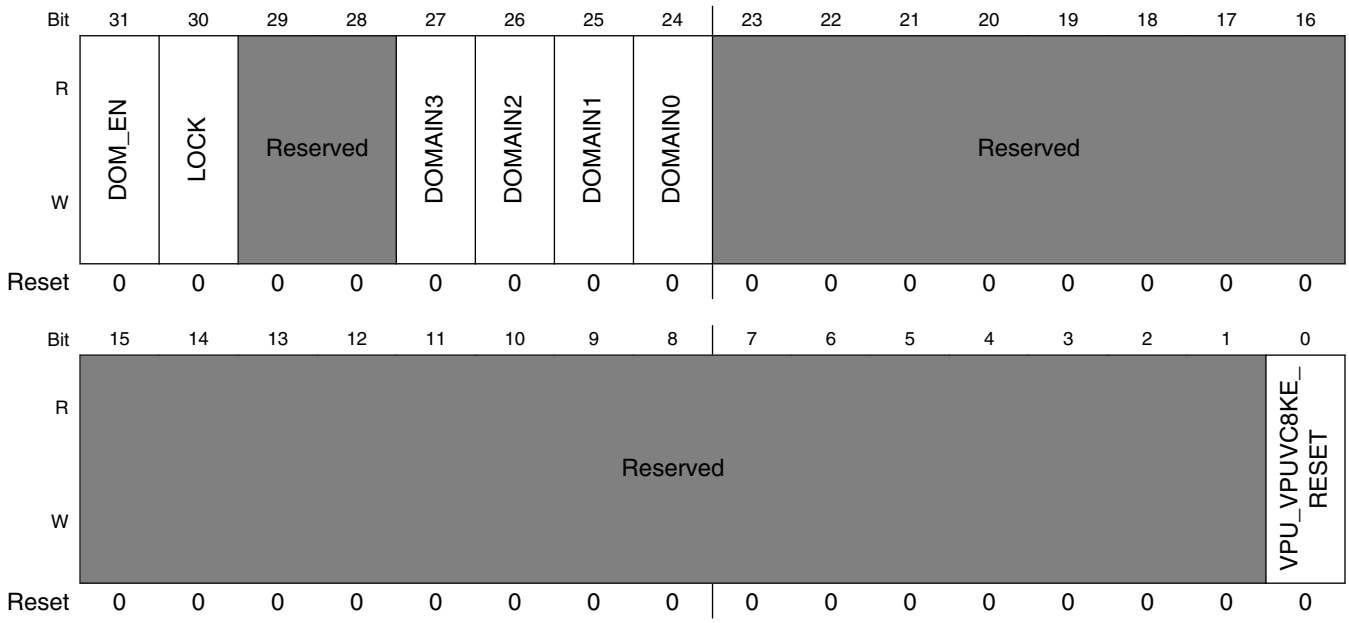
Table continues on the next page...

SRC_VPU_G2_RCR field descriptions (continued)

Field	Description
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 VPU_G2_RESET	VPU G2 reset active 1

6.5.5.19 VPU VC8000E Reset Control Register (SRC_VPUVC8KE_RCR)

Address: 3039_0000h base + 50h offset = 3039_0050h



SRC_VPUVC8KE_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register

Table continues on the next page...

SRC_VPUVC8KE_RCR field descriptions (continued)

Field	Description
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 VPU_ VPUVC8KE_ RESET	VPU VC8000E reset active 1

6.5.5.20 NOC Wrapper Reset Control Register (SRC_NOC_RCR)

Address: 3039_0000h base + 54h offset = 3039_0054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DOM_EN	LOCK	Reserved		DOMAIN3	DOMAIN2	DOMAIN1	DOMAIN0	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															NOC_RESET
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_NOC_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set.

Table continues on the next page...

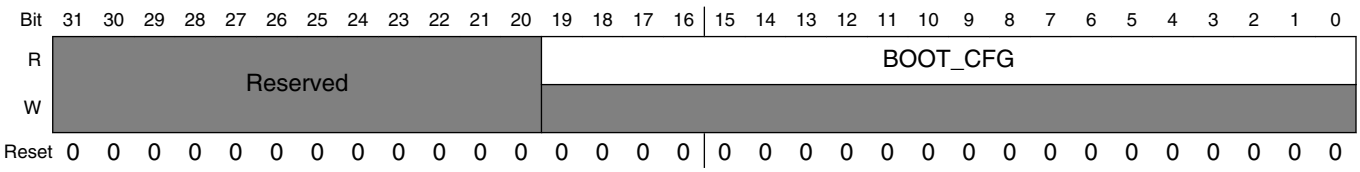
SRC_NOC_RCR field descriptions (continued)

Field	Description
	0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 NOC_RESET	NOC reset active 1

6.5.5.21 SRC Boot Mode Register 1 (SRC_SBMR1)

The Boot Mode register (SBMR) contains bits that reflect the status of Boot Mode Pins of the chip. The reset value is configuration dependent (depending on boot/fuses/IO pads).

Address: 3039_0000h base + 58h offset = 3039_0058h



SRC_SBMR1 field descriptions

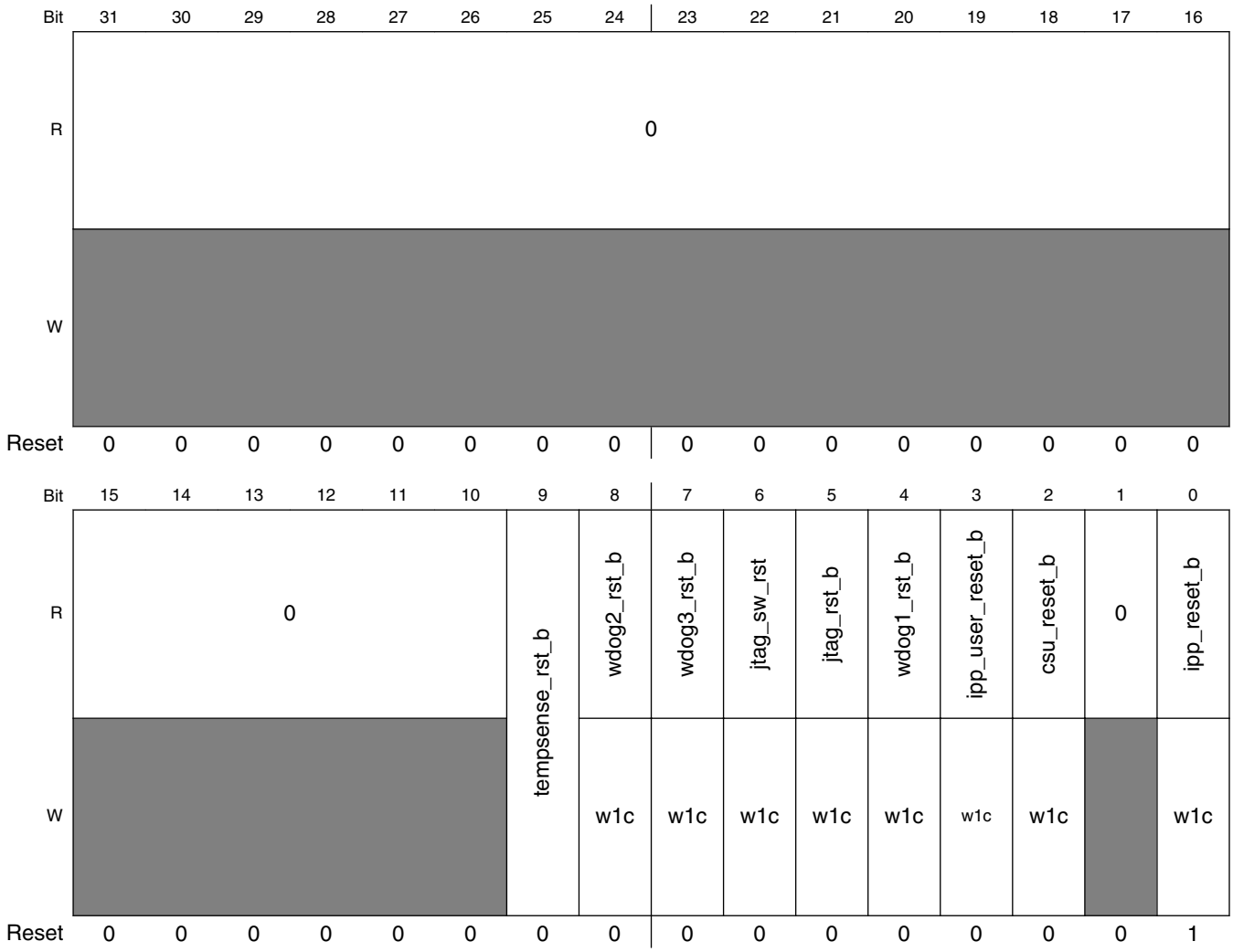
Field	Description
31–20 -	This field is reserved. Reserved
BOOT_CFG	Refer to fusemap.

6.5.5.22 SRC Reset Status Register (SRC_SRSR)

The SRSR is a write to one clear register which records the source of the reset events for the chip. The SRC reset status register will capture all the reset sources that have occurred. This register is reset on ipp_reset_b. This is a read-write register.

For bit[9-0] - writing zero does not have any effect. Writing one will clear the corresponding bit. The individual bits can be cleared by writing one to that bit. When the system comes out of reset, this register will have bits set corresponding to all the reset sources that occurred during system reset. Software has to take care to clear this register by writing one after every reset that occurs so that the register will contain the information of recently occurred reset.

Address: 3039_0000h base + 5Ch offset = 3039_005Ch



SRC_SRSR field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value 0.
9 tempsense_rst_b	Temper Sensor software reset. Indicates whether the reset was the result of software reset from on-chip Temperature Sensor. 0 Reset is not a result of software reset from Temperature Sensor. 1 Reset is a result of software reset from Temperature Sensor.
8 wdog2_rst_b	IC Watchdog2 Time-out reset. Indicates whether the reset was the result of the watchdog2 time-out event. 0 Reset is not a result of the watchdog4 time-out event. 1 Reset is a result of the watchdog4 time-out event.
7 wdog3_rst_b	IC Watchdog3 Time-out reset. Indicates whether the reset was the result of the watchdog3 time-out event. 0 Reset is not a result of the watchdog3 time-out event. 1 Reset is a result of the watchdog3 time-out event.

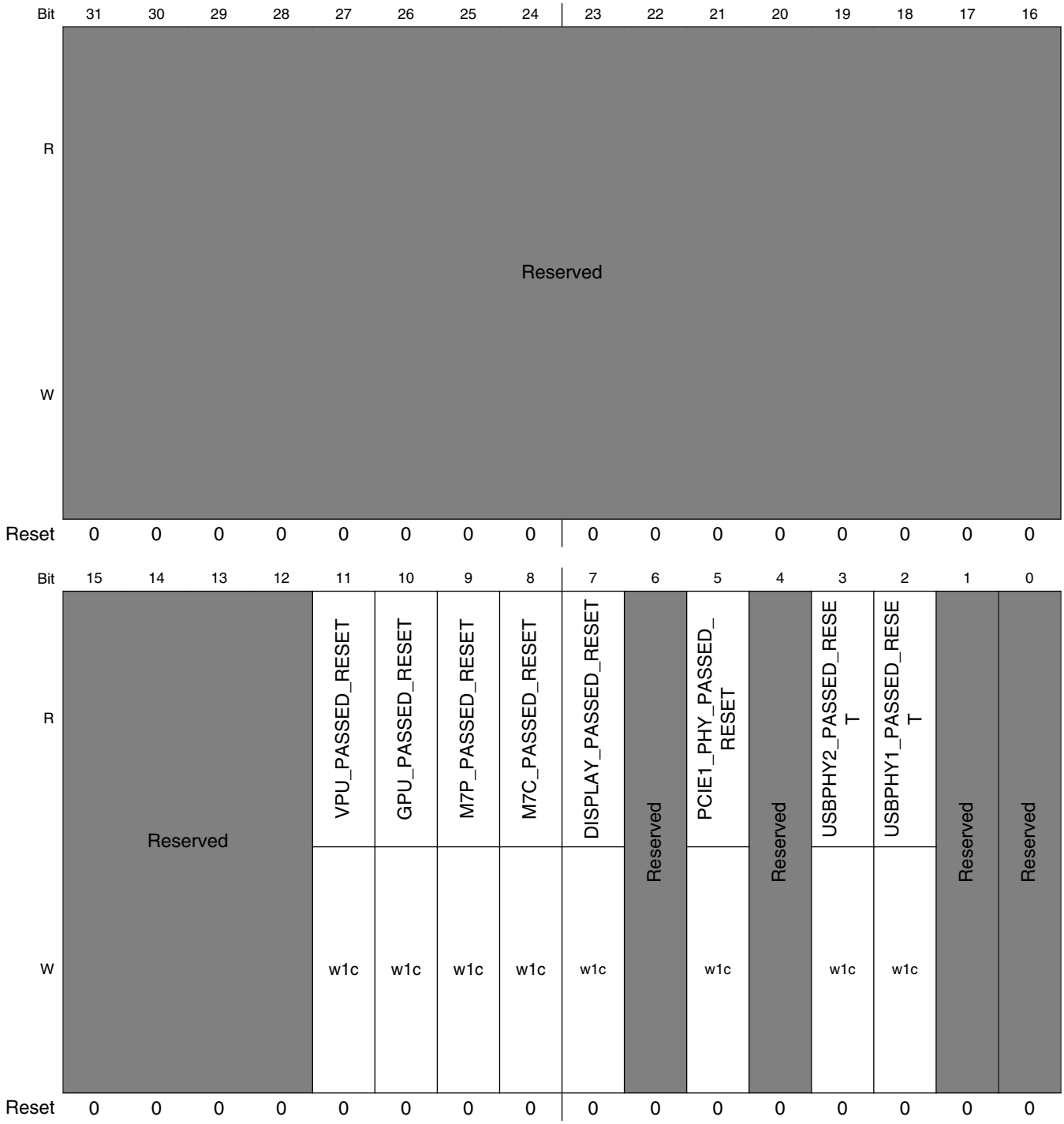
Table continues on the next page...

SRC_SRSR field descriptions (continued)

Field	Description
6 jtag_sw_rst	JTAG software reset. Indicates whether the reset was the result of software reset from JTAG. 0 Reset is not a result of software reset from JTAG. 1 Reset is a result of software reset from JTAG.
5 jtag_rst_b	HIGH - Z JTAG reset. Indicates whether the reset was the result of HIGH-Z reset from JTAG. 0 Reset is not a result of HIGH-Z reset from JTAG. 1 Reset is a result of HIGH-Z reset from JTAG.
4 wdog1_rst_b	IC Watchdog1 Time-out reset. Indicates whether the reset was the result of the watchdog time-out event. 0 Reset is not a result of the watchdog1 time-out event. 1 Reset is a result of the watchdog1 time-out event.
3 ipp_user_reset_b	Indicates whether the reset was the result of the ipp_user_reset_b qualified reset. 0 Reset is not a result of the ipp_user_reset_b qualified as COLD reset event. 1 Reset is a result of the ipp_user_reset_b qualified as COLD reset event.
2 csu_reset_b	Indicates whether the reset was the result of the csu_reset_b input. **NOTE:** If case the csu_reset_b occurred during a WARM reset process, during the phase that ipg_clk is not available yet, then the occurrence of CSU reset will not be reflected in this bit. 0 Reset is not a result of the csu_reset_b event. 1 Reset is a result of the csu_reset_b event.
1 Reserved	This read-only field is reserved and always has the value 0.
0 ipp_reset_b	Indicates whether reset was the result of ipp_reset_b pin (Power-up sequence) 0 Reset is not a result of ipp_reset_b pin. 1 Reset is a result of ipp_reset_b pin.

6.5.5.23 SRC Interrupt Status Register (SRC_SISR)

Address: 3039_0000h base + 68h offset = 3039_0068h

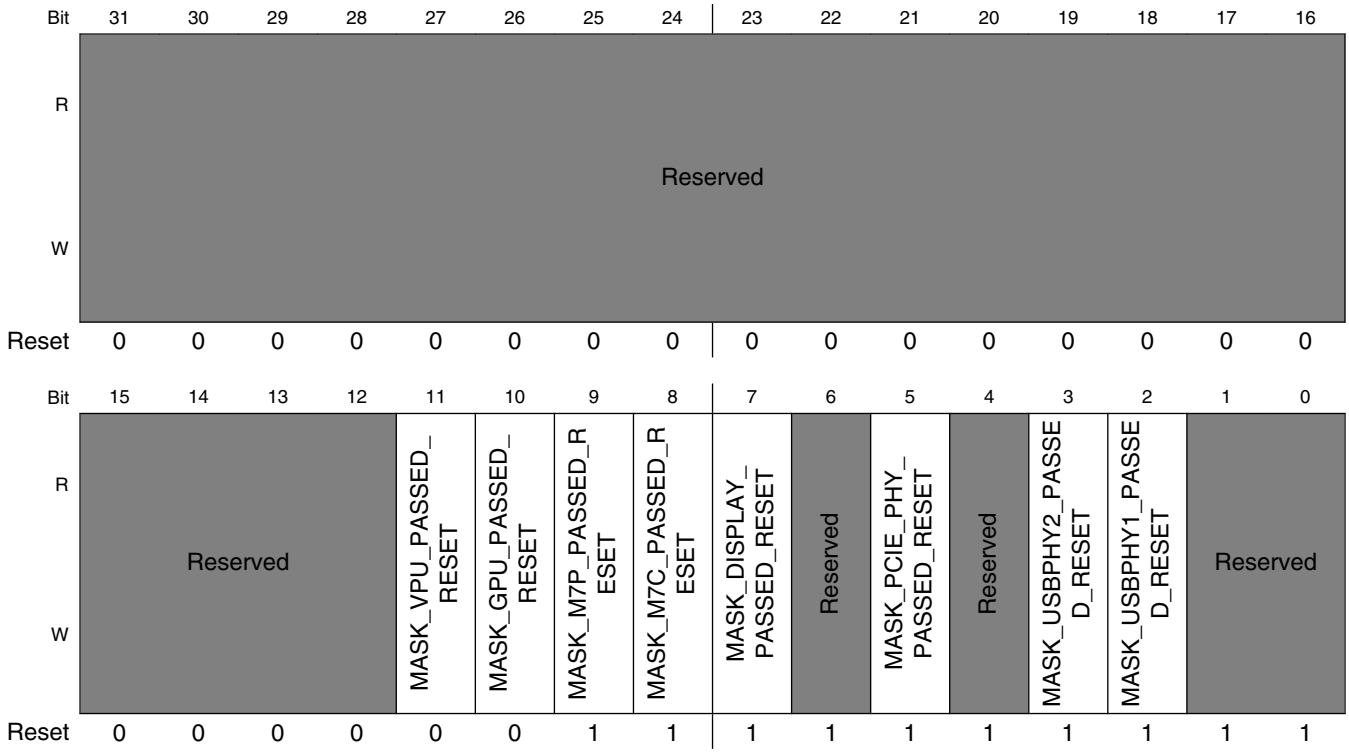


SRC_SISR field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11 VPU_PASSED_RESET	Interrupt generated to indicate that VPU passed software reset and is ready to be used 0 interrupt generated not due to VPU reset 1 interrupt generated due to VPU reset
10 GPU_PASSED_RESET	Interrupt generated to indicate that GPU passed software reset and is ready to be used 0 interrupt generated not due to GPU reset 1 interrupt generated due to GPU reset
9 M7P_PASSED_RESET	Interrupt generated to indicate that m7 platform passed software reset and is ready to be used 0 interrupt generated not due to m7 platform reset 1 interrupt generated due to m7 platform reset
8 M7C_PASSED_RESET	Interrupt generated to indicate that m7 core passed software reset and is ready to be used 0 interrupt generated not due to m7core reset 1 interrupt generated due to m7core reset
7 DISPLAY_PASSED_RESET	Interrupt generated to indicate that DISPLAY passed software reset and is ready to be used 0 Interrupt generated not due to DISPLAY passed reset 1 Interrupt generated due to DISPLAY passed reset
6 -	This field is reserved. Reserved
5 PCIE1_PHY_PASSED_RESET	Interrupt generated to indicate that PCIE1 PHY passed software reset and is ready to be used 0 Interrupt generated not due to PCIE1 PHY passed reset 1 Interrupt generated due to PCIE1 PHY passed reset
4 -	This field is reserved. Reserved
3 USBPHY2_PASSED_RESET	Interrupt generated to indicate that USB PHY2 passed software reset and is ready to be used 0 Interrupt generated not due to USB PHY2 passed reset 1 Interrupt generated due to USB PHY2 passed reset
2 USBPHY1_PASSED_RESET	Interrupt generated to indicate that USB PHY1 passed software reset and is ready to be used 0 Interrupt generated not due to USB PHY1 passed reset 1 Interrupt generated due to USB PHY1 passed reset
1 -	This field is reserved. Reserved
0 -	This field is reserved. Reserved

6.5.5.24 SRC Interrupt Mask Register (SRC_SIMR)

Address: 3039_0000h base + 6Ch offset = 3039_006Ch



SRC_SIMR field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11 MASK_VPU_PASSED_RESET	Mask interrupt generation due to VPU passed reset 0 do not mask interrupt due to VPU passed reset - interrupt will be created 1 mask interrupt due to VPU passed reset
10 MASK_GPU_PASSED_RESET	Mask interrupt generation due to GPU passed reset 0 do not mask interrupt due to GPU passed reset - interrupt will be created 1 mask interrupt due to GPU passed reset
9 MASK_M7P_PASSED_RESET	mask interrupt generation due to m7 platform passed reset 0 do not mask interrupt due to m7 platform passed reset - interrupt will be created 1 mask interrupt due to m7 platform passed reset
8 MASK_M7C_PASSED_RESET	mask interrupt generation due to m7 core passed reset 0 do not mask interrupt due to m7 core passed reset - interrupt will be created 1 mask interrupt due to m7 core passed reset

Table continues on the next page...

SRC_SIMR field descriptions (continued)

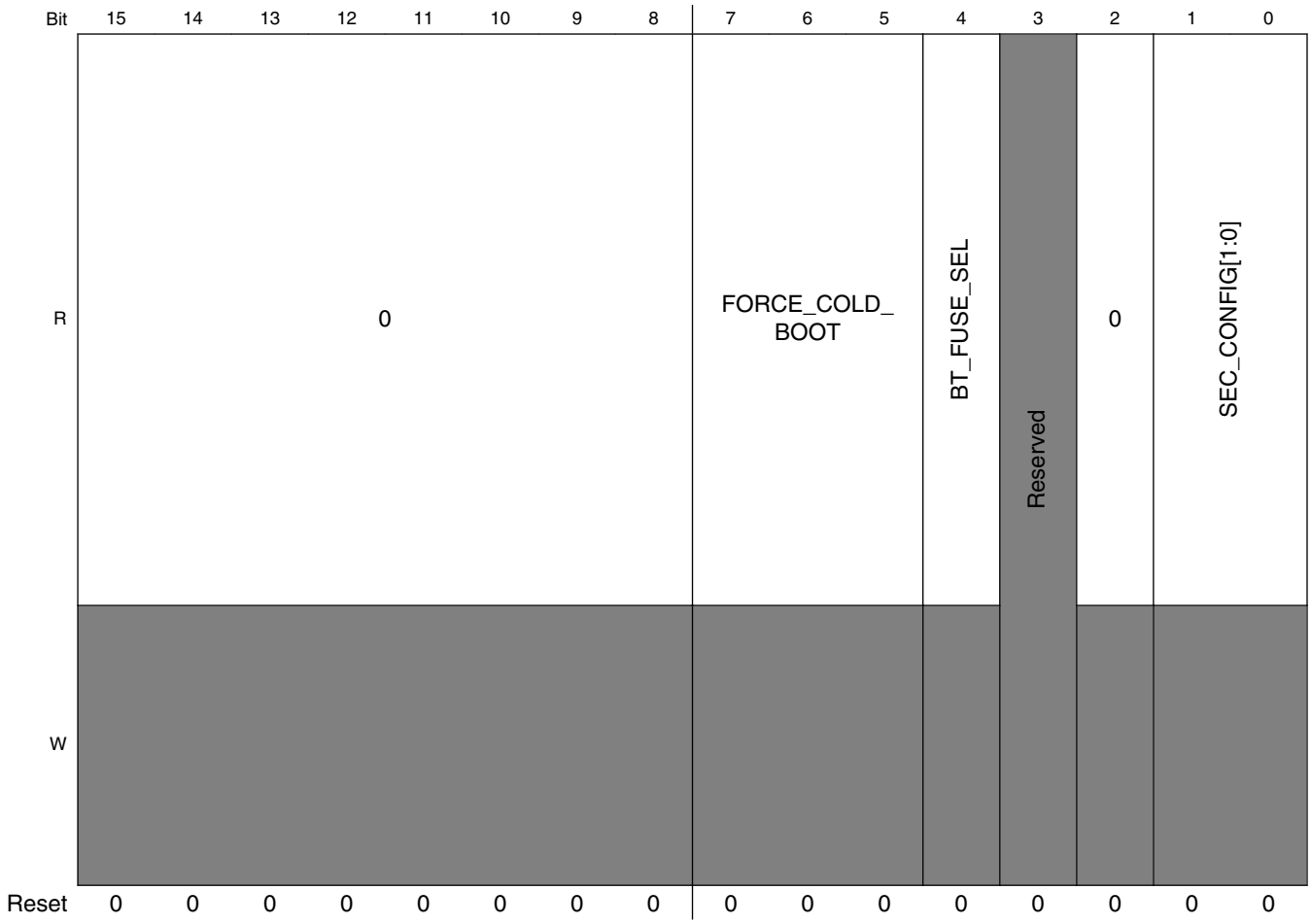
Field	Description
7 MASK_ DISPLAY_ PASSED_ RESET	Mask interrupt generation due to display passed reset 0 do not mask interrupt due to display passed reset - interrupt will be created 1 mask interrupt due to display passed reset
6 -	This field is reserved. Reserved
5 MASK_PCIE_ PHY_PASSED_ RESET	Mask interrupt generation due to PCIE PHY passed reset 0 do not mask interrupt due to PCIE PHY passed reset - interrupt will be created 1 mask interrupt due to PCIE PHY passed reset
4 -	This field is reserved. Reserved
3 MASK_USBPHY 2_PASSED_RES ET	mask interrupt generation due to USB PHY2 passed reset 0 do not mask interrupt due to USB PHY2 passed reset - interrupt will be created 1 mask interrupt due to USB PHY2 passed reset
2 MASK_USBPHY 1_PASSED_RES ET	mask interrupt generation due to USB PHY1 passed reset 0 do not mask interrupt due to USB PHY1 passed reset - interrupt will be created 1 mask interrupt due to USB PHY1 passed reset
-	This field is reserved. Reserved

6.5.5.25 SRC Boot Mode Register 2 (SRC_SBMR2)

The Boot Mode register (SBMR), contains bits that reflect the status of Boot Mode Pins of the chip. The default values for those bits depends on the values of pins/fuses during reset sequence.

Address: 3039_0000h base + 70h offset = 3039_0070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				IPP_BOOT_MODE				0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



SRC_SBMR2 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–24 IPP_BOOT_ MODE	IPP_BOOT_MODE shows the latched state of the BOOT_MODE3, BOOT_MODE2, BOOT_MODE1 and BOOT_MODE0 signals on the rising edge of POR_B. See the Boot mode pin settings section of System Boot.
23–8 Reserved	This read-only field is reserved and always has the value 0.
7–5 FORCE_COLD_ BOOT	See Fusemap for additional information.
4 BT_FUSE_SEL	BT_FUSE_SEL (connected to gpio bt_fuse_sel) shows the state of the BT_FUSE_SEL fuse. See Fusemap for additional information on this fuse.
3 -	This field is reserved. Reserved
2 Reserved	This read-only field is reserved and always has the value 0.
SEC_ CONFIG[1:0]	SEC_CONFIG[1] shows the state of the SEC_CONFIG[1] fuse and SEC_CONFIG[0] shows the state of the SEC_CONFIG[0] fuse. See Fusemap for additional information on this fuse.

6.5.5.26 SRC General Purpose Register 1 (SRC_GPR1)

Address: 3039_0000h base + 74h offset = 3039_0074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																C0_START_ADDRH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR1 field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
C0_START_ADDRH	Core0 start reset address: RVBARADDR0 = {SRC_GPR1[15:0], SRC_GPR2[21:2]}

6.5.5.27 SRC General Purpose Register 2 (SRC_GPR2)

Address: 3039_0000h base + 78h offset = 3039_0078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										C0_START_ADDRL																					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR2 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
C0_START_ADDRL	Core0 start reset address: RVBARADDR0 = {SRC_GPR1[15:0], SRC_GPR2[21:2]}

6.5.5.28 SRC General Purpose Register 3 (SRC_GPR3)

Address: 3039_0000h base + 7Ch offset = 3039_007Ch

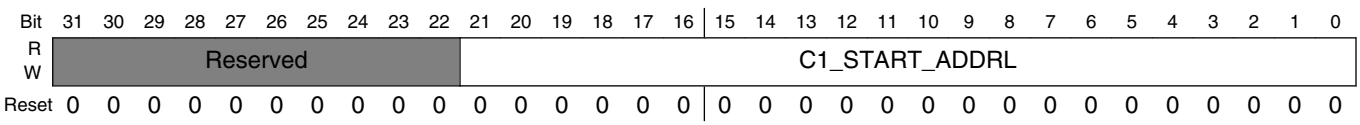
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																C1_START_ADDRH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR3 field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
C1_START_ADDRH	Core1 start reset address: RVBARADDR1 = {SRC_GPR3[15:0], SRC_GPR4[21:2]}

6.5.5.29 SRC General Purpose Register 4 (SRC_GPR4)

Address: 3039_0000h base + 80h offset = 3039_0080h

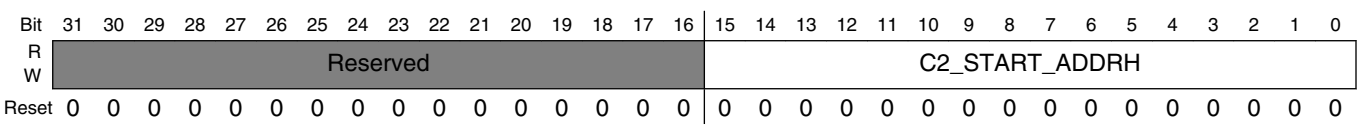


SRC_GPR4 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
C1_START_ADDRL	Core1 start reset address: RVBARADDR1 = {SRC_GPR3[15:0], SRC_GPR4[21:2]}

6.5.5.30 SRC General Purpose Register 5 (SRC_GPR5)

Address: 3039_0000h base + 84h offset = 3039_0084h



SRC_GPR5 field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
C2_START_ADDRH	Core2 start reset address: RVBARADDR2 = {SRC_GPR5[15:0], SRC_GPR6[21:2]}

6.5.5.31 SRC General Purpose Register 6 (SRC_GPR6)

Address: 3039_0000h base + 88h offset = 3039_0088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										C2_START_ADDRL																					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR6 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
C2_START_ADDRL	Core2 start reset address: RVBARADDR2 = {SRC_GPR5[15:0], SRC_GPR6[21:2]}

6.5.5.32 SRC General Purpose Register 7 (SRC_GPR7)

Address: 3039_0000h base + 8Ch offset = 3039_008Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																C3_START_ADDRH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR7 field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
C3_START_ADDRH	Core3 start reset address: RVBARADDR3 = {SRC_GPR7[15:0], SRC_GPR8[21:2]}

6.5.5.33 SRC General Purpose Register 8 (SRC_GPR8)

Address: 3039_0000h base + 90h offset = 3039_0090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										C3_START_ADDRL																					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR8 field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
C3_START_ADDRL	Core3 start reset address: RVBARADDR3 = {SRC_GPR7[15:0], SRC_GPR8[21:2]}

6.5.5.34 SRC General Purpose Register 9 (SRC_GPR9)

Reserved for Internal Use.

Address: 3039_0000h base + 94h offset = 3039_0094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR9 field descriptions

Field	Description
-	This field is reserved. Reserved.

6.5.5.35 SRC General Purpose Register 10 (SRC_GPR10)

Address: 3039_0000h base + 98h offset = 3039_0098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

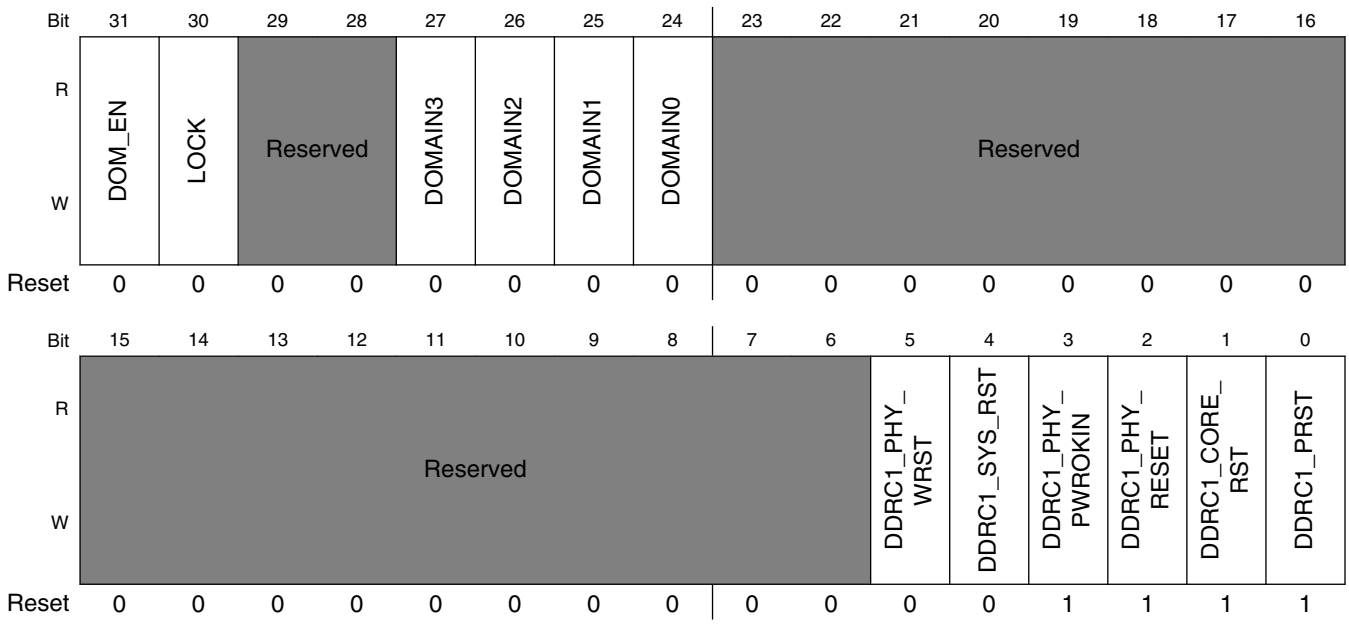
SRC_GPR10 field descriptions

Field	Description
-	This field is reserved. Reserved.

6.5.5.36 SRC DDR Controller Reset Control Register (SRC_DDRC_RCR)

DDR Controller Reset Control Register

Address: 3039_0000h base + 1000h offset = 3039_1000h



SRC_DDRC_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1.

Table continues on the next page...

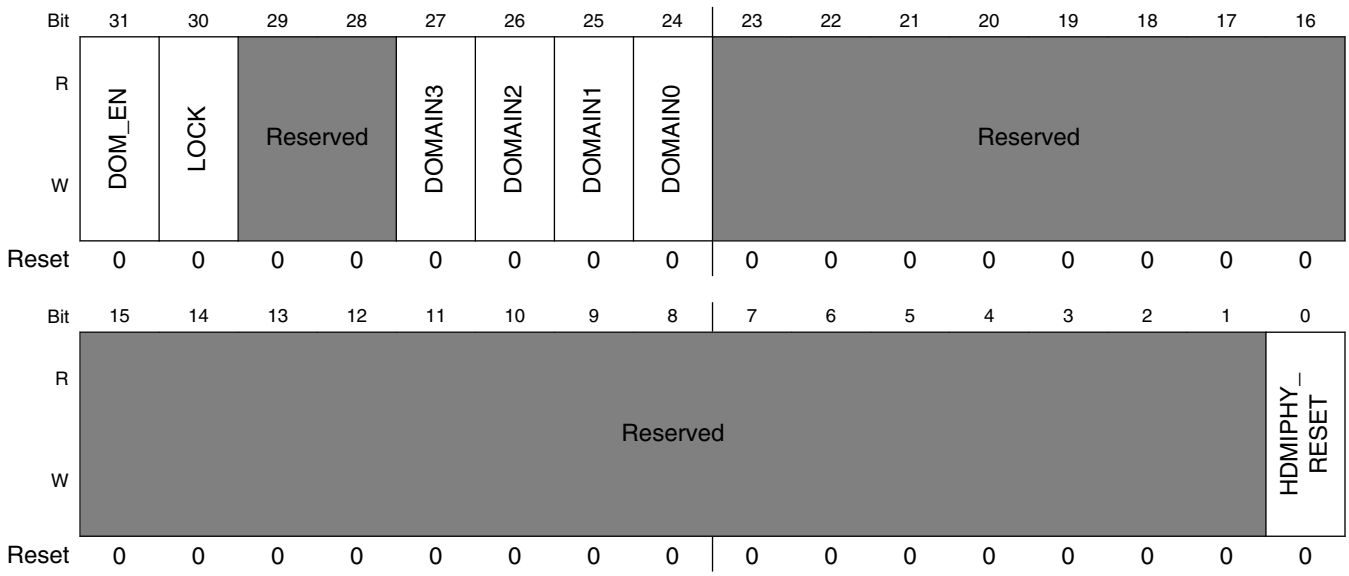
SRC_DDRC_RCR field descriptions (continued)

Field	Description
	0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain0 cannot write to this register. 1 This register is assigned to domain0. The master from domain0 can write to this register
23–6 -	This field is reserved. Reserved
5 DDRC1_PHY_WRST	Active 1
4 DDRC1_SYS_RST	Active 1
3 DDRC1_PHY_PWROKIN	0 De-assert DDR controller 1 Assert DDR Controller
2 DDRC1_PHY_RESET	0 De-assert DDR controller 1 Assert DDR Controller
1 DDRC1_CORE_RST	DDR Controller core_ddrc_rstn and aresetn. 0 De-assert DDR controller aresetn and core_ddrc_rstn 1 Assert DDR Controller preset and DDR PHY reset
0 DDRC1_PRST	DDR Controller preset and DDR PHY reset. **NOTE:** This reset can only be released when DDR Controller clock inputs are active and stable for 30 cycles 0 De-assert DDR Controller preset and DDR PHY reset reset 1 Assert DDR Controller preset and DDR PHY reset

6.5.5.37 HDMIPHY Reset Control Register (SRC_HDMIPHY_RCR)

The HDMI PHY Reset Control Register contains bits that control the HDMI PHY reset generation.

Address: 3039_0000h base + 1008h offset = 3039_1008h



SRC_HDMIPHY_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register

Table continues on the next page...

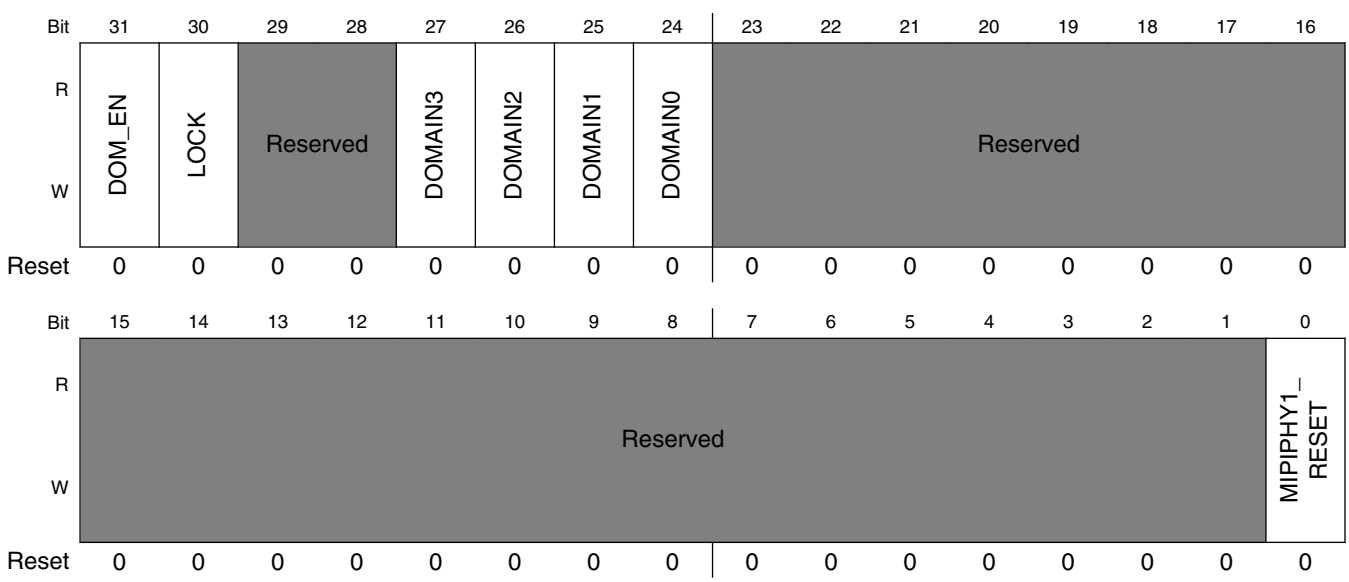
SRC_HDMIPHY_RCR field descriptions (continued)

Field	Description
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 HDMIPHY_ RESET	Self-clearing SW reset for HDMI PHY 0 Do not assert HDMI PHY reset 1 Assert HDMI PHY reset

6.5.5.38 MIPI PHY1 Reset Control Register (SRC_MIPIPHY1_RCR)

The MIPI PHY1 Reset Control Register contains bits that control the MIPI PHY1 reset generation.

Address: 3039_0000h base + 100Ch offset = 3039_100Ch



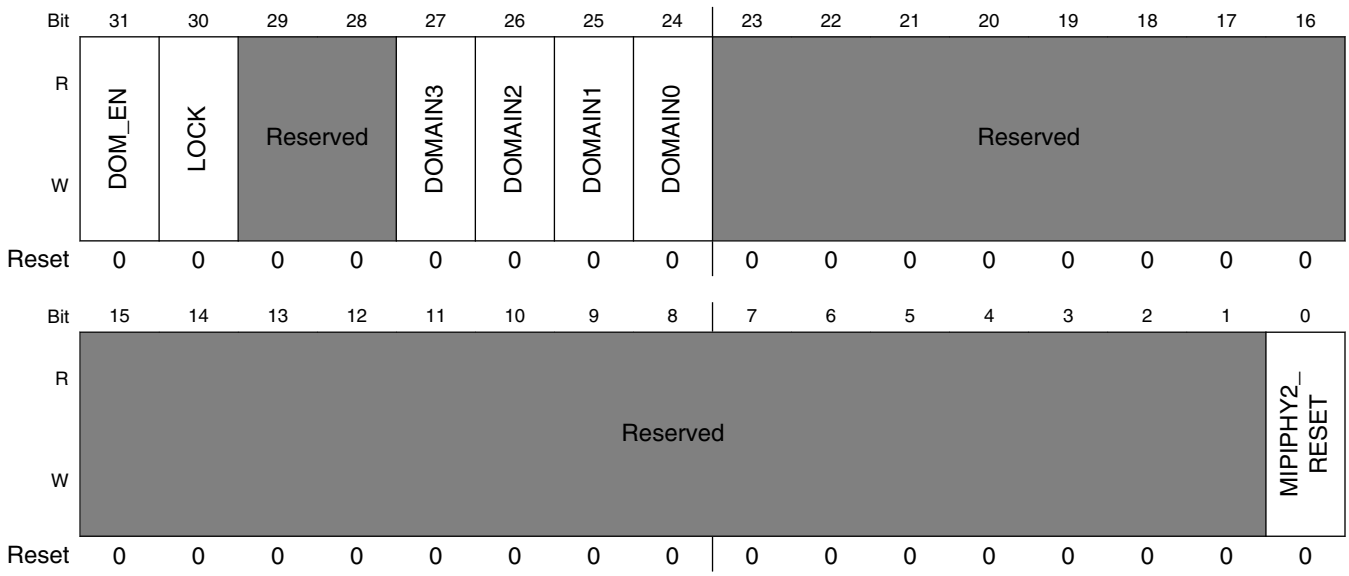
SRC_MIPIPHY1_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 MIPIPHY1_ RESET	Self-clearing SW reset for MIPI PHY1 0 Do not assert MIPI PHY1 reset 1 Assert MIPI PHY1 reset

6.5.5.39 MIPI PHY2 Reset Control Register (SRC_MIPIPHY2_RCR)

The MIPI PHY2 Reset Control Register contains bits that control the MIPI PHY2 reset generation.

Address: 3039_0000h base + 1010h offset = 3039_1010h



SRC_MIPIPHY2_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register

Table continues on the next page...

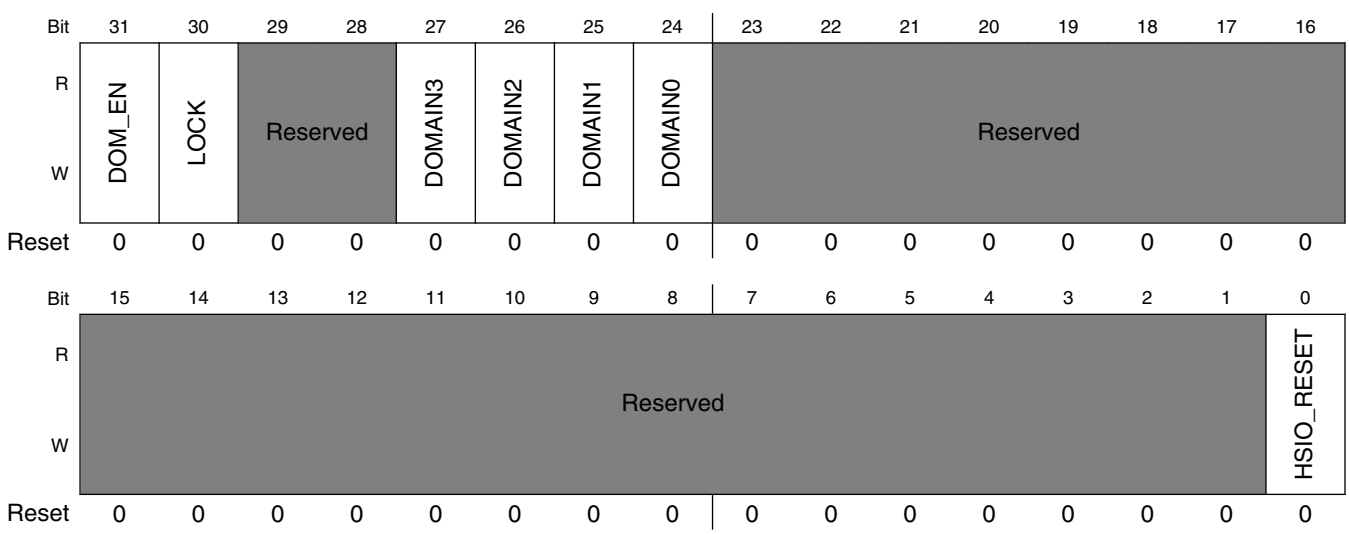
SRC_MIPIPHY2_RCR field descriptions (continued)

Field	Description
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 MIPIPHY2_ RESET	Self-clearing SW reset for MIPI PHY2 0 Do not assert MIPI PHY2 reset 1 Assert MIPI PHY2 reset

6.5.5.40 HSIO Reset Control Register (SRC_HSIO_RCR)

The HSIOMIX Reset Control Register contains bits that control the HSIOMIX reset generation.

Address: 3039_0000h base + 1014h offset = 3039_1014h



SRC_HSIO_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register.

Table continues on the next page...

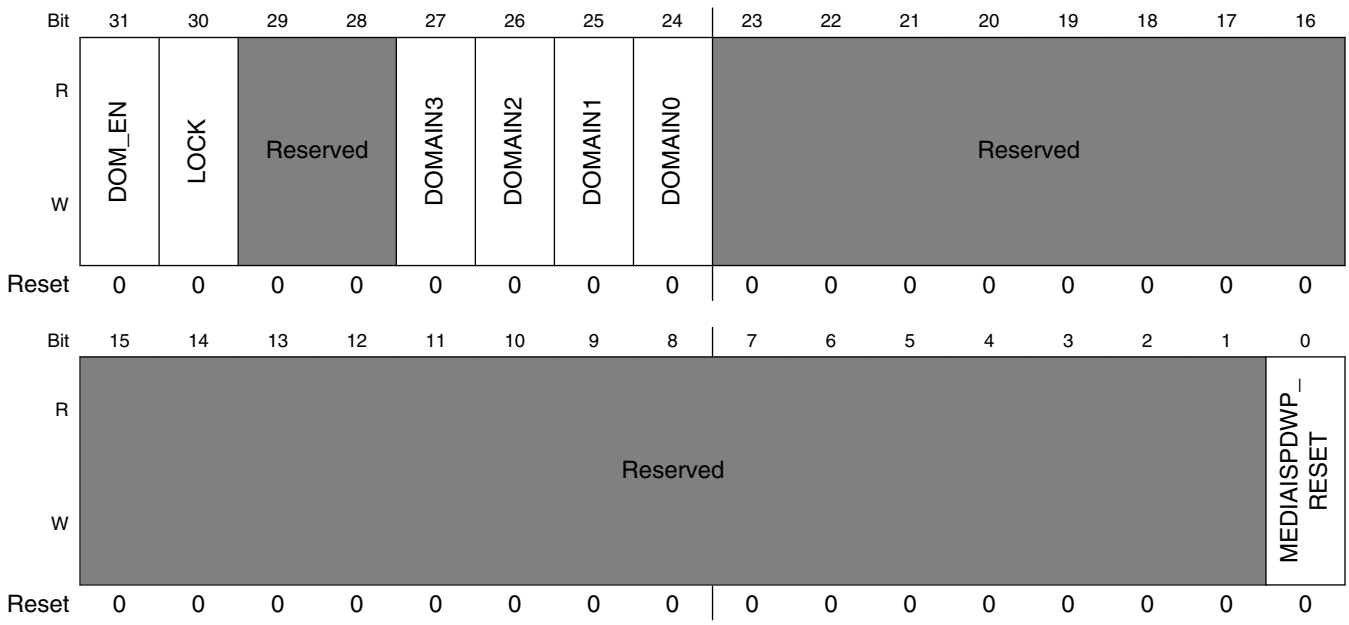
SRC_HSIO_RCR field descriptions (continued)

Field	Description
	NOTE: [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 HSIO_RESET	Self-clearing SW reset for HSIOMIX 0 Do not assert HSIOMIX reset 1 Assert HSIOMIX reset

6.5.5.41 MEDIAMIX ISP and Dewarp Reset Control Register (SRC_MEDIAISPDWP_RCR)

The MEDIAMIX ISP and Dewarp Reset Control Register contains bits that control the MEDIAMIX ISP and Dewarp reset generation.

Address: 3039_0000h base + 1018h offset = 3039_1018h



SRC_MEDIAISPDWP_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1.

Table continues on the next page...

SRC_MEDIAISPDWP_RCR field descriptions (continued)

Field	Description
	0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 MEDIAISPDWP_ RESET	Self-clearing SW reset for MEDIAMIX ISP and Dewarp 0 Do not assert MEDIAMIX ISP and Dewarp reset 1 Assert MEDIAMIX ISP and Dewarp reset

6.6 Watchdog Timer (WDOG)

6.6.1 Overview

The Watchdog Timer (WDOG) protects against system failures by providing a method by which to escape from unexpected events or programming errors.

Once the WDOG is activated, it must be serviced by the software on a periodic basis. If servicing does not take place, the timer times out. Upon timeout, the WDOG asserts the internal system reset signal, WDOG_RESET_B_DEB to the System Reset Controller (SRC).

There is also a provision for WDOG(ipp_wdog) signal assertion by timeout counter expiration. There is an option of programmable interrupt generation before the counter actually times out. The time at which the interrupt needs to be generated prior to counter timeout is programmable. There is a power down counter which is enabled out of any reset (POR, Warm/Cold). This counter has a fixed timeout period of 16 seconds, upon which it asserts the WDOG(ipp_wdog) signal.

Flow diagrams for the timeout counter, power down counter and interrupt operations are are shown in Flow Diagrams.

6.6.1.1 Block Diagram

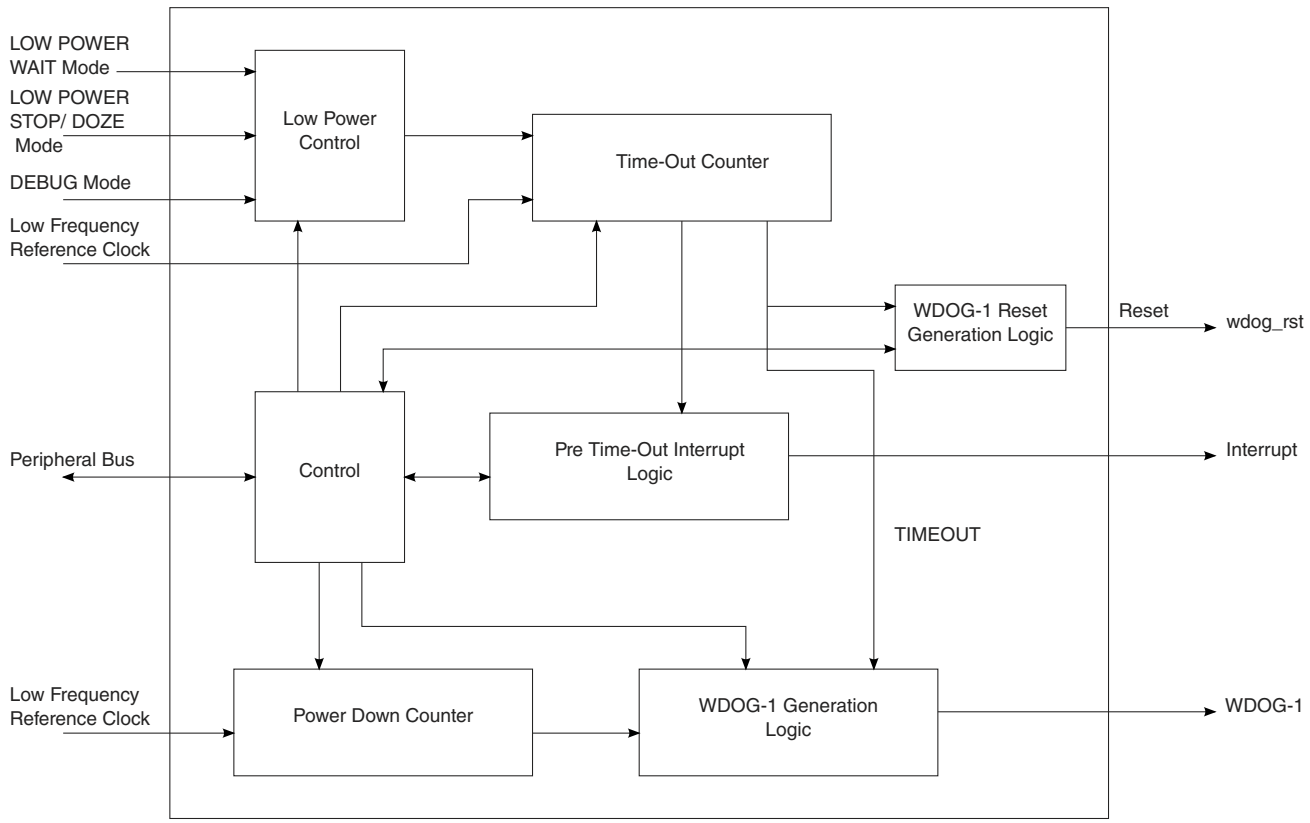


Figure 6-33. WDOG Diagram

6.6.1.2 Features

The WDOG features are listed below:

- Configurable timeout counter with timeout periods from 0.5 to 128 seconds which, after timeout expiration, result in the assertion of WDOG_RESET_B_DEB reset signal.
- Time resolution of 0.5 seconds
- Configurable timeout counter that can be programmed to run or stop during low-power modes
- Configurable timeout counter that can be programmed to run or stop during DEBUG mode
- Programmable interrupt generation prior to timeout
- The duration between interrupt and timeout events can be programmed from 0 to 127.5 seconds in steps of 0.5 seconds.

- Power down counter with fixed timeout period of 16 seconds, which if not disabled after reset will assert WDOG_B (ipp_wdog) signal low
- Power down counter will be enabled out of any reset (POR, Warm / Cold reset) by default.

6.6.2 Functional description

This section provides a complete functional description of the block.

6.6.2.1 Watchdog mechanism and system integration

The modules are disabled by default (after reset). WDOG1 will be configured during boot while WDOG2 is dedicated for secure world purposes and will be activated by TZ software if required. The TZ watchdog (TZ watchdog) module protects against TZ starvation by providing a method of escaping normal mode and forcing a switch to the TZ mode.TZ starvation is a situation where the normal OS prevents switching to the TZ mode. Such a situation is undesirable as it can compromise the system's security.

Once the TZ WDOG module is activated, it must be serviced by TZ on a periodic basis. If servicing does not take place, the timer times out. Upon a timeout, the TZ WDOG asserts a TZ-mapped interrupt that forces switching to the TZ mode. If it is still not serviced, the TZ WDOG asserts a security violation signal to the CSU. The TZ WDOG module cannot be programmed or de-activated by normal mode software.

The WDOG modules operate as follows:

- If servicing does not take place, the timer times out and the wdog_rst_b signal is activated (low)
- Interrupt can be generated before the counter actually times out
- The wdog_rst_b signal can be activated by software
- There is a power-down counter which gets enabled out of any reset. This counter has a fixed timeout period of 16 seconds upon which it will assert the ipp_wdog_b signal.

6.6.2.2 Timeout event

The WDOG provides timeout periods from 0.5 to 128 seconds with a time resolution of 0.5 seconds.

The user can determine the timeout period by writing to the WDOG timeout field (WT[7:0]) in the Watchdog Control Register (WCR). The WDOG must be enabled by setting the WDE bit of Watchdog Control Register (WCR) for the timeout counter to start running. After the WDOG is enabled, the counter is activated, loads the timeout value and begins to count down from this programmed value. The timer will time out when the counter reaches zero and the WDOG outputs a system reset signal, WDOG_RESET_B_DEB and asserts WDOG_B (ipp_wdog_b) (WDT bit should be set in Watchdog Control Register (WCR)).

However, the timeout condition can be prevented by reloading the counter with the new timeout value (WT[7:0] of WDOG_WCR) if a service routine (see Servicing WDOG to reload the counter) is performed before the counter reaches zero. If any system errors occur which prevent the software from servicing the Watchdog Service Register (WSR), the timeout condition occurs. By performing the service routine, the WDOG reloads its counter to the timeout value indicated by bits WT[7:0] of the Watchdog Control Register (WCR) and it restarts the countdown.

A system reset will reset the counter and place it in the idle state at any time during the countdown. The counter flow diagram is shown in Flow Diagrams.

NOTE

The timeout value is reloaded to the counter either at the time WDOG is enabled or after the service routine has been performed.

6.6.2.2.1 Servicing WDOG to reload the counter

To reload a timeout value to the counter, the proper service sequence begins by writing 0x_5555 followed by 0x_AAAA to the Watchdog Service Register (WSR). Any number of instructions can be executed between the two writes. If the WDOG_WSR is not loaded with 0x_5555 prior to writing 0x_AAAA to the WDOG_WSR, the counter is not reloaded. If any value other than 0x_AAAA is written to the WDOG_WSR after 0x_5555, the counter is not reloaded. This service sequence will reload the counter with the timeout value WT[7:0] of Watchdog Control Register (WCR). The timeout value can be changed at any point; it is reloaded when WDOG is serviced by the core.

6.6.2.3 Interrupt event

Prior to timeout, the WDOG can generate an interrupt which can be considered a warning that timeout will occur shortly.

The duration between interrupt event and timeout event can be controlled by writing to the WICT field of Watchdog Interrupt Control Register (WICR). It can vary between 0 and 127.5 seconds. If the WDOG is serviced (Servicing WDOG to reload the counter) before the interrupt generation, the counter will be reloaded with the timeout value WT[7:0] of Watchdog Control Register (WCR) and the interrupt will not be triggered.

6.6.2.4 Power-down counter event

The power-down counter inside WDOG will be enabled out of reset. This counter has a fixed timeout value of 16 seconds, after which it will drive the WDOG_B (ipp_wdog) signal low.

To prevent this, the software must disable this counter by clearing the PDE bit of Watchdog Miscellaneous Control Register (WMCR) within 16 seconds of reset deassertion. Once disabled, this counter can't be enabled again until the next system reset occurs. This feature is intended to prevent the hanging up of cores after reset, as WDOG is not enabled out of reset.

6.6.2.5 Low power modes

6.6.2.5.1 STOP and DOZE mode

If the WDOG timer disable bit for low power STOP and DOZE mode (WDZST) bit in the Watchdog Control Register (WCR), is cleared, the WDOG timer continues to operate using the low frequency reference clock. If the low power enable (WDZST) bit is set, the WDOG timer operation will be suspended in low power STOP or DOZE mode. Upon exiting low power STOP or DOZE mode, the WDOG operation returns to what it was prior to entering the STOP or DOZE mode.

6.6.2.5.2 WAIT mode

If the WDOG timer disable bit for low power WAIT mode (WDW) bit in the Watchdog Control Register (WCR), is cleared, the WDOG timer continues to operate using the low frequency reference clock i.e *ipg_clk_32k* clock (32.768 kHz frequency clock). If the low power WAIT enable (WDW) bit is set, the WDOG timer operation will be suspended. Upon exiting low power WAIT mode, the WDOG operation returns to what it was prior to entering the WAIT mode.

NOTE

The WDOG timer won't be able to detect events that happen for periods shorter than one low frequency reference clock cycle.

For example, in repeated WAIT mode entry or exit, if the RUN mode time is less than one low frequency reference clock cycle and if the WDW bit is set, the WDOG timer may never time out, even though the system is in RUN mode for a finite duration; WDOG may not see a low frequency reference clock edge during its wake time.

6.6.2.6 Debug mode

The WDOG timer can be configured for continual operation, or for suspension during debug mode. If the WDOG debug enable (WDBG) bit is set in the Watchdog Control Register (WCR), the WDOG timer operation is suspended in debug mode. If the WDBG bit is set and the debug mode (*ipg_debug* signal assertion) is entered, WDOG timer operation is suspended after two low frequency reference (*ipg_clk_32k*) clocks. Similarly, WDOG timer operation continues after two low frequency reference clocks of debug mode exit. Register read and write accesses in debug mode continue to function normally. Also, while in debug mode, the WDE bit of Watchdog Control Register (WCR) can be enabled/disabled directly. If the WDOG debug enable (WDBG) bit is cleared then WDOG timer operation is not suspended. The power-down counter is not affected by debug mode entry/exit.

NOTE

If the WDE bit of Watchdog Control Register (WCR) is set/cleared while in debug mode, it remains set/cleared even after exiting debug mode.

6.6.2.7 Operations

6.6.2.7.1 Watchdog reset generation

The WDOG generated reset signal WDOG_RESET_B_DEB is asserted by the following operations:

- A software write to the Software Reset Signal (SRS) bit of the Watchdog Control Register (WCR).
- WDOG timeout. See Timeout event.

The wdog_rst_b will be asserted for one clock cycle of low frequency reference clock for both a timeout condition and a software write occurrence. It remains asserted for 1 clock cycle of low frequency reference clock even if a system reset is asserted in between. Figure 6-35 shows the timing diagram of this signal due to a timeout condition.

6.6.2.7.2 WDOG_B generation

The WDOG asserts WDOG_B in the following scenarios:

- Software write to WDA bit of Watchdog Control Register (WCR). WDOG_B signal remains asserted as long as the WDA bit is "0".
- WDOG timeout condition, WDT bit of Watchdog Control Register (WCR) must be set for this scenario. A description of the timeout condition can be found in the Timeout event. WDOG_B (ipp_wdog) signal remains asserted until a power-on reset (POR) occurs. It gets cleared after the POR occurs (not due to any other system reset). Figure 6-36 shows the timing diagram of WDOG_B (ipp_wdog) due to timeout condition.
- WDOG power-down counter timeout, PDE bit of Watchdog Miscellaneous Control Register (WMCR) should not be cleared for this scenario. A description of this counter can be found in the Power-down counter event. WDOG_B (ipp_wdog) signal remains asserted for one clock cycle of low frequency reference clock (*ipg_clk_32k*).

Figure 6-34 shows the scenarios under which WDOG_B(ipp_wdog_b) gets asserted.

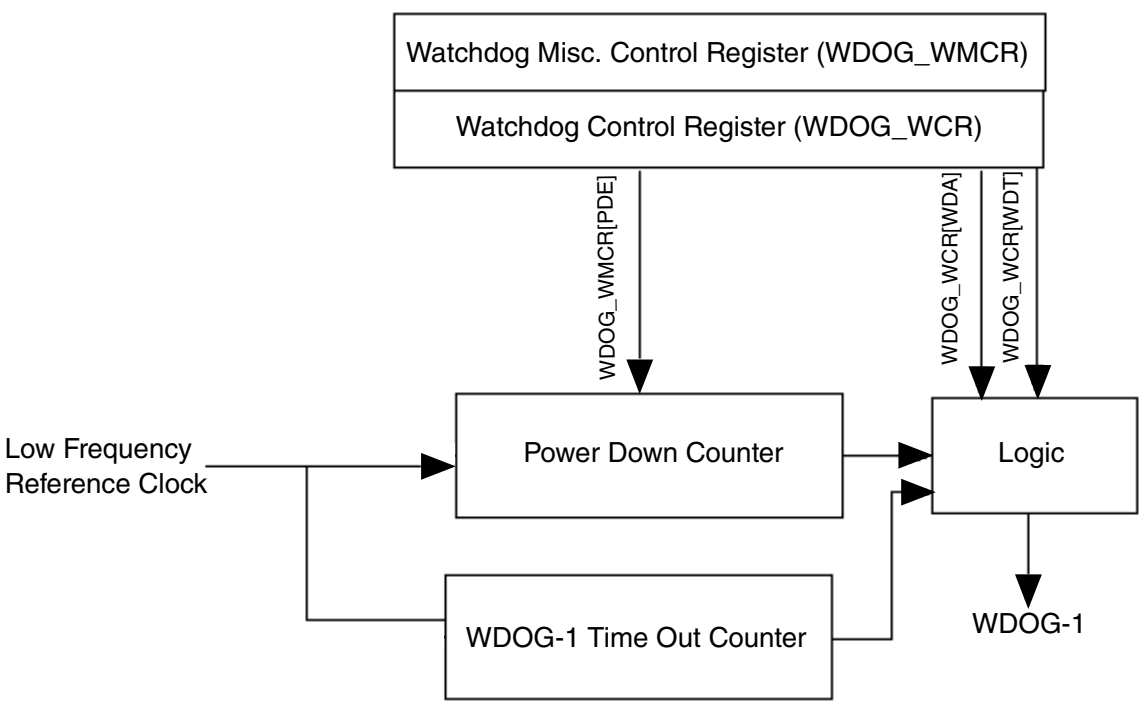


Figure 6-34. WDOG_B (ipp_wdog) generation

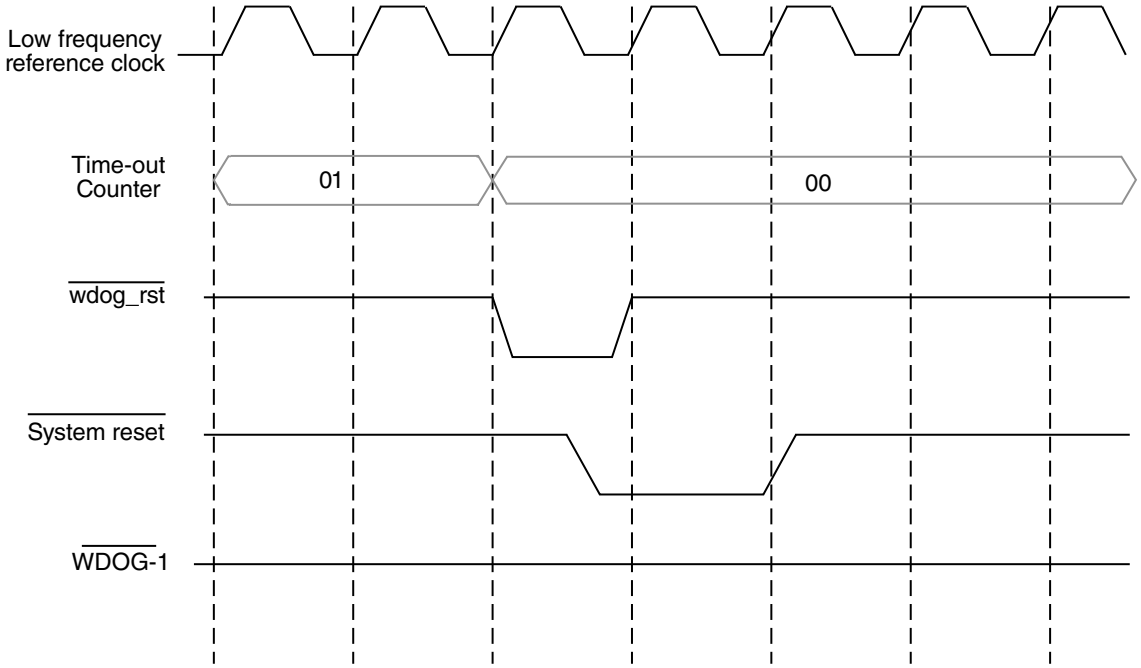


Figure 6-35. WDOG timeout condition/WDT bit is not set

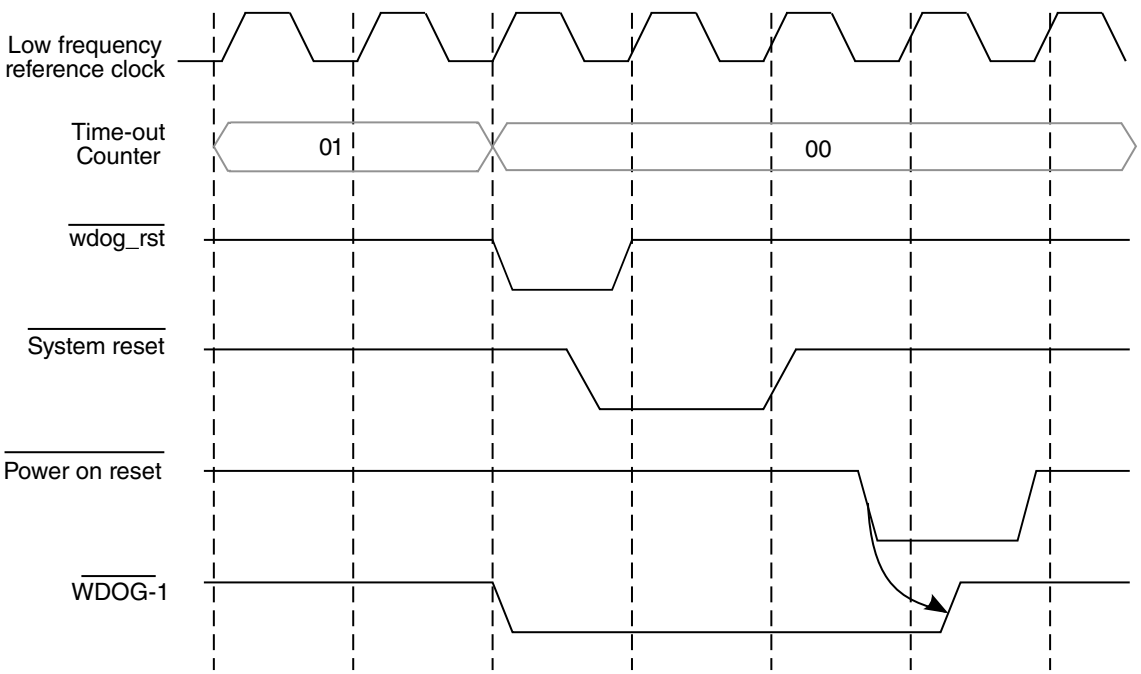


Figure 6-36. WDOG timeout condition/WDT bit is set

6.6.2.8 Clocks

This section describes clocks and special clocking requirements of the block.

The WDOG uses the low frequency reference clock for its counter and control operations. The peripheral bus clock is used for register read/write operations.

The following table describes the clock sources for WDOG. Please see for clock setting, configuration and gating information.

Table 6-43. WDOG Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	IP Global functional clock. All functionality inside the WDOG module is synchronized to this clock.
ipg_clk_s	ipg_clk_root	IP slave bus clock. This clock is synchronized to ipg_clk and is only used for register read/write operations.
ipg_clk_32k	ckil_sync_clk_root	Low frequency (32.768 kHz) clock that continues to run in low-power mode. It is assumed that the Clock Controller will provide this clock signal synchronized to ipg_clk in the normal mode, and switch to a non-synchronized signal in low-power mode when the ipg_clk is off.

6.6.2.9 Reset

The block is reset by a system reset and the WDOG counter will be disabled. The power-down counter is enabled and starts counting.

6.6.2.10 Interrupt

The WDOG has the feature of Interrupt generation before timeout.

The interrupt will be generated only if the WIE bit in Watchdog Interrupt Control Register (WICR) is set. The exact time at which the interrupt should occur (prior to timeout) depends on the value of WICT field of Watchdog Interrupt Control Register (WICR). For example, if the WICT field has a value 0x04, then the interrupt will be generated two seconds prior to timeout. Once the interrupt is triggered the WTIS bit in Watchdog Interrupt Control Register (WICR) will be set. The software needs to clear this bit to deassert the interrupt. If the WDOG is serviced before the interrupt generation then the counter will be reloaded with the timeout value WT[7:0] of Watchdog Control Register (WCR) and interrupt would not be triggered.

6.6.2.11 Flow Diagrams

The flow diagrams for WDOG operation are shown below.

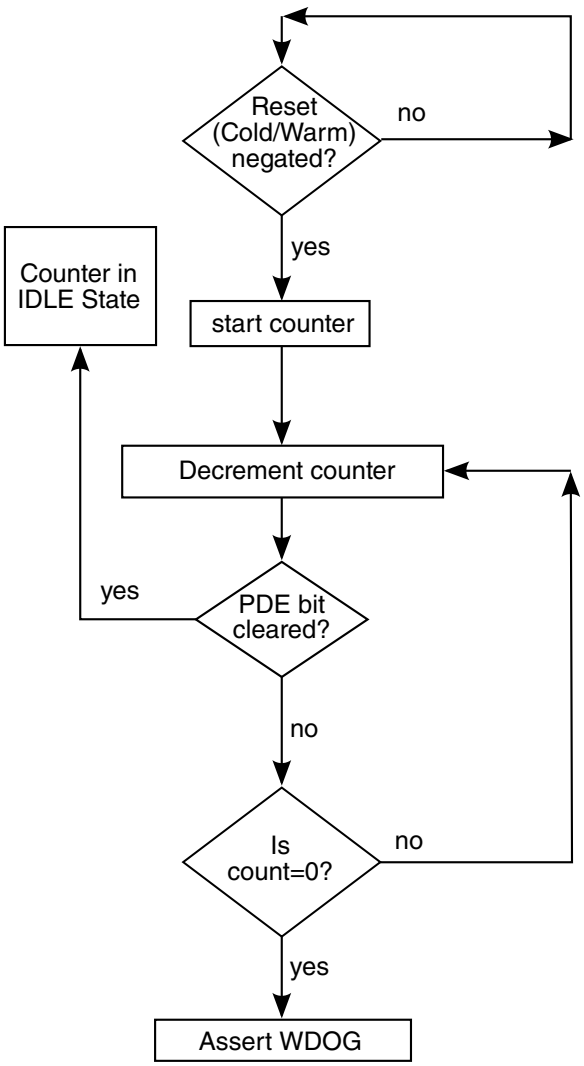


Figure 6-37. Power-Down Counter Flow Diagram

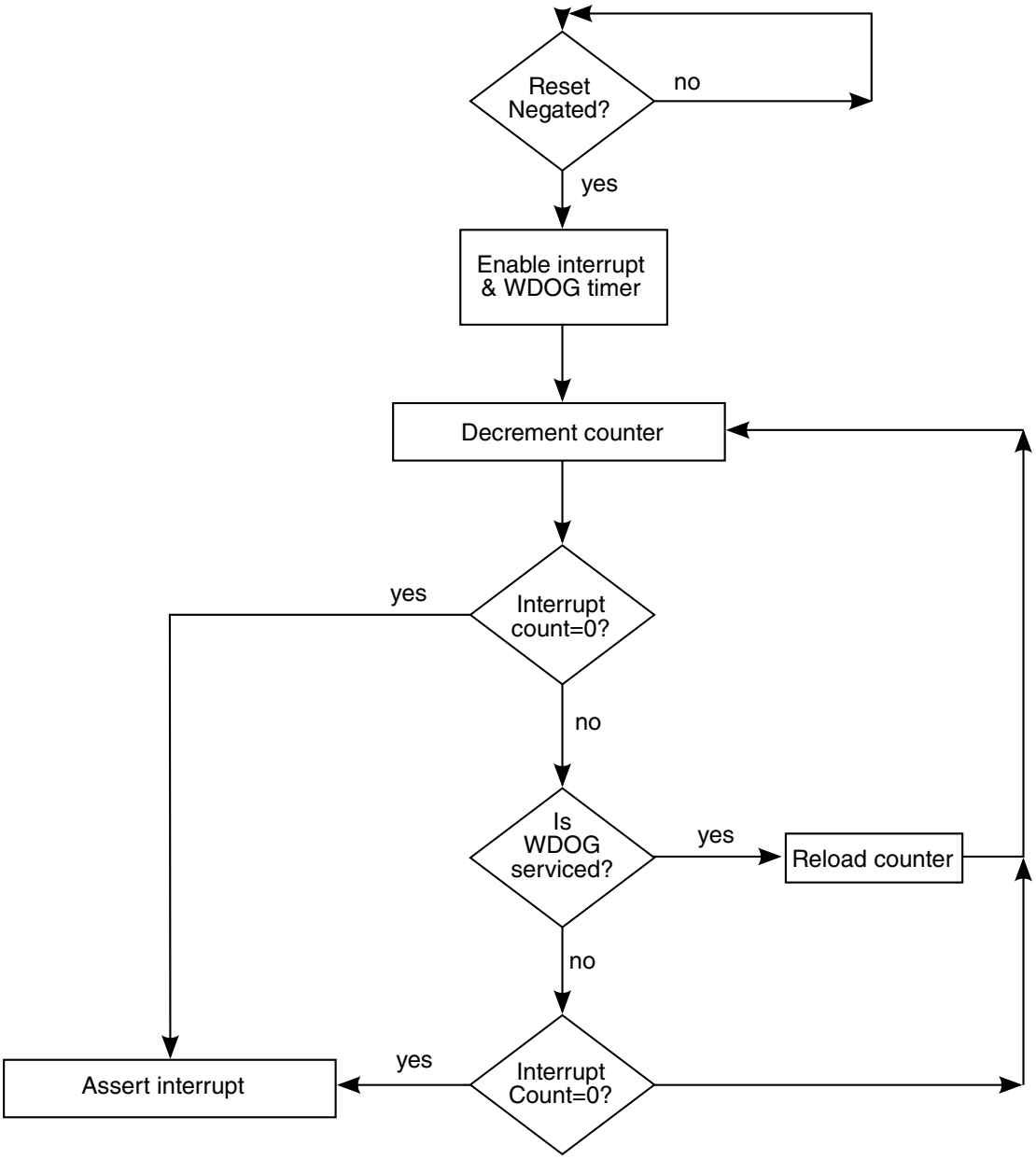


Figure 6-38. Interrupt Generation Flow Diagram

6.6.3 External signals

Table 6-44. WDOG External Signals

Signal	Description	Direction
WDOG_ANY	Global WDOG signal	IO
WDOG_B	This signal will power down the chip.	IO
WDOG_RESET_B_DEB	This signal is a reset source for the chip.	O

6.6.4 Initialization

The following sequence should be performed for WDOG initialization.

- PDE bit of Watchdog Miscellaneous Control Register (WMCR) should be cleared to disable the power down counter.
- WT field of Watchdog Control Register (WCR) should be programmed for sufficient timeout value.
- WDOG should be enabled by setting WDE bit of Watchdog Control Register (WCR) so that the timeout counter loads the WT field value of Watchdog Control Register (WCR) and starts counting.

6.6.5 WDOG Memory Map/Register Definition

6.6.5.1 WDOG register descriptions

The WDOG has user-accessible, 16-bit registers used to configure, operate, and monitor the state of the Watchdog Timer. Byte operations can be performed on these registers. If a 32-bit access is performed,the WDOG will not generate a peripheral bus error but will behave normally, like a 16-Bit access, making read/write possible. A 32-Bit access should be avoided, as the system may go to an unknown state.

6.6.5.1.1 WDOG memory map

WDOG1 base address: 3028_0000h

WDOG2 base address: 3029_0000h

WDOG3 base address: 302A_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Watchdog Control Register (WCR)	16	RW	0030
2	Watchdog Service Register (WSR)	16	RW	0000
4	Watchdog Reset Status Register (WRSR)	16	RO	0000
6	Watchdog Interrupt Control Register (WICR)	16	RW	0004
8	Watchdog Miscellaneous Control Register (WMCR)	16	RW	0001

6.6.5.1.2 Watchdog Control Register (WCR)

The Watchdog Control Register (WDOG_WCR) controls the WDOG operation.

- WDZST, WDBG and WDW are write-once only bits. Once the software does a write access to these bits, they will be locked and cannot be reprogrammed until the next system reset assertion.
- WDE is a write one once only bit. Once software performs a write "1" operation to this bit it cannot be reset/cleared until the next system reset.
- WDT is also a write one once only bit. Once software performs a write "1" operation to this bit it cannot be reset/cleared until the next POR. This bit does not get reset/cleared due to any system reset.

6.6.5.1.2.1 Offset

Register	Offset
WCR	0h

6.6.5.1.2.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WT								WDW	SRE	WDA	SRS	WDT	WDE	WDBG	WDZST
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

6.6.5.1.2.3 Fields

Field	Description
15-8 WT	WT Watchdog Time-out Field. This 8-bit field contains the time-out value that is loaded into the Watchdog counter after the service routine has been performed or after the Watchdog is enabled. After reset, WT[7:0] must have a value written to it before enabling the Watchdog otherwise count value of zero which is 0.5 seconds is loaded into the counter. **NOTE:** The time-out value can be written at any point of time but it is loaded to the counter at the time when WDOG is enabled or after the service routine has been performed. For more information see Timeout event . 00000000 - - 0.5 Seconds (Default). 00000001 - - 1.0 Seconds.

Table continues on the next page...

Field	Description
	00000010 - - 1.5 Seconds. 00000011 - - 2.0 Seconds. 11111111 - - 128 Seconds.
7 WDW	WDW Watchdog Disable for Wait. This bit determines the operation of WDOG during Low Power WAIT mode. This is a write once only bit. 0 - Continue WDOG timer operation (Default). 1 - Suspend WDOG timer operation.
6 SRE	Software Reset Extension. Required to be set to 1 when used in conjunction with the Software Reset Signal (SRS). 0 - Reserved 1 - This bit must be set to 1.
5 WDA	WDA WDOG_B assertion. Controls the software assertion of the WDOG_B signal. 0 - Assert WDOG_B output. 1 - No effect on system (Default).
4 SRS	SRS Software Reset Signal. Controls the software assertion of the WDOG-generated reset signal WDOG_RESET_B_DEB . This bit automatically resets to "1" after it has been asserted to "0". **NOTE:** This bit does not generate the software reset to the block. 0 - Assert system reset signal. 1 - No effect on the system (Default).
3 WDT	WDT WDOG_B Time-out assertion. Determines if the WDOG_B gets asserted upon a Watchdog Time-out Event. This is a write-one once only bit. **NOTE:** There is no effect on WDOG_RESET_B_DEB (WDOG Reset) upon writing on this bit. WDOG_B gets asserted along with WDOG_RESET_B_DEB if this bit is set. 0 - No effect on WDOG_B (Default). 1 - Assert WDOG_B upon a Watchdog Time-out event.
2 WDE	WDE Watchdog Enable. Enables or disables the WDOG block. This is a write one once only bit. It is not possible to clear this bit by a software write, once the bit is set. **NOTE:** This bit can be set/reset in debug mode (exception). 0 - Disable the Watchdog (Default). 1 - Enable the Watchdog.
1 WDBG	WDBG Watchdog DEBUG Enable. Determines the operation of the WDOG during DEBUG mode. This bit is write once only. 0 - Continue WDOG timer operation (Default). 1 - Suspend the watchdog timer.
0	WDZST

Field	Description
WDZST	Watchdog Low Power. Determines the operation of the WDOG during low-power modes. This bit is write once-only. **NOTE:** The WDOG can continue/suspend the timer operation in the low-power modes (STOP and DOZE mode). 0 - Continue timer operation (Default). 1 - Suspend the watchdog timer.

6.6.5.1.3 Watchdog Service Register (WSR)

When enabled, the WDOG requires that a service sequence be written to the Watchdog Service Register (WSR) to prevent the timeout condition.

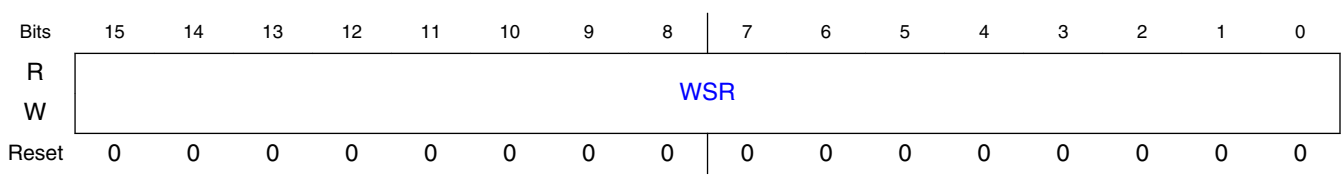
NOTE

Executing the service sequence will reload the WDOG timeout counter.

6.6.5.1.3.1 Offset

Register	Offset
WSR	2h

6.6.5.1.3.2 Diagram



6.6.5.1.3.3 Fields

Field	Description
15-0 WSR	WSR Watchdog Service Register. This 16-bit field contains the Watchdog service sequence. Both writes must occur in the order listed prior to the time-out, but any number of instructions can be executed between the two writes. The service sequence must be performed as follows: 0101010101010101 - Write to the Watchdog Service Register (WDOG_WSR). 1010101010101010 - Write to the Watchdog Service Register (WDOG_WSR).

6.6.5.1.4 Watchdog Reset Status Register (WRSR)

The WRSR is a read-only register that records the source of the output reset assertion. It is not cleared by a hard reset. Therefore, only one bit in the WRSR will always be asserted high. The register will always indicate the source of the last reset generated due to WDOG. Read access to this register is with one wait state. Any write performed on this register will generate a Peripheral Bus Error .

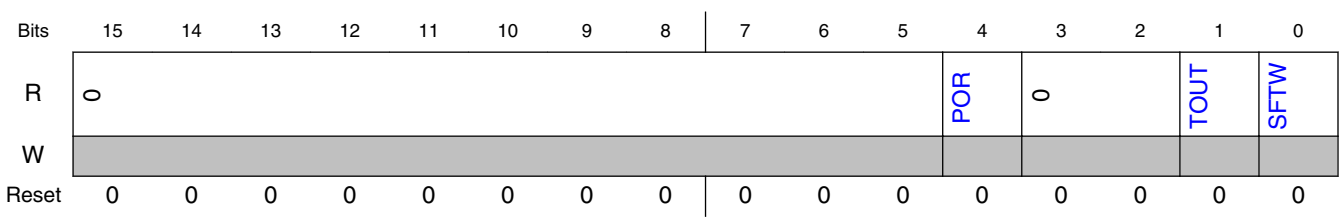
A reset can be generated by the following sources, as listed in priority from highest to lowest:

- Watchdog Time-out
- Software Reset

6.6.5.1.4.1 Offset

Register	Offset
WRSR	4h

6.6.5.1.4.2 Diagram



6.6.5.1.4.3 Fields

Field	Description
15-5 —	Reserved.
4 POR	POR Power On Reset. Indicates whether the reset is the result of a power on reset. 0 - Reset is not the result of a power on reset. 1 - Reset is the result of a power on reset.

Table continues on the next page...

Field	Description
3-2 —	Reserved.
1 TOUT	TOUT Timeout. Indicates whether the reset is the result of a WDOG timeout. 0 - Reset is not the result of a WDOG timeout. 1 - Reset is the result of a WDOG timeout.
0 SFTW	SFTW Software Reset. Indicates whether the reset is the result of a WDOG software reset by asserting SRS bit 0 - Reset is not the result of a software reset. 1 - Reset is the result of a software reset.

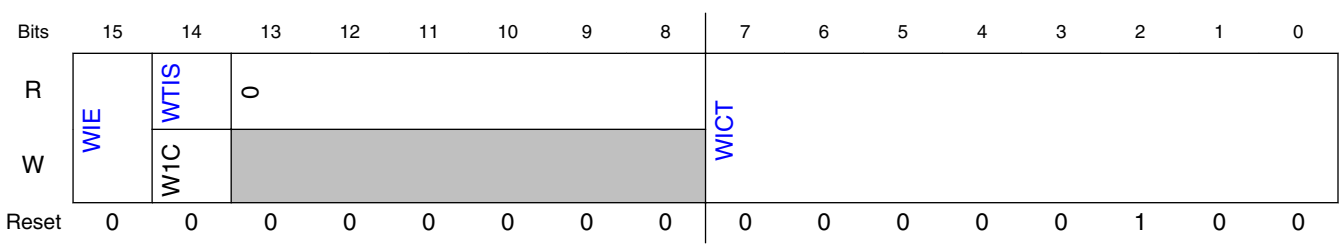
6.6.5.1.5 Watchdog Interrupt Control Register (WICR)

The WDOG_WICR controls the WDOG interrupt generation.

6.6.5.1.5.1 Offset

Register	Offset
WICR	6h

6.6.5.1.5.2 Diagram



6.6.5.1.5.3 Fields

Field	Description
15 WIE	WIE Watchdog Timer Interrupt enable bit. Reset value is 0.

Table continues on the next page...

Field	Description
	NOTE: This bit is a write once only bit. Once the software does a write access to this bit, it will get locked and cannot be reprogrammed until the next system reset assertion 0 - Disable Interrupt (Default). 1 - Enable Interrupt.
14 WTIS	WTIS Watchdog TImer Interrupt Status bit will reflect the timer interrupt status, whether interrupt has occurred or not. Once the interrupt has been triggered software must clear this bit by writing 1 to it. 0 - No interrupt has occurred (Default). 1 - Interrupt has occurred
13-8 —	Reserved.
7-0 WICT	WICT Watchdog Interrupt Count Time-out (WICT) field determines, how long before the counter time-out must the interrupt occur. The reset value is 0x04 implies interrupt will occur 2 seconds before time-out. The maximum value that can be programmed to WICT field is 127.5 seconds with a resolution of 0.5 seconds. **NOTE:** This field is write once only. Once the software does a write access to this field, it will get locked and cannot be reprogrammed until the next system reset assertion. 00000000 - WICT[7:0] = Time duration between interrupt and time-out is 0 seconds. 00000001 - WICT[7:0] = Time duration between interrupt and time-out is 0.5 seconds. 00000100 - WICT[7:0] = Time duration between interrupt and time-out is 2 seconds (Default). 11111111 - WICT[7:0] = Time duration between interrupt and time-out is 127.5 seconds.

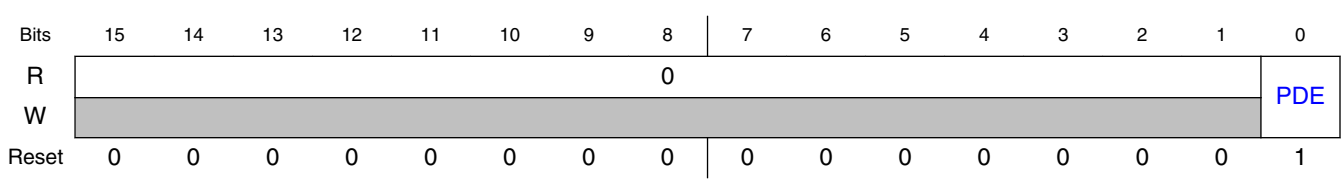
6.6.5.1.6 Watchdog Miscellaneous Control Register (WMCR)

WDOG_WMCR Controls the Power Down counter operation.

6.6.5.1.6.1 Offset

Register	Offset
WMCR	8h

6.6.5.1.6.2 Diagram



6.6.5.1.6.3 Fields

Field	Description
15-1 —	Reserved.
0 PDE	PDE Power Down Enable bit. Reset value of this bit is 1, which means the power down counter inside the WDOG is enabled after reset. The software must write 0 to this bit to disable the counter within 16 seconds of reset de-assertion. Once disabled this counter cannot be enabled again. See Power-down counter event for operation of this counter. **NOTE:** This bit is write-one once only bit. Once software sets this bit it cannot be reset until the next system reset. 0 - Power Down Counter of WDOG is disabled. 1 - Power Down Counter of WDOG is enabled (Default).

Chapter 7
Interrupts and DMA

7.1 Interrupts and DMA Events

7.1.1 Overview

This chapter provides the interrupt assignments of the Arm domain in A53 Interrupts, CM7 Interrupts, and the DMA events in SDMA event mapping

NOTE

JPEG is not supported.

7.1.2 A53 Interrupts

The Global Interrupt Controller (GIC) collects up to 160 interrupt requests from all the chip sources and provides an interface to the Cortex A53 CPU.

Each interrupt can be configured as a normal or a secure interrupt. Software force registers and software priority masking are also supported. The following table describes the A53 interrupt sources.

Table 7-1. Arm Domain Interrupt Summary

IRQ	Module	Logic	Interrupt Description
0	Boot	-	Used to notify cores on exception condition while boot
1	DAP	-	DAP Interrupt
2	SDMA1	-	AND of all 48 SDMA1 interrupts (events) from all the channels
3	GPU3D	-	GPU3D Interrupt
4	SNVS_LP_WRAPPER	OR	ON-OFF button press shorter than 5 secs (pulse event)

Table continues on the next page...

Table 7-1. Arm Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
4	SNVS_HP_WRAPPER	OR	ON-OFF button press shorter than 5 secs (pulse event)
5	LCDIF1	-	LCDIF1 Interrupt
6	LCDIF2	-	LCDIF2 Interrupt
7	VPU	-	VPU G1 Decoder Interrupt
8	VPU	-	VPU G2 Decoder Interrupt
9	QOS	-	QOS Interrupt
10	WDOG3	-	Watchdog Timer reset
11	HS	-	HS Interrupt Request
12	APBHDMA	OR4	GPMI operation channel 0 description complete interrupt
12	APBHDMA	OR4	GPMI operation channel 1 description complete interrupt
12	APBHDMA	OR4	GPMI operation channel 2 description complete interrupt
12	APBHDMA	OR4	GPMI operation channel 3 description complete interrupt
13	NPU	-	Machine Learning Processor Interrupt
14	RAWNAND	-	BCH operation complete interrupt
15	RAWNAND	-	GPMI operation TIMEOUT ERROR interrupt
16	ISI	-	ISI Camera Channel 0 Interrupt
17	MIPI_CSI1	-	MIPI CSI 1 Interrupt
18	MIPI_DSI	-	MIPI DSI Interrupt
19	SNVS_HP_WRAPPER	-	SRTC Consolidated Interrupt. Non TZ.
20	SNVS_HP_WRAPPER	-	SRTC Security Interrupt. TZ.
21	CSU	-	CSU Interrupt Request. Indicates to the processor that one or more alarm inputs were asserted
22	USDHC1	-	uSDHC1 Enhanced SDHC Interrupt Request
23	USDHC2	-	uSDHC2 Enhanced SDHC Interrupt Request
24	USDHC3	-	uSDHC3 Enhanced SDHC Interrupt Request
25	GPU2D	-	GPU2D Interrupt
26	UART1	-	UART-1 ORed interrupt
27	UART2	-	UART-2 ORed interrupt
28	UART3	-	UART-3 ORed interrupt
29	UART4	-	UART-4 ORed interrupt
30	VPU	-	VPU Encoder Interrupt
31	ECSPI1	-	eCSPI1 interrupt request line to the core.
32	ECSPI2	-	eCSPI2 interrupt request line to the core.
33	ECSPI3	-	eCSPI3 interrupt request line to the core.
34	SDMA3	-	AND of all 48 SDMA3 interrupts (events) from all the channels

Table continues on the next page...

Table 7-1. Arm Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
35	I2C1	-	I2C-1 Interrupt
36	I2C2	-	I2C-2 Interrupt
37	I2C3	-	I2C-3 Interrupt
38	I2C4	-	I2C-4 Interrupt
39	RDC	-	RDC Interrupt
40	USB1	-	USB-1 Interrupt
41	USB2	-	USB-2 Interrupt
42	ISI	-	ISI Camera Channel 1 Interrupt
43	HDMI_TX	-	HDMI TX Subsystem Interrupt
44	MICFIL	-	Digital Microphone interface voice activity detector event interrupt
45	MICFIL	-	Digital Microphone interface voice activity detector error interrupt
46	GPT6	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1, 2 & 3 Interrupt lines
47	SCTR	-	System Counter Interrupt [0]
48	SCTR	-	System Counter Interrupt [1]
49	ANAMIX	OR	TempSensor (Temperature alarm).
49	ANAMIX	OR	TempSensor (Temperature critical alarm).
50	SAI3	OR4	SAI3 Receive Interrupt
50	SAI3	OR4	SAI3 Receive Async Interrupt
50	SAI3	OR4	SAI3 Transmit Interrupt
50	SAI3	OR4	SAI3 Transmit Async Interrupt
51	GPT5	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1, 2 & 3 Interrupt lines
52	GPT4	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1, 2 & 3 Interrupt lines
53	GPT3	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1, 2 & 3 Interrupt lines
54	GPT2	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1, 2 & 3 Interrupt lines
55	GPT1	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1, 2 & 3 Interrupt lines
56	GPIO1	-	Active HIGH Interrupt from INT7 from GPIO
57	GPIO1	-	Active HIGH Interrupt from INT6 from GPIO
58	GPIO1	-	Active HIGH Interrupt from INT5 from GPIO
59	GPIO1	-	Active HIGH Interrupt from INT4 from GPIO

Table continues on the next page...

Table 7-1. Arm Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
60	GPIO1	-	Active HIGH Interrupt from INT3 from GPIO
61	GPIO1	-	Active HIGH Interrupt from INT2 from GPIO
62	GPIO1	-	Active HIGH Interrupt from INT1 from GPIO
63	GPIO1	-	Active HIGH Interrupt from INT0 from GPIO
64	GPIO1	-	Combined interrupt indication for GPIO1 signal 0 throughout 15
65	GPIO1	-	Combined interrupt indication for GPIO1 signal 16 throughout 31
66	GPIO2	-	Combined interrupt indication for GPIO2 signal 0 throughout 15
67	GPIO2	-	Combined interrupt indication for GPIO2 signal 16 throughout 31
68	GPIO3	-	Combined interrupt indication for GPIO3 signal 0 throughout 15
69	GPIO3	-	Combined interrupt indication for GPIO3 signal 16 throughout 31
70	GPIO4	-	Combined interrupt indication for GPIO4 signal 0 throughout 15
71	GPIO4	-	Combined interrupt indication for GPIO4 signal 16 throughout 31
72	GPIO5	-	Combined interrupt indication for GPIO5 signal 0 throughout 15
73	GPIO5	-	Combined interrupt indication for GPIO5 signal 16 throughout 31
74	ISP1	OR	ISP 1 ISP Interrupts
74	ISP1	OR	ISP 1 SMIA Interrupts
74	ISP1	OR	ISP 1 MIPI1 Interrupts
74	ISP1	OR	ISP 1 MIPI2 Interrupts
75	ISP2	OR	ISP 2 ISP Interrupts
75	ISP2	OR	ISP 2 SMIA Interrupts
75	ISP2	OR	ISP 2 MIPI1 Interrupts
75	ISP2	OR	ISP 2 MIPI2 Interrupts
76	I2C5	-	I2C-5 Interrupt
77	I2C6	-	I2C-6 Interrupt
78	WDOG1	-	Watchdog 1 Timer reset
79	WDOG2	-	Watchdog 2 Timer reset
80	MIPI_CSI2	-	MIPI CSI 2 Interrupt
81	PWM1	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
82	PWM2	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line

Table continues on the next page...

Table 7-1. Arm Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
83	PWM3	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
84	PWM4	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
85	CCMSRCGPCMIX	-	CCM, Interrupt Request 1
86	CCMSRCGPCMIX	-	CCM, Interrupt Request 2
87	CCMSRCGPCMIX	-	GPC, Interrupt Request 1
88	MU1	-	Interrupt to A53 (A53,M7 MU)
89	CCMSRCGPCMIX	-	SRC interrupt request
90	SAI5	OR4	SAI5 Receive Interrupt
90	SAI5	OR4	SAI5 Receive Async Interrupt
90	SAI5	OR4	SAI5 Transmit Interrupt
90	SAI5	OR4	SAI5 Transmit Async Interrupt
90	SAI6	OR4	SAI6 Receive Interrupt
90	SAI6	OR4	SAI6 Receive Async Interrupt
90	SAI6	OR4	SAI6 Transmit Interrupt
90	SAI6	OR4	SAI6 Transmit Async Interrupt
91	CAAM	-	RTIC Interrupt
92	CPU	OR	Performance Unit Interrupts from Quad-A53 platform
92	CPU	OR	Performance Unit Interrupts from Quad-A53 platform
92	CPU	OR	Performance Unit Interrupts from Quad-A53 platform
92	CPU	OR	Performance Unit Interrupts from Quad-A53 platform
93	CPU	OR	CTI trigger outputs from Quad-A53 platform
93	CPU	OR	CTI trigger outputs from Quad-A53 platform
93	CPU	OR	CTI trigger outputs from Quad-A53 platform
93	CPU	OR	CTI trigger outputs from Quad-A53 platform
94	CCMSRCGPCMIX	-	Combined CPU WDOG interrupts (4x) out of SRC.
95	SAI1	OR4	SAI1 Receive Interrupt
95	SAI1	OR4	SAI1 Receive Async Interrupt
95	SAI1	OR4	SAI1 Transmit Interrupt
95	SAI1	OR4	SAI1 Transmit Async Interrupt
96	SAI2	OR4	SAI2 Receive Interrupt
96	SAI2	OR4	SAI2 Receive Async Interrupt
96	SAI2	OR4	SAI2 Transmit Interrupt
96	SAI2	OR4	SAI2 Transmit Async Interrupt

Table continues on the next page...

Table 7-1. Arm Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
97	MU1	-	Interrupt to M7 (A53, M7 MU)
98	DDR	OR	DRAM controller Interrupt for DRAM controller's performance monitor
98	DDR	OR	DRAM controller Scrubber Interrupt indicating one full address range sweep
99	DDR	-	DRAM Controller Error Interrupt for DFI error
100	Dewarp	-	Dewarp Interrupt
101	CPU	-	Error indicator for AXI transaction with a write response error condition.
102	CPU	-	Error indicator for L2 RAM double-bit ECC error.
103	SDMA2	-	AND of all 48 SDMA2 interrupts (events) from all the channels
104	SJC	-	Interrupt triggered by SJC register
105	CAAM_WRAPPER	-	CAAM interrupt queue for JQ
106	CAAM_WRAPPER	-	CAAM interrupt queue for JQ
107	Flexspi	-	Flexspi Interrupt
108	TZASC	-	TZASC (PL380) interrupt
109	MICFIL	-	Digital Microphone interface interrupt
110	MICFIL	-	Digital Microphone interface error interrupt
111	SAI7	OR4	SAI7 Receive Interrupt
111	SAI7	OR4	SAI7 Receive Async Interrupt
111	SAI7	OR4	SAI7 Transmit Interrupt
111	SAI7	OR4	SAI7 Transmit Async Interrupt
112	perfmon1	-	General interrupt
113	perfmon2	-	General interrupt
114	CAAM_WRAPPER	-	CAAM interrupt queue for JQ
115	CAAM_WRAPPER	-	Recoverable error interrupt
116	HS	-	HS Interrupt Request
117	CM7	OR	CTI trigger outputs from CM7 platform
117	CM7	OR	CTI trigger outputs from CM7 platform
118	ENET1	OR4	MAC 0 Receive Buffer Done
118	ENET1	OR4	MAC 0 Receive Frame Done
118	ENET1	OR4	MAC 0 Transmit Buffer Done
118	ENET1	OR4	MAC 0 Transmit Frame Done
119	ENET1	OR4	MAC 0 Receive Buffer Done
119	ENET1	OR4	MAC 0 Receive Frame Done
119	ENET1	OR4	MAC 0 Transmit Buffer Done
119	ENET1	OR4	MAC 0 Transmit Frame Done
120	ENET1	OR	MAC 0 Periodic Timer Overflow
120	ENET1	OR	MAC 0 Time Stamp Available
120	ENET1	OR	MAC 0 Payload Receive Error

Table continues on the next page...

Table 7-1. Arm Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
120	ENET1	OR	MAC 0 Transmit FIFO Underrun
120	ENET1	OR	MAC 0 Collision Retry Limit
120	ENET1	OR	MAC 0 Late Collision
120	ENET1	OR	MAC 0 Ethernet Bus Error
120	ENET1	OR	MAC 0 MII Data Transfer Done
120	ENET1	OR	MAC 0 Receive Buffer Done
120	ENET1	OR	MAC 0 Receive Frame Done
120	ENET1	OR	MAC 0 Transmit Buffer Done
120	ENET1	OR	MAC 0 Transmit Frame Done
120	ENET1	OR	MAC 0 Graceful Stop
120	ENET1	OR	MAC 0 Babbling Transmit Error
120	ENET1	OR	MAC 0 Babbling Receive Error
120	ENET1	OR	MAC 0 Receive Flush Frame0
120	ENET1	OR	MAC 0 Receive Flush Frame1
120	ENET1	OR	MAC 0 Receive Flush Frame2
120	ENET1	OR	MAC 0 Wakeup Request (sync)
120	ENET1	OR	MAC 0 Babbling Receive Error
120	ENET1	OR	MAC 0 Wakeup Request (sync)
121	ENET1	-	MAC 0 1588 Timer Interrupt – synchronous
122	ASRC	-	ASRC Interrupt
123	PCIE_CTRL1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
124	PCIE_CTRL1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
125	PCIE_CTRL1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
126	PCIE_CTRL1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
127	PCIE_CTRL1	OR	Channels [63:32] interrupts requests
127	PCIE_CTRL1	OR	Channels [63:32] interrupts requests
128	AUDIO_XCVR	-	eARC Interrupt 0
129	AUDIO_XCVR	-	eARC Interrupt 1
130	AUD2HTX	-	Audio to HDMI TX Audio Link Master Interrupt
131	EDMA1	-	Audio Subsystem eDMA Error Interrupt
132	EDMA1	-	Audio Subsystem eDMA Channel Interrupts, Logical OR of channels [15:0]
133	EDMA1	-	Audio Subsystem eDMA Channel Interrupts, Logical OR of channels [31:16]
134	ENET_QOS	-	ENET QOS TSN Interrupt from PMT
135	ENET_QOS	OR4	ENET QOS TSN LPI RX exit Interrupt
135	ENET_QOS	OR4	ENET QOS TSN Host System Interrupt

Table continues on the next page...

Table 7-1. Arm Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
135	ENET_QOS	OR4	ENET QOS TSN Host System RX Channel Interrupts, Logical OR of channels[4:0]
135	ENET_QOS	OR4	ENET QOS TSN Host System TX Channel Interrupts, Logical OR of channels[4:0]
136	MU2	-	Interrupt to A53 (A53, Audio Processor MU)
137	MU2	-	Interrupt to Audio Processor (A53, Audio Processor MU)
138	MU3	-	Interrupt to M7 (M7, Audio Processor MU)
139	MU3	-	Interrupt to Audio Processor (M7, Audio Processor MU)
140	PCIE_CTRL1	-	RC/EP message transaction Interrupt
141	PCIE_CTRL1	-	RC/EP PME Message and Error Interrupt
142	CAN_FD1	OR	CAN-FD1 Interrupt from busoff
142	CAN_FD1	OR	CAN-FD1 Interrupt from CAN line error
142	CAN_FD1	OR	CAN-FD1 ORed interrupts from ipi_int_MB
142	CAN_FD1	OR	CAN-FD1 Rx warning Interrupt
142	CAN_FD1	OR	CAN-FD1 Tx warning Interrupt
142	CAN_FD1	OR	CAN-FD1 Interrupt from wake up
142	CAN_FD1	OR	CAN-FD1 Interrupt from match in PN
142	CAN_FD1	OR	CAN-FD1 Interrupt from timeout in PN
142	CAN_FD1	OR	CAN-FD1 Busoff done interrupt
142	CAN_FD1	OR	CAN-FD1 FD error interrupt
143	CAN_FD1	OR	CAN-FD1 Correctable error interrupt
143	CAN_FD1	OR	CAN-FD1 Non correctable error int host
143	CAN_FD1	OR	CAN-FD1 Non correctable error int internal
144	CAN_FD2	OR	CAN-FD2 Interrupt from busoff
144	CAN_FD2	OR	CAN-FD2 Interrupt from CAN line error
144	CAN_FD2	OR	CAN-FD2 ORed interrupts from ipi_int_MB
144	CAN_FD2	OR	CAN-FD2 Rx warning Interrupt
144	CAN_FD2	OR	CAN-FD2 Tx warning Interrupt
144	CAN_FD2	OR	CAN-FD2 Interrupt from wake up
144	CAN_FD2	OR	CAN-FD2 Interrupt from match in PN
144	CAN_FD2	OR	CAN-FD2 Interrupt from timeout in PN
144	CAN_FD2	OR	CAN-FD2 Busoff done interrupt
144	CAN_FD2	OR	CAN-FD2 FD error interrupt
145	CAN_FD2	OR	CAN-FD2 Correctable error interrupt
145	CAN_FD2	OR	CAN-FD2 Non correctable error int host
145	CAN_FD2	OR	CAN-FD2 Non correctable error int internal
146	AUDIO_XCVR	-	eARC PHY - SPDIF wakeup interrupt
147	DDR	OR	DRAM Controller Error Interrupt for address protection fault.

Table continues on the next page...

Table 7-1. Arm Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
147	DDR	OR	DRAM Controller Error Interrupt for correctable ECC error detected
147	DDR	OR	DRAM Controller Error Interrupt for uncorrectable ECC error detected
148	USB1	-	USB-1 Wake-up Interrupt
149	USB2	-	USB-2 Wake-up Interrupt
150-159	Reserved	-	Reserved

7.1.3 CM7 Interrupts

The Nested Vectored Interrupt Controller (NVIC) collects up to 160 interrupt requests from all chip sources and provides an interface to the Cortex M7 Core.

The following table describes the M7 interrupt sources.

Table 7-2. CM7 Interrupt Summary

IRQ	Module	Logic	Interrupt Description
0	Reserved	-	Reserved
1	DAP	-	DAP Interrupt
2	SDMA1	-	AND of all 48 SDMA1 interrupts (events) from all the channels
3	GPU3D	-	GPU3D Interrupt
4	SNVS_LP_WRAPPER	OR	ON-OFF button press shorter than 5 secs (pulse event)
4	SNVS_HP_WRAPPER	OR	ON-OFF button press shorter than 5 secs (pulse event)
5	LCDIF1	-	LCDIF1 Interrupt
6	LCDIF2	-	LCDIF2 Interrupt
7	VPU	-	VPU G1 Decoder Interrupt
8	VPU	-	VPU G2 Decoder Interrupt
9	QOS	-	QOS Interrupt
10	WDOG3	-	Watchdog Timer reset
11	HS_CP1	-	HS Interrupt Request (CP1_semaphore_int)
12	APBHDMA	OR4	GPMI operation channel 0 description complete interrupt
12	APBHDMA	OR4	GPMI operation channel 1 description complete interrupt
12	APBHDMA	OR4	GPMI operation channel 2 description complete interrupt

Table continues on the next page...

Table 7-2. CM7 Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
12	APBHDMA	OR4	GPMI operation channel 3 description complete interrupt
13	NPU	-	Machine Learning Processor Interrupt
14	RAWNAND	-	BCH operation complete interrupt
15	RAWNAND	-	GPMI operation TIMEOUT ERROR interrupt
16	ISI	-	ISI Camera Channel 0 Interrupt
17	MIPI_CSI1	-	MIPI CSI 1 Interrupt
18	MIPI_DSI	-	MIPI DSI Interrupt
19	SNVS_HP_WRAPPER	-	SRTC Consolidated Interrupt. Non TZ.
20	SNVS_HP_WRAPPER	-	SRTC Security Interrupt. TZ.
21	CSU	-	CSU Interrupt Request. Indicates to the processor that one or more alarm inputs were asserted
22	USDHC1	-	uSDHC1 Enhanced SDHC Interrupt Request
23	USDHC2	-	uSDHC2 Enhanced SDHC Interrupt Request
24	USDHC3	-	uSDHC3 Enhanced SDHC Interrupt Request
25	GPU2D	-	GPU2D Interrupt
26	UART1	-	UART-1 ORed interrupt
27	UART2	-	UART-2 ORed interrupt
28	UART3	-	UART-3 ORed interrupt
29	UART4	-	UART-4 ORed interrupt
30	VPU	-	VPU Encoder Interrupt
31	ECSPI1	-	eCSPI1 interrupt request line to the core.
32	ECSPI2	-	eCSPI2 interrupt request line to the core.
33	ECSPI3	-	eCSPI3 interrupt request line to the core.
34	SDMA3	-	AND of all 48 SDMA3 interrupts (events) from all the channels
35	I2C1	-	I2C-1 Interrupt
36	I2C2	-	I2C-2 Interrupt
37	I2C3	-	I2C-3 Interrupt
38	I2C4	-	I2C-4 Interrupt
39	RDC	-	RDC Interrupt
40	USB1	-	USB-1 Interrupt
41	USB2	-	USB-2 Interrupt
42	ISI	-	ISI Camera Channel 1 Interrupt
43	HDMI_TX	-	HDMI TX Subsystem Interrupt
44	MICFIL	-	Digital Microphone interface voice activity detector event interrupt
45	MICFIL	-	Digital Microphone interface voice activity detector error interrupt

Table continues on the next page...

Table 7-2. CM7 Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
46	GPT6	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1, 2 & 3 Interrupt lines
47	SCTR	-	System Counter Interrupt [0]
48	SCTR	-	System Counter Interrupt [1]
49	ANAMIX	OR	TempSensor (Temperature alarm).
49	ANAMIX	OR	TempSensor (Temperature critical alarm).
50	SAI3	OR4	SAI3 Receive Interrupt
50	SAI3	OR4	SAI3 Receive Async Interrupt
50	SAI3	OR4	SAI3 Transmit Interrupt
50	SAI3	OR4	SAI3 Transmit Async Interrupt
51	GPT5	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1, 2 & 3 Interrupt lines
52	GPT4	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1, 2 & 3 Interrupt lines
53	GPT3	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1, 2 & 3 Interrupt lines
54	GPT2	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1, 2 & 3 Interrupt lines
55	GPT1	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1, 2 & 3 Interrupt lines
56	GPIO1	-	Active HIGH Interrupt from INT7 from GPIO
57	GPIO1	-	Active HIGH Interrupt from INT6 from GPIO
58	GPIO1	-	Active HIGH Interrupt from INT5 from GPIO
59	GPIO1	-	Active HIGH Interrupt from INT4 from GPIO
60	GPIO1	-	Active HIGH Interrupt from INT3 from GPIO
61	GPIO1	-	Active HIGH Interrupt from INT2 from GPIO
62	GPIO1	-	Active HIGH Interrupt from INT1 from GPIO
63	GPIO1	-	Active HIGH Interrupt from INT0 from GPIO
64	GPIO1	-	Combined interrupt indication for GPIO1 signal 0 throughout 15
65	GPIO1	-	Combined interrupt indication for GPIO1 signal 16 throughout 31
66	GPIO2	-	Combined interrupt indication for GPIO2 signal 0 throughout 15
67	GPIO2	-	Combined interrupt indication for GPIO2 signal 16 throughout 31
68	GPIO3	-	Combined interrupt indication for GPIO3 signal 0 throughout 15

Table continues on the next page...

Table 7-2. CM7 Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
69	GPIO3	-	Combined interrupt indication for GPIO3 signal 16 throughout 31
70	GPIO4	-	Combined interrupt indication for GPIO4 signal 0 throughout 15
71	GPIO4	-	Combined interrupt indication for GPIO4 signal 16 throughout 31
72	GPIO5	-	Combined interrupt indication for GPIO5 signal 0 throughout 15
73	GPIO5	-	Combined interrupt indication for GPIO5 signal 16 throughout 31
74	ISP1	OR	ISP 1 ISP Interrupts
74	ISP1	OR	ISP 1 SMIA Interrupts
74	ISP1	OR	ISP 1 MIPI1 Interrupts
74	ISP1	OR	ISP 1 MIPI2 Interrupts
74	ISP1	OR	ISP 1 Memory Interface Interrupts
75	ISP2	OR	ISP 2 ISP Interrupts
75	ISP2	OR	ISP 2 SMIA Interrupts
75	ISP2	OR	ISP 2 MIPI1 Interrupts
75	ISP2	OR	ISP 2 MIPI2 Interrupts
75	ISP2	OR	ISP 2 Memory Interface Interrupts
76	I2C5	-	I2C-5 Interrupt
77	I2C6	-	I2C-6 Interrupt
78	WDOG1	-	Watchdog 1 Timer reset
79	WDOG2	-	Watchdog 2 Timer reset
80	MIPI_CSI2	-	MIPI CSI 2 Interrupt
81	PWM1	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
82	PWM2	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
83	PWM3	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
84	PWM4	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
85	CCMSRCGPCMIX	-	CCM, Interrupt Request 1
86	CCMSRCGPCMIX	-	CCM, Interrupt Request 2
87	CCMSRCGPCMIX	-	GPC, Interrupt Request 1
88	MU1	-	Interrupt to A53 (A53,M7 MU)
89	CCMSRCGPCMIX	-	SRC interrupt request
90	SAI5	OR4	SAI5 Receive Interrupt

Table continues on the next page...

Table 7-2. CM7 Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
90	SAI5	OR4	SAI5 Receive Async Interrupt
90	SAI5	OR4	SAI5 Transmit Interrupt
90	SAI5	OR4	SAI5 Transmit Async Interrupt
90	SAI6	OR4	SAI6 Receive Interrupt
90	SAI6	OR4	SAI6 Receive Async Interrupt
90	SAI6	OR4	SAI6 Transmit Interrupt
90	SAI6	OR4	SAI6 Transmit Async Interrupt
91	CAAM	-	RTIC Interrupt
92	CPU	OR	Performance Unit Interrupts from Quad-A53 platform
92	CPU	OR	Performance Unit Interrupts from Quad-A53 platform
92	CPU	OR	Performance Unit Interrupts from Quad-A53 platform
92	CPU	OR	Performance Unit Interrupts from Quad-A53 platform
93	CPU	OR	CTI trigger outputs from Quad-A53 platform
93	CPU	OR	CTI trigger outputs from Quad-A53 platform
93	CPU	OR	CTI trigger outputs from Quad-A53 platform
93	CPU	OR	CTI trigger outputs from Quad-A53 platform
94	CCMSRCGPCMIX	-	Combined CPU WDOG interrupts (4x) out of SRC.
95	SAI1	OR4	SAI1 Receive Interrupt
95	SAI1	OR4	SAI1 Receive Async Interrupt
95	SAI1	OR4	SAI1 Transmit Interrupt
95	SAI1	OR4	SAI1 Transmit Async Interrupt
96	SAI2	OR4	SAI2 Receive Interrupt
96	SAI2	OR4	SAI2 Receive Async Interrupt
96	SAI2	OR4	SAI2 Transmit Interrupt
96	SAI2	OR4	SAI2 Transmit Async Interrupt
97	MU1	-	Interrupt to M7 (A53, M7 MU)
98	DDR	OR	DRAM controller Interrupt for DRAM controller's performance monitor
98	DDR	OR	DRAM controller Scrubber Interrupt indicating one full address range sweep
99	DDR	-	DRAM Controller Error Interrupt for DFI error
100	Dewarp	-	Dewarp Interrupt
101	CPU	-	Error indicator for AXI transaction with a write response error condition.
102	CPU	-	Error indicator for L2 RAM double-bit ECC error.
103	SDMA2	-	AND of all 48 SDMA2 interrupts (events) from all the channels

Table continues on the next page...

Table 7-2. CM7 Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
104	SJC	-	Interrupt triggered by SJC register
105	CAAM_WRAPPER	-	CAAM interrupt queue for JQ
106	CAAM_WRAPPER	-	CAAM interrupt queue for JQ
107	Flexspi	-	Flexspi Interrupt
108	TZASC	-	TZASC (PL380) interrupt
109	MICFIL	-	Digital Microphone interface interrupt
110	MICFIL	-	Digital Microphone interface error interrupt
111	SAI7	OR4	SAI7 Receive Interrupt
111	SAI7	OR4	SAI7 Receive Async Interrupt
111	SAI7	OR4	SAI7 Transmit Interrupt
111	SAI7	OR4	SAI7 Transmit Async Interrupt
112	perfmon1	-	General interrupt
113	perfmon2	-	General interrupt
114	CAAM_WRAPPER	-	CAAM interrupt queue for JQ
115	CAAM_WRAPPER	-	Recoverable error interrupt
116	HS_CP0	-	HS Interrupt Request (CP0_semaphore_int)
117	CM7	OR	CTI trigger outputs from CM7 platform
117	CM7	OR	CTI trigger outputs from CM7 platform
118	ENET1	OR4	MAC 0 Receive Buffer Done
118	ENET1	OR4	MAC 0 Receive Frame Done
118	ENET1	OR4	MAC 0 Transmit Buffer Done
118	ENET1	OR4	MAC 0 Transmit Frame Done
119	ENET1	OR4	MAC 0 Receive Buffer Done
119	ENET1	OR4	MAC 0 Receive Frame Done
119	ENET1	OR4	MAC 0 Transmit Buffer Done
119	ENET1	OR4	MAC 0 Transmit Frame Done
120	ENET1	OR	MAC 0 Periodic Timer Overflow
120	ENET1	OR	MAC 0 Time Stamp Available
120	ENET1	OR	MAC 0 Payload Receive Error
120	ENET1	OR	MAC 0 Transmit FIFO Underrun
120	ENET1	OR	MAC 0 Collision Retry Limit
120	ENET1	OR	MAC 0 Late Collision
120	ENET1	OR	MAC 0 Ethernet Bus Error
120	ENET1	OR	MAC 0 MII Data Transfer Done
120	ENET1	OR	MAC 0 Receive Buffer Done
120	ENET1	OR	MAC 0 Receive Frame Done
120	ENET1	OR	MAC 0 Transmit Buffer Done
120	ENET1	OR	MAC 0 Transmit Frame Done
120	ENET1	OR	MAC 0 Graceful Stop

Table continues on the next page...

Table 7-2. CM7 Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
120	ENET1	OR	MAC 0 Babbling Transmit Error
120	ENET1	OR	MAC 0 Babbling Receive Error
120	ENET1	OR	MAC 0 Receive Flush Frame0
120	ENET1	OR	MAC 0 Receive Flush Frame1
120	ENET1	OR	MAC 0 Receive Flush Frame2
120	ENET1	OR	MAC 0 Wakeup Request (sync)
120	ENET1	OR	MAC 0 Babbling Receive Error
120	ENET1	OR	MAC 0 Wakeup Request (sync)
121	ENET1	-	MAC 0 1588 Timer Interrupt – synchronous
122	ASRC	-	ASRC Interrupt
123	PCIE_CTRL1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
124	PCIE_CTRL1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
125	PCIE_CTRL1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
126	PCIE_CTRL1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
127	PCIE_CTRL1	OR	Channels [63:32] interrupts requests
127	PCIE_CTRL1	OR	Channels [63:32] interrupts requests
128	AUDIO_XCVR	-	eARC Interrupt 0
129	AUDIO_XCVR	-	eARC Interrupt 1
130	AUD2HTX	-	Audio to HDMI TX Audio Link Master Interrupt
131	EDMA1	-	Audio Subsystem eDMA Error Interrupt
132	EDMA1	-	Audio Subsystem eDMA Channel Interrupts, Logical OR of channels [15:0]
133	EDMA1	-	Audio Subsystem eDMA Channel Interrupts, Logical OR of channels [31:16]
134	ENET_QOS	-	ENET QOS TSN Interrupt from PMT
135	ENET_QOS	OR4	ENET QOS TSN LPI RX exit Interrupt
135	ENET_QOS	OR4	ENET QOS TSN Host System Interrupt
135	ENET_QOS	OR4	ENET QOS TSN Host System RX Channel Interrupts, Logical OR of channels[4:0]
135	ENET_QOS	OR4	ENET QOS TSN Host System TX Channel Interrupts, Logical OR of channels[4:0]
136	MU2	-	Interrupt to A53 (A53, Audio Processor MU)
137	MU2	-	Interrupt to Audio Processor (A53, Audio Processor MU)
138	MU3	-	Interrupt to M7 (M7, Audio Processor MU)
139	MU3	-	Interrupt to Audio Processor (M7, Audio Processor MU)
140	PCIE_CTRL1	-	RC/EP message transaction Interrupt

Table continues on the next page...

Table 7-2. CM7 Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
141	PCIE_CTRL1	-	RC/EP PME Message and Error Interrupt
142	CAN_FD1	OR	CAN-FD1 Interrupt from busoff
142	CAN_FD1	OR	CAN-FD1 Interrupt from CAN line error
142	CAN_FD1	OR	CAN-FD1 ORed interrupts from ipi_int_MB
142	CAN_FD1	OR	CAN-FD1 Rx warning Interrupt
142	CAN_FD1	OR	CAN-FD1 Tx warning Interrupt
142	CAN_FD1	OR	CAN-FD1 Interrupt from wake up
142	CAN_FD1	OR	CAN-FD1 Interrupt from match in PN
142	CAN_FD1	OR	CAN-FD1 Interrupt from timeout in PN
142	CAN_FD1	OR	CAN-FD1 Busoff done interrupt
142	CAN_FD1	OR	CAN-FD1 FD error interrupt
143	CAN_FD1	OR	CAN-FD1 Correctable error interrupt
143	CAN_FD1	OR	CAN-FD1 Non correctable error int host
143	CAN_FD1	OR	CAN-FD1 Non correctable error int internal
144	CAN_FD2	OR	CAN-FD2 Interrupt from busoff
144	CAN_FD2	OR	CAN-FD2 Interrupt from CAN line error
144	CAN_FD2	OR	CAN-FD2 ORed interrupts from ipi_int_MB
144	CAN_FD2	OR	CAN-FD2 Rx warning Interrupt
144	CAN_FD2	OR	CAN-FD2 Tx warning Interrupt
144	CAN_FD2	OR	CAN-FD2 Interrupt from wake up
144	CAN_FD2	OR	CAN-FD2 Interrupt from match in PN
144	CAN_FD2	OR	CAN-FD2 Interrupt from timeout in PN
144	CAN_FD2	OR	CAN-FD2 Busoff done interrupt
144	CAN_FD2	OR	CAN-FD2 FD error interrupt
145	CAN_FD2	OR	CAN-FD2 Correctable error interrupt
145	CAN_FD2	OR	CAN-FD2 Non correctable error int host
145	CAN_FD2	OR	CAN-FD2 Non correctable error int internal
146	AUDIO_XCVR	-	eARC PHY - SPDIF wakeup interrupt
147	DDR	OR	DRAM Controller Error Interrupt for address protection fault.
147	DDR	OR	DRAM Controller Error Interrupt for correctable ECC error detected
147	DDR	OR	DRAM Controller Error Interrupt for uncorrectable ECC error detected
148	USB1	-	USB-1 Wake-up Interrupt
149	USB2	-	USB-2 Wake-up Interrupt
150-159	Reserved	-	Reserved

7.1.4 Audio DSP Interrupts

The Audio DSP has 32 interrupt inputs, configured as shown in the following table. Interrupts 19-27 come from the 3-channel Interrupt Steer (IRQ_STEER) block. These 9 interrupts from the SoC are generated by an instance of the IRQ_STEER, which is used to reduce the160 system interrupt request sources down to 9 interrupts for the Audio DSP.

Table 7-3. Audio DSP Interrupts

IRQ	Type	Level	Source	Description
0 - 1	Level	1	-	Reserved
2	Timer.0	2	Internal	-
3	Timer.1	3	Internal	-
4 - 6	Edge	2	-	Reserved
7	Level	2	MU2 and MU3	From MU2 and MU3 (the two MU instances for communication with the Audio DSP).
8	Sw	1	Internal	-
9	Sw	2	Internal	-
10	Writeerr	2	Internal	-
11 - 18	Level	2	-	Reserved
19	Level	2	irqstr_irq[0]	From IRQ_STEER
20	Level	2	irqstr_irq[1]	From IRQ_STEER
21	Level	2	irqstr_irq[2]	From IRQ_STEER
22	Level	2	irqstr_irq[3]	From IRQ_STEER
23	Level	2	irqstr_irq[4]	From IRQ_STEER
24	Level	2	irqstr_irq[5]	From IRQ_STEER
25	Level	2	irqstr_irq[6]	From IRQ_STEER
26	Level	2	irqstr_irq[7]	From IRQ_STEER
27	Level	2	irqstr_irq[8]	From IRQ_STEER
28 - 30	Level	2	-	Reserved
31	profiling	3	-	-

All the 3 channels of the IRQ_STEER module are mapped to the Audio DSP (HiFi4). The input and output IRQ mapping is shown in the table below.

Table 7-4. IRQ_STEER Mapping

Input IRQ	Channel #	Output IRQ	Audio DSP IRQ
IRQ_IN[31:0]	CH0	IRQ[0]	HIFI_IRQ[19]

Table continues on the next page...

Table 7-4. IRQ_STEER Mapping (continued)

Input IRQ	Channel #	Output IRQ	Audio DSP IRQ
IRQ_IN[63:32]		IRQ[1]	HIFI_IRQ[20]
IRQ_IN[95:64]			
IRQ_IN[127:96]		IRQ[2]	HIFI_IRQ[21]
IRQ_IN[159:128]			
IRQ_IN[31:0]	CH1	IRQ[3]	HIFI_IRQ[22]
IRQ_IN[63:32]		IRQ[4]	HIFI_IRQ[23]
IRQ_IN[95:64]			
IRQ_IN[127:96]		IRQ[5]	HIFI_IRQ[24]
IRQ_IN[159:128]			
IRQ_IN[31:0]	CH2	IRQ[6]	HIFI_IRQ[25]
IRQ_IN[63:32]		IRQ[7]	HIFI_IRQ[26]
IRQ_IN[95:64]			
IRQ_IN[127:96]		IRQ[8]	HIFI_IRQ[27]

7.1.5 HDMI Interrupts

The 1-channel IRQ_STEER module for the HDMI subsystem combines the subsystem's 15 internal interrupts into 1 external interrupt that is routed to the GIC (IRQ[43]).

The HDMI subsystem interrupts are mapped to the IRQ_STEER as shown in the table below.

Table 7-5. HDMI Interrupts

IRQ Source	IRQ_STEER Input	Description
u_hdmi_tx.ointerrupt	u_irq_steer.irq_in[0]	HDMI TX General Interrupt
u_hdmi_tx.ointerruptwakeup	u_irq_steer.irq_in[1]	HDMI TX Wakeup Interrupt
Reserved	u_irq_steer.irq_in[2]	Reserved
u_blk_ctl.htx_hpd_low2high	u_irq_steer.irq_in[3]	HDMI TX HPD – HDMI Cable Connect Interrupt
u_blk_ctl.htx_hpd_high2low	u_irq_steer.irq_in[4]	HDMI TX HPD – HDMI Cable Disconnect Interrupt
u_blk_ctl.htx_ready_low2high	u_irq_steer.irq_in[5]	HDMI TX PHY powered down Interrupt
u_blk_ctl.htx_ready_high2low	u_irq_steer.irq_in[6]	HDMI TX PHY powered up & ready to transmit Interrupt
u_snps_trng_tx.O_irq	u_irq_steer.irq_in[7]	Ring Oscillator Interrupt
u_lcdif.lcdif_irq	u_irq_steer.irq_in[8]	LCDIF General Interrupt
Reserved	u_irq_steer.irq_in[9]	Reserved
u_vsfd.hrv_mwr_irq_out	u_irq_steer.irq_in[10]	HRV_MWR General Interrupt
u_vsfd.plb_irq_out	u_irq_steer.irq_in[11]	Pixel Line Buffer General Interrupt

Table continues on the next page...

Table 7-5. HDMI Interrupts (continued)

IRQ Source	IRQ_STEER Input	Description
u_video_link_slv.htx_pvi_irq_out	u_irq_steer.irq_in[12]	HDMI TX Pixel Link General Interrupt
u_fdcc.fdcc_irq	u_irq_steer.irq_in[13]	Frequency Dynamic Clock Comp Interrupt
u_pai.htx_pai_irq	u_irq_steer.irq_in[14]	Audio Link General Interrupt

The IRQ_STEER module for the HDMI subsystem has 1 channel, while the IRQ_STEER module for the Audio DSP has 3 channels. Please note this difference when referring to the registers in the IRQ_STEER chapter.

7.1.6 SDMA event mapping

The following table shows the DMA request signals for peripherals in the chip.

NOTE

For the eDMA module, when using the End of Packet feature, the user should set the following CHn_SBR register bits appropriately.

- Bit 22 is for local (means go straight to peripheral) read from peripheral
- Bit 21 is for local (means go straight to peripheral) write to peripheral

Table 7-6. SDMA1 event mapping

SDMA	Module	Description
0	ECSPI1	eCSPI1 Rx request
1	ECSPI1	eCSPI1 Tx request
2	ECSPI2	eCSPI2 Rx request
3	ECSPI2	eCSPI2 Tx request
4	ECSPI3	eCSPI3 Rx request
5	ECSPI3	eCSPI3 Tx request
6	-	Reserved
7	-	Reserved
8	CAN_FD1	CAN-FD1 DMA event
9	CAN_FD2	CAN-FD2 DMA event
10	-	Reserved
11	-	Reserved
12	-	Reserved
13	-	Reserved

Table continues on the next page...

Table 7-6. SDMA1 event mapping (continued)

SDMA	Module	Description
14	IOMUX	External DMA from pad through IOMUX #1
15	IOMUX	External DMA from pad through IOMUX #2
16	I2C5	I2C5 DMA event
17	I2C6	I2C6 DMA event
18	I2C1	I2C1 DMA event
19	I2C2	I2C2 DMA event
20	I2C3	I2C3 DMA event
21	I2C4	I2C4 DMA event
22	UART1	Rx FIFO
23	UART1	Tx FIFO
24	UART2	Rx FIFO
25	UART2	Tx FIFO
26	UART3	Rx FIFO
27	UART3	Tx FIFO
28	UART4	Rx FIFO
29	UART4	Tx FIFO
30	-	Reserved
31	-	Reserved
32	ENET_QOS	ENET_QOS 1588 Event 0
33	ENET_QOS	ENET_QOS 1588 Event 1
34	ENET_QOS	ENET_QOS 1588 Event 2
35	ENET_QOS	ENET_QOS 1588 Event 3
36	Flexspi	Flexspi DMA TX request
37	Flexspi	Flexspi DMA RX request
38	GPT1	GPT1 counter event
39	GPT2	GPT2 counter event
40	GPT3	GPT3 counter event
41	-	Reserved
42	-	Reserved
43	-	Reserved
44	ENET1	ENET1 1588 Event 2
45	ENET1	ENET1 1588 Event 0
46	ENET1	ENET1 1588 Event 3
47	ENET1	ENET1 1588 Event 1

Table 7-7. SDMA2/SDMA3 event mapping

SDMA	Module	Description
0	SAI1	SAI-1 receive DMA request

Table continues on the next page...

Table 7-7. SDMA2/SDMA3 event mapping (continued)

SDMA	Module	Description
1	SAI1	SAI-1 transmit DMA request
2	SAI2	SAI-2 receive DMA request
3	SAI2	SAI-2 transmit DMA request
4	SAI3	SAI-3 receive DMA request
5	SAI3	SAI-3 transmit DMA request
6	-	Reserved
7	-	Reserved
8	SAI5	SAI-5 receive DMA request
9	SAI5	SAI-5 transmit DMA request
10	SAI6	SAI-6 receive DMA request
11	SAI6	SAI-6 transmit DMA request
12	SAI7	SAI-7 receive DMA request
13	SAI7	SAI-7 transmit DMA request
14	IOMUX	external DMA from pad through IOMUX #1
15	IOMUX	external DMA from pad through IOMUX #2
16	ASRC	ASRC Context 0 receive DMA request
17	ASRC	ASRC Context 0 transmit DMA request
18	ASRC	ASRC Context 1 receive DMA request
19	ASRC	ASRC Context 1 transmit DMA request
20	ASRC	ASRC Context 2 receive DMA request
21	ASRC	ASRC Context 2 transmit DMA request
22	ASRC	ASRC Context 3 receive DMA request
23	ASRC	ASRC Context 3 transmit DMA request
24	MICFIL	PDM Digital Microphone Interface DMA request
25	-	Reserved
26	AUD2HTX	AUDIO Link Master To HDMI TX DMA request
27	-	Reserved
28	-	Reserved
29	-	Reserved
30	AUDIO_XCVR	eARC RX DMA request
31	AUDIO_XCVR	eARC TX DMA request
32	-	Reserved
33	-	Reserved
34	-	Reserved
35	-	Reserved
36	-	Reserved
37	-	Reserved
38	GPT4	GPT4 counter event
39	GPT5	GPT5 counter event

Table continues on the next page...

Table 7-7. SDMA2/SDMA3 event mapping (continued)

SDMA	Module	Description
40	GPT6	GPT6 counter event
41	-	Reserved
42	-	Reserved
43	-	Reserved
44	-	Reserved
45	-	Reserved
46	-	Reserved
47	-	Reserved

7.2 Smart Direct Memory Access Controller (SDMA)

7.2.1 Overview

The Smart Direct Memory Access (SDMA) controller offers highly-competitive DMA features combined with software-based virtual-DMA flexibility. It enables data transfers between peripheral I/O devices and internal/external memories.

The SDMA controller helps maximize system performance by off-loading the Arm core in dynamic data routing.

7.2.1.1 Block Diagram

The figure below shows a block diagram of the SDMA controller. It includes the custom RISC core along with its RAM, ROM, DMA units, and the scheduler.

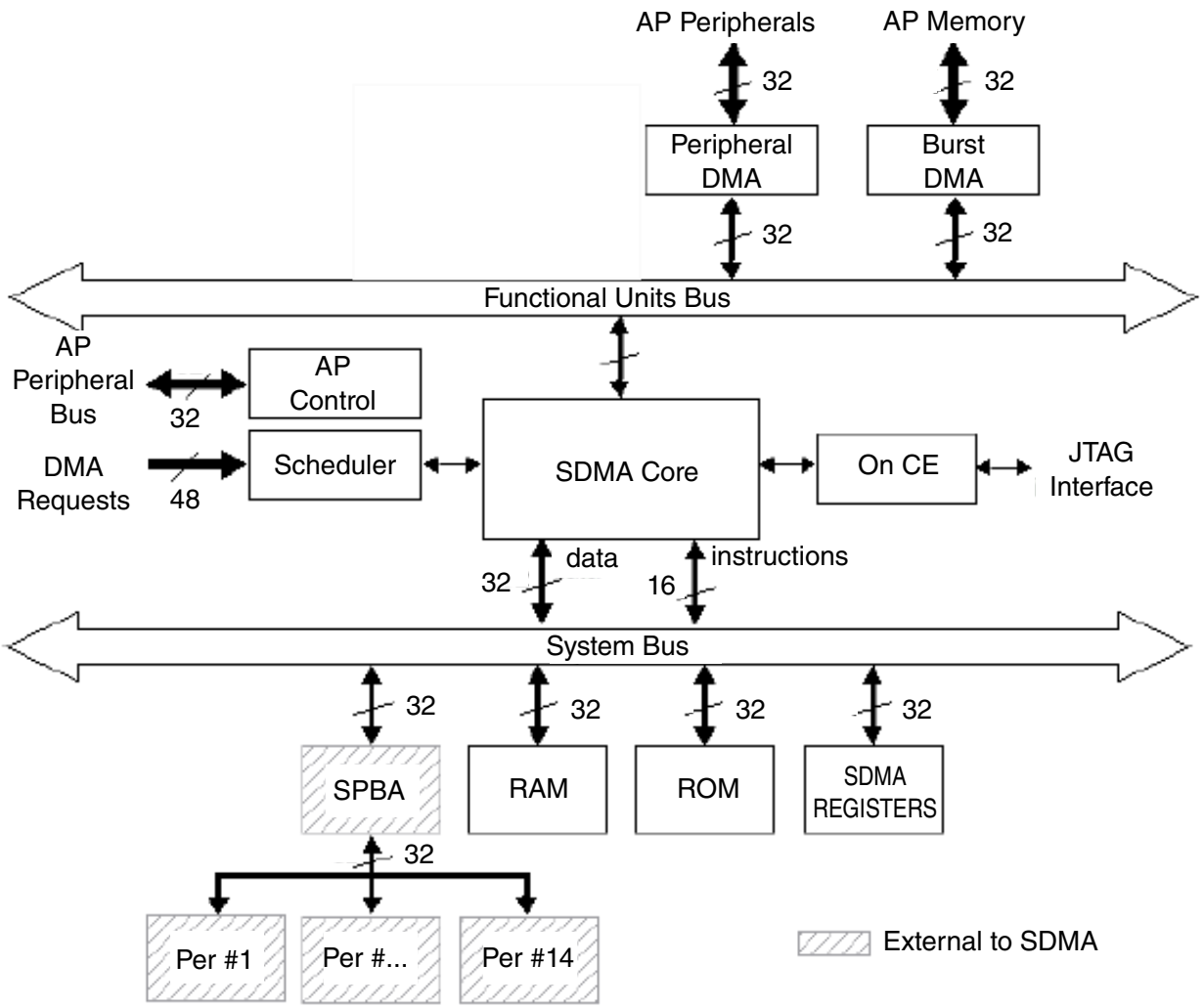


Figure 7-1. SDMA Block Diagram

The SDMA core executes short routines that perform DMA transfers; these routines are called *scripts*. The SDMA core interfaces to its own memory via the SDMA system bus. The SDMA system bus supports a 32-bit data path and a 16-bit address bus. The system bus datapath is used for both 16-bit instruction (program) memory access and 32-bit data access. DMA units interface to the core via the Functional Unit Bus and use dedicated registers to perform DMA transfers.

The SDMA memory contains a ROM and a RAM. The ROM contains startup scripts (for example, boot code) and other common utilities, which are referenced by the scripts that reside in the RAM. The internal RAM is divided into a context area and a script area (more details about this mapping are available in Instruction Memory Map and Data Memory Map).

Every transfer channel requires one context area to keep the contents of all the core and unit registers while inactive. Channel scripts are downloaded into the internal RAM by the SDMA using a dedicated channel that is started during the boot sequence. Downloads are invoked using commands and pointers provided by the Arm platform. Every channel contains a corresponding channel script located in RAM and/or ROM that can be reconfigured independently as-needed. Channel scripts can be stored in an external memory and downloaded when needed. The SDMA can be configured with any mixture of scripts to enable an endless combination of supported services.

The scheduler monitors and detects DMA requests, mapping them to channels, and mapping individual channels to a pre-configured priority. At any given point, the scheduler presents the highest priority channel that requires service to the SDMA core. A special SDMA core instruction is used to "conditionally yield" the current channel being executed to an eligible channel that requires service. If (and only if) there is an eligible channel pending, will the current channel execution be preempted.

There are two yield instructions that differently determine the eligible channels: In the first version, eligible channels are pending channels with a strictly higher priority than the current channel priority. In the second version (yieldge), eligible channels are pending channels with a priority that is greater or equal to the current channel priority. The scheduler detects devices that need service through its 48 DMA request inputs. After a request is detected, the scheduler determines the channel(s) that is (are) triggered by this request and marks it (them) as pending in the "Channel Pending (EP)" register. The priorities of all the pending channels are continuously evaluated in order to update the highest pending priority. The channel pending flag is cleared by the channel script when the transfer has completed.

The Arm platform control block contains the control registers used to configure the 32 individual channels. There are 48 Channel Enable registers, and every register maps one DMA request to any desired combination of channels. The 32 Priority registers are used to assign a programmable 1-of-7 level priority to every possible channel. This block also contains all other control registers that the Arm platform can access.

The 48 DMA requests that are connected to the scheduler come from a variety of sources. The "receive register full" and "transmit register empty" signals found in the UART and USB ports are typical examples of DMA requests that can be connected to the SDMA. These requests can be used to trigger a specific SDMA channel, or several channels.

There is an OnCE compatible debug port for product development. The OnCE includes support for setting breakpoints, single-step and trace, and register dump capability. In addition, all memory locations are accessible from the debug port.

7.2.1.2 Features

The following are the SDMA features:

- Multi-channel DMA supporting up to 32 time-division multiplexed DMA channels
- Hardware or software driven triggers for each channel
- 48 hardware driven triggers that can be mapped to any channel.
- Memory accesses including linear addressing, FIFO addressing and 2D addressing
- Fast context-switching with two-level, priority-based preemptive multi-tasking
- 16-bit instruction-set micro-RISC engine (the SDMA core)
- Two DMA units with some or all the following features:
  - Auto-flush and prefetch capability
  - Flexible address management (increment, decrement, and no address changes on source and destination address)
  - Misaligned data-transfer support
  - Uni-directional and bi-directional flows (copy mode)
  - Up to eight-word buffers for configurable burst transfers
- Support of byte-swapping
- An available API and library of scripts
- Little-Endian and Big-Endian modes
- Hardware handshakes for low-power entry sequence
- Security support to lock contents of the SDMA script RAM.
- 4-Kbyte ROM containing startup scripts (for example, boot code) and other common utilities that can be referenced by RAM-located scripts
- 8-Kbyte RAM area is divided into a processor context area and a code space area used to store channel scripts that are downloaded from the system memory
- Debug support, including a OnCE port, real-time monitors, and embedded cross-trigger events
- Supported clock frequencies in process:
  - Configurable clock options for the SDMA core and the Arm platform DMA units
    - 1:2 ratio with maximum of SDMA core running at Arm platform Peripheral Bus speed and DMA running at max DMA frequency.
    - 1:1 ratio when both SDMA core and Arm platform DMA clocks are set to the Arm platform Peripheral Bus speed.
- Peripheral bus interface for configuration register programming by the Arm platform
- The SDMA RISC engine (arithmetic and logic operations), which is referred to as the "SDMA core."
- An internal peripheral bus connected to the Shared Peripherals Bus Interface (SPBA) that enables access to up to 14 shared peripherals. SDMA supports 32-bit accesses to word peripherals and 16-bit accesses to half-word peripherals.

- The peripheral DMA unit that is hooked-up to the Arm platform Crossbar Switch to service Arm peripherals
- The burst DMA unit is able to perform burst accesses to the external memory
- All the DMA units are 32-bit AHB masters. They are connected to different buses, thus allowing concurrent accesses.

7.2.2 External Signals

The table found here describes the external signals of SDMA.

7.2.3 Functional Description

The figure below shows the SDMA topology, and is composed of the following components:

- SDMA Core (SDMA Core)
- SDMA Scheduler (Scheduler)
- Functional Units:
  - Burst DMA (Burst DMA Unit)
  - Peripheral DMA (Peripheral DMA Unit)
- Arm platform Control for Arm control register access.
- Internal RAM and ROM Memory (SDMA Programming Model)
- OnCE debug Port (The OnCE Controller)

The functional unit bus provides access by the SDMA core to the DMA units. The system bus provides access to SDMA internal memory and also supports up to 14 peripherals.

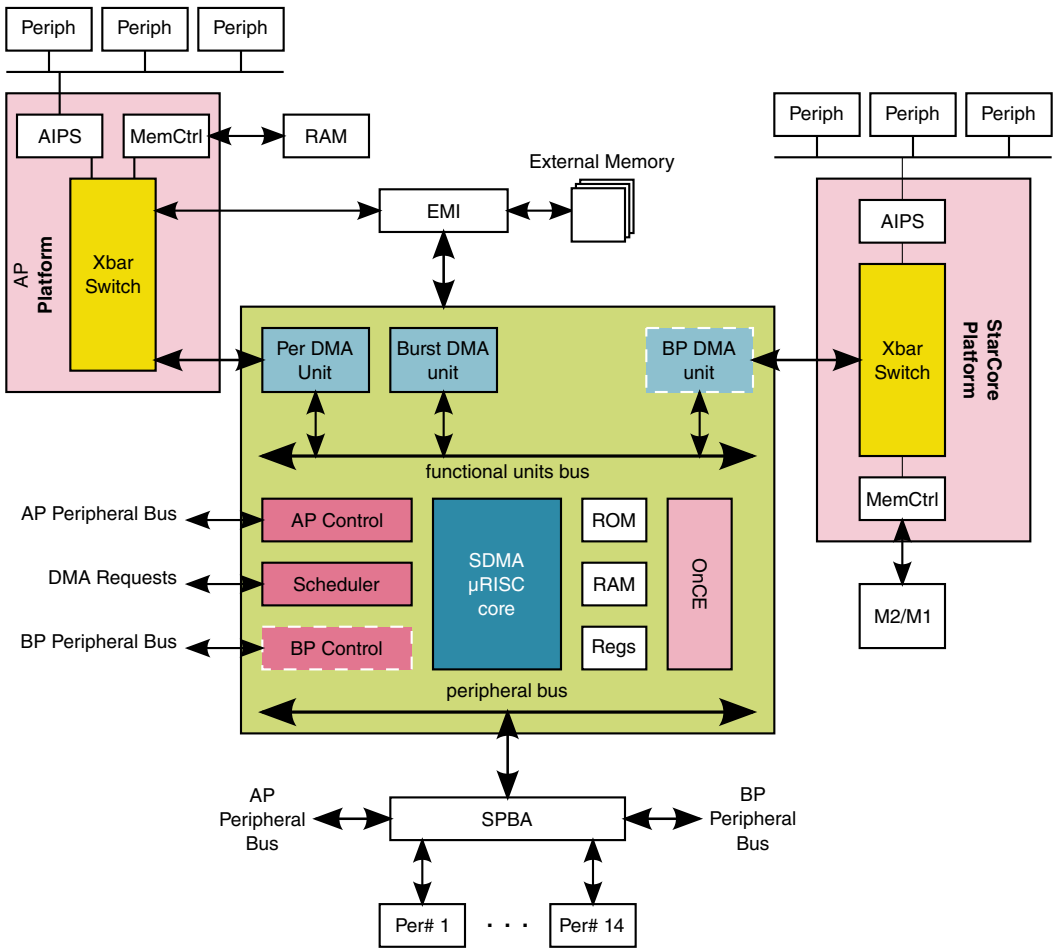


Figure 7-2. SDMA Connections

7.2.3.1 SDMA Core

The SDMA core is a customized RISC-like processor that is specifically developed to control DMA units and perform L1 tasks like byte-stuffing or framing.

The SDMA core incorporates on-chip debug capability using the OnCE.

The SDMA core is based on a 32-bit register architecture with 16-bit instructions. There are eight general purpose 32-bit registers, four flags (T, LM, SF, and DF), and four PCU registers (PC, RPC, SPC, and EPC) that can address 16,384 16-bit instructions.

7.2.3.1.1 SDMA Core Structure

The figure found here shows the structure of the SDMA core. It also shows the different registers, calculation resources, and possible data movements.

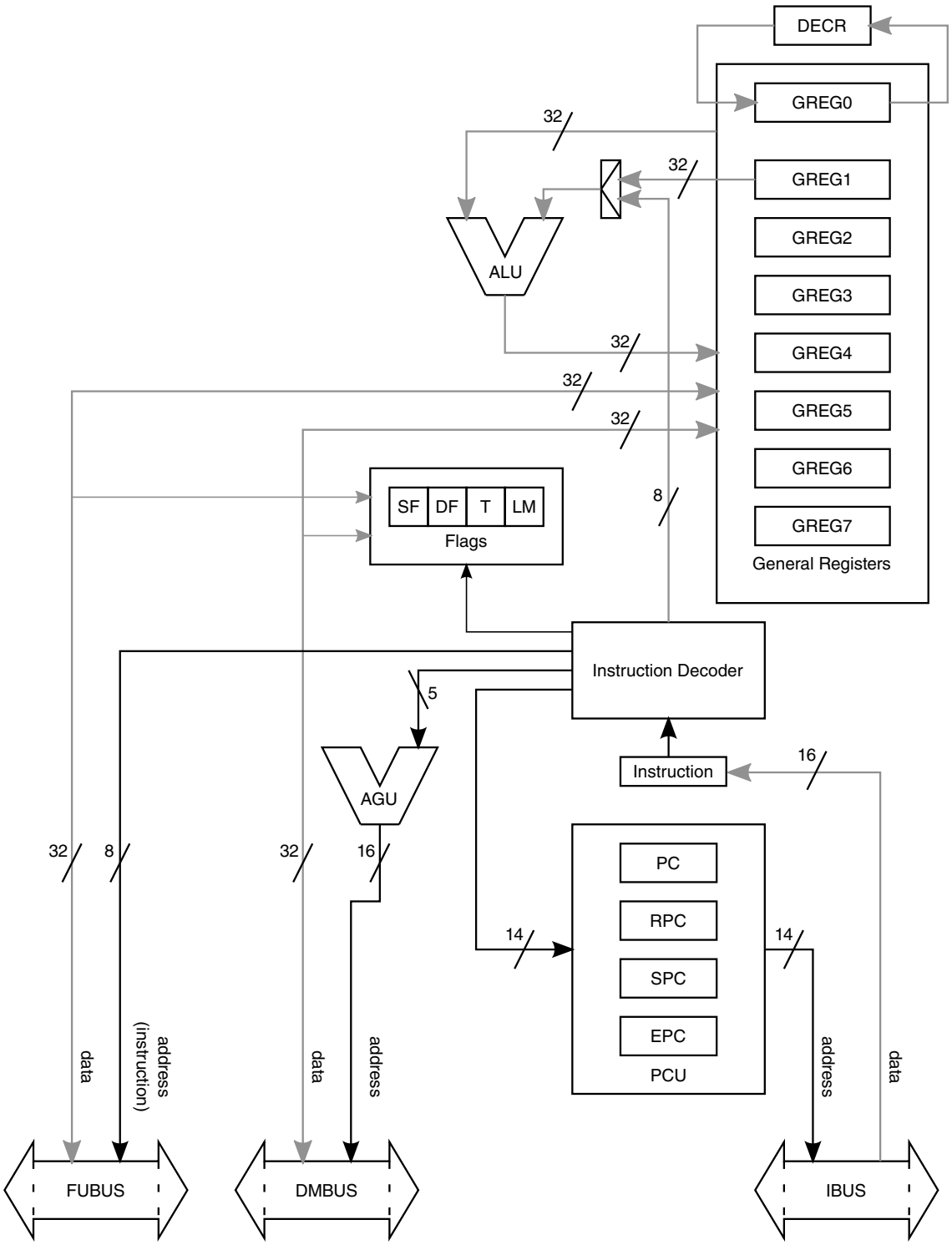


Figure 7-3. SDMA Core

- The Program Control Unit (PCU) is described in Program Control Unit (PCU). It handles the state of the core and generates the instruction fetch addresses. Instructions are retrieved from the Instruction Bus (IBUS) and stored in the SDMA

core instruction register prior to their decoding. The PCU contains the following registers:
    - The Program Counter (PC) contains the address of the current instruction.
    - The Return Program Counter (RPC) contains the address of the instruction that follows a jump to the subroutine.
    - The Start Program Counter (SPC) contains the address of the first instruction of the current hardware loop.
    - End Program Counter (EPC) contains the address of the last instruction of the current hardware loop.
- The other core registers are the general purpose registers (GREGn) and the flags.
    - The general purpose registers can be used to hold data and addresses. They can be loaded with immediate values (for example, 8-bit data that are encoded in the instruction), results of calculations that were performed with the ALU, 32-bit data that comes from the memory or peripherals via the Data Memory Bus (DMBUS), 32-bit data that comes from the DMAs via the Functional Units Bus (FUBUS) or another general purpose register. Their content can be the operands of the ALU, the data to send on either bus (DMBUS or FUBUS), or a pointer to memory (DMBUS address).
    - The general register 0 (GREG0) is also the hardware loop counter. In hardware loops, it cannot be used for any other purpose. This register uses a dedicated decrement unit (DECR) shown in Figure 7-3.
    - The flags reflect the status of operations:
        - SF and DF are set when the last load or store on either bus (FUBUS or DMBUS) received an error response.
        - LM is set when the core is executing instructions inside a hardware loop.
        - T is set when the ALU operation result was 0 or the loop counter reaches 0 (the latter is preponderant when an ALU operation is the last instruction of a hardware loop).
- The ALU has two operands: any general register and either a second general register or an immediate value. The result is always stored into the first general register. A NOP function can be utilized by moving a register's contents into itself (For example, the instruction: mov R0,R0).
- The 16-bit instructions are fetched via the instruction bus (IBUS) whose address is driven by the PC. The SDMA RAM and ROM are visible to the core as 16-bit devices through this interface.
- The memory (RAM and ROM), memory mapped registers, and external peripherals are accessed via the DMBUS. The address is always taken from a general register whose content is added to a 5-bit immediate value. This is the only available addressing mode. The DMBUS is a 32-bit data bus. Except for the peripherals that

are external to the SDMA, the address accuracy is the 32-bit word (for example, adding 1 to an address points to the next word, not the next byte).
- The functional units are accessed via the FUBUS connection. The data is exchanged with any general register, but the address (which in fact is the instruction and the selector of the functional unit) comes from an 8-bit field of the corresponding load or store.

7.2.3.1.2 Program Control Unit (PCU)

This part of the SDMA core is dedicated to the control of the RISC engine, as implied by the instructions that are executed. Its behavior is determined by the instruction type and the inputs of the SDMA.

It contains the PC, RPC, SPC, and EPC registers that are described in SDMA Core Structure.

7.2.3.1.2.1 Instruction Types

The state sequence and the delay of execution vary according to the type of the instruction. There are six possible categories of instructions, as follows:

1. Standard: Most of the instructions belong to this category, and always last 1 cycle.
2. ldf/stf: These are respectively the load and store instructions that access the functional units. They last 1+n cycles where n is the number of wait-states of the targeted functional unit.
3. ld/st: These are the load and store instructions that access the memory and peripherals. They last 1+n cycles where n is the number of wait-states of the targeted device (1 for the ROM, RAM, and memory mapped registers, 1 + the external peripheral wait-states). These instructions always last at least two cycles, but the core is able to handle them in one cycle. The first wait-state is inserted outside the core.
4. Branch: These are all the instructions that cause the Program Counter to point to another instruction other than the following one (for example, one that breaks the sequential flow). There are the absolute jumps, the conditional branches, the jump to the sub-routines, and the return from the sub-routine.
5. Loop,Modified Load or Store: The hardware loop instruction modifies the potential behavior of any load or store inside the loop (for example, when the LM flag is set). A jump may be implied after any such load or store if it received an error. The error causes an early exit of the loop, which means a jump to the instruction that follows the one that is pointed to by EPC. An additional cycle is required by the PCU to perform the jump (+1 to the ld/st/ldf/stf original execution delay). Although there is usually an implicit jump after the last instruction of the loop when the PC goes back to SPC, this is performed at no cycle cost.

6. Done: The done, yield, or yieldge instructions are used to control channel switching. When no channel switching is performed, these instructions last a single cycle. When there is a change of channel or context switch, the delay is variable and depends on many factors (as detailed in Context Switching).

7.2.3.1.2.2 PCU States

The PCU state is visible through outputs of the SDMA (see Real-Time Debug Outputs) or the OnCE status register(see OnCE Status Register (OSTAT)).

The PCU state is a four-bit field that can take the values shown in the following table. Figure 7-4 shows the possible state transitions and the corresponding conditions.

Table 7-8. PCU States

Value	State	Description
0	Program	This is the usual instruction cycle.
1	Data	This state is inserted when there are wait-states during a load or a store on the data bus (ld/st type).
2	Change of Flow	This is the second cycle of any instruction that breaks the sequence of instructions (branch and done types). This state lasts only a single cycle; it is always followed by the Program state.
3	Error in Loop	This state is used when an error causes a hardware loop exit (loop-modified load or store type). This state only lasts a single cycle; it is always followed by the Program state.
4	Debug	The SDMA is stopped in debug mode.
5	Functional Unit	This state is inserted when there are wait-states during a load or a store on the functional units bus (ldf/stf type).
6	Sleep	No script is running: The core is idle after saving the last channel context.
7	Save	The context switch FSM is saving the current channel.
8	Program in Sleep	Same as Program except there is no associated channel, this state is used when instructions are executed after entering debug mode, whereas the core was in either Sleep mode.
9	Data in Sleep	This is the same as Data except there is no associated channel.
10	Change of Flow in Sleep	This is the same as Change of Flow except there is no associated channel. This state only lasts a single cycle, and is always followed by the Program in Sleep state.
11	Error in Loop in Sleep	This is the same as Error in Loop except there is no associated. channel. This state only lasts a single cycle, and is always followed by the Program in Sleep state.
12	Debug in Sleep	This is the same as Debug except the core was put in debug mode when no channel was active.
13	Functional Unit in Sleep	This is the same as Functional Unit except there is no associated channel.
14	Sleep after Reset	This shows that no script is running, and the core is idle after a reset. When a channel becomes active, no context is restored but the core starts its boot program located at address 0 (or the address available in register in Channel 0 Boot Address (SDMAARM_CHN0ADDR)).
15	Restore	The context switch FSM is restoring the next channel context.

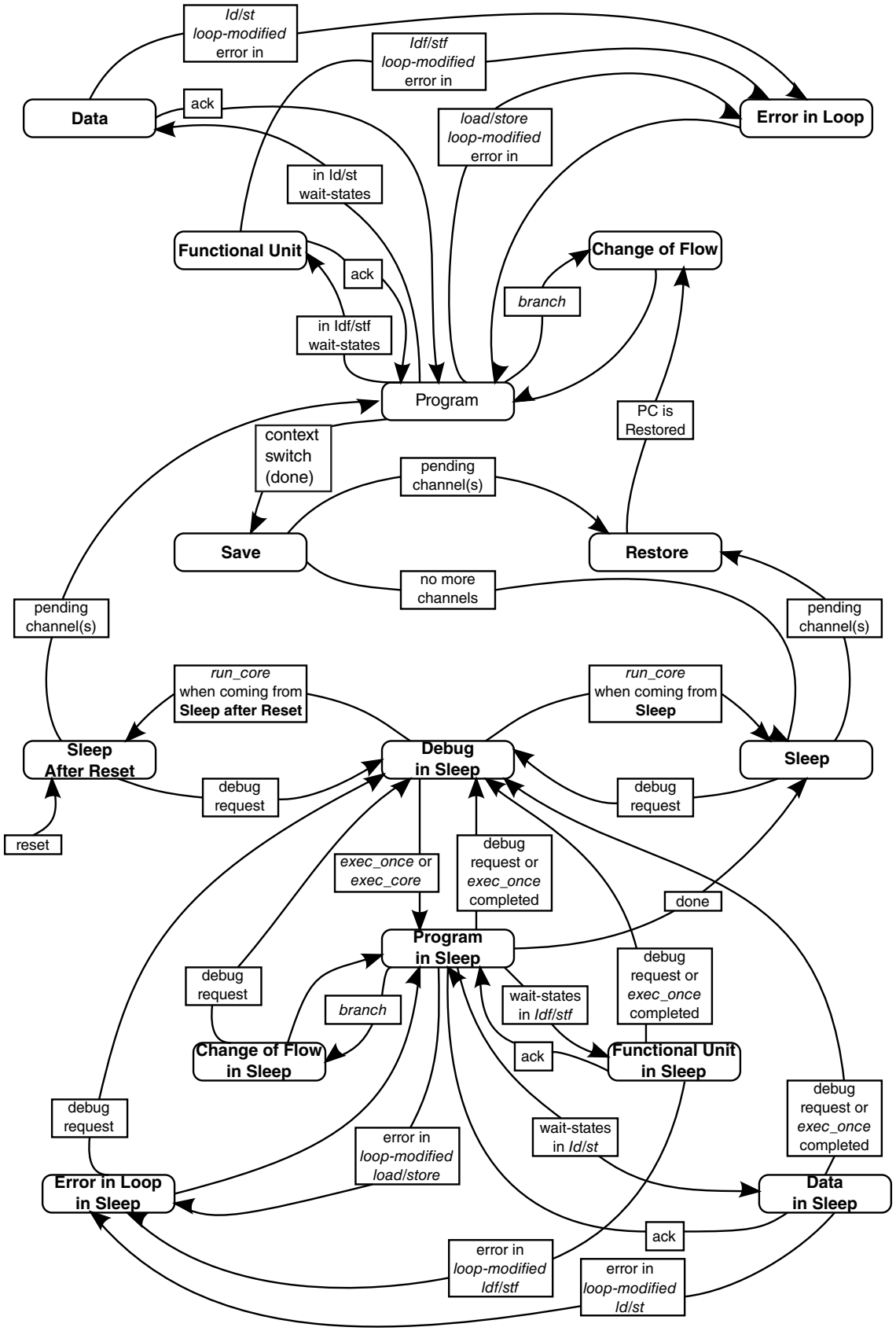


Figure 7-4. PCU State Diagram

7.2.3.1.3 SDMA Core Memory

The SDMA has two memory spaces: one for the instructions and one for the data. As both spaces share the same resources (ROM and RAM devices), the system bus manages possible conflicts when the core accesses the same resource for both an instruction read and a data read or write.

Program and data memory is further described in Address Space.

Instructions of 16-bit width are stored in 32-bit wide devices and can be accessed as data. The mapping is Big Endian: an even instruction address (terminated by 0) accesses the most significant part of the 32-bit data (bits [31:16]), and an odd instruction address (terminated by 1) accesses the least significant part of the 32-bit data (bits [15:0]). Instructions can be fetched out of internal ROM or RAM.

Data can be read from ROM, RAM, memory mapped registers, and external peripherals, and written to the same devices (except the ROM).

The ROM contains bootload scripts, channel scripts, and common subroutines which may be referenced by channel scripts elsewhere in the ROM or RAM.

The RAM is divided into a context area and a code space area which may be used to store channel scripts. The RAM contains undefined values after a hardware reset. Channel scripts and initial context values are downloaded into RAM using channel 0 which is reserved for bootload functions.

7.2.3.2 Scheduler

All channel scheduling hardware is included in the Scheduler.

7.2.3.2.1 Primary Functions

The scheduler is a hardware-based design used to coordinate the timely execution of 32 virtual DMA channels by the SDMA core on the basis of channel status and priority.

The scheduler performs the following functions:

- Monitors, detects, and registers the occurrence of any one of the 48 DMA requests
- Links a specific request to a channel or group of channels (channel mapping)
- Ignores requests that are not mapped to a previously configured channel
- Maintains a list of all the channels that are requesting service
- Assigns a pre-programmed priority level (1 of 7) to every channel requesting service
- Detects and flags overrun/underrun conditions

7.2.3.2.2 Channels and DMA Requests

7.2.3.2.2.1 Channels

A Virtual Channel (hereafter simply called a channel) manages a flow of data through the SDMA. Flows are typically unidirectional.

The SDMA can have up to 32 simultaneously operating channels, numbered from 0 to 31. Channel 0 is usually dedicated to control the SDMA script downloading. All the channels can be assigned by the Arm platform software.

7.2.3.2.2.2 DMA Requests

A DMA request is caused by externally (for example, external to the SDMA) controlled conditions (for example, UART receive FIFO reaches a threshold). The SDMA currently supports up to 48 DMA requests.

7.2.3.2.2.3 Mapping from DMA Requests to Channels and Priorities

A channel can stall waiting on a single DMA request. A single DMA request can awake more than one channel (in fact, any request can awake any combination of channels).

The mapping between DMA requests and channels is program-controlled. There is a storage element assigned for each of the 48 requests that contains a bitmap table of the channels that are awakened by the event.

Every channel also has a three-bit register that indicates its priority.

7.2.3.2.3 Scheduler Functional Description

Scheduler Overview describes the behavior of the SDMA scheduler-from the channel enabling conditions to the highest priority pending channel selection.

7.2.3.2.3.1 Scheduler Overview

The scheduler algorithm is built in hardware. It is provided with possibilities for the Arm platform to control its behavior.

The scheduler processes incoming DMA requests, maps detected requests to 0, one, or several channels, maintains a list of channels that are requesting service (pending channels), identifies the top priority and its associated channel, and selects the next active channel when the current channel yields.

The following figure shows a functional overview.

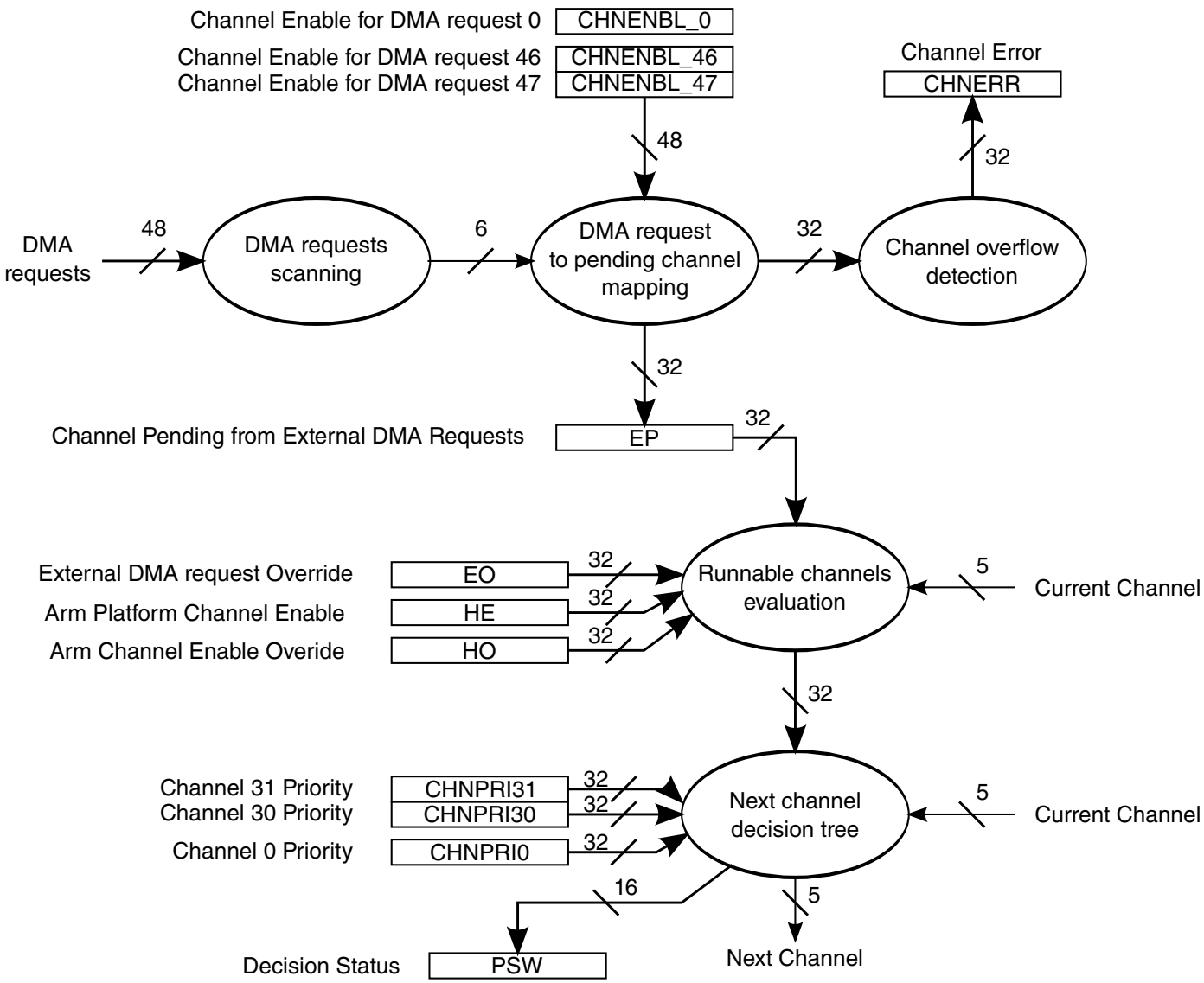


Figure 7-5. SDMA Hardware Scheduler

7.2.3.2.3.2 DMA Requests Scanning

The scheduler contains a 48-bit edge detection device that detects the rising edge of every DMA request and transmits the request number to the next stage.

The DMA requests are assumed to be generated on the same reference clock as the SDMA core clock; they are detected as soon as the signal goes from a 1-to-n-cycles low state to a 1-to-m-cycles high state.

This system is able to detect single-cycle pulses as well as level-based DMA requests such as a FIFO threshold crossing. In this case, the SDMA provides a memory mapped register that can be used by the channel script to monitor the DMA requests lines, and thus determines whether the data transfer is done or not done, and then continues with the transfer or closes the channel.

When several DMA requests are detected at the same time, they are forwarded to the next scheduler stage at the rate of one request per cycle. No request is lost.

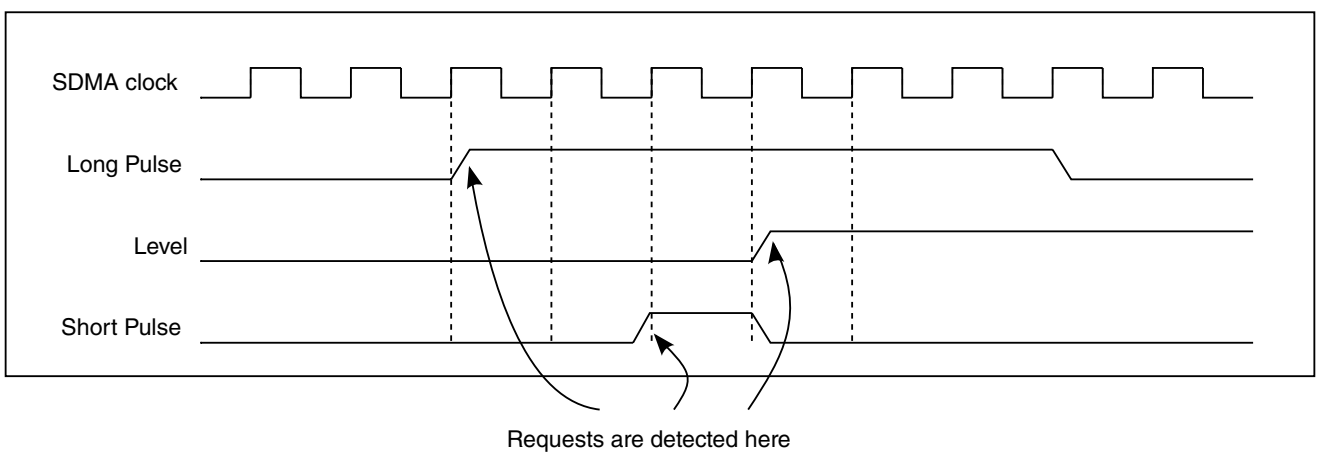


Figure 7-6. Examples of Valid DMA Requests

The DMA request inputs are connected to various sources that depend on the SoC. The exact list of DMA request inputs and their associated number is available in each respective project-specific chapter.

7.2.3.2.3.3 Mapping DMA Requests to Pending Channels

Whenever a DMA request is detected by the first stage, its number is used in the second stage to determine the channels that have to be activated.

This is performed with an array of 48 registers that are 32 bits wide: There are 48 Channel Enable Registers (CHNENBLn), one register per DMA request. The DMA request number selects the Channel Enable Registers, and every bit of this 32-bit register indicates that the corresponding channel must be activated when it is a 1.

This information is passed on the EP register. For every bit of the Channel Enable Register that is set, the corresponding bit of the EP register is also set, and the remaining bits of EP are left unchanged. The transformation of EP is summarized by the following equation:

EP = EP or CHNENBLn

The EP register is used to know which channels require service because they received a DMA request.

Typical contents of the CHNENBLn registers are all 0s, except for a single bit set. For example, a DMA request triggers one channel, but all 0s or several 1s are possible. One DMA request could activate several channels, and the channel execution sequence can be controlled by the channel priorities and numbers, as explained in the next sections. The following table illustrates an example configuration.

NOTE

From the table, the DMA request 0 is programmed to simultaneously trigger channels 0, 1, and 31. Also, DMA requests 30-47 are not used in this example. The remaining channels 2 to 30, are configured to be triggered by DMA requests 29 to 1, respectively.

Table 7-9. Channel Enable RAM Programming Example

DMA Request Number	Channel																															
	31																															0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table continues on the next page...

Table 7-9. Channel Enable RAM Programming Example (continued)

DMA Request Number	Channel																															
	31																															0
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table continues on the next page...

Table 7-9. Channel Enable RAM Programming Example (continued)

DMA Request Number	Channel																															
	31																															0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.2.3.2.3.4 Channel Overflow

A channel overflow occurs when a DMA request requires service from channel *n* by setting bit *n* of the register EP, but this bit is already set, meaning channel *n* is already pending. This can come from an overrun/underrun condition.

This detection is possible only when the DMA requests are pulses, because a level-based DMA request stays high until it is serviced, even though an underrun or overrun condition occurs, thus preventing another edge detection of the DMA request.

The channel overflow information is saved in the 32-bit CHNERR register (1 bit per channel). You can configure the SDMA to trigger an interrupt to the Arm platform when there are 1s in CHNERR. Every bit of CHNERR is masked with the corresponding bit of INTRMASK and if it gives a 1, the corresponding bit of INTR is set, triggering the interrupt.

7.2.3.2.3.5 Runnable Channels Evaluation

The EP register is used in conjunction with several other 32-bit registers to determine the channels that are runnable.

Registers EO, DO, HO and HE, are controlled by the Arm platform. EP is controlled by the DMA requests and their mapping to channels.

Several channels may be runnable at any given time. The i^{th} channel is runnable if (and only if) the condition below is true:

(HE[i] or HO[i]) and (DO[i]) and (EP[i] or EO[i])

After reset, the HE[i], HO[i], EP[i], and EO[i] bits are all cleared whereas the DO[i] bits are all set. The functions associated with DO are not available for this device. When DO[i] is set, the scheduler condition becomes:

(HE[i] or HO[i]) and (EP[i] or EO[i])

The registers in these equations are controlled as follows:

- Arm platform (host) channel enable flag HE[i] may be set or cleared by the Arm platform with the HSTART and STOP_STAT registers. It can also be cleared by the i^{th} channel script.

  Typical usage is for the Arm platform to set this flag to activate the channel. The flag is cleared by the SDMA core when the transfer is done.

- Externally triggered channel pending flag EP[i] is set by the scheduler when the channel was activated by a DMA request. It can be cleared by the i^{th} channel script.
- The Arm platform channel override flag HO[i] may be set or cleared by the Arm platform. When set, it enables the i^{th} channel to run without the involvement of the Arm platform.

  Typical usage is for the Arm platform to set this flag for channels that do not need Arm platform supervision such as channels that are controlled by DMA request events (EP).

- DO should always be set to 1 so that the runnable channel evaluation considers only HO, HE, EP, and EO.
- Externally triggered channel override flag EO[i] may be set or cleared by the Arm platform. When set, it prevents the i^{th} channel from stopping and stalling on incoming peripheral DMA requests. This is the case when the channel is not handling data transfers with peripherals (for example, a memory to memory transfer).

The SDMA can clear the HE[i], and EP[i] bits by means of a done or notify instruction. The done instruction causes a reschedule; thus, enabling another channel to preempt the current one, while the notify instruction does not. The done and notify instructions can clear either HE[i] or EP[i] (never more than one at a time).

Table 7-10. Runnable Channel Selection Control

Register	Set by	Cleared By
HO	Write to HOSTOVR register	Write to HOSTOVR register
HE	Write to HSTART register	Write to STOP_STAT register or by the channel script with the done or notify instructions.
DO	Write to DSPOVR register	Write to DSPOVR register
EO	Write to EVTOVER register	Write to EVTOVER register
EP	Set by external DMA request event input.	By the channel script with the done or notify instructions

7.2.3.2.3.6 Next Channel Decision Tree

The next channel number is computed from the runnable channels list, the current channel number, and their respective priorities.

It is re-evaluated every cycle, but is only used when the current channel yields or terminates by executing a yield, yieldge, or done instruction.

The decision tree is based on the selection of the runnable channel that has the highest priority.

The highest priority channel is selected according to the following rules:

- Runnable channels are sorted by priority.
- If one of the channels with the highest priority had been preempted by a channel with a higher priority, but did not want to yield to a channel of the same priority (for example, it executed a yield, not a yieldge), it is elected as the next channel.
- The channels that belong to the highest priority group are sorted by their number and the channel that has the highest number in this group becomes the next channel. For example, if priorities are the same, channel 31 will be selected before channel 30.

When the current channel requires a reschedule with a yield(ge) or a done instruction, the context switch decision is based on the instruction parameter, the current channel number and priority, and the next channel number and priority. The possible cases are all listed in the following table. The grayed cells correspond to unusual cases that should not occur with a typical usage of the SDMA.

Table 7-11. Channel Switching Decision with a yield, yield(ge), or done

Instruction	Current Channel	Next Channel	Priorities Comparison	New Running Channel/Comments
yield (done 0)	Runnable	Not runnable	none	Current
	Runnable	Runnable	Current > Next	Current
			Current = Next	Current

Table continues on the next page...

Table 7-11. Channel Switching Decision with a yield, yield(ge), or done (continued)

Instruction	Current Channel	Next Channel	Priorities Comparison	New Running Channel/Comments
			Current < Next	Next, [1]
	Not runnable	Not runnable	none	none, [2] (occurs when the channel was disabled by the Arm platform)
	Not runnable	Runnable	none	Next[1] (occurs when the channel was disabled by the Arm platform)
yieldge (done 1)	Runnable	Not runnable	none	Current
	Runnable	Runnable	Current > Next	Current
			Current = Next	Next[1]
			Current < Next	Next[1]
	Not runnable	Not runnable	none	none[2] (occurs when the channel was disabled by the Arm platform)
	Not runnable	Runnable	none	Next[1] (occurs when the channel was disabled by the Arm platform)
done (done>1)	Not runnable	Not runnable	none	none[2]
	Runnable	Not runnable	none	Current[3] (occurs when the done instruction does not disable the channel runnable condition)
	Not runnable	Runnable	none	Next[1]
	Runnable	Runnable	none	Current[3] (occurs when the done instruction does not disable the channel runnable condition)

1. Current channel script execution is stopped, its context is saved; the next channel context is restored and its script execution resumes
2. Current channel context is saved and SDMA enters IDLE mode
3. Current channel context is saved, then restored, and the current channel script resumes execution

Finally, when the SDMA is in IDLE mode and a runnable channel is elected as the next channel, its context is immediately restored and the script execution resumes.

The *combinatorial-decision* tree supports dynamic modifications of the EP, EO, HE, HO, and DO flags as well as dynamic modifications of the channel priorities. The propagation times are detailed in Scheduler Pipeline Timing Diagram.

The decision tree status is available in the PSW register, which is continuously updated. It contains the next channel priority, the next channel number, the current channel priority, and the current channel number. When a priority is read as 0, it means the channel is not runnable.

A few examples of decisions are presented below:

- Channel 31 is running with priority 5, channels 13 and 24 are pending with the same priority 5; channel 24 is eligible as the next channel since 24 > 13.
- Channel 31 is running with priority 7, channels 13 and 24 are pending with priority 5; channel 31 is the next channel because its priority is greater than the other pending channels.
- Channels 7, 23, and 29 are pending with the same priority. Channel 7 is active and runs a yieldge; it is preempted by channel 29. After a period of time, channel 29 runs a yieldge, it is then preempted by channel 23 that is the selected channel since channel 29 is the current channel. Later, channel 23 runs a yieldge and is preempted by channel 29. Channels 23 and 29 will go on switching after every yieldge until one of them terminates. It is only at that point that channel 7 becomes eligible again.
- Channel 11 is running with priority 3, and channel 15 is pending with priority 4. When the channel 31 script executes a yield instruction, it gets preempted by channel 15; then channels 6 and 18 with priority 3 become pending. Because channel 11 was preempted after executing a yield and there is no pending channel with a strictly greater priority, it is eligible as the next channel (although its number 11 < 18).

7.2.3.2.3.7 Scheduler State Diagram

The Figure 7-7 summarizes the behavior of the SDMA scheduler with details about the exact mechanism of the priority decision tree. It is important to understand the scheduler is a hardwired pipeline, which means all the stages are performed simultaneously every cycle, but a change on any given stage is reflected on the next stage after the delays presented in Scheduler Pipeline Timing Diagram.

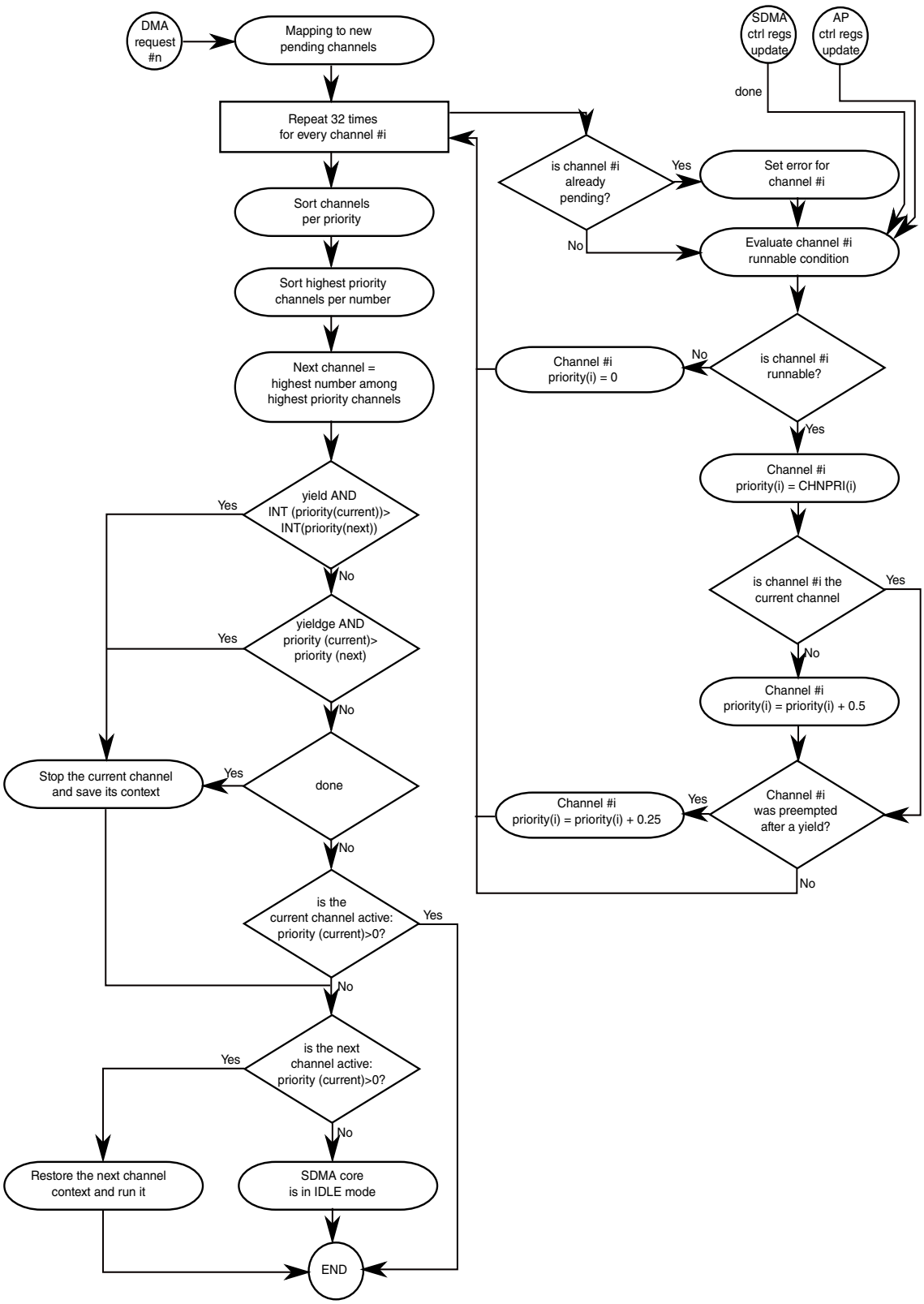


Figure 7-7. Scheduler State Diagram

7.2.3.2.3.8 Scheduler Pipeline Timing Diagram

The SDMA scheduler process of DMA-request and control-register modifications is not immediate.

The figure below shows the exact delays of all the tasks. The reference clock is the SDMA core clock.

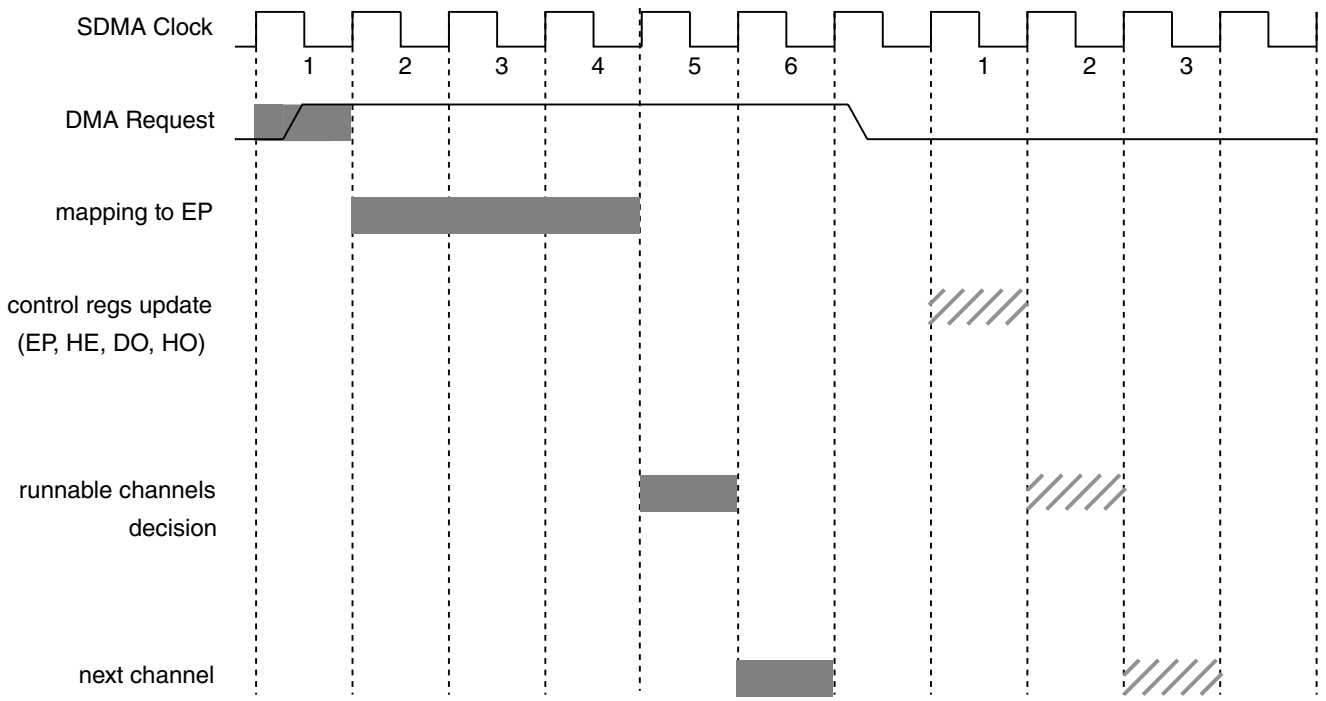


Figure 7-8. Scheduler Timing Diagram

Two numbers can be inferred from this timing diagram. First, it takes six SDMA core clock cycles to update the next channel from a DMA request. Second, it takes three SDMA core clock cycles to update the next channel from a direct modification of the condition registers (EP, DO, HE, or HO) by any processor. The processors that can modify these bits include SDMA with a done instruction or the Arm platform with a write access through the corresponding control port on their respective peripheral bus).

7.2.3.2.3.9 Channel-DMA Request Mapping

The 48 DMA request inputs to the SDMA scheduler are listed in project-specific chapters. Refer to the respective chapters for this information.

7.2.3.2.3.10 Examples: How to Start a Channel

A channel can be started when the following equation is true for channel *i*:

(HE[i] or HO[i]) and (DO[i]) and (EP[i] or EO[i])

Once this equation is true, the scheduler can start this channel according to the priority of all pending channels. Several examples of configuration are listed below:

1. To start a channel triggered by Arm platform software:
   - Initially, configure HO[i]=0, DO[i]=1, and EO[i]=1 using registers indicated in Table 7-10.
   - Arm platform software triggers the channel by writing to the HSTART register to set HE[i]=1, thereby setting the above equation true.
2. To start a channel triggered by DMA request event.
   - Initially, configure HO[i]=1, DO[i]=1, and EO[i]=0 using registers indicated in Table 7-10.
   - The DMA request is asserted to trigger the channel by setting EP[i]=1, which makes the above equation true.

7.2.3.2.4 Context Switching

On execution of a done or yield(ge) instruction, the current channel may be changed either because it has finished (which necessarily happens when the done instruction is executed), or it was preempted by a higher priority channel (which is possible but not systematic when the yield(ge) is executed).

Upon a channel change the SDMA goes through a context switch procedure.

When the current channel yields or ends, the context for that channel is saved into the context RAM locations for that channel. When the next channel starts running, its context is first restored from RAM.

Since context RAM is not yet initialized by reset, there will be no context restore at the beginning of the first channel (bootload channel) run after reset. It is expected that the bootload channel will be used to initialize the context for all other channels. When the bootload channel finishes running or yields, SDMA will enter its SAVE state and save that channel's context into RAM. Then, if the bootload channel is called again later, the context will be restored from RAM when the channel starts again.

The context structure for each channel is defined in Context Switching-Programming and Table 7-16. There will be one context area reserved for each channel. When a channel ends or yields, the SDMA core registers are automatically saved into the context RAM and later restored from the context RAM when the channel is next run. The total RAM

space reserved for 32-channel contexts is either 3K or 4K depending on whether the SMSZ bit is set in the CHN0ADDR register, which enables an additional 8 words of scratch RAM for each context.

7.2.3.2.4.1 Context Switch Modes

The exact procedure to save the context of the old channel, and to restore the context of the new channel depends on the context switch mode selected by the Arm platform in the CONFIG control register.

The following are the context switch modes:

- By default, the "dynamic" context switch is set. This mode provides the most efficient context switch for an average of eight cycles to stop the current channel, save its context, restore the next channel context, and resume its execution. It consists of saving modified registers of the current channel in the background (for example, during the channel execution)-which leaves very few registers to save when the switch is decided-resuming execution of the next channel as soon as possible (for example, when the minimal set of registers is restored), and continuing the restore phase during this execution.
- In "dynamic with no loop" mode, the same principle is followed except the modified registers are only saved in the background when the loop flag is not set. This mode offers almost the same effectiveness as the previous one, but it prevents the system from accessing the RAM during loops to save power. This is the recommended mode for an efficient context-switch when the loop bodies are short.
- In "dynamic power" mode, no background saving is performed, which reduces power consumption to the minimum. The modified registers are only saved when the context switch starts. The restore phase is the same as before. This is the mode that achieves the optimal power consumption at the cost of a slower context-switch.
- In a "static" context switch, all the registers are saved when a context switch is decided, and all the registers are restored before starting the execution of the new channel. This mode enables a predictable behavior of the context switch since all the registers are restored prior to the channel start and all registers are saved after the channel termination.

NOTE

Static context mode should be used for the first channel called after reset to ensure that the all context RAM for that channel is initialized during the context SAVE phase when the channel is done or yields. Subsequent calls to the same channel or different channels may use any of the dynamic context modes. This will ensure that all context locations for the bootload

channel are initialized, and prevent undefined values in context RAM from being loaded during the context restore if the channel is re-started later.

7.2.3.2.4.2 Context Switch Procedure

The Program Control Unit goes into the *save* state, the current context is spilled into memory, and the next channel context is restored according to the context-switch mode that was selected by the Arm platform.

The context switch procedure is as follows:

1. Load the current context's spill base address.
2. Spill the modified registers of the current channel to memory according to the selected context switch mode while the channel is running.

   On a done or yield(ge) that causes the channel preemption, the PCU goes into the *save* state. In *static* mode, all the registers are saved; whereas, in either *dynamic* mode, the registers that were modified but not yet saved are then saved, and the PCU registers and flags are finally saved.

3. Put the SDMA core into *sleep* and wait for new channels to be serviced. This step is skipped if there are pending channels when the current channel is saved.

   As soon as there is at least one pending channel, the PCU goes into its *restore* state to restore the context of the channel that was elected by the scheduler.

   Once a channel is elected, it remains the current channel until its script requests a rescheduling operation with a done or yield(ge) instruction. That means the current channel cannot be modified by the Arm platform, even if it is no more runnable or if its priority is modified.

   The Arm platform can however force a reschedule by writing the corresponding bit in the CONFIG register, which has the same effect as if the script had executed a done instruction. That feature should only be used to stop the SDMA in emergency cases.

4. Load the context base-address of the new channel.

   In "static" mode, all the registers are restored. In either "dynamic" modes, only the PCU registers are restored.

   The new channel is running. In "static" mode, no more activity regarding context restoring or saving is performed. In either "dynamic" modes, the registers are restored in the background every time an access to the context RAM is possible, and

priority is given to restoring the registers that are required by the next instruction to be executed. When a register has not been restored and the next instruction needs it, this instruction gets stalled until the register was restored.

In "dynamic" and "dynamic with no loop" modes, background saving of dirty registers is performed every time an access to the context RAM is possible and allowed by the context switch mode.

NOTE

The contents of a channel context space in the context RAM depends on the selected context switch mode. In "dynamic" and "dynamic with no loop" modes, the contents of the context RAM tend to match the contents of the SDMA registers (except for the PCU registers and flags that are never saved in the background). In "dynamic power" and "static" modes, the contents of the context RAM remain unchanged until the channel terminates with a done or gets preempted.

7.2.3.2.4.3 Context Map in Memory

Refer to Context Switching-Programming.

7.2.3.3 Functional Units

The functional units are small systems that are used by the SDMA core to handle data transfers between the core and a bus domain external to the SDMA.

The SDMA core is able to control and exchange data with these systems by sending instructions and reading or writing data from/to the functional units' registers via the FUBUS. This is done with the ldf and stf instructions.

The following sections provide introductions to the available functional units. Functional Units Programming Model provides descriptions the functional units' behaviors.

7.2.3.3.1 Burst DMA Unit

The burst DMA unit enables the SDMA core to perform data transfers to and from the Arm platform memory.

It is optimized for accessing SDRAM-like devices. It does not provide control to assign a privilege level to the DMA access. The burst DMA unit provides the SDMA with means to do the following:

- Perform up to 8-beat read and write bursts to the Arm platform memory, which optimizes throughput when accessing SDRAM-type devices because of an internal, 36-byte FIFO
- Access the Arm platform memory at once or twice the SDMA core frequency
- Copy data from one Arm platform memory location to another Arm platform memory location at the Arm platform bus speed, which provides a very high throughput
- Control the method for addressing the Arm platform memory (automatic increment of addresses or frozen addresses-the former aimed at accessing RAM-like memory and the latter aimed at accessing single-address FIFOs)
- Enable or disable automatic prefetch when reading data from the Arm platform memory. When the prefetch mode is selected, the burst DMA automatically triggers external bursts to fill its FIFO without waiting for the SDMA core to request the corresponding data, greatly improving throughput.
- Rely on the DMA to automatically flush its FIFO content when there is enough data to generate an 8-beat burst to the Arm platform memory. Or, it forces a flush when a data transfer must terminate.
- In the former case, the SDMA core may only be stalled when it tries writing data and there is not enough room left in the FIFO. In the latter case, the core is stalled until the data is effectively written to the Arm platform memory.

  In automatic flush mode, the core receives an acknowledge that does not reflect the actual error status when the data is effectively written into the Arm platform memory. This error status is retrieved by a later access to the burst DMA. Terminating a write data transfer with a forced flush command guarantees that any bus error to the Arm platform memory is caught.

- Handle address alignment issues between the Arm platform memory map and the SDMA core data. This enables the core to read or write 32-bit data from the burst DMA, whereas the corresponding Arm platform address is not 32-bit aligned. This drastically improves the SDMA scripts' efficiency since the same loop that transfers 32 bits at a time can be used regardless of the start and end addresses in the Arm platform memory space.

This unit structure and registers are described in Burst DMA Structure and Burst DMA Registers.

7.2.3.3.1.1 Burst DMA Structure

The burst DMA is essentially made up of a 36-byte FIFO, address registers, and a controlling state-machine. The 36-byte FIFO enables eight-word buffering with address alignment, and the state-machine manages clock adaptation when required.

The burst DMA is depicted in the figure below.

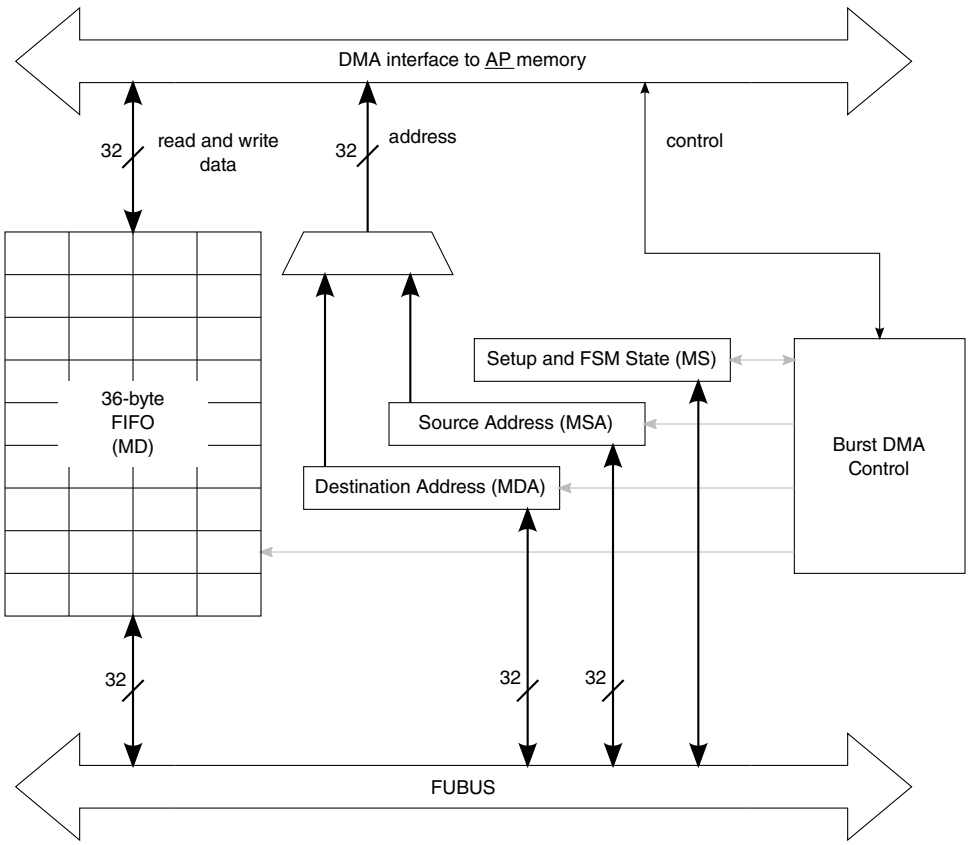


Figure 7-9. Burst DMA Structure

7.2.3.3.1.2 Burst DMA Registers

There are four registers, as follows, that may be accessed from the SDMA core:

- MSA (Memory Source Address) - Holds the source byte address in the Arm platform memory map for reading data from this location. This register is automatically modified every time the core reads new data from the FIFO.
- MDA (Memory Destination Address) - Holds the destination byte address in the Arm platform memory map for writing data to this location. This register is automatically modified every time the core writes new data into the FIFO.
- MD (Memory Data) - Labels the 36-byte FIFO access point: Reading a byte, halfword, or word from MD respectively retrieves the first 1, 2, or 4 bytes of the

FIFO (for example, the bytes that were stored first by the DMA state-machine when transferring data from the Arm platform memory).

- When the FIFO does not hold as many bytes as required by the SDMA core, the core is stalled until the missing bytes are read from the Arm platform memory. In the case of prefetch mode, the DMA controller decides when it should start a burst to Arm platform memory in order to reduce the risk to not have the required data for the future accesses of the core. When there is no prefetching, a burst is triggered when the required data is not available in the FIFO.

  Writing a byte, halfword, or word to MD stores 1, 2, or 4 bytes, respectively, at the end of the FIFO (for example, these bytes are transmitted to the Arm platform memory after all the other bytes that were previously stored in the FIFO). When the FIFO does not have enough room left to hold the written data, the SDMA core is stalled until a sufficient amount of FIFO contents are flushed out to the Arm platform memory. Flushing is decided by the DMA controller when there are enough bytes in the FIFO to perform the largest allowed burst to Arm platform memory (the exact size depends on the burst start address and the AHB 1 Kbyte boundary rule). However, the SDMA core has the ability to force the flushing operation at any time, for example, when at the end of the data transfer, prior to channel closure.

- MS (Memory Setup) - Contains the state of the burst DMA control, the two flags that define whether each address register is incremented after every access to the external memory, and another flag that is set when a bus error occurred.

7.2.3.3.1.3 Burst DMA Data Transfers

Three typical usages have been identified that involve the burst DMA: the data transfer startpoint, the endpoint, or both.

Every case requires a different procedure, as listed in the following sections:

7.2.3.3.1.3.1 Data Retrieval from the Arm platform Memory

The following steps retrieve data from Arm platform memory using the burst DMA unit:

- Set up the MS flags to reflect the mode for the source address (incremented or frozen according to the type of accessed device: memory or peripheral FIFO), then initialize the source address register itself (MSA).
- Read data from the FIFO using the *ldf MD* instruction as many times as needed. If an error occurred during the fetch from Arm platform memory, the DMA control tags the error status on the data and the SDMA core SF flag is set when reading this data from the FIFO.

7.2.3.3.1.3.2 Storing Data Into the Arm platform Memory

The following steps store data from Arm platform memory using the burst DMA unit:

- Set up the MS flags to reflect the mode for the destination address (incremented or frozen according to the type of accessed device: memory or peripheral FIFO), then initialize the destination address register itself (MDA).
- Store data into the FIFO using the *stf MD* instruction as many times as needed.
- When the transfer is finished and if the DMA worked in automatic flush mode, force the flush of the FIFO. This instruction is stalled until all the FIFO data is effectively sent to the Arm platform memory and the error status of the transfer is available in the DF flag.

7.2.3.3.1.3.3 Transferring Data Between Two Arm platform Memory Locations-Burst DMA Unit

The following steps copy data between two Arm platform memory locations using the burst DMA unit:

- Set up the MS flags to reflect the modes for the source and destination addresses (all the combinations are possible), then initialize the source address register (MSA) and the destination address register (MDA). Both addresses must be word-aligned.
- Use as many *stf MD* instructions with the *COPY* flag as needed. Every instruction triggers a burst read of a given number of words from the source address (this number is provided to the burst DMA via the SDMA core general purpose register, which is referenced in the *stf* instruction). Once all the data is loaded into the FIFO, the DMA empties it with a write burst of the same count to the destination address. The DMA acknowledges prior to instruction completion, which frees the SDMA core for other tasks at no delay cost.
- Once the transfer is done, there should be a final access to the burst DMA to check the error status.

7.2.3.3.2 Peripheral DMA Unit

The peripheral DMA unit is the second functional unit that connects the SDMA to the Arm platform memory.

Unlike the burst DMA, it does not support burst transfers and is optimized for accessing peripherals. It does not provide control to assign a privilege level to the DMA access. Its feature list comprises the following:

- Access to the Arm platform peripherals or memory at once or twice the SDMA core frequency

- Data copy from one Arm platform memory location to another Arm platform memory location at memory bus speed, improving throughput
- Control of the method for addressing the Arm platform memory (automatic increment or decrement of addresses or frozen addresses, the first ones aimed at accessing RAM-like memory and the last one aimed at accessing single-address FIFOs)
- Selectable automatic prefetch when reading data from the Arm platform memory. In prefetch mode, the peripheral DMA automatically fetches another data-without waiting for the SDMA core to request it-when its data register is empty, which improves the throughput
- Selectable automatic flush. In this mode, the SDMA core may only be stalled when it tries writing data and the previous write operation is not finished yet; whereas, in forced flush mode, the core is stalled until the data is effectively written to the Arm platform memory.
- In automatic flush mode, the core receives an acknowledge that does not reflect the actual error status when the data is effectively written into the Arm platform memory or the peripheral. This error status is retrieved by a later access to the peripheral DMA. Terminating a write data transfer with a forced flush command guarantees that any bus error to the Arm platform memory has been caught.

This unit structure and registers are described in Peripheral DMA Structure and Peripheral DMA Registers.

7.2.3.3.2.1 Peripheral DMA Structure

The peripheral DMA is made up of a 32-bit data register, two address registers, and a controlling state-machine. The state-machine manages clock adaptation, when required.

It is shown in the following figure.

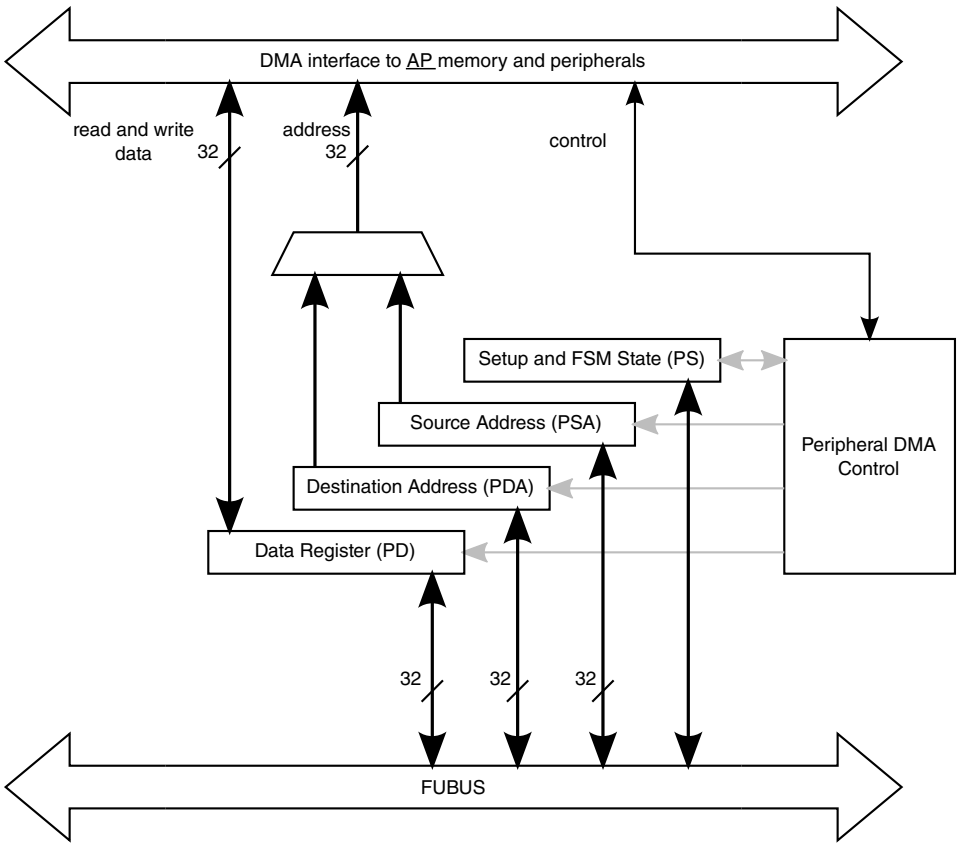


Figure 7-10. Peripheral DMA structure

7.2.3.3.2.2 Peripheral DMA Registers

According to Figure 7-10, the peripheral DMA has four registers that may be read or written by the SDMA core:

- *PD (Peripheral Data)* is the DMA 32-bit data register.
- *PSA (Peripheral Source Address)* holds the source byte address in the Arm platform memory map for reading data from this location. This register is automatically modified every time the core reads a new data from PD.

- *PDA (Peripheral Destination Address)* holds the destination byte address in the Arm platform memory map for writing data to this location. This register is automatically modified every time the core writes a new data into PD.
- *PS (Peripheral Setup)* contains the state of the peripheral DMA control, two configuration fields that define the way address registers are modified after every data access, two additional configuration fields that define the data size to access the source and destination devices, and another field that contains the latest transfer error status.

7.2.3.3.2.3 Peripheral DMA Data Transfers

There are three typical usages that involve the peripheral DMA, whether it is the data transfer start-point, endpoint, or both.

Every case requires a different procedure, as described in Data Retrieval from the Arm platform Memory or Peripheral, Storing Data into the Arm platform Memory or Peripheral, and Transferring Data Between Two Arm platform Memory Locations-Peripheral DMA Unit.

7.2.3.3.2.3.1 Data Retrieval from the Arm platform Memory or Peripheral

The following steps retrieve data from Arm platform memory using the peripheral DMA unit:

- Set up the PS fields to reflect the mode and data size for the source (incremented, decremented, or frozen address register; 8-bit, 16-bit, or 32-bit data transfers), then initialize the source address register itself (PSA) with an address that is aligned to the programmed data size.
- Read data from PD using the ldf PD instruction as many times as needed. If an error occurs during the fetch from the Arm platform memory or peripheral, the DMA control tags the error status on the data and the SDMA core SF flag is set when reading this data from PD.

7.2.3.3.2.3.2 Storing Data into the Arm platform Memory or Peripheral

The following steps store data to Arm platform memory using the peripheral DMA unit:

- Set up the PS fields to reflect the mode and data size for the destination (incremented, decremented, or frozen address register; 8-bit, 16-bit, or 32-bit data transfers), then initialize the destination address register itself (PDA) with an address that is aligned to the programmed data size.

- Store data into PD using the *stf PD* instruction as many times as needed.
- When the transfer is finished and if the peripheral DMA worked in automatic flush mode, force the flush of PD. This instruction is stalled until PD contents are effectively sent to the Arm platform memory or peripheral, and the error status of the transfer is available in the DF flag.

7.2.3.3.2.3.3 Transferring Data Between Two Arm platform Memory Locations-Peripheral DMA Unit

The following steps copy data between two Arm platform memory locations using the peripheral DMA unit:

- Set up the PS fields to reflect the modes and data size for the source and destination addresses (all the combinations of addressing modes are possible, but both data sizes must be identical), then initialize the source address register (PSA) and the destination address register (PDA). Both addresses must be aligned with the programmed data size.
- Use as many *stf PD* instructions with the *COPY* flag as needed. Every instruction triggers a single read from the source address; a single write of the received data immediately follows. The DMA acknowledges prior to instruction completion, which frees the SDMA core for other tasks at no delay cost.
- Once the transfer is done, there should be a final access to the peripheral DMA to check the error status.

7.2.3.4 SDMA Security Support

The SDMA provides support to SDMA software to block unauthorized updates to the scripts in RAM.

SDMA supports the following Security modes:

- Open Mode: has full control to load scripts and context into SDMA RAM. This is the default mode.
- Locked Mode: The Arm platform loads scripts and channel contexts at startup when it is still executing known safe software. When finished, it locks the SDMA to prevent further updates to RAM and selected registers. More details described in Locked Mode.

7.2.3.4.1 Locked Mode

The LOCK bit in the SDMA_LOCK register provides support for SDMA scripts to freeze RAM contents after the initial bootload routine to prevent future unauthorized updates to SDMA RAM.

After initial RAM contents are uploaded, Arm platform software can set the LOCK bit to secure the RAM contents to prevent future updates by an unauthorized. After the LOCK bit is written with a '1', the SDMA is "locked" until reset.

The LOCK bit can be read in the SDMA's internal memory map in the LOCK register (see Section SDMA LOCK (SDMAARM_SDMA_LOCK)). SDMA scripts which load information into RAM can check the value of the LOCK bit to determine if an upload to RAM is allowed. If not allowed, the script can refuse to allow the request to copy data into the RAM to continue. The exact use of the LOCK bit in SDMA scripts for security control will be described in SDMA software documentation (see SDMA Scripts).

While SDMA is locked, attempts to write to the SDMA_LOCK, CHN0ADR, ILLINSTADDR, and ONCE_ENB registers will be ignored. All registers remain readable. Writes to other registers are still allowed.

Once the SDMA is locked, the LOCK bit can only be cleared by a reset. A hardware reset will always clear the LOCK bit. A software reset initiated by writing to the RESET register will only clear the LOCK bit if the SRESET_LOCK_CLR bit in the SDMA_LOCK register is set. Since SDMA_LOCK register cannot be updated if SDMA is locked, the SRESET_LOCK_CLR bit must be configured before setting the LOCK bit. The SREST_LOCK_CLR bit will also be cleared by resets that clear the LOCK bit.

The SDMA RISC core uses the ILLINST and CHN0ADDR registers as pointers to determine where to jump to after an illegal instruction or upon boot after a reset. The LOCK bit prevents updates to these registers to protect against unauthorized changes to these pointers.

While SDMA is locked, the ONCE_ENB register cannot be written to prevent the OnCE under Arm platform control from being used to gain access to SDMA internal memory. If Arm platform control of the OnCE is enabled before setting the LOCK bit, the Arm platform can use the ONCE for debug purpose after LOCK is set.

7.2.3.5 OnCE and PCU Debug States

The SDMA has two different debug modes in which the OnCE performs debug instructions.

Refer to Figure 7-4 for an example of the PCU states in debug. The following are the two debug states:

- When a channel is running (that is, when CCR and CCPRI are different from 0, which can be read in the PSW register), SDMA can execute a SoftBkpt instruction from the channel script or receive a debug request. When either happens, the SDMA enters its "Classical" *Debug* state, which is described in OnCE and Real-Time Debug.
- When a channel is not running, the SDMA can be in *Sleep* state or in *Sleep after Reset* state. If a debug request is sent to the core, it enters its *Debug in Sleep* state. This debug mode works similarly to the "Classical" *Debug* state, except it returns to the original state (*Sleep* or *Sleep after Reset*) when the debug mode is left via the exec_core instruction of the OnCE. From this *Debug in Sleep* state, the SDMA can execute a program whereas no channel is running. If a new debug request is sent to the core or if a SoftBkpt is executed, it comes back to this *Debug in Sleep* state.

The OnCE is provided with several instructions that can be executed when the core is in either debug state. The following table summarizes the behavior of these OnCE debug instructions. There exists other secondary OnCE instructions that are described in OnCE and Real-Time Debug.

Table 7-12. SDMA in Debug Mode

Instruction	Debug	Debug in Sleep
exec_once	exec_once <instruction> SDMA executes the <instruction> and returns to the *Debug* state. The Program Counter (PC) is not incremented. This command must not be used with an instruction that modifies the PC value.	exec_once <instruction> SDMA executes the <instruction> and returns to the *Debug in Sleep* state. The Program Counter (PC) is not incremented. This command must not be used with an instruction that modifies the PC value.
run_core	run_core <instruction> SDMA executes the <instruction>, leaves the *Debug* state and continues executing the channel script from the position where it stopped. This command must not be used with an instruction that modifies the PC value.	run_core <instruction> SDMA executes the <instruction> and returns to its *Sleep* or *Sleep after Reset* initial state. This command must not be used with an instruction that modifies the PC value.
exec_core	exec_core <instruction> It is similar to run_core except it requires an instruction that changes the PC value (jump, branch...): the SDMA jumps to the new PC value, leaves the *Debug* state and starts executing instructions from this new PC value.	exec_core <instruction> If the previous state was *Sleep after Reset*, the SDMA returns to this state, and Chn0Addr value overrides the PC value. Otherwise, the SDMA jumps to the new PC value and starts executing instructions from this new PC.

NOTE

The feature exec_core in *Debug in Sleep* after *Sleep after Reset* was added for the Channel boot (channel 0) to allow the debugger to return to *Sleep after Reset* state with a new PC

value. The SDMA will be ready to boot at the Chn0Addr address.

7.2.3.6 SDMA Clocks and Low Power Modes

The SDMA receives several root clocks from the SoC clock controller block and performs adaptive clock gating to optimize its power consumption. From a user standpoint, clock gating and power mode selection are fully automatized inside the SDMA.

Root clock control is available from the SoC clock controller block.

There are numerous clock sources that are used in the SDMA. They belong to one of two possible clock domains listed in the following table, and have frequency constraints within each domain. Clocks are considered asynchronous between domains.

Within the Arm platform/SDMA clock domain, all clocks must come from the same DPLL. The Arm platform DMA interfaces (peripheral DMA and burst DMA) receive their clock from the Arm platform DMA clock source whose frequency can be once or twice the frequency of the SDMA core clock. The DMA interfaces are designed to work at the Arm platform DMA frequency, but the SDMA core is physically limited to a maximum 104 MHz frequency. Since this is lower than the maximum Arm platform DMA frequency, the SDMA core clock is tied to the Arm platform peripheral clock frequency.

The Arm platform Peripheral Bus Clock source must be an exact sub-frequency of the SDMA Core clock source (any integer value greater or equal to 1).

Table 7-13. Clocking Scheme

Clock Domain	Source Clock	Comments
Arm platform	SDMA core (SDMA main core)	Source clock for the core and all its operations; this clock is thus used by most of the SDMA sub-blocks.
	Arm platform DMA	DMA interface for the peripheral DMA and the burst DMA. It is balanced with the main clock source, and its frequency is either once or twice the main clock frequency.
	Arm platform peripheral	Connection to the Arm platform peripheral bus. It is a sub-frequency of the main clock frequency.
JTAG	TCK	Clock for JTAG access, limited to maximum of 1/8 of the SDMA core clock frequency.

The JTAG clock is sampled by the SDMA main clock to determine its rising edge. This simplifies design and clock management, but it also adds a ratio constraint between those two clocks. It is guaranteed the JTAG interface works properly when the frequency of TCK is lower than $1/8^{th}$ of the frequency of the SDMA main clock (which is about 8 MHz when the SDMA core clock frequency is 66 MHz).

7.2.3.6.1 Clock Gating and Low Power Modes

The SDMA automatically performs power saving without requiring user involvement. It implements two levels of automatic clock gating.

7.2.3.6.1.1 Coarse Clock Gating

Every sub-block clock comes from one of the five available sources, and is gated with the sub-block specific enabling condition.

The following table displays the sub-block clocks and their source. It also indicates the relationships that may exist between different sub-blocks clock enables.

Table 7-14. Sub-blocks Clocks

Sub-block	Source Clocks	Enabling Condition and Comments	Related Enabling Conditions
Core	SDMA Main Core	The core sub-block clock is running when the core is not in one of its sleep states (Sleep or Sleep after Reset) or there is a pending channel. Typically, the core sub-block clock is stopped once all the channels are processed and the core enters its sleep state. A new pending channel awakes the core sub-block clock.	None
Memories	SDMA Main Core	The clock activation only occurs during a core access.	Disabled when Core sub-block clock is disabled or no memory access in progress
Scheduler	SDMA Main Core	Its clock only runs when scheduling is needed: for example, when there are pending channels, upon reception of a DMA request, and anytime the Arm platform modifies the channel running conditions.	None
Arm platform Control	SDMA Main Core & Arm platform peripheral	The Arm platform peripheral clock is solely used to determine the frequency ratio with the SDMA main clock. The control registers' clock is based on *SDMA main clock*; it is active when the Arm platform or the SDMA modifies the contents of one of these registers.	None
Burst DMA	SDMA Main Core & Arm platform DMA	The burst DMA has two clocks: The first clock is derived from the SDMA main core clock and drives registers that are connected to the FUBUS. The second clock is derived from the Arm platform DMA clock and drives registers that are connected to the Arm platform DMA bus outside the SDMA. Both clocks are enabled	Disabled when Core sub-block clock is disabled

Table continues on the next page...

Table 7-14. Sub-blocks Clocks (continued)

Sub-block	Source Clocks	Enabling Condition and Comments	Related Enabling Conditions
		during active phases of data transfers (for example, these clocks are turned off when the burst DMA is not used by the running channel script).	
Peripheral DMA	SDMA Main Core & Arm platform DMA	The peripheral DMA has two clocks: The first clock is derived from SDMA main clock and drives registers that are connected to the FUBUS. The second clock is derived from the Arm platform DMA clock and drives registers that are connected to the Arm platform DMA bus outside the SDMA. Both clocks are enabled during active phases of data transfers (for example, these clocks are turned off when the peripheral DMA is not used by the running channel script).	Disabled when Core sub-block clock is disabled
OnCE	SDMA Main Core	The OnCE clock is derived from main source clock. It is disabled by default. In order to use the OnCE, its clock must be explicitly turned on, either by enabling the OnCE access from the Arm platform peripheral bus (register ONCE_ENB), or by driving the clk_gating_off input pin high. This is a SDMA input whose driver depends on the SoC implementation (typically a JTAG controller). The OnCE also receives the TCK input, which is the JTAG clock. It does not use it as a functional clock; the TCK input is sampled instead. Refer to Synchronization Implementation.	When enabled, all other clocks are systematically on (clock gating is off)

7.2.3.6.1.2 Refined Clock Gating

The SDMA implements a second level of clock gating on a register-per-register basis.

Unlike the first level that covers all the SDMA flip-flops, except the synchronizers (only five flip-flops are always running), the second level is only available for eligible registers, which amounts to about 90% of the SDMA flip-flops.

These gated registers are only clocked when the hardware logic detects a new data loading. This additional gating further reduces dynamic power consumption.

7.2.3.6.1.3 Low Power Modes and User Control

Power savings are automatically managed by the SDMA hardware without any user involvement; however, one can distinguish three different power modes: SLEEP, RUN, and DEBUG.

The following table describes these modes, and shows how to switch from one mode to another.

Table 7-15. Power Modes

Power Mode	Sub-blocks							Comments
	Core	Memories	Scheduler	Arm platform Control	Burst DMA	Peripheral DMA	OnCE	
SLEEP	off[1]	off	wait[2]	wait	off	off	off	Set when the PCU state is either *Sleep* or *Sleep after Reset* and the SDMA is not in DEBUG mode. This is the default mode after reset.
RUN	on[3]	wait	wait	wait	wait	wait	off	Set for the other PCU states that are reachable out of debug: *Program*, *Data*, *Change of Flow*, *Error in Loop*, *Debug*, *Functional Unit*, *Save,* or *Restore*.
DEBUG	on	on	on	on	on	on	on	Set regardless of the PCU state when clock gating is turned off to use the OnCE features (either *clk_gating_off* pin high or ONCE_ENB[0] set).

1. *off*: no clock
2. *wait*: only clocked when accessed or stimulated
3. *on*: clock is always running

It is possible to control the SDMA power mode. The procedures to force the SDMA into either mode are described in SLEEP Mode.

7.2.3.6.1.3.1 SLEEP Mode

This is the default mode after reset; therefore, resetting the SDMA forces this mode.

However, the common procedure is as follows:

- Ensure the *clk_gating_off* pin is low and ONCE_ENB[0] is cleared.
- Disable all channels (via the STOP_STAT control register, and the HO, DO, EO if necessary).
- Wait for the active channels to complete or force a reschedule via the reschedule bit in the RESET register.
- The SDMA is in SLEEP mode making it possible to completely shut off its clock from the chip level clock controller using the procedure described in Stop Mode Response.

7.2.3.6.1.3.2 RUN Mode

This is the default mode when a channel is running:

- Ensure the *clk_gating_off* pin is low and ONCE_ENB[0] is cleared.
- Activate at least one channel (via the HSTART control registers, a DMA request, and/or the HO, DO, EO register bits).

7.2.3.6.1.3.3 DEBUG Mode

The DEBUG mode must be set when one needs to use the debugging facilities of the SDMA.

- Ensure the SDMA clocks are running from the CCM.
- Set the *clk_gating_off* pin high or use the SDMA to set ONCE_ENB[0].

7.2.3.6.1.4 Stop Mode Response

The SDMA receives a stop request from the chip level clock controller. This request may be asserted when the chip enters the stop low power mode.

If the SDMA is running when the request is received, then the SDMA will complete all pending channels before returning to the SLEEP state. The SDMA sends an acknowledgement to the clock controller when the SLEEP state is entered indicating that the SDMAs clocks can be turned off.

7.2.3.6.2 Reset

After reset (either received from the reset block or a software reset required by the Arm platform), the SDMA is in IDLE mode. It will start its boot code located at address 0 once a channel is activated.

Activating a channel can be done by the Arm platform after programming a positive priority and setting the channel bit in the EVTPEND register.

There will not be a context RESTORE for the first channel (bootload channel) called after a reset because the context data in RAM has not been initialized. Static context mode should be used for the first channel called after reset to ensure that the all context RAM for that channel is initialized. Subsequent calls to the same channel or different channels may use any of the dynamic context modes

7.2.3.7 Software Interface

Appendix A fully describes the SDMA Application Programming Interface (API).

7.2.3.8 Initialization Information

This section discusses the following:

- Hardware Reset
- Channel Script Execution
- Initialization and Script Execution Setup Sequence

7.2.3.8.1 Hardware Reset

After reset, the program RAM, context RAM, data RAM, and RAM containing the channel enable registers (CHNENBLn) have unpredictable contents.

The active register set is assigned to channel 0 and the PC is initialized to all zeros. However, since the channel enable register is all zeros, there are no active channels and the SDMA is halted waiting for the boot channel to start.

The Arm platform will have to setup the SDMA in order to boot it. The CONFIG register must be initialized to determine the DMA/core clock ratio (1 or 2). Channel Enable Registers must also be initialized.

To start up the SDMA, the Arm platform first creates some channel control blocks (CCB) and buffer descriptors (BD) in Arm platform memory for the boot channel (channel 0) and then initializes the channel 0 pointer register (SDMA_MC0PTR) to the address of the first control block. Data Structures for Boot Code and Channel Scripts provides an overview of the data structure for the CCB and BD's. The SDMA_HSTART, SDMA_HOSTOVR and SDMA_EVTOVR registers are then configured according to Runnable Channels Evaluation to allow channel 0 to run.

Upon being enabled, the SDMA begins executing the script located at the address indicated by the Channel 0 Boot Address register (SDMA_CHN0ADDR) in the program memory. The reset value of SDMA_CHN0ADDR points to the default bootload script in ROM. This ROM script will read the channel 0 pointer register (SDMA_MC0PTR) to determine the location of the Channel Control Block (SDMA_CCB) in Arm platform memory. The script will then begin fetching by DMA the first channel control block which contains a pointer to the location channel 0 Buffer Descriptor chain which is also fetched via DMA. If the buffer descriptor contains a valid command, the script interprets the command in each buffer descriptor and proceeds to implement the command and move on to the next buffer descriptor control block. The buffer descriptor commands for channel zero are typically set up to load SDMA's program RAM, Data RAM, and initial values for the channel contexts. Some channel scripts expect particular parameters to be passed

There are two ways to make the SDMA boot on a user-defined script. The OnCE (either via its JTAG interface or its Arm platform Control interface) can be used to download any code in the SDMA RAM and force the SDMA to boot on that code. Also, the SDMA_CHN0ADDR register in the Arm platform programming model can be modified to point to user code in RAM which would need to either have been loaded via the ONCE or default bootload routine (ex before a S/W reset).

7.2.3.8.2 Channel Script Execution

The execution of an SDMA script depends on both the instructions that make up the script, the data context upon which it operates, and commands or parameters allowed to the buffer. All these items must be initialized before the script is allowed to execute.

Each of the 32 channels has a separate context, but may share scripts and locations in data RAM.

Channels are initialized by the Arm platform by using channel 0 to download any required scripts and data values and the channels initial context. The context contains all the initial values of the SDMA core registers. This includes the Program Counter (PC) which is set to the start of the desired script in SDMA program memory.

The Arm platform selects which trigger conditions that must occur for the channel to start by configuring the SDMA_CHNENBL, SDMA_HOSTOVR and SDMA_EVTOVR registers. The trigger events include Arm platform setting HE (SDMA_HSTART) or a hardware DMA request asserts an event input to SDMA. The channel can become active according to its priority compared with other runnable channels when the selected trigger(s) cause the condition described in Runnable Channels Evaluation to evaluate as true.

The specific parameters to be passed to each script in the buffer descriptor or context are documented in the software documentation for each script. Please refer to SDMA Scripts for complete script documentation. Buffer Descriptor Format provides an overview of the buffer descriptor format.

7.2.3.8.3 Initialization and Script Execution Setup Sequence

To summarize, the following steps are minimally required to setup SDMA and run channel scripts.

- Perform Hardware Reset. The program RAM, context RAM, data RAM and SDMA_CHNENBLn registers have unpredictable contents after this reset.
- Initialize SDMA_CHNENBLn registers to map DMA request events to desired channels.

- Configure SDMA_CHNPRIn registers to select priority for runnable channels. A non-zero priority is required for the channel to run.
- Configure the SDMA_CONFIG register to select DMA to SDMA core clock ratio .
- Set up channel control blocks and buffer descriptors in Arm platform to specify the loading of SDMA program RAM and channel contexts for each SDMA channel to be used. Reference Data Structures for Boot Code and Channel Scripts.
- Configure SDMA_MC0PTR register with base address of Arm platform Channel Control Block base address.
- Initialize SDMA_CHNENBLn registers to map DMA request events to associated channel. Reference Mapping DMA Requests to Pending Channels.
- Configure SDMA_CHNPRIn registers to set priority for each channel to be run.
- For each channel to be run, configure SDMA_HOSTOVR (HO) and SDMA_EVTOVR (EO) registers to select which events (hardware and/or software trigger events) must occur for the channel to be runnable. Reference Runnable Channels Evaluation.
- Set bit 0 of the SDMA_HSTART register to set HE[0] and allow Channel 0 to run (assumes EO[0] andDO[0] were both set in previous step). This will cause SDMA to load the program RAM and channel contexts configured previously.
- Wait for Channel 0 to finish running. This is indicated by HI[0]=1 in the SDMA_SDMA_INTR register, or by optional interrupt to the Arm platform.
- Set the LOCK bit in the SDMA_SDMA_LOCK register to prevent un-authorized uploads of data to SDMA RAM.
- Additional channel scripts can now be run by enabling the selected software or hardware trigger event according to Runnable Channels Evaluation.

7.2.3.9 SDMA Programming Model

This section describes the programming model for the SDMA RISC engine, including its processor, memory, and internal control registers.

All addresses are related to the internal SDMA memory map, which is completely different from the Arm platform memory maps. The Arm platform processor has no access to any hardware resource described, except when those resources are described in Arm Platform Memory Map and Control Register Summary. .

7.2.3.9.1 State and Registers Per Channel

The SDMA can be seen as a set of 32 identical devices that are able to perform one data transfer channel each. Only one channel can work at a time, but every channel state is available at any time.

This chapter lists the components of every channel state.

7.2.3.9.2 General Purpose Registers

Each channel has eight general purpose registers of 32 bits for use by scripts. General register 0 has a dedicated function for the loop instruction, but otherwise can be used for any purpose.

7.2.3.9.3 Functional Unit State

Each channel context has some state that is part of the functional units.

The specific allocation of this state is part of the functional unit definition that is described in Burst DMA Unit Programming, Peripheral DMA Unit Programming .

This state must be saved/restored on context switches.

7.2.3.9.3.1 Program Counter Register (PC)

The PC is 14 bits. Since instructions are 16 bits in width and all memory in the SDMA is 32 bits in width, the low order bit of the PC selects which half of the 32-bit word contains the current instruction.

A low order bit of zero selects the most significant half of the word.[1]

7.2.3.9.3.2 Flags

Each channel has the following four flags:

- The T bit reflects the status of some arithmetic and test instructions. It is set when the result of an addition or a subtraction is zero and cleared otherwise. It is also the copy of the tested bits. Finally, it can also be set when the loop counter (GReg0) reaches zero. When the last instruction of the hardware loop is an operation that can modify the T flag, its effect on T is discarded and replaced by the GReg0 status.
- Two additional bits, SF and DF, are used to indicate error conditions resulting from loading data sources and storing to destinations, respectively. Access errors set these bits, and successful transactions clear them. They can also be cleared by specific instructions (CLRF and loop). The source fault (SF) is updated by the loads LD and LDF; the destination fault (DF) is updated by the stores ST and STF.

1. For example, big-Endian.

- Access errors are caused by several conditions including writing to the ROM, writing to a read-only memory mapped register, accessing an unmapped address, or any transfer error received by a peripheral when it is accessed.

  The SF and DF flags have a major impact on the behavior of the hardware loop: If SF or DF is set when starting a hardware loop and it is not masked by the loop instruction, the loop body will not be executed. Inside the loop body, if a load or store sets the corresponding SF or DF flag, the loop exits immediately. Testing the status of the T flag at the end of the loop (as well as testing both SF and DF) tells if the loop exited abnormally as any anticipated exit prevents GReg0 from reaching the zero value and thus setting the T flag. This is also valid if the fault occurs at the last instruction of the last loop.

- The last flag is the loop mode flag, LM, which is composed of two bits. The most significant bit indicates when the processor is currently operating in loop mode. It is set by the loop instruction and is cleared after execution of the last instruction of the last loop. The least significant bit is set when the program counter points to the last instruction of a loop on the last path. It is used for a channel that is restored with this configuration to know that the next program counter is EPC. As with the dynamic context switch Greg0, which indicates when the program must get out of the loop, it can be restored only on the last instruction of the loop. This, however, is too late to fetch the next instruction after the loop.

7.2.3.9.3.3 Return Program Counter (RPC)

The RPC is 14 bits. It is set by the jump to the subroutine instructions and used by the return from the subroutine instructions.

Instructions are available to transfer its contents to and from a general register.

7.2.3.9.3.4 Loop Mode Start Program Counter (SPC)

The SPC is 14 bits. It is set by the loop instruction to the location immediately following it.

7.2.3.9.3.5 Loop Mode End Program Counter (EPC)

The EPC is 14 bits. It is set by the loop instruction to the location of the next instruction after the loop.

7.2.3.9.4 Context Switching-Programming

Each channel has a separate context consisting of the eight general purpose registers and additional registers representing the state of the functional units.

The active registers and functional units contain the context of the active channel. The context of inactive channels is stored in SDMA RAM, which is part of the SDMA address space.

In a function of the selected context switching mode (Context Switching), modified registers by the program can be saved in the channel RAM space while the program is going on. In every cycle, a write access to the RAM is possible.

On a done or yield(ge) instruction, SDMA goes into "real" context switching. In one of the dynamic modes, modified registers not previously saved, as well as the PC-Loop registers, are stored into the context area of the channel that will be closed. The new PC-Loop registers are loaded from the context area of the new channel. All other registers are restored while the program is executed, giving priority to registers used by the decoded instruction. Therefore, in the best case, only the PC and Loop registers should be saved and restored during this context-switching phase, which only requires five SDMA cycles.

In static mode, the context switch stores all registers in the old channel RAM space, and restores all registers from the new channel RAM space. It requires 26 SDMA cycles.

The address of the context memory for channel *i* is *CONTEXT_BASE* + 24**i* or *CONTEXT_BASE* + *32*i w*here CONTEXT_BASE equals 0x0800. The table below presents the layout of a channel context in memory:

Table 7-16. Layout of a Channel Context in Memory for SDMA

OFFSET	31	30	29-16	15	14	13-0
0	SF	-	RPC	T	-	PC
1	LM		EPC	DF	-	SPC
2	GR0					
3	GR1					
4	GR2					
5	GR3					
6	GR4					
7	GR5					
8	GR6					
9	GR7					
10	MDA (burst DMA)					
11	MSA (burst DMA)					
12	MS (burst DMA)					
13	MD (burst DMA)					

Table continues on the next page...

Table 7-16. Layout of a Channel Context in Memory for SDMA (continued)

14	PDA (peripheral DMA)
15	PSA (peripheral DMA)
16	PS (peripheral DMA)
17	PD (peripheral DMA)
18	
19	
20	Reserved[1]
21	Reserved[1]
22	Reserved[1]
23	Reserved[1]
24	Scratch RAM (optional)
25	Scratch RAM (optional)
26	Scratch RAM (optional)
27	Scratch RAM (optional)
28	Scratch RAM (optional)
29	Scratch RAM (optional)
30	Scratch RAM (optional)
31	Scratch RAM (optional)

7.2.3.9.5 Address Space

The SDMA has four internal buses which are listed here.

- The Instruction bus reads instructions from the memory. Its address map is described in Instruction Memory Map.
- The Data bus (DMBUS) accesses the same memories as those visible on the Instruction bus, some memory-mapped registers (scheduler status and OnCE registers), and up to 14 peripherals. Its address map is described in Data Memory Map.
- The Functional Units bus (FUBUS) accesses the , Burst DMA, Peripheral DMA . The addressing mechanism is further detailed in Functional Units Programming Model.
- The Context Switch bus reads/writes registers into context-switch RAM space. It is a 64-bit bus dedicated for accessing this RAM space for updating the context of the running channel. While the program is going on, this bus has the lowest priority compared to the Instruction and Data buses, except for restoring a register needed for the decoded instruction to be executed. On the save part of a context switch (when the PCU is in its slave state), this is the only one used. On the restore part, the Instruction bus has the priority to read the next instruction at the restored PC and

otherwise the Context Switch bus is used. It is not possible to control the actual data transfers that occur on this bus.

7.2.3.9.5.1 Instruction Memory Map

The instruction memory map is based on a 14-bit address bus and a 16-bit data (instruction) bus.

Instructions are fetched from either program ROM or program RAM. An SDMA script is able to change the contents of the program RAM, which is also visible from the data bus.

The first two instruction locations (at 0 and 1) are special. Location 0 is where the PC is set on reset. Location 1 is where the PC is set upon the execution of an illegal instruction. It is expected that both of these locations will contain a jmp to handle routines.

Table 7-17. SDMA Instruction Memory Space

Device	SDMA Address (Hex)	Base Address Label	Block Name	WS	Description
ROM	0x0000 ↓ 0x07FF	SDMA_IBUS_ROM_ADDR	-	0	4 Kbyte internal ROM with boot code and standard routines.
RAM	0x1000 ↓ 0x1FFF	SDMA_IBUS_RAM_ADDR	-	0	8 Kbyte internal RAM with channels context and user data/routines.

7.2.3.9.5.2 Data Memory Map

All of the data accessible to SDMA scripts make up the data memory space of the SDMA.

This address space has several components:

- ROM (also visible on the Instruction bus)
- RAM (also visible on the Instruction bus)
- Shared Peripherals Registers
- SDMA Internal Registers (scheduler, OnCE, and registers that are also accessible by the Arm platform)

SDMA scripts can read and write to the context RAM, data RAM, shared peripheral registers, and internal registers.

The address range is 16 bits and the data width is 32 bits. When accessing peripheral registers (USB and so on), the data width may be different. The exact address map for the peripherals depends on the project (as presented in each respective chapter).

Data access is performed with *ld* and *st* instructions that take the address from a general purpose register in the core (GReg*n*). The mapping between the general purpose register contents and the address bus is given in the following table:

Table 7-18. GReg*n* to DMBUS Address Mapping

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sz	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
address															

Grayed bits are simply discarded but they must be cleared to ensure forward-script compatibility.

- sz (bit 31) indicates the peripheral data width: 0 is used for a 32-bit peripheral and 1 is used for a 16-bit peripheral.
- address (bits 15 down to 0) is the address of the accessed resource (internal memory, internal register, or shared peripheral).

Table 7-19. SDMA Data Memory Space

Device	SDMA Address (Hex)	Size	Description
ROM	0x0000 → 0x03FF	4 Kbyte	4 Kbyte internal ROM with boot code and standard routines
Reserved	0x0400 → 0x07FF	4 Kbyte	4 Kbyte Reserved
RAM	0x0800 → 0x0FFF	8 Kbyte	8 Kbyte internal RAM with channels contexts and user data/routines
per1	0x1000 → 0x1FFF	16 Kbyte	*peripheral 1* memory space (4 Kbyte peripheral's address space)
per2	0x2000 → 0x2FFF	16 Kbyte	*peripheral 2* memory space (4 Kbyte peripheral's address space)
per3	0x3000 → 0x3FFF	16 Kbyte	*peripheral 3* memory space (4 Kbyte peripheral's address space)
per4	0x4000 → 0x4FFF	16 Kbyte	*peripheral 4* memory space (4 Kbyte peripheral's address space)
per5	0x5000 → 0x5FFF	16 Kbyte	*peripheral 5* memory space (4 Kbyte peripheral's address space)
per6	0x6000 → 0x6FFF	16 Kbyte	*peripheral 6* memory space (4 Kbyte peripheral's address space)
Registers	0x7000 → 0x7FFF	16 Kbyte	Memory mapped registers
per7	0x8000 → 0x8FFF	16 Kbyte	*peripheral 7* memory space (4 Kbyte peripheral's address space)
per8	0x9000 → 0x9FFF	16 Kbyte	*peripheral 8* memory space (4 Kbyte peripheral's address space)
per9	0xA000 → 0xAFFF	16 Kbyte	*peripheral 9* memory space (4 Kbyte peripheral's address space)
per10	0xB000 → 0xBFFF	16 Kbyte	*peripheral 10* memory space (4 Kbyte peripheral's address space)
per11	0xC000 → 0xCFFF	16 Kbyte	*peripheral 11* memory space (4 Kbyte peripheral's address space)
per12	0xD000 → 0xDFFF	16 Kbyte	*peripheral 12* memory space (4 Kbyte peripheral's address space)
per13	0xE000 → 0xEFFF	16 Kbyte	*peripheral 13* memory space (4 Kbyte peripheral's address space)
per14	0xF000 → 0xFFFF	16 Kbyte	*peripheral 14* memory space (4 Kbyte peripheral's address space)

7.2.3.10 SDMA Initialization

Appendix A describes the setup of the SDMA . This section provides a quick description of several initialization procedures.

NOTE

There may be differences with the actual implementation in the API.

7.2.3.10.1 Hardware Reset-SDMA

After reset, the RAM that holds contexts, data, scripts, and the DMA request-channels matrix has unpredictable content.

The core registers are all reset to 0, including the PC; the PCU state is *Sleep after Reset*. No channel can be activated because all of the priorities are also reset to 0.

7.2.3.10.2 Standard Boot Sequence

The following is the standard boot sequence:

1. Initialize the CONFIG register-detailed in Configuration Register (SDMAARM_CONFIG)-to determine the Arm platform DMA/core clock ratio (1 or 2)
2. Initialize the DMA request-channels matrix (seeChannel Enable RAM (SDMAARM_CHNENBL*n*)).
3. Program the channel control registers-Channel Event Override (SDMAARM_EVTOVR), Channel BP Override (SDMAARM_DSPOVR), Channel BP Override (SDMA_HOSTOVR), and Channel Event Pending (SDMAARM_EVTPEND)-according to the channel allocation.
4. Perform any necessary setup as required by the standard boot script in ROM (this is described in Appendix A).
5. Trigger channel 0 with the Channel Start (SDMAARM_HSTART) register, which starts the execution of the ROM script starting at address 0. This boot downloads channel scripts and contexts in RAM.

7.2.3.10.3 User-Defined Boot Sequence

The following is a user-defined boot sequence:

1. Initialize the Configuration Register (SDMAARM_CONFIG)Channel Enable RAM (SDMAARM_CHNENBL*n*), Channel Event Override (SDMAARM_EVTOVR), Channel BP Override (SDMAARM_DSPOVR), Channel Arm platform Override (SDMAARM_HOSTOVR), and Channel Event Pending (SDMAARM_EVTPEND).

2. Use the OnCE (either via its JTAG interface or its Arm platform control registers) to download any code in the SDMA RAM. Accessing the Memory describes how to write data to the RAM via the OnCE.
3. Use the OnCE instructions to make the PC default value point to the new boot script start address, or rely on the ROM startup script, which first jumps to the address in Channel 0 Boot Address (SDMAARM_CHN0ADDR). (This register default address points to the standard boot script.)

7.2.3.10.4 Script Loading and Context Initialization

The execution of an SDMA script depends on both the instructions that make up the script and the data context upon which it operates. Both must be initialized before the script is allowed to execute.

Each of the 32 channels has a separate data context, but may share scripts and locations in the data RAM.

The Arm platform manages the space in program RAM and data RAM. It also manages the assignment of SDMA channels to the device drivers that need them. Channels are initialized by the Arm platform via the channel 0 boot script. The boot channel downloads any required scripts with their data and the channels' initial contexts. Every context contains all the initial values of the registers, including the PC. Then the Arm platform can enable any channel that becomes active and begins fetching and executing instructions from its script.

7.2.3.11 Instruction Description

The following sections introduce the instruction of the SDMA.

Instruction set details are available in Instruction Set.

7.2.3.11.1 Scheduling Instructions

The following are scheduling instructions:

- done-The instruction causes certain scheduling or interrupt bits to be set or cleared, which may cause a change in the schedule-ability of the running channel. Then the instruction causes the SDMA to evaluate the current scheduling priorities and to choose the highest priority ready channel. If this channel is not the current channel, a context switch will take place. If there are no runnable channels, the SDMA will enter the stopped mode. The done 5 has a special usage reserved for debug, as explained in Debug Instructions.

- yield-These instructions are special cases of the done instruction. They do not modify the scheduling bits, but allow the highest pending channel (if it exists) to preempt the current channel if the pending channel priority is strictly greater than the current channel priority.
- yieldge-These instructions are special cases of the done instruction. They do not modify the scheduling bits, but allow the highest pending channel (if it exists) to preempt the current channel if the pending channel priority is strictly greater or equal to the current channel priority.
- notify-The notify instruction affects the scheduling bits, but does not cause rescheduling.

7.2.3.11.2 Conditional Branch Instructions

The conditional branch instructions of an 8-bit displacement, which is sign-extended and added to the current PC (which points to the next instruction) if the condition is satisfied.

Otherwise, control passes to the next sequential instruction.

- BF-Branch if False. The branch is taken if the T bit in the processor status is zero (false).
- BT-Branch if True. The branch is taken if the T bit in the processor status is one (true).
- BSF-Branch if Source Fault. The branch is taken if the SF bit in the processor status is one.
- BDF-Branch if Destination Fault. The branch is taken if the DF bit in the processor status is one.

7.2.3.11.3 Unconditional Jump Instructions

There are two varieties of unconditional control transfers: an absolute transfer and a through-register transfer.

Absolute transfers have a 14-bit address field that replaces the current PC.

- JMP-Jump. Causes the processor to jump to an absolute address encoded in the instruction itself.
- JSR-Jump to Subroutine. Causes the processor to jump to a subroutine, the address of which is encoded in the instruction itself.

- JMPR-Jump through Register. Causes the processor to jump to an absolute address contained in a General register. This instruction is meant to be used when more than one level of subroutines are required.
- JSRR-Jump to Subroutine through Register. Causes the processor to jump to a subroutine, the address of which is contained in a General register. This instruction is meant to be used when more than one level of subroutines are required.

7.2.3.11.4 Subroutine Return Instructions

The following are subroutine return instructions:

- RET-Return from Subroutine. The RET restores the contents of RPC to PC.
- LDRPC-Load from RPC to Register. THe LDRPC instruction is meant to be used when more than one level of subroutines are required. It stores the contents of RPC in any General register.

7.2.3.11.5 Loop Instruction

The following is a loop instruction:

LOOP-Enters Loop Mode. Before entering loop mode, the loop instruction can optionally clear the fault flags (SF and/or DF) based on a 2-bit field in the instruction. This feature is linked to the fact that setting SF or DF in loop mode will cause an immediate exit of the loop.

7.2.3.11.6 Miscellaneous Instructions

The following are miscellaneous instructions:

- CLRF-Clear Fault Flags. This instruction clears any combination of SF and DF.
- MOV r,s-This moves data from GReg[s] to GReg[r].
- LDI r,immediate-This loads GReg[r] with a zero-extended immediate value.

7.2.3.11.7 Logic Instructions

The following are logic instructions:

- XORr,s-This performs an exclusive or between GReg[r] and GReg[s], and stores the result in GReg[r].
- XORIr,immediate-This performs an exclusive or between GReg[r] and a zero-extended immediate value, and stores the result in GReg[r].
- ORr,s-This performs an or between GReg[r] and GReg[s], and stores the result in GReg[r].

- ORIr,immediate-This performs an or between GReg[r] and a zero-extended immediate value and, stores the result in GReg[r].
- ANDNr,s-This performs an and between GReg[r] and the negated GReg[s], and stores the result in GReg[r].
- ANDNIr,immediate-This performs an and between GReg[r] and the negated zero-extended immediate value, and stores the result in GReg[r].
- ANDr,s-This performs an and between GReg[r] and GReg[s], and stores the result in GReg[r].
- ANDIr,immediate-This performs an and between GReg[r] and a zero-extended immediate value, and stores the result in GReg[r].

7.2.3.11.8 Arithmetic Instructions

Arithmetic instructions modify the T bit in the processor status according to the result of the operation. The T bit is set if the result is zero, otherwise it is cleared.

- ADD r,s-This performs the addition of GReg[r] and GReg[s], and stores the result in GReg[r].
- ADDI r,immediate-This performs the addition of GReg[r] and a zero-extended immediate value, and stores the result in GReg[r].
- SUB r,s-This performs the subtraction of GReg[s] from GReg[r], and stores the result in GReg[r].
- SUBIr,immediate-This performs the subtraction of a zero-extended immediate value from GReg[r], and stores the result in GReg[r].

7.2.3.11.9 Compare Instructions

Compare instructions modify the T bit in the processor status according to the result of the operation. The T bit is set if the comparison is true, otherwise it is cleared.

NOTE

Only one version of the immediate form is implemented. Non-equality comparisons to immediate values will require two instructions.

- CMPEQ r,s-This sets T when registers GReg[r] and GReg[s] are equal.
- CMPEQIr,immediate-This sets T when register GReg[r] and the zero-extended immediate value are equal.
- CMPLTr,s-This sets T when register GReg[r] is less than and not equal to GReg[s]. The comparison is signed.
- CMPHS r,s-This sets T when register GReg[r] is greater than or equal to GReg[s]. The comparison is signed.

7.2.3.11.10 Test Instructions

Test instructions modify the T bit in the processor status according to the result of the operation. The T bit is set if any bit in the result is one, otherwise it is cleared.

- TSTr,s-This performs an and between GReg[r] and GReg[s], and sets T if the result is not zero.
- TSTIr,immediate-This performs an and between GReg[r] and a zero-extended immediate value, and sets T if the result is not zero.

7.2.3.11.11 Byte Permutation Instructions

These instructions shuffle the bytes in a register. For the purpose of describing these instructions, have the bytes in a register be numbered from the most significant as b_3, b_2, b_1, b_0.

- RORBr-The rotate right byte. The result is b_0, b_3, b_2, b_1.
- REVBr-The reverse bytes in word. The result is b_0, b_1, b_2, b_3.
- REVBLOr-The reverse, two low-order bytes. The result is b_3, b_2, b_0, b_1.

7.2.3.11.12 Bit Shift Instructions

The following are bit shift instructions:

- ROR1r-The rotate right 1 bit. This instruction does a circular right shift of 1 bit.
- LSR1r-The logical shift right 1 bit. This instruction shifts all bits to the right by 1. The high order bit is replaced by a 0.
- ASR1r-The arithmetic shift right 1 bit. This instruction shifts all bits to the right by 1. The high order bit is replaced by itself.
- LSL1r-The logical shift left 1 bit. This instruction shifts all bits to the left by 1. The low order bit is replaced by zero.

7.2.3.11.13 Bit Manipulation Instructions

- BCLRIr,n-The bit clear is immediate; clears bit number *i* in register *r*.
- BSETIr,n-The bit set is immediate; sets bit number *i* in register *r*.
- BTSTIr,n-The bit test is immediate; tests bit number *i* in register *r* (T becomes equal to the selected register bit).

7.2.3.11.14 SDMA Memory Access Instructions

All memory accesses are 32 bits.

Any memory location that is implemented with less than 32 bits (for example, peripheral registers) causes unimplemented bits to be read as 0s.

All memory accesses will cause either the SF or DF flags in the processor status to be set if they cause a fault.

What constitutes a fault, especially when accessing peripheral registers, is a property of the memory location.

- LDr,(b,d)-The load instruction creates an address by adding the displacement field (d) to the contents of the base register (b). The SDMA location at the resulting address is read and placed in the destination register (r).
- STr,(b,d)-The store instruction creates an address in the same manner as the load instruction. The register (r) is stored in the SDMA location at the resulting address.

7.2.3.11.15 Functional Unit Instructions

The functional unit instructions have an 8-bit field that is placed on the functional unit bus.

Some of these bits are used to select which functional unit should be involved in the transfer. The remaining bits are decoded by the selected functional unit so their specific use depends on the functional unit. See Functional Units Programming Model.

There are two functional unit instructions, as follows:

- LDFr,fub-The 8-bit field is placed on the functional unit bus and a read is issued to the selected functional unit. As a result of this instruction, the SF may be set in the processor status.
- STFr,fub-The 8-bit field is placed on the functional unit bus and a write is issued to the selected functional unit. As a result of this instruction, the DF may be set in the processor status.

7.2.3.11.16 Illegal Instructions

All instruction encodings that are illegal cause the following actions:

- The current PC (which points to one beyond the offending instruction) is put in the EPC register.
- The loop mode bit is cleared.
- The PC is set to the value stored in the Illegal Instruction Trap Address (SDMAARM_ILLINSTADDR) register (the default value is 0x0001).

ILLEGAL-Although any instruction other than those indicated in the SDMA specification will trigger the illegal instruction mechanism, the ILLEGAL instruction code is preferred as it will always be kept as *illegal* in the possible future versions of the SDMA core.

7.2.3.11.17 Debug Instructions

The following are debug instructions:

- SOFTBKPT-The software breakpoint instruction causes the core to stop and enter debug mode. The core can then be accessed and started by the OnCE debug block only.
- done 5-This instruction is used for debugging, as it copies the contents of the PCU registers and flags to the context memory. Information on this instruction is described in Saving the Context.
- CpShReg-This instruction copies the context memory into the PCU registers and flags. Modifying the corresponding memory location before executing this instruction enables you to have the channel continue from a new instruction address. This instruction is described in Restoring the Context.

7.2.3.12 Functional Units Programming Model

The functional unit instructions cause an 8-bit code, found in the low eight bits of the instruction, to be asserted on the functional unit control bus.

Some of these bits are used to select one of several functional units. Functional units which can be selected include SDMA registers such as MSA and MSD which are not mapped in the SDMA memory map, and are accessible only through the functional unit bus. These Functional Unit Registers are listed in the following table. In order to establish a programming convention, assume the selection bits are some number of the most significant bits of the 8-bit code. Furthermore, some number of the least significant bits is decoded by a given functional unit to establish the type of operation to perform.

Table 7-20. Functional Unit Registers

Functional Unit	Register	Register Name	Section/Page
Burst DMA Unit Programming	SDMSA	Memory Source Address Register	Memory Source Address Register (MSA)
	MDA	Memory Destination Address Register	Memory Destination Address Register (MDA)
	MD	Memory Data Buffer Register	Memory Data Buffer Register (MD)

Table continues on the next page...

Table 7-20. Functional Unit Registers (continued)

Functional Unit	Register	Register Name	Section/Page
			(Write) Burst DMA Write (stf) (Read) Burst DMA Read (ldf)
	MS	Memory State Register	State Register (MS)
Peripheral DMA Unit Programming	PSA	Peripheral Source Address Register	Peripheral Source Address Register (PSA)
	PDA	Peripheral Destination Address Register	Peripheral Destination Address Register (PDA)
	PD	Peripheral Data Buffer Register	Peripheral Data Register (PD) (Write) Peripheral DMA Write (stf)-Write Mode (Read) Peripheral DMA Read (ldf)-Read Mode
	PS	Peripheral State Register	Peripheral State Register (PS)

More information regarding the functional units can be found in Peripheral DMA Unit, and Burst DMA Unit.

7.2.3.12.1 Burst DMA Unit Programming

The DMA instructions control the DMA state machine and may cause a DMA cycle on the associated memory bus.

There are four registers associated with the burst DMA unit: a Memory Source Address register (MSA), a Memory Destination Address register (MDA), a Memory Data buffer (MD), and a state register (MS). The burst DMA has two different uses:

- A data transfer between External Memory Interface and SDMA general register
- A data transfer in copy mode where blocks of data are transferred from the source address to the destination address

7.2.3.12.1.1 Memory Source Address Register (MSA)

The source address register contains the pointer into EXTMC memory associated with the next read data transfer. It has byte granularity.

Reading the register with the ldf instruction has no side effects, and gives the address value in the EXTMC memory of the next data that is read by the SDMA during an ldf MD instruction.

Writing the source address register has two side effects: If the prefetch bit is set, a DMA read cycle (8-word read access) is issued with the new address. Any data still located in the buffer is lost. If there is valid write data in the buffer, it is necessary to force the DMA to completely flush it out before modifying MSA to guarantee all the data is effectively written to memory.

The MSA register has two modes of programming:

- Frozen-In frozen mode, the MSA register is not modified after DMA accesses.
- Incremented (default mode)-In incremental mode, MSA is incremented by the number of bytes transferred during read cycles.

7.2.3.12.1.2 Memory Destination Address Register (MDA)

The destination address register contains the pointer into EXTMC memory associated with the next write data transfer. It has byte granularity.

Reading the MDA register with the ldf instruction has no side effects. It gives the address value in the EXTMC memory where the next SDMA data (stf r,MD instruction) is stored when MD FIFO is flushed.

Writing the destination address register has one side effect. Any data still located in the buffer is lost. If there is valid write data in the buffer, it is necessary to force the DMA to completely flush it out before modifying MDA to guarantee all the data is effectively written to memory.

The MDA register has two modes of programming:

- Frozen-In frozen mode, the MDA register is not modified after DMA accesses.
- Incremented (default mode)-The MDA register is incremented by the number of bytes transferred during write cycles.

7.2.3.12.1.3 Memory Data Buffer Register (MD)

The data buffer register consists of a bank of 36 bytes that behave like FIFO.

This FIFO stores the eight words received when a read burst is triggered by the DMA (DMA is in read mode).

The MD register is in write mode after a writing in MDA or after an stf MD instruction.

In that case, a burst write access is automatically triggered when there are more than eight words in MD. For bandwidth optimization, any transfers between DMA and the EXTMC controller are based on burst accesses.

An ldf r,MD|SIZE instruction that reads the data buffer may cause a DMA cycle, as follows:

- If there are less bytes in the FIFO than the size parameter of the instruction. For instance, if only two bytes are available in MD and a 4-byte read is requested, a burst read access is executed to complete the two bytes.
- If the prefetch bit is set, and after reading there is enough space in the FIFO to store a full burst, a burst read access is triggered.

An stf r,MD|SIZE instruction that writes to the data buffer may cause a DMA cycle if the number of written bytes in MD is higher than 32 (eight words) or if the flush bit is set.

When DMA is used for data transfer between SDMA and EXTMC (reading or writing), no immediate error is possible because the block manages a data misalignment issue; therefore, it is allowed to read/write a word to/from a half-word address. However, the addresses (source or destination) must belong to the EXTMC memory mapping. The only potential error, in this mode, would be the error sent back by the EXTMC controller when an access to a super-user page is detected. The whole transfer on the DMA associated bus will be considered successful when there are no errors seen on the bus during the transfer. In copy mode, an immediate error could be returned to SDMA as described in Burst DMA Unit Error Management.

7.2.3.12.1.4 State Register (MS)

The state register contains the DMA state-machine value. It can be accessed in case of an error received during a transfer. MS is also accessed to set-up the conditional yielding feature.

The initialization value of this register is 0 and it consists of the following:

Table 7-21. SDMA_MS Structure

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	spriv	stype	0	0	dpriv	dtype
W																
R	0	0	0	0	y	d	e		0	0	n					
W																

Table 7-22. SDMA_MS Field Descriptions

Field	Description
31-22	Reserved

Table continues on the next page...

Table 7-22. SDMA_MS Field Descriptions (continued)

Field	Description
21 spriv	The spriv value is ignored for this device. 0 = valid value 1 = Reserved
20 stype	Source Mode. Indicates if MSA has to be incremented (or not) during accesses. 0 Frozen-MSA is not modified. 1 Incremented-MSA is incremented by the number of transferred bytes during read access.
19-18	Reserved
17 dpriv	The dpriv value is ignored for this device. 0 = valid value 1 = Reserved
16 dtype	Destination Mode. Indicates if MDA has to be incremented (or not) during accesses. 0 Frozen-MDA is not modified. 1 Incremented-MDA is incremented by the number of transferred bytes during write access.
15-12	Reserved
11 y	Conditional Yielding selector. When selected, theyield/yieldge instructions will not switch channels if the Burst DMA is in Write Mode, and it has less than four bytes in its FIFO. This is aimed at reducing the number of inefficient FIFO flushes due to context switches. 0 Always yields 1 Yields conditionally (when there are less than four bytes in the FIFO in write mode)
10 d	Access Direction or DMA Mode. DMA is in write mode when data was written into MD by stf MD instructions, or if a previous DMA cycle on the external bus was a write access. Writing MDA or MSA changes the DMA mode to the respective value. DMA is in read mode when a previous DMA cycle was a read access, and DMA stays in read mode when data is read by SDMA with an ldf MD instruction. Reading MDA or MSA does not change the DMA mode. 0 Read Mode 1 Write Mode
9-8 e	Error. Indicates if the previous access was acknowledged with a bus error. 00 No error was received. 01 *reserved* 10 Error mode 11 error read burst
7-6	Reserved
5-0 n	Number of bytes in the MD FIFO.

7.2.3.12.1.5 Burst DMA Write (stf)

When received from a stf instruction, the function code bits are interpreted as follows, depending on the addressed register:

Table 7-23. STF Code Bits

Register	7	6	5	4	3	2	1	0
MSA	s		p	freeze	r			spriv
MDA								dpriv
MD			f	cpy			sz	
MS								

Table 7-24. STF Code Bit Field Descriptions

Field	Description
7-6 s	Functional Unit selector 00 for Burst DMA
5 p (MSA)	Prefetch Flag 0 No prefetch 1 Prefetch required from new MSA
5 f (MD)	Forced Flush Flag 0 Automatic flush 1 FIFO contents are flushed (including the new written data).
4 freeze (MSA/MDA)	Address Freeze Mode 0 Address is normally incremented. 1 Address is frozen.
4 cpy (MD)	Copy Mode selection 0 Write Mode 1 Copy Mode
3-2 r	Register selection 00 MSA 01 MDA 10 MD 11 MS
1-0 sz (MD/MS)	Transfer Size 00 size 0 (no data stored in the FIFO) 01 byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)
0 spriv (MSA)	The spriv value is ignored for this device. 0 = valid value 1 = Reserved
0 dpriv (MDA)	The dpriv value is ignored for this device. 0 = valid value 1 = Reserved

The possible write instructions are listed in the table below (unused bits should always be cleared).

Table 7-25. Burst DMA STF Instruction List

Binary	Assembly	Comments
00_0_0_00_00	stf r,MSA	Writes content of the SDMA general register (r) to the source address register. MSA is in incremented mode.
00_0_1_00_00	stf r,MSA|FR	Writes content of the SDMA general register (r) to the source address register. MSA is in frozen mode.
00_1_0_00_00	stf r,MSA|PF	Writes content of the SDMA general register (r) to the source address register, and starts a read burst access. MSA is in incremented mode.
00_1_1_00_00	stf r,MSA|PF|FR	Writes content of the SDMA general register (r) to the source address register, and starts a read burst access.
00_0_0_01_00	stf r,MDA	Writes content of the SDMA general register (r) to the destination address register. MDA is in incremented mode.
00_0_1_01_00	stf r,MDA|FR	Writes content of the SDMA general register (r) to the destination address register. MDA is in frozen mode.
00_1_0_10_00	stf r,MD|SZ0|FL	No data transfers between the SDMA and MD, but all valid written data of the MD is flushed to the memory. An acknowledge or error is sent back to the SDMA core on transfer completion.
00_0_0_10_01	stf r,MD|SZ8	8-bit (byte) transfer to write buffer MD
00_1_0_10_01	stf r,MD|SZ8|FL	8-bit (byte) transfer to write buffer MD and flush after transfer. All valid written data of the MD is flushed to memory.
00_0_0_10_10	stf r,MD|SZ16	16-bit (half-word) transfer to write buffer MD
00_1_0_10_10	stf r,MD|SZ16|FL	16-bit (half-word) transfer to write buffer MD and flush after transfer. All valid written data of the MD is flushed to memory.
00_0_0_10_11	stf r,MD|SZ32	32-bit (word) transfer to write buffer MD
00_1_0_10_11	stf r,MD|SZ32|FL	32-bit (word) transfer to write buffer MD and flush after transfer. All valid written data of MD is flushed to memory.
00_0_1_10_00	stf r,MD|CPY	No data transfer between SDMA and MD but starts a copy transfer whose length is given by the 4 LSB of *r* register. (Maximum burst length is eight words.)
00_0_0_11_11	stf r,MS	32-bit (word) transfer to status register MS
00_0_0_11_00	stf r,MS|SZ0	Clears the error flag (if set). Other MS bits are unchanged; this instruction is also known as clref MS.

NOTE

When a flush bit is set, the SDMA flushes the FIFO including the newly written data. An acknowledge is sent to the core before the flush completes (except if size 0 is used). The goal of this flush bit is to force a flush, but it is recommended to use it only when needed (for example, when finishing a row of pixels during 2D data transfers). Indeed, if this bit is omitted and if there are more than 32 bytes in the FIFO, a burst write access is automatically triggered.

Since all the stf r,MD instructions (including the copy mode) acknowledge the SDMA core before the store is effective (except if size 0 is used), it is recommended to perform an ldf from MS before terminating a channel in order to check the final error status. (The ldf from MS will stall the core until all the data was flushed out and the transfer status is known.)

After every stf MD instruction, the MDA is incremented by the number of bytes that are written in MD, except when it is programmed in frozen mode.

7.2.3.12.1.6 Burst DMA Read (ldf)

When received from an ldf instruction, the function code bits are interpreted as follows, depending on the addressed register:

Table 7-26. LDF Code Bits

Register	7	6	5	4	3	2	1	0
MSA	s				r			
MDA								
MD			p				sz	
MS								

Table 7-27. LDF Code Bit Field Descriptions

Field	Description
7-6 s	Functional Unit selector 00 for Burst DMA
5 p (MD)	Prefetch Flag 0 no prefetch 1 automatic prefetch
3-2 r	Register selection 00 MSA 01 MDA 10 MD 11 MS
1-0 sz (MD)	Transfer Size 00 reserved 01 byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)

The table below lists the possible write instructions (unused bits should always be cleared).

Table 7-28. Burst DMA LDF Instruction List

Binary	Assembly	Comments
00_0_0_00_00	ldf r,MSA	Copies the source address register value into an SDMA general register. It gives the memory address of the next data that will be read with an ldf MD instruction.
00_0_0_01_00	ldf r,MDA	Copies the destination address register value into an SDMA general register. It gives the memory address where the next incoming data will be flushed.
00_0_0_10_01	ldf r,MD|SZ8	8-bit (byte) read
00_1_0_10_01	ldf r,MD|SZ8|PF	8-bit (byte) read. If after this reading and the MD FIFO is empty, a burst read access at the MSA address is triggered.
00_0_0_10_10	ldf r,MD|SZ16	16-bit (half-word) read
00_1_0_10_10	ldf r,MD|SZ16|PF	16-bit (half-word) read. If after this reading, and the MD FIFO is empty, a burst read access at the MSA address is triggered.
00_0_0_10_11	ldf r,MD|SZ32	32-bit (word) read
00_1_0_10_11	ldf r,MD|SZ32|PF	32-bit (word) read. If after this reading and the MD FIFO is empty, a burst read access at the MSA address is triggered.
00_0_0_11_00	ldf r,MS	Copy the status register value into an SDMA general register.

NOTE

Read data is 0-extended before writing in the SDMA general registers. When reading the MD register, the DMA takes data from the FIFO if it is available. If part or whole data is not in the FIFO, an external burst read access is performed to provide the missing data. The SDMA is stalled as long as the required read data is not complete.

After every reading, MSA is incremented by the number of read bytes from MD FIFO, except when MSA is programmed in frozen mode.

7.2.3.12.1.7 Prefetch/Flush and Auto-Flush Management-Burst DMA Unit

The prefetch and auto-flush management enables the SDMA RISC machine to go on while a DMA access is performed.

When the RISC core requires a prefetch (p = 1) to the Burst DMA, it will receive an immediate transfer acknowledge before the DMA has finished the external access. This enables the RISC core to do other things like accessing another DMA machine.

The basic principle in prefetch mode is for the DMA to anticipate data reads from the SDMA RISC engine by fetching external bursts of data as soon as there is enough space in the DMA FIFO to store it. If ever the RISC engine required data that is not available in the FIFO, the read acknowledge is delayed until the data is available, but it does not have to wait until the burst completes.

The auto-flush basic principle is similar: An automatic flush is triggered every time there are eight words to be written in the FIFO. If the FIFO is full and the RISC engine requires another write, it is stalled until the burst has started and enough space was freed in the FIFO to store that new data. This means the SDMA RISC engine does not have to wait for the completion of a burst to receive its acknowledge and continue its processing.

In particular, an auto-flush is executed when DMA is in write mode and if the following is true:

- If the FIFO is empty and the first write is to a word-aligned address of any size (ex: the 2 LSB of MDA[1:0]= 0x0), the auto-flush is triggered immediately after the write of the 32'nd byte.
- If the FIFO is empty, and if MDA is an odd byte address (1, 3, 5, 7,...) and an stf MD|SZ8 is executed, the byte is flushed to memory. Once MDA increments to a word aligned address, the auto-flush will be triggered every 32 bytes.
- If the FIFO is empty, and if MDA is a half-word address (2, 6, 0xA,...) and an stf MD|SZ16 is executed, the two bytes of the incoming data are flushed to memory. Once MDA increments to a word aligned address, the auto-flush will be triggered every 32 bytes.
- If the FIFO is empty, and if MDA is not a word-aligned address (ex 1, 2, 3, 5, 6, 7, 9,...), and an stf MD|SZ32 is executed, the first 1 to 3 bytes will be flushed up to the next word aligned address. Afterwards, an auto-flush will be triggered each time the FIFO receives 32-bytes.
- Therefore, if an stf MD|SZ32 is executed with MDA equal to 0x1 and with an empty MD FIFO, the bytes located at addresses 1, 2, and 3 are flushed, and the byte located at address 4 remains in MD FIFO. This solves the misalignment issue. Additionally, the next write instructions (stf) complete the FIFO until it contains eight words; then a burst write is executed by the DMA to empty the FIFO. Protocol on the external bus does not support bursts of different data types (byte, half-word, or word).

  For example, consider the case where data is written using a byte access, stf MD|SZ8. The value of MDA during the very first byte write determines when the auto-flush will occur as follows:

- If MDA=0x0, the flush occurs following the write of byte 32
- If MDA=0x1, the flush occurs following the write of byte 1, byte 3 and byte 35.
- If MDA=0x2, the flush occurs following the write of byte 2 and byte 34.

- If MDA=0x3, the flush occurs following the write of byte 1 and byte 33.
- If MDA=0x4, the flush occurs following the write of byte 32

The flush command forces the DMA to flush all MD valid bytes to the EXTMC controller. An acknowledge is sent immediately to the SDMA, and any potential error is reported on a future access. It is thus essential to conclude a transfer with a last read from MS, which will stall the core until all data was flushed out and returned to the transfer status (acknowledge or error).

NOTE

During this kind of auto-flush (which occurs only at the beginning of a misaligned write transfer) no acknowledge is sent back to the SDMA, which is stalled until a flush is completed.

7.2.3.12.1.8 Data Alignment and Endianness-Burst DMA Unit

7.2.3.12.1.8.1 Burst DMA in Read Mode

For every read access to MD, the data returned to the SDMA core and the new FIFO state depends on the MSA status and the access size.

The FIFO is considered as a stack of 36 bytes: Data is fetched externally on a 32-bit bus, but the valid bytes only are stored in the FIFO and left-aligned (for a transfer of consecutive words, it is only the first word that may be truncated). The following table shows the FIFO byte alignment strategy and the corresponding MSA, the returned data, and the new FIFO state for any access size of an internal read from MD.

Table 7-29. FIFO Read Configuration

Before read		Internal read access size	Read data	After read	
MSA[1:0]	FIFO state			MSA[1:0]	FIFO state
00	x0 x1 x2 x3 y0 y1 y2 y3 z0 z1 z2 z3 and so on...	sz8	00 00 00 x0	01	x1 x2 x3 y0 y1 y2 y3 z0
		sz16	00 00 x0 x1	10	x2 x3 y0 y1 y2 y3 z0 z1
		sz32	x0 x1 x2 x3	00	y0 y1 y2 y3 z0 z1 z2 z3
01	x1 x2 x3 y0 y1 y2 y3 z0 z1 z2 z3 t0 and so on...	sz8	00 00 00 x1	10	x2 x3 y0 y1 y2 y3 z0 z1
		sz16	00 00 x1 x2	11	x3 y0 y1 y2 y3 z0 z1 z2
		sz32	x1 x2 x3 y0	01	y1 y2 y3 z0

Table continues on the next page...

Table 7-29. FIFO Read Configuration (continued)

Before read		Internal read access size	Read data	After read	
MSA[1:0]	FIFO state			MSA[1:0]	FIFO state
					z1 z2 z3 t0
10	x2 x3 y0 y1 y2 y3 z0 z1 z2 z3 t0 t1 and so on...	sz8	00 00 00 x2	11	x3 y0 y1 y2 y3 z0 z1 z2
		sz16	00 00 x2 x3	00	y0 y1 y2 y3 z0 z1 z2 z3
		sz32	x2 x3 y0 y1	10	y2 y3 z0 z1 z2 z3 t0 t1
11	x3 y0 y1 y2 y3 z0 z1 z2 z3 t0 t1 t2 and so on...	sz8	00 00 00 x3	00	y0 y1 y2 y3 z0 z1 z2 z3
		sz16	00 00 x3 y0	01	y1 y2 y3 z0 z1 z2 z3 t0
		sz32	x3 y0 y1 y2	11	y3 z0 z1 z2 z3 t0 t1 t2

7.2.3.12.1.8.2 Burst DMA in Write Mode

For every write access to the MD, the new FIFO state depends on the MDA status and the access size.

The FIFO is considered as a stack of 36 bytes: Data is stored in the FIFO according to the internal access size and the former MDA value. The following table shows the FIFO byte alignment strategy corresponding to MDA, as well as the new FIFO state for any access size of an internal write to MD.

Table 7-30. FIFO Write Configuration

Before write		Internal write access size	Written data	After write	
MDA[1:0]	FIFO state			MDA[1:0]	FIFO state
00	tt uu vv ww ?? ?? ?? ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	01	tt uu vv ww x0 ?? ?? ?? ?? ?? ?? ??
		sz16	?? ?? x0 x1	10	tt uu vv ww x0 x1 ?? ?? ?? ?? ?? ??
		sz32	x0 x1 x2 x3	00	tt uu vv ww x0 x1 x2 x3 ?? ?? ?? ??
01	tt uu vv ww	sz8	?? ?? ?? x0	10	tt uu vv ww

Table continues on the next page...

Table 7-30. FIFO Write Configuration (continued)

Before write		Internal write access size	Written data	After write	
MDA[1:0]	FIFO state			MDA[1:0]	FIFO state
	xx ?? ?? ?? ?? ?? ?? ?? and so on...				xx x0 ?? ?? ?? ?? ?? ??
		sz16	?? ?? x0 x1	11	tt uu vv ww xx x0 x1 ?? ?? ?? ?? ??
		sz32	x0 x1 x2 x3	01	tt uu vv ww xx x0 x1 x2 x3 ?? ?? ??
10	tt uu vv ww xx yy ?? ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	11	tt uu vv ww xx yy x0 ?? ?? ?? ?? ??
		sz16	?? ?? x0 x1	00	tt uu vv ww xx yy x0 x1 ?? ?? ?? ??
		sz32	x0 x1 x2 x3	10	tt uu vv ww xx yy x0 x1 x2 x3 ?? ??
11	tt uu vv ww xx yy zz ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	00	tt uu vv ww xx yy zz x0 ?? ?? ?? ??
		sz16	?? ?? x0 x1	01	tt uu vv ww xx yy zz x0 x1 ?? ?? ??
		sz32	x0 x1 x2 x3	11	tt uu vv ww xx yy zz x0 x1 x2 x3 ??

NOTE

If the FIFO mode changes from a write to a read mode, all remaining written bytes in MD are lost but no error is returned. Typically, this happens if an ldf MD is executed after stf MD instructions. Before a mode change, it is recommended to force the flush of a potential remaining byte by a stfMD|SZ0|FL instruction. In the same way, if a FIFO mode changes from a read to a write mode, all prefetched data present in the FIFO is lost and no error is returned.

7.2.3.12.1.8.3 Endianness-Burst DMA Unit

Big and Little Endian are supported by the Burst DMA, but data is always stored in MD in Big Endian.

Byte manipulation is performed when data is exchanged with an Burst controller (for example, during read or write burst accesses).

7.2.3.12.1.9 Burst DMA Unit Copy Mode

A mechanism is available to perform fast Arm-to-Arm transfers.

Data does not flow through the SDMA core: It is kept in the DMA FIFO. This mechanism is selected when writing MD with a special option in the instruction code (copy flag).

It is possible to transfer up to eight words in one SDMA instruction (this does not mean in one cycle). In this mode, every time an stf MD|CPY is executed, a read burst is executed and directly followed by a write burst transfer. Burst transfers are limited to eight words. The size of the transfer (in words)-given by the SDMA general register (4 LSB)-is also limited to eight. The following SDMA code shows how 100 bytes could be copied from the MSA address to the MDA address. This is sample code only.

Burst DMA copy mode example

```
          ldi r0,@src
          stf r0,MSA                          // Source address setup
          ldi r1,@dst
          stf r1,MSA                          // Destination address setup
          ldi r0,0x64                         // data transfer counter
          ldi r1,0x8
MAIN_XFER:
          cmphs r0,r1                         // Is r0 >= 0x8
          bf LAST_XFER                      // If not, jump to last transfer label
          stf r1,MD|CPY                        // Copy 8 words from MSA to MDA address.
          subi r0,0x8                         // Decrement counter
          jmp MAIN_XFER                     // return to main transfer loop
LAST_XFER:
          stf r0,MD|CPY
```

The main transfer loop is executed 12 times; then r0 equals 4 and the last transfer loop is run.

In this mode, an acknowledge is transmitted to the core as soon as the read burst can start; thus, a first copy instruction returns an immediate acknowledge and subsequent copy instructions will be acknowledged as soon as the previous copy has finished.

7.2.3.12.1.10 Burst DMA Unit Error Management

Another point to consider is the management of errors.

Because the DMA immediately sends an acknowledge to the RISC core (except for the stf MS|SZ0|FLS instruction), it assumes no error will occur. If an error occurs, it is flagged (transfer error acknowledge) for the following DMA access.

This should not be a problem if the DMA is used properly. The MD accesses are meant to stall the SDMA as little as possible to optimize throughput and hide calculation time. Therefore, final access to MS should be performed before closing a channel. This access waits until any pending operation is finished in the burst DMA and gather any remaining error.

In copy mode, an error could be immediately returned to the SDMA on execution of the ldf copy or stf copy instruction. It happens when MSA or MDA are not word addresses (for example, 0[4]). This is because copy mode must only be used for transferring a large packet of aligned data.

When an error is received during a *read* transfer to the external bus, which may occur during the burst accesses, the MD FIFO contains the valid beats of the burst, and the error flag of MS is set to 2'b11 (error read burst). It is possible to read MS ("n" field) to know how much valid data remains in MD and when MD is empty (after ldf instructions). The next read MD instruction sets the MS error flag to 2'b10 (error mode), and an error is sent back to the SDMA core. In error mode, it is possible to read MSA, which gives the address of the error data. Any attempt to read or write MD, or to modify MDA or MSA in error mode, gives rise to an error; therefore, an error flag must be reset by clearing MS at the end of the SDMA code section responsible for error management.

In "error read burst" mode, writing MDA, MSA, or MD, or starting a copy transfer by a stf MD|COPY instruction will cancel the error mode. The following table shows when an immediate error is sent back according to the executed instruction.

Table 7-31. Possibilities in ERROR READ BURST Mode

DMA Instruction	Immediate Error	Comments
stf rn, MD stf rn, MSA (|U |PF) stf rn, MDA stf rn,MD|COPY	NO	Error mode is reset. MSA, MDA, or MD are updated and a DMA cycle may start. For the stf MD|COPY, a copy loop is executed.
stf rn, MS	NO	MS is updated.

Table continues on the next page...

Table 7-31. Possibilities in ERROR READ BURST Mode (continued)

DMA Instruction	Immediate Error	Comments
ldf rn, MS ldf rn, MSA ldf rn, MDA	NO	MS, MSA, and MDA could be read in ERROR READ mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, MD	YES/NO	Immediate error if there is no more data available for read in the FIFO.

When an error is received during a *write* transfer, the error is reported to the next DMA access. In this case, an error is sent to the SDMA core and the DMA goes to its error mode. Reading MS gives the number of bytes that remain in MD; reading MDA gives the address of the error data. Any attempt to read or write MD, or to modify MDA or MSA in error mode, give rise to an error; therefore, an error flag must be reset by clearing MS at the end of the SDMA code section responsible for error management.

Table 7-32. Possibilities in ERROR Mode

DMA Instruction	Immediate Error	Comments
stf rn, MD stf rn, MSA stf rn, MDA	Yes	Any attempt to modify MD, MSA, MDA will raise an immediate error and burst DMA remains in error mode. When address registers are write-accessed, an error is returned.
stf rn, MS	No	This is the only way to exit error mode. MS[9:8] must be reset by an stf MS|SZ0 instruction.
ldf rn, MS ldf rn, MSA ldf rn, MDA	No	MS, MSA, and MDA could be read in error mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, MD	Yes	Whatever the DMA direction (read or write), an ldf rn triggers an immediate error.

7.2.3.12.1.11 Conditional Yielding-Burst DMA Unit

The standard SDMA transfer is based upon a hardware loop that has the following structure:

Hardware Loop

```
loop
load Rn,source            // can be ldf or ld
<computation>            // can be done through functional units
store Rn,dest              // can be st or stf
done 0               // yield
```

This structure needs to be kept independent of the functional units' particularities regarding the context switch. However, there can be variations in the context switch's efficiency, which can depend on the number of data received up to that point, and on the data itself.

The DMA, with its 8-word burst capability, has a preferable context switch period when its address register is 8-word aligned: It is the only moment that occurs once every eight loops when the succession of bursts is not broken by the context switch. When this is not the case, a context switch requires the storing (or loading) of less than eight words, which requires separate accesses and is far less efficient. The rest of the 8-word packet is stored (or loaded) after the context restore, and this is done as separate accesses.

The proposed solution is a conditional yielding, which occurs only when the DMA is in an optimum state. It does not require any modification to the scripts. The condition is decided at the DMA level.

The DMA can be programmed in two modes-conditional or always-true-for every channel, which provides complete flexibility. By default, the DMA is not in conditional mode.

The DMA condition is computed from the FIFO fill level and the various modes, as follows:

- When copy mode is selected, regardless of the transfer direction ('read' or 'write'), the condition is always true.
- In read mode, the condition is always true.
- In write mode, the condition is true when there are four bytes or less in the FIFO; it is false when there are more than four bytes. The 4-byte limit comes from the possibility of saving those bytes as MD with absolutely no impact on the bus accesses.

The aim at conditional yielding is to avoid splitting bus accesses (especially bursts).

7.2.3.12.2 Peripheral DMA Unit Programming

The peripheral DMA unit is connected to the Multi-Layer DMA Crossbar Switch of the Arm platform.

Its goal is to perform data transfers between any blocks connected to the DMA bus of this platform. These blocks are either peripherals or memories. The peripheral DMA could be seen as the Arm platform DMA controller.

The DMA performs data transfers in three modes:

- Read mode, where data is read from peripherals or from memory connected to the Arm platform and copied in a SDMA general register.

- Write mode, where data of a general register has to be written in a peripheral or a memory.
- Copy mode, where data is read from a peripheral (or memory) at a source address (PSA) and automatically written to a peripheral (or memory) at a destination address (PDA).

In copy mode, no SDMA general register is involved as transferred data only goes through the data register of the DMA.

The peripheral DMA has three addressing modes: frozen, incremented, and decremented, as follows:

- Frozen mode-When source or destination addresses are frozen, their value is not modified after a transfer. This mode is typically used for addressing peripheral FIFOs located at a fixed address.
- Incremented mode-When source or destination addresses are in incremented mode, after every transfer they are incremented by the number of bytes transferred.
- Decremented mode-In decremented mode, addresses are decremented by the number of bytes transferred.

The peripheral DMA registers are as follows:

- Two, 32-bit address registers (PSA and PDA) that respectively contain the source address for a read access and the destination address for a write access
- A 32-bit status register (PS) that contains information on the peripheral DMA configuration, such as the number of valid bytes in the data register, the error flag, the source and destination address mode, and so on.
- A 32-bit data register (PD) that stores data involved in a data transfer

7.2.3.12.2.1 Peripheral Source Address Register (PSA)

The source address register contains a pointer to a source peripheral or a memory associated with the next read data transfer. It has byte granularity.

It is based on the following:

- A 32-bit register (PSA) to store the address value
- A 2-bit register (stype) to store the source address mode (frozen, incremented, or decremented)
- A 2-bit register (ssize) to store the source target data path size (byte, half-word, or word)

Reading the register with the ldf instruction has no side effects and gives the address value of the next data that will be read by the SDMA during an ldf MD instruction. Writing the source address register may have side effects. If there is valid write data in the data register and the source address is changed, the write data is discarded. If the prefetch bit is set, a DMA read cycle is issued with the new address.

When PSA is to be written, you must specify the source target address mode, providing its size (byte, half-word, or word). This enables omission of the size field in all ldf MD instructions. When DMA performs a read cycle, its size is given by the value of the PSA source size register (ssize). If source is a memory in incremented mode, first programmed in word mode (stf PSA|SZ32|I), and if an SDMA script needs to read bytes from this memory, the size of the source target must be updated before executing new accesses. The source address mode and its size are given by labels added to the stf PSA instruction as described in the write section. The ssize and stype registers are part of the DMA status register (PS).

Writing to PSA may issue an immediate error if the source size is not compatible with the value to be written into the PSA register. For instance, writing a 2 in PSA and specifying that it is memory-accessed in word mode creates an immediate error.

7.2.3.12.2.2 Peripheral Destination Address Register (PDA)

The destination address register contains a pointer to a source peripheral or a memory associated with the next write data transfer. It has byte granularity.

It is based on the following:

- A 32-bit register (PDA) to store the address value
- A 2-bit register (dtype) to store the destination address mode (frozen, incremented, or decremented)
- A 2-bit register (dsize) to store the destination target data path size (byte, half-word, or word)

Reading the register with the ldf instruction has no side effects, and gives the address value of the next data that will be written by SDMA during an stfMD instruction. Writing the destination register has no side effect. Similar to the PSA register, the destination address mode and source are specified in the stf PDA instruction and may also generate an error in case of incorrect programming.

7.2.3.12.2.3 Peripheral Data Register (PD)

The data register of the peripheral DMA is a 32-bit register. When the destination address is correctly set up, any writing to PD will automatically flush the new input data.

The number of SDMA bytes that will be transferred is given by the PDA size register. Unlike other SDMA DMAs, PD is not a FIFO: It is not used to accumulate bytes that from the SDMA and must be packed before being sent to external memories. In read mode, and if the source address is correctly set up, an ldf instruction will empty PD. If a prefetch is required along with the instruction, the DMA will initiate a new read transfer.

Reading PD in prefetch mode only stalls the SDMA when the prefetched data is not yet available. Writing PD only stalls the SDMA if the previous write operation was not completed. As soon as the previous operation is over, the acknowledge is sent back to the SDMA RISC engine.

An error flag-part of PS-is set when an external access fails. The error is thus reported to the next SDMA instruction that involves the peripheral DMA.

7.2.3.12.2.4 Peripheral State Register (PS)

The state register contains the DMA state-machine value. It can be accessed in case of an error received during a transfer.

Although all PS fields can be written by an stf instruction, it is recommended to access only the error bit (to reset it). Modifying other PS fields will provide an un-guaranteed DMA behavior.

The initialization value of PS is 0, and it consists of the following structure:

Table 7-33. PS Structure

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	ssize		stype		dsize		dtype	
W																
R	0	0	0	0	0	d	e		0	0	0	0	0	n		
W																

Table 7-34. PS Field Descriptions

Field	Description
31-24	Reserved
23-22 ssize	Source Target Size. Determines the size of the read transfers on the external bus. It should match the accessed device characteristics. 00 *reserved* 01 Byte (8 bits) 10 half-word (16 bits)

Table continues on the next page...

Table 7-34. PS Field Descriptions (continued)

Field	Description
	11 word (32 bits)
21-20 stype	Source address Mode. Determines whether PSA is incremented, decremented, or kept unmodified after every read from the external bus. 00 Frozen Mode 01 Incremented Mode 10 Decremented Mode 11 *reserved*
19-18 dsize	Destination Target Size. Determines the size of the write transfers on the external bus. It should match the accessed device characteristics. 00 *reserved* 01 Byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)
17-16 dtype	Destination address Mode. Determines whether PDA is incremented, decremented, or kept unmodified after every write on the external bus. 00 Frozen Mode 01 Incremented Mode 10 Decremented Mode 11 *reserved*
15-11	Reserved
10 d	Direction Flag or DMA Mode. DMA is in write mode when data was written into PD by stf PD instructions, or if a previous DMA cycle on the external bus was a write access. Writing PDA or PSA does not change the DMA mode. DMA is in read mode when a previous DMA cycle was a read access, and DMA stays in read mode when data is read by the SDMA with an ldf PD instruction. Reading PDA or PSA does not change the DMA mode. 0 Read Mode 1 Write Mode
9-8 e	Error. Indicates if the previous access was acknowledged with a bus error. 00 No error was received. 01 *reserved* 10 Error mode 11 Error read
7-3	Reserved
2-0 n	number of bytes in PD

NOTE

dtype, dsize, stype, and ssize are updated when PSA and PDA are written.

7.2.3.12.2.5 Peripheral DMA Write (stf)-Write Mode

When written by an stf instruction, the function code bits are interpreted as follows:

Table 7-35. STF Code Bits

Register	7	6	5	4	3	2	1	0
PSA	s		p	ar	am		sz	
PDA								
PD			pdsel					
PS			pssel					

Table 7-36. STF Code Bits Field Descriptions

Field	Description
7-6 s	Functional Unit selector 11 for Peripheral DMA
5 p (PSA)	Prefetch Flag 0 no prefetch 1 automatic prefetch
4 ar (PSA/PDA)	Address Register Selector 0 PSA 1 PDA
3-2 am (PSA/PDA)	Address Mode. Determines how PSA or PDA is modified after every read or write access to the PD. 00 Frozen-Address registers are not modified after the transfer. 01 Incremented-Address registers are incremented by the number of transferred bytes. 10 Decremented-Address registers are decremented by the number of transferred bytes. 11 Updated-PSA and PDA are not modified. Either address mode is not modified, but the width of the data path is updated by the sz field.
1-0 sz	Transfer Size 00 *reserved* 01 byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)
5-0 pdsel	PD access selector 001000 is the only valid option
5-0 pssel	PS access selector 111111 writes to PS 001100 only clears the error flag in PS

Due to the large number of possible stf instructions, the following table provides only a short list of all the possible write instructions:

Table 7-37. Peripheral DMA STF Instruction List

Binary	Assembly	Comments
11_00_00_01 11_00_00_10 11_00_00_11	stf Rn, PSA|SZ8 |F stf Rn, PSA|SZ16|F stf Rn, PSA|SZ32|F	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is frozen.
11_10_00_01 11_10_00_10 11_10_00_11	stf Rn,PSA|SZ8 |F|PF stf Rn,PSA |SZ16|F|PF stf Rn,PSA |SZ32|F|PF	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • 1, 2, or 4 bytes are *fetched* from the peripheral source. • Source address is frozen.
11_00_01_01 11_00_01_10 11_00_01_11	stf Rn, PSA|SZ8 |I stf Rn, PSA|SZ16|I stf Rn, PSA|SZ32|I	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA + 1,2 or 4 after read PD.
11_10_01_01 11_10_01_10 11_10_01_11	stf Rn, PSA|SZ8 |I|PF stf Rn, PSA|SZ16|I|PF stf Rn, PSA|SZ32|I|PF	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA + 1, 2, or 4 after read PD. • 1, 2, or 4 bytes are *fetched* from the peripheral source.
11_00_10_01 11_00_10_10 11_00_10_11	stf Rn, PSA|SZ8 |D stf Rn, PSA|SZ16|D stf Rn, PSA|SZ32|D	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA-1,2, or 4 after read PD.
11_10_10_01 11_10_10_10 11_10_10_11	stf Rn, PSA|SZ8 |D|PF stf Rn, PSA|SZ16|D|PF stf Rn, PSA|SZ32|D|PF	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA-1,2, or 4 after read PD. • 1, 2, or 4 bytes are *fetched* from the peripheral source.
11_00_11_01 11_00_11_10 11_00_11_11	stf Rn, PSA|SZ8 |U stf Rn, PSA|SZ16 |U stf Rn, PSA|SZ32 |U	• *Update* source pointer to memory, which becomes a pointer to a memory accessed in byte, half-word, or word. • PSA value is not modified by Rn. • Bytes present in PD are lost.
11_10_11_01 11_10_11_10 11_10_11_11	stf Rn, PSA|SZ8 |PF|U stf Rn, PSA|SZ16 |PF|U stf Rn, PSA|SZ32 |PF|U	• *Update* source pointer, which becomes a pointer to a target accessed in byte, half-word, or word. • PSA value is not modified by Rn. • Bytes present in PD are lost. • 1, 2, or 4 bytes are *fetched* from the memory source.
11_01_00_01 11_01_00_10 11_01_00_11	stf Rn, PDA|SZ8 |F stf Rn, PDA|SZ16|F stf Rn, PDA|SZ32|F	• Destination is a byte, half-word, or word target at the Rn address, and any further PD write instructions will trigger byte, half-word, or word access to the destination. • Destination address is frozen.

Table continues on the next page...

Table 7-37. Peripheral DMA STF Instruction List (continued)

Binary	Assembly	Comments
11_01_01_01 11_01_01_10 11_01_01_11	stf Rn, PDA|SZ8 |I stf Rn, PDA|SZ16|I stf Rn, PDA|SZ32|I	• Destination is a byte, half-word, or word target at the Rn address, and any further PD write instructions will trigger byte, half-word, or word access to the destination. • Destination address is in incremented mode: PDA = PDA + 1, 2, or 4 after write PD.
11_01_10_01 11_01_10_10 11_01_10_11	stf Rn, PDA|SZ8 |D stf Rn, PDA|SZ16|D stf Rn, PDA|SZ32|D	• Destination is a byte, half-word, or word target at the Rn address, and any further PD write instructions will trigger byte, half-word, or word access to the destination. • Destination address is in incremented mode: PDA = PDA-1, 2, or 4 after write PD.
11_01_11_01 11_01_11_10 11_01_11_11	stf Rn, PDA|SZ8 |U stf Rn, PDA|SZ16 |U stf Rn, PDA|SZ32 |U	• *Update* destination pointer to memory, which becomes a pointer to a memory accessed in byte, half-word, or word. • PDA value is not modified by Rn • bytes present in PD are lost
11_00_10_00	stf Rn, PD	• Write "dsize" bytes of Rn in PD and automatically flush to destination target
11_11_11_11	stf Rn, PS	• Write status register
11_00_11_00	stf Rn,clrefPS	• Clear error flag if set

NOTE

When writing PD, size information is not important: It is embedded in the dsize field of PDA register. If dsize is 1, 2, or 4, then one, two, or four bytes from Rn is written to the PD register, and automatically flushed out to the destination target.

7.2.3.12.2.6 Peripheral DMA Read (ldf)-Read Mode

When received from an ldf instruction, the function code bits are interpreted as follows.

Table 7-38. LDF Code Bits

Register	7	6	5	4	3	2	1	0
PSA	s			ar	a			
PDA								
PD			p	cpy				
PS			pssel					

Table 7-39. LDF Code Bits Descriptions

Field	Description
7-6	Functional Unit selector
s	11 for Peripheral DMA

Table continues on the next page...

Table 7-39. LDF Code Bits Descriptions (continued)

Field	Description
5 p (PD)	Prefetch Flag 0 no prefetch 1 automatic prefetch
4 ar (PSA/PDA)	Address Register Selector 0 PSA 1 PDA
4 cpy (PD)	Copy Mode 0 standard access 1 copy mode access
3 a	Register Set selection 0 PSA or PDA 1 PD or PS
5-0 pssel	PS access selector 111111 is the only valid option to read PS

Table 7-40. Peripheral DMA LDF Instruction List

Binary	Assembly	Comments
11_0_0_0_000	ldf Rn, PSA	Reads 32-bit of PSA value
11_0_1_0_000	ldf Rn, PDA	Reads 32-bit of PDA value
11_0_0_1_000	ldf Rn, PD	Reads programmed source size bytes of PD (0-extended)
11_1_0_1_000	ldf Rn, PD|PF	Reads programmed source size bytes of PD (0-extended), and starts a prefetch at PSA address.
11_0_1_1_000	ldf Rn, PD|COPY	Starts a copy transfer from the source target at the PSA address to the destination target at the PDA address. No data transmits through Rn, but Rn contents are lost (Rn is loaded with PD temporary contents that are *not* the copied data).
11_111111	ldf Rn, PS	Reads 32-bit of PS value

NOTE

When reading PD, size information is not important: It is embedded in the ssize field of the PSA register. If ssize is 1, 2, or 4, the one, two, or four bytes is transferred from PD to Rn. Read data is 0-extended.

7.2.3.12.2.7 Peripheral DMA Unit Copy Mode

Like burst DMA, the peripheral DMA unit has a copy mode that is used when data transfers do not involve SDMA general registers.

Data is read from the source target at a PSA address, stored in PD, and then automatically flushed to the destination target at the PDA address. Copy mode is only available for transfers that involve two targets of the same data path width.

Since copy mode is invoked with an ldf instruction, the *loaded* general purpose register loses its previous contents. (However, the new contents are unpredictable as they depend on temporary values that are seen on the external DMA bus.)

7.2.3.12.2.8 Error Management

Peripheral DMA generates two kinds of errors: the immediate error that sanctioned incorrect register programming; and the error triggered by the previous access and stored in the error flag of PS until a DMA instruction is executed.

7.2.3.12.2.8.1 Immediate Errors

The following table lists all incorrect DMA register setups.

Table 7-41. Immediate Errors with Peripheral DMA

Rn[1:0] values	DMA instruction	Comments
0x01 0x11	stf Rn, PSA|SZ16|F stf Rn, PSA|SZ16|I stf Rn, PDA|SZ16|F stf Rn, PDA|SZ16|I	If PSA points to a half-word peripheral or to a half-word address in memory, its value must be 0 modulo 2.
0x01 0x10 0x11	stf Rn, PSA|SZ32|F stf Rn, PSA|SZ32|I stf Rn, PDA|SZ32|F stf Rn, PDA|SZ32|I	If PSA points to a word peripheral or to a word address in memory, its value must be 0 modulo 4.
PSA[1:0]-PDA[1:0]	DMA instruction	Comments
0x01 0x10 0x11	stf Rn, PSA|SZ32|U stf Rn, PDA|SZ32|U	When PDA or PSA is updated and becomes a pointer to a word address in memory, its content must be 0 modulo 4.
0x01 0x11	stf Rn, PSA|SZ16|U stf Rn, PDA|SZ16|U	When PDA or PSA is updated and becomes a pointer to a half-word address in memory, its content must be 0 modulo 2.
Read/Write PD instruction	Comments	
stf Rn,PD ldf Rn,PD	If PDA size (dsize) has never been set up before an stf PD instruction (dsize=0) If PSA size (ssize) has never been set up before an ldf PD instruction (ssize=0)	
ldf Rn,PD|CPY	Copy mode is possible only between two targets whose data path width is identical. It is P8↔P8, P16↔P16, or P32↔P32 regardless of the way the address registers are incremented.	

7.2.3.12.2.8.2 Data Transfer Errors

When PSA and PDA are correctly set up, the only error that may arise for an ldf PD or stf PD instruction would be the error of the previous DMA cycle.

Error handling is driven by a single consideration: When an error occurred during a data read on the DMA interface, this error should appear as a transfer error to the core when the core attempts to retrieve the data that was not successfully read from the accessed device (memory or peripheral).

When an error occurred during a write access to the DMA interface, the data is still available in PD and should not be destroyed by subsequent core accesses: The core must be warned about the error issue.

There are three error handling mechanisms for each case: Read Error (First Phase), Write Error and Read Error (Second Phase), and Copy Mode Errors handling.

7.2.3.12.2.8.3 Read Error (First Phase)

If an error occurred during a prefetch command, the peripheral DMA enters its ERROR READ mode (PS[9:8]=11). In this mode, the error is reported on the next ldf PD instruction and writing PSA, PDA, or PD will cancel the error flag.

The block returns no error mode and instructions are normally executed (a DMA cycle may be triggered). Similarly, initiating a copy transfer will reset the error flag and start a copy transfer. The following table details which instructions can be executed in this mode.

Table 7-42. Possibilities in ERROR READ Mode

DMA Instruction	Immediate Error	Comments
stf rn, PD stf rn, PSA (IU IPF) stf rn, PDA ldf rn,PD|COPY	NO	Error mode is reset, PSA or PDA are updated, or a write cycle is started. For the ldf PD|COPY, a copy loop is executed.
stf rn, PS	NO	PS is updated.
ldf rn, PS ldf rn, PSA ldf rn, PDA	NO	PS, PSA, and PDA could be read in ERROR READ mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, PD	YES	Error of the previous read access is reported here and the peripheral DMA enters its ERROR mode.

7.2.3.12.2.8.4 Write Error and Read Error (Second Phase)

The peripheral DMA enters its ERROR mode (PS[9:8]=10) when the previous DMA write cycle failed, or, as explained in Read Error (First Phase), when an ldf PD is executed while the block is in ERROR READ mode. When a DMA cycle failed, address registers (PSA, PDA) are not modified and continue to point to the problematic address. In ERROR mode, stf instructions may raise an immediate error, and ldf instructions will not (as detailed in the table below).

Table 7-43. Possibilities in ERROR Mode

DMA Instruction	Immediate Error	Comments
stf rn, PD stf rn, PSA stf rn, PDA	YES	Any attempt to modify PD, PSA, or PDA will raise an immediate error, and the peripheral DMA stays in ERROR mode. When address registers are write accessed, an error is returned.
stf rn, PS	NO	This is the only way to exit the ERROR mode. PS[3] must be reset by an stf PS instruction.
ldf rn, PS ldf rn, PSA ldf rn, PDA	NO	PS, PSA, and PDA could be read in ERROR mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, PD	YES	Whatever the DMA direction (read or write), an ldf rn, PD instruction will show an immediate error.

7.2.3.12.2.8.5 Copy Mode Errors

Because copy mode is a write access that follows a read access, there are two possible cases of bus error.

When the read access incurs a bus error, the peripheral DMA behaves exactly as described in Read Error (First Phase) and Write Error and Read Error (Second Phase) : It enters its ERROR READ mode, and so on.

When the error occurred during the write access of the copy transfer, the DMA enables the core to retrieve the data that was read because it is assumed the read from the peripheral removed the data from its source device. Therefore, the data to be flushed is still in PD. Any subsequent access to PD triggers an error to the core, which should execute its error handling procedure.

Once the ERROR mode is left (after writing to PS), it is possible for the core to retrieve the data in PD with an ldf instruction or try to flush PD contents once again (for example, when the error was due to a full FIFO and the script waited for the FIFO to be emptied) with another ldf instruction in copy mode. This latter instruction detects that there is valid data in PD, tries to flush it, and thus skips the read phase of the copy instruction. This is a different behavior from the usual stf PD instruction that overwrites PD with the selected

General Purpose register contents. The same mechanism can be used any time PD holds data that is not written because of a bus error on the DMA interface; when the data was written via a copy instruction, or via the usual stf PD instruction.

7.2.3.12.2.8.6 Error Check Example

The following code illustrates an example checking for both immediate and data transfer errors on a store to the PD register. The first bdf instruction checks for an immediate error, but if a data transfer error occurred it is reported until the next instruction to access the Peripheral DMA. A second check of the error flags is done after the ldf PS instruction. The value of PS here can be ignored. The act of reading any register in Peripheral DMA while it is in an error mode that returns the error to the core to set either the SF or DF flag. Any error returned on an ldf command sets the SF flag and any error returned on an stf instruction sets the DF flag. This can create a situation as shown in the example where a bus error during a DMA write which would normally be considered as a destination fault is reported as a source fault because the error was reported to the SDMA core during an ldf instruction.

Table 7-44. Peripheral DMA Error Check

Function	Instruction	Comment
clrf	0	Clear SF and DF flags
stf	R4, PD	Write data to memory
bdf	error_routine	Check for immediate error from write to PD.
ldf	r3, PS	Read PS (PS value in R3 can be ignored)
bsf	error_routine	Check for bus error from "stf R4,PD". SF is set because it is a ldf instruction, even though the original error was a destination fault.

7.2.3.12.2.9 Peripheral DMA Unit Prefetch/Flush Management

There is no flush bit because every time data is stored in PD by a stf PD instruction-assuming PDA is correctly programmed-it is automatically flushed to the destination.

An acknowledge is returned in the cycle of the DMA instruction, and the SDMA is only stalled by an instruction that addresses the peripheral DMA when the previous DMA access is not over.

7.2.3.12.3 OnCE and Real-Time Debug

The On-Chip Emulation block (OnCE) is the debug interface to the SDMA.

It supports the access to all core internal devices (registers, memory, and so on), and provides a set of mechanisms that control the core. The OnCE is accessed by JTAG ports at the chip's board level, or by the host via its peripheral bus.

To reduce the size of the hardware material involved, all tasks supported by the OnCE are performed on the SDMA core. The architecture of the SDMA OnCE is relatively simple and very flexible.

The commands supported by the SDMA OnCE are listed in the following sections.

7.2.3.12.3.1 Memory and Register Access

A set of mechanisms is provided to access SDMA memory and register locations. Both reading and writing are allowed. The access is supported if the processor is in debug mode.

Those registers can also be accessed through the Arm platform Control interface when the OnCE is controlled by the Arm platform, as described in the "Using BP" section.

7.2.3.12.3.2 Hardware Breakpoints

An event detection unit is implemented to support memory breakpoints. The unit watches the data exchanged between the SDMA memory bus and the core.

A debug request is sent to the core when matching conditions occur. The unit supports mixed conditions based on address range, access type, and data value. Event detection unit configuration registers are memory mapped in the SDMA space (see Arm platform Channel 0 Pointer (SDMAARM_MC0PTR)): You can modify them through a regular memory access or the Arm platform control interface.

7.2.3.12.3.3 Watchpoints

One output pin is provided to monitor matching trigger conditions that are defined in the event detection unit.

7.2.3.12.3.4 Software Breakpoints

The SDMA instruction set contains a software breakpoint. Upon executing a software breakpoint instruction, the core suspends normal execution and enters debug mode.

No hardware step execution mode is implemented in the OnCE, but this feature may be implemented at the software level with this instruction.

7.2.3.12.3.5 Core Control

Commands are provided to monitor and control processor activity. You can halt the core, rerun the core from another address location, and get processor status.

Any hardware breakpoint on the instruction bus is not supported, but this feature may be implemented by inserting a software breakpoints program.

7.2.3.13 The OnCE Controller

The OnCE controller receives commands from the Arm platform or from the JTAG controller. Each command is interpreted before being sent to the core.

7.2.3.13.1 OnCE Commands

A small set of commands supports the communication between the OnCE and the external world.

This command set enables you to perform any of the following tasks: control processor activity, save core context, and execute an SDMA instruction from the OnCE. Combined together, these tasks perform more complex commands.

A full OnCE command contains a 4-bit instruction (the OnCE command opcode) and a variable length data field (the OnCE data). During command execution, the OnCE data is transferred in a OnCE internal register before being exchanged with the SDMA. Some data values are also exported. This mechanism creates a link between the processor and the external world. Nine commands are defined: The following table presents their formats.

Table 7-45. OnCE Command Opcode Values

Instruction Opcode	Name	Action	Register	Data Field Size	Mode
0000	rstatus	Reads the OnCE status register	STATUS	16-bit	normal/debug
0001	dmov	Updates general register GReg1	GREG1	32-bit	debug

Table continues on the next page...

Table 7-45. OnCE Command Opcode Values (continued)

Instruction Opcode	Name	Action	Register	Data Field Size	Mode
0010	exec_once	Runs the instruction from the SDMA instruction register	INSTRUCTION	16-bit	debug
0011	run_core	Returns to normal execution	BYPASS	1-bit	debug
0100	exec_core	Returns to normal execution via a jump instruction that specifies the new address	INSTRUCTION	16-bit	debug
0101	debug_rqst	Stops the core after execution of current instruction	BYPASS	1-bit	normal
0110	rbuffer	Reads the real time buffer	RTB	32-bit	normal/debug
0111-1110	reserved	Reserved	BYPASS	1-bit	normal/debug
1111	bypass	Bypasses TARM platform controller	BYPASS	1-bit	normal/debug

Each instruction corresponds to a specific action performed on the OnCE. The nature of the associated data field is clearly identified. The dmov command is followed by a 32-bit data value (which is a data value for the SDMA); the exec_once and the exec_core commands are followed by a 16-bit data value (which is an instruction for the SDMA); the rstatus command is followed by a 16-bit control value (which is the content of the OnCE status register); the rbuffer command is followed by a 32-bit data value. The debug_rqst and the run_core commands are followed by a single bit data field (this is a bypass value). Finally, the bypass instruction enables the SDMA JTAG TAP controller to be daisy-chained with another JTAG TAP controller. This is a JTAG-only feature. The set of commands is simple, but enables you to perform any possible task on the SDMA during a debug process.

7.2.3.13.2 Sending Commands to the OnCE Controller

The JTAG access is the standard access to the OnCE, but sometimes the JTAG is not available to fix some bugs (if the chip is in production for instance), an additional access is then required. Therefore, one Arm platform access to the OnCE is provided.

7.2.3.13.2.1 Using the JTAG Interface

A serial access is performed through the five JTAG pins TCK, TRST, TMS, TDI, and TDO. A Test Access Port controller is provided to decode the TMS control signal.

It produces shift-enable signals (shift_ir and shift_dr), and updates enable signals (update_ir and update_dr). It is fully compliant with the IEEE 1149.1 testability (JTAG) standard.

During the shift_ir state, the command opcode is shifted into the OnCE controller (for example, the signal from the TDI pin is shifted into the command register and the TDO pin receives the signal shifted out). After transferring the four bits of the command, an update_ir signal is asserted and the command is decoded. The target data register is now clearly identified and the corresponding control signal is produced, as follows: bypass enable signal (bp_en), instruction enable signal (inst_en), data enable (data_en), and status enable signal (stat_en).

During the shift_dr state, the TDI signal is shifted into one of the following target registers: bypass register (1 bit), SDMA instruction register (16 bits), SDMA data register (32 bits), or OnCE status register (16 bits). The TDO pin is connected to the output of the selected register to receive the signals shifted out.

The JTAG access is disabled when the Arm platform access is enabled.

7.2.3.13.2.2 Using the Arm platform

The Arm platform access to the OnCE is not the standard access, but it is required if the JTAG is not available.

For example, if the SDMA ROM is out of use on a chip in production, and the Arm platform needs to download new code and restart the SDMA, the OnCE can easily perform this operation. This type of debug operation justifies the use of an Arm platform access to the OnCE.

To drive the OnCE, the Arm platform uses some registers contained in the Arm platform Control block of the SDMA. These registers are accessed through the Arm platform peripheral bus. Most of these registers are connected to another register in the OnCE controller. Thus, accessing one of these registers is equivalent to accessing the associated register in the OnCE controller.

The set of registers in the Arm platform Control block is listed below:

- ONCE_ENB register (1 bit, read/write)-This 1-bit register enables the Arm platform access to the OnCE. When this bit is set, the signals from the JTAG are ignored. When it is cleared, all writing operations to the following registers through the Host Control interface are ignored. This register is reset on a JTAG reset.
- ONCE_CMD register (4 bits, read/write)-This 4-bit register receives the command opcode. It is connected to the command register in the controller. A write access to this register causes the associated command to be executed on the OnCE. For example, after writing "0001" in this register, a dmov command is executed.

NOTE

On the Arm platform side, the rstatus and bypass commands are not supported. This register is reset on a JTAG reset.

- ONCE_DATA register (32 bits, read/write)-This 32-bit register is connected to the SDMA data register. This register is used when executing a dmov or rbuffer command.

NOTE

Before requesting a dmov command, the 32-bit data to transfer must be written in the ONCE_DATA register. At the end of the execution, the register is updated with GReg1 former value. This register is reset on a JTAG reset.

- ONCE_INSTR register (16 bits, read/write)-This 16-bit register is connected to the SDMA instruction register. This register is used when executing an exec_core or an exec_once command.

NOTE

Before requesting an exec_core or an exec_once command, the appropriate instruction must be written in the ONCE_INSTR register. This register is reset on a JTAG reset.

- ONCE_STAT register (16 bits, read only)-A read access to the ONCE_STAT register returns the content of the OnCE status register (OSTAT). This register is read only.
- The bypass register is not useful when the Arm platform controls the OnCE, therefore no register is defined in the Arm platform Control block to access the bypass register.

7.2.3.13.2.3 Conflicts Between the JTAG and the Arm platform Accesses

When Arm platform access to the SDMA OnCE is enabled (that is, when the bit in the ONCE_ENB register is set), the JTAG access is disabled. This guarantees that the block is not accessed at the same time on both sides.

It is possible to check whether the JTAG access to the SDMA OnCE is enabled from the JTAG port. When the JTAG access is disabled, the SDMA TDO always returns 1. The check requires the following steps:

- Execute a dmov command from debug mode (with neither 0xffffffff nor 0x0 as dmov value: 0x5a5a5a5a is good).

- Execute another dmov command (the value here is not important).
- The returned value from the latter dmov command should be the original one if the JTAG access is enabled; if it is 0xffffffff instead of the original input value, this means the JTAG access is disabled.

7.2.3.13.3 Executing a Command from the OnCE

All the commands defined in OnCE Commands can be accessed through the JTAG. The Arm platform can access all these commands except the rstatus command.

On the Arm platform side, the OnCE status is directly accessed by reading the ONCE_STAT register.

7.2.3.13.3.1 Nature of the Commands

Two types of commands may be distinguished. First, there are two commands that do not interact with the core: rstatus and rbuffer. Those commands may be requested at any time: They do not depend on the core status.

NOTE

Each of these commands exports a data value or a status value from the SDMA.

There are also commands that interact with the core: dmov, run_core, exec_core, exec_once, and debug_rqst. These commands are core status dependent, as follows:

- During user mode only the debug_rqst is taken into account.
- During debug mode, all these commands are taken into account except the debug_rqst. For example, an exec_once command requested while not in debug mode has no effect.

7.2.3.13.3.2 Execution Request

The SDMA starts executing a task in debug mode when requested by the OnCE controller. The execution starting time depends on the type of access used to communicate with the OnCE.

If the JTAG is used, the request is send after decoding the update_dr state in the TAP controller. Therefore, always cross this state when sending a command through the JTAG. If the OnCE is driven from the Arm platform side, the request is sent after detecting a write access to the ONCE_CMD register. All the registers involved in this operation must be loaded first.

The following is an example of an exec_core command execution from the Arm platform side: After writing '010' in the ONCE_CMD register, the OnCE controller asks the SDMA to execute the instruction contained in the ONCE_INSTR register. The instruction involved should be available in the ONCE_INSTR register before the beginning of the execution.

7.2.3.13.3.3 Command Execution

The following list shows the commands and details how each command is executed:

- rstatus command execution-The rstatus command exports the content of the OnCE status register (OSR). If the JTAG is used, the status information is captured in the OnCE status register during the capture_dr state, and shifted out after 16 TCK clock cycles in the shift_dr state. The rstatus command is not supported on the Arm platform side, but a status register is provided instead. The rstatus may be performed in both debug and user modes.
- dmov command execution-The dmov command accesses SDMA internal registers. Executing a dmov instruction exchanges the 32-bit data values between the SDMA data register and the general register GReg[1].
- If the JTAG is used, the content of GReg1 is captured in the SDMA data register during the capture_dr state, then it is shifted out after 32 TCK clock cycles in the shift_dr state. During the update_dr state, GReg1 is updated with the new, shifted-in 32-bit data value. If the OnCE is driven from the Arm platform side, the data values contained in GReg1 and the SDMA data register are exchanged after detecting a write access to the ONCE_CMD register. The ONCE_DATA register must therefore be loaded first.

- exec_once command execution-The exec_once command executes the instruction loaded in the SDMA instruction register. The command may only be requested from debug mode. The SDMA returns to debug mode at the end of the execution.
- Change of flow instructions as well as instructions that may cause a context switch are not supported: The comprehensive list comprises done/yield/yiedge (except done 5), BF, BT, BSF, BDF, JMP, JSR, JMPR, JSRR, RET, and LOOP, as well as all the illegal instructions.

  No other command should be requested before the SDMA returns to debug mode. The SDMA status (for example, whether it is in debug mode or not) can be detected by polling with the rstatus OnCE command, monitoring the debug_mode pin, or checking the OnCE Status Register (SDMAARM_ONCE_STAT) register via the Arm platform control interface.

NOTE

Most of the instructions are single-cycle, which omits the step of polling the status. Loads and stores to DMA units are typical instructions that might require this polling.

If the JTAG is used, the 16-bit instruction is shifted in the SDMA instruction register after 16 TCK clock cycles in the shift_dr state. A request is sent to the core when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the Arm platform side, the request is sent to the SDMA when detecting a write access to the ONCE_CMD register. The ONCE_INSTR register must be therefore be loaded first.

- run_core command execution-The run_core command leaves debug mode and resume normal program execution. The next instruction executed is the last instruction decoded before entering debug mode. Be sure to restore core context before re-running the core. This procedure is detailed in Restoring the Context.
- If the JTAG is used, a 1-bit bypass value is shifted in the bypass register in the shift_dr state. The SDMA is rerun when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the Arm platform side, the core is rerun when detecting a write access to the ONCE_CMD register.

- exec_core command execution-The exec_core command resumes program execution from any address. The 16-bit instruction provided with the exec_core overwrites the last instruction decoded before entering debug mode. This command is designed to support change of flow instructions, so that a program execution can be restarted from any address. After executing an exec_core command, the SDMA leaves debug mode. The exec_core command is usually used with a jmp instruction.
- If the JTAG is used, the 16-bit branch instruction is shifted in the SDMA instruction register after 16 TCK clock cycles in the shift_dr state. The SDMA is rerun when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the Arm platform side, the SDMA reruns when detecting a write access to the ONCE_CMD register. The ONCE_INSTR register must therefore be loaded first. For example, to restart the SDMA from the program address 0x100, the instruction loaded should be a jump to address 0x100 instruction.

- debug_rqst command execution-The debug_rqst command puts the SDMA in debug mode. If the JTAG is used, a 1-bit bypass value is shifted in the bypass register during the shift_dr state. A debug request is sent to the SDMA when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the Arm platform side, the debug request is sent when detecting a write access to the ONCE_CMD register. When the SDMA is already in debug mode, this command is simply ignored.
- rbuffer command execution-The rbuffer command exports the content of the real time buffer (RTB). If the JTAG is used, the content of the real time buffer (RTB) is

captured in the SDMA data register during the capture_dr state. The register is completely shifted out after maintaining the shift_dr state during 32 TCK clock cycles. If the OnCE is driven from the Arm platform side, the content of the RTB is captured in the ONCE_DATA register after detecting a write access to the ONCE_CMD register.

- bypass command execution-This command is only available from the JTAG interface. It enables daisy-chaining of the SDMA JTAG TAP controller with other JTAG TAP controllers. This command does not change the SDMA state and can be executed in any mode (run, debug, or sleep). It selects the bypass register of the TAP controller.

7.2.3.13.4 Registers Descriptions

See SDMACORE, and SDMAARM, for detailed information on each register.

7.2.3.13.4.1 Event Cell Counter Register (ECOUNT)

The event cell counter register is a 16-bit register that contains the number of times minus one that an event detection occurs before generating a debug request.

This register should be written before attempting to use the event detection counter during an event detection process. The event cell counter register is cleared on a JTAG reset.

7.2.3.13.4.2 Event Cell Address Registers (EAA or EAB)

The event cell contains two address registers-the event cell address register (a), called EAA, and the event cell address register (b), called EAB. Every address register is a 16-bit register that stores a user-defined address value. This value computes one of the following address conditions: addra_cond or addrb_cond. Every address register is cleared on a JTAG reset.

7.2.3.13.4.3 Event Cell Address Mask Register (EAM)

The event cell address mask register is a 16-bit register that contains a user-defined address mask value. This mask is applied to the address value latched from the memory address bus before comparing addresses.

NOTE

There is a common address mask value for the two address comparators. If bit *i* of this register is set, then bit *i* of the

address value latched from the memory bus does not influence the result of the address comparison. The event cell address mask register is cleared on a JTAG reset.

7.2.3.13.4.4 Event Cell Data Register (ED)

The event cell data register is a 32-bit register that contains a user-defined data value. This data value is an input for the data comparator, which generates the data_cond condition.

The event cell data register is cleared on a JTAG reset.

7.2.3.13.4.5 Event Cell Data Mask Register (EDM)

The event cell data mask register is a 32-bit register that contains a user-defined data mask value. This mask is applied to the data value latched from the memory bus before comparing data.

Setting bit *i* of the event cell data mask register means that bit *i* of the data value latched from the address bus does not influence the result of the data comparison. The event cell data mask register is cleared on a JTAG reset.

7.2.3.13.4.6 Real Time Buffer Register (RTB)

The real Time Buffer register is a 32-bit register that stores and retrieves run-time information without putting the SDMA in debug mode.

Refer to Real Time Buffer for more details.

7.2.3.13.4.7 Event Control Register (ECTL)

The event cell control register is a 16-bit register that defines cell event occurrence conditions.

The event cell control register is cleared on a JTAG reset. See also OnCE Event Detection Unit for more details.

7.2.3.13.4.8 Trace Buffer (TB)

The Trace Buffer register retrieves the information in the Trace Buffer.

See Trace Buffer for more details.

7.2.3.13.4.9 OnCE Status Register (OSTAT)

The OnCE status register is a 16-bit register that contains processor and event detection unit status. The OSTAT is a read-only register.

Refer to OnCE Status Register (SDMAARM_ONCE_STAT) for detailed description of the individual fields in the OSTAT register.

The following figure shows the OSTAT structure.

Table 7-46. OnCE Status Register (OnCE)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PST[3:0]				RCV	EDR	ODR	SWB	MST					ECDR[2:0]		

Where PST[3:0] is the SDMA core state, RCV is set when the real-time buffer (RTB) is modified. EDR, ODR, and SWB are set, respectively, when the SDMA has entered debug mode because of an external debug request, a OnCE debug_rqst command, or a software breakpoint. MST is set when the OnCE is controlled from the Arm platform control interface, and when ECDR is a three-flag set that shows the event cell condition(s) that put the core in debug mode. The OSTAT never provides more than one reason for entering debug mode.

There are two ways of accessing OSTAT content, as follows:

1. Send an rstatus command to the OnCE controller through the JTAG, or read the ONCE_STAT register through the Arm platform access. Executing the rstatus command through the JTAG can be performed in both user and debug modes.
2. Perform an SDMA read access to the location in the SDMA core memory map (OSTAT register) debug mode using the exec_once command. With this method of access, the SDMA state reflected by the PST (processor status bit) is always DATA.

The register may also be accessed by a running application.

7.2.3.13.5 JTAG Interface Requirements

Because the signals received from the JTAG (running on TCK) are transferred to the OnCE controller (running on the SDMA clock), a synchronization mechanism is required.

7.2.3.13.5.1 TCK Speed Limitation

In the JTAG top-level layer, the TDO signal is always captured on a TCK falling edge. To guarantee a stable TDO signal from the SDMA during this operation, a falling edge detection is performed on TCK.

Before being latched in the *I* flip-flop (see Figure 7-11) on TCK falling edge, the TDO signal must be stable at the input of the flip-flop. This condition is verified if the TCK period is superior to the following delay:

worst-case edge detection delay + negative-edge signal propagation delay + JTAG top-level logic propagation delay

The frequency relationship, TCK < CLK/8, limitation guarantees that all operations are performed as expected.

7.2.3.13.5.2 Synchronization Implementation

The figure found here shows the synchronization mechanism.

Flip-flops tck0, tck1, and tck2 perform falling- and rising-edge detections on TCK. They generate the posedge_detected and negedge_detected nets that are used to sample the TDI and TMS inputs into the respective tdi and tms flip-flops, and update the tdo flip-flop to yield the TDO output. In the design, the only signal that might go metastable is the output of the tck0 flip-flop. This signal is captured in the tck1 flip-flop and no logical operation is performed on it to minimize a metastability propagation risk.

The TDI and TMS flip-flops also cannot go metastable: The propagation time of the rising-edge detection signal through tck0, tck1, and tck2 guarantees that the TDI and TMS inputs are stable when captured in the TDI and TMS flip-flops.

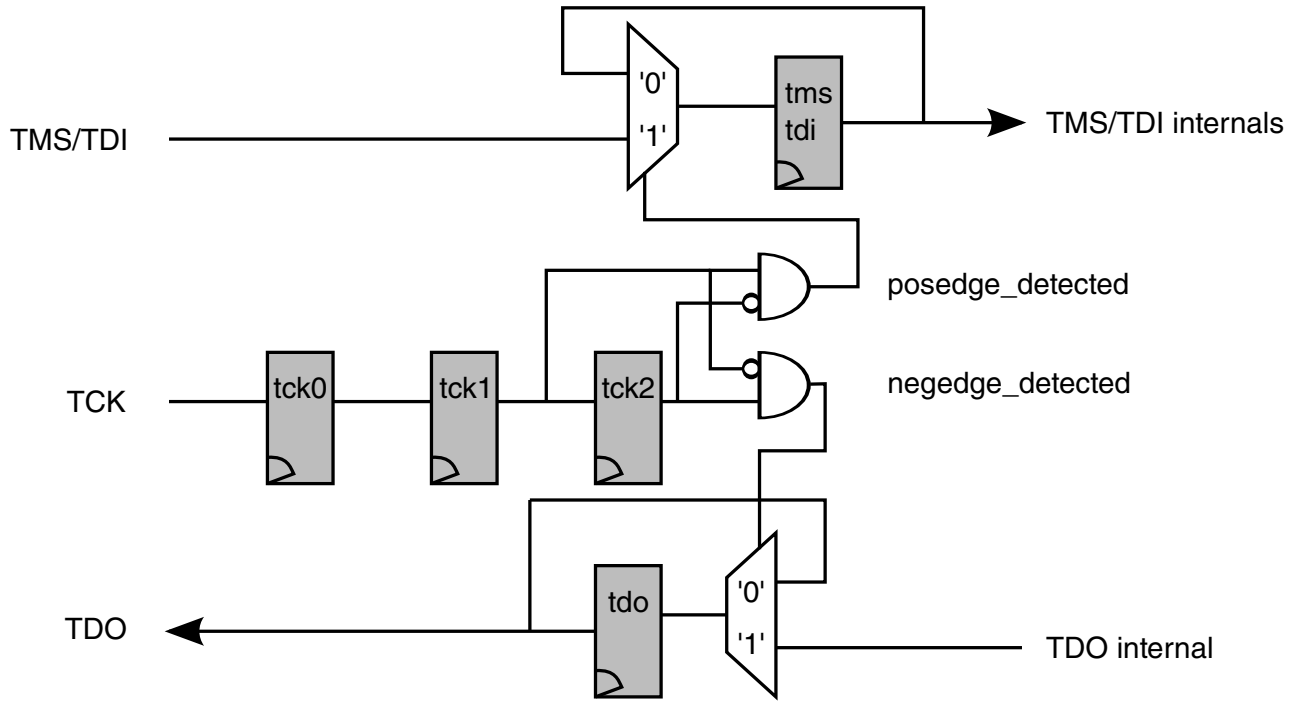


Figure 7-11. OnCE Synchronization Layer

The following figure shows synchronization timings. It takes three CLK clock cycles to synchronize TDI on the SDMA clock.

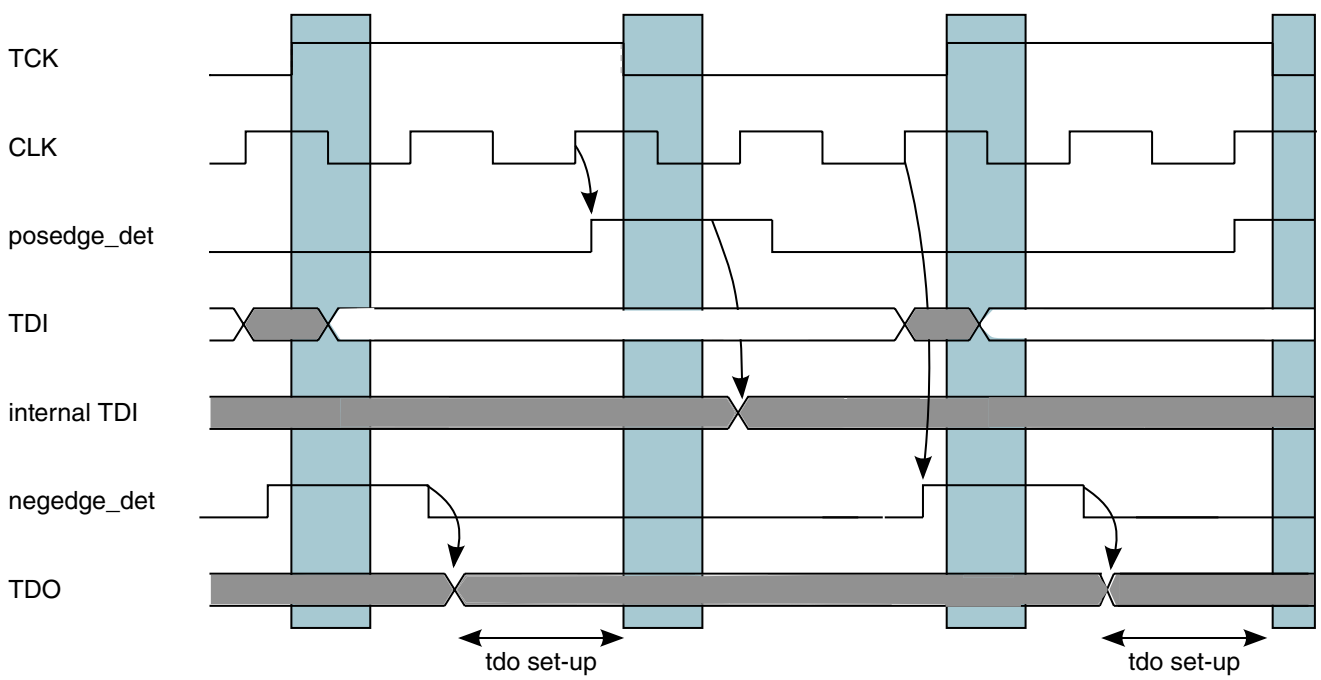


Figure 7-12. Synchronization Timings

7.2.3.13.5.3 JTAG Controller Start-Up Recommended Procedure

To ensure correct TAP controller initialization, it is recommended to use the following procedure:

1. Assert JTAG reset TRSTB (for example, set low).
2. Set TMS low.
3. Wait for 1 TCK clock.
4. Release JTAG reset TRSTB (for example, set high).
5. Wait for a minimum of five TCK cycles.

7.2.3.14 Using the OnCE

This section provides the elements necessary to run the OnCE during a debug process.

In addition to the basic set of commandsdescribed in OnCE Commands, more complex commands can be built to meet users' requirements.

7.2.3.14.1 Activating Clocks in Debug Mode

For power consumption issues, some clocks in the SDMA are disabled when not needed.

This is the case for instances when the SDMA is in sleep mode. Clock gating management depends on the interface used to control the OnCE.

- For the JTAG access, the SDMA clock gating must be turned off via the clk_gating_off input.
- For the Arm platform access, the SDMA clock gating is automatically turned off when the Arm platform access is enabled (see OnCE Enable (SDMAARM_ONCE_ENB)).

7.2.3.14.2 Getting the Current Status

Most of the commands the OnCE supports have an impact on the status of the SDMA.

It is not permissible to request the execution of an instruction on the SDMA from the OnCE while the SDMA is not in debug mode. Such a violation may cause unpredictable behavior, and it might be necessary to reset the SDMA.

Therefore, the value of the PST bits provided in the OnCE status register should always be checked before sending any request to the SDMA.

7.2.3.14.3 Methods of Entering Debug Mode

A debug request may be asserted at any time, but it is not always taken into account immediately. Debug mode cannot be entered in the middle of an instruction, or during the save or restore states of a context switch.

The request is ignored when the core is already in debug mode. Refer to Figure 7-4, which shows all possible transitions to the debug state, as there are several ways to enter debug mode.

7.2.3.14.3.1 External Debug Request During Reset

To enter debug mode after exiting reset, the external debug line has to be maintained high. This line is handled by the JTAG top-level block.

NOTE

The SDMA detects the debug requests only if the SDMA clock is running (see Activating Clocks in Debug Mode). The debug request line should be not be maintained high when the SDMA is in debug mode.

NOTE

The debug_rqst command (from the OnCE command set) is not supported during system reset.

7.2.3.14.3.2 Debug Request During Normal Activity

During normal activity, the SDMA enters debug mode when the following is true:

1. If the debug request line from the JTAG top-level is asserted, or
2. If the OnCE controller receives a debug_rqst command.

   The debug_rqst command can be sent by the JTAG access or by an access on the Arm platform side (if the Arm platform access is enabled).

7.2.3.14.3.3 Software Breakpoint Instruction

The SDMA enters debug mode at the end of the execution of a software breakpoint instruction. This instruction must be inserted in program flow executed by the core.

7.2.3.14.3.4 Event Detection Unit Matching Condition

If the event detection is enabled, a debug request is sent to the core after detecting a matching condition on the SDMA memory bus.

See OnCE Event Detection Unit for more details.

7.2.3.14.4 Executing Instructions in Debug Mode

The OnCE supports a mechanism to execute instructions in debug mode. If the SDMA is in debug mode, then the exec_once command can be used to execute an SDMA instruction from the OnCE controller. The SDMA returns to debug mode at the end of each execution.

Some instructions are not supported by the exec_once command: done/yield/yiedge (except done 5), BF, BT, BSF, BDF, JMP, JSR, JMPR, JSRR, RET, and LOOP, as well as all the illegal instructions are not supported.

NOTE

While instructions are executed in debug mode from the OnCE, the program counter of the SDMA is not incremented.

7.2.3.14.5 Command Sequences Examples

This section provides examples of command sequences that run the SDMA in debug mode. These sequences are available for both the Arm platform and JTAG accesses.

The following presents the syntax used in this section. The data field provided with each command is put in parenthesis with the command name. A '-' is used if the data field provided is a *don't care* value.

```
my_command(data_field);                 // executing my_command with a data field
my_command(-);                        // executing my_command with a don't care data field
```

The value returned by the command (if there is one) is referred by an assignment. In case the value returned by the command is not used, the assignment is omitted. For an Arm platform access, the value returned (it is always a data value) is obtained by reading back into the SDMA data register.

```
data_out = my_command(data_in); // returning a data value
```

To clarify the syntax, the instructions' opcodes are referred to by their names. In practice, use the corresponding 16-bit encoding.

7.2.3.14.5.1 Getting the SDMA Status

NOTE

Before executing any command that affects the SDMA (like dmov or exec_once), check that the SDMA is in debug mode.

Use the following snippet:

```
rstatus();        // read SDMA status until the SDMA is in debug mode
...
rstatus();
```

If the SDMA is not in debug mode, then a debug request must be generated. In this case, the SDMA enters debug mode at the end of the execution of the current instruction. Use this snippet:

```
debug_rqst(-);   // debug request
```

In the following sections, it is assumed that the SDMA was successfully put into debug mode.

7.2.3.14.5.2 Saving the Context

The first debug task is to save the SDMA context, which is the content of the eight general-purpose registers, the loop and PC-related registers, and the flags.

Use the general register GReg[1] as an intermediate register to export the entire context of the SDMA.

The following example shows how to save GReg[0], GReg[1], GReg[2] and GReg[3]. The sequence of commands used to export additional general registers is very similar to this.

Save GReg[0], GReg[1], GReg[2], and GReg[3]

```
GReg1_data = dmov(-);                          // the value exported is the content of
GReg[1]
exec_once("mov GReg1,GReg0");                      // puts the content of GReg[0] into
GReg[1]
GReg0_data = dmov(-);                          // the value exported is the content of
GReg[0]
exec_once("mov GReg1, GReg2");                     // puts the content of GReg[2] into
GReg[1]
GReg2_data = dmov(-);                          // the value exported is the content of
GReg[2]
exec_once("mov GReg1, GReg3");                     // puts the content of GReg[3] into
GReg[1]
GReg3_data = dmov(-);                          // the value exported is the content of
GReg[3]
```

Get the value of the internal flags (SF, DF, T, and LM), of the loop related registers (EPC and SPC), and of the PC-related registers (PC and RPC). Use a done 5, which is the formatting instruction dedicated to the debug. This instruction formats the flags and the

values contained in the registers. It also writes the resulting values into the channel context memory. It should not be used when entering debug from the IDLE state (for example, with no active channel script running on the SDMA), because it will update a channel context that may belong to any channel.

```
exec_once("done 5");                    // formatting the value of flags and registers
```

At this point, the channel context should be up-to-date in memory, and debug operations should now be possible. However, the context can be exported with the following instructions:

Exporting the Context

```
dmov(ctx_base_addr);                                  // loading GReg[1] with the channel
context base address
exec_once("ld GReg0,(GReg1,0)");                          // get RPC-PC into GReg0
exec_once("ld GReg1, (GReg1,1)");                         // get SPC-EPC into GReg1
Loop_data = dmov(-);                                // read back the value of Loop registers
exec_once("mov GReg1, GReg0");                          // puts the PC info into GReg1
PC_data = dmov(-);                                  // reads back the content of the PC registers
```

After this sequence of operations, the entire SDMA context is exported via the OnCE.

7.2.3.14.5.3 Restoring the Context

At this point in the operation, restore the context of the SDMA. It can be different from the original context located in memory, and the content previously saved into the debugging application via the OnCE.

The example found hereshows how it is possible to modify the current channel context.

Modifying the Current Channel Context

```
dmov(Loop_data);                                    // put Loop former value into GReg[1]
exec_once("mov GReg0, GReg1");                          // copy to GReg[0]
dmov(PC_data);                                      // put PC former value into GReg[1]
exec_once("mov GReg2, GReg1");                          // copy to GReg[2]
dmov(ctx_base_addr);                                  // put channel context base address into
GReg[1]
exec_once("st GReg0, (GReg1,1)");                         // restore Loop context
exec_once("st GReg2, (GReg1,0)");                         // restore PC context
```

Once the context in memory is the desired context (with or without applying the previous instruction sequence), it can be restored to the *real* PC and loop registers in the SDMA core:

```
exec_once("cpShReg");                    // restore flags and PC & loop related registers
```

After this command, the SDMA core PC, RPC, SPC, EPC registers, as well as the flags contain the same data as what is stored in the context RAM for the current channel.

The following example shows how to restore the context of general registers GReg[0], GReg[1], GReg[2] and GReg[3].

Restoring the General Register Context

```
dmov(GReg3_data);                    // put GReg[3] restore value in GReg[1]
exec_once("mov GReg3, GReg1"); // restore GReg[3]
dmov(GReg2_data);                    // put GReg[2] restore value in GReg[1]
exec_once("mov GReg2, GReg1"); // restore GReg[2]
dmov(GReg0_data);                    // put GReg[0] restore value in GReg[1]
exec_once("mov GReg0, GReg1"); // restore GReg[0]
dmov(GReg1_data);                    // restore GReg[1]
```

At this point, it is possible to restart the normal program execution.

NOTE

Every SDMA core general register value can be modified by a mov instruction, which makes modification of these registers easy during debug. Unfortunately, there is no such instruction as a mov to directly modify the contents of either PCU register or flag (PC, RPC, SPC, EPC, T, LM, SF, or DF). The cpShReg instruction is meant to provide a means for changing these register contents via the context memory.

7.2.3.14.5.4 Accessing the Memory

In the example shown here, it is assumed that the SDMA context is entirely saved. If true, it is permissible to modify the general purpose registers during debugging activity.

To perform a memory read access, the target address is stored via the OnCE in GReg[1], then the load instruction is executed on the SDMA (the data loaded from the memory overwrites the address contained in GReg[1]), and then the result value is read back via the OnCE.

```
macro READ:            dmov(target_addr);                                  // put the target
address in GReg[1]
                    exec_once("ld GReg1,(GReg1,0)");                        // execute the
load instruction
                    res_data = dmov(-);                                 // exports the result
data value
```

For a memory write access, the target address is written in GReg[0], and the value to store is written in GReg[1]. Then the store instruction is executed on the SDMA.

```
macro WRITE:            dmov(target_addr);                                   // puts the
target address in GReg[1]
                    exec_once("mov GReg0,GReg1");                          // puts the target
address in GReg[0]
                    dmov(target_data);                                  // puts the target
data in GReg[1]
                    exec_once("st GReg1,(GReg0,0)");                          // performs the
store operation
```

This sequence is shown as an example; however, many other sequences are possible.

NOTE

This sequence of commands can also be applied to memory-mapped registers.

7.2.3.14.5.5 Resuming Program Execution

Before resuming program execution, it is assumed that the SDMA context is properly restored. There are two ways to restart the SDMA.

Start by executing the last instruction fetched before entering debug mode, as follows.

```
run_core(-);                          // resume execution from where we stopped before
```

If necessary, restart the execution from a different address. In this case, use the exec_core command. The data field provided with this command must be the encoding of a jump instruction.

```
exec_core("jmp start_addr"); // rerun the SDMA from another address
```

In these two examples, the SDMA exits debug mode and keeps executing the code fetched from the memory.

7.2.3.14.5.6 Single Stepping in RAM

To execute a program step-by-step from the RAM, insert software breakpoints in the program flow at appropriate places so that the SDMA only executes one instruction before returning to debug mode.

First, read the next instruction to execute in the RAM. Then, depending on the value of this instruction, compute the address where a software breakpoint instruction should be inserted. The instruction at the corresponding address must be saved, and, the software breakpoint instruction is inserted. After restarting the SDMA, there is only one instruction executed before meeting the software breakpoint.

The following example shows the macro functions READ and WRITE, which correspond to the sequence of commands (described above) used to access the memory.

NOTE

The data read from the memory are 32-bit values, while the instructions are 16-bit values only. This is why it is best to only use addresses divided by two when accessing the memory.

READ and WRITE Macro Functions

```
next_instr = READ(run_addr/2);                        // read the next instruction to execute
// the tool now has to compute the address where the breakpoint
// instruction should be inserted, this address is the "bkpt_addr"
```

```
instr_save = READ(bkpt_addr/2);                          // save the instruction before
overwriting
STORE("bkpt instruction",bkpt_addr/2);                        // store the bkpt instruction
in memory
exec_core("jmp run_addr");                              // rerun the SDMA
rstatus(-);                                          // wait for the SDMA to enter debug mode
...
rstatus(-);
STORE(instr_save,bpkt_addr/2);                           // restore the instruction
overwritten
```

In case of branched conditional instructions, a breakpoint instruction should be written at the two possible target addresses.

7.2.3.14.5.7 Single Stepping in ROM

No single-step mechanism is supported in ROM. The program code can be loaded in the RAM, where the single-step mechanism can be executed.

7.2.3.14.6 OnCE Event Detection Unit

The event detection unit watches signals from the data memory bus (DMBUS), which the SDMA core uses to access its RAM, ROM, and memory mapped registers.

A debug request is sent to the OnCE controller when user-defined conditions on address and/or data values are true.

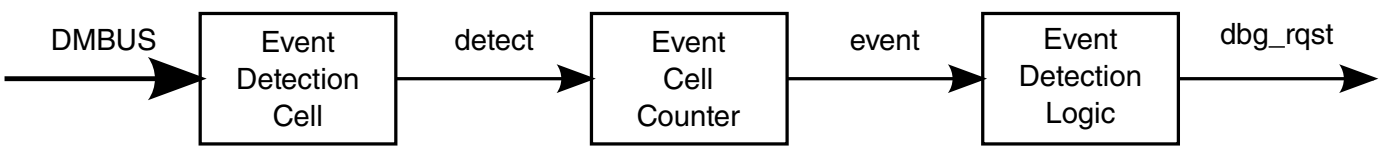


Figure 7-13. Event Detection Unit

A counter, provided with the detection cell, is decreased after an event detection. A debug request is sent to the core only when the counter reaches the value of 0. It is possible to disable the use of the counter if a debug request has to be generated after each event detection.

The event cell is the basic block that supports hardware breakpoints on an address value and/or data values coming from the SDMA memory bus. The trigger condition that generates the debug request is a mixed condition based on those values.

The following figure shows the event cell architecture. The event cell contains the address (stored in the memory address register) and the data (stored in the memory data register) used during the last memory access. There are some user-defined reference

values located in memory mapped registers-the event cell addresses, the event cell address mask, the event cell data, and the event cell data mask. These registers are accessed by standard load/store instructions just like regular memory locations.

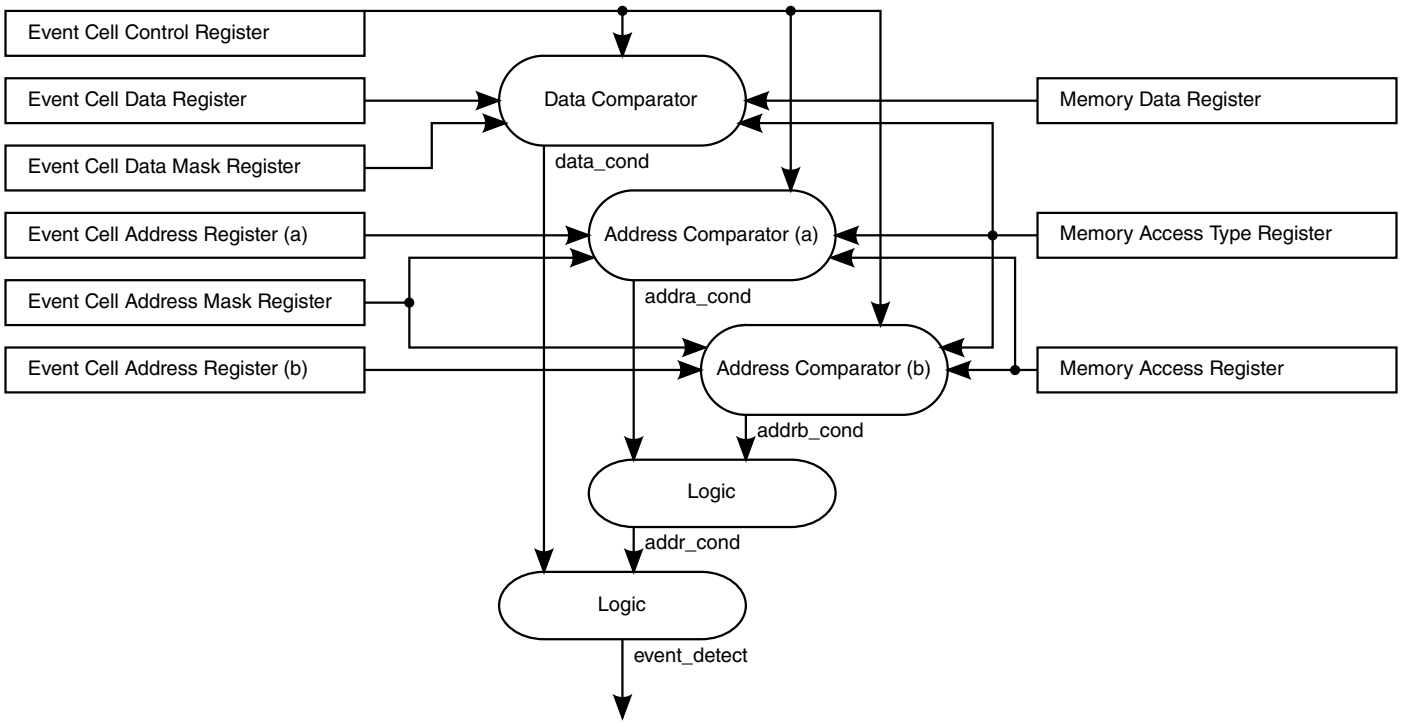


Figure 7-14. Event Cell Architecture

To define a memory breakpoint, three conditions are taken into account: The first two conditions are comparisons of the current memory address with user-defined reference addresses (these conditions are called addressA and addressB). The third condition consists of a comparison between the data received on the DMBUS and a user-defined reference data (this condition is called data). An intermediate address condition is set to express a dependency between addressA and addressB conditions.

7.2.3.14.7 Clock Gating and Reset

This section details how to use the clocks and handle the reset signals.

7.2.3.14.7.1 Clocks

Because the SDMA uses clock gating to save power, it is necessary to disable the clock gating and force the clocks to be enabled when using the OnCE.

When the OnCE is accessed through its JTAG interface, clock gating must be disabled outside the SDMA via a dedicated SDMA input port clk_gating_off. The reason why detection is not performed automatically by the SDMA internal hardware is that it would cost power to monitor activity on the JTAG interface.

When the OnCE is accessed through the Arm platform Control interface, clock gating is automatically turned off. This is done when bit 0 of the ONCE_ENB register (see OnCE Enable (SDMAARM_ONCE_ENB)) is set. A write access to this register is possible even when the OnCE clock is not running. If the Arm platform access is used, the bit in the ONCE_ENB register must be set before any attempt to access any other OnCE register.

7.2.3.14.7.2 Resets

The OnCE reset is different from the SDMA main reset.

Normally, activating the SDMA reset while keeping the OnCE reset inactive (when possible) enables you to reset the core without having to reprogram the OnCE.

7.2.3.14.8 Real Time Features

To rebuild the skeleton of a program execution, it is necessary to store the addresses of the program instructions where jumps are taken: A trace buffer is therefore provided. A real time buffer has also been added to receive data values written during a program execution.

The content of this register may be exported through JTAG ports without stopping the core.

7.2.3.14.8.1 Trace Buffer

The Trace Buffer is a 32-stage buffer that contains appropriate information to identify the 32 last changes of flow detected during a program execution.

The following figure shows an overview of the Trace Buffer.

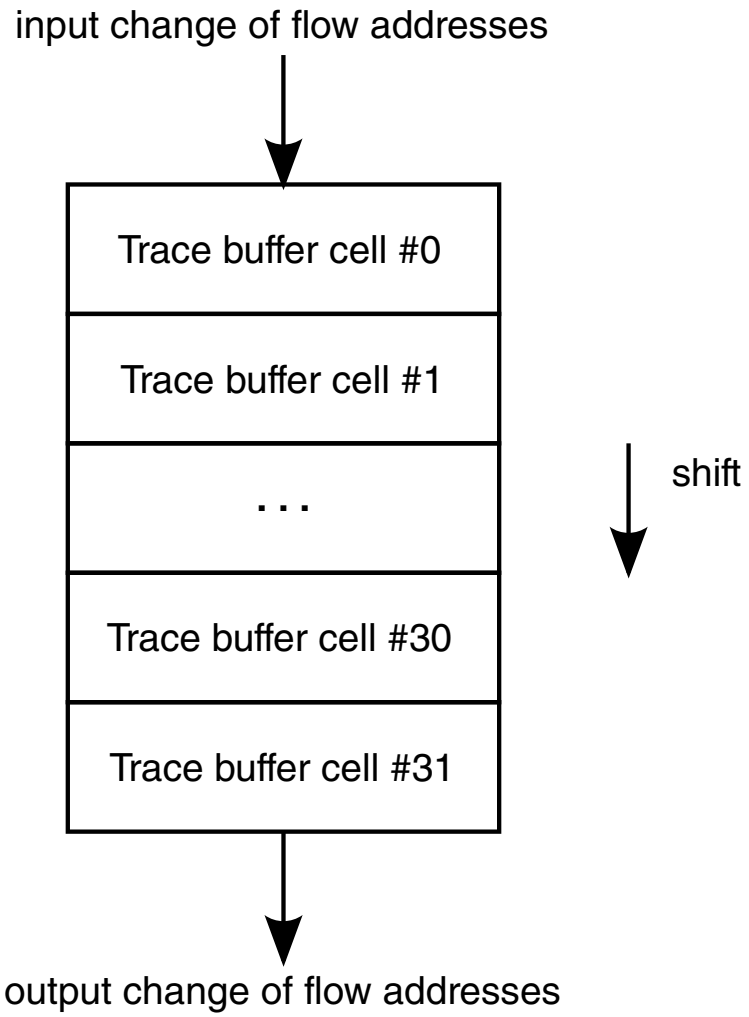


Figure 7-15. Trace Buffer

Each cell of the trace buffer contains two reference addresses and a flag. The flag is set when the addresses stored in the cell correspond to a valid change of flow; otherwise, the flag is cleared. The three most significant bits are unused.

After every change of flow detection, the address of current instruction and the address of the target instruction are stored at the top of the Trace Buffer (cell #0). The flag in the cell is set to indicate that a valid change of flow was detected. Former cell values are shifted one level down. The Trace Buffer contains the 32 last changes of flow. All the flags are reset on a software or a hardware reset, and after each transition from debug mode to user mode.

A memory mapped register of SDMA core, the Trace Buffer register (TB), is provided to read the content of the Trace Buffer. This operation should be done in debug mode. Performing a read access to the Trace Buffer register returns the content of the bottom of the Trace Buffer (cell #31). After every read access, the trace buffer is shifted one level down, and the flag at the top of the trace buffer is cleared.

A typical OnCE command sequence that retrieves the oldest change-of-flow information is a follows:

```
exec_once("mov r1, TB");                              // stores the oldest change-of-flow in
GReg1
dmov(-);                                          // retrieves GReg1 contents
```

This sequence requires the SDMA to be put in debug mode.

7.2.3.14.8.2 Real Time Buffer

The Real Time Buffer register (RTB) is a memory mapped register that can be accessed as a regular memory location by the SDMA core during program execution. This register is located in the OnCE.

Executing ar rbuffer command (see The OnCE Controller for further details) exports the content of this register through JTAG ports.

When a write access is performed at the memory location corresponding to the RTB, the receive flag (for example, the RCV bit) is set in the OnCE Status Register (OSR). This flag is cleared at the end of the execution of a rbuffer command.

NOTE

Every write access to the RTB memory location updates the RTB register even if the RCV flag is set. The RTB is cleared on a JTAG reset.

7.2.3.14.8.3 Emulation Pin

The debug_matched_event emulation pin reflects the matching condition status detected by the Event Detection Unit.

Since it can be necessary to detect conditions without triggering debug requests, it is possible to disable the generation of debug requests by the Event Detection Unit and still have the matching condition available on the emulation pin. This can be done by clearing the EN flag in the ECTL register.

7.2.3.14.8.4 Real-Time Debug Outputs

The table found here shows the debug signals that are available at the SDMA boundaries. Their availability at chip boundaries depends on the project.

Table 7-47. Real-Time Debug Output Pins

Pin	Description
debug_core_state[3:0]	The core_state bits reflect the state of the SDMA core. • The "Program" state is the usual instruction execution cycle. • The "Data" state is inserted when there are wait-states during a load or a store on the data bus (ld or st). • The "Change of Flow" state is the second cycle of any instruction that breaks the sequence of instructions (jumps and channel switching instructions). • The "Change of Flow in Loop" state is used when an error causes a hardware loop exit. • The "Debug" state means the SDMA is in debug mode. • The "Functional Unit" state is inserted when there are wait-states during a load or a store on the functional units bus (ldf or stf). • In "Sleep" modes, no script is running (this is the core idle state); the "after Reset" is slightly different because no context restoring phase will happen when a channel is triggered: The script located at address 0 is executed (boot operation). • The "in Sleep" states are the same as above except they do not have any corresponding channel: they are used when entering debug mode after reset; the reason is that it is necessary to return to the "Sleep after Reset" state when leaving debug mode. 0 Program 1 Data 2 Change of Flow 3 Change of Flow in Loop 4 Debug 5 Functional Unit 6 Sleep 7 Context Switch Saving Channel 8 Program in Sleep 9 Data in Sleep 10 Change of Flow in Sleep 11 Change of Flow in Loop in Sleep 12 Debug in Sleep 13 Functional Unit in Sleep 14 Sleep after Reset 15 Context Switch Restoring Channel
debug_yield	Pulse that is active when a yield (done 0) or a yieldge (done 1) instruction is executed. 0 - 1 yield/yieldge executed
debug_core_run	Active when the SDMA core is executing instructions. 0 Debug or sleep mode

Table continues on the next page...

Table 7-47. Real-Time Debug Output Pins (continued)

Pin	Description
	1 Run mode
debug_event_channel_sel	Indicates if debug_event_channel displays current channel or last received event 0- debug_event_channel[5:0] gives the number of the current channel 1- debug_event_channel[5:0] gives the number of the last received event
debug_event_channel[5:0]	Gives the number of any DMA request as soon as it is received or the number of the current channel. The value of debug_event_channel_sel indicates if debug_event_channel displays the current channel or last received event. The signal debug_event_channel_sel must be observed to determine what information is provided on debug_event_chanel at any given time.
debug_pc[13:0]	Program Counter value; it has a meaning when the core is in run mode.
debug_mode	Set when the core is in debug. 0 - 1 Core is in debug
debug_bus_error	Set when an error was received during a load or a store (ld, st, ldf, or stf instruction) and registered in SF or DF flag. 0 No error during last load/store 1 Error during last load/store
debug_bus_device[4:0]	Indicates the device or functional unit that is accessed by the current instruction. The debug_bus_device output is always valid when in sleep mode, debug mode, or executing any instruction that does not access the functional units or the memory mapped devices, "no access" is output. 0 No access 1 MSA 2 MDA 3 MD 4 MS 5 PSA 6 PDA 7 PD 8 PS 9 RESERVED 10 RESERVED 11 RESERVED 12 RESERVED 13 CA 14 CS 15 Reserved 16 Memory (RAM or ROM) 17 Memory mapped register

Table continues on the next page...

Table 7-47. Real-Time Debug Output Pins (continued)

Pin	Description
	18 Peripheral #1 19 Peripheral #2 20 Peripheral #3 21 Peripheral #4 22 Peripheral #5 23 Peripheral #6 24 Peripheral #7 25 Peripheral #8 26 Peripheral #9 27 Peripheral #10 28 Peripheral #11 29 Peripheral #12 30 Peripheral #13 31 Peripheral #14
debug_bus_rwb	Indicates the direction of the access given by debug_bus_device 0 Write access (st or stf) 1 Read access (ld or ldf)
debug_matched_dmbus	Pulse indicating the OnCE event detection unit has detected a match on the data bus during an access to memory (RAM or ROM), a memory mapped register or a peripheral that is hooked to the SDMA. 0 - 1 data bus match detected
debug_rtbuffer_write	Pulse indicating when the real-time buffer is written by the core. 0 - 1 RTB was modified
debug_evt_chn_lines[7:0]	Eight lines that generate short pulses when DMA requests are received or channels are (re)started. Every line is controlled through two parameters defined in registers Cross-Trigger Events Configuration Register 1 (SDMAARM_XTRIG_CONF1) (as described in SDMAARM). The following two parameters are available for every line: • CNF-Indicates what is monitored on the line: 0 for a channel start, 1 for a DMA request reception • NUM[5:0]-Gives the number of the DMA request or channel to monitor

The matched_event emulation pin reflects the matching condition status detected by the Event Detection Unit. Because it can be necessary to detect conditions without triggering debug requests, it is possible to disable the generation of debug requests by the Event Detection Unit and still have the matching condition available on the emulation pin. This can be done by clearing the EN flag in the ECTL register.

All real-time debug outputs are disabled by default (for example, they are stuck to 0) to avoid power consumption when they are not used. They are enabled when bit 11 (RTDOBS) of the Configuration Register (SDMAARM_CONFIG) is set. Signals provided to the system JTAG controller for SDMA debug mode status will also be enabled when the *clk_gating_off* input is asserted.

7.2.4 Instruction Set

7.2.4.1 Instruction Encoding

This section presents a short summary of the instruction codes. All context switch instructions are listed for information only; they cannot function properly out of the context switch routine.

```
x...x - don't care

rrr - destination/source general register

sss - additional source general register

bbb - general register used as address base register

ddddd - address displacement

nnnnn - bit number
uuuuuuuu - function unit command bits

pppppppp - branch displacement (signed)

iiiiiiii - 8-bit immediate

jjj - control bit to clear

ff - flag to clear
00000jjj00000000        - done (done,yield,wait)
00000jjj00000001        - notify
00000xxx00000010        - reserved
00000xxx00000011        - reserved
00000xxx00000100        - reserved
0000000000000101     - softBkpt
0000000100000101     - reserved
0000001000000101     - reserved
0000001100000101     - reserved
0000010000000101     - reserved
0000010100000101     - reserved
0000011000000101     - reserved
0000011100000101     - reserved
0000000000000110     - ret
0000000100000110     - reserved
0000001000000110     - reserved
0000001100000110     - reserved
0000010000000110     - reserved
0000010100000110     - reserved
0000011000000110     - reserved
0000011100000110     - reserved
```

```
000000ff00000111       - clrf ff
0000010000000111     - reserved
0000010100000111     - reserved
0000011000000111     - reserved
0000011100000111     - illegal
00000rrr00001000        - jmpr r
00000rrr00001001        - jsrr
00000rrr00001010        - ldrpc r
00000rrr00001011        - reserved
00000rrr000011xx          - reserved
00000rrr00010000        - revb
00000rrr00010001        - revblo
00000rrr00010010        - rorb
00000rrr00010011        - reserved
00000rrr00010100        - ror1
00000rrr00010101        - lsr1
00000rrr00010110        - asr1
00000rrr00010111        - lsl1
00000rrr001nnnnn        - bclri r,n
00000rrr010nnnnn        - bseti r,n
00000rrr011nnnnn        - btsti r,n
00000xxx10000xxx           - reserved
00000rrr10001sss           - mov
00000rrr10010sss           - xor
00000rrr10011sss           - add
00000rrr10100sss           - sub
00000rrr10101sss           - or
00000rrr10110sss           - andn
00000rrr10111sss           - and
00000rrr11000sss           - tst
00000rrr11001sss           - cmpeq
00000rrr11010sss           - cmplt
00000rrr11011sss           - cmphs
0000011011100000     - reserved
0000011011100001     - reserved
0000011011100010     - cpShReg
0000011011100011     - reserved
0000011011100100     - reserved
0000011011100101     - reserved
0000011011100110     - reserved
0000011011100111     - reserved
00000xxx11101xxx           - reserved
00000xxx11110xxx           - reserved
00000xxx11111xxx           - reserved
00001rrriiiiiiii                - ldi r,i
00010rrriiiiiiii                - xori r,i
00011rrriiiiiiii                - addi r,i
00100rrriiiiiiii                - subi r,i
00101rrriiiiiiii                - ori r,i
00110rrriiiiiiii                - andni r,i
00111rrriiiiiiii                - andi r,i
01000rrriiiiiiii                - tsti r,i
01001rrriiiiiiii                - cmpeqi r,i
01010rrrddddbbb         - ld r,(d,b)
01011rrrddddbbb         - st r,u
01100rrruuuuuuuu         - ldf r,u
01101rrruuuuuuuu         - stf r,u
011100xxxxxxxxxx               - reserved
011101xxxxxxxxxx               - reserved
011110ffnnnnnnnn       - Loop ff flags are reset
01111100pppppppp     - bf pc=pc+signed(pppppppp)+1
01111101pppppppp     - bt pc=pc+signed(pppppppp)+1
01111110pppppppp     - bsf pc=pc+signed(pppppppp)+1
01111111pppppppp     - bdf pc=pc+signed(pppppppp)+1
10aaaaaaaaaaaaaa     - jmp absolute
11aaaaaaaaaaaaaa     - jsr absolute
```

7.2.4.2 SDMA Instruction Set

This section describes all the useful instructions from the SDMA set.

Table 7-48. SDMA Instruction List

Instruction	Description	Page
ADD	Addition	ADD (Addition)
ADDI	Add with Immediate Value	ADDI (Add with Immediate Value)
AND	Logical AND	AND (Logical AND)
ANDI	Logical AND with Immediate Value	ANDI (Logical AND with Immediate Value)
ANDN	Logical AND NOT	ANDN (Logical AND NOT)
ANDNI	Logical AND with Negated Immediate Value	ANDNI (Logical AND with Negated Immediate Value)
ASR1	Arithmetic Shift Right by 1 Bit	ASR1 (Arithmetic Shift Right by 1 Bit)
BCLRI	Bit Clear Immediate	BCLRI1 (Bit Clear Immediate)
BDF	Conditional Branch if Destination Fault	BDF (Conditional Branch if Destination Fault)
BF	Conditional Branch if False	Functional Units Programming Model
BSETI	Bit Set Immediate	BSETI (Bit Set Immediate)
BSF	Conditional Branch if Source Fault	BSF (Conditional Branch if Source Fault)
BT	Conditional Branch if True	BT (Conditional Branch if True)
BTSTI	Bit Test immediate	BTSTI (Bit Test immediate)
CLRF	Clear Arm platform flags	CLRF (Clear Arm platform flags)
CMPEQ	Compare for Equal	CMPEQ (Compare for Equal)
CMPEQI	Compare with Immediate for Equal	CMPEQI (Compare with Immediate for Equal)
CMPHS	Compare for Higher or Same	CMPHS (Compare for Higher or Same)
CMPLT	Compare for Less Than	CMPLT (Compare for Less Than)
cpShReg	Update Context of PCU Registers and Flags	cpShReg (Update Context of PCU Registers and Flag)
DONE	DONE, Yield	DONE (DONE, Yield)
ILLEGAL	ILLEGAL Instruction	ILLEGAL (ILLEGAL Instruction)
JMP	Unconditional Jump Immediate	JMP (Unconditional Jump Immediate)
JMPR	Unconditional Jump	JMPR (Unconditional Jump)
JSR	Unconditional Jump to Subroutine Immediate	JSR (Unconditional Jump to Subroutine Immediate)
JSRR	Unconditional Jump to Subroutine	JSRR (Unconditional Jump to Subroutine)
LD	Load Register	LD (Load Register)
LDF	Load Register from Functional Unit	LDF (Load Register from Functional Unit)
LDI	Load Register with Immediate Value	LDI (Load Register with Immediate Value)
LDRPC	Load from RPC to Register	LDRPC (Load from RPC to Register)

Table continues on the next page...

Table 7-48. SDMA Instruction List (continued)

Instruction	Description	Page
LOOP	Hardware Loop	LOOP (Hardware Loop)
LSL1	Logical Shift Left by 1 Bit	LSL1 (Logical Shift Left by 1 Bit)
LSR1	Logical Shift Right by 1 Bit	LSR1 (Logical Shift Right by 1 Bit)
MOV	Logical Move	MOV (Logical Move)
NOTIFY	Notify to Arm platform	NOTIFY (Notify to Arm platform)
OR	Logical OR	OR (Logical OR)
ORI	Logical OR with Immediate Value	ORI (Logical OR with Immediate Value)
RET	Return from Subroutine	RET (Return from Subroutine)
REVB	Reverse Byte Order	REVB (Reverse Byte Order)
REVBLO	Reverse Low Order Bytes	Reverse Low Order Bytes(REVBLO)
ROR1	Rotate Right by 1 Bit	ROR1 (Rotate Right by 1 Bit)
RORB	Rotate Right by 1 Byte	RORB (Rotate Right by 1 Byte)
SOFTBKPT	Software Breakpoint	SOFTBKPT (Software Breakpoint)
ST	Store Register	ST (Store Register)
STF	Store Register in Functional Unit	STF (Store Register in Functional Unit)
SUB	Subtract	SUB (Subtract)
SUBI	Subtract with Immediate	SUBI (Subtract with Immediate)
TST	Test with Zero	TST (Test with Zero)
TSTI	Test Immediate	TSTI (Test Immediate)
XOR	Logical Exclusive OR	XOR (Logical Exclusive OR)
XORI	Exclusive OR with Immediate	XORI (Exclusive OR with Immediate)

7.2.4.2.1 ADD (Addition)

Operation:

```
GReg[r] ← GReg[s] + GReg[r]

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: add r,s

Example: add 0,3
```

ADD GReg[3] and GReg[0] and store the result in GReg[0]

CPU Flags: T

Cycles: 1

Description: Performs the ADDition of the source general register *s* and the destination general register *r*, and stores the result in the destination general register *r*. The T flag is set if the result of the operation is 0. It is cleared if the result is not 0.

Instruction Format:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	0	1	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.2 ADDI (Add with Immediate Value)

Operation:

```
GReg[r] ← GReg[r] + immediate

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: addi r,immediate

Example: add 6,112
```

ADD GReg[6] and decimal value 112 and store the result in GReg[6]

CPU Flags: T

Cycles: 1

Description: Adds a 0-extended immediate value to a general register; stores the result in the general register. The flag T is set when the result of the operation is 0; otherwise, it is cleared. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

 ...

11111110 - 254

11111111 - 255
```

7.2.4.2.3 AND (Logical AND)

Operation:

```
GReg[r] ← GReg[s] & GReg[r]
```

Assembler:

```
Syntax: and r,s

Example: and 1,2
```

AND GReg[1] and GReg[2] and store the result in GReg[1]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the AND of the source general register *s* and the destination general register *r*, and stores the result in the destination general register *r*.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	1	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.4 ANDI (Logical AND with Immediate Value)

Operation:

```
GReg[r] ← GReg[r] & immediate
```

Assembler:

```
Syntax: andi r,immediate

Example: andi 7,45
```

AND GReg[7] and decimal value 45 and store the result in GReg[7]

CPU Flags: unaffected

Cycles: 1

Description: Performs an AND between a 0-extended immediate value and a general register; stores the result in the general register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.5 ANDN (Logical AND NOT)

Operation:

```
GReg[r] ← ~GReg[s] & GReg[r]
```

Assembler:

```
Syntax:andn r,s

Example: andn 3,4
```

AND GReg[3] and NOT GReg[4] (bit inverted) and store the result in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the AND of the negation of the source general register *s* and the destination general register *r*, and stores the result in the destination general register *r*.

Instruction Format:

Table 7-49. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	1	0	s	s	s

Instruction Fields:

rrr /sss - destination register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.6 ANDNI (Logical AND with Negated Immediate Value)

Operation:

```
GReg[r] ← GReg[r] & ~immediate
```

Assembler:

```
Syntax: andni r,immediate

Example: andni 0,2
```

AND GReg[0] and decimal value -3 (inverted 32-bit value 2) and store the result in GReg[0]

CPU Flags: unaffected

Cycles: 1

Description: Performs an AND between the negation of a 0-extended 8-bit immediate value and a general register; stores the result in the general register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.7 ASR1 (Arithmetic Shift Right by 1 Bit)

Operation:

```
GReg[r]:{b31,b30,...,b1,b0} ← GReg[r]:{b31,b31,b30,...,b1}
```

Assembler:

```
Syntax: asr1 r

Example: asr1 3
```

divide by 2 the signed value of GReg[3] and store the result in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Shift the bits of any general register to the right and keep the same sign: The left bit (bit 31) is kept untouched.

Instruction Format:

Table 7-50. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	1	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.8 BCLRI1 (Bit Clear Immediate)

Operation:

```
GReg[r]:{b31,...,b(i+1),0,b(i-1),...,b0} ← GReg[r]:{b31,...,b(i+1),b(i),b(i-1),...,b0}
```

Assembler:

```
Syntax: bclri r,i

Example: bclri 1,12
```

clear bit 12 in GReg[1]

CPU Flags: Unaffected

Cycles: 1

Description: Clear the bit of register r specified by the 5-bit immediate field

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	1	i	i	i	i	i

rrr - register field:

```
000 - GReg[0]
```

```
001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiii - immediate value:

```
00000 - 0

00001 - 1

...

11110 - 30

11111 - 31
```

7.2.4.2.9 BDF (Conditional Branch if Destination Fault)

Operation:

```
if (DF == 1) PC ← PC + 1 + displacement else PC ← PC + 1
```

Assembler:

```
Syntax:bdf label

Example: bdf LLL
```

Jump to LLL if DF is set, or go to the next instruction if DF is cleared; the displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: If flag DF is set, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag DF is cleared, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	p	p	p	p	p	p	p	p

Instruction Fields:

pppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110 - 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 - (-2)

11111111 - (-1)
```

7.2.4.2.10 BF (Conditional Branch if False)

Operation:

```
if (T == 0)

PC ← PC + 1 + displacement

else

PC ← PC + 1
```

Assembler:

```
Syntax: bf label

Example: bf LLL
```

Jump to LLL if T is cleared, or go to the next instruction if T is set. The displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: Conditional branch: If flag T is cleared, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag T is set, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	0	0	p	p	p	p	p	p	p	p

Instruction Fields:

ppppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110 - 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 - (-2)

11111111 - (-1)
```

7.2.4.2.11 BSETI (Bit Set Immediate)

Operation:

```
GReg[r]:{b31,...,b(i+1),1,b(i-1),...,b0} ← GReg[r]:{b31,...,b(i+1),b(i),b(i-1),...,b0}
```

Assembler:

```
Syntax: bseti r,i

Example: bseti 6,5
```

Set bit 5 in GReg[6]

CPU Flags: Unaffected

Cycles: 1

Description: Sets bit number i in the selected General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	1	0	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]
```

```
010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiii - bit number field:

```
00000 - 0

00001 - 1

...

11110 - 30

11111 - 31
```

7.2.4.2.12 BSF (Conditional Branch if Source Fault)

Operation:

```
if (SF == 1) PC ← PC + 1 + displacement else PC ← PC + 1
```

Assembler:

```
Syntax: bsf label

Example: bsf LLL
```

Jump to LLL if SF is set, or go to the next instruction if SF is cleared. The displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: Conditional branch: If flag SF is set, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag SF is cleared, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	0	p	p	p	p	p	p	p	p

Instruction Fields:

pppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110 - 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 - (-2)

11111111 - (-1)
```

7.2.4.2.13 BT (Conditional Branch if True)

Operation

```
if (T == 1)

PC ← PC + 1 + displacement

else

PC ← PC + 1
```

Assembler

```
Syntax: bt label

bt LLL
```

Jump to LLL if T is set, or go to the next instruction if T is cleared. The displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: Conditional branch: If flag T is set, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag T is cleared, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	0	1	p	p	p	p	p	p	p	p

pppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110- 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 -  (-2)

11111111 - (-1)
```

7.2.4.2.14 BTSTI (Bit Test immediate)

Operation:

```
T ← GReg[r]:b(i)
```

Assembler:

```
Syntax: btsti r,i

Example: btsti 2,29
```

Test bit 29 in GReg[2] and copy its value in flag T

CPU flags: T

Cycles: 1

Description: T is loaded with the value of bit number i from the selected general register.

Instruction Format:

Table 7-51. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	1	1	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]
```

```
010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiii - bit number field:

```
0000 - 0

0001 - 1

...

11110 - 30

11111 - 31
```

7.2.4.2.15 CLRF (Clear Arm platform flags)

Operation:

```
 if (ff%2 == 0)

SF ← 0

if (ff/2 == 0)

DF ← 0
```

Assembler:

```
Syntax: clrf ff

Example: clrf 2
```

Clear flag SF and keep flag DF unchanged

CPU Flags: SF, DF

Cycles: 1

Description: Clears a selection of the Arm platform fault flags: SF, DF, both SF and DF or none can be cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	f	f	0	0	0	0	0	1	1	1

Instruction Fields:

ff - flags field:

```
00 - clear SF and clear DF

01 - clear DF

10 - clear

SF 11 - no clear
```

7.2.4.2.16 CMPEQ (Compare for Equal)

Operation:

```
T ← (GReg[s] == GReg[r])
```

Assembler:

```
Syntax: cmpeq r,s

Example: cmpeq 7,5
```

Compare GReg[7] and GReg[5] and set flag T if they are equal

CPU flags: T

Cycles: 1

Description: Subtracts the destination general register *r* from the source general register *s*, and sets T if the result is 0, clears T if the result is not 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	0	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.17 CMPEQI (Compare with Immediate for Equal)

Operation:

```
T ← (GReg[r] == immediate)
```

Assembler:

```
Syntax: cmpeqi r,immediate

Example: cmpeqi 2,13
```

Compare GReg[2] and decimal value 13 and set flag T if they are equal

CPU Flags: T

Cycles: 1

Description: Subtracts the 0-extended 8-bit immediate value from the general register, and sets T if the result is 0, clears T if the result is not 0. The immediate value is the low-order byte of the instruction.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - destination register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254
```

```
11111111 - 255
```

7.2.4.2.18 CMPHS (Compare for Higher or Same)

Operation:

```
T ← (GReg[r] ≥ GReg[s])
```

Assembler:

```
Syntax: cmphs r,s

Example: cmphs 0,1
```

Compare GReg[0] and GReg[1] and set flag T if GReg[0] is higher than or equal to GReg[1]

CPU Flags: T

Cycles: 1

Description: Compares the destination general register *r* and the source general register *s*, and sets T if the destination general register *r* is higher than or equal to the source general register *s*, clears T otherwise. The comparison is unsigned.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	1	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.19 CMPLT (Compare for Less Than)

Operation:

```
T ← (GReg[r] < GReg[s])
```

Assembler:

```
Syntax: cmplt r,s

Example: cmplt 7,4
```

Compare GReg[7] and GReg[4] and set flag T if GReg[7] is lower than GReg[4]

CPU Flags: T

Cycles: 1

Description: Compares the destination general register *r* and the source general register *s*, and sets T if the destination general register *r* is lower than the source general register *s*, clears T otherwise. The comparison is signed.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	1	0	s	s	s

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.20 cpShReg (Update Context of PCU Registers and Flag)

Assembler:

```
Syntax: cpShReg
```

CPU Flags: none

Cycles: 1

Description: SF, RPC, T, PC,LM, EPC, DF, and SPC registers are updated according to the value of their corresponding bits in the context memory. This instruction must only be used in debug mode via the OnCE. It reverses the done *5* operation.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	0

7.2.4.2.21 DONE (DONE, Yield)

Operation:

```
if (jjj&6 == 2) HE[CCR] ← 0

if (jjj == 3) HI[CCR] ← 1

if (jjj == 4) EP[CCR] ← 0


if ((jjj == 0) && (NCP > CCP)) CCR ← NCR

else if ((jjj == 1) && (NCP >= CCP))

CCR ← NCR

else

CCR ← NCR
```

(CCR stands for Current Channel Register; NCR stands for Next Channel Register)

Assembler:

```
Syntax: done jjj

Example: done 3
```

Clear HE bit for the current channel, send an interrupt to the Arm platform for the current channel and reschedule.

CPU Flags: Unaffected

Cycles: Variable if a context switch is done, 1 otherwise

Description: Clears one of the channel enabling bits (HE or EP for the corresponding channel number) if required. Sends an interrupt to the corresponding Arm platform by setting the appropriate flag, if required (HI for the corresponding channel number). Reschedules according to the mode and the NCP (Next Channel Priority) and CCP (Current Channel Priority) values. According to the scheduling decision, the NCR (Next Channel Register) is copied to the CCR (Current Channel Register) and channel contexts are switched. If several channels with the same highest priority are pending, they are ordered by their number from 31 down to 0. The higher number is selected (for example, channel 26 is selected if channels 3, 12, 14, and 26 with the same highest priority are pending). If no flag is modified, the reschedule can allow the replacement of the current

channel by another channel with a priority strictly greater than the current channel priority (yield). Or, it can allow the replacement of the current channel by another channel with a priority greater than or equal to the current channel priority (yieldge). In the latter case, the selected channel will always be the first one with the same priority, starting from channel number 31 down to channel 0 (the current channel does not belong to the set of selectable channels).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	j	j	j	0	0	0	0	0	0	0	0

jjj - Channel Flags field:

000 - No channel flags affected: Reschedule only if the next channel priority is greater than current channel priority (yield)

001 - No channel flags affected: Reschedule only if the next channel priority is greater than or equal to the current channel priority (yieldge)

010 - Clear HE for the current channel and reschedule 011 - Clear HE, set HI for the current channel and reschedule 100 - Clear EP for the current channel and reschedule

101 - Reserved for debug to copy relevant registers into context memory

110 - RESERVED

111 - RESERVED

For the scheduling rules, refer to Scheduler Functional Description. Every possible done instruction is further described as follows:

- done 0/yield is executed by a channel script when it accepts preemption by a higher priority channel;
- done 1/yieldge is executed by a channel script when it accepts preemption by a higher priority channel and it also accepts a roll-up with other channels that have the same priority;
- done 2 is executed by a channel script that was triggered by a Arm platform start via the Channel Start (SDMAARM_HSTART) register, when its task is completed and it requires termination;
- done 3 is executed by a channel script that was triggered by a Arm platform start via the Channel Start (SDMAARM_HSTART) register, when its task is completed, it requires termination and it needs to trigger an interrupt to the Arm platform upon closure;

- done 4 is executed by a channel script that was triggered by a DMA request, when its task is completed and it requires termination;
- done 5 is used in debug mode only; it copies the PCU registers and flags to the context memory of the current channel;

7.2.4.2.22 ILLEGAL (ILLEGAL Instruction)

Operation:

```
PC ← 0001
```

Assembler:

```
Syntax: illegal
```

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the Illegal instruction routine located at address 0001. All unauthorized instructions result in an Illegal instruction behavior; however, the ILLEGAL instruction must be used to guarantee software compatibility with future versions of the SDMA.

Instruction Format

Table 7-52. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

7.2.4.2.23 JMP (Unconditional Jump Immediate)

Operation:

```
PC ← absolute_address
```

Assembler:

```
Syntax: jmp label

Example: jmp LLL
```

The assembler translates the label to the exact address

CPU Flags:Unaffected

Cycles: 2

Description: Jumps to the absolute address contained the lower 14 bits of the instruction (the PC is a 14-bit register).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	a	a	a	a	a	a	a	a	a	a	a	a	a	a

aaaaaaaaaaaaaa - address field:

```
00000000000000 - 0

00000000000001 - 1

...

11111111111110 - 16382

11111111111111 - 16383
```

7.2.4.2.24 JMPR (Unconditional Jump)

Operation:

```
PC ← GReg[r]
```

Assembler:

```
Syntax: jmpr r

Example: jmpr 0
```

Jump to address stored in GReg[0]

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the absolute address contained in a General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	0	1	0	0	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]
```

```
010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.25 JSR (Unconditional Jump to Subroutine Immediate)

Operation:

```
RPC ← PC + 1

PC ← absolute_address
```

Assembler:

```
Syntax: jsr r

Example:jsr LLL
```

Jumps to subroutine starting at LLL; the assembler translates the label to exact address

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the subroutine located at the absolute address contained the lower 14 bits of the instruction (the PC is a 14-bit register).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	a	a	a	a	a	a	a	a	a	a	a	a	a	a

aaaaaaaaaaaaaa - address field:

```
00000000000000 - 0

00000000000001 - 1

...

11111111111110 - 16382

11111111111111 - 16383
```

7.2.4.2.26 JSRR (Unconditional Jump to Subroutine)

Operation:

```
RPC ← PC + 1

PC ← GReg[r]
```

Assembler:

```
Syntax: jsrr r

Example:jsrr 5
```

Jumps to subroutine located at address stored in GReg[5]

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the subroutine at address contained in a General Register

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	0	1	0	0	1

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.27 LD (Load Register)

Operation:

```
GReg[r] ← [GReg[b] + displacement]

if (transfer_error)

SF ← 1

else

SF ← 0
```

Assembler:

```
Syntax: ld r,(b,displacement)

Example: ld 1,(2,23)
```

Loads data into GReg[1]; the data is located at address obtained by adding decimal value 23 to GReg[2]

CPU Flags: SF

Cycles: 2+n where n is 0 for ROM, RAM or memory mapped registers, and n is the number of wait-states of the peripheral for a peripheral access

Description: Adds a 5-bit 0-extended displacement to a base address in General Register b; the result is the address of the data to fetch on the DM bus. The data received from the bus is stored in the destination General Register r. If an error occurs during the transfer, the flag SF is set, else it is cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	r	r	r	d	d	d	d	d	b	b	b

rrr / bbb - register field:

```
 000 - GReg[0]

 001 - GReg[1]

 ...

 111 - GReg[7]
```

ddddd - displacement value:

```
00000 - 0

00001 - 1

...

11111 - 31
```

7.2.4.2.28 LDF (Load Register from Functional Unit)

Operation:

```
GReg[r] ← [fu_address]

if (transfer_error)

SF ← 1
```

```
else

SF ←0
```

fu_address is an 8-bit field and depends on addressed functional unit

Assembler:

```
Syntax: ldf r,fu_address

Example: ldf 0,13
```

Loads data coming from the Burst DMA register MD into GReg[0]; it is a 32-bit access with no prefetch

CPU Flags: SF

Cycles: 1+n where n is the number of wait-states that may be inserted by the functional unit

Description: Sends an 8-bit address on the Functional Unit Bus (FU bus) and stores the data received from the bus in the destination General Register r. If an error occurs during the transfer, the flag SF is set, else it is cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	r	r	r	f	f	f	f	f	f	f	f

See the following sections for more details of the LDF instruction usage with each functional unit:

- Burst DMA Read (ldf) for Burst DMA
- Peripheral DMA Read (ldf)-Read Mode for Peripheral DMA

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

ffffffff - functional unit source register and action (unspecified values are reserved):

```
00000000 - MSA

00000100 - MDA

00001001 - MD byte

00001010 - MD halfword

00001011 - MD word

00001100 - MS

00101001 - MD byte - prefetch

00101010 - MD halfword - prefetch

00101011 - MD word - prefetch

01000000 - DSA



11000000 - PSA

11001000 - PD

11010000 - PDA

11011000 - PD in copy mode (rrr contents are lost)

11101000 - PD - prefetch next data

11111111 - PS
```

7.2.4.2.29 LDI (Load Register with Immediate Value)

Operation:

```
GReg[r] ← immediate
```

Assembler:

```
Syntax: ldi r,immediate

Example: ldi 6,1
```

loads decimal value 1 into GReg[6]

CPU Flags: Unaffected

Cycles: 1

Description: Stores a 0-extended immediate value in a General Register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.30 LDRPC (Load from RPC to Register)

Operation:

```
GReg[r] ← RPC
```

Assembler:

```
Syntax: ldrpc r

Example: ldrpc 3
```

copies RPC to GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Stores the contents of the RPC in a General Register. That instruction may be used to have more than one level of subroutines.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	0	1	0	1	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.31 LOOP (Hardware Loop)

Operation:

```
if (ff%2 == 0)
   SF ← 0
if (ff/2 == 0)
  DF ← 0
if ((GReg[0] == 0) || (SF == 1) || (DF == 1))
   PC ← PC + loop_size + 1
else
   {
      SPC ← PC + 1
      EPC ← PC + loop_size + 1
      LM ← 1
      PC ← PC + 1
   }
```

during every instruction execution in the loop:

```
if ((SF == 1) || (DF == 1))
 {
```

```
    LM ← 0
    PC ← EPC
  }
else if ((PC + 1) == EPC)
 {
    GReg[0] ← GReg[0] - 1
    if (GReg[0] == 0)
      {
         LM ← 0
         PC ← EPC
       }
     else
         PC ← SPC
   }
else
  PC ← nextPC(instruction)
```

after the execution of the last instruction of the loop body:

```
if (GReg[0] == 0)
   T ← 1
else
   T ← 0
```

Assembler:

```
Syntax: loop n{,ff}
Example: loop 3,1
```

Executes GReg[0] times the instructions comprised between PC+1 and PC+3 (included); ff=1 clears the DF flag before starting the loop. When omitted, the ff field is set to 0 (clearing both SF and DF).

CPU Flags: LM[1:0], T

Cycles: 2 when the loop count (GReg[0]) is 0 or SF or DF is set at loop start, 1+1 when the loop starts but exits abnormally (SF or DF set inside the loop which adds 1 cycle to the offending load or store to jump to EPC), 1 when the loop is executed normally

Description: The loop instruction executes a sequence of instructions several times. The number of times is given by the contents of GReg[0], the loop counter. SDMA will jump to the first instruction after the end of the loop if the value in GReg[0] is 0. Otherwise the SDMA enters loop mode. It sets the most significant bit of the LM flag that will only be reset once the last instruction of the last loop is executed. The instructions in the loop are executed GReg[0] times. The management of fault flags (SF and DF) is as follows. When entering the hardware loop, SF and DF can be cleared according to the ff field of the instruction. After that operation, if any flag is still set the loop will not be executed. The SDMA will jump to the first instruction after the end of the loop without entering loop mode. During the execution of the loop, if any fault flag is set by a LD, LDF, ST, or STF instruction, the SDMA will immediately exit loop mode and jump to the first instruction after the end of the loop. In that case, GReg0 is not decremented for that last piece of the loop body execution (even if the SF or DF flag is set at the last instruction of the loop body). The T flag reflects the state of GReg[0] after the end of the loop, which is an indicator of the complete execution of the loop. If the loop exited because of an error (SF or DF set), GReg[0] will not be 0 at the end of the loop, hence T will be cleared. If the loop executes without fault, GReg[0] will be 0 at the end of the loop, hence T will be set. The boundary case when a source or destination fault occurs at the last instruction of the last loop is considered as an anticipated exit of the loop, which causes the T flag to be cleared. If the last instruction executed before leaving the hardware loop also tries to modify the T flag, the flag is updated according to the value of GReg[0], NOT according to the result of the last executed instruction.

Limitations:

1. 1. Jump instructions (JMP, JMPR, JSR, JSRR, BF, BT, BSF, BDF) are not allowed inside the hardware loop.
2. 2. GReg[0] cannot be written to inside the hardware loop (it can be read).
3. 3. The empty loop (0 instruction in the body) is forbidden.
4. 4. If GReg[0] == 0 at the start of the loop, which causes a jump to EPC, the T flag is not updated.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	0	f	f	n	n	n	n	n	n	n	n

Instruction Fields:

```
ff - flags field:

00 - clear SF and clear DF

01 - clear DF

10 - clear SF
```

```
11 - no clear
```

nnnnnnnn - loop size

```
00000000 - empty loop: forbidden value

00000001 - 1 instruction in the loop

00000010 - 2 instructions in the loop

...

11111111 - 255 instructions in the loop
```

7.2.4.2.32 LSL1 (Logical Shift Left by 1 Bit)

Operation:

```
GReg[r]:{b30,...,b1,b0,0} ← GReg[r]:{b31,b30,...,b1,b0}
```

Assembler:

```
Syntax: lsl1 r

Example: lsl1 2
```

multiplies by 2 the value in GReg[2]

CPU Flags: Unaffected

Cycles: 1

Description: Shift the bits of any General Register to the left. The right bit (bit 0) is set to 0. No overflow is detected by the hardware.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	1	1

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]
```

```
111 - GReg[7]
```

7.2.4.2.33 LSR1 (Logical Shift Right by 1 Bit)

Operation:

```
GReg[r]:{0,b31,b30,...,b1} ← GReg[r]:{b31,b30,...,b1,b0}
```

Assembler:

```
Syntax: lsr1 r

Example: lsr1 4
```

divides by 2 the unsigned value contained in GReg[4]

CPU Flags: Unaffected

Cycles: 1

Description: Shift the bits of any General Register to the right. The left bit (bit 31) is set to 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	0	1

Instruction Fields:

rrr - destination register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.34 MOV (Logical Move)

Operation:

```
GReg[r] ← GReg[s]
```

Assembler:

```
Syntax: mov r,s

Example: mov 4,0
```

copies GReg[0] to GReg[4]

CPU Flags: Unaffected

Cycles: 1

Description: Move the contents of the source General Register s to the destination General Register r.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	0	0	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.35 NOTIFY (Notify to Arm platform)

Operation:

```
 if (jjj & 4 == 0)

 {

   if (jjj&2 == 2)

      HE[CCR] ← 0

   if (jjj&1== 1)

       HI[CCR] ← 1

  }

else if (jjj == 4)
```

```
   EP[CCR] ← 0
else
```

(CCR stands for Current Channel Register)

Assembler:

```
Syntax: notify jjj
Example: notify 3
```

clears the HE bit for the current channel and sends an interrupt to the Host for the current channel

CPU Flags: Unaffected

Cycles: 1

Description: Clears one of the channel enabling bits (HE or EP for the corresponding channel number) if required, sends an interrupt to the corresponding Arm platform by setting the appropriate flag if required (HI for the corresponding channel number).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	j	j	j	0	0	0	0	0	0	0	1

jjj - Channel Flags field:

```
000 - unused
001 - set HI for the current channel
010 - clear HE for the current channel
011 - clear HE, set HI for the current channel
100 - clear EP for the current channel
101 - RESERVED
110 - RESERVED
111 - RESERVED
```

7.2.4.2.36 OR (Logical OR)

Operation:

```
GReg[r] ← GReg[s] | GReg[r]
```

Assembler:

```
Syntax: or r,s
```

```
Example: or 3,6
```

ORs GReg[3] and GReg[6] and stores the result in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the OR of the source General Register s and the destination General Register r, and stores the result in the destination General Register r.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	0	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.37 ORI (Logical OR with Immediate Value)

Operation:

```
GReg[r] ← GReg[r] | immediate
```

Assembler:

```
Syntax: ori r,immediate

Example: ori 1,56
```

ORs GReg[1] and the decimal value 56 and stores the result in GReg[1]

CPU Flags: unaffected

Cycles: 1

Description: Performs an OR between a 0-extended 8-bit immediate value and a General Register; stores the result in the General Register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.38 RET (Return from Subroutine)

Operation:

```
PC ← RPC
```

Assembler:

```
Syntax: ret
```

CPU Flags: Unaffected

Cycles: 2

Description: Return from subroutine.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

7.2.4.2.39 REVB (Reverse Byte Order)

Operation:

```
GReg[r]:{B3,B2,B1,B0} ← GReg[r]:{B0,B1,B2,B3}
```

Assembler:

```
Syntax: revb r

Example:  revb 5
```

reverses bytes order in GReg[5]

CPU Flags: Unaffected

Cycles: 1

Description: Reverse the byte order of any General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	0	0	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.40 Reverse Low Order Bytes(REVBLO)

Operation:

```
GReg[r]:{B3,B2,B0,B1} ← GReg[r]:{B3,B2,B1,B0}
```

Assembler:

```
Syntax: revblo r

Example: revblo 0
```

reverses low order bytes in GReg[0]

CPU Flags: Unaffected

Cycles: 1

Description: Reverse both low order bytes of any General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	0	0	1

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.41 ROR1 (Rotate Right by 1 Bit)

Operation:

```
GReg[r]:{b0,b31,b30,...,b1} ← GReg[r]:{b31,b30,...,b1,b0}
```

Assembler:

```
Syntax: ror1 r

Example: ror1 3
```

rotates bits to the right in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Rotate the bits of any General Register to the right.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	0	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.42 RORB (Rotate Right by 1 Byte)

Operation:

```
GReg[r]:{B0,B3,B2,B1} ← GReg[r]:{B3,B2,B1,B0}
```

Assembler:

```
Syntax: rorb r

Example: rorb 2
```

rotates bytes to the right in GReg[2]

CPU Flags: Unaffected

Cycles: 1

Description: Rotate the bytes of any General Register to the right.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	0	1	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.43 SOFTBKPT (Software Breakpoint)

Operation:

Stops the current script and enters debug mode

Assembler:

```
softbkpt
```

CPU Flags: Unaffected

Description: When the core executes this instruction, it has the same effect as receiving a debug request from the OnCE or via the external debug request input: the script execution halts, the PCU enters its debug state and waits for the OnCE commands that are described in OnCE and Real-Time Debug.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

7.2.4.2.44 ST (Store Register)

Operation:

```
[GReg[b] + displacement] ← GReg[r]
```

```
if (transfer_error)

 DF ← 1

 else

DF ← 0
```

Assembler:

```
Syntax: st r,(b,displacement)

Example: st 7,(0,9)
```

stores the value from GReg[7] into memory at address obtained by adding decimal value 9 to GReg[0]

CPU Flags: DF

Cycles: 2+n where n is 0 for ROM, RAM or memory mapped registers, and n is the number of wait-states of the peripheral for a peripheral access

Description: Adds a 5-bit 0-extended displacement to a base address in General Register b; the result is the address of the data to store on the DM bus. The data sent on the bus comes from the source General Register r. If an error occurs during the transfer, the flag DF is set, else it is cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	1	r	r	r	d	d	d	d	d	b	b	b

Instruction Fields:

rrr / bbb - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

ddddd - displacement value:

```
00000 - 0
```

```
00001 - 1

...

11111 - 31
```

7.2.4.2.45 STF (Store Register in Functional Unit)

Operation:

```
[fu_address] ← GReg[r] 0

if (transfer_error) 0

DF ← 1 0

else 0

DF ← 0
```

fu_address is an 8-bit field

Assembler:

```
Syntax: stf r,fu_address

Example: stf 3,0x2B
```

stores the 32-bit contents of GReg[3] to the Burst DMA register MD; waits until the flush to external memory is completed

CPU Flags: DF

Cycles: 1+n where n is the number of wait-states that may be inserted by the functional unit

Description: Sends an 8-bit address on the Functional Unit Bus (FU bus) and sends the contents of the source General Register r on the bus. If an error occurs during the transfer, the flag DF is set, else it is cleared.

Table 7-53. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	1	r	r	r	f	f	f	f	f	f	f	f

See the following sections for more details of the STF instruction usage with each functional unit:

- Burst DMA Write (stf) for Burst DMA
- Peripheral DMA Write (stf)-Write Mode for Peripheral DMA

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

ffffffff - functional unit destination register and action (unspecified values are reserved):

00000000 - MSA in incremented mode

00000100 - MDA in incremented mode

00001001 - MD byte

00001010 - MD halfword

00001011 - MD word

00001100 - clear MS error flag

00001111 - MS

00010000 - MSA in frozen mode

00010100 - MDA in frozen mode

00011000 - MD in copy mode - number of words in rrr

00100000 - MSA in incremented mode - start prefetch

00101000 - MD no data - flush

00101001 - MD byte - flush

00101010 - MD halfword - flush

00101011 - MD word - flush

00110000 - MSA in frozen mode - start prefetch

11000001 - PSA in frozen mode - 8-bit data width

11000010 - PSA in frozen mode - 16-bit data width

11000011 - PSA in frozen mode - 32-bit data width

11000101 - PSA in incremented mode - 8-bit data width

11000110 - PSA in incremented mode - 16-bit data width

11000111 - PSA in incremented mode - 32-bit data width

11001000 - PD

11001001 - PSA in decremented mode - 8-bit data width

11001010 - PSA in decremented mode - 16-bit data width

11001011 - PSA in decremented mode - 32-bit data width

11001100 - clear PS error flag

11001101 - PSA data width becomes 8-bit

11001110 - PSA data width becomes 16-bit

11001111 - PSA data width becomes 32-bit

11010001 - PDA in frozen mode - 8-bit data width

11010010 - PDA in frozen mode - 16-bit data width

11010011 - PDA in frozen mode - 32-bit data width

11010101 - PDA in incremented mode - 8-bit data width

11010110 - PDA in incremented mode - 16-bit data width

11010111 - PDA in incremented mode - 32-bit data width

11011001 - PDA in decremented mode - 8-bit data width

11011010 - PDA in decremented mode - 16-bit data width

11011011 - PDA in decremented mode - 32-bit data width

11011101 - PDA data width becomes 8-bit

11011110 - PDA data width becomes 16-bit

11011111 - PDA data width becomes 32-bit

11100001 - PSA in frozen mode - 8-bit data width - prefetch data

11100010 - PSA in frozen mode - 16-bit data width - prefetch data

11100011 - PSA in frozen mode - 32-bit data width - prefetch data

11100101 - PSA in incremented mode - 8-bit data width - prefetch data

11100110 - PSA in incremented mode - 16-bit data width - prefetch data

11100111 - PSA in incremented mode - 32-bit data width - prefetch data

11101001 - PSA in decremented mode - 8-bit data width - prefetch data

11101010 - PSA in decremented mode - 16-bit data width - prefetch data

11101011 - PSA in decremented mode - 32-bit data width - prefetch data

11101101 - PSA data width becomes 8-bit - prefetch data

11101110 - PSA data width becomes 16-bit - prefetch data

11101111 - PSA data width becomes 32-bit - prefetch data

11111111- PS

7.2.4.2.46 SUB (Subtract)

Operation:

```
GReg[r] ← GReg[r] - GReg[s]

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: sub r,s

Example: sub 4,7
```

SUBtracts GReg[7] from GReg[4] and stores the result in GReg[4]

CPU Flags: T

Cycles: 1

Description: Subtracts the source General Register s from the destination General Register r, and stores the result in the destination General Register r. The T flag is set if the result of the operation is 0; it is cleared if the result is not 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	0	0	s	s	s

Instruction Fields:

rrr / sss - register fields:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.47 SUBI (Subtract with Immediate)

Operation:

```
GReg[r] ← GReg[r] - immediate

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: sub r,immediate

Example: sub 1,255
```

SUBtracts decimal value 255 from GReg[1] and stores the result in GReg[1]

CPU Flags: T

Cycles: 1

Description: Subtracts a 0-extended 8-bit immediate value from a General Register; stores the result in the General Register. The flag T is set when the result of the operation is 0; otherwise, it is cleared. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.48 TST (Test with Zero)

Operation:

```
T ← ((GReg[s] & GReg[r]) != 0)
```

Assembler:

```
Syntax: tst r,s

Example: tst 2,3
```

ANDs GReg[2] and GReg[3] and sets T if the result is non-null

CPU Flags: T

Cycles: 1

Description: Performs the AND of the source General Register s and the destination General Register r, and sets T if the result is not 0, clears T if the result is 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	0	0	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.49 TSTI (Test Immediate)

Operation:

```
T ← ((GReg[r] & immediate) != 0)
```

Assembler:

```
Syntax: tsti r,immediate

Example: tsti 5,13
```

ANDs GReg[5] and decimal value 13 and sets T if the result is non-null

CPU Flags: T

Cycles: 1

Description: Performs the AND of a 0-extended 8-bit immediate value and the destination General Register r, and sets T if the result is not 0, clears T if the result is 0. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - destination register field:

```
000 - GReg[0]

001 - GReg[1]
```

```
010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.50 XOR (Logical Exclusive OR)

Operation:

```
GReg[r] ← GReg[s] ^ GReg[r]
```

Assembler:

```
Syntax: xor r,s

Example: xor 0,3
```

XORs GReg[0] and GReg[3] and stores the result in GReg[0]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the eXclusive OR of the source General Register s and the destination General Register r, and stores the result in the destination General Register r.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	0	1	0	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]
```

```
010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.51 XORI (Exclusive OR with Immediate)

Operation:

```
GReg[r] ← GReg[r] ^ immediate
```

Assembler:

```
Syntax: xori r,immediate

Example: xor 7,5
```

XORs GReg[5] and decimal value 5 and stores the result in GReg[7]

CPU Flags: Unaffected

Cycles: 1

Description: Performs an eXclusive OR between a 0-extended 8-bit immediate value and a General Register; stores the result in the General Register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]
```

```
111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.52 YIELD, YIELDGE (DONE, Yield)

By default, unsupported assembler syntax. Can be aliased to the corresponding done instructions (yield = done 0; yieldge = done 1). Refer to the done instruction description DONE (DONE, Yield) .

7.2.5 Software Restrictions

7.2.5.1 Unsupported Burst DMA Access Sequence

The SDMA does not support triggering a pre-fetch followed by a flush of the Burst DMA without reading or writing any data. If the flush occurs while the background pre-fetch DMA operation is still in progress, it could result in un-defined behavior.

An example of the sequence which could result in undefined results is shown in the following example:

Instruction sequence not supported

```
stf r1, MSA|PF          ; Update source address, triggers data pre-fetch in the
                        ; background
mov R0,R0               ; Execute multiple assembly instructions, none of which
                        ; read
mov R0,R0               ; or write data to/from MD
stf MD|SZ0|FL           ; Flush FIFO without writing data. If the pre-fetch is still
                        ; in progress when this instruction is executed, there
                        ; could be undefined operation
```

A work-around to avoid any undesirable results is to first read MD to ensure the pre-fetch is complete before the flush is attempted.

Work-Around to previous example

```
stf r1, MSA|PF          ; Update source address, triggers data pre-fetch.
```

```
mov R0,R0              ; Execute multiple assembly instructions, none of which
                       ; read
mov R0,R0              ; or write data to/from MD
ldf r2, MD             ; dummy read of MD to ensure pre-fetch is complete
                       ; before the next instruction
stf MD|SZ0|FL          ; Flush FIFO without writing data
```

7.2.6 Application Notes

7.2.6.1 Data Structures for Boot Code and Channel Scripts

SDMA boot code downloads the different channel contexts and the scripts that will be executed on SDMA channels during the application.

The boot code is run after reset when channel 0 is started by the Arm platform. The boot code is also known as channel 0 script.

The boot code is based on the Channel Control Block (CCB) and Buffer Descriptor (BD) mechanisms that are data structures located into the Arm platform memory space. With these data structures, it is possible to instruct SDMA to download scripts and contexts but also to dump a context or a script to a destination data buffer. Channel scripts also use the CCB and BD data structures to pass instructions and/or pointers to data to be copied.

The format, processing, and field definition of the CCB and BD are defined and performed entirely by the software script rather than the SDMA hardware. An overview of the format and structure is provided here, but for complete details refer to the SDMA software documentation (see SDMA Scripts).

The CCB and BD data structures are accessed by SDMA using DMA and processed by the SDMA scripts. The ROM contains common sub-routines for processing these data structures which may be called by the bootload and channel scripts.

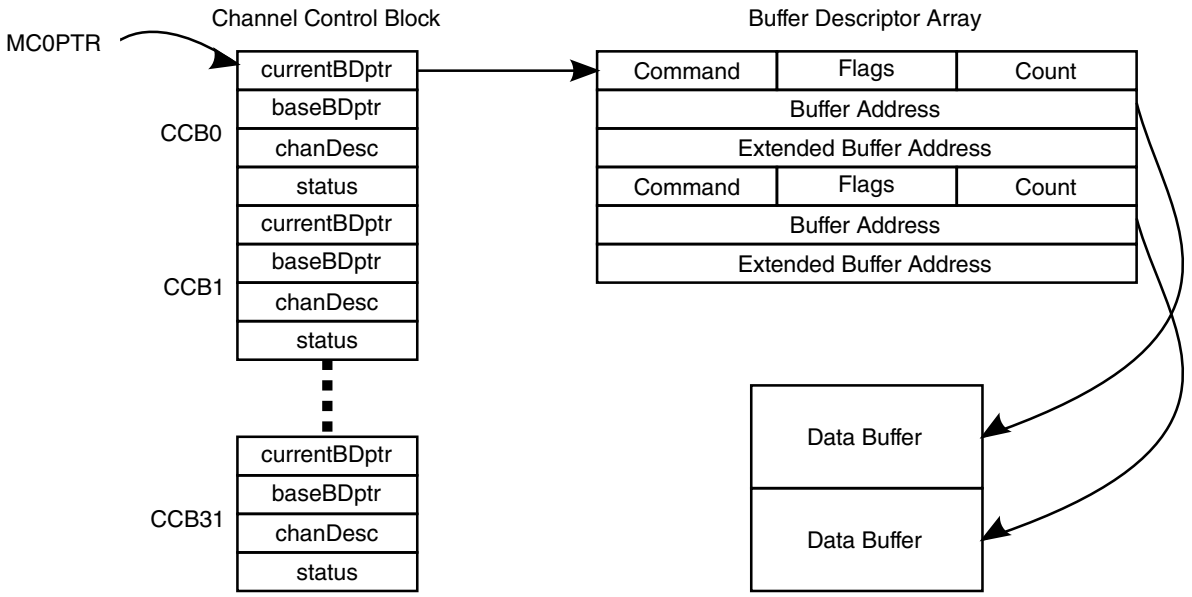


Figure 7-16. Data Structures Layout

The previous figure shows an example how these data structures are linked to pass command and pointers to data buffers. The SDMA's MC0PTR register holds the base address of the Channel 0 Control Block (CCB0). The Channel 0 control block holds a pointer to the array of buffer descriptors. The buffer descriptors are used to tell the channel 0 (boot channel) what to do as described Buffer Descriptor Format.

7.2.6.1.1 Buffer Descriptor Format

Buffer descriptors are three longs (32-bit words) in size as, shown in the figure found here.

A buffer descriptor describes the properties of the data buffer it points to. The buffer descriptors can be used for linear or circular data buffers in the Arm platform processor memory. The CCB contains a pointer to the base BD as well as the current BD.

Table 7-54. Buffer Descriptor

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Command								-	-	L	R	I	C	W	D	Count															
Buffer Address																															
Extended Buffer Address																															

Table 7-55. Buffer Descriptor Field Descriptions

Field	Description
31-24 Command	Command. The command field is used to differentiate operations performed within a script when the script accesses this particular buffer descriptor. The use of this field can be defined by the script. The command values defined for the bootload script are defined in Buffer Descriptor Commands for Bootload scripts. Refer to the individual script definition in script library documents in SDMA Scripts for command field definitions for other scripts.
23	Reserved
22	Reserved
21 L	Last Buffer Descriptor: This bit is set in SDMA IPC scripts to indicate to the receiving Core that the transfer has ended. Whenever the source finishes transferring the count it wanted to transfer, it sets LAST_BIT in the destination BD, to let the destination know that transfer is over.This bit also tells the destination software that when it processes the destination BDs, they need not process any BD after the BD with the LAST_BIT set.For example, when the DSP prepares a single buffer descriptor with count equals to 25 and Arm platform prepares a single buffer descriptor with count equals 100. When 25 bytes have been transferred from DSP to Arm platform, the DSP buffer descriptor is normally closed while the Arm platform buffer descriptor will have the L bit set and the byte count updated to 25.
20 R	erroR. Indicates an error occurred on the channel's buffer descriptor requested command. Some scripts may overwrite the command field with an error code indicating the source of the error. 0 No Error 1 Error
19 I	Interrupt. When SDMA has finished to process data transfer attached to this buffer descriptor, send an interrupt to the Arm platform. 0 No Interrupt 1 Interrupt the processor when BD is complete
18 C	Continuous. This buffer is allowed to receive multiple transmit buffers or is allowed to transmit to multiple receive buffers.The Continuous bit is decoded at the end of the processing of a BD to determine if the SDMA script must open a new BD to potentially continue the data transfer. 0 No further buffer descriptors 1 SDMA should move to the next Buffer descriptor after this one
17 W	Wrap. Indicates if this buffer descriptor is the last one for the channel control block. When encountering this bit set, the SDMA scripts updates the CurrentBD pointer to point to the first Buffer Descriptor of the array. This bit is set if the Arm platform wants to organize the array of BD in a circular way (like a ring). When all BD have been processed and if Wrap bit and CONtinuous bit are set in the last BD, the SDMA script will wrap around and it will try to re-open the first BD. 0 No Error 1 Wrap to first buffer descriptor after this one is processed.
16 D	D - "Done": bit 16: indicates the "ownership" of the buffer descriptor. When D=0 the host owns the buffer descriptor; when D=1 SDMA owns the buffer descriptor. In the case of the channel 0, D=1 indicates the SDMA has not yet processed this buffer, D=0 indicates the SDMA has processed this buffer. 0 Arm platform owns the buffer. 1 SDMA owns the buffer
15-0 Count	Count. the count field (bit 15-0) indicates the size of the data to be transmitted, the size of the data buffer pointed to by the buffer descriptor. The SDMA memory structure is different for program memory (16-bits shorts/half-words) and data memory (32-bits long). For channel 0 buffer descriptors, Count is expressed in 16-bit half-words when PM is addressed and in 32-bit words when DM is addressed. Count is typically expressed in bytes for other channel scripts, but the unit is dependant on the script.
31-0	Buffer address. Address pointer to the data buffer.
31-0	Extended buffer address. Additional pointer or other information required by some scripts.

The buffer descriptors form an array of programmable size. If the last buffer descriptor is marked by the Wrap flag-bit W=1, the array of buffer descriptor is treated as a ring with some logically continuous portion owned by the Arm platform with D=0, and the remainder owned by the SDMA with D=1. The count field of the buffer descriptor indicates how much data has been transmitted.

If Arm platform has prepared 3 buffers to be filled by the SDMA script, it has also prepared 3 BD, one for each buffer. The *Cont* and *Wrap* bits are used to organize the buffers in a circular way. For example, *CONT*inous bit is set to 1 in the 2 first BDs and *Wrap* is set in the 3rd BD. The SDMA script opens and processes BD#1. Since *CONT*inous bit is set for this BD, the SDMA will open the second BD and it will process it. Each time a BD is processed, its *Done* bit is reset by the SDMA. After the 3rd BD, if *CONT*inous is not set but if *Wrap* is set, the SDMA script stops here and the next time the channel will be triggered, the script will open the BD pointed by the currentBDptr pointer of the CCB and it will correspond to the first buffer descriptor.

If the *CONT*inous bit and *Wrap* bits are both set in the 3rd BD, the script will close it and it will try to open the first BD. An error may occur at this point if the BD#1 has already been processed and its *Done* bit is 0. The SDMA script cannot process a BD with a *Done* bit to 0. It means the BD is not ready to be processed. To avoid this situation, the *CONT*inous bit should not be set for the last BD if *Wrap* is set, and the Interrupt flag must set for the last BD. It will warn the owner of the BD that all the BDs have been processed and it has to re-set to 1 the *Done* bit of all the BD's if it desires the SDMA to fill them again. Basically, if the Arm platform expects the SDMA to fill up the buffers in a circular fashion, then it's the responsibility of the Arm platform to set the *Done* bit of a buffer descriptor at an appropriate time.

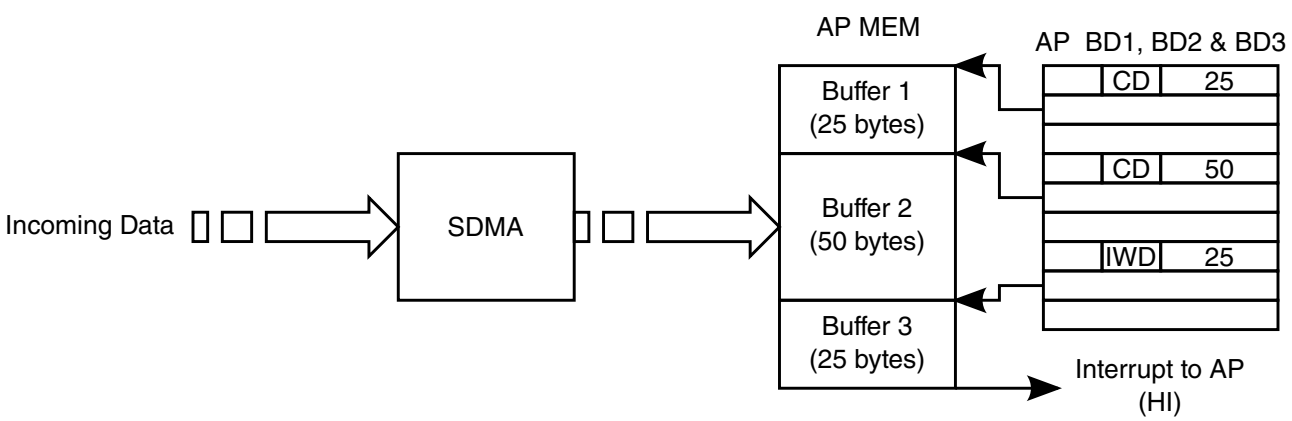


Figure 7-17. Buffer Descriptor Flow

The previous figure shows an example buffer descriptor flow. When the incoming data is stored and fills the first buffer of 25 bytes, the SDMA script opens the second BD because the CONTinuous bit was set. Then next incoming data is put in the second buffer. After receiving 50 bytes, the second buffer descriptor is also closed. The Done bit is reset and the third BD is opened. After receiving another 25 bytes, the third buffer is full and an interrupt is sent to the Arm platform because the Interrupt flag is set in the 3rd BD. The CONTinuous flag is not present the transfer is over. The next time the script will be triggered, the BD to be opened will be the first buffer descriptor since the Wrap flag was set in the 3rd BD. It is the Arm platform responsibility to set the Done bit of all the BD if it wants to use the same buffers.

7.2.6.1.2 Buffer Descriptor Commands for Bootload scripts

The command field of the buffer descriptor is defined separately for each script.

The following table lists the buffer descriptor commands defined for the channel 0 bootload script.

Table 7-56. Channel Zero Buffer Descriptor Commands

Command Field (binary)	Command	Description	Buffer Address	Extended Buffer Address
0000_0001 (0x01)	C0_SET_DM	Load SDMA data memory (RAM) from Arm platform memory buffer	Arm platform memory source address	SDMA memory destination address
0000_0010 (0x02)	C0_GET_DM	Copy SDMA data memory (RAM) to Arm platform memory buffer	Arm platform memory destination address	SDMA memory source address
0000_0100 (0x04)	C0_SET_PM	Load SDMA program memory (RAM) from Arm platform memory buffer	Arm platform memory source address	SDMA memory destination address
0000_0110 (0x06)	C0_GET_PM	Copy SDMA program memory (RAM) to Arm platform memory buffer	Arm platform memory destination address	SDMA memory source address
cccc_c111 (0x07 | CHN)	C0_SETCTX	Load Context for channel cccc into SDMA RAM from Arm platform memory buffer	Arm Platform memory source address	-
cccc_c011 (0x03 | CHN)	C0_GETCTXT	Copy Context for channel ccccc from SDMA RAM to Arm platform memory buffer	Arm platform memory destination address	-

The Channel 0 bootload commands are summarized as follows:

- C0_SET_[PM-DM]: load the buffer descriptor data in the SDMA local memory at the address pointed to by the "extended buffer address" field. The SDMA RAM can be seen as a Program Memory (PM, 16-bit address) or Data Memory (DM 32-bit address). When C0_SET_PM is used, the count field is expressed in "shorts" (16-bit half words), this command can be used to download scripts. When C0_SET_DM is

used, the count field is expressed in "long" (32-bit words), this command can be used to download channel contexts to the context channel area in RAM.

- C0_GET_[PM-DM]: write to the buffer descriptor's data buffer the content of the SDMA local memory from the address pointed to by the "extended buffer address" field for the length defined by the count in the buffer descriptor. C0_GET_PM is used to dump some part of the Program Memory (may be used to dump context of a channel), therefore count is expressed in "shorts"; while C0_GET_DM is used to dump to the buffer descriptor's data buffer, so the count field is in "longs."
- C0_SETCTX: load a context into the SDMA context page area. The handling script decodes the channel number from the 5 MSB of the command field of the buffer descriptor. Using the channel number the script computes the offset of the context data pointer for the channel relative to the context page base to use as the destination address in SDMA memory. Then the C0_SET_DM command explained above is invoked to load SDMA RAM from memory. The counter indicates the size in words of the context structure.
- Command value: (in binary) cccc c111, where ccccc is the channel number (5 bits). For instance, 0x0F means set context for channel 1, 0xFF means set context for channel 31.

- C0_GETCTX: write to the buffer descriptor's data buffer the content of the SDMA context page area. The handling script decodes the channel number from the 5 MSB of the command field of the buffer descriptor. Using this channel number, the script computes the offset of the context data pointer for the channel relative to the context page base to use as the source address for the copy. Then the C0_GET_DM command explained above is invoked to copy the context to memory. The counter indicates the size in words of the context structure.
- Command value: (in binary): cccc c011, where ccccc is the channel number (5 bits). For instance, 0x03 means get context of channel 1, 0xFB means get context of channel 31.

NOTE

To download channel context, C0_SETDM and C0_SETCTXT command can be used but the second one is easier because the channel number is embedded into the command field, whereas with the C0_SETDM, the pointer to the channel context area must be written into the extended buffer address field of the buffer descriptor.

7.2.6.1.3 Example of Buffer Descriptors for Channel 0.

Figure 7-19 illustrates the buffer descriptors that must be set in Arm platform memory space, before execution of boot code, to download contexts and scripts of channels 1, 4, and 10. After boot code execution, SDMA memory will be populated with the different contexts and scripts as presented in the following figure.

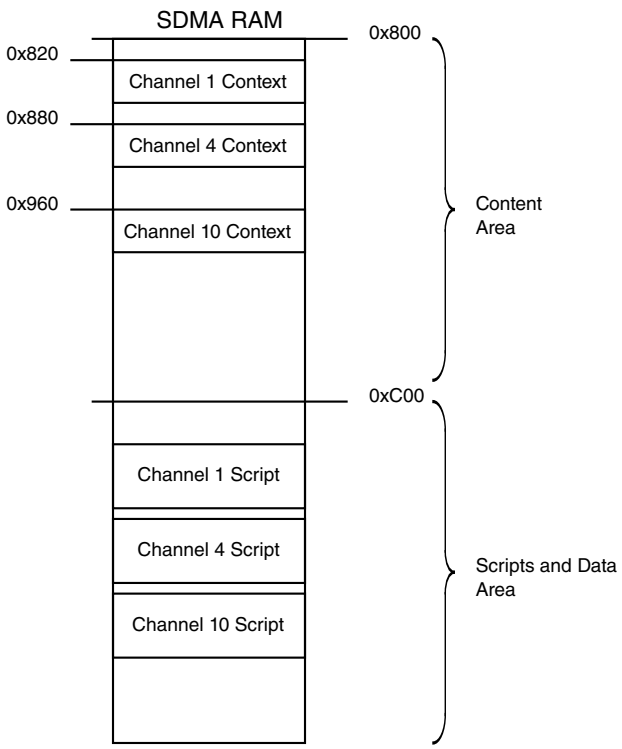
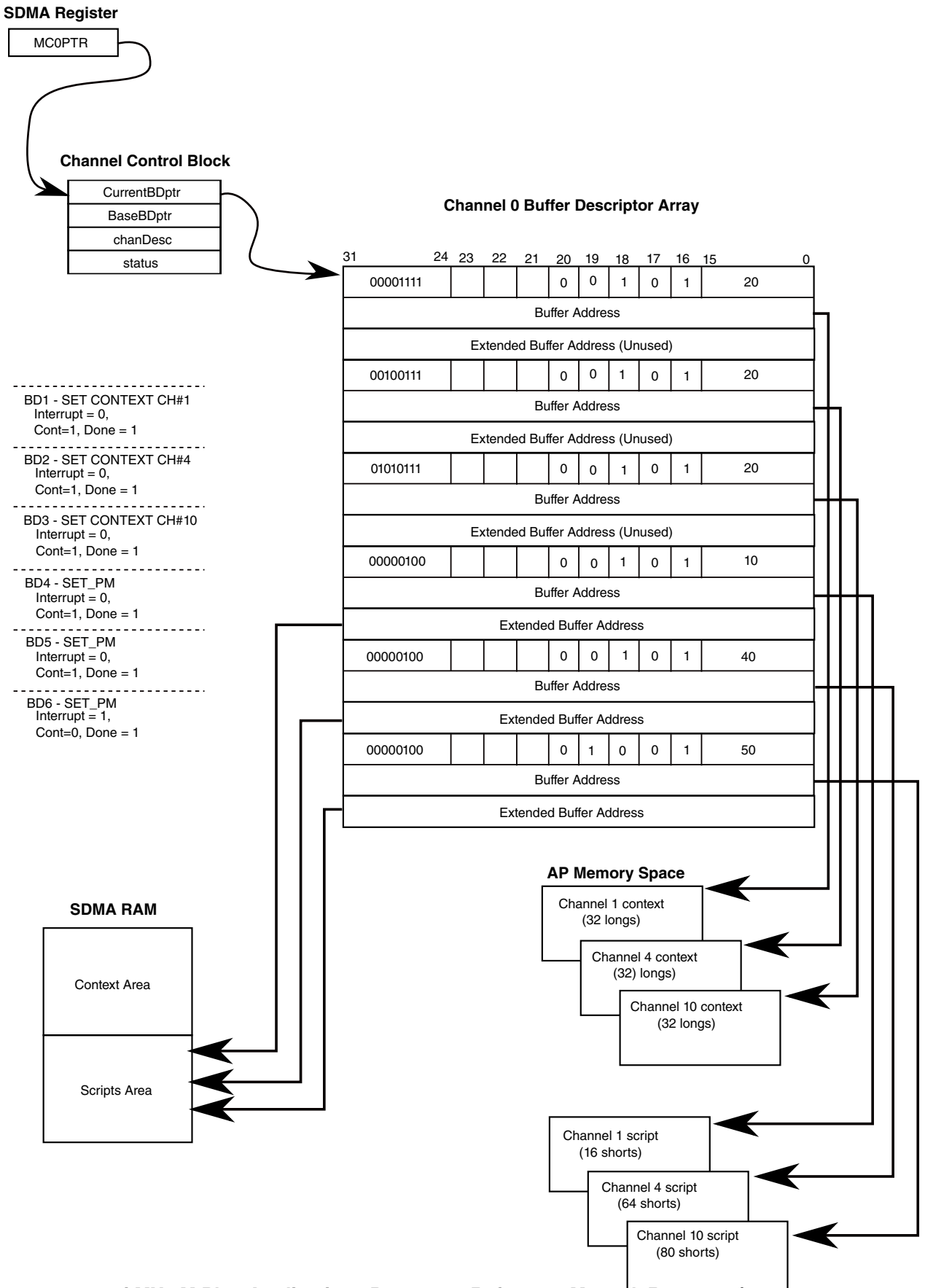


Figure 7-18. Example of SDMA RAM After Boot Session


SDMA Register
MC0PTR
Channel Control Block
CurrentBDptr
BaseBDptr
chanDesc
status
Channel 0 Buffer Descriptor Array
31 24 23 22 21 20 19 18 17 16 15 0
00001111 0 0 1 0 1 20
Buffer Address
Extended Buffer Address (Unused)
00100111 0 0 1 0 1 20
Buffer Address
Extended Buffer Address (Unused)
01010111 0 0 1 0 1 20
Buffer Address
Extended Buffer Address (Unused)
00000100 0 0 1 0 1 10
Buffer Address
Extended Buffer Address
00000100 0 0 1 0 1 40
Buffer Address
Extended Buffer Address
00000100 0 1 0 0 1 50
Buffer Address
Extended Buffer Address
BD1 - SET CONTEXT CH#1
Interrupt = 0,
Cont=1, Done = 1
BD2 - SET CONTEXT CH#4
Interrupt = 0,
Cont=1, Done = 1
BD3 - SET CONTEXT CH#10
Interrupt = 0,
Cont=1, Done = 1
BD4 - SET_PM
Interrupt = 0,
Cont=1, Done = 1
BD5 - SET_PM
Interrupt = 0,
Cont=1, Done = 1
BD6 - SET_PM
Interrupt = 1,
Cont=0, Done = 1
AP Memory Space
Channel 1 context
(32 longs)
Channel 4 context
(32) longs)
Channel 10 context
(32 longs)
SDMA RAM
Context Area
Scripts Area
Channel 1 script
(16 shorts)
Channel 4 script
(64 shorts)
Channel 10 script
(80 shorts)

7.2.6.1.4 Channel Context

There are 32 channel context memory structures pointed to by the local save area pointer. These channel context memory structures are fixed.

The script in the SDMA computes the memory offset for a given channel based on the structure length and channel number. Figure below shows the structure of the channel context as it is saved in the SDMA local memory (RAM).

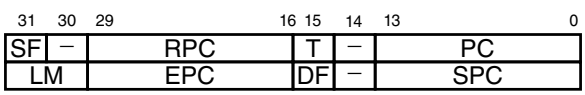
A channel context consists in 24 words, one per register. A total of 32 words are reserved for every channel. The additional 8 words are called scratch ram and they are dedicated to each channel. This memory area is commonly used for stack management.

The structure is divided in 4 areas:

- Channel status registers
- General purpose registers
- Functional units state registers reflecting the state of the Arm platform DMAs (Burst and Peripheral DMA).
- Scratch RAM

The details of the channel context status registers are described in the following figure.

The PC field of the first long register must point to the SDMA RAM address where the script that will be executed on the channel is located and this value equals the one stored in the extended buffer address of the buffer descriptor with C0_SETPM command.



SF: Source fault while loading data
RPC: Return program counter
T: Test bit: status of arithmetic and test instructions
PC: Program counter
LM: Loop mode
EPC: Loop end program counter
DF: Destination fault while storing data
SPC: Loop Start program counter

Figure 7-20. SDMA State Registers (ShPC, ShLoop)

7.2.6.2 Typical Data Transfer Supported by SDMA DMA Units

This section presents a library of SDMA scripts that perform data transfers through the peripheral DMA and the burst DMA units.

The Arm platform memory and peripherals are devices that either the peripheral DMA or the burst DMA can access. The scripts are given for a peripheral DMA whose address registers are programmed in incremented mode when internal memory is involved. See the following table for the summary.

Table 7-57. Typical Data Transfers Summary

Data Transfer	Peripheral DMA	Burst DMA	Comments
Arm platform External Memory ↔ Arm platform External Memory		3	Copy mode Script example, see Burst DMA Unit Copy Mode and External Memory to External Memory.
Arm platform Peripheral ↔ Arm platform Peripheral	3		Copy mode if same data path width Script example, see Peripheral to Peripheral Transfer.
Arm platform External Memory ↔ Arm platform Peripheral	3	3	Data transit through SDMA Script example, see Transfer Between Peripheral and External Memory.
Arm platform External Memory ↔ Arm platform Internal Memory		3	Copy mode Script example, see Transfer Between External Memory and Internal Memory.
Arm platform Internal Memory ↔ Arm platform Internal Memory		3	Copy mode Script example, see Internal Memory to Internal Memory.
Arm platform Internal memory ↔ Arm platform Peripheral	3		Data transit through SDMA Script example, see Transfer Between Peripheral and Internal Memory.

NOTE

These scripts are provided as examples of how to use DMA blocks to perform required data transfers: They are not "official" programs.

7.2.6.2.1 External Memory to External Memory

This section describes the SDMA script that performs data moves in external memory.

For this particular data transfer, only the burst DMA is used. It is programmed in copy mode, so no data transmits through an SDMA general register.

The SDMA core only monitors data transfer status. It is assumed source and destination address values are already present in two SDMA general registers (r1 and r2). For this example, it is also assumed that a 32-bit word-to-move for source-to-destination address is present in r0 and equals 64.

Data Moves in External Memory

```
1         stf r1,MSA                          // Source address setup

2         stf r2,MDA                          // Destination address setup

3         ldi r0,0x64                         // 64 words must be transferred from MSA to
MDA

4         ldi r1,0x8

MAIN_XFER:

5         cmphs r0,r1                         // Is r0 >= 0x8

6         bf LAST_XFER                      // If not, jump to last transfer label

7         stf r1,MD|CPY                        // Copy 8 words from MSA to MDA address.

8         subi r0,0x8                         // Decrement counter

9         jmp MAIN_XFER                     // return to main transfer loop

LAST_XFER:

10        stf r0,MD|CPY                        // perform last transfer
```

All instructions are performed in one cycle (jumps excepted). Instruction 7 triggers a copy transfer: A read burst access of 8-word starts, data is staged in MD and then a write burst of 8 words is executed. Instruction 8, 9, 5, and 6 are executed while the burst access is in progress. If this access is not complete when instruction 7 is executed a second time, SDMA stalls on this instruction as long as the previous copy transfer is not over. In this case, the instruction is no longer a one-cycle instruction.

During the main loop (MAIN_XFER), r1 always equals 8, so burst lengths are 8 words. On the last ldf |CPY instruction (10), r1 equals the reminder of r0 divided by 8; therefore, the length of bursts triggered in copy mode equal r1 value, which is between 1 and 7.

7.2.6.2.2 Peripheral to Peripheral Transfer

For this data transfer, only the peripheral DMA is used.

It is programmed in copy mode, so no data will transmit through the SDMA general register used in the ldf instruction, but the contents of the general register are lost. The SDMA core only monitors the transfer.

7.2.6.2.2.1 Source and Destination Target Have the Same Data Path Width

When the source and destination target have the same data path width, the following is true:

- Source target is a *half-word* (16-bit) peripheral located at address 0x1002.
- Destination is a *half-word* (16-bit) peripheral located at address 0x2006.

It is assumed the address values are already present in two SDMA general registers (r1, r2). The script for a transfer of 10 half-word is as follows:

Same Data Path Width for Source and Destination

```
//SETUP SECTION
1         stf r1, PSA|SZ16|F                    //r1=0x1002 Source address register setup
2         stf r2, PDA|SZ16|F                    //r2=0x2006 Destination address register
setup
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0xa                           //loop counter is 10
//MAIN LOOP TRANFER
copy_loop:

5         loop 2,0
6           ldf r7,PD|CPY                      //Reads 1 half-word from src and writes to
dest.
7           yield
8         bdf ERROR_DURING_XFER
ERROR_ADDR_SETUP:
                    //correction of PSA/PDA setup and jumps to main loop transfer
ERROR_DURING_XFER:
//flag error is set,
//PS can be read to know if error occurs during read or write access.
```

If a data transfer must occur between two word peripherals, only the setup section should be updated. The transfer itself is always performed by the hardware loop instruction.

All instructions are executed in one cycle (change of flow excepted). On instruction 6, a single read access is triggered, read data is staged in PD, and a write-to-destination is executed. When the transfers are in progress, the SDMA can execute he next instructions in parallel. If instruction 6, which performs the copy transfer, is executed while the previous access is not over, SDMA is stalled and instruction ldf is a multi-cycle instruction.

7.2.6.2.2.2 Source and Destination Target Have a Different Data Path Width

When the source and destination target have a different data path width, copy mode cannot be used, and any attempt to initiate a copy transfer immediately raises an error, which is stored in the SF flag.

The following example shows the SDMA code that could transfer 10 words from a *word* (32-bit) peripheral to a *half-word* peripheral whose addresses are preliminary and stored in r1 and r2.

Different Data Path Width for Source and Destination

```
//SETUP SECTION
1         stf r1, PSA|SZ32|F|PF                  //r1=0x1000 and prefetch data
2         stf r2, PDA|SZ16|F                    //r2=0x2006
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0xa                           //loop counter is 10
//MAIN LOOP TRANFER
main_loop_xfer_16_16:
5         loop 6,0
6           ldf r7,PD                         //copy 32-bit of PD in r7
7           stf r7,PD                          //store 16 LSB of r7 in PD and a flush is
executed
8           rorb r7
9           rorb r7                          //16 MSB --> 16 LSB
10          stf r7,PD                          //store 16 LSB of r6 in PD and a flush.
11          yield
```

On instruction 1, when the source address register is programmed and a data prefetch is required, a read access is executed. In parallel, the SDMA executes instructions 2 to 5. On instruction 6, the SDMA tries to read data that was fetched by instruction 1. If data is ready, the ldf will be a one cycle instruction; otherwise, the SDMA is stalled as long as the read access is not finished. Then, the 16 LSB of the read data is stored in PD and automatically flushed to the destination peripheral. In parallel, the SDMA executes the rotation instructions (8, 9), and stores the 16 MSB of the read data into PD. If a previous write access is finished, instruction 10 will be a one-cycle instruction.

The main loop transfer may appear inefficient, but due to wait states imposed to the peripheral DMA each time an external access is performed, a software pipeline is in place. During the time needed to flush PD, the SDMA executes the move and rotation operations. SDMA executes instructions in parallel with DMA accesses.

7.2.6.2.3 Transfer Between Peripheral and External Memory

7.2.6.2.3.1 Peripheral to External Memory Transfer

A transfer from a peripheral to the external memory controller involves the peripheral DMA and the burst DMA.

The code for transferring 100 word from word peripheral to the external memory would be as follows:

Peripheral to External Memory Transfer

```
//SETUP SECTION source and destination addresses are already in r1 and r2
1         stf r1, PSA|SZ16|F|PF                   //r1=0x1000 and prefetch 32-bit data
2         stf r2, MDA                           //r2=0x2000, setup burst DMA destination
address
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0x64                           //loop counter is 100
5
          //MAIN LOOP TRANFER
6         loop 3,0
7           ldf r1,PD|PF                        // read 32 bits of PD and initiate a new read
```

```
access.
8           stf r1,MD|32                    // store 32 bits of r1 in the MD fifo.
9           yield
10        ldf r1,PD                       // last word data is read
11        stf r1,MD|32|FL                   // to flush all remaining bytes of MD
```

On instruction 1, the source address register of the peripheral DMA is programmed and data is fetched. This data is stored in PD and the SDMA reads PD during instruction 7, which is a one-cycle instruction that is read-access finished. On the same instruction (7), a data prefetch is required and a read access to the source peripheral is executed. In parallel, the SDMA stored the previous read data into the data register of MD. When MD (which is an eight-word FIFO) is full, a burst write access is executed to empty the FIFO. As long as the next SDMA instructions do not access the burst DMA, they will be one-cycle instructions. The following figures show how the peripheral DMA and burst DMA work in parallel.

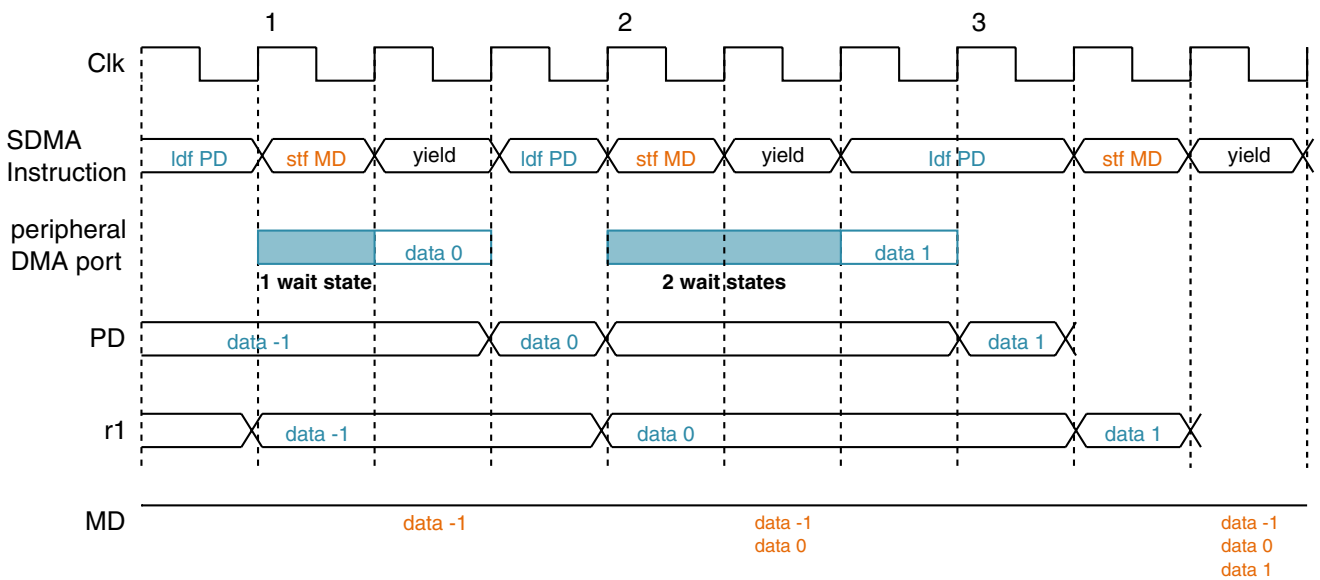


Figure 7-21. Peripheral to External Memory Example (1)

As seen in the figure above, the read access triggered by the ldf PD instruction is symbolized by the blue bar when in progress. After wait states, the read data (data 0, data 1) is stored in PD on the clk rising edge. On edge 2, data 0 is available in PD so it can be transferred to the SDMA general register r1, and then stored in MD FIFO. On edge 3, data 1 is not in PD; therefore, SDMA is stalled on the ldf instruction, which lasts two cycles. The figure below shows an example of when MD FIFO is full with data.

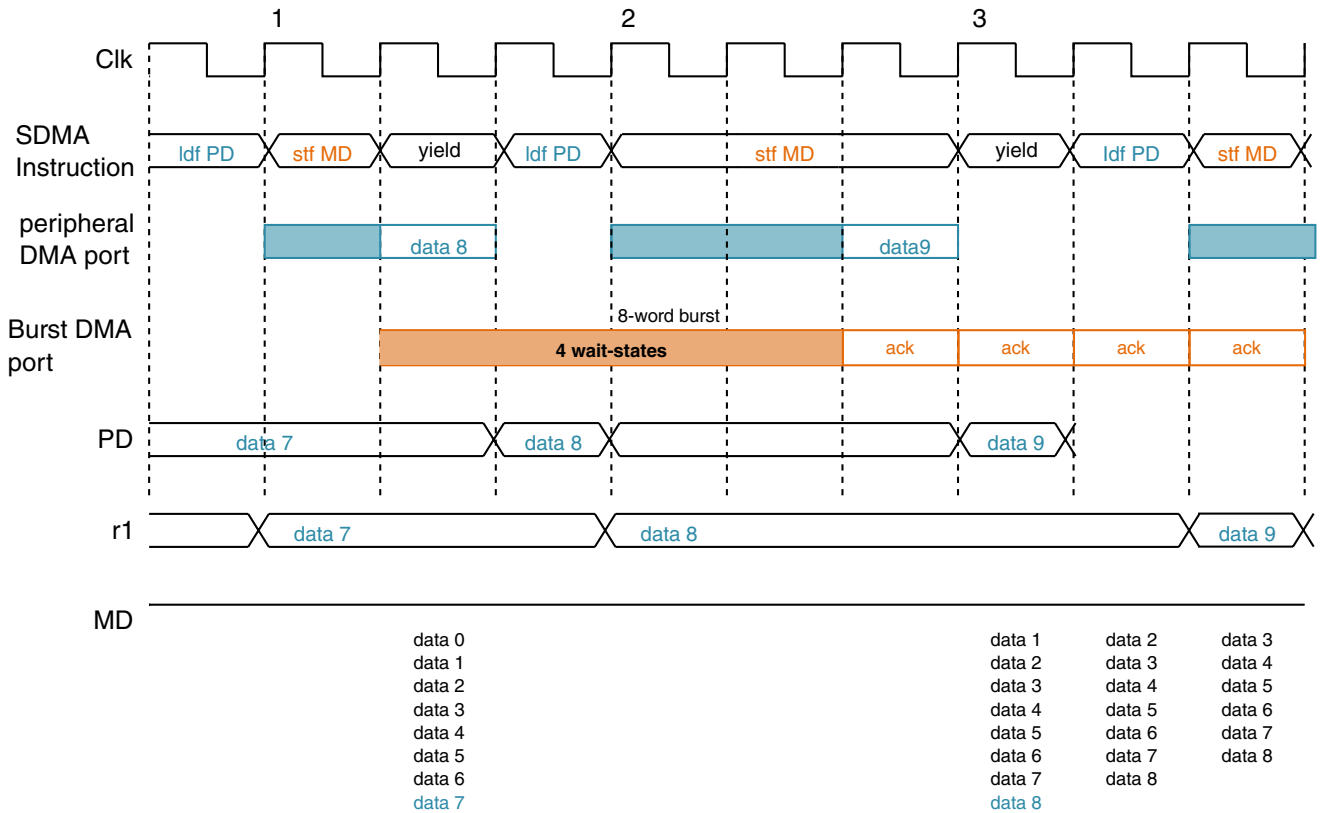


Figure 7-22. Peripheral to External Memory Example (2)

In the previous figure, the write bar means the burst DMA is performing a write burst access. The latency to have the first write acknowledge is four cycles. SDMA is stalled on instruction stf because no acknowledge was received, MD FIFO is full, and there is no empty slot to store data 9. When an acknowledge is sampled by the burst DMA, FIFO is shifted and data 8 is written. As long as there is at least one empty slot in MD FIFO, the stf MD instruction lasts one cycle.

7.2.6.2.3.2 External Memory to Peripheral Transfer

A transfer from the external memory to a peripheral involves the peripheral DMA and the burst DMA.

The code for transferring 100 word from external memory to a word peripheral would be as follows:

External Memory to Peripheral Transfer

```
//SETUP SECTION source and destination addresses are already in r1 and r2
1         stf r1, MSA|PF             //r1=0x1000 and starts a 8-word read burst
2         stf r2, PDA|SZ32|P          //r2=0x2010, setup peripheral DMA destination address
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0x64               //loop counter is 100
//MAIN LOOP TRANFER
```

```
6          loop 3,0
7            ldf r1,MD|32|PF          // read 32 bits of MD and initiate a new read access
                                   // if MD is empty after this reading.
8            stf r1,PD              // store 32 bits of r1 in the PD.
9            yield
           10 ldf r1,MD|32            // last word data is read
11         stf r1,PD                 // last write access
```

On instruction 1, a read burst of 8 words begins. Read data is staged into MD. On instruction 7 (and if data is available in MD), 32 bits are copied into r1. Then instruction 8 writes them into PD and an automatic flush is executed. The SDMA core, peripheral DMA, and burst DMA can work in parallel as long as no SDMA instruction tries to start a new write access on the peripheral DMA while the previous access is still in progress, or as long as there is data in MD when the SDMA tries to read it.

7.2.6.2.4 Transfer Between External Memory and Internal Memory

Since the internal memory (Arm platform RAM) is accessed via the peripheral DMA and the external memory is accessed via the burst DMA, the SDMA scripts that are described in Transfer Between Peripheral and External Memory can be reused. The exception is that the peripheral DMA address registers (PSA or PDA, depending on the script) should be programmed in incremented mode rather than frozen mode.

7.2.6.2.4.1 Internal Memory to Internal Memory

The internal memory can only be accessed via the peripheral DMA, so the script described in Peripheral to Peripheral Transfer can be reused with a different programming of the peripheral DMA address registers.

7.2.6.2.4.2 Transfer Between Peripheral and Internal Memory

For this transfer, the peripheral DMA is also used in copy mode.

The SDMA script is very similar to the one described in Peripheral to Peripheral Transfer, except for the peripheral DMA address registers programming.

7.2.7 Arm Platform Memory Map and Control Register Definitions

The Arm platform controls the SDMA by means of several interface registers. Those registers are described in the current section.

All registers are clocked with the SDMA clock (which means the Arm platform must ensure that the SDMA clock is running when it wants to access any register).

SDMAARM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_0000	Arm platform Channel 0 Pointer (SDMAARM1_MC0PTR)	32	R/W	0000_0000h	7.2.7.1/1220
30BD_0004	Channel Interrupts (SDMAARM1_INTR)	32	w1c	0000_0000h	7.2.7.2/1220
30BD_0008	Channel Stop/Channel Status (SDMAARM1_STOP_STAT)	32	w1c	0000_0000h	7.2.7.3/1220
30BD_000C	Channel Start (SDMAARM1_HSTART)	32	R/W	0000_0000h	7.2.7.4/1221
30BD_0010	Channel Event Override (SDMAARM1_EVTOVR)	32	R/W	0000_0000h	7.2.7.5/1221
30BD_0014	Channel BP Override (SDMAARM1_DSPOVR)	32	R/W	FFFF_FFFFh	7.2.7.6/1222
30BD_0018	Channel Arm platform Override (SDMAARM1_HOSTOVR)	32	R/W	0000_0000h	7.2.7.7/1222
30BD_001C	Channel Event Pending (SDMAARM1_EVTPEND)	32	w1c	0000_0000h	7.2.7.8/1222
30BD_0024	Reset Register (SDMAARM1_RESET)	32	R	0000_0000h	7.2.7.9/1223
30BD_0028	DMA Request Error Register (SDMAARM1_EVTERR)	32	R	0000_0000h	7.2.7.10/ 1224
30BD_002C	Channel Arm platform Interrupt Mask (SDMAARM1_INTRMASK)	32	R/W	0000_0000h	7.2.7.11/ 1224
30BD_0030	Schedule Status (SDMAARM1_PSW)	32	R	0000_0000h	7.2.7.12/ 1225
30BD_0034	DMA Request Error Register (SDMAARM1_EVTERRDBG)	32	R	0000_0000h	7.2.7.13/ 1225
30BD_0038	Configuration Register (SDMAARM1_CONFIG)	32	R/W	0000_0003h	7.2.7.14/ 1226
30BD_003C	SDMA LOCK (SDMAARM1_SDMA_LOCK)	32	R/W	0000_0000h	7.2.7.15/ 1227
30BD_0040	OnCE Enable (SDMAARM1_ONCE_ENB)	32	R/W	0000_0000h	7.2.7.16/ 1228
30BD_0044	OnCE Data Register (SDMAARM1_ONCE_DATA)	32	R/W	0000_0000h	7.2.7.17/ 1228
30BD_0048	OnCE Instruction Register (SDMAARM1_ONCE_INSTR)	32	R/W	0000_0000h	7.2.7.18/ 1229
30BD_004C	OnCE Status Register (SDMAARM1_ONCE_STAT)	32	R	0000_E000h	7.2.7.19/ 1229

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_0050	OnCE Command Register (SDMAARM1_ONCE_CMD)	32	R/W	0000_0000h	7.2.7.20/ 1231
30BD_0058	Illegal Instruction Trap Address (SDMAARM1_ILLINSTADDR)	32	R/W	0000_0001h	7.2.7.21/ 1231
30BD_005C	Channel 0 Boot Address (SDMAARM1_CHN0ADDR)	32	R/W	0000_0050h	7.2.7.22/ 1232
30BD_0060	DMA Requests (SDMAARM1_EVT_MIRROR)	32	R	0000_0000h	7.2.7.23/ 1233
30BD_0064	DMA Requests 2 (SDMAARM1_EVT_MIRROR2)	32	R	0000_0000h	7.2.7.24/ 1233
30BD_0070	Cross-Trigger Events Configuration Register 1 (SDMAARM1_XTRIG_CONF1)	32	R/W	0000_0000h	7.2.7.25/ 1234
30BD_0074	Cross-Trigger Events Configuration Register 2 (SDMAARM1_XTRIG_CONF2)	32	R/W	0000_0000h	7.2.7.26/ 1235
30BD_0100	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI0)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0104	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI1)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0108	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI2)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_010C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI3)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0110	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI4)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0114	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI5)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0118	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI6)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_011C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI7)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0120	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI8)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0124	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI9)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0128	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI10)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_012C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI11)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0130	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI12)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0134	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI13)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0138	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI14)	32	R/W	0000_0000h	7.2.7.27/ 1236

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_013C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI15)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0140	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI16)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0144	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI17)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0148	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI18)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_014C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI19)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0150	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI20)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0154	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI21)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0158	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI22)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_015C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI23)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0160	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI24)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0164	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI25)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0168	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI26)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_016C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI27)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0170	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI28)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0174	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI29)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0178	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI30)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_017C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI31)	32	R/W	0000_0000h	7.2.7.27/ 1236
30BD_0200	Channel Enable RAM (SDMAARM1_CHNENBL0)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0204	Channel Enable RAM (SDMAARM1_CHNENBL1)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0208	Channel Enable RAM (SDMAARM1_CHNENBL2)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_020C	Channel Enable RAM (SDMAARM1_CHNENBL3)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0210	Channel Enable RAM (SDMAARM1_CHNENBL4)	32	R/W	0000_0000h	7.2.7.28/ 1237

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_0214	Channel Enable RAM (SDMAARM1_CHNENBL5)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0218	Channel Enable RAM (SDMAARM1_CHNENBL6)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_021C	Channel Enable RAM (SDMAARM1_CHNENBL7)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0220	Channel Enable RAM (SDMAARM1_CHNENBL8)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0224	Channel Enable RAM (SDMAARM1_CHNENBL9)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0228	Channel Enable RAM (SDMAARM1_CHNENBL10)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_022C	Channel Enable RAM (SDMAARM1_CHNENBL11)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0230	Channel Enable RAM (SDMAARM1_CHNENBL12)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0234	Channel Enable RAM (SDMAARM1_CHNENBL13)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0238	Channel Enable RAM (SDMAARM1_CHNENBL14)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_023C	Channel Enable RAM (SDMAARM1_CHNENBL15)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0240	Channel Enable RAM (SDMAARM1_CHNENBL16)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0244	Channel Enable RAM (SDMAARM1_CHNENBL17)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0248	Channel Enable RAM (SDMAARM1_CHNENBL18)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_024C	Channel Enable RAM (SDMAARM1_CHNENBL19)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0250	Channel Enable RAM (SDMAARM1_CHNENBL20)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0254	Channel Enable RAM (SDMAARM1_CHNENBL21)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0258	Channel Enable RAM (SDMAARM1_CHNENBL22)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_025C	Channel Enable RAM (SDMAARM1_CHNENBL23)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0260	Channel Enable RAM (SDMAARM1_CHNENBL24)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0264	Channel Enable RAM (SDMAARM1_CHNENBL25)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0268	Channel Enable RAM (SDMAARM1_CHNENBL26)	32	R/W	0000_0000h	7.2.7.28/ 1237

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_026C	Channel Enable RAM (SDMAARM1_CHNENBL27)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0270	Channel Enable RAM (SDMAARM1_CHNENBL28)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0274	Channel Enable RAM (SDMAARM1_CHNENBL29)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0278	Channel Enable RAM (SDMAARM1_CHNENBL30)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_027C	Channel Enable RAM (SDMAARM1_CHNENBL31)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0280	Channel Enable RAM (SDMAARM1_CHNENBL32)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0284	Channel Enable RAM (SDMAARM1_CHNENBL33)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0288	Channel Enable RAM (SDMAARM1_CHNENBL34)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_028C	Channel Enable RAM (SDMAARM1_CHNENBL35)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0290	Channel Enable RAM (SDMAARM1_CHNENBL36)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0294	Channel Enable RAM (SDMAARM1_CHNENBL37)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_0298	Channel Enable RAM (SDMAARM1_CHNENBL38)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_029C	Channel Enable RAM (SDMAARM1_CHNENBL39)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_02A0	Channel Enable RAM (SDMAARM1_CHNENBL40)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_02A4	Channel Enable RAM (SDMAARM1_CHNENBL41)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_02A8	Channel Enable RAM (SDMAARM1_CHNENBL42)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_02AC	Channel Enable RAM (SDMAARM1_CHNENBL43)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_02B0	Channel Enable RAM (SDMAARM1_CHNENBL44)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_02B4	Channel Enable RAM (SDMAARM1_CHNENBL45)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_02B8	Channel Enable RAM (SDMAARM1_CHNENBL46)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_02BC	Channel Enable RAM (SDMAARM1_CHNENBL47)	32	R/W	0000_0000h	7.2.7.28/ 1237
30BD_1000	SDMA DONE0 Configuration (SDMAARM1_DONE0_CONFIG)	32	R/W	1F1F_1F1Fh	7.2.7.29/ 1237

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_1004	SDMA DONE1 Configuration (SDMAARM1_DONE1_CONFIG)	32	R/W	1F1F_1F1Fh	7.2.7.30/ 1239
30E0_0000	Arm platform Channel 0 Pointer (SDMAARM3_MC0PTR)	32	R/W	0000_0000h	7.2.7.1/1220
30E0_0004	Channel Interrupts (SDMAARM3_INTR)	32	w1c	0000_0000h	7.2.7.2/1220
30E0_0008	Channel Stop/Channel Status (SDMAARM3_STOP_STAT)	32	w1c	0000_0000h	7.2.7.3/1220
30E0_000C	Channel Start (SDMAARM3_HSTART)	32	R/W	0000_0000h	7.2.7.4/1221
30E0_0010	Channel Event Override (SDMAARM3_EVTOVR)	32	R/W	0000_0000h	7.2.7.5/1221
30E0_0014	Channel BP Override (SDMAARM3_DSPOVR)	32	R/W	FFFF_FFFFh	7.2.7.6/1222
30E0_0018	Channel Arm platform Override (SDMAARM3_HOSTOVR)	32	R/W	0000_0000h	7.2.7.7/1222
30E0_001C	Channel Event Pending (SDMAARM3_EVTPEND)	32	w1c	0000_0000h	7.2.7.8/1222
30E0_0024	Reset Register (SDMAARM3_RESET)	32	R	0000_0000h	7.2.7.9/1223
30E0_0028	DMA Request Error Register (SDMAARM3_EVTERR)	32	R	0000_0000h	7.2.7.10/ 1224
30E0_002C	Channel Arm platform Interrupt Mask (SDMAARM3_INTRMASK)	32	R/W	0000_0000h	7.2.7.11/ 1224
30E0_0030	Schedule Status (SDMAARM3_PSW)	32	R	0000_0000h	7.2.7.12/ 1225
30E0_0034	DMA Request Error Register (SDMAARM3_EVTERRDBG)	32	R	0000_0000h	7.2.7.13/ 1225
30E0_0038	Configuration Register (SDMAARM3_CONFIG)	32	R/W	0000_0003h	7.2.7.14/ 1226
30E0_003C	SDMA LOCK (SDMAARM3_SDMA_LOCK)	32	R/W	0000_0000h	7.2.7.15/ 1227
30E0_0040	OnCE Enable (SDMAARM3_ONCE_ENB)	32	R/W	0000_0000h	7.2.7.16/ 1228
30E0_0044	OnCE Data Register (SDMAARM3_ONCE_DATA)	32	R/W	0000_0000h	7.2.7.17/ 1228
30E0_0048	OnCE Instruction Register (SDMAARM3_ONCE_INSTR)	32	R/W	0000_0000h	7.2.7.18/ 1229
30E0_004C	OnCE Status Register (SDMAARM3_ONCE_STAT)	32	R	0000_E000h	7.2.7.19/ 1229
30E0_0050	OnCE Command Register (SDMAARM3_ONCE_CMD)	32	R/W	0000_0000h	7.2.7.20/ 1231
30E0_0058	Illegal Instruction Trap Address (SDMAARM3_ILLINSTADDR)	32	R/W	0000_0001h	7.2.7.21/ 1231
30E0_005C	Channel 0 Boot Address (SDMAARM3_CHN0ADDR)	32	R/W	0000_0050h	7.2.7.22/ 1232
30E0_0060	DMA Requests (SDMAARM3_EVT_MIRROR)	32	R	0000_0000h	7.2.7.23/ 1233
30E0_0064	DMA Requests 2 (SDMAARM3_EVT_MIRROR2)	32	R	0000_0000h	7.2.7.24/ 1233
30E0_0070	Cross-Trigger Events Configuration Register 1 (SDMAARM3_XTRIG_CONF1)	32	R/W	0000_0000h	7.2.7.25/ 1234

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30E0_0074	Cross-Trigger Events Configuration Register 2 (SDMAARM3_XTRIG_CONF2)	32	R/W	0000_0000h	7.2.7.26/ 1235
30E0_0100	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI0)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0104	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI1)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0108	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI2)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_010C	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI3)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0110	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI4)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0114	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI5)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0118	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI6)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_011C	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI7)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0120	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI8)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0124	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI9)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0128	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI10)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_012C	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI11)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0130	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI12)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0134	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI13)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0138	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI14)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_013C	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI15)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0140	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI16)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0144	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI17)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0148	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI18)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_014C	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI19)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0150	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI20)	32	R/W	0000_0000h	7.2.7.27/ 1236

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30E0_0154	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI21)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0158	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI22)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_015C	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI23)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0160	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI24)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0164	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI25)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0168	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI26)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_016C	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI27)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0170	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI28)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0174	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI29)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0178	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI30)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_017C	Channel Priority Registers (SDMAARM3_SDMA_CHNPRI31)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E0_0200	Channel Enable RAM (SDMAARM3_CHNENBL0)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0204	Channel Enable RAM (SDMAARM3_CHNENBL1)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0208	Channel Enable RAM (SDMAARM3_CHNENBL2)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_020C	Channel Enable RAM (SDMAARM3_CHNENBL3)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0210	Channel Enable RAM (SDMAARM3_CHNENBL4)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0214	Channel Enable RAM (SDMAARM3_CHNENBL5)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0218	Channel Enable RAM (SDMAARM3_CHNENBL6)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_021C	Channel Enable RAM (SDMAARM3_CHNENBL7)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0220	Channel Enable RAM (SDMAARM3_CHNENBL8)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0224	Channel Enable RAM (SDMAARM3_CHNENBL9)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0228	Channel Enable RAM (SDMAARM3_CHNENBL10)	32	R/W	0000_0000h	7.2.7.28/ 1237

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30E0_022C	Channel Enable RAM (SDMAARM3_CHNENBL11)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0230	Channel Enable RAM (SDMAARM3_CHNENBL12)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0234	Channel Enable RAM (SDMAARM3_CHNENBL13)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0238	Channel Enable RAM (SDMAARM3_CHNENBL14)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_023C	Channel Enable RAM (SDMAARM3_CHNENBL15)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0240	Channel Enable RAM (SDMAARM3_CHNENBL16)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0244	Channel Enable RAM (SDMAARM3_CHNENBL17)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0248	Channel Enable RAM (SDMAARM3_CHNENBL18)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_024C	Channel Enable RAM (SDMAARM3_CHNENBL19)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0250	Channel Enable RAM (SDMAARM3_CHNENBL20)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0254	Channel Enable RAM (SDMAARM3_CHNENBL21)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0258	Channel Enable RAM (SDMAARM3_CHNENBL22)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_025C	Channel Enable RAM (SDMAARM3_CHNENBL23)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0260	Channel Enable RAM (SDMAARM3_CHNENBL24)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0264	Channel Enable RAM (SDMAARM3_CHNENBL25)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0268	Channel Enable RAM (SDMAARM3_CHNENBL26)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_026C	Channel Enable RAM (SDMAARM3_CHNENBL27)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0270	Channel Enable RAM (SDMAARM3_CHNENBL28)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0274	Channel Enable RAM (SDMAARM3_CHNENBL29)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0278	Channel Enable RAM (SDMAARM3_CHNENBL30)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_027C	Channel Enable RAM (SDMAARM3_CHNENBL31)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0280	Channel Enable RAM (SDMAARM3_CHNENBL32)	32	R/W	0000_0000h	7.2.7.28/ 1237

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30E0_0284	Channel Enable RAM (SDMAARM3_CHNENBL33)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0288	Channel Enable RAM (SDMAARM3_CHNENBL34)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_028C	Channel Enable RAM (SDMAARM3_CHNENBL35)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0290	Channel Enable RAM (SDMAARM3_CHNENBL36)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0294	Channel Enable RAM (SDMAARM3_CHNENBL37)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_0298	Channel Enable RAM (SDMAARM3_CHNENBL38)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_029C	Channel Enable RAM (SDMAARM3_CHNENBL39)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_02A0	Channel Enable RAM (SDMAARM3_CHNENBL40)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_02A4	Channel Enable RAM (SDMAARM3_CHNENBL41)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_02A8	Channel Enable RAM (SDMAARM3_CHNENBL42)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_02AC	Channel Enable RAM (SDMAARM3_CHNENBL43)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_02B0	Channel Enable RAM (SDMAARM3_CHNENBL44)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_02B4	Channel Enable RAM (SDMAARM3_CHNENBL45)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_02B8	Channel Enable RAM (SDMAARM3_CHNENBL46)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_02BC	Channel Enable RAM (SDMAARM3_CHNENBL47)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E0_1000	SDMA DONE0 Configuration (SDMAARM3_DONE0_CONFIG)	32	R/W	1F1F_1F1Fh	7.2.7.29/ 1237
30E0_1004	SDMA DONE1 Configuration (SDMAARM3_DONE1_CONFIG)	32	R/W	1F1F_1F1Fh	7.2.7.30/ 1239
30E1_0000	Arm platform Channel 0 Pointer (SDMAARM2_MC0PTR)	32	R/W	0000_0000h	7.2.7.1/1220
30E1_0004	Channel Interrupts (SDMAARM2_INTR)	32	w1c	0000_0000h	7.2.7.2/1220
30E1_0008	Channel Stop/Channel Status (SDMAARM2_STOP_STAT)	32	w1c	0000_0000h	7.2.7.3/1220
30E1_000C	Channel Start (SDMAARM2_HSTART)	32	R/W	0000_0000h	7.2.7.4/1221
30E1_0010	Channel Event Override (SDMAARM2_EVTOVR)	32	R/W	0000_0000h	7.2.7.5/1221
30E1_0014	Channel BP Override (SDMAARM2_DSPOVR)	32	R/W	FFFF_FFFFh	7.2.7.6/1222
30E1_0018	Channel Arm platform Override (SDMAARM2_HOSTOVR)	32	R/W	0000_0000h	7.2.7.7/1222
30E1_001C	Channel Event Pending (SDMAARM2_EVTPEND)	32	w1c	0000_0000h	7.2.7.8/1222
30E1_0024	Reset Register (SDMAARM2_RESET)	32	R	0000_0000h	7.2.7.9/1223

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30E1_0028	DMA Request Error Register (SDMAARM2_EVTERR)	32	R	0000_0000h	7.2.7.10/ 1224
30E1_002C	Channel Arm platform Interrupt Mask (SDMAARM2_INTRMASK)	32	R/W	0000_0000h	7.2.7.11/ 1224
30E1_0030	Schedule Status (SDMAARM2_PSW)	32	R	0000_0000h	7.2.7.12/ 1225
30E1_0034	DMA Request Error Register (SDMAARM2_EVTERRDBG)	32	R	0000_0000h	7.2.7.13/ 1225
30E1_0038	Configuration Register (SDMAARM2_CONFIG)	32	R/W	0000_0003h	7.2.7.14/ 1226
30E1_003C	SDMA LOCK (SDMAARM2_SDMA_LOCK)	32	R/W	0000_0000h	7.2.7.15/ 1227
30E1_0040	OnCE Enable (SDMAARM2_ONCE_ENB)	32	R/W	0000_0000h	7.2.7.16/ 1228
30E1_0044	OnCE Data Register (SDMAARM2_ONCE_DATA)	32	R/W	0000_0000h	7.2.7.17/ 1228
30E1_0048	OnCE Instruction Register (SDMAARM2_ONCE_INSTR)	32	R/W	0000_0000h	7.2.7.18/ 1229
30E1_004C	OnCE Status Register (SDMAARM2_ONCE_STAT)	32	R	0000_E000h	7.2.7.19/ 1229
30E1_0050	OnCE Command Register (SDMAARM2_ONCE_CMD)	32	R/W	0000_0000h	7.2.7.20/ 1231
30E1_0058	Illegal Instruction Trap Address (SDMAARM2_ILLINSTADDR)	32	R/W	0000_0001h	7.2.7.21/ 1231
30E1_005C	Channel 0 Boot Address (SDMAARM2_CHN0ADDR)	32	R/W	0000_0050h	7.2.7.22/ 1232
30E1_0060	DMA Requests (SDMAARM2_EVT_MIRROR)	32	R	0000_0000h	7.2.7.23/ 1233
30E1_0064	DMA Requests 2 (SDMAARM2_EVT_MIRROR2)	32	R	0000_0000h	7.2.7.24/ 1233
30E1_0070	Cross-Trigger Events Configuration Register 1 (SDMAARM2_XTRIG_CONF1)	32	R/W	0000_0000h	7.2.7.25/ 1234
30E1_0074	Cross-Trigger Events Configuration Register 2 (SDMAARM2_XTRIG_CONF2)	32	R/W	0000_0000h	7.2.7.26/ 1235
30E1_0100	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI0)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0104	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI1)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0108	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI2)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_010C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI3)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0110	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI4)	32	R/W	0000_0000h	7.2.7.27/ 1236

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30E1_0114	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI5)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0118	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI6)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_011C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI7)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0120	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI8)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0124	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI9)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0128	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI10)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_012C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI11)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0130	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI12)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0134	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI13)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0138	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI14)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_013C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI15)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0140	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI16)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0144	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI17)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0148	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI18)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_014C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI19)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0150	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI20)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0154	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI21)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0158	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI22)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_015C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI23)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0160	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI24)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0164	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI25)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0168	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI26)	32	R/W	0000_0000h	7.2.7.27/ 1236

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30E1_016C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI27)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0170	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI28)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0174	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI29)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0178	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI30)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_017C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI31)	32	R/W	0000_0000h	7.2.7.27/ 1236
30E1_0200	Channel Enable RAM (SDMAARM2_CHNENBL0)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0204	Channel Enable RAM (SDMAARM2_CHNENBL1)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0208	Channel Enable RAM (SDMAARM2_CHNENBL2)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_020C	Channel Enable RAM (SDMAARM2_CHNENBL3)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0210	Channel Enable RAM (SDMAARM2_CHNENBL4)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0214	Channel Enable RAM (SDMAARM2_CHNENBL5)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0218	Channel Enable RAM (SDMAARM2_CHNENBL6)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_021C	Channel Enable RAM (SDMAARM2_CHNENBL7)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0220	Channel Enable RAM (SDMAARM2_CHNENBL8)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0224	Channel Enable RAM (SDMAARM2_CHNENBL9)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0228	Channel Enable RAM (SDMAARM2_CHNENBL10)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_022C	Channel Enable RAM (SDMAARM2_CHNENBL11)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0230	Channel Enable RAM (SDMAARM2_CHNENBL12)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0234	Channel Enable RAM (SDMAARM2_CHNENBL13)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0238	Channel Enable RAM (SDMAARM2_CHNENBL14)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_023C	Channel Enable RAM (SDMAARM2_CHNENBL15)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0240	Channel Enable RAM (SDMAARM2_CHNENBL16)	32	R/W	0000_0000h	7.2.7.28/ 1237

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30E1_0244	Channel Enable RAM (SDMAARM2_CHNENBL17)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0248	Channel Enable RAM (SDMAARM2_CHNENBL18)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_024C	Channel Enable RAM (SDMAARM2_CHNENBL19)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0250	Channel Enable RAM (SDMAARM2_CHNENBL20)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0254	Channel Enable RAM (SDMAARM2_CHNENBL21)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0258	Channel Enable RAM (SDMAARM2_CHNENBL22)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_025C	Channel Enable RAM (SDMAARM2_CHNENBL23)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0260	Channel Enable RAM (SDMAARM2_CHNENBL24)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0264	Channel Enable RAM (SDMAARM2_CHNENBL25)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0268	Channel Enable RAM (SDMAARM2_CHNENBL26)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_026C	Channel Enable RAM (SDMAARM2_CHNENBL27)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0270	Channel Enable RAM (SDMAARM2_CHNENBL28)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0274	Channel Enable RAM (SDMAARM2_CHNENBL29)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0278	Channel Enable RAM (SDMAARM2_CHNENBL30)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_027C	Channel Enable RAM (SDMAARM2_CHNENBL31)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0280	Channel Enable RAM (SDMAARM2_CHNENBL32)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0284	Channel Enable RAM (SDMAARM2_CHNENBL33)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0288	Channel Enable RAM (SDMAARM2_CHNENBL34)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_028C	Channel Enable RAM (SDMAARM2_CHNENBL35)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0290	Channel Enable RAM (SDMAARM2_CHNENBL36)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0294	Channel Enable RAM (SDMAARM2_CHNENBL37)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_0298	Channel Enable RAM (SDMAARM2_CHNENBL38)	32	R/W	0000_0000h	7.2.7.28/ 1237

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30E1_029C	Channel Enable RAM (SDMAARM2_CHNENBL39)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_02A0	Channel Enable RAM (SDMAARM2_CHNENBL40)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_02A4	Channel Enable RAM (SDMAARM2_CHNENBL41)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_02A8	Channel Enable RAM (SDMAARM2_CHNENBL42)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_02AC	Channel Enable RAM (SDMAARM2_CHNENBL43)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_02B0	Channel Enable RAM (SDMAARM2_CHNENBL44)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_02B4	Channel Enable RAM (SDMAARM2_CHNENBL45)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_02B8	Channel Enable RAM (SDMAARM2_CHNENBL46)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_02BC	Channel Enable RAM (SDMAARM2_CHNENBL47)	32	R/W	0000_0000h	7.2.7.28/ 1237
30E1_1000	SDMA DONE0 Configuration (SDMAARM2_DONE0_CONFIG)	32	R/W	1F1F_1F1Fh	7.2.7.29/ 1237
30E1_1004	SDMA DONE1 Configuration (SDMAARM2_DONE1_CONFIG)	32	R/W	1F1F_1F1Fh	7.2.7.30/ 1239

7.2.7.1 Arm platform Channel 0 Pointer (SDMAARMx_MC0PTR)

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MC0PTR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_MC0PTR field descriptions

Field	Description
MC0PTR	**Channel 0 Pointer** contains the 32-bit address, in Arm platform memory, of channel 0 control block (the boot channel). Appendix A fully describes the SDMA Application Programming Interface (API). The Arm platform has a read/write access and the SDMA has a read-only access.

7.2.7.2 Channel Interrupts (SDMAARMx_INTR)

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HI[31:0]																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_INTR field descriptions

Field	Description
HI[31:0]	The Arm platform Interrupts register contains the 32 HI[i] bits. If any bit is set, it will cause an interrupt to the Arm platform. This register is a "write-ones" register to the Arm platform. When the Arm platform sets a bit in this register the corresponding HI[i] bit is cleared. The interrupt service routine should clear individual channel bits when their interrupts are serviced, failure to do so will cause continuous interrupts. The SDMA is responsible for setting the HI[i] bit corresponding to the current channel when the corresponding `done` instruction is executed.

7.2.7.3 Channel Stop/Channel Status (SDMAARMx_STOP_STAT)

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HE																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_STOP_STAT field descriptions

Field	Description
HE	This 32-bit register gives access to the Arm platform Enable bits. There is one bit for every channel. This register is a "write-ones" register to the Arm platform. When the Arm platform writes 1 in bit i of this register, it clears the HE[i] and HSTART[i] bits. Reading this register yields the current state of the HE[i] bits.

7.2.7.4 Channel Start (SDMAARMx_HSTART)

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HSTART_HE																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_HSTART field descriptions

Field	Description
HSTART_HE	The HSTART_HE registers are 32 bits wide with one bit for every channel. When a bit is written to 1, it enables the corresponding channel. Two physical registers are accessed with that address (HSTART and HE), which enables the Arm platform to trigger a channel a second time before the first trigger is processed. • This register is a "write-ones" register to the Arm platform. Neither HSTART[i] bit can be set while the corresponding HE[i] bit is cleared. • When the Arm platform tries to set the HSTART[i] bit by writing a one (if the corresponding HE[i] bit is clear), the bit in the HSTART[i] register will remain cleared and the HE[i] bit will be set. • If the corresponding HE[i] bit was already set, the HSTART[i] bit will be set. The next time the SDMA channel *i* attempts to clear the HE[i] bit by means of a `done` instruction, the bit in the HSTART[i] register will be cleared and the HE[i] bit will take the old value of the HSTART[i] bit. • Reading this register yields the current state of the HSTART[i] bits. This mechanism enables the Arm platform to pipeline two HSTART commands per channel.

7.2.7.5 Channel Event Override (SDMAARMx_EVTOVR)

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_EVTOVR field descriptions

Field	Description
EO	The Channel Event Override register contains the 32 EO[i] bits. A bit set in this register causes the SDMA to ignore DMA requests when scheduling the corresponding channel.

7.2.7.6 Channel BP Override (SDMAARMx_DSPOVR)

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DO																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

SDMAARMx_DSPOVR field descriptions

Field	Description
DO	This register is reserved. All DO bits should be set to the reset value of 1. A setting of 0 will prevent SDMA channels from starting according to the condition described in Runnable Channels Evaluation. 0 - Reserved 1 - Reset value.

7.2.7.7 Channel Arm platform Override (SDMAARMx_HOSTOVR)

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_HOSTOVR field descriptions

Field	Description
HO	The Channel Arm platform Override register contains the 32 HO[i] bits. A bit set in this register causes the SDMA to ignore the Arm platform enable bit (HE) when scheduling the corresponding channel.

7.2.7.8 Channel Event Pending (SDMAARMx_EVTPEND)

Address: Base address + 1Ch offset

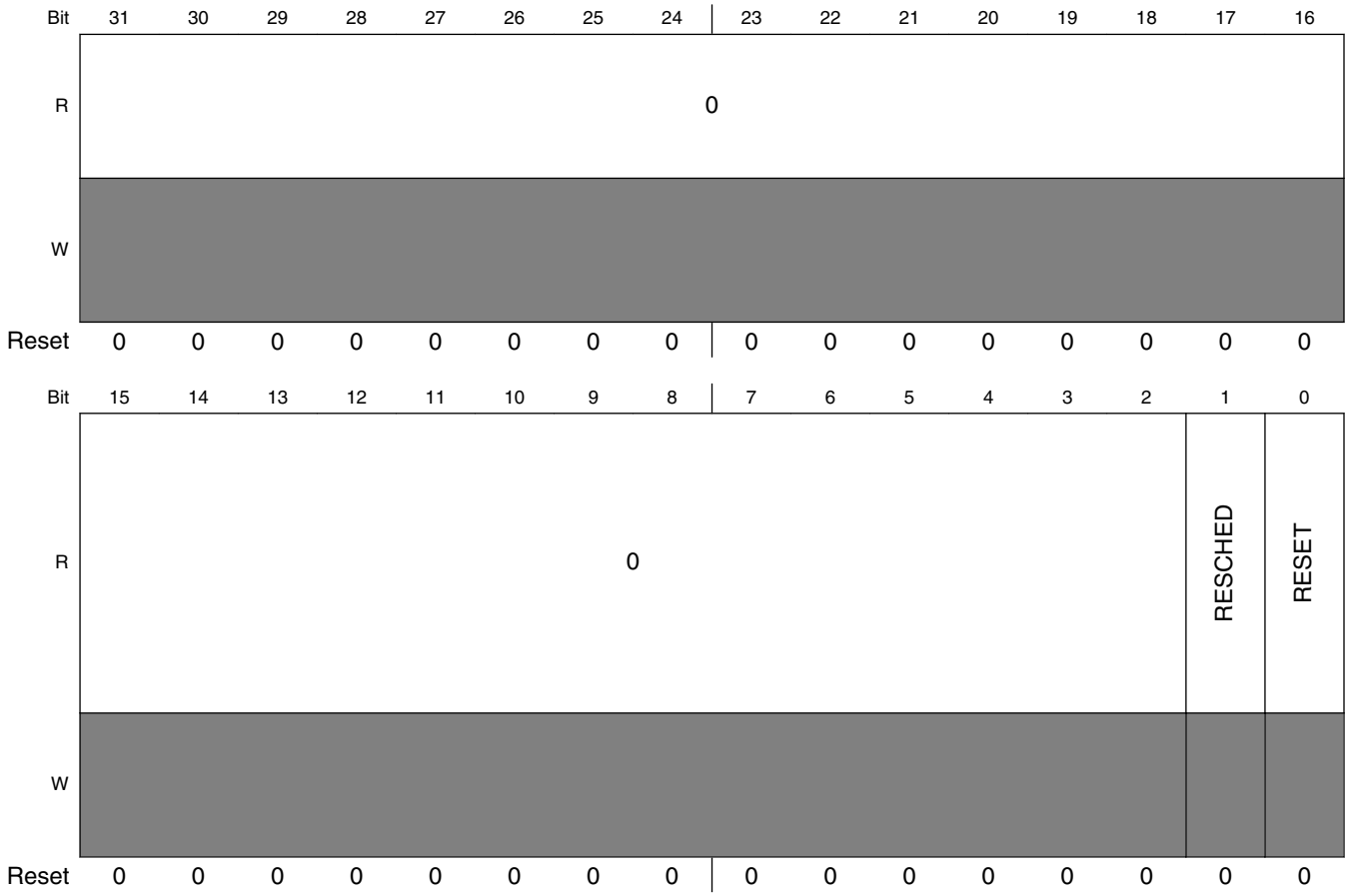
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EP																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_EVTPEND field descriptions

Field	Description
EP	The Channel Event Pending register contains the 32 EP[i] bits. Reading this register enables the Arm platform to determine what channels are pending after the reception of a DMA request. • Setting a bit in this register causes the SDMA to reevaluate scheduling as if a DMA request mapped on this channel had occurred. This is useful for starting up channels, so that initialization is done before awaiting the first request. The scheduler can also set bits in the EVTPEND register according to the received DMA requests. • The EP[i] bit may be cleared by the `done` instruction when running the channel *i* script. This a "write-ones" mechanism: Writing a '0' does not clear the corresponding bit.

7.2.7.9 Reset Register (SDMAARMx_RESET)

Address: Base address + 24h offset

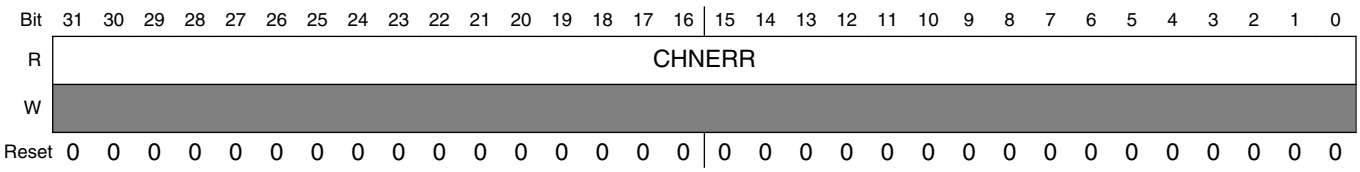


SDMAARMx_RESET field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
1 RESCHED	When set, this bit forces the SDMA to reschedule as if a script had executed a `done` instruction. This enables the Arm platform to recover from a runaway script on a channel by clearing its HE[i] bit via the STOP register, and then forcing a reschedule via the RESCHED bit. The RESCHED bit is cleared when the context switch starts.
0 RESET	When set, this bit causes the SDMA to be held in a software reset. The internal reset signal is held low 16 cycles; the RESET bit is automatically cleared when the internal reset signal rises.

7.2.7.10 DMA Request Error Register (SDMAARMx_EVTERR)

Address: Base address + 28h offset

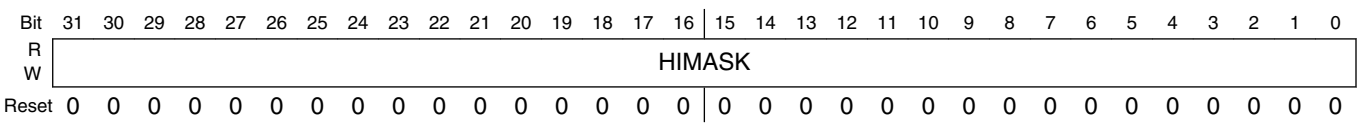


SDMAARMx_EVTERR field descriptions

Field	Description
CHNERR	This register is used by the SDMA to warn the Arm platform when an incoming DMA request was detected and it triggers a channel that is already pending or being serviced. This probably means there is an overflow of data for that channel. • An interrupt is sent to the Arm platform if the corresponding channel bit is set in the INTRMASK register. • This is a "write-ones" register for the scheduler. It is only able to set the flags. The flags are cleared when the register is read by the Arm platform or during SDMA reset. • The CHNERR[i] bit is set when a DMA request that triggers channel *i* is received through the corresponding input pins and the EP[i] bit is already set; the EVTERR[i] bit is unaffected if the Arm platform tries to set the EP[i] bit, whereas, that EP[i] bit is already set.

7.2.7.11 Channel Arm platform Interrupt Mask (SDMAARMx_INTRMASK)

Address: Base address + 2Ch offset

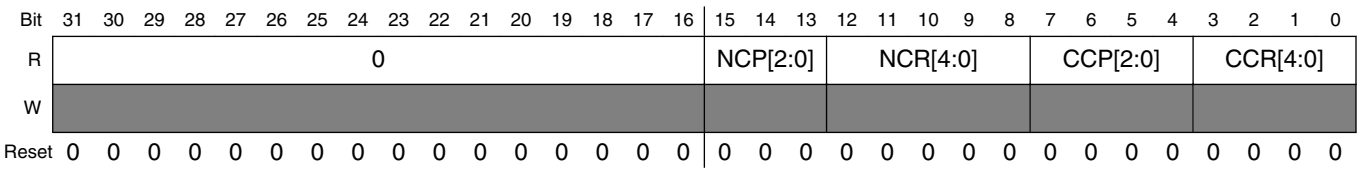


SDMAARMx_INTRMASK field descriptions

Field	Description
HIMASK	The Interrupt Mask Register contains 32 interrupt generation mask bits. If bit HIMASK[i] is set, the HI[i] bit is set and an interrupt is sent to the Arm platform when a DMA request error is detected on channel *i* (for example, EVTERR[i] is set).

7.2.7.12 Schedule Status (SDMAARMx_PSW)

Address: Base address + 30h offset

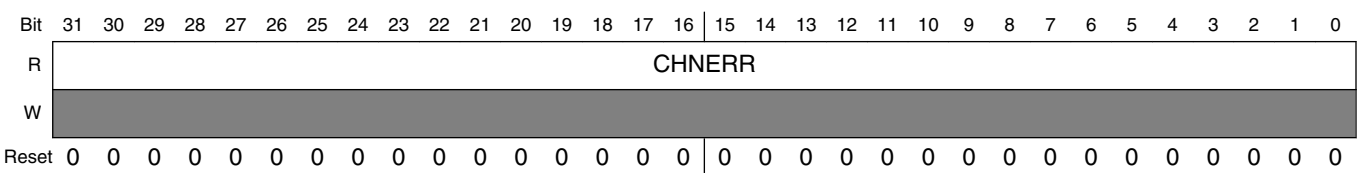


SDMAARMx_PSW field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–13 NCP[2:0]	The Next Channel Priority gives the next pending channel priority. When the priority is 0, it means there is no pending channel and the NCR value has no meaning. 0 No running channel 1 Active channel priority
12–8 NCR[4:0]	The Next Channel Register indicates the number of the next scheduled pending channel with the highest priority.
7–4 CCP[2:0]	The Current Channel Priority indicates the priority of the current active channel. When the priority is 0, no channel is running: The SDMA is idle and the CCR value has no meaning. In the case that the SDMA has finished running the channel and has entered sleep state, CCP will indicate the priority of previous running channel. 0 No running channel 1 Active channel priority
CCR[4:0]	The Current Channel Register indicates the number of the channel that is being executed by the SDMA. SDMA. In the case that the SDMA has finished running the channel and has entered sleep state, CCR will indicate the previous running channel.

7.2.7.13 DMA Request Error Register (SDMAARMx_EVTERRDBG)

Address: Base address + 34h offset

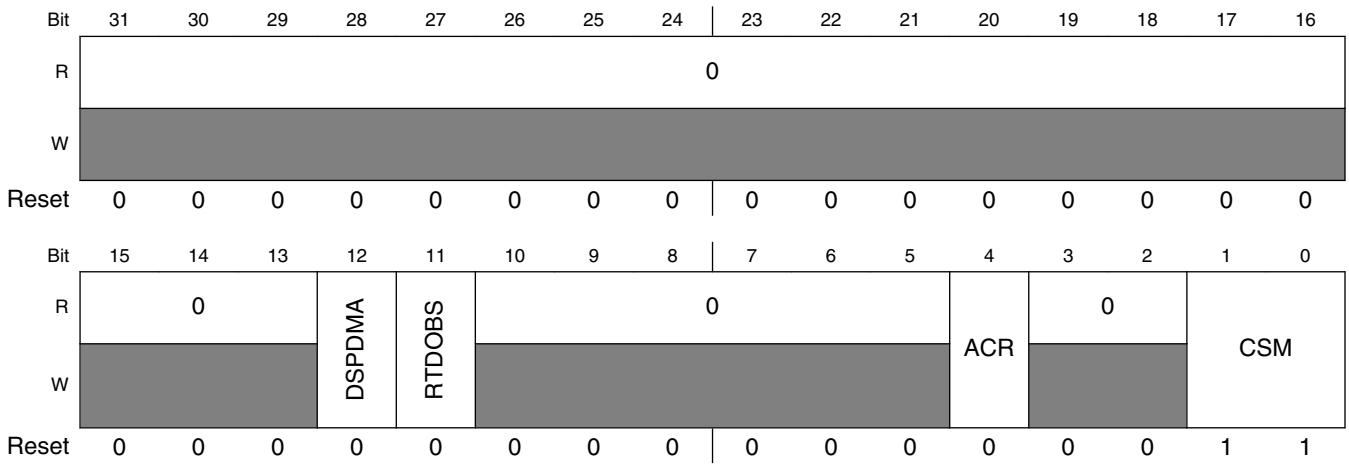


SDMAARMx_EVTERRDBG field descriptions

Field	Description
CHNERR	This register is the same as EVTERR, except reading it does not clear its contents. This address is meant to be used in debug mode. The Arm platform OnCE may check this register value without modifying it.

7.2.7.14 Configuration Register (SDMAARMx_CONFIG)

Address: Base address + 38h offset



SDMAARMx_CONFIG field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value 0.
12 DSPDMA	This bit's function is reserved and should be configured as zero. 0 - Reset Value 1 - Reserved
11 RTDOBS	Indicates if Real-Time Debug pins are used: They do not toggle by default in order to reduce power consumption. 0 RTD pins disabled 1 RTD pins enabled
10–5 Reserved	This read-only field is reserved and always has the value 0.
4 ACR	Arm platform DMA / SDMA Core Clock Ratio. Selects the clock ratio between Arm platform DMA interfaces (burst DMA and peripheral DMA) and the internal SDMA core clock. The frequency selection is determined separately by the chip clock controller. This bit has to match the configuration of the chip clock controller that generates the clocks used in the SDMA. 0 Arm platform DMA interface frequency equals twice core frequency 1 Arm platform DMA interface frequency equals core frequency
3–2 Reserved	This read-only field is reserved and always has the value 0.
CSM	Selects the Context Switch Mode. The Arm platform has a read/write access. The SDMA cannot modify that register. The value at reset is 3, which selects the dynamic context switch by default. That register can be modified at anytime but the new context switch configuration will only be taken into account at the start of the next restore phase. NOTE: The first call to SDMA's channel 0 Bootload script after reset should use static context switch mode to ensure the context RAM for channel 0 is initialized in the channel SAVE Phase. After Channel 0 is run once, then any of the dynamic context modes can be used.

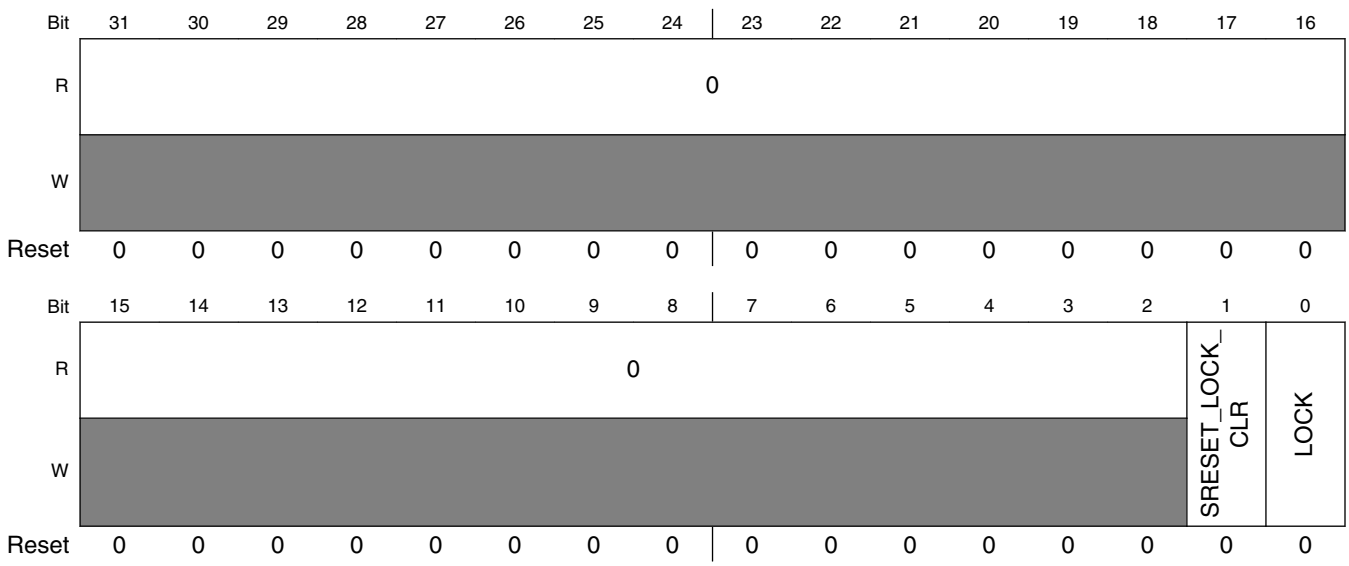
Table continues on the next page...

SDMAARMx_CONFIG field descriptions (continued)

Field	Description
	0 static 1 dynamic low power 2 dynamic with no loop 3 dynamic

7.2.7.15 SDMA LOCK (SDMAARMx_SDMA_LOCK)

Address: Base address + 3Ch offset



SDMAARMx_SDMA_LOCK field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
1 SRESET_LOCK_CLR	The SRESET_LOCK_CLR bit determine if the LOCK bit is cleared on a software reset triggered by writing to the RESET register. This bit cannot be changed if LOCK=1. SREST_LOCK_CLR is cleared by conditions that clear the LOCK bit. 0 Software Reset does not clear the LOCK bit. 1 Software Reset clears the LOCK bit.
0 LOCK	The LOCK bit is used to restrict access to update SDMA script memory through ROM channel zero scripts and through the OnCE interface under Arm platform control. The LOCK bit is set: • The SDMA_LOCK, ONCE_ENB,CH0ADDR, and ILLINSTADDR registers cannot be written. These registers can be read, but writes are ignored. • SDMA software executing out of ROM or RAM may check the LOCK bit in the LOCK register Lock Status Register (SDMACORE_SDMA_LOCK) to determine if certain operations are allowed, such as up-loading new scripts.

Table continues on the next page...

SDMAARMx_SDMA_LOCK field descriptions (continued)

Field	Description
	Once the LOCK bit is set to 1, only a reset can clear it. The LOCK bit is cleared by a hardware reset. LOCK is cleared by a software reset only if SRESET_LOCK_CLR is set. 0 LOCK disengaged. 1 LOCK enabled.

7.2.7.16 OnCE Enable (SDMAARMx_ONCE_ENB)

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															ENB
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_ONCE_ENB field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 ENB	The OnCE Enable register selects the OnCE control source: When cleared (0), the OnCE registers are accessed through the JTAG interface; when set (1), the OnCE registers may be accessed by the Arm platform through the addresses described, as follows. • After reset, the OnCE registers are accessed through the JTAG interface. • Writing a 1 to ENB enables the Arm platform to access the ONCE_* as any other SDMA control register. • When cleared (0), all the ONCE_xxx registers cannot be written. The value of ENB cannot be changed if the LOCK bit in the SDMA_LOCK register is set.

7.2.7.17 OnCE Data Register (SDMAARMx_ONCE_DATA)

Address: Base address + 44h offset

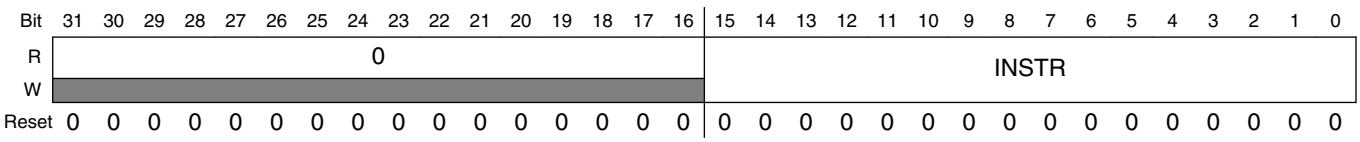
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_ONCE_DATA field descriptions

Field	Description
DATA	Data register of the OnCE JTAG controller. Refer to OnCE and Real-Time Debug for information on this register.

7.2.7.18 OnCE Instruction Register (SDMAARMx_ONCE_INSTR)

Address: Base address + 48h offset

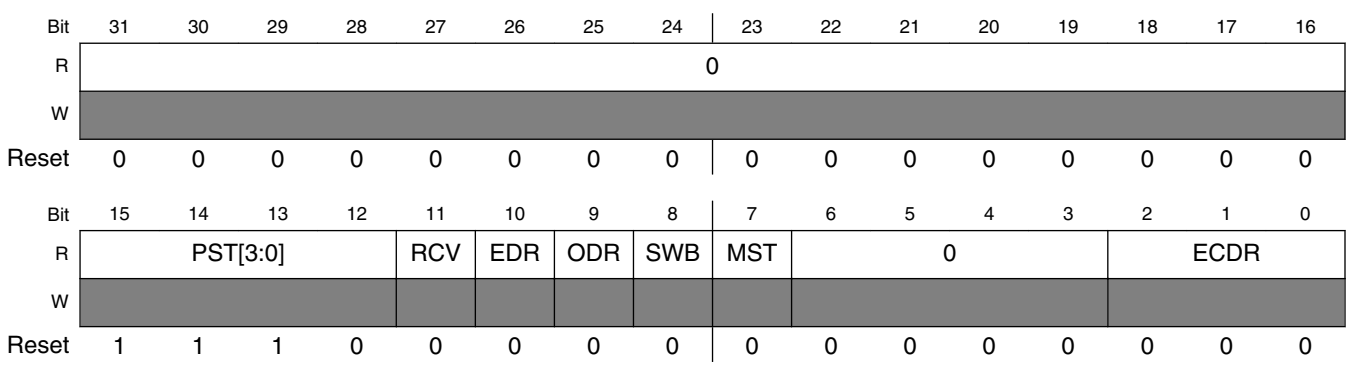


SDMAARMx_ONCE_INSTR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
INSTR	Instruction register of the OnCE JTAG controller. Refer to OnCE and Real-Time Debug for information on this register.

7.2.7.19 OnCE Status Register (SDMAARMx_ONCE_STAT)

Address: Base address + 4Ch offset



SDMAARMx_ONCE_STAT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–12 PST[3:0]	The Processor Status bits reflect the state of the SDMA RISC engine. Its states are as follows: • The "Program" state is the usual instruction execution cycle. • The "Data" state is inserted when there are wait-states during a load or a store on the data bus (ld or st).

Table continues on the next page...

SDMAARMx_ONCE_STAT field descriptions (continued)

Field	Description
	• The "Change of Flow" state is the second cycle of any instruction that breaks the sequence of instructions (jumps and channel switching instructions). • The "Change of Flow in Loop" state is used when an error causes a hardware loop exit. • The "Debug" state means the SDMA is in debug mode. • The "Functional Unit" state is inserted when there are wait-states during a load or a store on the functional units bus (ldf or stf). • In "Sleep" modes, no script is running (this is the RISC engine idle state). The "after Reset" is slightly different because no context restoring phase will happen when a channel is triggered: The script located at address 0 will be executed (boot operation). • The "in Sleep" states are the same as above except they do not have any corresponding channel: They are used when entering debug mode after reset. The reason is that it is necessary to return to the "Sleep after Reset" state when leaving debug mode. 0 Program 1 Data 2 Change of Flow 3 Change of Flow in Loop 4 Debug 5 Functional Unit 6 Sleep 7 Save 8 Program in Sleep 9 Data in Sleep 10 Change of Flow in Sleep 11 Change Flow in Loop in Sleep 12 Debug in Sleep 13 Functional Unit in Sleep 14 Sleep after Reset 15 Restore
11 RCV	After each write access to the real time buffer (RTB), the RCV bit is set. This bit is cleared after execution of an `rbuffer` command and on a JTAG reset.
10 EDR	This flag is raised when the SDMA has entered debug mode after an external debug request.
9 ODR	This flag is raised when the SDMA has entered debug mode after a OnCE debug request.
8 SWB	This flag is raised when the SDMA has entered debug mode after a software breakpoint.
7 MST	This flag is raised when the OnCE is controlled from the Arm platform peripheral interface. 0 The JTAG interface controls the OnCE. 1 The Arm platform peripheral interface controls the OnCE.
6–3 Reserved	This read-only field is reserved and always has the value 0.
ECDR	Event Cell Debug Request. If the debug request comes from the event cell, the reason for entering debug mode is given by the EDR bits. If all three bits of the EDR are reset, then it did not generate any debug request. If the cell did generate a debug request, then at least one of the EDR bits is set (the meaning of the encoding is given below). The encoding of the EDR bits is useful to find out more precisely why the debug request was generated. A debug request from an event cell is generated for a specific combination of the addra_cond, addrb_cond, and data_cond conditions. The value of those fields is given by the EDR bits.

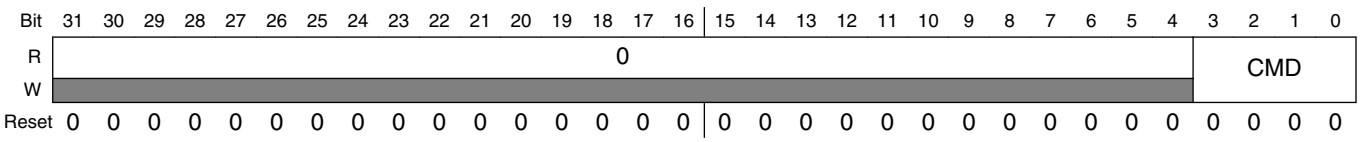
Table continues on the next page...

SDMAARMx_ONCE_STAT field descriptions (continued)

Field	Description
	0 1 matched addra_cond 1 1 matched addrb_cond 2 1 matched data_cond

7.2.7.20 OnCE Command Register (SDMAARMx_ONCE_CMD)

Address: Base address + 50h offset

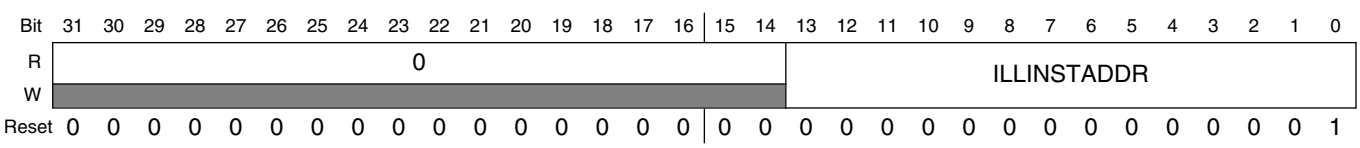


SDMAARMx_ONCE_CMD field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value 0.
CMD	Writing to this register will cause the OnCE to execute the command that is written. When needed, the ONCE_DATA and ONCE_INSTR registers should be loaded with the correct value before writing the command to that register. For a list of the OnCE commands and their usage, see OnCE and Real-Time Debug. **NOTE:** 7-15 *reserved* 0 rstatus 1 dmov 2 exec_once 3 run_core 4 exec_core 5 debug_rqst 6 rbuffer

7.2.7.21 Illegal Instruction Trap Address (SDMAARMx_ILLINSTADDR)

Address: Base address + 58h offset



SDMAARMx_ILLINSTADDR field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
ILLINSTADDR	The Illegal Instruction Trap Address is the address where the SDMA jumps when an illegal instruction is executed. It is 0x0001 after reset. The value of ILLINSTADDR cannot be changed if the LOCK bit in the SDMA_LOCK register is set.

7.2.7.22 Channel 0 Boot Address (SDMAARMx_CHN0ADDR)

Address: Base address + 5Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SMSZ	CHN0ADDR													
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0

SDMAARMx_CHN0ADDR field descriptions

Field	Description
31–15 Reserved	This read-only field is reserved and always has the value 0.
14 SMSZ	The bit 14 (Scratch Memory Size) determines if scratch memory must be available after every channel context. After reset, it is equal to 0, which defines a RAM space of 24 words for each channel. All of this area stores the channel context. By setting this bit, 32 words are reserved for every channel context, which gives eight additional words that can be used by the channel script to store any type of data. Those words are never erased by the context switching mechanism. The value of SMSZ cannot be changed if the LOCK bit in the SDMA_LOCK register is set. 0 24 words per context 1 32 words per context
CHN0ADDR	This 14-bit register is used by the boot code of the SDMA. After reset, it points to the standard boot routine in ROM (channel 0 routine). By changing this address, you can perform a boot sequence with your own routine. The very first instructions of the boot code fetch the contents of this register (it is also mapped in the SDMA memory space) and jump to the given address. The reset value is 0x0050 (decimal 80). The value of CHN0ADDR cannot be changed if the LOCK bit in the SDMA_LOCK register is set.

7.2.7.23 DMA Requests (SDMAARMx_EVT_MIRROR)

Address: Base address + 60h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EVENTS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_EVT_MIRROR field descriptions

Field	Description
EVENTS	This register reflects the DMA requests received by the SDMA for events 31-0. The Arm platform and the SDMA have a read-only access. There is one bit associated with each of 32 DMA request events. This information may be useful during debug of the blocks that generate the DMA requests. The EVT_MIRROR register is cleared following read access. 0 DMA request event not pending 1 DMA request event pending

7.2.7.24 DMA Requests 2 (SDMAARMx_EVT_MIRROR2)

Address: Base address + 64h offset

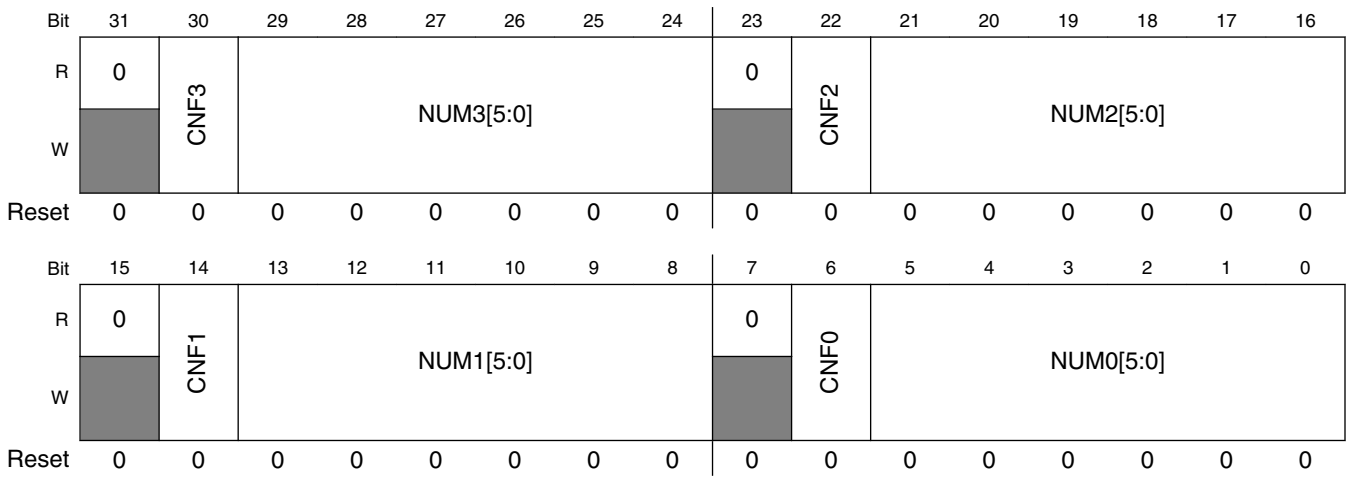
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EVENTS[47:32]															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_EVT_MIRROR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
EVENTS[47:32]	This register reflects the DMA requests received by the SDMA for events 47-32. The Arm platform and the SDMA have a read-only access. There is one bit associated with each of DMA request events. This information may be useful during debug of the blocks that generate the DMA requests. The EVT_MIRROR2 register is cleared following read access. 0 - DMA request event not pending 1- DMA request event pending

7.2.7.25 Cross-Trigger Events Configuration Register 1 (SDMAARMx_XTRIG_CONF1)

Address: Base address + 70h offset



SDMAARMx_XTRIG_CONF1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30 CNF3	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by the reception of a DMA request or by the starting of a channel script execution. 0 channel 1 DMA request
29–24 NUM3[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
23 Reserved	This read-only field is reserved and always has the value 0.
22 CNF2	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
21–16 NUM2[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
15 Reserved	This read-only field is reserved and always has the value 0.
14 CNF1	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution.

Table continues on the next page...

SDMAARMx_XTRIG_CONF1 field descriptions (continued)

Field	Description
	0 channel 1 DMA request
13–8 NUM1[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
7 Reserved	This read-only field is reserved and always has the value 0.
6 CNF0	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
NUM0[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.

7.2.7.26 Cross-Trigger Events Configuration Register 2 (SDMAARMx_XTRIG_CONF2)

Address: Base address + 74h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	CNF7	NUM7[5:0]						0	CNF6	NUM6[5:0]					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	CNF5	NUM5[5:0]						0	CNF4	NUM4[5:0]					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_XTRIG_CONF2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30 CNF7	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request

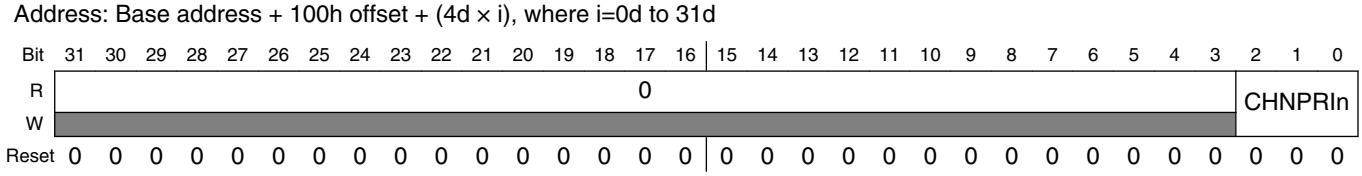
Table continues on the next page...

SDMAARMx_XTRIG_CONF2 field descriptions (continued)

Field	Description
29–24 NUM7[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
23 Reserved	This read-only field is reserved and always has the value 0.
22 CNF6	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
21–16 NUM6[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
15 Reserved	This read-only field is reserved and always has the value 0.
14 CNF5	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution 0 channel 1 DMA request
13–8 NUM5[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
7 Reserved	This read-only field is reserved and always has the value 0.
6 CNF4	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
NUM4[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.

7.2.7.27 Channel Priority Registers (SDMAARMx_SDMA_CHNPRI*n*)

Address: Base address + 100h offset + (4d × i), where i=0d to 31d



SDMAARMx_SDMA_CHNPRI*n* field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
CHNPRIn	This contains the priority of channel number *n*. Useful values are between 1 and 7; 0 is reserved by the SDMA hardware to determine when there is no pending channel. Reset value is 0, which prevents the channels from starting.

7.2.7.28 Channel Enable RAM (SDMAARMx_CHNENBL*n*)

Address: Base address + 200h offset + (4d × i), where i=0d to 47d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ENBLn																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_CHNENBL*n* field descriptions

Field	Description
ENBLn	This 32-bit value selects the channels that are triggered by the DMA request number *n*. If ENBLn[i] is set to 1, bit EP[i] will be set when the DMA request *n* is received. These 48 32-bit registers are physically located in a RAM, with no known reset value. It is thus essential for the Arm platform to program them before any DMA request is triggered to the SDMA, otherwise an unpredictable combination of channels may be started.

7.2.7.29 SDMA DONE0 Configuration (SDMAARMx_DONE0_CONFIG)

Address: Base address + 1000h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DONE_SEL3	SW_DONE_DIS3	0	CH_SEL3					DONE_SEL2	SW_DONE_DIS2	0	CH_SEL2				
W																
Reset	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DONE_SEL1	SW_DONE_DIS1	0	CH_SEL1					DONE_SEL0	SW_DONE_DIS0	0	CH_SEL0				
W																
Reset	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

SDMAARMx_DONE0_CONFIG field descriptions

Field	Description
31 DONE_SEL3	Select Done from SW or HW for channel 3 0 HW 1 SW
30 SW_DONE_DIS3	Disable SW Done for channel 3 0 Enable 1 Disable
29 Reserved	This read-only field is reserved and always has the value 0.
28–24 CH_SEL3	Select event for channel 3 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected. 00000 - Event 0 00001 - Event 1 ... 11111 - Event 31
23 DONE_SEL2	Select Done from SW or HW for channel 2 0 HW 1 SW
22 SW_DONE_DIS2	Disable SW Done for channel 2 0 Enable 1 Disable
21 Reserved	This read-only field is reserved and always has the value 0.
20–16 CH_SEL2	Select event for channel 2 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected. 00000 - Event 0 00001 - Event 1 ... 11111 - Event 31
15 DONE_SEL1	Select Done from SW or HW for channel 1 0 HW 1 SW
14 SW_DONE_DIS1	Disable SW Done for channel 1 0 Enable 1 Disable
13 Reserved	This read-only field is reserved and always has the value 0.
12–8 CH_SEL1	Select event for channel 1 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected.

Table continues on the next page...

SDMAARMx_DONE0_CONFIG field descriptions (continued)

Field	Description
	00000 - Event 0 00001 - Event 1 ... 11111 - Event 31
7 DONE_SEL0	Select Done from SW or HW for channel 0 0 HW 1 SW
6 SW_DONE_DIS0	Disable SW Done for channel 0 0 Enable 1 Disable
5 Reserved	This read-only field is reserved and always has the value 0.
CH_SEL0	Select event for channel 0 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected. 00000 - Event 0 00001 - Event 1 ... 11111 - Event 31

7.2.7.30 SDMA DONE1 Configuration (SDMAARMx_DONE1_CONFIG)

Address: Base address + 1004h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DONE_SEL7	SW_DONE_DIS7	0	CH_SEL7					DONE_SEL6	SW_DONE_DIS6	0	CH_SEL6				
W																
Reset	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DONE_SEL5	SW_DONE_DIS5	0	CH_SEL5					DONE_SEL4	SW_DONE_DIS4	0	CH_SEL4				
W																
Reset	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

SDMAARMx_DONE1_CONFIG field descriptions

Field	Description
31 DONE_SEL7	Select Done from SW or HW for channel 7 0 HW 1 SW
30 SW_DONE_DIS7	Disable SW Done for channel 7 0 Enable 1 Disable
29 Reserved	This read-only field is reserved and always has the value 0.
28–24 CH_SEL7	Select event for channel 7 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected. 00000 - Event 0 00001 - Event 1 ... 11111 - Event 31
23 DONE_SEL6	Select Done from SW or HW for channel 6 0 HW 1 SW
22 SW_DONE_DIS6	Disable SW Done for channel 6 0 Enable 1 Disable
21 Reserved	This read-only field is reserved and always has the value 0.
20–16 CH_SEL6	Select event for channel 6 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected. 00000 - Event 0 00001 - Event 1 ... 11111 - Event 31
15 DONE_SEL5	Select Done from SW or HW for channel 5 0 HW 1 SW
14 SW_DONE_DIS5	Disable SW Done for channel 5 0 Enable 1 Disable
13 Reserved	This read-only field is reserved and always has the value 0.
12–8 CH_SEL5	Select event for channel 5 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected.

Table continues on the next page...

SDMAARMx_DONE1_CONFIG field descriptions (continued)

Field	Description
	00000 - Event 0 00001 - Event 1 ... 11111 - Event 31
7 DONE_SEL4	Select Done from SW or HW for channel 4 0 HW 1 SW
6 SW_DONE_DIS4	Disable SW Done for channel 4 0 Enable 1 Disable
5 Reserved	This read-only field is reserved and always has the value 0.
CH_SEL4	Select event for channel 4 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected. 00000 - Event 0 00001 - Event 1 ... 11111 - Event 31

7.2.8 BP Memory Map and Control Register Definitions

The following section describes SDMA control registers available to the BP.

NOTE

These registers are physically implemented in all platforms, but are not accessible when the SDMA BP control port is not connected. Reset values are calculated to allow the system to work when those registers cannot be accessed.

All registers are clocked with the SDMA clock (which means the SDMA clock must be running when the BP wants to access any register).

SDMABP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_0000	Channel 0 Pointer (SDMABP1_DC0PTR)	32	R/W	0000_0000h	7.2.8.1/1243

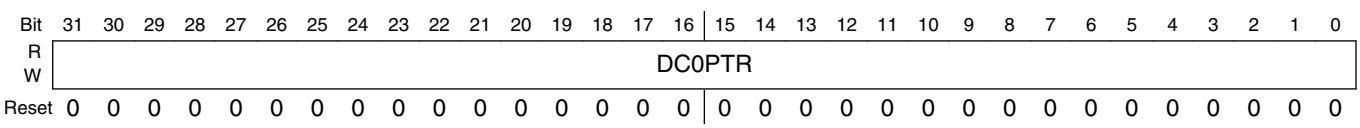
Table continues on the next page...

SDMABP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_0004	Channel Interrupts (SDMABP1_INTR)	32	w1c	0000_0000h	7.2.8.2/1243
30BD_0008	Channel Stop/Channel Status (SDMABP1_STOP_STAT)	32	R/W	0000_0000h	7.2.8.3/1243
30BD_000C	Channel Start (SDMABP1_DSTART)	32	R	0000_0000h	7.2.8.4/1244
30BD_0028	DMA Request Error Register (SDMABP1_EVTERR)	32	R	0000_0000h	7.2.8.5/1244
30BD_002C	Channel DSP Interrupt Mask (SDMABP1_INTRMASK)	32	R/W	0000_0000h	7.2.8.6/1245
30BD_0034	DMA Request Error Register (SDMABP1_EVTERRDBG)	32	R	0000_0000h	7.2.8.7/1245
30E0_0000	Channel 0 Pointer (SDMABP3_DC0PTR)	32	R/W	0000_0000h	7.2.8.1/1243
30E0_0004	Channel Interrupts (SDMABP3_INTR)	32	w1c	0000_0000h	7.2.8.2/1243
30E0_0008	Channel Stop/Channel Status (SDMABP3_STOP_STAT)	32	R/W	0000_0000h	7.2.8.3/1243
30E0_000C	Channel Start (SDMABP3_DSTART)	32	R	0000_0000h	7.2.8.4/1244
30E0_0028	DMA Request Error Register (SDMABP3_EVTERR)	32	R	0000_0000h	7.2.8.5/1244
30E0_002C	Channel DSP Interrupt Mask (SDMABP3_INTRMASK)	32	R/W	0000_0000h	7.2.8.6/1245
30E0_0034	DMA Request Error Register (SDMABP3_EVTERRDBG)	32	R	0000_0000h	7.2.8.7/1245
30E1_0000	Channel 0 Pointer (SDMABP2_DC0PTR)	32	R/W	0000_0000h	7.2.8.1/1243
30E1_0004	Channel Interrupts (SDMABP2_INTR)	32	w1c	0000_0000h	7.2.8.2/1243
30E1_0008	Channel Stop/Channel Status (SDMABP2_STOP_STAT)	32	R/W	0000_0000h	7.2.8.3/1243
30E1_000C	Channel Start (SDMABP2_DSTART)	32	R	0000_0000h	7.2.8.4/1244
30E1_0028	DMA Request Error Register (SDMABP2_EVTERR)	32	R	0000_0000h	7.2.8.5/1244
30E1_002C	Channel DSP Interrupt Mask (SDMABP2_INTRMASK)	32	R/W	0000_0000h	7.2.8.6/1245
30E1_0034	DMA Request Error Register (SDMABP2_EVTERRDBG)	32	R	0000_0000h	7.2.8.7/1245

7.2.8.1 Channel 0 Pointer (SDMABPx_DC0PTR)

Address: Base address + 0h offset



SDMABPx_DC0PTR field descriptions

Field	Description
DC0PTR	**Channel 0 Pointer** contains the 32-bit address, in BP memory, of the array of channel control blocks starting with the one for channel 0 (the control channel). This register should be initialized by the BP before it enables a channel (for example, channel 0). See the API document SDMA Scripts User Manual for the use of this register. The BP has a read/write access and the SDMA has a read-only access.

7.2.8.2 Channel Interrupts (SDMABPx_INTR)

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DI																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABPx_INTR field descriptions

Field	Description
DI	The BP Interrupts register contains the 32 DI[i] bits. If any bit is set, it will cause an interrupt to the BP. • This register is a "write-ones" register to the BP. When the BP sets a bit in this register, the corresponding DI[i] bit is cleared. • The interrupt service routine should clear individual channel bits when their interrupts are serviced; failure to do so will cause continuous interrupts. • The SDMA is responsible for setting the DI[i] bit corresponding to the current channel when the corresponding `done` instruction is executed.

7.2.8.3 Channel Stop/Channel Status (SDMABPx_STOP_STAT)

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DE																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABPx_STOP_STAT field descriptions

Field	Description
DE	This 32-bit register gives access to the BP (DSP) Enable bits, DE. There is one bit for every channel. • This register is a "write-ones" register to the BP. • When the BP writes 1 in bit *i* of this register, it clears the DE[i] and DSTART[i] bits. • Reading this register yields the current state of the DE[i] bits.

7.2.8.4 Channel Start (SDMABPx_DSTART)

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DSTART_DE																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABPx_DSTART field descriptions

Field	Description
DSTART_DE	The DSTART_DE registers are 32 bits wide with one bit for every channel. • When a bit is written to 1, it enables the corresponding channel. • Two physical registers are accessed with that address (DSTART and DE), which enables the BP to trigger a channel a second time before the first trigger was processed. • This register is a "write-ones" register to the BP. Neither DSTART[i] bit can be set while the corresponding DE[i] bit is cleared. • When the BP tries to set the DSTART[i] bit by writing a one (if the corresponding DE[i] bit is clear), the bit in the DSTART[i] register will remain cleared and the DE[i] bit will be set. If the corresponding DE[i] bit was already set, the DSTART[i] bit will be set. • The next time the SDMA channel *i* attempts to clear the DE[i] bit by means of a `done` instruction, the bit in the DSTART[i] register will be cleared and the DE[i] bit will take the old value of the DSTART[i] bit. • Reading this register yields the current state of the DSTART[i] bits. This mechanism enables the BP to pipeline two DSTART commands per channel.

7.2.8.5 DMA Request Error Register (SDMABPx_EVTERR)

Address: Base address + 28h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHNERR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABPx_EVTERR field descriptions

Field	Description
CHNERR	This register is used by the SDMA to warn the BP when an incoming DMA request was detected; it then triggers a channel that is already pending or being serviced, which may mean there is an overflow of data for that channel. An interrupt is sent to the BP if the corresponding channel bit is set in the INTRMASK register. • This is a "write-ones" register for the scheduler. It is only able to set the flags. The flags are cleared when the register is read by the BP or during an SDMA reset. • The CHNERR[i] bit is set when a DMA request that triggers channel *i* is received through the corresponding input pins and the EP[i] bit is already set. The EVTERR[i] bit is unaffected if the BP tries to set the EP[i] bit when that EP[i] bit is already set.

7.2.8.6 Channel DSP Interrupt Mask (SDMABPx_INTRMASK)

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DIMASK																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABPx_INTRMASK field descriptions

Field	Description
DIMASK	The Interrupt Mask Register contains 32 interrupt generation mask bits. If bit DIMASK[i] is set, the DI[i] bit is set and an interrupt is sent to the BP when a DMA request error is detected on channel *i* (for example, EVTERR[i] is set).

7.2.8.7 DMA Request Error Register (SDMABPx_EVTERRDBG)

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHNERR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABPx_EVTERRDBG field descriptions

Field	Description
CHNERR	This register is the same as EVTERR except reading it does not clear its contents. This address is meant to be used in debug mode. The BP OnCE may check this register value without modifying it.

7.2.9 SDMA Internal (Core) Memory Map and Internal Register Definitions

The actual SDMA memory mapped registers are summarized in the following sections; for peripherals' memory maps, refer to the respective chapters.

The following definitions serve as a key for the SDMA internal register summary.

SDMACORE memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_0000	Arm platform Channel 0 Pointer (SDMACORE1_MC0PTR)	32	R	0000_0000h	7.2.9.1/1249
30BD_0008	Current Channel Pointer (SDMACORE1_CCPTR)	32	R	0000_0000h	7.2.9.2/1249
30BD_000C	Current Channel Register (SDMACORE1_CCR)	32	R	0000_0000h	7.2.9.3/1250
30BD_0010	Highest Pending Channel Register (SDMACORE1_NCR)	32	R	0000_0000h	7.2.9.4/1250
30BD_0014	External DMA Requests Mirror (SDMACORE1_EVENTS)	32	R	0000_0000h	7.2.9.5/1251
30BD_0018	Current Channel Priority (SDMACORE1_CCPRI)	32	R	0000_0000h	7.2.9.6/1252
30BD_001C	Next Channel Priority (SDMACORE1_NCPRI)	32	R	0000_0000h	7.2.9.7/1252

Table continues on the next page...

SDMACORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_0020	OnCE Event Cell Counter (SDMACORE1_ECOUNT)	32	R/W	0000_0000h	7.2.9.8/1253
30BD_0024	OnCE Event Cell Control Register (SDMACORE1_ECTL)	32	R/W	0000_0000h	7.2.9.9/1253
30BD_0028	OnCE Event Address Register A (SDMACORE1_EAA)	32	R/W	0000_0000h	7.2.9.10/ 1255
30BD_002C	OnCE Event Cell Address Register B (SDMACORE1_EAB)	32	R/W	0000_0000h	7.2.9.11/ 1255
30BD_0030	OnCE Event Cell Address Mask (SDMACORE1_EAM)	32	R/W	0000_0000h	7.2.9.12/ 1255
30BD_0034	OnCE Event Cell Data Register (SDMACORE1_ED)	32	R/W	0000_0000h	7.2.9.13/ 1256
30BD_0038	OnCE Event Cell Data Mask (SDMACORE1_EDM)	32	R/W	0000_0000h	7.2.9.14/ 1256
30BD_003C	OnCE Real-Time Buffer (SDMACORE1_RTB)	32	R/W	0000_0000h	7.2.9.15/ 1257
30BD_0040	OnCE Trace Buffer (SDMACORE1_TB)	32	R	0000_0000h	7.2.9.16/ 1257
30BD_0044	OnCE Status (SDMACORE1_OSTAT)	32	R	0000_0000h	7.2.9.17/ 1258
30BD_0048	Channel 0 Boot Address (SDMACORE1_MCHN0ADDR)	32	R	0000_0000h	7.2.9.18/ 1260
30BD_004C	ENDIAN Status Register (SDMACORE1_ENDIANNESS)	32	R	0000_0001h	7.2.9.19/ 1261
30BD_0054	Lock Status Register (SDMACORE1_SDMA_LOCK)	32	R	0000_0000h	7.2.9.20/ 1262
30BD_0058	External DMA Requests Mirror #2 (SDMACORE1_EVENTS2)	32	R	0000_0000h	7.2.9.21/ 1262
30E0_0000	Arm platform Channel 0 Pointer (SDMACORE3_MC0PTR)	32	R	0000_0000h	7.2.9.1/1249
30E0_0008	Current Channel Pointer (SDMACORE3_CCPTR)	32	R	0000_0000h	7.2.9.2/1249
30E0_000C	Current Channel Register (SDMACORE3_CCR)	32	R	0000_0000h	7.2.9.3/1250
30E0_0010	Highest Pending Channel Register (SDMACORE3_NCR)	32	R	0000_0000h	7.2.9.4/1250
30E0_0014	External DMA Requests Mirror (SDMACORE3_EVENTS)	32	R	0000_0000h	7.2.9.5/1251
30E0_0018	Current Channel Priority (SDMACORE3_CCPRI)	32	R	0000_0000h	7.2.9.6/1252
30E0_001C	Next Channel Priority (SDMACORE3_NCPRI)	32	R	0000_0000h	7.2.9.7/1252
30E0_0020	OnCE Event Cell Counter (SDMACORE3_ECOUNT)	32	R/W	0000_0000h	7.2.9.8/1253
30E0_0024	OnCE Event Cell Control Register (SDMACORE3_ECTL)	32	R/W	0000_0000h	7.2.9.9/1253
30E0_0028	OnCE Event Address Register A (SDMACORE3_EAA)	32	R/W	0000_0000h	7.2.9.10/ 1255
30E0_002C	OnCE Event Cell Address Register B (SDMACORE3_EAB)	32	R/W	0000_0000h	7.2.9.11/ 1255
30E0_0030	OnCE Event Cell Address Mask (SDMACORE3_EAM)	32	R/W	0000_0000h	7.2.9.12/ 1255
30E0_0034	OnCE Event Cell Data Register (SDMACORE3_ED)	32	R/W	0000_0000h	7.2.9.13/ 1256

Table continues on the next page...

SDMACORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30E0_0038	OnCE Event Cell Data Mask (SDMACORE3_EDM)	32	R/W	0000_0000h	7.2.9.14/ 1256
30E0_003C	OnCE Real-Time Buffer (SDMACORE3_RTB)	32	R/W	0000_0000h	7.2.9.15/ 1257
30E0_0040	OnCE Trace Buffer (SDMACORE3_TB)	32	R	0000_0000h	7.2.9.16/ 1257
30E0_0044	OnCE Status (SDMACORE3_OSTAT)	32	R	0000_0000h	7.2.9.17/ 1258
30E0_0048	Channel 0 Boot Address (SDMACORE3_MCHN0ADDR)	32	R	0000_0000h	7.2.9.18/ 1260
30E0_004C	ENDIAN Status Register (SDMACORE3_ENDIANNESS)	32	R	0000_0001h	7.2.9.19/ 1261
30E0_0054	Lock Status Register (SDMACORE3_SDMA_LOCK)	32	R	0000_0000h	7.2.9.20/ 1262
30E0_0058	External DMA Requests Mirror #2 (SDMACORE3_EVENTS2)	32	R	0000_0000h	7.2.9.21/ 1262
30E1_0000	Arm platform Channel 0 Pointer (SDMACORE2_MC0PTR)	32	R	0000_0000h	7.2.9.1/1249
30E1_0008	Current Channel Pointer (SDMACORE2_CCPTR)	32	R	0000_0000h	7.2.9.2/1249
30E1_000C	Current Channel Register (SDMACORE2_CCR)	32	R	0000_0000h	7.2.9.3/1250
30E1_0010	Highest Pending Channel Register (SDMACORE2_NCR)	32	R	0000_0000h	7.2.9.4/1250
30E1_0014	External DMA Requests Mirror (SDMACORE2_EVENTS)	32	R	0000_0000h	7.2.9.5/1251
30E1_0018	Current Channel Priority (SDMACORE2_CCPRI)	32	R	0000_0000h	7.2.9.6/1252
30E1_001C	Next Channel Priority (SDMACORE2_NCPRI)	32	R	0000_0000h	7.2.9.7/1252
30E1_0020	OnCE Event Cell Counter (SDMACORE2_ECOUNT)	32	R/W	0000_0000h	7.2.9.8/1253
30E1_0024	OnCE Event Cell Control Register (SDMACORE2_ECTL)	32	R/W	0000_0000h	7.2.9.9/1253
30E1_0028	OnCE Event Address Register A (SDMACORE2_EAA)	32	R/W	0000_0000h	7.2.9.10/ 1255
30E1_002C	OnCE Event Cell Address Register B (SDMACORE2_EAB)	32	R/W	0000_0000h	7.2.9.11/ 1255
30E1_0030	OnCE Event Cell Address Mask (SDMACORE2_EAM)	32	R/W	0000_0000h	7.2.9.12/ 1255
30E1_0034	OnCE Event Cell Data Register (SDMACORE2_ED)	32	R/W	0000_0000h	7.2.9.13/ 1256
30E1_0038	OnCE Event Cell Data Mask (SDMACORE2_EDM)	32	R/W	0000_0000h	7.2.9.14/ 1256
30E1_003C	OnCE Real-Time Buffer (SDMACORE2_RTB)	32	R/W	0000_0000h	7.2.9.15/ 1257
30E1_0040	OnCE Trace Buffer (SDMACORE2_TB)	32	R	0000_0000h	7.2.9.16/ 1257
30E1_0044	OnCE Status (SDMACORE2_OSTAT)	32	R	0000_0000h	7.2.9.17/ 1258
30E1_0048	Channel 0 Boot Address (SDMACORE2_MCHN0ADDR)	32	R	0000_0000h	7.2.9.18/ 1260

Table continues on the next page...

SDMACORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30E1_004C	ENDIAN Status Register (SDMACORE2_ENDIANNESS)	32	R	0000_0001h	7.2.9.19/ 1261
30E1_0054	Lock Status Register (SDMACORE2_SDMA_LOCK)	32	R	0000_0000h	7.2.9.20/ 1262
30E1_0058	External DMA Requests Mirror #2 (SDMACORE2_EVENTS2)	32	R	0000_0000h	7.2.9.21/ 1262

7.2.9.1 Arm platform Channel 0 Pointer (SDMACOREx_MC0PTR)

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MC0PTR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_MC0PTR field descriptions

Field	Description
MC0PTR	Contains the address-in the Arm platform memory space-of the initial SDMA context and scripts that are loaded by the SDMA boot script running on channel 0.

7.2.9.2 Current Channel Pointer (SDMACOREx_CCPTR)

Address: Base address + 8h offset

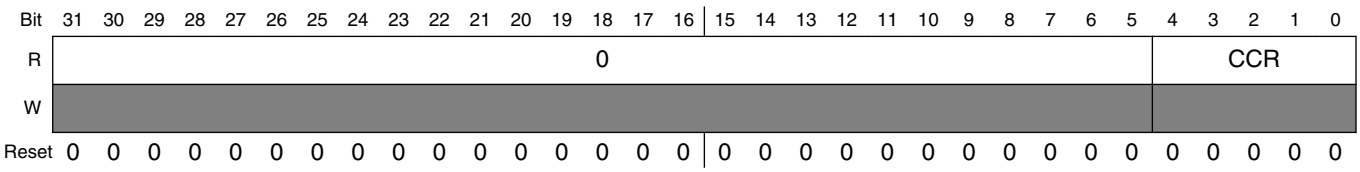
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CCPTR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_CCPTR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
CCPTR	Contains the start address of the context data for the current channel: Its value is *CONTEXT_BASE* + 24* *CCR* or *CONTEXT_BASE* + 32* *CCR* where CONTEXT_BASE = 0x0800. The value 24 or 32 is selected according to the programmed channel scratch RAM size in the register shown in Channel 0 Boot Address (SDMAARM_CHN0ADDR) .

7.2.9.3 Current Channel Register (SDMACOREx_CCR)

Address: Base address + Ch offset

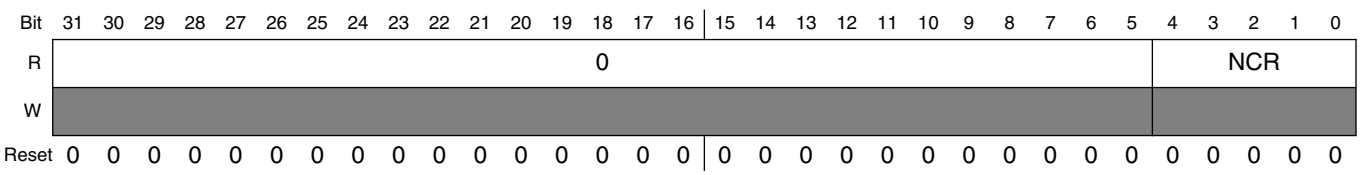


SDMACOREx_CCR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
CCR	Contains the number of the current running channel whose context is installed. In the case that the SDMA has finished running the channel and has entered sleep state, CCR will indicate the previous running channel. The PST bits in the OSTAT register indicate when the SDMA is in sleep state.

7.2.9.4 Highest Pending Channel Register (SDMACOREx_NCR)

Address: Base address + 10h offset



SDMACOREx_NCR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
NCR	Contains the number of the pending channel that the scheduler has selected to run next.

7.2.9.5 External DMA Requests Mirror (SDMACOREx_EVENTS)

NOTE

This register is very useful in the case of DMA requests that are active when a peripheral FIFO level is above the programmed watermark. The activation of the DMA request (rising edge) is detected by the SDMA logic and it can enable one or several channels. One of the channels accesses the peripheral and reads or writes a number of data that matches the watermark level (for example, if the watermark is four words, the channel reads or writes four words).

If the channel is effectively executed long after the DMA request was received, reading or writing the watermark number of data may not be sufficient to reset the DMA request (for example, if the FIFO watermark is four and at the channel execution it already contains nine pieces of data). This means no new rising edge may be detected by the SDMA, although there still remains transfers to perform. Therefore, if the channel were terminated at that time, it would not be restarted, causing potential overrun or underrun of the peripheral.

The proposed mechanism is for the channel to check this register after it has performed the "watermark" number of accesses to the peripheral. If the bit for the DMA request that triggers this channel is set, it means there is still another watermark number of data to transfer. This goes on until the bit is cleared. The same script can be used for multiple channels that require this behavior. The script can determine its channel number from the CCR register and infer the corresponding DMA request bit to check. It needs a reference table that is coherent with the request-channel matrix that the Arm platform programmed.

Address: Base address + 14h offset

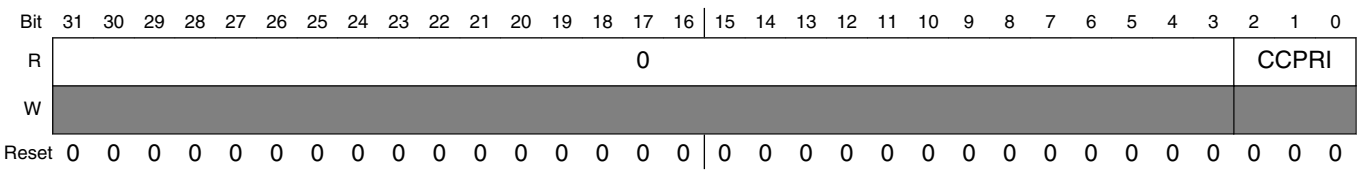
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EVENTS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_EVENTS field descriptions

Field	Description
EVENTS	Reflects the status of the SDMA's external DMA requests. It is meant to allow any channel to monitor the states of these SDMA inputs. This register displays EVENTS 0-31. The EVENTS2 register displays events 32-47.

7.2.9.6 Current Channel Priority (SDMACOREx_CCPRI)

Address: Base address + 18h offset

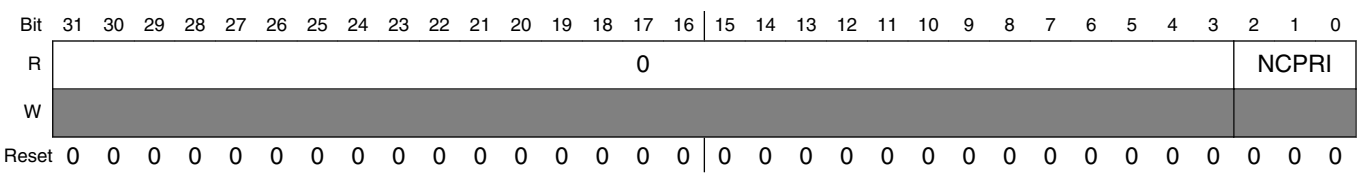


SDMACOREx_CCPRI field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
CCPRI	Contains the 3-bit priority of the channel whose context is installed. It is 0 when no channel is running. **NOTE:** 1-7 current channel priority 0 no running channel

7.2.9.7 Next Channel Priority (SDMACOREx_NCPRI)

Address: Base address + 1Ch offset

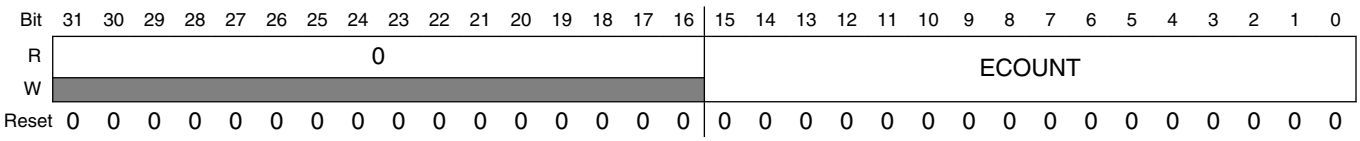


SDMACOREx_NCPRI field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
NCPRI	Contains the 3-bit priority of the channel the scheduler has selected to run next. It is 0 when no other channel is pending.

7.2.9.8 OnCE Event Cell Counter (SDMACOREx_ECOUNT)

Address: Base address + 20h offset

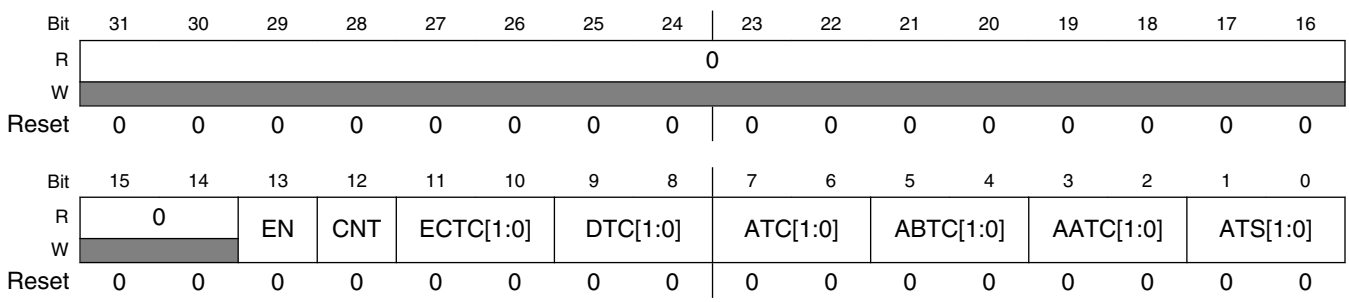


SDMACOREx_ECOUNT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
ECOUNT	The event cell counter contains the number of times minus one that an event detection must occur before generating a debug request. • This register should be written before any attempt to use the event detection counter during an event detection process. • The counter is cleared on a JTAG reset.

7.2.9.9 OnCE Event Cell Control Register (SDMACOREx_ECTL)

Address: Base address + 24h offset



SDMACOREx_ECTL field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13 EN	Event Cell Enable. If the EN bit is set, the event cell is allowed to generate debug requests (the cell is awakened). If it is cleared, the event detection unit is disabled and no hardware breakpoint is generated, but matching conditions are still reflected on the emulation pin. 0 Cell is disabled. 1 Cell is enabled.
12 CNT	Event Counter Enable. The event counter enable bit determines if the cell counter is used during the event detection. In order to use the event counter during an event detection process, the event cell counter register should be loaded with a value equal to the number of times minus one that an event occurs before a debug request is sent. After every event detection, the counter is decreased. When the counter reaches

Table continues on the next page...

SDMACOREx_ECTL field descriptions (continued)

Field	Description
	the value 0, the event detection cell sends a debug request to the core. The event counter register should be written and the EN bit should be set before each new event detection process uses the event counter. 0 Counter is disabled. 1 Counter is enabled.
11–10 ECTC[1:0]	The event cell trigger condition bits select the combination of address and data matching conditions that generate the final address/data condition. During program execution, if this event cell trigger condition goes to 1, a debug request is sent to the SDMA. The EN bit must be set to enable the debug request generation. 00 address ONLY 01 data ONLY 10 address AND data 11 address OR data
9–8 DTC[1:0]	The data trigger condition bits define when data is considered matching after comparison with the data register of the event detection unit. The operations are performed on unsigned values. 00 equal 01 not equal 10 greater than 11 less than
7–6 ATC[1:0]	The address trigger condition bits select how the two address conditions (addressA and addressB) are combined to define the global address matching condition. The supported combinations are described, as follows. 00 addressA ONLY 01 addrA AND addrB 10 addrA OR addrB 11 reserved
5–4 ABTC[1:0]	The Address B Trigger Condition (ABTC) controls the operations performed by address comparator B. All operations are performed on unsigned values. This comparator B outputs the addressB condition. 00 equal 01 not equal 10 greater than 11 less than
3–2 AATC[1:0]	The Address A Trigger Condition (AATC) controls the operations performed by address comparator A. All operations are performed on unsigned values. This comparator A outputs the addressA condition. 00 equal 01 not equal 10 greater than 11 less than
ATS[1:0]	The access type select bits define the memory access type required on the SDMA memory bus. 00 read ONLY 01 write ONLY 10 read or write 11 -

7.2.9.10 OnCE Event Address Register A (SDMACORE*x*_EAA)

Address: Base address + 28h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EAA															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE*x*_EAA field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
EAA	Event Cell Address Register A computes an address A condition. It is cleared on a JTAG reset.

7.2.9.11 OnCE Event Cell Address Register B (SDMACORE*x*_EAB)

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EAB															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE*x*_EAB field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
EAB	Event Cell Address Register B computes an address B condition. It is cleared on a JTAG reset.

7.2.9.12 OnCE Event Cell Address Mask (SDMACORE*x*_EAM)

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EAM															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE*x*_EAM field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

SDMACORE*x*_EAM field descriptions (continued)

Field	Description
EAM	The Event Cell Address Mask contains a user-defined address mask value. This mask is applied to the address value latched from the memory address bus before performing the address comparison. **NOTE:** There is a common address mask value for both address comparators. If bit *i* of this register is set, then bit *i* of the address value latched from the memory bus does not influence the result of the address comparison. The register is cleared on a JTAG reset.

7.2.9.13 OnCE Event Cell Data Register (SDMACORE*x*_ED)

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ED																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE*x*_ED field descriptions

Field	Description
ED	The event cell data register contains a user defined data value. This data value is an input for the data comparator which generates the data condition. It is cleared on a JTAG reset.

7.2.9.14 OnCE Event Cell Data Mask (SDMACORE*x*_EDM)

Address: Base address + 38h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EDM																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE*x*_EDM field descriptions

Field	Description
EDM	The event cell data mask register contains the user-defined data mask value. • This mask is applied to the data value latched from the memory bus before performing the data comparison. • Setting bit *i* of the event cell data mask register means that bit *i* of the data value latched from the address bus does not influence the result of the data comparison. • The data mask is cleared on a JTAG reset.

7.2.9.15 OnCE Real-Time Buffer (SDMACOREx_RTB)

Address: Base address + 3Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RTB																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_RTB field descriptions

Field	Description
RTB	The Real Time Buffer register stores and retrieves run time information without putting the SDMA in debug mode. Writing to that register triggers a pulse on a specific real-time debug pin whose connection depends on the chip implementation. The RTB value can be accessed by the OnCE under Arm platform or JTAG control using the rbuffer command.

7.2.9.16 OnCE Trace Buffer (SDMACOREx_TB)

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			TBF	TADDR											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

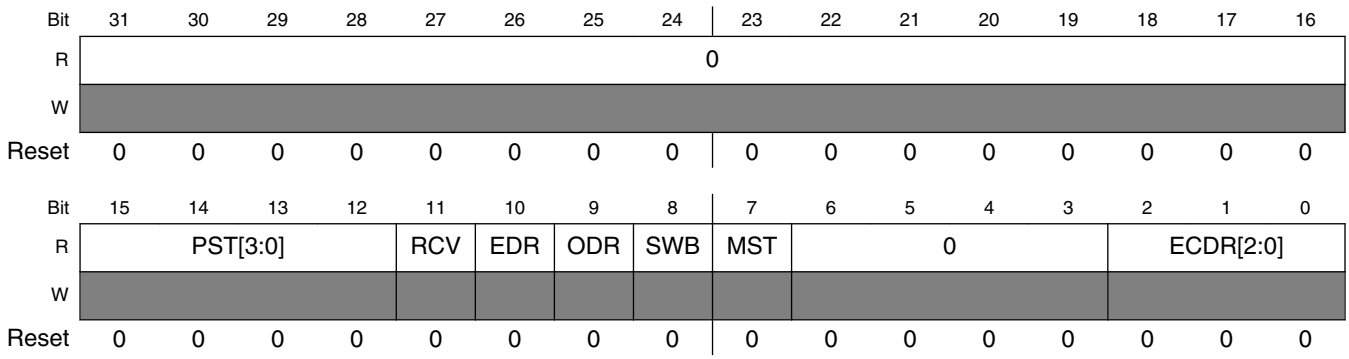
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TADDR			CHFADDR												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_TB field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28 TBF	The Trace Buffer Flag is set when the buffer contains the addresses of a valid change of flow. The contents of the buffer should be ignored otherwise. 0 Invalid information 1 Valid information
27–14 TADDR	The target address is the address taken after the execution of the change of flow instruction.
CHFADDR	The change of flow address is the address where the change of flow is taken when executing a change of flow instruction.

7.2.9.17 OnCE Status (SDMACOREx_OSTAT)

Address: Base address + 44h offset



SDMACOREx_OSTAT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–12 PST[3:0]	The Processor Status bits reflect the state of the SDMA RISC engine. • The "Program" state is the usual instruction execution cycle. • The "Data" state is inserted when there are wait-states during a load or a store on the data bus (ld or st). • The "Change of Flow" state is the second cycle of any instruction that breaks the sequence of instructions (jumps and channel-switching instructions). • The "Change of Flow in Loop" state is used when an error causes a hardware loop exit. • The "Debug" state means the SDMA is in debug mode. • The "Functional Unit" state is inserted when there are wait-states during a load or a store on the functional units bus (ldf or stf). • In "Sleep" modes, no script is running (this is the RISC engine idle state). The "after Reset" is slightly different because no context restoring phase will happen when a channel is triggered: The script located at address 0 will be executed (boot operation). • The "in Sleep" states are the same as above except they do not have any corresponding channel. They are used when entering debug mode after reset; the reason is that it is necessary to return to the "Sleep after Reset" state when leaving debug mode. 0 Program 1 Data 2 Change of Flow 3 Change of Flow in Loop 4 Debug 5 Functional Unit 6 Sleep 7 Save 8 Program in Sleep 9 Data in Sleep 10 Change of Flow in Sleep 11 Change Flow Loop Sleep 12 Debug in Sleep 13 Functional Unit in Sleep

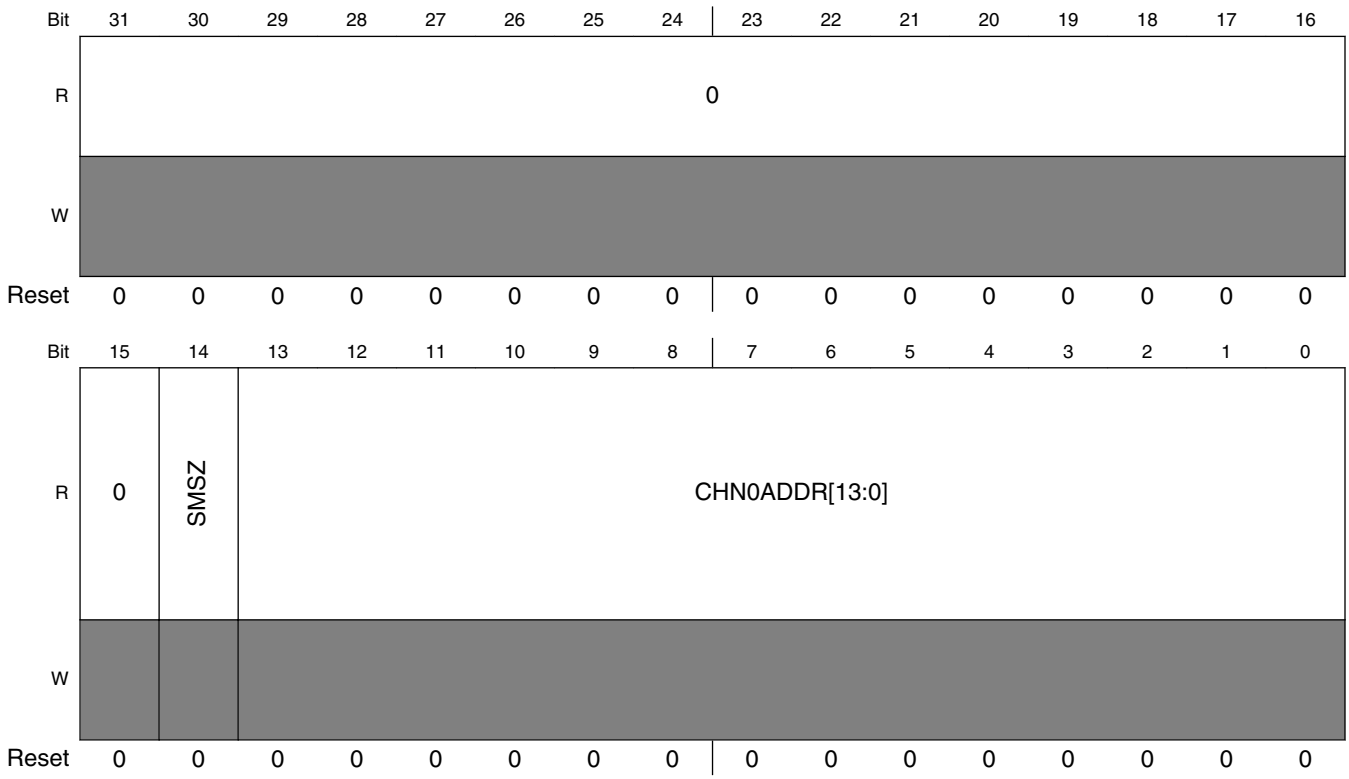
Table continues on the next page...

SDMACOREx_OSTAT field descriptions (continued)

Field	Description
	14 Sleep after Reset 15 Restore
11 RCV	After each write access to the real time buffer (RTB), the RCV bit is set. This bit is cleared after execution of an `rbuffer` command and on a JTAG reset.
10 EDR	This flag is raised when the SDMA has entered debug mode after an external debug request.
9 ODR	This flag is raised when the SDMA has entered debug mode after a OnCE debug request.
8 SWB	This flag is raised when the SDMA has entered debug mode after a software breakpoint.
7 MST	This flag is raised when the OnCE is controlled from the Arm platform peripheral interface. 0 JTAG interface controls the OnCE. 1 Arm platform peripheral interface controls the OnCE.
6–3 Reserved	This read-only field is reserved and always has the value 0.
ECDR[2:0]	Event Cell Debug Request. If the debug request comes from the event cell, the reason for entering debug mode is given by the EDR bits. The encoding of the EDR bits is useful to find out more precisely why the debug request was generated. A debug request from an event cell is generated for a specific combination of the addressA, addressB, and data conditions; the value of those fields is given by the EDR bits. If all three bits of the EDR are reset, then it did not generate any debug request. If the cell did generate a debug request, then at least one EDR bit is set; the meaning of the encoding is as follows: 0 1 matched addressA condition 1 1 matched addressB condition 2 1 matched data condition

7.2.9.18 Channel 0 Boot Address (SDMACOREx_MCHN0ADDR)

Address: Base address + 48h offset

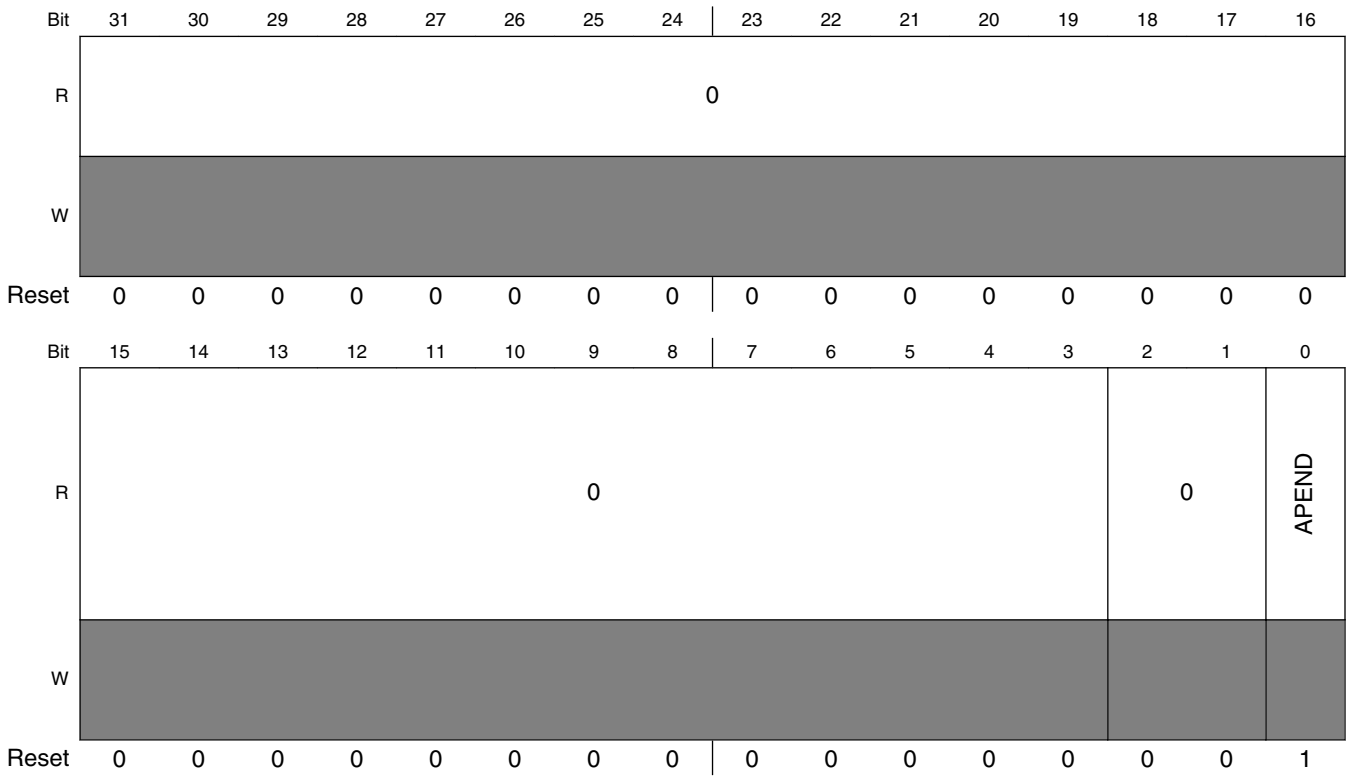


SDMACOREx_MCHN0ADDR field descriptions

Field	Description
31–15 Reserved	This read-only field is reserved and always has the value 0.
14 SMSZ	The bit 14 (Scratch Memory Size) determines if scratch memory must be available after every channel context. After reset, it is equal to 0, which defines a RAM space of 24 words for each channel. All of this area stores the channel context. By setting this bit, 32 words are reserved for every channel context, which gives eight additional words that can be used by the channel script to store any type of data. Those words are never erased by the context switching mechanism. 0 24 words per context 1 32 words per context
CHN0ADDR[13:0]	Contains the address of the channel 0 routine programmed by the Arm platform; it is loaded into a general register at the very start of the boot and the SDMA jumps to the address it contains. By default, it points to the standard boot routine in ROM.

7.2.9.19 ENDIAN Status Register (SDMACOREx_ENDIANNESS)

Address: Base address + 4Ch offset

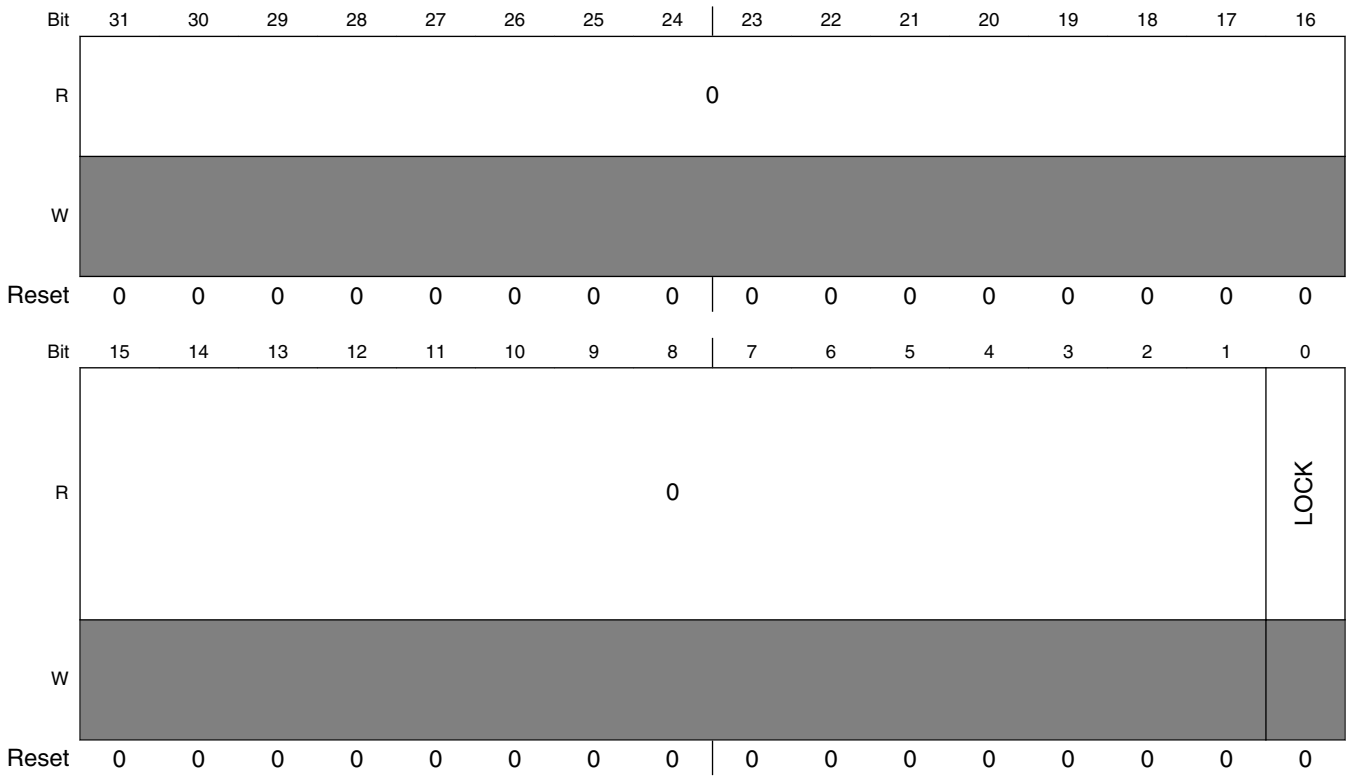


SDMACOREx_ENDIANNESS field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 APEND	APEND indicates the endian mode of the Peripheral and Burst DMA interfaces. This bit is tied to logic '1' indicating little-endian mode. 0 - Arm platform is in big-endian mode 1 - Arm platform is in little-endian mode

7.2.9.20 Lock Status Register (SDMACOREx_SDMA_LOCK)

Address: Base address + 54h offset

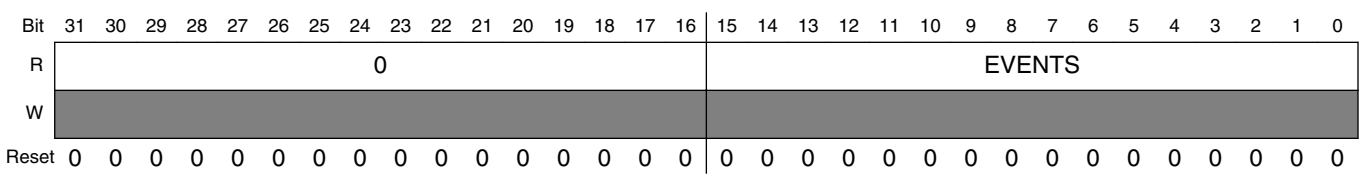


SDMACOREx_SDMA_LOCK field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 LOCK	The LOCK bit reports the value of the LOCK bit in the SDMA_LOCK status register. SDMA software may use this value to determine if certain operations such as loading of new scripts is allowed. 0 - LOCK bit clear 1 - LOCK bit set

7.2.9.21 External DMA Requests Mirror #2 (SDMACOREx_EVENTS2)

Address: Base address + 58h offset



SDMACOREx_EVENTS2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
EVENTS	Reflects the status of the SDMA's external DMA requests. It is meant to allow any channel to monitor the states of these SDMA inputs. This register displays EVENTS 32-47. The separate EVENTS register displays events 0-31.

7.2.10 SDMA Peripheral Registers

Refer to the respective peripherals' chapters for more information.

7.3 Enhanced Direct Memory Access (eDMA)

7.3.1 Introduction

The enhanced direct memory access (eDMA) controller is capable of performing complex data transfers with minimal intervention from a host processor. The hardware microarchitecture includes:

- A DMA engine that performs:
  - Source address and destination address calculations
  - Data-movement operations
- Local memory containing transfer control descriptors for each of the 32 channels

7.3.1.1 eDMA system block diagram

Figure 7-23 illustrates the components of the eDMA system, including the eDMA module (engine).

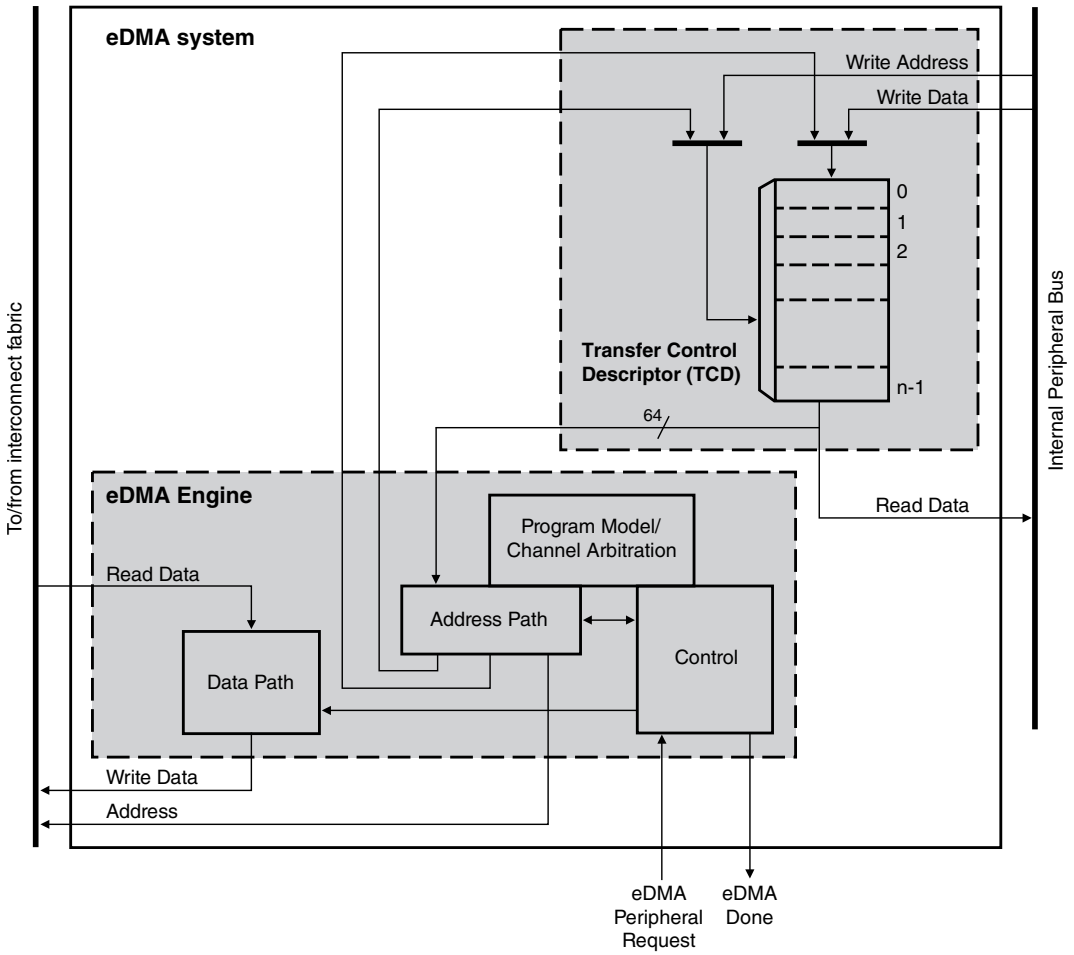


Figure 7-23. eDMA system block diagram

7.3.1.2 Block parts

The eDMA module is partitioned into two major modules: the eDMA engine and the transfer control descriptor local memory.

The eDMA engine is further partitioned into four submodules:

Table 7-58. eDMA engine submodules

Submodule	Function
Address path	This block: • Implements a primary channel and secondary (preempt) channel • Manages all master bus-address calculations All the channels provide the same functionality. This structure allows data transfers associated with one channel to be preempted after the completion of a read/write sequence if a higher priority channel activation is asserted while the primary channel is active. After a channel is activated, it runs until the minor loop is completed, unless preempted by a higher priority channel. This provides a mechanism (enabled by CH*n*_PRI[ECP]) where a large data transfer can be preempted to minimize the time another channel is blocked from execution.

Table continues on the next page...

Table 7-58. eDMA engine submodules (continued)

Submodule	Function
	When any channel is selected to execute, the contents of its TCD are read from local memory and loaded into the address path channel x registers for a normal start and into channel y registers for a preemption start. After the minor loop completes execution, the address path hardware writes the new values for the TCD*n*_{SADDR, DADDR, CITER} back to local memory. If the major iteration count is exhausted, additional processing is performed, including the final address pointer updates, reloading the TCD*n*_CITER field, and a possible fetch of a new TCD*n* from memory as part of a scatter/gather operation. See Dynamic scatter/gather for more details.
Data path	This block implements the bus master read/write data path. It includes a data buffer and the necessary multiplex logic to support any required data alignment. The internal read data bus is the primary input, and the internal write data bus is the primary output. The address and data path modules directly support the 2-stage pipelined internal bus. The address path module represents the first stage of the bus pipeline (address phase), and the data path module implements the second stage of the pipeline (data phase).
Program model/channel arbitration	This block implements the first section of the eDMA programming model as well as the channel arbitration logic. The programming model registers are connected to the internal peripheral bus. The eDMA peripheral request inputs and interrupt request outputs are also connected to this block (via control logic).
Control	This block provides all the control functions for the eDMA engine. For data transfers where the source and destination sizes are equal, the eDMA engine performs a series of source read/ destination write operations until the number of bytes specified in the minor loop byte count has been moved from the source to the destination. For descriptors where the sizes are not equal, multiple accesses of the smaller size data are required for each reference of the larger size. As an example, if the source size references 16-bit data and the destination is 32-bit data, the eDMA performs two reads, then one 32-bit write.

The transfer control descriptor local memory is further partitioned into:

Table 7-59. Transfer control descriptor memory

Submodule	Description
Memory controller	This logic implements the required dual-ported controller, and manages accesses from the eDMA engine as well as references from the internal peripheral bus. As noted earlier, in simultaneous accesses, the eDMA engine is given priority and the peripheral transaction is stalled.
Memory array	TCD storage for each channel's transfer profile.

7.3.1.3 Features

The eDMA is a highly programmable data-transfer engine optimized to minimize any required intervention from the host processor. It is intended for use in applications where the data size to be transferred is statically known and not defined within the transferred data itself. The eDMA module features:

- All data movement via dual-address transfers: read from source, write to destination

    - Programmable source and destination addresses and transfer size
    - Support for complex address calculations
- 32-channel implementation that performs complex data transfers with minimal intervention from a host processor
    - Internal data buffer, used as temporary storage for all transfers
    - Connections to the crossbar switch for bus mastering the data movement
- TCD organized to support two-deep, nested transfer operations
    - 32-byte TCD stored in local memory for each channel
    - An inner data transfer loop defined by a minor byte transfer count
    - An outer data transfer loop defined by a major iteration count
- Channel activation via one of three methods:
    - Explicit software initiation
    - Initiation via a channel-to-channel linking mechanism for continuous transfers
    - Peripheral-paced hardware requests, one per channel
- Fixed-priority and round-robin channel arbitration
- Channel completion reported via programmable interrupt requests
    - One interrupt per channel, which can be asserted at completion of major iteration count
    - Programmable error terminations per channel that are logically summed together to form one error interrupt to the interrupt controller
- Programmable support for scatter/gather DMA processing
- Support for complex data structures

In the discussion of this module, *n* is used to reference the channel number.

7.3.2 Modes of operation

The eDMA operates in the following modes:

Table 7-60. Modes of operation

Mode	Description
Normal	In Normal mode, eDMA transfers data between a source and a destination. The source and destination can be a memory block or an I/O block capable of operation with eDMA. A service request initiates a transfer of a specific number of bytes (NBYTES) as specified in the TCD. The minor loop is the sequence of read-write operations that transfers these NBYTES per service request. Each service request executes one iteration of the major loop, which transfers NBYTES of data.
Debug	eDMA operation is configurable in Debug mode via the control register: • If CSR[EDBG] is cleared to 0, eDMA continues to operate. • If CSR[EDBG] is set to 1, eDMA stops transferring data. If Debug mode is entered when a channel is active, eDMA continues operation until the channel retires.

7.3.3 Functional description

The operation of eDMA is described in the following subsections.

7.3.3.1 eDMA basic data flow

The basic flow of a data transfer can be partitioned into three segments.

As shown in the following diagram, the first segment involves the channel activation:

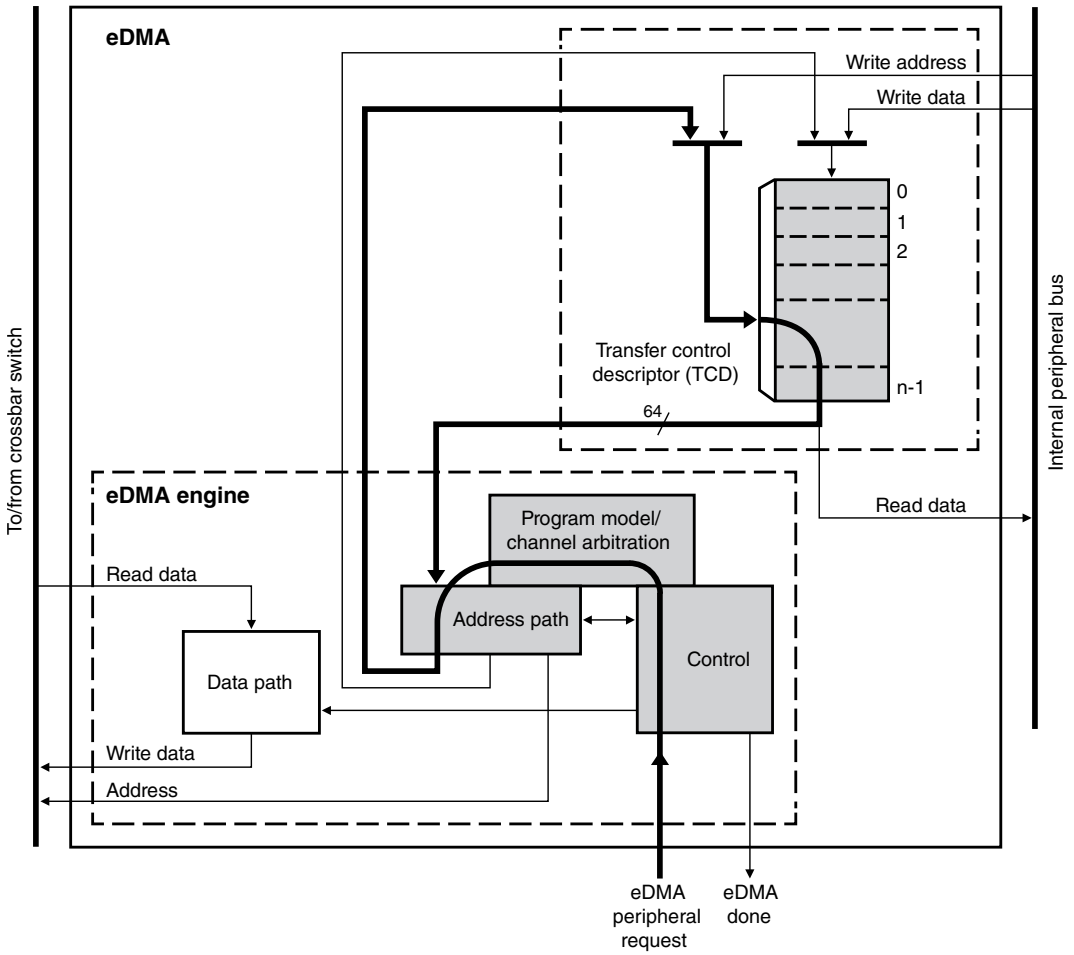


Figure 7-24. eDMA operation, part 1

This example uses the assertion of the eDMA peripheral request signal to request service for channel *n*. Channel activation via software and the TCD*n*_CSR[START] field follows the same basic flow as peripheral requests. The eDMA request input signal is registered internally and then routed through the eDMA engine: first through the control module, then into the program model and channel arbitration.

In the next cycle, the channel arbitration begins using fixed-priority plus the optional round-robin algorithm. After arbitration is complete, the activated channel number is sent through the address path and converted into the required address to access the local memory for TCD*n*. Next, the TCD memory is accessed and the required descriptor is read from the local memory and then loaded into the eDMA engine address path's primary or secondary channel execution registers. The TCD memory is 64 bits wide to minimize the time needed to fetch the activated channel descriptor and load it into the address path registers.

The following diagram illustrates the second part of the basic data flow:

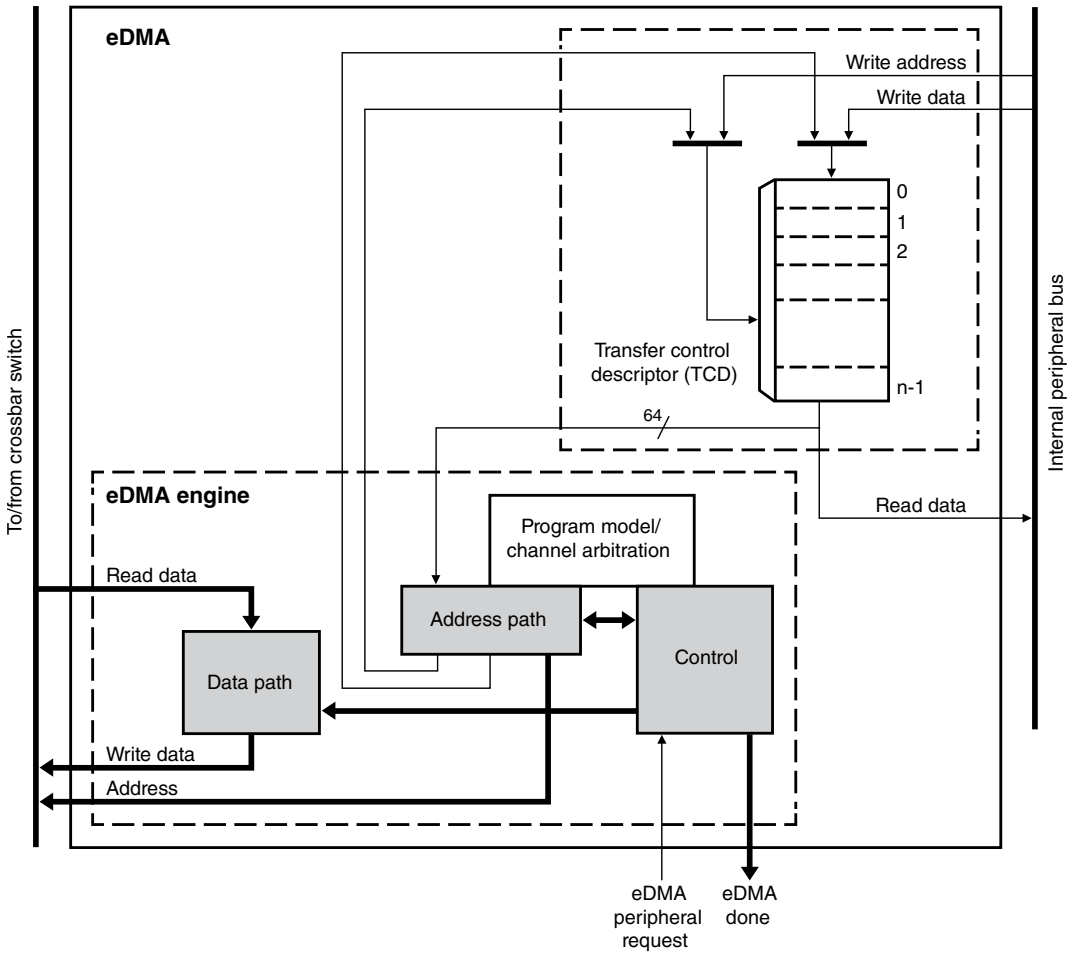


Figure 7-25. eDMA operation, part 2

The modules associated with the data transfer (address path, data path, and control) go through the required sequence of source reads and destination writes to perform the actual data movement. The source reads are initiated, and the fetched data is temporarily stored in the data path block until it is gated onto the internal bus during the destination write. This source read/destination write processing continues until the byte count, NBYTES, has been transferred.

After NBYTES of data has been moved, the final phase of the basic data flow is performed. In this segment, the address path logic performs the required updates to certain fields in the appropriate TCD (for example, SADDR, DADDR, CITER). If the major iteration count is exhausted, additional operations are performed. These include the final address adjustments and reloading of the BITER field into the CITER field. Assertion of an optional interrupt request also occurs at this time, as does a possible fetch of a new TCD from memory using the scatter/gather address pointer included in the descriptor (if scatter/gather is enabled). The updates to the TCD memory and the assertion of an interrupt request are shown in the following diagram.

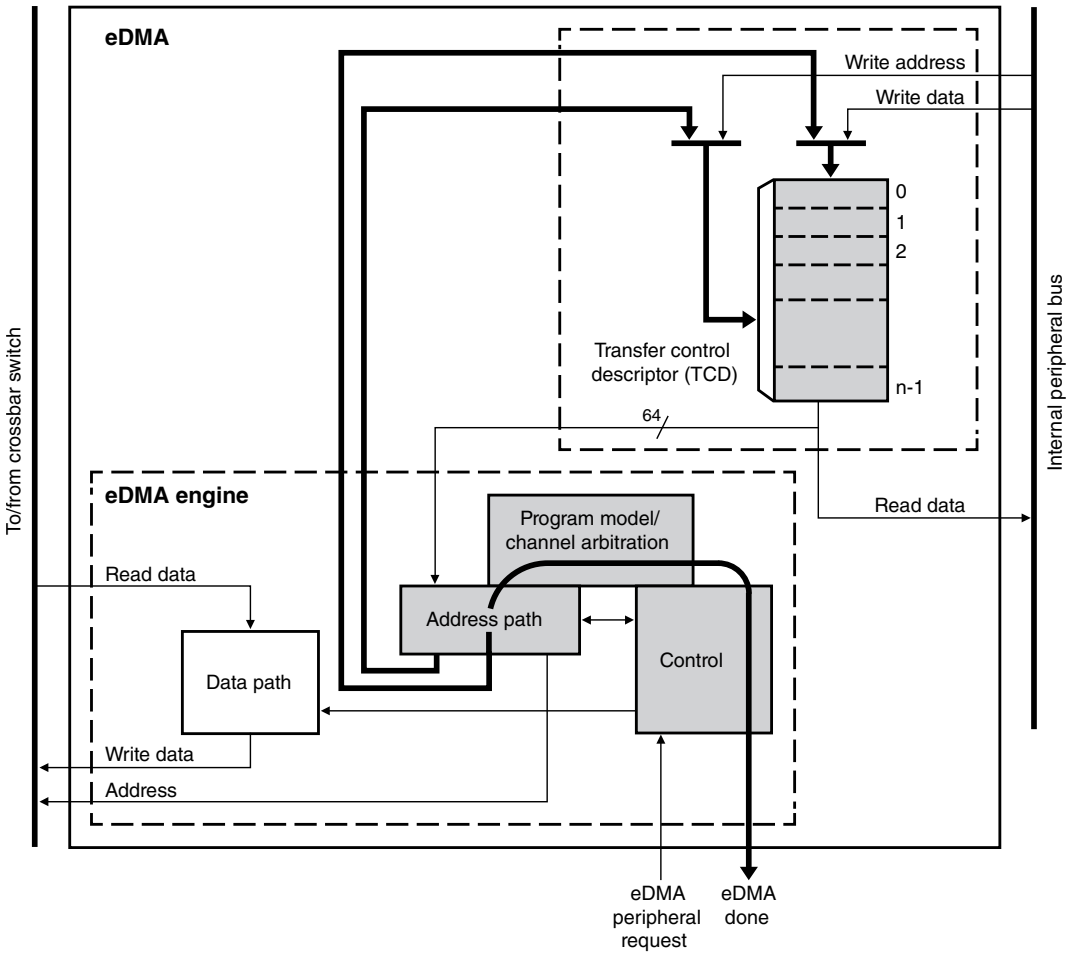


Figure 7-26. eDMA operation, part 3

7.3.3.2 Fault reporting and handling

Channel errors are reported in the Error Status register (CHn_CSR and TCDn_CSR) and can be caused by any of the following:

- A configuration error, which is an illegal setting in the transfer control descriptor
- An active channel canceled via a "cancel transfer with error" hardware or software request
- An error termination to a bus master read or write cycle

A configuration error is reported when an inconsistent state is represented by one of these factors:

- Starting source or destination address
- Source or destination offsets
- Minor loop byte count
- Transfer size

Each of these possible causes is detailed below:

- The addresses and offsets must be aligned on zero-modulo-transfer-sized boundaries.
- The minor loop byte count must be a multiple of the source and destination transfer sizes.
- All source reads and destination writes must be configured to the natural boundary of the programmed transfer size.

Note

To aid in debugging, set the Halt After Error field in the DMA's Control Status register, CSR[HAE]. Upon any error condition, the DMA is halted after the error is recorded. The DMA remains halted and does not process any channel service requests. After the error is fixed, the DMA may be enabled again by clearing the Halt field, CSR[HALT].

- If a scatter/gather operation is enabled upon channel completion, a configuration error is reported if the scatter/gather address (DLAST_SGA) is not aligned on a 32-byte boundary.
- If minor loop channel linking is enabled upon channel completion, a configuration error is reported when the link is attempted if the TCDn_CITER[ELINK] field does not equal the TCDn_BITER[ELINK] field.

If enabled, all configuration error conditions, except the scatter/gather and minor-loop link errors, are reported as the channel activates and asserts an error interrupt request. A scatter/gather configuration error is reported when the scatter/gather operation begins at major loop completion if properly enabled. A minor loop channel link configuration error is reported when the link operation is serviced at minor loop completion.

The occurrence of any error causes the eDMA engine to stop normal processing of the active channel immediately (it goes to its error processing states and the transaction to the system bus still has pipeline effect), and the appropriate channel field in the eDMA error register is set to 1. At the same time, the details of the error condition are loaded into the Error Status register (CHn_CSR and TCDn_CSR). The major loop complete indicators, setting the transfer control descriptor DONE flag, and the possible assertion of an interrupt request are not affected when an error is detected.

After the error status has been updated, the eDMA engine continues operating by servicing the next appropriate channel. A channel that experiences an error condition is not automatically disabled. If a channel is terminated by an error and then issues another service request before the error is fixed, that channel executes and terminates with the same error condition.

The error status fields are read-only. These error indicators are sticky and cannot be cleared. They show the last recorded error until the DMA is reset. The valid field (VLD) is used to determine if a new error condition exists. This field is the logical OR of each channel's error interrupt field (ERR).

After the software has resolved any errors and cleared all of the error interrupt fields, the valid field is cleared to 0 but the cause of the last error is still indicated.

7.3.3.3 Channel preemption

Channel preemption is enabled on a per-channel basis by setting the CHn_PRI[ECP] field. Channel preemption allows the executing channel’s data transfers to temporarily suspend in favor of starting a higher-priority channel. After the preempting channel has completed all of its minor loop data transfers, the preempted channel is restored and resumes execution.

After the restored channel completes one read/write sequence, it is again eligible for preemption. If any higher priority channel is requesting service, the restored channel is suspended, and the higher-priority channel is serviced. Nested preemption, that is, attempting to preempt a preempting channel, is not supported. After a preempting channel begins execution, it cannot be preempted.

A channel’s ability to preempt another channel can be disabled by setting CHn_PRI[DPA] to 1. When a channel’s preempt ability is disabled, that channel cannot suspend a lower-priority channel’s data transfer, regardless of the lower-priority channel’s ECP setting. This allows for a pool of low-priority, large-data-moving channels to be defined.

You can configure these low-priority channels to not preempt each other, thus preventing a low-priority channel from consuming the preempt slot normally available to a true high-priority channel. When you enable round-robin channel arbitration mode (CSR[ERCA] is set to 1), any channel with a priority level equal to 0 (CHn_PRI[APL] = 0) has preemption disabled and cannot preempt another channel.

7.3.4 Initialization/application information

The following sections discuss initialization of the eDMA and programming considerations.

7.3.4.1 eDMA initialization

To initialize the eDMA:

1. Write to the MP_CSR if a configuration other than the default is wanted.
2. Write the channel priority levels to the CHn_PRI registers and group priority levels to the CHn_GRPRI registers if a configuration other than the default is wanted.
3. Enable error interrupts in the CHn_CSR[EEI] registers if they are wanted.
4. Write the 32-byte TCD for each channel that may request service.
5. Enable any hardware service requests via the CHn_CSR[ERQ] registers.
6. Request channel service via either:
   - Software: setting TCDn_CSR[START]
   - Hardware: slave device asserting its eDMA peripheral request signal

After any channel requests service, a channel is selected for execution based on the arbitration and priority levels written into the programmer's model. The eDMA engine reads the entire TCD, including the TCD control and status fields, as shown in Table 7-61, for the selected channel into its internal address path module.

As the TCD is read, the first transfer is initiated on the internal bus, unless a configuration error is detected. Transfers from the source, defined by TCDn_SADDR, to the destination, defined by TCD_DADDR, continue until the number of bytes specified by TCDn_NBYTES are transferred.

When the transfer is complete, the eDMA engine's local TCDn_SADDR, TCDn_DADDR, and TCDn_CITER are written back to the main TCD memory and any minor loop channel linking is performed, if enabled. If the major loop is exhausted, then eDMA executes further post-processing, such as interrupts, major loop channel linking, and scatter/gather operations, if enabled.

Table 7-61. TCD control and status (TCDn_CSR) fields

TCDn_CSR field name	Description
START	Control field to start the channel explicitly when using a software-initiated DMA service (automatically cleared by hardware)
EEOP	Control field to enable end-of-packet processing
ESDA	Control field to enable storing of the destination address to system memory after the major loop completes
DREQ	Control field to disable hardware-initiated DMA service requests after major loop completion
BWC	Control field for throttling the bandwidth control of a channel
ESG	Control field to enable the scatter-gather feature

Table continues on the next page...

Table 7-61. TCD control and status (TCDn_CSR) fields (continued)

TCDn_CSR field name	Description
INTHALF	Control field to enable interrupt when major loop is half-complete
INTMAJOR	Control field to enable interrupt when major loop completes

Table 7-62. Channel control and status (CHn_CSR) fields

CHn_CSR field name	Description
ACTIVE	Status field indicating the channel is currently in execution
DONE	Status field indicating major loop completion (cleared by software when a channel begins execution)
EEI	Control field to enable error interrupts
EARQ	Control field to enable external, asynchronous wakeup event in conjunction with the ERQ field
ERQ	Control field to enable hardware service requests

The following figure shows how each DMA request initiates one minor-loop transfer, or iteration, without CPU intervention. DMA arbitration can occur after each minor loop, and one level of minor loop DMA preemption is allowed. The number of minor loops in a major loop is specified by the beginning iteration count (BITER).

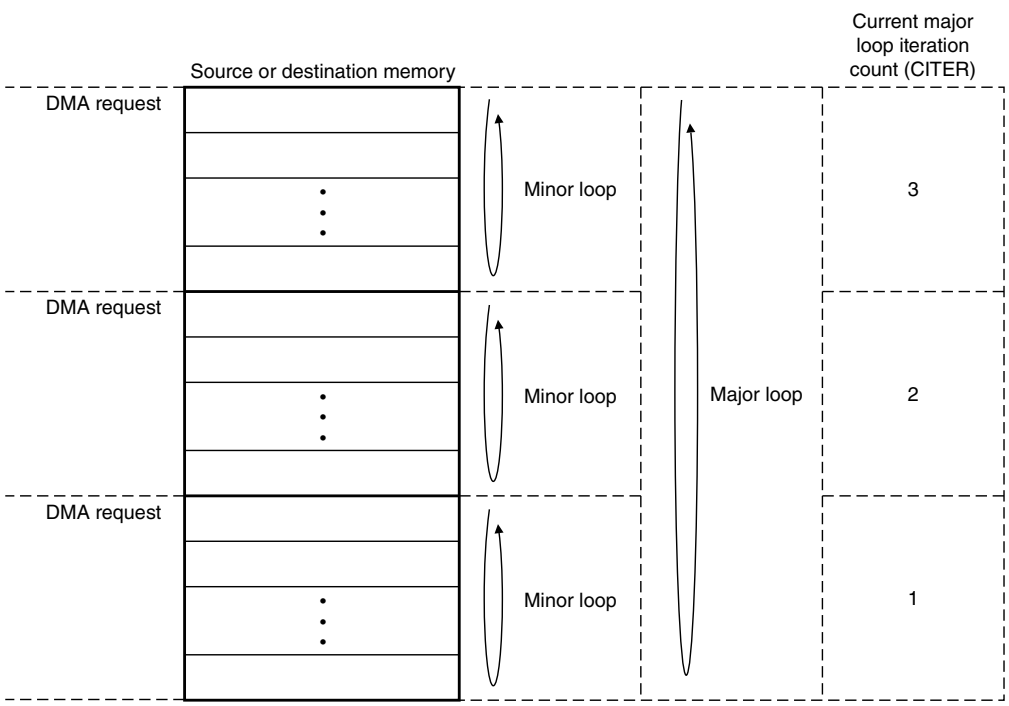


Figure 7-27. Example of multiple loop iterations

The following figure lists the memory array terms and how the TCD settings are related.

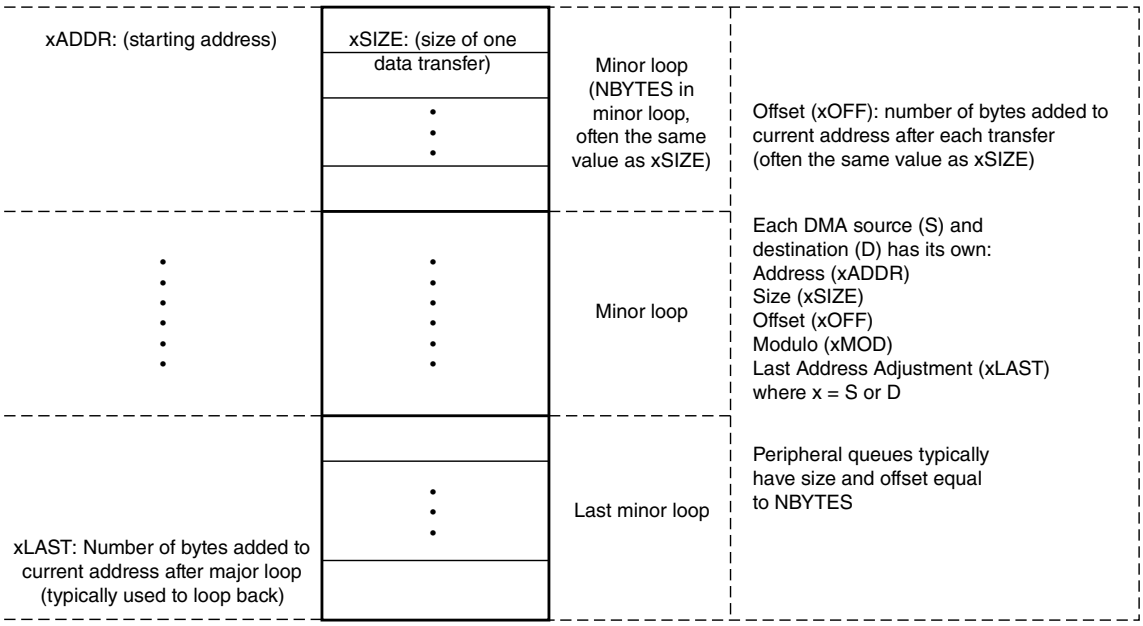


Figure 7-28. Memory array terms

7.3.4.2 eDMA arbitration

The eDMA uses a layered arbitration scheme composed of multiple priority levels. The eDMA uses a fixed-priority arbitration scheme with optional round-robin arbitration under specific conditions. The priorities are evaluated in the following order:

Table 7-63. eDMA arbitration priorities

Priority	Scheme	Description
1 (Highest)	Arbitration group priority	Each channel is assigned an arbitration group via the CHn_GRPRI registers. Priority is given to the highest value (31 being the highest possible value) down to the lowest value (zero, the default).
2	Channel priority	Each channel is assigned a channel priority level via the CHn_PRI registers. The channel priority is a relative priority level within an arbitration group. Priority is given to the highest value (seven being the highest possible value) down to the lowest value (zero, the default). Channel priorities within each arbitration group need not be unique. If multiple channels have the same channel priority level, the channel number will be used to determine priority as defined in row three.

Table continues on the next page...

Table 7-63. eDMA arbitration priorities (continued)

Priority	Scheme	Description
3	Channel number	When two or more channels have the same arbitration group priority and channel priority, the channel number (CHn_NUM) is used to determine the highest priority. Priority is giver to the highest channel number. Lowest priority is channel 0. The channel numbers are static and cannot be changed in the programmer's model.
4 (Lowest)	Round-robin	When round-robin is enabled, any channel configured for round-robin operation has lowest priority within an arbitration group. Round-robin is enabled by setting the MP_CSR[ERCA] field to 1. After being enabled, channels with a channel priority of zero (CHn_PRI=0) will use round-robin arbitration. Round-robin arbitration will rotate the channel selection among the channels requesting service with CHn_PRI=0 within the arbitration group. Any non-zero channel within the arbitration group will continue to use fixed-priority arbitration, and if requesting service will be selected over any round-robin channels.

For fixed arbitration, the overall priority can be considered a number composed of three concatenated priority levels: CHn_GRPRI :CHn_PRI:CH_NUM. The largest number has the highest priority and the lowest number has the lowest priority.

For round-robin arbitration, the priority number is CHn_GRPRI :0:X. The module rotates through the CHn_PRI=0 channels requesting service without regard to priority among these channels. Any channel within the arbitration group for which CHn_PRI is greater than 0 will be serviced before the round-robin channels.

7.3.4.3 Programming errors

The eDMA performs various tests on the transfer control descriptor to verify consistency in the descriptor data.

The channel number causing the error is recorded in the Error Status register (CHn_CSR and TCDn_CSR). If the error source is not removed before the next activation of the problematic channel, the error is detected and recorded again. Setting the halt after error field, CSR[HAE], will halt the DMA and prevent reoccurrence of the error.

7.3.4.4 Arbitration mode considerations

This section discusses arbitration considerations for eDMA.

7.3.4.4.1 Fixed group arbitration, fixed channel arbitration

In this mode, eDMA selects for execution the channel service request from the highest-priority channel in the highest-priority group. If eDMA is programmed so that the channels within a high-priority group have a high number of requests or large data transfers, that group may consume all the bandwidth of the eDMA controller. That is, no lower-priority groups are serviced if there is always at least one DMA request pending on a channel in the highest-priority group when the controller arbitrates the next DMA request. The advantage of this scenario is that latency can be small for channels that need to be serviced quickly.

7.3.4.4.2 Fixed group arbitration, round-robin channel arbitration

The highest-priority group with a request is serviced. Lower-priority groups are serviced if no pending requests exist in the higher-priority groups.

Within each group, channels are serviced starting with the highest non-zero channel priority. For all channels with a channel priority programmed to 0, selection begins with the highest channel number requesting service and then rotates through to the lowest channel number requesting service. The round-robin channel arbitration can provide a fairness mechanism to lower-priority channels.

This scenario could cause the same bandwidth consumption problem as indicated in Fixed group arbitration, fixed channel arbitration, but all the channels in the highest-priority group will be serviced. Service latency is short on the highest-priority group, but could potentially be very much longer as the group priority decreases.

7.3.4.5 Performing DMA transfers

This section presents examples on how to perform DMA transfers with the eDMA.

7.3.4.5.1 Single request

To perform a simple transfer of n bytes of data with one activation, set the major loop to one (TCD*n*_CITER = TCD*n*_BITER = 1). The data transfer begins after the channel service request is acknowledged and the channel is selected to execute. After the transfer is complete, the CH*n*_CSR[DONE] field is set to 1 and an interrupt is generated if properly enabled.

For example, the following TCD entry is configured to transfer 16 bytes of data. The eDMA is programmed for one iteration of the major loop transferring 16 bytes per iteration. The source memory has a byte-wide memory port located at 0x1000. The destination memory has a 32-bit port located at 0x2000. The address offsets are programmed in increments to match the transfer size: one byte for the source, and four bytes for the destination. The final source and destination addresses are adjusted to return to their beginning values.

```
TCDn_CITER = TCDn_BITER = 1
TCDn_NBYTES = 16
TCDn_SADDR = 0x1000
TCDn_SOFF = 1
TCDn_ATTR[SSIZE] = 0
TCDn_SLAST = -16
TCDn_DADDR = 0x2000
TCDn_DOFF = 4
TCDn_ATTR[DSIZE] = 2
TCDn_DLAST_SGA= -16
TCDn_CSR[INTMAJ] = 1
TCDn_CSR[START] = 1 (should be written last after all other fields have been initialized)
All other TCDn fields = 0
```

This generates the following event sequence:

1. User write to the TCD*n*_CSR[START] field requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes:
   - CH*n*_CSR[DONE] = 0
   - TCD*n*_CSR[START] = 0
   - CH*n*_CSR[ACTIVE] = 1
4. eDMA engine reads: channel TCD data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:
   a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
   b. Write 32 bits to location 0x2000 → first iteration of the minor loop.
   c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.

d. Write 32 bits to location 0x2004 → second iteration of the minor loop.

e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.

f. Write 32 bits to location 0x2008 → third iteration of the minor loop.

g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.

h. Write 32 bits to location 0x200C → last iteration of the minor loop → major loop complete.

6. The eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 1 (TCD*n*_BITER).
7. The eDMA engine writes: CH*n*_CSR[ACTIVE] = 0, CH*n*_CSR[DONE] = 1, CH*n*_INT[INT] = 1.
8. The channel retires and the eDMA goes idle or services the next channel.

7.3.4.5.2 Multiple requests

The following example transfers 32 bytes via two hardware requests, but is otherwise the same as the previous example. The only fields that change are the major loop iteration count and the final address offsets. The eDMA is programmed for two iterations of the major loop, transferring 16 bytes per iteration. After the channel's hardware requests are enabled via the CHn_CSR[ERQ] register field, the slave device initiates channel service requests.

```
TCDn_CITER = TCDn_BITER = 2
TCDn_SLAST = -32
TCDn_DLAST_SGA = -32
```

This would generate the following sequence of events:

1. First hardware (eDMA peripheral) requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: CH*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, CH*n*_CSR[ACTIVE] = 1.
4. eDMA engine reads: channel TCD*n* data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:

a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.

b. Write 32 bits to location 0x2000 → first iteration of the minor loop.

c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.

d. Write 32 bits to location 0x2004 → second iteration of the minor loop.

e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.

f. Write 32 bits to location 0x2008 → third iteration of the minor loop.

g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.

h. Write 32 bits to location 0x200C → last iteration of the minor loop.

6. eDMA engine writes: TCD*n*_SADDR = 0x1010, TCD*n*_DADDR = 0x2010, TCD*n*_CITER = 1.
7. eDMA engine writes: CH*n*_CSR[ACTIVE] = 0.
8. The channel retires, which concludes one iteration of the major loop. The eDMA goes idle or services the next channel.
9. Second hardware (eDMA peripheral) requests channel service.
10. The channel is selected by arbitration for servicing.
11. eDMA engine writes: CH*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, CH*n*_CSR[ACTIVE] = 1.
12. eDMA engine reads: Channel TCD data from local memory to internal register file.
13. The source-to-destination transfers are executed as follows:

    a. Read byte from location 0x1010, read byte from location 0x1011, read byte from 0x1012, read byte from 0x1013.

    b. Write 32 bits to location 0x2010 → first iteration of the minor loop.

    c. Read byte from location 0x1014, read byte from location 0x1015, read byte from 0x1016, read byte from 0x1017.

    d. Write 32 bits to location 0x2014 → second iteration of the minor loop.

e. Read byte from location 0x1018, read byte from location 0x1019, read byte from 0x101A, read byte from 0x101B.

f. Write 32 bits to location 0x2018 → third iteration of the minor loop.

g. Read byte from location 0x101C, read byte from location 0x101D, read byte from 0x101E, read byte from 0x101F.

h. Write 32 bits to location 0x201C → last iteration of the minor loop → major loop complete.

14. eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 2 (TCD*n*_BITER).

15. eDMA engine writes: CH*n*_CSR[ACTIVE] = 0, CH*n*_CSR[DONE] = 1, CH*n*_INT[INT] = 1.

16. The channel retires, which concludes with the major loop complete. The eDMA goes idle or services the next channel.

7.3.4.5.3 Using the modulo feature

The modulo feature of the eDMA allows implementation of a circular data queue in which the size of the queue is a power of 2. xMOD is a 5-bit field for the source and destination in the TCD, and it specifies which lower address bits increment from their original value after the address+offset calculation. All upper address bits remain the same as in the original value. A setting of 0 for this field disables the modulo feature. Modulo addressing applies to cases where the minor loop offset is enabled; that is, the upper address bits remain the same after the minor loop offset is added to the source or destination address.

The following table shows how the transfer addresses are specified based on the setting of the MOD field. Here a circular buffer is created where the address wraps to the original value but the 28 upper address bits (0x1234567*x*) retain their original value. In this example, the source address is set to 0x12345670, the offset is set to four bytes, and the MOD field is set to four, which allows for a 2^4 byte (16 byte) queue size.

Table 7-64. Modulo example

Transfer number	Address
1	0x12345670
2	0x12345674
3	0x12345678
4	0x1234567C
5	0x12345670

Table continues on the next page...

Table 7-64. Modulo example (continued)

Transfer number	Address
6	0x12345674

7.3.4.6 Monitoring transfer descriptor status

This section discusses how to monitor eDMA status.

7.3.4.6.1 Testing for minor loop completion

There are two methods to test for minor loop completion when using software-initiated service requests.

1. The first method is to read the TCDn_CITER field and test for a change.
2. The second method, extracted from the sequence shown below, is to test the TCDn_CSR[START] field and the CHn_CSR[ACTIVE] field. The minor-loop-complete condition is indicated by both fields reading 0 after TCDn_CSR[START] is set to 1. Polling the CHn_CSR[ACTIVE] field only may be inconclusive because the active status may be missed if the channel execution is short in duration.

The CHn_CSR and TCDn_CSR status fields execute the following sequence for a software-activated channel:

Stage	TCDn_CSR field	CHn_CSR fields		State
	START	ACTIVE	DONE	
1	1	0	0	Initiate channel service request via software.
2	0	1	0	Channel is executing.
3a	0	0	0	Channel has completed the minor loop and is idle.
3b	0	0	1	Channel has completed the major loop and is idle.

The best method to test for minor-loop completion when using hardware-initiated (that is, peripheral-initiated) service requests is to read the TCDn_CITER field and test for a change. The hardware request and acknowledge handshake signals are not visible in the programmer's model.

The TCD status fields execute the following sequence for a hardware-activated channel:

Stage	TCDn_CSR field	CHn_CSR fields		State
	START	ACTIVE	DONE	
1	0	0	0	Initiate channel service request via hardware (peripheral request asserted).
2	0	1	0	Channel is executing.
3a	0	0	0	Channel has completed the minor loop and is idle.
3b	0	0	1	Channel has completed the major loop and is idle.

For both activation types, the major-loop-complete status is explicitly indicated via the CHn_CSR[DONE] field.

The TCDn_CSR[START] field is cleared to 0 automatically when the channel begins execution, regardless of how the channel activates.

7.3.4.6.2 Reading the transfer descriptors of active channels

The eDMA reads back the true TCDn_SADDR, TCDn_DADDR, and TCDn_NBYTES values if they are read when a channel executes. The true values of SADDR, DADDR, and NBYTES are the values the eDMA engine currently uses in its internal register file, and not the values in the TCD local memory for that channel. The addresses, SADDR and DADDR, and NBYTES (which decrements to zero as the transfer progresses), can give an indication of the progress of the transfer. All other values are read back from the TCD local memory.

7.3.4.6.3 Checking channel preemption status

A preemptive situation is one in which a preempt-enabled channel is executing and a higher-priority request becomes active. When round-robin channel arbitration mode is enabled, all channels with their channel priority set to 0 lose their preempt ability. Channel priorities of 0 are treated as equal, that is, they are constantly rotating, when round-robin arbitration mode is enabled.

The CHn_CSR[ACTIVE] field for the preempted channel remains asserted throughout the preemption. The preempted channel is temporarily suspended when the preempting channel executes one major loop iteration. If two CHn_CSR[ACTIVE] fields are set simultaneously in the global TCD map, a higher-priority channel is actively preempting a lower-priority channel.

7.3.4.7 Channel linking

Channel linking (or chaining) is a mechanism in which one channel sets the TCDn_CSR[START] field of another channel (or itself), thus initiating a service request for that channel. When properly enabled, the eDMA engine automatically performs this operation at the major or minor loop completion.

The minor loop channel linking occurs at the completion of the minor loop (or one iteration of the major loop). The TCDn_CITER[ELINK] field determines whether a minor loop link is requested. When enabled, the channel link is made after each iteration of the major loop except for the last. When the major loop is exhausted, only the major loop channel link fields are used to determine if a channel link should be made. For example, using an initial field setting of:

```
TCDn_CITER[ELINK] = 1
TCDn_CITER[LINKCH] = 0xC
TCDn_CITER[CITER] value = 0x4
TCDn_CSR[MAJORELINK] = 1
TCDn_CSR[MAJORLINKCH] = 0x7
```

executes as:

1. Minor loop done → set TCD12_CSR[START] field
2. Minor loop done → set TCD12_CSR[START] field
3. Minor loop done → set TCD12_CSR[START] field
4. Minor loop done, major loop done→ set TCD7_CSR[START] field

When minor loop linking is enabled (TCDn_CITER[ELINK] = 1), the TCDn_CITER[CITER] field uses a nine-bit vector to form the current iteration count. When minor loop linking is disabled (TCDn_CITER[ELINK] = 0), the TCDn_CITER[CITER] field uses a 15-bit vector to form the current iteration count. The bits associated with the TCDn_CITER[LINKCH] field are concatenated onto the CITER value to increase the range of the CITER.

Note

The TCDn_CITER[ELINK] field and the TCDn_BITER[ELINK] field must be equal — if they are not, a configuration error is reported. The CITER and BITER vector widths must be equal to calculate the major loop halfway done interrupt point.

The following table summarizes how a DMA channel can link to another DMA channel, that is, use another channel's TCD, at the end of a loop.

Table 7-65. Channel linking parameters

Wanted link behavior	TCD control field name	Description
Link at end of minor loop	TCDn_CITER[ELINK]	Enable channel-to-channel linking on minor loop completion (current iteration)
	TCDn_CITER[LINKCH]	Link channel number when linking at end of minor loop (current iteration)
Link at end of major loop	TCDn_CSR[MAJORELINK]	Enable channel-to-channel linking on major loop completion
	TCDn_CSR[MAJORLINKCH]	Link channel number when linking at end of major loop

7.3.4.8 Dynamic programming

This section provides recommended methods to change the programming model during channel execution.

7.3.4.8.1 Dynamically changing the channel priority

To change group or channel priority levels:

1. Halt the DMA by writing 1 to the CSR[HALT] field.
2. Change the group or channel priorities as wanted.
3. Enable normal DMA operations by writing 0 to the CSR[HALT] field.

7.3.4.8.2 Dynamic channel linking

Dynamic channel linking is the process of setting the TCDn_CSR[MAJORELINK] field during channel execution (see the diagram in TCD structure). This field is read from the TCD local memory at the end of channel execution, thus allowing you to enable the feature during channel execution.

Because you are allowed to change the configuration during execution, you need a coherency model. Consider the scenario where you attempt to execute a dynamic channel link by enabling the TCDn_CSR[MAJORELINK] field at the same time the eDMA engine is retiring the channel. TCDn_CSR[MAJORELINK] would be set in the programmer's model, but it would be unclear whether the actual link was made before the channel retired.

We recommend that you use the following coherency model when executing a dynamic channel link request.

1. Write 1 to the TCDn_CSR[MAJORELINK] field.
2. Read back the TCDn_CSR[MAJORELINK] field.
3. Test the TCDn_CSR[MAJORELINK] request status:

- If TCDn_CSR[MAJORELINK] = 1, the dynamic link attempt was successful.
- If TCDn_CSR[MAJORELINK] = 0, the attempted dynamic link did not succeed (the channel was already retiring).

For this request, the TCD local memory controller forces the TCDn_CSR[MAJORELINK] field to 0 on any writes to a channel's TCDn_CSR[7:0] after that channel's CHn_CSR[DONE] field is set to 1, indicating the major loop is complete.

NOTE

You must clear the CHn_CSR[DONE] field to 0 before writing to the TCDn_CSR[MAJORELINK] field. The CHn_CSR[DONE] field is cleared to 0 automatically by the eDMA engine after a channel begins execution.

7.3.4.8.3 Dynamic scatter/gather

Scatter/gather is the process of automatically loading a new TCD into a channel. It allows a DMA channel to use multiple TCDs; this enables a DMA channel to scatter the DMA data to multiple destinations or gather it from multiple sources. When scatter/gather is enabled and the channel has finished its major loop, a new TCD is fetched from system memory and loaded into that channel's descriptor location in the eDMA programmer's model, thus replacing the current descriptor.

Because you are allowed to change the configuration during execution, you need a coherency model. Consider the scenario where you attempt to execute a dynamic scatter/gather operation by enabling the TCDn_CSR[ESG] field at the same time the eDMA engine is retiring the channel. The TCDn_CSR[ESG] field would be set in the programmer's model, but it would be unclear whether the actual scatter/gather request was honored before the channel retired.

Two methods are recommended for executing a dynamic scatter/gather request. Whenever the TCDn_CSR is written, the TCD local memory controller forces the TCDn_CSR[ESG] field to 0 on any writes to a channel's TCDn_CSR [7:0] after that channel's CHn_CSR[DONE] field has been set to 1, indicating the major loop is complete. If attempting to set the ESG, ensure the DONE field is cleared to 0.

NOTE

You must clear the CHn_CSR[DONE] field to 0 before writing the TCDn_CSR[MAJORELINK] or TCDn_CSR[ESG] fields. The CHn_CSR[DONE] field is cleared to 0 automatically by the eDMA engine after a channel begins execution and is set to 1 upon major loop completion.

7.3.4.8.3.1 Method 1 (channel not using major loop channel linking)

For a channel not using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request.

When the TCDn_CSR[MAJORELINK] field is 0, the TCDn_CSR[MAJORLINKCH] field is not used by the eDMA. In this case, the TCDn_CSR[MAJORLINKCH] bits may be used for other purposes. This method uses the TCDn_CSR[MAJORLINKCH] field as a TCDn_CSR identification (ID).

When the descriptors are built, write a unique TCDn_CSR ID in the TCDn_CSR[MAJORLINKCH] field for each TCDn_CSR associated with a channel using dynamic scatter/gather.

1. Write a 1 to the TCDn_CSR[DREQ] field. Should a dynamic scatter/gather attempt fail, setting the TCDn_CSR[DREQ] field to 1 will prevent future hardware activation of this channel. This stops the channel from executing with a destination address (daddr) that was calculated using a scatter/gather address (written in the next step) instead of a DLAST final offset value.
2. Write the TCDn_DLAST_SGA field with the scatter/gather address.
3. Write a 1 to the TCDn_CSR[ESG] field.
4. Read back the 16-bit TCDn_CSR control/status field.
5. Test the TCDn_CSR[ESG] request status and TCDn_CSR[MAJORLINKCH] value:
   - If ESG = 1, the dynamic scatter/gather attempt was successful.
   - If ESG = 0 and the MAJORLINKCH (ID) did not change, the dynamic scatter/gather attempt was not successful (the channel was already retiring).
   - If ESG = 0 and the MAJORLINKCH (ID) changed, the dynamic scatter/gather attempt was successful (the new TCDn_CSR's ESG value cleared the ESG field to 0).

7.3.4.8.3.2 Method 2 (channel using major loop channel linking)

For a channel using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request. This method uses the TCDn_DLAST_SGA field as a TCD identification (ID).

1. Write a 1 to the TCDn_CSR[DREQ] field. Should a dynamic scatter/gather attempt fail, setting the DREQ field to 1 will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (DADDR) that was calculated using a scatter/gather address (written in the next step) instead of a DLAST final offset value.
2. Write the TCDn_DLAST_SGA field with the scatter/gather address.

3. Write a 1 to the TCDn_CSR[ESG] field.
4. Read back the TCDn_CSR[ESG] field.
5. Test the TCDn_CSR[ESG] request status:
   - If ESG = 1, the dynamic scatter/gather attempt was successful.
   - If ESG = 0, read the 32-bit TCDn_DLAST_SGA field.
   - If ESG = 0 and the TCDn_DLAST_SGA did not change, the dynamic scatter/gather attempt was not successful (the channel was already retiring).
   - If ESG = 0 and the TCDn_DLAST_SGA changed, the dynamic scatter/gather attempt was successful (the new TCDn_CSR's ESG value cleared the ESG field to 0).

7.3.4.9 Suspend/resume a DMA channel with active hardware service requests

The DMA allows you to move data from memory or peripheral registers to another location in memory or to peripheral registers without CPU interaction. After the DMA and peripherals are configured and active, it is rare but supported to suspend a peripheral's service request dynamically. In this scenario, there are certain restrictions to disabling a DMA hardware service request. For coherency, you must follow a specific procedure. This section provides guidance on how to coherently suspend and resume a Direct Memory Access (DMA) channel when the DMA is triggered by a slave module such as the Serial Peripheral Interface (SPI), Sigma Delta Analog to Digital Convertor (SDADC), or other module.

7.3.4.9.1 Suspend an active DMA channel

To suspend an active DMA channel:

1. Stop the DMA service request at the peripheral first. Confirm it has been disabled by reading back the appropriate register in the peripheral.
2. Check the DMA's Hardware Request Status (MP_HRS) to ensure there is no service request to the DMA channel being suspended. Then disable the hardware service request by clearing the ERQ field to 0 on the appropriate DMA channel.

For example, assume the SPI is set as a master for transmitting data via a DMA service request when the TXFIFO has an empty slot. The DMA will transfer the next command and data to the TXFIFO upon the request. If you need to suspend the DMA/SPI transfer loop, perform the following steps:

1. Disable the DMA service request at the source by writing 0 to DSPI_RSER[TFFF_RE]. Confirm that DSPI_RSER[TFFF_RE] is 0.

2. Ensure there is no DMA service request from the SPI by verifying that MP_HRS[HRS] is 0 for the appropriate channel. If no service request is present, disable the DMA channel by clearing the channel's ERQ field to 0. If a service request is present, wait until the request has been processed and the HRS field reads 0.

7.3.4.9.2 Resume a DMA channel

To resume a DMA channel:

1. Enable the DMA service request on the appropriate channel by setting its ERQ field to 1.
2. Enable the DMA service request at the peripheral.

7.3.5 Memory map/register definition

The eDMA programming model is partitioned into three parts:

1. The first part defines a number of registers providing overall control functions and is known as the management page.
2. The second part corresponds to the channel (CH) control, status, and configuration.
3. The third part corresponds to the local TCD memory.

7.3.5.1 TCD memory

Each channel requires a 32-byte transfer control descriptor for defining the data movement operation. Each TCDn definition is presented as 11 registers of 16 or 32 bits.

7.3.5.2 TCD initialization

Prior to activating a channel, you must initialize its TCD with the appropriate transfer profile.

7.3.5.3 TCD structure

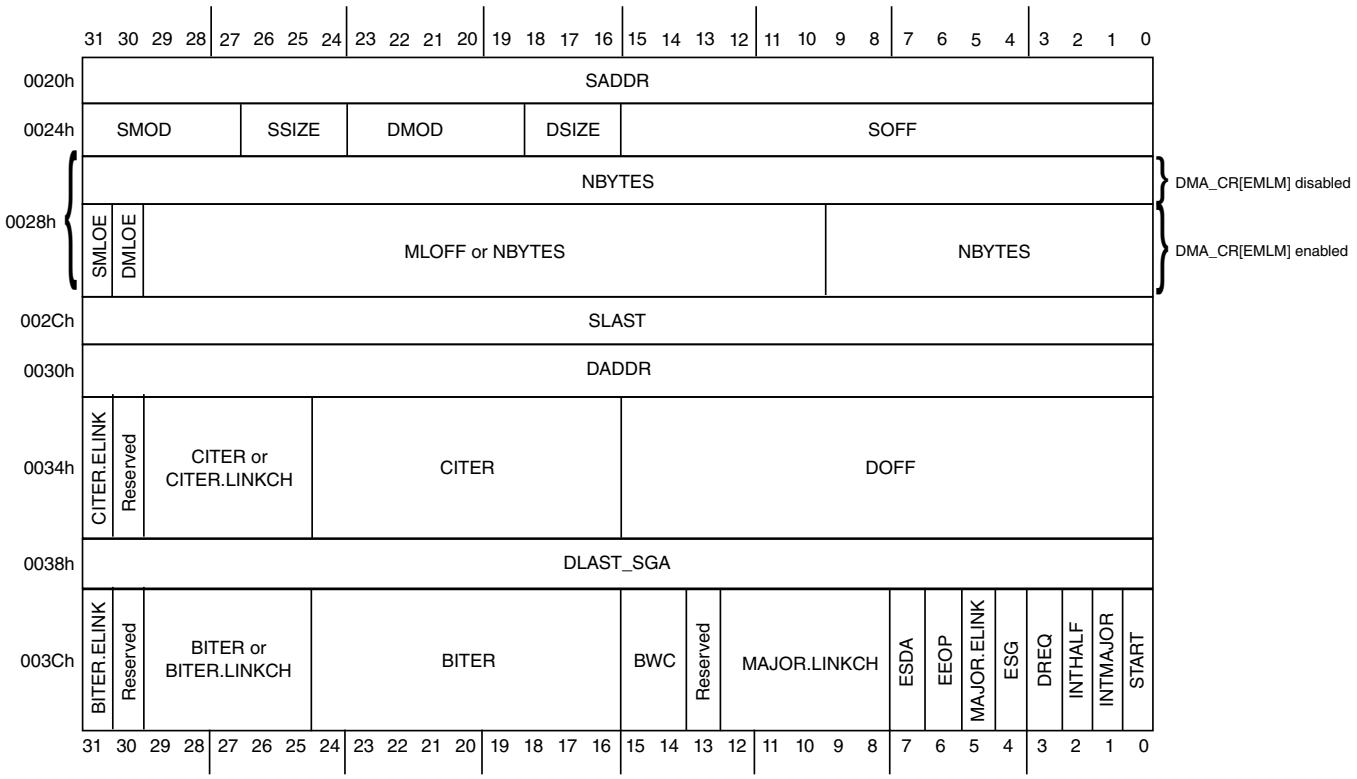


Figure 7-29. TCD structure

7.3.5.4 Accesses to reserved memory and fields

- Reading reserved fields in a register returns the value of zero.
- Writes to reserved fields in a register are ignored.
- Reading or writing a reserved memory location generates a bus error.

7.3.5.5 DMA MP register descriptions

7.3.5.5.1 MP memory map

MP base address: 30E3_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Management Page Control (MP_CSR)	32	RW	0030_0000h
4h	Management Page Error Status (MP_ES)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8h	Management Page Interrupt Request Status (MP_INT)	32	RO	0000_0000h
Ch	Management Page Hardware Request Status (MP_HRS)	32	RO	0000_0000h
100h - 17Ch	Channel Arbitration Group (CH0_GRPRI - CH31_GRPRI)	32	RW	0000_0000h

7.3.5.5.2 Management Page Control (MP_CSR)

7.3.5.5.2.1 Offset

Register	Offset
MP_CSR	0h

7.3.5.5.2.2 Function

The Management Page Control register defines the basic operating configuration of the DMA.

Arbitration uses a two-tier priority system; group and channel priority. The eDMA assigns each channel to a priority group. Group arbitration is fixed-priority and cannot be changed. Channel arbitration uses fixed priority and may be configured to use a selective round-robin scheme for specified channels within each priority group. For fixed-priority arbitration, eDMA selects for execution the highest priority channel requesting service in the highest priority arbitration group.

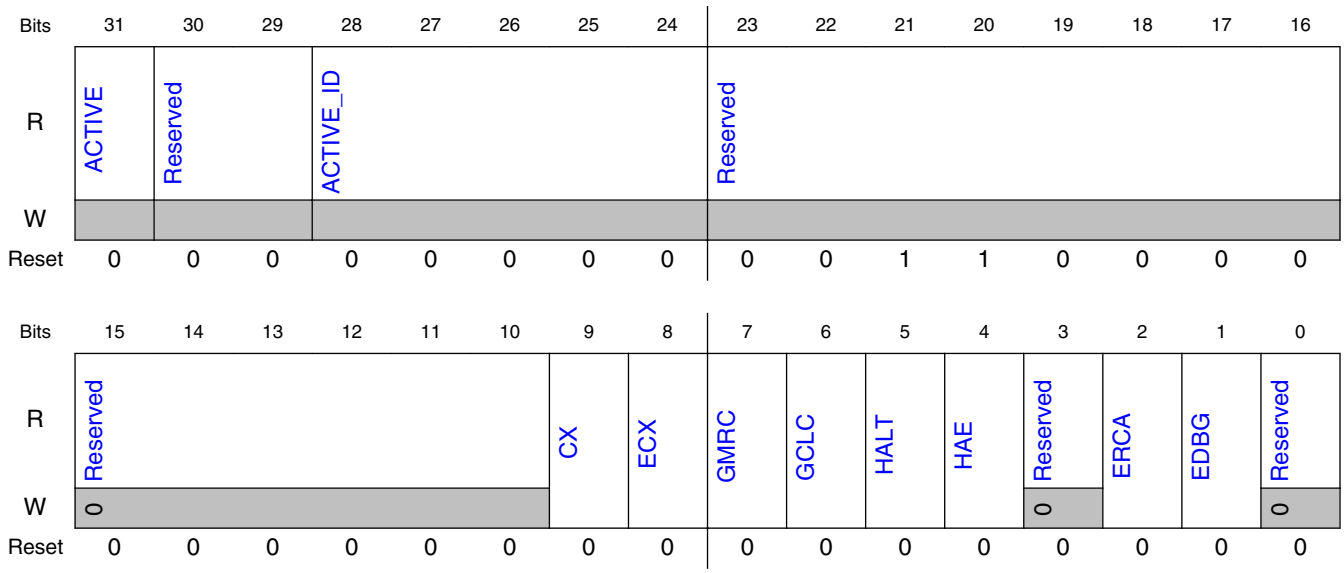
The channel priority registers assign the relative priorities within each arbitration group; see CHn_PRI. All channels with a non-zero CHn_PRI value use fixed-priority arbitration.

When you enable round-robin arbitration, all channels with channel priority set to zero do not have a priority and, of those channels requesting service, are cycled through (from high to low channel number) without regard to priority relative to each other within the same priority group. Any channel with a non-zero CHn_PRI value automatically has a higher priority over the round-robin channels. A channel's priority group is assigned in Channel Arbitration Group (CH0_GRPRI - CH31_GRPRI).

NOTE

For correct operation, changes to the CSR[ERCA, GCLC, GMRC] fields must be performed when the DMA channels are inactive; that is, when the CSR[ACTIVE] field is 0.

7.3.5.5.2.3 Diagram



7.3.5.5.2.4 Fields

Field	Function
31 ACTIVE	DMA Active Status 0b - eDMA is idle 1b - eDMA is executing a channel
30-29 —	Reserved
28-24 ACTIVE_ID	Active Channel ID This field identifies the channel number that is executing when the ACTIVE bit is 1.
23-16 —	Reserved
15-10 —	Reserved
9 CX	Cancel Transfer When set to 1, this field cancels the remaining data transfer, stops the executing channel, and forces the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. CX clears itself to 0 after the cancel has been honored. This cancel retires the channel normally as if the minor loop had been completed. 0b - Normal operation 1b - Cancel the remaining data transfer
8 ECX	Cancel Transfer With Error Cancellation of the remaining data transfer is similar to that of the CX field. Execution of the the channel is stopped and the minor loop is forced to finish. The cancellation takes effect after the last write of the current read/write sequence. The ECX field clears itself to 0 after the cancel is honored. In addition to

Table continues on the next page...

Field	Function
	cancelling the transfer, ECX treats the cancel as an error condition, thus updating Management Page Error Status (MP_ES) and generating an optional error interrupt. 0b - Normal operation 1b - Cancel the remaining data transfer
7 GMRC	Global Master ID Replication Control **NOTE:** If master ID replication is disabled, the privileged protection level (Supervisor mode) for DMA transfers is used. 0b - Master ID replication disabled for all channels 1b - Master ID replication available and controlled by each channel's CHn_SBR[EMI] setting
6 GCLC	Global Channel Linking Control 0b - Channel linking disabled for all channels 1b - Channel linking available and controlled by each channel's link settings
5 HALT	Halt DMA Operations This field stalls the start of any new channels. Executing channels are allowed to complete. Channel execution resumes when this field is cleared to 0. 0b - Normal operation 1b - Stall the start of any new channels
4 HAE	Halt After Error When this field is set to 1, any error causes the HALT field to be set to 1. Then all service requests are ignored until the HALT field is cleared to 0. 0b - Normal operation 1b - Any error causes the HALT field to be set to 1
3 —	Reserved
2 ERCA	Enable Round Robin Channel Arbitration 0b - Round-robin channel arbitration disabled. Fixed priority arbitration used for channel selection within each group 1b - Round-robin channel arbitration enabled. Round-robin arbitration used for channel selection within each group
1 EDBG	Enable Debug When in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete. DMA resumes channel execution when the system exits debug mode or clears the EDBG field to 0. 0b - Debug mode disabled. When in debug mode, the DMA continues to operate 1b - Debug mode is enabled. When in debug mode, the DMA stalls the start of a new channel
0 —	Reserved

7.3.5.5.3 Management Page Error Status (MP_ES)

7.3.5.5.3.1 Offset

Register	Offset
MP_ES	4h

7.3.5.5.3.2 Function

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- An illegal setting in the transfer control descriptor
- An error termination to a bus master read or write cycle
- An uncorrectable error that occurred when the device was accessing the TCD SRAM
- A "cancel transfer with error" request was made via the corresponding cancel transfer field or input signal

Upon any error condition, the software must initialize the TCD of the channel that contains the error, as it is in an incomplete state after an error. See Fault reporting and handling for more details.

7.3.5.5.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VLD	Reserved		ERRCHN					0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							ECX	SAE	SOE	DAE	DOE	NCE	SGE	SBE	DBE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.3.5.5.3.4 Fields

Field	Function
31 VLD	Valid Logical OR of all ERR status fields. 0b - No ERR fields are set to 1 1b - At least one ERR field is set to 1, indicating a valid error exists that software has not cleared
30-29 —	Reserved
28-24 ERRCHN	Error Channel Number or Canceled Channel Number The channel number of the last recorded error or last recorded error-canceled transfer.
23-9	Reserved

Table continues on the next page...

Field	Function
—	
8 ECX	Transfer Canceled The ECX operation is a management page function. When employed, the targeted channel's CHn_ES register reports an unspecified error; that is, only the ERR field is set to 1. The management page has full view of the error condition. 0b - No canceled transfers 1b - Last recorded entry was a canceled transfer by the error cancel transfer input
7 SAE	Source Address Error When this field is 1, it indicates that TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SADDR field
6 SOE	Source Offset Error When this field is 1, it indicates that TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error When this field is 1, it indicates that TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DADDR field
4 DOE	Destination Offset Error When this field is 1, it indicates that TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DOFF field
3 NCE	NBYTES/CITER Configuration Error This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0b - No NBYTES/CITER configuration error 1b - The last recorded error was NBYTES equal to zero or a CITER not equal to BITER error. Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. 0b - No scatter/gather configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0b - No source bus error 1b - Last recorded error was a bus error on a source read
0 DBE	Destination Bus Error 0b - No destination bus error 1b - Last recorded error was a bus error on a destination write

7.3.5.5.4 Management Page Interrupt Request Status (MP_INT)

7.3.5.5.4.1 Offset

Register	Offset
MP_INT	8h

7.3.5.5.4.2 Function

This register shows the current state of the interrupt service requests for all eDMA channels.

7.3.5.5.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	INT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.3.5.5.4.4 Fields

Field	Function
31-0 INT	Interrupt Request Status The INT register presents the interrupt request status for each eDMA channel. Depending on the appropriate field setting in the transfer control descriptors, the eDMA engine generates an interrupt on data transfer completion. The eDMA routes channel interrupt requests to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate field in the channel's interrupt request register, CHn_INT, thus negating the interrupt request. 0b - Interrupt request for corresponding channel not present 1b - Interrupt request for corresponding channel present

7.3.5.5.5 Management Page Hardware Request Status (MP_HRS)

7.3.5.5.5.1 Offset

Register	Offset
MP_HRS	Ch

7.3.5.5.5.2 Function

The hardware request status register (HRS) shows the current state of the hardware service request signaling as seen by eDMA's arbitration logic.

7.3.5.5.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HRS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HRS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.3.5.5.5.4 Fields

Field	Function
31-0 HRS	Hardware Request Status The HRS bit for its respective channel remains asserted for the period when a hardware request is present on the channel. After the request is completed and the channel is free, the hardware automatically clears the corresponding HRS bit to 0. 0b - Hardware service request for corresponding channel is not present 1b - Hardware service request for corresponding channel is present

7.3.5.5.6 Channel Arbitration Group (CH0_GRPRI - CH31_GRPRI)

7.3.5.5.6.1 Offset

For n = 0 to 31:

Register	Offset
CHn_GRPRI	100h + (n × 4h)

7.3.5.5.6.2 Function

The contents of this register define the arbitration group associated with each channel. Using a fixed-priority group arbitration scheme, eDMA evaluates the arbitration group priorities by numeric value from highest group number to lowest; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. The range of the group priority values is limited to the values of 0 through 31. Within each arbitration group, the channel priority assignment CH*n*_PRI determines the highest-priority channel.

7.3.5.5.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											GRPRI				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.3.5.5.6.4 Fields

Field	Function
31-5 —	Reserved
4-0 GRPRI	Arbitration Group For Channel n Fixed-priority arbitration group number.

7.3.5.6 DMA TCD register descriptions

7.3.5.6.1 TCD memory map

TCD base address: 1_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h - 1_F000h	Channel Control and Status (CH0_CSR - CH31_CSR)	32	RW	0000_0000h
4h - 1_F004h	Channel Error Status (CH0_ES - CH31_ES)	32	W1C	0000_0000h
8h - 1_F008h	Channel Interrupt Status (CH0_INT - CH31_INT)	32	W1C	0000_0000h
Ch - 1_F00Ch	Channel System Bus (CH0_SBR - CH31_SBR)	32	RW	0000_8002h
10h - 1_F010h	Channel Priority (CH0_PRI - CH31_PRI)	32	RW	0000_0000h
20h - 1_F020h	TCD Source Address (TCD0_SADDR - TCD31_SADDR)	32	RW	Table 7-629
24h - 1_F024h	TCD Signed Source Address Offset (TCD0_SOFF - TCD31_SOFF)	16	RW	Table 7-629
26h - 1_F026h	TCD Transfer Attributes (TCD0_ATTR - TCD31_ATTR)	16	RW	Table 7-629
28h - 1_F028h	TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD31_NBYTES_MLOFFNO)	32	RW	Table 7-629
28h - 1_F028h	TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD31_NBYTES_MLOFFYES)	32	RW	Table 7-629
2Ch - 1_F02Ch	TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD31_SLAST_SDA)	32	RW	Table 7-629
30h - 1_F030h	TCD Destination Address (TCD0_DADDR - TCD31_DADDR)	32	RW	Table 7-629
34h - 1_F034h	TCD Signed Destination Address Offset (TCD0_DOFF - TCD31_DOFF)	16	RW	Table 7-629
36h - 1_F036h	TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD31_CITER_ELINKNO)	16	RW	Table 7-629
36h - 1_F036h	TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD31_CITER_ELINKYES)	16	RW	Table 7-629
38h - 1_F038h	TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD31_DLAST_SGA)	32	RW	Table 7-629
3Ch - 1_F03Ch	TCD Control and Status (TCD0_CSR - TCD31_CSR)	16	RW	Table 7-629
3Eh - 1_F03Eh	TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD31_BITER_ELINKNO)	16	RW	Table 7-629
3Eh - 1_F03Eh	TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD31_BITER_ELINKYES)	16	RW	Table 7-629

7.3.5.6.2 Channel Control and Status (CH0_CSR - CH31_CSR)

7.3.5.6.2.1 Offset

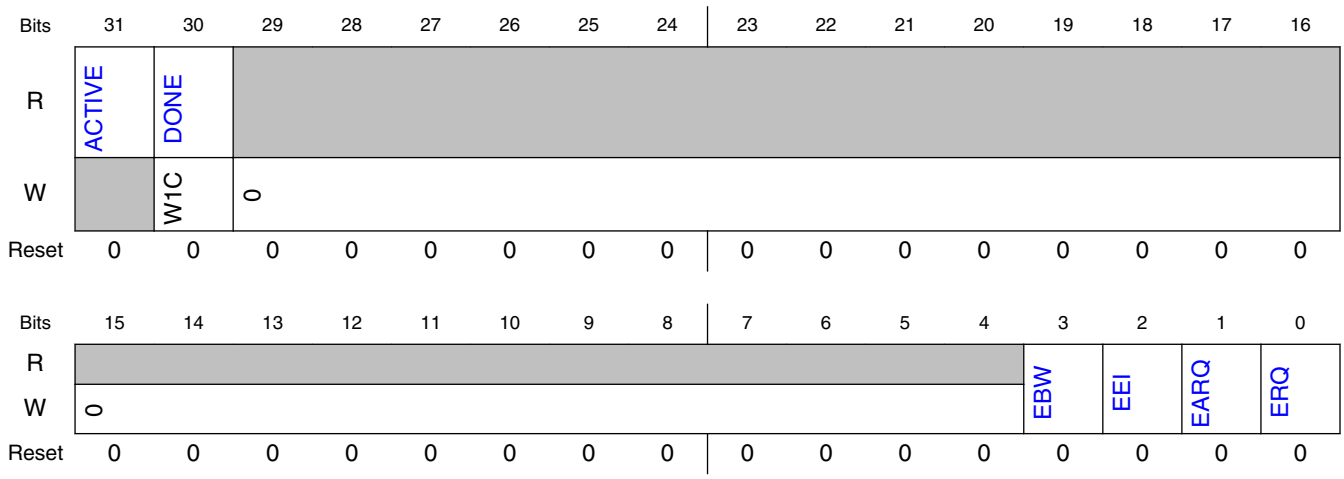
For n = 0 to 31:

Register	Offset
CHn_CSR	0h + (n × 1000h)

7.3.5.6.2.2 Function

This register contains several fields related to hardware and interrupt requests, configuration, and status for the given channel.

7.3.5.6.2.3 Diagram



7.3.5.6.2.4 Fields

Field	Function
31 ACTIVE	Channel Active The ACTIVE field indicates the channel was selected by arbitration and is executing the prescribed transfers. The eDMA sets it to 1 when channel service begins, and clears it to 0 as the minor loop completes or when any error condition is detected. Except for dynamic scatter/gather or dynamic channel linking, you must not modify the transfer control descriptor when a channel is active.
30 DONE	Channel Done The DONE field indicates the eDMA has completed the major loop. The eDMA engine sets this field as the CITER count reaches zero. If enabled, the eDMA generates an interrupt request corresponding to this completed channel. The software clears it, or the hardware clears it when the channel is activated. **NOTE:** This field must be cleared to 0 before writing the MAJORELINK or ESG fields.
29-4 —	Reserved
3 EBW	Enable Buffered Writes When buffered writes are enabled, all writes except for the last write sequence of the minor loop are signaled by the eDMA as bufferable. 0b - Buffered writes on system bus disabled. Buffered writes on system bus disabled 1b - Buffered writes on system bus enabled. Bufferable write signal asserted on all system bus writes except during last write sequence
2 EEI	Enable Error Interrupt

Table continues on the next page...

Field	Function
	The EEI field enables the error interrupt signal for the channel. The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller. 0b - Error signal for corresponding channel does not generate error interrupt 1b - Assertion of error signal for corresponding channel generates error interrupt request
1 EARQ	Enable Asynchronous DMA Request In Stop Mode For Channel The enable asynchronous DMA request field (EARQ) does not affect DMA operations. When set to 1, this field allows the hardware service request enable field (ERQ) to propagate out of the DMA to the power controller. When cleared to 0, this field masks the hardware service request enable field to the power controller. 0b - Disable asynchronous DMA request for the channel 1b - Enable asynchronous DMA request for the channel
0 ERQ	Enable DMA Request Disable a channel's hardware service request at the source before clearing the channel's ERQ field. The DMA hardware request input signal and the enable request field (ERQ) must be asserted before a channel's hardware service request is accepted. The state of the eDMA enable request field does not affect a channel service request made explicitly through software or channel linking. The state of the ERQ field does not affect the channel's START field. 0b - DMA hardware request signal for corresponding channel disabled 1b - DMA hardware request signal for corresponding channel enabled

7.3.5.6.3 Channel Error Status (CH0_ES - CH31_ES)

7.3.5.6.3.1 Offset

For n = 0 to 31:

Register	Offset
CHn_ES	4h + (n × 1000h)

7.3.5.6.3.2 Function

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- An illegal setting in the transfer control descriptor
- An error termination to a bus master read or write cycle

The ERR field signals the presence of an error for the channel. The eDMA engine signals the occurrence of an error condition by setting the appropriate field in this register. The outputs of this register are enabled by the contents of the CHn_CSR[EEI] field, then logically summed across all channels to form an error interrupt request, which may be routed to the interrupt controller.

During the execution of the interrupt service routine associated with any DMA errors, it is software's responsibility to clear the appropriate bit, negating the error-interrupt request. The normal DMA channel completion indicators (setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request) are not affected when eDMA detects an error. The contents of this ERR register field can also be polled because a non-zero value indicates the presence of a channel error, regardless of the state of the EEI mask.

The state of any given channel's error indicators is affected by writes to this register. Writing a 1 to the ERR field clears the channel's error status, and writing a 0 has no effect.

An unspecified error, where only the ERR field is set to 1, indicates that either a transfer was cancelled with an error. The Management Page Error Status register has full view of the error condition.

See Fault reporting and handling for more details.

7.3.5.6.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERR	Reserved														
W	W1C	0														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								SAE	SOE	DAE	DOE	NCE	SGE	SBE	DBE
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.3.5.6.3.4 Fields

Field	Function
31 ERR	Error In Channel 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
30-8 —	Reserved
7 SAE	Source Address Error TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SADDR field

Table continues on the next page...

Field	Function
6 SOE	Source Offset Error TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DADDR field
4 DOE	Destination Offset Error TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DOFF field
3 NCE	NBYTES/CITER Configuration Error This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0b - No NBYTES/CITER configuration error 1b - Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. 0b - No scatter/gather configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0b - No source bus error 1b - Last recorded error was bus error on source read
0 DBE	Destination Bus Error 0b - No destination bus error 1b - Last recorded error was bus error on destination write

7.3.5.6.4 Channel Interrupt Status (CH0_INT - CH31_INT)

7.3.5.6.4.1 Offset

For n = 0 to 31:

Register	Offset
CHn_INT	8h + (n × 1000h)

7.3.5.6.4.2 Function

The INT field signals the presence of an interrupt request for the channel. Depending on the appropriate bit setting in the transfe control descriptors, the eDMA engine generates an interrupt on data transfer completion.

The outputs of this register are directly routed to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit, negating the interrupt request. On writes to INT, a 1 clears the channel's interrupt request. A zero has no effect on the channel's current interrupt status.

7.3.5.6.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															INT
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.3.5.6.4.4 Fields

Field	Function
31-1 —	Reserved
0 INT	Interrupt Request 0b - Interrupt request for corresponding channel cleared 1b - Interrupt request for corresponding channel active

7.3.5.6.5 Channel System Bus (CH0_SBR - CH31_SBR)

7.3.5.6.5.1 Offset

For n = 0 to 31:

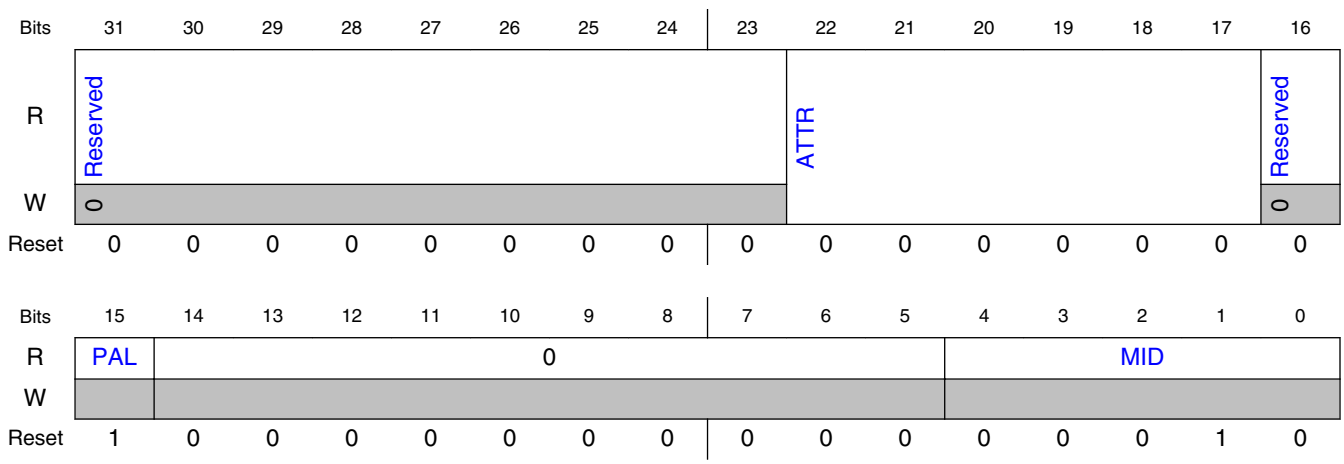
Register	Offset
CHn_SBR	Ch + (n × 1000h)

7.3.5.6.5.2 Function

The Channel System Bus register places identification and attribute information on the system bus interface for the eDMA.

The ATTR register outputs the register values onto the system bus interface for further decoding by the security system.

7.3.5.6.5.3 Diagram



7.3.5.6.5.4 Fields

Field	Function
31-23 —	Reserved
22-17 ATTR	Attribute Output DMA's system bus attribute output value.
16 —	Reserved
15 PAL	Privileged Access Level This field controls DMA's protection level on the system bus when the channel is active. **NOTE:** When you enable master ID replication, the value captured in this register is the privilege level of the core or other master writing the channel's transfer control descriptor, which is the lower byte of TCD*n*_CSR. 0b - User protection level for DMA transfers 1b - Privileged protection level for DMA transfers
14-5 —	Reserved
4-0	Master ID

Field	Function
MID	This field controls the DMA's master ID on the system bus when the channel is active. **NOTE:** The ID captured in this register reflects the master ID of the core or other master writing the channel's control attributes, which are in the lower byte of TCD*n*_CSR.

7.3.5.6.6 Channel Priority (CH0_PRI - CH31_PRI)

7.3.5.6.6.1 Offset

For n = 0 to 31:

Register	Offset
CHn_PRI	10h + (n × 1000h)

7.3.5.6.6.2 Function

The contents of these registers define unique priorities associated with each channel within the same channel group. Channel grouping is programmed via Channel Arbitration Group (CH0_GRPRI - CH31_GRPRI).

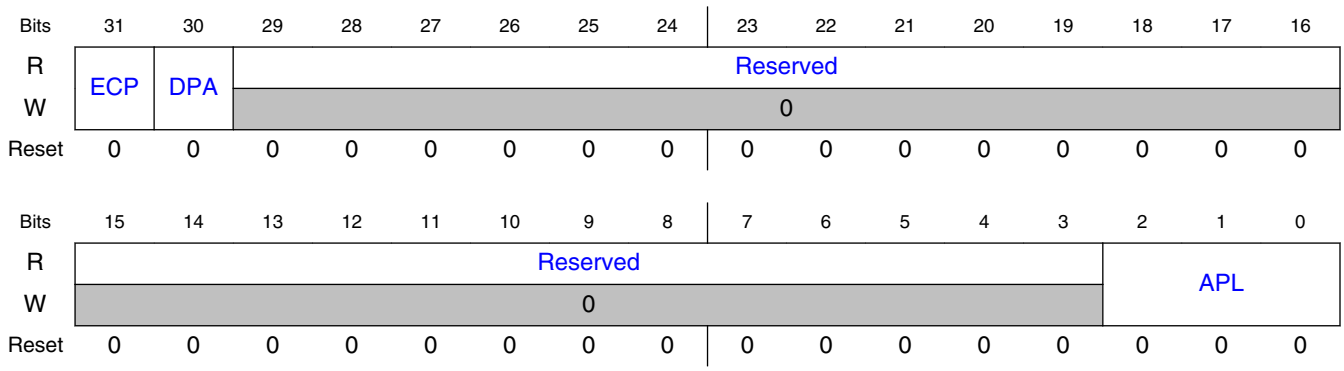
The channel priorities within a group are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. Software must program the channel priorities with unique values; otherwise, channel numbers with the same, non-zero value, will be selected based on channel number with the higher channel number having higher priority.

If more than one channel in a group has an arbitration priority level value of zero, then the arbitration mode field MP_CSR[ERCA] is used to determine the arbitration scheme for all channels with APL=0 within a group.

When you enable round-robin channel arbitration (MP_CSR[ERCA] = 1), all channels with APL=0 within a group will use a round-robin arbitration scheme, which rotates among these channels requesting service without regard to priority. Round-robin provides a fairness mechanism within an arbitration group.

When you enable fixed-priority channel arbitration (MP_CSR[ERCA] = 0), eDMA selects channels with APL=0 based on channel number, with the higher channel number having higher priority.

7.3.5.6.6.3 Diagram



7.3.5.6.6.4 Fields

Field	Function
31 ECP	Enable Channel Preemption 0b - Channel cannot be suspended by a higher-priority channel's service request 1b - Channel can be temporarily suspended by a higher-priority channel's service request
30 DPA	Disable Preempt Ability 0b - Channel can suspend a lower-priority channel 1b - Channel cannot suspend any other channel, regardless of channel priority
29-3 —	Reserved
2-0 APL	Arbitration Priority Level Channel priority level for arbitration within the assigned arbitration group.

7.3.5.6.7 TCD Source Address (TCD0_SADDR - TCD31_SADDR)

7.3.5.6.7.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_SADDR	20h + (n × 1000h)

7.3.5.6.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

7.3.5.6.7.3 Fields

Field	Function
31-0 SADDR	Source Address Memory address pointing to the source data.

7.3.5.6.8 TCD Signed Source Address Offset (TCD0_SOFF - TCD31_SOFF)

7.3.5.6.8.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_SOFF	24h + (n × 1000h)

7.3.5.6.8.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SOFF															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

7.3.5.6.8.3 Fields

Field	Function
15-0	Source Address Signed Offset

Field	Function
SOFF	Sign-extended offset applied to the current source address to form the next-state value as each source read is completed.

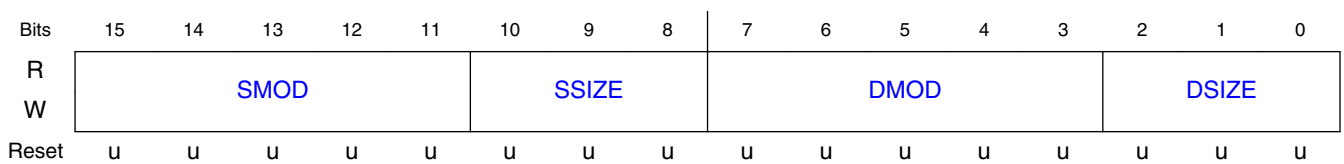
7.3.5.6.9 TCD Transfer Attributes (TCD0_ATTR - TCD31_ATTR)

7.3.5.6.9.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_ATTR	26h + (n × 1000h)

7.3.5.6.9.2 Diagram



7.3.5.6.9.3 Fields

Field	Function
15-11 SMOD	Source Address Modulo This field defines a specific address range, which is the value after the SADDR + SOFF calculation is performed on the original register value. Setting this field makes it easy to implement a circular data queue. For data queues requiring power-of-2-sized bytes, the queue must start at a 0-modulo-size address and the SMOD field must be set to the appropriate value for the queue, freezing the required number of upper address bits. The value programmed into this field specifies the number of lower address bits that are allowed to change. For a circular queue application, you typically set TCDn_SOFF[SOFF] to the transfer size to implement post-increment addressing, with the SMOD function constraining the addresses to a 0-modulo-size range. 00000b - Source address modulo feature disabled 00001b - Source address modulo feature enabled for any non-zero value [1-31]
10-8 SSIZE	Source Data Transfer Size 000b - 8-bit 001b - 16-bit 010b - 32-bit 011b - 64-bit

Table continues on the next page...

Field	Function
	100b - 16-byte 101b - 32-byte 110b - 64-byte 111b - Reserved
7-3 DMOD	Destination Address Modulo See the SMOD definition.
2-0 DSIZE	Destination Data Transfer Size See the SSIZE definition.

7.3.5.6.10 TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD31_NBYTES_MLOFFNO)

7.3.5.6.10.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_NBYTES_MLOFFNO	28h + (n × 1000h)

7.3.5.6.10.2 Function

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offsets are address offset values added to the final source address (TCDn_SADDR), or destination address (TCDn_DADDR), upon minor loop completion. Minor loop completion is when the channel has finished the service request and has transferred NBYTES. When minor loop offsets are enabled, the minor loop offset value (TCDn_NBYTES_MLOFFYES[MLOFF]) is added to the final source address (TCDn_SADDR), to the final destination address (TCDn_DADDR), or to both, prior to the addresses being written back to the TCD. If the major loop is complete, the minor loop offset is ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

When minor loop mapping is enabled (SMLOE or DMLOE is 1), TCDn_NBYTES_MLOFFNO /TCDn_NBYTES_MLOFFYES is redefined. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is used to specify multiple fields:

- A source enable bit (SMLOE) to specify the minor loop offset must be applied to the source address (TCDn_SADDR) upon minor loop completion
- A destination enable bit (DMLOE) to specify the minor loop offset must be applied to the destination address (TCDn_DADDR) upon minor loop completion
- The sign extended minor loop offset value (MLOFF)

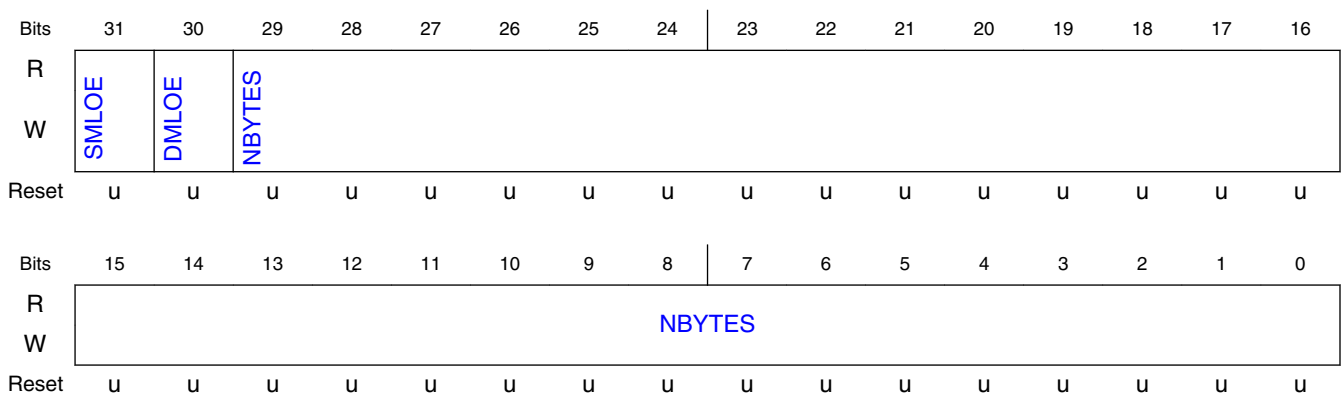
The same offset value (MLOFF) is used for both source and destination minor loop offsets. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is defined as follows:

- If SMLOE = 0 and DMLOE = 0, then see the TCDn_NBYTES_MLOFFNO register description.
- If either SMLOE or DMLOE is 1, then see the TCDn_NBYTES_MLOFFYES register description.

7.3.5.6.10.3 Diagram



7.3.5.6.10.4 Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0b - Minor loop offset not applied to SADDR 1b - Minor loop offset applied to SADDR
30	Destination Minor Loop Offset Enable

Table continues on the next page...

Field	Function
DMLOE	Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0b - Minor loop offset not applied to DADDR 1b - Minor loop offset applied to DADDR
29-0 NBYTES	Number of Bytes To Transfer Per Service Request Number of bytes to be transferred for each service request of the channel. When a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the byte transfer count has been reached. This process is normally an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the byte count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major loop iteration count (CITER) is decremented by one and written back to the TCD memory. If the major iteration count is complete, additional processing is performed.

7.3.5.6.11 TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD31_NBYTES_MLOFFYES)

7.3.5.6.11.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_NBYTES_MLOFFYES	28h + (n × 1000h)

7.3.5.6.11.2 Function

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offset is an address offset value added to the final source address (TCDn_SADDR) or destination address (TCDn_DADDR) upon minor loop completion. Minor loop completion occurs when the channel has finished the service request and has transferred NBYTES. Minor loop offsets are enabled by setting either the source enable bit (SMLOE) or the destination enable bit (DMLOE).

The source enable bit (SMLOE) specifies the minor loop offset value (MLOFF) that is to be applied to the source address (TCDn_SADDR) upon minor loop completion. The destination enable bit (DMLOE) specifies the minor loop offset (MLOFF) that is to be applied to the destination address (TCDn_DADDR) upon minor loop completion.

If the major loop is complete, the minor loop offsets are ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

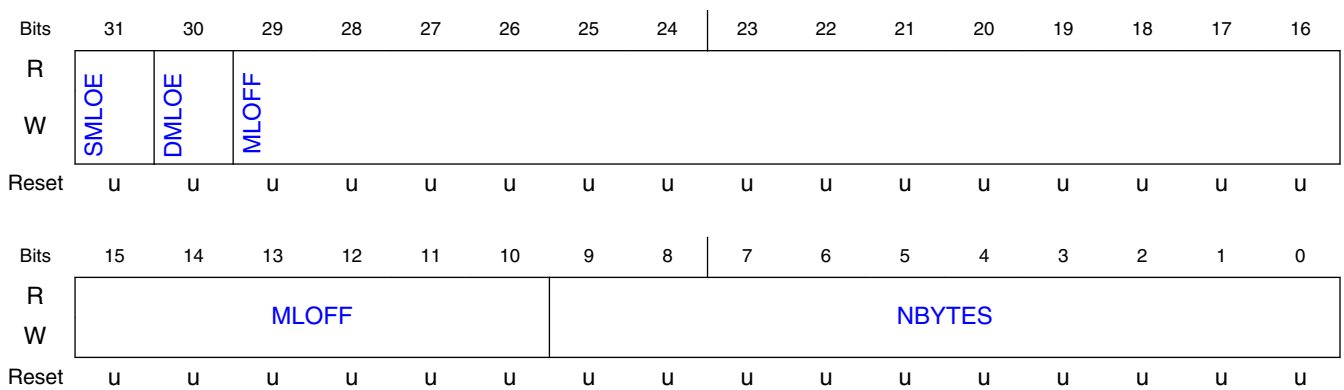
When you enable the minor loop offset overlay (either SMLOE or DMLOE is 1), eDMA redefines TCDn_NBYTES_MLOFFNO /TCDn_NBYTES_MLOFFYES. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES specifies the sign-extended minor loop offset value (MLOFF). The same offset value (MLOFF) applies to both source and destination minor loop offsets. When the minor loop offset is enabled, you must align it to the transfer size of the source or destination it is associated with. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFNO) defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFYES is defined as follows:

- If either minor loop offset is enabled (SMLOE or DMLOE = 1), then see the TCDn_NBYTES_MLOFFYES register description.
- If SMLOE and DMLOE are both 0, then see the TCDn_NBYTES_MLOFFNO register description.

7.3.5.6.11.3 Diagram



7.3.5.6.11.4 Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0b - Minor loop offset not applied to SADDR 1b - Minor loop offset applied to SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion.

Table continues on the next page...

Field	Function
	0b - Minor loop offset not applied to DADDR 1b - Minor loop offset applied to DADDR
29-10 MLOFF	Minor Loop Offset If SMLOE or DMLOE is 1, this field represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes.
9-0 NBYTES	Number of Bytes To Transfer Per Service Request The number of bytes to be transferred in each service request of the channel. As a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the minor byte transfer count has been reached. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major iteration count is decremented and restored to the TCD memory. If the major iteration count is complete, additional processing is performed.

7.3.5.6.12 TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD31_SLAST_SDA)

7.3.5.6.12.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_SLAST_SDA	2Ch + (n × 1000h)

7.3.5.6.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SLAST_SDA															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SLAST_SDA															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

7.3.5.6.12.3 Fields

Field	Function
31-0 SLAST_SDA	Last Source Address Adjustment / Store DADDR Address Source last address adjustment or the system memory address for destination address (DADDR) storage. If (TCDn_CSR[ESDA] = 0), then: • Adjustment value is added to the source address at the completion of the major iteration count. This value can be used to restore the source address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final source address adjustment. Otherwise: • This address points to the 32-bit-aligned memory location where the destination address (DADDR) is to be stored in system memory. By saving the final destination address in system memory via the ESDA feature, you are able to compute the size of a variable destination data buffer by simply subtracting the beginning DADDR from the final, saved DADDR. This feature is used together with the scatter/gather operation to prevent the loss of the final DADDR, which is overwritten during the scatter/gather operation. The "Store Destination Address" (SDA) value must be a 32-bit-aligned location because the eDMA forces the lower two address bits of the SLAST_SDA field to zero when ESDA is enabled. The module performs this write operation when the major loop is done; that is, when the major iteration count (CITER) decrements to zero.

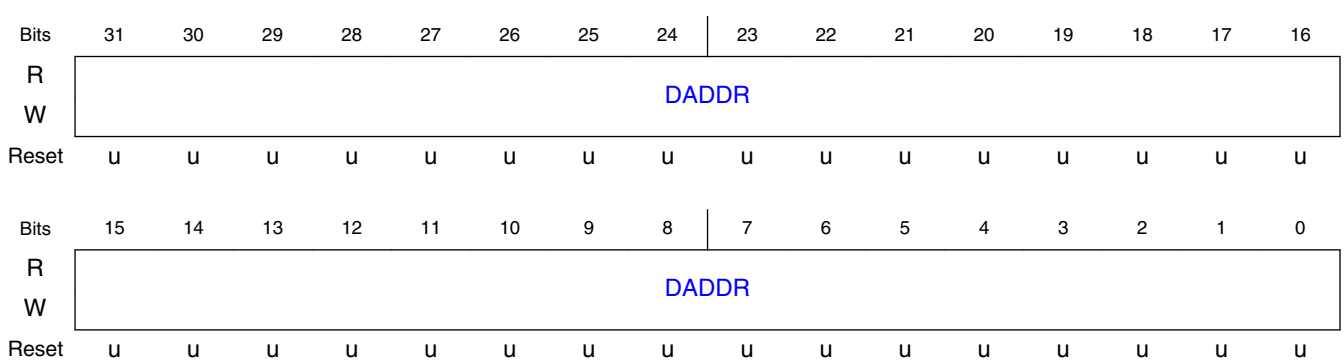
7.3.5.6.13 TCD Destination Address (TCD0_DADDR - TCD31_DADDR)

7.3.5.6.13.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_DADDR	30h + (n × 1000h)

7.3.5.6.13.2 Diagram



7.3.5.6.13.3 Fields

Field	Function
31-0 DADDR	Destination Address Memory address pointing to the destination data.

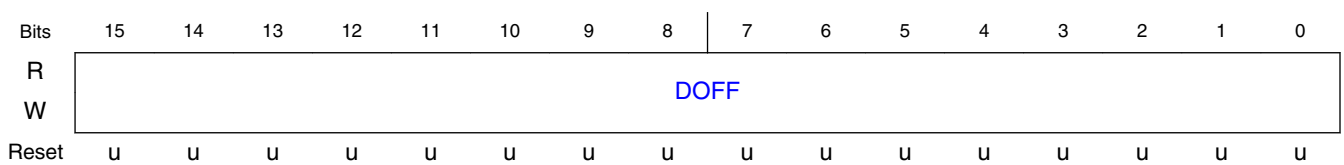
7.3.5.6.14 TCD Signed Destination Address Offset (TCD0_DOFF - TCD31_DOFF)

7.3.5.6.14.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_DOFF	34h + (n × 1000h)

7.3.5.6.14.2 Diagram



7.3.5.6.14.3 Fields

Field	Function
15-0 DOFF	Destination Address Signed Offset Sign-extended offset that is applied to the current destination address to form the next-state value as each destination write is completed.

7.3.5.6.15 TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD31_CITER_ELINKNO)

7.3.5.6.15.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_CITER_ELINKNO	36h + (n × 1000h)

7.3.5.6.15.2 Function

If TCDn_CITER[ELINK] is 0, the TCDn_CITER register is defined as follows.

7.3.5.6.15.3 Diagram

7.3.5.6.15.4 Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel to 1. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking. **NOTE:** This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE:** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE:** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

7.3.5.6.16 TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD31_CITER_ELINKYES)

7.3.5.6.16.1 Offset

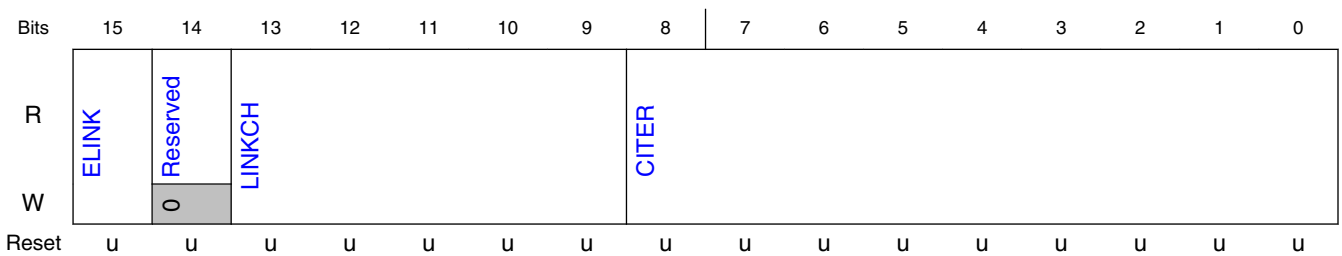
For n = 0 to 31:

Register	Offset
TCDn_CITER_ELINKYES	36h + (n × 1000h)

7.3.5.6.16.2 Function

If TCDn_CITER[ELINK] is 1, the TCDn_CITER register is defined as follows.

7.3.5.6.16.3 Diagram



7.3.5.6.16.4 Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. When enabled, an internal mechanism sets the TCDn_CSR[START] field of the specified channel (LINKCH) upon minor loop completion. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking. **NOTE:** This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14 —	Reserved
13-9 LINKCH	Minor Loop Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted the eDMA engine initiates a channel service request to the channel defined by this field by writing that channel's TCDn_CSR[START] field to 1.
8-0 CITER	Current Major Iteration Count

Field	Function
	This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to the TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE:** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE:** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

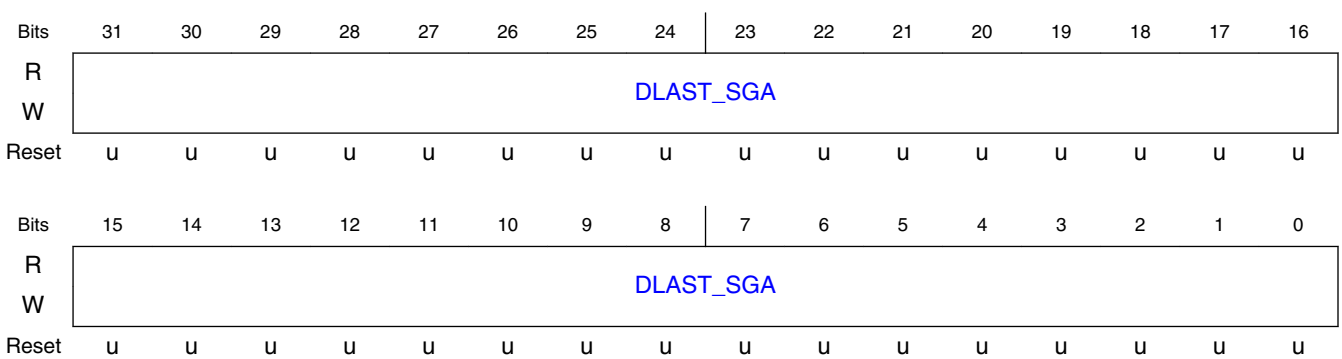
7.3.5.6.17 TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD31_DLAST_SGA)

7.3.5.6.17.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_DLAST_SGA	38h + (n × 1000h)

7.3.5.6.17.2 Diagram



7.3.5.6.17.3 Fields

Field	Function
31-0 DLAST_SGA	Last Destination Address Adjustment / Scatter Gather Address Adjustment of the last destination address or the memory address for the next transfer control descriptor to be loaded into this channel (scatter/gather). If (TCDn_CSR[ESG] = 0) then:

Field	Function
	• Adjustment value is added to the destination address at the completion of the major iteration count. This value can apply to restore the destination address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final destination address adjustment. Otherwise: • This address points to the beginning of a 0-modulo 32-byte region containing the next transfer control descriptor to be loaded into this channel. This channel reload is performed as the major iteration count completes. The scatter/gather address must be 0-modulo 32-byte, or else a configuration error is reported.

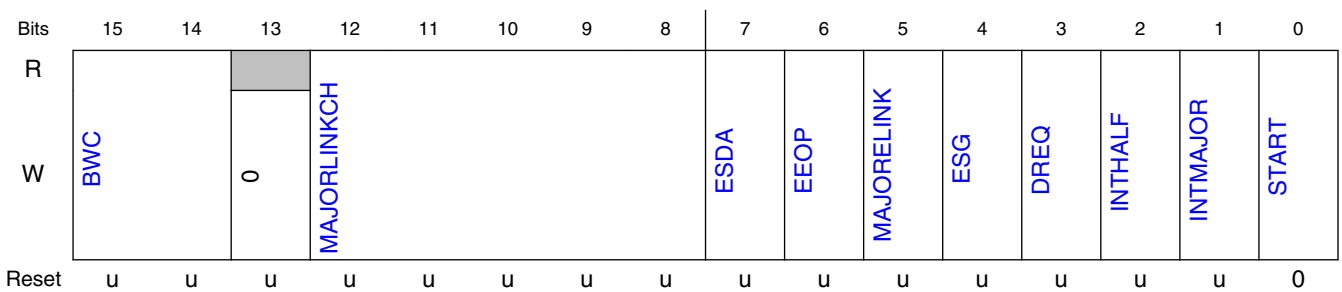
7.3.5.6.18 TCD Control and Status (TCD0_CSR - TCD31_CSR)

7.3.5.6.18.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_CSR	3Ch + (n × 1000h)

7.3.5.6.18.2 Diagram



7.3.5.6.18.3 Fields

Field	Function
15-14 BWC	Bandwidth Control Throttles the amount of bus bandwidth consumed by the eDMA. Generally, as the eDMA processes the minor loop, it continuously generates read/write sequences until the minor count is exhausted. This field forces eDMA to stall after the completion of each read/write access, to control the bus request bandwidth seen by the system bus interconnect. **NOTE:** If the source and destination sizes are equal, this field is ignored between the first and second transfers and after the last write of each minor loop. This behavior is a side effect of reducing start-up latency.

Table continues on the next page...

Field	Function
	00b - No eDMA engine stalls 01b - Reserved 10b - eDMA engine stalls for 4 cycles after each R/W 11b - eDMA engine stalls for 8 cycles after each R/W
13 —	Reserved
12-8 MAJORLINKCH	Major Loop Link Channel Number If (MAJORELINK = 0) then: • No channel-to-channel linking, or chaining, is performed after the major loop counter is exhausted. Otherwise: • After the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field to 1.
7 ESDA	Enable Store Destination Address As the channel completes the major loop by either the current iteration counter (CITER) decrementing to 0, or by receiving an enabled end-of-packet signal, this field enables writing the destination address (DADDR) to the address stored in the SLAST_SDA field. The value written to system memory is the last DADDR value prior to the DLAST_SGA offset being applied, or overwritten by an enabled scatter/gather operation. When the SDA bit is 1, SLAST_SDA contains the write pointer instead of the final source address offset. Because this is a pointer and not a final offset, a last source address offset of zero is applied to SADDR instead of the SLAST_SGA value. 0b - Ability to store destination address to system memory disabled 1b - Ability to store destination address to system memory enabled
6 EEOP	Enable End-Of-Packet Processing When enabled by the EEOP field, an end-of-packet hardware input signal directs eDMA to discontinue executing the active channel, and to treat the shutdown as the major-loop-completed event. If the EEOP field is 1, the end-of-packet signal from supported peripherals is accepted. If the EEOP field is 0, the end-of-packet input is ignored. With an end-of-packet retirement, the current TCD destination address (or ESDA-saved destination address), minus the software-saved initial address (DADDR), reflects the total amount of data transferred. 0b - End-of-packet operation disabled 1b - End-of-packet hardware input signal enabled
5 MAJORELINK	Enable Link When Major Loop Complete As the channel completes the major loop, this flag enables linking to another channel defined by MAJORLINKCH. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. **NOTE:** To support the dynamic linking coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
4 ESG	Enable Scatter/Gather Processing As the channel completes the major loop, this flag enables scatter/gather processing in the current channel. If enabled, the eDMA engine uses TCDn_DLAST_SGA as a memory pointer to a 0-modulo 32-bit address containing a 32-byte data structure, which is loaded as the transfer control descriptor into local memory. **NOTE:** To support the dynamic scatter/gather coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0b - Current channel's TCD is normal format 1b - Current channel's TCD specifies scatter/gather format.

Table continues on the next page...

Field	Function
3 DREQ	Disable Request If this flag is 1, the eDMA hardware automatically clears the corresponding ERQ bit when the current major iteration count reaches 0. 0b - No operation. Channel's ERQ field not affected 1b - Clear the ERQ field to 0 upon major loop completion, thus disabling hardware service requests. Channel's ERQ field cleared to 0 when major loop complete
2 INTHALF	Enable Interrupt If Major Counter Half-complete If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count reaches the halfway point. Specifically, the comparison performed by the eDMA engine is (CITER = (BITER/2)). This halfway point interrupt request is provided to support double-buffered, also known as ping-pong, schemes, or other types of data movement where the processor needs an early indication of the transfer's progress. **NOTE:** If BITER = 1, do not use INTHALF; use INTMAJOR instead. 0b - Halfway point interrupt disabled 1b - Halfway point interrupt enabled
1 INTMAJOR	Enable Interrupt If Major count complete If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count (CITER) reaches 0. 0b - End-of-major loop interrupt disabled 1b - End-of-major loop interrupt enabled
0 START	Channel Start If this flag is 1, the channel is requesting service. The eDMA hardware automatically clears this flag to 0 after the channel begins execution. 0b - Channel not explicitly started 1b - Channel explicitly started via a software-initiated service request

7.3.5.6.19 TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD31_BITER_ELINKNO)

7.3.5.6.19.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_BITER_ELINKNO	3Eh + (n × 1000h)

7.3.5.6.19.2 Function

If the TCDn_BITER[ELINK] field is 0, the TCDn_BITER register is defined as follows.

7.3.5.6.19.3 Diagram

7.3.5.6.19.4 Fields

Field	Function
15 ELINK	Enables Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

7.3.5.6.20 TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD31_BITER_ELINKYES)

7.3.5.6.20.1 Offset

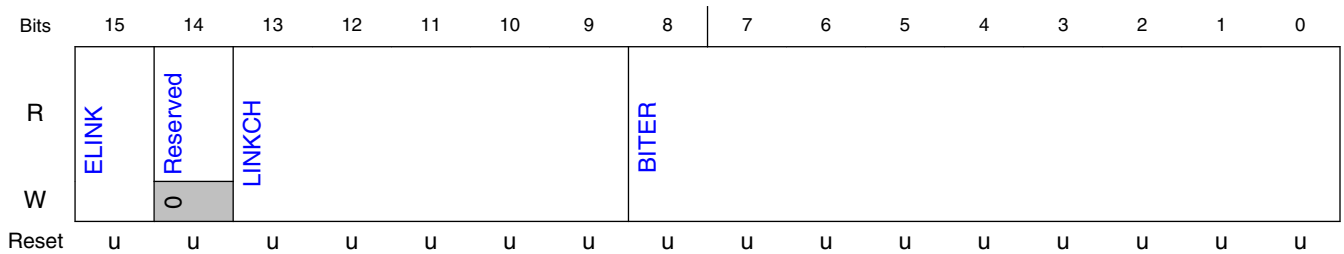
For n = 0 to 31:

Register	Offset
TCDn_BITER_ELINKYES	3Eh + (n × 1000h)

7.3.5.6.20.2 Function

If the TCDn_BITER[ELINK] field is set, the TCDn_BITER register is defined as follows.

7.3.5.6.20.3 Diagram



7.3.5.6.20.4 Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking disables, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14 —	Reserved
13-9 LINKCH	Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field.
8-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

7.4 Interrupt Request Steering (IRQ_STEER)

7.4.1 Overview

The Interrupt Request Steering (IRQ_STEER) module redirects/steers the incoming interrupts to output interrupts of a selected/designated channel as specified by a set of configuration registers.

7.4.1.1 Block diagram

A high-level block diagram of the IRQ_STEER module is provided below.

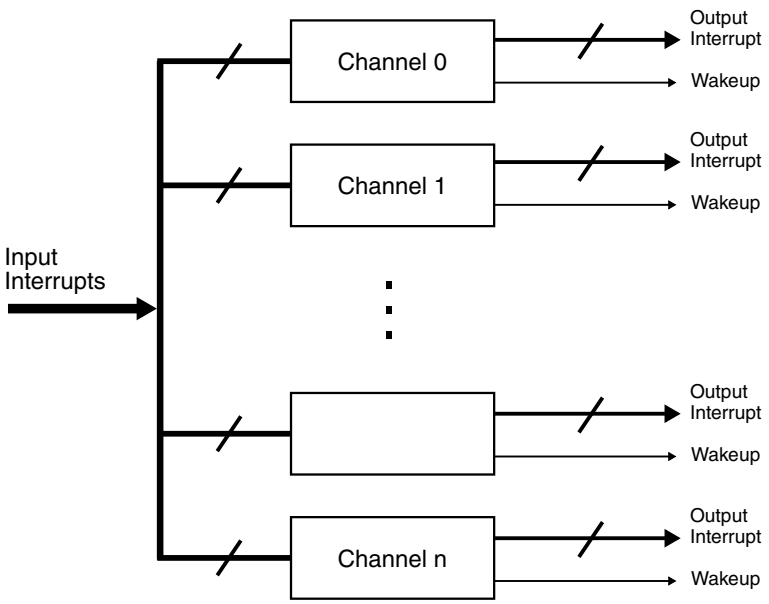


Figure 7-30. IRQ_STEER block diagram

The following figure shows the high-level operation of one channel of the IRQ_STEER module.

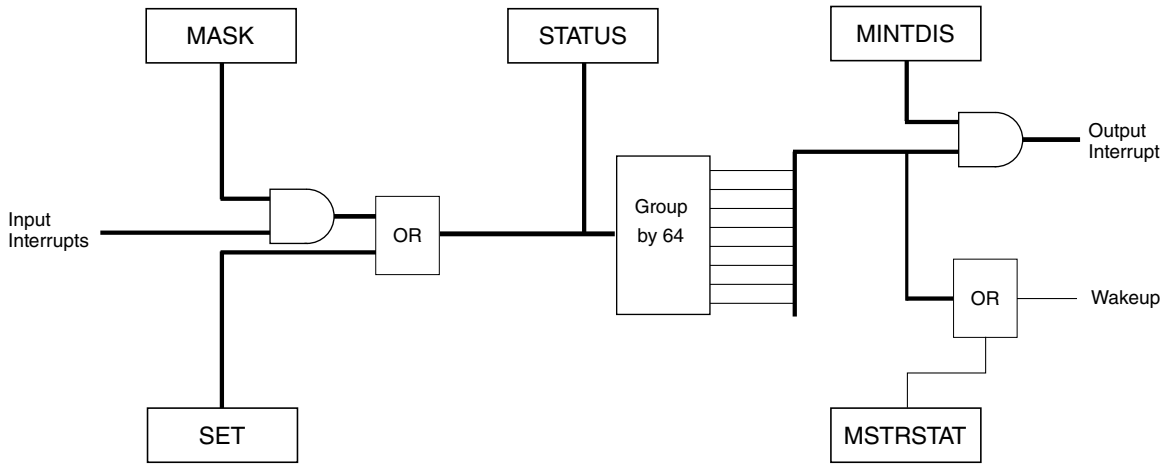


Figure 7-31. IRQ_STEER channel block diagram

7.4.1.2 Features

The IRQ_STEER module supports:

- 3 IRQ channels
- 160 interrupts per channel

7.4.2 Functional description

The IRQ_STEER module allows the user to steer 160 interrupts into one of 3 channels.

Each channel in the IRQ_STEER module supports up to 160 interrupts, and has its own set of registers consisting of:

- 5 CHn_MASK registers
- 5 CHn_SET registers
- 5 CHn_STATUS registers
- One CHn_MINTDIS register
- One CHn_MSTRSTAT register

Since this module supports 3 channels, there are 3 sets of these registers. Each set of registers will have its own base offset, which must be added to the relative offset shown in the IRQ_STEER memory map.

7.4.2.1 Interrupt Mapping

The following table shows the correlation between an interrupt and its register and bit offset for the CHn_MASK, CHn_SET, and CHn_STATUS registers of each channel.

For example:

- For a = 0, the row represents the mapping of an interrupt to the bit offset for the CHn_MASK**0**, CHn_SET**0**, and CHn_STATUS**0** registers.
- For a = 1, the row represents the mapping of an interrupt to the bit offset for the CHn_MASK**1**, CHn_SET**1**, and CHn_STATUS**1** registers.

Table 7-66. Interrupt Mapping (incomplete)

a	Bit offset																															
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	159	158	157	156	155	154	153	152	151	150	149	148	147	146	145	144	143	142	141	140	139	138	137	136	135	134	133	132	131	130	129	128
1	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
2	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
3	63	62	61	60	59	58	57	56	55	54	53	52	52	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

7.4.2.2 Clocks

The following table describes the clock sources for the IRQ_STEER module. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 7-67. IRQ_STEER Clocks

Clock name	Description
ipg_clk	Peripheral clock

7.4.3 External Signals

There are no external signals pinned out for this module.

7.4.4 Memory Map / Register Definition

This section includes the IRQ_STEER module memory map and detailed descriptions of all registers.

7.4.4.1 IRQ_STEER register descriptions

7.4.4.1.1 IRQ_STEER memory map

irq_steer_audio_processor base address: 30A8_0000h

irq_steer_hdmi_tx base address: 32FC_2000h

Offset	Register	Width (In bits)	Access	Reset value
4h - 14h	Channel n Interrupt Mask Register (CHn_MASK0 - CHn_MASK4)	32	RW	0000_0000h
18h - 28h	Channel n Interrupt Set Register (CHn_SET0 - CHn_SET4)	32	RW	0000_0000h
2Ch - 3Ch	Channel n Interrupt Status Register (CHn_STATUS0 - CHn_STATUS4)	32	RO	0000_0000h
40h	Channel n Master Interrupt Disable Register (CHn_MINTDIS)	32	RW	0000_0000h
44h	Channel n Master Status Register (CHn_MSTRSTAT)	32	RO	0000_0000h

7.4.4.1.2 Channel n Interrupt Mask Register (CHn_MASK0 - CHn_MASK4)

7.4.4.1.2.1 Offset

Register	Offset
CHn_MASK0	4h
CHn_MASK1	8h
CHn_MASK2	Ch
CHn_MASK3	10h
CHn_MASK4	14h

7.4.4.1.2.2 Function

The MASK registers are used to mask any of the 160 individual interrupts.

7.4.4.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MASKFLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASKFLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.4.4.1.2.4 Fields

Field	Function
31-0 MASKFLD	Mask bits See Table 7-66 for the correlation between an interrupt and its register and bit offset. 00000000000000000000000000000000b - Mask interrupt 00000000000000000000000000000001b - Do not mask interrupt

7.4.4.1.3 Channel n Interrupt Set Register (CHn_SET0 - CHn_SET4)

7.4.4.1.3.1 Offset

Register	Offset
CHn_SET0	18h
CHn_SET1	1Ch
CHn_SET2	20h
CHn_SET3	24h
CHn_SET4	28h

7.4.4.1.3.2 Function

This register is used to force an interrupt.

7.4.4.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FORCEFLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FORCEFLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.4.4.1.3.4 Fields

Field	Function
31-0 FORCEFLD	Force interrupt. See Table 7-66 for the correlation between an interrupt and its register and bit offset. 00000000000000000000000000000000b - Normal operation 00000000000000000000000000000001b - Force interrupt

7.4.4.1.4 Channel n Interrupt Status Register (CHn_STATUS0 - CHn_STATUS4)

7.4.4.1.4.1 Offset

Register	Offset
CHn_STATUS0	2Ch
CHn_STATUS1	30h
CHn_STATUS2	34h
CHn_STATUS3	38h
CHn_STATUS4	3Ch

7.4.4.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.4.4.1.4.3 Fields

Field	Function
31-0 STATUS	Status of an interrupt See Table 7-66 for the correlation between an interrupt and its register and bit offset. 00000000000000000000000000000000b - Interrupt is not set. 00000000000000000000000000000001b - Interrupt is set.

7.4.4.1.5 Channel n Master Interrupt Disable Register (CHn_MINTDIS)

7.4.4.1.5.1 Offset

Register	Offset
CHn_MINTDIS	40h

7.4.4.1.5.2 Function

Table 7-68. Interrupt disable mapping

STATUS bit	Interrupts disabled
2	191 - 128
1	127 - 64
0	63 - 0

7.4.4.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													DISABLE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.4.4.1.5.4 Fields

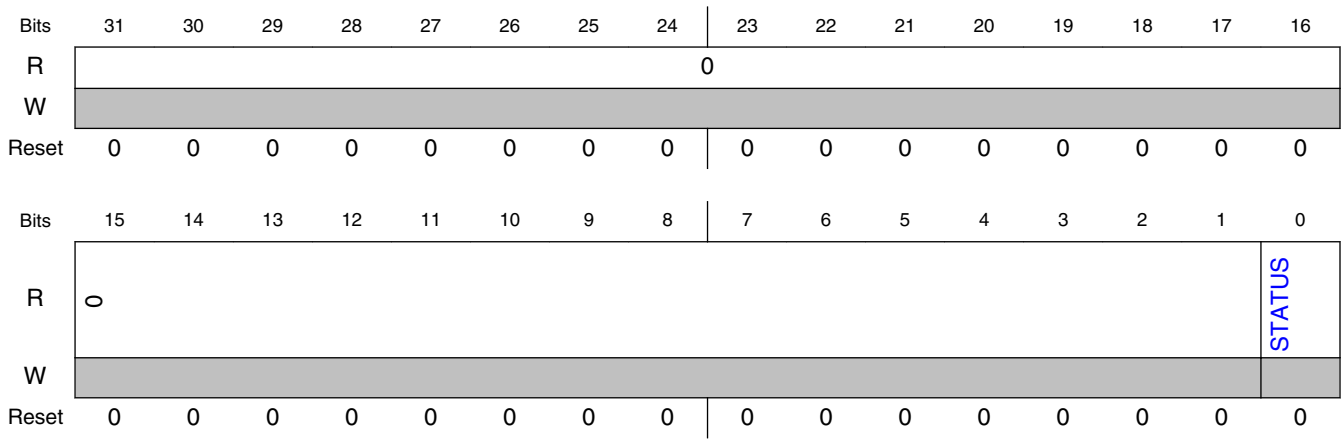
Field	Function
31-3 —	Reserved
2-0 DISABLE	Each bit of this field disables the corresponding interrupts in table above. 000b - Enable interrupts 001b - Disable interrupts

7.4.4.1.6 Channel n Master Status Register (CHn_MSTRSTAT)

7.4.4.1.6.1 Offset

Register	Offset
CHn_MSTRSTAT	44h

7.4.4.1.6.2 Diagram



7.4.4.1.6.3 Fields

Field	Function
31-1 —	Reserved
0 STATUS	Status of all interrupts 0b - No interrupts are asserted. 1b - At least one interrupt is asserted.

Chapter 8
Chip IO and Pinmux

8.1 External Signals and Pin Multiplexing

8.1.1 Overview

The chip contains a limited number of pins, most of which have multiple signal options. These signal-to-pin and pin-to-signal options are selected by the input-output multiplexer called IOMUX. The IOMUX is also used to configure other pin characteristics, such as voltage level, drive strength, and hysteresis.

The muxing options table lists the external signals grouped by the module instance, the muxing options for each signal, and the registers used to route the signal to the chosen pad.

8.1.1.1 Muxing Options

Instance	Port	Pad	Mode
CCM	CCM_CLKO1	ECSPI2_MISO	ALT4
		GPIO1_IO14	ALT6
	CCM_CLKO2	ECSPI2_SS0	ALT4
		GPIO1_IO15	ALT6
	CCM_ENET_PHY_REF_CLK_ROOT	GPIO1_IO00	ALT1
	CCM_EXT_CLK1	GPIO1_IO00	ALT6
	CCM_EXT_CLK2	GPIO1_IO01	ALT6
	CCM_EXT_CLK3	GPIO1_IO06	ALT6
	CCM_EXT_CLK4	GPIO1_IO07	ALT6
	CCM_PMIC_READY	GPIO1_IO05	ALT5
		GPIO1_IO11	ALT5

Table continues on the next page...

Instance	Port	Pad	Mode
	CCM_PMIC_STBY_REQ	PMIC_STBY_REQ	ALT0
	REF_CLK_24M	GPIO1_IO01	ALT5
	REF_CLK_32K	GPIO1_IO00	ALT5
CORESIGHT	CORESIGHT_EVENTI	SD2_WP	ALT6
	CORESIGHT_EVENTO	NAND_WP_B	ALT6
	CORESIGHT_TRACE00	NAND_CE1_B	ALT6
	CORESIGHT_TRACE01	NAND_CE2_B	ALT6
	CORESIGHT_TRACE02	NAND_CE3_B	ALT6
	CORESIGHT_TRACE03	NAND_CLE	ALT6
	CORESIGHT_TRACE04	NAND_DATA00	ALT6
	CORESIGHT_TRACE05	NAND_DATA01	ALT6
	CORESIGHT_TRACE06	NAND_DATA02	ALT6
	CORESIGHT_TRACE07	NAND_DATA03	ALT6
	CORESIGHT_TRACE08	NAND_DATA04	ALT6
	CORESIGHT_TRACE09	NAND_DATA05	ALT6
	CORESIGHT_TRACE10	NAND_DATA06	ALT6
	CORESIGHT_TRACE11	NAND_DATA07	ALT6
	CORESIGHT_TRACE12	NAND_DQS	ALT6
	CORESIGHT_TRACE13	NAND_RE_B	ALT6
	CORESIGHT_TRACE14	NAND_READY_B	ALT6
	CORESIGHT_TRACE15	NAND_WE_B	ALT6
	CORESIGHT_TRACE_CLK	NAND_ALE	ALT6
	CORESIGHT_TRACE_CTL	NAND_CE0_B	ALT6
DRAM	DRAM_AC00	DRAM_AC00	No Muxing
	DRAM_AC01	DRAM_AC01	No Muxing
	DRAM_AC02	DRAM_AC02	No Muxing
	DRAM_AC03	DRAM_AC03	No Muxing
	DRAM_AC04	DRAM_AC04	No Muxing
	DRAM_AC05	DRAM_AC05	No Muxing
	DRAM_AC06	DRAM_AC06	No Muxing
	DRAM_AC07	DRAM_AC07	No Muxing
	DRAM_AC08	DRAM_AC08	No Muxing
	DRAM_AC09	DRAM_AC09	No Muxing
	DRAM_AC10	DRAM_AC10	No Muxing
	DRAM_AC11	DRAM_AC11	No Muxing
	DRAM_AC12	DRAM_AC12	No Muxing
	DRAM_AC13	DRAM_AC13	No Muxing
	DRAM_AC14	DRAM_AC14	No Muxing
	DRAM_AC15	DRAM_AC15	No Muxing
	DRAM_AC16	DRAM_AC16	No Muxing
	DRAM_AC17	DRAM_AC17	No Muxing

Table continues on the next page...

Instance	Port	Pad	Mode
	DRAM_AC19	DRAM_AC19	No Muxing
	DRAM_AC20	DRAM_AC20	No Muxing
	DRAM_AC21	DRAM_AC21	No Muxing
	DRAM_AC22	DRAM_AC22	No Muxing
	DRAM_AC23	DRAM_AC23	No Muxing
	DRAM_AC24	DRAM_AC24	No Muxing
	DRAM_AC25	DRAM_AC25	No Muxing
	DRAM_AC26	DRAM_AC26	No Muxing
	DRAM_AC27	DRAM_AC27	No Muxing
	DRAM_AC28	DRAM_AC28	No Muxing
	DRAM_AC29	DRAM_AC29	No Muxing
	DRAM_AC30	DRAM_AC30	No Muxing
	DRAM_AC31	DRAM_AC31	No Muxing
	DRAM_AC32	DRAM_AC32	No Muxing
	DRAM_AC33	DRAM_AC33	No Muxing
	DRAM_AC34	DRAM_AC34	No Muxing
	DRAM_AC35	DRAM_AC35	No Muxing
	DRAM_AC36	DRAM_AC36	No Muxing
	DRAM_AC37	DRAM_AC37	No Muxing
	DRAM_AC38	DRAM_AC38	No Muxing
	DRAM_ALERT_N	DRAM_ALERT_N	No Muxing
	DRAM_DM0	DRAM_DM0	No Muxing
	DRAM_DM1	DRAM_DM1	No Muxing
	DRAM_DM2	DRAM_DM2	No Muxing
	DRAM_DM3	DRAM_DM3	No Muxing
	DRAM_DQ00	DRAM_DQ00	No Muxing
	DRAM_DQ01	DRAM_DQ01	No Muxing
	DRAM_DQ02	DRAM_DQ02	No Muxing
	DRAM_DQ03	DRAM_DQ03	No Muxing
	DRAM_DQ04	DRAM_DQ04	No Muxing
	DRAM_DQ05	DRAM_DQ05	No Muxing
	DRAM_DQ06	DRAM_DQ06	No Muxing
	DRAM_DQ07	DRAM_DQ07	No Muxing
	DRAM_DQ08	DRAM_DQ08	No Muxing
	DRAM_DQ09	DRAM_DQ09	No Muxing
	DRAM_DQ10	DRAM_DQ10	No Muxing
	DRAM_DQ11	DRAM_DQ11	No Muxing
	DRAM_DQ12	DRAM_DQ12	No Muxing
	DRAM_DQ13	DRAM_DQ13	No Muxing
	DRAM_DQ14	DRAM_DQ14	No Muxing
	DRAM_DQ15	DRAM_DQ15	No Muxing

Table continues on the next page...

Instance	Port	Pad	Mode
	DRAM_DQ16	DRAM_DQ16	No Muxing
	DRAM_DQ17	DRAM_DQ17	No Muxing
	DRAM_DQ18	DRAM_DQ18	No Muxing
	DRAM_DQ19	DRAM_DQ19	No Muxing
	DRAM_DQ20	DRAM_DQ20	No Muxing
	DRAM_DQ21	DRAM_DQ21	No Muxing
	DRAM_DQ22	DRAM_DQ22	No Muxing
	DRAM_DQ23	DRAM_DQ23	No Muxing
	DRAM_DQ24	DRAM_DQ24	No Muxing
	DRAM_DQ25	DRAM_DQ25	No Muxing
	DRAM_DQ26	DRAM_DQ26	No Muxing
	DRAM_DQ27	DRAM_DQ27	No Muxing
	DRAM_DQ28	DRAM_DQ28	No Muxing
	DRAM_DQ29	DRAM_DQ29	No Muxing
	DRAM_DQ30	DRAM_DQ30	No Muxing
	DRAM_DQ31	DRAM_DQ31	No Muxing
	DRAM_DQS0_N	DRAM_DQS0_N	No Muxing
	DRAM_DQS0_P	DRAM_DQS0_P	No Muxing
	DRAM_DQS1_N	DRAM_DQS1_N	No Muxing
	DRAM_DQS1_P	DRAM_DQS1_P	No Muxing
	DRAM_DQS2_N	DRAM_DQS2_N	No Muxing
	DRAM_DQS2_P	DRAM_DQS2_P	No Muxing
	DRAM_DQS3_N	DRAM_DQS3_N	No Muxing
	DRAM_DQS3_P	DRAM_DQS3_P	No Muxing
	DRAM_RESET_N	DRAM_RESET_N	No Muxing
	DRAM_VREF	DRAM_VREF	No Muxing
	DRAM_ZN	DRAM_ZN	No Muxing
EARC	EARC_AUX	EARC_AUX	No Muxing
	EARC_P_UTIL	EARC_P_UTIL	No Muxing
	EARC_N_HPD	EARC_N_HPD	No Muxing
ECSPI1	ECSPI1_MISO	ECSPI1_MISO	ALT0
		I2C2_SCL	ALT3
	ECSPI1_MOSI	ECSPI1_MOSI	ALT0
		I2C1_SDA	ALT3
	ECSPI1_SCLK	ECSPI1_SCLK	ALT0
		I2C1_SCL	ALT3
	ECSPI1_SS0	ECSPI1_SS0	ALT0
		I2C2_SDA	ALT3
ECSPI2	ECSPI2_MISO	ECSPI2_MISO	ALT0
		SD2_DATA3	ALT2
		I2C4_SCL	ALT3

Table continues on the next page...

Instance	Port	Pad	Mode
	ECSPI2_MOSI	ECSPI2_MOSI	ALT0
		SD2_CMD	ALT2
		I2C3_SDA	ALT3
	ECSPI2_SCLK	ECSPI2_SCLK	ALT0
		SD2_CLK	ALT2
		I2C3_SCL	ALT3
	ECSPI2_SS0	ECSPI2_SS0	ALT0
		SD2_DATA2	ALT2
		I2C4_SDA	ALT3
ECSPI3	ECSPI3_MISO	UART2_RXD	ALT1
	ECSPI3_MOSI	UART1_TXD	ALT1
	ECSPI3_SCLK	UART1_RXD	ALT1
	ECSPI3_SS0	UART2_TXD	ALT1
ENET_QOS	ENET_QOS_1588_EVENT0_AUX_IN	GPIO1_IO08	ALT4
	ENET_QOS_1588_EVENT0_IN	GPIO1_IO08	ALT1
	ENET_QOS_1588_EVENT0_OUT	GPIO1_IO09	ALT1
	ENET_QOS_1588_EVENT1_AUX_IN	I2C2_SCL	ALT4
	ENET_QOS_1588_EVENT1_IN	I2C2_SCL	ALT1
	ENET_QOS_1588_EVENT1_OUT	I2C2_SDA	ALT1
	ENET_QOS_1588_EVENT2_AUX_IN	SAI2_TXD0	ALT4
	ENET_QOS_1588_EVENT2_IN	SAI2_TXD0	ALT2
	ENET_QOS_1588_EVENT2_OUT	SAI2_RXD0	ALT2
	ENET_QOS_1588_EVENT3_AUX_IN	SAI2_MCLK	ALT4
	ENET_QOS_1588_EVENT3_IN	SAI2_MCLK	ALT2
	ENET_QOS_1588_EVENT3_OUT	SAI2_TXFS	ALT2
	ENET_QOS_INPUT=ENET_QOS_TX_CLK, OUTPUT=CCM_ENET_QOS_REF_CLK_ROOT	ENET_TD2	ALT1
	ENET_QOS_MDC	ENET_MDC	ALT0
		GPIO1_IO06	ALT1
		I2C1_SCL	ALT1
	ENET_QOS_MDIO	ENET_MDIO	ALT0

Table continues on the next page...

Instance	Port	Pad	Mode
		GPIO1_IO07	ALT1
		I2C1_SDA	ALT1
	ENET_QOS_RGMII_RD0	ENET_RD0	ALT0
	ENET_QOS_RGMII_RD1	ENET_RD1	ALT0
	ENET_QOS_RGMII_RD2	ENET_RD2	ALT0
	ENET_QOS_RGMII_RD3	ENET_RD3	ALT0
	ENET_QOS_RGMII_RX_CTL	ENET_RX_CTL	ALT0
	ENET_QOS_RGMII_RXC	ENET_RXC	ALT0
	ENET_QOS_RGMII_TD0	ENET_TD0	ALT0
	ENET_QOS_RGMII_TD1	ENET_TD1	ALT0
	ENET_QOS_RGMII_TD2	ENET_TD2	ALT0
	ENET_QOS_RGMII_TD3	ENET_TD3	ALT0
	ENET_QOS_RGMII_TX_CTL	ENET_TX_CTL	ALT0
	ENET_QOS_RGMII_TXC	ENET_TXC	ALT0
	ENET_QOS_RX_ER	ENET_RXC	ALT1
	ENET_QOS_TX_ER	ENET_TXC	ALT1
ENET1	ENET1_1588_EVENT0_IN	SAI1_RXFS	ALT4
	ENET1_1588_EVENT0_OUT	SAI1_RXC	ALT4
	ENET1_1588_EVENT1_IN	SAI1_RXD0	ALT4
	ENET1_1588_EVENT1_OUT	SAI1_RXD1	ALT4
	ENET1_INPUT=ENET1_TX_CLK, OUTPUT=CCM_ENET_REF_CLK_ROOT	SD1_RESET_B	ALT1
	ENET1_INPUT=ENET1_TX_CLK, OUTPUT=CCM_ENET_REF_CLK_ROOT	SAI1_MCLK	ALT4
	ENET1_MDC	SD1_CLK	ALT1
		SAI1_RXD2	ALT4
	ENET1_MDIO	SD1_CMD	ALT1
		SAI1_RXD3	ALT4
	ENET1_RGMII_RD0	SD1_DATA2	ALT1
		SAI1_RXD4	ALT4
	ENET1_RGMII_RD1	SD1_DATA3	ALT1
		SAI1_RXD5	ALT4
	ENET1_RGMII_RD2	SAI1_RXD6	ALT4
	ENET1_RGMII_RD3	SAI1_RXD7	ALT4
	ENET1_RGMII_RX_CTL	SD1_DATA6	ALT1
		SAI1_TXFS	ALT4
	ENET1_RGMII_RXC	SAI1_TXC	ALT4
	ENET1_RGMII_TD0	SD1_DATA1	ALT1

Table continues on the next page...

Instance	Port	Pad	Mode
		SAI1_TXD0	ALT4
	ENET1_RGMII_TD1	SD1_DATA0	ALT1
		SAI1_TXD1	ALT4
	ENET1_RGMII_TD2	SAI1_TXD2	ALT4
	ENET1_RGMII_TD3	SAI1_TXD3	ALT4
	ENET1_RGMII_TX_CTL	SD1_DATA4	ALT1
		SAI1_TXD4	ALT4
	ENET1_RGMII_TXC	SAI1_TXD5	ALT4
	ENET1_RX_ER	SD1_DATA7	ALT1
		SAI1_TXD6	ALT4
	ENET1_TX_ER	SD1_DATA5	ALT1
		SAI1_TXD7	ALT4
FLEXCAN1	FLEXCAN1_RX	SAI2_TXC	ALT3
		SPDIF_RX	ALT4
		HDMI_DDC_SDA	ALT4
		SAI5_RXD2	ALT6
	FLEXCAN1_TX	SAI2_RXC	ALT3
		SPDIF_TX	ALT4
		HDMI_DDC_SCL	ALT4
		SAI5_RXD1	ALT6
FLEXCAN2	FLEXCAN2_RX	SAI2_MCLK	ALT3
		UART3_TXD	ALT4
		HDMI_HPD	ALT4
		SAI5_MCLK	ALT6
	FLEXCAN2_TX	SAI2_TXD0	ALT3
		UART3_RXD	ALT4
		HDMI_CEC	ALT4
		SAI5_RXD3	ALT6
GPIO1	GPIO1_IO00	GPIO1_IO00	ALT0
	GPIO1_IO01	GPIO1_IO01	ALT0
	GPIO1_IO02	GPIO1_IO02	ALT0
	GPIO1_IO03	GPIO1_IO03	ALT0
	GPIO1_IO04	GPIO1_IO04	ALT0
	GPIO1_IO05	GPIO1_IO05	ALT0
	GPIO1_IO06	GPIO1_IO06	ALT0
	GPIO1_IO07	GPIO1_IO07	ALT0
	GPIO1_IO08	GPIO1_IO08	ALT0
	GPIO1_IO09	GPIO1_IO09	ALT0
	GPIO1_IO10	GPIO1_IO10	ALT0
	GPIO1_IO11	GPIO1_IO11	ALT0
	GPIO1_IO12	GPIO1_IO12	ALT0

Table continues on the next page...

Instance	Port	Pad	Mode
	GPIO1_IO13	GPIO1_IO13	ALT0
	GPIO1_IO14	GPIO1_IO14	ALT0
	GPIO1_IO15	GPIO1_IO15	ALT0
	GPIO1_IO16	ENET_MDC	ALT5
	GPIO1_IO17	ENET_MDIO	ALT5
	GPIO1_IO18	ENET_TD3	ALT5
	GPIO1_IO19	ENET_TD2	ALT5
	GPIO1_IO20	ENET_TD1	ALT5
	GPIO1_IO21	ENET_TD0	ALT5
	GPIO1_IO22	ENET_TX_CTL	ALT5
	GPIO1_IO23	ENET_TXC	ALT5
	GPIO1_IO24	ENET_RX_CTL	ALT5
	GPIO1_IO25	ENET_RXC	ALT5
	GPIO1_IO26	ENET_RD0	ALT5
	GPIO1_IO27	ENET_RD1	ALT5
	GPIO1_IO28	ENET_RD2	ALT5
	GPIO1_IO29	ENET_RD3	ALT5
GPIO2	GPIO2_IO00	SD1_CLK	ALT5
	GPIO2_IO01	SD1_CMD	ALT5
	GPIO2_IO02	SD1_DATA0	ALT5
	GPIO2_IO03	SD1_DATA1	ALT5
	GPIO2_IO04	SD1_DATA2	ALT5
	GPIO2_IO05	SD1_DATA3	ALT5
	GPIO2_IO06	SD1_DATA4	ALT5
	GPIO2_IO07	SD1_DATA5	ALT5
	GPIO2_IO08	SD1_DATA6	ALT5
	GPIO2_IO09	SD1_DATA7	ALT5
	GPIO2_IO10	SD1_RESET_B	ALT5
	GPIO2_IO11	SD1_STROBE	ALT5
	GPIO2_IO12	SD2_CD_B	ALT5
	GPIO2_IO13	SD2_CLK	ALT5
	GPIO2_IO14	SD2_CMD	ALT5
	GPIO2_IO15	SD2_DATA0	ALT5
	GPIO2_IO16	SD2_DATA1	ALT5
	GPIO2_IO17	SD2_DATA2	ALT5
	GPIO2_IO18	SD2_DATA3	ALT5
	GPIO2_IO19	SD2_RESET_B	ALT5
	GPIO2_IO20	SD2_WP	ALT5
GPIO3	GPIO3_IO00	NAND_ALE	ALT5
	GPIO3_IO01	NAND_CE0_B	ALT5
	GPIO3_IO02	NAND_CE1_B	ALT5

Table continues on the next page...

Instance	Port	Pad	Mode
	GPIO3_IO03	NAND_CE2_B	ALT5
	GPIO3_IO04	NAND_CE3_B	ALT5
	GPIO3_IO05	NAND_CLE	ALT5
	GPIO3_IO06	NAND_DATA00	ALT5
	GPIO3_IO07	NAND_DATA01	ALT5
	GPIO3_IO08	NAND_DATA02	ALT5
	GPIO3_IO09	NAND_DATA03	ALT5
	GPIO3_IO10	NAND_DATA04	ALT5
	GPIO3_IO11	NAND_DATA05	ALT5
	GPIO3_IO12	NAND_DATA06	ALT5
	GPIO3_IO13	NAND_DATA07	ALT5
	GPIO3_IO14	NAND_DQS	ALT5
	GPIO3_IO15	NAND_RE_B	ALT5
	GPIO3_IO16	NAND_READY_B	ALT5
	GPIO3_IO17	NAND_WE_B	ALT5
	GPIO3_IO18	NAND_WP_B	ALT5
	GPIO3_IO19	SAI5_RXFS	ALT5
	GPIO3_IO20	SAI5_RXC	ALT5
	GPIO3_IO21	SAI5_RXD0	ALT5
	GPIO3_IO22	SAI5_RXD1	ALT5
	GPIO3_IO23	SAI5_RXD2	ALT5
	GPIO3_IO24	SAI5_RXD3	ALT5
	GPIO3_IO25	SAI5_MCLK	ALT5
	GPIO3_IO26	HDMI_DDC_SCL	ALT5
	GPIO3_IO27	HDMI_DDC_SDA	ALT5
	GPIO3_IO28	HDMI_CEC	ALT5
	GPIO3_IO29	HDMI_HPD	ALT5
GPIO4	GPIO4_IO00	SAI1_RXFS	ALT5
	GPIO4_IO01	SAI1_RXC	ALT5
	GPIO4_IO02	SAI1_RXD0	ALT5
	GPIO4_IO03	SAI1_RXD1	ALT5
	GPIO4_IO04	SAI1_RXD2	ALT5
	GPIO4_IO05	SAI1_RXD3	ALT5
	GPIO4_IO06	SAI1_RXD4	ALT5
	GPIO4_IO07	SAI1_RXD5	ALT5
	GPIO4_IO08	SAI1_RXD6	ALT5
	GPIO4_IO09	SAI1_RXD7	ALT5
	GPIO4_IO10	SAI1_TXFS	ALT5
	GPIO4_IO11	SAI1_TXC	ALT5
	GPIO4_IO12	SAI1_TXD0	ALT5
	GPIO4_IO13	SAI1_TXD1	ALT5

Table continues on the next page...

Instance	Port	Pad	Mode
	GPIO4_IO14	SAI1_TXD2	ALT5
	GPIO4_IO15	SAI1_TXD3	ALT5
	GPIO4_IO16	SAI1_TXD4	ALT5
	GPIO4_IO17	SAI1_TXD5	ALT5
	GPIO4_IO18	SAI1_TXD6	ALT5
	GPIO4_IO19	SAI1_TXD7	ALT5
	GPIO4_IO20	SAI1_MCLK	ALT5
	GPIO4_IO21	SAI2_RXFS	ALT5
	GPIO4_IO22	SAI2_RXC	ALT5
	GPIO4_IO23	SAI2_RXD0	ALT5
	GPIO4_IO24	SAI2_TXFS	ALT5
	GPIO4_IO25	SAI2_TXC	ALT5
	GPIO4_IO26	SAI2_TXD0	ALT5
	GPIO4_IO27	SAI2_MCLK	ALT5
	GPIO4_IO28	SAI3_RXFS	ALT5
	GPIO4_IO29	SAI3_RXC	ALT5
	GPIO4_IO30	SAI3_RXD	ALT5
	GPIO4_IO31	SAI3_TXFS	ALT5
GPIO5	GPIO5_IO00	SAI3_TXC	ALT5
	GPIO5_IO01	SAI3_TXD	ALT5
	GPIO5_IO02	SAI3_MCLK	ALT5
	GPIO5_IO03	SPDIF_TX	ALT5
	GPIO5_IO04	SPDIF_RX	ALT5
	GPIO5_IO05	SPDIF_EXT_CLK	ALT5
	GPIO5_IO06	ECSPI1_SCLK	ALT5
	GPIO5_IO07	ECSPI1_MOSI	ALT5
	GPIO5_IO08	ECSPI1_MISO	ALT5
	GPIO5_IO09	ECSPI1_SS0	ALT5
	GPIO5_IO10	ECSPI2_SCLK	ALT5
	GPIO5_IO11	ECSPI2_MOSI	ALT5
	GPIO5_IO12	ECSPI2_MISO	ALT5
	GPIO5_IO13	ECSPI2_SS0	ALT5
	GPIO5_IO14	I2C1_SCL	ALT5
	GPIO5_IO15	I2C1_SDA	ALT5
	GPIO5_IO16	I2C2_SCL	ALT5
	GPIO5_IO17	I2C2_SDA	ALT5
	GPIO5_IO18	I2C3_SCL	ALT5
	GPIO5_IO19	I2C3_SDA	ALT5
	GPIO5_IO20	I2C4_SCL	ALT5
	GPIO5_IO21	I2C4_SDA	ALT5
	GPIO5_IO22	UART1_RXD	ALT5

Table continues on the next page...

Instance	Port	Pad	Mode
	GPIO5_IO23	UART1_TXD	ALT5
	GPIO5_IO24	UART2_RXD	ALT5
	GPIO5_IO25	UART2_TXD	ALT5
	GPIO5_IO26	UART3_RXD	ALT5
	GPIO5_IO27	UART3_TXD	ALT5
	GPIO5_IO28	UART4_RXD	ALT5
	GPIO5_IO29	UART4_TXD	ALT5
GPT1	GPT1_CAPTURE1	SAI3_TXC	ALT3
		UART4_TXD	ALT3
	GPT1_CAPTURE2	SAI3_TXD	ALT3
		UART3_RXD	ALT3
	GPT1_CLK	SAI3_RXC	ALT3
		UART3_TXD	ALT3
	GPT1_COMPARE1	SPDIF_TX	ALT3
		UART4_RXD	ALT3
	GPT1_COMPARE2	SPDIF_RX	ALT3
		UART2_TXD	ALT3
	GPT1_COMPARE3	SPDIF_EXT_CLK	ALT3
		UART2_RXD	ALT3
GPT2	GPT2_CLK	I2C3_SCL	ALT2
GPT3	GPT3_CLK	I2C3_SDA	ALT2
HDMI	HDMI_CEC	HDMI_CEC	ALT0
	HDMI_HPD	HDMI_HPD	ALT0
	HDMI_HPD_O	HDMI_HPD	ALT1
	HDMI_SCL	HDMI_DDC_SCL	ALT0
	HDMI_SDA	HDMI_DDC_SDA	ALT0
	HDMI_TX0N	HDMI_TX0_N	No Muxing
	HDMI_TX0P	HDMI_TX0_P	No Muxing
	HDMI_TX1N	HDMI_TX1_N	No Muxing
	HDMI_TX1P	HDMI_TX1_P	No Muxing
	HDMI_TX2N	HDMI_TX2_N	No Muxing
	HDMI_TX2P	HDMI_TX2_P	No Muxing
	HDMI_TXCN	HDMI_TXC_N	No Muxing
	HDMI_TXCP	HDMI_TXC_P	No Muxing
	HDMI_REXT	HDMI_REXT	No Muxing
I2C1	I2C1_SCL	I2C1_SCL	ALT0
		ECSPI1_SCLK	ALT2
		SD1_DATA4	ALT3
	I2C1_SDA	I2C1_SDA	ALT0
		ECSPI1_MOSI	ALT2
		SD1_DATA5	ALT3

Table continues on the next page...

Instance	Port	Pad	Mode
I2C2	I2C2_SCL	I2C2_SCL	ALT0
		ECSPI1_MISO	ALT2
		SD1_DATA6	ALT3
	I2C2_SDA	I2C2_SDA	ALT0
		ECSPI1_SS0	ALT2
		SD1_DATA7	ALT3
I2C3	I2C3_SCL	I2C3_SCL	ALT0
		ECSPI2_SCLK	ALT2
		SD1_RESET_B	ALT3
		NAND_DQS	ALT4
		NAND_READY_B	ALT4
	I2C3_SDA	I2C3_SDA	ALT0
		ECSPI2_MOSI	ALT2
		SD1_STROBE	ALT3
		NAND_CE3_B	ALT4
		NAND_WE_B	ALT4
I2C4	I2C4_SCL	I2C4_SCL	ALT0
		SD2_DATA1	ALT2
		ECSPI2_MISO	ALT2
		SD1_DATA2	ALT3
		NAND_CE1_B	ALT4
		NAND_WP_B	ALT4
	I2C4_SDA	I2C4_SDA	ALT0
		SD2_DATA0	ALT2
		ECSPI2_SS0	ALT2
		SD1_DATA3	ALT3
		NAND_CE2_B	ALT4
		NAND_DATA02	ALT4
I2C5	I2C5_SCL	SPDIF_TX	ALT2
		SD1_CLK	ALT3
		SAI5_RXD0	ALT3
		HDMI_DDC_SCL	ALT3
	I2C5_SDA	SPDIF_RX	ALT2
		SD1_CMD	ALT3
		SAI5_MCLK	ALT3
		HDMI_DDC_SDA	ALT3
I2C6	I2C6_SCL	SD1_DATA0	ALT3
		SAI5_RXFS	ALT3
		HDMI_CEC	ALT3
		UART4_RXD	ALT4
	I2C6_SDA	SD1_DATA1	ALT3

Table continues on the next page...

Instance	Port	Pad	Mode
		SAI5_RXC	ALT3
		HDMI_HPD	ALT3
		UART4_TXD	ALT4
ISP	ISP_FL_TRIG_0	GPIO1_IO00	ALT3
		NAND_ALE	ALT3
	ISP_FL_TRIG_1	GPIO1_IO05	ALT3
		NAND_DATA03	ALT4
	ISP_FLASH_TRIG_0	GPIO1_IO02	ALT3
		NAND_DATA00	ALT3
	ISP_FLASH_TRIG_1	GPIO1_IO07	ALT3
		NAND_DATA05	ALT4
	ISP_PRELIGHT_TRIG_0	GPIO1_IO03	ALT3
		NAND_DATA01	ALT3
	ISP_PRELIGHT_TRIG_1	GPIO1_IO08	ALT3
		NAND_DATA06	ALT4
	ISP_SHUTTER_OPEN_0	GPIO1_IO04	ALT3
		NAND_DQS	ALT3
	ISP_SHUTTER_OPEN_1	GPIO1_IO09	ALT3
		NAND_DATA07	ALT4
	ISP_SHUTTER_TRIG_0	GPIO1_IO01	ALT3
		NAND_CE0_B	ALT3
	ISP_SHUTTER_TRIG_1	GPIO1_IO06	ALT3
		NAND_DATA04	ALT4
JTAG	JTAG_MODE	JTAG_MOD	ALT0
	JTAG_TCK	JTAG_TCK	ALT0
	JTAG_TDI	JTAG_TDI	ALT0
	JTAG_TDO	JTAG_TDO	ALT0
	JTAG_TMS	JTAG_TMS	ALT0
LVDS0	LVDS0_CLKN	LVDS0_CLK_N	No Muxing
	LVDS0_CLKP	LVDS0_CLK_P	No Muxing
	LVDS0_D0N	LVDS0_D0_N	No Muxing
	LVDS0_D0P	LVDS0_D0_P	No Muxing
	LVDS0_D1N	LVDS0_D1_N	No Muxing
	LVDS0_D1P	LVDS0_D1_P	No Muxing
	LVDS0_D2N	LVDS0_D2_N	No Muxing
	LVDS0_D2P	LVDS0_D2_P	No Muxing
	LVDS0_D3N	LVDS0_D3_N	No Muxing
	LVDS0_D3P	LVDS0_D3_P	No Muxing
	LVDS1_CLKN	LVDS1_CLK_N	No Muxing
	LVDS1_CLKP	LVDS1_CLK_P	No Muxing
	LVDS1_D0N	LVDS1_D0_N	No Muxing

Table continues on the next page...

Instance	Port	Pad	Mode
	LVDS1_D0P	LVDS1_D0_P	No Muxing
	LVDS1_D1N	LVDS1_D1_N	No Muxing
	LVDS1_D1P	LVDS1_D1_P	No Muxing
	LVDS1_D2N	LVDS1_D2_N	No Muxing
	LVDS1_D2P	LVDS1_D2_P	No Muxing
	LVDS1_D3N	LVDS1_D3_N	No Muxing
	LVDS1_D3P	LVDS1_D3_P	No Muxing
M7	M7_NMI	GPIO1_IO05	ALT1
MIPI CSI1	MIPI_CSI1_CLK_N	MIPI_CSI1_CLK_N	No Muxing
	MIPI_CSI1_CLK_P	MIPI_CSI1_CLK_P	No Muxing
	MIPI_CSI1_D0_N	MIPI_CSI1_D0_N	No Muxing
	MIPI_CSI1_D0_P	MIPI_CSI1_D0_P	No Muxing
	MIPI_CSI1_D1_N	MIPI_CSI1_D1_N	No Muxing
	MIPI_CSI1_D1_P	MIPI_CSI1_D1_P	No Muxing
	MIPI_CSI1_D2_N	MIPI_CSI1_D2_N	No Muxing
	MIPI_CSI1_D2_P	MIPI_CSI1_D2_P	No Muxing
	MIPI_CSI1_D3_N	MIPI_CSI1_D3_N	No Muxing
	MIPI_CSI1_D3_P	MIPI_CSI1_D3_P	No Muxing
	MIPI1_VREG	MIPI_VREG1_CAP	No Muxing
MIPI CSI2	MIPI_CSI2_CLK_N	MIPI_CSI2_CLK_N	No Muxing
	MIPI_CSI2_CLK_P	MIPI_CSI2_CLK_P	No Muxing
	MIPI_CSI2_D0_N	MIPI_CSI2_D0_N	No Muxing
	MIPI_CSI2_D0_P	MIPI_CSI2_D0_P	No Muxing
	MIPI_CSI2_D1_N	MIPI_CSI2_D1_N	No Muxing
	MIPI_CSI2_D1_P	MIPI_CSI2_D1_P	No Muxing
	MIPI_CSI2_D2_N	MIPI_CSI2_D2_N	No Muxing
	MIPI_CSI2_D2_P	MIPI_CSI2_D2_P	No Muxing
	MIPI_CSI2_D3_N	MIPI_CSI2_D3_N	No Muxing
	MIPI_CSI2_D3_P	MIPI_CSI2_D3_P	No Muxing
	MIPI2_VREG	MIPI_VREG2_CAP	No Muxing
MIPI DSI1	MIPI_DSI1_CLK_N	MIPI_DSI1_CLK_N	No Muxing
	MIPI_DSI1_CLK_P	MIPI_DSI1_CLK_P	No Muxing
	MIPI_DSI1_D0_N	MIPI_DSI1_D0_N	No Muxing
	MIPI_DSI1_D0_P	MIPI_DSI1_D0_P	No Muxing
	MIPI_DSI1_D1_N	MIPI_DSI1_D1_N	No Muxing
	MIPI_DSI1_D1_P	MIPI_DSI1_D1_P	No Muxing
	MIPI_DSI1_D2_N	MIPI_DSI1_D2_N	No Muxing
	MIPI_DSI1_D2_P	MIPI_DSI1_D2_P	No Muxing
	MIPI_DSI1_D3_N	MIPI_DSI1_D3_N	No Muxing
	MIPI_DSI1_D3_P	MIPI_DSI1_D3_P	No Muxing
NAND	NAND_ALE	NAND_ALE	ALT0

Table continues on the next page...

Instance	Port	Pad	Mode
	NAND_CE0_B	NAND_CE0_B	ALT0
	NAND_CE1_B	NAND_CE1_B	ALT0
	NAND_CE2_B	NAND_CE2_B	ALT0
	NAND_CE3_B	NAND_CE3_B	ALT0
	NAND_CLE	NAND_CLE	ALT0
	NAND_DATA00	NAND_DATA00	ALT0
	NAND_DATA01	NAND_DATA01	ALT0
	NAND_DATA02	NAND_DATA02	ALT0
	NAND_DATA03	NAND_DATA03	ALT0
	NAND_DATA04	NAND_DATA04	ALT0
	NAND_DATA05	NAND_DATA05	ALT0
	NAND_DATA06	NAND_DATA06	ALT0
	NAND_DATA07	NAND_DATA07	ALT0
	NAND_DQS	NAND_DQS	ALT0
	NAND_RE_B	NAND_RE_B	ALT0
	NAND_READY_B	NAND_READY_B	ALT0
	NAND_WE_B	NAND_WE_B	ALT0
	NAND_WP_B	NAND_WP_B	ALT0
PCIE1	PCIE1_CLKREQ_B	I2C4_SCL	ALT2
		UART4_RXD	ALT2
	PCIE_REF_PAD_CLK_N	PCIE_REF_PAD_CLK_N	No Muxing
	PCIE_REF_PAD_CLK_P	PCIE_REF_PAD_CLK_P	No Muxing
	PCIE_RESREF	PCIE_RESREF	No Muxing
	PCIE_RXN_N	PCIE_RXN_N	No Muxing
	PCIE_RXN_P	PCIE_RXN_P	No Muxing
	PCIE_TXN_N	PCIE_TXN_N	No Muxing
	PCIE_TXN_P	PCIE_TXN_P	No Muxing
PDM	PDM_BIT_STREAM0	ENET_TD1	ALT3
		ENET_RD1	ALT3
		SAI1_RXD0	ALT3
		SD2_DATA0	ALT4
		SAI5_RXD0	ALT4
		SAI3_RXFS	ALT6
	PDM_BIT_STREAM1	ENET_TD2	ALT3
		ENET_RD0	ALT3
		SAI1_RXD1	ALT3
		SD2_DATA1	ALT4
		SAI5_RXD1	ALT4
		SAI2_RXC	ALT6
		SAI2_TXC	ALT6
		SAI3_RXD	ALT6

Table continues on the next page...

Instance	Port	Pad	Mode
	PDM_BIT_STREAM2	ENET_TD3	ALT3
		ENET_RXC	ALT3
		SAI1_RXD2	ALT3
		SD2_DATA2	ALT4
		SAI5_RXD2	ALT4
		SAI2_RXFS	ALT6
		SAI2_TXFS	ALT6
		SAI3_TXC	ALT6
	PDM_BIT_STREAM3	ENET_MDIO	ALT3
		ENET_RX_CTL	ALT3
		SAI1_RXD3	ALT3
		SD2_DATA3	ALT4
		SAI5_RXD3	ALT4
		SAI2_RXD0	ALT6
		SAI3_TXFS	ALT6
	PDM_CLK	ENET_TD0	ALT3
		ENET_RD2	ALT3
		SAI1_RXC	ALT3
		SAI1_TXD7	ALT3
		SD2_CMD	ALT4
		SAI5_RXC	ALT4
		SAI3_RXC	ALT6
PWM1	PWM1_OUT	GPIO1_IO01	ALT1
		SPDIF_EXT_CLK	ALT1
		I2C4_SDA	ALT1
		GPIO1_IO08	ALT2
		SAI5_MCLK	ALT2
PWM2	PWM2_OUT	SPDIF_RX	ALT1
		I2C4_SCL	ALT1
		GPIO1_IO09	ALT2
		GPIO1_IO11	ALT2
		SAI5_RXD0	ALT2
		GPIO1_IO13	ALT5
PWM3	PWM3_OUT	SPDIF_TX	ALT1
		I2C3_SDA	ALT1
		GPIO1_IO10	ALT2
		SAI5_RXC	ALT2
		GPIO1_IO14	ALT5
PWM4	PWM4_OUT	SAI3_MCLK	ALT1
		I2C3_SCL	ALT1
		SAI5_RXFS	ALT2

Table continues on the next page...

Instance	Port	Pad	Mode
		GPIO1_IO15	ALT5
QSPI	QSPI_A_DATA0	NAND_DATA00	ALT1
	QSPI_A_DATA1	NAND_DATA01	ALT1
	QSPI_A_DATA2	NAND_DATA02	ALT1
	QSPI_A_DATA3	NAND_DATA03	ALT1
	QSPI_A_DATA4	NAND_DATA04	ALT3
	QSPI_A_DATA5	NAND_DATA05	ALT3
	QSPI_A_DATA6	NAND_DATA06	ALT3
	QSPI_A_DATA7	NAND_DATA07	ALT3
	QSPI_A_DQS	NAND_DQS	ALT1
	QSPI_A_SCLK	NAND_ALE	ALT1
	QSPI_A_SS0_B	NAND_CE0_B	ALT1
	QSPI_A_SS1_B	NAND_CE1_B	ALT1
	QSPI_B_DATA0	NAND_DATA04	ALT1
	QSPI_B_DATA1	NAND_DATA05	ALT1
	QSPI_B_DATA2	NAND_DATA06	ALT1
	QSPI_B_DATA3	NAND_DATA07	ALT1
	QSPI_B_DQS	NAND_RE_B	ALT1
	QSPI_B_SCLK	NAND_CLE	ALT1
	QSPI_B_SS0_B	NAND_CE2_B	ALT1
	QSPI_B_SS1_B	NAND_CE3_B	ALT1
SAI1	SAI1_MCLK	SAI1_MCLK	ALT0
	SAI1_RX_BCLK	SAI1_RXC	ALT0
	SAI1_RX_DATA0	SAI1_RXD0	ALT0
	SAI1_RX_DATA1	SAI1_RXD1	ALT0
	SAI1_RX_DATA2	SAI1_RXD2	ALT0
	SAI1_RX_DATA3	SAI1_RXD3	ALT0
	SAI1_RX_DATA4	SAI1_RXD4	ALT0
	SAI1_RX_DATA5	SAI1_RXD5	ALT0
	SAI1_RX_DATA6	SAI1_RXD6	ALT0
	SAI1_RX_DATA7	SAI1_RXD7	ALT0
	SAI1_RX_SYNC	SAI1_RXFS	ALT0
		SAI1_RXD5	ALT3
	SAI1_TX_BCLK	SAI1_TXC	ALT0
		SAI5_MCLK	ALT1
		SAI1_MCLK	ALT2
	SAI1_TX_DATA0	SAI1_TXD0	ALT0
		SAI5_RXFS	ALT1
	SAI1_TX_DATA1	SAI1_TXD1	ALT0
		SAI5_RXC	ALT1
		SAI1_RXD0	ALT2

Table continues on the next page...

Instance	Port	Pad	Mode
	SAI1_TX_DATA2	SAI1_TXD2	ALT0
		SAI5_RXD0	ALT1
	SAI1_TX_DATA3	SAI1_TXD3	ALT0
		SAI5_RXD1	ALT1
	SAI1_TX_DATA4	SAI1_TXD4	ALT0
		SAI5_RXD2	ALT1
		SAI1_RXD7	ALT3
	SAI1_TX_DATA5	SAI1_TXD5	ALT0
		SAI5_RXD3	ALT1
	SAI1_TX_DATA6	SAI1_TXD6	ALT0
	SAI1_TX_DATA7	SAI1_TXD7	ALT0
	SAI1_TX_SYNC	SAI1_TXFS	ALT0
		SAI5_RXD1	ALT2
		SAI5_RXD2	ALT2
		SAI5_RXD3	ALT2
		SAI1_RXD7	ALT2
SAI2	SAI2_MCLK	SAI2_MCLK	ALT0
	SAI2_RX_BCLK	SAI2_RXC	ALT0
	SAI2_RX_DATA0	SAI2_RXD0	ALT0
	SAI2_RX_DATA1	SAI3_RXFS	ALT1
		SAI2_RXFS	ALT3
	SAI2_RX_DATA2	SAI3_RXC	ALT1
	SAI2_RX_DATA3	SAI3_RXD	ALT1
	SAI2_RX_SYNC	SAI2_RXFS	ALT0
	SAI2_TX_BCLK	SAI2_TXC	ALT0
	SAI2_TX_DATA0	SAI2_TXD0	ALT0
	SAI2_TX_DATA1	SAI3_TXFS	ALT1
		SAI2_RXD0	ALT3
		SAI2_TXFS	ALT3
	SAI2_TX_DATA2	SAI3_TXC	ALT1
	SAI2_TX_DATA3	SAI3_TXD	ALT1
	SAI2_TX_SYNC	SAI2_TXFS	ALT0
SAI3	SAI3_MCLK	SAI3_MCLK	ALT0
		NAND_DQS	ALT2
		SAI2_MCLK	ALT6
	SAI3_RX_BCLK	SAI3_RXC	ALT0
	SAI3_RX_DATA0	SAI3_RXD	ALT0
		NAND_DATA00	ALT2
	SAI3_RX_DATA1	SAI3_RXFS	ALT3
	SAI3_RX_SYNC	SAI3_RXFS	ALT0
	SAI3_TX_BCLK	SAI3_TXC	ALT0

Table continues on the next page...

Instance	Port	Pad	Mode
		NAND_ALE	ALT2
	SAI3_TX_DATA0	SAI3_TXD	ALT0
		NAND_CE0_B	ALT2
	SAI3_TX_DATA1	SAI3_TXFS	ALT3
	SAI3_TX_SYNC	SAI3_TXFS	ALT0
		NAND_DATA01	ALT2
SAI5	SAI5_MCLK	SAI5_MCLK	ALT0
		SAI2_MCLK	ALT1
		SAI3_MCLK	ALT2
	SAI5_RX_BCLK	SAI5_RXC	ALT0
		SAI3_RXC	ALT2
	SAI5_RX_DATA0	SAI5_RXD0	ALT0
		SAI3_RXD	ALT2
	SAI5_RX_DATA1	SAI5_RXD1	ALT0
		SAI3_TXFS	ALT2
	SAI5_RX_DATA2	SAI5_RXD2	ALT0
		SAI3_TXC	ALT2
	SAI5_RX_DATA3	SAI5_RXD3	ALT0
		SAI3_TXD	ALT2
	SAI5_RX_SYNC	SAI5_RXFS	ALT0
		SAI3_RXFS	ALT2
	SAI5_TX_BCLK	SAI2_RXC	ALT1
		SAI5_RXD2	ALT3
	SAI5_TX_DATA0	SAI2_RXD0	ALT1
		SAI5_RXD3	ALT3
	SAI5_TX_DATA1	SAI2_TXFS	ALT1
		SAI2_RXFS	ALT2
	SAI5_TX_DATA2	SAI2_TXC	ALT1
	SAI5_TX_DATA3	SAI2_TXD0	ALT1
	SAI5_TX_SYNC	SAI2_RXFS	ALT1
		SAI5_RXD1	ALT3
SAI6	SAI6_MCLK	SAI1_RXD7	ALT1
		SAI1_TXD7	ALT1
		ENET_TX_CTL	ALT2
	SAI6_RX_BCLK	SAI1_TXD4	ALT1
		ENET_TD0	ALT2
		SAI1_RXD4	ALT2
	SAI6_RX_DATA0	SAI1_TXD5	ALT1
		ENET_TD2	ALT2
		SAI1_RXD5	ALT2
	SAI6_RX_SYNC	SAI1_TXD6	ALT1

Table continues on the next page...

Instance	Port	Pad	Mode
		ENET_TD1	ALT2
		SAI1_RXD6	ALT2
	SAI6_TX_BCLK	SAI1_RXD4	ALT1
		ENET_TD3	ALT2
		SAI1_TXD4	ALT2
	SAI6_TX_DATA0	SAI1_RXD5	ALT1
		ENET_MDC	ALT2
		SAI1_TXD5	ALT2
	SAI6_TX_SYNC	SAI1_RXD6	ALT1
		ENET_MDIO	ALT2
		SAI1_TXD6	ALT2
SAI7	SAI7_MCLK	ENET_RD3	ALT2
		ECSPI2_MISO	ALT3
	SAI7_RX_BCLK	ENET_RD2	ALT2
		ECSPI1_MOSI	ALT3
	SAI7_RX_DATA0	ENET_RD0	ALT2
		ECSPI1_MISO	ALT3
	SAI7_RX_SYNC	ENET_RD1	ALT2
		ECSPI1_SCLK	ALT3
	SAI7_TX_BCLK	ENET_RXC	ALT2
		ECSPI2_SCLK	ALT3
	SAI7_TX_DATA0	ENET_TXC	ALT2
		ECSPI2_MOSI	ALT3
	SAI7_TX_SYNC	ENET_RX_CTL	ALT2
		ECSPI1_SS0	ALT3
SDMA1	SDMA1_EXT_EVENT0	GPIO1_IO03	ALT5
	SDMA1_EXT_EVENT1	GPIO1_IO04	ALT5
SDMA2	SDMA2_EXT_EVENT0	GPIO1_IO09	ALT5
	SDMA2_EXT_EVENT1	GPIO1_IO12	ALT5
SJC	SJC_DE_B	GPIO1_IO02	ALT7
SNVS	SNVS_ONOFF	ONOFF	ALT0
	SNVS_PMIC_ON_REQ	PMIC_ON_REQ	ALT0
	SNVS_POR_B	POR_B	ALT0
	SNVS_RTC	RTC_XTALI	ALT0
SPDIF1	SPDIF1_EXT_CLK	SPDIF_EXT_CLK	ALT0
		SAI3_TXD	ALT4
	SPDIF1_IN	SPDIF_RX	ALT0
		ENET_RD3	ALT3
		SD2_DATA3	ALT3
		SAI3_RXFS	ALT4
		SAI3_MCLK	ALT6

Table continues on the next page...

Instance	Port	Pad	Mode
	SPDIF1_OUT	SPDIF_TX	ALT0
		ENET_TX_CTL	ALT3
		SD2_DATA2	ALT3
		SAI3_MCLK	ALT4
SRC	SRC_BOOT_MODE0	BOOT_MODE0	ALT0
	SRC_BOOT_MODE1	BOOT_MODE1	ALT0
	SRC_BOOT_MODE2	BOOT_MODE2	ALT0
	SRC_BOOT_MODE3	BOOT_MODE3	ALT0
UART1	UART1_CTS_B	UART3_RXD	ALT1
		SD1_DATA1	ALT4
		SAI2_TXFS	ALT4
	UART1_RTS_B	UART3_TXD	ALT1
		SD1_DATA0	ALT4
		SAI2_RXD0	ALT4
	UART1_RX	UART1_RXD	ALT0
		SD1_CMD	ALT4
		SAI2_RXC	ALT4
	UART1_TX	UART1_TXD	ALT0
		SD1_CLK	ALT4
		SAI2_RXFS	ALT4
UART2	UART2_CTS_B	UART4_RXD	ALT1
		SD1_DATA5	ALT4
		SAI3_RXC	ALT4
	UART2_RTS_B	UART4_TXD	ALT1
		SD1_DATA4	ALT4
		SAI3_RXD	ALT4
	UART2_RX	UART2_RXD	ALT0
		SD2_DATA0	ALT3
		SD1_DATA3	ALT4
		SAI3_TXFS	ALT4
	UART2_TX	UART2_TXD	ALT0
		SD2_DATA1	ALT3
		SD1_DATA2	ALT4
		SAI3_TXC	ALT4
UART3	UART3_CTS_B	ECSPI1_MISO	ALT1
		SD1_STROBE	ALT4
	UART3_RTS_B	ECSPI1_SS0	ALT1
		SD1_RESET_B	ALT4
	UART3_RX	UART3_RXD	ALT0
		ECSPI1_SCLK	ALT1
		SD1_DATA7	ALT4

Table continues on the next page...

Instance	Port	Pad	Mode
		NAND_ALE	ALT4
	UART3_TX	UART3_TXD	ALT0
		ECSPI1_MOSI	ALT1
		SD1_DATA6	ALT4
		NAND_CE0_B	ALT4
UART4	UART4_CTS_B	ECSPI2_MISO	ALT1
		NAND_DATA02	ALT3
	UART4_RTS_B	ECSPI2_SS0	ALT1
		NAND_DATA03	ALT3
	UART4_RX	UART4_RXD	ALT0
		ECSPI2_SCLK	ALT1
		SD2_CLK	ALT3
		NAND_CLE	ALT4
		NAND_DATA00	ALT4
	UART4_TX	UART4_TXD	ALT0
		ECSPI2_MOSI	ALT1
		SD2_CMD	ALT3
		NAND_DATA01	ALT4
		NAND_RE_B	ALT4
USB1	USB1_OC	GPIO1_IO13	ALT1
	USB1_PWR	GPIO1_IO12	ALT1
	USB1_DN	USB1_D_N	No muxing
	USB1_DP	USB1_D_P	No muxing
	USB1_RESREF	USB1_TXRTUNE	No muxing
	USB1_RX_N	USB1_RX_N	No muxing
	USB1_RX_P	USB1_RX_P	No muxing
	USB1_TX_N	USB1_TX_N	No muxing
	USB1_TX_P	USB1_TX_P	No muxing
	USB1_VBUS	USB1_VBUS	No muxing
USB2	USB2_OC	GPIO1_IO15	ALT1
	USB2_PWR	GPIO1_IO14	ALT1
	USB2_DN	USB2_D_N	No muxing
	USB2_DP	USB2_D_P	No muxing
	USB2_RESREF	USB2_TXRTUNE	No muxing
	USB2_RX_N	USB2_RX_N	No muxing
	USB2_RX_P	USB2_RX_P	No muxing
	USB2_TX_N	USB2_TX_N	No muxing
	USB2_TX_P	USB2_TX_P	No muxing
	USB2_VBUS	USB2_VBUS	No muxing
USDHC1	USDHC1_CD_B	GPIO1_IO06	ALT5
	USDHC1_CLK	SD1_CLK	ALT0

Table continues on the next page...

Instance	Port	Pad	Mode
	USDHC1_CMD	SD1_CMD	ALT0
	USDHC1_DATA0	SD1_DATA0	ALT0
	USDHC1_DATA1	SD1_DATA1	ALT0
	USDHC1_DATA2	SD1_DATA2	ALT0
	USDHC1_DATA3	SD1_DATA3	ALT0
	USDHC1_DATA4	SD1_DATA4	ALT0
	USDHC1_DATA5	SD1_DATA5	ALT0
	USDHC1_DATA6	SD1_DATA6	ALT0
	USDHC1_DATA7	SD1_DATA7	ALT0
	USDHC1_RESET_B	SD1_RESET_B	ALT0
	USDHC1_STROBE	SD1_STROBE	ALT0
	USDHC1_VSELECT	GPIO1_IO03	ALT1
	USDHC1_WP	GPIO1_IO07	ALT5
USDHC2	USDHC2_CD_B	SD2_CD_B	ALT0
	USDHC2_CLK	SD2_CLK	ALT0
	USDHC2_CMD	SD2_CMD	ALT0
	USDHC2_DATA0	SD2_DATA0	ALT0
	USDHC2_DATA1	SD2_DATA1	ALT0
	USDHC2_DATA2	SD2_DATA2	ALT0
	USDHC2_DATA3	SD2_DATA3	ALT0
	USDHC2_RESET_B	SD2_RESET_B	ALT0
		GPIO1_IO08	ALT5
	USDHC2_VSELECT	GPIO1_IO04	ALT1
	USDHC2_WP	SD2_WP	ALT0
USDHC3	USDHC3_CD_B	NAND_DATA02	ALT2
		I2C2_SCL	ALT2
		GPIO1_IO14	ALT4
		ENET_TD1	ALT6
	USDHC3_CLK	NAND_WE_B	ALT2
		ENET_RD2	ALT6
	USDHC3_CMD	NAND_WP_B	ALT2
		ENET_RD3	ALT6
	USDHC3_DATA0	NAND_DATA04	ALT2
		ENET_TX_CTL	ALT6
	USDHC3_DATA1	NAND_DATA05	ALT2
		ENET_TXC	ALT6
	USDHC3_DATA2	NAND_DATA06	ALT2
		ENET_RX_CTL	ALT6
	USDHC3_DATA3	NAND_DATA07	ALT2
		ENET_RXC	ALT6
	USDHC3_DATA4	NAND_RE_B	ALT2

Table continues on the next page...

Instance	Port	Pad	Mode
		ENET_RD0	ALT6
	USDHC3_DATA5	NAND_CE2_B	ALT2
		ENET_MDIO	ALT6
	USDHC3_DATA6	NAND_CE3_B	ALT2
		ENET_TD3	ALT6
	USDHC3_DATA7	NAND_CLE	ALT2
		ENET_TD2	ALT6
	USDHC3_RESET_B	NAND_READY_B	ALT2
		UART3_RXD	ALT2
		GPIO1_IO09	ALT4
		ENET_RD1	ALT6
	USDHC3_STROBE	NAND_CE1_B	ALT2
		ENET_MDC	ALT6
	USDHC3_VSELECT	UART3_TXD	ALT2
		GPIO1_IO11	ALT4
	USDHC3_WP	NAND_DATA03	ALT2
		I2C2_SDA	ALT2
		GPIO1_IO15	ALT4
		ENET_TD0	ALT6
WDOG1	WDOG1_WDOG_ANY	GPIO1_IO02	ALT5
	WDOG1_WDOG_B	GPIO1_IO02	ALT1
XTALOSC	XTALI_24M	XTALI_24M	No Muxing
	XTALO_24M	XTALO_24M	No Muxing

8.2 IOMUX Controller (IOMUXC)

8.2.1 Overview

The IOMUX Controller (IOMUXC), together with the IOMUX, enables the IC to share one pad to several functional blocks. This sharing is done by multiplexing the pad's input and output signals.

Every module requires a specific pad setting (such as pull up or keeper), and for each pad, there are up to 8 muxing options (called ALT modes). The pad settings parameters are controlled by the IOMUXC.

The IOMUX consists only of combinatorial logic combined from several basic IOMUX cells. Each basic IOMUX cell handles only one pad signal's muxing.

Figure 8-1 illustrates the IOMUX/IOMUXC connectivity in the system.

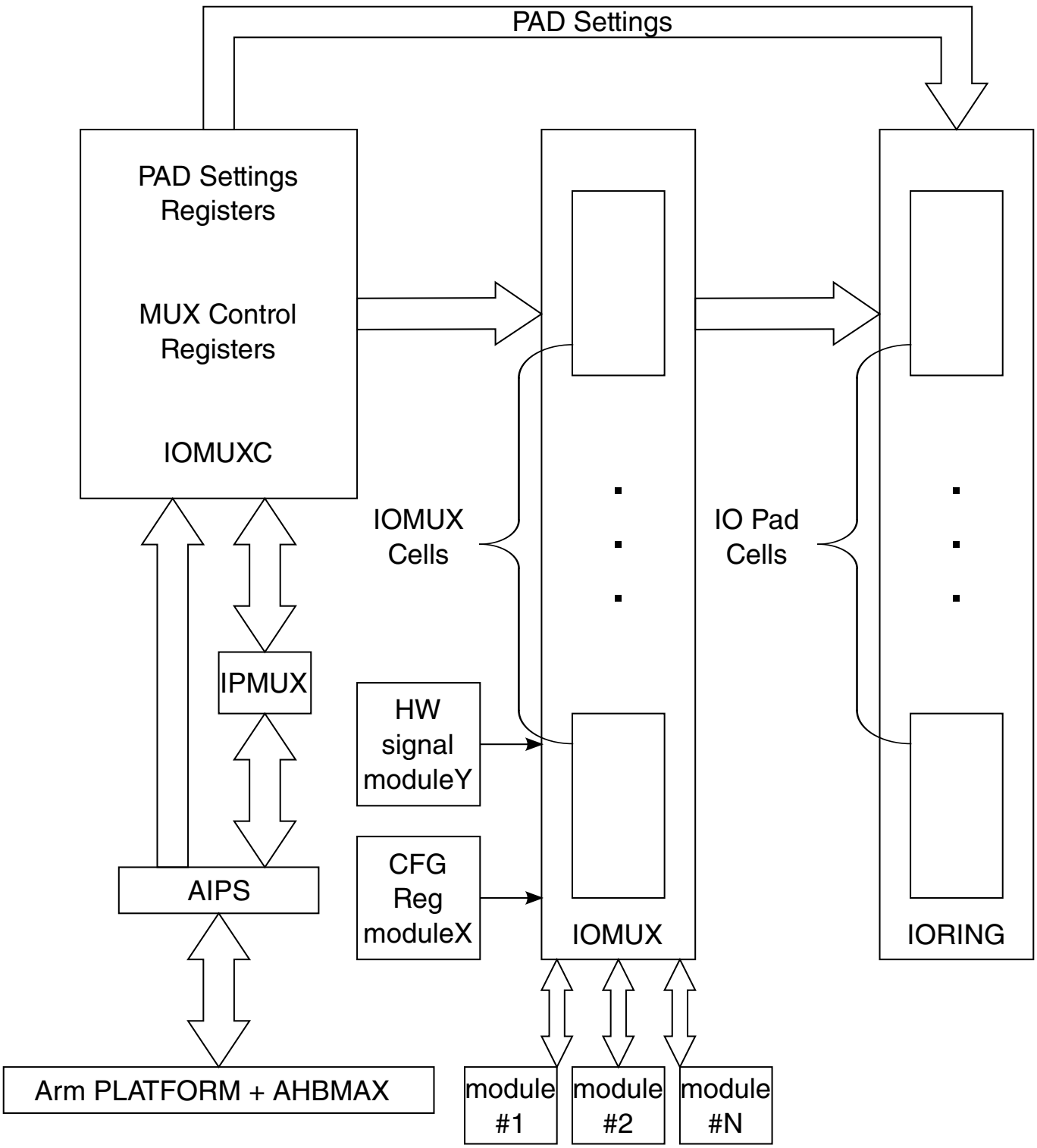


Figure 8-1. IOMUX SoC Level Block Diagram

8.2.1.1 Block Diagram

The high level illustration of the IO cells is shown in Figure 8-3

8.2.1.2 Features

The IOMUXC features include:

- 32-bit software mux control registers (IOMUXC_SW_MUX_CTL_PAD_<PAD NAME> or IOMUXC_SW_MUX_CTL_GRP_<GROUP NAME>) to configure 1 of 8 alternate (ALT) MUX_MODE fields of each pad or a predefined group of pads and to enable the forcing of an input path of the pad(s) (SION bit).
- 32-bit software pad control registers (IOMUXC_SW_PAD_CTL_PAD_<PAD_NAME> or IOMUXC_SW_PAD_CTL_GRP_<GROUP NAME>) to configure specific pad settings of each pad, or a predefined group of pads.
- 32-bit general purpose registers - several (GPR0 to GPR*n*) 32-bit registers according to SoC requirements for any usage.
- 32-bit input select control registers to control the input path to a module when more than one pad drives this module input.

Each SW MUX/PAD CTL IOMUXC register handles only one pad or one pad's group.

Only the minimum number of registers required by software are implemented by hardware. For example, if only ALT0 and ALT1 modes are used on Pad x then only one bit register will be generated as the MUX_MODE control field in the software mux control register of Pad x.

The software mux control registers may allow the forcing of pads to become input (input path enabled) regardless of the functional direction driven. This may be useful for loopback and GPIO data capture.

8.2.2 Functional description

This section provides a complete functional description of the block.

The IOMUXC consists of two sub-blocks:

- IOMUXC_REGISTERS includes all of the IOMUXC registers (see Features).
- IOMUXC_LOGIC includes all of the IOMUXC combinatorial logic (IP interface controls, address decoder, observability muxes).

The IOMUX consists of a number (about the number of pads in the SoC) of basic iomux_cell units. If only one functional mode is required for a specific pad, there is no need for IOMUX and the signals can be connected directly from the module to the I/O. The IOMUX cell is required whenever two or more functional modes are required for a specific pad or when one functional mode and the one test mode are required.

The basic iomux_cell design, which allows two levels of HW signal control (in ALT6 and ALT7 modes - ALT7 gets highest priority) is shown in Figure 8-2.

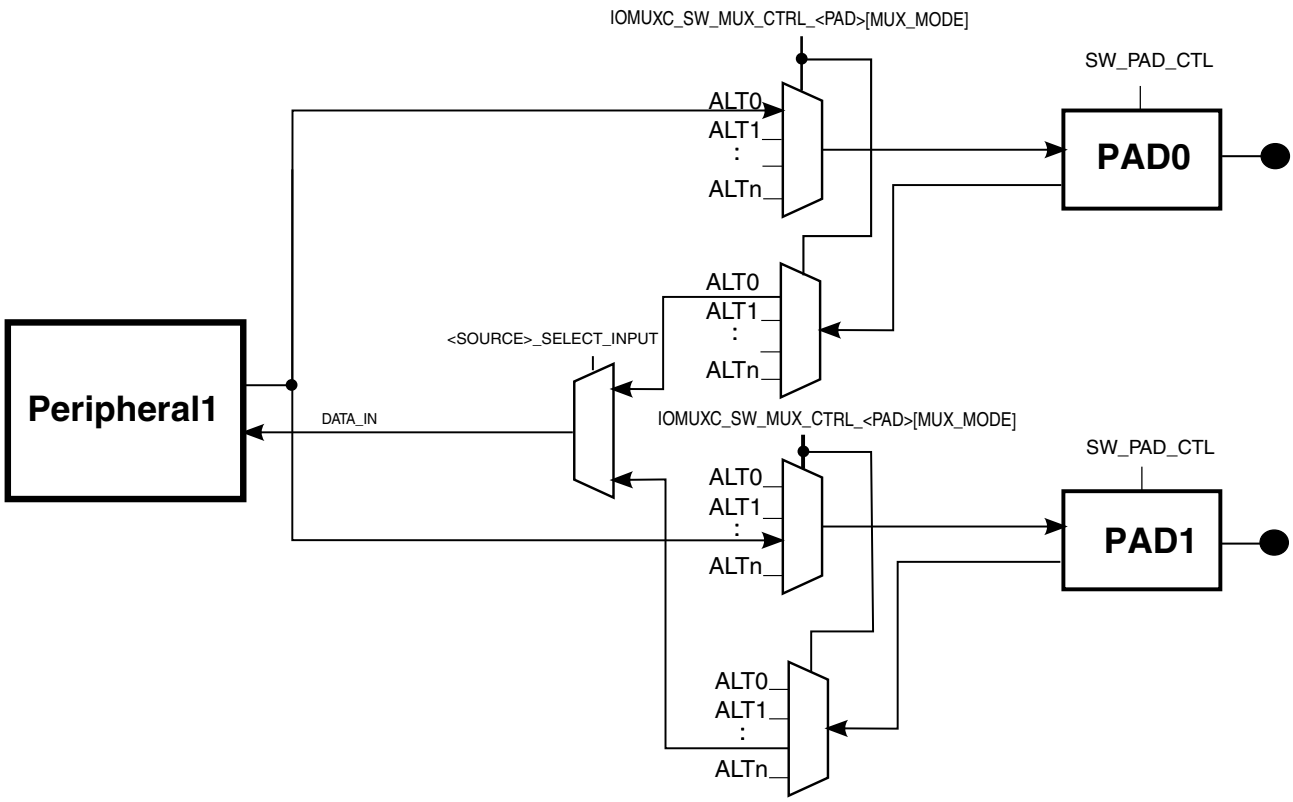


Figure 8-2. IOMUX Cell Block Diagram

8.2.2.1 GPIO pad features

The GPIO pad includes the following features:

- Wide-range voltage interface
  - 1.8V ~ 3.3V I/O interface
- CMOS input / Schmitt trigger Input
- 3-state and open-drain output
- Two slew rate control levels
- Programmable feature support
  - Controllable input enable

- Controllable CMOS/Schmitt trigger input
- Controllable pull-up/pull-down resistor
- Controllable output drive strength (x1 / x2 / x4 / x6)
- Controllable slew rate control (slow slew / fast slew)

8.2.2.1.1 Pull up/Pull down control

Pull up/Pull down function is controlled by the PE and PS pin.

Table 8-1. Pull up / Pull down control truth table

Mode	State	
	PE	PS
Disable	0	X
Pull-down enable	1	0
Pull-up enable	1	1

8.2.2.1.2 Input control

The IS pin selects CMOS and Schmitt trigger.

Table 8-2. Input control truth table

Mode	State	
	IE	IS
Disable	0	X
CMOS input	1	0
Schmitt trigger input	1	1

8.2.2.1.3 Output driver control

Output drive strength is controlled by the DS0, DS1 pin.

Table 8-3. Drive strength control truth table

State		Driver Strength
DS1	DS0	
0	0	X1
1	0	X2
0	1	X4
1	1	X6

The SR pin controls slew-rate of output driver.

Table 8-4. Slew-rate of output driver

State	Driver Slew
SR	
0	Fast Slew
1	Slow Slew

8.2.2.2 ALT6 and ALT7 extended muxing modes

The ALT7 and ALT6 extended muxing modes allow any signal in the system (such as fuse, pad input, JTAG, or software register) to override any software configuration and to force the ALT6/ALT7 muxing mode.

It also allows an IOMUX software register to control a group of pads.

8.2.2.3 SW Loopback through SION bit

A limited option exists to override the default pad functionality and force the input path to be active (ipp_ibe==1'b1) regardless of the value driven by the corresponding module. This can be done by setting the SION (Software Input On) bit in the IOMUXC_SW_MUX_CTL register (when available) to "1".

Uses include:

- LoopBack - Module x drives the pad and also receives pad value as an input.
- GPIO Capture - Module x drives the pad and the value is captured by GPIO.

8.2.2.4 Daisy chain - multi pads driving same module input pin

In some cases, more than one pad may drive a single module input pin. Such cases require the addition of one more level of IOMUXing; all of these input signals are muxed, and a dedicated software controlled register controls the mux in order to select the required input path.

A module port involved in "daisy chain" requires two software configuration commands, one for selecting the mode for this pad (programable via the IOMUXC_SW_MUX_CTL_<PAD> registers) and one for defining it as the input path (via the daisy chain registers).

This means that a module port involved in "daisy chain" requires two software configuration commands, one for selecting the mode for this pad (programable via the IOMUXC_SW_MUX_CTL_<PAD> registers) and one for defining it as the input path (via the daisy chain registers). The daisy chain is illustrated in the figure below.

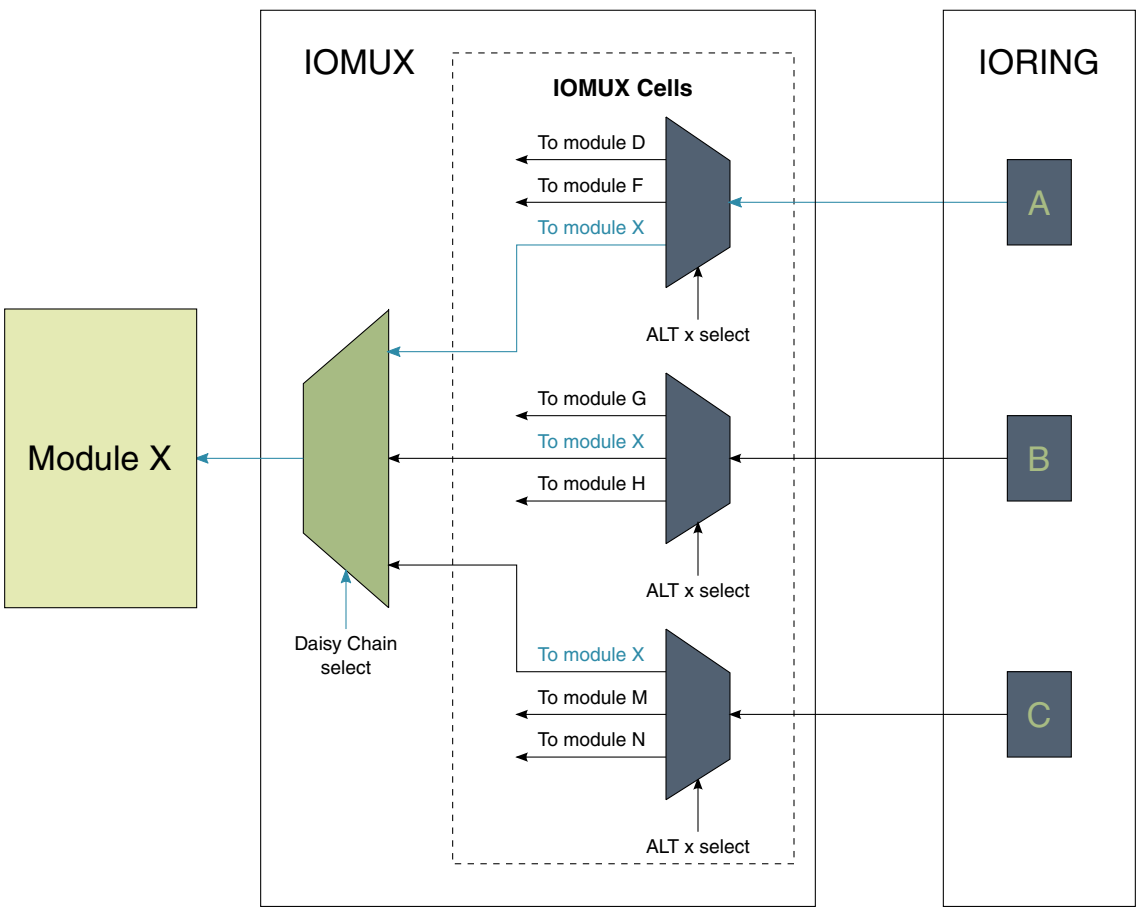


Figure 8-3. Daisy chain illustration

8.2.2.5 Clocks

The table found here describes the clock sources for IOMUXC.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 8-5. IOMUXC Clocks

Clock name	Clock Root	Description
ipg_clk_s	ipg_clk_root	Peripheral access clock

8.2.3 IOMUXC GPR Memory Map/Register Definition

IOMUXC_GPR memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3034_0000	General Purpose Register 0 (IOMUXC_GPR_GPR0)	32	R/W	0000_0000h	8.2.3.1/1366
3034_0004	General Purpose Register 1 (IOMUXC_GPR_GPR1)	32	R/W	0001_0000h	8.2.3.2/1367
3034_0008	General Purpose Register 2 (IOMUXC_GPR_GPR2)	32	R/W	0000_0000h	8.2.3.3/1369
3034_000C	General Purpose Register 3 (IOMUXC_GPR_GPR3)	32	R/W	0000_0000h	8.2.3.4/1369
3034_0010	General Purpose Register 4 (IOMUXC_GPR_GPR4)	32	R/W	0000_0000h	8.2.3.5/1370
3034_0014	General Purpose Register 5 (IOMUXC_GPR_GPR5)	32	R/W	0000_0000h	8.2.3.6/1373
3034_0018	General Purpose Register 6 (IOMUXC_GPR_GPR6)	32	R/W	0000_0000h	8.2.3.7/1374
3034_001C	General Purpose Register 7 (IOMUXC_GPR_GPR7)	32	R/W	0000_0000h	8.2.3.8/1374
3034_0020	General Purpose Register 8 (IOMUXC_GPR_GPR8)	32	R/W	0000_0000h	8.2.3.9/1375
3034_0024	General Purpose Register 9 (IOMUXC_GPR_GPR9)	32	R/W	0000_0000h	8.2.3.10/ 1375
3034_0028	General Purpose Register 10 (IOMUXC_GPR_GPR10)	32	R/W	0000_0008h	8.2.3.11/ 1376
3034_002C	General Purpose Register 11 (IOMUXC_GPR_GPR11)	32	R/W	0000_0200h	8.2.3.12/ 1377
3034_0030	General Purpose Register 12 (IOMUXC_GPR_GPR12)	32	R/W	0000_4000h	8.2.3.13/ 1379
3034_0034	General Purpose Register 13 (IOMUXC_GPR_GPR13)	32	R/W	0000_0000h	8.2.3.14/ 1380
3034_0038	General Purpose Register 14 (IOMUXC_GPR_GPR14)	32	R/W	0349_4000h	8.2.3.15/ 1382
3034_003C	General Purpose Register 15 (IOMUXC_GPR_GPR15)	32	R/W	0000_0000h	8.2.3.16/ 1383
3034_0040	General Purpose Register 16 (IOMUXC_GPR_GPR16)	32	R/W	0000_0000h	8.2.3.17/ 1384

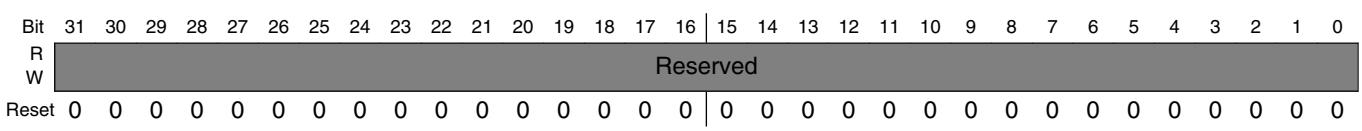
Table continues on the next page...

IOMUXC_GPR memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3034_0044	General Purpose Register 17 (IOMUXC_GPR_GPR17)	32	R/W	0000_0000h	8.2.3.18/ 1385
3034_0048	General Purpose Register 18 (IOMUXC_GPR_GPR18)	32	R/W	0000_0000h	8.2.3.19/ 1385
3034_004C	General Purpose Register 19 (IOMUXC_GPR_GPR19)	32	R	See section	8.2.3.20/ 1385
3034_0050	General Purpose Register 20 (IOMUXC_GPR_GPR20)	32	R/W	0000_0000h	8.2.3.21/ 1386
3034_0054	General Purpose Register 21 (IOMUXC_GPR_GPR21)	32	R/W	See section	8.2.3.22/ 1389
3034_0058	General Purpose Register 22 (IOMUXC_GPR_GPR22)	32	R/W	0000_0001h	8.2.3.23/ 1391
3034_005C	General Purpose Register 23 (IOMUXC_GPR_GPR23)	32	R/W	0000_0000h	8.2.3.24/ 1392
3034_0060	General Purpose Register 24 (IOMUXC_GPR_GPR24)	32	R	See section	8.2.3.25/ 1392

8.2.3.1 General Purpose Register 0 (IOMUXC_GPR_GPR0)

Address: 3034_0000h base + 0h offset = 3034_0000h

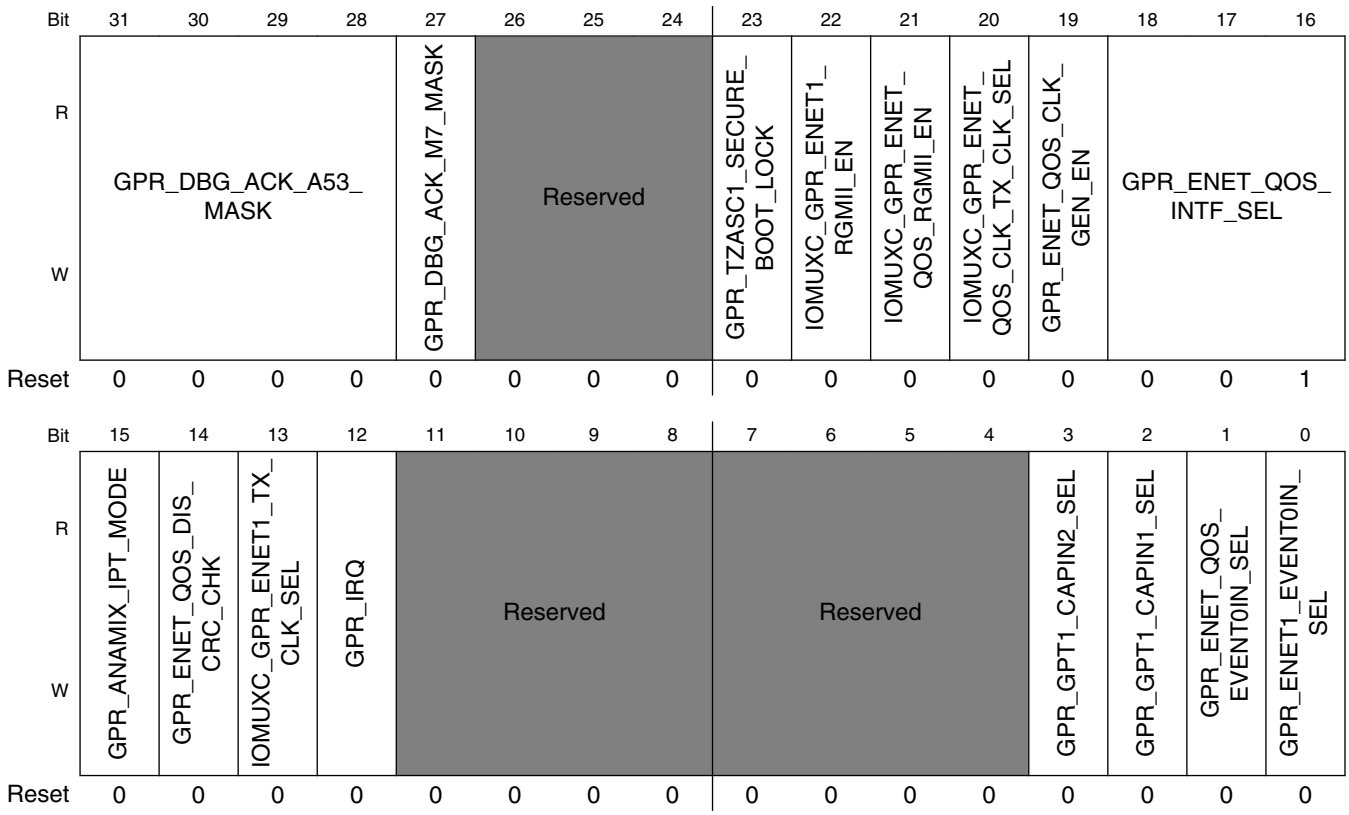


IOMUXC_GPR_GPR0 field descriptions

Field	Description
-	This field is reserved.

8.2.3.2 General Purpose Register 1 (IOMUXC_GPR_GPR1)

Address: 3034_0000h base + 4h offset = 3034_0004h



IOMUXC_GPR_GPR1 field descriptions

Field	Description
31–28 GPR_DBG_ACK_A53_MASK	Mask debug ack from each CA53 core 0 unmasked 1 mask to 0
27 GPR_DBG_ACK_M7_MASK	Mask debug ack from each CM7 0 unmasked 1 mask to 0
26–24 -	This field is reserved.
23 GPR_TZASC1_SECURE_BOOT_LOCK	secure_boot_lock for TZASC
22 IOMUXC_GPR_ENET1_RGMII_EN	ENET1 TX clock direction select for RGMII or MII 0 MII (input) 1 RGMII (output)

Table continues on the next page...

IOMUXC_GPR_GPR1 field descriptions (continued)

Field	Description
21 IOMUXC_GPR_ENET_QOS_RGMII_EN	ENET QOS TX clock direction select for RGMII or MII 0 MII(input) 1 RGMII(output)
20 IOMUXC_GPR_ENET_QOS_CLK_TX_CLK_SEL	SOI bit for the pad (iomuxc_sw_input_on_pad_enet_td2) should also be set. 1 ENET QOS RMII clock comes from ccm->pad->loopback 0 ENET QOS RMII clock comes from external PHY or OSC
19 GPR_ENET_QOS_CLK_GEN_EN	Enable clk generate module for ENET QoS 0 disable 1 enable
18–16 GPR_ENET_QOS_INTF_SEL	Select ENET QOS working mode 0 MII 1 RGMII 4 RMII
15 GPR_ANAMIX_IPT_MODE	Mask ANAMIX ipt_mode 0 masked to 0 1 unmasked
14 GPR_ENET_QOS_DIS_CRC_CHK	Disable CRC check feature 0 do not disable 1 disable
13 IOMUXC_GPR_ENET1_TX_CLK_SEL	SOI bit for the pad (iomuxc_sw_input_on_pad_enet_td2) should also be set 1 ENET1 RMII clock comes from ccm->pad->loopback 0 ENET1 RMII clock comes from external PHY or OSC
12 GPR_IRQ	Generate IRQ on IRQ0
11–8 -	This field is reserved.
7–4 -	This field is reserved.
3 GPR_GPT1_CAPIN2_SEL	Selector for GPT1 Capture in Channel 2 0 IOMUX 1 ENET QOS TIMIER1 EVENT
2 GPR_GPT1_CAPIN1_SEL	Selector for GPT1 Capture in Channel 1 0 IOMUX 1 ENET1 TIMIER1 EVENT
1 GPR_ENET_QOS_EVENT0IN_SEL	Selector for ENET QoS EVENT0 IN 0 IOMUX; 1 GPT1 CMPOUT2

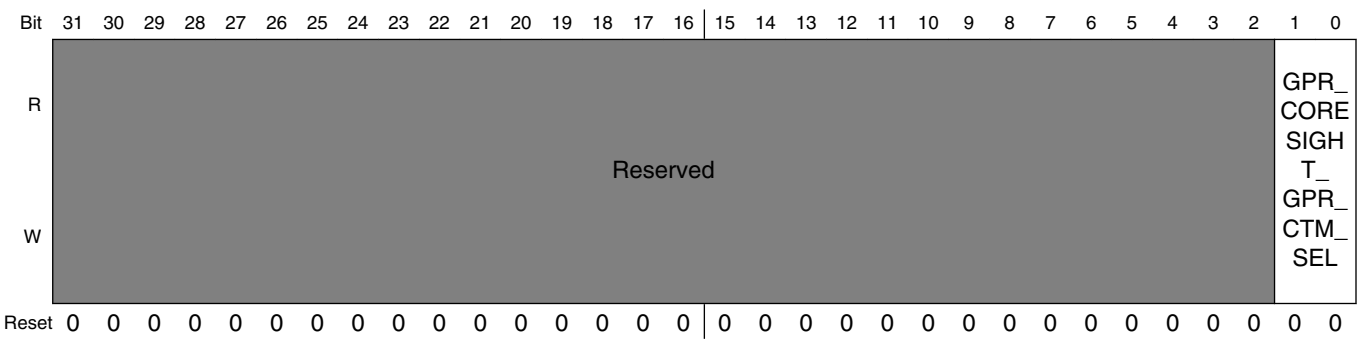
Table continues on the next page...

IOMUXC_GPR_GPR1 field descriptions (continued)

Field	Description
0 GPR_ENET1_ EVENT0IN_SEL	Selector for ENET1 EVENT0 IN 0 IOMUX 1 GPT1 CMPOUT2

8.2.3.3 General Purpose Register 2 (IOMUXC_GPR_GPR2)

Address: 3034_0000h base + 8h offset = 3034_0008h

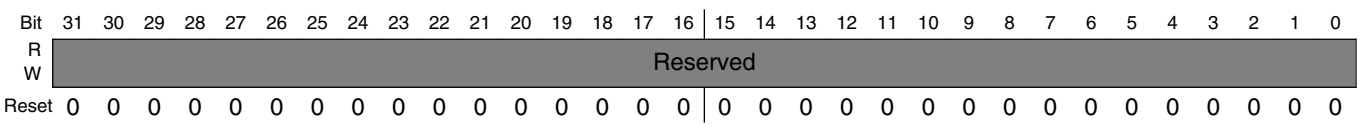


IOMUXC_GPR_GPR2 field descriptions

Field	Description
31–2 -	This field is reserved.
GPR_ CORESIGHT_ GPR_CTM_SEL	Select for Coresight master

8.2.3.4 General Purpose Register 3 (IOMUXC_GPR_GPR3)

Address: 3034_0000h base + Ch offset = 3034_000Ch

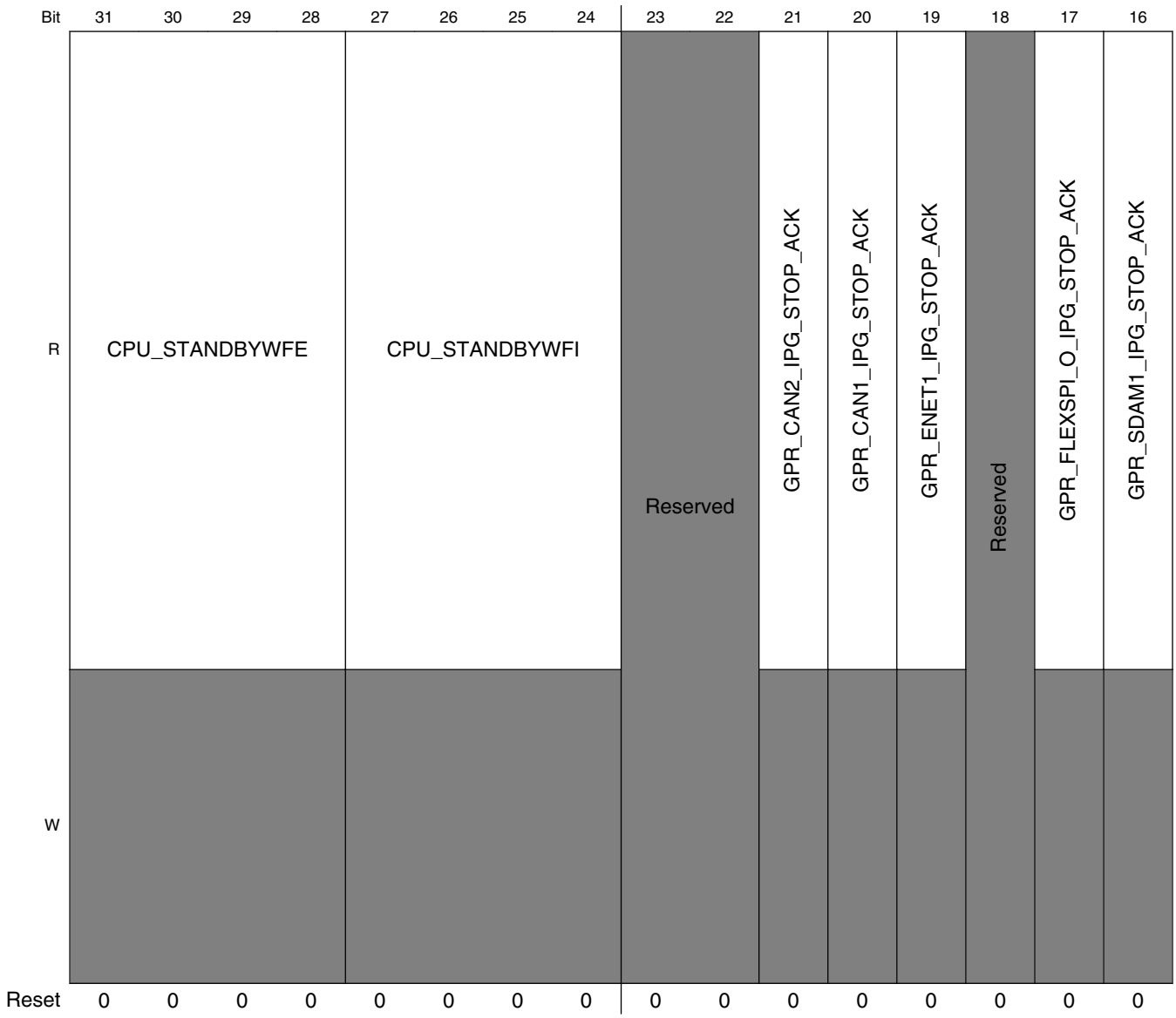


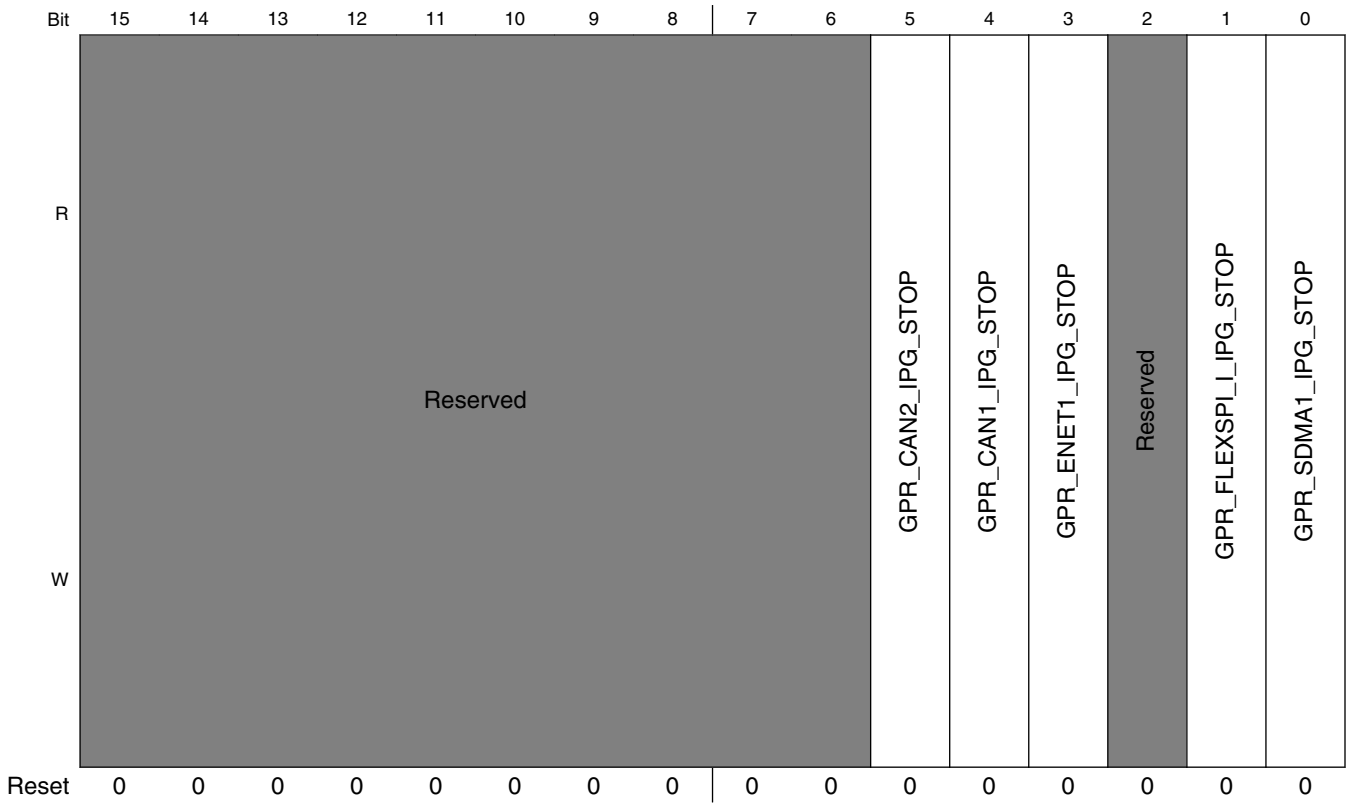
IOMUXC_GPR_GPR3 field descriptions

Field	Description
-	This field is reserved.

8.2.3.5 General Purpose Register 4 (IOMUXC_GPR_GPR4)

Address: 3034_0000h base + 10h offset = 3034_0010h





IOMUXC_GPR_GPR4 field descriptions

Field	Description
31–28 CPU_ STANDBYWFE	From CA53. Status of CPU STANDBYWFE low power states. MSB: status of core 3 STANDBYWFE low power state LSB: status of core 0 STANDBYWFE low power state 1 WFE
27–24 CPU_ STANDBYWFI	From CA53. Status of CPU STANDBYWFI low power states. MSB: status of core 3 STANDBYWFI low power state LSB: status of core 0 STANDBYWFI low power state 1 WFI
23–22 -	This field is reserved.
21 GPR_CAN2_ IPG_STOP_ACK	CAN2 stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode
20 GPR_CAN1_ IPG_STOP_ACK	CAN1 stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode

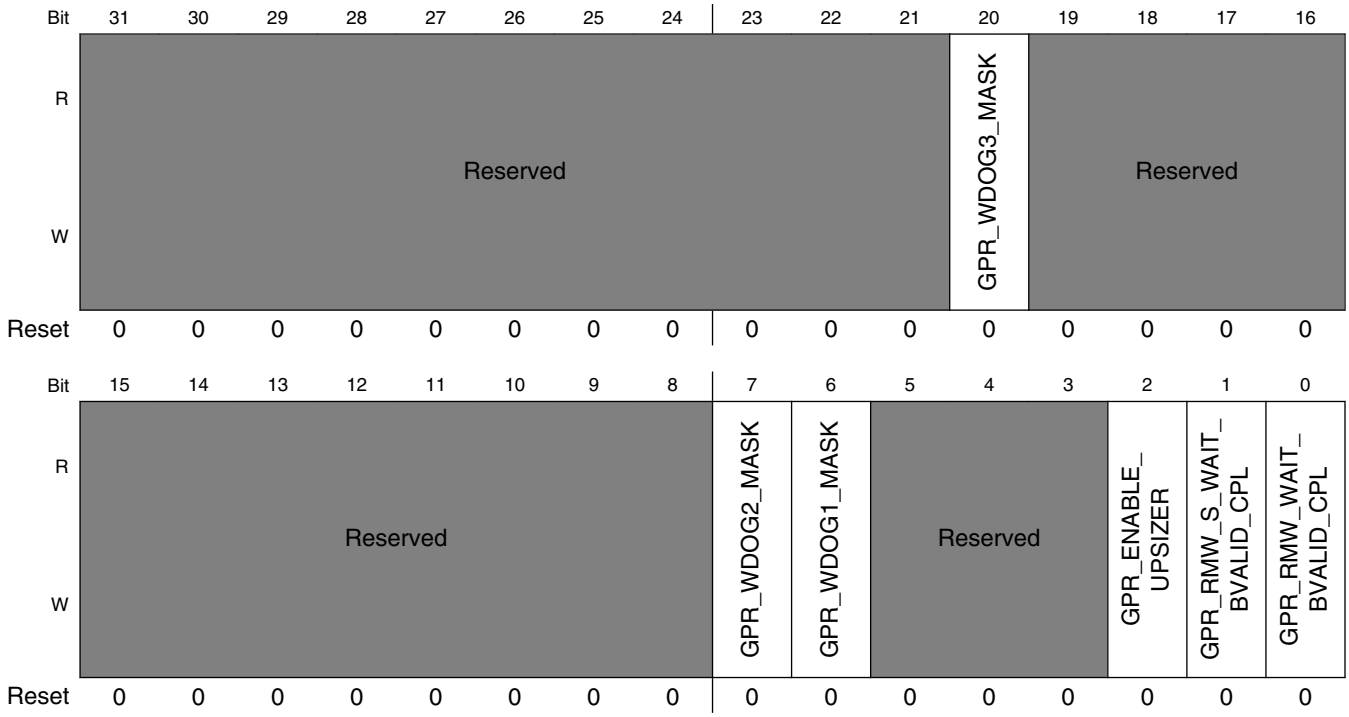
Table continues on the next page...

IOMUXC_GPR_GPR4 field descriptions (continued)

Field	Description
19 GPR_ENET1_ IPG_STOP_ACK	ENET1 stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode
18 -	This field is reserved.
17 GPR_FLEXSPI_ O_IPG_STOP_ ACK	FLEXSPI stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode
16 GPR_SDAM1_ IPG_STOP_ACK	SDMA1 stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode
15–6 -	This field is reserved.
5 GPR_CAN2_ IPG_STOP	CAN2 stop request 0 stop request off 1 stop request on
4 GPR_CAN1_ IPG_STOP	CAN1 stop request 0 stop request off 1 stop request on
3 GPR_ENET1_ IPG_STOP	ENET1 stop request 0 stop request off 1 stop request on
2 -	This field is reserved.
1 GPR_FLEXSPI_ I_IPG_STOP	FLEXSPI stop request 0 stop request off 1 stop request on
0 GPR_SDMA1_ IPG_STOP	SDMA1 stop request 0 stop request off 1 stop request on

8.2.3.6 General Purpose Register 5 (IOMUXC_GPR_GPR5)

Address: 3034_0000h base + 14h offset = 3034_0014h



IOMUXC_GPR_GPR5 field descriptions

Field	Description
31–21 -	This field is reserved.
20 GPR_WDOG3_MASK	This bit is only used to mask the internal WDOG3 int signal for output of GPIO1_IO02.ALT5_OUT, which is combined with WDOG1/2/3. 0 WDOG3 low will make the GPIO1_IO02.ALT5_OUT low 1 WDOG3 low will NOT impact the GPIO1_IO02.ALT5_OUT
19–8 -	This field is reserved.
7 GPR_WDOG2_MASK	This bit is only used to mask the internal WDOG2 int signal for output of GPIO1_IO02.ALT5_OUT, which is combined with WDOG1/2/3. 0 WDOG2 low will make the GPIO1_IO02.ALT5_OUT low 1 WDOG2 low will NOT impact the GPIO1_IO02.ALT5_OUT
6 GPR_WDOG1_MASK	Normally, WDOG1 output is in GPIO1_IO02.ALT1_OUT. This bit is only used to mask the internal WDOG1 int signal for output of GPIO1_IO02.ALT5_OUT, which is combined with WDOG1/2/3. 0 WDOG1 low will make the GPIO1_IO02.ALT5_OUT low 1 WDOG1 low will NOT impact the GPIO1_IO02.ALT5_OUT
5–3 -	This field is reserved.

Table continues on the next page...

IOMUXC_GPR_GPR5 field descriptions (continued)

Field	Description
2 GPR_ENABLE_UPSIZER	Enable to upsize 32=bit SDMA1 burst transaction to 64-bit
1 GPR_RMW_S_WAIT_BVALID_CPL	If this bit set to 1, RMW will write back the next data after bvalid_s is 1
0 GPR_RMW_WAIT_BVALID_CPL	If this bit set to 1, RMW will write back the next data after bvalid_s is 1

8.2.3.7 General Purpose Register 6 (IOMUXC_GPR_GPR6)

Address: 3034_0000h base + 18h offset = 3034_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR_M7_INITVTOR																Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR6 field descriptions

Field	Description
31–7 GPR_M7_INITVTOR	Decides where CM7 boots up out of reset
-	This field is reserved.

8.2.3.8 General Purpose Register 7 (IOMUXC_GPR_GPR7)

Address: 3034_0000h base + 1Ch offset = 3034_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR7 field descriptions

Field	Description
-	This field is reserved.

8.2.3.9 General Purpose Register 8 (IOMUXC_GPR_GPR8)

Address: 3034_0000h base + 20h offset = 3034_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR8 field descriptions

Field	Description
-	This field is reserved.

8.2.3.10 General Purpose Register 9 (IOMUXC_GPR_GPR9)

Address: 3034_0000h base + 24h offset = 3034_0024h

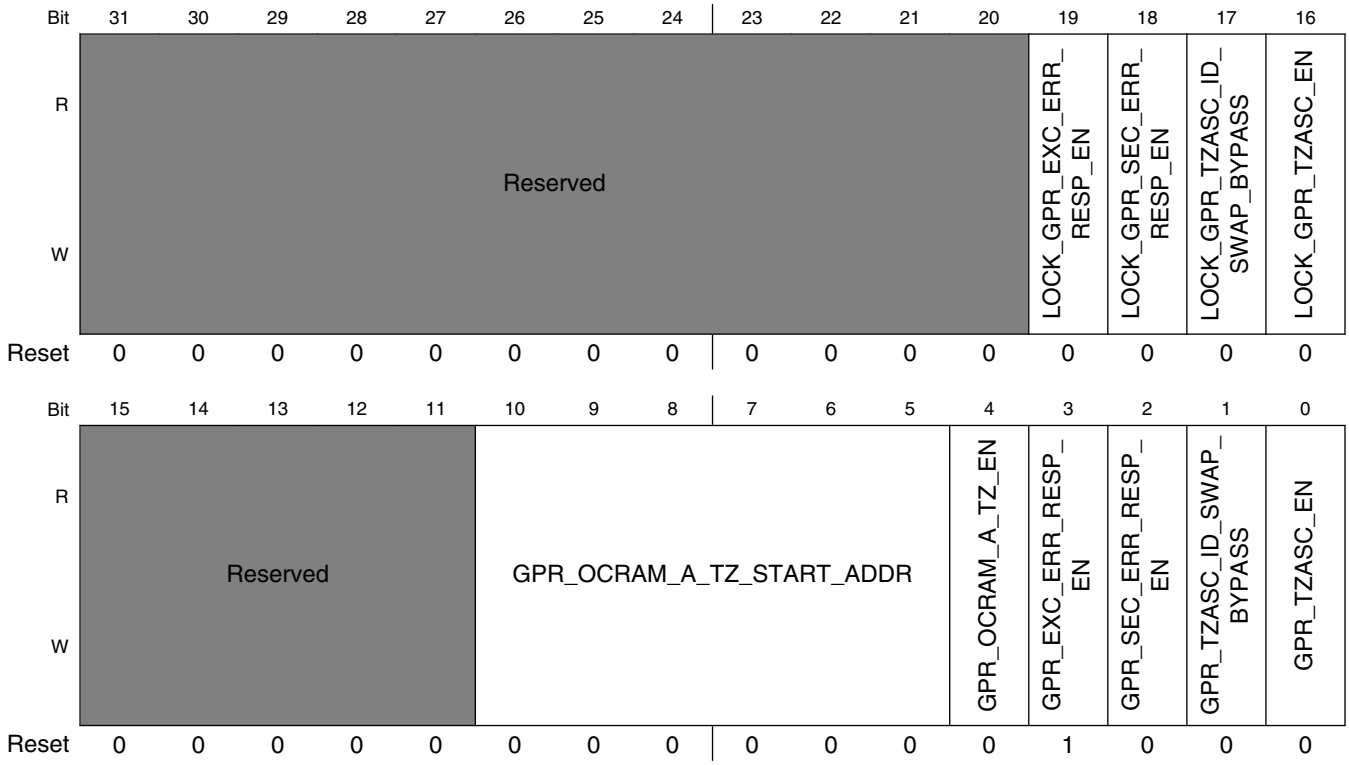
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR9 field descriptions

Field	Description
-	This field is reserved.

8.2.3.11 General Purpose Register 10 (IOMUXC_GPR_GPR10)

Address: 3034_0000h base + 28h offset = 3034_0028h



IOMUXC_GPR_GPR10 field descriptions

Field	Description
31–20 -	This field is reserved.
19 LOCK_GPR_EXC_ERR_RESP_EN	Lock bit for GPR_EXC_ERR_RESP_EN
18 LOCK_GPR_SEC_ERR_RESP_EN	Lock bit for GPR_SEC_ERR_RESP_EN
17 LOCK_GPR_TZASC_ID_SWAP_BYPASS	Lock bit for GPR_TZASC_ID_SWAP_BYPASS
16 LOCK_GPR_TZASC_EN	Lock bit for GPR_TZASC_EN
15–11 -	This field is reserved.

Table continues on the next page...

IOMUXC_GPR_GPR10 field descriptions (continued)

Field	Description
10–5 GPR_OCRAM_A_TZ_START_ADDR	OCRAM Audio Trustzone start address
4 GPR_OCRAM_A_TZ_EN	OCRAM Audio Trustzone Enable
3 GPR_EXC_ERR_RESP_EN	mem security gasket exclusive error response enable
2 GPR_SEC_ERR_RESP_EN	mem security gasket security error response enable
1 GPR_TZASC_ID_SWAP_BYPASS	Connect to id_swap_bypass input on tzasc_id_wrap.
0 GPR_TZASC_EN	Connect to tzasc_en input on tzasc_id_wrap.

8.2.3.12 General Purpose Register 11 (IOMUXC_GPR_GPR11)

Address: 3034_0000h base + 2Ch offset = 3034_002Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		LOCK_GPR_OCRAM_S_TZ_START_ADDR			LOCK_GPR_OCRAM_S_TZ_EN	LOCK_GPR_CAAM_CAAM_IPS_MANAGER	LOCK_GPR_OCRAM_TZ_START_ADDR								LOCK_GPR_OCRAM_TZ_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

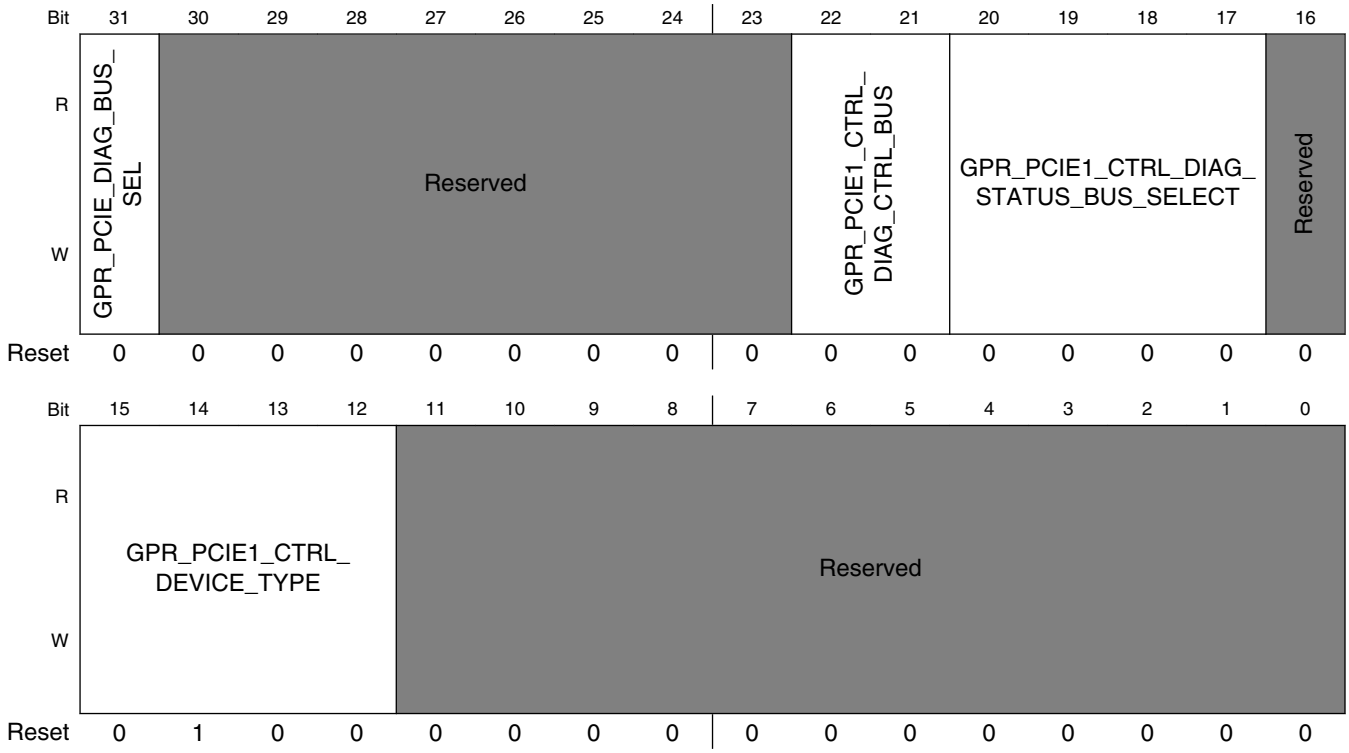
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	GPR_OCRAM_S_TZ_START_ADDR				GPR_OCRAM_S_TZ_EN	GPR_CAAM_CAAM_IPS_MANAGER	GPR_OCRAM_TZ_START_ADDR								GPR_OCRAM_TZ_EN
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR11 field descriptions

Field	Description
31–30 -	This field is reserved.
29–27 LOCK_GPR_ OCRAM_S_TZ_ START_ADDR	Lock bit for GPR_OCRAM_S_TZ_START_ADDR[15:12]
26 LOCK_GPR_ OCRAM_S_TZ_ EN	Lock bit for GPR_OCRAM_S_TZ_EN
25 LOCK_GPR_ CAAM_CAAM_ IPS_MANAGER	Lock bit for GPR_CAAM_CAAM_IPS_MANAGER[4]
24–17 LOCK_GPR_ OCRAM_TZ_ START_ADDR	Lock bit for GPR_OCRAM_TZ_START_ADDR[19:12]
16 LOCK_GPR_ OCRAM_TZ_EN	Lock bit for GPR_OCRAM_TZ_EN
15 -	This field is reserved.
14–11 GPR_OCRAM_ S_TZ_START_ ADDR	OCRAM_S Trustzone start address
10 GPR_OCRAM_ S_TZ_EN	OCRAM_S Trustzone Enable
9 GPR_CAAM_ CAAM_IPS_ MANAGER	Used to control whether CAAM manager page and Job ring registers are controlled by CSU/RDC slot 0 not controlled by CSU/RDC slot 1 controlled by CSU/RDC slot
8–1 GPR_OCRAM_ TZ_START_ ADDR	OCRAM Trustzone start address
0 GPR_OCRAM_ TZ_EN	OCRAM Trustzone Enable

8.2.3.13 General Purpose Register 12 (IOMUXC_GPR_GPR12)

Address: 3034_0000h base + 30h offset = 3034_0030h

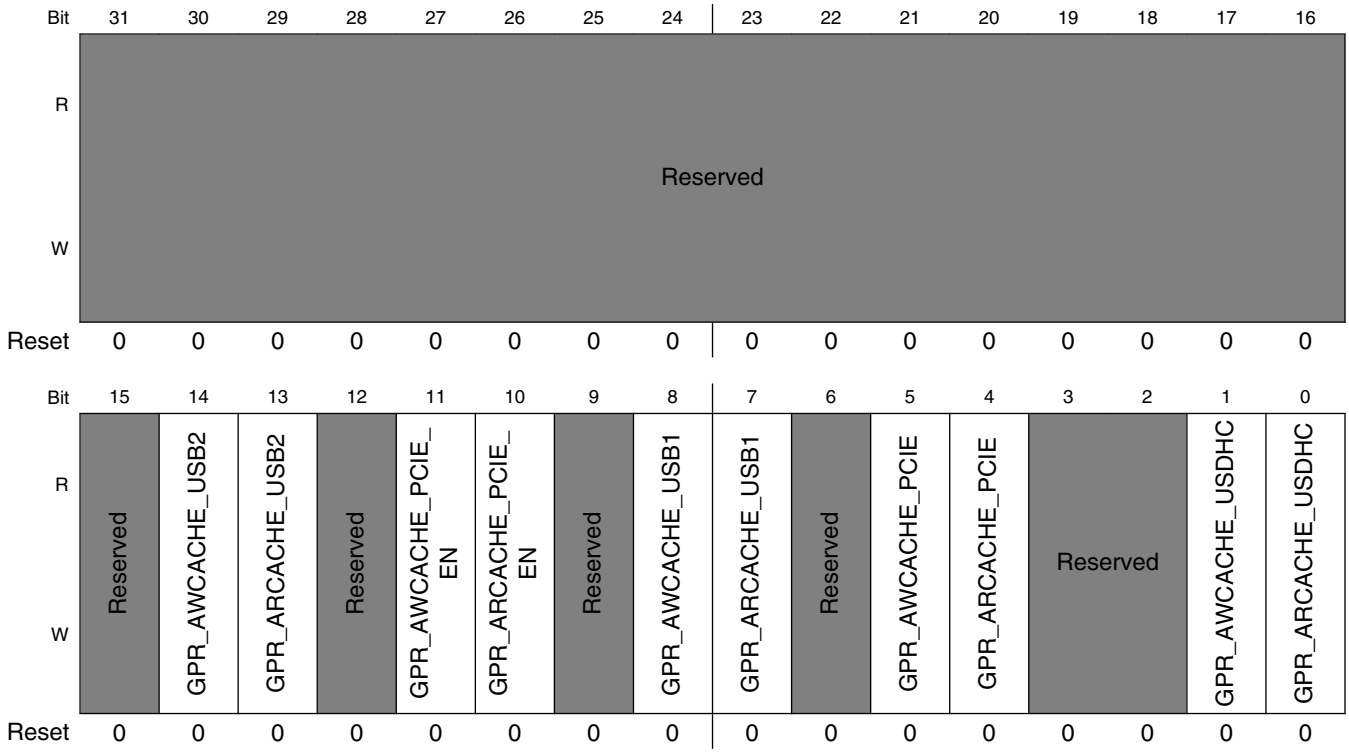


IOMUXC_GPR_GPR12 field descriptions

Field	Description
31 GPR_PCIE_DIAG_BUS_SEL	To PCIe CTRL
30–23 -	This field is reserved.
22–21 GPR_PCIE1_CTRL_DIAG_CTRL_BUS	To PCIe CTRL
20–17 GPR_PCIE1_CTRL_DIAG_STATUS_BUS_SELECT	To PCIe CTRL
16 -	This field is reserved.
15–12 GPR_PCIE1_CTRL_DEVICE_TYPE	To PCIe CTRL
-	This field is reserved.

8.2.3.14 General Purpose Register 13 (IOMUXC_GPR_GPR13)

Address: 3034_0000h base + 34h offset = 3034_0034h



IOMUXC_GPR_GPR13 field descriptions

Field	Description
31–16 -	This field is reserved.
15 -	This field is reserved.
14 GPR_ AWCACHE_ USB2	Control the awcache[1] of usb master transaction
13 GPR_ ARCACHE_ USB2	Control the arcache[1] of usb master transaction
12 -	This field is reserved.
11 GPR_ AWCACHE_ PCIE_EN	Enable the GPR control of AWCACHE[1] of PCIE1 master transaction

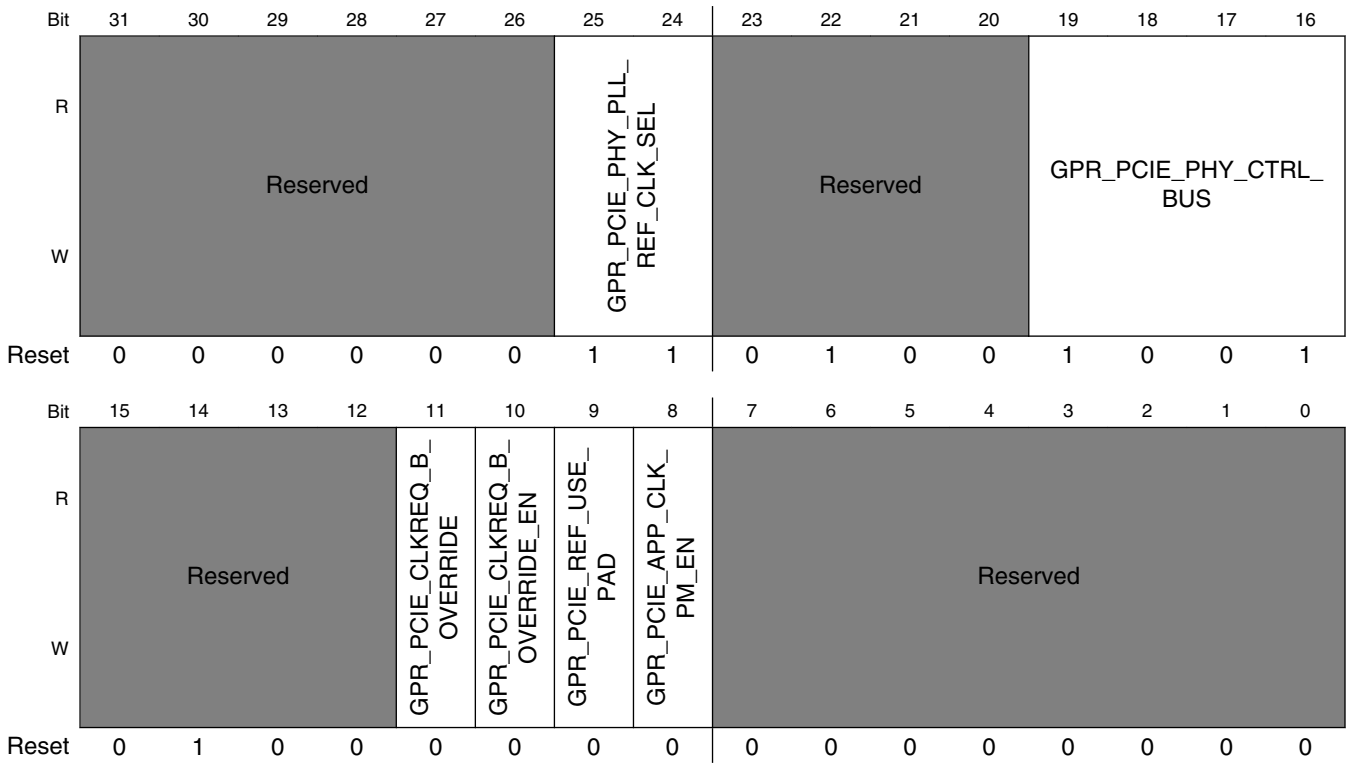
Table continues on the next page...

IOMUXC_GPR_GPR13 field descriptions (continued)

Field	Description
10 GPR_ ARCACHE_ PCIE_EN	Enable the GPR control of ARCACHE[1] of PCIE1 master transaction
9 -	This field is reserved.
8 GPR_ AWCACHE_ USB1	Control the awcache[1] of usb master transaction
7 GPR_ ARCACHE_ USB1	Control the arcache[1] of usb master transaction
6 -	This field is reserved.
5 GPR_ AWCACHE_ PCIE	Control the awcache[1] of pcie master transaction
4 GPR_ ARCACHE_PCIE	Control the arcache[1] of pcie master transaction
3–2 -	This field is reserved.
1 GPR_ AWCACHE_ USDHC	sim_m.awcache_m_d_6/7/8[1]
0 GPR_ ARCACHE_ USDHC	sim_m.arcache_m_d_6/7/8[1]

8.2.3.15 General Purpose Register 14 (IOMUXC_GPR_GPR14)

Address: 3034_0000h base + 38h offset = 3034_0038h



IOMUXC_GPR_GPR14 field descriptions

Field	Description
31–26 -	This field is reserved.
25–24 GPR_PCIE_PHY_PLL_REF_CLK_SEL	PCIE PHY reference clock select 00 N/A 01 Selects reference clock from XO (pll_refclk_from_xo) 10 Selects reference clock from IO (ext_ref_clkp/n) 11 Selects reference clock from SOC PLL (pll_refclk_from_syspll)
23–20 -	This field is reserved.
19–16 GPR_PCIE_PHY_CTRL_BUS	To PCIe CTRL
15–12 -	This field is reserved.
11 GPR_PCIE_CLKREQ_B_OVERRIDE	Control the PCIE_CLKREQ_B to the pad together with CLKREQ_B from controller

Table continues on the next page...

IOMUXC_GPR_GPR14 field descriptions (continued)

Field	Description
10 GPR_PCIE_ CLKREQ_B_ OVERRIDE_EN	Control the PCIE_CLKREQ_B to the pad together with CLKREQ_B from controller
9 GPR_PCIE_ REF_USE_PAD	To PCIe CTRL
8 GPR_PCIE_ APP_CLK_PM_ EN	To PCIe CTRL
-	This field is reserved.

8.2.3.16 General Purpose Register 15 (IOMUXC_GPR_GPR15)

Address: 3034_0000h base + 3Ch offset = 3034_003Ch

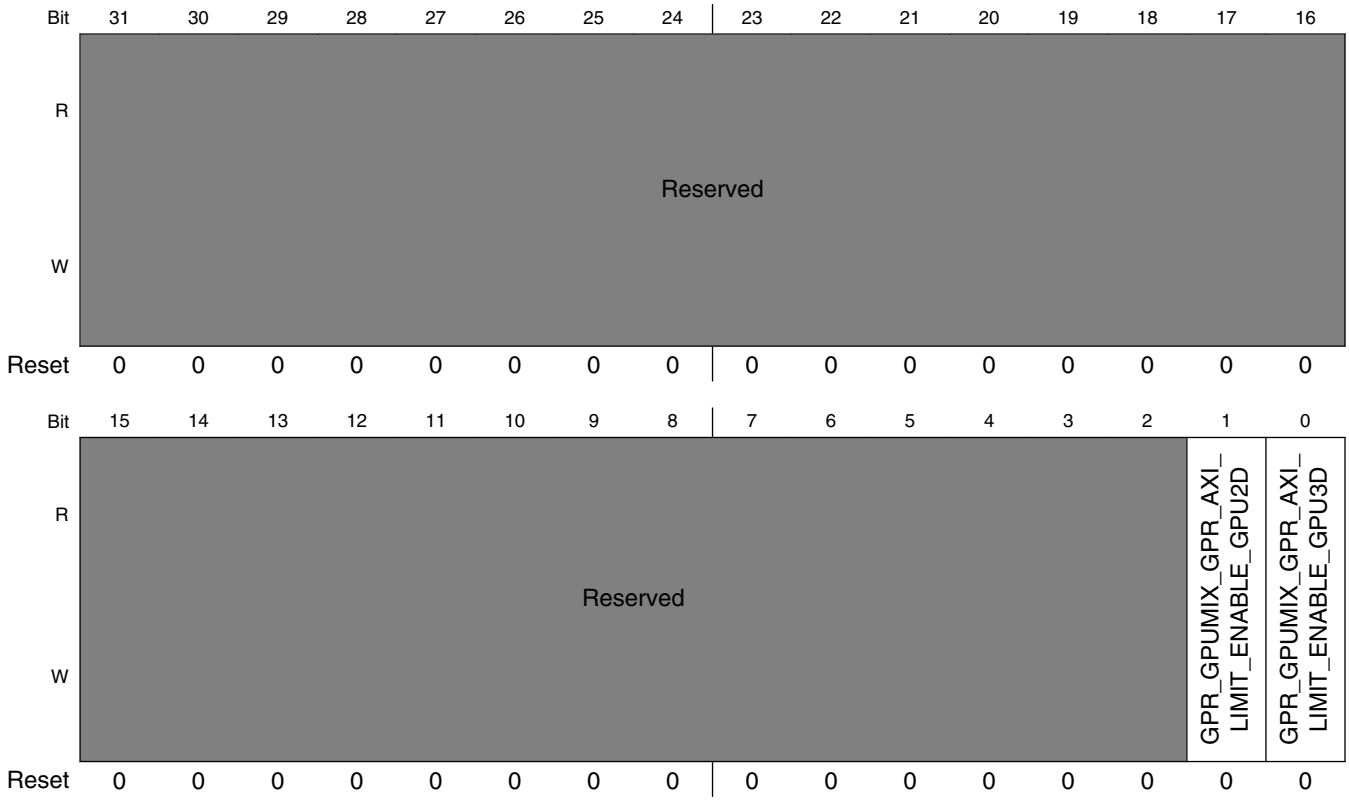
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR_GPUMIX_GPR_AXI_LIMIT_BEAT_LIMIT_GPU2D																GPR_GPUMIX_GPR_AXI_LIMIT_BEAT_LIMIT_GPU3D															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR15 field descriptions

Field	Description
31–16 GPR_GPUMIX_ GPR_AXI_ LIMIT_BEAT_ LIMIT_GPU2D	GPU2D AXI OTR limit beat limit
GPR_GPUMIX_ GPR_AXI_ LIMIT_BEAT_ LIMIT_GPU3D	GPU3D AXI OTR limit beat limit

8.2.3.17 General Purpose Register 16 (IOMUXC_GPR_GPR16)

Address: 3034_0000h base + 40h offset = 3034_0040h



IOMUXC_GPR_GPR16 field descriptions

Field	Description
31–2 -	This field is reserved.
1 GPR_GPUMIX_GPR_AXI_LIMIT_ENABLE_GPU2D	GPU2D AXI OTR limit gasket eanble
0 GPR_GPUMIX_GPR_AXI_LIMIT_ENABLE_GPU3D	GPU3D AXI OTR limit gasket eanble

8.2.3.18 General Purpose Register 17 (IOMUXC_GPR_GPR17)

Address: 3034_0000h base + 44h offset = 3034_0044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR17 field descriptions

Field	Description
-	This field is reserved.

8.2.3.19 General Purpose Register 18 (IOMUXC_GPR_GPR18)

Address: 3034_0000h base + 48h offset = 3034_0048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR18 field descriptions

Field	Description
-	This field is reserved.

8.2.3.20 General Purpose Register 19 (IOMUXC_GPR_GPR19)

Address: 3034_0000h base + 4Ch offset = 3034_004Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PCIE_DIAG_STATUS																															
W																																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- For reset: [31:0] - N/A

IOMUXC_GPR_GPR19 field descriptions

Field	Description
PCIE_DIAG_ STATUS	From PCIE

8.2.3.21 General Purpose Register 20 (IOMUXC_GPR_GPR20)

This chip supports 34-bit addresses. This register contains address bits [33:32] for each AXI master.

Address: 3034_0000h base + 50h offset = 3034_0050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GPR_APBHDMA_M_D_4_HADDR32	GPR_APBHDMA_M_D_4_HADDR33	GPR_CORESIGHT_M_A_0_ARADDR32	GPR_CORESIGHT_M_A_0_ARADDR33	GPR_CORESIGHT_M_A_0_AWADDR32	GPR_CORESIGHT_M_A_0_AWADDR33	GPR_DAP_M_D_0_HADDR32	GPR_DAP_M_D_0_HADDR33	GPR_ENET1_M_E_0_ARADDR32	GPR_ENET1_M_E_0_ARADDR33	GPR_ENET1_M_E_0_AWADDR32	GPR_ENET1_M_E_0_AWADDR33	GPR_ENET1_M_E_1_ARADDR32	GPR_ENET1_M_E_1_ARADDR33	GPR_ENET1_M_E_1_AWADDR32	GPR_ENET1_M_E_1_AWADDR33
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPR_RAWNAND_M_D_5_ARADDR32	GPR_RAWNAND_M_D_5_ARADDR33	GPR_RAWNAND_M_D_5_AWADDR32	GPR_RAWNAND_M_D_5_AWADDR33	GPR_USDHC1_M_D_6_ARADDR32	GPR_USDHC1_M_D_6_ARADDR33	GPR_USDHC1_M_D_6_AWADDR32	GPR_USDHC1_M_D_6_AWADDR33	GPR_USDHC2_M_D_7_ARADDR32	GPR_USDHC2_M_D_7_ARADDR33	GPR_USDHC2_M_D_7_AWADDR32	GPR_USDHC2_M_D_7_AWADDR33	GPR_USDHC3_M_D_8_ARADDR32	GPR_USDHC3_M_D_8_ARADDR33	GPR_USDHC3_M_D_8_AWADDR32	GPR_USDHC3_M_D_8_AWADDR33
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR20 field descriptions

Field	Description
31 GPR_ APBHDMA_M_ D_4_HADDR32	HADDR bit 32 of apbhdma
30 GPR_ APBHDMA_M_ D_4_HADDR33	HADDR bit 33 of apbhdma
29 GPR_ CORESIGHT_M_ A_0_ARADDR32	ARADDR bit 32 of CoreSight
28 GPR_ CORESIGHT_M_ A_0_ARADDR33	ARADDR bit 33 of CoreSight

Table continues on the next page...

IOMUXC_GPR_GPR20 field descriptions (continued)

Field	Description
27 GPR_ CORESIGHT_M_ A_0_AWADDR32	AWADDR bit 32 of CoreSight
26 GPR_ CORESIGHT_M_ A_0_AWADDR33	AWADDR bit 33 of CoreSight
25 GPR_DAP_M_ D_0_HADDR32	AHB address bit 32 of DAP
24 GPR_DAP_M_ D_0_HADDR33	AHB address bit 33 of DAP
23 GPR_ENET1_M_ E_0_ARADDR32	AXI read in fabric m_e_0 address bit 32 of ENET1
22 GPR_ENET1_M_ E_0_ARADDR33	AXI read in fabric m_e_0 address bit 33 of ENET1
21 GPR_ENET1_M_ E_0_AWADDR32	AXI write in fabric m_e_0 address bit 32 of ENET1
20 GPR_ENET1_M_ E_0_AWADDR33	AXI write in fabric m_e_0 address bit 33 of ENET1
19 GPR_ENET1_M_ E_1_ARADDR32	AXI read in fabric m_e_1 address bit 32 of ENET1
18 GPR_ENET1_M_ E_1_ARADDR33	AXI read in fabric m_e_1 address bit 33 of ENET1
17 GPR_ENET1_M_ E_1_AWADDR32	AXI write in fabric m_e_1 address bit 32 of ENET1
16 GPR_ENET1_M_ E_1_AWADDR33	AXI write in fabric m_e_1 address bit 33 of ENET1
15 GPR_ RAWNAND_M_ D_5_ARADDR32	AXI read address bit 32 of RAWNAND
14 GPR_ RAWNAND_M_ D_5_ARADDR33	AXI read address bit 33 of RAWNAND
13 GPR_ RAWNAND_M_	AXI write address bit 32 of RAWNAND

Table continues on the next page...

IOMUXC_GPR_GPR20 field descriptions (continued)

Field	Description
D_5_ AWADDR32	
12 GPR_ RAWNAND_M_ D_5_ AWADDR33	AXI write address bit 33 of RAWNAND
11 GPR_USDHC1_ M_D_6_ ARADDR32	AXI read address bit 32 of USDHC1
10 GPR_USDHC1_ M_D_6_ ARADDR33	AXI read address bit 33 of USDHC1
9 GPR_USDHC1_ M_D_6_ AWADDR32	AXI write address bit 32 of USDHC1
8 GPR_USDHC1_ M_D_6_ AWADDR33	AXI write address bit 33 of USDHC1
7 GPR_USDHC2_ M_D_7_ ARADDR32	AXI read address bit 32 of USDHC2
6 GPR_USDHC2_ M_D_7_ ARADDR33	AXI read address bit 33 of USDHC2
5 GPR_USDHC2_ M_D_7_ AWADDR32	AXI write address bit 32 of USDHC2
4 GPR_USDHC2_ M_D_7_ AWADDR33	AXI write address bit 33 of USDHC2
3 GPR_USDHC3_ M_D_8_ ARADDR32	AXI read address bit 32 of USDHC3
2 GPR_USDHC3_ M_D_8_ ARADDR33	AXI read address bit 33 of USDHC3

Table continues on the next page...

IOMUXC_GPR_GPR20 field descriptions (continued)

Field	Description
1 GPR_USDHC3_M_D_8_AWADDR32	AXI write address bit 32 of USDHC3
0 GPR_USDHC3_M_D_8_AWADDR33	AXI write address bit 33 of USDHC3

8.2.3.22 General Purpose Register 21 (IOMUXC_GPR_GPR21)

Address: 3034_0000h base + 54h offset = 3034_0054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR_SDMA1_M_D_2_HADDR32	GPR_SDMA1_M_D_2_HADDR33	GPR_SDMA1_M_D_3_ARADDR32	GPR_SDMA1_M_D_3_ARADDR33	GPR_SDMA1_M_D_3_AWADDR32	GPR_SDMA1_M_D_3_AWADDR33	Reserved									
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:

- For reset: -

  [31:26] - 0x0

  -

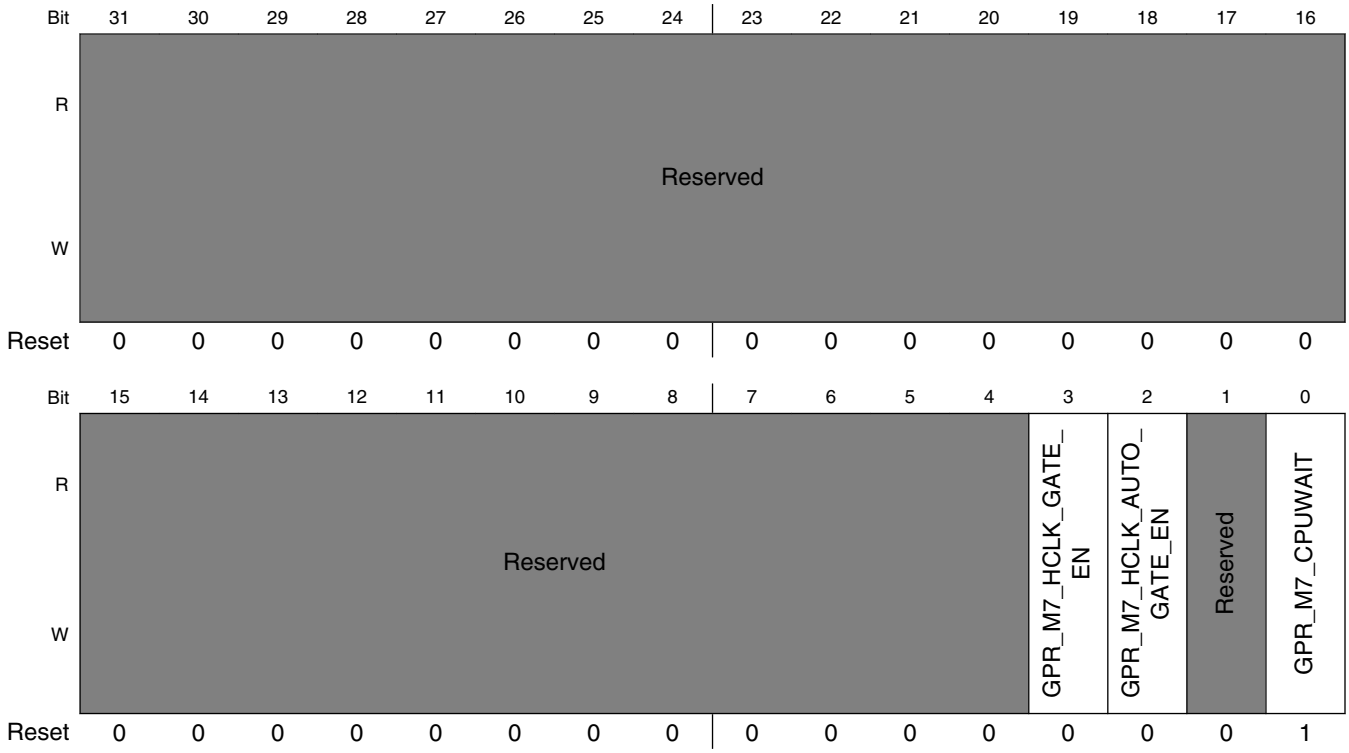
  [25:0] - N/A

IOMUXC_GPR_GPR21 field descriptions

Field	Description
31 GPR_SDMA1_M_D_2_HADDR32	AHB address bit 32 of SDMA1
30 GPR_SDMA1_M_D_2_HADDR33	AHB address bit 33 of SDMA1
29 GPR_SDMA1_M_D_3_ARADDR32	AXI read address bit 32 of SDMA1
28 GPR_SDMA1_M_D_3_ARADDR33	AXI read address bit 33 of SDMA1
27 GPR_SDMA1_M_D_3_AWADDR32	AXI write address bit 32 of SDMA1
26 GPR_SDMA1_M_D_3_AWADDR33	AXI write address bit 33 of SDMA1
-	This field is reserved.

8.2.3.23 General Purpose Register 22 (IOMUXC_GPR_GPR22)

Address: 3034_0000h base + 58h offset = 3034_0058h



IOMUXC_GPR_GPR22 field descriptions

Field	Description
31–16 -	This field is reserved.
15–4 -	This field is reserved.
3 GPR_M7_HCLK_GATE_EN	Gate off CM7 hclk 0 gate on 1 gate off
2 GPR_M7_HCLK_AUTO_GATE_EN	Gate CM7 hclk automatically 0 depends on the value of GPR_M7_HCLK_GATE_EN 1 ignore the value of GPR_M7_HCLK_GATE_EN
1 -	This field is reserved.
0 GPR_M7_CPUWAIT	CM7 in wait mode 0 do not let CM7 enter wait mode 1 let CM7 enter wait mode

8.2.3.24 General Purpose Register 23 (IOMUXC_GPR_GPR23)

Address: 3034_0000h base + 5Ch offset = 3034_005Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR23 field descriptions

Field	Description
-	This field is reserved.

8.2.3.25 General Purpose Register 24 (IOMUXC_GPR_GPR24)

Address: 3034_0000h base + 60h offset = 3034_0060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:

- For reset: -

  [31:24] - N/A

  -

  [23:0] - 0x0

IOMUXC_GPR_GPR24 field descriptions

Field	Description
-	This field is reserved.

8.2.4 IOMUXC Memory Map/Register Definition

IOMUXC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0014	SW_MUX_CTL_PAD_GPIO1_IO00 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO00)	32	R/W	0000_0000h	8.2.4.1/1412
3033_0018	SW_MUX_CTL_PAD_GPIO1_IO01 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO01)	32	R/W	0000_0000h	8.2.4.2/1413

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_001C	SW_MUX_CTL_PAD_GPIO1_IO02 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO02)	32	R/W	0000_0000h	8.2.4.3/1414
3033_0020	SW_MUX_CTL_PAD_GPIO1_IO03 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO03)	32	R/W	0000_0000h	8.2.4.4/1416
3033_0024	SW_MUX_CTL_PAD_GPIO1_IO04 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO04)	32	R/W	0000_0000h	8.2.4.5/1417
3033_0028	SW_MUX_CTL_PAD_GPIO1_IO05 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO05)	32	R/W	0000_0000h	8.2.4.6/1418
3033_002C	SW_MUX_CTL_PAD_GPIO1_IO06 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO06)	32	R/W	0000_0000h	8.2.4.7/1420
3033_0030	SW_MUX_CTL_PAD_GPIO1_IO07 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO07)	32	R/W	0000_0000h	8.2.4.8/1421
3033_0034	SW_MUX_CTL_PAD_GPIO1_IO08 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO08)	32	R/W	0000_0000h	8.2.4.9/1422
3033_0038	SW_MUX_CTL_PAD_GPIO1_IO09 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO09)	32	R/W	0000_0000h	8.2.4.10/ 1424
3033_003C	SW_MUX_CTL_PAD_GPIO1_IO10 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO10)	32	R/W	0000_0000h	8.2.4.11/ 1425
3033_0040	SW_MUX_CTL_PAD_GPIO1_IO11 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO11)	32	R/W	0000_0000h	8.2.4.12/ 1426
3033_0044	SW_MUX_CTL_PAD_GPIO1_IO12 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO12)	32	R/W	0000_0000h	8.2.4.13/ 1427
3033_0048	SW_MUX_CTL_PAD_GPIO1_IO13 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO13)	32	R/W	0000_0000h	8.2.4.14/ 1428
3033_004C	SW_MUX_CTL_PAD_GPIO1_IO14 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO14)	32	R/W	0000_0000h	8.2.4.15/ 1429
3033_0050	SW_MUX_CTL_PAD_GPIO1_IO15 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO15)	32	R/W	0000_0000h	8.2.4.16/ 1431
3033_0054	SW_MUX_CTL_PAD_ENET_MDC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDC)	32	R/W	0000_0005h	8.2.4.17/ 1432
3033_0058	SW_MUX_CTL_PAD_ENET_MDIO SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO)	32	R/W	0000_0005h	8.2.4.18/ 1433
3033_005C	SW_MUX_CTL_PAD_ENET_TD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD3)	32	R/W	0000_0005h	8.2.4.19/ 1435
3033_0060	SW_MUX_CTL_PAD_ENET_TD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD2)	32	R/W	0000_0005h	8.2.4.20/ 1436
3033_0064	SW_MUX_CTL_PAD_ENET_TD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD1)	32	R/W	0000_0005h	8.2.4.21/ 1437
3033_0068	SW_MUX_CTL_PAD_ENET_TD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD0)	32	R/W	0000_0005h	8.2.4.22/ 1439
3033_006C	SW_MUX_CTL_PAD_ENET_TX_CTL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TX_CTL)	32	R/W	0000_0005h	8.2.4.23/ 1440
3033_0070	SW_MUX_CTL_PAD_ENET_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TXC)	32	R/W	0000_0005h	8.2.4.24/ 1441

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0074	SW_MUX_CTL_PAD_ENET_RX_CTL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RX_CTL)	32	R/W	0000_0005h	8.2.4.25/ 1443
3033_0078	SW_MUX_CTL_PAD_ENET_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RXC)	32	R/W	0000_0005h	8.2.4.26/ 1444
3033_007C	SW_MUX_CTL_PAD_ENET_RD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD0)	32	R/W	0000_0005h	8.2.4.27/ 1445
3033_0080	SW_MUX_CTL_PAD_ENET_RD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD1)	32	R/W	0000_0005h	8.2.4.28/ 1447
3033_0084	SW_MUX_CTL_PAD_ENET_RD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD2)	32	R/W	0000_0005h	8.2.4.29/ 1448
3033_0088	SW_MUX_CTL_PAD_ENET_RD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD3)	32	R/W	0000_0005h	8.2.4.30/ 1449
3033_008C	SW_MUX_CTL_PAD_SD1_CLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CLK)	32	R/W	0000_0005h	8.2.4.31/ 1451
3033_0090	SW_MUX_CTL_PAD_SD1_CMD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CMD)	32	R/W	0000_0005h	8.2.4.32/ 1452
3033_0094	SW_MUX_CTL_PAD_SD1_DATA0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0)	32	R/W	0000_0005h	8.2.4.33/ 1453
3033_0098	SW_MUX_CTL_PAD_SD1_DATA1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1)	32	R/W	0000_0005h	8.2.4.34/ 1455
3033_009C	SW_MUX_CTL_PAD_SD1_DATA2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2)	32	R/W	0000_0005h	8.2.4.35/ 1456
3033_00A0	SW_MUX_CTL_PAD_SD1_DATA3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3)	32	R/W	0000_0005h	8.2.4.36/ 1457
3033_00A4	SW_MUX_CTL_PAD_SD1_DATA4 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA4)	32	R/W	0000_0005h	8.2.4.37/ 1459
3033_00A8	SW_MUX_CTL_PAD_SD1_DATA5 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA5)	32	R/W	0000_0005h	8.2.4.38/ 1460
3033_00AC	SW_MUX_CTL_PAD_SD1_DATA6 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA6)	32	R/W	0000_0005h	8.2.4.39/ 1461
3033_00B0	SW_MUX_CTL_PAD_SD1_DATA7 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA7)	32	R/W	0000_0005h	8.2.4.40/ 1463
3033_00B4	SW_MUX_CTL_PAD_SD1_RESET_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_RESET_B)	32	R/W	0000_0005h	8.2.4.41/ 1464
3033_00B8	SW_MUX_CTL_PAD_SD1_STROBE SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_STROBE)	32	R/W	0000_0005h	8.2.4.42/ 1465
3033_00BC	SW_MUX_CTL_PAD_SD2_CD_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CD_B)	32	R/W	0000_0005h	8.2.4.43/ 1467
3033_00C0	SW_MUX_CTL_PAD_SD2_CLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CLK)	32	R/W	0000_0005h	8.2.4.44/ 1468
3033_00C4	SW_MUX_CTL_PAD_SD2_CMD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CMD)	32	R/W	0000_0005h	8.2.4.45/ 1469
3033_00C8	SW_MUX_CTL_PAD_SD2_DATA0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0)	32	R/W	0000_0005h	8.2.4.46/ 1470

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_00CC	SW_MUX_CTL_PAD_SD2_DATA1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1)	32	R/W	0000_0005h	8.2.4.47/ 1471
3033_00D0	SW_MUX_CTL_PAD_SD2_DATA2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2)	32	R/W	0000_0005h	8.2.4.48/ 1473
3033_00D4	SW_MUX_CTL_PAD_SD2_DATA3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3)	32	R/W	0000_0005h	8.2.4.49/ 1474
3033_00D8	SW_MUX_CTL_PAD_SD2_RESET_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_RESET_B)	32	R/W	0000_0005h	8.2.4.50/ 1475
3033_00DC	SW_MUX_CTL_PAD_SD2_WP SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_WP)	32	R/W	0000_0005h	8.2.4.51/ 1476
3033_00E0	SW_MUX_CTL_PAD_NAND_ALE SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_ALE)	32	R/W	0000_0005h	8.2.4.52/ 1477
3033_00E4	SW_MUX_CTL_PAD_NAND_CE0_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE0_B)	32	R/W	0000_0005h	8.2.4.53/ 1479
3033_00E8	SW_MUX_CTL_PAD_NAND_CE1_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE1_B)	32	R/W	0000_0005h	8.2.4.54/ 1480
3033_00EC	SW_MUX_CTL_PAD_NAND_CE2_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE2_B)	32	R/W	0000_0005h	8.2.4.55/ 1482
3033_00F0	SW_MUX_CTL_PAD_NAND_CE3_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE3_B)	32	R/W	0000_0005h	8.2.4.56/ 1483
3033_00F4	SW_MUX_CTL_PAD_NAND_CLE SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CLE)	32	R/W	0000_0005h	8.2.4.57/ 1484
3033_00F8	SW_MUX_CTL_PAD_NAND_DATA00 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00)	32	R/W	0000_0005h	8.2.4.58/ 1486
3033_00FC	SW_MUX_CTL_PAD_NAND_DATA01 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01)	32	R/W	0000_0005h	8.2.4.59/ 1487
3033_0100	SW_MUX_CTL_PAD_NAND_DATA02 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02)	32	R/W	0000_0005h	8.2.4.60/ 1489
3033_0104	SW_MUX_CTL_PAD_NAND_DATA03 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03)	32	R/W	0000_0005h	8.2.4.61/ 1490
3033_0108	SW_MUX_CTL_PAD_NAND_DATA04 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA04)	32	R/W	0000_0005h	8.2.4.62/ 1491
3033_010C	SW_MUX_CTL_PAD_NAND_DATA05 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05)	32	R/W	0000_0005h	8.2.4.63/ 1493
3033_0110	SW_MUX_CTL_PAD_NAND_DATA06 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06)	32	R/W	0000_0005h	8.2.4.64/ 1494
3033_0114	SW_MUX_CTL_PAD_NAND_DATA07 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07)	32	R/W	0000_0005h	8.2.4.65/ 1496
3033_0118	SW_MUX_CTL_PAD_NAND_DQS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DQS)	32	R/W	0000_0005h	8.2.4.66/ 1497
3033_011C	SW_MUX_CTL_PAD_NAND_RE_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_RE_B)	32	R/W	0000_0005h	8.2.4.67/ 1499
3033_0120	SW_MUX_CTL_PAD_NAND_READY_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_READY_B)	32	R/W	0000_0005h	8.2.4.68/ 1500

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0124	SW_MUX_CTL_PAD_NAND_WE_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_WE_B)	32	R/W	0000_0005h	8.2.4.69/ 1501
3033_0128	SW_MUX_CTL_PAD_NAND_WP_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B)	32	R/W	0000_0005h	8.2.4.70/ 1503
3033_012C	SW_MUX_CTL_PAD_SAI5_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXFS)	32	R/W	0000_0005h	8.2.4.71/ 1504
3033_0130	SW_MUX_CTL_PAD_SAI5_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXC)	32	R/W	0000_0005h	8.2.4.72/ 1505
3033_0134	SW_MUX_CTL_PAD_SAI5_RXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD0)	32	R/W	0000_0005h	8.2.4.73/ 1507
3033_0138	SW_MUX_CTL_PAD_SAI5_RXD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD1)	32	R/W	0000_0005h	8.2.4.74/ 1508
3033_013C	SW_MUX_CTL_PAD_SAI5_RXD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD2)	32	R/W	0000_0005h	8.2.4.75/ 1510
3033_0140	SW_MUX_CTL_PAD_SAI5_RXD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD3)	32	R/W	0000_0005h	8.2.4.76/ 1511
3033_0144	SW_MUX_CTL_PAD_SAI5_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_MCLK)	32	R/W	0000_0005h	8.2.4.77/ 1513
3033_0148	SW_MUX_CTL_PAD_SAI1_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXFS)	32	R/W	0000_0005h	8.2.4.78/ 1514
3033_014C	SW_MUX_CTL_PAD_SAI1_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXC)	32	R/W	0000_0005h	8.2.4.79/ 1515
3033_0150	SW_MUX_CTL_PAD_SAI1_RXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD0)	32	R/W	0000_0005h	8.2.4.80/ 1517
3033_0154	SW_MUX_CTL_PAD_SAI1_RXD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD1)	32	R/W	0000_0005h	8.2.4.81/ 1518
3033_0158	SW_MUX_CTL_PAD_SAI1_RXD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD2)	32	R/W	0000_0005h	8.2.4.82/ 1519
3033_015C	SW_MUX_CTL_PAD_SAI1_RXD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD3)	32	R/W	0000_0005h	8.2.4.83/ 1521
3033_0160	SW_MUX_CTL_PAD_SAI1_RXD4 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD4)	32	R/W	0000_0005h	8.2.4.84/ 1522
3033_0164	SW_MUX_CTL_PAD_SAI1_RXD5 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD5)	32	R/W	0000_0005h	8.2.4.85/ 1523
3033_0168	SW_MUX_CTL_PAD_SAI1_RXD6 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD6)	32	R/W	0000_0005h	8.2.4.86/ 1525
3033_016C	SW_MUX_CTL_PAD_SAI1_RXD7 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD7)	32	R/W	0000_0005h	8.2.4.87/ 1526
3033_0170	SW_MUX_CTL_PAD_SAI1_TXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXFS)	32	R/W	0000_0005h	8.2.4.88/ 1527
3033_0174	SW_MUX_CTL_PAD_SAI1_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXC)	32	R/W	0000_0005h	8.2.4.89/ 1529
3033_0178	SW_MUX_CTL_PAD_SAI1_TXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD0)	32	R/W	0000_0005h	8.2.4.90/ 1530

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_017C	SW_MUX_CTL_PAD_SAI1_TXD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD1)	32	R/W	0000_0005h	8.2.4.91/ 1531
3033_0180	SW_MUX_CTL_PAD_SAI1_TXD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD2)	32	R/W	0000_0005h	8.2.4.92/ 1532
3033_0184	SW_MUX_CTL_PAD_SAI1_TXD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD3)	32	R/W	0000_0005h	8.2.4.93/ 1534
3033_0188	SW_MUX_CTL_PAD_SAI1_TXD4 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD4)	32	R/W	0000_0005h	8.2.4.94/ 1535
3033_018C	SW_MUX_CTL_PAD_SAI1_TXD5 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD5)	32	R/W	0000_0005h	8.2.4.95/ 1536
3033_0190	SW_MUX_CTL_PAD_SAI1_TXD6 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD6)	32	R/W	0000_0005h	8.2.4.96/ 1538
3033_0194	SW_MUX_CTL_PAD_SAI1_TXD7 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD7)	32	R/W	0000_0005h	8.2.4.97/ 1539
3033_0198	SW_MUX_CTL_PAD_SAI1_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_MCLK)	32	R/W	0000_0005h	8.2.4.98/ 1540
3033_019C	SW_MUX_CTL_PAD_SAI2_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_RXFS)	32	R/W	0000_0005h	8.2.4.99/ 1542
3033_01A0	SW_MUX_CTL_PAD_SAI2_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_RXC)	32	R/W	0000_0005h	8.2.4.100/ 1543
3033_01A4	SW_MUX_CTL_PAD_SAI2_RXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_RXD0)	32	R/W	0000_0005h	8.2.4.101/ 1545
3033_01A8	SW_MUX_CTL_PAD_SAI2_TXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_TXFS)	32	R/W	0000_0005h	8.2.4.102/ 1546
3033_01AC	SW_MUX_CTL_PAD_SAI2_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_TXC)	32	R/W	0000_0005h	8.2.4.103/ 1548
3033_01B0	SW_MUX_CTL_PAD_SAI2_TXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_TXD0)	32	R/W	0000_0005h	8.2.4.104/ 1549
3033_01B4	SW_MUX_CTL_PAD_SAI2_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_MCLK)	32	R/W	0000_0005h	8.2.4.105/ 1550
3033_01B8	SW_MUX_CTL_PAD_SAI3_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_RXFS)	32	R/W	0000_0005h	8.2.4.106/ 1552
3033_01BC	SW_MUX_CTL_PAD_SAI3_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_RXC)	32	R/W	0000_0005h	8.2.4.107/ 1553
3033_01C0	SW_MUX_CTL_PAD_SAI3_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_RXD)	32	R/W	0000_0005h	8.2.4.108/ 1555
3033_01C4	SW_MUX_CTL_PAD_SAI3_TXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_TXFS)	32	R/W	0000_0005h	8.2.4.109/ 1556
3033_01C8	SW_MUX_CTL_PAD_SAI3_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_TXC)	32	R/W	0000_0005h	8.2.4.110/ 1558
3033_01CC	SW_MUX_CTL_PAD_SAI3_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_TXD)	32	R/W	0000_0005h	8.2.4.111/ 1559
3033_01D0	SW_MUX_CTL_PAD_SAI3_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_MCLK)	32	R/W	0000_0005h	8.2.4.112/ 1561

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_01D4	SW_MUX_CTL_PAD_SPDIF_TX SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SPDIF_TX)	32	R/W	0000_0005h	8.2.4.113/ 1562
3033_01D8	SW_MUX_CTL_PAD_SPDIF_RX SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SPDIF_RX)	32	R/W	0000_0005h	8.2.4.114/ 1563
3033_01DC	SW_MUX_CTL_PAD_SPDIF_EXT_CLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SPDIF_EXT_CLK)	32	R/W	0000_0005h	8.2.4.115/ 1565
3033_01E0	SW_MUX_CTL_PAD_ECSPI1_SCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SCLK)	32	R/W	0000_0005h	8.2.4.116/ 1566
3033_01E4	SW_MUX_CTL_PAD_ECSPI1_MOSI SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MOSI)	32	R/W	0000_0005h	8.2.4.117/ 1567
3033_01E8	SW_MUX_CTL_PAD_ECSPI1_MISO SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MISO)	32	R/W	0000_0005h	8.2.4.118/ 1569
3033_01EC	SW_MUX_CTL_PAD_ECSPI1_SS0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SS0)	32	R/W	0000_0005h	8.2.4.119/ 1570
3033_01F0	SW_MUX_CTL_PAD_ECSPI2_SCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_SCLK)	32	R/W	0000_0005h	8.2.4.120/ 1571
3033_01F4	SW_MUX_CTL_PAD_ECSPI2_MOSI SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MOSI)	32	R/W	0000_0005h	8.2.4.121/ 1573
3033_01F8	SW_MUX_CTL_PAD_ECSPI2_MISO SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MISO)	32	R/W	0000_0005h	8.2.4.122/ 1574
3033_01FC	SW_MUX_CTL_PAD_ECSPI2_SS0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_SS0)	32	R/W	0000_0005h	8.2.4.123/ 1575
3033_0200	SW_MUX_CTL_PAD_I2C1_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C1_SCL)	32	R/W	0000_0005h	8.2.4.124/ 1577
3033_0204	SW_MUX_CTL_PAD_I2C1_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C1_SDA)	32	R/W	0000_0005h	8.2.4.125/ 1578
3033_0208	SW_MUX_CTL_PAD_I2C2_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C2_SCL)	32	R/W	0000_0005h	8.2.4.126/ 1579
3033_020C	SW_MUX_CTL_PAD_I2C2_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C2_SDA)	32	R/W	0000_0005h	8.2.4.127/ 1580
3033_0210	SW_MUX_CTL_PAD_I2C3_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C3_SCL)	32	R/W	0000_0005h	8.2.4.128/ 1582
3033_0214	SW_MUX_CTL_PAD_I2C3_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C3_SDA)	32	R/W	0000_0005h	8.2.4.129/ 1583
3033_0218	SW_MUX_CTL_PAD_I2C4_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C4_SCL)	32	R/W	0000_0005h	8.2.4.130/ 1584
3033_021C	SW_MUX_CTL_PAD_I2C4_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C4_SDA)	32	R/W	0000_0005h	8.2.4.131/ 1585
3033_0220	SW_MUX_CTL_PAD_UART1_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART1_RXD)	32	R/W	0000_0005h	8.2.4.132/ 1586
3033_0224	SW_MUX_CTL_PAD_UART1_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART1_TXD)	32	R/W	0000_0005h	8.2.4.133/ 1587
3033_0228	SW_MUX_CTL_PAD_UART2_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART2_RXD)	32	R/W	0000_0005h	8.2.4.134/ 1588

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_022C	SW_MUX_CTL_PAD_UART2_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART2_TXD)	32	R/W	0000_0005h	8.2.4.135/ 1590
3033_0230	SW_MUX_CTL_PAD_UART3_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART3_RXD)	32	R/W	0000_0005h	8.2.4.136/ 1591
3033_0234	SW_MUX_CTL_PAD_UART3_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART3_TXD)	32	R/W	0000_0005h	8.2.4.137/ 1592
3033_0238	SW_MUX_CTL_PAD_UART4_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART4_RXD)	32	R/W	0000_0005h	8.2.4.138/ 1594
3033_023C	SW_MUX_CTL_PAD_UART4_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART4_TXD)	32	R/W	0000_0005h	8.2.4.139/ 1595
3033_0240	SW_MUX_CTL_PAD_HDMI_DDC_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_HDMI_DDC_SCL)	32	R/W	0000_0005h	8.2.4.140/ 1596
3033_0244	SW_MUX_CTL_PAD_HDMI_DDC_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_HDMI_DDC_SDA)	32	R/W	0000_0005h	8.2.4.141/ 1598
3033_0248	SW_MUX_CTL_PAD_HDMI_CEC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_HDMI_CEC)	32	R/W	0000_0005h	8.2.4.142/ 1599
3033_024C	SW_MUX_CTL_PAD_HDMI_HPD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_HDMI_HPD)	32	R/W	0000_0005h	8.2.4.143/ 1600
3033_0250	SW_PAD_CTL_PAD_BOOT_MODE0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0)	32	R/W	0000_0194h	8.2.4.144/ 1601
3033_0254	SW_PAD_CTL_PAD_BOOT_MODE1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1)	32	R/W	0000_0194h	8.2.4.145/ 1603
3033_0258	SW_PAD_CTL_PAD_BOOT_MODE2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE2)	32	R/W	0000_0194h	8.2.4.146/ 1605
3033_025C	SW_PAD_CTL_PAD_BOOT_MODE3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE3)	32	R/W	0000_0194h	8.2.4.147/ 1607
3033_0260	SW_PAD_CTL_PAD_JTAG_MOD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD)	32	R/W	0000_0114h	8.2.4.148/ 1609
3033_0264	SW_PAD_CTL_PAD_JTAG_TDI SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI)	32	R/W	0000_0154h	8.2.4.149/ 1611
3033_0268	SW_PAD_CTL_PAD_JTAG_TMS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS)	32	R/W	0000_0154h	8.2.4.150/ 1613
3033_026C	SW_PAD_CTL_PAD_JTAG_TCK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK)	32	R/W	0000_0154h	8.2.4.151/ 1615
3033_0270	SW_PAD_CTL_PAD_JTAG_TDO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO)	32	R/W	0000_0154h	8.2.4.152/ 1617
3033_0274	SW_PAD_CTL_PAD_GPIO1_IO00 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO00)	32	R/W	0000_0106h	8.2.4.153/ 1619
3033_0278	SW_PAD_CTL_PAD_GPIO1_IO01 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO01)	32	R/W	0000_0106h	8.2.4.154/ 1621
3033_027C	SW_PAD_CTL_PAD_GPIO1_IO02 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO02)	32	R/W	0000_0146h	8.2.4.155/ 1623

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0280	SW_PAD_CTL_PAD_GPIO1_IO03 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO03)	32	R/W	0000_0106h	8.2.4.156/ 1625
3033_0284	SW_PAD_CTL_PAD_GPIO1_IO04 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO04)	32	R/W	0000_0106h	8.2.4.157/ 1627
3033_0288	SW_PAD_CTL_PAD_GPIO1_IO05 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO05)	32	R/W	0000_0146h	8.2.4.158/ 1629
3033_028C	SW_PAD_CTL_PAD_GPIO1_IO06 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO06)	32	R/W	0000_0106h	8.2.4.159/ 1631
3033_0290	SW_PAD_CTL_PAD_GPIO1_IO07 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO07)	32	R/W	0000_0106h	8.2.4.160/ 1633
3033_0294	SW_PAD_CTL_PAD_GPIO1_IO08 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO08)	32	R/W	0000_0106h	8.2.4.161/ 1635
3033_0298	SW_PAD_CTL_PAD_GPIO1_IO09 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO09)	32	R/W	0000_0106h	8.2.4.162/ 1637
3033_029C	SW_PAD_CTL_PAD_GPIO1_IO10 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO10)	32	R/W	0000_0106h	8.2.4.163/ 1639
3033_02A0	SW_PAD_CTL_PAD_GPIO1_IO11 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO11)	32	R/W	0000_0106h	8.2.4.164/ 1641
3033_02A4	SW_PAD_CTL_PAD_GPIO1_IO12 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO12)	32	R/W	0000_0106h	8.2.4.165/ 1643
3033_02A8	SW_PAD_CTL_PAD_GPIO1_IO13 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO13)	32	R/W	0000_0106h	8.2.4.166/ 1645
3033_02AC	SW_PAD_CTL_PAD_GPIO1_IO14 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO14)	32	R/W	0000_0106h	8.2.4.167/ 1647
3033_02B0	SW_PAD_CTL_PAD_GPIO1_IO15 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO15)	32	R/W	0000_0106h	8.2.4.168/ 1649
3033_02B4	SW_PAD_CTL_PAD_ENET_MDC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDC)	32	R/W	0000_0106h	8.2.4.169/ 1651
3033_02B8	SW_PAD_CTL_PAD_ENET_MDIO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO)	32	R/W	0000_0106h	8.2.4.170/ 1653
3033_02BC	SW_PAD_CTL_PAD_ENET_TD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD3)	32	R/W	0000_0106h	8.2.4.171/ 1655
3033_02C0	SW_PAD_CTL_PAD_ENET_TD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD2)	32	R/W	0000_0106h	8.2.4.172/ 1657
3033_02C4	SW_PAD_CTL_PAD_ENET_TD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD1)	32	R/W	0000_0106h	8.2.4.173/ 1659
3033_02C8	SW_PAD_CTL_PAD_ENET_TD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD0)	32	R/W	0000_0106h	8.2.4.174/ 1661
3033_02CC	SW_PAD_CTL_PAD_ENET_TX_CTL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TX_CTL)	32	R/W	0000_0106h	8.2.4.175/ 1663
3033_02D0	SW_PAD_CTL_PAD_ENET_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TXC)	32	R/W	0000_0106h	8.2.4.176/ 1665
3033_02D4	SW_PAD_CTL_PAD_ENET_RX_CTL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RX_CTL)	32	R/W	0000_0106h	8.2.4.177/ 1667

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_02D8	SW_PAD_CTL_PAD_ENET_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RXC)	32	R/W	0000_0106h	8.2.4.178/ 1669
3033_02DC	SW_PAD_CTL_PAD_ENET_RD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD0)	32	R/W	0000_0106h	8.2.4.179/ 1671
3033_02E0	SW_PAD_CTL_PAD_ENET_RD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD1)	32	R/W	0000_0106h	8.2.4.180/ 1673
3033_02E4	SW_PAD_CTL_PAD_ENET_RD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD2)	32	R/W	0000_0106h	8.2.4.181/ 1675
3033_02E8	SW_PAD_CTL_PAD_ENET_RD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD3)	32	R/W	0000_0106h	8.2.4.182/ 1677
3033_02EC	SW_PAD_CTL_PAD_SD1_CLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CLK)	32	R/W	0000_0106h	8.2.4.183/ 1679
3033_02F0	SW_PAD_CTL_PAD_SD1_CMD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CMD)	32	R/W	0000_0106h	8.2.4.184/ 1681
3033_02F4	SW_PAD_CTL_PAD_SD1_DATA0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0)	32	R/W	0000_0106h	8.2.4.185/ 1683
3033_02F8	SW_PAD_CTL_PAD_SD1_DATA1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1)	32	R/W	0000_0106h	8.2.4.186/ 1685
3033_02FC	SW_PAD_CTL_PAD_SD1_DATA2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2)	32	R/W	0000_0106h	8.2.4.187/ 1687
3033_0300	SW_PAD_CTL_PAD_SD1_DATA3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3)	32	R/W	0000_0106h	8.2.4.188/ 1689
3033_0304	SW_PAD_CTL_PAD_SD1_DATA4 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA4)	32	R/W	0000_0106h	8.2.4.189/ 1691
3033_0308	SW_PAD_CTL_PAD_SD1_DATA5 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA5)	32	R/W	0000_0106h	8.2.4.190/ 1693
3033_030C	SW_PAD_CTL_PAD_SD1_DATA6 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA6)	32	R/W	0000_0106h	8.2.4.191/ 1695
3033_0310	SW_PAD_CTL_PAD_SD1_DATA7 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA7)	32	R/W	0000_0106h	8.2.4.192/ 1697
3033_0314	SW_PAD_CTL_PAD_SD1_RESET_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_RESET_B)	32	R/W	0000_0106h	8.2.4.193/ 1699
3033_0318	SW_PAD_CTL_PAD_SD1_STROBE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_STROBE)	32	R/W	0000_0106h	8.2.4.194/ 1701
3033_031C	SW_PAD_CTL_PAD_SD2_CD_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CD_B)	32	R/W	0000_0106h	8.2.4.195/ 1703
3033_0320	SW_PAD_CTL_PAD_SD2_CLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CLK)	32	R/W	0000_0106h	8.2.4.196/ 1705
3033_0324	SW_PAD_CTL_PAD_SD2_CMD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CMD)	32	R/W	0000_0106h	8.2.4.197/ 1707
3033_0328	SW_PAD_CTL_PAD_SD2_DATA0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0)	32	R/W	0000_0106h	8.2.4.198/ 1709
3033_032C	SW_PAD_CTL_PAD_SD2_DATA1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1)	32	R/W	0000_0106h	8.2.4.199/ 1711

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0330	SW_PAD_CTL_PAD_SD2_DATA2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2)	32	R/W	0000_0106h	8.2.4.200/ 1713
3033_0334	SW_PAD_CTL_PAD_SD2_DATA3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3)	32	R/W	0000_0106h	8.2.4.201/ 1715
3033_0338	SW_PAD_CTL_PAD_SD2_RESET_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_RESET_B)	32	R/W	0000_0106h	8.2.4.202/ 1717
3033_033C	SW_PAD_CTL_PAD_SD2_WP SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_WP)	32	R/W	0000_0106h	8.2.4.203/ 1719
3033_0340	SW_PAD_CTL_PAD_NAND_ALE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_ALE)	32	R/W	0000_0106h	8.2.4.204/ 1721
3033_0344	SW_PAD_CTL_PAD_NAND_CE0_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE0_B)	32	R/W	0000_0106h	8.2.4.205/ 1723
3033_0348	SW_PAD_CTL_PAD_NAND_CE1_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE1_B)	32	R/W	0000_0106h	8.2.4.206/ 1725
3033_034C	SW_PAD_CTL_PAD_NAND_CE2_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE2_B)	32	R/W	0000_0106h	8.2.4.207/ 1727
3033_0350	SW_PAD_CTL_PAD_NAND_CE3_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE3_B)	32	R/W	0000_0106h	8.2.4.208/ 1729
3033_0354	SW_PAD_CTL_PAD_NAND_CLE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CLE)	32	R/W	0000_0106h	8.2.4.209/ 1731
3033_0358	SW_PAD_CTL_PAD_NAND_DATA00 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA00)	32	R/W	0000_0106h	8.2.4.210/ 1733
3033_035C	SW_PAD_CTL_PAD_NAND_DATA01 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01)	32	R/W	0000_0106h	8.2.4.211/ 1735
3033_0360	SW_PAD_CTL_PAD_NAND_DATA02 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA02)	32	R/W	0000_0106h	8.2.4.212/ 1737
3033_0364	SW_PAD_CTL_PAD_NAND_DATA03 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03)	32	R/W	0000_0106h	8.2.4.213/ 1739
3033_0368	SW_PAD_CTL_PAD_NAND_DATA04 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA04)	32	R/W	0000_0106h	8.2.4.214/ 1741
3033_036C	SW_PAD_CTL_PAD_NAND_DATA05 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA05)	32	R/W	0000_0106h	8.2.4.215/ 1743
3033_0370	SW_PAD_CTL_PAD_NAND_DATA06 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA06)	32	R/W	0000_0106h	8.2.4.216/ 1745
3033_0374	SW_PAD_CTL_PAD_NAND_DATA07 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07)	32	R/W	0000_0106h	8.2.4.217/ 1747
3033_0378	SW_PAD_CTL_PAD_NAND_DQS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DQS)	32	R/W	0000_0106h	8.2.4.218/ 1749
3033_037C	SW_PAD_CTL_PAD_NAND_RE_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_RE_B)	32	R/W	0000_0106h	8.2.4.219/ 1751
3033_0380	SW_PAD_CTL_PAD_NAND_READY_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_READY_B)	32	R/W	0000_0106h	8.2.4.220/ 1753
3033_0384	SW_PAD_CTL_PAD_NAND_WE_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_WE_B)	32	R/W	0000_0106h	8.2.4.221/ 1755

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0388	SW_PAD_CTL_PAD_NAND_WP_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_WP_B)	32	R/W	0000_0106h	8.2.4.222/ 1757
3033_038C	SW_PAD_CTL_PAD_SAI5_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXFS)	32	R/W	0000_0106h	8.2.4.223/ 1759
3033_0390	SW_PAD_CTL_PAD_SAI5_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXC)	32	R/W	0000_0106h	8.2.4.224/ 1761
3033_0394	SW_PAD_CTL_PAD_SAI5_RXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD0)	32	R/W	0000_0106h	8.2.4.225/ 1763
3033_0398	SW_PAD_CTL_PAD_SAI5_RXD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD1)	32	R/W	0000_0106h	8.2.4.226/ 1765
3033_039C	SW_PAD_CTL_PAD_SAI5_RXD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD2)	32	R/W	0000_0106h	8.2.4.227/ 1767
3033_03A0	SW_PAD_CTL_PAD_SAI5_RXD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD3)	32	R/W	0000_0106h	8.2.4.228/ 1769
3033_03A4	SW_PAD_CTL_PAD_SAI5_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_MCLK)	32	R/W	0000_0106h	8.2.4.229/ 1771
3033_03A8	SW_PAD_CTL_PAD_SAI1_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXFS)	32	R/W	0000_0106h	8.2.4.230/ 1773
3033_03AC	SW_PAD_CTL_PAD_SAI1_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXC)	32	R/W	0000_0106h	8.2.4.231/ 1775
3033_03B0	SW_PAD_CTL_PAD_SAI1_RXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD0)	32	R/W	0000_0106h	8.2.4.232/ 1777
3033_03B4	SW_PAD_CTL_PAD_SAI1_RXD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD1)	32	R/W	0000_0106h	8.2.4.233/ 1779
3033_03B8	SW_PAD_CTL_PAD_SAI1_RXD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD2)	32	R/W	0000_0106h	8.2.4.234/ 1781
3033_03BC	SW_PAD_CTL_PAD_SAI1_RXD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD3)	32	R/W	0000_0106h	8.2.4.235/ 1783
3033_03C0	SW_PAD_CTL_PAD_SAI1_RXD4 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD4)	32	R/W	0000_0106h	8.2.4.236/ 1785
3033_03C4	SW_PAD_CTL_PAD_SAI1_RXD5 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD5)	32	R/W	0000_0106h	8.2.4.237/ 1787
3033_03C8	SW_PAD_CTL_PAD_SAI1_RXD6 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD6)	32	R/W	0000_0106h	8.2.4.238/ 1789
3033_03CC	SW_PAD_CTL_PAD_SAI1_RXD7 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD7)	32	R/W	0000_0106h	8.2.4.239/ 1791
3033_03D0	SW_PAD_CTL_PAD_SAI1_TXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXFS)	32	R/W	0000_0106h	8.2.4.240/ 1793
3033_03D4	SW_PAD_CTL_PAD_SAI1_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXC)	32	R/W	0000_0106h	8.2.4.241/ 1795
3033_03D8	SW_PAD_CTL_PAD_SAI1_TXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD0)	32	R/W	0000_0106h	8.2.4.242/ 1797
3033_03DC	SW_PAD_CTL_PAD_SAI1_TXD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD1)	32	R/W	0000_0106h	8.2.4.243/ 1799

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_03E0	SW_PAD_CTL_PAD_SAI1_TXD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD2)	32	R/W	0000_0106h	8.2.4.244/ 1801
3033_03E4	SW_PAD_CTL_PAD_SAI1_TXD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD3)	32	R/W	0000_0106h	8.2.4.245/ 1803
3033_03E8	SW_PAD_CTL_PAD_SAI1_TXD4 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD4)	32	R/W	0000_0106h	8.2.4.246/ 1805
3033_03EC	SW_PAD_CTL_PAD_SAI1_TXD5 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD5)	32	R/W	0000_0106h	8.2.4.247/ 1807
3033_03F0	SW_PAD_CTL_PAD_SAI1_TXD6 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD6)	32	R/W	0000_0106h	8.2.4.248/ 1809
3033_03F4	SW_PAD_CTL_PAD_SAI1_TXD7 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD7)	32	R/W	0000_0106h	8.2.4.249/ 1811
3033_03F8	SW_PAD_CTL_PAD_SAI1_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_MCLK)	32	R/W	0000_0106h	8.2.4.250/ 1813
3033_03FC	SW_PAD_CTL_PAD_SAI2_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_RXFS)	32	R/W	0000_0106h	8.2.4.251/ 1815
3033_0400	SW_PAD_CTL_PAD_SAI2_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_RXC)	32	R/W	0000_0106h	8.2.4.252/ 1817
3033_0404	SW_PAD_CTL_PAD_SAI2_RXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_RXD0)	32	R/W	0000_0106h	8.2.4.253/ 1819
3033_0408	SW_PAD_CTL_PAD_SAI2_TXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_TXFS)	32	R/W	0000_0106h	8.2.4.254/ 1821
3033_040C	SW_PAD_CTL_PAD_SAI2_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_TXC)	32	R/W	0000_0106h	8.2.4.255/ 1823
3033_0410	SW_PAD_CTL_PAD_SAI2_TXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_TXD0)	32	R/W	0000_0106h	8.2.4.256/ 1825
3033_0414	SW_PAD_CTL_PAD_SAI2_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_MCLK)	32	R/W	0000_0106h	8.2.4.257/ 1827
3033_0418	SW_PAD_CTL_PAD_SAI3_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_RXFS)	32	R/W	0000_0106h	8.2.4.258/ 1829
3033_041C	SW_PAD_CTL_PAD_SAI3_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_RXC)	32	R/W	0000_0106h	8.2.4.259/ 1831
3033_0420	SW_PAD_CTL_PAD_SAI3_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_RXD)	32	R/W	0000_0106h	8.2.4.260/ 1833
3033_0424	SW_PAD_CTL_PAD_SAI3_TXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_TXFS)	32	R/W	0000_0106h	8.2.4.261/ 1835
3033_0428	SW_PAD_CTL_PAD_SAI3_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_TXC)	32	R/W	0000_0106h	8.2.4.262/ 1837
3033_042C	SW_PAD_CTL_PAD_SAI3_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_TXD)	32	R/W	0000_0106h	8.2.4.263/ 1839
3033_0430	SW_PAD_CTL_PAD_SAI3_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_MCLK)	32	R/W	0000_0106h	8.2.4.264/ 1841
3033_0434	SW_PAD_CTL_PAD_SPDIF_TX SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SPDIF_TX)	32	R/W	0000_0106h	8.2.4.265/ 1843

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0438	SW_PAD_CTL_PAD_SPDIF_RX SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SPDIF_RX)	32	R/W	0000_0106h	8.2.4.266/ 1845
3033_043C	SW_PAD_CTL_PAD_SPDIF_EXT_CLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SPDIF_EXT_CLK)	32	R/W	0000_0106h	8.2.4.267/ 1847
3033_0440	SW_PAD_CTL_PAD_ECSPI1_SCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SCLK)	32	R/W	0000_0106h	8.2.4.268/ 1849
3033_0444	SW_PAD_CTL_PAD_ECSPI1_MOSI SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MOSI)	32	R/W	0000_0106h	8.2.4.269/ 1851
3033_0448	SW_PAD_CTL_PAD_ECSPI1_MISO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MISO)	32	R/W	0000_0106h	8.2.4.270/ 1853
3033_044C	SW_PAD_CTL_PAD_ECSPI1_SS0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SS0)	32	R/W	0000_0106h	8.2.4.271/ 1855
3033_0450	SW_PAD_CTL_PAD_ECSPI2_SCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SCLK)	32	R/W	0000_0106h	8.2.4.272/ 1857
3033_0454	SW_PAD_CTL_PAD_ECSPI2_MOSI SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MOSI)	32	R/W	0000_0106h	8.2.4.273/ 1859
3033_0458	SW_PAD_CTL_PAD_ECSPI2_MISO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MISO)	32	R/W	0000_0106h	8.2.4.274/ 1861
3033_045C	SW_PAD_CTL_PAD_ECSPI2_SS0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SS0)	32	R/W	0000_0106h	8.2.4.275/ 1863
3033_0460	SW_PAD_CTL_PAD_I2C1_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C1_SCL)	32	R/W	0000_0106h	8.2.4.276/ 1865
3033_0464	SW_PAD_CTL_PAD_I2C1_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C1_SDA)	32	R/W	0000_0106h	8.2.4.277/ 1867
3033_0468	SW_PAD_CTL_PAD_I2C2_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C2_SCL)	32	R/W	0000_0106h	8.2.4.278/ 1869
3033_046C	SW_PAD_CTL_PAD_I2C2_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C2_SDA)	32	R/W	0000_0106h	8.2.4.279/ 1871
3033_0470	SW_PAD_CTL_PAD_I2C3_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C3_SCL)	32	R/W	0000_0106h	8.2.4.280/ 1873
3033_0474	SW_PAD_CTL_PAD_I2C3_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C3_SDA)	32	R/W	0000_0106h	8.2.4.281/ 1875
3033_0478	SW_PAD_CTL_PAD_I2C4_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C4_SCL)	32	R/W	0000_0106h	8.2.4.282/ 1877
3033_047C	SW_PAD_CTL_PAD_I2C4_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C4_SDA)	32	R/W	0000_0106h	8.2.4.283/ 1879
3033_0480	SW_PAD_CTL_PAD_UART1_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART1_RXD)	32	R/W	0000_0106h	8.2.4.284/ 1881
3033_0484	SW_PAD_CTL_PAD_UART1_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART1_TXD)	32	R/W	0000_0106h	8.2.4.285/ 1883
3033_0488	SW_PAD_CTL_PAD_UART2_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART2_RXD)	32	R/W	0000_0106h	8.2.4.286/ 1885
3033_048C	SW_PAD_CTL_PAD_UART2_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART2_TXD)	32	R/W	0000_0106h	8.2.4.287/ 1887

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0490	SW_PAD_CTL_PAD_UART3_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART3_RXD)	32	R/W	0000_0106h	8.2.4.288/ 1889
3033_0494	SW_PAD_CTL_PAD_UART3_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART3_TXD)	32	R/W	0000_0106h	8.2.4.289/ 1891
3033_0498	SW_PAD_CTL_PAD_UART4_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART4_RXD)	32	R/W	0000_0106h	8.2.4.290/ 1893
3033_049C	SW_PAD_CTL_PAD_UART4_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART4_TXD)	32	R/W	0000_0106h	8.2.4.291/ 1895
3033_04A0	SW_PAD_CTL_PAD_HDMI_DDC_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_HDMI_DDC_SCL)	32	R/W	0000_0106h	8.2.4.292/ 1897
3033_04A4	SW_PAD_CTL_PAD_HDMI_DDC_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_HDMI_DDC_SDA)	32	R/W	0000_0106h	8.2.4.293/ 1899
3033_04A8	SW_PAD_CTL_PAD_HDMI_CEC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_HDMI_CEC)	32	R/W	0000_0106h	8.2.4.294/ 1901
3033_04AC	SW_PAD_CTL_PAD_HDMI_HPD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_HDMI_HPD)	32	R/W	0000_0106h	8.2.4.295/ 1903
3033_04B0	SW_PAD_CTL_PAD_CLKIN1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_CLKIN1)	32	R/W	0000_0106h	8.2.4.296/ 1905
3033_04B4	SW_PAD_CTL_PAD_CLKIN2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_CLKIN2)	32	R/W	0000_0106h	8.2.4.297/ 1907
3033_04B8	SW_PAD_CTL_PAD_CLKOUT1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_CLKOUT1)	32	R/W	0000_0106h	8.2.4.298/ 1909
3033_04BC	SW_PAD_CTL_PAD_CLKOUT2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_CLKOUT2)	32	R/W	0000_0106h	8.2.4.299/ 1911
3033_04C0	AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_0 DAISY Register (IOMUXC_AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_0)	32	R/W	0000_0000h	8.2.4.300/ 1912
3033_04C4	AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_1 DAISY Register (IOMUXC_AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_1)	32	R/W	0000_0000h	8.2.4.301/ 1913
3033_04C8	AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_2 DAISY Register (IOMUXC_AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_2)	32	R/W	0000_0000h	8.2.4.302/ 1914
3033_04CC	AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_3 DAISY Register (IOMUXC_AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_3)	32	R/W	0000_0000h	8.2.4.303/ 1914
3033_04D0	AUDIOMIX_SAI1_RXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI1_RXSYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.304/ 1915

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_04D4	AUDIOMIX_SAI1_TXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI1_TXBCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.305/ 1916
3033_04D8	AUDIOMIX_SAI1_TXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI1_TXSYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.306/ 1916
3033_04DC	AUDIOMIX_SAI2_RXDATA_SELECT_INPUT_1 DAISY Register (IOMUXC_AUDIOMIX_SAI2_RXDATA_SELECT_INPUT_1)	32	R/W	0000_0000h	8.2.4.307/ 1917
3033_04E0	AUDIOMIX_SAI3_MCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI3_MCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.308/ 1918
3033_04E4	AUDIOMIX_SAI3_RXDATA_SELECT_INPUT_0 DAISY Register (IOMUXC_AUDIOMIX_SAI3_RXDATA_SELECT_INPUT_0)	32	R/W	0000_0000h	8.2.4.309/ 1919
3033_04E8	AUDIOMIX_SAI3_TXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI3_TXBCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.310/ 1920
3033_04EC	AUDIOMIX_SAI3_TXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI3_TXSYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.311/ 1921
3033_04F0	AUDIOMIX_SAI5_MCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI5_MCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.312/ 1921
3033_04F4	AUDIOMIX_SAI5_RXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI5_RXBCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.313/ 1922
3033_04F8	AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_0 DAISY Register (IOMUXC_AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_0)	32	R/W	0000_0000h	8.2.4.314/ 1923
3033_04FC	AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_1 DAISY Register (IOMUXC_AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_1)	32	R/W	0000_0000h	8.2.4.315/ 1923
3033_0500	AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_2 DAISY Register (IOMUXC_AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_2)	32	R/W	0000_0000h	8.2.4.316/ 1924
3033_0504	AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_3 DAISY Register (IOMUXC_AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_3)	32	R/W	0000_0000h	8.2.4.317/ 1925
3033_0508	AUDIOMIX_SAI5_RXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI5_RXSYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.318/ 1925
3033_050C	AUDIOMIX_SAI5_TXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI5_TXBCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.319/ 1926
3033_0510	AUDIOMIX_SAI5_TXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI5_TXSYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.320/ 1927

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0514	AUDIOMIX_SAI6_MCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI6_MCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.321/ 1927
3033_0518	AUDIOMIX_SAI6_RXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI6_RXBCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.322/ 1928
3033_051C	AUDIOMIX_SAI6_RXDATA_SELECT_INPUT_0 DAISY Register (IOMUXC_AUDIOMIX_SAI6_RXDATA_SELECT_INPUT_0)	32	R/W	0000_0000h	8.2.4.323/ 1929
3033_0520	AUDIOMIX_SAI6_RXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI6_RXSYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.324/ 1929
3033_0524	AUDIOMIX_SAI6_TXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI6_TXBCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.325/ 1930
3033_0528	AUDIOMIX_SAI6_TXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI6_TXSYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.326/ 1931
3033_052C	AUDIOMIX_SAI7_MCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI7_MCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.327/ 1932
3033_0530	AUDIOMIX_SAI7_RXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI7_RXBCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.328/ 1933
3033_0534	AUDIOMIX_SAI7_RXDATA_SELECT_INPUT_0 DAISY Register (IOMUXC_AUDIOMIX_SAI7_RXDATA_SELECT_INPUT_0)	32	R/W	0000_0000h	8.2.4.329/ 1934
3033_0538	AUDIOMIX_SAI7_RXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI7_RXSYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.330/ 1935
3033_053C	AUDIOMIX_SAI7_TXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI7_TXBCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.331/ 1936
3033_0540	AUDIOMIX_SAI7_TXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI7_TXSYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.332/ 1937
3033_0544	AUDIOMIX_EARC_PHY_SPDIF_IN_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_EARC_PHY_SPDIF_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.333/ 1937
3033_0548	AUDIOMIX_SPDIF_EXTCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SPDIF_EXTCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.334/ 1938
3033_054C	CAN1_CANRX_SELECT_INPUT DAISY Register (IOMUXC_CAN1_CANRX_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.335/ 1939
3033_0550	CAN2_CANRX_SELECT_INPUT DAISY Register (IOMUXC_CAN2_CANRX_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.336/ 1939
3033_0554	CCM_GPC_PMIC_VFUNCTIONAL_READY_SELECT_INPUT DAISY Register	32	R/W	0000_0000h	8.2.4.337/ 1940

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
	(IOMUXC_CCM_GPC_PMIC_VFUNCTIONAL_READY_SELECT_INPUT)				
3033_0558	ECSPI1_CSPI_CLK_IN_SELECT_INPUT DAISY Register (IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.338/ 1941
3033_055C	ECSPI1_MISO_SELECT_INPUT DAISY Register (IOMUXC_ECSPI1_MISO_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.339/ 1942
3033_0560	ECSPI1_MOSI_SELECT_INPUT DAISY Register (IOMUXC_ECSPI1_MOSI_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.340/ 1943
3033_0564	ECSPI1_SS_B_SELECT_INPUT_0 DAISY Register (IOMUXC_ECSPI1_SS_B_SELECT_INPUT_0)	32	R/W	0000_0000h	8.2.4.341/ 1944
3033_0568	ECSPI2_CSPI_CLK_IN_SELECT_INPUT DAISY Register (IOMUXC_ECSPI2_CSPI_CLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.342/ 1944
3033_056C	ECSPI2_MISO_SELECT_INPUT DAISY Register (IOMUXC_ECSPI2_MISO_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.343/ 1945
3033_0570	ECSPI2_MOSI_SELECT_INPUT DAISY Register (IOMUXC_ECSPI2_MOSI_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.344/ 1946
3033_0574	ECSPI2_SS_B_SELECT_INPUT_0 DAISY Register (IOMUXC_ECSPI2_SS_B_SELECT_INPUT_0)	32	R/W	0000_0000h	8.2.4.345/ 1946
3033_0578	ENET1_IPG_CLK_RMII_SELECT_INPUT DAISY Register (IOMUXC_ENET1_IPG_CLK_RMII_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.346/ 1947
3033_057C	ENET1_MDIO_SELECT_INPUT DAISY Register (IOMUXC_ENET1_MDIO_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.347/ 1948
3033_0580	ENET1_RXDATA_0_SELECT_INPUT DAISY Register (IOMUXC_ENET1_RXDATA_0_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.348/ 1949
3033_0584	ENET1_RXDATA_1_SELECT_INPUT DAISY Register (IOMUXC_ENET1_RXDATA_1_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.349/ 1950
3033_0588	ENET1_RXEN_SELECT_INPUT DAISY Register (IOMUXC_ENET1_RXEN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.350/ 1951
3033_058C	ENET1_RXERR_SELECT_INPUT DAISY Register (IOMUXC_ENET1_RXERR_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.351/ 1952
3033_0590	ENET_QOS_GMII_MDI_I_SELECT_INPUT DAISY Register (IOMUXC_ENET_QOS_GMII_MDI_I_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.352/ 1952
3033_0594	GPT1_CAPIN1_SELECT_INPUT DAISY Register (IOMUXC_GPT1_CAPIN1_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.353/ 1953
3033_0598	GPT1_CAPIN2_SELECT_INPUT DAISY Register (IOMUXC_GPT1_CAPIN2_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.354/ 1954
3033_059C	GPT1_CLKIN_SELECT_INPUT DAISY Register (IOMUXC_GPT1_CLKIN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.355/ 1955
3033_05A0	PCIE_CLKREQ_B_SELECT_INPUT DAISY Register (IOMUXC_PCIE_CLKREQ_B_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.356/ 1956
3033_05A4	I2C1_SCL_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C1_SCL_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.357/ 1956
3033_05A8	I2C1_SDA_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C1_SDA_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.358/ 1957

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_05AC	I2C2_SCL_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C2_SCL_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.359/ 1958
3033_05B0	I2C2_SDA_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C2_SDA_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.360/ 1958
3033_05B4	I2C3_SCL_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C3_SCL_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.361/ 1959
3033_05B8	I2C3_SDA_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C3_SDA_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.362/ 1960
3033_05BC	I2C4_SCL_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C4_SCL_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.363/ 1960
3033_05C0	I2C4_SDA_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C4_SDA_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.364/ 1961
3033_05C4	I2C5_SCL_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C5_SCL_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.365/ 1962
3033_05C8	I2C5_SDA_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C5_SDA_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.366/ 1962
3033_05CC	I2C6_SCL_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C6_SCL_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.367/ 1963
3033_05D0	I2C6_SDA_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C6_SDA_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.368/ 1964
3033_05D4	ISP_FL_TRIG_0_SELECT_INPUT DAISY Register (IOMUXC_ISP_FL_TRIG_0_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.369/ 1965
3033_05D8	ISP_FL_TRIG_1_SELECT_INPUT DAISY Register (IOMUXC_ISP_FL_TRIG_1_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.370/ 1966
3033_05DC	ISP_SHUTTER_TRIG_0_SELECT_INPUT DAISY Register (IOMUXC_ISP_SHUTTER_TRIG_0_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.371/ 1967
3033_05E0	ISP_SHUTTER_TRIG_1_SELECT_INPUT DAISY Register (IOMUXC_ISP_SHUTTER_TRIG_1_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.372/ 1968
3033_05E4	UART1_UART_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART1_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.373/ 1968
3033_05E8	UART1_UART_RXD_MUX_SELECT_INPUT DAISY Register (IOMUXC_UART1_UART_RXD_MUX_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.374/ 1969
3033_05EC	UART2_UART_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART2_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.375/ 1970
3033_05F0	UART2_UART_RXD_MUX_SELECT_INPUT DAISY Register (IOMUXC_UART2_UART_RXD_MUX_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.376/ 1970
3033_05F4	UART3_UART_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART3_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.377/ 1971
3033_05F8	UART3_UART_RXD_MUX_SELECT_INPUT DAISY Register (IOMUXC_UART3_UART_RXD_MUX_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.378/ 1972
3033_05FC	UART4_UART_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART4_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.379/ 1972

Table continues on the next page...

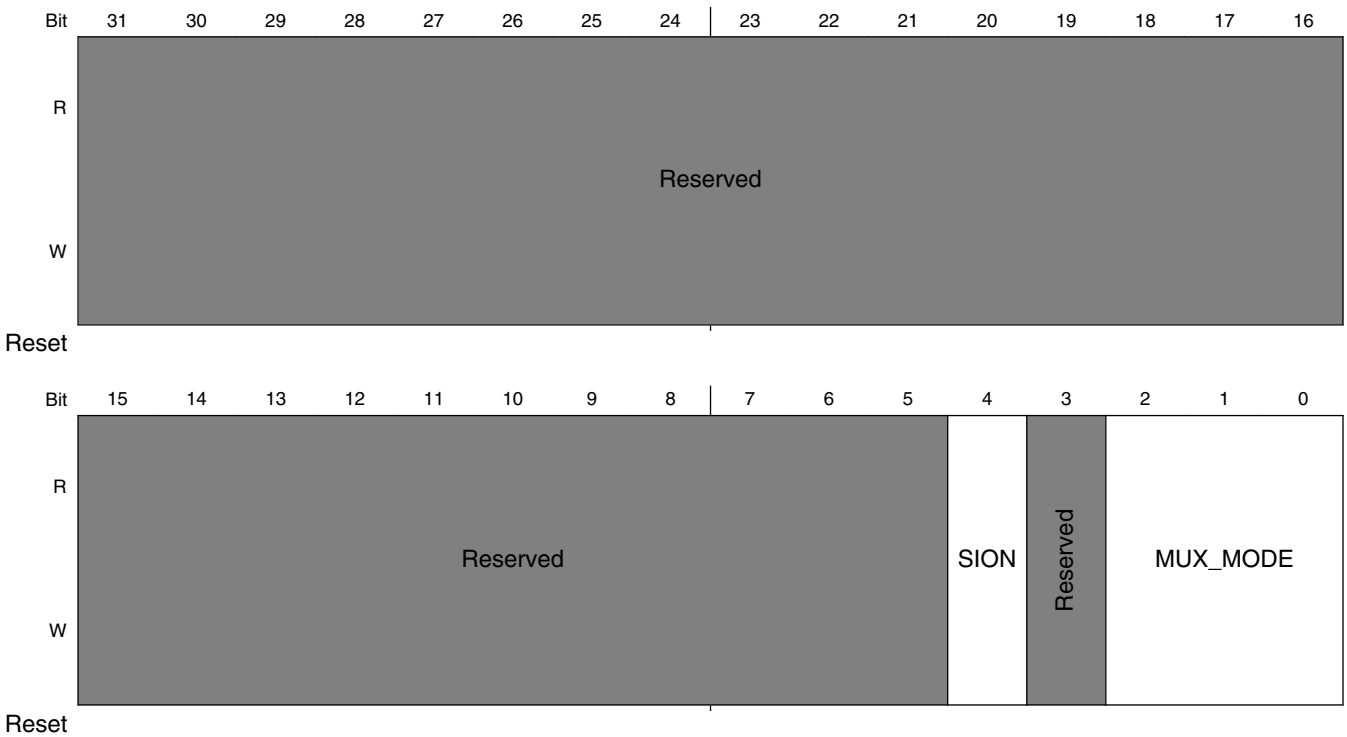
IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0600	UART4_UART_RXD_MUX_SELECT_INPUT DAISY Register (IOMUXC_UART4_UART_RXD_MUX_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.380/ 1973
3033_0604	USDHC3_CARD_CLK_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_CARD_CLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.381/ 1974
3033_0608	USDHC3_CARD_DET_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_CARD_DET_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.382/ 1974
3033_060C	USDHC3_CMD_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_CMD_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.383/ 1975
3033_0610	USDHC3_DAT0_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT0_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.384/ 1976
3033_0614	USDHC3_DAT1_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT1_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.385/ 1977
3033_0618	USDHC3_DAT2_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT2_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.386/ 1978
3033_061C	USDHC3_DAT3_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT3_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.387/ 1979
3033_0620	USDHC3_DAT4_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT4_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.388/ 1980
3033_0624	USDHC3_DAT5_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT5_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.389/ 1981
3033_0628	USDHC3_DAT6_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT6_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.390/ 1982
3033_062C	USDHC3_DAT7_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT7_IN_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.391/ 1983
3033_0630	USDHC3_STROBE_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_STROBE_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.392/ 1984
3033_0634	USDHC3_WP_ON_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_WP_ON_SELECT_INPUT)	32	R/W	0000_0000h	8.2.4.393/ 1984

8.2.4.1 SW_MUX_CTL_PAD_GPIO1_IO00 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO00)

SW_MUX_CTL Register

Address: 3033_0000h base + 14h offset = 3033_0014h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO00 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO00 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO1_IO00. 000 **ALT0_GPIO1_IO[0]** — Select mux mode: ALT0 mux port: GPIO1_IO00 of instance: gpio1 001 **ALT1_CCM_ENET_PHY_REF_CLK_ROOT** — Select mux mode: ALT1 mux port: CCM_ENET_PHY_REF_CLK_ROOT of instance: ccm

Table continues on the next page...

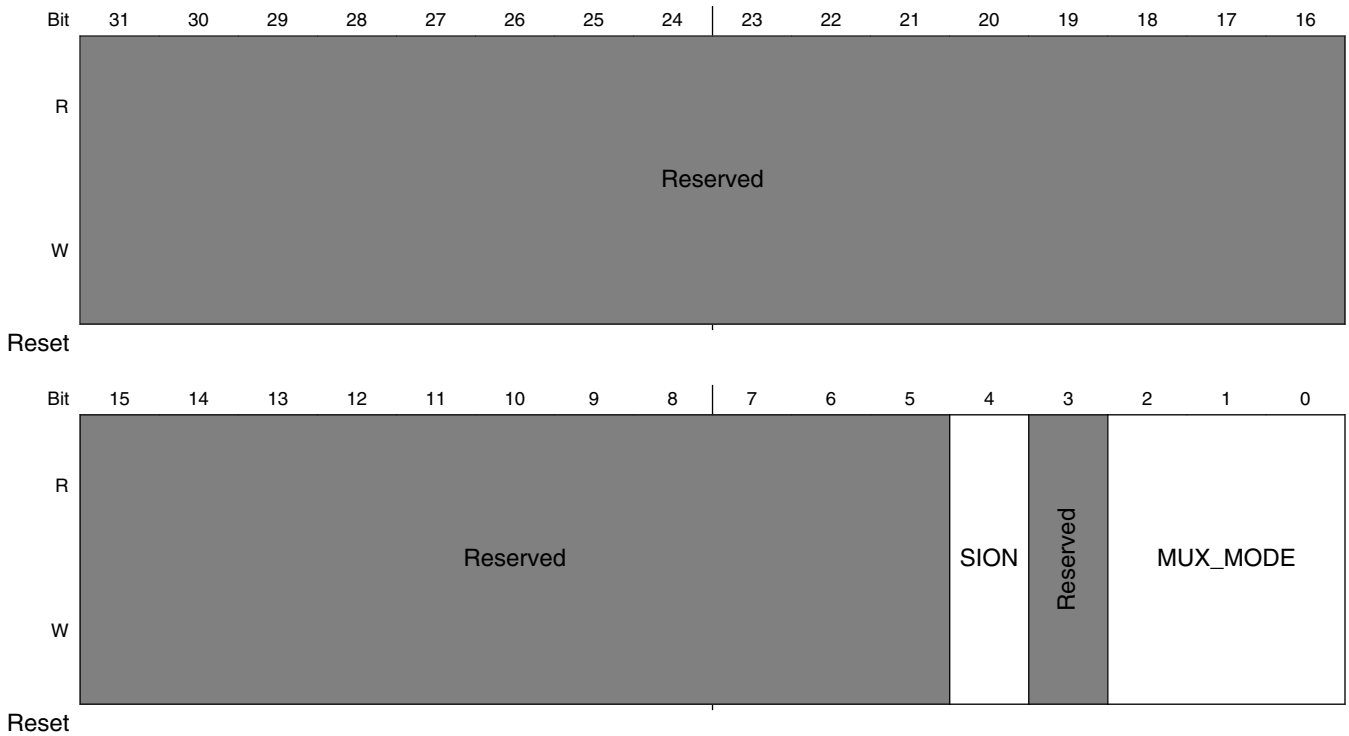
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO00 field descriptions (continued)

Field	Description
	011 **ALT3_ISP_FL_TRIG_0** — Select mux mode: ALT3 mux port: ISP_FL_TRIG_0 of instance: isp 101 **ALT5_REF_CLK_32K** — Select mux mode: ALT5 mux port: REF_CLK_32K of instance: anamix 110 **ALT6_CCM_EXT_CLK1** — Select mux mode: ALT6 mux port: CCM_EXT_CLK1 of instance: ccm

8.2.4.2 SW_MUX_CTL_PAD_GPIO1_IO01 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO01)

SW_MUX_CTL Register

Address: 3033_0000h base + 18h offset = 3033_0018h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO01 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO01 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

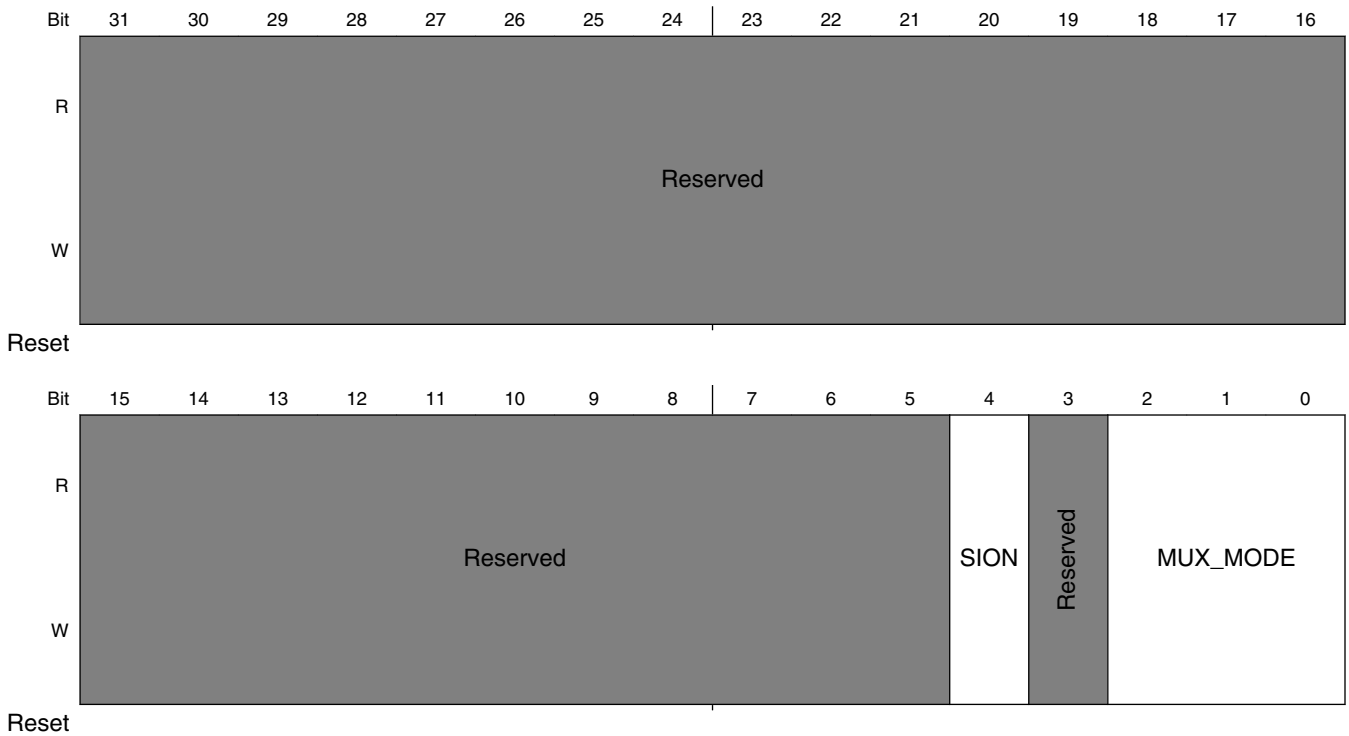
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO01 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO1_IO01. 000 **ALT0_GPIO1_IO[1]** — Select mux mode: ALT0 mux port: GPIO1_IO01 of instance: gpio1 001 **ALT1_PWM1_OUT** — Select mux mode: ALT1 mux port: PWM1_OUT of instance: pwm1 011 **ALT3_ISP_SHUTTER_TRIG_0** — Select mux mode: ALT3 mux port: ISP_SHUTTER_TRIG_0 of instance: isp 101 **ALT5_REF_CLK_24M** — Select mux mode: ALT5 mux port: REF_CLK_24M of instance: anamix 110 **ALT6_CCM_EXT_CLK2** — Select mux mode: ALT6 mux port: CCM_EXT_CLK2 of instance: ccm

8.2.4.3 SW_MUX_CTL_PAD_GPIO1_IO02 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO02)

SW_MUX_CTL Register

Address: 3033_0000h base + 1Ch offset = 3033_001Ch



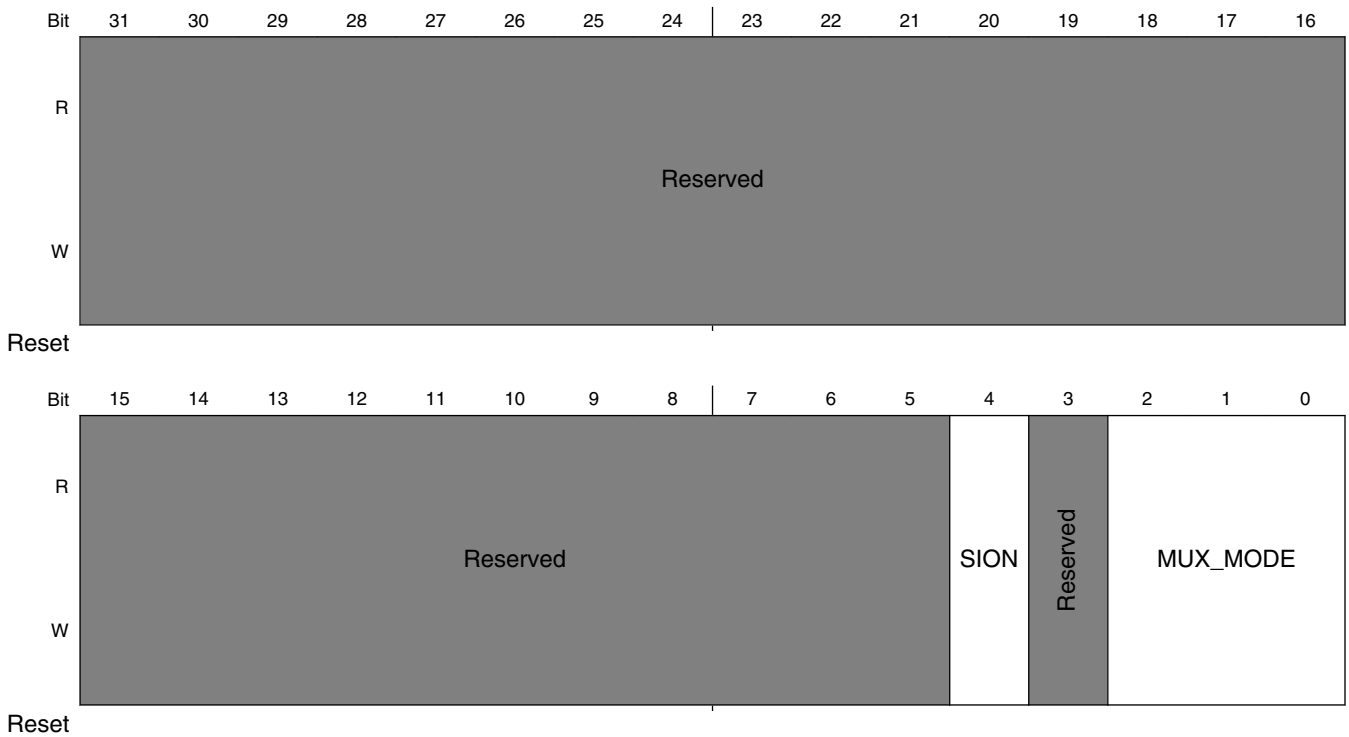
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO02 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO02 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO1_IO02. 000 **ALT0_GPIO1_IO[2]** — Select mux mode: ALT0 mux port: GPIO1_IO02 of instance: gpio1 001 **ALT1_WDOG1_WDOG_B** — Select mux mode: ALT1 mux port: WDOG1_WDOG_B of instance: wdog1 011 **ALT3_ISP_FLASH_TRIG_0** — Select mux mode: ALT3 mux port: ISP_FLASH_TRIG_0 of instance: isp 101 **ALT5_WDOG1_WDOG_ANY** — Select mux mode: ALT5 mux port: WDOG1_WDOG_ANY of instance: wdog1 111 **ALT7_SJC_DE_B** — Select mux mode: ALT7 mux port: SJC_DE_B of instance: sjc

8.2.4.4 SW_MUX_CTL_PAD_GPIO1_IO03 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO03)

SW_MUX_CTL Register

Address: 3033_0000h base + 20h offset = 3033_0020h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO03 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO03 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO1_IO03. 000 **ALT0_GPIO1_IO[3]** — Select mux mode: ALT0 mux port: GPIO1_IO03 of instance: gpio1 001 **ALT1_USDHC1_VSELECT** — Select mux mode: ALT1 mux port: USDHC1_VSELECT of instance: usdhc1

Table continues on the next page...

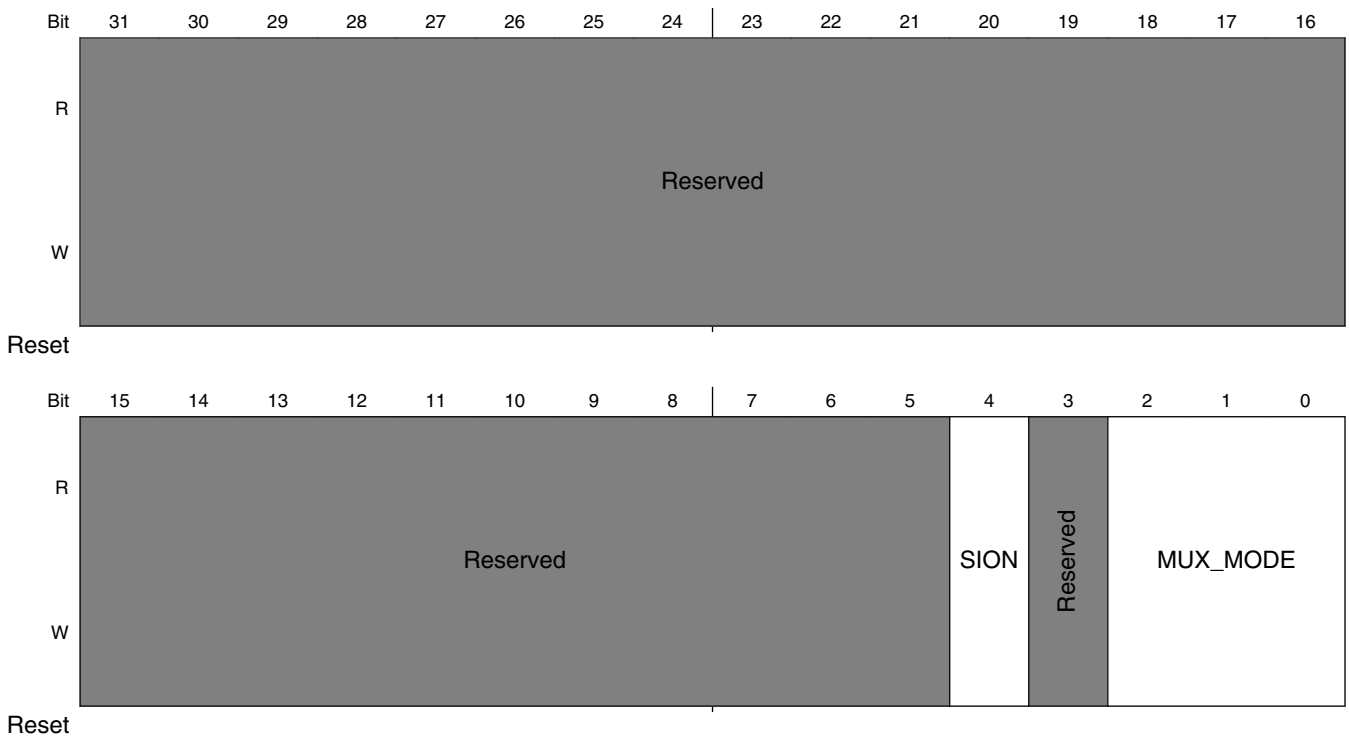
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO03 field descriptions (continued)

Field	Description
	011 **ALT3_ISP_PRELIGHT_TRIG_0** — Select mux mode: ALT3 mux port: ISP_PRELIGHT_TRIG_0 of instance: isp 101 **ALT5_SDMA1_EXT_EVENT[0]** — Select mux mode: ALT5 mux port: SDMA1_EXT_EVENT00 of instance: sdma1

8.2.4.5 SW_MUX_CTL_PAD_GPIO1_IO04 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO04)

SW_MUX_CTL Register

Address: 3033_0000h base + 24h offset = 3033_0024h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO04 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO04 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

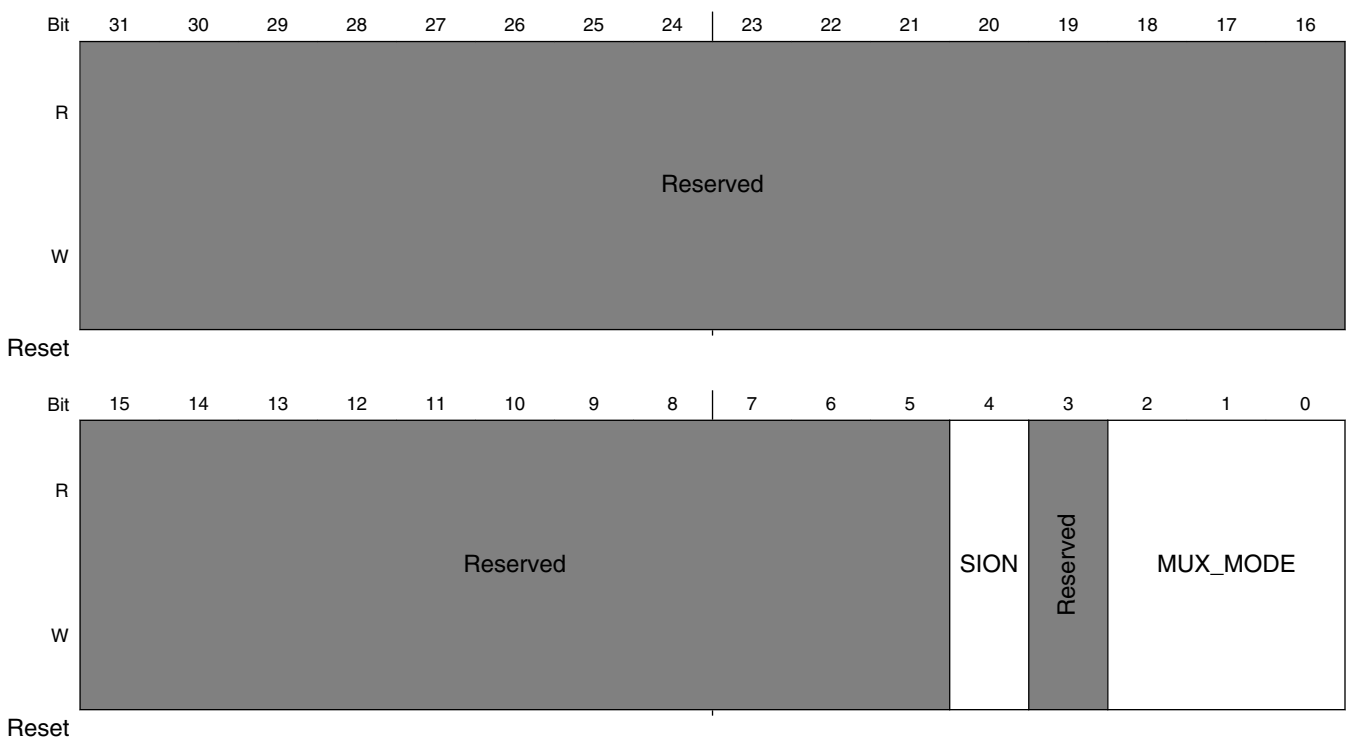
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO04 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO1_IO04. 000 **ALT0_GPIO1_IO[4]** — Select mux mode: ALT0 mux port: GPIO1_IO04 of instance: gpio1 001 **ALT1_USDHC2_VSELECT** — Select mux mode: ALT1 mux port: USDHC2_VSELECT of instance: usdhc2 011 **ALT3_ISP_SHUTTER_OPEN_0** — Select mux mode: ALT3 mux port: ISP_SHUTTER_OPEN_0 of instance: isp 101 **ALT5_SDMA1_EXT_EVENT[1]** — Select mux mode: ALT5 mux port: SDMA1_EXT_EVENT01 of instance: sdma1

8.2.4.6 SW_MUX_CTL_PAD_GPIO1_IO05 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO05)

SW_MUX_CTL Register

Address: 3033_0000h base + 28h offset = 3033_0028h



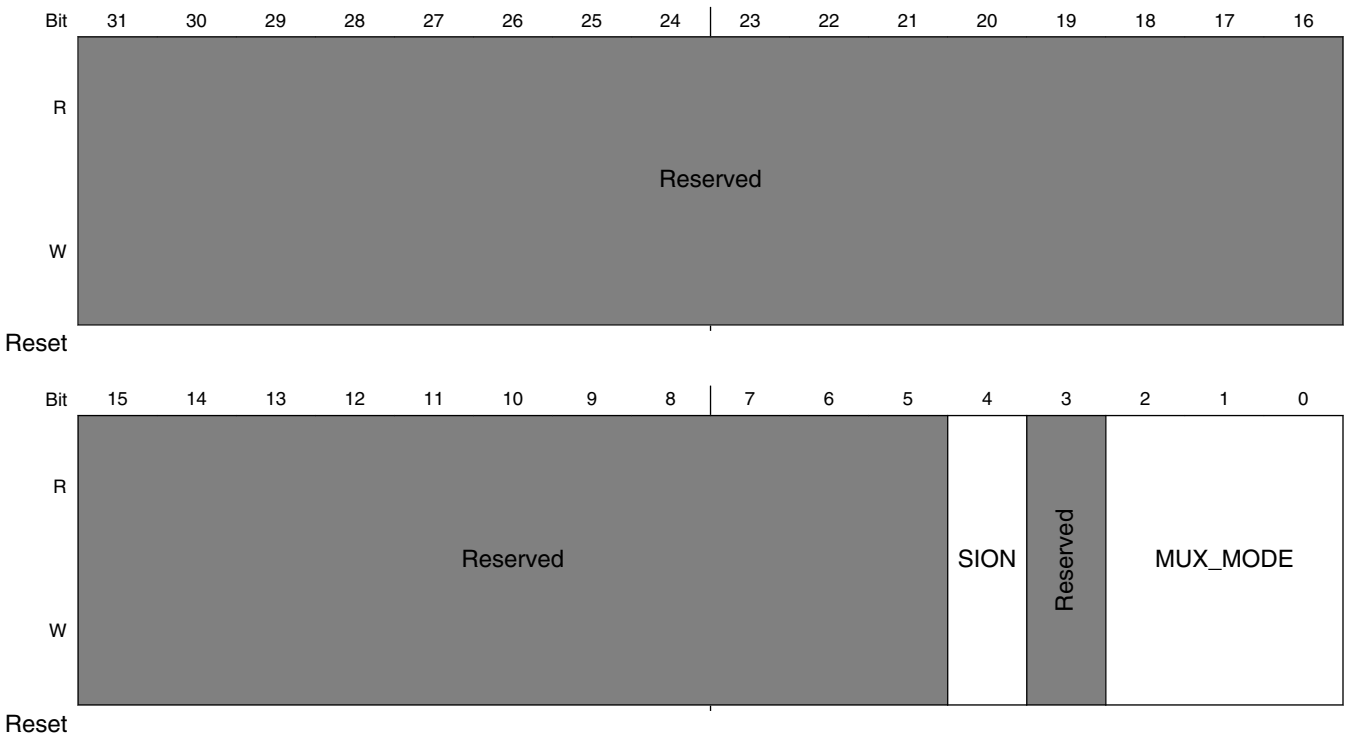
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO05 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO05 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO1_IO05. 000 **ALT0_GPIO1_IO[5]** — Select mux mode: ALT0 mux port: GPIO1_IO05 of instance: gpio1 001 **ALT1_M7_NMI** — Select mux mode: ALT1 mux port: M7_NMI of instance: m7 011 **ALT3_ISP_FL_TRIG_1** — Select mux mode: ALT3 mux port: ISP_FL_TRIG_1 of instance: isp 101 **ALT5_CCM_PMIC_READY** — Select mux mode: ALT5 mux port: CCM_PMIC_READY of instance: ccm

8.2.4.7 SW_MUX_CTL_PAD_GPIO1_IO06 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO06)

SW_MUX_CTL Register

Address: 3033_0000h base + 2Ch offset = 3033_002Ch



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO06 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO06 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO1_IO06. 000 **ALT0_GPIO1_IO[6]** — Select mux mode: ALT0 mux port: GPIO1_IO06 of instance: gpio1 001 **ALT1_ENET_QOS_MDC** — Select mux mode: ALT1 mux port: ENET_QOS_MDC of instance: enet_qos

Table continues on the next page...

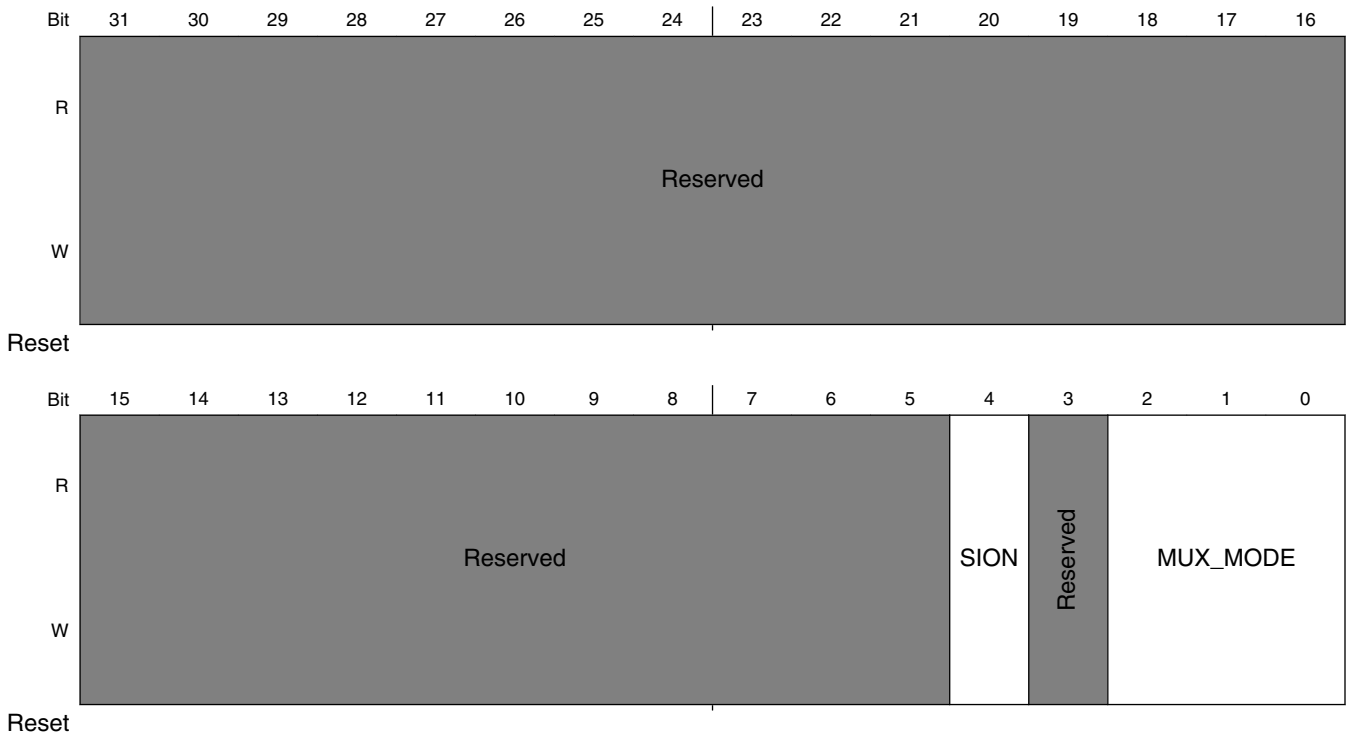
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO06 field descriptions (continued)

Field	Description
	011 **ALT3_ISP_SHUTTER_TRIG_1** — Select mux mode: ALT3 mux port: ISP_SHUTTER_TRIG_1 of instance: isp 101 **ALT5_USDHC1_CD_B** — Select mux mode: ALT5 mux port: USDHC1_CD_B of instance: usdhc1 110 **ALT6_CCM_EXT_CLK3** — Select mux mode: ALT6 mux port: CCM_EXT_CLK3 of instance: ccm

8.2.4.8 SW_MUX_CTL_PAD_GPIO1_IO07 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO07)

SW_MUX_CTL Register

Address: 3033_0000h base + 30h offset = 3033_0030h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO07 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO07 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

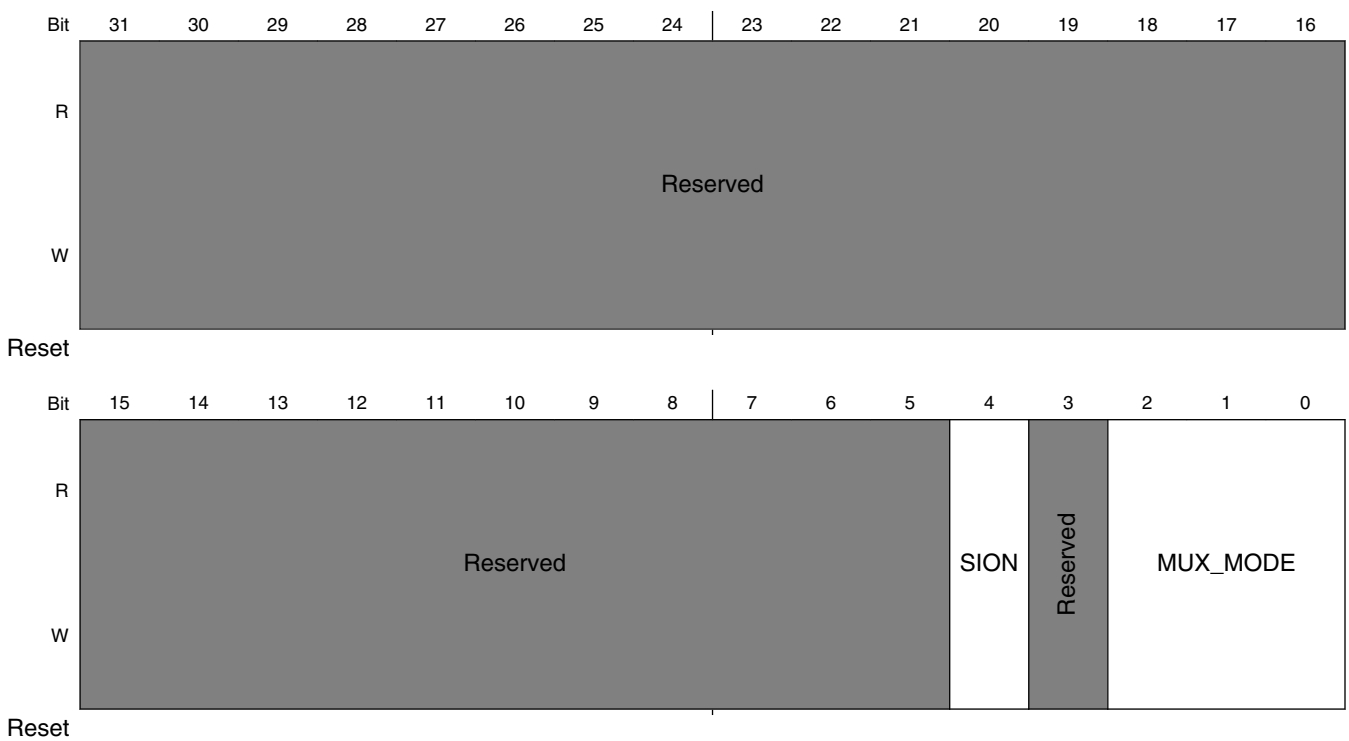
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO07 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO1_IO07. 000 **ALT0_GPIO1_IO[7]** — Select mux mode: ALT0 mux port: GPIO1_IO07 of instance: gpio1 001 **ALT1_ENET_QOS_MDIO** — Select mux mode: ALT1 mux port: ENET_QOS_MDIO of instance: enet_qos 011 **ALT3_ISP_FLASH_TRIG_1** — Select mux mode: ALT3 mux port: ISP_FLASH_TRIG_1 of instance: isp 101 **ALT5_USDHC1_WP** — Select mux mode: ALT5 mux port: USDHC1_WP of instance: usdhc1 110 **ALT6_CCM_EXT_CLK4** — Select mux mode: ALT6 mux port: CCM_EXT_CLK4 of instance: ccm

8.2.4.9 SW_MUX_CTL_PAD_GPIO1_IO08 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO08)

SW_MUX_CTL Register

Address: 3033_0000h base + 34h offset = 3033_0034h



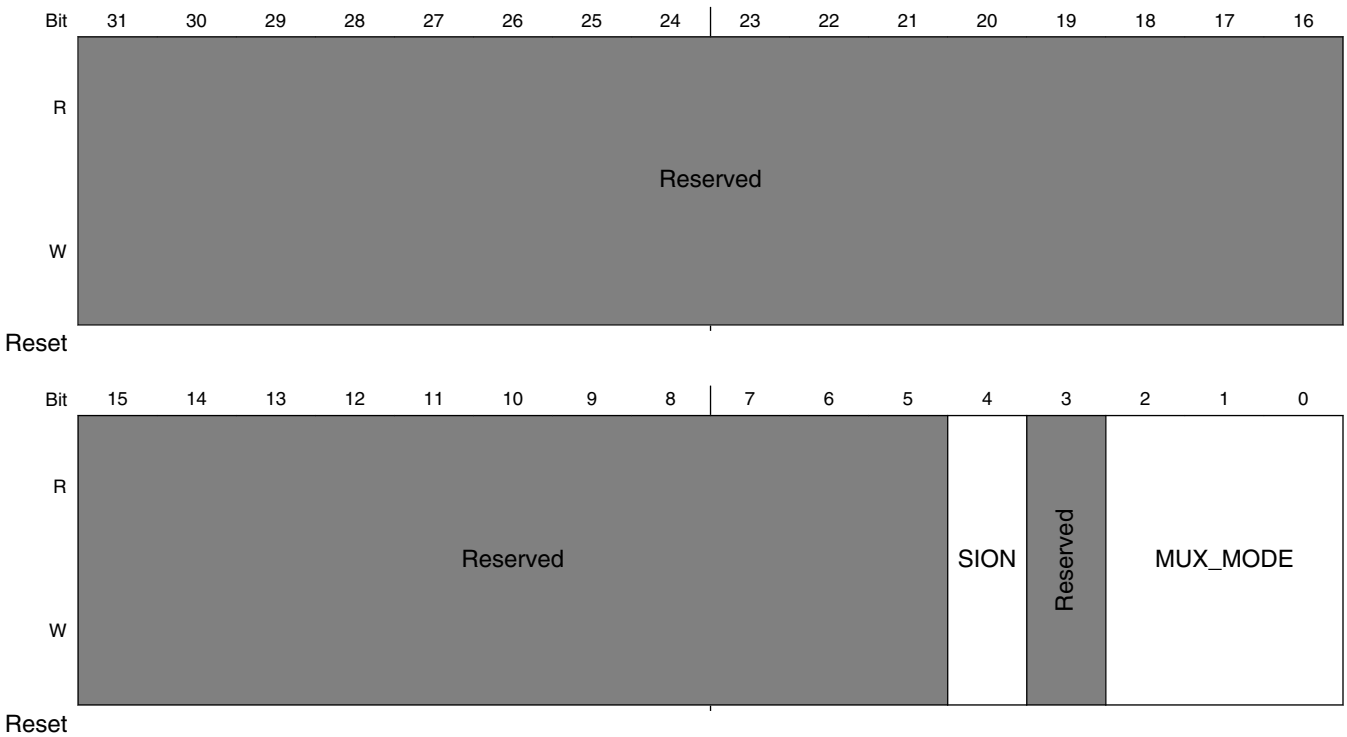
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO08 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO08 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO1_IO08. 000 **ALT0_GPIO1_IO[8]** — Select mux mode: ALT0 mux port: GPIO1_IO08 of instance: gpio1 001 **ALT1_ENET_QOS_1588_EVENT0_IN** — Select mux mode: ALT1 mux port: ENET_QOS_1588_EVENT0_IN of instance: enet_qos 010 **ALT2_PWM1_OUT** — Select mux mode: ALT2 mux port: PWM1_OUT of instance: pwm1 011 **ALT3_ISP_PRELIGHT_TRIG_1** — Select mux mode: ALT3 mux port: ISP_PRELIGHT_TRIG_1 of instance: isp 100 **ALT4_ENET_QOS_1588_EVENT0_AUX_IN** — Select mux mode: ALT4 mux port: ENET_QOS_1588_EVENT0_AUX_IN of instance: enet_qos 101 **ALT5_USDHC2_RESET_B** — Select mux mode: ALT5 mux port: USDHC2_RESET_B of instance: usdhc2

8.2.4.10 SW_MUX_CTL_PAD_GPIO1_IO09 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO09)

SW_MUX_CTL Register

Address: 3033_0000h base + 38h offset = 3033_0038h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO09 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO09 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO1_IO09. 000 **ALT0_GPIO1_IO[9]** — Select mux mode: ALT0 mux port: GPIO1_IO09 of instance: gpio1 001 **ALT1_ENET_QOS_1588_EVENT0_OUT** — Select mux mode: ALT1 mux port: ENET_QOS_1588_EVENT0_OUT of instance: enet_qos

Table continues on the next page...

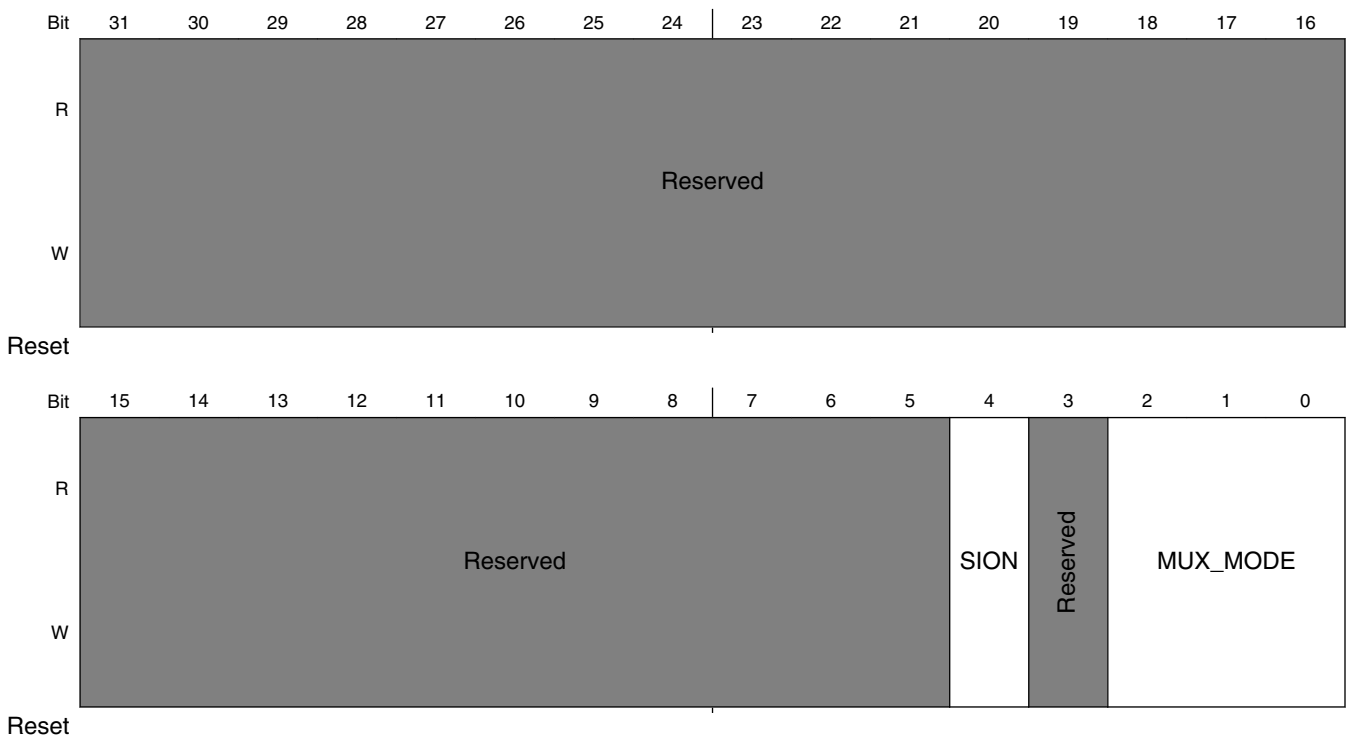
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO09 field descriptions (continued)

Field	Description
	010 **ALT2_PWM2_OUT** — Select mux mode: ALT2 mux port: PWM2_OUT of instance: pwm2 011 **ALT3_ISP_SHUTTER_OPEN_1** — Select mux mode: ALT3 mux port: ISP_SHUTTER_OPEN_1 of instance: isp 100 **ALT4_USDHC3_RESET_B** — Select mux mode: ALT4 mux port: USDHC3_RESET_B of instance: usdhc3 101 **ALT5_SDMA2_EXT_EVENT[0]** — Select mux mode: ALT5 mux port: SDMA2_EXT_EVENT00 of instance: SDMA2

8.2.4.11 SW_MUX_CTL_PAD_GPIO1_IO10 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO10)

SW_MUX_CTL Register

Address: 3033_0000h base + 3Ch offset = 3033_003Ch



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO10 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality.

Table continues on the next page...

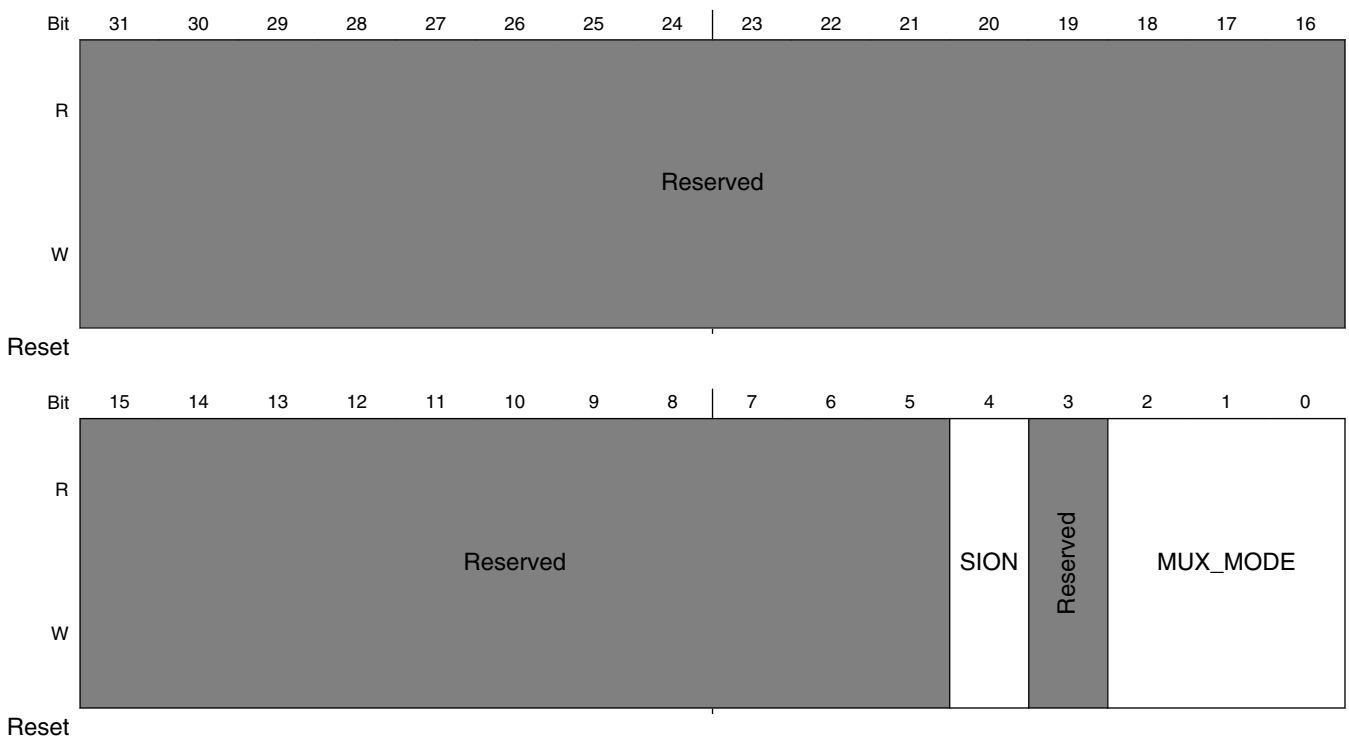
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO10 field descriptions (continued)

Field	Description
	1 **ENABLED** — Force input path of pad GPIO1_IO10 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: GPIO1_IO10. 000 **ALT0_GPIO1_IO[10]** — Select mux mode: ALT0 mux port: GPIO1_IO10 of instance: gpio1 010 **ALT2_PWM3_OUT** — Select mux mode: ALT2 mux port: PWM3_OUT of instance: pwm3

8.2.4.12 SW_MUX_CTL_PAD_GPIO1_IO11 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO11)

SW_MUX_CTL Register

Address: 3033_0000h base + 40h offset = 3033_0040h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO11 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved

Table continues on the next page...

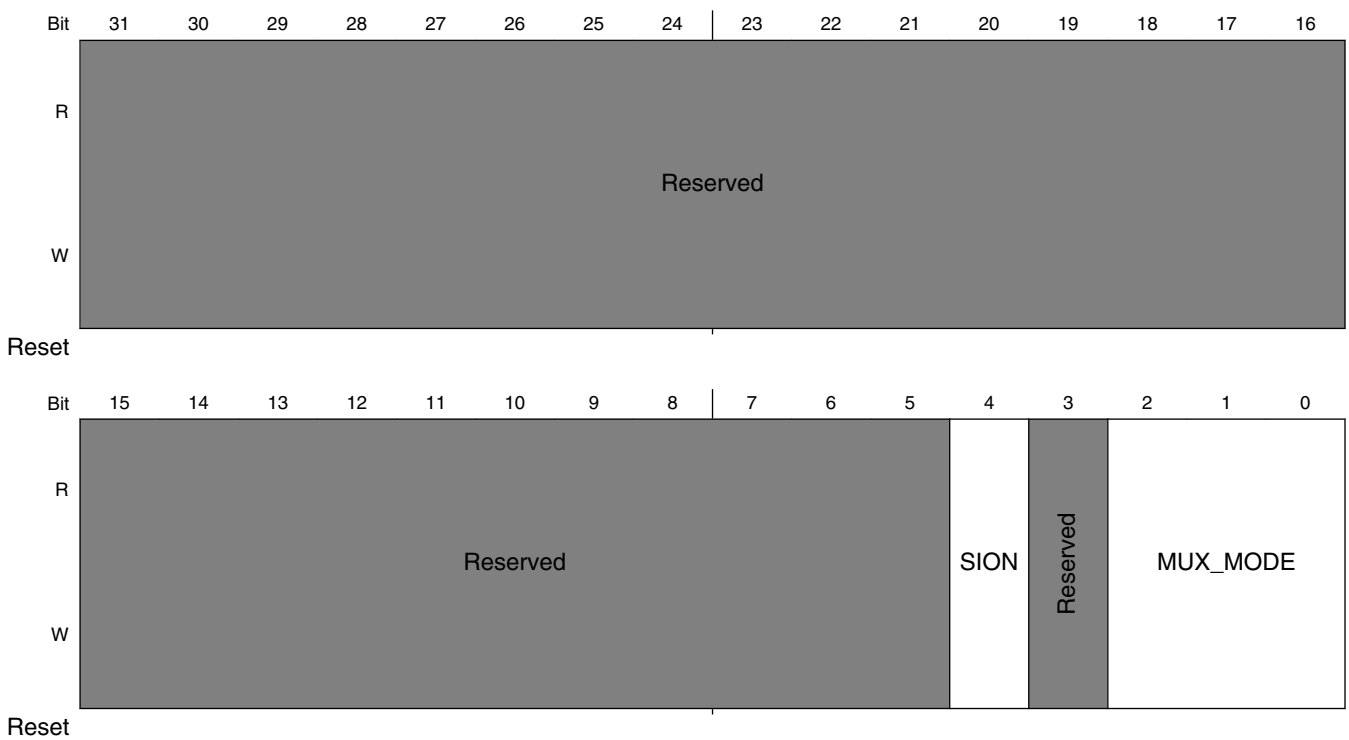
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO11 field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO11 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO1_IO11. 000 **ALT0_GPIO1_IO[11]** — Select mux mode: ALT0 mux port: GPIO1_IO11 of instance: gpio1 010 **ALT2_PWM2_OUT** — Select mux mode: ALT2 mux port: PWM2_OUT of instance: pwm2 100 **ALT4_USDHC3_VSELECT** — Select mux mode: ALT4 mux port: USDHC3_VSELECT of instance: usdhc3 101 **ALT5_CCM_PMIC_READY** — Select mux mode: ALT5 mux port: CCM_PMIC_READY of instance: ccm

8.2.4.13 SW_MUX_CTL_PAD_GPIO1_IO12 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO12)

SW_MUX_CTL Register

Address: 3033_0000h base + 44h offset = 3033_0044h



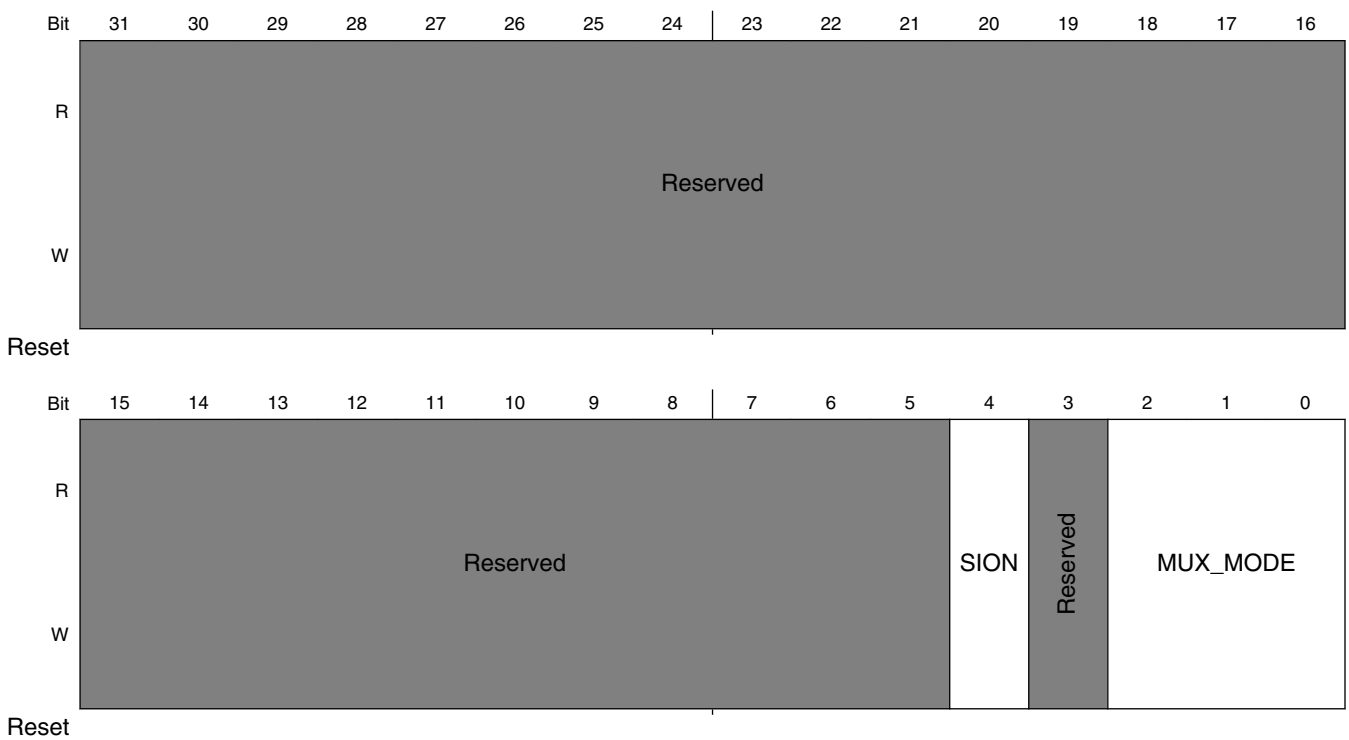
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO12 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO12 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO1_IO12. 000 **ALT0_GPIO1_IO[12]** — Select mux mode: ALT0 mux port: GPIO1_IO12 of instance: gpio1 001 **ALT1_USB1_PWR** — Select mux mode: ALT1 mux port: usb1_PWR of instance: usb1 101 **ALT5_SDMA2_EXT_EVENT[1]** — Select mux mode: ALT5 mux port: SDMA2_EXT_EVENT01 of instance: SDMA2

8.2.4.14 SW_MUX_CTL_PAD_GPIO1_IO13 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO13)

SW_MUX_CTL Register

Address: 3033_0000h base + 48h offset = 3033_0048h



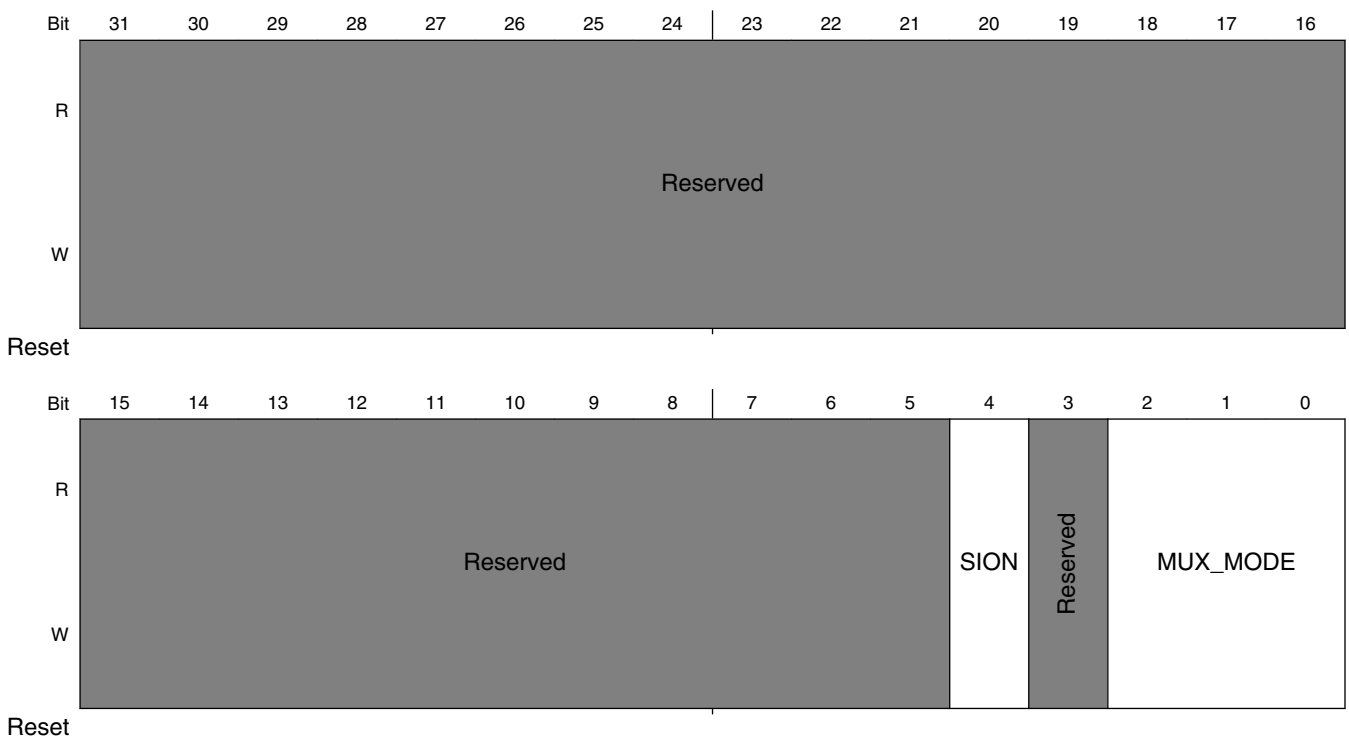
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO13 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO13 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO1_IO13. 000 **ALT0_GPIO1_IO[13]** — Select mux mode: ALT0 mux port: GPIO1_IO13 of instance: gpio1 001 **ALT1_USB1_OC** — Select mux mode: ALT1 mux port: usb1_OC of instance: usb1 101 **ALT5_PWM2_OUT** — Select mux mode: ALT5 mux port: PWM2_OUT of instance: pwm2

8.2.4.15 SW_MUX_CTL_PAD_GPIO1_IO14 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO14)

SW_MUX_CTL Register

Address: 3033_0000h base + 4Ch offset = 3033_004Ch



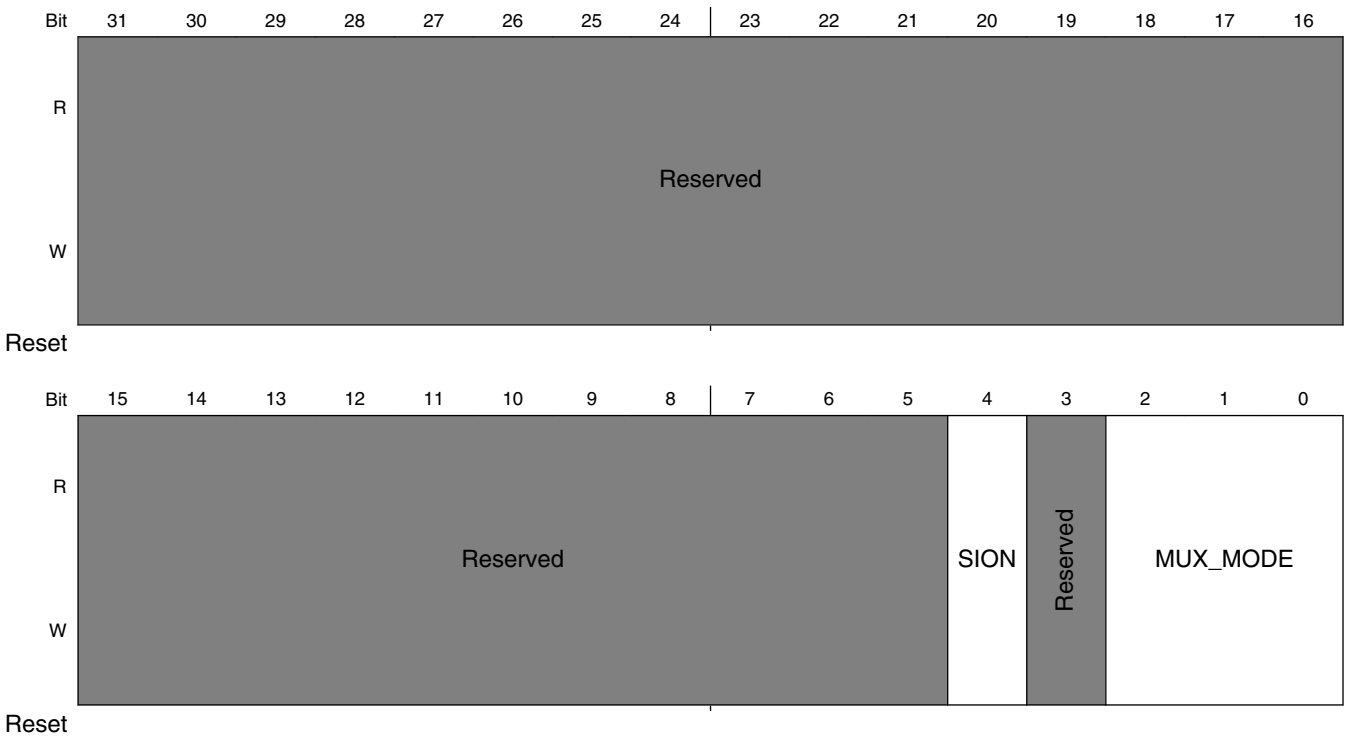
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO14 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO14 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO1_IO14. 000 **ALT0_GPIO1_IO[14]** — Select mux mode: ALT0 mux port: GPIO1_IO14 of instance: gpio1 001 **ALT1_USB2_PWR** — Select mux mode: ALT1 mux port: usb2_PWR of instance: usb2 100 **ALT4_USDHC3_CD_B** — Select mux mode: ALT4 mux port: USDHC3_CD_B of instance: usdhc3 101 **ALT5_PWM3_OUT** — Select mux mode: ALT5 mux port: PWM3_OUT of instance: pwm3 110 **ALT6_CCM_CLKO1** — Select mux mode: ALT6 mux port: CCM_CLKO1 of instance: ccm

8.2.4.16 SW_MUX_CTL_PAD_GPIO1_IO15 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO15)

SW_MUX_CTL Register

Address: 3033_0000h base + 50h offset = 3033_0050h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO15 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad GPIO1_IO15 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO1_IO15. 000 **ALT0_GPIO1_IO[15]** — Select mux mode: ALT0 mux port: GPIO1_IO15 of instance: gpio1 001 **ALT1_USB2_OC** — Select mux mode: ALT1 mux port: usb2_OC of instance: usb2 100 **ALT4_USDHC3_WP** — Select mux mode: ALT4 mux port: USDHC3_WP of instance: usdhc3

Table continues on the next page...

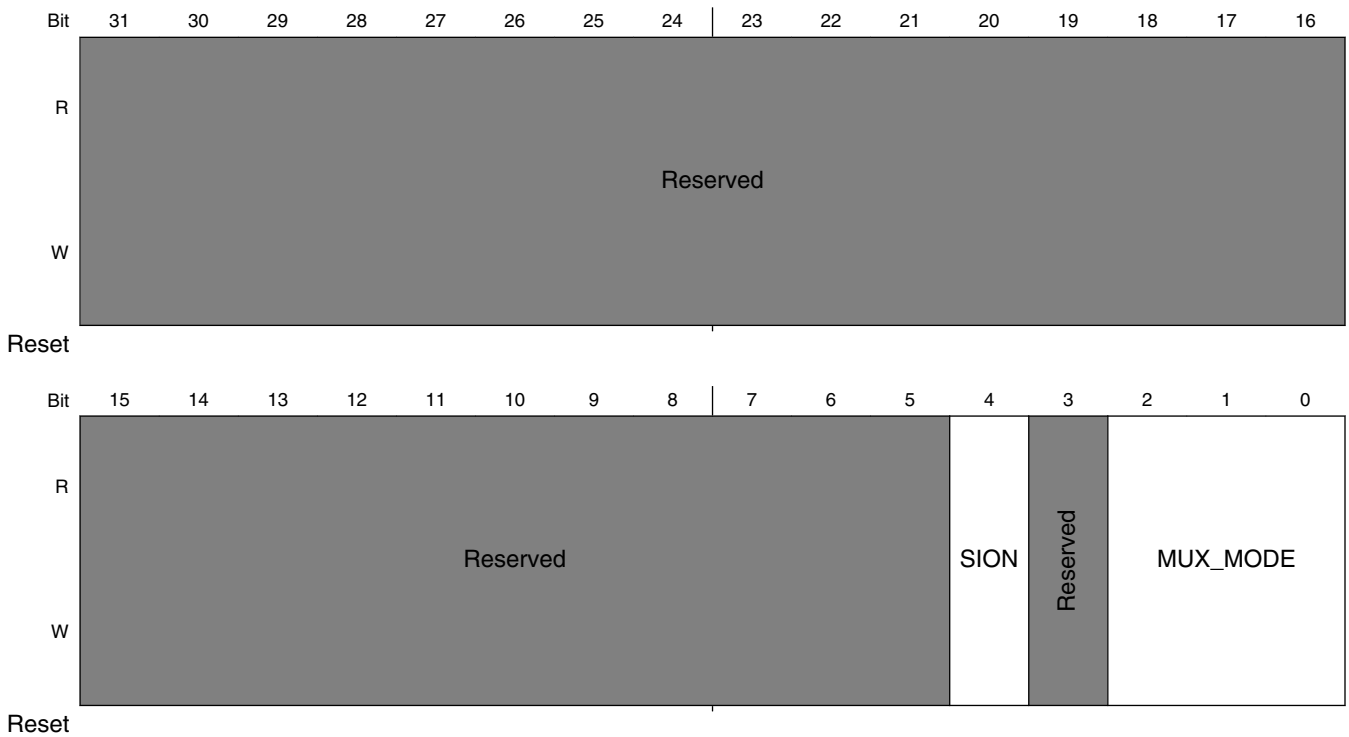
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO15 field descriptions (continued)

Field	Description
	101 **ALT5_PWM4_OUT** — Select mux mode: ALT5 mux port: PWM4_OUT of instance: pwm4 110 **ALT6_CCM_CLKO2** — Select mux mode: ALT6 mux port: CCM_CLKO2 of instance: ccm

8.2.4.17 SW_MUX_CTL_PAD_ENET_MDC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDC)

SW_MUX_CTL Register

Address: 3033_0000h base + 54h offset = 3033_0054h



IOMUXC_SW_MUX_CTL_PAD_ENET_MDC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_MDC 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved

Table continues on the next page...

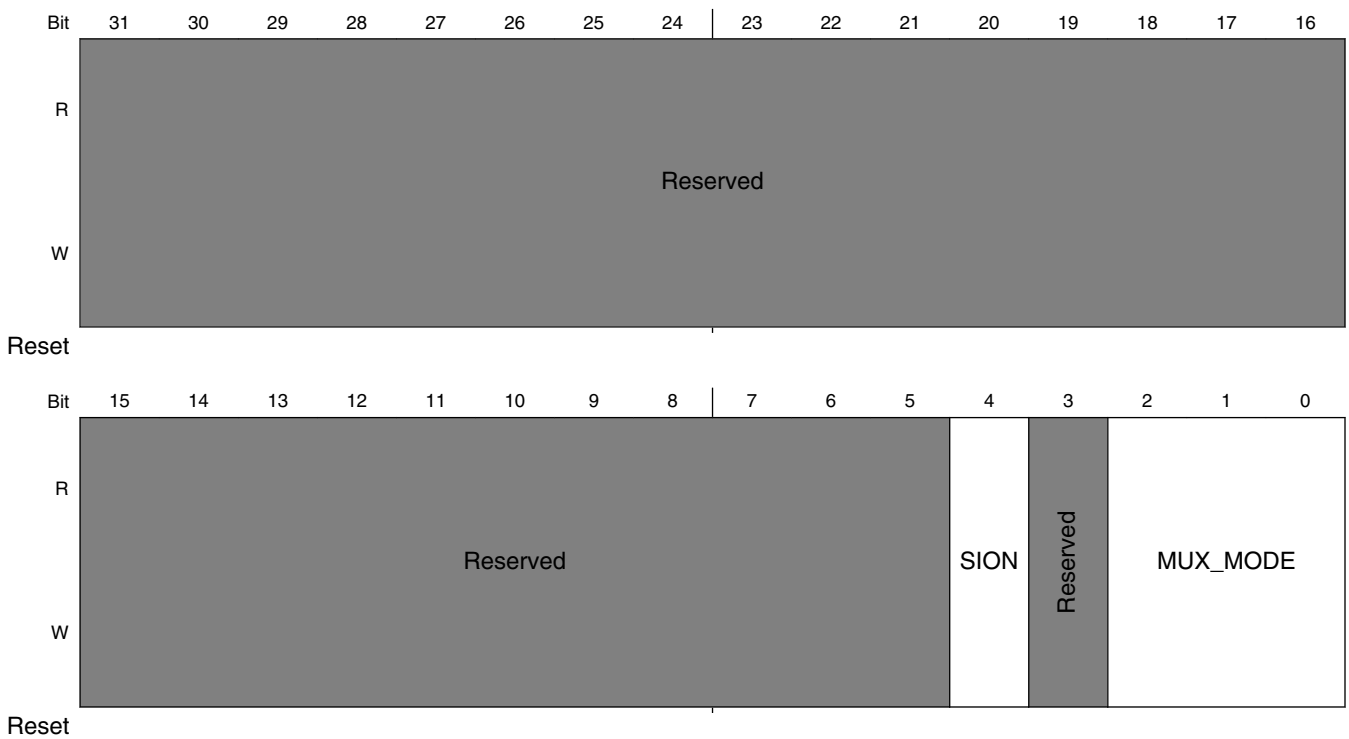
IOMUXC_SW_MUX_CTL_PAD_ENET_MDC field descriptions (continued)

Field	Description
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: ENET_MDC. 000 **ALT0_ENET_QOS_MDC** — Select mux mode: ALT0 mux port: ENET_QOS_MDC of instance: enet_qos 010 **ALT2_AUDIOMIX_SAI6_TX_DATA[0]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI6_TX_DATA00 of instance: sai6 101 **ALT5_GPIO1_IO[16]** — Select mux mode: ALT5 mux port: GPIO1_IO16 of instance: gpio1 110 **ALT6_USDHC3_STROBE** — Select mux mode: ALT6 mux port: USDHC3_STROBE of instance: usdhc3

8.2.4.18 SW_MUX_CTL_PAD_ENET_MDIO SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO)

SW_MUX_CTL Register

Address: 3033_0000h base + 58h offset = 3033_0058h



IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO field descriptions

Field	Description
31–5 -	This field is reserved. Reserved

Table continues on the next page...

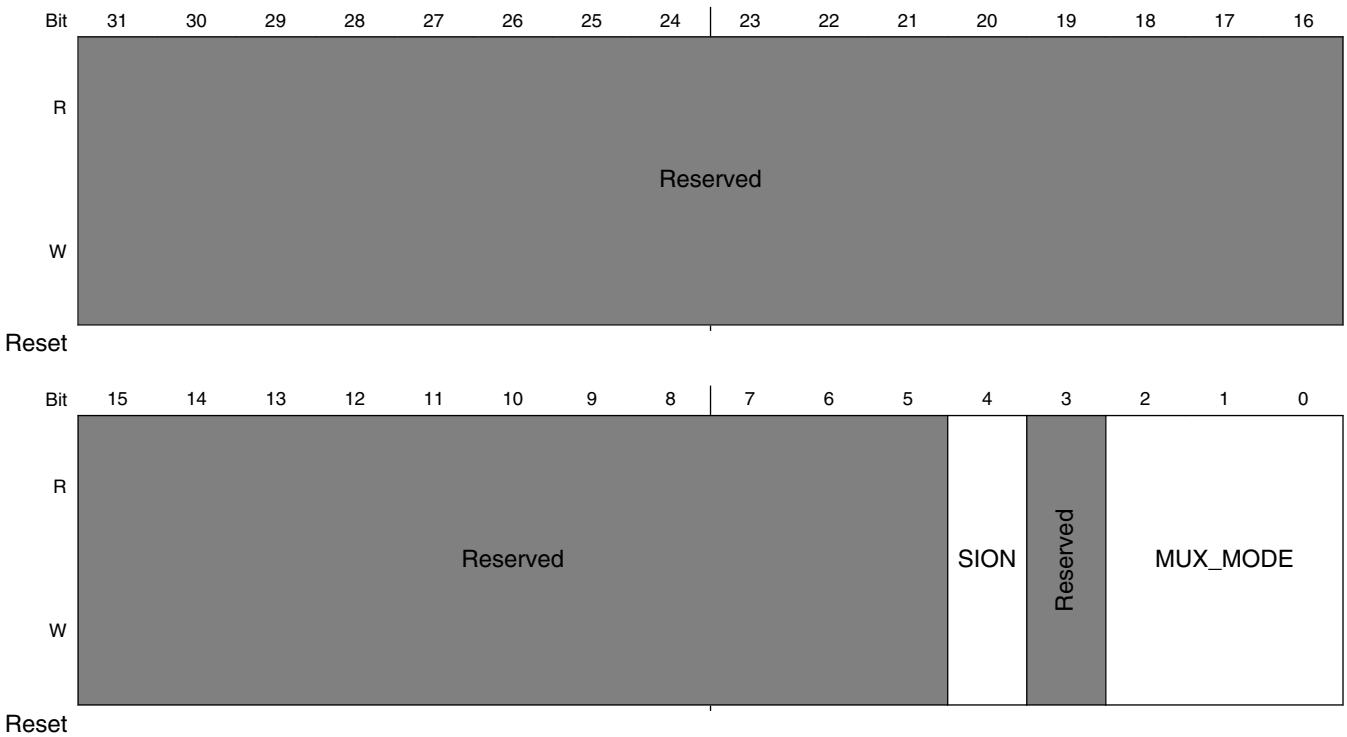
IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_MDIO 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET_MDIO. 000 **ALT0_ENET_QOS_MDIO** — Select mux mode: ALT0 mux port: ENET_QOS_MDIO of instance: enet_qos 010 **ALT2_AUDIOMIX_SAI6_TX_SYNC** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI6_TX_SYNC of instance: sai6 011 **ALT3_AUDIOMIX_PDM_BIT_STREAM[3]** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_BIT_STREAM03 of instance: pdm 101 **ALT5_GPIO1_IO[17]** — Select mux mode: ALT5 mux port: GPIO1_IO17 of instance: gpio1 110 **ALT6_USDHC3_DATA5** — Select mux mode: ALT6 mux port: USDHC3_DATA5 of instance: usdhc3

8.2.4.19 SW_MUX_CTL_PAD_ENET_TD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD3)

SW_MUX_CTL Register

Address: 3033_0000h base + 5Ch offset = 3033_005Ch



IOMUXC_SW_MUX_CTL_PAD_ENET_TD3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_TD3 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET_TD3. 000 **ALT0_ENET_QOS_RGMII_TD3** — Select mux mode: ALT0 mux port: ENET_QOS_RGMII_TD3 of instance: enet_qos 010 **ALT2_AUDIOMIX_SAI6_TX_BCLK** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI6_TX_BCLK of instance: sai6

Table continues on the next page...

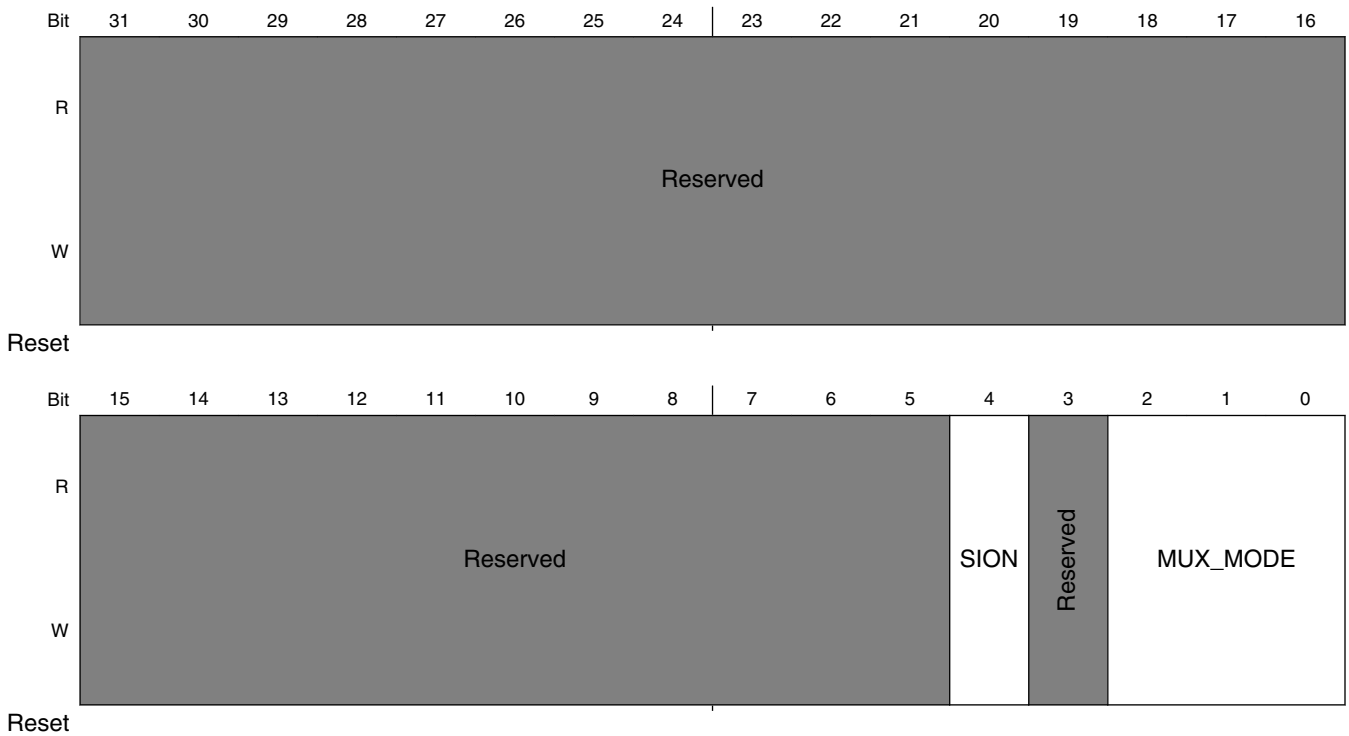
IOMUXC_SW_MUX_CTL_PAD_ENET_TD3 field descriptions (continued)

Field	Description
	011 **ALT3_AUDIOMIX_PDM_BIT_STREAM[2]** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_BIT_STREAM02 of instance: pdm 101 **ALT5_GPIO1_IO[18]** — Select mux mode: ALT5 mux port: GPIO1_IO18 of instance: gpio1 110 **ALT6_USDHC3_DATA6** — Select mux mode: ALT6 mux port: USDHC3_DATA6 of instance: usdhc3

8.2.4.20 SW_MUX_CTL_PAD_ENET_TD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD2)

SW_MUX_CTL Register

Address: 3033_0000h base + 60h offset = 3033_0060h



IOMUXC_SW_MUX_CTL_PAD_ENET_TD2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_TD2 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

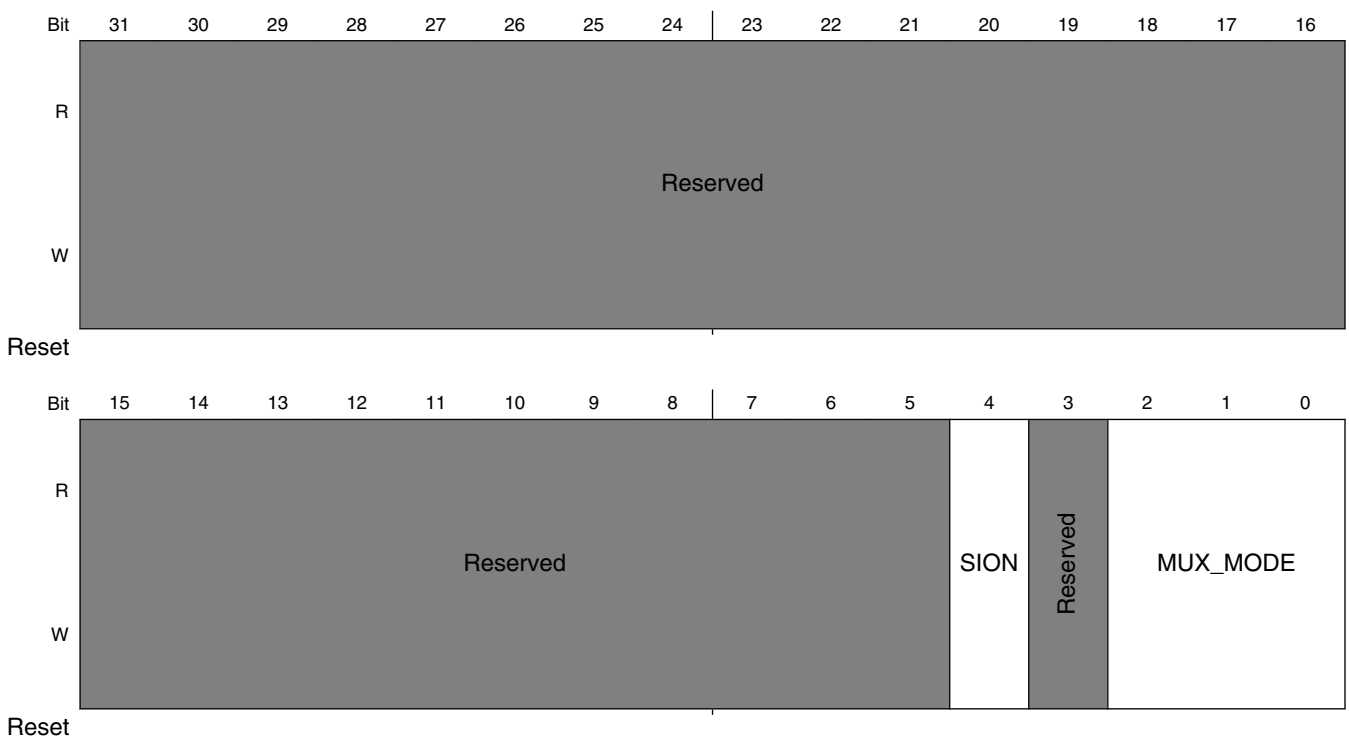
IOMUXC_SW_MUX_CTL_PAD_ENET_TD2 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: ENET_TD2. 000 **ALT0_ENET_QOS_RGMII_TD2** — Select mux mode: ALT0 mux port: ENET_QOS_RGMII_TD2 of instance: enet_qos 001 **ALT1_CCM_ENET_QOS_CLOCK_GENERATE_REF_CLK** — Select mux mode: ALT1 mux port: CCM_ENET_QOS_CLOCK_GENERATE_REF_CLK of instance: ccm_enet_qos_clock_generate 010 **ALT2_AUDIOMIX_SAI6_RX_DATA[0]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI6_RX_DATA00 of instance: sai6 011 **ALT3_AUDIOMIX_PDM_BIT_STREAM[1]** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_BIT_STREAM01 of instance: pdm 101 **ALT5_GPIO1_IO[19]** — Select mux mode: ALT5 mux port: GPIO1_IO19 of instance: gpio1 110 **ALT6_USDHC3_DATA7** — Select mux mode: ALT6 mux port: USDHC3_DATA7 of instance: usdhc3

8.2.4.21 SW_MUX_CTL_PAD_ENET_TD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD1)

SW_MUX_CTL Register

Address: 3033_0000h base + 64h offset = 3033_0064h



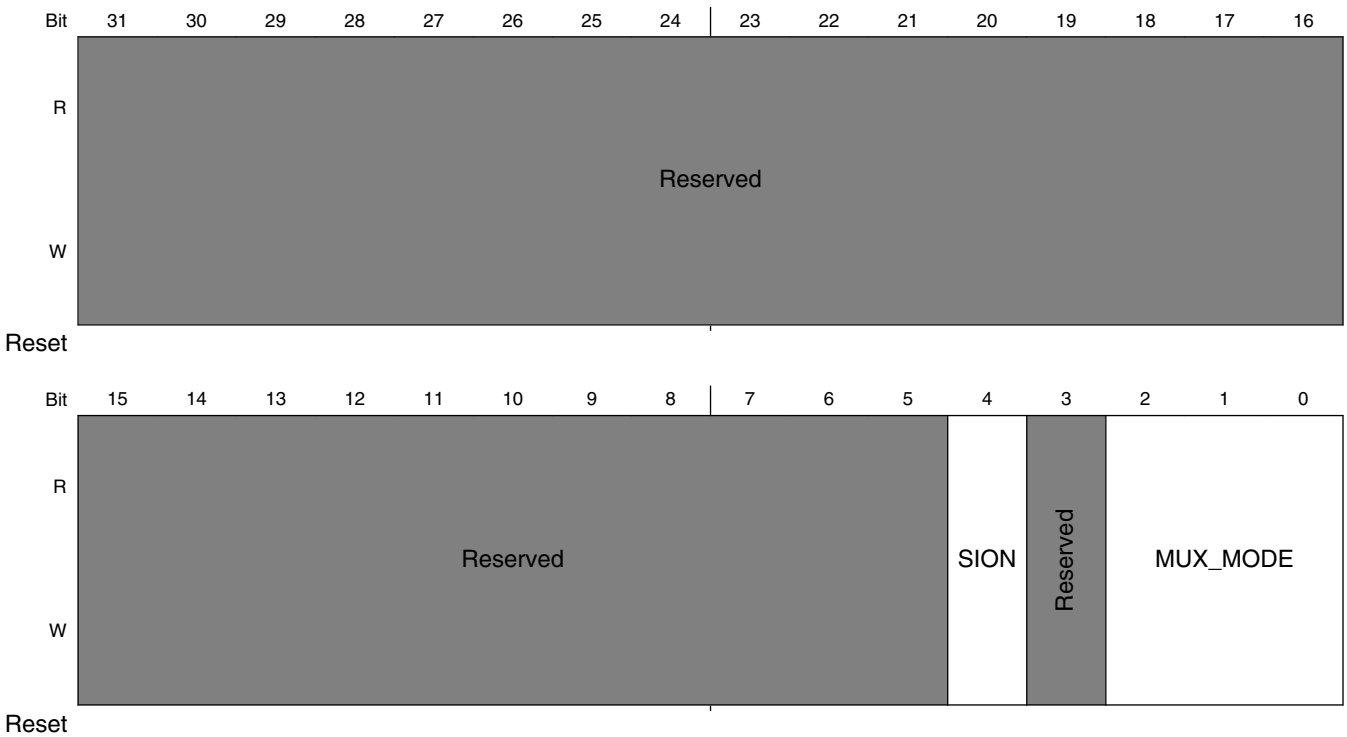
IOMUXC_SW_MUX_CTL_PAD_ENET_TD1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_TD1 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET_TD1. 000 **ALT0_ENET_QOS_RGMII_TD1** — Select mux mode: ALT0 mux port: ENET_QOS_RGMII_TD1 of instance: enet_qos 010 **ALT2_AUDIOMIX_SAI6_RX_SYNC** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI6_RX_SYNC of instance: sai6 011 **ALT3_AUDIOMIX_PDM_BIT_STREAM[0]** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_BIT_STREAM00 of instance: pdm 101 **ALT5_GPIO1_IO[20]** — Select mux mode: ALT5 mux port: GPIO1_IO20 of instance: gpio1 110 **ALT6_USDHC3_CD_B** — Select mux mode: ALT6 mux port: USDHC3_CD_B of instance: usdhc3

8.2.4.22 SW_MUX_CTL_PAD_ENET_TD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 68h offset = 3033_0068h



IOMUXC_SW_MUX_CTL_PAD_ENET_TD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_TD0 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET_TD0. 000 **ALT0_ENET_QOS_RGMII_TD0** — Select mux mode: ALT0 mux port: ENET_QOS_RGMII_TD0 of instance: enet_qos 010 **ALT2_AUDIOMIX_SAI6_RX_BCLK** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI6_RX_BCLK of instance: sai6

Table continues on the next page...

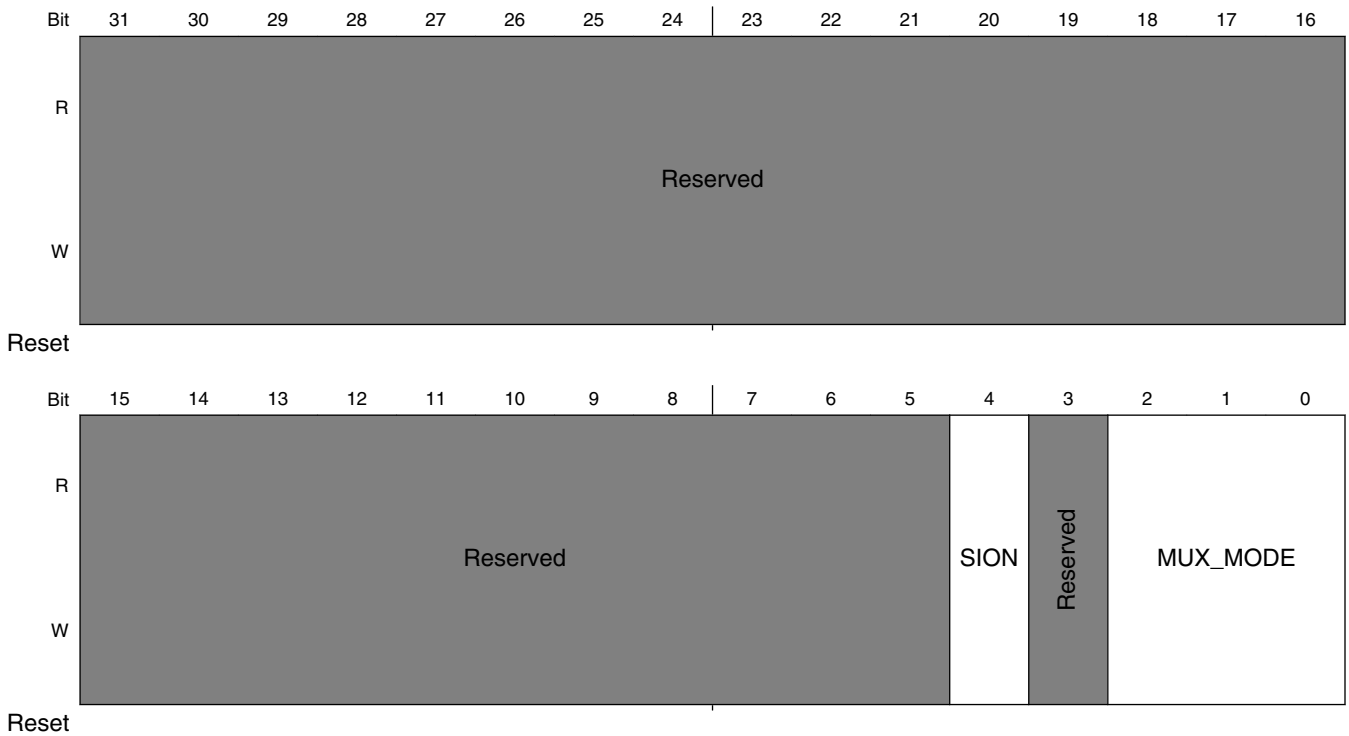
IOMUXC_SW_MUX_CTL_PAD_ENET_TD0 field descriptions (continued)

Field	Description
	011 **ALT3_AUDIOMIX_PDM_CLK** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_CLK of instance: pdm 101 **ALT5_GPIO1_IO[21]** — Select mux mode: ALT5 mux port: GPIO1_IO21 of instance: gpio1 110 **ALT6_USDHC3_WP** — Select mux mode: ALT6 mux port: USDHC3_WP of instance: usdhc3

8.2.4.23 SW_MUX_CTL_PAD_ENET_TX_CTL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TX_CTL)

SW_MUX_CTL Register

Address: 3033_0000h base + 6Ch offset = 3033_006Ch



IOMUXC_SW_MUX_CTL_PAD_ENET_TX_CTL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_TX_CTL 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

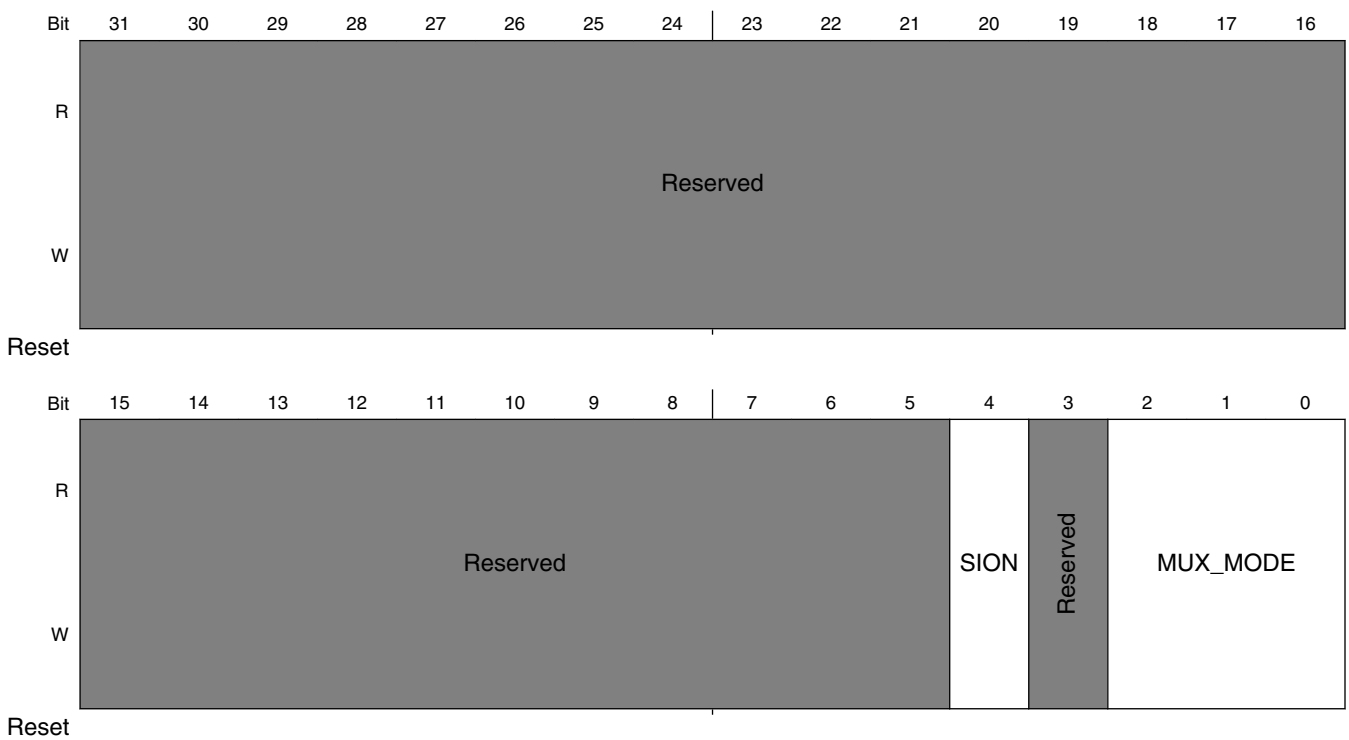
IOMUXC_SW_MUX_CTL_PAD_ENET_TX_CTL field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET_TX_CTL. 000 **ALT0_ENET_QOS_RGMII_TX_CTL** — Select mux mode: ALT0 mux port: ENET_QOS_RGMII_TX_CTL of instance: enet_qos 010 **ALT2_AUDIOMIX_SAI6_MCLK** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI6_MCLK of instance: sai6 011 **ALT3_AUDIOMIX_SPDIF1_OUT** — Select mux mode: ALT3 mux port: AUDIOMIX_SPDIF1_OUT of instance: spdif 101 **ALT5_GPIO1_IO[22]** — Select mux mode: ALT5 mux port: GPIO1_IO22 of instance: gpio1 110 **ALT6_USDHC3_DATA0** — Select mux mode: ALT6 mux port: USDHC3_DATA0 of instance: usdhc3

8.2.4.24 SW_MUX_CTL_PAD_ENET_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 70h offset = 3033_0070h



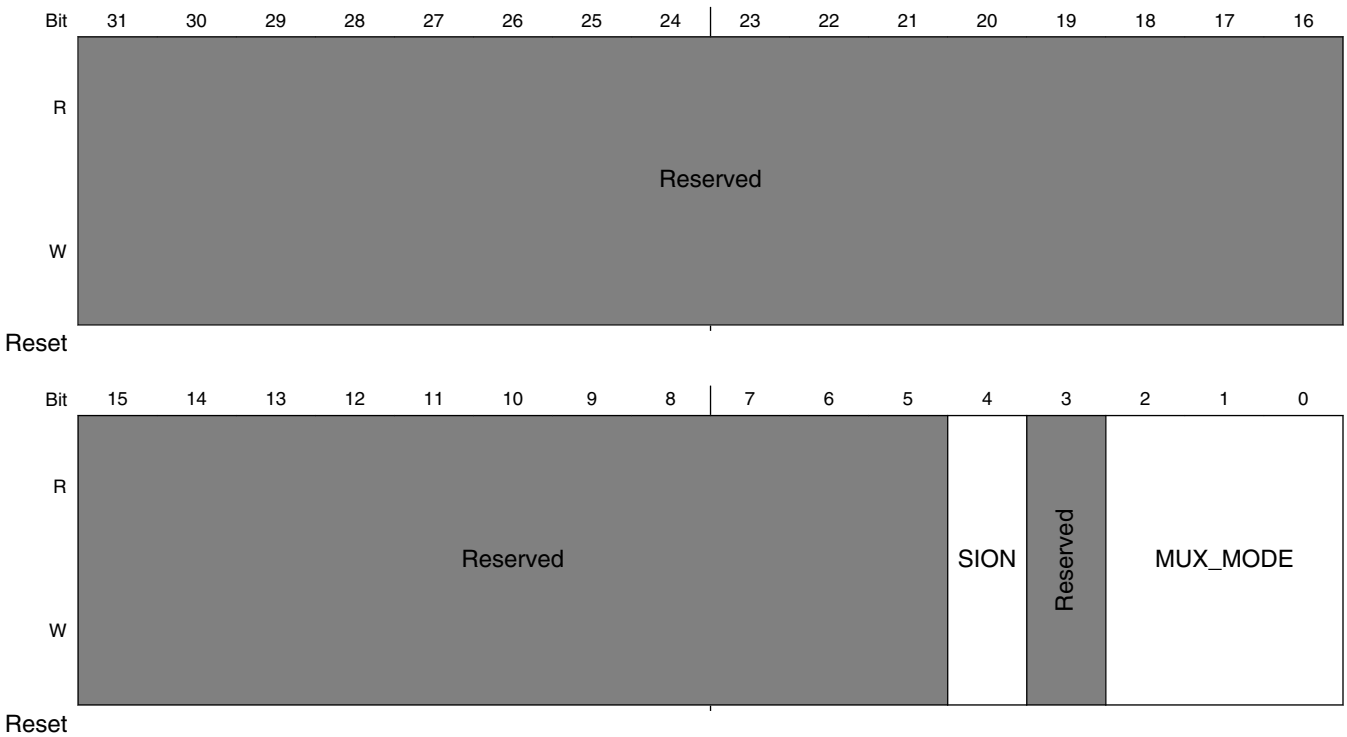
IOMUXC_SW_MUX_CTL_PAD_ENET_TXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_TXC 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET_TXC. 000 **ALT0_CCM_ENET_QOS_CLOCK_GENERATE_TX_CLK** — Select mux mode: ALT0 mux port: CCM_ENET_QOS_CLOCK_GENERATE_TX_CLK of instance: ccm_enet_qos_clock_generate 001 **ALT1_ENET_QOS_TX_ER** — Select mux mode: ALT1 mux port: ENET_QOS_TX_ER of instance: enet_qos 010 **ALT2_AUDIOMIX_SAI7_TX_DATA[0]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI7_TX_DATA00 of instance: sai7 101 **ALT5_GPIO1_IO[23]** — Select mux mode: ALT5 mux port: GPIO1_IO23 of instance: gpio1 110 **ALT6_USDHC3_DATA1** — Select mux mode: ALT6 mux port: USDHC3_DATA1 of instance: usdhc3

8.2.4.25 SW_MUX_CTL_PAD_ENET_RX_CTL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RX_CTL)

SW_MUX_CTL Register

Address: 3033_0000h base + 74h offset = 3033_0074h



IOMUXC_SW_MUX_CTL_PAD_ENET_RX_CTL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_RX_CTL 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET_RX_CTL. 000 **ALT0_ENET_QOS_RGMII_RX_CTL** — Select mux mode: ALT0 mux port: ENET_QOS_RGMII_RX_CTL of instance: enet_qos 010 **ALT2_AUDIOMIX_SAI7_TX_SYNC** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI7_TX_SYNC of instance: sai7

Table continues on the next page...

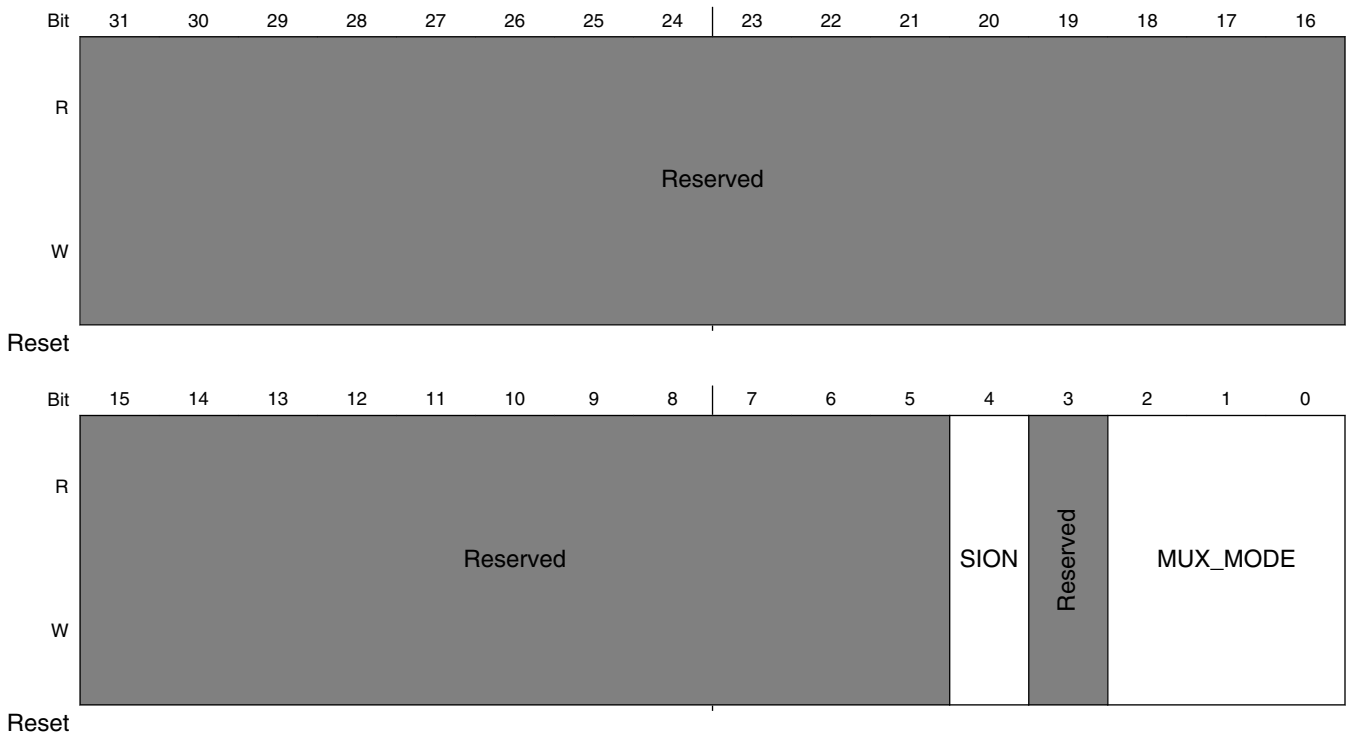
IOMUXC_SW_MUX_CTL_PAD_ENET_RX_CTL field descriptions (continued)

Field	Description
	011 **ALT3_AUDIOMIX_PDM_BIT_STREAM[3]** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_BIT_STREAM03 of instance: pdm 101 **ALT5_GPIO1_IO[24]** — Select mux mode: ALT5 mux port: GPIO1_IO24 of instance: gpio1 110 **ALT6_USDHC3_DATA2** — Select mux mode: ALT6 mux port: USDHC3_DATA2 of instance: usdhc3

8.2.4.26 SW_MUX_CTL_PAD_ENET_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 78h offset = 3033_0078h



IOMUXC_SW_MUX_CTL_PAD_ENET_RXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_RXC 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

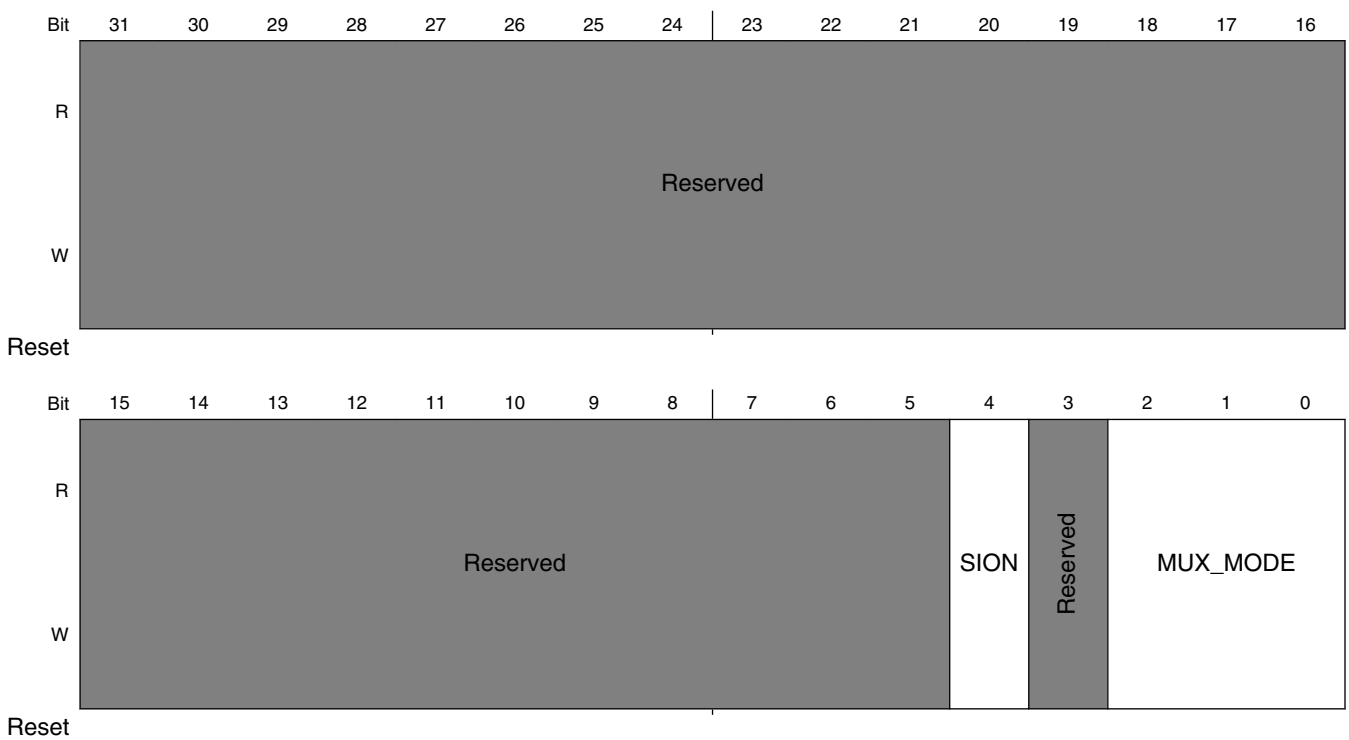
IOMUXC_SW_MUX_CTL_PAD_ENET_RXC field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: ENET_RXC. 000 **ALT0_CCM_ENET_QOS_CLOCK_GENERATE_RX_CLK** — Select mux mode: ALT0 mux port: CCM_ENET_QOS_CLOCK_GENERATE_RX_CLK of instance: ccm_enet_qos_clock_generate 001 **ALT1_ENET_QOS_RX_ER** — Select mux mode: ALT1 mux port: ENET_QOS_RX_ER of instance: enet_qos 010 **ALT2_AUDIOMIX_SAI7_TX_BCLK** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI7_TX_BCLK of instance: sai7 011 **ALT3_AUDIOMIX_PDM_BIT_STREAM[2]** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_BIT_STREAM02 of instance: pdm 101 **ALT5_GPIO1_IO[25]** — Select mux mode: ALT5 mux port: GPIO1_IO25 of instance: gpio1 110 **ALT6_USDHC3_DATA3** — Select mux mode: ALT6 mux port: USDHC3_DATA3 of instance: usdhc3

8.2.4.27 SW_MUX_CTL_PAD_ENET_RD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 7Ch offset = 3033_007Ch



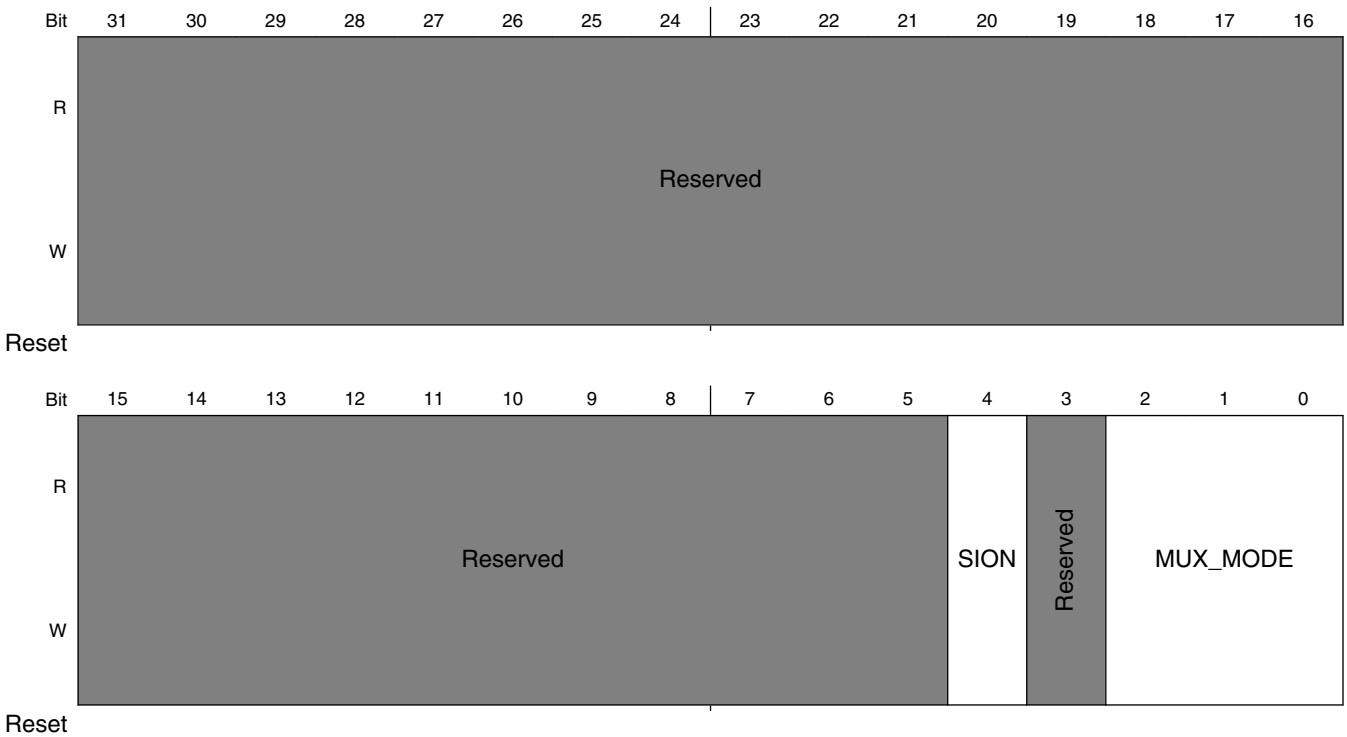
IOMUXC_SW_MUX_CTL_PAD_ENET_RD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_RD0 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET_RD0. 000 **ALT0_ENET_QOS_RGMII_RD0** — Select mux mode: ALT0 mux port: ENET_QOS_RGMII_RD0 of instance: enet_qos 010 **ALT2_AUDIOMIX_SAI7_RX_DATA[0]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI7_RX_DATA00 of instance: sai7 011 **ALT3_AUDIOMIX_PDM_BIT_STREAM[1]** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_BIT_STREAM01 of instance: pdm 101 **ALT5_GPIO1_IO[26]** — Select mux mode: ALT5 mux port: GPIO1_IO26 of instance: gpio1 110 **ALT6_USDHC3_DATA4** — Select mux mode: ALT6 mux port: USDHC3_DATA4 of instance: usdhc3

8.2.4.28 SW_MUX_CTL_PAD_ENET_RD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD1)

SW_MUX_CTL Register

Address: 3033_0000h base + 80h offset = 3033_0080h



IOMUXC_SW_MUX_CTL_PAD_ENET_RD1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_RD1 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET_RD1. 000 **ALT0_ENET_QOS_RGMII_RD1** — Select mux mode: ALT0 mux port: ENET_QOS_RGMII_RD1 of instance: enet_qos 010 **ALT2_AUDIOMIX_SAI7_RX_SYNC** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI7_RX_SYNC of instance: sai7

Table continues on the next page...

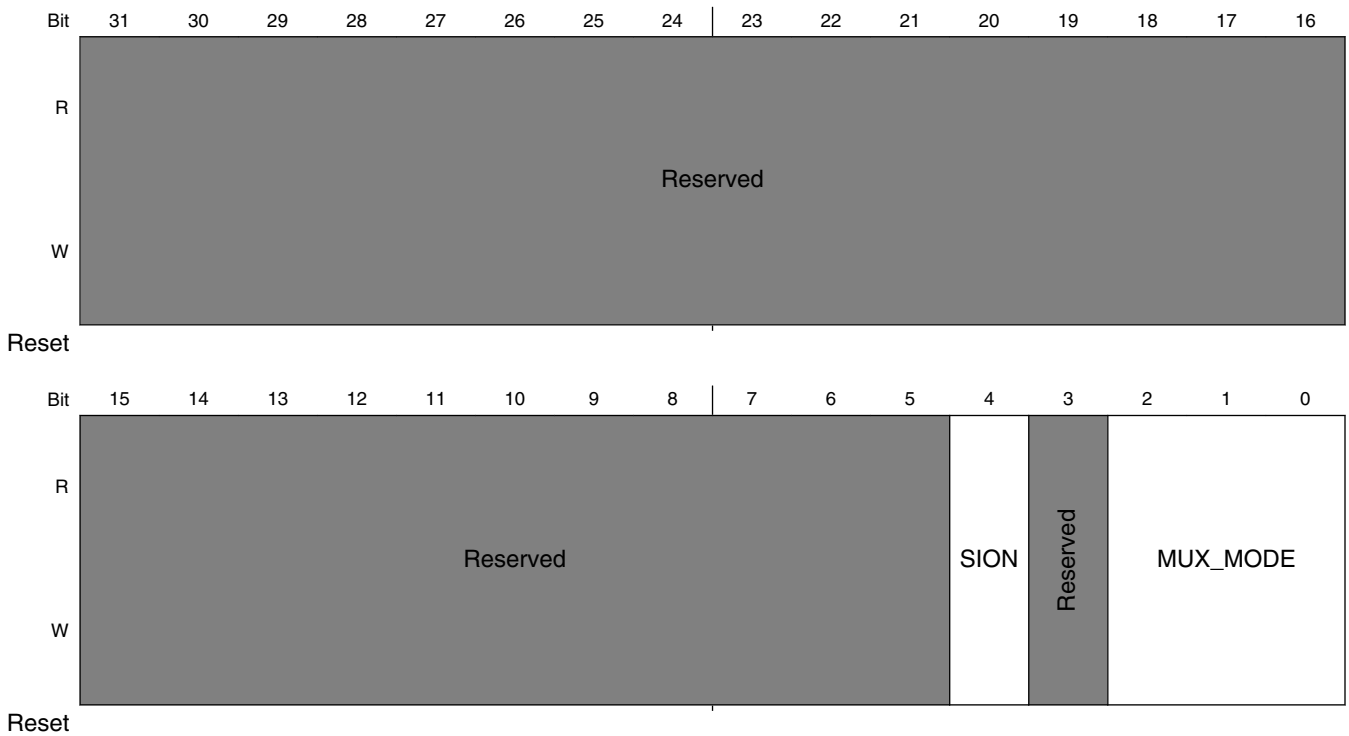
IOMUXC_SW_MUX_CTL_PAD_ENET_RD1 field descriptions (continued)

Field	Description
	011 **ALT3_AUDIOMIX_PDM_BIT_STREAM[0]** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_BIT_STREAM00 of instance: pdm 101 **ALT5_GPIO1_IO[27]** — Select mux mode: ALT5 mux port: GPIO1_IO27 of instance: gpio1 110 **ALT6_USDHC3_RESET_B** — Select mux mode: ALT6 mux port: USDHC3_RESET_B of instance: usdhc3

8.2.4.29 SW_MUX_CTL_PAD_ENET_RD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD2)

SW_MUX_CTL Register

Address: 3033_0000h base + 84h offset = 3033_0084h



IOMUXC_SW_MUX_CTL_PAD_ENET_RD2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_RD2 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

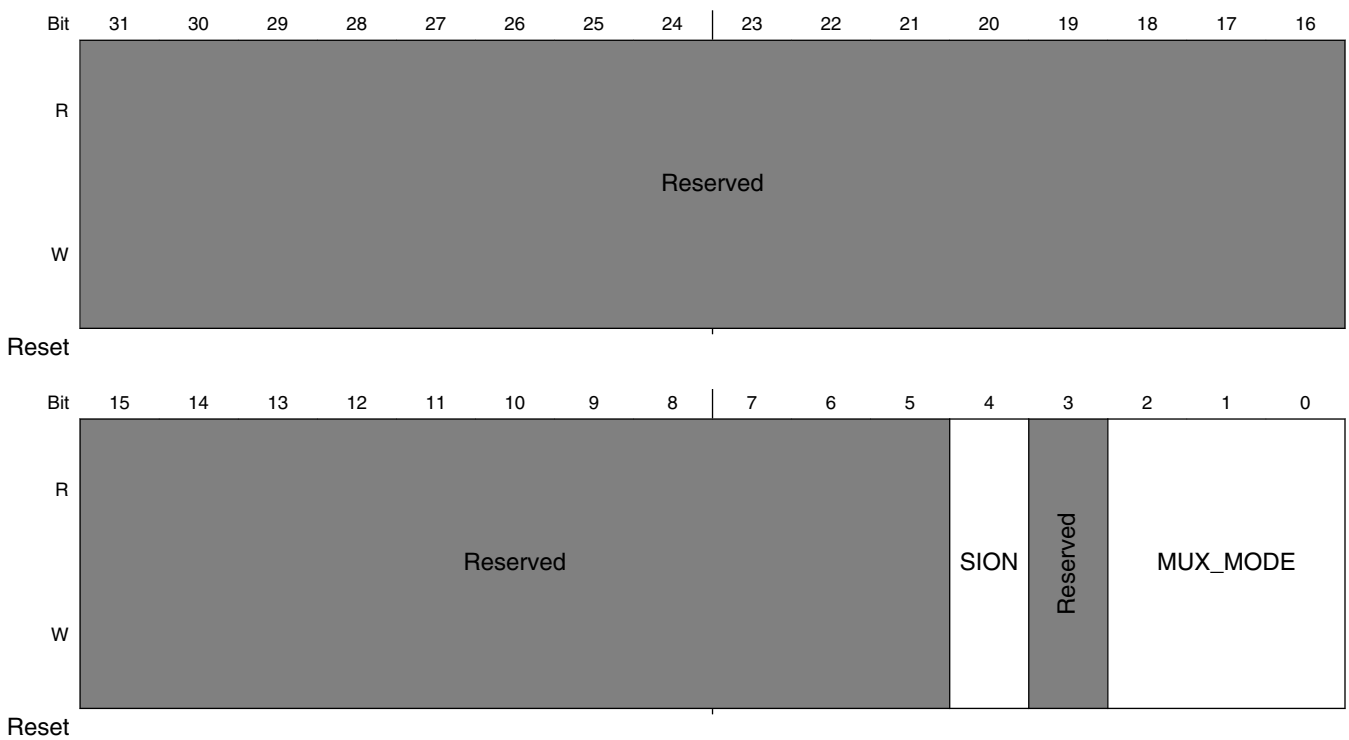
IOMUXC_SW_MUX_CTL_PAD_ENET_RD2 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET_RD2. 000 **ALT0_ENET_QOS_RGMII_RD2** — Select mux mode: ALT0 mux port: ENET_QOS_RGMII_RD2 of instance: enet_qos 010 **ALT2_AUDIOMIX_SAI7_RX_BCLK** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI7_RX_BCLK of instance: sai7 011 **ALT3_AUDIOMIX_PDM_CLK** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_CLK of instance: pdm 101 **ALT5_GPIO1_IO[28]** — Select mux mode: ALT5 mux port: GPIO1_IO28 of instance: gpio1 110 **ALT6_USDHC3_CLK** — Select mux mode: ALT6 mux port: USDHC3_CLK of instance: usdhc3

8.2.4.30 SW_MUX_CTL_PAD_ENET_RD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD3)

SW_MUX_CTL Register

Address: 3033_0000h base + 88h offset = 3033_0088h



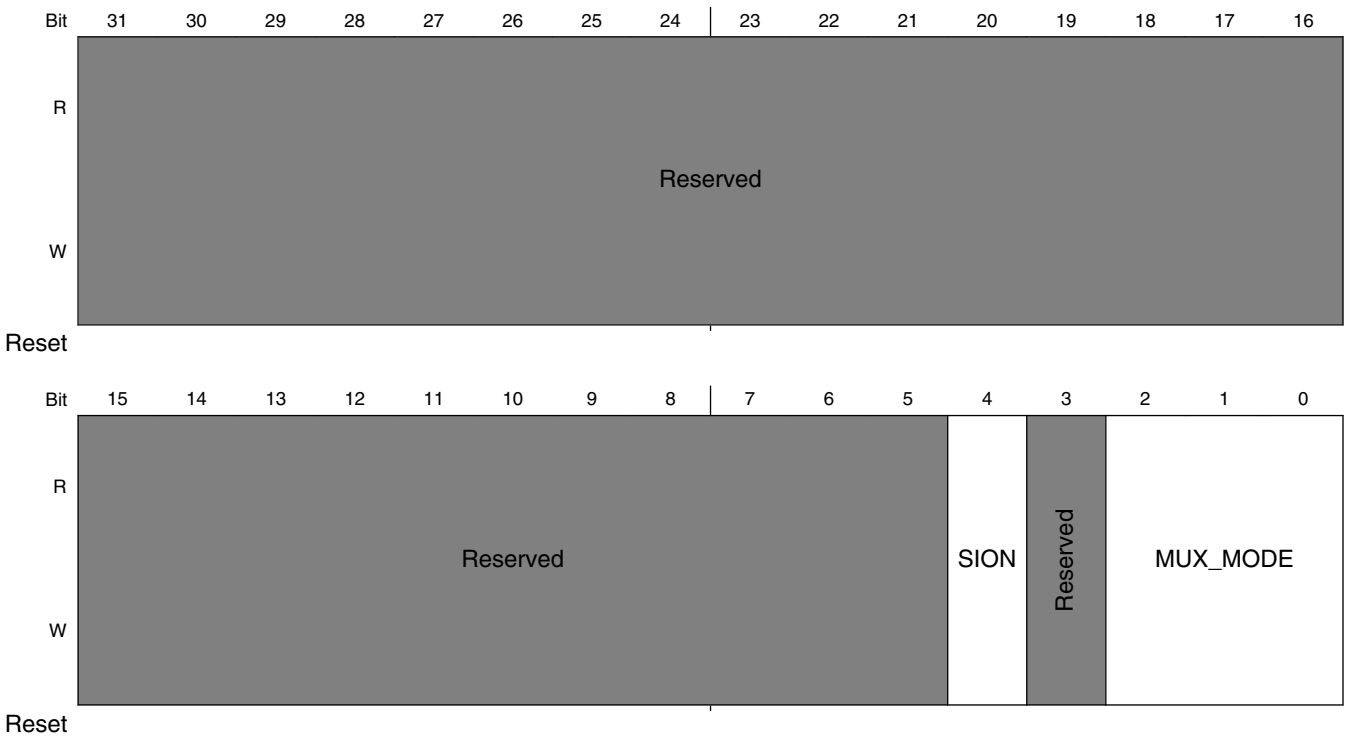
IOMUXC_SW_MUX_CTL_PAD_ENET_RD3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ENET_RD3 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ENET_RD3. 000 **ALT0_ENET_QOS_RGMII_RD3** — Select mux mode: ALT0 mux port: ENET_QOS_RGMII_RD3 of instance: enet_qos 010 **ALT2_AUDIOMIX_SAI7_MCLK** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI7_MCLK of instance: sai7 011 **ALT3_AUDIOMIX_SPDIF1_IN** — Select mux mode: ALT3 mux port: AUDIOMIX_SPDIF1_IN of instance: spdif 101 **ALT5_GPIO1_IO[29]** — Select mux mode: ALT5 mux port: GPIO1_IO29 of instance: gpio1 110 **ALT6_USDHC3_CMD** — Select mux mode: ALT6 mux port: USDHC3_CMD of instance: usdhc3

8.2.4.31 SW_MUX_CTL_PAD_SD1_CLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 8Ch offset = 3033_008Ch



IOMUXC_SW_MUX_CTL_PAD_SD1_CLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_CLK 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD1_CLK. 000 **ALT0_USDHC1_CLK** — Select mux mode: ALT0 mux port: USDHC1_CLK of instance: usdhc1 001 **ALT1_ENET1_MDC** — Select mux mode: ALT1 mux port: ENET1_MDC of instance: enet1 011 **ALT3_I2C5_SCL** — Select mux mode: ALT3 mux port: I2C5_SCL of instance: i2c5

Table continues on the next page...

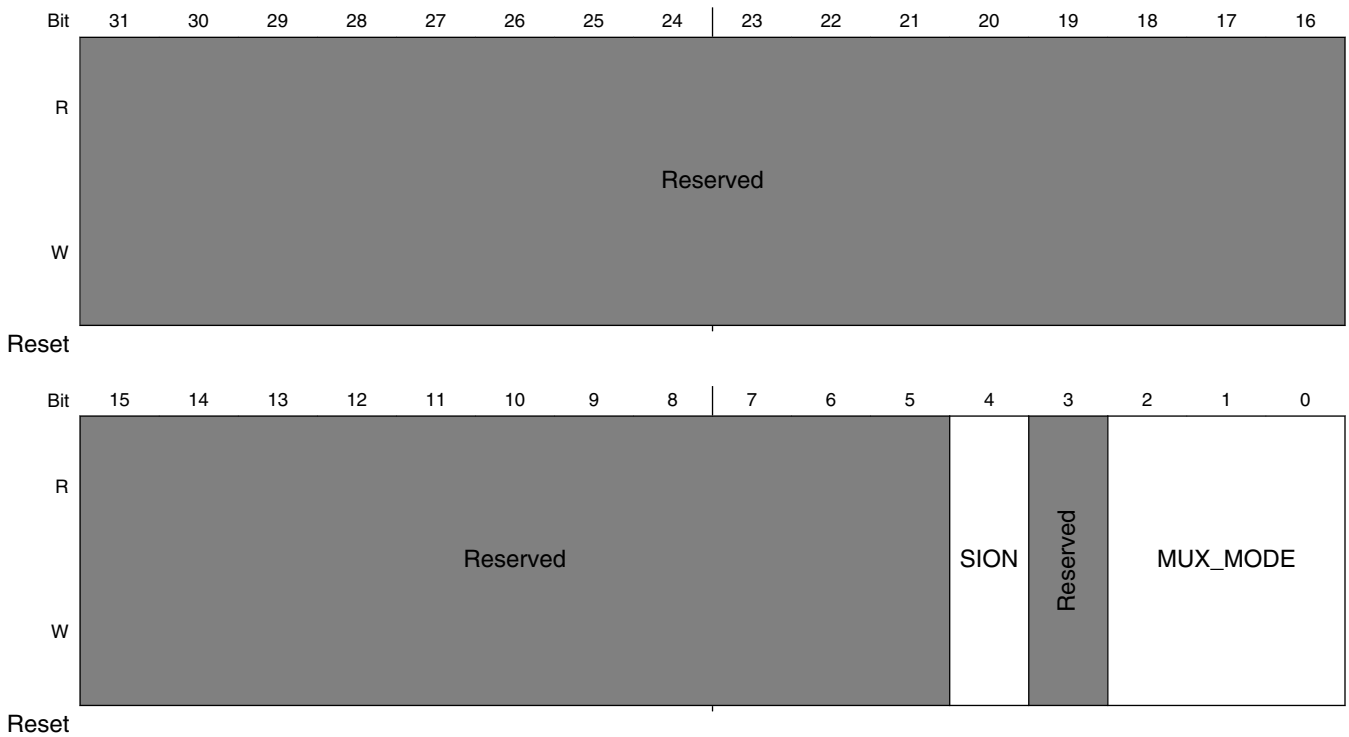
IOMUXC_SW_MUX_CTL_PAD_SD1_CLK field descriptions (continued)

Field	Description
	100 **ALT4_UART1_TX** — Select mux mode: ALT4 mux port: UART1_TX of instance: uart1 101 **ALT5_GPIO2_IO[0]** — Select mux mode: ALT5 mux port: GPIO2_IO00 of instance: gpio2

8.2.4.32 SW_MUX_CTL_PAD_SD1_CMD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CMD)

SW_MUX_CTL Register

Address: 3033_0000h base + 90h offset = 3033_0090h



IOMUXC_SW_MUX_CTL_PAD_SD1_CMD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_CMD 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved

Table continues on the next page...

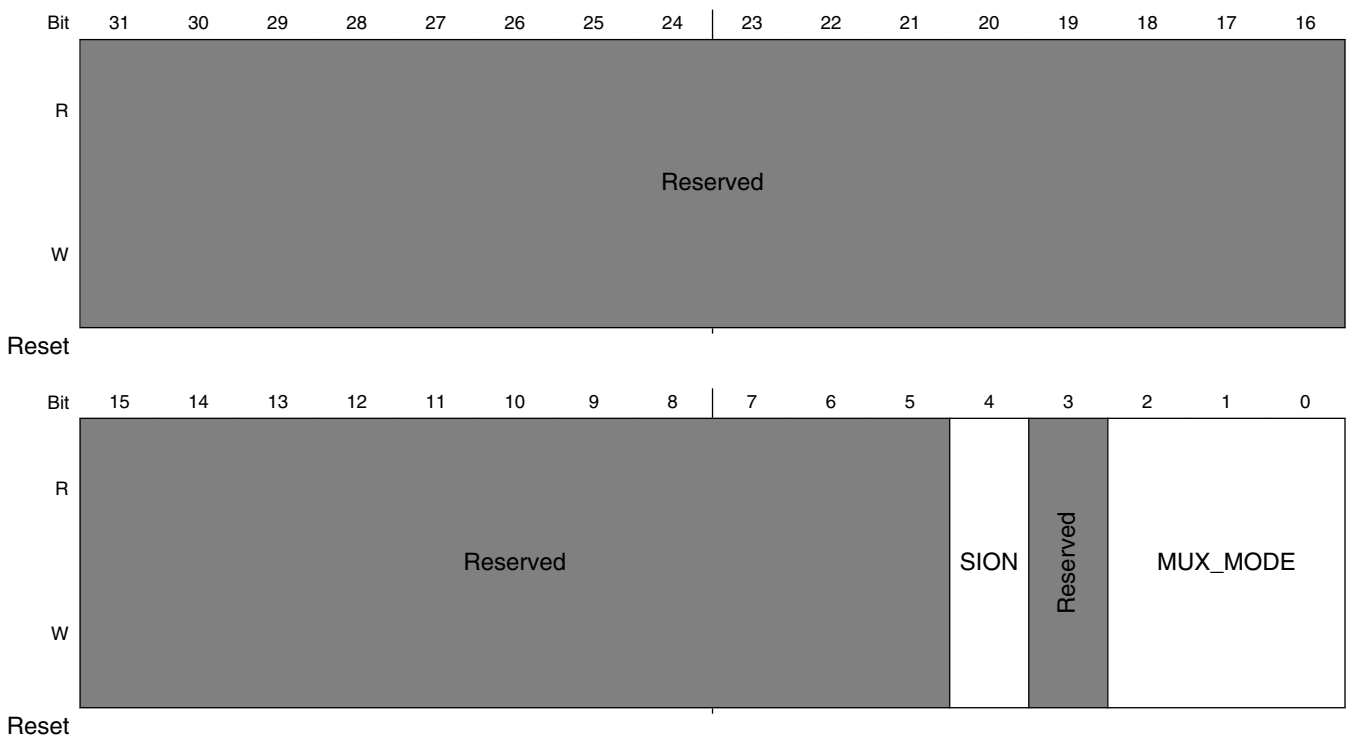
IOMUXC_SW_MUX_CTL_PAD_SD1_CMD field descriptions (continued)

Field	Description
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD1_CMD. 000 **ALT0_USDHC1_CMD** — Select mux mode: ALT0 mux port: USDHC1_CMD of instance: usdhc1 001 **ALT1_ENET1_MDIO** — Select mux mode: ALT1 mux port: ENET1_MDIO of instance: enet1 011 **ALT3_I2C5_SDA** — Select mux mode: ALT3 mux port: I2C5_SDA of instance: i2c5 100 **ALT4_UART1_RX** — Select mux mode: ALT4 mux port: UART1_RX of instance: uart1 101 **ALT5_GPIO2_IO[1]** — Select mux mode: ALT5 mux port: GPIO2_IO01 of instance: gpio2

8.2.4.33 SW_MUX_CTL_PAD_SD1_DATA0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0)

SW_MUX_CTL Register

Address: 3033_0000h base + 94h offset = 3033_0094h



IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved

Table continues on the next page...

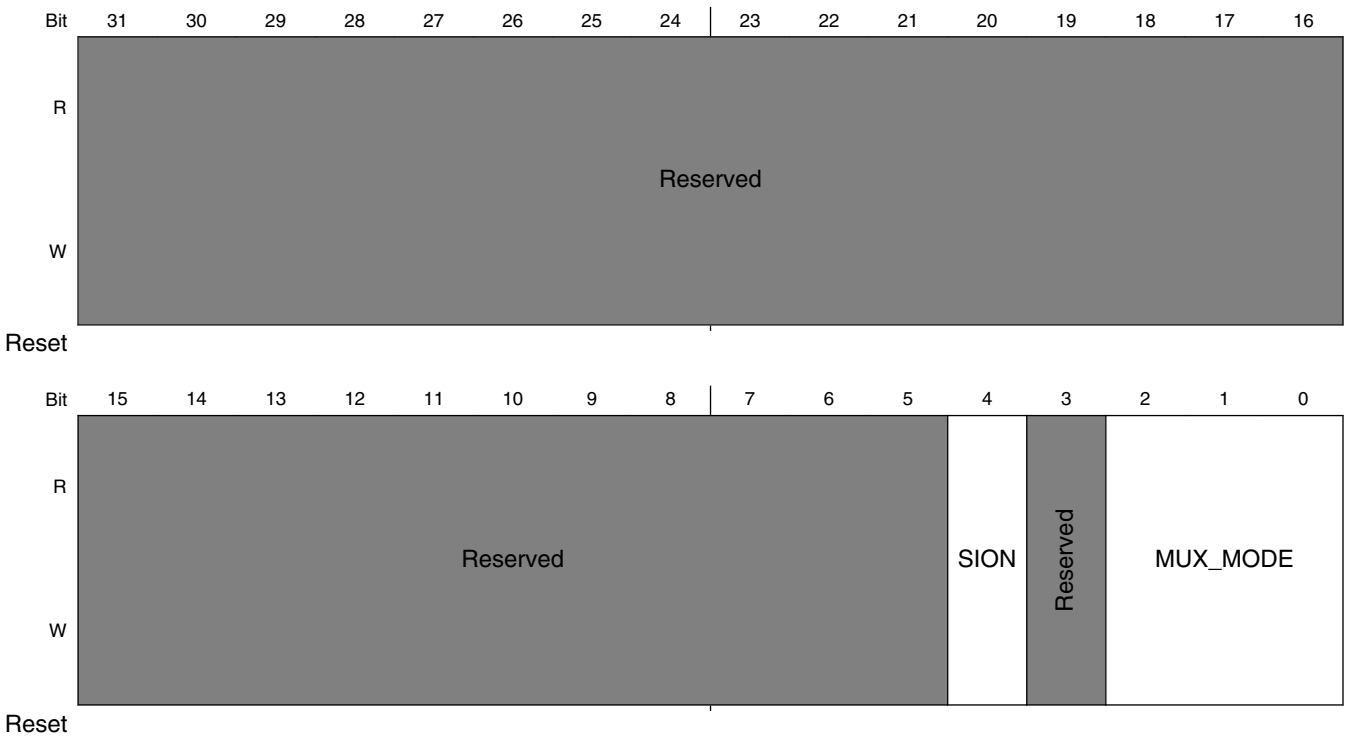
IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0 field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_DATA0 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD1_DATA0. 000 **ALT0_USDHC1_DATA0** — Select mux mode: ALT0 mux port: USDHC1_DATA0 of instance: usdhc1 001 **ALT1_ENET1_RGMII_TD1** — Select mux mode: ALT1 mux port: ENET1_RGMII_TD1 of instance: enet1 011 **ALT3_I2C6_SCL** — Select mux mode: ALT3 mux port: I2C6_SCL of instance: i2c6 100 **ALT4_UART1_RTS_B** — Select mux mode: ALT4 mux port: UART1_RTS_B of instance: uart1 101 **ALT5_GPIO2_IO[2]** — Select mux mode: ALT5 mux port: GPIO2_IO02 of instance: gpio2

8.2.4.34 SW_MUX_CTL_PAD_SD1_DATA1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1)

SW_MUX_CTL Register

Address: 3033_0000h base + 98h offset = 3033_0098h



IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_DATA1 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD1_DATA1. 000 **ALT0_USDHC1_DATA1** — Select mux mode: ALT0 mux port: USDHC1_DATA1 of instance: usdhc1 001 **ALT1_ENET1_RGMII_TD0** — Select mux mode: ALT1 mux port: ENET1_RGMII_TD0 of instance: enet1

Table continues on the next page...

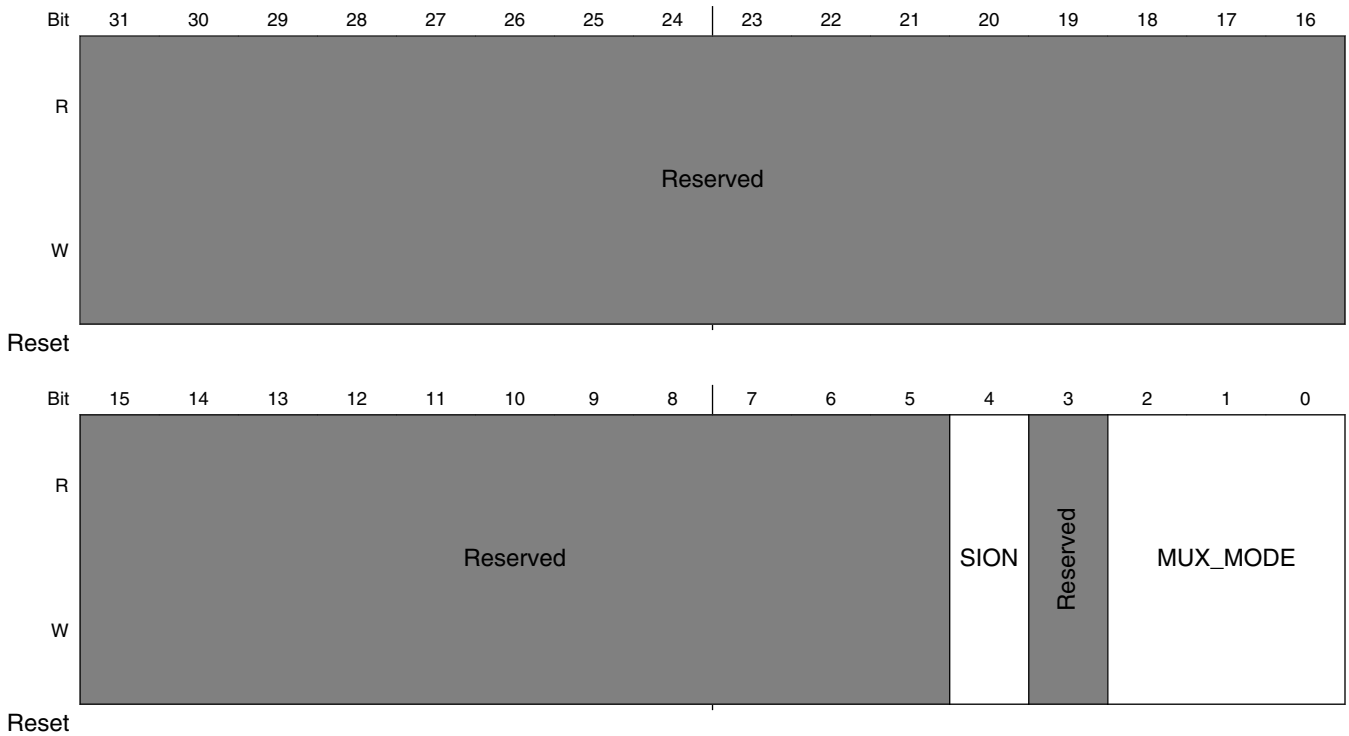
IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1 field descriptions (continued)

Field	Description
	011 **ALT3_I2C6_SDA** — Select mux mode: ALT3 mux port: I2C6_SDA of instance: i2c6 100 **ALT4_UART1_CTS_B** — Select mux mode: ALT4 mux port: UART1_CTS_B of instance: uart1 101 **ALT5_GPIO2_IO[3]** — Select mux mode: ALT5 mux port: GPIO2_IO03 of instance: gpio2

8.2.4.35 SW_MUX_CTL_PAD_SD1_DATA2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2)

SW_MUX_CTL Register

Address: 3033_0000h base + 9Ch offset = 3033_009Ch



IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_DATA2 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

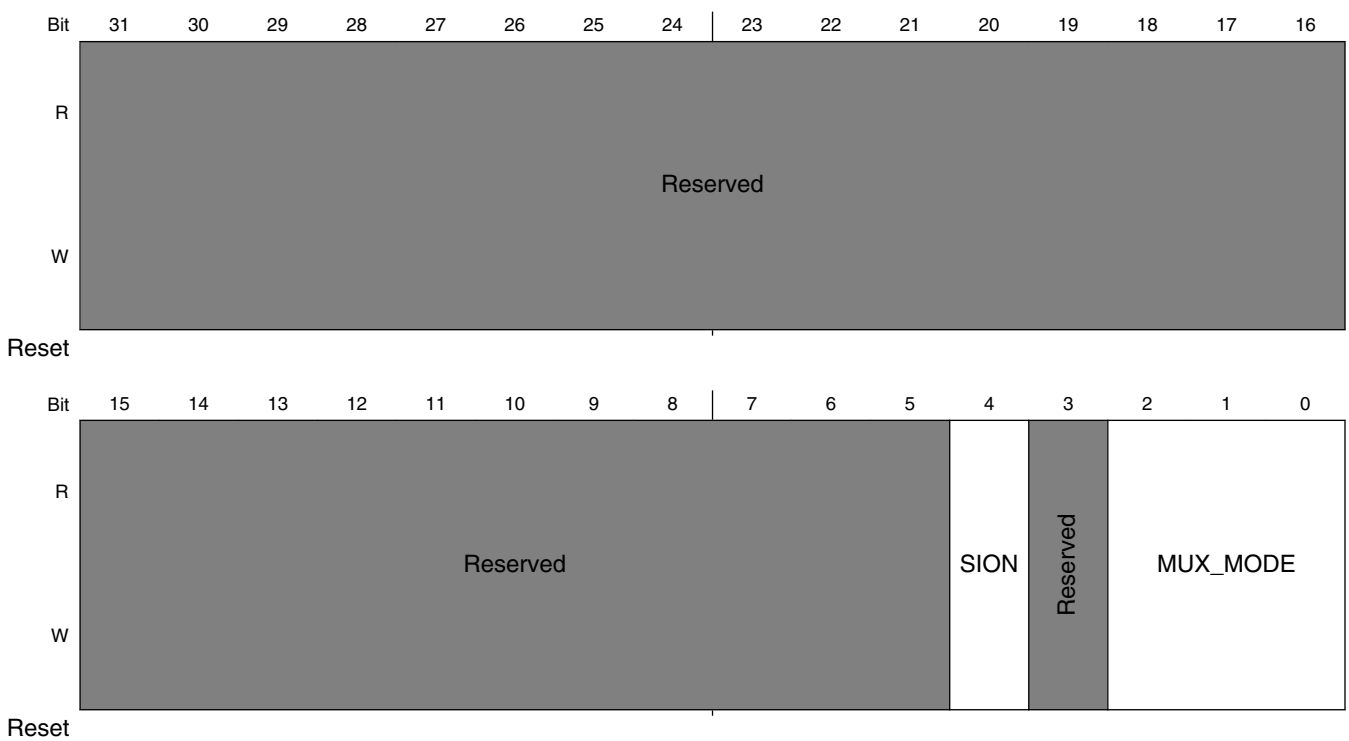
IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD1_DATA2. 000 **ALT0_USDHC1_DATA2** — Select mux mode: ALT0 mux port: USDHC1_DATA2 of instance: usdhc1 001 **ALT1_ENET1_RGMII_RD0** — Select mux mode: ALT1 mux port: ENET1_RGMII_RD0 of instance: enet1 011 **ALT3_I2C4_SCL** — Select mux mode: ALT3 mux port: I2C4_SCL of instance: i2c4 100 **ALT4_UART2_TX** — Select mux mode: ALT4 mux port: UART2_TX of instance: uart2 101 **ALT5_GPIO2_IO[4]** — Select mux mode: ALT5 mux port: GPIO2_IO04 of instance: gpio2

8.2.4.36 SW_MUX_CTL_PAD_SD1_DATA3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3)

SW_MUX_CTL Register

Address: 3033_0000h base + A0h offset = 3033_00A0h



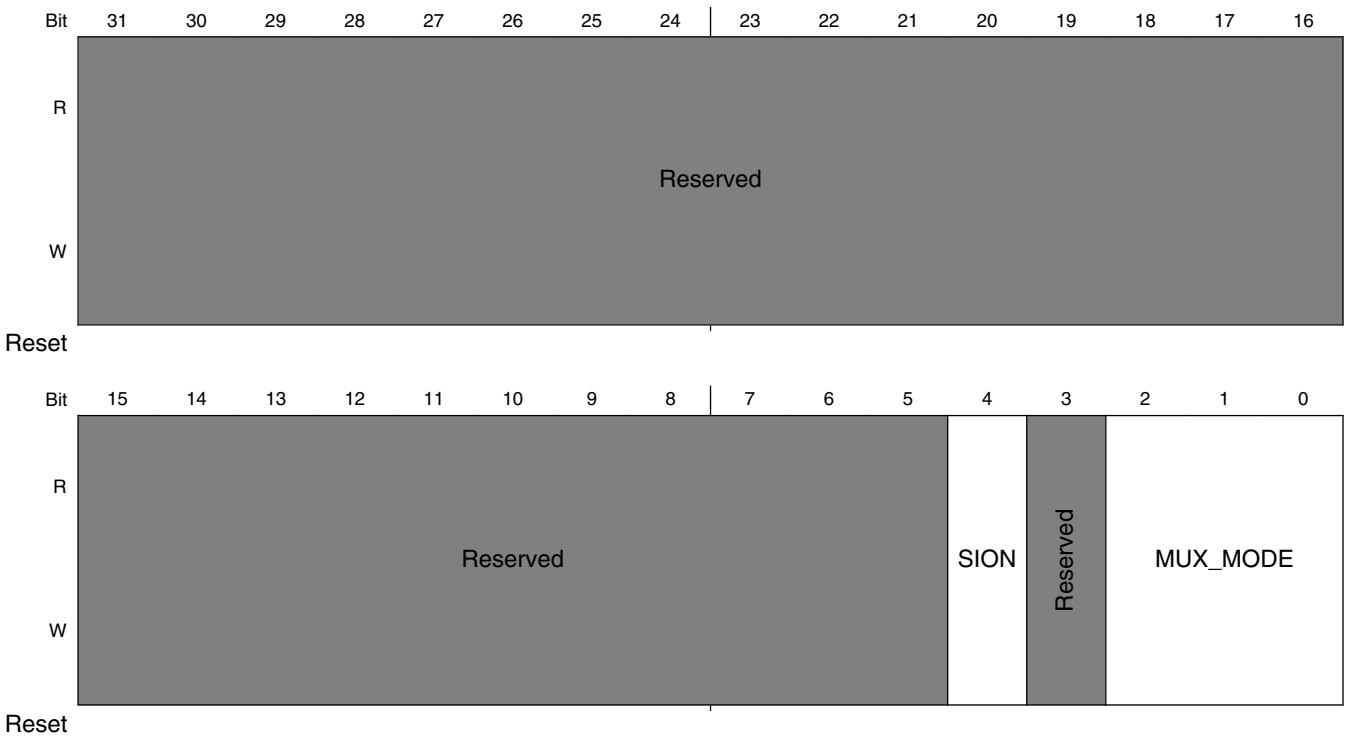
IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_DATA3 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD1_DATA3. 000 **ALT0_USDHC1_DATA3** — Select mux mode: ALT0 mux port: USDHC1_DATA3 of instance: usdhc1 001 **ALT1_ENET1_RGMII_RD1** — Select mux mode: ALT1 mux port: ENET1_RGMII_RD1 of instance: enet1 011 **ALT3_I2C4_SDA** — Select mux mode: ALT3 mux port: I2C4_SDA of instance: i2c4 100 **ALT4_UART2_RX** — Select mux mode: ALT4 mux port: UART2_RX of instance: uart2 101 **ALT5_GPIO2_IO[5]** — Select mux mode: ALT5 mux port: GPIO2_IO05 of instance: gpio2

8.2.4.37 SW_MUX_CTL_PAD_SD1_DATA4 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA4)

SW_MUX_CTL Register

Address: 3033_0000h base + A4h offset = 3033_00A4h



IOMUXC_SW_MUX_CTL_PAD_SD1_DATA4 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_DATA4 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD1_DATA4. 000 **ALT0_USDHC1_DATA4** — Select mux mode: ALT0 mux port: USDHC1_DATA4 of instance: usdhc1 001 **ALT1_ENET1_RGMII_TX_CTL** — Select mux mode: ALT1 mux port: ENET1_RGMII_TX_CTL of instance: enet1

Table continues on the next page...

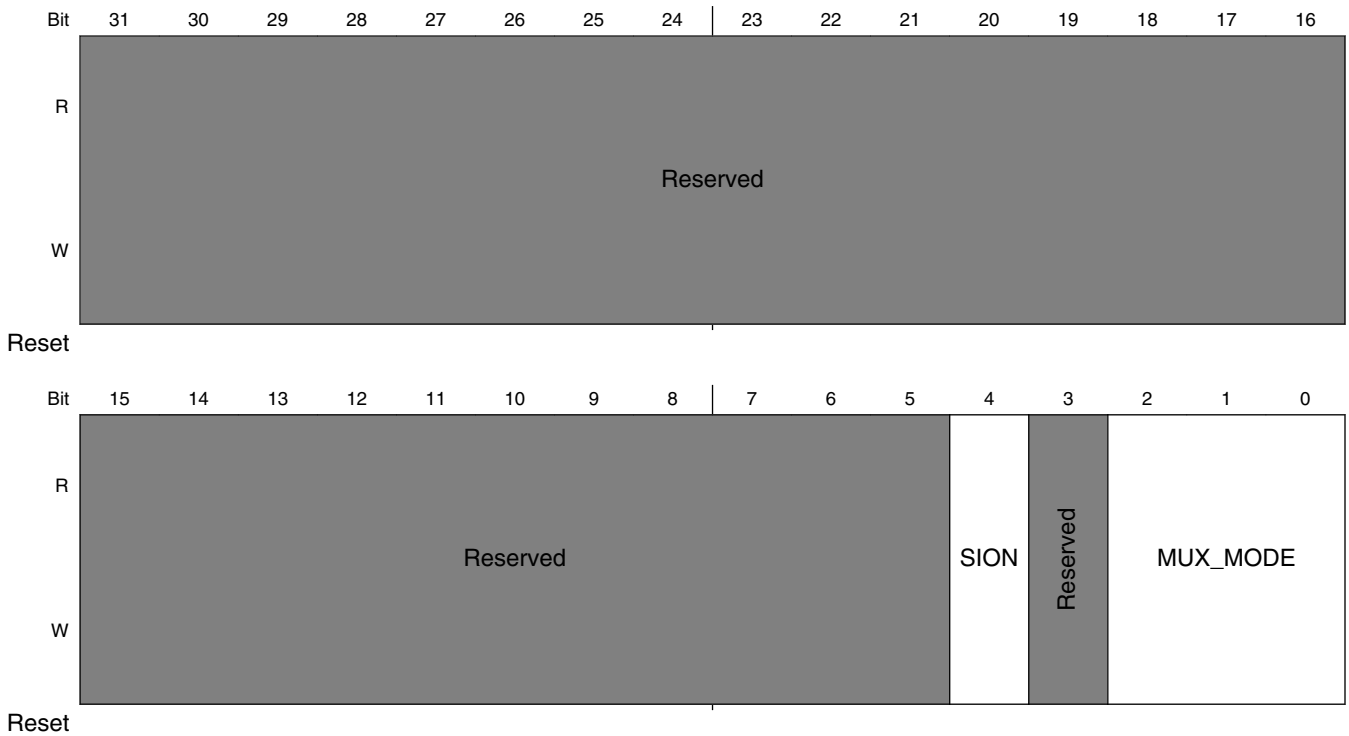
IOMUXC_SW_MUX_CTL_PAD_SD1_DATA4 field descriptions (continued)

Field	Description
	011 **ALT3_I2C1_SCL** — Select mux mode: ALT3 mux port: I2C1_SCL of instance: i2c1 100 **ALT4_UART2_RTS_B** — Select mux mode: ALT4 mux port: UART2_RTS_B of instance: uart2 101 **ALT5_GPIO2_IO[6]** — Select mux mode: ALT5 mux port: GPIO2_IO06 of instance: gpio2

8.2.4.38 SW_MUX_CTL_PAD_SD1_DATA5 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA5)

SW_MUX_CTL Register

Address: 3033_0000h base + A8h offset = 3033_00A8h



IOMUXC_SW_MUX_CTL_PAD_SD1_DATA5 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_DATA5 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

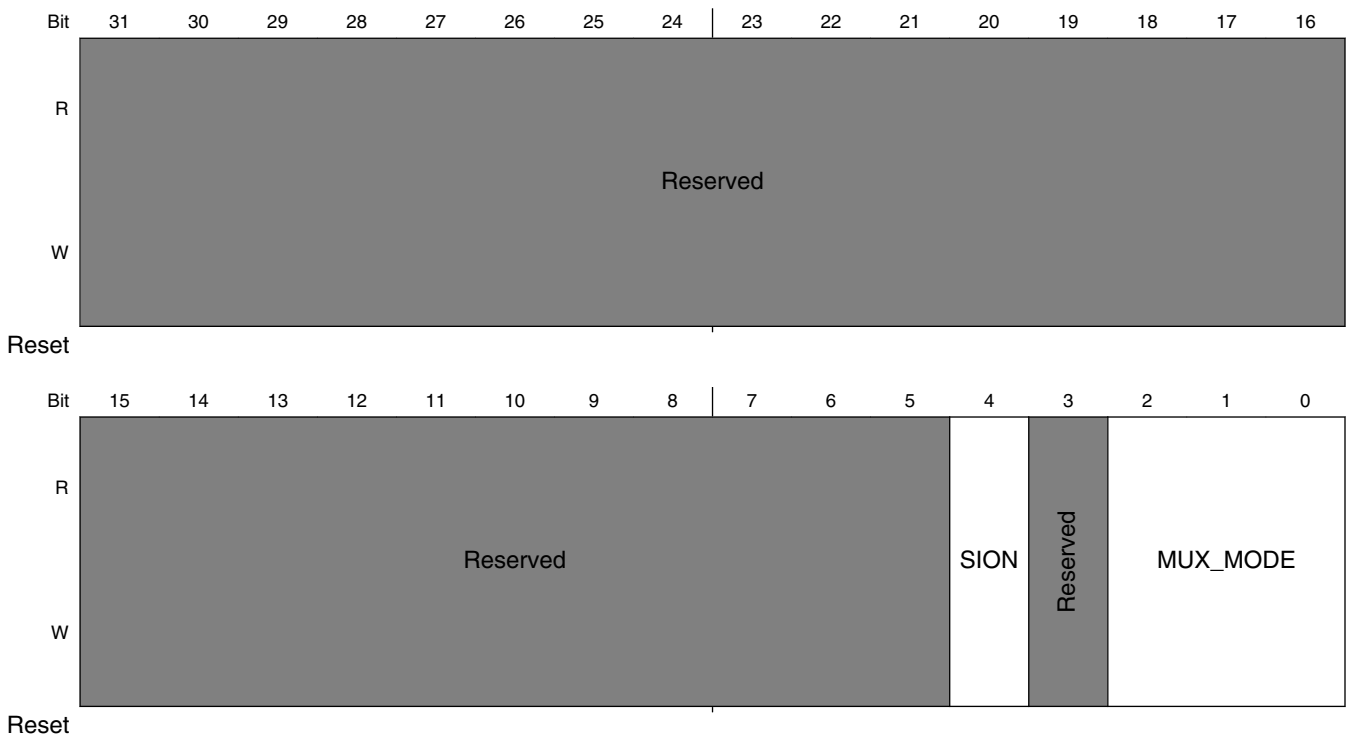
IOMUXC_SW_MUX_CTL_PAD_SD1_DATA5 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD1_DATA5. 000 **ALT0_USDHC1_DATA5** — Select mux mode: ALT0 mux port: USDHC1_DATA5 of instance: usdhc1 001 **ALT1_ENET1_TX_ER** — Select mux mode: ALT1 mux port: ENET1_TX_ER of instance: enet1 011 **ALT3_I2C1_SDA** — Select mux mode: ALT3 mux port: I2C1_SDA of instance: i2c1 100 **ALT4_UART2_CTS_B** — Select mux mode: ALT4 mux port: UART2_CTS_B of instance: uart2 101 **ALT5_GPIO2_IO[7]** — Select mux mode: ALT5 mux port: GPIO2_IO07 of instance: gpio2

8.2.4.39 SW_MUX_CTL_PAD_SD1_DATA6 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA6)

SW_MUX_CTL Register

Address: 3033_0000h base + ACh offset = 3033_00ACh



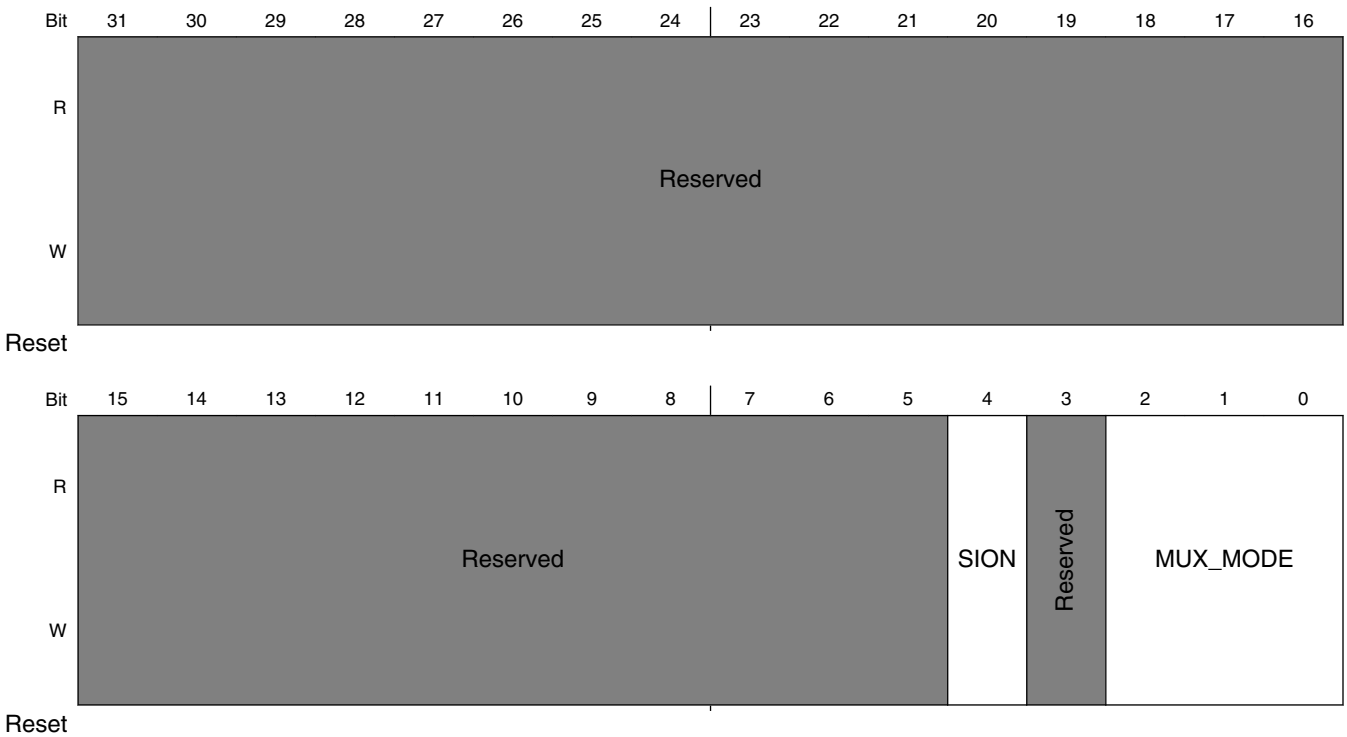
IOMUXC_SW_MUX_CTL_PAD_SD1_DATA6 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_DATA6 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD1_DATA6. 000 **ALT0_USDHC1_DATA6** — Select mux mode: ALT0 mux port: USDHC1_DATA6 of instance: usdhc1 001 **ALT1_ENET1_RGMII_RX_CTL** — Select mux mode: ALT1 mux port: ENET1_RGMII_RX_CTL of instance: enet1 011 **ALT3_I2C2_SCL** — Select mux mode: ALT3 mux port: I2C2_SCL of instance: i2c2 100 **ALT4_UART3_TX** — Select mux mode: ALT4 mux port: UART3_TX of instance: uart3 101 **ALT5_GPIO2_IO[8]** — Select mux mode: ALT5 mux port: GPIO2_IO08 of instance: gpio2

8.2.4.40 SW_MUX_CTL_PAD_SD1_DATA7 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA7)

SW_MUX_CTL Register

Address: 3033_0000h base + B0h offset = 3033_00B0h



IOMUXC_SW_MUX_CTL_PAD_SD1_DATA7 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_DATA7 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD1_DATA7. 000 **ALT0_USDHC1_DATA7** — Select mux mode: ALT0 mux port: USDHC1_DATA7 of instance: usdhc1 001 **ALT1_ENET1_RX_ER** — Select mux mode: ALT1 mux port: ENET1_RX_ER of instance: enet1

Table continues on the next page...

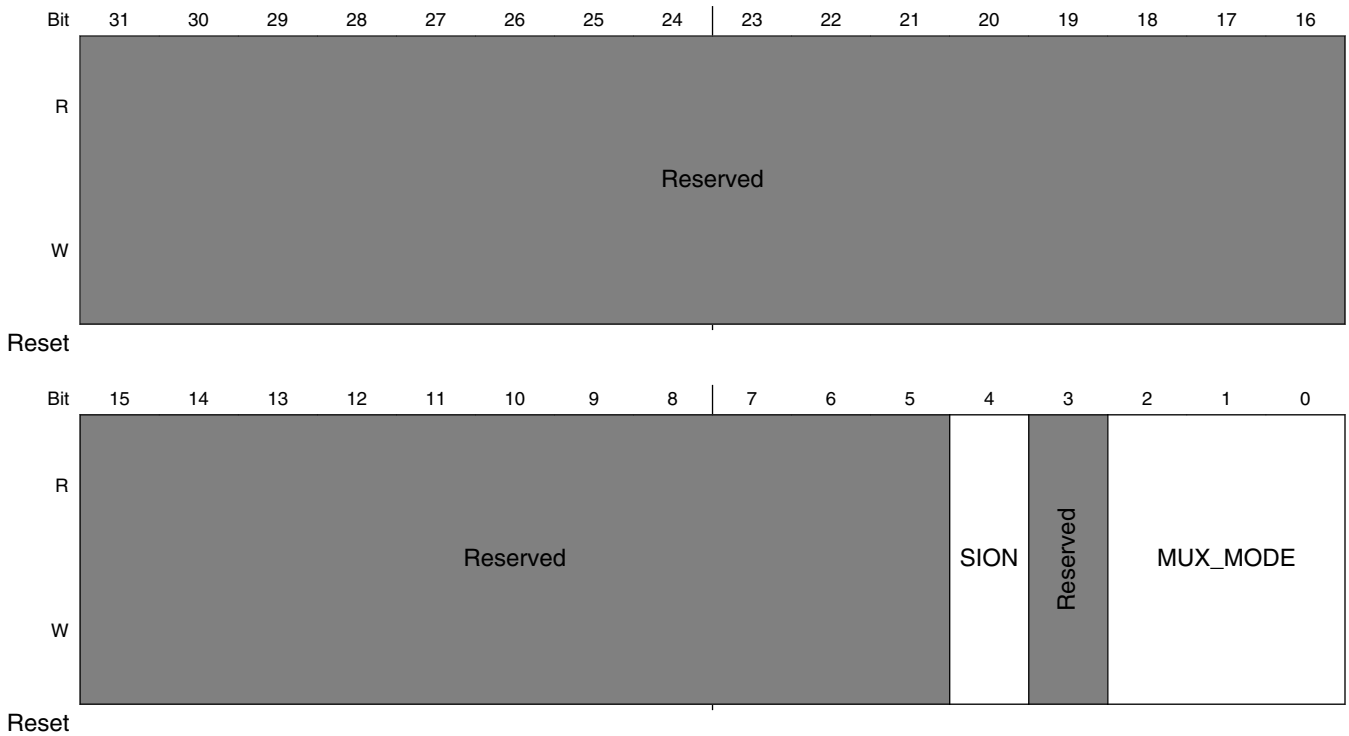
IOMUXC_SW_MUX_CTL_PAD_SD1_DATA7 field descriptions (continued)

Field	Description
	011 **ALT3_I2C2_SDA** — Select mux mode: ALT3 mux port: I2C2_SDA of instance: i2c2 100 **ALT4_UART3_RX** — Select mux mode: ALT4 mux port: UART3_RX of instance: uart3 101 **ALT5_GPIO2_IO[9]** — Select mux mode: ALT5 mux port: GPIO2_IO09 of instance: gpio2

8.2.4.41 SW_MUX_CTL_PAD_SD1_RESET_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_RESET_B)

SW_MUX_CTL Register

Address: 3033_0000h base + B4h offset = 3033_00B4h



IOMUXC_SW_MUX_CTL_PAD_SD1_RESET_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_RESET_B 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

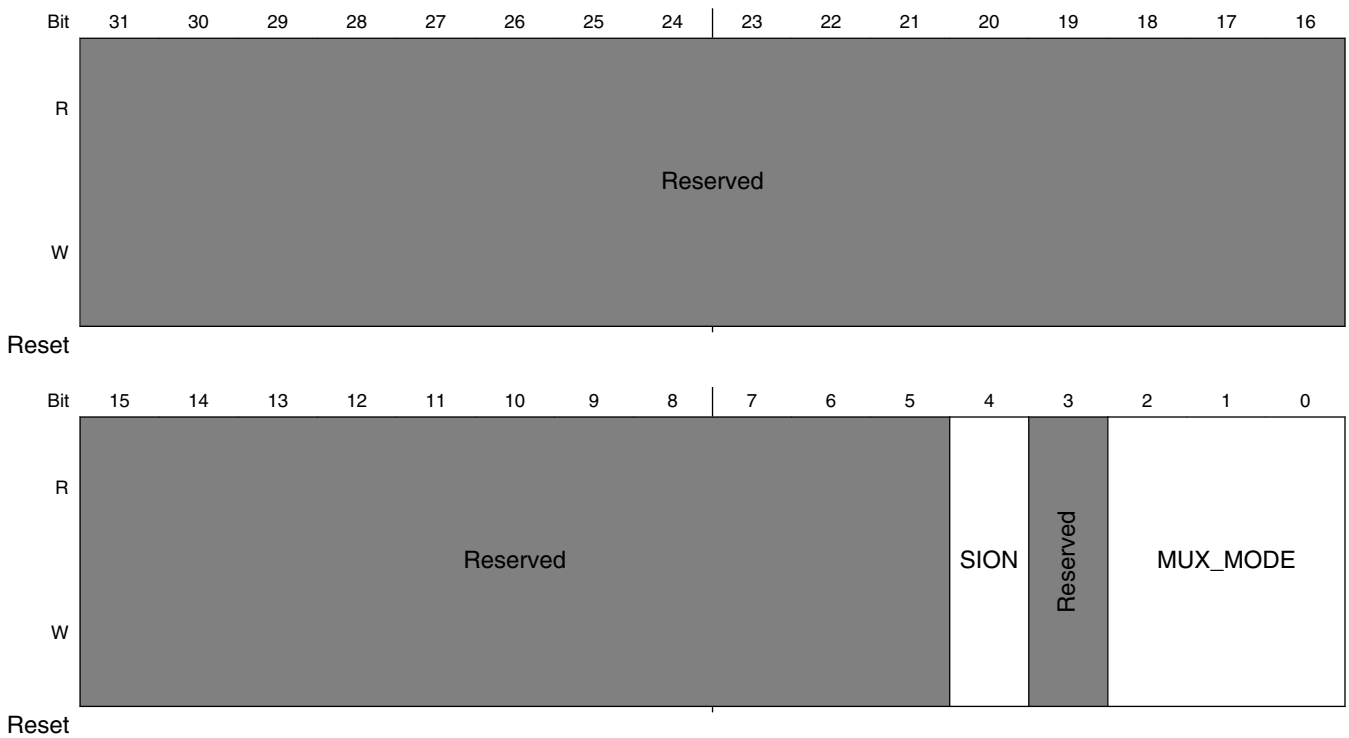
IOMUXC_SW_MUX_CTL_PAD_SD1_RESET_B field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD1_RESET_B. 000 **ALT0_USDHC1_RESET_B** — Select mux mode: ALT0 mux port: USDHC1_RESET_B of instance: usdhc1 001 **ALT1_ENET1_TX_CLK** — Select mux mode: ALT1 mux port: ENET1_TX_CLK of instance: enet1 011 **ALT3_I2C3_SCL** — Select mux mode: ALT3 mux port: I2C3_SCL of instance: i2c3 100 **ALT4_UART3_RTS_B** — Select mux mode: ALT4 mux port: UART3_RTS_B of instance: uart3 101 **ALT5_GPIO2_IO[10]** — Select mux mode: ALT5 mux port: GPIO2_IO10 of instance: gpio2

8.2.4.42 SW_MUX_CTL_PAD_SD1_STROBE SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_STROBE)

SW_MUX_CTL Register

Address: 3033_0000h base + B8h offset = 3033_00B8h



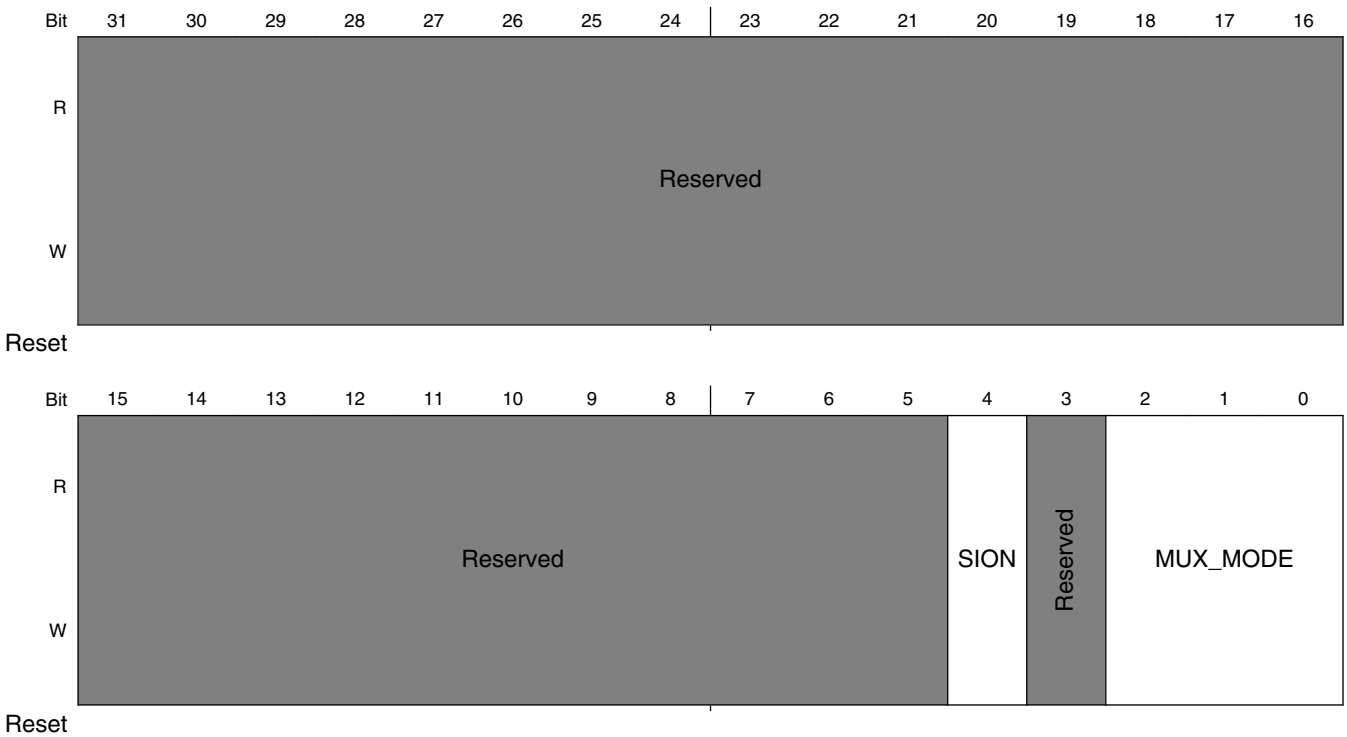
IOMUXC_SW_MUX_CTL_PAD_SD1_STROBE field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD1_STROBE 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD1_STROBE. 000 **ALT0_USDHC1_STROBE** — Select mux mode: ALT0 mux port: USDHC1_STROBE of instance: usdhc1 011 **ALT3_I2C3_SDA** — Select mux mode: ALT3 mux port: I2C3_SDA of instance: i2c3 100 **ALT4_UART3_CTS_B** — Select mux mode: ALT4 mux port: UART3_CTS_B of instance: uart3 101 **ALT5_GPIO2_IO[11]** — Select mux mode: ALT5 mux port: GPIO2_IO11 of instance: gpio2

8.2.4.43 SW_MUX_CTL_PAD_SD2_CD_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CD_B)

SW_MUX_CTL Register

Address: 3033_0000h base + BCh offset = 3033_00BCh



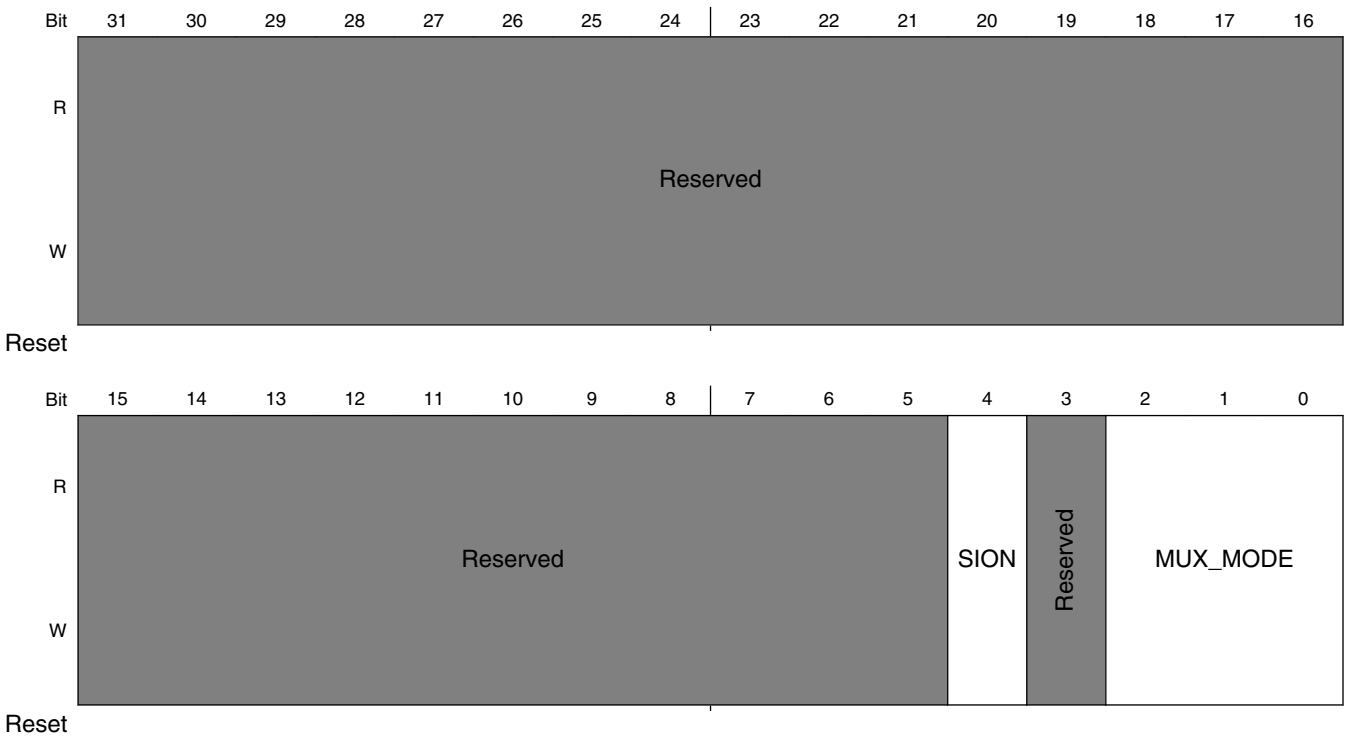
IOMUXC_SW_MUX_CTL_PAD_SD2_CD_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_CD_B 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD2_CD_B. 000 **ALT0_USDHC2_CD_B** — Select mux mode: ALT0 mux port: USDHC2_CD_B of instance: usdhc2 101 **ALT5_GPIO2_IO[12]** — Select mux mode: ALT5 mux port: GPIO2_IO12 of instance: gpio2

8.2.4.44 SW_MUX_CTL_PAD_SD2_CLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CLK)

SW_MUX_CTL Register

Address: 3033_0000h base + C0h offset = 3033_00C0h



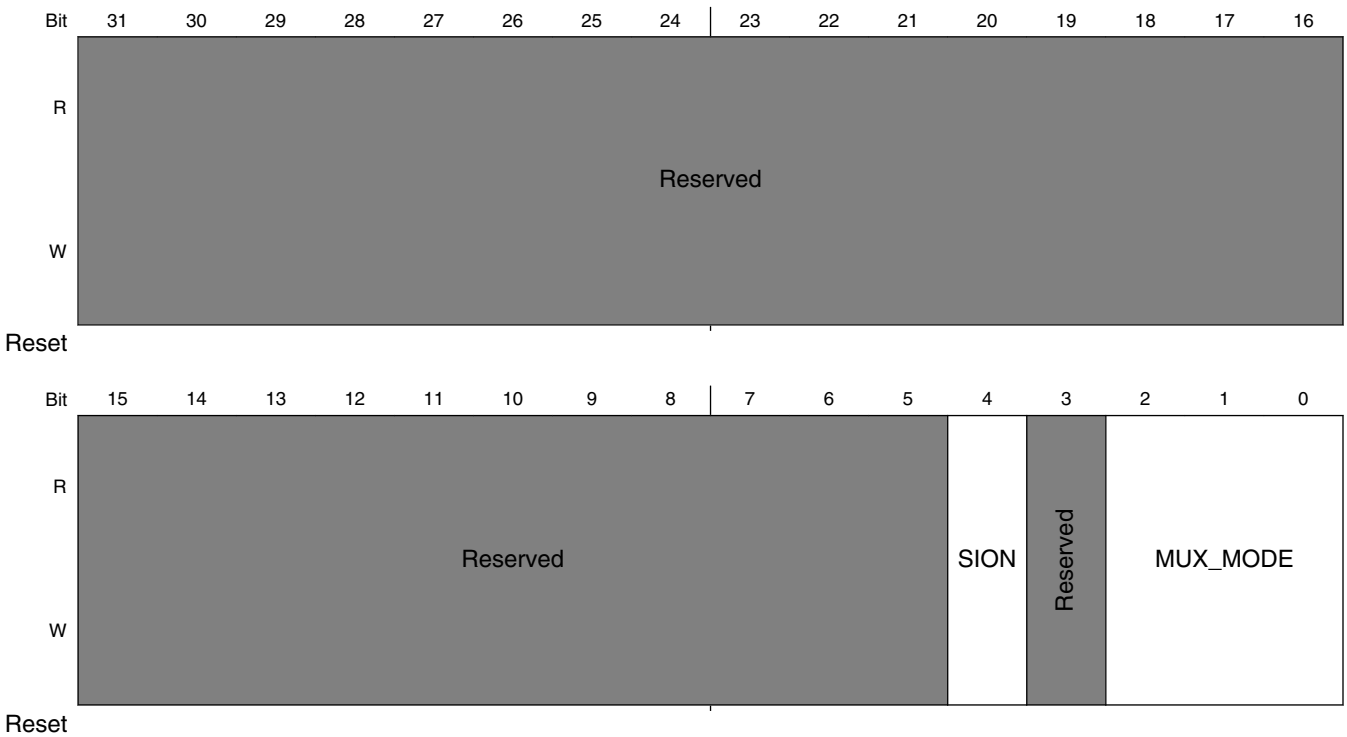
IOMUXC_SW_MUX_CTL_PAD_SD2_CLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_CLK 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD2_CLK. 000 **ALT0_USDHC2_CLK** — Select mux mode: ALT0 mux port: USDHC2_CLK of instance: usdhc2 010 **ALT2_ECSPI2_SCLK** — Select mux mode: ALT2 mux port: ECSPI2_SCLK of instance: ecspi2 011 **ALT3_UART4_RX** — Select mux mode: ALT3 mux port: UART4_RX of instance: uart4 101 **ALT5_GPIO2_IO[13]** — Select mux mode: ALT5 mux port: GPIO2_IO13 of instance: gpio2

8.2.4.45 SW_MUX_CTL_PAD_SD2_CMD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CMD)

SW_MUX_CTL Register

Address: 3033_0000h base + C4h offset = 3033_00C4h



IOMUXC_SW_MUX_CTL_PAD_SD2_CMD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_CMD 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: SD2_CMD. 000 **ALT0_USDHC2_CMD** — Select mux mode: ALT0 mux port: USDHC2_CMD of instance: usdhc2 010 **ALT2_ECSPI2_MOSI** — Select mux mode: ALT2 mux port: ECSPI2_MOSI of instance: ecspi2 011 **ALT3_UART4_TX** — Select mux mode: ALT3 mux port: UART4_TX of instance: uart4

Table continues on the next page...

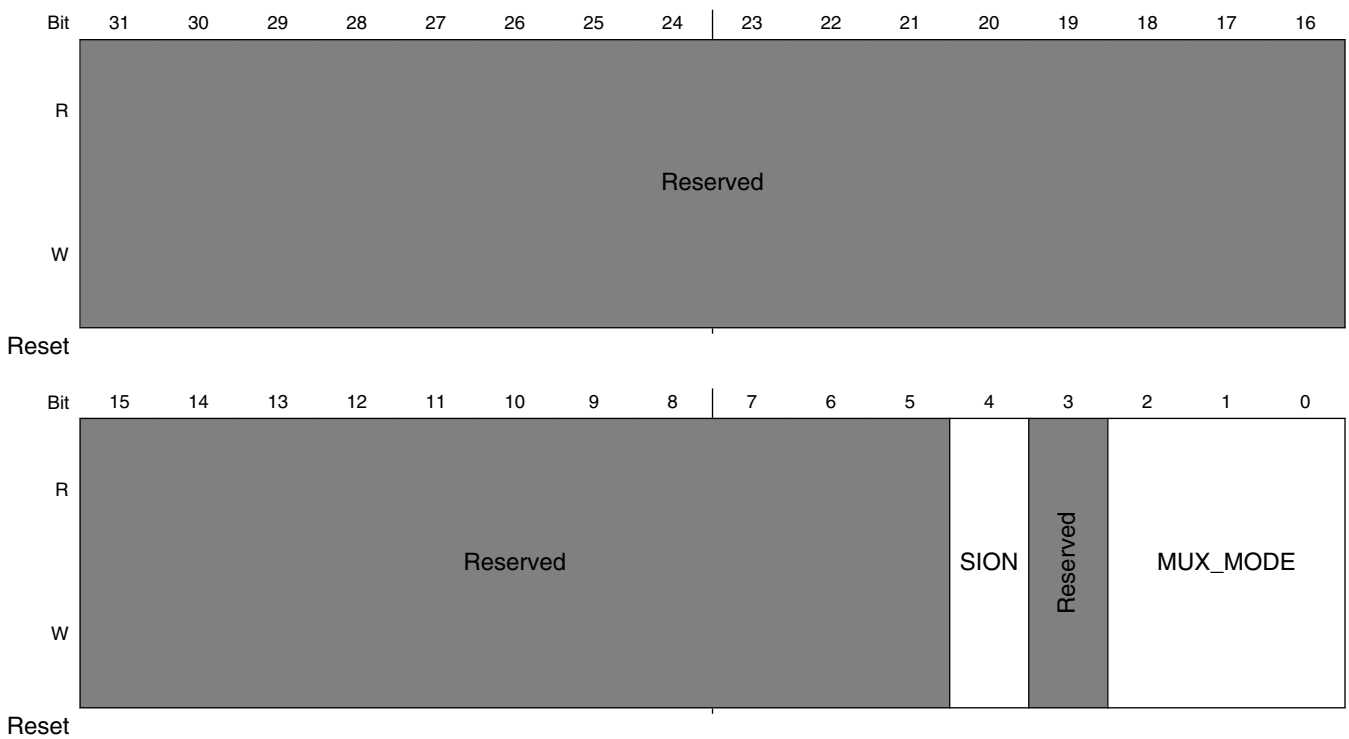
IOMUXC_SW_MUX_CTL_PAD_SD2_CMD field descriptions (continued)

Field	Description
	100 **ALT4_AUDIOMIX_PDM_CLK** — Select mux mode: ALT4 mux port: AUDIOMIX_PDM_CLK of instance: pdm 101 **ALT5_GPIO2_IO[14]** — Select mux mode: ALT5 mux port: GPIO2_IO14 of instance: gpio2

8.2.4.46 SW_MUX_CTL_PAD_SD2_DATA0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0)

SW_MUX_CTL Register

Address: 3033_0000h base + C8h offset = 3033_00C8h



IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_DATA0 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

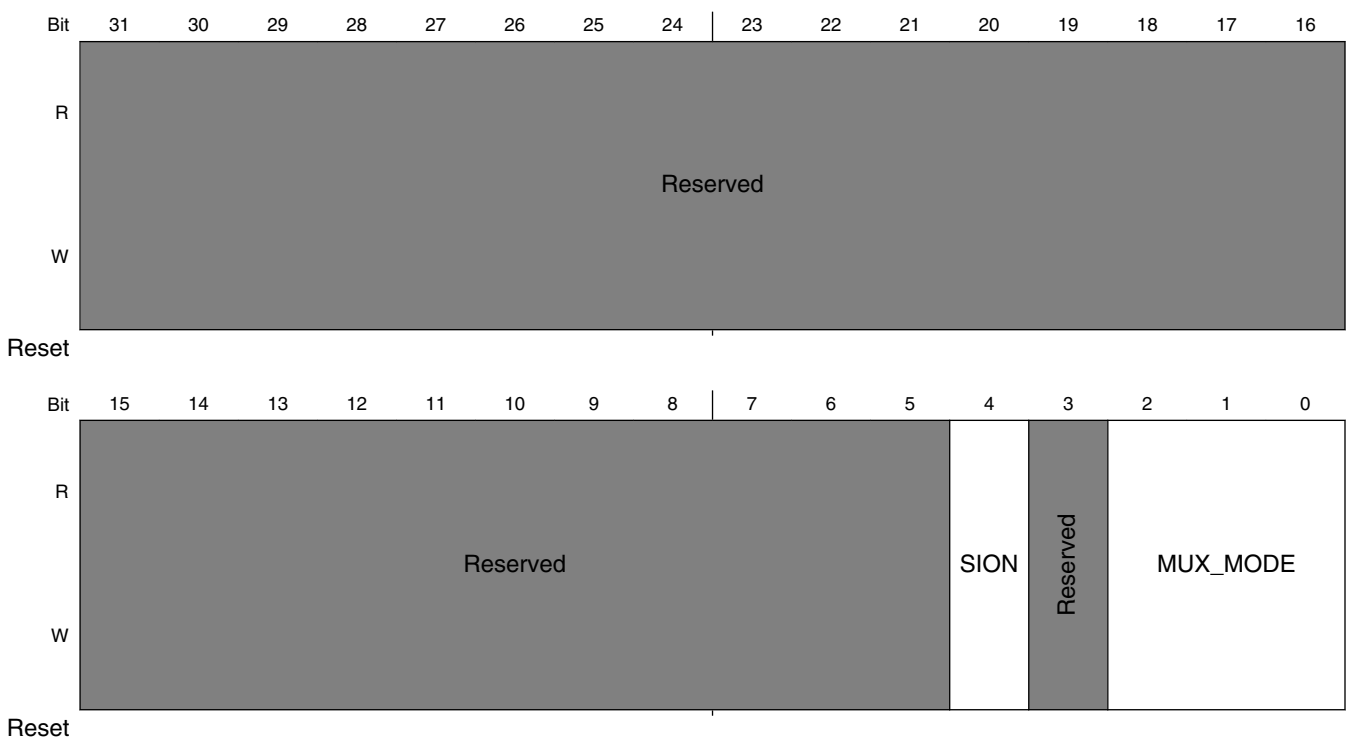
IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: SD2_DATA0. 000 **ALT0_USDHC2_DATA0** — Select mux mode: ALT0 mux port: USDHC2_DATA0 of instance: usdhc2 010 **ALT2_I2C4_SDA** — Select mux mode: ALT2 mux port: I2C4_SDA of instance: i2c4 011 **ALT3_UART2_RX** — Select mux mode: ALT3 mux port: UART2_RX of instance: uart2 100 **ALT4_AUDIOMIX_PDM_BIT_STREAM[0]** — Select mux mode: ALT4 mux port: AUDIOMIX_PDM_BIT_STREAM00 of instance: pdm 101 **ALT5_GPIO2_IO[15]** — Select mux mode: ALT5 mux port: GPIO2_IO15 of instance: gpio2

8.2.4.47 SW_MUX_CTL_PAD_SD2_DATA1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1)

SW_MUX_CTL Register

Address: 3033_0000h base + CCh offset = 3033_00CCh



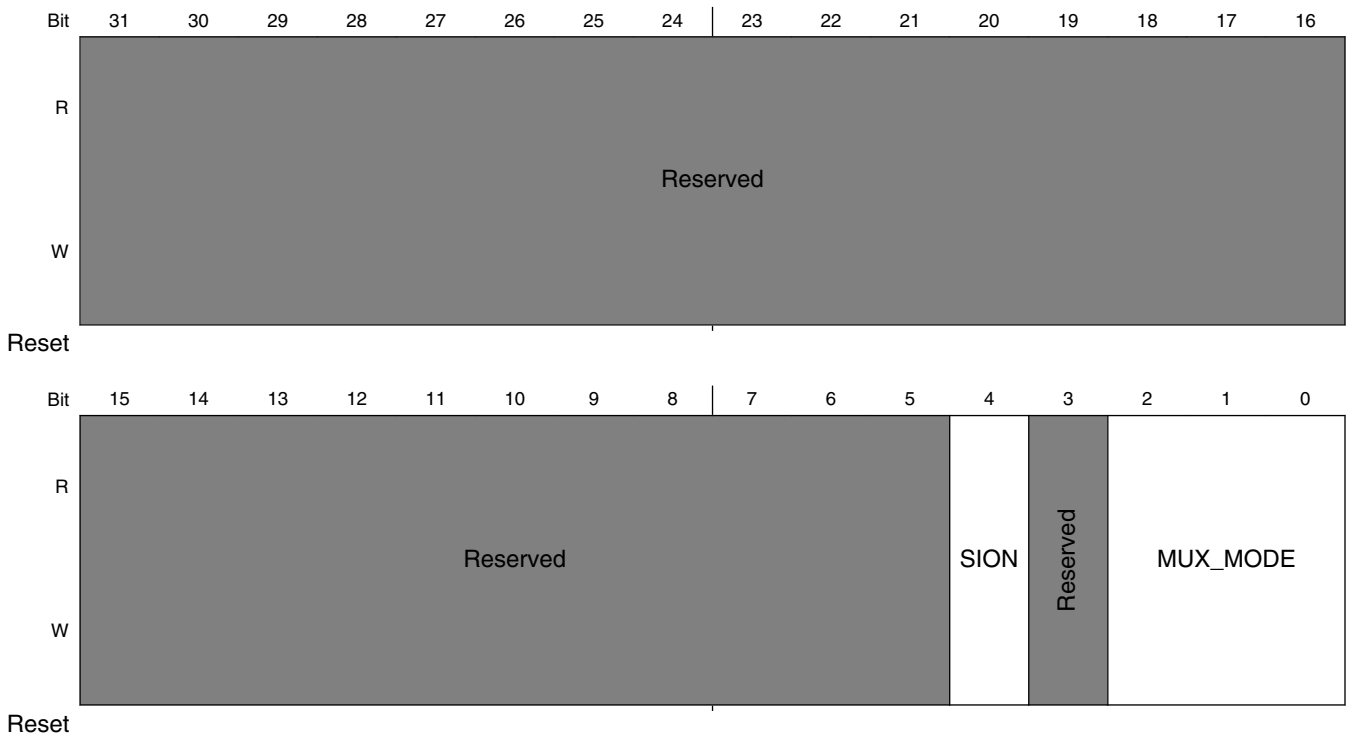
IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_DATA1 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: SD2_DATA1. 000 **ALT0_USDHC2_DATA1** — Select mux mode: ALT0 mux port: USDHC2_DATA1 of instance: usdhc2 010 **ALT2_I2C4_SCL** — Select mux mode: ALT2 mux port: I2C4_SCL of instance: i2c4 011 **ALT3_UART2_TX** — Select mux mode: ALT3 mux port: UART2_TX of instance: uart2 100 **ALT4_AUDIOMIX_PDM_BIT_STREAM[1]** — Select mux mode: ALT4 mux port: AUDIOMIX_PDM_BIT_STREAM01 of instance: pdm 101 **ALT5_GPIO2_IO[16]** — Select mux mode: ALT5 mux port: GPIO2_IO16 of instance: gpio2

8.2.4.48 SW_MUX_CTL_PAD_SD2_DATA2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2)

SW_MUX_CTL Register

Address: 3033_0000h base + D0h offset = 3033_00D0h



IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_DATA2 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: SD2_DATA2. 000 **ALT0_USDHC2_DATA2** — Select mux mode: ALT0 mux port: USDHC2_DATA2 of instance: usdhc2 010 **ALT2_ECSPI2_SS0** — Select mux mode: ALT2 mux port: ECSPI2_SS0 of instance: ecspi2

Table continues on the next page...

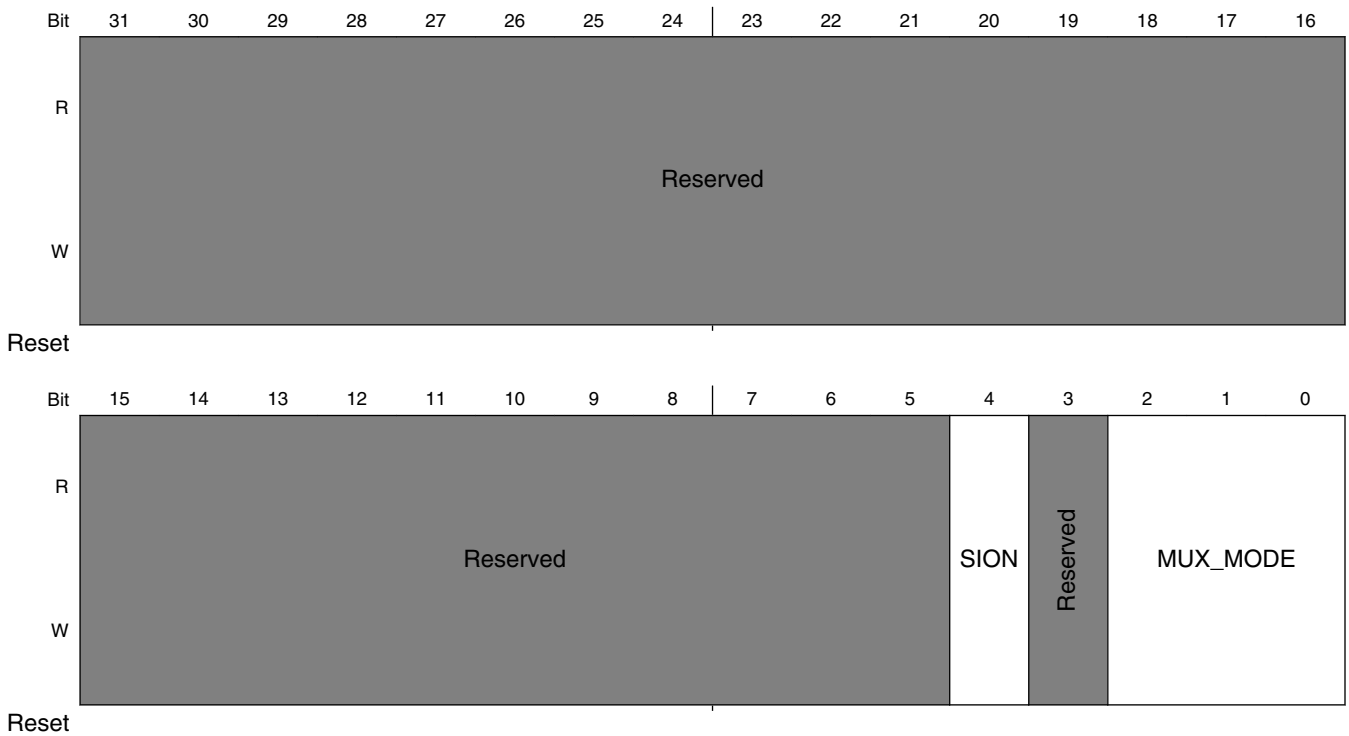
IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2 field descriptions (continued)

Field	Description
	011 **ALT3_AUDIOMIX_SPDIF1_OUT** — Select mux mode: ALT3 mux port: AUDIOMIX_SPDIF1_OUT of instance: spdif 100 **ALT4_AUDIOMIX_PDM_BIT_STREAM[2]** — Select mux mode: ALT4 mux port: AUDIOMIX_PDM_BIT_STREAM02 of instance: pdm 101 **ALT5_GPIO2_IO[17]** — Select mux mode: ALT5 mux port: GPIO2_IO17 of instance: gpio2

8.2.4.49 SW_MUX_CTL_PAD_SD2_DATA3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3)

SW_MUX_CTL Register

Address: 3033_0000h base + D4h offset = 3033_00D4h



IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_DATA3 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

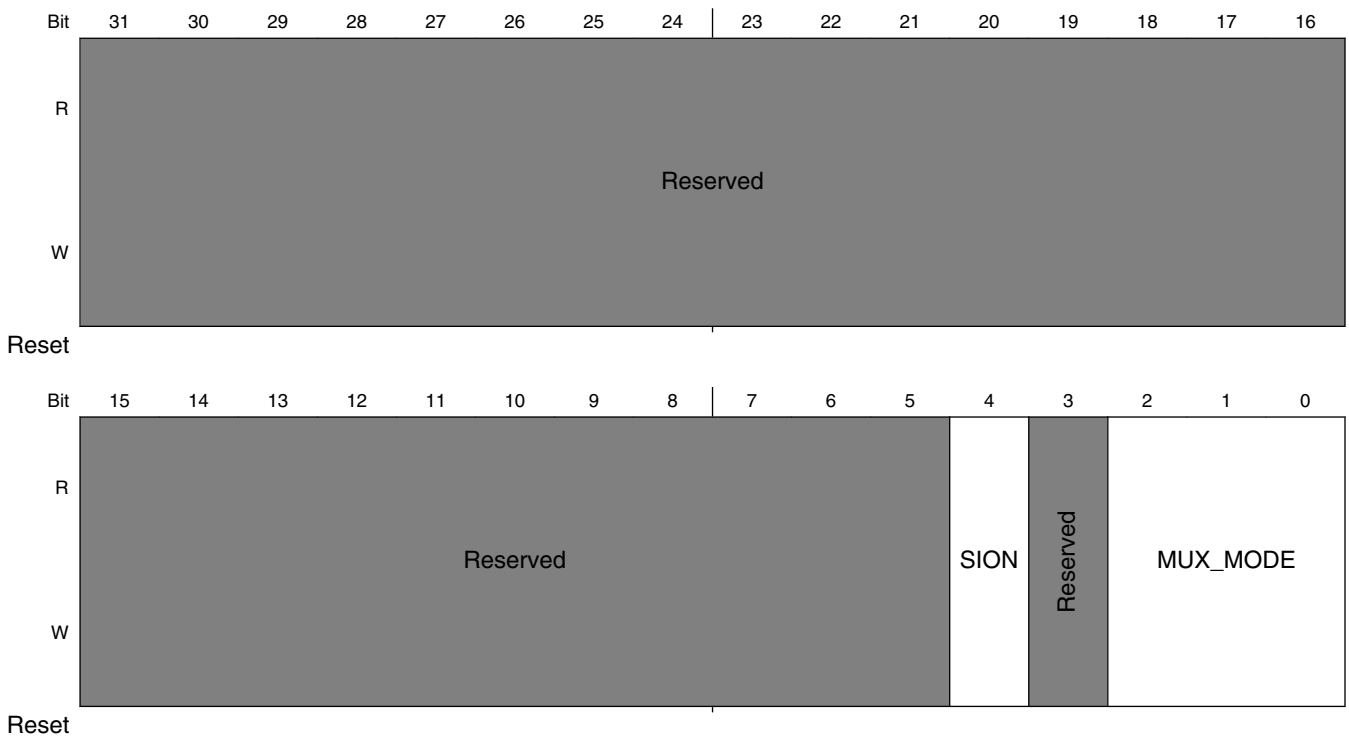
IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD2_DATA3. 000 **ALT0_USDHC2_DATA3** — Select mux mode: ALT0 mux port: USDHC2_DATA3 of instance: usdhc2 010 **ALT2_ECSPI2_MISO** — Select mux mode: ALT2 mux port: ECSPI2_MISO of instance: ecspi2 011 **ALT3_AUDIOMIX_SPDIF1_IN** — Select mux mode: ALT3 mux port: AUDIOMIX_SPDIF1_IN of instance: spdif 100 **ALT4_AUDIOMIX_PDM_BIT_STREAM[3]** — Select mux mode: ALT4 mux port: AUDIOMIX_PDM_BIT_STREAM03 of instance: pdm 101 **ALT5_GPIO2_IO[18]** — Select mux mode: ALT5 mux port: GPIO2_IO18 of instance: gpio2

8.2.4.50 SW_MUX_CTL_PAD_SD2_RESET_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_RESET_B)

SW_MUX_CTL Register

Address: 3033_0000h base + D8h offset = 3033_00D8h



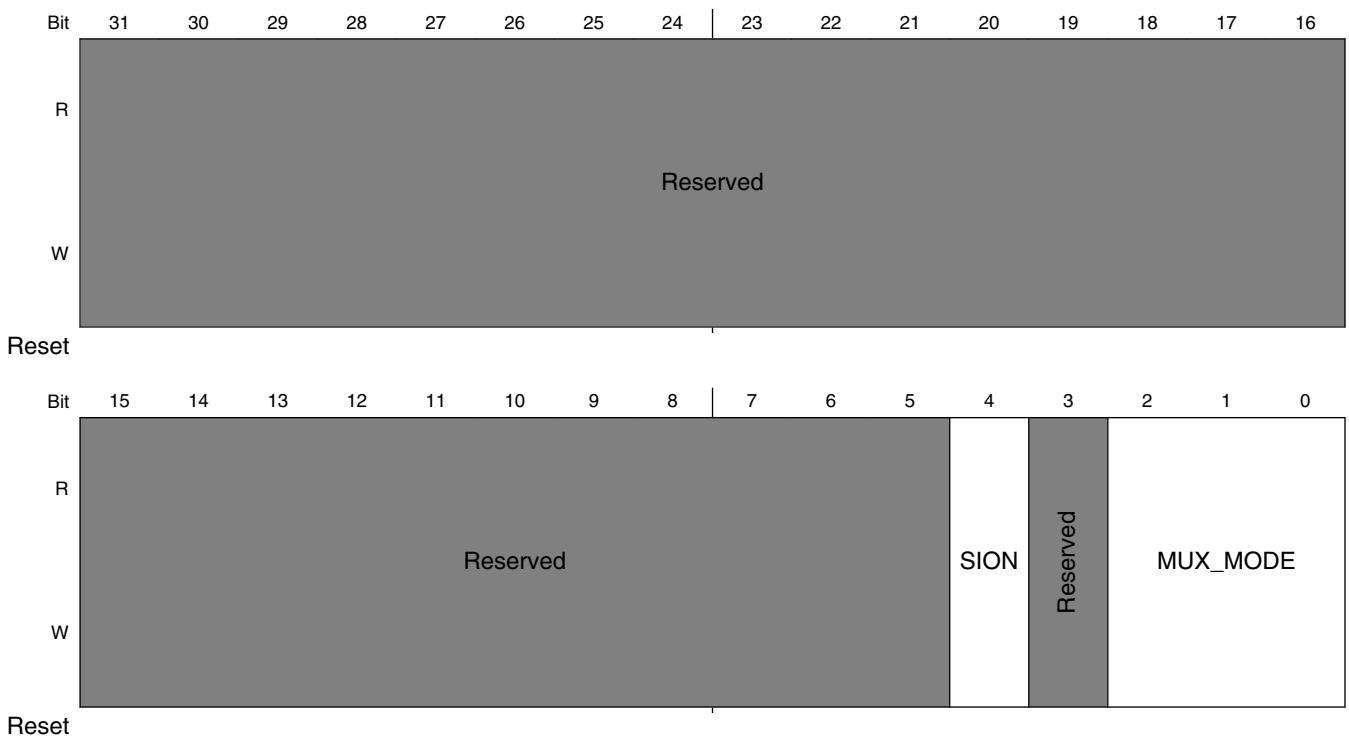
IOMUXC_SW_MUX_CTL_PAD_SD2_RESET_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_RESET_B 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: SD2_RESET_B. 000 **ALT0_USDHC2_RESET_B** — Select mux mode: ALT0 mux port: USDHC2_RESET_B of instance: usdhc2 101 **ALT5_GPIO2_IO[19]** — Select mux mode: ALT5 mux port: GPIO2_IO19 of instance: gpio2

8.2.4.51 SW_MUX_CTL_PAD_SD2_WP SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_WP)

SW_MUX_CTL Register

Address: 3033_0000h base + DCh offset = 3033_00DCh



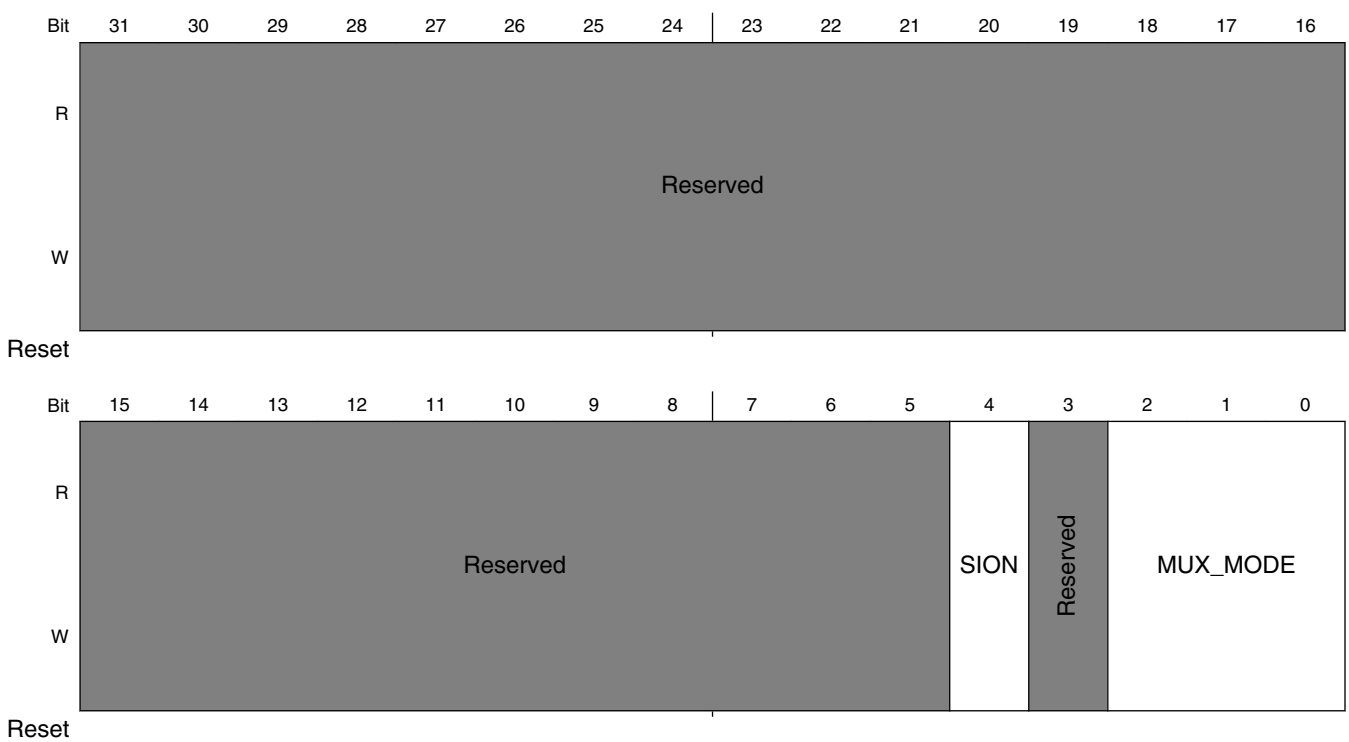
IOMUXC_SW_MUX_CTL_PAD_SD2_WP field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SD2_WP 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SD2_WP. 000 **ALT0_USDHC2_WP** — Select mux mode: ALT0 mux port: USDHC2_WP of instance: usdhc2 101 **ALT5_GPIO2_IO[20]** — Select mux mode: ALT5 mux port: GPIO2_IO20 of instance: gpio2 110 **ALT6_CORESIGHT_EVENTI** — Select mux mode: ALT6 mux port: CORESIGHT_EVENTI of instance: coresight

8.2.4.52 SW_MUX_CTL_PAD_NAND_ALE SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_ALE)

SW_MUX_CTL Register

Address: 3033_0000h base + E0h offset = 3033_00E0h



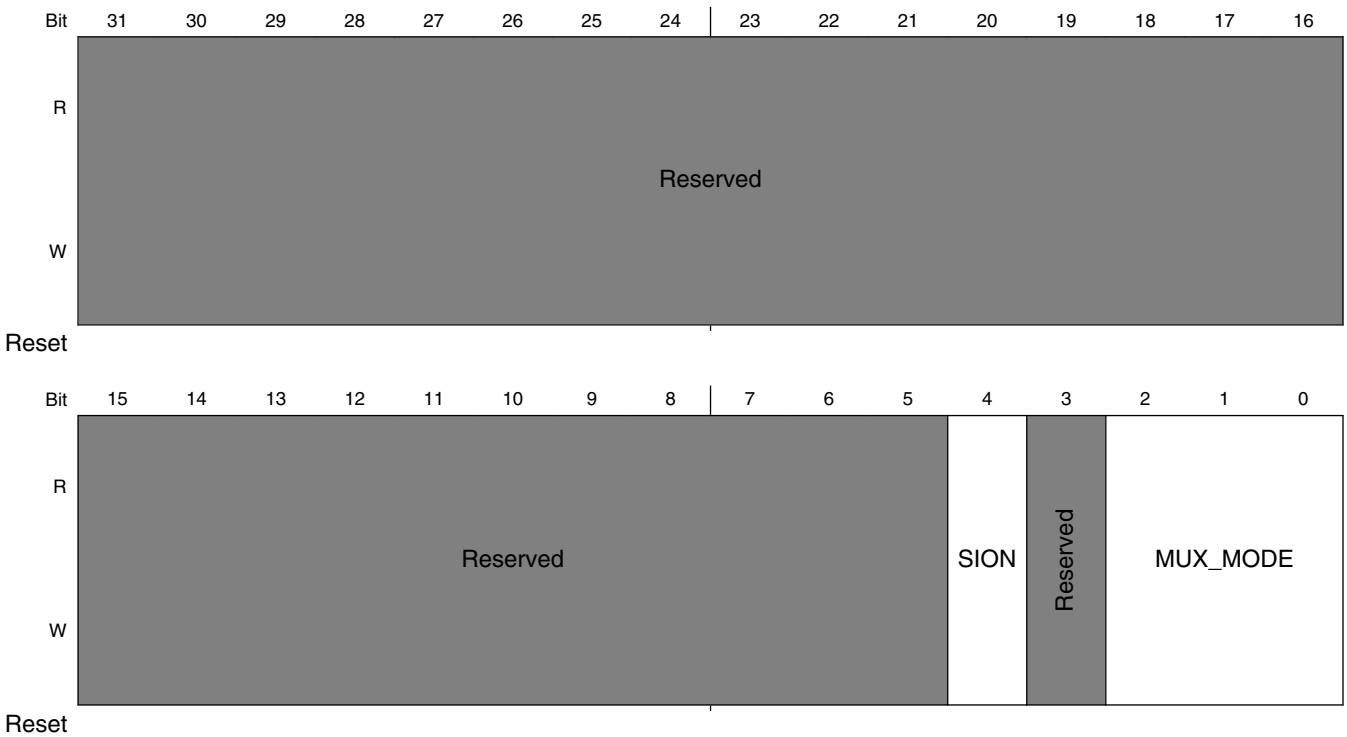
IOMUXC_SW_MUX_CTL_PAD_NAND_ALE field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_ALE 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: NAND_ALE. 000 **ALT0_NAND_ALE** — Select mux mode: ALT0 mux port: NAND_ALE of instance: nand 001 **ALT1_FLEXSPI_A_SCLK** — Select mux mode: ALT1 mux port: FLEXSPI_A_SCLK of instance: flexspi 010 **ALT2_AUDIOMIX_SAI3_TX_BCLK** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI3_TX_BCLK of instance: sai3 011 **ALT3_ISP_FL_TRIG_0** — Select mux mode: ALT3 mux port: ISP_FL_TRIG_0 of instance: isp 100 **ALT4_UART3_RX** — Select mux mode: ALT4 mux port: UART3_RX of instance: uart3 101 **ALT5_GPIO3_IO[0]** — Select mux mode: ALT5 mux port: GPIO3_IO00 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE_CLK** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE_CLK of instance: coresight

8.2.4.53 SW_MUX_CTL_PAD_NAND_CE0_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE0_B)

SW_MUX_CTL Register

Address: 3033_0000h base + E4h offset = 3033_00E4h



IOMUXC_SW_MUX_CTL_PAD_NAND_CE0_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_CE0_B 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: NAND_CE0_B. 000 **ALT0_NAND_CE0_B** — Select mux mode: ALT0 mux port: NAND_CE0_B of instance: nand 001 **ALT1_FLEXSPI_A_SS0_B** — Select mux mode: ALT1 mux port: FLEXSPI_A_SS0_B of instance: flexspi

Table continues on the next page...

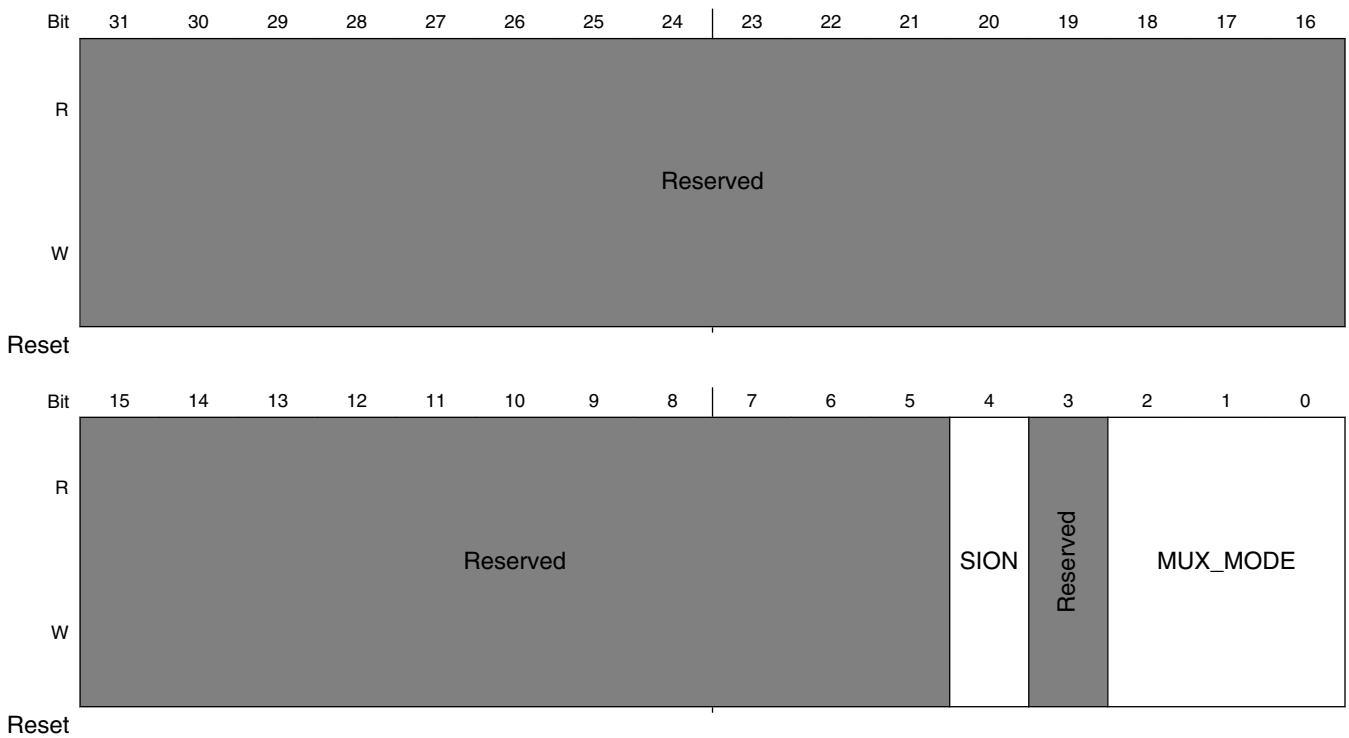
IOMUXC_SW_MUX_CTL_PAD_NAND_CE0_B field descriptions (continued)

Field	Description
	010 **ALT2_AUDIOMIX_SAI3_TX_DATA[0]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI3_TX_DATA00 of instance: sai3 011 **ALT3_ISP_SHUTTER_TRIG_0** — Select mux mode: ALT3 mux port: ISP_SHUTTER_TRIG_0 of instance: isp 100 **ALT4_UART3_TX** — Select mux mode: ALT4 mux port: UART3_TX of instance: uart3 101 **ALT5_GPIO3_IO[1]** — Select mux mode: ALT5 mux port: GPIO3_IO01 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE_CTL** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE_CTL of instance: coresight

8.2.4.54 SW_MUX_CTL_PAD_NAND_CE1_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE1_B)

SW_MUX_CTL Register

Address: 3033_0000h base + E8h offset = 3033_00E8h



IOMUXC_SW_MUX_CTL_PAD_NAND_CE1_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field.

Table continues on the next page...

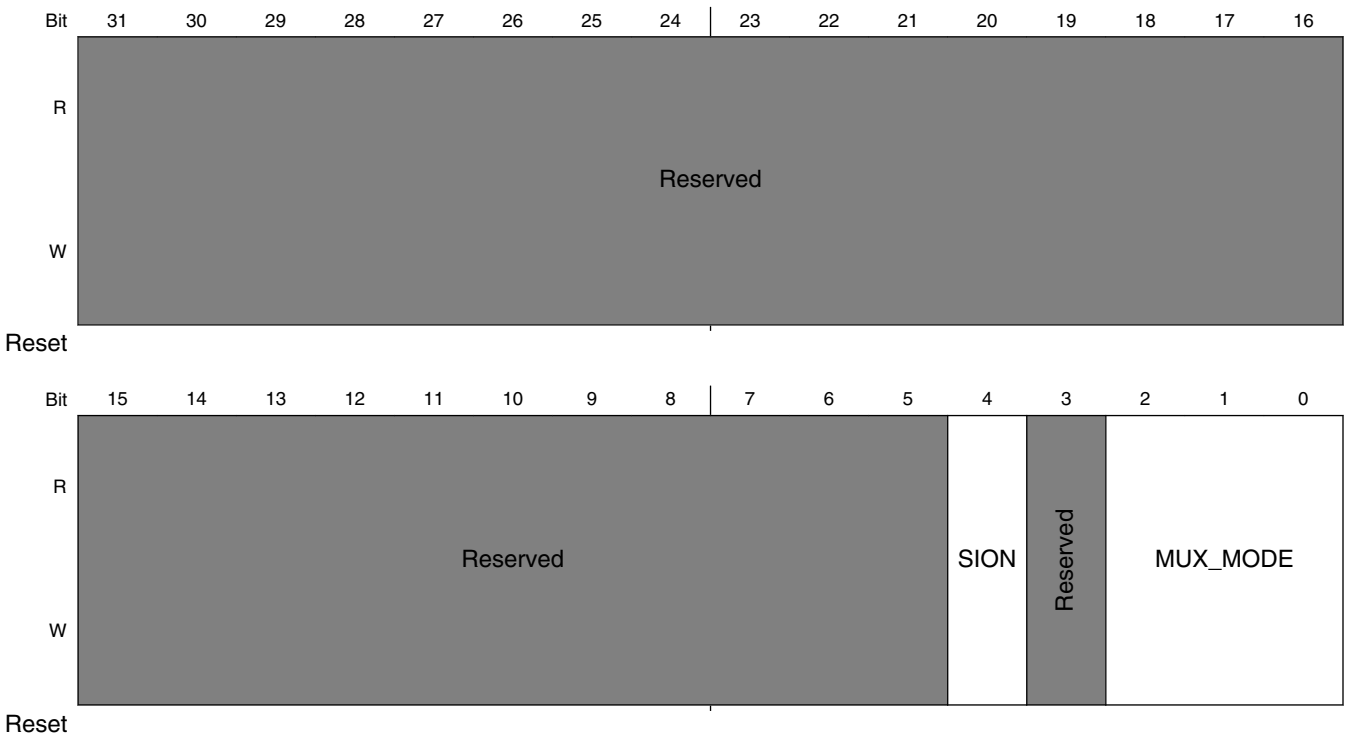
IOMUXC_SW_MUX_CTL_PAD_NAND_CE1_B field descriptions (continued)

Field	Description
	Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_CE1_B 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: NAND_CE1_B. 000 **ALT0_NAND_CE1_B** — Select mux mode: ALT0 mux port: NAND_CE1_B of instance: nand 001 **ALT1_FLEXSPI_A_SS1_B** — Select mux mode: ALT1 mux port: FLEXSPI_A_SS1_B of instance: flexspi 010 **ALT2_USDHC3_STROBE** — Select mux mode: ALT2 mux port: USDHC3_STROBE of instance: usdhc3 100 **ALT4_I2C4_SCL** — Select mux mode: ALT4 mux port: I2C4_SCL of instance: i2c4 101 **ALT5_GPIO3_IO[2]** — Select mux mode: ALT5 mux port: GPIO3_IO02 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[0]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE00 of instance: coresight

8.2.4.55 SW_MUX_CTL_PAD_NAND_CE2_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE2_B)

SW_MUX_CTL Register

Address: 3033_0000h base + ECh offset = 3033_00ECh



IOMUXC_SW_MUX_CTL_PAD_NAND_CE2_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_CE2_B 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: NAND_CE2_B. 000 **ALT0_NAND_CE2_B** — Select mux mode: ALT0 mux port: NAND_CE2_B of instance: nand 001 **ALT1_FLEXSPI_B_SS0_B** — Select mux mode: ALT1 mux port: FLEXSPI_B_SS0_B of instance: flexspi

Table continues on the next page...

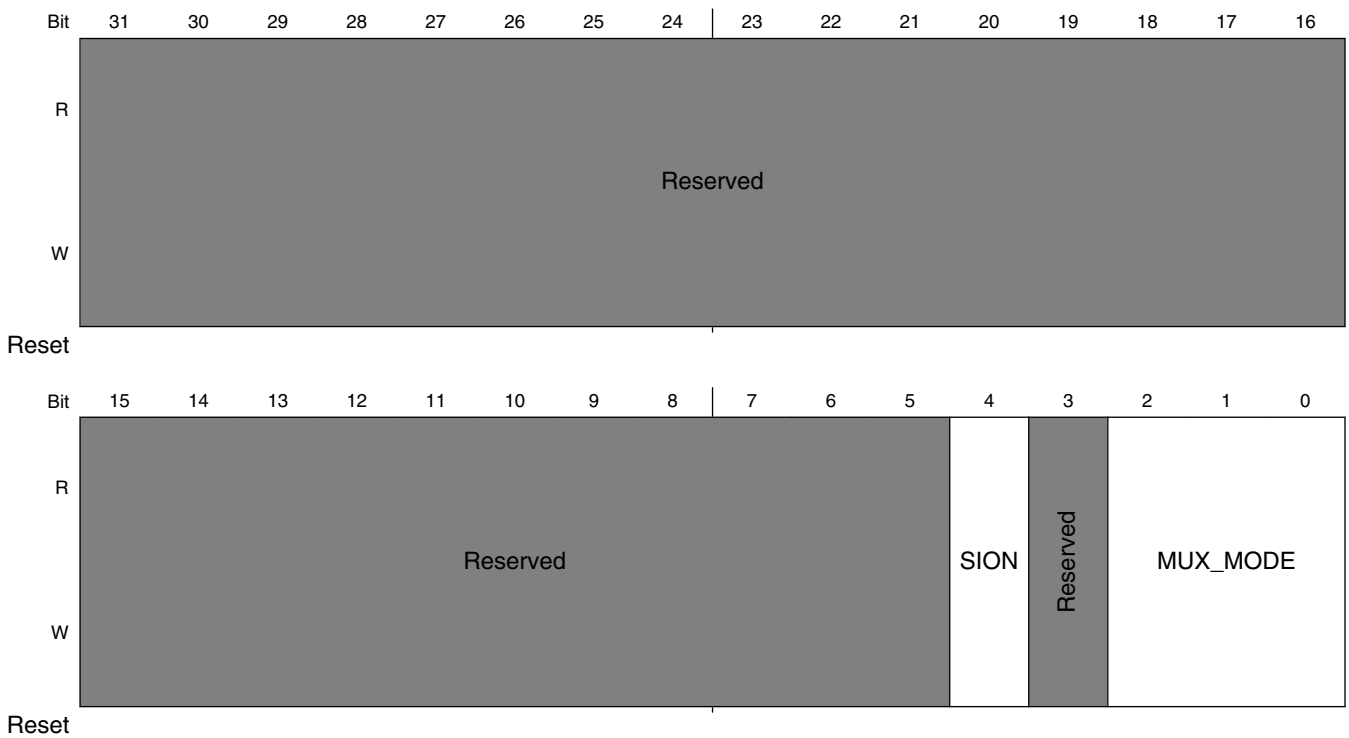
IOMUXC_SW_MUX_CTL_PAD_NAND_CE2_B field descriptions (continued)

Field	Description
	010 **ALT2_USDHC3_DATA5** — Select mux mode: ALT2 mux port: USDHC3_DATA5 of instance: usdhc3 100 **ALT4_I2C4_SDA** — Select mux mode: ALT4 mux port: I2C4_SDA of instance: i2c4 101 **ALT5_GPIO3_IO[3]** — Select mux mode: ALT5 mux port: GPIO3_IO03 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[1]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE01 of instance: coresight

8.2.4.56 SW_MUX_CTL_PAD_NAND_CE3_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE3_B)

SW_MUX_CTL Register

Address: 3033_0000h base + F0h offset = 3033_00F0h



IOMUXC_SW_MUX_CTL_PAD_NAND_CE3_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality.

Table continues on the next page...

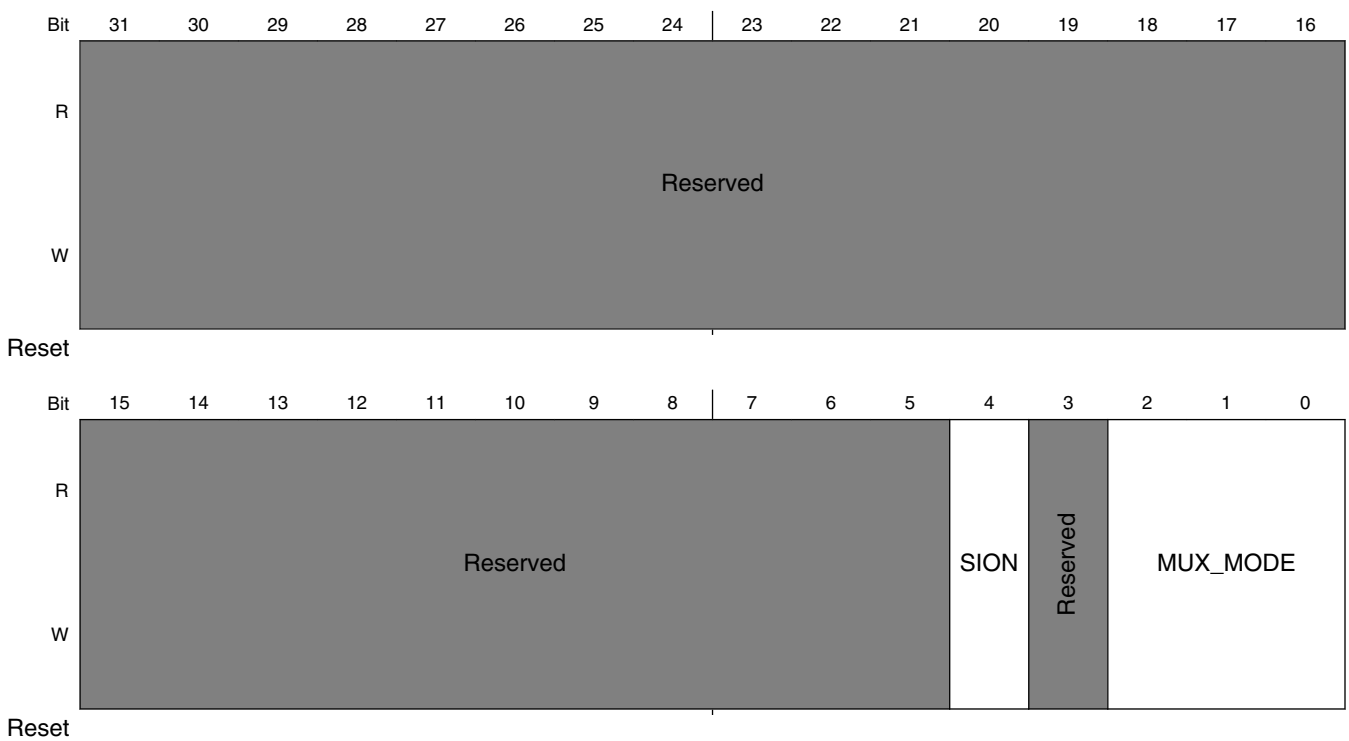
IOMUXC_SW_MUX_CTL_PAD_NAND_CE3_B field descriptions (continued)

Field	Description
	1 **ENABLED** — Force input path of pad NAND_CE3_B 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: NAND_CE3_B. 000 **ALT0_NAND_CE3_B** — Select mux mode: ALT0 mux port: NAND_CE3_B of instance: nand 001 **ALT1_FLEXSPI_B_SS1_B** — Select mux mode: ALT1 mux port: FLEXSPI_B_SS1_B of instance: flexspi 010 **ALT2_USDHC3_DATA6** — Select mux mode: ALT2 mux port: USDHC3_DATA6 of instance: usdhc3 100 **ALT4_I2C3_SDA** — Select mux mode: ALT4 mux port: I2C3_SDA of instance: i2c3 101 **ALT5_GPIO3_IO[4]** — Select mux mode: ALT5 mux port: GPIO3_IO04 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[2]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE02 of instance: coresight

8.2.4.57 SW_MUX_CTL_PAD_NAND_CLE SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CLE)

SW_MUX_CTL Register

Address: 3033_0000h base + F4h offset = 3033_00F4h



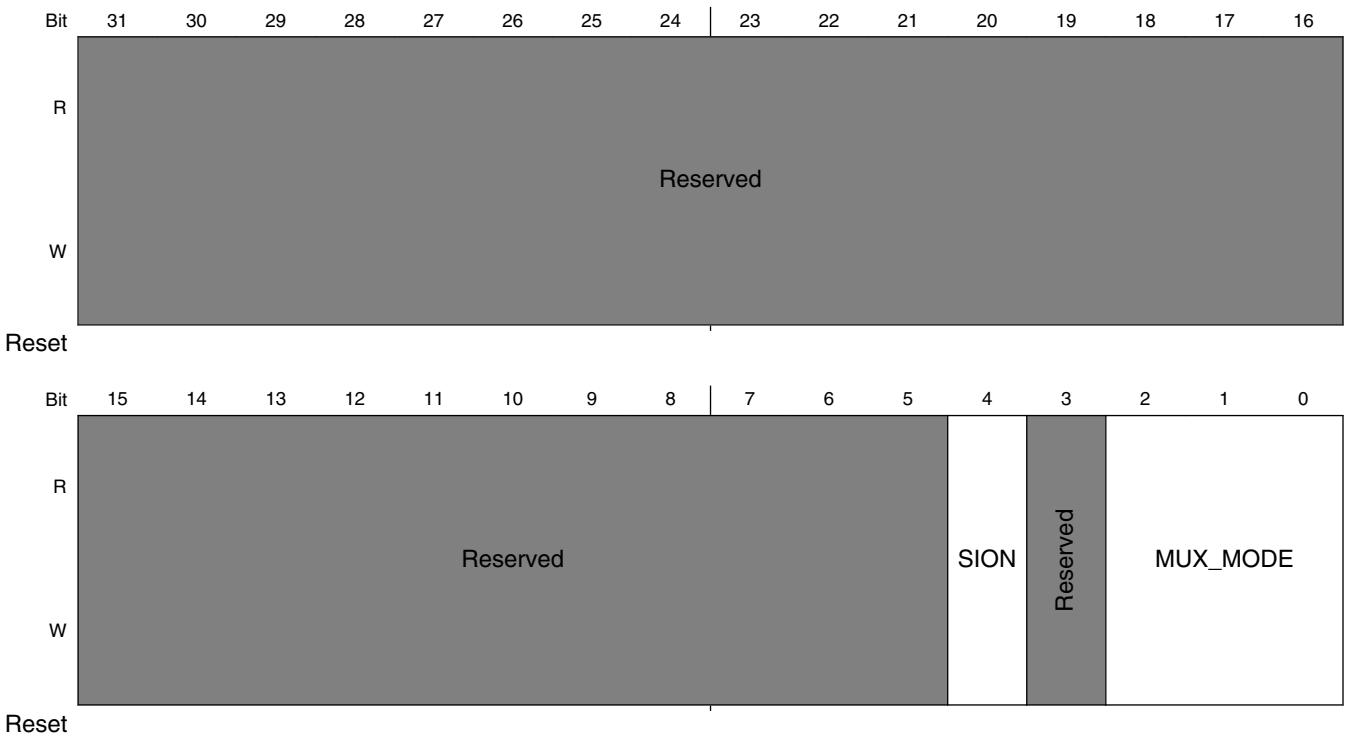
IOMUXC_SW_MUX_CTL_PAD_NAND_CLE field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_CLE 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: NAND_CLE. 000 **ALT0_NAND_CLE** — Select mux mode: ALT0 mux port: NAND_CLE of instance: nand 001 **ALT1_FLEXSPI_B_SCLK** — Select mux mode: ALT1 mux port: FLEXSPI_B_SCLK of instance: flexspi 010 **ALT2_USDHC3_DATA7** — Select mux mode: ALT2 mux port: USDHC3_DATA7 of instance: usdhc3 100 **ALT4_UART4_RX** — Select mux mode: ALT4 mux port: UART4_RX of instance: uart4 101 **ALT5_GPIO3_IO[5]** — Select mux mode: ALT5 mux port: GPIO3_IO05 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[3]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE03 of instance: coresight

8.2.4.58 SW_MUX_CTL_PAD_NAND_DATA00 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00)

SW_MUX_CTL Register

Address: 3033_0000h base + F8h offset = 3033_00F8h



IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_DATA00 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: NAND_DATA00. 000 **ALT0_NAND_DATA00** — Select mux mode: ALT0 mux port: NAND_DATA00 of instance: nand 001 **ALT1_FLEXSPI_A_DATA[0]** — Select mux mode: ALT1 mux port: FLEXSPI_A_DATA00 of instance: flexspi

Table continues on the next page...

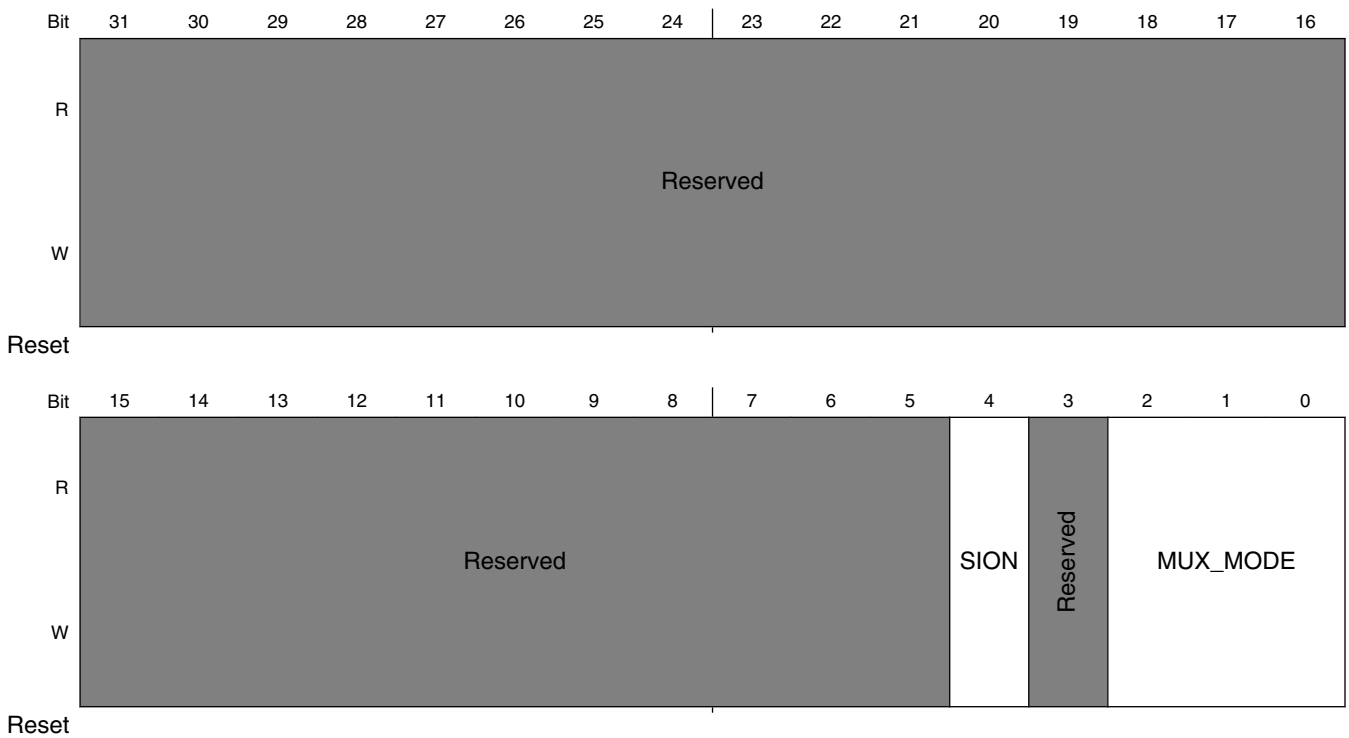
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00 field descriptions (continued)

Field	Description
	010 **ALT2_AUDIOMIX_SAI3_RX_DATA[0]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI3_RX_DATA00 of instance: sai3 011 **ALT3_ISP_FLASH_TRIG_0** — Select mux mode: ALT3 mux port: ISP_FLASH_TRIG_0 of instance: isp 100 **ALT4_UART4_RX** — Select mux mode: ALT4 mux port: UART4_RX of instance: uart4 101 **ALT5_GPIO3_IO[6]** — Select mux mode: ALT5 mux port: GPIO3_IO06 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[4]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE04 of instance: coresight

8.2.4.59 SW_MUX_CTL_PAD_NAND_DATA01 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01)

SW_MUX_CTL Register

Address: 3033_0000h base + FCh offset = 3033_00FCh



IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field.

Table continues on the next page...

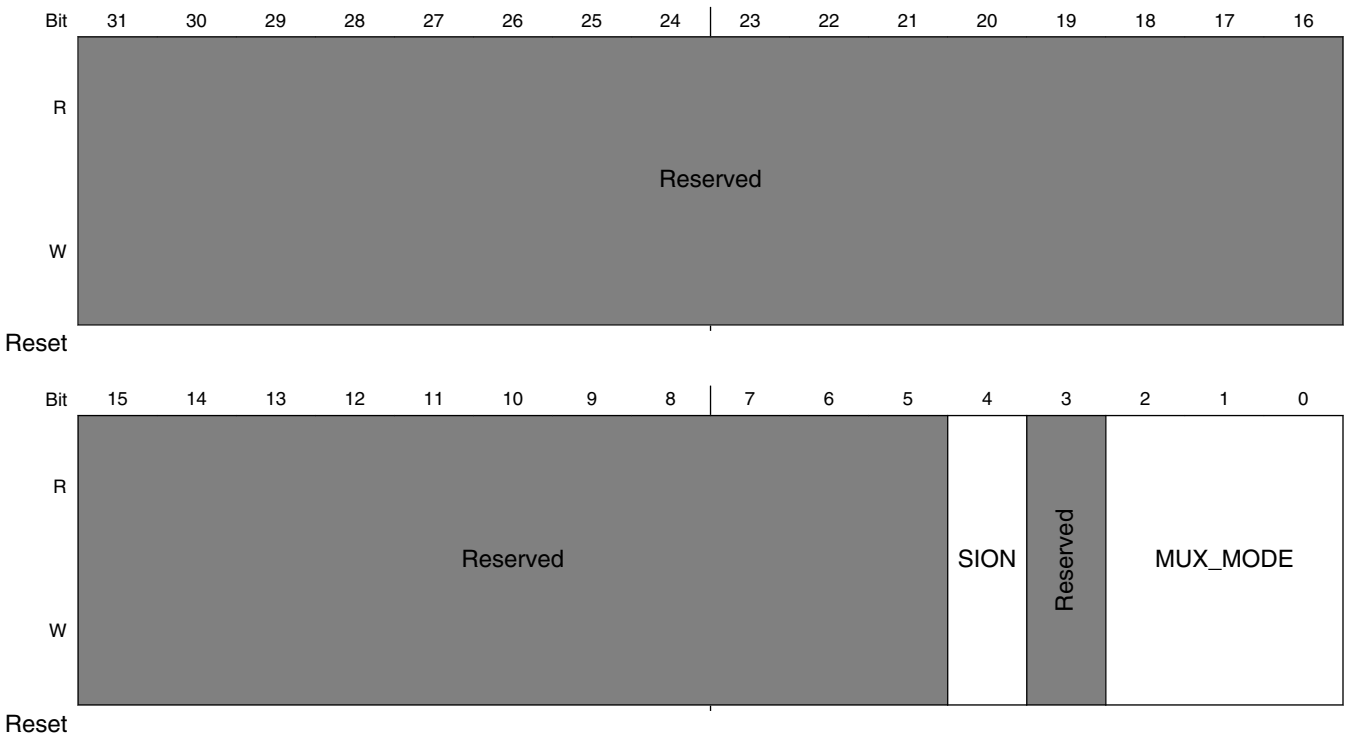
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01 field descriptions (continued)

Field	Description
	Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_DATA01 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: NAND_DATA01. 000 **ALT0_NAND_DATA01** — Select mux mode: ALT0 mux port: NAND_DATA01 of instance: nand 001 **ALT1_FLEXSPI_A_DATA[1]** — Select mux mode: ALT1 mux port: FLEXSPI_A_DATA01 of instance: flexspi 010 **ALT2_AUDIOMIX_SAI3_TX_SYNC** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI3_TX_SYNC of instance: sai3 011 **ALT3_ISP_PRELIGHT_TRIG_0** — Select mux mode: ALT3 mux port: ISP_PRELIGHT_TRIG_0 of instance: isp 100 **ALT4_UART4_TX** — Select mux mode: ALT4 mux port: UART4_TX of instance: uart4 101 **ALT5_GPIO3_IO[7]** — Select mux mode: ALT5 mux port: GPIO3_IO07 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[5]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE05 of instance: coresight

8.2.4.60 SW_MUX_CTL_PAD_NAND_DATA02 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02)

SW_MUX_CTL Register

Address: 3033_0000h base + 100h offset = 3033_0100h



IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_DATA02 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: NAND_DATA02. 000 **ALT0_NAND_DATA02** — Select mux mode: ALT0 mux port: NAND_DATA02 of instance: nand 001 **ALT1_FLEXSPI_A_DATA[2]** — Select mux mode: ALT1 mux port: FLEXSPI_A_DATA02 of instance: flexspi

Table continues on the next page...

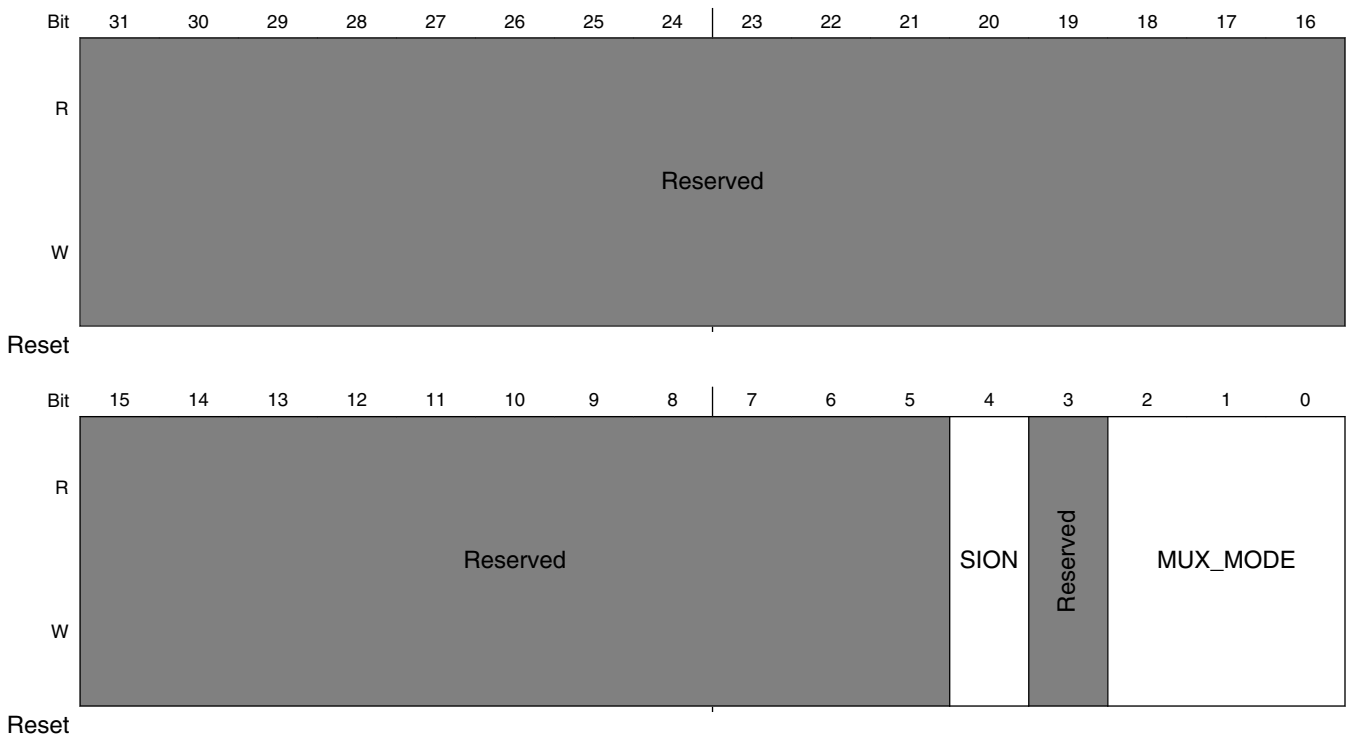
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02 field descriptions (continued)

Field	Description
	010 **ALT2_USDHC3_CD_B** — Select mux mode: ALT2 mux port: USDHC3_CD_B of instance: usdhc3 011 **ALT3_UART4_CTS_B** — Select mux mode: ALT3 mux port: UART4_CTS_B of instance: uart4 100 **ALT4_I2C4_SDA** — Select mux mode: ALT4 mux port: I2C4_SDA of instance: i2c4 101 **ALT5_GPIO3_IO[8]** — Select mux mode: ALT5 mux port: GPIO3_IO08 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[6]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE06 of instance: coresight

8.2.4.61 SW_MUX_CTL_PAD_NAND_DATA03 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03)

SW_MUX_CTL Register

Address: 3033_0000h base + 104h offset = 3033_0104h



IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality.

Table continues on the next page...

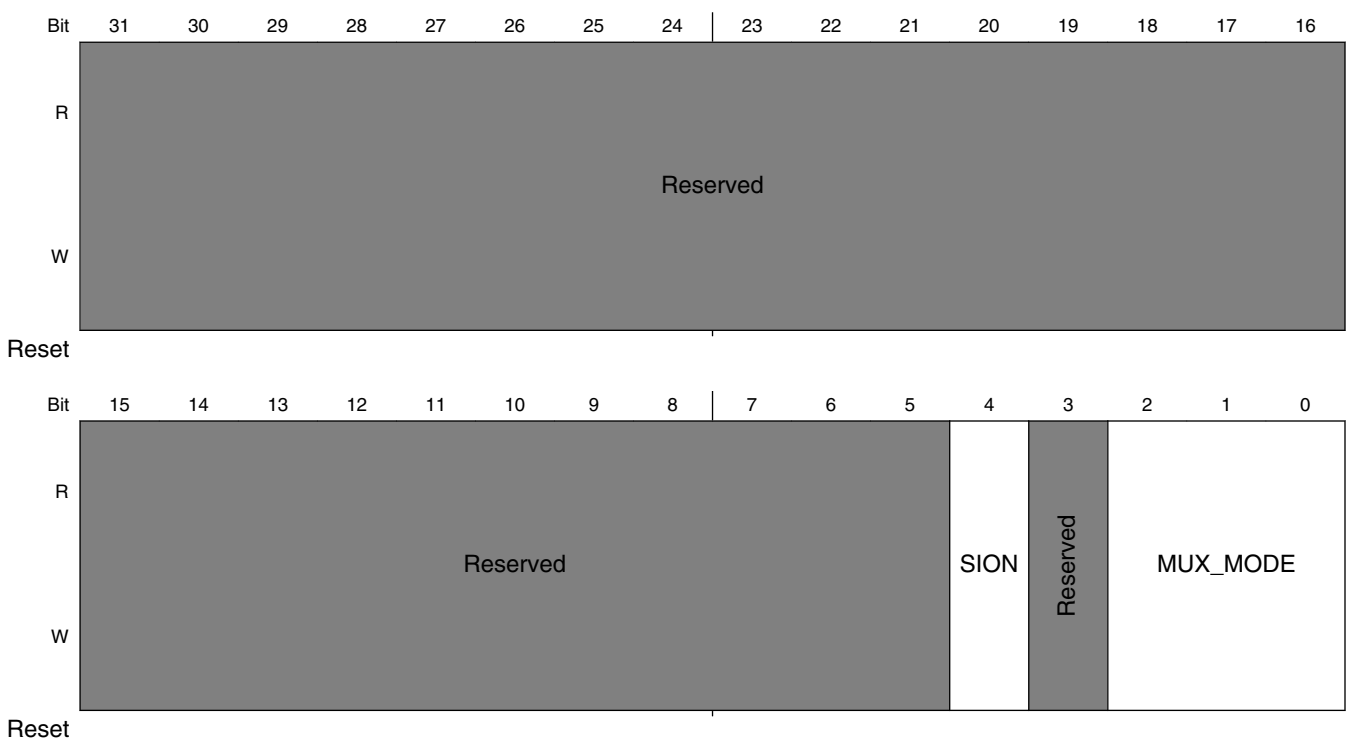
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03 field descriptions (continued)

Field	Description
	1 **ENABLED** — Force input path of pad NAND_DATA03 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: NAND_DATA03. 000 **ALT0_NAND_DATA03** — Select mux mode: ALT0 mux port: NAND_DATA03 of instance: nand 001 **ALT1_FLEXSPI_A_DATA[3]** — Select mux mode: ALT1 mux port: FLEXSPI_A_DATA03 of instance: flexspi 010 **ALT2_USDHC3_WP** — Select mux mode: ALT2 mux port: USDHC3_WP of instance: usdhc3 011 **ALT3_UART4_RTS_B** — Select mux mode: ALT3 mux port: UART4_RTS_B of instance: uart4 100 **ALT4_ISP_FL_TRIG_1** — Select mux mode: ALT4 mux port: ISP_FL_TRIG_1 of instance: isp 101 **ALT5_GPIO3_IO[9]** — Select mux mode: ALT5 mux port: GPIO3_IO09 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[7]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE07 of instance: coresight

8.2.4.62 SW_MUX_CTL_PAD_NAND_DATA04 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA04)

SW_MUX_CTL Register

Address: 3033_0000h base + 108h offset = 3033_0108h



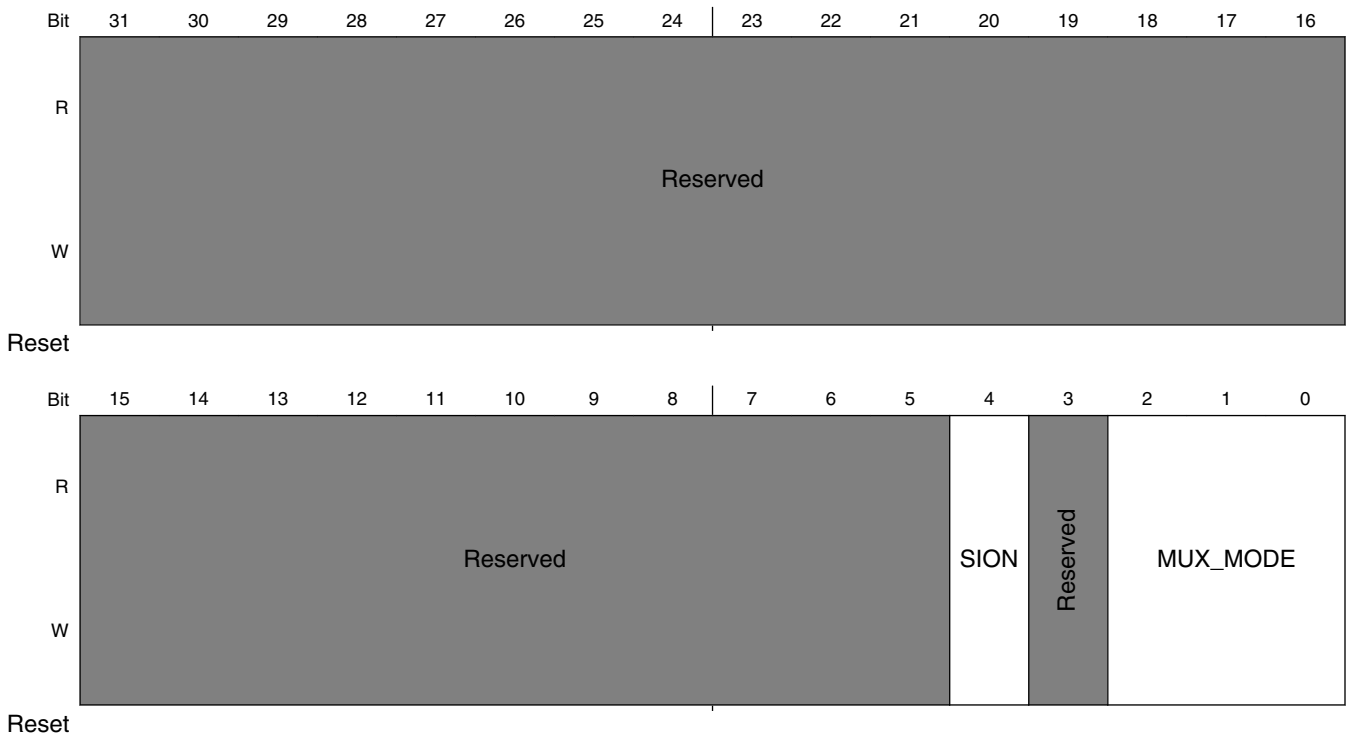
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA04 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_DATA04 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: NAND_DATA04. 000 **ALT0_NAND_DATA04** — Select mux mode: ALT0 mux port: NAND_DATA04 of instance: nand 001 **ALT1_FLEXSPI_B_DATA[0]** — Select mux mode: ALT1 mux port: FLEXSPI_B_DATA00 of instance: flexspi 010 **ALT2_USDHC3_DATA0** — Select mux mode: ALT2 mux port: USDHC3_DATA0 of instance: usdhc3 011 **ALT3_FLEXSPI_A_DATA[4]** — Select mux mode: ALT3 mux port: FLEXSPI_A_DATA04 of instance: flexspi 100 **ALT4_ISP_SHUTTER_TRIG_1** — Select mux mode: ALT4 mux port: ISP_SHUTTER_TRIG_1 of instance: isp 101 **ALT5_GPIO3_IO[10]** — Select mux mode: ALT5 mux port: GPIO3_IO10 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[8]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE08 of instance: coresight

8.2.4.63 SW_MUX_CTL_PAD_NAND_DATA05 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05)

SW_MUX_CTL Register

Address: 3033_0000h base + 10Ch offset = 3033_010Ch



IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_DATA05 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: NAND_DATA05. 000 **ALT0_NAND_DATA05** — Select mux mode: ALT0 mux port: NAND_DATA05 of instance: nand 001 **ALT1_FLEXSPI_B_DATA[1]** — Select mux mode: ALT1 mux port: FLEXSPI_B_DATA01 of instance: flexspi

Table continues on the next page...

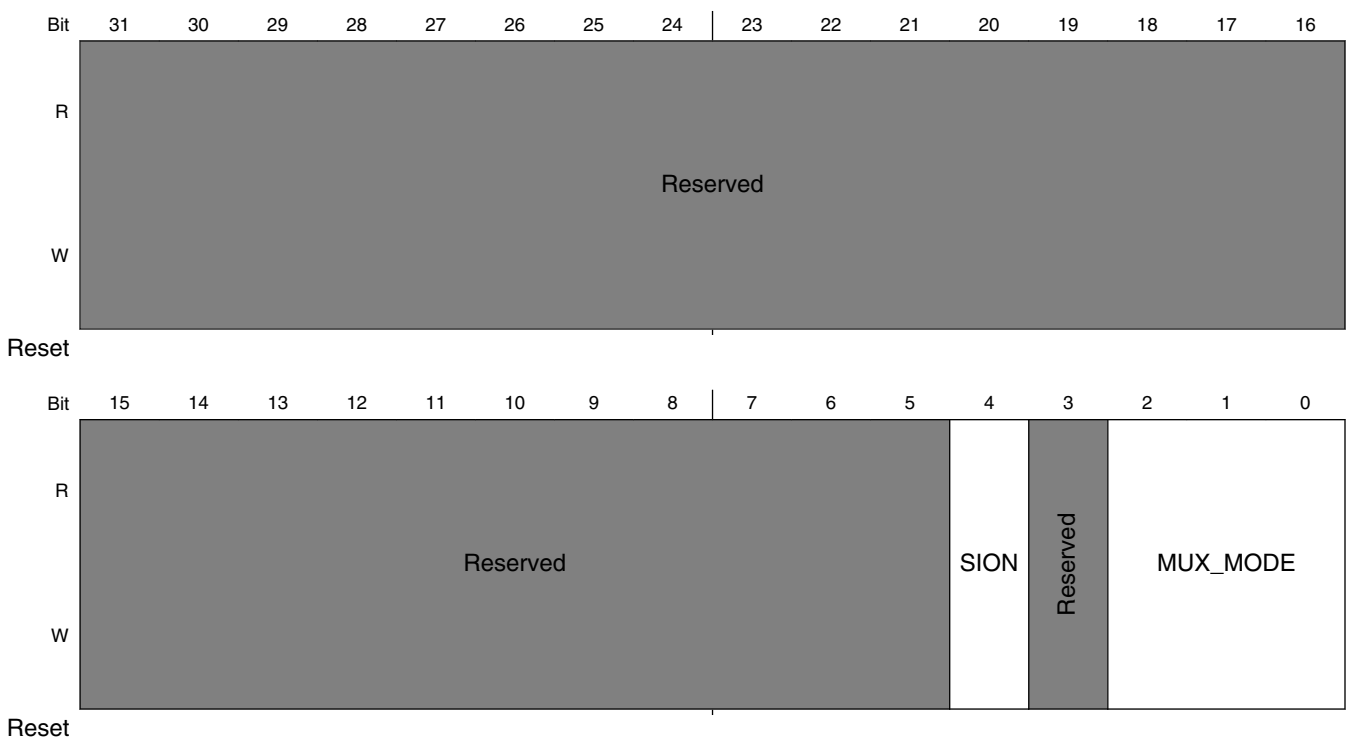
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05 field descriptions (continued)

Field	Description
	010 **ALT2_USDHC3_DATA1** — Select mux mode: ALT2 mux port: USDHC3_DATA1 of instance: usdhc3 011 **ALT3_FLEXSPI_A_DATA[5]** — Select mux mode: ALT3 mux port: FLEXSPI_A_DATA05 of instance: flexspi 100 **ALT4_ISP_FLASH_TRIG_1** — Select mux mode: ALT4 mux port: ISP_FLASH_TRIG_1 of instance: isp 101 **ALT5_GPIO3_IO[11]** — Select mux mode: ALT5 mux port: GPIO3_IO11 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[9]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE09 of instance: coresight

8.2.4.64 SW_MUX_CTL_PAD_NAND_DATA06 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06)

SW_MUX_CTL Register

Address: 3033_0000h base + 110h offset = 3033_0110h



IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved

Table continues on the next page...

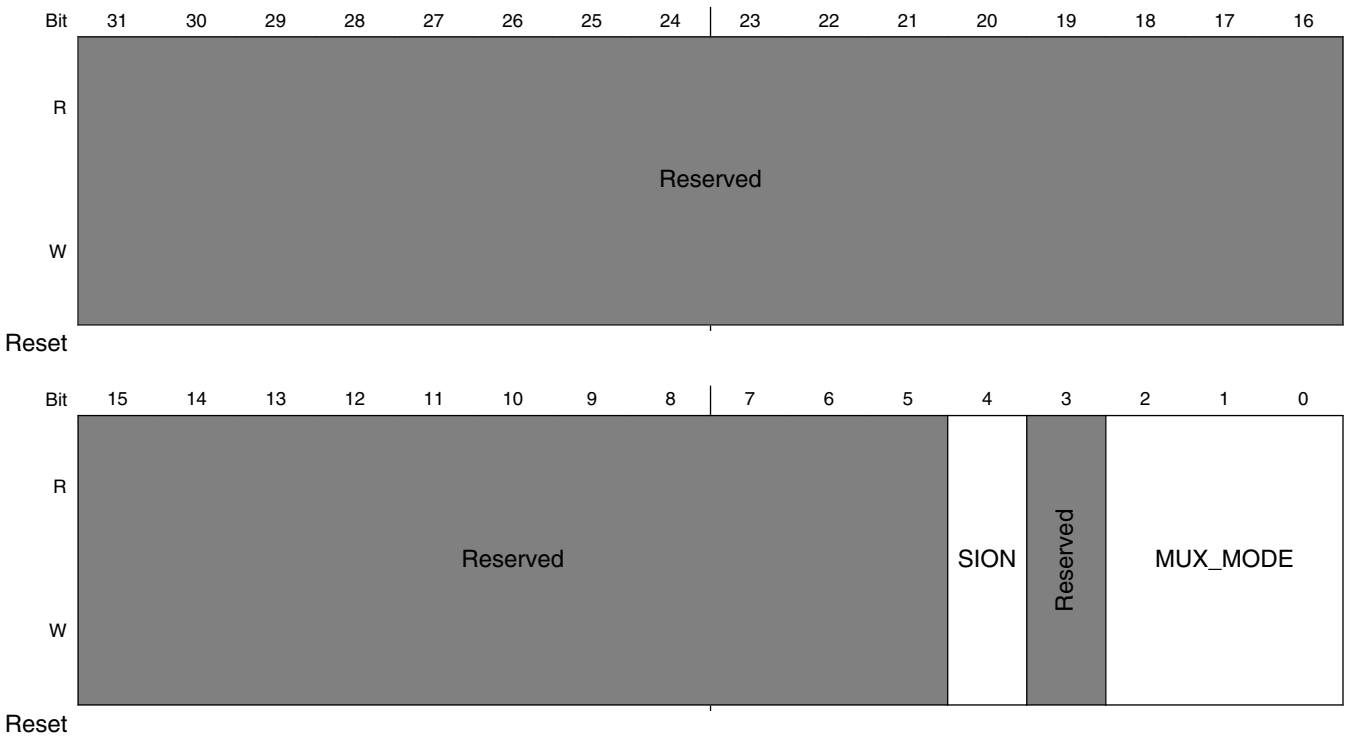
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06 field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_DATA06 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: NAND_DATA06. 000 **ALT0_NAND_DATA06** — Select mux mode: ALT0 mux port: NAND_DATA06 of instance: nand 001 **ALT1_FLEXSPI_B_DATA[2]** — Select mux mode: ALT1 mux port: FLEXSPI_B_DATA02 of instance: flexspi 010 **ALT2_USDHC3_DATA2** — Select mux mode: ALT2 mux port: USDHC3_DATA2 of instance: usdhc3 011 **ALT3_FLEXSPI_A_DATA[6]** — Select mux mode: ALT3 mux port: FLEXSPI_A_DATA06 of instance: flexspi 100 **ALT4_ISP_PRELIGHT_TRIG_1** — Select mux mode: ALT4 mux port: ISP_PRELIGHT_TRIG_1 of instance: isp 101 **ALT5_GPIO3_IO[12]** — Select mux mode: ALT5 mux port: GPIO3_IO12 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[10]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE10 of instance: coresight

8.2.4.65 SW_MUX_CTL_PAD_NAND_DATA07 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07)

SW_MUX_CTL Register

Address: 3033_0000h base + 114h offset = 3033_0114h



IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_DATA07 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: NAND_DATA07. 000 **ALT0_NAND_DATA07** — Select mux mode: ALT0 mux port: NAND_DATA07 of instance: nand 001 **ALT1_FLEXSPI_B_DATA[3]** — Select mux mode: ALT1 mux port: FLEXSPI_B_DATA03 of instance: flexspi

Table continues on the next page...

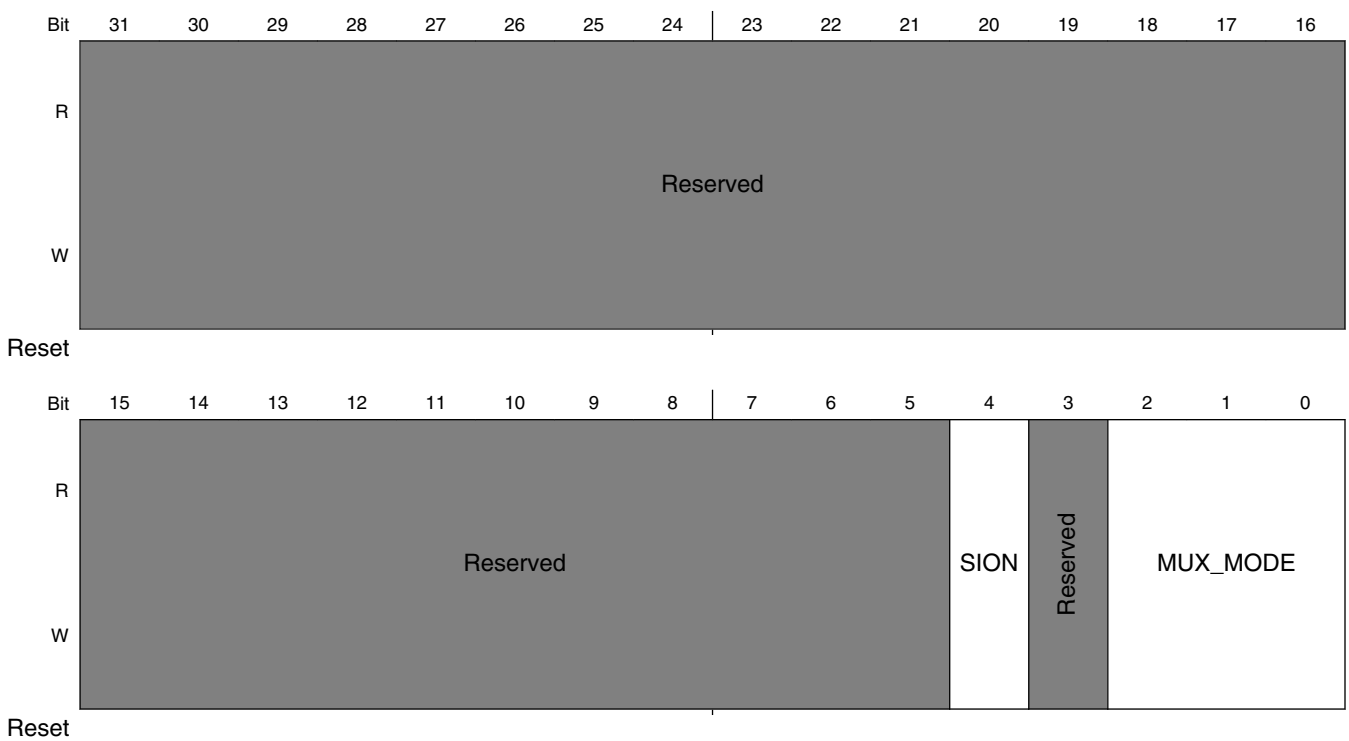
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07 field descriptions (continued)

Field	Description
	010 **ALT2_USDHC3_DATA3** — Select mux mode: ALT2 mux port: USDHC3_DATA3 of instance: usdhc3 011 **ALT3_FLEXSPI_A_DATA[7]** — Select mux mode: ALT3 mux port: FLEXSPI_A_DATA07 of instance: flexspi 100 **ALT4_ISP_SHUTTER_OPEN_1** — Select mux mode: ALT4 mux port: ISP_SHUTTER_OPEN_1 of instance: isp 101 **ALT5_GPIO3_IO[13]** — Select mux mode: ALT5 mux port: GPIO3_IO13 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[11]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE11 of instance: coresight

8.2.4.66 SW_MUX_CTL_PAD_NAND_DQS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DQS)

SW_MUX_CTL Register

Address: 3033_0000h base + 118h offset = 3033_0118h



IOMUXC_SW_MUX_CTL_PAD_NAND_DQS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved

Table continues on the next page...

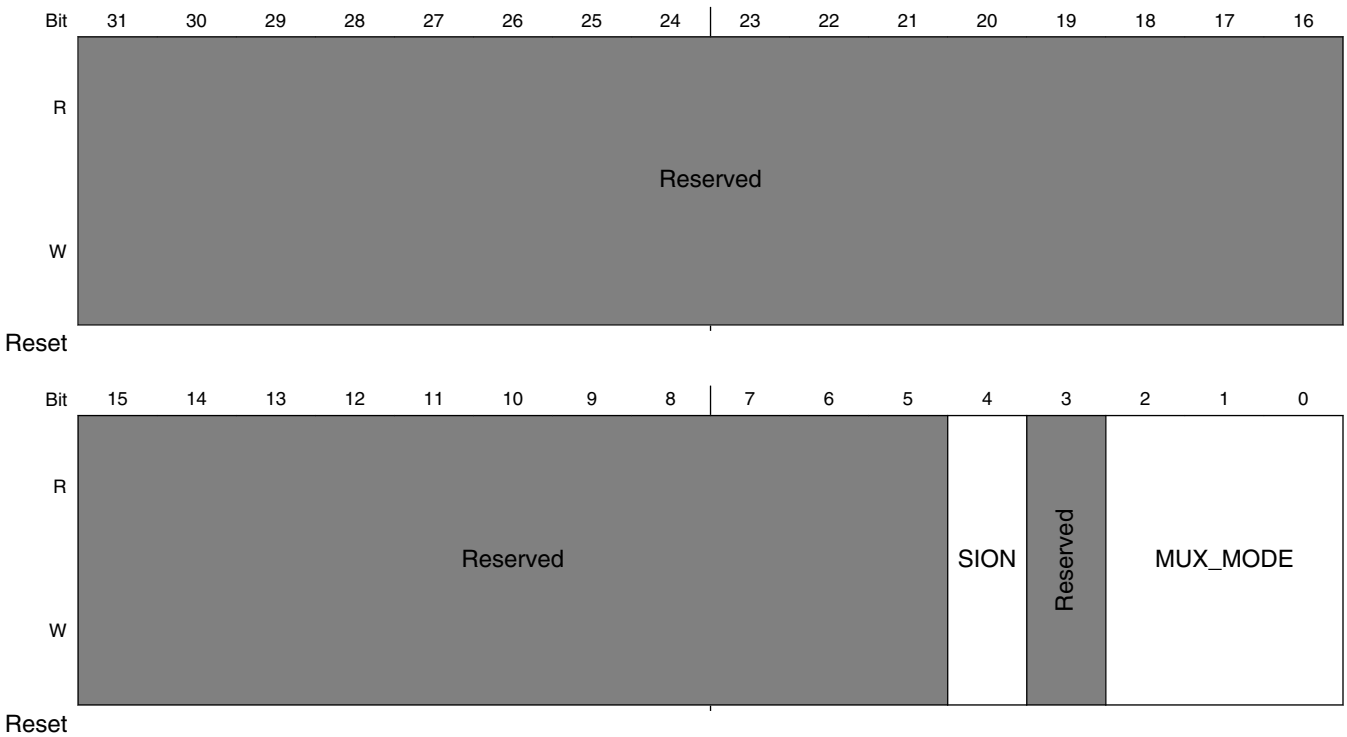
IOMUXC_SW_MUX_CTL_PAD_NAND_DQS field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_DQS 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: NAND_DQS. 000 **ALT0_NAND_DQS** — Select mux mode: ALT0 mux port: NAND_DQS of instance: nand 001 **ALT1_FLEXSPI_A_DQS** — Select mux mode: ALT1 mux port: FLEXSPI_A_DQS of instance: flexspi 010 **ALT2_AUDIOMIX_SAI3_MCLK** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI3_MCLK of instance: sai3 011 **ALT3_ISP_SHUTTER_OPEN_0** — Select mux mode: ALT3 mux port: ISP_SHUTTER_OPEN_0 of instance: isp 100 **ALT4_I2C3_SCL** — Select mux mode: ALT4 mux port: I2C3_SCL of instance: i2c3 101 **ALT5_GPIO3_IO[14]** — Select mux mode: ALT5 mux port: GPIO3_IO14 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[12]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE12 of instance: coresight

8.2.4.67 SW_MUX_CTL_PAD_NAND_RE_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_RE_B)

SW_MUX_CTL Register

Address: 3033_0000h base + 11Ch offset = 3033_011Ch



IOMUXC_SW_MUX_CTL_PAD_NAND_RE_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_RE_B 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: NAND_RE_B. 000 **ALT0_NAND_RE_B** — Select mux mode: ALT0 mux port: NAND_RE_B of instance: nand 001 **ALT1_FLEXSPI_B_DQS** — Select mux mode: ALT1 mux port: FLEXSPI_B_DQS of instance: flexspi

Table continues on the next page...

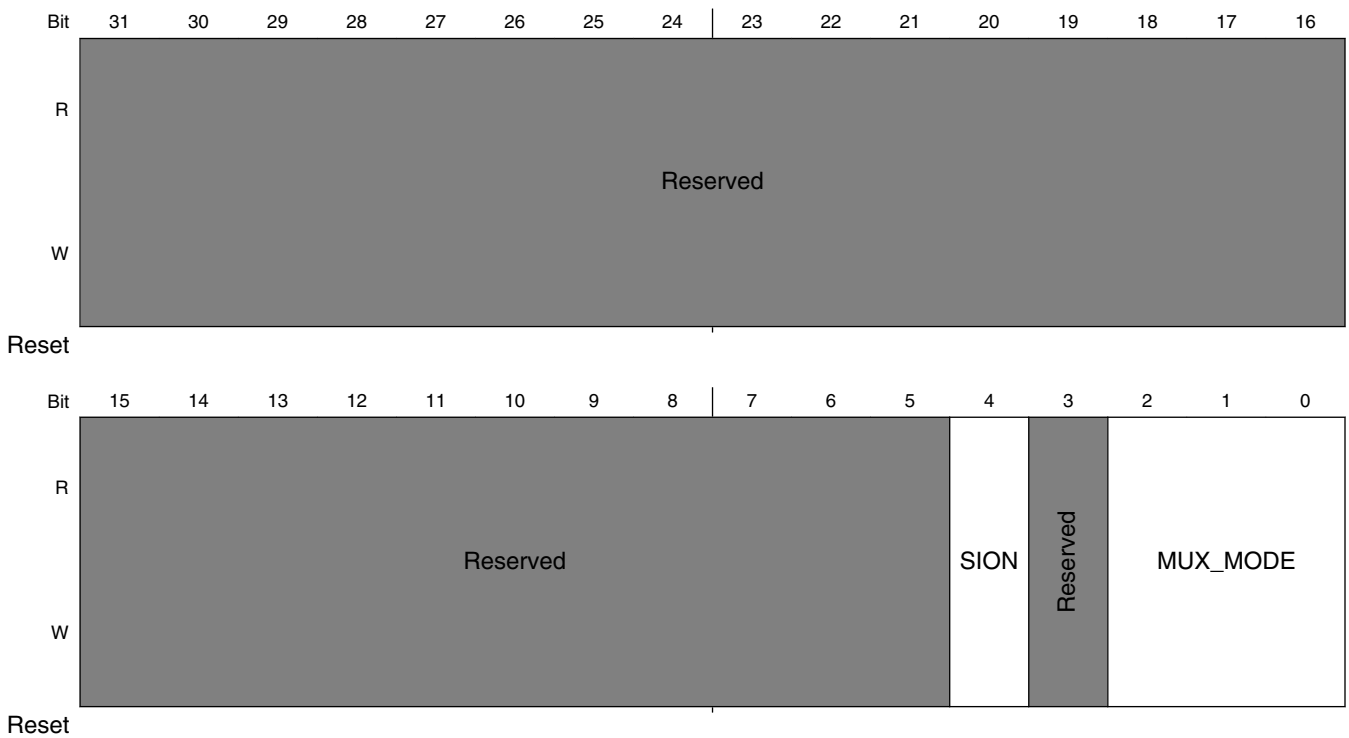
IOMUXC_SW_MUX_CTL_PAD_NAND_RE_B field descriptions (continued)

Field	Description
	010 **ALT2_USDHC3_DATA4** — Select mux mode: ALT2 mux port: USDHC3_DATA4 of instance: usdhc3 100 **ALT4_UART4_TX** — Select mux mode: ALT4 mux port: UART4_TX of instance: uart4 101 **ALT5_GPIO3_IO[15]** — Select mux mode: ALT5 mux port: GPIO3_IO15 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[13]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE13 of instance: coresight

8.2.4.68 SW_MUX_CTL_PAD_NAND_READY_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_READY_B)

SW_MUX_CTL Register

Address: 3033_0000h base + 120h offset = 3033_0120h



IOMUXC_SW_MUX_CTL_PAD_NAND_READY_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality.

Table continues on the next page...

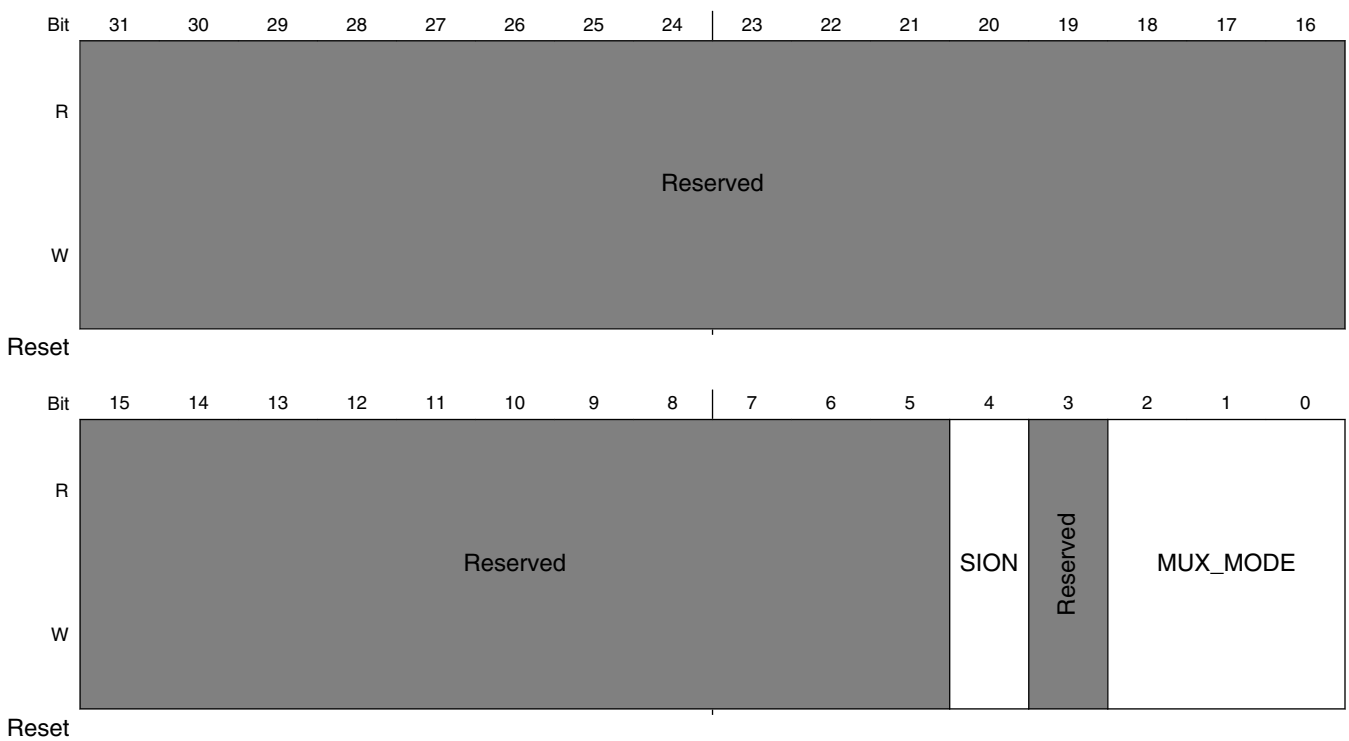
IOMUXC_SW_MUX_CTL_PAD_NAND_READY_B field descriptions (continued)

Field	Description
	1 **ENABLED** — Force input path of pad NAND_READY_B 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: NAND_READY_B. 000 **ALT0_NAND_READY_B** — Select mux mode: ALT0 mux port: NAND_READY_B of instance: nand 010 **ALT2_USDHC3_RESET_B** — Select mux mode: ALT2 mux port: USDHC3_RESET_B of instance: usdhc3 100 **ALT4_I2C3_SCL** — Select mux mode: ALT4 mux port: I2C3_SCL of instance: i2c3 101 **ALT5_GPIO3_IO[16]** — Select mux mode: ALT5 mux port: GPIO3_IO16 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[14]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE14 of instance: coresight

8.2.4.69 SW_MUX_CTL_PAD_NAND_WE_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_WE_B)

SW_MUX_CTL Register

Address: 3033_0000h base + 124h offset = 3033_0124h



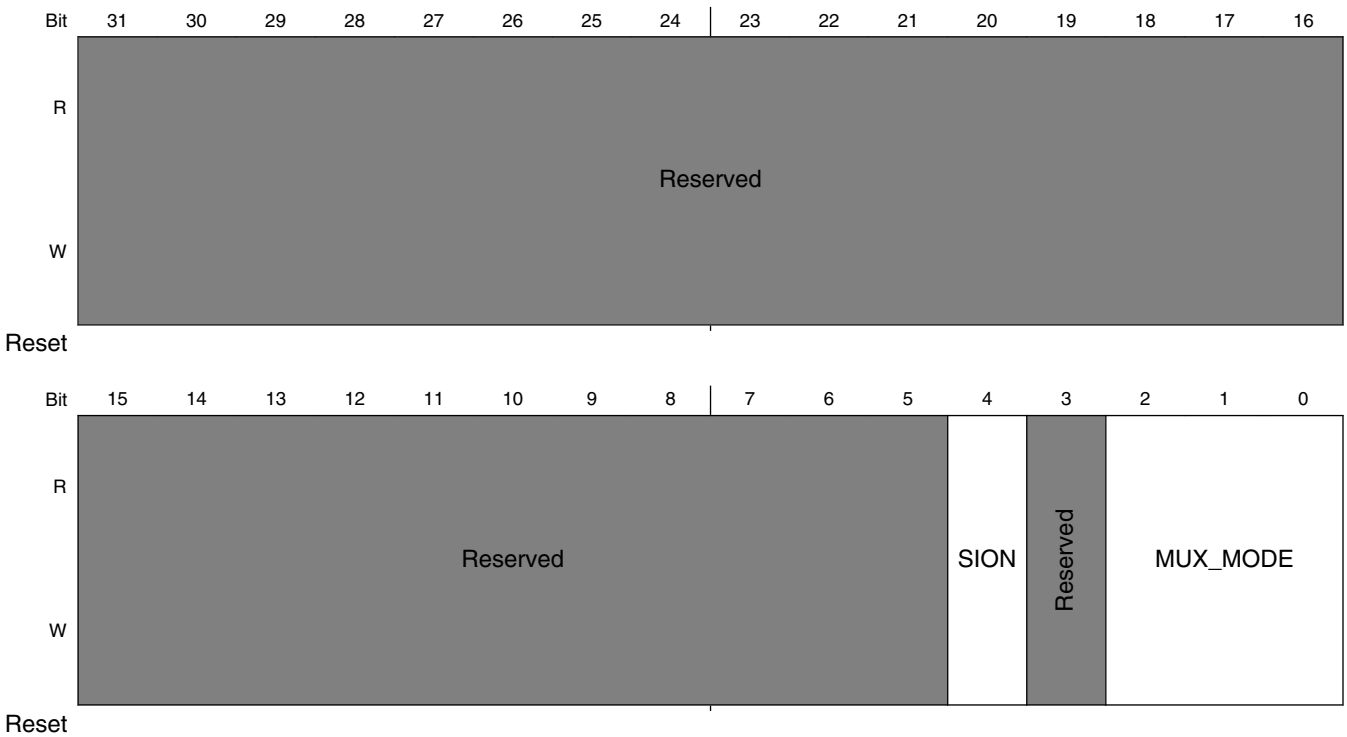
IOMUXC_SW_MUX_CTL_PAD_NAND_WE_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_WE_B 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: NAND_WE_B. 000 **ALT0_NAND_WE_B** — Select mux mode: ALT0 mux port: NAND_WE_B of instance: nand 010 **ALT2_USDHC3_CLK** — Select mux mode: ALT2 mux port: USDHC3_CLK of instance: usdhc3 100 **ALT4_I2C3_SDA** — Select mux mode: ALT4 mux port: I2C3_SDA of instance: i2c3 101 **ALT5_GPIO3_IO[17]** — Select mux mode: ALT5 mux port: GPIO3_IO17 of instance: gpio3 110 **ALT6_CORESIGHT_TRACE[15]** — Select mux mode: ALT6 mux port: CORESIGHT_TRACE15 of instance: coresight

8.2.4.70 SW_MUX_CTL_PAD_NAND_WP_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B)

SW_MUX_CTL Register

Address: 3033_0000h base + 128h offset = 3033_0128h



IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad NAND_WP_B 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: NAND_WP_B. 000 **ALT0_NAND_WP_B** — Select mux mode: ALT0 mux port: NAND_WP_B of instance: nand 010 **ALT2_USDHC3_CMD** — Select mux mode: ALT2 mux port: USDHC3_CMD of instance: usdhc3 100 **ALT4_I2C4_SCL** — Select mux mode: ALT4 mux port: I2C4_SCL of instance: i2c4

Table continues on the next page...

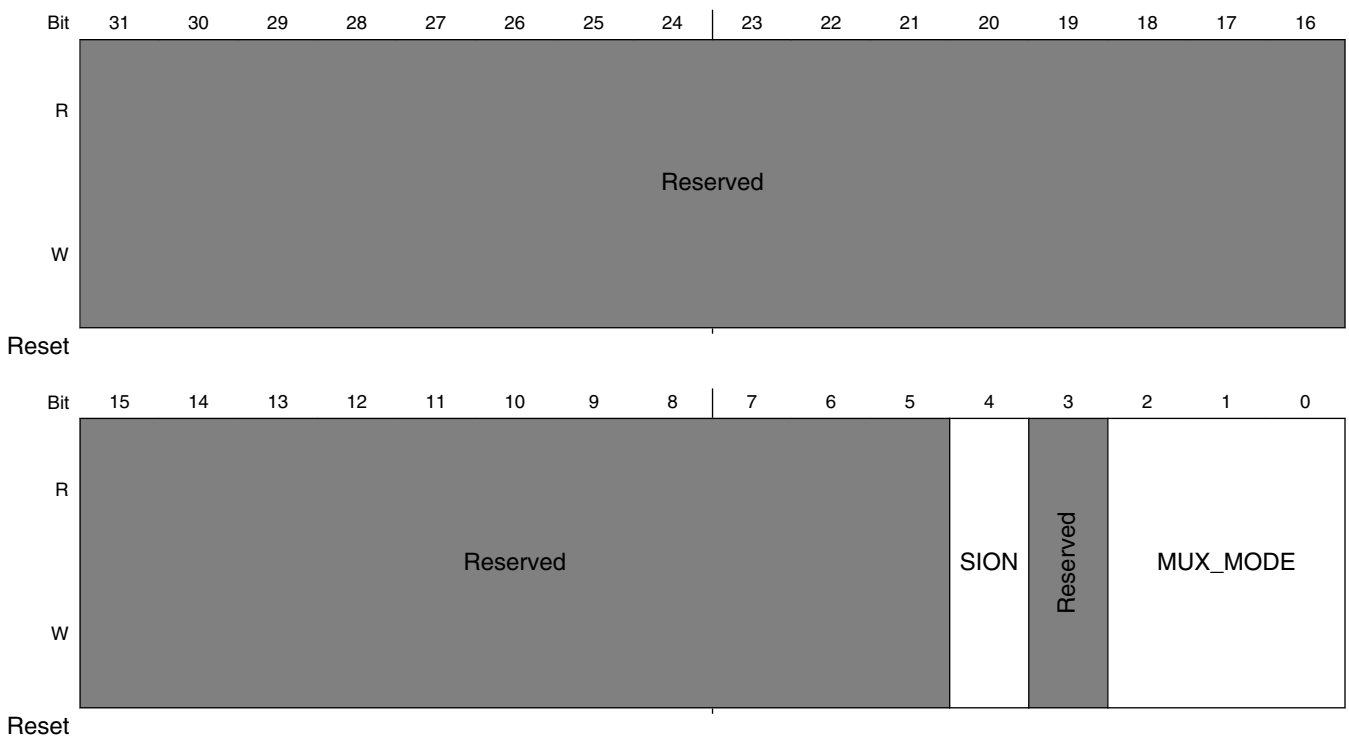
IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B field descriptions (continued)

Field	Description
	101 **ALT5_GPIO3_IO[18]** — Select mux mode: ALT5 mux port: GPIO3_IO18 of instance: gpio3 110 **ALT6_CORESIGHT_EVENTO** — Select mux mode: ALT6 mux port: CORESIGHT_EVENTO of instance: coresight

8.2.4.71 SW_MUX_CTL_PAD_SAI5_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 12Ch offset = 3033_012Ch



IOMUXC_SW_MUX_CTL_PAD_SAI5_RXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI5_RXFS 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

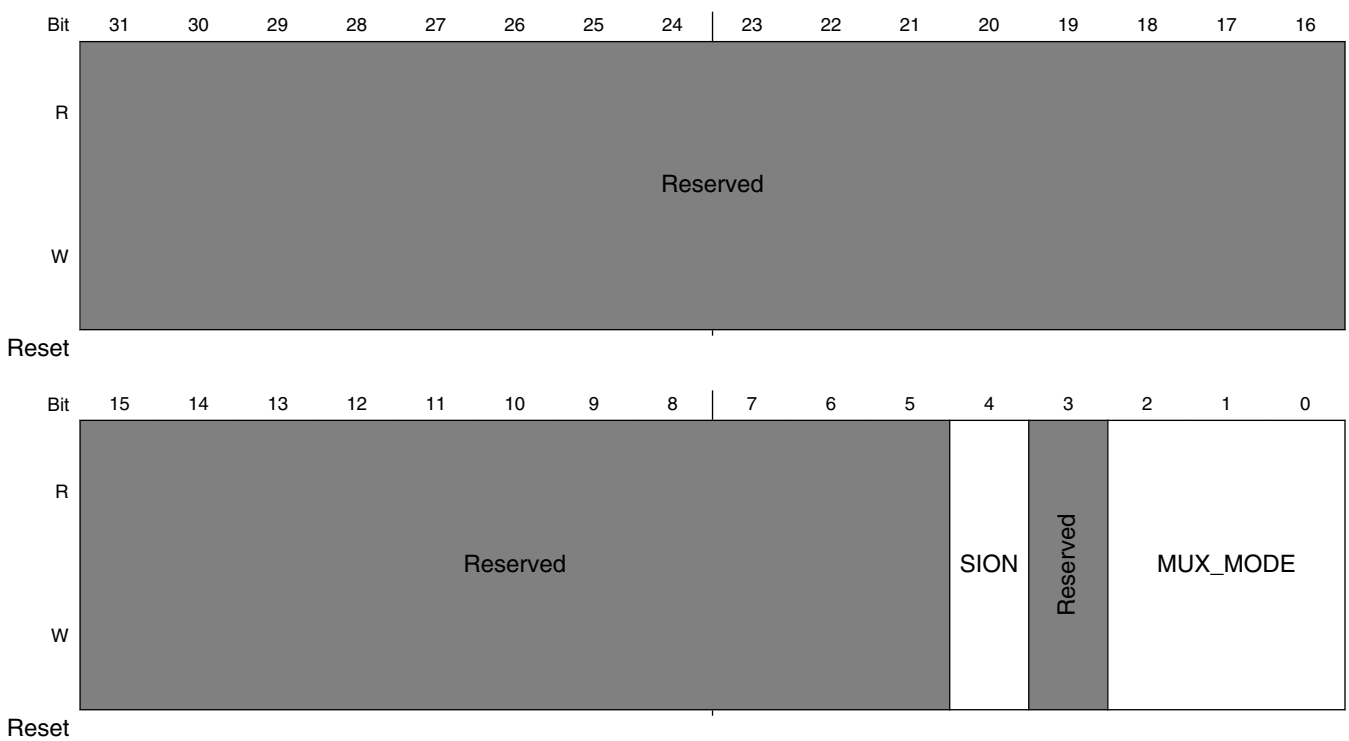
IOMUXC_SW_MUX_CTL_PAD_SAI5_RXFS field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SAI5_RXFS. 000 **ALT0_AUDIOMIX_SAI5_RX_SYNC** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI5_RX_SYNC of instance: sai5 001 **ALT1_AUDIOMIX_SAI1_TX_DATA[0]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI1_TX_DATA00 of instance: sai1 010 **ALT2_PWM4_OUT** — Select mux mode: ALT2 mux port: PWM4_OUT of instance: pwm4 011 **ALT3_I2C6_SCL** — Select mux mode: ALT3 mux port: I2C6_SCL of instance: i2c6 101 **ALT5_GPIO3_IO[19]** — Select mux mode: ALT5 mux port: GPIO3_IO19 of instance: gpio3

8.2.4.72 SW_MUX_CTL_PAD_SAI5_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 130h offset = 3033_0130h



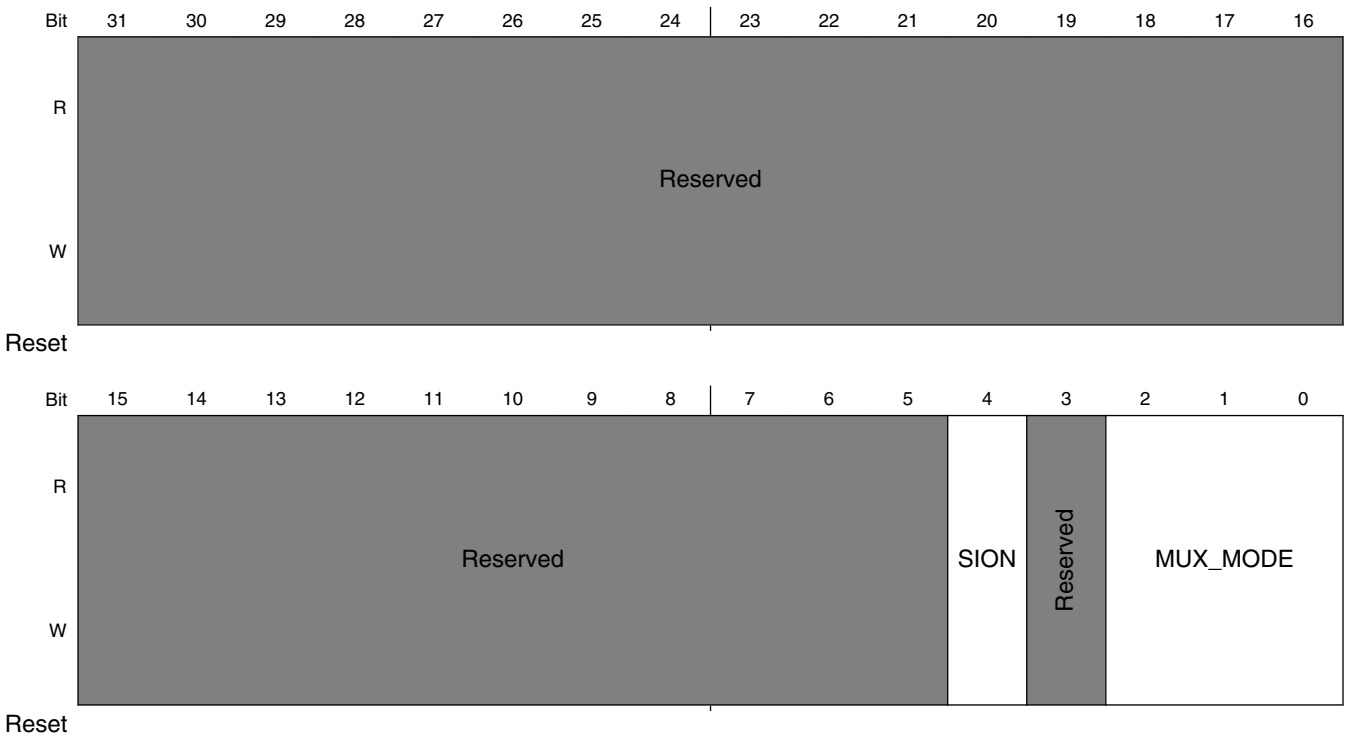
IOMUXC_SW_MUX_CTL_PAD_SAI5_RXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI5_RXC 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SAI5_RXC. 000 **ALT0_AUDIOMIX_SAI5_RX_BCLK** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI5_RX_BCLK of instance: sai5 001 **ALT1_AUDIOMIX_SAI1_TX_DATA[1]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI1_TX_DATA01 of instance: sai1 010 **ALT2_PWM3_OUT** — Select mux mode: ALT2 mux port: PWM3_OUT of instance: pwm3 011 **ALT3_I2C6_SDA** — Select mux mode: ALT3 mux port: I2C6_SDA of instance: i2c6 100 **ALT4_AUDIOMIX_PDM_CLK** — Select mux mode: ALT4 mux port: AUDIOMIX_PDM_CLK of instance: pdm 101 **ALT5_GPIO3_IO[20]** — Select mux mode: ALT5 mux port: GPIO3_IO20 of instance: gpio3

8.2.4.73 SW_MUX_CTL_PAD_SAI5_RXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 134h offset = 3033_0134h



IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI5_RXD0 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SAI5_RXD0. 000 **ALT0_AUDIOMIX_SAI5_RX_DATA[0]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI5_RX_DATA00 of instance: sai5 001 **ALT1_AUDIOMIX_SAI1_TX_DATA[2]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI1_TX_DATA02 of instance: sai1

Table continues on the next page...

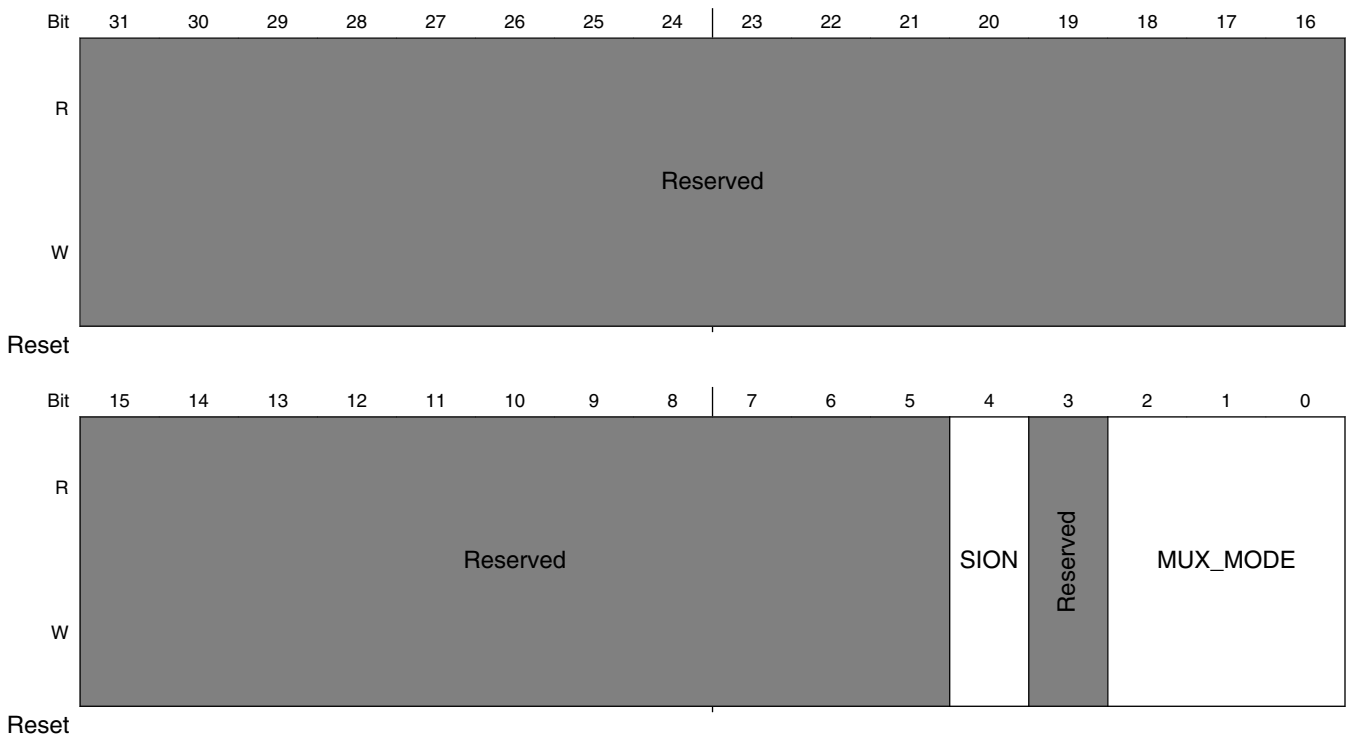
IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD0 field descriptions (continued)

Field	Description
	010 **ALT2_PWM2_OUT** — Select mux mode: ALT2 mux port: PWM2_OUT of instance: pwm2 011 **ALT3_I2C5_SCL** — Select mux mode: ALT3 mux port: I2C5_SCL of instance: i2c5 100 **ALT4_AUDIOMIX_PDM_BIT_STREAM[0]** — Select mux mode: ALT4 mux port: AUDIOMIX_PDM_BIT_STREAM00 of instance: pdm 101 **ALT5_GPIO3_IO[21]** — Select mux mode: ALT5 mux port: GPIO3_IO21 of instance: gpio3

8.2.4.74 SW_MUX_CTL_PAD_SAI5_RXD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD1)

SW_MUX_CTL Register

Address: 3033_0000h base + 138h offset = 3033_0138h



IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality.

Table continues on the next page...

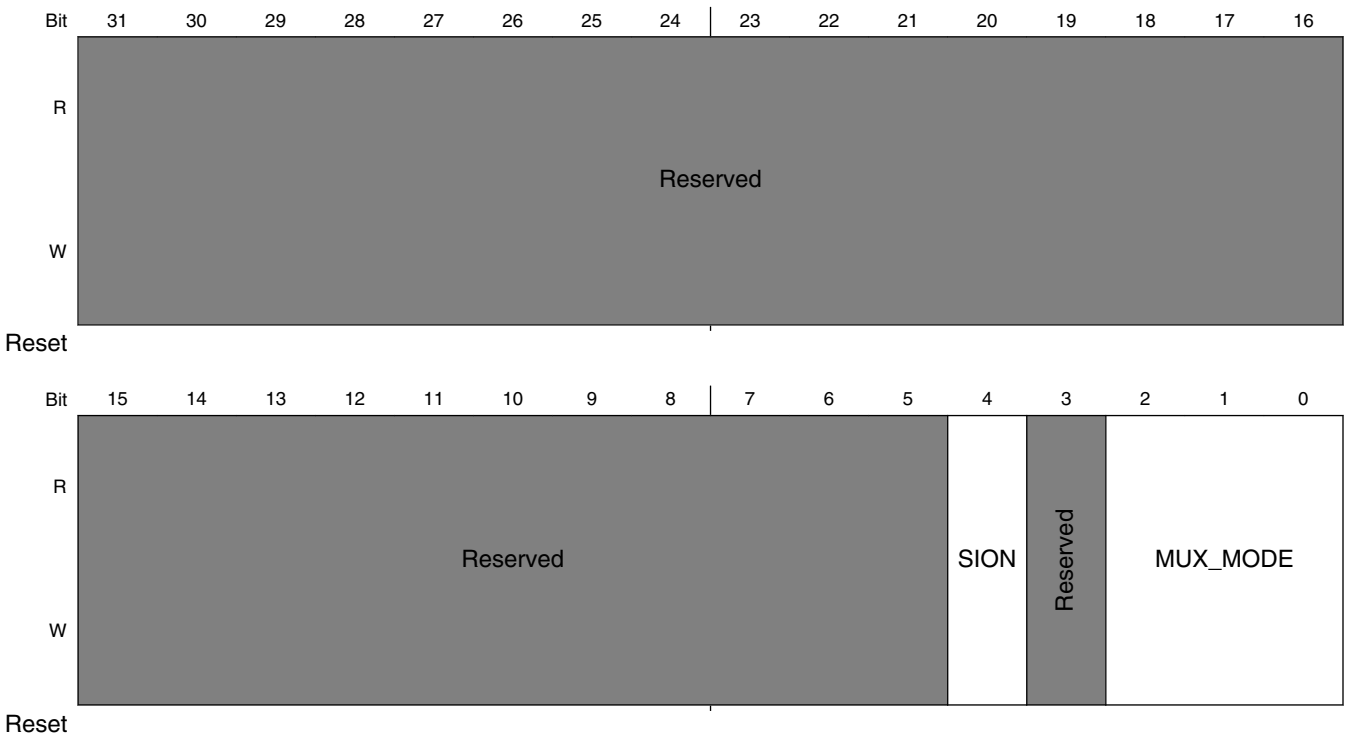
IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD1 field descriptions (continued)

Field	Description
	1 **ENABLED** — Force input path of pad SAI5_RXD1 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: SAI5_RXD1. 000 **ALT0_AUDIOMIX_SAI5_RX_DATA[1]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI5_RX_DATA01 of instance: sai5 001 **ALT1_AUDIOMIX_SAI1_TX_DATA[3]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI1_TX_DATA03 of instance: sai1 010 **ALT2_AUDIOMIX_SAI1_TX_SYNC** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI1_TX_SYNC of instance: sai1 011 **ALT3_AUDIOMIX_SAI5_TX_SYNC** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI5_TX_SYNC of instance: sai5 100 **ALT4_AUDIOMIX_PDM_BIT_STREAM[1]** — Select mux mode: ALT4 mux port: AUDIOMIX_PDM_BIT_STREAM01 of instance: pdm 101 **ALT5_GPIO3_IO[22]** — Select mux mode: ALT5 mux port: GPIO3_IO22 of instance: gpio3 110 **ALT6_CAN1_TX** — Select mux mode: ALT6 mux port: CAN1_TX of instance: can1

8.2.4.75 SW_MUX_CTL_PAD_SAI5_RXD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD2)

SW_MUX_CTL Register

Address: 3033_0000h base + 13Ch offset = 3033_013Ch



IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI5_RXD2 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: SAI5_RXD2. 000 **ALT0_AUDIOMIX_SAI5_RX_DATA[2]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI5_RX_DATA02 of instance: sai5 001 **ALT1_AUDIOMIX_SAI1_TX_DATA[4]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI1_TX_DATA04 of instance: sai1

Table continues on the next page...

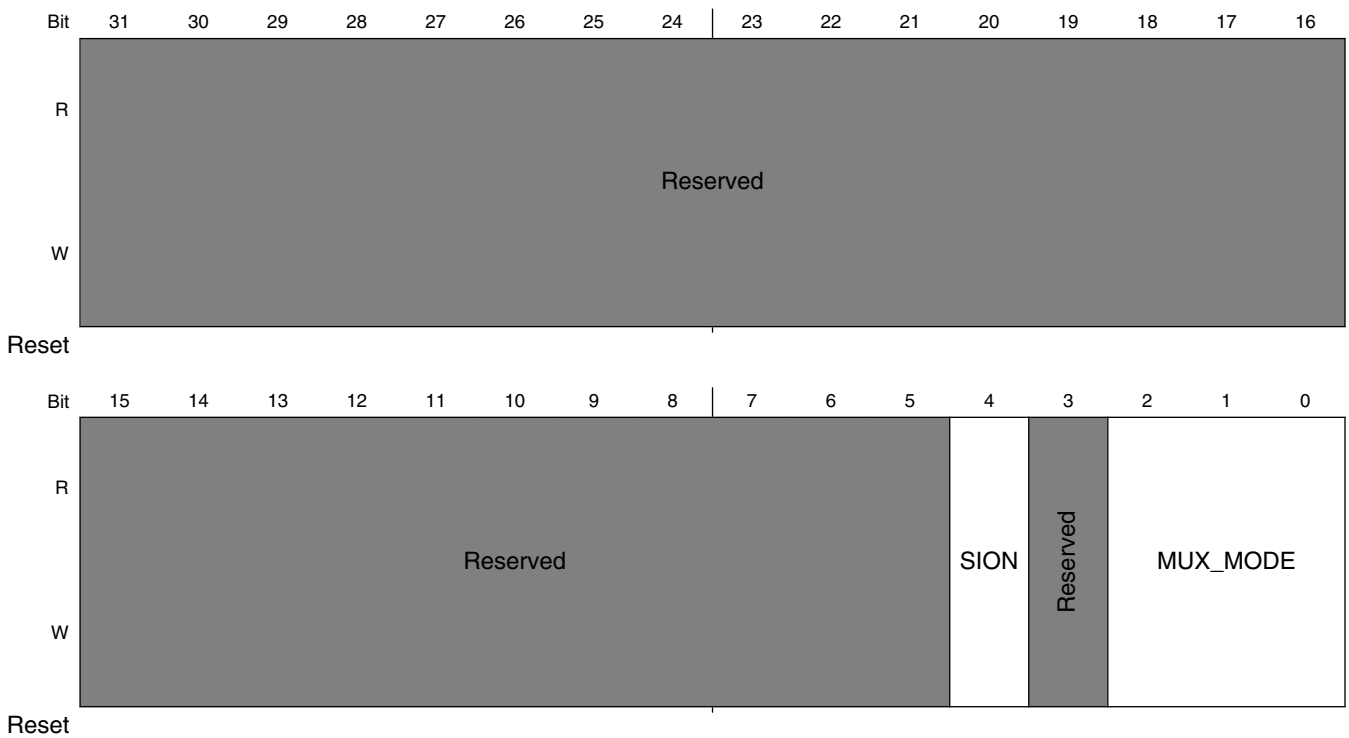
IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD2 field descriptions (continued)

Field	Description
	010 **ALT2_AUDIOMIX_SAI1_TX_SYNC** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI1_TX_SYNC of instance: sai1 011 **ALT3_AUDIOMIX_SAI5_TX_BCLK** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI5_TX_BCLK of instance: sai5 100 **ALT4_AUDIOMIX_PDM_BIT_STREAM[2]** — Select mux mode: ALT4 mux port: AUDIOMIX_PDM_BIT_STREAM02 of instance: pdm 101 **ALT5_GPIO3_IO[23]** — Select mux mode: ALT5 mux port: GPIO3_IO23 of instance: gpio3 110 **ALT6_CAN1_RX** — Select mux mode: ALT6 mux port: CAN1_RX of instance: can1

8.2.4.76 SW_MUX_CTL_PAD_SAI5_RXD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD3)

SW_MUX_CTL Register

Address: 3033_0000h base + 140h offset = 3033_0140h



IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field.

Table continues on the next page...

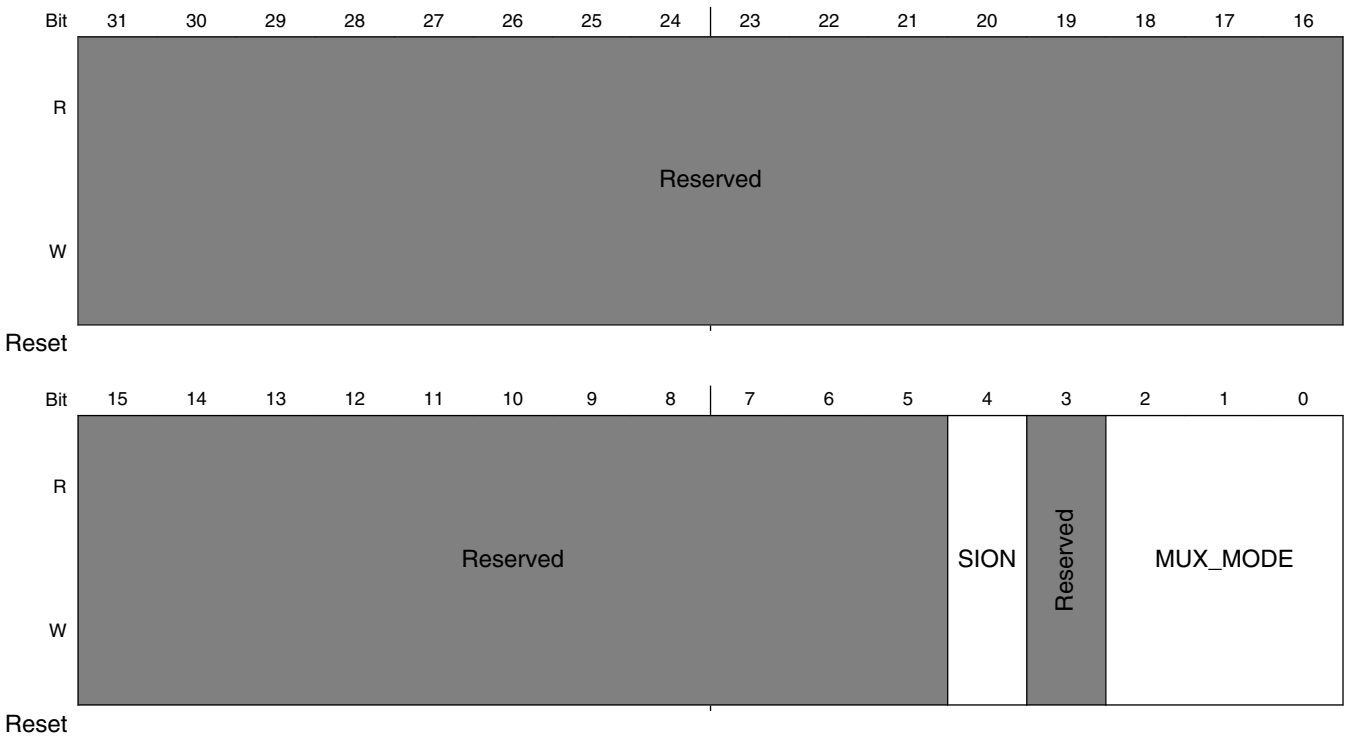
IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD3 field descriptions (continued)

Field	Description
	Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI5_RXD3 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: SAI5_RXD3. 000 **ALT0_AUDIOMIX_SAI5_RX_DATA[3]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI5_RX_DATA03 of instance: sai5 001 **ALT1_AUDIOMIX_SAI1_TX_DATA[5]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI1_TX_DATA05 of instance: sai1 010 **ALT2_AUDIOMIX_SAI1_TX_SYNC** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI1_TX_SYNC of instance: sai1 011 **ALT3_AUDIOMIX_SAI5_TX_DATA[0]** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI5_TX_DATA00 of instance: sai5 100 **ALT4_AUDIOMIX_PDM_BIT_STREAM[3]** — Select mux mode: ALT4 mux port: AUDIOMIX_PDM_BIT_STREAM03 of instance: pdm 101 **ALT5_GPIO3_IO[24]** — Select mux mode: ALT5 mux port: GPIO3_IO24 of instance: gpio3 110 **ALT6_CAN2_TX** — Select mux mode: ALT6 mux port: CAN2_TX of instance: can2

8.2.4.77 SW_MUX_CTL_PAD_SAI5_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_MCLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 144h offset = 3033_0144h



IOMUXC_SW_MUX_CTL_PAD_SAI5_MCLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI5_MCLK 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SAI5_MCLK. 000 **ALT0_AUDIOMIX_SAI5_MCLK** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI5_MCLK of instance: sai5 001 **ALT1_AUDIOMIX_SAI1_TX_BCLK** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI1_TX_BCLK of instance: sai1

Table continues on the next page...

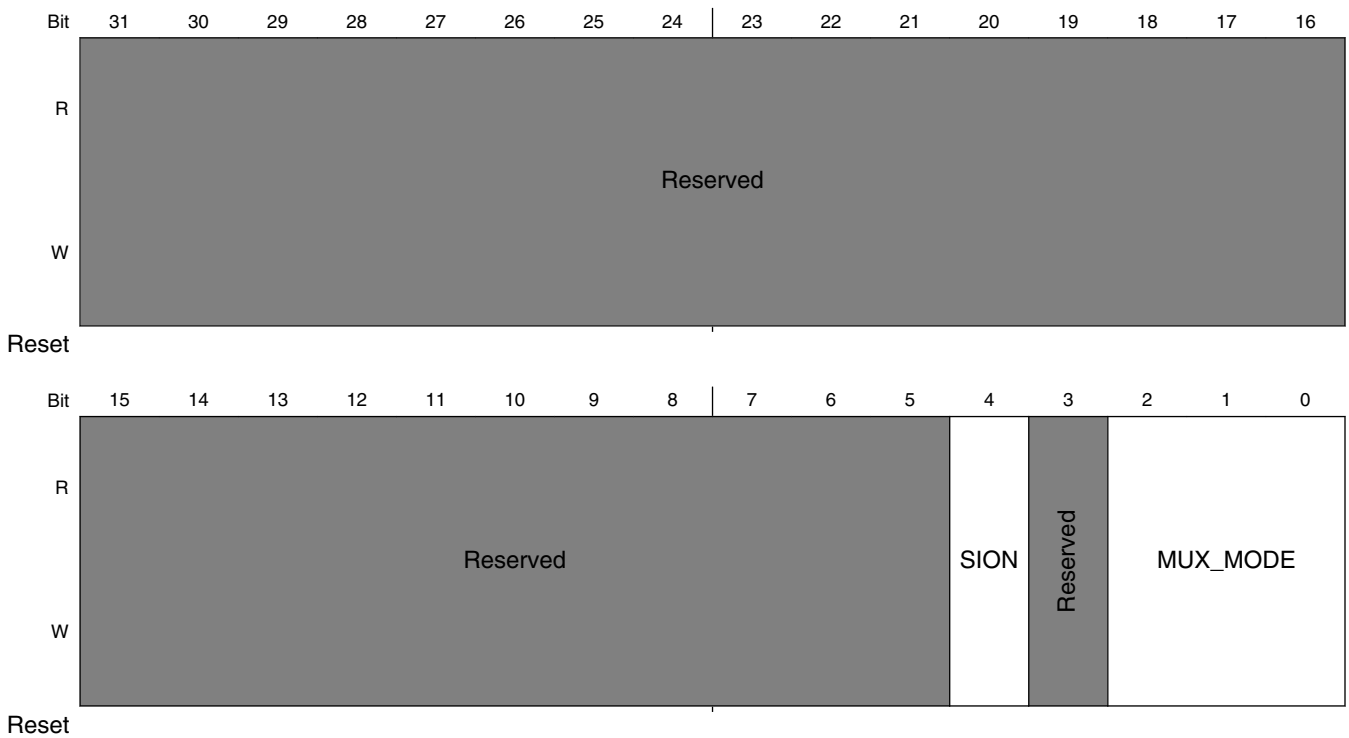
IOMUXC_SW_MUX_CTL_PAD_SAI5_MCLK field descriptions (continued)

Field	Description
	010 **ALT2_PWM1_OUT** — Select mux mode: ALT2 mux port: PWM1_OUT of instance: pwm1 011 **ALT3_I2C5_SDA** — Select mux mode: ALT3 mux port: I2C5_SDA of instance: i2c5 101 **ALT5_GPIO3_IO[25]** — Select mux mode: ALT5 mux port: GPIO3_IO25 of instance: gpio3 110 **ALT6_CAN2_RX** — Select mux mode: ALT6 mux port: CAN2_RX of instance: can2

8.2.4.78 SW_MUX_CTL_PAD_SAI1_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 148h offset = 3033_0148h



IOMUXC_SW_MUX_CTL_PAD_SAI1_RXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_RXFS 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

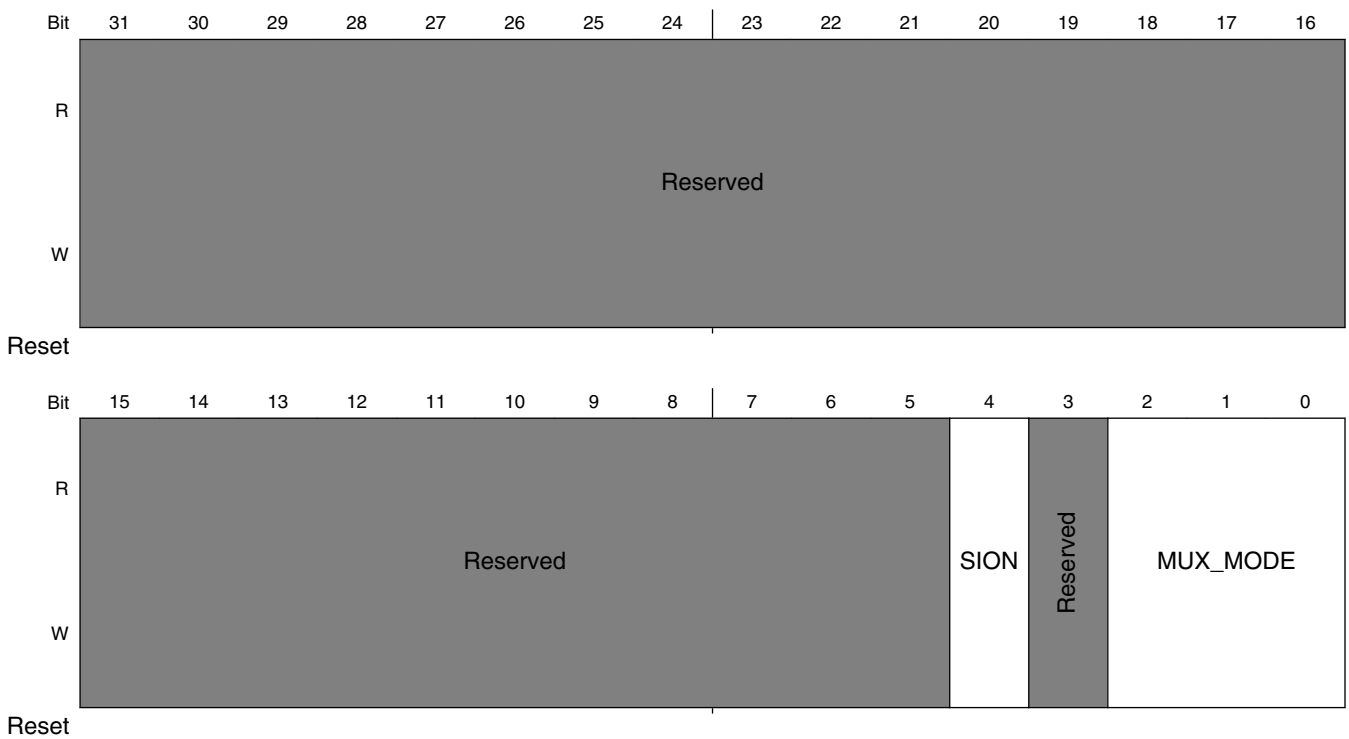
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXFS field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SAI1_RXFS. 000 **ALT0_AUDIOMIX_SAI1_RX_SYNC** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_RX_SYNC of instance: sai1 100 **ALT4_ENET1_1588_EVENT0_IN** — Select mux mode: ALT4 mux port: ENET1_1588_EVENT0_IN of instance: enet1 101 **ALT5_GPIO4_IO[0]** — Select mux mode: ALT5 mux port: GPIO4_IO00 of instance: gpio4

8.2.4.79 SW_MUX_CTL_PAD_SAI1_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 14Ch offset = 3033_014Ch



IOMUXC_SW_MUX_CTL_PAD_SAI1_RXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved

Table continues on the next page...

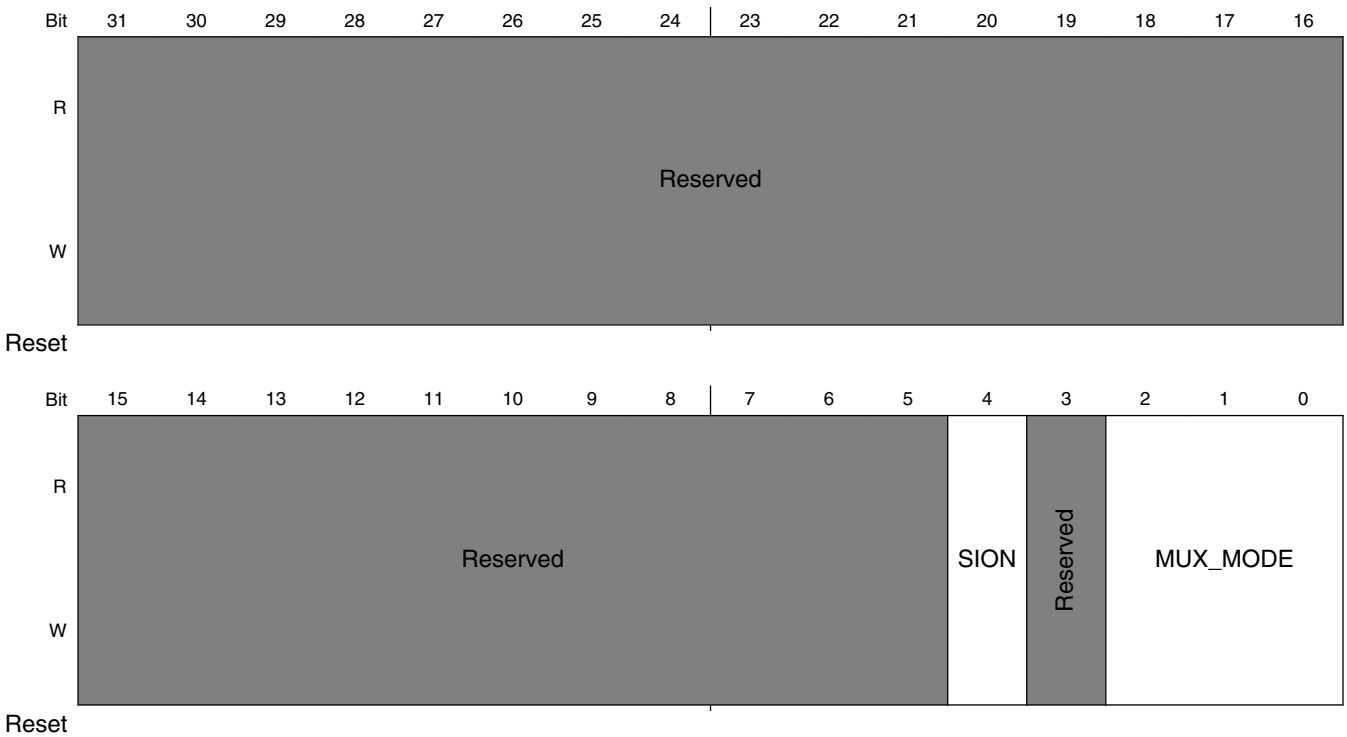
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXC field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_RXC 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SAI1_RXC. 000 **ALT0_AUDIOMIX_SAI1_RX_BCLK** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_RX_BCLK of instance: sai1 011 **ALT3_AUDIOMIX_PDM_CLK** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_CLK of instance: pdm 100 **ALT4_ENET1_1588_EVENT0_OUT** — Select mux mode: ALT4 mux port: ENET1_1588_EVENT0_OUT of instance: enet1 101 **ALT5_GPIO4_IO[1]** — Select mux mode: ALT5 mux port: GPIO4_IO01 of instance: gpio4

8.2.4.80 SW_MUX_CTL_PAD_SAI1_RXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 150h offset = 3033_0150h



IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_RXD0 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SAI1_RXD0. 000 **ALT0_AUDIOMIX_SAI1_RX_DATA[0]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_RX_DATA00 of instance: sai1 010 **ALT2_AUDIOMIX_SAI1_TX_DATA[1]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI1_TX_DATA01 of instance: sai1

Table continues on the next page...

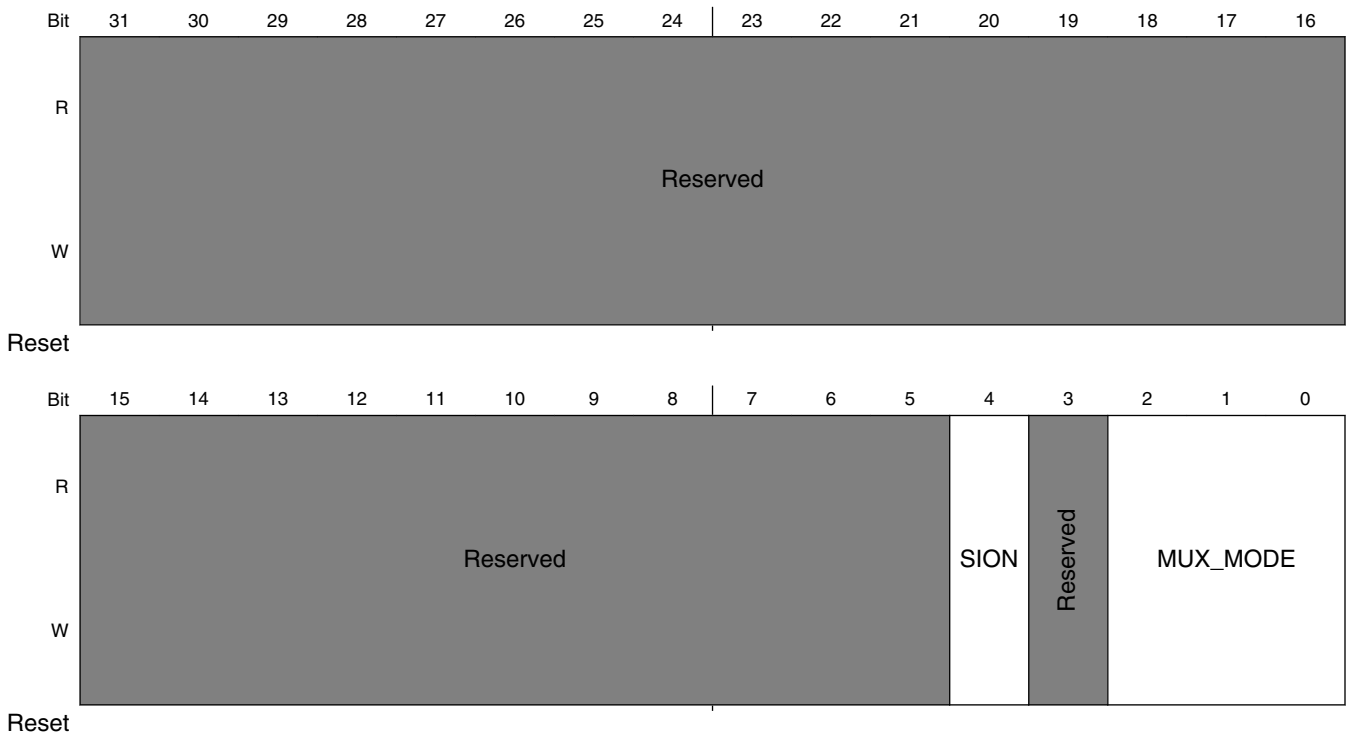
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD0 field descriptions (continued)

Field	Description
	011 **ALT3_AUDIOMIX_PDM_BIT_STREAM[0]** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_BIT_STREAM00 of instance: pdm 100 **ALT4_ENET1_1588_EVENT1_IN** — Select mux mode: ALT4 mux port: ENET1_1588_EVENT1_IN of instance: enet1 101 **ALT5_GPIO4_IO[2]** — Select mux mode: ALT5 mux port: GPIO4_IO02 of instance: gpio4

8.2.4.81 SW_MUX_CTL_PAD_SAI1_RXD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD1)

SW_MUX_CTL Register

Address: 3033_0000h base + 154h offset = 3033_0154h



IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_RXD1 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

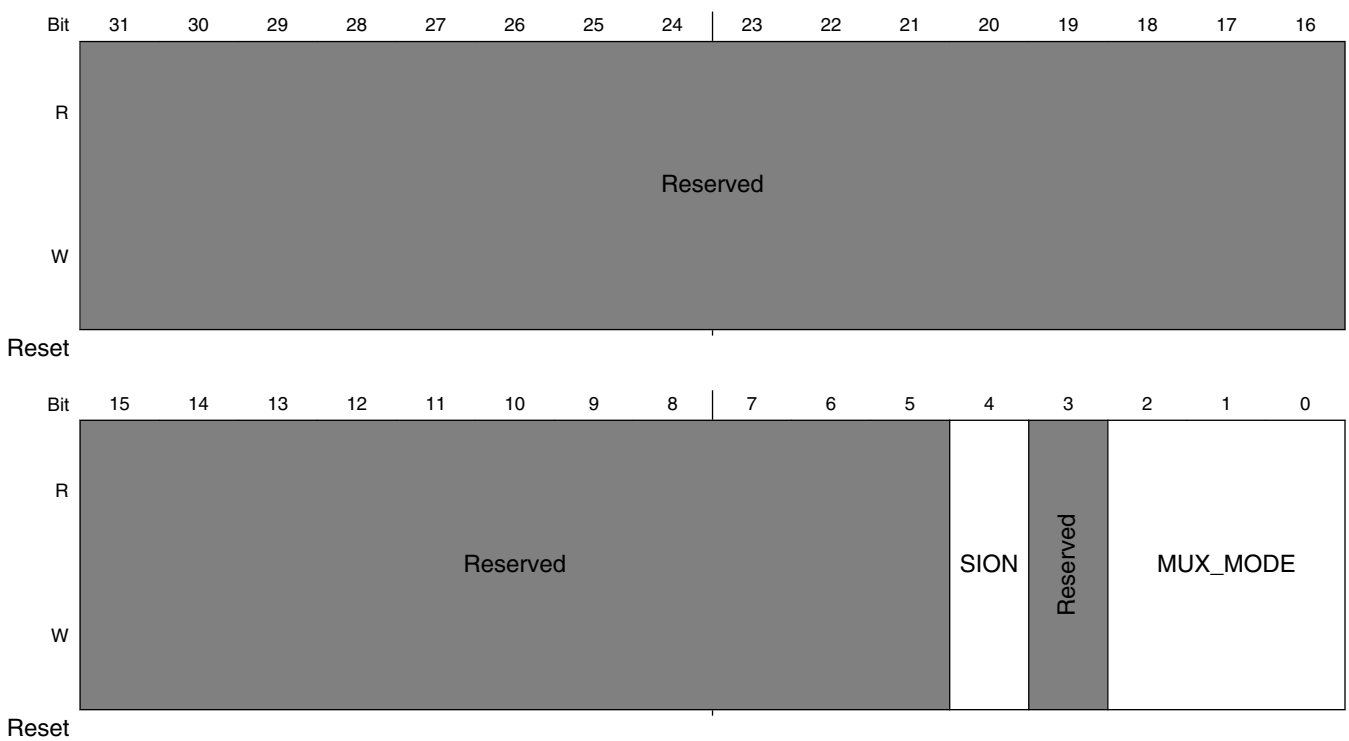
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD1 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SAI1_RXD1. 000 **ALT0_AUDIOMIX_SAI1_RX_DATA[1]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_RX_DATA01 of instance: sai1 011 **ALT3_AUDIOMIX_PDM_BIT_STREAM[1]** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_BIT_STREAM01 of instance: pdm 100 **ALT4_ENET1_1588_EVENT1_OUT** — Select mux mode: ALT4 mux port: ENET1_1588_EVENT1_OUT of instance: enet1 101 **ALT5_GPIO4_IO[3]** — Select mux mode: ALT5 mux port: GPIO4_IO03 of instance: gpio4

8.2.4.82 SW_MUX_CTL_PAD_SAI1_RXD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD2)

SW_MUX_CTL Register

Address: 3033_0000h base + 158h offset = 3033_0158h



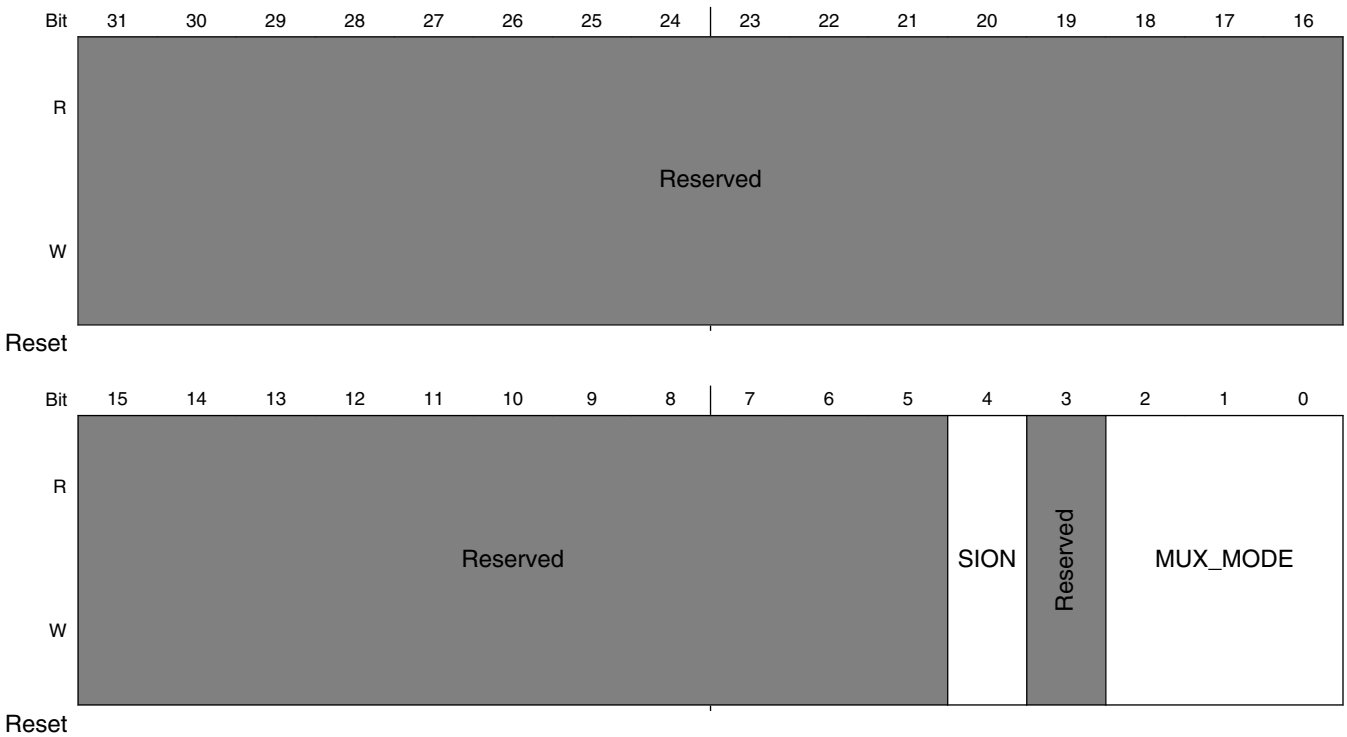
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_RXD2 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SAI1_RXD2. 000 **ALT0_AUDIOMIX_SAI1_RX_DATA[2]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_RX_DATA02 of instance: sai1 011 **ALT3_AUDIOMIX_PDM_BIT_STREAM[2]** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_BIT_STREAM02 of instance: pdm 100 **ALT4_ENET1_MDC** — Select mux mode: ALT4 mux port: ENET1_MDC of instance: enet1 101 **ALT5_GPIO4_IO[4]** — Select mux mode: ALT5 mux port: GPIO4_IO04 of instance: gpio4

8.2.4.83 SW_MUX_CTL_PAD_SAI1_RXD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD3)

SW_MUX_CTL Register

Address: 3033_0000h base + 15Ch offset = 3033_015Ch



IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_RXD3 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SAI1_RXD3. 000 **ALT0_AUDIOMIX_SAI1_RX_DATA[3]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_RX_DATA03 of instance: sai1 011 **ALT3_AUDIOMIX_PDM_BIT_STREAM[3]** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_BIT_STREAM03 of instance: pdm

Table continues on the next page...

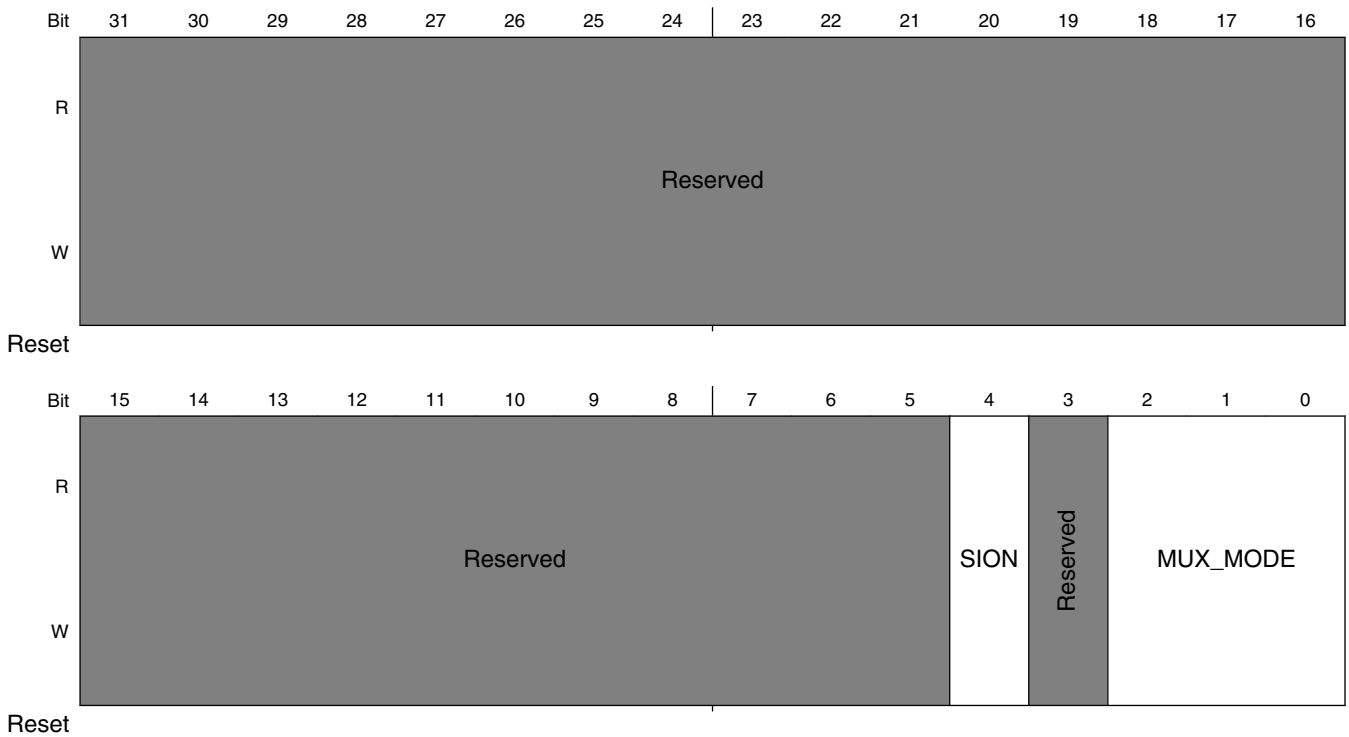
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD3 field descriptions (continued)

Field	Description
	100 **ALT4_ENET1_MDIO** — Select mux mode: ALT4 mux port: ENET1_MDIO of instance: enet1 101 **ALT5_GPIO4_IO[5]** — Select mux mode: ALT5 mux port: GPIO4_IO05 of instance: gpio4

8.2.4.84 SW_MUX_CTL_PAD_SAI1_RXD4 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD4)

SW_MUX_CTL Register

Address: 3033_0000h base + 160h offset = 3033_0160h



IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD4 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_RXD4 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved

Table continues on the next page...

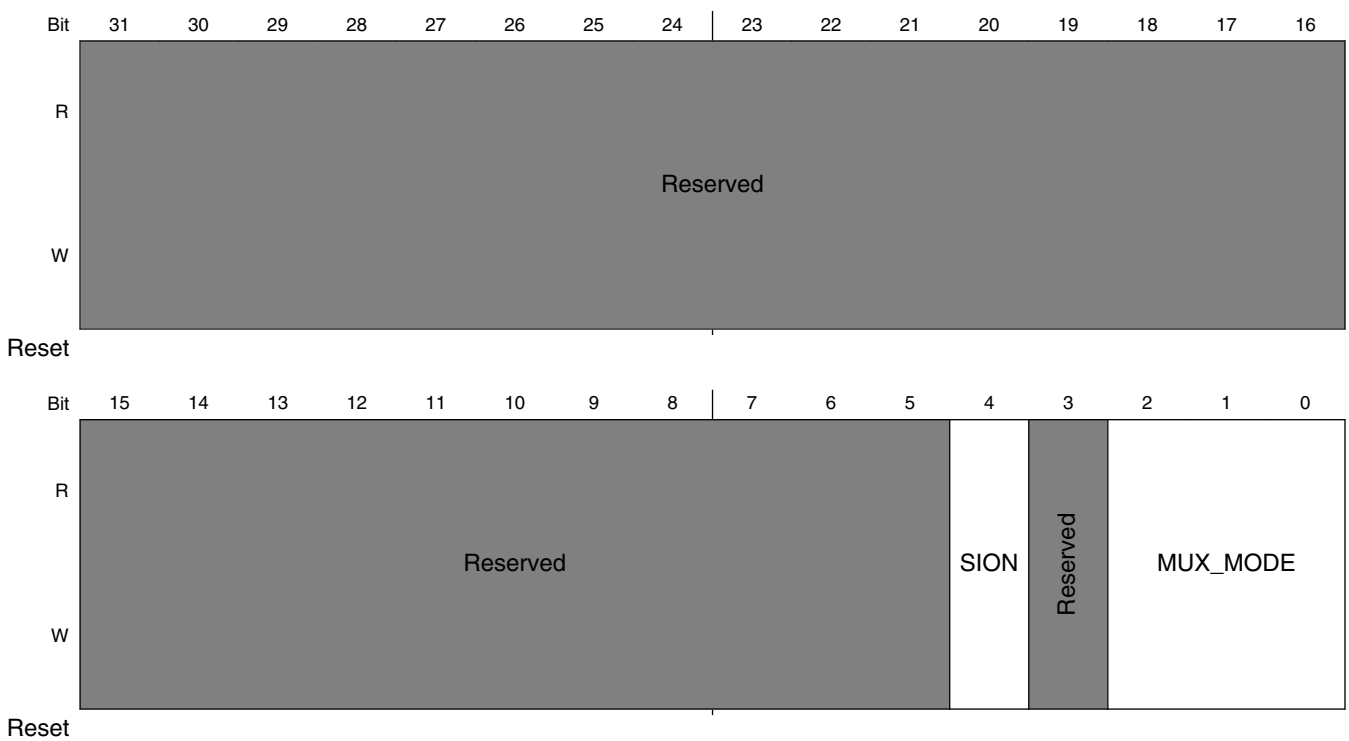
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD4 field descriptions (continued)

Field	Description
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SAI1_RXD4. 000 **ALT0_AUDIOMIX_SAI1_RX_DATA[4]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_RX_DATA04 of instance: sai1 001 **ALT1_AUDIOMIX_SAI6_TX_BCLK** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI6_TX_BCLK of instance: sai6 010 **ALT2_AUDIOMIX_SAI6_RX_BCLK** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI6_RX_BCLK of instance: sai6 100 **ALT4_ENET1_RGMII_RD0** — Select mux mode: ALT4 mux port: ENET1_RGMII_RD0 of instance: enet1 101 **ALT5_GPIO4_IO[6]** — Select mux mode: ALT5 mux port: GPIO4_IO06 of instance: gpio4

8.2.4.85 SW_MUX_CTL_PAD_SAI1_RXD5 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD5)

SW_MUX_CTL Register

Address: 3033_0000h base + 164h offset = 3033_0164h



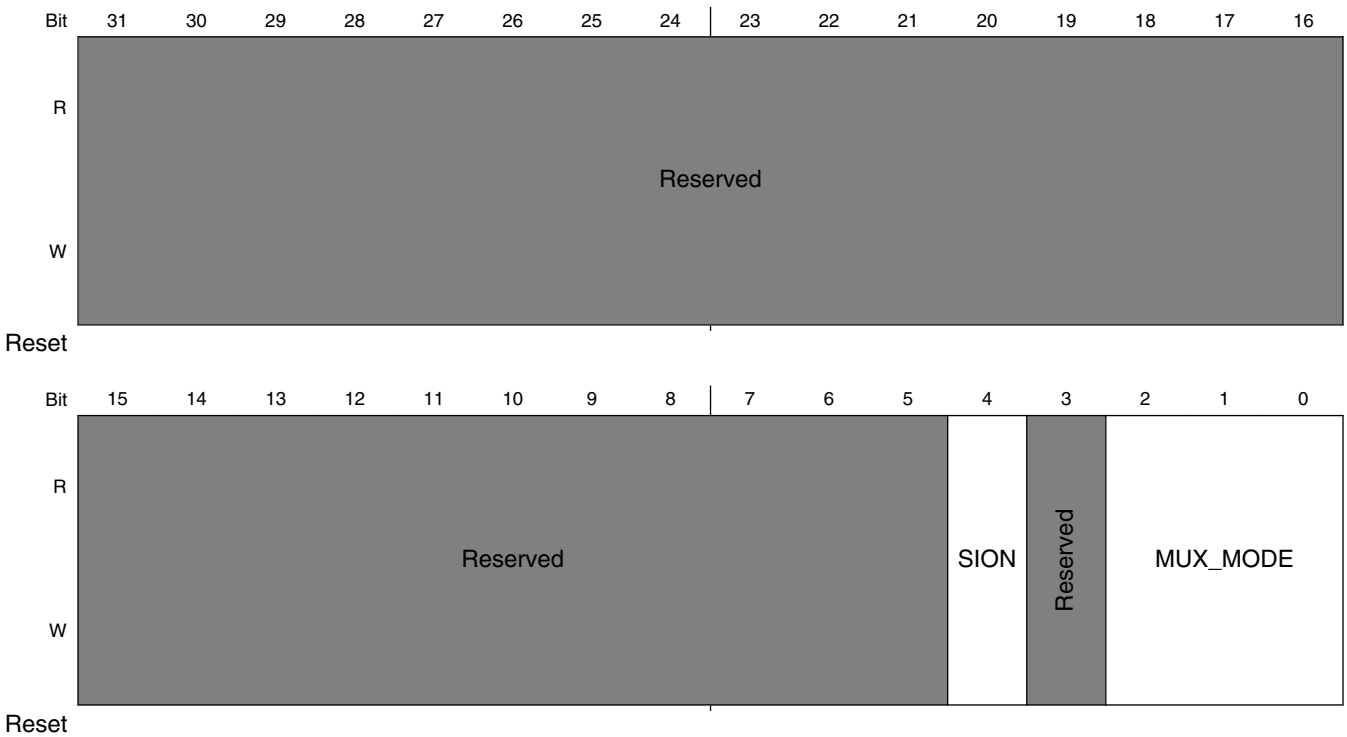
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD5 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_RXD5 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SAI1_RXD5. 000 **ALT0_AUDIOMIX_SAI1_RX_DATA[5]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_RX_DATA05 of instance: sai1 001 **ALT1_AUDIOMIX_SAI6_TX_DATA[0]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI6_TX_DATA00 of instance: sai6 010 **ALT2_AUDIOMIX_SAI6_RX_DATA[0]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI6_RX_DATA00 of instance: sai6 011 **ALT3_AUDIOMIX_SAI1_RX_SYNC** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI1_RX_SYNC of instance: sai1 100 **ALT4_ENET1_RGMII_RD1** — Select mux mode: ALT4 mux port: ENET1_RGMII_RD1 of instance: enet1 101 **ALT5_GPIO4_IO[7]** — Select mux mode: ALT5 mux port: GPIO4_IO07 of instance: gpio4

8.2.4.86 SW_MUX_CTL_PAD_SAI1_RXD6 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD6)

SW_MUX_CTL Register

Address: 3033_0000h base + 168h offset = 3033_0168h



IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD6 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_RXD6 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SAI1_RXD6. 000 **ALT0_AUDIOMIX_SAI1_RX_DATA[6]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_RX_DATA06 of instance: sai1 001 **ALT1_AUDIOMIX_SAI6_TX_SYNC** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI6_TX_SYNC of instance: sai6

Table continues on the next page...

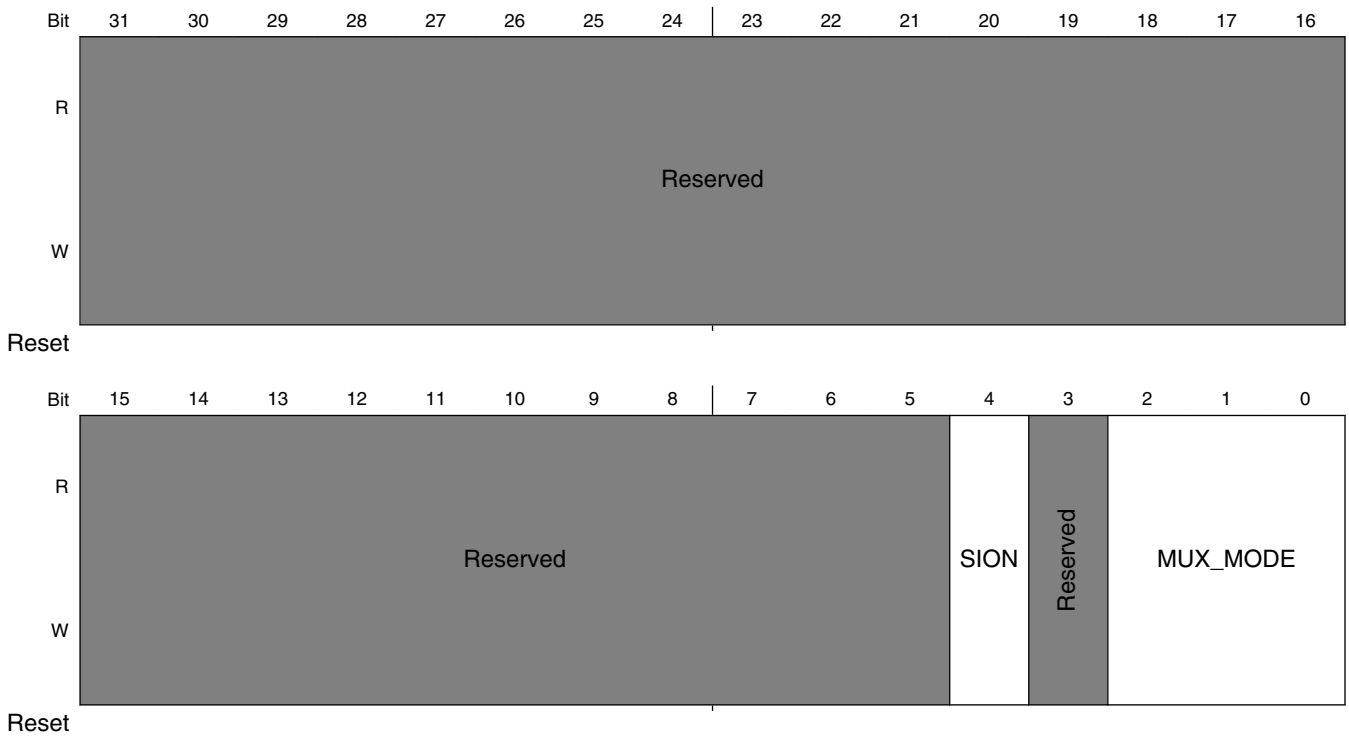
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD6 field descriptions (continued)

Field	Description
	010 **ALT2_AUDIOMIX_SAI6_RX_SYNC** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI6_RX_SYNC of instance: sai6 100 **ALT4_ENET1_RGMII_RD2** — Select mux mode: ALT4 mux port: ENET1_RGMII_RD2 of instance: enet1 101 **ALT5_GPIO4_IO[8]** — Select mux mode: ALT5 mux port: GPIO4_IO08 of instance: gpio4

8.2.4.87 SW_MUX_CTL_PAD_SAI1_RXD7 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD7)

SW_MUX_CTL Register

Address: 3033_0000h base + 16Ch offset = 3033_016Ch



IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD7 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_RXD7 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

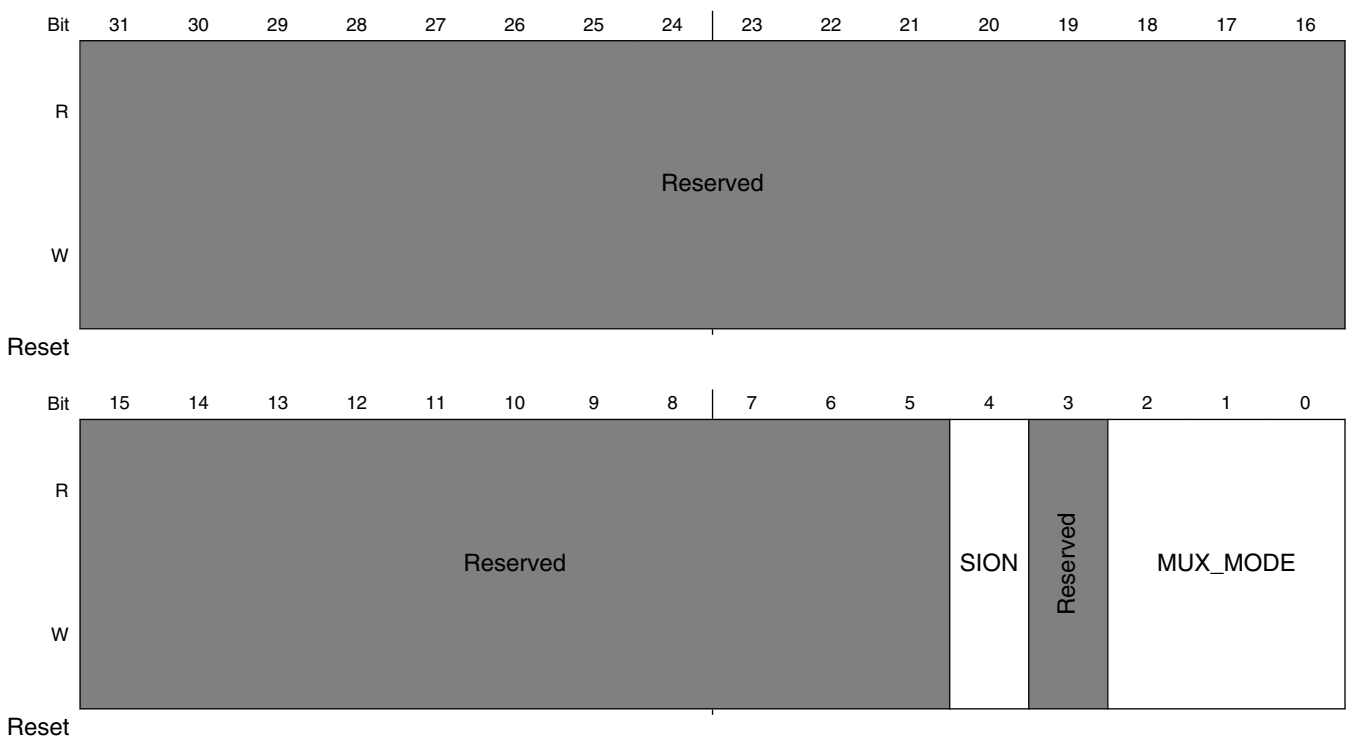
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD7 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SAI1_RXD7. 000 **ALT0_AUDIOMIX_SAI1_RX_DATA[7]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_RX_DATA07 of instance: sai1 001 **ALT1_AUDIOMIX_SAI6_MCLK** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI6_MCLK of instance: sai6 010 **ALT2_AUDIOMIX_SAI1_TX_SYNC** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI1_TX_SYNC of instance: sai1 011 **ALT3_AUDIOMIX_SAI1_TX_DATA[4]** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI1_TX_DATA04 of instance: sai1 100 **ALT4_ENET1_RGMII_RD3** — Select mux mode: ALT4 mux port: ENET1_RGMII_RD3 of instance: enet1 101 **ALT5_GPIO4_IO[9]** — Select mux mode: ALT5 mux port: GPIO4_IO09 of instance: gpio4

8.2.4.88 SW_MUX_CTL_PAD_SAI1_TXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 170h offset = 3033_0170h



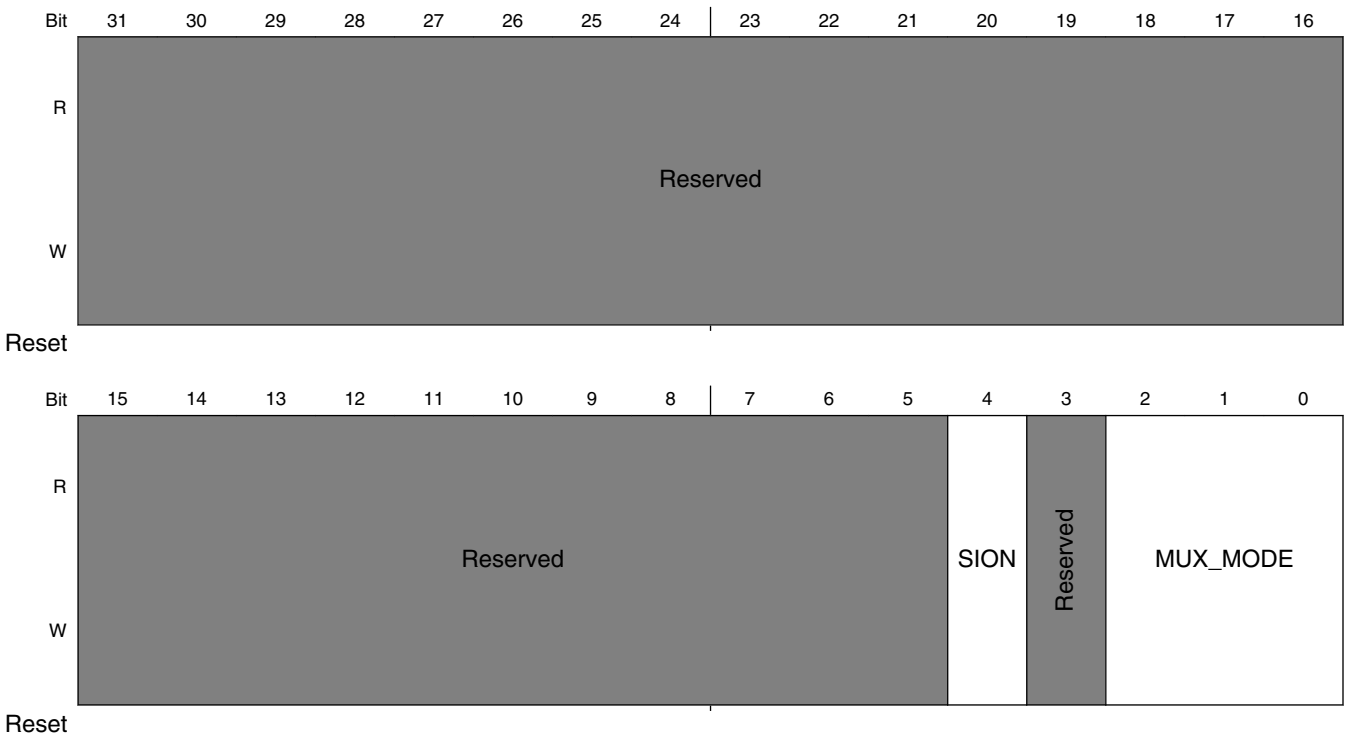
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_TXFS 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SAI1_TXFS. 000 **ALT0_AUDIOMIX_SAI1_TX_SYNC** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_TX_SYNC of instance: sai1 100 **ALT4_ENET1_RGMII_RX_CTL** — Select mux mode: ALT4 mux port: ENET1_RGMII_RX_CTL of instance: enet1 101 **ALT5_GPIO4_IO[10]** — Select mux mode: ALT5 mux port: GPIO4_IO10 of instance: gpio4

8.2.4.89 SW_MUX_CTL_PAD_SAI1_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 174h offset = 3033_0174h



IOMUXC_SW_MUX_CTL_PAD_SAI1_TXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_TXC 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SAI1_TXC. 000 **ALT0_AUDIOMIX_SAI1_TX_BCLK** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_TX_BCLK of instance: sai1

Table continues on the next page...

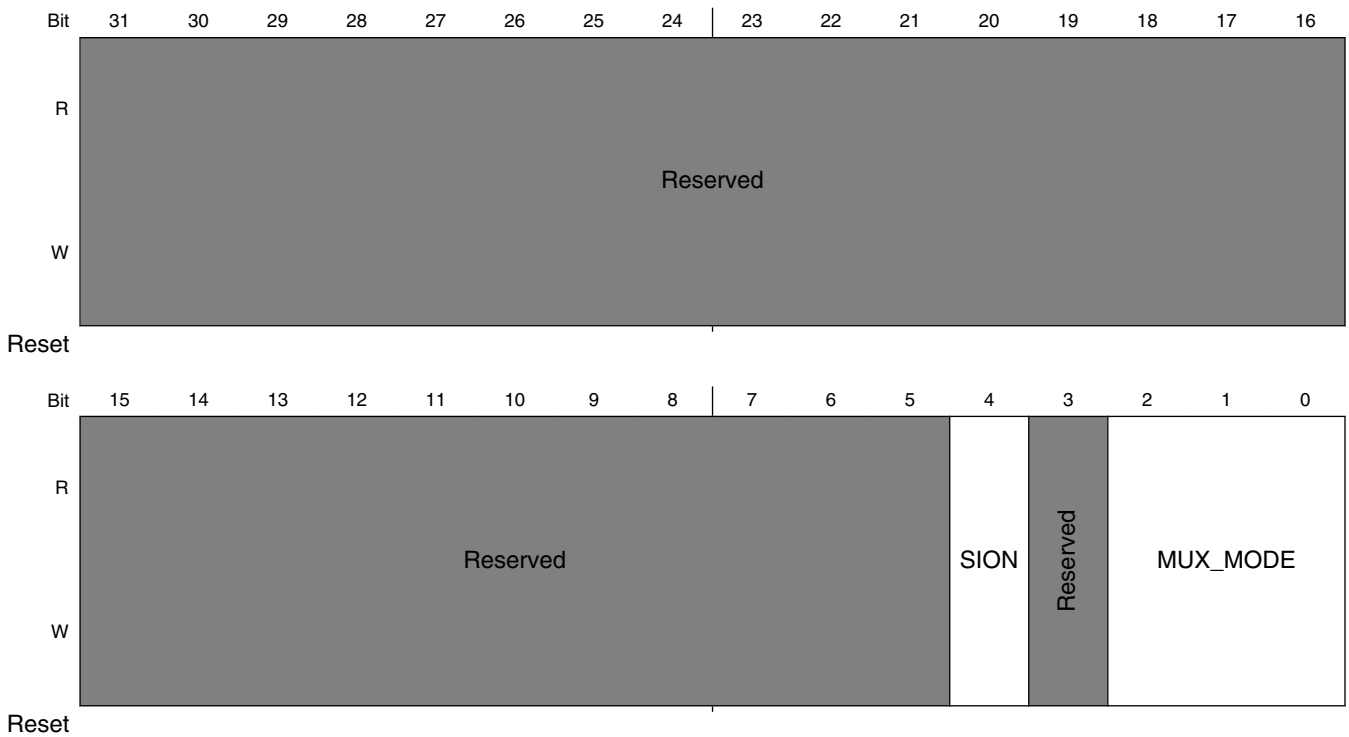
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXC field descriptions (continued)

Field	Description
	100 **ALT4_ENET1_RGMII_RXC** — Select mux mode: ALT4 mux port: ENET1_RGMII_RXC of instance: enet1 101 **ALT5_GPIO4_IO[11]** — Select mux mode: ALT5 mux port: GPIO4_IO11 of instance: gpio4

8.2.4.90 SW_MUX_CTL_PAD_SAI1_TXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 178h offset = 3033_0178h



IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_TXD0 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

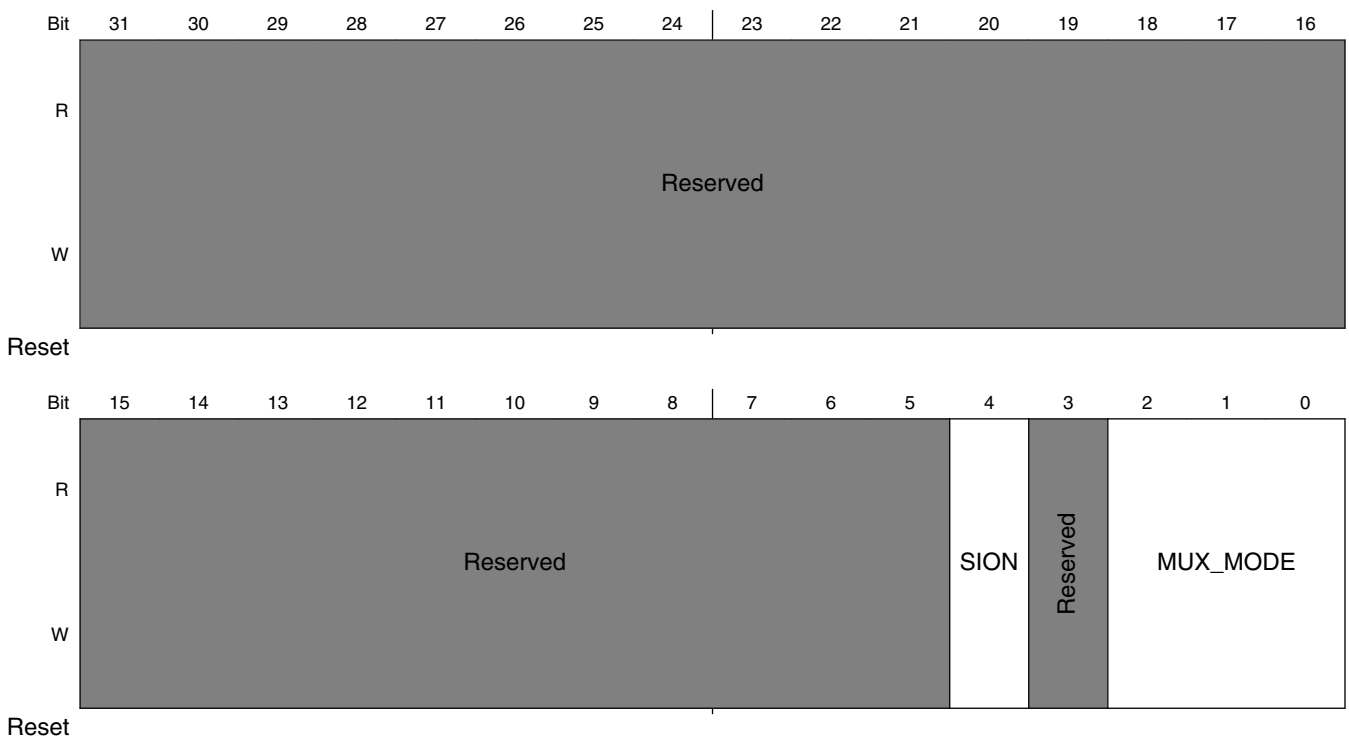
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD0 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SAI1_TXD0. 000 **ALT0_AUDIOMIX_SAI1_TX_DATA[0]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_TX_DATA00 of instance: sai1 100 **ALT4_ENET1_RGMII_TD0** — Select mux mode: ALT4 mux port: ENET1_RGMII_TD0 of instance: enet1 101 **ALT5_GPIO4_IO[12]** — Select mux mode: ALT5 mux port: GPIO4_IO12 of instance: gpio4

8.2.4.91 SW_MUX_CTL_PAD_SAI1_TXD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD1)

SW_MUX_CTL Register

Address: 3033_0000h base + 17Ch offset = 3033_017Ch



IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved

Table continues on the next page...

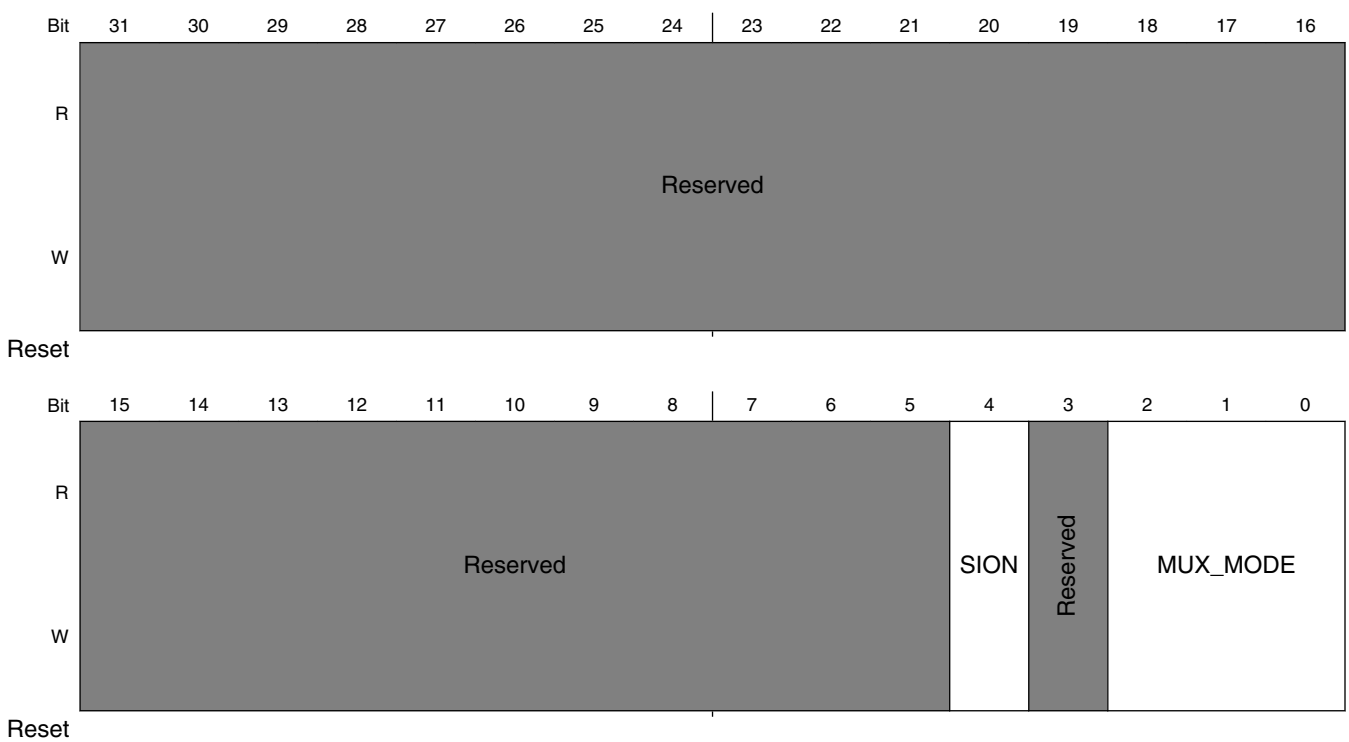
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD1 field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_TXD1 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SAI1_TXD1. 000 **ALT0_AUDIOMIX_SAI1_TX_DATA[1]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_TX_DATA01 of instance: sai1 100 **ALT4_ENET1_RGMII_TD1** — Select mux mode: ALT4 mux port: ENET1_RGMII_TD1 of instance: enet1 101 **ALT5_GPIO4_IO[13]** — Select mux mode: ALT5 mux port: GPIO4_IO13 of instance: gpio4

8.2.4.92 SW_MUX_CTL_PAD_SAI1_TXD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD2)

SW_MUX_CTL Register

Address: 3033_0000h base + 180h offset = 3033_0180h



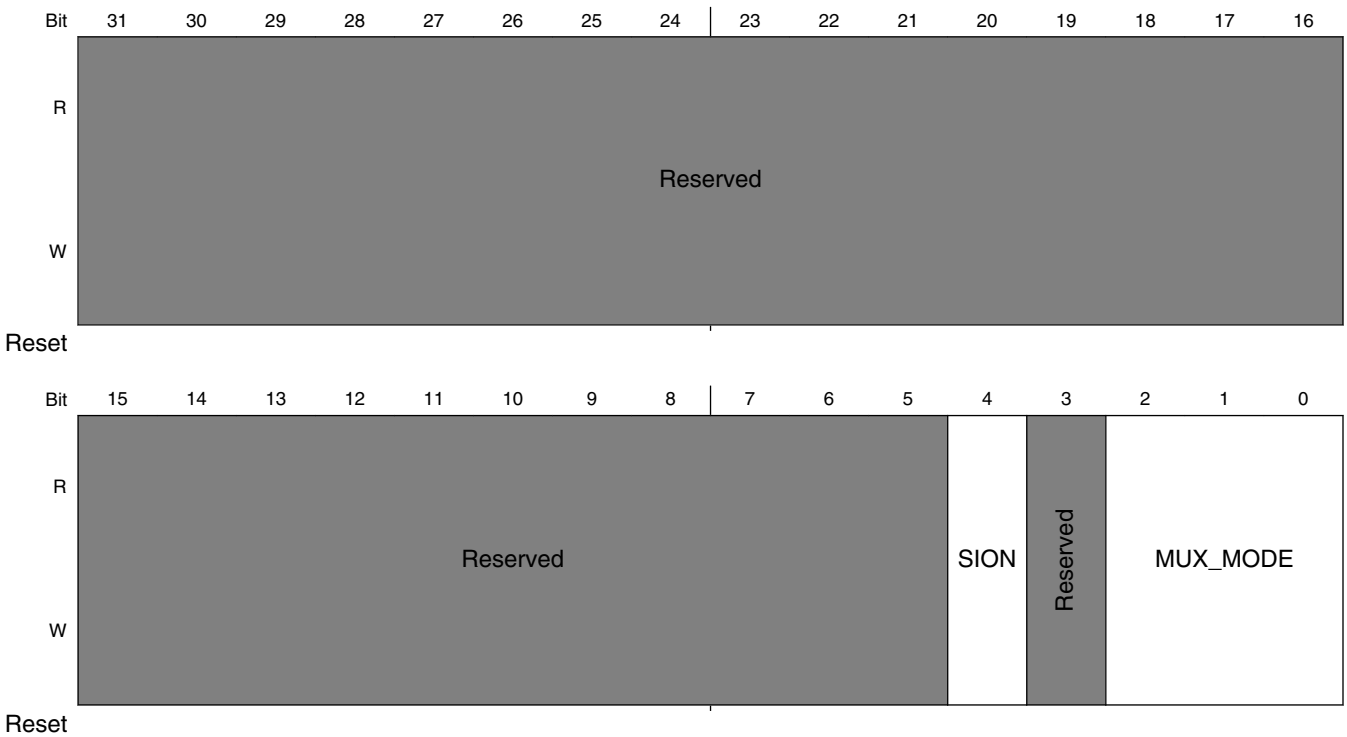
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_TXD2 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SAI1_TXD2. 000 **ALT0_AUDIOMIX_SAI1_TX_DATA[2]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_TX_DATA02 of instance: sai1 100 **ALT4_ENET1_RGMII_TD2** — Select mux mode: ALT4 mux port: ENET1_RGMII_TD2 of instance: enet1 101 **ALT5_GPIO4_IO[14]** — Select mux mode: ALT5 mux port: GPIO4_IO14 of instance: gpio4

8.2.4.93 SW_MUX_CTL_PAD_SAI1_TXD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD3)

SW_MUX_CTL Register

Address: 3033_0000h base + 184h offset = 3033_0184h



IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_TXD3 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: SAI1_TXD3. 000 **ALT0_AUDIOMIX_SAI1_TX_DATA[3]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_TX_DATA03 of instance: sai1

Table continues on the next page...

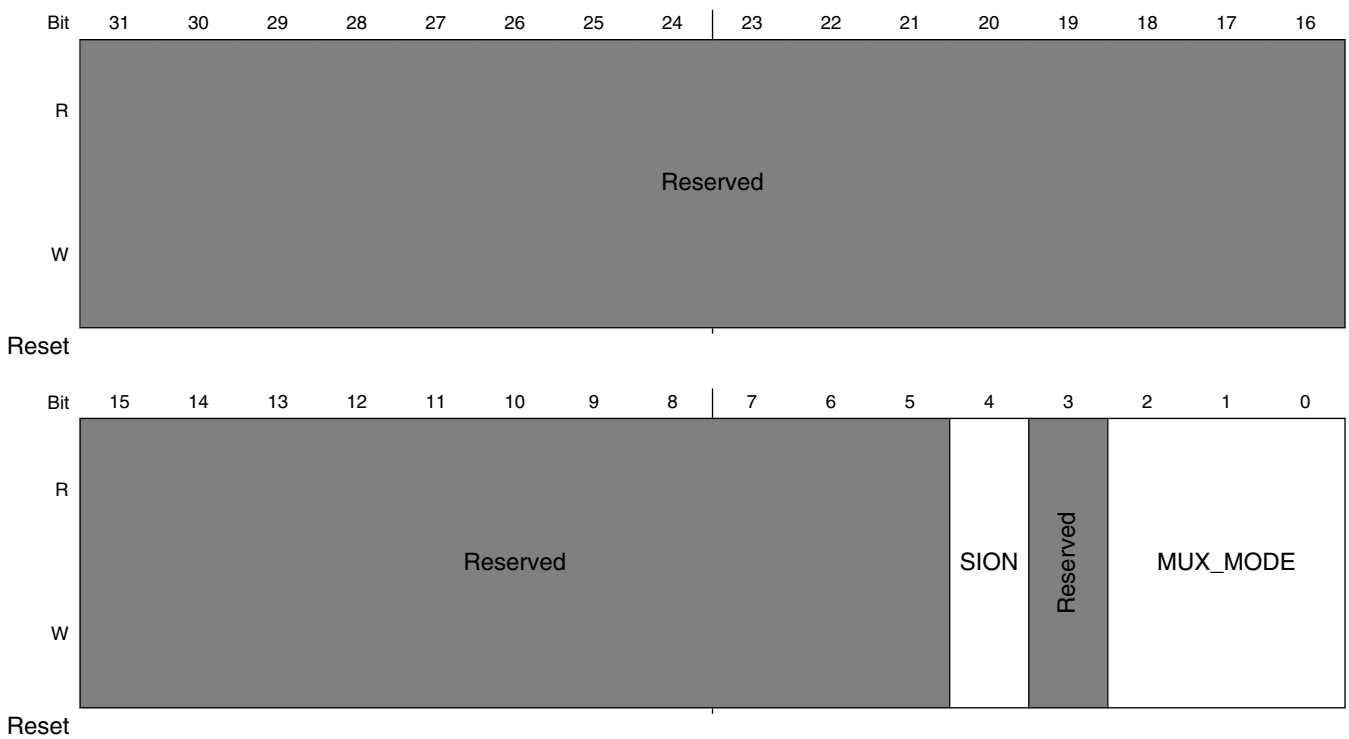
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD3 field descriptions (continued)

Field	Description
	100 **ALT4_ENET1_RGMII_TD3** — Select mux mode: ALT4 mux port: ENET1_RGMII_TD3 of instance: enet1 101 **ALT5_GPIO4_IO[15]** — Select mux mode: ALT5 mux port: GPIO4_IO15 of instance: gpio4

8.2.4.94 SW_MUX_CTL_PAD_SAI1_TXD4 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD4)

SW_MUX_CTL Register

Address: 3033_0000h base + 188h offset = 3033_0188h



IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD4 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_TXD4 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

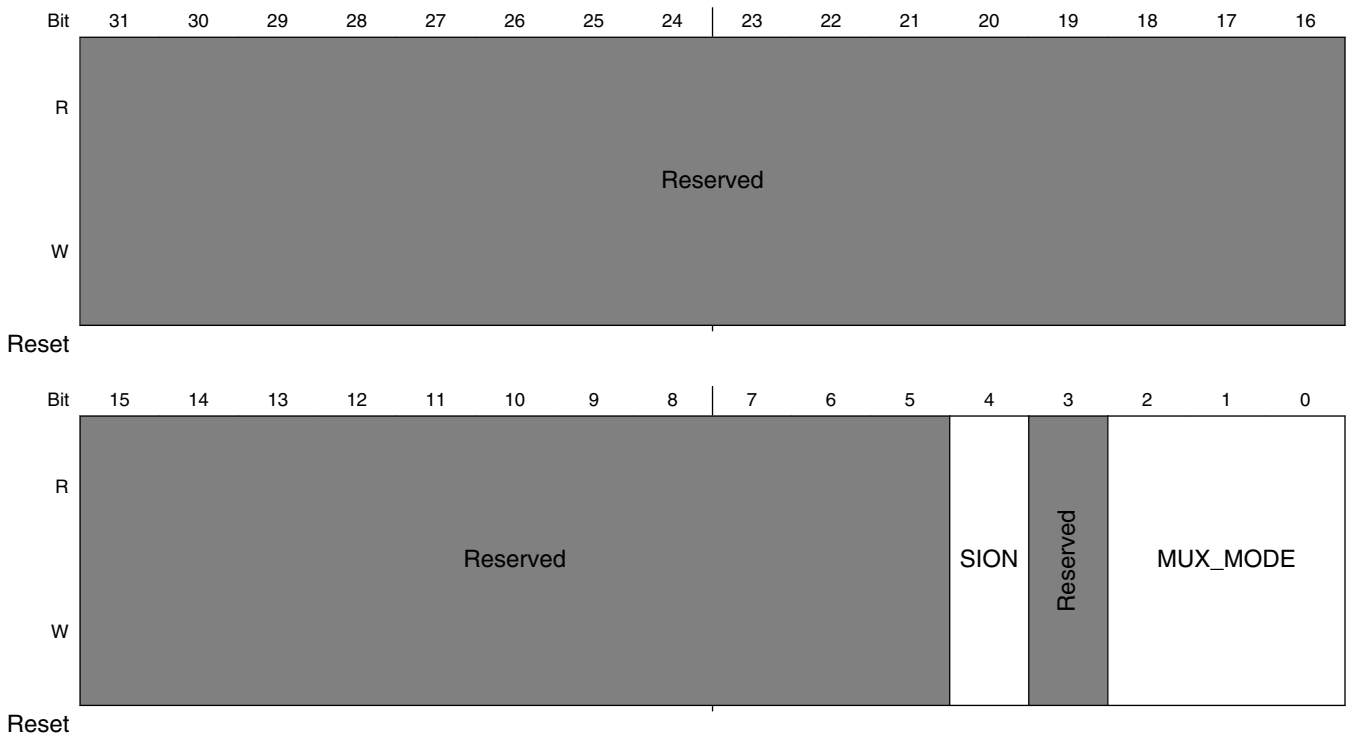
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD4 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SAI1_TXD4. 000 **ALT0_AUDIOMIX_SAI1_TX_DATA[4]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_TX_DATA04 of instance: sai1 001 **ALT1_AUDIOMIX_SAI6_RX_BCLK** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI6_RX_BCLK of instance: sai6 010 **ALT2_AUDIOMIX_SAI6_TX_BCLK** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI6_TX_BCLK of instance: sai6 100 **ALT4_ENET1_RGMII_TX_CTL** — Select mux mode: ALT4 mux port: ENET1_RGMII_TX_CTL of instance: enet1 101 **ALT5_GPIO4_IO[16]** — Select mux mode: ALT5 mux port: GPIO4_IO16 of instance: gpio4

8.2.4.95 SW_MUX_CTL_PAD_SAI1_TXD5 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD5)

SW_MUX_CTL Register

Address: 3033_0000h base + 18Ch offset = 3033_018Ch



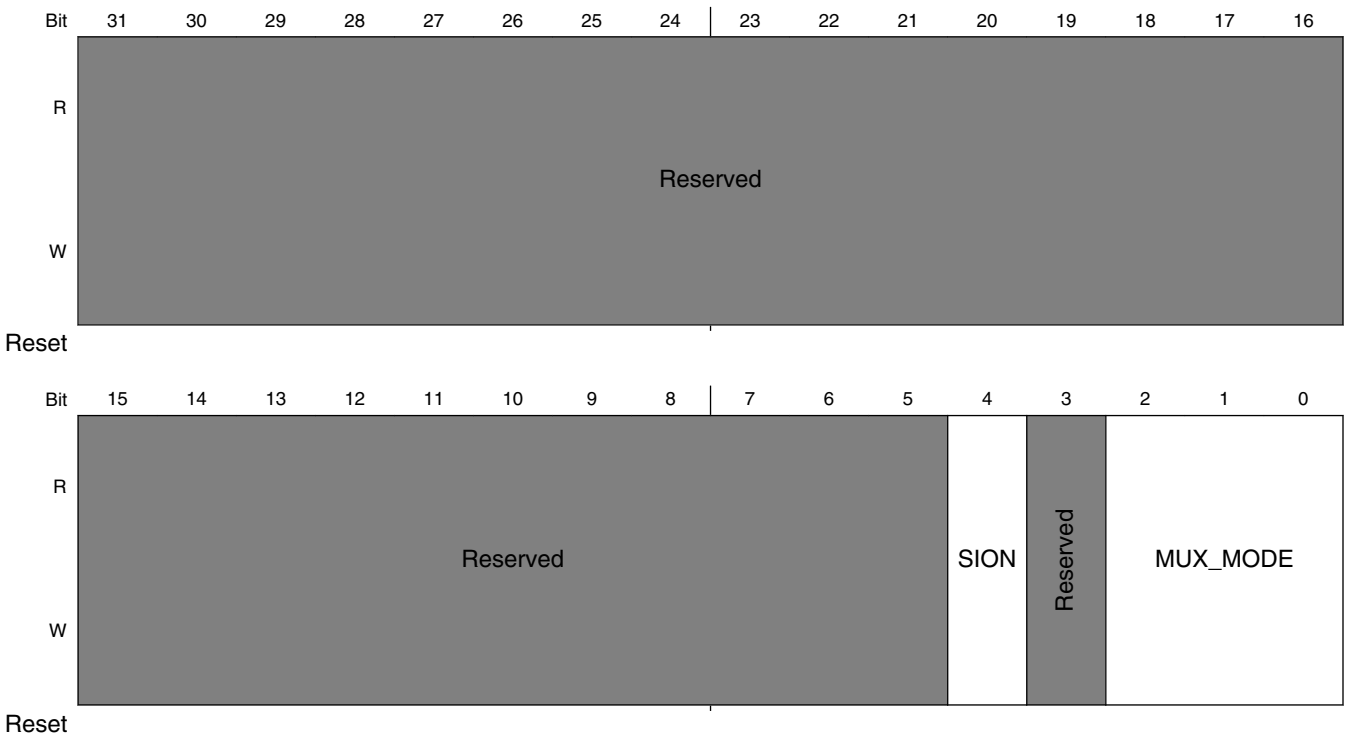
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD5 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_TXD5 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SAI1_TXD5. 000 **ALT0_AUDIOMIX_SAI1_TX_DATA[5]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_TX_DATA05 of instance: sai1 001 **ALT1_AUDIOMIX_SAI6_RX_DATA[0]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI6_RX_DATA00 of instance: sai6 010 **ALT2_AUDIOMIX_SAI6_TX_DATA[0]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI6_TX_DATA00 of instance: sai6 100 **ALT4_ENET1_RGMII_TXC** — Select mux mode: ALT4 mux port: ENET1_RGMII_TXC of instance: enet1 101 **ALT5_GPIO4_IO[17]** — Select mux mode: ALT5 mux port: GPIO4_IO17 of instance: gpio4

8.2.4.96 SW_MUX_CTL_PAD_SAI1_TXD6 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD6)

SW_MUX_CTL Register

Address: 3033_0000h base + 190h offset = 3033_0190h



IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD6 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_TXD6 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SAI1_TXD6. 000 **ALT0_AUDIOMIX_SAI1_TX_DATA[6]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_TX_DATA06 of instance: sai1 001 **ALT1_AUDIOMIX_SAI6_RX_SYNC** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI6_RX_SYNC of instance: sai6

Table continues on the next page...

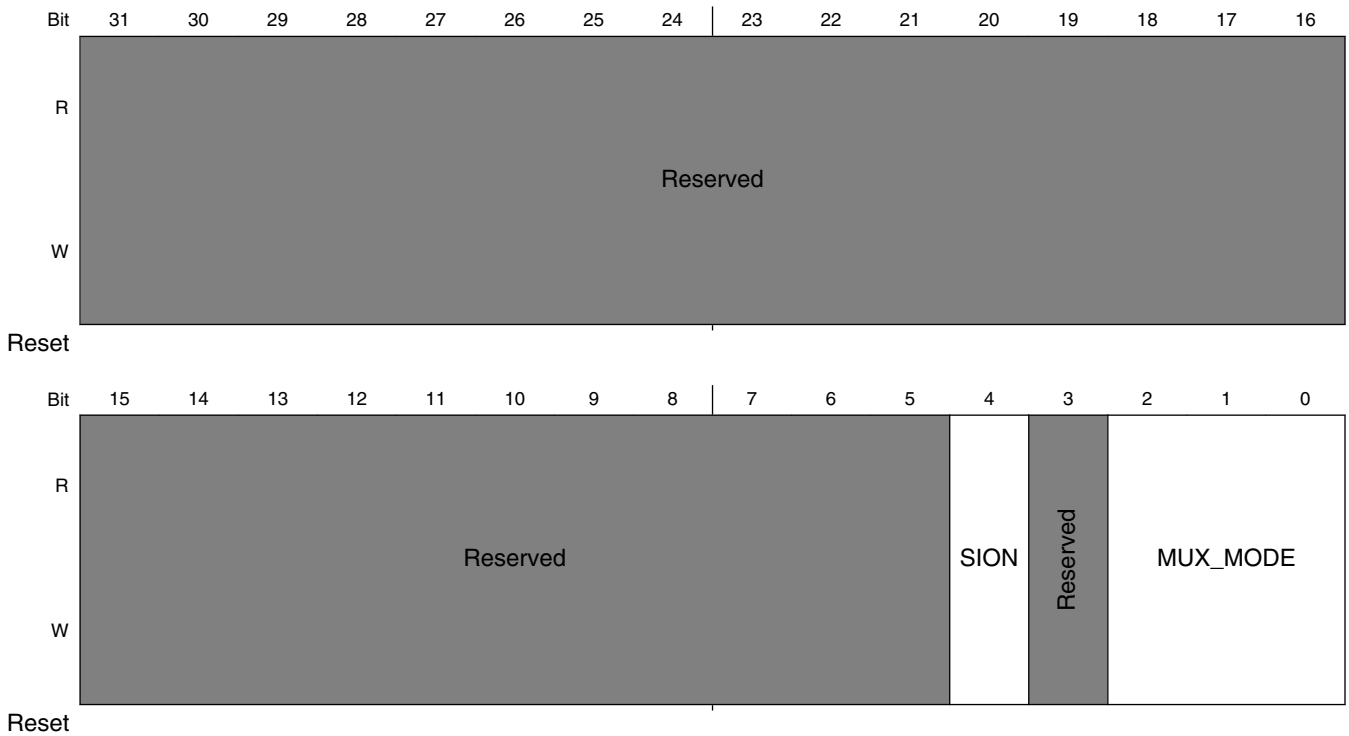
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD6 field descriptions (continued)

Field	Description
	010 **ALT2_AUDIOMIX_SAI6_TX_SYNC** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI6_TX_SYNC of instance: sai6 100 **ALT4_ENET1_RX_ER** — Select mux mode: ALT4 mux port: ENET1_RX_ER of instance: enet1 101 **ALT5_GPIO4_IO[18]** — Select mux mode: ALT5 mux port: GPIO4_IO18 of instance: gpio4

8.2.4.97 SW_MUX_CTL_PAD_SAI1_TXD7 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD7)

SW_MUX_CTL Register

Address: 3033_0000h base + 194h offset = 3033_0194h



IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD7 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_TXD7 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

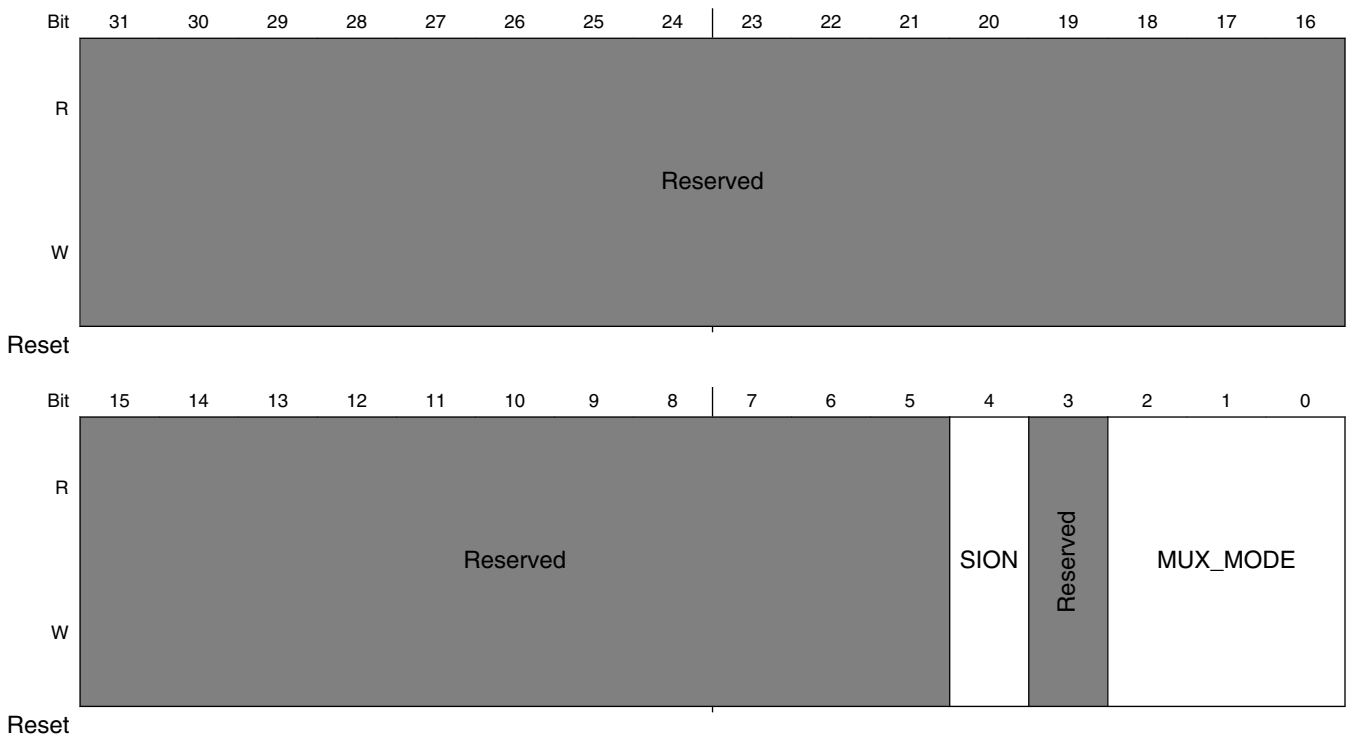
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD7 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SAI1_TXD7. 000 **ALT0_AUDIOMIX_SAI1_TX_DATA[7]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_TX_DATA07 of instance: sai1 001 **ALT1_AUDIOMIX_SAI6_MCLK** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI6_MCLK of instance: sai6 011 **ALT3_AUDIOMIX_PDM_CLK** — Select mux mode: ALT3 mux port: AUDIOMIX_PDM_CLK of instance: pdm 100 **ALT4_ENET1_TX_ER** — Select mux mode: ALT4 mux port: ENET1_TX_ER of instance: enet1 101 **ALT5_GPIO4_IO[19]** — Select mux mode: ALT5 mux port: GPIO4_IO19 of instance: gpio4

8.2.4.98 SW_MUX_CTL_PAD_SAI1_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_MCLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 198h offset = 3033_0198h



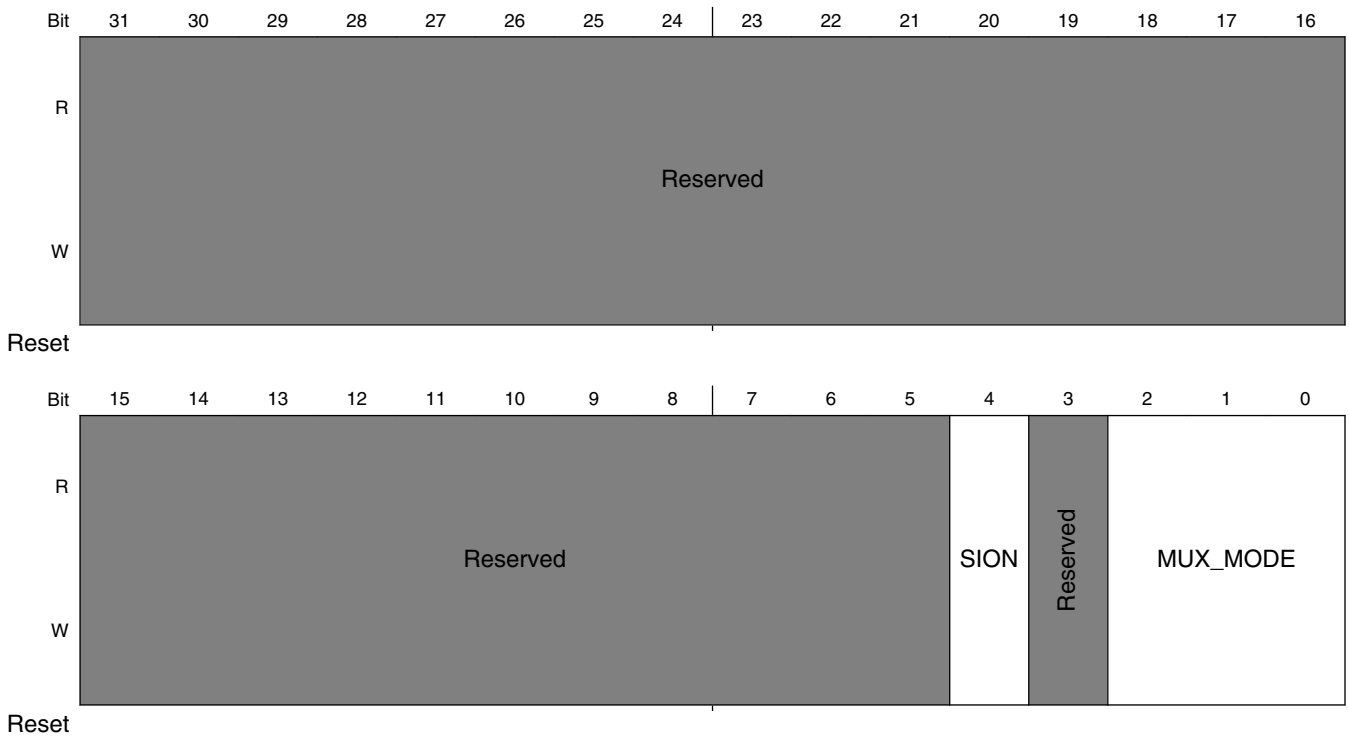
IOMUXC_SW_MUX_CTL_PAD_SAI1_MCLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI1_MCLK 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SAI1_MCLK. 000 **ALT0_AUDIOMIX_SAI1_MCLK** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI1_MCLK of instance: sai1 010 **ALT2_AUDIOMIX_SAI1_TX_BCLK** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI1_TX_BCLK of instance: sai1 100 **ALT4_ENET1_TX_CLK** — Select mux mode: ALT4 mux port: ENET1_TX_CLK of instance: enet1 101 **ALT5_GPIO4_IO[20]** — Select mux mode: ALT5 mux port: GPIO4_IO20 of instance: gpio4

8.2.4.99 SW_MUX_CTL_PAD_SAI2_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_RXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 19Ch offset = 3033_019Ch



IOMUXC_SW_MUX_CTL_PAD_SAI2_RXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI2_RXFS 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: SAI2_RXFS. 000 **ALT0_AUDIOMIX_SAI2_RX_SYNC** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI2_RX_SYNC of instance: sai2 001 **ALT1_AUDIOMIX_SAI5_TX_SYNC** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI5_TX_SYNC of instance: sai5

Table continues on the next page...

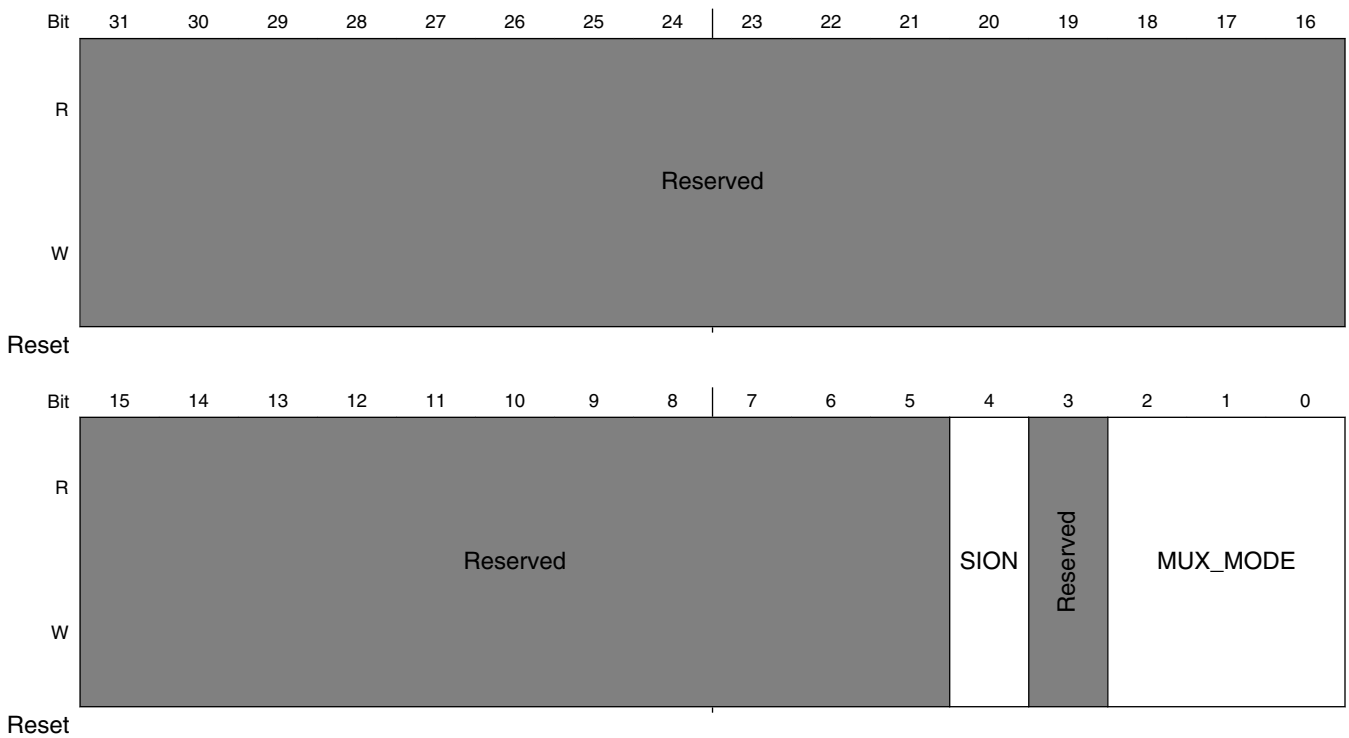
IOMUXC_SW_MUX_CTL_PAD_SAI2_RXFS field descriptions (continued)

Field	Description
	010 **ALT2_AUDIOMIX_SAI5_TX_DATA[1]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI5_TX_DATA01 of instance: sai5 011 **ALT3_AUDIOMIX_SAI2_RX_DATA[1]** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI2_RX_DATA01 of instance: sai2 100 **ALT4_UART1_TX** — Select mux mode: ALT4 mux port: UART1_TX of instance: uart1 101 **ALT5_GPIO4_IO[21]** — Select mux mode: ALT5 mux port: GPIO4_IO21 of instance: gpio4 110 **ALT6_AUDIOMIX_PDM_BIT_STREAM[2]** — Select mux mode: ALT6 mux port: AUDIOMIX_PDM_BIT_STREAM02 of instance: pdm

8.2.4.100 SW_MUX_CTL_PAD_SAI2_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_RXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 1A0h offset = 3033_01A0h



IOMUXC_SW_MUX_CTL_PAD_SAI2_RXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field.

Table continues on the next page...

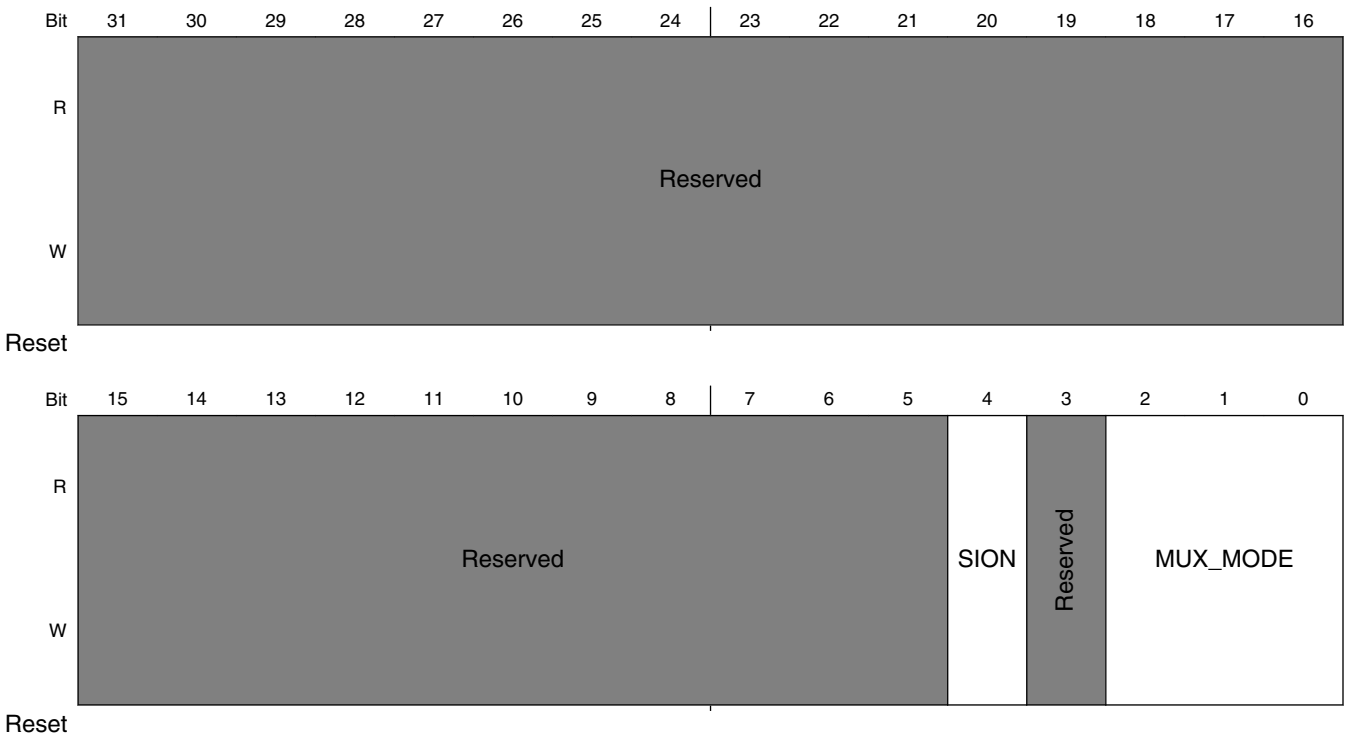
IOMUXC_SW_MUX_CTL_PAD_SAI2_RXC field descriptions (continued)

Field	Description
	Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI2_RXC 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: SAI2_RXC. 000 **ALT0_AUDIOMIX_SAI2_RX_BCLK** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI2_RX_BCLK of instance: sai2 001 **ALT1_AUDIOMIX_SAI5_TX_BCLK** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI5_TX_BCLK of instance: sai5 011 **ALT3_CAN1_TX** — Select mux mode: ALT3 mux port: CAN1_TX of instance: can1 100 **ALT4_UART1_RX** — Select mux mode: ALT4 mux port: UART1_RX of instance: uart1 101 **ALT5_GPIO4_IO[22]** — Select mux mode: ALT5 mux port: GPIO4_IO22 of instance: gpio4 110 **ALT6_AUDIOMIX_PDM_BIT_STREAM[1]** — Select mux mode: ALT6 mux port: AUDIOMIX_PDM_BIT_STREAM01 of instance: pdm

8.2.4.101 SW_MUX_CTL_PAD_SAI2_RXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_RXD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 1A4h offset = 3033_01A4h



IOMUXC_SW_MUX_CTL_PAD_SAI2_RXD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI2_RXD0 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: SAI2_RXD0. 000 **ALT0_AUDIOMIX_SAI2_RX_DATA[0]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI2_RX_DATA00 of instance: sai2 001 **ALT1_AUDIOMIX_SAI5_TX_DATA[0]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI5_TX_DATA00 of instance: sai5

Table continues on the next page...

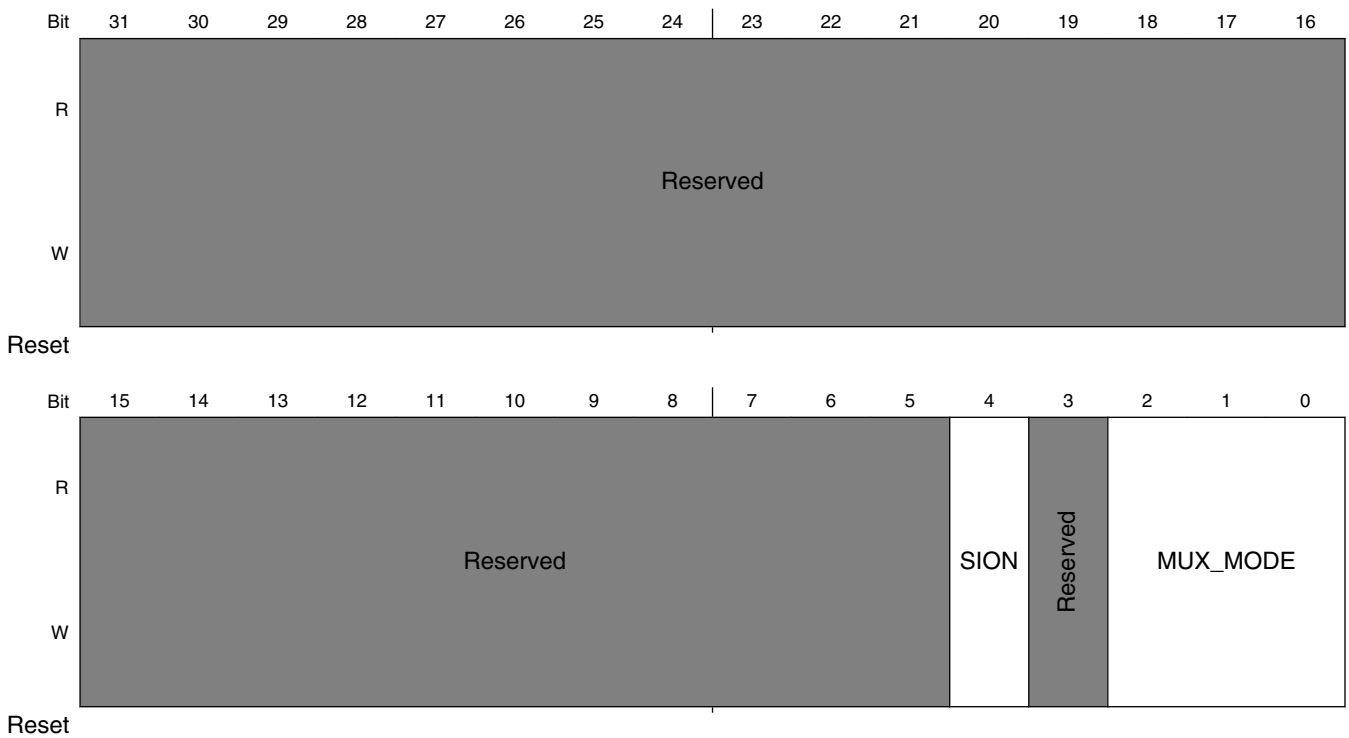
IOMUXC_SW_MUX_CTL_PAD_SAI2_RXD0 field descriptions (continued)

Field	Description
	010 **ALT2_ENET_QOS_1588_EVENT2_OUT** — Select mux mode: ALT2 mux port: ENET_QOS_1588_EVENT2_OUT of instance: enet_qos 011 **ALT3_AUDIOMIX_SAI2_TX_DATA[1]** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI2_TX_DATA01 of instance: sai2 100 **ALT4_UART1_RTS_B** — Select mux mode: ALT4 mux port: UART1_RTS_B of instance: uart1 101 **ALT5_GPIO4_IO[23]** — Select mux mode: ALT5 mux port: GPIO4_IO23 of instance: gpio4 110 **ALT6_AUDIOMIX_PDM_BIT_STREAM[3]** — Select mux mode: ALT6 mux port: AUDIOMIX_PDM_BIT_STREAM03 of instance: pdm

8.2.4.102 SW_MUX_CTL_PAD_SAI2_TXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_TXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 1A8h offset = 3033_01A8h



IOMUXC_SW_MUX_CTL_PAD_SAI2_TXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field.

Table continues on the next page...

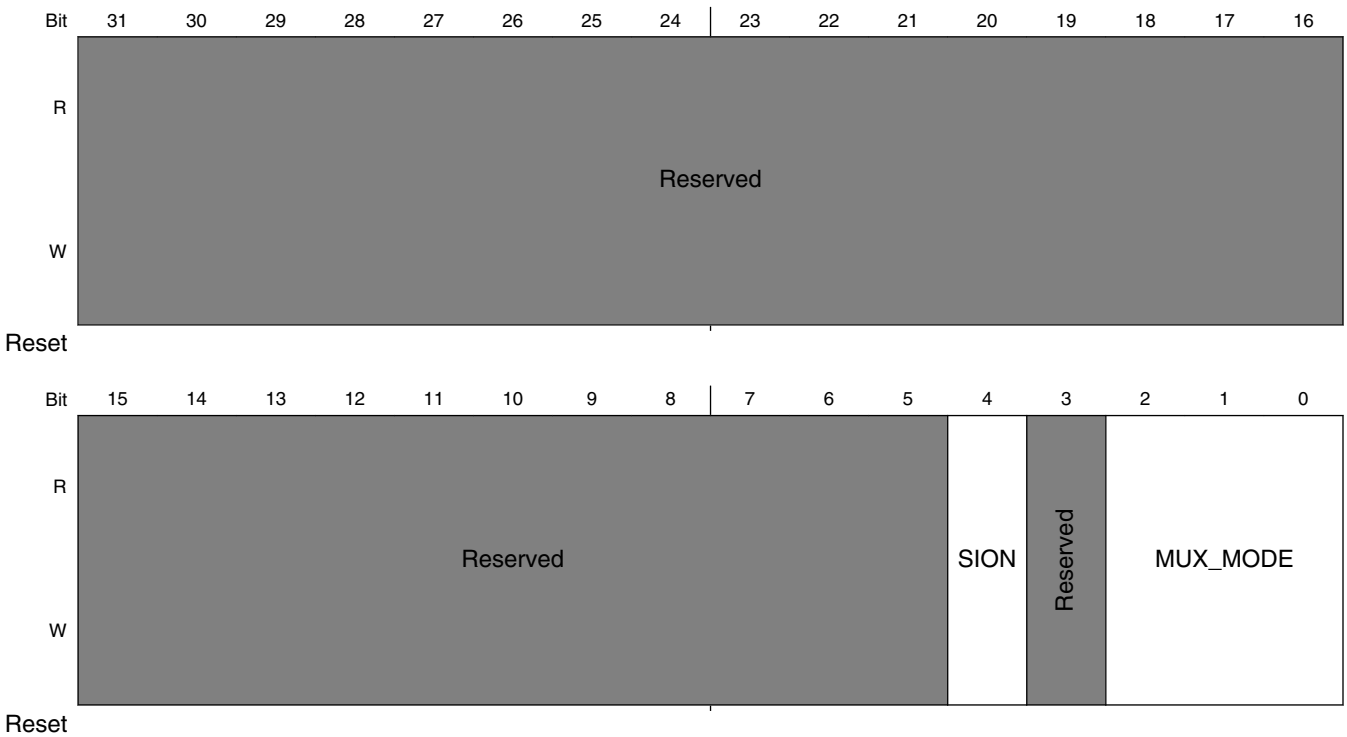
IOMUXC_SW_MUX_CTL_PAD_SAI2_TXFS field descriptions (continued)

Field	Description
	Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI2_TXFS 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: SAI2_TXFS. 000 **ALT0_AUDIOMIX_SAI2_TX_SYNC** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI2_TX_SYNC of instance: sai2 001 **ALT1_AUDIOMIX_SAI5_TX_DATA[1]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI5_TX_DATA01 of instance: sai5 010 **ALT2_ENET_QOS_1588_EVENT3_OUT** — Select mux mode: ALT2 mux port: ENET_QOS_1588_EVENT3_OUT of instance: enet_qos 011 **ALT3_AUDIOMIX_SAI2_TX_DATA[1]** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI2_TX_DATA01 of instance: sai2 100 **ALT4_UART1_CTS_B** — Select mux mode: ALT4 mux port: UART1_CTS_B of instance: uart1 101 **ALT5_GPIO4_IO[24]** — Select mux mode: ALT5 mux port: GPIO4_IO24 of instance: gpio4 110 **ALT6_AUDIOMIX_PDM_BIT_STREAM[2]** — Select mux mode: ALT6 mux port: AUDIOMIX_PDM_BIT_STREAM02 of instance: pdm

8.2.4.103 SW_MUX_CTL_PAD_SAI2_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_TXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 1ACh offset = 3033_01ACh



IOMUXC_SW_MUX_CTL_PAD_SAI2_TXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI2_TXC 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SAI2_TXC. 000 **ALT0_AUDIOMIX_SAI2_TX_BCLK** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI2_TX_BCLK of instance: sai2 001 **ALT1_AUDIOMIX_SAI5_TX_DATA[2]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI5_TX_DATA02 of instance: sai5

Table continues on the next page...

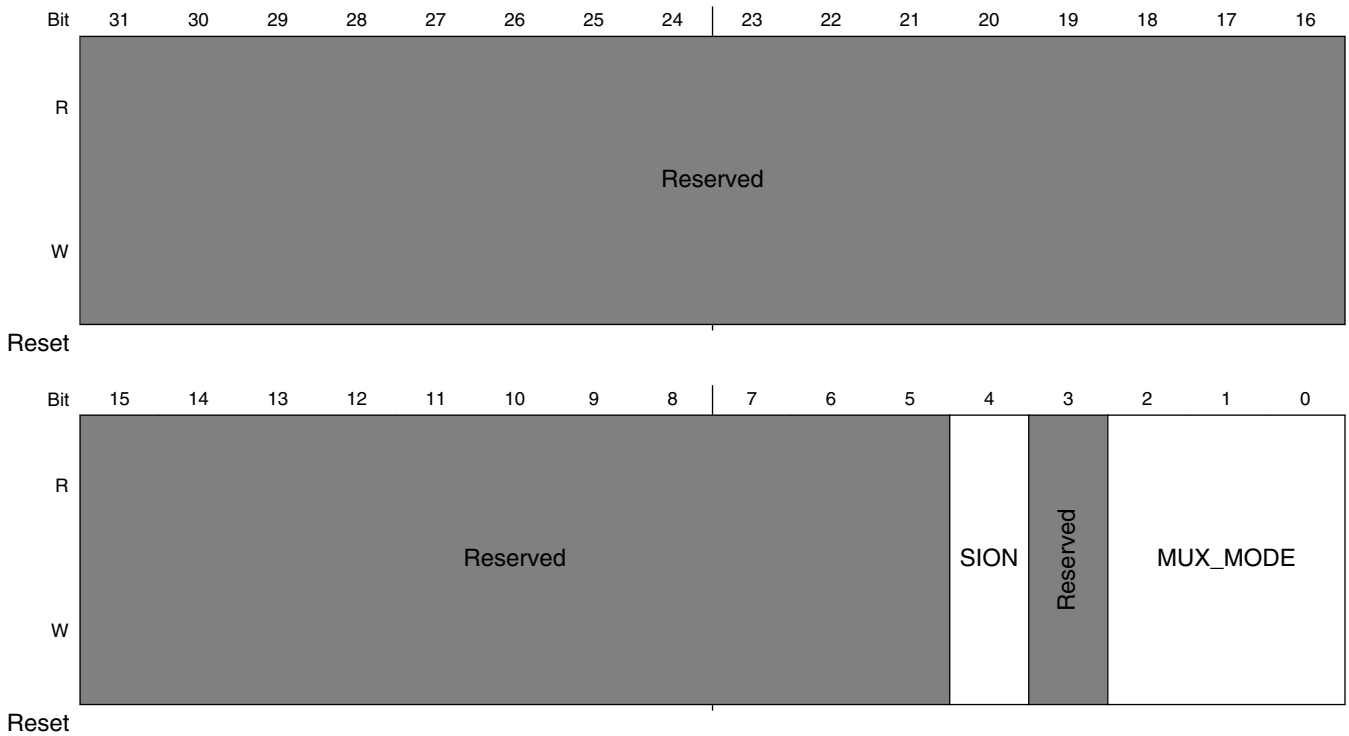
IOMUXC_SW_MUX_CTL_PAD_SAI2_TXC field descriptions (continued)

Field	Description
	011 **ALT3_CAN1_RX** — Select mux mode: ALT3 mux port: CAN1_RX of instance: can1 101 **ALT5_GPIO4_IO[25]** — Select mux mode: ALT5 mux port: GPIO4_IO25 of instance: gpio4 110 **ALT6_AUDIOMIX_PDM_BIT_STREAM[1]** — Select mux mode: ALT6 mux port: AUDIOMIX_PDM_BIT_STREAM01 of instance: pdm

8.2.4.104 SW_MUX_CTL_PAD_SAI2_TXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_TXD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 1B0h offset = 3033_01B0h



IOMUXC_SW_MUX_CTL_PAD_SAI2_TXD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI2_TXD0 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

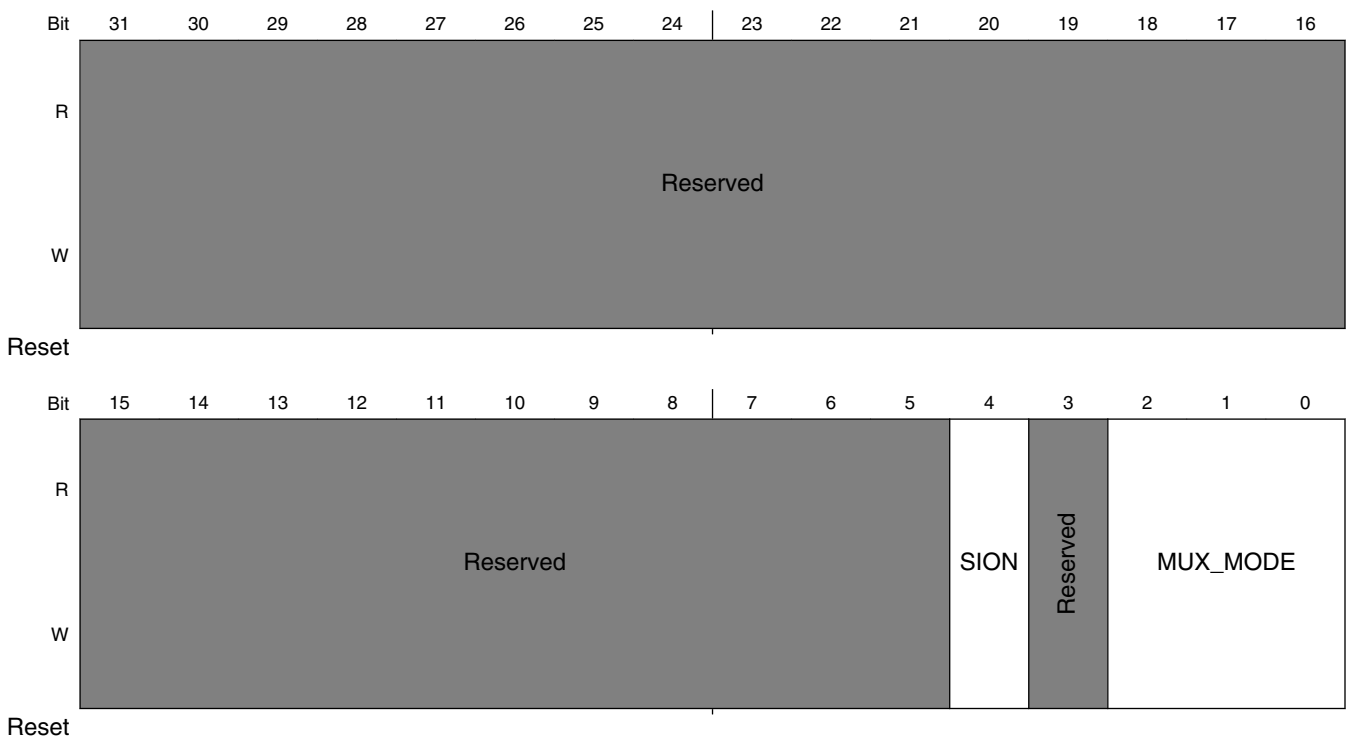
IOMUXC_SW_MUX_CTL_PAD_SAI2_TXD0 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SAI2_TXD0. 000 **ALT0_AUDIOMIX_SAI2_TX_DATA[0]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI2_TX_DATA00 of instance: sai2 001 **ALT1_AUDIOMIX_SAI5_TX_DATA[3]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI5_TX_DATA03 of instance: sai5 010 **ALT2_ENET_QOS_1588_EVENT2_IN** — Select mux mode: ALT2 mux port: ENET_QOS_1588_EVENT2_IN of instance: enet_qos 011 **ALT3_CAN2_TX** — Select mux mode: ALT3 mux port: CAN2_TX of instance: can2 100 **ALT4_ENET_QOS_1588_EVENT2_AUX_IN** — Select mux mode: ALT4 mux port: ENET_QOS_1588_EVENT2_AUX_IN of instance: enet_qos 101 **ALT5_GPIO4_IO[26]** — Select mux mode: ALT5 mux port: GPIO4_IO26 of instance: gpio4

8.2.4.105 SW_MUX_CTL_PAD_SAI2_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_MCLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 1B4h offset = 3033_01B4h



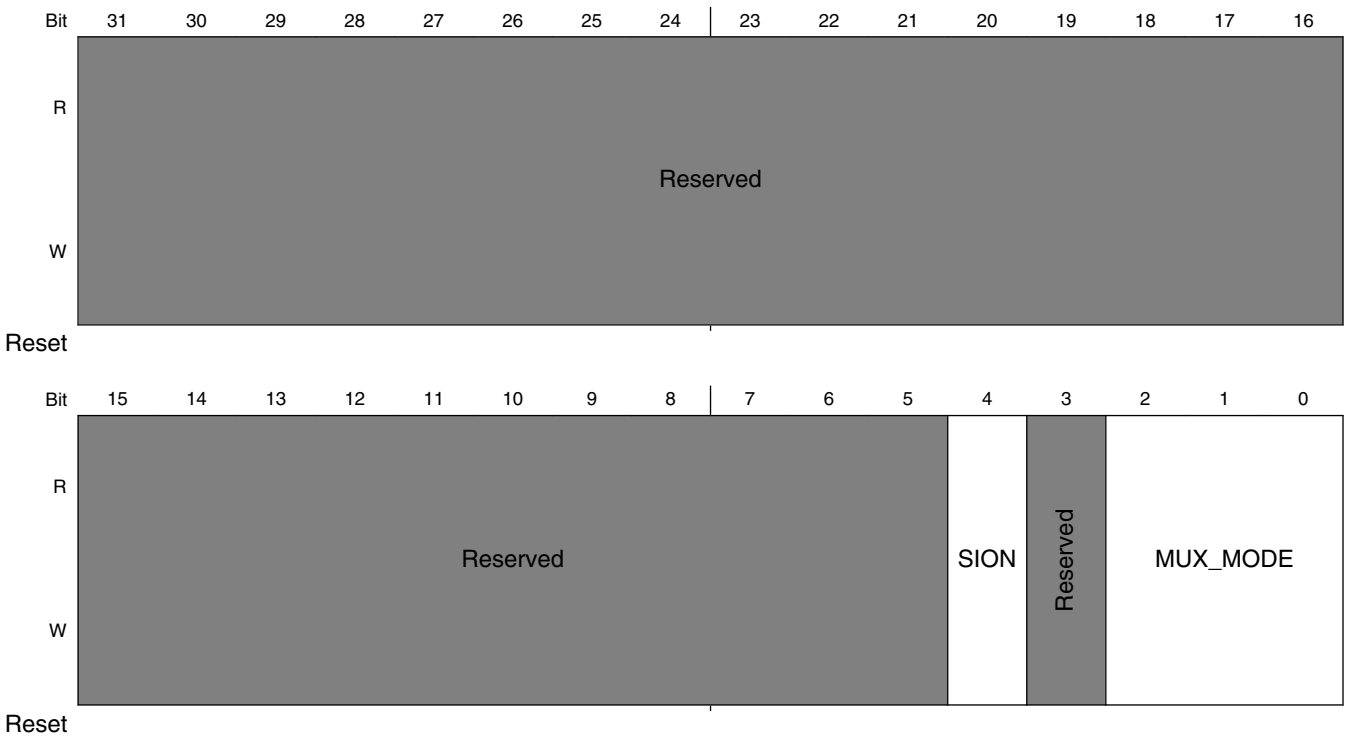
IOMUXC_SW_MUX_CTL_PAD_SAI2_MCLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI2_MCLK 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: SAI2_MCLK. 000 **ALT0_AUDIOMIX_SAI2_MCLK** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI2_MCLK of instance: sai2 001 **ALT1_AUDIOMIX_SAI5_MCLK** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI5_MCLK of instance: sai5 010 **ALT2_ENET_QOS_1588_EVENT3_IN** — Select mux mode: ALT2 mux port: ENET_QOS_1588_EVENT3_IN of instance: enet_qos 011 **ALT3_CAN2_RX** — Select mux mode: ALT3 mux port: CAN2_RX of instance: can2 100 **ALT4_ENET_QOS_1588_EVENT3_AUX_IN** — Select mux mode: ALT4 mux port: ENET_QOS_1588_EVENT3_AUX_IN of instance: enet_qos 101 **ALT5_GPIO4_IO[27]** — Select mux mode: ALT5 mux port: GPIO4_IO27 of instance: gpio4 110 **ALT6_AUDIOMIX_SAI3_MCLK** — Select mux mode: ALT6 mux port: AUDIOMIX_SAI3_MCLK of instance: sai3

8.2.4.106 SW_MUX_CTL_PAD_SAI3_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_RXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 1B8h offset = 3033_01B8h



IOMUXC_SW_MUX_CTL_PAD_SAI3_RXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI3_RXFS 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: SAI3_RXFS. 000 **ALT0_AUDIOMIX_SAI3_RX_SYNC** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI3_RX_SYNC of instance: sai3 001 **ALT1_AUDIOMIX_SAI2_RX_DATA[1]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI2_RX_DATA01 of instance: sai2

Table continues on the next page...

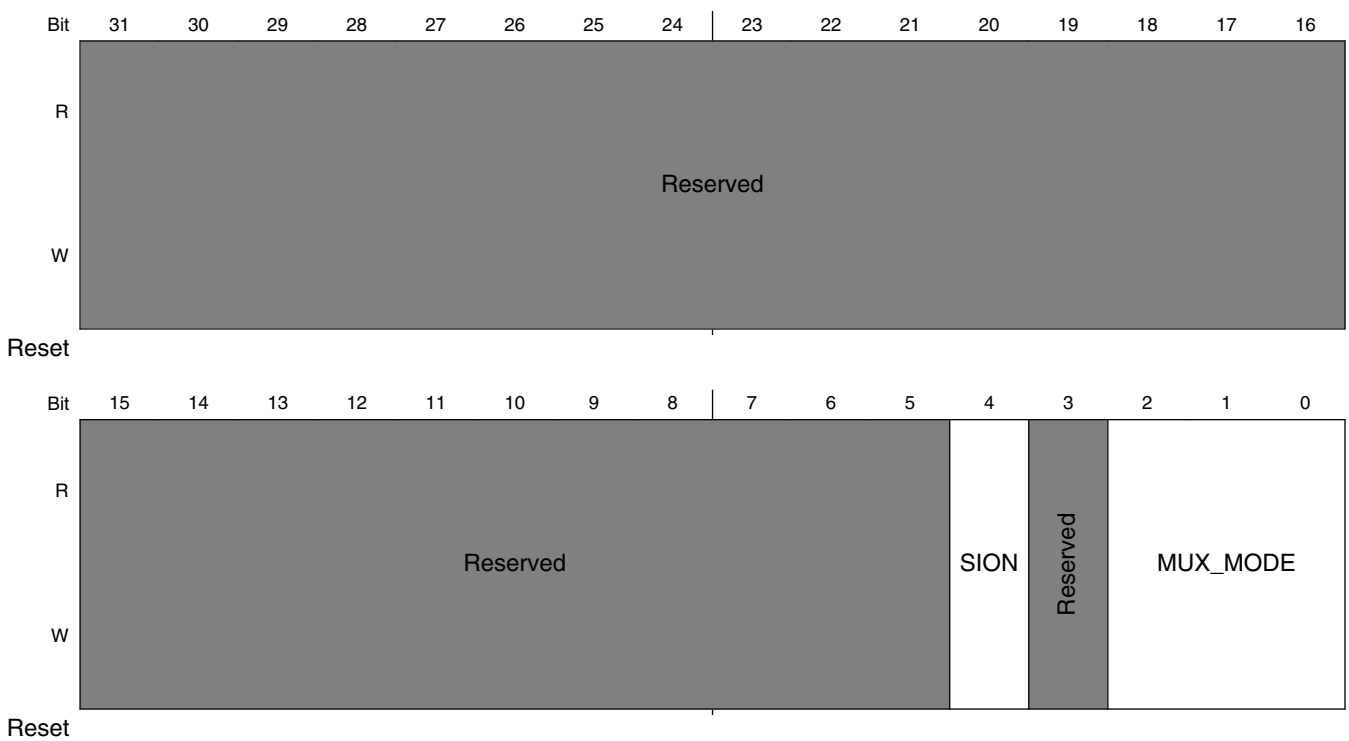
IOMUXC_SW_MUX_CTL_PAD_SAI3_RXFS field descriptions (continued)

Field	Description
	010 **ALT2_AUDIOMIX_SAI5_RX_SYNC** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI5_RX_SYNC of instance: sai5 011 **ALT3_AUDIOMIX_SAI3_RX_DATA[1]** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI3_RX_DATA01 of instance: sai3 100 **ALT4_AUDIOMIX_SPDIF1_IN** — Select mux mode: ALT4 mux port: AUDIOMIX_SPDIF1_IN of instance: spdif 101 **ALT5_GPIO4_IO[28]** — Select mux mode: ALT5 mux port: GPIO4_IO28 of instance: gpio4 110 **ALT6_AUDIOMIX_PDM_BIT_STREAM[0]** — Select mux mode: ALT6 mux port: AUDIOMIX_PDM_BIT_STREAM00 of instance: pdm

8.2.4.107 SW_MUX_CTL_PAD_SAI3_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_RXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 1BCh offset = 3033_01BCh



IOMUXC_SW_MUX_CTL_PAD_SAI3_RXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved

Table continues on the next page...

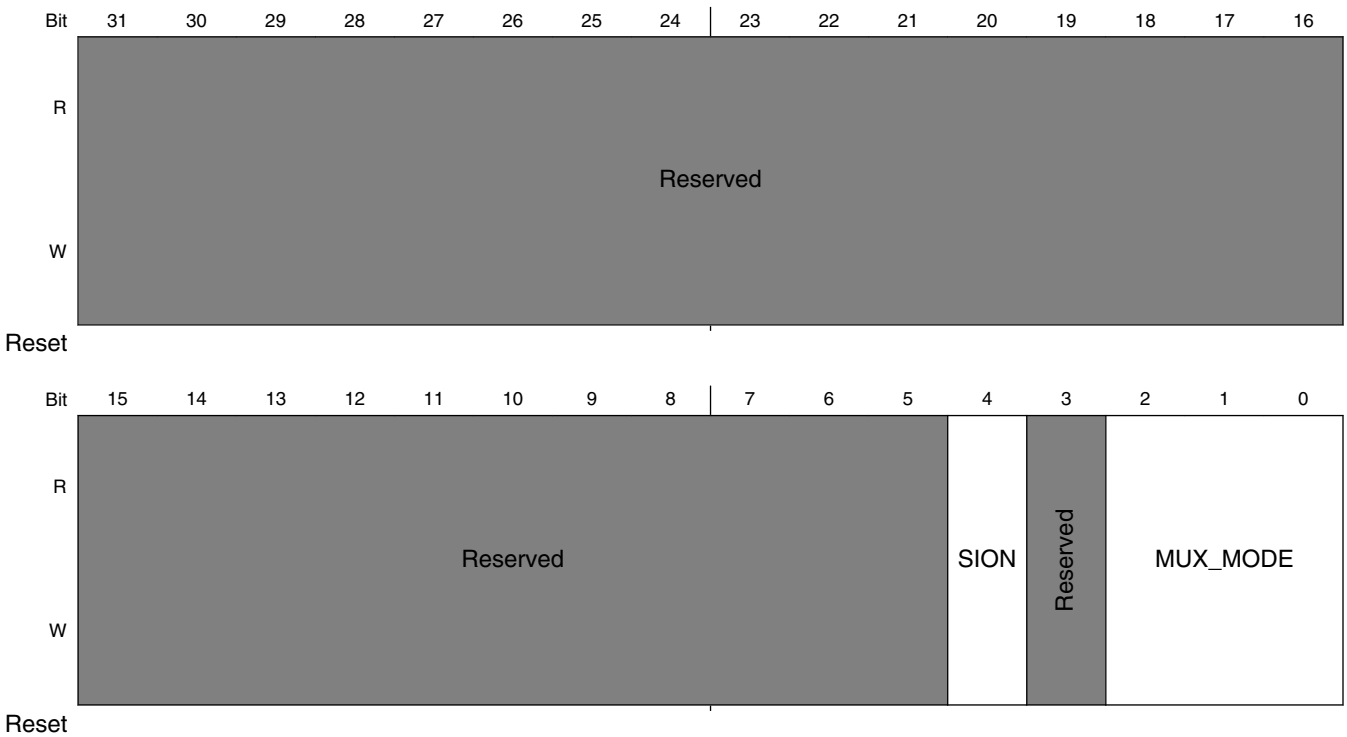
IOMUXC_SW_MUX_CTL_PAD_SAI3_RXC field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI3_RXC 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: SAI3_RXC. 000 **ALT0_AUDIOMIX_SAI3_RX_BCLK** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI3_RX_BCLK of instance: sai3 001 **ALT1_AUDIOMIX_SAI2_RX_DATA[2]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI2_RX_DATA02 of instance: sai2 010 **ALT2_AUDIOMIX_SAI5_RX_BCLK** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI5_RX_BCLK of instance: sai5 011 **ALT3_GPT1_CLK** — Select mux mode: ALT3 mux port: GPT1_CLK of instance: gpt1 100 **ALT4_UART2_CTS_B** — Select mux mode: ALT4 mux port: UART2_CTS_B of instance: uart2 101 **ALT5_GPIO4_IO[29]** — Select mux mode: ALT5 mux port: GPIO4_IO29 of instance: gpio4 110 **ALT6_AUDIOMIX_PDM_CLK** — Select mux mode: ALT6 mux port: AUDIOMIX_PDM_CLK of instance: pdm

8.2.4.108 SW_MUX_CTL_PAD_SAI3_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_RXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 1C0h offset = 3033_01C0h



IOMUXC_SW_MUX_CTL_PAD_SAI3_RXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI3_RXD 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: SAI3_RXD. 000 **ALT0_AUDIOMIX_SAI3_RX_DATA[0]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI3_RX_DATA00 of instance: sai3 001 **ALT1_AUDIOMIX_SAI2_RX_DATA[3]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI2_RX_DATA03 of instance: sai2

Table continues on the next page...

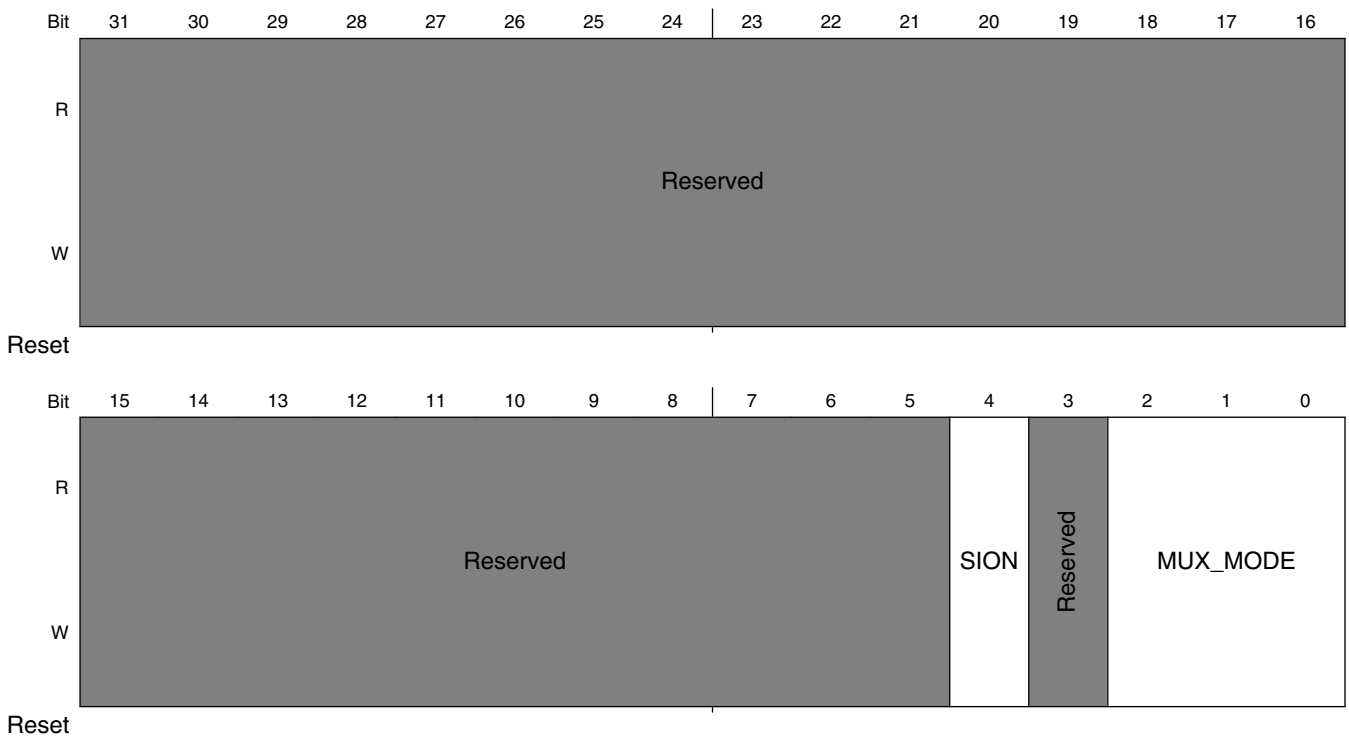
IOMUXC_SW_MUX_CTL_PAD_SAI3_RXD field descriptions (continued)

Field	Description
	010 **ALT2_AUDIOMIX_SAI5_RX_DATA[0]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI5_RX_DATA00 of instance: sai5 100 **ALT4_UART2_RTS_B** — Select mux mode: ALT4 mux port: UART2_RTS_B of instance: uart2 101 **ALT5_GPIO4_IO[30]** — Select mux mode: ALT5 mux port: GPIO4_IO30 of instance: gpio4 110 **ALT6_AUDIOMIX_PDM_BIT_STREAM[1]** — Select mux mode: ALT6 mux port: AUDIOMIX_PDM_BIT_STREAM01 of instance: pdm

8.2.4.109 SW_MUX_CTL_PAD_SAI3_TXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_TXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 1C4h offset = 3033_01C4h



IOMUXC_SW_MUX_CTL_PAD_SAI3_TXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality.

Table continues on the next page...

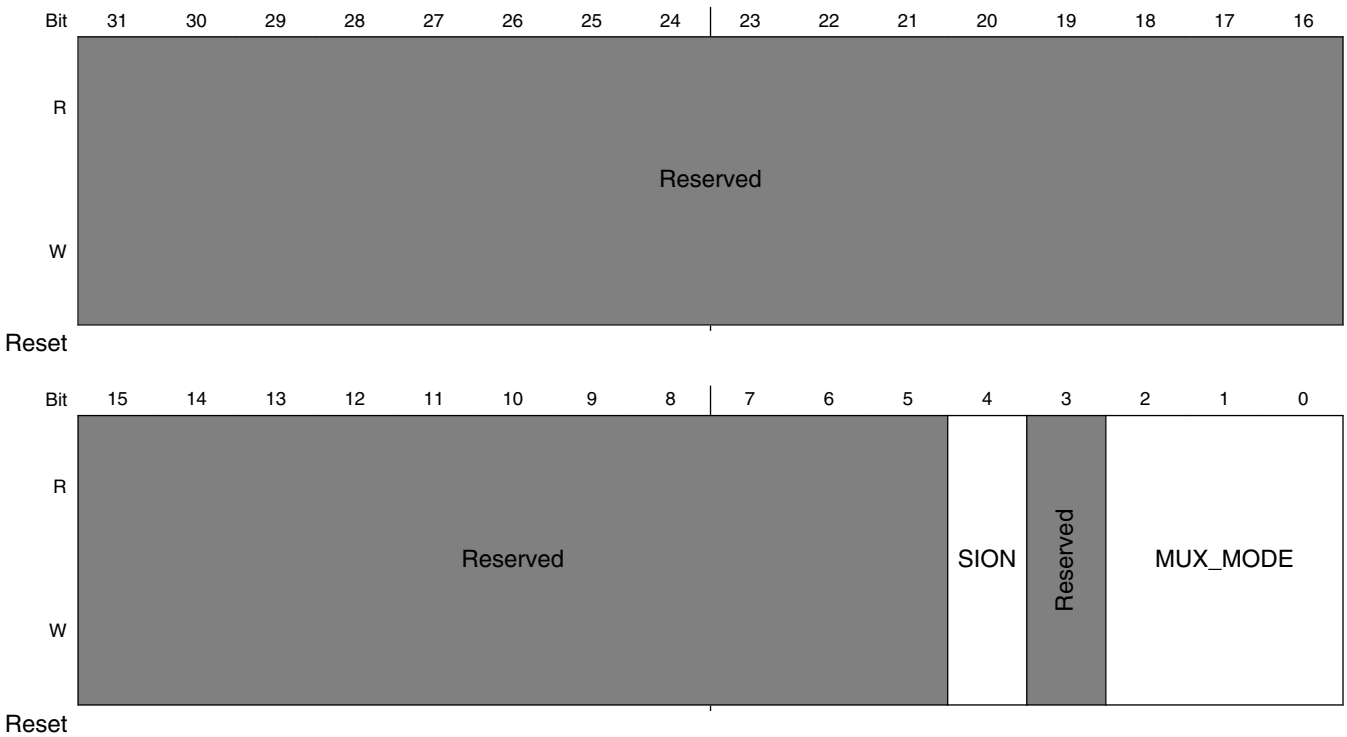
IOMUXC_SW_MUX_CTL_PAD_SAI3_TXFS field descriptions (continued)

Field	Description
	1 **ENABLED** — Force input path of pad SAI3_TXFS 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: SAI3_TXFS. 000 **ALT0_AUDIOMIX_SAI3_TX_SYNC** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI3_TX_SYNC of instance: sai3 001 **ALT1_AUDIOMIX_SAI2_TX_DATA[1]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI2_TX_DATA01 of instance: sai2 010 **ALT2_AUDIOMIX_SAI5_RX_DATA[1]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI5_RX_DATA01 of instance: sai5 011 **ALT3_AUDIOMIX_SAI3_TX_DATA[1]** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI3_TX_DATA01 of instance: sai3 100 **ALT4_UART2_RX** — Select mux mode: ALT4 mux port: UART2_RX of instance: uart2 101 **ALT5_GPIO4_IO[31]** — Select mux mode: ALT5 mux port: GPIO4_IO31 of instance: gpio4 110 **ALT6_AUDIOMIX_PDM_BIT_STREAM[3]** — Select mux mode: ALT6 mux port: AUDIOMIX_PDM_BIT_STREAM03 of instance: pdm

8.2.4.110 SW_MUX_CTL_PAD_SAI3_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_TXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 1C8h offset = 3033_01C8h



IOMUXC_SW_MUX_CTL_PAD_SAI3_TXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI3_TXC 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: SAI3_TXC. 000 **ALT0_AUDIOMIX_SAI3_TX_BCLK** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI3_TX_BCLK of instance: sai3 001 **ALT1_AUDIOMIX_SAI2_TX_DATA[2]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI2_TX_DATA02 of instance: sai2

Table continues on the next page...

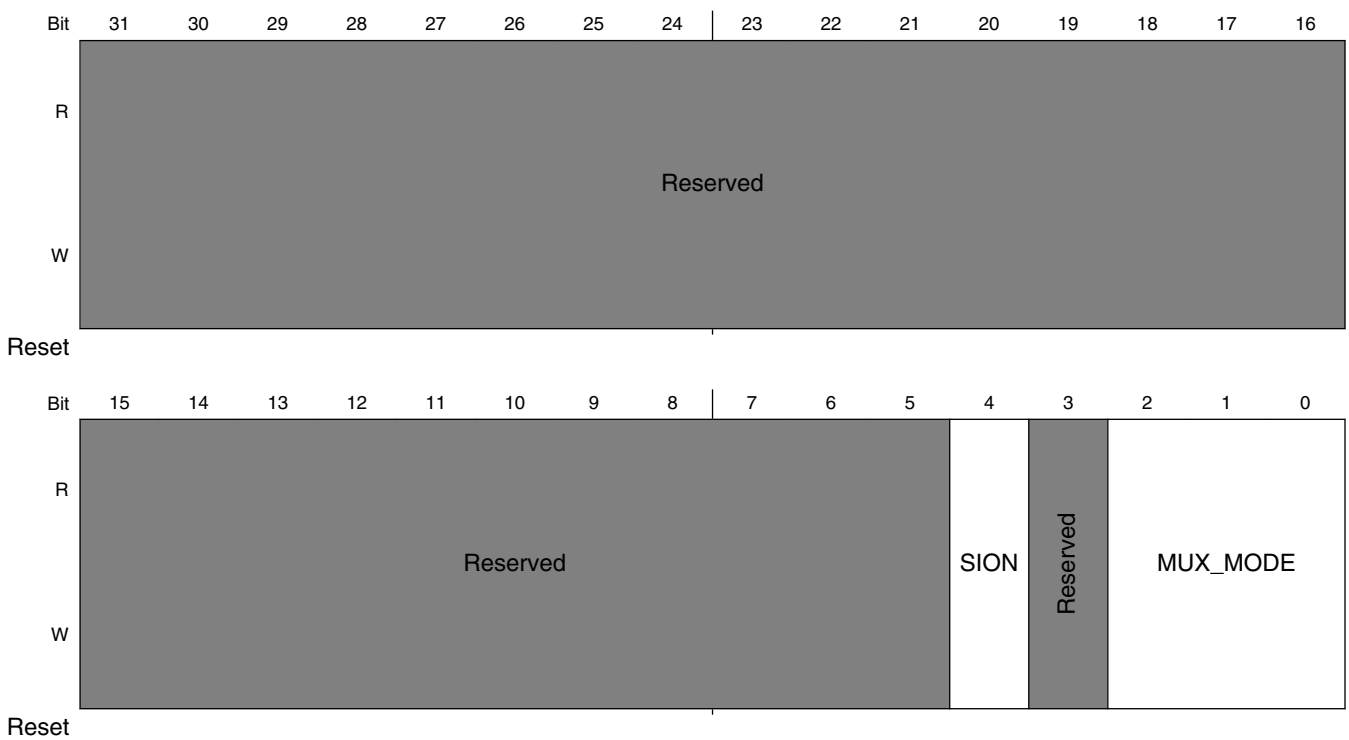
IOMUXC_SW_MUX_CTL_PAD_SAI3_TXC field descriptions (continued)

Field	Description
	010 **ALT2_AUDIOMIX_SAI5_RX_DATA[2]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI5_RX_DATA02 of instance: sai5 011 **ALT3_GPT1_CAPTURE1** — Select mux mode: ALT3 mux port: GPT1_CAPTURE1 of instance: gpt1 100 **ALT4_UART2_TX** — Select mux mode: ALT4 mux port: UART2_TX of instance: uart2 101 **ALT5_GPIO5_IO[0]** — Select mux mode: ALT5 mux port: GPIO5_IO00 of instance: gpio5 110 **ALT6_AUDIOMIX_PDM_BIT_STREAM[2]** — Select mux mode: ALT6 mux port: AUDIOMIX_PDM_BIT_STREAM02 of instance: pdm

8.2.4.111 SW_MUX_CTL_PAD_SAI3_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_TXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 1CCh offset = 3033_01CCh



IOMUXC_SW_MUX_CTL_PAD_SAI3_TXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field.

Table continues on the next page...

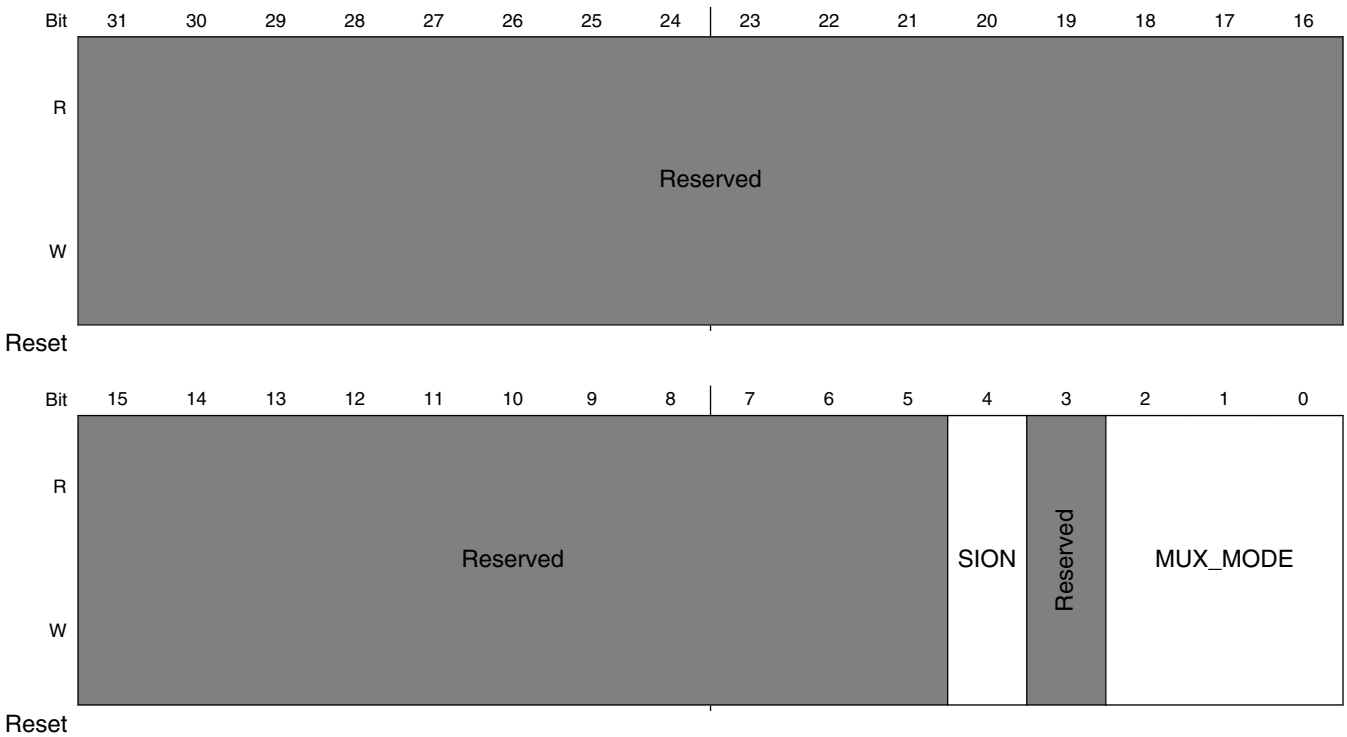
IOMUXC_SW_MUX_CTL_PAD_SAI3_TXD field descriptions (continued)

Field	Description
	Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI3_TXD 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SAI3_TXD. 000 **ALT0_AUDIOMIX_SAI3_TX_DATA[0]** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI3_TX_DATA00 of instance: sai3 001 **ALT1_AUDIOMIX_SAI2_TX_DATA[3]** — Select mux mode: ALT1 mux port: AUDIOMIX_SAI2_TX_DATA03 of instance: sai2 010 **ALT2_AUDIOMIX_SAI5_RX_DATA[3]** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI5_RX_DATA03 of instance: sai5 011 **ALT3_GPT1_CAPTURE2** — Select mux mode: ALT3 mux port: GPT1_CAPTURE2 of instance: gpt1 100 **ALT4_AUDIOMIX_SPDIF1_EXT_CLK** — Select mux mode: ALT4 mux port: AUDIOMIX_SPDIF1_EXT_CLK of instance: spdif 101 **ALT5_GPIO5_IO[1]** — Select mux mode: ALT5 mux port: GPIO5_IO01 of instance: gpio5

8.2.4.112 SW_MUX_CTL_PAD_SAI3_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_MCLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 1D0h offset = 3033_01D0h



IOMUXC_SW_MUX_CTL_PAD_SAI3_MCLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SAI3_MCLK 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: SAI3_MCLK. 000 **ALT0_AUDIOMIX_SAI3_MCLK** — Select mux mode: ALT0 mux port: AUDIOMIX_SAI3_MCLK of instance: sai3 001 **ALT1_PWM4_OUT** — Select mux mode: ALT1 mux port: PWM4_OUT of instance: pwm4

Table continues on the next page...

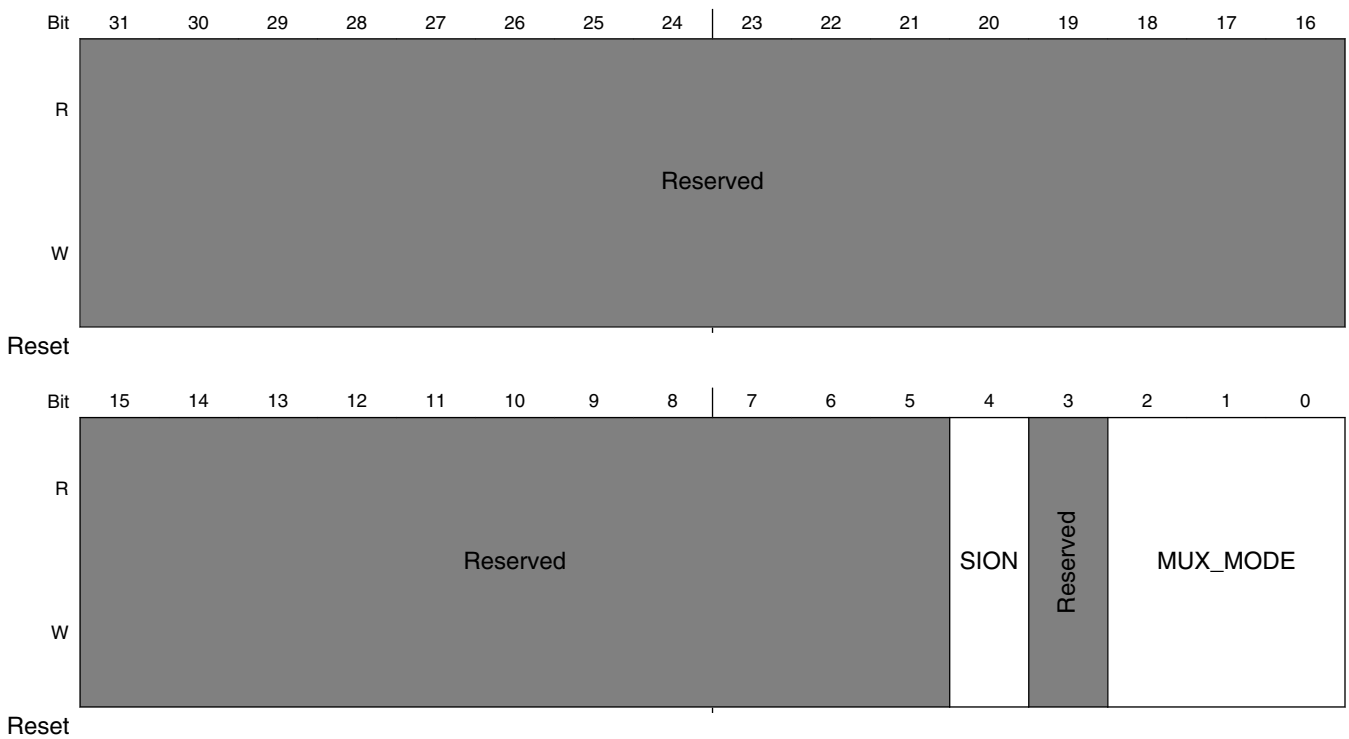
IOMUXC_SW_MUX_CTL_PAD_SAI3_MCLK field descriptions (continued)

Field	Description
	010 **ALT2_AUDIOMIX_SAI5_MCLK** — Select mux mode: ALT2 mux port: AUDIOMIX_SAI5_MCLK of instance: sai5 100 **ALT4_AUDIOMIX_SPDIF1_OUT** — Select mux mode: ALT4 mux port: AUDIOMIX_SPDIF1_OUT of instance: spdif 101 **ALT5_GPIO5_IO[2]** — Select mux mode: ALT5 mux port: GPIO5_IO02 of instance: gpio5 110 **ALT6_AUDIOMIX_SPDIF1_IN** — Select mux mode: ALT6 mux port: AUDIOMIX_SPDIF1_IN of instance: spdif

8.2.4.113 SW_MUX_CTL_PAD_SPDIF_TX SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SPDIF_TX)

SW_MUX_CTL Register

Address: 3033_0000h base + 1D4h offset = 3033_01D4h



IOMUXC_SW_MUX_CTL_PAD_SPDIF_TX field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality.

Table continues on the next page...

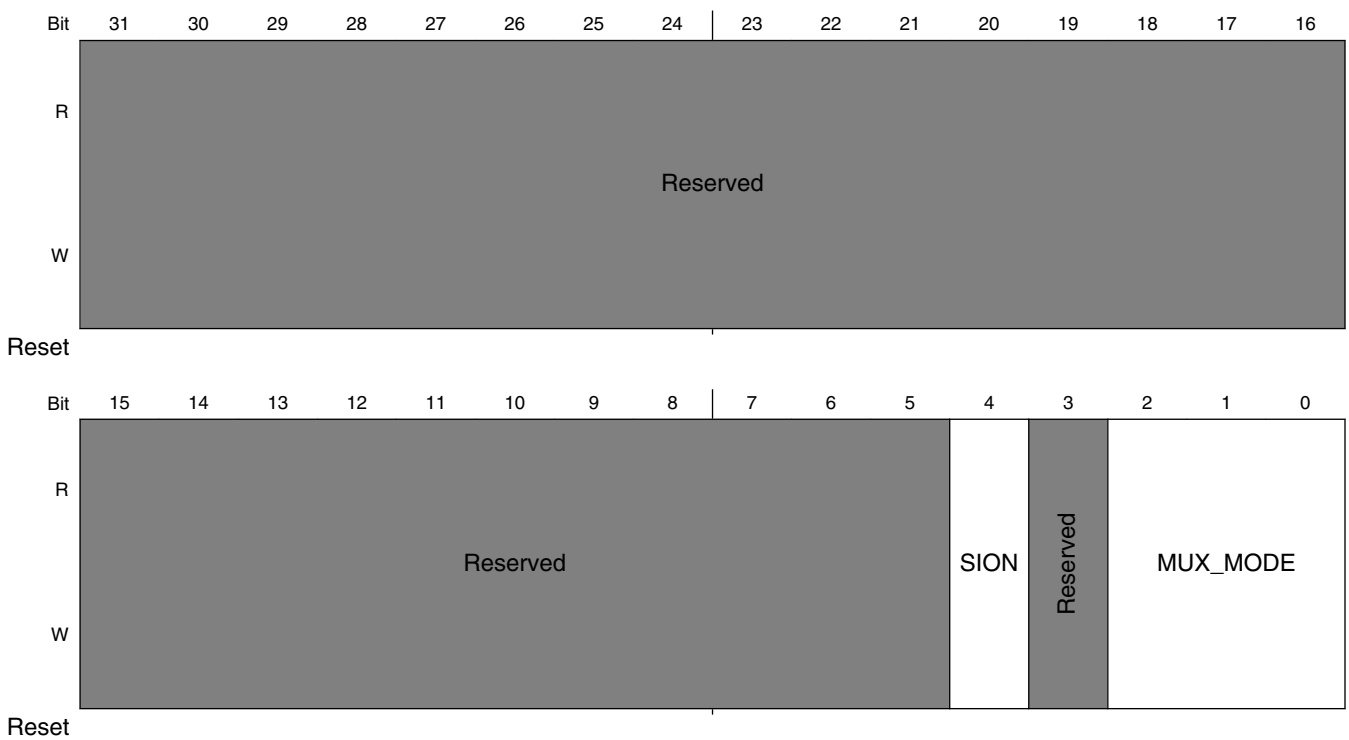
IOMUXC_SW_MUX_CTL_PAD_SPDIF_TX field descriptions (continued)

Field	Description
	1 **ENABLED** — Force input path of pad SPDIF_TX 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SPDIF_TX. 000 **ALT0_AUDIOMIX_SPDIF1_OUT** — Select mux mode: ALT0 mux port: AUDIOMIX_SPDIF1_OUT of instance: spdif 001 **ALT1_PWM3_OUT** — Select mux mode: ALT1 mux port: PWM3_OUT of instance: pwm3 010 **ALT2_I2C5_SCL** — Select mux mode: ALT2 mux port: I2C5_SCL of instance: i2c5 011 **ALT3_GPT1_COMPARE1** — Select mux mode: ALT3 mux port: GPT1_COMPARE1 of instance: gpt1 100 **ALT4_CAN1_TX** — Select mux mode: ALT4 mux port: CAN1_TX of instance: can1 101 **ALT5_GPIO5_IO[3]** — Select mux mode: ALT5 mux port: GPIO5_IO03 of instance: gpio5

8.2.4.114 SW_MUX_CTL_PAD_SPDIF_RX SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SPDIF_RX)

SW_MUX_CTL Register

Address: 3033_0000h base + 1D8h offset = 3033_01D8h



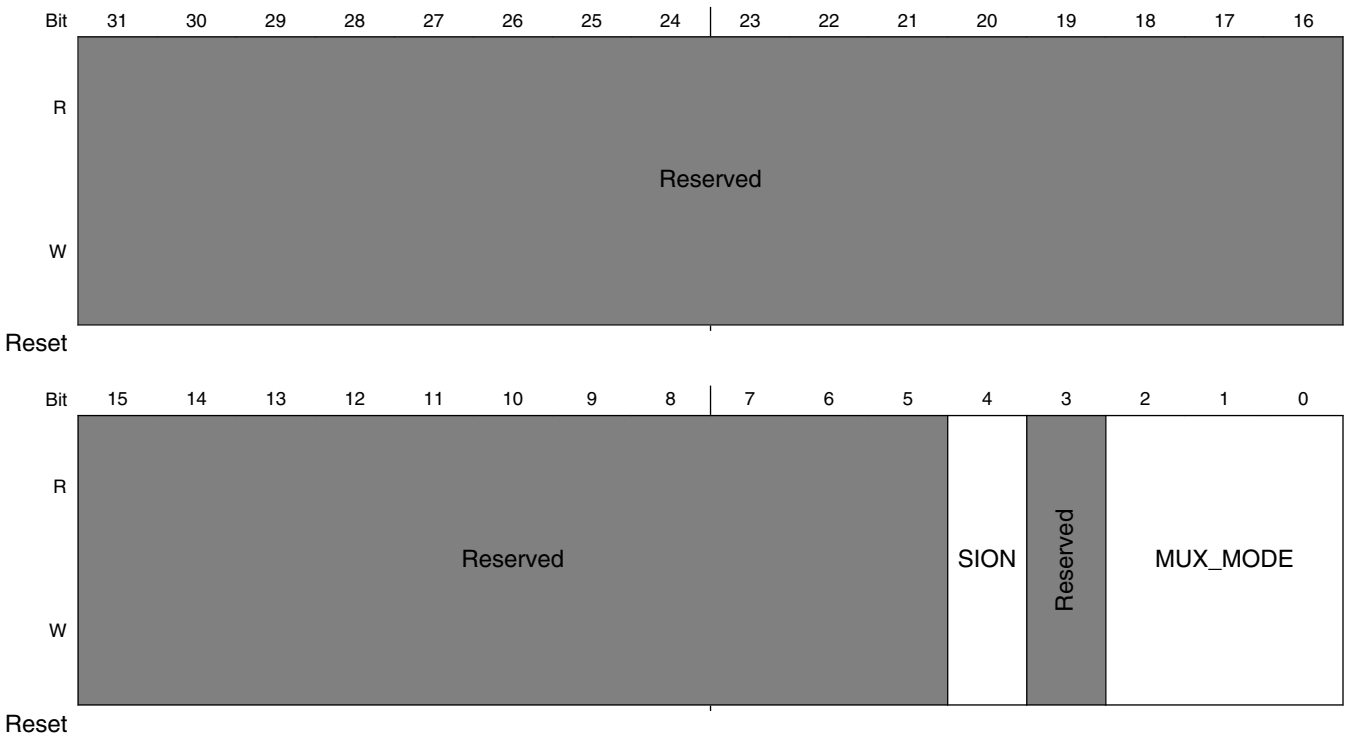
IOMUXC_SW_MUX_CTL_PAD_SPDIF_RX field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SPDIF_RX 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SPDIF_RX. 000 **ALT0_AUDIOMIX_SPDIF1_IN** — Select mux mode: ALT0 mux port: AUDIOMIX_SPDIF1_IN of instance: spdif 001 **ALT1_PWM2_OUT** — Select mux mode: ALT1 mux port: PWM2_OUT of instance: pwm2 010 **ALT2_I2C5_SDA** — Select mux mode: ALT2 mux port: I2C5_SDA of instance: i2c5 011 **ALT3_GPT1_COMPARE2** — Select mux mode: ALT3 mux port: GPT1_COMPARE2 of instance: gpt1 100 **ALT4_CAN1_RX** — Select mux mode: ALT4 mux port: CAN1_RX of instance: can1 101 **ALT5_GPIO5_IO[4]** — Select mux mode: ALT5 mux port: GPIO5_IO04 of instance: gpio5

8.2.4.115 SW_MUX_CTL_PAD_SPDIF_EXT_CLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SPDIF_EXT_CLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 1DCh offset = 3033_01DCh



IOMUXC_SW_MUX_CTL_PAD_SPDIF_EXT_CLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad SPDIF_EXT_CLK 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: SPDIF_EXT_CLK. 011 **ALT3_GPT1_COMPARE3** — Select mux mode: ALT3 mux port: GPT1_COMPARE3 of instance: gpt1 101 **ALT5_GPIO5_IO[5]** — Select mux mode: ALT5 mux port: GPIO5_IO05 of instance: gpio5

Table continues on the next page...

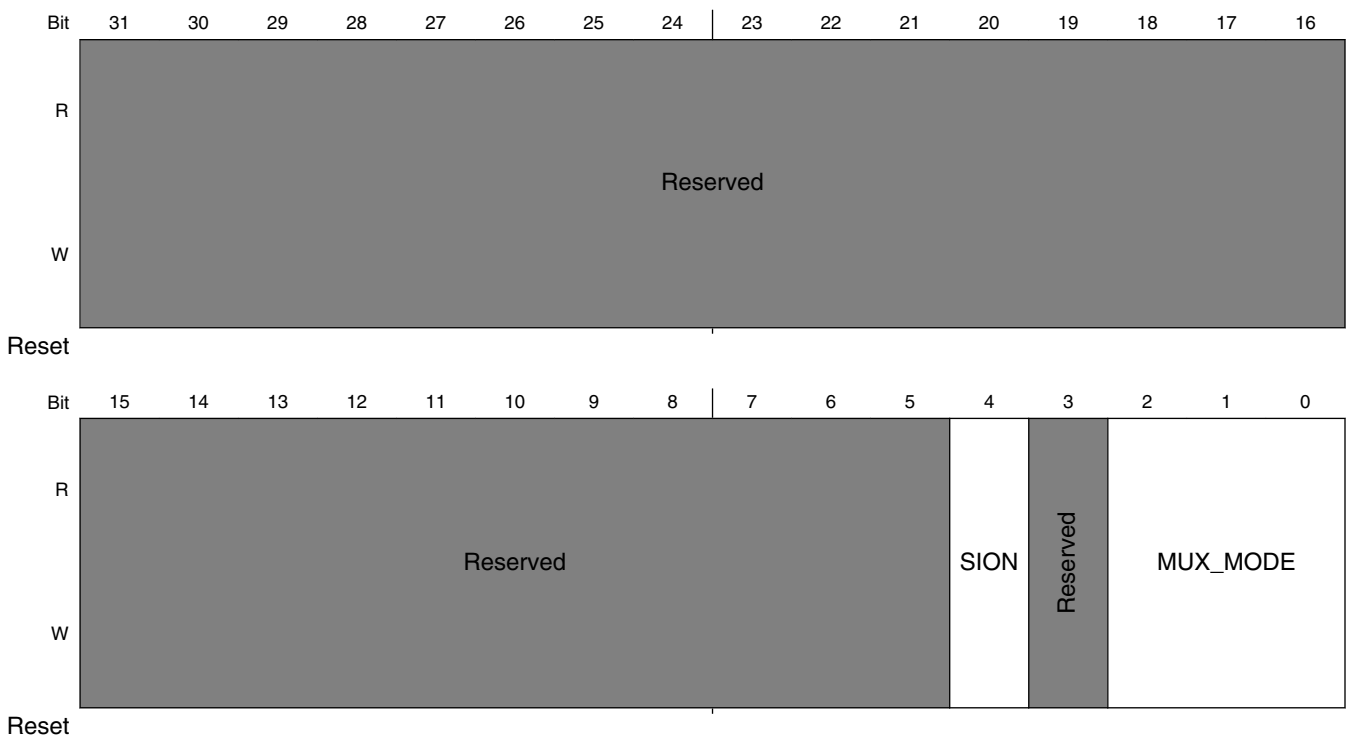
IOMUXC_SW_MUX_CTL_PAD_SPDIF_EXT_CLK field descriptions (continued)

Field	Description
	000 **ALT0_AUDIOMIX_SPDIF1_EXT_CLK** — Select mux mode: ALT0 mux port: AUDIOMIX_SPDIF1_EXT_CLK of instance: spdif 001 **ALT1_PWM1_OUT** — Select mux mode: ALT1 mux port: PWM1_OUT of instance: pwm1

8.2.4.116 SW_MUX_CTL_PAD_ECSPI1_SCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SCLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 1E0h offset = 3033_01E0h



IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SCLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ECSPI1_SCLK 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

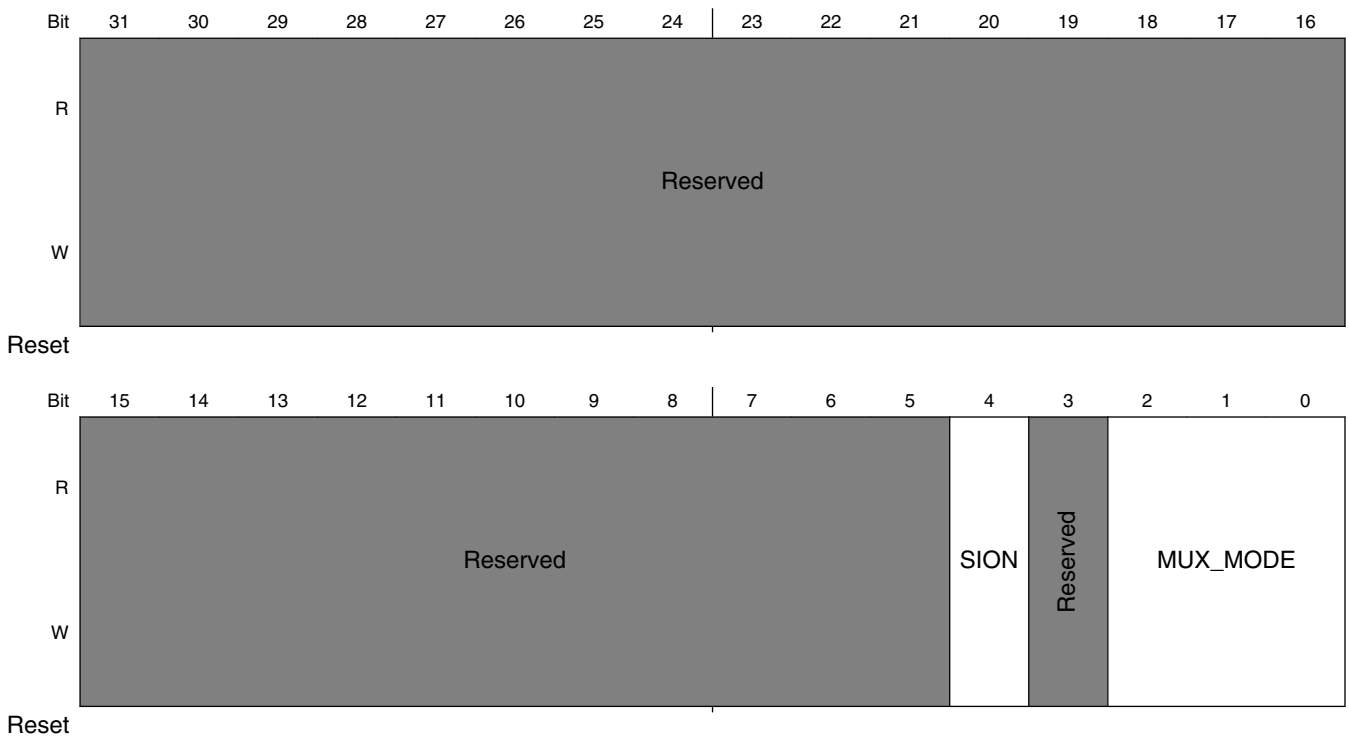
IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SCLK field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ECSPI1_SCLK. 000 **ALT0_ECSPI1_SCLK** — Select mux mode: ALT0 mux port: ECSPI1_SCLK of instance: ecspi1 001 **ALT1_UART3_RX** — Select mux mode: ALT1 mux port: UART3_RX of instance: uart3 010 **ALT2_I2C1_SCL** — Select mux mode: ALT2 mux port: I2C1_SCL of instance: i2c1 011 **ALT3_AUDIOMIX_SAI7_RX_SYNC** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI7_RX_SYNC of instance: sai7 101 **ALT5_GPIO5_IO[6]** — Select mux mode: ALT5 mux port: GPIO5_IO06 of instance: gpio5

8.2.4.117 SW_MUX_CTL_PAD_ECSPI1_MOSI SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MOSI)

SW_MUX_CTL Register

Address: 3033_0000h base + 1E4h offset = 3033_01E4h



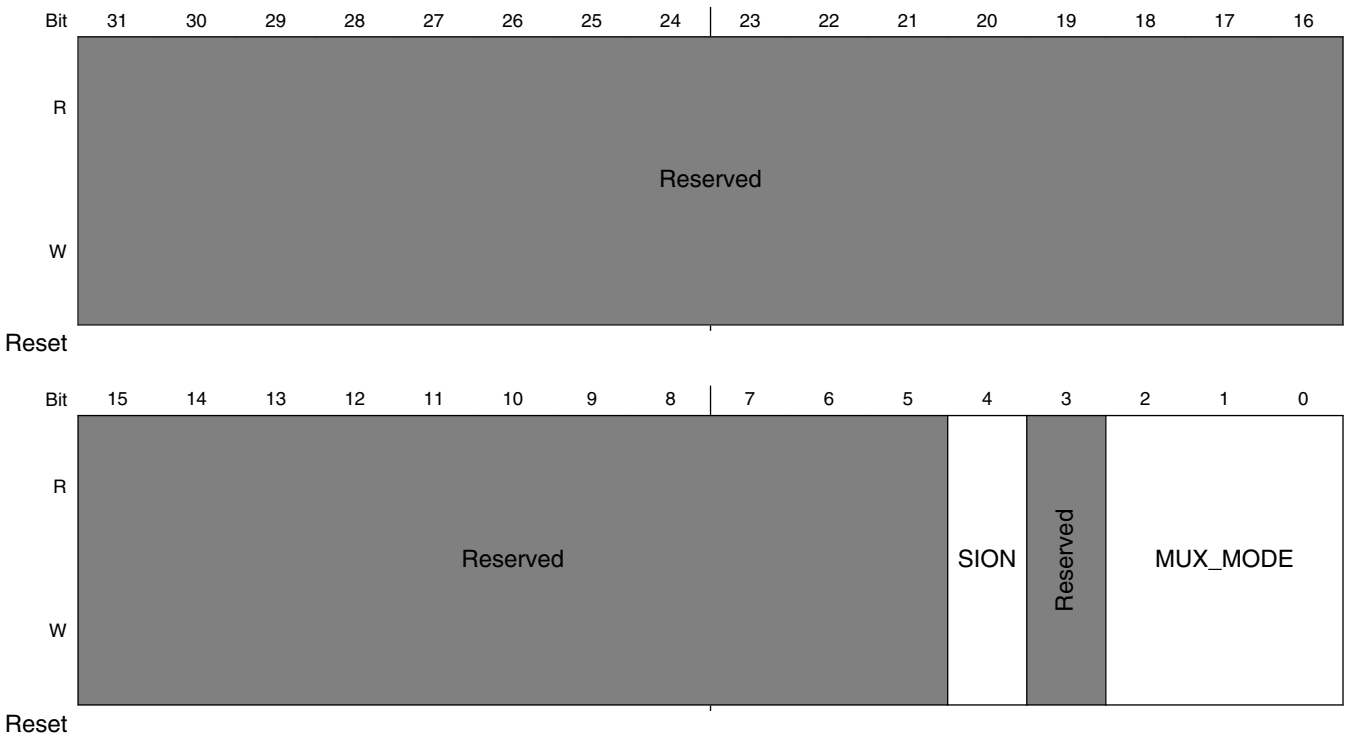
IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MOSI field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ECSPI1_MOSI 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ECSPI1_MOSI. 000 **ALT0_ECSPI1_MOSI** — Select mux mode: ALT0 mux port: ECSPI1_MOSI of instance: ecspi1 001 **ALT1_UART3_TX** — Select mux mode: ALT1 mux port: UART3_TX of instance: uart3 010 **ALT2_I2C1_SDA** — Select mux mode: ALT2 mux port: I2C1_SDA of instance: i2c1 011 **ALT3_AUDIOMIX_SAI7_RX_BCLK** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI7_RX_BCLK of instance: sai7 101 **ALT5_GPIO5_IO[7]** — Select mux mode: ALT5 mux port: GPIO5_IO07 of instance: gpio5

8.2.4.118 SW_MUX_CTL_PAD_ECSPI1_MISO SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MISO)

SW_MUX_CTL Register

Address: 3033_0000h base + 1E8h offset = 3033_01E8h



IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MISO field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ECSPI1_MISO 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ECSPI1_MISO. 000 **ALT0_ECSPI1_MISO** — Select mux mode: ALT0 mux port: ECSPI1_MISO of instance: ecspi1 001 **ALT1_UART3_CTS_B** — Select mux mode: ALT1 mux port: UART3_CTS_B of instance: uart3 010 **ALT2_I2C2_SCL** — Select mux mode: ALT2 mux port: I2C2_SCL of instance: i2c2

Table continues on the next page...

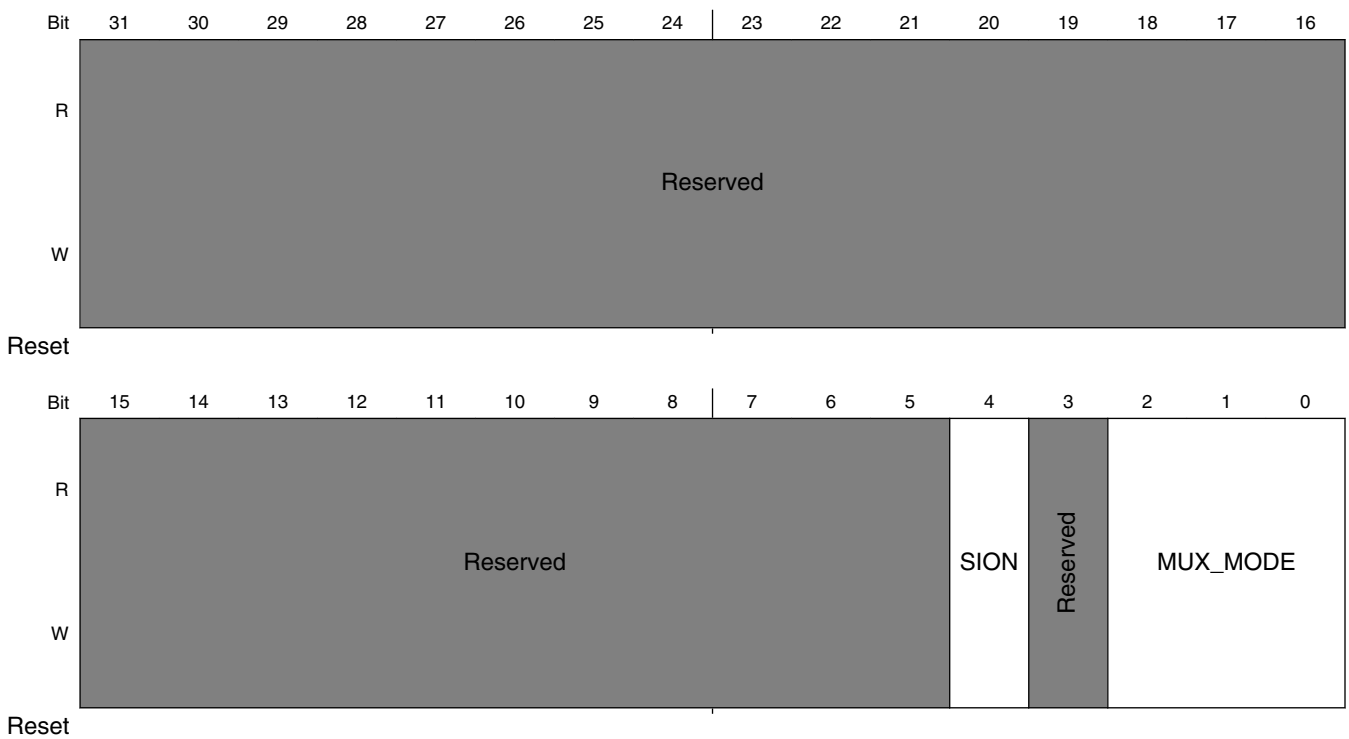
IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MISO field descriptions (continued)

Field	Description
	011 **ALT3_AUDIOMIX_SAI7_RX_DATA[0]** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI7_RX_DATA00 of instance: sai7 101 **ALT5_GPIO5_IO[8]** — Select mux mode: ALT5 mux port: GPIO5_IO08 of instance: gpio5

8.2.4.119 SW_MUX_CTL_PAD_ECSPI1_SS0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SS0)

SW_MUX_CTL Register

Address: 3033_0000h base + 1ECh offset = 3033_01ECh



IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SS0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ECSPI1_SS0 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

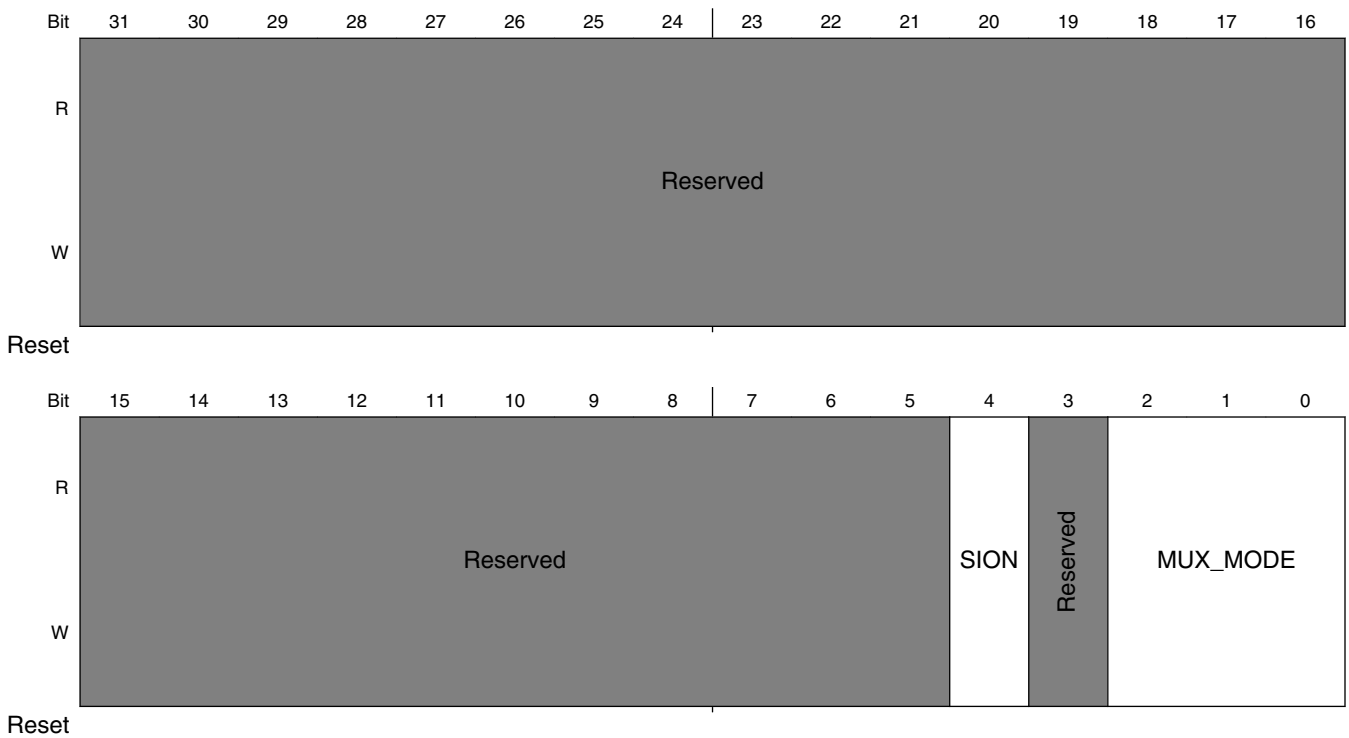
IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SS0 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ECSPI1_SS0. 000 **ALT0_ECSPI1_SS0** — Select mux mode: ALT0 mux port: ECSPI1_SS0 of instance: ecspi1 001 **ALT1_UART3_RTS_B** — Select mux mode: ALT1 mux port: UART3_RTS_B of instance: uart3 010 **ALT2_I2C2_SDA** — Select mux mode: ALT2 mux port: I2C2_SDA of instance: i2c2 011 **ALT3_AUDIOMIX_SAI7_TX_SYNC** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI7_TX_SYNC of instance: sai7 101 **ALT5_GPIO5_IO[9]** — Select mux mode: ALT5 mux port: GPIO5_IO09 of instance: gpio5

8.2.4.120 SW_MUX_CTL_PAD_ECSPI2_SCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_SCLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 1F0h offset = 3033_01F0h



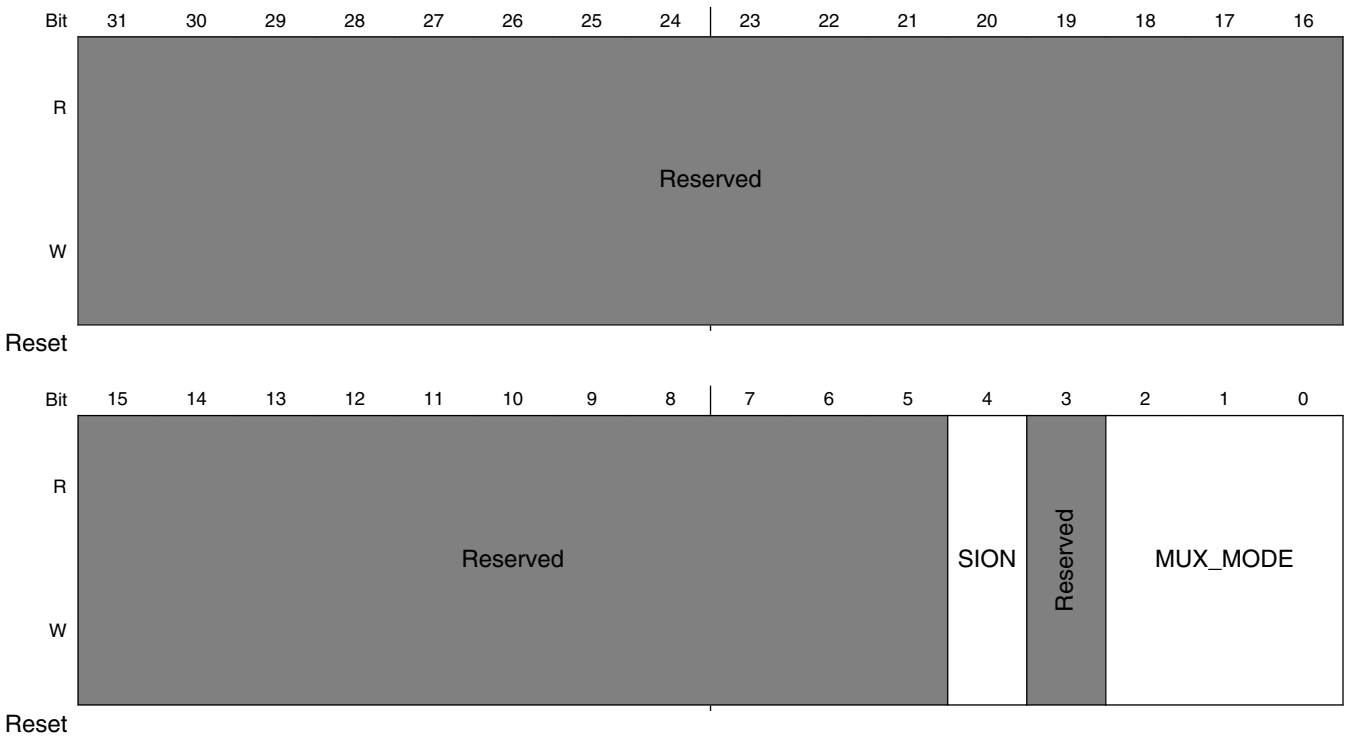
IOMUXC_SW_MUX_CTL_PAD_ECSPI2_SCLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ECSPI2_SCLK 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ECSPI2_SCLK. 000 **ALT0_ECSPI2_SCLK** — Select mux mode: ALT0 mux port: ECSPI2_SCLK of instance: ecspi2 001 **ALT1_UART4_RX** — Select mux mode: ALT1 mux port: UART4_RX of instance: uart4 010 **ALT2_I2C3_SCL** — Select mux mode: ALT2 mux port: I2C3_SCL of instance: i2c3 011 **ALT3_AUDIOMIX_SAI7_TX_BCLK** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI7_TX_BCLK of instance: sai7 101 **ALT5_GPIO5_IO[10]** — Select mux mode: ALT5 mux port: GPIO5_IO10 of instance: gpio5

8.2.4.121 SW_MUX_CTL_PAD_ECSPI2_MOSI SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MOSI)

SW_MUX_CTL Register

Address: 3033_0000h base + 1F4h offset = 3033_01F4h



IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MOSI field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ECSPI2_MOSI 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ECSPI2_MOSI. 000 **ALT0_ECSPI2_MOSI** — Select mux mode: ALT0 mux port: ECSPI2_MOSI of instance: ecspi2 001 **ALT1_UART4_TX** — Select mux mode: ALT1 mux port: UART4_TX of instance: uart4 010 **ALT2_I2C3_SDA** — Select mux mode: ALT2 mux port: I2C3_SDA of instance: i2c3

Table continues on the next page...

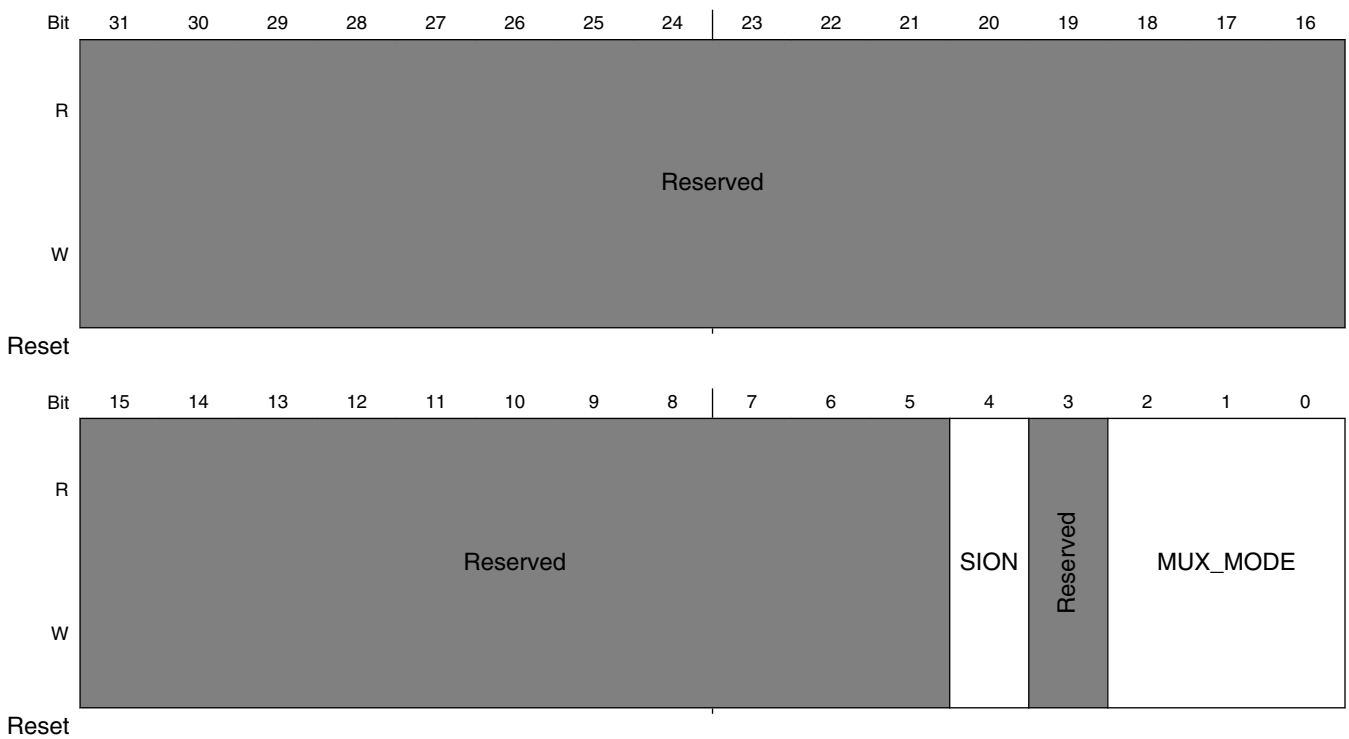
IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MOSI field descriptions (continued)

Field	Description
	011 **ALT3_AUDIOMIX_SAI7_TX_DATA[0]** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI7_TX_DATA00 of instance: sai7 101 **ALT5_GPIO5_IO[11]** — Select mux mode: ALT5 mux port: GPIO5_IO11 of instance: gpio5

8.2.4.122 SW_MUX_CTL_PAD_ECSPI2_MISO SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MISO)

SW_MUX_CTL Register

Address: 3033_0000h base + 1F8h offset = 3033_01F8h



IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MISO field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ECSPI2_MISO 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

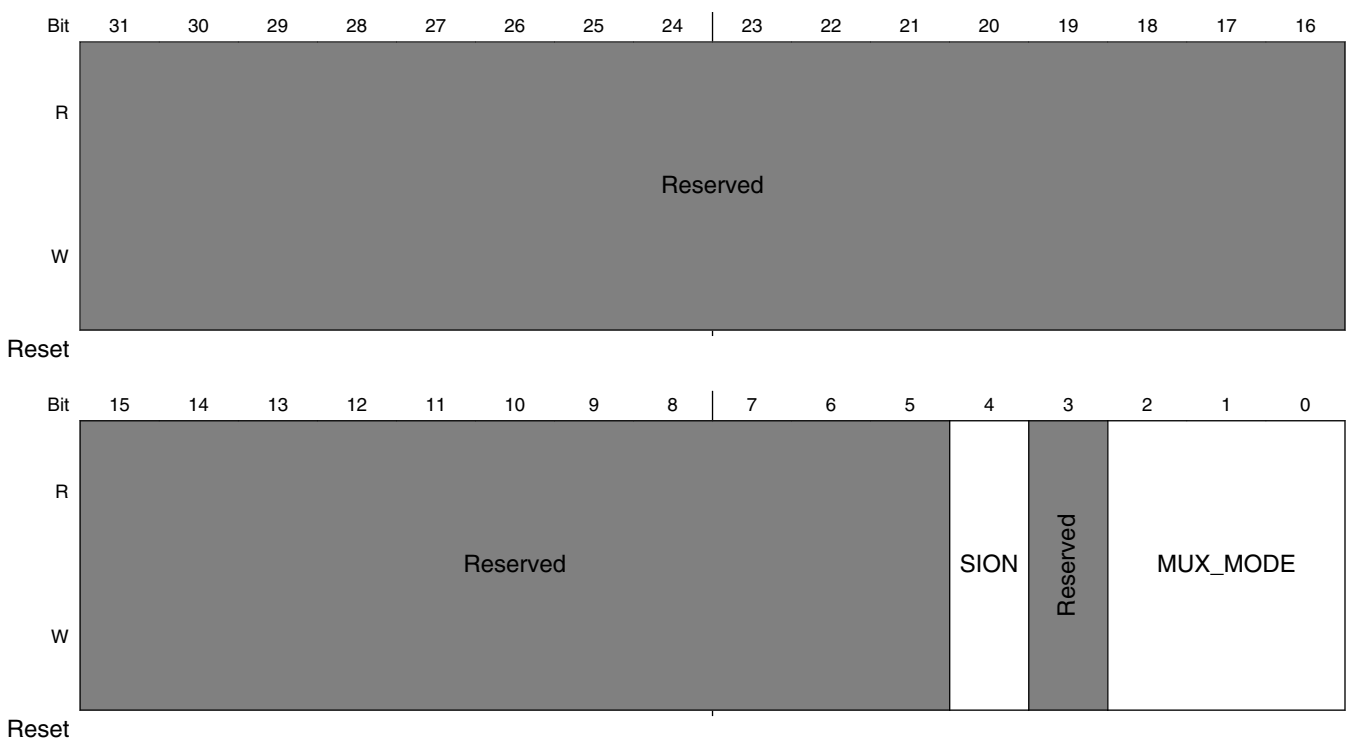
IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MISO field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: ECSPI2_MISO. 101 **ALT5_GPIO5_IO[12]** — Select mux mode: ALT5 mux port: GPIO5_IO12 of instance: gpio5 000 **ALT0_ECSPI2_MISO** — Select mux mode: ALT0 mux port: ECSPI2_MISO of instance: ecspi2 001 **ALT1_UART4_CTS_B** — Select mux mode: ALT1 mux port: UART4_CTS_B of instance: uart4 010 **ALT2_I2C4_SCL** — Select mux mode: ALT2 mux port: I2C4_SCL of instance: i2c4 011 **ALT3_AUDIOMIX_SAI7_MCLK** — Select mux mode: ALT3 mux port: AUDIOMIX_SAI7_MCLK of instance: sai7 100 **ALT4_CCM_CLKO1** — Select mux mode: ALT4 mux port: CCM_CLKO1 of instance: ccm

8.2.4.123 SW_MUX_CTL_PAD_ECSPI2_SS0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_SS0)

SW_MUX_CTL Register

Address: 3033_0000h base + 1FCh offset = 3033_01FCh



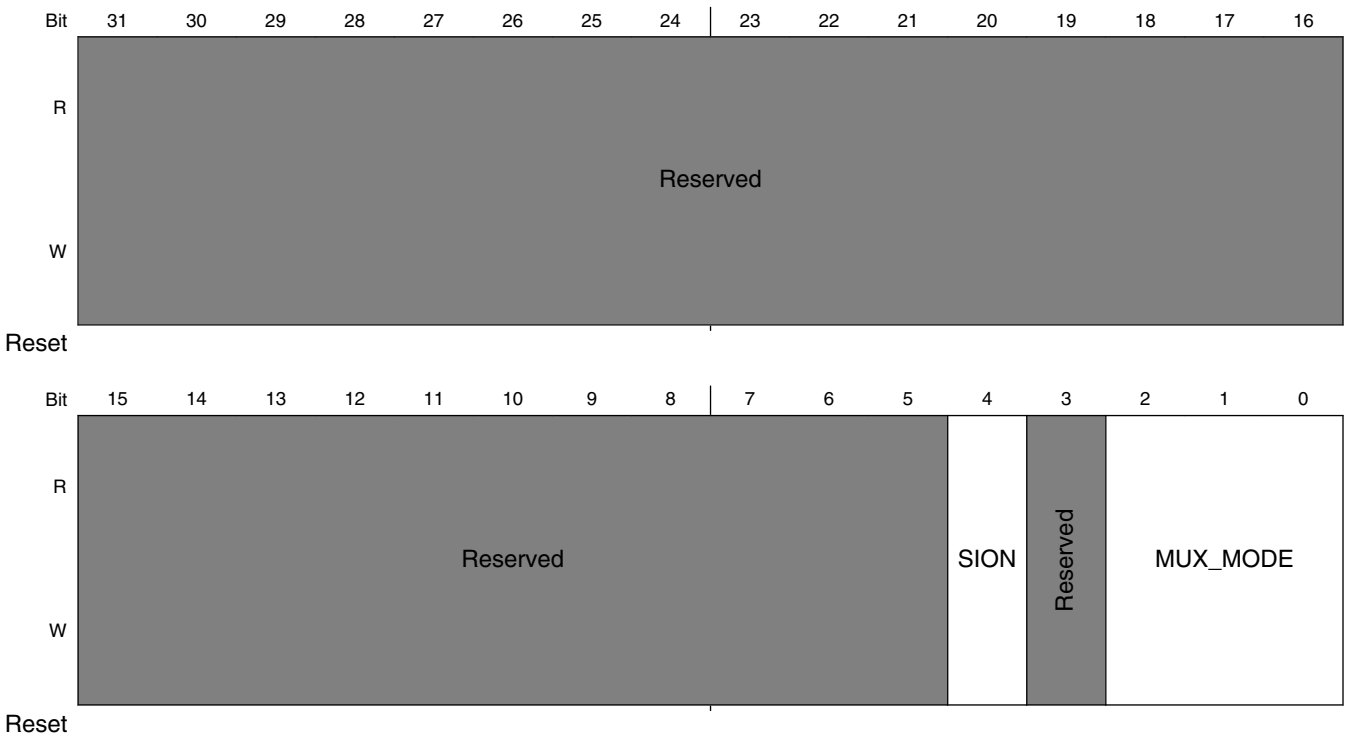
IOMUXC_SW_MUX_CTL_PAD_ECSPI2_SS0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad ECSPI2_SS0 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: ECSPI2_SS0. 000 **ALT0_ECSPI2_SS0** — Select mux mode: ALT0 mux port: ECSPI2_SS0 of instance: ecspi2 001 **ALT1_UART4_RTS_B** — Select mux mode: ALT1 mux port: UART4_RTS_B of instance: uart4 010 **ALT2_I2C4_SDA** — Select mux mode: ALT2 mux port: I2C4_SDA of instance: i2c4 100 **ALT4_CCM_CLKO2** — Select mux mode: ALT4 mux port: CCM_CLKO2 of instance: ccm 101 **ALT5_GPIO5_IO[13]** — Select mux mode: ALT5 mux port: GPIO5_IO13 of instance: gpio5

8.2.4.124 SW_MUX_CTL_PAD_I2C1_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C1_SCL)

SW_MUX_CTL Register

Address: 3033_0000h base + 200h offset = 3033_0200h



IOMUXC_SW_MUX_CTL_PAD_I2C1_SCL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad I2C1_SCL 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: I2C1_SCL. 000 **ALT0_I2C1_SCL** — Select mux mode: ALT0 mux port: I2C1_SCL of instance: i2c1 001 **ALT1_ENET_QOS_MDC** — Select mux mode: ALT1 mux port: ENET_QOS_MDC of instance: enet_qos

Table continues on the next page...

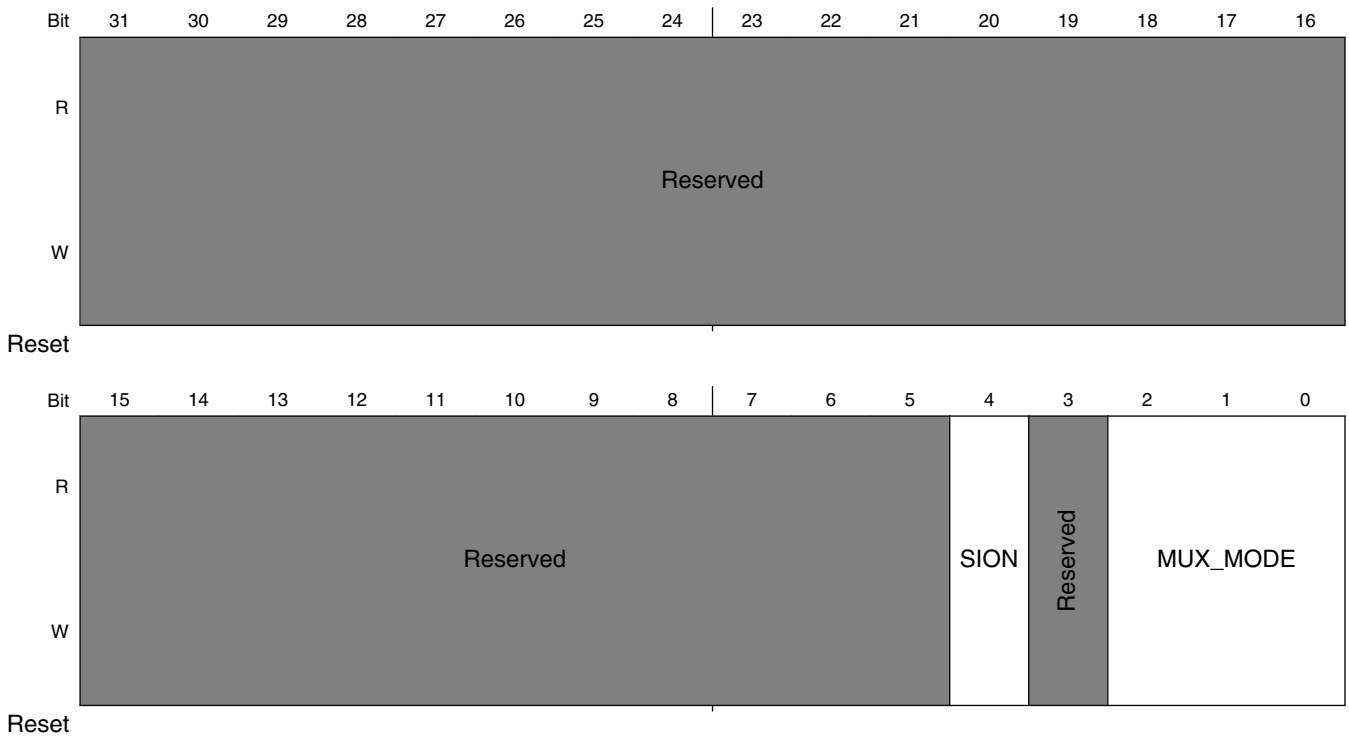
IOMUXC_SW_MUX_CTL_PAD_I2C1_SCL field descriptions (continued)

Field	Description
	011 **ALT3_ECSPI1_SCLK** — Select mux mode: ALT3 mux port: ECSPI1_SCLK of instance: ecspi1 101 **ALT5_GPIO5_IO[14]** — Select mux mode: ALT5 mux port: GPIO5_IO14 of instance: gpio5

8.2.4.125 SW_MUX_CTL_PAD_I2C1_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C1_SDA)

SW_MUX_CTL Register

Address: 3033_0000h base + 204h offset = 3033_0204h



IOMUXC_SW_MUX_CTL_PAD_I2C1_SDA field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad I2C1_SDA 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved

Table continues on the next page...

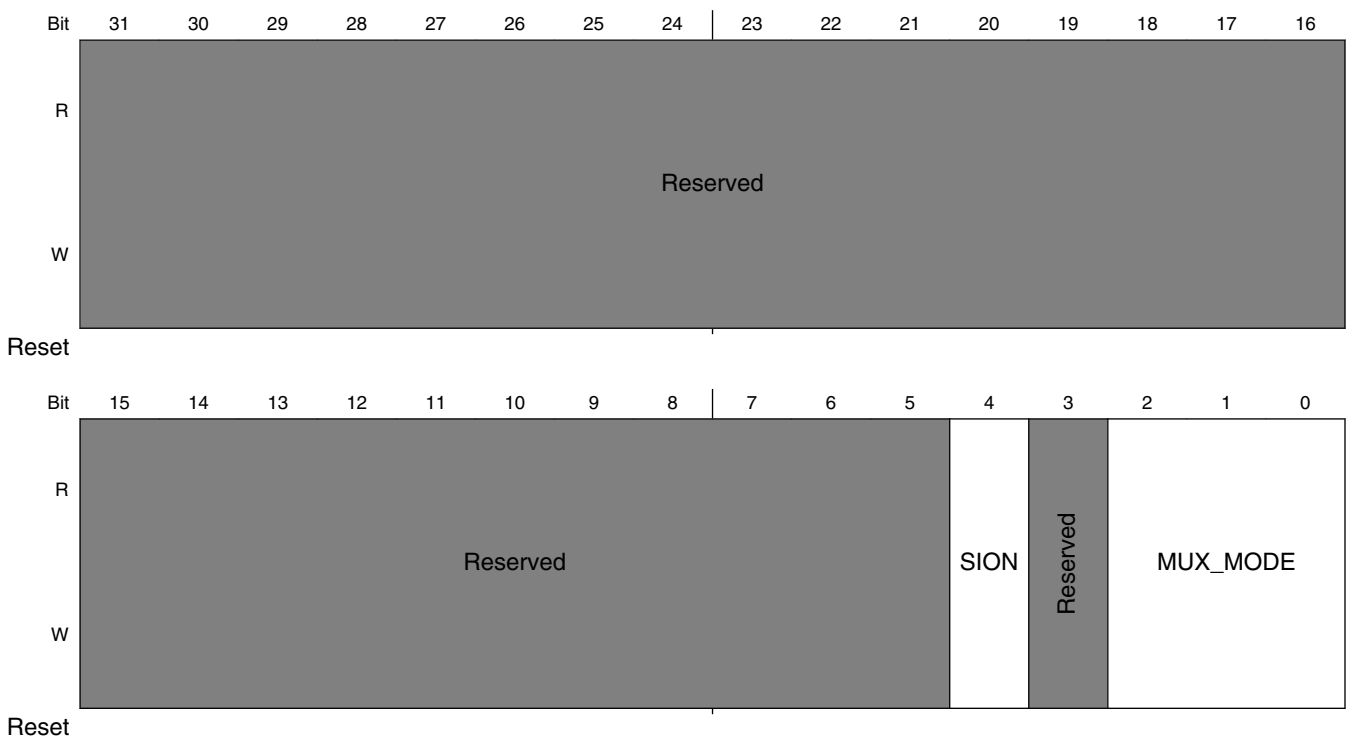
IOMUXC_SW_MUX_CTL_PAD_I2C1_SDA field descriptions (continued)

Field	Description
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: I2C1_SDA. 000 **ALT0_I2C1_SDA** — Select mux mode: ALT0 mux port: I2C1_SDA of instance: i2c1 001 **ALT1_ENET_QOS_MDIO** — Select mux mode: ALT1 mux port: ENET_QOS_MDIO of instance: enet_qos 011 **ALT3_ECSPI1_MOSI** — Select mux mode: ALT3 mux port: ECSPI1_MOSI of instance: ecspi1 101 **ALT5_GPIO5_IO[15]** — Select mux mode: ALT5 mux port: GPIO5_IO15 of instance: gpio5

8.2.4.126 SW_MUX_CTL_PAD_I2C2_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C2_SCL)

SW_MUX_CTL Register

Address: 3033_0000h base + 208h offset = 3033_0208h



IOMUXC_SW_MUX_CTL_PAD_I2C2_SCL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field.

Table continues on the next page...

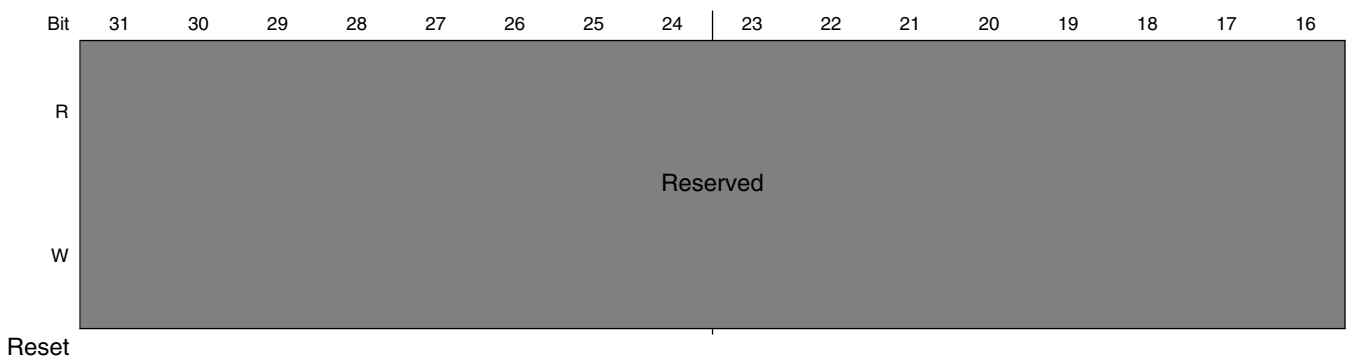
IOMUXC_SW_MUX_CTL_PAD_I2C2_SCL field descriptions (continued)

Field	Description
	Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad I2C2_SCL 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: I2C2_SCL. 000 **ALT0_I2C2_SCL** — Select mux mode: ALT0 mux port: I2C2_SCL of instance: i2c2 001 **ALT1_ENET_QOS_1588_EVENT1_IN** — Select mux mode: ALT1 mux port: ENET_QOS_1588_EVENT1_IN of instance: enet_qos 010 **ALT2_USDHC3_CD_B** — Select mux mode: ALT2 mux port: USDHC3_CD_B of instance: usdhc3 011 **ALT3_ECSPI1_MISO** — Select mux mode: ALT3 mux port: ECSPI1_MISO of instance: ecspi1 100 **ALT4_ENET_QOS_1588_EVENT1_AUX_IN** — Select mux mode: ALT4 mux port: ENET_QOS_1588_EVENT1_AUX_IN of instance: enet_qos 101 **ALT5_GPIO5_IO[16]** — Select mux mode: ALT5 mux port: GPIO5_IO16 of instance: gpio5

8.2.4.127 SW_MUX_CTL_PAD_I2C2_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C2_SDA)

SW_MUX_CTL Register

Address: 3033_0000h base + 20Ch offset = 3033_020Ch



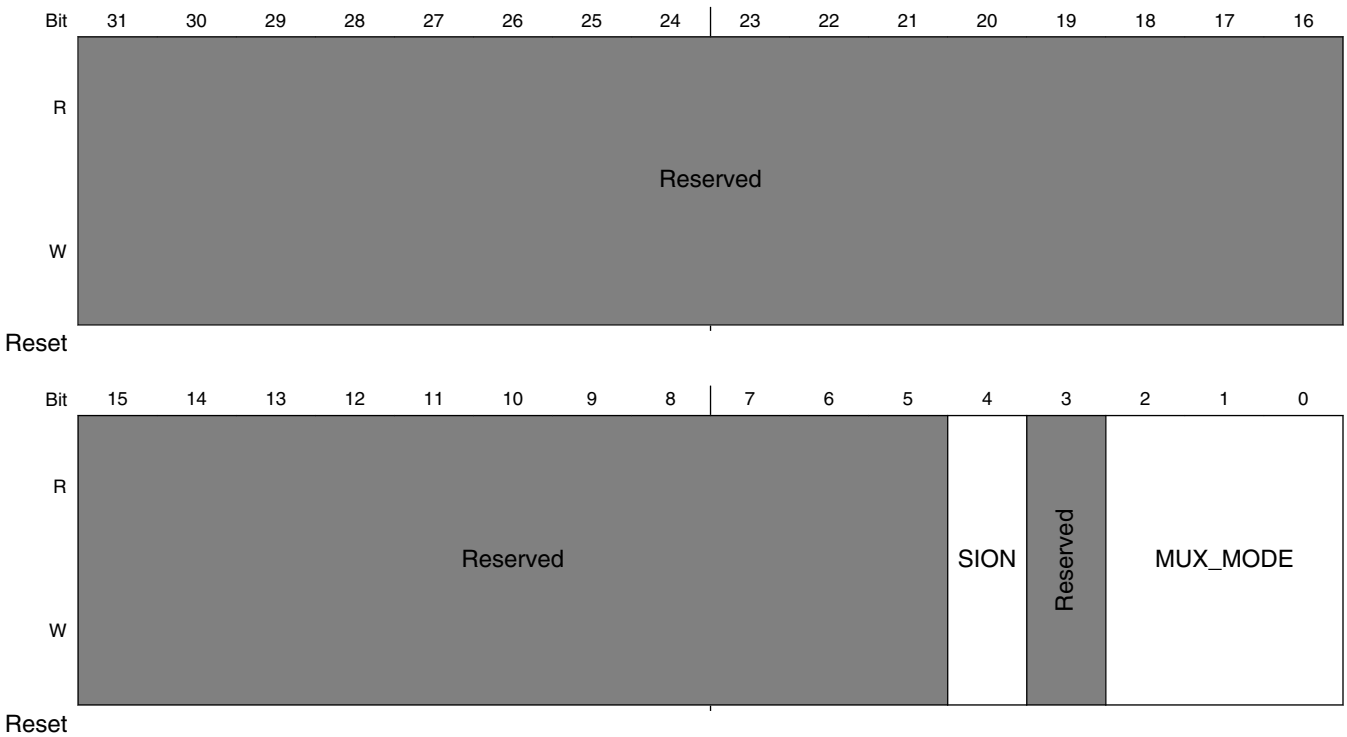
IOMUXC_SW_MUX_CTL_PAD_I2C2_SDA field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad I2C2_SDA 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: I2C2_SDA. 000 **ALT0_I2C2_SDA** — Select mux mode: ALT0 mux port: I2C2_SDA of instance: i2c2 001 **ALT1_ENET_QOS_1588_EVENT1_OUT** — Select mux mode: ALT1 mux port: ENET_QOS_1588_EVENT1_OUT of instance: enet_qos 010 **ALT2_USDHC3_WP** — Select mux mode: ALT2 mux port: USDHC3_WP of instance: usdhc3 011 **ALT3_ECSPI1_SS0** — Select mux mode: ALT3 mux port: ECSPI1_SS0 of instance: ecspi1 101 **ALT5_GPIO5_IO[17]** — Select mux mode: ALT5 mux port: GPIO5_IO17 of instance: gpio5

8.2.4.128 SW_MUX_CTL_PAD_I2C3_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C3_SCL)

SW_MUX_CTL Register

Address: 3033_0000h base + 210h offset = 3033_0210h



IOMUXC_SW_MUX_CTL_PAD_I2C3_SCL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad I2C3_SCL 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: I2C3_SCL. 000 **ALT0_I2C3_SCL** — Select mux mode: ALT0 mux port: I2C3_SCL of instance: i2c3 001 **ALT1_PWM4_OUT** — Select mux mode: ALT1 mux port: PWM4_OUT of instance: pwm4 010 **ALT2_GPT2_CLK** — Select mux mode: ALT2 mux port: GPT2_CLK of instance: gpt2

Table continues on the next page...

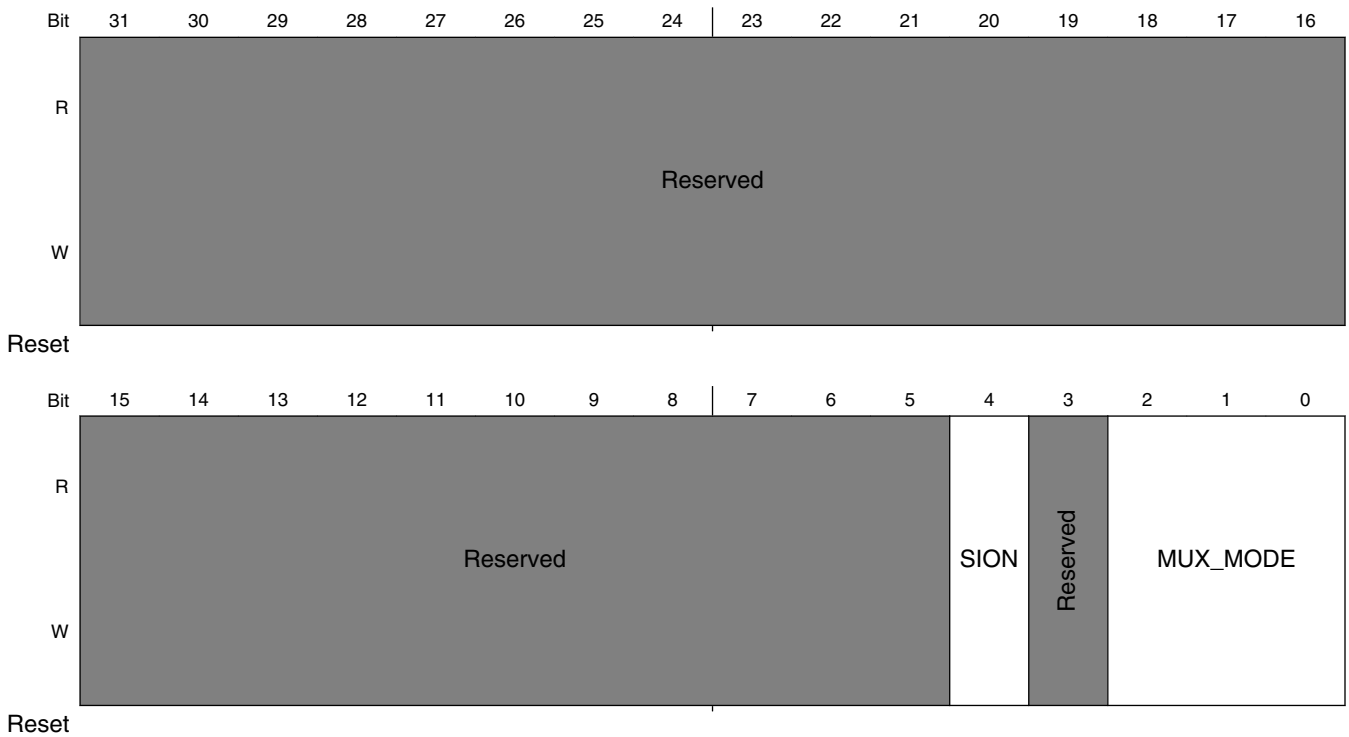
IOMUXC_SW_MUX_CTL_PAD_I2C3_SCL field descriptions (continued)

Field	Description
	011 **ALT3_ECSPI2_SCLK** — Select mux mode: ALT3 mux port: ECSPI2_SCLK of instance: ecspi2 101 **ALT5_GPIO5_IO[18]** — Select mux mode: ALT5 mux port: GPIO5_IO18 of instance: gpio5

8.2.4.129 SW_MUX_CTL_PAD_I2C3_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C3_SDA)

SW_MUX_CTL Register

Address: 3033_0000h base + 214h offset = 3033_0214h



IOMUXC_SW_MUX_CTL_PAD_I2C3_SDA field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad I2C3_SDA 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved

Table continues on the next page...

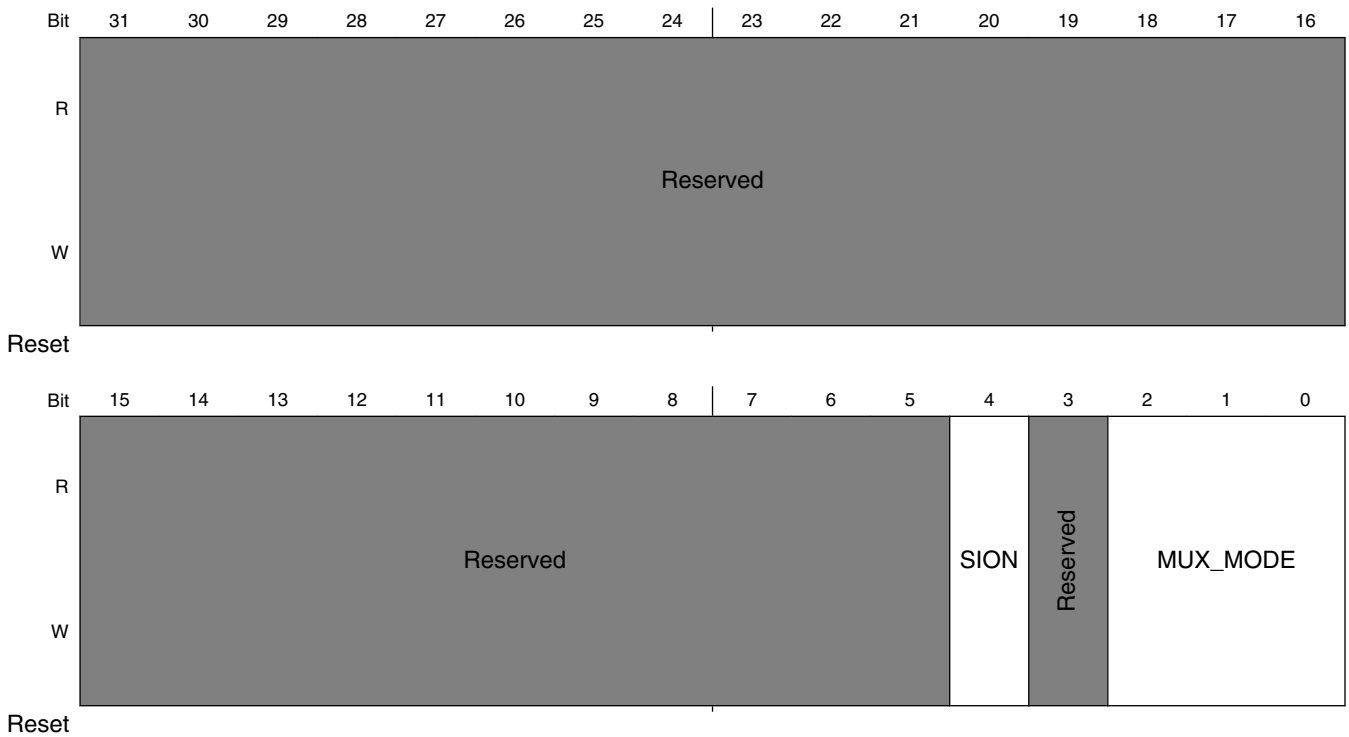
IOMUXC_SW_MUX_CTL_PAD_I2C3_SDA field descriptions (continued)

Field	Description
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: I2C3_SDA. 000 **ALT0_I2C3_SDA** — Select mux mode: ALT0 mux port: I2C3_SDA of instance: i2c3 001 **ALT1_PWM3_OUT** — Select mux mode: ALT1 mux port: PWM3_OUT of instance: pwm3 010 **ALT2_GPT3_CLK** — Select mux mode: ALT2 mux port: GPT3_CLK of instance: gpt3 011 **ALT3_ECSPI2_MOSI** — Select mux mode: ALT3 mux port: ECSPI2_MOSI of instance: ecspi2 101 **ALT5_GPIO5_IO[19]** — Select mux mode: ALT5 mux port: GPIO5_IO19 of instance: gpio5

8.2.4.130 SW_MUX_CTL_PAD_I2C4_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C4_SCL)

SW_MUX_CTL Register

Address: 3033_0000h base + 218h offset = 3033_0218h



IOMUXC_SW_MUX_CTL_PAD_I2C4_SCL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved

Table continues on the next page...

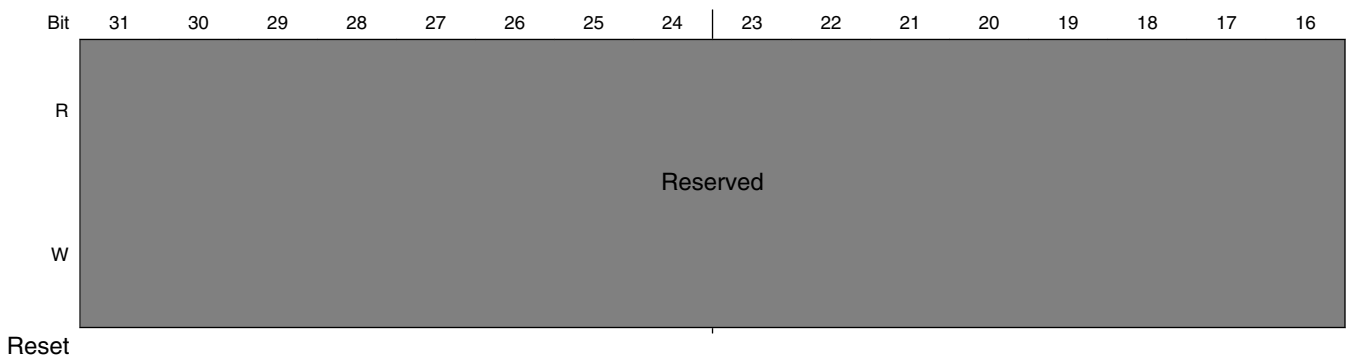
IOMUXC_SW_MUX_CTL_PAD_I2C4_SCL field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad I2C4_SCL 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: I2C4_SCL. 000 **ALT0_I2C4_SCL** — Select mux mode: ALT0 mux port: I2C4_SCL of instance: i2c4 001 **ALT1_PWM2_OUT** — Select mux mode: ALT1 mux port: PWM2_OUT of instance: pwm2 010 **ALT2_PCIE_CLKREQ_B** — Select mux mode: ALT2 mux port: PCIE_CLKREQ_B of instance: pcie 011 **ALT3_ECSPI2_MISO** — Select mux mode: ALT3 mux port: ECSPI2_MISO of instance: ecspi2 101 **ALT5_GPIO5_IO[20]** — Select mux mode: ALT5 mux port: GPIO5_IO20 of instance: gpio5

8.2.4.131 SW_MUX_CTL_PAD_I2C4_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C4_SDA)

SW_MUX_CTL Register

Address: 3033_0000h base + 21Ch offset = 3033_021Ch



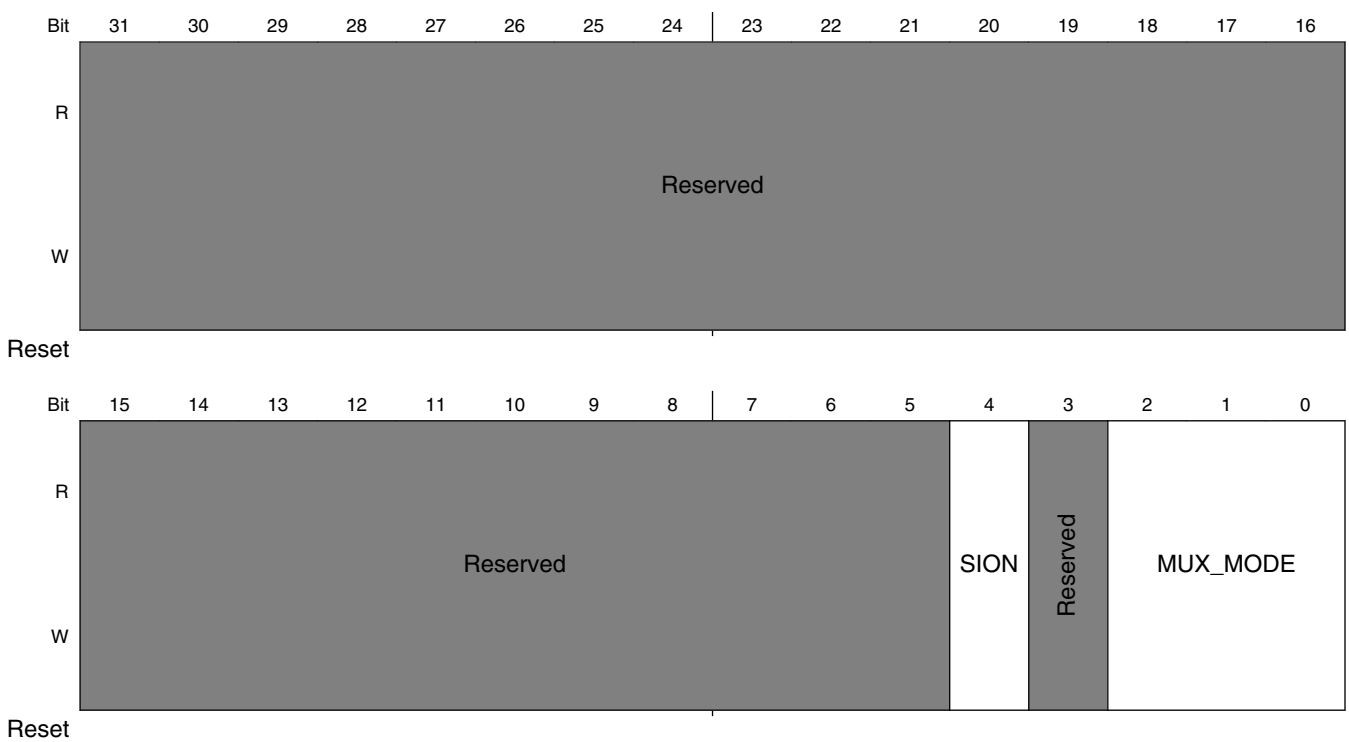
IOMUXC_SW_MUX_CTL_PAD_I2C4_SDA field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad I2C4_SDA 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: I2C4_SDA. 000 **ALT0_I2C4_SDA** — Select mux mode: ALT0 mux port: I2C4_SDA of instance: i2c4 001 **ALT1_PWM1_OUT** — Select mux mode: ALT1 mux port: PWM1_OUT of instance: pwm1 011 **ALT3_ECSPI2_SS0** — Select mux mode: ALT3 mux port: ECSPI2_SS0 of instance: ecspi2 101 **ALT5_GPIO5_IO[21]** — Select mux mode: ALT5 mux port: GPIO5_IO21 of instance: gpio5

8.2.4.132 SW_MUX_CTL_PAD_UART1_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART1_RXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 220h offset = 3033_0220h



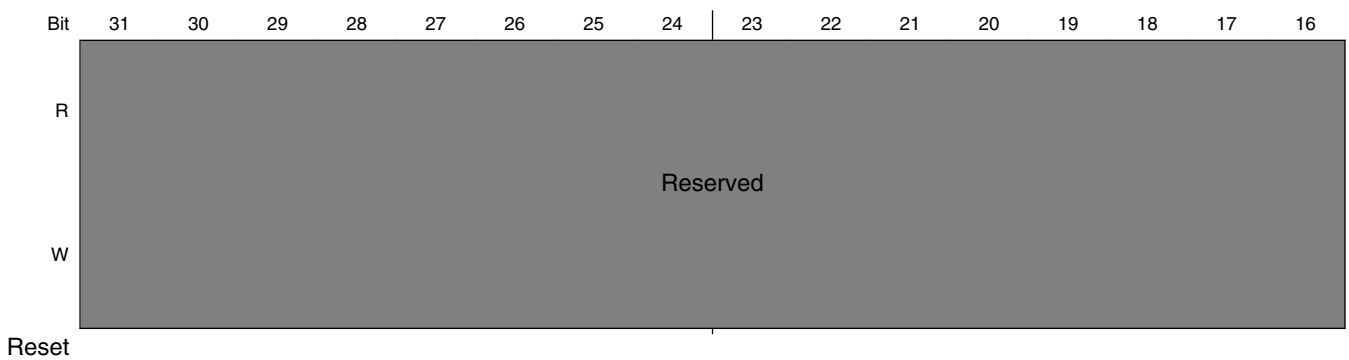
IOMUXC_SW_MUX_CTL_PAD_UART1_RXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad UART1_RXD 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: UART1_RXD. 000 **ALT0_UART1_RX** — Select mux mode: ALT0 mux port: UART1_RX of instance: uart1 001 **ALT1_ECSPI3_SCLK** — Select mux mode: ALT1 mux port: ECSPI3_SCLK of instance: ecspi3 101 **ALT5_GPIO5_IO[22]** — Select mux mode: ALT5 mux port: GPIO5_IO22 of instance: gpio5

8.2.4.133 SW_MUX_CTL_PAD_UART1_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART1_TXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 224h offset = 3033_0224h



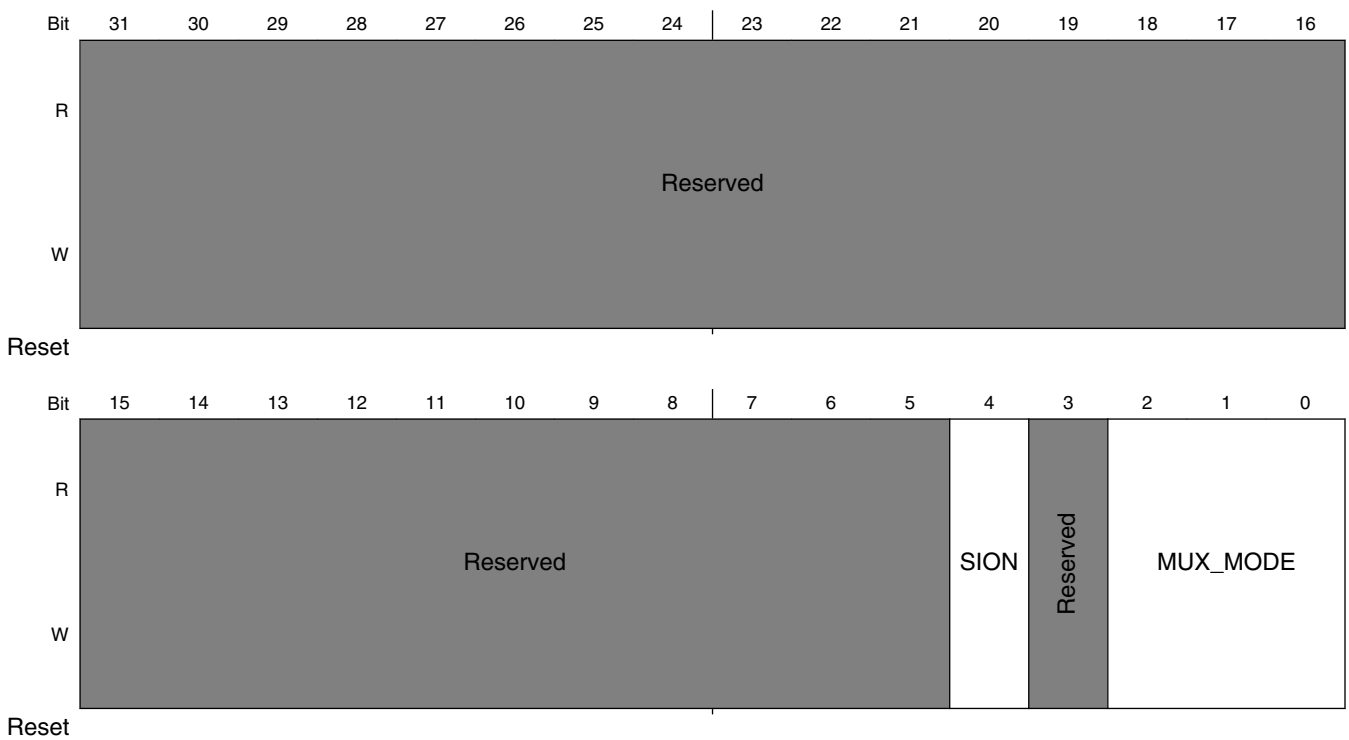
IOMUXC_SW_MUX_CTL_PAD_UART1_TXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad UART1_TXD 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: UART1_TXD. 000 **ALT0_UART1_TX** — Select mux mode: ALT0 mux port: UART1_TX of instance: uart1 001 **ALT1_ECSPI3_MOSI** — Select mux mode: ALT1 mux port: ECSPI3_MOSI of instance: ecspi3 101 **ALT5_GPIO5_IO[23]** — Select mux mode: ALT5 mux port: GPIO5_IO23 of instance: gpio5

8.2.4.134 SW_MUX_CTL_PAD_UART2_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART2_RXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 228h offset = 3033_0228h



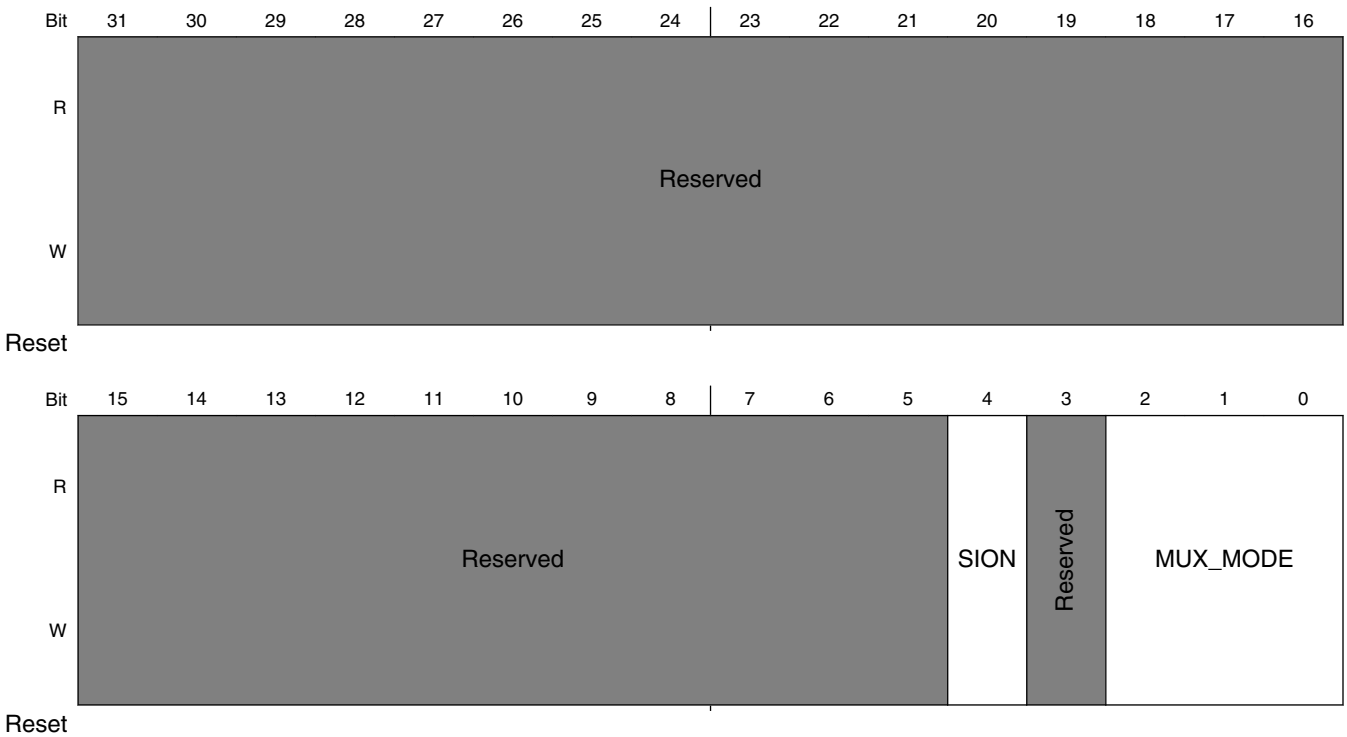
IOMUXC_SW_MUX_CTL_PAD_UART2_RXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad UART2_RXD 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: UART2_RXD. 000 **ALT0_UART2_RX** — Select mux mode: ALT0 mux port: UART2_RX of instance: uart2 001 **ALT1_ECSPI3_MISO** — Select mux mode: ALT1 mux port: ECSPI3_MISO of instance: ecspi3 011 **ALT3_GPT1_COMPARE3** — Select mux mode: ALT3 mux port: GPT1_COMPARE3 of instance: gpt1 101 **ALT5_GPIO5_IO[24]** — Select mux mode: ALT5 mux port: GPIO5_IO24 of instance: gpio5

8.2.4.135 SW_MUX_CTL_PAD_UART2_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART2_TXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 22Ch offset = 3033_022Ch



IOMUXC_SW_MUX_CTL_PAD_UART2_TXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad UART2_TXD 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: UART2_TXD. 000 **ALT0_UART2_TX** — Select mux mode: ALT0 mux port: UART2_TX of instance: uart2 001 **ALT1_ECSPI3_SS0** — Select mux mode: ALT1 mux port: ECSPI3_SS0 of instance: ecspi3

Table continues on the next page...

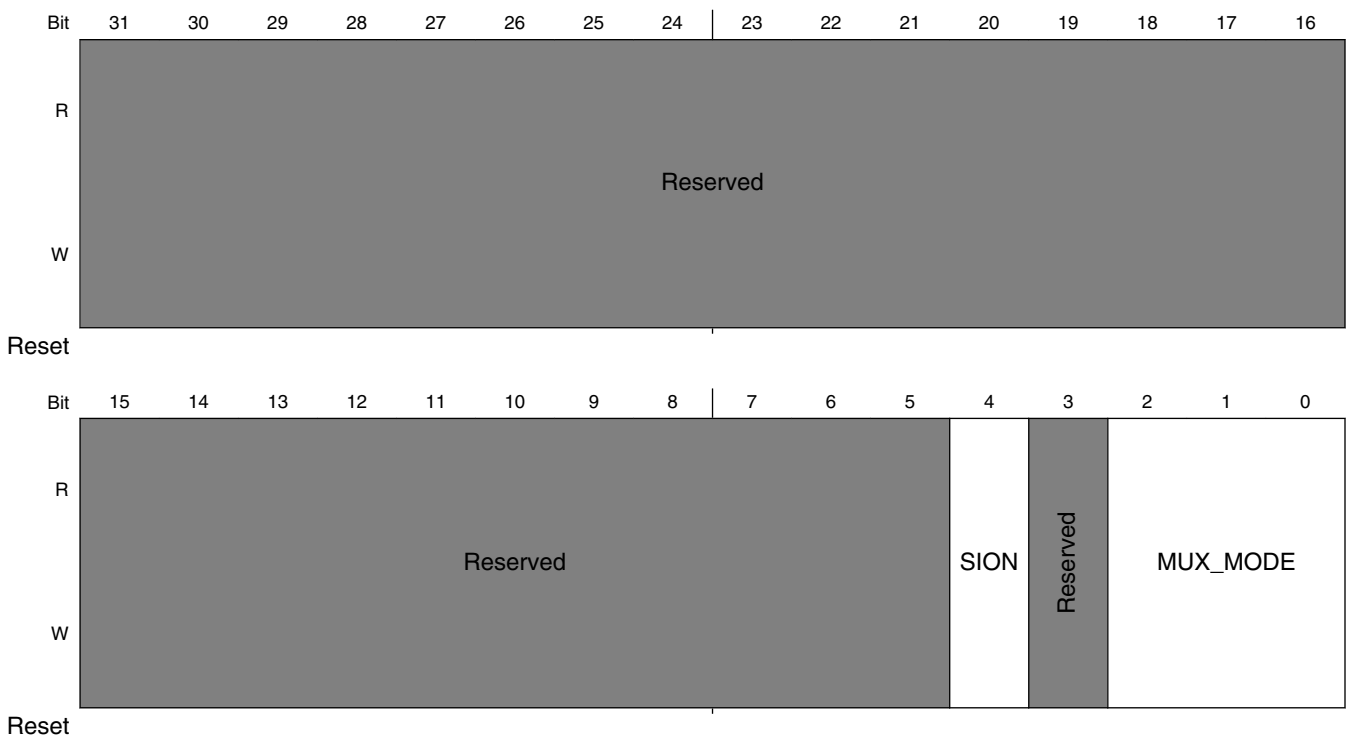
IOMUXC_SW_MUX_CTL_PAD_UART2_TXD field descriptions (continued)

Field	Description
	011 **ALT3_GPT1_COMPARE2** — Select mux mode: ALT3 mux port: GPT1_COMPARE2 of instance: gpt1 101 **ALT5_GPIO5_IO[25]** — Select mux mode: ALT5 mux port: GPIO5_IO25 of instance: gpio5

8.2.4.136 SW_MUX_CTL_PAD_UART3_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART3_RXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 230h offset = 3033_0230h



IOMUXC_SW_MUX_CTL_PAD_UART3_RXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad UART3_RXD 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

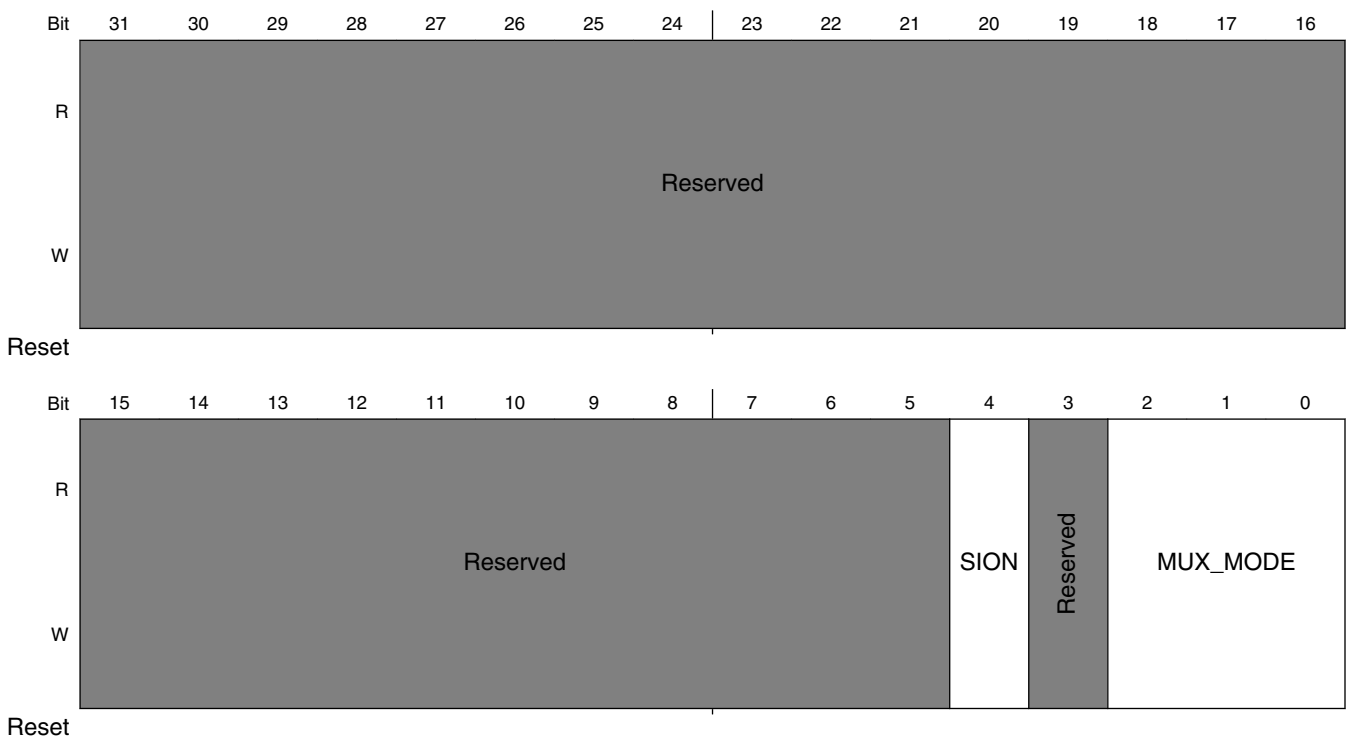
IOMUXC_SW_MUX_CTL_PAD_UART3_RXD field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: UART3_RXD. 000 **ALT0_UART3_RX** — Select mux mode: ALT0 mux port: UART3_RX of instance: uart3 001 **ALT1_UART1_CTS_B** — Select mux mode: ALT1 mux port: UART1_CTS_B of instance: uart1 010 **ALT2_USDHC3_RESET_B** — Select mux mode: ALT2 mux port: USDHC3_RESET_B of instance: usdhc3 011 **ALT3_GPT1_CAPTURE2** — Select mux mode: ALT3 mux port: GPT1_CAPTURE2 of instance: gpt1 100 **ALT4_CAN2_TX** — Select mux mode: ALT4 mux port: CAN2_TX of instance: can2 101 **ALT5_GPIO5_IO[26]** — Select mux mode: ALT5 mux port: GPIO5_IO26 of instance: gpio5

8.2.4.137 SW_MUX_CTL_PAD_UART3_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART3_TXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 234h offset = 3033_0234h



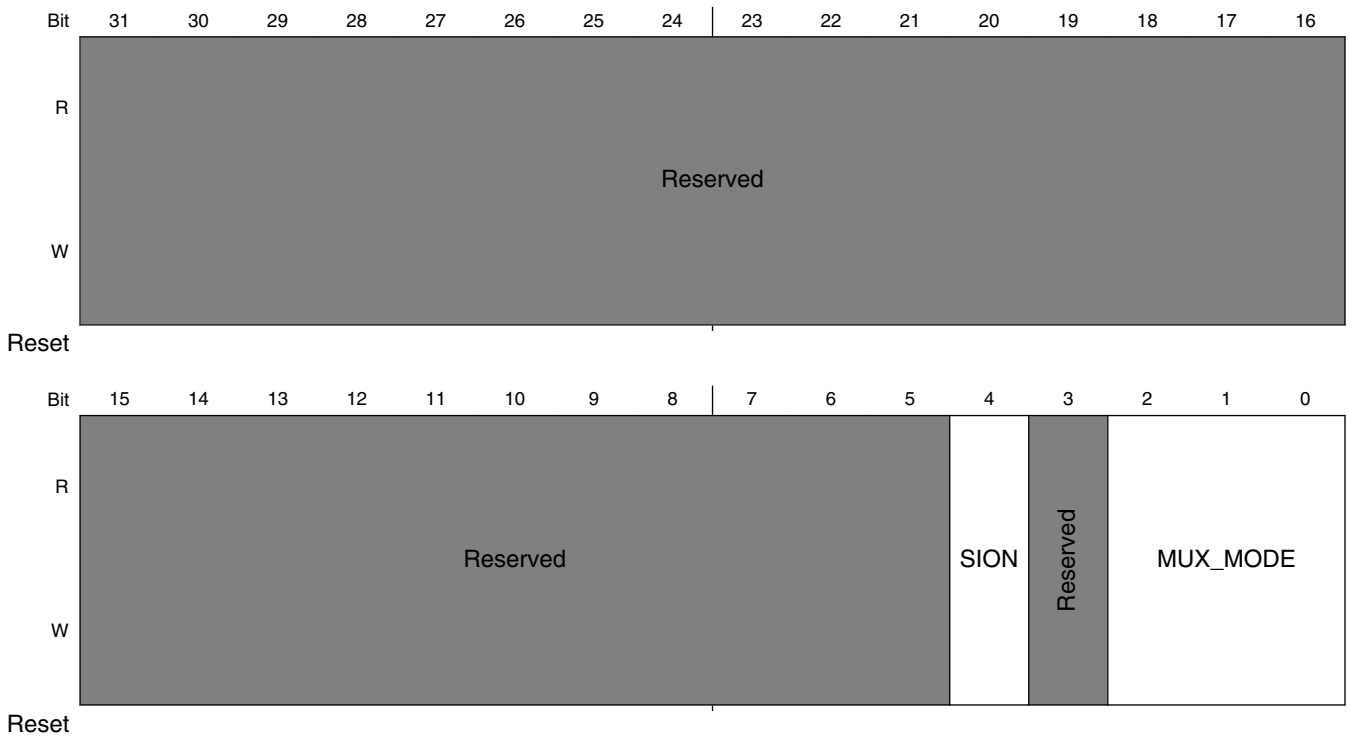
IOMUXC_SW_MUX_CTL_PAD_UART3_TXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad UART3_TXD 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: UART3_TXD. 000 **ALT0_UART3_TX** — Select mux mode: ALT0 mux port: UART3_TX of instance: uart3 001 **ALT1_UART1_RTS_B** — Select mux mode: ALT1 mux port: UART1_RTS_B of instance: uart1 010 **ALT2_USDHC3_VSELECT** — Select mux mode: ALT2 mux port: USDHC3_VSELECT of instance: usdhc3 011 **ALT3_GPT1_CLK** — Select mux mode: ALT3 mux port: GPT1_CLK of instance: gpt1 100 **ALT4_CAN2_RX** — Select mux mode: ALT4 mux port: CAN2_RX of instance: can2 101 **ALT5_GPIO5_IO[27]** — Select mux mode: ALT5 mux port: GPIO5_IO27 of instance: gpio5

8.2.4.138 SW_MUX_CTL_PAD_UART4_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART4_RXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 238h offset = 3033_0238h



IOMUXC_SW_MUX_CTL_PAD_UART4_RXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad UART4_RXD 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: UART4_RXD. 000 **ALT0_UART4_RX** — Select mux mode: ALT0 mux port: UART4_RX of instance: uart4 001 **ALT1_UART2_CTS_B** — Select mux mode: ALT1 mux port: UART2_CTS_B of instance: uart2 010 **ALT2_PCIE_CLKREQ_B** — Select mux mode: ALT2 mux port: PCIE_CLKREQ_B of instance: pcie

Table continues on the next page...

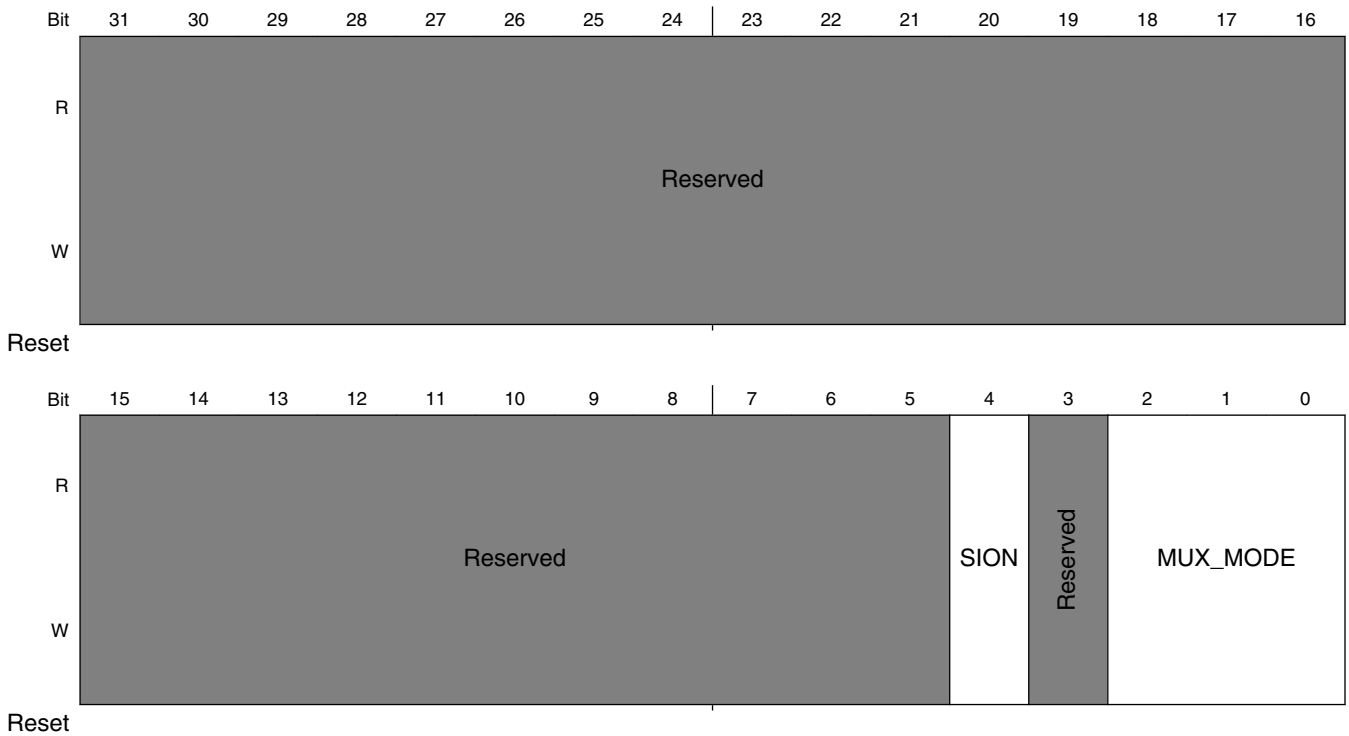
IOMUXC_SW_MUX_CTL_PAD_UART4_RXD field descriptions (continued)

Field	Description
	011 **ALT3_GPT1_COMPARE1** — Select mux mode: ALT3 mux port: GPT1_COMPARE1 of instance: gpt1 100 **ALT4_I2C6_SCL** — Select mux mode: ALT4 mux port: I2C6_SCL of instance: i2c6 101 **ALT5_GPIO5_IO[28]** — Select mux mode: ALT5 mux port: GPIO5_IO28 of instance: gpio5

8.2.4.139 SW_MUX_CTL_PAD_UART4_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART4_TXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 23Ch offset = 3033_023Ch



IOMUXC_SW_MUX_CTL_PAD_UART4_TXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad UART4_TXD 0 **DISABLED** — Input Path is determined by functionality

Table continues on the next page...

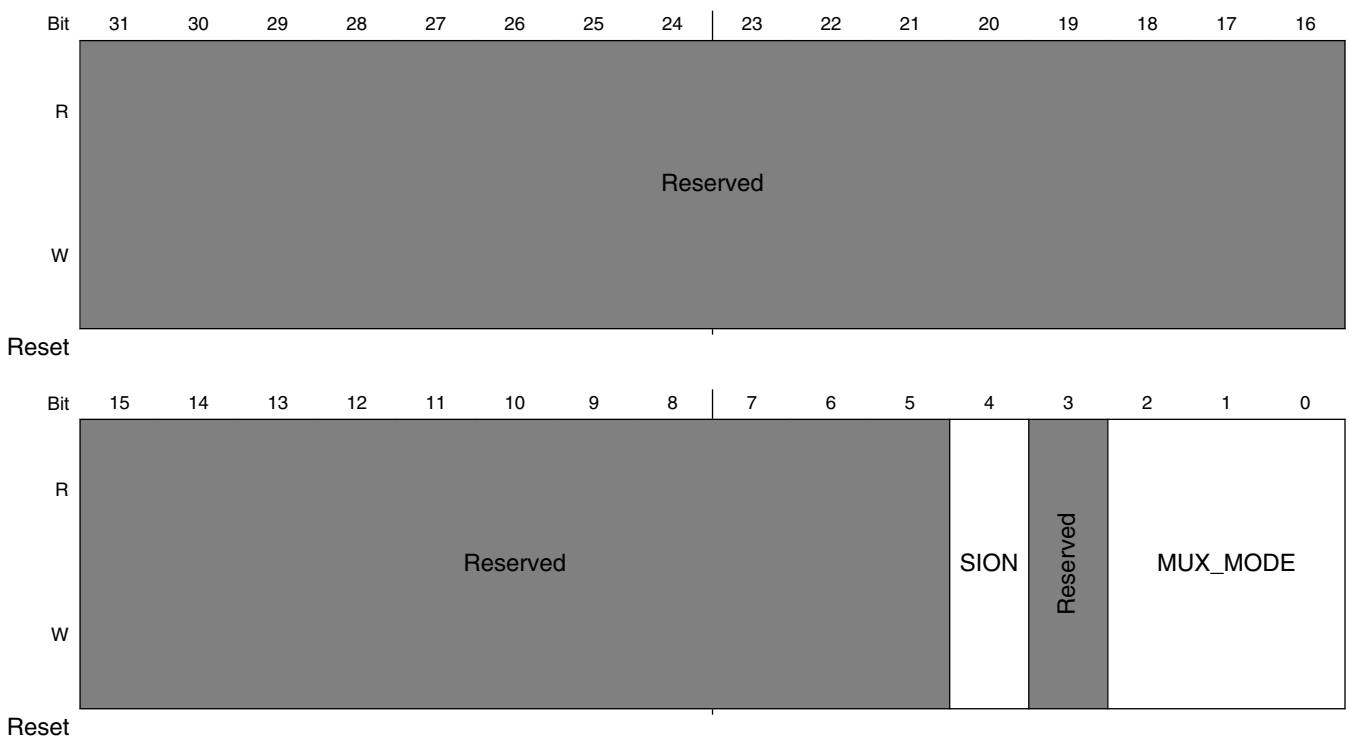
IOMUXC_SW_MUX_CTL_PAD_UART4_TXD field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: UART4_TXD. 000 **ALT0_UART4_TX** — Select mux mode: ALT0 mux port: UART4_TX of instance: uart4 001 **ALT1_UART2_RTS_B** — Select mux mode: ALT1 mux port: UART2_RTS_B of instance: uart2 011 **ALT3_GPT1_CAPTURE1** — Select mux mode: ALT3 mux port: GPT1_CAPTURE1 of instance: gpt1 100 **ALT4_I2C6_SDA** — Select mux mode: ALT4 mux port: I2C6_SDA of instance: i2c6 101 **ALT5_GPIO5_IO[29]** — Select mux mode: ALT5 mux port: GPIO5_IO29 of instance: gpio5

8.2.4.140 SW_MUX_CTL_PAD_HDMI_DDC_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_HDMI_DDC_SCL)

SW_MUX_CTL Register

Address: 3033_0000h base + 240h offset = 3033_0240h



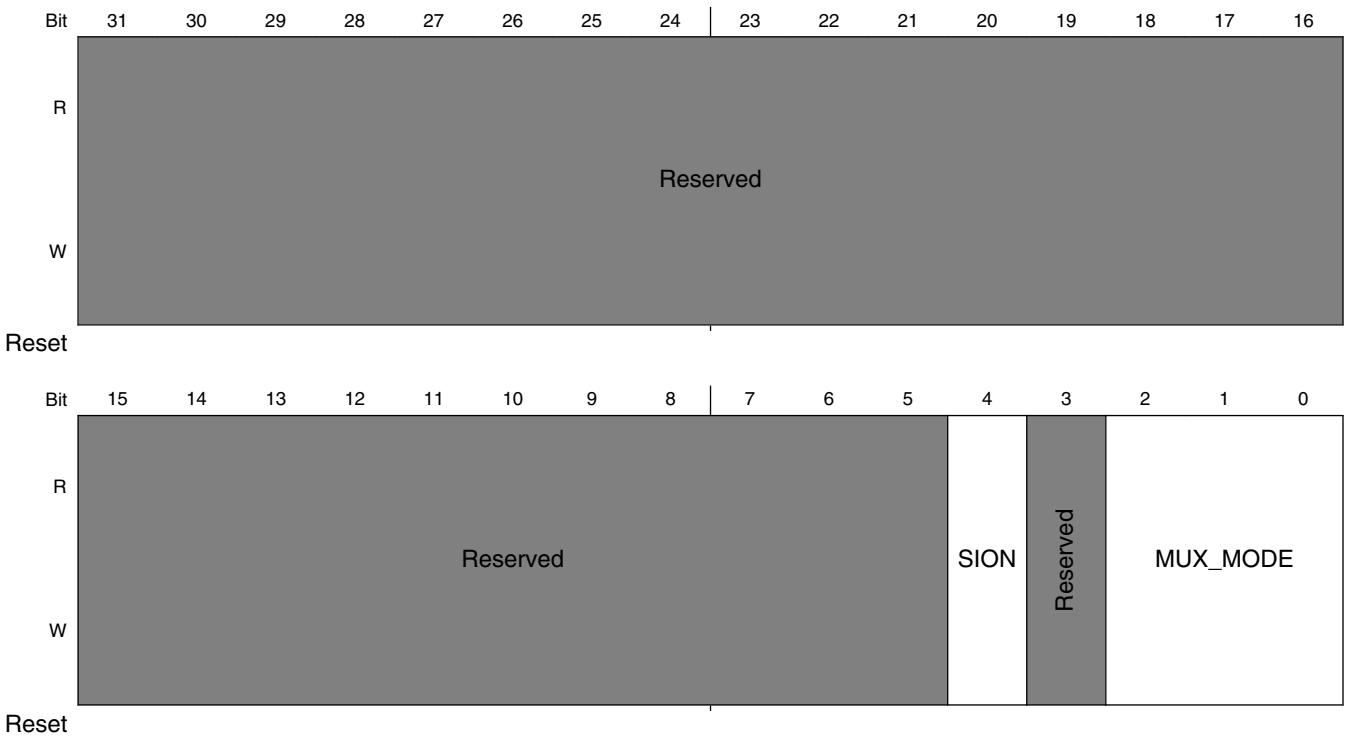
IOMUXC_SW_MUX_CTL_PAD_HDMI_DDC_SCL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad HDMI_DDC_SCL 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: HDMI_DDC_SCL. 000 **ALT0_HDMIMIX_HDMI_SCL** — Select mux mode: ALT0 mux port: HDMIMIX_HDMI_SCL of instance: hdmi 011 **ALT3_I2C5_SCL** — Select mux mode: ALT3 mux port: I2C5_SCL of instance: i2c5 100 **ALT4_CAN1_TX** — Select mux mode: ALT4 mux port: CAN1_TX of instance: can1 101 **ALT5_GPIO3_IO[26]** — Select mux mode: ALT5 mux port: GPIO3_IO26 of instance: gpio3

8.2.4.141 SW_MUX_CTL_PAD_HDMI_DDC_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_HDMI_DDC_SDA)

SW_MUX_CTL Register

Address: 3033_0000h base + 244h offset = 3033_0244h



IOMUXC_SW_MUX_CTL_PAD_HDMI_DDC_SDA field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad HDMI_DDC_SDA 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: HDMI_DDC_SDA. 000 **ALT0_HDMIMIX_HDMI_SDA** — Select mux mode: ALT0 mux port: HDMIMIX_HDMI_SDA of instance: hdmi 011 **ALT3_I2C5_SDA** — Select mux mode: ALT3 mux port: I2C5_SDA of instance: i2c5

Table continues on the next page...

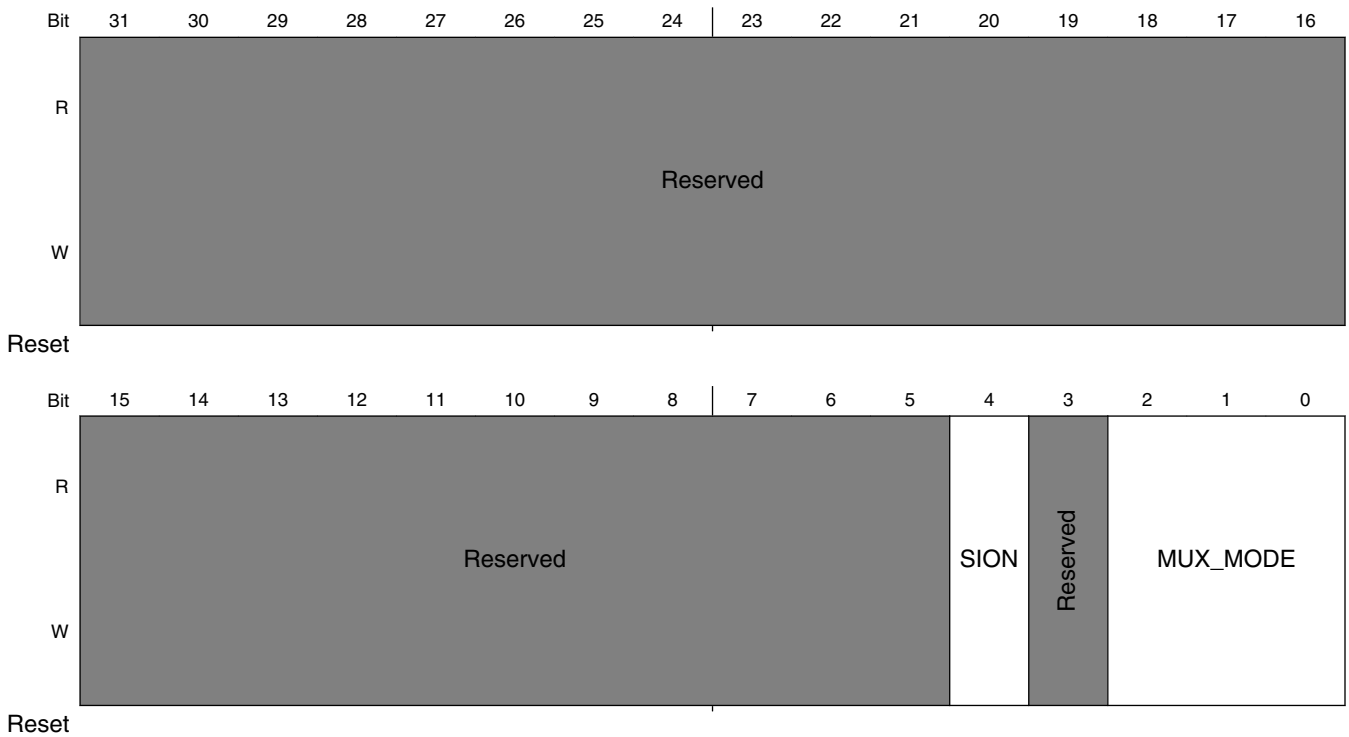
IOMUXC_SW_MUX_CTL_PAD_HDMI_DDC_SDA field descriptions (continued)

Field	Description
	100 **ALT4_CAN1_RX** — Select mux mode: ALT4 mux port: CAN1_RX of instance: can1 101 **ALT5_GPIO3_IO[27]** — Select mux mode: ALT5 mux port: GPIO3_IO27 of instance: gpio3

8.2.4.142 SW_MUX_CTL_PAD_HDMI_CEC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_HDMI_CEC)

SW_MUX_CTL Register

Address: 3033_0000h base + 248h offset = 3033_0248h



IOMUXC_SW_MUX_CTL_PAD_HDMI_CEC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad HDMI_CEC 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved

Table continues on the next page...

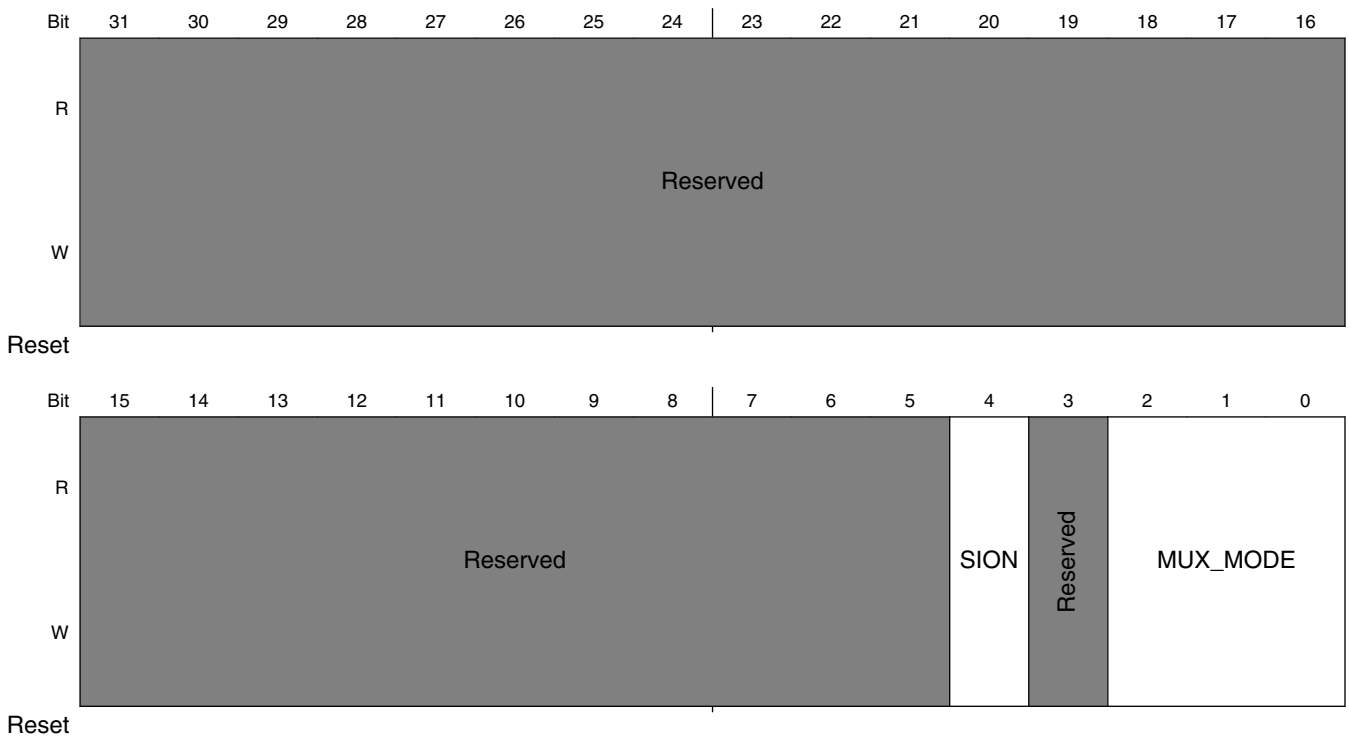
IOMUXC_SW_MUX_CTL_PAD_HDMI_CEC field descriptions (continued)

Field	Description
MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: HDMI_CEC. 000 **ALT0_HDMIMIX_HDMI_CEC** — Select mux mode: ALT0 mux port: HDMIMIX_HDMI_CEC of instance: hdmi 011 **ALT3_I2C6_SCL** — Select mux mode: ALT3 mux port: I2C6_SCL of instance: i2c6 100 **ALT4_CAN2_TX** — Select mux mode: ALT4 mux port: CAN2_TX of instance: can2 101 **ALT5_GPIO3_IO[28]** — Select mux mode: ALT5 mux port: GPIO3_IO28 of instance: gpio3

8.2.4.143 SW_MUX_CTL_PAD_HDMI_HPD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_HDMI_HPD)

SW_MUX_CTL Register

Address: 3033_0000h base + 24Ch offset = 3033_024Ch



IOMUXC_SW_MUX_CTL_PAD_HDMI_HPD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field.

Table continues on the next page...

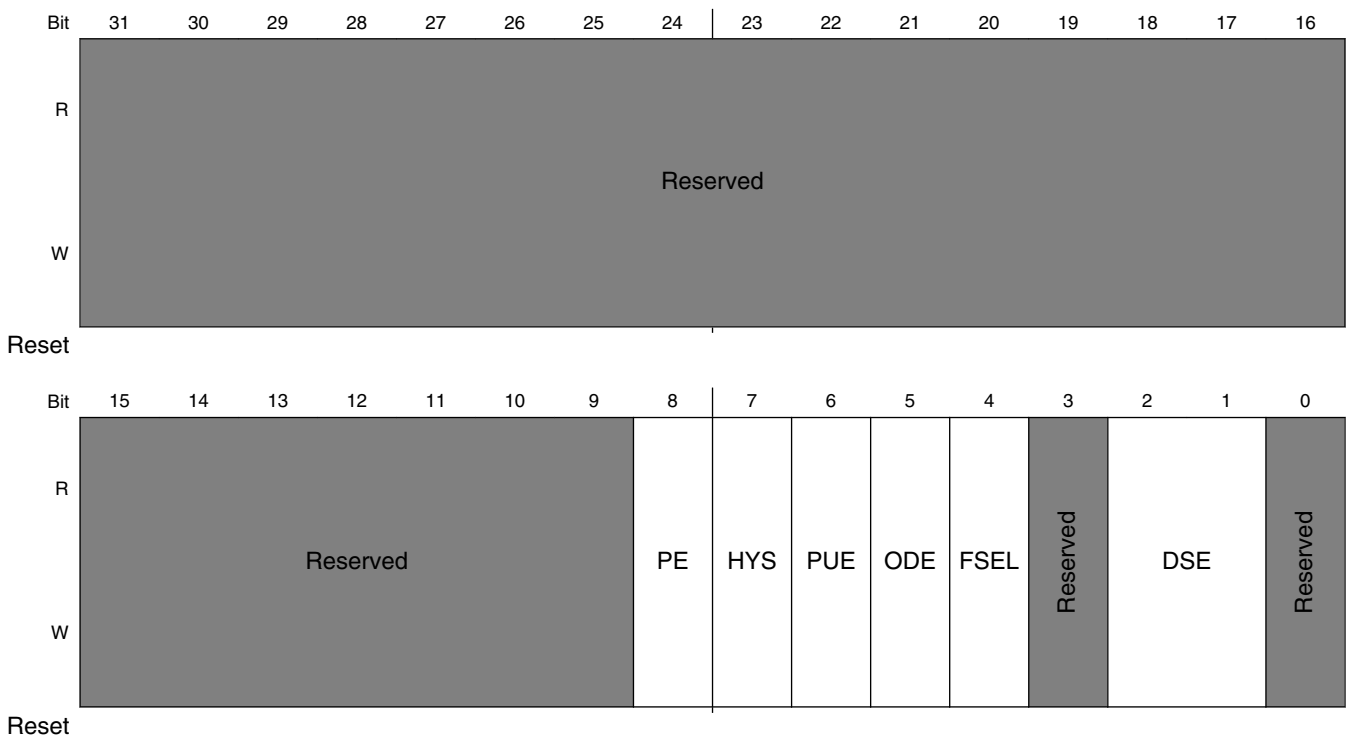
IOMUXC_SW_MUX_CTL_PAD_HDMI_HPD field descriptions (continued)

Field	Description
	Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 **ENABLED** — Force input path of pad HDMI_HPD 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: HDMI_HPD. 000 **ALT0_HDMIMIX_HDMI_HPD** — Select mux mode: ALT0 mux port: HDMIMIX_HDMI_HPD of instance: hdmi 001 **ALT1_AUDIOMIX_HDMI_HPD_O** — Select mux mode: ALT1 mux port: AUDIOMIX_HDMI_HPD_O of instance: hdmi 011 **ALT3_I2C6_SDA** — Select mux mode: ALT3 mux port: I2C6_SDA of instance: i2c6 100 **ALT4_CAN2_RX** — Select mux mode: ALT4 mux port: CAN2_RX of instance: can2 101 **ALT5_GPIO3_IO[29]** — Select mux mode: ALT5 mux port: GPIO3_IO29 of instance: gpio3

8.2.4.144 SW_PAD_CTL_PAD_BOOT_MODE0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0)

SW_PAD_CTL Register

Address: 3033_0000h base + 250h offset = 3033_0250h



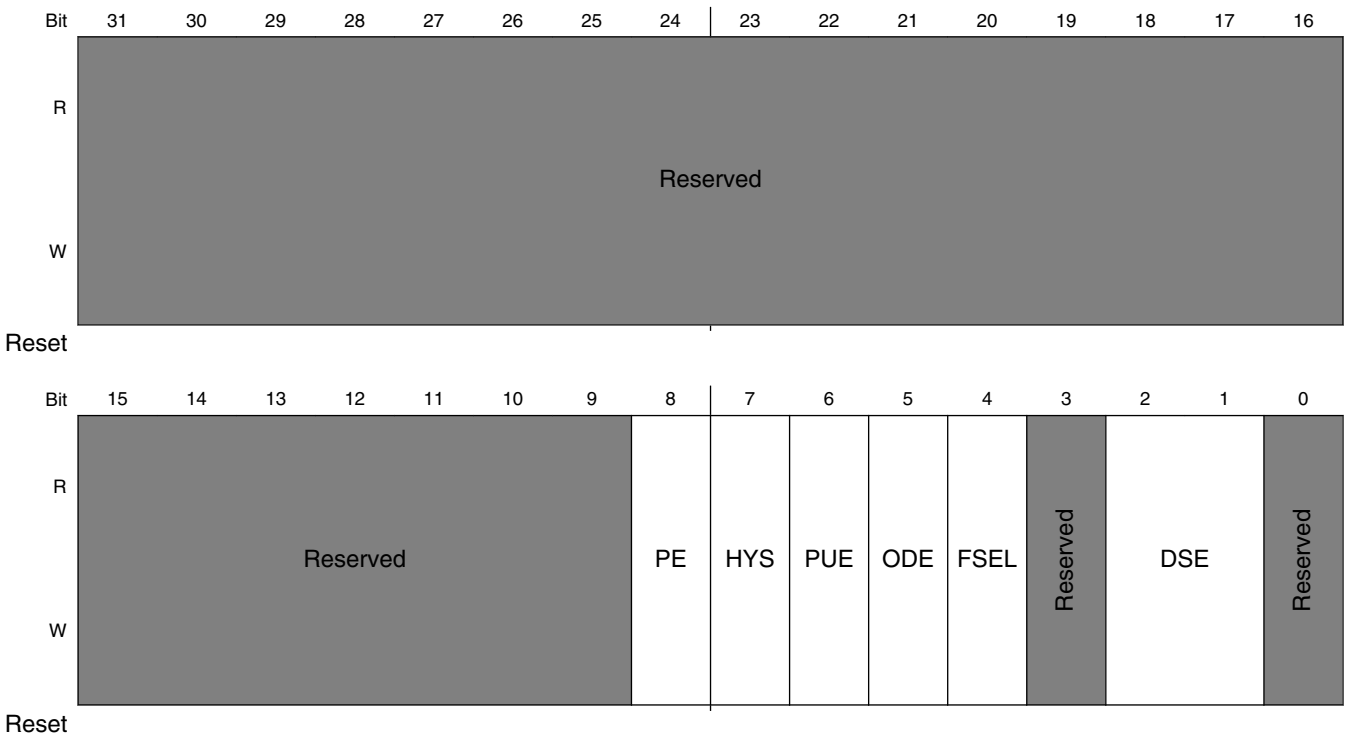
IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: BOOT_MODE0 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: BOOT_MODE0 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: BOOT_MODE0 0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: BOOT_MODE0 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: BOOT_MODE0 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: BOOT_MODE0 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.145 SW_PAD_CTL_PAD_BOOT_MODE1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1)

SW_PAD_CTL Register

Address: 3033_0000h base + 254h offset = 3033_0254h



IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: BOOT_MODE1 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: BOOT_MODE1 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: BOOT_MODE1

Table continues on the next page...

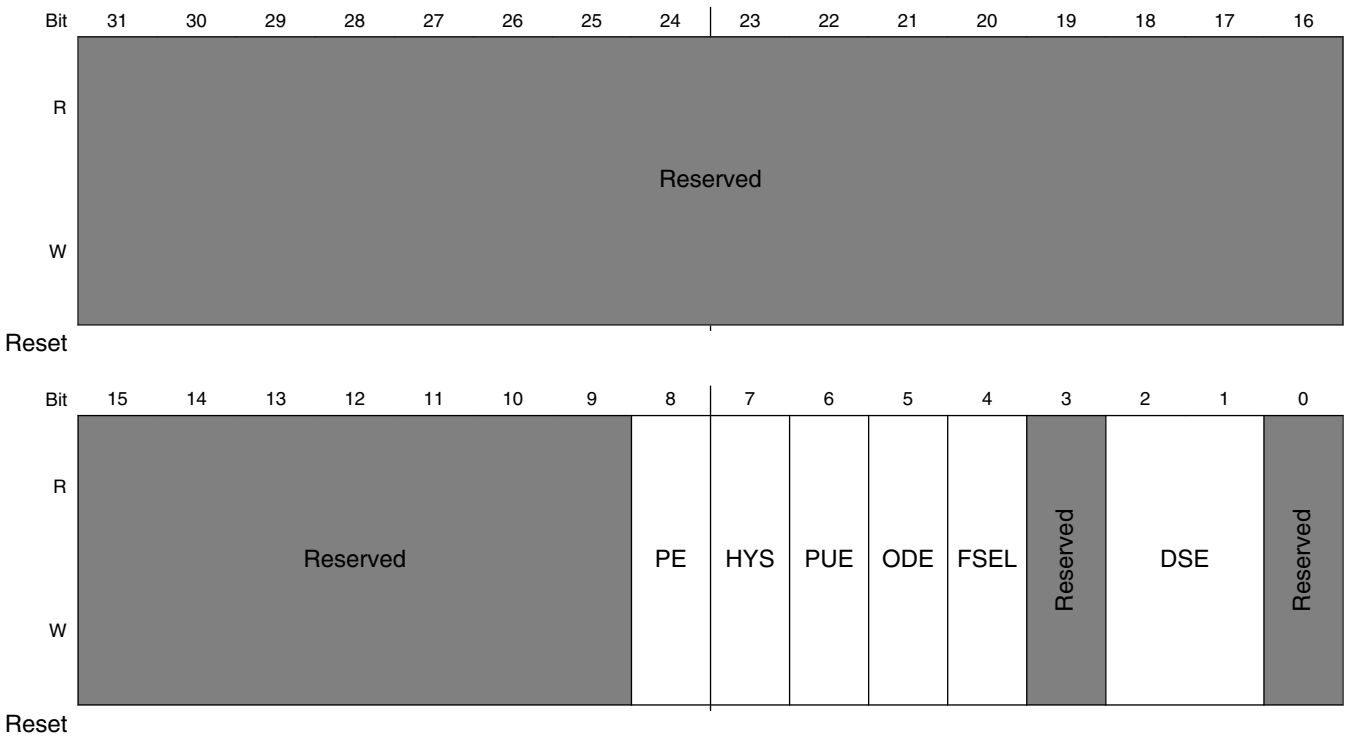
IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: BOOT_MODE1 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: BOOT_MODE1 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: BOOT_MODE1 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.146 SW_PAD_CTL_PAD_BOOT_MODE2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE2)

SW_PAD_CTL Register

Address: 3033_0000h base + 258h offset = 3033_0258h



IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: BOOT_MODE2 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: BOOT_MODE2 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: BOOT_MODE2

Table continues on the next page...

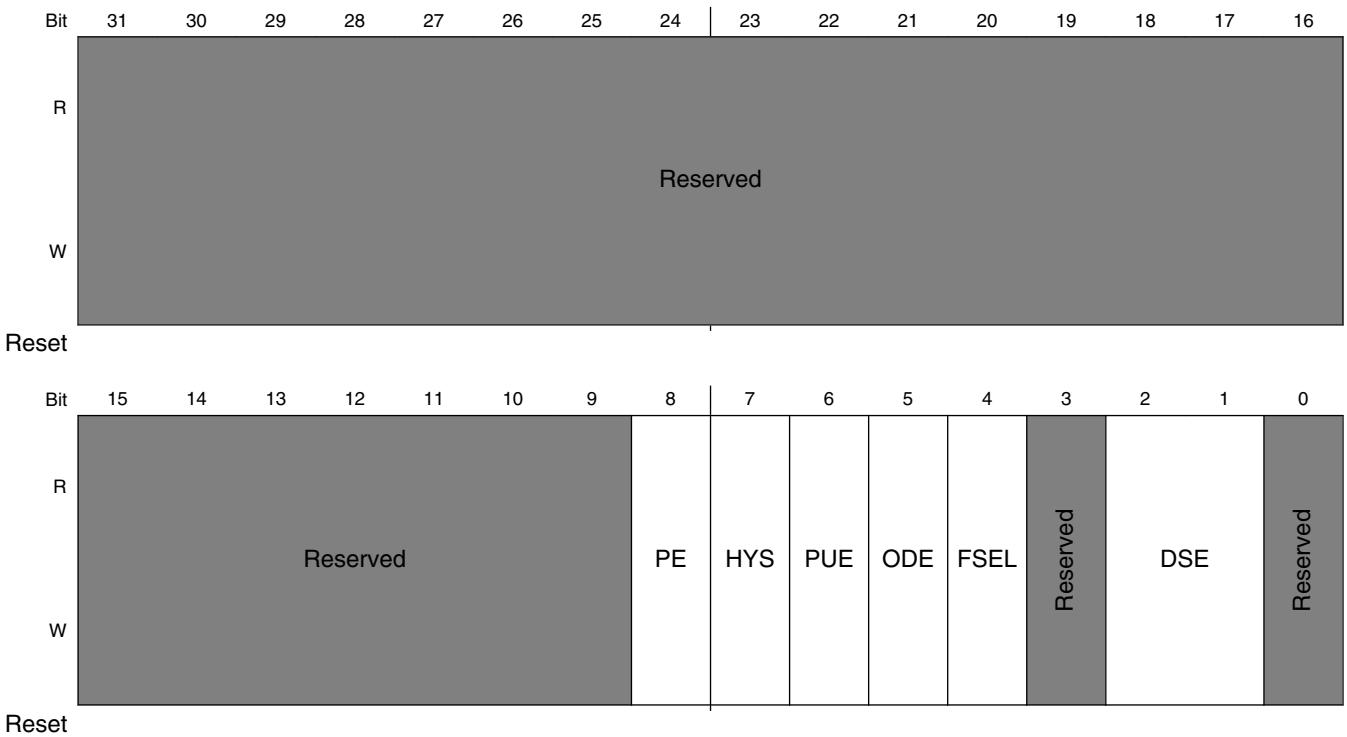
IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE2 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: BOOT_MODE2 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: BOOT_MODE2 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: BOOT_MODE2 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.147 SW_PAD_CTL_PAD_BOOT_MODE3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE3)

SW_PAD_CTL Register

Address: 3033_0000h base + 25Ch offset = 3033_025Ch



IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: BOOT_MODE3 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: BOOT_MODE3 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: BOOT_MODE3

Table continues on the next page...

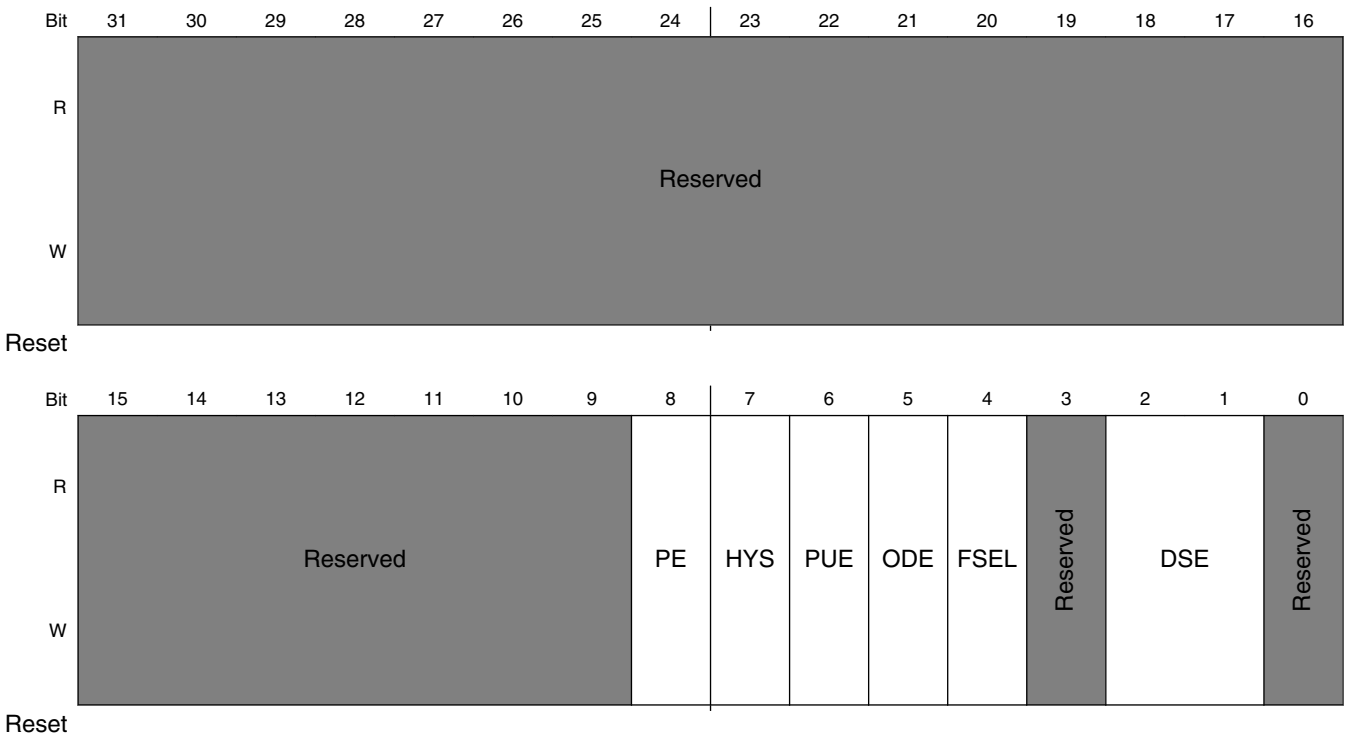
IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE3 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: BOOT_MODE3 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: BOOT_MODE3 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: BOOT_MODE3 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.148 SW_PAD_CTL_PAD_JTAG_MOD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD)

SW_PAD_CTL Register

Address: 3033_0000h base + 260h offset = 3033_0260h



IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: JTAG_MOD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: JTAG_MOD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: JTAG_MOD

Table continues on the next page...

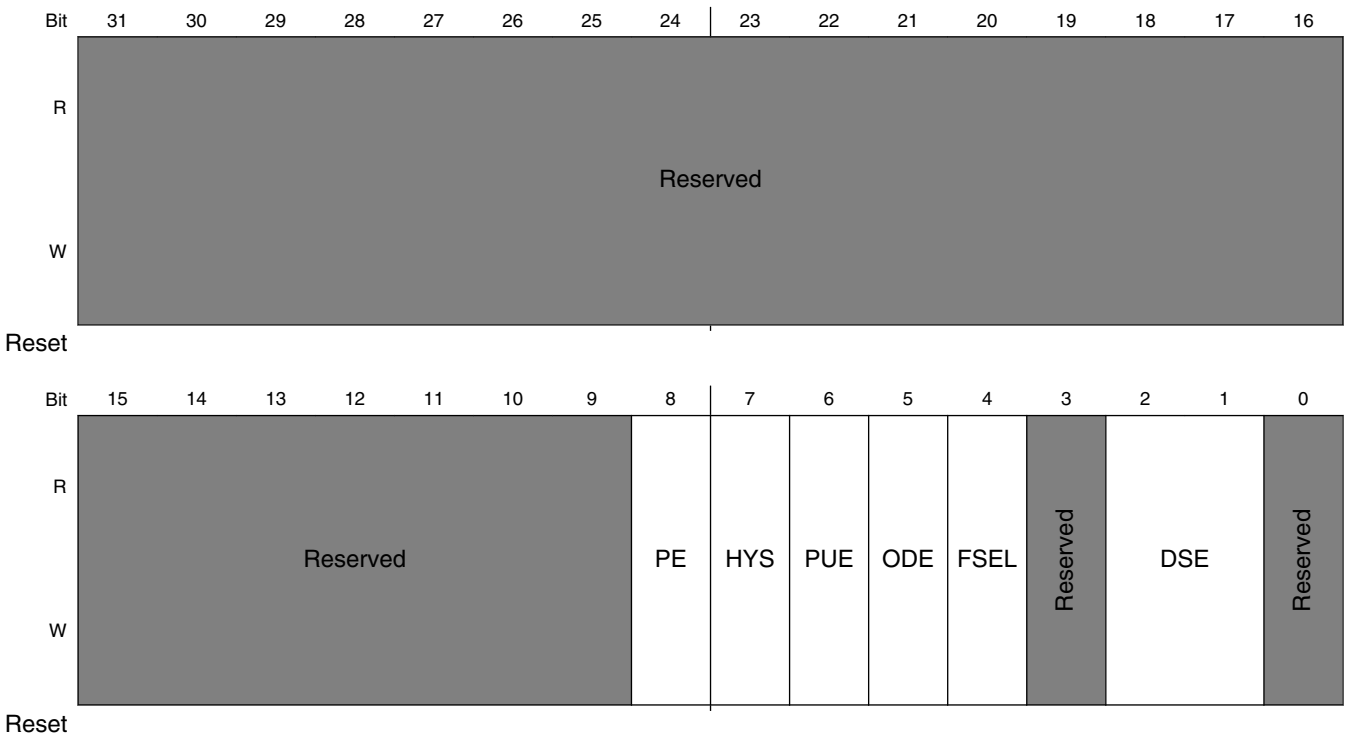
IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: JTAG_MOD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: JTAG_MOD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: JTAG_MOD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.149 SW_PAD_CTL_PAD_JTAG_TDI SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI)

SW_PAD_CTL Register

Address: 3033_0000h base + 264h offset = 3033_0264h



IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: JTAG_TDI 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: JTAG_TDI 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: JTAG_TDI

Table continues on the next page...

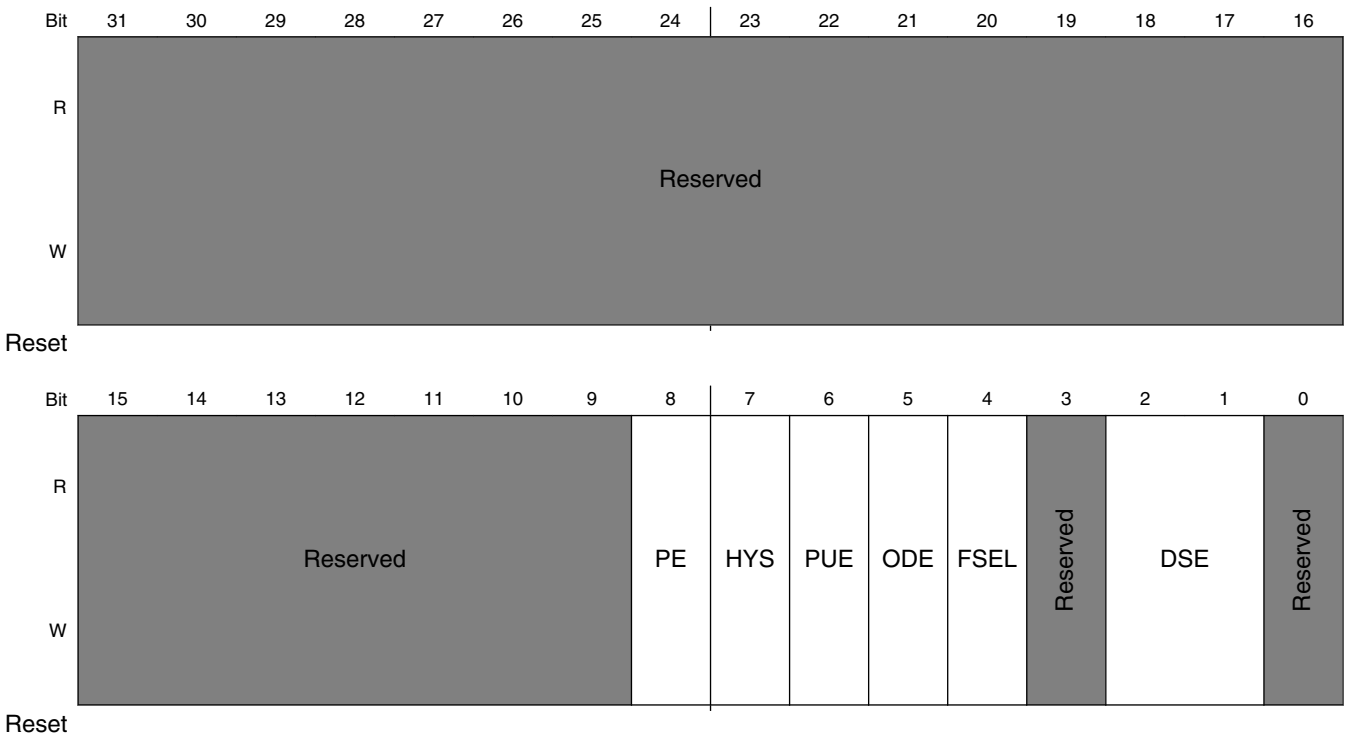
IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: JTAG_TDI 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: JTAG_TDI 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: JTAG_TDI 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.150 SW_PAD_CTL_PAD_JTAG_TMS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS)

SW_PAD_CTL Register

Address: 3033_0000h base + 268h offset = 3033_0268h



IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: JTAG_TMS 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: JTAG_TMS 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: JTAG_TMS

Table continues on the next page...

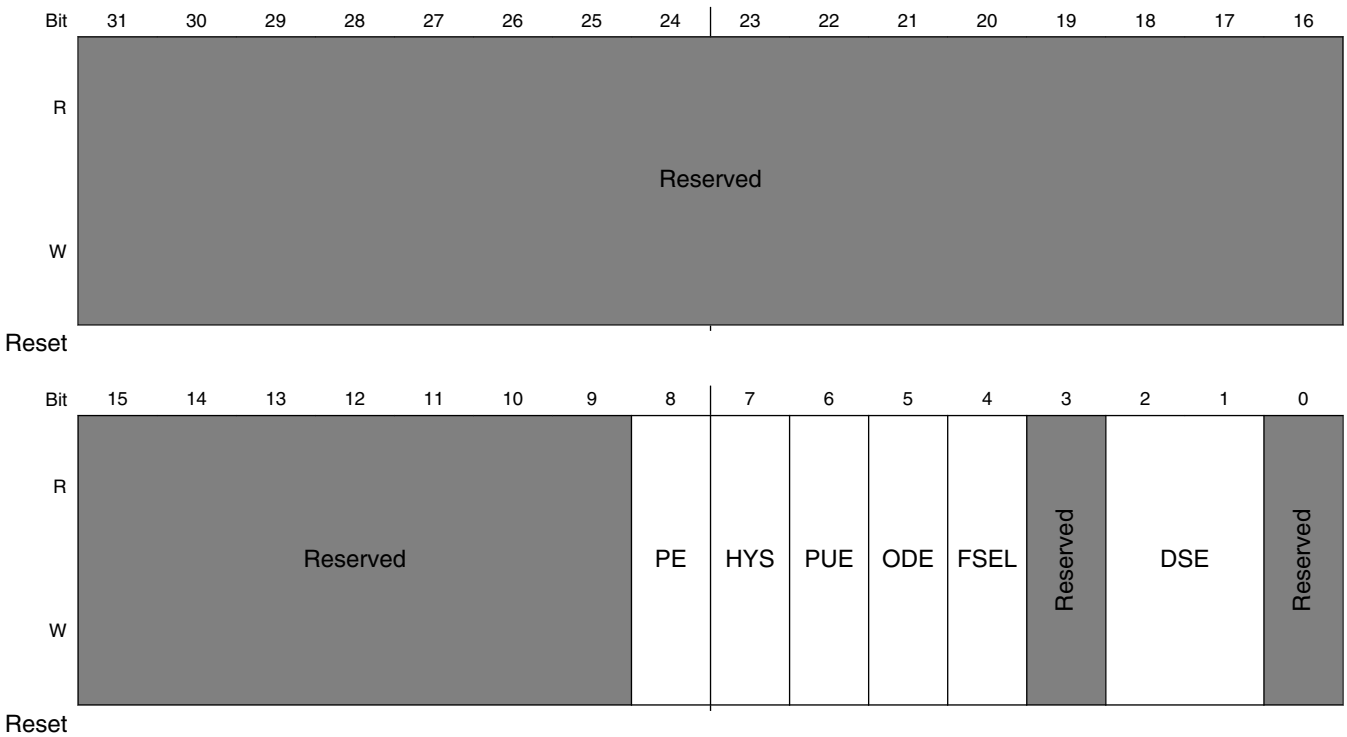
IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: JTAG_TMS 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: JTAG_TMS 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: JTAG_TMS 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.151 SW_PAD_CTL_PAD_JTAG_TCK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK)

SW_PAD_CTL Register

Address: 3033_0000h base + 26Ch offset = 3033_026Ch



IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: JTAG_TCK 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: JTAG_TCK 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: JTAG_TCK

Table continues on the next page...

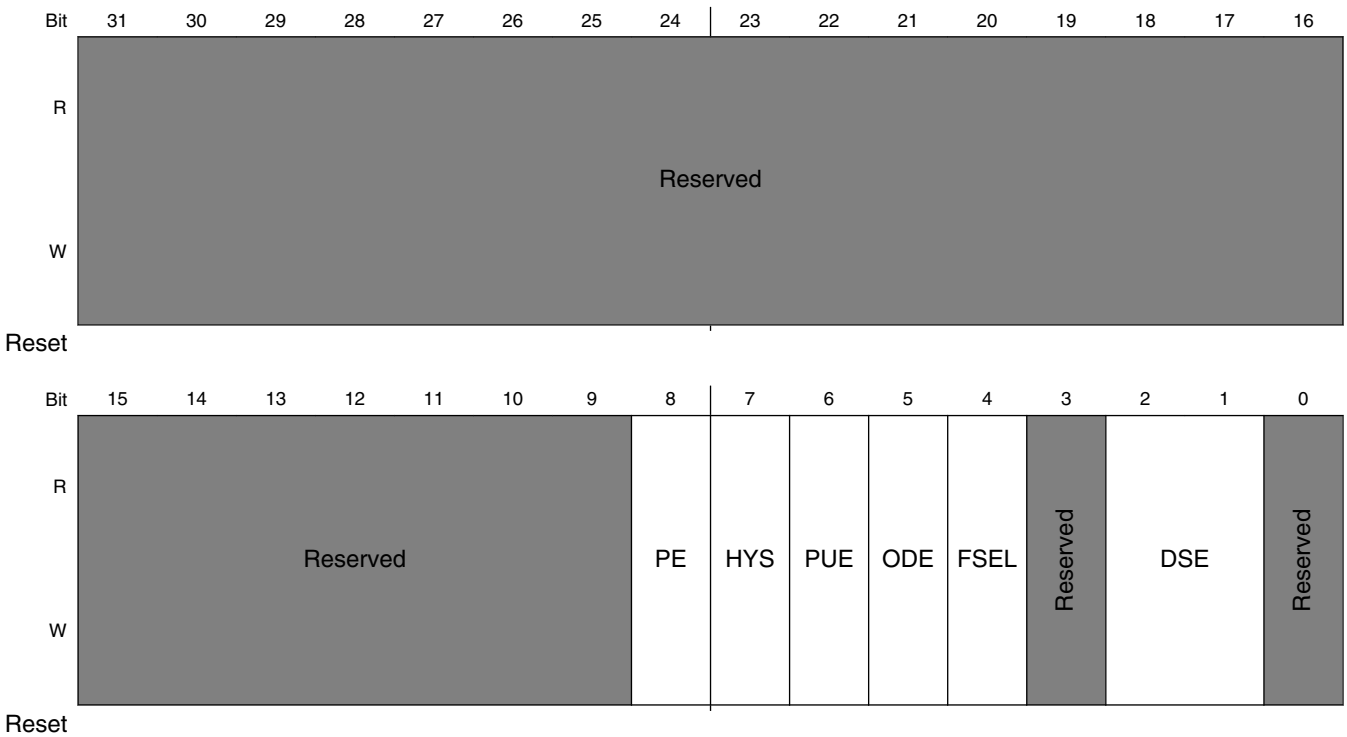
IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: JTAG_TCK 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: JTAG_TCK 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: JTAG_TCK 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.152 SW_PAD_CTL_PAD_JTAG_TDO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO)

SW_PAD_CTL Register

Address: 3033_0000h base + 270h offset = 3033_0270h



IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: JTAG_TDO 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: JTAG_TDO 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: JTAG_TDO

Table continues on the next page...

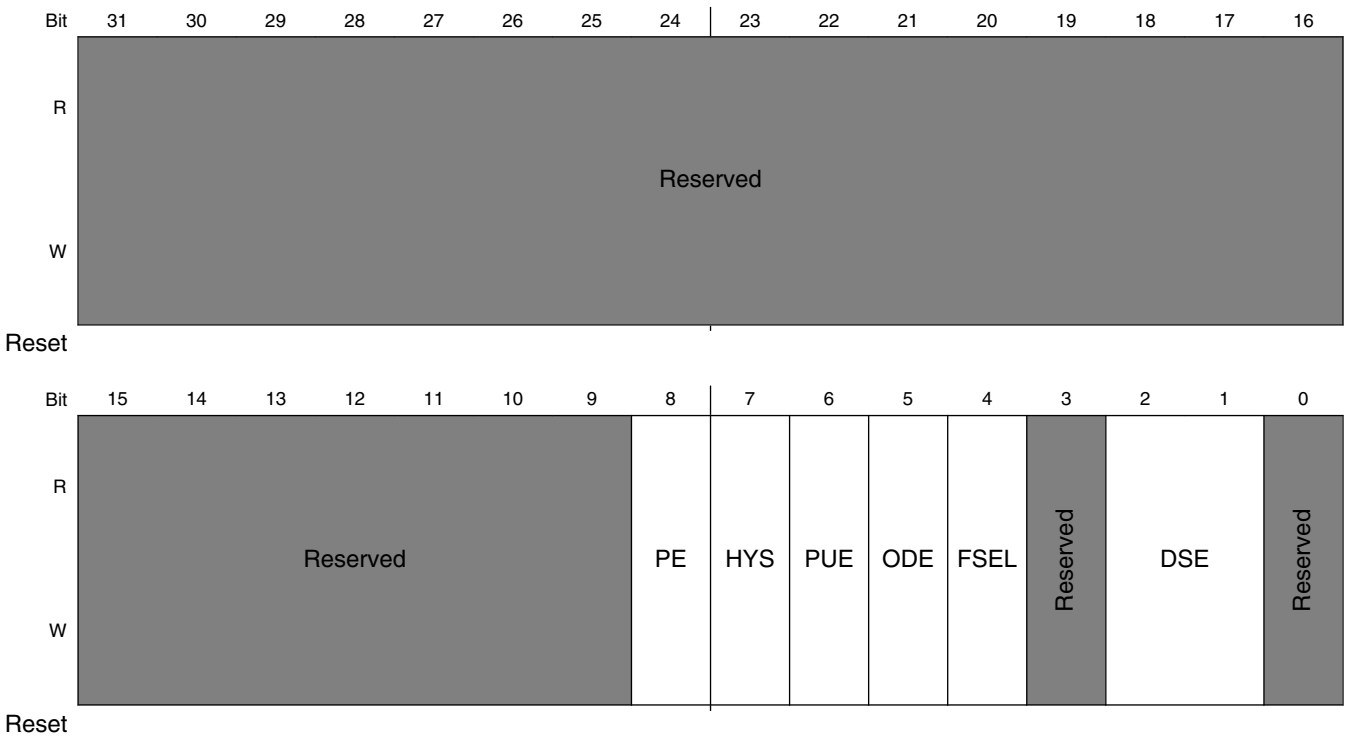
IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: JTAG_TDO 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: JTAG_TDO 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: JTAG_TDO 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.153 SW_PAD_CTL_PAD_GPIO1_IO00 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO00)

SW_PAD_CTL Register

Address: 3033_0000h base + 274h offset = 3033_0274h



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO00 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO00 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO00 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO00

Table continues on the next page...

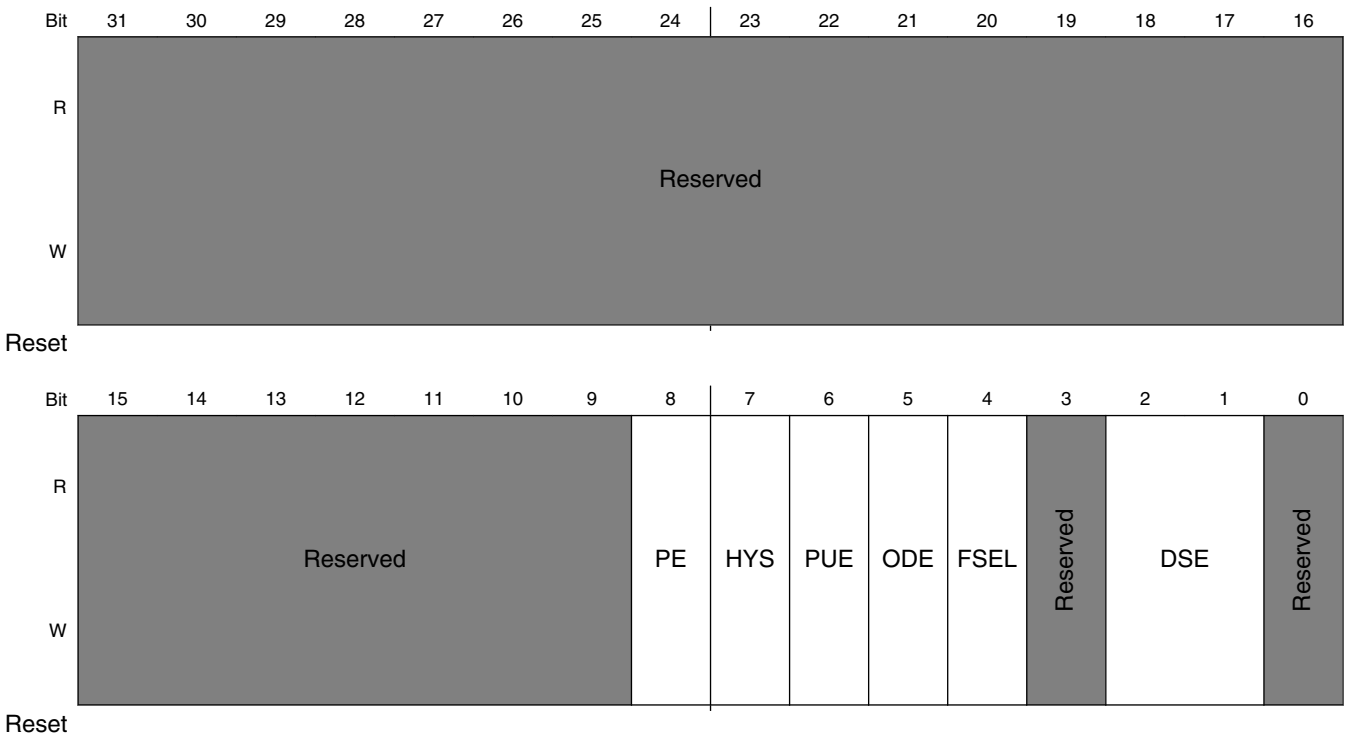
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO00 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO00 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO00 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO00 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.154 SW_PAD_CTL_PAD_GPIO1_IO01 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO01)

SW_PAD_CTL Register

Address: 3033_0000h base + 278h offset = 3033_0278h



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO01 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO01 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO01 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO01

Table continues on the next page...

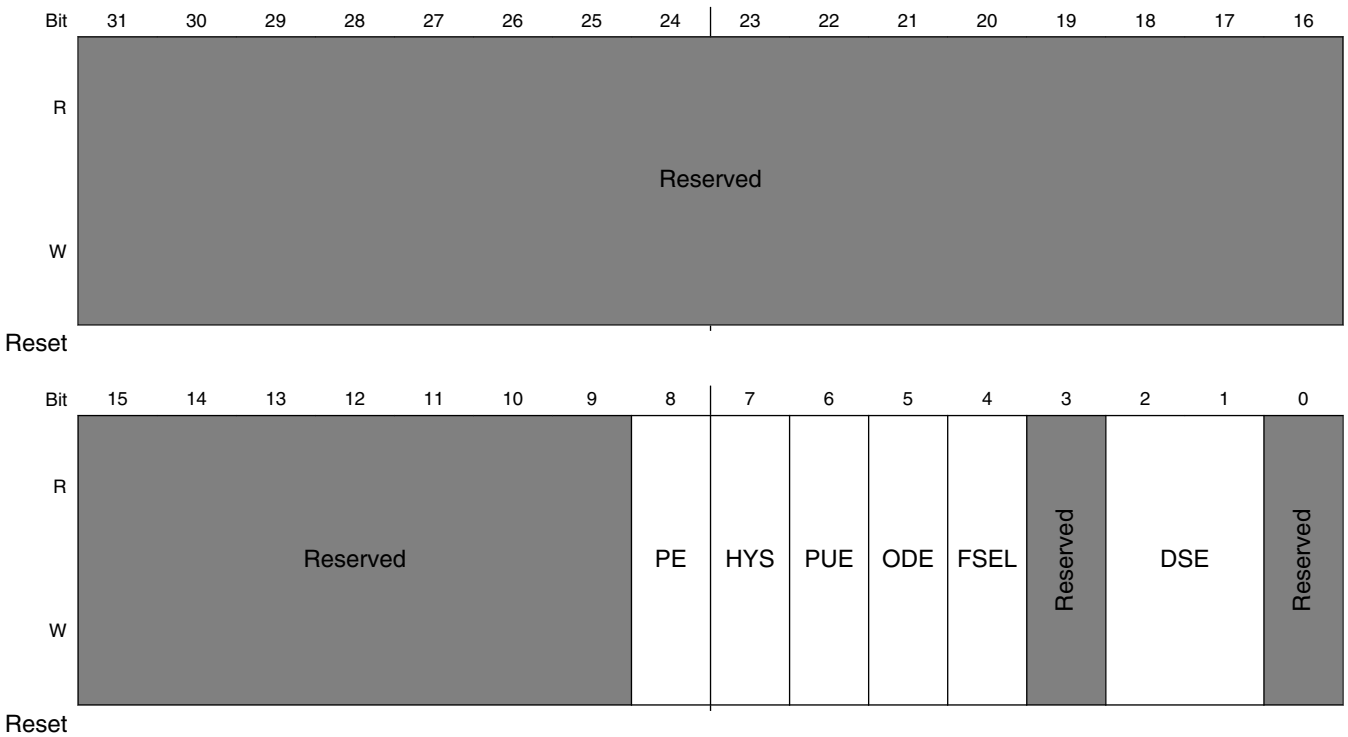
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO01 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO01 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO01 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO01 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.155 SW_PAD_CTL_PAD_GPIO1_IO02 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO02)

SW_PAD_CTL Register

Address: 3033_0000h base + 27Ch offset = 3033_027Ch



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO02 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO02 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO02 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO02

Table continues on the next page...

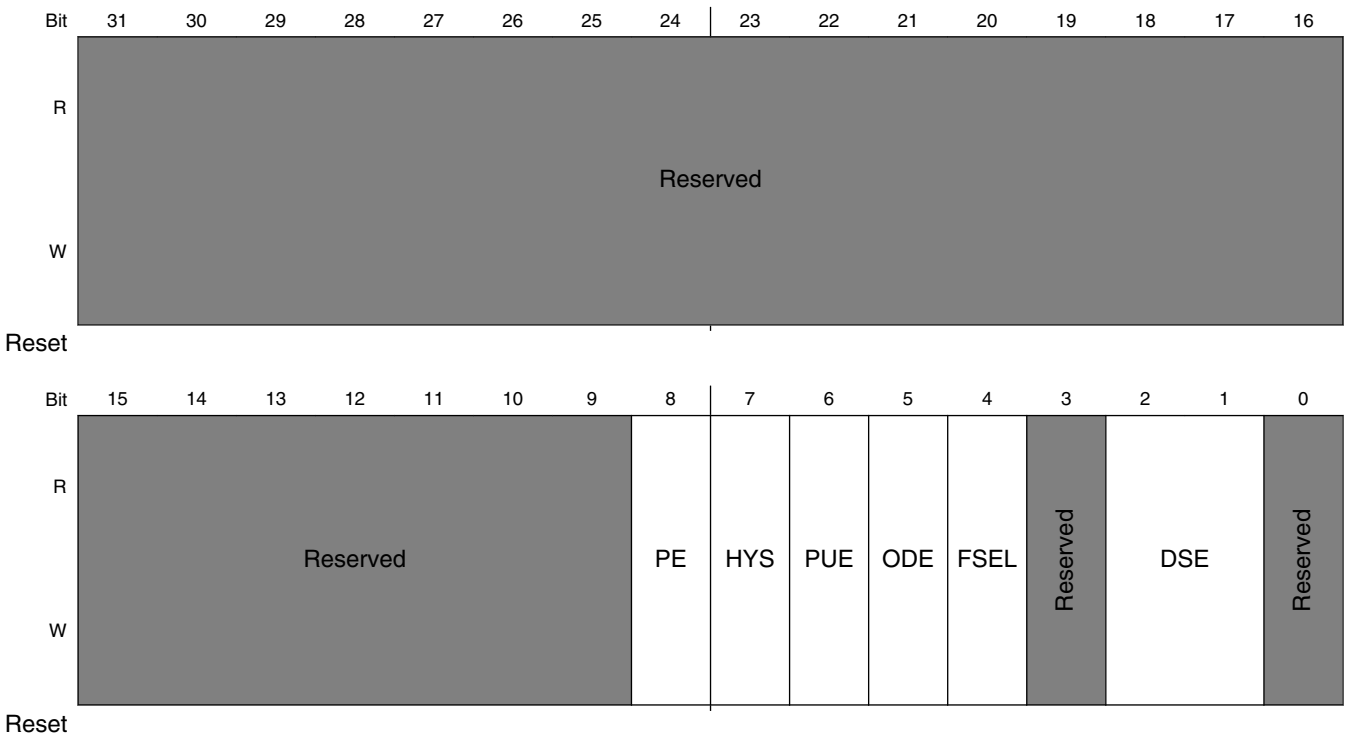
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO02 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO02 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO02 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO02 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.156 SW_PAD_CTL_PAD_GPIO1_IO03 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO03)

SW_PAD_CTL Register

Address: 3033_0000h base + 280h offset = 3033_0280h



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO03 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO03 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO03 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO03

Table continues on the next page...

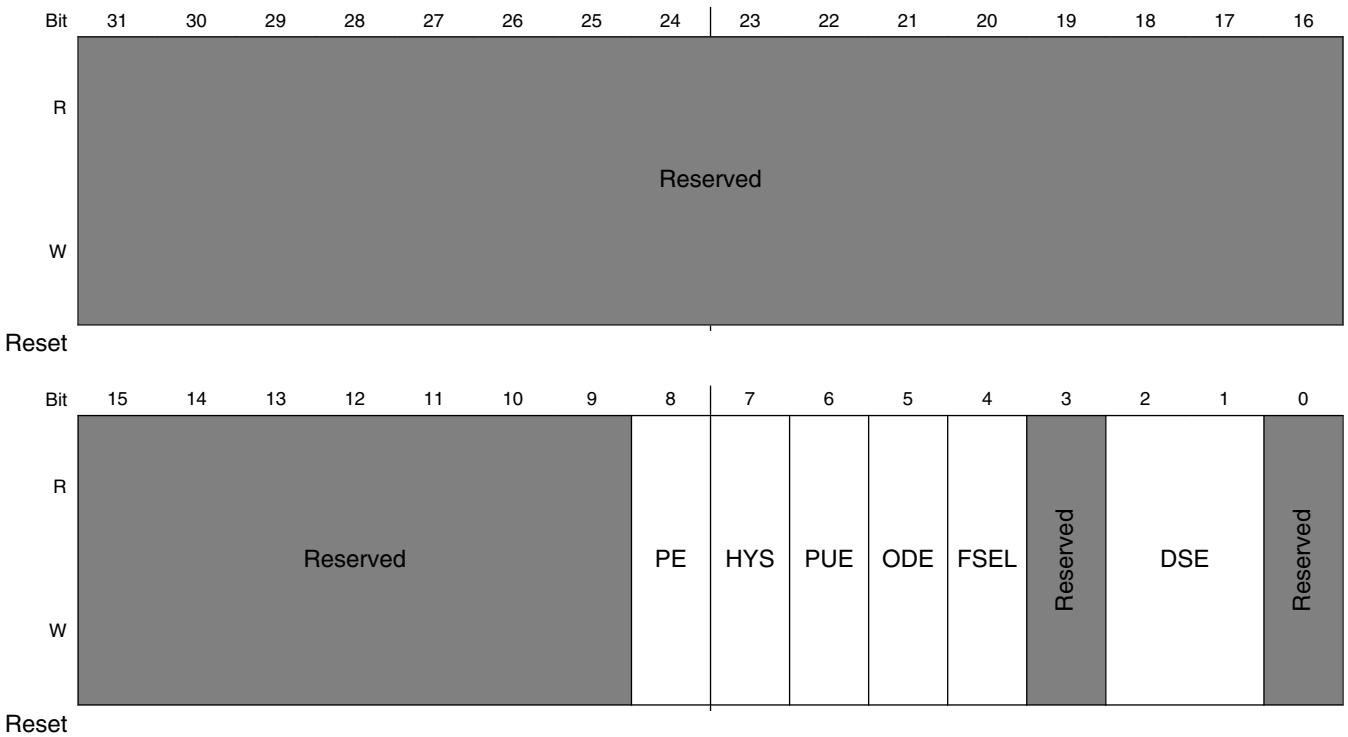
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO03 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO03 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO03 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO03 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.157 SW_PAD_CTL_PAD_GPIO1_IO04 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO04)

SW_PAD_CTL Register

Address: 3033_0000h base + 284h offset = 3033_0284h



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO04 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO04 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO04 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO04

Table continues on the next page...

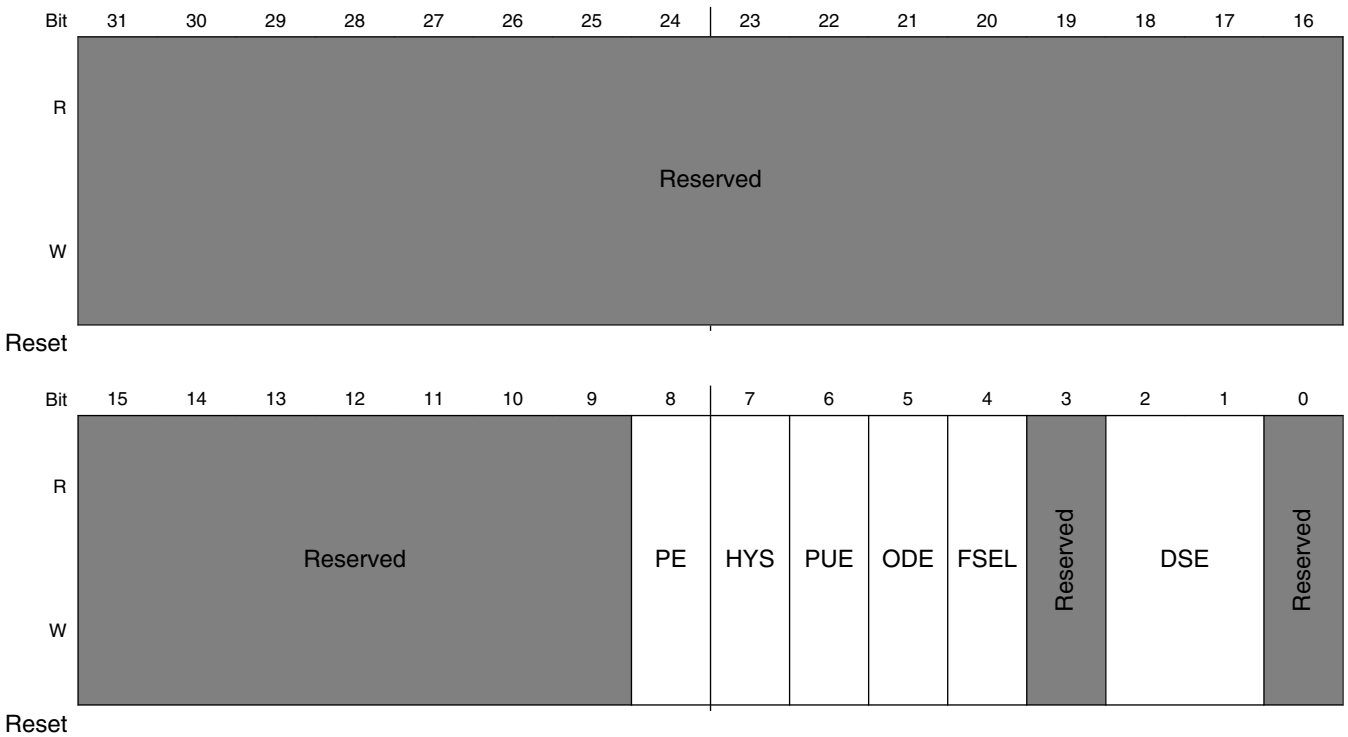
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO04 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO04 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO04 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO04 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.158 SW_PAD_CTL_PAD_GPIO1_IO05 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO05)

SW_PAD_CTL Register

Address: 3033_0000h base + 288h offset = 3033_0288h



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO05 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO05 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO05 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO05

Table continues on the next page...

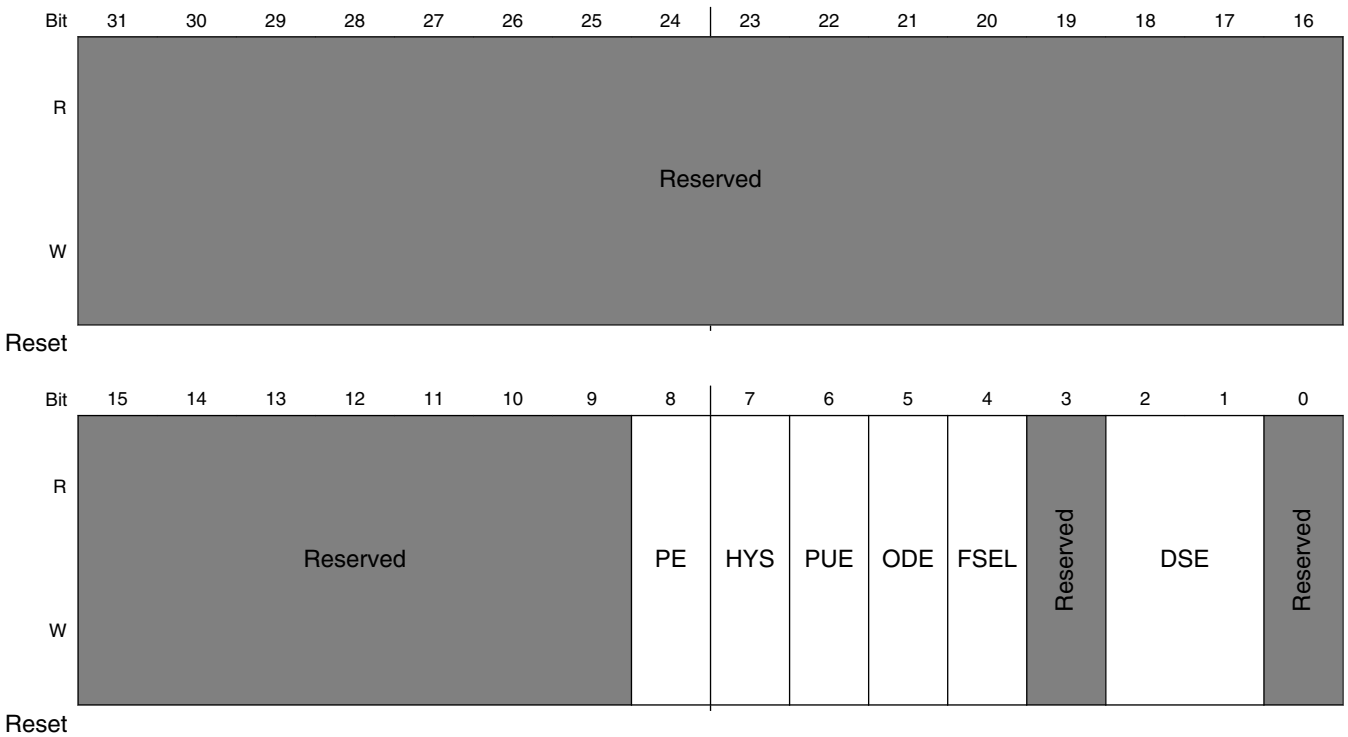
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO05 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO05 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO05 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO05 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.159 SW_PAD_CTL_PAD_GPIO1_IO06 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO06)

SW_PAD_CTL Register

Address: 3033_0000h base + 28Ch offset = 3033_028Ch



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO06 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO06 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO06 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO06

Table continues on the next page...

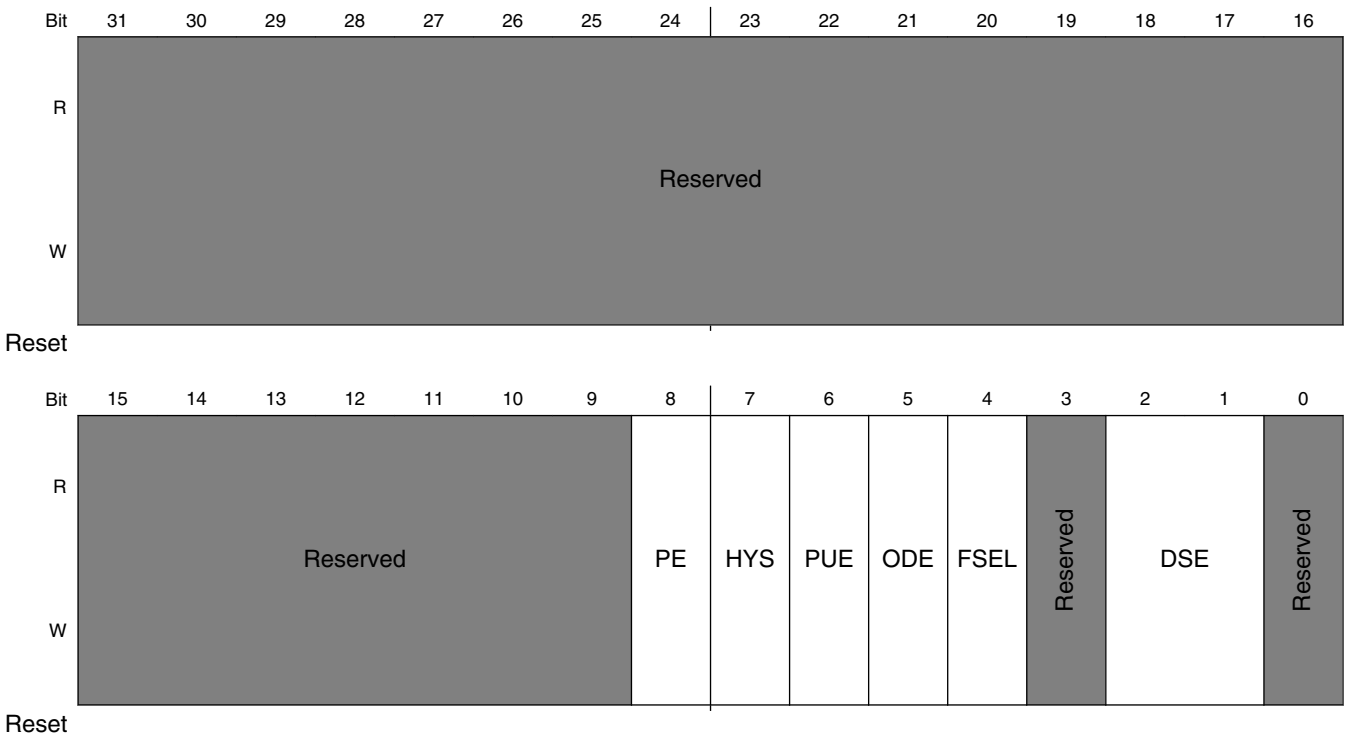
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO06 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO06 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO06 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO06 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.160 SW_PAD_CTL_PAD_GPIO1_IO07 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO07)

SW_PAD_CTL Register

Address: 3033_0000h base + 290h offset = 3033_0290h



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO07 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO07 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO07 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO07

Table continues on the next page...

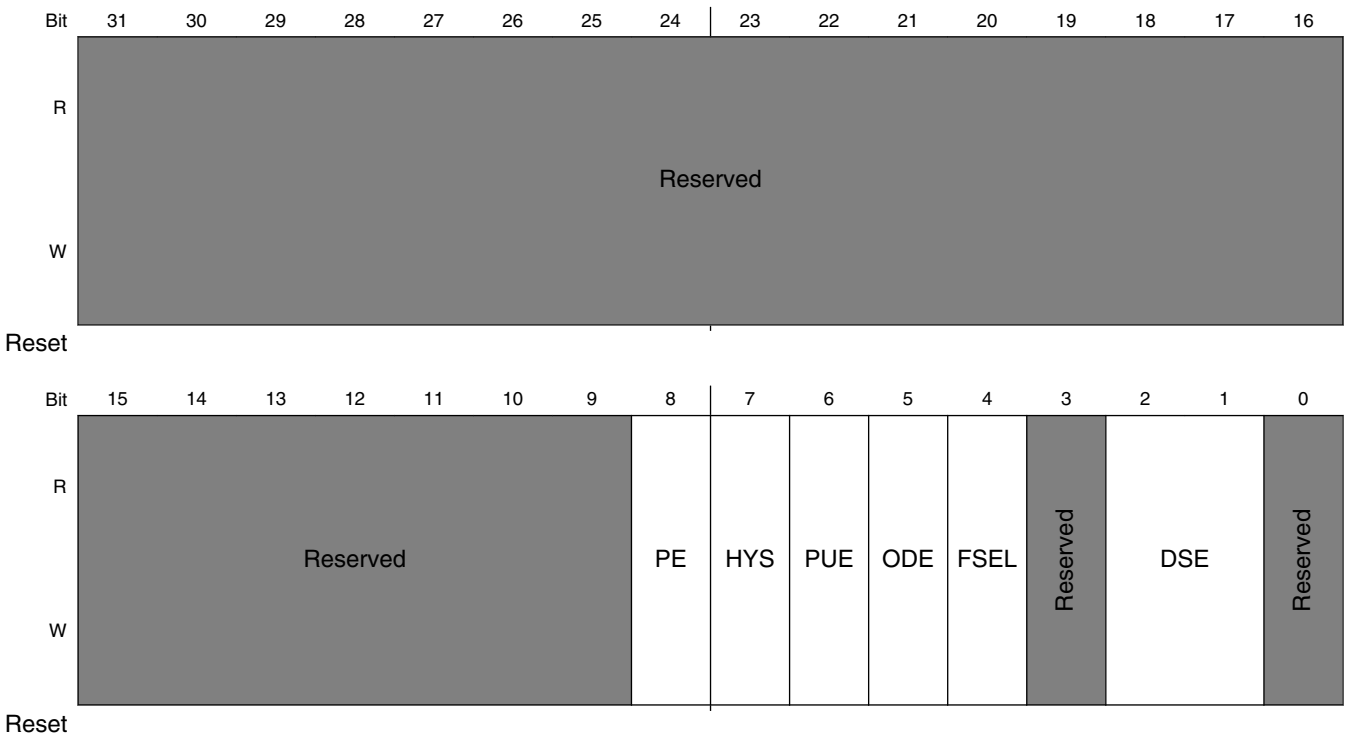
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO07 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO07 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO07 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO07 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.161 SW_PAD_CTL_PAD_GPIO1_IO08 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO08)

SW_PAD_CTL Register

Address: 3033_0000h base + 294h offset = 3033_0294h



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO08 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO08 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO08 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO08

Table continues on the next page...

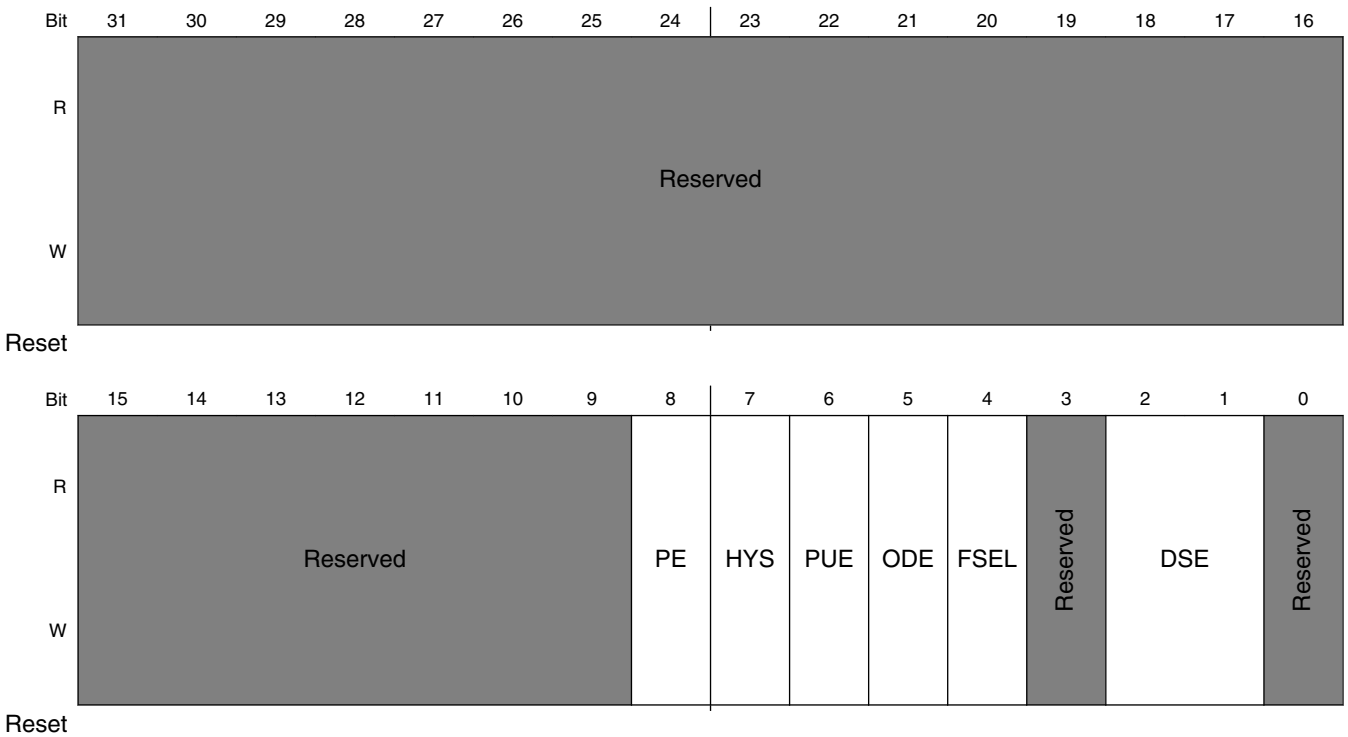
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO08 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO08 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO08 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO08 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.162 SW_PAD_CTL_PAD_GPIO1_IO09 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO09)

SW_PAD_CTL Register

Address: 3033_0000h base + 298h offset = 3033_0298h



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO09 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO09 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO09 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO09

Table continues on the next page...

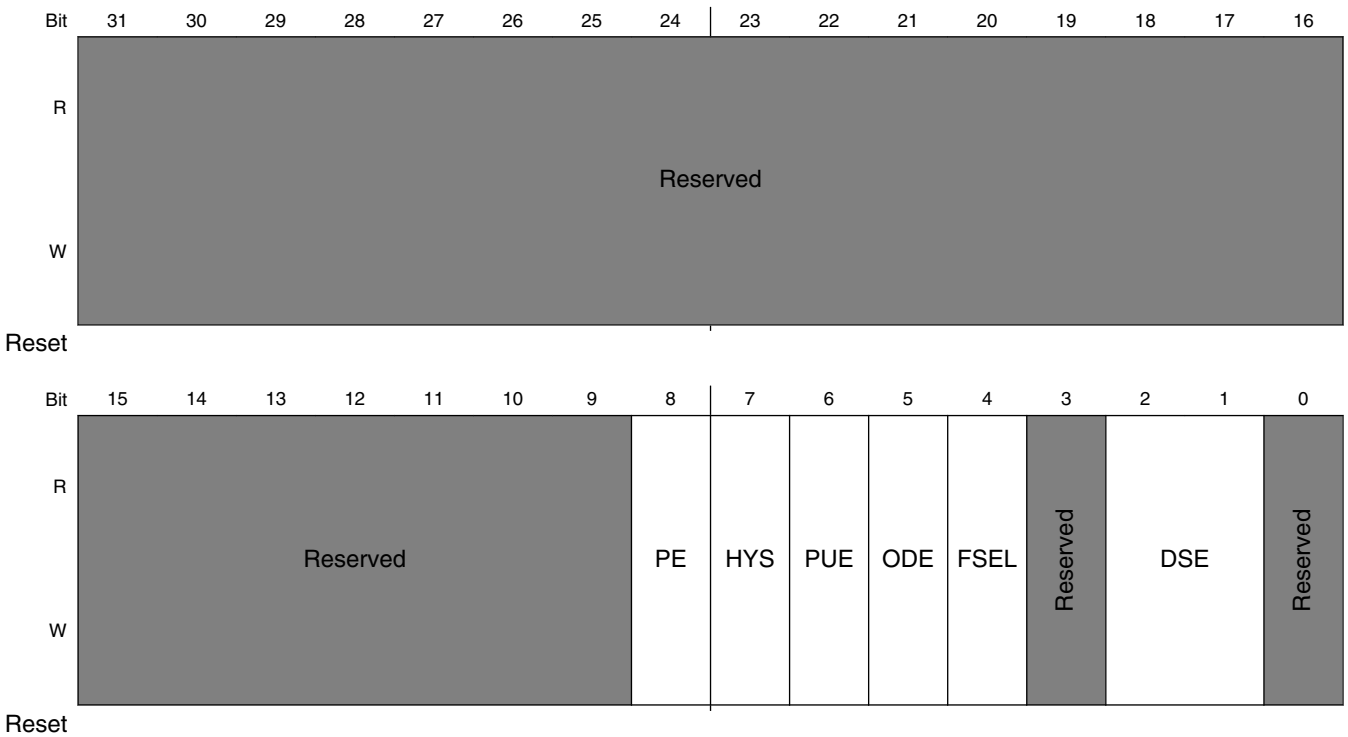
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO09 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO09 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO09 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO09 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.163 SW_PAD_CTL_PAD_GPIO1_IO10 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO10)

SW_PAD_CTL Register

Address: 3033_0000h base + 29Ch offset = 3033_029Ch



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO10 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO10 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO10 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO10

Table continues on the next page...

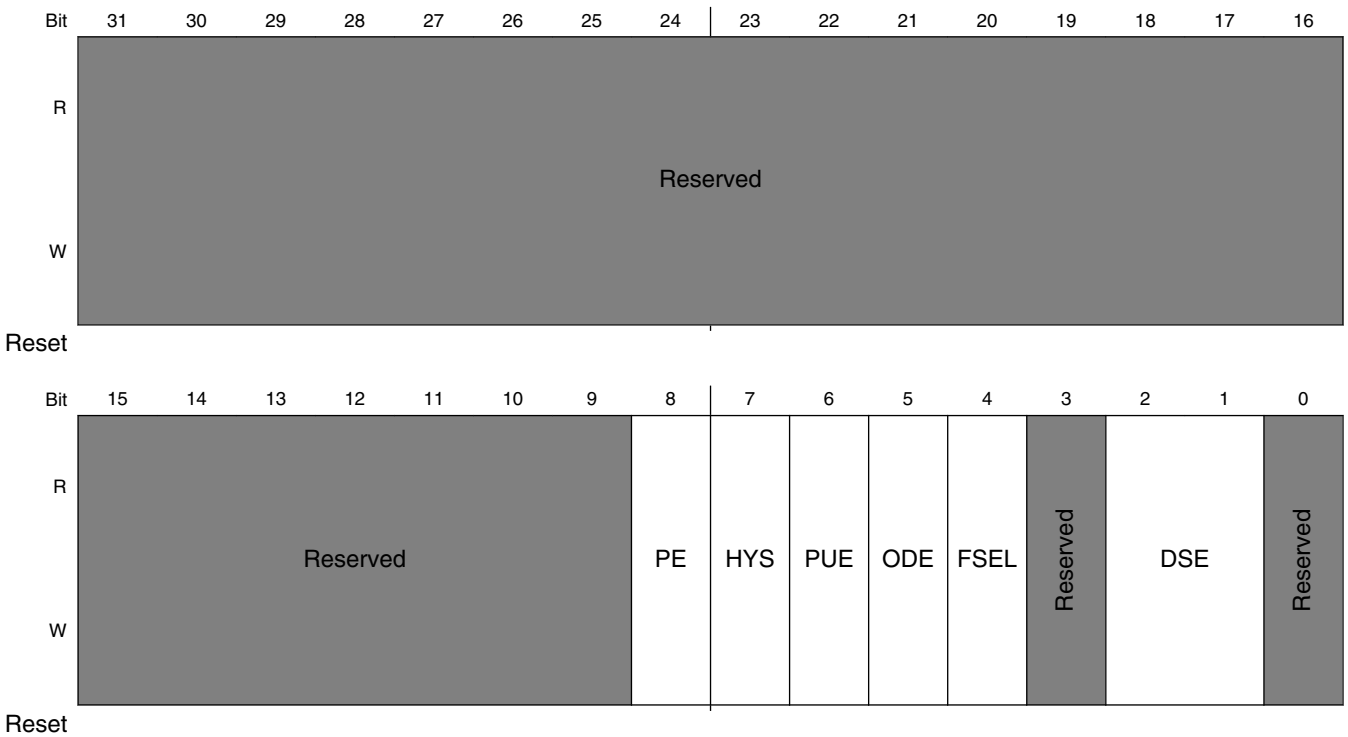
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO10 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO10 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO10 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO10 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.164 SW_PAD_CTL_PAD_GPIO1_IO11 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO11)

SW_PAD_CTL Register

Address: 3033_0000h base + 2A0h offset = 3033_02A0h



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO11 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO11 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO11 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO11

Table continues on the next page...

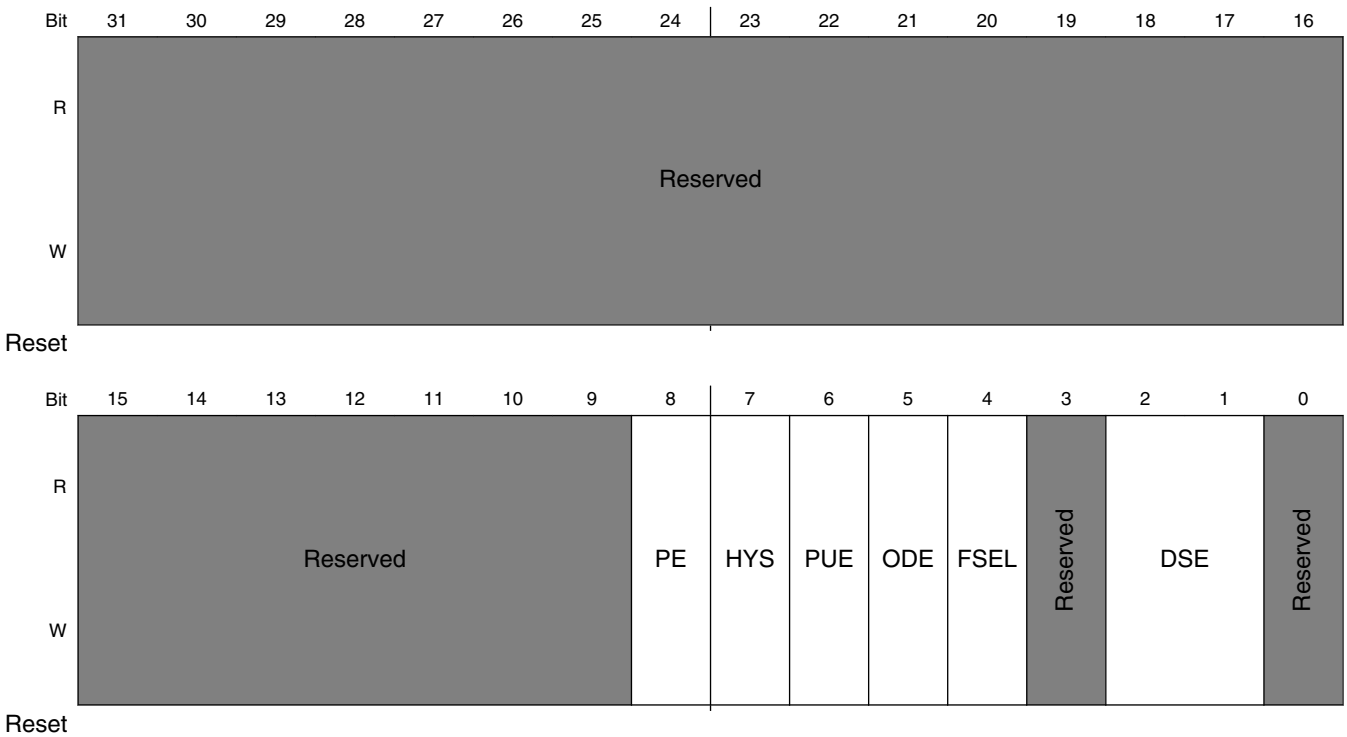
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO11 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO11 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO11 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO11 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.165 SW_PAD_CTL_PAD_GPIO1_IO12 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO12)

SW_PAD_CTL Register

Address: 3033_0000h base + 2A4h offset = 3033_02A4h



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO12 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO12 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO12 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO12

Table continues on the next page...

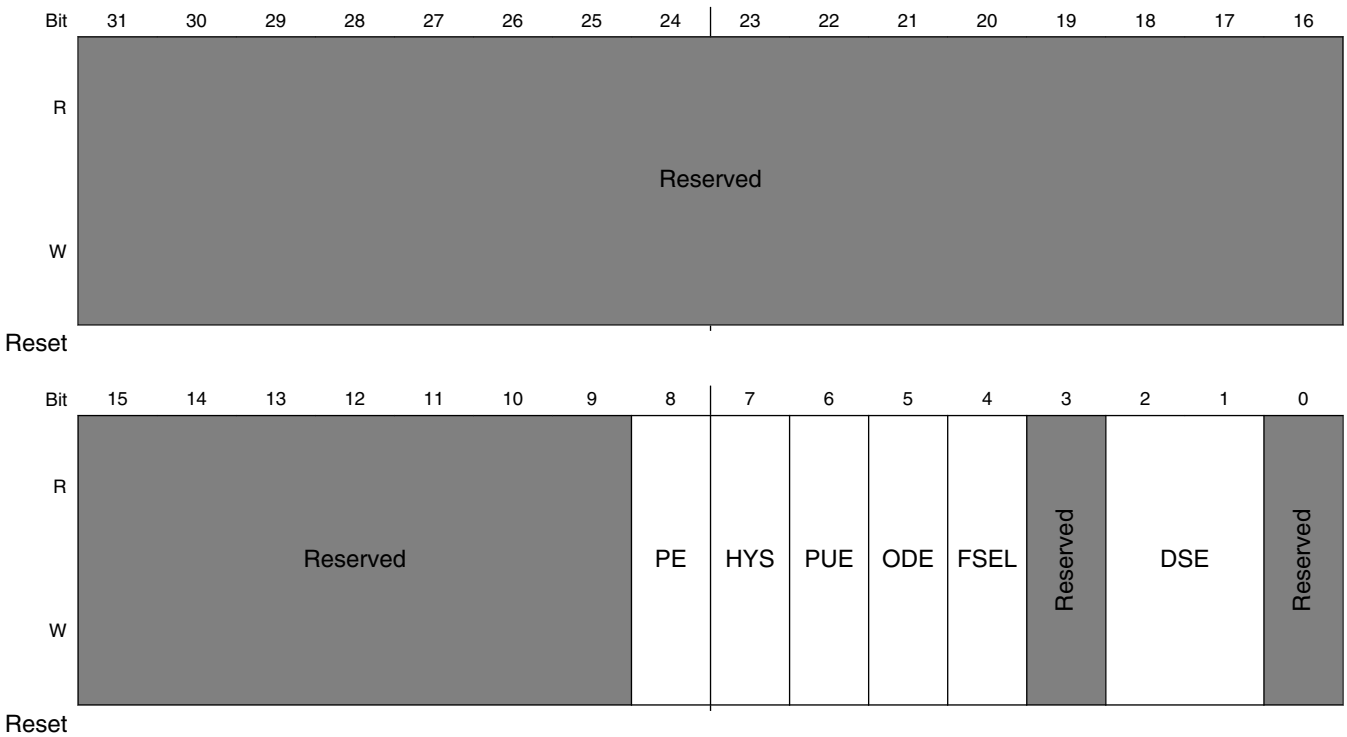
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO12 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO12 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO12 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO12 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.166 SW_PAD_CTL_PAD_GPIO1_IO13 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO13)

SW_PAD_CTL Register

Address: 3033_0000h base + 2A8h offset = 3033_02A8h



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO13 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO13 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO13 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO13

Table continues on the next page...

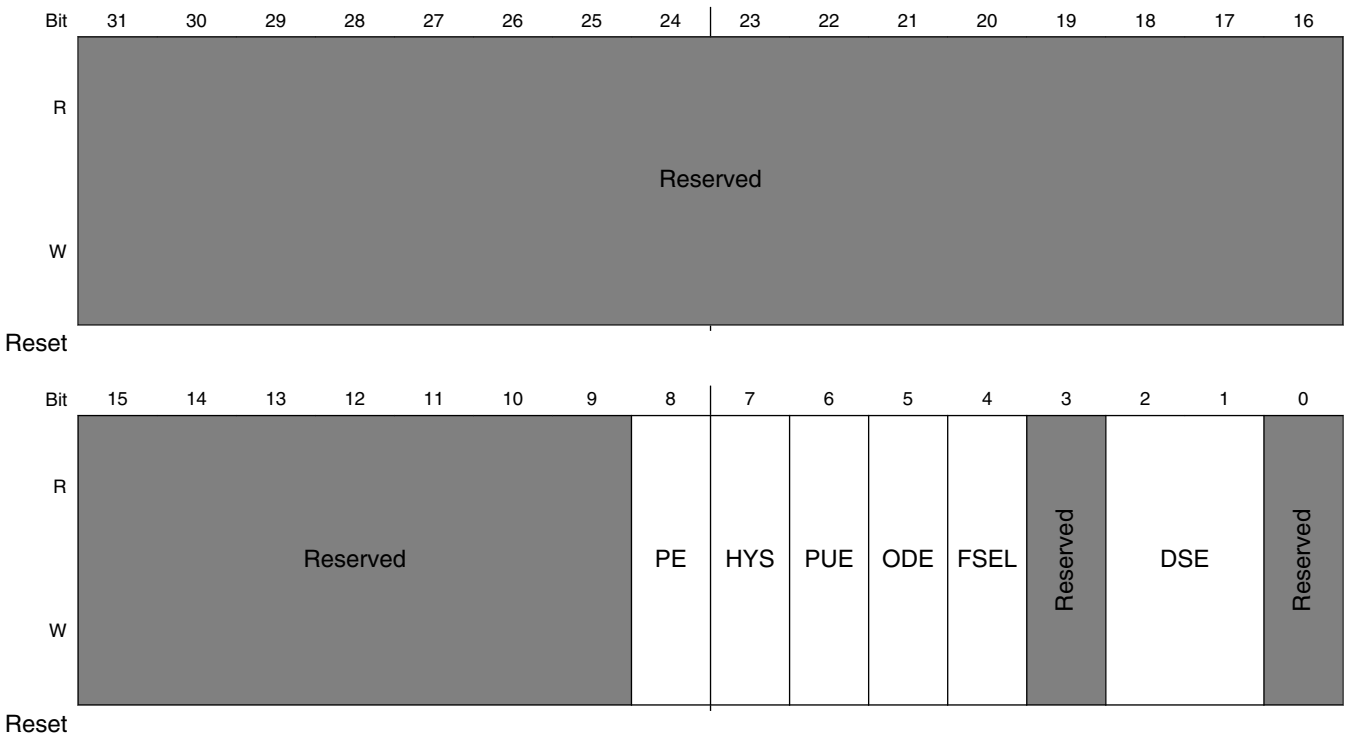
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO13 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO13 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO13 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO13 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.167 SW_PAD_CTL_PAD_GPIO1_IO14 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO14)

SW_PAD_CTL Register

Address: 3033_0000h base + 2ACh offset = 3033_02ACh



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO14 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO14 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO14 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO14

Table continues on the next page...

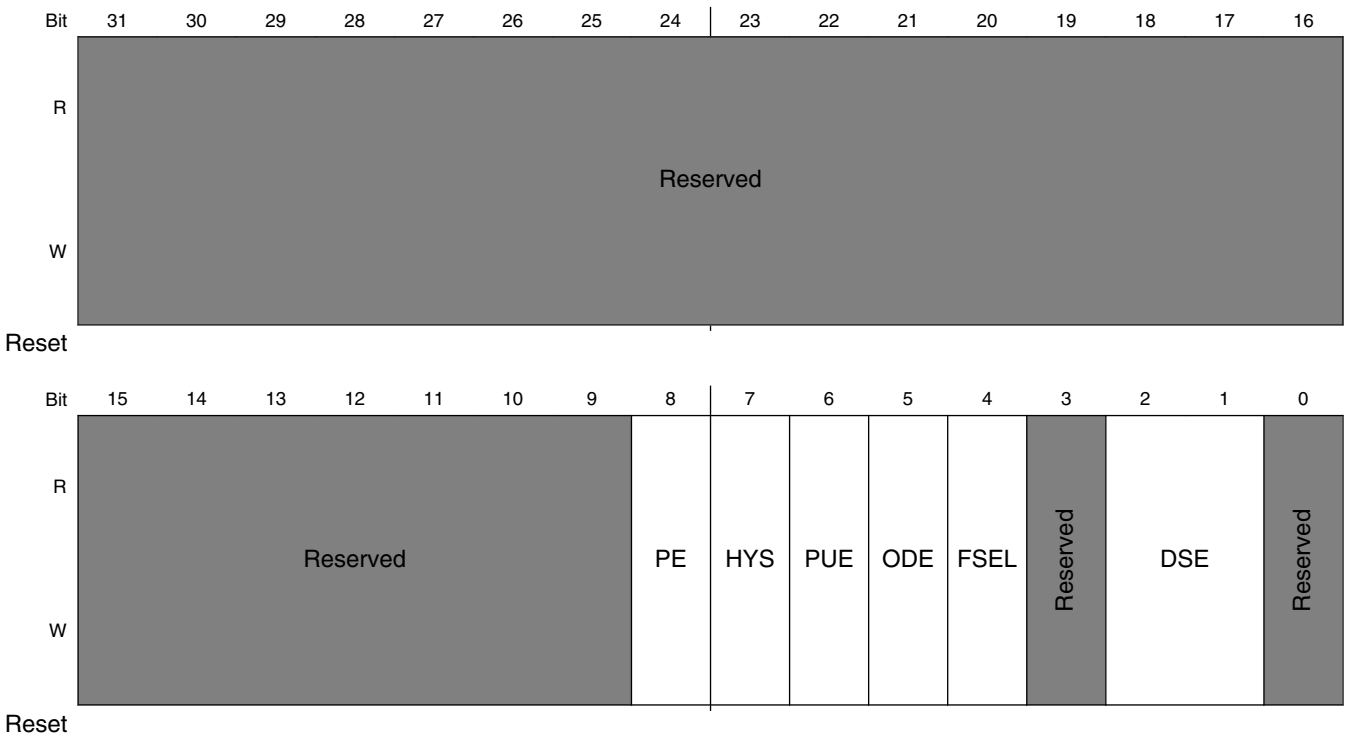
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO14 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO14 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO14 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO14 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.168 SW_PAD_CTL_PAD_GPIO1_IO15 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO15)

SW_PAD_CTL Register

Address: 3033_0000h base + 2B0h offset = 3033_02B0h



IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO15 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: GPIO1_IO15 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: GPIO1_IO15 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO15

Table continues on the next page...

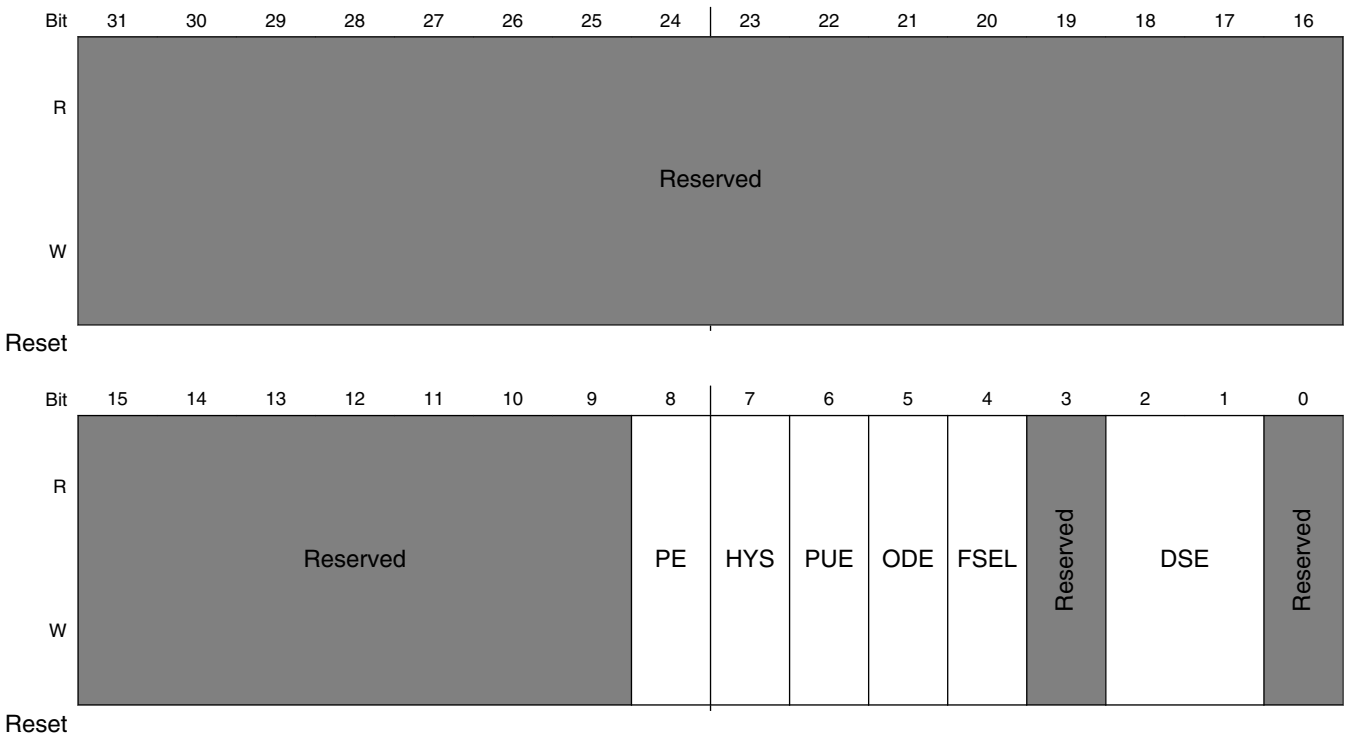
IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO15 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: GPIO1_IO15 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: GPIO1_IO15 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO15 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.169 SW_PAD_CTL_PAD_ENET_MDC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDC)

SW_PAD_CTL Register

Address: 3033_0000h base + 2B4h offset = 3033_02B4h



IOMUXC_SW_PAD_CTL_PAD_ENET_MDC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_MDC 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_MDC 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_MDC

Table continues on the next page...

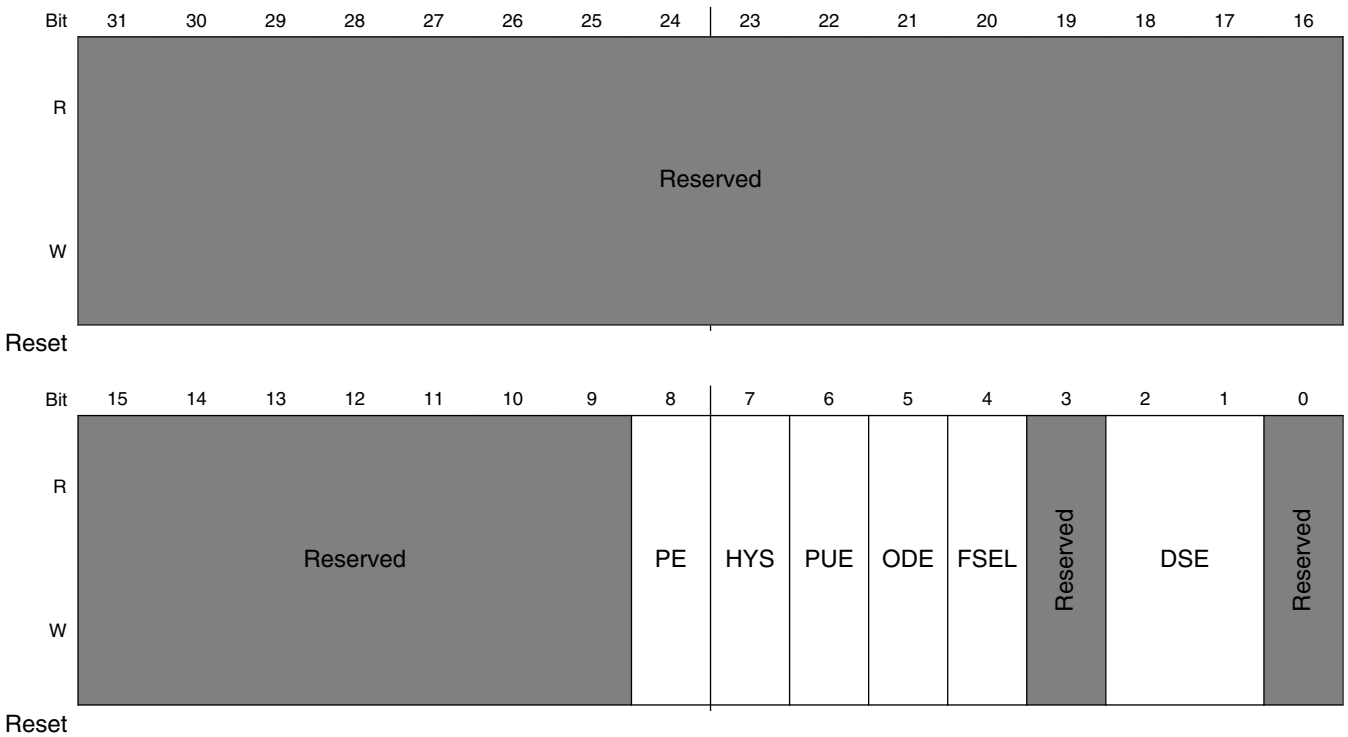
IOMUXC_SW_PAD_CTL_PAD_ENET_MDC field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_MDC 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_MDC 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_MDC 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.170 SW_PAD_CTL_PAD_ENET_MDIO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO)

SW_PAD_CTL Register

Address: 3033_0000h base + 2B8h offset = 3033_02B8h



IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_MDIO 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_MDIO 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_MDIO

Table continues on the next page...

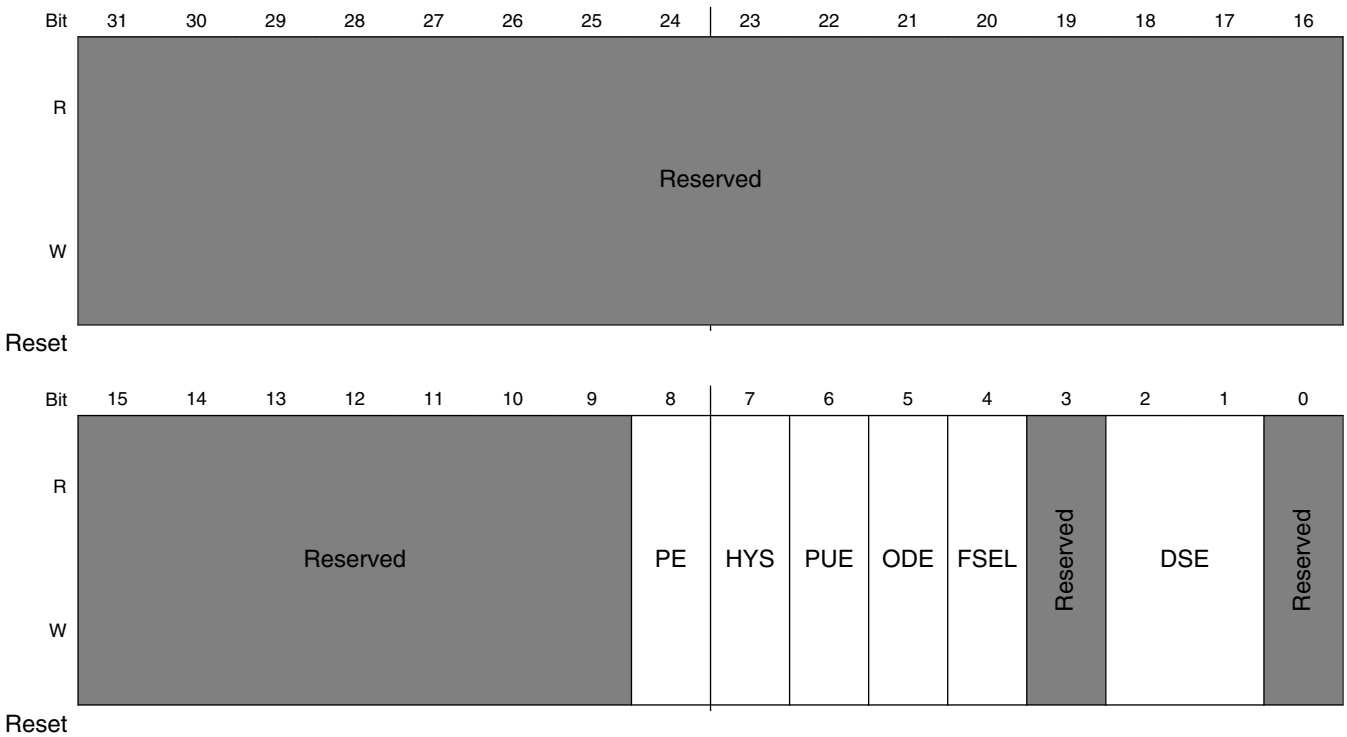
IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_MDIO 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_MDIO 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_MDIO 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.171 SW_PAD_CTL_PAD_ENET_TD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD3)

SW_PAD_CTL Register

Address: 3033_0000h base + 2BCh offset = 3033_02BCh



IOMUXC_SW_PAD_CTL_PAD_ENET_TD3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_TD3 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_TD3 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_TD3

Table continues on the next page...

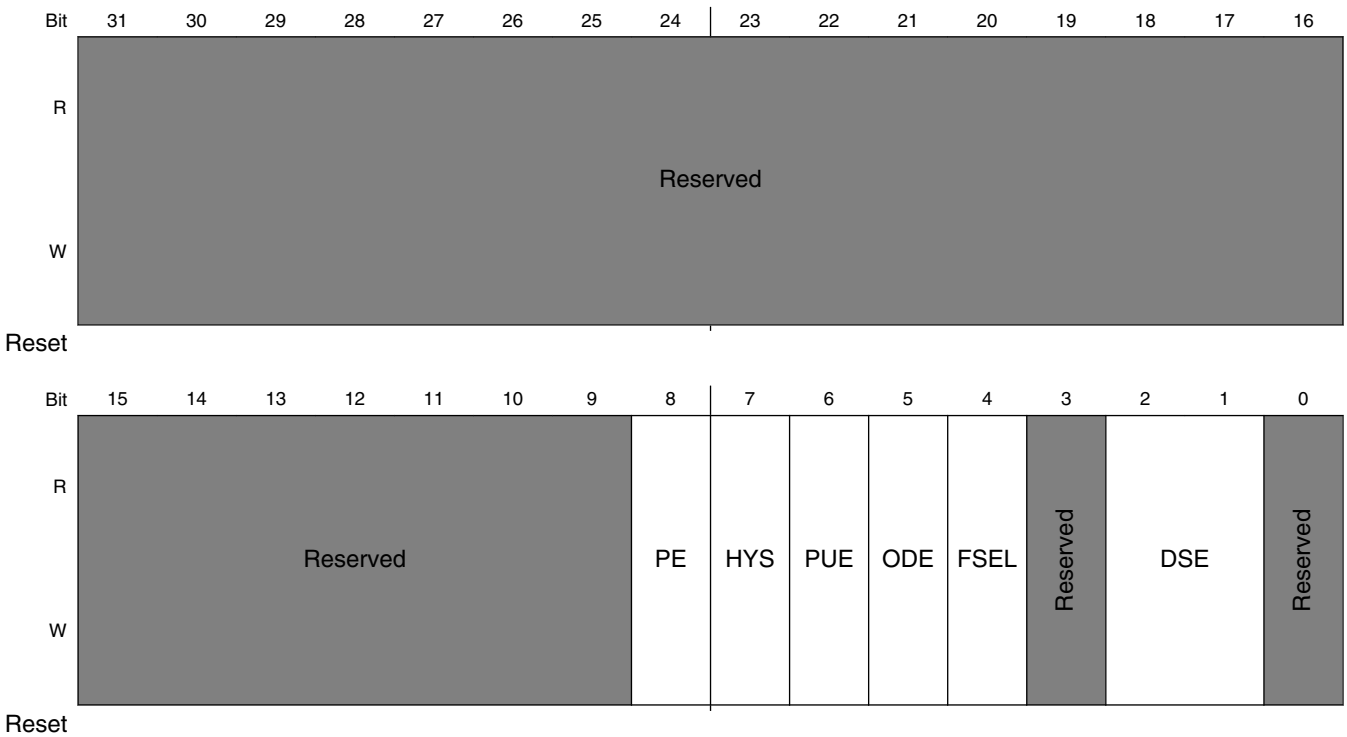
IOMUXC_SW_PAD_CTL_PAD_ENET_TD3 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_TD3 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_TD3 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_TD3 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.172 SW_PAD_CTL_PAD_ENET_TD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD2)

SW_PAD_CTL Register

Address: 3033_0000h base + 2C0h offset = 3033_02C0h



IOMUXC_SW_PAD_CTL_PAD_ENET_TD2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_TD2 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_TD2 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_TD2

Table continues on the next page...

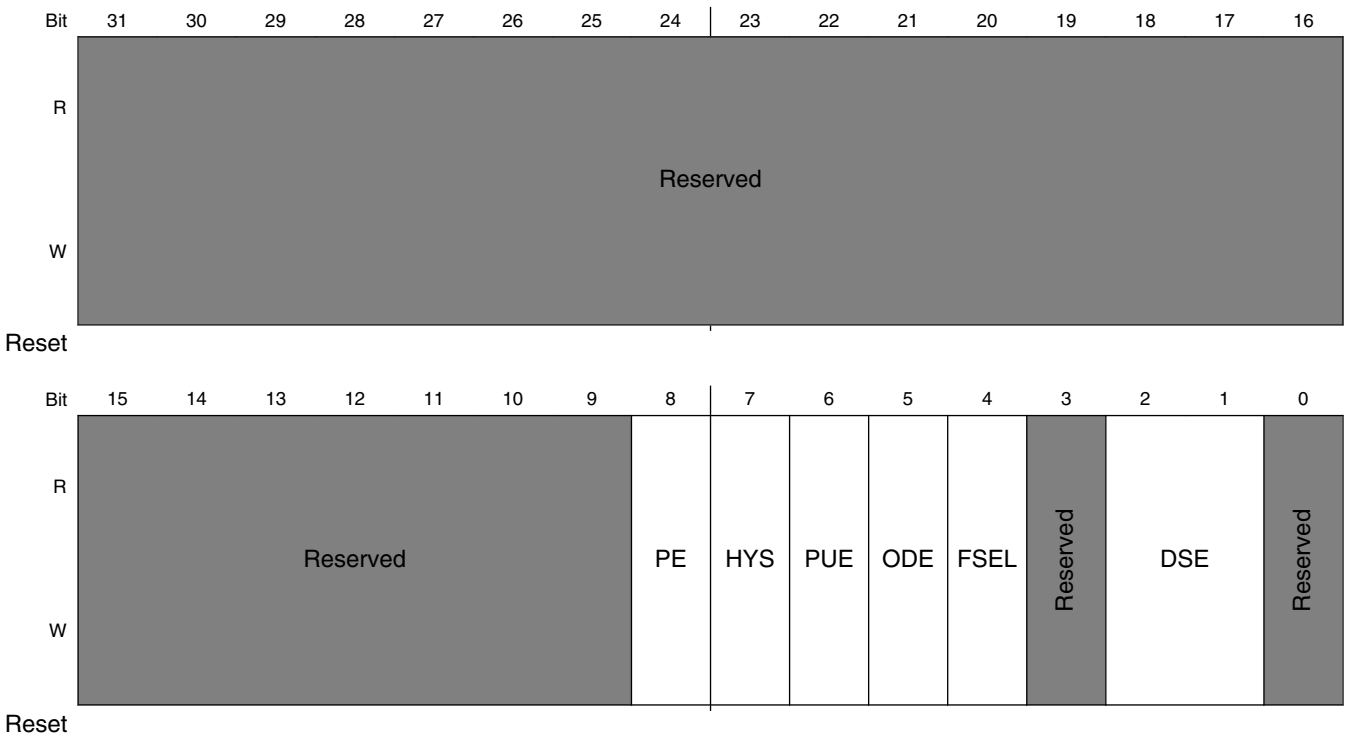
IOMUXC_SW_PAD_CTL_PAD_ENET_TD2 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_TD2 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_TD2 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_TD2 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.173 SW_PAD_CTL_PAD_ENET_TD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD1)

SW_PAD_CTL Register

Address: 3033_0000h base + 2C4h offset = 3033_02C4h



IOMUXC_SW_PAD_CTL_PAD_ENET_TD1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_TD1 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_TD1 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_TD1

Table continues on the next page...

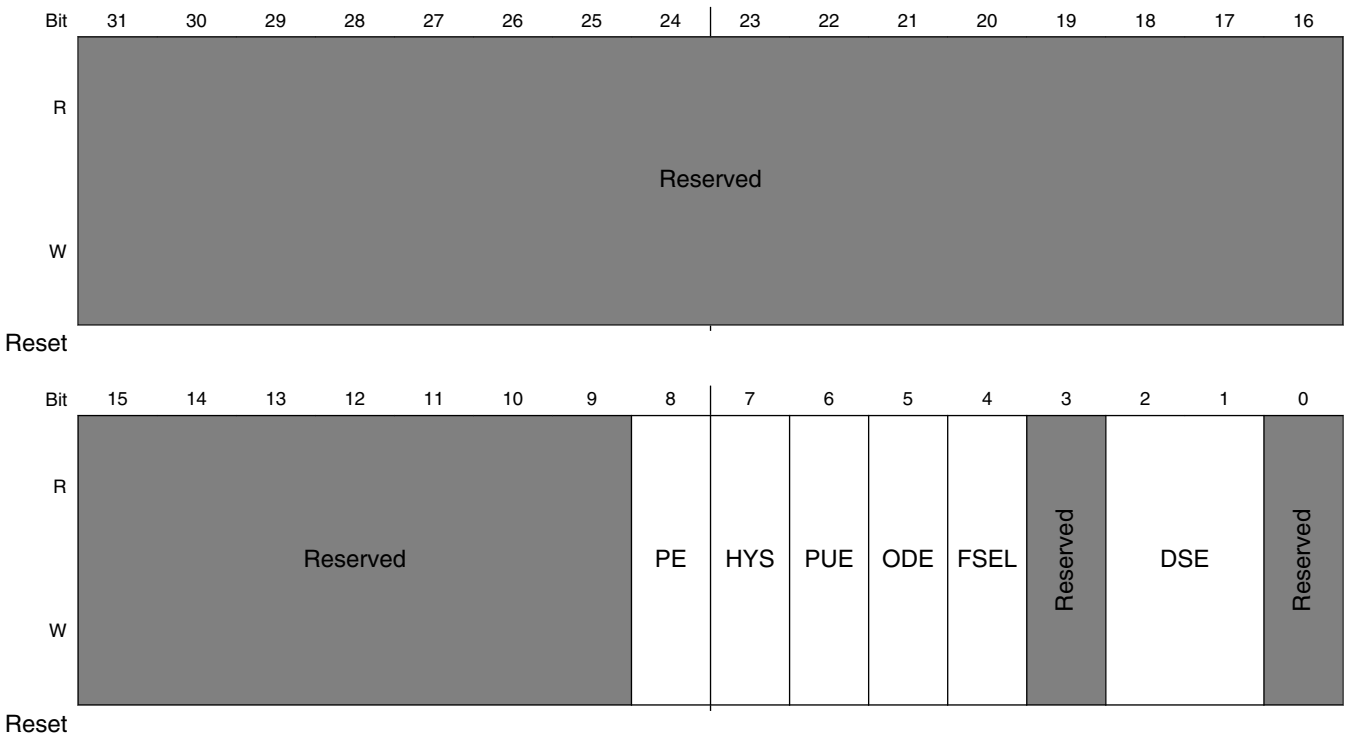
IOMUXC_SW_PAD_CTL_PAD_ENET_TD1 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_TD1 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_TD1 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_TD1 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.174 SW_PAD_CTL_PAD_ENET_TD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 2C8h offset = 3033_02C8h



IOMUXC_SW_PAD_CTL_PAD_ENET_TD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_TD0 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_TD0 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_TD0

Table continues on the next page...

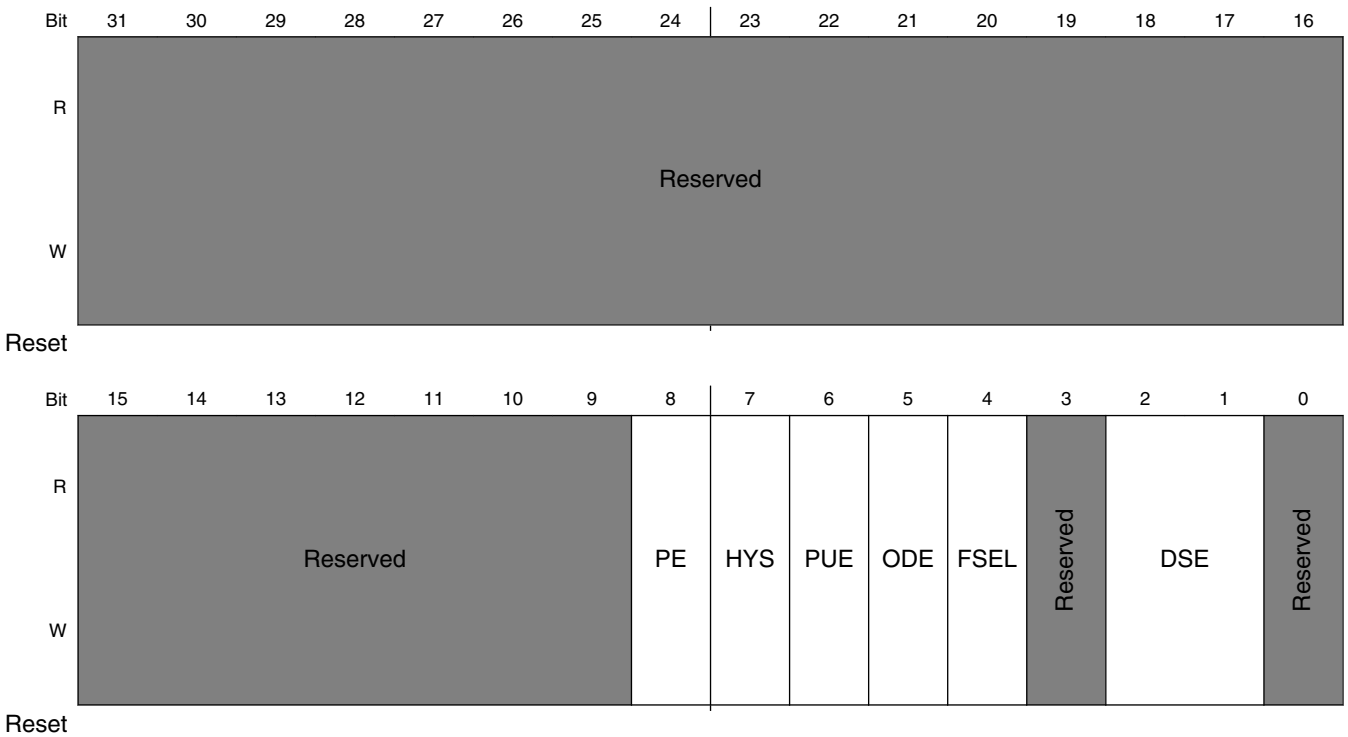
IOMUXC_SW_PAD_CTL_PAD_ENET_TD0 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_TD0 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_TD0 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_TD0 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.175 SW_PAD_CTL_PAD_ENET_TX_CTL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TX_CTL)

SW_PAD_CTL Register

Address: 3033_0000h base + 2CCh offset = 3033_02CCh



IOMUXC_SW_PAD_CTL_PAD_ENET_TX_CTL field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_TX_CTL 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_TX_CTL 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_TX_CTL

Table continues on the next page...

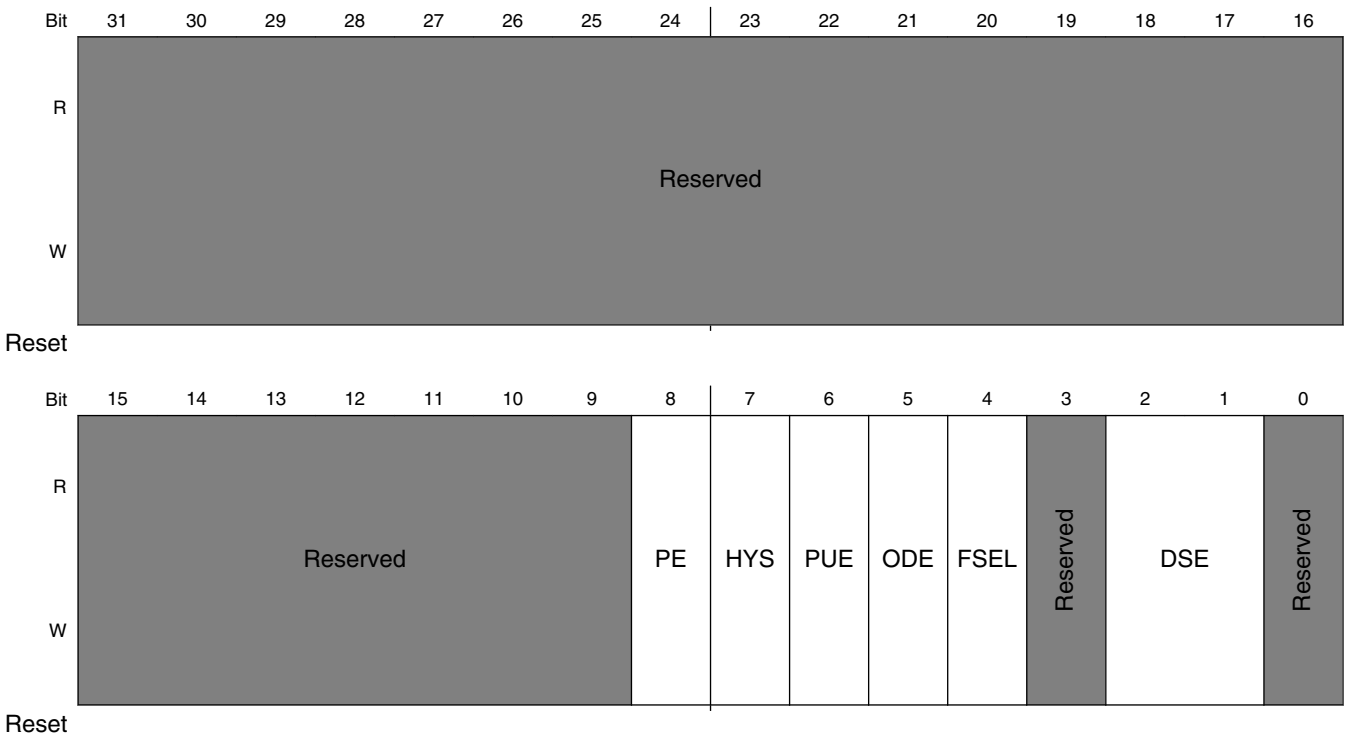
IOMUXC_SW_PAD_CTL_PAD_ENET_TX_CTL field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_TX_CTL 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_TX_CTL 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_TX_CTL 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.176 SW_PAD_CTL_PAD_ENET_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 2D0h offset = 3033_02D0h



IOMUXC_SW_PAD_CTL_PAD_ENET_TXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_TXC 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_TXC 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_TXC

Table continues on the next page...

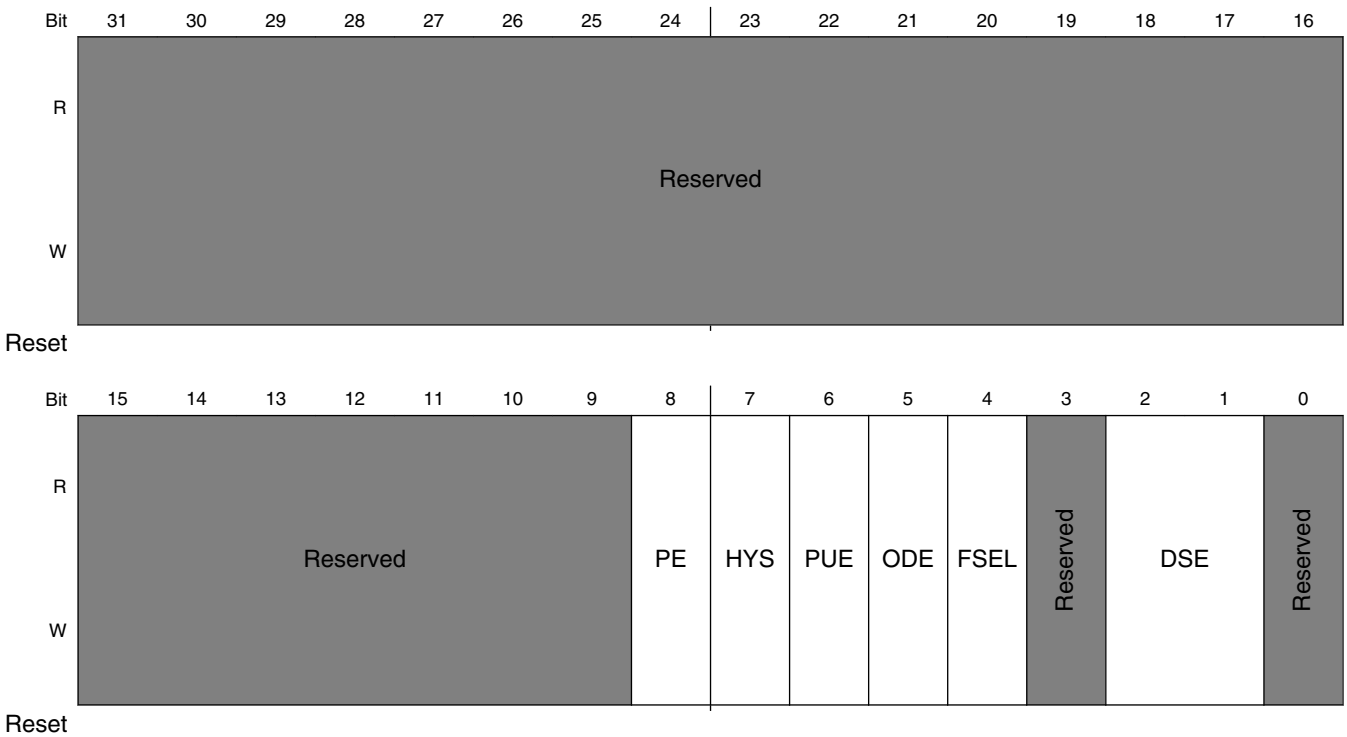
IOMUXC_SW_PAD_CTL_PAD_ENET_TXC field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_TXC 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_TXC 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_TXC 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.177 SW_PAD_CTL_PAD_ENET_RX_CTL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RX_CTL)

SW_PAD_CTL Register

Address: 3033_0000h base + 2D4h offset = 3033_02D4h



IOMUXC_SW_PAD_CTL_PAD_ENET_RX_CTL field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_RX_CTL 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_RX_CTL 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_RX_CTL

Table continues on the next page...

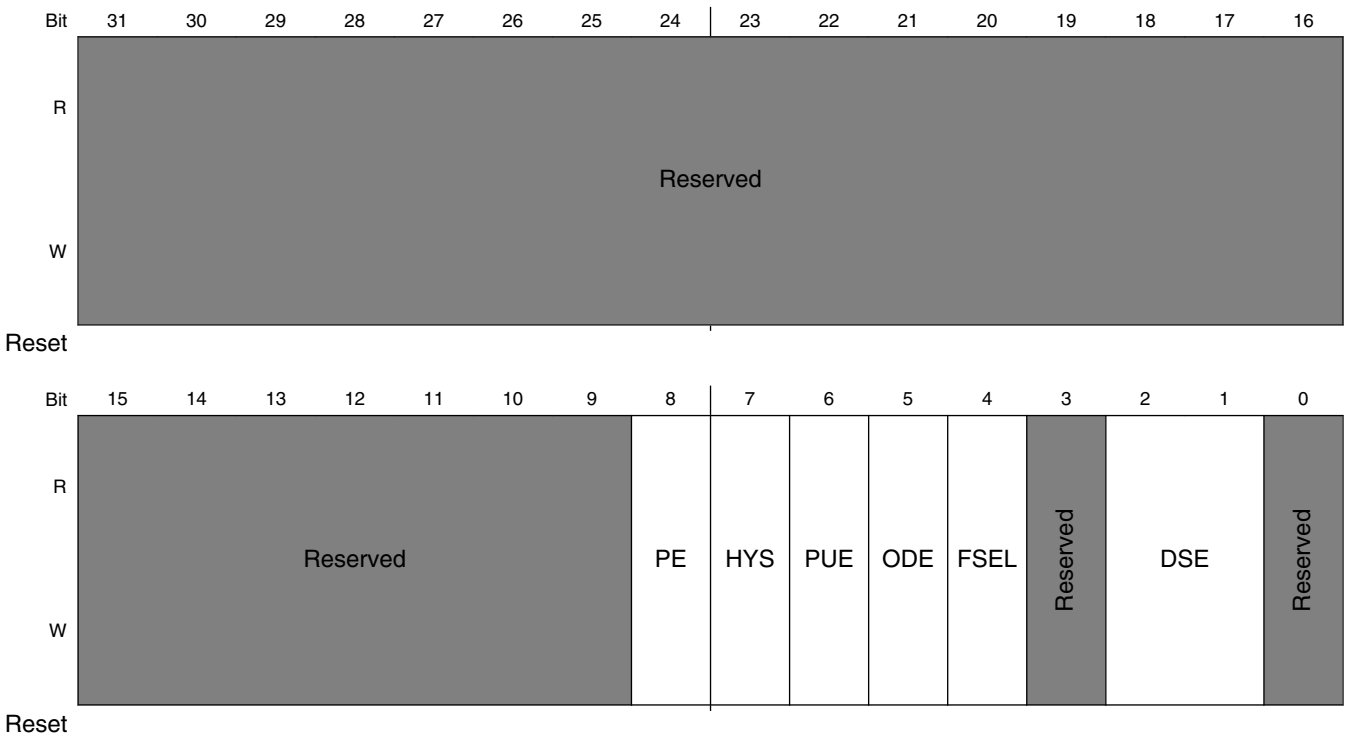
IOMUXC_SW_PAD_CTL_PAD_ENET_RX_CTL field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_RX_CTL 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_RX_CTL 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_RX_CTL 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.178 SW_PAD_CTL_PAD_ENET_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 2D8h offset = 3033_02D8h



IOMUXC_SW_PAD_CTL_PAD_ENET_RXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_RXC 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_RXC 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_RXC

Table continues on the next page...

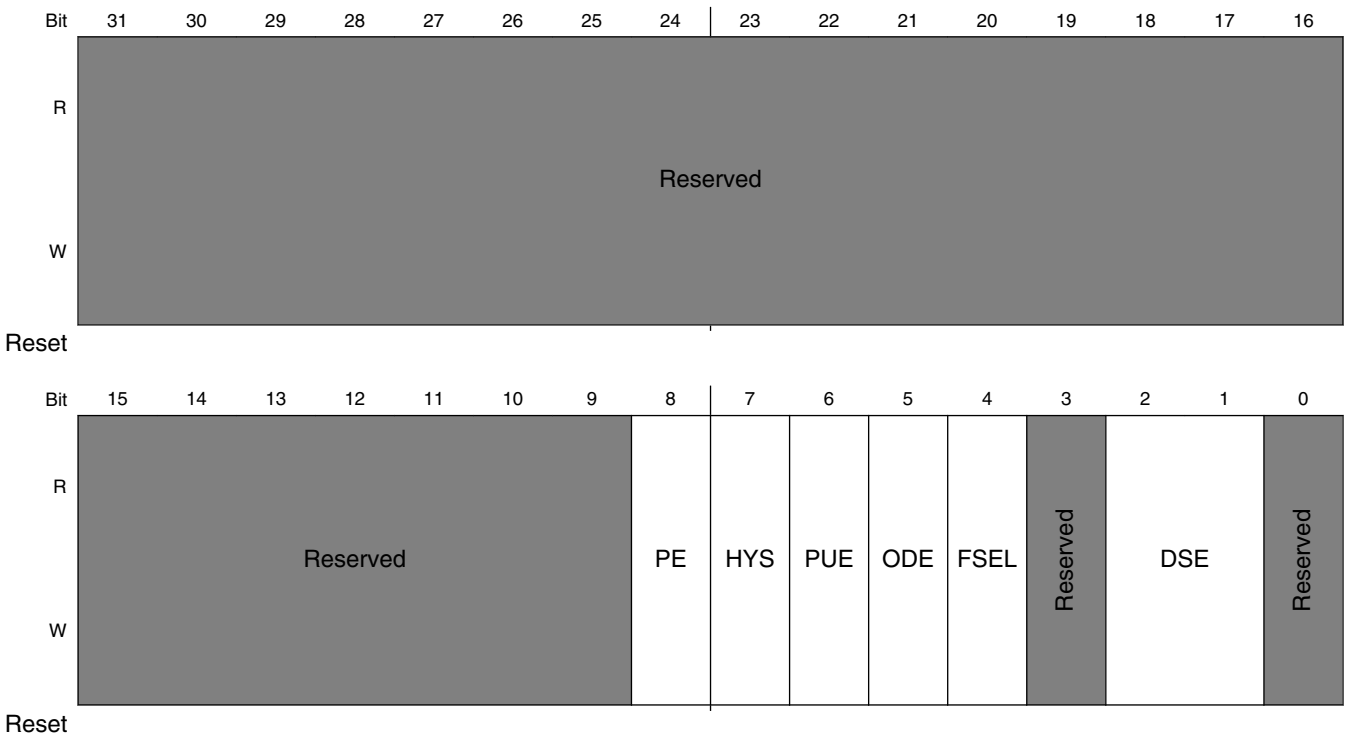
IOMUXC_SW_PAD_CTL_PAD_ENET_RXC field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_RXC 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_RXC 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_RXC 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.179 SW_PAD_CTL_PAD_ENET_RD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 2DCh offset = 3033_02DCh



IOMUXC_SW_PAD_CTL_PAD_ENET_RD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_RD0 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_RD0 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_RD0

Table continues on the next page...

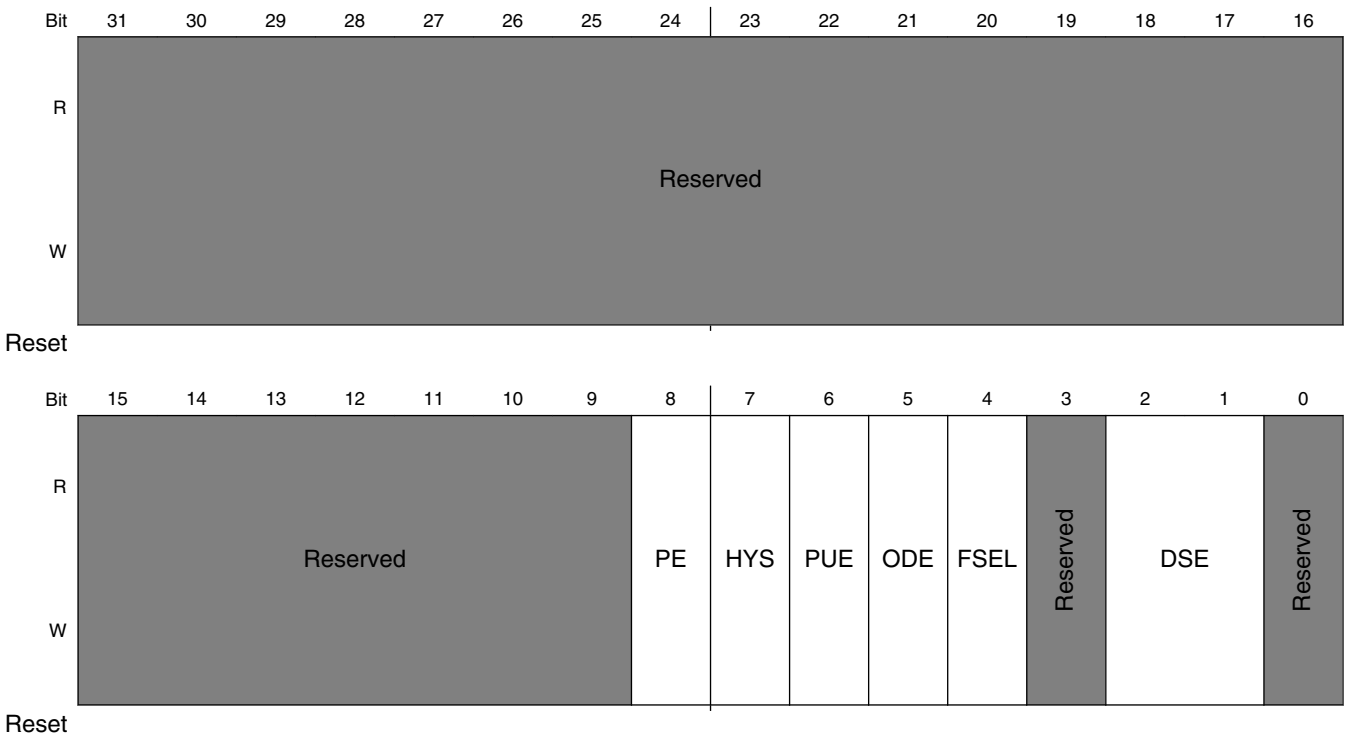
IOMUXC_SW_PAD_CTL_PAD_ENET_RD0 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_RD0 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_RD0 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_RD0 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.180 SW_PAD_CTL_PAD_ENET_RD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD1)

SW_PAD_CTL Register

Address: 3033_0000h base + 2E0h offset = 3033_02E0h



IOMUXC_SW_PAD_CTL_PAD_ENET_RD1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_RD1 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_RD1 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_RD1

Table continues on the next page...

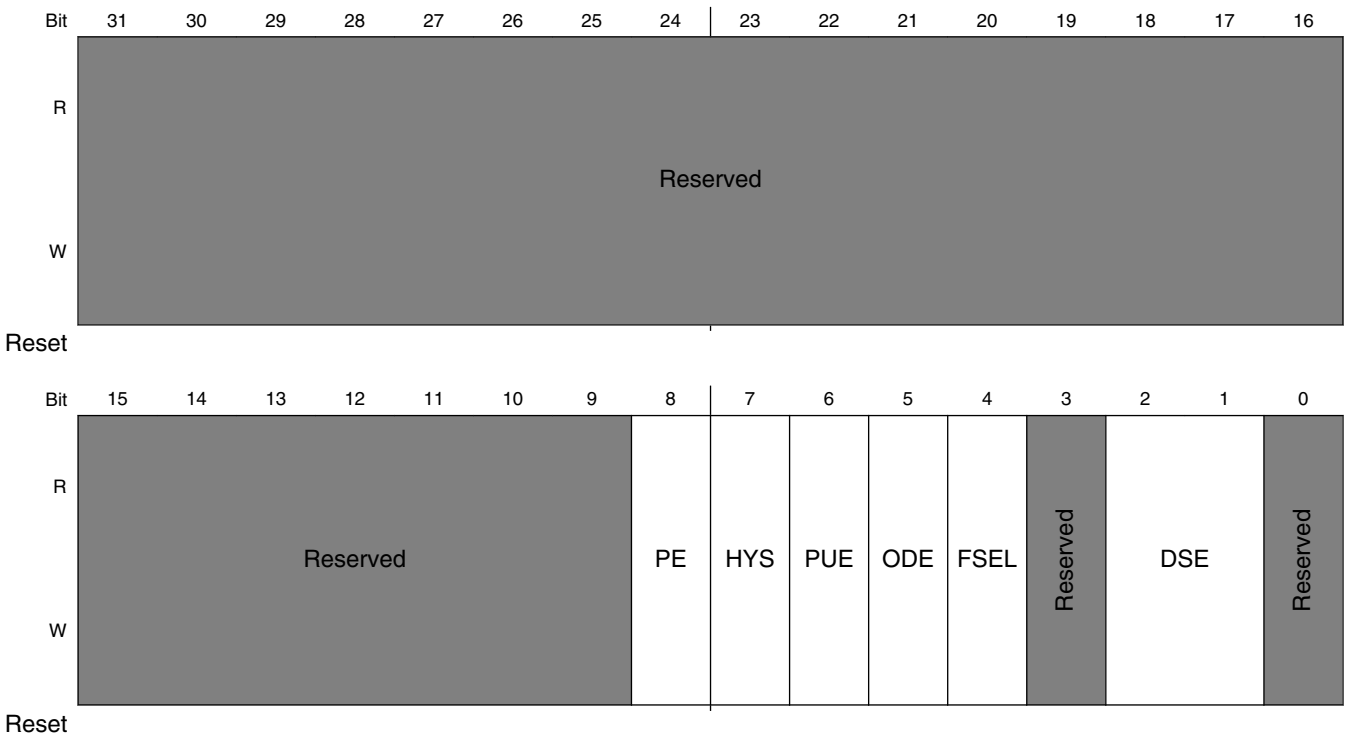
IOMUXC_SW_PAD_CTL_PAD_ENET_RD1 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_RD1 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_RD1 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_RD1 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.181 SW_PAD_CTL_PAD_ENET_RD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD2)

SW_PAD_CTL Register

Address: 3033_0000h base + 2E4h offset = 3033_02E4h



IOMUXC_SW_PAD_CTL_PAD_ENET_RD2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_RD2 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_RD2 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_RD2

Table continues on the next page...

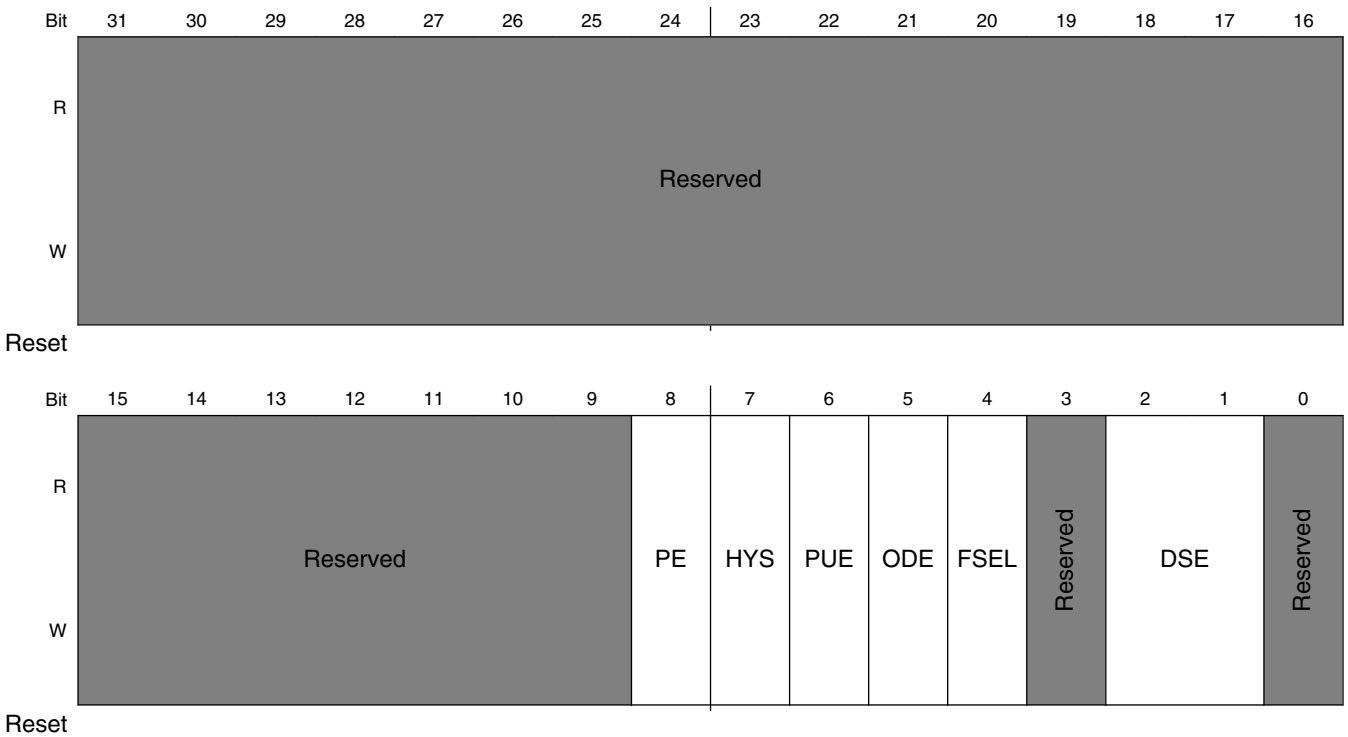
IOMUXC_SW_PAD_CTL_PAD_ENET_RD2 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_RD2 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_RD2 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_RD2 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.182 SW_PAD_CTL_PAD_ENET_RD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD3)

SW_PAD_CTL Register

Address: 3033_0000h base + 2E8h offset = 3033_02E8h



IOMUXC_SW_PAD_CTL_PAD_ENET_RD3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ENET_RD3 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ENET_RD3 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ENET_RD3

Table continues on the next page...

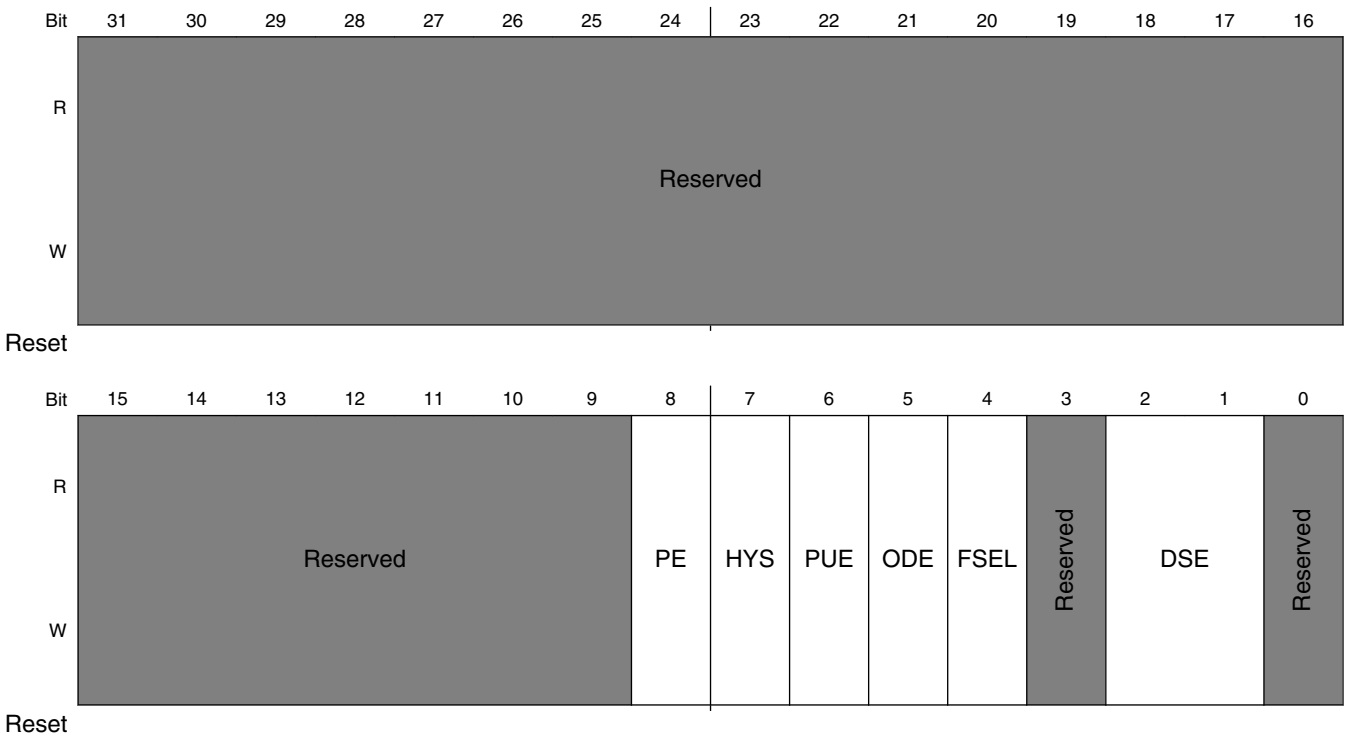
IOMUXC_SW_PAD_CTL_PAD_ENET_RD3 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ENET_RD3 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ENET_RD3 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ENET_RD3 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.183 SW_PAD_CTL_PAD_SD1_CLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 2ECh offset = 3033_02ECh



IOMUXC_SW_PAD_CTL_PAD_SD1_CLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD1_CLK 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD1_CLK 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD1_CLK

Table continues on the next page...

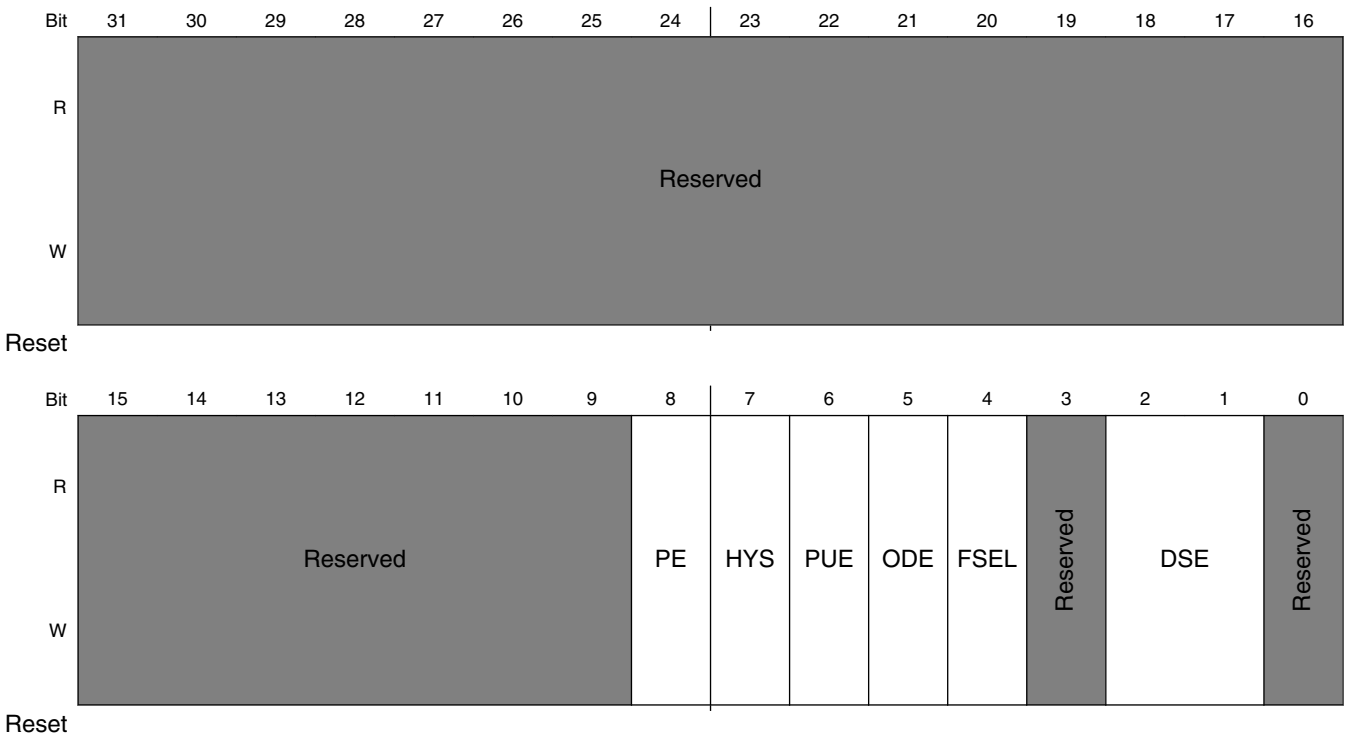
IOMUXC_SW_PAD_CTL_PAD_SD1_CLK field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD1_CLK 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD1_CLK 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_CLK 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.184 SW_PAD_CTL_PAD_SD1_CMD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CMD)

SW_PAD_CTL Register

Address: 3033_0000h base + 2F0h offset = 3033_02F0h



IOMUXC_SW_PAD_CTL_PAD_SD1_CMD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD1_CMD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD1_CMD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD1_CMD

Table continues on the next page...

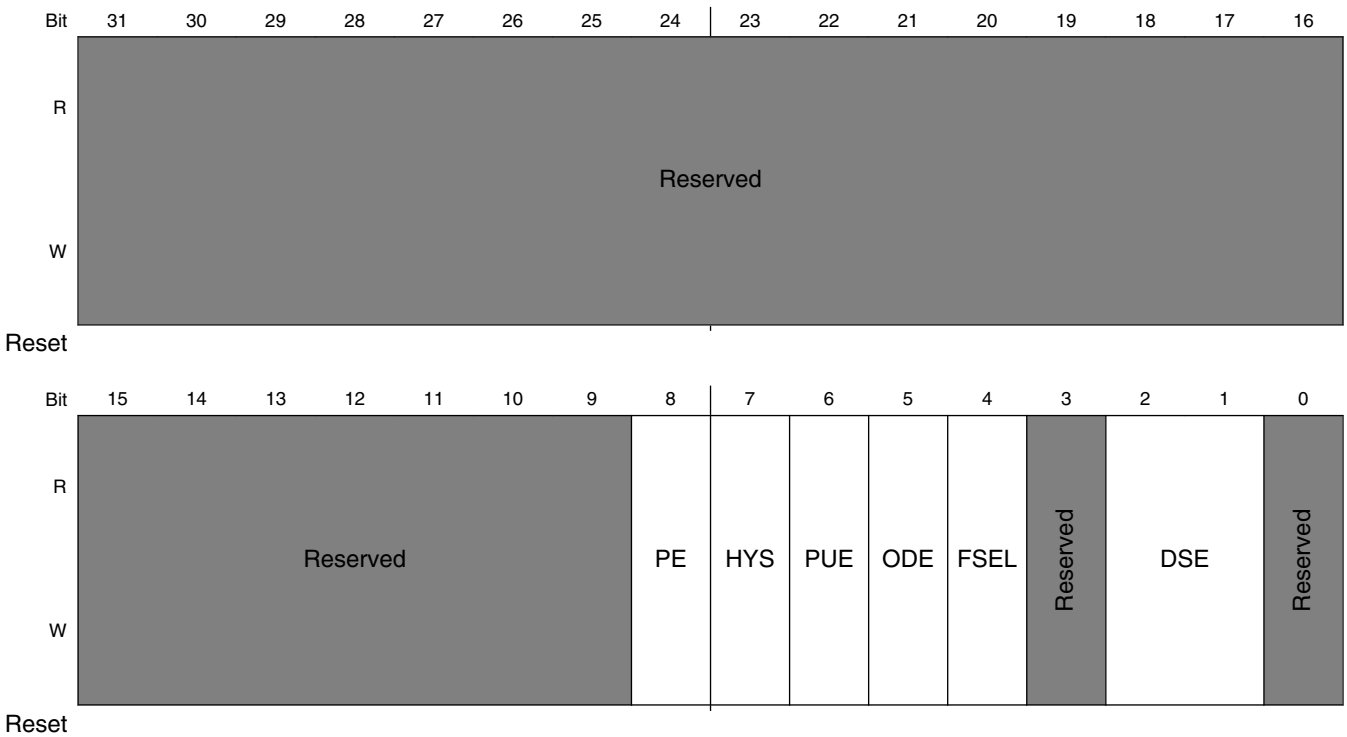
IOMUXC_SW_PAD_CTL_PAD_SD1_CMD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD1_CMD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD1_CMD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_CMD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.185 SW_PAD_CTL_PAD_SD1_DATA0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0)

SW_PAD_CTL Register

Address: 3033_0000h base + 2F4h offset = 3033_02F4h



IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD1_DATA0 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD1_DATA0 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD1_DATA0

Table continues on the next page...

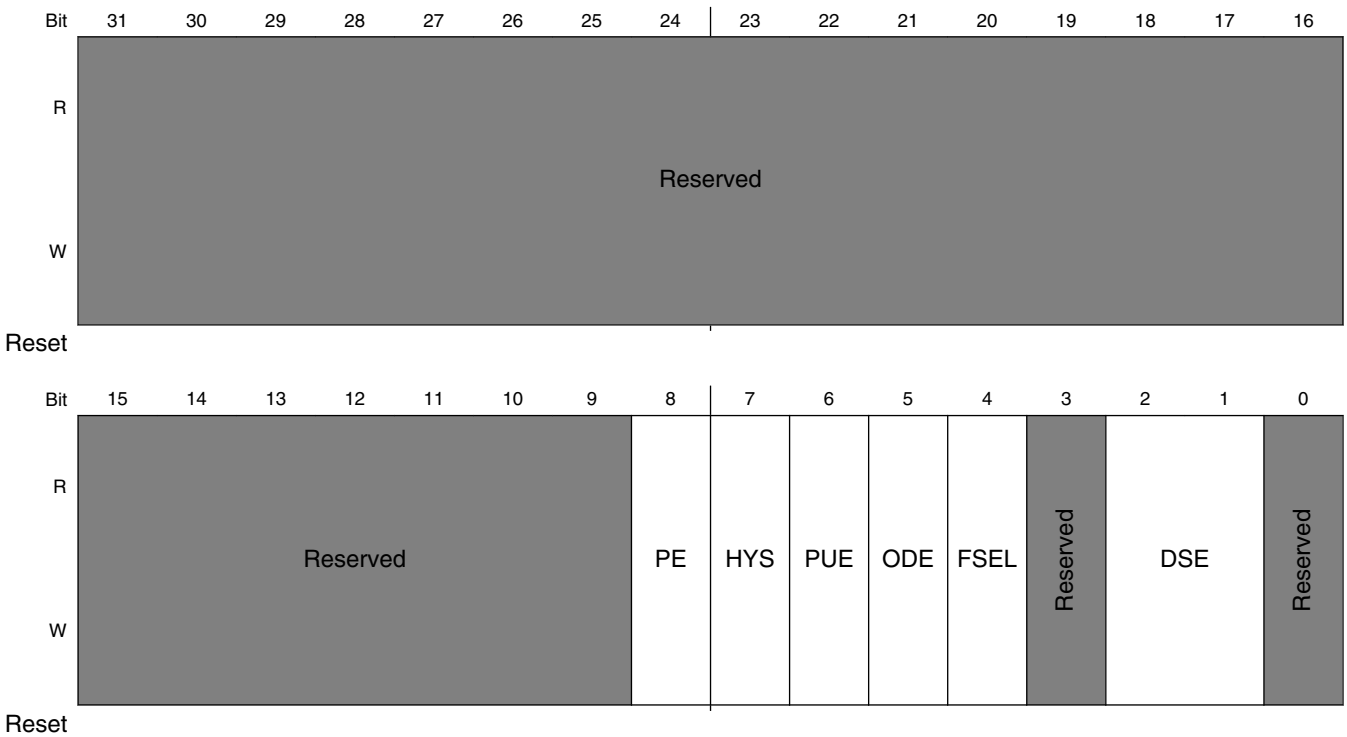
IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD1_DATA0 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD1_DATA0 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA0 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.186 SW_PAD_CTL_PAD_SD1_DATA1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1)

SW_PAD_CTL Register

Address: 3033_0000h base + 2F8h offset = 3033_02F8h



IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD1_DATA1 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD1_DATA1 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD1_DATA1

Table continues on the next page...

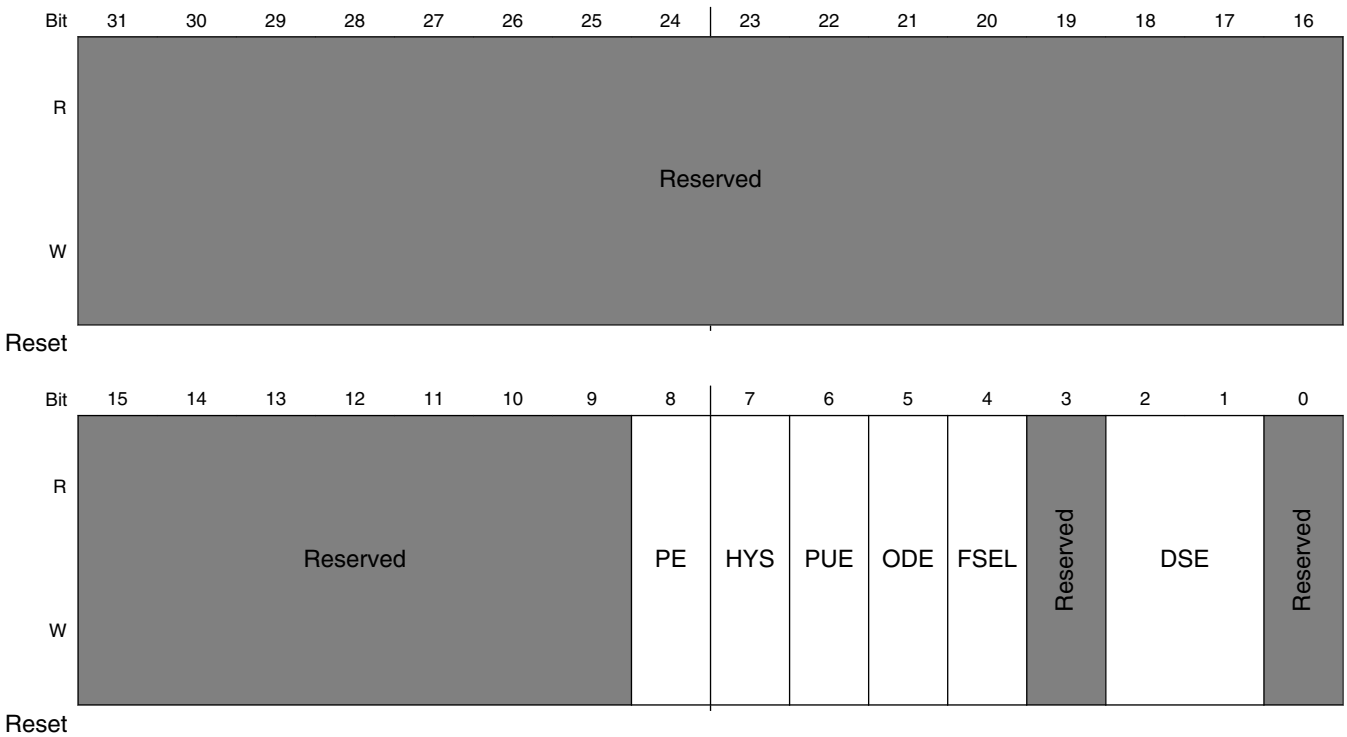
IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD1_DATA1 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD1_DATA1 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA1 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.187 SW_PAD_CTL_PAD_SD1_DATA2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2)

SW_PAD_CTL Register

Address: 3033_0000h base + 2FCh offset = 3033_02FCh



IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD1_DATA2 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD1_DATA2 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD1_DATA2

Table continues on the next page...

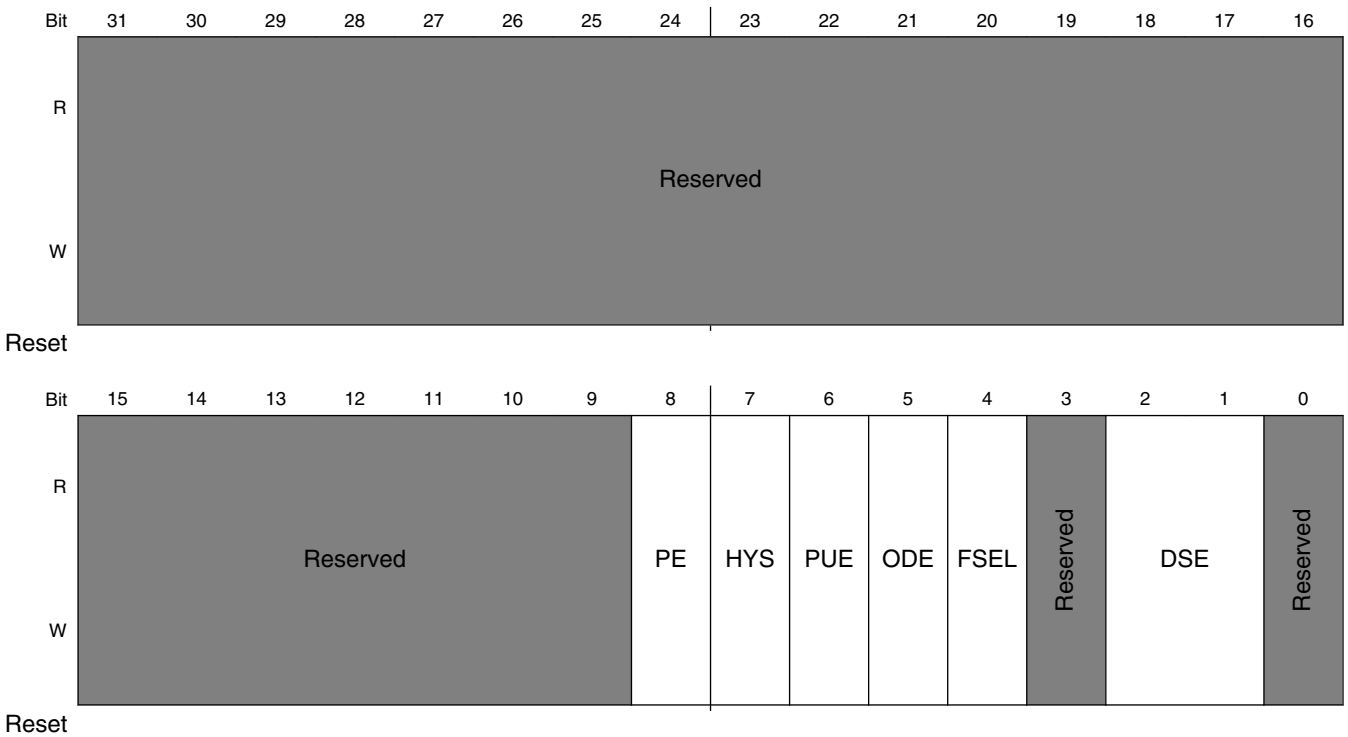
IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD1_DATA2 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD1_DATA2 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA2 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.188 SW_PAD_CTL_PAD_SD1_DATA3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3)

SW_PAD_CTL Register

Address: 3033_0000h base + 300h offset = 3033_0300h



IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD1_DATA3 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD1_DATA3 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD1_DATA3

Table continues on the next page...

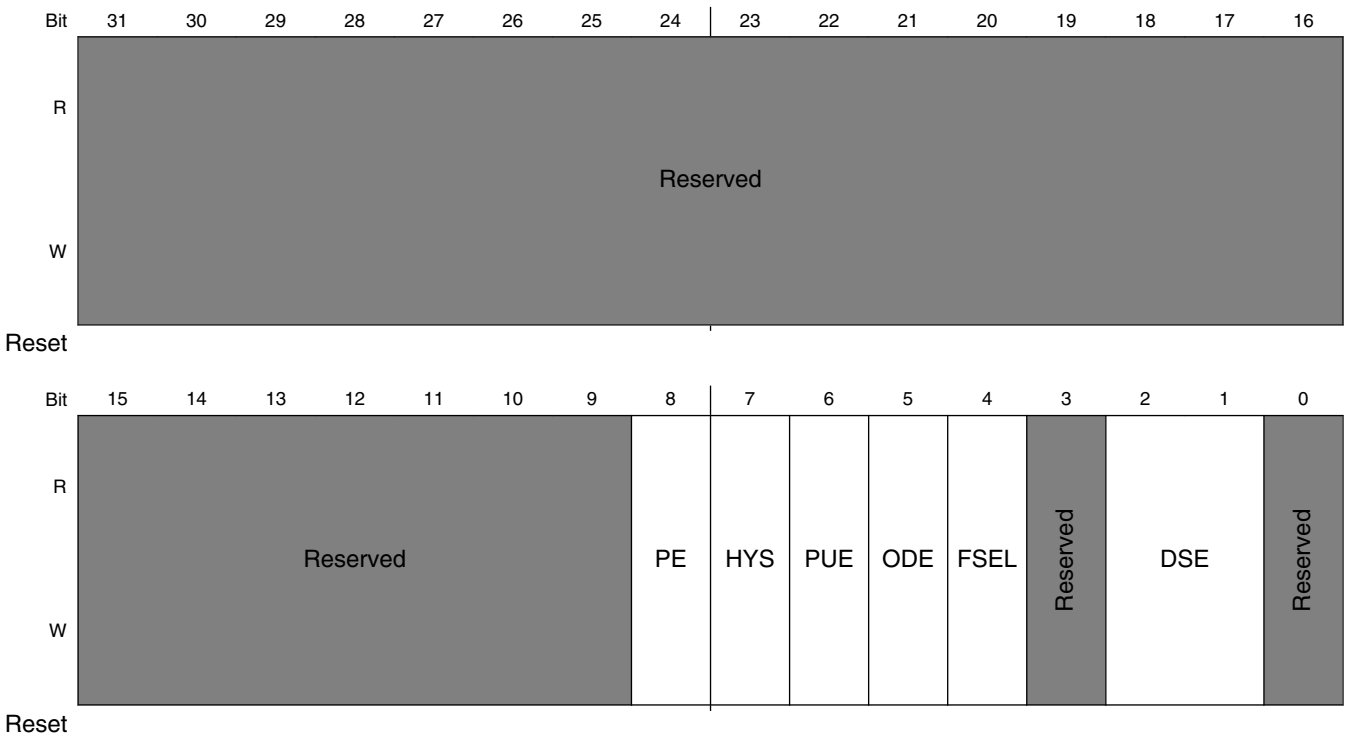
IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD1_DATA3 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD1_DATA3 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA3 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.189 SW_PAD_CTL_PAD_SD1_DATA4 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA4)

SW_PAD_CTL Register

Address: 3033_0000h base + 304h offset = 3033_0304h



IOMUXC_SW_PAD_CTL_PAD_SD1_DATA4 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD1_DATA4 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD1_DATA4 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD1_DATA4

Table continues on the next page...

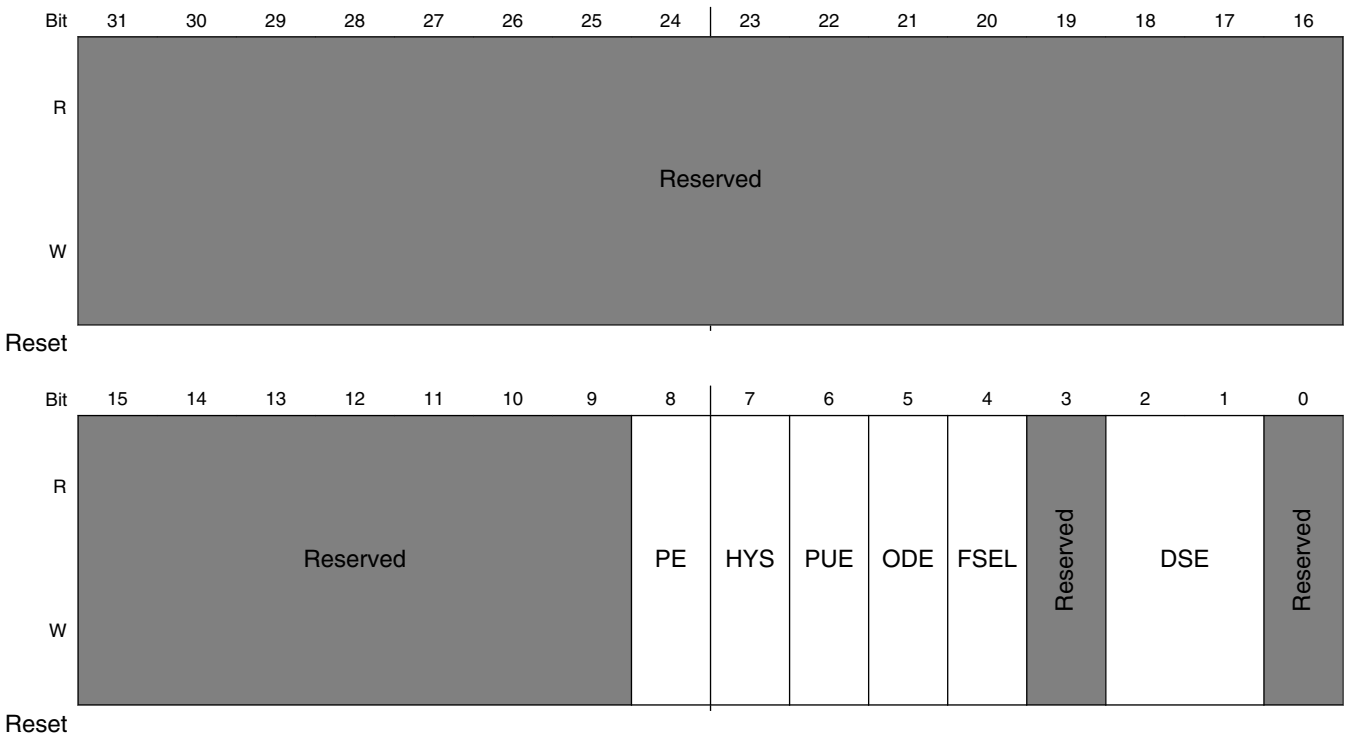
IOMUXC_SW_PAD_CTL_PAD_SD1_DATA4 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD1_DATA4 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD1_DATA4 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA4 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.190 SW_PAD_CTL_PAD_SD1_DATA5 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA5)

SW_PAD_CTL Register

Address: 3033_0000h base + 308h offset = 3033_0308h



IOMUXC_SW_PAD_CTL_PAD_SD1_DATA5 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD1_DATA5 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD1_DATA5 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD1_DATA5

Table continues on the next page...

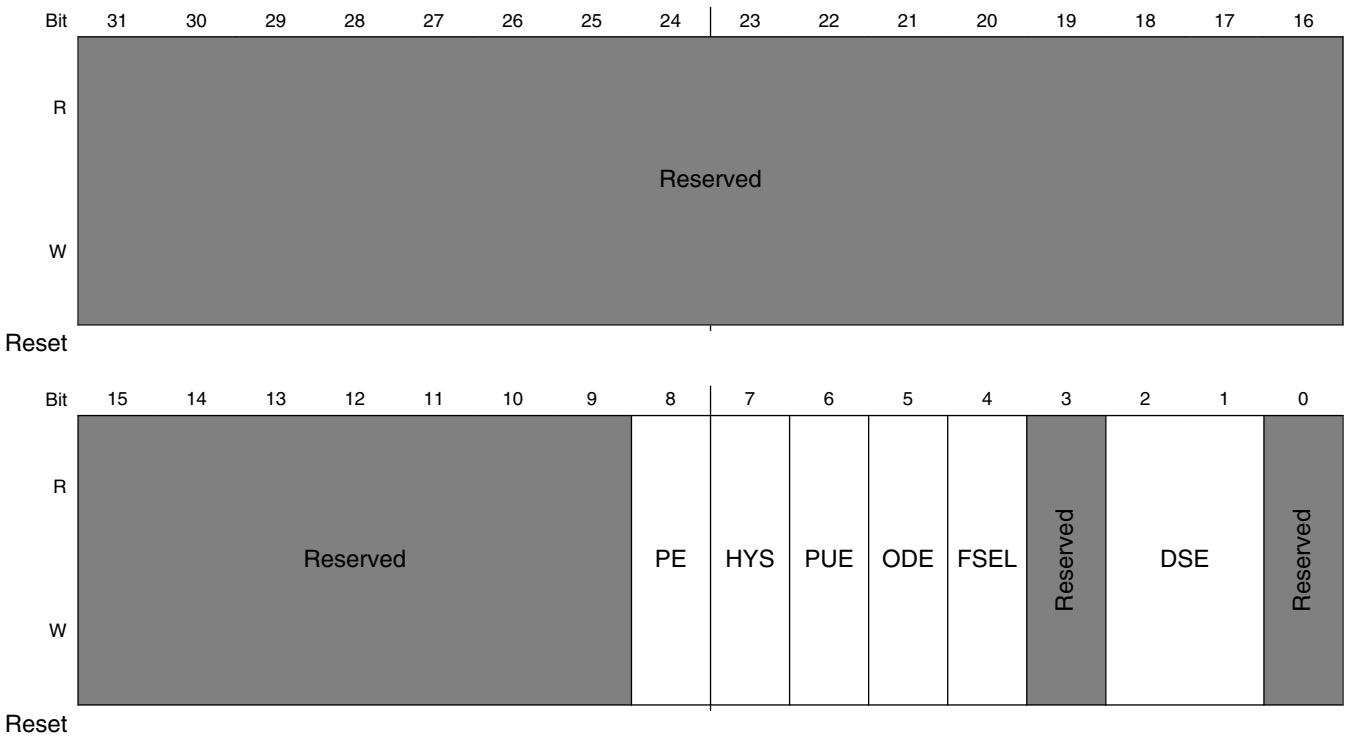
IOMUXC_SW_PAD_CTL_PAD_SD1_DATA5 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD1_DATA5 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD1_DATA5 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA5 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.191 SW_PAD_CTL_PAD_SD1_DATA6 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA6)

SW_PAD_CTL Register

Address: 3033_0000h base + 30Ch offset = 3033_030Ch



IOMUXC_SW_PAD_CTL_PAD_SD1_DATA6 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD1_DATA6 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD1_DATA6 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD1_DATA6

Table continues on the next page...

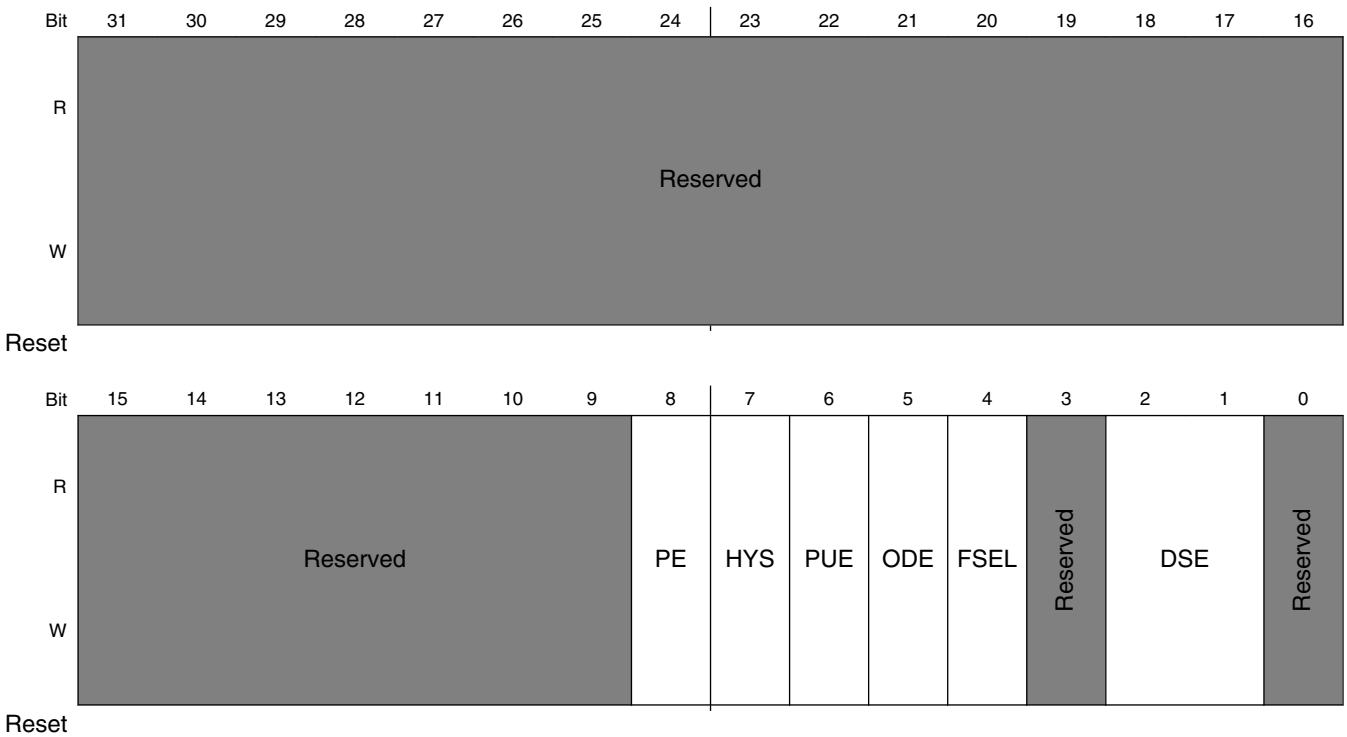
IOMUXC_SW_PAD_CTL_PAD_SD1_DATA6 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD1_DATA6 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD1_DATA6 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA6 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.192 SW_PAD_CTL_PAD_SD1_DATA7 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA7)

SW_PAD_CTL Register

Address: 3033_0000h base + 310h offset = 3033_0310h



IOMUXC_SW_PAD_CTL_PAD_SD1_DATA7 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD1_DATA7 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD1_DATA7 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD1_DATA7

Table continues on the next page...

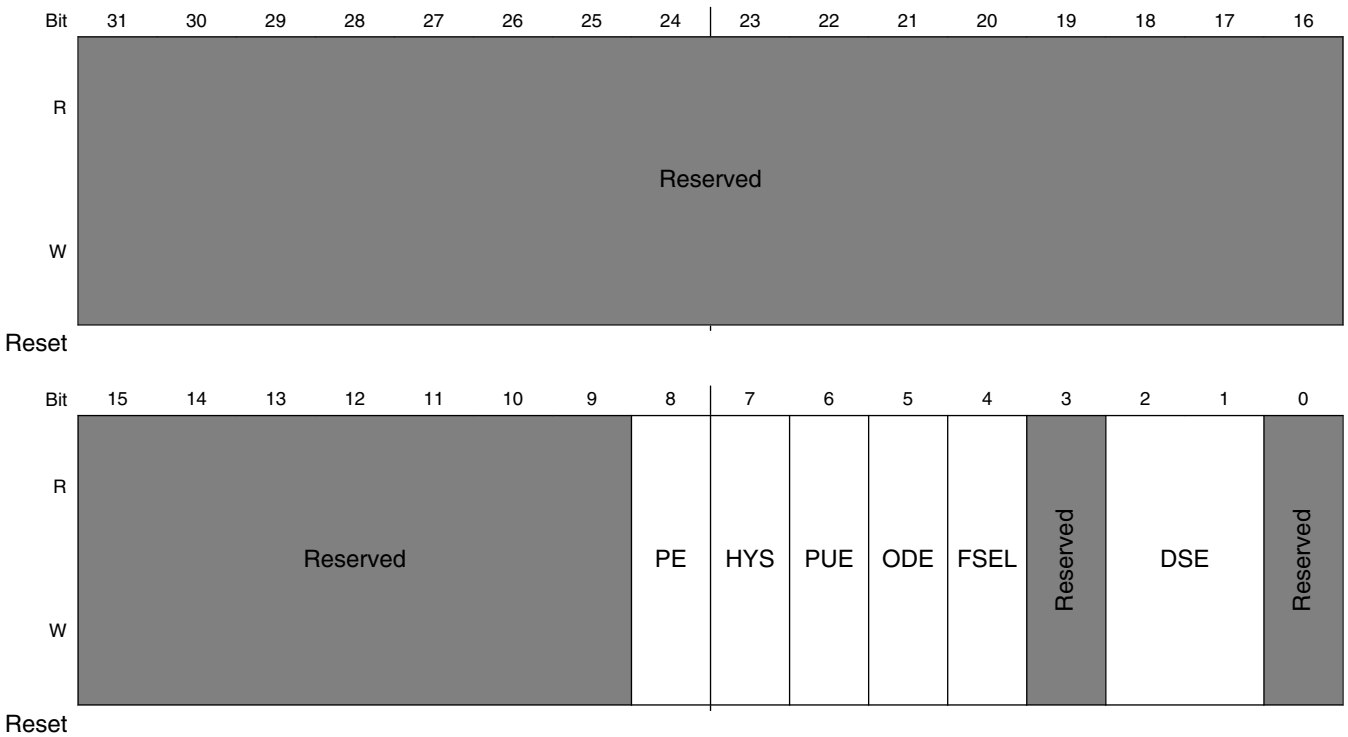
IOMUXC_SW_PAD_CTL_PAD_SD1_DATA7 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD1_DATA7 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD1_DATA7 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA7 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.193 SW_PAD_CTL_PAD_SD1_RESET_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_RESET_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 314h offset = 3033_0314h



IOMUXC_SW_PAD_CTL_PAD_SD1_RESET_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD1_RESET_B 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD1_RESET_B 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD1_RESET_B

Table continues on the next page...

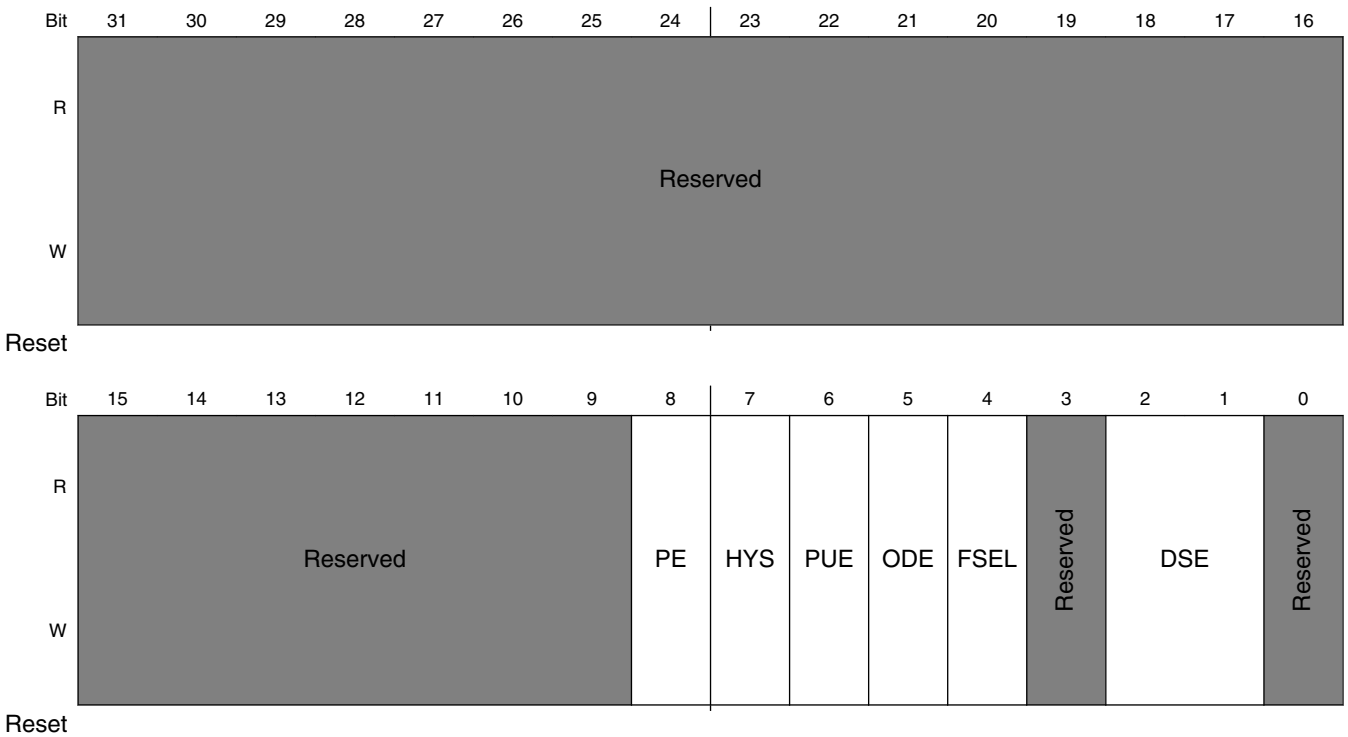
IOMUXC_SW_PAD_CTL_PAD_SD1_RESET_B field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD1_RESET_B 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD1_RESET_B 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_RESET_B 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.194 SW_PAD_CTL_PAD_SD1_STROBE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_STROBE)

SW_PAD_CTL Register

Address: 3033_0000h base + 318h offset = 3033_0318h



IOMUXC_SW_PAD_CTL_PAD_SD1_STROBE field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD1_STROBE 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD1_STROBE 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD1_STROBE

Table continues on the next page...

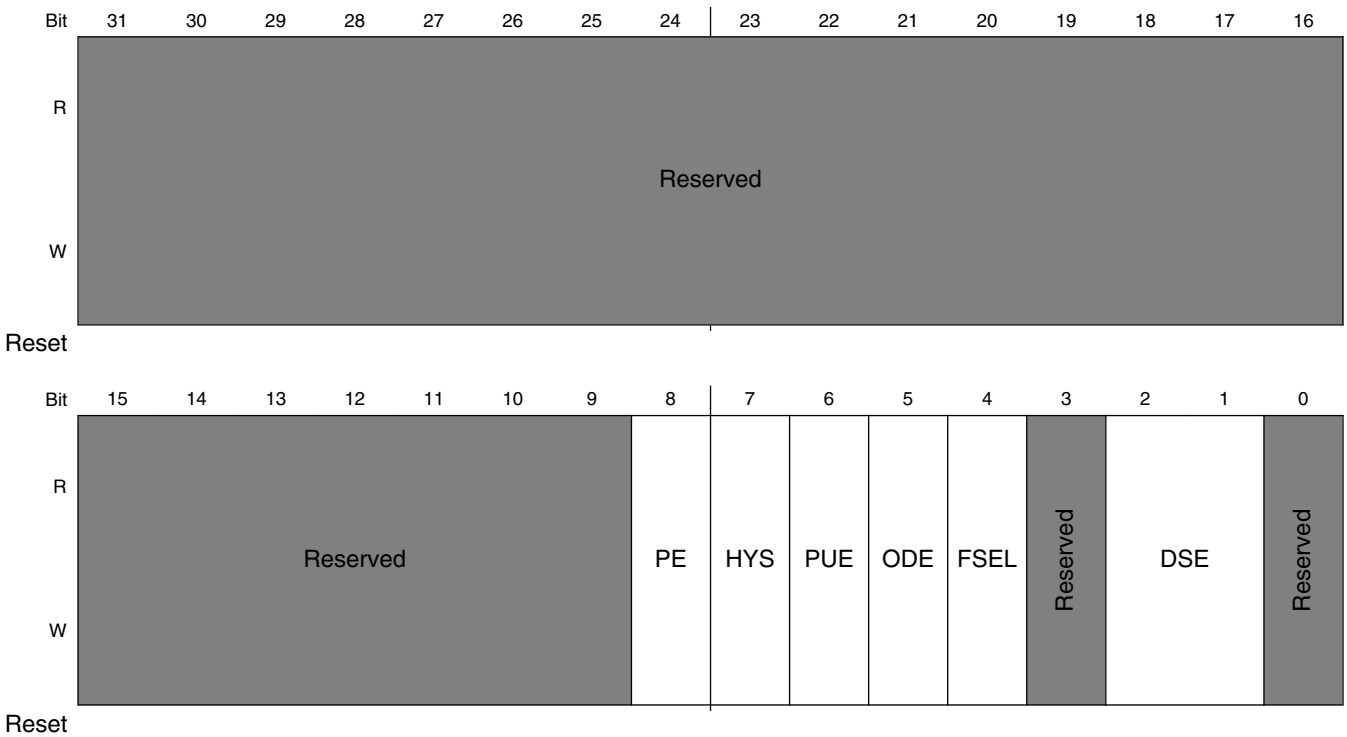
IOMUXC_SW_PAD_CTL_PAD_SD1_STROBE field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD1_STROBE 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD1_STROBE 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_STROBE 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.195 SW_PAD_CTL_PAD_SD2_CD_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CD_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 31Ch offset = 3033_031Ch



IOMUXC_SW_PAD_CTL_PAD_SD2_CD_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD2_CD_B 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD2_CD_B 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD2_CD_B

Table continues on the next page...

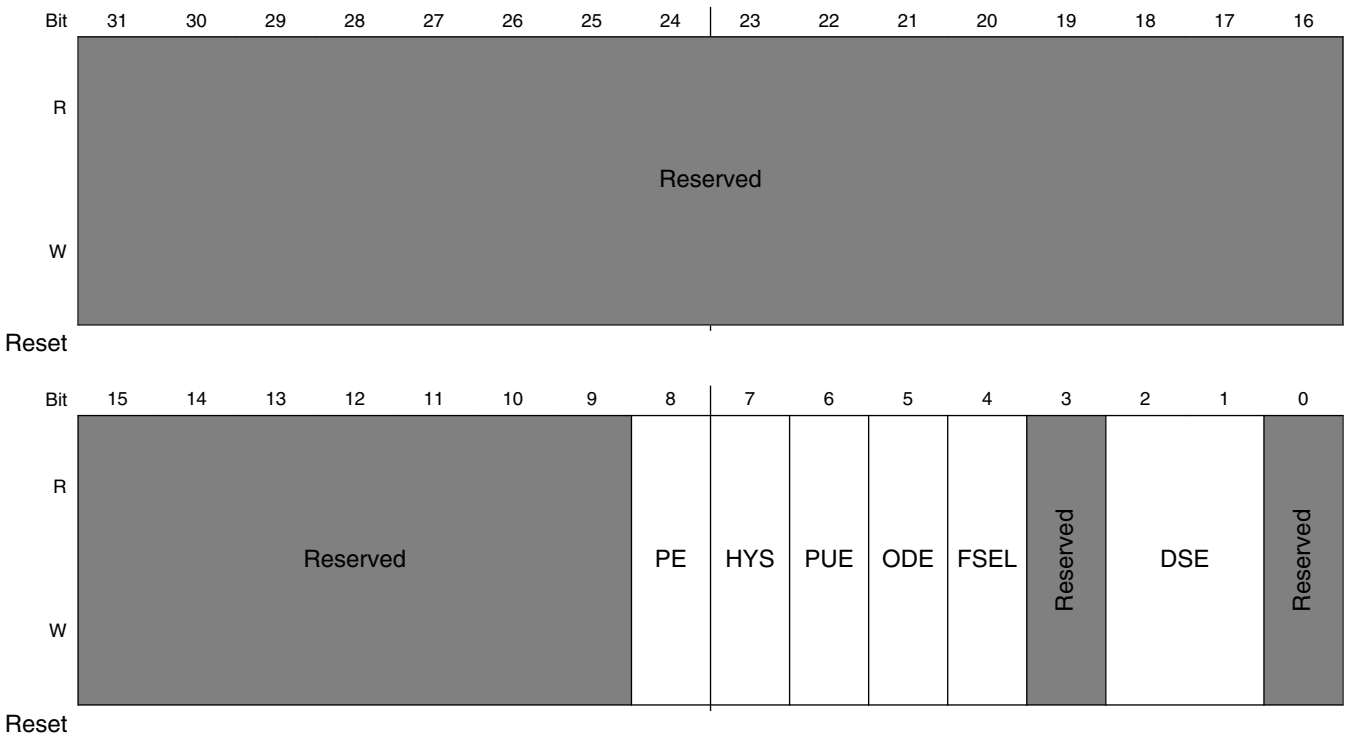
IOMUXC_SW_PAD_CTL_PAD_SD2_CD_B field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD2_CD_B 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD2_CD_B 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_CD_B 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.196 SW_PAD_CTL_PAD_SD2_CLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 320h offset = 3033_0320h



IOMUXC_SW_PAD_CTL_PAD_SD2_CLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD2_CLK 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD2_CLK 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD2_CLK

Table continues on the next page...

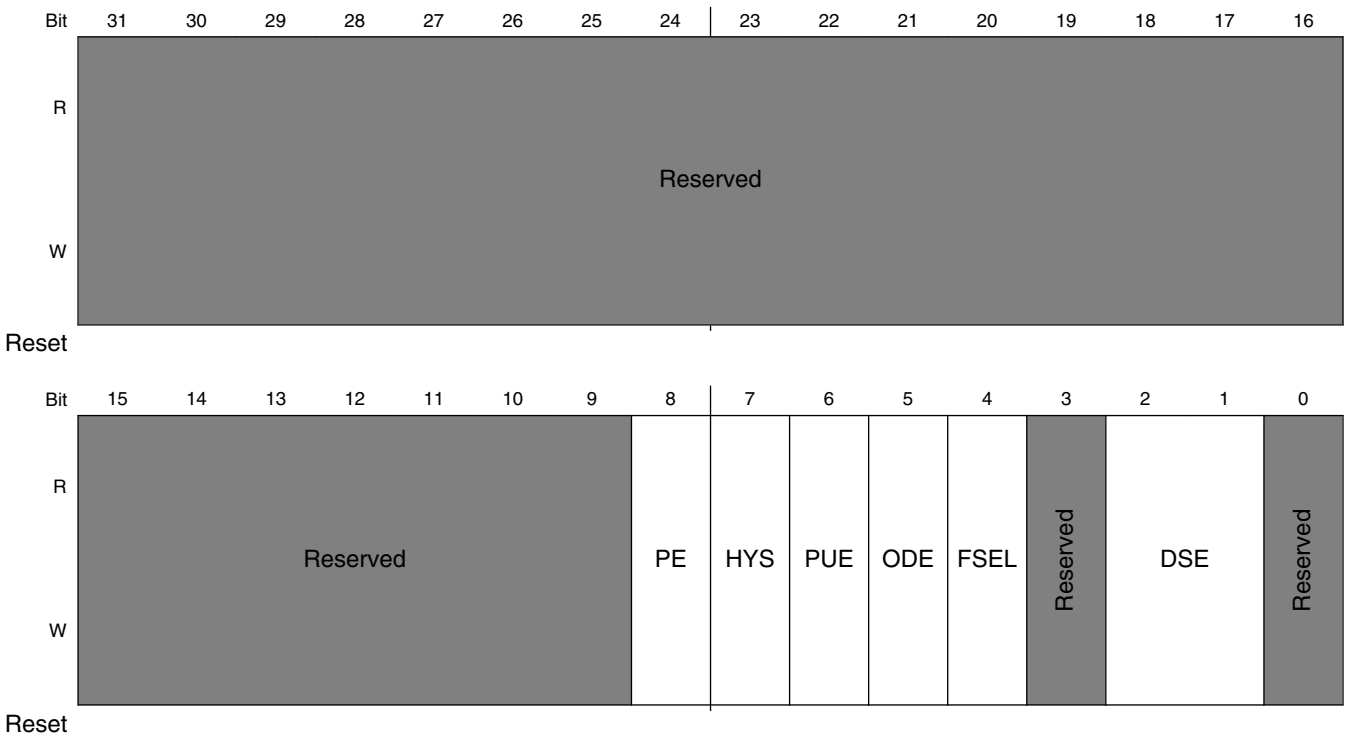
IOMUXC_SW_PAD_CTL_PAD_SD2_CLK field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD2_CLK 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD2_CLK 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_CLK 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.197 SW_PAD_CTL_PAD_SD2_CMD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CMD)

SW_PAD_CTL Register

Address: 3033_0000h base + 324h offset = 3033_0324h



IOMUXC_SW_PAD_CTL_PAD_SD2_CMD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD2_CMD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD2_CMD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD2_CMD

Table continues on the next page...

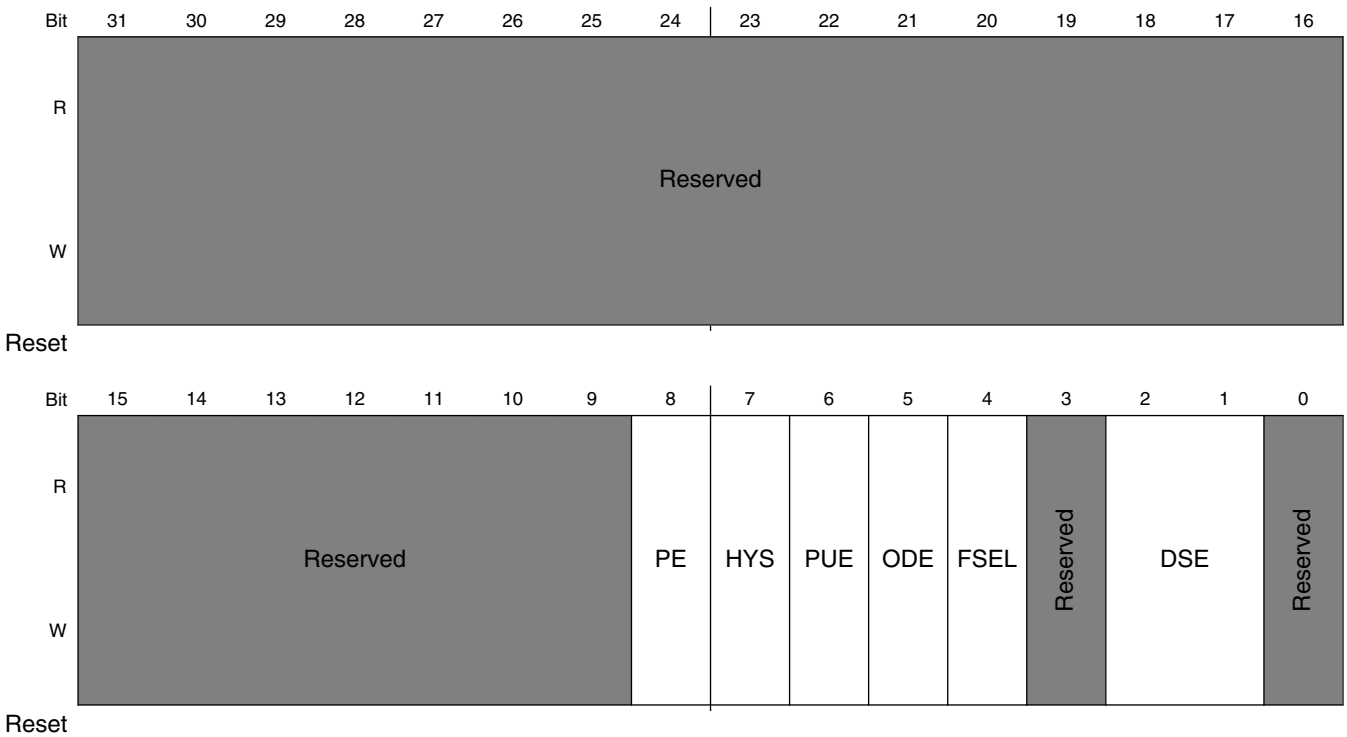
IOMUXC_SW_PAD_CTL_PAD_SD2_CMD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD2_CMD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD2_CMD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_CMD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.198 SW_PAD_CTL_PAD_SD2_DATA0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0)

SW_PAD_CTL Register

Address: 3033_0000h base + 328h offset = 3033_0328h



IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD2_DATA0 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD2_DATA0 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD2_DATA0

Table continues on the next page...

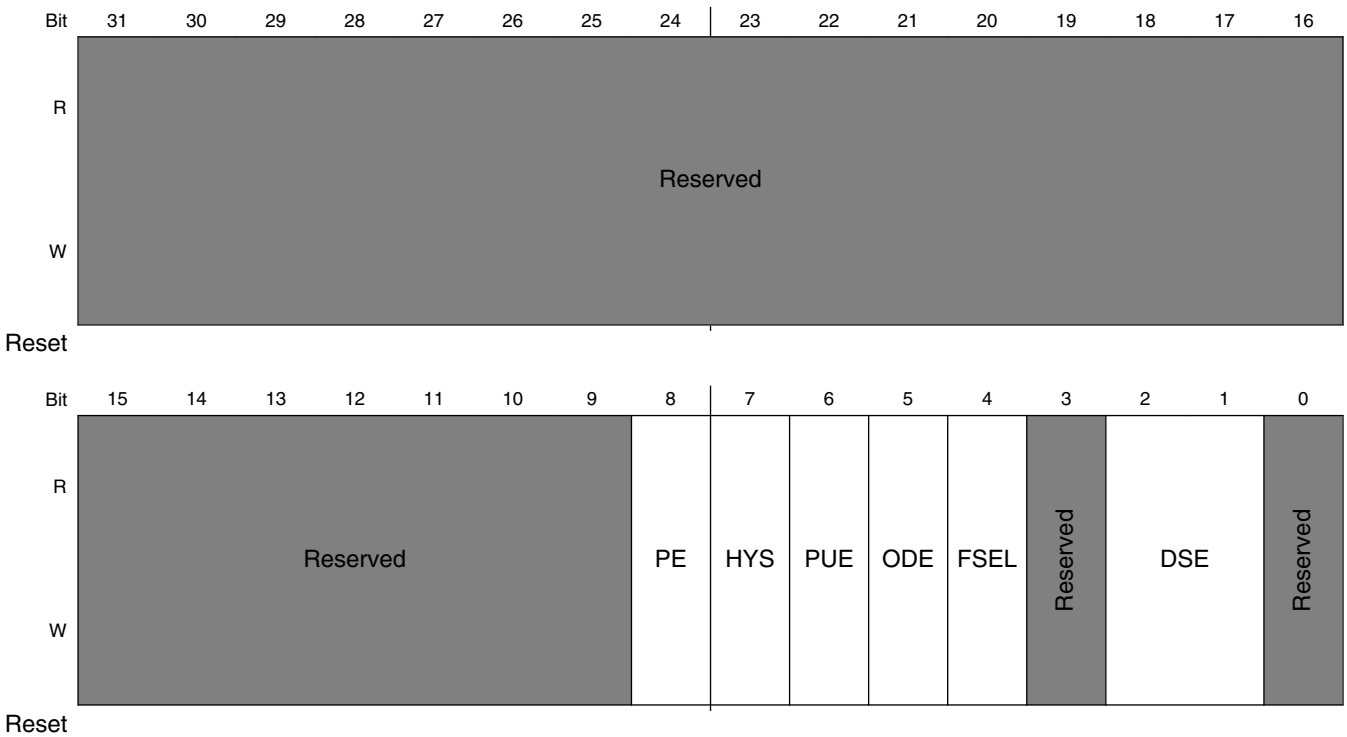
IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD2_DATA0 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD2_DATA0 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA0 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.199 SW_PAD_CTL_PAD_SD2_DATA1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1)

SW_PAD_CTL Register

Address: 3033_0000h base + 32Ch offset = 3033_032Ch



IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD2_DATA1 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD2_DATA1 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD2_DATA1

Table continues on the next page...

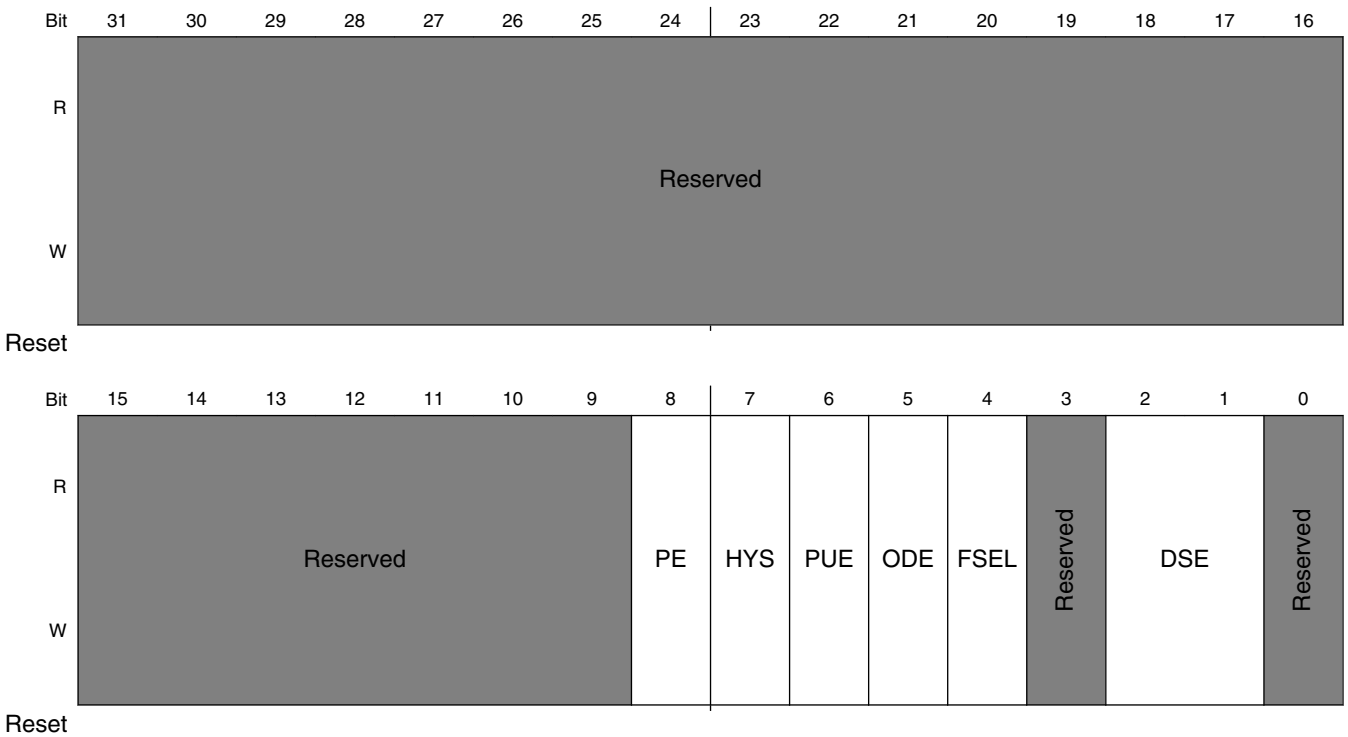
IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD2_DATA1 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD2_DATA1 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA1 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.200 SW_PAD_CTL_PAD_SD2_DATA2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2)

SW_PAD_CTL Register

Address: 3033_0000h base + 330h offset = 3033_0330h



IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD2_DATA2 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD2_DATA2 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD2_DATA2

Table continues on the next page...

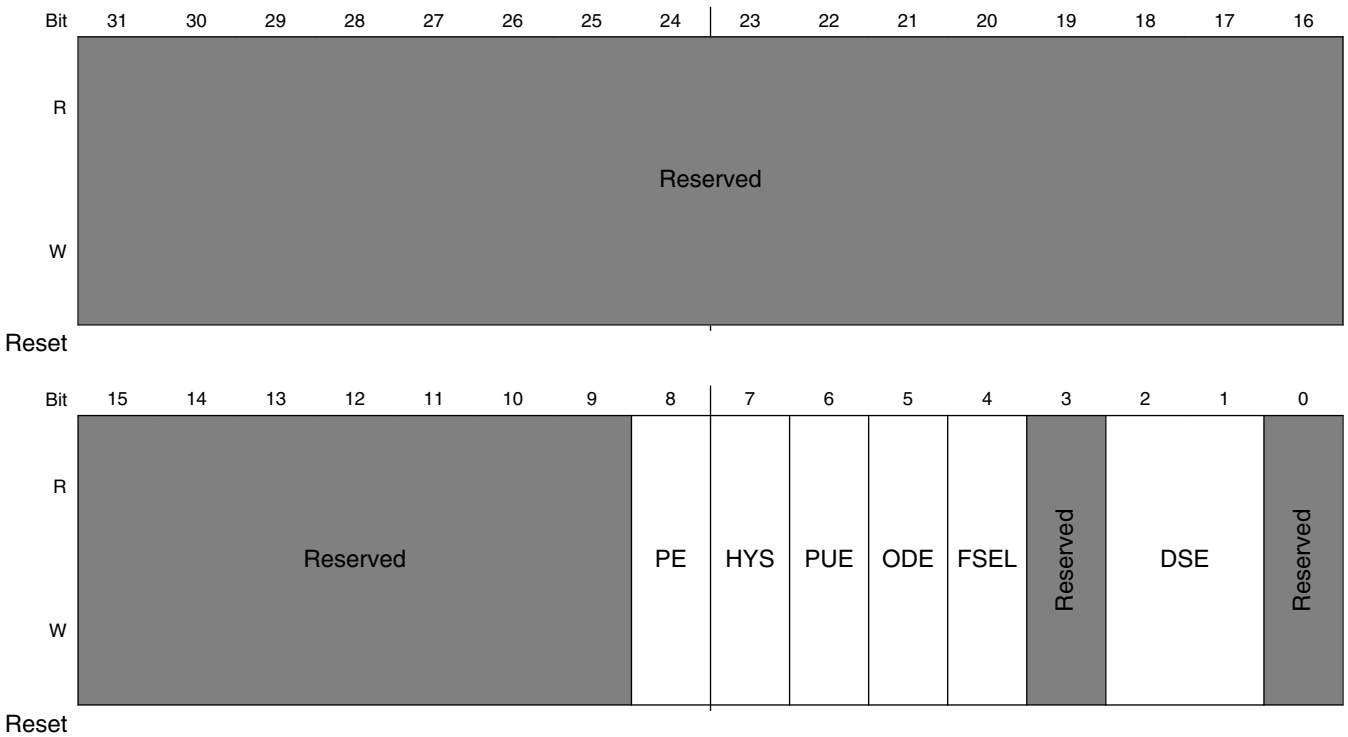
IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD2_DATA2 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD2_DATA2 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA2 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.201 SW_PAD_CTL_PAD_SD2_DATA3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3)

SW_PAD_CTL Register

Address: 3033_0000h base + 334h offset = 3033_0334h



IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD2_DATA3 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD2_DATA3 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD2_DATA3

Table continues on the next page...

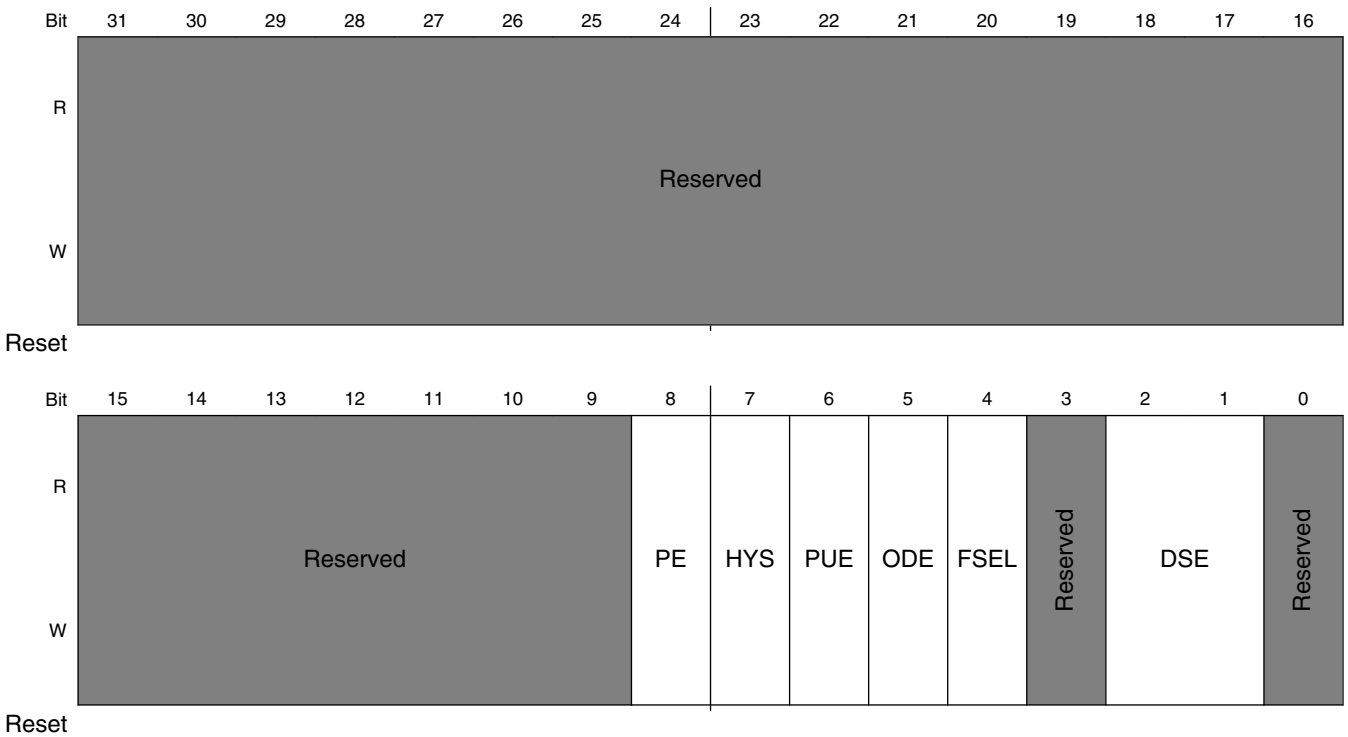
IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD2_DATA3 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD2_DATA3 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA3 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.202 SW_PAD_CTL_PAD_SD2_RESET_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_RESET_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 338h offset = 3033_0338h



IOMUXC_SW_PAD_CTL_PAD_SD2_RESET_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD2_RESET_B 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD2_RESET_B 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD2_RESET_B

Table continues on the next page...

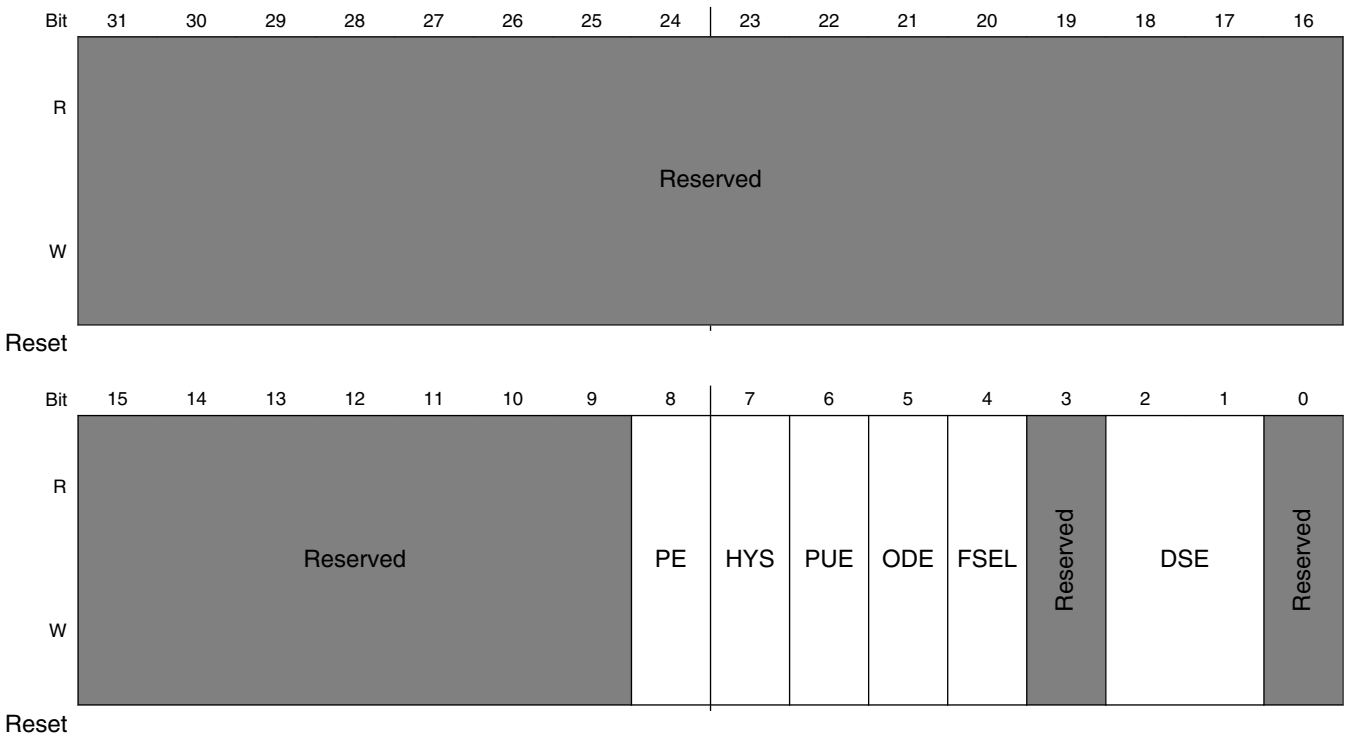
IOMUXC_SW_PAD_CTL_PAD_SD2_RESET_B field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD2_RESET_B 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD2_RESET_B 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_RESET_B 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.203 SW_PAD_CTL_PAD_SD2_WP SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_WP)

SW_PAD_CTL Register

Address: 3033_0000h base + 33Ch offset = 3033_033Ch



IOMUXC_SW_PAD_CTL_PAD_SD2_WP field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SD2_WP 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SD2_WP 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SD2_WP

Table continues on the next page...

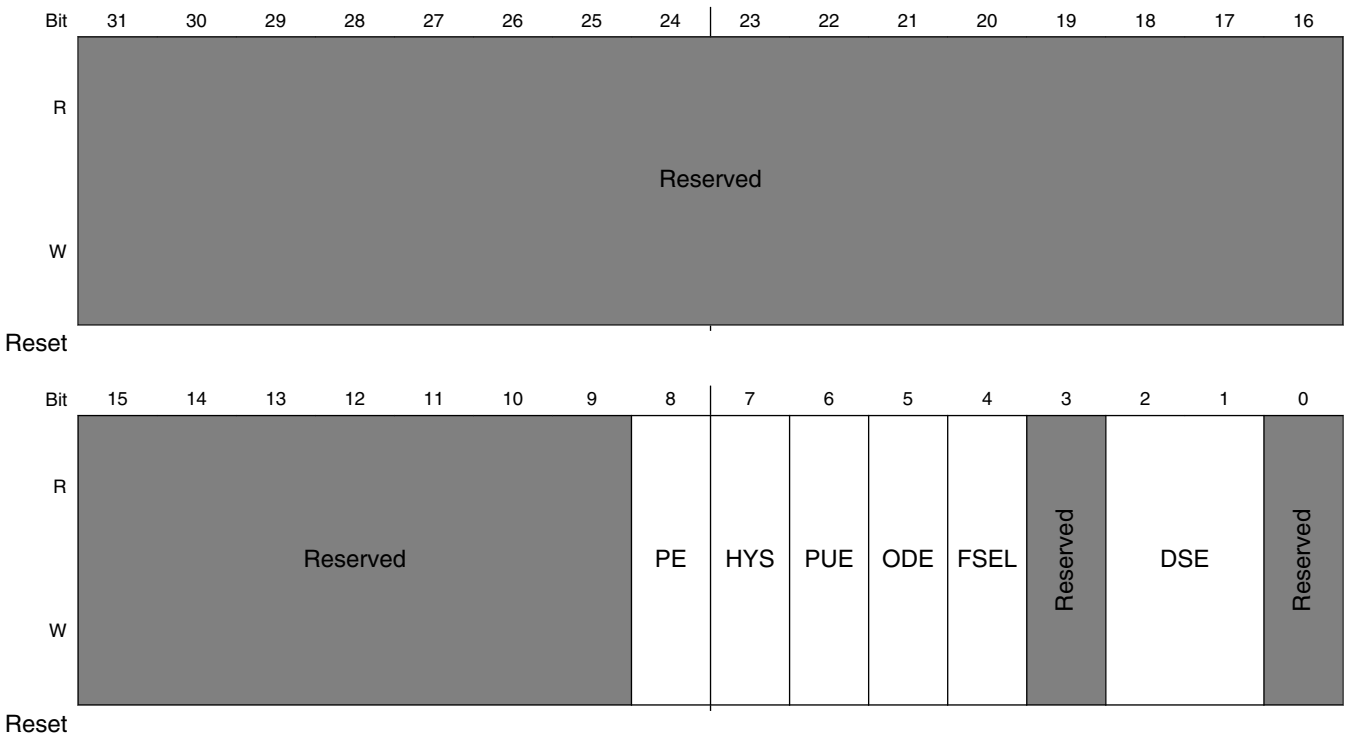
IOMUXC_SW_PAD_CTL_PAD_SD2_WP field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SD2_WP 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SD2_WP 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_WP 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.204 SW_PAD_CTL_PAD_NAND_ALE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_ALE)

SW_PAD_CTL Register

Address: 3033_0000h base + 340h offset = 3033_0340h



IOMUXC_SW_PAD_CTL_PAD_NAND_ALE field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_ALE 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_ALE 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_ALE

Table continues on the next page...

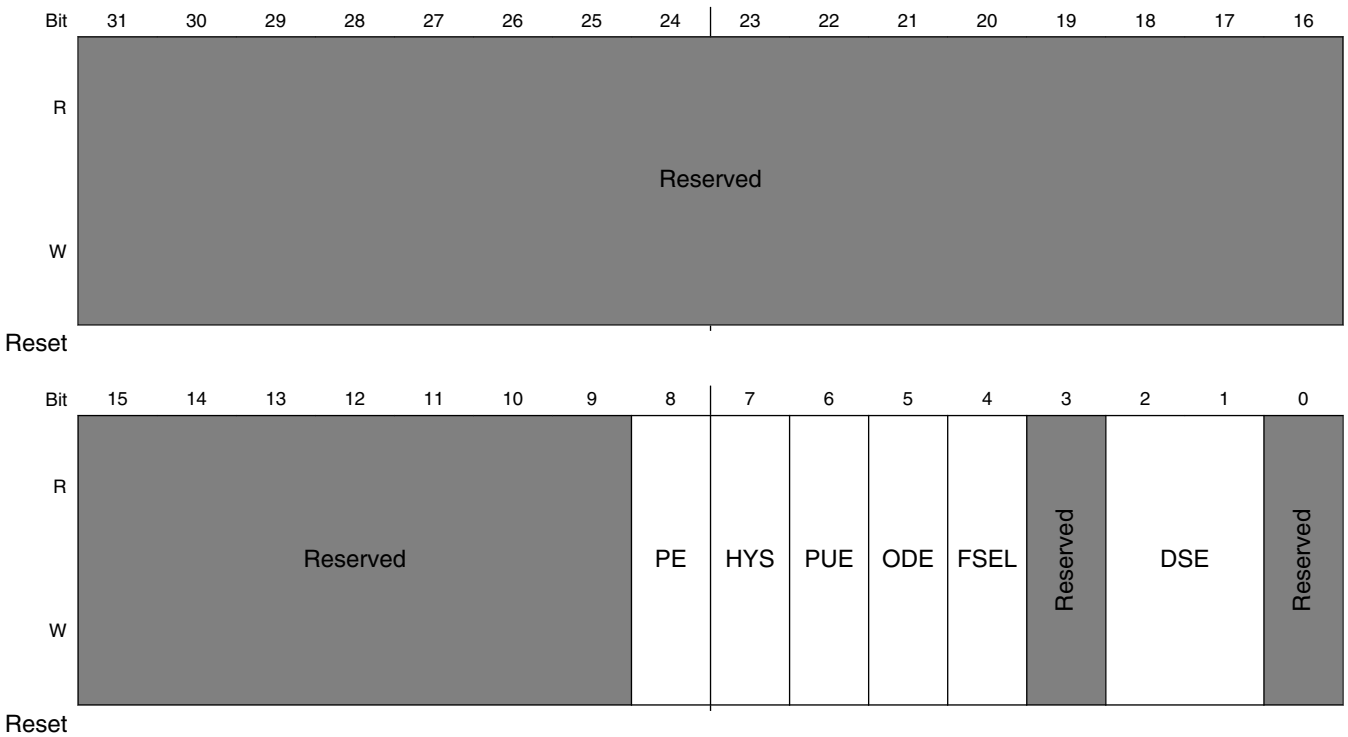
IOMUXC_SW_PAD_CTL_PAD_NAND_ALE field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_ALE 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_ALE 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_ALE 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.205 SW_PAD_CTL_PAD_NAND_CE0_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE0_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 344h offset = 3033_0344h



IOMUXC_SW_PAD_CTL_PAD_NAND_CE0_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_CE0_B 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_CE0_B 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_CE0_B

Table continues on the next page...

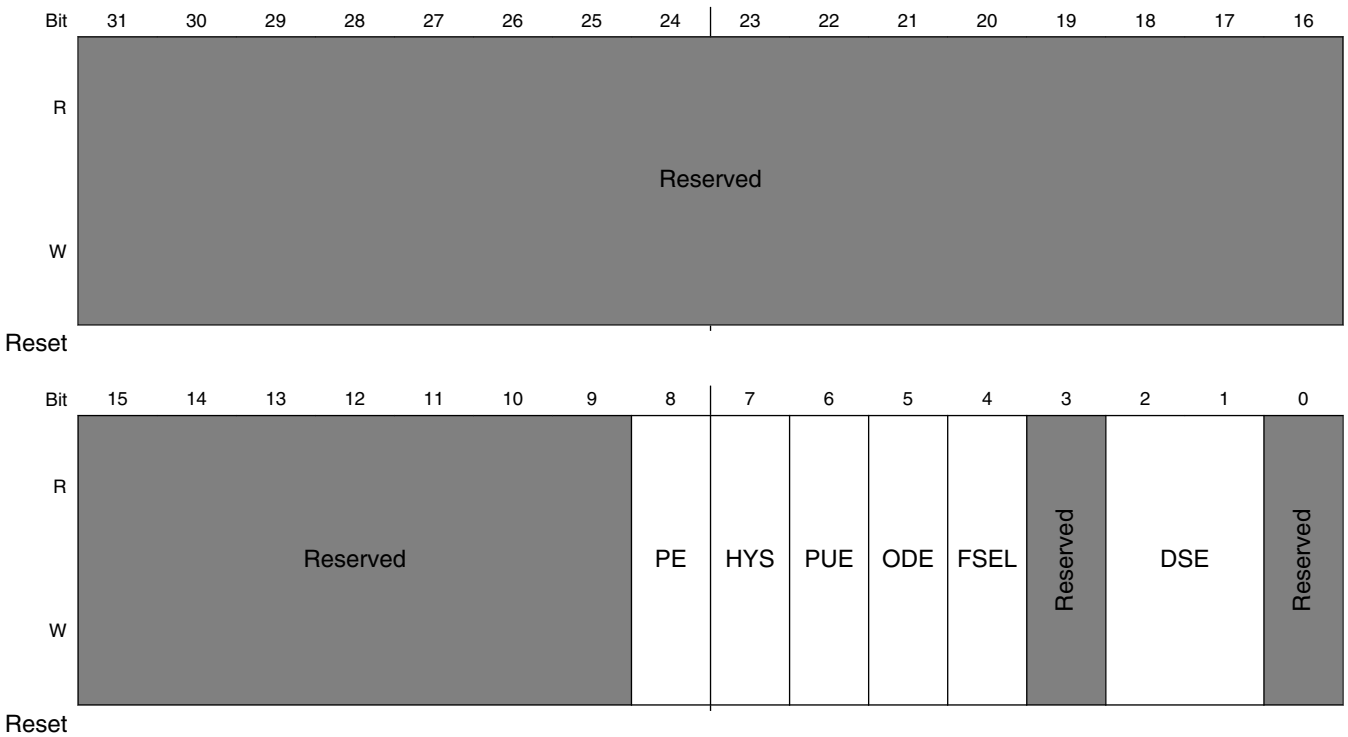
IOMUXC_SW_PAD_CTL_PAD_NAND_CE0_B field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_CE0_B 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_CE0_B 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_CE0_B 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.206 SW_PAD_CTL_PAD_NAND_CE1_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE1_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 348h offset = 3033_0348h



IOMUXC_SW_PAD_CTL_PAD_NAND_CE1_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_CE1_B 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_CE1_B 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_CE1_B

Table continues on the next page...

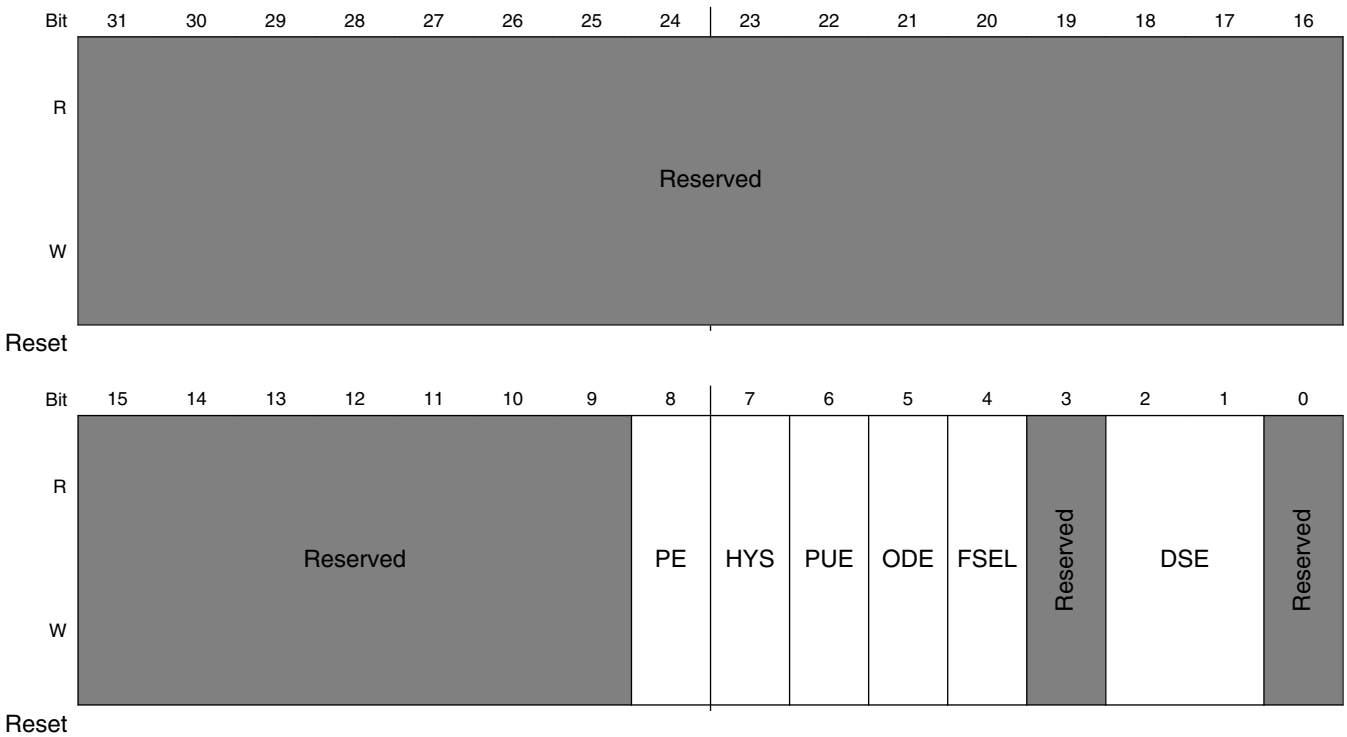
IOMUXC_SW_PAD_CTL_PAD_NAND_CE1_B field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_CE1_B 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_CE1_B 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_CE1_B 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.207 SW_PAD_CTL_PAD_NAND_CE2_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE2_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 34Ch offset = 3033_034Ch



IOMUXC_SW_PAD_CTL_PAD_NAND_CE2_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_CE2_B 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_CE2_B 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_CE2_B

Table continues on the next page...

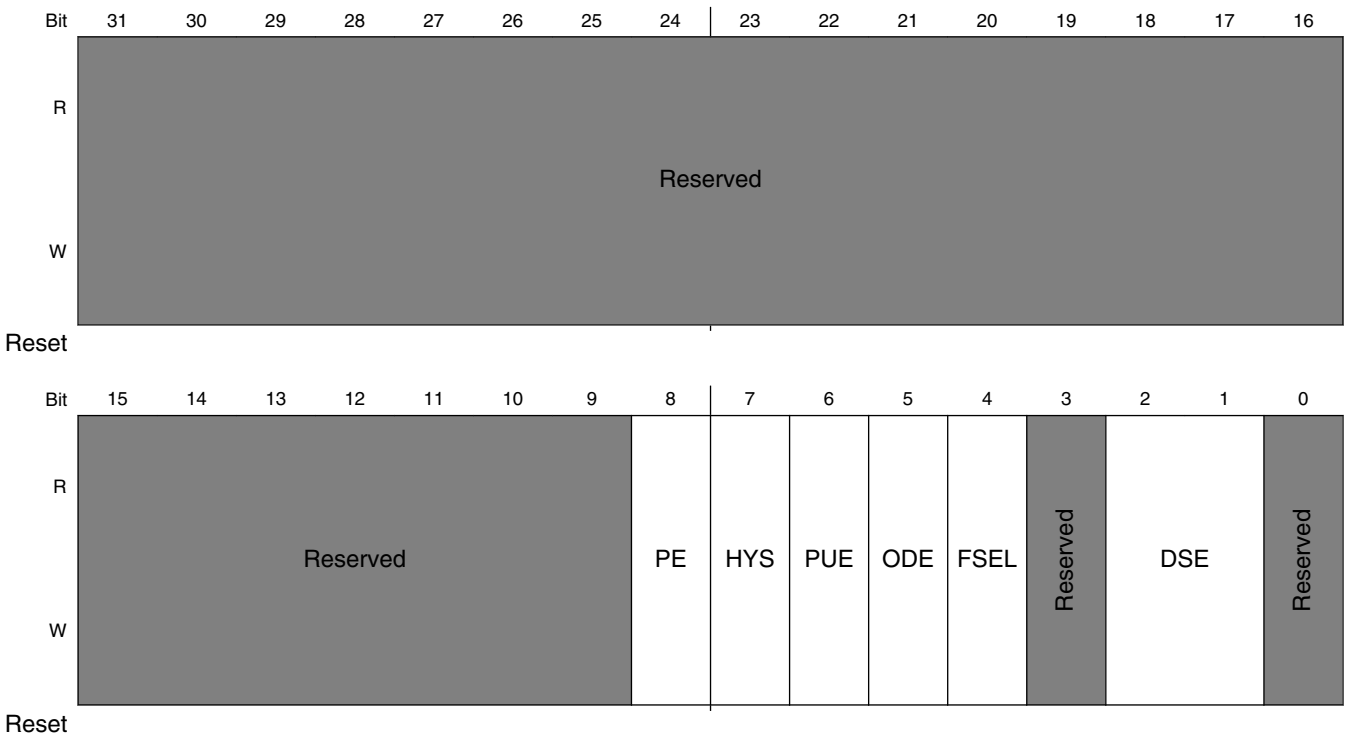
IOMUXC_SW_PAD_CTL_PAD_NAND_CE2_B field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_CE2_B 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_CE2_B 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_CE2_B 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.208 SW_PAD_CTL_PAD_NAND_CE3_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE3_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 350h offset = 3033_0350h



IOMUXC_SW_PAD_CTL_PAD_NAND_CE3_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_CE3_B 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_CE3_B 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_CE3_B

Table continues on the next page...

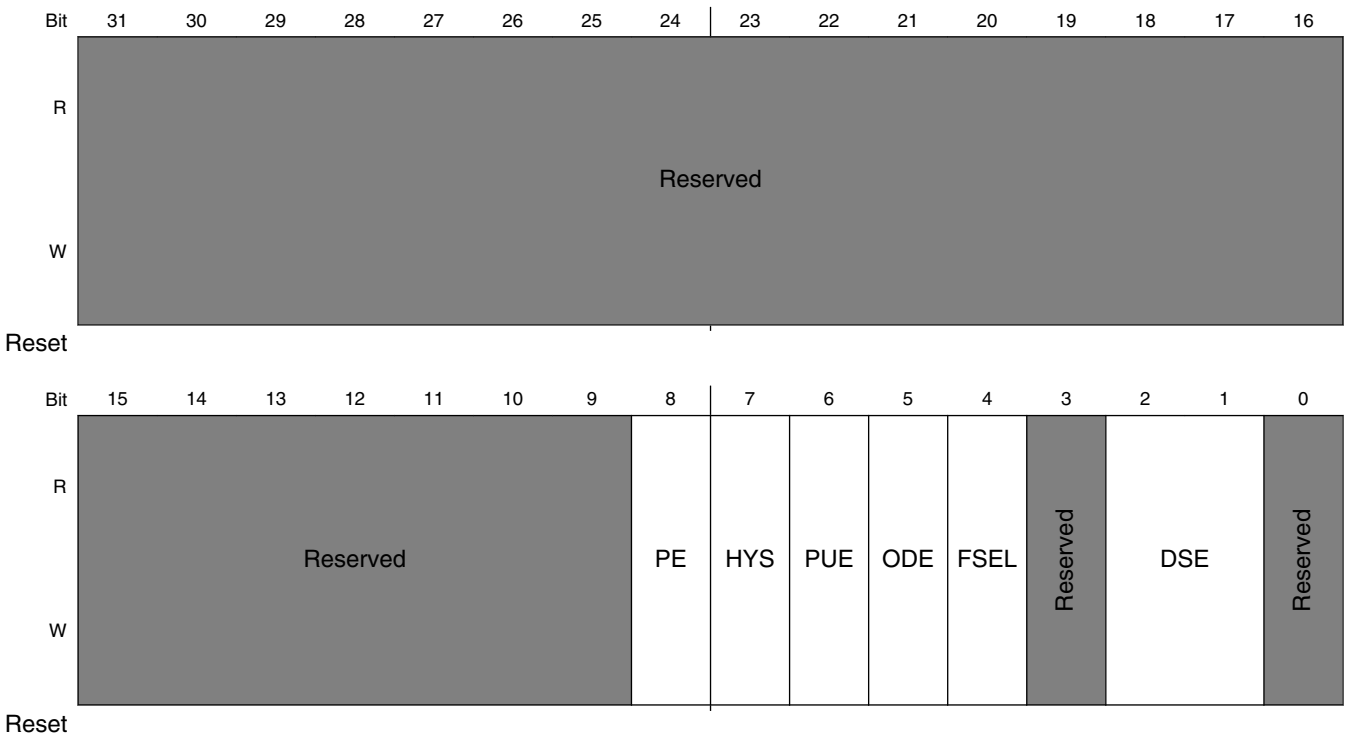
IOMUXC_SW_PAD_CTL_PAD_NAND_CE3_B field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_CE3_B 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_CE3_B 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_CE3_B 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.209 SW_PAD_CTL_PAD_NAND_CLE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CLE)

SW_PAD_CTL Register

Address: 3033_0000h base + 354h offset = 3033_0354h



IOMUXC_SW_PAD_CTL_PAD_NAND_CLE field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_CLE 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_CLE 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_CLE

Table continues on the next page...

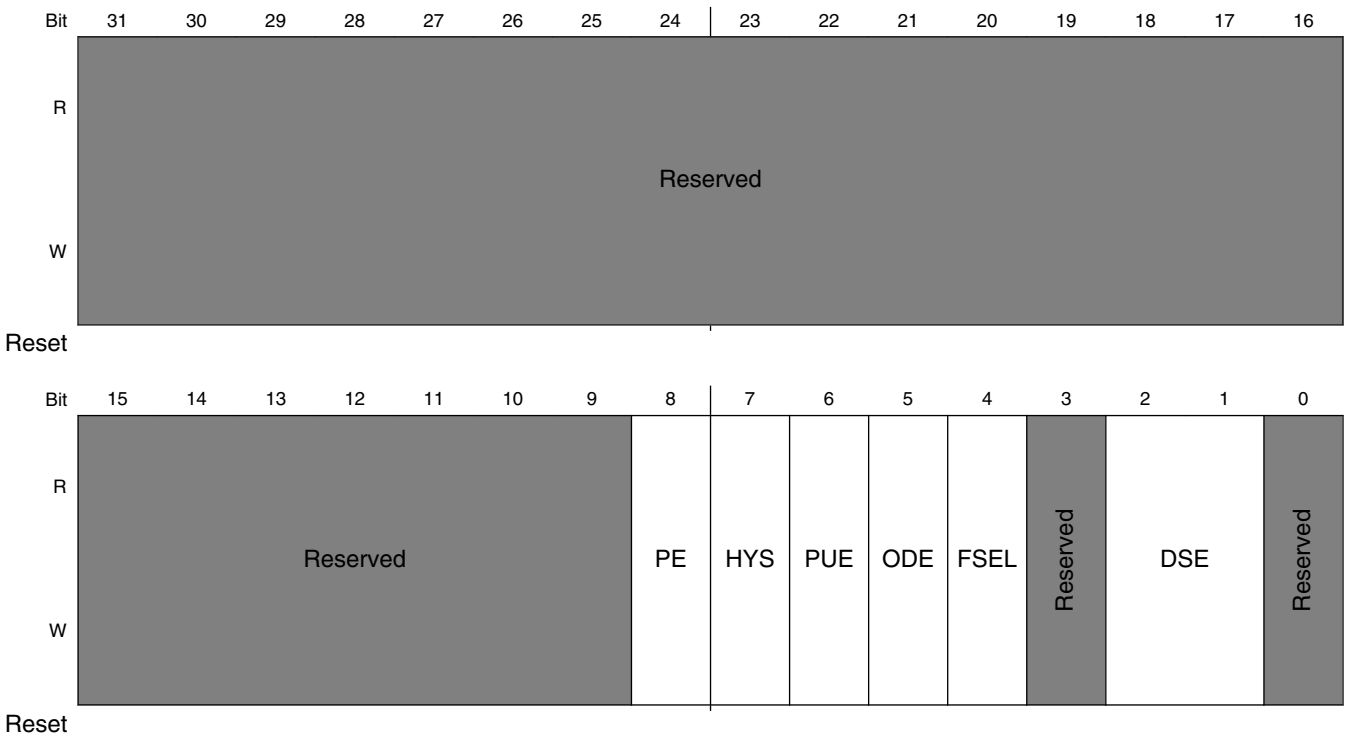
IOMUXC_SW_PAD_CTL_PAD_NAND_CLE field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_CLE 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_CLE 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_CLE 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.210 SW_PAD_CTL_PAD_NAND_DATA00 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA00)

SW_PAD_CTL Register

Address: 3033_0000h base + 358h offset = 3033_0358h



IOMUXC_SW_PAD_CTL_PAD_NAND_DATA00 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_DATA00 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_DATA00 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_DATA00

Table continues on the next page...

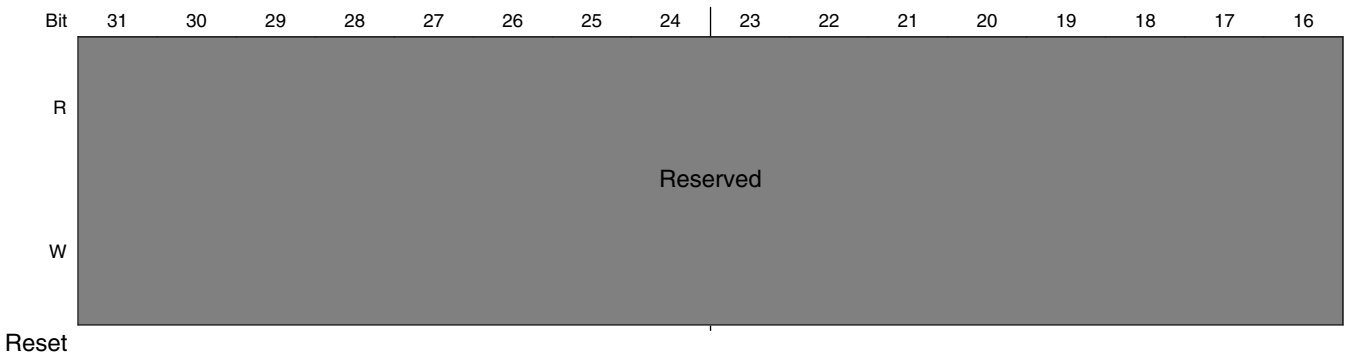
IOMUXC_SW_PAD_CTL_PAD_NAND_DATA00 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_DATA00 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_DATA00 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA00 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.211 SW_PAD_CTL_PAD_NAND_DATA01 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01)

SW_PAD_CTL Register

Address: 3033_0000h base + 35Ch offset = 3033_035Ch



IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_DATA01 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_DATA01 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_DATA01

Table continues on the next page...

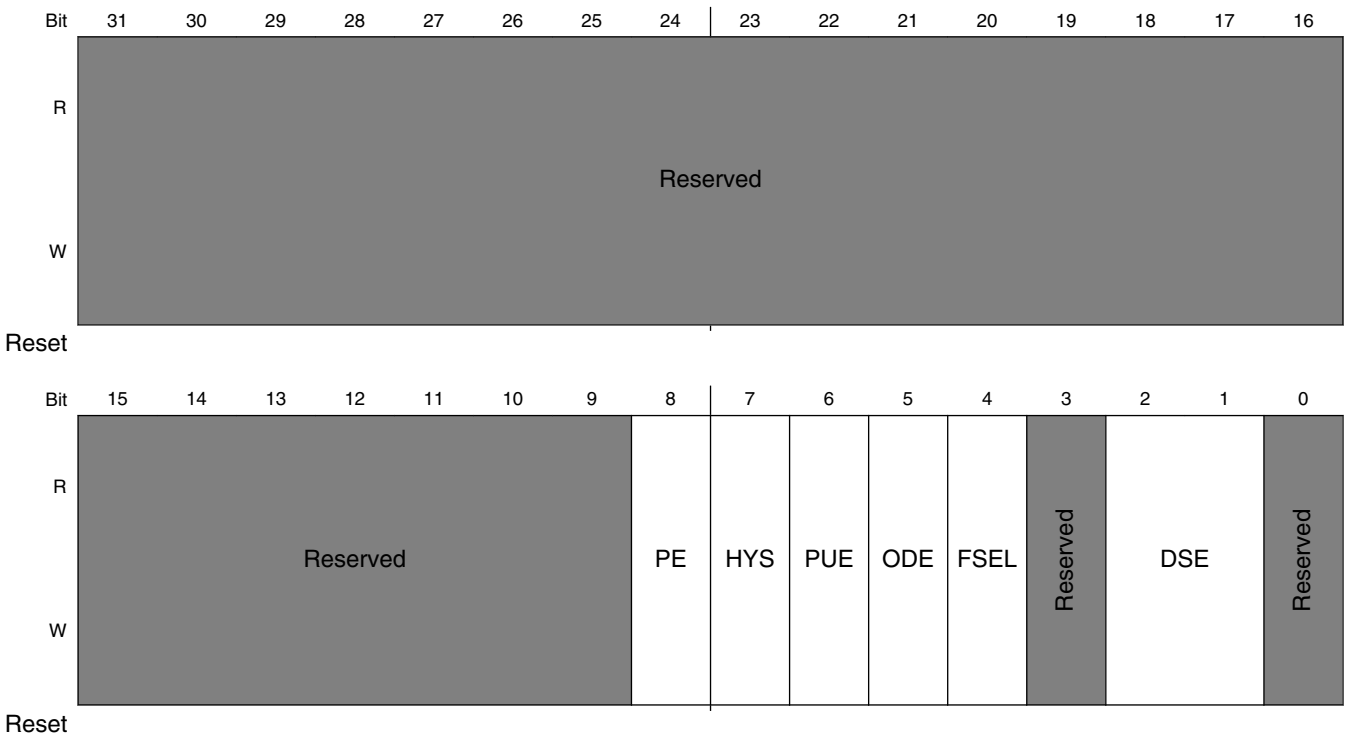
IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_DATA01 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_DATA01 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA01 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.212 SW_PAD_CTL_PAD_NAND_DATA02 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA02)

SW_PAD_CTL Register

Address: 3033_0000h base + 360h offset = 3033_0360h



IOMUXC_SW_PAD_CTL_PAD_NAND_DATA02 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_DATA02 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_DATA02 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_DATA02

Table continues on the next page...

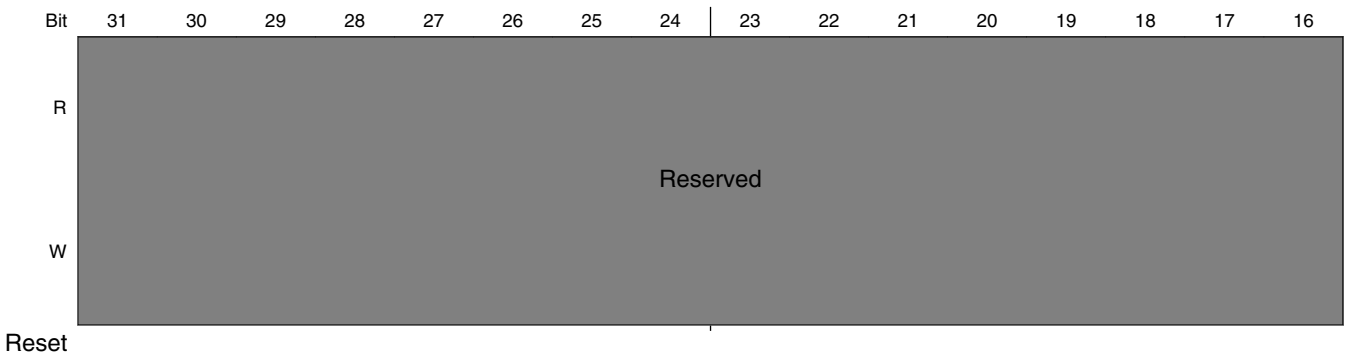
IOMUXC_SW_PAD_CTL_PAD_NAND_DATA02 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_DATA02 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_DATA02 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA02 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.213 SW_PAD_CTL_PAD_NAND_DATA03 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03)

SW_PAD_CTL Register

Address: 3033_0000h base + 364h offset = 3033_0364h



IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_DATA03 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_DATA03 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_DATA03

Table continues on the next page...

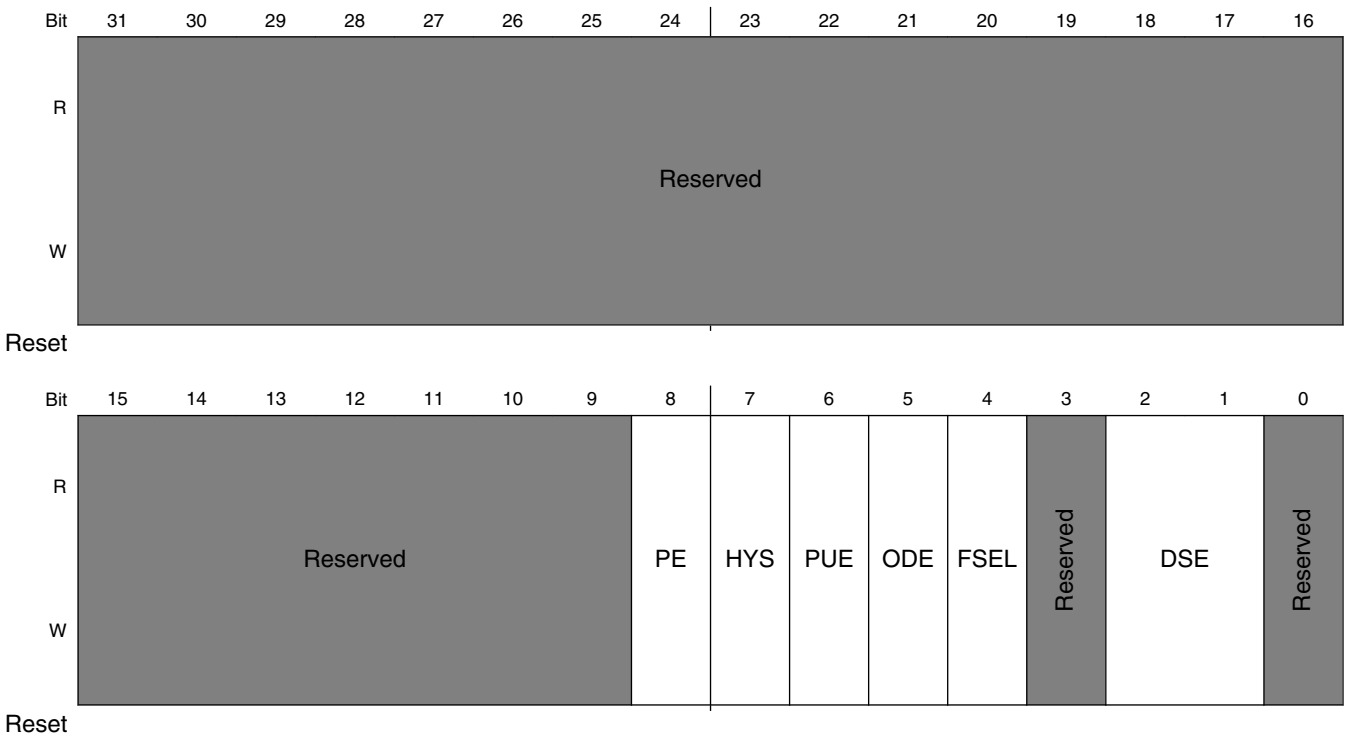
IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_DATA03 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_DATA03 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA03 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.214 SW_PAD_CTL_PAD_NAND_DATA04 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA04)

SW_PAD_CTL Register

Address: 3033_0000h base + 368h offset = 3033_0368h



IOMUXC_SW_PAD_CTL_PAD_NAND_DATA04 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_DATA04 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_DATA04 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_DATA04

Table continues on the next page...

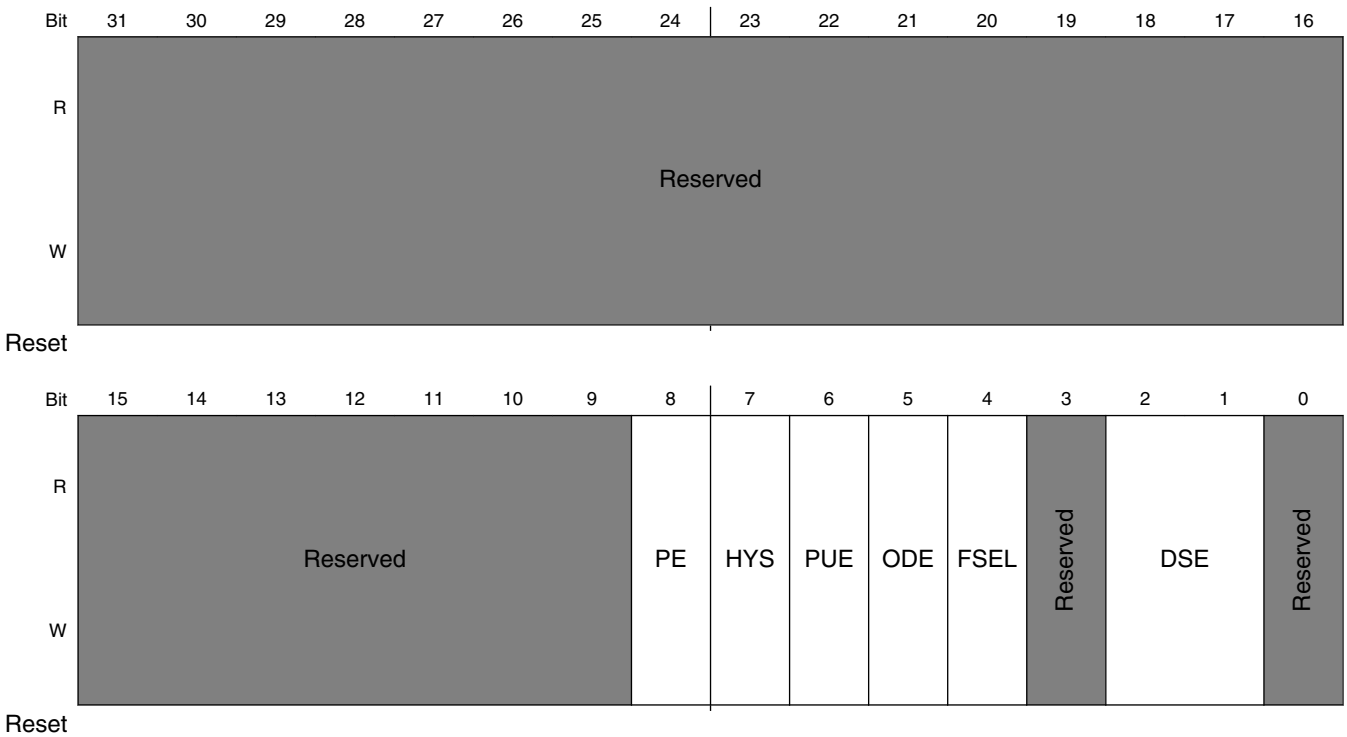
IOMUXC_SW_PAD_CTL_PAD_NAND_DATA04 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_DATA04 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_DATA04 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA04 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.215 SW_PAD_CTL_PAD_NAND_DATA05 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA05)

SW_PAD_CTL Register

Address: 3033_0000h base + 36Ch offset = 3033_036Ch



IOMUXC_SW_PAD_CTL_PAD_NAND_DATA05 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_DATA05 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_DATA05 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_DATA05

Table continues on the next page...

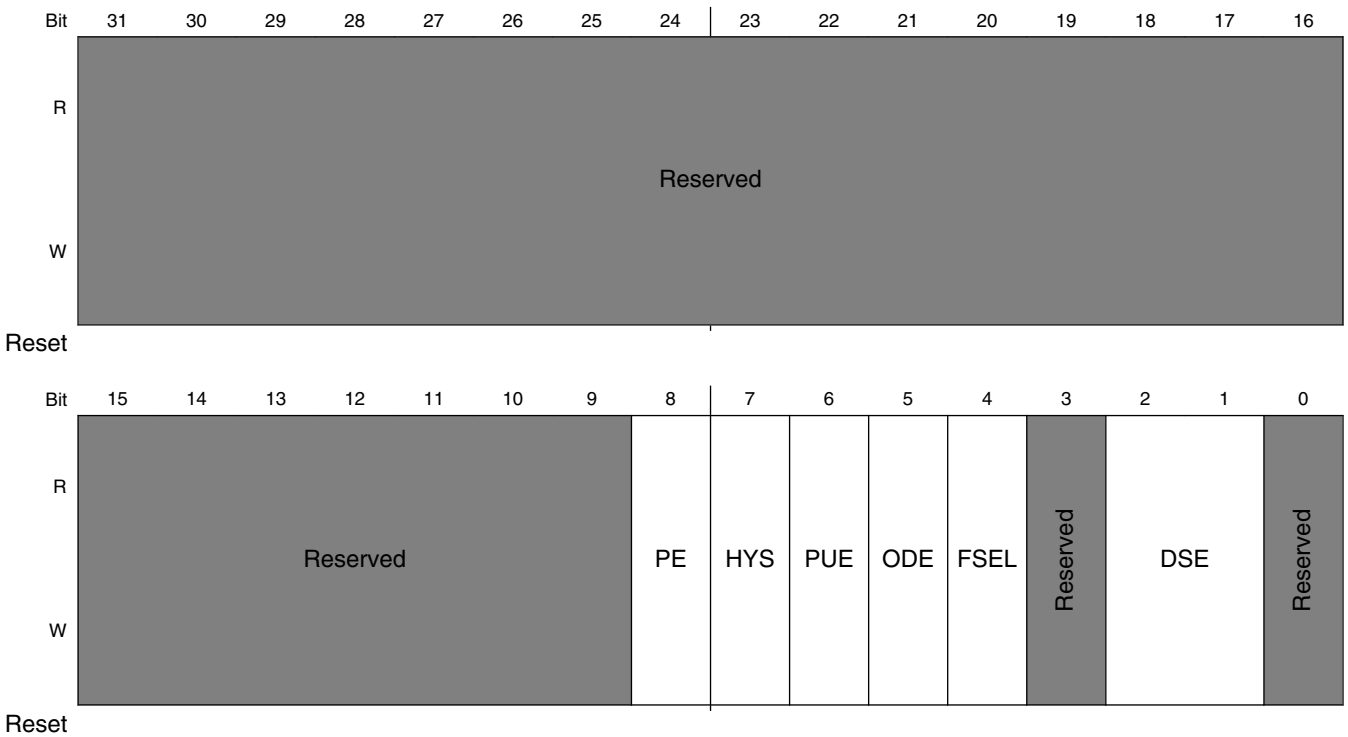
IOMUXC_SW_PAD_CTL_PAD_NAND_DATA05 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_DATA05 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_DATA05 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA05 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.216 SW_PAD_CTL_PAD_NAND_DATA06 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA06)

SW_PAD_CTL Register

Address: 3033_0000h base + 370h offset = 3033_0370h



IOMUXC_SW_PAD_CTL_PAD_NAND_DATA06 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_DATA06 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_DATA06 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_DATA06

Table continues on the next page...

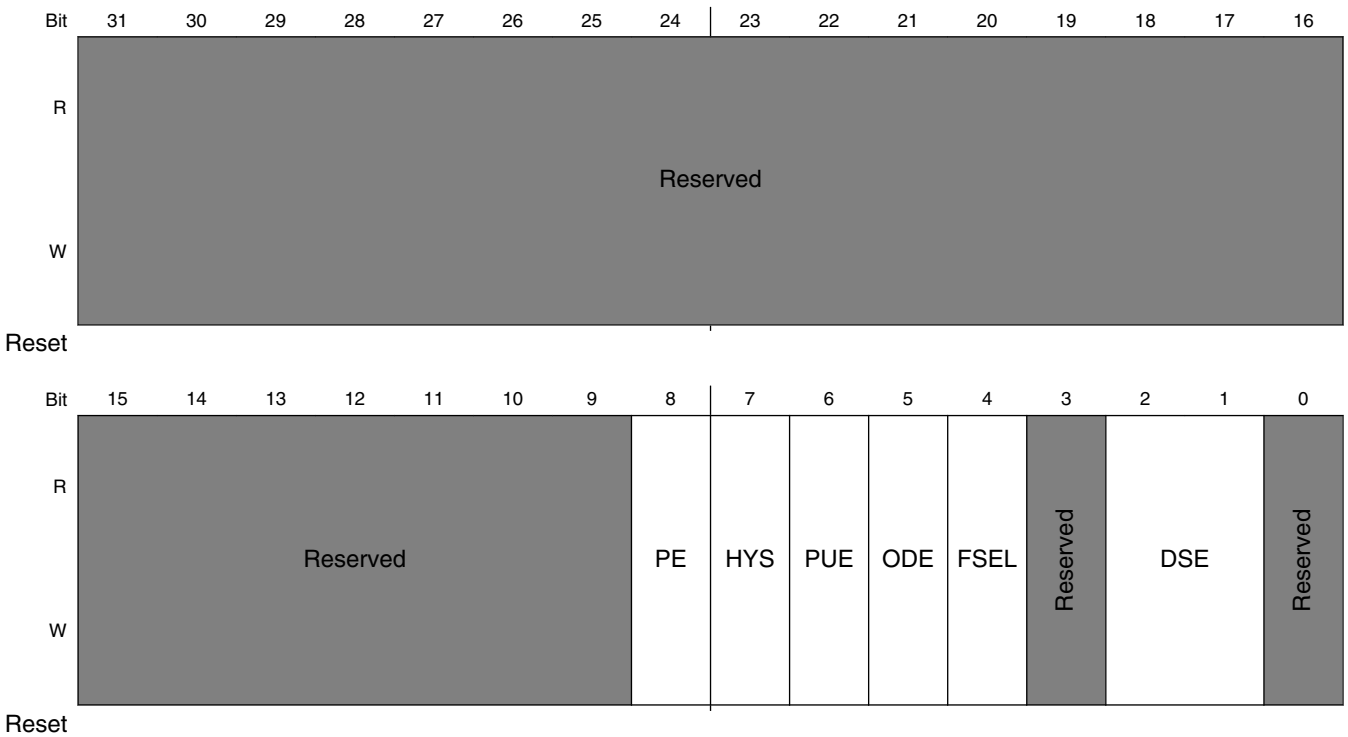
IOMUXC_SW_PAD_CTL_PAD_NAND_DATA06 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_DATA06 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_DATA06 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA06 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.217 SW_PAD_CTL_PAD_NAND_DATA07 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07)

SW_PAD_CTL Register

Address: 3033_0000h base + 374h offset = 3033_0374h



IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_DATA07 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_DATA07 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_DATA07

Table continues on the next page...

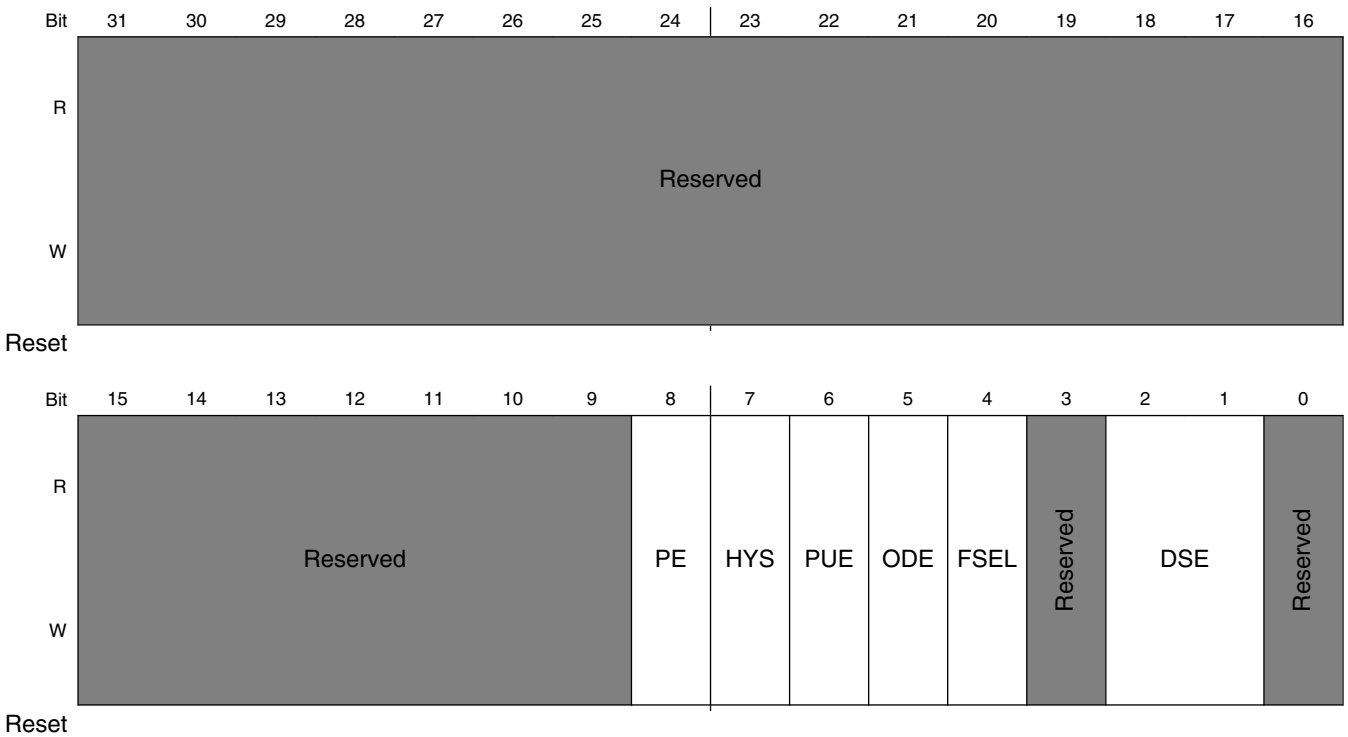
IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_DATA07 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_DATA07 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA07 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.218 SW_PAD_CTL_PAD_NAND_DQS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DQS)

SW_PAD_CTL Register

Address: 3033_0000h base + 378h offset = 3033_0378h



IOMUXC_SW_PAD_CTL_PAD_NAND_DQS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_DQS 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_DQS 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_DQS

Table continues on the next page...

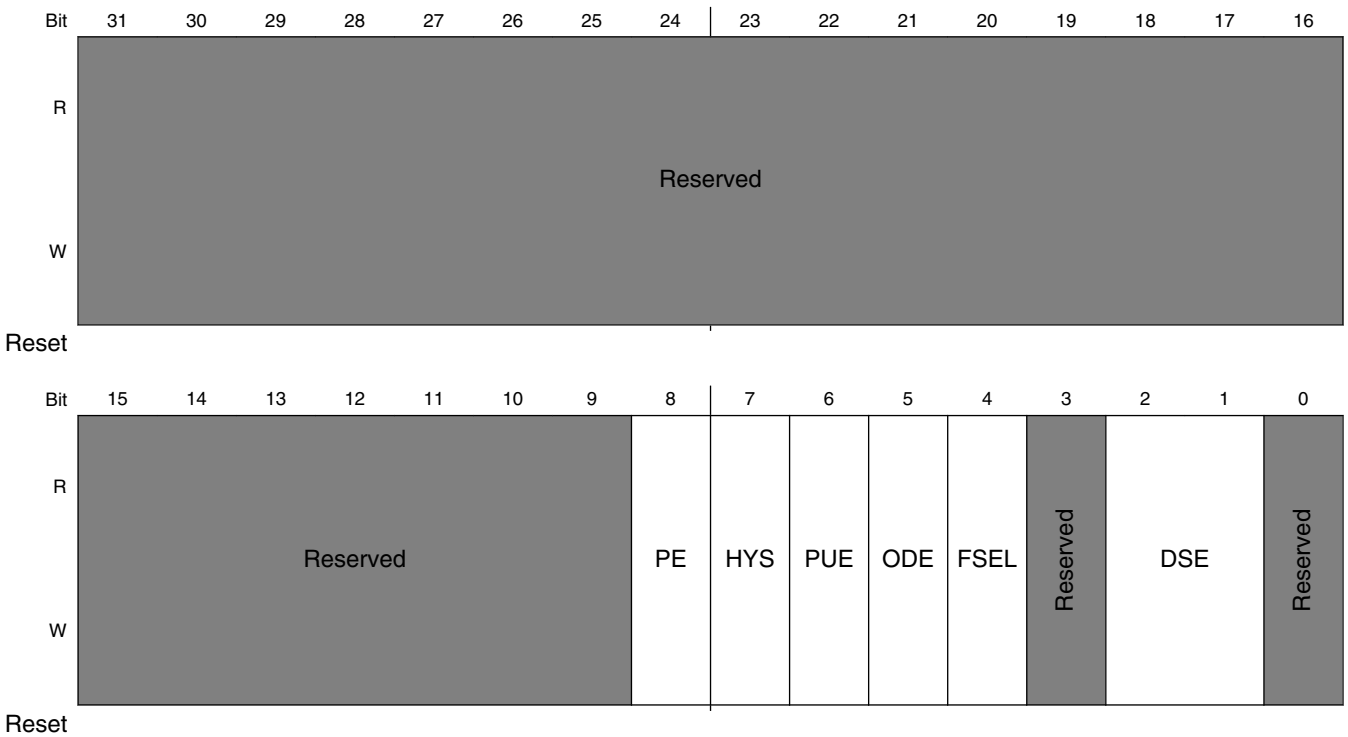
IOMUXC_SW_PAD_CTL_PAD_NAND_DQS field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_DQS 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_DQS 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_DQS 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.219 SW_PAD_CTL_PAD_NAND_RE_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_RE_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 37Ch offset = 3033_037Ch



IOMUXC_SW_PAD_CTL_PAD_NAND_RE_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_RE_B 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_RE_B 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_RE_B

Table continues on the next page...

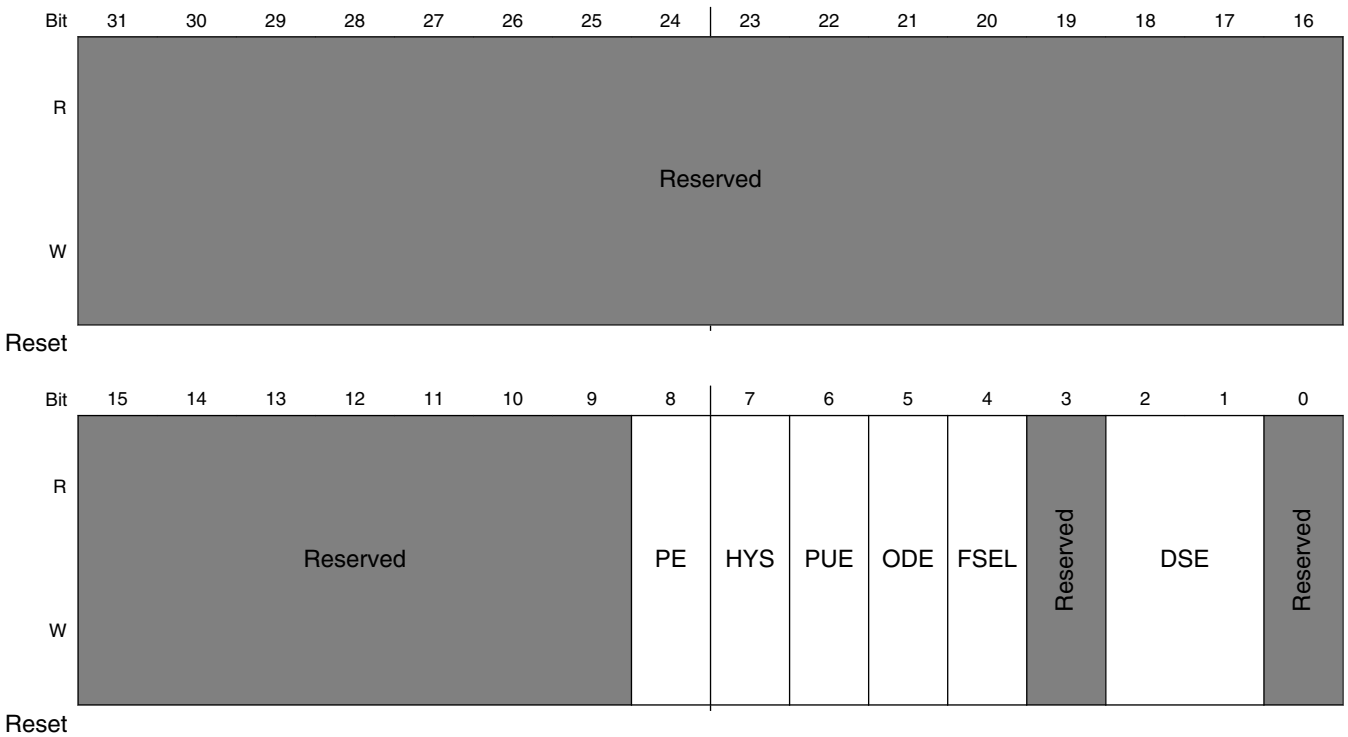
IOMUXC_SW_PAD_CTL_PAD_NAND_RE_B field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_RE_B 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_RE_B 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_RE_B 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.220 SW_PAD_CTL_PAD_NAND_READY_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_READY_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 380h offset = 3033_0380h



IOMUXC_SW_PAD_CTL_PAD_NAND_READY_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_READY_B 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_READY_B 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_READY_B

Table continues on the next page...

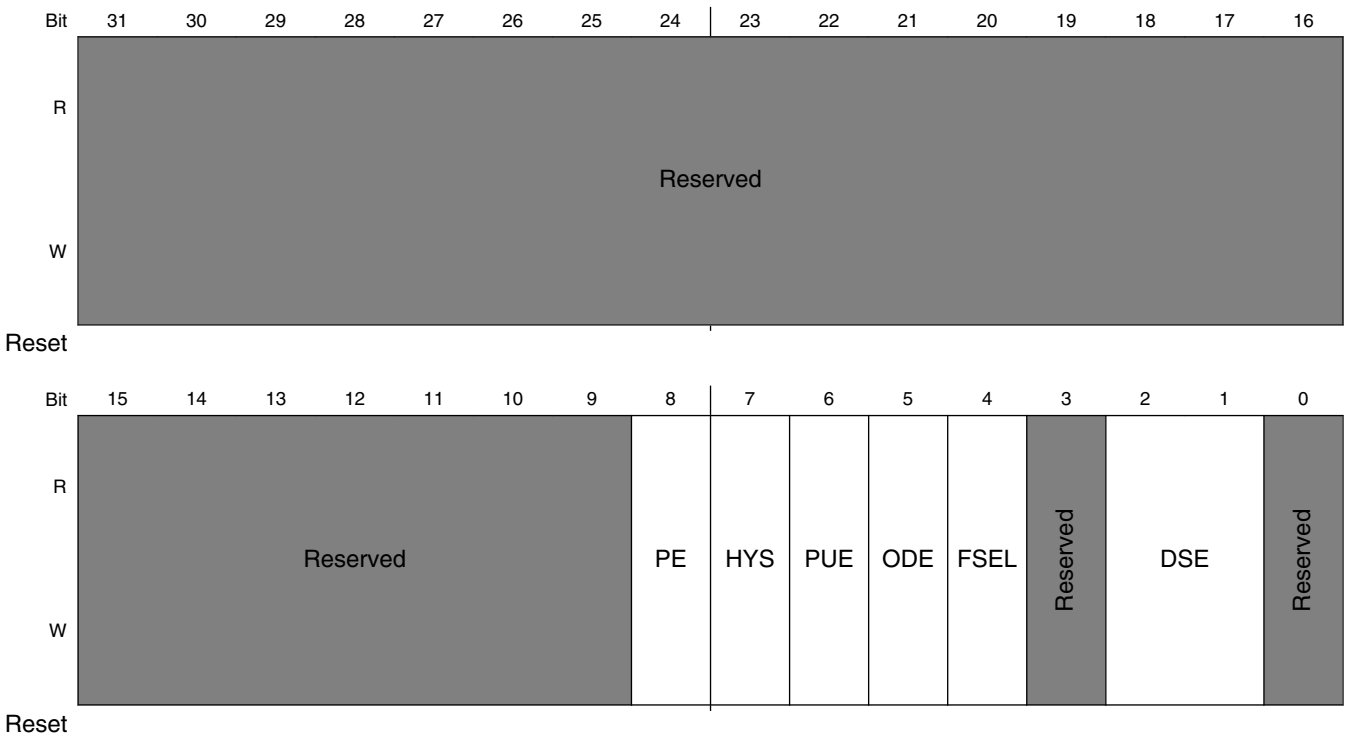
IOMUXC_SW_PAD_CTL_PAD_NAND_READY_B field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_READY_B 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_READY_B 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_READY_B 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.221 SW_PAD_CTL_PAD_NAND_WE_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_WE_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 384h offset = 3033_0384h



IOMUXC_SW_PAD_CTL_PAD_NAND_WE_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_WE_B 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_WE_B 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_WE_B

Table continues on the next page...

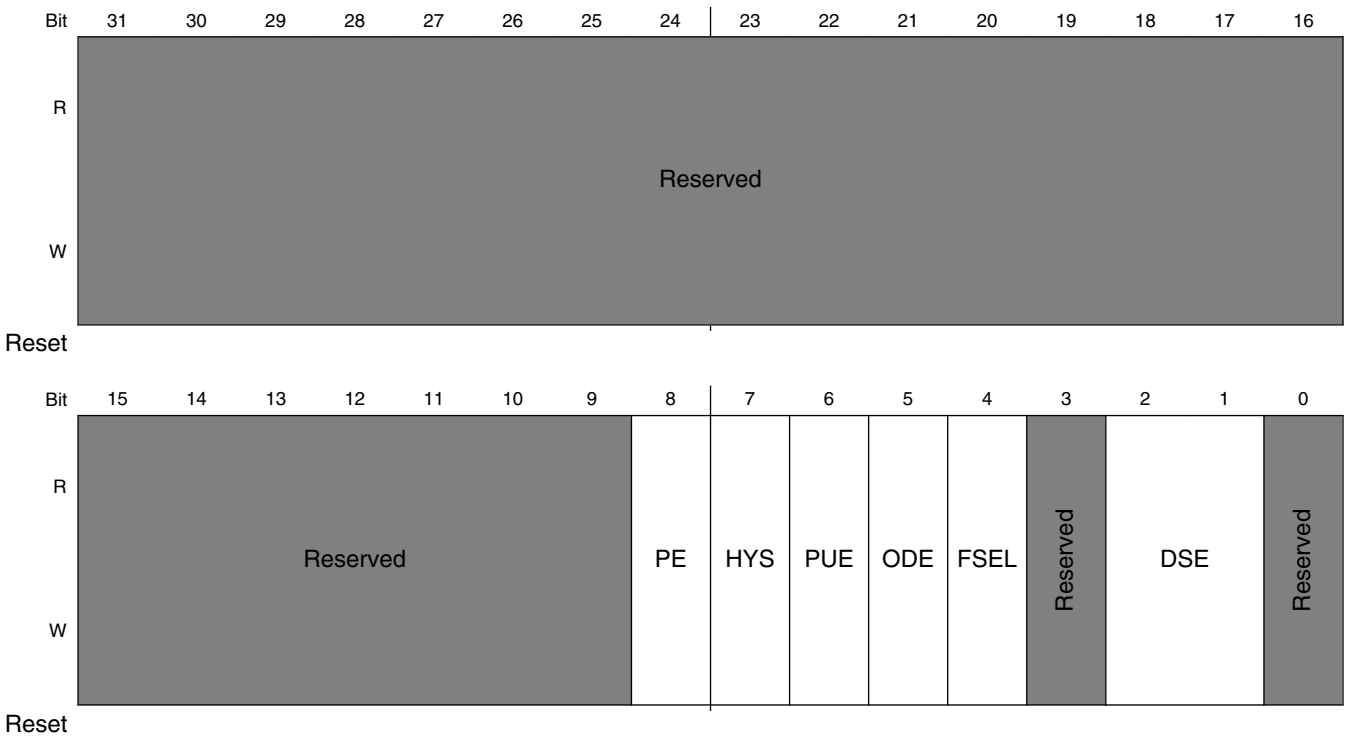
IOMUXC_SW_PAD_CTL_PAD_NAND_WE_B field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_WE_B 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_WE_B 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_WE_B 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.222 SW_PAD_CTL_PAD_NAND_WP_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_WP_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 388h offset = 3033_0388h



IOMUXC_SW_PAD_CTL_PAD_NAND_WP_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: NAND_WP_B 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: NAND_WP_B 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: NAND_WP_B

Table continues on the next page...

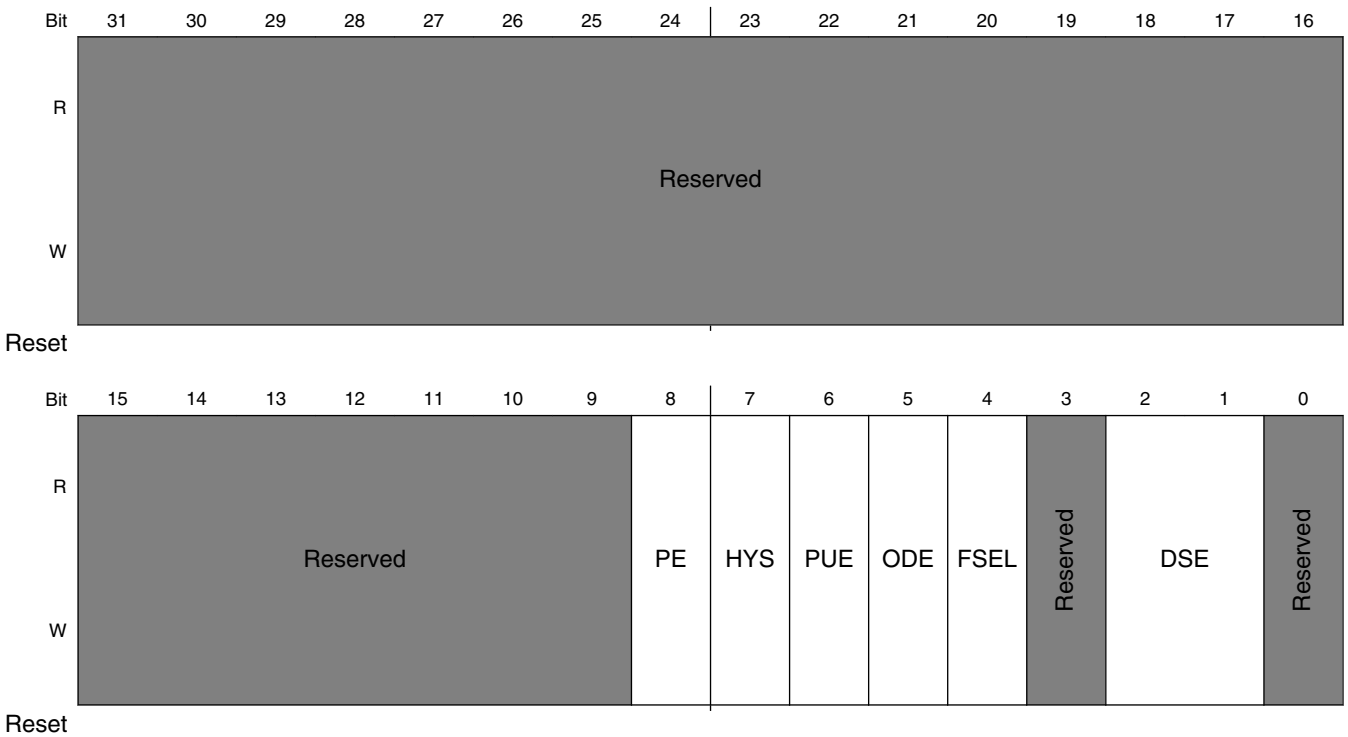
IOMUXC_SW_PAD_CTL_PAD_NAND_WP_B field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: NAND_WP_B 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: NAND_WP_B 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: NAND_WP_B 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.223 SW_PAD_CTL_PAD_SAI5_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 38Ch offset = 3033_038Ch



IOMUXC_SW_PAD_CTL_PAD_SAI5_RXFS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI5_RXFS 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI5_RXFS 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI5_RXFS

Table continues on the next page...

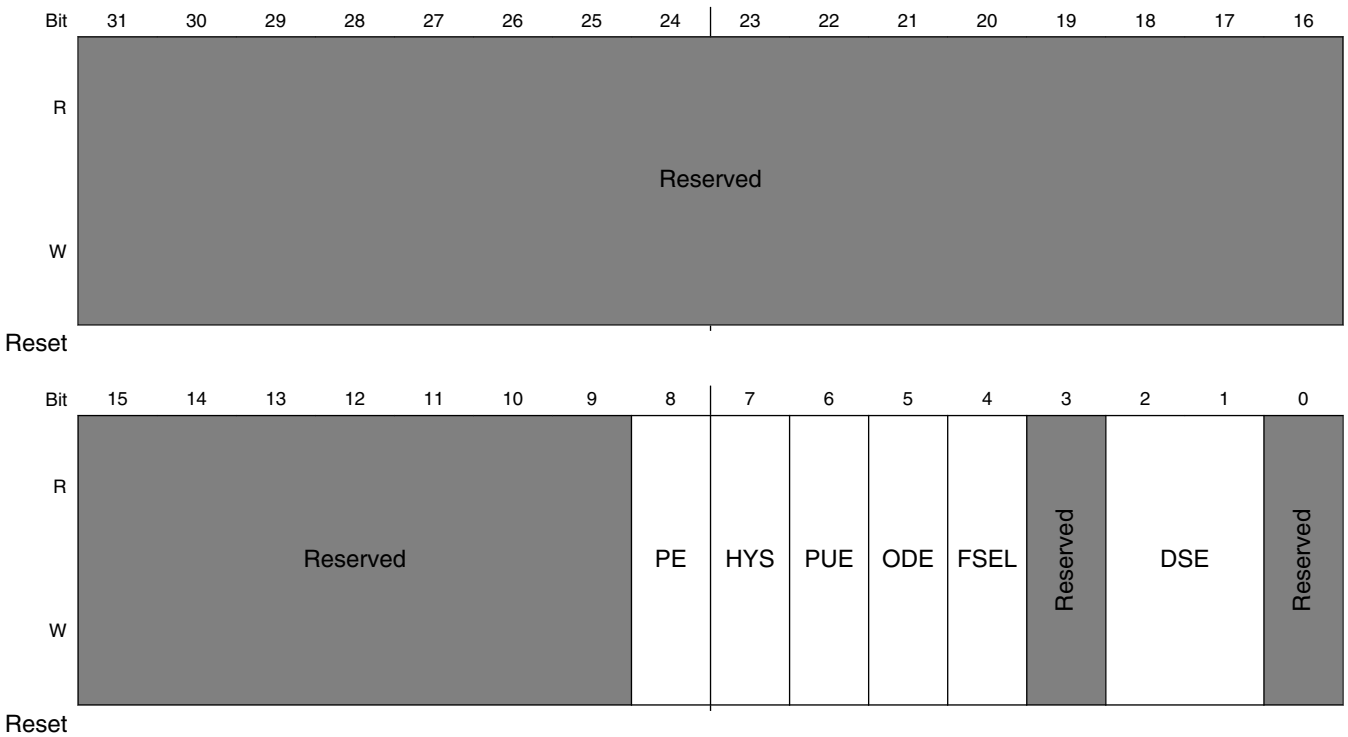
IOMUXC_SW_PAD_CTL_PAD_SAI5_RXFS field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI5_RXFS 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI5_RXFS 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI5_RXFS 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.224 SW_PAD_CTL_PAD_SAI5_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 390h offset = 3033_0390h



IOMUXC_SW_PAD_CTL_PAD_SAI5_RXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI5_RXC 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI5_RXC 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI5_RXC

Table continues on the next page...

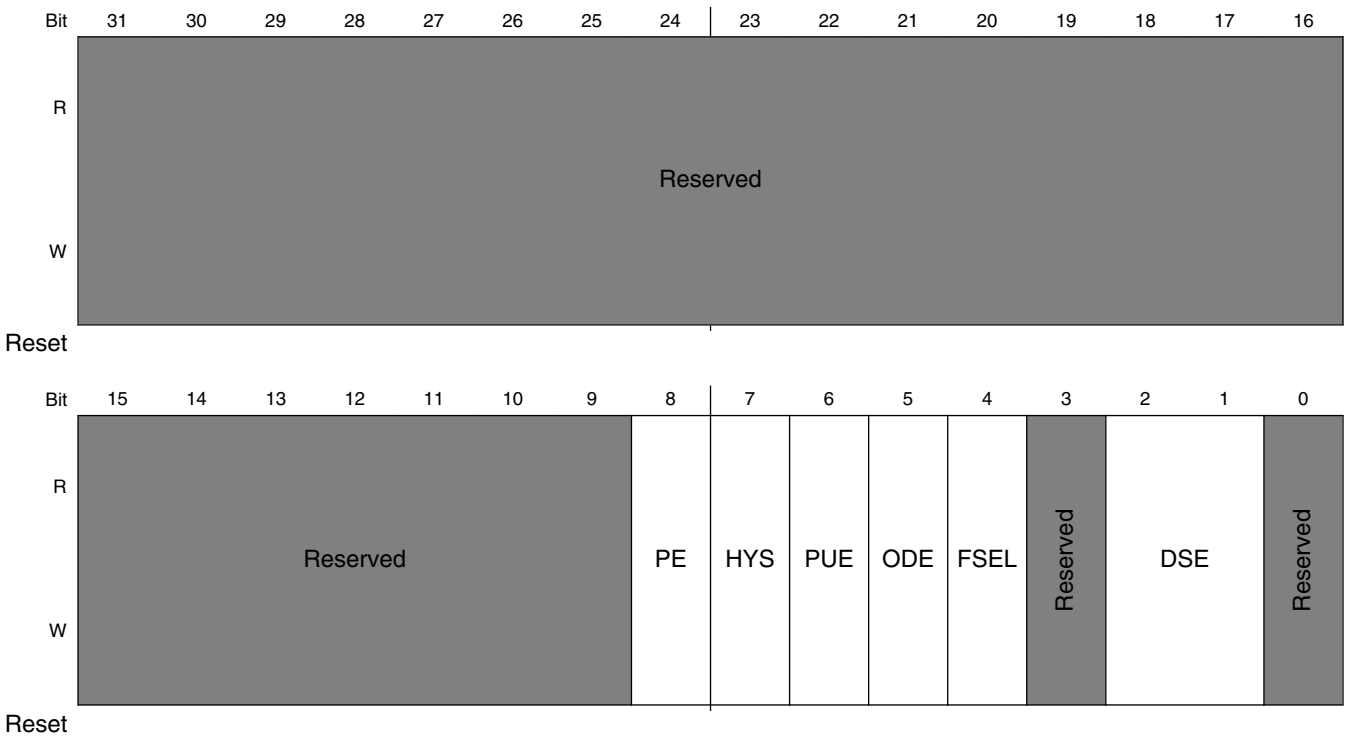
IOMUXC_SW_PAD_CTL_PAD_SAI5_RXC field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI5_RXC 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI5_RXC 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI5_RXC 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.225 SW_PAD_CTL_PAD_SAI5_RXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 394h offset = 3033_0394h



IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI5_RXD0 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI5_RXD0 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI5_RXD0

Table continues on the next page...

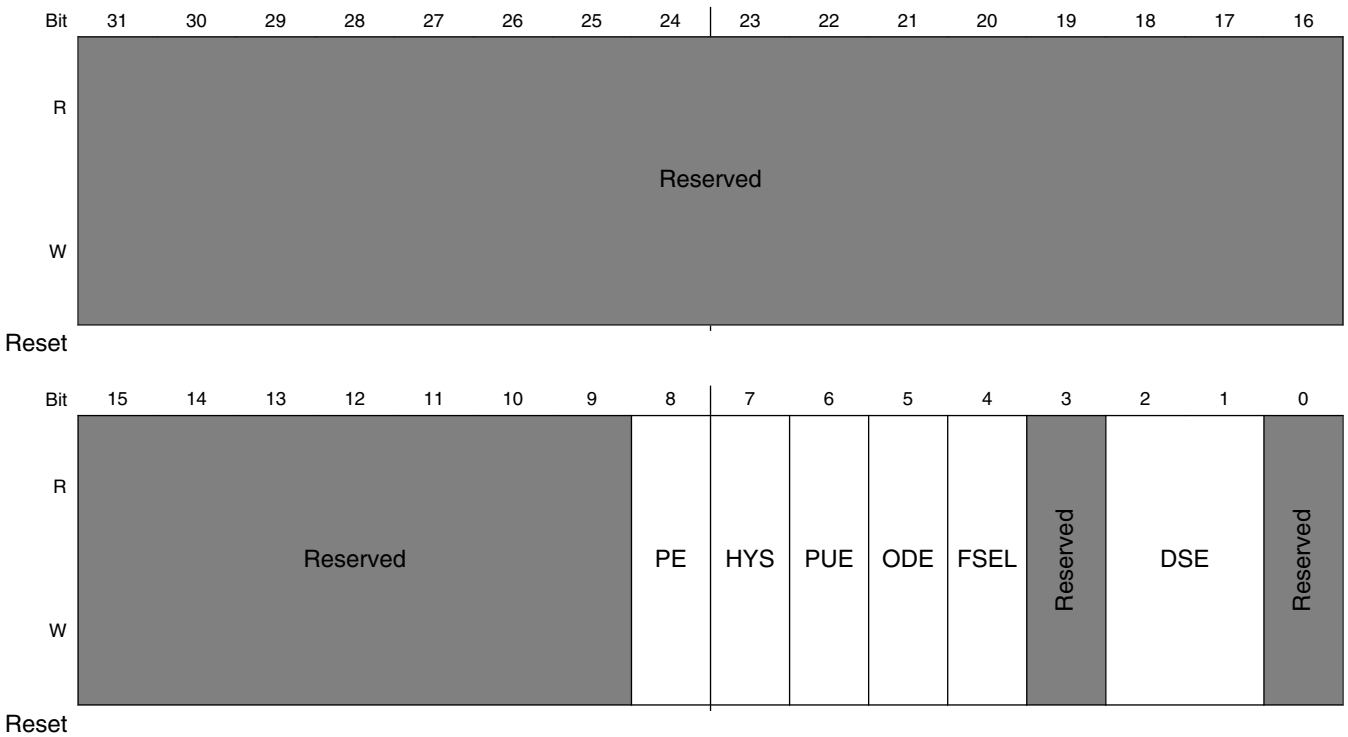
IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD0 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI5_RXD0 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI5_RXD0 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI5_RXD0 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.226 SW_PAD_CTL_PAD_SAI5_RXD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD1)

SW_PAD_CTL Register

Address: 3033_0000h base + 398h offset = 3033_0398h



IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI5_RXD1 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI5_RXD1 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI5_RXD1

Table continues on the next page...

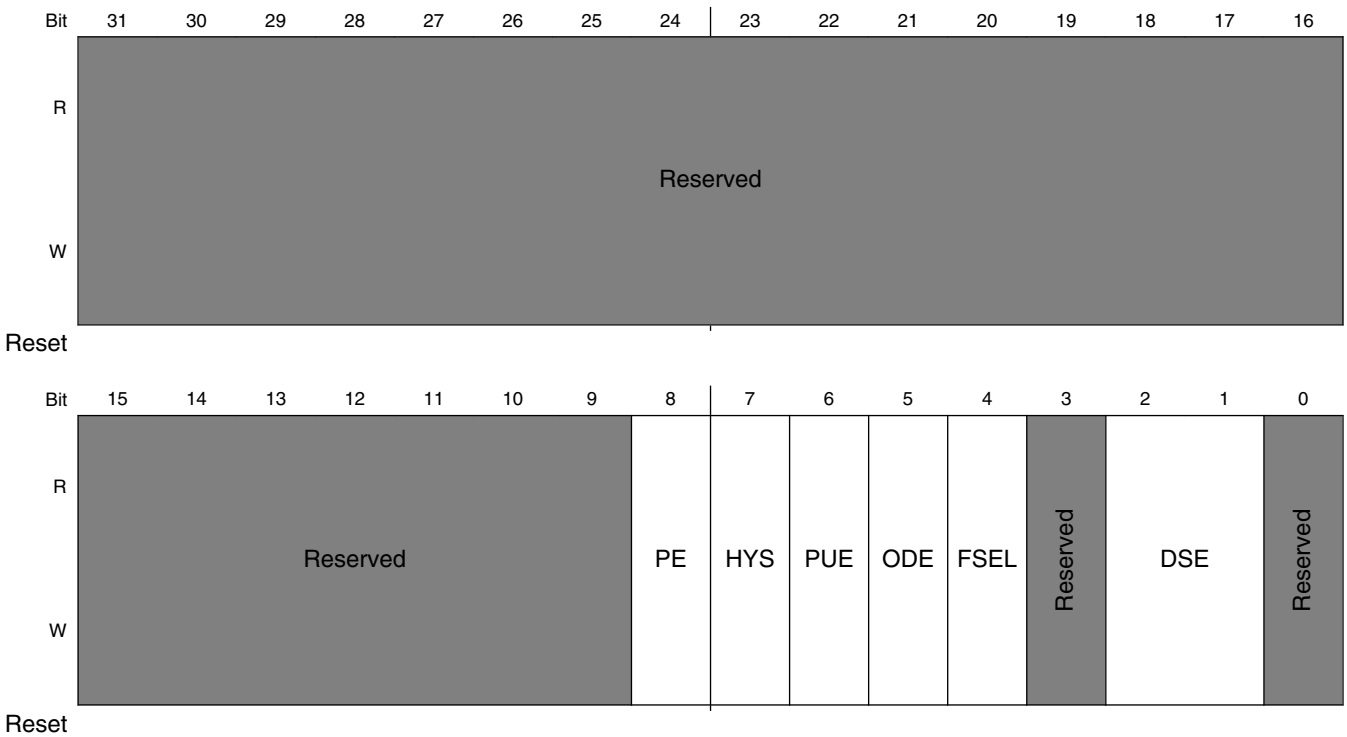
IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD1 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI5_RXD1 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI5_RXD1 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI5_RXD1 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.227 SW_PAD_CTL_PAD_SAI5_RXD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD2)

SW_PAD_CTL Register

Address: 3033_0000h base + 39Ch offset = 3033_039Ch



IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI5_RXD2 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI5_RXD2 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI5_RXD2

Table continues on the next page...

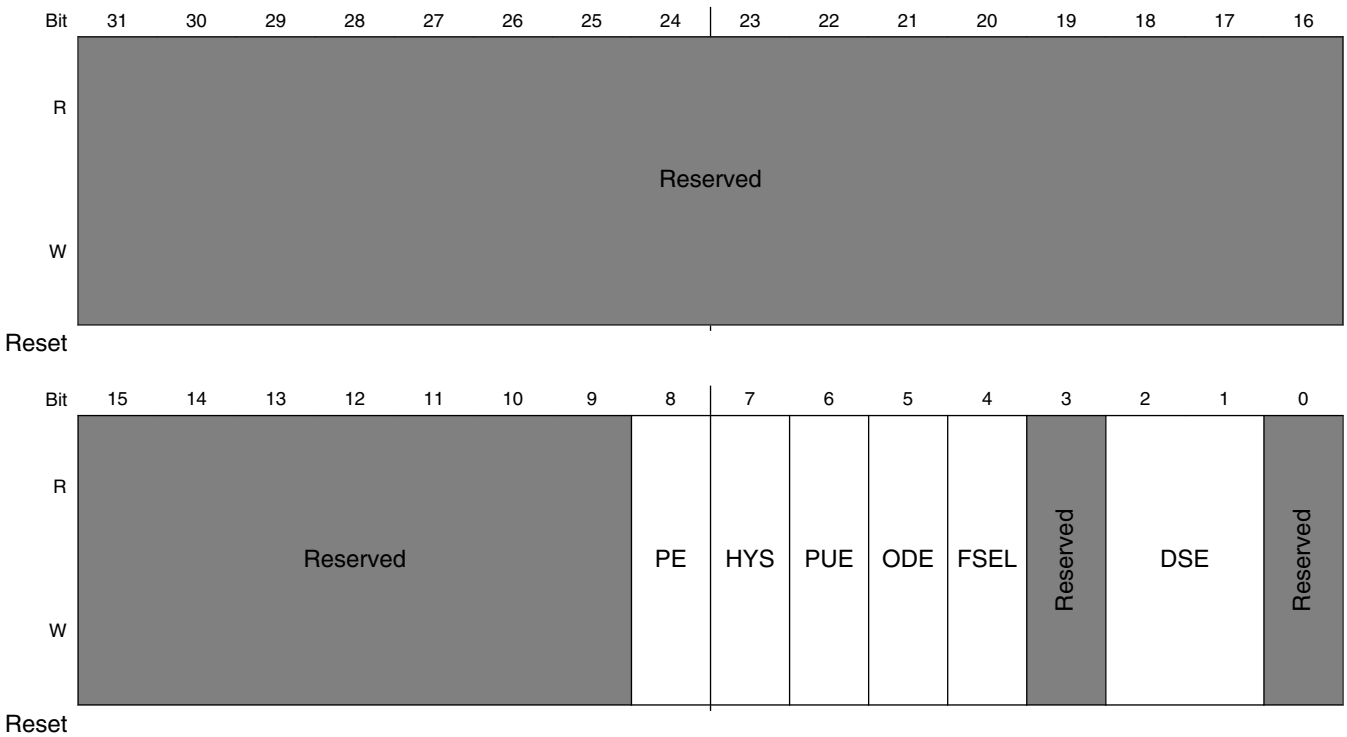
IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD2 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI5_RXD2 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI5_RXD2 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI5_RXD2 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.228 SW_PAD_CTL_PAD_SAI5_RXD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD3)

SW_PAD_CTL Register

Address: 3033_0000h base + 3A0h offset = 3033_03A0h



IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI5_RXD3 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI5_RXD3 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI5_RXD3

Table continues on the next page...

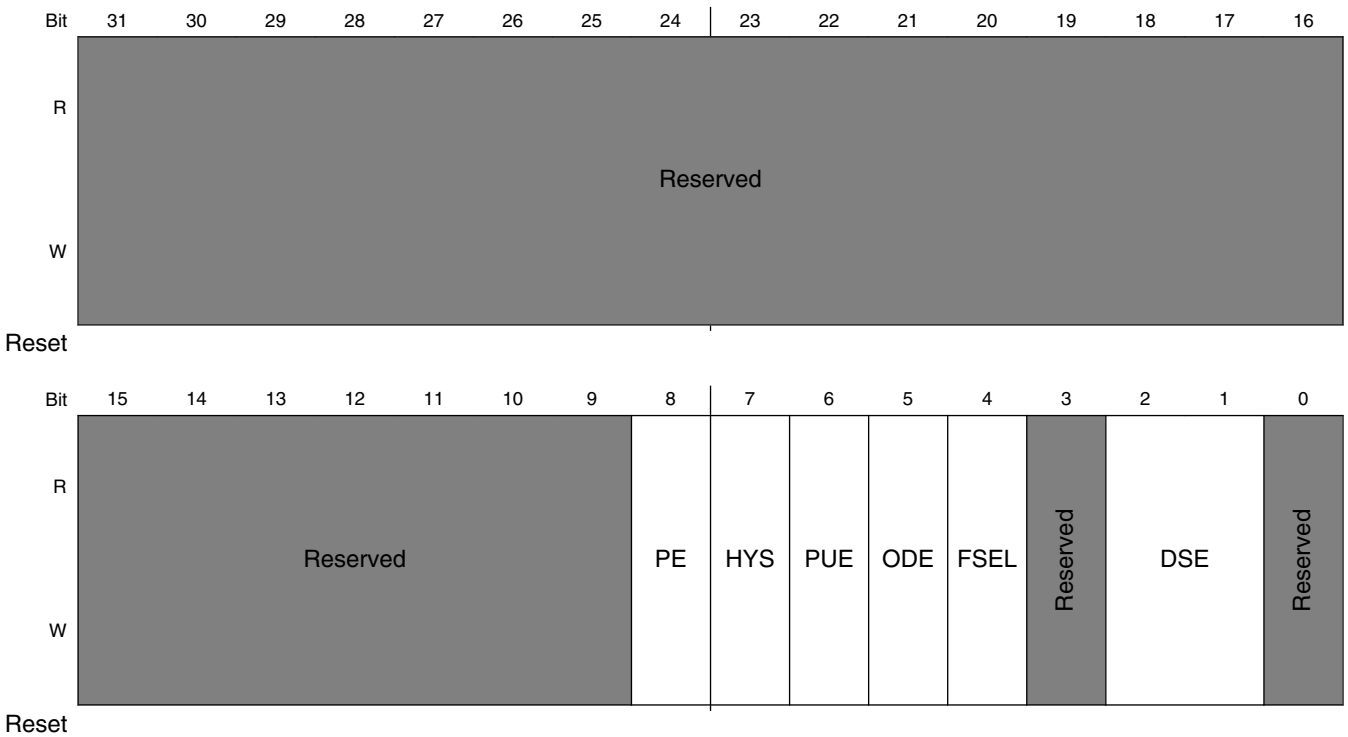
IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD3 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI5_RXD3 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI5_RXD3 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI5_RXD3 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.229 SW_PAD_CTL_PAD_SAI5_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_MCLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 3A4h offset = 3033_03A4h



IOMUXC_SW_PAD_CTL_PAD_SAI5_MCLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI5_MCLK 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI5_MCLK 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI5_MCLK

Table continues on the next page...

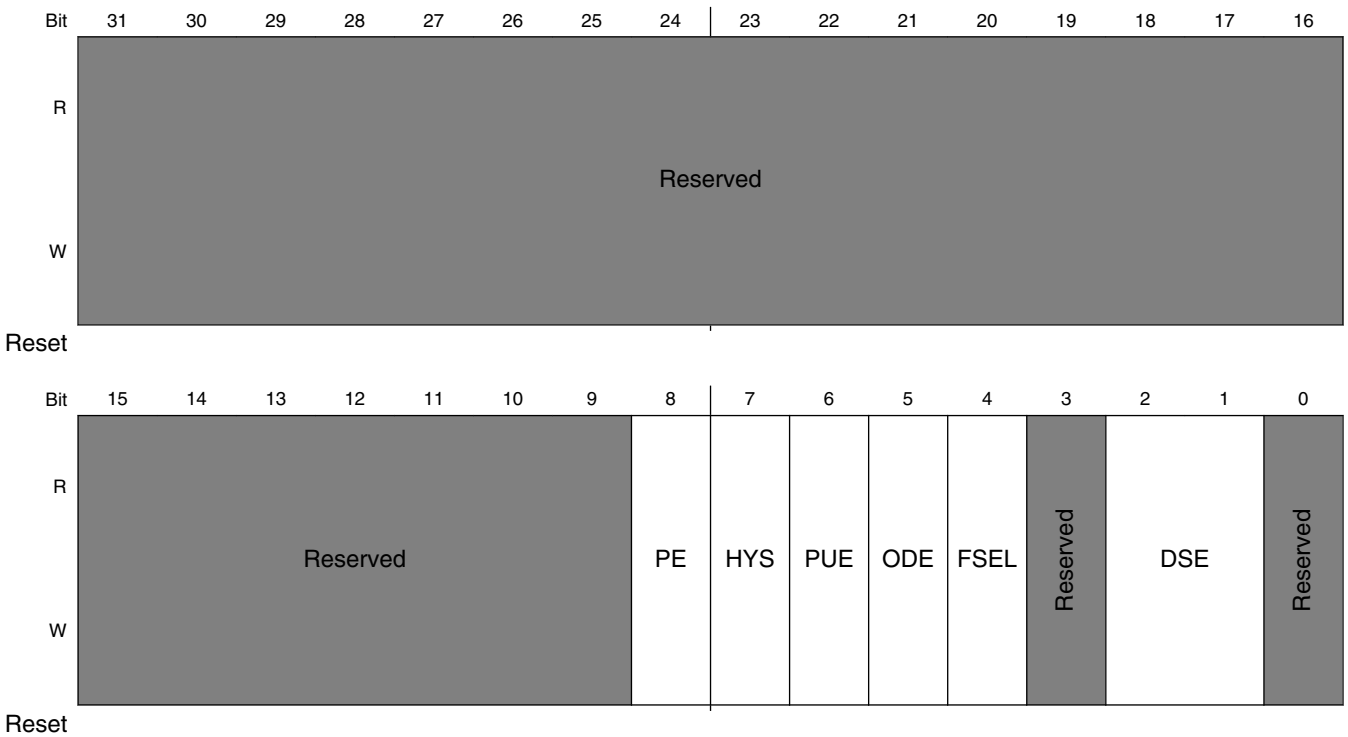
IOMUXC_SW_PAD_CTL_PAD_SAI5_MCLK field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI5_MCLK 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI5_MCLK 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI5_MCLK 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.230 SW_PAD_CTL_PAD_SAI1_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 3A8h offset = 3033_03A8h



IOMUXC_SW_PAD_CTL_PAD_SAI1_RXFS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_RXFS 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_RXFS 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_RXFS

Table continues on the next page...

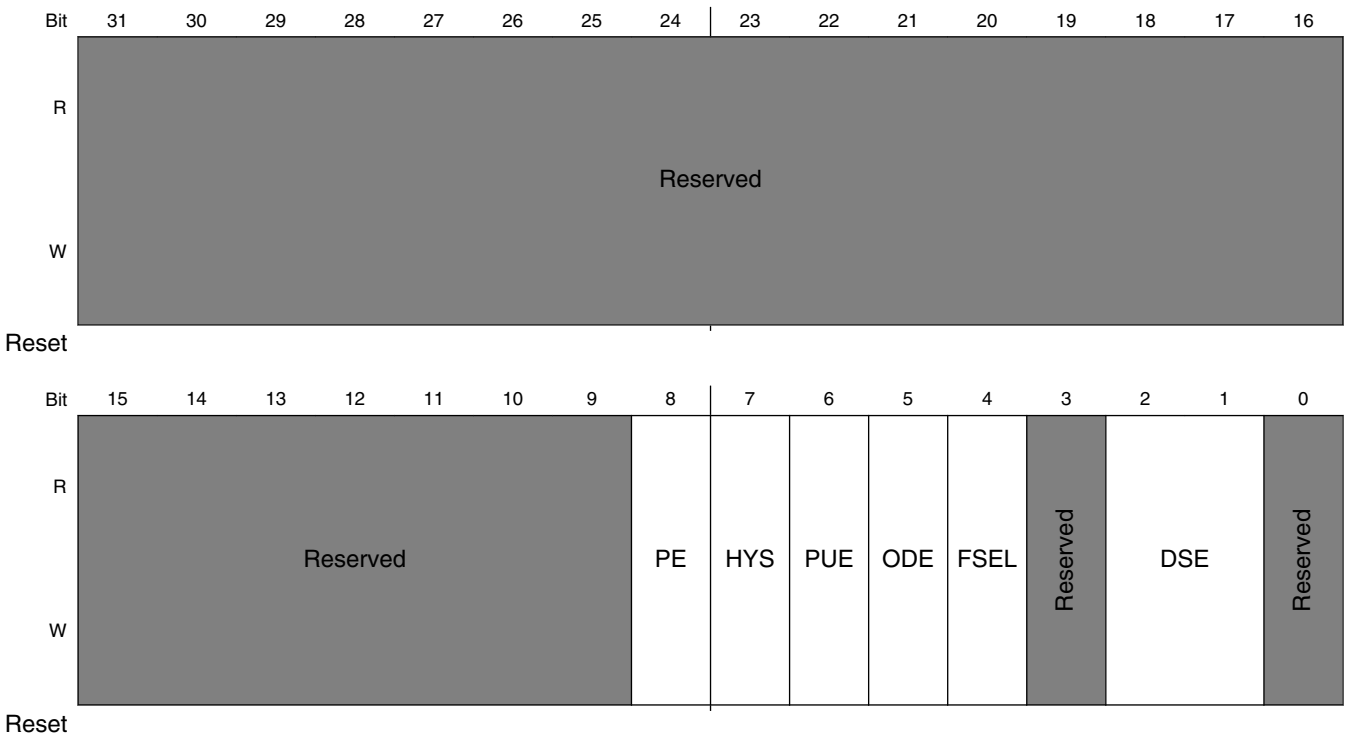
IOMUXC_SW_PAD_CTL_PAD_SAI1_RXFS field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_RXFS 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_RXFS 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXFS 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.231 SW_PAD_CTL_PAD_SAI1_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 3ACh offset = 3033_03ACh



IOMUXC_SW_PAD_CTL_PAD_SAI1_RXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_RXC 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_RXC 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_RXC

Table continues on the next page...

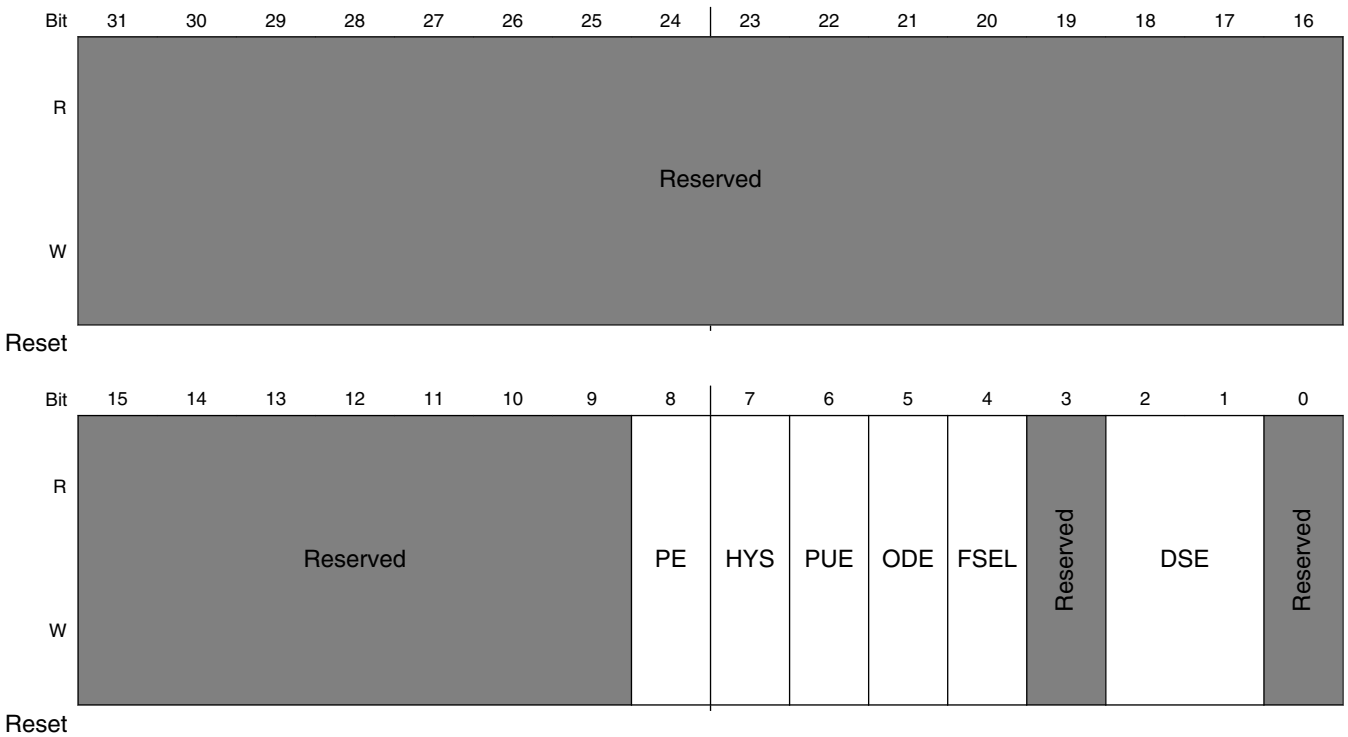
IOMUXC_SW_PAD_CTL_PAD_SAI1_RXC field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_RXC 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_RXC 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXC 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.232 SW_PAD_CTL_PAD_SAI1_RXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 3B0h offset = 3033_03B0h



IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_RXD0 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_RXD0 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_RXD0

Table continues on the next page...

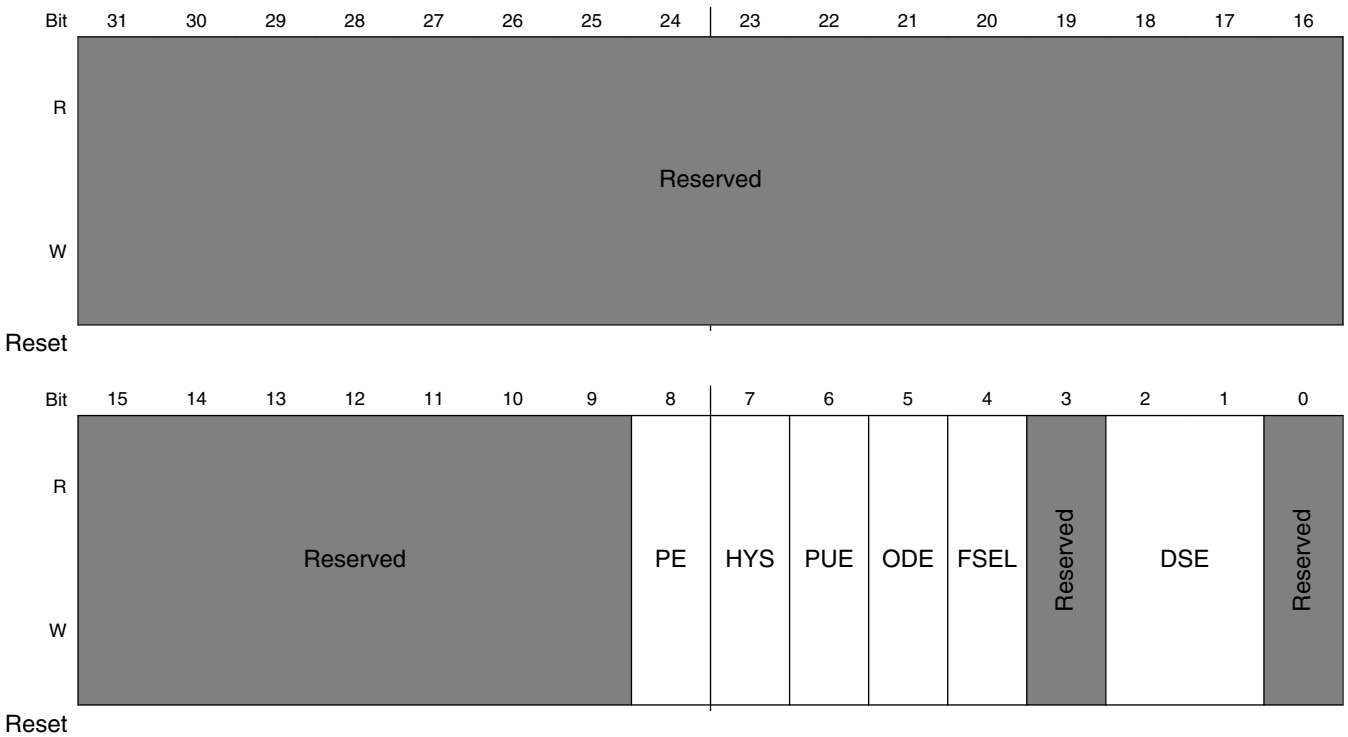
IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD0 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_RXD0 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_RXD0 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD0 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.233 SW_PAD_CTL_PAD_SAI1_RXD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD1)

SW_PAD_CTL Register

Address: 3033_0000h base + 3B4h offset = 3033_03B4h



IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_RXD1 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_RXD1 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_RXD1

Table continues on the next page...

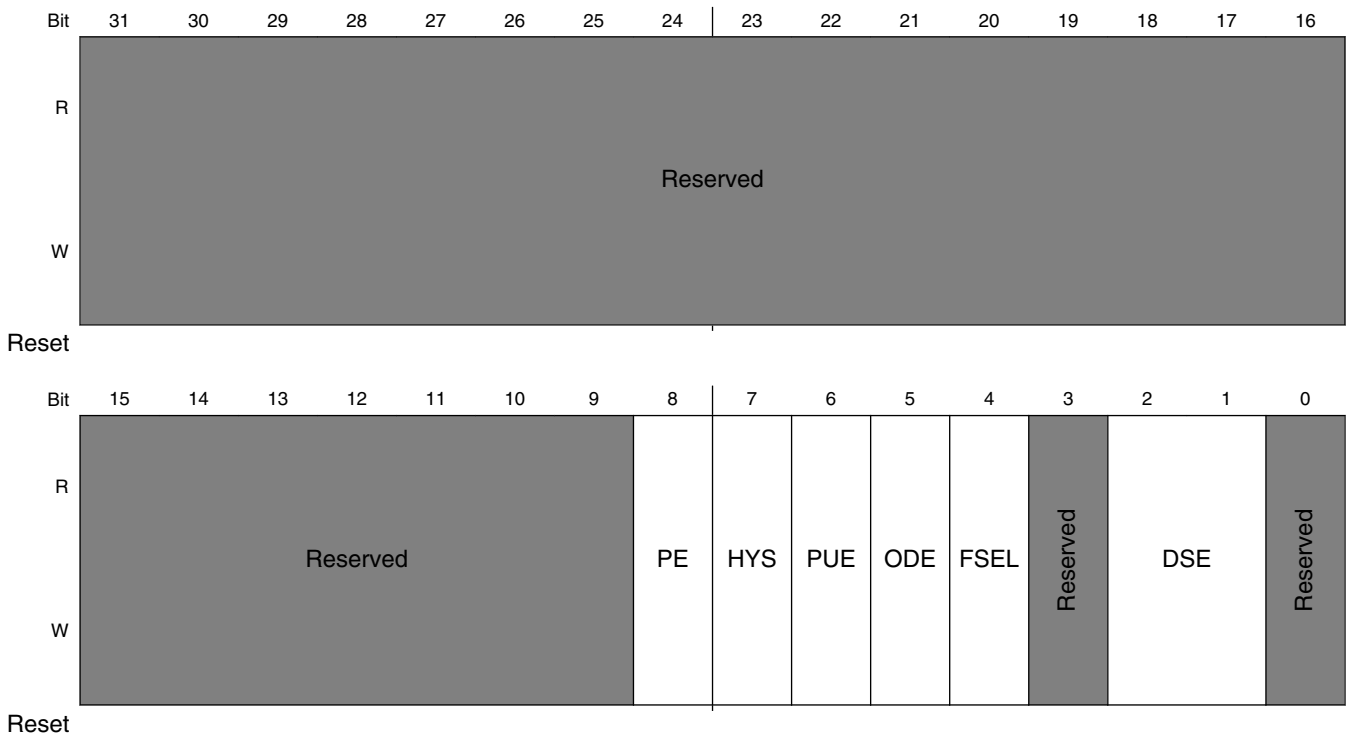
IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD1 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_RXD1 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_RXD1 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD1 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.234 SW_PAD_CTL_PAD_SAI1_RXD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD2)

SW_PAD_CTL Register

Address: 3033_0000h base + 3B8h offset = 3033_03B8h



IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_RXD2 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_RXD2 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_RXD2

Table continues on the next page...

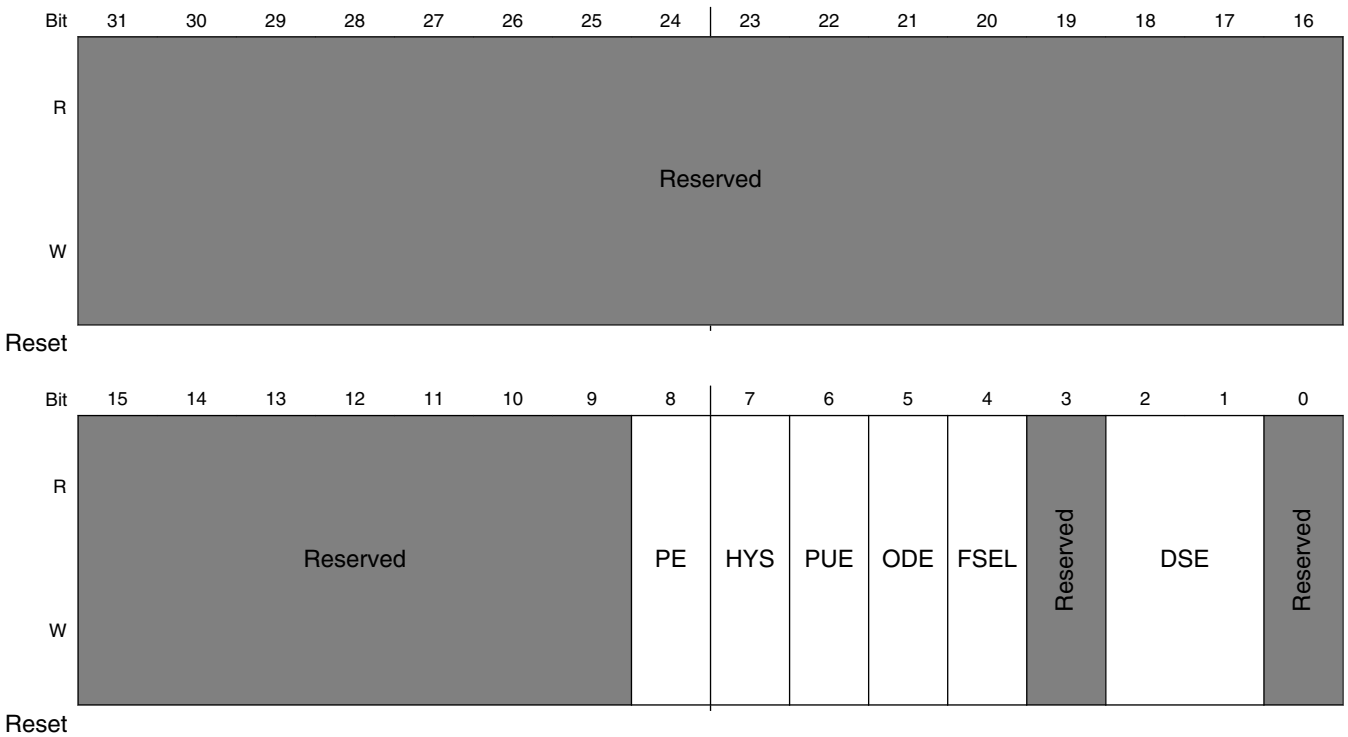
IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD2 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_RXD2 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_RXD2 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD2 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.235 SW_PAD_CTL_PAD_SAI1_RXD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD3)

SW_PAD_CTL Register

Address: 3033_0000h base + 3BCh offset = 3033_03BCh



IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_RXD3 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_RXD3 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_RXD3

Table continues on the next page...

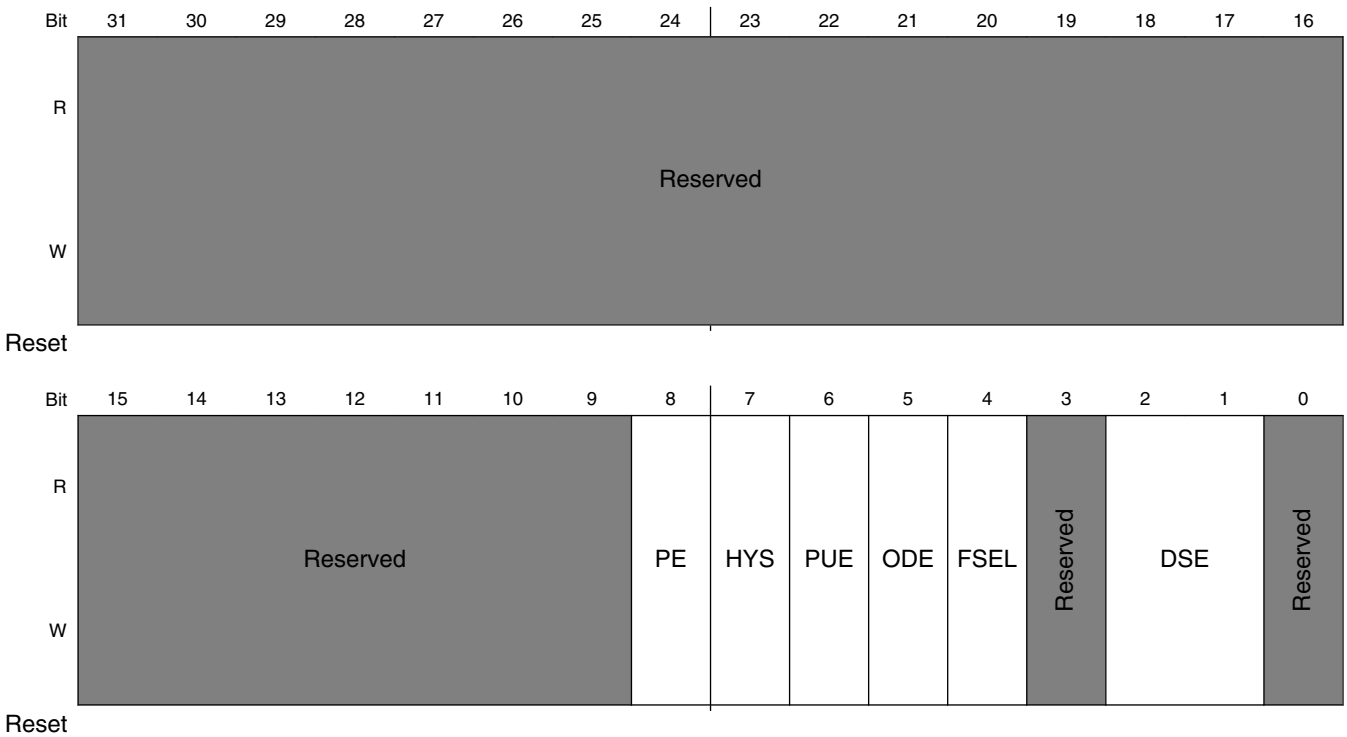
IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD3 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_RXD3 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_RXD3 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD3 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.236 SW_PAD_CTL_PAD_SAI1_RXD4 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD4)

SW_PAD_CTL Register

Address: 3033_0000h base + 3C0h offset = 3033_03C0h



IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD4 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_RXD4 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_RXD4 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_RXD4

Table continues on the next page...

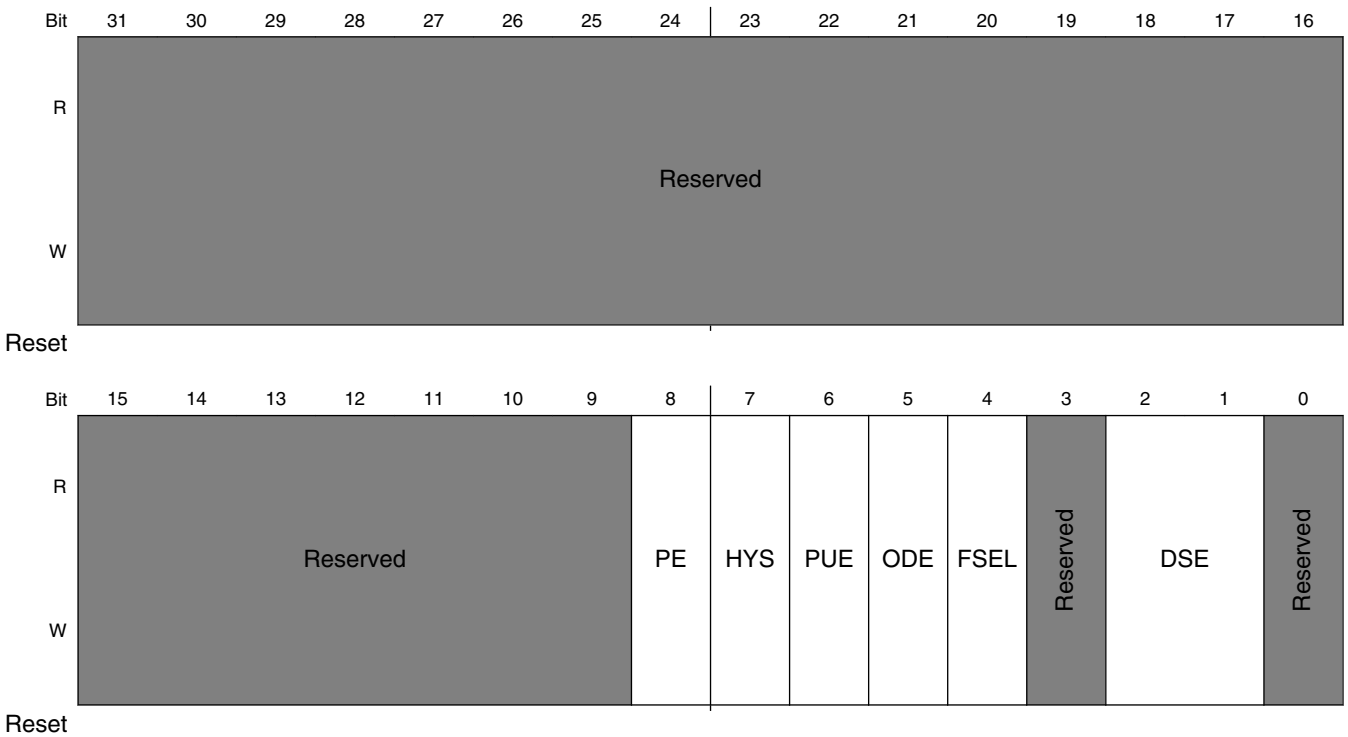
IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD4 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_RXD4 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_RXD4 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD4 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.237 SW_PAD_CTL_PAD_SAI1_RXD5 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD5)

SW_PAD_CTL Register

Address: 3033_0000h base + 3C4h offset = 3033_03C4h



IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD5 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_RXD5 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_RXD5 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_RXD5

Table continues on the next page...

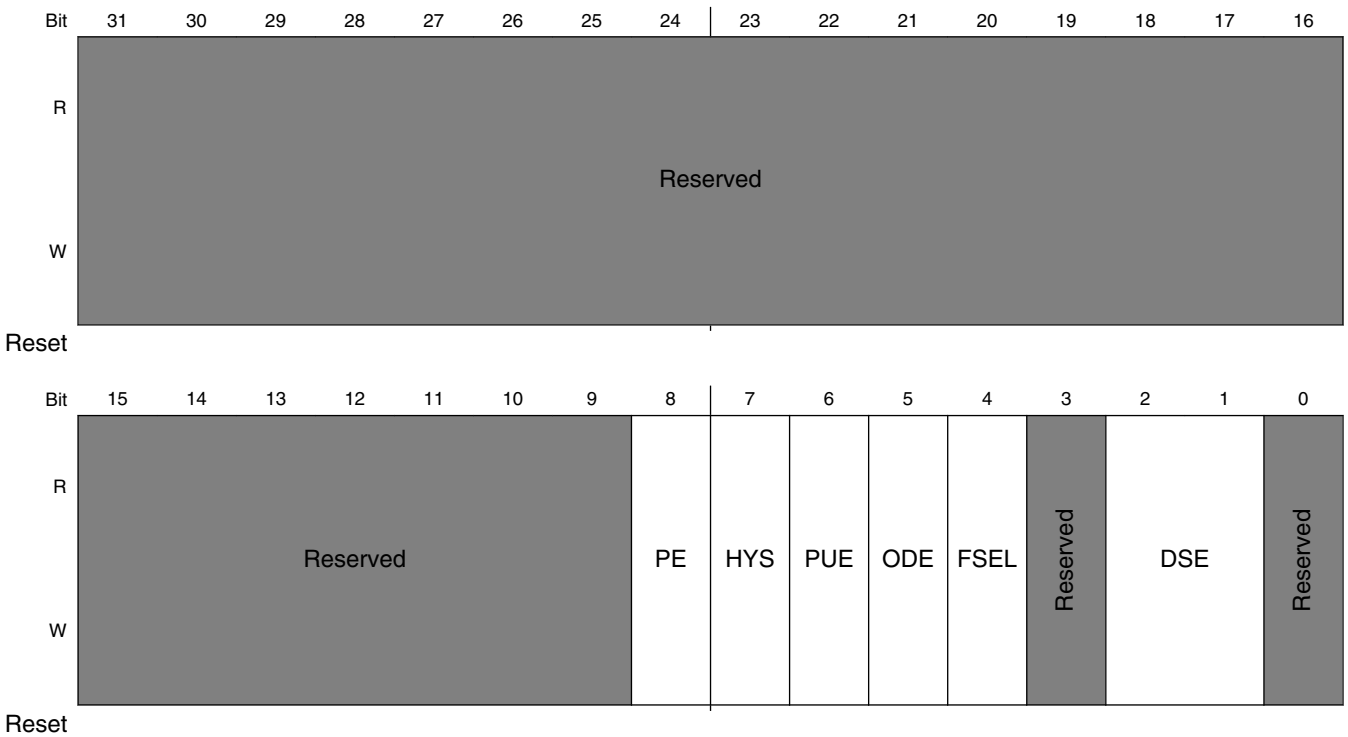
IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD5 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_RXD5 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_RXD5 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD5 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.238 SW_PAD_CTL_PAD_SAI1_RXD6 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD6)

SW_PAD_CTL Register

Address: 3033_0000h base + 3C8h offset = 3033_03C8h



IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD6 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_RXD6 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_RXD6 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_RXD6

Table continues on the next page...

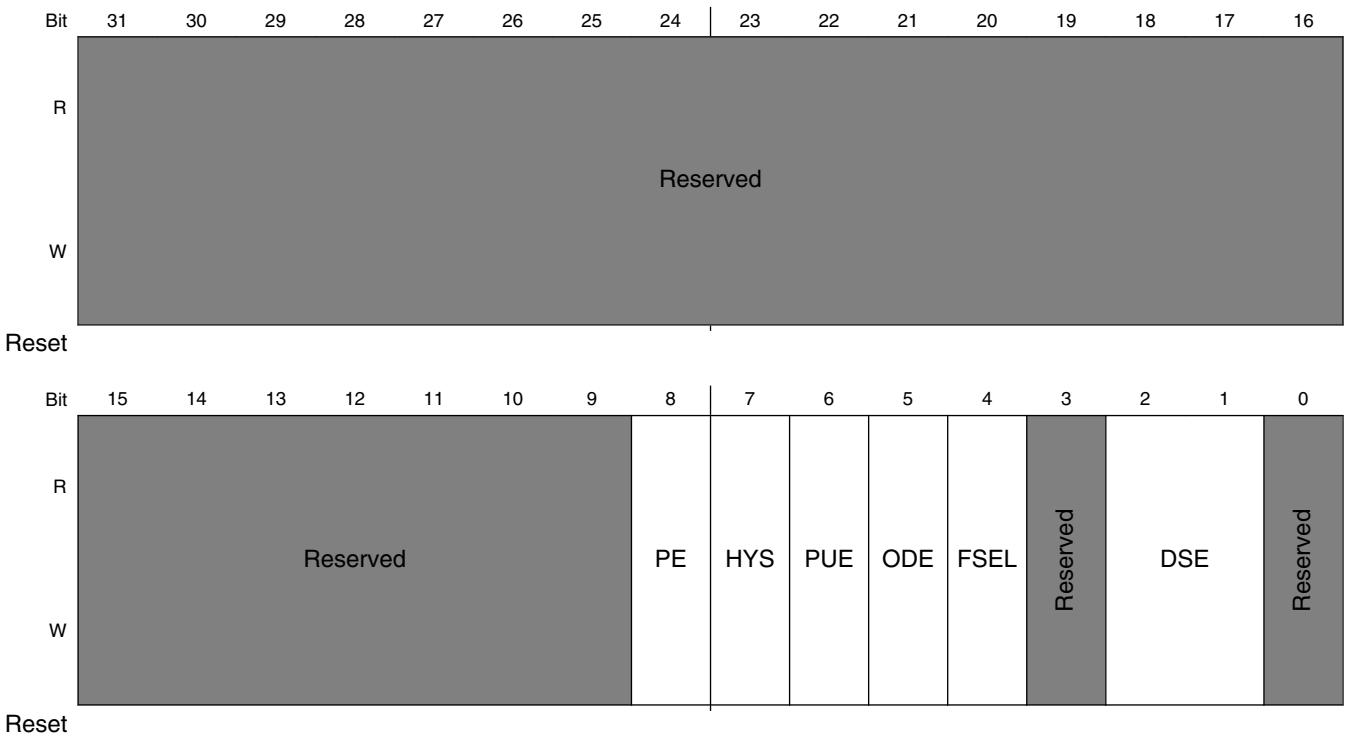
IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD6 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_RXD6 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_RXD6 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD6 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.239 SW_PAD_CTL_PAD_SAI1_RXD7 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD7)

SW_PAD_CTL Register

Address: 3033_0000h base + 3CCh offset = 3033_03CCh



IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD7 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_RXD7 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_RXD7 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_RXD7

Table continues on the next page...

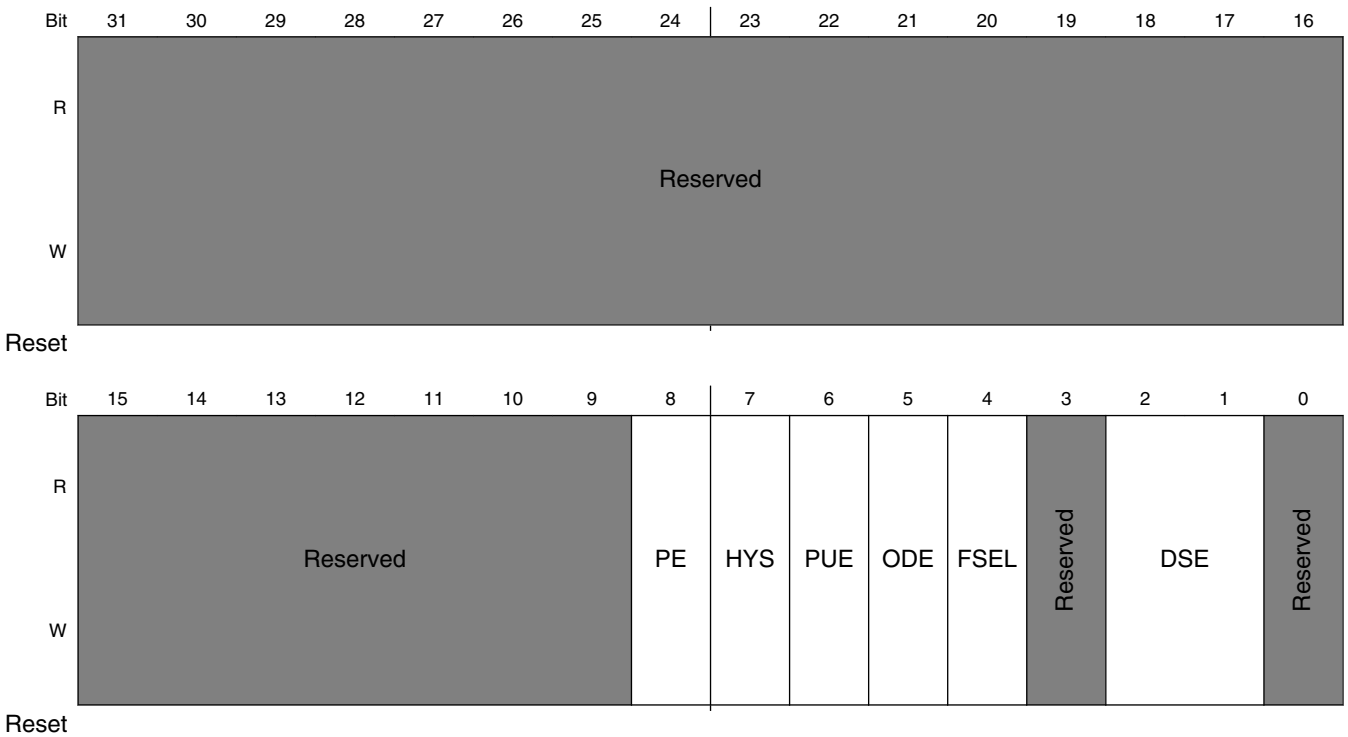
IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD7 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_RXD7 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_RXD7 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD7 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.240 SW_PAD_CTL_PAD_SAI1_TXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 3D0h offset = 3033_03D0h



IOMUXC_SW_PAD_CTL_PAD_SAI1_TXFS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_TXFS 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_TXFS 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_TXFS

Table continues on the next page...

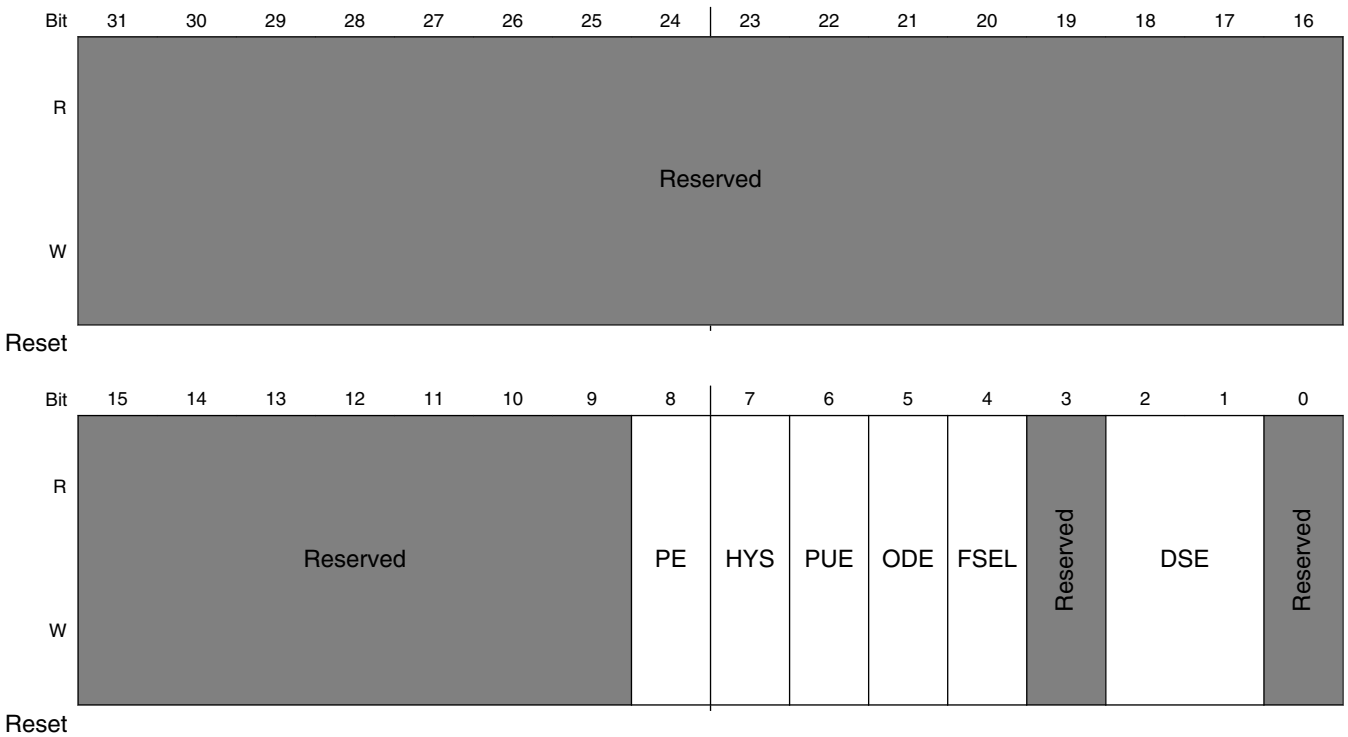
IOMUXC_SW_PAD_CTL_PAD_SAI1_TXFS field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_TXFS 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_TXFS 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXFS 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.241 SW_PAD_CTL_PAD_SAI1_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 3D4h offset = 3033_03D4h



IOMUXC_SW_PAD_CTL_PAD_SAI1_TXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_TXC 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_TXC 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_TXC

Table continues on the next page...

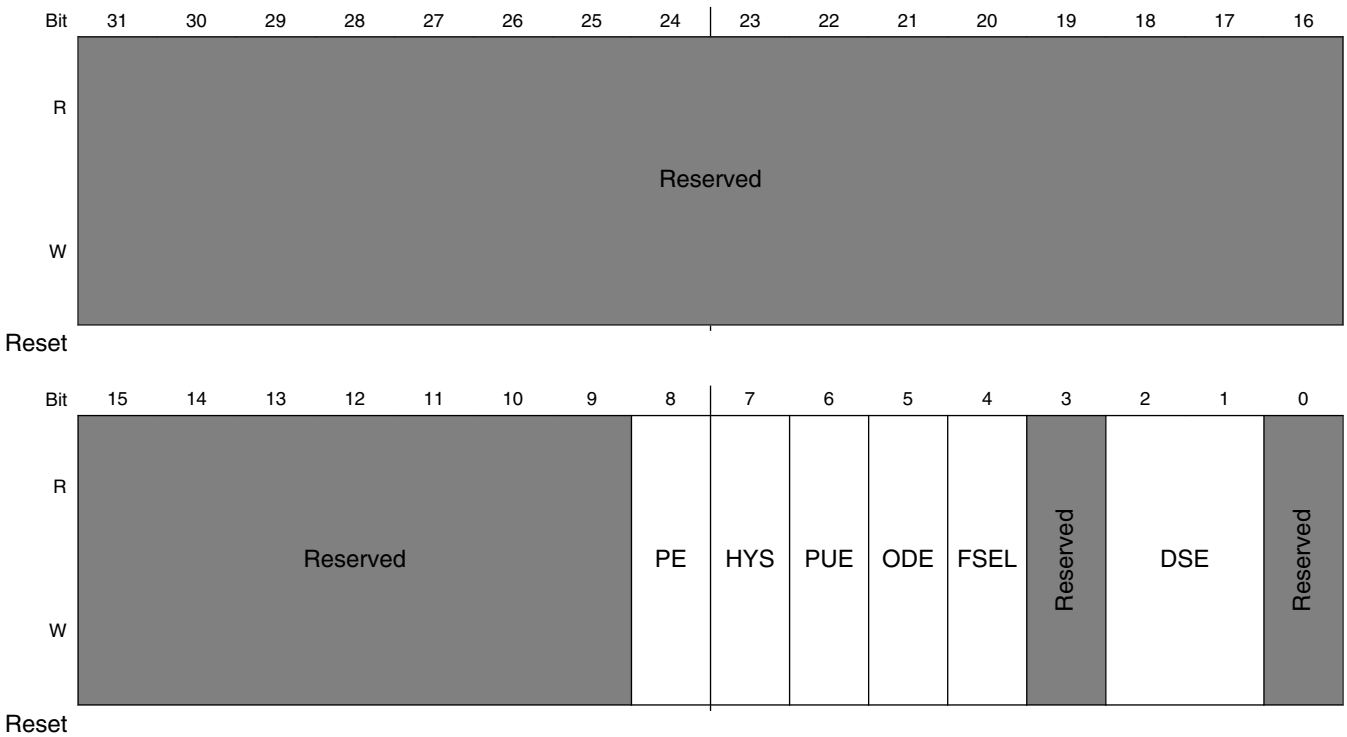
IOMUXC_SW_PAD_CTL_PAD_SAI1_TXC field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_TXC 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_TXC 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXC 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.242 SW_PAD_CTL_PAD_SAI1_TXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 3D8h offset = 3033_03D8h



IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_TXD0 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_TXD0 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_TXD0

Table continues on the next page...

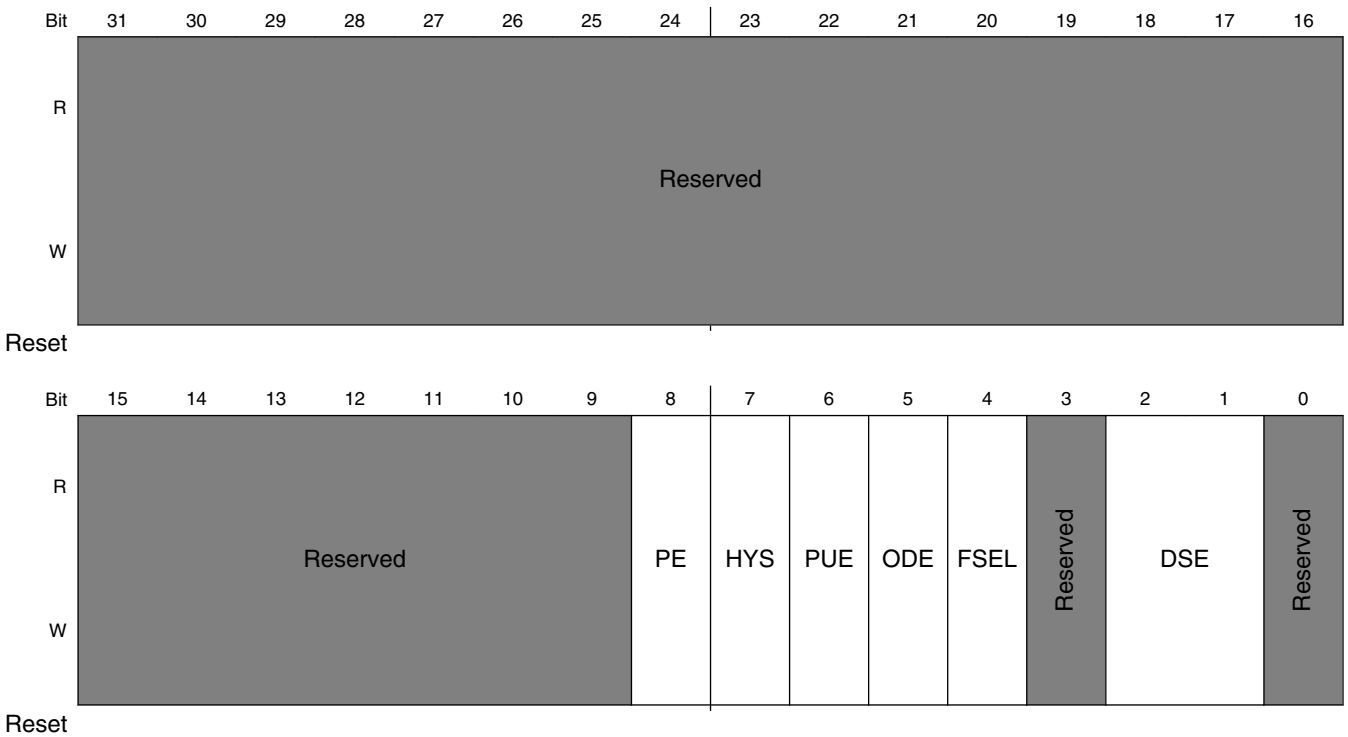
IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD0 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_TXD0 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_TXD0 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD0 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.243 SW_PAD_CTL_PAD_SAI1_TXD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD1)

SW_PAD_CTL Register

Address: 3033_0000h base + 3DCh offset = 3033_03DCh



IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_TXD1 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_TXD1 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_TXD1

Table continues on the next page...

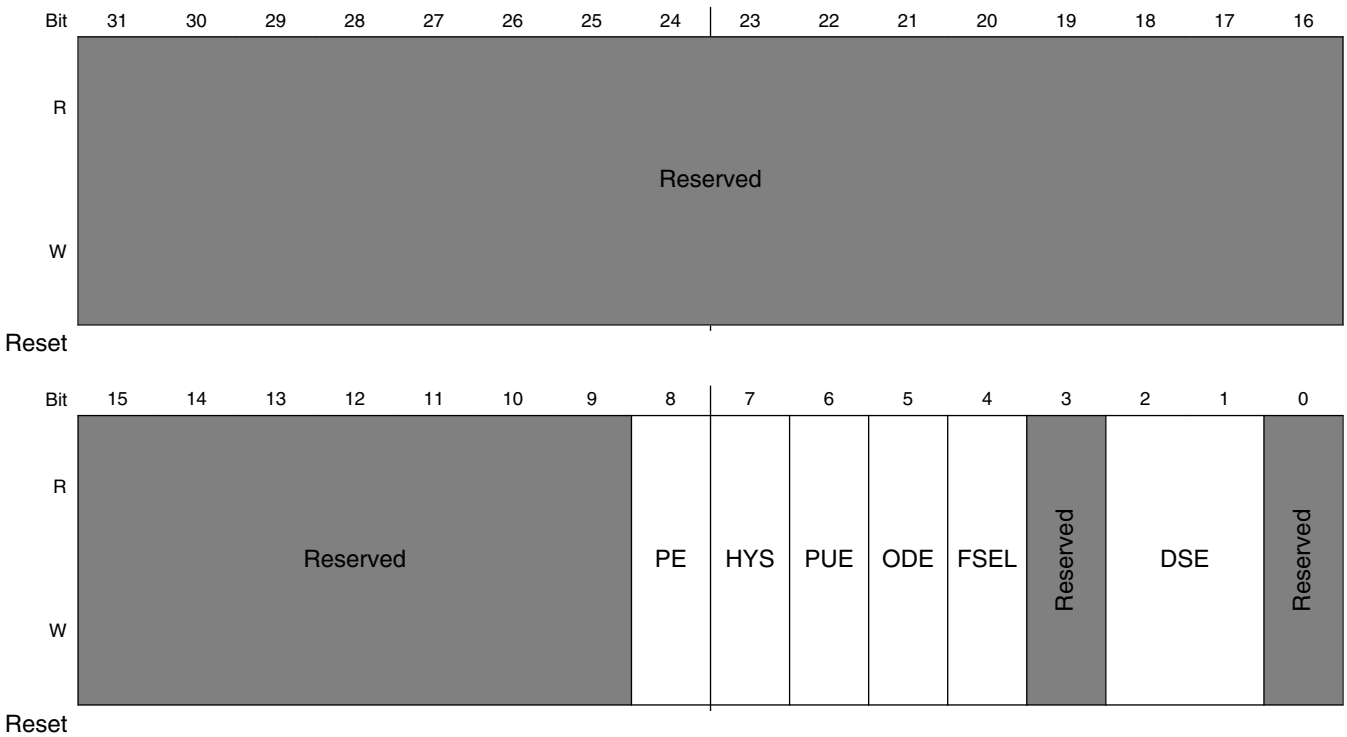
IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD1 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_TXD1 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_TXD1 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD1 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.244 SW_PAD_CTL_PAD_SAI1_TXD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD2)

SW_PAD_CTL Register

Address: 3033_0000h base + 3E0h offset = 3033_03E0h



IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_TXD2 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_TXD2 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_TXD2

Table continues on the next page...

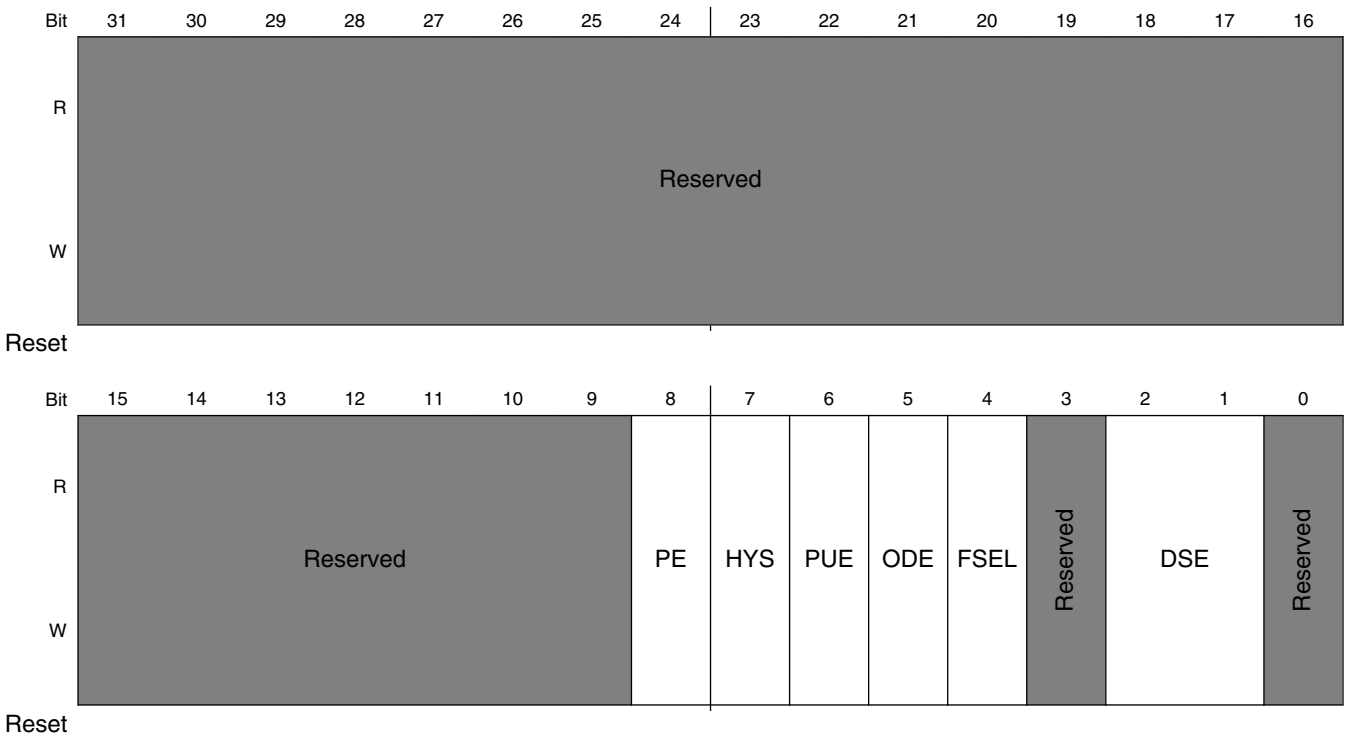
IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD2 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_TXD2 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_TXD2 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD2 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.245 SW_PAD_CTL_PAD_SAI1_TXD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD3)

SW_PAD_CTL Register

Address: 3033_0000h base + 3E4h offset = 3033_03E4h



IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_TXD3 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_TXD3 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_TXD3

Table continues on the next page...

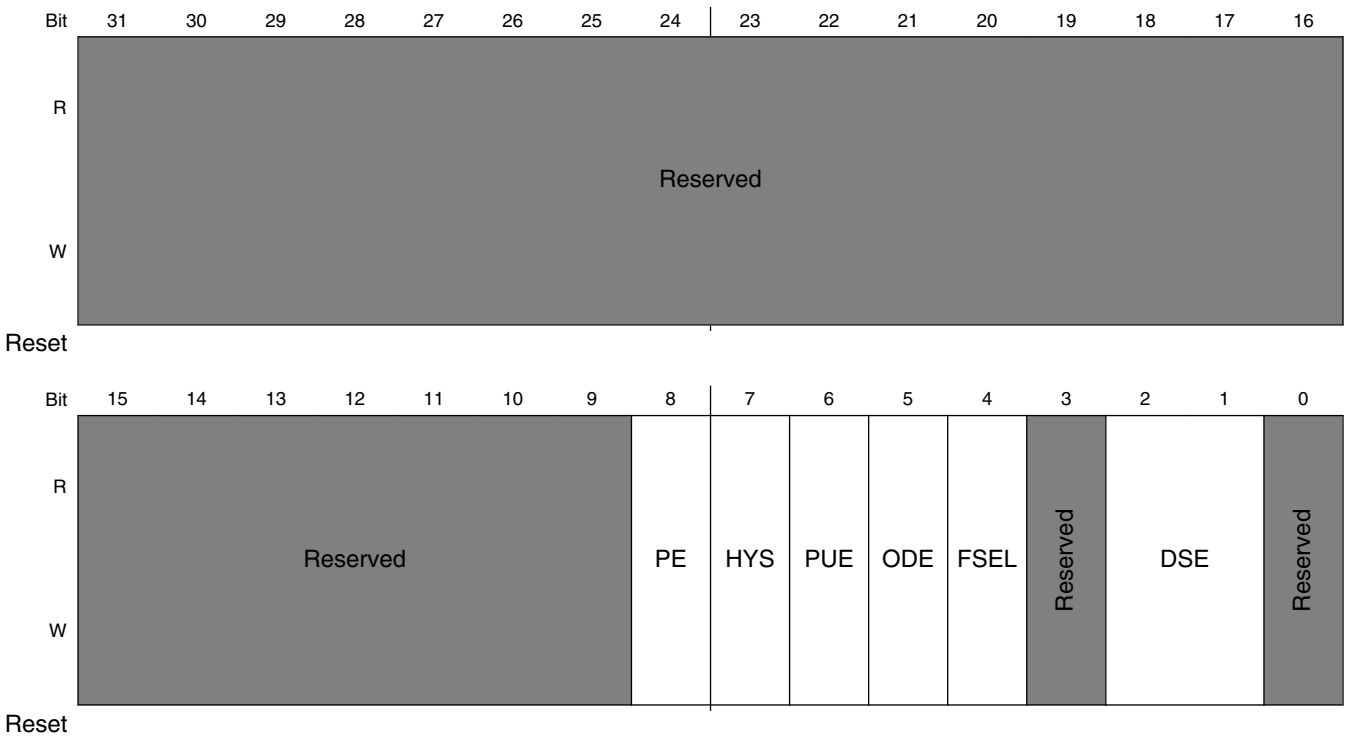
IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD3 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_TXD3 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_TXD3 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD3 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.246 SW_PAD_CTL_PAD_SAI1_TXD4 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD4)

SW_PAD_CTL Register

Address: 3033_0000h base + 3E8h offset = 3033_03E8h



IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD4 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_TXD4 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_TXD4 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_TXD4

Table continues on the next page...

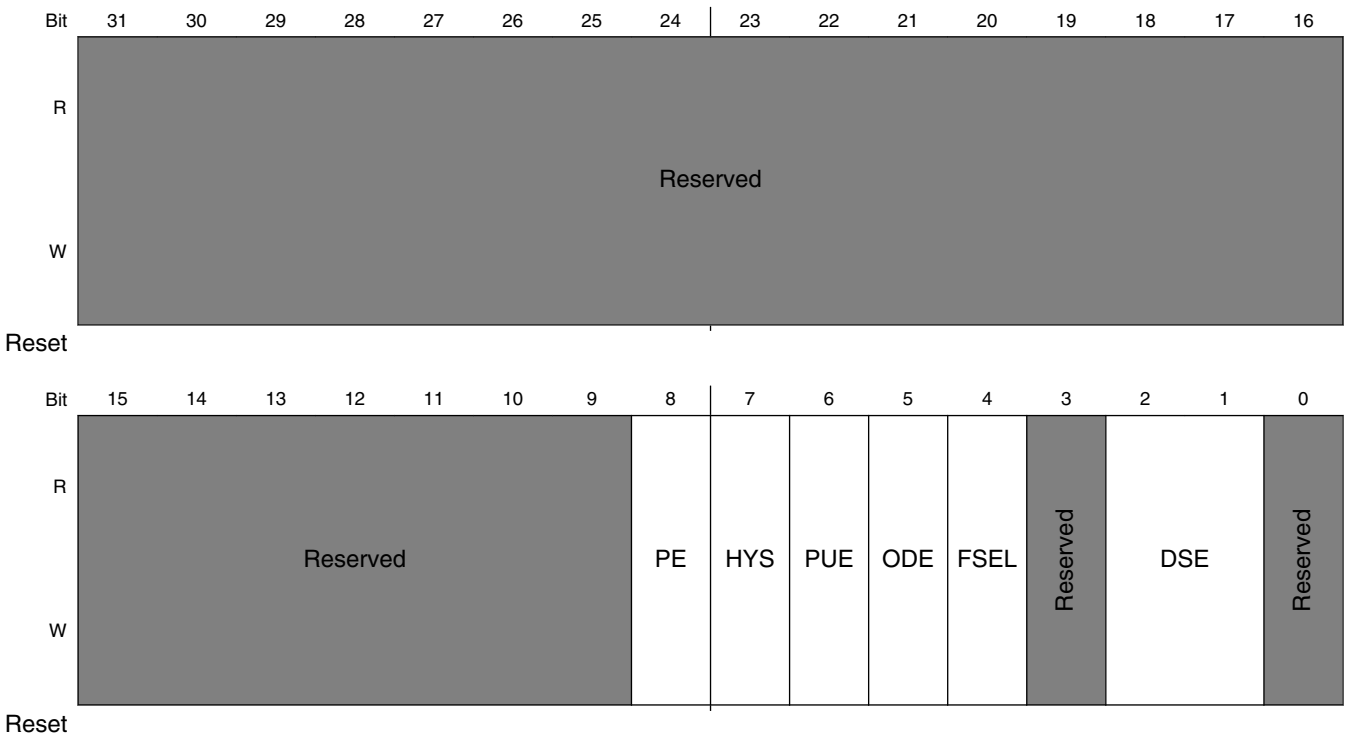
IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD4 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_TXD4 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_TXD4 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD4 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.247 SW_PAD_CTL_PAD_SAI1_TXD5 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD5)

SW_PAD_CTL Register

Address: 3033_0000h base + 3ECh offset = 3033_03ECh



IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD5 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_TXD5 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_TXD5 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_TXD5

Table continues on the next page...

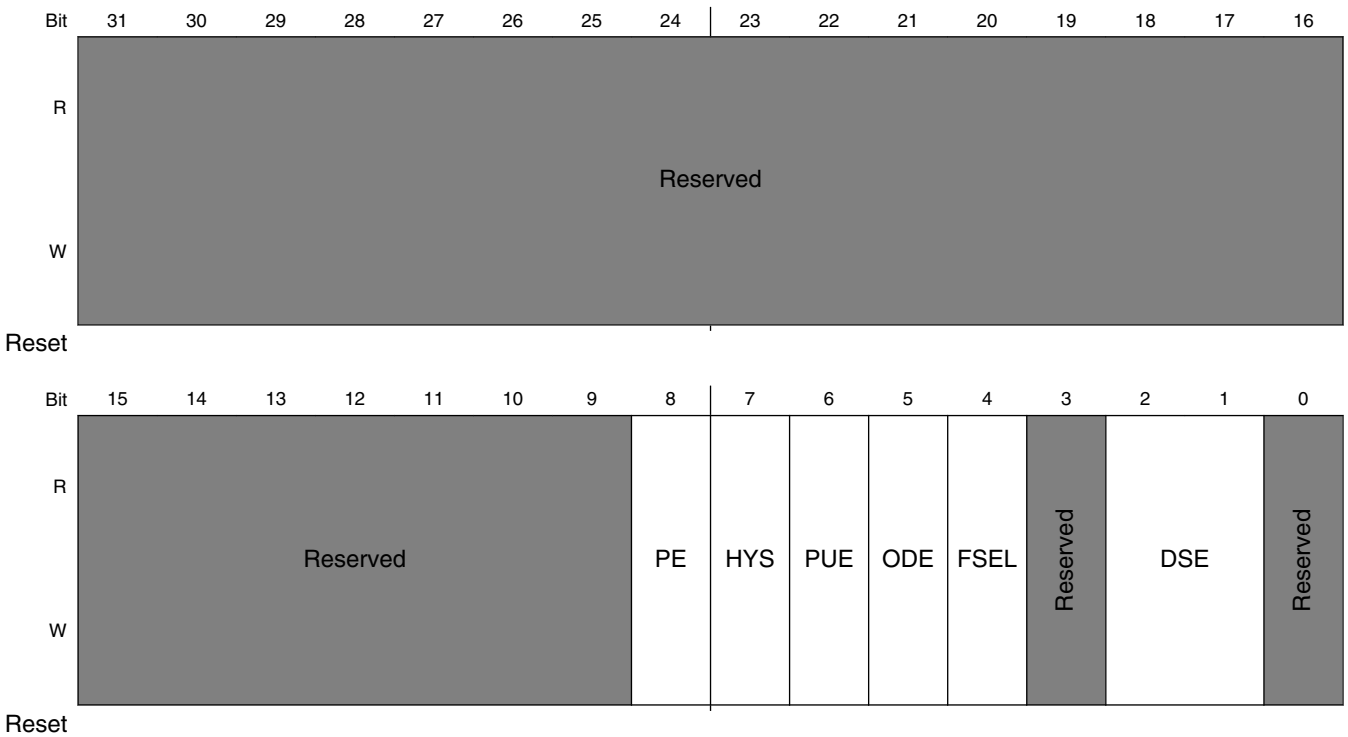
IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD5 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_TXD5 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_TXD5 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD5 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.248 SW_PAD_CTL_PAD_SAI1_TXD6 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD6)

SW_PAD_CTL Register

Address: 3033_0000h base + 3F0h offset = 3033_03F0h



IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD6 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_TXD6 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_TXD6 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_TXD6

Table continues on the next page...

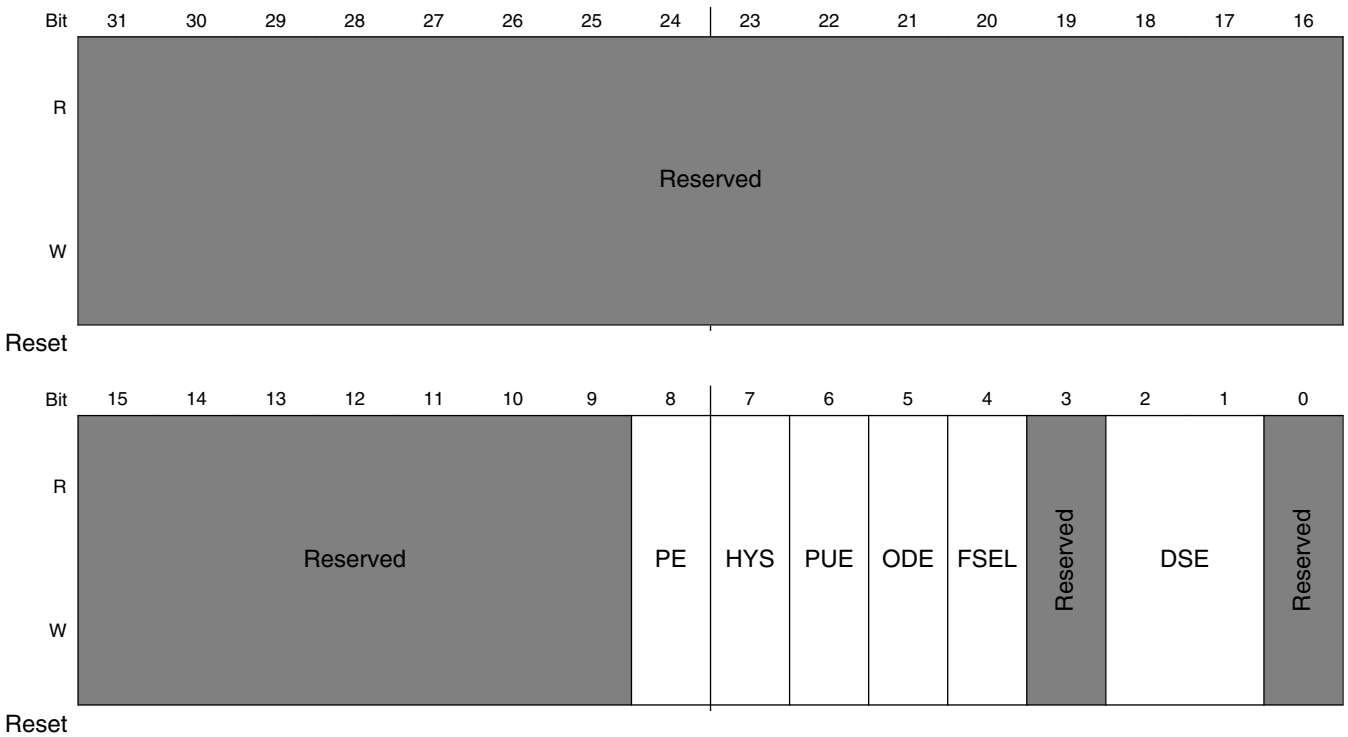
IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD6 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_TXD6 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_TXD6 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD6 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.249 SW_PAD_CTL_PAD_SAI1_TXD7 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD7)

SW_PAD_CTL Register

Address: 3033_0000h base + 3F4h offset = 3033_03F4h



IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD7 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_TXD7 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_TXD7 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_TXD7

Table continues on the next page...

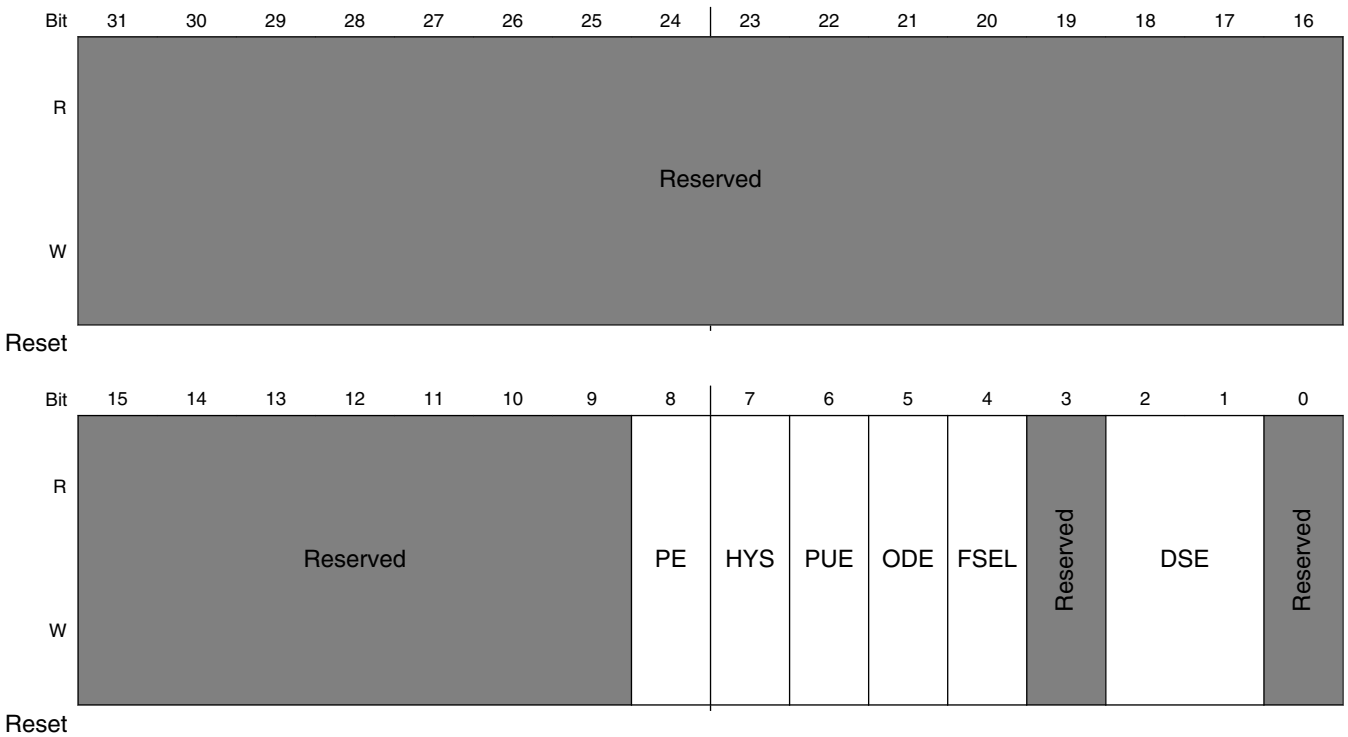
IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD7 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_TXD7 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_TXD7 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD7 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.250 SW_PAD_CTL_PAD_SAI1_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_MCLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 3F8h offset = 3033_03F8h



IOMUXC_SW_PAD_CTL_PAD_SAI1_MCLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI1_MCLK 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI1_MCLK 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI1_MCLK

Table continues on the next page...

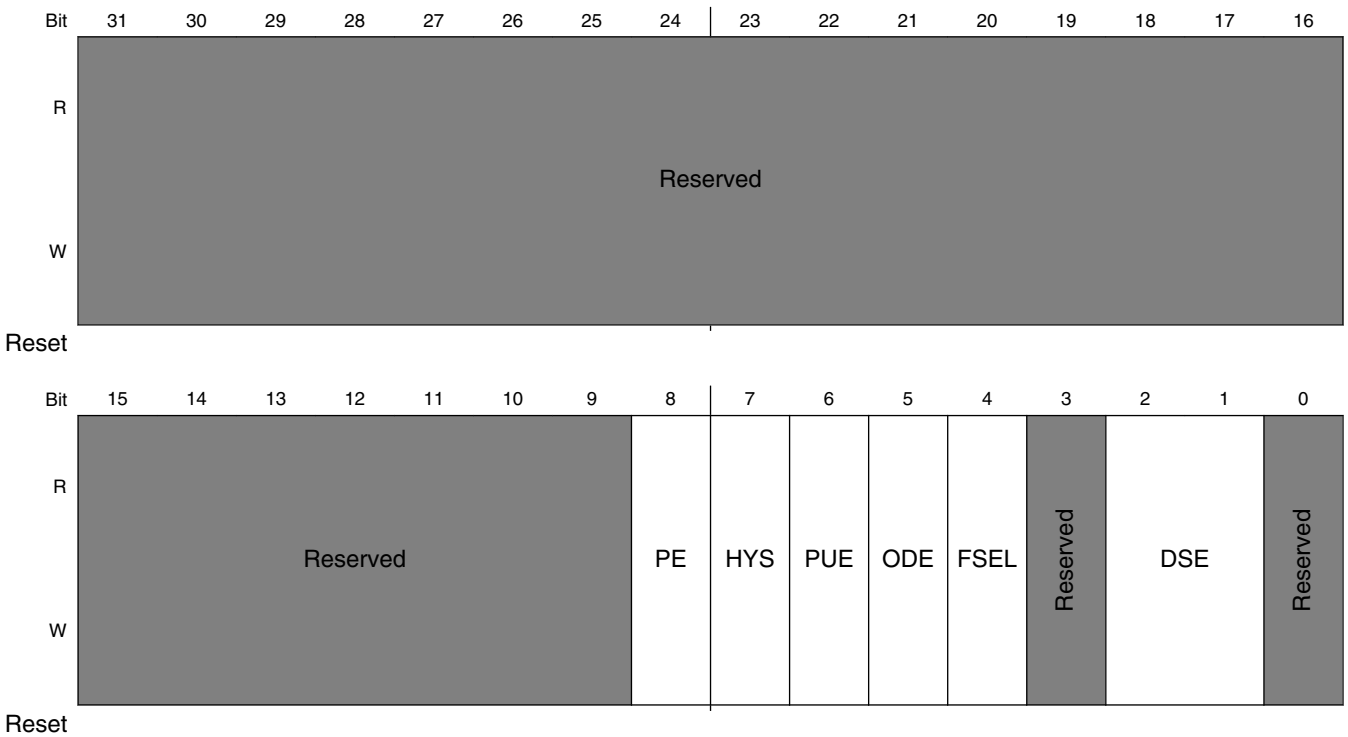
IOMUXC_SW_PAD_CTL_PAD_SAI1_MCLK field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI1_MCLK 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI1_MCLK 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI1_MCLK 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.251 SW_PAD_CTL_PAD_SAI2_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_RXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 3FCh offset = 3033_03FCh



IOMUXC_SW_PAD_CTL_PAD_SAI2_RXFS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI2_RXFS 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI2_RXFS 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI2_RXFS

Table continues on the next page...

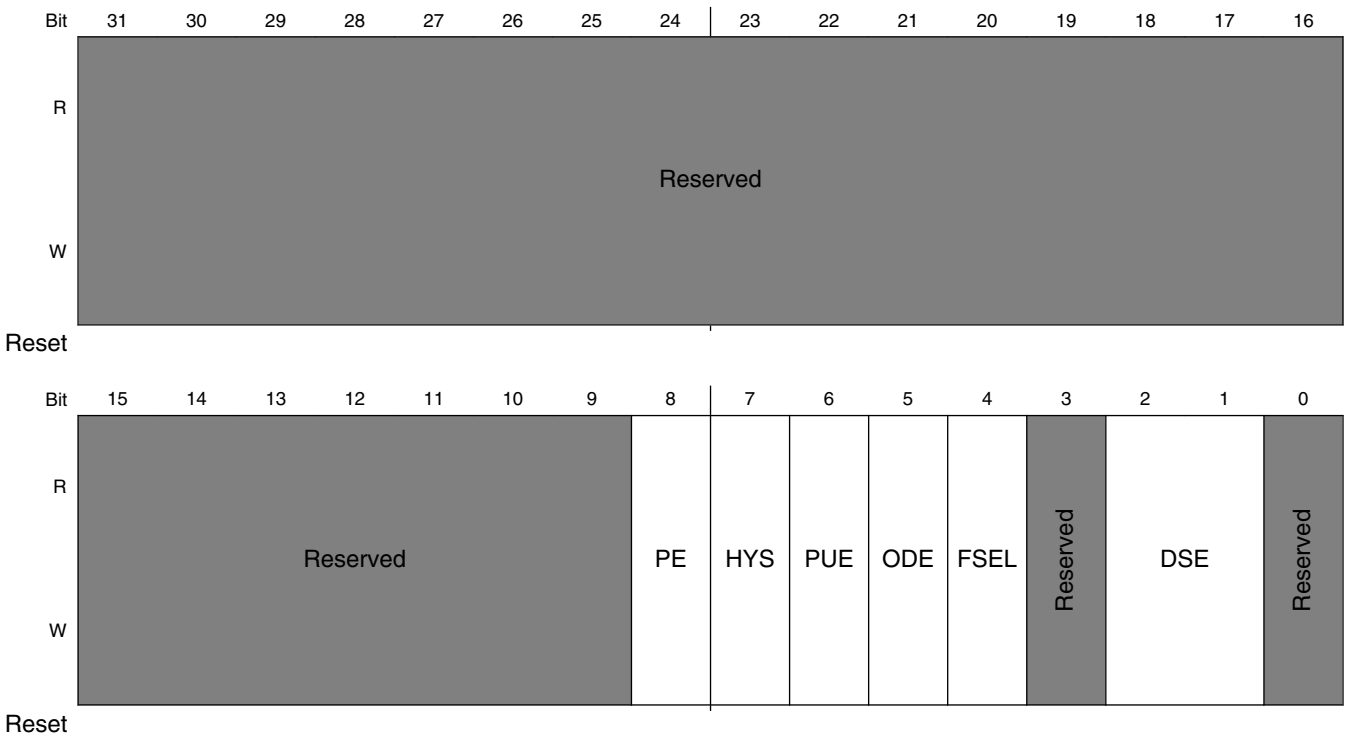
IOMUXC_SW_PAD_CTL_PAD_SAI2_RXFS field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI2_RXFS 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI2_RXFS 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI2_RXFS 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.252 SW_PAD_CTL_PAD_SAI2_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_RXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 400h offset = 3033_0400h



IOMUXC_SW_PAD_CTL_PAD_SAI2_RXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI2_RXC 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI2_RXC 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI2_RXC

Table continues on the next page...

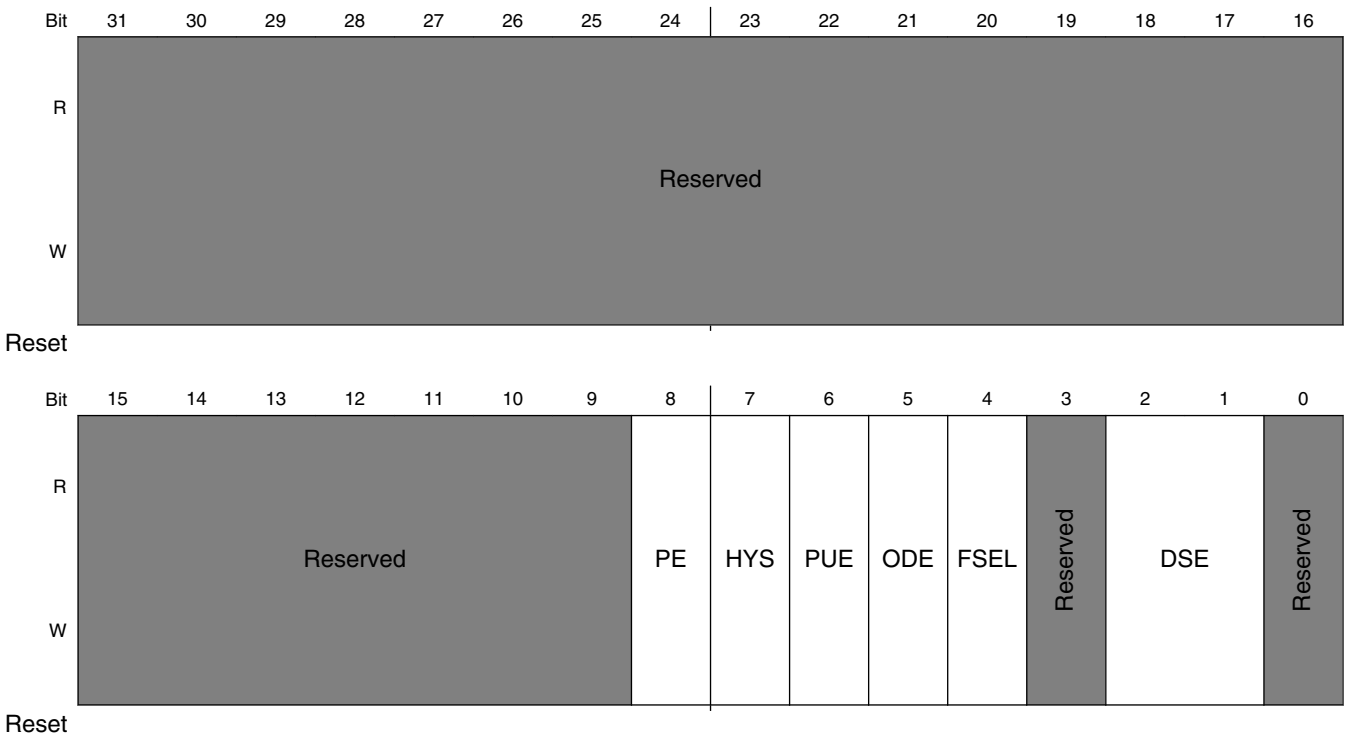
IOMUXC_SW_PAD_CTL_PAD_SAI2_RXC field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI2_RXC 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI2_RXC 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI2_RXC 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.253 SW_PAD_CTL_PAD_SAI2_RXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_RXD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 404h offset = 3033_0404h



IOMUXC_SW_PAD_CTL_PAD_SAI2_RXD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI2_RXD0 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI2_RXD0 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI2_RXD0

Table continues on the next page...

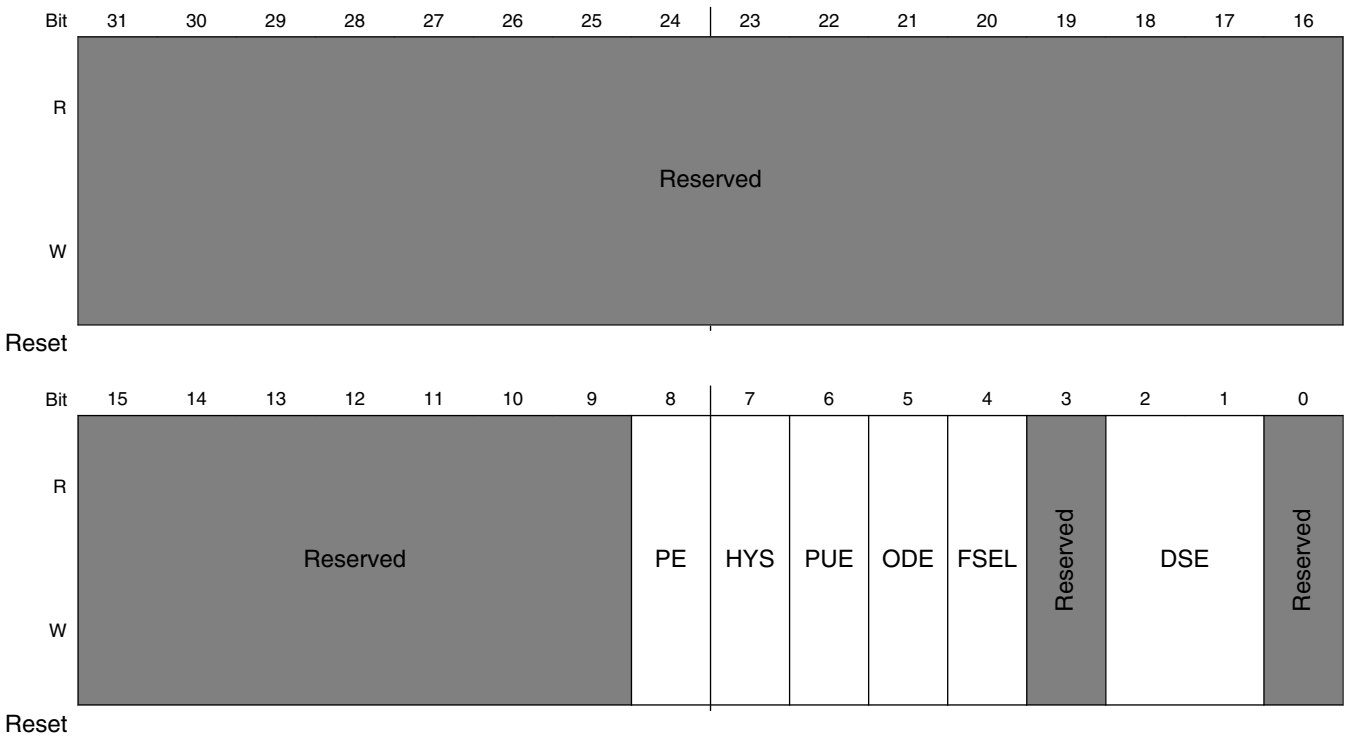
IOMUXC_SW_PAD_CTL_PAD_SAI2_RXD0 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI2_RXD0 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI2_RXD0 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI2_RXD0 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.254 SW_PAD_CTL_PAD_SAI2_TXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_TXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 408h offset = 3033_0408h



IOMUXC_SW_PAD_CTL_PAD_SAI2_TXFS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI2_TXFS 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI2_TXFS 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI2_TXFS

Table continues on the next page...

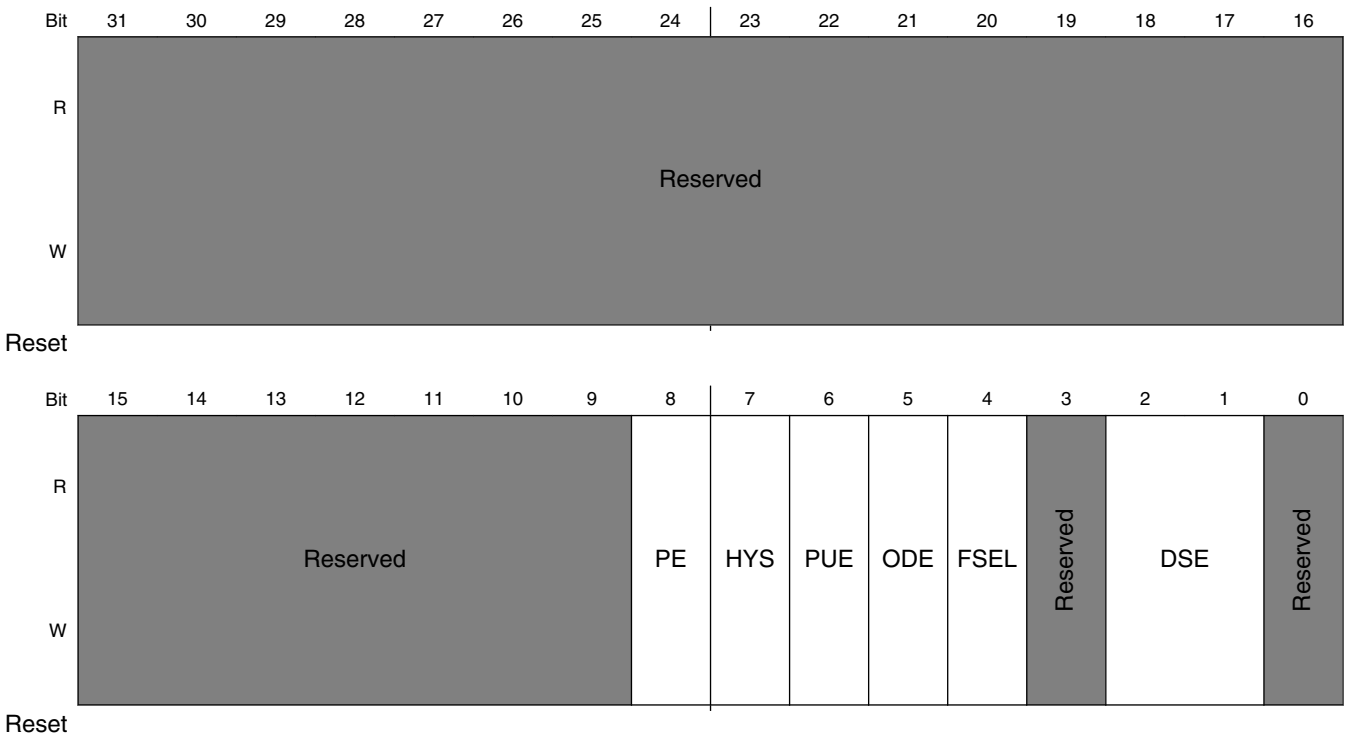
IOMUXC_SW_PAD_CTL_PAD_SAI2_TXFS field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI2_TXFS 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI2_TXFS 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI2_TXFS 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.255 SW_PAD_CTL_PAD_SAI2_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_TXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 40Ch offset = 3033_040Ch



IOMUXC_SW_PAD_CTL_PAD_SAI2_TXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI2_TXC 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI2_TXC 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI2_TXC

Table continues on the next page...

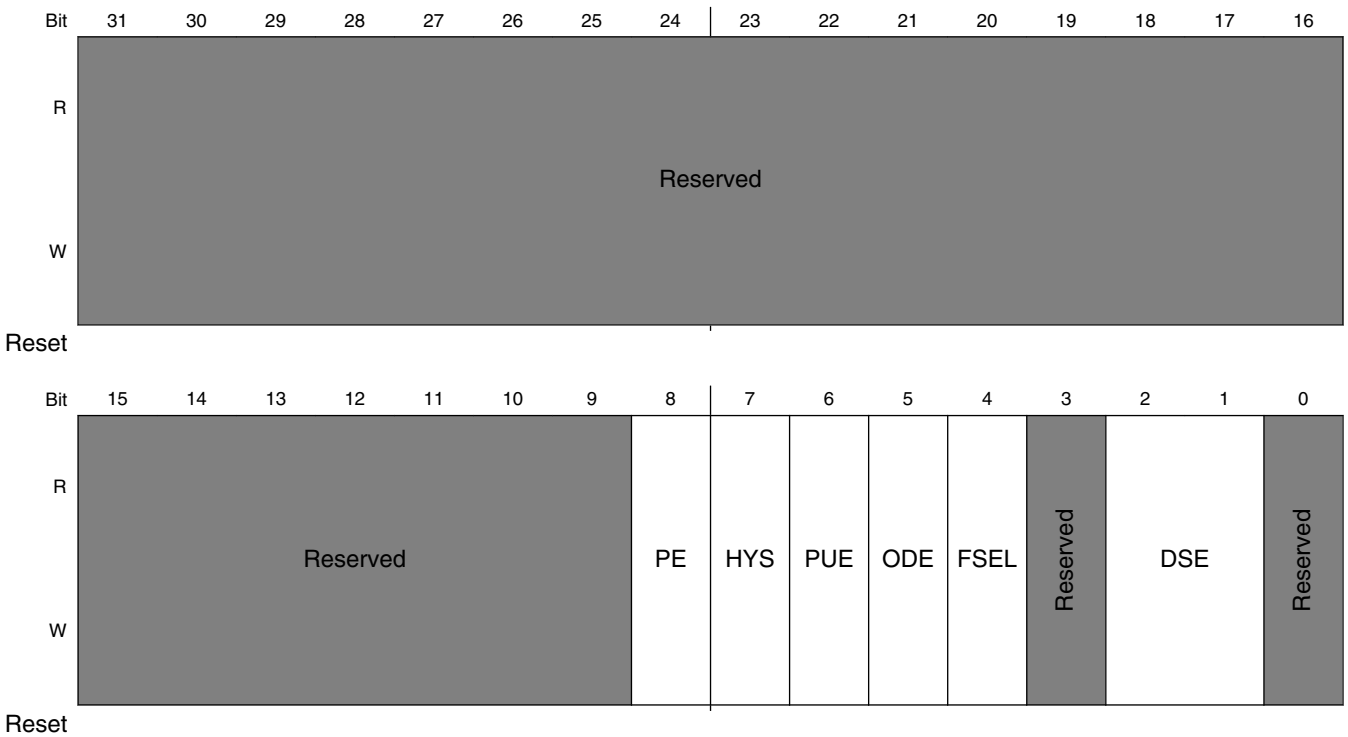
IOMUXC_SW_PAD_CTL_PAD_SAI2_TXC field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI2_TXC 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI2_TXC 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI2_TXC 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.256 SW_PAD_CTL_PAD_SAI2_TXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_TXD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 410h offset = 3033_0410h



IOMUXC_SW_PAD_CTL_PAD_SAI2_TXD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI2_TXD0 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI2_TXD0 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI2_TXD0

Table continues on the next page...

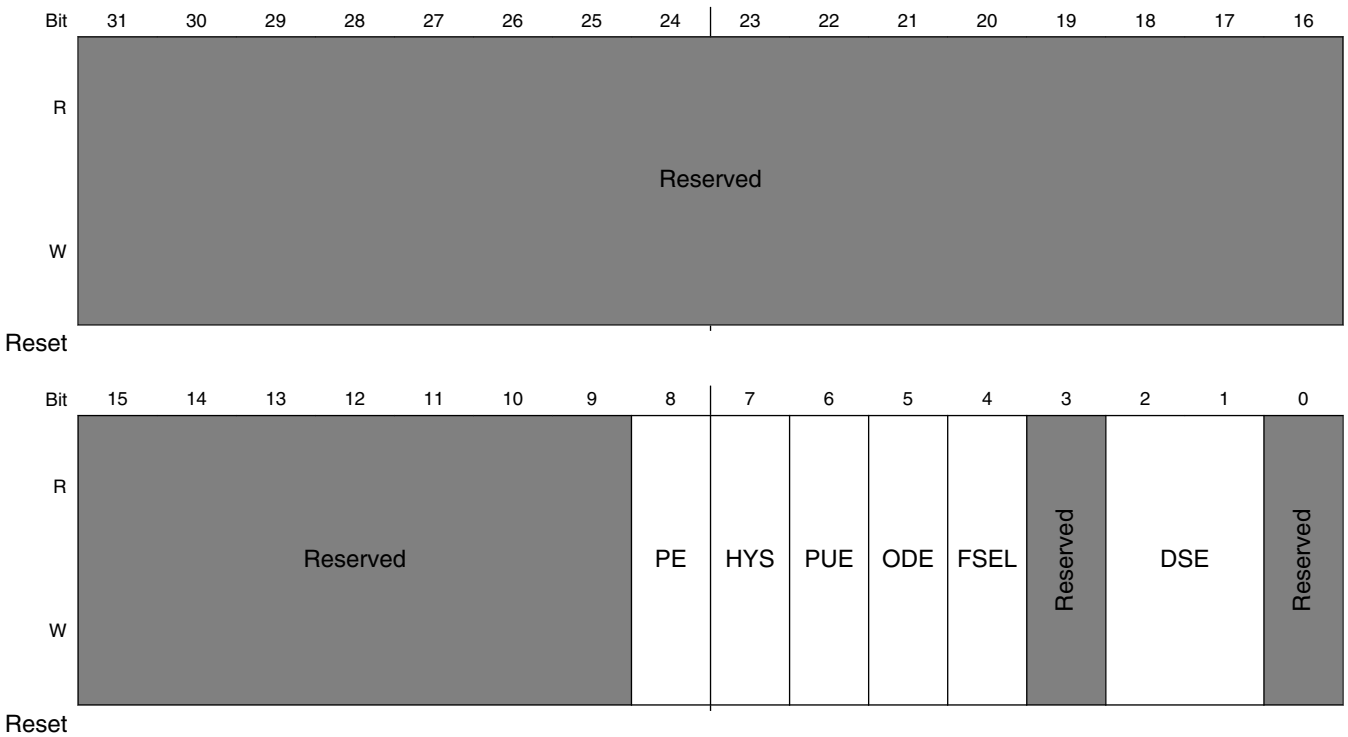
IOMUXC_SW_PAD_CTL_PAD_SAI2_TXD0 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI2_TXD0 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI2_TXD0 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI2_TXD0 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.257 SW_PAD_CTL_PAD_SAI2_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_MCLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 414h offset = 3033_0414h



IOMUXC_SW_PAD_CTL_PAD_SAI2_MCLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI2_MCLK 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI2_MCLK 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI2_MCLK

Table continues on the next page...

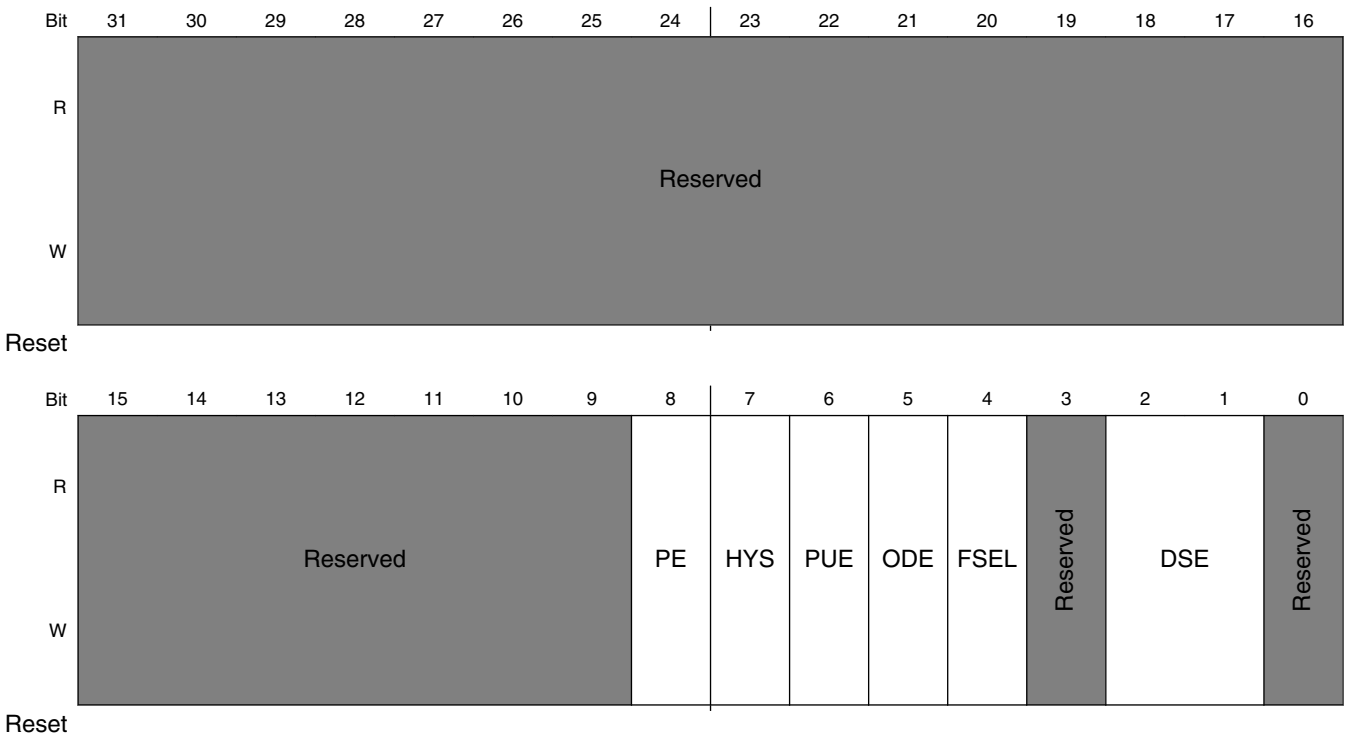
IOMUXC_SW_PAD_CTL_PAD_SAI2_MCLK field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI2_MCLK 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI2_MCLK 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI2_MCLK 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.258 SW_PAD_CTL_PAD_SAI3_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_RXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 418h offset = 3033_0418h



IOMUXC_SW_PAD_CTL_PAD_SAI3_RXFS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI3_RXFS 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI3_RXFS 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI3_RXFS

Table continues on the next page...

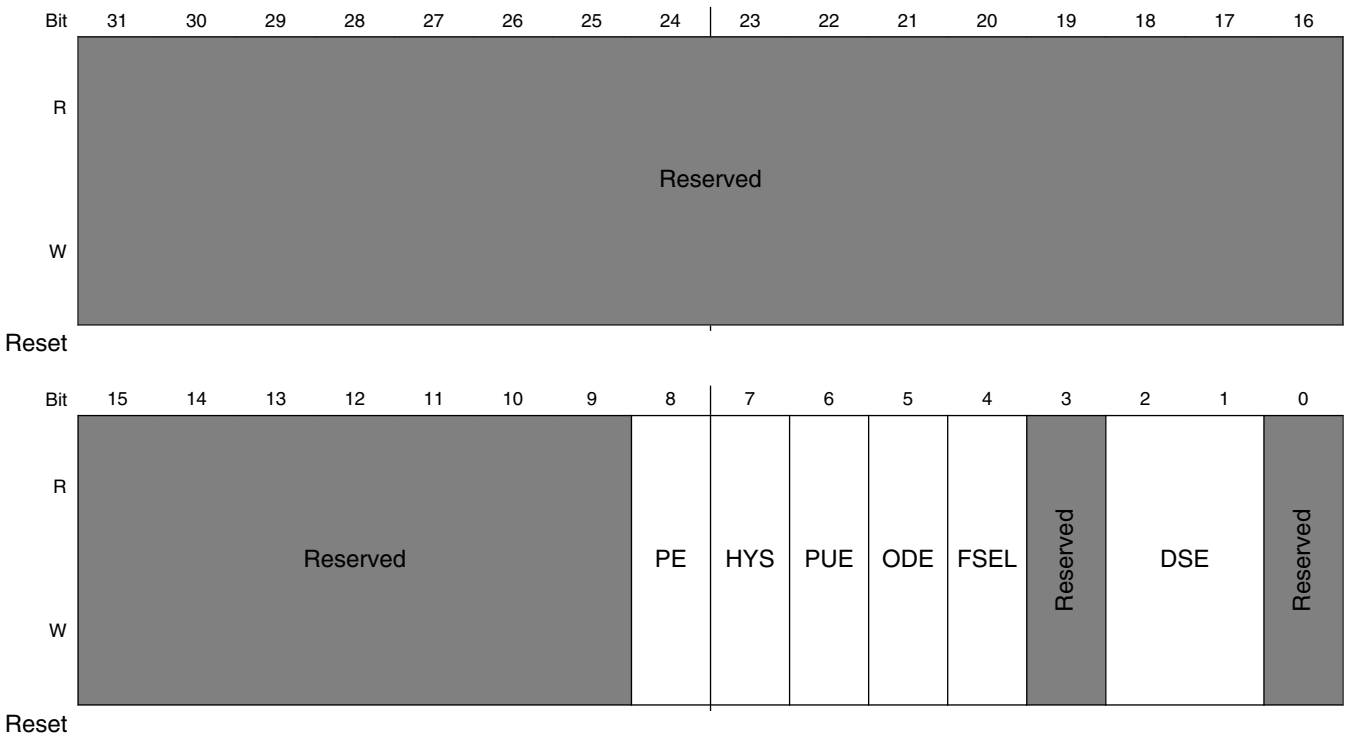
IOMUXC_SW_PAD_CTL_PAD_SAI3_RXFS field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI3_RXFS 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI3_RXFS 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI3_RXFS 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.259 SW_PAD_CTL_PAD_SAI3_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_RXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 41Ch offset = 3033_041Ch



IOMUXC_SW_PAD_CTL_PAD_SAI3_RXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI3_RXC 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI3_RXC 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI3_RXC

Table continues on the next page...

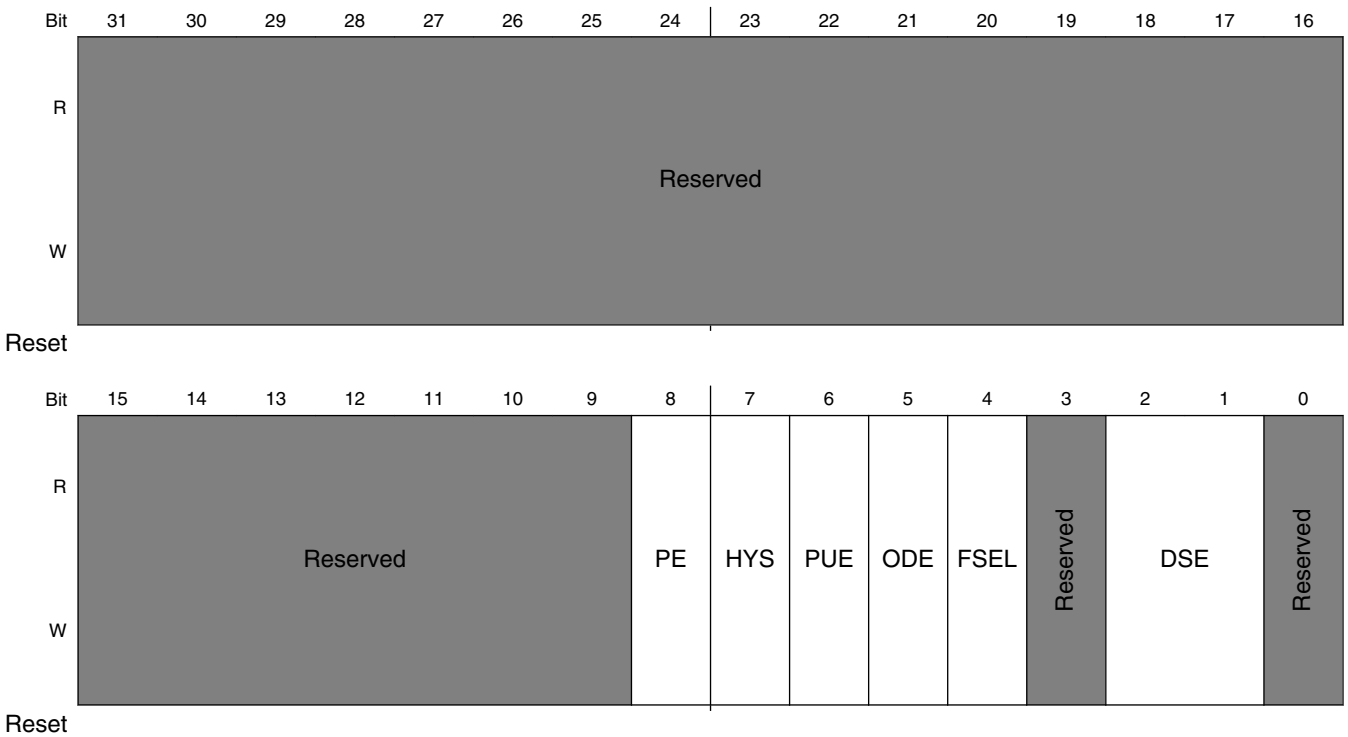
IOMUXC_SW_PAD_CTL_PAD_SAI3_RXC field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI3_RXC 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI3_RXC 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI3_RXC 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.260 SW_PAD_CTL_PAD_SAI3_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_RXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 420h offset = 3033_0420h



IOMUXC_SW_PAD_CTL_PAD_SAI3_RXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI3_RXD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI3_RXD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI3_RXD

Table continues on the next page...

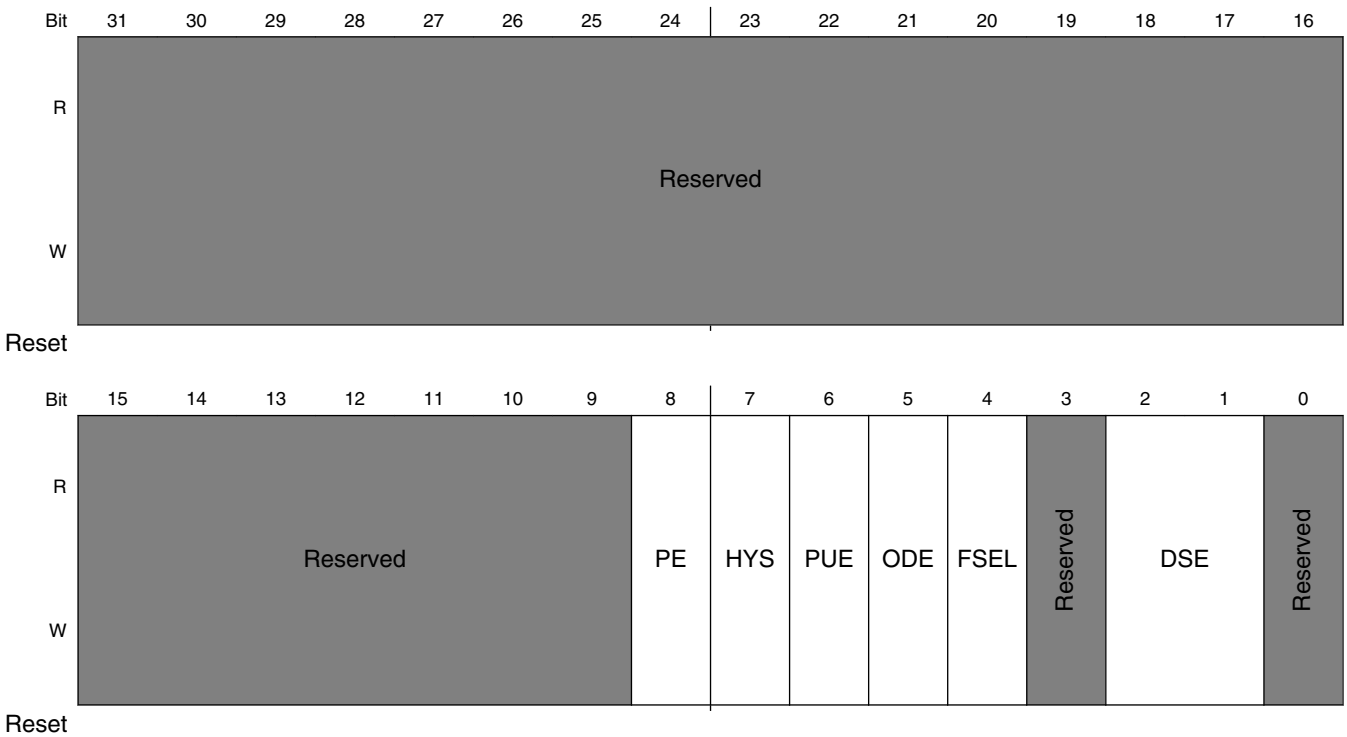
IOMUXC_SW_PAD_CTL_PAD_SAI3_RXD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI3_RXD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI3_RXD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI3_RXD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.261 SW_PAD_CTL_PAD_SAI3_TXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_TXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 424h offset = 3033_0424h



IOMUXC_SW_PAD_CTL_PAD_SAI3_TXFS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI3_TXFS 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI3_TXFS 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI3_TXFS

Table continues on the next page...

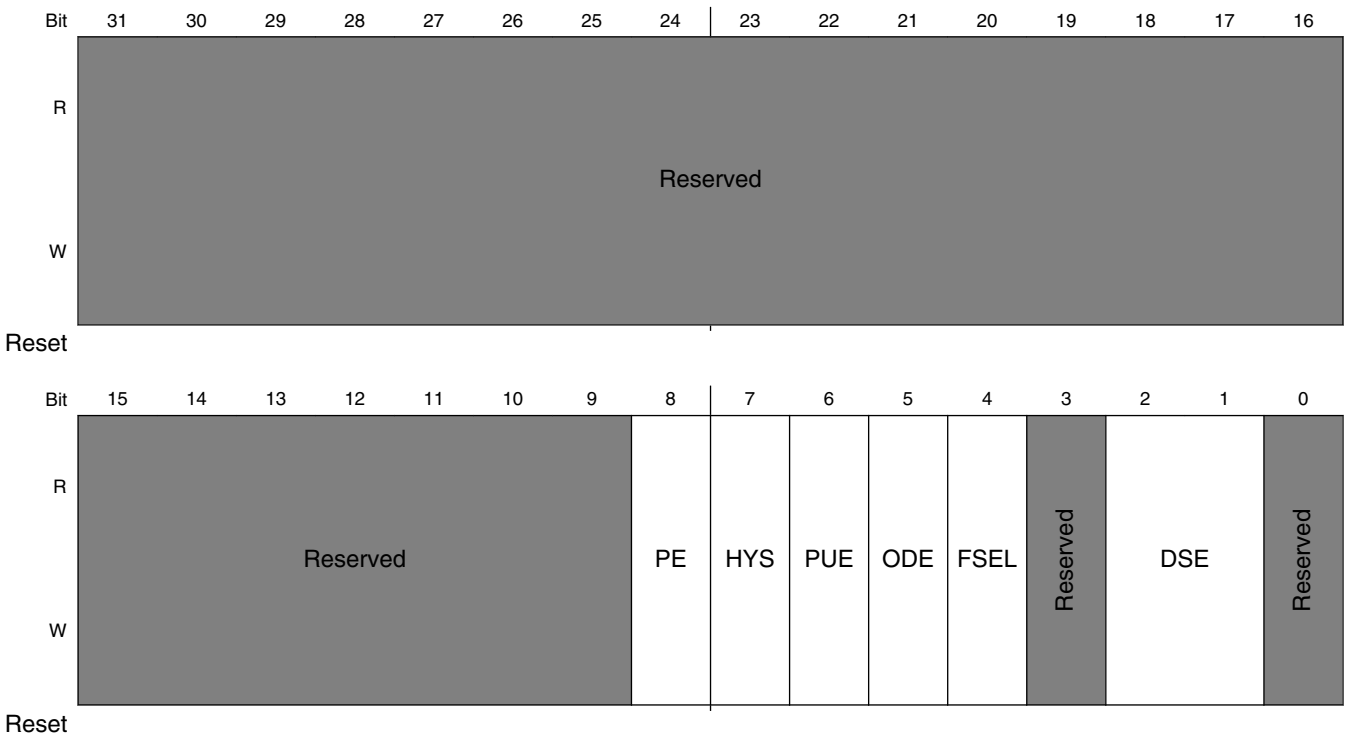
IOMUXC_SW_PAD_CTL_PAD_SAI3_TXFS field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI3_TXFS 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI3_TXFS 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI3_TXFS 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.262 SW_PAD_CTL_PAD_SAI3_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_TXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 428h offset = 3033_0428h



IOMUXC_SW_PAD_CTL_PAD_SAI3_TXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI3_TXC 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI3_TXC 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI3_TXC

Table continues on the next page...

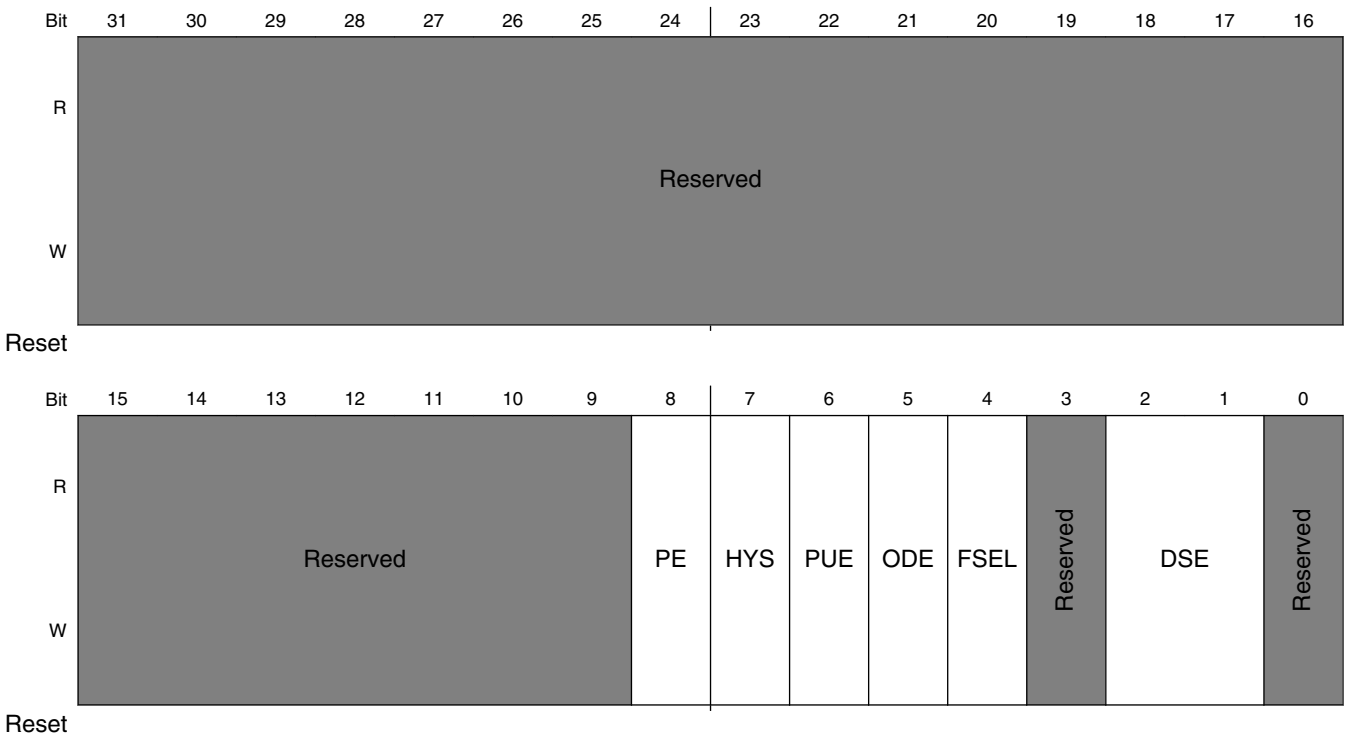
IOMUXC_SW_PAD_CTL_PAD_SAI3_TXC field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI3_TXC 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI3_TXC 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI3_TXC 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.263 SW_PAD_CTL_PAD_SAI3_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_TXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 42Ch offset = 3033_042Ch



IOMUXC_SW_PAD_CTL_PAD_SAI3_TXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI3_TXD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI3_TXD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI3_TXD

Table continues on the next page...

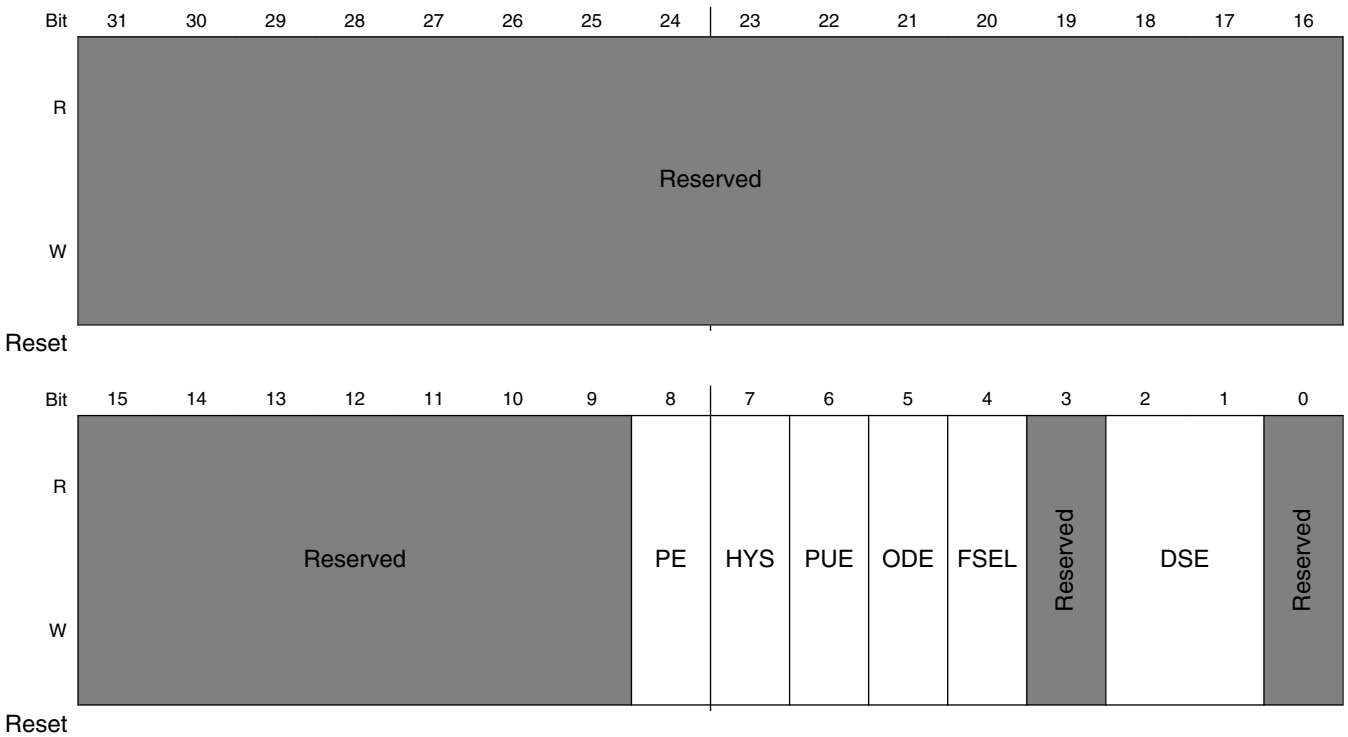
IOMUXC_SW_PAD_CTL_PAD_SAI3_TXD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI3_TXD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI3_TXD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI3_TXD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.264 SW_PAD_CTL_PAD_SAI3_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_MCLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 430h offset = 3033_0430h



IOMUXC_SW_PAD_CTL_PAD_SAI3_MCLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SAI3_MCLK 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SAI3_MCLK 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SAI3_MCLK

Table continues on the next page...

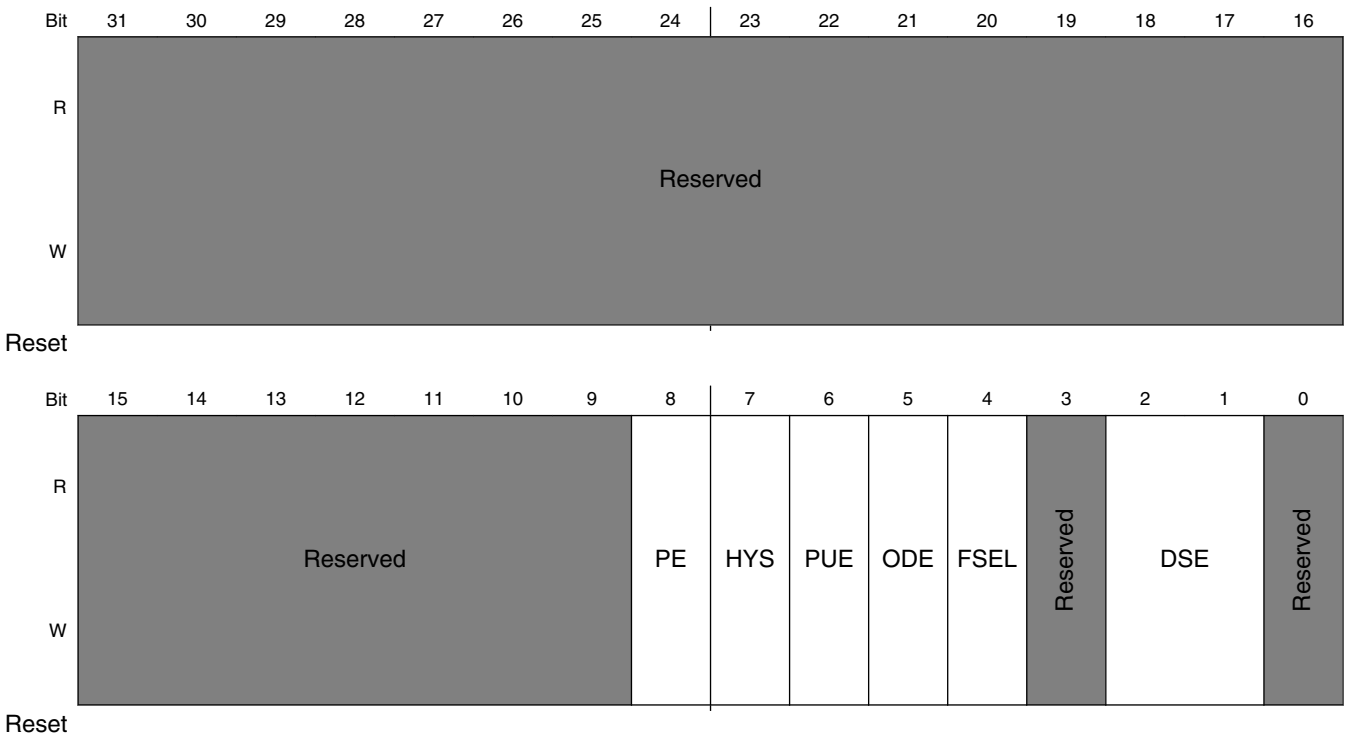
IOMUXC_SW_PAD_CTL_PAD_SAI3_MCLK field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SAI3_MCLK 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SAI3_MCLK 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SAI3_MCLK 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.265 SW_PAD_CTL_PAD_SPDIF_TX SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SPDIF_TX)

SW_PAD_CTL Register

Address: 3033_0000h base + 434h offset = 3033_0434h



IOMUXC_SW_PAD_CTL_PAD_SPDIF_TX field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SPDIF_TX 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SPDIF_TX 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SPDIF_TX

Table continues on the next page...

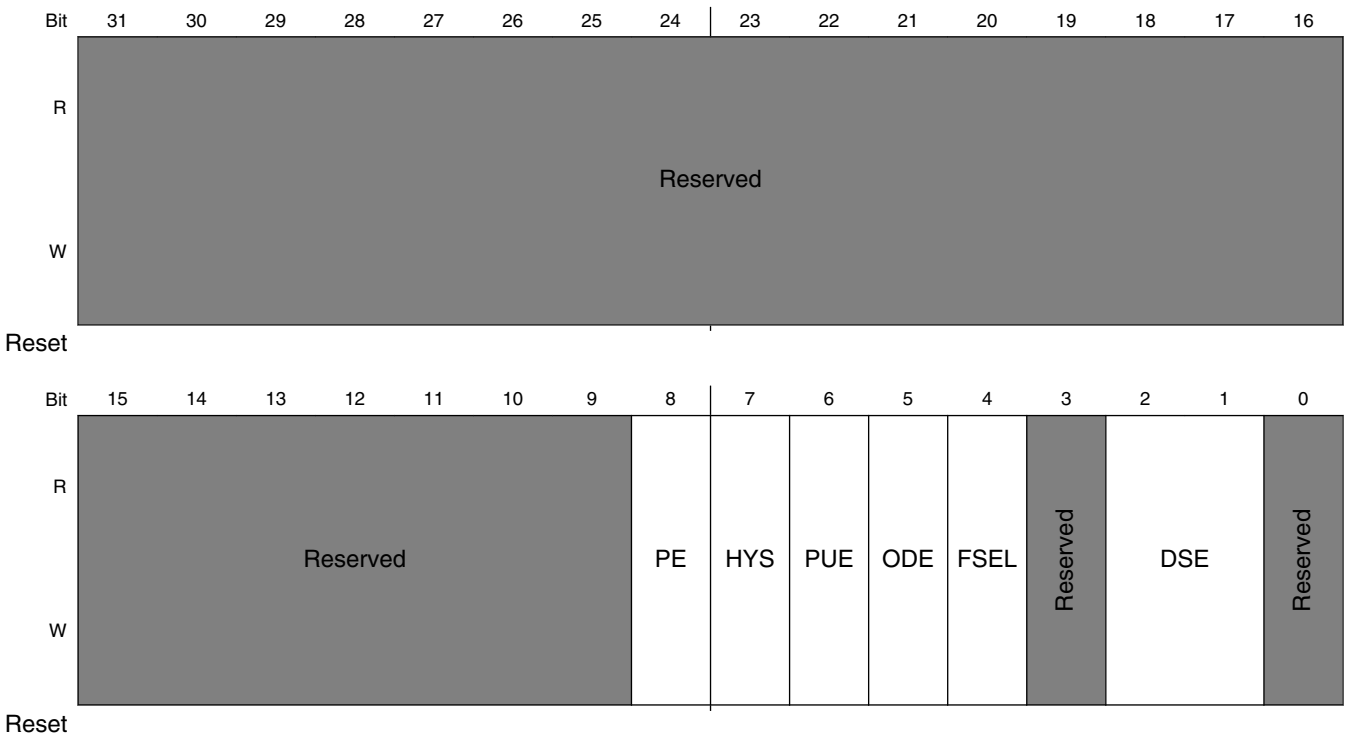
IOMUXC_SW_PAD_CTL_PAD_SPDIF_TX field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SPDIF_TX 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SPDIF_TX 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SPDIF_TX 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.266 SW_PAD_CTL_PAD_SPDIF_RX SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SPDIF_RX)

SW_PAD_CTL Register

Address: 3033_0000h base + 438h offset = 3033_0438h



IOMUXC_SW_PAD_CTL_PAD_SPDIF_RX field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SPDIF_RX 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SPDIF_RX 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SPDIF_RX

Table continues on the next page...

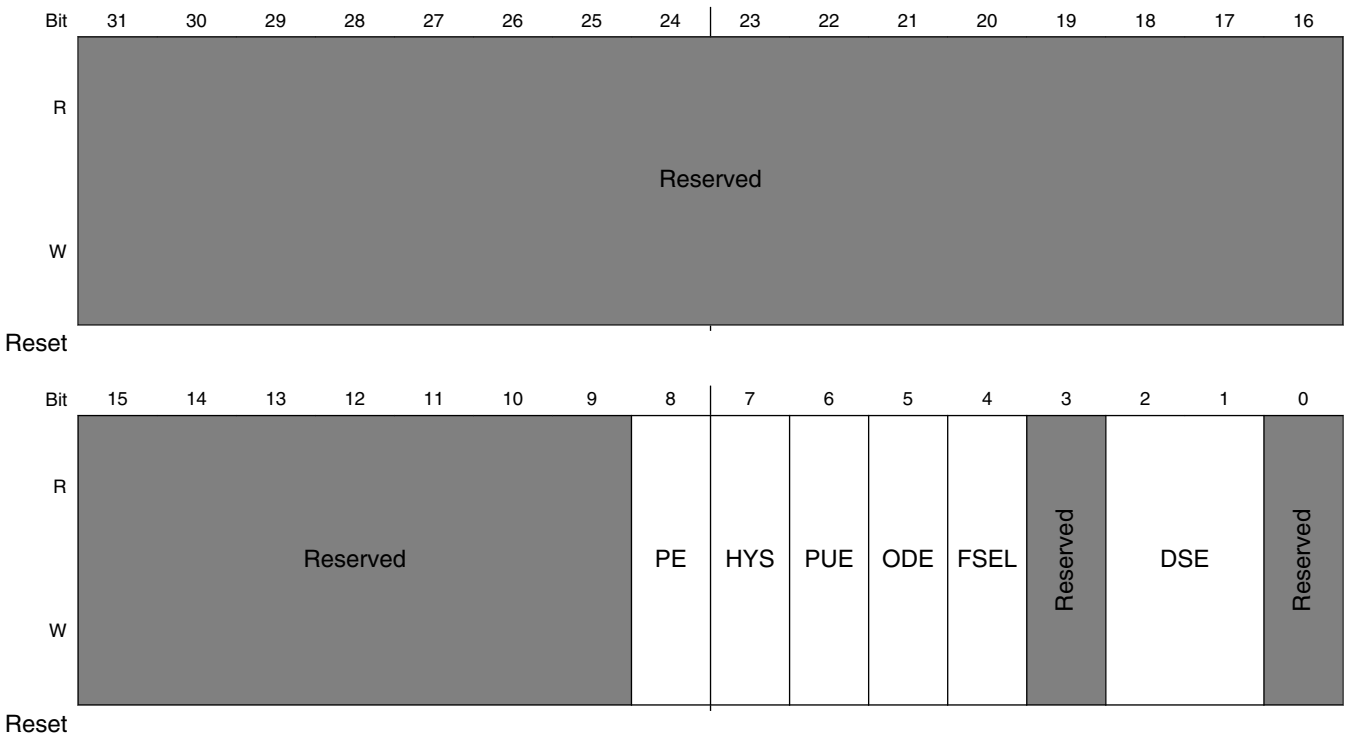
IOMUXC_SW_PAD_CTL_PAD_SPDIF_RX field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SPDIF_RX 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SPDIF_RX 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SPDIF_RX 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.267 SW_PAD_CTL_PAD_SPDIF_EXT_CLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SPDIF_EXT_CLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 43Ch offset = 3033_043Ch



IOMUXC_SW_PAD_CTL_PAD_SPDIF_EXT_CLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: SPDIF_EXT_CLK 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: SPDIF_EXT_CLK 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: SPDIF_EXT_CLK

Table continues on the next page...

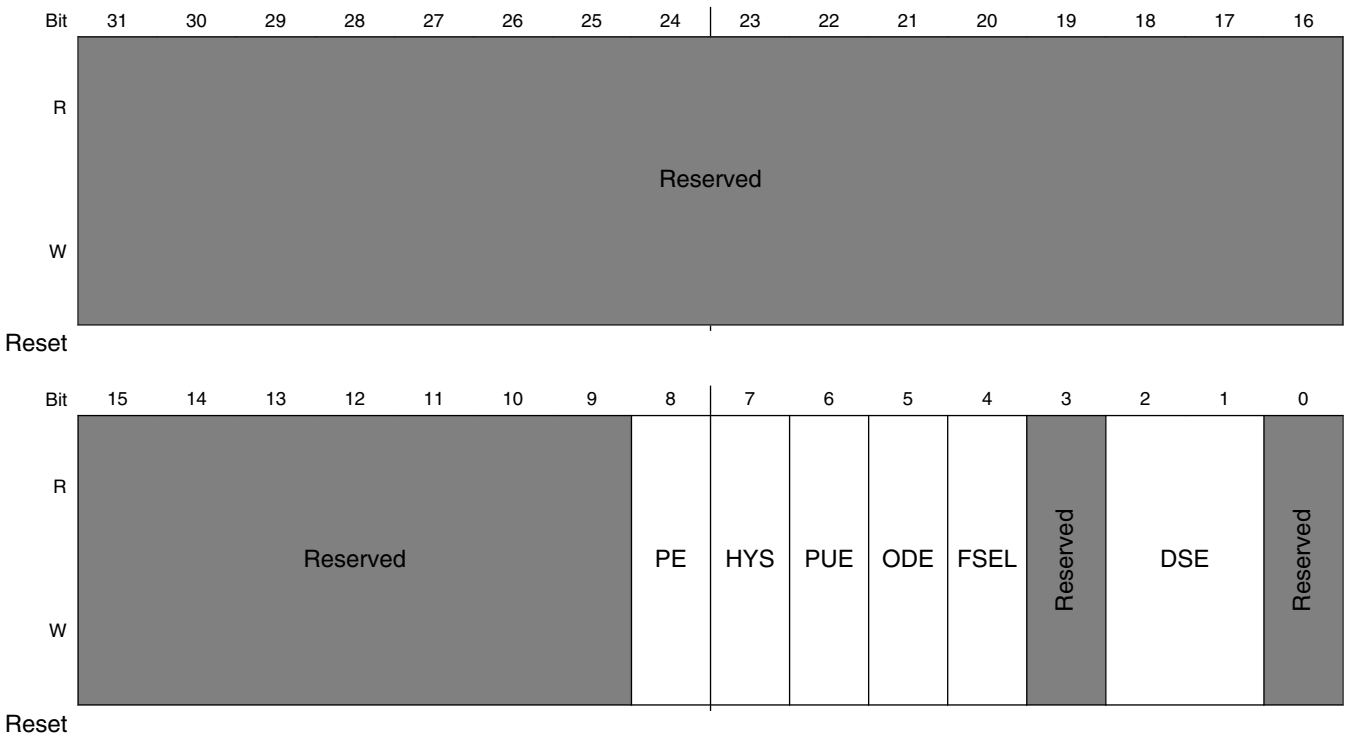
IOMUXC_SW_PAD_CTL_PAD_SPDIF_EXT_CLK field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: SPDIF_EXT_CLK 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: SPDIF_EXT_CLK 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: SPDIF_EXT_CLK 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.268 SW_PAD_CTL_PAD_ECSPI1_SCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SCLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 440h offset = 3033_0440h



IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SCLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ECSPI1_SCLK 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ECSPI1_SCLK 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ECSPI1_SCLK

Table continues on the next page...

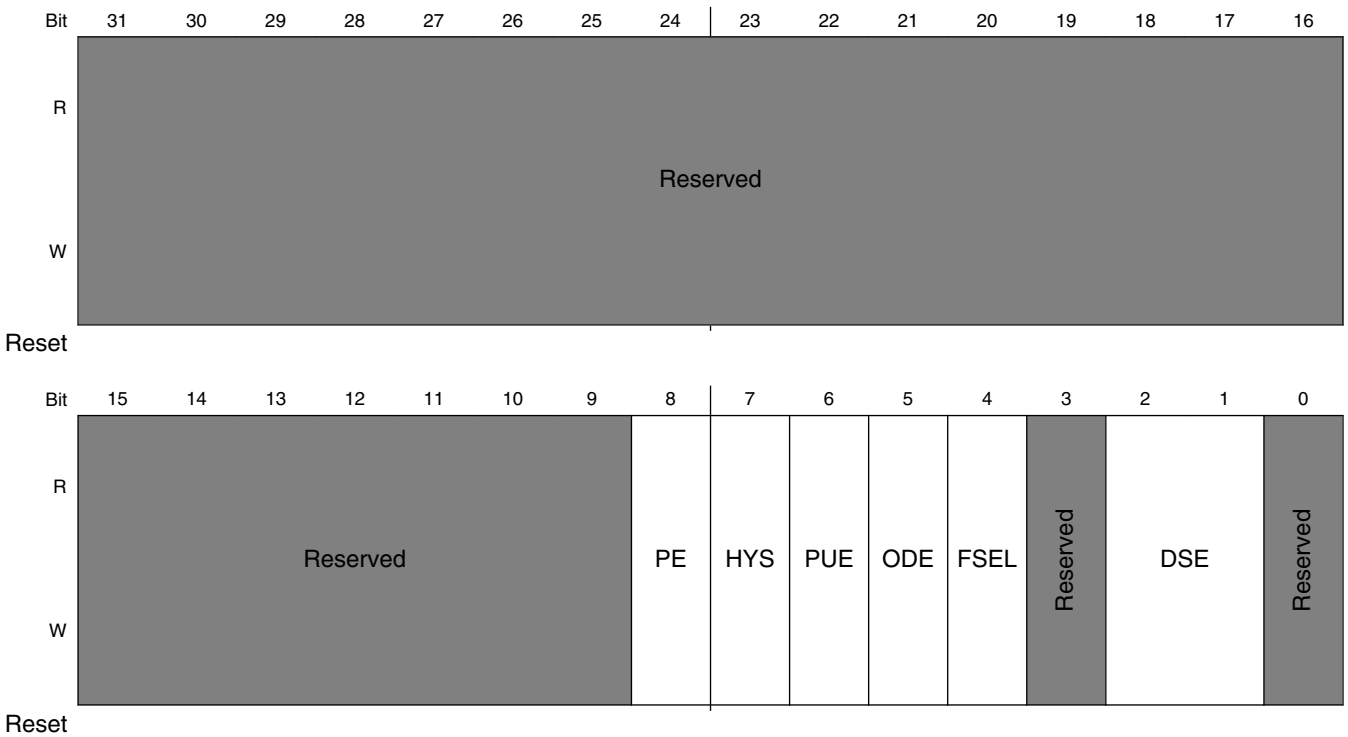
IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SCLK field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ECSPI1_SCLK 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ECSPI1_SCLK 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ECSPI1_SCLK 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.269 SW_PAD_CTL_PAD_ECSPI1_MOSI SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MOSI)

SW_PAD_CTL Register

Address: 3033_0000h base + 444h offset = 3033_0444h



IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MOSI field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ECSPI1_MOSI 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ECSPI1_MOSI 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ECSPI1_MOSI

Table continues on the next page...

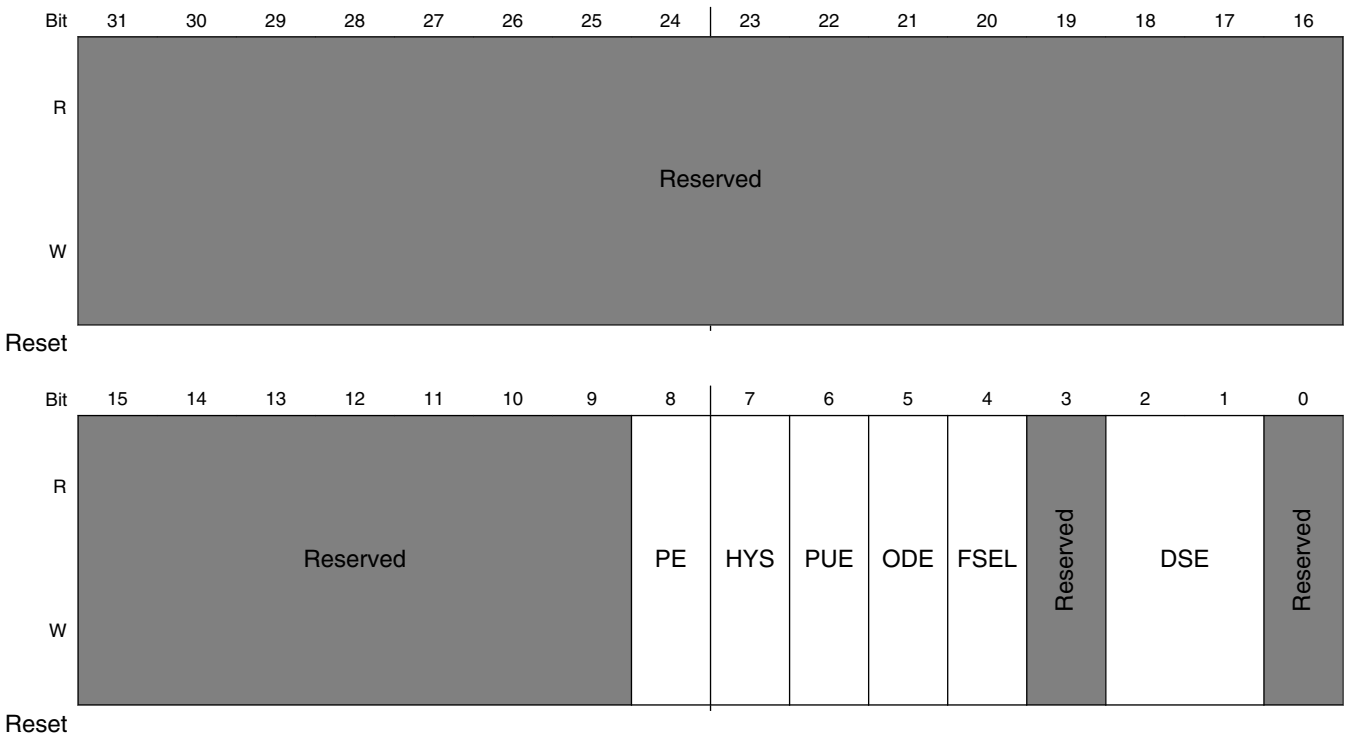
IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MOSI field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ECSPI1_MOSI 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ECSPI1_MOSI 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ECSPI1_MOSI 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.270 SW_PAD_CTL_PAD_ECSPI1_MISO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MISO)

SW_PAD_CTL Register

Address: 3033_0000h base + 448h offset = 3033_0448h



IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MISO field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ECSPI1_MISO 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ECSPI1_MISO 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ECSPI1_MISO

Table continues on the next page...

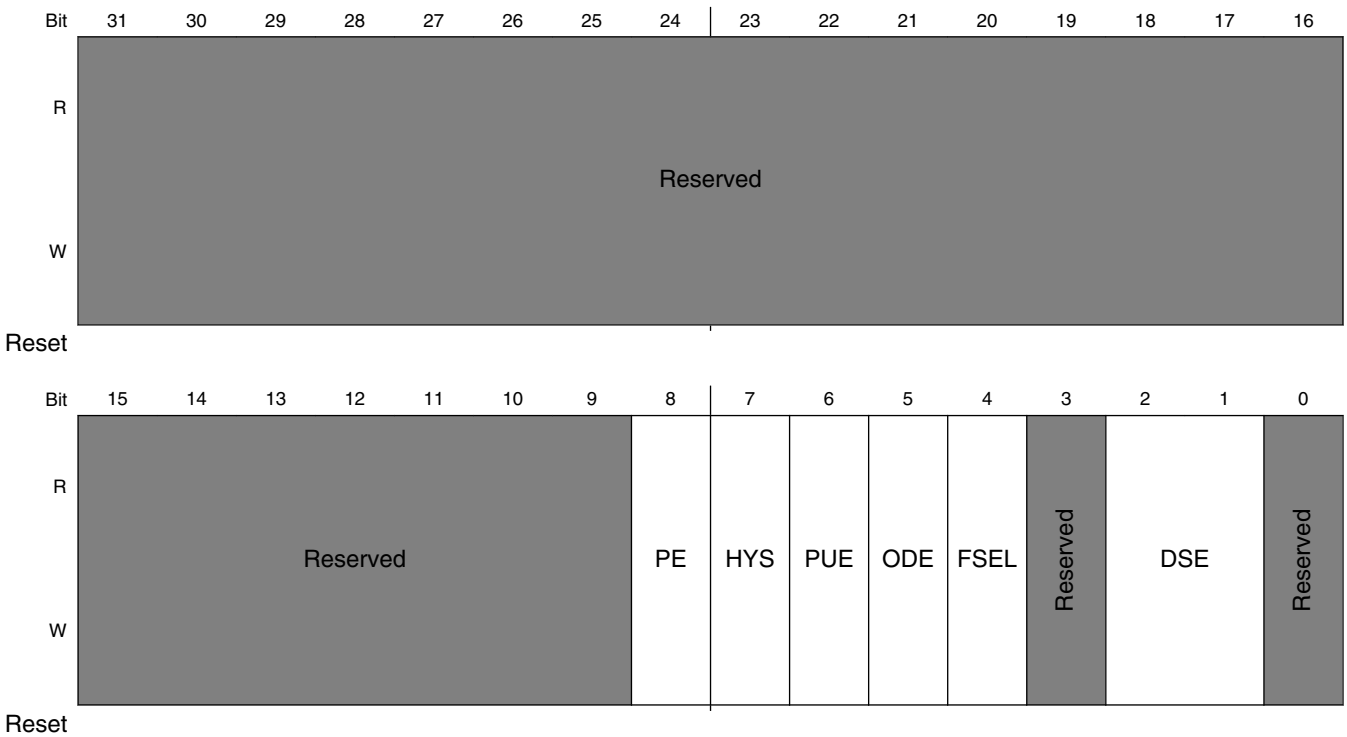
IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MISO field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ECSPI1_MISO 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ECSPI1_MISO 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ECSPI1_MISO 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.271 SW_PAD_CTL_PAD_ECSPI1_SS0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SS0)

SW_PAD_CTL Register

Address: 3033_0000h base + 44Ch offset = 3033_044Ch



IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SS0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ECSPI1_SS0 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ECSPI1_SS0 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ECSPI1_SS0

Table continues on the next page...

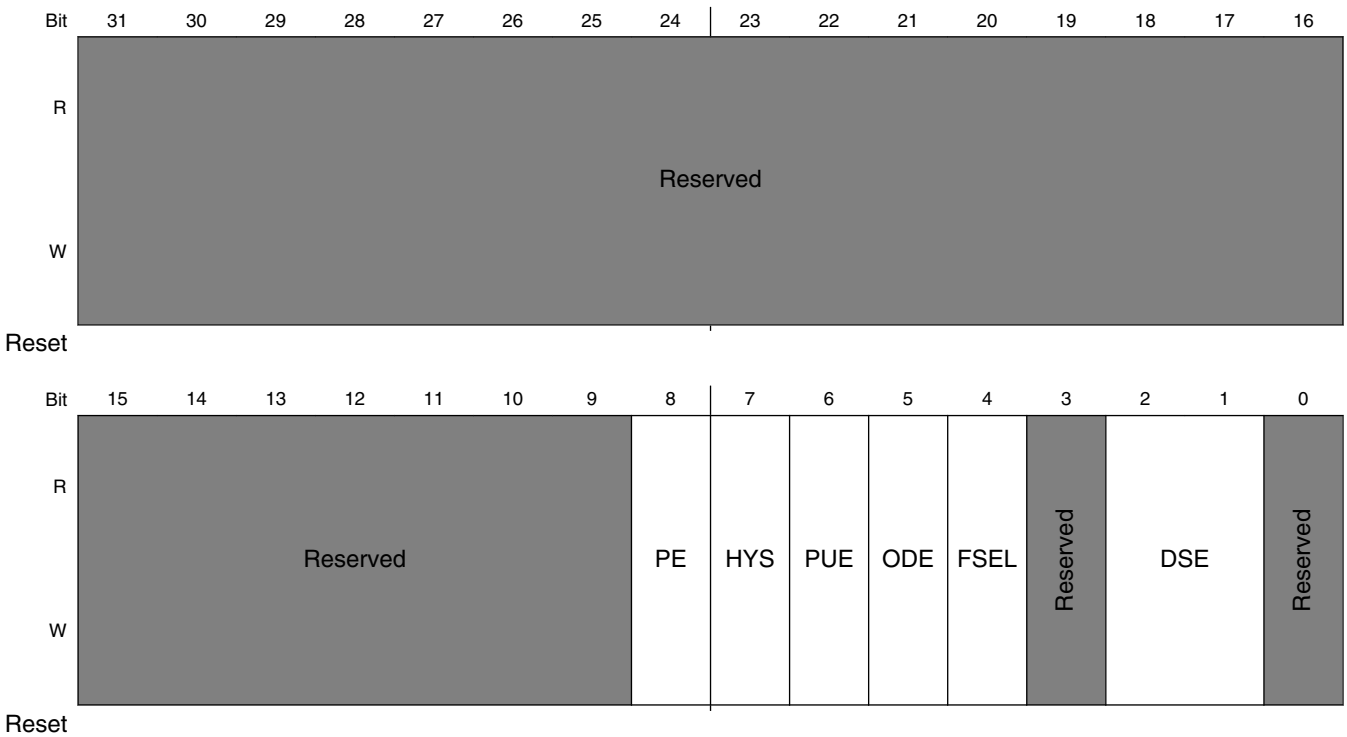
IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SS0 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ECSPI1_SS0 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ECSPI1_SS0 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ECSPI1_SS0 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.272 SW_PAD_CTL_PAD_ECSPI2_SCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SCLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 450h offset = 3033_0450h



IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SCLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ECSPI2_SCLK 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ECSPI2_SCLK 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ECSPI2_SCLK

Table continues on the next page...

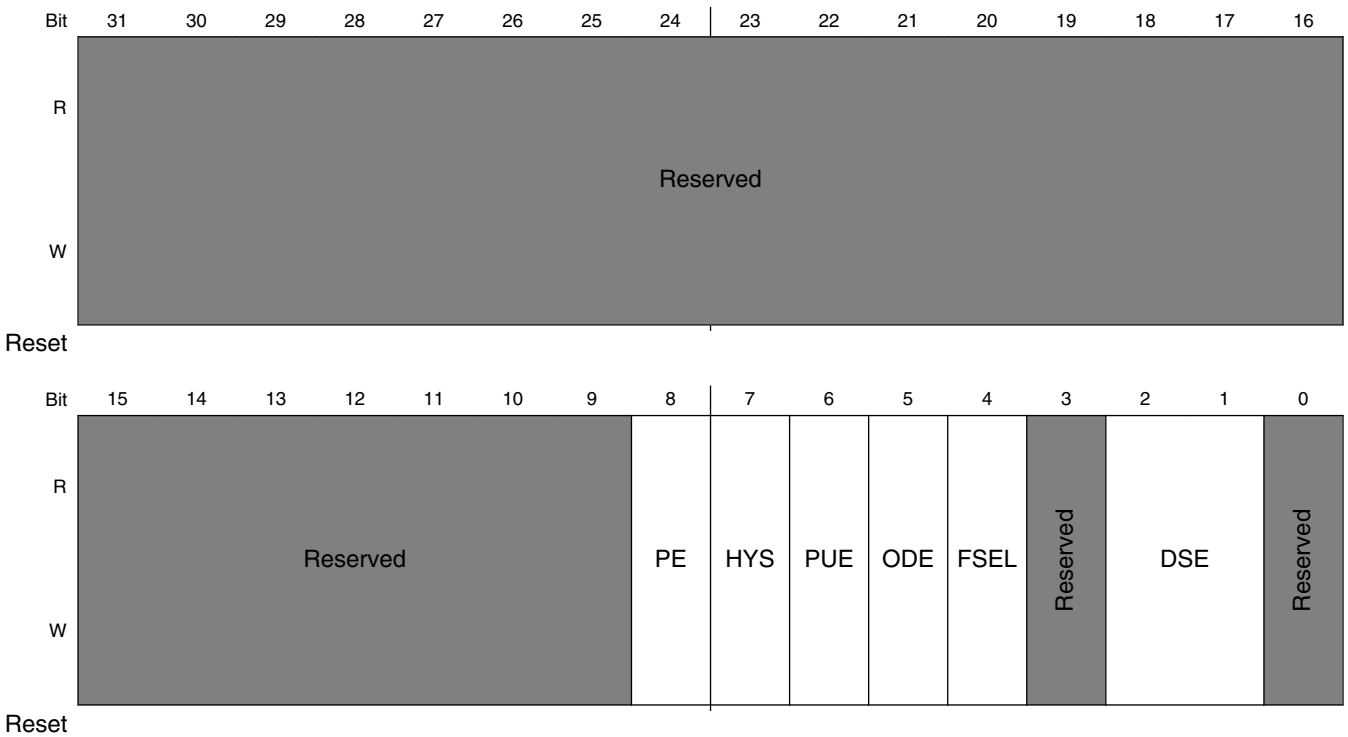
IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SCLK field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ECSPI2_SCLK 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ECSPI2_SCLK 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ECSPI2_SCLK 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.273 SW_PAD_CTL_PAD_ECSPI2_MOSI SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MOSI)

SW_PAD_CTL Register

Address: 3033_0000h base + 454h offset = 3033_0454h



IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MOSI field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ECSPI2_MOSI 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ECSPI2_MOSI 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ECSPI2_MOSI

Table continues on the next page...

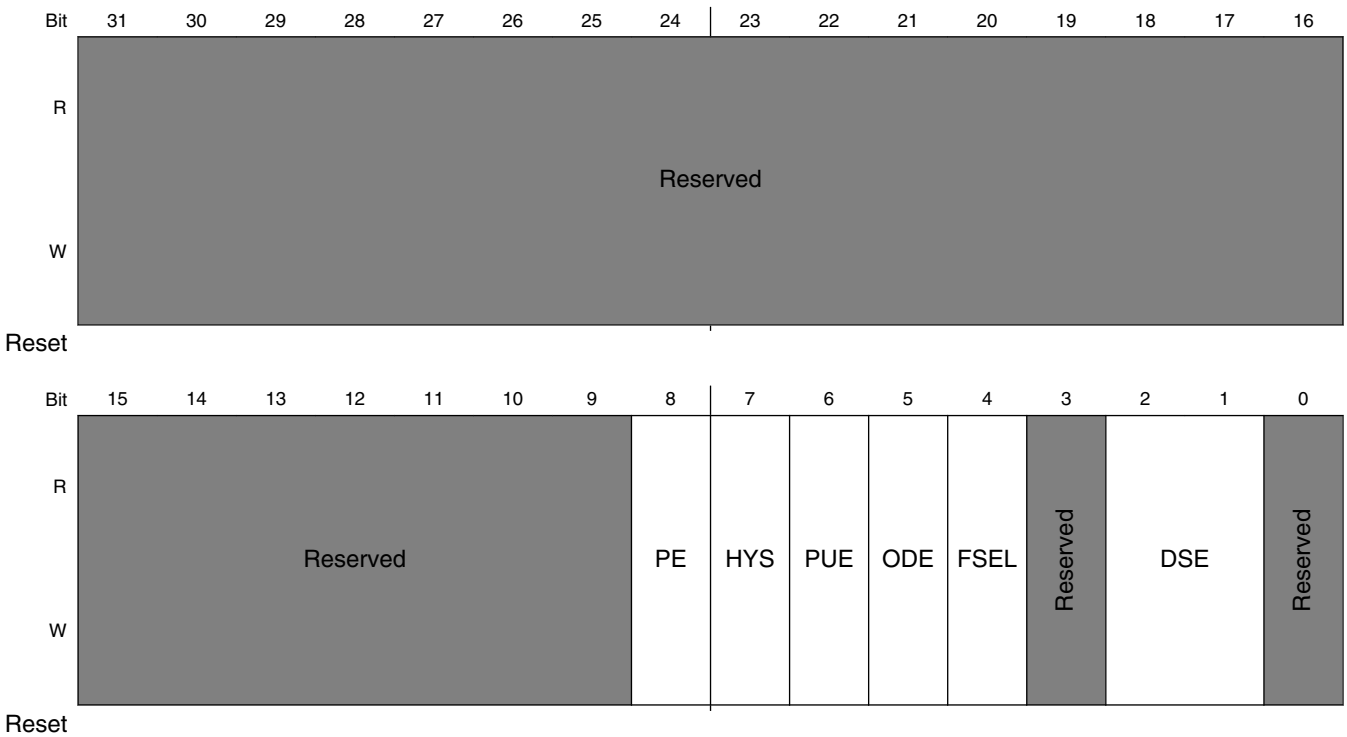
IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MOSI field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ECSPI2_MOSI 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ECSPI2_MOSI 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ECSPI2_MOSI 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.274 SW_PAD_CTL_PAD_ECSPI2_MISO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MISO)

SW_PAD_CTL Register

Address: 3033_0000h base + 458h offset = 3033_0458h



IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MISO field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ECSPI2_MISO 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ECSPI2_MISO 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ECSPI2_MISO

Table continues on the next page...

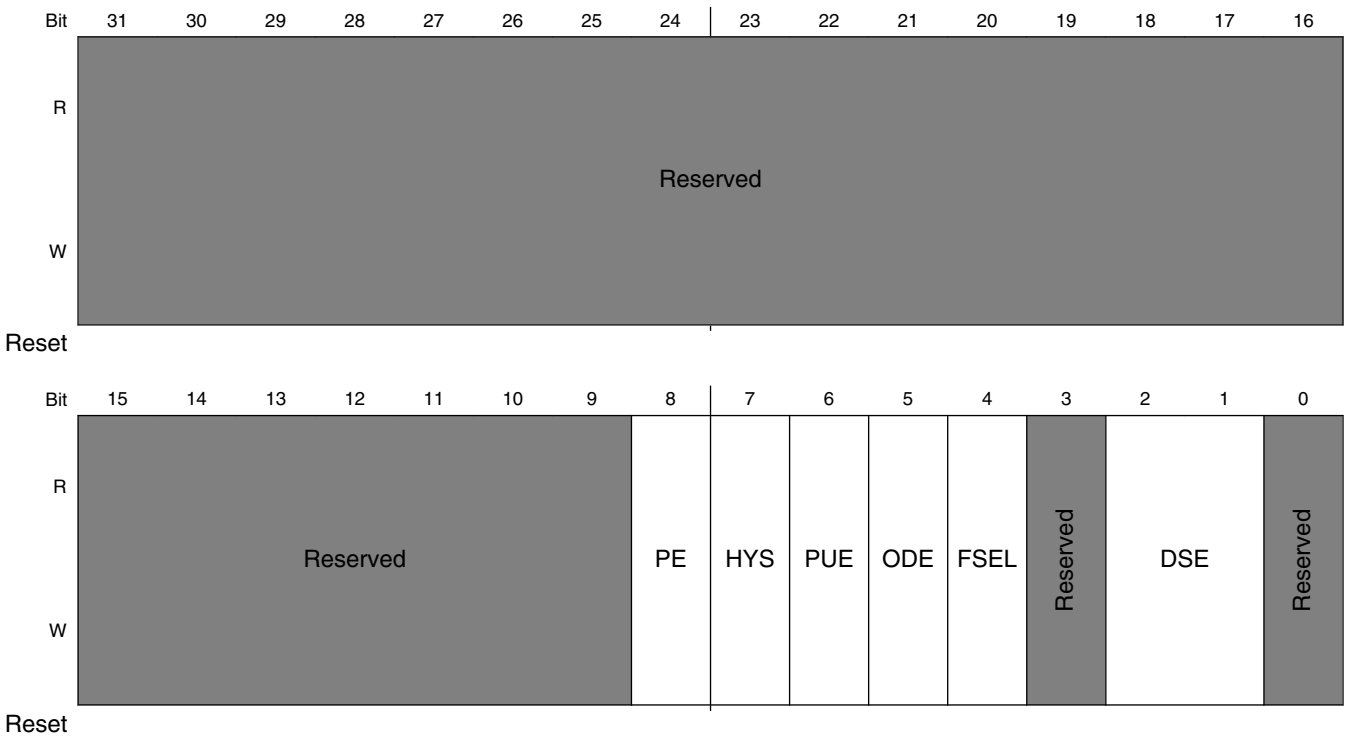
IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MISO field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ECSPI2_MISO 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ECSPI2_MISO 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ECSPI2_MISO 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.275 SW_PAD_CTL_PAD_ECSPI2_SS0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SS0)

SW_PAD_CTL Register

Address: 3033_0000h base + 45Ch offset = 3033_045Ch



IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SS0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: ECSPI2_SS0 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: ECSPI2_SS0 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: ECSPI2_SS0

Table continues on the next page...

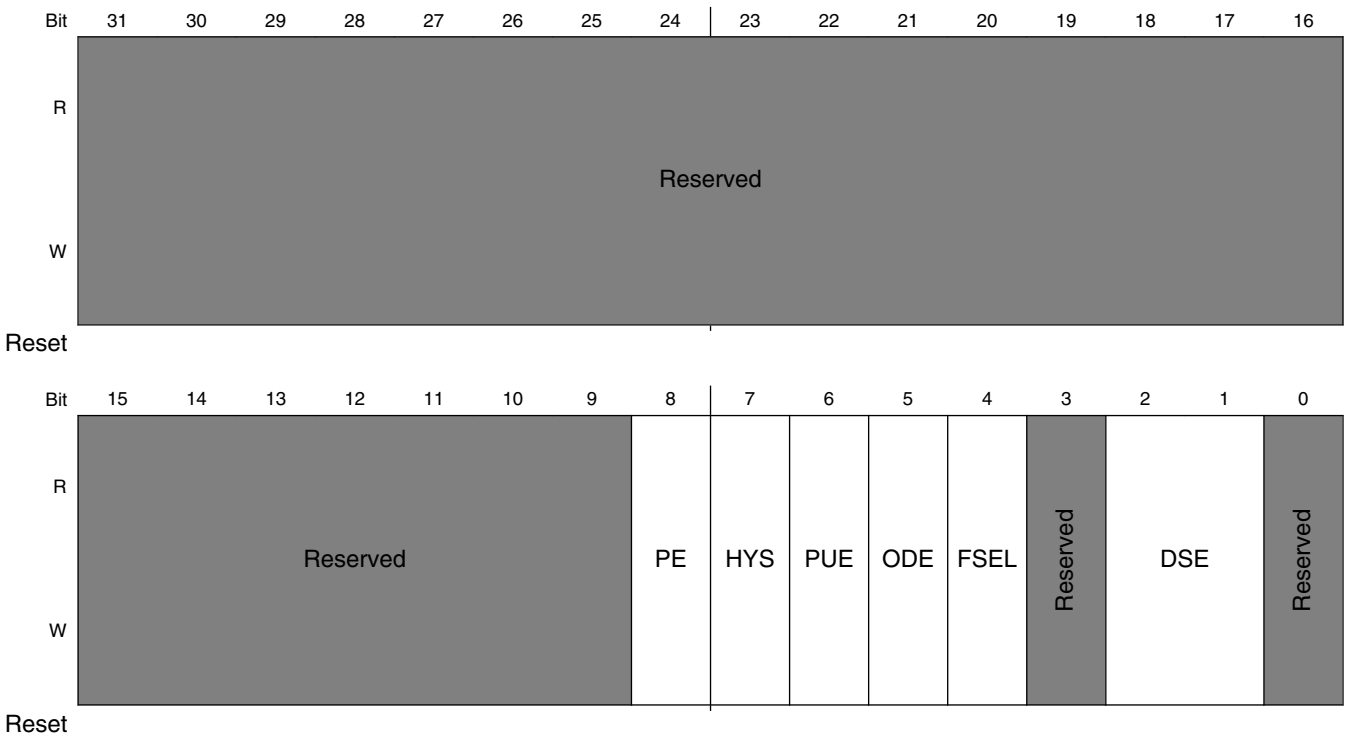
IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SS0 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: ECSPI2_SS0 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: ECSPI2_SS0 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: ECSPI2_SS0 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.276 SW_PAD_CTL_PAD_I2C1_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C1_SCL)

SW_PAD_CTL Register

Address: 3033_0000h base + 460h offset = 3033_0460h



IOMUXC_SW_PAD_CTL_PAD_I2C1_SCL field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: I2C1_SCL 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: I2C1_SCL 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: I2C1_SCL

Table continues on the next page...

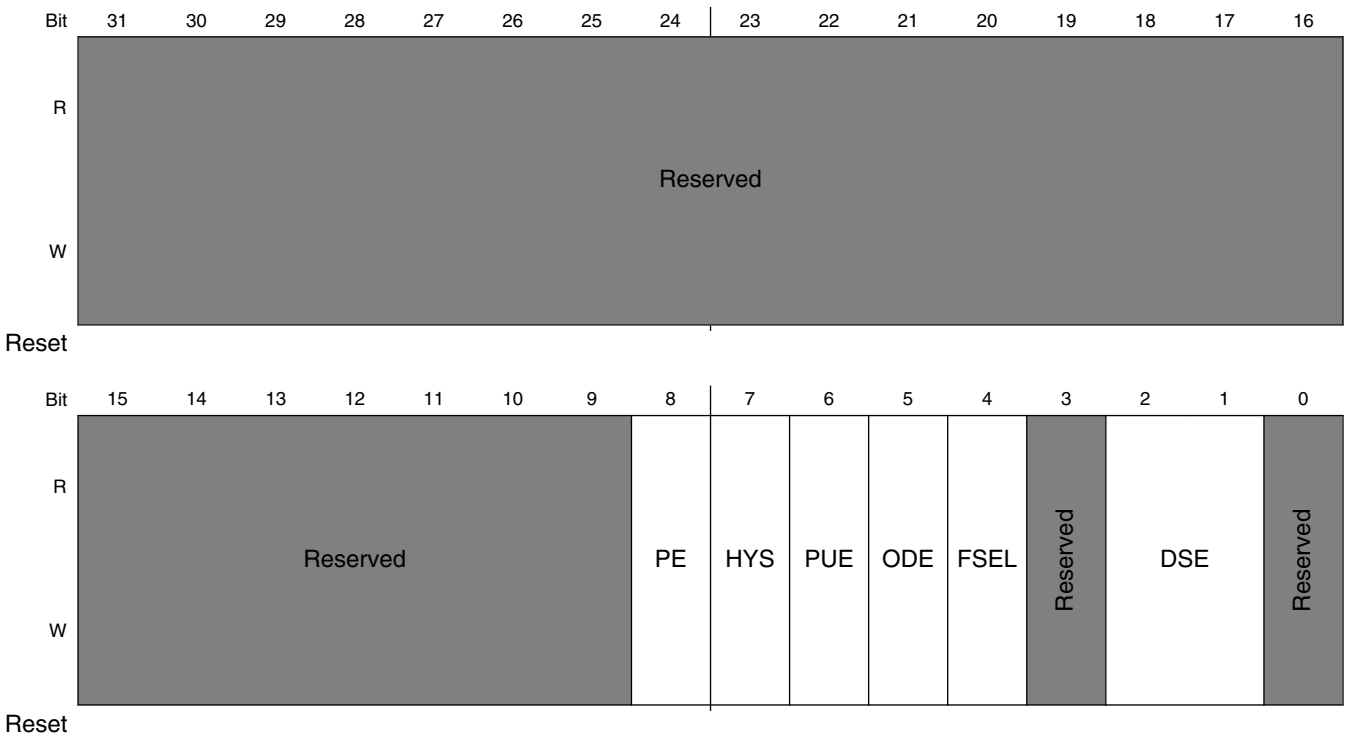
IOMUXC_SW_PAD_CTL_PAD_I2C1_SCL field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: I2C1_SCL 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: I2C1_SCL 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: I2C1_SCL 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.277 SW_PAD_CTL_PAD_I2C1_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C1_SDA)

SW_PAD_CTL Register

Address: 3033_0000h base + 464h offset = 3033_0464h



IOMUXC_SW_PAD_CTL_PAD_I2C1_SDA field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: I2C1_SDA 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: I2C1_SDA 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: I2C1_SDA

Table continues on the next page...

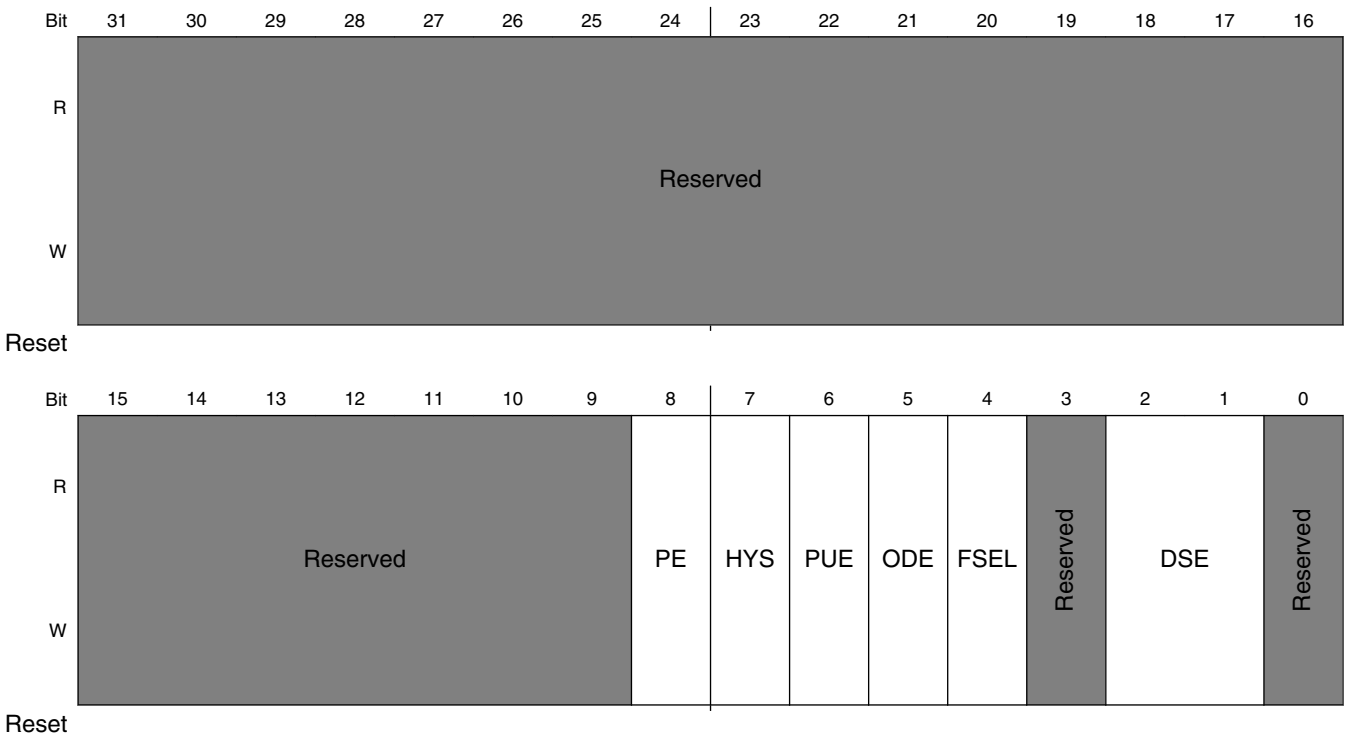
IOMUXC_SW_PAD_CTL_PAD_I2C1_SDA field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: I2C1_SDA 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: I2C1_SDA 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: I2C1_SDA 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.278 SW_PAD_CTL_PAD_I2C2_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C2_SCL)

SW_PAD_CTL Register

Address: 3033_0000h base + 468h offset = 3033_0468h



IOMUXC_SW_PAD_CTL_PAD_I2C2_SCL field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: I2C2_SCL 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: I2C2_SCL 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: I2C2_SCL

Table continues on the next page...

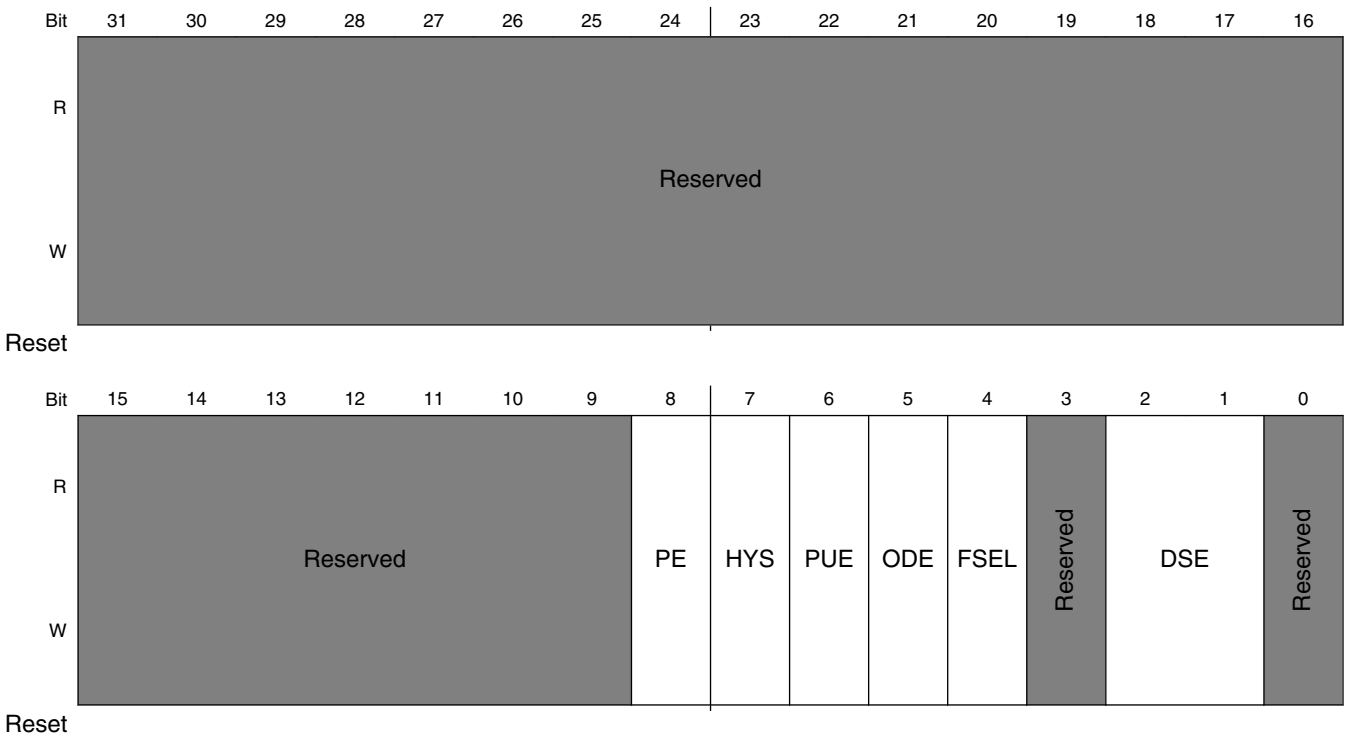
IOMUXC_SW_PAD_CTL_PAD_I2C2_SCL field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: I2C2_SCL 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: I2C2_SCL 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: I2C2_SCL 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.279 SW_PAD_CTL_PAD_I2C2_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C2_SDA)

SW_PAD_CTL Register

Address: 3033_0000h base + 46Ch offset = 3033_046Ch



IOMUXC_SW_PAD_CTL_PAD_I2C2_SDA field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: I2C2_SDA 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: I2C2_SDA 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: I2C2_SDA

Table continues on the next page...

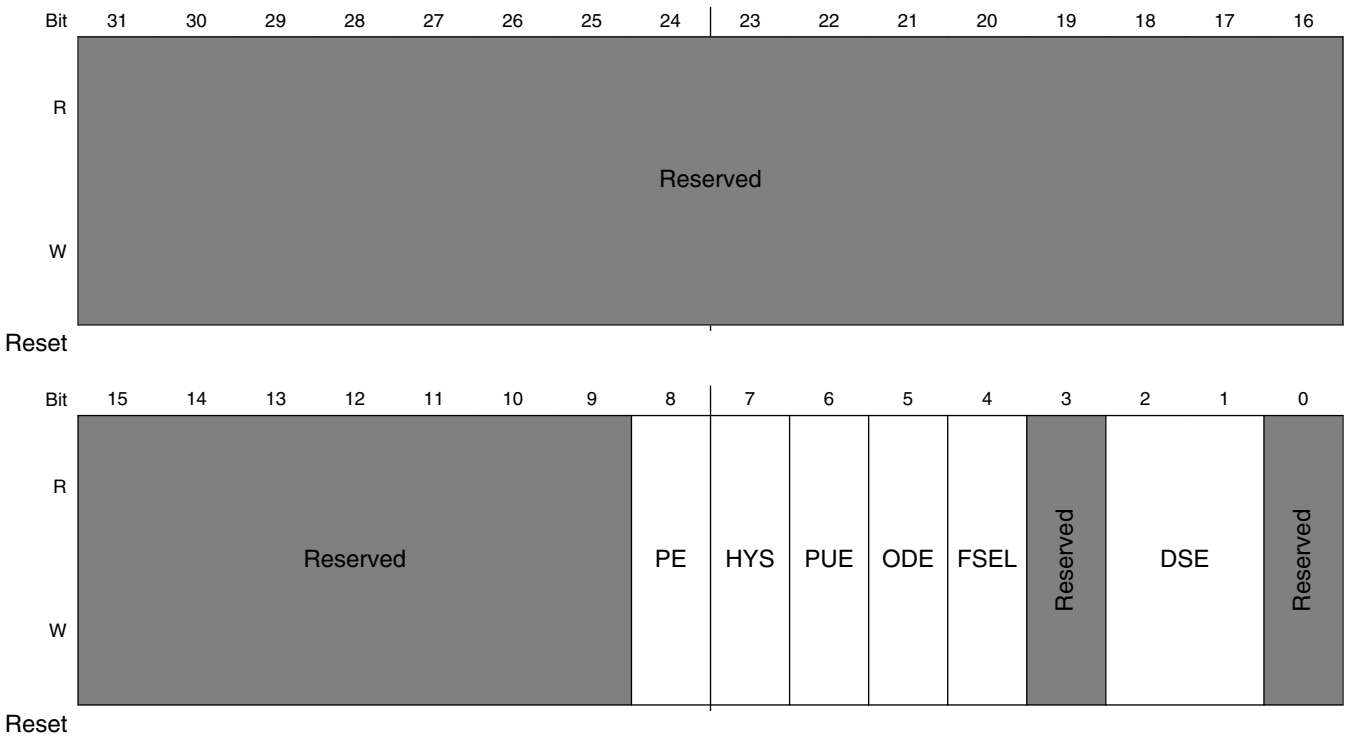
IOMUXC_SW_PAD_CTL_PAD_I2C2_SDA field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: I2C2_SDA 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: I2C2_SDA 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: I2C2_SDA 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.280 SW_PAD_CTL_PAD_I2C3_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C3_SCL)

SW_PAD_CTL Register

Address: 3033_0000h base + 470h offset = 3033_0470h



IOMUXC_SW_PAD_CTL_PAD_I2C3_SCL field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: I2C3_SCL 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: I2C3_SCL 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: I2C3_SCL

Table continues on the next page...

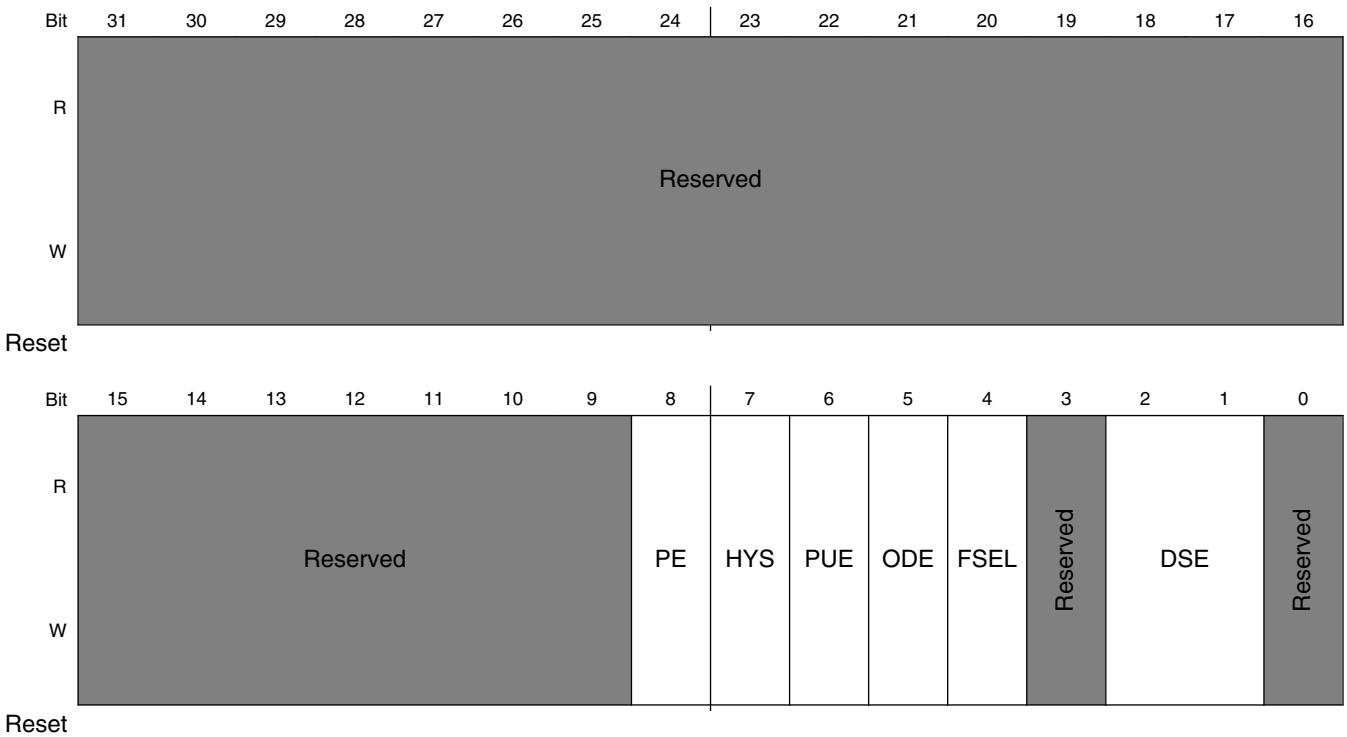
IOMUXC_SW_PAD_CTL_PAD_I2C3_SCL field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: I2C3_SCL 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: I2C3_SCL 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: I2C3_SCL 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.281 SW_PAD_CTL_PAD_I2C3_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C3_SDA)

SW_PAD_CTL Register

Address: 3033_0000h base + 474h offset = 3033_0474h



IOMUXC_SW_PAD_CTL_PAD_I2C3_SDA field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: I2C3_SDA 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: I2C3_SDA 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: I2C3_SDA

Table continues on the next page...

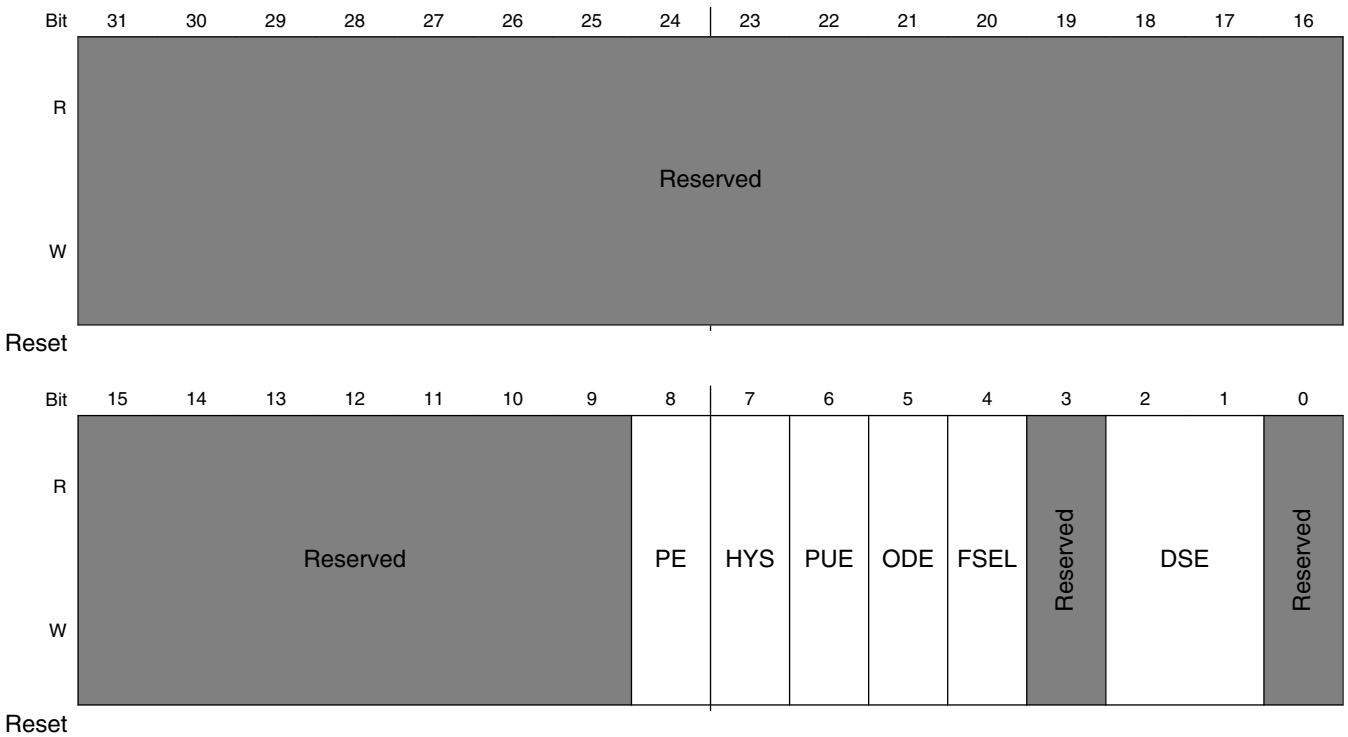
IOMUXC_SW_PAD_CTL_PAD_I2C3_SDA field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: I2C3_SDA 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: I2C3_SDA 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: I2C3_SDA 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.282 SW_PAD_CTL_PAD_I2C4_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C4_SCL)

SW_PAD_CTL Register

Address: 3033_0000h base + 478h offset = 3033_0478h



IOMUXC_SW_PAD_CTL_PAD_I2C4_SCL field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: I2C4_SCL 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: I2C4_SCL 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: I2C4_SCL

Table continues on the next page...

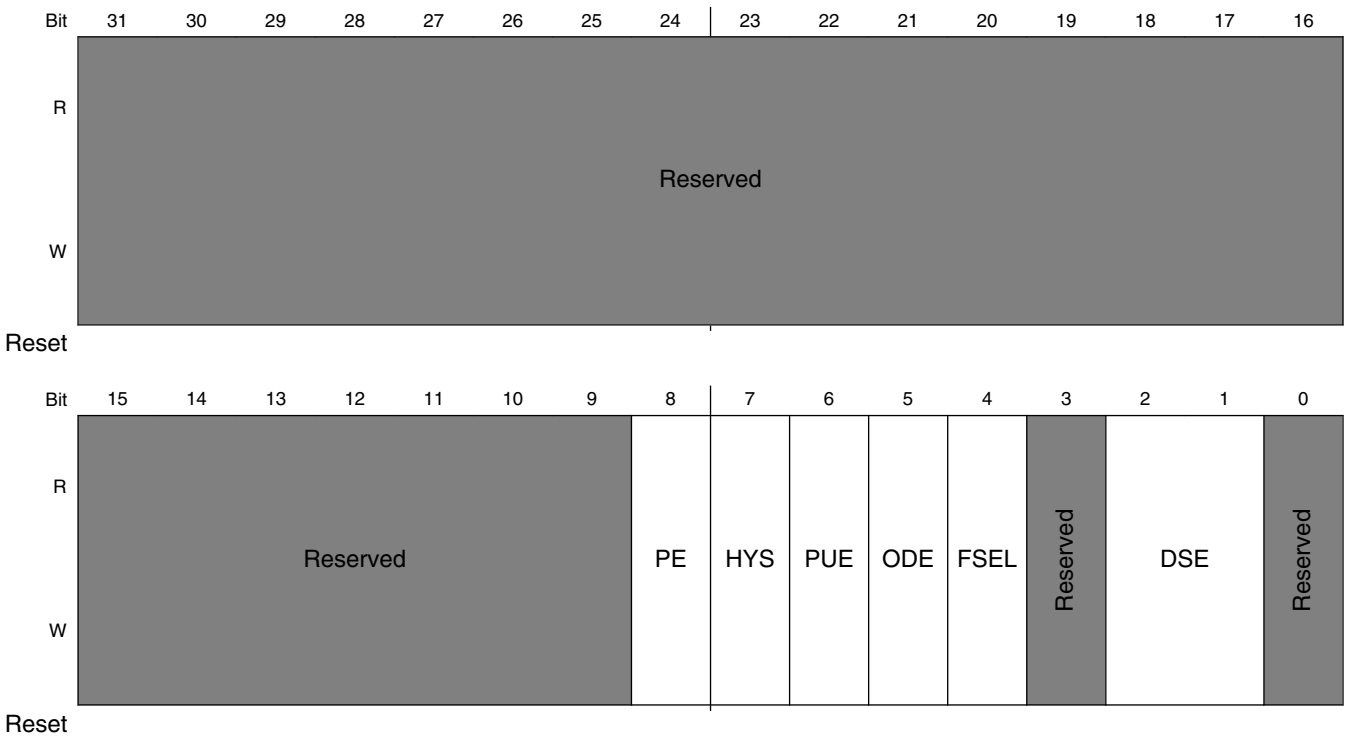
IOMUXC_SW_PAD_CTL_PAD_I2C4_SCL field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: I2C4_SCL 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: I2C4_SCL 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: I2C4_SCL 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.283 SW_PAD_CTL_PAD_I2C4_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C4_SDA)

SW_PAD_CTL Register

Address: 3033_0000h base + 47Ch offset = 3033_047Ch



IOMUXC_SW_PAD_CTL_PAD_I2C4_SDA field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: I2C4_SDA 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: I2C4_SDA 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: I2C4_SDA

Table continues on the next page...

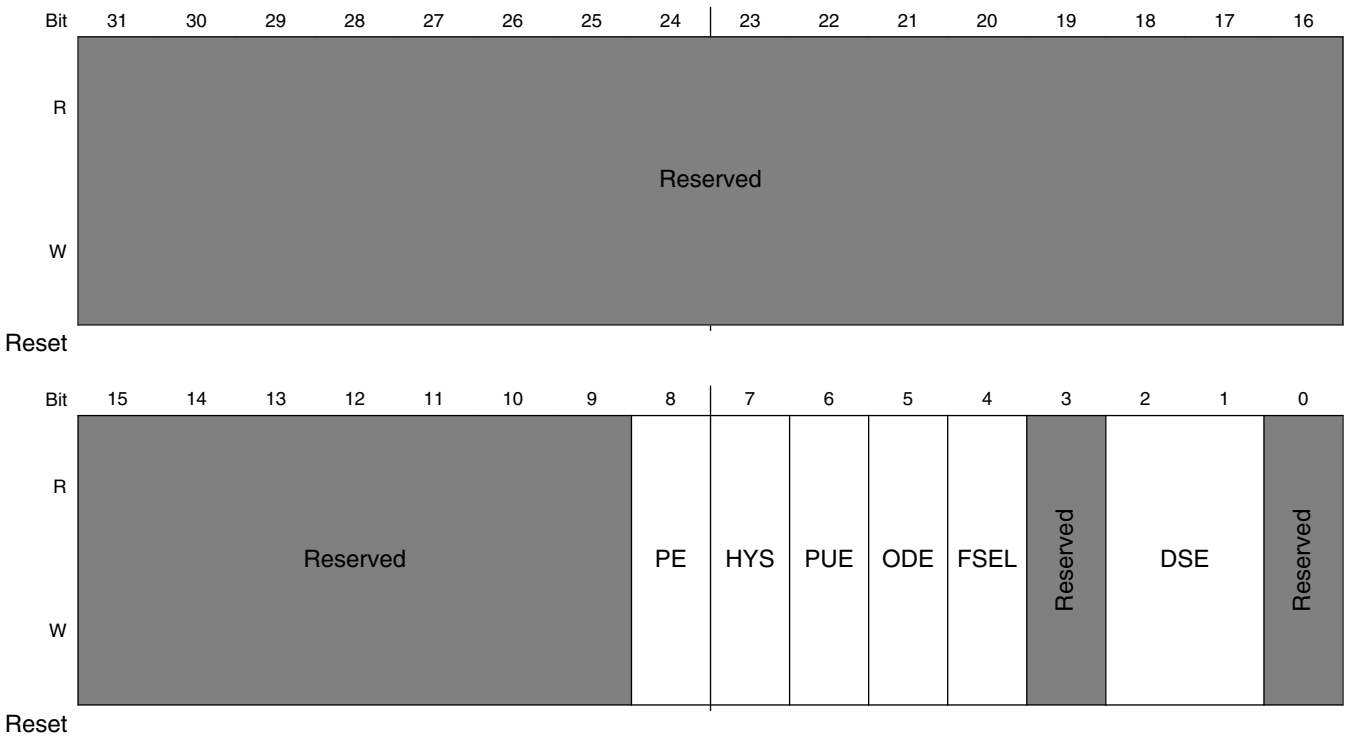
IOMUXC_SW_PAD_CTL_PAD_I2C4_SDA field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: I2C4_SDA 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: I2C4_SDA 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: I2C4_SDA 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.284 SW_PAD_CTL_PAD_UART1_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART1_RXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 480h offset = 3033_0480h



IOMUXC_SW_PAD_CTL_PAD_UART1_RXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: UART1_RXD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: UART1_RXD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: UART1_RXD

Table continues on the next page...

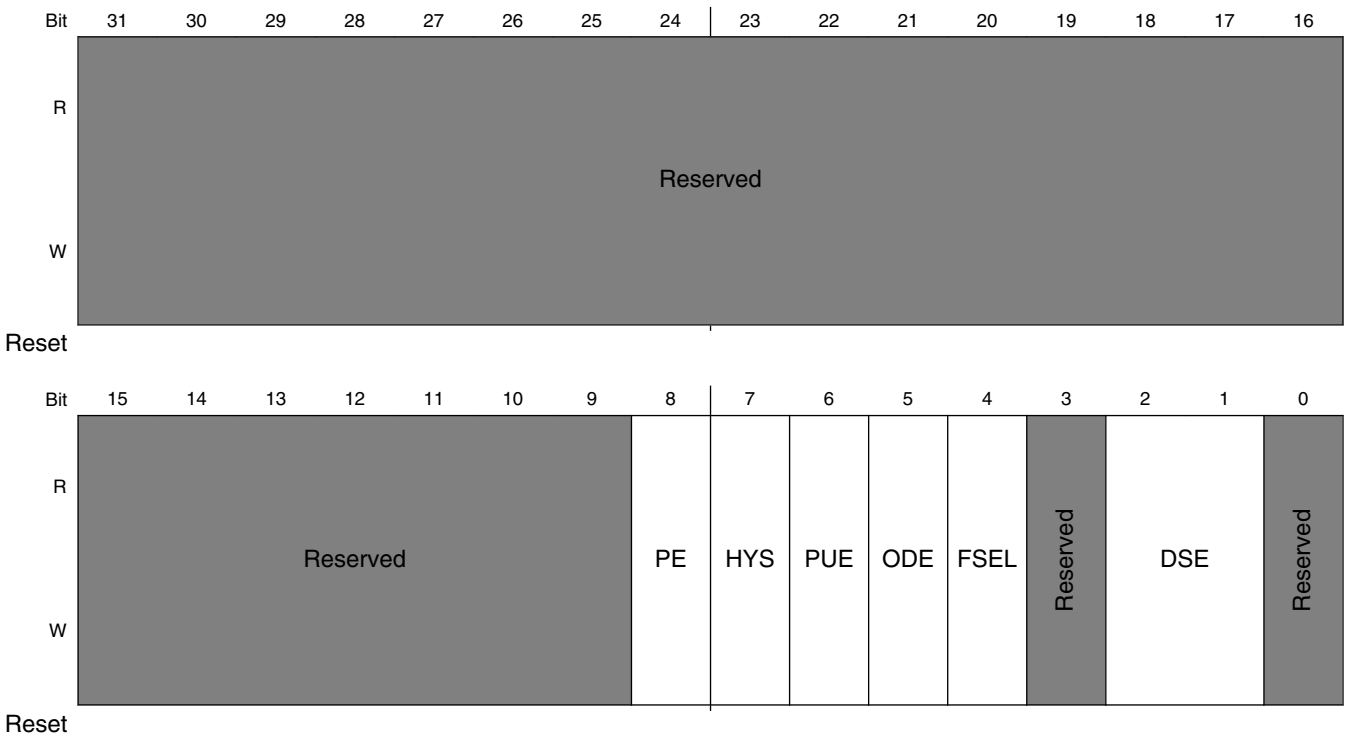
IOMUXC_SW_PAD_CTL_PAD_UART1_RXD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: UART1_RXD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: UART1_RXD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: UART1_RXD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.285 SW_PAD_CTL_PAD_UART1_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART1_TXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 484h offset = 3033_0484h



IOMUXC_SW_PAD_CTL_PAD_UART1_TXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: UART1_TXD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: UART1_TXD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: UART1_TXD

Table continues on the next page...

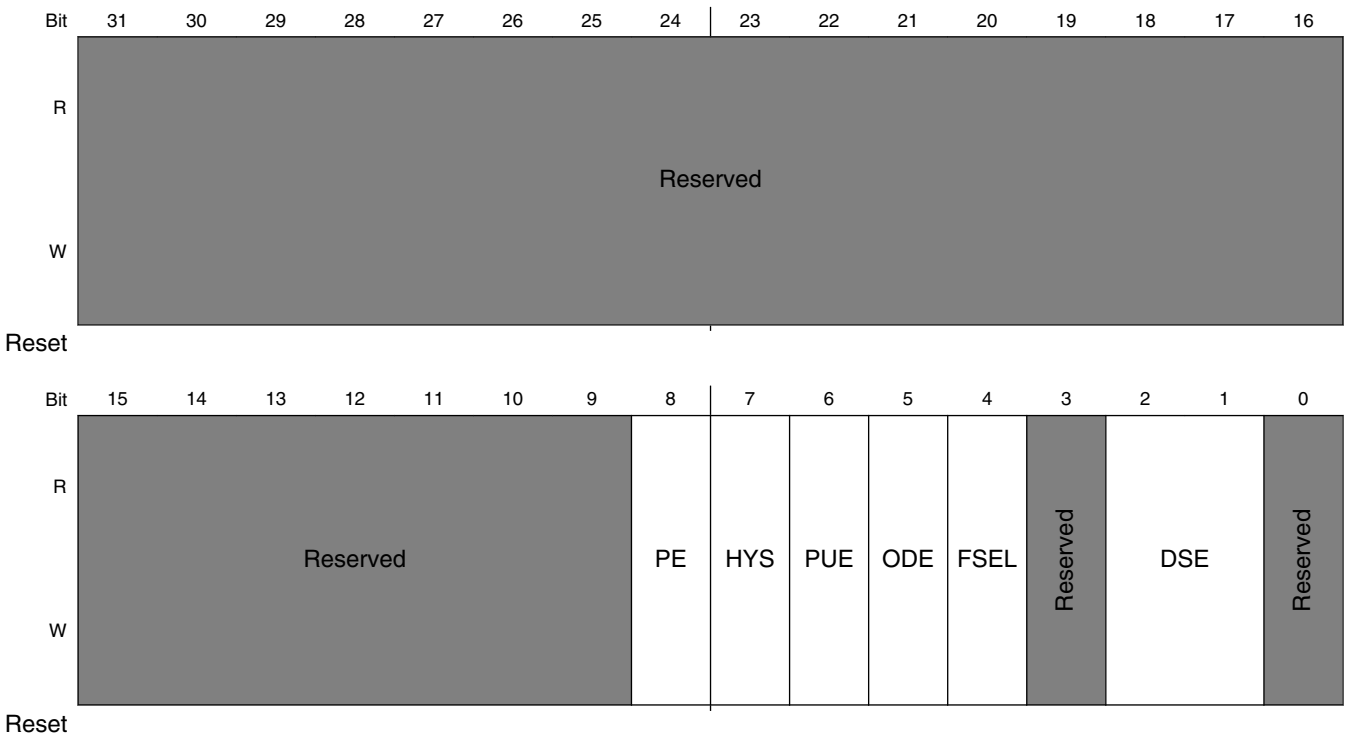
IOMUXC_SW_PAD_CTL_PAD_UART1_TXD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: UART1_TXD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: UART1_TXD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: UART1_TXD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.286 SW_PAD_CTL_PAD_UART2_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART2_RXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 488h offset = 3033_0488h



IOMUXC_SW_PAD_CTL_PAD_UART2_RXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: UART2_RXD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: UART2_RXD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: UART2_RXD

Table continues on the next page...

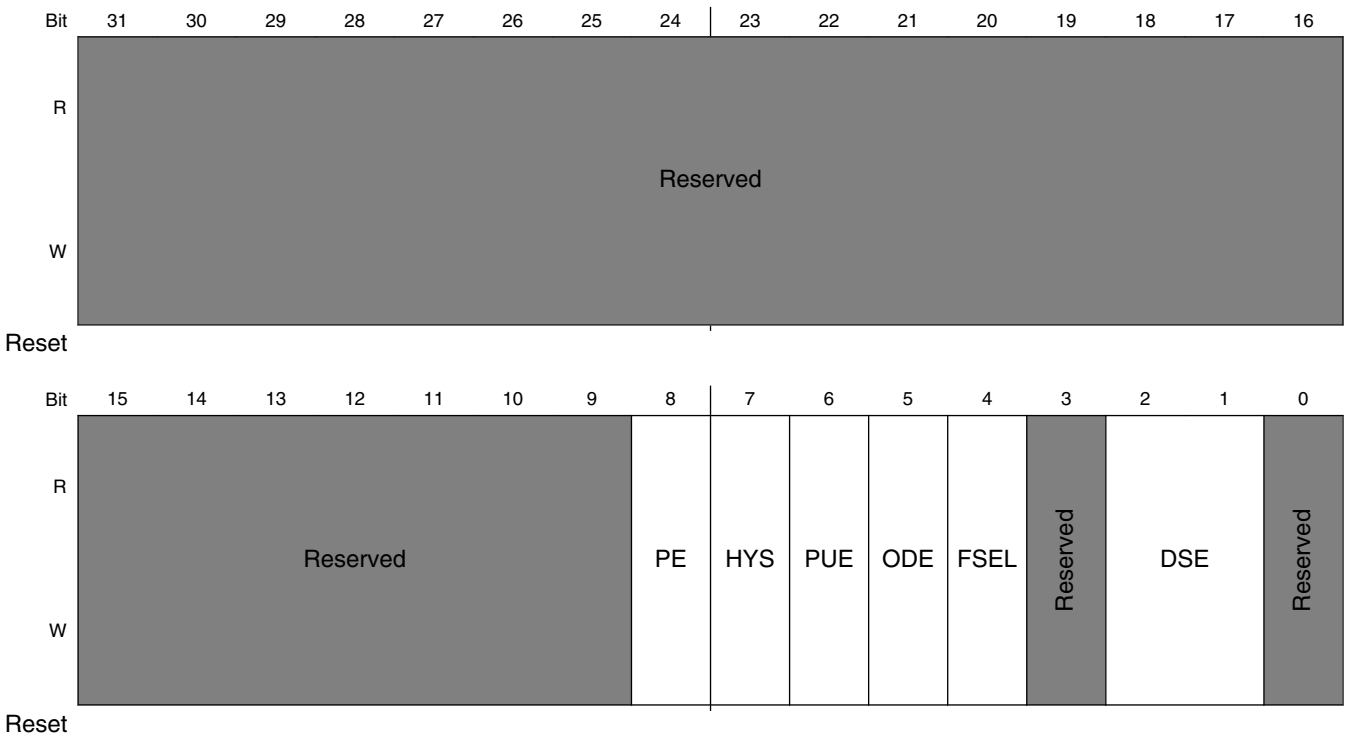
IOMUXC_SW_PAD_CTL_PAD_UART2_RXD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: UART2_RXD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: UART2_RXD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: UART2_RXD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.287 SW_PAD_CTL_PAD_UART2_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART2_TXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 48Ch offset = 3033_048Ch



IOMUXC_SW_PAD_CTL_PAD_UART2_TXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: UART2_TXD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: UART2_TXD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: UART2_TXD

Table continues on the next page...

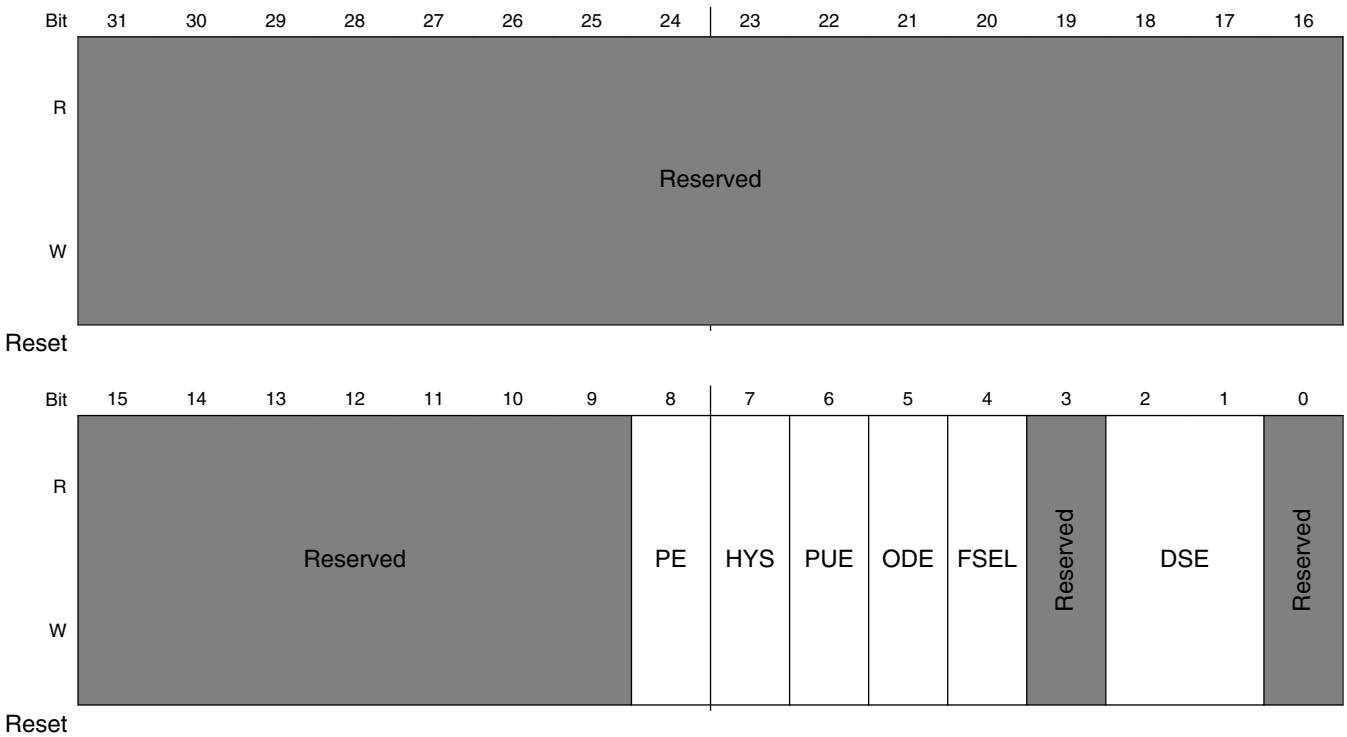
IOMUXC_SW_PAD_CTL_PAD_UART2_TXD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: UART2_TXD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: UART2_TXD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: UART2_TXD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.288 SW_PAD_CTL_PAD_UART3_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART3_RXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 490h offset = 3033_0490h



IOMUXC_SW_PAD_CTL_PAD_UART3_RXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: UART3_RXD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: UART3_RXD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: UART3_RXD

Table continues on the next page...

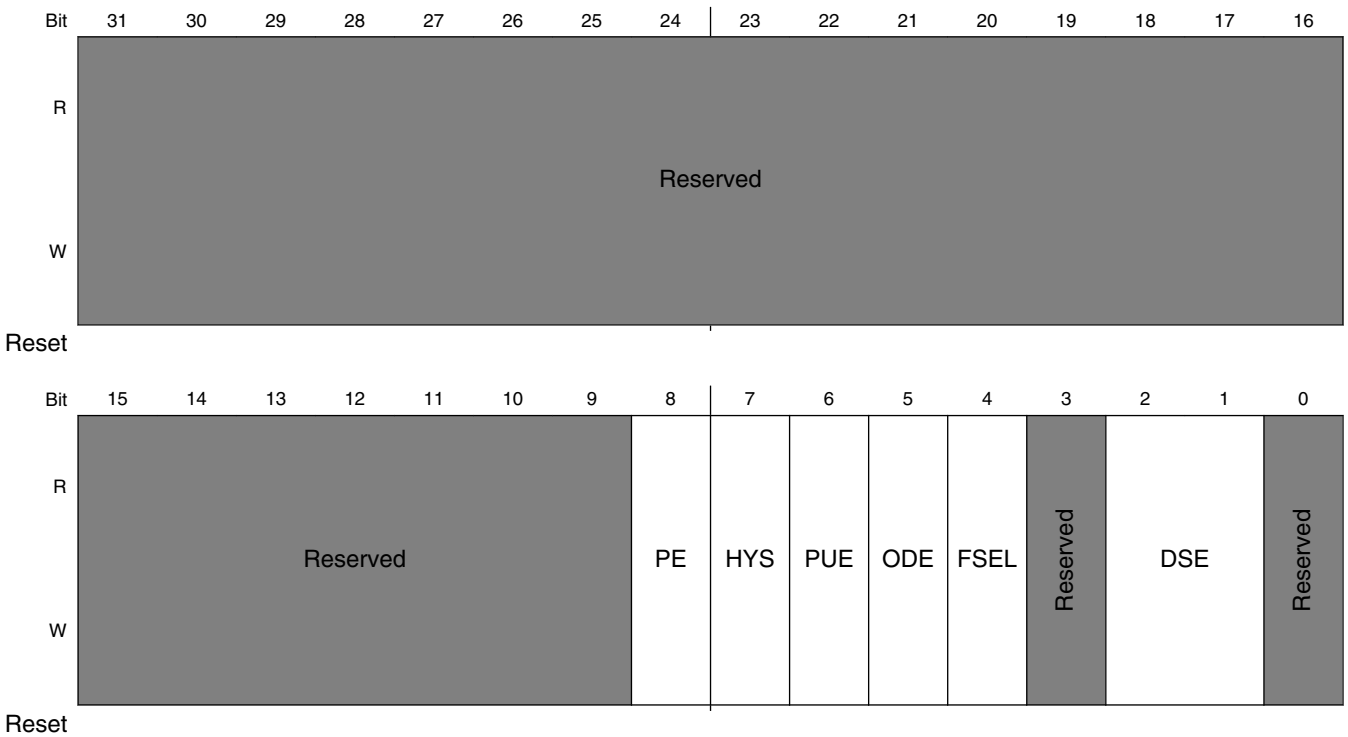
IOMUXC_SW_PAD_CTL_PAD_UART3_RXD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: UART3_RXD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: UART3_RXD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: UART3_RXD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.289 SW_PAD_CTL_PAD_UART3_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART3_TXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 494h offset = 3033_0494h



IOMUXC_SW_PAD_CTL_PAD_UART3_TXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: UART3_TXD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: UART3_TXD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: UART3_TXD

Table continues on the next page...

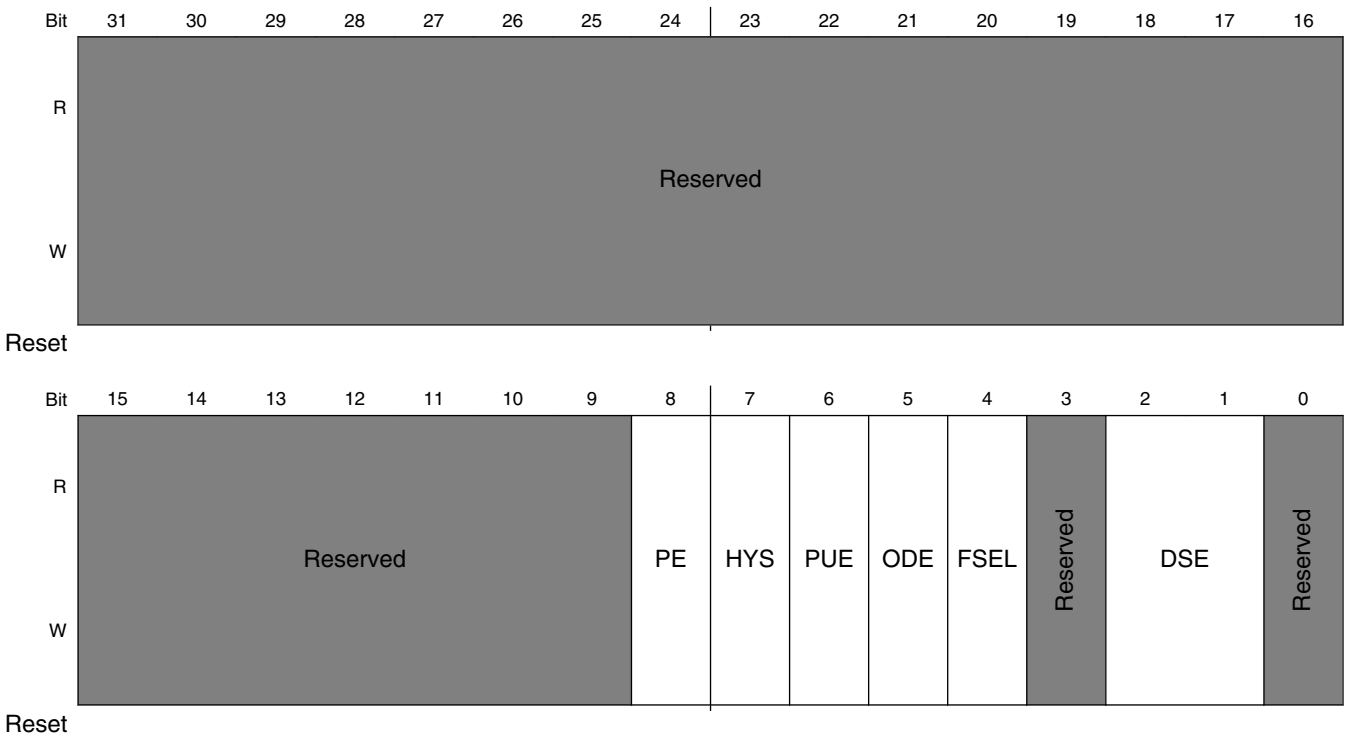
IOMUXC_SW_PAD_CTL_PAD_UART3_TXD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: UART3_TXD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: UART3_TXD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: UART3_TXD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.290 SW_PAD_CTL_PAD_UART4_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART4_RXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 498h offset = 3033_0498h



IOMUXC_SW_PAD_CTL_PAD_UART4_RXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: UART4_RXD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: UART4_RXD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: UART4_RXD

Table continues on the next page...

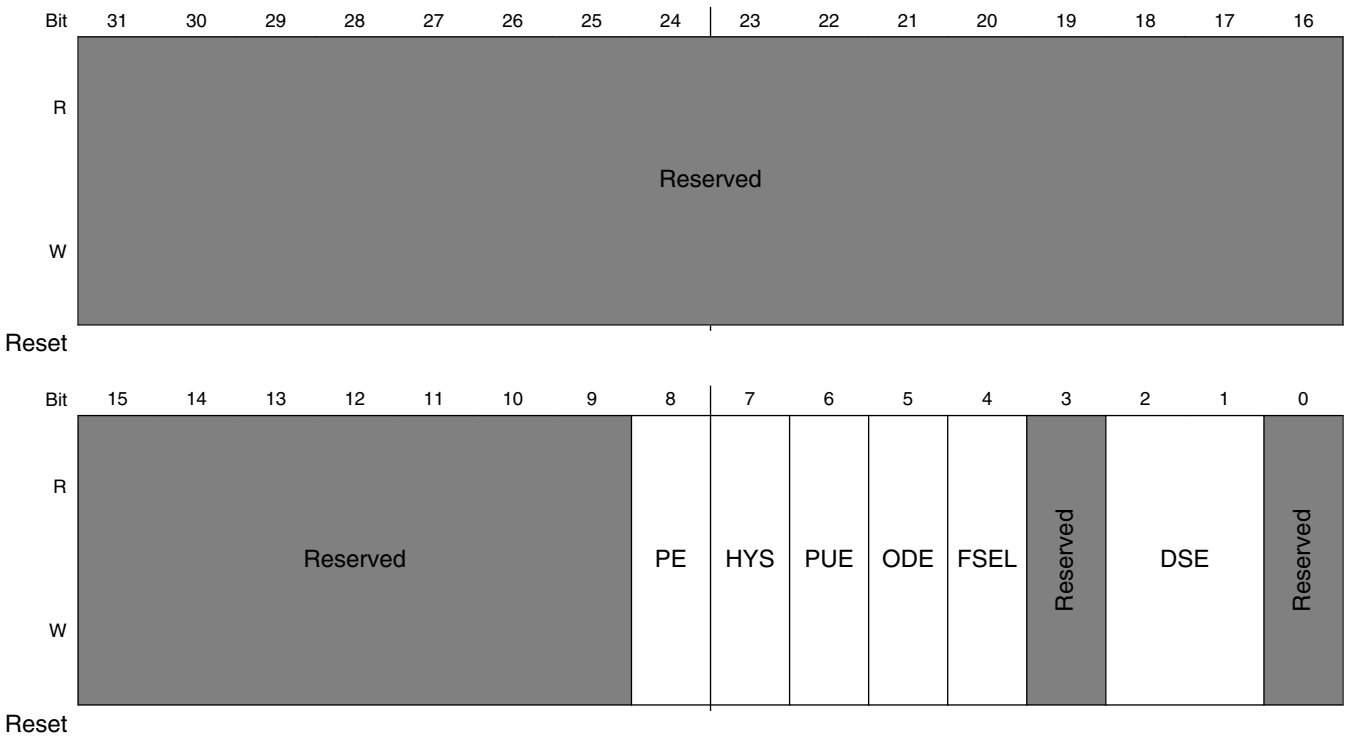
IOMUXC_SW_PAD_CTL_PAD_UART4_RXD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: UART4_RXD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: UART4_RXD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: UART4_RXD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.291 SW_PAD_CTL_PAD_UART4_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART4_TXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 49Ch offset = 3033_049Ch



IOMUXC_SW_PAD_CTL_PAD_UART4_TXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: UART4_TXD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: UART4_TXD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: UART4_TXD

Table continues on the next page...

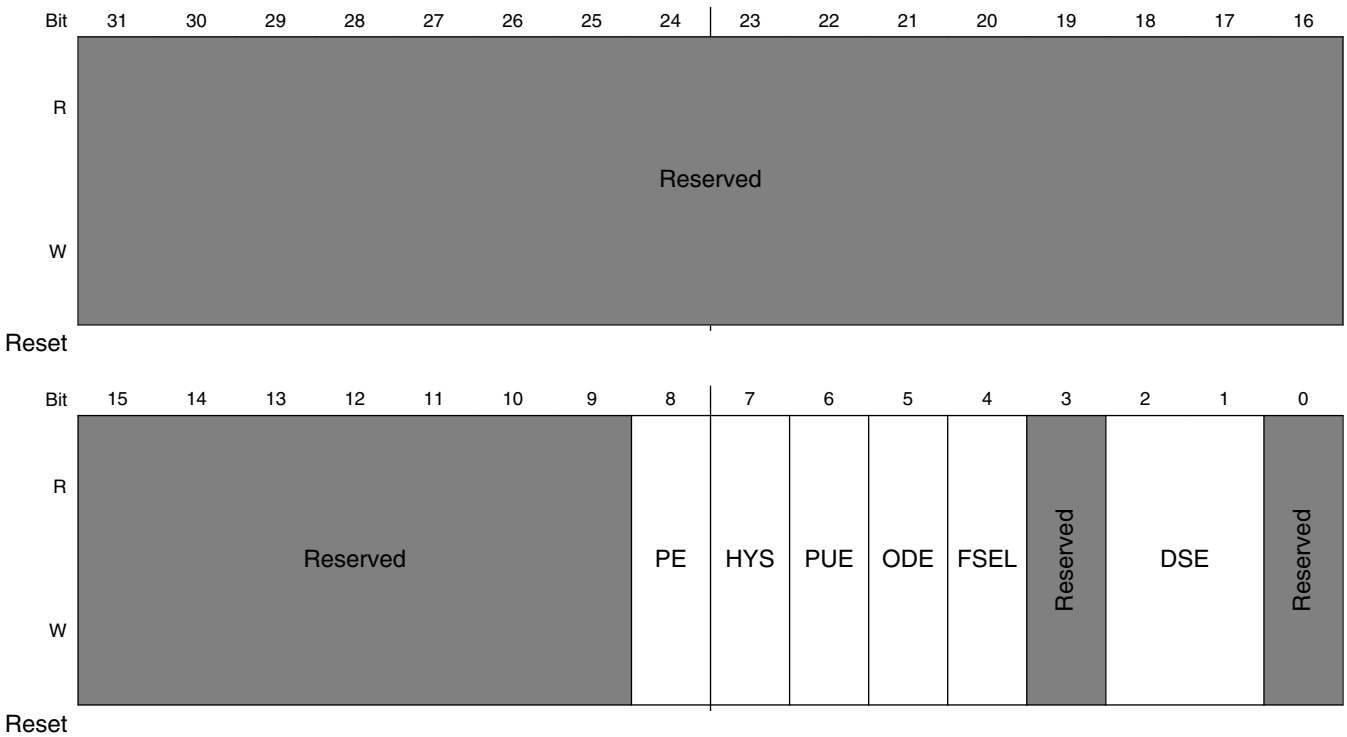
IOMUXC_SW_PAD_CTL_PAD_UART4_TXD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: UART4_TXD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: UART4_TXD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: UART4_TXD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.292 SW_PAD_CTL_PAD_HDMI_DDC_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_HDMI_DDC_SCL)

SW_PAD_CTL Register

Address: 3033_0000h base + 4A0h offset = 3033_04A0h



IOMUXC_SW_PAD_CTL_PAD_HDMI_DDC_SCL field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: HDMI_DDC_SCL 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: HDMI_DDC_SCL 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: HDMI_DDC_SCL

Table continues on the next page...

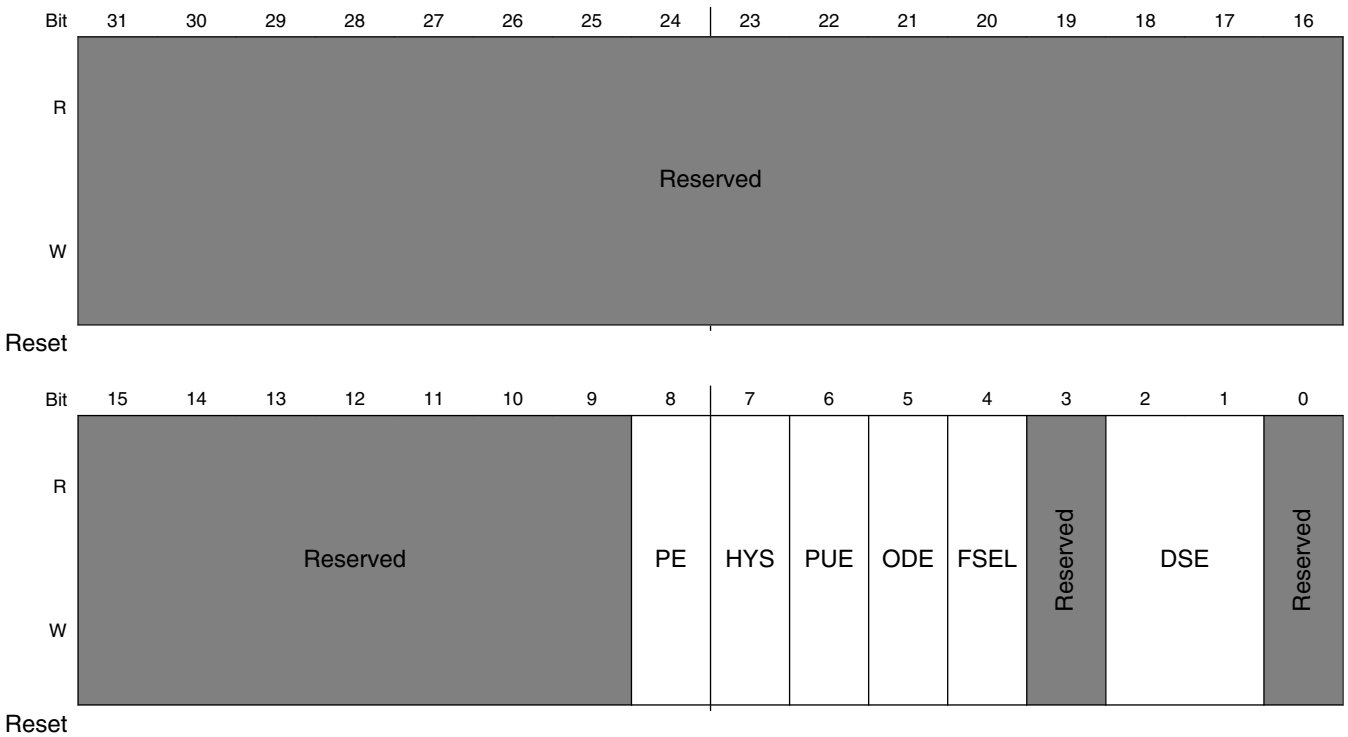
IOMUXC_SW_PAD_CTL_PAD_HDMI_DDC_SCL field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: HDMI_DDC_SCL 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: HDMI_DDC_SCL 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: HDMI_DDC_SCL 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.293 SW_PAD_CTL_PAD_HDMI_DDC_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_HDMI_DDC_SDA)

SW_PAD_CTL Register

Address: 3033_0000h base + 4A4h offset = 3033_04A4h



IOMUXC_SW_PAD_CTL_PAD_HDMI_DDC_SDA field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: HDMI_DDC_SDA 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: HDMI_DDC_SDA 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: HDMI_DDC_SDA

Table continues on the next page...

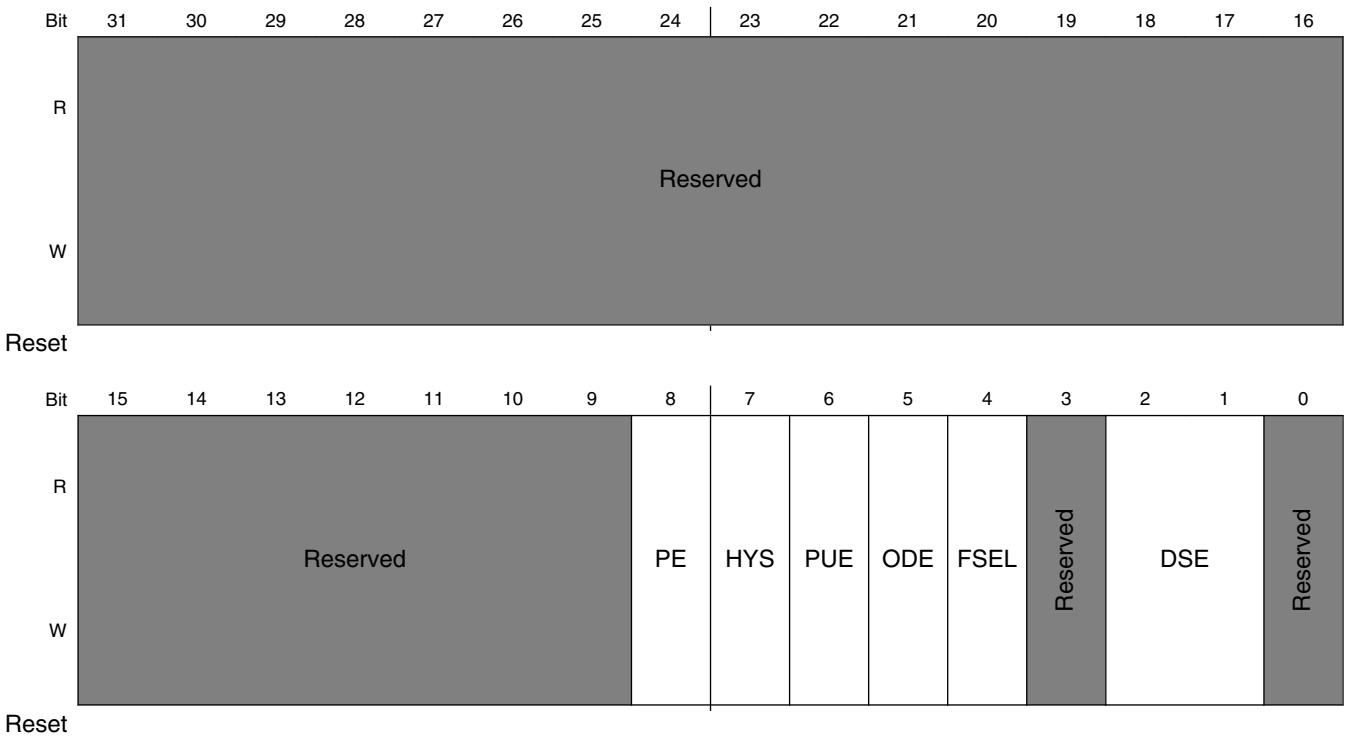
IOMUXC_SW_PAD_CTL_PAD_HDMI_DDC_SDA field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: HDMI_DDC_SDA 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: HDMI_DDC_SDA 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: HDMI_DDC_SDA 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.294 SW_PAD_CTL_PAD_HDMI_CEC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_HDMI_CEC)

SW_PAD_CTL Register

Address: 3033_0000h base + 4A8h offset = 3033_04A8h



IOMUXC_SW_PAD_CTL_PAD_HDMI_CEC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: HDMI_CEC 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: HDMI_CEC 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: HDMI_CEC

Table continues on the next page...

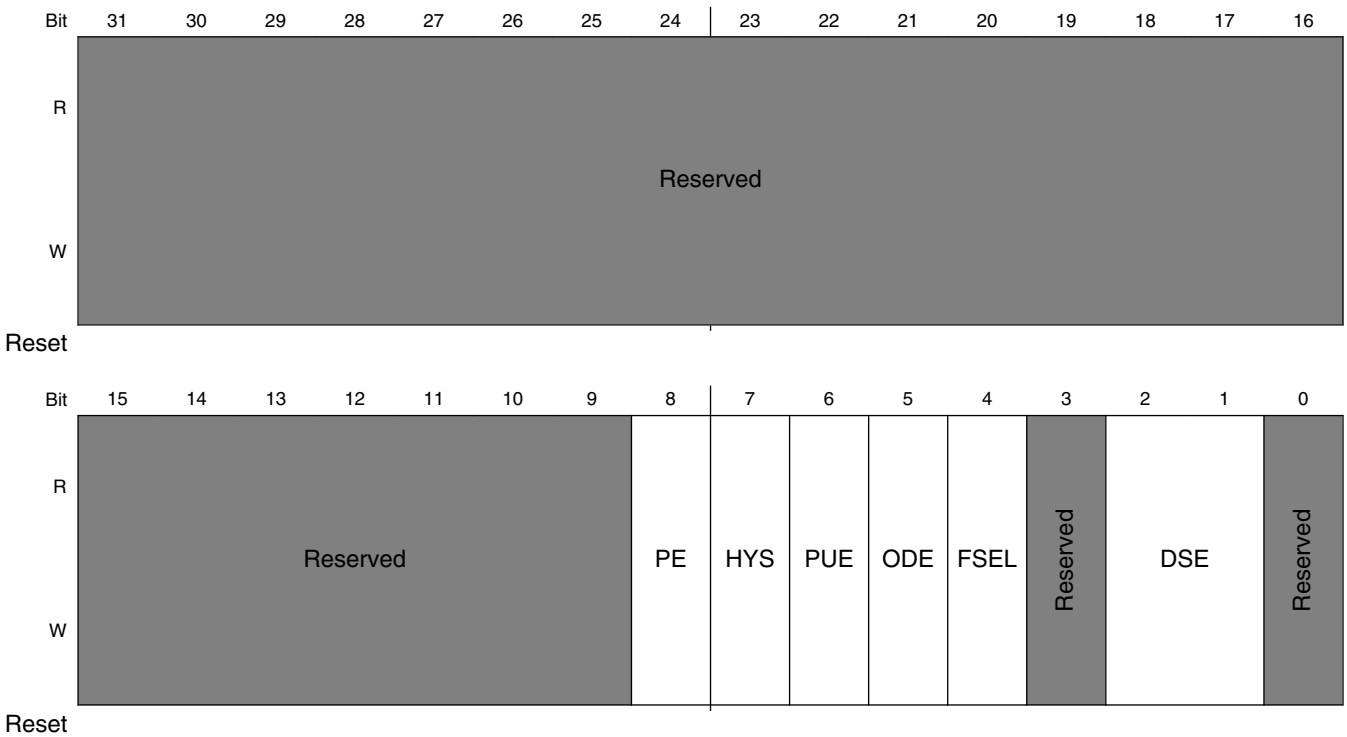
IOMUXC_SW_PAD_CTL_PAD_HDMI_CEC field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: HDMI_CEC 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: HDMI_CEC 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: HDMI_CEC 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.295 SW_PAD_CTL_PAD_HDMI_HPD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_HDMI_HPD)

SW_PAD_CTL Register

Address: 3033_0000h base + 4ACh offset = 3033_04ACh



IOMUXC_SW_PAD_CTL_PAD_HDMI_HPD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: HDMI_HPD 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: HDMI_HPD 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: HDMI_HPD

Table continues on the next page...

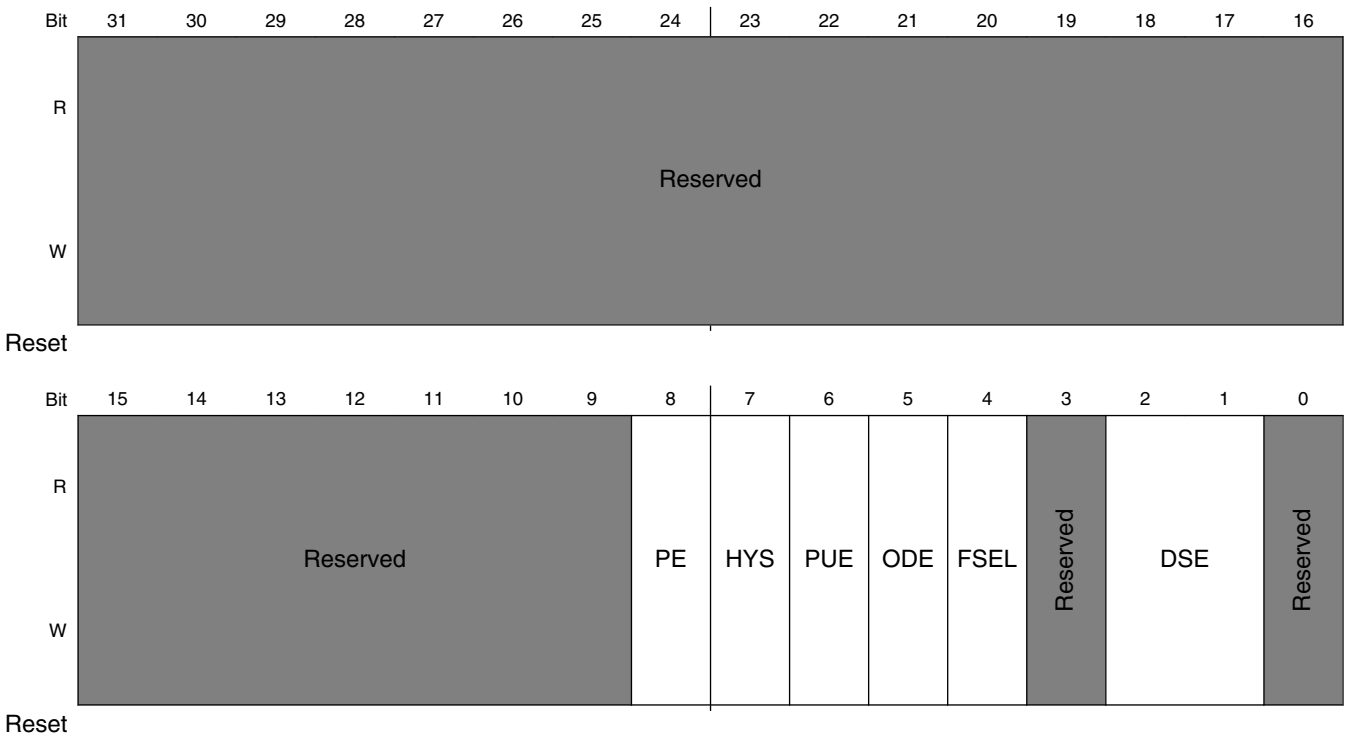
IOMUXC_SW_PAD_CTL_PAD_HDMI_HPD field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: HDMI_HPD 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: HDMI_HPD 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: HDMI_HPD 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.296 SW_PAD_CTL_PAD_CLKIN1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_CLKIN1)

SW_PAD_CTL Register

Address: 3033_0000h base + 4B0h offset = 3033_04B0h



IOMUXC_SW_PAD_CTL_PAD_CLKIN1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: CLKIN1 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: CLKIN1 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: CLKIN1

Table continues on the next page...

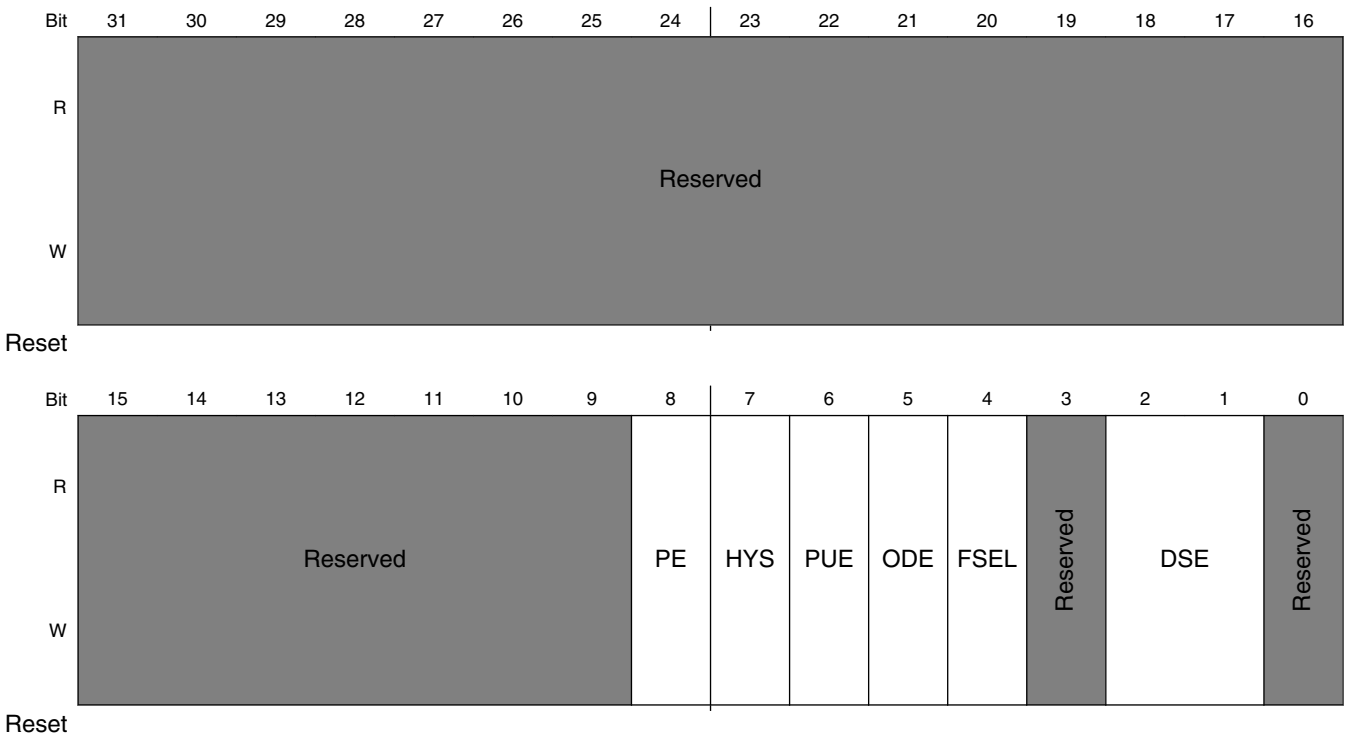
IOMUXC_SW_PAD_CTL_PAD_CLKIN1 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: CLKIN1 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: CLKIN1 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: CLKIN1 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.297 SW_PAD_CTL_PAD_CLKIN2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_CLKIN2)

SW_PAD_CTL Register

Address: 3033_0000h base + 4B4h offset = 3033_04B4h



IOMUXC_SW_PAD_CTL_PAD_CLKIN2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: CLKIN2 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: CLKIN2 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: CLKIN2

Table continues on the next page...

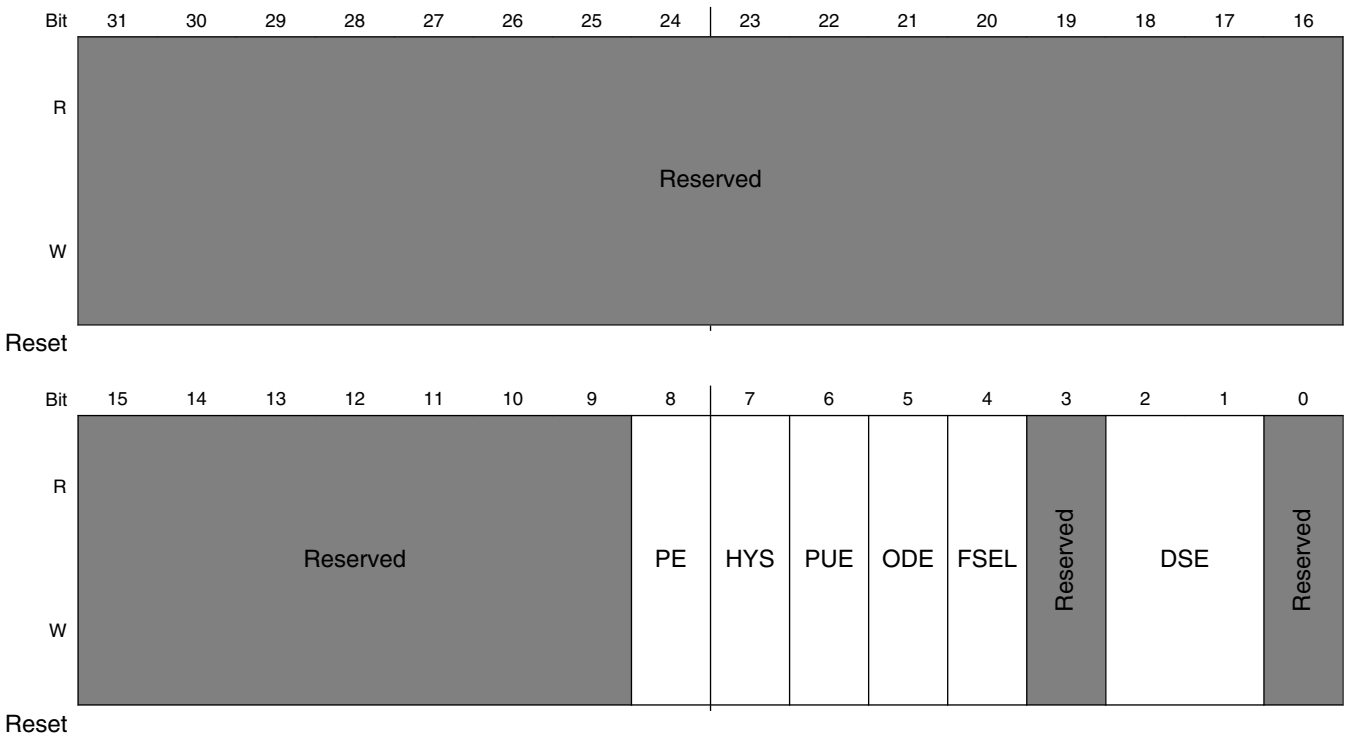
IOMUXC_SW_PAD_CTL_PAD_CLKIN2 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: CLKIN2 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: CLKIN2 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: CLKIN2 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.298 SW_PAD_CTL_PAD_CLKOUT1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_CLKOUT1)

SW_PAD_CTL Register

Address: 3033_0000h base + 4B8h offset = 3033_04B8h



IOMUXC_SW_PAD_CTL_PAD_CLKOUT1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: CLKOUT1 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: CLKOUT1 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: CLKOUT1

Table continues on the next page...

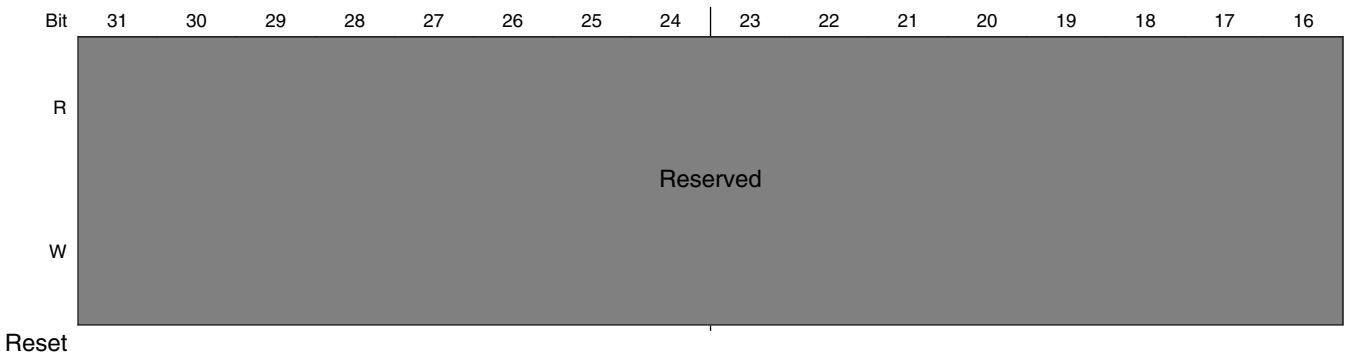
IOMUXC_SW_PAD_CTL_PAD_CLKOUT1 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: CLKOUT1 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: CLKOUT1 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: CLKOUT1 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.299 SW_PAD_CTL_PAD_CLKOUT2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_CLKOUT2)

SW_PAD_CTL Register

Address: 3033_0000h base + 4BCh offset = 3033_04BCh



IOMUXC_SW_PAD_CTL_PAD_CLKOUT2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PE	Pull Select Field Select one out of next values for pad: CLKOUT2 0 **PE_0_PULL_DISABLE** — Pull Disable 1 **PE_1_PULL_ENABLE** — Pull Enable
7 HYS	Input Select Field Select one out of next values for pad: CLKOUT2 0 **HYS_0_CMOS** — CMOS 1 **HYS_1_SCHMITT** — Schmitt
6 PUE	Pull Up / Down Config. Field Select one out of next values for pad: CLKOUT2

Table continues on the next page...

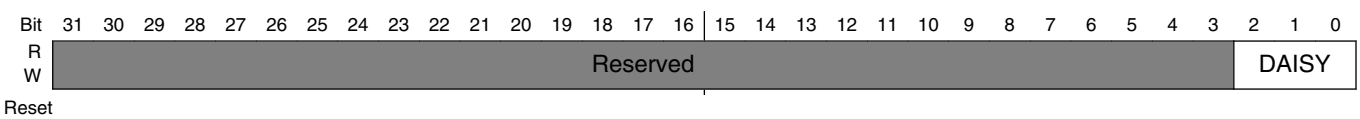
IOMUXC_SW_PAD_CTL_PAD_CLKOUT2 field descriptions (continued)

Field	Description
	0 **PUE_0_WEAK_PULL_DOWN** — Weak pull down 1 **PUE_1_WEAK_PULL_UP** — Weak pull up
5 ODE	Open Drain Field Select one out of next values for pad: CLKOUT2 0 **ODE_0_OPEN_DRAIN_DISABLE** — Open Drain Disable 1 **ODE_1_OPEN_DRAIN_ENABLE** — Open Drain Enable
4 FSEL	Slew Rate Field Select one out of next values for pad: CLKOUT2 0 **FSEL_0_SLOW_SLEW_RATE** — Slow Slew Rate (SR=1) 1 **FSEL_1_FAST_SLEW_RATE** — Fast Slew Rate (SR=0)
3 -	This field is reserved. Reserved
2–1 DSE	Drive Strength Field Select one out of next values for pad: CLKOUT2 00 **DSE_X1** — X1 10 **DSE_X2** — X2 01 **DSE_X4** — X4 11 **DSE_X6** — X6
0 -	This field is reserved. Reserved

8.2.4.300 AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_0 DAISY Register (IOMUXC_AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT

DAISY Register

Address: 3033_0000h base + 4C0h offset = 3033_04C0h



IOMUXC_AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_0 field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain.

Table continues on the next page...

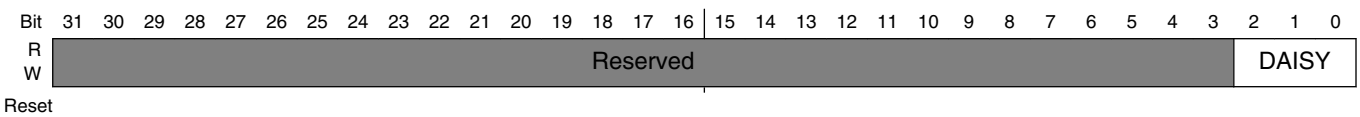
IOMUXC_AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_0 field descriptions (continued)

Field	Description
	Instance: audiomix, In Pin: _pdm_mic_pdm_bitstream[0] 000 **SELECT_ENET_TD1_ALT3** — Selecting Pad: ENET_TD1 for Mode: ALT3 001 **SELECT_ENET_RD1_ALT3** — Selecting Pad: ENET_RD1 for Mode: ALT3 010 **SELECT_SD2_DATA0_ALT4** — Selecting Pad: SD2_DATA0 for Mode: ALT4 011 **SELECT_SAI5_RXD0_ALT4** — Selecting Pad: SAI5_RXD0 for Mode: ALT4 100 **SELECT_SAI1_RXD0_ALT3** — Selecting Pad: SAI1_RXD0 for Mode: ALT3 101 **SELECT_SAI3_RXFS_ALT6** — Selecting Pad: SAI3_RXFS for Mode: ALT6

8.2.4.301 AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_1 DAISY Register (IOMUXC_AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT

DAISY Register

Address: 3033_0000h base + 4C4h offset = 3033_04C4h



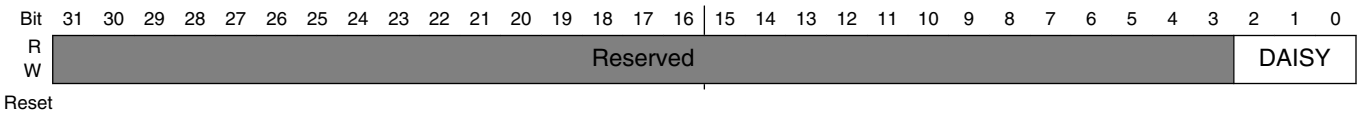
IOMUXC_AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_1 field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _pdm_mic_pdm_bitstream[1] 000 **SELECT_ENET_TD2_ALT3** — Selecting Pad: ENET_TD2 for Mode: ALT3 001 **SELECT_ENET_RD0_ALT3** — Selecting Pad: ENET_RD0 for Mode: ALT3 010 **SELECT_SD2_DATA1_ALT4** — Selecting Pad: SD2_DATA1 for Mode: ALT4 011 **SELECT_SAI5_RXD1_ALT4** — Selecting Pad: SAI5_RXD1 for Mode: ALT4 100 **SELECT_SAI1_RXD1_ALT3** — Selecting Pad: SAI1_RXD1 for Mode: ALT3 101 **SELECT_SAI2_RXC_ALT6** — Selecting Pad: SAI2_RXC for Mode: ALT6 110 **SELECT_SAI2_TXC_ALT6** — Selecting Pad: SAI2_TXC for Mode: ALT6 111 **SELECT_SAI3_RXD_ALT6** — Selecting Pad: SAI3_RXD for Mode: ALT6

8.2.4.302 AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_2 DAISY Register (IOMUXC_AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT

DAISY Register

Address: 3033_0000h base + 4C8h offset = 3033_04C8h



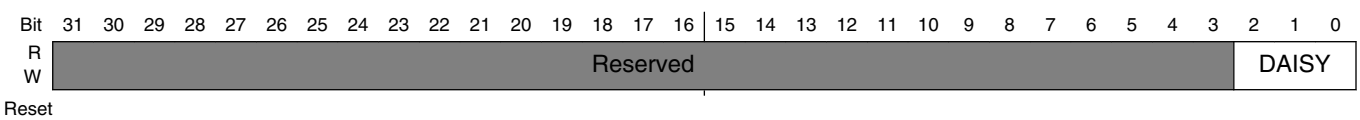
IOMUXC_AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_2 field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _pdm_mic_pdm_bitstream[2] 000 **SELECT_ENET_TD3_ALT3** — Selecting Pad: ENET_TD3 for Mode: ALT3 001 **SELECT_ENET_RXC_ALT3** — Selecting Pad: ENET_RXC for Mode: ALT3 010 **SELECT_SD2_DATA2_ALT4** — Selecting Pad: SD2_DATA2 for Mode: ALT4 011 **SELECT_SAI5_RXD2_ALT4** — Selecting Pad: SAI5_RXD2 for Mode: ALT4 100 **SELECT_SAI1_RXD2_ALT3** — Selecting Pad: SAI1_RXD2 for Mode: ALT3 101 **SELECT_SAI2_RXFS_ALT6** — Selecting Pad: SAI2_RXFS for Mode: ALT6 110 **SELECT_SAI2_TXFS_ALT6** — Selecting Pad: SAI2_TXFS for Mode: ALT6 111 **SELECT_SAI3_TXC_ALT6** — Selecting Pad: SAI3_TXC for Mode: ALT6

8.2.4.303 AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_3 DAISY Register (IOMUXC_AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT

DAISY Register

Address: 3033_0000h base + 4CCh offset = 3033_04CCh



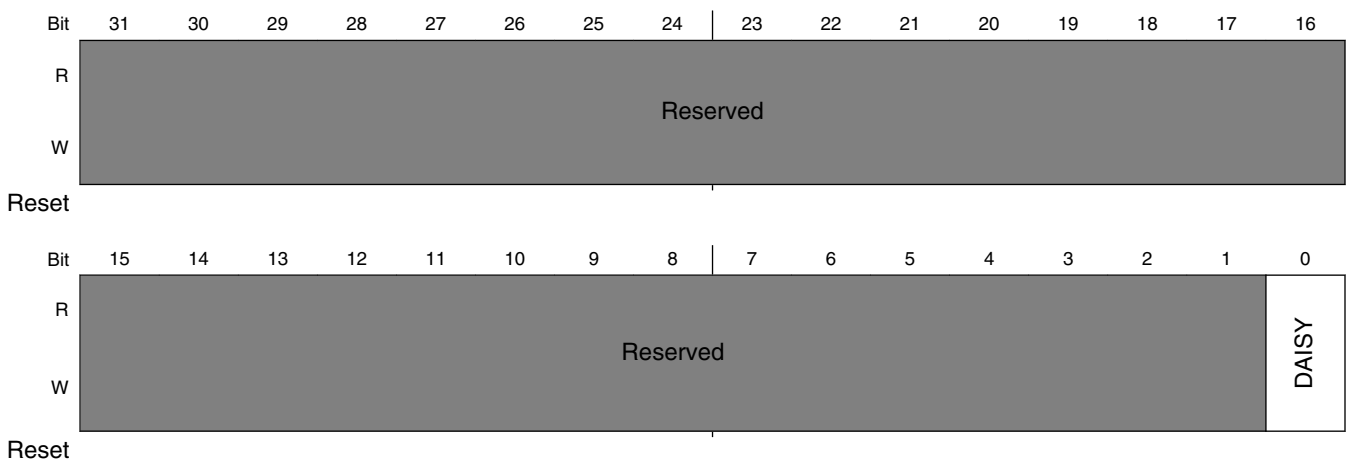
IOMUXC_AUDIOMIX_PDM_MIC_PDM_BITSTREAM_SELECT_INPUT_3 field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _pdm_mic_pdm_bitstream[3] 000 **SELECT_ENET_MDIO_ALT3** — Selecting Pad: ENET_MDIO for Mode: ALT3 001 **SELECT_ENET_RX_CTL_ALT3** — Selecting Pad: ENET_RX_CTL for Mode: ALT3 010 **SELECT_SD2_DATA3_ALT4** — Selecting Pad: SD2_DATA3 for Mode: ALT4 011 **SELECT_SAI5_RXD3_ALT4** — Selecting Pad: SAI5_RXD3 for Mode: ALT4 100 **SELECT_SAI1_RXD3_ALT3** — Selecting Pad: SAI1_RXD3 for Mode: ALT3 101 **SELECT_SAI2_RXD0_ALT6** — Selecting Pad: SAI2_RXD0 for Mode: ALT6 110 **SELECT_SAI3_TXFS_ALT6** — Selecting Pad: SAI3_TXFS for Mode: ALT6

8.2.4.304 AUDIOMIX_SAI1_RXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI1_RXSYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4D0h offset = 3033_04D0h



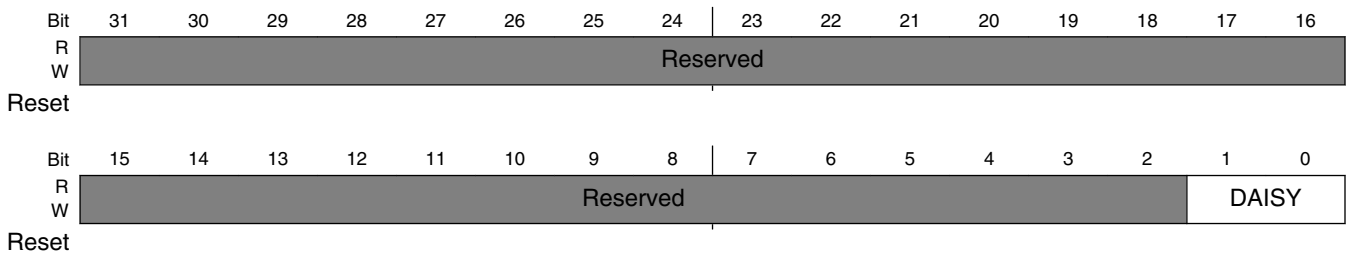
IOMUXC_AUDIOMIX_SAI1_RXSYNC_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai1_rxsync 0 **SELECT_SAI1_RXFS_ALT0** — Selecting Pad: SAI1_RXFS for Mode: ALT0 1 **SELECT_SAI1_RXD5_ALT3** — Selecting Pad: SAI1_RXD5 for Mode: ALT3

8.2.4.305 AUDIOMIX_SAI1_TXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI1_TXBCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4D4h offset = 3033_04D4h



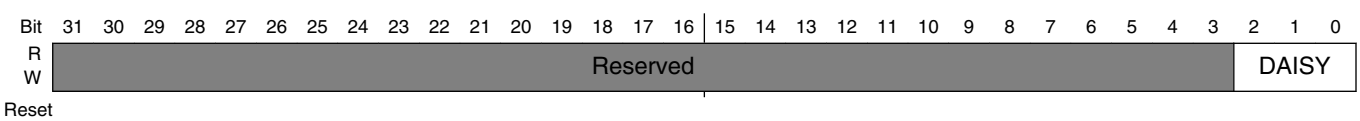
IOMUXC_AUDIOMIX_SAI1_TXBCLK_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai1_txbclk 00 **SELECT_SAI5_MCLK_ALT1** — Selecting Pad: SAI5_MCLK for Mode: ALT1 01 **SELECT_SAI1_TXC_ALT0** — Selecting Pad: SAI1_TXC for Mode: ALT0 10 **SELECT_SAI1_MCLK_ALT2** — Selecting Pad: SAI1_MCLK for Mode: ALT2

8.2.4.306 AUDIOMIX_SAI1_TXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI1_TXSYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4D8h offset = 3033_04D8h



IOMUXC_AUDIOMIX_SAI1_TXSYNC_SELECT_INPUT field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai1_txsync

Table continues on the next page...

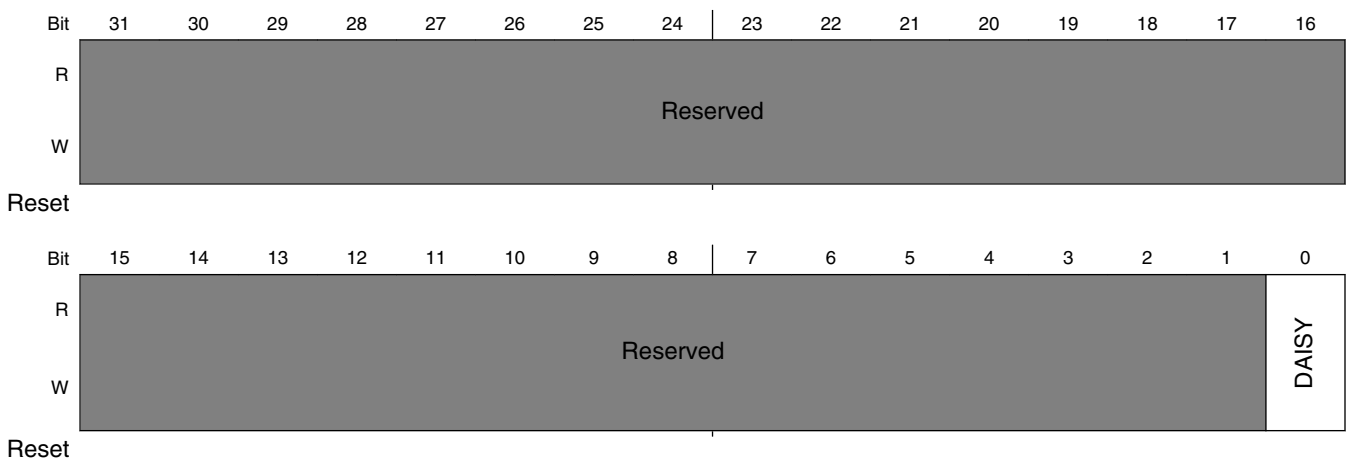
IOMUXC_AUDIOMIX_SAI1_TXSYNC_SELECT_INPUT field descriptions (continued)

Field	Description
	000 **SELECT_SAI5_RXD1_ALT2** — Selecting Pad: SAI5_RXD1 for Mode: ALT2 001 **SELECT_SAI5_RXD2_ALT2** — Selecting Pad: SAI5_RXD2 for Mode: ALT2 010 **SELECT_SAI5_RXD3_ALT2** — Selecting Pad: SAI5_RXD3 for Mode: ALT2 011 **SELECT_SAI1_RXD7_ALT2** — Selecting Pad: SAI1_RXD7 for Mode: ALT2 100 **SELECT_SAI1_TXFS_ALT0** — Selecting Pad: SAI1_TXFS for Mode: ALT0

8.2.4.307 AUDIOMIX_SAI2_RXDATA_SELECT_INPUT_1 DAISY Register (IOMUXC_AUDIOMIX_SAI2_RXDATA_SELECT_INPUT_1)

DAISY Register

Address: 3033_0000h base + 4DCh offset = 3033_04DCh



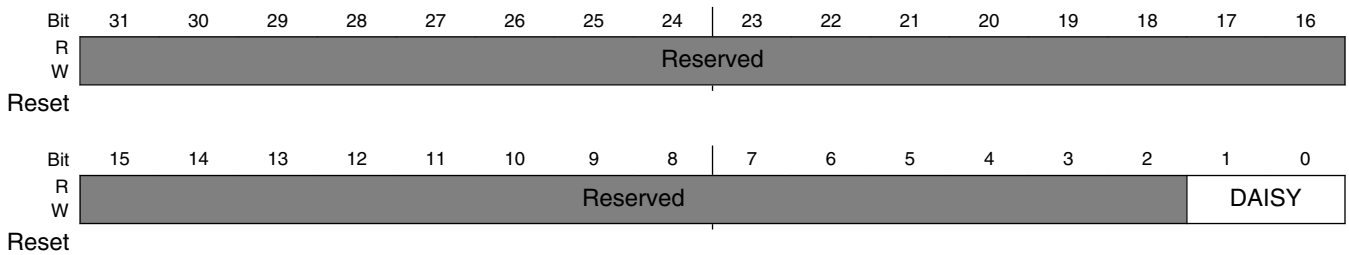
IOMUXC_AUDIOMIX_SAI2_RXDATA_SELECT_INPUT_1 field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai2_rxdata[1] 0 **SELECT_SAI2_RXFS_ALT3** — Selecting Pad: SAI2_RXFS for Mode: ALT3 1 **SELECT_SAI3_RXFS_ALT1** — Selecting Pad: SAI3_RXFS for Mode: ALT1

8.2.4.308 AUDIOMIX_SAI3_MCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI3_MCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4E0h offset = 3033_04E0h



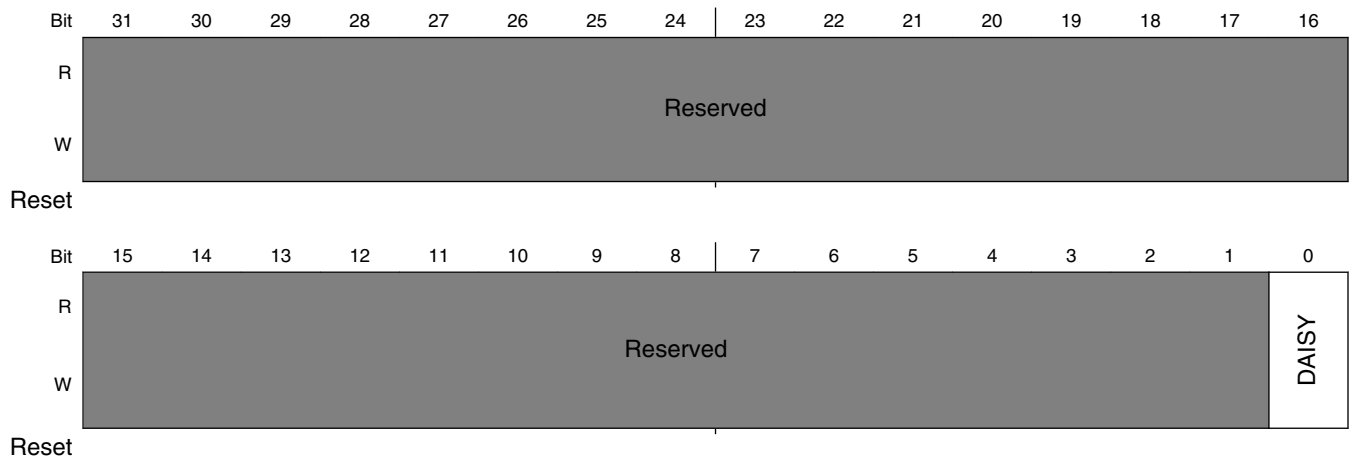
IOMUXC_AUDIOMIX_SAI3_MCLK_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai3_mclk 00 **SELECT_NAND_DQS_ALT2** — Selecting Pad: NAND_DQS for Mode: ALT2 01 **SELECT_SAI2_MCLK_ALT6** — Selecting Pad: SAI2_MCLK for Mode: ALT6 10 **SELECT_SAI3_MCLK_ALT0** — Selecting Pad: SAI3_MCLK for Mode: ALT0

8.2.4.309 AUDIOMIX_SAI3_RXDATA_SELECT_INPUT_0 DAISY Register (IOMUXC_AUDIOMIX_SAI3_RXDATA_SELECT_INPUT_0)

DAISY Register

Address: 3033_0000h base + 4E4h offset = 3033_04E4h



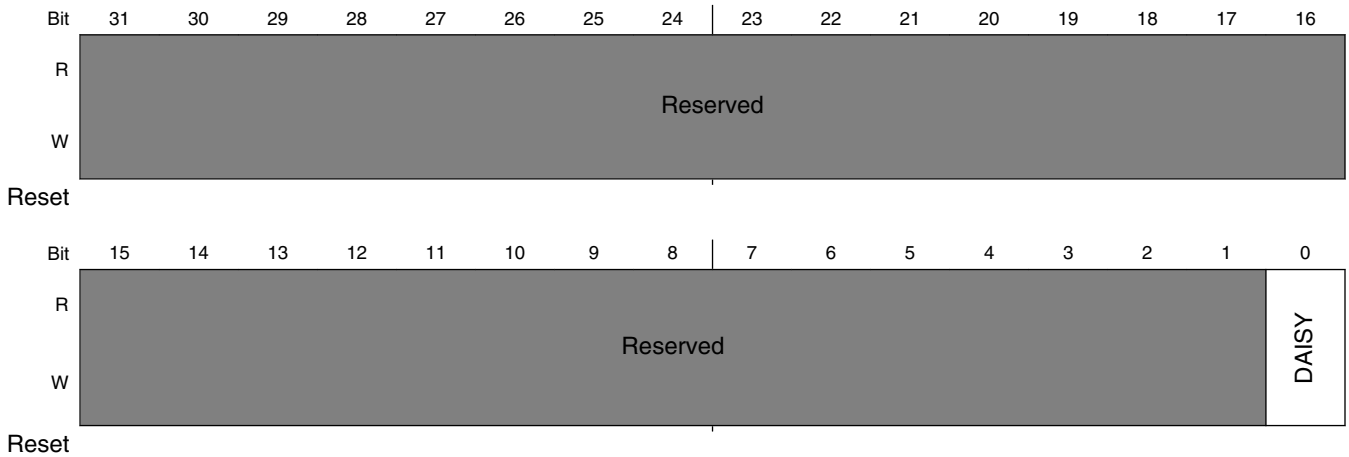
IOMUXC_AUDIOMIX_SAI3_RXDATA_SELECT_INPUT_0 field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai3_rxdata[0] 0 **SELECT_NAND_DATA00_ALT2** — Selecting Pad: NAND_DATA00 for Mode: ALT2 1 **SELECT_SAI3_RXD_ALT0** — Selecting Pad: SAI3_RXD for Mode: ALT0

8.2.4.310 AUDIOMIX_SAI3_TXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI3_TXBCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4E8h offset = 3033_04E8h



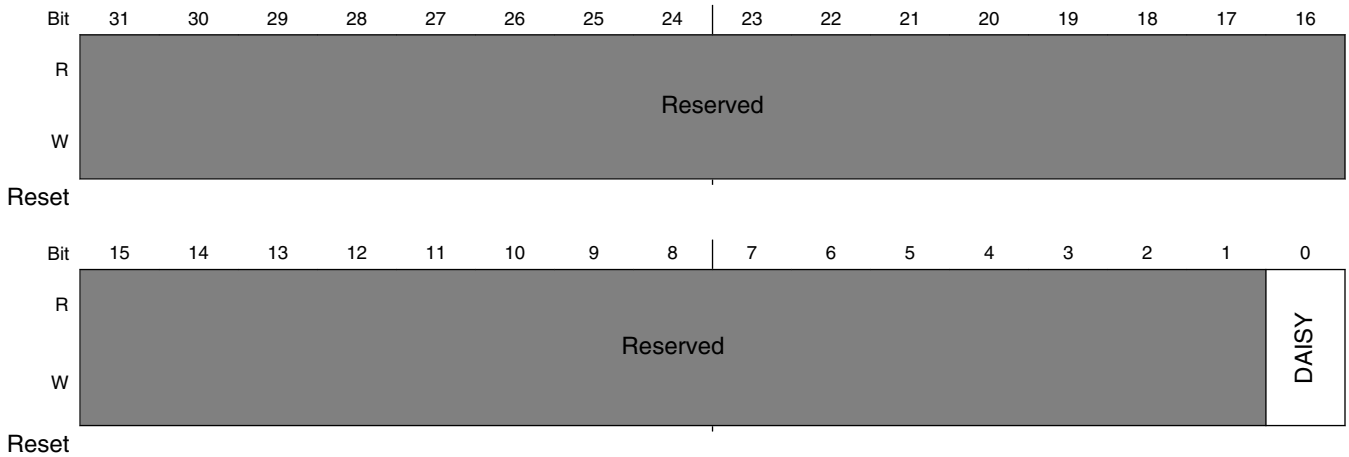
IOMUXC_AUDIOMIX_SAI3_TXBCLK_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai3_txbclk 0 **SELECT_NAND_ALE_ALT2** — Selecting Pad: NAND_ALE for Mode: ALT2 1 **SELECT_SAI3_TXC_ALT0** — Selecting Pad: SAI3_TXC for Mode: ALT0

8.2.4.311 AUDIOMIX_SAI3_TXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI3_TXSYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4ECh offset = 3033_04ECh



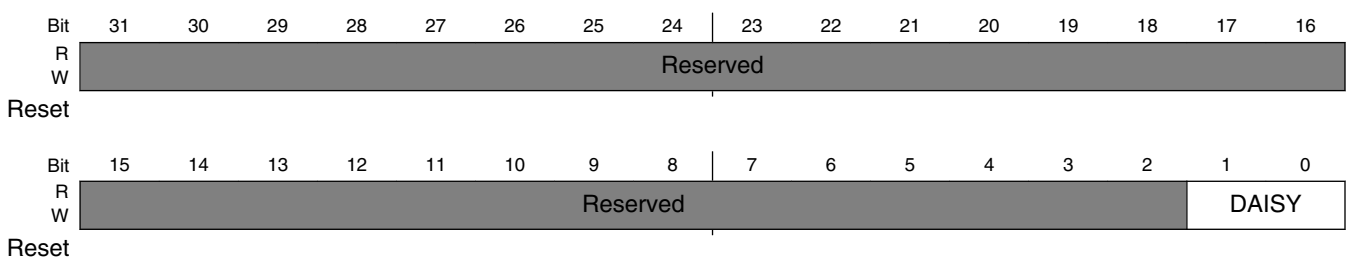
IOMUXC_AUDIOMIX_SAI3_TXSYNC_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai3_txsync 0 **SELECT_NAND_DATA01_ALT2** — Selecting Pad: NAND_DATA01 for Mode: ALT2 1 **SELECT_SAI3_TXFS_ALT0** — Selecting Pad: SAI3_TXFS for Mode: ALT0

8.2.4.312 AUDIOMIX_SAI5_MCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI5_MCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4F0h offset = 3033_04F0h



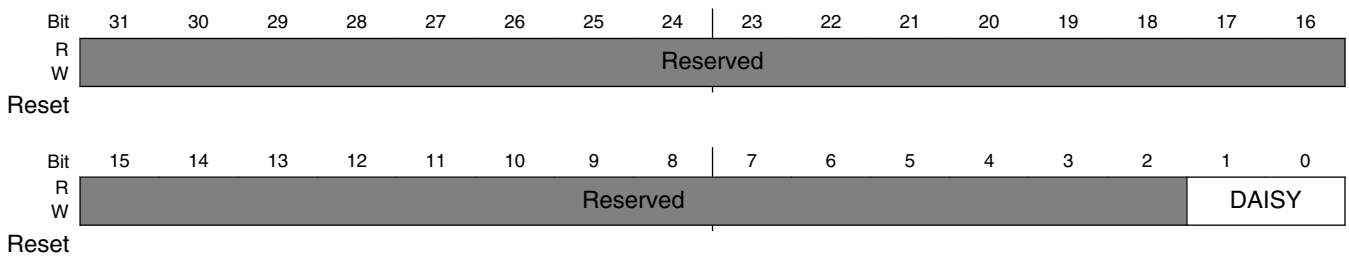
IOMUXC_AUDIOMIX_SAI5_MCLK_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai5_mclk 00 **SELECT_SAI5_MCLK_ALT0** — Selecting Pad: SAI5_MCLK for Mode: ALT0 10 **SELECT_SAI2_MCLK_ALT1** — Selecting Pad: SAI2_MCLK for Mode: ALT1 11 **SELECT_SAI3_MCLK_ALT2** — Selecting Pad: SAI3_MCLK for Mode: ALT2

8.2.4.313 AUDIOMIX_SAI5_RXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI5_RXBCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4F4h offset = 3033_04F4h



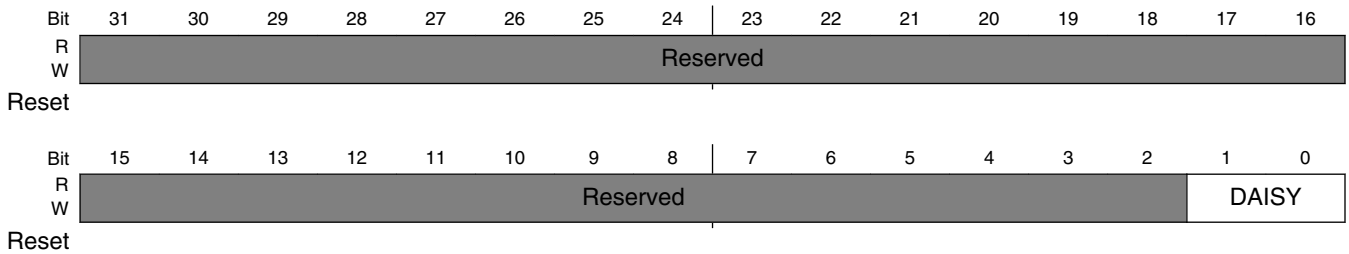
IOMUXC_AUDIOMIX_SAI5_RXBCLK_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai5_rxbclk 00 **SELECT_SAI5_RXC_ALT0** — Selecting Pad: SAI5_RXC for Mode: ALT0 10 **SELECT_SAI3_RXC_ALT2** — Selecting Pad: SAI3_RXC for Mode: ALT2

8.2.4.314 AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_0 DAISY Register (IOMUXC_AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_0)

DAISY Register

Address: 3033_0000h base + 4F8h offset = 3033_04F8h



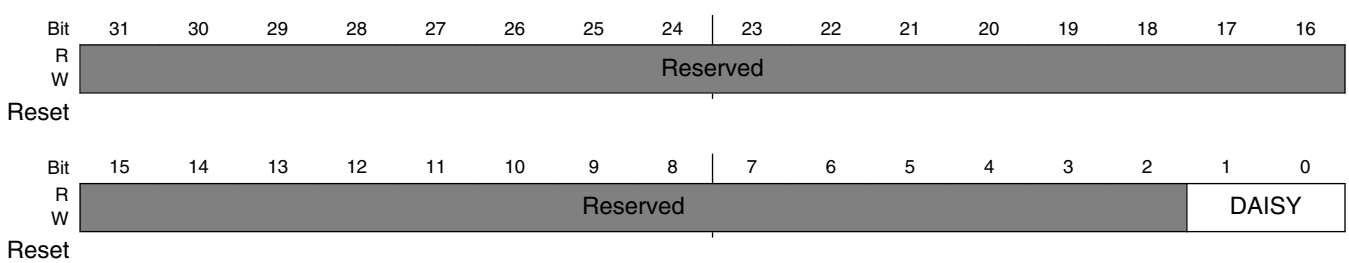
IOMUXC_AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_0 field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai5_rxdata[0] 00 **SELECT_SAI5_RXD0_ALT0** — Selecting Pad: SAI5_RXD0 for Mode: ALT0 10 **SELECT_SAI3_RXD_ALT2** — Selecting Pad: SAI3_RXD for Mode: ALT2

8.2.4.315 AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_1 DAISY Register (IOMUXC_AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_1)

DAISY Register

Address: 3033_0000h base + 4FCh offset = 3033_04FCh



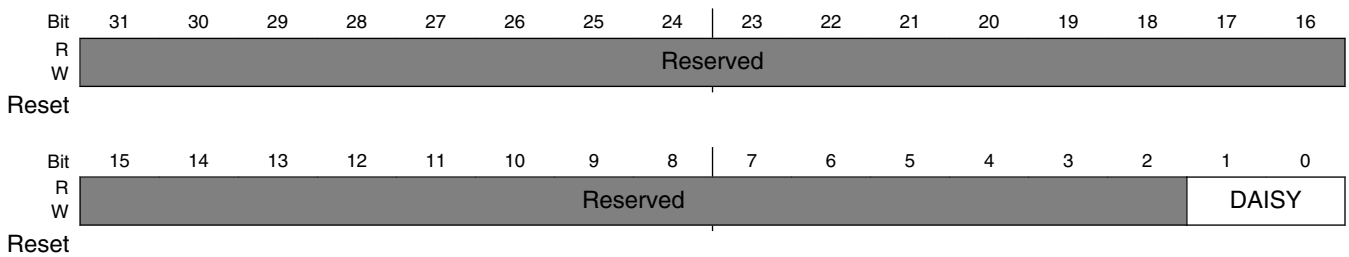
IOMUXC_AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_1 field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai5_rxdata[1] 00 **SELECT_SAI5_RXD1_ALT0** — Selecting Pad: SAI5_RXD1 for Mode: ALT0 10 **SELECT_SAI3_TXFS_ALT2** — Selecting Pad: SAI3_TXFS for Mode: ALT2

8.2.4.316 AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_2 DAISY Register (IOMUXC_AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_2)

DAISY Register

Address: 3033_0000h base + 500h offset = 3033_0500h



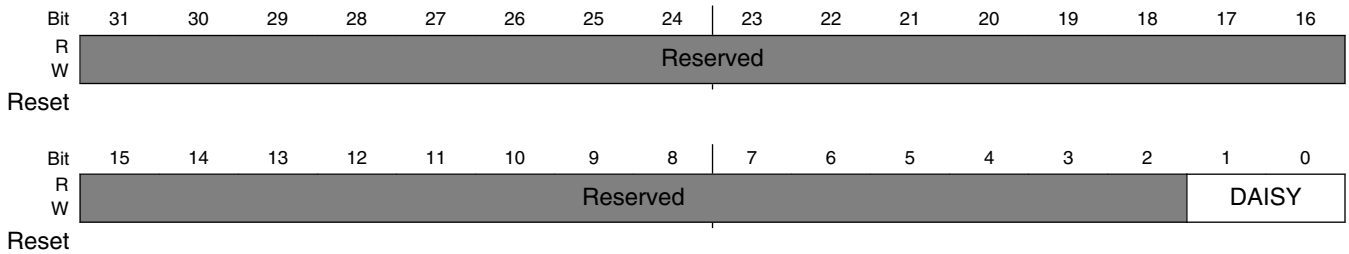
IOMUXC_AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_2 field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai5_rxdata[2] 00 **SELECT_SAI5_RXD2_ALT0** — Selecting Pad: SAI5_RXD2 for Mode: ALT0 10 **SELECT_SAI3_TXC_ALT2** — Selecting Pad: SAI3_TXC for Mode: ALT2

8.2.4.317 AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_3 DAISY Register (IOMUXC_AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_3)

DAISY Register

Address: 3033_0000h base + 504h offset = 3033_0504h



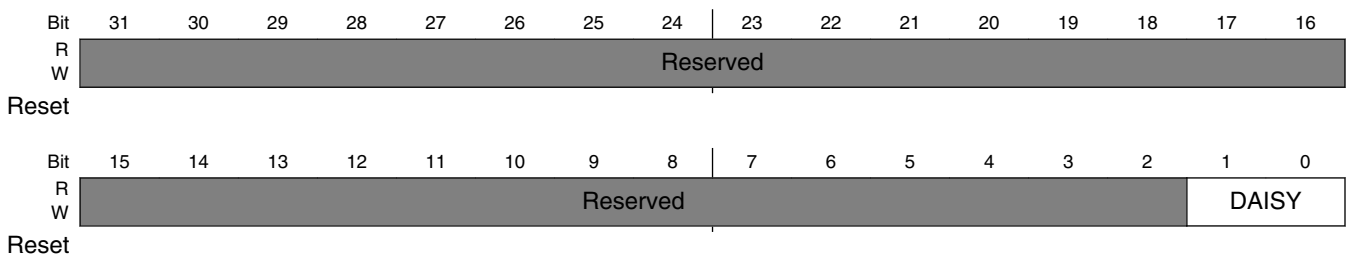
IOMUXC_AUDIOMIX_SAI5_RXDATA_SELECT_INPUT_3 field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai5_rxdata[3] 00 **SELECT_SAI5_RXD3_ALT0** — Selecting Pad: SAI5_RXD3 for Mode: ALT0 10 **SELECT_SAI3_TXD_ALT2** — Selecting Pad: SAI3_TXD for Mode: ALT2

8.2.4.318 AUDIOMIX_SAI5_RXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI5_RXSYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 508h offset = 3033_0508h



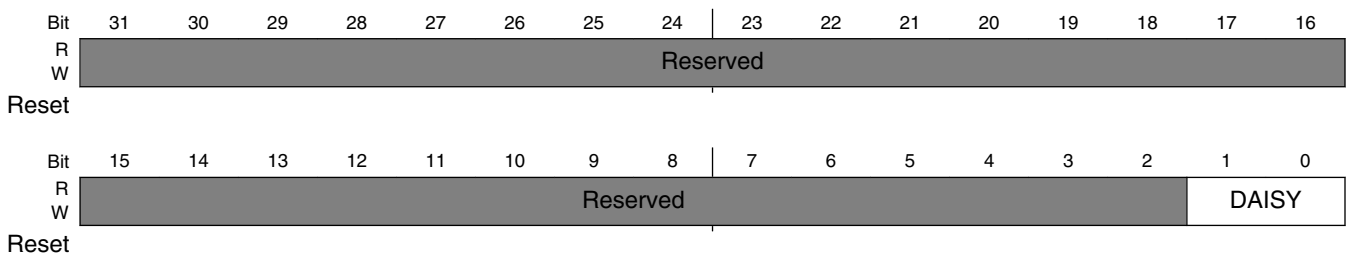
IOMUXC_AUDIOMIX_SAI5_RXSYNC_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai5_rxsync 00 **SELECT_SAI5_RXFS_ALT0** — Selecting Pad: SAI5_RXFS for Mode: ALT0 10 **SELECT_SAI3_RXFS_ALT2** — Selecting Pad: SAI3_RXFS for Mode: ALT2

8.2.4.319 AUDIOMIX_SAI5_TXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI5_TXBCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 50Ch offset = 3033_050Ch



IOMUXC_AUDIOMIX_SAI5_TXBCLK_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai5_txbclk 00 **SELECT_SAI5_RXD2_ALT3** — Selecting Pad: SAI5_RXD2 for Mode: ALT3 10 **SELECT_SAI2_RXC_ALT1** — Selecting Pad: SAI2_RXC for Mode: ALT1

8.2.4.320 AUDIOMIX_SAI5_TXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI5_TXSYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 510h offset = 3033_0510h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset																

IOMUXC_AUDIOMIX_SAI5_TXSYNC_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai5_txsync 00 **SELECT_SAI5_RXD1_ALT3** — Selecting Pad: SAI5_RXD1 for Mode: ALT3 10 **SELECT_SAI2_RXFS_ALT1** — Selecting Pad: SAI2_RXFS for Mode: ALT1

8.2.4.321 AUDIOMIX_SAI6_MCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI6_MCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 514h offset = 3033_0514h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset																

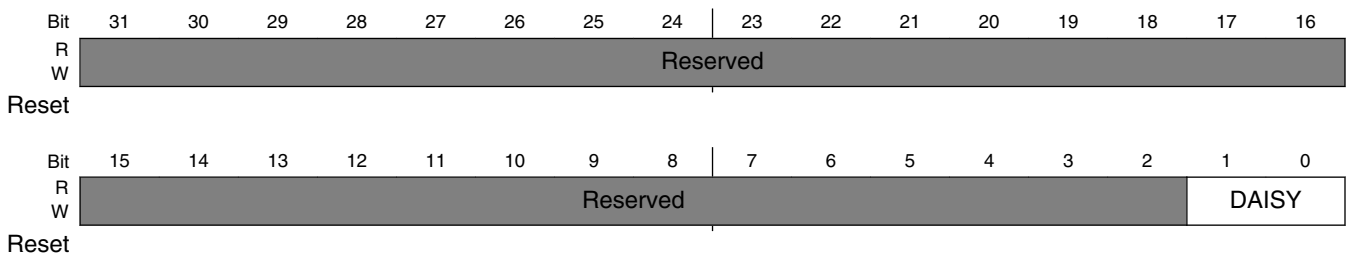
IOMUXC_AUDIOMIX_SAI6_MCLK_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai6_mclk 00 **SELECT_ENET_TX_CTL_ALT2** — Selecting Pad: ENET_TX_CTL for Mode: ALT2 01 **SELECT_SAI1_RXD7_ALT1** — Selecting Pad: SAI1_RXD7 for Mode: ALT1 10 **SELECT_SAI1_TXD7_ALT1** — Selecting Pad: SAI1_TXD7 for Mode: ALT1

8.2.4.322 AUDIOMIX_SAI6_RXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI6_RXBCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 518h offset = 3033_0518h



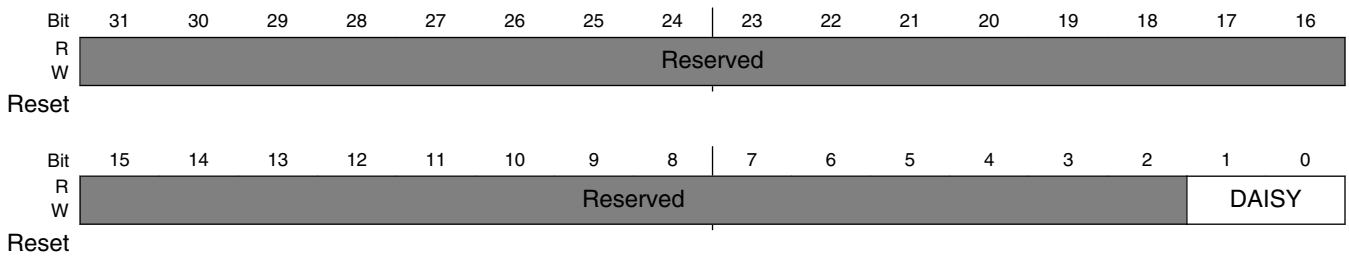
IOMUXC_AUDIOMIX_SAI6_RXBCLK_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai6_rxbclk 00 **SELECT_ENET_TD0_ALT2** — Selecting Pad: ENET_TD0 for Mode: ALT2 01 **SELECT_SAI1_RXD4_ALT2** — Selecting Pad: SAI1_RXD4 for Mode: ALT2 10 **SELECT_SAI1_TXD4_ALT1** — Selecting Pad: SAI1_TXD4 for Mode: ALT1

8.2.4.323 AUDIOMIX_SAI6_RXDATA_SELECT_INPUT_0 DAISY Register (IOMUXC_AUDIOMIX_SAI6_RXDATA_SELECT_INPUT_0)

DAISY Register

Address: 3033_0000h base + 51Ch offset = 3033_051Ch



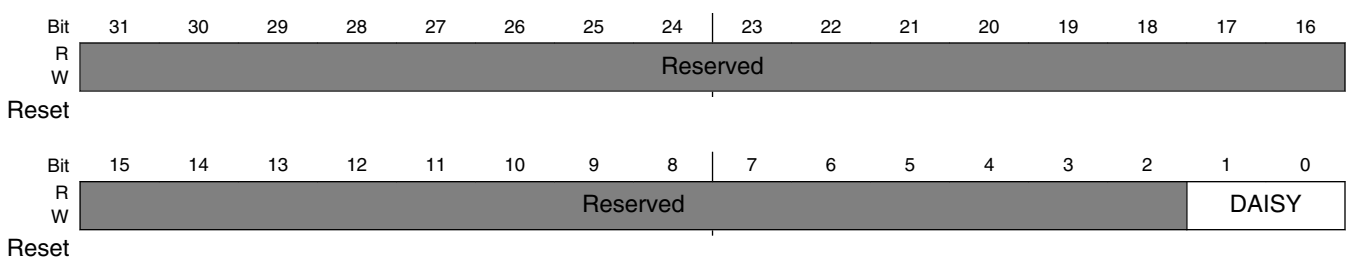
IOMUXC_AUDIOMIX_SAI6_RXDATA_SELECT_INPUT_0 field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai6_rxdata[0] 00 **SELECT_ENET_TD2_ALT2** — Selecting Pad: ENET_TD2 for Mode: ALT2 01 **SELECT_SAI1_RXD5_ALT2** — Selecting Pad: SAI1_RXD5 for Mode: ALT2 10 **SELECT_SAI1_TXD5_ALT1** — Selecting Pad: SAI1_TXD5 for Mode: ALT1

8.2.4.324 AUDIOMIX_SAI6_RXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI6_RXSYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 520h offset = 3033_0520h



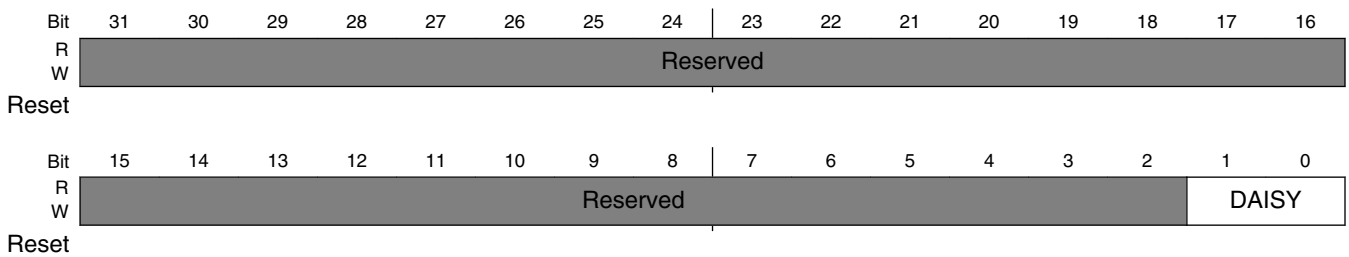
IOMUXC_AUDIOMIX_SAI6_RXSYNC_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai6_rxsync 00 **SELECT_ENET_TD1_ALT2** — Selecting Pad: ENET_TD1 for Mode: ALT2 01 **SELECT_SAI1_RXD6_ALT2** — Selecting Pad: SAI1_RXD6 for Mode: ALT2 10 **SELECT_SAI1_TXD6_ALT1** — Selecting Pad: SAI1_TXD6 for Mode: ALT1

8.2.4.325 AUDIOMIX_SAI6_TXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI6_TXBCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 524h offset = 3033_0524h



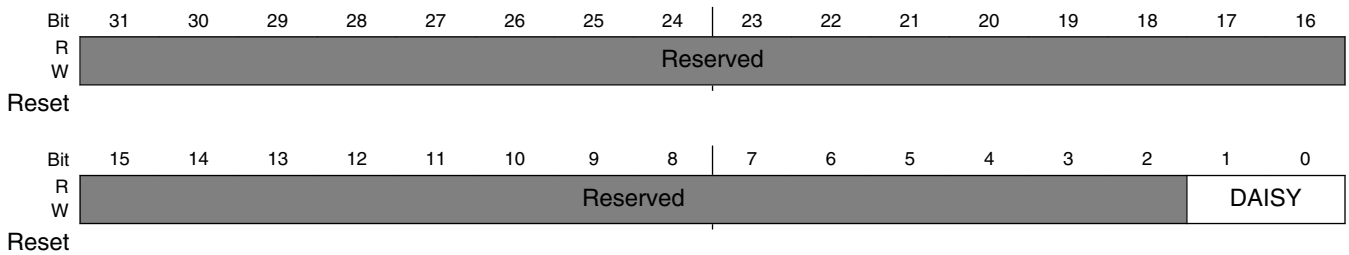
IOMUXC_AUDIOMIX_SAI6_TXBCLK_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai6_txbclk 00 **SELECT_ENET_TD3_ALT2** — Selecting Pad: ENET_TD3 for Mode: ALT2 01 **SELECT_SAI1_RXD4_ALT1** — Selecting Pad: SAI1_RXD4 for Mode: ALT1 10 **SELECT_SAI1_TXD4_ALT2** — Selecting Pad: SAI1_TXD4 for Mode: ALT2

8.2.4.326 AUDIOMIX_SAI6_TXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI6_TXSYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 528h offset = 3033_0528h



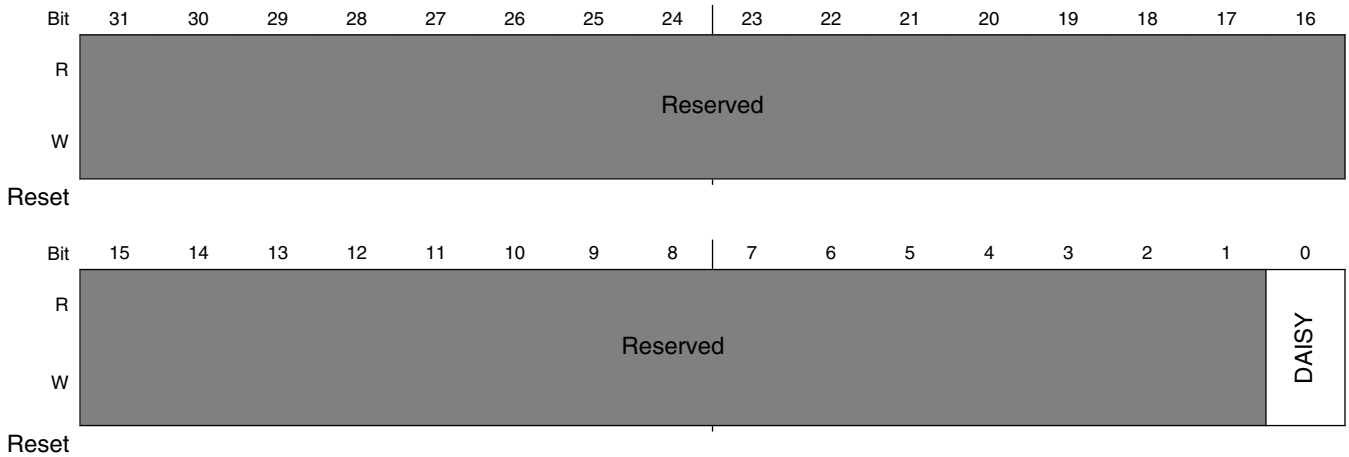
IOMUXC_AUDIOMIX_SAI6_TXSYNC_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai6_txsync 00 **SELECT_ENET_MDIO_ALT2** — Selecting Pad: ENET_MDIO for Mode: ALT2 01 **SELECT_SAI1_RXD6_ALT1** — Selecting Pad: SAI1_RXD6 for Mode: ALT1 10 **SELECT_SAI1_TXD6_ALT2** — Selecting Pad: SAI1_TXD6 for Mode: ALT2

8.2.4.327 AUDIOMIX_SAI7_MCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI7_MCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 52Ch offset = 3033_052Ch



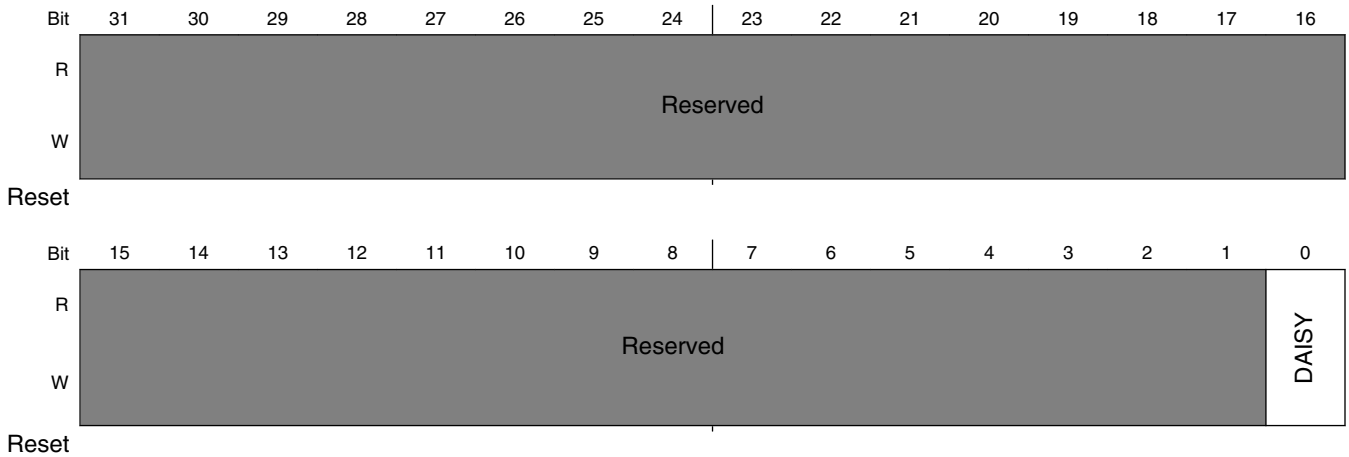
IOMUXC_AUDIOMIX_SAI7_MCLK_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai7_mclk 0 **SELECT_ENET_RD3_ALT2** — Selecting Pad: ENET_RD3 for Mode: ALT2 1 **SELECT_ECSPI2_MISO_ALT3** — Selecting Pad: ECSPI2_MISO for Mode: ALT3

8.2.4.328 AUDIOMIX_SAI7_RXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI7_RXBCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 530h offset = 3033_0530h



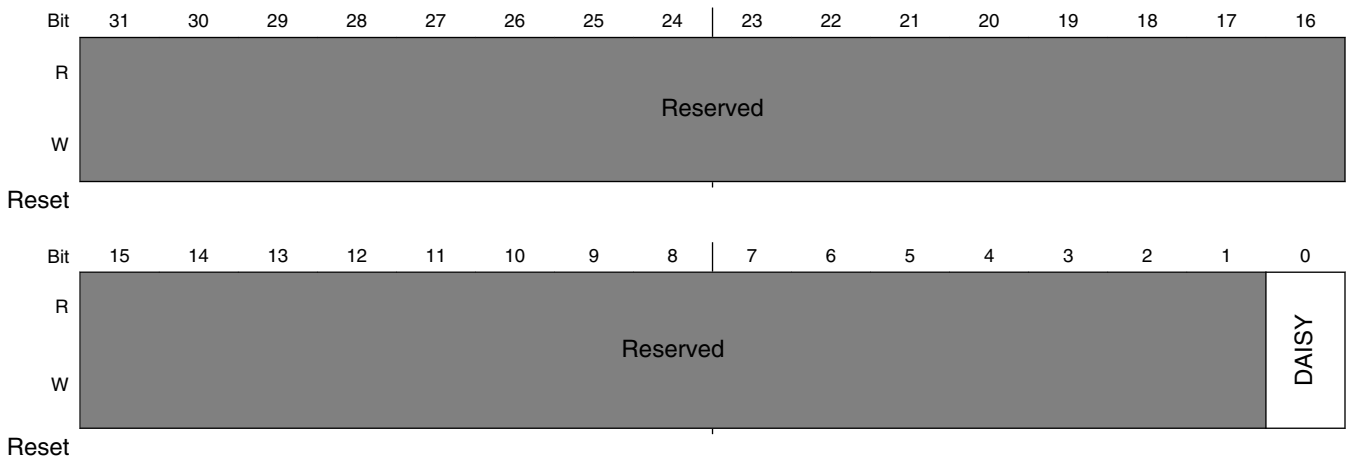
IOMUXC_AUDIOMIX_SAI7_RXBCLK_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai7_rxbclk 0 **SELECT_ENET_RD2_ALT2** — Selecting Pad: ENET_RD2 for Mode: ALT2 1 **SELECT_ECSPI1_MOSI_ALT3** — Selecting Pad: ECSPI1_MOSI for Mode: ALT3

8.2.4.329 AUDIOMIX_SAI7_RXDATA_SELECT_INPUT_0 DAISY Register (IOMUXC_AUDIOMIX_SAI7_RXDATA_SELECT_INPUT_0)

DAISY Register

Address: 3033_0000h base + 534h offset = 3033_0534h



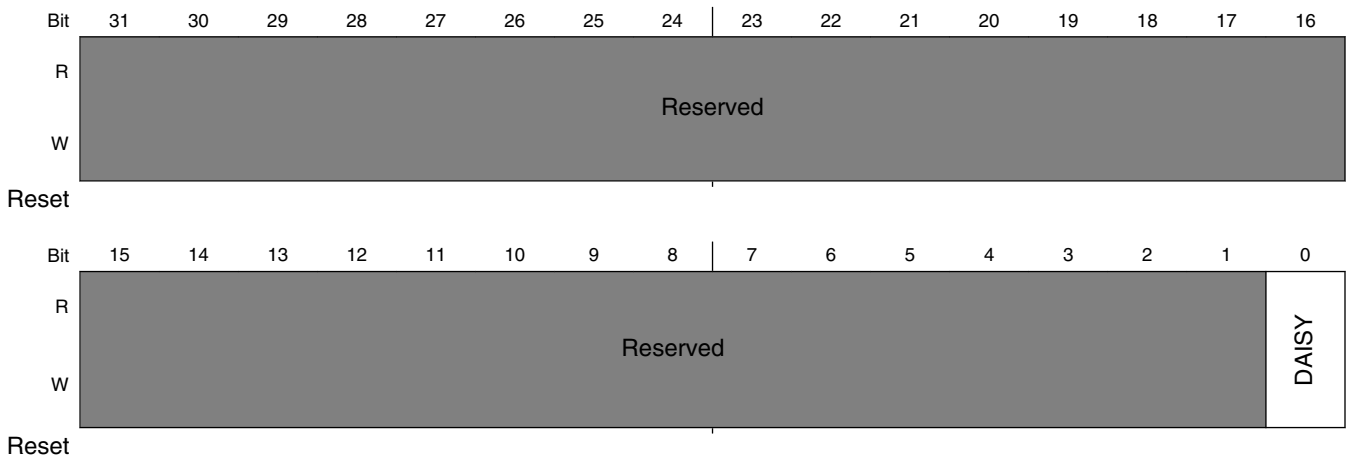
IOMUXC_AUDIOMIX_SAI7_RXDATA_SELECT_INPUT_0 field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai7_rxdata[0] 0 **SELECT_ENET_RD0_ALT2** — Selecting Pad: ENET_RD0 for Mode: ALT2 1 **SELECT_ECSPI1_MISO_ALT3** — Selecting Pad: ECSPI1_MISO for Mode: ALT3

8.2.4.330 AUDIOMIX_SAI7_RXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI7_RXSYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 538h offset = 3033_0538h



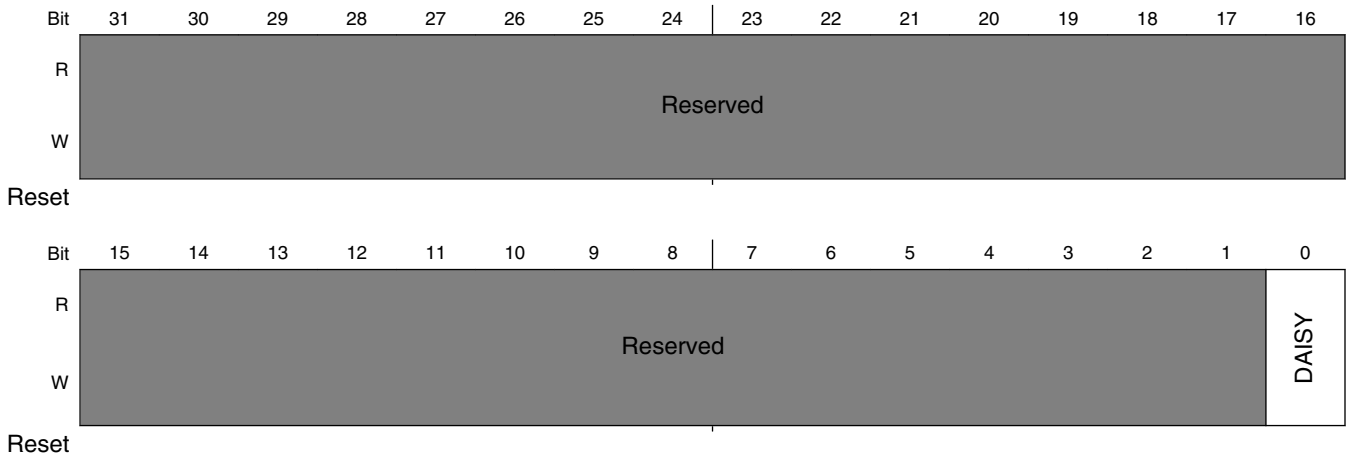
IOMUXC_AUDIOMIX_SAI7_RXSYNC_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai7_rxsync 0 **SELECT_ENET_RD1_ALT2** — Selecting Pad: ENET_RD1 for Mode: ALT2 1 **SELECT_ECSPI1_SCLK_ALT3** — Selecting Pad: ECSPI1_SCLK for Mode: ALT3

8.2.4.331 AUDIOMIX_SAI7_TXBCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI7_TXBCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 53Ch offset = 3033_053Ch



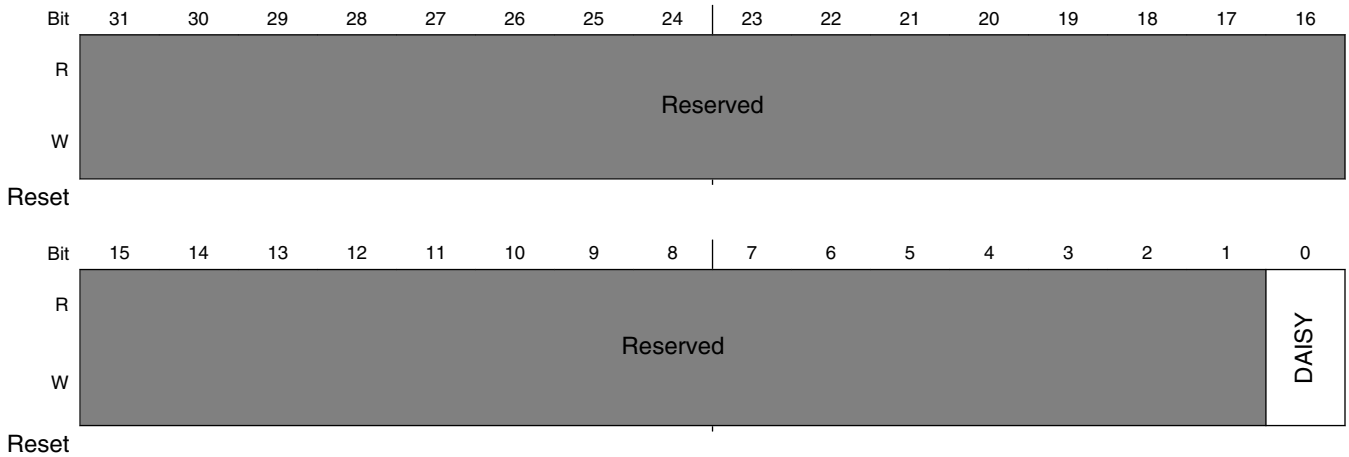
IOMUXC_AUDIOMIX_SAI7_TXBCLK_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai7_txbclk 0 **SELECT_ENET_RXC_ALT2** — Selecting Pad: ENET_RXC for Mode: ALT2 1 **SELECT_ECSPI2_SCLK_ALT3** — Selecting Pad: ECSPI2_SCLK for Mode: ALT3

8.2.4.332 AUDIOMIX_SAI7_TXSYNC_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SAI7_TXSYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 540h offset = 3033_0540h



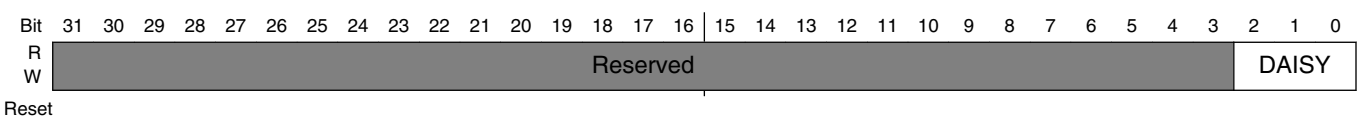
IOMUXC_AUDIOMIX_SAI7_TXSYNC_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: _sai7_txsync 0 **SELECT_ENET_RX_CTL_ALT2** — Selecting Pad: ENET_RX_CTL for Mode: ALT2 1 **SELECT_ECSPI1_SS0_ALT3** — Selecting Pad: ECSPI1_SS0 for Mode: ALT3

8.2.4.333 AUDIOMIX_EARC_PHY_SPDIF_IN_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_EARC_PHY_SPDIF_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 544h offset = 3033_0544h



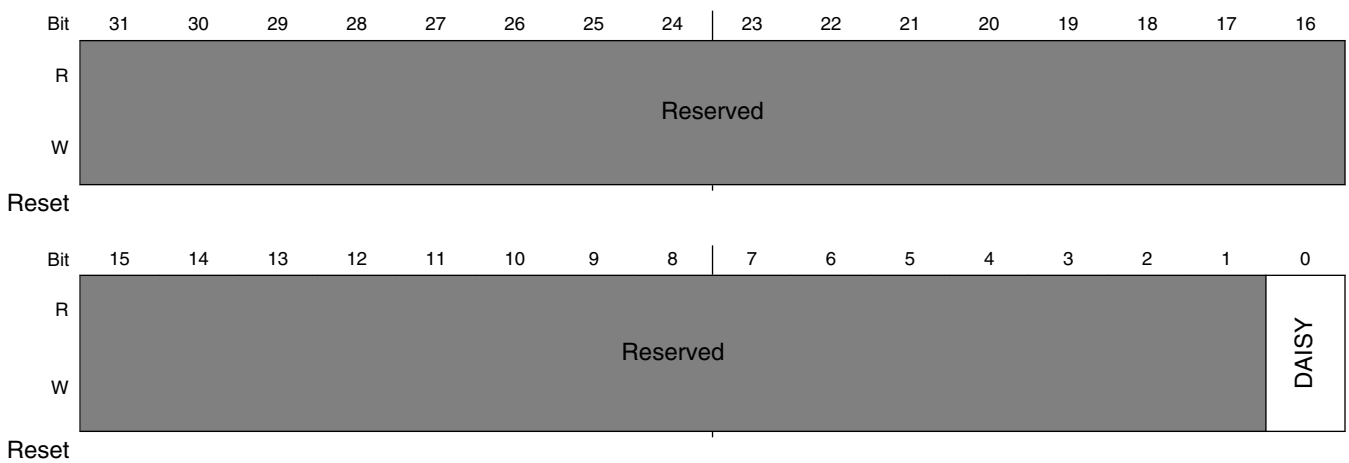
IOMUXC_AUDIOMIX_EARC_PHY_SPDIF_IN_SELECT_INPUT field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: earc_phy_spdif_in 000 **SELECT_ENET_RD3_ALT3** — Selecting Pad: ENET_RD3 for Mode: ALT3 001 **SELECT_SD2_DATA3_ALT3** — Selecting Pad: SD2_DATA3 for Mode: ALT3 010 **SELECT_SAI3_RXFS_ALT4** — Selecting Pad: SAI3_RXFS for Mode: ALT4 011 **SELECT_SAI3_MCLK_ALT6** — Selecting Pad: SAI3_MCLK for Mode: ALT6 100 **SELECT_SPDIF_RX_ALT0** — Selecting Pad: SPDIF_RX for Mode: ALT0

8.2.4.334 AUDIOMIX_SPDIF_EXTCLK_SELECT_INPUT DAISY Register (IOMUXC_AUDIOMIX_SPDIF_EXTCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 548h offset = 3033_0548h



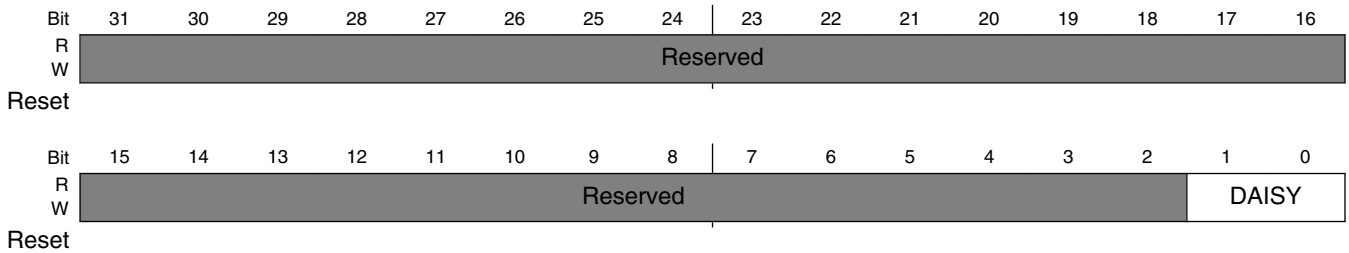
IOMUXC_AUDIOMIX_SPDIF_EXTCLK_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audiomix, In Pin: spdif_extclk 0 **SELECT_SAI3_TXD_ALT4** — Selecting Pad: SAI3_TXD for Mode: ALT4 1 **SELECT_SPDIF_EXT_CLK_ALT0** — Selecting Pad: SPDIF_EXT_CLK for Mode: ALT0

8.2.4.335 CAN1_CANRX_SELECT_INPUT DAISY Register (IOMUXC_CAN1_CANRX_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 54Ch offset = 3033_054Ch



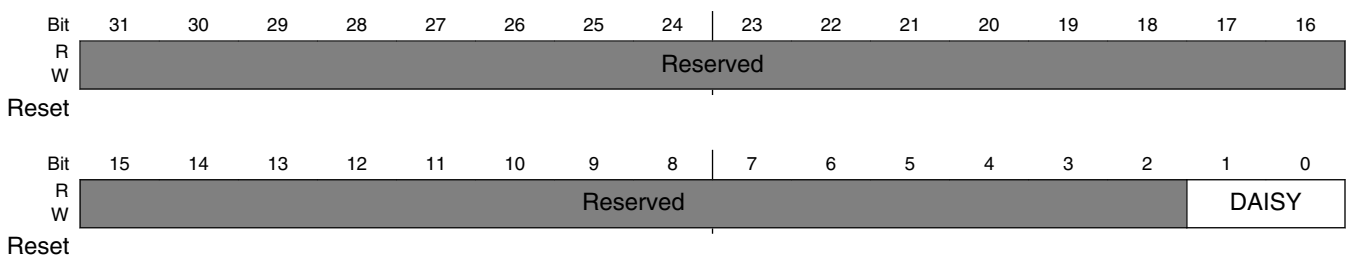
IOMUXC_CAN1_CANRX_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: can1, In Pin: _canrx 00 **SELECT_SAI5_RXD2_ALT6** — Selecting Pad: SAI5_RXD2 for Mode: ALT6 01 **SELECT_SAI2_TXC_ALT3** — Selecting Pad: SAI2_TXC for Mode: ALT3 10 **SELECT_SPDIF_RX_ALT4** — Selecting Pad: SPDIF_RX for Mode: ALT4 11 **SELECT_HDMI_DDC_SDA_ALT4** — Selecting Pad: HDMI_DDC_SDA for Mode: ALT4

8.2.4.336 CAN2_CANRX_SELECT_INPUT DAISY Register (IOMUXC_CAN2_CANRX_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 550h offset = 3033_0550h



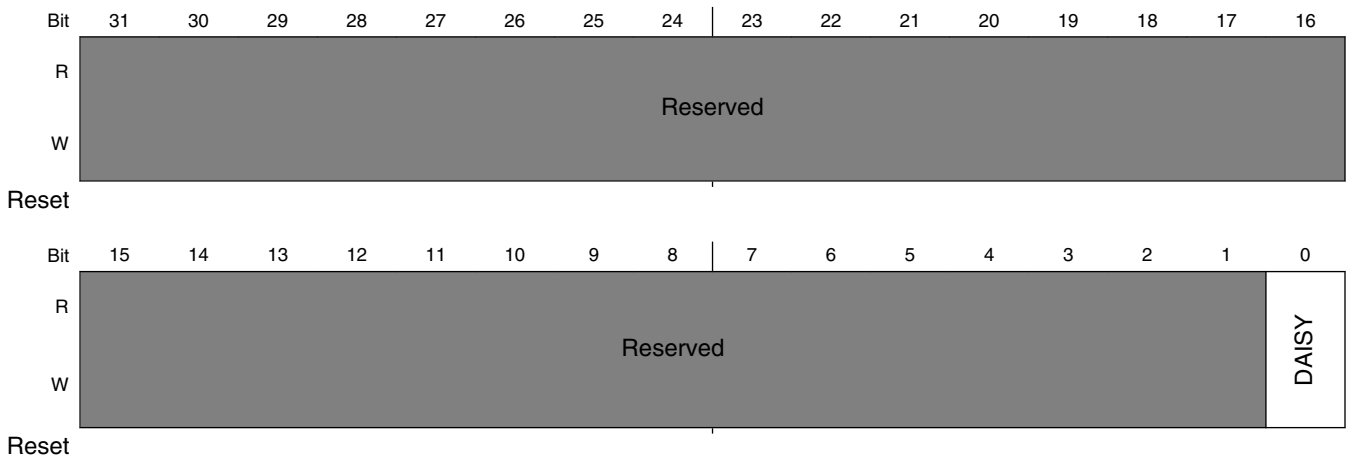
IOMUXC_CAN2_CANRX_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: can2, In Pin: _canrx 00 **SELECT_SAI5_MCLK_ALT6** — Selecting Pad: SAI5_MCLK for Mode: ALT6 01 **SELECT_SAI2_MCLK_ALT3** — Selecting Pad: SAI2_MCLK for Mode: ALT3 10 **SELECT_UART3_TXD_ALT4** — Selecting Pad: UART3_TXD for Mode: ALT4 11 **SELECT_HDMI_HPD_ALT4** — Selecting Pad: HDMI_HPD for Mode: ALT4

8.2.4.337 CCM_GPC_PMIC_VFUNCTIONAL_READY_SELECT_INPUT DAISY Register (IOMUXC_CCM_GPC_PMIC_VFUNCTIONAL_READY_SELECT_INPU

DAISY Register

Address: 3033_0000h base + 554h offset = 3033_0554h



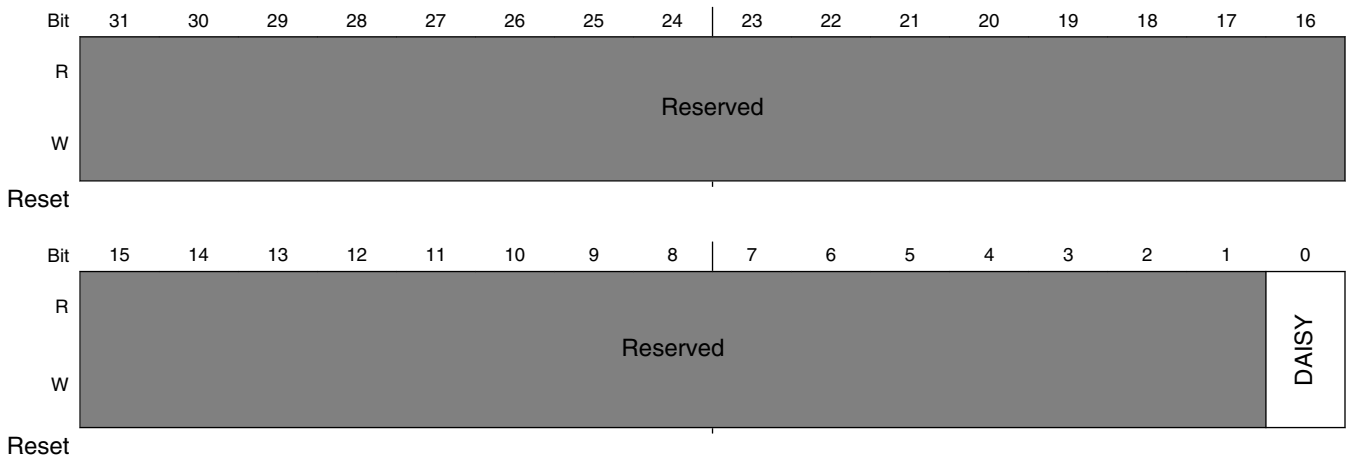
IOMUXC_CCM_GPC_PMIC_VFUNCTIONAL_READY_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ccm, In Pin: gpc_pmic_vfunctional_ready 0 **SELECT_GPIO1_IO05_ALT5** — Selecting Pad: GPIO1_IO05 for Mode: ALT5 1 **SELECT_GPIO1_IO11_ALT5** — Selecting Pad: GPIO1_IO11 for Mode: ALT5

8.2.4.338 ECSPI1_CSPI_CLK_IN_SELECT_INPUT DAISY Register (IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 558h offset = 3033_0558h



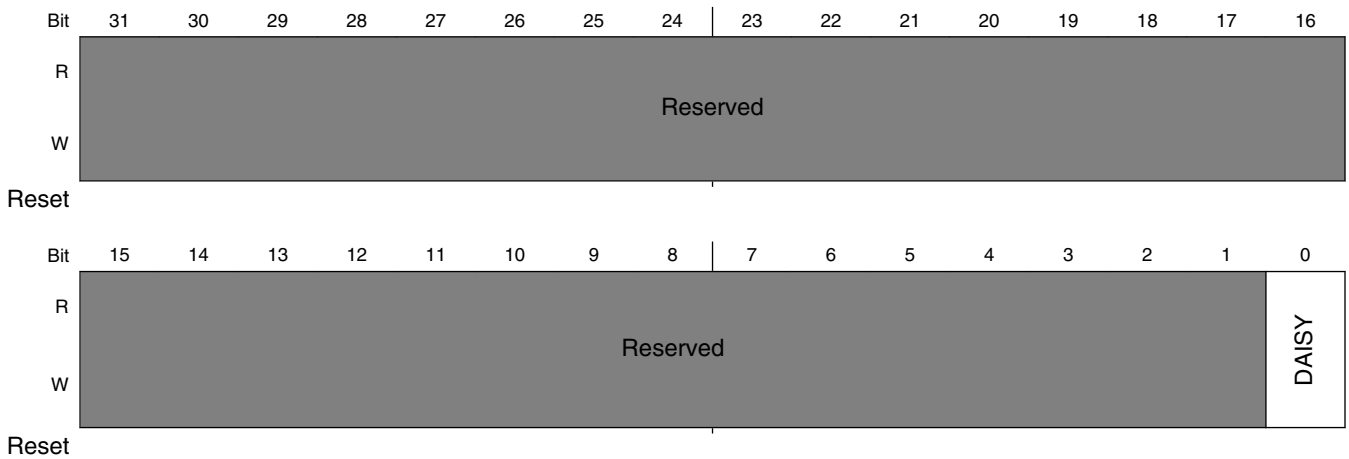
IOMUXC_ECSPI1_CSPI_CLK_IN_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi1, In Pin: _cspi_clk_in 0 **SELECT_ECSPI1_SCLK_ALT0** — Selecting Pad: ECSPI1_SCLK for Mode: ALT0 1 **SELECT_I2C1_SCL_ALT3** — Selecting Pad: I2C1_SCL for Mode: ALT3

8.2.4.339 ECSPI1_MISO_SELECT_INPUT DAISY Register (IOMUXC_ECSPI1_MISO_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 55Ch offset = 3033_055Ch



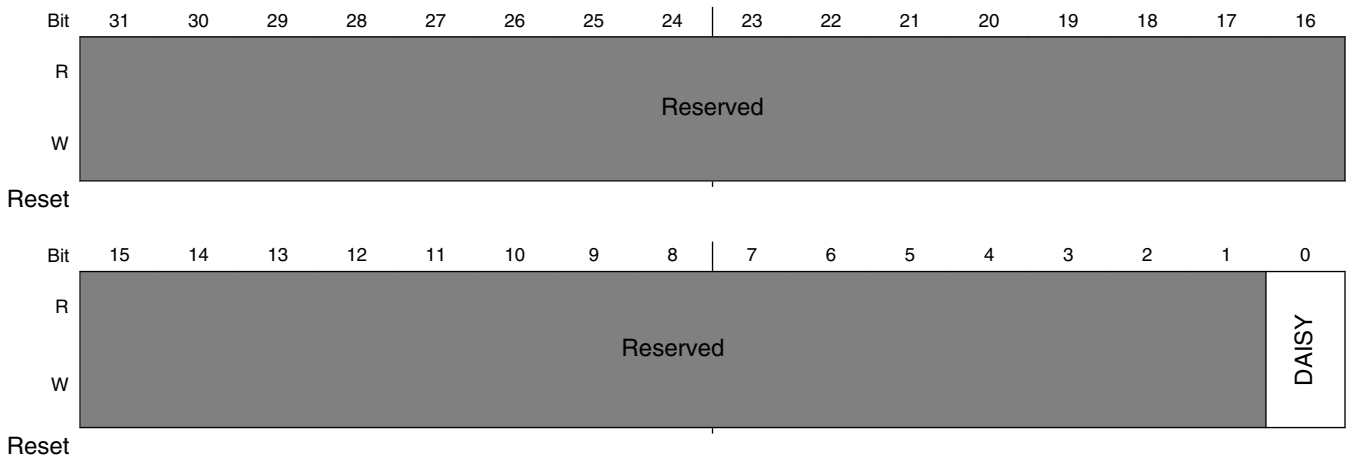
IOMUXC_ECSPI1_MISO_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi1, In Pin: _miso 0 **SELECT_ECSPI1_MISO_ALT0** — Selecting Pad: ECSPI1_MISO for Mode: ALT0 1 **SELECT_I2C2_SCL_ALT3** — Selecting Pad: I2C2_SCL for Mode: ALT3

8.2.4.340 ECSPI1_MOSI_SELECT_INPUT DAISY Register (IOMUXC_ECSPI1_MOSI_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 560h offset = 3033_0560h



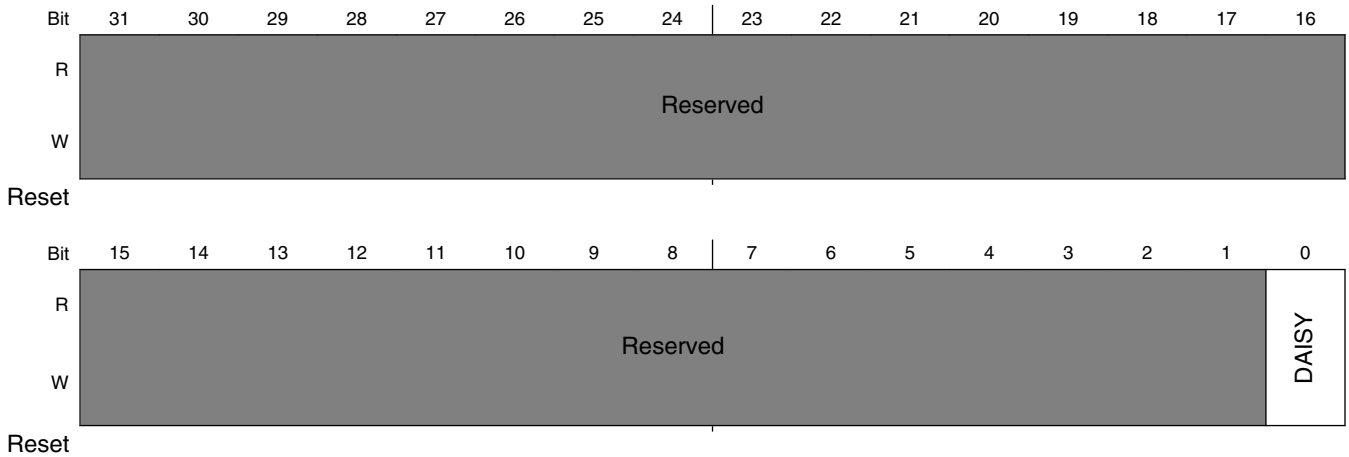
IOMUXC_ECSPI1_MOSI_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi1, In Pin: _mosi 0 **SELECT_ECSPI1_MOSI_ALT0** — Selecting Pad: ECSPI1_MOSI for Mode: ALT0 1 **SELECT_I2C1_SDA_ALT3** — Selecting Pad: I2C1_SDA for Mode: ALT3

8.2.4.341 ECSPI1_SS_B_SELECT_INPUT_0 DAISY Register (IOMUXC_ECSPI1_SS_B_SELECT_INPUT_0)

DAISY Register

Address: 3033_0000h base + 564h offset = 3033_0564h



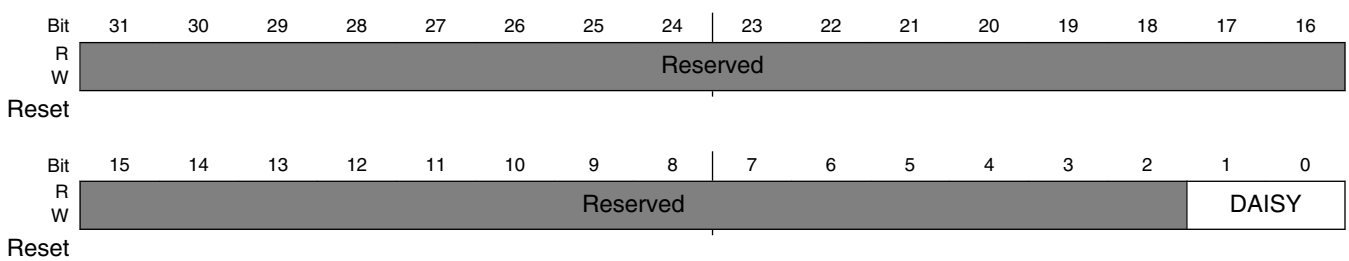
IOMUXC_ECSPI1_SS_B_SELECT_INPUT_0 field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi1, In Pin: _ss_b[0] 0 **SELECT_ECSPI1_SS0_ALT0** — Selecting Pad: ECSPI1_SS0 for Mode: ALT0 1 **SELECT_I2C2_SDA_ALT3** — Selecting Pad: I2C2_SDA for Mode: ALT3

8.2.4.342 ECSPI2_CSPI_CLK_IN_SELECT_INPUT DAISY Register (IOMUXC_ECSPI2_CSPI_CLK_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 568h offset = 3033_0568h



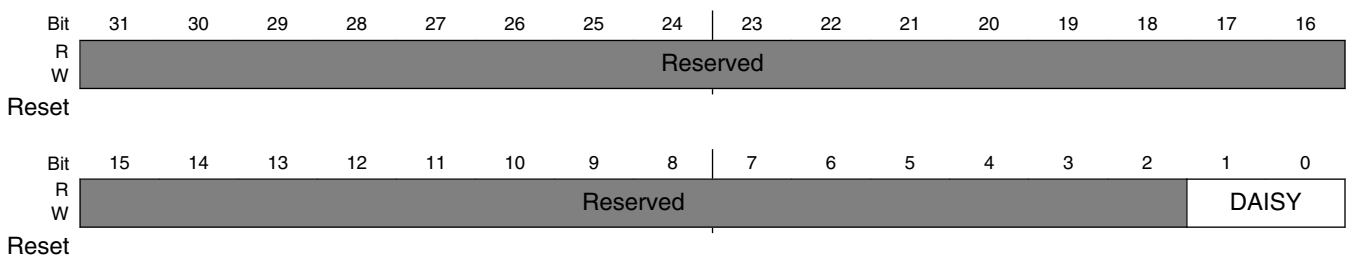
IOMUXC_ECSPI2_CSPI_CLK_IN_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi2, In Pin: _cspi_clk_in 00 **SELECT_SD2_CLK_ALT2** — Selecting Pad: SD2_CLK for Mode: ALT2 01 **SELECT_ECSPI2_SCLK_ALT0** — Selecting Pad: ECSPI2_SCLK for Mode: ALT0 10 **SELECT_I2C3_SCL_ALT3** — Selecting Pad: I2C3_SCL for Mode: ALT3

8.2.4.343 ECSPI2_MISO_SELECT_INPUT DAISY Register (IOMUXC_ECSPI2_MISO_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 56Ch offset = 3033_056Ch



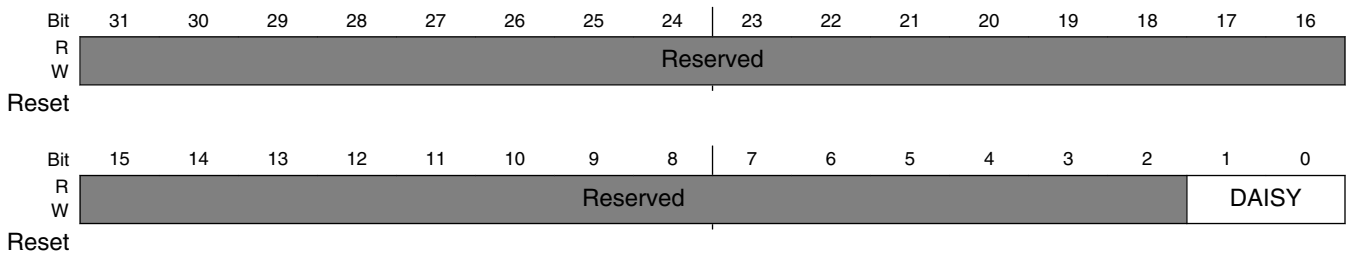
IOMUXC_ECSPI2_MISO_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi2, In Pin: _miso 00 **SELECT_SD2_DATA3_ALT2** — Selecting Pad: SD2_DATA3 for Mode: ALT2 01 **SELECT_ECSPI2_MISO_ALT0** — Selecting Pad: ECSPI2_MISO for Mode: ALT0 10 **SELECT_I2C4_SCL_ALT3** — Selecting Pad: I2C4_SCL for Mode: ALT3

8.2.4.344 ECSPI2_MOSI_SELECT_INPUT DAISY Register (IOMUXC_ECSPI2_MOSI_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 570h offset = 3033_0570h



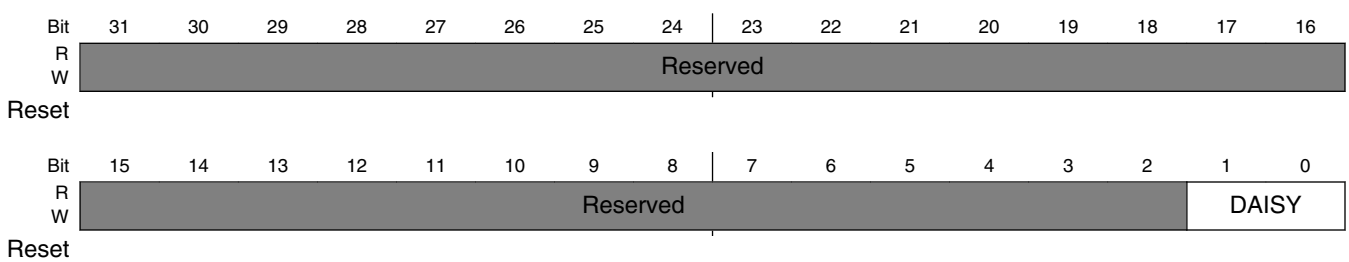
IOMUXC_ECSPI2_MOSI_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi2, In Pin: _mosi 00 **SELECT_SD2_CMD_ALT2** — Selecting Pad: SD2_CMD for Mode: ALT2 01 **SELECT_ECSPI2_MOSI_ALT0** — Selecting Pad: ECSPI2_MOSI for Mode: ALT0 10 **SELECT_I2C3_SDA_ALT3** — Selecting Pad: I2C3_SDA for Mode: ALT3

8.2.4.345 ECSPI2_SS_B_SELECT_INPUT_0 DAISY Register (IOMUXC_ECSPI2_SS_B_SELECT_INPUT_0)

DAISY Register

Address: 3033_0000h base + 574h offset = 3033_0574h



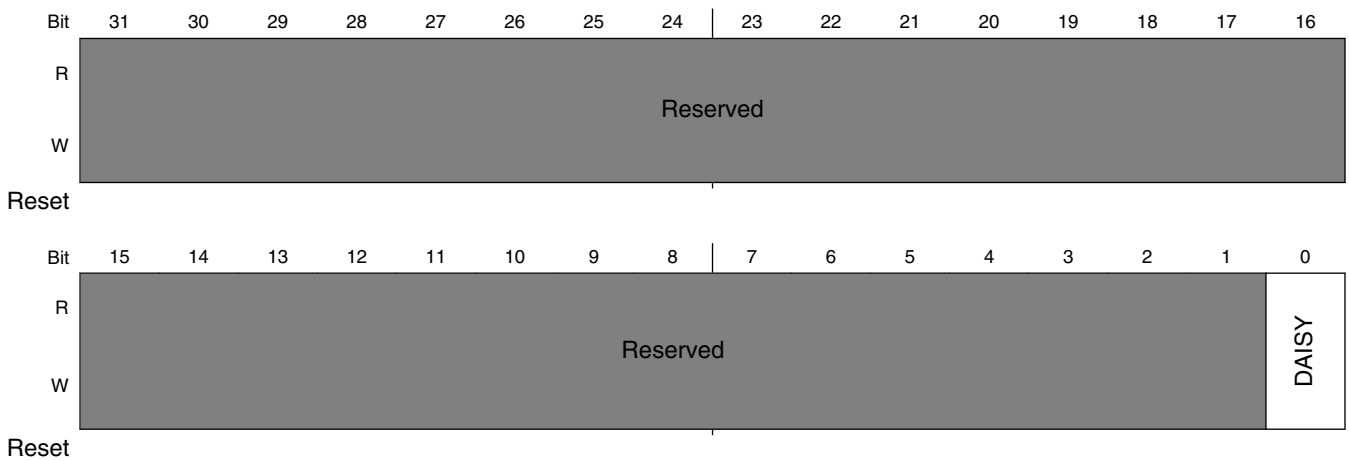
IOMUXC_ECSPI2_SS_B_SELECT_INPUT_0 field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi2, In Pin: _ss_b[0] 00 **SELECT_SD2_DATA2_ALT2** — Selecting Pad: SD2_DATA2 for Mode: ALT2 01 **SELECT_ECSPI2_SS0_ALT0** — Selecting Pad: ECSPI2_SS0 for Mode: ALT0 10 **SELECT_I2C4_SDA_ALT3** — Selecting Pad: I2C4_SDA for Mode: ALT3

8.2.4.346 ENET1_IPG_CLK_RMII_SELECT_INPUT DAISY Register (IOMUXC_ENET1_IPG_CLK_RMII_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 578h offset = 3033_0578h



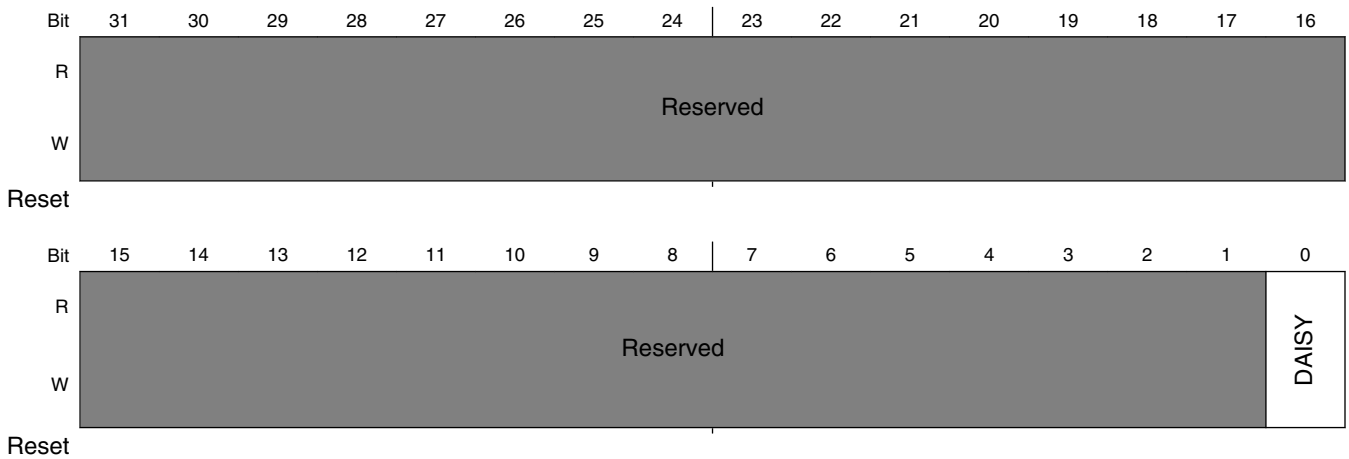
IOMUXC_ENET1_IPG_CLK_RMII_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet1, In Pin: ipg_clk_rmii 0 **SELECT_SD1_RESET_B_ALT1** — Selecting Pad: SD1_RESET_B for Mode: ALT1 1 **SELECT_SAI1_MCLK_ALT4** — Selecting Pad: SAI1_MCLK for Mode: ALT4

8.2.4.347 ENET1_MDIO_SELECT_INPUT DAISY Register (IOMUXC_ENET1_MDIO_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 57Ch offset = 3033_057Ch



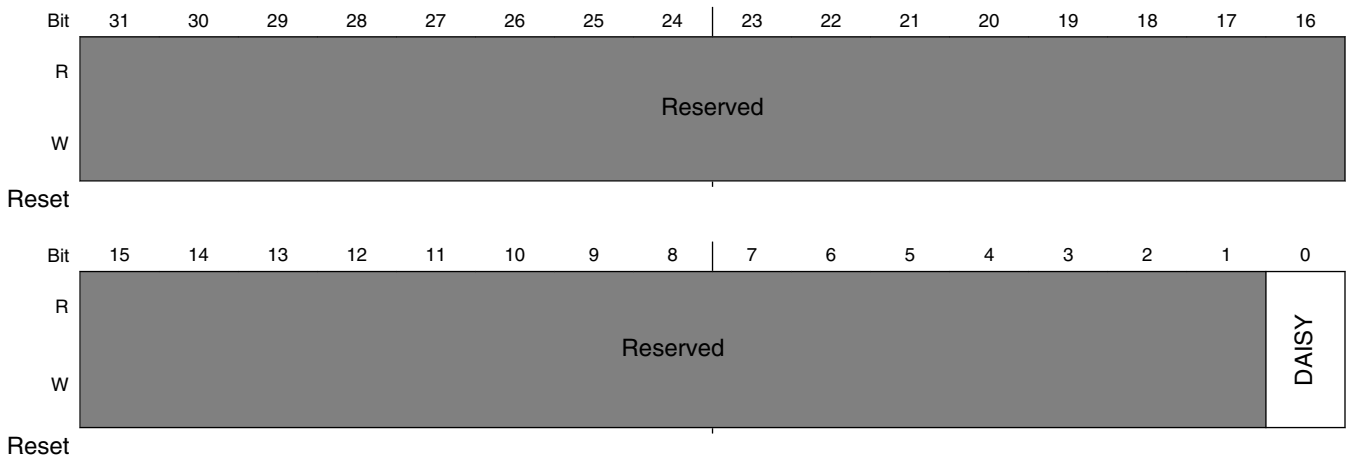
IOMUXC_ENET1_MDIO_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet1, In Pin: _mdio 0 **SELECT_SD1_CMD_ALT1** — Selecting Pad: SD1_CMD for Mode: ALT1 1 **SELECT_SAI1_RXD3_ALT4** — Selecting Pad: SAI1_RXD3 for Mode: ALT4

8.2.4.348 ENET1_RXDATA_0_SELECT_INPUT DAISY Register (IOMUXC_ENET1_RXDATA_0_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 580h offset = 3033_0580h



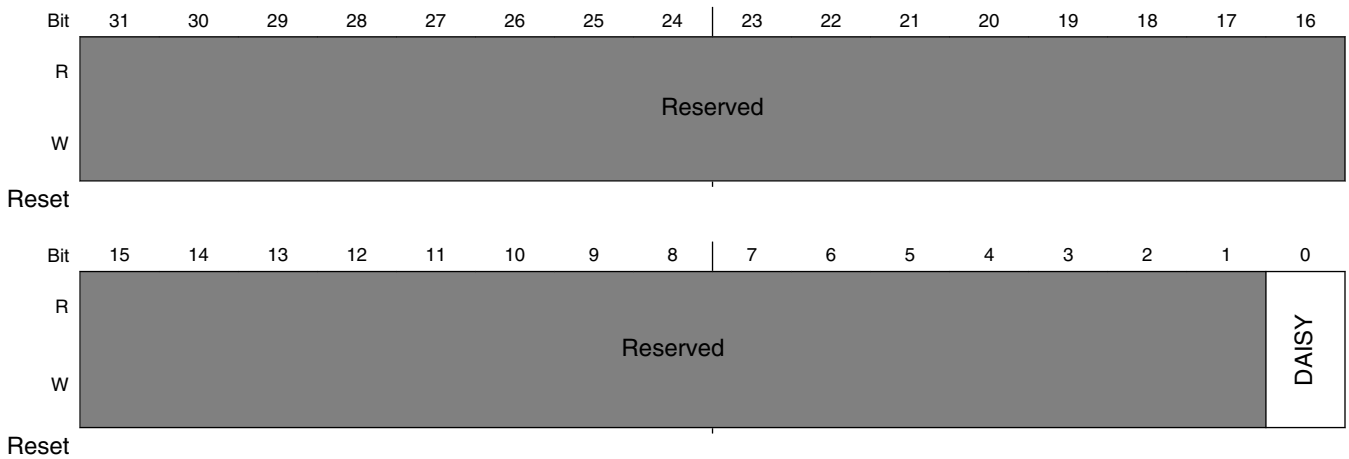
IOMUXC_ENET1_RXDATA_0_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet1, In Pin: _rxdata_0 0 **SELECT_SD1_DATA2_ALT1** — Selecting Pad: SD1_DATA2 for Mode: ALT1 1 **SELECT_SAI1_RXD4_ALT4** — Selecting Pad: SAI1_RXD4 for Mode: ALT4

8.2.4.349 ENET1_RXDATA_1_SELECT_INPUT DAISY Register (IOMUXC_ENET1_RXDATA_1_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 584h offset = 3033_0584h



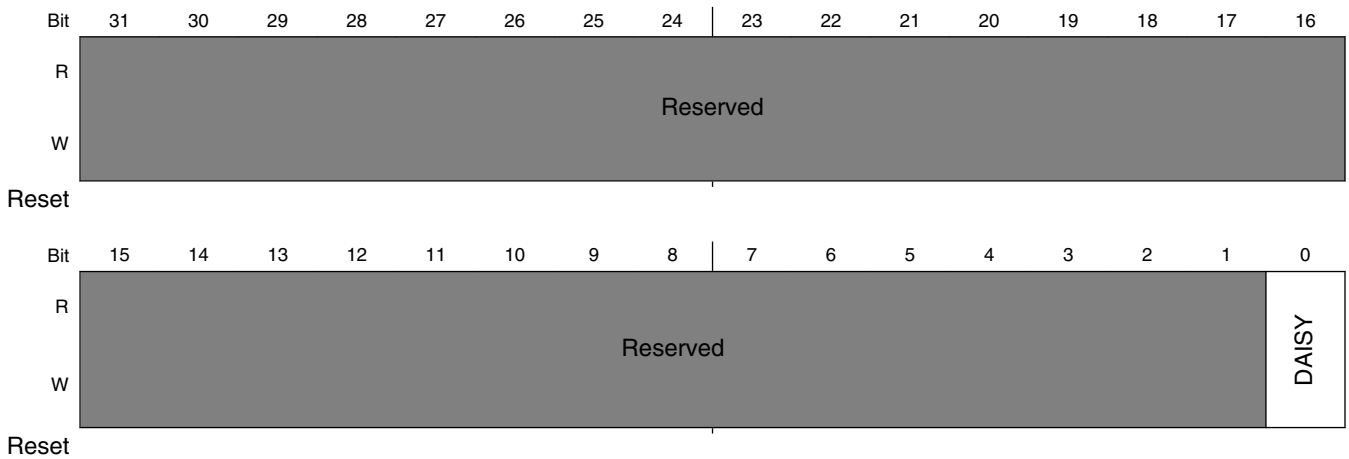
IOMUXC_ENET1_RXDATA_1_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet1, In Pin: _rxdata_1 0 **SELECT_SD1_DATA3_ALT1** — Selecting Pad: SD1_DATA3 for Mode: ALT1 1 **SELECT_SAI1_RXD5_ALT4** — Selecting Pad: SAI1_RXD5 for Mode: ALT4

8.2.4.350 ENET1_RXEN_SELECT_INPUT DAISY Register (IOMUXC_ENET1_RXEN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 588h offset = 3033_0588h



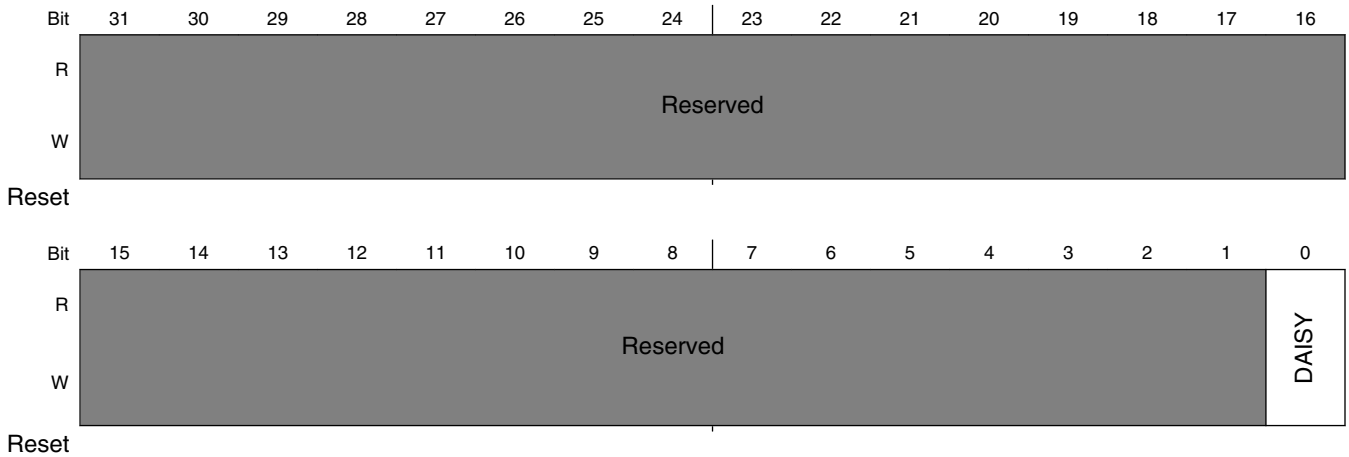
IOMUXC_ENET1_RXEN_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet1, In Pin: _rxen 0 **SELECT_SD1_DATA6_ALT1** — Selecting Pad: SD1_DATA6 for Mode: ALT1 1 **SELECT_SAI1_TXFS_ALT4** — Selecting Pad: SAI1_TXFS for Mode: ALT4

8.2.4.351 ENET1_RXERR_SELECT_INPUT DAISY Register (IOMUXC_ENET1_RXERR_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 58Ch offset = 3033_058Ch



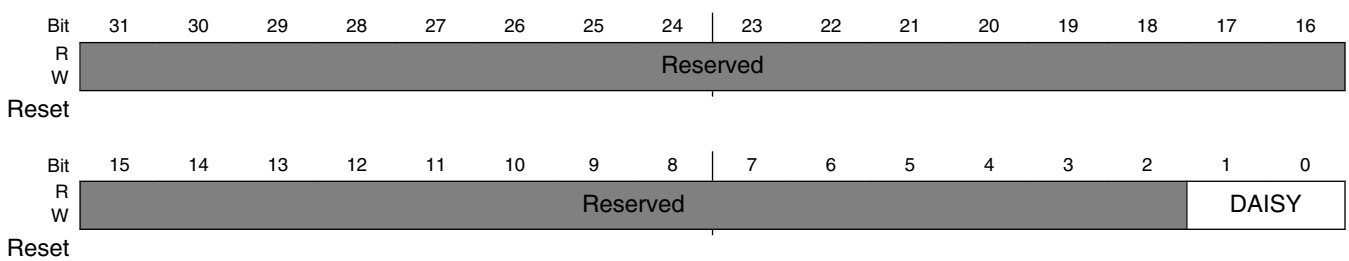
IOMUXC_ENET1_RXERR_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet1, In Pin: _rxerr 0 **SELECT_SD1_DATA7_ALT1** — Selecting Pad: SD1_DATA7 for Mode: ALT1 1 **SELECT_SAI1_TXD6_ALT4** — Selecting Pad: SAI1_TXD6 for Mode: ALT4

8.2.4.352 ENET_QOS_GMII_MDI_I_SELECT_INPUT DAISY Register (IOMUXC_ENET_QOS_GMII_MDI_I_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 590h offset = 3033_0590h



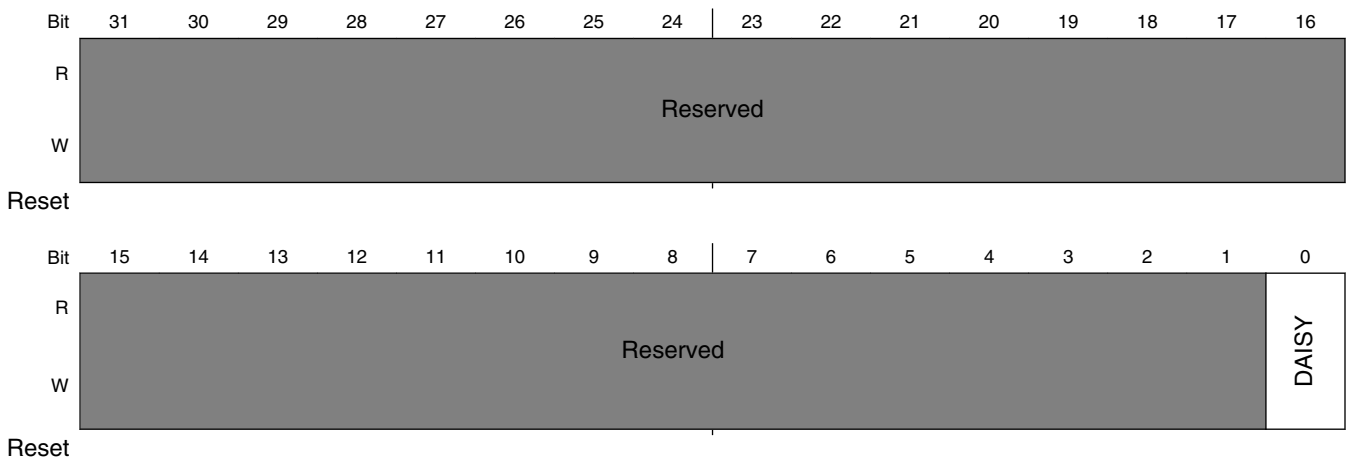
IOMUXC_ENET_QOS_GMII_MDI_I_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet_qos, In Pin: gmii_mdi_i 00 **SELECT_GPIO1_IO07_ALT1** — Selecting Pad: GPIO1_IO07 for Mode: ALT1 01 **SELECT_ENET_MDIO_ALT0** — Selecting Pad: ENET_MDIO for Mode: ALT0 10 **SELECT_I2C1_SDA_ALT1** — Selecting Pad: I2C1_SDA for Mode: ALT1

8.2.4.353 GPT1_CAPIN1_SELECT_INPUT DAISY Register (IOMUXC_GPT1_CAPIN1_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 594h offset = 3033_0594h



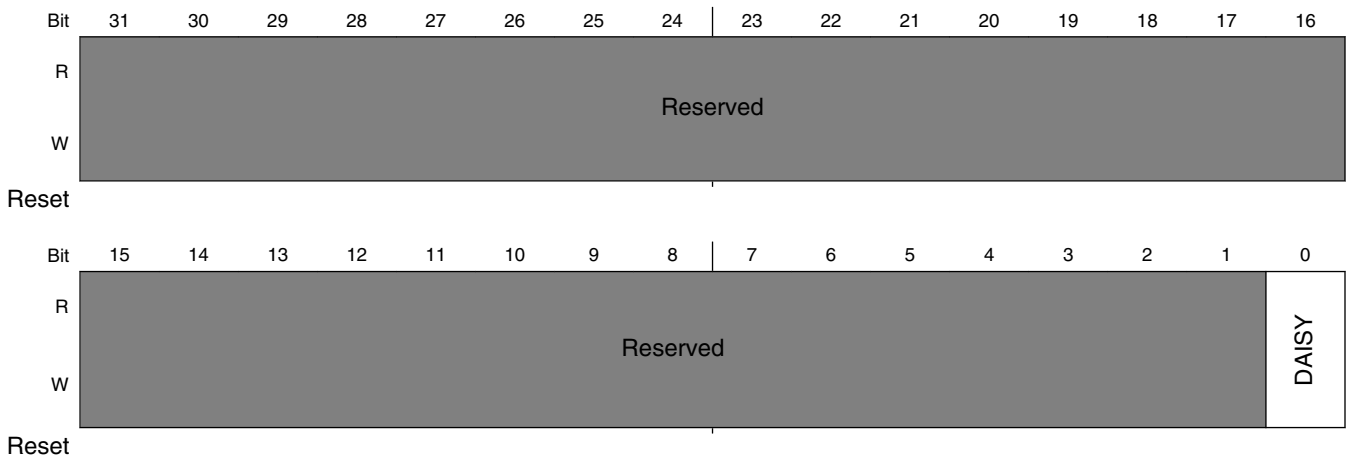
IOMUXC_GPT1_CAPIN1_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: gpt1, In Pin: _capin1 0 **SELECT_SAI3_TXC_ALT3** — Selecting Pad: SAI3_TXC for Mode: ALT3 1 **SELECT_UART4_TXD_ALT3** — Selecting Pad: UART4_TXD for Mode: ALT3

8.2.4.354 GPT1_CAPIN2_SELECT_INPUT DAISY Register (IOMUXC_GPT1_CAPIN2_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 598h offset = 3033_0598h



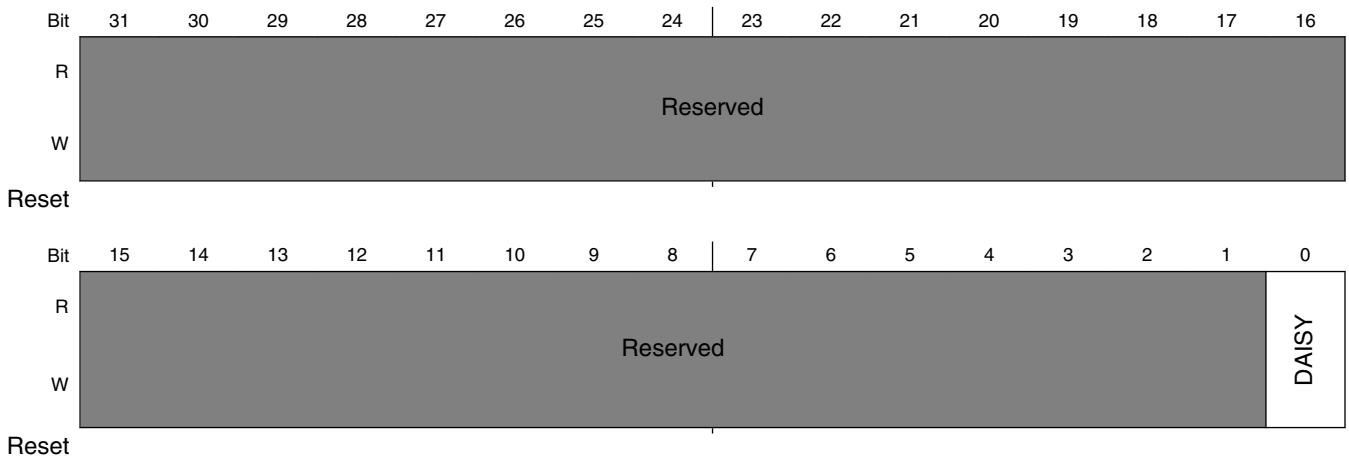
IOMUXC_GPT1_CAPIN2_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: gpt1, In Pin: _capin2 0 **SELECT_SAI3_TXD_ALT3** — Selecting Pad: SAI3_TXD for Mode: ALT3 1 **SELECT_UART3_RXD_ALT3** — Selecting Pad: UART3_RXD for Mode: ALT3

8.2.4.355 GPT1_CLKIN_SELECT_INPUT DAISY Register (IOMUXC_GPT1_CLKIN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 59Ch offset = 3033_059Ch



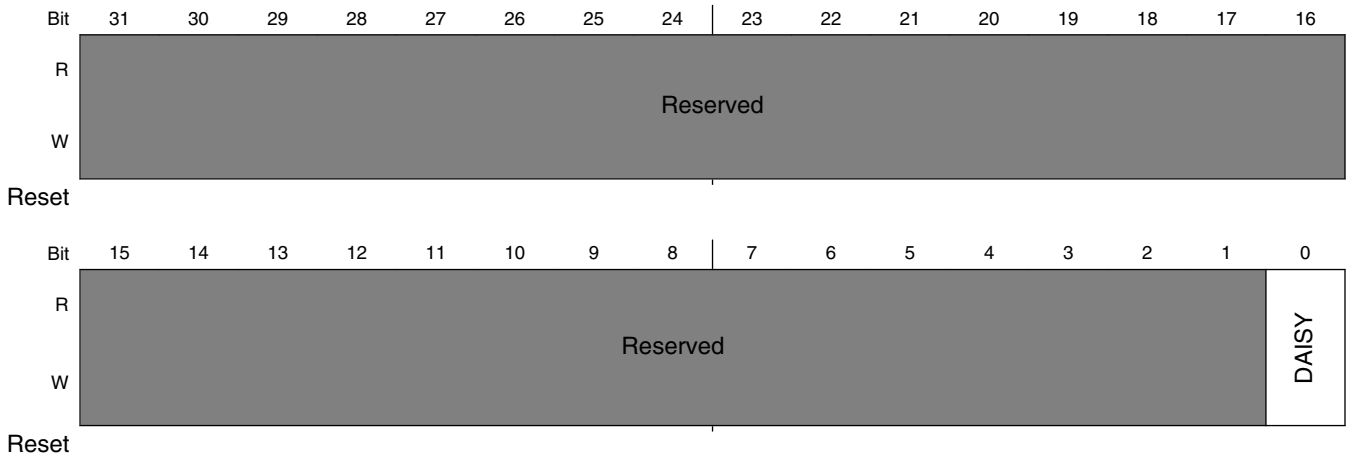
IOMUXC_GPT1_CLKIN_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: gpt1, In Pin: _clkin 0 **SELECT_SAI3_RXC_ALT3** — Selecting Pad: SAI3_RXC for Mode: ALT3 1 **SELECT_UART3_TXD_ALT3** — Selecting Pad: UART3_TXD for Mode: ALT3

8.2.4.356 PCIE_CLKREQ_B_SELECT_INPUT DAISY Register (IOMUXC_PCIE_CLKREQ_B_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5A0h offset = 3033_05A0h



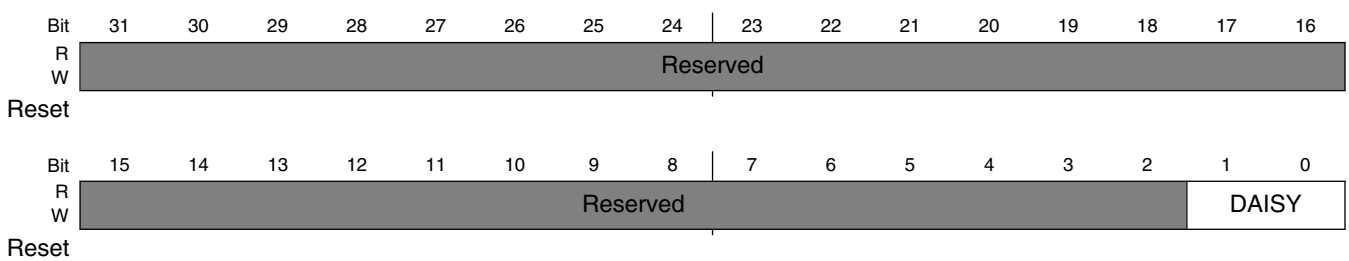
IOMUXC_PCIE_CLKREQ_B_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: pcie, In Pin: pcie_clkreq_b 0 **SELECT_I2C4_SCL_ALT2** — Selecting Pad: I2C4_SCL for Mode: ALT2 1 **SELECT_UART4_RXD_ALT2** — Selecting Pad: UART4_RXD for Mode: ALT2

8.2.4.357 I2C1_SCL_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C1_SCL_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5A4h offset = 3033_05A4h



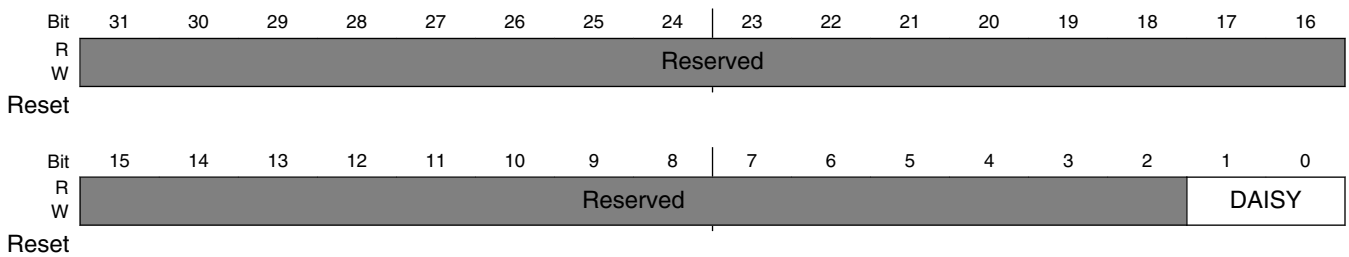
IOMUXC_I2C1_SCL_IN_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c1, In Pin: _scl_in 00 **SELECT_SD1_DATA4_ALT3** — Selecting Pad: SD1_DATA4 for Mode: ALT3 01 **SELECT_ECSPI1_SCLK_ALT2** — Selecting Pad: ECSPI1_SCLK for Mode: ALT2 10 **SELECT_I2C1_SCL_ALT0** — Selecting Pad: I2C1_SCL for Mode: ALT0

8.2.4.358 I2C1_SDA_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C1_SDA_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5A8h offset = 3033_05A8h



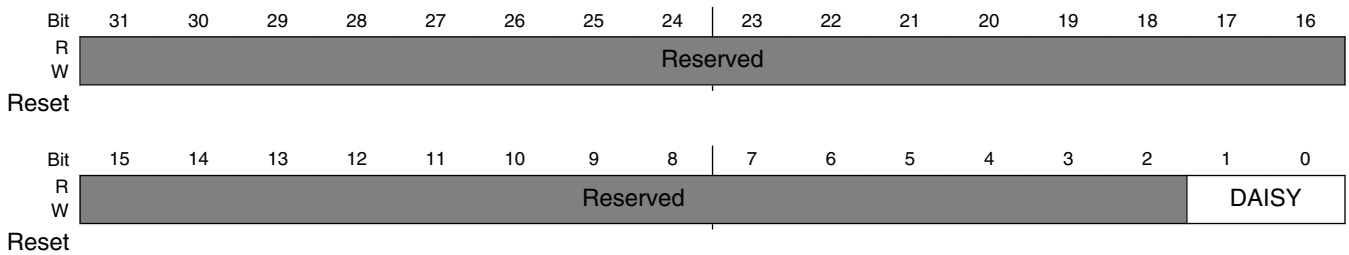
IOMUXC_I2C1_SDA_IN_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c1, In Pin: _sda_in 00 **SELECT_SD1_DATA5_ALT3** — Selecting Pad: SD1_DATA5 for Mode: ALT3 01 **SELECT_ECSPI1_MOSI_ALT2** — Selecting Pad: ECSPI1_MOSI for Mode: ALT2 10 **SELECT_I2C1_SDA_ALT0** — Selecting Pad: I2C1_SDA for Mode: ALT0

8.2.4.359 I2C2_SCL_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C2_SCL_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5ACh offset = 3033_05ACh



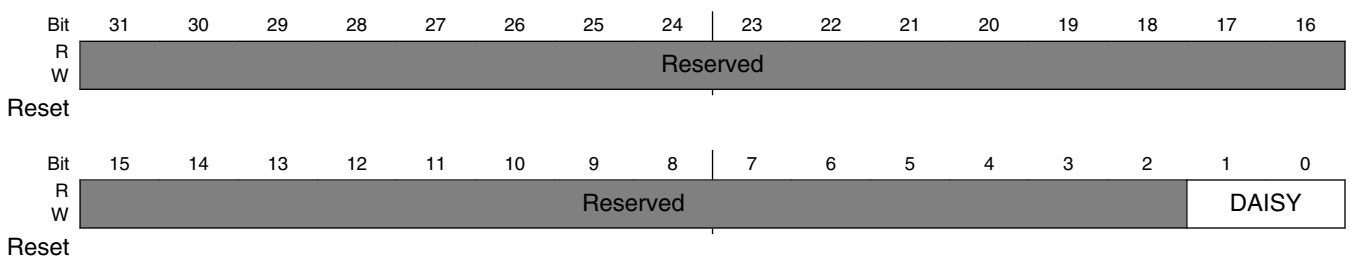
IOMUXC_I2C2_SCL_IN_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c2, In Pin: _scl_in 00 **SELECT_SD1_DATA6_ALT3** — Selecting Pad: SD1_DATA6 for Mode: ALT3 01 **SELECT_ECSPI1_MISO_ALT2** — Selecting Pad: ECSPI1_MISO for Mode: ALT2 10 **SELECT_I2C2_SCL_ALT0** — Selecting Pad: I2C2_SCL for Mode: ALT0

8.2.4.360 I2C2_SDA_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C2_SDA_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5B0h offset = 3033_05B0h



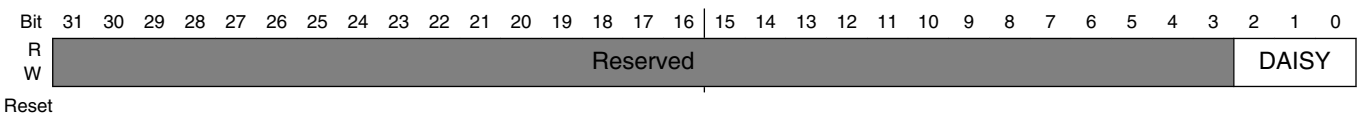
IOMUXC_I2C2_SDA_IN_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c2, In Pin: _sda_in 00 **SELECT_SD1_DATA7_ALT3** — Selecting Pad: SD1_DATA7 for Mode: ALT3 01 **SELECT_ECSPI1_SS0_ALT2** — Selecting Pad: ECSPI1_SS0 for Mode: ALT2 10 **SELECT_I2C2_SDA_ALT0** — Selecting Pad: I2C2_SDA for Mode: ALT0

8.2.4.361 I2C3_SCL_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C3_SCL_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5B4h offset = 3033_05B4h



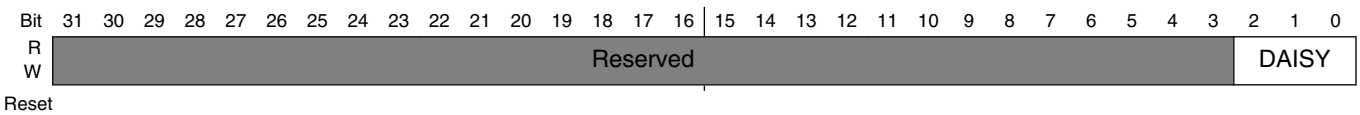
IOMUXC_I2C3_SCL_IN_SELECT_INPUT field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c3, In Pin: _scl_in 000 **SELECT_SD1_RESET_B_ALT3** — Selecting Pad: SD1_RESET_B for Mode: ALT3 001 **SELECT_NAND_DQS_ALT4** — Selecting Pad: NAND_DQS for Mode: ALT4 010 **SELECT_NAND_READY_B_ALT4** — Selecting Pad: NAND_READY_B for Mode: ALT4 011 **SELECT_ECSPI2_SCLK_ALT2** — Selecting Pad: ECSPI2_SCLK for Mode: ALT2 100 **SELECT_I2C3_SCL_ALT0** — Selecting Pad: I2C3_SCL for Mode: ALT0

8.2.4.362 I2C3_SDA_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C3_SDA_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5B8h offset = 3033_05B8h



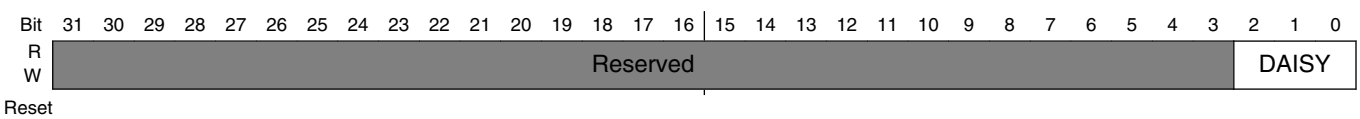
IOMUXC_I2C3_SDA_IN_SELECT_INPUT field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c3, In Pin: _sda_in 000 **SELECT_SD1_STROBE_ALT3** — Selecting Pad: SD1_STROBE for Mode: ALT3 001 **SELECT_NAND_CE3_B_ALT4** — Selecting Pad: NAND_CE3_B for Mode: ALT4 010 **SELECT_NAND_WE_B_ALT4** — Selecting Pad: NAND_WE_B for Mode: ALT4 011 **SELECT_ECSPI2_MOSI_ALT2** — Selecting Pad: ECSPI2_MOSI for Mode: ALT2 100 **SELECT_I2C3_SDA_ALT0** — Selecting Pad: I2C3_SDA for Mode: ALT0

8.2.4.363 I2C4_SCL_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C4_SCL_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5BCh offset = 3033_05BCh



IOMUXC_I2C4_SCL_IN_SELECT_INPUT field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c4, In Pin: _scl_in 000 **SELECT_SD1_DATA2_ALT3** — Selecting Pad: SD1_DATA2 for Mode: ALT3

Table continues on the next page...

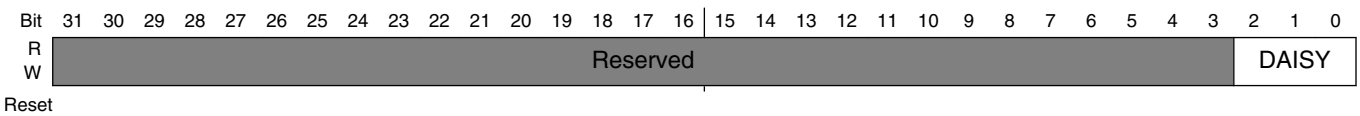
IOMUXC_I2C4_SCL_IN_SELECT_INPUT field descriptions (continued)

Field	Description
	001 **SELECT_SD2_DATA1_ALT2** — Selecting Pad: SD2_DATA1 for Mode: ALT2 010 **SELECT_NAND_CE1_B_ALT4** — Selecting Pad: NAND_CE1_B for Mode: ALT4 011 **SELECT_NAND_WP_B_ALT4** — Selecting Pad: NAND_WP_B for Mode: ALT4 100 **SELECT_ECSPI2_MISO_ALT2** — Selecting Pad: ECSPI2_MISO for Mode: ALT2 101 **SELECT_I2C4_SCL_ALT0** — Selecting Pad: I2C4_SCL for Mode: ALT0

8.2.4.364 I2C4_SDA_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C4_SDA_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5C0h offset = 3033_05C0h



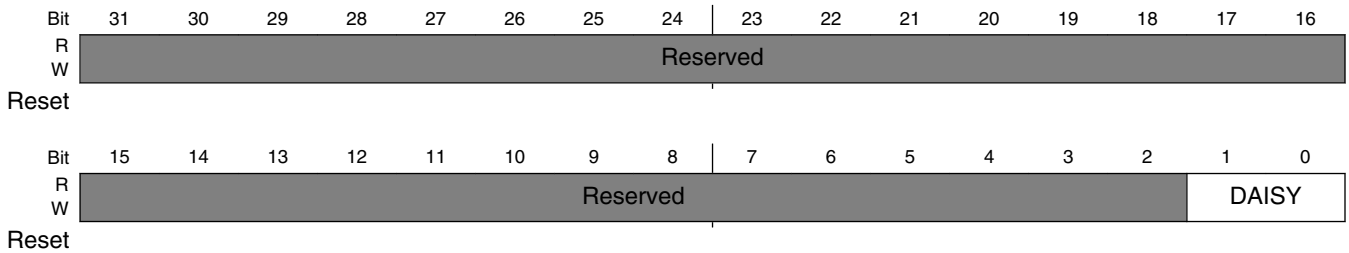
IOMUXC_I2C4_SDA_IN_SELECT_INPUT field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c4, In Pin: _sda_in 000 **SELECT_SD1_DATA3_ALT3** — Selecting Pad: SD1_DATA3 for Mode: ALT3 001 **SELECT_SD2_DATA0_ALT2** — Selecting Pad: SD2_DATA0 for Mode: ALT2 010 **SELECT_NAND_CE2_B_ALT4** — Selecting Pad: NAND_CE2_B for Mode: ALT4 011 **SELECT_NAND_DATA02_ALT4** — Selecting Pad: NAND_DATA02 for Mode: ALT4 100 **SELECT_ECSPI2_SS0_ALT2** — Selecting Pad: ECSPI2_SS0 for Mode: ALT2 101 **SELECT_I2C4_SDA_ALT0** — Selecting Pad: I2C4_SDA for Mode: ALT0

8.2.4.365 I2C5_SCL_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C5_SCL_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5C4h offset = 3033_05C4h



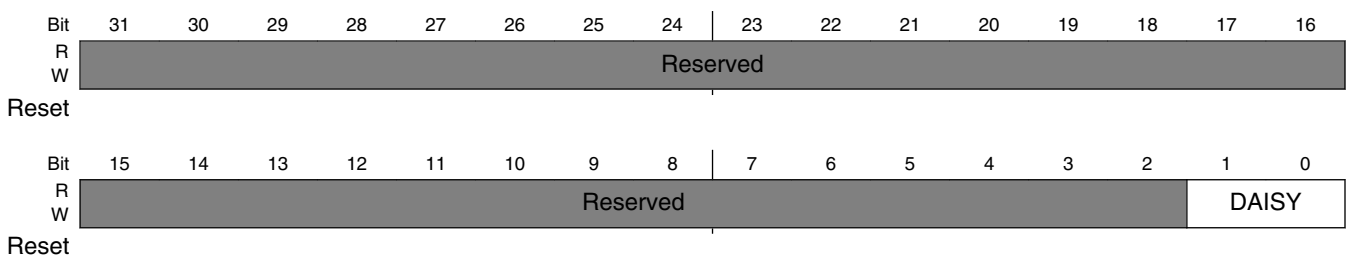
IOMUXC_I2C5_SCL_IN_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c5, In Pin: _scl_in 00 **SELECT_SD1_CLK_ALT3** — Selecting Pad: SD1_CLK for Mode: ALT3 01 **SELECT_SAI5_RXD0_ALT3** — Selecting Pad: SAI5_RXD0 for Mode: ALT3 10 **SELECT_SPDIF_TX_ALT2** — Selecting Pad: SPDIF_TX for Mode: ALT2 11 **SELECT_HDMI_DDC_SCL_ALT3** — Selecting Pad: HDMI_DDC_SCL for Mode: ALT3

8.2.4.366 I2C5_SDA_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C5_SDA_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5C8h offset = 3033_05C8h



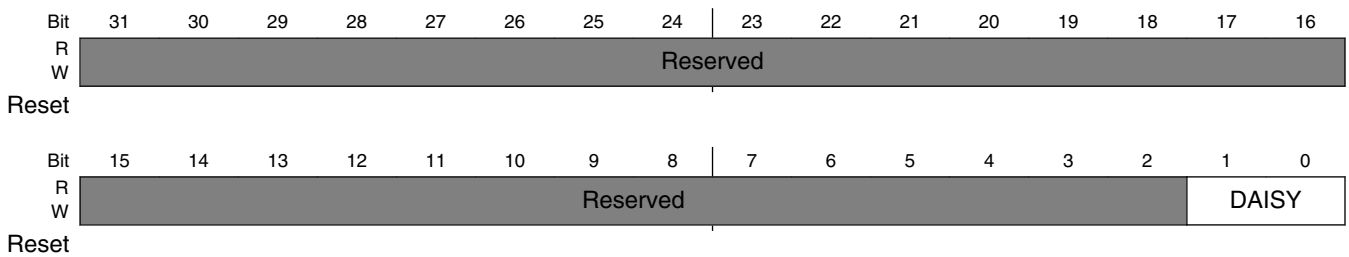
IOMUXC_I2C5_SDA_IN_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c5, In Pin: _sda_in 00 **SELECT_SD1_CMD_ALT3** — Selecting Pad: SD1_CMD for Mode: ALT3 01 **SELECT_SAI5_MCLK_ALT3** — Selecting Pad: SAI5_MCLK for Mode: ALT3 10 **SELECT_SPDIF_RX_ALT2** — Selecting Pad: SPDIF_RX for Mode: ALT2 11 **SELECT_HDMI_DDC_SDA_ALT3** — Selecting Pad: HDMI_DDC_SDA for Mode: ALT3

8.2.4.367 I2C6_SCL_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C6_SCL_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5CCh offset = 3033_05CCh



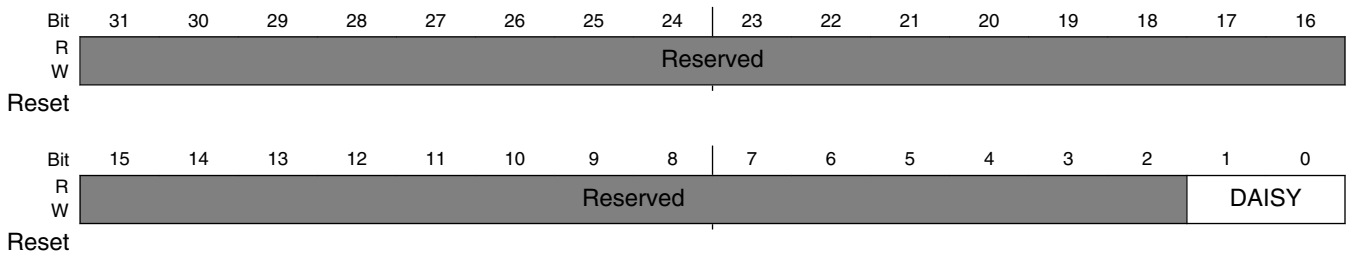
IOMUXC_I2C6_SCL_IN_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c6, In Pin: _scl_in 00 **SELECT_SD1_DATA0_ALT3** — Selecting Pad: SD1_DATA0 for Mode: ALT3 01 **SELECT_SAI5_RXFS_ALT3** — Selecting Pad: SAI5_RXFS for Mode: ALT3 10 **SELECT_UART4_RXD_ALT4** — Selecting Pad: UART4_RXD for Mode: ALT4 11 **SELECT_HDMI_CEC_ALT3** — Selecting Pad: HDMI_CEC for Mode: ALT3

8.2.4.368 I2C6_SDA_IN_SELECT_INPUT DAISY Register (IOMUXC_I2C6_SDA_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5D0h offset = 3033_05D0h



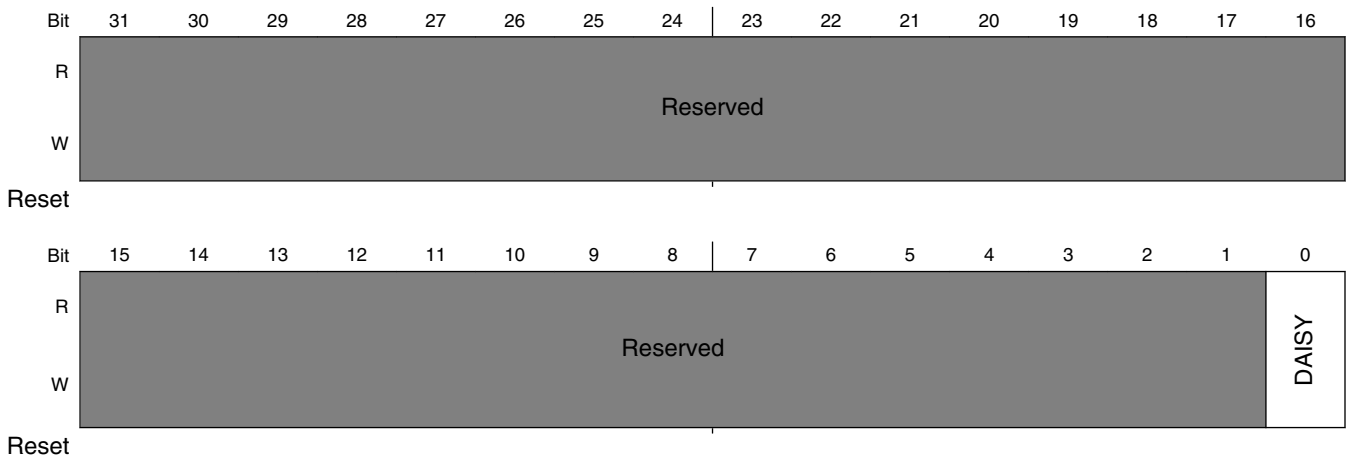
IOMUXC_I2C6_SDA_IN_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c6, In Pin: _sda_in 00 **SELECT_SD1_DATA1_ALT3** — Selecting Pad: SD1_DATA1 for Mode: ALT3 01 **SELECT_SAI5_RXC_ALT3** — Selecting Pad: SAI5_RXC for Mode: ALT3 10 **SELECT_UART4_TXD_ALT4** — Selecting Pad: UART4_TXD for Mode: ALT4 11 **SELECT_HDMI_HPD_ALT3** — Selecting Pad: HDMI_HPD for Mode: ALT3

8.2.4.369 ISP_FL_TRIG_0_SELECT_INPUT DAISY Register (IOMUXC_ISP_FL_TRIG_0_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5D4h offset = 3033_05D4h



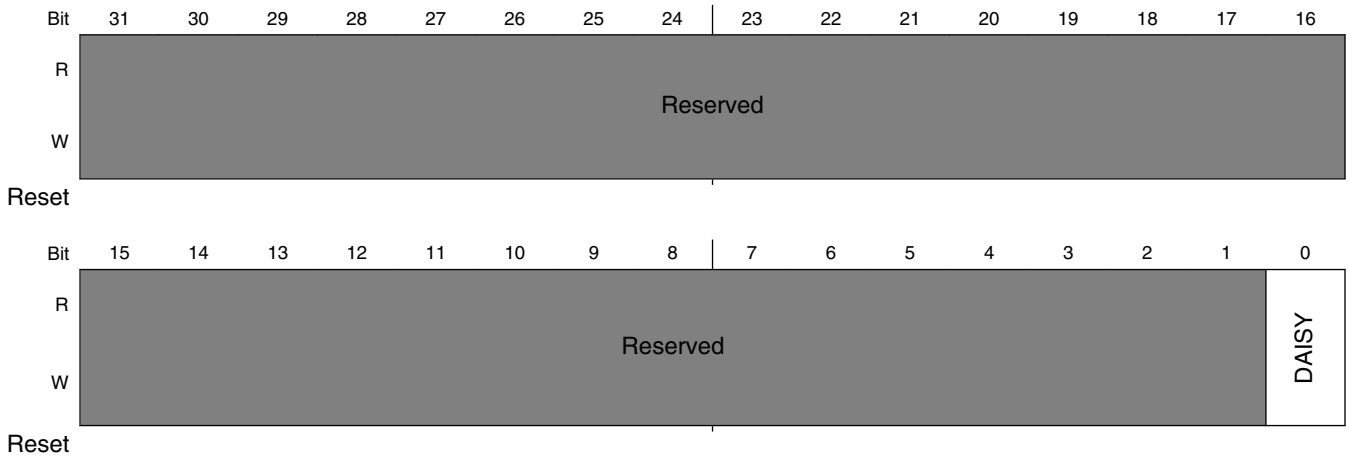
IOMUXC_ISP_FL_TRIG_0_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: isp, In Pin: isp_fl_trig_0 0 **SELECT_GPIO1_IO00_ALT3** — Selecting Pad: GPIO1_IO00 for Mode: ALT3 1 **SELECT_NAND_ALE_ALT3** — Selecting Pad: NAND_ALE for Mode: ALT3

8.2.4.370 ISP_FL_TRIG_1_SELECT_INPUT DAISY Register (IOMUXC_ISP_FL_TRIG_1_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5D8h offset = 3033_05D8h



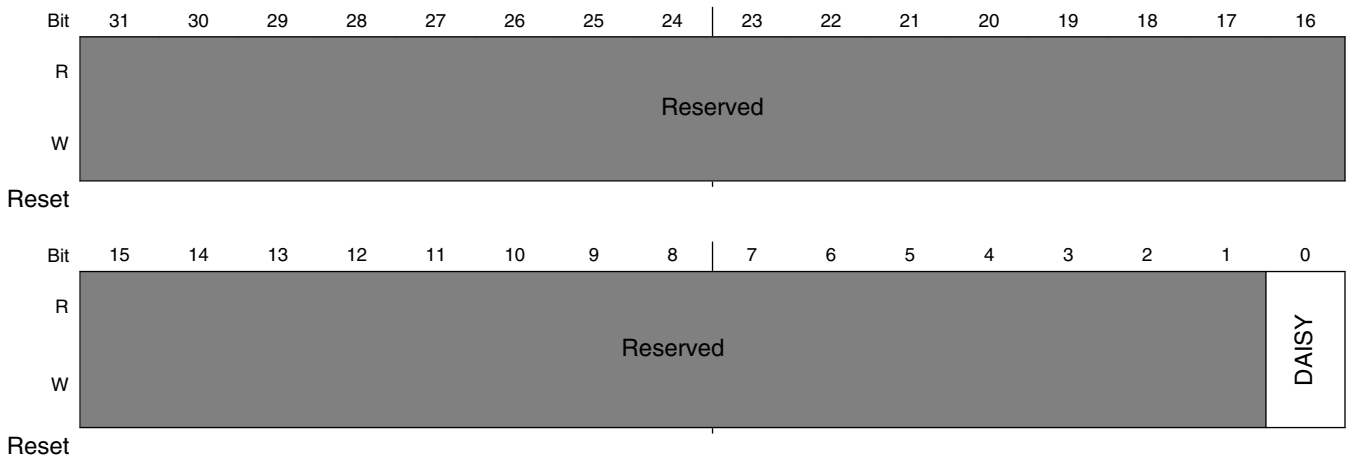
IOMUXC_ISP_FL_TRIG_1_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: isp, In Pin: isp_fl_trig_1 0 **SELECT_GPIO1_IO05_ALT3** — Selecting Pad: GPIO1_IO05 for Mode: ALT3 1 **SELECT_NAND_DATA03_ALT4** — Selecting Pad: NAND_DATA03 for Mode: ALT4

8.2.4.371 ISP_SHUTTER_TRIG_0_SELECT_INPUT DAISY Register (IOMUXC_ISP_SHUTTER_TRIG_0_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5DCh offset = 3033_05DCh



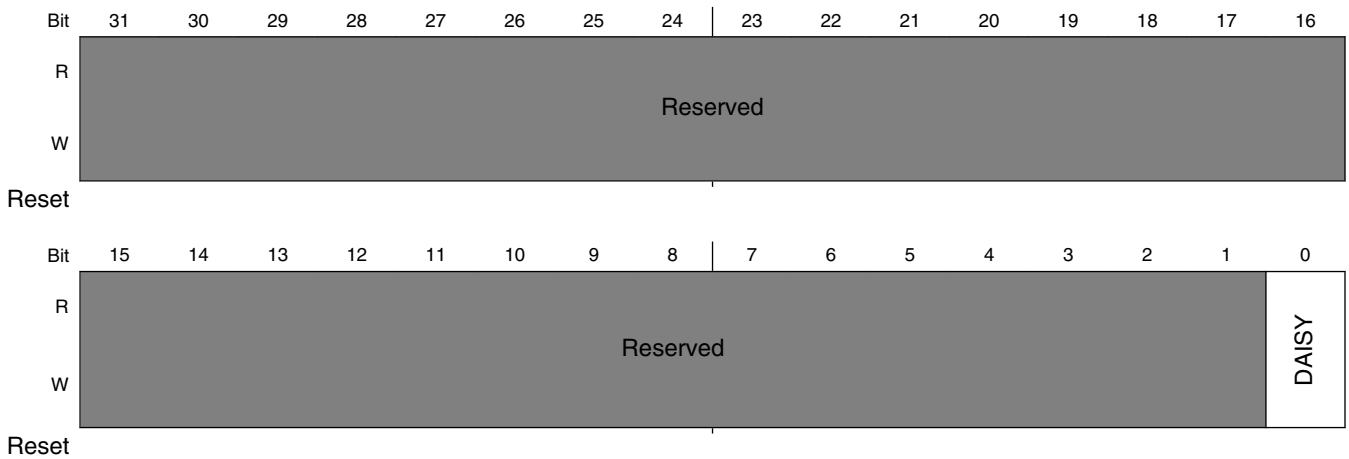
IOMUXC_ISP_SHUTTER_TRIG_0_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: isp, In Pin: isp_shutter_trig_0 0 **SELECT_GPIO1_IO01_ALT3** — Selecting Pad: GPIO1_IO01 for Mode: ALT3 1 **SELECT_NAND_CE0_B_ALT3** — Selecting Pad: NAND_CE0_B for Mode: ALT3

8.2.4.372 ISP_SHUTTER_TRIG_1_SELECT_INPUT DAISY Register (IOMUXC_ISP_SHUTTER_TRIG_1_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5E0h offset = 3033_05E0h



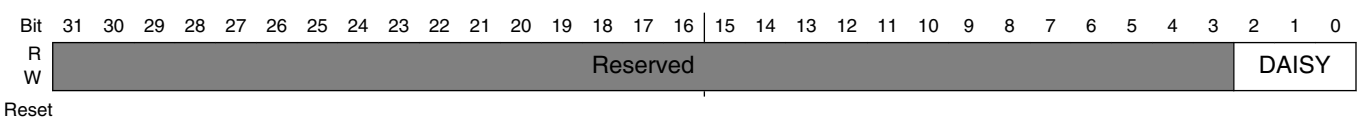
IOMUXC_ISP_SHUTTER_TRIG_1_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: isp, In Pin: isp_shutter_trig_1 0 **SELECT_GPIO1_IO06_ALT3** — Selecting Pad: GPIO1_IO06 for Mode: ALT3 1 **SELECT_NAND_DATA04_ALT4** — Selecting Pad: NAND_DATA04 for Mode: ALT4

8.2.4.373 UART1_UART_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART1_UART_RTS_B_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5E4h offset = 3033_05E4h



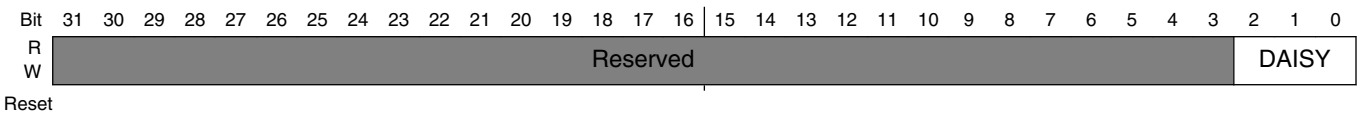
IOMUXC_UART1_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart1, In Pin: _uart_rts_b 000 **SELECT_SD1_DATA0_ALT4** — Selecting Pad: SD1_DATA0 for Mode: ALT4 001 **SELECT_SD1_DATA1_ALT4** — Selecting Pad: SD1_DATA1 for Mode: ALT4 010 **SELECT_SAI2_RXD0_ALT4** — Selecting Pad: SAI2_RXD0 for Mode: ALT4 011 **SELECT_SAI2_TXFS_ALT4** — Selecting Pad: SAI2_TXFS for Mode: ALT4 100 **SELECT_UART3_RXD_ALT1** — Selecting Pad: UART3_RXD for Mode: ALT1 101 **SELECT_UART3_TXD_ALT1** — Selecting Pad: UART3_TXD for Mode: ALT1

8.2.4.374 UART1_UART_RXD_MUX_SELECT_INPUT DAISY Register (IOMUXC_UART1_UART_RXD_MUX_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5E8h offset = 3033_05E8h



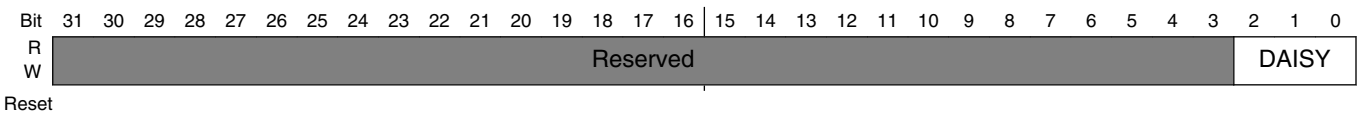
IOMUXC_UART1_UART_RXD_MUX_SELECT_INPUT field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart1, In Pin: _uart_rxd_mux 000 **SELECT_SD1_CLK_ALT4** — Selecting Pad: SD1_CLK for Mode: ALT4 001 **SELECT_SD1_CMD_ALT4** — Selecting Pad: SD1_CMD for Mode: ALT4 010 **SELECT_SAI2_RXFS_ALT4** — Selecting Pad: SAI2_RXFS for Mode: ALT4 011 **SELECT_SAI2_RXC_ALT4** — Selecting Pad: SAI2_RXC for Mode: ALT4 100 **SELECT_UART1_RXD_ALT0** — Selecting Pad: UART1_RXD for Mode: ALT0 101 **SELECT_UART1_TXD_ALT0** — Selecting Pad: UART1_TXD for Mode: ALT0

8.2.4.375 UART2_UART_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART2_UART_RTS_B_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5ECh offset = 3033_05ECh



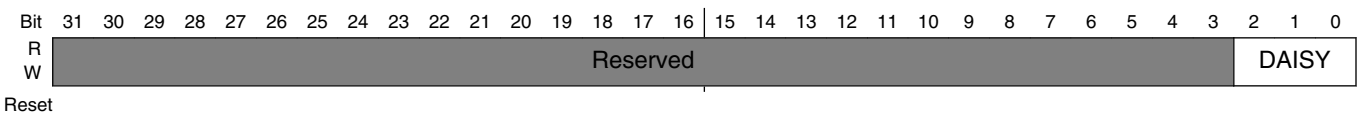
IOMUXC_UART2_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart2, In Pin: _uart_rts_b 000 **SELECT_SD1_DATA4_ALT4** — Selecting Pad: SD1_DATA4 for Mode: ALT4 001 **SELECT_SD1_DATA5_ALT4** — Selecting Pad: SD1_DATA5 for Mode: ALT4 010 **SELECT_SAI3_RXC_ALT4** — Selecting Pad: SAI3_RXC for Mode: ALT4 011 **SELECT_SAI3_RXD_ALT4** — Selecting Pad: SAI3_RXD for Mode: ALT4 100 **SELECT_UART4_RXD_ALT1** — Selecting Pad: UART4_RXD for Mode: ALT1 101 **SELECT_UART4_TXD_ALT1** — Selecting Pad: UART4_TXD for Mode: ALT1

8.2.4.376 UART2_UART_RXD_MUX_SELECT_INPUT DAISY Register (IOMUXC_UART2_UART_RXD_MUX_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5F0h offset = 3033_05F0h



IOMUXC_UART2_UART_RXD_MUX_SELECT_INPUT field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart2, In Pin: _uart_rxd_mux

Table continues on the next page...

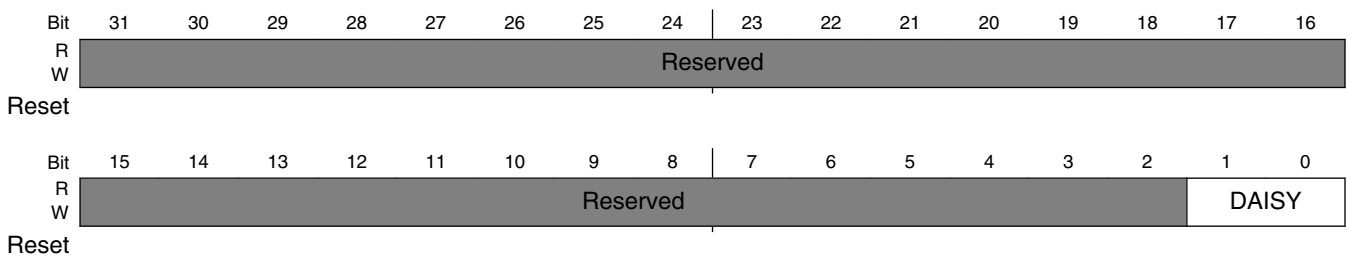
IOMUXC_UART2_UART_RXD_MUX_SELECT_INPUT field descriptions (continued)

Field	Description
	000 **SELECT_SD1_DATA2_ALT4** — Selecting Pad: SD1_DATA2 for Mode: ALT4 001 **SELECT_SD1_DATA3_ALT4** — Selecting Pad: SD1_DATA3 for Mode: ALT4 010 **SELECT_SD2_DATA0_ALT3** — Selecting Pad: SD2_DATA0 for Mode: ALT3 011 **SELECT_SD2_DATA1_ALT3** — Selecting Pad: SD2_DATA1 for Mode: ALT3 100 **SELECT_SAI3_TXFS_ALT4** — Selecting Pad: SAI3_TXFS for Mode: ALT4 101 **SELECT_SAI3_TXC_ALT4** — Selecting Pad: SAI3_TXC for Mode: ALT4 110 **SELECT_UART2_RXD_ALT0** — Selecting Pad: UART2_RXD for Mode: ALT0 111 **SELECT_UART2_TXD_ALT0** — Selecting Pad: UART2_TXD for Mode: ALT0

8.2.4.377 UART3_UART_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART3_UART_RTS_B_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5F4h offset = 3033_05F4h



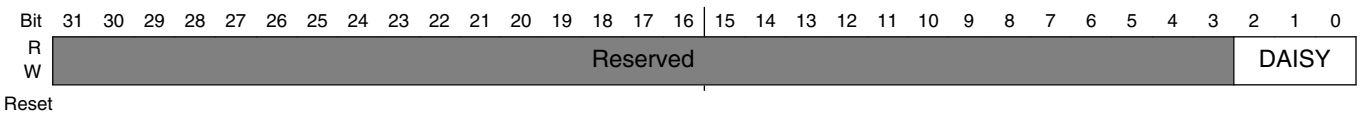
IOMUXC_UART3_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart3, In Pin: _uart_rts_b 00 **SELECT_SD1_RESET_B_ALT4** — Selecting Pad: SD1_RESET_B for Mode: ALT4 01 **SELECT_SD1_STROBE_ALT4** — Selecting Pad: SD1_STROBE for Mode: ALT4 10 **SELECT_ECSPI1_MISO_ALT1** — Selecting Pad: ECSPI1_MISO for Mode: ALT1 11 **SELECT_ECSPI1_SS0_ALT1** — Selecting Pad: ECSPI1_SS0 for Mode: ALT1

8.2.4.378 UART3_UART_RXD_MUX_SELECT_INPUT DAISY Register (IOMUXC_UART3_UART_RXD_MUX_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5F8h offset = 3033_05F8h



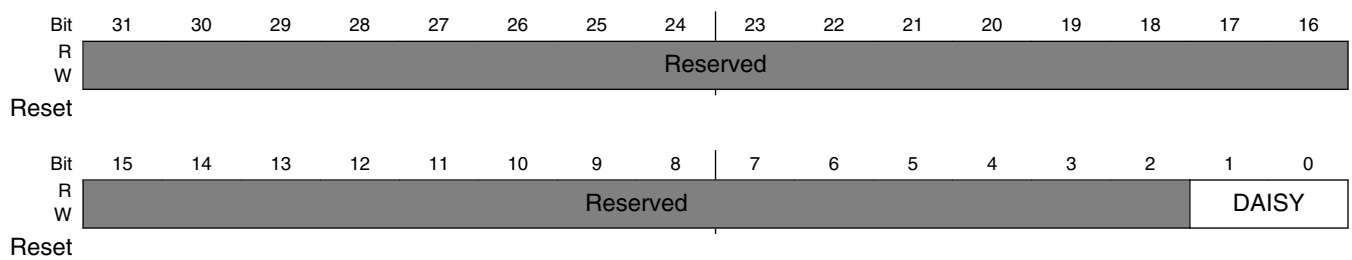
IOMUXC_UART3_UART_RXD_MUX_SELECT_INPUT field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart3, In Pin: _uart_rxd_mux 000 **SELECT_SD1_DATA6_ALT4** — Selecting Pad: SD1_DATA6 for Mode: ALT4 001 **SELECT_SD1_DATA7_ALT4** — Selecting Pad: SD1_DATA7 for Mode: ALT4 010 **SELECT_NAND_ALE_ALT4** — Selecting Pad: NAND_ALE for Mode: ALT4 011 **SELECT_NAND_CE0_B_ALT4** — Selecting Pad: NAND_CE0_B for Mode: ALT4 100 **SELECT_ECSPI1_SCLK_ALT1** — Selecting Pad: ECSPI1_SCLK for Mode: ALT1 101 **SELECT_ECSPI1_MOSI_ALT1** — Selecting Pad: ECSPI1_MOSI for Mode: ALT1 110 **SELECT_UART3_RXD_ALT0** — Selecting Pad: UART3_RXD for Mode: ALT0 111 **SELECT_UART3_TXD_ALT0** — Selecting Pad: UART3_TXD for Mode: ALT0

8.2.4.379 UART4_UART_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART4_UART_RTS_B_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 5FCh offset = 3033_05FCh



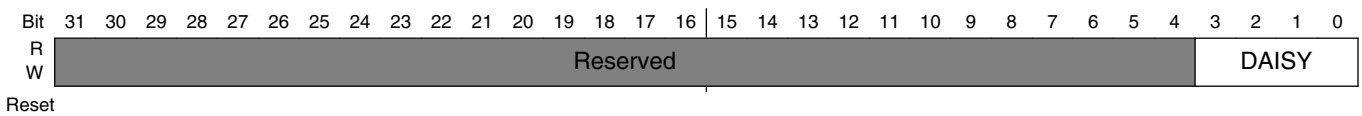
IOMUXC_UART4_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart4, In Pin: _uart_rts_b 00 **SELECT_NAND_DATA02_ALT3** — Selecting Pad: NAND_DATA02 for Mode: ALT3 01 **SELECT_NAND_DATA03_ALT3** — Selecting Pad: NAND_DATA03 for Mode: ALT3 10 **SELECT_ECSPI2_MISO_ALT1** — Selecting Pad: ECSPI2_MISO for Mode: ALT1 11 **SELECT_ECSPI2_SS0_ALT1** — Selecting Pad: ECSPI2_SS0 for Mode: ALT1

8.2.4.380 UART4_UART_RXD_MUX_SELECT_INPUT DAISY Register (IOMUXC_UART4_UART_RXD_MUX_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 600h offset = 3033_0600h



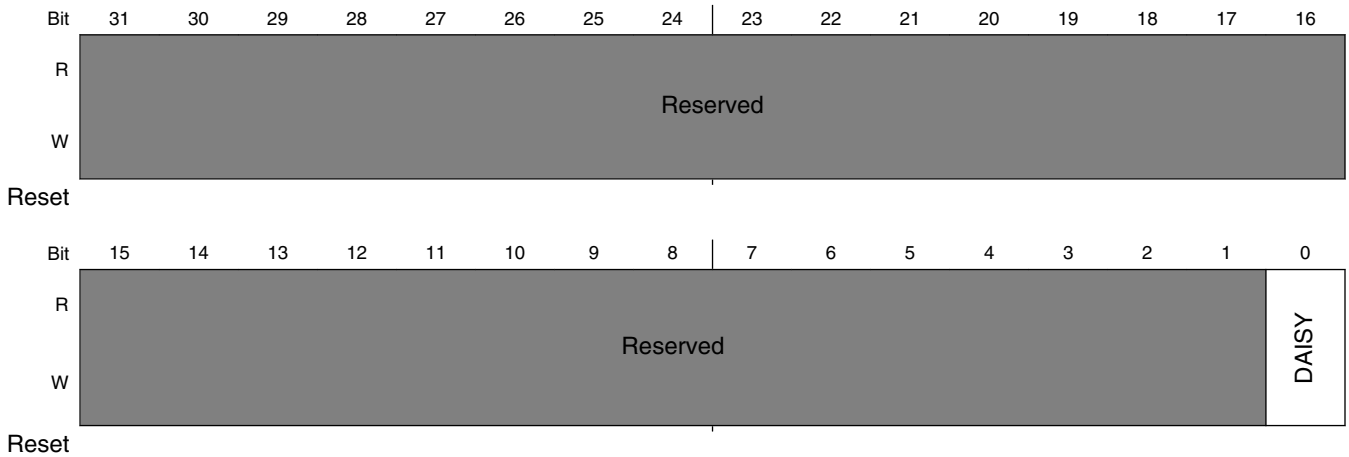
IOMUXC_UART4_UART_RXD_MUX_SELECT_INPUT field descriptions

Field	Description
31–4 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart4, In Pin: _uart_rxd_mux 0000 **SELECT_SD2_CLK_ALT3** — Selecting Pad: SD2_CLK for Mode: ALT3 0001 **SELECT_SD2_CMD_ALT3** — Selecting Pad: SD2_CMD for Mode: ALT3 0010 **SELECT_NAND_CLE_ALT4** — Selecting Pad: NAND_CLE for Mode: ALT4 0011 **SELECT_NAND_DATA00_ALT4** — Selecting Pad: NAND_DATA00 for Mode: ALT4 0100 **SELECT_NAND_DATA01_ALT4** — Selecting Pad: NAND_DATA01 for Mode: ALT4 0101 **SELECT_NAND_RE_B_ALT4** — Selecting Pad: NAND_RE_B for Mode: ALT4 0110 **SELECT_ECSPI2_SCLK_ALT1** — Selecting Pad: ECSPI2_SCLK for Mode: ALT1 0111 **SELECT_ECSPI2_MOSI_ALT1** — Selecting Pad: ECSPI2_MOSI for Mode: ALT1 1000 **SELECT_UART4_RXD_ALT0** — Selecting Pad: UART4_RXD for Mode: ALT0 1001 **SELECT_UART4_TXD_ALT0** — Selecting Pad: UART4_TXD for Mode: ALT0

8.2.4.381 USDHC3_CARD_CLK_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_CARD_CLK_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 604h offset = 3033_0604h



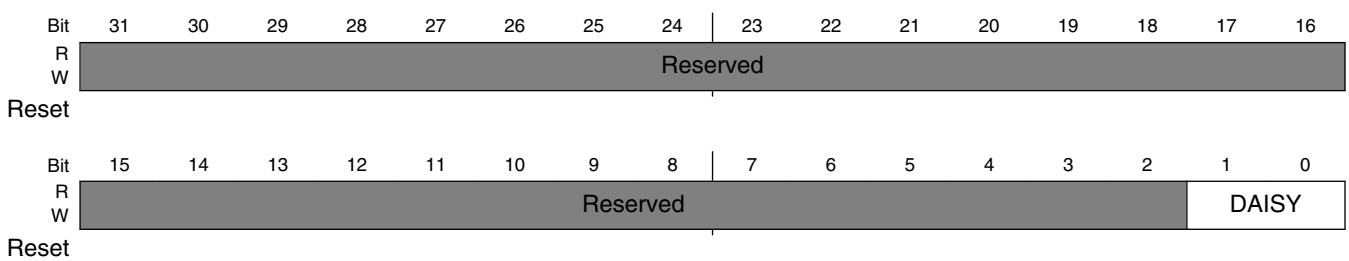
IOMUXC_USDHC3_CARD_CLK_IN_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: _card_clk_in 0 **SELECT_ENET_RD2_ALT6** — Selecting Pad: ENET_RD2 for Mode: ALT6 1 **SELECT_NAND_WE_B_ALT2** — Selecting Pad: NAND_WE_B for Mode: ALT2

8.2.4.382 USDHC3_CARD_DET_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_CARD_DET_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 608h offset = 3033_0608h



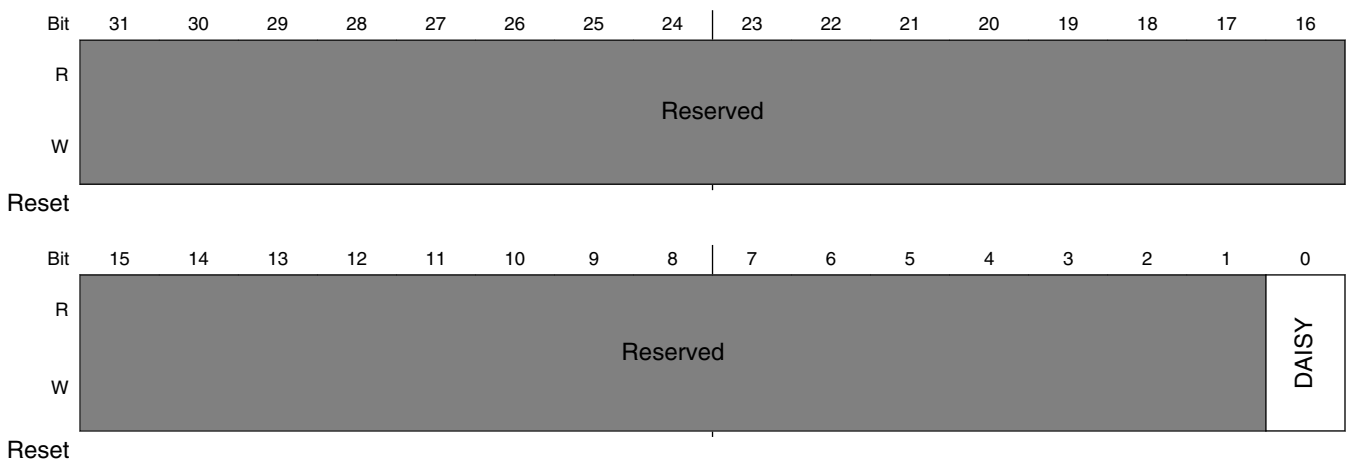
IOMUXC_USDHC3_CARD_DET_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: _card_det 00 **SELECT_GPIO1_IO14_ALT4** — Selecting Pad: GPIO1_IO14 for Mode: ALT4 01 **SELECT_ENET_TD1_ALT6** — Selecting Pad: ENET_TD1 for Mode: ALT6 10 **SELECT_NAND_DATA02_ALT2** — Selecting Pad: NAND_DATA02 for Mode: ALT2 11 **SELECT_I2C2_SCL_ALT2** — Selecting Pad: I2C2_SCL for Mode: ALT2

8.2.4.383 USDHC3_CMD_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_CMD_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 60Ch offset = 3033_060Ch



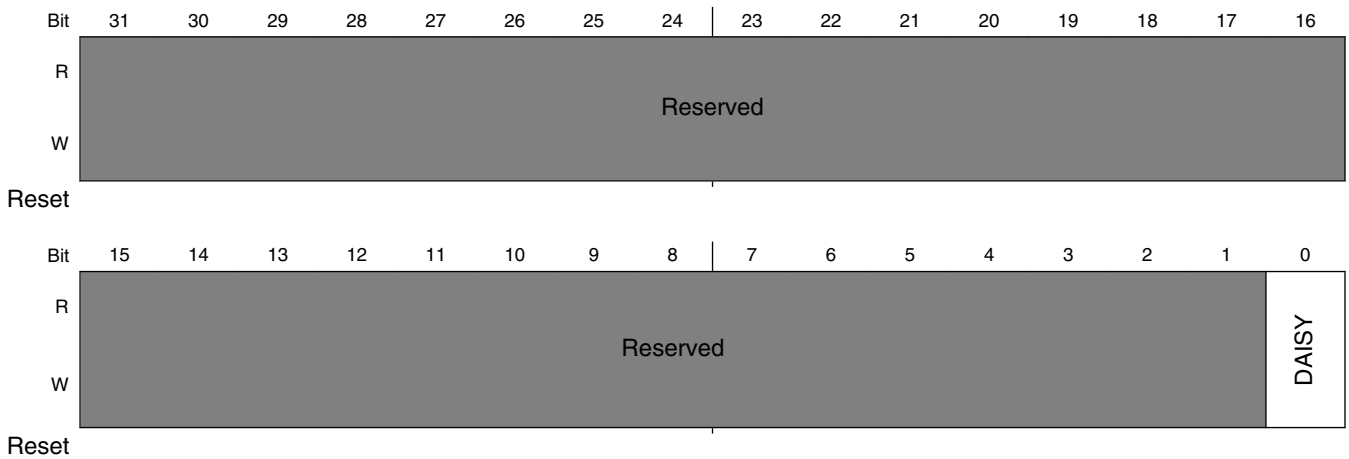
IOMUXC_USDHC3_CMD_IN_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: _cmd_in 0 **SELECT_ENET_RD3_ALT6** — Selecting Pad: ENET_RD3 for Mode: ALT6 1 **SELECT_NAND_WP_B_ALT2** — Selecting Pad: NAND_WP_B for Mode: ALT2

8.2.4.384 USDHC3_DAT0_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT0_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 610h offset = 3033_0610h



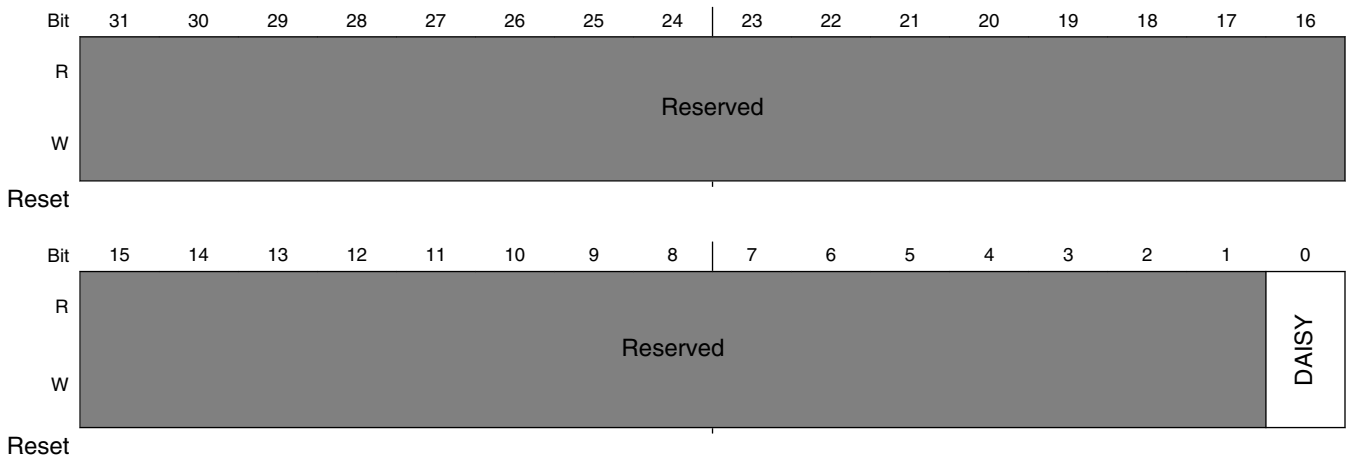
IOMUXC_USDHC3_DAT0_IN_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: _dat0_in 0 **SELECT_ENET_TX_CTL_ALT6** — Selecting Pad: ENET_TX_CTL for Mode: ALT6 1 **SELECT_NAND_DATA04_ALT2** — Selecting Pad: NAND_DATA04 for Mode: ALT2

8.2.4.385 USDHC3_DAT1_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT1_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 614h offset = 3033_0614h



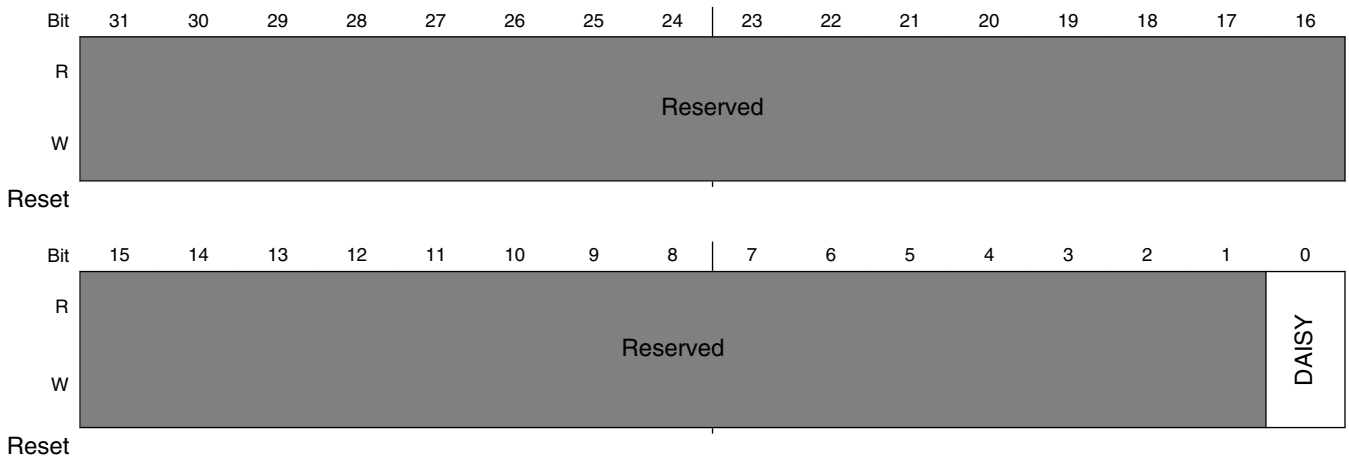
IOMUXC_USDHC3_DAT1_IN_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: _dat1_in 0 **SELECT_ENET_TXC_ALT6** — Selecting Pad: ENET_TXC for Mode: ALT6 1 **SELECT_NAND_DATA05_ALT2** — Selecting Pad: NAND_DATA05 for Mode: ALT2

8.2.4.386 USDHC3_DAT2_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT2_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 618h offset = 3033_0618h



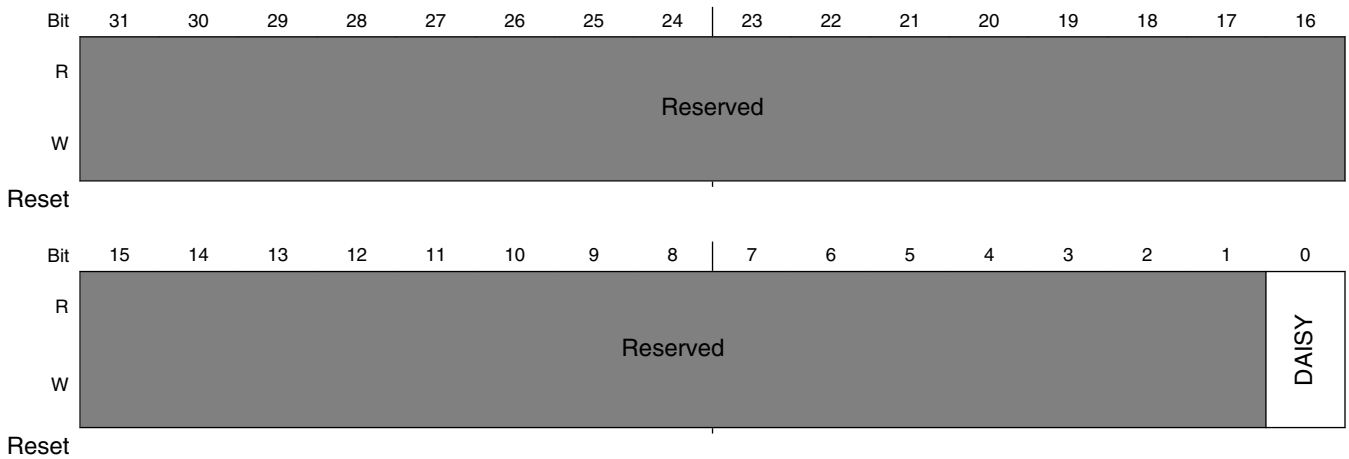
IOMUXC_USDHC3_DAT2_IN_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: _dat2_in 0 **SELECT_ENET_RX_CTL_ALT6** — Selecting Pad: ENET_RX_CTL for Mode: ALT6 1 **SELECT_NAND_DATA06_ALT2** — Selecting Pad: NAND_DATA06 for Mode: ALT2

8.2.4.387 USDHC3_DAT3_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT3_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 61Ch offset = 3033_061Ch



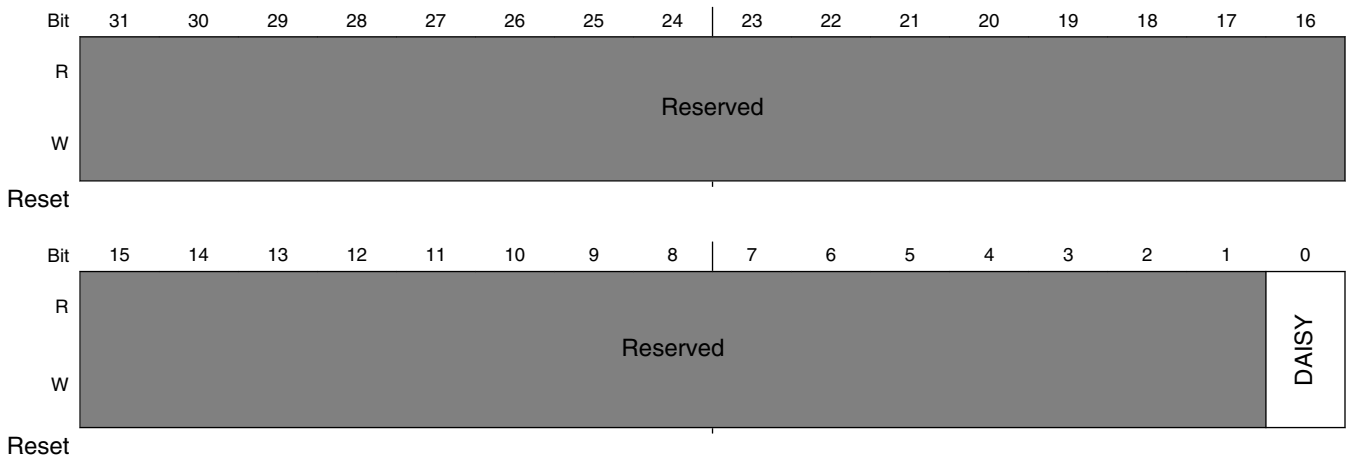
IOMUXC_USDHC3_DAT3_IN_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: _dat3_in 0 **SELECT_ENET_RXC_ALT6** — Selecting Pad: ENET_RXC for Mode: ALT6 1 **SELECT_NAND_DATA07_ALT2** — Selecting Pad: NAND_DATA07 for Mode: ALT2

8.2.4.388 USDHC3_DAT4_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT4_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 620h offset = 3033_0620h



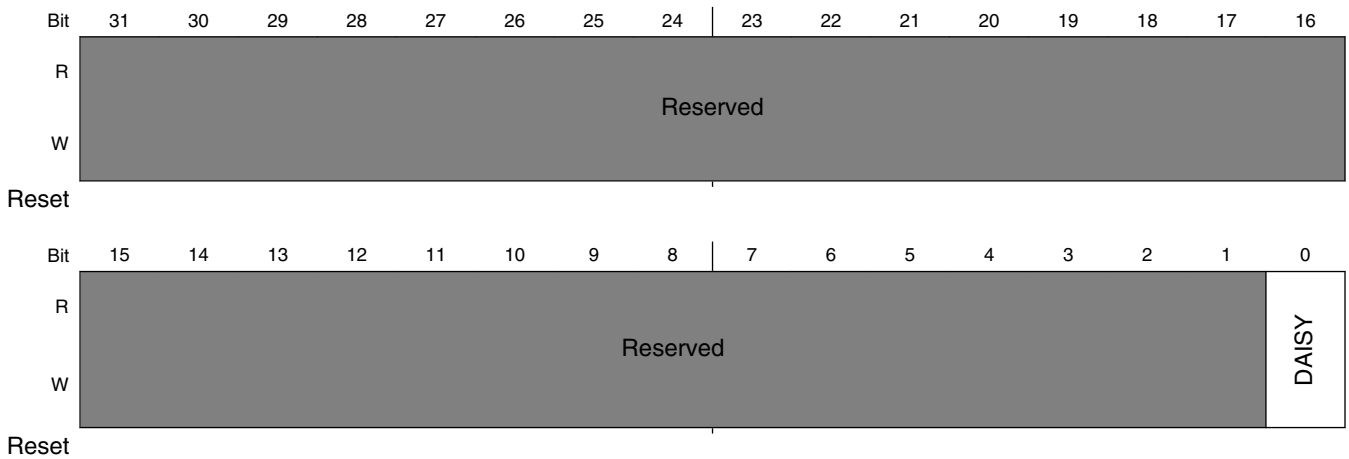
IOMUXC_USDHC3_DAT4_IN_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: _dat4_in 0 **SELECT_ENET_RD0_ALT6** — Selecting Pad: ENET_RD0 for Mode: ALT6 1 **SELECT_NAND_RE_B_ALT2** — Selecting Pad: NAND_RE_B for Mode: ALT2

8.2.4.389 USDHC3_DAT5_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT5_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 624h offset = 3033_0624h



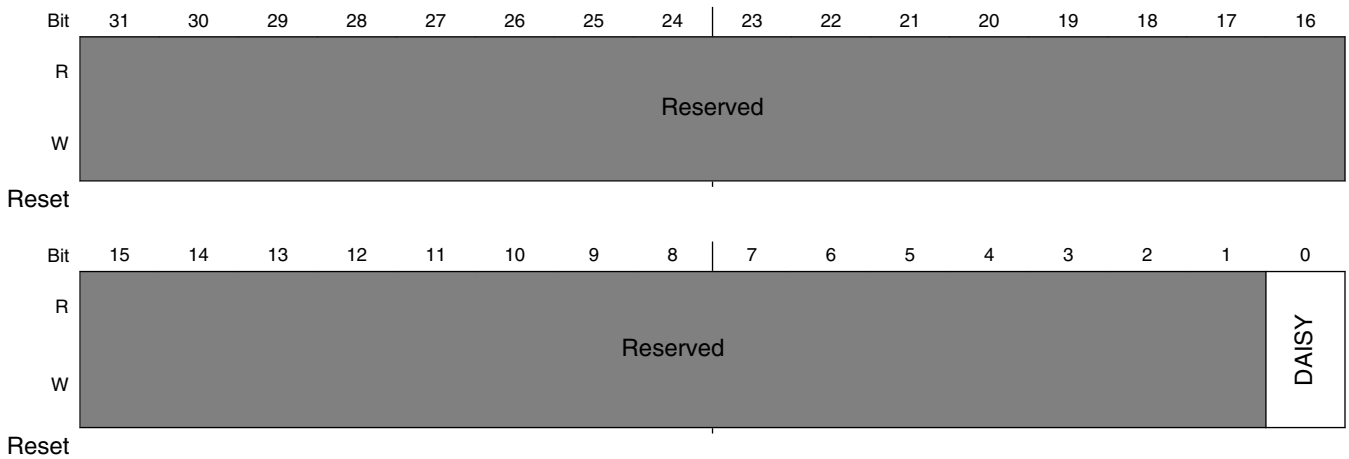
IOMUXC_USDHC3_DAT5_IN_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: _dat5_in 0 **SELECT_ENET_MDIO_ALT6** — Selecting Pad: ENET_MDIO for Mode: ALT6 1 **SELECT_NAND_CE2_B_ALT2** — Selecting Pad: NAND_CE2_B for Mode: ALT2

8.2.4.390 USDHC3_DAT6_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT6_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 628h offset = 3033_0628h



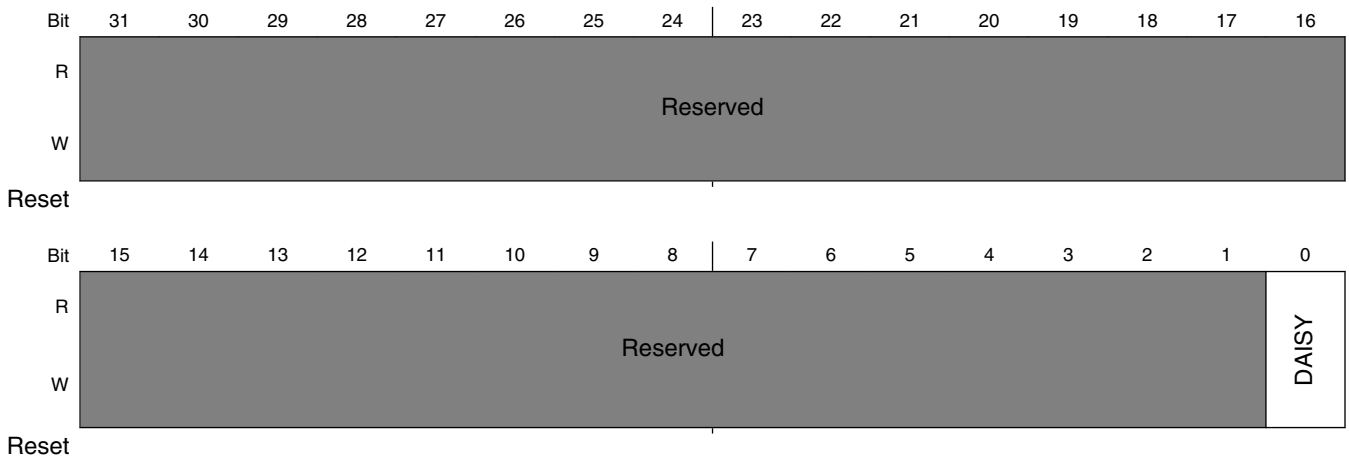
IOMUXC_USDHC3_DAT6_IN_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: _dat6_in 0 **SELECT_ENET_TD3_ALT6** — Selecting Pad: ENET_TD3 for Mode: ALT6 1 **SELECT_NAND_CE3_B_ALT2** — Selecting Pad: NAND_CE3_B for Mode: ALT2

8.2.4.391 USDHC3_DAT7_IN_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_DAT7_IN_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 62Ch offset = 3033_062Ch



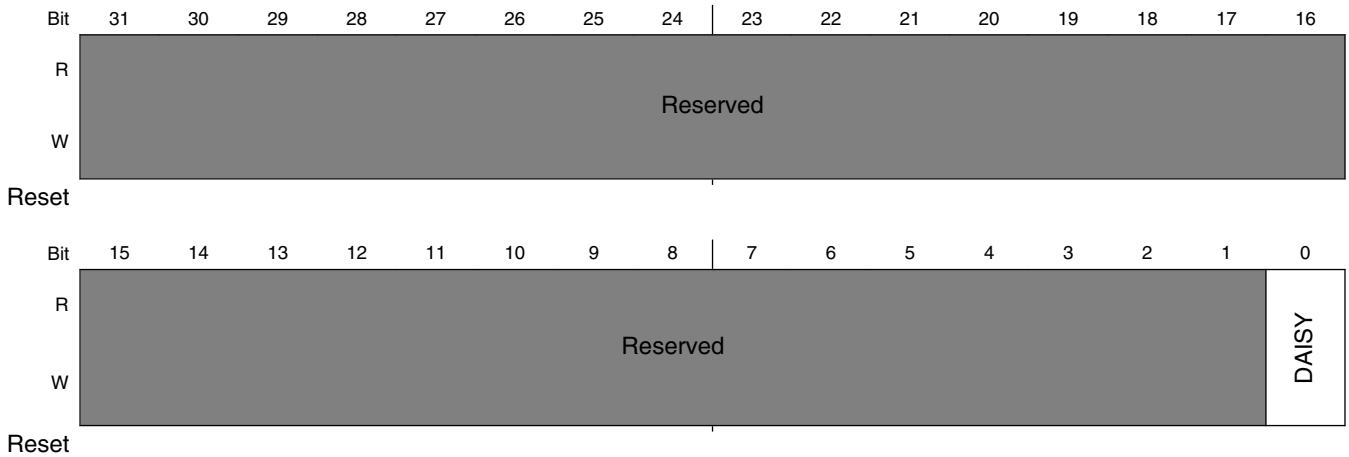
IOMUXC_USDHC3_DAT7_IN_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: _dat7_in 0 **SELECT_ENET_TD2_ALT6** — Selecting Pad: ENET_TD2 for Mode: ALT6 1 **SELECT_NAND_CLE_ALT2** — Selecting Pad: NAND_CLE for Mode: ALT2

8.2.4.392 USDHC3_STROBE_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_STROBE_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 630h offset = 3033_0630h



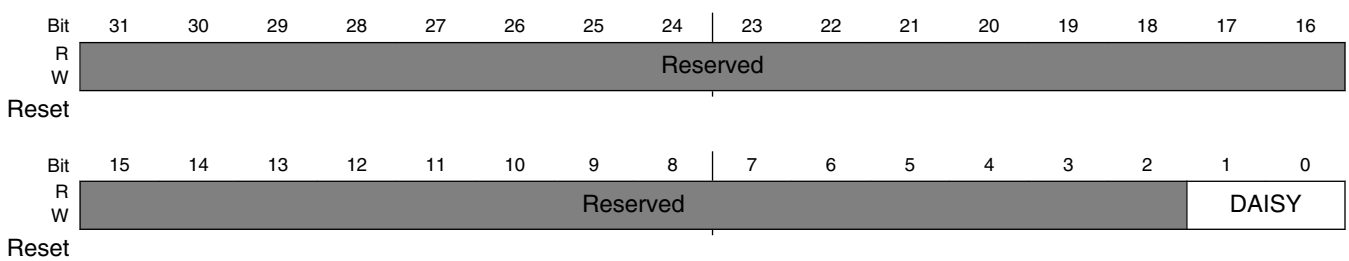
IOMUXC_USDHC3_STROBE_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: _strobe 0 **SELECT_ENET_MDC_ALT6** — Selecting Pad: ENET_MDC for Mode: ALT6 1 **SELECT_NAND_CE1_B_ALT2** — Selecting Pad: NAND_CE1_B for Mode: ALT2

8.2.4.393 USDHC3_WP_ON_SELECT_INPUT DAISY Register (IOMUXC_USDHC3_WP_ON_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 634h offset = 3033_0634h



IOMUXC_USDHC3_WP_ON_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc3, In Pin: _wp_on 00 **SELECT_GPIO1_IO15_ALT4** — Selecting Pad: GPIO1_IO15 for Mode: ALT4 01 **SELECT_ENET_TD0_ALT6** — Selecting Pad: ENET_TD0 for Mode: ALT6 10 **SELECT_NAND_DATA03_ALT2** — Selecting Pad: NAND_DATA03 for Mode: ALT2 11 **SELECT_I2C2_SDA_ALT2** — Selecting Pad: I2C2_SDA for Mode: ALT2

8.3 General Purpose Input/Output (GPIO)

8.3.1 Overview

The General-Purpose Input/Output (GPIO) peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs.

When configured as an output, it is possible to write to an internal register to control the state driven on the output pin. When configured as an input, it is possible to detect the state of the input by reading the state of an internal register. In addition, the GPIO peripheral can produce CORE interrupts.

8.3.1.1 Block Diagram

The GPIO subsystem contains multiple GPIO blocks, which can generate and control up to 32 signals for general purpose.

The block diagram for the GPIO block is provided below.

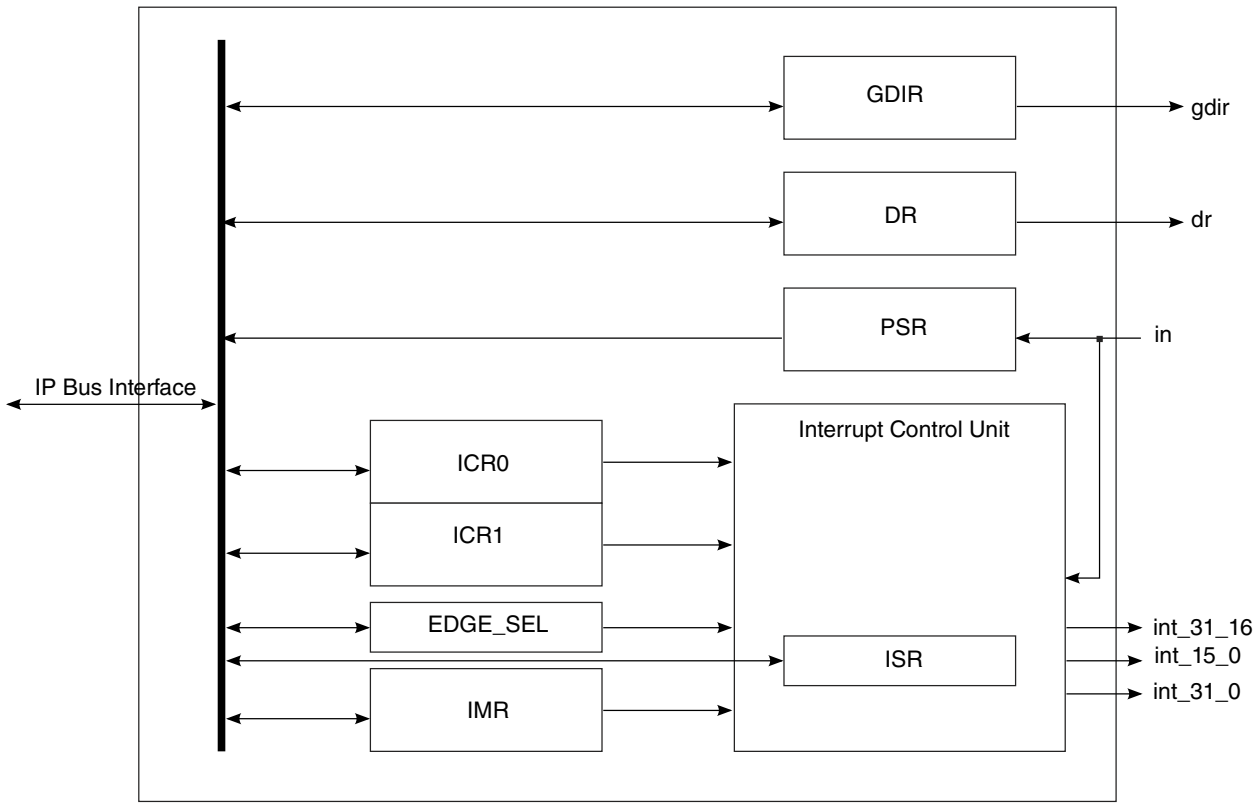


Figure 8-4. GPIO Block Diagram

8.3.1.2 Features

The GPIO includes the following features:

- General purpose input/output logic capabilities:
  - Drives specific data to output using the data register (DR)
  - Controls the direction of the signal using the GPIO direction register (GDIR)
  - Enables the core to sample the status of the corresponding inputs by reading the pad sample register (PSR).
- GPIO interrupt capabilities:
  - Supports 32 interrupts
  - Identifies interrupt edges
  - Generates three active-high interrupts to the SoC interrupt controller

8.3.2 Functional Description

The GPIO is one of the blocks controlling the IOMUX of the chip.

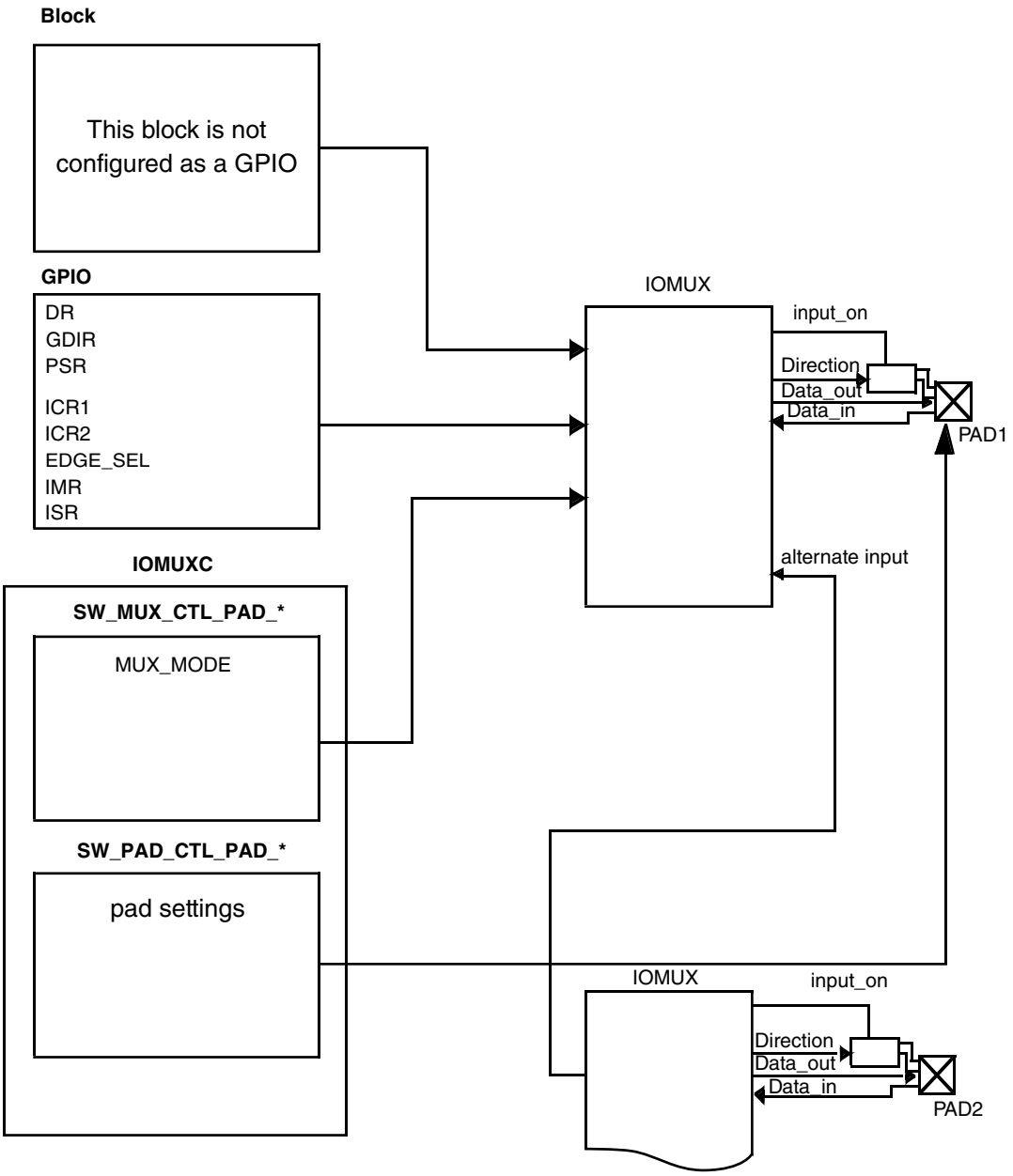


Figure 8-5. Chip IOMUX Scheme

A GPIO signal can operate as a general-purpose input/output when the IOMUX is set to GPIO mode. Each GPIO signal may be independently configured as either an input or an output using the GPIO direction register (GDIR).

When configured as an output (GDIR[GDIR] = 1), the value in the data bit in the GPIO data register (DR) is driven on the corresponding GPIO line. When a signal is configured as an input (GDIR[GDIR] = 0), the state of the input can be read from the corresponding PSR[PSR] bit.

The GPIO functionality is provided through eight registers, an edge-detect circuit, and interrupt generation logic.

The eight registers are:

- Data register (DR)
- GPIO direction register (GDIR)
- Pad sample register (PSR)
- Interrupt control registers (ICR1, ICR2)
- Edge select register (EDGE_SEL)
- Interrupt mask register (IMR)
- Interrupt status register (ISR)

The above registers are described in detail in the Register section.

Each GPIO input has a dedicated edge-detect circuit which can be configured through software to detect rising edges, falling edges, logic low-levels or logic high-levels on the input signals. The outputs of the edge detect circuits are optionally masked by setting the corresponding bit in the interrupt mask register (IMR). These qualified outputs are OR'ed together to generate two one-bit interrupt lines:

- Combined interrupt indication for GPIOx signals 0 - 15
- Combined interrupt indication for GPIOx signals 16 - 31

In addition, GPIO1 provides visibility to each of its 8 low order interrupt sources (i.e. GPIO1 interrupt n, for n = 0 – 7). However, individual interrupt indications from other GPIOx are not available.

The GPIO edge detection is described further in Interrupt Control Unit.

8.3.2.1 Interrupt Control Unit

In addition to the general-purpose input/output function, the edge-detect logic in the GPIO peripheral reflects whether a transition has occurred on a given GPIO signal that is configured as an input (GDIR[GDIR] = 0). The interrupt control registers (ICR1 and ICR2) may be used to independently configure the interrupt condition of each input signal (low-to-high transition, high-to-low transition, low, or high). For information about ICR1 and ICR2 settings, see the Register section.

The interrupt control unit is built of 32 interrupt control subunits, where each subunit handles a single interrupt line.

8.3.2.2 Clocks

The following table describes the clock sources for GPIO. Please see the CCM for clock setting, configuration and gating information.

Table 8-6. GPIO Clocks

Clock name	Description
ipg_clk_s	Peripheral access clock

8.3.3 External Signals

The following table describes the external signals of GPIO.

Table 8-7. GPIO External Signals

Signal	Description
IOn	GPIO Signal

8.3.4 Application Information

8.3.4.1 GPIO Read Mode

The programming sequence for reading input signals should be as follows:

1. Configure IOMUX to select GPIO mode (via IOMUX Controller (IOMUXC)).
2. Configure GPIO direction register to input (GDIR[GDIR] = 0b).
3. Read value from data register/pad status register.

A pseudocode description to read [input3:input0] values is as follows:

```
// SET INPUTS TO GPIO MODE.
write sw_mux_ctl_<input0>_<input1>_<input2>_<input3>, 32'h00000000
// SET GDIR TO INPUT.
write GDIR[31:4,input3_bit, input2_bit, input1_bit, input0_bit,] 32'hxxxxxxx0
// READ INPUT VALUE FROM DR.
read DR
// READ INPUT VALUE FROM PSR.
read PSR
```

NOTE

While the GPIO direction is set to input (GDIR[GDIR] = 0), a read access to DR does not return DR data. Instead, it returns the PSR data, which is the corresponding input signal value.

8.3.4.2 GPIO Write Mode

The programming sequence for driving output signals should be as follows:

1. Configure IOMUX to select GPIO mode (via IOMUXC). If needed, enable SION to read the loopback pad value through PSR.
2. Configure GPIO direction register to output (GDIR[GDIR] = 1b).
3. Write value to data register (DR).

A pseudocode description to drive 4'b0101 on [output3:output0] is as follows:

```
// SET PADS TO GPIO MODE VIA IOMUX.
write sw_mux_ctl_pad_<output[0-3]>.mux_mode, <GPIO_MUX_MODE>
// Enable loopback to capture pad value into PSR in output mode
write sw_mux_ctl_pad_<output[0-3]>.sion, 1
// SET GDIR=1 TO OUTPUT BITS.
write GDIR[31:4,output3_bit,output2_bit, output1_bit, output0_bit,] 32'hxxxxxxxF
// WRITE OUTPUT VALUE=4'b0101 TO DR.
write DR, 32'hxxxxxxx5
// READ OUTPUT VALUE FROM PSR ONLY.
read_cmp PSR, 32'hxxxxxxx5
```

8.3.5 Memory Map/Register Definition

This section includes the memory map and descriptions of all registers.

8.3.5.1 GPIO register descriptions

There are eight 32-bit GPIO registers. All registers are accessible from the IP interface. Only 32-bit access is supported.

The GPIO memory map is shown in the following table.

8.3.5.1.1 GPIO memory map

GPIO1 base address: 3020_0000h

GPIO2 base address: 3021_0000h

GPIO3 base address: 3022_0000h

GPIO4 base address: 3023_0000h

GPIO5 base address: 3024_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	GPIO data register (DR)	32	RW	0000_0000h
4h	GPIO direction register (GDIR)	32	RW	0000_0000h
8h	GPIO pad status register (PSR)	32	RO	0000_0000h
Ch	GPIO interrupt configuration register1 (ICR1)	32	RW	0000_0000h
10h	GPIO interrupt configuration register2 (ICR2)	32	RW	0000_0000h
14h	GPIO interrupt mask register (IMR)	32	RW	0000_0000h
18h	GPIO interrupt status register (ISR)	32	W1C	0000_0000h
1Ch	GPIO edge select register (EDGE_SEL)	32	RW	0000_0000h

8.3.5.1.2 GPIO data register (DR)

8.3.5.1.2.1 Offset

Register	Offset
DR	0h

8.3.5.1.2.2 Function

The 32-bit DR register stores data that is ready to be driven to the output lines. If the IOMUXC is in GPIO mode and a given GPIO direction bit is set, then the corresponding DR bit is driven to the output. If a given GPIO direction bit is cleared, then a read of DR register reflects the value of the corresponding signal. Two wait states are required in read access for synchronization.

The results of a read of a DR bit depends on the IOMUXC input mode settings and the corresponding GDIR bit as follows:

- If GDIR[*n*] is set and IOMUXC input mode is GPIO, then reading DR[*n*] returns the contents of DR[*n*].
- If GDIR[*n*] is cleared and IOMUXC input mode is GPIO, then reading DR[*n*] returns the corresponding input signal's value.

- If GDIR[*n*] is set and IOMUXC input mode is not GPIO, then reading DR[*n*] returns the contents of DR[*n*].
- If GDIR[*n*] is cleared and IOMUXC input mode is not GPIO, then reading DR[*n*] always returns zero.

8.3.5.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.5.1.2.4 Fields

Field	Function
31-0 DR	DR data bits This register defines the value of the GPIO output when the signal is configured as an output (GDIR[n]=1). Writes to this register are stored in a register. Reading DR register returns the value stored in the register if the signal is configured as an output (GDIR[n]=1), or the input signal's value if configured as an input (GDIR[n]=0). **NOTE:** The IOMUXC must be configured to GPIO mode for the DR register value to connect with the signal. Reading the data register with the input path disabled always returns a zero value.

8.3.5.1.3 GPIO direction register (GDIR)

8.3.5.1.3.1 Offset

Register	Offset
GDIR	4h

8.3.5.1.3.2 Function

The GDIR register functions as direction control when the IOMUXC is in GPIO mode. Each bit specifies the direction of a one-bit signal. The mapping of each DIR bit to a corresponding SoC signal is determined by the SoC's pin assignment and the IOMUX table. For more details consult the IOMUXC chapter.

8.3.5.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GDIR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GDIR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.5.1.3.4 Fields

Field	Function
31-0 GDIR	GPIO direction bits Bit n of this register defines the direction of the GPIO[n] signal. **NOTE:** The GDIR register affects only the direction of the I/O signal when the corresponding bit in the IOMUXC is configured for GPIO. 0b - GPIO is configured as input. 1b - GPIO is configured as output.

8.3.5.1.4 GPIO pad status register (PSR)

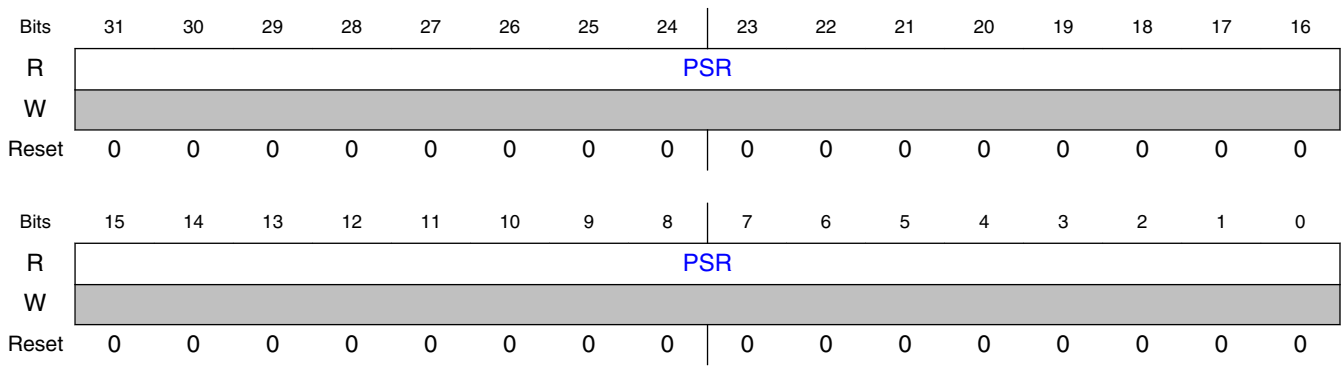
8.3.5.1.4.1 Offset

Register	Offset
PSR	8h

8.3.5.1.4.2 Function

The PSR register is a read-only register. Each bit stores the value of the corresponding input signal (as configured in the IOMUX). This register is clocked with the ipg_clk_s clock, meaning that the input signal is sampled only when accessing this location. Two wait states are required any time this register is accessed for synchronization.

8.3.5.1.4.3 Diagram



8.3.5.1.4.4 Fields

Field	Function
31-0 PSR	GPIO pad status bits Reading the PSR register returns the state of the corresponding input signal. **NOTE:** The IOMUXC must be configured to GPIO mode for the PSR register to reflect the state of the corresponding signal.

8.3.5.1.5 GPIO interrupt configuration register1 (ICR1)

8.3.5.1.5.1 Offset

Register	Offset
ICR1	Ch

8.3.5.1.5.2 Function

The ICR1 register contains 16 two-bit fields, where each field specifies the interrupt configuration for a different input signal.

8.3.5.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ICR15		ICR14		ICR13		ICR12		ICR11		ICR10		ICR9		ICR8	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ICR7		ICR6		ICR5		ICR4		ICR3		ICR2		ICR1		ICR0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.5.1.5.4 Fields

Field	Function
31-30 ICR15	Interrupt configuration field for GPIO interrupt 15 This bit field controls the active condition of the interrupt function for GPIO interrupt 15. 00b - Interrupt 15 is low-level sensitive. 01b - Interrupt 15 is high-level sensitive. 10b - Interrupt 15 is rising-edge sensitive. 11b - Interrupt 15 is falling-edge sensitive.
29-28 ICR14	Interrupt configuration field for GPIO interrupt 14 This bit field controls the active condition of the interrupt function for GPIO interrupt 14. 00b - Interrupt 14 is low-level sensitive. 01b - Interrupt 14 is high-level sensitive. 10b - Interrupt 14 is rising-edge sensitive. 11b - Interrupt 14 is falling-edge sensitive.
27-26 ICR13	Interrupt configuration field for GPIO interrupt 13 This bit field controls the active condition of the interrupt function for GPIO interrupt 13. 00b - Interrupt 13 is low-level sensitive. 01b - Interrupt 13 is high-level sensitive. 10b - Interrupt 13 is rising-edge sensitive. 11b - Interrupt 13 is falling-edge sensitive.
25-24 ICR12	Interrupt configuration field for GPIO interrupt 12 This bit field controls the active condition of the interrupt function for GPIO interrupt 12. 00b - Interrupt 12 is low-level sensitive. 01b - Interrupt 12 is high-level sensitive. 10b - Interrupt 12 is rising-edge sensitive. 11b - Interrupt 12 is falling-edge sensitive.
23-22 ICR11	Interrupt configuration field for GPIO interrupt 11 This bit field controls the active condition of the interrupt function for GPIO interrupt 11. 00b - Interrupt 11 is low-level sensitive. 01b - Interrupt 11 is high-level sensitive. 10b - Interrupt 11 is rising-edge sensitive.

Table continues on the next page...

Field	Function
	11b - Interrupt 11 is falling-edge sensitive.
21-20 ICR10	Interrupt configuration field for GPIO interrupt 10 This bit field controls the active condition of the interrupt function for GPIO interrupt 10. 00b - Interrupt 10 is low-level sensitive. 01b - Interrupt 10 is high-level sensitive. 10b - Interrupt 10 is rising-edge sensitive. 11b - Interrupt 10 is falling-edge sensitive.
19-18 ICR9	Interrupt configuration field for GPIO interrupt 9 This bit field controls the active condition of the interrupt function for GPIO interrupt 9. 00b - Interrupt 9 is low-level sensitive. 01b - Interrupt 9 is high-level sensitive. 10b - Interrupt 9 is rising-edge sensitive. 11b - Interrupt 9 is falling-edge sensitive.
17-16 ICR8	Interrupt configuration field for GPIO interrupt 8 This bit field controls the active condition of the interrupt function for GPIO interrupt 8. 00b - Interrupt 8 is low-level sensitive. 01b - Interrupt 8 is high-level sensitive. 10b - Interrupt 8 is rising-edge sensitive. 11b - Interrupt 8 is falling-edge sensitive.
15-14 ICR7	Interrupt configuration field for GPIO interrupt 7 This bit field controls the active condition of the interrupt function for GPIO interrupt 7. 00b - Interrupt 7 is low-level sensitive. 01b - Interrupt 7 is high-level sensitive. 10b - Interrupt 7 is rising-edge sensitive. 11b - Interrupt 7 is falling-edge sensitive.
13-12 ICR6	Interrupt configuration field for GPIO interrupt 6 This bit field controls the active condition of the interrupt function for GPIO interrupt 6. 00b - Interrupt 6 is low-level sensitive. 01b - Interrupt 6 is high-level sensitive. 10b - Interrupt 6 is rising-edge sensitive. 11b - Interrupt 6 is falling-edge sensitive.
11-10 ICR5	Interrupt configuration field for GPIO interrupt 5 This bit field controls the active condition of the interrupt function for GPIO interrupt 5. 00b - Interrupt 5 is low-level sensitive. 01b - Interrupt 5 is high-level sensitive. 10b - Interrupt 5 is rising-edge sensitive. 11b - Interrupt 5 is falling-edge sensitive.
9-8 ICR4	Interrupt configuration field for GPIO interrupt 4 This bit field controls the active condition of the interrupt function for GPIO interrupt 4. 00b - Interrupt 4 is low-level sensitive. 01b - Interrupt 4 is high-level sensitive. 10b - Interrupt 4 is rising-edge sensitive. 11b - Interrupt 4 is falling-edge sensitive.
7-6 ICR3	Interrupt configuration field for GPIO interrupt 3 This bit field controls the active condition of the interrupt function for GPIO interrupt 3. 00b - Interrupt 3 is low-level sensitive.

Table continues on the next page...

Field	Function
	01b - Interrupt 3 is high-level sensitive. 10b - Interrupt 3 is rising-edge sensitive. 11b - Interrupt 3 is falling-edge sensitive.
5-4 ICR2	Interrupt configuration field for GPIO interrupt 2 This bit field controls the active condition of the interrupt function for GPIO interrupt 2. 00b - Interrupt 2 is low-level sensitive. 01b - Interrupt 2 is high-level sensitive. 10b - Interrupt 2 is rising-edge sensitive. 11b - Interrupt 2 is falling-edge sensitive.
3-2 ICR1	Interrupt configuration field for GPIO interrupt 1 This bit field controls the active condition of the interrupt function for GPIO interrupt 1. 00b - Interrupt 1 is low-level sensitive. 01b - Interrupt 1 is high-level sensitive. 10b - Interrupt 1 is rising-edge sensitive. 11b - Interrupt 1 is falling-edge sensitive.
1-0 ICR0	Interrupt configuration field for GPIO interrupt 0 This bit field controls the active condition of the interrupt function for GPIO interrupt 0. 00b - Interrupt 0 is low-level sensitive. 01b - Interrupt 0 is high-level sensitive. 10b - Interrupt 0 is rising-edge sensitive. 11b - Interrupt 0 is falling-edge sensitive.

8.3.5.1.6 GPIO interrupt configuration register2 (ICR2)

8.3.5.1.6.1 Offset

Register	Offset
ICR2	10h

8.3.5.1.6.2 Function

The ICR2 register contains 16 two-bit fields, where each field specifies the interrupt configuration for a different input signal.

8.3.5.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ICR31		ICR30		ICR29		ICR28		ICR27		ICR26		ICR25		ICR24	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ICR23		ICR22		ICR21		ICR20		ICR19		ICR18		ICR17		ICR16	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.5.1.6.4 Fields

Field	Function
31-30 ICR31	Interrupt configuration field for GPIO interrupt 31 This bit field controls the active condition of the interrupt function for GPIO interrupt 31. 00b - Interrupt 31 is low-level sensitive. 01b - Interrupt 31 is high-level sensitive. 10b - Interrupt 31 is rising-edge sensitive. 11b - Interrupt 31 is falling-edge sensitive.
29-28 ICR30	Interrupt configuration field for GPIO interrupt 30 This bit field controls the active condition of the interrupt function for GPIO interrupt 30. 00b - Interrupt 30 is low-level sensitive. 01b - Interrupt 30 is high-level sensitive. 10b - Interrupt 30 is rising-edge sensitive. 11b - Interrupt 30 is falling-edge sensitive.
27-26 ICR29	Interrupt configuration field for GPIO interrupt 29 This bit field controls the active condition of the interrupt function for GPIO interrupt 29. 00b - Interrupt 29 is low-level sensitive. 01b - Interrupt 29 is high-level sensitive. 10b - Interrupt 29 is rising-edge sensitive. 11b - Interrupt 29 is falling-edge sensitive.
25-24 ICR28	Interrupt configuration field for GPIO interrupt 28 This bit field controls the active condition of the interrupt function for GPIO interrupt 28. 00b - Interrupt 28 is low-level sensitive. 01b - Interrupt 28 is high-level sensitive. 10b - Interrupt 28 is rising-edge sensitive. 11b - Interrupt 28 is falling-edge sensitive.
23-22 ICR27	Interrupt configuration field for GPIO interrupt 27 This bit field controls the active condition of the interrupt function for GPIO interrupt 27. 00b - Interrupt 27 is low-level sensitive. 01b - Interrupt 27 is high-level sensitive. 10b - Interrupt 27 is rising-edge sensitive. 11b - Interrupt 27 is falling-edge sensitive.

Table continues on the next page...

Field	Function
21-20	Interrupt configuration field for GPIO interrupt 26
ICR26	This bit field controls the active condition of the interrupt function for GPIO interrupt 26. 00b - Interrupt 26 is low-level sensitive. 01b - Interrupt 26 is high-level sensitive. 10b - Interrupt 26 is rising-edge sensitive. 11b - Interrupt 26 is falling-edge sensitive.
19-18	Interrupt configuration field for GPIO interrupt 25
ICR25	This bit field controls the active condition of the interrupt function for GPIO interrupt 25. 00b - Interrupt 25 is low-level sensitive. 01b - Interrupt 25 is high-level sensitive. 10b - Interrupt 25 is rising-edge sensitive. 11b - Interrupt 25 is falling-edge sensitive.
17-16	Interrupt configuration field for GPIO interrupt 24
ICR24	This bit field controls the active condition of the interrupt function for GPIO interrupt 24. 00b - Interrupt 24 is low-level sensitive. 01b - Interrupt 24 is high-level sensitive. 10b - Interrupt 24 is rising-edge sensitive. 11b - Interrupt 24 is falling-edge sensitive.
15-14	Interrupt configuration field for GPIO interrupt 23
ICR23	This bit field controls the active condition of the interrupt function for GPIO interrupt 23. 00b - Interrupt 23 is low-level sensitive. 01b - Interrupt 23 is high-level sensitive. 10b - Interrupt 23 is rising-edge sensitive. 11b - Interrupt 23 is falling-edge sensitive.
13-12	Interrupt configuration field for GPIO interrupt 22
ICR22	This bit field controls the active condition of the interrupt function for GPIO interrupt 22. 00b - Interrupt 22 is low-level sensitive. 01b - Interrupt 22 is high-level sensitive. 10b - Interrupt 22 is rising-edge sensitive. 11b - Interrupt 22 is falling-edge sensitive.
11-10	Interrupt configuration field for GPIO interrupt 21
ICR21	This bit field controls the active condition of the interrupt function for GPIO interrupt 21. 00b - Interrupt 21 is low-level sensitive. 01b - Interrupt 21 is high-level sensitive. 10b - Interrupt 21 is rising-edge sensitive. 11b - Interrupt 21 is falling-edge sensitive.
9-8	Interrupt configuration field for GPIO interrupt 20
ICR20	This bit field controls the active condition of the interrupt function for GPIO interrupt 20. 00b - Interrupt 20 is low-level sensitive. 01b - Interrupt 20 is high-level sensitive. 10b - Interrupt 20 is rising-edge sensitive. 11b - Interrupt 20 is falling-edge sensitive.
7-6	Interrupt configuration field for GPIO interrupt 19
ICR19	This bit field controls the active condition of the interrupt function for GPIO interrupt 19. 00b - Interrupt 19 is low-level sensitive. 01b - Interrupt 19 is high-level sensitive.

Table continues on the next page...

Field	Function
	10b - Interrupt 19 is rising-edge sensitive. 11b - Interrupt 19 is falling-edge sensitive.
5-4 ICR18	Interrupt configuration field for GPIO interrupt 18 This bit field controls the active condition of the interrupt function for GPIO interrupt 18. 00b - Interrupt 18 is low-level sensitive. 01b - Interrupt 18 is high-level sensitive. 10b - Interrupt 18 is rising-edge sensitive. 11b - Interrupt 18 is falling-edge sensitive.
3-2 ICR17	Interrupt configuration field for GPIO interrupt 17 This bit field controls the active condition of the interrupt function for GPIO interrupt 17. 00b - Interrupt 17 is low-level sensitive. 01b - Interrupt 17 is high-level sensitive. 10b - Interrupt 17 is rising-edge sensitive. 11b - Interrupt 17 is falling-edge sensitive.
1-0 ICR16	Interrupt configuration field for GPIO interrupt 16 This bit field controls the active condition of the interrupt function for GPIO interrupt 16. 00b - Interrupt 16 is low-level sensitive. 01b - Interrupt 16 is high-level sensitive. 10b - Interrupt 16 is rising-edge sensitive. 11b - Interrupt 16 is falling-edge sensitive.

8.3.5.1.7 GPIO interrupt mask register (IMR)

8.3.5.1.7.1 Offset

Register	Offset
IMR	14h

8.3.5.1.7.2 Function

The IMR register contains masking bits for each interrupt line.

8.3.5.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IMR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.5.1.7.4 Fields

Field	Function
31-0 IMR	Interrupt Mask bits This register is used to enable or disable the interrupt function on each of the 32 GPIO signals. Bit IMR[n] (n=0...31) controls interrupt n as follows: 0b - Interrupt n is disabled. 1b - Interrupt n is enabled.

8.3.5.1.8 GPIO interrupt status register (ISR)

8.3.5.1.8.1 Offset

Register	Offset
ISR	18h

8.3.5.1.8.2 Function

The ISR register functions as an interrupt status indicator. Each bit indicates whether an interrupt condition has been met for the corresponding input signal. When an interrupt condition is met (as determined by the corresponding interrupt condition register field), the corresponding bit in this register is set. Two wait states are required in read access for synchronization. One wait state is required for reset.

8.3.5.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ISR															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.5.1.8.4 Fields

Field	Function
31-0 ISR	Interrupt status bits Bit n of this register is asserted (active high) when the active condition (as determined by the corresponding ICR bit) is detected on the GPIO input and is waiting for service. The value of this register is independent of the value in IMR register. When the active condition has been detected, the corresponding bit remains set until cleared by software. Status flags are cleared by writing a 1 to the corresponding bit position.

8.3.5.1.9 GPIO edge select register (EDGE_SEL)

8.3.5.1.9.1 Offset

Register	Offset
EDGE_SEL	1Ch

8.3.5.1.9.2 Function

The EDGE_SEL register may be used to override the ICR registers' configuration. If the GPIO_EDGE_SEL bit is set, then a rising edge or falling edge in the corresponding signal generates an interrupt. This register provides backward compatibility. On reset all bits are cleared (ICR is not overridden).

8.3.5.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPIO_EDGE_SEL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPIO_EDGE_SEL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.5.1.9.4 Fields

Field	Function
31-0 GPIO_EDGE_SEL	Edge select When EDGE_SEL[*n*] is set, the GPIO disregards the ICR[*n*] setting, and detects any edge on the corresponding input signal.

Chapter 9
External Memory

9.1 External Memory Overview

9.1.1 External Memory Overview

9.1.1.1 DRAM Interface

The SDRAM configurations supported by this chip are provided in the following table.

Table 9-1. DRAM controller supported SDRAM configurations

Parameter	LPDDR4	DDR4
Number of Controllers	1	1
Number of Channels	2	N/A
Number of Chip Selects	2	1
Bus Width	32-bit	32-bit
Maximum Clock Frequency	4000 MT/s	3200 MT/s

The DRAM controller and PHY also support the following:

- Inline ECC feature for LPDDR4 and DDR4
- Various low power modes, clock and power gated operations, that are defined for this chip. In addition, it can place the external DRAM into and out of its self-refresh mode as requested by different low-power operating modes.

9.1.1.2 SD/eMMC/SDIO Support

The chip has three Ultra Secured Digital Host Controller (uSDHC) modules for the SD/eMMC interface. It provides the interface between the host system and SD/SDIO/eMMC memory devices. The key features include:

- Supports SD/SDIO standard, up to version 3.0
- Supports eMMC standard, up to version 5.1
- Supports 3.3V and 1.8V operation, but does not support 1.2V operation
- Supports 1-bit / 4-bit SD and SDIO modes, 1-bit / 4-bit / 8-bit eMMC modes
  - Up to 400 Mbps of data transfer for SDIO cards using 4 parallel data lines in SDR mode
  - Up to 800 Mbps of data transfer for SDIO card using 4 parallel data lines in DDR mode
  - Up to 400 Mbps of data transfer for SDXC cards using 4 parallel data lines in SDR mode
  - Up to 800 Mbps of data transfer for SDXC card using 4 parallel data lines in DDR mode
  - Up to 1600 Mbps of data transfer for eMMC using 8 parallel data lines in SDR mode
  - Up to 3200 Mbps of data transfer for eMMC using 8 parallel data lines in DDR mode
- Two uSDHC controllers (SD1 and SD3) can support up to 8-bit interface, the other controller (SD2) can only support up to 4-bit interface.

9.1.1.3 Raw NAND Support

The Raw NAND Flash controller consists of three components:

- General Purpose Media Interface (GPMI) as the NAND controller pin interface
- AHB-to-APBH Bridge with DMA (APBH_DMA) as the DMA engine that drives the GPMI module
- 62-bit Correcting ECC Accelerator (BCH) as the 62-bit error correction hardware engine with an AXI bus master and a private connection to GPMI

The Raw NAND Flash controller support following key features:

- 8-bit NAND Flash, up to 4 devices supported by 4 chip-selects and 1 ganged ready/busy
- ONFI 3.2 complaint, synchronous clock rate of up to 100 MHz with data rate of up to 200 MB/s
- Supports ONFI NAND for Micron and SK Hynix, and Toggle NAND for Kioxia and Samsung[1]
- BCH62 for ECC, up to 200MB/s

1. Testing has been performed on select memory devices. Please contact your NXP representative for more details.

9.1.1.4 xSPI Interface

There is one Flexible Serial Peripheral Interface (FlexSPI) block that acts as an interface to one external Octal SPI serial flash device, or up to two external Quad SPI serial flash devices, each with up to four bidirectional data lines. The key features include:

- Channel A can be configured as 1/2/4/8-bit operation (Channel B is not usable when 8-bit operation is used)
- Channel B can be configured as 1/2/4-bit operation
- Support both dual-channel or single-channel operation
- Support both SDR mode and DDR mode
- Support up to 166MHz SDR Mode and 166MHz DDR Mode (with external Flash device DQS input)
- Support up to 133MHz SDR Mode and 66MHz DDR Mode (with internal DQS loopback mode)

9.2 DDR Controller (DDRC)

9.2.1 Overview

The DDRC (DDR Controller) is a memory controller that interfaces DDR based memories. The following types of SDRAMs are supported by the DDRC:

- DDR4
- LPDDR4

NOTE

The information within this block guide is Synopsys Proprietary. Used with permission.

9.2.1.1 Block diagram

The block diagram for the DDR Controller is shown below.

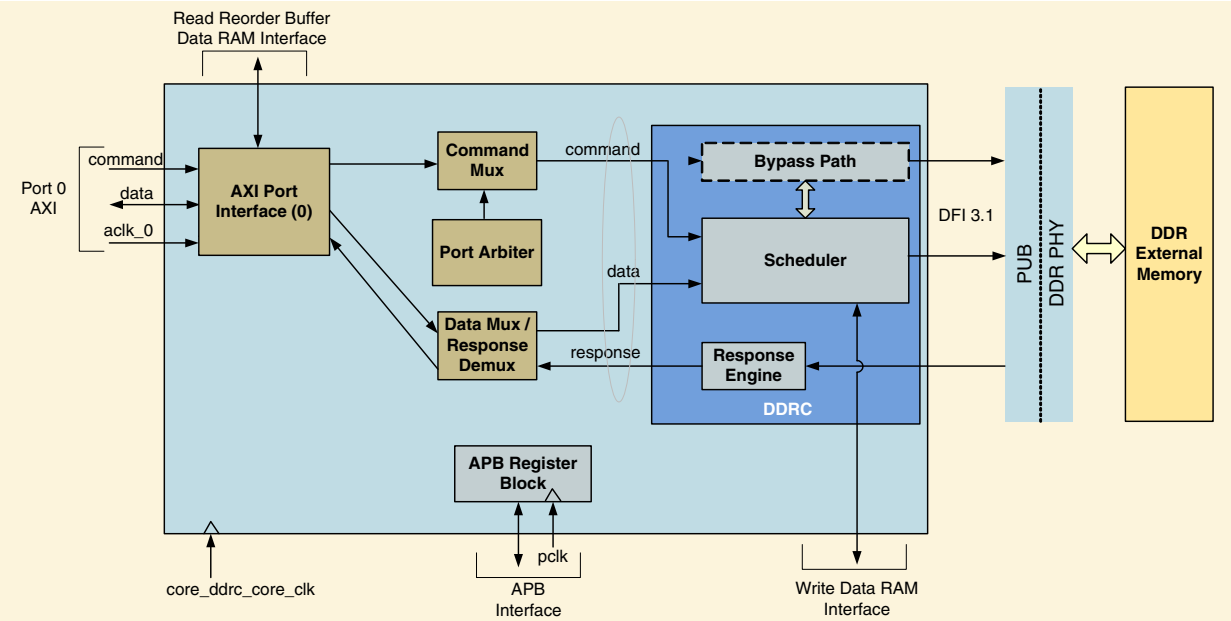


Figure 9-1. DDRC block diagram

The DDRC can also be configured to use a single custom host port interface with no port arbitration. In this configuration, it receives read requests, write requests, and data through the single port lowest latency HIF. Write requests are sent to the DDRC followed by the associated write data from the SoC core. Read requests are made with a tag field. The DDRC returns the tag field with the data when it returns the read data.

The PHY Utility Block (PUB) is an adaption layer between the controller and the PHY, handling control function.

9.2.1.2 Features

The DDRC includes the following features:

- Scalable 1:1 or 1:2 frequency ratio architecture
- Direct software request control or programmable internal control for ZQ short calibration cycles
- Support for ZQ long calibration after self-refresh exit
- Direct software request control or programmable internal control for ZQ calibration cycles
- Support for ZQ Reset feature through software
- Dynamic scheduling to optimize bandwidth and latency

- Read and write buffers in fully associative CAMs, 32 each
- Delayed writes for optimum performance on SDRAM data bus
- Out of order execution of commands maximises the SDRAM efficiency
- Programmable SDRAM parameters
- Supports BL16 burst length
- Supports 1,2 memory ranks
- Control options to avoid starvation of lower priorities
- Guaranteed coherency for write-after-read (WAR) and read-after-write (RAW) hazards
- Write combine to allow multiple writes to the same address to be combined into a single write to SDRAM; supported for same starting address
- Paging policy selectable by configuration registers as any of the following:
  - Leave pages open after accesses, or
  - Close page when there are no further accesses available in the controller for that page, or
  - Auto-precharge with each access, with an optimization for page-close mode which leaves the page open after a flush for read-write and write-read collision cases
- Supports automatic SDRAM power-down entry and exit caused by lack of transaction arrival for programmable time
- Supports automatic Clock Stop entry and exit caused by lack of transaction arrival
- Supports automatic DDRC low power mode operation caused by lack of transaction arrival for programmable time via Hardware Low Power Interface
- Supports self-refresh entry and exit as follows:
  - Supports automatic self-refresh entry and exit caused by lack of transaction arrival for programmable time
  - Support for self-refresh entry and exit under software control
- Support for dynamically changing clock frequency while in self-refresh:
  - Shadow timing registers provided to allow fast frequency changing
- Support for explicit SDRAM mode register updates under software control
- Flexible address mapper logic to allow application specific mapping of row, column, bank, and rank bits
- User-selectable refresh control options:
  - Controller-generated auto-refreshes at programmable average intervals
  - In multi-rank designs, an offset can be applied to each rank's refresh timer to allow rank refreshes to expire at different times (this can increase efficiency by allowing traffic to continue to other ranks while a given rank is being refreshed)
  - Ability to group up to 8 controller-generated refreshes together to be issued consecutively (this reduces the frequency of page closings, increasing overall efficiency)

    - When controller-generated refreshes are grouped, some refreshes can be issued speculatively when the controller is idle for a programmable period of time
    - Ability to disable controller-generated auto-refreshes
    - Ability to issue a refresh through direct software request
    - When DDR4 is used, fine granularity refresh can be selected
- Optional ECC support, when controller is configured for 64-bit/32-bit/16-bit SDRAM data width, including the following:
    - Support for single error correction / double error detection (SEC/DED)
    - Automatic data scrubbing of correctable errors
    - Support for highly efficient read-modify-write when byte enables are used with ECC enabled
    - Automatic logging of both correctable and uncorrectable errors
    - Ability to “poison” the write data by adding uncorrectable errors, for use in testing ECC error handling
- DDR4 protocol up to DDR4-3200 speed grade is supported by the DDRC. The following features are supported:
    - Data Bus Inversion (DBI)
    - Geardown mode may be programmed where allowed by memory and PHY specification
    - Maximum power saving mode
    - Multi-purpose register (MPR) reads and writes
    - Per DRAM addressability
    - Fine granularity refresh
    - Cyclic redundancy check (CRC) with retry
    - Command/address latency
    - Programmable Preamble
- Supports in-line ECC

9.2.2 Functional description

The DDRC converts system bus transactions into memory commands on the DFI interface that are compliant with the DDR protocols.

The main components of the block are:

- **AXI Port Interface (XPI) block**: This block provides the interface to the application ports. It provides bus protocol handling, data buffering and reordering for read data, data bus size conversion (upsizing or downsizing), and memory burst address alignment. Read data is stored in a SRAM, read re-order buffer and returned in order, to the AXI ports. The SRAM may be instantiated as embedded memory external to the DDRC or be implemented as flops within the DDRC.

- **Port Arbiter (PA) block**: This block provides latency sensitive, priority based arbitration between the addresses issued by the XPIs (by the ports).
- **DDR Controller (DDRC) block**: This block contains a logical CAM (Content Addressable Memory), which can be synthesized using standard cells. This holds information on the commands, which is used by the scheduling algorithms to optimally schedule commands to be sent to the PHY, based on priority, bank/rank status and DDR timing constraints. A bypass path is also provided (optionally).
- **APB Register Block**: This block contains the software accessible registers.

Please see Block diagram for the DDRC block diagram.

The details on DDRC are as follows:

- Write data is stored in DDRC until its associated command is issued to the PHY.
- Read data is handled by the response engine in the DDRC and is returned in the order of scheduled read commands on the HIF.
- ECC is handled by logic modules within the DDRC in the write data path and in the response engine.

1:2 frequency ratio term is used for the frequency ratio between the DDRC clock (core_ddrc_core_clk) and the memory clock.

In 1:2 frequency ratio mode, the DDRC clock rate is one half the memory clock rate (referred to as HDR, half data rate).

The terms “DFI clock” and “DFI PHY clock” are used in this document in accordance with the DFI specification:

- The DFI clock is the clock on which the DFI interface runs. It is the core_ddrc_core_clk.
- The DFI PHY clock is the SDRAM clock. In 1:2 frequency ratio, this has twice the frequency of the DFI clock.

NOTE

The DDR Controller to memory clock frequency ratio is hard-coded to 1:2

9.2.2.1 AXI Port Interface (XPI)

The AXI protocol is burst-based with read and write request channels that specify the host ID for the request, start byte address, burst length, burst size, and burst type. This information is processed by the interface and is used subsequently by the DDRC.

The XPI interfaces the AXI application port to the DDRC and performs the following main functions:

- Read address generation
- Write address generation
- Read data generation
- Read data and response generation
- Write response generation

The XPI converts AXI bursts into read and write requests, which are forwarded to the Port Arbiter (PA). In the opposite direction, the XPI converts the responses from the DDRC into appropriate AXI responses.

The interface between XPI and PA is same as HIF (but with independent channels for read and write commands). Each AXI port can be independently configured to operate with a clock that is synchronous or asynchronous to the memory controller clock.

The AXI Specification requires that transactions must not cross a 4K address boundary. AXI compliant masters must meet this requirement.

9.2.2.1.1 Read Address Channel

The AXI read address channel has the following features:

- The read transaction can be of any length up to 256 for incremental bursts, 16 for wrapping bursts.
- The burst types supported are incremental and wrapping.
- The burst start address can be unaligned to the AXI data width boundaries
- The size of the burst can be less than the full width of the AXI data bus (also known as sub-sized transfers).

The signals on the read address channel are registered into the XPI in accordance with the AXI valid/ready handshaking protocol and are synchronous with the AXI clock (aclk).

Read requests are accepted if the request can be written into the Read Address Queue (RAQ). This is used to store all the read address requests. There are two address queues on a given port, named Blue and Red address queues. Clock domain crossing from aclk to DDRC clock (core_ddrc_core_clk) is performed in the RAQ block. The depth of the RAQ can be configured to suit the system requirements of the user.

After registering the read address channel by the RAQ, the following operations are performed:

- Generation of new read requests is based on alignment (derived from AXI address and size), burst lengths (derived from AXI length and memory burst length), and

burst type (derived from AXI incremental or wrapping). Each AXI burst is divided into packets of length equal to the memory burst length (BL2, BL4, BL8, BL16).
- Generation of new HIF address in the case of unaligned burst and burst expansion.

In general, realignment to a memory burst boundary potentially causes some data beats to be discarded (affecting bandwidth) and potentially introduces additional latency on the read data and response channel.

The arready output is defaulted to logic one and remains in logic one unless the RAQ becomes full or a WRAP burst is requested (arready is low on the next cycle after a WRAP burst is accepted). The read transaction is popped from the RAQ when it is not empty.

The XPI handles the generation of the token which is used by the DDRC for identifying the read command and corresponding data.

9.2.2.1.2 Write Address Channel

Write Address Queue (WAQ) is used to store all the addresses for write requests from a given port. There is a single queue for all AXI IDs from a given port. The write address channel behavior is similar to the read address channel.

Write address and read address channels are independent and the ordering between the write and read requests may not be preserved.

To preserve the sequence, a higher-level protocol needs to wait for read/write response before sending the next transaction.

Transactions across the ports are independent and can be issued in any order.

The write command is not forwarded to the DDRC until the write data is collected and the strobes are evaluated.

9.2.2.1.3 Read Data and Response Channel

The XPI handles the common response interface to the DDRC to process the read data from the memory.

AXI read data and response channel has a single data storage queue (RDQ - Read Data Queue) with two clocks—AXI clock for reading and DDRC clock for writing.

Data from different IDs are stored in the same queue and are returned in the order of read address acceptance. You can configure the depth of the queue independently for each port.

The FSM handling the AXI handshake as well as the FIFO push and pop has the following features:

- Based on the information stored, filter the dummy beats
- Based on the signal hif_rdata_last, generate the rlast

SLVERR response may be returned for the read transactions for the following cases:

- ECC uncorrected error detected at the DFI
- InvalidLPDDR4 row address

9.2.2.1.3.1 Read Reorder Buffer

The read data can be returned from the DDRC in a different order from which the read commands are forwarded from the XPI. (due to the re-ordering of read commands in the DDRC to maximize SDRAM bandwidth). A read reorder buffer is implemented in each port to reorder the read data for that port to the same order as the order of the AXI read commands.

The AXI protocol allows the read data for transactions of different IDs to be interleaved. To reduce potential delays where read data for one ID is blocked waiting for data associated with another ID, the read reorder buffer is organized as number of virtual channels (See Figure 9-2). The Read Reorder Virtual Channel is a mechanism to allow independent read data reordering between multiple groups of AXI IDs.

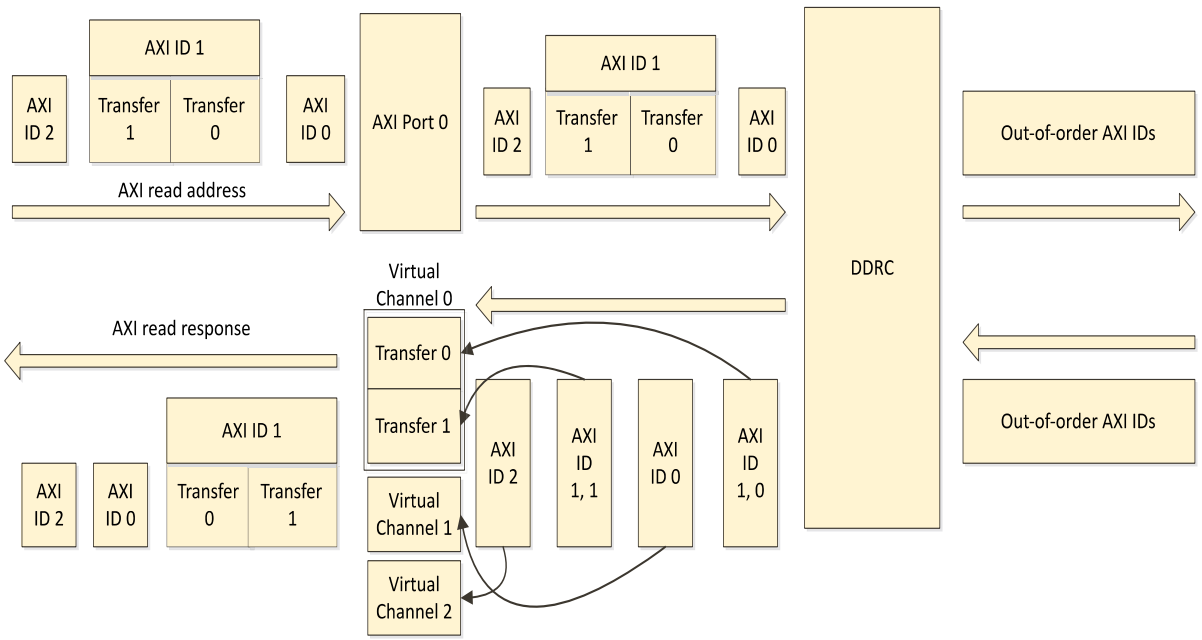


Figure 9-2. RRB Configured with Three Virtual Channels

9.2.2.1.4 Write data Channel

The AXI write data channel has a data storage queue (WDQ - Write Data Queue) with the following clocks:

- AXI clock (aclk_n) for writing and
- DDRC clock (core_ddrc_core_clk) for reading

This queue is used for the clock domain crossing between the AXI clock domain and the controller clock domain and acts as a registering layer in the case of a synchronous interface.

At the output of WDQ, some beats of a write data packet can be masked depending on alignment, burst size and burst length.

Write data from each port is forwarded to the DDRC, which forwards the data to a common data storage.

9.2.2.1.5 Write Response Channel

The write response is generated once the last beat of write data, for a given AXI burst, is accepted by the DDRC.

SLVERR response may be returned in the following cases:

- Invalid LPDDR4 row address

9.2.2.1.6 AXI Data Channel Interleaving

A given AXI port (XPI) can access both data channels and low granularity interleaving between the two data channels is possible by mapping a suitable LSB address bit which is at the memory burst boundary. In that case, for example, a long AXI burst can be expanded to HIF commands so that back to back accesses are to alternating data channels. This setting achieves the highest efficiency possible from a single AXI port.

On the read data path, data order - based on AXI ordering rules - has to be preserved within each channel and between both channels. The read reorder buffer reorders the data coming from a given channel. The data ordered from both channels is then reordered by the data channel reorder logic. Thus two separate data SRAMs are instantiated, one for each channel, whose outputs are combined to provide a single wide AXI read data channel.

By setting the correct system mapping, one port can be set to access one data channel and another port can be set to access the other data channel. This configuration saves logic area in XPI block.

9.2.2.2 Host Interface (HIF)

HIF is an internal interface when the DDRC is configured to include a multi port arbiter. In this case, the AXI Port Interface and/or Port Arbiter handle all signals on this interface, effectively acting as the SoC core.

Read, write, as well as write data are received through the HIF. Following the receipt and acceptance of a write the DDRC requests write data from the interface. The interface subsequently provides the independently timed write data in the requested order (which matches the order in which requests are made to the DDRC). Both read and write requests can be reordered later, before being output on the DFI interface.

Requests to the DDRC are throttled in two ways:

- A credit mechanism that ensures that buffer space is available for any request the SoC core makes to the DDRC, prior to the request being made.
- An independent stall mechanism that throttles requests when address collisions take place or if entering self-refresh through the software self-refresh or hardware low power self-refresh.

Data is handled separately. The DDRC throttles data by potentially waiting after it receives a write before requesting the associated data. If there is collision, then the throttling lasts until the collision is cleared in the DDRC. Once write data is requested, it is always accepted by the DDRC.

9.2.2.2.1 Credit Mechanism

The DDRC employs a credit mechanism to ensure that buffers do not overflow. The interface making the request to the DDRC, can only request commands for which it has been granted “credits” to issue.

Credits are tracked seperately for the following three command types:

- High Priority read(HPR)
- Low Prioroity rad(LPR)
- Write

Credits are counted for each command type independently according to the following rules:

- Initially the interface has zero credits.
- The interface logic must include counters to track the number of credits granted by the DDRC and decremented due to commands issued by the interface logic to the DDRC. The interface logic must have separate credit counters for each individual command type.
- Following the de-assertion of reset to the DDRC, credits are issued to the interface for each command type. A given credit count increments every time the DDRC issues a credit, indicated by the assertion of the appropriate *_credit signal on the rising edge of core_ddrc_core_clk. The credit counting logic implemented externally to the DDRC must run on the same clock.
- When the credit count is greater than zero, the interface can issue requests of that type to the DDRC. Each time a request is issued to the DDRC, the associated credit count is decremented.
- The arbiter or the host can send Variable Priority Read (VPR) and Variable Priority Write (VPW) commands.
- VPR commands do not have a pre-allocated storage resource in the Read CAM (unlike HPR commands). VPR commands share the same resource with the LPR

commands. As far as the credit mechanism is concerned, the VPR commands are counted in the LPR credit bucket.
- Similarly, VPW commands do not have a per-allocated storage resource in the Write CAM. VPW commands share the same resource with the Normal Write commands. As far as credit mechanism is concerned, the VPW commands are counted in the WR credit bucket.

- hif_cmd_valid
- hif_cmd_type
- hif_cmd_pri
- hif_lpr_credit
- hif_hpr_credit
- hif_wr_credit
- hif_cmd_stall

9.2.2.3 DFI Interface

The DDRC contains a DFI MC (Memory Controller) interface, which is used to connect to a DFIcompliant PHY, and to transfer address, control and data to the PHY. The DDRC interfaces to a PUB block which acts as the DFI PHY interface.

The DFI interface is sub-divided into the following interface groups:

- Control Interface
- Write Data Interface
- Read Data Interface
- Update Interface
- Status Interface
- Training Interface (PHY-independent mode supported by the DDRC)
- Low Power Control Interface

9.2.2.3.1 1:2 Frequency Ratio Mode Considerations

The DDRC, including its DFI interface runs at the half data rate (HDR) clock (referred to as 1:2 frequency ratio mode in the DFI Specification), so all DFI command and address signals are twice the width of the equivalent DDR SDRAM signals. DFI data signals are four times the width of the equivalent DDR SDRAM signals. The DDRC operates with an HDR clock. The PHY handles HDR-to-SDR conversion on the address bus to memory and HDR/DDR conversion on the data buses. In 1:2 frequency ratio mode, core_ddrc_core_clk runs at half the frequency of CK/CK#.

9.2.2.3.2 Control Interface

The control interface is a reflection of the SDRAM control interface including address, bank, chip select, row strobe, column strobe, write enable, clock enable and ODT control, as applicable for the memory technology. This interface consists of the following signals:

- dfi_address
- dfi_bank
- dfi_cas_n
- dfi_cke
- dfi_cs
- dfi_odt
- dfi_ras_n
- dfi_reset_n
- dfi_we_n
- dfi_bg
- dfi_act_n

9.2.2.3.3 Write Data Interface

The write data interface handles the transmission of write data across the DFI interface. The DFI Specification defines signals and timing relationships. This interface consists of the following signals:

- dfi_wrdata
- dfi_wrdata_en
- dfi_wrdata_mask

The timing of this interface is user-configurable, to support all DFI-compliant PHYs through the DFITMG0[dfi_tphy_wrlat] and DFITMG0[dfi_tphy_wrdata] bits. For DFI 1:2, DFITMG0[dfi_wrdata_use_dfi_phy_clk] determines whether DFITMG0[dfi_tphy_wrlat] and DFITMG0[dfi_tphy_wrdata] are in terms of SDR or HDR clock cycles. If HDR, timing for dfi_wrdata* generation for write commands on phase 0/phase 1 are the same and dfi_wrdata* signals are aligned to HDR clock. If SDR, timing for dfi_wrdata* generation for write commands on phase 0/phase 1 are treated separately and dfi_wrdata* signals are not guaranteed to be aligned to HDR clock. Check your PHY requirements for correct programming.

In DDR4 configurations or LPDDR4 configurations, the signal dfi_wrdata_mask can act as Write DBI signal depending on the user programming of the SDRAM mode register.

Also, as per JEDEC Specification, the polarity of dfi_wrdata_mask is inverted when using DDR4 memories (if MSTR[ddr4] = 1). Therefore, for DDR4, if a bit of dfi_wrdata_mask is ‘0’, the corresponding byte is masked.

9.2.2.3.3.1 Support for dfi_wrdata_cs/dfi_rddata_cs Signals (DFI 3.1)

The signal dfi_wrdata_cs is driven as follows:

- dfi_wrdata_cs is driven according to the relevant chip select tphy_wrcslat DFI PHY clock cycles after any command that generates dfi_wrdata. This includes a write, and in DDR4, a PDA (write). It is guaranteed to remain at this value for a minimum of tphy_wrcsgap plus the time for the write data (usually MSTR[burst_rdwr]).

Similarly, for dfi_rddata_c:

- dfi_rddata_cs is driven according to the relevant chip select tphy_rdcslat DFI PHY clock cycles after any command that generates dfi_rddata. This includes a Read, in LPDDR4, a MRR and in DDR4, a MPR (read). It is guaranteed to remain at this value for a minimum of tphy_rdcsgap plus the time for the read data (usually MSTR[burst_rdwr]).

The table outlines the DFI timing value and the equivalent register in the DDRC.

Table 9-2. DFI timing values for dfi_wrdata_cs/dfi_rddata_cs

DFI Timing Value	DDRC Register
$t_{phy_wrcslat}$	DFITMG2[dfi_tphy_wrcslat]
$t_{phy_wrcsgap}$	RANKCTL[diff_rank_wr_gap][1]
$t_{phy_rdcslat}$	DFITMG2[dfi_tphy_rdcslat]
$t_{phy_rdcsgap}$	RANKCTL[diff_rank_rd_gap][1]

NOTE

1. This register is in terms of DFI clocks (controller clocks), while tphy_wrcsgap is in terms of DFI PHY clocks.

9.2.2.3.4 Read Data Interface

The read data interface handles the return of read data across the DFI interface. The DFI Specification defines signals and timing relationships. This interface consists of the following signals:

- dfi_rddata
- dfi_rddata_en

- dfi_rddata_valid
- dfi_rddata_dbi

The timing of this interface is user-configurable, to support all DFI-compliant PHYs via the DFITMG0[t_rddata_en] bit.

For DFI 1:2, DFITMG0[dfi_rddata_use_dfi_phy_clk] determines whether DFITMG0[dfi_t_rddata_en] is in terms of SDR or HDR clock cycles. If HDR, timing for dfi_rddata_en generation for read commands on phase 0/phase 1 is the same and dfi_rddata* signals are aligned to HDR clock. If SDR, timing for dfi_rddata* generation for read commands on phase 0/phase 1 are treated separately and dfi_rddata* signals are not guaranteed to be aligned to HDR clock. Check the PHY requirements for correct programming.

In DDR4 configurations, the signal dfi_rddata_dbi is the read DBI signal.

The DDRC has been verified with the PHYs. It assumes that the DFI parameter tphy_rdlat does not exceed 48 cycles.It also assumes that the DFI parameter trddata_en does not exceed the SDRAM read latency (RL).

9.2.2.3.5 Update Interface

During system operation, the system may require updates to internal settings to compensate for environmental conditions. To ensure that updates do not interfere with signals on the SDRAM interface, the DFI interface supports update modes where the DFI read, write, and control interface are suspended from normal activity. The DFI Specification defines both MC-initiated and PHY-initiated updates.

9.2.2.3.5.1 MC-initiated Updates

This interface consists of the following signals:

- dfi_ctrlupd_req
- dfi_ctrlupd_ack
- dfi_ctrlupd_ack2

MC-initiated updates can be acknowledged or ignored by the PHY. DFI MC-initiated updates are performed periodically by DDRC.

9.2.2.3.5.2 PHY-initiated Updates

This interface consists of the following signals:

- dfi_phyupd_req
- dfi_phyupd_type
- dfi_phyupd_ack

9.2.2.3.6 Status Interface

The DFI interface requires status information for initialization and clock control to the SDRAM devices. These signals are used to convey information between the MC and PHY. This interface consists of the following signals:

- dfi_init_start
- dfi_init_complete
- dfi_frequency
- dfi_dram_clk_disable
- dfi_alert_n (this signal replaces dfi_parity_error).

The DDRC does not support the following signals (defined as optional in the DFI Specification):

- dfi_data_byte_disable
- dfi_freq_ratio

9.2.2.3.6.1 Initialization

The dfi_init_start signal is used to trigger the PHY initialization by setting it to 1. It is driven by the DDRC through the APB interface with the DFIMISC[dfi_init_start read-write] register field. The signal dfi_init_complete indicates that the PHY has completed its initialization. This information can be read/polled by the DDRC through the APB interface with the DFISTAT[dfi_init_complete] read-only register field.

9.2.2.3.6.2 Clock Disabling

The dfi_dram_clk_disable signal depends on setting of the PWRCTL[en_dfi_dram_clk_disable] bit.

9.2.2.3.6.3 Frequency Change

The interface consists of three signals:

- dfi_init_start: Output from the controller to the PHY
- dfi_init_complete: Output from the PHY to the controller
- dfi_frequency: Output from the controller to the PHY

The procedure is as follows:

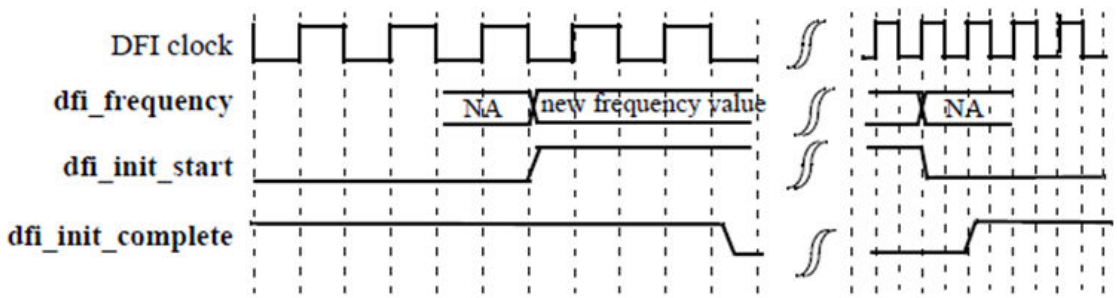


Figure 9-3. Frequency Change

dfi_frequency indicates the operating frequency of the system. This signal should change only at initialization, during a DFI frequency change operation, or other times that the system defines.This signal should be constant during normal operation.The number of supported frequencies and the mapping of signal values to clock frequencies are defined by the PHY, system, or both. For timing, the dfi_frequency signal must be set to a legal value and remain unchanged when dfi_init_start is asserted.The dfi_frequency signal can change any time when dfi_init_start is low and should be ignored at this time.

The behavior of the dfi_init_start signal is dependent on the dfi_init_complete signal.

If the PHY accepts the frequency change request, it must deassert the dfi_init_complete signal within tinit_start cycles of the dfi_init_start assertion. The MC continues to hold the dfi_init_start signal asserted until the clock frequency change has been completed. The de-assertion should be used by the PHY to reinitialize on the new clock frequency.

If the frequency change is not acknowledged (the dfi_init_complete signal remains asserted), the dfi_init_start signal must de-assert after tinit_start cycles. This requirement needs to be handled by the system as dfi_init_start is programmed through the software.

9.2.2.3.7 Low Power interface

In a DDR memory subsystem, it is advantageous to place the PHY in a low power state when the DDRC has knowledge that the memory subsystem remains idle for a period of time. Depending on the state of the system, the DDRC communicates state information to the PHY allowing the PHY to enter the appropriate power saving state. This interface consists of signals that are used to inform the PHY of a low power mode opportunity, as well as how quickly the DDRC requires the PHY to resume normal operation. This interface consists of the following signals:

- dfi_lp_req
- dfi_lp_wakeup
- dfi_lp_ack Software control is provided in DFILPCFG0/DFILPCFG1 registers for the following:
- This is an optional interface for a DFI-compliant PHY. The interface signals always exist in the DDRC. But the software can enable this interface in self-refresh and/or power-down and/or deep power-down mode and/or maximum power saving mode - DFILPCFG0[dfi_lp_en_pd], DFILPCFG0[dfi_lp_en_sr], DFILPCFG0[dfi_lp_en_dpd] and DFILPCFG1[dfi_lp_en_mpsm] respectively.
- Software control over what is driven on dfi_lp_wakeup signal (and its associated timing) in selfrefresh or power-down or deep power-down or maximum power saving mode - DFILPCFG0[dfi_lp_wakeup_pd], DFILPCFG0[dfi_lp_wakeup_sr], DFILPCFG0[dfi_lp_wakeup_dpd] and DFILPCFG1[dfi_lp_wakeup_mpsm] respectively.
- The timing of this DFI low power interface - DFILPCFG0[dfi_tlp_resp].
- ctl_idle output pin is a sideband signal (not specified in DFI). This is driven the same as dfi_lp_req if enabled via software (DFIMISC[ctl_idle_en]=1 and DFI Low Power Interface is also enabled via DFILPCFG0/DFILPCFG1 registers). If configured for Shared AC, dfi_lp_req signal of both data channels must be asserted for ctl_idle to be asserted.

9.2.2.3.8 PHY Master Interface

This interface consists of the following signals:

- dfi_phymstr_req
- dfi_phymstr_cs_state
- dfi_phymstr_state_sel
- dfi_phymstr_type
- dfi_phymstr_ack

9.2.2.4 APB Interface

The DDRC provides a dedicated APB 3.0 bus interface that is used to access the DDRC software programmable registers. The DDRC converts APB reads and writes into accesses to the internal register file.

9.2.2.4.1 Register Classes

There are three classes of registers in the DDRC:

- Dynamic Registers
- Quasi Dynamic Registers
- Static Registers

Dynamic registers can be written at any time during operation.

Static registers can be written only when the controller is in reset.

Quasi dynamic registers can be written when the controller is in reset and some specific conditions outside reset. To write them, SWCTL[sw_done] has to be set to 0 at the beginning of the programming sequence and set back to 1 at the end. After that, SWSTAT[sw_done_ack] has to be polled for acknowledge.

9.2.2.5 Clocks

The following table provides the clocks for the DDRC module.

Table 9-3. Clocks

Clock	Description
core_ddrc_core_clk	The main clock for the DDRC. The core_ddrc_core_clk clock is synchronous to the PHY clock.
pclk	APB clock

9.2.2.6 Interrupts

The DDRC consists of the following interrupts:

- Interrupt for DRAM controller's performance monitor
- Error Interrupt for DFI error
- Scrubber Interrupt indicating one full address range sweep
- Error Interrupt for address protection fault

- Error Interrupt for correctable ECC error detected
- Error Interrupt for uncorrectable ECC error detected

Refer to the Interrupts chapter for details on the interrupt assignments.

9.2.3 External Signals

There are no external signals pinned out for the DDRC.

9.2.4 Initialization and Application Information

NXP provides a DDR Tool to help users perform LPDDR4/DDR4 training, stress test, and DDR initial code generation for u-boot. Please refer to the DDR Tool User's Guide for more information.

9.2.5 Memory Map and register definition

This section includes the DDRC module memory map and detailed descriptions of all registers.

NOTE

The information within this block guide is Synopsys Proprietary. Used with permission.

9.2.5.1 DDRC register descriptions

9.2.5.1.1 DDRC memory map

DDRC base address: 3D40_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Master Register0 (MSTR)	32	RW	0304_0001h
4h	Operating Mode Status Register (STAT)	32	R	0000_0000h
8h	Operating Mode Status Register (MSTR1)	32	RW	0000_0000h
Ch	Operating Mode Status Register (MRCTRL3)	32	RW	0000_0003h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
10h	Mode Register Read/Write Control Register 0. (MRCTRL0)	32	RW	0000_0030h
14h	Mode Register Read/Write Control Register 1 (MRCTRL1)	32	RW	0000_0000h
18h	Mode Register Read/Write Status Register (MRSTAT)	32	R	0000_0000h
1Ch	Mode Register Read/Write Control Register 2 (MRCTRL2)	32	RW	0000_0000h
20h	Temperature Derate Enable Register (DERATEEN)	32	RW	0000_0000h
24h	Temperature Derate Interval Register (DERATEINT)	32	RW	0080_0000h
30h	Low Power Control Register (PWRCTL)	32	RW	0000_0000h
34h	Low Power Timing Register (PWRTMG)	32	RW	0040_2010h
38h	Hardware Low Power Control Register (HWLPCTL)	32	RW	0000_0003h
50h	Refresh Control Register 0 (RFSHCTL0)	32	RW	0021_0000h
54h	Refresh Control Register 1 (RFSHCTL1)	32	RW	0000_0000h
60h	Refresh Control Register 3 (RFSHCTL3)	32	RW	0000_0000h
64h	Refresh Timing Register (RFSHTMG)	32	RW	0062_008Ch
70h	ECC Configuration Register 0 (ECCCFG0)	32	RW	003F_7F40h
74h	ECC Configuration Register 1 (ECCCFG1)	32	RW	0000_00B0h
78h	SECDED ECC status register (ECCSTAT)	32	RO	0000_0000h
7Ch	ECC clear register (ECCCTL)	32	RW	0000_0000h
80h	ECC error counter register (ECCERRCNT)	32	RO	0000_0000h
84h	ECC corrected error address register 0 (ECCCADDR0)	32	RO	0000_0000h
88h	ECC corrected error address register 1 (ECCCADDR1)	32	RO	0000_0000h
8Ch	ECC corrected syndrome register 0 (ECCCSYN0)	32	RO	0000_0000h
90h	ECC corrected syndrome register 1 (ECCCSYN1)	32	RO	0000_0000h
94h	ECC corrected syndrome register 2 (ECCCSYN2)	32	RO	0000_0000h
98h	ECC corrected data bit mask register 0 (ECCBITMASK0)	32	RO	0000_0000h
9Ch	ECC corrected data bit mask register 1 (ECCBITMASK1)	32	ROZ	0000_0000h
A0h	ECC corrected data bit mask register 2 (ECCBITMASK2)	32	RO	0000_0000h
A4h	ECC uncorrected error address register 0 (ECCUADDR0)	32	RO	0000_0000h
A8h	ECC uncorrected error address register 1 (ECCUADDR1)	32	RO	0000_0000h
D0h	SDRAM Initialization Register 0 (INIT0)	32	RW	0002_004Eh
D4h	SDRAM Initialization Register 1 (INIT1)	32	RW	0000_0000h
DCh	SDRAM Initialization Register 3 (INIT3)	32	RW	0000_0510h
E0h	SDRAM Initialization Register 4 (INIT4)	32	RW	0000_0000h
E4h	SDRAM Initialization Register 5 (INIT5)	32	RW	0010_0004h
E8h	SDRAM Initialization Register 6 (INIT6)	32	RW	0000_0000h
ECh	SDRAM Initialization Register 7 (INIT7)	32	RW	0000_0000h
F4h	Rank Control Register (RANKCTL)	32	RW	0000_066Fh
100h	SDRAM Timing Register 0 (DRAMTMG0)	32	RW	0F10_1B0Fh
104h	SDRAM Timing Register 1 (DRAMTMG1)	32	RW	0008_0414h
108h	SDRAM Timing Register 2 (DRAMTMG2)	32	RW	0305_060Dh
10Ch	SDRAM Timing Register 3 (DRAMTMG3)	32	RW	0050_400Ch

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
110h	SDRAM Timing Register 4 (DRAMTMG4)	32	RW	0504_0405h
114h	SDRAM Timing Register 5 (DRAMTMG5)	32	RW	0505_0403h
118h	SDRAM Timing Register 6 (DRAMTMG6)	32	RW	0202_0005h
11Ch	SDRAM Timing Register 7 (DRAMTMG7)	32	RW	0000_0202h
120h	SDRAM Timing Register 8 (DRAMTMG8)	32	RW	0303_4405h
124h	SDRAM Timing Register 9 (DRAMTMG9)	32	RW	0004_040Dh
128h	SDRAM Timing Register 10 (DRAMTMG10)	32	RW	001C_180Ah
12Ch	SDRAM Timing Register 11 (DRAMTMG11)	32	RW	440C_021Ch
130h	SDRAM Timing Register 12 (DRAMTMG12)	32	RW	0002_0610h
134h	SDRAM Timing Register 13 (DRAMTMG13)	32	RW	1C20_0004h
138h	SDRAM Timing Register 14 (DRAMTMG14)	32	RW	0000_00A0h
13Ch	SDRAM Timing Register 15 (DRAMTMG15)	32	RW	0000_0000h
180h	ZQ Control Register 0 (ZQCTL0)	32	RW	0200_0040h
184h	ZQ Control Register 1 (ZQCTL1)	32	RW	0200_0100h
188h	ZQ Control Register 2 (ZQCTL2)	32	RW	0000_0000h
18Ch	ZQ Status Register (ZQSTAT)	32	R	0000_0000h
190h	DFI Timing Register 0 (DFITMG0)	32	RW	0702_0002h
194h	DFI Timing Register 1 (DFITMG1)	32	RW	0000_0404h
198h	DFI Low Power Configuration Register 0 (DFILPCFG0)	32	RW	0700_0000h
19Ch	DFI Low Power Configuration Register 1 (DFILPCFG1)	32	RW	0000_0000h
1A0h	DFI Update Register 0 (DFIUPD0)	32	RW	0040_0003h
1A4h	DFI Update Register 1 (DFIUPD1)	32	RW	0001_0001h
1A8h	DFI Update Register 2 (DFIUPD2)	32	RW	8000_0000h
1B0h	DFI Miscellaneous Control Register (DFIMISC)	32	RW	0000_0001h
1B4h	DFI Timing Register 2 (DFITMG2)	32	RW	0000_0202h
1B8h	DFI Timing Register 3 (DFITMG3)	32	RW	0000_0000h
1BCh	DFI Status Register (DFISTAT)	32	R	0000_0000h
1C0h	DM/DBI Control Register (DBICTL)	32	RW	0000_0001h
200h	Address Map Register 0 (ADDRMAP0)	32	RW	0000_0000h
204h	Address Map Register 1 (ADDRMAP1)	32	RW	0000_0000h
208h	Address Map Register 2 (ADDRMAP2)	32	RW	0000_0000h
20Ch	Address Map Register 3 (ADDRMAP3)	32	RW	0000_0000h
210h	Address Map Register 4 (ADDRMAP4)	32	RW	0000_0000h
214h	Address Map Register 5 (ADDRMAP5)	32	RW	0000_0000h
218h	Address Map Register 6 (ADDRMAP6)	32	RW	0000_0000h
21Ch	Address Map Register 7 (ADDRMAP7)	32	RW	0000_0000h
220h	Address Map Register 8 (ADDRMAP8)	32	RW	0000_0000h
224h	Address Map Register 9 (ADDRMAP9)	32	RW	0000_0000h
228h	Address Map Register 10 (ADDRMAP10)	32	RW	0000_0000h
22Ch	Address Map Register 11 (ADDRMAP11)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
240h	ODT Configuration Register (ODTCFG)	32	RW	0400_0400h
244h	ODT/Rank Map Register (ODTMAP)	32	RW	0000_2211h
300h	Debug Register 0 (DBG0)	32	RW	0000_0000h
304h	Debug Register 1 (DBG1)	32	RW	0000_0000h
308h	CAM Debug Register (DBGCAM)	32	R	0000_0000h
30Ch	Command Debug Register (DBGCMD)	32	RW	0000_0000h
310h	Status Debug Register (DBGSTAT)	32	R	0000_0000h
320h	Software Register Programming Control Enable (SWCTL)	32	RW	0000_0001h
324h	Software Register Programming Control Status (SWSTAT)	32	R	0000_0001h
36Ch	AXI Poison Configuration Register. (POISONCFG)	32	RW	0011_0011h
370h	AXI Poison Status Register (POISONSTAT)	32	R	0000_0000h
3FCh	Port Status Register (PSTAT)	32	R	0000_0000h
400h	Port Common Configuration Register (PCCFG)	32	RW	0000_0000h
404h	Port n Configuration Read Register (PCFGR_0)	32	RW	0000_0000h
408h	Port n Configuration Write Register (PCFGW_0)	32	RW	0000_4000h
490h	Port n Control Register (PCTRL_0)	32	RW	0000_0000h
2020h	[SHADOW] Temperature Derate Enable Register (DERATEEN_SHADOW)	32	RW	0000_0000h
2024h	[SHADOW] Temperature Derate Interval Register (DERATEINT_SHADOW)	32	RW	0080_0000h
2050h	[SHADOW] Refresh Control Register 0 (RFSHCTL0_SHADOW)	32	RW	0021_0000h
2064h	[SHADOW] Refresh Timing Register (RFSHTMG_SHADOW)	32	RW	0062_008Ch
20DCh	[SHADOW] SDRAM Initialization Register 3 (INIT3_SHADOW)	32	RW	0000_0510h
20E0h	[SHADOW] SDRAM Initialization Register 4 (INIT4_SHADOW)	32	RW	0000_0000h
20E8h	[SHADOW] SDRAM Initialization Register 6 (INIT6_SHADOW)	32	RW	0000_0000h
20ECh	[SHADOW] SDRAM Initialization Register 7 (INIT7_SHADOW)	32	RW	0000_0000h
2100h	[SHADOW] SDRAM Timing Register 0 (DRAMTMG0_SHADOW)	32	RW	0F10_1B0Fh
2104h	[SHADOW] SDRAM Timing Register 1 (DRAMTMG1_SHADOW)	32	RW	0008_0414h
2108h	[SHADOW] SDRAM Timing Register 2 (DRAMTMG2_SHADOW)	32	RW	0305_060Dh
210Ch	[SHADOW] SDRAM Timing Register 3 (DRAMTMG3_SHADOW)	32	RW	0050_400Ch
2110h	[SHADOW] SDRAM Timing Register 4 (DRAMTMG4_SHADOW)	32	RW	0504_0405h
2114h	[SHADOW] SDRAM Timing Register 5 (DRAMTMG5_SHADOW)	32	RW	0505_0403h
2118h	[SHADOW] SDRAM Timing Register 6 (DRAMTMG6_SHADOW)	32	RW	0202_0005h
211Ch	[SHADOW] SDRAM Timing Register 7 (DRAMTMG7_SHADOW)	32	RW	0000_0202h
2120h	[SHADOW] SDRAM Timing Register 8 (DRAMTMG8_SHADOW)	32	RW	0303_4405h
2124h	[SHADOW] SDRAM Timing Register 9 (DRAMTMG9_SHADOW)	32	RW	0004_040Dh
2128h	[SHADOW] SDRAM Timing Register 10 (DRAMTMG10_SHADOW)	32	RW	001C_180Ah
212Ch	[SHADOW] SDRAM Timing Register 11 (DRAMTMG11_SHADOW)	32	RW	440C_021Ch
2130h	[SHADOW] SDRAM Timing Register 12 (DRAMTMG12_SHADOW)	32	RW	0002_0610h
2134h	[SHADOW] SDRAM Timing Register 13 (DRAMTMG13_SHADOW)	32	RW	1C20_0004h

Table continues on the next page...

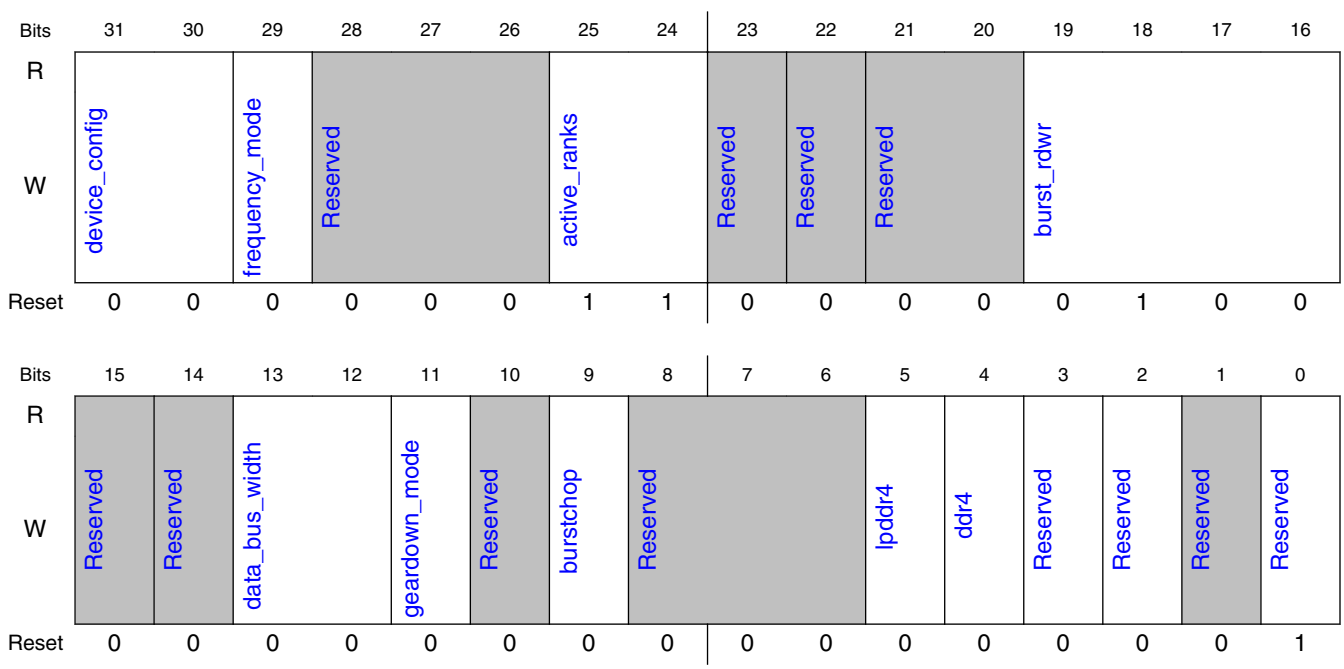
Offset	Register	Width (In bits)	Access	Reset value
2138h	[SHADOW] SDRAM Timing Register 14 (DRAMTMG14_SHADOW)	32	RW	0000_00A0h
213Ch	[SHADOW] SDRAM Timing Register 15 (DRAMTMG15_SHADOW)	32	RW	0000_0000h
2180h	[SHADOW] ZQ Control Register 0 (ZQCTL0_SHADOW)	32	RW	0200_0040h
2190h	[SHADOW] DFI Timing Register 0 (DFITMG0_SHADOW)	32	RW	0702_0002h
2194h	[SHADOW] DFI Timing Register 1 (DFITMG1_SHADOW)	32	RW	0000_0404h
21B4h	[SHADOW] DFI Timing Register 2 (DFITMG2_SHADOW)	32	RW	0000_0202h
21B8h	[SHADOW] DFI Timing Register 3 (DFITMG3_SHADOW)	32	RW	0000_0000h
2240h	[SHADOW] ODT Configuration Register (ODTCFG_SHADOW)	32	RW	0400_0400h

9.2.5.1.2 Master Register0 (MSTR)

9.2.5.1.2.1 Offset

Register	Offset
MSTR	0h

9.2.5.1.2.2 Diagram



9.2.5.1.2.3 Fields

Field	Function
31-30 device_config	Indicates the configuration of the device used in the system. 00b - Reserved 01b - Reserved 10b - x16 device (Only Valid for DDR4) 11b - x32 device
29 frequency_mode	Choose which registers are used. 0b - Original Registers 1b - Shadow Registers
28-26 —	Reserved
25-24 active_ranks	Only present for multi-rank configurations. Each bit represents one rank. For two-rank configurations, only bits[25:24] are present. • 1 - populated • 0 - unpopulated LSB is the lowest rank number. For 2 ranks following combinations are legal • 01 - One rank • 11 - Two ranks • Others- Reserved For 4 ranks following combinations are legal: • 0001 - One rank • 0011 - Two ranks • 1111 - Four ranks
23 —	Reserved
22 —	Reserved
21-20 —	Reserved
19-16 burst_rdwr	SDRAM burst length used All other values are reserved. This controls the burst size used to access the SDRAM. This must match the burst length mode register setting in the SDRAM. (For BC4/8 on-the-fly mode of DDR4, set this field to 0x0100) 0001b - Reserved 0010b - Burst length of 4 0100b - Burst length of 8 1000b - Burst length of 16 (only supported for LPDDR4)
15 —	Reserved
14 —	Reserved
13-12 data_bus_width	Selects proportion of DQ bus width that is used by the SDRAM - 00 - Full DQ bus width to SDRAM - 01 - Half DQ bus width to SDRAM - 10 - Quarter DQ bus width to SDRAM - 11 - Reserved Note that half bus

Table continues on the next page...

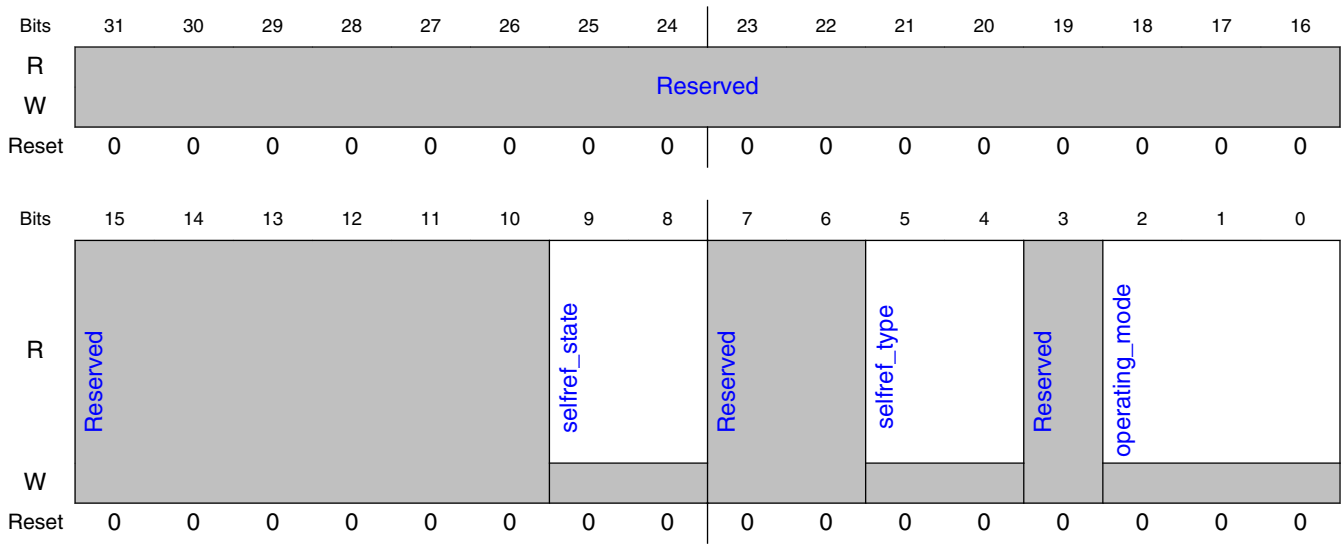
Field	Function
	width mode is only supported when the SDRAM bus width is a multiple of 16, and quarter bus width mode is only supported when the SDRAM bus width is a multiple of 32 and the configuration parameter MEMC_QBUS_SUPPORT is set. Bus width refers to DQ bus width (excluding any ECC width).
11 geardown_mode	1 indicates put the DRAM in geardown mode (2N) and 0 indicates put the DRAM in normal mode (1N). This register can be changed, only when the Controller is in self-refresh mode. This signal must be set the same value as MR3 bit A3. Note: Geardown mode is not supported if the configuration parameter MEMC_CMD_RTN2IDLE is set Note: Geardown mode is not supported if the configuration parameter DDRC_SHARED_AC is set (in Shared-AC mode) and the register value is don't care
10 —	Reserved
9 burstchop	When set, enable burst-chop (BC4 or 8 on-the-fly) in DDR4. Burst Chop for Reads is exercised only in HIF configurations (DDRC_INCL_ARB not set) and if in full bus width mode (MSTR.data_bus_width = 00) and if MEMC_BURST_LENGTH=8 or 16. Burst Chop for Writes is exercised only if Partial Writes enabled (DDRC_PARTIAL_WR=1) and if CRC is disabled (CRCPARCTL1.crc_enable = 0). If DDR4 CRC/parity retry is enabled (CRCPARCTL1.crc_parity_retry_enable = 1), burst chop is not supported, and this bit must be set to '0'. BC4 (fixed) mode is not supported.
8-6 —	Reserved
5 lpddr4	Select LPDDR4 SDRAM - 1 - LPDDR4 SDRAM device in use. - 0 - non-LPDDR4 device in use Present only in designs configured to support LPDDR4.
4 ddr4	Select DDR4 SDRAM - 1 - DDR4 SDRAM device in use. - 0 - non-DDR4 device in use Present only in designs configured to support DDR4.
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 —	Reserved

9.2.5.1.3 Operating Mode Status Register (STAT)

9.2.5.1.3.1 Offset

Register	Offset
STAT	4h

9.2.5.1.3.2 Diagram



9.2.5.1.3.3 Fields

Field	Function
31-10 —	Reserved
9-8 selfref_state	Self refresh state. This indicates self refresh or self refresh power down state for LPDDR4. This register is used for frequency change and MRR/MRW access during self refresh. 00b - SDRAM is not in Self Refresh. 01b - Self refresh 1 10b - Self refresh power down 11b - Self refresh
7-6 —	Reserved
5-4 selfref_type	Flags if Self Refresh (except LPDDR4) or SR-Powerdown (LPDDR4) is entered and if it was under Automatic Self Refresh control only or not. 00b - SDRAM is not in Self Refresh (except LPDDR4) or SR-Powerdown (LPDDR4). If retry is enabled by CRCPARCTRL1.crc_parity_retry_enable, this also indicates SRE command is still in parity error window or retry is in-progress. 10b - SDRAM is in Self Refresh (except LPDDR4) or SR-Powerdown (LPDDR4), which was not caused solely under Automatic Self Refresh control. It could have been caused by Hardware Low Power Interface and/or Software (reg_ddrc_selfref_sw). If retry is enabled, this guarantees SRE command is executed correctly without parity 11b - SDRAM is in Self Refresh (except LPDDR4) or SR-Powerdown (LPDDR4), which was caused by Automatic Self Refresh only. If retry is enabled, this guarantees SRE command is executed correctly without parity error.
3 —	Reserved
2-0 operating_mode	Operating mode This is 3-bits wide in configurations with LPDDR4/DDR4 support

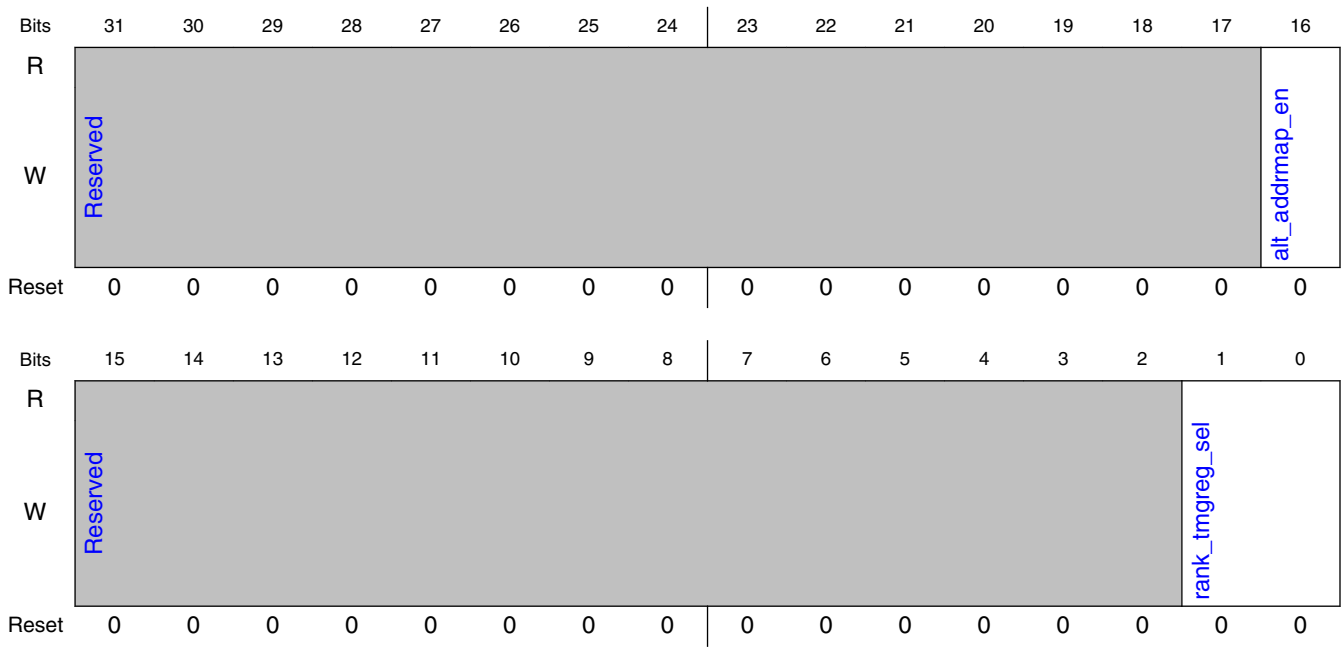
Field	Function
	DDR4 designs: • 000 - Init • 001 - Normal • 010 - Power-down • 011 - Self refresh • 1XX - Deep power-down / Maximum Power Saving Mode LPDDR4 designs: • 000 - Init • 001 - Normal • 010 - Power-down • 011 - Self refresh / Self refresh power-down

9.2.5.1.4 Operating Mode Status Register (MSTR1)

9.2.5.1.4.1 Offset

Register	Offset
MSTR1	8h

9.2.5.1.4.2 Diagram



9.2.5.1.4.3 Fields

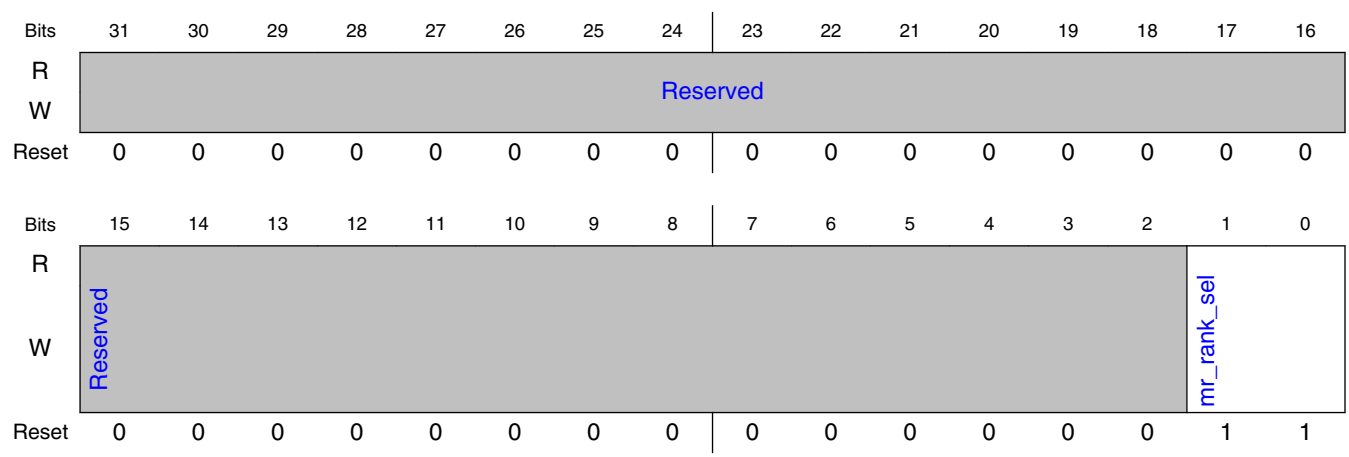
Field	Function
31-17 —	Reserved
16 alt_addrmap_en	Enable Alternative Address Map 0b - Disable Alternative Address Map 1b - Enable Alternative Address Map
15-2 —	Reserved
1-0 rank_tmgreg_sel	rank_tmgreg_sel Indicates which register set is used for each rank. Each bit represents one rank. (LSB is the lowest rank number.) **NOTE:** Only some of timing registers in DRAMTMGx are duplicated as MRAMTMGx for MRAM and its width is expanded for MRAM. This register can switch the duplicate registers, otherwise use DRAMTMGx. This register is only for DDR4 ST-MRAM rank, otherwise must be set to 0. This register must be set up while the Controller is in reset. 00b - USE DRAMTMGx registers for the rank 01b - USE MRAMTMGx registers for the rank

9.2.5.1.5 Operating Mode Status Register (MRCTRL3)

9.2.5.1.5.1 Offset

Register	Offset
MRCTRL3	Ch

9.2.5.1.5.2 Diagram



9.2.5.1.5.3 Fields

Field	Function
31-2 —	Reserved
1-0 mr_rank_sel	mr_rank_sel Controls which rank is accessed by MRCTRL0.mr_wr. Normally, it is desired to access all ranks, so all bits should be set to 1. Examples (assume uMCTL2 is configured for 8 ranks): • 0x01 - select rank 0 only • 0x02 - select rank 1 only • 0x05 - select ranks 0 and 2

9.2.5.1.6 Mode Register Read/Write Control Register 0. (MRCTRL0)

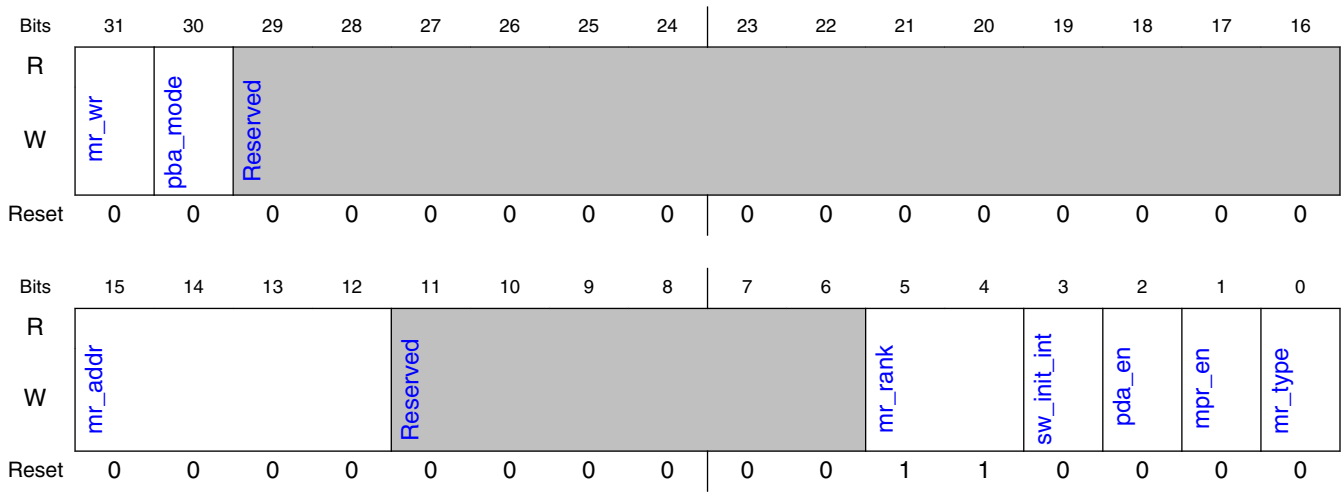
9.2.5.1.6.1 Offset

Register	Offset
MRCTRL0	10h

9.2.5.1.6.2 Function

Mode Register Read/Write Control Register 0. Note: Do not enable more than one of the following fields simultaneously: - sw_init_int - pda_en - mpr_en

9.2.5.1.6.3 Diagram



9.2.5.1.6.4 Fields

Field	Function
31 mr_wr	Setting this register bit to 1 triggers a mode register read or write operation. When the MR operation is complete, the DDRC automatically clears this bit. The other register fields of this register must be written in a separate APB transaction, before setting this mr_wr bit. It is recommended NOT to set this signal if in Init, Deep power-down or MPSM operating modes.
30 pba_mode	Indicates whether PBA access is executed. When setting this bit to 1 along with setting pda_en to 1, DDRC initiates PBA access instead of PDA access. - 0 - Per DRAM Addressability mode - 1 - Per Buffer Addressability mode The completion of PBA access is confirmed by MRSTAT.pda_done in the same way as PDA.
29-16 —	Reserved
15-12 mr_addr	Address of the mode register that is to be written to. Don't Care for LPDDR4 (see MRCTRL1.mr_data for mode register addressing in LPDDR4).In case of DDR4, the bit[3:2] corresponds to the bank group bits. 0000b - MR0 0001b - MR1 0010b - MR2 0011b - MR3 0100b - MR4 0101b - MR5 0110b - MR6 0111b - MR7
11-6 —	Reserved
5-4 mr_rank	Controls which rank is accessed by MRCTRL0.mr_wr. Normally, it is desired to access all ranks, so all bits should be set to 1.Examples (assume DDRC is configured for 4 ranks): 0x1 - select rank 0 only 0x2 - select rank 1 only 0x5 - select ranks 0 and 2 0xA - select ranks 1 and 3 0xF - select ranks 0, 1, 2 and 3
3 sw_init_int	Indicates whether Software intervention is allowed via MRCTRL0/MRCTRL1 before automatic SDRAM initialization routine or not. For DDR4, this bit can be used to initialize the DDR4 RCD (MR7) before

Table continues on the next page...

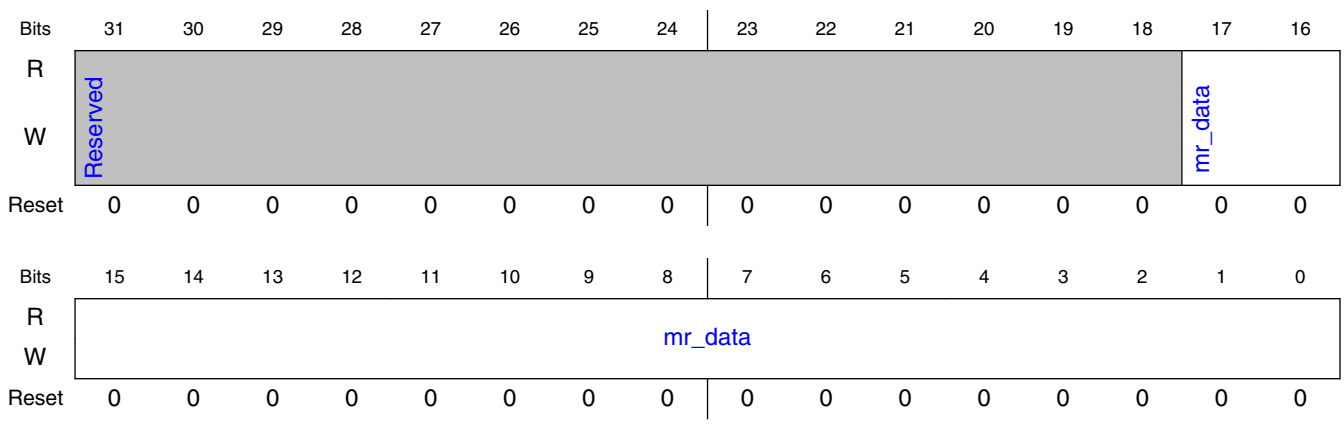
Field	Function
	automatic SDRAM initialization. For LPDDR4, this bit can be used to program additional mode registers before automatic SDRAM initialization if necessary. In LPDDR4 independent channel mode, note that this must be programmed to both channels beforehand. Note that this must be cleared to 0 after completing Software operation. Otherwise, SDRAM initialization routine will not re-start. 0b - Software intervention is not allowed 1b - Software intervention is allowed
2 pda_en	Indicates whether the mode register operation is MRS in PDA mode or not. Note that when pba_mode=1, PBA access is initiated instead of PDA access. 0b - MRS 1b - MRS in Per DRAM Addressability
1 mpr_en	Indicates whether the mode register operation is MRS or WR/RD for MPR (only supported for DDR4). 0b - MRS 1b - WR/RD for MPR
0 mr_type	Indicates whether the mode register operation is read or write. 0b - Write 1b - Read

9.2.5.1.7 Mode Register Read/Write Control Register 1 (MRCTRL1)

9.2.5.1.7.1 Offset

Register	Offset
MRCTRL1	14h

9.2.5.1.7.2 Diagram



9.2.5.1.7.3 Fields

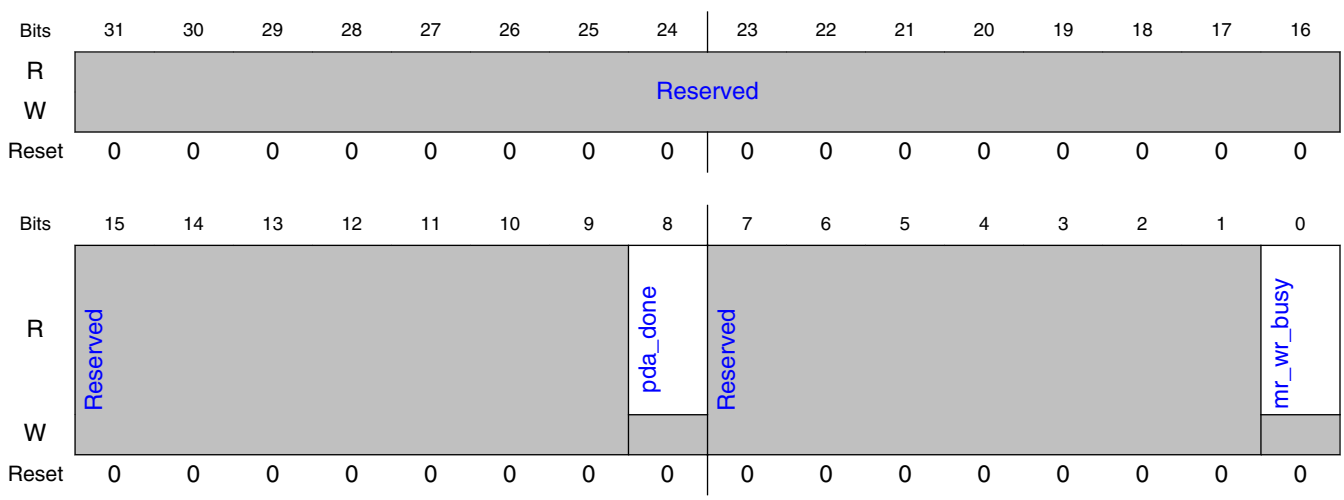
Field	Function
31-18 —	Reserved
17-0 mr_data	Mode register write data for all non-LPDDR4 modes. For LPDDR4, MRCTRL1[15:0] are interpreted as [15:8] MR Address [7:0] MR data for writes, don't care for reads. This is 18-bits wide in configurations with DDR4 support and 16-bits in all other configurations.

9.2.5.1.8 Mode Register Read/Write Status Register (MRSTAT)

9.2.5.1.8.1 Offset

Register	Offset
MRSTAT	18h

9.2.5.1.8.2 Diagram



9.2.5.1.8.3 Fields

Field	Function
31-9 —	Reserved
8 pda_done	The SoC core may initiate a MR write operation in PDA/PBA mode only if this signal is low. This signal goes high when three consecutive MRS commands related to the PDA/PBA mode are issued to the SDRAM. This signal goes low when MRCTRL0.pda_en becomes 0. Therefore, it is recommended to

Table continues on the next page...

Field	Function
	write MRCTRL0.pda_en to 0 after this signal goes high in order to prepare to perform PDA operation next time 0b - Indicates that mode register write operation related to PDA/PBA is in progress or has not started yet. 1b - Indicates that mode register write operation related to PDA/PBA has competed.
7-1 —	Reserved
0 mr_wr_busy	The SoC core may initiate a MR write operation only if this signal is low. This signal goes high in the clock after the DDRC accepts the MRW/MRR request. It goes low when the MRW/MRR command is issued to the SDRAM. It is recommended not to perform MRW/MRR commands when 'MRSTAT.mr_wr_busy' is high. 0b - Indicates that the SoC core can initiate a mode register write operation 1b - Indicates that mode register write operation is in progress

9.2.5.1.9 Mode Register Read/Write Control Register 2 (MRCTRL2)

9.2.5.1.9.1 Offset

Register	Offset
MRCTRL2	1Ch

9.2.5.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr_device_sel															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	mr_device_sel															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.9.3 Fields

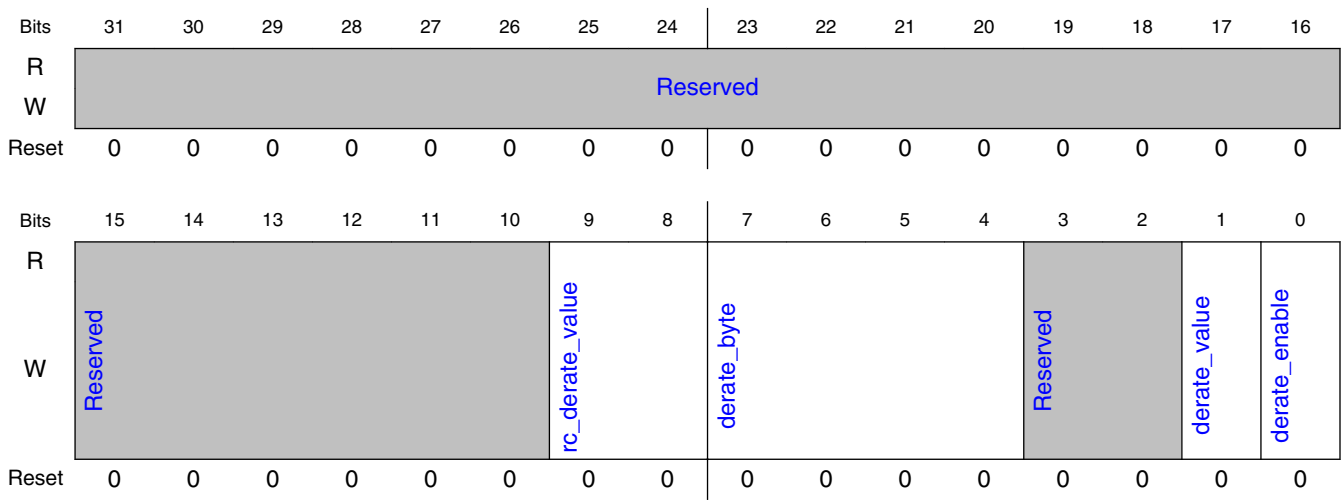
Field	Function
31-0 mr_device_sel	Indicates the device(s) to be selected during the MRS that happens in PDA mode. Each bit is associated with one device. For example, bit[0] corresponds to Device 0, bit[1] to Device 1 etc. A '1' should be programmed to indicate that the MRS command should be applied to that device. A '0' indicates that the MRS commands should be skipped for that device.

9.2.5.1.10 Temperature Derate Enable Register (DERATEEN)

9.2.5.1.10.1 Offset

Register	Offset
DERATEEN	20h

9.2.5.1.10.2 Diagram



9.2.5.1.10.3 Fields

Field	Function
31-10 —	Reserved
9-8 rc_derate_value	Derate value of tRC for LPDDR4. Present only in designs configured to support LPDDR4. The required number of cycles for derating can be determined by dividing 3.75ns by the core_ddrc_core_clk period, and rounding up the next integer. 00b - Derating uses +1 01b - Derating uses +2 10b - Derating uses +3 11b - Derating uses +4
7-4 derate_byte	Derate byte Present only in designs configured to support LPDDR4. Indicates which byte of the MRR data is used for derating. The maximum valid value depends on MEMC_DRAM_TOTAL_DATA_WIDTH.
3-2 —	Reserved

Table continues on the next page...

Field	Function
1 derate_value	Derate value. Present only in designs configured to support LPDDR4. For LPDDR4, if the period of core_ddrc_core_clk is less than 1.875ns, this register field should be set to 1; otherwise it should be set to 0. 0b - Derating uses +1 1b - Derating uses +2
0 derate_enable	Enables derating. Present only in designs configured to support LPDDR4. This field must be set to '0' for non-LPDDR4 mode. 0b - Timing parameter derating is disabled 1b - Timing parameter derating is enabled using MR4 read value.

9.2.5.1.11 Temperature Derate Interval Register (DERATEINT)

9.2.5.1.11.1 Offset

Register	Offset
DERATEINT	24h

9.2.5.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr4_read_interval															
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	mr4_read_interval															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.11.3 Fields

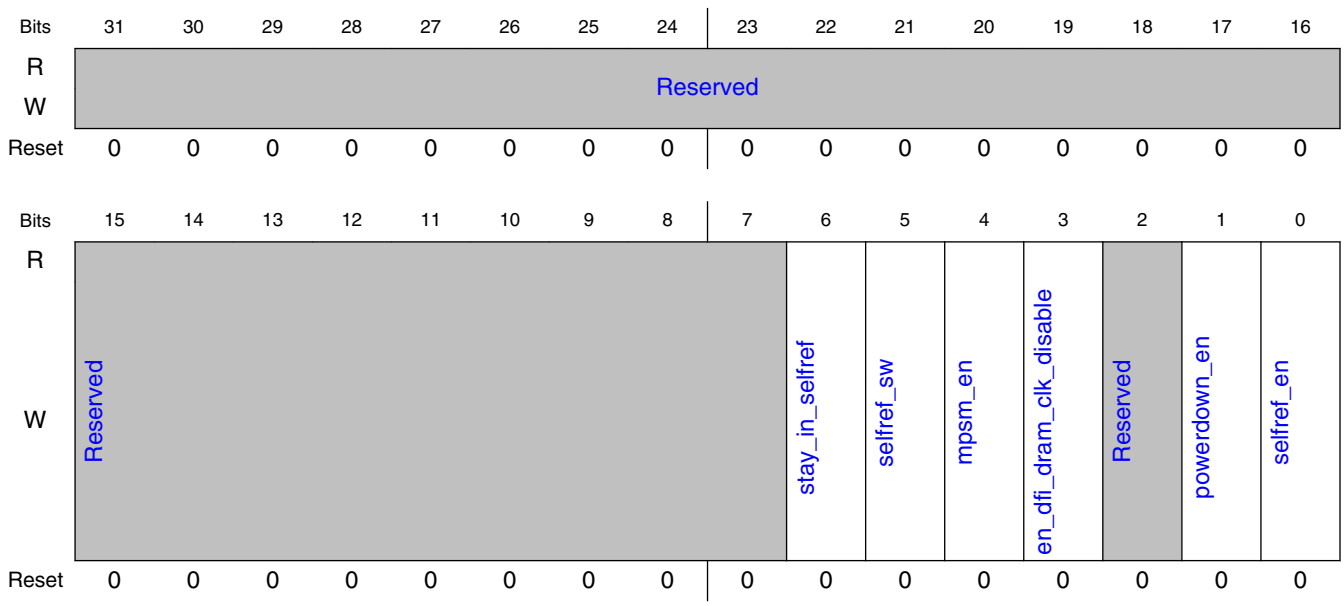
Field	Function
31-0 mr4_read_interval	Interval between two MR4 reads, used to derate the timing parameters. Present only in designs configured to support LPDDR4. This register must not be set to zero. Unit: DFI clock cycle.

9.2.5.1.12 Low Power Control Register (PWRCTL)

9.2.5.1.12.1 Offset

Register	Offset
PWRCTL	30h

9.2.5.1.12.2 Diagram



9.2.5.1.12.3 Fields

Field	Function
31-7 —	Reserved
6 stay_in_selfref	Self refresh state is an intermediate state to enter to Self refresh power down state or exit Self refresh power down state for LPDDR4. This register controls transition from the Self refresh state. - 1 - Prohibit transition from Self refresh state - 0 - Allow transition from Self refresh state 0b - 1b -
5 selfref_sw	A value of 1 to this register causes system to move to Self Refresh state immediately, as long as it is not in INIT or DPD/MPSM operating_mode. This is referred to as Software Entry/Exit to Self Refresh. 0b - Software Exit from Self Refresh 1b - Software Entry to Self Refresh
4 mpsm_en	When this is 1, the DDRC puts the SDRAM into maximum power saving mode when the transaction store is empty. This register must be reset to '0' to bring DDRC out of maximum power saving mode. Present only in designs configured to support DDR4. For non-DDR4, this register should not be set to 1. Note that

Table continues on the next page...

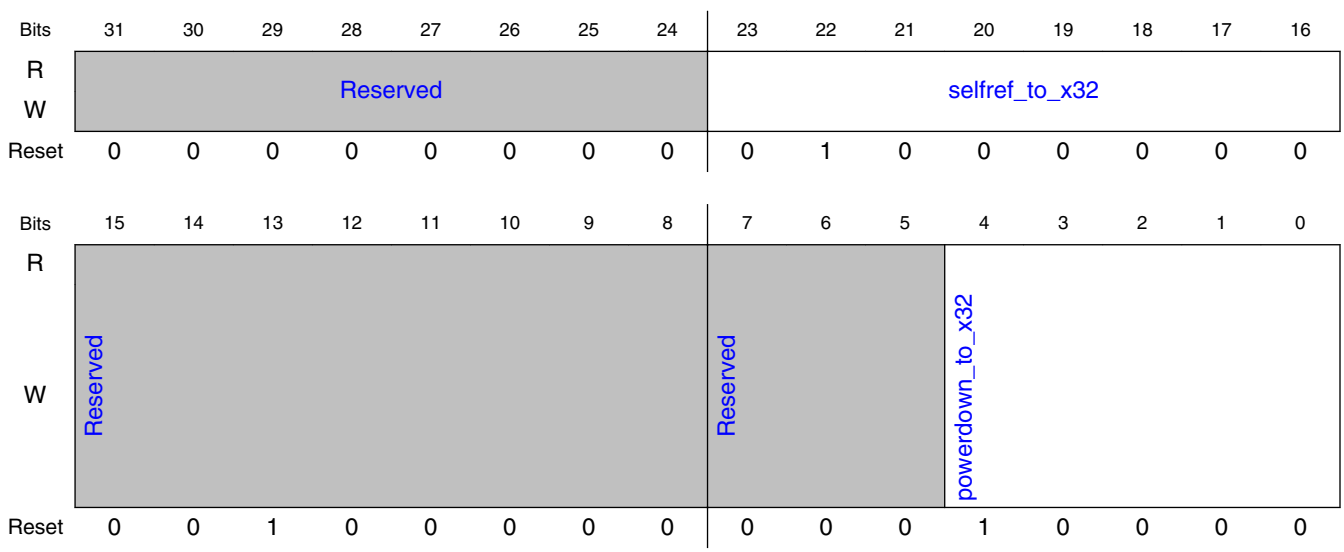
Field	Function
	MPSM is not supported when using a DDR PHY, if the PHY parameter DDRC_AC_CS_USE is disabled, as the MPSM exit sequence requires the chip-select signal to toggle. FOR PERFORMANCE ONLY.
3 en_dfi_dram_clk_disable	Enable the assertion of dfi_dram_clk_disable whenever a clock is not required by the SDRAM. If set to 0, dfi_dram_clk_disable is never asserted. Assertion of dfi_dram_clk_disable is as follows: In DDR4, can be asserted in following: in Self Refresh in Maximum Power Saving Mode In LPDDR4, can be asserted in following: in Self Refresh Power Down in Power Down during Normal operation (Clock Stop)
2 —	Reserved
1 powerdown_en	If true then the DDRC goes into power-down after a programmable number of cycles "maximum idle clocks before power down" (PWRTMG.powerdown_to_x32). This register bit may be re-programmed during the course of normal operation.
0 selfref_en	If true then the DDRC puts the SDRAM into Self Refresh after a programmable number of cycles "maximum idle clocks before Self Refresh (PWRTMG.selfref_to_x32)". This register bit may be re-programmed during the course of normal operation.

9.2.5.1.13 Low Power Timing Register (PWRTMG)

9.2.5.1.13.1 Offset

Register	Offset
PWRTMG	34h

9.2.5.1.13.2 Diagram



9.2.5.1.13.3 Fields

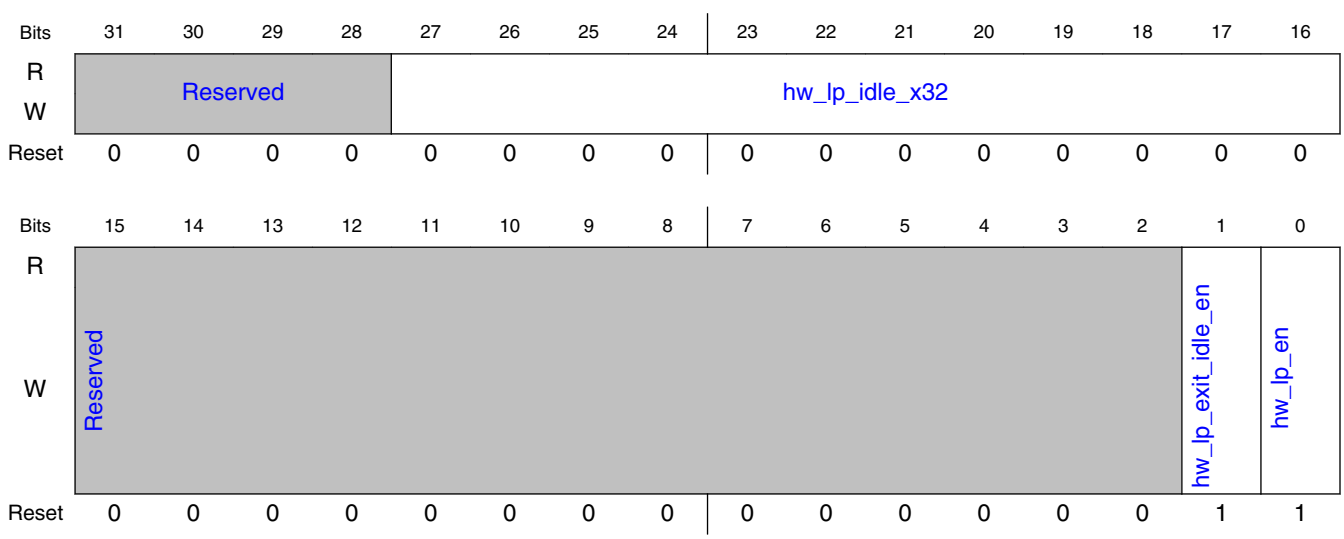
Field	Function
31-24 —	Reserved
23-16 selfref_to_x32	After this many clocks of the DDRC command channel being idle the DDRC automatically puts the SDRAM into Self Refresh. The DDRC command channel is considered idle when there are no HIF commands outstanding. This must be enabled in the PWRCTL.selfref_en. Unit: Multiples of 32 DFI clocks. FOR PERFORMANCE ONLY.
15-8 —	Reserved
7-5 —	Reserved
4-0 powerdown_to_x32	After this many clocks of the DDRC command channel being idle the DDRC automatically puts the SDRAM into power-down. The DDRC command channel is considered idle when there are no HIF commands outstanding. This must be enabled in the PWRCTL.powerdown_en. Unit: Multiples of 32 DFI clocks FOR PERFORMANCE ONLY.

9.2.5.1.14 Hardware Low Power Control Register (HWLPCTL)

9.2.5.1.14.1 Offset

Register	Offset
HWLPCTL	38h

9.2.5.1.14.2 Diagram



9.2.5.1.14.3 Fields

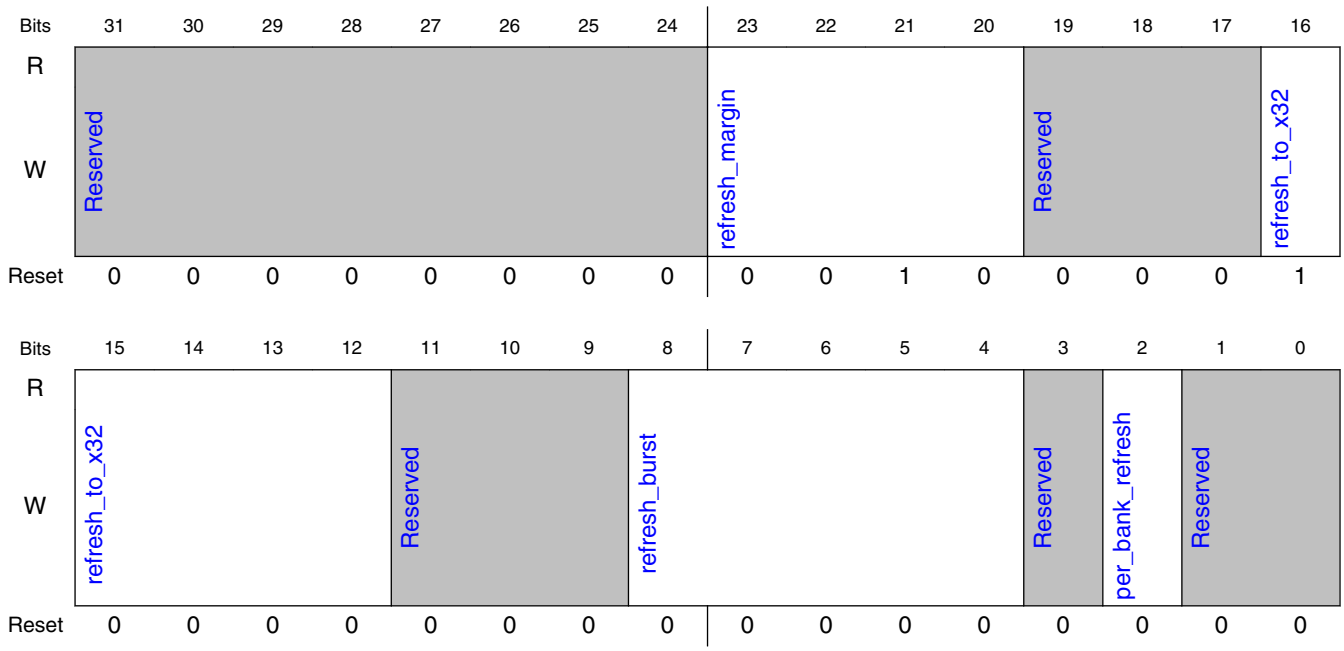
Field	Function
31-28 —	Reserved
27-16 hw_lp_idle_x32	Hardware idle period. The cactive_ddrc output is driven low if the DDRC command channel is idle for hw_lp_idle * 32 cycles if not in INIT or DPD/MPSM operating_mode. The DDRC command channel is considered idle when there are no HIF commands outstanding. The hardware idle function is disabled when hw_lp_idle_x32=0. Unit: Multiples of 32 DFI clocks. FOR PERFORMANCE ONLY.
15-2 —	Reserved
1 hw_lp_exit_idle_en	When this bit is programmed to 1 the cactive_in_ddrc pin of the DDRC can be used to exit from the automatic clock stop, automatic power down or automatic self-refresh modes. Note, it will not cause exit of Self-Refresh that was caused by Hardware Low Power Interface and/or Software (PWRCTL.selfref_sw).
0 hw_lp_en	Enable for Hardware Low Power Interface.

9.2.5.1.15 Refresh Control Register 0 (RFSHCTL0)

9.2.5.1.15.1 Offset

Register	Offset
RFSHCTL0	50h

9.2.5.1.15.2 Diagram



9.2.5.1.15.3 Fields

Field	Function
31-24 —	Reserved
23-20 refresh_margin	Threshold value in number of DFI clock cycles before the critical refresh or page timer expires. A critical refresh is to be issued before this threshold is reached. It is recommended that this not be changed from the default value, currently shown as 0x2. It must always be less than internally used t_rfc_nom_x32. Note that, in LPDDR4, internally used t_rfc_nom_x32 may be equal to RFSHTMG.t_rfc_nom_x32>>2 if derating is enabled (DERATEEN.derate_enable=1). Otherwise, internally used t_rfc_nom_x32 will be equal to RFSHTMG.t_rfc_nom_x32. Unit: Multiples of 32 DFI clocks.
19-17 —	Reserved
16-12 refresh_to_x32	If the refresh timer (tRFCnom, also known as tREFI) has expired at least once, but it has not expired (RFSHCTL0.refresh_burst+1) times yet, then a speculative refresh may be performed. A speculative refresh is a refresh performed at a time when refresh would be useful, but before it is absolutely required. When the SDRAM bus is idle for a period of time determined by this RFSHCTL0.refresh_to_x32 and the refresh timer has expired at least once since the last refresh, then a speculative refresh is performed. Speculative refreshes continues successively until there are no refreshes pending or until new reads or writes are issued to the DDRC. FOR PERFORMANCE ONLY. Unit: Multiples of 32 DFI clocks.
11-9 —	Reserved
8-4 refresh_burst	The programmed value + 1 is the number of refresh timeouts that is allowed to accumulate before traffic is blocked and the refreshes are forced to execute. Closing pages to perform a refresh is a one-time penalty that must be paid for each group of refreshes. Therefore, performing refreshes in a burst reduces the per-refresh penalty of these page closings. Higher numbers for RFSHCTL.refresh_burst slightly

Table continues on the next page...

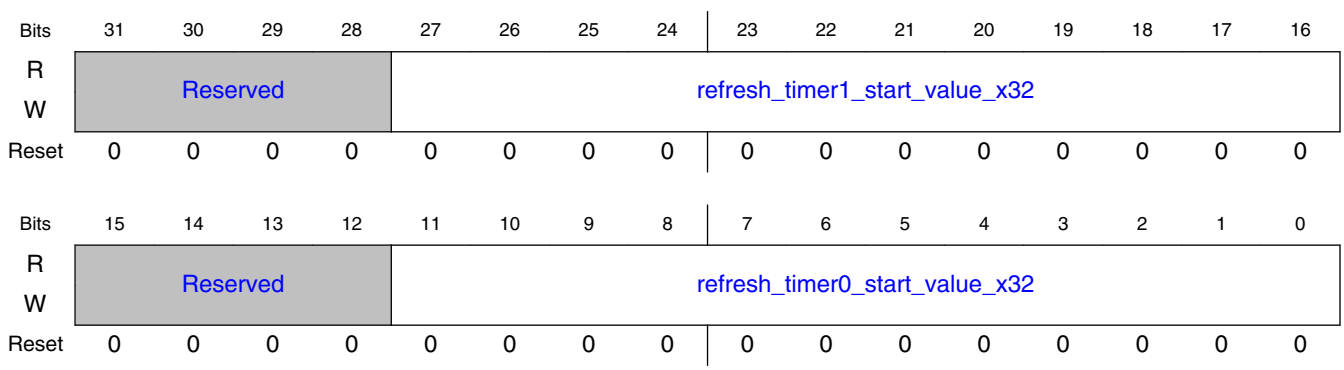
Field	Function
	increases utilization; lower numbers decreases the worst-case latency associated with refreshes. - 0 - single refresh - 1 - burst-of-2 refresh - 7 - burst-of-8 refresh. In DDR4 mode, according to Fine Granularity feature, 8 refreshes can be postponed in 1X mode, 16 refreshes in 2X mode and 32 refreshes in 4X mode. If using PHY-initiated updates, care must be taken in the setting of RFSHCTL0.refresh_burst, to ensure that tRFCmax is not violated due to a PHY-initiated update occurring shortly before a refresh burst was due. In this situation, the refresh burst will be delayed until the PHY-initiated update is complete.
3 —	Reserved
2 per_bank_refresh	Per bank refresh allows traffic to flow to other banks. Present only in designs configured to support LPDDR4. 0b - All bank refresh 1b - Per bank refresh
1-0 —	Reserved

9.2.5.1.16 Refresh Control Register 1 (RFSHCTL1)

9.2.5.1.16.1 Offset

Register	Offset
RFSHCTL1	54h

9.2.5.1.16.2 Diagram



9.2.5.1.16.3 Fields

Field	Function
31-28 —	Reserved

Table continues on the next page...

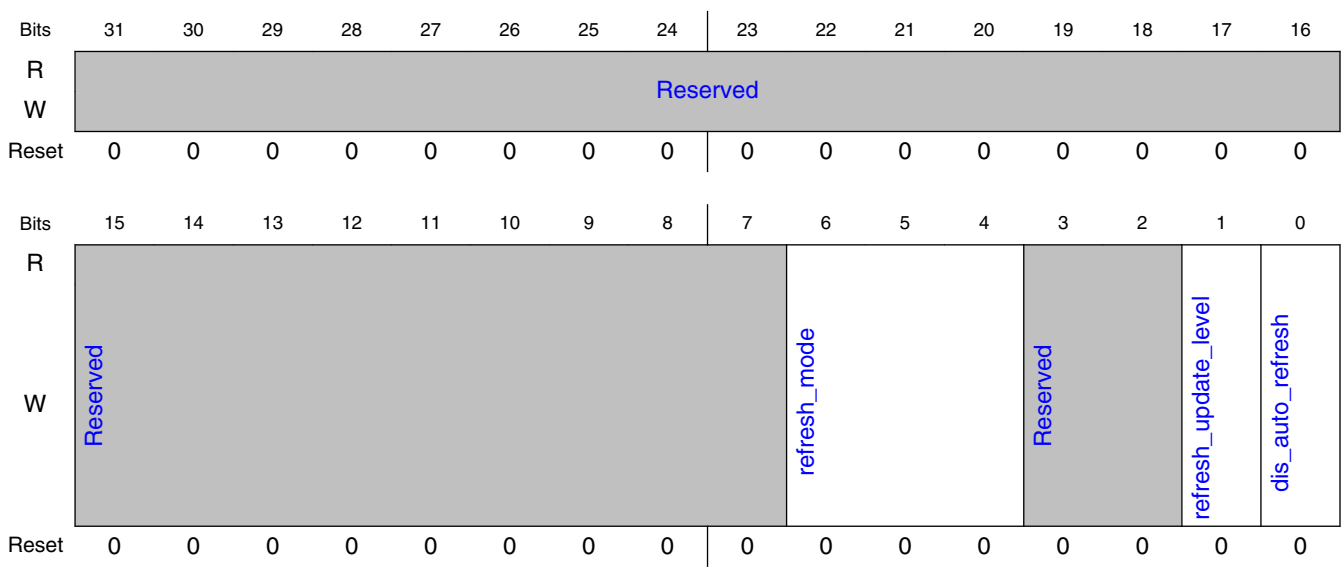
Field	Function
27-16 refresh_timer1_start_value_x32	Refresh timer start for rank 1 (only present in multi-rank configurations). This is useful in staggering the refreshes to multiple ranks to help traffic to proceed. This is explained in Refresh Controls section of architecture chapter. Unit: Multiples of 32 DFI clock cycles. FOR PERFORMANCE ONLY.
15-12 —	Reserved
11-0 refresh_timer0_start_value_x32	Refresh timer start for rank 0 (only present in multi-rank configurations). This is useful in staggering the refreshes to multiple ranks to help traffic to proceed. This is explained in Refresh Controls section of architecture chapter. Unit: Multiples of 32 DFI clock cycles. FOR PERFORMANCE ONLY.

9.2.5.1.17 Refresh Control Register 3 (RFSHCTL3)

9.2.5.1.17.1 Offset

Register	Offset
RFSHCTL3	60h

9.2.5.1.17.2 Diagram



9.2.5.1.17.3 Fields

Field	Function
31-7	Reserved

Table continues on the next page...

Field	Function
—	
6-4 refresh_mode	Fine Granularity Refresh Mode - 000 - Fixed 1x (Normal mode) - 001 - Fixed 2x - 010 - Fixed 4x - 101 - Enable on the fly 2x (not supported) - 110 - Enable on the fly 4x (not supported) - Everything else - reserved Note: Only Fixed 1x mode is supported if RFSHCTL3.dis_auto_refresh = 1. Note: The on-the-fly modes are not supported in this version of the DDRC. Note: This must be set up while the Controller is in reset or while the Controller is in self-refresh mode. Changing this during normal operation is not allowed. Making this a dynamic register will be supported in future version of the DDRC. Note: This register field has effect only if a DDR4 SDRAM device is in use (MSTR.ddr4 = 1).
3-2 —	Reserved
1 refresh_update_level	Toggle this signal (either from 0 to 1 or from 1 to 0) to indicate that the refresh register(s) have been updated. refresh_update_level must not be toggled when the DDRC is in reset (core_ddrc_rstn = 0). The refresh register(s) are automatically updated when exiting reset.
0 dis_auto_refresh	When '1', disable auto-refresh generated by the DDRC. When auto-refresh is disabled, the SoC core must generate refreshes using the registers reg_ddrc_rank0_refresh, reg_ddrc_rank1_refresh, reg_ddrc_rank2_refresh and reg_ddrc_rank3_refresh. When dis_auto_refresh transitions from 0 to 1, any pending refreshes are immediately scheduled by the DDRC. If DDR4 CRC/parity retry is enabled (CRCPARCTL1.crc_parity_retry_enable = 1), disable auto-refresh is not supported, and this bit must be set to '0'. (DDR4 only) If FGR mode is enabled (RFSHCTL3.refresh_mode > 0), disable auto-refresh is not supported, and this bit must be set to '0'. This register field is changeable on the fly.

9.2.5.1.18 Refresh Timing Register (RFSHTMG)

9.2.5.1.18.1 Offset

Register	Offset
RFSHTMG	64h

9.2.5.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				t_rfc_nom_x32											
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						t_rfc_min									
Reset	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0

9.2.5.1.18.3 Fields

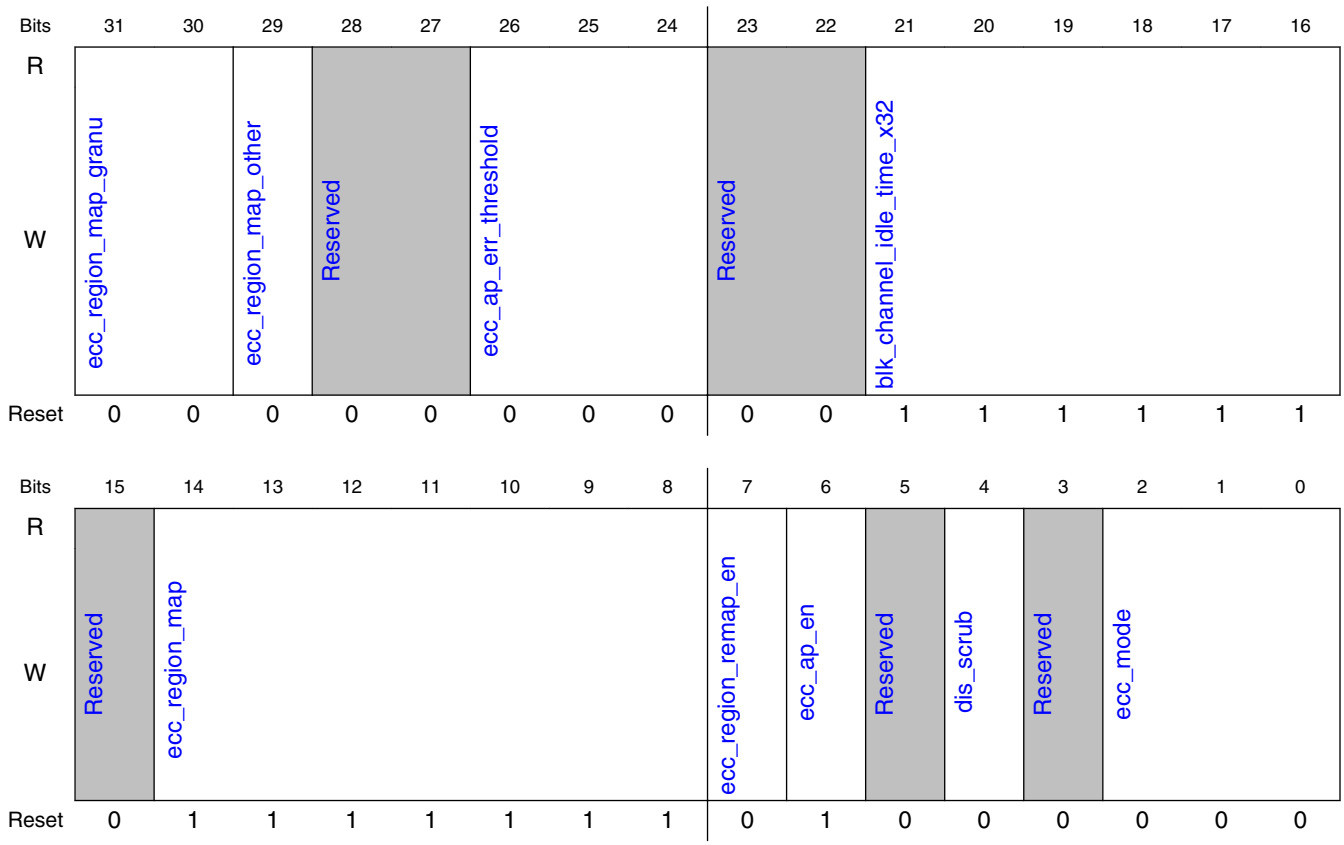
Field	Function
31-28 —	Reserved
27-16 t_rfc_nom_x32	tREFI: Average time interval between refreshes per rank (Specification: 7.8us for DDR4. See JEDEC specification for LPDDR4). For LPDDR4: - if using all-bank refreshes (RFSHCTL0.per_bank_refresh = 0), this register should be set to tREFIab - if using per-bank refreshes (RFSHCTL0.per_bank_refresh = 1), this register should be set to tREFIpb When the controller is operating in 1:2 frequency ratio mode, program this to (tREFI/2), no rounding up. In DDR4 mode, tREFI value is different depending on the refresh mode. The user should program the appropriate value from the spec based on the value programmed in the refresh mode register. Note that RFSHTMG.t_rfc_nom_x32 * 32 must be greater than RFSHTMG.t_rfc_min, and RFSHTMG.t_rfc_nom_x32 must be greater than 0x1. - Non-DDR4 or DDR4 Fixed 1x mode: RFSHTMG.t_rfc_nom_x32 must be less than or equal to 0xFFE. - DDR4 Fixed 2x mode: RFSHTMG.t_rfc_nom_x32 must be less than or equal to 0x7FF. - DDR4 Fixed 4x mode: RFSHTMG.t_rfc_nom_x32 must be less than or equal to 0x3FF. Unit: Multiples of 32 clocks.
15-10 —	Reserved
9-0 t_rfc_min	tRFC (min): Minimum time from refresh to refresh or activate. When the controller is operating in 1:1 mode, t_rfc_min should be set to RoundUp(tRFCmin/tCK). When the controller is operating in 1:2 mode, t_rfc_min should be set to RoundUp(RoundUp(tRFCmin/tCK)/2). In LPDDR4 mode: - if using all-bank refreshes, the tRFCmin value in the above equations is equal to tRFCab - if using per-bank refreshes, the tRFCmin value in the above equations is equal to tRFCpb. In DDR4 mode, the tRFCmin value in the above equations is different depending on the refresh mode (fixed 1X,2X,4X) and the device density. The user should program the appropriate value from the spec based on the 'refresh_mode' and the device density that is used. Unit: Clocks.

9.2.5.1.19 ECC Configuration Register 0 (ECCCFG0)

9.2.5.1.19.1 Offset

Register	Offset
ECCCFG0	70h

9.2.5.1.19.2 Diagram



9.2.5.1.19.3 Fields

Field	Function
31-30 ecc_region_map_granu	Indicates granularity of selectable protected region Define one region size for ECCCFG0.ecc_region_map. 0 - 1/8 of memory spaces 1 - 1/16 of memory spaces 2 - 1/32 of memory spaces 3 - 1/64 of memory spaces
29 ecc_region_map_other	When ECCCFG0[ecc_region_map_granu] > 0, there is a region which is not controlled by ecc_region_map. This register defines the region to be protected or non-protected for Inline ECC. This register is valid only when ECCCFG0[ecc_region_map_granu]>0 && ECCCFG0[ecc_mode]=4. 0 - Non-Protected 1 - Protected
28-27 —	Reserved
26-24	Sets threshold for address parity error.

Table continues on the next page...

Field	Function
ecc_ap_err_threshold	ECCAPSTAT[ecc_ap_err] is asserted if number of ECC errors (correctable/uncorrectable) within one burst exceeds this threshold. This register value must be less than "Total number of ECC checks within one burst" when this feature is used, "Total number of ECC check within one burst" is calculated by (DRAM Data width) x (DRAM BL) / 64.
23-22 —	Reserved
21-16 blk_channel_idle_time_x32	Indicates the number of cycles on HIF interface with no access to protected regions which causes flush of all the block channels. In order to flush block channel, the controller injects write ECC command (when there is no incoming HIF command) if there is any write in the block and then stop tracking the block address. Unit: Multiples of 32 DFI clock cycles. 0 - Indicates no timeout (feature is disabled, not supported with this version) 1 - Indicates 32 cycles 2 - Indicates 2*32 cycles, and so on
15 —	Reserved
14-8 ecc_region_map	Selectable Protected Region setting. Memory space is divided to 8/16/32/64 regions which is determined by ECCCFG0.ecc_region_map_granu. Note: Highest 1/8 memory space is always ECC region. Lowest 7 regions are Selectable Protected Regions. The Selectable Protected Regions can be protected/non-protected selectively by ecc_region_map[6:0]. Other upper regions are non-protected region if any. Each bit of ecc_region_map[6:0] correspond to each of lowest 7 regions respectively. In order to protect a region with ECC, set the corresponding bit to 1, otherwise set to 0. All "0"s is invalid - there must be at least one protected region if inline ECC is enabled through ECCCFG0[ecc_mode] register. All regions are protected with the following setting. ecc_region_map=7'b1111111 ecc_region_map_granu=0 Only first 1/64 region is protected with the following setting. ecc_region_map=7'b0000001 ecc_region_map_granu=3
7 ecc_region_remap_en	Enables remapping ECC region feature. Only supported when inline ECC is enabled. 0 - Disable 1 - Enable
6 ecc_ap_en	Enables address protection feature. Only supported when inline ECC is enabled. 0 - Disable 1 - Enable
5 —	Reserved
4 dis_scrub	Disables ECC scrubs. Valid only when ECCCFG0[ecc_mode] = 3'b100 or 3'b101. **NOTE:** Scrub is not supported in inline ECC mode and the register value is don't care.
3 —	Reserved

Table continues on the next page...

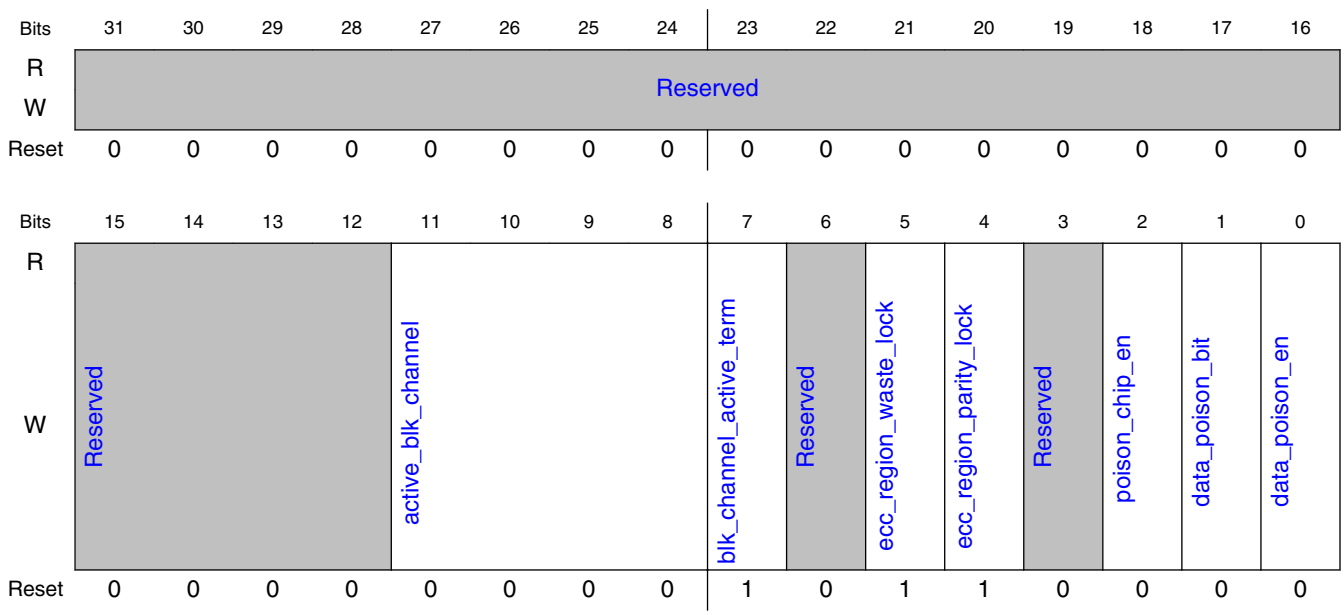
Field	Function
2-0 ecc_mode	ECC mode indicator. 000 - ECC disabled 100 - ECC enabled - SEC/DED over 1 beat 101 - ECC enabled - Advanced ECC X4/X8 all other settings are reserved for future use

9.2.5.1.20 ECC Configuration Register 1 (ECCCFG1)

9.2.5.1.20.1 Offset

Register	Offset
ECCCFG1	74h

9.2.5.1.20.2 Diagram



9.2.5.1.20.3 Fields

Field	Function
31-12 —	Reserved

Table continues on the next page...

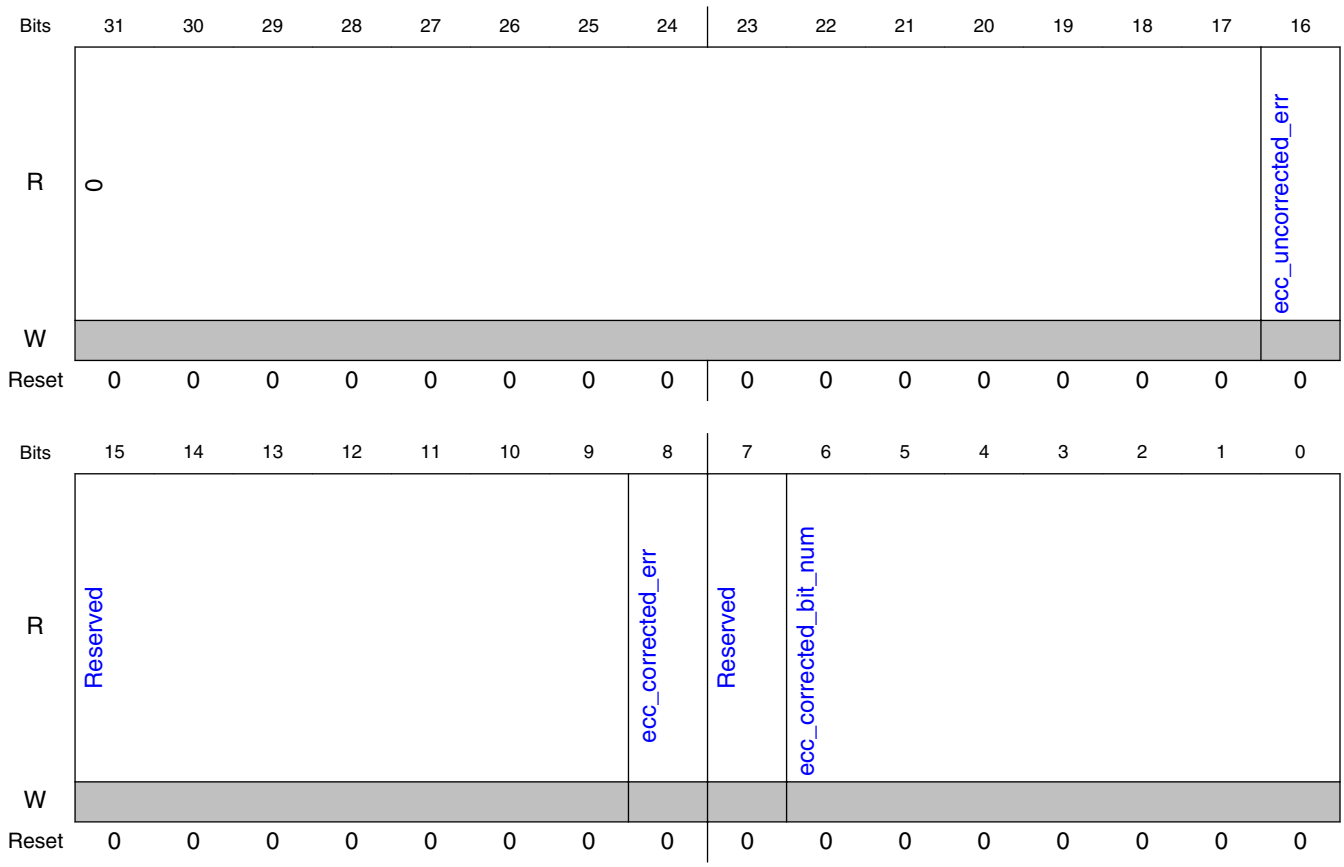
Field	Function
11-8 active_blk_channel	Indicated the number of active block channels. Total number of ECC block channels are defined by MEMC_NO_OF_BLK_CHANNEL hardware parameter. This register can limit the number of available channels. For example, if set to 0, only one channel is active and therefore block interleaving is disabled. The valid range is from 0 to MEMC_NO_OF_BLK_CHANNEL-1.
7 blk_channel_active_term	If enabled, block channel is terminated when full block write or full block read is performed (all address within block are written or read). This is debug register, and this must be set to 1 for normal operation. 0 - Disable (only for debug purpose) 1 - Enable (default)
6 —	Reserved
5 ecc_region_waste_lock	Locks the remaining waste parts of the ECC region (hole) that are not locked by ecc_region_parity_lock. 1 - Locked; if this region is accessed, error response is generated 0 - Unlocked; this region can be accessed normally, similar to non-ECC protected region
4 ecc_region_parity_lock	Locks the parity section of the ECC region (hole) which is the highest system address part of the memory that stores ECC parity for protected region. 1 - Locked; if this region is accessed, error response is generated 0 - Unlocked; this region can be accessed normally, similar to non-ECC protected region
3 —	Reserved
2 poison_chip_en	Indicates the data poison based on chip (that is, persistently poisons the DRAM data once its cs is selected to mimic chip failure). It is valid if ECCCFG1[data_poison_en]=1. 0 - Disable 1 - Enable. Persistently poison the write data to rank address that matches ECCPOISONADDR0[ecc_poison_rank].
1 data_poison_bit	Selects whether to poison 1 or 2 bits. 0 - 2-bit (uncorrectable) data poisoning 1 - 1-bit (correctable) data poisoning, if ECCCFG1[data_poison_en]=1
0 data_poison_en	Enables ECC data poisoning - introduces ECC errors on writes to address specified by the ECCPOISONADDR0/1 registers. This field must be set to 0 if ECC is disabled (ECCCFG0[ecc_mode] = 0).

9.2.5.1.21 SECDED ECC status register (ECCSTAT)

9.2.5.1.21.1 Offset

Register	Offset
ECCSTAT	78h

9.2.5.1.21.2 Diagram



9.2.5.1.21.3 Fields

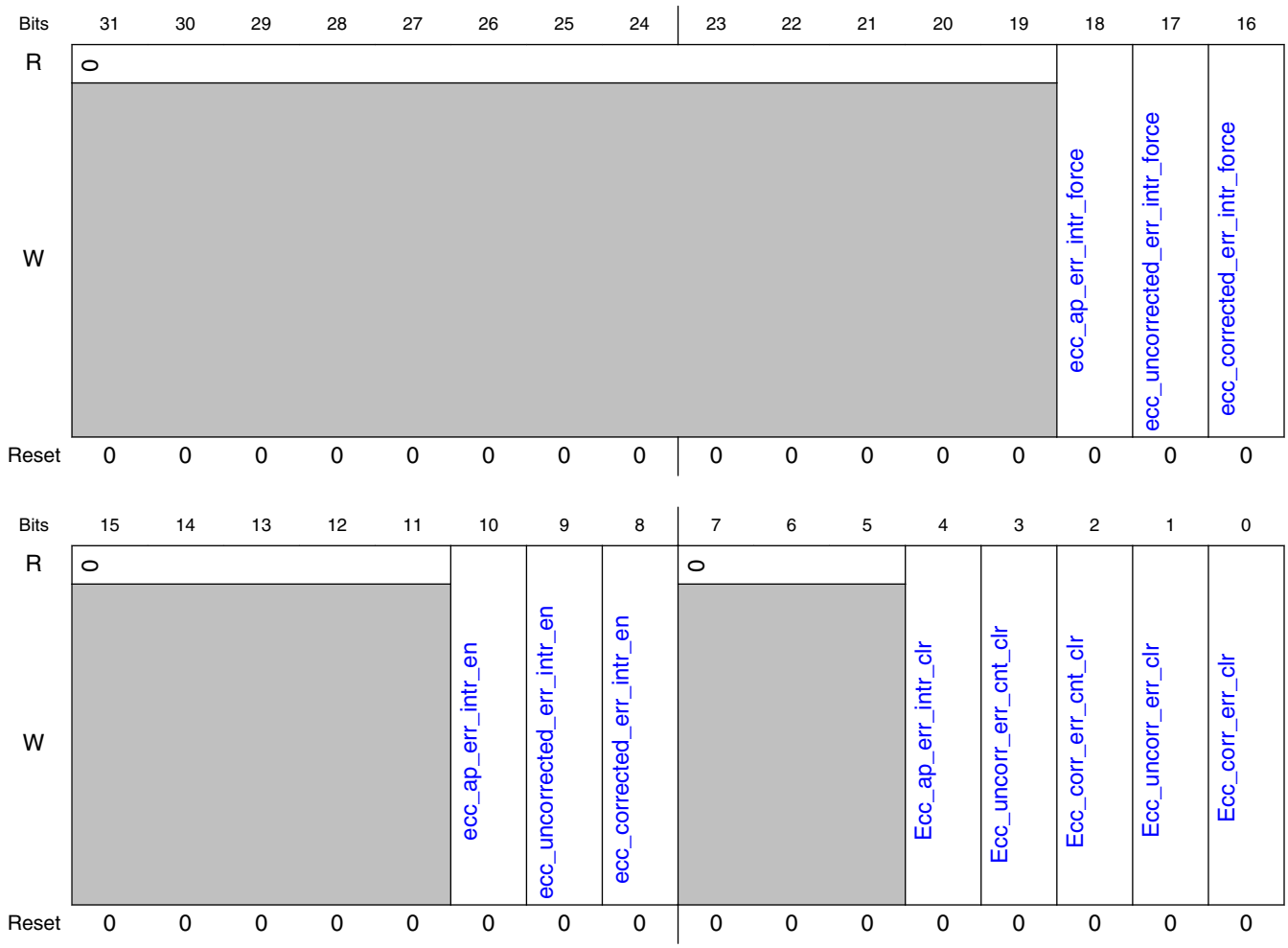
Field	Function
31-17 —	Reserved
16 ecc_uncorrected_err	ECC uncorrected err Double bit error indicator. When 1, indicates a double bit error was detected.
15-9 —	Reserved
8 ecc_corrected_err	ECC corrected err Single bit error indication. When 1, indicates a single bit error was detected and corrected.
7 —	Reserved
6-0 ecc_corrected_bit_num	ECC corrected bit num Indicates the bit number corrected by single-bit ECC error. This field is 7 bits wide in order to handle 72 bits present in a single lane.

9.2.5.1.22 ECC clear register (ECCCTL)

9.2.5.1.22.1 Offset

Register	Offset
ECCCTL	7Ch

9.2.5.1.22.2 Diagram



9.2.5.1.22.3 Fields

Field	Function
31-19	Reserved

Table continues on the next page...

Field	Function
—	
18 ecc_ap_err_intr_force	ECC AP ERR INT FORCE Interrupt force bit for ecc_ap_err_intr. Setting this register causes the output interrupt to assert. This bit is automatically cleared by the controller. There is no interaction between functionally triggering an interrupt and forcing an interrupt. Both are mutually exclusive.
17 ecc_uncorrected_err_intr_force	ECC UNCORR ERR INT FORCE Interrupt force bit for ecc_uncorrected_err_intr. Setting this register causes the output interrupt to assert. This bit is automatically cleared by the controller. There is no interaction between functionally triggering an interrupt and forcing an interrupt. Both are mutually exclusive.
16 ecc_corrected_err_intr_force	ECC CORR ERR INT FORCE Interrupt force bit for ecc_corrected_err_intr. Setting this register causes the output interrupt to assert. This bit is automatically cleared by the controller. There is no interaction between functionally triggering an interrupt and forcing an interrupt. Both are mutually exclusive.
15-11 —	Reserved
10 ecc_ap_err_intr_en	ECC AP ERR INT EN Interrupt enable bit for ecc_ap_err_intr. 0: interrupt is disabled 1: interrupt is enabled
9 ecc_uncorrected_err_intr_en	ECC UNCORR ERR INT EN Interrupt enable bit for ecc_uncorrected_err_intr. 0: interrupt is disabled 1: interrupt is enabled
8 ecc_corrected_err_intr_en	ECC CORR ERR INT EN Interrupt enable bit for ecc_corrected_err_intr. 0: interrupt is disabled 1: interrupt is enabled
7-5 —	Reserved
4 Ecc_ap_err_intr_clr	ECC AP ERR INT CLR Interrupt clear bit for ecc_ap_err_intr. If this bit is set, the ECCAPSTAT.ecc_ap_err/ecc_ap_err_intr is cleared. The controller automatically clears this bit.
3 Ecc_uncorr_err_cnt_clr	ECC UNCORR ERR CNT CLR Setting this register to 1 clears the currently stored uncorrected error count. The ECCERRCNT.ecc_uncorr_err_cnt register is cleared by this operation. The controller automatically clears this bit.
2 Ecc_corr_err_cnt_clr	ECC CORR ERR CNT CLR Setting this register to 1 clears the currently stored corrected error count. The ECCERRCNT.ecc_corr_err_cnt register is cleared by this operation. The controller automatically clears this bit.
1 Ecc_uncorr_err_clr	ECC UNCORR ERR CLR Setting this register bit to 1 clears the currently stored uncorrected ECC error. The following registers are cleared ECCSTAT.ecc_uncorrected_err. The controller automatically clears this bit.

Table continues on the next page...

Field	Function
0 Ecc_corr_err_clr	ECC CORR ERR CLR Setting this register bit to 1 clears the currently stored corrected ECC error. The following registers are cleared ECCSTAT.ecc_corrected_err. The controller automatically clears this bit.

9.2.5.1.23 ECC error counter register (ECCERRCNT)

9.2.5.1.23.1 Offset

Register	Offset
ECCERRCNT	80h

9.2.5.1.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Ecc_uncorr_err_cnt															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Ecc_corr_err_cnt															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.23.3 Fields

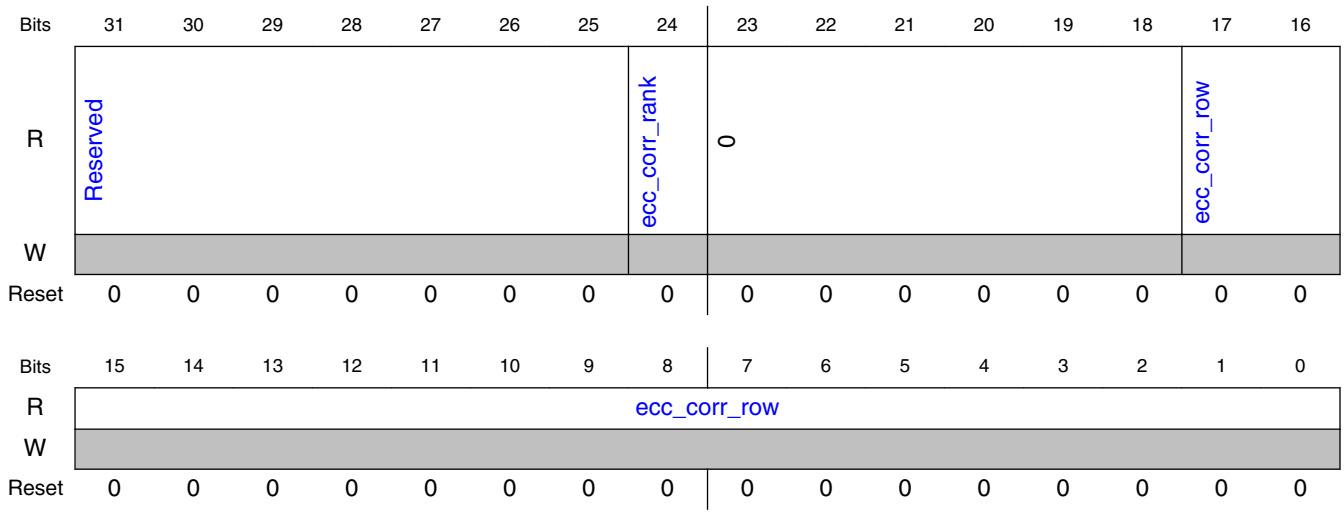
Field	Function
31-16 Ecc_uncorr_err_cnt	ECC uncorr err cnt Indicates the number of uncorrectable errors detected.
15-0 Ecc_corr_err_cnt	ECC corr err cnt Indicates the number of correctable errors detected.

9.2.5.1.24 ECC corrected error address register 0 (ECCCADDR0)

9.2.5.1.24.1 Offset

Register	Offset
ECCCADDR0	84h

9.2.5.1.24.2 Diagram



9.2.5.1.24.3 Fields

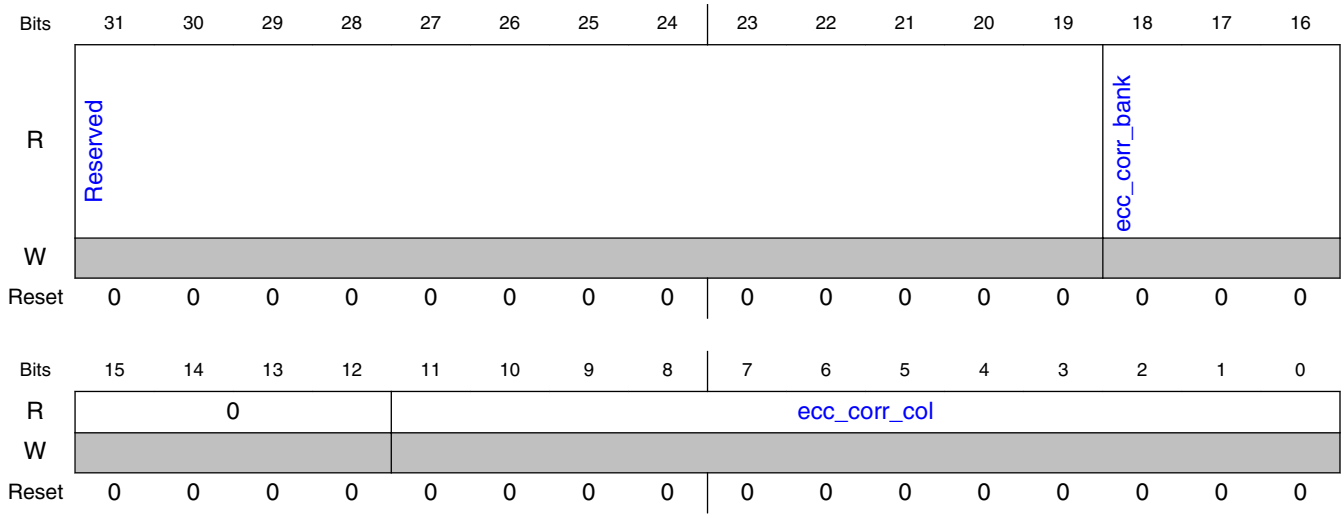
Field	Function
31-25 —	Reserved
24 ecc_corr_rank	ECC corr rank Indicates the rank number of read resulting in a corrected ECC error.
23-18 —	Reserved
17-0 ecc_corr_row	ECC corr row Indicates the row number of read resulting in a corrected ECC error.

9.2.5.1.25 ECC corrected error address register 1 (ECCCADDR1)

9.2.5.1.25.1 Offset

Register	Offset
ECCCADDR1	88h

9.2.5.1.25.2 Diagram



9.2.5.1.25.3 Fields

Field	Function
31-19 —	Reserved
18-16 ecc_corr_bank	ECC corr bank Indicates the bank number of a read resulting in a corrected ECC error.
15-12 —	Reserved
11-0 ecc_corr_col	ECC corr col Indicates the column number of a read resulting in a corrected ECC error.

9.2.5.1.26 ECC corrected syndrome register 0 (ECCCSYN0)

9.2.5.1.26.1 Offset

Register	Offset
ECCCSYN0	8Ch

9.2.5.1.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ecc_corr_syndromes_31_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ecc_corr_syndromes_31_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.26.3 Fields

Field	Function
31-0 ecc_corr_syndromes_31_0	ECC corr synd Indicates the data pattern that resulted in a corrected error.

9.2.5.1.27 ECC corrected syndrome register 1 (ECCCSYN1)

9.2.5.1.27.1 Offset

Register	Offset
ECCCSYN1	90h

9.2.5.1.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ecc_corr_syndromes_63_32															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ecc_corr_syndromes_63_32															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.27.3 Fields

Field	Function
31-0 ecc_corr_syndromes_63_32	ECC corr synd Indicates the data pattern that resulted in a corrected error.

9.2.5.1.28 ECC corrected syndrome register 2 (ECCCSYN2)

9.2.5.1.28.1 Offset

Register	Offset
ECCCSYN2	94h

9.2.5.1.28.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								ecc_corr_syndromes_71_64							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.28.3 Fields

Field	Function
31-8 —	Reserved
7-0 ecc_corr_syndromes_71_64	ECC corr synd Indicates the data pattern that resulted in a corrected error.

9.2.5.1.29 ECC corrected data bit mask register 0 (ECCBITMASK0)

9.2.5.1.29.1 Offset

Register	Offset
ECCBITMASK0	98h

9.2.5.1.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ecc_corr_bit_mask_31_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ecc_corr_bit_mask_31_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.29.3 Fields

Field	Function
31-0 ecc_corr_bit_mask_31_0	ECC corr bit mask Indicates the mask for the corrected data portion. 1 on any bit indicates that the bit has been corrected by the ECC logic. 0 on any bit indicates that the bit has not been corrected by the ECC logic. This register accumulates data over multiple ECC errors to give an overall indication of which bits are being fixed. It is cleared by writing a 1 to ECCCTL.ecc_corrected_err_clr.

9.2.5.1.30 ECC corrected data bit mask register 1 (ECCBITMASK1)

9.2.5.1.30.1 Offset

Register	Offset
ECCBITMASK1	9Ch

9.2.5.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.30.3 Fields

Field	Function
31-0 ecc_corr_bit_mask_63_32	ECC corr bit mask Indicates the mask for the corrected data portion. 1 on any bit indicates that the bit has been corrected by the ECC logic. 0 on any bit indicates that the bit has not been corrected by the ECC logic. This register accumulates data over multiple ECC errors to give an overall indication of which bits are being fixed. It is cleared by writing a 1 to ECCCTL.ecc_corrected_err_clr.

9.2.5.1.31 ECC corrected data bit mask register 2 (ECCBITMASK2)

9.2.5.1.31.1 Offset

Register	Offset
ECCBITMASK2	A0h

9.2.5.1.31.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ecc_corr_bit_mask_71_64							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.31.3 Fields

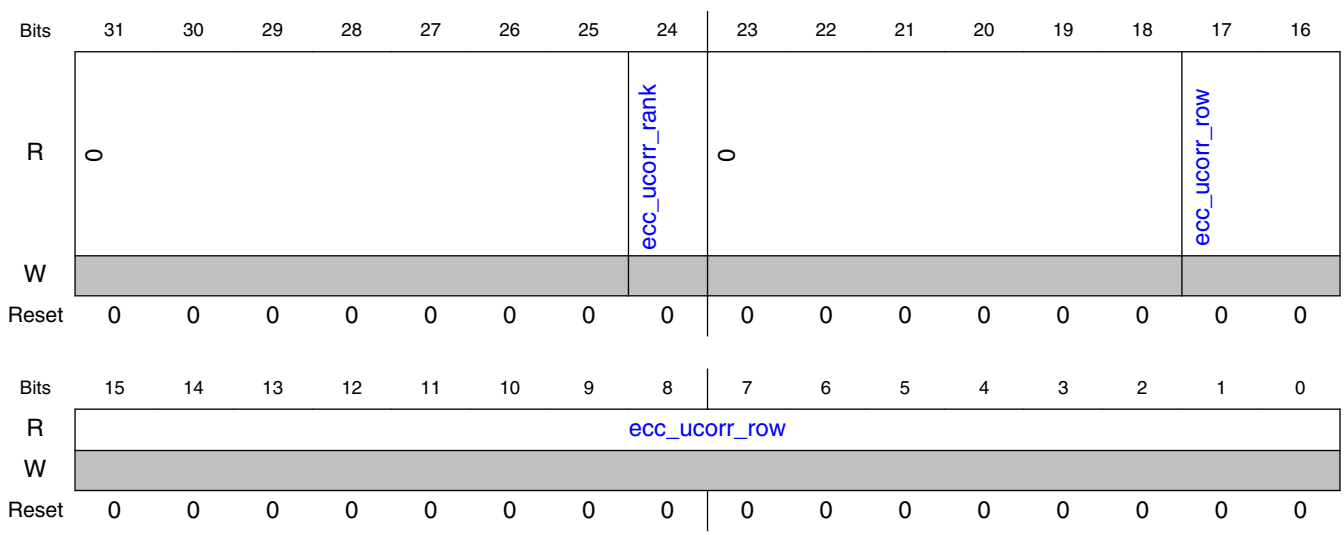
Field	Function
31-8 —	Reserved
7-0 ecc_corr_bit_mask_71_64	ECC corr bit mask Indicates the mask for the corrected data portion. 1 on any bit indicates that the bit has been corrected by the ECC logic. 0 on any bit indicates that the bit has not been corrected by the ECC logic. This register accumulates data over multiple ECC errors to give an overall indication of which bits are being fixed. It is cleared by writing a 1 to ECCCTL.ecc_corrected_err_clr.

9.2.5.1.32 ECC uncorrected error address register 0 (ECCUADDR0)

9.2.5.1.32.1 Offset

Register	Offset
ECCUADDR0	A4h

9.2.5.1.32.2 Diagram



9.2.5.1.32.3 Fields

Field	Function
31-25	Reserved

Table continues on the next page...

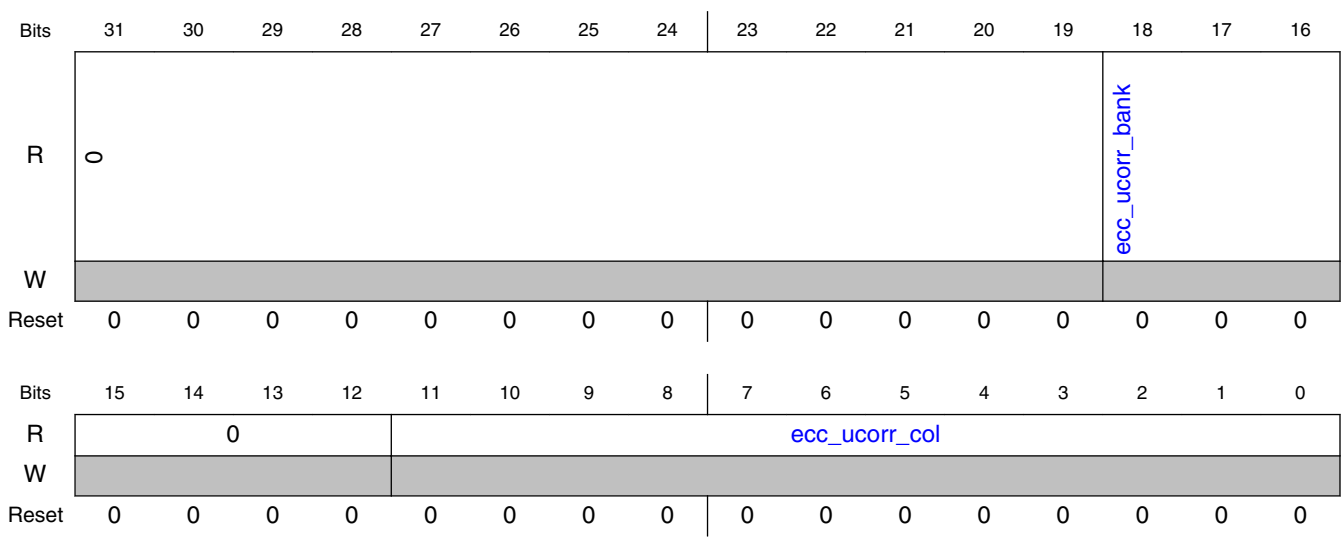
Field	Function
—	
24 ecc_ucorr_rank	ECC uncorr rank Indicates the rank number of read resulting in an uncorrected ecc error.
23-18 —	Reserved
17-0 ecc_ucorr_row	ECC uncorr row Indicates the row number of read resulting in an uncorrected ecc error.

9.2.5.1.33 ECC uncorrected error address register 1 (ECCUADDR1)

9.2.5.1.33.1 Offset

Register	Offset
ECCUADDR1	A8h

9.2.5.1.33.2 Diagram



9.2.5.1.33.3 Fields

Field	Function
31-19 —	Reserved

Table continues on the next page...

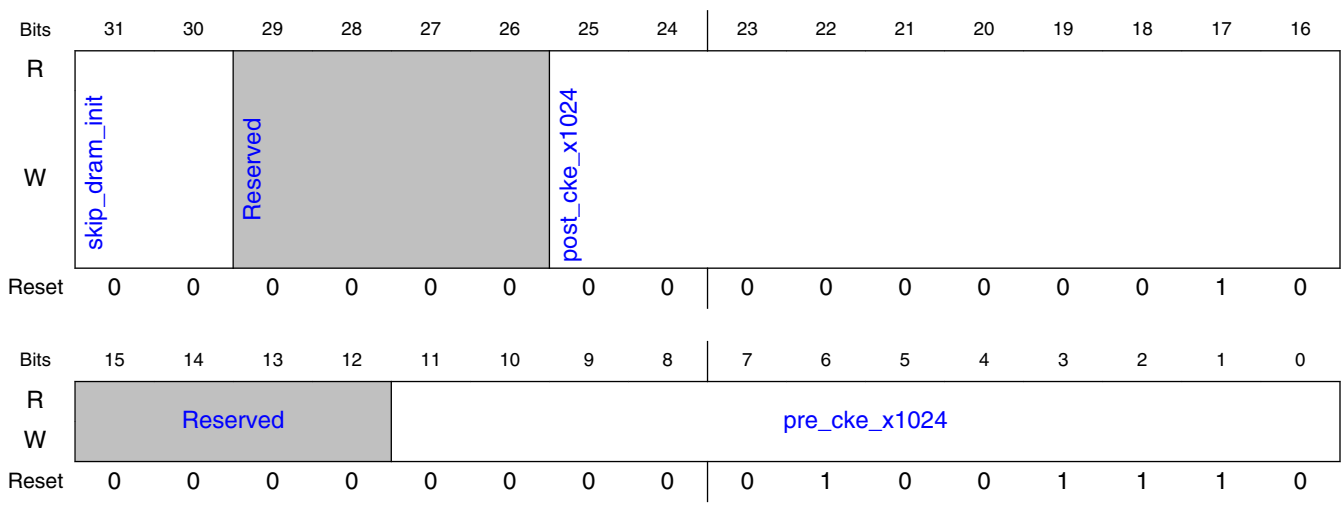
Field	Function
18-16 ecc_ucorr_bank	ECC uncorr bank Indicates the bank number of a read resulting in an uncorrected ECC error.
15-12 —	Reserved
11-0 ecc_ucorr_col	ECC uncorr col Indicates the column number of a read resulting in an uncorrected ECC error.

9.2.5.1.34 SDRAM Initialization Register 0 (INIT0)

9.2.5.1.34.1 Offset

Register	Offset
INIT0	D0h

9.2.5.1.34.2 Diagram



9.2.5.1.34.3 Fields

Field	Function
31-30 skip_dram_init	If lower bit is enabled the SDRAM initialization routine is skipped. The upper bit decides what state the controller starts up in when reset is removed - 00 - SDRAM Intialization routine is run after power-up - 01 - SDRAM Initialization routine is skipped after power-up. Controller starts up in Normal Mode - 11 - SDRAM Initialization routine is skipped after power-up. Controller starts up in Self-refresh Mode - 10 - SDRAM Initialization routine is run after power-up. 00b - SDRAM Initialization routine is run after power-up

Table continues on the next page...

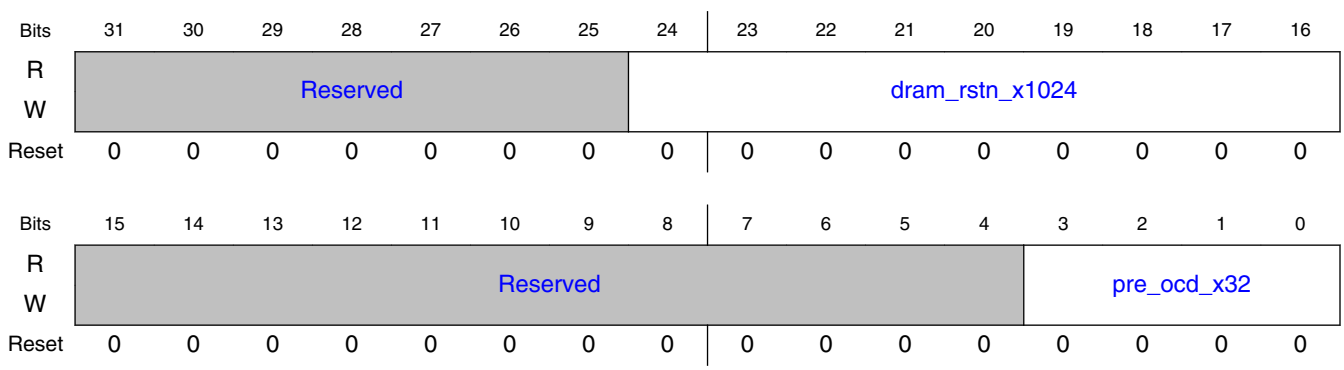
Field	Function
	01b - SDRAM Initialization routine is skipped after power-up 10b - SDRAM Initialization routine is run after power-up 11b - SDRAM Initialization routine is skipped after power-up
29-26 —	Reserved
25-16 post_cke_x1024	Cycles to wait after driving CKE high to start the SDRAM initialization sequence. Unit: 1024 DFI clock cycles. LPDDR4 typically requires this to be programmed for a delay of 2 us. When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value.
15-12 —	Reserved
11-0 pre_cke_x1024	Cycles to wait after reset before driving CKE high to start the SDRAM initialization sequence. Unit: 1024 DFI clock cycles. LPDDR4: tINIT3 of 2 ms (min). When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value.

9.2.5.1.35 SDRAM Initialization Register 1 (INIT1)

9.2.5.1.35.1 Offset

Register	Offset
INIT1	D4h

9.2.5.1.35.2 Diagram



9.2.5.1.35.3 Fields

Field	Function
31-25	Reserved

Table continues on the next page...

Field	Function
—	
24-16 dram_rstn_x1024	Number of cycles to assert SDRAM reset signal during init sequence. For use with a DDR PHY, this should be set to a minimum of 1. When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value. Unit: 1024 DFI clock cycles.
15-4 —	Reserved
3-0 pre_ocd_x32	Wait period before driving the OCD complete command to SDRAM. Unit: Counts of a global timer that pulses every 32 DFI clock cycles. There is no known specific requirement for this; it may be set to zero.

9.2.5.1.36 SDRAM Initialization Register 3 (INIT3)

9.2.5.1.36.1 Offset

Register	Offset
INIT3	DCh

9.2.5.1.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	emr															
Reset	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0

9.2.5.1.36.3 Fields

Field	Function
31-16 mr	DDR4: Value loaded into MR0 register. LPDDR4 - Value to write to MR1 register
15-0 emr	DDR4: Value to write to MR1 register Set bit 7 to 0. If PHY-evaluation mode training is enabled, this bit is set appropriately by the DDRC during write leveling. LPDDR4 - Value to write to MR2 register

9.2.5.1.37 SDRAM Initialization Register 4 (INIT4)

9.2.5.1.37.1 Offset

Register	Offset
INIT4	E0h

9.2.5.1.37.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	emr2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	emr3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.37.3 Fields

Field	Function
31-16 emr2	DDR4: Value to write to MR2 register. LPDDR4: Value to write to MR3 register.
15-0 emr3	DDR4: Value to write to MR3 register. LPDDR4: Value to write to MR13 register

9.2.5.1.38 SDRAM Initialization Register 5 (INIT5)

9.2.5.1.38.1 Offset

Register	Offset
INIT5	E4h

9.2.5.1.38.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								dev_zqinit_x32							
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

9.2.5.1.38.3 Fields

Field	Function
31-24 —	Reserved
23-16 dev_zqinit_x32	ZQ initial calibration, tZQINIT. Present only for DDR4. DDR4 requires 1024 SDRAM clock cycles. When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value. Unit: 32 DFI clock cycles.
15-0 —	Reserved

9.2.5.1.39 SDRAM Initialization Register 6 (INIT6)

9.2.5.1.39.1 Offset

Register	Offset
INIT6	E8h

9.2.5.1.39.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	mr5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.39.3 Fields

Field	Function
31-16 mr4	DDR4- Value to be loaded into SDRAM MR4 registers. Used in DDR4 designs only.
15-0 mr5	DDR4- Value to be loaded into SDRAM MR5 registers. Used in DDR4 designs only.

9.2.5.1.40 SDRAM Initialization Register 7 (INIT7)

9.2.5.1.40.1 Offset

Register	Offset
INIT7	ECh

9.2.5.1.40.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr6															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.40.3 Fields

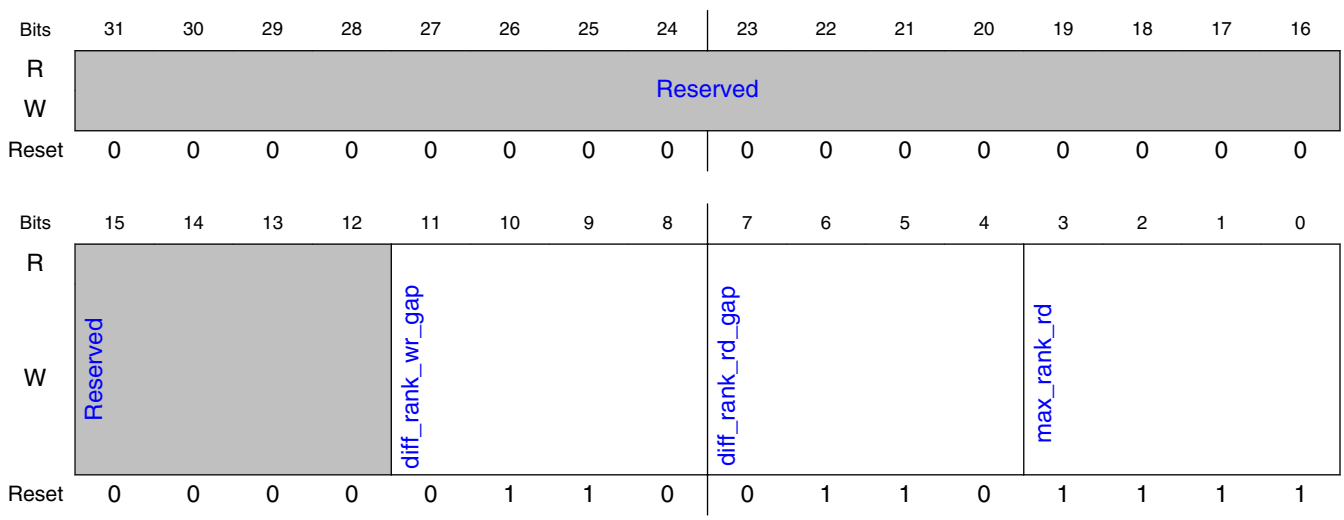
Field	Function
31-16 mr6	DDR4- Value to be loaded into SDRAM MR6 registers. Used in DDR4 designs only.
15-0 —	Reserved

9.2.5.1.41 Rank Control Register (RANKCTL)

9.2.5.1.41.1 Offset

Register	Offset
RANKCTL	F4h

9.2.5.1.41.2 Diagram



9.2.5.1.41.3 Fields

Field	Function
31-12 —	Reserved

Table continues on the next page...

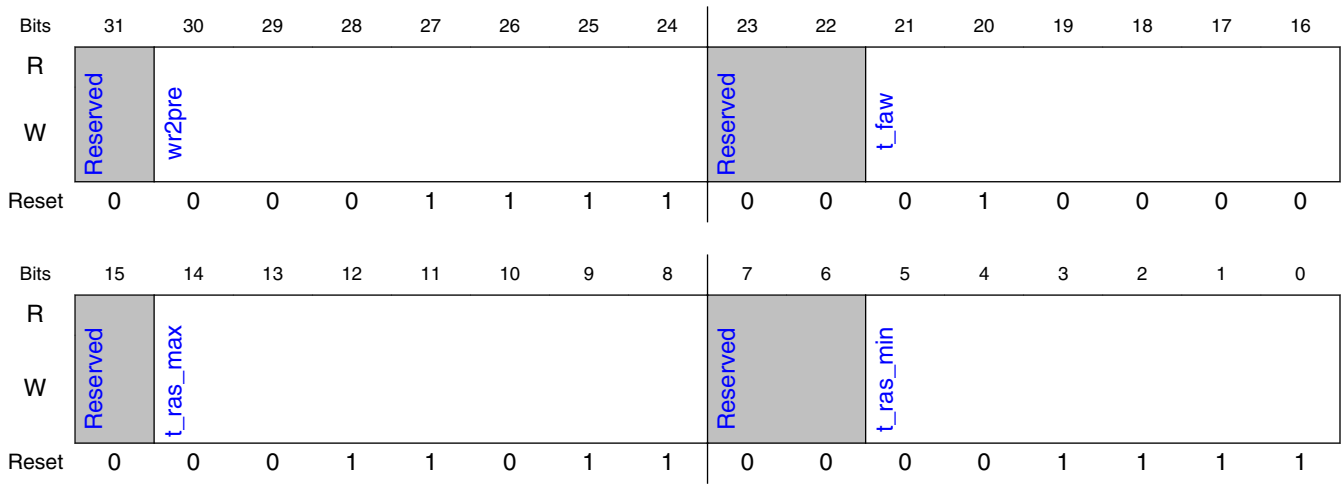
Field	Function
11-8 diff_rank_wr_gap	Only present for multi-rank configurations. Indicates the number of clocks of gap in data responses when performing consecutive writes to different ranks. This is used to switch the delays in the PHY to match the rank requirements. This value should consider both PHY requirement and ODT requirement. - PHY requirement: tphy_wrcsgap + 1 (see PHY databook for value of tphy_wrcsgap) If CRC feature is enabled, should be increased by 1. If write preamble is set to 2tCK(DDR4/LPDDR4 only), should be increased by 1. If write postamble is set to 1.5tCK(LPDDR4 only), should be increased by 1. - ODT requirement: The value programmed in this register takes care of the ODT switch off timing requirement when switching ranks during writes. For LPDDR4, the requirement is ODTLoff - ODTLon - BL/2 + 1 When the controller is operating in 1:1 mode, program this to the larger of PHY requirement or ODT requirement. When the controller is operating in 1:2 mode, program this to the larger value divided by two and round it up to the next integer.
7-4 diff_rank_rd_gap	Only present for multi-rank configurations. Indicates the number of clocks of gap in data responses when performing consecutive reads to different ranks. This is used to switch the delays in the PHY to match the rank requirements. This value should consider both PHY requirement and ODT requirement. - PHY requirement: tphy_rdcsgap + 1 (see PHY databook for value of tphy_rdcsgap) If read preamble is set to 2tCK(DDR4/LPDDR4 only), should be increased by 1. If read postamble is set to 1.5tCK(LPDDR4 only), should be increased by 1. - ODT requirement: The value programmed in this register takes care of the ODT switch off timing requirement when switching ranks during reads. When the controller is operating in 1:1 mode, program this to the larger of PHY requirement or ODT requirement. When the controller is operating in 1:2 mode, program this to the larger value divided by two and round it up to the next integer.
3-0 max_rank_rd	Only present for multi-rank configurations. Background: Reads to the same rank can be performed back-to-back. Reads to different ranks require additional gap dictated by the register RANKCTL.diff_rank_rd_gap. This is to avoid possible data bus contention as well as to give PHY enough time to switch the delay when changing ranks. The DDRC arbitrates for bus access on a cycle-by-cycle basis; therefore after a read is scheduled, there are few clock cycles (determined by the value on RANKCTL.diff_rank_rd_gap register) in which only reads from the same rank are eligible to be scheduled. This prevents reads from other ranks from having fair access to the data bus. This parameter represents the maximum number of reads that can be scheduled consecutively to the same rank. After this number is reached, a delay equal to RANKCTL.diff_rank_rd_gap is inserted by the scheduler to allow all ranks a fair opportunity to be scheduled. Higher numbers increase bandwidth utilization, lower numbers increase fairness. This feature can be DISABLED by setting this register to 0. When set to 0, the Controller will stay on the same rank as long as commands are available for it. Minimum programmable value is 0 (feature disabled) and maximum programmable value is 0xF. FOR PERFORMANCE ONLY.

9.2.5.1.42 SDRAM Timing Register 0 (DRAMTMG0)

9.2.5.1.42.1 Offset

Register	Offset
DRAMTMG0	100h

9.2.5.1.42.2 Diagram



9.2.5.1.42.3 Fields

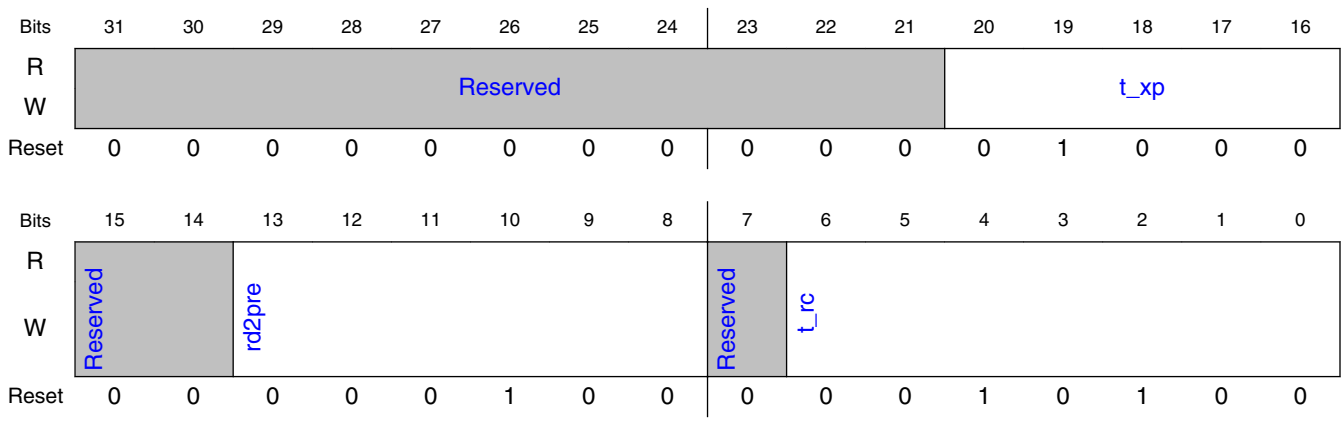
Field	Function
31 —	Reserved
30-24 wr2pre	Minimum time between write and precharge to same bank. Unit: Clocks Specifications: WL + BL/2 + tWR = approximately 8 cycles + 15 ns = 14 clocks @400MHz and less for lower frequencies where: - WL = write latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM. BST (burst terminate) is not supported at present. - tWR = Write recovery time. This comes directly from the SDRAM specification. Add one extra cycle for LPDDR4 for this parameter. When the controller is operating in 1:2 frequency ratio mode, 1T mode, divide the above value by 2. No rounding up. When the controller is operating in 1:2 frequency ratio mode, 2T mode or LPDDR4 mode, divide the above value by 2 and round it up to the next integer value.
23-22 —	Reserved
21-16 t_faw	tFAW Valid only when 8 or more banks(or banks x bank groups) are present. In 8-bank design, at most 4 banks must be activated in a rolling window of tFAW cycles. When the controller is operating in 1:2 frequency ratio mode, program this to (tFAW/2) and round up to next integer value. In a 4-bank design, set this register to 0x1 independent of the 1:1/1:2 frequency mode. Unit: Clocks
15 —	Reserved
14-8 t_ras_max	tRAS(max): Maximum time between activate and precharge to same bank. This is the maximum time that a page can be kept open Minimum value of this register is 1. Zero is invalid. When the controller is operating in 1:2 frequency ratio mode, program this to (tRAS(max)-1)/2. No rounding up. Unit: Multiples of 1024 clocks.
7-6 —	Reserved
5-0 t_ras_min	tRAS(min): Minimum time between activate and precharge to the same bank. When the controller is operating in 1:2 frequency mode, 1T mode, program this to tRAS(min)/2. No rounding up. When the controller is operating in 1:2 frequency ratio mode, 2T mode or LPDDR4 mode, program this to (tRAS(min)/2) and round it up to the next integer value. Unit: Clocks

9.2.5.1.43 SDRAM Timing Register 1 (DRAMTMG1)

9.2.5.1.43.1 Offset

Register	Offset
DRAMTMG1	104h

9.2.5.1.43.2 Diagram



9.2.5.1.43.3 Fields

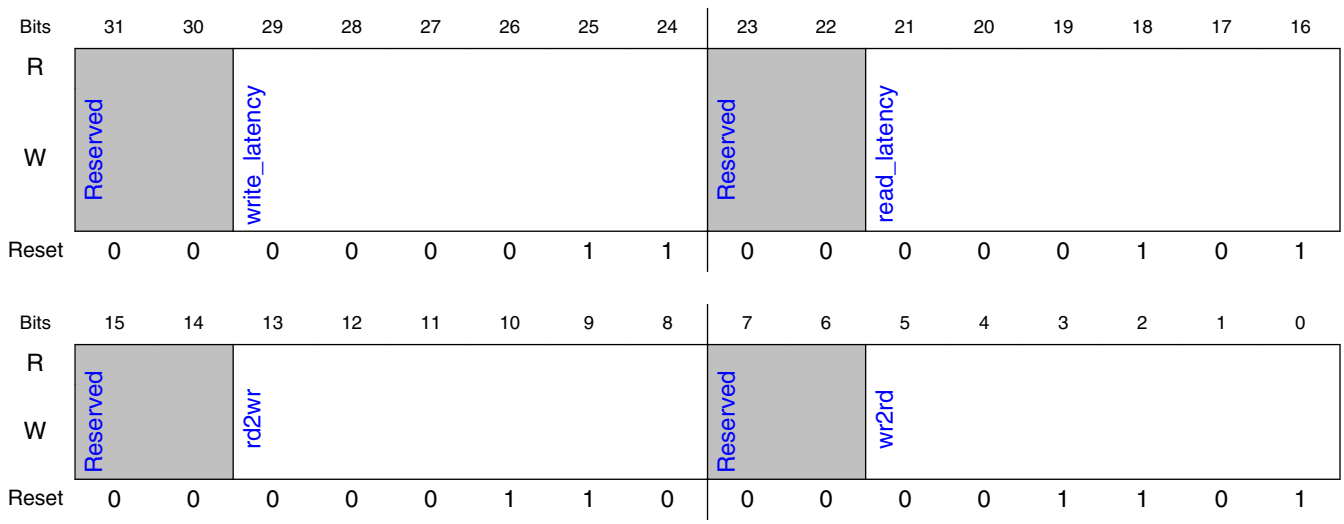
Field	Function
31-21 —	Reserved
20-16 t_xp	tXP: Minimum time after power-down exit to any operation. If C/A parity for DDR4 is used, set to (tXP +PL) instead. When the controller is operating in 1:2 frequency ratio mode, program this to (tXP/2) and round it up to the next integer value. Units: Clocks
15-14 —	Reserved
13-8 rd2pre	tRTP: Minimum time from read to precharge of same bank. DDR4: Max of following two equations: tAL + max (tRTP, 4) or, RL + BL/2 - tRP (*). LPDDR4: BL/2 + max(tRTP,8) - 8 (*). When both DDR4 SDRAM and ST-MRAM are used simultaneously, use SDRAM's tRP value for calculation. When the controller is operating in 1:2 mode, 1T mode, divide the above value by 2. No rounding up. When the controller is operating in 1:2 mode, 2T mode or LPDDR4 mode, divide the above value by 2 and round it up to the next integer value. Unit: Clocks.
7 —	Reserved
6-0 t_rc	tRC: Minimum time between activates to same bank. When the controller is operating in 1:2 frequency ratio mode, program this to (tRC/2) and round up to next integer value. Unit: Clocks.

9.2.5.1.44 SDRAM Timing Register 2 (DRAMTMG2)

9.2.5.1.44.1 Offset

Register	Offset
DRAMTMG2	108h

9.2.5.1.44.2 Diagram



9.2.5.1.44.3 Fields

Field	Function
31-30 —	Reserved
29-24 write_latency	Set to WL Time from write command to write data on SDRAM interface. This must be set to WL. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer. Unit: clocks
23-22 —	Reserved
21-16 read_latency	Set to RL Time from read command to read data on SDRAM interface. This must be set to RL.When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer. Unit: clocks
15-14 —	Reserved
13-8	DDR4: RL + BL/2 + 1 + WR_PREAMBLE - WL. LPDDR4(DQ ODT is Disabled): RL + BL/2 + RU(tDQSCKmax/tCK) + WR_PREAMBLE + RD_POSTAMBLE - WL. LPDDR4(DQ ODT is Enabled) : RL

Table continues on the next page...

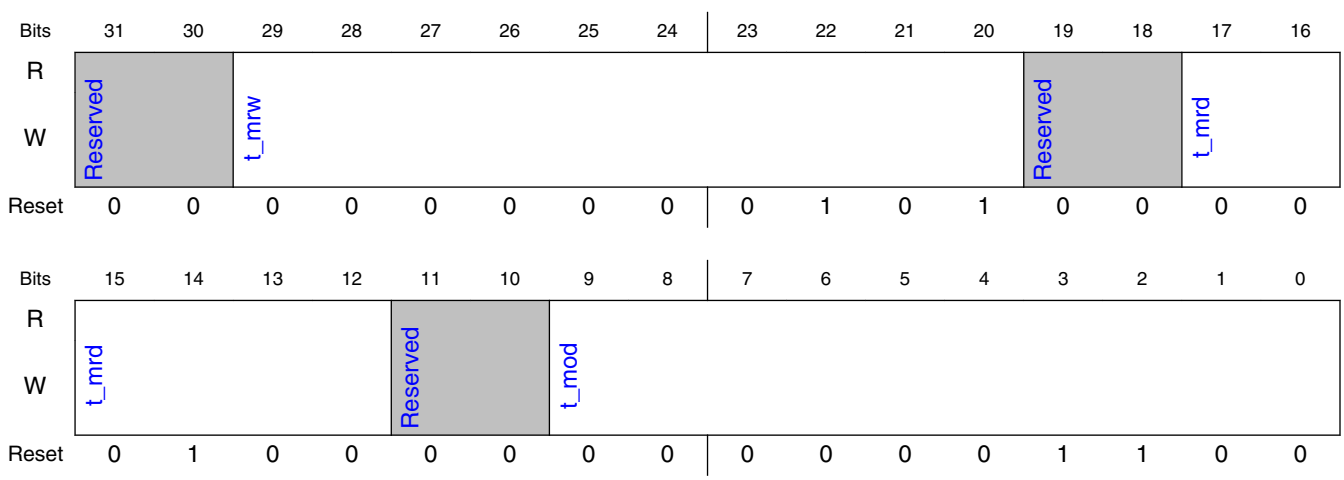
Field	Function
rd2wr	+ BL/2 + RU(tDQSCKmax/tCK) + RD_POSTAMBLE - ODTLon - RU(tODTon(min)/tCK). Minimum time from read command to write command. Include time for bus turnaround and all per-bank, per-rank, and global constraints. Please see the relevant PHY databook for details of what should be included here. Unit: Clocks. Where: - WL = write latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM - RL = read latency = CAS latency - WR_PREAMBLE = write preamble. This is unique to DDR4 and LPDDR4. - RD_POSTAMBLE = read postamble. This is unique to LPDDR4. For LPDDR4, if derating is enabled (DERATEEN.derate_enable=1), derated tDQSCKmax should be used. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer.
7-6 —	Reserved
5-0 wr2rd	DDR4: CWL + PL + BL/2 + tWTR_L Others: CWL + BL/2 + tWTR In DDR4, minimum time from write command to read command for same bank group. In others, minimum time from write command to read command. Includes time for bus turnaround, recovery times, and all per-bank, per-rank, and global constraints. Unit: Clocks. Where: - CWL = CAS write latency - PL = Parity latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM - tWTR_L = internal write to read command delay for same bank group. This comes directly from the SDRAM specification. - tWTR = internal write to read command delay. This comes directly from the SDRAM specification. Add one extra cycle for LPDDR4 operation. When the controller is operating in 1:2 mode, divide the value calculated using the above equation by 2, and round it up to next integer.

9.2.5.1.45 SDRAM Timing Register 3 (DRAMTMG3)

9.2.5.1.45.1 Offset

Register	Offset
DRAMTMG3	10Ch

9.2.5.1.45.2 Diagram



9.2.5.1.45.3 Fields

Field	Function
31-30 —	Reserved
29-20 t_mrw	Time to wait after a mode register write or read (MRW or MRR). Present only for LPDDR4. Set this to the larger of tMRW and tMRWCKEL. When the controller is operating in 1:2 frequency ratio mode, program this to the above values divided by 2 and round it up to the next integer value.
19-18 —	Reserved
17-12 t_mrd	tMRD: Cycles to wait after a mode register write or read. Depending on the connected SDRAM, tMRD represents: DDR4: Time from MRS to MRS command. LPDDR4: Time from MRS to non-MRS command. When the controller is operating in 1:2 frequency ratio mode, program this to (tMRD/2) and round it up to the next integer value. If C/A parity for DDR4 is used, set to tMRD_PAR(tMOD+PL) instead.
11-10 —	Reserved
9-0 t_mod	tMOD: Parameter used only in DDR4. Cycles between load mode command and following non-load mode command. If C/A parity for DDR4 is used, set to tMOD_PAR(tMOD+PL) instead. Set to tMOD if controller is operating in 1:1 frequency ratio mode, or tMOD/2 (rounded up to next integer) if controller is operating in 1:2 frequency ratio mode.

9.2.5.1.46 SDRAM Timing Register 4 (DRAMTMG4)

9.2.5.1.46.1 Offset

Register	Offset
DRAMTMG4	110h

9.2.5.1.46.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			t_rcd					Reserved				t_ccd			
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				t_rrd				Reserved			t_rp				
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1

9.2.5.1.46.3 Fields

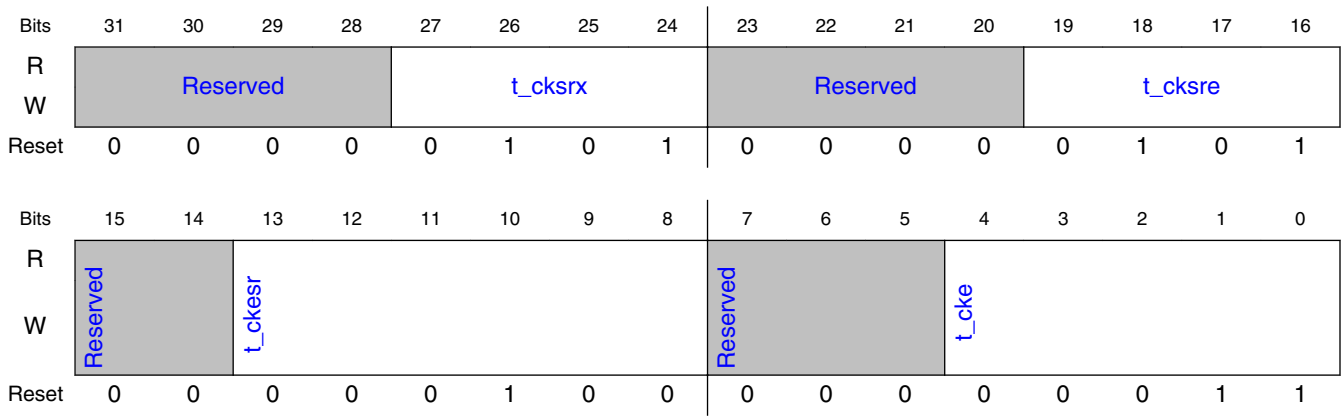
Field	Function
31-29 —	Reserved
28-24 t_rcd	tRCD - tAL: Minimum time from activate to read or write command to same bank. When the controller is operating in 1:2 frequency ratio mode, program this to ((tRCD - tAL)/2) and round it up to the next integer value. Minimum value allowed for this register is 1, which implies minimum (tRCD - tAL) value to be 2 when the controller is operating in 1:2 frequency ratio mode. Unit: Clocks.
23-20 —	Reserved
19-16 t_ccd	DDR4: tCCD_L: This is the minimum time between two reads or two writes for same bank group. Others: tCCD: This is the minimum time between two reads or two writes. When the controller is operating in 1:2 frequency ratio mode, program this to (tCCD_L/2 or tCCD/2) and round it up to the next integer value. Unit: clocks.
15-12 —	Reserved
11-8 t_rrd	DDR4: tRRD_L: Minimum time between activates from bank "a" to bank "b" for same bank group. Others: tRRD: Minimum time between activates from bank "a" to bank "b" When the controller is operating in 1:2 frequency ratio mode, program this to (tRRD_L/2 or tRRD/2) and round it up to the next integer value. Unit: Clocks.
7-5 —	Reserved
4-0 t_rp	tRP: Minimum time from precharge to activate of same bank. When the controller is operating in 1:1 frequency ratio mode, t_rp should be set to RoundUp(tRP/tCK). When the controller is operating in 1:2 frequency ratio mode, t_rp should be set to RoundDown(RoundUp(tRP/tCK)/2) + 1. When the controller is operating in 1:2 frequency ratio mode in LPDDR4, t_rp should be set to RoundUp(RoundUp(tRP/tCK)/2). Unit: Clocks.

9.2.5.1.47 SDRAM Timing Register 5 (DRAMTMG5)

9.2.5.1.47.1 Offset

Register	Offset
DRAMTMG5	114h

9.2.5.1.47.2 Diagram



9.2.5.1.47.3 Fields

Field	Function
31-28 —	Reserved
27-24 t_cksrx	This is the time before Self Refresh Exit that CK is maintained as a valid clock before issuing SRX. Specifies the clock stable time before SRX. Recommended settings: - LPDDR4: tCKCKEH. - DDR4: tCKSRX When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
23-20 —	Reserved
19-16 t_cksre	This is the time after Self Refresh Down Entry that CK is maintained as a valid clock. Specifies the clock disable delay after SRE. Recommended settings: LPDDR4: tCKCKEL. DDR4: max (10 ns, 5 tCK) (+ PL(parity latency)(*)) (*)Only if CRCPARCTL1.caparity_disable_before_sr=0, this register should be increased by PL. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
15-14 —	Reserved
13-8 t_ckesr	Minimum CKE low width for Self refresh or Self refresh power down entry to exit timing in memory clock cycles. Recommended settings: LPDDR4: max(tCKELPD, tSR). DDR4: tCKE + 1 (+ PL(parity latency)(*)) (*)Only if CRCPARCTL1.caparity_disable_before_sr=0, this register should be increased by PL. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
7-5 —	Reserved
4-0 t_cke	Minimum number of cycles of CKE HIGH/LOW during power-down and self refresh. LPDDR4 mode: Set this to the larger of tCKE, tCKELPD or tSR. - non-LPDDR4 designs: Set this to tCKE value. When the controller is operating in 1:2 frequency ratio mode, program this to (value described above)/2 and round it up to the next integer value. Unit: Clocks.

9.2.5.1.48 SDRAM Timing Register 6 (DRAMTMG6)

9.2.5.1.48.1 Offset

Register	Offset
DRAMTMG6	118h

9.2.5.1.48.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												t_ckcsx			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

9.2.5.1.48.3 Fields

Field	Function
31-4 —	Reserved
3-0 t_ckcsx	This is the time before Clock Stop Exit that CK is maintained as a valid clock before issuing Clock Stop Exit. Specifies the clock stable time before next command after Clock Stop Exit. Recommended settings: - LPDDR4: tXP + 2 When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for LPDDR4 devices.

9.2.5.1.49 SDRAM Timing Register 7 (DRAMTMG7)

9.2.5.1.49.1 Offset

Register	Offset
DRAMTMG7	11Ch

9.2.5.1.49.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				t_ckpde				Reserved				t_ckpdx			
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

9.2.5.1.49.3 Fields

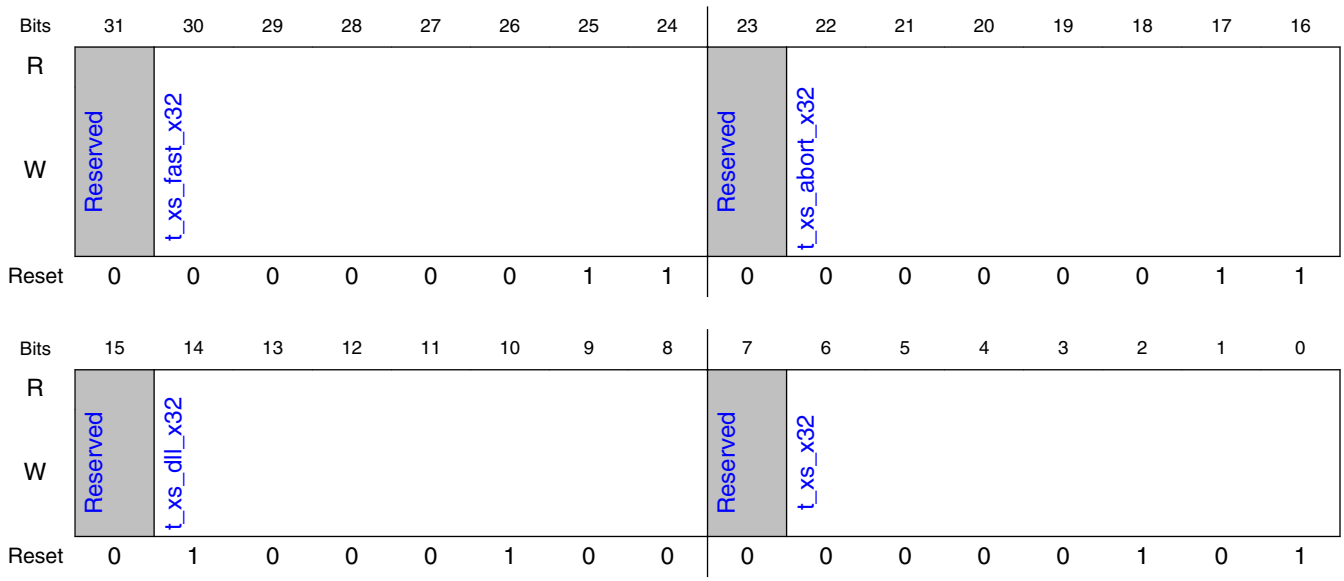
Field	Function
31-12 —	Reserved
11-8 t_ckpde	This is the time after Power Down Entry that CK is maintained as a valid clock. Specifies the clock disable delay after PDE. Recommended settings: - LPDDR4: tCKCKEL. When using DDR4 SDRAM, this register should be set to the same value as DRAMTMG5.t_cksre. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for LPDDR4 devices.
7-4 —	Reserved
3-0 t_ckpdx	This is the time before Power Down Exit that CK is maintained as a valid clock before issuing PDX. Specifies the clock stable time before PDX. Recommended settings: - LPDDR4: 2 . When using DDR4 SDRAM, this register should be set to the same value as DRAMTMG5.t_cksrx. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for LPDDR4 devices.

9.2.5.1.50 SDRAM Timing Register 8 (DRAMTMG8)

9.2.5.1.50.1 Offset

Register	Offset
DRAMTMG8	120h

9.2.5.1.50.2 Diagram



9.2.5.1.50.3 Fields

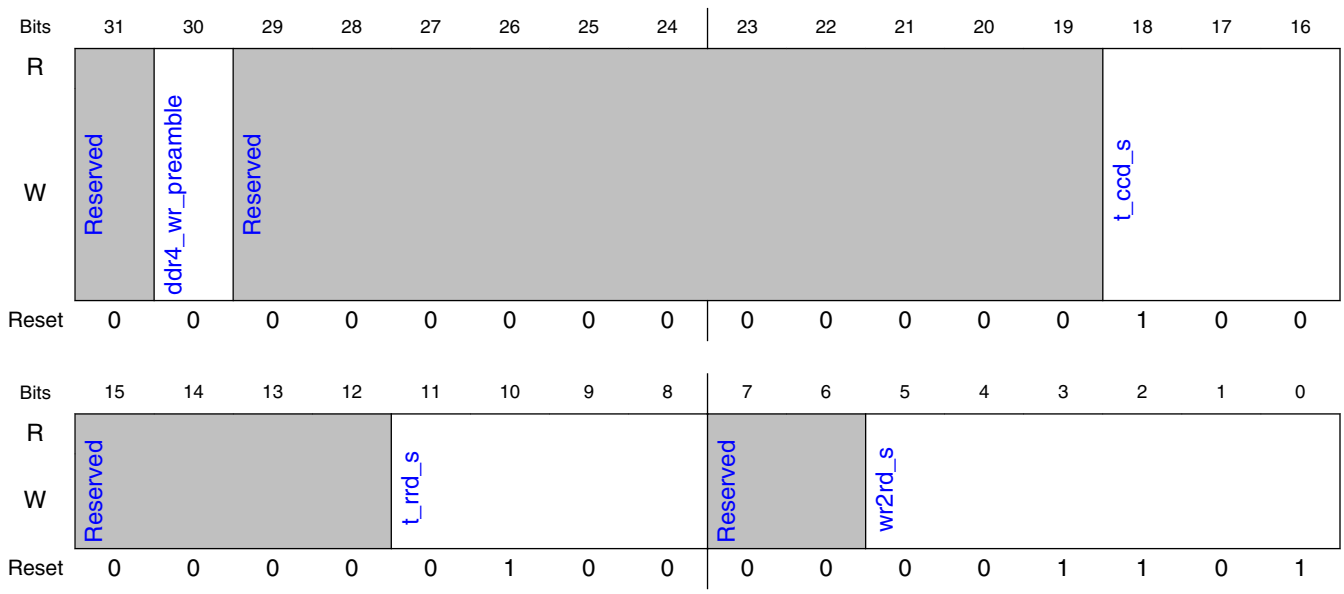
Field	Function
31 —	Reserved
30-24 t_xs_fast_x32	tXS_FAST: Exit Self Refresh to ZQCL, ZQCS and MRS (only CL, WR, RTP and Geardown mode). When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: This is applicable to only ZQCL/ZQCS commands. Note: Ensure this is less than or equal to t_xs_x32.
23 —	Reserved
22-16 t_xs_abort_x32	tXS_ABORT: Exit Self Refresh to commands not requiring a locked DLL in Self Refresh Abort. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: Ensure this is less than or equal to t_xs_x32.
15 —	Reserved
14-8 t_xs_dll_x32	tXSDLL: Exit Self Refresh to commands requiring a locked DLL. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: Used only for DDR4 SDRAM.
7 —	Reserved
6-0 t_xs_x32	tXS: Exit Self Refresh to commands not requiring a locked DLL. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: Used only for DDR4 SDRAM.

9.2.5.1.51 SDRAM Timing Register 9 (DRAMTMG9)

9.2.5.1.51.1 Offset

Register	Offset
DRAMTMG9	124h

9.2.5.1.51.2 Diagram



9.2.5.1.51.3 Fields

Field	Function
31 —	Reserved
30 ddr4_wr_preamble	DDR4 Write preamble mode - 0: 1tCK preamble - 1: 2tCK preamble Present only with MEMC_FREQ_RATIO=2
29-19 —	Reserved
18-16 t_ccd_s	tCCD_S: This is the minimum time between two reads or two writes for different bank group. For bank switching (from bank "a" to bank "b"), the minimum time is this value + 1. When the controller is operating in 1:2 frequency ratio mode, program this to (tCCD_S/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: clocks.
15-12 —	Reserved

Table continues on the next page...

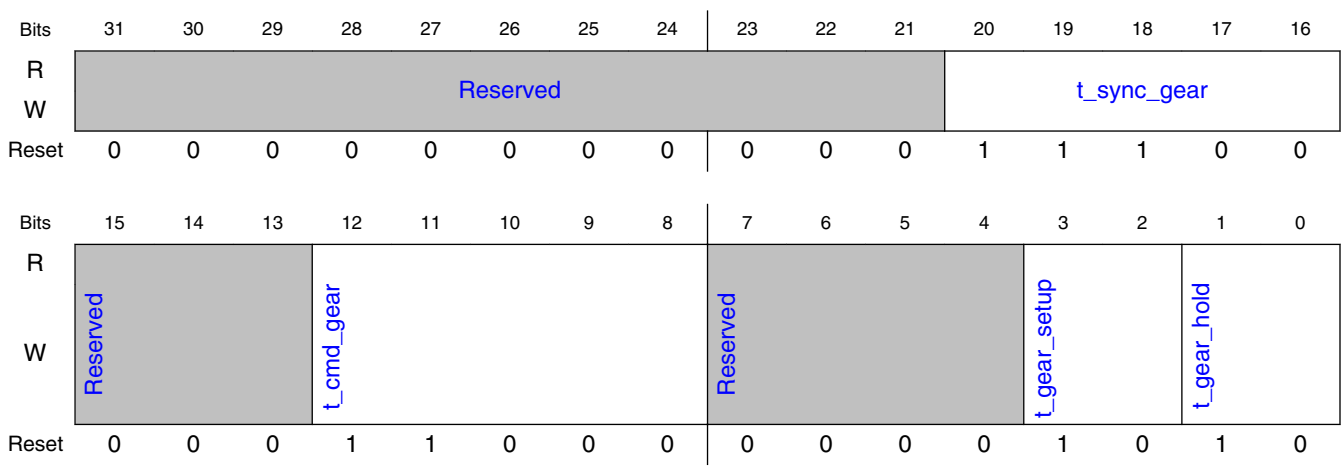
Field	Function
11-8 t_rrd_s	tRRD_S: Minimum time between activates from bank "a" to bank "b" for different bank group. When the controller is operating in 1:2 frequency ratio mode, program this to (tRRD_S/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: Clocks.
7-6 —	Reserved
5-0 wr2rd_s	CWL + PL + BL/2 + tWTR_S Minimum time from write command to read command for different bank group. Includes time for bus turnaround, recovery times, and all per-bank, per-rank, and global constraints. Present only in designs configured to support DDR4. Unit: Clocks. Where: - CWL = CAS write latency - PL = Parity latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM - tWTR_S = internal write to read command delay for different bank group. This comes directly from the SDRAM specification. When the controller is operating in 1:2 mode, divide the value calculated using the above equation by 2, and round it up to next integer.

9.2.5.1.52 SDRAM Timing Register 10 (DRAMTMG10)

9.2.5.1.52.1 Offset

Register	Offset
DRAMTMG10	128h

9.2.5.1.52.2 Diagram



9.2.5.1.52.3 Fields

Field	Function
31-21	Reserved

Table continues on the next page...

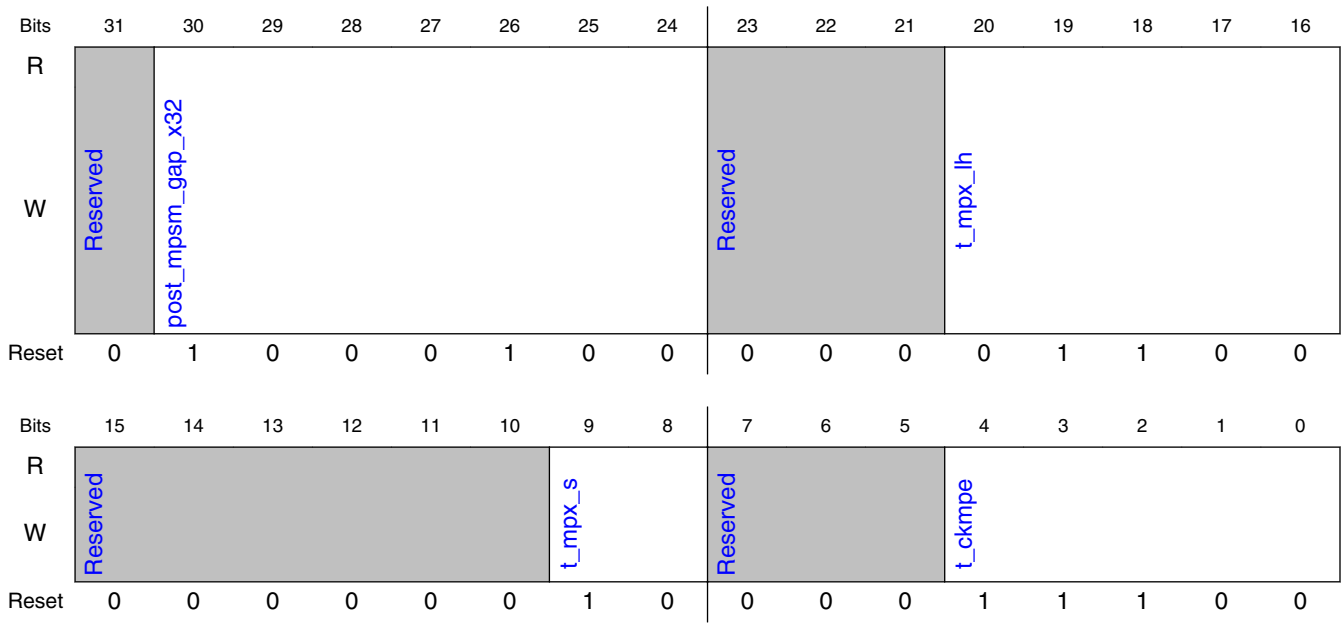
Field	Function
—	
20-16 t_sync_gear	Indicates the time between MRS command and the sync pulse time. This must be even number of clocks. For DDR4-2666 and DDR4-3200, this parameter is defined as tMOD(min)+4nCK tMOD(min) is greater of 24nCK or 15ns 15ns / .625ns = 24 Max value for this register is 24+4 = 28 When the controller is operating in 1:2 mode, program this to (tSYNC_GEAR/2) and round it up to the next integer value. Unit: Clocks
15-13 —	Reserved
12-8 t_cmd_gear	Sync pulse to first valid command. For DDR4-2666 and DDR4-3200, this parameter is defined as tMOD(min) tMOD(min) is greater of 24nCK or 15ns 15ns / .625ns = 24 Max value for this register is 24 When the controller is operating in 1:2 mode, program this to (tCMD_GEAR/2) and round it up to the next integer value. Unit: Clocks
7-4 —	Reserved
3-2 t_gear_setup	Geardown setup time. Minimum value of this register is 1. Zero is invalid. For DDR4-2666 and DDR4-3200, this parameter is defined as 2 clks When the controller is operating in 1:2 frequency ratio mode, program this to (tGEAR_setup/2) and round it up to the next integer value. Unit: Clocks
1-0 t_gear_hold	Geardown hold time. Minimum value of this register is 1. Zero is invalid. For DDR4-2666 and DDR4-3200, this parameter is defined as 2 clks When the controller is operating in 1:2 frequency ratio mode, program this to (tGEAR_hold/2) and round it up to the next integer value. Unit: Clocks

9.2.5.1.53 SDRAM Timing Register 11 (DRAMTMG11)

9.2.5.1.53.1 Offset

Register	Offset
DRAMTMG11	12Ch

9.2.5.1.53.2 Diagram



9.2.5.1.53.3 Fields

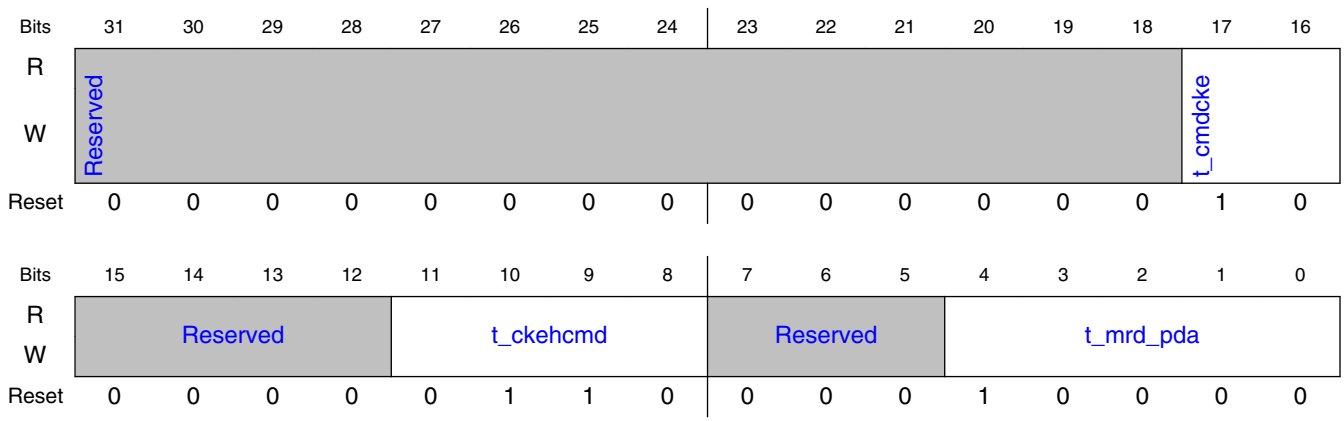
Field	Function
31 —	Reserved
30-24 post_mpsm_gap_x32	tXMPDLL: This is the minimum Exit MPSM to commands requiring a locked DLL. When the controller is operating in 1:2 frequency ratio mode, program this to (tXMPDLL/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: Multiples of 32 clocks.
23-21 —	Reserved
20-16 t_mpx_lh	tMPX_LH: This is the minimum CS_n Low hold time to CKE rising edge. When the controller is operating in 1:2 frequency ratio mode, program this to RoundUp(tMPX_LH/2)+1. Present only in designs configured to support DDR4. Unit: clocks.
15-10 —	Reserved
9-8 t_mpx_s	tMPX_S: Minimum time CS setup time to CKE. When the controller is operating in 1:2 frequency ratio mode, program this to (tMPX_S/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: Clocks.
7-5 —	Reserved
4-0 t_ckmpe	tCKMPE: Minimum valid clock requirement after MPSM entry. Present only in designs configured to support DDR4. Unit: Clocks. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer.

9.2.5.1.54 SDRAM Timing Register 12 (DRAMTMG12)

9.2.5.1.54.1 Offset

Register	Offset
DRAMTMG12	130h

9.2.5.1.54.2 Diagram



9.2.5.1.54.3 Fields

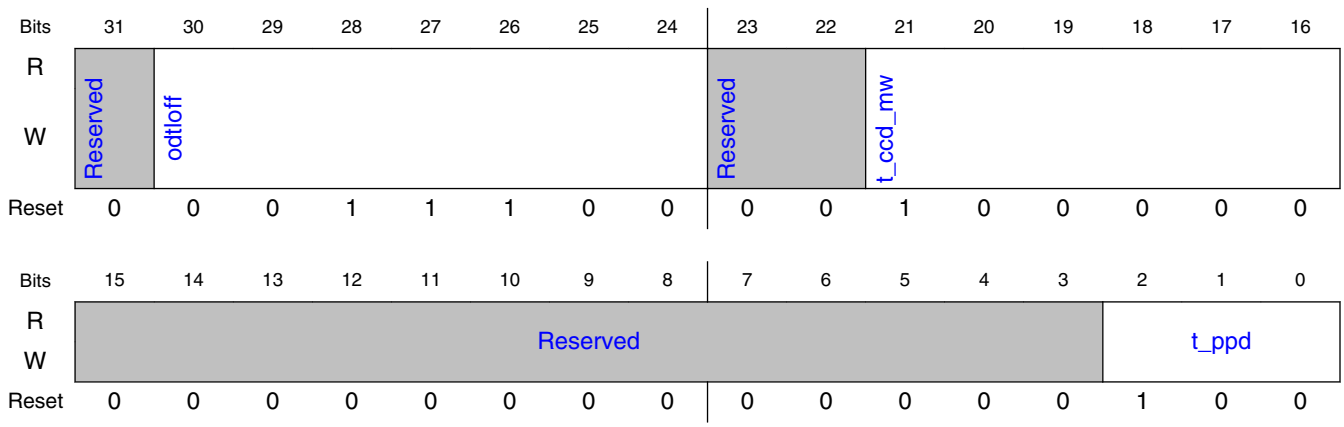
Field	Function
31-18 —	Reserved
17-16 t_cmdcke	tCMDCKE: Delay from valid command to CKE input LOW. Set this to the larger of tESCKE or tCMDCKE When the controller is operating in 1:2 frequency ratio mode, program this to (max(tESCKE, tCMDCKE)/2) and round it up to the next integer value.
15-12 —	Reserved
11-8 t_ckehcmd	tCKEHCMD: Valid command requirement after CKE input HIGH. When the controller is operating in 1:2 frequency ratio mode, program this to (tCKEHCMD/2) and round it up to the next integer value.
7-5 —	Reserved
4-0 t_mrd_pda	tMRD_PDA: This is the Mode Register Set command cycle time in PDA mode. When the controller is operating in 1:2 frequency ratio mode, program this to (tMRD_PDA/2) and round it up to the next integer value.

9.2.5.1.55 SDRAM Timing Register 13 (DRAMTMG13)

9.2.5.1.55.1 Offset

Register	Offset
DRAMTMG13	134h

9.2.5.1.55.2 Diagram



9.2.5.1.55.3 Fields

Field	Function
31 —	Reserved
30-24 odtloff	LPDDR4: tODTLoff: This is the latency from CAS-2 command to tODToff reference. When the controller is operating in 1:2 frequency ratio mode, program this to (tODTLoff/2) and round it up to the next integer value. Unit: Clocks.
23-22 —	Reserved
21-16 t_ccd_mw	LPDDR4: tCCDMW: This is the minimum time from write or masked write to masked write command for same bank. When the controller is operating in 1:2 frequency ratio mode, program this to (tCCDMW/2) and round it up to the next integer value. Unit: Clocks.
15-3 —	Reserved
2-0 t_ppd	LPDDR4: tPPD: This is the minimum time from precharge to precharge command. When the controller is operating in 1:2 frequency ratio mode, program this to (tPPD/2) and round it up to the next integer value. Unit: Clocks.

9.2.5.1.56 SDRAM Timing Register 14 (DRAMTMG14)

9.2.5.1.56.1 Offset

Register	Offset
DRAMTMG14	138h

9.2.5.1.56.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				t_xsr											
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

9.2.5.1.56.3 Fields

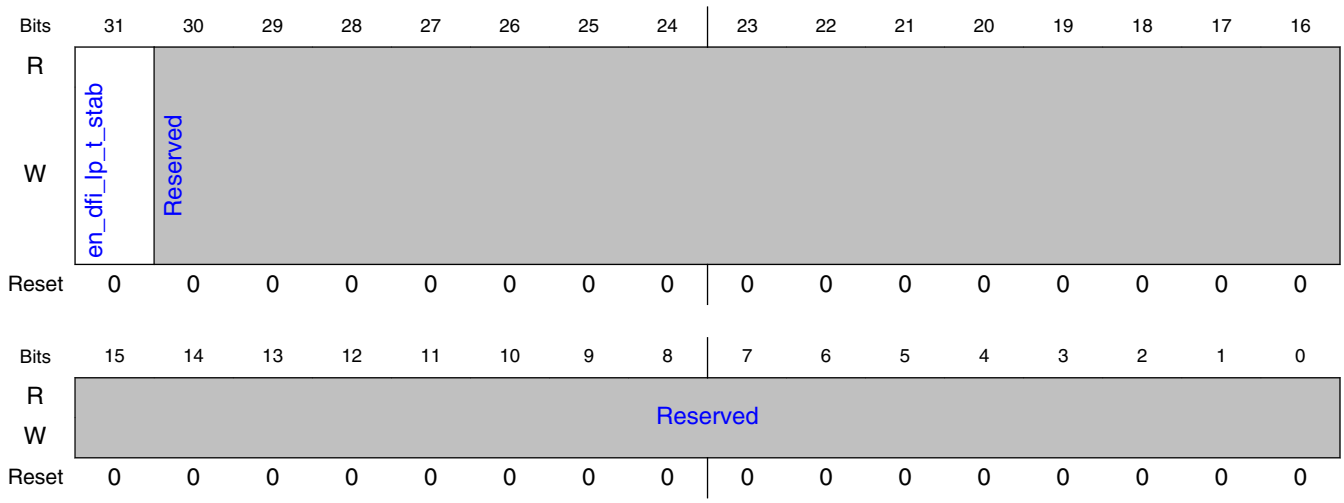
Field	Function
31-12 —	Reserved
11-0 t_xsr	tXSR: Exit Self Refresh to any command. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Note: Used only for LPDDR4 mode.

9.2.5.1.57 SDRAM Timing Register 15 (DRAMTMG15)

9.2.5.1.57.1 Offset

Register	Offset
DRAMTMG15	13Ch

9.2.5.1.57.2 Diagram



9.2.5.1.57.3 Fields

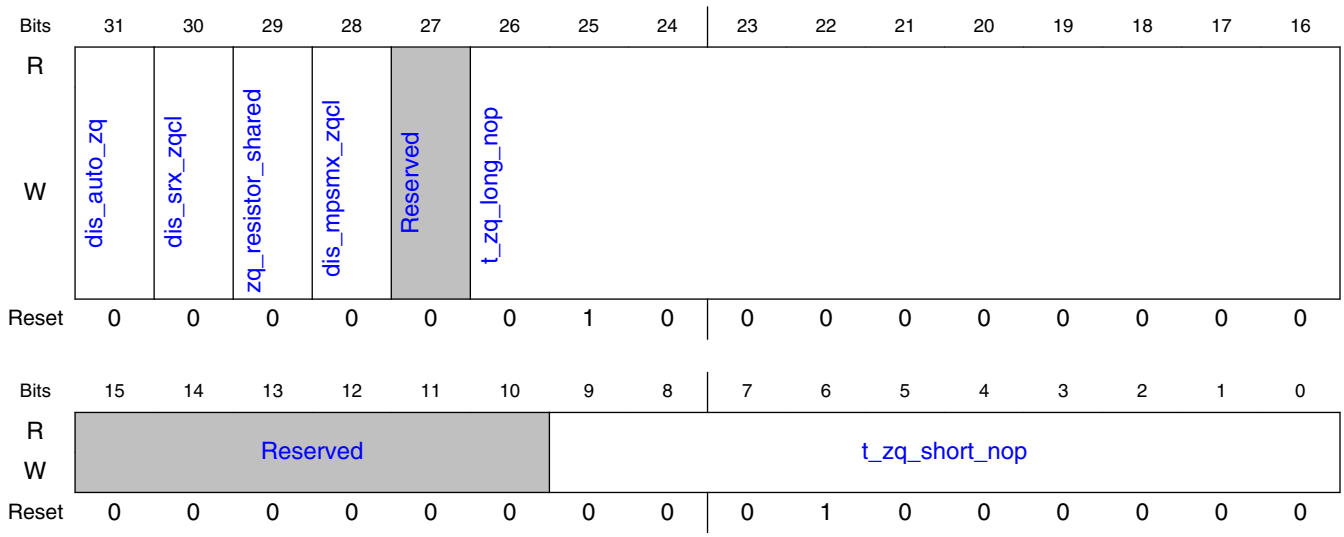
Field	Function
31 en_dfi_lp_t_stab	Enable DFI tSTAB 0b - Disable using tSTAB when exiting DFI LP 1b - Enable using tSTAB when exiting DFI LP. Needs to be set when the PHY is stopping the clock during DFI LP to save maximum power.
30-0 —	Reserved

9.2.5.1.58 ZQ Control Register 0 (ZQCTL0)

9.2.5.1.58.1 Offset

Register	Offset
ZQCTL0	180h

9.2.5.1.58.2 Diagram



9.2.5.1.58.3 Fields

Field	Function
31 dis_auto_zq	Disable Auto ZQCS/MPC 0b - Internally generate ZQCS/MPC(ZQ calibration) commands based on ZQCTL1.t_zq_short_interval_x1024. 1b - Disable DDRC generation of ZQCS/MPC(ZQ calibration) command. Register DBGCMD.zq_calib_short can be used instead to issue ZQ calibration request from APB module.
30 dis_srx_zqcl	Disable ZQCL/MPC 0b - Enable issuing of ZQCL/MPC(ZQ calibration) command at Self-Refresh/SR-Powerdown exit. 1b - Disable issuing of ZQCL/MPC(ZQ calibration) command at Self-Refresh/SR-Powerdown exit.
29 zq_resistor_shared	ZQ resistor sharing 0b - ZQ resistor is not shared. 1b - Denotes that ZQ resistor is shared between ranks. Means ZQinit/ZQCL/ZQCS/MPC(ZQ calibration) commands are sent to one rank at a time with tZQinit/tZQCL/tZQCS/tZQCAL/tZQLAT timing met between commands so that commands to different ranks do not overlap.
28 dis_mpsmx_zqcl	Do not issue ZQCL command at Maximum Power Save Mode exit if the DDRC_SHARED_AC configuration parameter is set. Program it to 1'b1. The software can send ZQCS after exiting MPSM mode. 0b - Enable issuing of ZQCL command at Maximum Power Saving Mode exit. Only applicable when run in DDR4 mode. This is only present for designs supporting DDR4 devices. 1b - Disable issuing of ZQCL command at Maximum Power Saving Mode exit. Only applicable when run in DDR4 mode.
27 —	Reserved
26-16 t_zq_long_nop	tZQoper for DDR4, tZQCAL for LPDDR4: Number of DFI clock cycles of NOP required after a ZQCL (ZQ calibration long)/MPC(ZQ Start) command is issued to SDRAM. When the controller is operating in 1:2 frequency ratio mode: DDR4: program this to tZQoper/2 and round it up to the next integer value. LPDDR4: program this to tZQCAL/2 and round it up to the next integer value.
15-10	Reserved

Table continues on the next page...

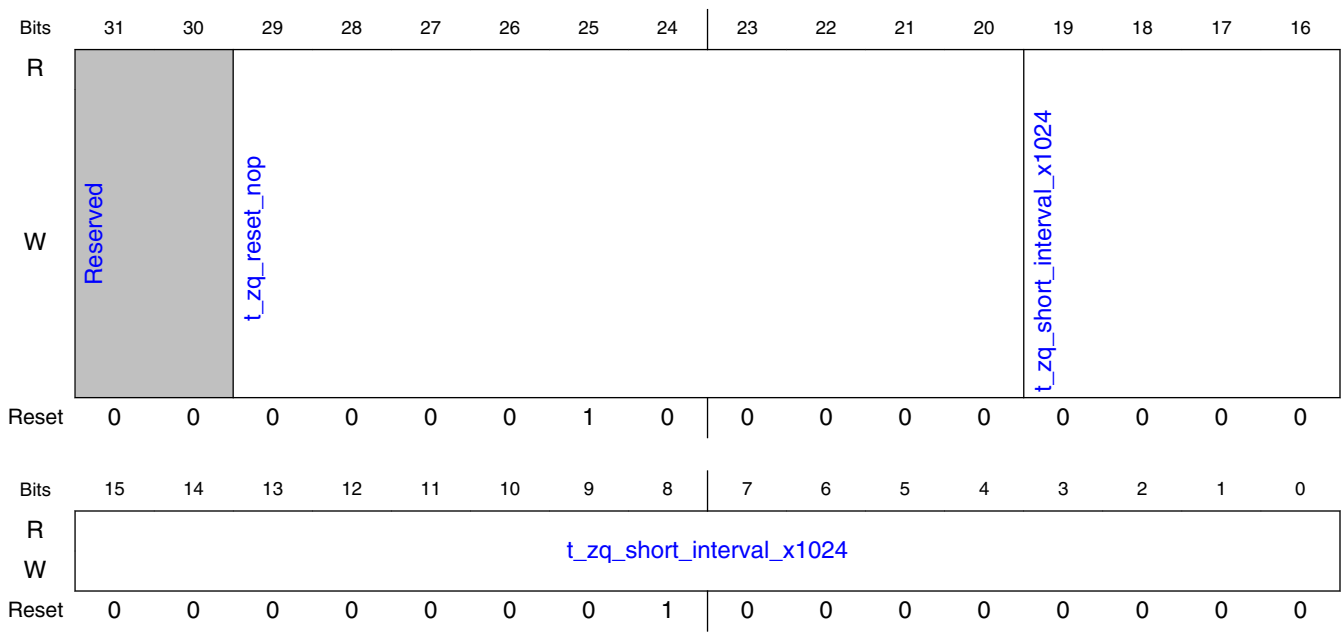
Field	Function
—	
9-0 t_zq_short_nop	tZQCS for DDR4, tZQLAT for LPDDR4: Number of DFI clock cycles of NOP required after a ZQCS (ZQ calibration short)/MPC(ZQ Latch) command is issued to SDRAM. When the controller is operating in 1:2 frequency ratio mode, program this to tZQCS/2 and round it up to the next integer value.

9.2.5.1.59 ZQ Control Register 1 (ZQCTL1)

9.2.5.1.59.1 Offset

Register	Offset
ZQCTL1	184h

9.2.5.1.59.2 Diagram



9.2.5.1.59.3 Fields

Field	Function
31-30 —	Reserved

Table continues on the next page...

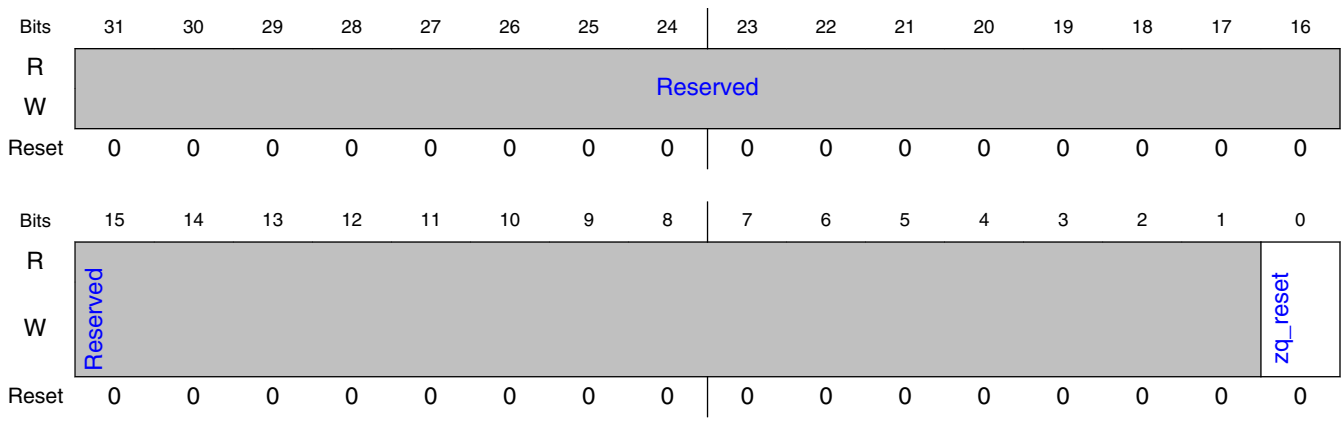
Field	Function
29-20 t_zq_reset_nop	tZQReset: Number of DFI clock cycles of NOP required after a ZQReset (ZQ calibration Reset) command is issued to SDRAM. When the controller is operating in 1:2 frequency ratio mode, program this to tZQReset/2 and round it up to the next integer value. This is only present for LPDDR4.
19-0 t_zq_short_interval_x1024	Average interval to wait between automatically issuing ZQCS (ZQ calibration short)/MPC(ZQ calibration) commands to DDR4/LPDDR4 devices. Meaningless, if ZQCTL0.dis_auto_zq=1. Unit: 1024 DFI clock cycles.

9.2.5.1.60 ZQ Control Register 2 (ZQCTL2)

9.2.5.1.60.1 Offset

Register	Offset
ZQCTL2	188h

9.2.5.1.60.2 Diagram



9.2.5.1.60.3 Fields

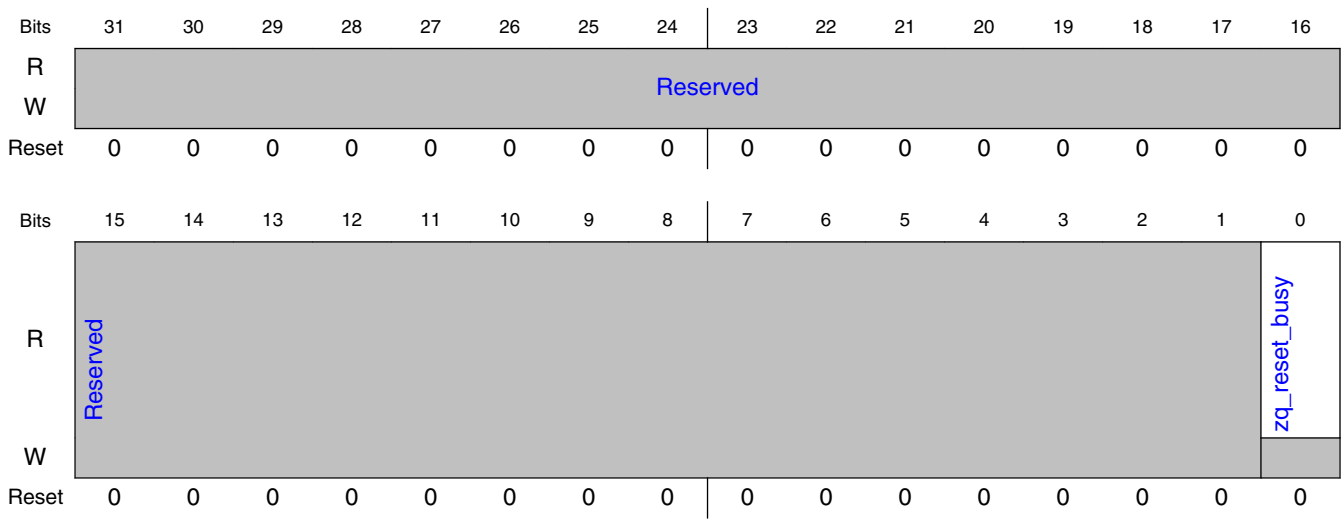
Field	Function
31-1 —	Reserved
0 zq_reset	Setting this register bit to 1 triggers a ZQ Reset operation. When the ZQ Reset operation is complete, the DDRC automatically clears this bit. It is recommended NOT to set this signal if in Init, Self-Refresh(except LPDDR4) or SR-Powerdown(LPDDR4) or Deep power-down operating modes. This is only present for LPDDR4.

9.2.5.1.61 ZQ Status Register (ZQSTAT)

9.2.5.1.61.1 Offset

Register	Offset
ZQSTAT	18Ch

9.2.5.1.61.2 Diagram



9.2.5.1.61.3 Fields

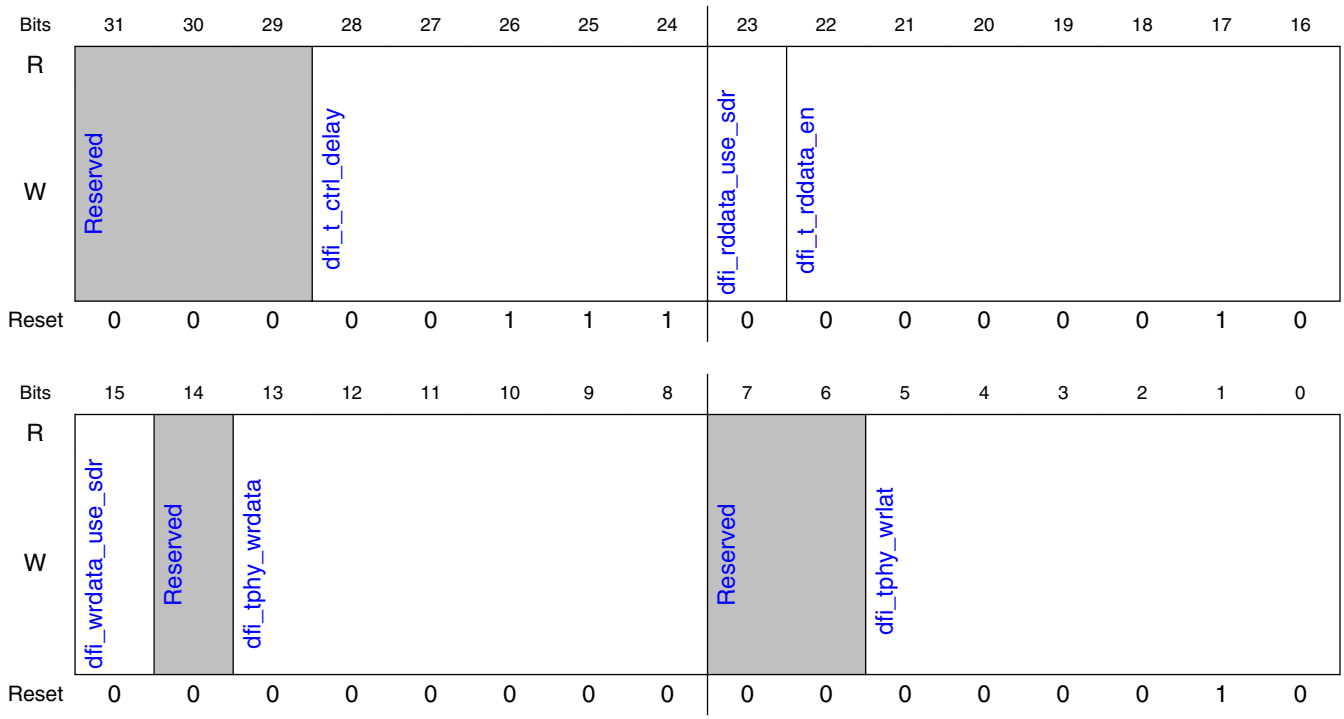
Field	Function
31-1 —	Reserved
0 zq_reset_busy	SoC core may initiate a ZQ Reset operation only if this signal is low. This signal goes high in the clock after the DDRC accepts the ZQ Reset request. It goes low when the ZQ Reset command is issued to the SDRAM and the associated NOP period is over. It is recommended not to perform ZQ Reset commands when this signal is high. 0b - Indicates that the SoC core can initiate a ZQ Reset operation 1b - Indicates that ZQ Reset operation is in progress

9.2.5.1.62 DFI Timing Register 0 (DFITMG0)

9.2.5.1.62.1 Offset

Register	Offset
DFITMG0	190h

9.2.5.1.62.2 Diagram



9.2.5.1.62.3 Fields

Field	Function
31-29 —	Reserved
28-24 dfi_t_ctrl_delay	Specifies the number of DFI clock cycles after an assertion or de-assertion of the DFI control signals that the control signals at the PHY-DRAM interface reflect the assertion or de-assertion. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value..
23 dfi_rddata_use_sdr	Defines whether dfi_rddata_en/dfi_rddata/dfi_rddata_valid is generated using HDR (DFI clock) or SDR (DFI PHY clock) values. Selects whether value in DFITMG0.dfi_t_rddata_en is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles: - 0 in terms of HDR (DFI clock) cycles - 1 in terms of SDR (DFI PHY clock) cycles Refer to PHY specification for correct value.
22-16 dfi_t_rddata_en	Time from the assertion of a read command on the DFI interface to the assertion of the dfi_rddata_en signal. Refer to PHY specification for correct value. This corresponds to the DFI parameter trddata_en.Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_rddata_use_sdr.

Table continues on the next page...

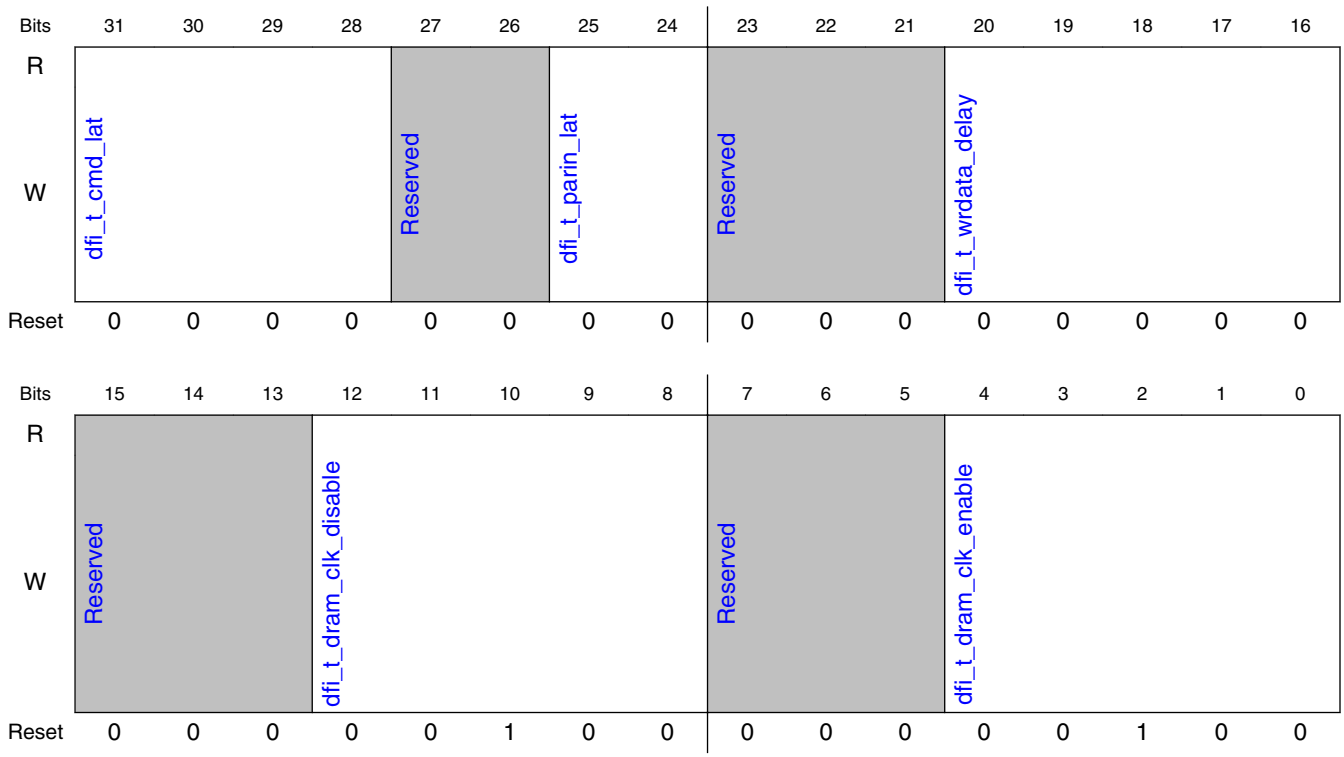
Field	Function
15 dfi_wrdata_use_sdr	Defines whether dfi_wrdata_en/dfi_wrdata/dfi_wrdata_mask is generated using HDR (DFI clock) or SDR (DFI PHY clock) values Selects whether value in DFITMG0.dfi_tphy_wrlat is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles Selects whether value in DFITMG0.dfi_tphy_wrdata is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles - 0 in terms of HDR (DFI clock) cycles - 1 in terms of SDR (DFI PHY clock) cycles Refer to PHY specification for correct value.
14 —	Reserved
13-8 dfi_tphy_wrdata	Specifies the number of clock cycles between when dfi_wrdata_en is asserted to when the associated write data is driven on the dfi_wrdata signal. This corresponds to the DFI timing parameter tphy_wrdata. Refer to PHY specification for correct value. Note, max supported value is 8. Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_wrdata_use_sdr.
7-6 —	Reserved
5-0 dfi_tphy_wrlat	Write latency Number of clocks from the write command to write data enable (dfi_wrdata_en). This corresponds to the DFI timing parameter tphy_wrlat. Refer to PHY specification for correct value. Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_wrdata_use_sdr.

9.2.5.1.63 DFI Timing Register 1 (DFITMG1)

9.2.5.1.63.1 Offset

Register	Offset
DFITMG1	194h

9.2.5.1.63.2 Diagram



9.2.5.1.63.3 Fields

Field	Function
31-28 dfi_t_cmd_lat	Specifies the number of DFI PHY clock cycles between when the dfi_cs signal is asserted and when the associated command is driven. This field is used for CAL mode, should be set to '0' or the value which matches the CAL mode register setting in the DRAM. If the PHY can add the latency for CAL mode, this should be set to '0'. Valid Range: 0, 3, 4, 5, 6, and 8
27-26 —	Reserved
25-24 dfi_t_parin_lat	Specifies the number of DFI PHY clock cycles between when the dfi_cs signal is asserted and when the associated dfi_parity_in signal is driven.
23-21 —	Reserved
20-16 dfi_t_wrdata_delay	Specifies the number of DFI clock cycles between when the dfi_wrdata_en signal is asserted and when the corresponding write data transfer is completed on the DRAM bus. This corresponds to the DFI timing parameter twrdata_delay. Refer to PHY specification for correct value. For DFI 3.0 PHY, set to twrdata_delay, a new timing parameter introduced in DFI 3.0. For DFI 2.1 PHY, set to tphy_wrdata + (delay of DFI write data to the DRAM). Value to be programmed is in terms of DFI clocks, not PHY clocks. In FREQ_RATIO=2, divide PHY's value by 2 and round up to next integer. If using DFITMG0.dfi_wrdata_use_sdr=1, add 1 to the value. Unit: Clocks
15-13 —	Reserved

Table continues on the next page...

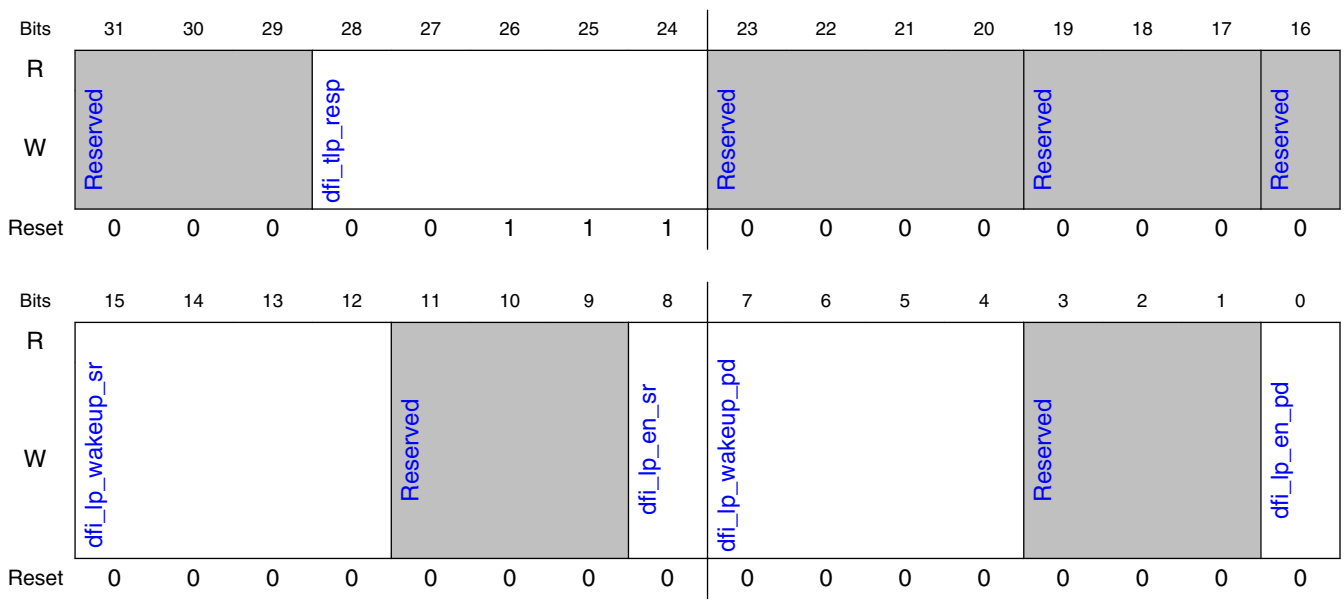
Field	Function
12-8 dfi_t_dram_clk_disable	Specifies the number of DFI clock cycles from the assertion of the dfi_dram_clk_disable signal on the DFI until the clock to the DRAM memory devices, at the PHY-DRAM boundary, maintains a low value. If the DFI clock and the memory clock are not phase aligned, this timing parameter should be rounded up to the next integer value.
7-5 —	Reserved
4-0 dfi_t_dram_clk_enable	Specifies the number of DFI clock cycles from the de-assertion of the dfi_dram_clk_disable signal on the DFI until the first valid rising edge of the clock to the DRAM memory devices, at the PHY-DRAM boundary. If the DFI clock and the memory clock are not phase aligned, this timing parameter should be rounded up to the next integer value.

9.2.5.1.64 DFI Low Power Configuration Register 0 (DFILPCFG0)

9.2.5.1.64.1 Offset

Register	Offset
DFILPCFG0	198h

9.2.5.1.64.2 Diagram



9.2.5.1.64.3 Fields

Field	Function
31-29 —	Reserved
28-24 dfi_tlp_resp	Setting in DFI clock cycles for DFI's tlp_resp time. Same value is used for both Power Down, Self Refresh, Deep Power Down and Maximum Power Saving modes. DFI 2.1 specification onwards, recommends using a fixed value of 7 always.
23-20 —	Reserved
19-17 —	Reserved
16 —	Reserved
15-12 dfi_lp_wakeup_sr	Value in DFI clpck cycles to drive on dfi_lp_wakeup signal when Self Refresh mode is entered. Determines the DFI's tlp_wakeup time: 0000b - 16 cycles 0001b - 32 cycles 0010b - 64 cycles 0011b - 128 cycles 0100b - 256 cycles 0101b - 512 cycles 0110b - 1024 cycles 0111b - 2048 cycles 1000b - 4096 cycles 1001b - 8192 cycles 1010b - 16384 cycles 1011b - 32768 cycles 1100b - 65536 cycles 1101b - 131072 cycles 1110b - 262144 cycles 1111b - Unlimited cycles
11-9 —	Reserved
8 dfi_lp_en_sr	Enables DFI Low Power interface handshaking during Self Refresh Entry/Exit. - 0 - Disabled - 1 - Enabled 0b - Disabled 1b - Enabled
7-4 dfi_lp_wakeup_pd	Value in DFI clock cycles to drive on dfi_lp_wakeup signal when Power Down mode is entered. Determines the DFI's tlp_wakeup time: 0000b - 16 cycles 0001b - 32 cycles 0010b - 64 cycles 0011b - 128 cycles 0100b - 256 cycles 0101b - 512 cycles 0110b - 1024 cycles 0111b - 2048 cycles 1000b - 4096 cycles 1001b - 8192 cycles 1010b - 16384 cycles 1011b - 32768 cycles 1100b - 65536 cycles

Table continues on the next page...

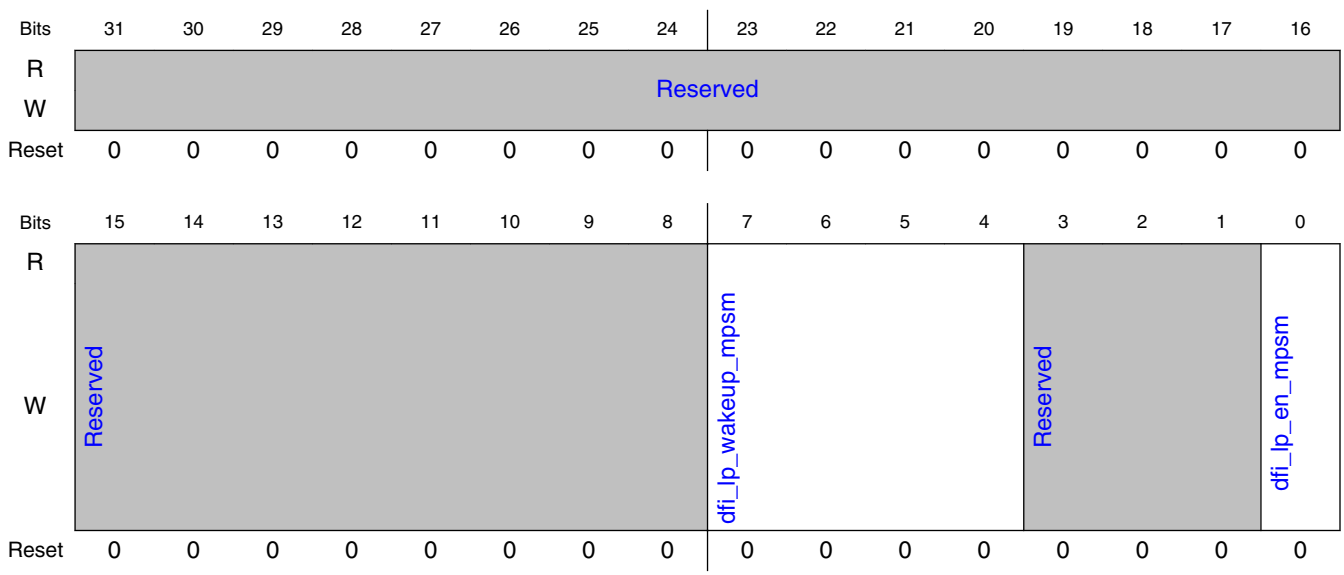
Field	Function
	1101b - 131072 cycles 1110b - 262144 cycles 1111b - Unlimited cycles
3-1 —	Reserved
0 dfi_lp_en_pd	Enables DFI Low Power interface handshaking during Power Down Entry/Exit. - 0 - Disabled - 1 - Enabled

9.2.5.1.65 DFI Low Power Configuration Register 1 (DFILPCFG1)

9.2.5.1.65.1 Offset

Register	Offset
DFILPCFG1	19Ch

9.2.5.1.65.2 Diagram



9.2.5.1.65.3 Fields

Field	Function
31-8 —	Reserved

Table continues on the next page...

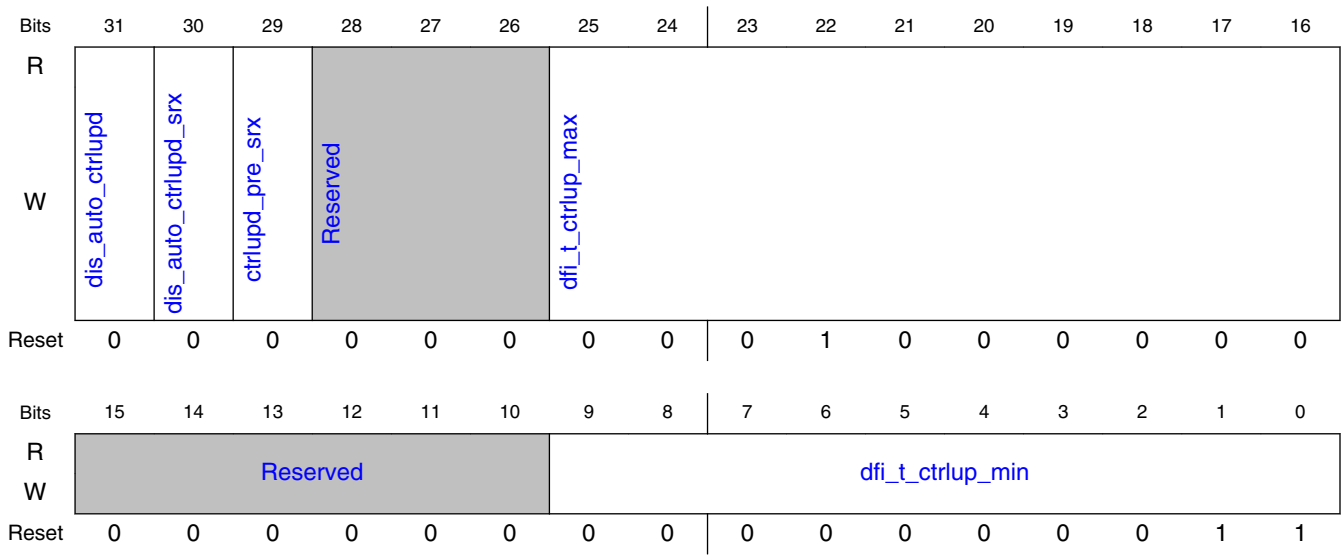
Field	Function
7-4 dfi_lp_wakeup_mpsm	Value in DFI clock cycles to drive on dfi_lp_wakeup signal when Maximum Power Saving Mode is entered. Determines the DFI's tlp_wakeup time: 0000b - 16 cycles 0001b - 32 cycles 0010b - 64 cycles 0011b - 128 cycles 0100b - 256 cycles 0101b - 512 cycles 0110b - 1024 cycles 0111b - 2048 cycles 1000b - 4096 cycles 1001b - 8192 cycles 1010b - 16384 cycles 1011b - 32768 cycles 1100b - 65536 cycles 1101b - 131072 cycles 1110b - 262144 cycles 1111b - Unlimited cycles
3-1 —	Reserved
0 dfi_lp_en_mpsm	Enables DFI Low Power interface handshaking during Maximum Power Saving Mode Entry/Exit. - 0 - Disabled - 1 - Enabled This is only present for designs supporting DDR4 devices.

9.2.5.1.66 DFI Update Register 0 (DFIUPD0)

9.2.5.1.66.1 Offset

Register	Offset
DFIUPD0	1A0h

9.2.5.1.66.2 Diagram



9.2.5.1.66.3 Fields

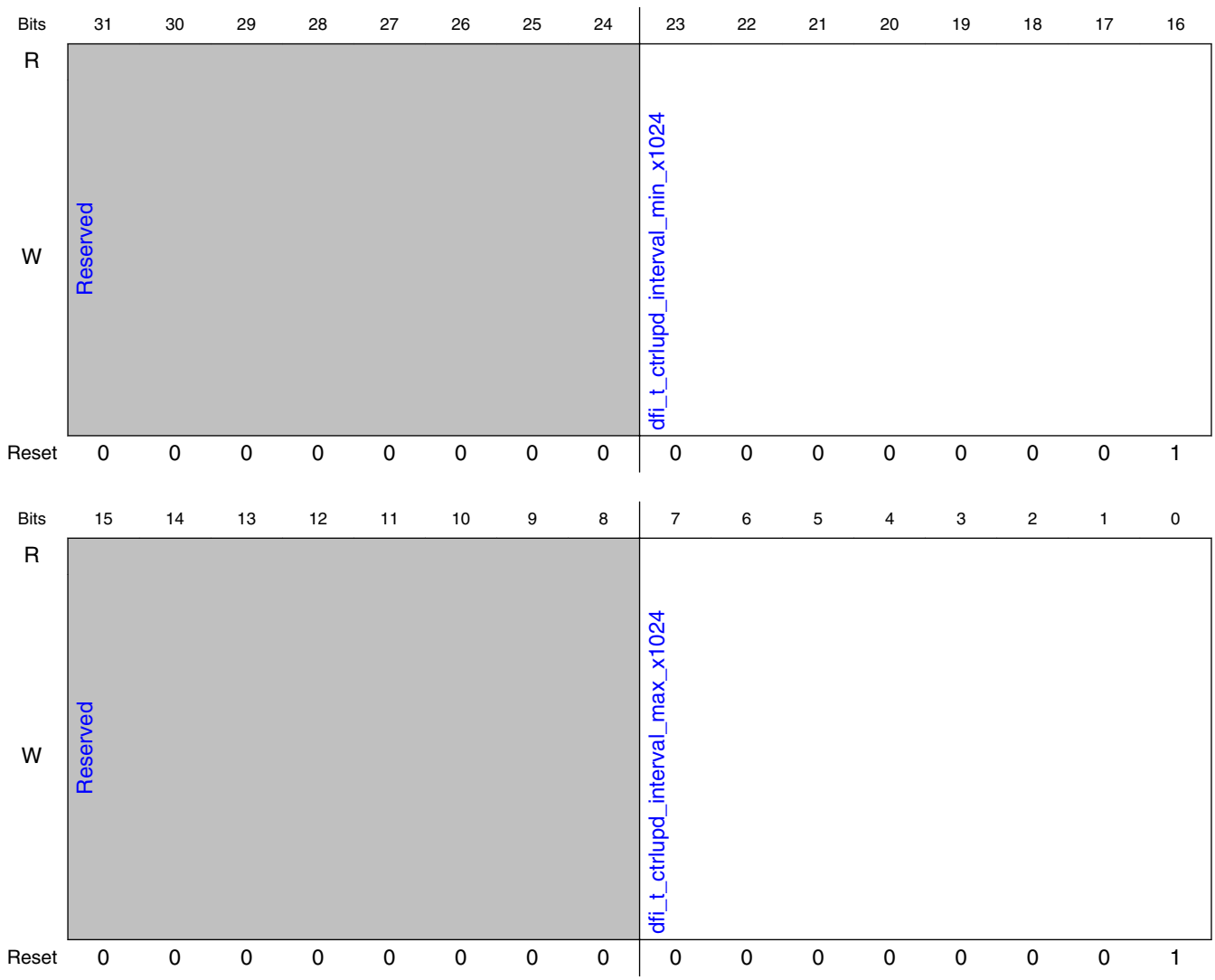
Field	Function
31 dis_auto_ctrlupd	automatic dfi_ctrlupd_req generation by the DDRC 0b - DDRC issues dfi_ctrlupd_req periodically. 1b - disable the automatic dfi_ctrlupd_req generation by the DDRC. The core must issue the dfi_ctrlupd_req signal using register reg_ddrc_ctrlupd.
30 dis_auto_ctrlupd_srx	Auto ctrlupd request generation 0b - DDRC issues a dfi_ctrlupd_req before or after exiting self-refresh, depending on DFIUPD0.ctrlupd_pre_srx. 1b - disable the automatic dfi_ctrlupd_req generation by the DDRC at self-refresh exit.
29 ctrlupd_pre_srx	Selects dfi_ctrlupd_req requirements at SRX: - 0 : send ctrlupd after SRX - 1 : send ctrlupd before SRX If DFIUPD0.dis_auto_ctrlupd_srx=1, this register has no impact, because no dfi_ctrlupd_req will be issued when SRX. 0b - send ctrlupd after SRX 1b - send ctrlupd before SRX
28-26 —	Reserved
25-16 dfi_t_ctrlup_max	Specifies the maximum number of DFI clock cycles that the dfi_ctrlupd_req signal can assert. Lowest value to assign to this variable is 0x40.
15-10 —	Reserved
9-0 dfi_t_ctrlup_min	Specifies the minimum number of DFI clock cycles that the dfi_ctrlupd_req signal must be asserted. The DDRC expects the PHY to respond within this time. If the PHY does not respond, the DDRC will de-assert dfi_ctrlupd_req after dfi_t_ctrlup_min + 2 cycles. Lowest value to assign to this variable is 0x3.

9.2.5.1.67 DFI Update Register 1 (DFIUPD1)

9.2.5.1.67.1 Offset

Register	Offset
DFIUPD1	1A4h

9.2.5.1.67.2 Diagram



9.2.5.1.67.3 Fields

Field	Function
31-24	Reserved

Table continues on the next page...

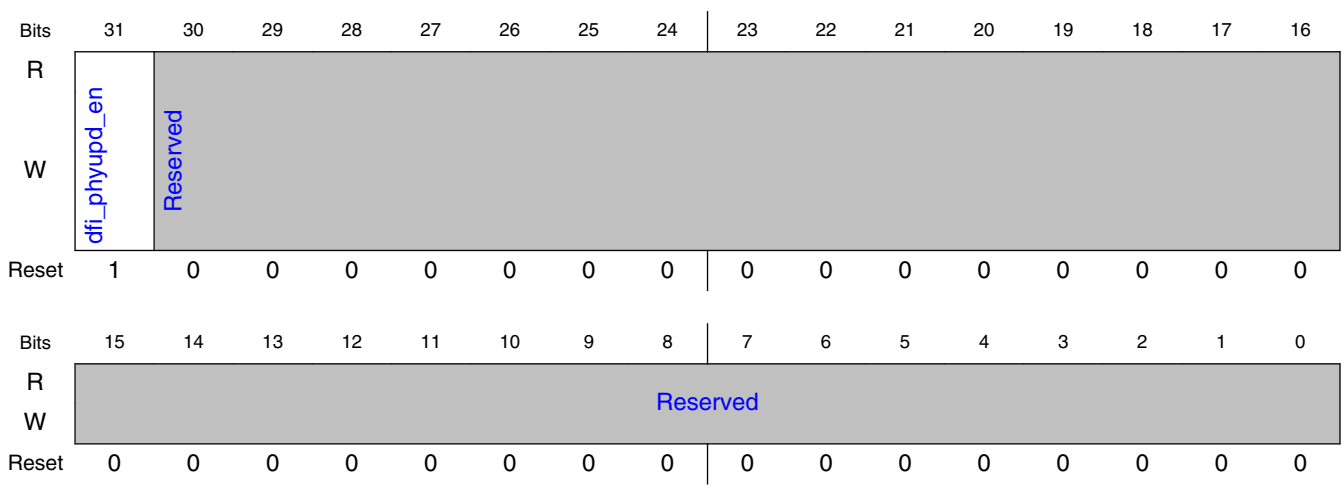
Field	Function
—	
23-16 dfi_t_ctrlupd_interval_min_x1024	This is the minimum amount of time between DDRC initiated DFI update requests (which is executed whenever the DDRC is idle). Set this number higher to reduce the frequency of update requests, which can have a small impact on the latency of the first read request when the DDRC is idle. Minimum allowed value for this field is 1. Unit: 1024 DFI clock cycles
15-8 —	Reserved
7-0 dfi_t_ctrlupd_interval_max_x1024	This is the maximum amount of time between DDRC initiated DFI update requests. This timer resets with each update request; when the timer expires dfi_ctrlupd_req is sent and traffic is blocked until the dfi_ctrlupd_ackx is received. PHY can use this idle time to recalibrate the delay lines to the DLLs. The DFI controller update is also used to reset PHY FIFO pointers in case of data capture errors. Updates are required to maintain calibration over PVT, but frequent updates may impact performance. Minimum allowed value for this field is 1. Note: Value programmed for DFIUPD1.dfi_t_ctrlupd_interval_max_x1024 must be greater than DFIUPD1.dfi_t_ctrlupd_interval_min_x1024. Unit: 1024 DFI clock cycles

9.2.5.1.68 DFI Update Register 2 (DFIUPD2)

9.2.5.1.68.1 Offset

Register	Offset
DFIUPD2	1A8h

9.2.5.1.68.2 Diagram



9.2.5.1.68.3 Fields

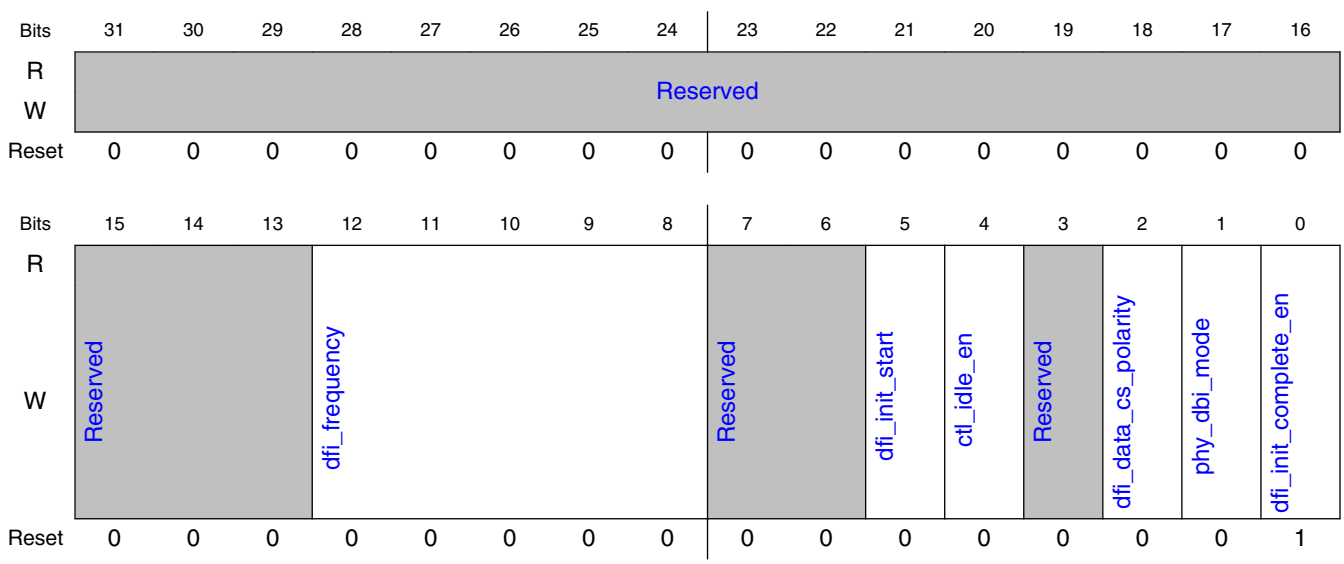
Field	Function
31 dfi_phyupd_en	Enables the support for acknowledging PHY-initiated updates: 0b - Disabled 1b - Enabled
30-0 —	Reserved

9.2.5.1.69 DFI Miscellaneous Control Register (DFIMISC)

9.2.5.1.69.1 Offset

Register	Offset
DFIMISC	1B0h

9.2.5.1.69.2 Diagram



9.2.5.1.69.3 Fields

Field	Function
31-13 —	Reserved

Table continues on the next page...

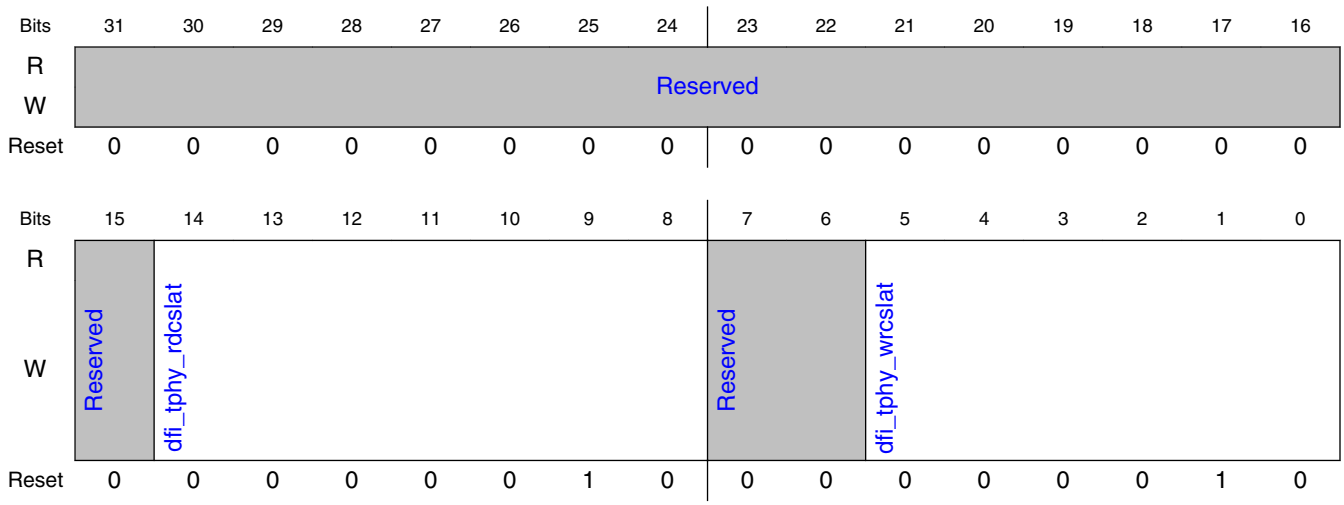
Field	Function
12-8 dfi_frequency	Indicates the operating frequency of the system. The number of supported frequencies and the mapping of signal values to clock frequencies are defined by the PHY.
7-6 —	Reserved
5 dfi_init_start	PHY init start request signal.When asserted it triggers the PHY init start request
4 ctl_idle_en	Enables support of ctl_idle signal, which is non-DFI related pin specific to certain PHYs. See signal description of ctl_idle signal for further details of ctl_idle functionality.
3 —	Reserved
2 dfi_data_cs_polarity	Defines polarity of dfi_wrdata_cs and dfi_rddata_cs signals. 0b - Signals are active low 1b - Signals are active high
1 phy_dbi_mode	DBI implemented in DDRC or PHY. Present only in designs configured to support DDR4 and LPDDR4. 0b - DDRC implements DBI functionality. 1b - PHY implements DBI functionality.
0 dfi_init_complete_en	PHY initialization complete enable signal. When asserted the dfi_init_complete signal can be used to trigger SDRAM initialisation

9.2.5.1.70 DFI Timing Register 2 (DFITMG2)

9.2.5.1.70.1 Offset

Register	Offset
DFITMG2	1B4h

9.2.5.1.70.2 Diagram



9.2.5.1.70.3 Fields

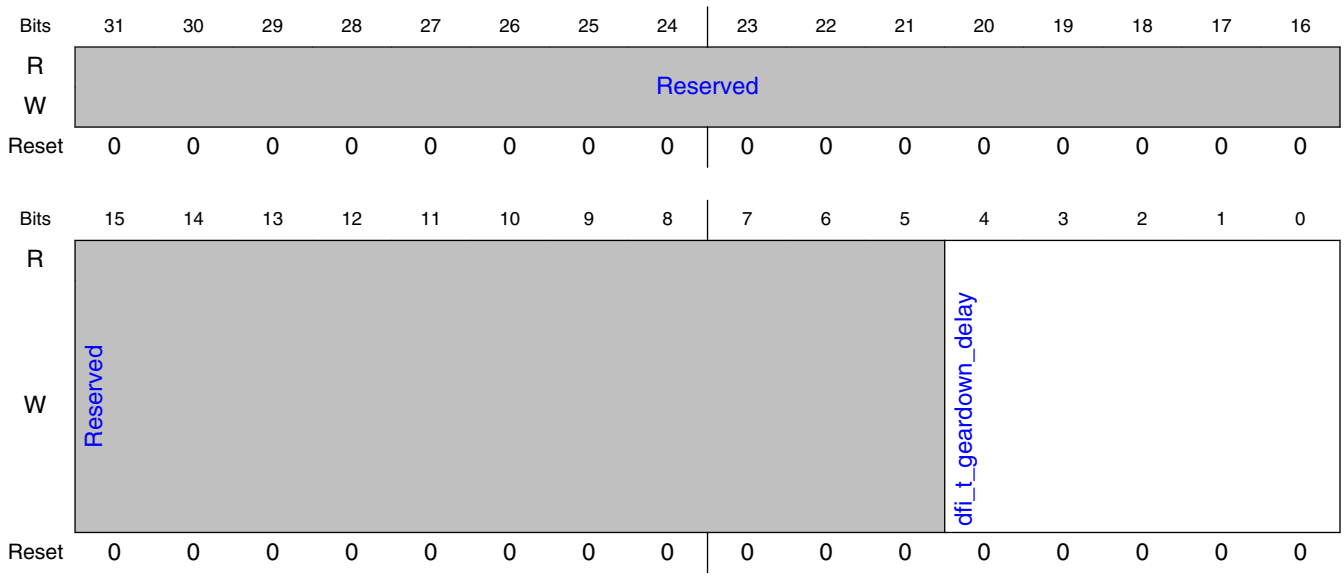
Field	Function
31-15 —	Reserved
14-8 dfi_tphy_rdcslat	Number of DFI PHY clock cycles between when a read command is sent on the DFI control interface and when the associated dfi_rddata_cs signal is asserted. This corresponds to the DFI timing parameter tphy_rdcslat. Refer to PHY specification for correct value.
7-6 —	Reserved
5-0 dfi_tphy_wrcslat	Number of DFI PHY clock cycles between when a write command is sent on the DFI control interface and when the associated dfi_wrdata_cs signal is asserted. This corresponds to the DFI timing parameter tphy_wrcslat. Refer to PHY specification for correct value.

9.2.5.1.71 DFI Timing Register 3 (DFITMG3)

9.2.5.1.71.1 Offset

Register	Offset
DFITMG3	1B8h

9.2.5.1.71.2 Diagram



9.2.5.1.71.3 Fields

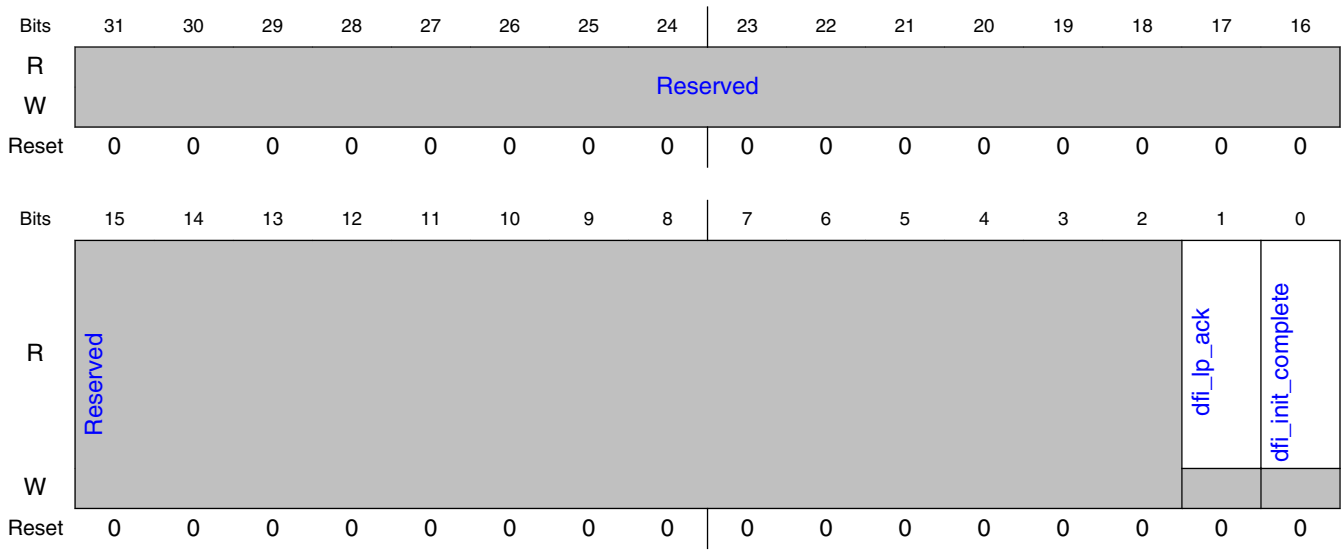
Field	Function
31-5 —	Reserved
4-0 dfi_t_geardown_delay	The delay from dfi_geardown_en assertion to the time of the PHY being ready to receive commands. Refer to PHY specification for correct value. When the controller is operating in 1:2 frequency ratio mode, program this to (tgeardown_delay/2) and round it up to the next integer value. Unit: Clocks

9.2.5.1.72 DFI Status Register (DFISTAT)

9.2.5.1.72.1 Offset

Register	Offset
DFISTAT	1BCh

9.2.5.1.72.2 Diagram



9.2.5.1.72.3 Fields

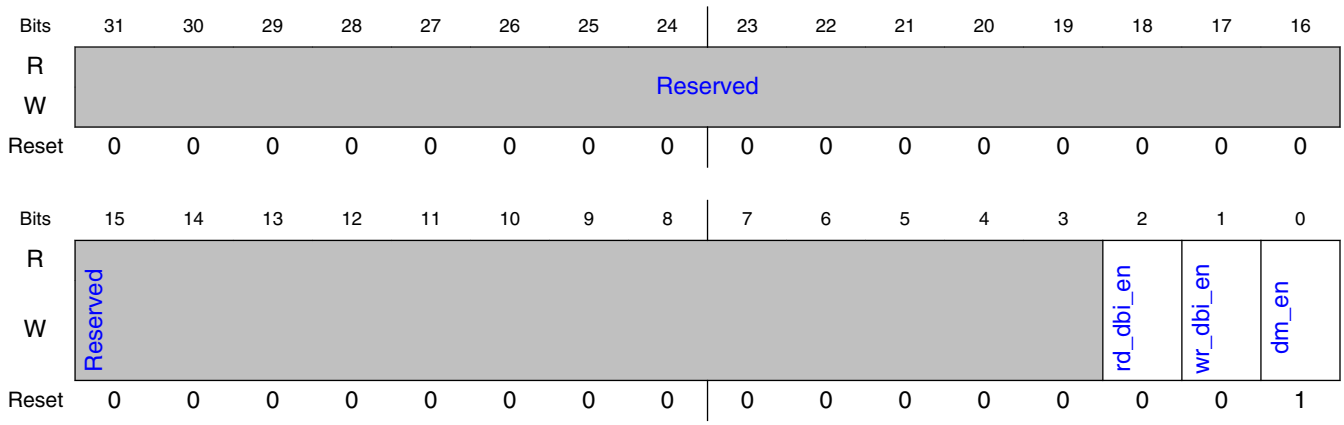
Field	Function
31-2 —	Reserved
1 dfi_lp_ack	Stores the value of the dfi_lp_ack input to the controller.
0 dfi_init_complete	The status flag register which announces when the DFI initialization has been completed. The DFI INIT triggered by dfi_init_start signal and then the dfi_init_complete flag is polled to know when the initialization is done.

9.2.5.1.73 DM/DBI Control Register (DBICTL)

9.2.5.1.73.1 Offset

Register	Offset
DBICTL	1C0h

9.2.5.1.73.2 Diagram



9.2.5.1.73.3 Fields

Field	Function
31-3 —	Reserved
2 rd_dbi_en	Read DBI enable signal in DDRC. - 0 - Read DBI is disabled. - 1 - Read DBI is enabled. This signal must be set the same value as DRAM's mode register. - DDR4: MR5 bit A12. When x4 devices are used, this signal must be set to 0. - LPDDR4: MR3[6]
1 wr_dbi_en	This signal must be set the same value as DRAM's mode register. - DDR4: MR5 bit A11. When x4 devices are used, this signal must be set to 0. - LPDDR4: MR3[7] 0b - Write DBI is disabled 1b - Write DBI is enabled.
0 dm_en	DM enable signal in DDRC. This signal must be set the same logical value as DRAM's mode register. - DDR4: Set this to same value as MR5 bit A10. When x4 devices are used, this signal must be set to 0. - LPDDR4: Set this to inverted value of MR13[5] which is opposite polarity from this signal 0b - DM is disabled 1b - DM is enabled

9.2.5.1.74 Address Map Register 0 (ADDRMAP0)

9.2.5.1.74.1 Offset

Register	Offset
ADDRMAP0	200h

9.2.5.1.74.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											addrmap_cs_bit0				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.74.3 Fields

Field	Function
31-5 —	Reserved
4-0 addrmap_cs_bit0	Selects the HIF address bit used as rank address bit 0. Valid Range: 0 to 28, and 31 Internal Base: 6 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 31, rank address bit 0 is set to 0.

9.2.5.1.75 Address Map Register 1 (ADDRMAP1)

9.2.5.1.75.1 Offset

Register	Offset
ADDRMAP1	204h

9.2.5.1.75.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											addrmap_bank_b2				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			addrmap_bank_b1					Reserved			addrmap_bank_b0				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.75.3 Fields

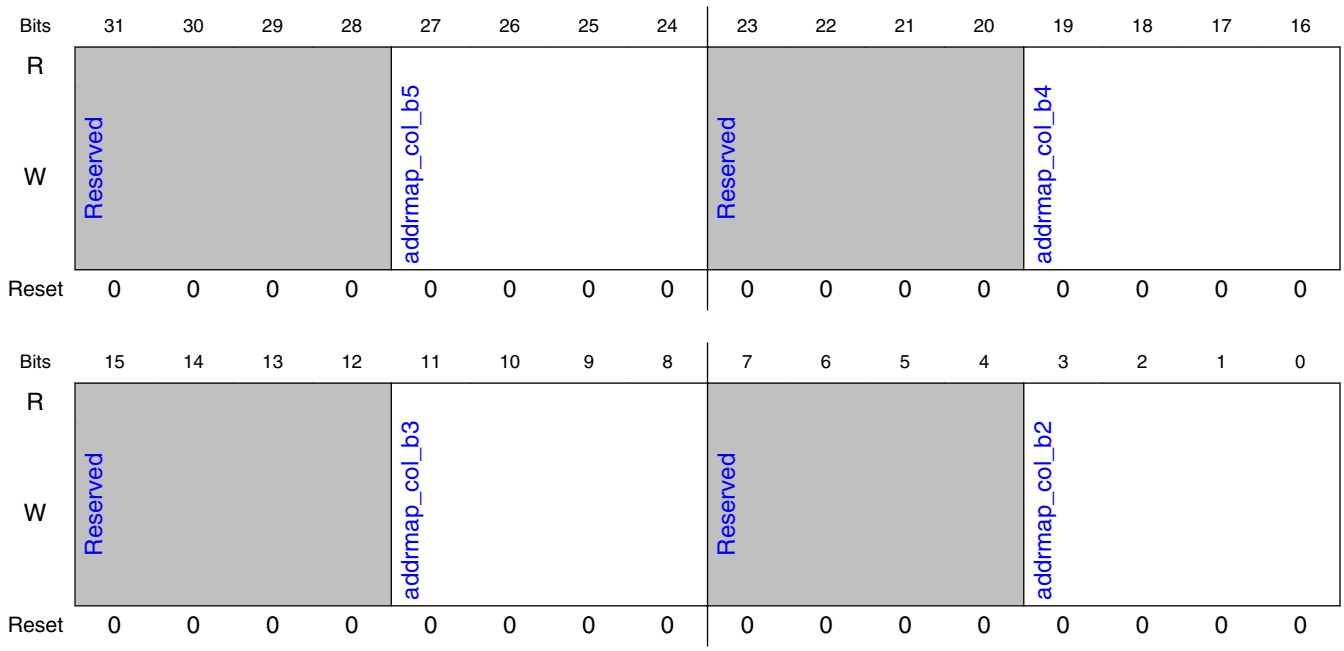
Field	Function
31-21 —	Reserved
20-16 addrmap_bank_b2	Selects the HIF address bit used as bank address bit 2. Valid Range: 0 to 30 and 31 Internal Base: 4 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 31, bank address bit 2 is set to 0.
15-13 —	Reserved
12-8 addrmap_bank_b1	Selects the HIF address bits used as bank address bit 1. Valid Range: 0 to 31 Internal Base: 3 The selected HIF address bit for each of the bank address bits is determined by adding the internal base to the value of this field.
7-5 —	Reserved
4-0 addrmap_bank_b0	Selects the HIF address bits used as bank address bit 0. Valid Range: 0 to 31 Internal Base: 2 The selected HIF address bit for each of the bank address bits is determined by adding the internal base to the value of this field.

9.2.5.1.76 Address Map Register 2 (ADDRMAP2)

9.2.5.1.76.1 Offset

Register	Offset
ADDRMAP2	208h

9.2.5.1.76.2 Diagram



9.2.5.1.76.3 Fields

Field	Function
31-28 —	Reserved
27-24 addrmap_col_b5	- Full bus width mode: Selects the HIF address bit used as column address bit 5. - Half bus width mode: Selects the HIF address bit used as column address bit 6. - Quarter bus width mode: Selects the HIF address bit used as column address bit 7 . Valid Range: 0 to 7, and 15 Internal Base: 5 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0.
23-20 —	Reserved
19-16 addrmap_col_b4	- Full bus width mode: Selects the HIF address bit used as column address bit 4. - Half bus width mode: Selects the HIF address bit used as column address bit 5. - Quarter bus width mode: Selects the HIF address bit used as column address bit 6. Valid Range: 0 to 7, and 15 Internal Base: 4 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0.
15-12 —	Reserved
11-8 addrmap_col_b3	- Full bus width mode: Selects the HIF address bit used as column address bit 3. - Half bus width mode: Selects the HIF address bit used as column address bit 4. - Quarter bus width mode: Selects the HIF address bit used as column address bit 5. Valid Range: 0 to 7 Internal Base: 3 The selected HIF address bit is determined by adding the internal base to the value of this field. Note, if DDRC_INCL_ARB=1, MEMC_BURST_LENGTH=16, Full bus width (MSTR.data_bus_width=00) and BL16 (MSTR.burst_rdwr=1000), it is recommended to program this to 0.
7-4	Reserved

Table continues on the next page...

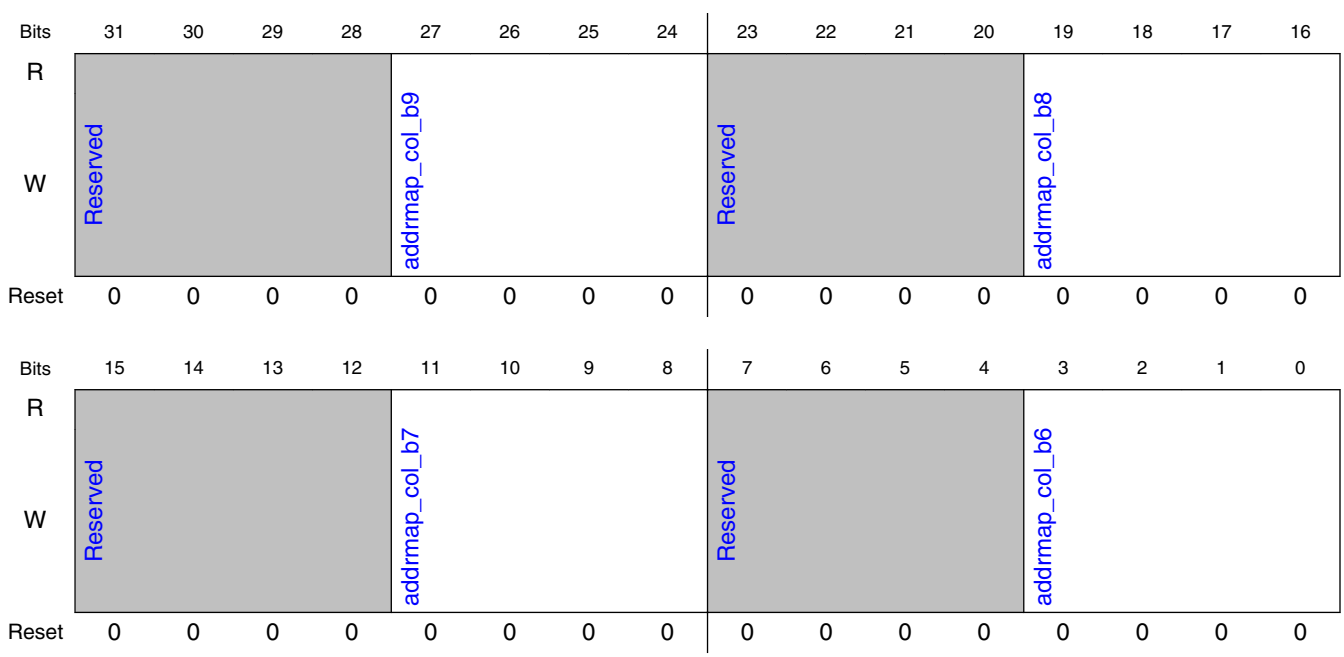
Field	Function
—	
3-0 addrmap_col_b2	- Full bus width mode: Selects the HIF address bit used as column address bit 2. - Half bus width mode: Selects the HIF address bit used as column address bit 3. - Quarter bus width mode: Selects the HIF address bit used as column address bit 4. Valid Range: 0 to 7 Internal Base: 2 The selected HIF address bit is determined by adding the internal base to the value of this field. Note, if DDRC_INCL_ARB=1 and MEMC_BURST_LENGTH=8, it is required to program this to 0 unless: - in Half or Quarter bus width (MSTR.data_bus_width!=00) and - PCCFG.bl_exp_mode==1 and either - In DDR4 and ADDRMAP8.addrmap_bg_b0==0 or - In LPDDR4 and ADDRMAP1.addrmap_bank_b0==0 If DDRC_INCL_ARB=1 and MEMC_BURST_LENGTH=16, it is required to program this to 0 unless: - in Half or Quarter bus width (MSTR.data_bus_width!=00) and - PCCFG.bl_exp_mode==1 and - In DDR4 and ADDRMAP8.addrmap_bg_b0==0 Otherwise, if MEMC_BURST_LENGTH=8 and Full Bus Width (MSTR.data_bus_width==00), it is recommended to program this to 0 so that HIF[2] maps to column address bit 2. If MEMC_BURST_LENGTH=16 and Full Bus Width (MSTR.data_bus_width==00), it is recommended to program this to 0 so that HIF[2] maps to column address bit 2. If MEMC_BURST_LENGTH=16 and Half Bus Width (MSTR.data_bus_width==01), it is recommended to program this to 0 so that HIF[2] maps to column address bit 3.

9.2.5.1.77 Address Map Register 3 (ADDRMAP3)

9.2.5.1.77.1 Offset

Register	Offset
ADDRMAP3	20Ch

9.2.5.1.77.2 Diagram



9.2.5.1.77.3 Fields

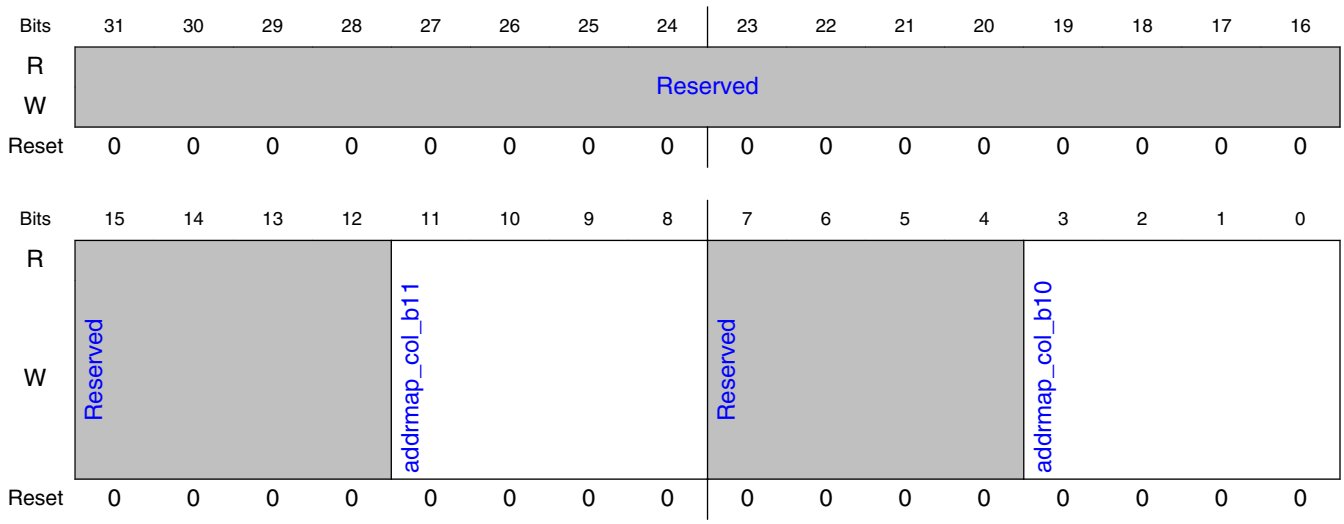
Field	Function
31-28 —	Reserved
27-24 addrmap_col_b9	- Full bus width mode: Selects the HIF address bit used as column address bit 9. - Half bus width mode: Selects the HIF address bit used as column address bit 11. - Quarter bus width mode: Selects the HIF address bit used as column address bit 13. Valid Range: 0 to 7, and 15 Internal Base: 9 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0.
23-20 —	Reserved
19-16 addrmap_col_b8	- Full bus width mode: Selects the HIF address bit used as column address bit 8. - Half bus width mode: Selects the HIF address bit used as column address bit 9. - Quarter bus width mode: Selects the HIF address bit used as column address bit 11. Valid Range: 0 to 7, and 15 Internal Base: 8 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0.
15-12 —	Reserved
11-8 addrmap_col_b7	- Full bus width mode: Selects the HIF address bit used as column address bit 7. - Half bus width mode: Selects the HIF address bit used as column address bit 8. - Quarter bus width mode: Selects the HIF address bit used as column address bit 9. Valid Range: 0 to 7, and 15 Internal Base: 7 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0.
7-4 —	Reserved
3-0 addrmap_col_b6	- Full bus width mode: Selects the HIF address bit used as column address bit 6. - Half bus width mode: Selects the HIF address bit used as column address bit 7. - Quarter bus width mode: Selects the HIF address bit used as column address bit 8. Valid Range: 0 to 7, and 15 Internal Base: 6 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0.

9.2.5.1.78 Address Map Register 4 (ADDRMAP4)

9.2.5.1.78.1 Offset

Register	Offset
ADDRMAP4	210h

9.2.5.1.78.2 Diagram



9.2.5.1.78.3 Fields

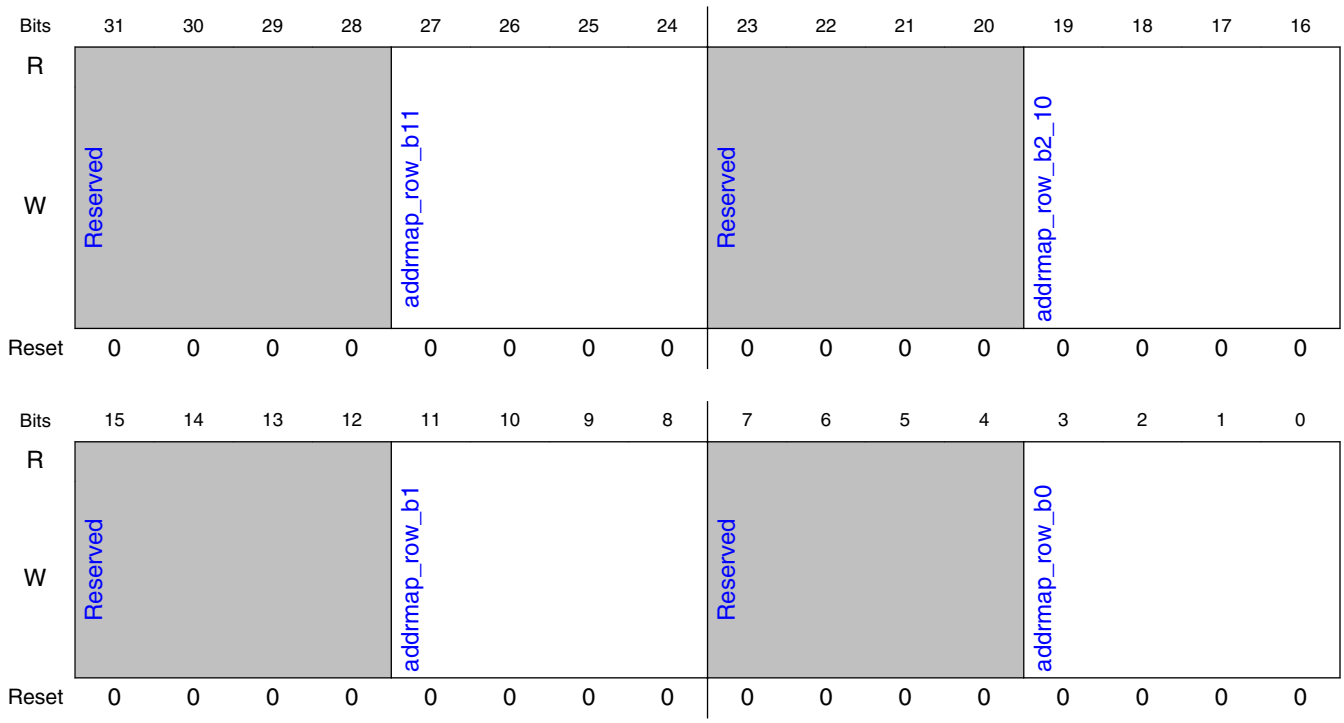
Field	Function
31-12 —	Reserved
11-8 addrmap_col_b11	- Full bus width mode: Selects the HIF address bit used as column address bit 13. - Half bus width mode: Unused. To make it unused, this should be tied to 4'hF. - Quarter bus width mode: Unused. To make it unused, this must be tied to 4'hF. Valid Range: 0 to 7, and 15 Internal Base: 11 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0.
7-4 —	Reserved
3-0 addrmap_col_b10	- Full bus width mode: Selects the HIF address bit used as column address bit 11. - Half bus width mode: Selects the HIF address bit used as column address bit 13. - Quarter bus width mode: UNUSED. To make it unused, this must be tied to 4'hF. Valid Range: 0 to 7, and 15 Internal Base: 10 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0.

9.2.5.1.79 Address Map Register 5 (ADDRMAP5)

9.2.5.1.79.1 Offset

Register	Offset
ADDRMAP5	214h

9.2.5.1.79.2 Diagram



9.2.5.1.79.3 Fields

Field	Function
31-28 —	Reserved
27-24 addrmap_row_b11	Selects the HIF address bit used as row address bit 11. Valid Range: 0 to 11, and 15 Internal Base: 17 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 11 is set to 0.
23-20 —	Reserved
19-16 addrmap_row_b2_10	Selects the HIF address bits used as row address bits 2 to 10. Valid Range: 0 to 11, and 15 Internal Base: 8 (for row address bit 2), 9 (for row address bit 3), 10 (for row address bit 4) etc increasing to 16 (for row address bit 10) The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. When value 15 is used the values of row address bits 2 to 10 are defined by registers ADDRMAP9, ADDRMAP10, ADDRMAP11.
15-12 —	Reserved
11-8 addrmap_row_b1	Selects the HIF address bits used as row address bit 1. Valid Range: 0 to 11 Internal Base: 7 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field.
7-4 —	Reserved

Table continues on the next page...

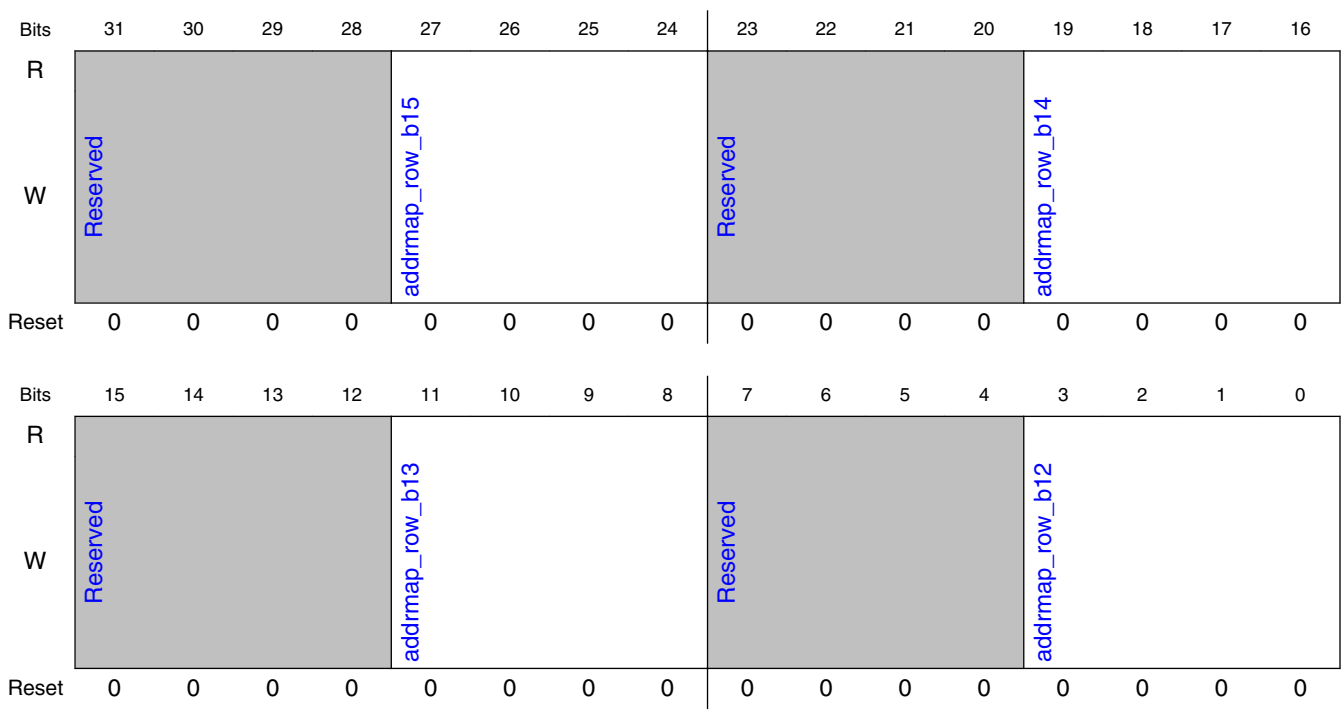
Field	Function
3-0 addrmap_row_b0	Selects the HIF address bits used as row address bit 0. Valid Range: 0 to 11 Internal Base: 6 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field.

9.2.5.1.80 Address Map Register 6 (ADDRMAP6)

9.2.5.1.80.1 Offset

Register	Offset
ADDRMAP6	218h

9.2.5.1.80.2 Diagram



9.2.5.1.80.3 Fields

Field	Function
31-28 —	Reserved

Table continues on the next page...

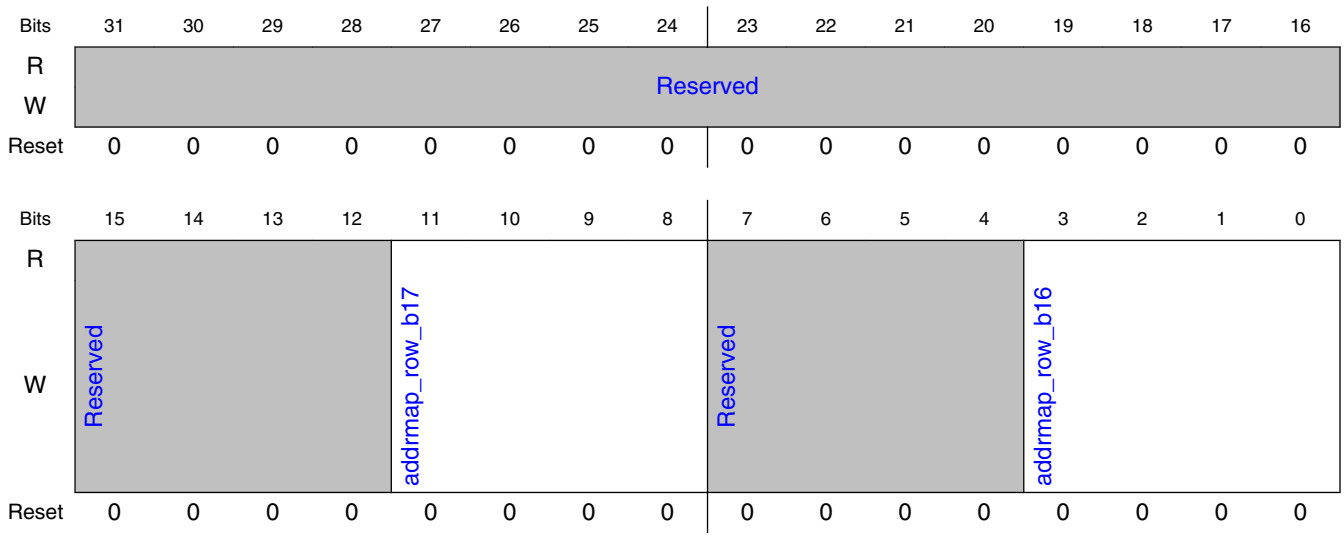
Field	Function
27-24 addrmap_row_b15	Selects the HIF address bit used as row address bit 15. Valid Range: 0 to 11, and 15 Internal Base: 21 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 15 is set to 0.
23-20 —	Reserved
19-16 addrmap_row_b14	Selects the HIF address bit used as row address bit 14. Valid Range: 0 to 11, and 15 Internal Base: 20 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 14 is set to 0.
15-12 —	Reserved
11-8 addrmap_row_b13	Selects the HIF address bit used as row address bit 13. Valid Range: 0 to 11, and 15 Internal Base: 19 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 13 is set to 0.
7-4 —	Reserved
3-0 addrmap_row_b12	Selects the HIF address bit used as row address bit 12. Valid Range: 0 to 11, and 15 Internal Base: 18 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 12 is set to 0.

9.2.5.1.81 Address Map Register 7 (ADDRMAP7)

9.2.5.1.81.1 Offset

Register	Offset
ADDRMAP7	21Ch

9.2.5.1.81.2 Diagram



9.2.5.1.81.3 Fields

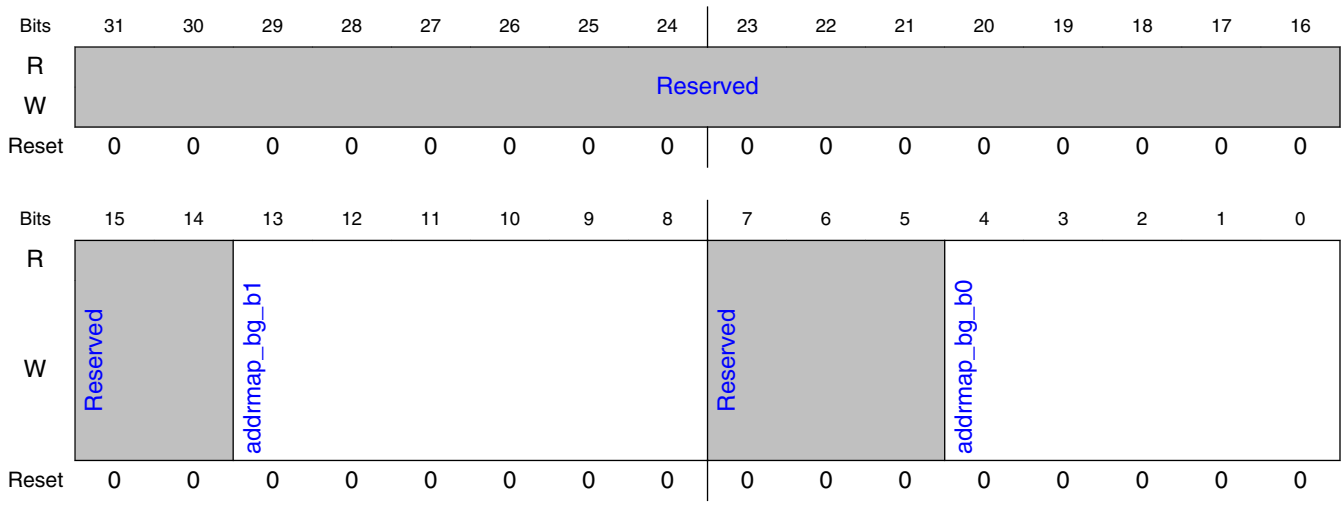
Field	Function
31-12 —	Reserved
11-8 addrmap_row_b17	Selects the HIF address bit used as row address bit 17. Valid Range: 0 to 11, and 15 Internal Base: 23 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 17 is set to 0.
7-4 —	Reserved
3-0 addrmap_row_b16	Selects the HIF address bit used as row address bit 16. Valid Range: 0 to 11, and 15 Internal Base: 22 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 16 is set to 0.

9.2.5.1.82 Address Map Register 8 (ADDRMAP8)

9.2.5.1.82.1 Offset

Register	Offset
ADDRMAP8	220h

9.2.5.1.82.2 Diagram



9.2.5.1.82.3 Fields

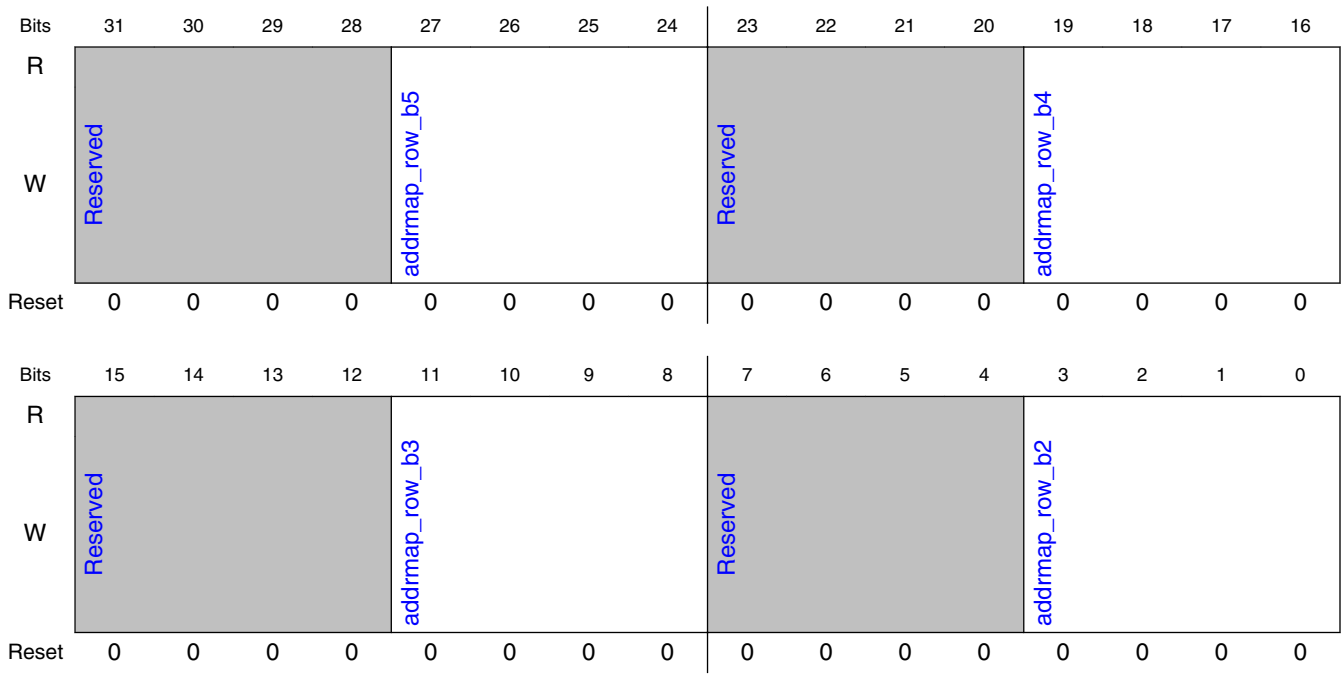
Field	Function
31-14 —	Reserved
13-8 addrmap_bg_b1	Selects the HIF address bits used as bank group address bit 1. Valid Range: 0 to 31, and 63 Internal Base: 3 The selected HIF address bit for each of the bank group address bits is determined by adding the internal base to the value of this field. If set to 63, bank group address bit 1 is set to 0.
7-5 —	Reserved
4-0 addrmap_bg_b0	Selects the HIF address bits used as bank group address bit 0. Valid Range: 0 to 31 Internal Base: 2 The selected HIF address bit for each of the bank group address bits is determined by adding the internal base to the value of this field.

9.2.5.1.83 Address Map Register 9 (ADDRMAP9)

9.2.5.1.83.1 Offset

Register	Offset
ADDRMAP9	224h

9.2.5.1.83.2 Diagram



9.2.5.1.83.3 Fields

Field	Function
31-28 —	Reserved
27-24 addrmap_row_b5	Selects the HIF address bits used as row address bit 5. Valid Range: 0 to 11 Internal Base: 11 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
23-20 —	Reserved
19-16 addrmap_row_b4	Selects the HIF address bits used as row address bit 4. Valid Range: 0 to 11 Internal Base: 10 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
15-12 —	Reserved
11-8 addrmap_row_b3	Selects the HIF address bits used as row address bit 3. Valid Range: 0 to 11 Internal Base: 9 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
7-4 —	Reserved

Table continues on the next page...

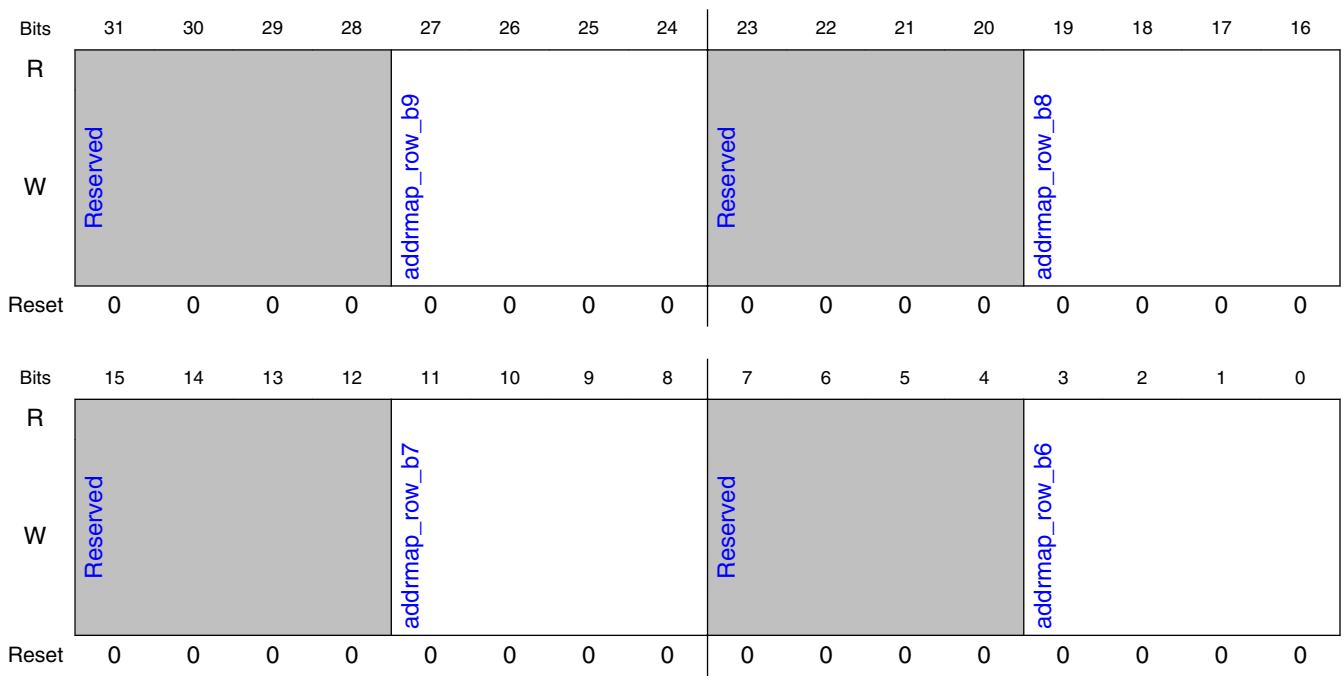
Field	Function
3-0 addrmap_row_b2	Selects the HIF address bits used as row address bit 2. Valid Range: 0 to 11 Internal Base: 8 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.

9.2.5.1.84 Address Map Register 10 (ADDRMAP10)

9.2.5.1.84.1 Offset

Register	Offset
ADDRMAP10	228h

9.2.5.1.84.2 Diagram



9.2.5.1.84.3 Fields

Field	Function
31-28 —	Reserved

Table continues on the next page...

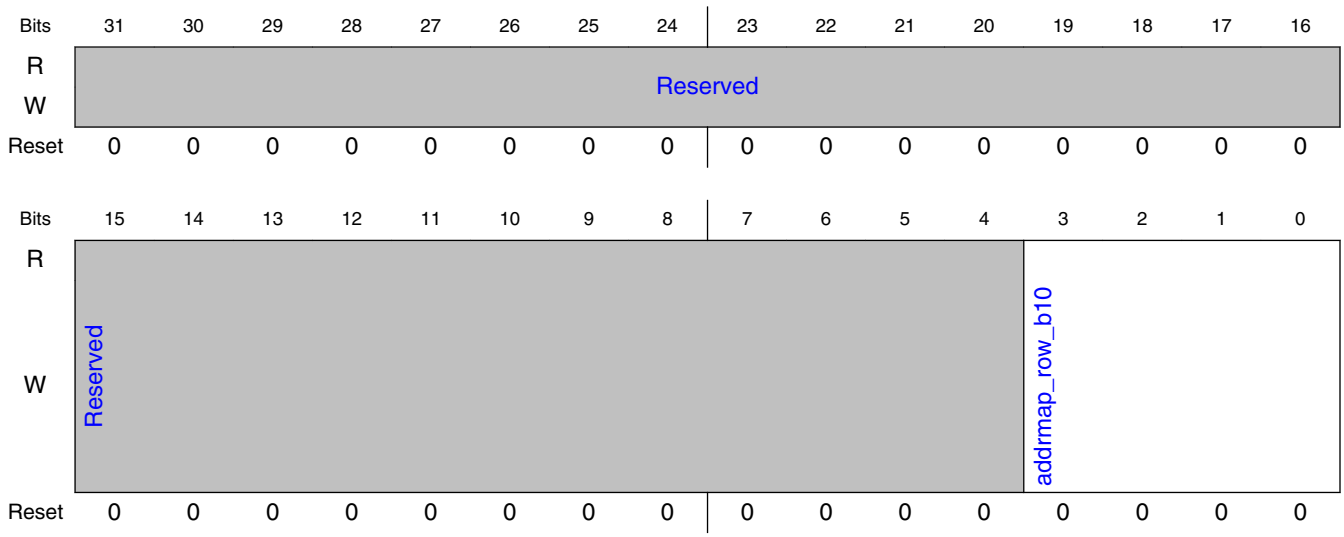
Field	Function
27-24 addrmap_row_b9	Selects the HIF address bits used as row address bit 9. Valid Range: 0 to 11 Internal Base: 15 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
23-20 —	Reserved
19-16 addrmap_row_b8	Selects the HIF address bits used as row address bit 8. Valid Range: 0 to 11 Internal Base: 14 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
15-12 —	Reserved
11-8 addrmap_row_b7	Selects the HIF address bits used as row address bit 7. Valid Range: 0 to 11 Internal Base: 13 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
7-4 —	Reserved
3-0 addrmap_row_b6	Selects the HIF address bits used as row address bit 6. Valid Range: 0 to 11 Internal Base: 12 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.

9.2.5.1.85 Address Map Register 11 (ADDRMAP11)

9.2.5.1.85.1 Offset

Register	Offset
ADDRMAP11	22Ch

9.2.5.1.85.2 Diagram



9.2.5.1.85.3 Fields

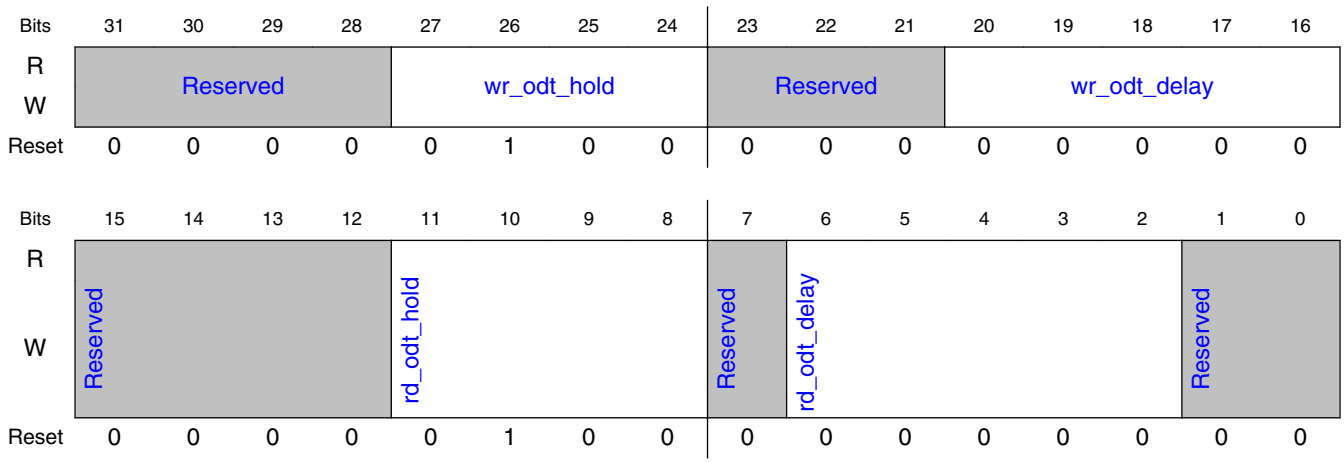
Field	Function
31-4 —	Reserved
3-0 addrmap_row_b10	Selects the HIF address bits used as row address bit 10. Valid Range: 0 to 11 Internal Base: 16 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.

9.2.5.1.86 ODT Configuration Register (ODTCFG)

9.2.5.1.86.1 Offset

Register	Offset
ODTCFG	240h

9.2.5.1.86.2 Diagram



9.2.5.1.86.3 Fields

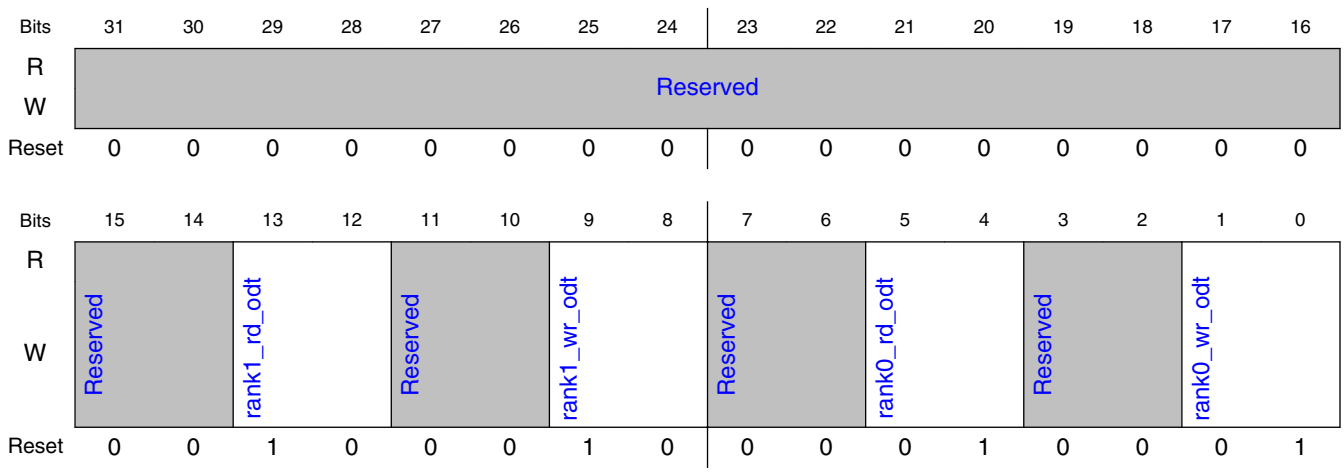
Field	Function
31-28 —	Reserved
27-24 wr_odt_hold	DFI PHY clock cycles to hold ODT for a write command. The minimum supported value is 2. Recommended values: DDR4: - BL8: 5 + WR_PREAMBLE + CRC_MODE WR_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble) CRC_MODE = 0 (not CRC mode), 1 (CRC mode).
23-21 —	Reserved
20-16 wr_odt_delay	The delay, in DFI PHY clock cycles, from issuing a write command to setting ODT values associated with that command. ODT setting must remain constant for the entire time that DQS is driven by the DDRC. Recommended values: DDR4: - DFITMG1.dfi_t_cmd_lat (to adjust for CAL mode).
15-12 —	Reserved
11-8 rd_odt_hold	DFI PHY clock cycles to hold ODT for a read command. The minimum supported value is 2. Recommended values: DDR4: - BL8: 5 + RD_PREAMBLE RD_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble).
7 —	Reserved
6-2 rd_odt_delay	The delay, in DFI PHY clock cycles, from issuing a read command to setting ODT values associated with that command. ODT setting must remain constant for the entire time that DQS is driven by the DDRC. Recommended values: DDR4: - CL - CWL - RD_PREAMBLE + WR_PREAMBLE + DFITMG1.dfi_t_cmd_lat (to adjust for CAL mode) WR_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble) RD_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble). If (CL - CWL - RD_PREAMBLE + WR_PREAMBLE) < 0, DDRC does not support ODT for read operation.
1-0 —	Reserved

9.2.5.1.87 ODT/Rank Map Register (ODTMAP)

9.2.5.1.87.1 Offset

Register	Offset
ODTMAP	244h

9.2.5.1.87.2 Diagram



9.2.5.1.87.3 Fields

Field	Function
31-14 —	Reserved
13-12 rank1_rd_odt	Indicates which remote ODTs must be turned on during a read from rank 1. Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here. Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, etc. For each rank, set its bit to 1 to enable its ODT. Present only in configurations that have 2 or more ranks
11-10 —	Reserved
9-8 rank1_wr_odt	Indicates which remote ODTs must be turned on during a write to rank 1. Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here. Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, etc. For each rank, set its bit to 1 to enable its ODT. Present only in configurations that have 2 or more ranks
7-6 —	Reserved
5-4 rank0_rd_odt	Indicates which remote ODTs must be turned on during a read from rank 0. Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here. Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, etc. For each rank, set its bit to 1 to enable its ODT.

Table continues on the next page...

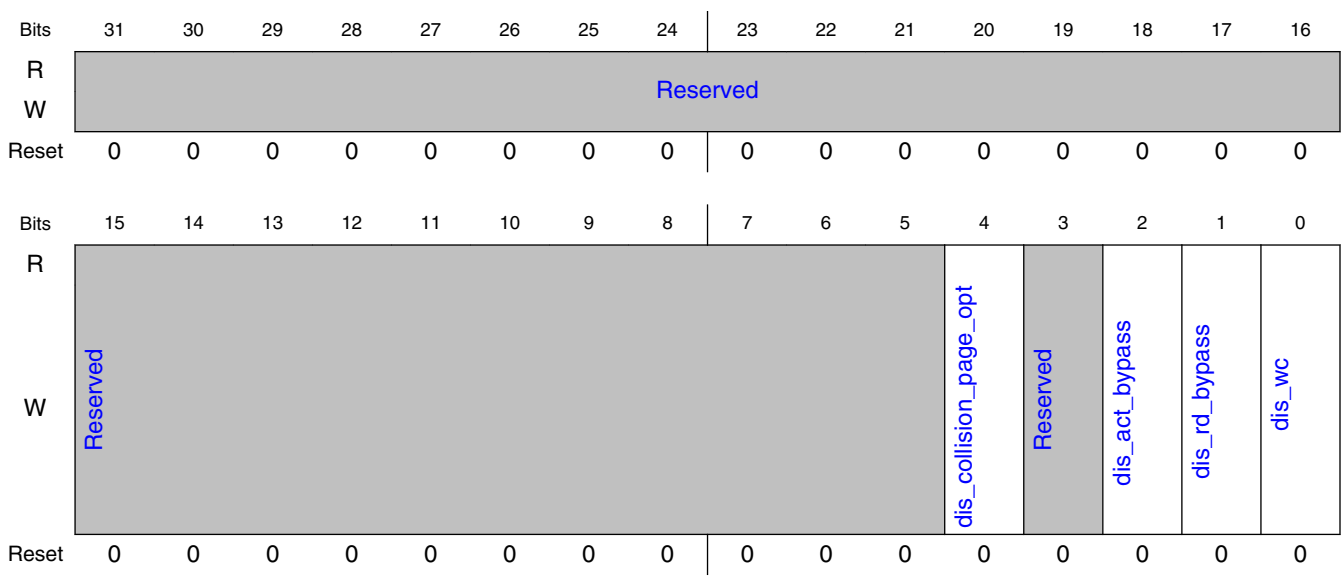
Field	Function
3-2 —	Reserved
1-0 rank0_wr_odt	Indicates which remote ODTs must be turned on during a write to rank 0. Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here. Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, etc. For each rank, set its bit to 1 to enable its ODT.

9.2.5.1.88 Debug Register 0 (DBG0)

9.2.5.1.88.1 Offset

Register	Offset
DBG0	300h

9.2.5.1.88.2 Diagram



9.2.5.1.88.3 Fields

Field	Function
31-5 —	Reserved
4	When this is set to '0', auto-precharge is disabled for the flushed command in a collision case. Collision cases are write followed by read to same address, read followed by write to same address, or write

Table continues on the next page...

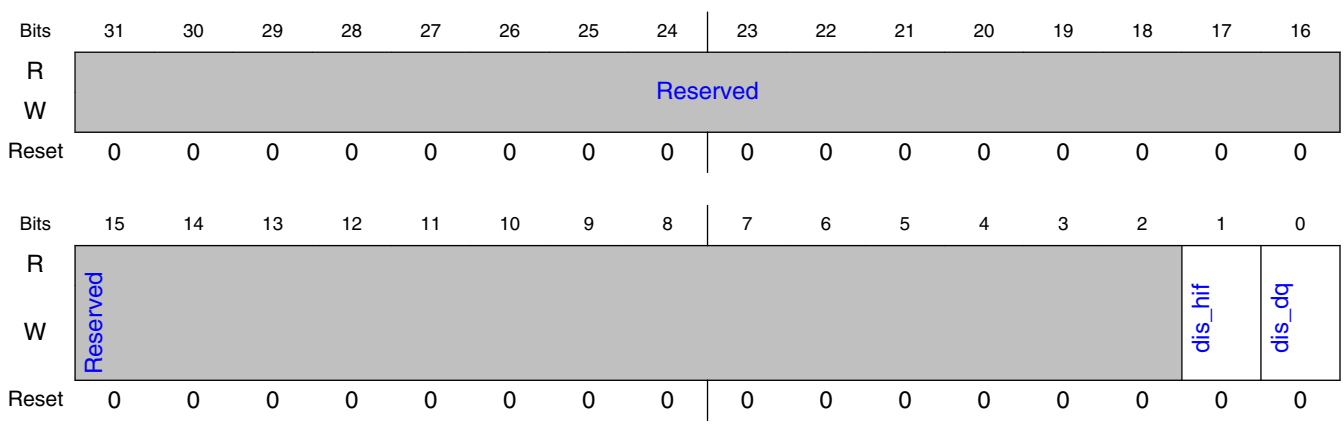
Field	Function
dis_collision_page_opt	followed by write to same address with DBG0.dis_wc bit = 1 (where same address comparisons exclude the two address bits representing critical word). FOR DEBUG ONLY.
3 —	Reserved
2 dis_act_bypass	Only present in designs supporting activate bypass. When 1, disable bypass path for high priority read activates FOR DEBUG ONLY.
1 dis_rd_bypass	Only present in designs supporting read bypass. When 1, disable bypass path for high priority read page hits FOR DEBUG ONLY.
0 dis_wc	When 1, disable write combine. FOR DEBUG ONLY

9.2.5.1.89 Debug Register 1 (DBG1)

9.2.5.1.89.1 Offset

Register	Offset
DBG1	304h

9.2.5.1.89.2 Diagram



9.2.5.1.89.3 Fields

Field	Function
31-2 —	Reserved

Table continues on the next page...

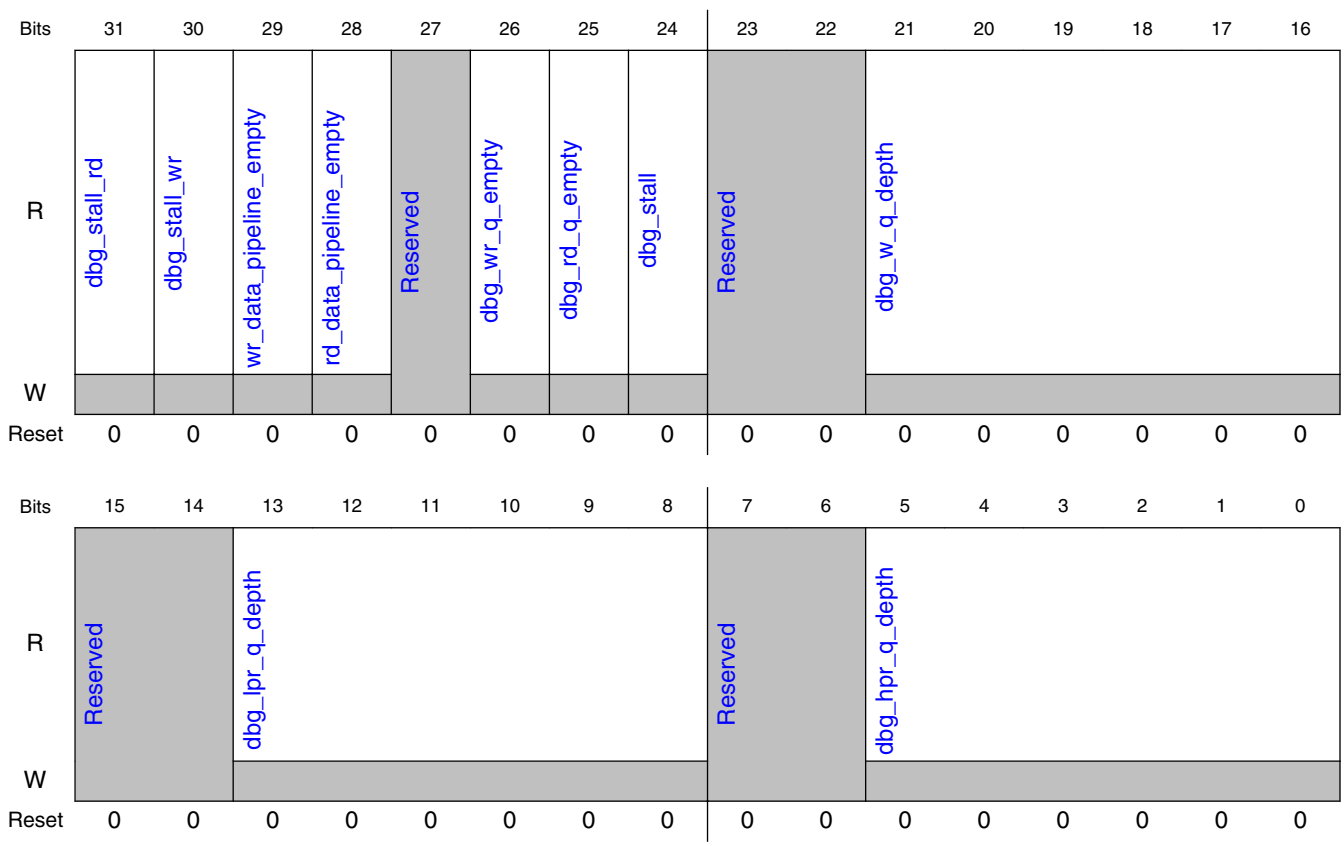
Field	Function
1 dis_hif	When 1, DDRC asserts the HIF command signal hif_cmd_stall. DDRC will ignore the hif_cmd_valid and all other associated request signals. This bit is intended to be switched on-the-fly.
0 dis_dq	When 1, DDRC will not de-queue any transactions from the CAM. Bypass is also disabled. All transactions are queued in the CAM. No reads or writes are issued to SDRAM as long as this is asserted. This bit may be used to prevent reads or writes being issued by the DDRC, which makes it safe to modify certain register fields associated with reads and writes (see User Guide for details). After setting this bit, it is strongly recommended to poll DBGCAM.wr_data_pipeline_empty and DBGCAM.rd_data_pipeline_empty, before making changes to any registers which affect reads and writes. This will ensure that the relevant logic in the DDRC is idle. This bit is intended to be switched on-the-fly.

9.2.5.1.90 CAM Debug Register (DBGCAM)

9.2.5.1.90.1 Offset

Register	Offset
DBGCAM	308h

9.2.5.1.90.2 Diagram



9.2.5.1.90.3 Fields

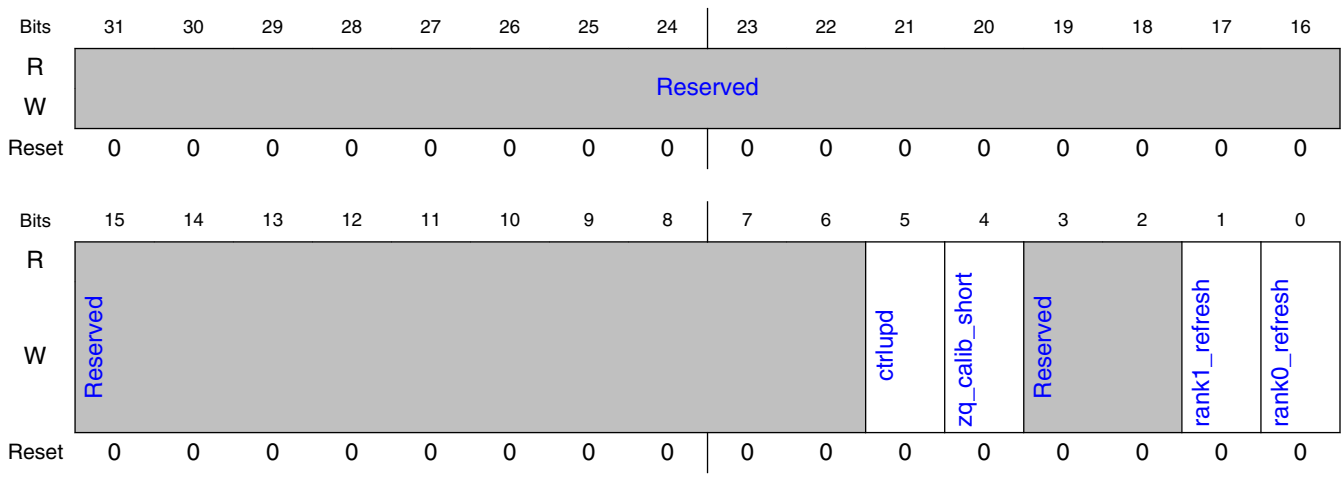
Field	Function
31 dbg_stall_rd	Stall for Read channel FOR DEBUG ONLY
30 dbg_stall_wr	Stall for Write channel FOR DEBUG ONLY
29 wr_data_pipeline_empty	This bit indicates that the write data pipeline on the DFI interface is empty. This register is intended to be polled at least twice after setting DBG1.dis_dq, to ensure that all remaining commands/data have completed.
28 rd_data_pipeline_empty	This bit indicates that the read data pipeline on the DFI interface is empty. This register is intended to be polled at least twice after setting DBG1.dis_dq, to ensure that all remaining commands/data have completed.
27 —	Reserved
26 dbg_wr_q_empty	When 1, all the Write command queues and Write data buffers inside DDRC are empty. This register is to be used for debug purpose. An example use-case scenario: When Controller enters Self-Refresh using the Low-Power entry sequence, Controller is expected to have executed all the commands in its queues and the write and read data drained. Hence this register should be 1 at that time. FOR DEBUG ONLY
25 dbg_rd_q_empty	When 1, all the Read command queues and Read data buffers inside DDRC are empty. This register is to be used for debug purpose. An example use-case scenario: When Controller enters Self-Refresh using the Low-Power entry sequence, Controller is expected to have executed all the commands in its queues and the write and read data drained. Hence this register should be 1 at that time. FOR DEBUG ONLY
24 dbg_stall	Stall FOR DEBUG ONLY
23-22 —	Reserved
21-16 dbg_w_q_depth	Write queue depth The last entry of WR queue is reserved for ECC SCRUB operation. This entry is not included in the calculation of the queue depth. FOR DEBUG ONLY
15-14 —	Reserved
13-8 dbg_lpr_q_depth	Low priority read queue depth The last entry of Lpr queue is reserved for ECC SCRUB operation. This entry is not included in the calculation of the queue depth. FOR DEBUG ONLY
7-6 —	Reserved
5-0 dbg_hpr_q_depth	High priority read queue depth FOR DEBUG ONLY

9.2.5.1.91 Command Debug Register (DBGCMD)

9.2.5.1.91.1 Offset

Register	Offset
DBGCMD	30Ch

9.2.5.1.91.2 Diagram



9.2.5.1.91.3 Fields

Field	Function
31-6 —	Reserved
5 ctrlupd	Setting this register bit to 1 indicates to the DDRC to issue a dfi_ctrlupd_req to the PHY. When this request is stored in the DDRC, the bit is automatically cleared. This operation must only be performed when DFIUPD0.dis_auto_ctrlupd=1.
4 zq_calib_short	Setting this register bit to 1 indicates to the DDRC to issue a ZQCS (ZQ calibration short)/MPC(ZQ calibration) command to the SDRAM. When this request is stored in the DDRC, the bit is automatically cleared. This operation can be performed only when ZQCTL0.dis_auto_zq=1. It is recommended NOT to set this register bit if in Init operating mode. This register bit is ignored when in Self-Refresh(except LPDDR4) and SR-Powerdown(LPDDR4) and Deep power-down operating modes and Maximum Power Saving Mode.
3-2 —	Reserved
1 rank1_refresh	Setting this register bit to 1 indicates to the DDRC to issue a refresh to rank 1. Writing to this bit causes DBGSTAT.rank1_refresh_busy to be set. When DBGSTAT.rank1_refresh_busy is cleared, the command has been stored in DDRC. For 3DS configuration, refresh is sent to rank index 1. This operation can be performed only when RFSHCTL3.dis_auto_refresh=1. It is recommended NOT to set this register bit if in Init or Deep power-down operating modes or Maximum Power Saving Mode.

Table continues on the next page...

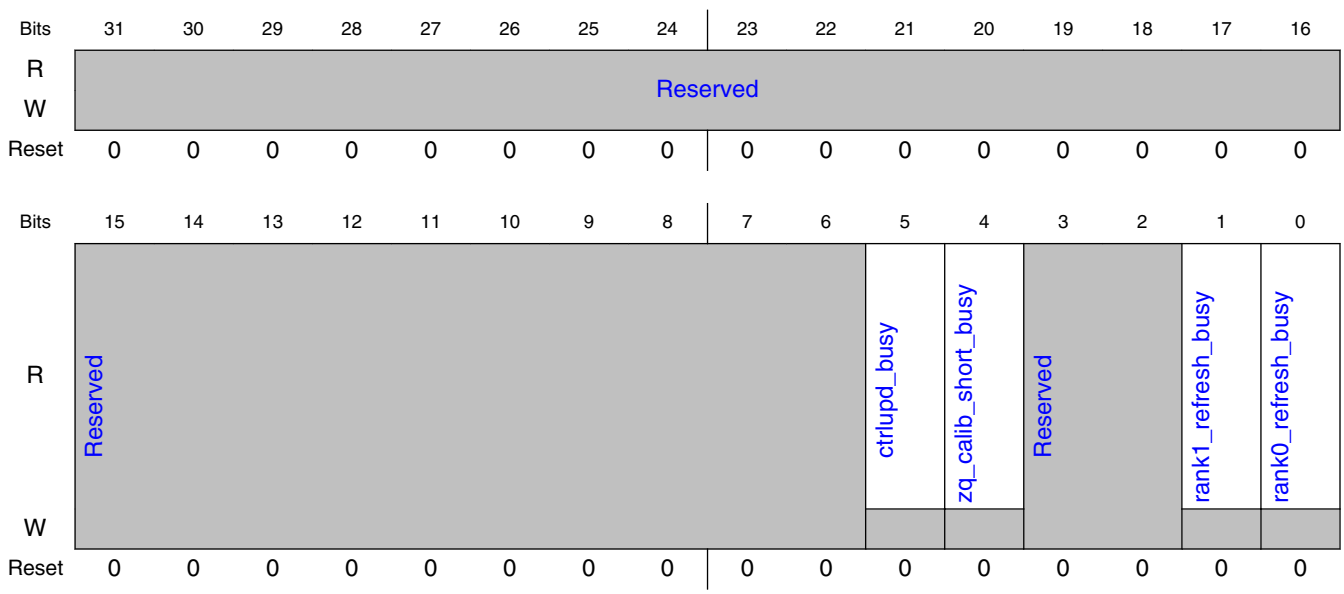
Field	Function
0 rank0_refresh	Setting this register bit to 1 indicates to the DDRC to issue a refresh to rank 0. Writing to this bit causes DBGSTAT.rank0_refresh_busy to be set. When DBGSTAT.rank0_refresh_busy is cleared, the command has been stored in DDRC. For 3DS configuration, refresh is sent to rank index 0. This operation can be performed only when RFSHCTL3.dis_auto_refresh=1. It is recommended NOT to set this register bit if in Init or Deep power-down operating modes or Maximum Power Saving Mode.

9.2.5.1.92 Status Debug Register (DBGSTAT)

9.2.5.1.92.1 Offset

Register	Offset
DBGSTAT	310h

9.2.5.1.92.2 Diagram



9.2.5.1.92.3 Fields

Field	Function
31-6 —	Reserved
5 ctrlupd_busy	SoC core may initiate a ctrlupd operation only if this signal is low. This signal goes high in the clock after the DDRC accepts the ctrlupd request. It goes low when the ctrlupd operation is initiated in the DDRC. It is recommended not to perform ctrlupd operations when this signal is high. - 0 - Indicates that the SoC

Table continues on the next page...

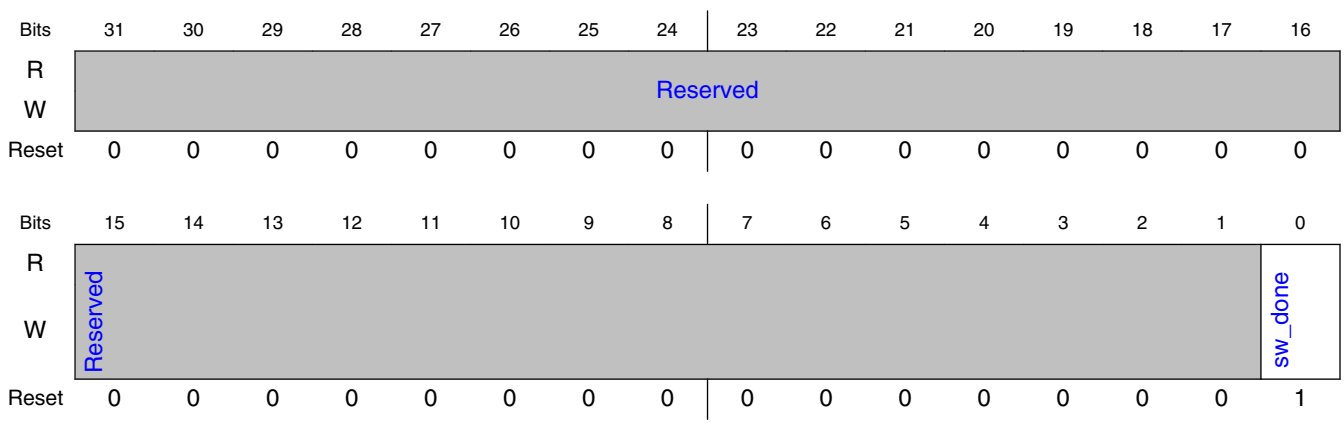
Field	Function
	core can initiate a ctrlupd operation - 1 - Indicates that ctrlupd operation has not been initiated yet in the DDRC
4 zq_calib_short_busy	SoC core may initiate a ZQCS (ZQ calibration short) operation only if this signal is low. This signal goes high in the clock after the DDRC accepts the ZQCS request. It goes low when the ZQCS operation is initiated in the DDRC. It is recommended not to perform ZQCS operations when this signal is high. - 0 - Indicates that the SoC core can initiate a ZQCS operation - 1 - Indicates that ZQCS operation has not been initiated yet in the DDRC
3-2 —	Reserved
1 rank1_refresh_busy	SoC core may initiate a rank1_refresh operation (refresh operation to rank 1) only if this signal is low. This signal goes high in the clock after DBGCMD.rank1_refresh is set to one. It goes low when the rank1_refresh operation is stored in the DDRC. It is recommended not to perform rank1_refresh operations when this signal is high. - 0 - Indicates that the SoC core can initiate a rank1_refresh operation - 1 - Indicates that rank1_refresh operation has not been stored yet in the DDRC
0 rank0_refresh_busy	SoC core may initiate a rank0_refresh operation (refresh operation to rank 0) only if this signal is low. This signal goes high in the clock after DBGCMD.rank0_refresh is set to one. It goes low when the rank0_refresh operation is stored in the DDRC. It is recommended not to perform rank0_refresh operations when this signal is high. - 0 - Indicates that the SoC core can initiate a rank0_refresh operation - 1 - Indicates that rank0_refresh operation has not been stored yet in the DDRC

9.2.5.1.93 Software Register Programming Control Enable (SWCTL)

9.2.5.1.93.1 Offset

Register	Offset
SWCTL	320h

9.2.5.1.93.2 Diagram



9.2.5.1.93.3 Fields

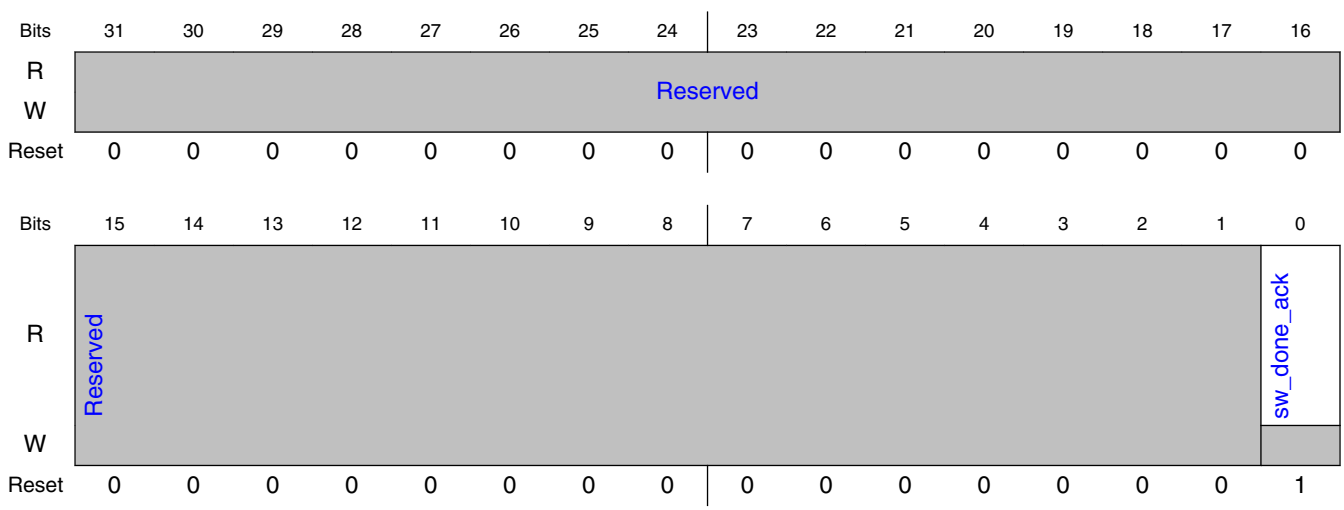
Field	Function
31-1 —	Reserved
0 sw_done	Enable quasi-dynamic register programming outside reset. Program register to 0 to enable quasi-dynamic programming. Set back register to 1 once programming is done.

9.2.5.1.94 Software Register Programming Control Status (SWSTAT)

9.2.5.1.94.1 Offset

Register	Offset
SWSTAT	324h

9.2.5.1.94.2 Diagram



9.2.5.1.94.3 Fields

Field	Function
31-1 —	Reserved
0 sw_done_ack	Register programming done. This register is the echo of SWCTL.sw_done. Wait for sw_done value 1 to propagate to sw_done_ack at the end of the programming sequence to ensure that the correct registers values are propagated to the destination clock domains.

9.2.5.1.95 AXI Poison Configuration Register. (POISONCFG)

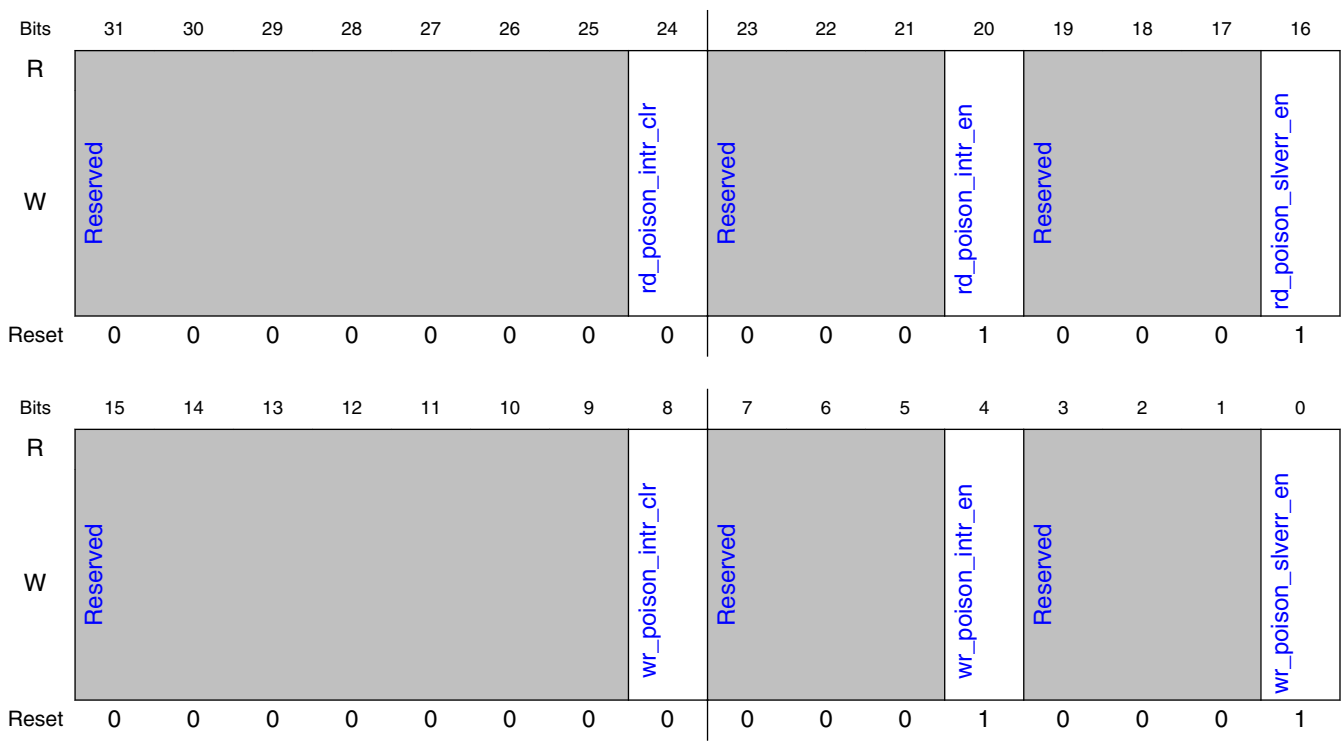
9.2.5.1.95.1 Offset

Register	Offset
POISONCFG	36Ch

9.2.5.1.95.2 Function

AXI Poison Configuration Register. Common for all AXI ports

9.2.5.1.95.3 Diagram



9.2.5.1.95.4 Fields

Field	Function
31-25 —	Reserved
24	Interrupt clear for read transaction poisoning. Allow 2/3 clock cycles for correct value to propagate to core logic and clear the interrupts.

Table continues on the next page...

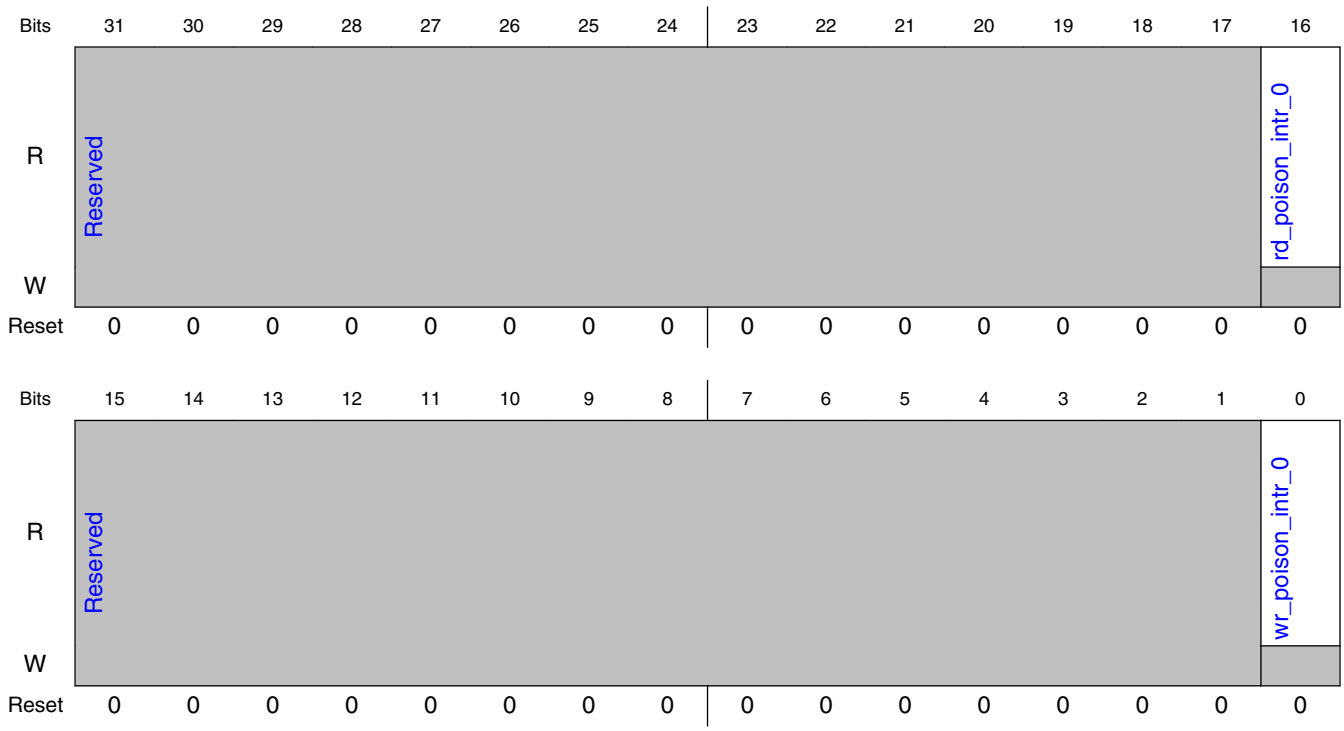
Field	Function
rd_poison_intr_clr	
23-21 —	Reserved
20 rd_poison_intr_en	If set to 1, enables interrupts for read transaction poisoning
19-17 —	Reserved
16 rd_poison_slverr_en	If set to 1, enables SLVERR response for read transaction poisoning
15-9 —	Reserved
8 wr_poison_intr_clr	Interrupt clear for write transaction poisoning. Allow 2/3 clock cycles for correct value to propagate to core logic and clear the interrupts.
7-5 —	Reserved
4 wr_poison_intr_en	If set to 1, enables interrupts for write transaction poisoning
3-1 —	Reserved
0 wr_poison_slverr_en	If set to 1, enables SLVERR response for write transaction poisoning

9.2.5.1.96 AXI Poison Status Register (POISONSTAT)

9.2.5.1.96.1 Offset

Register	Offset
POISONSTAT	370h

9.2.5.1.96.2 Diagram



9.2.5.1.96.3 Fields

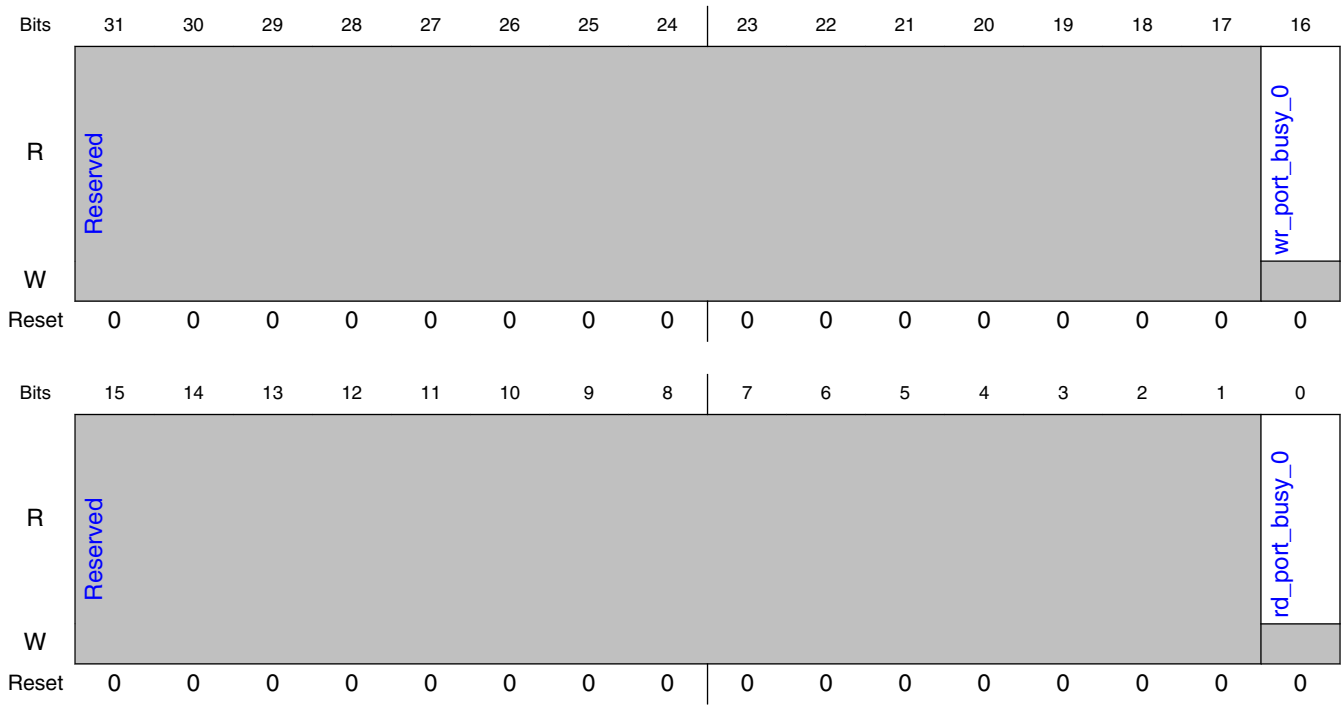
Field	Function
31-17 —	Reserved
16 rd_poison_intr_0	Read transaction poisoning error interrupt for port 0. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock.
15-1 —	Reserved
0 wr_poison_intr_0	Write transaction poisoning error interrupt for port 0. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock.

9.2.5.1.97 Port Status Register (PSTAT)

9.2.5.1.97.1 Offset

Register	Offset
PSTAT	3FCh

9.2.5.1.97.2 Diagram



9.2.5.1.97.3 Fields

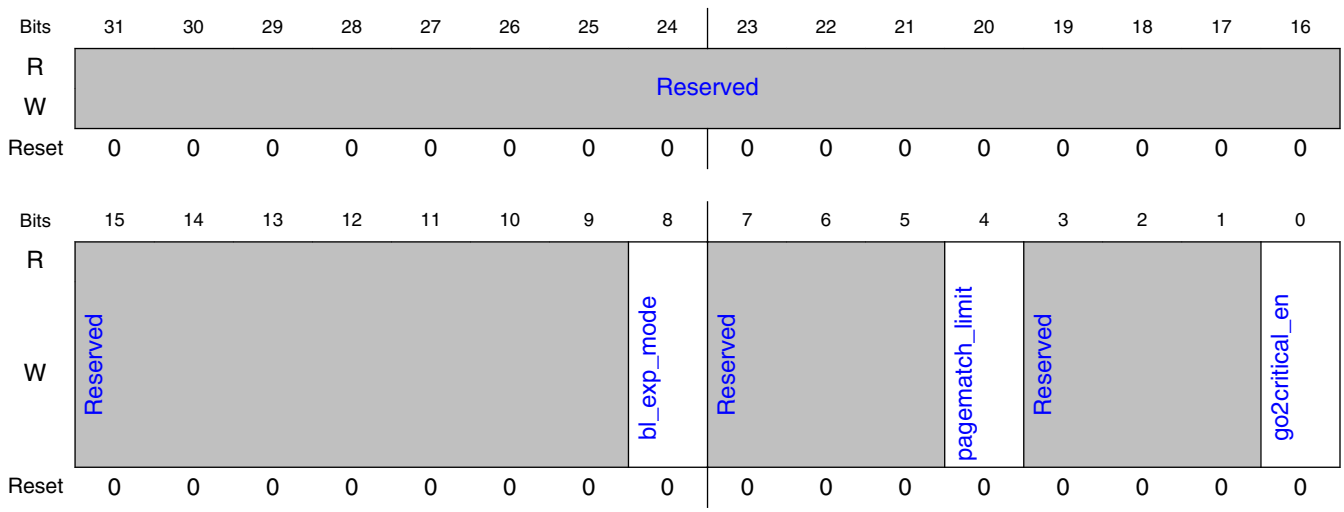
Field	Function
31-17 —	Reserved
16 wr_port_busy_0	Indicates if there are outstanding writes for AXI port 0.
15-1 —	Reserved
0 rd_port_busy_0	Indicates if there are outstanding reads for AXI port 0.

9.2.5.1.98 Port Common Configuration Register (PCCFG)

9.2.5.1.98.1 Offset

Register	Offset
PCCFG	400h

9.2.5.1.98.2 Diagram



9.2.5.1.98.3 Fields

Field	Function
31-9 —	Reserved
8 bl_exp_mode	Burst length expansion mode. By default (i.e. bl_exp_mode==0) XPI expands every AXI burst into multiple HIF commands, using the memory burst length as a unit. If set to 1, then XPI will use half of the memory burst length as a unit. This applies to both reads and writes. When MSTR.data_bus_width==00, setting bl_exp_mode to 1 has no effect. This can be used in cases where Partial Writes is enabled (DDRC_PARTIAL_WR=1), in order to avoid or minimize t_ccd_l penalty in DDR4 and t_ccd_mw penalty in LPDDR4. Hence, bl_exp_mode=1 is only recommended if DDR4 or LPDDR4. Note that if DBICTL.reg_ddrc_dm_en=0, functionality is not supported in the following cases: - DDRC_PARTIAL_WR=0 - DDRC_PARTIAL_WR=1, MSTR.data_bus_width=01, MEMC_BURST_LENGTH=8 and MSTR.burst_rdwr=1000 (LPDDR4 only) - DDRC_PARTIAL_WR=1, MSTR.data_bus_width=01, MEMC_BURST_LENGTH=4 and MSTR.burst_rdwr=0100 (DDR4 only), with either MSTR.reg_ddrc_burstchop=0 or CRCPARCTL1.reg_ddrc_crc_enable=1 Functionality is also not supported if Data Channel Interleave is enabled
7-5 —	Reserved

Table continues on the next page...

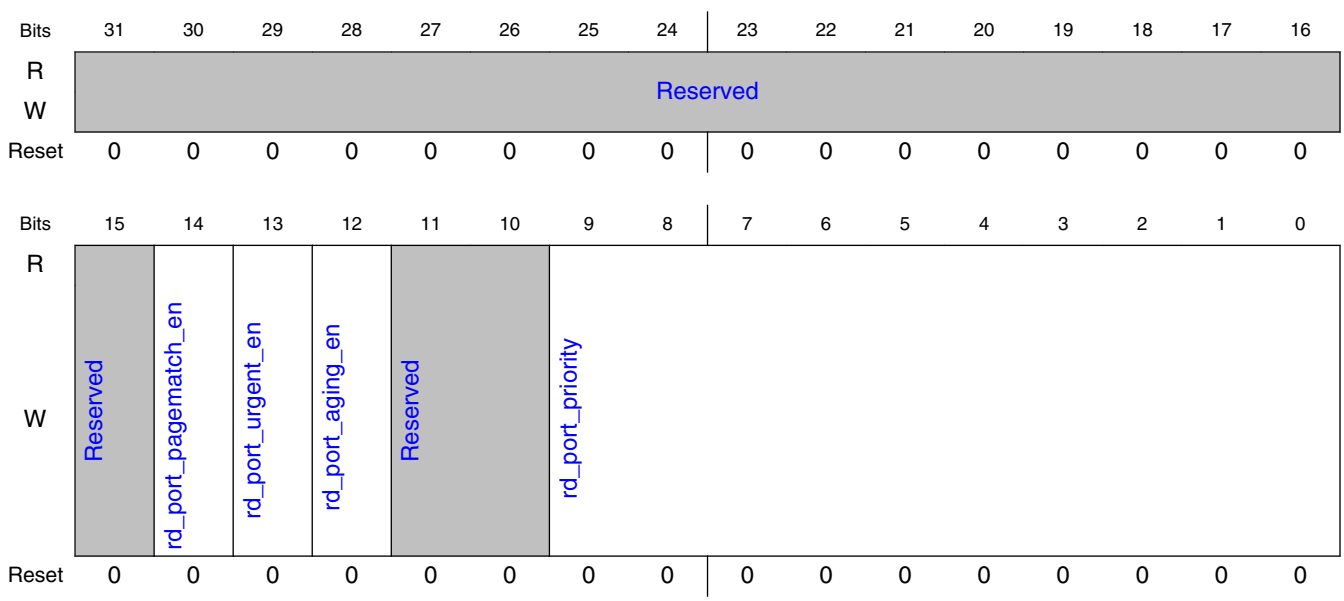
Field	Function
4 pagematch_limit	Page match four limit. If set to 1, limits the number of consecutive same page DDRC transactions that can be granted by the Port Arbiter to four when Page Match feature is enabled. If set to 0, there is no limit imposed on number of consecutive same page DDRC transactions.
3-1 —	Reserved
0 go2critical_en	If set to 1 (enabled), sets co_gs_go2critical_wr and co_gs_go2critical_lpr/co_gs_go2critical_hpr signals going to DDRC based on urgent input (awurgent, arurgent) coming from AXI master. If set to 0 (disabled), co_gs_go2critical_wr and co_gs_go2critical_lpr/co_gs_go2critical_hpr signals at DDRC are driven to 1b'0.

9.2.5.1.99 Port n Configuration Read Register (PCFGR_0)

9.2.5.1.99.1 Offset

Register	Offset
PCFGR_0	404h

9.2.5.1.99.2 Diagram



9.2.5.1.99.3 Fields

Field	Function
31-15	Reserved

Table continues on the next page...

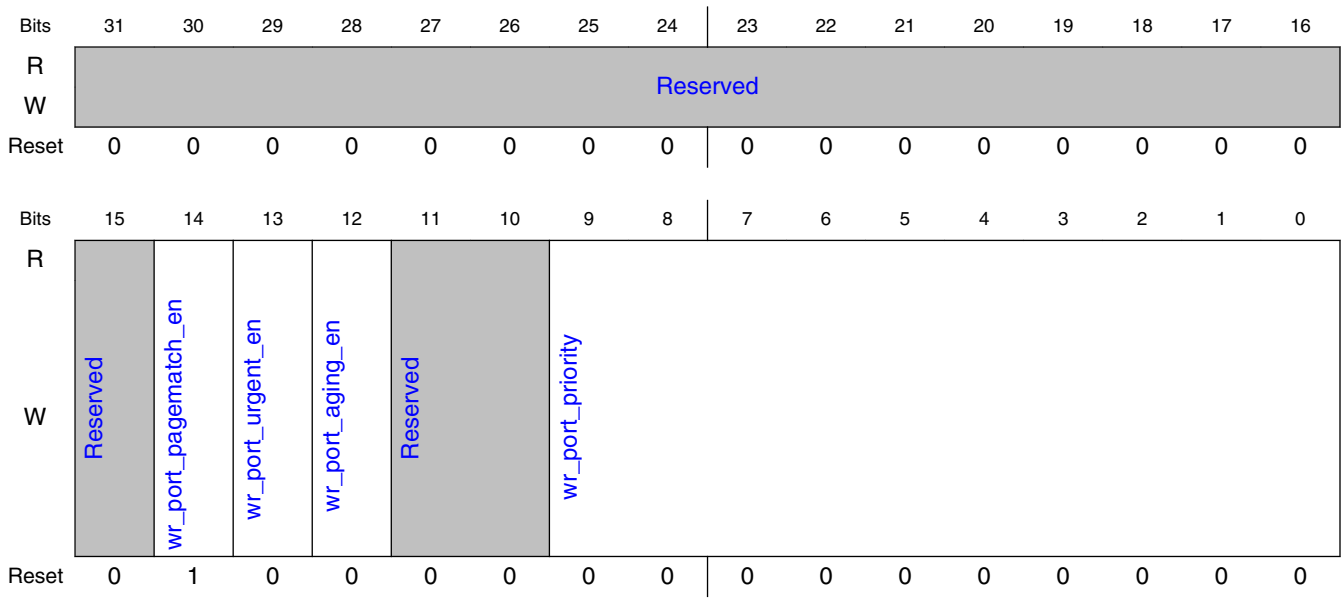
Field	Function
—	
14 rd_port_pagematch_en	If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also related PCCFG.pagematch_limit register.
13 rd_port_urgent_en	If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_lpr/co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
12 rd_port_aging_en	If set to 1, enables aging function for the read channel of the port.
11-10 —	Reserved
9-0 rd_port_priority	Determines the initial load value of read aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis. Note: The two LSBs of this register field are tied internally to 2'b00.

9.2.5.1.100 Port n Configuration Write Register (PCFGW_0)

9.2.5.1.100.1 Offset

Register	Offset
PCFGW_0	408h

9.2.5.1.100.2 Diagram



9.2.5.1.100.3 Fields

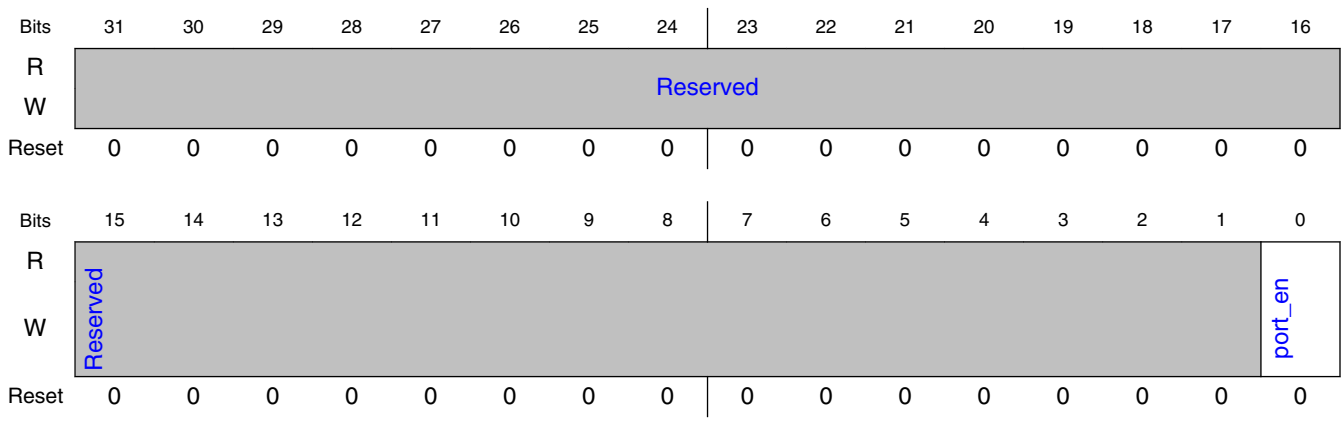
Field	Function
31-15 —	Reserved
14 wr_port_pagematch_en	If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also related PCCFG.pagematch_limit register.
13 wr_port_urgent_en	If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
12 wr_port_aging_en	If set to 1, enables aging function for the write channel of the port.
11-10 —	Reserved
9-0 wr_port_priority	Determines the initial load value of write aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the write channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level. For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. Note: The two LSBs of this register field are tied internally to 2'b00.

9.2.5.1.101 Port n Control Register (PCTRL_0)

9.2.5.1.101.1 Offset

Register	Offset
PCTRL_0	490h

9.2.5.1.101.2 Diagram



9.2.5.1.101.3 Fields

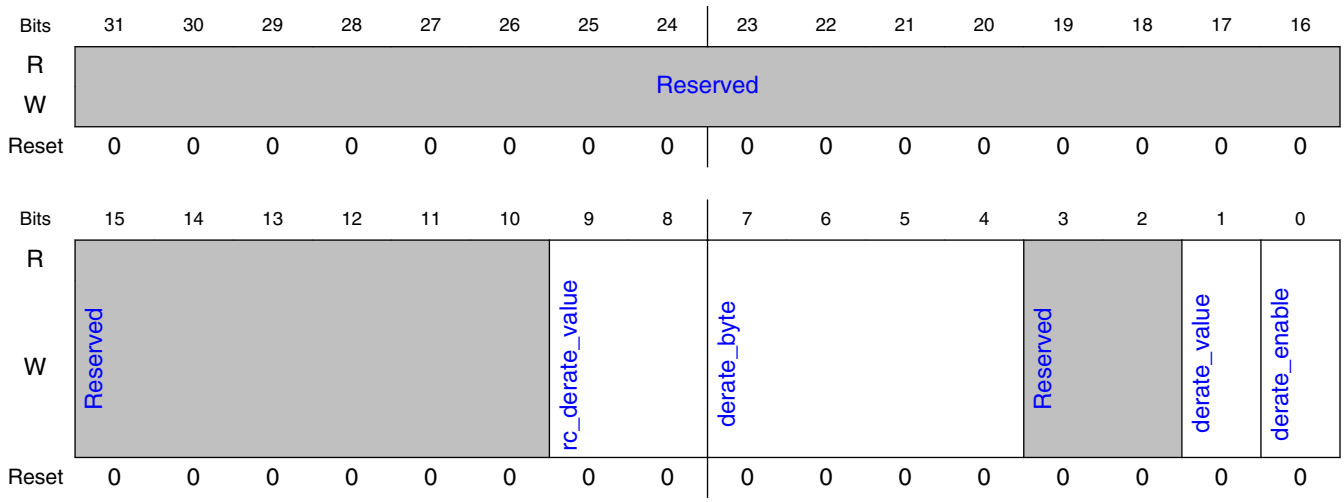
Field	Function
31-1 —	Reserved
0 port_en	Enables AXI port n.

9.2.5.1.102 [SHADOW] Temperature Derate Enable Register (DERATEEN_SHADOW)

9.2.5.1.102.1 Offset

Register	Offset
DERATEEN_SHADOW	2020h

9.2.5.1.102.2 Diagram



9.2.5.1.102.3 Fields

Field	Function
31-10 —	Reserved
9-8 rc_derate_value	Derate value of tRC for LPDDR4 - 0 - Derating uses +1. - 1 - Derating uses +2. - 2 - Derating uses +3. - 3 - Derating uses +4. Present only in designs configured to support LPDDR4. The required number of cycles for derating can be determined by dividing 3.75ns by the core_ddrc_core_clk period, and rounding up the next integer.
7-4 derate_byte	Derate byte. Present only for LPDDR4. Indicates which byte of the MRR data is used for derating. The maximum valid value depends on MEMC_DRAM_TOTAL_DATA_WIDTH.
3-2 —	Reserved
1 derate_value	Derate value - 0 - Derating uses +1. - 1 - Derating uses +2. Present only for LPDDR4. If the period of core_ddrc_core_clk is less than 1.875ns, this register field should be set to 1; otherwise it should be set to 0.
0 derate_enable	Enables derating - 0 - Timing parameter derating is disabled - 1 - Timing parameter derating is enabled using MR4 read value. Present only for LPDDR4. This field must be set to '0' for non-LPDDR4 mode.

9.2.5.1.103 [SHADOW] Temperature Derate Interval Register (DERATEINT_SHADOW)

9.2.5.1.103.1 Offset

Register	Offset
DERATEINT_SHADOW	2024h

9.2.5.1.103.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr4_read_interval															
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	mr4_read_interval															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.103.3 Fields

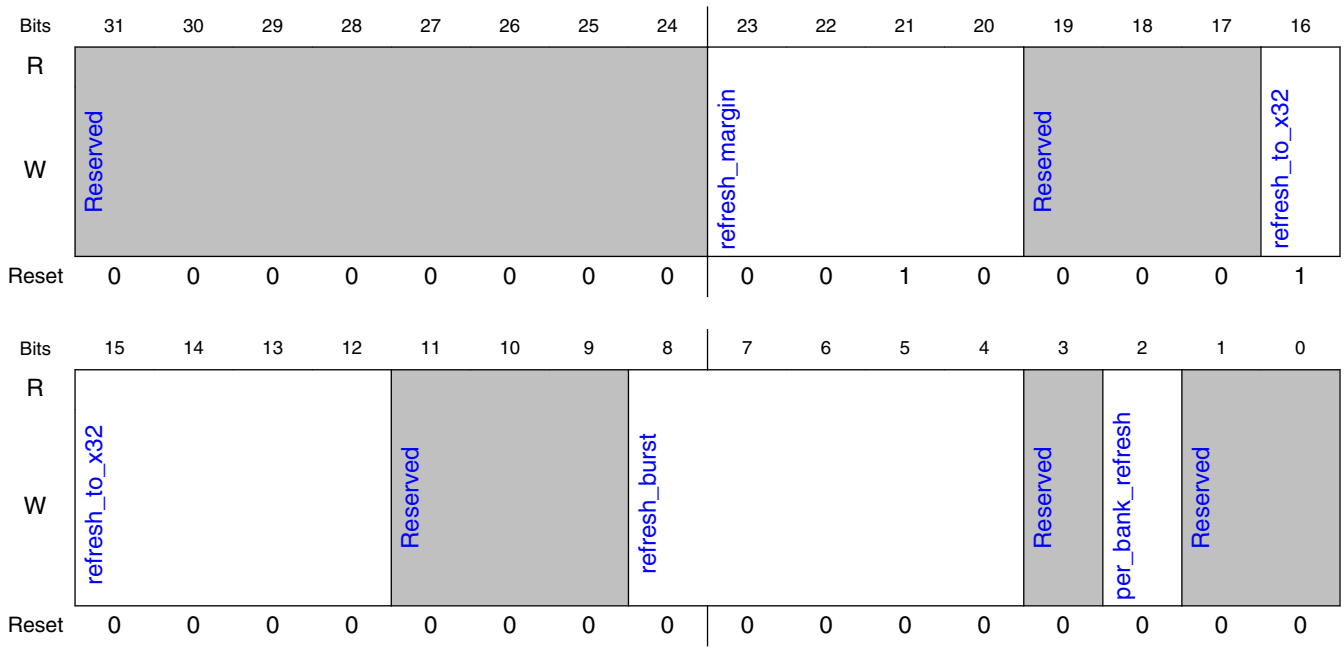
Field	Function
31-0 mr4_read_interval	Interval between two MR4 reads, used to derate the timing parameters. Present only for LPDDR4. This register must not be set to zero. Unit: DFI clock cycle.

9.2.5.1.104 [SHADOW] Refresh Control Register 0 (RFSHCTL0_SHADOW)

9.2.5.1.104.1 Offset

Register	Offset
RFSHCTL0_SHADOW	2050h

9.2.5.1.104.2 Diagram



9.2.5.1.104.3 Fields

Field	Function
31-24 —	Reserved
23-20 refresh_margin	Threshold value in number of DFI clock cycles before the critical refresh or page timer expires. A critical refresh is to be issued before this threshold is reached. It is recommended that this not be changed from the default value, currently shown as 0x2. It must always be less than internally used t_rfc_nom_x32. Note that, in LPDDR4, internally used t_rfc_nom_x32 may be equal to RFSHTMG.t_rfc_nom_x32>>2 if derating is enabled (DERATEEN.derate_enable=1). Otherwise, internally used t_rfc_nom_x32 will be equal to RFSHTMG.t_rfc_nom_x32. Unit: Multiples of 32 DFI clocks.
19-17 —	Reserved
16-12 refresh_to_x32	If the refresh timer (tRFCnom, also known as tREFI) has expired at least once, but it has not expired (RFSHCTL0.refresh_burst+1) times yet, then a speculative refresh may be performed. A speculative refresh is a refresh performed at a time when refresh would be useful, but before it is absolutely required. When the SDRAM bus is idle for a period of time determined by this RFSHCTL0.refresh_to_x32 and the refresh timer has expired at least once since the last refresh, then a speculative refresh is performed. Speculative refreshes continues successively until there are no refreshes pending or until new reads or writes are issued to the DDRC. FOR PERFORMANCE ONLY. Unit: Multiples of 32 DFI clocks.
11-9 —	Reserved
8-4 refresh_burst	The programmed value + 1 is the number of refresh timeouts that is allowed to accumulate before traffic is blocked and the refreshes are forced to execute. Closing pages to perform a refresh is a one-time penalty that must be paid for each group of refreshes. Therefore, performing refreshes in a burst reduces the per-refresh penalty of these page closings. Higher numbers for RFSHCTL.refresh_burst slightly

Table continues on the next page...

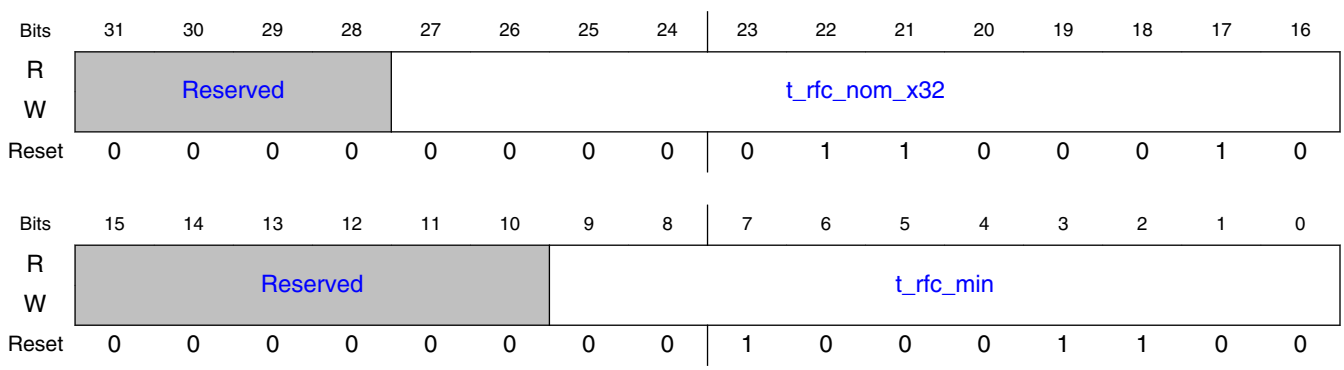
Field	Function
	increases utilization; lower numbers decreases the worst-case latency associated with refreshes. - 0 - single refresh - 1 - burst-of-2 refresh - 7 - burst-of-8 refresh. In DDR4 mode, according to Fine Granularity feature, 8 refreshes can be postponed in 1X mode, 16 refreshes in 2X mode and 32 refreshes in 4X mode. If using PHY-initiated updates, care must be taken in the setting of RFSHCTL0.refresh_burst, to ensure that tRFCmax is not violated due to a PHY-initiated update occurring shortly before a refresh burst was due. In this situation, the refresh burst will be delayed until the PHY-initiated update is complete.
3 —	Reserved
2 per_bank_refresh	- 1 - Per bank refresh; - 0 - All bank refresh. Per bank refresh allows traffic to flow to other banks. Present only for LPDDR4.
1-0 —	Reserved

9.2.5.1.105 [SHADOW] Refresh Timing Register (RFSHTMG_SHADOW)

9.2.5.1.105.1 Offset

Register	Offset
RFSHTMG_SHADOW	2064h

9.2.5.1.105.2 Diagram



9.2.5.1.105.3 Fields

Field	Function
31-28 —	Reserved

Table continues on the next page...

Field	Function
27-16 t_rfc_nom_x32	tREFI: Average time interval between refreshes per rank (Specification: 7.8us for DDR4. See JEDEC specification for LPDDR4). For LPDDR4: - if using all-bank refreshes (RFSHCTL0.per_bank_refresh = 0), this register should be set to tREFIab - if using per-bank refreshes (RFSHCTL0.per_bank_refresh = 1), this register should be set to tREFIpb When the controller is operating in 1:2 frequency ratio mode, program this to (tREFI/2), no rounding up. In DDR4 mode, tREFI value is different depending on the refresh mode. The user should program the appropriate value from the spec based on the value programmed in the refresh mode register. Note that RFSHTMG.t_rfc_nom_x32 * 32 must be greater than RFSHTMG.t_rfc_min, and RFSHTMG.t_rfc_nom_x32 must be greater than 0x1. - Non-DDR4 or DDR4 Fixed 1x mode: RFSHTMG.t_rfc_nom_x32 must be less than or equal to 0xFFE. - DDR4 Fixed 2x mode: RFSHTMG.t_rfc_nom_x32 must be less than or equal to 0x7FF. - DDR4 Fixed 4x mode: RFSHTMG.t_rfc_nom_x32 must be less than or equal to 0x3FF. Unit: Multiples of 32 clocks.
15-10 —	Reserved
9-0 t_rfc_min	tRFC (min): Minimum time from refresh to refresh or activate. When the controller is operating in 1:1 mode, t_rfc_min should be set to RoundUp(tRFCmin/tCK). When the controller is operating in 1:2 mode, t_rfc_min should be set to RoundUp(RoundUp(tRFCmin/tCK)/2). In LPDDR4 mode: - if using all-bank refreshes, the tRFCmin value in the above equations is equal to tRFCab - if using per-bank refreshes, the tRFCmin value in the above equations is equal to tRFCpb In DDR4 mode, the tRFCmin value in the above equations is different depending on the refresh mode (fixed 1X,2X,4X) and the device density. The user should program the appropriate value from the spec based on the 'refresh_mode' and the device density that is used. Unit: Clocks.

9.2.5.1.106 [SHADOW] SDRAM Initialization Register 3 (INIT3_SHADOW)

9.2.5.1.106.1 Offset

Register	Offset
INIT3_SHADOW	20DCh

9.2.5.1.106.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	emr															
Reset	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0

9.2.5.1.106.3 Fields

Field	Function
31-16 mr	Bit 8 is for DLL and the setting here is ignored. The DDRC sets this bit appropriately. DDR4: Value loaded into MR0 register. LPDDR4 - Value to write to MR1 register
15-0 emr	Bits 9:7 are for OCD and the setting in this register is ignored. The DDRC sets those bits appropriately. DDR4: Value to write to MR1 register Set bit 7 to 0. If PHY-evaluation mode training is enabled, this bit is set appropriately by the DDRC during write leveling. LPDDR4 - Value to write to MR2 register

9.2.5.1.107 [SHADOW] SDRAM Initialization Register 4 (INIT4_SHADOW)

9.2.5.1.107.1 Offset

Register	Offset
INIT4_SHADOW	20E0h

9.2.5.1.107.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	emr2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	emr3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.107.3 Fields

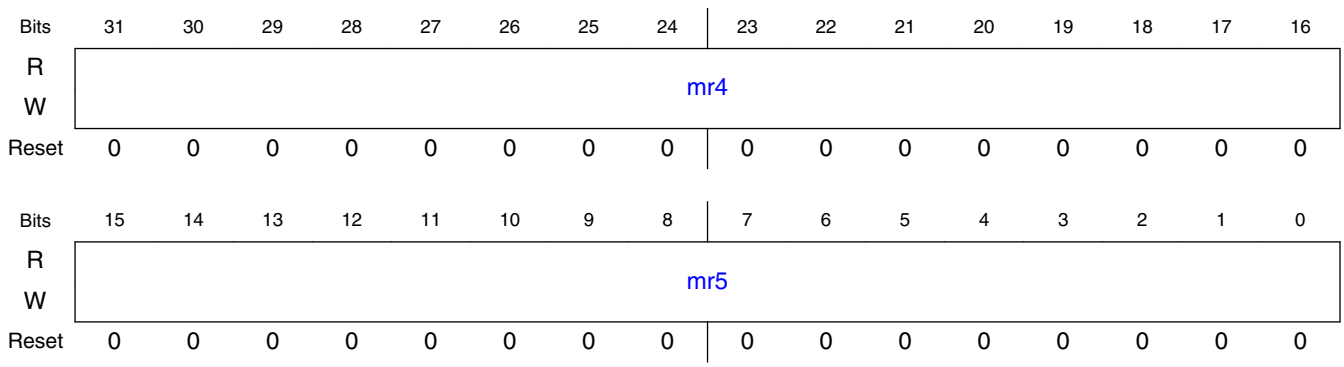
Field	Function
31-16 emr2	DDR4: Value to write to MR2 register. LPDDR4: Value to write to MR3 register.
15-0 emr3	DDR4: Value to write to MR3 register. LPDDR4: Value to write to MR13 register.

9.2.5.1.108 [SHADOW] SDRAM Initialization Register 6 (INIT6_SHADOW)

9.2.5.1.108.1 Offset

Register	Offset
INIT6_SHADOW	20E8h

9.2.5.1.108.2 Diagram



9.2.5.1.108.3 Fields

Field	Function
31-16 mr4	DDR4- Value to be loaded into SDRAM MR4 registers. Used in DDR4 designs only.
15-0 mr5	DDR4- Value to be loaded into SDRAM MR5 registers. Used in DDR4 designs only.

9.2.5.1.109 [SHADOW] SDRAM Initialization Register 7 (INIT7_SHADOW)

9.2.5.1.109.1 Offset

Register	Offset
INIT7_SHADOW	20ECh

9.2.5.1.109.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr6															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.5.1.109.3 Fields

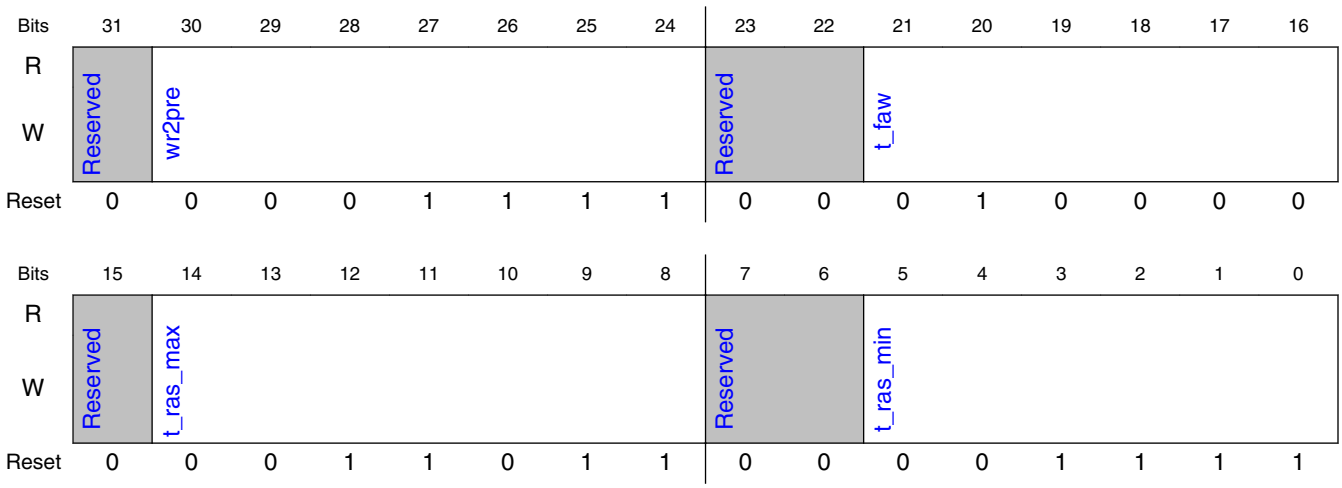
Field	Function
31-16 mr6	DDR4- Value to be loaded into SDRAM MR6 registers. Used in DDR4 designs only.
15-0 —	Reserved

9.2.5.1.110 [SHADOW] SDRAM Timing Register 0 (DRAMTMG0_SHADOW)

9.2.5.1.110.1 Offset

Register	Offset
DRAMTMG0_SHADOW	2100h

9.2.5.1.110.2 Diagram



9.2.5.1.110.3 Fields

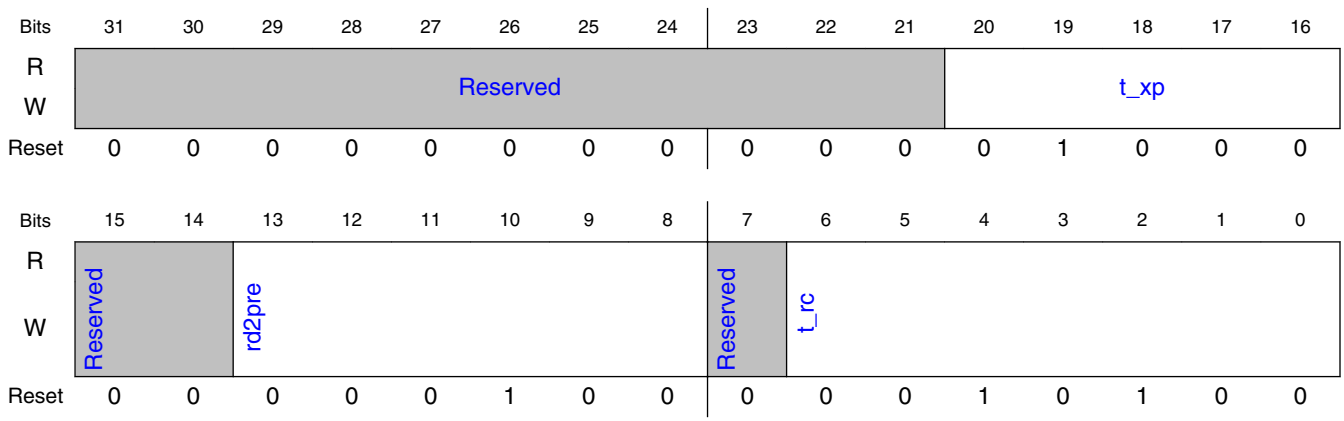
Field	Function
31 —	Reserved
30-24 wr2pre	Minimum time between write and precharge to same bank. Unit: Clocks Specifications: WL + BL/2 + tWR = approximately 8 cycles + 15 ns = 14 clocks @400MHz and less for lower frequencies where: - WL = write latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM. BST (burst terminate) is not supported at present. - tWR = Write recovery time. This comes directly from the SDRAM specification. Add one extra cycle for LPDDR4 for this parameter. When the controller is operating in 1:2 frequency ratio mode, 1T mode, divide the above value by 2. No rounding up. When the controller is operating in 1:2 frequency ratio mode, 2T mode or LPDDR4 mode, divide the above value by 2 and round it up to the next integer value.
23-22 —	Reserved
21-16 t_faw	tFAW Valid only when 8 or more banks(or banks x bank groups) are present. In 8-bank design, at most 4 banks must be activated in a rolling window of tFAW cycles. When the controller is operating in 1:2 frequency ratio mode, program this to (tFAW/2) and round up to next integer value. In a 4-bank design, set this register to 0x1 independent of the 1:1/1:2 frequency mode. Unit: Clocks
15 —	Reserved
14-8 t_ras_max	tRAS(max): Maximum time between activate and precharge to same bank. This is the maximum time that a page can be kept open Minimum value of this register is 1. Zero is invalid. When the controller is operating in 1:2 frequency ratio mode, program this to (tRAS(max)-1)/2. No rounding up. Unit: Multiples of 1024 clocks.
7-6 —	Reserved
5-0 t_ras_min	tRAS(min): Minimum time between activate and precharge to the same bank. When the controller is operating in 1:2 frequency mode, 1T mode, program this to tRAS(min)/2. No rounding up. When the controller is operating in 1:2 frequency ratio mode, 2T mode or LPDDR4 mode, program this to (tRAS(min)/2) and round it up to the next integer value. Unit: Clocks

9.2.5.1.111 [SHADOW] SDRAM Timing Register 1 (DRAMTMG1_SHADOW)

9.2.5.1.111.1 Offset

Register	Offset
DRAMTMG1_SHADOW	2104h

9.2.5.1.111.2 Diagram



9.2.5.1.111.3 Fields

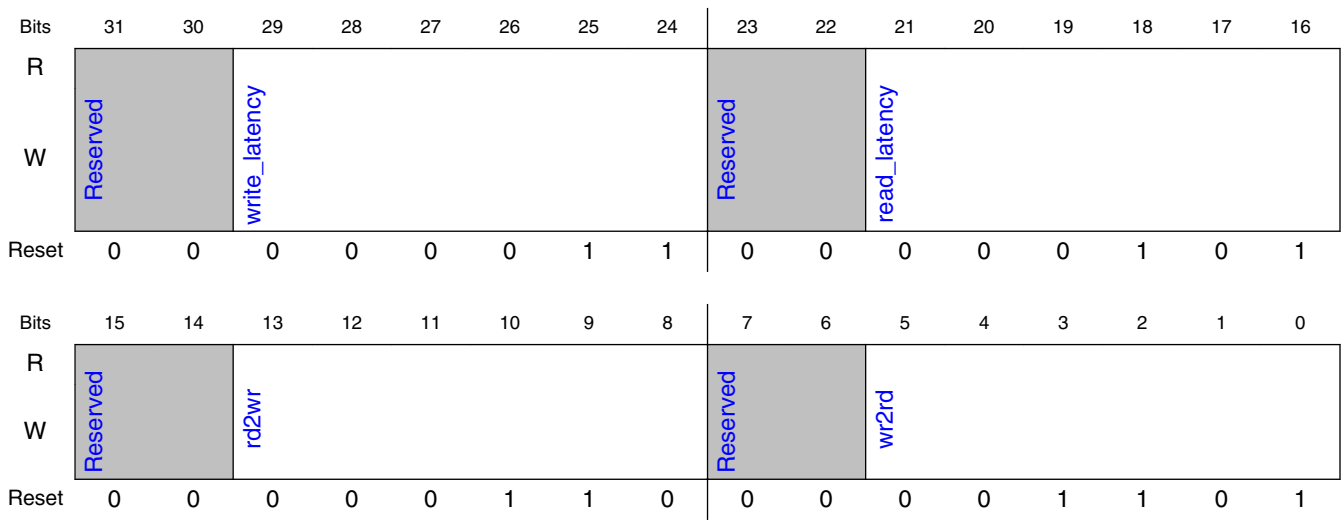
Field	Function
31-21 —	Reserved
20-16 t_xp	tXP: Minimum time after power-down exit to any operation. If C/A parity for DDR4 is used, set to (tXP +PL) instead. When the controller is operating in 1:2 frequency ratio mode, program this to (tXP/2) and round it up to the next integer value. Units: Clocks
15-14 —	Reserved
13-8 rd2pre	tRTP: Minimum time from read to precharge of same bank. - DDR4: Max of following two equations: tAL + max (tRTP, 4) or, RL + BL/2 - tRP (*). - LPDDR4: BL/2 + max(tRTP,8) - 8 (*). When both DDR4 SDRAM and ST-MRAM are used simultaneously, use SDRAM's tRP value for calculation. When the controller is operating in 1:2 mode, 1T mode, divide the above value by 2. No rounding up. When the controller is operating in 1:2 mode, 2T mode or LPDDR4 mode, divide the above value by 2 and round it up to the next integer value. Unit: Clocks.
7 —	Reserved
6-0 t_rc	tRC: Minimum time between activates to same bank. When the controller is operating in 1:2 frequency ratio mode, program this to (tRC/2) and round up to next integer value. Unit: Clocks.

9.2.5.1.112 [SHADOW] SDRAM Timing Register 2 (DRAMTMG2_SHADOW)

9.2.5.1.112.1 Offset

Register	Offset
DRAMTMG2_SHADOW	2108h

9.2.5.1.112.2 Diagram



9.2.5.1.112.3 Fields

Field	Function
31-30 —	Reserved
29-24 write_latency	Set to WL Time from write command to write data on SDRAM interface. This must be set to WL. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer. Unit: clocks
23-22 —	Reserved
21-16 read_latency	Set to RL Time from read command to read data on SDRAM interface. This must be set to RL.When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer. Unit: clocks
15-14 —	Reserved
13-8	DDR4: RL + BL/2 + 1 + WR_PREAMBLE - WL. LPDDR4(DQ ODT is Disabled): RL + BL/2 + RU(tDQSCKmax/tCK) + WR_PREAMBLE + RD_POSTAMBLE - WL. LPDDR4(DQ ODT is Enabled) : RL

Table continues on the next page...

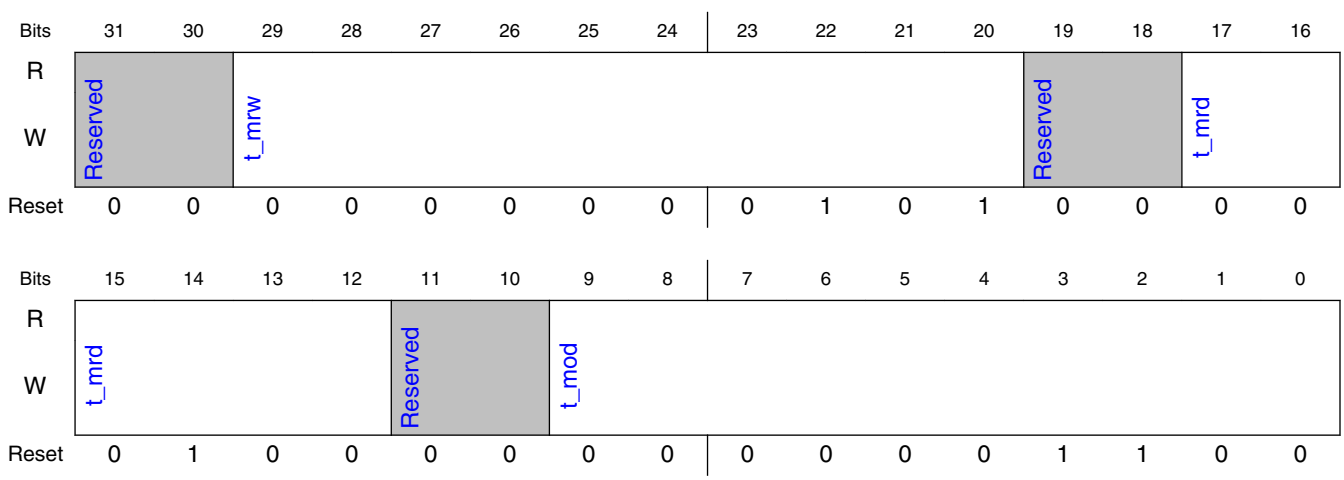
Field	Function
rd2wr	+ BL/2 + RU(tDQSCKmax/tCK) + RD_POSTAMBLE - ODTLon - RU(tODTon(min)/tCK). Minimum time from read command to write command. Include time for bus turnaround and all per-bank, per-rank, and global constraints. Please see the relevant PHY databook for details of what should be included here. Unit: Clocks. Where: - WL = write latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM - RL = read latency = CAS latency - WR_PREAMBLE = write preamble. This is unique to DDR4 and LPDDR4. - RD_POSTAMBLE = read postamble. This is unique to LPDDR4. For LPDDR4, if derating is enabled (DERATEEN.derate_enable=1), derated tDQSCKmax should be used. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer.
7-6 —	Reserved
5-0 wr2rd	DDR4: CWL + PL + BL/2 + tWTR_L Others: CWL + BL/2 + tWTR In DDR4, minimum time from write command to read command for same bank group. In others, minimum time from write command to read command. Includes time for bus turnaround, recovery times, and all per-bank, per-rank, and global constraints. Unit: Clocks. Where: - CWL = CAS write latency - PL = Parity latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM - tWTR_L = internal write to read command delay for same bank group. This comes directly from the SDRAM specification. - tWTR = internal write to read command delay. This comes directly from the SDRAM specification. Add one extra cycle for LPDDR4 operation. When the controller is operating in 1:2 mode, divide the value calculated using the above equation by 2, and round it up to next integer.

9.2.5.1.113 [SHADOW] SDRAM Timing Register 3 (DRAMTMG3_SHADOW)

9.2.5.1.113.1 Offset

Register	Offset
DRAMTMG3_SHADOW	210Ch

9.2.5.1.113.2 Diagram



9.2.5.1.113.3 Fields

Field	Function
31-30 —	Reserved
29-20 t_mrw	Time to wait after a mode register write or read (MRW or MRR). Present only for LPDDR4. LPDDR4: Set this to the larger of tMRW and tMRWCKEL. When the controller is operating in 1:2 frequency ratio mode, program this to the above values divided by 2 and round it up to the next integer value.
19-18 —	Reserved
17-12 t_mrd	tMRD: Cycles to wait after a mode register write or read. Depending on the connected SDRAM, tMRD represents: DDR4: Time from MRS to MRS command. LPDDR4: Time from MRS to non-MRS command. When the controller is operating in 1:2 frequency ratio mode, program this to (tMRD/2) and round it up to the next integer value. If C/A parity for DDR4 is used, set to tMRD_PAR(tMOD+PL) instead.
11-10 —	Reserved
9-0 t_mod	tMOD: Parameter used only in DDR4. Cycles between load mode command and following non-load mode command. If C/A parity for DDR4 is used, set to tMOD_PAR(tMOD+PL) instead. Set to tMOD if controller is operating in 1:1 frequency ratio mode, or tMOD/2 (rounded up to next integer) if controller is operating in 1:2 frequency ratio mode.

9.2.5.1.114 [SHADOW] SDRAM Timing Register 4 (DRAMTMG4_SHADOW)

9.2.5.1.114.1 Offset

Register	Offset
DRAMTMG4_SHADOW	2110h

9.2.5.1.114.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			t_rcd					Reserved				t_ccd			
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				t_rrd				Reserved				t_rp			
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1

9.2.5.1.114.3 Fields

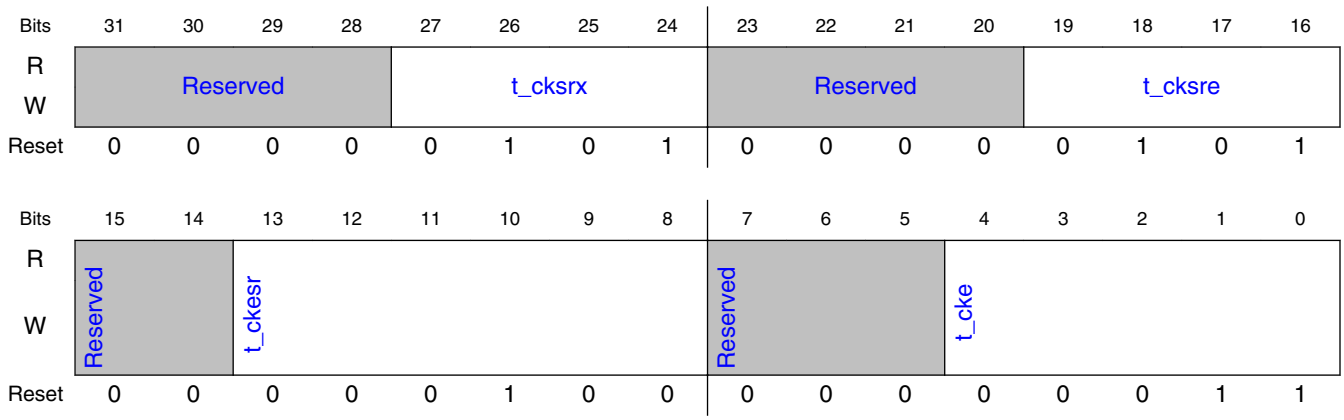
Field	Function
31-29 —	Reserved
28-24 t_rcd	tRCD - tAL: Minimum time from activate to read or write command to same bank. When the controller is operating in 1:2 frequency ratio mode, program this to ((tRCD - tAL)/2) and round it up to the next integer value. Minimum value allowed for this register is 1, which implies minimum (tRCD - tAL) value to be 2 when the controller is operating in 1:2 frequency ratio mode. Unit: Clocks.
23-20 —	Reserved
19-16 t_ccd	DDR4: tCCD_L: This is the minimum time between two reads or two writes for same bank group. Others: tCCD: This is the minimum time between two reads or two writes. When the controller is operating in 1:2 frequency ratio mode, program this to (tCCD_L/2 or tCCD/2) and round it up to the next integer value. Unit: clocks.
15-12 —	Reserved
11-8 t_rrd	DDR4: tRRD_L: Minimum time between activates from bank "a" to bank "b" for same bank group. Others: tRRD: Minimum time between activates from bank "a" to bank "b" When the controller is operating in 1:2 frequency ratio mode, program this to (tRRD_L/2 or tRRD/2) and round it up to the next integer value. Unit: Clocks.
7-5 —	Reserved
4-0 t_rp	tRP: Minimum time from precharge to activate of same bank. When the controller is operating in 1:1 frequency ratio mode, t_rp should be set to RoundUp(tRP/tCK). When the controller is operating in 1:2 frequency ratio mode, t_rp should be set to RoundDown(RoundUp(tRP/tCK)/2) + 1. When the controller is operating in 1:2 frequency ratio mode in LPDDR4, t_rp should be set to RoundUp(RoundUp(tRP/tCK)/2). Unit: Clocks.

9.2.5.1.115 [SHADOW] SDRAM Timing Register 5 (DRAMTMG5_SHADOW)

9.2.5.1.115.1 Offset

Register	Offset
DRAMTMG5_SHADOW	2114h

9.2.5.1.115.2 Diagram



9.2.5.1.115.3 Fields

Field	Function
31-28 —	Reserved
27-24 t_cksrx	This is the time before Self Refresh Exit that CK is maintained as a valid clock before issuing SRX. Specifies the clock stable time before SRX. Recommended settings: LPDDR4: tCKCKEH. DDR4: tCKSRX When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
23-20 —	Reserved
19-16 t_cksre	This is the time after Self Refresh Down Entry that CK is maintained as a valid clock. Specifies the clock disable delay after SRE. Recommended settings: LPDDR4: tCKCKEL. DDR4: max (10 ns, 5 tCK) (+ PL(parity latency)(*)) (*)Only if CRCPARCTL1.caparity_disable_before_sr=0, this register should be increased by PL. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
15-14 —	Reserved
13-8 t_ckesr	Minimum CKE low width for Self refresh or Self refresh power down entry to exit timing in memory clock cycles. Recommended settings: LPDDR4: max(tCKELPD, tSR). DDR4: tCKE + 1 (+ PL(parity latency)(*)) (*)Only if CRCPARCTL1.caparity_disable_before_sr=0, this register should be increased by PL. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
7-5 —	Reserved
4-0 t_cke	Minimum number of cycles of CKE HIGH/LOW during power-down and self refresh. - LPDDR4 mode: Set this to the larger of tCKE, tCKELPD or tSR. - non-LPDDR4 designs: Set this to tCKE value. When the controller is operating in 1:2 frequency ratio mode, program this to (value described above)/2 and round it up to the next integer value. Unit: Clocks.

9.2.5.1.116 [SHADOW] SDRAM Timing Register 6 (DRAMTMG6_SHADOW)

9.2.5.1.116.1 Offset

Register	Offset
DRAMTMG6_SHADOW	2118h

9.2.5.1.116.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												t_ckcsx			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

9.2.5.1.116.3 Fields

Field	Function
31-4 —	Reserved
3-0 t_ckcsx	This is the time before Clock Stop Exit that CK is maintained as a valid clock before issuing Clock Stop Exit. Specifies the clock stable time before next command after Clock Stop Exit. Recommended settings: LPDDR4: tXP + 2. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for designs supporting LPDDR4 devices.

9.2.5.1.117 [SHADOW] SDRAM Timing Register 7 (DRAMTMG7_SHADOW)

9.2.5.1.117.1 Offset

Register	Offset
DRAMTMG7_SHADOW	211Ch

9.2.5.1.117.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				t_ckpde				Reserved				t_ckpdx			
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

9.2.5.1.117.3 Fields

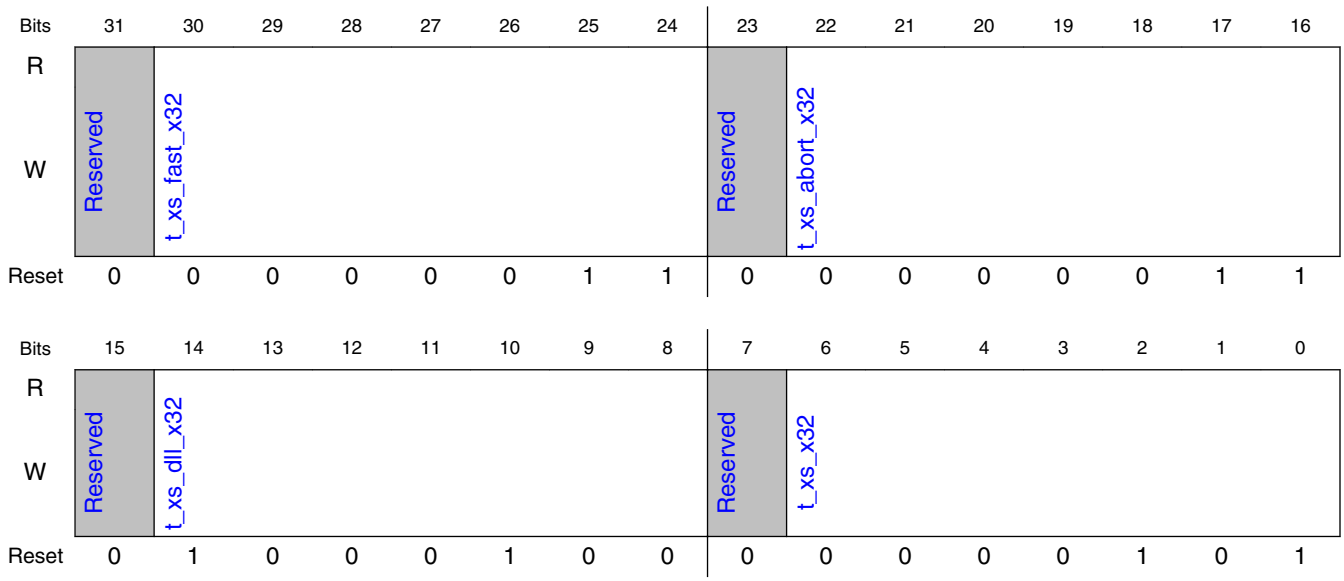
Field	Function
31-12 —	Reserved
11-8 t_ckpde	This is the time after Power Down Entry that CK is maintained as a valid clock. Specifies the clock disable delay after PDE. Recommended settings: LPDDR4: tCKCKEL. When using DDR4 SDRAM, this register should be set to the same value as DRAMTMG5.t_cksre. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for designs supporting LPDDR4 devices.
7-4 —	Reserved
3-0 t_ckpdx	This is the time before Power Down Exit that CK is maintained as a valid clock before issuing PDX. Specifies the clock stable time before PDX. Recommended settings: LPDDR4: 2. When using DDR4 SDRAM, this register should be set to the same value as DRAMTMG5.t_cksrx. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for designs supporting LPDDR4 devices.

9.2.5.1.118 [SHADOW] SDRAM Timing Register 8 (DRAMTMG8_SHADOW)

9.2.5.1.118.1 Offset

Register	Offset
DRAMTMG8_SHADOW	2120h

9.2.5.1.118.2 Diagram



9.2.5.1.118.3 Fields

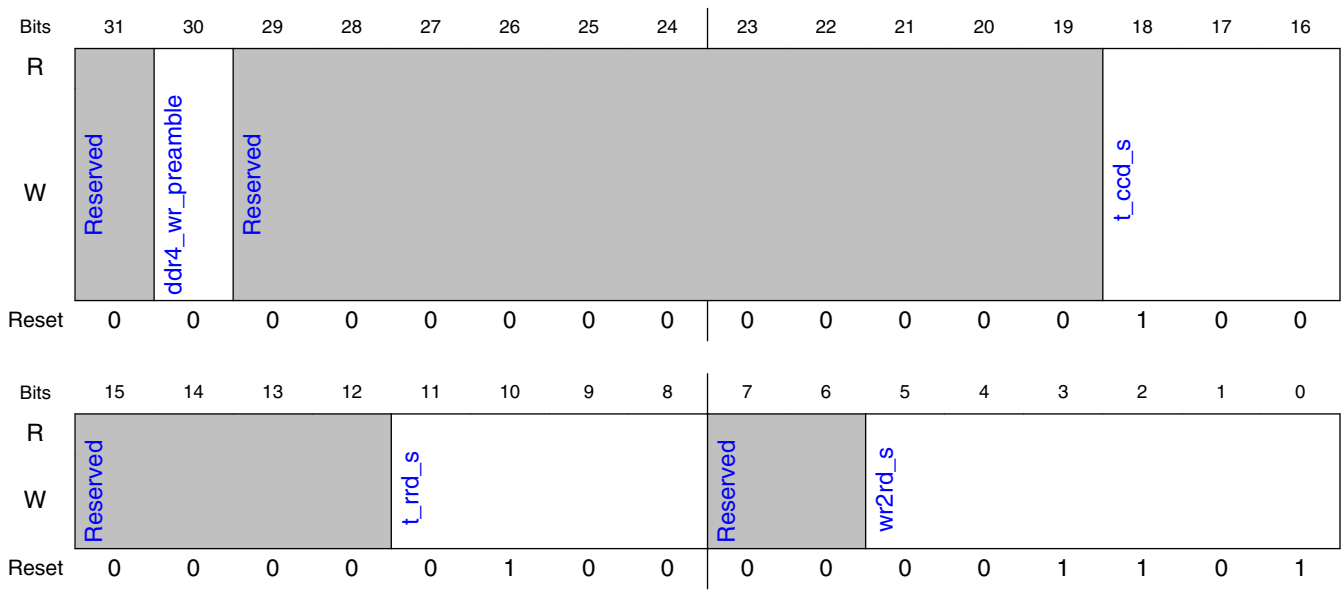
Field	Function
31 —	Reserved
30-24 t_xs_fast_x32	tXS_FAST: Exit Self Refresh to ZQCL, ZQCS and MRS (only CL, WR, RTP and Geardown mode). When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: This is applicable to only ZQCL/ZQCS commands. Note: Ensure this is less than or equal to t_xs_x32.
23 —	Reserved
22-16 t_xs_abort_x32	tXS_ABORT: Exit Self Refresh to commands not requiring a locked DLL in Self Refresh Abort. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: Ensure this is less than or equal to t_xs_x32.
15 —	Reserved
14-8 t_xs_dll_x32	tXSDLL: Exit Self Refresh to commands requiring a locked DLL. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: Used only for DDR4 SDRAM.
7 —	Reserved
6-0 t_xs_x32	tXS: Exit Self Refresh to commands not requiring a locked DLL. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: Used only for DDR4 SDRAM.

9.2.5.1.119 [SHADOW] SDRAM Timing Register 9 (DRAMTMG9_SHADOW)

9.2.5.1.119.1 Offset

Register	Offset
DRAMTMG9_SHADOW	2124h

9.2.5.1.119.2 Diagram



9.2.5.1.119.3 Fields

Field	Function
31 —	Reserved
30 ddr4_wr_preamble	DDR4 Write preamble mode - 0: 1tCK preamble - 1: 2tCK preamble Present only with MEMC_FREQ_RATIO=2
29-19 —	Reserved
18-16 t_ccd_s	tCCD_S: This is the minimum time between two reads or two writes for different bank group. For bank switching (from bank "a" to bank "b"), the minimum time is this value + 1. When the controller is operating in 1:2 frequency ratio mode, program this to (tCCD_S/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: clocks.
15-12 —	Reserved

Table continues on the next page...

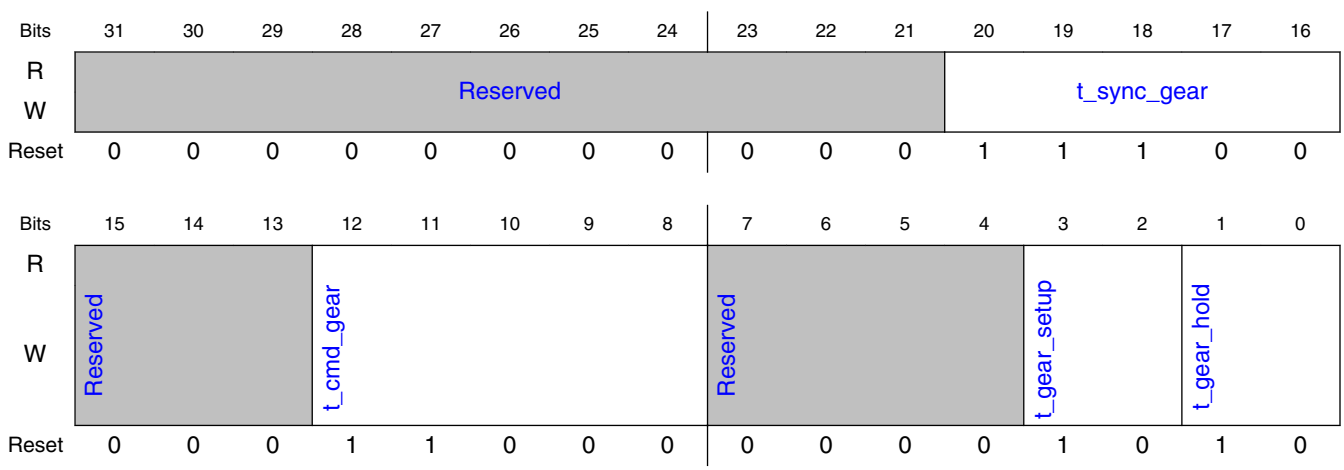
Field	Function
11-8 t_rrd_s	tRRD_S: Minimum time between activates from bank "a" to bank "b" for different bank group. When the controller is operating in 1:2 frequency ratio mode, program this to (tRRD_S/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: Clocks.
7-6 —	Reserved
5-0 wr2rd_s	CWL + PL + BL/2 + tWTR_S Minimum time from write command to read command for different bank group. Includes time for bus turnaround, recovery times, and all per-bank, per-rank, and global constraints. Present only in designs configured to support DDR4. Unit: Clocks. Where: - CWL = CAS write latency - PL = Parity latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM - tWTR_S = internal write to read command delay for different bank group. This comes directly from the SDRAM specification. When the controller is operating in 1:2 mode, divide the value calculated using the above equation by 2, and round it up to next integer.

9.2.5.1.120 [SHADOW] SDRAM Timing Register 10 (DRAMTMG10_SHADOW)

9.2.5.1.120.1 Offset

Register	Offset
DRAMTMG10_SHADOW	2128h

9.2.5.1.120.2 Diagram



9.2.5.1.120.3 Fields

Field	Function
31-21	Reserved

Table continues on the next page...

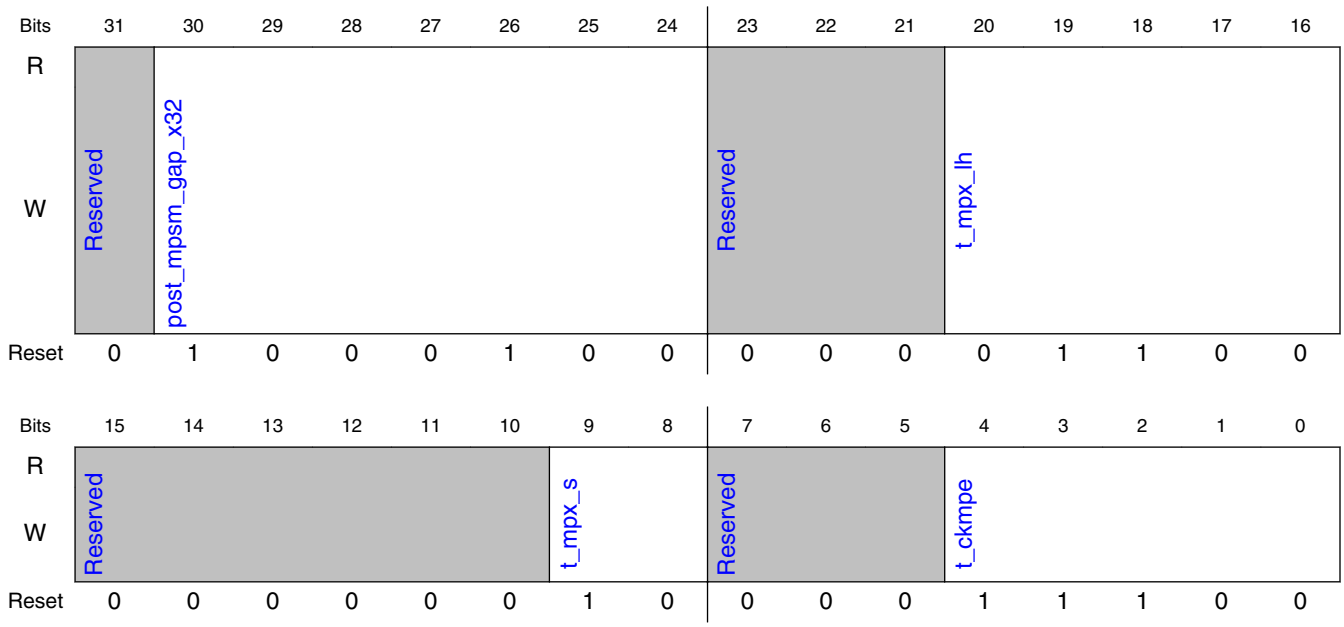
Field	Function
—	
20-16 t_sync_gear	Indicates the time between MRS command and the sync pulse time. This must be even number of clocks. For DDR4-2666 and DDR4-3200, this parameter is defined as tMOD(min)+4nCK tMOD(min) is greater of 24nCK or 15ns 15ns / .625ns = 24 Max value for this register is 24+4 = 28 When the controller is operating in 1:2 mode, program this to (tSYNC_GEAR/2) and round it up to the next integer value. Unit: Clocks
15-13 —	Reserved
12-8 t_cmd_gear	Sync pulse to first valid command. For DDR4-2666 and DDR4-3200, this parameter is defined as tMOD(min) tMOD(min) is greater of 24nCK or 15ns 15ns / .625ns = 24 Max value for this register is 24 When the controller is operating in 1:2 mode, program this to (tCMD_GEAR/2) and round it up to the next integer value. Unit: Clocks
7-4 —	Reserved
3-2 t_gear_setup	Geardown setup time. Minimum value of this register is 1. Zero is invalid. For DDR4-2666 and DDR4-3200, this parameter is defined as 2 clks When the controller is operating in 1:2 frequency ratio mode, program this to (tGEAR_setup/2) and round it up to the next integer value. Unit: Clocks
1-0 t_gear_hold	Geardown hold time. Minimum value of this register is 1. Zero is invalid. For DDR4-2666 and DDR4-3200, this parameter is defined as 2 clks When the controller is operating in 1:2 frequency ratio mode, program this to (tGEAR_hold/2) and round it up to the next integer value. Unit: Clocks

9.2.5.1.121 [SHADOW] SDRAM Timing Register 11 (DRAMTMG11_SHADOW)

9.2.5.1.121.1 Offset

Register	Offset
DRAMTMG11_SHADOW	212Ch

9.2.5.1.121.2 Diagram



9.2.5.1.121.3 Fields

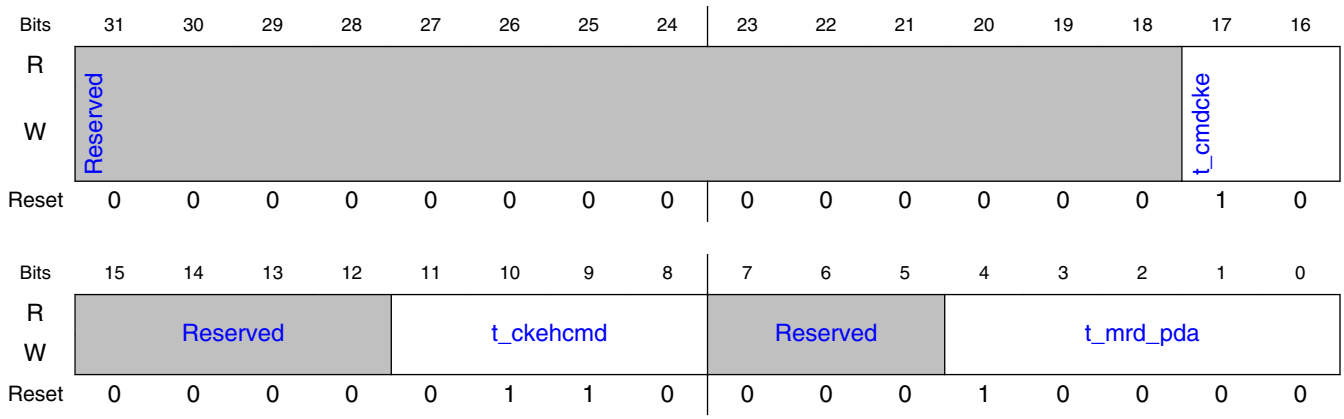
Field	Function
31 —	Reserved
30-24 post_mpsm_gap_x32	tXMPDLL: This is the minimum Exit MPSM to commands requiring a locked DLL. When the controller is operating in 1:2 frequency ratio mode, program this to (tXMPDLL/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: Multiples of 32 clocks.
23-21 —	Reserved
20-16 t_mpx_lh	tMPX_LH: This is the minimum CS_n Low hold time to CKE rising edge. When the controller is operating in 1:2 frequency ratio mode, program this to RoundUp(tMPX_LH/2)+1. Present only in designs configured to support DDR4. Unit: clocks.
15-10 —	Reserved
9-8 t_mpx_s	tMPX_S: Minimum time CS setup time to CKE. When the controller is operating in 1:2 frequency ratio mode, program this to (tMPX_S/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: Clocks.
7-5 —	Reserved
4-0 t_ckmpe	tCKMPE: Minimum valid clock requirement after MPSM entry. Present only in designs configured to support DDR4. Unit: Clocks. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer.

9.2.5.1.122 [SHADOW] SDRAM Timing Register 12 (DRAMTMG12_SHADOW)

9.2.5.1.122.1 Offset

Register	Offset
DRAMTMG12_SHADOW	2130h

9.2.5.1.122.2 Diagram



9.2.5.1.122.3 Fields

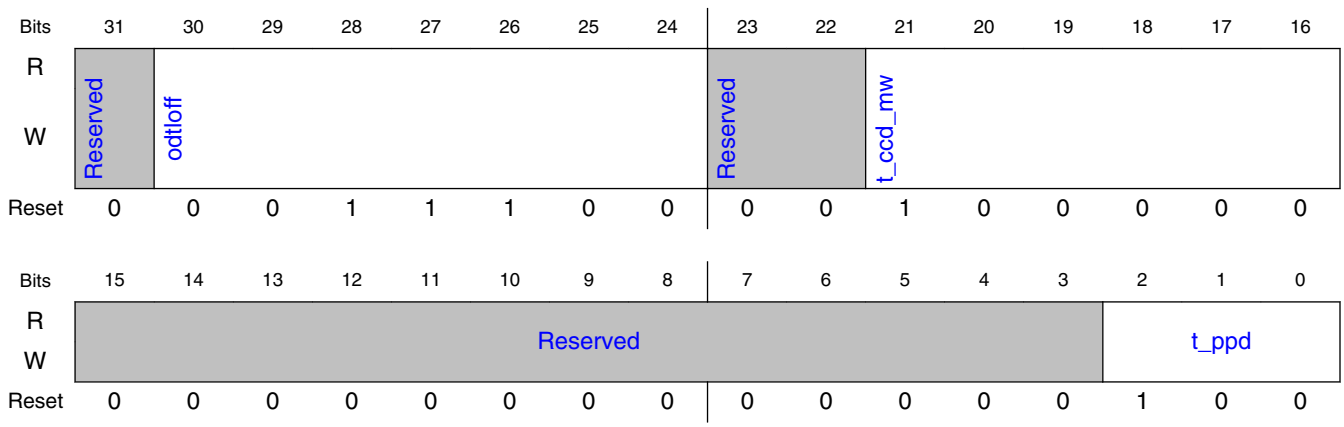
Field	Function
31-18 —	Reserved
17-16 t_cmdcke	tCMDCKE: Delay from valid command to CKE input LOW. Set this to the larger of tESCKE or tCMDCKE When the controller is operating in 1:2 frequency ratio mode, program this to (max(tESCKE, tCMDCKE)/2) and round it up to the next integer value.
15-12 —	Reserved
11-8 t_ckehcmd	tCKEHCMD: Valid command requirement after CKE input HIGH. When the controller is operating in 1:2 frequency ratio mode, program this to (tCKEHCMD/2) and round it up to the next integer value.
7-5 —	Reserved
4-0 t_mrd_pda	tMRD_PDA: This is the Mode Register Set command cycle time in PDA mode. When the controller is operating in 1:2 frequency ratio mode, program this to (tMRD_PDA/2) and round it up to the next integer value.

9.2.5.1.123 [SHADOW] SDRAM Timing Register 13 (DRAMTMG13_SHADOW)

9.2.5.1.123.1 Offset

Register	Offset
DRAMTMG13_SHADOW	2134h

9.2.5.1.123.2 Diagram



9.2.5.1.123.3 Fields

Field	Function
31 —	Reserved
30-24 odtloff	LPDDR4: tODTLoff: This is the latency from CAS-2 command to tODToff reference. When the controller is operating in 1:2 frequency ratio mode, program this to (tODTLoff/2) and round it up to the next integer value. Unit: Clocks.
23-22 —	Reserved
21-16 t_ccd_mw	LPDDR4: tCCDMW: This is the minimum time from write or masked write to masked write command for same bank. When the controller is operating in 1:2 frequency ratio mode, program this to (tCCDMW/2) and round it up to the next integer value. Unit: Clocks.
15-3 —	Reserved
2-0 t_ppd	LPDDR4: tPPD: This is the minimum time from precharge to precharge command. When the controller is operating in 1:2 frequency ratio mode, program this to (tPPD/2) and round it up to the next integer value. Unit: Clocks.

9.2.5.1.124 [SHADOW] SDRAM Timing Register 14 (DRAMTMG14_SHADOW)

9.2.5.1.124.1 Offset

Register	Offset
DRAMTMG14_SHADOW	2138h

9.2.5.1.124.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				t_xsr											
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

9.2.5.1.124.3 Fields

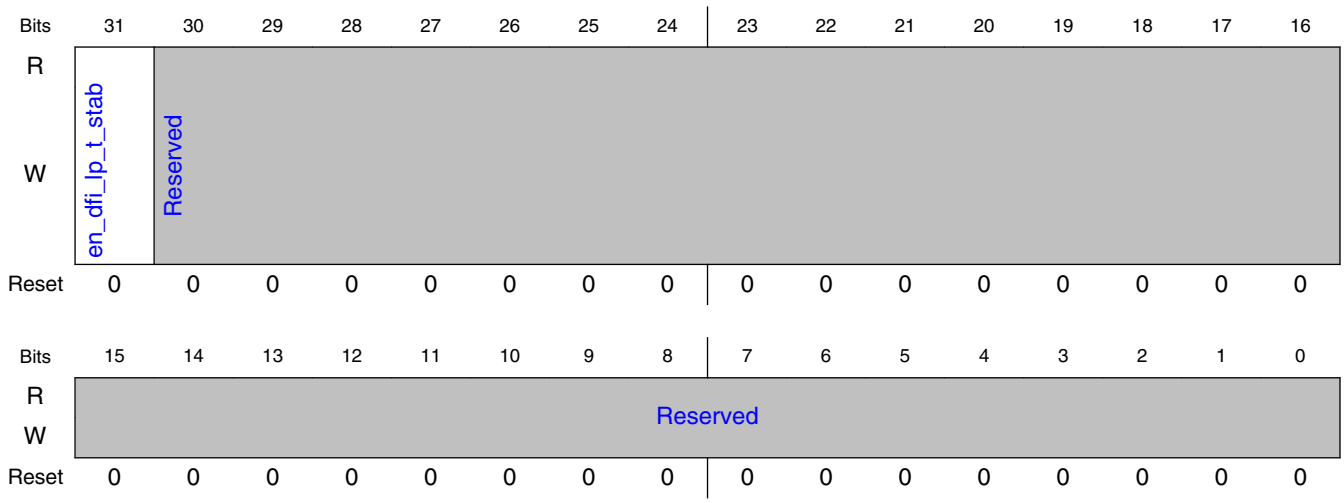
Field	Function
31-12 —	Reserved
11-0 t_xsr	tXSR: Exit Self Refresh to any command. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Note: Used only for LPDDR4 mode.

9.2.5.1.125 [SHADOW] SDRAM Timing Register 15 (DRAMTMG15_SHADOW)

9.2.5.1.125.1 Offset

Register	Offset
DRAMTMG15_SHADOW	213Ch

9.2.5.1.125.2 Diagram



9.2.5.1.125.3 Fields

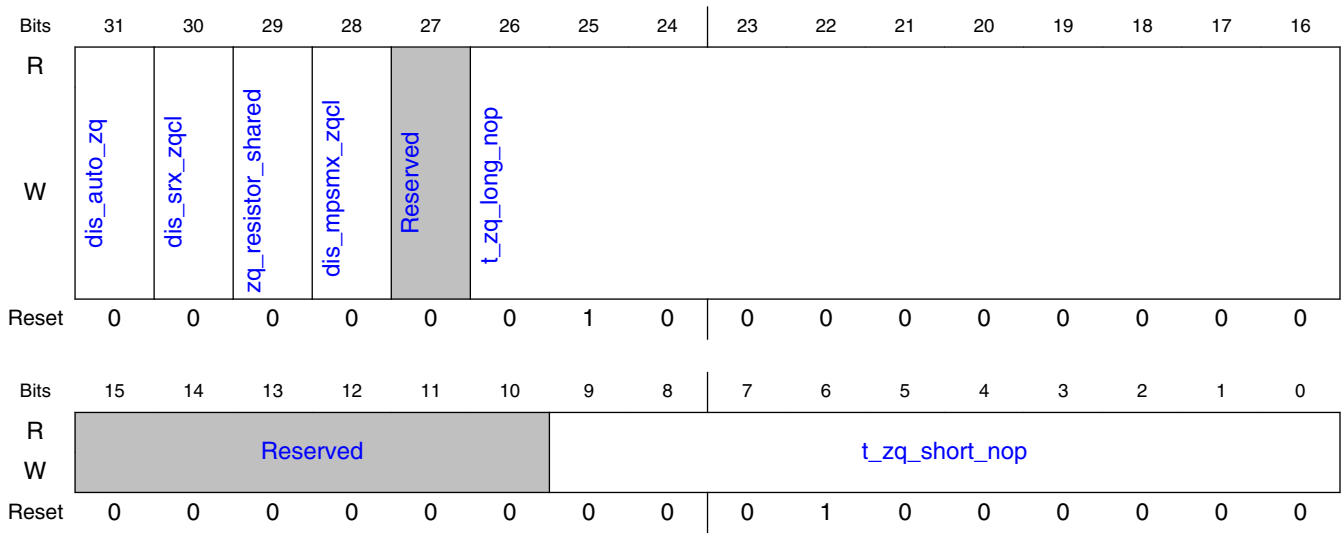
Field	Function
31 en_dfi_lp_t_stab	- 1 - Enable using tSTAB when exiting DFI LP. Needs to be set when the PHY is stopping the clock during DFI LP to save maximum power. - 0 - Disable using tSTAB when exiting DFI LP
30-0 —	Reserved

9.2.5.1.126 [SHADOW] ZQ Control Register 0 (ZQCTL0_SHADOW)

9.2.5.1.126.1 Offset

Register	Offset
ZQCTL0_SHADOW	2180h

9.2.5.1.126.2 Diagram



9.2.5.1.126.3 Fields

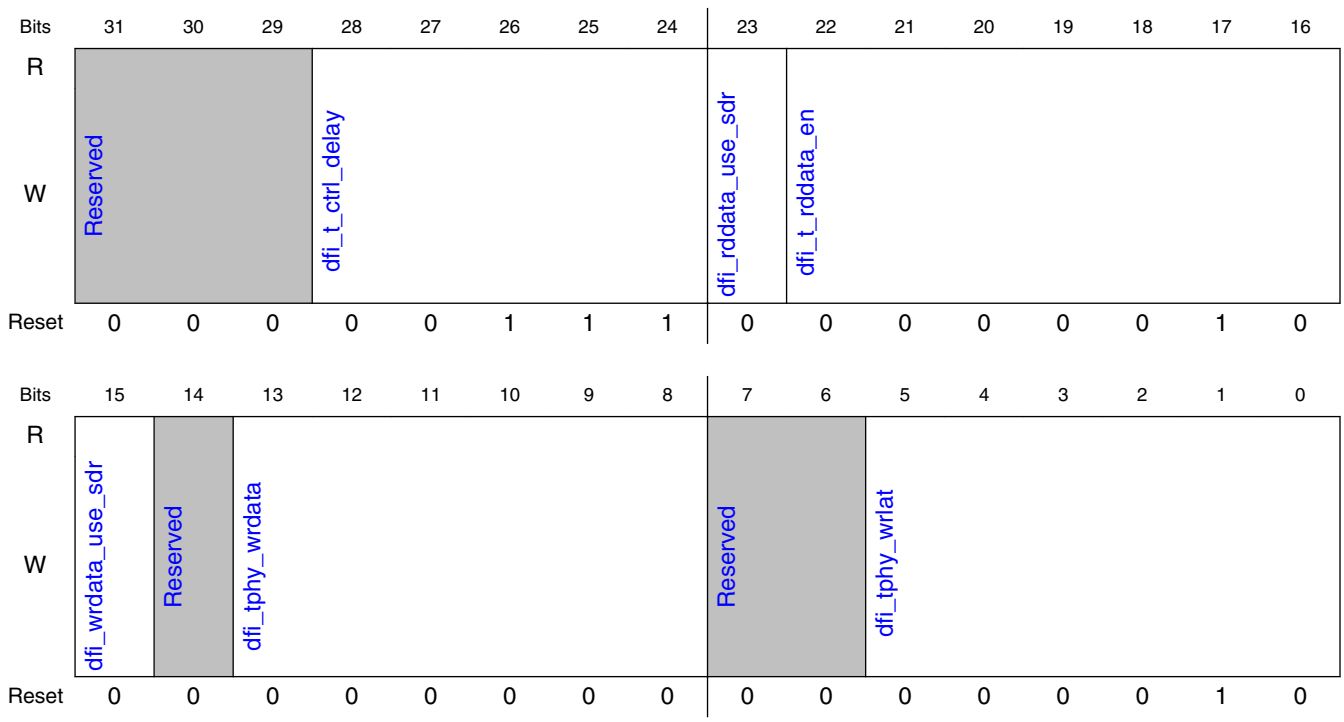
Field	Function
31 dis_auto_zq	- 1 - Disable DDRC generation of ZQCS/MPC(ZQ calibration) command. Register DBGCMD.zq_calib_short can be used instead to issue ZQ calibration request from APB module. - 0 - Internally generate ZQCS/MPC(ZQ calibration) commands based on ZQCTL1.t_zq_short_interval_x1024.
30 dis_srx_zqcl	1 - Disable issuing of ZQCL/MPC(ZQ calibration) command at Self-Refresh/SR-Powerdown exit. 0 - Enable issuing of ZQCL/MPC(ZQ calibration) command at Self-Refresh/SR-Powerdown exit.
29 zq_resistor_shared	- 1 - Denotes that ZQ resistor is shared between ranks. Means ZQinit/ZQCL/ZQCS/MPC(ZQ calibration) commands are sent to one rank at a time with tZQinit/tZQCL/tZQCS/tZQCAL/tZQLAT timing met between commands so that commands to different ranks do not overlap. - 0 - ZQ resistor is not shared.
28 dis_mpsmx_zqcl	- 1 - Disable issuing of ZQCL command at Maximum Power Saving Mode exit. Only applicable when run in DDR4 mode. - 0 - Enable issuing of ZQCL command at Maximum Power Saving Mode exit. Only applicable when run in DDR4 mode. This is only present for designs supporting DDR4 devices. Note: Do not issue ZQCL command at Maximum Power Save Mode exit if the DDRC_SHARED_AC configuration parameter is set. Program it to 1'b1. The software can send ZQCS after exiting MPSM mode.
27 —	Reserved
26-16 t_zq_long_nop	tZQoper for DDR4, tZQCAL for LPDDR4: Number of DFI clock cycles of NOP required after a ZQCL (ZQ calibration long)/MPC(ZQ Start) command is issued to SDRAM. When the controller is operating in 1:2 frequency ratio mode: DDR4: program this to tZQoper/2 and round it up to the next integer value. LPDDR4: program this to tZQCAL/2 and round it up to the next integer value.
15-10 —	Reserved
9-0 t_zq_short_nop	tZQCS for DDR4, tZQLAT for LPDDR4: Number of DFI clock cycles of NOP required after a ZQCS (ZQ calibration short)/MPC(ZQ Latch) command is issued to SDRAM. When the controller is operating in 1:2 frequency ratio mode, program this to tZQCS/2 and round it up to the next integer value.

9.2.5.1.127 [SHADOW] DFI Timing Register 0 (DFITMG0_SHADOW)

9.2.5.1.127.1 Offset

Register	Offset
DFITMG0_SHADOW	2190h

9.2.5.1.127.2 Diagram



9.2.5.1.127.3 Fields

Field	Function
31-29 —	Reserved
28-24 dfi_t_ctrl_delay	Specifies the number of DFI clock cycles after an assertion or de-assertion of the DFI control signals that the control signals at the PHY-DRAM interface reflect the assertion or de-assertion. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value.
23 dfi_rddata_use_sdr	Defines whether dfi_rddata_en/dfi_rddata/dfi_rddata_valid is generated using HDR (DFI clock) or SDR (DFI PHY clock) values. Selects whether value in DFITMG0.dfi_t_rddata_en is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles: - 0 in terms of HDR (DFI clock) cycles - 1 in terms of SDR (DFI PHY clock) cycles Refer to PHY specification for correct value.

Table continues on the next page...

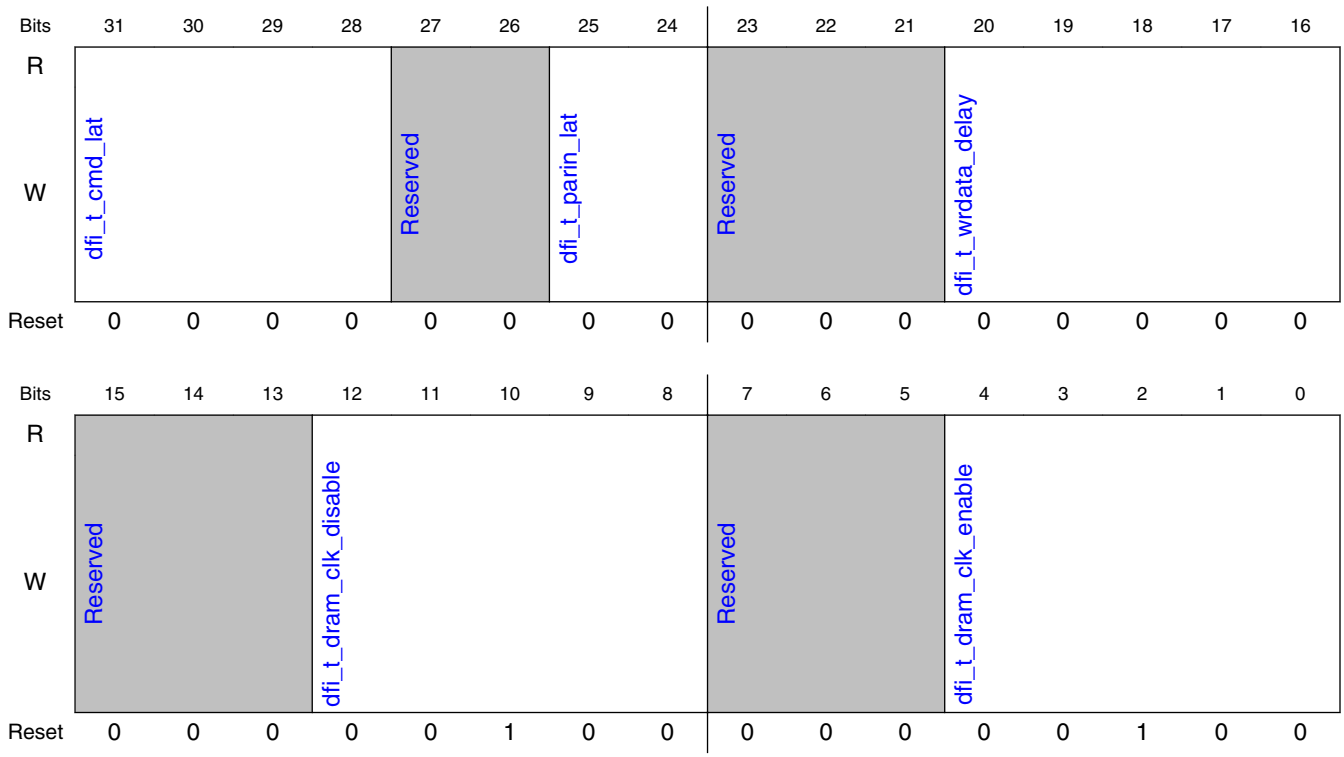
Field	Function
22-16 dfi_t_rddata_en	Time from the assertion of a read command on the DFI interface to the assertion of the dfi_rddata_en signal. Refer to PHY specification for correct value. This corresponds to the DFI parameter trddata_en.Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_rddata_use_sdr.
15 dfi_wrdata_use_sdr	Defines whether dfi_wrdata_en/dfi_wrdata/dfi_wrdata_mask is generated using HDR (DFI clock) or SDR (DFI PHY clock) values Selects whether value in DFITMG0.dfi_tphy_wrlat is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles Selects whether value in DFITMG0.dfi_tphy_wrdata is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles - 0 in terms of HDR (DFI clock) cycles - 1 in terms of SDR (DFI PHY clock) cycles Refer to PHY specification for correct value.
14 —	Reserved
13-8 dfi_tphy_wrdata	Specifies the number of clock cycles between when dfi_wrdata_en is asserted to when the associated write data is driven on the dfi_wrdata signal. This corresponds to the DFI timing parameter tphy_wrdata. Refer to PHY specification for correct value. Note, max supported value is 8. Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_wrdata_use_sdr.
7-6 —	Reserved
5-0 dfi_tphy_wrlat	Write latency Number of clocks from the write command to write data enable (dfi_wrdata_en). This corresponds to the DFI timing parameter tphy_wrlat. Refer to PHY specification for correct value.Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_wrdata_use_sdr.

9.2.5.1.128 [SHADOW] DFI Timing Register 1 (DFITMG1_SHADOW)

9.2.5.1.128.1 Offset

Register	Offset
DFITMG1_SHADOW	2194h

9.2.5.1.128.2 Diagram



9.2.5.1.128.3 Fields

Field	Function
31-28 dfi_t_cmd_lat	Specifies the number of DFI PHY clock cycles between when the dfi_cs signal is asserted and when the associated command is driven. This field is used for CAL mode, should be set to '0' or the value which matches the CAL mode register setting in the DRAM. If the PHY can add the latency for CAL mode, this should be set to '0'. Valid Range: 0, 3, 4, 5, 6, and 8
27-26 —	Reserved
25-24 dfi_t_parin_lat	Specifies the number of DFI PHY clock cycles between when the dfi_cs signal is asserted and when the associated dfi_parity_in signal is driven.
23-21 —	Reserved
20-16 dfi_t_wrdata_delay	Specifies the number of DFI clock cycles between when the dfi_wrdata_en signal is asserted and when the corresponding write data transfer is completed on the DRAM bus. This corresponds to the DFI timing parameter twrdata_delay. Refer to PHY specification for correct value. For DFI 3.0 PHY, set to twrdata_delay, a new timing parameter introduced in DFI 3.0. For DFI 2.1 PHY, set to tphy_wrdata + (delay of DFI write data to the DRAM). Value to be programmed is in terms of DFI clocks, not PHY clocks. In FREQ_RATIO=2, divide PHY's value by 2 and round up to next integer. If using DFITMG0.dfi_wrdata_use_sdr=1, add 1 to the value. Unit: Clocks
15-13 —	Reserved

Table continues on the next page...

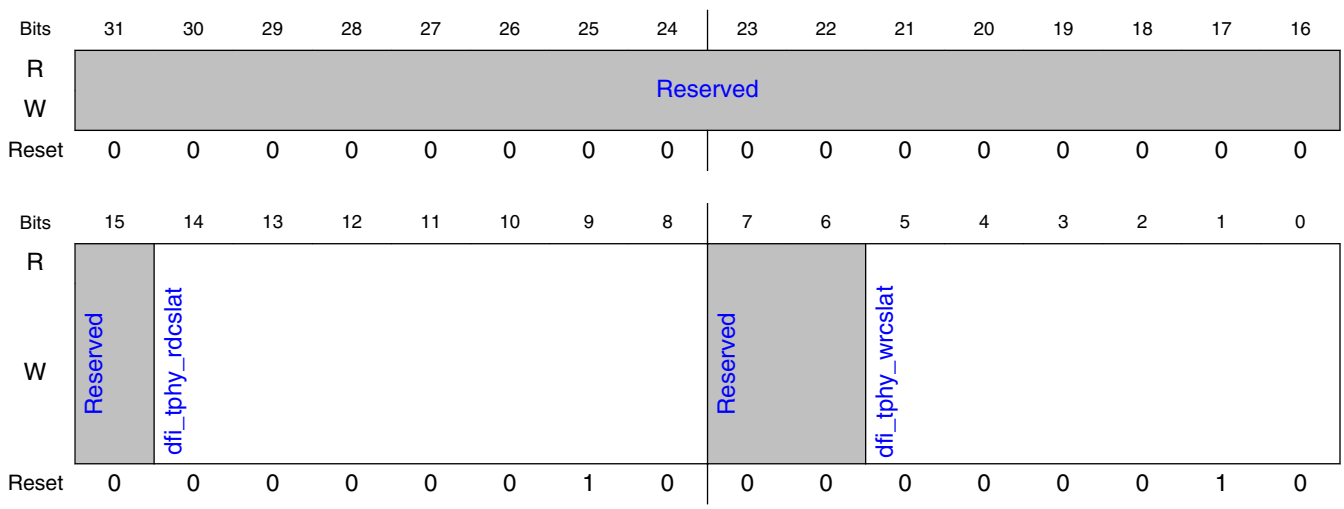
Field	Function
12-8 dfi_t_dram_clk_disable	Specifies the number of DFI clock cycles from the assertion of the dfi_dram_clk_disable signal on the DFI until the clock to the DRAM memory devices, at the PHY-DRAM boundary, maintains a low value. If the DFI clock and the memory clock are not phase aligned, this timing parameter should be rounded up to the next integer value.
7-5 —	Reserved
4-0 dfi_t_dram_clk_enable	Specifies the number of DFI clock cycles from the de-assertion of the dfi_dram_clk_disable signal on the DFI until the first valid rising edge of the clock to the DRAM memory devices, at the PHY-DRAM boundary. If the DFI clock and the memory clock are not phase aligned, this timing parameter should be rounded up to the next integer value.

9.2.5.1.129 [SHADOW] DFI Timing Register 2 (DFITMG2_SHADOW)

9.2.5.1.129.1 Offset

Register	Offset
DFITMG2_SHADOW	21B4h

9.2.5.1.129.2 Diagram



9.2.5.1.129.3 Fields

Field	Function
31-15 —	Reserved

Table continues on the next page...

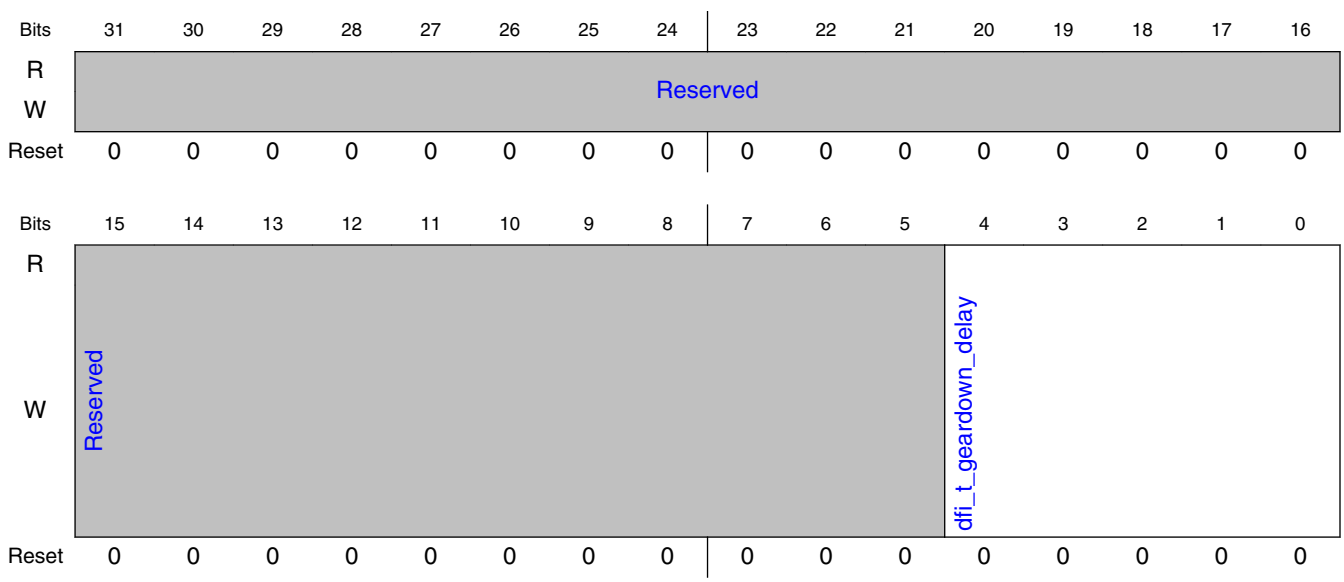
Field	Function
14-8 dfi_tphy_rdcslat	Number of DFI PHY clock cycles between when a read command is sent on the DFI control interface and when the associated dfi_rddata_cs signal is asserted. This corresponds to the DFI timing parameter tphy_rdcslat. Refer to PHY specification for correct value.
7-6 —	Reserved
5-0 dfi_tphy_wrcslat	Number of DFI PHY clock cycles between when a write command is sent on the DFI control interface and when the associated dfi_wrdata_cs signal is asserted. This corresponds to the DFI timing parameter tphy_wrcslat. Refer to PHY specification for correct value.

9.2.5.1.130 [SHADOW] DFI Timing Register 3 (DFITMG3_SHADOW)

9.2.5.1.130.1 Offset

Register	Offset
DFITMG3_SHADOW	21B8h

9.2.5.1.130.2 Diagram



9.2.5.1.130.3 Fields

Field	Function
31-5	Reserved

Table continues on the next page...

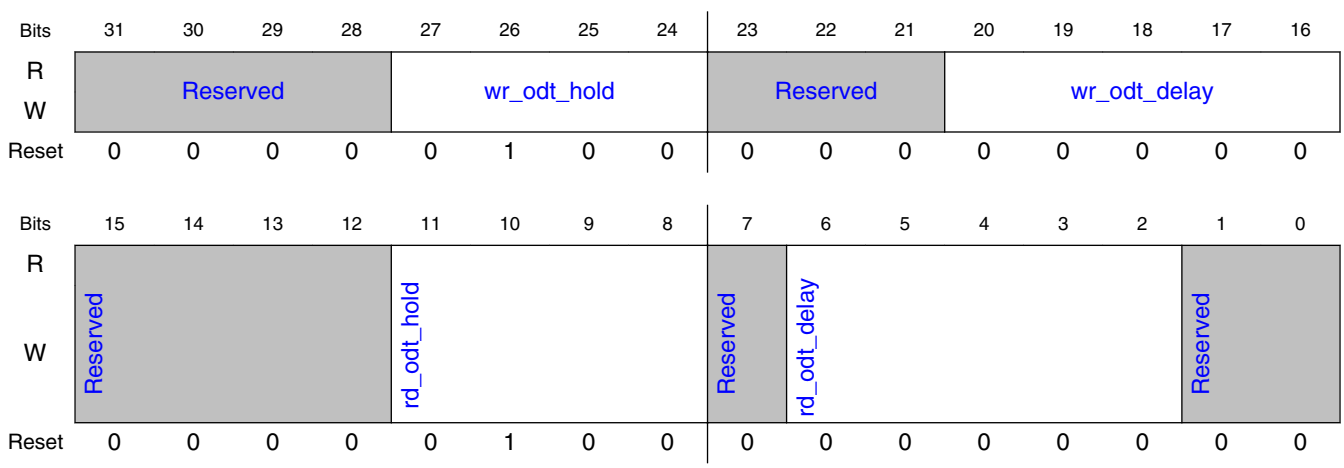
Field	Function
—	
4-0 dfi_t_geardown_delay	The delay from dfi_geardown_en assertion to the time of the PHY being ready to receive commands. Refer to PHY specification for correct value. When the controller is operating in 1:2 frequency ratio mode, program this to (tgeardown_delay/2) and round it up to the next integer value. Unit: Clocks

9.2.5.1.131 [SHADOW] ODT Configuration Register (ODTCFG_SHADOW)

9.2.5.1.131.1 Offset

Register	Offset
ODTCFG_SHADOW	2240h

9.2.5.1.131.2 Diagram



9.2.5.1.131.3 Fields

Field	Function
31-28 —	Reserved
27-24 wr_odt_hold	DFI PHY clock cycles to hold ODT for a write command. The minimum supported value is 2. DDR4: - BL8: 5 + WR_PREAMBLE + CRC_MODE WR_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble) CRC_MODE = 0 (not CRC mode), 1 (CRC mode).
23-21 —	Reserved

Table continues on the next page...

Field	Function
20-16 wr_odt_delay	The delay, in DFI PHY clock cycles, from issuing a write command to setting ODT values associated with that command. ODT setting must remain constant for the entire time that DQS is driven by the DDRC. Recommended values: DDR4: - DFITMG1.dfi_t_cmd_lat (to adjust for CAL mode).
15-12 —	Reserved
11-8 rd_odt_hold	DFI PHY clock cycles to hold ODT for a read command. The minimum supported value is 2. Recommended values: DDR4: - BL8: 5 + RD_PREAMBLE RD_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble)
7 —	Reserved
6-2 rd_odt_delay	The delay, in DFI PHY clock cycles, from issuing a read command to setting ODT values associated with that command. ODT setting must remain constant for the entire time that DQS is driven by the DDRC. Recommended values: DDR4: - CL - CWL - RD_PREAMBLE + WR_PREAMBLE + DFITMG1.dfi_t_cmd_lat (to adjust for CAL mode) WR_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble) RD_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble). If (CL - CWL - RD_PREAMBLE + WR_PREAMBLE) < 0, DDRC does not support ODT for read operation.
1-0 —	Reserved

9.3 DDR BLK_CTRL

9.3.1 Overview

The DDRMIX domain contains a control register known as DDR Block Control (DDR BLK_CTRL). This register controls the DDR device selection for supported DDR devices.

9.3.1.1 Block Diagram

The DDR BLK_CTRL interface connects to the DDRMIX domain peripherals. A block diagram illustrating the DDRMIX domain is shown below.

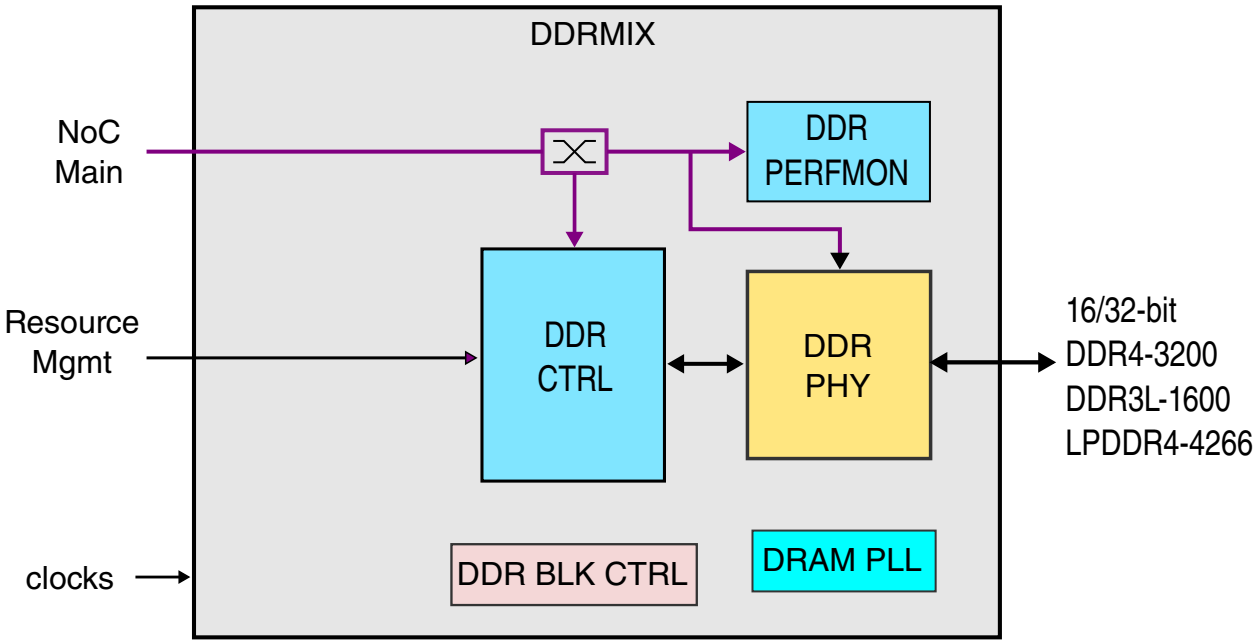


Figure 9-4. DDRMIX Block Diagram

9.3.2 Memory Map and register definition

This section includes the DDR BLK_CTRL module memory map and detailed descriptions of all registers.

9.3.2.1 register descriptions

9.3.2.1.1 memory map

DDR_BLK_CTRL base address: 3D00_0000h

Offset	Register	Width (In bits)	Access	Reset value
100h	DDR Subsystem General Purpose Register 0 (DDR_SS_GPR0)	32	RW	0000_0000h

9.3.2.1.2 DDR Subsystem General Purpose Register 0 (DDR_SS_GPR0)

9.3.2.1.2.1 Offset

Register	Offset
DDR_SS_GPR0	100h

9.3.2.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DDR_MODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DDR_MODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.3.2.1.2.3 Fields

Field	Function
31-0 DDR_MODE	DDR Mode 00000000000000000000000000000000b - DDR4/DDR4L Mode 00000000000000000000000000000001b - LPDDR4 Mode

9.4 DDR PHY (DDR_PHY)

9.4.1 Overview

The Double Data Rate Physical Layer (DDR PHY), provides an interface between universal controllers and external DDR4/LPDDR4 SDRAM devices.

NOTE

The information within this block guide is Synopsys Proprietary. Used with permission.

9.4.1.1 Block Diagram

The block diagram of the DDR PHY is provided below.

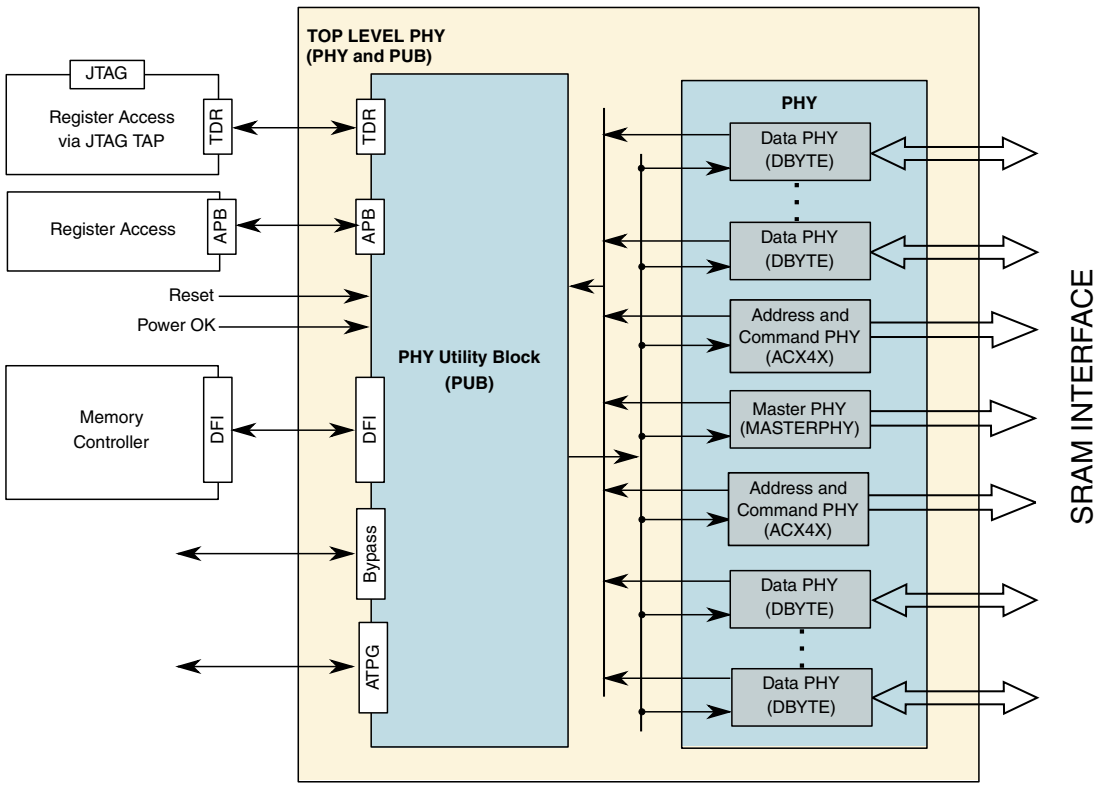


Figure 9-5. DDR PHYBlock Diagram

9.4.1.2 Features

The following features are supported in the DDR PHY:

- DDR4 and LPDDR4 operation
- Multiple memory rank support:
  - Four ranks supported for DDR4
  - Two ranks supported for LPDDR4
- DFI interface (DFI v4.0 Addendum 2, March 2015)
- At-speed loopback testing on both the address and data channels
- Mux-scan ATPG
- High-speed PLL used to generate the high speed clock for data transmit and high speed digital pipelines
- PHY Utility Block (PUB), which provides configuration and control
- Power on reset (POR) circuit, DRAM reset, DRAM alert and PHY debug
- A global voltage reference generation and observation circuit
- Impedance calibration circuits

9.4.2 Functional Description

The DDR PHY is composed of the following:

- DBYTE - The eight-lane data IO unit
- ACX4 - The four-lane address/command hard IP unit
- MASTERPHY - Contains the PLL, the voltage reference, thermal calibration, and reset
- PUB - PHY Utility Block

9.4.2.1 DDR Address/Command PHY (ACX4)

The DDR SDRAM address/command PHY (DDRPHYACX4) provides an address and command interface to the external SDRAM memories. A memory interface typically contains a single address/command PHY and one or more data PHYs (one for each byte lane).

The AC consists of multiple design slices. The address/command (AC) slice is used for all address and command and clk/cs/odt signals. Each slice contains all logic required for both the output (address/command) and input (loopback) paths of the respective signals.

All clocking inside the AC is derived from an input clock, which is generated by the PLL in the MASTERPHY. The PLL generates the higher frequency clocks that are is used inside the macro and minimizes input clock jitter.

9.4.2.1.1 ACX4 Transmit Lanes

The ACX4 component contains 4 transmit only lanes which are grouped as follows:

- Clock group: {CK_t, CK_c}
- CA group: {ras_n, cas_n, we_n, addr…}
- CKE,CS,ODT group{CKE, CS_n, ODT…}

In DDR4 systems, 12 ACs are required to control one 72b/64b channel with four independent ranks.

In LPDDR4 systems, the 12 ACs are broken up into two independently controllable groups, enabling up to two independent 32b channels (each with a maximum of two ranks.

9.4.2.1.2 ACX4 Command Path

Each SDRAM address and command signal is driven from the memory controller to the SDRAM through an address/command slice. Controlled address/command pipeline logic inside the AC slice converts the DFI-rate signals from the controller to single data rate (SDR) Address/Command signals going to the SDRAM. The last launching register inside the address/command slice is clocked by a delay-shifted DDR clock. For SDR operation, all outputs remain constants for two cycles at a time.

The path for the SDRAM clock (CK) is through a standard AC slice which is configured to act as a dedicated clock (CK) slice. This shifted address/command output makes the CK centered into the data eye of the address/command signals.

All AC outputs going to the SDRAM have a LCDL, which can be used to align or shift the outputs of the data PHY. This delay element can also be used to provide delay adjustment during functional or loopback modes.

9.4.2.1.3 ACX4 Loopback/DFT Paths

For every lane there will be a loopback-only receive path. The loopback path is stimulated and checked by utilizing a FirmWare image.

9.4.2.2 DDR Data PHY (DBYTE)

The DDR SDRAM Data PHY (DBYTE) provides data interface to external SDRAM memory. Features of the DBYTE include:

- High speed digital logic pipeline for transmit datapaths to the SDRAM
- High speed digital logic pipeline for receive datapaths from the SDRAM
- Drivers and receivers for the DQ and DQS signals as well as all the necessary high voltage analog macros to interface off chip
- I/O component that includes PVT-compensated on-die termination (ODT) and output impedance
- Loop-back test mode and PRBS checkers to help verify the integrity of the internal datapath of all DBYTE I/O lanes

9.4.2.2.1 DBYTE Write Path

Each SDRAM data is driven from the PUB to the SDRAM through a data (DQ) slice. Controlled data pipeline logic inside the DBYTE slice converts the DFI-rate signals from the PUB to double-data rate (DDR) signals going to the SDRAM. The last launching register inside the data slice is clocked by the DDR clock.

The path for the SDRAM write strobe is through a dedicated data strobe (DQS) slice. The output register of the DQS slice is clocked by a delayed version of PclkIn. PclkIn drives the output register in the DQ slice is shifted using the DLL in training mode. This centers the write data strobe signal in the write data eye.

All DBYTE outputs going to the SDRAM, including write data, write data strobe read DQS gate, driver power down, receiver power down, on-die termination enable, and output enable have a DLL that is used to deskew the outputs in order to maximize the write data eye.

Apart from the DQS pipeline, the DQS slice also contains a similar pipeline for output enables, on-die termination enable, driver power down, and receiver power down. This matches the I/O control signals to the corresponding data and data strobe.

9.4.2.2.2 DBYTE Read and Loopback Path

Read data from the SDRAM is written into a four-deep asynchronous FIFO that is implemented inside the DQ slice.

NOTE

The FIFO is 16 deep in terms of the DRAM DQS writing into the FIFO; it is 8 deep on the core side when a half-speed DFICLK is employed and it is 8 deep on the core side when a full speed DFICLK is employed.

The FIFO is written using the DQS clock and read using the Controller clock. A per-bit delay line at the input of the DQ slice is used to optionally de-skew read data with respect to the read strobe in order to maximize the read data eye.

The read data strobe path and a read DQS gate path is implemented inside the DQS slice. A delay line at the input of the slice shifts the read DQS by nominal 90 degrees (with the actual value arrived at through training) so that the strobe is centered into the read data eye before clocking the data into the FIFO. A Bit- Wise Delay Line (BDL) in the read DQS path is used to match a corresponding bit de-skew BDL in the DQ slice.

The DQS slice also generates the write pointers for the data FIFO. These pointers, together with the strobe, are then distributed to all DQ slices. The PUB generates a trained read-valid signal to read the data from the PHY.

The DBI uses DQS Slice 1, configured to behave as a DQ slice in x8 mode only.

9.4.2.2.2.1 DBYTE Loopback/DFT Paths

High speed loopback is implemented on all DBYTE I/Os. The DBYTE read path also doubles as the loopback path. During loopback, the PUB enables the gating of the DQS signal in the same way as it would normally gate the DQS during reads. This allows the looped back data to be clocked into the read data FIFO using the looped back DQS signal.

9.4.2.2.2.2 Memory Controller based DBYTE Loopback

This high speed PHY loopback mode uses all the DBYTE control and data paths in mission mode like operation modes. In this mode, the memory controller issues simultaneous Read/Write commands to the PHY, resulting in transmitting and receiving the same data on each DQ pad.

9.4.2.3 Master PHY (MASTERPHY)

The Master block contains the PLL, the Primary Voltage Reference, thermal calibration, reset, and I/O cells for test and control.

9.4.2.3.1 PLL

The PLL in the MASTER is a simplified PLL that has DFICLK as an input and creates an internal clock by quadrupling the DFICLK frequency. The PLL in the MASTER can only multiply the input clock frequency to produce an output clock that is four times the frequency of the input clock. The internal clock is the output of the MASTER to the ACX4 and DBYTE.

9.4.2.3.1.1 Optimal PLL Settings for Mission Mode Frequency Bins

Table 9-4. Optimal PLL Settings for Mission Mode Frequency Bins (PLL Max Bandwidth = 2 MHz)

pllin min (MHz)	pllin max (MHz)	Input div. ratio	Fbk min (MHz)	Fbk max (MHz)	Fbk div. ratio	VCO min (MHz)	VCO max (MHz)	PllCtrl2 PllFreqSel[4:0]	PLLTestMode int_half[8] prop_half[5] vco_mode[2] overflow[1]	PllCtrl1 PllCpPropCtrl[8:5]	PllCtrl1 PllCpIntCtrl[4:0]	PllCtrl4 PllCpPropGsCtrl[8:5]	PllCtrl4 PllCpIntGsCtrl[4:0]
937.5	1067	8	117.2	133.4	4	3750	4268	11000 ’b	1011’b	0011’b	00000 ’b	1011’b	11111 ’b
625	937.5	8	78.13	117.2	4	2500	3750	11001 ’b	1011’b	0100’b	00000 ’b	1011’b	11111 ’b
468.75	625	4	117.2	156.3	8	3750	5000	01010 ’b	1011’b	0011’b	00000 ’b	1011’b	11111 ’b
312.5	468.75	4	78.13	117.2	8	2500	3750	01011 ’b	1011’b	0100’b	00000 ’b	1011’b	11111 ’b
234.4	312.5	2	117.2	156.3	16	3750	5000	00110 ’b	1011’b	0011’b	00000 ’b	1011’b	11111 ’b
166	234.4	2	83	117.2	16	2656	3750	00111 ’b	1011’b	0100’b	00000 ’b	1011’b	11111 ’b

9.4.2.3.2 Voltage Reference

This primary voltage reference generates the VREF global out used by the receivers in the ACX4 and DBYTE.

9.4.2.3.3 Calibration

The MASTER contains the analog blocks to perform output drive and input termination impedance calibration. Through the use of a digital control state machine, located in the PUB, the full mixed-signal calibration loop matches the impedance of the driver and termination with that of an external precision resistor. The resulting digital calibration code is fed to the address/command, DQ, and DQS blocks to control their impedance.

9.4.2.3.4 Reset

Reset takes the asynchronous reset bump, the PwrOk bump, and dfi_reset_n signal and combines them and synchronizes to the DFI clock. This is used by the ACX4 and DBYTE.

9.4.2.4 PHY Utility Block (PUB)

The following is an overview of the different functionality of the PHY Utility Block (PUB):

- Registers required to configure, control, train and self-test the PHY. Also contains registers required to initialize the DRAM.
- Initialization - this includes locking of the PLL, delay line calibration and calibration of I/O impedances.
- Data Training- train the PHY for optimum timing margins. This includes finding the best positioning of the DQS gating window during reads, write leveling, data bit deskew, and optimizing the read and write data eye.
- Implements a DFI-compliant interface between the external controller and the PHY.

9.4.2.4.1 DDR PHY Control and Configuration

The DDR PHY's PUB registers provide the required configuration, control, training and self-testing of the PHY. Also contains registers required to initialize the DRAM.

Refer to the DDR PHY register descriptions for more information.

9.4.2.4.2 DFI Interface

The DDR PHY and the DDR Controller both require a standard interface to transfer address, data, and control between these two blocks. These interfaces are DFI compliant. The PUB contains the DFI interface for the PHY.

For more information regarding the DFI specification, refer to the following:

- DDR PHY Interface (DFI) Specification, Revision 3.1 (Preliminary), 21, March 2014
- DDR PHY Interface (DFI) Specification Preliminary DFI 4.0 Specification - Addendum to DFI 3.1, 2 April 2014

9.4.2.5 Clocks

The following table provides the clocks for the DDR PHY module.

Table 9-5. Clocks

Clock	Description
DFICLK	DFI Clock
APBCLK	APB Clock
DfiCtlClk	DFI Controller Clock
BypassPclk	PHY High speed Bypass Clock

9.4.3 External Signals

The following table provides the external signals for the DDR PHY module.

Table 9-6. External Signals

Signal	Description
ACn	Address and Command signals
ALERT_N	DDR4 alert
DMn	Data mask
DQn	Data
DQSn_N/P	Data strobe
RESET_N	Reset
VREF	Voltage reference for receivers
ZN	Calibration external reference resistor

9.4.4 Initialization and Training Sequence

The following outlines the general initialization and training sequence. Unless otherwise stated, these functions occur within the DDR PHY.

1. Bring up VDD, VDDQ, and VAA. This occurs outside of the DDR PHY.
2. Start the clocks and reset the PHY. This also occurs outside of the PHY.

3. Initialize PHY configuration.
4. Load the PHY training firmware into instruction memory SRAM.
5. Set the PHY input clocks to the desired frequency. This occurs outside of the PHY.
6. Write the Message Block parameters for the training firmware.There are several locations in the data memory that must be written for the training firmware to have these parameters.
7. Execute the training firmware.
8. Once the training firmware completes, the results are returned to the data memory in the message block structure. This message block can be read. The result may be that another frequency needs to be trained (return to step 5).
9. Load the PHY Initialization Engine registers with the provided initialization sequence.
10. Initialize the PHY to mission mode by performing a DFI initialization sequence per the DFI specification.
11. The PHY is now in mission mode.

9.4.5 Memory Map and Register Definition

This section includes the DDRC PHY module memory map and detailed descriptions of all registers.

NOTE

The information within this block guide is Synopsys Proprietary. Used with permission.

NOTE

Only LPDDR4 and DDR4 operation is supported.

9.4.5.1 DDRPHYA_DBYTE register descriptions

9.4.5.1.1 DDRPHYA_DBYTE Memory map

DDR PHY address block

DDRPHYA_DBYTE0 base address: 3C04_0000h

DDRPHYA_DBYTE1 base address: 3C04_4000h

DDRPHYA_DBYTE2 base address: 3C04_8000h

DDRPHYA_DBYTE3 base address: 3C04_C000h

Offset	Register	Width (In bits)	Access	Reset value
280h - 29Ch	Maps Phy DQ lane to memory DQ0 (Dq0LnSel - Dq7LnSel)	16	RW	0000h

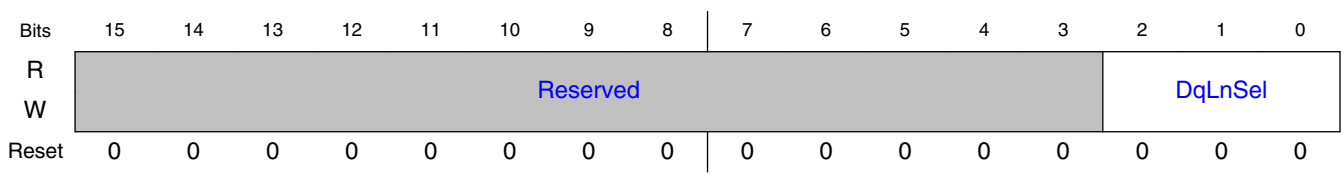
9.4.5.1.2 Maps Phy DQ lane to memory DQ0 (Dq0LnSel - Dq7LnSel)

9.4.5.1.2.1 Offset

For n = 0 to 7:

Register	Offset
DqnLnSel	140h + (n × 2h)

9.4.5.1.2.2 Diagram



9.4.5.1.2.3 Fields

Field	Function
15-3 —	Reserved
2-0 DqLnSel	Supports mapping of PHY dq to dram dq within a byte (swizzle). Supports mapping of PHY dq to dram dq within a byte (swizzle). Intended to undo the swizzle of board-level phy-to-dram connections. such that MRR operations of binary counters may be return correct values. Note that this register is per-dbyte such that the sizzle may be different per dbyte. Each register in a byte's set of DqLnSel must have a unique value within the set. For example, if, on this dbyte, PHY lane 3 is connected to memory dq0 (on this dbyte), then Register Dq0LnSel for this dbyte should be 3.

9.5 AHB-to-APBH Bridge with DMA (APBH-Bridge-DMA)

9.5.1 Overview

The AHB-to-APBH bridge provides the chip with an inexpensive peripheral attachment bus running on the AHB's HCLK. (The H in APBH denotes that the APB*H* is synchronous to HCLK .)

As shown in the figure below, the AHB-to-APBH bridge includes the AHB-to-APB PIO bridge for a memory-mapped I/O to the APB devices, as well as a central DMA facility for devices on this bus and a vectored interrupt controller for the ARM core. Each one of the APB peripherals, including the vectored interrupt controller, is documented in their respective chapters.

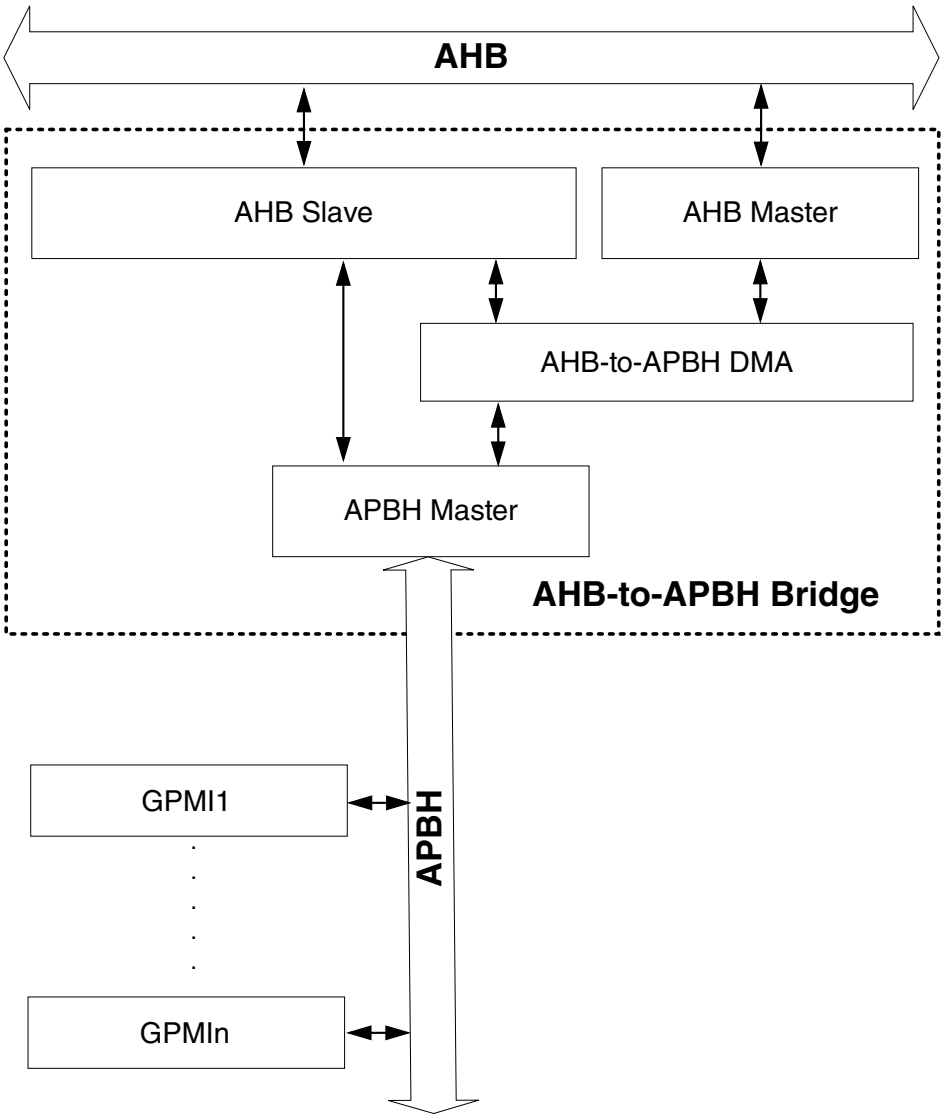


Figure 9-6. AHB-to-APBH Bridge DMA Block Diagram

The DMA controller uses the APBH bus to transfer read and write data to and from each peripheral. There is no separate DMA bus for these devices. Contention between the DMA's use of the APBH bus and the AHB-to-APB bridge functions' use of the APBH is mediated by an internal arbitration logic. For contention between these two units, the DMA is favored and the AHB slave will report "not ready" through its HREADY output until the bridge transfer can complete. The arbiter tracks repeated lockouts and inverts the priority, guaranteeing the ARM platform every fourth transfer on the APB.

9.5.2 Functional Description

9.5.2.1 Clocks

The table found here describes the clock sources for APBH. Please see clock control block for clock setting, configuration and gating information.

Table 9-7. APBH Clocks

Clock name	Description
hclk	Module clock

9.5.2.2 APBH DMA

The DMA supports four channels of DMA services, as shown in the following table. The shared DMA resource allows each independent channel to follow a simple chained command list. Command chains are built up using the general structure, as shown in Figure 9-7.

Table 9-8. APBH DMA channel assignments

APBH DMA Channel #	Usage
0	GPMI0
1	GPMI1
2	GPMI2
3	GPMI3

A single command structure or channel command word specifies a number of operations to be performed by the DMA in support of a given device. Thus, the ARM platform can set up large units of work, chaining together many DMA channel command words, pass them off to the DMA, and have no further concern for the device until the DMA completion interrupt occurs. The goal is to have enough intelligence in the DMA and the devices to keep the interrupt frequency from any device below 1 KHz (arrival intervals longer than 1 ms).

A single command structure can issue 32-bit PIO write operations to key registers in the associated device using the same APB bus and controls that it uses to write DMA data bytes to the device. For example, this allows a chain of operations to be issued to the GPMI controller to send NAND command bytes, address bytes, and data transfers where the command and the address structure is completely under software control, but the

administration of that transfer is handled autonomously by the DMA. Each DMA structure can have 0–15 PIO words appended to it. The CMDPIOWORDS field, if non-zero, instructs the DMA engine to copy these words to the APB, beginning at the first register address offset for the peripheral and and incrementing the register offset each cycle.

The DMA master generates only normal read/write transfers to the APBH. It does *not* generate set, clear, or toggle (SCT) transfers.

After any requested PIO words have been transferred to the peripheral, the DMA examines the two-bit command field in the channel command structure. Table 9-9 shows the four commands implemented by the DMA.

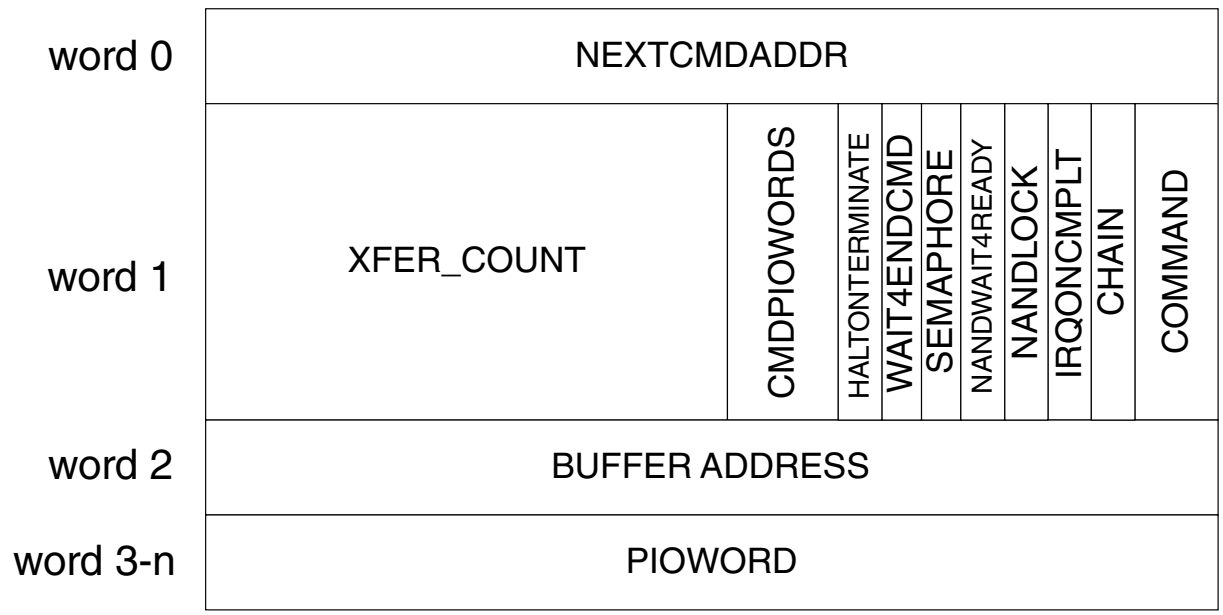


Figure 9-7. AHB-to-APBH Bridge DMA channel command structure

Table 9-9. APBH DMA commands

DMA Command	Usage
00	NO_DMA_XFER. Perform any requested PIO word transfers, but terminate the command before any DMA transfer.
01	DMA_WRITE. Perform any requested PIO word transfers, then perform a DMA transfer from the peripheral for the specified number of bytes.
10	DMA_READ. Perform any requested PIO word transfers and then perform a DMA transfer to the peripheral for the specified number of bytes.
11	DMA_SENSE. Perform any requested PIO word transfers, then perform a conditional branch to the next chained device. Follow the NEXTCMD_ADDR pointer if the peripheral sense is false. Follow the BUFFER_ADDRESS as a chain pointer if the peripheral sense line is true. This command becomes a no-operation for any channel other than a GPMI channel.

DMA_WRITE operations copy data bytes to the system memory (on-chip RAM or SDRAM) from the associated peripheral.

DMA_READ operations copy data bytes to the APB peripheral from the system memory. The DMA engine contains a shared byte aligner that aligns bytes from system memory to or from the peripherals. Peripherals always assume little-endian-aligned data arrives or departs on their 32-bit APB. The DMA_READ transfer uses the BUFFER_ADDRESS word in the command structure to point to the DMA data buffer to be read by the DMA_READ command.

The NO_DMA_XFER command is used to write PIO words to a device without performing any DMA data byte transfers. This command is useful in such applications as activating the NAND devices CHECKSTATUS operation. The check status command reads a status byte from the NAND device, performs an XOR and MASK against an expected value supplied as part of the PIO transfer. Once the read check completes (see NAND Read Status Polling Example), the NO_DMA_XFER command completes. The result in the peripheral is that its sense line is driven by the results of the comparison. The sense flip-flop is only updated by CHECKSTATUS for the device that is executed. At some future point, the chain contains a DMA command structure with the fourth and final command value, that is, the DMA_SENSE command.

As each DMA command completes, it triggers the DMA to load the next DMA command structure in the chain. The normal flow list of DMA commands is found by following the NEXTCMD_ADDR pointer in the DMA command structure. The DMA_SENSE command uses the DMA buffer pointer word of the command structure to point to an alternate DMA command structure chain or list. The DMA_SENSE command examines the sense line of the associated peripheral. If the sense line is false, then the DMA follows the standard list found whose next command is found from the pointer in the NEXTCMD_ADDR word of the command structure. If the sense line is true, then the DMA follows the alternate list whose next command is found from the pointer in the DMA Buffer Pointer word of the DMA_SENSE command structure (see Figure 9-7). The sense command ignores the CHAIN bit, so that both pointers must be valid when the DMA comes to a sense command.

If the wait-for-end-command bit (WAIT4ENDCMD) is set in a command structure, the DMA channel waits for the device to signal completion of a command by toggling the endcmd signal before proceeding to load and execute the next command structure. Then, if DECREMENT_SEMAPHORE is set, the semaphore is decremented after the end command is seen.

A detailed bit-field view of the DMA command structure is shown in the following table, which shows a field that specifies the number of bytes to be transferred by this DMA command. The transfer-count mechanism is duplicated in the associated peripheral, either as an implied or as a specified count in the peripheral.

Table 9-10. DMA channel command word in system memory

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
NEXT_COMMAND_ADDRESS																															
Number DMA Bytes to Transfer																Number PIO Words to Write							HALTONTERMINATE	WAIT4ENDCMD	DECREMENT SEMAPHORE	NANDwAIT4READY	NANDLOCK	IRQ_COMPLETE	CHAIN	COMMAND	
DMA Buffer or Alternate CCW																															
Zero or More PIO Words to Write to the Associated Peripheral Starting at its Base Address on the APBH Bus																															

Figure 9-7 also shows the CHAIN bit in bit 2 of the second word of the command structure. This bit is set to 1, if the NEXT_COMMAND_ADDRESS contains a pointer to another DMA command structure. If a null pointer (0) is loaded into the NEXT_COMMAND_ADDRESS, it is not detected by the DMA hardware. Only the CHAIN bit indicates whether a valid list exists beyond the current structure.

If the IRQ_COMPLETE bit is set in the command structure, then the last act of the DMA before loading the next command is to set the interrupt-status bit corresponding to the current channel. The sticky interrupt request bit in the DMA CSR remains set until cleared by the software. It can be used to interrupt the ARM platform.

The NAND_LOCK bit is monitored by the DMA channel arbiter. Once a NAND channel (from channel 0 to channel 3) succeeds in the arbiter with its NAND_LOCK bit set, then the arbiter ignores the other NAND channels until a command is completed in which the NAND_LOCK is not set. Notice that the semantic here is that the NAND_LOCK state is to limit scheduling of a non-locked DMA. A DMA channel can go from unlocked to locked in the arbiter at the beginning of a command when the NAND_LOCK bit is set. When the last DMA command of an atomic sequence is completed, the lock should be removed. To accomplish this, the last command does not have the NAND_LOCK bit. It is still locked in the atomic state within the arbiter when the command starts, so that it is the only NAND command that can be executed. At the end, it drops from the atomic state within the arbiter.

The NAND_WAIT4READY bit also has a special use for GPMI channels (from channel 0 to channel 3), i.e., the NAND device channels. The GPMI peripheral supplies a sample of the ready line from the NAND device. This ready value is used to hold off of a command with this bit set until the ready line is asserted to 1. Once the arbiter sees a command with a wait-for-ready set, it holds off that channel until ready is asserted.

Receiving an IRQ for HALTONTERMINATE (HOT) is a feature in the APBH DMA descriptor that allows GPMI to signal to the DMA engine that an error has occurred. If a commmand is stalled due to an error, a HOT signal is sent from the peripheral to the DMA engine and causes an IRQ after terminating the DMA descriptor being executed.

Therefore, it is recommended that software use this signal as follows:

- Always set HALTONTERMINATE to 1 in a DMA descriptor. That way, if a peripheral signals HOT, the transfer will end, leaving the peripheral block and the DMA engine synchronized (but at the end of a command).
- When an IRQ from an APBH channel is received, and the IRQ is determined to be due to an error (as opposed to an IRQONCOMPLETE interrupt) the software should:
  - Reset the channel.
  - Determine the error from error reporting in the peripheral block, then manage the error in the peripheral that is attached to that channel in whatever appropriate way exists for that device (software recovery, device reset, block reset, etc).

Each channel has an eight-bit counting semaphore that controls whether it is in the idle state. When the semaphore is non-zero, the channel is ready to run, process commands and perform DMA transfers. Whenever a command finishes its DMA transfer, it checks the DECREMENT_SEMAPHORE bit. If set, it decrements the counting semaphore. If the semaphore goes to 0 as a result, then the channel enters the idle state and remains there until the semaphore is incremented by the software. When the semaphore goes to non-zero and the channel is in its idle state, then it uses the value in the APBH_CHn_NXTCMDAR register (next command address register) to fetch a pointer to the next command to process.

NOTE

This is a double indirect case. This method allows the software to append to a running command list under the protection of the counting semaphore.

To start processing the first time, software creates the command list to be processed. It writes the address of the first command into the APBH_CHn_NXTCMDAR register, and then writes 1 to the counting semaphore in APBH_CHn_SEMA. The DMA channel loads APBH_CHn_CURCMDAR register and then enters the normal state machine processing

for the next command. When the software writes a value to the counting semaphore, it is added to the semaphore count by hardware, protecting the case where both hardware and software are trying to change the semaphore on the same clock edge.

Software can examine the value of APBH_CHn_CURCMDAR at any time to determine the location of the command structure currently being processed.

9.5.2.3 NAND Read Status Polling Example

The following figure shows a more complicated scenario.

This subset of a NAND device workload shows that the first two command structures are used during the data-write phase of an NAND device write operation (CLE and ALE transfers omitted for clarity).

- After writing the data, one must wait until the NAND device status register indicates that the write charge has been transferred. This is built into the workload using a check status command in the NAND in a loop created from the next two DMA command structures.
- The NO_DMA_TRANSFER command is shown here performing the read check, followed by a DMA_SENSE command to branch the DMA command structure list, based on the status of a bit in the external NAND device.

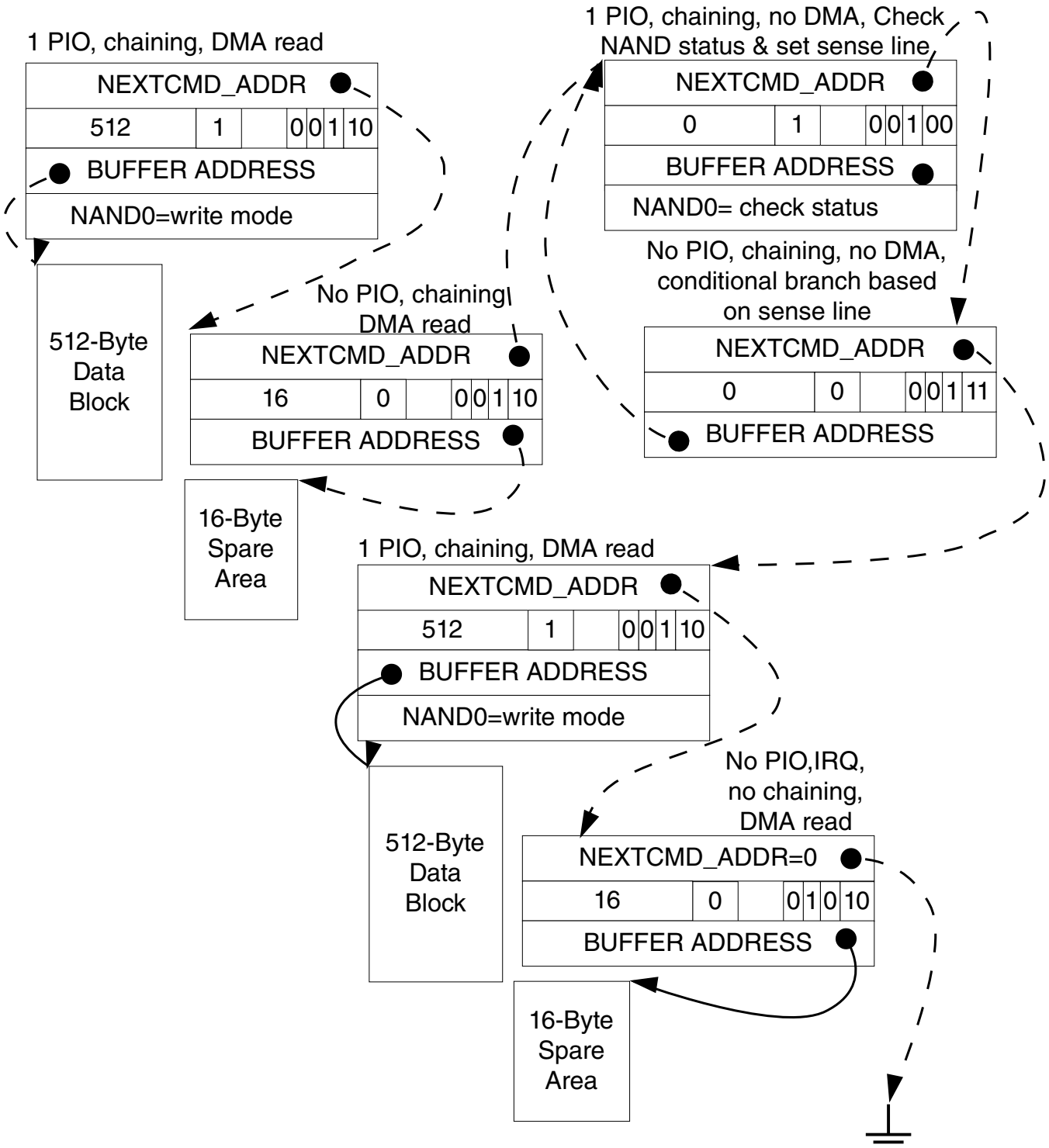


Figure 9-8. AHB-to-APBH Bridge DMA NAND Read Status Polling with DMA Sense Command

The example in the above figure shows the workload continuing immediately to the next NAND page transfer. However, one could perform a second sense operation to see if an error has occurred after the write. One could then point the sense command alternate branch at a NO_DMA_XFER command with the interrupt bit set. If the CHAIN bit is not

set on this failure branch, then the ARM platform is interrupted immediately, and the channel process is also immediately terminated in the presence of a workload-detected NAND error bit.

Note that each word of the three-word DMA command structure corresponds to a PIO register of the DMA that is accessible on the APBH bus. Normally, the DMA copies the next command structure onto these registers for processing at the start of each command by following the value of the pointer previously loaded into the NEXTCMD_ADDR register.

To start DMA processing for the first command, initialize the PIO registers of the desired channel, as follows:

- First, load the next command address register with a pointer to the first command to be loaded.
- Then, write 1 to the counting semaphore register. This causes the DMA to schedule the targeted channel for the DMA command structure load, just as if it had finished its previous command.

9.5.3 APBH register descriptions

APBH Hardware Register Format Summary

9.5.3.1 APBH memory map

APBH base address: 3300_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	AHB to APBH Bridge Control and Status Register 0 (CTRL0)	32	RW	E000_0000h
4h	AHB to APBH Bridge Control and Status Register 0 (CTRL0_SET)	32	RW	E000_0000h
8h	AHB to APBH Bridge Control and Status Register 0 (CTRL0_CLR)	32	RW	E000_0000h
Ch	AHB to APBH Bridge Control and Status Register 0 (CTRL0_TOG)	32	RW	E000_0000h
10h	AHB to APBH Bridge Control and Status Register 1 (CTRL1)	32	RW	0000_0000h
14h	AHB to APBH Bridge Control and Status Register 1 (CTRL1_SET)	32	RW	0000_0000h
18h	AHB to APBH Bridge Control and Status Register 1 (CTRL1_CLR)	32	RW	0000_0000h
1Ch	AHB to APBH Bridge Control and Status Register 1 (CTRL1_TOG)	32	RW	0000_0000h
20h	AHB to APBH Bridge Control and Status Register 2 (CTRL2)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
24h	AHB to APBH Bridge Control and Status Register 2 (CTRL2_SET)	32	RW	0000_0000h
28h	AHB to APBH Bridge Control and Status Register 2 (CTRL2_CLR)	32	RW	0000_0000h
2Ch	AHB to APBH Bridge Control and Status Register 2 (CTRL2_TOG)	32	RW	0000_0000h
30h	AHB to APBH Bridge Channel Register (CHANNEL_CTRL)	32	RW	0000_0000h
34h	AHB to APBH Bridge Channel Register (CHANNEL_CTRL_SET)	32	RW	0000_0000h
38h	AHB to APBH Bridge Channel Register (CHANNEL_CTRL_CLR)	32	RW	0000_0000h
3Ch	AHB to APBH Bridge Channel Register (CHANNEL_CTRL_TOG)	32	RW	0000_0000h
50h	AHB to APBH DMA burst size (DMA_BURST_SIZE)	32	RW	0055_5555h
60h	AHB to APBH DMA Debug Register (DEBUG)	32	RW	0000_0000h
100h	APBH DMA Channel n Current Command Address Register (CH0_CURCMDAR)	32	RO	0000_0000h
110h	APBH DMA Channel n Next Command Address Register (CH0_NXTCMDAR)	32	RW	0000_0000h
120h	APBH DMA Channel n Command Register (CH0_CMD)	32	RO	0000_0000h
130h	APBH DMA Channel n Buffer Address Register (CH0_BAR)	32	RO	0000_0000h
140h	APBH DMA Channel n Semaphore Register (CH0_SEMA)	32	RW	0000_0000h
150h	AHB to APBH DMA Channel n Debug Information (CH0_DEBUG1)	32	RO	00A0_0000h
160h	AHB to APBH DMA Channel n Debug Information (CH0_DEBUG2)	32	RO	0000_0000h
170h	APBH DMA Channel n Current Command Address Register (CH1_CURCMDAR)	32	RO	0000_0000h
180h	APBH DMA Channel n Next Command Address Register (CH1_NXTCMDAR)	32	RW	0000_0000h
190h	APBH DMA Channel n Command Register (CH1_CMD)	32	RO	0000_0000h
1A0h	APBH DMA Channel n Buffer Address Register (CH1_BAR)	32	RO	0000_0000h
1B0h	APBH DMA Channel n Semaphore Register (CH1_SEMA)	32	RW	0000_0000h
1C0h	AHB to APBH DMA Channel n Debug Information (CH1_DEBUG1)	32	RO	00A0_0000h
1D0h	AHB to APBH DMA Channel n Debug Information (CH1_DEBUG2)	32	RO	0000_0000h
1E0h	APBH DMA Channel n Current Command Address Register (CH2_CURCMDAR)	32	RO	0000_0000h
1F0h	APBH DMA Channel n Next Command Address Register (CH2_NXTCMDAR)	32	RW	0000_0000h
200h	APBH DMA Channel n Command Register (CH2_CMD)	32	RO	0000_0000h
210h	APBH DMA Channel n Buffer Address Register (CH2_BAR)	32	RO	0000_0000h
220h	APBH DMA Channel n Semaphore Register (CH2_SEMA)	32	RW	0000_0000h
230h	AHB to APBH DMA Channel n Debug Information (CH2_DEBUG1)	32	RO	00A0_0000h
240h	AHB to APBH DMA Channel n Debug Information (CH2_DEBUG2)	32	RO	0000_0000h
250h	APBH DMA Channel n Current Command Address Register (CH3_CURCMDAR)	32	RO	0000_0000h
260h	APBH DMA Channel n Next Command Address Register (CH3_NXTCMDAR)	32	RW	0000_0000h
270h	APBH DMA Channel n Command Register (CH3_CMD)	32	RO	0000_0000h
280h	APBH DMA Channel n Buffer Address Register (CH3_BAR)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
290h	APBH DMA Channel n Semaphore Register (CH3_SEMA)	32	RW	0000_0000h
2A0h	AHB to APBH DMA Channel n Debug Information (CH3_DEBUG1)	32	RO	00A0_0000h
2B0h	AHB to APBH DMA Channel n Debug Information (CH3_DEBUG2)	32	RO	0000_0000h
2C0h	APBH DMA Channel n Current Command Address Register (CH4_CURCMDAR)	32	RO	0000_0000h
2D0h	APBH DMA Channel n Next Command Address Register (CH4_NXTCMDAR)	32	RW	0000_0000h
2E0h	APBH DMA Channel n Command Register (CH4_CMD)	32	RO	0000_0000h
2F0h	APBH DMA Channel n Buffer Address Register (CH4_BAR)	32	RO	0000_0000h
300h	APBH DMA Channel n Semaphore Register (CH4_SEMA)	32	RW	0000_0000h
310h	AHB to APBH DMA Channel n Debug Information (CH4_DEBUG1)	32	RO	00A0_0000h
320h	AHB to APBH DMA Channel n Debug Information (CH4_DEBUG2)	32	RO	0000_0000h
330h	APBH DMA Channel n Current Command Address Register (CH5_CURCMDAR)	32	RO	0000_0000h
340h	APBH DMA Channel n Next Command Address Register (CH5_NXTCMDAR)	32	RW	0000_0000h
350h	APBH DMA Channel n Command Register (CH5_CMD)	32	RO	0000_0000h
360h	APBH DMA Channel n Buffer Address Register (CH5_BAR)	32	RO	0000_0000h
370h	APBH DMA Channel n Semaphore Register (CH5_SEMA)	32	RW	0000_0000h
380h	AHB to APBH DMA Channel n Debug Information (CH5_DEBUG1)	32	RO	00A0_0000h
390h	AHB to APBH DMA Channel n Debug Information (CH5_DEBUG2)	32	RO	0000_0000h
3A0h	APBH DMA Channel n Current Command Address Register (CH6_CURCMDAR)	32	RO	0000_0000h
3B0h	APBH DMA Channel n Next Command Address Register (CH6_NXTCMDAR)	32	RW	0000_0000h
3C0h	APBH DMA Channel n Command Register (CH6_CMD)	32	RO	0000_0000h
3D0h	APBH DMA Channel n Buffer Address Register (CH6_BAR)	32	RO	0000_0000h
3E0h	APBH DMA Channel n Semaphore Register (CH6_SEMA)	32	RW	0000_0000h
3F0h	AHB to APBH DMA Channel n Debug Information (CH6_DEBUG1)	32	RO	00A0_0000h
400h	AHB to APBH DMA Channel n Debug Information (CH6_DEBUG2)	32	RO	0000_0000h
410h	APBH DMA Channel n Current Command Address Register (CH7_CURCMDAR)	32	RO	0000_0000h
420h	APBH DMA Channel n Next Command Address Register (CH7_NXTCMDAR)	32	RW	0000_0000h
430h	APBH DMA Channel n Command Register (CH7_CMD)	32	RO	0000_0000h
440h	APBH DMA Channel n Buffer Address Register (CH7_BAR)	32	RO	0000_0000h
450h	APBH DMA Channel n Semaphore Register (CH7_SEMA)	32	RW	0000_0000h
460h	AHB to APBH DMA Channel n Debug Information (CH7_DEBUG1)	32	RO	00A0_0000h
470h	AHB to APBH DMA Channel n Debug Information (CH7_DEBUG2)	32	RO	0000_0000h
480h	APBH DMA Channel n Current Command Address Register (CH8_CURCMDAR)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
490h	APBH DMA Channel n Next Command Address Register (CH8_NXTCMDAR)	32	RW	0000_0000h
4A0h	APBH DMA Channel n Command Register (CH8_CMD)	32	RO	0000_0000h
4B0h	APBH DMA Channel n Buffer Address Register (CH8_BAR)	32	RO	0000_0000h
4C0h	APBH DMA Channel n Semaphore Register (CH8_SEMA)	32	RW	0000_0000h
4D0h	AHB to APBH DMA Channel n Debug Information (CH8_DEBUG1)	32	RO	00A0_0000h
4E0h	AHB to APBH DMA Channel n Debug Information (CH8_DEBUG2)	32	RO	0000_0000h
4F0h	APBH DMA Channel n Current Command Address Register (CH9_CURCMDAR)	32	RO	0000_0000h
500h	APBH DMA Channel n Next Command Address Register (CH9_NXTCMDAR)	32	RW	0000_0000h
510h	APBH DMA Channel n Command Register (CH9_CMD)	32	RO	0000_0000h
520h	APBH DMA Channel n Buffer Address Register (CH9_BAR)	32	RO	0000_0000h
530h	APBH DMA Channel n Semaphore Register (CH9_SEMA)	32	RW	0000_0000h
540h	AHB to APBH DMA Channel n Debug Information (CH9_DEBUG1)	32	RO	00A0_0000h
550h	AHB to APBH DMA Channel n Debug Information (CH9_DEBUG2)	32	RO	0000_0000h
560h	APBH DMA Channel n Current Command Address Register (CH10_CURCMDAR)	32	RO	0000_0000h
570h	APBH DMA Channel n Next Command Address Register (CH10_NXTCMDAR)	32	RW	0000_0000h
580h	APBH DMA Channel n Command Register (CH10_CMD)	32	RO	0000_0000h
590h	APBH DMA Channel n Buffer Address Register (CH10_BAR)	32	RO	0000_0000h
5A0h	APBH DMA Channel n Semaphore Register (CH10_SEMA)	32	RW	0000_0000h
5B0h	AHB to APBH DMA Channel n Debug Information (CH10_DEBUG1)	32	RO	00A0_0000h
5C0h	AHB to APBH DMA Channel n Debug Information (CH10_DEBUG2)	32	RO	0000_0000h
5D0h	APBH DMA Channel n Current Command Address Register (CH11_CURCMDAR)	32	RO	0000_0000h
5E0h	APBH DMA Channel n Next Command Address Register (CH11_NXTCMDAR)	32	RW	0000_0000h
5F0h	APBH DMA Channel n Command Register (CH11_CMD)	32	RO	0000_0000h
600h	APBH DMA Channel n Buffer Address Register (CH11_BAR)	32	RO	0000_0000h
610h	APBH DMA Channel n Semaphore Register (CH11_SEMA)	32	RW	0000_0000h
620h	AHB to APBH DMA Channel n Debug Information (CH11_DEBUG1)	32	RO	00A0_0000h
630h	AHB to APBH DMA Channel n Debug Information (CH11_DEBUG2)	32	RO	0000_0000h
640h	APBH DMA Channel n Current Command Address Register (CH12_CURCMDAR)	32	RO	0000_0000h
650h	APBH DMA Channel n Next Command Address Register (CH12_NXTCMDAR)	32	RW	0000_0000h
660h	APBH DMA Channel n Command Register (CH12_CMD)	32	RO	0000_0000h
670h	APBH DMA Channel n Buffer Address Register (CH12_BAR)	32	RO	0000_0000h
680h	APBH DMA Channel n Semaphore Register (CH12_SEMA)	32	RW	0000_0000h
690h	AHB to APBH DMA Channel n Debug Information (CH12_DEBUG1)	32	RO	00A0_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
6A0h	AHB to APBH DMA Channel n Debug Information (CH12_DEBUG2)	32	RO	0000_0000h
6B0h	APBH DMA Channel n Current Command Address Register (CH13_CURCMDAR)	32	RO	0000_0000h
6C0h	APBH DMA Channel n Next Command Address Register (CH13_NXTCMDAR)	32	RW	0000_0000h
6D0h	APBH DMA Channel n Command Register (CH13_CMD)	32	RO	0000_0000h
6E0h	APBH DMA Channel n Buffer Address Register (CH13_BAR)	32	RO	0000_0000h
6F0h	APBH DMA Channel n Semaphore Register (CH13_SEMA)	32	RW	0000_0000h
700h	AHB to APBH DMA Channel n Debug Information (CH13_DEBUG1)	32	RO	00A0_0000h
710h	AHB to APBH DMA Channel n Debug Information (CH13_DEBUG2)	32	RO	0000_0000h
720h	APBH DMA Channel n Current Command Address Register (CH14_CURCMDAR)	32	RO	0000_0000h
730h	APBH DMA Channel n Next Command Address Register (CH14_NXTCMDAR)	32	RW	0000_0000h
740h	APBH DMA Channel n Command Register (CH14_CMD)	32	RO	0000_0000h
750h	APBH DMA Channel n Buffer Address Register (CH14_BAR)	32	RO	0000_0000h
760h	APBH DMA Channel n Semaphore Register (CH14_SEMA)	32	RW	0000_0000h
770h	AHB to APBH DMA Channel n Debug Information (CH14_DEBUG1)	32	RO	00A0_0000h
780h	AHB to APBH DMA Channel n Debug Information (CH14_DEBUG2)	32	RO	0000_0000h
790h	APBH DMA Channel n Current Command Address Register (CH15_CURCMDAR)	32	RO	0000_0000h
7A0h	APBH DMA Channel n Next Command Address Register (CH15_NXTCMDAR)	32	RW	0000_0000h
7B0h	APBH DMA Channel n Command Register (CH15_CMD)	32	RO	0000_0000h
7C0h	APBH DMA Channel n Buffer Address Register (CH15_BAR)	32	RO	0000_0000h
7D0h	APBH DMA Channel n Semaphore Register (CH15_SEMA)	32	RW	0000_0000h
7E0h	AHB to APBH DMA Channel n Debug Information (CH15_DEBUG1)	32	RO	00A0_0000h
7F0h	AHB to APBH DMA Channel n Debug Information (CH15_DEBUG2)	32	RO	0000_0000h
800h	APBH Bridge Version Register (VERSION)	32	RO	0301_0000h

9.5.3.2 AHB to APBH Bridge Control and Status Register 0 (CTRL0)

9.5.3.2.1 Offset

Register	Offset	Description
CTRL0	0h	AHB to APBH Bridge Control and Status Register 0
CTRL0_SET	4h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL0

Table continues on the next page...

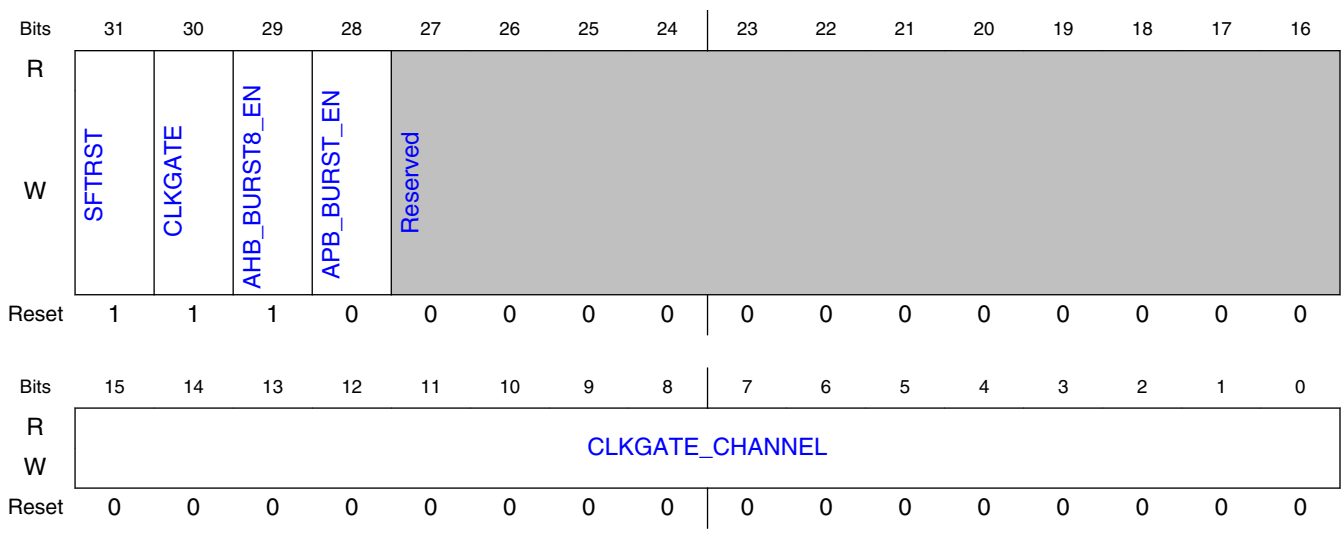
Register	Offset	Description
CTRL0_CLR	8h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL0
CTRL0_TOG	Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL0

9.5.3.2.2 Function

The APBH CTRL 0 provides overall control of the AHB to APBH bridge and DMA.

This register contains module softreset, clock gating, channel clock gating/freeze bits.

9.5.3.2.3 Diagram



9.5.3.2.4 Fields

Field	Function
31 SFTRST	SFTRST Set this bit to zero to enable normal APBH DMA operation. Set this bit to one (default) to disable clocking with the APBH DMA and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset the APBH DMA block to its default state.
30 CLKGATE	CLKGATE This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block.
29 AHB_BURST8_EN	AHB_BURST8_EN Set this bit to one (default) to enable AHB 8-beat burst. Set to zero to disable 8-beat burst on AHB interface.
28	APB_BURST_EN

Table continues on the next page...

Field	Function
APB_BURST_EN	Set this bit to one to enable apb master do a continous transfers when a device request a burst dma. Set to zero will treat a burst dma request as 4/8 individual requests.
27-16 —	RSVD0 Reserved, always set to zero.
15-0 CLKGATE_CHANNEL	CLKGATE_CHANNEL These bits must be set to zero for normal operation of each channel. When set to one they gate off the individual clocks to the channels. 0000000000000001b - 0000000000000010b - 0000000000000100b - 0000000000001000b - 0000000000010000b - 0000000000100000b - 0000000001000000b - 0000000010000000b - 0000000100000000b -

9.5.3.3 AHB to APBH Bridge Control and Status Register 1 (CTRL1)

9.5.3.3.1 Offset

Register	Offset	Description
CTRL1	10h	AHB to APBH Bridge Control and Status Register 1
CTRL1_SET	14h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL1
CTRL1_CLR	18h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL1
CTRL1_TOG	1Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL1

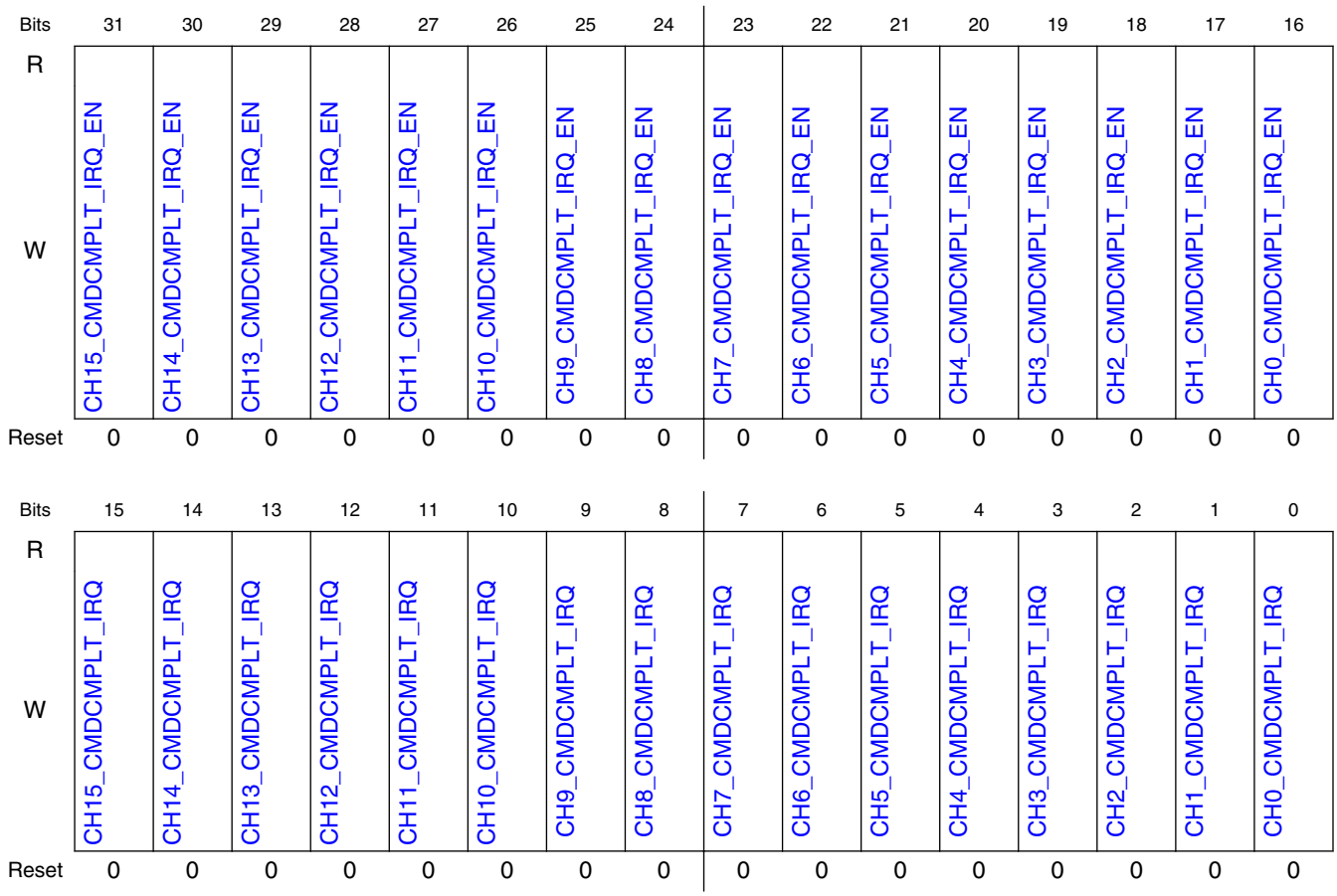
9.5.3.3.2 Function

The APBH CTRL one provides overall control of the interrupts generated by the AHB to APBH DMA. This register contains the per channel interrupt status bits and the per channel interrupt enable bits. Each channel has a dedicated interrupt vector in the vectored interrupt controller.

EXAMPLE

```
// use bitfield write macro
BF_WR(APBH_CTRL1, CH5_CMDCMPLT_IRQ, 0);
// or, assign to register struct's bitfield
BF_APBH_CTRL1.CH5_CMDCMPLT_IRQ = 0;
```

9.5.3.3.3 Diagram



9.5.3.3.4 Fields

Field	Function
31 CH15_CMDCMPLT_IRQ_EN	CH15_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 15.
30 CH14_CMDCMPLT_IRQ_EN	CH14_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 14.
29 CH13_CMDCMPLT_IRQ_EN	CH13_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 13.
28 CH12_CMDCMPLT_IRQ_EN	CH12_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 12.
27	CH11_CMDCMPLT_IRQ_EN

Table continues on the next page...

Field	Function
CH11_CMDCMPLT_IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 11.
26 CH10_CMDCMPLT_IRQ_EN	CH10_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 10.
25 CH9_CMDCMPLT_IRQ_EN	CH9_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 9.
24 CH8_CMDCMPLT_IRQ_EN	CH8_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 8.
23 CH7_CMDCMPLT_IRQ_EN	CH7_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 7.
22 CH6_CMDCMPLT_IRQ_EN	CH6_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 6.
21 CH5_CMDCMPLT_IRQ_EN	CH5_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 5.
20 CH4_CMDCMPLT_IRQ_EN	CH4_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 4.
19 CH3_CMDCMPLT_IRQ_EN	CH3_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 3.
18 CH2_CMDCMPLT_IRQ_EN	CH2_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 2.
17 CH1_CMDCMPLT_IRQ_EN	CH1_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 1.
16 CH0_CMDCMPLT_IRQ_EN	CH0_CMDCMPLT_IRQ_EN Setting this bit enables the generation of an interrupt request for APBH DMA channel 0.
15 CH15_CMDCMPLT_IRQ	CH15_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA Channel 15. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
14 CH14_CMDCMPLT_IRQ	CH14_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA Channel 14. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
13 CH13_CMDCMPLT_IRQ	CH13_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA Channel 13. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.

Table continues on the next page...

Field	Function
12 CH12_CMDCMPLT_IRQ	CH12_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA Channel 12. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
11 CH11_CMDCMPLT_IRQ	CH11_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA Channel 11. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
10 CH10_CMDCMPLT_IRQ	CH10_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA Channel 10. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
9 CH9_CMDCMPLT_IRQ	CH9_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA Channel 9. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
8 CH8_CMDCMPLT_IRQ	CH8_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA Channel 8. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
7 CH7_CMDCMPLT_IRQ	CH7_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA channel 7. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
6 CH6_CMDCMPLT_IRQ	CH6_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA channel 6. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
5 CH5_CMDCMPLT_IRQ	CH5_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA channel 5. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
4 CH4_CMDCMPLT_IRQ	CH4_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA channel 4. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
3 CH3_CMDCMPLT_IRQ	CH3_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA channel 3. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
2 CH2_CMDCMPLT_IRQ	CH2_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA channel 2. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
1 CH1_CMDCMPLT_IRQ	CH1_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA channel 1. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
0 CH0_CMDCMPLT_IRQ	CH0_CMDCMPLT_IRQ Interrupt request status bit for APBH DMA channel 0. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.

9.5.3.4 AHB to APBH Bridge Control and Status Register 2 (CTRL2)

9.5.3.4.1 Offset

Register	Offset	Description
CTRL2	20h	AHB to APBH Bridge Control and Status Register 2
CTRL2_SET	24h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL2
CTRL2_CLR	28h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL2
CTRL2_TOG	2Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL2

9.5.3.4.2 Function

The APBH CTRL 2 provides channel error interrupts generated by the AHB to APBH DMA. This register contains the per channel interrupt status bits and the per channel interrupt enable bits. Each channel has a dedicated interrupt vector in the vectored interrupt controller.

EXAMPLE

```
// use bitfield write macro
BF_WR(APBH_CTRL1, CH5_CMDCMPLT_IRQ, 0);
// or, assign to register struct's bitfield
BF_APBH_CTRL1.CH5_CMDCMPLT_IRQ = 0;
```

9.5.3.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CH15_ERROR_STATUS	CH14_ERROR_STATUS	CH13_ERROR_STATUS	CH12_ERROR_STATUS	CH11_ERROR_STATUS	CH10_ERROR_STATUS	CH9_ERROR_STATUS	CH8_ERROR_STATUS	CH7_ERROR_STATUS	CH6_ERROR_STATUS	CH5_ERROR_STATUS	CH4_ERROR_STATUS	CH3_ERROR_STATUS	CH2_ERROR_STATUS	CH1_ERROR_STATUS	CH0_ERROR_STATUS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH15_ERROR_IRQ	CH14_ERROR_IRQ	CH13_ERROR_IRQ	CH12_ERROR_IRQ	CH11_ERROR_IRQ	CH10_ERROR_IRQ	CH9_ERROR_IRQ	CH8_ERROR_IRQ	CH7_ERROR_IRQ	CH6_ERROR_IRQ	CH5_ERROR_IRQ	CH4_ERROR_IRQ	CH3_ERROR_IRQ	CH2_ERROR_IRQ	CH1_ERROR_IRQ	CH0_ERROR_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.5.3.4.4 Fields

Field	Function
31 CH15_ERROR_STATUS	CH15_ERROR_STATUS Error status bit for APBH DMA Channel 15. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
30 CH14_ERROR_STATUS	CH14_ERROR_STATUS Error status bit for APBH DMA Channel 14. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
29 CH13_ERROR_STATUS	CH13_ERROR_STATUS Error status bit for APBH DMA Channel 13. Valid when corresponding Error IRQ is set. 1 - AHB bus error

Table continues on the next page...

Field	Function
	0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
28 CH12_ERROR_STATUS	CH12_ERROR_STATUS Error status bit for APBH DMA Channel 12. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
27 CH11_ERROR_STATUS	CH11_ERROR_STATUS Error status bit for APBH DMA Channel 11. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
26 CH10_ERROR_STATUS	CH10_ERROR_STATUS Error status bit for APBH DMA Channel 10. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
25 CH9_ERROR_STATUS	CH9_ERROR_STATUS Error status bit for APBH DMA Channel 9. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
24 CH8_ERROR_STATUS	CH8_ERROR_STATUS Error status bit for APBH DMA Channel 8. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
23 CH7_ERROR_STATUS	CH7_ERROR_STATUS Error status bit for APBX DMA Channel 7. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
22 CH6_ERROR_STATUS	CH6_ERROR_STATUS Error status bit for APBX DMA Channel 6. Valid when corresponding Error IRQ is set.

Table continues on the next page...

Field	Function
	1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
21 CH5_ERROR_STATUS	CH5_ERROR_STATUS Error status bit for APBX DMA Channel 5. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
20 CH4_ERROR_STATUS	CH4_ERROR_STATUS Error status bit for APBX DMA Channel 4. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
19 CH3_ERROR_STATUS	CH3_ERROR_STATUS Error status bit for APBX DMA Channel 3. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
18 CH2_ERROR_STATUS	CH2_ERROR_STATUS Error status bit for APBX DMA Channel 2. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
17 CH1_ERROR_STATUS	CH1_ERROR_STATUS Error status bit for APBX DMA Channel 1. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
16 CH0_ERROR_STATUS	CH0_ERROR_STATUS Error status bit for APBX DMA Channel 0. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0b - An early termination from the device causes error IRQ. 1b - An AHB bus error causes error IRQ.
15	CH15_ERROR_IRQ

Table continues on the next page...

Field	Function
CH15_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 15. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
14 CH14_ERROR_IRQ	CH14_ERROR_IRQ Error interrupt status bit for APBH DMA Channel 14. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
13 CH13_ERROR_IRQ	CH13_ERROR_IRQ Error interrupt status bit for APBH DMA Channel 13. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
12 CH12_ERROR_IRQ	CH12_ERROR_IRQ Error interrupt status bit for APBH DMA Channel 12. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
11 CH11_ERROR_IRQ	CH11_ERROR_IRQ Error interrupt status bit for APBH DMA Channel 11. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
10 CH10_ERROR_IRQ	CH10_ERROR_IRQ Error interrupt status bit for APBH DMA Channel 10. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
9 CH9_ERROR_IRQ	CH9_ERROR_IRQ Error interrupt status bit for APBH DMA Channel 9. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
8 CH8_ERROR_IRQ	CH8_ERROR_IRQ Error interrupt status bit for APBH DMA Channel 8. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
7 CH7_ERROR_IRQ	CH7_ERROR_IRQ Error interrupt status bit for APBX DMA Channel 7. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
6 CH6_ERROR_IRQ	CH6_ERROR_IRQ Error interrupt status bit for APBX DMA Channel 6. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
5 CH5_ERROR_IRQ	CH5_ERROR_IRQ Error interrupt status bit for APBX DMA Channel 5. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
4 CH4_ERROR_IRQ	CH4_ERROR_IRQ Error interrupt status bit for APBX DMA Channel 4. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
3 CH3_ERROR_IRQ	CH3_ERROR_IRQ Error interrupt status bit for APBX DMA Channel 3. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
2 CH2_ERROR_IRQ	CH2_ERROR_IRQ Error interrupt status bit for APBX DMA Channel 2. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.
1 CH1_ERROR_IRQ	CH1_ERROR_IRQ Error interrupt status bit for APBX DMA Channel 1. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.

Table continues on the next page...

Field	Function
0 CH0_ERROR_IRQ	CH0_ERROR_IRQ Error interrupt status bit for APBX DMA Channel 0. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to ARM.

9.5.3.5 AHB to APBH Bridge Channel Register (CHANNEL_CTRL)

9.5.3.5.1 Offset

Register	Offset	Description
CHANNEL_CTRL	30h	AHB to APBH Bridge Channel Register
CHANNEL_CTRL_SET	34h	Writing a 1 to a bit in this register sets the corresponding bit in CHANNEL_CTRL
CHANNEL_CTRL_CLR	38h	Writing a 1 to a bit in this register clears the corresponding bit in CHANNEL_CTRL
CHANNEL_CTRL_TOG	3Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CHANNEL_CTRL

9.5.3.5.2 Function

The APBH CHANNEL CTRL provides reset/freeze control of each DMA channel. This register contains individual channel reset/freeze bits.

9.5.3.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RESET_CHANNEL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FREEZE_CHANNEL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.5.3.5.4 Fields

Field	Function
31-16 RESET_CHANNEL	RESET_CHANNEL Setting a bit in this field causes the DMA controller to take the corresponding channel through its reset state. The bit is reset after the channel resources are cleared. 0000000000000001b - 0000000000000010b - 0000000000000100b - 0000000000001000b - 0000000000010000b - 0000000000100000b - 0000000001000000b - 0000000010000000b - 0000000100000000b -
15-0 FREEZE_CHANNEL	FREEZE_CHANNEL Setting a bit in this field will freeze the DMA channel associated with it. This field is a direct input to the DMA channel arbiter. When frozen, the channel is deined access to the central DMA resources. 0000000000000001b - 0000000000000010b - 0000000000000100b - 0000000000001000b - 0000000000010000b - 0000000000100000b - 0000000001000000b - 0000000010000000b - 0000000100000000b -

9.5.3.6 AHB to APBH DMA burst size (DMA_BURST_SIZE)

9.5.3.6.1 Offset

Register	Offset
DMA_BURST_SIZE	50h

9.5.3.6.2 Function

This register programs the apbh burst size of the APBH DMA devices when a DMA burst request is issued.

This register provides a mechanism for assigning the device.

9.5.3.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		CH8	
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH7		CH6		CH5		CH4		CH3		CH2		CH1		CH0	
W																
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

9.5.3.6.4 Fields

Field	Function
31-30	CH15
—	Reserved.
29-28	CH14
—	Reserved.
27-26	CH13
—	Reserved.
25-24	CH12
—	Reserved.
23-22	CH11
—	Reserved.
21-20	CH10
—	Reserved.
19-18	CH9
—	Reserved.
17-16	CH8
CH8	DMA burst size for SSP. 00b - 01b - 10b -
15-14	CH7
CH7	DMA burst size for GPMI channel 7. Do not change. GPMI only support burst size 4.
13-12	CH6
CH6	DMA burst size for GPMI channel 6. Do not change. GPMI only support burst size 4.
11-10	CH5
CH5	DMA burst size for GPMI channel 5. Do not change. GPMI only support burst size 4.

Table continues on the next page...

Field	Function
9-8 CH4	CH4 DMA burst size for GPMI channel 4. Do not change. GPMI only support burst size 4.
7-6 CH3	CH3 DMA burst size for GPMI channel 3. Do not change. GPMI only support burst size 4.
5-4 CH2	CH2 DMA burst size for GPMI channel 2. Do not change. GPMI only support burst size 4.
3-2 CH1	CH1 DMA burst size for GPMI channel 1. Do not change. GPMI only support burst size 4.
1-0 CH0	CH0 DMA burst size for GPMI channel 0. Do not change. GPMI only support burst size 4.

9.5.3.7 AHB to APBH DMA Debug Register (DEBUG)

9.5.3.7.1 Offset

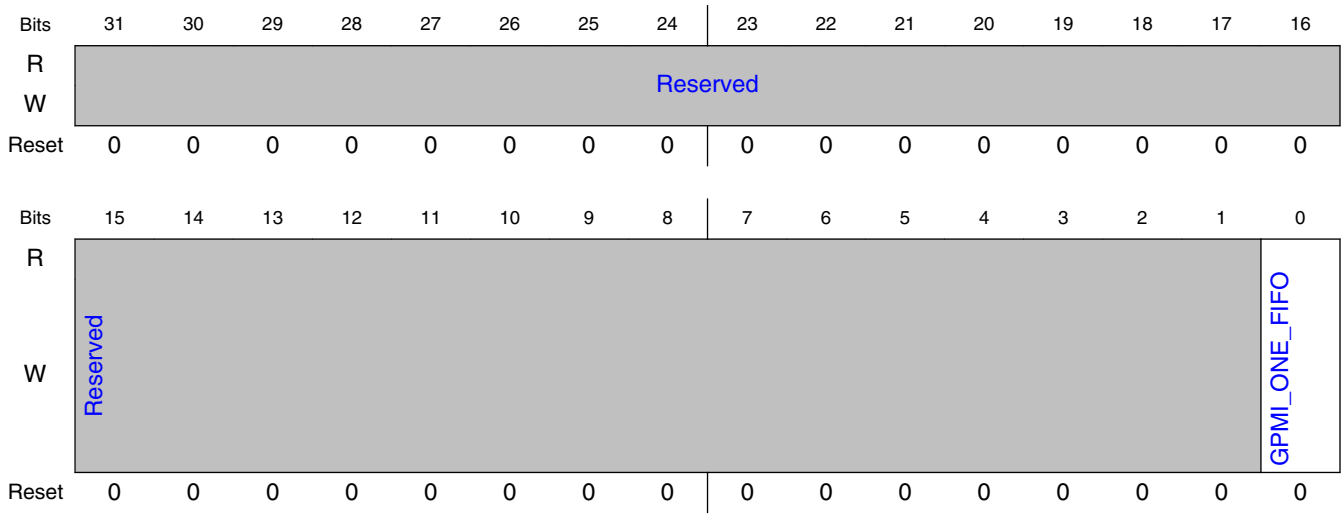
Register	Offset
DEBUG	60h

9.5.3.7.2 Function

This register is for debug purpose.

The debug register is for internal use only. Not recommend for customer useage.

9.5.3.7.3 Diagram



9.5.3.7.4 Fields

Field	Function
31-1 —	- Reserved, always set to zero.
0 GPMI_ONE_FIFO	GPMI_ONE_FIFO Set to 0ne and the 8 GPMI channels will share the DMA FIFO, and when set to zero, the 8 GPMI channels will use its own DMA FIFO.

9.5.3.8 APBH DMA Channel n Current Command Address Register (CH0_CURCMDAR - CH15_CURCMDAR)

9.5.3.8.1 Offset

For a = 0 to 15:

Register	Offset
CHa_CURCMDAR	100h + (a × 70h)

9.5.3.8.2 Function

The APBH DMA channel n current command address register points to the multiword command that is currently being executed. Commands are threaded on the command address.

APBH DMA Channel n is controlled by a variable sized command structure. This register points to the command structure currently being executed.

EXAMPLE

```
// read the whole register, since there is only one field
pCurCmd = (apbh_chn_cmd_t *) APBH_CHn_CURCMDAR_RD(0);
// or, use multi-register bitfield read macro
pCurCmd = (apbh_chn_cmd_t *) BF_RDn(APBH_CHn_CURCMDAR, 0, CMD_ADDR);
// or, assign from bitfield of indexed register's struct
pCurCmd = (apbh_chn_cmd_t *) APBH_CHn_CURCMDAR(0).CMD_ADDR;
```

9.5.3.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMD_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMD_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.5.3.8.4 Fields

Field	Function
31-0 CMD_ADDR	CMD_ADDR Pointer to command structure currently being processed for channel n.

9.5.3.9 APBH DMA Channel n Next Command Address Register (CH0_NXTCMDAR - CH15_NXTCMDAR)

9.5.3.9.1 Offset

For a = 0 to 15:

Register	Offset
CHa_NXTCMDAR	110h + (a × 70h)

9.5.3.9.2 Function

The APBH DMA Channel n Next Command Address register contains the address of the next multiword command to be executed. Commands are threaded on the command address. Set CHAIN to 1 in the DMA command word to process command lists.

APBH DMA Channel n is controlled by a variable sized command structure. Software loads this register with the address of the first command structure to process and increments the Channel n semaphore to start processing. This register points to the next command structure to be executed when the current command is completed.

EXAMPLE

```
// write the entire register, since there is only one field
APBH_CHn_NXTCMDAR_WR(0, (reg32_t) pCommandTwoStructure);
// or, use multi-registerbitfield write macro
BF_WRn(APBH_CHn_NXTCMDAR, 0, (reg32_t) pCommandTwoStructure);
// or, assign to bitfield of indexed register's struct
APBH_CHn_NXTCMDAR(0).CMD_ADDR = (reg32_t) pCommandTwoStructure;
```

9.5.3.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CMD_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMD_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.5.3.9.4 Fields

Field	Function
31-0 CMD_ADDR	CMD_ADDR Pointer to next command structure for channel n.

9.5.3.10 APBH DMA Channel n Command Register (CH0_CMD - CH15_CMD)

9.5.3.10.1 Offset

For a = 0 to 15:

Register	Offset
CHa_CMD	120h + (a × 70h)

9.5.3.10.2 Function

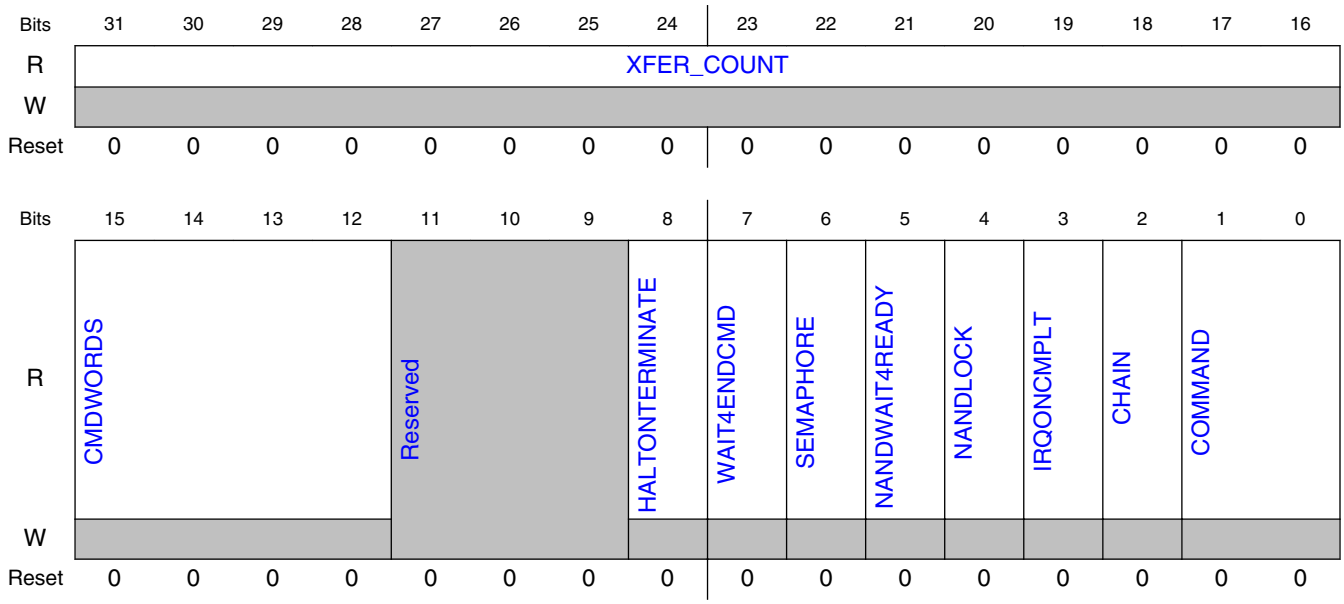
The APBH DMA Channel n command register specifies the DMA transaction to perform for the current command chain item.

The command register controls the overall operation of each DMA command for this channel. It includes the number of bytes to transfer to or from the device, the number of APB PIO command words included with this command structure, whether to interrupt at command completion, whether to chain an additional command to the end of this one and whether this transfer is a read or write DMA transfer.

EXAMPLE

```
apbh_chn_cmd_t dma_cmd;
// transfer 512 bytes
dma_cmd.XFER_COUNT = 512;
// transfer to system memory from peripheral device
dma_cmd.COMMAND = BV_APBH_CHn_CMD_COMMAND__DMA_WRITE;
// chain an additional command structure on to the list
dma_cmd.CHAIN = 1;
// generate an interrupt on completion of this command structure
dma_cmd.IRQONCMPLT = 1;
```

9.5.3.10.3 Diagram



9.5.3.10.4 Fields

Field	Function
31-16 XFER_COUNT	XFER_COUNT This field indicates the number of bytes to transfer to or from the appropriate PIO register in the GPMI0 device. A value of 0 indicates a 64 KBytes transfer.
15-12 CMDWORDS	CMDWORDS This field indicates the number of command words to send to the GPMI0, starting with the base PIO address of the GPMI0 control register and incrementing from there. Zero means transfer NO command words
11-9 —	- Reserved, always set to zero.
8 HALTONTERMINATE	HALTONTERMINATE A value of one indicates that the channel will immeditately terminate the current descriptor and halt the DMA channel if a terminate signal is set. A value of 0 will still cause an immediate terminate of the channel if the terminate signal is set, but the channel will continue as if the count had been exhausted, meaning it will honor IRQONCMPLT, CHAIN, SEMAPHORE, and WAIT4ENDCMD.
7 WAIT4ENDCMD	WAIT4ENDCMD A value of one indicates that the channel will wait for the end of command signal to be sent from the APBH device to the DMA before starting the next DMA command.
6 SEMAPHORE	SEMAPHORE A value of one indicates that the channel will decrement its semaphore at the completion of the current command structure. If the semaphore decrements to zero, then this channel stalls until software increments it again.
5	NANDWAIT4READY

Table continues on the next page...

Field	Function
NANDWAIT4READY	A value of one indicates that the NAND DMA channel will will wait until the NAND device reports "ready" before executing the command. It is ignored for non-NAND DMA channels.
4 NANDLOCK	NANDLOCK A value of one indicates that the NAND DMA channel will remain "locked" in the arbiter at the expense of other NAND DMA channels. It is ignored for non-NAND DMA channels.
3 IRQONCMPLT	IRQONCMPLT A value of one indicates that the channel will cause the interrupt status bit to be set upon completion of the current command, i.e. after the DMA transfer is complete.
2 CHAIN	CHAIN A value of one indicates that another command is chained onto the end of the current command structure. At the completion of the current command, this channel will follow the pointer in APBH_CHn_CMDAR to find the next command.
1-0 COMMAND	COMMAND This bitfield indicates the type of current command: 00b - Perform any requested PIO word transfers but terminate command before any DMA transfer. 01b - Perform any requested PIO word transfers and then perform a DMA transfer from the peripheral for the specified number of bytes. 10b - Perform any requested PIO word transfers and then perform a DMA transfer to the peripheral for the specified number of bytes. 11b - Perform any requested PIO word transfers and then perform a conditional branch to the next chained device. Follow the NEXCMD_ADDR pointer if the perpheral sense is true. Follow the BUFFER_ADDRESS as a chain pointer if the peripheral sense line is false.

9.5.3.11 APBH DMA Channel n Buffer Address Register (CH0_BAR - CH15_BAR)

9.5.3.11.1 Offset

For a = 0 to 15:

Register	Offset
CHa_BAR	130h + (a × 70h)

9.5.3.11.2 Function

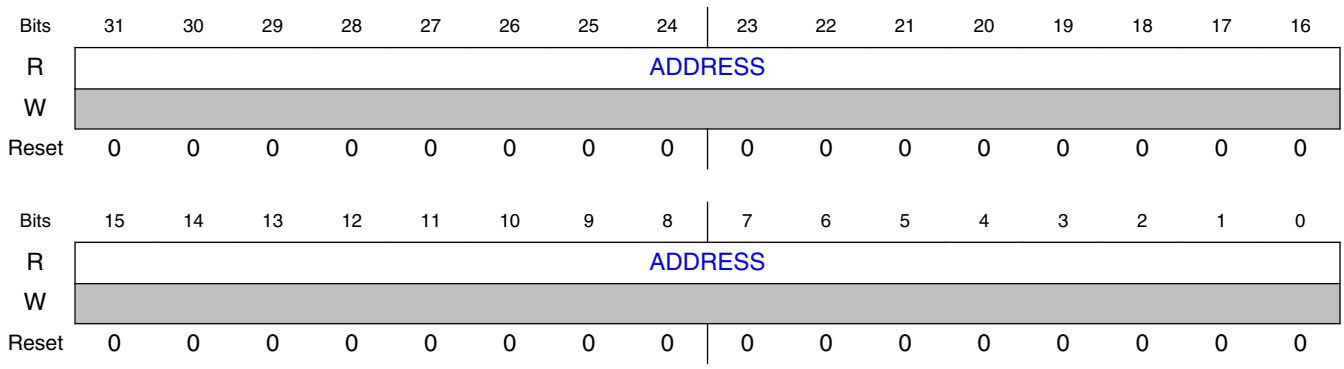
The APBH DMA Channel n buffer address register contains a pointer to the data buffer for the transfer. For immediate forms, the data is taken from this register. This is a byte address which means transfers can start on any byte boundary.

This register holds a pointer to the data buffer in system memory. After the command values have been read into the DMA controller and the device controlled by this channel, then the DMA transfer will begin, to or from the buffer pointed to by this register.

EXAMPLE

```
apbh_chn_bar_t dma_data;
dma_data.ADDRESS = (reg32_t) pDataBuffer;
```

9.5.3.11.3 Diagram



9.5.3.11.4 Fields

Field	Function
31-0 ADDRESS	ADDRESS Address of system memory buffer to be read or written over the AHB bus.

9.5.3.12 APBH DMA Channel n Semaphore Register (CH0_SEMA - CH15_SEMA)

9.5.3.12.1 Offset

For a = 0 to 15:

Register	Offset
CHa_SEMA	140h + (a × 70h)

9.5.3.12.2 Function

The APBH DMA Channel n semaphore register is used to synchronize the ARM platform instruction stream and the DMA chain processing state.

Each DMA channel has an 8 bit counting semaphore that is used to synchronize between the program stream and and the DMA chain processing. DMA processing continues until the DMA attempts to decrement a semaphore that has already reached a value of zero. When the attempt is made, the DMA channel is stalled until software increments the semaphore count.

EXAMPLE

```
// increment semaphore by two
BF_WR(APBH_CHn_SEMA, 0, INCREMENT_SEMA, 2);
// get instantaneous value
current_sema = BF_RD(APBH_CHn_SEMA, 0, PHORE);
```

9.5.3.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								PHORE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								INCREMENT_SEMA							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.5.3.12.4 Fields

Field	Function
31-24 —	- Reserved, always set to zero.
23-16 PHORE	PHORE This read-only field shows the current (instantaneous) value of the semaphore counter.
15-8 —	- Reserved, always set to zero.
7-0 INCREMENT_SEMA	INCREMENT_SEMA The value written to this field is added to the semaphore count in an atomic way such that simultaneous software adds and DMA hardware substracts happening on the same clock are protected. This bit field reads back a value of 0x00. Writing a value of 0x02 increments the semaphore count by two, unless the DMA channel decrements the count on the same clock, then the count is incremented by a net one.

9.5.3.13 AHB to APBH DMA Channel n Debug Information (CH0_DEBUG1 - CH15_DEBUG1)

9.5.3.13.1 Offset

For a = 0 to 15:

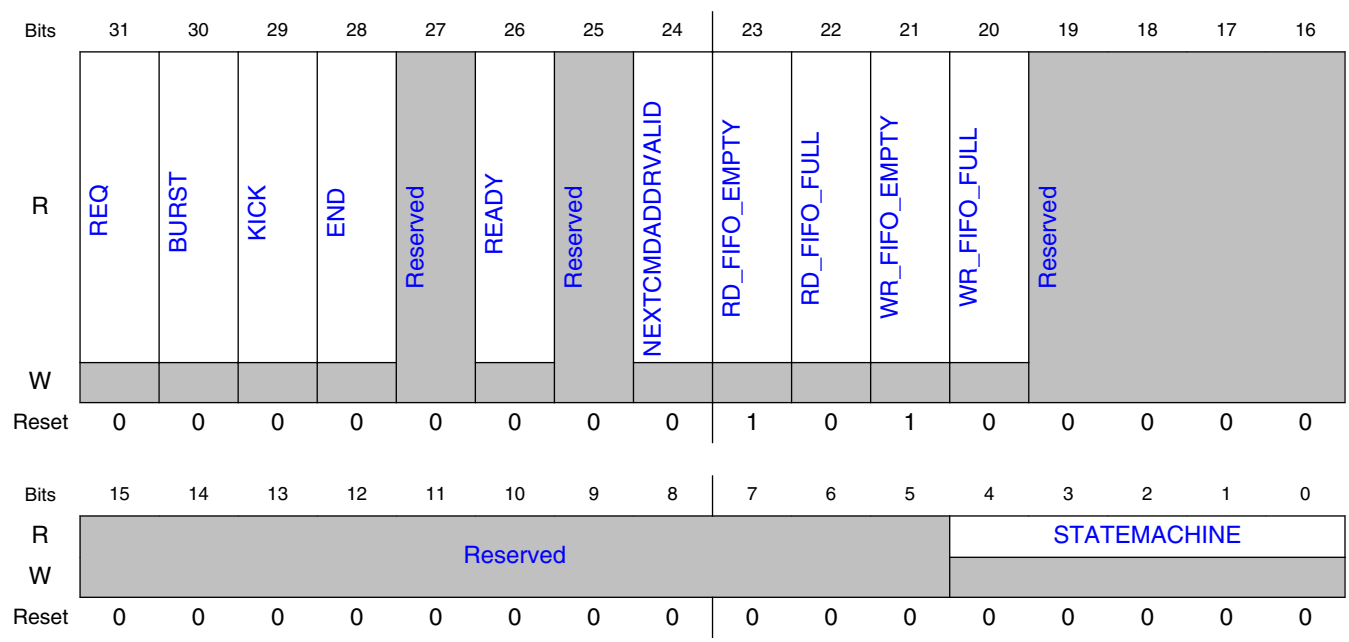
Register	Offset
CHa_DEBUG1	150h + (a × 70h)

9.5.3.13.2 Function

This register gives debug visibility into the APBH DMA Channel n state machine and controls.

This register allows debug visibility of the APBH DMA Channel n.

9.5.3.13.3 Diagram



9.5.3.13.4 Fields

Field	Function
31 REQ	REQ This bit reflects the current state of the DMA Request Signal from the APB device
30 BURST	BURST This bit reflects the current state of the DMA Burst Signal from the APB device
29 KICK	KICK This bit reflects the current state of the DMA Kick Signal sent to the APB Device
28 END	END This bit reflects the current state of the DMA End Command Signal sent from the APB Device
27 —	SENSE This bit is reserved for this DMA Channel and always reads 0.
26 READY	READY This bit is reserved for this DMA Channel and always reads 0.
25 —	LOCK This bit is reserved for this Channel and always reads 0.
24 NEXTCMDADDRVALID	NEXTCMDADDRVALID This bit reflects the internal bit which indicates whether the channel's next command address is valid.
23 RD_FIFO_EMPTY	RD_FIFO_EMPTY This bit reflects the current state of the DMA Channel's Read FIFO Empty signal.
22 RD_FIFO_FULL	RD_FIFO_FULL This bit reflects the current state of the DMA Channel's Read FIFO Full signal.
21 WR_FIFO_EMPTY	WR_FIFO_EMPTY This bit reflects the current state of the DMA Channel's Write FIFO Empty signal.
20 WR_FIFO_FULL	WR_FIFO_FULL This bit reflects the current state of the DMA Channel's Write FIFO Full signal.
19-5 —	RSVD1 Reserved
4-0 STATEMACHINE	STATEMACHINE PIO Display of the DMA Channel n state machine state. 00000b - This is the idle state of the DMA state machine. 00001b - State in which the DMA is waiting to receive the first word of a command. 00010b - State in which the DMA is waiting to receive the third word of a command. 00011b - State in which the DMA is waiting to receive the second word of a command. 00100b - The state machine processes the descriptor command field in this state and branches accordingly. 00101b - The state machine waits in this state for the PIO APB cycles to complete. 00110b - State in which the DMA is waiting to receive the fourth word of a command, or waiting to receive the PIO words when PIO count is greater than 1.

Field	Function
	00111b - This state determines whether another PIO cycle needs to occur before starting DMA transfers. 01000b - During a read transfers, the state machine enters this state waiting for the last bytes to be pushed out on the APB. 01001b - When an AHB read request occurs, the state machine waits in this state for the AHB transfer to complete. 01100b - During DMA Write transfers, the state machine waits in this state until the AHB master arbiter accepts the request from this channel. 01101b - During DMA Read transfers, the state machine waits in this state until the AHB master arbiter accepts the request from this channel. 01110b - Upon completion of the DMA transfers, this state checks the value of the Chain bit and branches accordingly. 01111b - The state machine goes to this state after the DMA transfers are complete, and determines what step to take next. 10100b - When a terminate signal is set, the state machine enters this state until the current AHB transfer is completed. 10101b - When the Wait for Command End bit is set, the state machine enters this state until the DMA device indicates that the command is complete. 11100b - During DMA Write transfers, the state machine waits in this state until the AHB master completes the write to the AHB memory space. 11101b - If HALTONTERMINATE is set and a terminate signal is set, the state machine enters this state and effectively halts. A channel reset is required to exit this state 11110b - If the Chain bit is a 0, the state machine enters this state and effectively halts. 11111b - When the NAND Wait for Ready bit is set, the state machine enters this state until the GPMI device indicates that the external device is ready.

9.5.3.14 AHB to APBH DMA Channel n Debug Information (CH0_DEBUG2 - CH15_DEBUG2)

9.5.3.14.1 Offset

For a = 0 to 15:

Register	Offset
CHa_DEBUG2	160h + (a × 70h)

9.5.3.14.2 Function

This register gives debug visibility for the APB and AHB byte counts for DMA Channel n.

This register allows debug visibility of the APBH DMA Channel n.

9.5.3.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	APB_BYTES															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AHB_BYTES															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.5.3.14.4 Fields

Field	Function
31-16 APB_BYTES	APB_BYTES This value reflects the current number of APB bytes remaining to be transfered in the current transfer.
15-0 AHB_BYTES	AHB_BYTES This value reflects the current number of AHB bytes remaining to be transfered in the current transfer.

9.5.3.15 APBH Bridge Version Register (VERSION)

9.5.3.15.1 Offset

Register	Offset
VERSION	800h

9.5.3.15.2 Function

This register always returns a known read value for debug purposes it indicates the version of the block.

This register indicates the RTL version in use.

EXAMPLE

```
if (APBH_VERSION.B.MAJOR != 3)
  Error();
```

9.5.3.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STEP															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.5.3.15.4 Fields

Field	Function
31-24 MAJOR	MAJOR Fixed read-only value reflecting the MAJOR field of the RTL version.
23-16 MINOR	MINOR Fixed read-only value reflecting the MINOR field of the RTL version.
15-0 STEP	STEP Fixed read-only value reflecting the stepping of the RTL version.

9.6 62BIT Correcting ECC Accelerator (BCH)

9.6.1 Overview

The hardware ECC accelerator provides a forward error-correction function for improving the reliability of various storage media that may be attached to the device.

For example, NAND flash devices use a spare area to store ECC codes to correct some hard bit errors in data stored within the device, allowing higher device yields and, therefore, lower NAND device costs.

The Bose, Ray-Chaudhuri, Hocquenghem (BCH) Encoder and Decoder module is capable of correcting from 2 to 62 single bit errors within a block of data no larger than about 1900 bytes (512 bytes or 1024 bytes are typical) in applications such as protecting data and resources stored on modern NAND flash devices. The correction level in the BCH block is programmable to provide flexibility for varying applications and configurations of flash page size. The design can be programmed to encode protection of

2 to 62 bit errors when writing flash and to correct the corresponding number of errors on decode. The correction level when decoding MUST be programmed to the same correction level as was used during the encode phase.

BCH-codes are a type of block-code, which implies that all error-correction is performed over a block of N-symbols. The BCH operation will be performed over GF(2^{13} = 8192) or GF(2^{14} = 16384), which is the Galois Field consisting of 8191 or 16383 one-bit symbols. BCH-encoding (or encode for any block-code) can be performed by two algorithms: systematic encoding or multiplicative encoding. Systematic encoding is the process of reading all the symbols which constitute a block, dividing continuously these symbols by the generator polynomial for the GF(8192) or GF(16384) and appending the resulting ***t*** parity symbols to the block to create a BCH codeword (where ***t*** is the number of correctable bits).

The BCH encode process creates *t**13 (or *t**14)-bit parity symbols for each data block when the data is written to the flash device. The parity symbols are written to the flash device after the corresponding data block, and together these are collectively called the codeword. The codeword can be used during the decode process to correct errors that occur in either the data or parity blocks.

The BCH decoder processes code words in a 4-step fashion:

1. Syndrome Calculation (SC): This is the process of reading in all of the symbols of the codeword and continuously dividing by the generator polynomial for the field. 2**t* syndromes must be calculated for each codeword and inspection of the syndromes determines if there are errors: a non-zero set of syndromes indicates one or more errors. This process is implemented parallel hardware to minimize processing time since it must be done every time the decode is performed.
2. Key Equation Solver (KES): The syndromes represent 2t-linear equations with 2t-unknown variables. The process of solving these equations and selecting from the numerous solutions constitutes the KES module. When the KES block completes its operations, it generates an error locator polynomial (sigma) that is used in the proceeding block to determine the locations and values of the errors.
3. Chien Search (CS): This block takes input from the KES block and uses the Chien Algorithm for finding the locations of the errors based on the error locator polynomial. The method basically involves substituting all 8191 symbols from the GF(8192) or 16383 symbols from the GF(16383) into the locator polynomial. All evaluations that produce a zero solution indicate locations of the various errors. Since each located error corresponds to a single bit, the bit in the original data may be corrected by simply flipping the polarity of the incorrect location.

4. Correction: this block has to convert the symbol index and mask information to memory byte indexes and masks.

The BCH block, shown in the figure, was designed to operate in a pipelined fashion to maximize throughput. Aside from the initial latency to fill the pipeline stages, the BCH throughput is about 7/4 cycles/byte. Thus, the bottleneck in performing NAND reads and error corrections is the BCH rate. Current GPMI read rates are approximately 1/2 cycles/byte maximally for the current generation of NAND flash. Fortunately, BCH has a different master clock from GPMI, this gives some flexibility to match the throughput rate. The CPU is not directly involved in generating parity symbols, checking for errors, or correcting them.

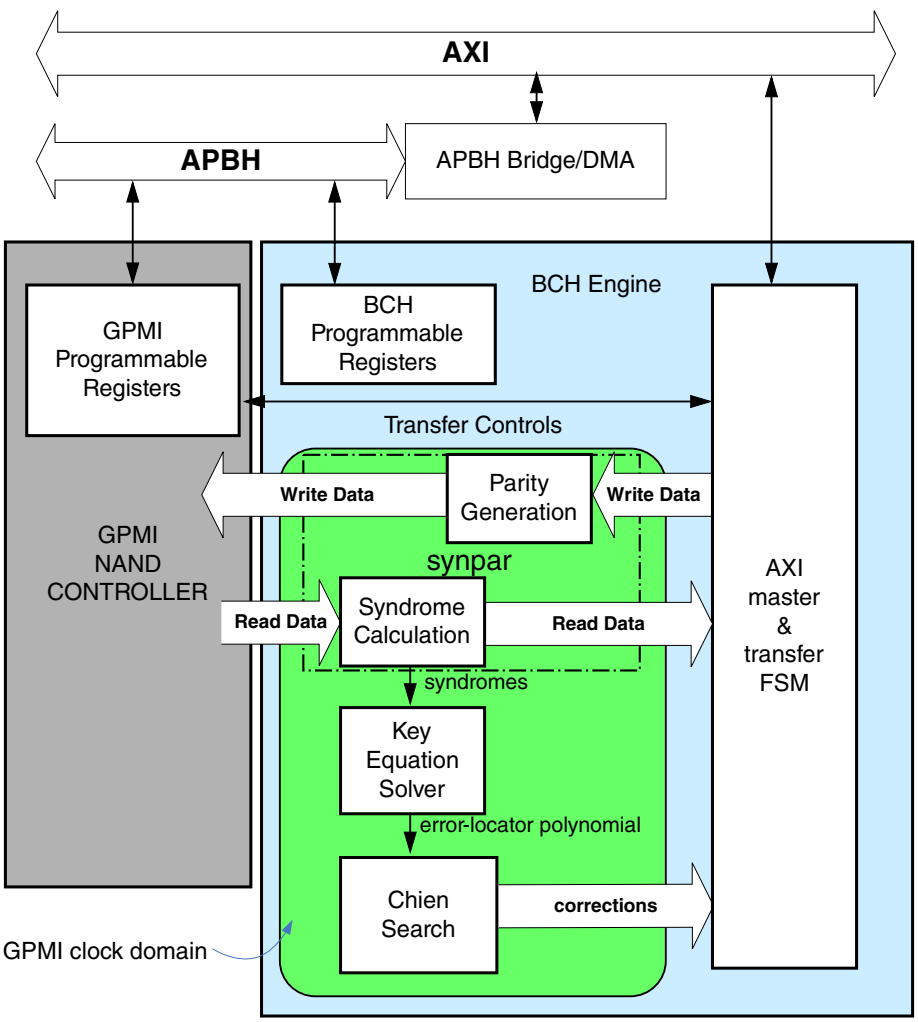


Figure 9-9. Hardware BCH Accelerator

9.6.2 Functional Description

9.6.2.1 Operation

Before performing any NAND flash read or write operations, software should first program the BCH's flash layout registers (see Flash Page Layout) to specify how data is to be formatted on the flash device. The BCH hardware allows full programmability over the flash page layout to enable users flexibility in balancing ECC correction levels and ever-changing flash page sizes.

To initiate a NAND Flash write, software will program a GPMI DMA operation. The DMA need only program the GPMI control registers (and handle the requisite flash addressing handshakes) since the BCH will handle all data operations using its AXI bus interface. The BCH will then send the data to the GPMI controller to be written to flash as it computes the parity symbols. At the end of each data block the BCH will insert the parity symbols into the data stream so that the GPMI sees only a continuous stream of data to be written.

NAND Flash read operations operate in a similar manner. As the GPMI controller reads the device, all data is sent to the BCH hardware for error detection/correction. The BCH controller writes all incoming read data to system memory and in parallel computes the syndromes used to detect bit errors. If errors are detected within a block, the BCH hardware activates the error correction logic to determine where bit errors have occurred and ultimately correct them in the data buffer in system memory. After an entire flash page has been read and corrected, the BCH will signal an interrupt to the CPU.

The figure below indicates how data read from the GPMI is operated on within the BCH hardware. As the BCH receives data from the GPMI (top row), it is written to memory by the BCH's Bus Interface Unit (BIU) (second row). For blocks requiring correction, the KES logic will be activated after the entire block has been received. Once the error locator polynomial has been computed, the corrections are determined by the Chien Search and fed back to the BIU, which performs a read, modify, write operation on the buffer in memory to correct the data.

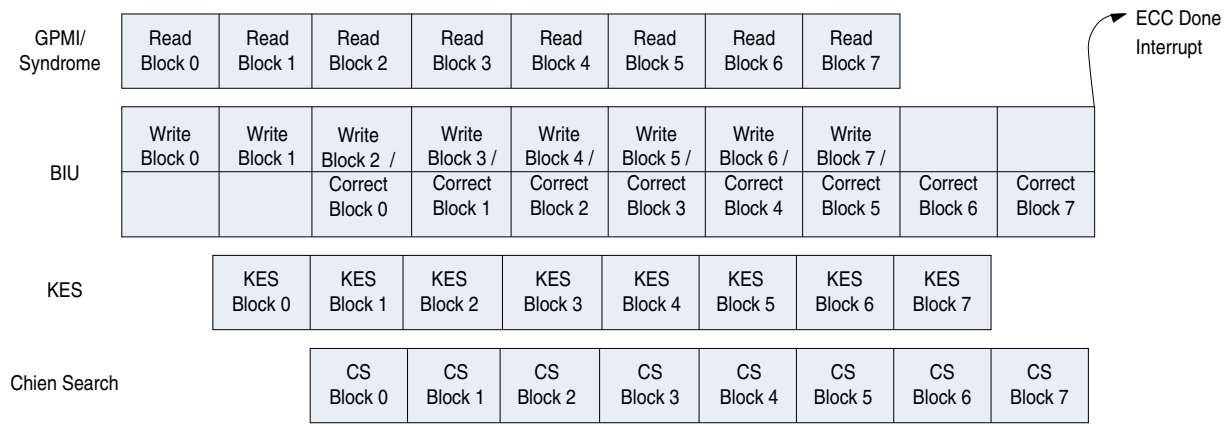


Figure 9-10. Block Pipeline while Reading Flash

9.6.2.1.1 BCH Limitations and Assumptions

- The BCH is programmable to support 2 to 62 bit error correction. ECC0 is supported as a pass-through, non-correcting mode.
- Data block sizes must be a multiple of 4 bytes and be aligned in system memory.
- The BCH supports a programmable number of metadata/auxiliary data bytes, from 0 to 255.
- Metadata will be written at the beginning of the flash page to facilitate fast access for filesystem operations.
- Metadata may be treated as an independent block for ECC purposes or combined with the first data block to conserve bits in the flash.
- The BCH does not support a partial page write (this can be accomplished by programming the BCH layout registers such that the BCH only sees a portion of the page).
- Flash read operations can read the entire page or the first block on the page.
- The BCH also supports a memory-to-memory mode of operation that does not require the use of DMA or the GPMI.

9.6.2.1.2 Flash Page Layout

The BCH supports a fully programmable flash page layout. The BCH maintains four independent layout registers that can describe four completely different NAND devices or layouts.

When the BCH initiates an operation, it selects one of the layouts by using the chip select as an index into the BCH_LAYOUTSELECT register that determines which layout should be used for the operation.

Three possible (generic) flash layout schemes are supported, as indicated in the figure below. (In each case, the metadata size may also be programmed to 0 bytes). Metadata may either be combined with the first block of data or the size of the first data block can be programmed to 0 to allow the metadata to be protected by its own ECC parity bits.

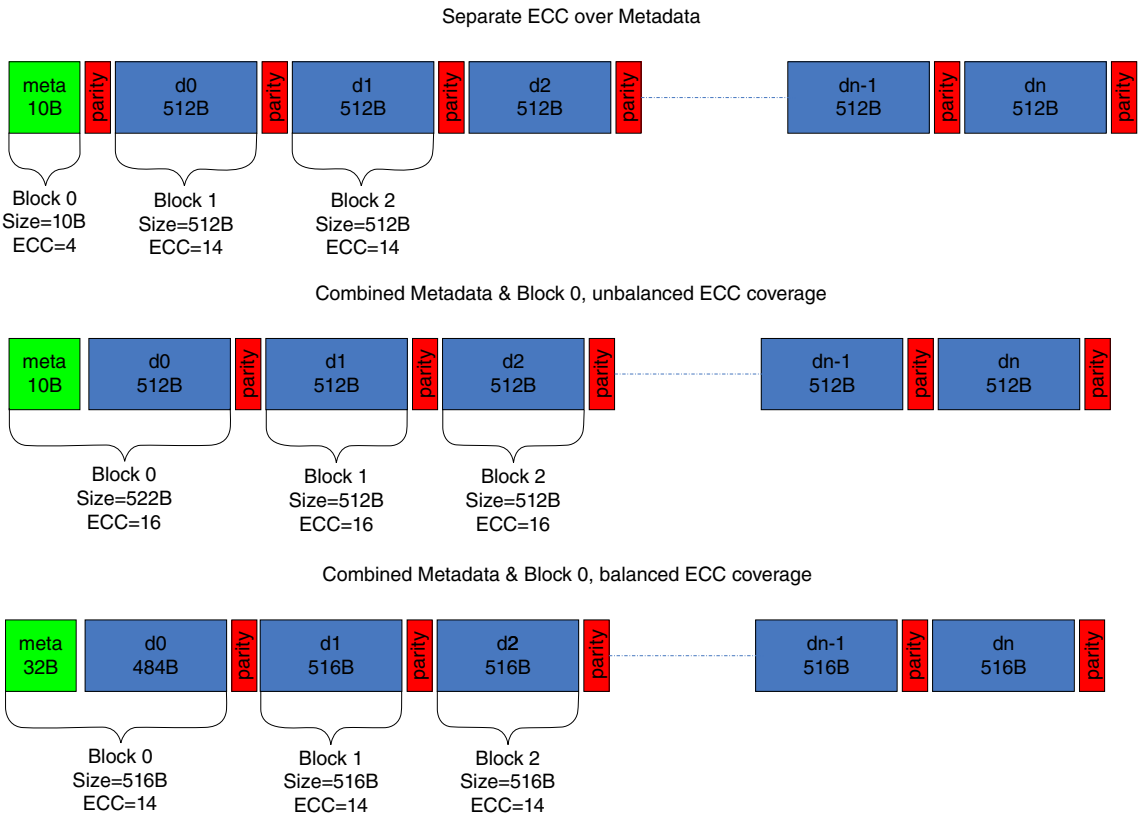


Figure 9-11. FLASH Page Layout Options

Each layout is determined by a pair of registers that define the following parameters:

- DATA0_SIZE: Indicates the number of data bytes in the first block on the page (this should not include parity or metadata bytes). This should be set to 0 when the metadata is to be covered separately with its own ECC. This must be a multiple of 4 bytes.
- ECC0: Indicates the ECC level to be used for the first block on the flash (data0+metadata).
- META_SIZE: Indicates the number of bytes (from 0-255) that are stored as metadata.

- NBLOCKS: Indicates the number of subsequent DATAN blocks on the flash, or the number of blocks following the DATA0 block.
- DATAN_SIZE: Indicates the number of data bytes in all subsequent data blocks. This MUST be a multiple of 4 bytes.
- ECCN: Indicates the ECC level to be used for the subsequent data blocks.
- GF0 or GFN: Indicates the Galois field the meta / data blocks are using
- PAGE_SIZE: Indicates the total number of bytes available per page on the physical flash device. This includes the spare area and is typically 4096+128, 4096+218, or 2048+64 bytes.

9.6.2.1.3 Determining the ECC layout for a device

Since the BCH is programmable, a system can trade off ECC levels for flash size and layout configurations.

The following examples indicate how to determine a valid layout based on the required storage space and flash size. For all cases, the size of the parity will be 13 (or 14 for GF(2^{14}))×ECC level ***bits***-- so for ECC8, 13 (or 14) bytes are required (per block).

9.6.2.1.3.1 4K+218 flash, 10 bytes metadata, 512 byte data blocks, separate metadata, Assuming GF(2^{13})

In this case, we have 8 data blocks each consisting of 512 bytes. Since the flash has 218 spare bytes (1744 bits), first estimate an ECC level for the data blocks by first subtracting the number of metadata bytes from the spare bytes (218 - 10 = 208 bytes = 1664 bits) then dividing the number of bits by 8 (number of blocks) and then by 13 (bits per ECC level).

(218 - 10) × 8 = 1664/13(8) = 16

Therefore all the data blocks could be covered by ECC16 if the metadata had no parity. This isn't acceptable, so assume ECC14 for all the data blocks. Now calculate the number of free bits for the metadata parity as

1664 - (14) × 13 × 8 = 208

Therefore, 208 bits remain for metadata parity. Dividing by 13 (bits/ECC) gives 16, so the metadata can be covered with ECC16. The settings for this device would then be

Table 9-11. Settings for 4K+218 FLASH

Setting	Value
PAGE_SIZE	4096+218=4314=0x10DA
META_SIZE	10=0x0A
DATA0_SIZE	0
ECC0	16=0x10
GF0	GF(2^{13})
DATAN_SIZE	512=0x200 (in register interface, assigned as 0x80)
ECCN	14=0x0E
GFN	GF(2^{13})
NBLOCKS	8

9.6.2.1.3.2 4K+128 flash, 10 bytes metadata, 1024 byte data blocks, separate metadata, assuming GF(2^{13}) for data and GF(2^{14}) for metadata

This flash will have 118 bytes available for ECC (after subtracting the metadata size), therefore, 994 bits.

Dividing by 4×14 (number of blocks × ECC level) we get 17.75, therefore we can support ECC16 on the data blocks. The number of free spare bits becomes 944 - 16 × 4 × 14 = 944 - 896 = 48, divided by 13 = 3.69, therefore the metadata can be also covered by ECC2.

Table 9-12. Settings for 4K+128 FLASH

Setting	Value
PAGE_SIZE	4096+128=4224=0x1080
META_SIZE	10=0x0A
DATA0_SIZE	0
ECC0	2
GF0	GF(2^{13})
DATAN_SIZE	1024=0x400 (in register interface, assigned as 0x100)
ECCN	16
GFN	GF(2^{14})
NBLOCKS	4

In this case, there will be additional unused spare bits, with the BCH will pad out with zeros.

9.6.2.1.4 Data Buffers in System Memory

While the data on the flash is interleaved with parity symbols, the BCH assumes that the data buffers in memory are contiguous.

Metadata read from the flash will be stored to the location pointed to by the GPMI_AUXILIARY register and data will be written to the address specified in the GPMI_PAYLOAD register as is shown in the following figure where the block length is 512 bytes for example. Since the number of blocks on a flash page is programmable, the BCH also writes individual block correction status to the auxiliary pointer at the word-aligned address following the end of the metadata. Optionally, the computed syndromes may also be written to the auxiliary area if the DEBUGSYNDROME bit is set in the control register.

As blocks complete processing, the bus master will accumulate the status for each block and write it to the auxiliary data buffer following the metadata. The metadata area will be padded with 0's until the next word boundary and the status for blocks 0-3 will be written to the next word. The status for subsequent blocks will then be written to the buffer. The status for the first block (metadata block) is also stored in the STATUS_BLK0 register in the BCH_STATUS register. The completion codes for the blocks are indicated in the Table 9-13. Note that the definition of the bytes and their ordering in the auxiliary and payload storage areas are user defined. When this data is read back from the flash and put into memory, it will resemble the original buffer that was written out to the flash.

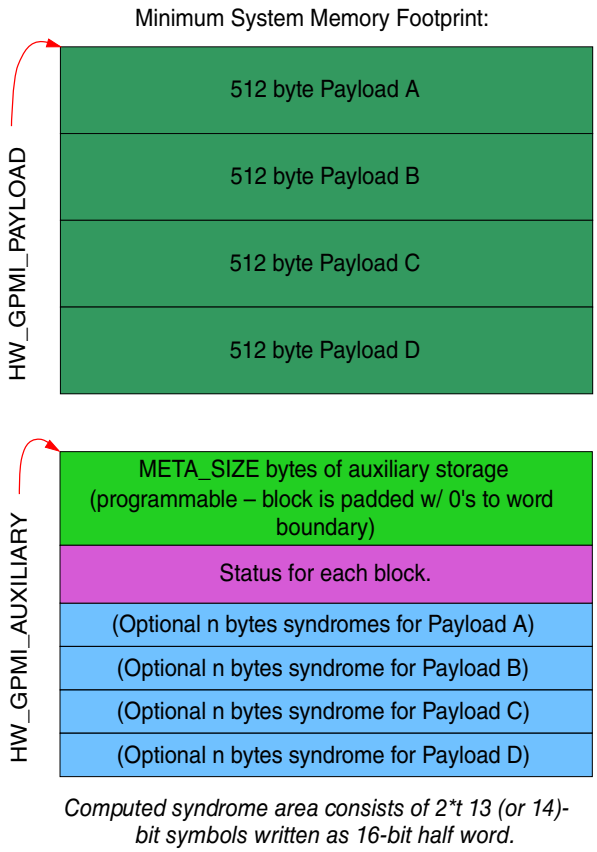
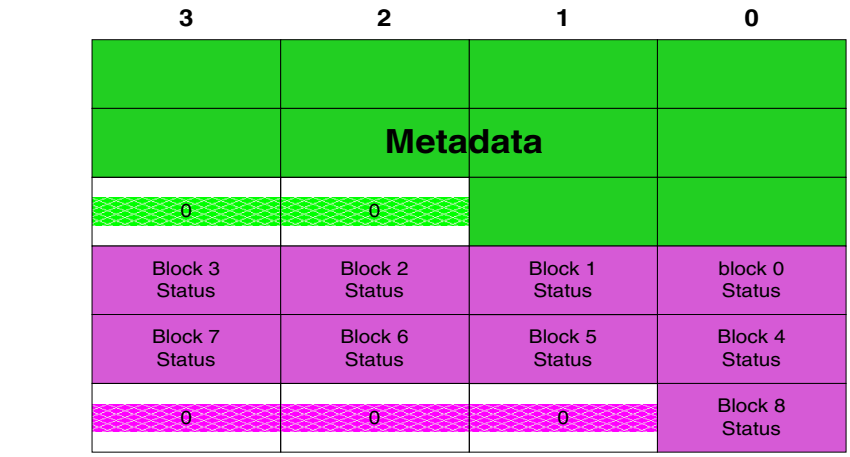


Figure 9-12. BCH Data Buffers in Memory

Table 9-13. Status Block Completion Codes

Code	Description
0xFF	Block is erased
0xFE	Block is uncorrectable
0x00	No errors found
0x01-0x3E	Number of errors corrected

The following figure shows the layout of the bytes within the status field.



Status bytes are allocated based on the NBLOCKS programmed into the flash format register. The number of status bytes are computed by NBLOCK+1. The status area will be padded with zeros to the next word boundary.

Syndrome data written for debug purposes follows the end of the status block.

Figure 9-13. Memory-to-Memory Operations

9.6.2.2 Memory to Memory (Loopback) Operation

The BCH supports a memory-to-memory mode of operation where both the encoded and decoded buffers reside in system memory.

This can be useful for applications where data must be protected by ECC, but the storage device does not reside on the GPMI bus.

The BCH operation in memory to memory mode is much simpler than in GPMI mode since DMAs are not required to manage the operation. Instead, software simply writes the BCH_DATAPTR and BCH_METAPTR with the addresses of the data and metadata (auxiliary) buffers and the BCH_ENCODEPTR with the address of the buffer for encoded data. To initiate the operation, software simply sets the M2M_ENCODE and M2M_ENABLE bits in the control register. The BCH can be programmed to either issue an interrupt at the end of the operation or software may poll the status bits for completion.

Memory to memory decode operations work in a similar manner. The encoded data address is written to the BCH_ENCODEPTR and the data and meta pointers are written to buffers that correspond to the desired decoded data addresses. To initiate a decode, software must set the M2M_ENCODE bit to 0 while writing the M2M_ENABLE bit. Note that the addresses written to the BCH_DATAPTR, BCH_METAPTR and BCH_ENCODEPTR registers should always be aligned on a 4 byte boundary. In other words, the 2 lower bits of the address should always be written with zeros.

9.6.2.3 Programming the BCH/GPMI Interfaces

Programming the BCH for NAND operations consists largely of disabling the soft reset and clock bits (SFTRST and CLKGATE) from the BCH_CTRL register and then programming the flash layout registers to correspond to the format of the attached NAND device(s).

The BCH_LAYOUTSELECT register should also be programmed to map the chip select of each attached device into one of the four layout registers.

The bulk of the programming is actually applied to the GPMI through PIO operations embedded in DMA command structures. The DMA will perform all the requisite handshaking with the GPMI interface to negotiate the address portion of the transfer, then the BCH will handle all the movement of data from memory to the GPMI (writes) or the GPMI to memory (reads). The BCH will direct all data blocks to the buffer pointed to by the PAYLOAD_BUFFER and the metadata will be written to the AUXILIARY_BUFFER. Both of these registers are located in the GPMI PIO data space and are communicated to the BCH hardware at the beginning of the transfer. Thus, the normal multi-NAND DMA based device interleaving is preserved, that is, four NANDs on four separate chip selects can be scheduled for read or write operations using the BCH. Whichever channel finishes its ready wait first and enters the DMA arbiter with its lock bit set owns the GPMI command interface and through it owns the BCH resources for the duration of its processing.

9.6.2.3.1 BCH Encoding for NAND Writes

The BCH encoder flowchart in Figure 9-14 shows the detailed steps involved in programming and using the BCH encoder. This flowchart shows how to use the BCH block with the GPMI.

To use the BCH encoder with the GPMI's DMA, create a DMA command chain containing ten descriptor structures, as shown in Figure 9-16 and detailed in the DMA structure code example that follows it in DMA Structure Code Example. The ten descriptors perform the following tasks:

1. Disable the BCH block (in case it was enabled) and issue NAND write setup command byte (under CLE) and address bytes (under ALE).
2. Configure and enable the BCH and GPMI blocks to perform the NAND write.
3. Disable the BCH block and issue NAND write execute command byte (under CLE).
4. Wait for the NAND device to finish writing the data by watching the ready signal.

5. Check for NAND timeout through PSENSE.
6. Issue NAND status command byte (under CLE).
7. Read the status and compare against expected.
8. If status is incorrect or incomplete, branch to error handling descriptor chain.
9. Otherwise, write is complete and emit GPMI interrupt.

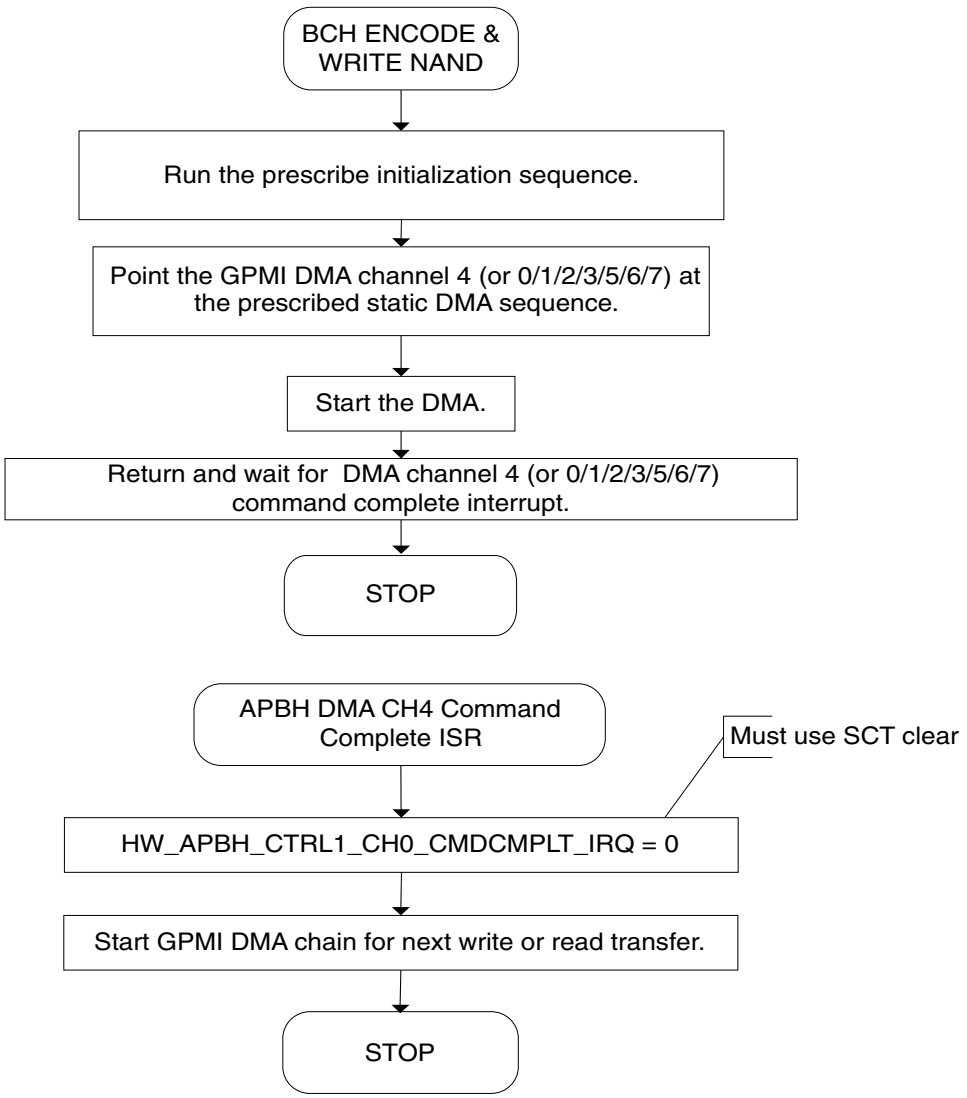


Figure 9-14. BCH Encode Flowchart

Descriptor Legend

NEXT CMD ADDR										
CMD	<=	xfer_count	cmdwords	wait4endcmd	semaphore	nandwait4ready	nandlock	irqoncmplt	chain	command
BUFFER ADDR										
HW_GPMI_CTRL0	<=	command_mode	word_length	lock_cs	CS	address	address_increment	xfer_count		
HW_GPMI_COMPARE	<=	mask				reference				
HW_GPMI_ECCCTRL	<=	ecc_cmd			enable_ecc				buffer_mask	
HW_GPMI_ECCCOUNT										
HW_GPMI_PAYLOAD										
HW_GPMI_AUXILIARY										

Figure 9-15. BCH DMA Descriptor Legend

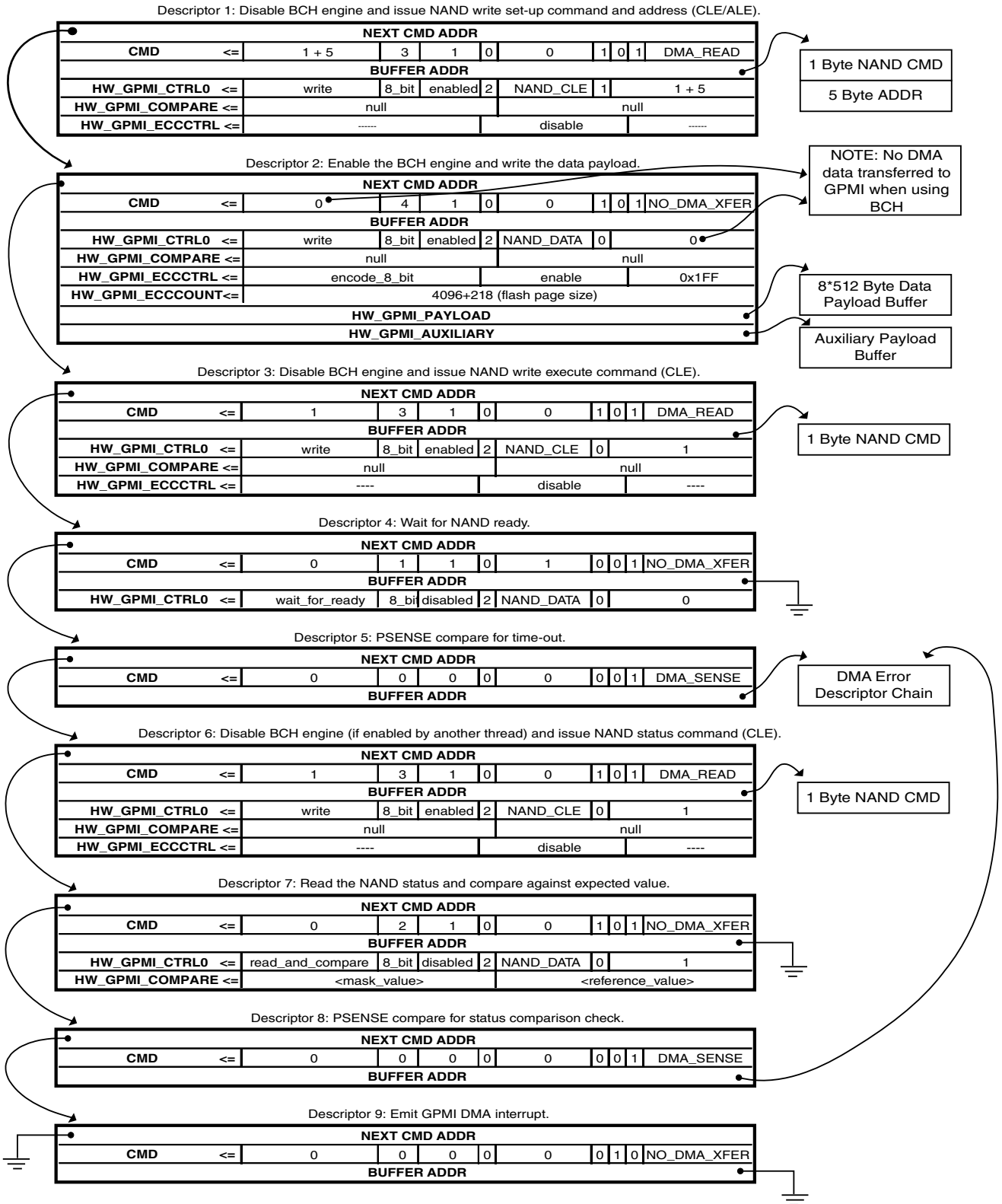


Figure 9-16. BCH Encode DMA Descriptor Chain

9.6.2.3.1.1 DMA Structure Code Example

The following code sample illustrates the coding for one write transaction involving 4096 bytes of data payload (eight 512-byte blocks) and 10 bytes of auxiliary payload (also referred to as metadata) to a 4K NAND page sitting on GPMI CS2.

```
//-----------------------------------------------------------------------------
// generic DMA/GPMI/ECC descriptor struct, order sensitive!
//-----------------------------------------------------------------------------
typedef struct {
  // DMA related fields
  unsigned int dma_nxtcmdar;
  unsigned int dma_cmd;
  unsigned int dma_bar;
  // GPMI related fields
  unsigned int gpmi_ctrl0;
  unsigned int gpmi_compare;
  unsigned int gpmi_eccctrl;
  unsigned int gpmi_ecccount;
  unsigned int gpmi_data_ptr;
  unsigned int gpmi_aux_ptr;
} GENERIC_DESCRIPTOR;
//-----------------------------------------------------------------------------
// allocate 10 descriptors for doing a NAND ECC Write
//-----------------------------------------------------------------------------
GENERIC_DESCRIPTOR write[10];
//-----------------------------------------------------------------------------
// DMA descriptor pointer to handle error conditions from psense checks
//-----------------------------------------------------------------------------
unsigned int * dma_error_handler;
//-----------------------------------------------------------------------------
// 8 byte NAND command and address buffer
// any alignment is ok, it is read by the GPMI DMA
//   byte 0 is write setup command
//   bytes 1-5 is the NAND address
//   byte 6 is write execute command
//   byte 7 is status command
//-----------------------------------------------------------------------------
unsigned char nand_cmd_addr_buffer[8];
//-----------------------------------------------------------------------------
// 4096 byte payload buffer used for reads or writes
// needs to be word aligned
//-----------------------------------------------------------------------------
unsigned int write_payload_buffer[(4096/4)];
//-----------------------------------------------------------------------------
// 65 byte meta-data to be written to NAND
// needs to be word aligned
//-----------------------------------------------------------------------------
unsigned int write_aux_buffer[65];
//-----------------------------------------------------------------------------
// Descriptor 1: issue NAND write setup command (CLE/ALE)
//-----------------------------------------------------------------------------
write[0].dma_nxtcmdar = &write[1];                          // point to the next descriptor
write[0].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1 + 5)|   // 1 byte command, 5 byte address
                   BF_APBH_CHn_CMD_CMDWORDS      (3)    |   // send 3 words to the GPMI
                 BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)    |   // wait for command to finish
before
                                                            //    continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)    |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)    |
                   BF_APBH_CHn_CMD_NANDLOCK      (1)    |   // prevent other DMA channels
from
                                                            //    taking over
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0)    |
                   BF_APBH_CHn_CMD_CHAIN         (1)    |   // follow chain to next command
```

```
                BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ); // read data from DMA, write to
NAND
write[0].dma_bar = &nand_cmd_addr_buffer;           // byte 0 write setup, bytes 1 - 5 NAND
address
// 3 words sent to the GPMI
write[0].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)    | // write to the NAND
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)    |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)  |
                    BF_GPMI_CTRL0_CS              (2)      | // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE) |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (1)       | // send command and
address
                    BF_GPMI_CTRL0_XFER_COUNT       (1 + 5);    // 1 byte command, 5 byte
address
write[0].gpmi_compare = NULL;                            // field not used but necessary to
set eccctrl
write[0].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE); // disable the ECC block
//--------------------------------------------------------------------------
// Descriptor 2: write the data payload (DATA)
//--------------------------------------------------------------------------
write[1].dma_nxtcmdar = &write[2];                       // point to the next descriptor
write[1].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT      (0)|  // NOTE: No DMA data transfer
                   BF_APBH_CHn_CMD_CMDWORDS        (4)|  // send 4 words to the GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD     (1)|  // Wait to end
                   BF_APBH_CHn_CMD_SEMAPHORE       (0)|
                   BF_APBH_CHn_CMD_NANDWAIT4READY  (0)|
                   BF_APBH_CHn_CMD_NANDLOCK        (1)|  // maintain resource lock
                   BF_APBH_CHn_CMD_IRQONCMPLT      (0)|
                   BF_APBH_CHn_CMD_CHAIN           (1)|  // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, DMA_NO_XFER);  // No data transferred
write[1].dma_bar = &write_payload_buffer;                  // pointer for the 4K byte
data area
// 4 words sent to the GPMI
write[1].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)    | // write to the NAND
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)    |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)  |
                    BF_GPMI_CTRL0_CS                (2)       | // must correspond to NAND
CS used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA)|
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)       |
                      BF_GPMI_CTRL0_XFER_COUNT         (0);        // NOTE: this field
contains
                                                                 //  the total amount
                              // DMA transferred to GPMI via DMA (0)!
write[1].gpmi_compare = NULL;                              // field not used but necessary
to
set eccctrl
write[1].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ECC_CMD,    ENCODE_8_BIT) | // specify t = 8
mode
                        BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, ENABLE)       | // enable ECC module
                     BF_GPMI_ECCCTRL_BUFFER_MASK   (0x1FF);          // write all 8 data
blocks
                                                                   //   and 1 aux block
write[1].gpmi_ecccount = BF_GPMI_ECCCOUNT_COUNT(4096+218);      // specify number of bytes
                                                                //   written to NAND
write[1].gpmi_data_pointer = &write_payload_pointer;            // data buffer address
write[1].gpmi_aux_pointer  = &write_aux_pointer;                // metadata pointer
//--------------------------------------------------------------------------
// Descriptor 3: issue NAND write execute command (CLE)
//--------------------------------------------------------------------------
write[2].dma_nxtcmdar = &write[3];                         // point to the next descriptor
write[2].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1) |     // 1 byte command
                   BF_APBH_CHn_CMD_CMDWORDS      (3) |     // send 3 words to the GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1) |     // wait for command to finish
before
                                                          //    continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0) |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0) |
                   BF_APBH_CHn_CMD_NANDLOCK      (1) |     // maintain resource lock
```

```
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0) |
                  BF_APBH_CHn_CMD_CHAIN         (1) |     // follow chain to next command
                BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ);   // read data from DMA, write to
NAND
write[2].dma_bar = &nand_cmd_addr_buffer[6];             // point to byte 6, write execute
command
// 3 words sent to the GPMI
write[2].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)    | // write to the NAND
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)    |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)  |
                    BF_GPMI_CTRL0_CS               (2)      | // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE) |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)       |
                      BF_GPMI_CTRL0_XFER_COUNT         (1);       // 1 byte command
write[2].gpmi_compare = NULL;                           // field not used but necessary to set
eccctrl
write[2].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE); // disable the ECC block
//--------------------------------------------------------------------------
// Descriptor 4: wait for ready (CLE)
//--------------------------------------------------------------------------
write[3].dma_nxtcmdar = &write[4];                       // point to the next descriptor

write[3].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0) |   // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS      (1) |   // send 1 word to the GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD   (1) |   // wait for command to finish before
                                                         //      continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0) |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(1) |   // wait for nand to be ready
                   BF_APBH_CHn_CMD_NANDLOCK      (0) |   // relinquish nand lock
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0) |
                   BF_APBH_CHn_CMD_CHAIN         (1) |   // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);          // no dma transfer
write[3].dma_bar = NULL;                                                // field not used
// 1 word sent to the GPMI
write[3].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WAIT_FOR_READY) | // wait for NAND
ready
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)         |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)      |
                  BF_GPMI_CTRL0_CS              (2)         | // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA)     |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)            |
                      BF_GPMI_CTRL0_XFER_COUNT         (0);
//--------------------------------------------------------------------------
// Descriptor 5: psense compare (time out check)
//--------------------------------------------------------------------------
write[4].dma_nxtcmdar = &write[5];                         // point to the next descriptor
write[4].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)       | // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS      (0)       | // no words sent to GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD   (0)       | // do not wait to continue
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)       |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)       |
                   BF_APBH_CHn_CMD_NANDLOCK      (0)       |
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0)       |
                   BF_APBH_CHn_CMD_CHAIN         (1)       | // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, DMA_SENSE);  // perform a sense check
write[4].dma_bar = dma_error_handler;                // if sense check fails, branch to error
handler
//--------------------------------------------------------------------------
// Descriptor 6: issue NAND status command (CLE)
//--------------------------------------------------------------------------
write[5].dma_nxtcmdar = &write[6];                        // point to the next descriptor
write[5].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1)    |  // 1 byte command
                   BF_APBH_CHn_CMD_CMDWORDS      (3)    |  // send 3 words to the GPMI
                 BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)    |  // wait for command to finish
before
continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)    |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)    |
```

```
                    BF_APBH_CHn_CMD_NANDLOCK      (1)    |  // prevent other DMA channels from
taking over
                     BF_APBH_CHn_CMD_IRQONCMPLT    (0)    |
                     BF_APBH_CHn_CMD_CHAIN         (1)    |  // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ);  // read data from DMA, write to
NAND
write[5].dma_bar = &nand_cmd_addr_buffer[7];                 // point to byte 7, status
command
write[5].gpmi_compare = NULL;                              // field not used but necessary to set
eccctrl
write[5].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE); // disable the ECC block
// 3 words sent to the GPMI
write[5].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)    | // write to the NAND
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)    |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)  |
                    BF_GPMI_CTRL0_CS               (2)     | // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE) |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)       |
                      BF_GPMI_CTRL0_XFER_COUNT         (1);        // 1 byte command
//--------------------------------------------------------------------------
// Descriptor 7: read status and compare (DATA)
//--------------------------------------------------------------------------
write[6].dma_nxtcmdar = &write[7];                           // point to the next descriptor
write[6].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)  |    // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS      (2)  |    // send 2 words to the GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)  |    // wait for command to finish
before
                                                           //      continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)  |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)  |
                   BF_APBH_CHn_CMD_NANDLOCK      (1)  |    // maintain resource lock
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0)  |
                   BF_APBH_CHn_CMD_CHAIN         (1)  |    // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);  // no dma transfer
write[6].dma_bar = NULL;                                     // field not used
// 2 word sent to the GPMI
write[6].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, READ_AND_COMPARE) | // read from the
                                                                  //       NAND and
//                                                               // compare to expect
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)     |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)  |
                   BF_GPMI_CTRL0_CS             (2)       |   // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA) |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)        |
                      BF_GPMI_CTRL0_XFER_COUNT         (1);
write[6].gpmi_compare = <MASK_AND_REFERENCE_VALUE>;  // NOTE: mask and reference values are
NAND
                                                          // SPECIFIC to evaluate the NAND
status
//--------------------------------------------------------------------------
// Descriptor 8: psense compare (time out check)
//--------------------------------------------------------------------------
write[7].dma_nxtcmdar = &write[8];                             // point to the next descriptor
write[7].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)         | // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS      (0)         | // no words sent to GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD   (0)         | // do not wait to continue
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)         |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)         |
                   BF_APBH_CHn_CMD_NANDLOCK      (0)         | // relinquish nand lock
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0)         |
                   BF_APBH_CHn_CMD_CHAIN         (1)         | // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, DMA_SENSE);  // perform a sense check
write[7].dma_bar = dma_error_handler;              // if sense check fails, branch to error
handler
//--------------------------------------------------------------------------
// Descriptor 9: emit GPMI interrupt
//--------------------------------------------------------------------------
write[8].dma_nxtcmdar = NULL;                                // not used since this is
```

```
last
descriptor
write[8].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT     (0)          | // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS       (0)          | // no words sent to GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD    (0)          | // do not wait to continue
                   BF_APBH_CHn_CMD_SEMAPHORE      (0)          |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)           |
                   BF_APBH_CHn_CMD_NANDLOCK       (0)          |
                   BF_APBH_CHn_CMD_IRQONCMPLT     (1)          | // emit GPMI interrupt
                   BF_APBH_CHn_CMD_CHAIN          (0)          | // terminate DMA chain
processing
                   BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);  // no dma transfer
```

9.6.2.3.1.2 Using the BCH Encoder

To use the BCH encoder, first turn off the module-wide soft reset bit in both the GPMI and BCH blocks before starting any DMA activity.

Turning off the soft reset must take place by itself, prior to programming the rest of the control registers. Turn off the BCH bus master soft reset bit. Turn off the clock gate bits.

Program the remainder of the GPMI, BCH and APBH DMA as follows:

```
// bring APBH out of reset
APBH_CTRL0_CLR(BM_APBH_CTRL0_SFRST);
APBH_CTRL0_CLR(BM_APBH_CTRL0_CLKGATE);

// bring BCH out of reset
BCH_CTRL_CLR(BM_BCH_CTRL_SFTRST);
BCH_CTRL_CLR(BM_BCH_CTRL_CLKGATE);

// bring gpmi out of reset
GPMI_CTRL0_CLR(BM_GPMI_CTRL0_SFTRST);
GPMI_CTRL0_CLR(BM_GPMI_CTRL0_CLKGATE);
GPMI_CTRL1_SET(BM_GPMI_CTRL1_DEV_RESET | // deassert reset
                   BM_GPMI_CTRL1_BCH_MODE  );    // enable BCH mode

// enable pinctrl
PINCTRL_CTRL_WR(0x00000000);

// enable gpmi pins
PINCTRL_MUXSEL0_CLR(0x0000ffff);    // data bits
PINCTRL_MUXSEL1_CLR(0x03ffffff);    // control bits
PINCTRL_MUXSEL8_CLR(0x0003f3ff);    // control bits
PINCTRL_MUXSEL8_SET(0x00015155);    // control bits
```

Note that for writing NANDs (ECC encoding), only GPMI DMA command complete interrupts are used. The BCH engine is used for writing to the NAND but may optionally produces an interrupt. From the sample code in DMA Structure Code Example :

- DMA descriptor 1 prepares the NAND for data write by using the GPMI to issue a write setup command byte under CLE, then sends a 5-byte address under ALE. The BCH engine is disabled and not used for these commands.
- DMA descriptor 2 enables the BCH engine for encoding to begin the initial writing of the NAND data by specifying where the data and auxiliary payload are coming from in system memory.

- DMA descriptor 3 issues the write commit command byte under CLE to the NAND.
- DMA descriptor 4 waits for the NAND to complete the write commit/transfer by watching the NAND's ready line status. This descriptor relinquishes the NANDLOCK on the GPMI to enable the other DMA channels to initiate NAND transactions on different NAND CS lines.
- DMA descriptor 6 issues a NAND status command byte under "CLE" to check the status of the NAND device following the page write.
- DMA descriptor 7 reads back the NAND status and compares the status with an expected value. If there are differences, then the DMA processing engine follows an error-handling DMA descriptor path.
- DMA descriptor 8 disables the BCH engine and emits a GPMI interrupt to indicate that the NAND write has been completed.

9.6.2.3.2 BCH Decoding for NAND Reads

When a page is read from NAND flash, BCH syndromes will be computed and, if correctable errors are found, they will be corrected on a per block basis within the NAND page. This decoding process is fully overlapped with other NAND data reads and with CPU execution. The BCH decoder flowchart in the figure below shows the steps involved in programming the decoder. The hardware flow of reading and decoding a 4096-byte page is shown in Figure 9-18.

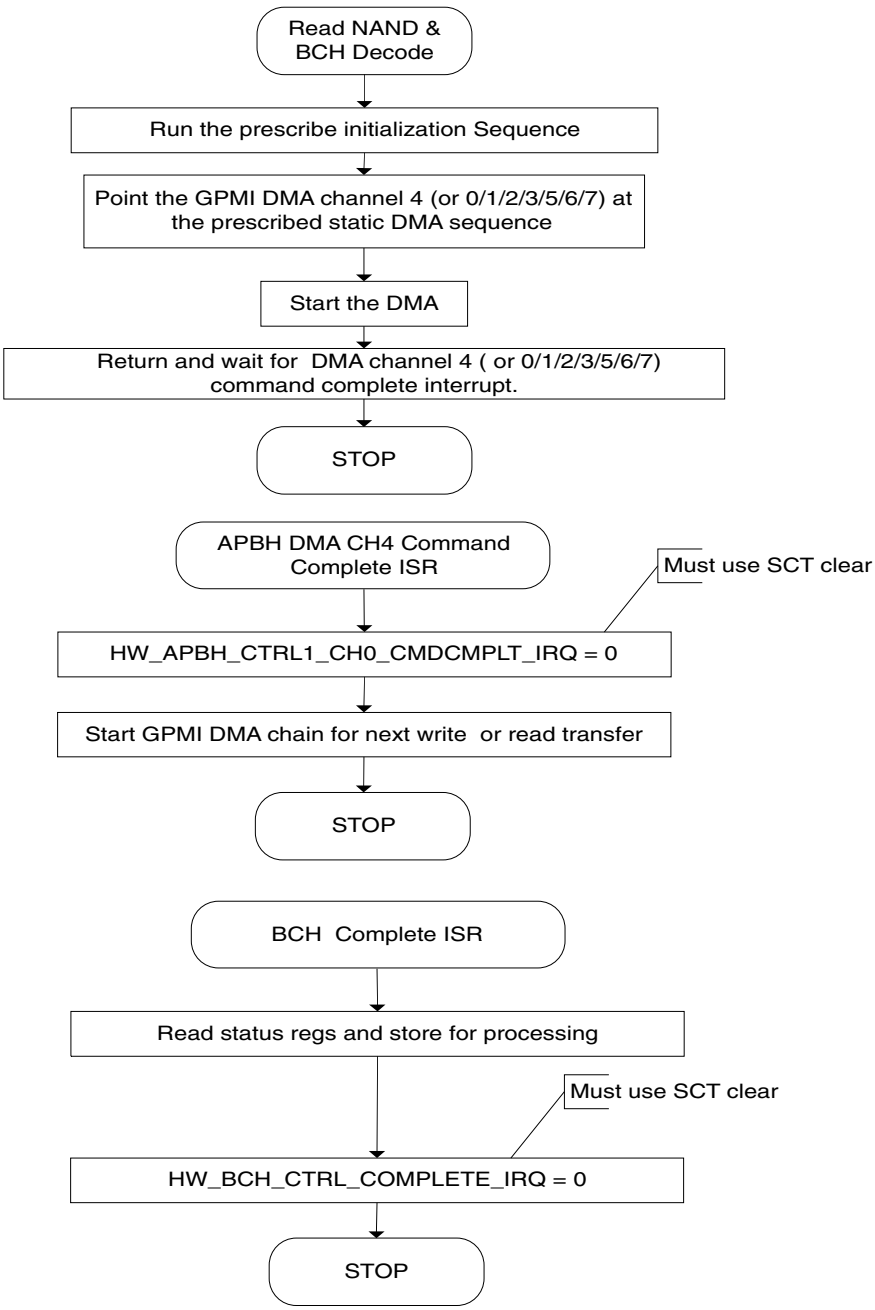


Figure 9-17. BCH Decode Flowchart

Conceptually, an APHB DMA Channel (0,1,2 or 3) command chain with seven command structures linked together is used to perform the BCH decode operation (as shown in Figure 9-18).

Note

The GPMI's DMA command structures controls the BCH decode operation.

To use the BCH decoder with the GPMI's DMA, create a DMA command chain containing seven descriptor structures, as shown in the figure below and detailed in the DMA structure code example that follows it in DMA Structure Code Example. The seven DMA descriptors perform the following tasks:

1. Issue NAND read setup command byte (under "CLE") and address bytes (under "ALE").
2. Issue NAND read execute command byte (under "CLE").
3. Wait for the NAND device to complete accessing the block data by watching the ready signal.
4. Check for NAND timeout through "PSENSE".
5. Configure and enable the BCH block and read the NAND block data.
6. Disable the BCH block.
7. Descriptor NOP to allow NANDLOCK in the previous descriptor to the thread-safe.

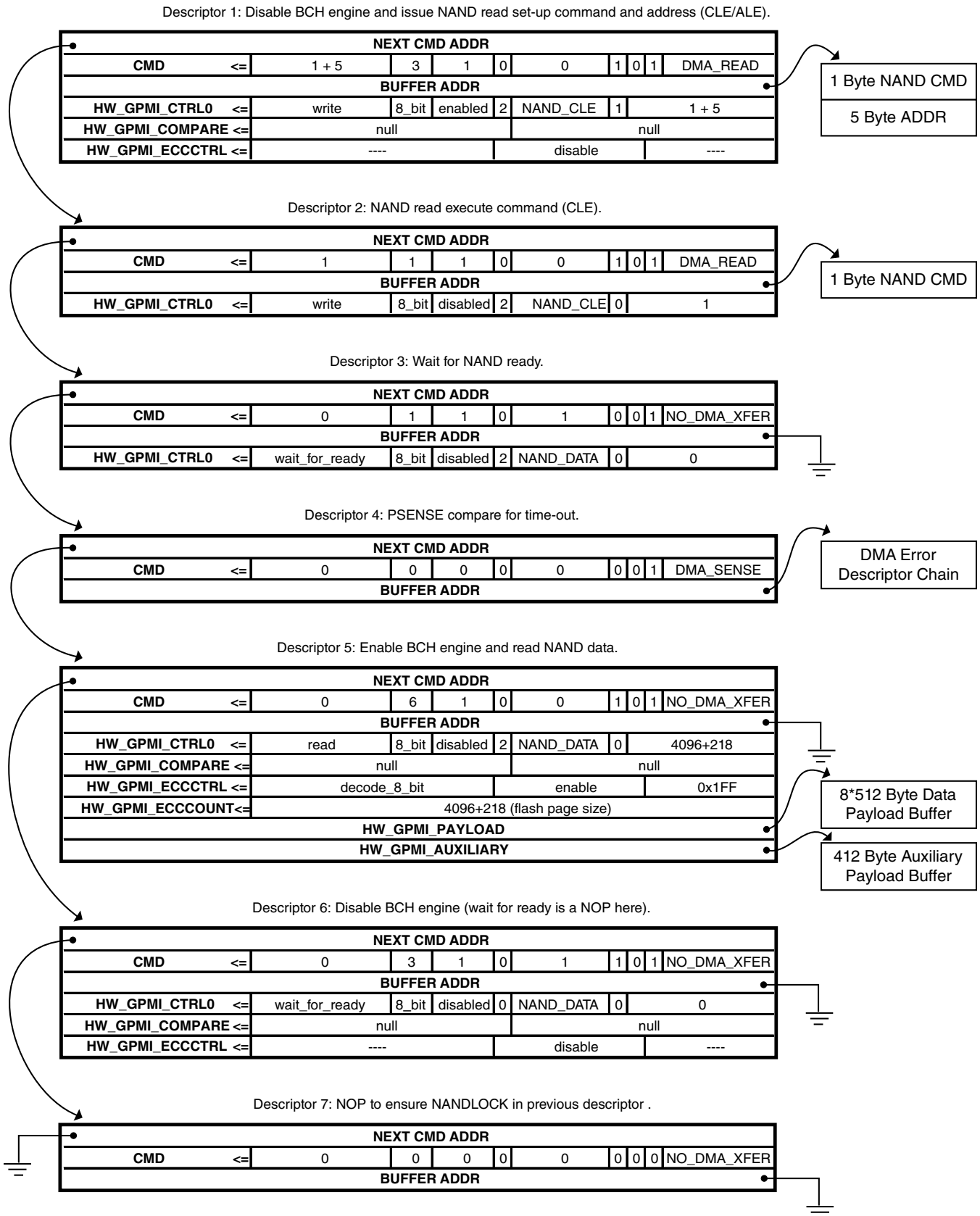


Figure 9-18. BCH Decode DMA Descriptor Chain

9.6.2.3.2.1 DMA Structure Code Example

The following sample code illustrates the coding for one read transaction, consisting of a seven DMA command structure chain for reading all 4096 bytes of payload data (eight 512-byte blocks) and 65 bytes of metadata with the associative parity bytes (8 * (18) + 9) from a 4K NAND page sitting on GPMI CS2.

```
//-----------------------------------------------------------------------------
// generic DMA/GPMI/ECC descriptor struct, order sensitive!
//-----------------------------------------------------------------------------
typedef struct {
  // DMA related fields
  unsigned int dma_nxtcmdar;
  unsigned int dma_cmd;
  unsigned int dma_bar;
  // GPMI related fields
  unsigned int gpmi_ctrl0;
  unsigned int gpmi_compare;
  unsigned int gpmi_eccctrl;
  unsigned int gpmi_ecccount;
  unsigned int gpmi_data_ptr;
  unsigned int gpmi_aux_ptr;
} GENERIC_DESCRIPTOR;
//-----------------------------------------------------------------------------
// allocate 7 descriptors for doing a NAND ECC Read
//-----------------------------------------------------------------------------
GENERIC_DESCRIPTOR read[7];
//-----------------------------------------------------------------------------
// DMA descriptor pointer to handle error conditions from psense checks
//-----------------------------------------------------------------------------
unsigned int * dma_error_handler;
//-----------------------------------------------------------------------------
// 7 byte NAND command and address buffer
// any alignment is ok, it is read by the GPMI DMA
//   byte 0 is read setup command
//   bytes 1-5 is the NAND address
//   byte 6 is read execute command
//-----------------------------------------------------------------------------
unsigned char nand_cmd_addr_buffer[7];
//-----------------------------------------------------------------------------
// 4096 byte payload buffer used for reads or writes
// needs to be word aligned
//-----------------------------------------------------------------------------
unsigned int read_payload_buffer[(4096/4)];
//-----------------------------------------------------------------------------
// 412 byte auxiliary buffer used for reads
// needs to be word aligned
//-----------------------------------------------------------------------------
unsigned int read_aux_buffer[(412/4)];
//-----------------------------------------------------------------------------
// Descriptor 1: issue NAND read setup command (CLE/ALE)
//-----------------------------------------------------------------------------
read[0].dma_nxtcmdar = &read[1];                              // point to the next descriptor
read[0].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1 + 5) |  // 1 byte command, 5 byte address
                  BF_APBH_CHn_CMD_CMDWORDS      (3)     |  // send 3 words to the GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)     |  // wait for command to finish
                                                           //     before continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)     |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)     |
                  BF_APBH_CHn_CMD_NANDLOCK      (1)     |  // prevent other DMA channels from
                                                           //    taking over
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)     |
                  BF_APBH_CHn_CMD_CHAIN         (1)     |   // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ);  // read data from DMA, write to
```

```
NAND
read[0].dma_bar = &nand_cmd_addr_buffer;            // byte 0 read setup, bytes 1 - 5 NAND
address
// 3 words sent to the GPMI
read[0].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)     | // write to the NAND
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)     |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)   |
                     BF_GPMI_CTRL0_CS                 (2)        | // must correspond to NAND
CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE)  |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (1)        | // send command and address
                     BF_GPMI_CTRL0_XFER_COUNT         (1 + 5);    // 1 byte command, 5 byte
address
read[0].gpmi_compare = NULL;                                     // field not used but necessary to set
eccctrl
read[0].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE); // disable the ECC block
//-------------------------------------------------------------------------
// Descriptor 2: issue NAND read execute command (CLE)
//-------------------------------------------------------------------------
read[1].dma_nxtcmdar = &read[2];                                 // point to the next descriptor
read[1].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1)    |   // 1 byte read command
                  BF_APBH_CHn_CMD_CMDWORDS      (1)    |   // send 1 word to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)    |   // wait for command to finish
before
                                                         //    continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)    |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)    |
                  BF_APBH_CHn_CMD_NANDLOCK      (1)    |   // prevent other DMA channels from
                                                         //     taking over
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)    |
                  BF_APBH_CHn_CMD_CHAIN         (1)    |   // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ);     // read data from DMA, write
to NAND
read[1].dma_bar = &nand_cmd_addr_buffer[6];              // point to byte 6, read execute
command
// 1 word sent to the GPMI
read[1].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)     | // write to the NAND
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)     |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)  |
                     BF_GPMI_CTRL0_CS                 (2)        | // must correspond to NAND
CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE)  |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)        |
                     BF_GPMI_CTRL0_XFER_COUNT         (1);        // 1 byte command
//-------------------------------------------------------------------------
// Descriptor 3: wait for ready (DATA)
//-------------------------------------------------------------------------
read[2].dma_nxtcmdar = &read[3];                                 // point to the next descriptor
read[2].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)      | // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (1)      | // send 1 word to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)      | // wait for command to finish
before
                                                         //     continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)      |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(1)      | // wait for nand to be ready
                  BF_APBH_CHn_CMD_NANDLOCK      (0)      | // relinquish nand lock
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)      |
                  BF_APBH_CHn_CMD_CHAIN         (1)      | // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);   // no dma transfer
read[2].dma_bar = NULL;                                          // field not used
// 1 word sent to the GPMI
read[2].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WAIT_FOR_READY) | // wait for NAND
ready
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)          |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)       |
                     BF_GPMI_CTRL0_CS                 (2)             | // must correspond
to NAND CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA)      |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)             |
                     BF_GPMI_CTRL0_XFER_COUNT         (0);
```

```
//------------------------------------------------------------------------
// Descriptor 4: psense compare (time out check)
//------------------------------------------------------------------------
read[3].dma_nxtcmdar = &read[4];                                    // point to the next
descriptor
read[3].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)         |       // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (0)         |       // no words sent to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (0)         |       // do not wait to continue
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)         |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)         |
                  BF_APBH_CHn_CMD_NANDLOCK      (0)         |
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)         |
                  BF_APBH_CHn_CMD_CHAIN         (1)         |       // follow chain to next
command
                  BV_FLD(APBH_CHn_CMD, COMMAND, DMA_SENSE);         // perform a sense check
read[3].dma_bar = dma_error_handler;                        // if sense check fails, branch to
error handler
//------------------------------------------------------------------------
// Descriptor 5: read 4K page plus 65 byte meta-data Nand data
//               and send it to ECC block (DATA)
//------------------------------------------------------------------------
read[4].dma_nxtcmdar = &read[5];                              // point to the next descriptor
read[4].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)     | // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (6)     | // send 6 words to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)     | // wait for command to finish before
                                                          //       continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)     |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)     |
                  BF_APBH_CHn_CMD_NANDLOCK      (1)     | // prevent other DMA channels from
taking over
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)     | // ECC block generates BCH interrupt
                                                          //      on completion
                  BF_APBH_CHn_CMD_CHAIN         (1)     | // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);     // no DMA transfer,
                                                                  // ECC block handles
transfer
read[4].dma_bar = NULL;                                           // field not used
// 6 words sent to the GPMI
read[4].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, READ)       |  // read from the NAND
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)      |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)   |
                     BF_GPMI_CTRL0_CS                 (2)         |  // must correspond to
NAND CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA)  |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)         |
                     BF_GPMI_CTRL0_XFER_COUNT         (4096+218);    // eight 512 byte data
blocks
                                                                     // metadata, and parity

read[4].gpmi_compare = NULL;                              // field not used but necessary to set
eccctrl
// GPMI ECCCTRL PIO This launches the 4K byte transfer through BCH's
// bus master. Setting the ECC_ENABLE bit redirects the data flow
// within the GPMI so that read data flows to the BCH engine instead
// of flowing to the GPMI's DMA channel.
read[4].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ECC_CMD,    DECODE_8_BIT) |    // specify t = 8
mode
                       BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, ENABLE)       |    // enable ECC
module
                       BF_GPMI_ECCCTRL_BUFFER_MASK     (0X1FF);  // read all 8 data blocks
and 1 aux block
read[4].gpmi_ecccount = BF_GPMI_ECCCOUNT_COUNT(4096+218);         // specify number of bytes
                                                                  //        read from NAND
read[4].gpmi_data_ptr = &read_payload_buffer;                     // pointer for the 4K byte
                                                                  //    data area
read[4].gpmi_aux_ptr = &read_aux_buffer;                          // pointer for the 65 byte
aux area +
                                                                  // parity and syndrome
bytes for both
                                                                 // data and aux blocks.
```

```
//------------------------------------------------------------------------
// Descriptor 6: disable ECC block
//------------------------------------------------------------------------
read[5].dma_nxtcmdar = &read[6];                                  // point to the next descriptor
read[5].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)       | // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (3)       | // send 3 words to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)       | // wait for command to finish
before
                                                            //     continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)       |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(1)       | // wait for nand to be ready
                  BF_APBH_CHn_CMD_NANDLOCK      (1)       | // need nand lock to be
                                                            // thread safe while turn-off BCH
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)       |
                  BF_APBH_CHn_CMD_CHAIN         (1)       | // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);      // no dma transfer
read[5].dma_bar = NULL;                                           // field not used
// 3 words sent to the GPMI
read[5].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, READ)      |
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)     |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)  |
                     BF_GPMI_CTRL0_CS                 (2)        | // must correspond to
NAND CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA) |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)        |
                     BF_GPMI_CTRL0_XFER_COUNT         (0);
read[5].gpmi_compare = NULL;                                      // field not used but necessary to set
eccctrl
read[5].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE);  // disable the ECC block
//------------------------------------------------------------------------
// Descriptor 7: deassert nand lock
//------------------------------------------------------------------------
read[6].dma_nxtcmdar = NULL;                                      // not used since this is last
descriptor
read[6].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)   |   // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (0)   |   // no words sent to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (0)   |   // wait for command to finish
before
                                                            //        continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)   |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)   |
                  BF_APBH_CHn_CMD_NANDLOCK      (0)   |   // relinquish nand lock
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)   |   // BCH engine generates interrupt
                  BF_APBH_CHn_CMD_CHAIN         (0)   |   // terminate DMA chain processing
                  BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);      // no dma transfer
read[6].dma_bar = NULL;                                           // field not used
```

9.6.2.3.2.2 Using the Decoder

As illustrated in Figure 9-18 and the sample code in DMA Structure Code Example :

- DMA descriptor 1 prepares the NAND for data read by using the GPMI to issue a NAND read setup command byte under CLE, then sends a 5-byte address under ALE. The BCH engine is not used for these commands.
- DMA descriptor 2 issues a one-byte read execute command to the NAND device that triggers its read access. The NAND then goes not ready.
- DMA descriptor 3 performs a wait for ready operation allowing the DMA chain to remain dormant until the NAND device completes its read access time.

- DMA descriptor 5 handles the reading and error correction of the NAND data. This command's PIOs activate the BCH engine to write the read NAND data to system memory and to process it for any errors that need to be corrected. This DMA descriptor contains two PIO values that are system memory addresses pointing to the PAYLOAD data area and to the AUXILIARY data area. These addresses are used by the BCH engine's AHB master to move data into system memory and to correct it. While this example is reading an entire 4K page—payload plus metadata—it is equally possible to read just one 512-byte payload block or just the uniquely protected metadata block in a single 7 DMA structure transfer.
- DMA descriptor 6 disables the BCH engine with the NANDLOCK asserted. This is necessary to ensure that the GPMI resource is not arbitrated to another DMA channel when multiple DMA channels are active concurrently.
- DMA descriptor 7 de-asserts the NANDLOCK to free up the GPMI resource to another channel.

As the BCH block receives data from the GPMI:

- The decoder transforms the read NAND data block into a BCH code word and computes the codeword syndrome.
- If no errors are present, then the BCH block can immediately report back to firmware. This report is passed as the BCH_CTRL_COMPLETE_IRQ interrupt status bit and the associated status registers in BCH_STATUS0/1 registers.
- If an error is present, then the BCH block corrects the necessary data block or parity block bytes, if possible (not all errors are correctable).

As the BCH decoder reads the data and parity blocks, it records a special condition, i.e., that all of the bits of a payload data block or metadata block are one, including any associated parity bytes. The all-ones case for both parity and data indicates an erased block in the NAND device.

The BCH_STATUS0 register contains a 4-bit field that indicates the final status of the auxiliary block. A value of 0x0 indicates no errors found for a block.

- A value of 1 to 20 inclusive indicates that many correctable errors were found and fixed.
- A value of 0xFE indicates uncorrectable errors detected on the block.

- A value of 0xFF indicates that the block was in the special ALL ONES state and is therefore considered to be an ERASED block.
- All other values are disallowed by the hardware design.

Recall that up to eight NAND devices can have DMA chains in-flight at once, i.e. they can all be contending for access to the GPMI data bus. It is impossible to predict which NAND device will enter the BCH engine with a transfer first, because each chain includes a wait4ready command structure. As a result, firmware should look at the BCH_STATUS0_COMPLETED_CE bit field to determine which block is being reported in the status register. There is also a 16-bit HANDLE field in the GPMI_ECCCTRL register that is passed down the pipeline with each transaction. This handle field can be used to speed firmware's detection of which transaction is being reported.

These examples of reading and writing have focused on full page transfers of 4K page NAND devices. Other device configurations can be specified by changing the ECCOUNT field in the GPMI registers and reprogramming the BCH's FLASHnLAYOUTm registers.

The BCH and GPMI blocks are designed to be very efficient at reading single 512 (or 1024)-byte pages in one transaction. With no errors, the transaction takes less than 20 HCLKs longer than the time to read the raw data from the NAND.

To summarize, the APBH DMA command chain for a BCH decode operation is shown in Figure 9-18. Seven DMA command structures must be present for each NAND read transaction decoded by the BCH. The seven DMA command structures for multiple NAND read transaction blocks can be chained together to make larger units of work for the BCH, and each will produce an appropriate error report in the BCH PIO space. Multiple NAND devices can have such multiple chains scheduled. The results can come back out of order with respect to the multiple chains.

9.6.2.3.3 Interrupts

There are two interrupt sources used in processing BCH protected NAND read and write transfers.

Since all BCH operations are initiated by GPMI DMA command structures, the DMA completion interrupt for the GPMI is an important ISR. Both of the flow charts of Figure 9-14 and Figure 9-17 show the GPMI DMA complete ISR skeleton. In both reads and writes, the GPMI DMA completion interrupt is used to schedule work ***INTO*** the error correction pipeline. As the front end processing completes, the DMA interrupt is generated and additional work, such as DMA chains, are passed to the GPMI DMA to keep it *fed*. For write operations, this is the only interrupt that is generated for processing the NAND write transfer.

For reads, however, two interrupts are needed. Every read is started by a GPMI DMA command chain and the front end queue is fed as described above. The back end of the read pipeline is drained by monitoring the BCH completion interrupt found int HW_BCH_CTRL_COMPLETE_IRQ.

BCH transaction consists of reading or writing all of the blocks requested in the HW_GPMI_ECCCTRL_BUFFER_MASK bit field. As every read transaction completes, it posts the status of all of the blocks to the HW_BCH_STATUS0 and HW_BCH_STATUS1 registers and sets the completion interrupt. The five stages of the BCH read pipeline completes, one in the GPMI and four in the BCH, are independently stalled as they complete and try to deliver to the next stage in the data flow. Several of these stages can be skipped if no-errors are found or once an uncorrectable error is found in a block.

In any case, the final stage will stall if the status register is busy waiting for the CPU to take status register results. The hardware monitors the state of the HW_BCH_CTRL_COMPLETE_IRQ bit. If it is still set when the last pipeline stage is ready to post data, then the stage will stall. It follows that the next previous stage will stall when it is ready to hand off work to the final stage, and so on up the pipeline.

CAUTION

It is important that firmware read the STATUS0/1 results and save them before clearing the interrupt request bit. Otherwise, a transaction and its results could be completely lost.

9.6.2.3.4 Randomizer

BCH ECC has a Randomizer module that is interfaced through the GPMI APBHDMA chain. The Randomizer can generate random data based on BCH ECC encoded/decoded data. It can be employed to reduce the disturbances caused by a neighboring cell in the NAND chip, thus reducing bit errors.

To enable the Randomizer module, set GPMI_ECCCTRL[RANDOMIZER_ENABLE] to 1, then set GPMI_ECCCOUNT[RANDOMIZER_PAGE] to select randomizer page number needed to be randomized. All these registers can be programmed by the DMA chain. The randomized data should start from the zero column address and be the size of the whole NAND page. If the randomizer function is enabled, GPMI_ECCCTRL[ENABLE_ECC] should also be enabled. To bypass BCH error correction function, set BCH_FLASHxLAYOUT0[ECC0] and BCH_FLASHxLAYOUT1[ECCN] to 0.

9.6.2.4 Behavior During Reset

A soft reset (SFTRST) can take multiple clock periods to complete, so do NOT set CLKGATE when setting SFTRST.

The reset process gates the clocks automatically. The exemplary code is shown below.

```
// A soft reset can take multiple clocks to complete, so do NOT gate the
// clock when setting soft reset. The reset process will gate the clock
// automatically. Poll until this has happened before subsequently
// preparing soft-reset and clock gate
  BCH_CTRL_CLR(BM_BCH_CTRL_SFTRST);
  BCH_CTRL_CLR(BM_BCH_CTRL_CLKGATE);
// asserting soft-reset
  BCH_CTRL_SET(BM_BCH_CTRL_SFTRST);
// waiting for confirmation of soft-reset
  while (!BCH_CTRL.B.CLKGATE)
  {
// busy wait
  }
// Done.
  BCH_CTRL_CLR(BM_BCH_CTRL_SFTRST);
  BCH_CTRL_CLR(BM_BCH_CTRL_CLKGATE);
```

9.6.3 BCH register descriptions

BCH Hardware Register Format Summary

9.6.3.1 BCH memory map

BCH base address: 3300_4000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Hardware BCH ECC Accelerator Control Register (CTRL)	32	RW	C000_0000h
4h	Hardware BCH ECC Accelerator Control Register (CTRL_SET)	32	RW	C000_0000h
8h	Hardware BCH ECC Accelerator Control Register (CTRL_CLR)	32	RW	C000_0000h
Ch	Hardware BCH ECC Accelerator Control Register (CTRL_TOG)	32	RW	C000_0000h
10h	Hardware ECC Accelerator Status Register 0 (STATUS0)	32	RO	0000_0010h
20h	Hardware ECC Accelerator Mode Register (MODE)	32	RW	0000_0000h
30h	Hardware BCH ECC Loopback Encode Buffer Register (ENCODEPTR)	32	RW	0000_0000h
40h	Hardware BCH ECC Loopback Data Buffer Register (DATAPTR)	32	RW	0000_0000h
50h	Hardware BCH ECC Loopback Metadata Buffer Register (METAPTR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
70h	Hardware ECC Accelerator Layout Select Register (LAYOUTSELECT)	32	RW	E4E4_E4E4h
80h	Hardware BCH ECC Flash 0 Layout 0 Register (FLASH0LAYOUT0)	32	RW	070A_4080h
90h	Hardware BCH ECC Flash 0 Layout 1 Register (FLASH0LAYOUT1)	32	RW	10DA_4080h
A0h	Hardware BCH ECC Flash 1 Layout 0 Register (FLASH1LAYOUT0)	32	RW	070A_4080h
B0h	Hardware BCH ECC Flash 1 Layout 1 Register (FLASH1LAYOUT1)	32	RW	10DA_4080h
C0h	Hardware BCH ECC Flash 2 Layout 0 Register (FLASH2LAYOUT0)	32	RW	070A_4080h
D0h	Hardware BCH ECC Flash 2 Layout 1 Register (FLASH2LAYOUT1)	32	RW	10DA_4080h
E0h	Hardware BCH ECC Flash 3 Layout 0 Register (FLASH3LAYOUT0)	32	RW	070A_4080h
F0h	Hardware BCH ECC Flash 3 Layout 1 Register (FLASH3LAYOUT1)	32	RW	10DA_4080h
100h	Hardware BCH ECC Debug Register0 (DEBUG0)	32	RW	0000_0000h
104h	Hardware BCH ECC Debug Register0 (DEBUG0_SET)	32	RW	0000_0000h
108h	Hardware BCH ECC Debug Register0 (DEBUG0_CLR)	32	RW	0000_0000h
10Ch	Hardware BCH ECC Debug Register0 (DEBUG0_TOG)	32	RW	0000_0000h
110h	KES Debug Read Register (DBGKESREAD)	32	RO	0000_0000h
120h	Chien Search Debug Read Register (DBGCSFEREAD)	32	RO	0000_0000h
130h	Syndrome Generator Debug Read Register (DBGSYNDGENREAD)	32	RO	0000_0000h
140h	Bus Master and ECC Controller Debug Read Register (DBGAHBMREAD)	32	RO	0000_0000h
150h	Block Name Register (BLOCKNAME)	32	RO	2048_4342h
160h	BCH Version Register (VERSION)	32	RO	0100_0000h
170h	Hardware BCH ECC Debug Register 1 (DEBUG1)	32	RW	0000_0000h
174h	Hardware BCH ECC Debug Register 1 (DEBUG1_SET)	32	RW	0000_0000h
178h	Hardware BCH ECC Debug Register 1 (DEBUG1_CLR)	32	RW	0000_0000h
17Ch	Hardware BCH ECC Debug Register 1 (DEBUG1_TOG)	32	RW	0000_0000h

9.6.3.2 Hardware BCH ECC Accelerator Control Register (CTRL)

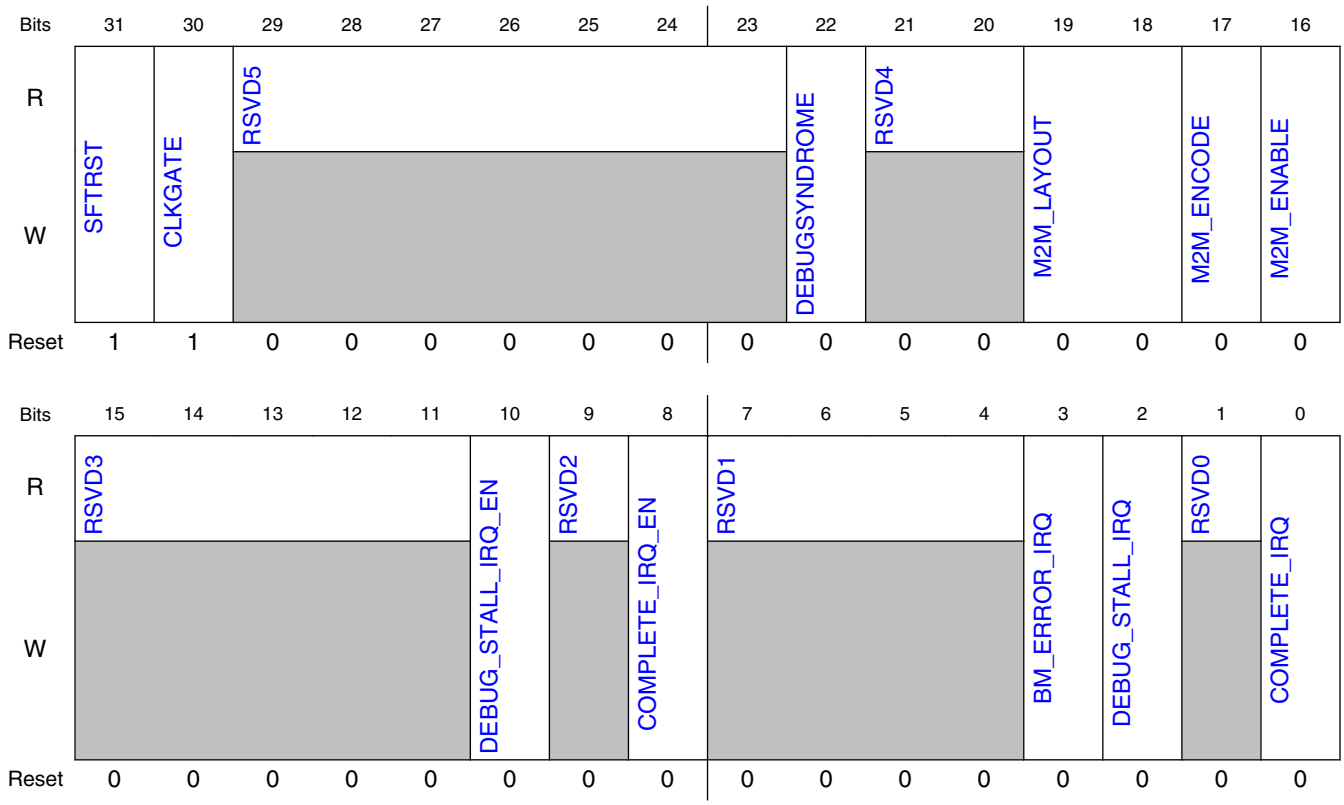
9.6.3.2.1 Offset

Register	Offset	Description
CTRL	0h	Hardware BCH ECC Accelerator Control Register
CTRL_SET	4h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL
CTRL_CLR	8h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL
CTRL_TOG	Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL

9.6.3.2.2 Function

The BCH CTRL provides overall control of the hardware ECC accelerator

9.6.3.2.3 Diagram



9.6.3.2.4 Fields

Field	Function
31 SFTRST	SFTRST Set this bit to 0 to enable normal BCH operation. Set this bit to 1 (default) to disable clocking with the BCH and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset the BCH block to its default state. This bit resets all state machines except for the AHB master state machine 0b - Allow BCH to operate normally. 1b - Hold BCH in reset.
30 CLKGATE	CLKGATE This bit must be set to 0 for normal operation. When set to 1 it gates off the clocks to the block. 0b - Allow BCH to operate normally. 1b - Do not clock BCH gates in order to minimize power consumption.
29-23	This field is reserved.

Table continues on the next page...

Field	Function
RSVD5	This read-only field is reserved and always has the value 0.
22 DEBUGSYNDROME	DEBUGSYNDROME (For debug purposes only). Enable write of computed syndromes to memory on BCH decode operations. Computed syndromes will be written to the auxiliary buffer after the status block. Syndromes will be written as padded 16-bit values.
21-20 RSVD4	This field is reserved. This read-only field is reserved and always has the value 0
19-18 M2M_LAYOUT	M2M_LAYOUT Selects the flash page format for memory-to-memory operations.
17 M2M_ENCODE	M2M_ENCODE Selects encode (parity generation) or decode (correction) mode for memory-to-memory operations.
16 M2M_ENABLE	M2M_ENABLE NOTE! WRITING THIS BIT INITIATES A MEMORY-TO-MEMORY OPERATION. The BCH module must be inactive (not processing data from the GPMI) when this bit is set. The M2M_ENCODE and M2M_LAYOUT bits as well as the ENCODEPTR, DATAPTR, and METAPTR registers are used for memory-to-memory operations and must be correctly programmed before writing this bit.
15-11 RSVD3	This field is reserved. This read-only field is reserved and always has the value 0
10 DEBUG_STALL_IRQ_EN	DEBUG_STALL_IRQ_EN 1 = interrupt on debug stall mode is enabled. The IRQ is raised on every block
9 RSVD2	This field is reserved. This read-only field is reserved and always has the value 0.
8 COMPLETE_IRQ_EN	COMPLETE_IRQ_EN 1 = interrupt on completion of correction is enabled.
7-4 RSVD1	This field is reserved. This read-only field is reserved and always has the value 0.
3 BM_ERROR_IRQ	BM_ERROR_IRQ AHB Bus interface Error Interrupt Status. Write a 1 to the SCT clear address to clear the interrupt status bit.
2 DEBUG_STALL_IRQ	DEBUG_STALL_IRQ DEBUG STALL Interrupt Status. Write a 1 to the SCT clear address to clear the interrupt status bit.
1 RSVD0	This field is reserved. This read-only field is reserved and always has the value 0.
0 COMPLETE_IRQ	COMPLETE_IRQ This bit indicates the state of the external interrupt line. Write a 1 to the SCT clear address to clear the interrupt status bit. NOTE: subsequent ECC completions will be held off as long as this bit is set. Be sure to read the data from BCH_STATUS0, 1 before clearing this interrupt bit.

9.6.3.3 Hardware ECC Accelerator Status Register 0 (STATUS0)

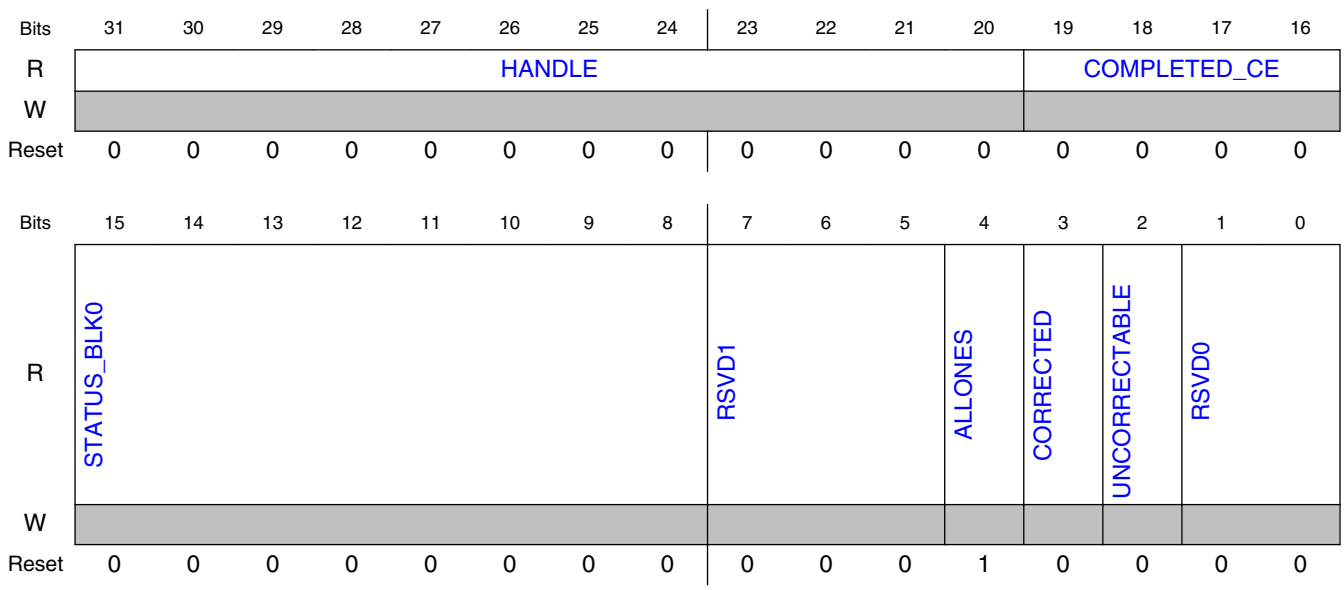
9.6.3.3.1 Offset

Register	Offset
STATUS0	10h

9.6.3.3.2 Function

The BCH STAT register provides visibility into the run-time status of the BCH and status information when processing is complete. It provides overall status of the hardware ECC accelerator.

9.6.3.3.3 Diagram



9.6.3.3.4 Fields

Field	Function
31-20 HANDLE	HANDLE Software supplies a 12 bit handle for this transfer as part of the GPMI DMA PIO operation that started the transaction. That handle passes down the pipeline and ends up here at the time the BCH interrupt is signaled.
19-16	COMPLETED_CE

Table continues on the next page...

Field	Function
COMPLETED_CE	This is the chip enable number corresponding to the NAND device from which this data came.
15-8 STATUS_BLK0	STATUS_BLK0 Count of symbols in error during processing of first block of flash (metadata block). The number of errors reported will be in the range of 0 to the ECC correction level for block 0. 00000000b - No errors found on block. 00000001b - One error found on block. 00000010b - One errors found on block. 00000011b - One errors found on block. 00000100b - One errors found on block. 11111110b - Block exhibited uncorrectable errors. 11111111b - Page is erased.
7-5 RSVD1	This field is reserved. This read-only field is reserved and always has the value 0.
4 ALLONES	ALLONES 1 = All data bits of this transaction are ONE.
3 CORRECTED	CORRECTED 1 = At least one correctable error encountered during last processing cycle.
2 UNCORRECTABLE	UNCORRECTABLE 1 = Uncorrectable error encountered during last processing cycle.
1-0 RSVD0	This field is reserved. This read-only field is reserved and always has the value 0.

9.6.3.4 Hardware ECC Accelerator Mode Register (MODE)

9.6.3.4.1 Offset

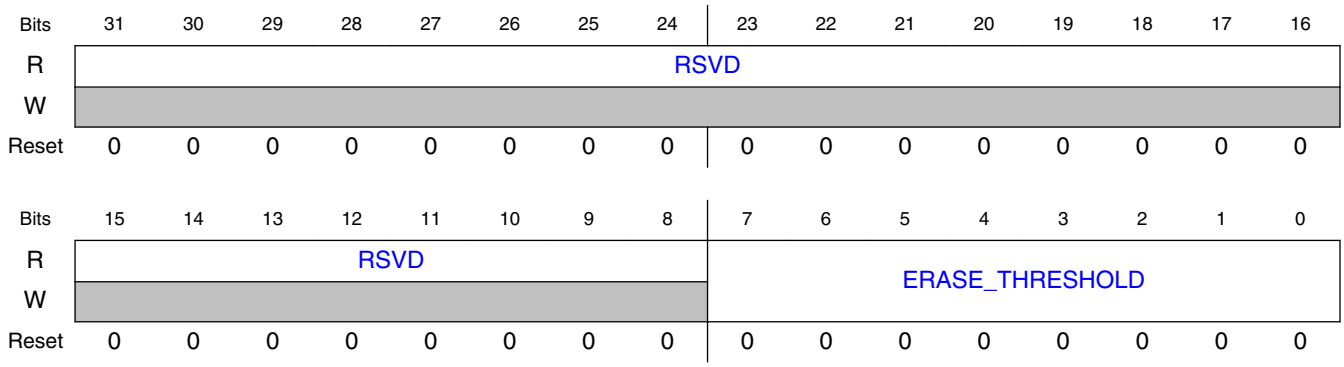
Register	Offset
MODE	20h

9.6.3.4.2 Function

The BCH MODE register provides additional mode controls.

Contains additional global mode controls for the BCH engine.

9.6.3.4.3 Diagram



9.6.3.4.4 Fields

Field	Function
31-8 RSVD	This field is reserved. This read-only field is reserved and always has the value 0.
7-0 ERASE_THRESHOLD	ERASE_THRESHOLD This value indicates the maximum number of zero bits on a flash subpage for it to be considered erased. For SLC NAND devices, this value should be programmed to 0 (meaning that the entire page should consist of bytes of 0xFF. For MLC NAND devices, bit errors may occur on reads (even on blank pages), so this threshold can be used to tune the erased page checking algorithm.

9.6.3.5 Hardware BCH ECC Loopback Encode Buffer Register (ENCODEPTR)

9.6.3.5.1 Offset

Register	Offset
ENCODEPTR	30h

9.6.3.5.2 Function

When performing memory to memory operations, indicates the address of the encode buffer. This register should be programmed before writing a 1 to the M2M_ENABLE bit in the CTRL register.

For memory to memory operations, this register is used as the pointer to the encoded data, which is an output when encoding and an input while decoding.

9.6.3.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.6.3.5.4 Fields

Field	Function
31-0 ADDR	ADDR Address pointer to encode buffer. This is the source for decode operations and the destination for encode operations. This value must be aligned on a 4 bytes boundary.

9.6.3.6 Hardware BCH ECC Loopback Data Buffer Register (DATAPTR)

9.6.3.6.1 Offset

Register	Offset
DATAPTR	40h

9.6.3.6.2 Function

When performing memory to memory operations, indicates the address of the data buffer.

For memory to memory operations, this register is used as the pointer to the data to encode or the destination buffer for decode operations.

9.6.3.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.6.3.6.4 Fields

Field	Function
31-0 ADDR	ADDR Address pointer to data buffer. This is the source for encode operations and the destination for decode operations. This register should be programmed before writing a 1 to the M2M_ENABLE bit in the CTRL register. This value must be aligned on a 4 byte boundary.

9.6.3.7 Hardware BCH ECC Loopback Metadata Buffer Register (METAPTR)

9.6.3.7.1 Offset

Register	Offset
METAPTR	50h

9.6.3.7.2 Function

When performing memory to memory operations, indicates the address of the metadata buffer.

For memory to memory operations, this register is used as the pointer to the metadata to encode or the extracted metadata for decode operations.

9.6.3.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.6.3.7.4 Fields

Field	Function
31-0 ADDR	ADDR Address pointer to metadata buffer. This is the source for encode metadata read operations and the destination for metadata decode operations. This register should be programmed before writing a 1 to the M2M_ENABLE bit in the CTRL register. This value must be aligned on a 4 bytes boundary.

9.6.3.8 Hardware ECC Accelerator Layout Select Register (LAYOUTSELECT)

9.6.3.8.1 Offset

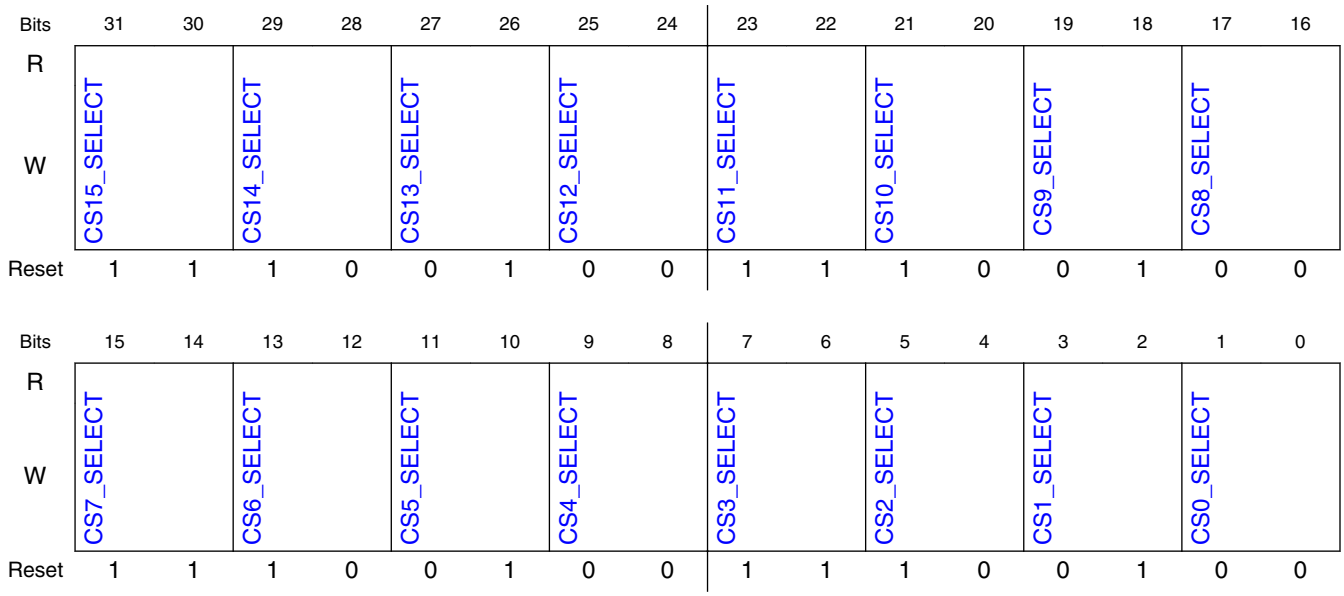
Register	Offset
LAYOUTSELECT	70h

9.6.3.8.2 Function

The BCH LAYOUTSELECT register provides a mapping of chip selects to layout registers.

When the BCH engine receives a request to process a data block from the GPMI interface, it will use this register to map the incoming chip select to one of the four possible flash layout registers

9.6.3.8.3 Diagram



9.6.3.8.4 Fields

Field	Function
31-30 CS15_SELECT	CS15_SELECT Selects which layout is used for chip select 15.
29-28 CS14_SELECT	CS14_SELECT Selects which layout is used for chip select 14.
27-26 CS13_SELECT	CS13_SELECT Selects which layout is used for chip select 13.
25-24 CS12_SELECT	CS12_SELECT Selects which layout is used for chip select 12.
23-22 CS11_SELECT	CS11_SELECT Selects which layout is used for chip select 11.
21-20 CS10_SELECT	CS10_SELECT Selects which layout is used for chip select 10.
19-18 CS9_SELECT	CS9_SELECT Selects which layout is used for chip select 9.
17-16 CS8_SELECT	CS8_SELECT Selects which layout is used for chip select 8.
15-14 CS7_SELECT	CS7_SELECT Selects which layout is used for chip select 7.
13-12 CS6_SELECT	CS6_SELECT Selects which layout is used for chip select 6.

Table continues on the next page...

Field	Function
11-10 CS5_SELECT	CS5_SELECT Selects which layout is used for chip select 5.
9-8 CS4_SELECT	CS4_SELECT Selects which layout is used for chip select 4.
7-6 CS3_SELECT	CS3_SELECT Selects which layout is used for chip select 3.
5-4 CS2_SELECT	CS2_SELECT Selects which layout is used for chip select 2.
3-2 CS1_SELECT	CS1_SELECT Selects which layout is used for chip select 1.
1-0 CS0_SELECT	CS0_SELECT Selects which layout is used for chip select 0.

9.6.3.9 Hardware BCH ECC Flash 0 Layout 0 Register (FLASH0LAYOUT0)

9.6.3.9.1 Offset

Register	Offset
FLASH0LAYOUT0	80h

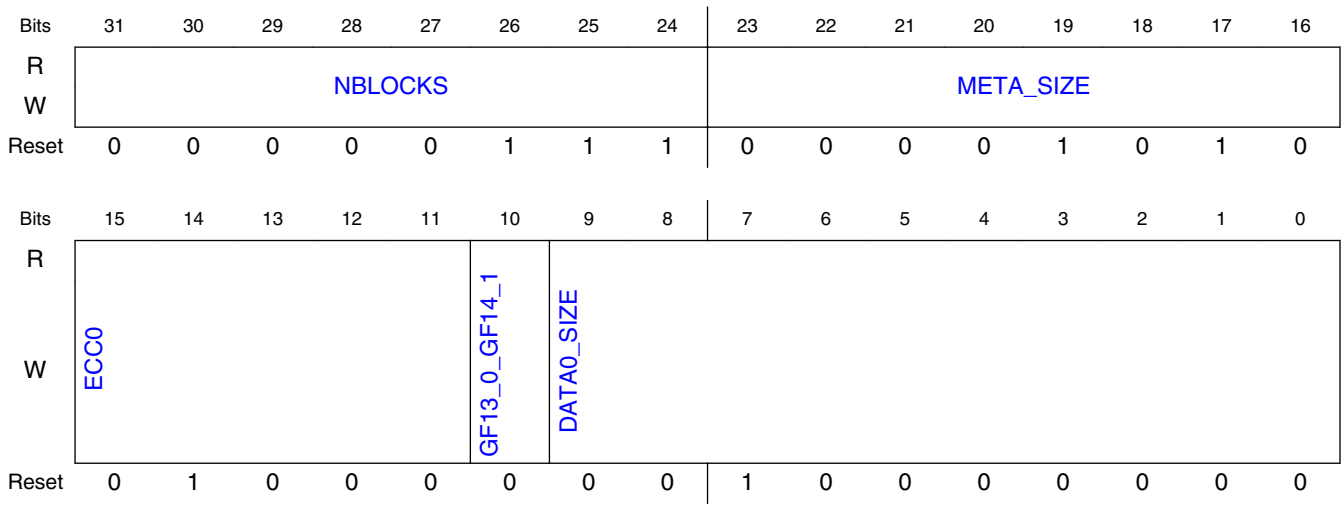
9.6.3.9.2 Function

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH0LAYOUT1 register to control the format for the devices selecting layout 0 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading / writing the flash page to control data, metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks.

See sections Flash Page Layout and Determining the ECC layout for a device for more detail information on setting up the flash layout registers.

9.6.3.9.3 Diagram



9.6.3.9.4 Fields

Field	Function
31-24 NBLOCKS	NBLOCKS Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that 8 subsequent blocks are present for a total of 9 blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.
23-16 META_SIZE	META_SIZE Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design support from 0 to 255 bytes for metadata—if set to 0, no metadata will be stored. Metadata is stored before the associated data in block 0. If the DATA0_SIZE field is programmed to a 0, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and covered by a single parity block.
15-11 ECC0	ECC0 Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field. 00000b - No ECC to be performed 00001b - ECC 2 to be performed 00010b - ECC 4 to be performed 11110b - ECC 60 to be performed 11111b - ECC 62 to be performed
10 GF13_0_GF14_1	GF13_0_GF14_1 Select GF13 or GF14: 0-GF13; 1-GF14

Table continues on the next page...

Field	Function
9-0 DATA0_SIZE	DATA0_SIZE Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to 0, the first block only contains metadata.

9.6.3.10 Hardware BCH ECC Flash 0 Layout 1 Register (FLASH0LAYOUT1)

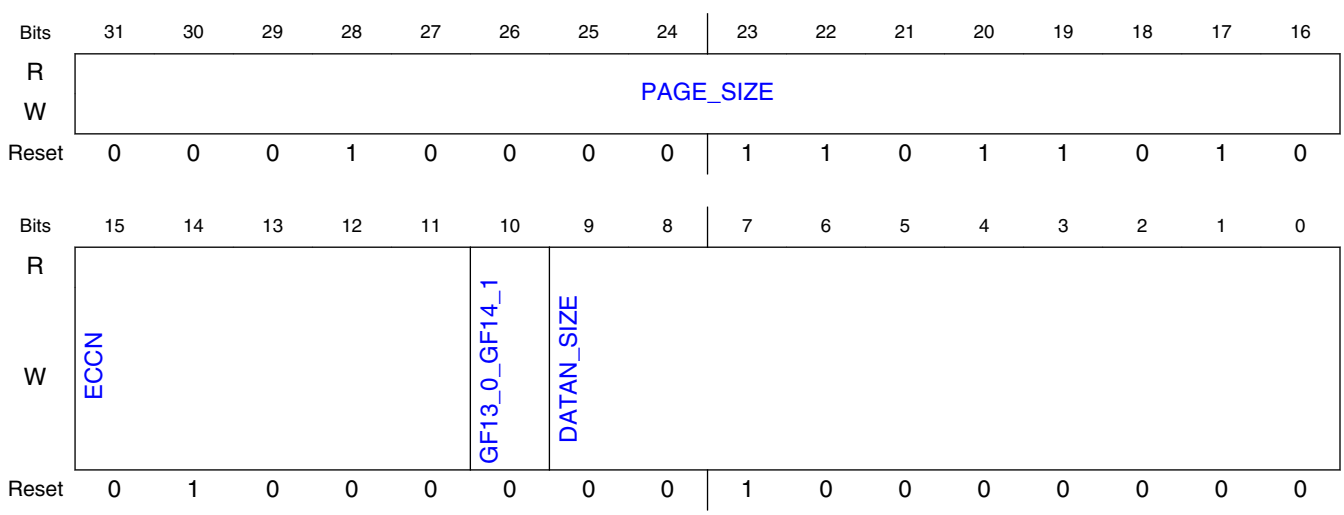
9.6.3.10.1 Offset

Register	Offset
FLASH0LAYOUT1	90h

9.6.3.10.2 Function

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH0LAYOUT0 register to control the format for the device selecting layout 0 in the LAYOUTSELECT register.

9.6.3.10.3 Diagram



9.6.3.10.4 Fields

Field	Function
31-16 PAGE_SIZE	PAGE_SIZE Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15-11 ECCN	ECCN Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). 00000b - No ECC to be performed 00001b - ECC 2 to be performed 00010b - ECC 4 to be performed 11110b - ECC 60 to be performed 11111b - ECC 62 to be performed
10 GF13_0_GF14_1	GF13_0_GF14_1 Select GF13 or GF14: 0-GF13; 1-GF14
9-0 DATAN_SIZE	DATAN_SIZE Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

9.6.3.11 Hardware BCH ECC Flash 1 Layout 0 Register (FLASH1LAYOUT0)

9.6.3.11.1 Offset

Register	Offset
FLASH1LAYOUT0	A0h

9.6.3.11.2 Function

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH1LAYOUT1 register to control the format for the devices selecting layout 1 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading / writing the flash page to control data,

metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks.

See sections Flash Page Layout and Determining the ECC layout for a device for more detail information on setting up the flash layout registers.

9.6.3.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NBLOCKS								META_SIZE							
Reset	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECC0					GF13_0_GF14_1	DATA0_SIZE									
Reset	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0

9.6.3.11.4 Fields

Field	Function
31-24 NBLOCKS	NBLOCKS Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that 8 subsequent blocks are present for a total of 9 blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.
23-16 META_SIZE	META_SIZE Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design supports from 0 to 255 bytes for metadata—if set to 0, no metadata will be stored. Metadata is stored before the associated data is in block 0. If the DATA0_SIZE field is programmed to a 0, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and covered by a single parity block.
15-11 ECC0	ECC0 Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field. 00000b - No ECC to be performed 00001b - ECC 2 to be performed

Table continues on the next page...

Field	Function
	00010b - ECC 4 to be performed 11110b - ECC 60 to be performed 11111b - ECC 62 to be performed
10 GF13_0_GF14_1	GF13_0_GF14_1 Select GF13 or GF14: 0-GF13; 1-GF14
9-0 DATA0_SIZE	DATA0_SIZE Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to 0, the first block will contains metadata.

9.6.3.12 Hardware BCH ECC Flash 1 Layout 1 Register (FLASH1LAYOUT1)

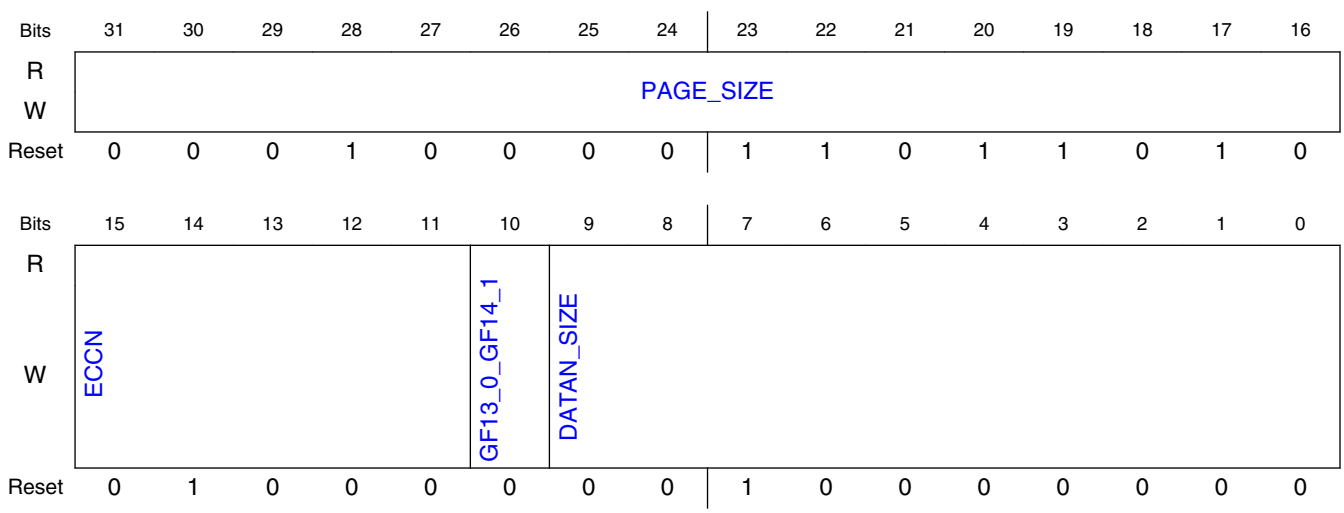
9.6.3.12.1 Offset

Register	Offset
FLASH1LAYOUT1	B0h

9.6.3.12.2 Function

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH1LAYOUT0 register to control the format for the device selecting layout 1 in the LAYOUTSELECT register.

9.6.3.12.3 Diagram



9.6.3.12.4 Fields

Field	Function
31-16 PAGE_SIZE	PAGE_SIZE Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15-11 ECCN	ECCN Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). 00000b - No ECC to be performed 00001b - ECC 2 to be performed 00010b - ECC 4 to be performed 11110b - ECC 60 to be performed 11111b - ECC 62 to be performed
10 GF13_0_GF14_1	GF13_0_GF14_1 Select GF13 or GF14: 0-GF13; 1-GF14
9-0 DATAN_SIZE	DATAN_SIZE Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

9.6.3.13 Hardware BCH ECC Flash 2 Layout 0 Register (FLASH2LAYOUT0)

9.6.3.13.1 Offset

Register	Offset
FLASH2LAYOUT0	C0h

9.6.3.13.2 Function

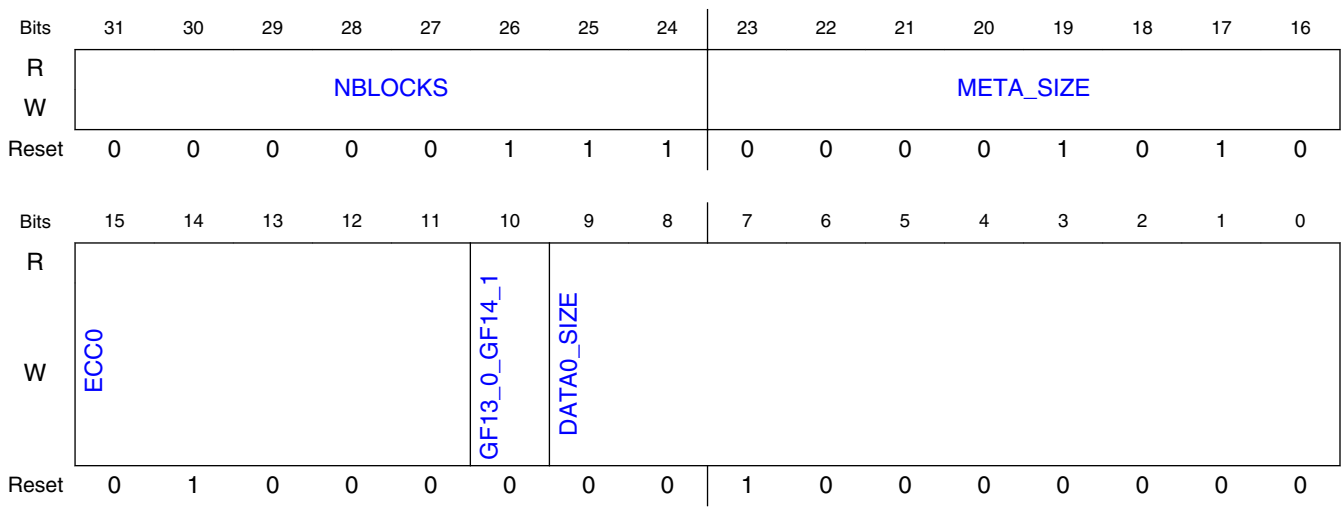
The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH2LAYOUT1 register to control the format for the devices selecting layout 2 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings

that are used by the BCH block while reading / writing the flash page to control data, metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks.

See sections Flash Page Layout and Determining the ECC layout for a device for more detail information on setting up the flash layout registers.

9.6.3.13.3 Diagram



9.6.3.13.4 Fields

Field	Function
31-24 NBLOCKS	NBLOCKS Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that eight subsequent blocks are present for a total of nine blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.
23-16 META_SIZE	META_SIZE Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design support from 0 to 255 bytes for metadata—if set to 0, no metadata will be stored. Metadata is stored before the associated data in block 0. If the DATA0_SIZE field is programmed to a 0, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and will be covered by a single parity block.
15-11 ECC0	ECC0 Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field.

Table continues on the next page...

Field	Function
	00000b - No ECC to be performed 00001b - ECC 2 to be performed 00010b - ECC 4 to be performed 11110b - ECC 60 to be performed 11111b - ECC 62 to be performed
10 GF13_0_GF14_1	GF13_0_GF14_1 Select GF13 or GF14: 0-GF13; 1-GF14
9-0 DATA0_SIZE	DATA0_SIZE Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to 0, the first block will only contain metadata.

9.6.3.14 Hardware BCH ECC Flash 2 Layout 1 Register (FLASH2LAYOUT1)

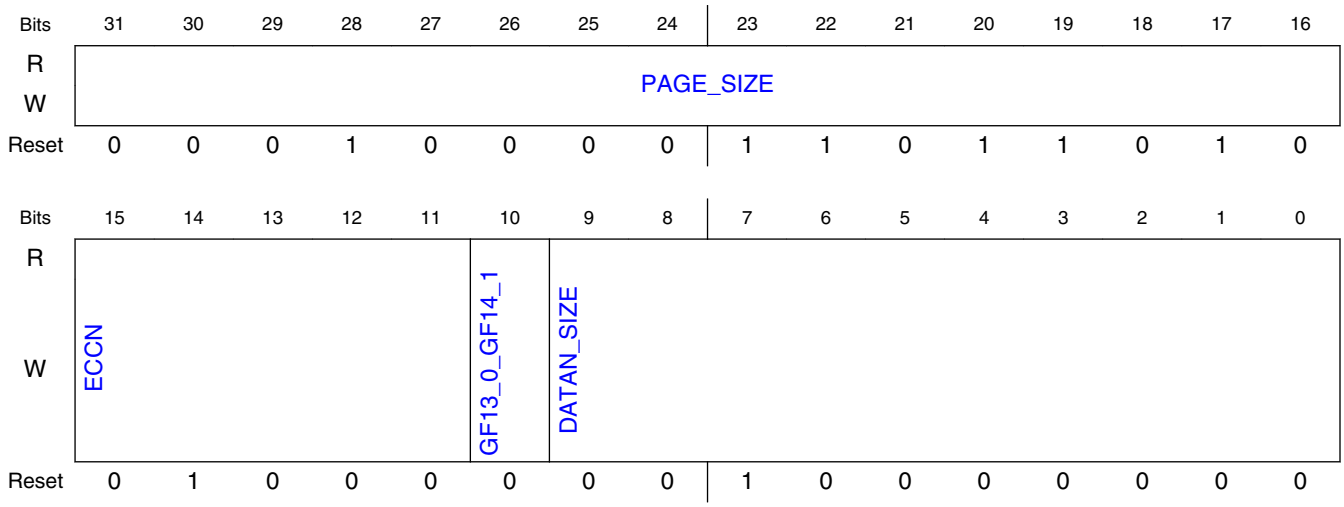
9.6.3.14.1 Offset

Register	Offset
FLASH2LAYOUT1	D0h

9.6.3.14.2 Function

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH2LAYOUT0 register to control the format for the device selecting layout 2 in the LAYOUTSELECT register.

9.6.3.14.3 Diagram



9.6.3.14.4 Fields

Field	Function
31-16 PAGE_SIZE	PAGE_SIZE Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15-11 ECCN	ECCN Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). 00000b - No ECC to be performed 00001b - ECC 2 to be performed 00010b - ECC 4 to be performed 11110b - ECC 60 to be performed 11111b - ECC 62 to be performed
10 GF13_0_GF14_1	GF13_0_GF14_1 Select GF13 or GF14: 0-GF13; 1-GF14
9-0 DATAN_SIZE	DATAN_SIZE Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

9.6.3.15 Hardware BCH ECC Flash 3 Layout 0 Register (FLASH3LAYOUT0)

9.6.3.15.1 Offset

Register	Offset
FLASH3LAYOUT0	E0h

9.6.3.15.2 Function

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH3LAYOUT1 register to control the format for the devices selecting layout 3 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading / writing the flash page to control data, metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks.

See sections Flash Page Layout and Determining the ECC layout for a device for more detail information on setting up the flash layout registers.

9.6.3.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NBLOCKS								META_SIZE							
Reset	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECC0					GF13_0_GF14_1	DATA0_SIZE									
Reset	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0

9.6.3.15.4 Fields

Field	Function
31-24 NBLOCKS	NBLOCKS Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that 8 subsequent blocks are present for a total of 9 blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.
23-16 META_SIZE	META_SIZE Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design support from 0 to 255 bytes for metadata—if set to 0, no metadata will be stored. Metadata is stored before the associated data in block 0. If the DATA0_SIZE field is programmed to a 0, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and will be covered by a single parity block.
15-11 ECC0	ECC0 Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field. 00000b - No ECC to be performed 00001b - ECC 2 to be performed 00010b - ECC 4 to be performed 11110b - ECC 60 to be performed 11111b - ECC 62 to be performed
10 GF13_0_GF14_1	GF13_0_GF14_1 Select GF13 or GF14: 0-GF13; 1-GF14
9-0 DATA0_SIZE	DATA0_SIZE Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to 0, the first block will only contain metadata.

9.6.3.16 Hardware BCH ECC Flash 3 Layout 1 Register (FLASH3LAYOUT1)

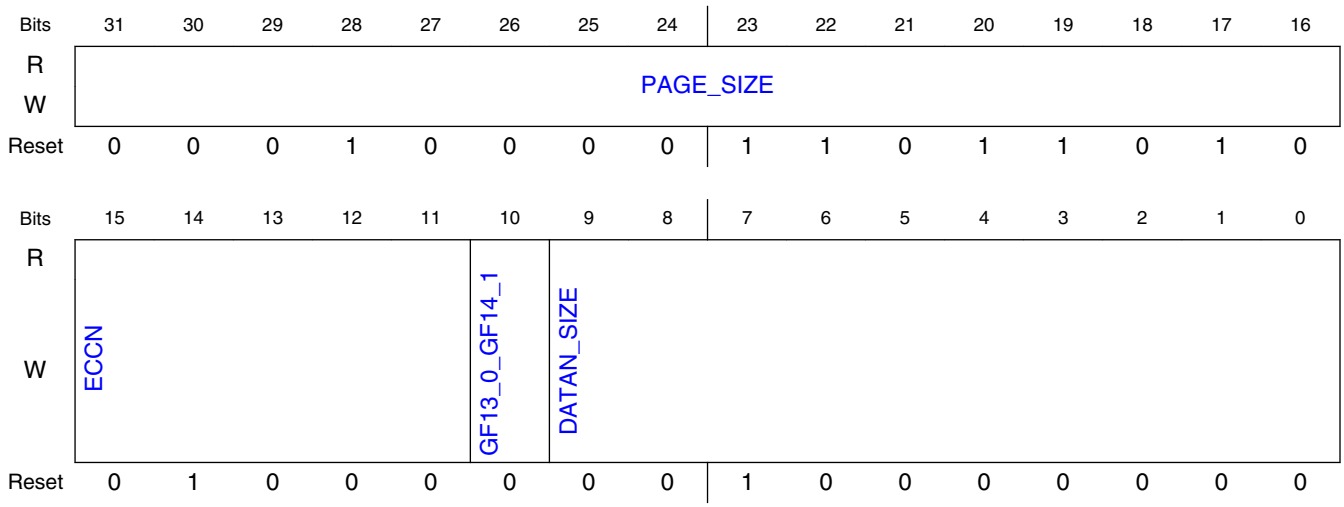
9.6.3.16.1 Offset

Register	Offset
FLASH3LAYOUT1	F0h

9.6.3.16.2 Function

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH3LAYOUT0 register to control the format for the device selecting layout 3 in the LAYOUTSELECT register.

9.6.3.16.3 Diagram



9.6.3.16.4 Fields

Field	Function
31-16 PAGE_SIZE	PAGE_SIZE Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15-11 ECCN	ECCN Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). 00000b - No ECC to be performed 00001b - ECC 2 to be performed 00010b - ECC 4 to be performed 11110b - ECC 60 to be performed 11111b - ECC 62 to be performed
10 GF13_0_GF14_1	GF13_0_GF14_1 Select GF13 or GF14: 0-GF13; 1-GF14
9-0 DATAN_SIZE	DATAN_SIZE Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

9.6.3.17 Hardware BCH ECC Debug Register0 (DEBUG0)

9.6.3.17.1 Offset

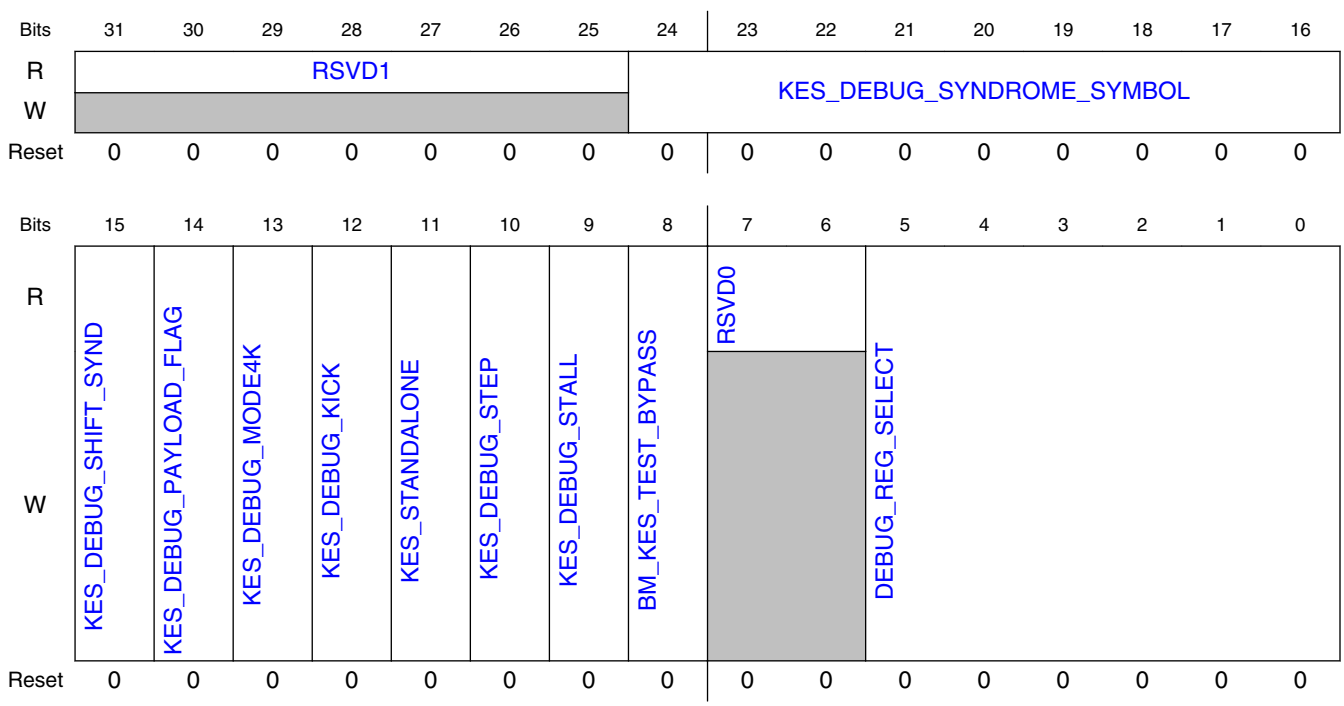
Register	Offset	Description
DEBUG0	100h	Hardware BCH ECC Debug Register0
DEBUG0_SET	104h	Writing a 1 to a bit in this register sets the corresponding bit in DEBUG0
DEBUG0_CLR	108h	Writing a 1 to a bit in this register clears the corresponding bit in DEBUG0
DEBUG0_TOG	10Ch	Writing a 1 to a bit in this register toggles the corresponding bit in DEBUG0

9.6.3.17.2 Function

The hardware BCH accelerator internal state machines and signals can be seen in the ECC debug register.

The BCH_DEBUG0 register provides access to various internal state information which might prove useful during hardware debug and validation.

9.6.3.17.3 Diagram



9.6.3.17.4 Fields

Field	Function
31-25 RSVD1	This field is reserved. This read-only field is reserved and always has the value 0.
24-16 KES_DEBUG_SYNDROME_SYMBOL	KES_DEBUG_SYNDROME_SYMBOL The 9 bit value in this bit field shifts into the syndrome register array at the input of the KES engine whenever BCH_DEBUG0_KES_DEBUG_SHIFT_SYND is toggled. 000000000b - Bus master address generator for SYND_GEN writes operates normally. 000000001b - Bus master address generator always addresses last four bytes in Auxiliary block.
15 KES_DEBUG_SHIFT_SYND	KES_DEBUG_SHIFT_SYND Toggling this bit causes the value in BCH_DEBUG0_KES_SYNDROME_SYMBOL to be shift into the syndrome register array at the input to the KES engine. After shifting in 16 symbols, one can kick off both KES and CF cycles by toggling BCH_DEBUG0_KES_DEBUG_KICK. Make sure that set KES_BCH_DEBUG0_KES_STANDALONE mode to 1 before kicking.
14 KES_DEBUG_PAYLOAD_FLAG	KES_DEBUG_PAYLOAD_FLAG When running the stand alone debug mode on the error calculator, the state of this bit is presented to the KES engine as the input payload flag. 1b - Payload is set for 512 bytes data block.
13 KES_DEBUG_MODE4K	KES_DEBUG_MODE4K When running the stand alone debug mode on the error calculator, the state of this bit is presented to the KES engine as the input mode (4K or 2K pages). 1b - Mode is set for 4K NAND pages.
12 KES_DEBUG_KICK	KES_DEBUG_KICK Toggling causes KES engine FSM to start as if kick by the Bus Master. This allows stand alone testing of the KES and Chien Search engines. Be sure to set KES_BCH_DEBUG0_KES_STANDALONE mode to 1 before kicking.
11 KES_STANDALONE	KES_STANDALONE Set to one, cause the KES engine to suppress toggling the KES_BM_DONE signal to the bus master and suppress toggling the CF_BM_DONE signal by the CF engine. 0b - Bus master address generator for SYND_GEN writes operates normally. 1b - Bus master address generator always addresses last four bytes in Auxiliary block.
10 KES_DEBUG_STEP	KES_DEBUG_STEP Toggling this bit causes the KES FSM to skip passed the stall state if it is in DEBUG_STALL mode and completed processing a block.
9 KES_DEBUG_STALL	KES_DEBUG_STALL Set to one to cause KES FSM to stall after notifying Chien search engine to start processing its block but before notifying the bus master that the KES computation is complete. This allows a diagnostic to stall the FSM after each blocks key equations are solved. This also has the effect of stalling the CSFE search engine so it's state can be examined after it finishes processing the KES stalled block. 0b - KES FSM proceeds to next block supplied by bus master. 1b - KES FSM waits after current equations are solved and the search engine is started.
8 BM_KES_TEST_BYPASS	BM_KES_TEST_BYPASS 1 = Point all SYND_GEN writes to dummy area at the end of the AUXILLIARY block so that diagnostics can preload all payload, parity bytes and computed syndrome bytes for test the KES engine. 0b - Bus master address generator for SYND_GEN writes operates normally.

Table continues on the next page...

Field	Function
	1b - Bus master address generator always addresses last four bytes in Auxiliary block.
7-6 RSVD0	This field is reserved. This read-only field is reserved and always has the value 0.
5-0 DEBUG_REG_SELECT	DEBUG_REG_SELECT The value loaded in this bit field is used to select the internal register state view of KES engine or the Chien search engine.

9.6.3.18 KES Debug Read Register (DBGKESREAD)

9.6.3.18.1 Offset

Register	Offset
DBGKESREAD	110h

9.6.3.18.2 Function

The hardware BCH ECC accelerator key equation solver internal state machines and signals can be seen in the ECC debug registers.

9.6.3.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VALUES															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VALUES															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.6.3.18.4 Fields

Field	Function
31-0	VALUES

Field	Function
VALUES	This register returns the ROM BIST CRC value after a BIST test.

9.6.3.19 Chien Search Debug Read Register (DBGCSFEREAD)

9.6.3.19.1 Offset

Register	Offset
DBGCSFEREAD	120h

9.6.3.19.2 Function

The hardware BCH ECC accelerator Chien Search internal state machines and signals can be seen in the ECC debug registers.

9.6.3.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VALUES															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VALUES															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.6.3.19.4 Fields

Field	Function
31-0	VALUES
VALUES	Reserved

9.6.3.20 Syndrome Generator Debug Read Register (DBGSYNDGENREAD)

9.6.3.20.1 Offset

Register	Offset
DBGSYNDGENREAD	130h

9.6.3.20.2 Function

The hardware BCH ECC accelerator syndrome generator internal state machines and signals can be seen in the ECC debug registers.

9.6.3.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VALUES															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VALUES															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.6.3.20.4 Fields

Field	Function
31-0 VALUES	VALUES Reserved

9.6.3.21 Bus Master and ECC Controller Debug Read Register (DBGAHBMREAD)

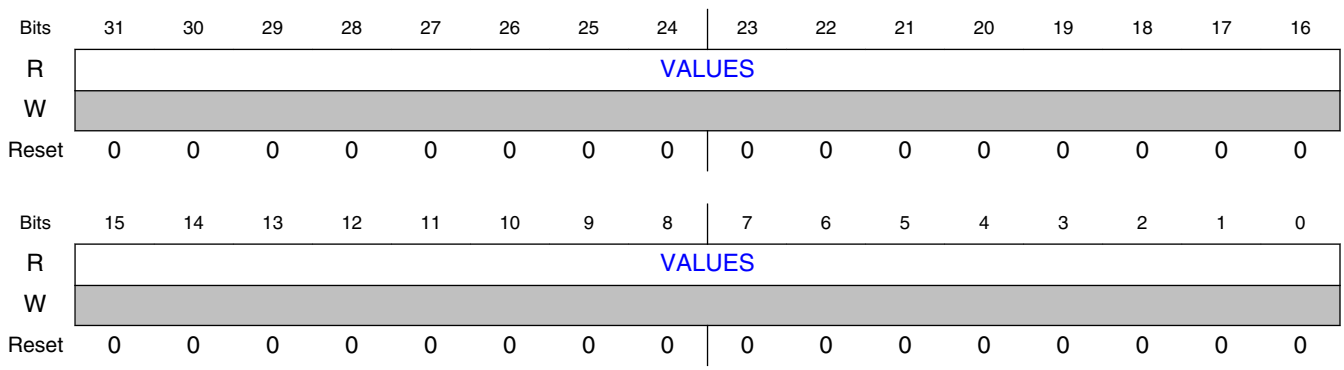
9.6.3.21.1 Offset

Register	Offset
DBGAHBMREAD	140h

9.6.3.21.2 Function

The hardware BCH ECC accelerator bus master, ECC controller internal state machines, and signals can be seen in the ECC debug registers.

9.6.3.21.3 Diagram



9.6.3.21.4 Fields

Field	Function
31-0 VALUES	VALUES Reserved

9.6.3.22 Block Name Register (BLOCKNAME)

9.6.3.22.1 Offset

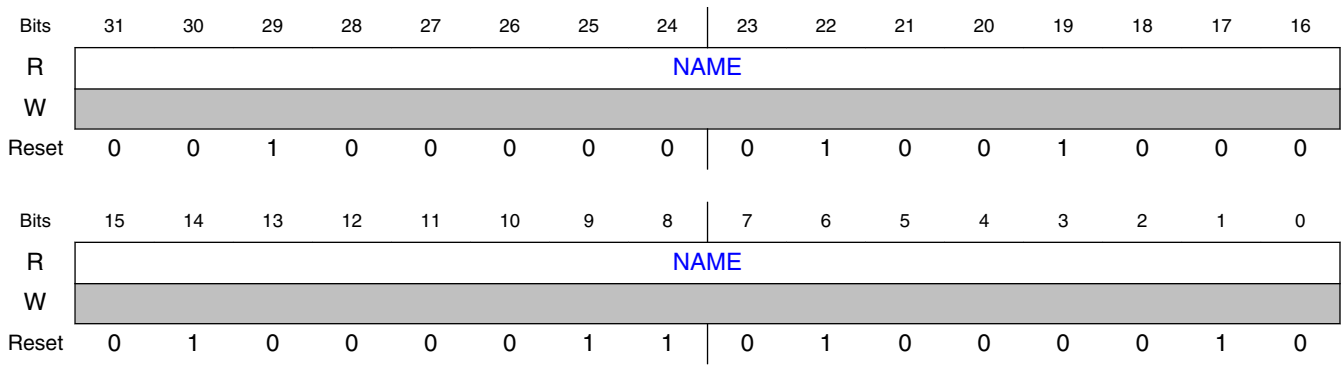
Register	Offset
BLOCKNAME	150h

9.6.3.22.2 Function

Read only view of the block name string BCH.

Fixed pattern read only value is for test purposes. It can be read as an ASCII string with the zero termination coming from the first byte of the BLOCKVERSION register.

9.6.3.22.3 Diagram



9.6.3.22.4 Fields

Field	Function
31-0 NAME	NAME The name is in the ASCII characters BCH (0x20, H, C, B).

9.6.3.23 BCH Version Register (VERSION)

9.6.3.23.1 Offset

Register	Offset
VERSION	160h

9.6.3.23.2 Function

This register always returns a known read value for debug purposes and indicates the version of the block and RTL version in use.

9.6.3.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STEP															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.6.3.23.4 Fields

Field	Function
31-24 MAJOR	MAJOR Fixed read-only value indicates the MAJOR field of the RTL version.
23-16 MINOR	MINOR Fixed read-only value indicates the MINOR field of the RTL version.
15-0 STEP	STEP Fixed read-only value reflecting the stepping of the RTL version.

9.6.3.24 Hardware BCH ECC Debug Register 1 (DEBUG1)

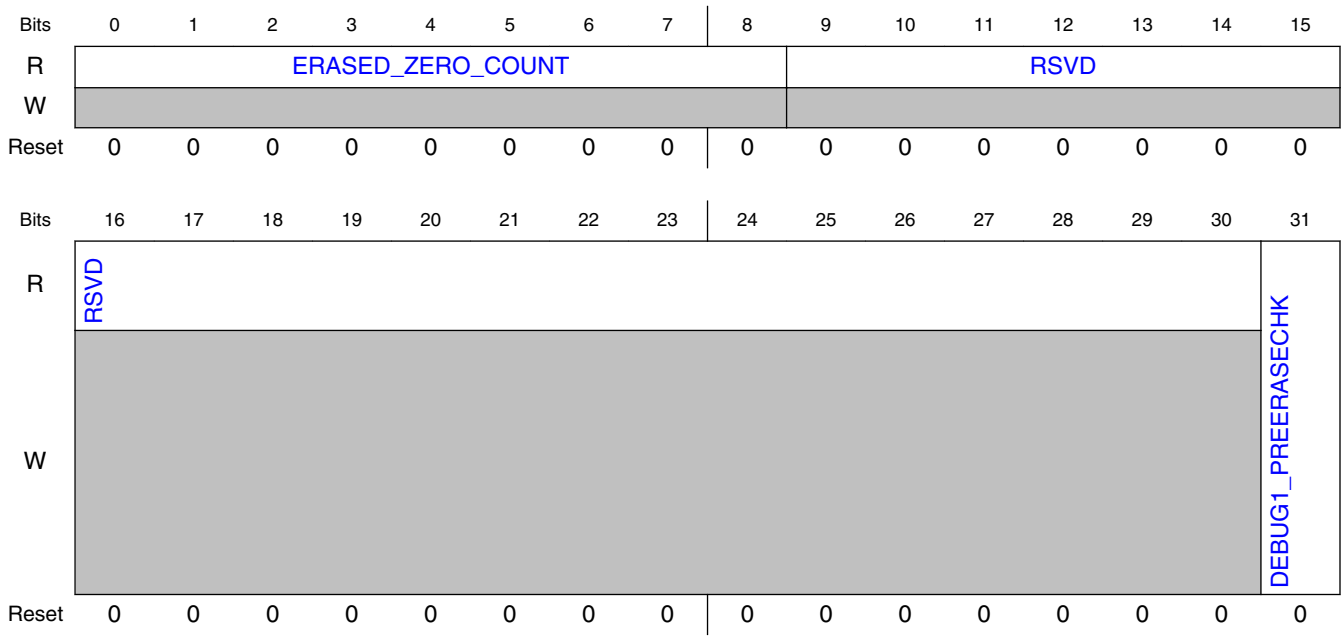
9.6.3.24.1 Offset

Register	Offset	Description
DEBUG1	170h	Hardware BCH ECC Debug Register 1
DEBUG1_SET	174h	Writing a 1 to a bit in this register sets the corresponding bit in DEBUG1
DEBUG1_CLR	178h	Writing a 1 to a bit in this register clears the corresponding bit in DEBUG1
DEBUG1_TOG	17Ch	Writing a 1 to a bit in this register toggles the corresponding bit in DEBUG1

9.6.3.24.2 Function

The BCH_DEBUG1 register provides erased zero count information and pre-erase check.

9.6.3.24.3 Diagram



9.6.3.24.4 Fields

Field	Function
0-8 ERASED_ZERO_COUNT	ERASED_ZERO_COUNT The zero counts on one page.
9-30 RSVD	This field is reserved. This read-only field is reserved and always has the value 0.
31 DEBUG1_PREERASECHK	DEBUG1_PREERASECHK Blank page enables pre-erase check. 0b - Turn off pre-erase check 1b - Turn on pre-erase check

9.7 General Purpose Media Interface (GPMI)

9.7.1 Overview

The GPMI controller is a flexible interface to supporting up to four NAND flash chip selects.

- ONFI 3.2, DDR mode, Samsung / Toshiba Toggle NAND protocol compatible.
- Fully configurable address and command behavior, providing support for future devices not yet specified.

The GPMI resides on the APBH. The GPMI also provides an interface to the BCH module to allow direct parity processing.

Registers are clocked on the HCLK domain. The I/O and pin timing are clocked on a dedicated GPMICLK domain. GPMICLK can be set to maximize I/O performance.

The following figure shows a block diagram of the GPMI controller.

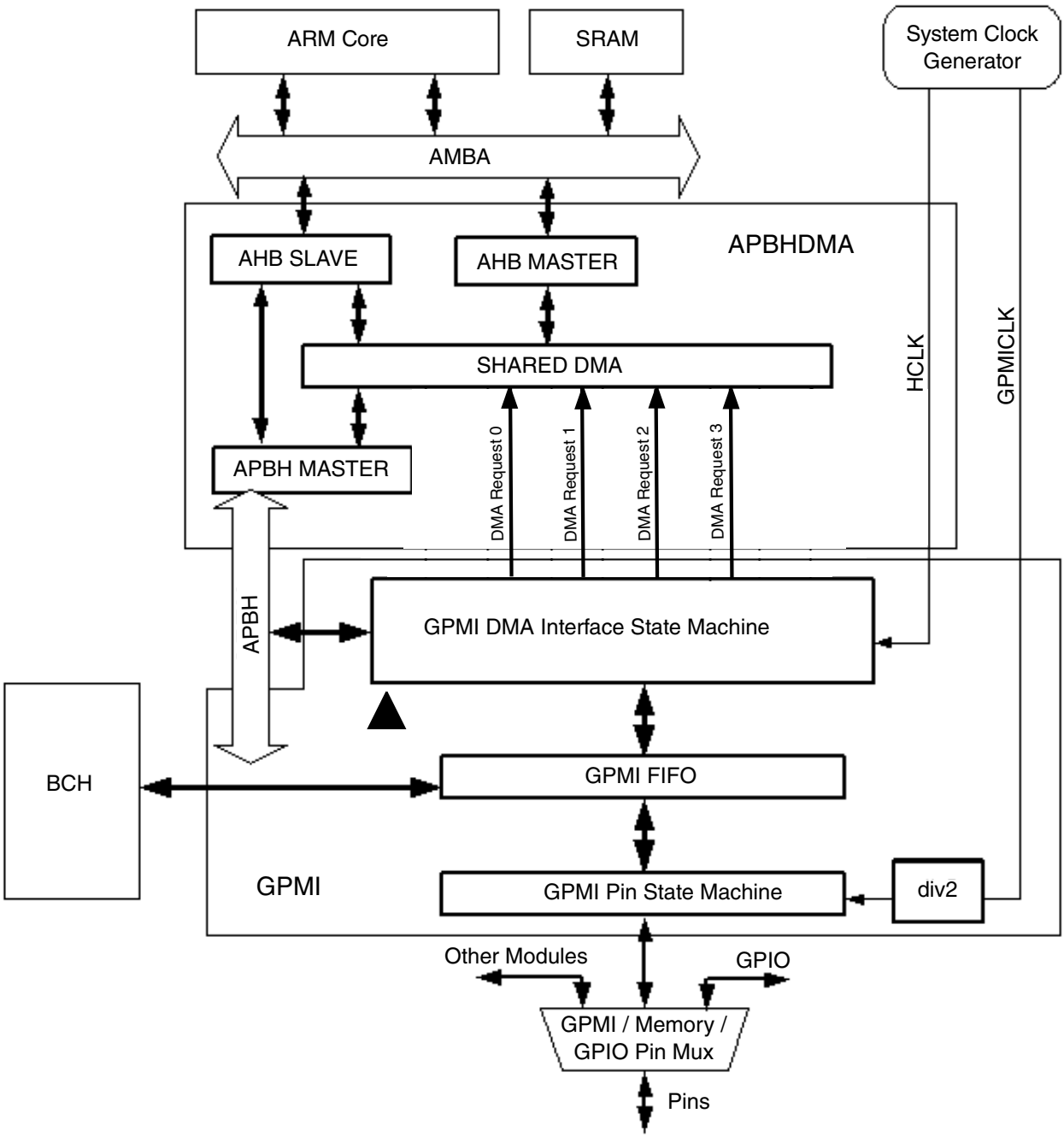


Figure 9-19. General-Purpose Media Interface Controller Block Diagram

9.7.2 Functional Description

9.7.2.1 GPMI NAND Mode

The general-purpose media interface has several features to efficiently support NAND:

- Individual chip select pins and ganged ready/busy pin for up to four NANDs.
- Individual state machine and DMA channel for each chip select.
- Special command modes work with DMA controller to perform all normal NAND functions without CPU intervention.
- Configurable timing based on a dedicated clock allows optimal balance of high NAND performance and low system power.

GPMI and DMA have been designed to handle complex multi-page operations without CPU intervention. The DMA uses a linked descriptor function with branching capability to automatically handle all of the operations needed to read/write multiple pages:

- **Data/Register Read/Write**-The GPMI can be programmed to read or write multiple cycles to the NAND address, command or data registers.
- **Wait for NAND Ready**-The GPMI's Wait-for-Ready mode can monitor the ready/busy signal of a single NAND flash and signal the DMA when the device has become ready. It also has a time-out counter and can indicate to the DMA that a time-out error has occurred. The DMAs can conditionally branch to a different descriptor in the case of an error.
- **Check Status**-The Read-and-Compare mode allows the GPMI to check NAND status against a reference. If an error is found, the GPMI can instruct the DMA to branch to an alternate descriptor, which attempts to fix the problem or asserts a CPU IRQ.

9.7.2.1.1 Multiple NAND Support

The GPMI supports up to four NAND chip selects, with ganged ready/busy pins. Since they share a data bus and control lines, the GPMI can only actively communicate with a single NAND at a time. However, all NANDs can concurrently perform internal read, write, or erase operations. With fast NAND flash and software support for concurrent NAND operations, this architecture allows the total throughput to approach the data bus speed, which can be as high as 50 MB/s (8-bit bus running at 50 MHz single clock edge) in asynchronous mode and 200MB/s (8-bit bus running at 100MHz both clock edges) in Source Synchronous mode.

There are two options for controlling the four NAND chip selects via the DMA interface. The first option is the one to one mapping, where the each DMA channel is tied to its own NAND chip select. For example DMA channel 'n' accesses only NAND attached to chip select 'n'. The second option is the decoupled mode where a DMA channel can access any or all NAND chip selects connected to the GPMI. A DMA channel will signify the NAND chip select it wants to access by writing its chip select value in the GPMI_CTRL0[CS] field and setting the GPMI_CTRL1[DECOUPLE_CS] to 1. This option is useful if software chooses to use only one DMA channel to access all the attached NAND devices.

9.7.2.1.2 GPMI NAND Timing and Clocking

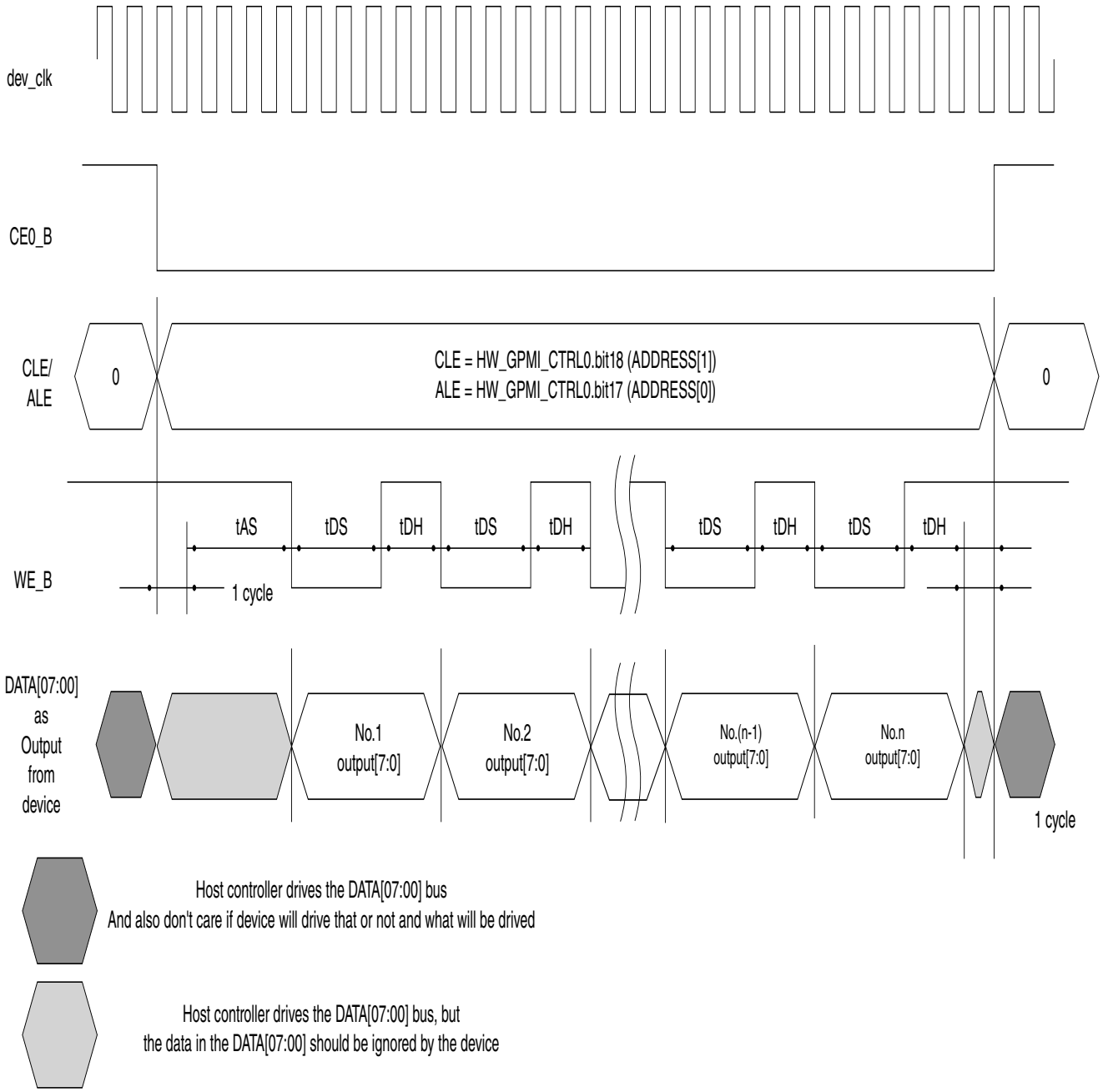
The dedicated clock, GPMICLK, is used as a timing reference for NAND flash I/O. Since various NANDs have different timing requirements, GPMICLK may need to be adjusted for each application.

While the actual pin timings are limited by the NAND chips used and the I/O pad configuration, the GPMI can support data bus speeds of up to 200 MHz x 8 bits. The actual read/write strobe timing parameters are adjusted as indicated in the register descriptions in Memory Map.

9.7.2.1.3 Basic NAND Timing

9.7.2.1.3.1 NAND Asynchronous Timing

The following figures illustrate the operation of the output (from host to device) timing parameters in NAND ONFI asynchronous mode.



- tAS is configurable by programming HW_GPMI_TIMING0 Address_Setup: in this example, Address_Setup = 4, tAS is equal to 4 dev_clk cycles.
- tDS is configurable by programming HW_GPMI_TIMING0 Data_Setup; in this example, Data_Setup = 3, tDS is equal to 3 dev_clk cycles
- tDH is configuarble by programming HW_GPMI_TIMING0 Data_Hold: in this example, Data_Hold = 2, tDH is equal to 2 dev_clk cycles
- tAS/tDS/tDH will extend, if the output data is not ready in device fifo.

Figure 9-20. Asynchronous Mode Basic Write Timing Diagram (command write, address write, or data write)

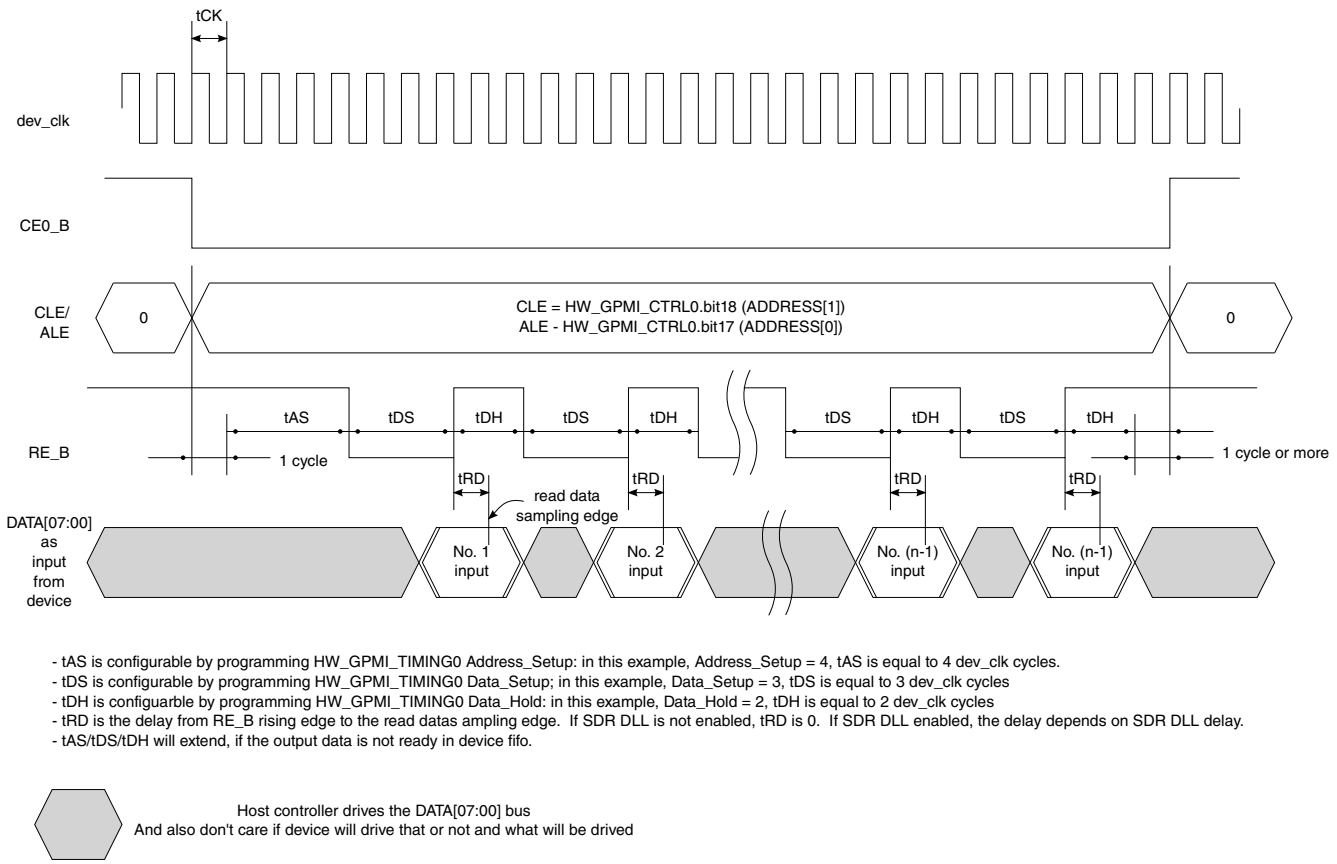


Figure 9-21. ONFI Asynchronous Mode Basic Read Timing Diagram (data read)

9.7.2.1.3.2 NAND Asynchronous EDO Mode Timing

In high-speed NANDs, the read data may not be valid until after the read strobe (NAND_RE_B) deasserts. This is the case when the minimum tDS is programmed to achieve higher bandwidth.

The GPMI implements a feedback read strobe to sample the read data. The feedback read strobe can be delayed to support fast nand EDO (Extended Data Out) timing where the read strobe may deassert before the read data is valid, and read data is valid for some time after read strobe. Nand EDO timings is applied typically for read cycle frequency above 33 MHz. See Figure 9-22.

The GPMI provides control over the amount of delay applied to the feedback read strobe. This delay depends on the maximum read access time (tREA) of the nand and the read pulse width (tRP) used to access the nand. tRP is specified by GPMI_TIMING0[DATA_SETUP] register. When (tREA + 4ns) is less than tRP, no

delay is required to sample to nand read data. (The 4ns provides adequate data setup time for the GPMI.) In this case set GPMI_CTRL1[HALF_PERIOD] = 0; GPMI_CTRL1[RDN_DELAY] = 0; GPMI_CTRL1[DLL_ENABLE] = 0.

When (tREA + 4ns) is greater than or equal to tRP, a delay of the feedback read strobe is required to sample to nand read data. This delay is equal to the difference between these two timings:

DELAY = tREA + 4ns - tRP.

Since the GPMI delay chain is limited to 6ns maximum, if DELAY > 6ns then increase tRP by increasing the value of GPMI_TIMING0[DATA_SETUP] until DELAY is less than or equal to 6ns.

The GPMI programming for this DELAY depends on the GPMICLK period. The GPMI DLL will not function properly if the GPMICLK period is greater than 12ns: disable the DLL if this is the case. If the GPMICLK period is greater than 6ns (and not greater than 12ns), set the GPMI_CTRL1[HALF_PERIOD]=1; This will cause the DLL reference period (RP) to be one-half of the GPMICLK period. If the GPMICLK period is 6ns or less then set the GPMI_CTRL1[HALF_PERIOD]=0; This will cause the DLL reference period (RP) to be equal to the GPMICLK period. DELAY is a multiple (0 to 1.875) of RP.

The GPMI_CTRL1[RDN_DELAY] is encoded as a 1-bit integer and 3-bit fraction delay factor. DELAY is a multiple of the delay factor and the reference period. See table below for details.

Table 9-14. RDN DELAY

HW_GPMI_CTRL1[RDN_DELAY]	Delay Factor
0	0.000
1	0.125
2	0.250
3	0.375
4	0.500
5	0.625
6	0.750
7	0.875
8	1.000
9	1.125
10	1.250
11	1.375
12	1.500
13	1.625

Table continues on the next page...

Table 9-14. RDN DELAY (continued)

HW_GPMI_CTRL1[RDN_DELAY]	Delay Factor
14	1.750
15	1.875

DELAY = DelayFactor x RP or DELAY = GPMI_CTRL1[RDN_DELAY] x 0.125 x RP.

Use this equation to calculate the value for GPMI_CTRL1[RDN_DELAY]. Then set GPMI_CTRL1[DLL_ENABLE]=1.

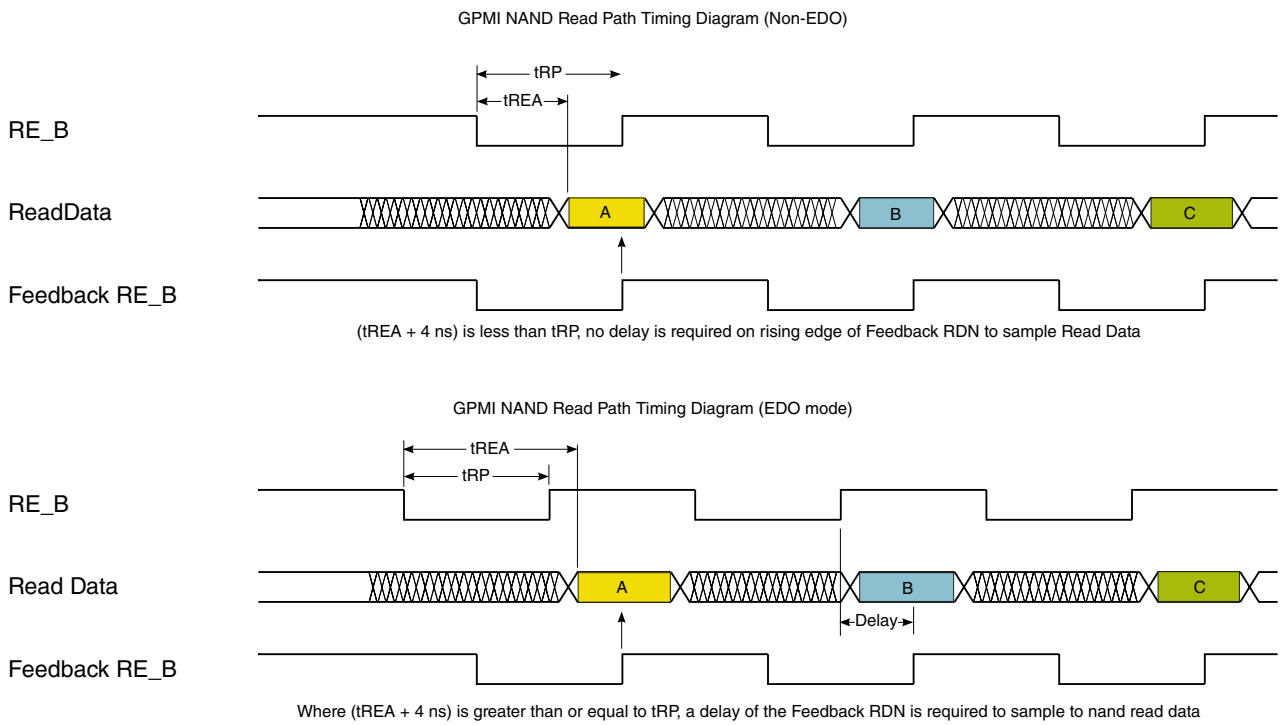


Figure 9-22. NAND Read Path Timing

For example, a NAND with tREAmax = 20 ns, tRPmin = 12 ns, and tRCmin = 25 ns (read cycle time) may be programmed as follows:

- GPMICLK clock frequency: Consider 480/6 = 80 MHz which is 12.5 ns clock period. This is too close to the minimum NAND spec if we program the data setup and hold to 1 GPMICLK cycle. Consider 480/7=68.57 MHz which is 14.58 ns clock period. With data setup and hold set to 1, we have a tRP of 14.58ns and a tRC of 29.16 ns (good margins).
- Since (tREA +4ns) is greater than tRP, required DELAY = tREA + 4ns - tRP = 20 + 4 - 14.58ns = 9.42 ns.

- GPMI_CTRL1[HALF_PERIOD] =0, since GPMICLK period is less than 16ns. So RP=GPMICLK period = 14.58ns.
- DELAY = GPMI_CTRL1[RDN_DELAY] x 0.125 x RP. 9.42 ns = GPMI_CTRL1[RDN_DELAY] x 0.125 x 14.58ns. GPMI_CTRL1[RDN_DELAY] = 5 (round of 5.169)

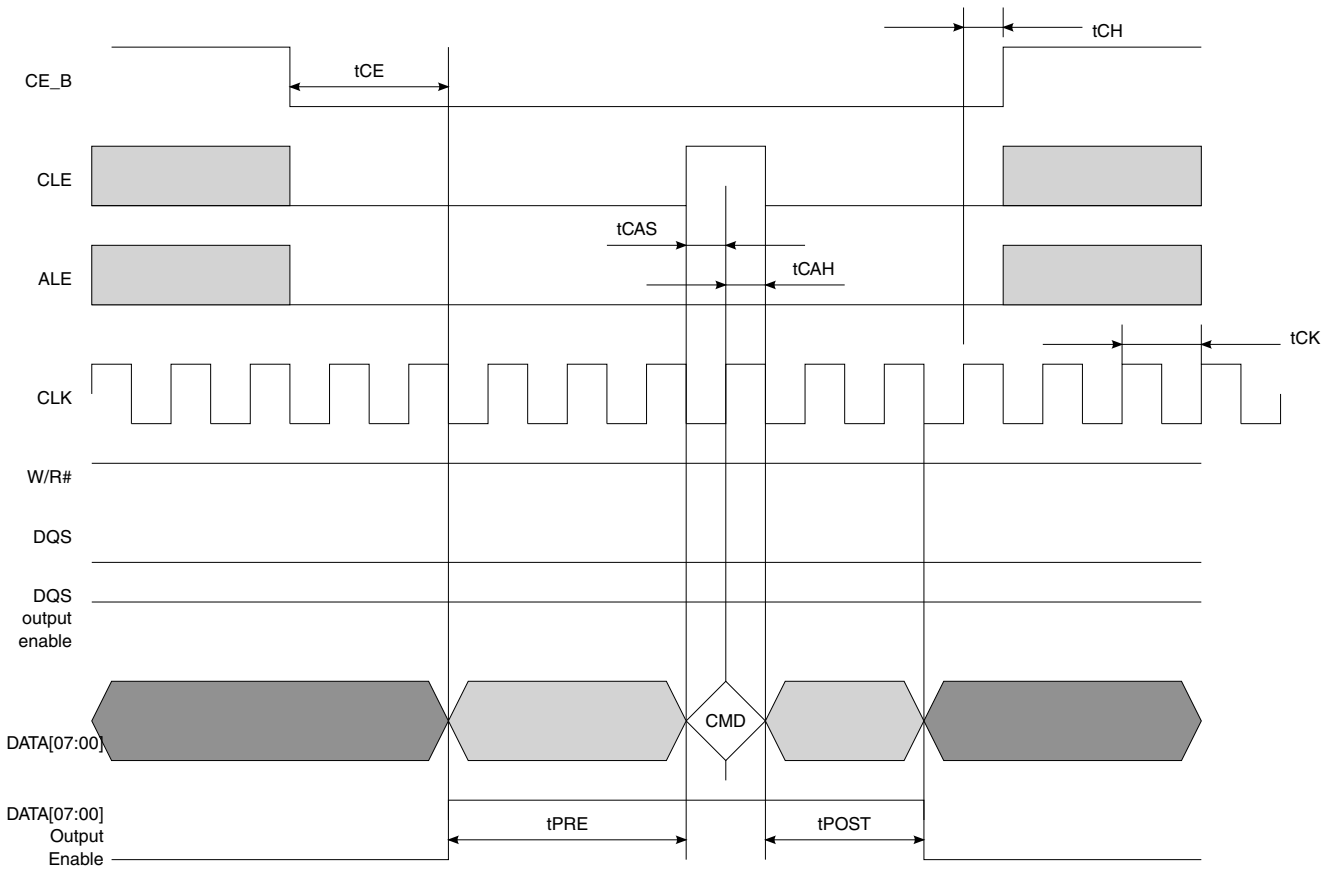
NOTE

It is recommended that the drive strength of NAND_RE_B and NAND_WE_B output pins be set to 8 mA. This will reduce the transition time under heavy loads. Low transition times will be important when NAND interface read and write cycle times are below 30 ns. The other GPMI pins may remain at 4 mA, since their frequency is only up to half that of NAND_RE_B and NAND_WE_B.

9.7.2.1.3.3 NAND ONFI Source Synchronous Mode Timing

NOTE

In the following figures, CLK shares the same pin as WE_B in Async Mode. And W/R# shares the same pin as RE_B in Async Mode.

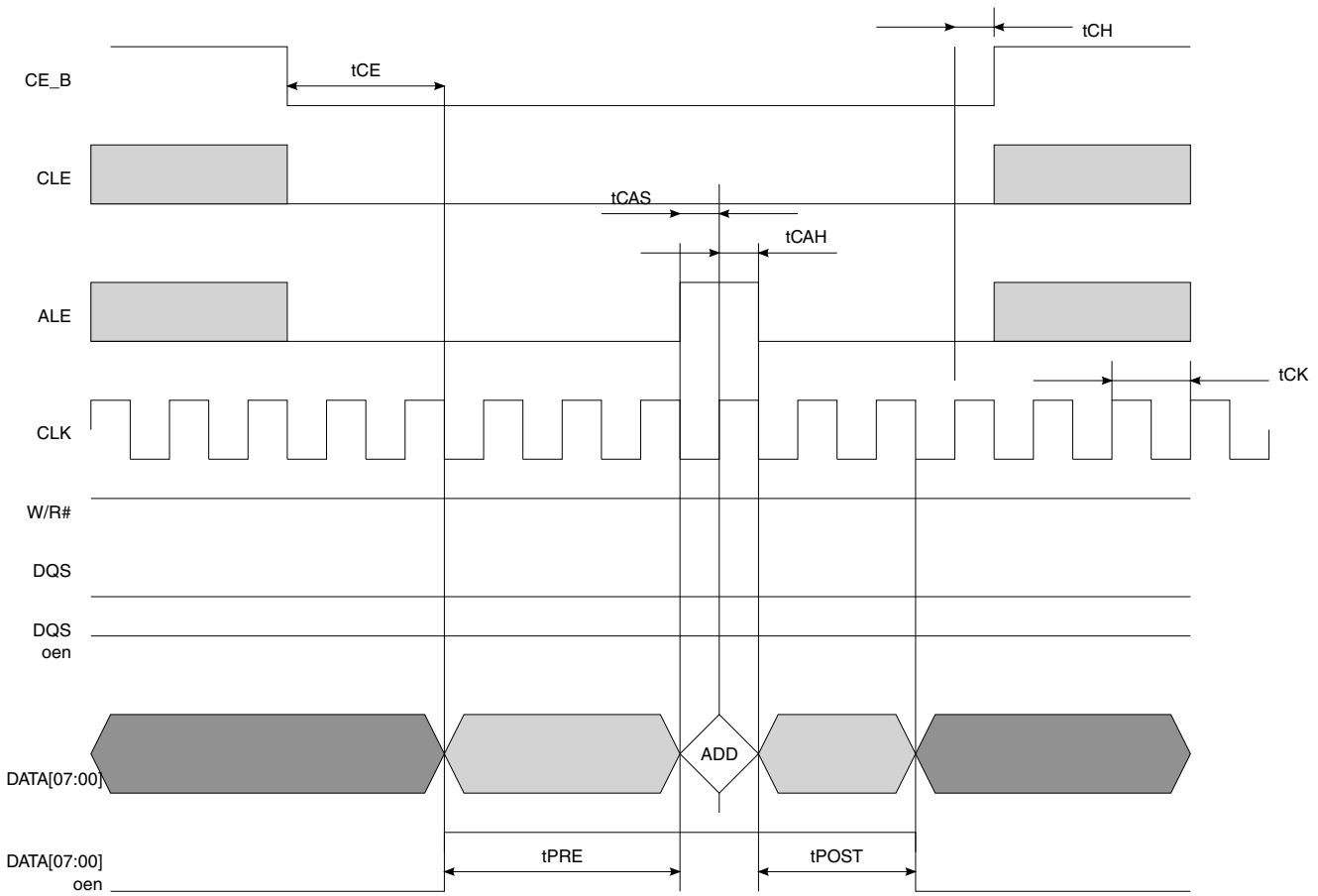


Host controller does not drives the DATA[07:00] bus
And also don't care if device will drive that or not and what will be drived

Host controller drives the DATA[07:00] bus, but
the data in the DATA[07:00] should be ignored by the device

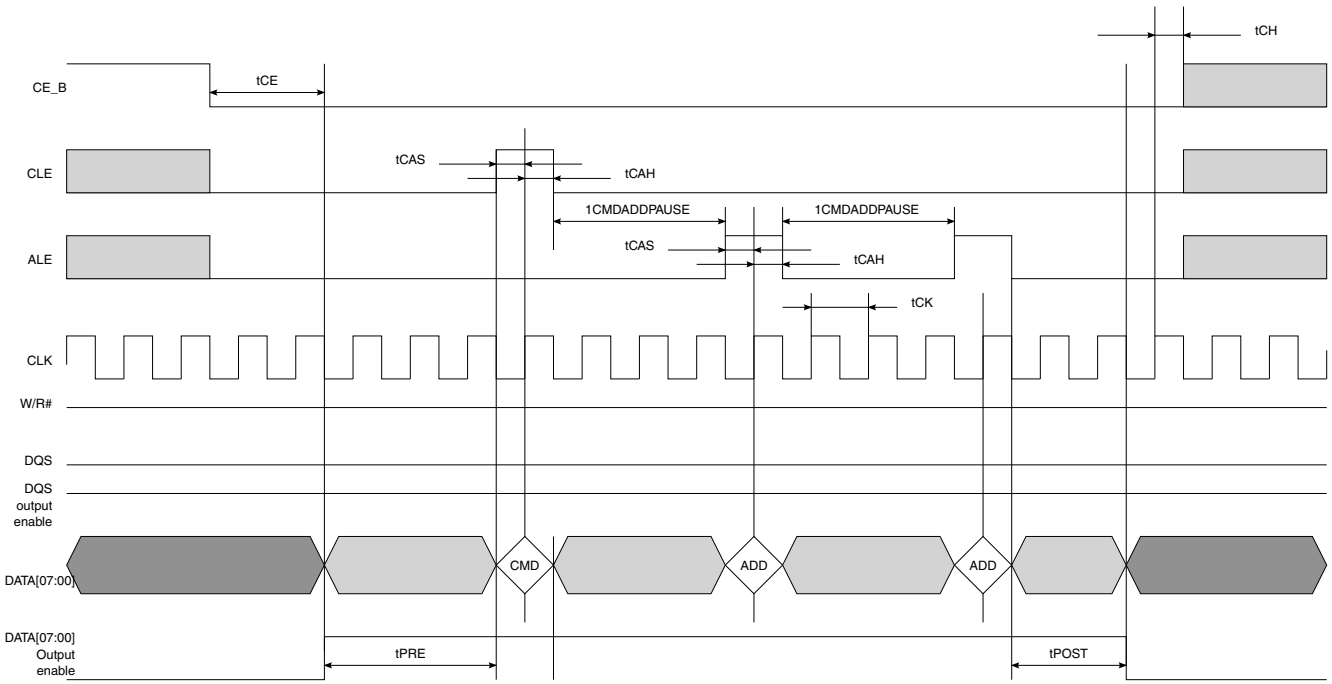
tCK is equal to device clock period
tCAS = 0.5 * tCK
tCAH = 0.5 * tCK
tCE = HW_GPMI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY * tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK

Figure 9-23. ONFI Source Synchronous Mode Basic Command Write Timing Diagram



tCK is equal to device clock period
tCAS = 0.5 * tCK
tCAH = 0.5 * tCK
tCE = HW_GPMI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY * tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK

Figure 9-24. ONFI Source Synchronous Mode Basic Address Write Timing Diagram

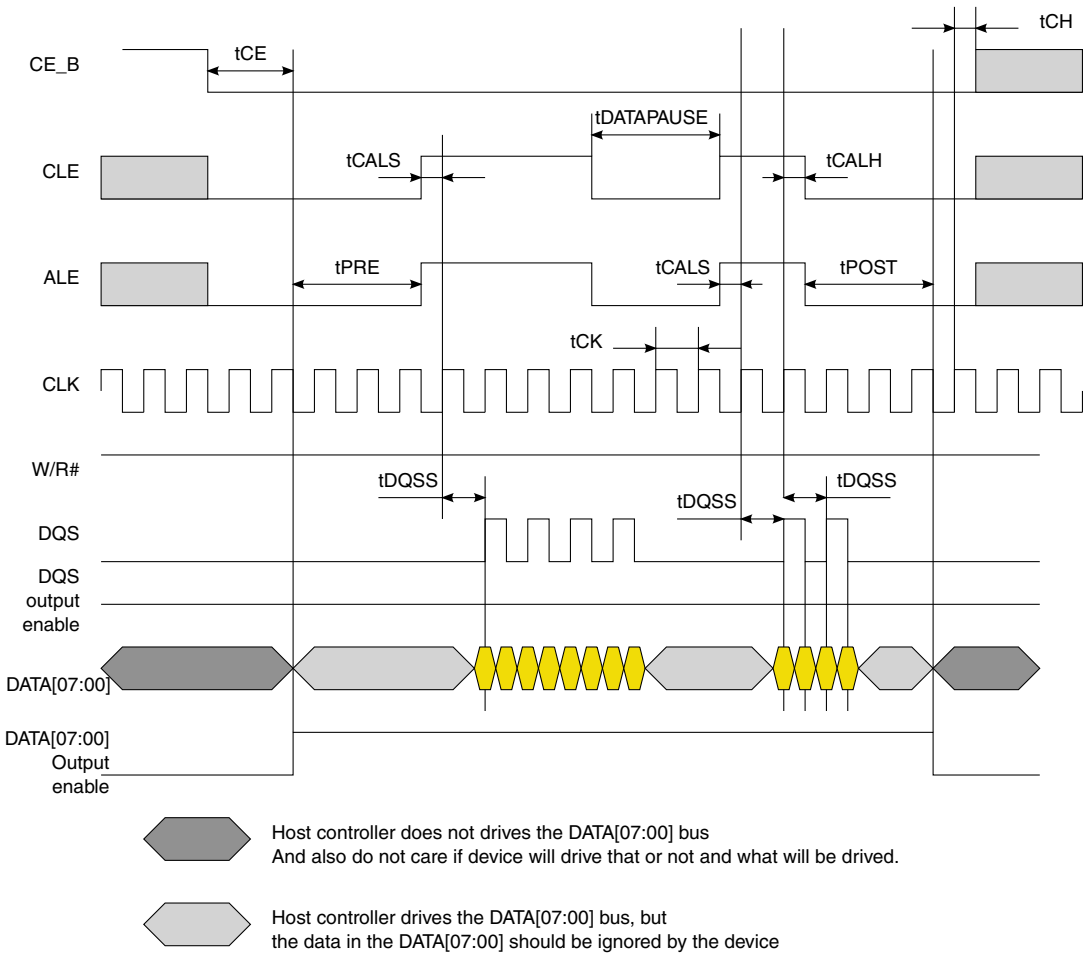


Host controller does not drives the DATA[07:00] bus
And also do not care if device will drive that or not and what will be drived.

Host controller drives the DATA[07:00] bus, but
the data in the DATA[07:00] should be ignored by the device

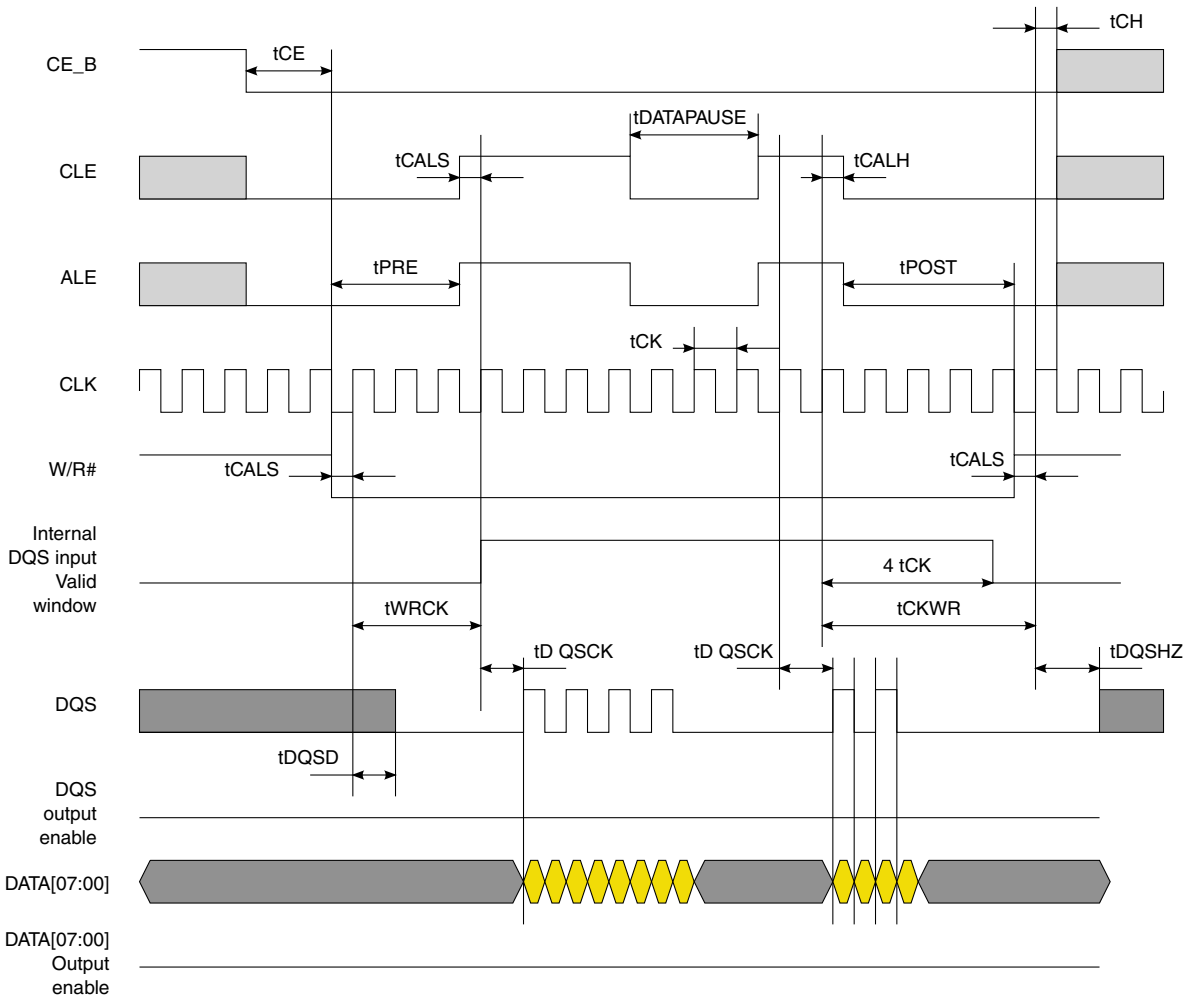
tCK is equal to device clock period
tCAS = 0.5 * tCK
tCAH = 0.5 *tCK
tCE = HW_GPMI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY * tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK
tCMDADDPAUSE = HW_GPMI_TIMING2.CMDADD_PAUSE * tCK

Figure 9-25. ONFI Source Synchronous Mode Command + Address Write Timing Diagram



tCK is equal to device clock period
tCE = HW_GPI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tCALS = 0.5 * tCK
tCALH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY *tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK
tDATAPAUSE = HW_GPMI_TIMING2.DATA_PAUSE * tCK
tDQSS = tCK

Figure 9-26. ONFI Source Synchronous Mode Data Write Timing Diagram



tCK is equal to device clock period
tCE = HW_GPI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tCALS = 0.5 * tCK
tCALH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY *tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK
tDATAPAUSE = HW_GPMI_TIMING2.DATA_PAUSE * tCK
tWRCK/tCKWR/tDQSD/tDASCK/tDASHZ are device parameters

Figure 9-27. ONFI Source Synchronous Mode Data Read Timing Diagram

9.7.2.1.3.4 NAND Toggle Mode Timing

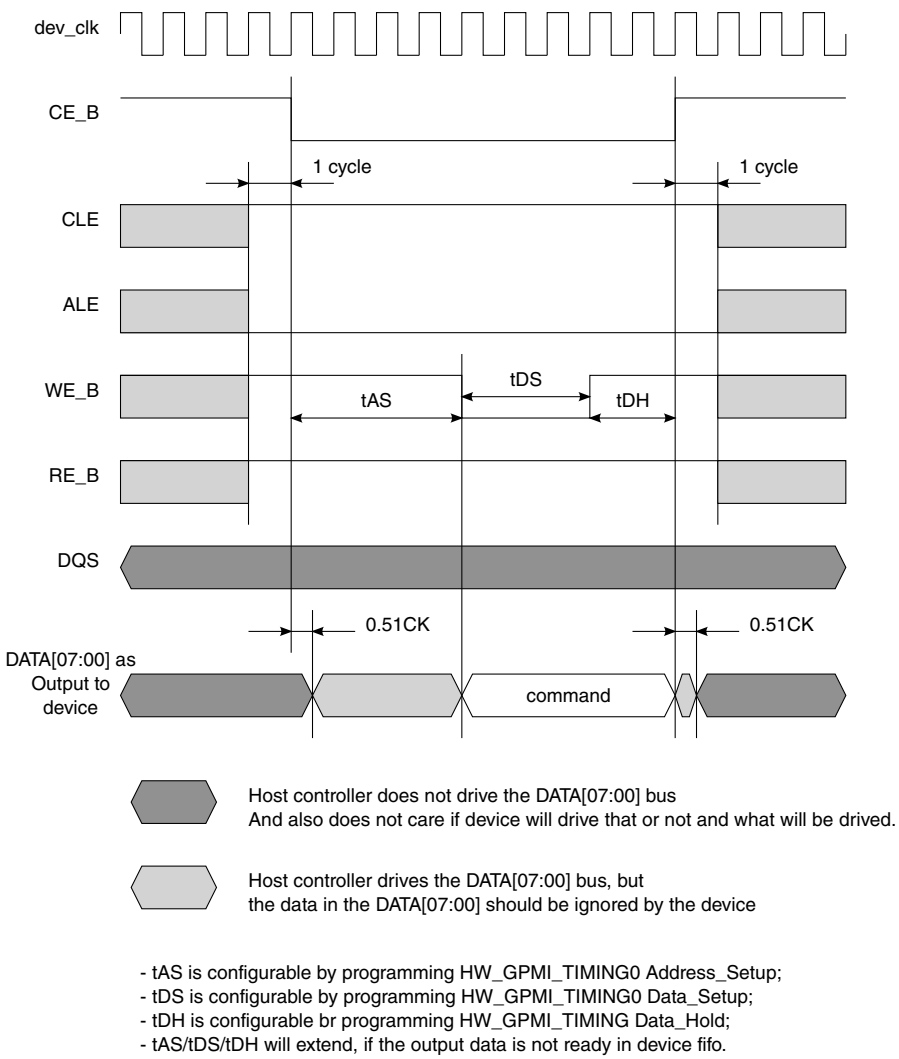
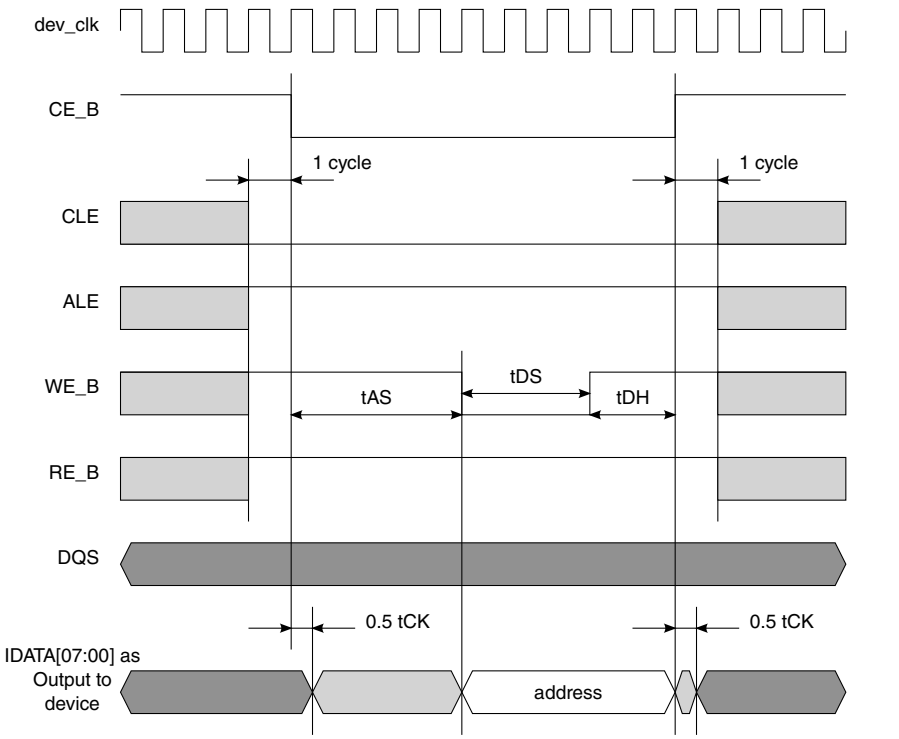


Figure 9-28. Samsung Toggle Mode Basic Command Write Timing Diagram



Host controller does not drive the DATA[07:00] bus
And also does not care if device will drive that or not and what will be drived.

Host controller drives the DATA[07:00] bus, but
the data in the DATA[07:00] should be ignored by the device

- tAS is configurable by programming HW_GPMI_TIMING0 Address_Setup;
- tDS is configurable by programming HW_GPMI_TIMING0 Data_Setup;
- tDH is configurable br programming HW_GPMI_TIMING Data_Hold;
- tAS/tDS/tDH will extend, if the output data is not ready in device fifo.

Figure 9-29. Samsung Toggle Mode Basic Address Write Timing Diagram

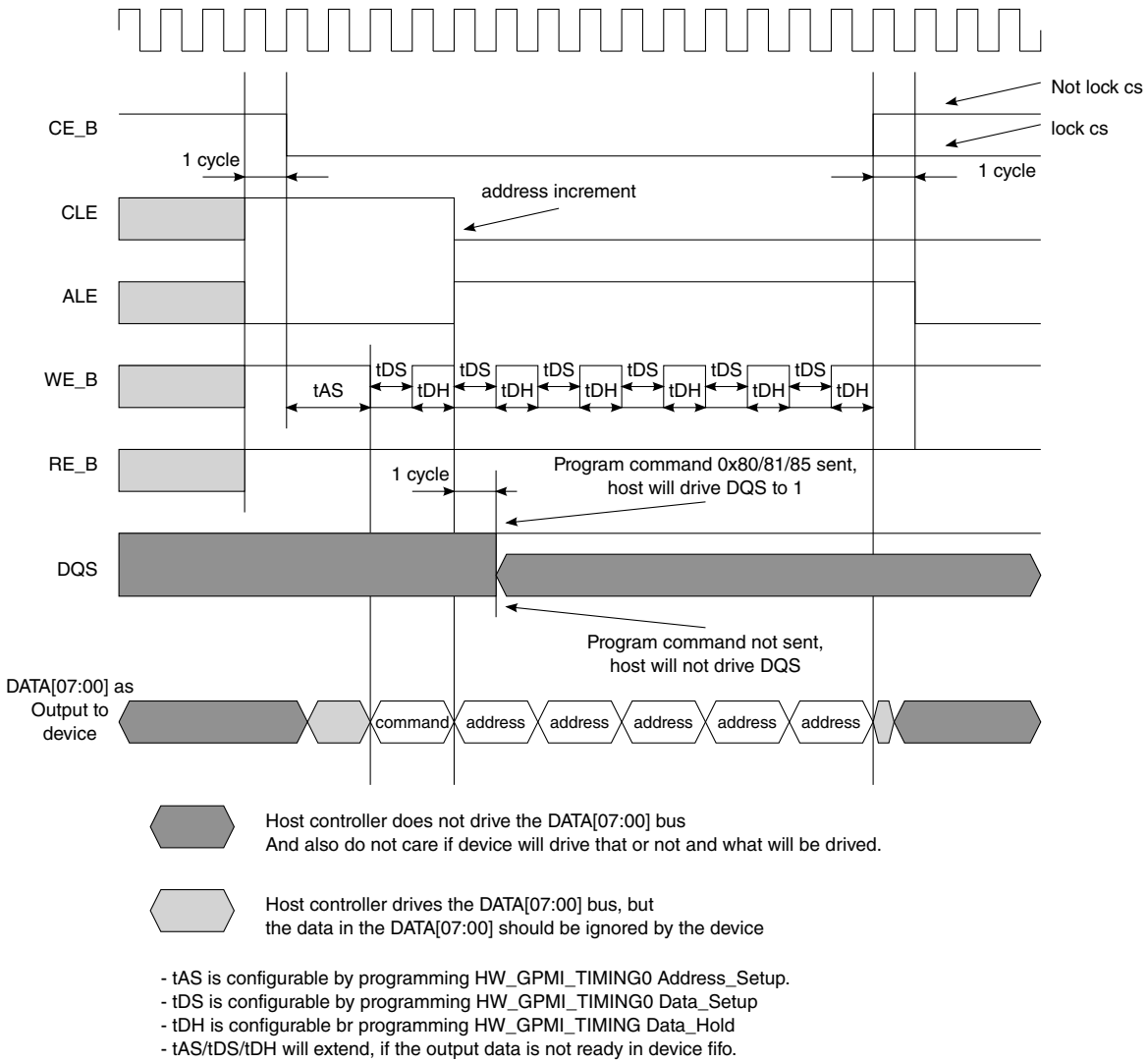


Figure 9-30. Samsung Toggle Mode Basic Command + Address Timing Diagram

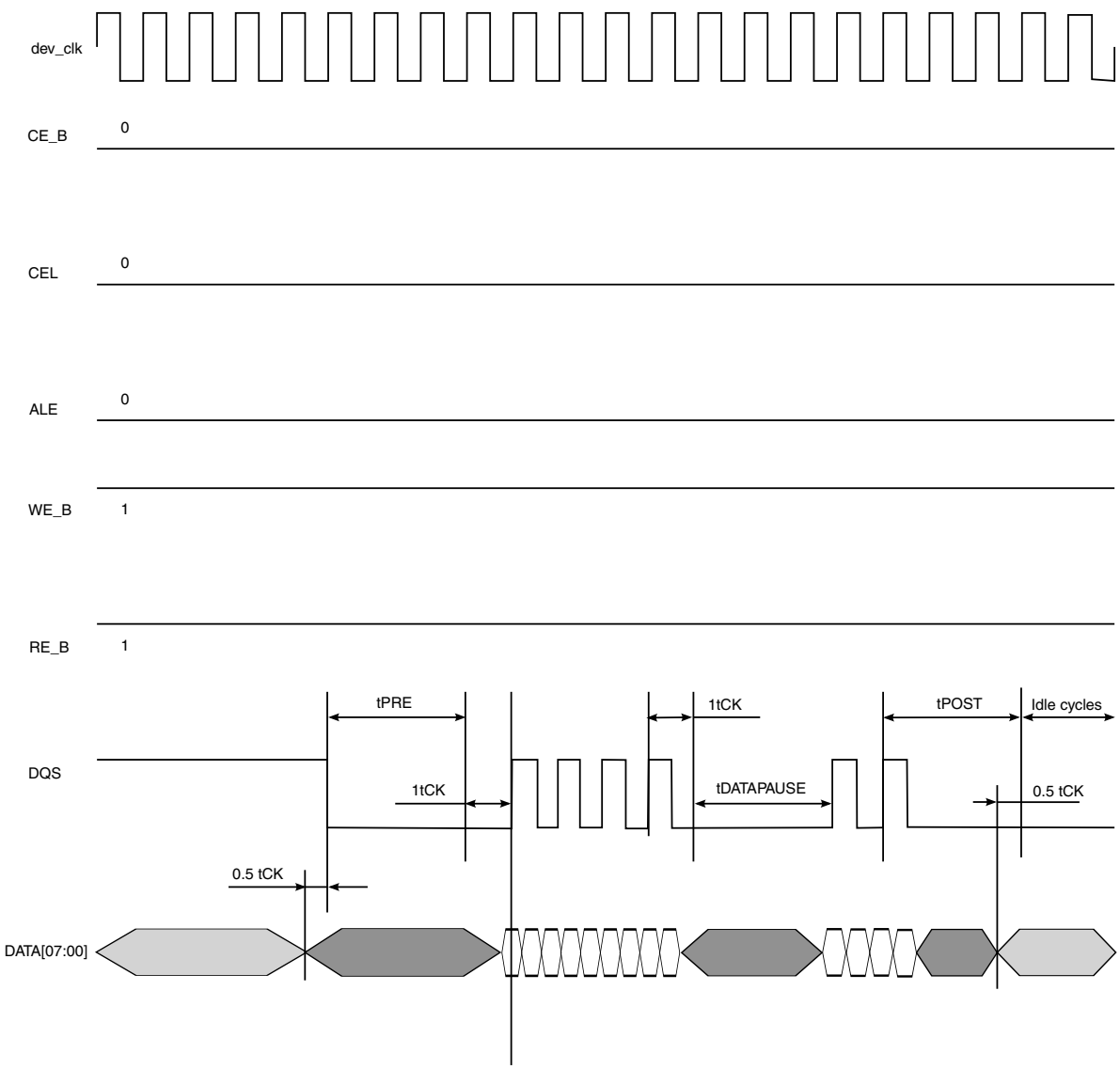
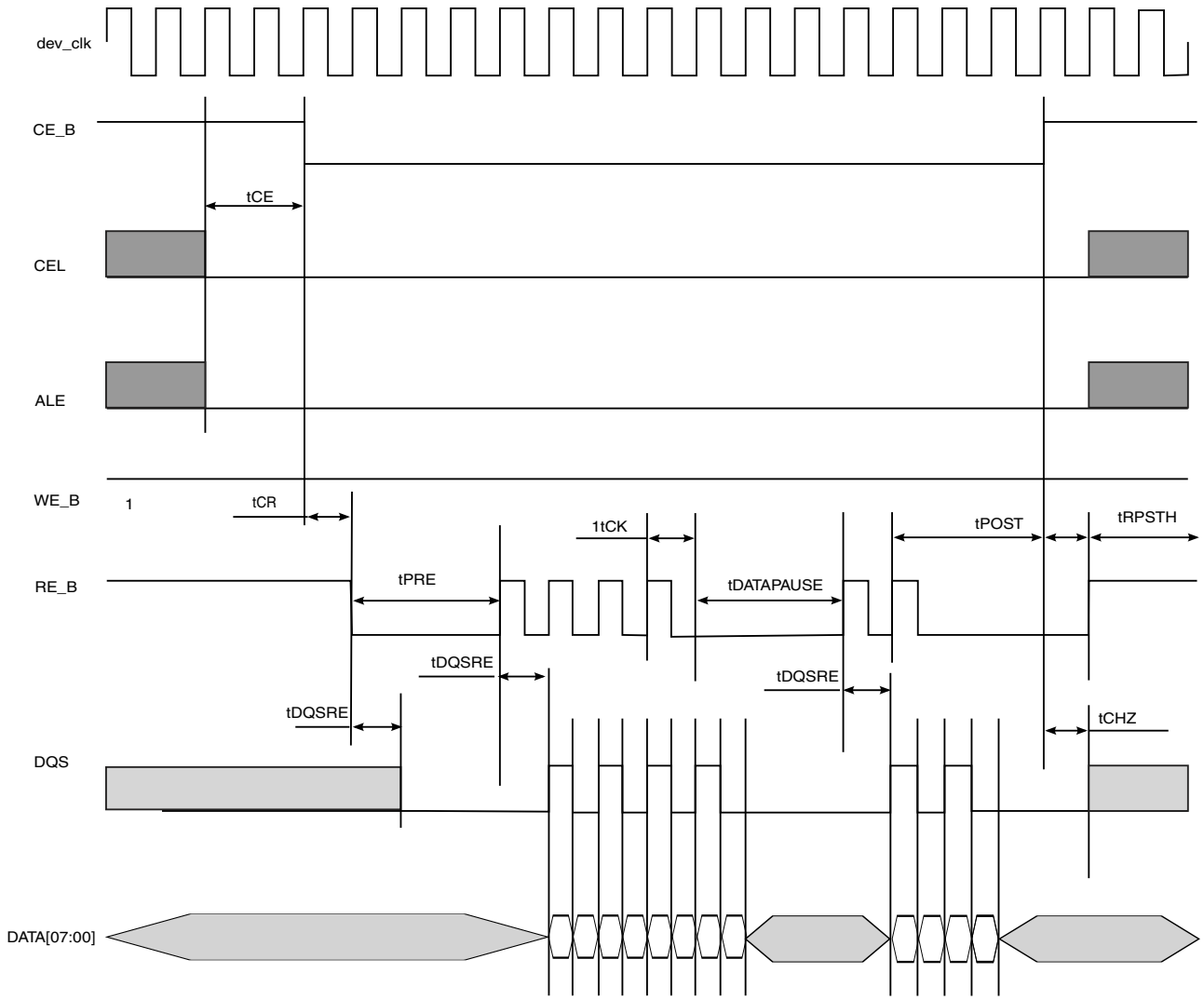


Figure 9-31. Toggle Mode Data Write Timing Diagram



Host controller does not drive the DATA[07:00] bus.
It also does not care if the device will drive or what will be driven.

Host controller drives the DATA[07:00] bus, but
the data in the DATA[07:00] should be ignored by device.

tCK is equal to device clock period

tCE = HW_GPMI_TIMING2.CE_DELAY* tCK

tPRE = (HW_GPMI_TIMING2.PREAMBLE_DELAY - HW_GPMI_TIMING2.TCR) * tCK

tPOST = HW_GPMI_TIMING2.POSTAMBLE_DELAY* tCK

tDATAPAUSE = HW_GPMI_TIMING2.DATA_PAUSE* tCK

tCR = (HW_GPMI_TIMING2.TCR + 1) * tCK

tRPSTH = HW_GPMI_TIMING2.TRPSTH * tCK

Figure 9-32. Toggle Mode Data Read Timing Diagram

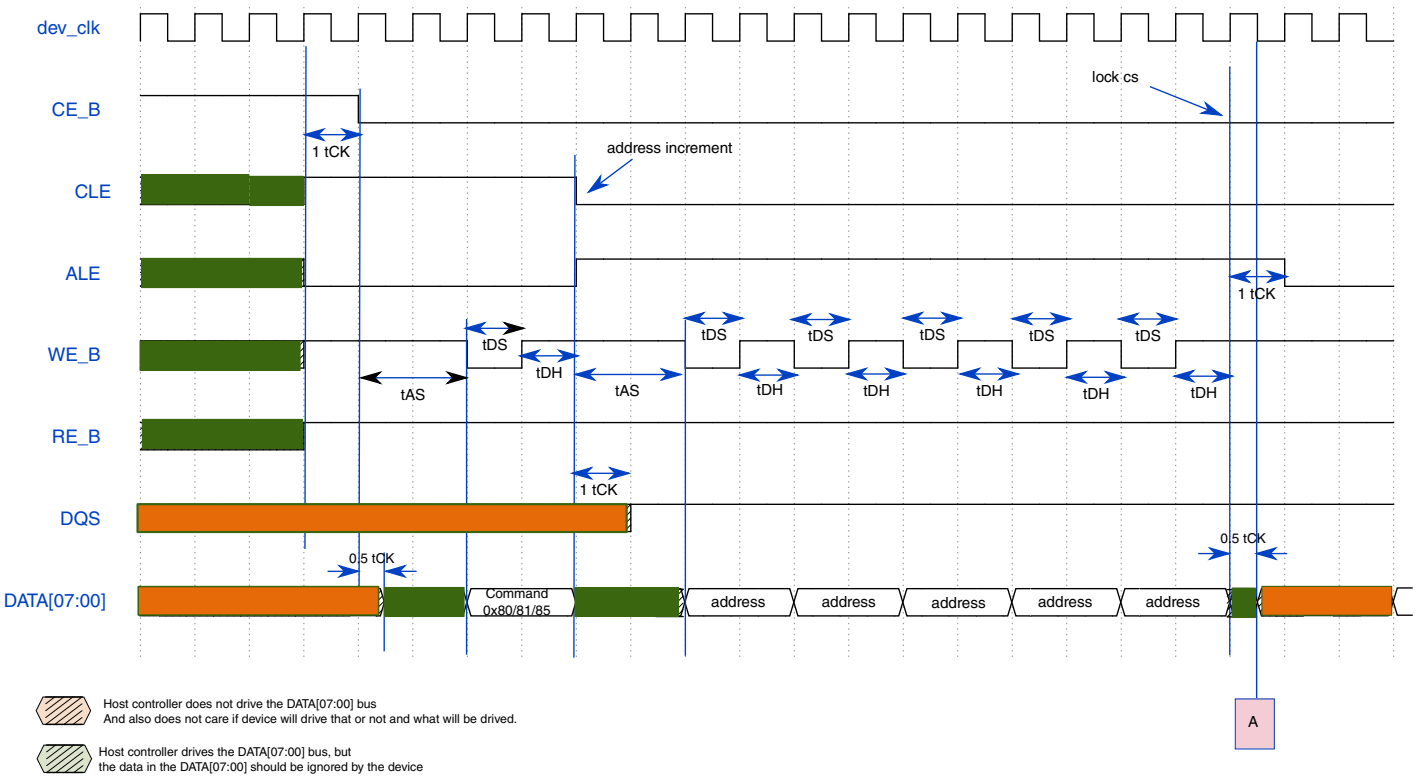
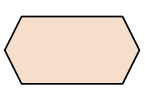
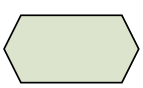


Figure 9-33. Toggle Mode Program Timing Diagram (A)



Host controller does not drive the DATA[07:00] bus
And also does not care if device will drive that or not and what will be drived.



Host controller drives the DATA[07:00] bus, but
the data in the DATA[07:00] should be ignored by the device

Figure 9-34. Toggle Mode Program Timing Diagram (B)

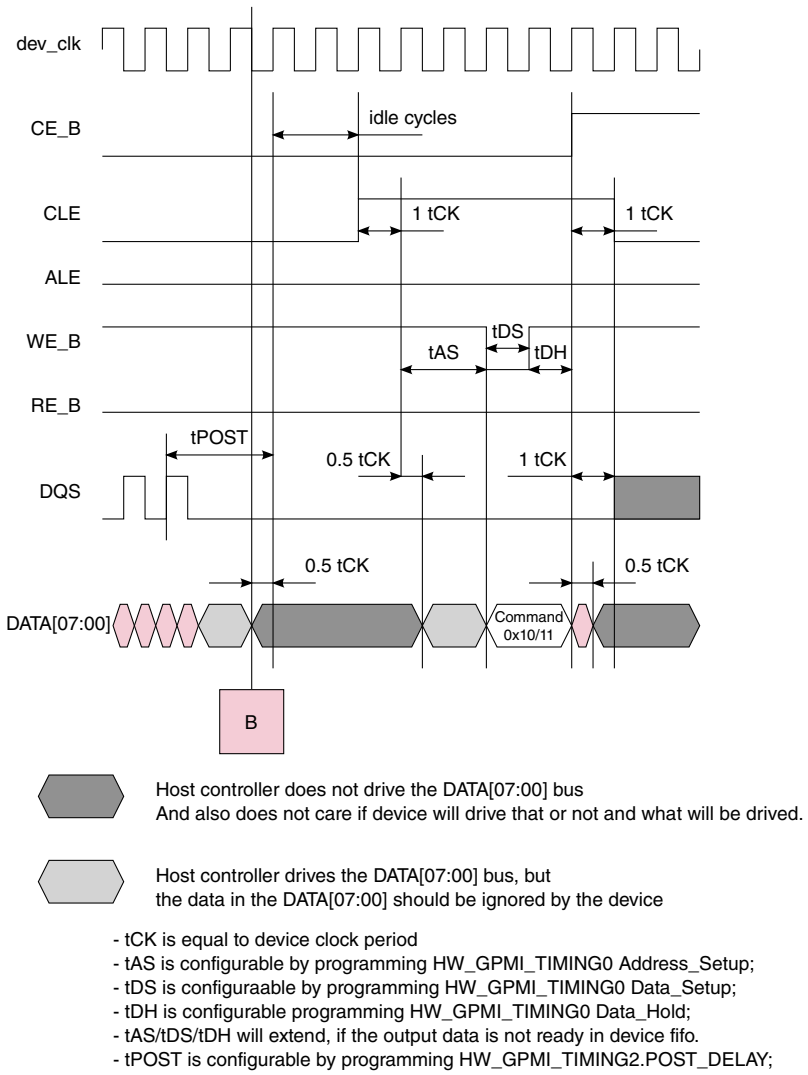


Figure 9-35. Toggle Mode Program Timing Diagram (C)

9.7.2.2 Hardware BCH Interface

The GPMI provides an interface to the BCH module. This reduces the SOC bus traffic and the software involvement.

The GPMI also provides an interface to the Randomizer module. The Randomizer can generate random data based on BCH ECC encoded / decoded data. It can be employed to reduce the disturbances caused by neighboring cells in the NAND chip, thus reducing bit errors. To enable the Randomizer module, set GPMI_ECCCTRL[RANDOMIZER_ENABLE] to 1, then set GPMI_ECCCOUNT[RANDOMIZER_PAGE] to select randomizer page number needed to be randomized. All these registers can be programmed by the DMA chain. The

randomized data should start from the zero column address and be the size of the whole NAND page. If the randomizer function is enabled, GPMI_ECCCTRL[ENABLE_ECC] should also be enabled. To bypass BCH error correction function, set BCH_FLASHxLAYOUT0[ECC0] and BCH_FLASHxLAYOUT1[ECCN] to 0.

When BCH ECC is enable, parity information is inserted on-the-fly during writes to 8-bit NAND devices. The BCH will supply payload and parity to the GPMI to write to the NAND. During NAND reads, parity is checked and ECC processing is performed after each read block. In this case the GPMI reads the NAND device and redirects the data and parity to the BCH module for ECC processing.

To program the BCH for NAND writes, remove the soft reset and clock gates from BCH_CTRL[SFTRST] and BCH_CTRL[CLKGATE]. The bulk of BCH programming is actually applied to the GPMI via PIO operations embedded in its DMA command structures. This has a subtle implication when writing to the GPMI ECC registers: access to the these registers must be written in progressive register order. Thus, to write to the GPMI_ECCCOUNT register, write first (in order) to registers GPMI_CTRL0, GPMI_COMPARE, and GPMI_ECCCTRL before writing to GPMI_ECCCOUNT. These additional register writes need to be accounted for in the CMDWORDS field of the respective DMA channel command register.

NOTE

Note that the GPMI_PAYLOAD and GPMI_AUXILIARY pointers need to be word-aligned for proper ECC operation. If those pointers are non-word-aligned, then the BCH engine will not operate properly and could possibly corrupt system memory in the adjoining memory regions.

9.7.2.3 Clocks

The table found here describes the clock sources for GPMI. Please see clock control module for clock setting, configuration and gating information.

Table 9-15. GPMI Clocks

Clock name	Description
bch_input_apb_clk	BCH to APBH input clock
gpmi_bch_input_bch_clk	BCH input clock
gpmi_bch_input_gpmi_io_clk	GPMI IO input clock
gpmi_input_apb_clk	GPMI to APBH clock

9.7.2.4 Behavior During Reset

A soft reset (SFTRST) can take multiple clock periods to complete, so do NOT set CLKGATE when setting SFTRST. The reset process gates the clocks automatically.

9.7.3 External Signals

The table found here describes the external signals of GPMI.

Table 9-16. GPMI External Signals

Signal	Description
NAND_ALE	Address latch enable signal
NAND_CEn_B	Chip enable signal
NAND_CLE	Command latch enable signal
NAND_DATAn	Data signal
NAND_DQS	DQS signal
NAND_RE_B	Read enable signal
NAND_READY	Ready signal
NAND_WE_B	Write enable signal
NAND_WP_B	Wait polarity signal
NAND_TESTER_TRIGGER	Tester trigger signal

9.7.4 GPMI register descriptions

The following registers provide control for programmable elements of the GPMI module.

NOTE

All ATA or UDMA features are not supported for the chip.

NOTE

GPMI does not support the Set feature command in Toggle mode. The NANDF_DQS output is only enabled in program operation for Toggle mode, but the Set feature command also needs to use the NANDF_DQS signal to write data to Toggle NAND flash. So the Set feature command in Toggle mode is not supported.

9.7.4.1 GPMI memory map

GPMI base address: 3300_2000h

Offset	Register	Width (In bits)	Access	Reset value
0h	GPMI Control Register 0 Description (CTRL0)	32	RW	C000_0000h
4h	GPMI Control Register 0 Description (CTRL0_SET)	32	RW	C000_0000h
8h	GPMI Control Register 0 Description (CTRL0_CLR)	32	RW	C000_0000h
Ch	GPMI Control Register 0 Description (CTRL0_TOG)	32	RW	C000_0000h
10h	GPMI Compare Register Description (COMPARE)	32	RW	0000_0000h
20h	GPMI Integrated ECC Control Register Description (ECCCTRL)	32	RW	0000_0400h
24h	GPMI Integrated ECC Control Register Description (ECCCTRL_SET)	32	RW	0000_0400h
28h	GPMI Integrated ECC Control Register Description (ECCCTRL_CLR)	32	RW	0000_0400h
2Ch	GPMI Integrated ECC Control Register Description (ECCCTRL_TOG)	32	RW	0000_0400h
30h	GPMI Integrated ECC Transfer Count Register Description (ECCCOUNT)	32	RW	0000_0000h
40h	GPMI Payload Address Register Description (PAYLOAD)	32	RW	0000_0000h
50h	GPMI Auxiliary Address Register Description (AUXILIARY)	32	RW	0000_0000h
60h	GPMI Control Register 1 Description (CTRL1)	32	RW	0004_0004h
64h	GPMI Control Register 1 Description (CTRL1_SET)	32	RW	0004_0004h
68h	GPMI Control Register 1 Description (CTRL1_CLR)	32	RW	0004_0004h
6Ch	GPMI Control Register 1 Description (CTRL1_TOG)	32	RW	0004_0004h
70h	GPMI Timing Register 0 Description (TIMING0)	32	RW	0001_0203h
80h	GPMI Timing Register 1 Description (TIMING1)	32	RW	0000_0000h
90h	GPMI Timing Register 2 Description (TIMING2)	32	RW	2302_3336h
A0h	GPMI DMA Data Transfer Register Description (DATA)	32	RW	0000_0000h
B0h	GPMI Status Register Description (STAT)	32	RO	0000_0005h
C0h	GPMI Debug Information Register Description (DEBUG)	32	RO	0000_0000h
D0h	GPMI Version Register Description (VERSION)	32	RO	0502_0000h
E0h	GPMI Debug2 Information Register Description (DEBUG2)	32	RW	0000_F100h
F0h	GPMI Debug3 Information Register Description (DEBUG3)	32	RO	0000_0000h
100h	GPMI Double Rate Read DLL Control Register Description (READ_DDR_DLL_CTRL)	32	RW	0000_0038h
110h	GPMI Double Rate Write DLL Control Register Description (WRITE_DDR_DLL_CTRL)	32	RW	0000_0038h
120h	GPMI Double Rate Read DLL Status Register Description (READ_DDR_DLL_STS)	32	RO	0000_0000h
130h	GPMI Double Rate Write DLL Status Register Description (WRITE_DDR_DLL_STS)	32	RO	0000_0000h

9.7.4.2 GPMI Control Register 0 Description (CTRL0)

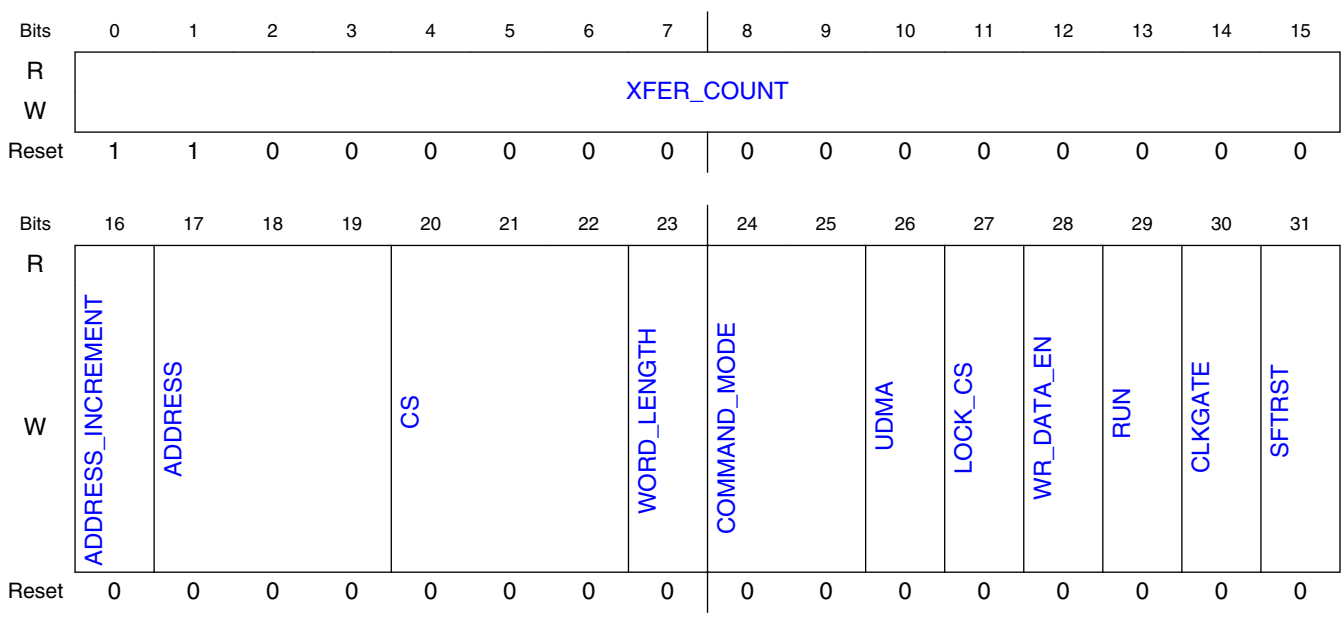
9.7.4.2.1 Offset

Register	Offset	Description
CTRL0	0h	GPMI Control Register 0 Description
CTRL0_SET	4h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL0
CTRL0_CLR	8h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL0
CTRL0_TOG	Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL0

9.7.4.2.2 Function

The GPMI control register 0 specifies the GPMI transaction to perform for the current command chain item.

9.7.4.2.3 Diagram



9.7.4.2.4 Fields

Field	Function
0-15 XFER_COUNT	XFER_COUNT Number of bytes to transfer for this command. A value of zero will transfer 64K bytes.
16 ADDRESS_INCREMENT	ADDRESS_INCREMENT In ATA mode, the address will increment with each cycle. In NAND mode, the address will increment once, after the first cycle (going from CLE to ALE). DISABLED = 0x0 Address does not increment. ENABLED = 0x1 Increment address.
17-19 ADDRESS	ADDRESS Specifies the three address lines for ATA mode. In NAND mode, use A0 for CLE and A1 for ALE. NAND_DATA = 0x0 In NAND mode, this address is used to read and write data bytes. NAND_CLE = 0x1 In NAND mode, this address is used to write command bytes. NAND_ALE = 0x2 In NAND mode, this address is used to write address bytes.
20-22 CS	CS Selects which chip select is active for this command. For ATA WAIT_FOR_READY command, this must be set to b01.
23 WORD_LENGTH	WORD_LENGTH This bit should only be changed when RUN==0. Reserve = 0x0 Reserved. 8_BIT = 0x1 8-bit Data Bus mode. 0b - Reserved. 1b - 8-bit Data Bus mode.
24-25 COMMAND_MODE	COMMAND_MODE WRITE = 0x0 Write mode. READ = 0x1 Read mode. READ_AND_COMPARE = 0x2 Read and Compare mode (setting sense flop). WAIT_FOR_READY = 0x3 Wait for Ready mode. For ATA WAIT_FOR_READY command set CS=01. 00b - Write mode. 01b - Read Mode. 10b - Read and Compare Mode (setting sense flop). 11b - Wait for Ready.
26 UDMA	UDMA DISABLED = 0x0 Use ATA-PIO mode on the external bus. ENABLED = 0x1 Use ATA-Ultra DMA mode on the external bus. 0b - Use ATA-PIO mode on the external bus. 1b - Use ATA-Ultra DMA mode on the external bus.
27 LOCK_CS	LOCK_CS For ATA/NAND mode: 0= Deassert chip select (CS) after RUN is complete. 1= Continue to assert chip select (CS) after RUN is complete. For Camera Mode: 0= Dont wait for VSYNC rising edge before capturing data. 1= Wait for VSYNC rising edge before capturing data (Camera mode only).

Table continues on the next page...

Field	Function
	DISABLED = 0x0 Deassert chip select (CS) after RUN is complete. ENABLED = 0x1 Continue to assert chip select (CS) after RUN is complete.
28 WR_DATA_EN	WR_DATA_EN When operation is writing at toggle/toggle2.0 mode, this bit should be set to 0x1.
29 RUN	RUN The GPMI is busy running a command whenever this bit is set to '1'. The GPMI is idle whenever this bit set to zero. This can be set to one by a CPU write. In addition, the DMA sets this bit each time a DMA command has finished its PIO transfer phase. IDLE = 0x0 The GPMI is idle. BUSY = 0x1 The GPMI is busy running a command.
30 CLKGATE	CLKGATE Set this bit zero for normal operation. Setting this bit to one (default), gates all of the block level clocks off for miniminizing AC energy consumption. RUN = 0x0 Allow GPMI to operate normally. NO_CLKS = 0x1 Do not clock GPMI gates in order to minimize power consumption.
31 SFTRST	SFTRST Set to zero for normal operation. When this bit is set to one (default), then the entire block is held in its reset state. This will not work if the CLKGATE bit is already set to '1'. CLKGATE must be cleared to '0' before issuing a soft reset. Also the GPMICLK must be running for this to work properly. RUN = 0x0 Allow GPMI to operate normally. RESET = 0x1 Hold GPMI in reset.

9.7.4.3 GPMI Compare Register Description (COMPARE)

9.7.4.3.1 Offset

Register	Offset
COMPARE	10h

9.7.4.3.2 Function

The GPMI compare register specifies the expect data and the xor mask for comparing to the status values read from the device. This register is used by the Read and Compare command.

GPMI_COMPARE 0x010

9.7.4.3.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	REFERENCE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.7.4.3.4 Fields

Field	Function
0-15 REFERENCE	REFERENCE 16-bit value which is XORed with data read from the NAND device.
16-31 MASK	MASK 16-bit mask which is applied after the read data is XORed with the REFERENCE bit field.

9.7.4.4 GPMI Integrated ECC Control Register Description (ECCCTRL)

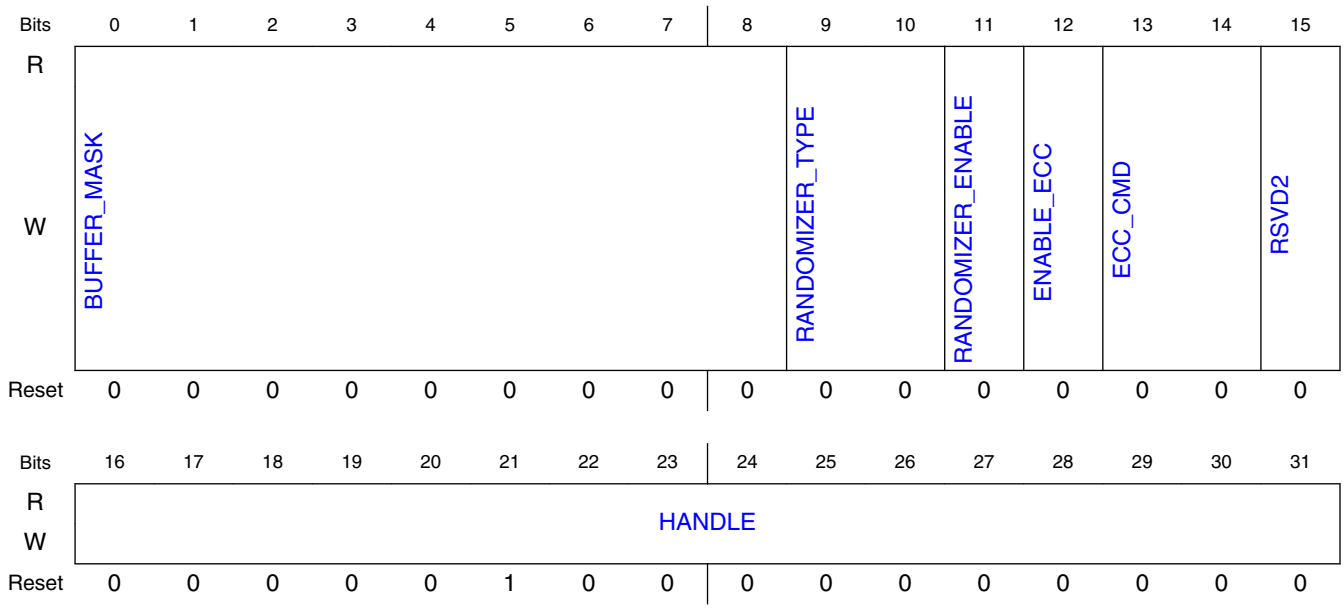
9.7.4.4.1 Offset

Register	Offset	Description
ECCCTRL	20h	GPMI Integrated ECC Control Register Description
ECCCTRL_SET	24h	Writing a 1 to a bit in this register sets the corresponding bit in ECCCTRL
ECCCTRL_CLR	28h	Writing a 1 to a bit in this register clears the corresponding bit in ECCCTRL
ECCCTRL_TOG	2Ch	Writing a 1 to a bit in this register toggles the corresponding bit in ECCCTRL

9.7.4.4.2 Function

The GPMI ECC control register handles configuration of the integrated ECC / Randomizer accelerator.

9.7.4.4.3 Diagram



9.7.4.4.4 Fields

Field	Function
0-8 BUFFER_MASK	BUFFER_MASK ECC buffer information. The BCH error correction only allows two configurations of the buffer mask - software may either read just the first block on the flash page or the entire flash page. Write operations must be for the entire flash page. Invalid buffer mask values will cause the DMA descriptor command to be terminated. BCH_AUXONLY = 0x100 Set to request transfer from only the auxiliary buffer (block 0 on flash). BCH_PAGE = 0x1FF Set to request transfer to/from the entire page.
9-10 RANDOMIZER_TYPE	RANDOMIZER_TYPE Set randomizer type 00b - Type 0 01b - Type 1 10b - Type 2
11 RANDOMIZER_ENABLE	RANDOMIZER_ENABLE Enable randomizer function. If this bit is set to enable, ENABLE_ECC should be also enable. 0b - disable 1b - enable
12 ENABLE_ECC	ENABLE_ECC Enable ECC processing of GPMI transfers. ENABLE = 0x1 Use integrated ECC for read and write transfers. DISABLE = 0x0 Integrated ECC remains in idle.

Table continues on the next page...

Field	Function
13-14 ECC_CMD	ECC_CMD ECC Command information. DECODE = 0x0 Decode. ENCODE = 0x1 Encode. RESERVE2 = 0x2 Reserved. RESERVE3 = 0x3 Reserved.
15 RSVD2	RSVD2 Always write zeroes to this bit field.
16-31 HANDLE	HANDLE This is a register available to software to attach an identifier to a transaction in progress. This handle will be available from the ECC register space when the completion interrupt occurs.

9.7.4.5 GPMI Integrated ECC Transfer Count Register Description (ECCCOUNT)

9.7.4.5.1 Offset

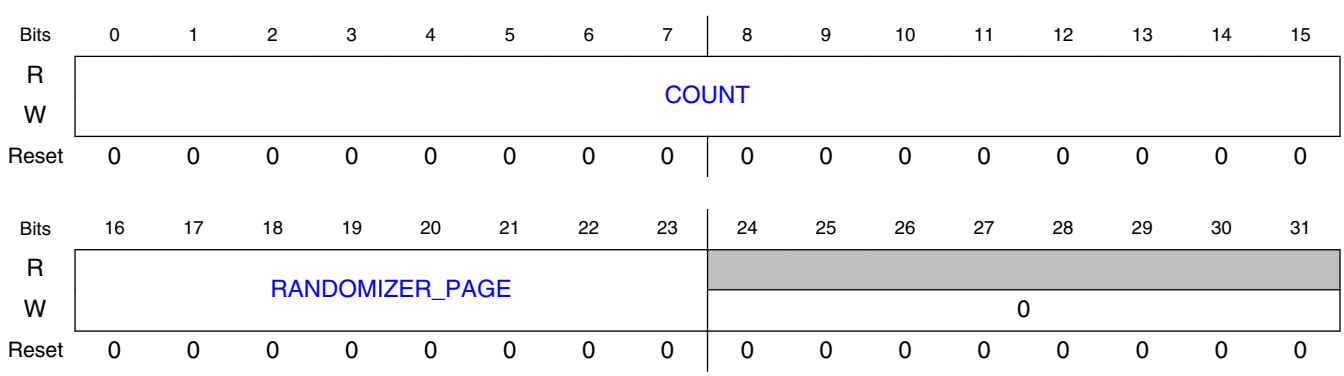
Register	Offset
ECCCOUNT	30h

9.7.4.5.2 Function

The GPMI ECC Transfer Count Register contains the count of bytes that flow through the ECC / Randomizer subsystem.

GPMI_ECCCOUNT 0x030

9.7.4.5.3 Diagram



9.7.4.5.4 Fields

Field	Function
0-15 COUNT	COUNT Number of bytes to pass through ECC. This is the GPMI transfer count plus the syndrome count that will be inserted into the stream by the ECC. In DMA2ECC_MODE this count must match the GPMI_CTRL0_XFER_COUNT. A value of zero will transfer 64K words.
16-23 RANDOMIZER_PAGE	RANDOMIZER_PAGE Set NAND page number needed to be randomized. The value is between 0-255
24-31 —	- Always write zeroes to this bit field.

9.7.4.6 GPMI Payload Address Register Description (PAYLOAD)

9.7.4.6.1 Offset

Register	Offset
PAYLOAD	40h

9.7.4.6.2 Function

The GPMI payload address register specifies the location of the data buffers in system memory. This value must be word aligned.

GPMI_PAYLOAD 0x040

9.7.4.6.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	RSVD0		ADDRESS													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	ADDRESS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.7.4.6.4 Fields

Field	Function
0-1 RSVD0	RSVD0 Always write zeroes to this bit field.
2-31 ADDRESS	ADDRESS Pointer to an array of one or more 512 byte payload buffers.

9.7.4.7 GPMI Auxiliary Address Register Description (AUXILIARY)

9.7.4.7.1 Offset

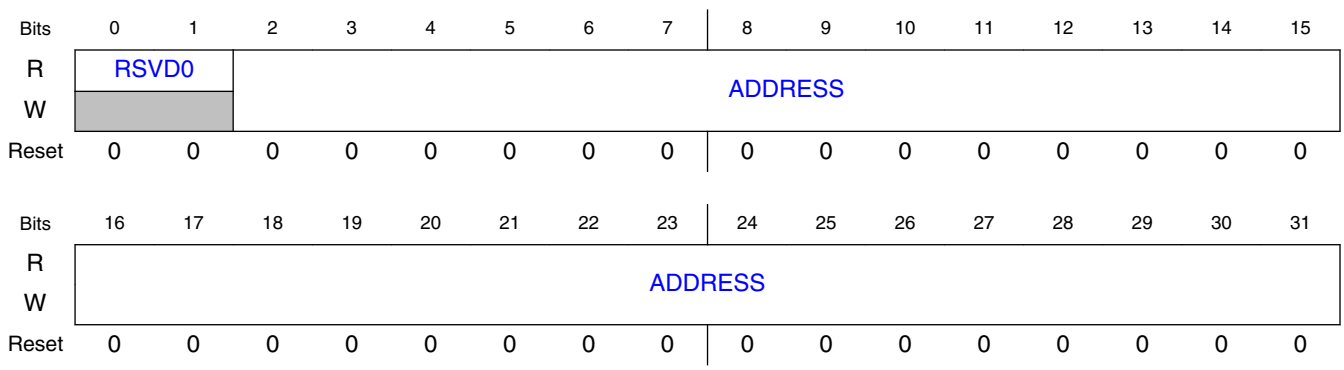
Register	Offset
AUXILIARY	50h

9.7.4.7.2 Function

The GPMI auxiliary address register specifies the location of the auxiliary buffers in system memory. This value must be word aligned.

GPMI_AUXILIARY 0x050

9.7.4.7.3 Diagram



9.7.4.7.4 Fields

Field	Function
0-1 RSVD0	RSVD0 Always write zeroes to this bit field.
2-31 ADDRESS	ADDRESS Pointer to ECC control structure and meta-data storage.

9.7.4.8 GPMI Control Register 1 Description (CTRL1)

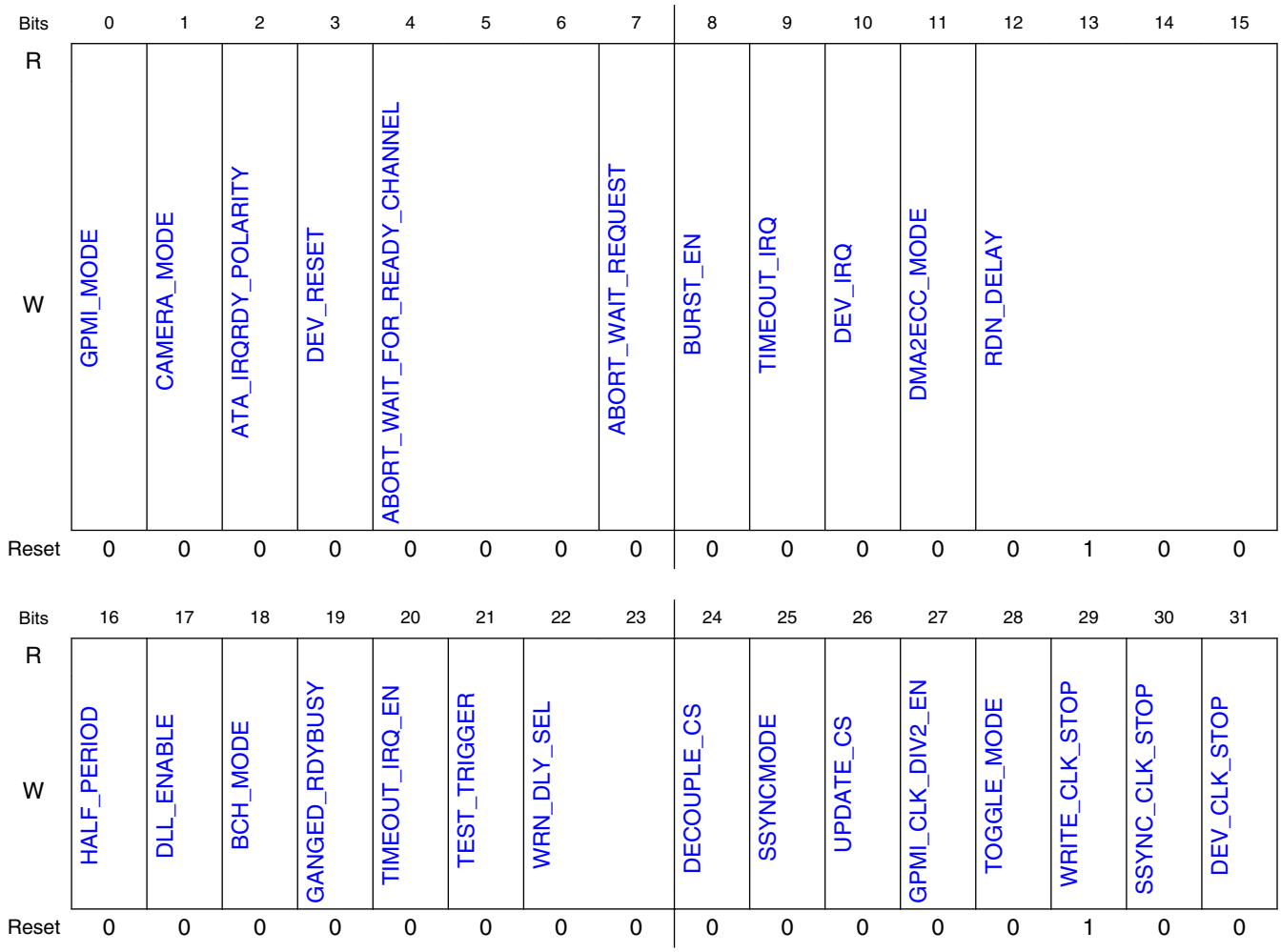
9.7.4.8.1 Offset

Register	Offset	Description
CTRL1	60h	GPMI Control Register 1 Description
CTRL1_SET	64h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL1
CTRL1_CLR	68h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL1
CTRL1_TOG	6Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL1

9.7.4.8.2 Function

The GPMI control register 1 specifies additional control fields that are not used on a per-transaction basis.

9.7.4.8.3 Diagram



9.7.4.8.4 Fields

Field	Function
0 GPMI_MODE	GPMI_MODE ATA mode is only supported on channel zero. If ATA mode is selected, then only channel three is available for NAND use. NAND = 0x0 NAND mode. ATA = 0x1 ATA mode. 0b - NAND mode. 1b - ATA mode.
1 CAMERA_MODE	CAMERA_MODE When set to 1 and ATA UDMA is enabled the UDMA interface becomes a camera interface.
2	ATA_IRQRDY_POLARITY

Table continues on the next page...

Field	Function
ATA_IRQRDY_POLARITY	For ATA MODE: Note NAND_RDY_BUSY[3:2] are not affected by this bit. ACTIVELOW = 0x0 ATA IORDY and IRQ are active low, or NAND_RDY_BUSY[1:0] are active low ready. ACTIVEHIGH = 0x1 ATA IORDY and IRQ are active high, or NAND_RDY_BUSY[1:0] are active high ready. 0 External ATA IORDY and IRQ are active low. 1 External ATA IORDY and IRQ are active high. For NAND MODE: 0b - External RDY_BUSY[1] and RDY_BUSY[0] pins are ready when low and busy when high. 1b - External RDY_BUSY[1] and RDY_BUSY[0] pins are ready when high and busy when low.
3 DEV_RESET	DEV_RESET ENABLED = 0x0 NANDF_WP_B(WPN) pin is held low (asserted). DISABLED = 0x1 NANDF_WP_B(WPN) pin is held high (de-asserted). 0b - NANDF_WP_B pin is held low (asserted). 1b - NANDF_WP_B pin is held high (de-asserted).
4-6 ABORT_WAIT_FOR_READY_CHANNEL	ABORT_WAIT_FOR_READY_CHANNEL Abort a wait for ready command on selected channel. Set the ABORT_WAIT_REQUEST to kick of operation.
7 ABORT_WAIT_REQUEST	ABORT_WAIT_REQUEST Request to abort "wait for ready" command on channel indicated by ABORT_WAIT_FOR_READY_CHANNEL. Hardware will clear this bit when abort is done.
8 BURST_EN	BURST_EN When set to 1 each DMA request will generate a 4-transfer burst on the APB bus.
9 TIMEOUT_IRQ	TIMEOUT_IRQ This bit is set when a timeout occurs using the Device_Busy_Timeout value. Write 0 to clear.
10 DEV_IRQ	DEV_IRQ This bit is set when an Interrupt is received from the ATA device. Write 0 to clear.
11 DMA2ECC_MODE	DMA2ECC_MODE This is mainly for testing HWECC without involving the Nand device. Setting this bit will cause DMA write data to redirected to HWECC module (instead of Nand Device) for encoding or decoding.
12-15 RDN_DELAY	RDN_DELAY This variable is a factor in the calculated delay to apply to the internal read strobe for correct read data sampling. The applied delay (AD) is between 0 and 1.875 times the reference period (RP). RP is one half of the GPMI clock period if HALF_PERIOD=1 otherwise it is the full GPMI clock period. The equation is: AD = RDN_DELAY x 0.125 x RP. This value must not exceed 16ns. This variable is used to achieve faster NAND access. For example if the Read Strobe is asserted from time 0 to 13ns but the read access time is 20ns, then choose AD=12ns will cause the data to be sampled at time 25ns (13+12) giving a 5ns data setup time. If RP=13ns then RDN_DELAY = 12/(0.125 x 13ns) = 7.38 (0111b). DLL_ENABLE must be zero while changing this field.

Table continues on the next page...

Field	Function
16 HALF_PERIOD	HALF_PERIOD Set this bit to 1 if the GPMI clock period is greater than 16ns for proper DLL operation. DLL_ENABLE must be zero while changing this field.
17 DLL_ENABLE	DLL_ENABLE Set this bit to 1 to enable the GPMI DLL. This is required for fast NAND reads (above 30 MHz read strobe). After setting this bit, wait 64 GPMI clock cycles for the DLL to lock before performing a NAND read.
18 BCH_MODE	BCH_MODE This bit selects which error correction unit will access GPMI. This bit must always be set to '1', since only the BCH unit is available in this design.
19 GANGED_RDYBUSY	GANGED_RDYBUSY Set this bit to 1 will force all Nand RDY_BUSY inputs to be sourced from (tied to) RDY_BUSY0. This will free up all, except one, RDY_BUSY input pins.
20 TIMEOUT_IRQ_EN	TIMEOUT_IRQ_EN Setting this bit to '1' will enable timeout IRQ for transfers in ATA mode only, and for WAIT_FOR_READY commands in both ATA and Nand mode. The Device_Busy_Timeout value is used for this timeout.
21 TEST_TRIGGER	TEST_TRIGGER Test Trigger Enable 0b - Disable 1b - Enable
22-23 WRN_DLY_SEL	WRN_DLY_SEL Since the GPMI write strobe (WRN) is a fast clock pin, the delay on this signal can be programmed to match the load on this pin. 0 = ~2ns; 1 = ~4ns; 2 = ~6ns; 3 = no delay.
24 DECOUPLE_CS	DECOUPLE_CS Decouple Chip Select from DMA Channel. Setting this bit to 1 will allow a DMA channel to specify any value in the CTRL0_CS register field. Software can use one DMA channel to access all 8 Nand devices.
25 SSYNCMODE	SSYNCMODE source synchronouse mode 1 or asynchrous mode 0. ASYNC = 0x0 Asynchronous mode. SSYNC = 0x1 Source Synchronous mode.
26 UPDATE_CS	UPDATE_CS force the CS value is be updated to external chip select pin, even GPMI is idle.
27 GPMI_CLK_DIV2_EN	GPMI_CLK_DIV2_EN This bit should be reset to 0 in asynchronous mode. The frequence ratio of (device clock : ccm gpmi clock) will be (1 : 1). This bit should be set to 1, in source synchronous mode or toggle mode. The frequence raito of (device clock : ccm gpmi clock) will be (1 : 2). enable the gpmi clk divider. 0b - internal factor-2 clock divider is disabled 1b - internal factor-2 clock divider is enabled.
28	TOGGLE_MODE enable samsung toggle mode.

Table continues on the next page...

Field	Function
TOGGLE_MODE	
29 WRITE_CLK_STOP	WRITE_CLK_STOP In onfi source synchronous mode, host may save power during the data write cycles by holding the CLK signal high (i.e. stopping the CLK). The host may only stop the CLK during data write, by setting this bit to 1, if the device supports this feature as indicated in the parameter page.
30 SSYNC_CLK_STOP	SSYNC_CLK_STOP set this bit to 1 will stop the source synchronous mode clk.
31 DEV_CLK_STOP	DEV_CLK_STOP set this bit to 1 will stop gpmi io working clk.

9.7.4.9 GPMI Timing Register 0 Description (TIMING0)

9.7.4.9.1 Offset

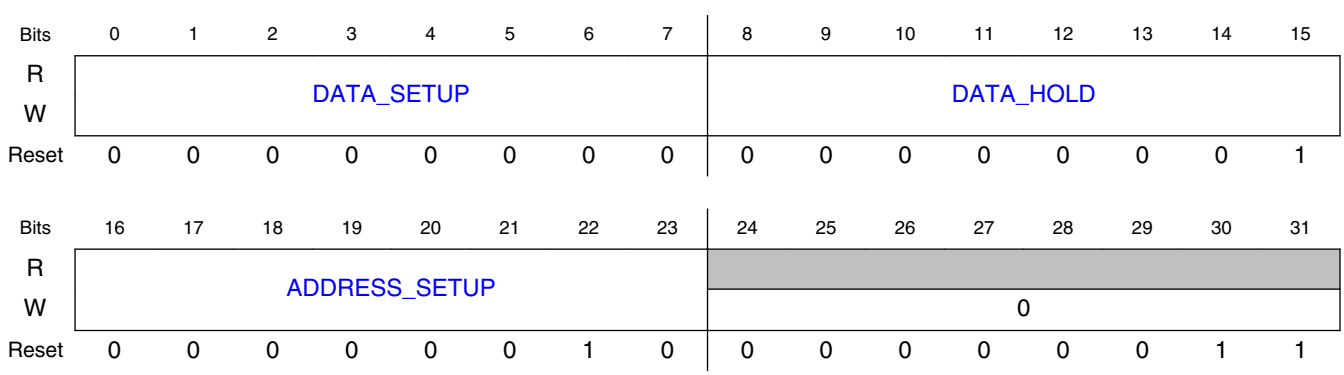
Register	Offset
TIMING0	70h

9.7.4.9.2 Function

The GPMI timing register 0 specifies the timing parameters that are used by the cycle state machine to guarantee the various setup, hold and cycle times for the external media type.

GPMI_TIMING0 0x070

9.7.4.9.3 Diagram



9.7.4.9.4 Fields

Field	Function
0-7 DATA_SETUP	DATA_SETUP Data bus setup time in GPMICLK cycles. Also the time that the data strobe is asserted in a cycle. This value must be greater than 2 for ATA devices that use IORDY to extend transfer cycles. A value of zero is interpreted as 256. For ATA PIO modes this is known in the ATA7 specification as ""DIOR-/DIOW-"
8-15 DATA_HOLD	DATA_HOLD Data bus hold time in GPMICLK cycles. Also the time that the data strobe is de-asserted in a cycle. A value of zero is interpreted as 256. For ATA PIO modes this is known in the ATA7 specification as "DIOR-/DIOW- recovery time"
16-23 ADDRESS_SETUP	ADDRESS_SETUP Number of GPMICLK cycles that the CE/ADDR signals are active before a strobe is asserted. A value of zero is interpreted as 0. For ATA PIO modes this is known in the ATA7 specification as "Address valid to DIOR-/DIOW- setup"
24-31 RSVD1	RSVD1 Always write zeroes to this bit field.

9.7.4.10 GPMI Timing Register 1 Description (TIMING1)

9.7.4.10.1 Offset

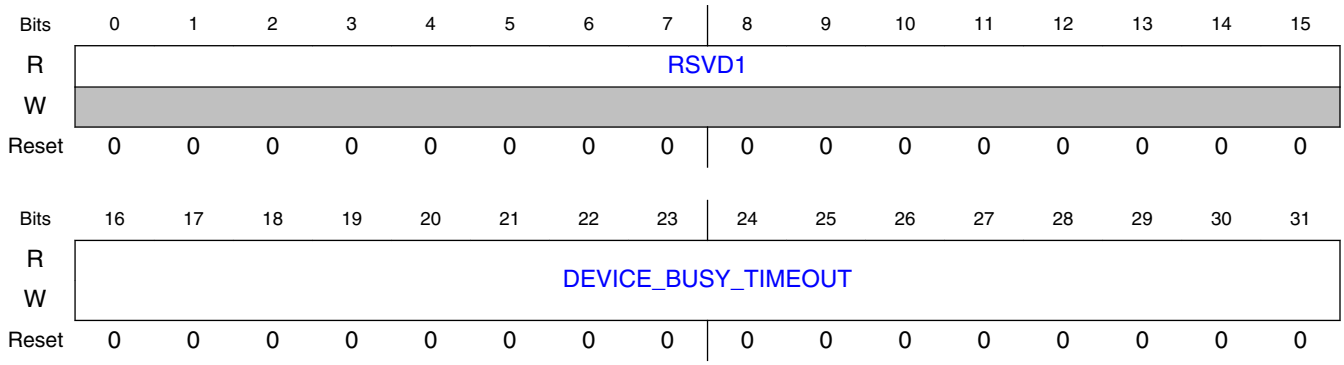
Register	Offset
TIMING1	80h

9.7.4.10.2 Function

The GPMI timing register 1 specifies the timeouts used when monitoring the NAND READY pin or the ATA IRQ and IOWAIT signals.

GPMI_TIMING1 0x080

9.7.4.10.3 Diagram



9.7.4.10.4 Fields

Field	Function
0-15 RSVD1	RSVD1 Always write zeroes to this bit field.
16-31 DEVICE_BUSY_TIMEOUT	DEVICE_BUSY_TIMEOUT Timeout waiting for NAND Ready/Busy or ATA IRQ. Used in WAIT_FOR_READY mode. This value is the number of GPMI_CLK cycles multiplied by 4096.

9.7.4.11 GPMI Timing Register 2 Description (TIMING2)

9.7.4.11.1 Offset

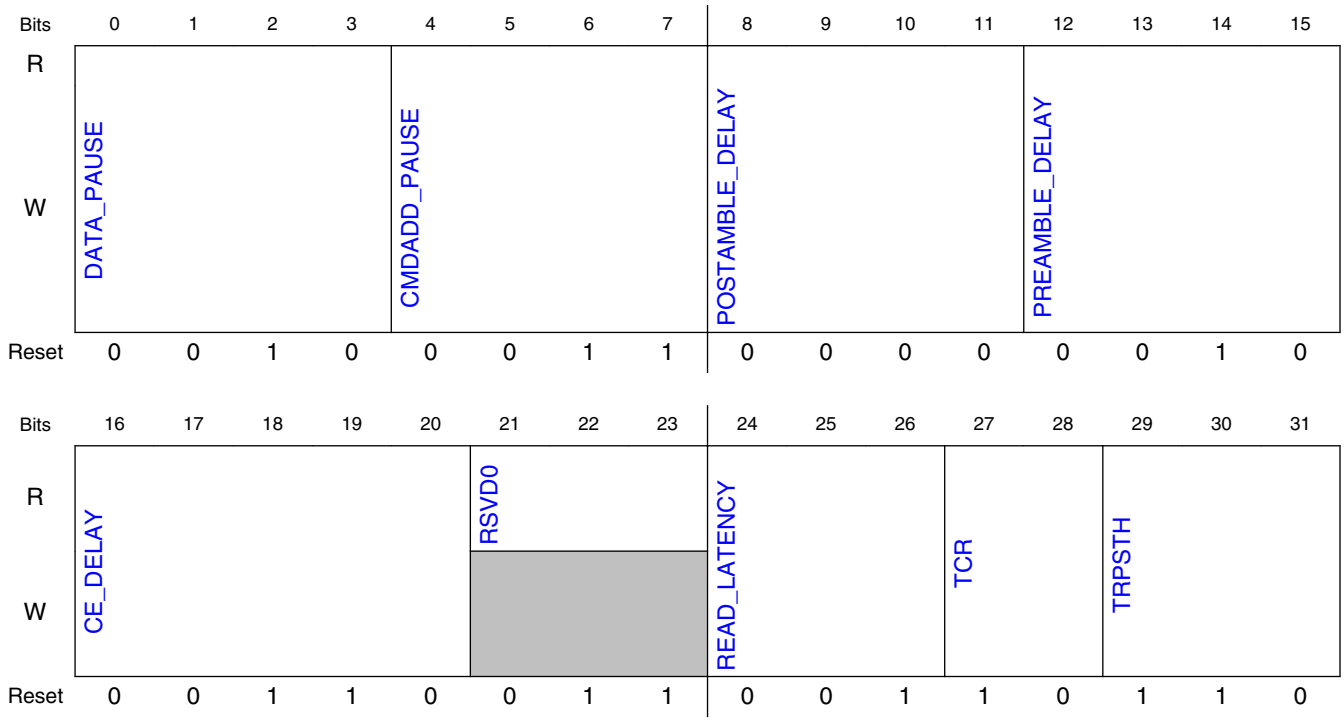
Register	Offset
TIMING2	90h

9.7.4.11.2 Function

The GPMI timing register 2 specifies the double data rate timing parameters that are used by the cycle state machine to guarantee the various cs delay, pre-amble delay, post-amble delay, command/address delay, data delay, TCR, TRPSTH, and read latency cycle times for the external media type.

GPMI_TIMING2 0x090

9.7.4.11.3 Diagram



9.7.4.11.4 Fields

Field	Function
0-3 DATA_PAUSE	DATA_PAUSE GPMI delay time from data pause to data resume in GPMICLK cycles. A value of zero is interpreted as 16.
4-7 CMDADD_PAUSE	CMDADD_PAUSE GPMI delay time from command or addres pause to command or address resume in GPMICLK cycles. A value of zero is interpreted as 16.
8-11 POSTAMBLE_DELAY	POSTAMBLE_DELAY GPMI post-amble delay in GPMICLK cycles. A value of zero is interpreted as 16.
12-15 PREAMBLE_DELAY	PREAMBLE_DELAY GPMI pre-amble delay in GPMICLK cycles. A value of zero is interpreted as 16.
16-20 CE_DELAY	CE_DELAY GPMI dealy from CEn assert to W/Rn changing edge. value of zero is interpreted as 32.
21-23 RSVD0	RSVD0 Always write zeroes to this bit field.
24-26	READ_LATENCY This field is for double data rate read latency configuration.

Table continues on the next page...

Field	Function
READ_LATENCY	others READ LATENCY is 3 000b - READ LATENCY is 0 001b - READ LATENCY is 1 010b - READ LATENCY is 2 011b - READ LATENCY is 3 100b - READ LATENCY is 4 101b - READ LATENCY is 5
27-28 TCR	TCR Only for Toggle NAND timing control delay (TCR+1) GPMICLK cycles for CEn_B low to RE_B low, 0 is less than or equal to TCR, which is less than the PREAMBLE_DELAY
29-31 TRPSTH	TRPSTH Only for Toggle NAND timing control delay TRPSTH GPMICLK cycles for CEn_B high to RE_B high, A value of zero is interpreted as 8

9.7.4.12 GPMI DMA Data Transfer Register Description (DATA)

9.7.4.12.1 Offset

Register	Offset
DATA	A0h

9.7.4.12.2 Function

The GPMI DMA data transfer register is used by the DMA to read or write data to or from the ATA/NAND control state machine.

GPMI_DATA 0x0A0

9.7.4.12.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.7.4.12.4 Fields

Field	Function
0-31	DATA
DATA	In 8-bit mode, one, two, three or four bytes can can be accessed to send the same number of bus cycles.

9.7.4.13 GPMI Status Register Description (STAT)

9.7.4.13.1 Offset

Register	Offset
STAT	B0h

9.7.4.13.2 Function

The GPMI control and status register provides a read back path for various operational states of the GPMI controller.

GPMI_STAT 0x0B0

9.7.4.13.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	PRESENT	FIFO_FULL	FIFO_EMPTY	INVALID_BUFFER_MASK	ATA_IRQ	RSVD1			DEV0_ERROR	DEV1_ERROR	DEV2_ERROR	DEV3_ERROR	DEV4_ERROR	DEV5_ERROR	DEV6_ERROR	DEV7_ERROR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	RDY_TIMEOUT								READY_BUSY							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

9.7.4.13.4 Fields

Field	Function
0 PRESENT	PRESENT UNAVAILABLE = 0x0 GPMI is not present in this product. AVAILABLE = 0x1 GPMI is present in this product. 0b - GPMI is not present in this product. 1b - GPMI is present is in this product.
1 FIFO_FULL	FIFO_FULL NOT_FULL = 0x0 FIFO is not full. FULL = 0x1 FIFO is full. 0b - FIFO is not full. 1b - FIFO is full.
2 FIFO_EMPTY	FIFO_EMPTY NOT_EMPTY = 0x0 FIFO is not empty. EMPTY = 0x1 FIFO is empty. 0b - FIFO is not empty. 1b - FIFO is empty.
3 INVALID_BUFFER_MASK	INVALID_BUFFER_MASK Buffer Mask Validity bit. 0b - ECC Buffer Mask is not invalid. 1b - ECC Buffer Mask is invalid.
4 ATA_IRQ	ATA_IRQ Status of the ATA_IRQ input pin.
5-7 RSVD1	RSVD1 Always write zeroes to this bit field.
8 DEV0_ERROR	DEV0_ERROR DMA channel 0 (Timeout or compare failure, depending on COMMAND_MODE). 0b - No error condition present on ATA/NAND Device accessed by DMA channel 0. 1b - An Error has occurred on ATA/NAND Device accessed by
9 DEV1_ERROR	DEV1_ERROR DMA channel 1 (Timeout or compare failure, depending on COMMAND_MODE). 0b - No error condition present on ATA/NAND Device accessed by DMA channel 1. 1b - An Error has occurred on ATA/NAND Device accessed by
10 DEV2_ERROR	DEV2_ERROR DMA channel 2 (Timeout or compare failure, depending on COMMAND_MODE). 0b - No error condition present on ATA/NAND Device accessed by DMA channel 2. 1b - An Error has occurred on ATA/NAND Device accessed by
11 DEV3_ERROR	DEV3_ERROR DMA channel 3 (Timeout or compare failure, depending on COMMAND_MODE). 0b - No error condition present on ATA/NAND Device accessed by DMA channel 3. 1b - An Error has occurred on ATA/NAND Device accessed by

Table continues on the next page...

Field	Function
12 DEV4_ERROR	DEV4_ERROR DMA channel 4 (Timeout or compare failure, depending on COMMAND_MODE). 0b - No error condition present on ATA/NAND Device accessed by DMA channel 4. 1b - An Error has occurred on ATA/NAND Device accessed by
13 DEV5_ERROR	DEV5_ERROR DMA channel 5 (Timeout or compare failure, depending on COMMAND_MODE). 0b - No error condition present on ATA/NAND Device accessed by DMA channel 5. 1b - An Error has occurred on ATA/NAND Device accessed by
14 DEV6_ERROR	DEV6_ERROR DMA channel 6 (Timeout or compare failure, depending on COMMAND_MODE). 0b - No error condition present on ATA/NAND Device accessed by DMA channel 6. 1b - An Error has occurred on ATA/NAND Device accessed by
15 DEV7_ERROR	DEV7_ERROR DMA channel 7 (Timeout or compare failure, depending on COMMAND_MODE). 0b - No error condition present on ATA/NAND Device accessed by DMA channel 7. 1b - An Error has occurred on ATA/NAND Device accessed by
16-23 RDY_TIMEOUT	RDY_TIMEOUT State of the RDY/BUSY Timeout Flags. When any bit is set to '1' in this field, it indicates that a time out has occurred while waiting for the ready state of the requested NAND device. Multiple bits may be set simultaneously. When GPMI_CTRL1_DECOUPLE_CS = 0, RDY_TIMEOUT[n] is associated with the NAND device on chip_select[n]. When GPMI_CTRL1_DECOUPLE_CS = 1, these flags become associated to a DMA channel instead of a NAND device. For example if DMA channel 6 sends a WAIT_FOR_READY command for NAND Device 2, and a timeout occured on READY_BUSY2, then READY_TIMEOUT[6] will be set instead of READY_TIMEOUT[2].
24-31 READY_BUSY	READY_BUSY Read-only view of NAND Ready_Busy Input pins.

9.7.4.14 GPMI Debug Information Register Description (DEBUG)

9.7.4.14.1 Offset

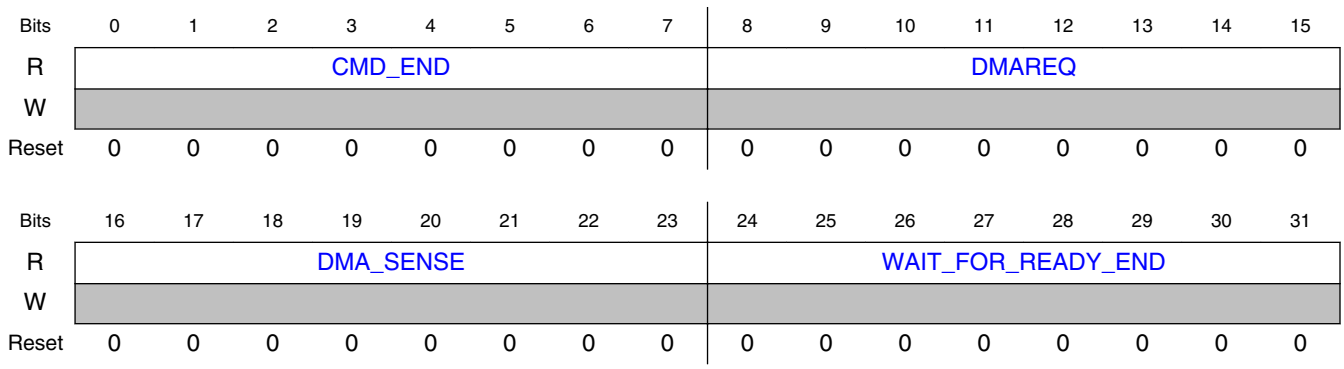
Register	Offset
DEBUG	C0h

9.7.4.14.2 Function

The GPMI debug information register provides a read back path for diagnostics to determine the current operating state of the GPMI controller.

GPMI_DEBUG 0x0C0

9.7.4.14.3 Diagram



9.7.4.14.4 Fields

Field	Function
0-7 CMD_END	CMD_END Read Only view of the Command End toggle signals to DMA. One per channel
8-15 DMAREQ	DMAREQ Read-only view of DMA request line for 8 DMA channels. A toggle on any bit position indicates a DMA request for the corresponding channel.
16-23 DMA_SENSE	DMA_SENSE Read-only view of sense state of the 8 DMA channels. A value of "1" in any bit position indicates that a read and compare command failed or a timeout occured for the corresponding channel.
24-31 WAIT_FOR_READY_END	WAIT_FOR_READY_END Read Only view of the Wait_For_Ready End toggle signals to DMA. One per channel

9.7.4.15 GPMI Version Register Description (VERSION)

9.7.4.15.1 Offset

Register	Offset
VERSION	D0h

9.7.4.15.2 Function

This register reflects the version number for the GPMI.

GPMI_VERSION 0x0D0

9.7.4.15.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	STEP															
W																
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	MINOR								MAJOR							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.7.4.15.4 Fields

Field	Function
0-15 STEP	STEP Fixed read-only value reflecting the stepping of the RTL version.
16-23 MINOR	MINOR Fixed read-only value reflecting the MINOR field of the RTL version.
24-31 MAJOR	MAJOR Fixed read-only value reflecting the MAJOR field of the RTL version.

9.7.4.16 GPMI Debug2 Information Register Description (DEBUG2)

9.7.4.16.1 Offset

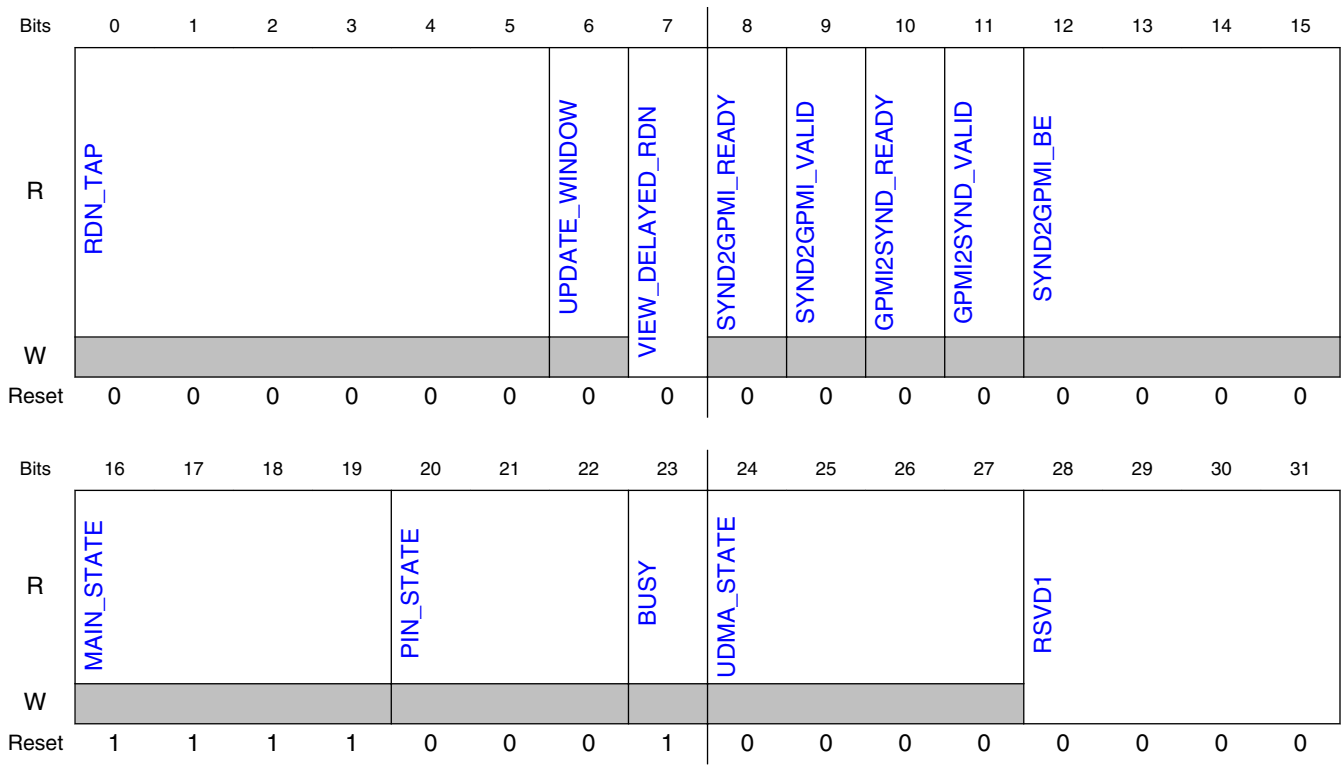
Register	Offset
DEBUG2	E0h

9.7.4.16.2 Function

The GPMI Debug2 information register provides a read back path for diagnostics to determine the current operating state of the GPMI controller.

GPMI_DEBUG2 0x0E0

9.7.4.16.3 Diagram



9.7.4.16.4 Fields

Field	Function
0-5 RDN_TAP	RDN_TAP This is the DLL tap calculated by the DLL controller. The selects the amount of delay form the DLL chain.
6 UPDATE_WINDOW	UPDATE_WINDOW A 1 indicates that the DLL is busy generating the required delay.
7 VIEW_DELAYED_RDN	VIEW_DELAYED_RDN Set to a 1 to select the delayed feedback RE_B to drive the GPMI_ADDR[0] (Nand CLE) pin. For debug purposes, this will allow you see if DLL is functioning properly.
8	SYND2GPMI_READY Data handshake Input from BCH.

Table continues on the next page...

Field	Function
SYND2GPMI_READY	
9 SYND2GPMI_VALID	SYND2GPMI_VALID Data handshake Input from BCH.
10 GPMI2SYND_READY	GPMI2SYND_READY Data handshake output to BCH.
11 GPMI2SYND_VALID	GPMI2SYND_VALID Data handshake output to BCH.
12-15 SYND2GPMI_BE	SYND2GPMI_BE Data byte enable Input from BCH.
16-19 MAIN_STATE	MAIN_STATE parameter MSM_IDLE = 4'h0, MSM_BYTCNT = 4'h1, MSM_WAITFE = 4'h2, MSM_WAITFR = 4'h3, MSM_DMAREQ = 4'h4, MSM_DMAACK = 4'h5, MSM_WAITFF = 4'h6, MSM_LDFIFO = 4'h7, MSM_LDDMAR = 4'h8, MSM_RDCMP = 4'h9, MSM_DONE = 4'hA. MSM_IDLE = 0x0 MSM_BYTCNT = 0x1 MSM_WAITFE = 0x2 MSM_WAITFR = 0x3 MSM_DMAREQ = 0x4 MSM_DMAACK = 0x5 MSM_WAITFF = 0x6 MSM_LDFIFO = 0x7 MSM_LDDMAR = 0x8 MSM_RDCMP = 0x9 MSM_DONE = 0xA
20-22 PIN_STATE	PIN_STATE parameter PSM_IDLE = 3'h0, PSM_BYTCNT = 3'h1, PSM_ADDR = 3'h2, PSM_STALL = 3'h3, PSM_STROBE = 3'h4, PSM_ATARDY = 3'h5, PSM_DHOLD = 3'h6, PSM_DONE = 3'h7. PSM_IDLE = 0x0 PSM_BYTCNT = 0x1 PSM_ADDR = 0x2 PSM_STALL = 0x3 PSM_STROBE = 0x4 PSM_ATARDY = 0x5 PSM_DHOLD = 0x6 PSM_DONE = 0x7
23 BUSY	BUSY

Table continues on the next page...

Field	Function
	When asserted the GPMI is busy. Undefined results may occur if any registers are written when BUSY is asserted. DISABLED = 0x0 The GPMI is not busy. ENABLED = 0x1 The GPMI is busy.
24-27 UDMA_STATE	UDMA_STATE USM_IDLE = 4'h0, idle USM_DMARQ = 4'h1, DMA req USM_ACK = 4'h2, DMA ACK USM_FIFO_E = 4'h3, Fifo empty USM_WPAUSE = 4'h4, WR DMA Paused by device USM_TSTRB = 4'h5, Toggle HSTROBE USM_CAPTUR = 4'h6, Capture Stage, (data sampled with DSTROBE is valid) USM_DATOUT = 4'h7, Change Burst DATAOUT USM_CRC = 4'h8, Source CRC to Device USM_WAIT_R = 4'h9, Waiting for DDMARDY- USM_END = 4'ha; Negate DMAACK (end of DMA) USM_WAIT_S = 4'hb, Waiting for DSTROBE USM_RPAUSE = 4'hc, Rd DMA Paused by Host USM_RSTOP = 4'hd, Rd DMA Stopped by Host USM_WTERM = 4'he, Wr DMA Termination State USM_RTERM = 4'hf, Rd DMA Termination state
28-31 RSVD1	RSVD1 Always write zeroes to this bit field.

9.7.4.17 GPMI Debug3 Information Register Description (DEBUG3)

9.7.4.17.1 Offset

Register	Offset
DEBUG3	F0h

9.7.4.17.2 Function

The GPMI Debug3 information register provides a read back path for diagnostics to determine the current operating state of the GPMI controller.

GPMI_DEBUG3 0x0F0

9.7.4.17.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	DEV_WORD_CNTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	APB_WORD_CNTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.7.4.17.4 Fields

Field	Function
0-15 DEV_WORD_CNTR	DEV_WORD_CNTR Reflects the number of bytes remains to be transferred on the ATA/Nand bus.
16-31 APB_WORD_CNTR	APB_WORD_CNTR Reflects the number of bytes remains to be transferred on the APB bus.

9.7.4.18 GPMI Double Rate Read DLL Control Register Description (READ_DDR_DLL_CTRL)

9.7.4.18.1 Offset

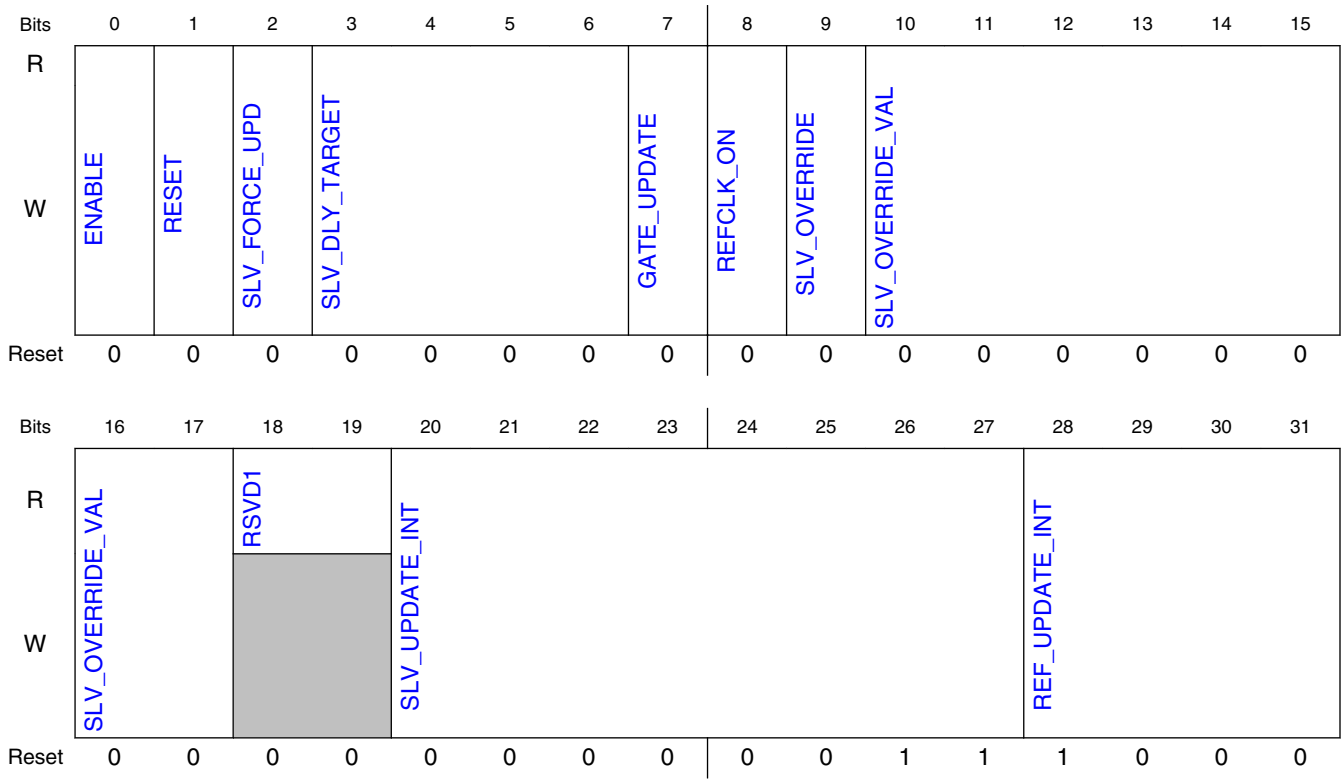
Register	Offset
READ_DDR_DLL_CTRL	100h

9.7.4.18.2 Function

GPMI DDR Read Delay Loop Lock Control Register. This register provides programmability in DDR mode for data input timing and data formats.

GPMI_READ_DDR_DLL_CTRL 0x100

9.7.4.18.3 Diagram



9.7.4.18.4 Fields

Field	Function
0 ENABLE	ENABLE Set this bit to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. Note that using the slave delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled.
1 RESET	RESET Setting this bit to 1 force a reset on DLL. This will cause the DLL to lose lock and re-calibrate to detect an GPMICLK half period phase shift. This signal is used by the DLL as edge-sensitive, so in order to create a subsequent reset, RESET must be taken low and then asserted again.
2 SLV_FORCE_UPD	SLV_FORCE_UPD Setting this bit to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line shall update automatically based on the SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered).
3-6 SLV_DLY_TARGET	SLV_DLY_TARGET The delay target for the read clock is can be programmed in 1/16th increments of an GPMICLK half-period. So the input read-clock can be delayed relative input data from (GPMICLK/2)/16 to GPMICLK/2.
7 GATE_UPDATE	GATE_UPDATE Setting this bit to 1, forces the slave delay line not update

Table continues on the next page...

Field	Function
8 REFCLK_ON	REFCLK_ON set this bit to 1 will turn on the reference clock
9 SLV_OVERRIDE	SLV_OVERRIDE Set this bit to 1 to Enable manual override for slave delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to tbe enabled using the ENABLE bit. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE=0
10-17 SLV_OVERRIDE_VAL	SLV_OVERRIDE_VAL When SLV_OVERRIDE=1 This field is used to select 1 of 256 physical taps manually. A value of 0 selects tap 1, and a value of 0x7f selects tap 256.
18-19 RSVD1	RSVD1 Reserved
20-27 SLV_UPDATE_INT	SLV_UPDATE_INT Setting a value greater than 0 in this field, shall over-ride the default slave delay-line update interval of 256 GPMICLK cycles. A value of 0 results in an update interval of 256 GPMICLK cycles (default setting). A value of 0x0f results in 15 cycles and so on. Note that software can always cause an update of the slave-delay line using the SLV_FORCE_UPDATE register. Note that the slave delay line will also update automatically when the reference DLL transitions to a locked state (from an un-locked state).
28-31 REF_UPDATE_INT	REF_UPDATE_INT This field allows the user to add additional delay cycles to the DLL control loop (reference delay line control). By default, the DLL control loop shall update every two GPMICLK cycles. Programming this field results in a DLL control loop update interval of (2 + REF_UPDATE_INT) * GPMICLK. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that may effect the delay (such as voltage and temperature)

9.7.4.19 GPMI Double Rate Write DLL Control Register Description (WRITE_DDR_DLL_CTRL)

9.7.4.19.1 Offset

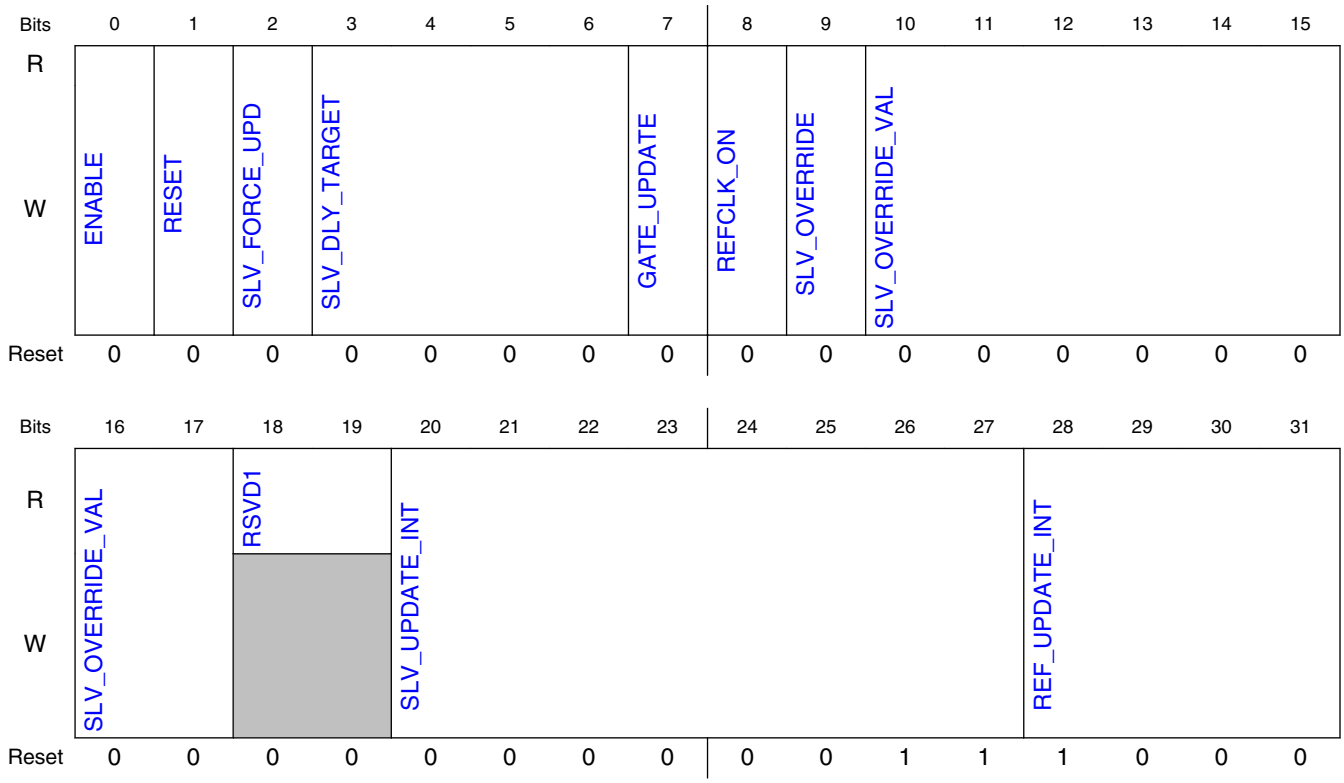
Register	Offset
WRITE_DDR_DLL_CTRL	110h

9.7.4.19.2 Function

GPMI DDR Write Delay Loop Lock Control Register. This register provides programmability in DDR mode for data output timing and data formats.

GPMI_WRITE_DDR_DLL_CTRL 0x110

9.7.4.19.3 Diagram



9.7.4.19.4 Fields

Field	Function
0 ENABLE	ENABLE Set this bit to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. Note that using the slave delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled.
1 RESET	RESET Setting this bit to 1 force a reset on DLL. This will cause the DLL to lose lock and re-calibrate to detect an GPMICLK half period phase shift. This signal is used by the DLL as edge-sensitive, so in order to create a subsequent reset, RESET must be taken low and then asserted again.
2 SLV_FORCE_UPD	SLV_FORCE_UPD Setting this bit to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line shall update automatically based on the SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered).
3-6 SLV_DLY_TARGET	SLV_DLY_TARGET The delay target for the read clock can be programmed in 1/16th increments of an GPMICLK half-period. So the input read-clock can be delayed relative input data from (GPMICLK/2)/16 to GPMICLK/2.
7 GATE_UPDATE	GATE_UPDATE Setting this bit to 1, forces the slave delay line not update

Table continues on the next page...

Field	Function
8 REFCLK_ON	REFCLK_ON set this bit to 1 will turn on the reference clock
9 SLV_OVERRIDE	SLV_OVERRIDE Set this bit to 1 to Enable manual override for slave delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to tbe enabled using the ENABLE bit. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE=0
10-17 SLV_OVERRIDE_VAL	SLV_OVERRIDE_VAL When SLV_OVERRIDE=1 This field is used to select 1 of 256 physical taps manually. A value of 0 selects tap 1, and a value of 0x7f selects tap 256.
18-19 RSVD1	RSVD1 Reserved
20-27 SLV_UPDATE_INT	SLV_UPDATE_INT Setting a value greater than 0 in this field, shall over-ride the default slave delay-line update interval of 256 GPMICLK cycles. A value of 0 results in an update interval of 256 GPMICLK cycles (default setting). A value of 0x0f results in 15 cycles and so on. Note that software can always cause an update of the slave-delay line using the SLV_FORCE_UPDATE register. Note that the slave delay line will also update automatically when the reference DLL transitions to a locked state (from an un-locked state).
28-31 REF_UPDATE_INT	REF_UPDATE_INT This field allows the user to add additional delay cycles to the DLL control loop (reference delay line control). By default, the DLL control loop shall update every two GPMICLK cycles. Programming this field results in a DLL control loop update interval of (2 + REF_UPDATE_INT) * GPMICLK. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that may effect the delay (such as voltage and temperature)

9.7.4.20 GPMI Double Rate Read DLL Status Register Description (READ_DDR_DLL_STS)

9.7.4.20.1 Offset

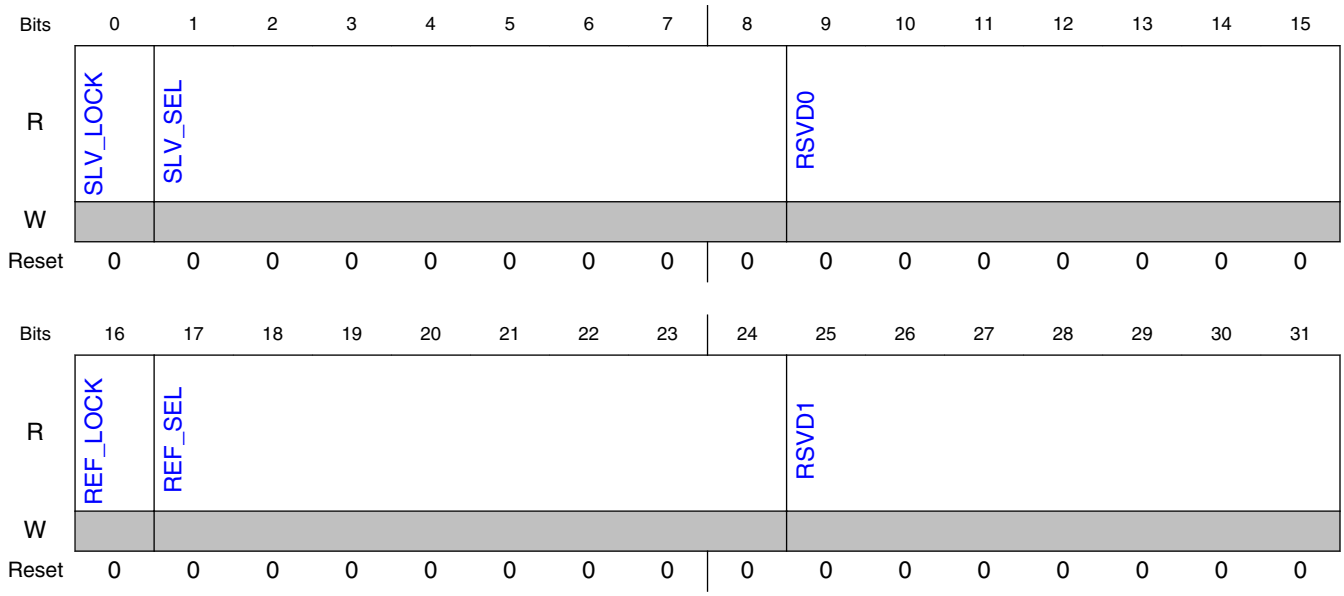
Register	Offset
READ_DDR_DLL_STS	120h

9.7.4.20.2 Function

GPMI Double Rate Read DLL Status Register, Read Only. GPMI DLL status fields are provided in this register.

GPMI_READ_DDR_DLL_STS 0x120

9.7.4.20.3 Diagram



9.7.4.20.4 Fields

Field	Function
0 SLV_LOCK	SLV_LOCK Slave delay-line lock status. This signifies that a valid calibration has been set to the slave-delay line and that the slave-delay line is implementing the programmed delay value.
1-8 SLV_SEL	SLV_SEL Slave delay line select status
9-15 RSVD0	RSVD0 Reserved
16 REF_LOCK	REF_LOCK Reference DLL lock status. This signifies that the DLL has detected and locked to a half-phase GPMICLK shift, allowing the slave delay-line to perform programmed clock delays.
17-24 REF_SEL	REF_SEL Reference delay line select status.
25-31 RSVD1	RSVD1 Reserved

9.7.4.21 GPMI Double Rate Write DLL Status Register Description (WRITE_DDR_DLL_STS)

9.7.4.21.1 Offset

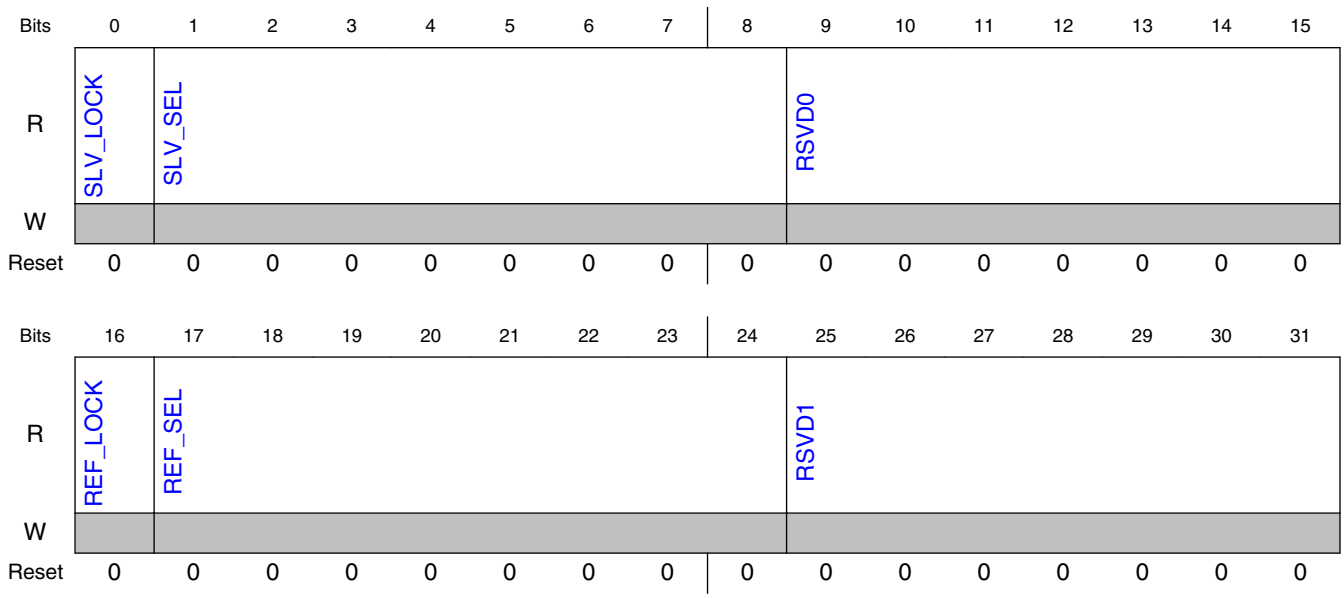
Register	Offset
WRITE_DDR_DLL_STS	130h

9.7.4.21.2 Function

GPMI Double Rate Write DLL Status Register, Read Only. GPMI DLL status fields are provided in this register.

GPMI_WRITE_DDR_DLL_STS 0x130

9.7.4.21.3 Diagram



9.7.4.21.4 Fields

Field	Function
0 SLV_LOCK	SLV_LOCK Slave delay-line lock status. This signifies that a valid calibration has been set to the slave-delay line and that the slave-delay line is implementing the programmed delay value.
1-8 SLV_SEL	SLV_SEL Slave delay line select status
9-15 RSVD0	RSVD0 Reserved

Table continues on the next page...

Field	Function
16 REF_LOCK	REF_LOCK Reference DLL lock status. This signifies that the DLL has detected and locked to a half-phase GPMICLK shift, allowing the slave delay-line to perform programmed clock delays.
17-24 REF_SEL	REF_SEL Reference delay line select status.
25-31 RSVD1	RSVD1 Reserved

Chapter 10
Mass Storage

10.1 Enhanced Configurable SPI (ECSPI)

10.1.1 Overview

The Enhanced Configurable Serial Peripheral Interface (ECSPI) is a full-duplex, synchronous, four-wire serial communication block.

The ECSPI contains a 64 x 32 receive buffer (RXFIFO) and a 64 x 32 transmit buffer (TXFIFO). With data FIFOs, the ECSPI allows rapid data communication with fewer software interrupts.

10.1.1.1 Block Diagram

The figure below shows a block diagram of the ECSPI.

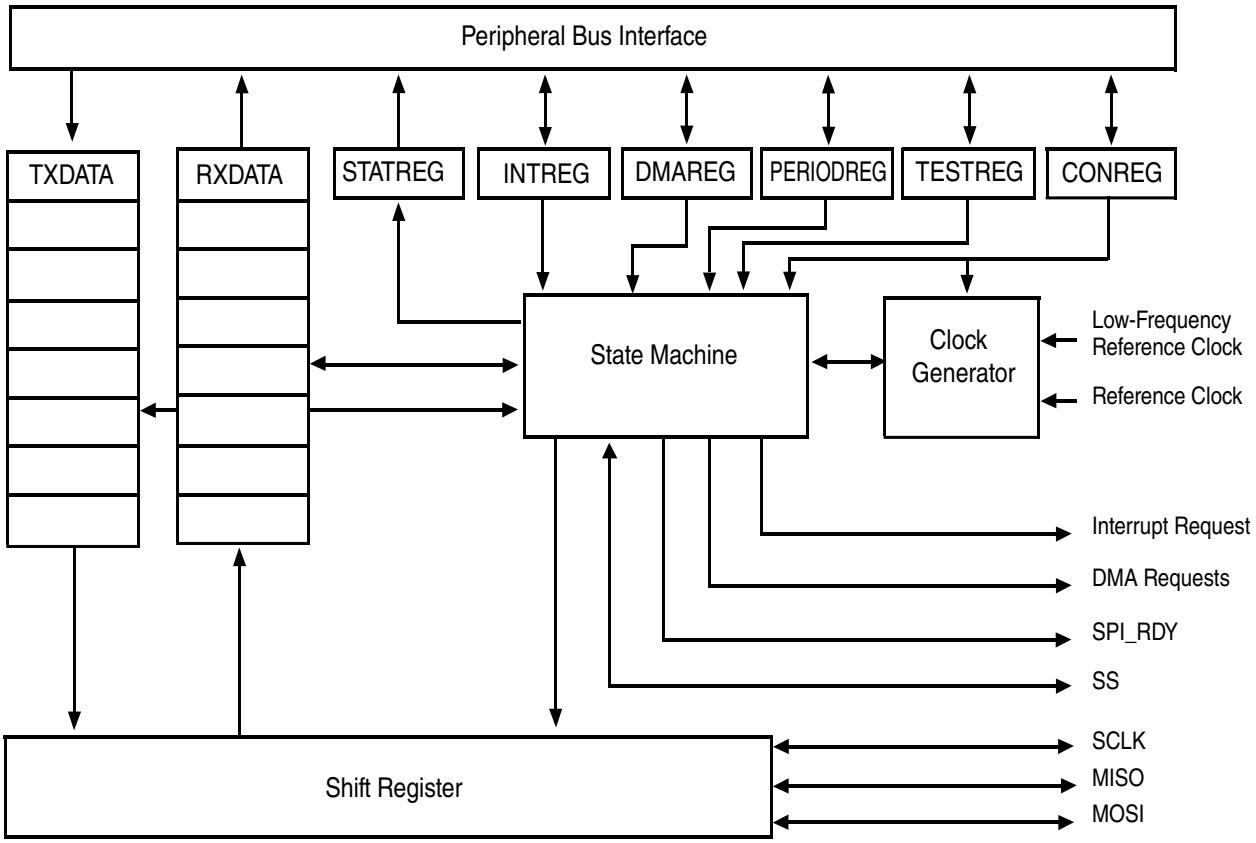


Figure 10-1. ECSPI Block Diagram

10.1.1.2 Features

Key features of the ECSPI include:

- Full-duplex synchronous serial interface
- Master/Slave configurable
- One Chip Select (SS) signal
- Transfer continuation function allows unlimited length data transfers
- 32-bit wide by 64-entry FIFO for both transmit and receive data
- Polarity and phase of the Chip Select (SS) and SPI Clock (SCLK) are configurable
- Direct Memory Access (DMA) support
- Refer to the product data sheet for the maximum operating frequency

10.1.1.3 Modes and Operations

The ECSPI supports the modes described in the indicated sections:

- Master Mode
- Slave Mode
- Low Power Modes

As described in Operations, the ECSPI supports the operations described in the indicated sections:

- Typical Master Mode
  - Master Mode with SPI_RDY
  - Master Mode with Wait States
  - Master Mode with SS_CTL[3:0] Control
  - Master Mode with Phase Control
- Typical Slave Mode

10.1.2 Functional Description

This section provides a complete functional description of the ECSPI. The figure found here shows the relationship of SCLK and data lines while ECSPI has been configured with different POL and PHA settings.

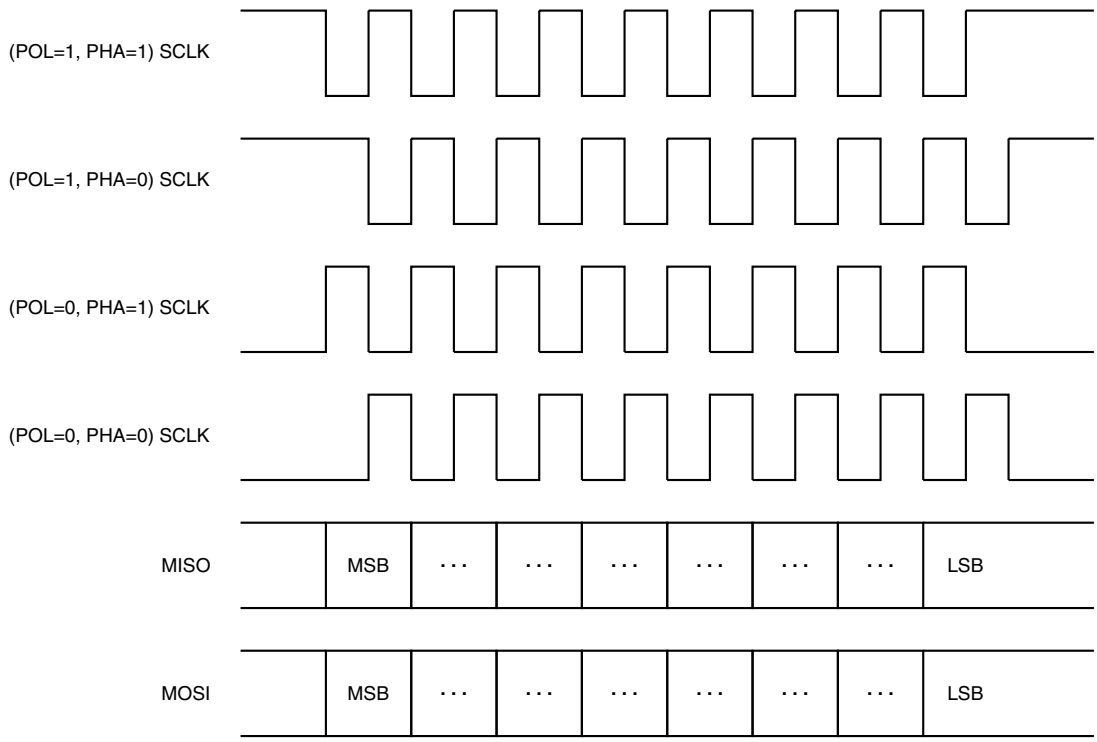


Figure 10-2. ECSPI SCLK, MISO, and MOSI Relationship

10.1.2.1 Master Mode

When the ECSPI is configured as a master, it uses a serial link to transfer data between the ECSPI and an external slave device.

The Chip Select (SS) signal and the clock signal (SCLK) are used to transfer data between two devices. If the external device is a transmit-only device, the ECSPI master's output port can be ignored and used for other purposes. In order to use the internal TXFIFO and RXFIFO, two auxiliary output signals, Chip Select (SS) and SPI_RDY, are used for data transfer rate control. Software can also configure the sample period control register to a fixed data transfer rate.

10.1.2.2 Slave Mode

When the ECSPI is configured as a slave, software can configure the ECSPI Control register to match the external SPI master's timing.

In this configuration, Chip Select (SS) becomes an input signal, and is used to control data transfers through the Shift register, as well as to load/store the data FIFO.

Slave mode only supports the case when ECSPIx_CONFIGREG[SS_CTL] is cleared. The accurate burst length should always be specified using the BURST_LENGTH parameter. ECSPIx_CONFIGREG[SS_CTL] set to 1 is not supported in slave mode.

10.1.2.3 Low Power Modes

The ECSPI does not operate under low power mode.

It holds its operation when its clock is gated off in master mode. In slave mode, the ECSPI does not respond when its clock is gated off.

10.1.2.4 Operations

The information found here describes the ECSPI's operations.

10.1.2.4.1 Typical Master Mode

The ECSPI master uses the Chip Select (SS) signal to enable an external SPI device, and uses the SCLK signal to transfer data in and out of the Shift register.

The SPI_RDY enables fast data communication with fewer software interrupts. By programming the ECSPI_PERIODREG register accordingly, the ECSPI can be used for a fixed data transfer rate.

When the ECSPI is in Master mode the SS, SCLK, and MOSI are output signals, and the MISO signal is an input.

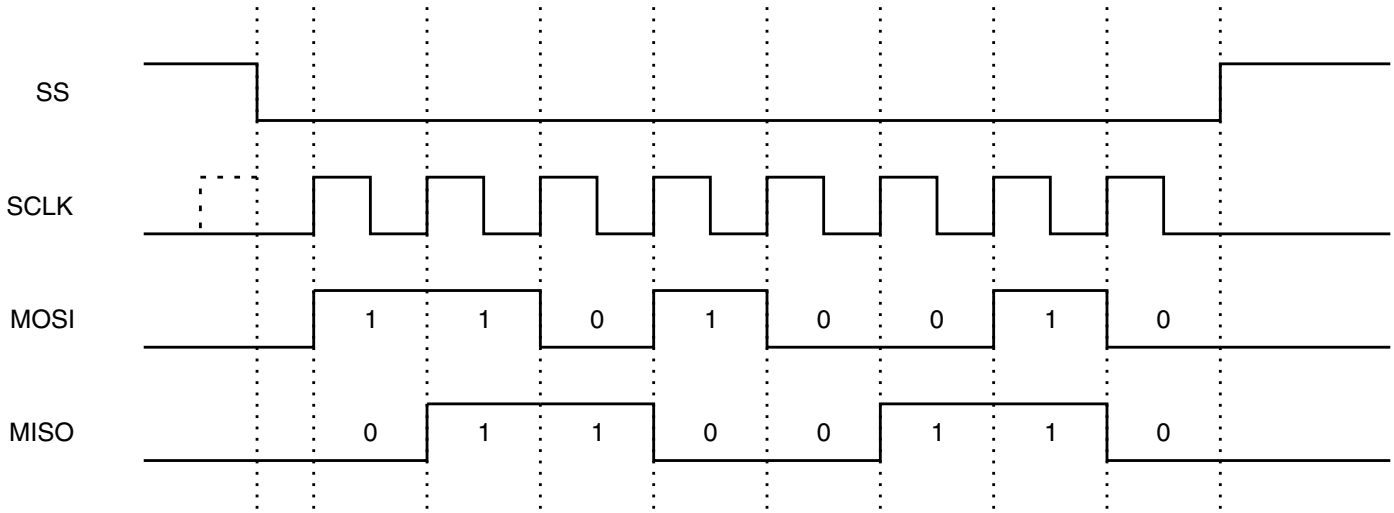


Figure 10-3. Typical SPI Burst (8-bit Transfer)

In the above figure, the Chip Select (SS) signal enables the selected external SPI device, and the SCLK synchronizes the data transfer. The MOSI and MISO signals change on rising edge of SCLK and the MISO signal is latched on the falling edge of the SCLK. The figure above shows a data of 0xD2 is shifted out, and a data of 0x66 is shifted in.

10.1.2.4.1.1 Master Mode with SPI_RDY

By default, the ECSPI does not use the SPI_RDY signal in master mode (MODE =1).

A SPI burst begins when the following events happen:

- The ECSPI is enabled, TXFIFO has data in it, and ECSPI_CONREG[XCH] bit or the ECSPI_CONREG[SMC] bit is set.
- When the SPI Data Ready Control (ECSPI_CONREG[DRCTL]) bits contains either 01 or 10, the SPI_RDY signal controls when a SPI burst starts.

A SPI burst is defined as a bus transaction that starts when the slave select is asserted and ends when the slave select is negated. The Chip Select (SS) signal will remain asserted until all the bits in a SPI burst are shifted out.

If ECSPI_CONREG[DRCTL] is set to 01, the SPI burst can be triggered only if a falling edge of the SPI_RDY signal has been detected.

The following figure shows the relationship between a SPI burst and the falling edge of SPI_RDY signal.

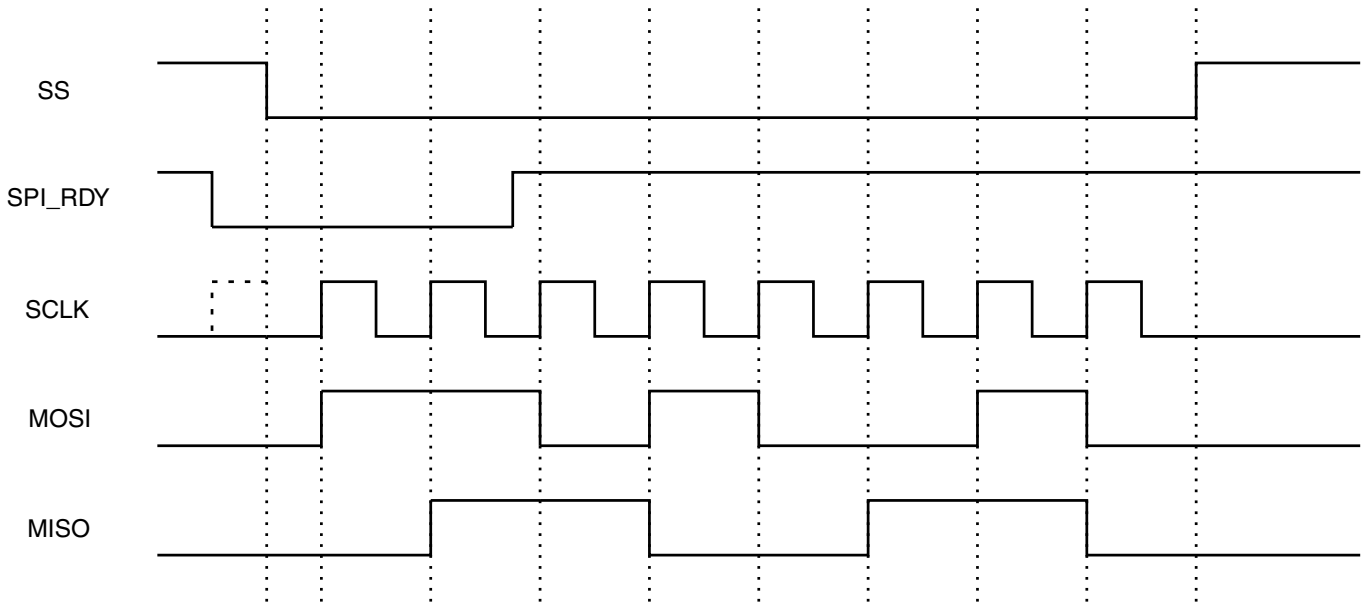


Figure 10-4. Relationship Between a SPI Burst and SPI_RDY: Falling-Edge Triggered

A SPI burst does not start until the falling edge of the SPI_RDY signal is detected. The next SPI burst starts when the next SPI_RDY falling edge is detected, after the last burst has finished.

If SPI Data Ready Control (ECSPI_CONREG[DRCTL]) is set to 10, the SPI burst can be triggered only if the SPI_RDY signal is low.

The following figure shows the relationship between a SPI burst and the SPI_RDY signal. The SPI burst does not begin until the SPI_RDY signal goes low. The ECSPI will keep transmitting SPI burst if the SPI_RDY signal remains low.

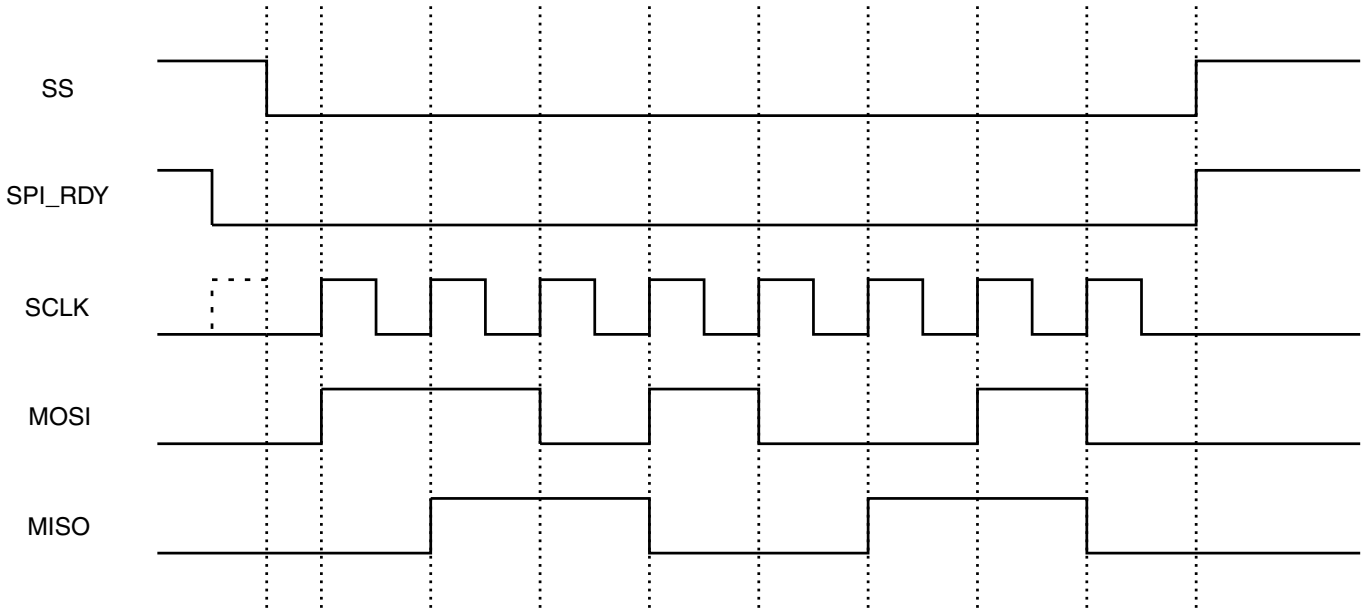


Figure 10-5. Relationship Between a SPI Burst and SPI_RDY: Low-Level Triggered

10.1.2.4.1.2 Master Mode with Wait States

Wait states can be inserted between SPI bursts. This provides a way for software to slow down the SPI burst to meet the timing requirements of a slower SPI device.

The following figure shows wait states inserted between SPI bursts.

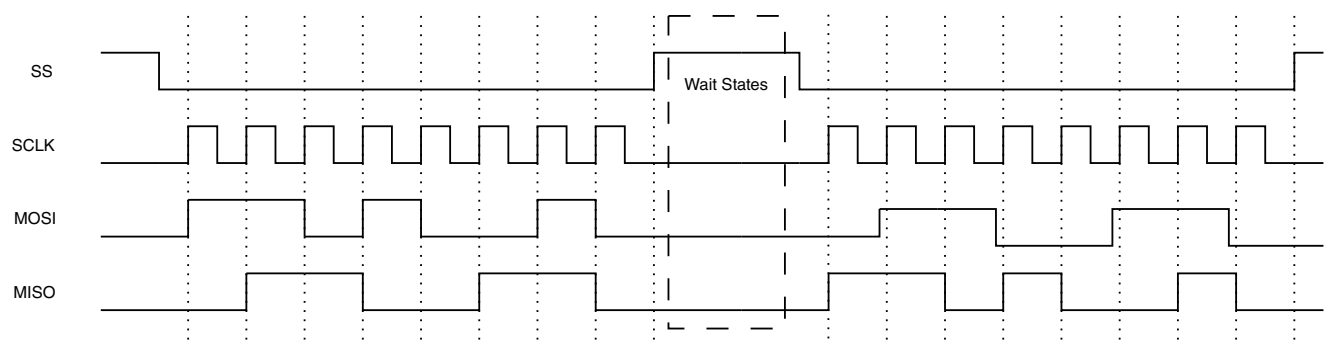


Figure 10-6. SPI Bursts with Wait States

In this case, the number of wait states is controlled by ECSPI_PERIODREG[SAMPLE PERIOD] and the wait states' clock source is selected by ECSPI_PERIODREG[CSRC].

10.1.2.4.1.3 Master Mode with SS_CTL[3:0] Control

The SPI SS Control (SS_CTL[3:0]) controls whether the current operation is single burst or multiple bursts.

When the SPI SS Wave Form Select (SS_CTL[3:0]) is set, the current operation is multiple bursts transfer. When the SPI SS Wave Form Select (SS_CTL[3:0]) bit is cleared, the current operation is single burst transfer. A SPI burst can contains multiple words as defined in the BURST LENGTH field of the ECSPI_CONREG register.

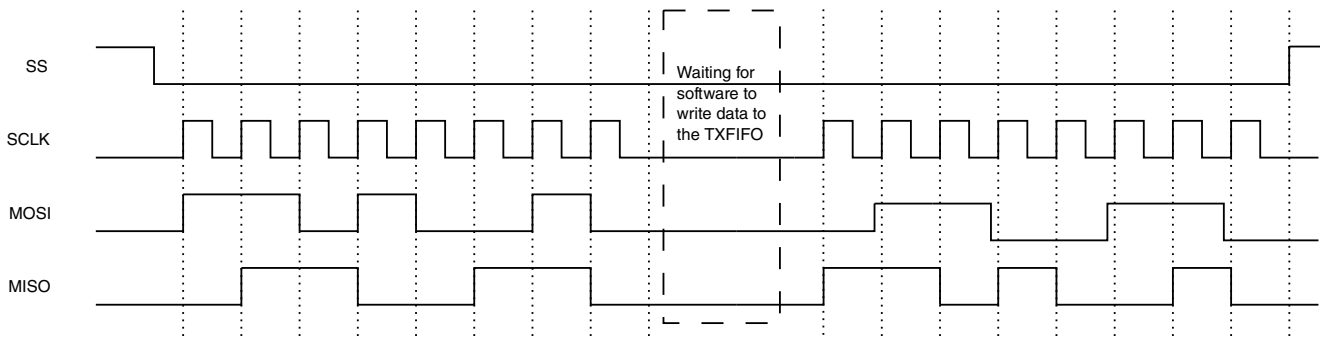


Figure 10-7. SPI Burst While SS_CTL[3:0] is Clear

In Figure 10-7, two 8-bit bursts in the TXFIFO have been combined and transmitted in one SPI burst. The maximum length of a single SPI burst is defined by the BURST LENGTH and limited by the FIFO size. (Figure 10-7 corresponds to a BURST LENGTH of 8.) This provides a way for transferring a longer SPI burst by writing data into TXFIFO while the ECSPI is transmitting.

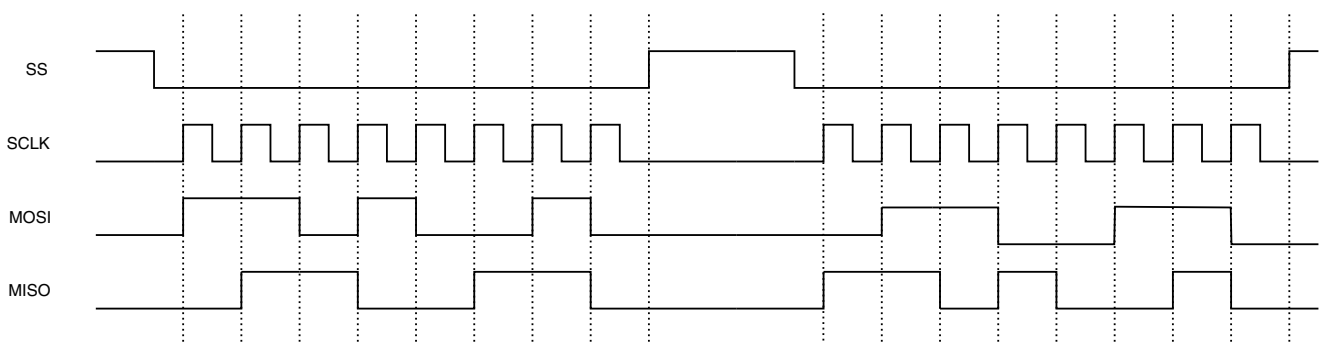


Figure 10-8. SPI Bursts While SS_CTL[3:0] is Set

In Figure 10-8, two FIFO entries are transmitted, one entry with each SPI burst. The ECSPI will continue to transmit SPI bursts until the TXFIFO is empty. When wait states can be inserted between SPI bursts, the SS will negate between SPI bursts until the wait states finish.

10.1.2.4.1.4 Master Mode with Phase Control

The Phase Control (ECSPI_CONREG[PHA]) bit controls how the transmit data shifts out and the receive data shifts in.

When the Phase control (ECSPI_CONREG[PHA]) bit is set, the transmit data will shift out on the rising edge of SCLK, and the receive data is latched on the falling edge of SCLK. The most-significant bit is output on the first rising SCLK edge.

When ECSPI_CONREG[PHA] is cleared, the transmit data is shifted out on the falling edge of SCLK and the receive data is latched on the rising edge of SCLK. The MSB is output when the host processor loads the transmitted data.

Inverting the SCLK polarity does not impact the edge-triggered operations because they are internal to the serial peripheral interface master. Figure 10-9 shows how SPI burst works with different POL and PHA configuration.

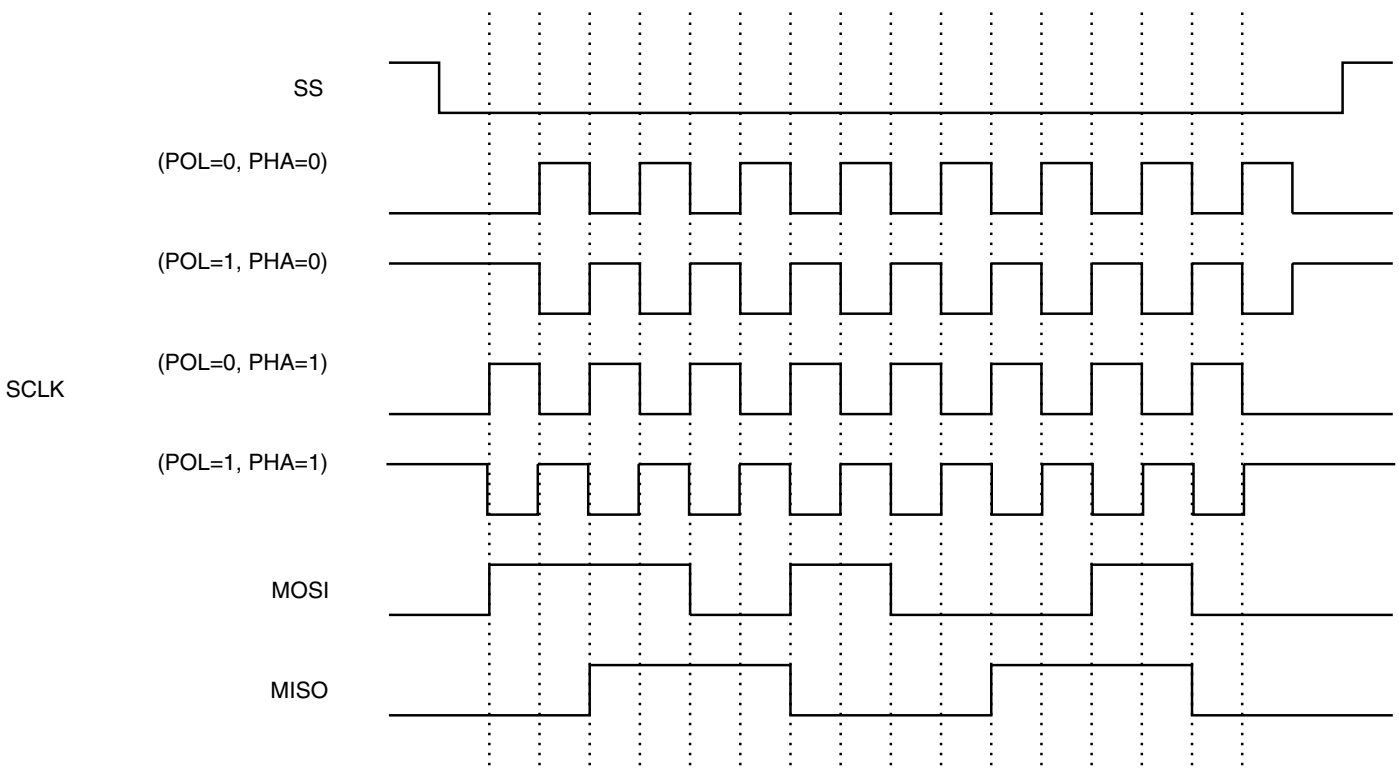


Figure 10-9. SPI Burst with Different POL and PHA Configurations

10.1.2.4.2 Typical Slave Mode

When the ECSPI is configured as a slave (Mode = 0), software can configure the ECSPI Control register to match the external SPI master's timing. The SPI master timing should match the ECSPI timing limitation in the product data sheet. In this configuration, SS becomes an input signal, and is used to latch data in and out of the internal data Shift registers, as well as to advance the data FIFO.

The SS, SCLK, and MOSI are inputs and MISO is output. Most of the timing diagrams are similar to the diagrams shown previously for the SPI in Master mode (Mode = 1), because the inputs come from a SPI master device.

However, the timing is different when SS is used to advance the data FIFO. When the SS_POL=0 is set while the ECSPI is configured in Slave mode, the data FIFO will advance on the rising edge of the SS signal. When the polarity is reversed (SS_POL = 1), the data FIFO will advance on the falling edge of the SS signal.

The figure below shows a SPI burst in which the data FIFO is advanced by the rising edge of the SS signal.

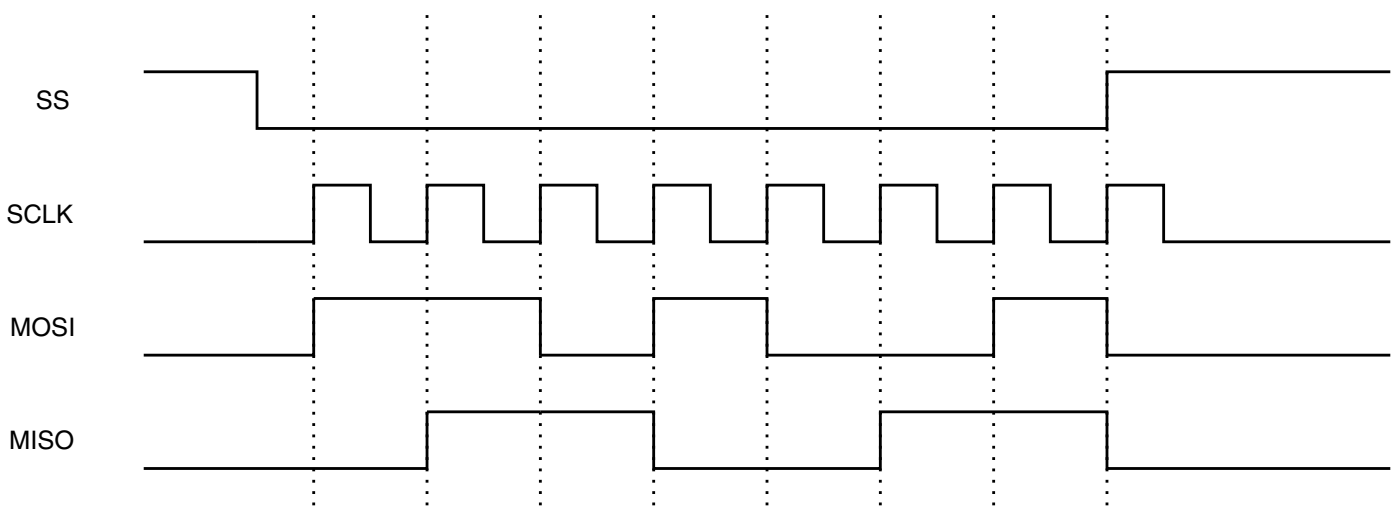


Figure 10-10. Advancing the Data FIFO on the Rising Edge of SS

In the above case, only the most significant 7 bits are loaded to the RXFIFO.

10.1.2.5 Reset

Whenever a device reset occurs, a reset is performed on the ECSPI, resetting all registers to their default values.

Software can reset the block using the CONREG[EN] bit; see ECSPI.

10.1.2.6 Interrupts

Interrupt control provides a way to manage the ECSPI FIFOs:

- For transmitting data, software can enable the TXFIFO empty, TXFIFO data request, and TXFIFO full interrupts to maintain the TXFIFO using an interrupt service routine.
- For receiving data, software can enable the RXFIFO ready, RXFIFO data request, and RXFIFO full interrupts to retrieve data from the RXFIFO using an interrupt service routine.

Other interrupt sources can be used to control or debug the SPI bursts:

- The transfer-completed interrupt means that there is no data left in the TXFIFO and that the data in the Shift register has been shifted out.
- The RXFIFO overflow interrupt means that the RXFIFO received more than 64 words and will not accept any other words.

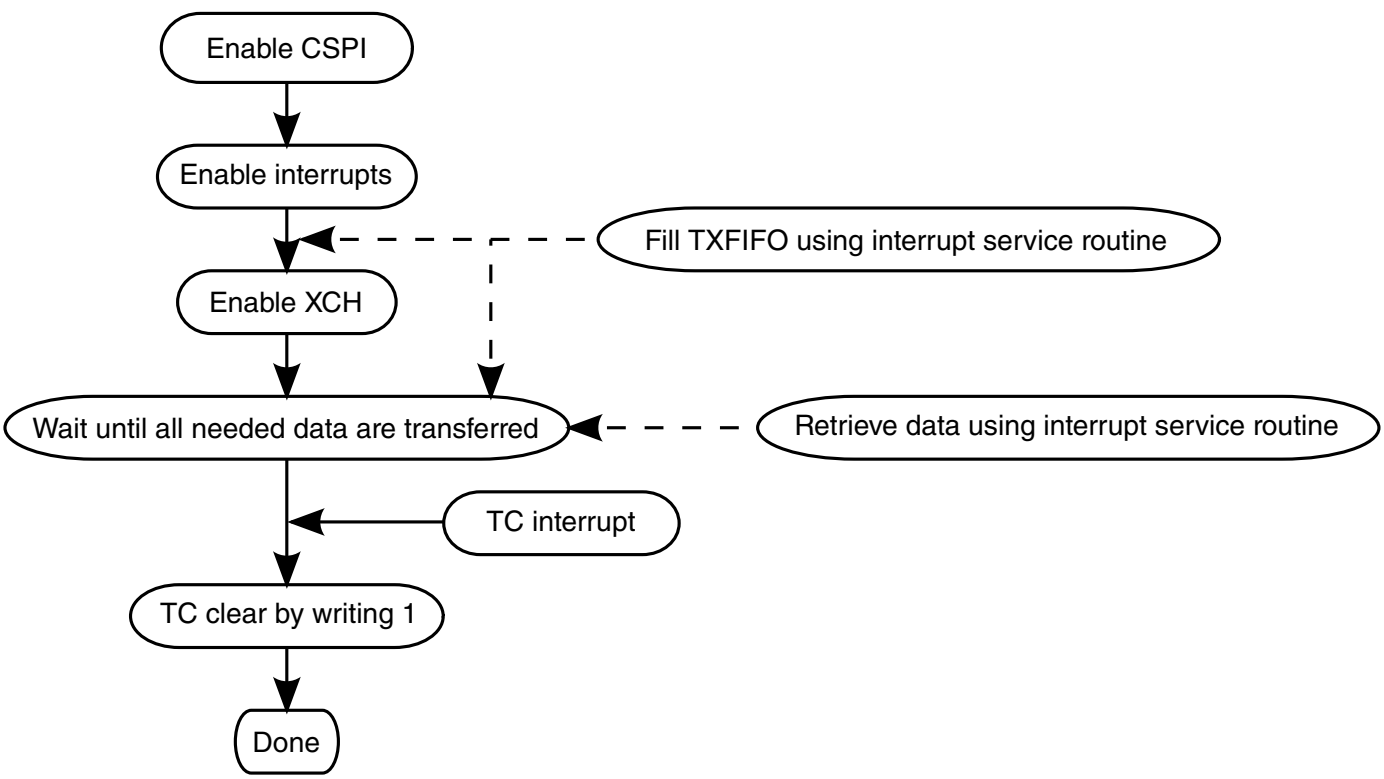


Figure 10-11. Program Sequence of SPI Burst Using Interrupt Control

NOTE

The TC bit does not provide a reliable indication that the transfer is complete. Under some conditions, the TC interrupt can occur before the transfer is completed. If the TC bit is used as an interrupt source, the XCH bit should be polled after the TC interrupt occurs to accurately confirm that the transfer is complete.

10.1.2.7 DMA

DMA control provides another method to utilize the FIFOs in the ECSPI. By using DMA request and acknowledge signals, larger amounts of data can be transferred, and will reduce interrupts and host processor loading. When the appropriate conditions are matched, the block will send out a DMA request.

The DMA can deal with the following conditions:

- TXFIFO empty
- TXFIFO data request
- RXFIFO data request
- RXFIFO full

The figure below shows a program sequence of SPI bursts using DMA control.

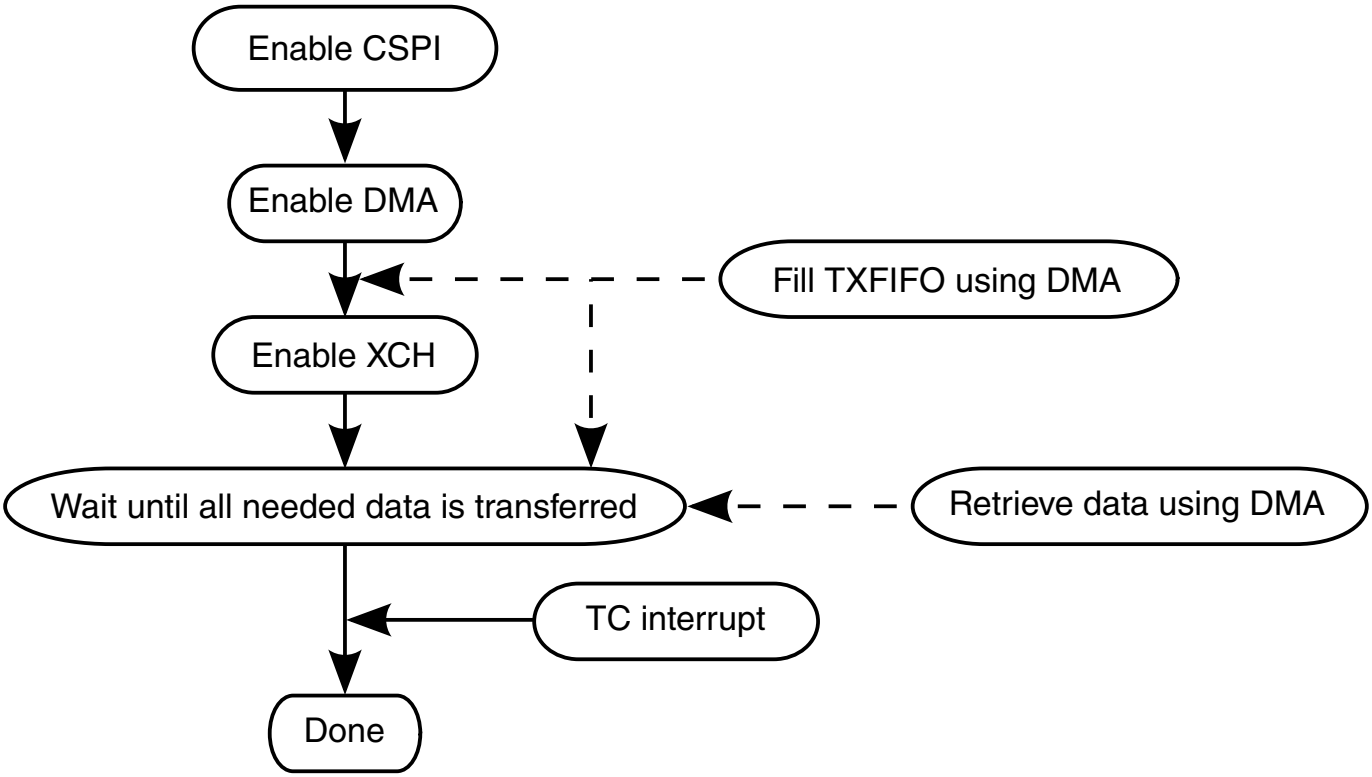


Figure 10-12. Program Sequence of SPI Burst Using DMA

NOTE

The TC bit does not provide a reliable indication that the transfer is complete. Under some conditions, the TC interrupt can occur before the transfer is completed. If the TC bit is used as an interrupt source, the XCH bit should be polled after the TC interrupt occurs to accurately confirm that the transfer is complete.

10.1.2.8 Byte Order

The ECSPI does not support byte re-ordering in hardware.

10.1.3 External Signals

Figure 10-13 shows the ECSPI in master mode connected to an external device in a one-way communication link.

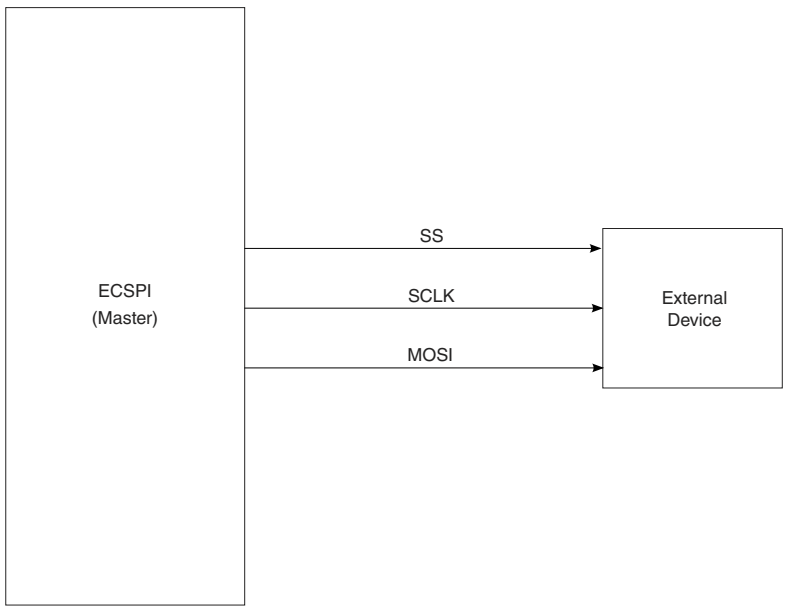


Figure 10-13. Example Connection Diagram

10.1.4 Initialization

This section provides initialization information for ECSPI.

To initialize the block:

1. Clear the EN bit in ECSPI_CONREG to reset the block.
2. Enable the clocks for ECSPI within the CCM.
3. Configure the Control Register and then set the EN bit in the ECSPI_CONREG to put ECSPI out of reset.
4. Configure corresponding IOMUX for ECSPI external signals.
5. Configure registers of ECSPI properly according to the specifications of the external SPI device.

10.1.5 Applications

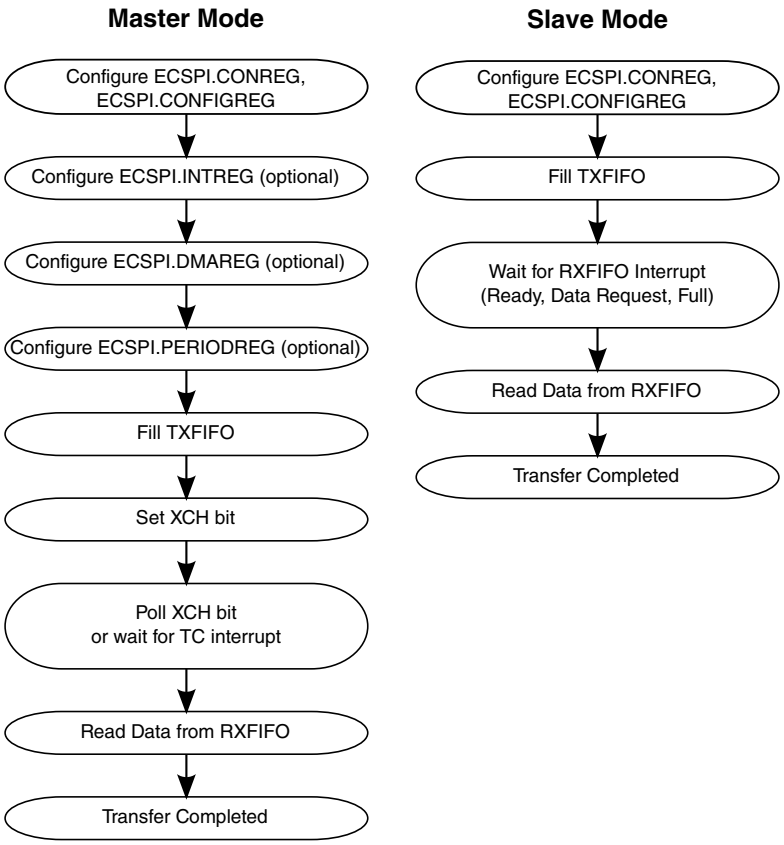


Figure 10-14. Flowchart of the ECSPI Operation

NOTE

The TC bit does not provide a reliable indication that the transfer is complete. Under some conditions, the TC interrupt can occur before the transfer is completed. If the TC bit is used as an interrupt source, the XCH bit should be polled after the TC interrupt occurs to accurately confirm that the transfer is complete.

10.1.6 ECSPI Memory Map/Register Definition

This section includes the block memory map and detailed descriptions of all registers. For the base address of a particular block instantiation, see the system memory map.

ECSPI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3082_0000	Receive Data Register (ECSPI1_RXDATA)	32	R	0000_0000h	10.1.6.1/ 2372
3082_0004	Transmit Data Register (ECSPI1_TXDATA)	32	W	0000_0000h	10.1.6.2/ 2373
3082_0008	Control Register (ECSPI1_CONREG)	32	R/W	0000_0000h	10.1.6.3/ 2373
3082_000C	Config Register (ECSPI1_CONFIGREG)	32	R/W	0000_0000h	10.1.6.4/ 2376
3082_0010	Interrupt Control Register (ECSPI1_INTREG)	32	R/W	0000_0000h	10.1.6.5/ 2378
3082_0014	DMA Control Register (ECSPI1_DMAREG)	32	R/W	0000_0000h	10.1.6.6/ 2379
3082_0018	Status Register (ECSPI1_STATREG)	32	R/W	0000_0003h	10.1.6.7/ 2381
3082_001C	Sample Period Control Register (ECSPI1_PERIODREG)	32	R/W	0000_0000h	10.1.6.8/ 2382
3082_0020	Test Control Register (ECSPI1_TESTREG)	32	R/W	0000_0000h	10.1.6.9/ 2384
3082_0040	Message Data Register (ECSPI1_MSGDATA)	32	W	0000_0000h	10.1.6.10/ 2385
3083_0000	Receive Data Register (ECSPI2_RXDATA)	32	R	0000_0000h	10.1.6.1/ 2372
3083_0004	Transmit Data Register (ECSPI2_TXDATA)	32	W	0000_0000h	10.1.6.2/ 2373
3083_0008	Control Register (ECSPI2_CONREG)	32	R/W	0000_0000h	10.1.6.3/ 2373
3083_000C	Config Register (ECSPI2_CONFIGREG)	32	R/W	0000_0000h	10.1.6.4/ 2376
3083_0010	Interrupt Control Register (ECSPI2_INTREG)	32	R/W	0000_0000h	10.1.6.5/ 2378
3083_0014	DMA Control Register (ECSPI2_DMAREG)	32	R/W	0000_0000h	10.1.6.6/ 2379
3083_0018	Status Register (ECSPI2_STATREG)	32	R/W	0000_0003h	10.1.6.7/ 2381
3083_001C	Sample Period Control Register (ECSPI2_PERIODREG)	32	R/W	0000_0000h	10.1.6.8/ 2382
3083_0020	Test Control Register (ECSPI2_TESTREG)	32	R/W	0000_0000h	10.1.6.9/ 2384
3083_0040	Message Data Register (ECSPI2_MSGDATA)	32	W	0000_0000h	10.1.6.10/ 2385
3084_0000	Receive Data Register (ECSPI3_RXDATA)	32	R	0000_0000h	10.1.6.1/ 2372
3084_0004	Transmit Data Register (ECSPI3_TXDATA)	32	W	0000_0000h	10.1.6.2/ 2373

Table continues on the next page...

ECSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3084_0008	Control Register (ECSPI3_CONREG)	32	R/W	0000_0000h	10.1.6.3/ 2373
3084_000C	Config Register (ECSPI3_CONFIGREG)	32	R/W	0000_0000h	10.1.6.4/ 2376
3084_0010	Interrupt Control Register (ECSPI3_INTREG)	32	R/W	0000_0000h	10.1.6.5/ 2378
3084_0014	DMA Control Register (ECSPI3_DMAREG)	32	R/W	0000_0000h	10.1.6.6/ 2379
3084_0018	Status Register (ECSPI3_STATREG)	32	R/W	0000_0003h	10.1.6.7/ 2381
3084_001C	Sample Period Control Register (ECSPI3_PERIODREG)	32	R/W	0000_0000h	10.1.6.8/ 2382
3084_0020	Test Control Register (ECSPI3_TESTREG)	32	R/W	0000_0000h	10.1.6.9/ 2384
3084_0040	Message Data Register (ECSPI3_MSGDATA)	32	W	0000_0000h	10.1.6.10/ 2385

10.1.6.1 Receive Data Register (ECSPIx_RXDATA)

The Receive Data register (ECSPI_RXDATA) is a read-only register that forms the top word of the 64 x 32 receive FIFO. This register holds the data received from an external SPI device during a data transaction. Only word-sized read operations are allowed.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ECSPI_RXDATA																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_RXDATA field descriptions

Field	Description
ECSPI_RXDATA	Receive Data. This register holds the top word of the receive data FIFO. The FIFO is advanced for each read of this register. The data read is undefined when the Receive Data Ready (RR) bit in the Interrupt Control/Status register is cleared. Zeros are read when ECSPI is disabled.

10.1.6.2 Transmit Data Register (ECSPIx_TXDATA)

The Transmit Data (ECSPI_TXDATA) register is a write-only data register that forms the bottom word of the 64 x 32 TXFIFO. The TXFIFO can be written to as long as it is not full, even when the SPI Exchange bit (XCH) in ECSPI_CONREG is set. This allows software to write to the TXFIFO during a SPI data exchange process. Writes to this register are ignored when the ECSPI is disabled (ECSPI_CONREG[EN] bit is cleared).

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																																
W	ECSPI_TXDATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_TXDATA field descriptions

Field	Description
ECSPI_TXDATA	Transmit Data. This register holds the top word of data loaded into the FIFO. Data written to this register must be a word operation. The number of bits actually transmitted is determined by the BURST_LENGTH field of the corresponding SPI Control register. If this field contains more bits than the number specified by BURST_LENGTH, the extra bits are ignored. For example, to transfer 10 bits of data, a 32-bit word must be written to this register. Bits 9-0 are shifted out and bits 31-10 are ignored. When the ECSPI is operating in Slave mode, zeros are shifted out when the FIFO is empty. Zeros are read when ECSPI is disabled.

10.1.6.3 Control Register (ECSPIx_CONREG)

The Control Register (ECSPI_CONREG) allows software to enable the ECSPI , configure its operating modes, specify the divider value, and SPI_RDY control signal, and define the transfer length.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BURST_LENGTH												CHANNEL_ SELECT		DRCTL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PRE_DIVIDER				POST_DIVIDER				CHANNEL_MODE				SMC	XCH	HT	EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_CONREG field descriptions

Field	Description
31–20 BURST_ LENGTH	Burst Length. This field defines the length of a SPI burst to be transferred. The Chip Select (SS) will remain asserted until all bits in a SPI burst are shifted out. A maximum of 2^12 bits can be transferred in a single SPI burst. In master mode, it controls the number of bits per SPI burst. Since the shift register always loads 32-bit data from transmit FIFO, only the n least-significant (n = BURST LENGTH + 1) will be shifted out. The remaining bits will be ignored. Number of Valid Bits in a SPI burst. 0x000 A SPI burst contains the 1 LSB in a word. 0x001 A SPI burst contains the 2 LSB in a word. 0x002 A SPI burst contains the 3 LSB in a word. ... 0x01F A SPI burst contains all 32 bits in a word. 0x020 A SPI burst contains the 1 LSB in first word and all 32 bits in second word. 0x021 A SPI burst contains the 2 LSB in first word and all 32 bits in second word. ... 0xFFE A SPI burst contains the 31 LSB in first word and 2^7 -1 words. 0xFFF A SPI burst contains 2^7 words.
19–18 CHANNEL_ SELECT	SPI CHANNEL SELECT bits. Selects the external SPI Master/Slave Device. In master mode, these two bits select the external slave device by asserting the Chip Select (SS) output. 00 Channel 0 is selected. Chip Select (SS) will be asserted. 01 Reserved. 10 Reserved. 11 Reserved.
17–16 DRCTL	SPI Data Ready Control. This field selects the utilization of the SPI_RDY signal in master mode. ECSPI checks this field before it starts an SPI burst. 00 The SPI_RDY signal is a don't care. 01 Burst will be triggered by the falling edge of the SPI_RDY signal (edge-triggered). 10 Burst will be triggered by a low level of the SPI_RDY signal (level-triggered). 11 Reserved.
15–12 PRE_DIVIDER	SPI Pre Divider. ECSPI uses a two-stage divider to generate the SPI clock. This field defines the pre-divider of the reference clock. 0000 Divide by 1. 0001 Divide by 2. 0010 Divide by 3. ... 1101 Divide by 14. 1110 Divide by 15. 1111 Divide by 16.
11–8 POST_DIVIDER	SPI Post Divider. ECSPI uses a two-stage divider to generate the SPI clock. This field defines the post-divider of the reference clock using the equation: 2^n . 0000 Divide by 1. 0001 Divide by 2. 0010 Divide by 4. ...

Table continues on the next page...

ECSPIx_CONREG field descriptions (continued)

Field	Description
	1110 Divide by 2 ^{14} . 1111 Divide by 2 ^{15} .
7–4 CHANNEL_ MODE	SPI CHANNEL MODE selects the mode for each SPI channel. CHANNEL MODE[3] is for SPI channel 3. CHANNEL MODE[2] is for SPI channel 2. CHANNEL MODE[1] is for SPI channel 1. CHANNEL MODE[0] is for SPI channel 0. 0 Slave mode. 1 Master mode.
3 SMC	Start Mode Control. This bit applies only to channels configured in Master mode (CHANNEL MODE = 1). It controls how the ECSPI starts a SPI burst, either through the SPI exchange bit, or immediately when the TXFIFO is written to. 0 SPI Exchange Bit (XCH) controls when a SPI burst can start. Setting the XCH bit will start a SPI burst or multiple bursts. This is controlled by the SPI SS Wave Form Select (SS_CTL). Refer to XCH and SS_CTL descriptions. 1 Immediately starts a SPI burst when data is written in TXFIFO.
2 XCH	SPI Exchange Bit. This bit applies only to channels configured in Master mode (CHANNEL MODE = 1). If the Start Mode Control (SMC) bit is cleared, writing a 1 to this bit starts one SPI burst or multiple SPI bursts according to the SPI SS Wave Form Select (SS_CTL). The XCH bit remains set while either the data exchange is in progress, or when the ECSPI is waiting for an active input if SPIRDY is enabled through DRCTL. This bit is cleared automatically when all data in the TXFIFO and the shift register has been shifted out. 0 Idle. 1 Initiates exchange (write) or busy (read).
1 HT	Hardware Trigger Enable. This bit is used in master mode only. It enables hardware trigger (HT) mode. Note, HT mode is not supported by this product. 0 Disable HT mode. 1 Enable HT mode.
0 EN	SPI Block Enable Control. This bit enables the ECSPI. This bit must be set before writing to other registers or initiating an exchange. Writing zero to this bit disables the block and resets the internal logic with the exception of the ECSPI_CONREG. The block's internal clocks are gated off whenever the block is disabled. 0 Disable the block. 1 Enable the block.

10.1.6.4 Config Register (ECSPIx_CONFIGREG)

The Config Register (ECSPI_CONFIGREG) allows software to configure each SPI channel, configure its operating modes, specify the phase and polarity of the clock, configure the Chip Select (SS), and define the HT transfer length. Note, HT mode is not supported by this product.

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			HT_LENGTH					SCLK_CTL				DATA_CTL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SS_POL				SS_CTL				SCLK_POL				SCLK_PHA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_CONFIGREG field descriptions

Field	Description
31–29 -	This field is reserved. Reserved
28–24 HT_LENGTH	HT LENGTH. This field defines the message length in HT Mode. Note, HT mode is not supported by this product. The length in bits of one message is (HT LENGTH + 1).
23–20 SCLK_CTL	SCLK CTL. This field controls the inactive state of SCLK for each SPI channel. SCLK CTL[3] is for SPI channel 3. SCLK CTL[2] is for SPI channel 2. SCLK CTL[1] is for SPI channel 1. SCLK CTL[0] is for SPI channel 0. 0 Stay low. 1 Stay high.
19–16 DATA_CTL	DATA CTL. This field controls inactive state of the data line for each SPI channel. DATA CTL[3] is for SPI channel 3. DATA CTL[2] is for SPI channel 2. DATA CTL[1] is for SPI channel 1. DATA CTL[0] is for SPI channel 0. 0 Stay high. 1 Stay low.
15–12 SS_POL	SPI SS Polarity Select. In both Master and Slave modes, this field selects the polarity of the Chip Select (SS) signal. SS POL[3] is reserved.

Table continues on the next page...

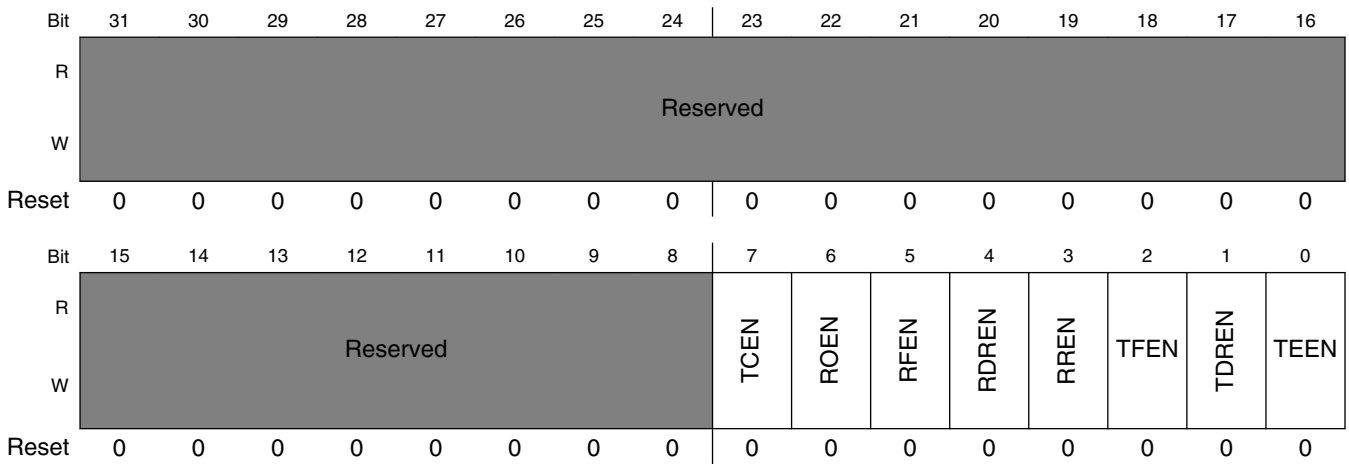
ECSPIx_CONFIGREG field descriptions (continued)

Field	Description
	SS POL[2] is reserved. SS POL[1] is reserved. SS POL[0] is for SPI channel 0. 0 Active low. 1 Active high.
11–8 SS_CTL	SPI SS Wave Form Select. In master mode, this field controls the output wave form of the Chip Select (SS) signal when the SMC (Start Mode Control) bit is cleared. The SS_CTL bits are ignored if the SMC bit is set. SS CTL[3] is reserved. SS CTL[2] is reserved. SS CTL[1] is reserved. SS CTL[0] is for SPI channel 0. In slave mode, this bit controls when the SPI burst is completed. An SPI burst is completed by the Chip Select (SS) signal edges. (SSPOL = 0: rising edge; SSPOL = 1: falling edge) The RXFIFO is advanced whenever a Chip Select (SS) signal edge is detected or the shift register contains 32-bits of valid data. 0 In master mode - only one SPI burst will be transmitted. 1 In master mode - Negate Chip Select (SS) signal between SPI bursts. Multiple SPI bursts will be transmitted. The SPI transfer will automatically stop when the TXFIFO is empty. 0 In slave mode - an SPI burst is completed when the number of bits received in the shift register is equal to (BURST LENGTH + 1). Only the n least-significant bits (n = BURST LENGTH[4:0] + 1) of the first received word are valid. All bits subsequent to the first received word in RXFIFO are valid. 1 Reserved
7–4 SCLK_POL	SPI Clock Polarity Control. This field controls the polarity of the SCLK signal. See Figure 10-9 for more information. SCLK_POL[3] is for SPI channel 3. SCLK_POL[2] is for SPI channel 2. SCLK_POL[1] is for SPI channel 1. SCLK_POL[0] is for SPI channel 0. 0 Active high polarity (0 = Idle). 1 Active low polarity (1 = Idle).
SCLK_PHA	SPI Clock/Data Phase Control. This field controls the clock/data phase relationship. See Figure 10-9 for more information. SCLK PHA[3] is for SPI channel 3. SCLK PHA[2] is for SPI channel 2. SCLK PHA[1] is for SPI channel 1. SCLK PHA[0] is for SPI channel 0. 0 Phase 0 operation. 1 Phase 1 operation.

10.1.6.5 Interrupt Control Register (ECSPIx_INTREG)

The Interrupt Control Register (ECSPI_INTREG) enables the generation of interrupts to the host processor. If the ECSPI is disabled, this register reads zero.

Address: Base address + 10h offset



ECSPIx_INTREG field descriptions

Field	Description
31–8 -	This field is reserved. Reserved
7 TCEN	Transfer Completed Interrupt enable. This bit enables the Transfer Completed Interrupt. 0 Disable 1 Enable
6 ROEN	RXFIFO Overflow Interrupt enable. This bit enables the RXFIFO Overflow Interrupt. 0 Disable 1 Enable
5 RFEN	RXFIFO Full Interrupt enable. This bit enables the RXFIFO Full Interrupt. 0 Disable 1 Enable
4 RDREN	RXFIFO Data Request Interrupt enable. This bit enables the RXFIFO Data Request Interrupt when the number of data entries in the RXFIFO is greater than RX_THRESHOLD. 0 Disable 1 Enable
3 RREN	RXFIFO Ready Interrupt enable. This bit enables the RXFIFO Ready Interrupt. 0 Disable 1 Enable

Table continues on the next page...

ECSPIx_INTREG field descriptions (continued)

Field	Description
2 TFEN	TXFIFO Full Interrupt enable. This bit enables the TXFIFO Full Interrupt. 0 Disable 1 Enable
1 TDREN	TXFIFO Data Request Interrupt enable. This bit enables the TXFIFO Data Request Interrupt when the number of data entries in the TXFIFO is less than or equal to TX_THRESHOLD. 0 Disable 1 Enable
0 TEEN	TXFIFO Empty Interrupt enable. This bit enables the TXFIFO Empty Interrupt. 0 Disable 1 Enable

10.1.6.6 DMA Control Register (ECSPIx_DMAREG)

The Direct Memory Access Control Register (ECSPI_DMAREG) provides software a way to use an on-chip DMA controller for ECSPI data. Internal DMA request signals enable direct data transfers between the ECSPI FIFOs and system memory. The ECSPI sends out DMA requests when the appropriate FIFO conditions are matched.

If the ECSPI is disabled, this register is read as 0.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RXTDEN	Reserved	RX_DMA_LENGTH						RXDEN	Reserved	RX_THRESHOLD					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEDEN	Reserved	TX_THRESHOLD					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_DMAREG field descriptions

Field	Description
31 RXTDEN	RXFIFO TAIL DMA Request/Interrupt Enable. This bit enables an internal counter that is increased at each read of the RXFIFO. This counter is cleared automatically when it reaches RX DMA LENGTH. If the number of words remaining in the RXFIFO is greater than or equal to RX DMA LENGTH, a DMA request/ interrupt is generated even if it is less than or equal to RX_THRESHOLD. 0 Disable 1 Enable
30 -	This field is reserved. Reserved
29–24 RX_DMA_ LENGTH	RX DMA LENGTH. This field defines the burst length of a DMA operation. Applies only when RXTDEN is set.
23 RXDEN	RXFIFO DMA Request Enable. This bit enables/disables the RXFIFO DMA Request. 0 Disable 1 Enable
22 -	This field is reserved. Reserved
21–16 RX_ THRESHOLD	RX THRESHOLD. This field defines the FIFO threshold that triggers a RX DMA/INT request. A RX DMA/INT request is issued when the number of data entries in the RXFIFO is greater than RX_THRESHOLD.
15–8 -	This field is reserved. Reserved
7 TEDEN	TXFIFO Empty DMA Request Enable. This bit enables/disables the TXFIFO Empty DMA Request. 0 Disable 1 Enable
6 -	This field is reserved. Reserved
TX_ THRESHOLD	TX THRESHOLD. This field defines the FIFO threshold that triggers a TX DMA/INT request. A TX DMA/INT request is issued when the number of data entries in the TXFIFO is not greater than TX_THRESHOLD.

10.1.6.7 Status Register (ECSPIx_STATREG)

The ECSPI Status Register (ECSPI_STATREG) reflects the status of the ECSPI's operating condition. If the ECSPI is disabled, this register reads 0x0000_0003.

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TC	RO	RF	RDR	RR	TF	TDR	TE
W									w1c	w1c						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

ECSPIx_STATREG field descriptions

Field	Description
31–8 -	This field is reserved. Reserved
7 TC	Transfer Completed Status bit. Writing 1 to this bit clears it. **NOTE:** The TC bit does not provide a reliable indication that the transfer is complete. Under some conditions, the TC interrupt can occur before the transfer is completed. If the TC bit is used as an interrupt source, the XCH bit should be polled after the TC interrupt occurs to accurately confirm that the transfer is complete. 0 Transfer in progress. 1 Transfer completed.
6 RO	RXFIFO Overflow. When set, this bit indicates that RXFIFO has overflowed. Writing 1 to this bit clears it. 0 RXFIFO has no overflow. 1 RXFIFO has overflowed.
5 RF	RXFIFO Full. This bit is set when the RXFIFO is full. 0 Not Full. 1 Full.
4 RDR	RXFIFO Data Request. 0 When RXTDE is set - Number of data entries in the RXFIFO is not greater than RX_THRESHOLD. 1 When RXTDE is set - Number of data entries in the RXFIFO is greater than RX_THRESHOLD or a DMA TAIL DMA condition exists. 0 When RXTDE is clear - Number of data entries in the RXFIFO is not greater than RX_THRESHOLD. 1 When RXTDE is clear - Number of data entries in the RXFIFO is greater than RX_THRESHOLD.

Table continues on the next page...

ECSPIx_STATREG field descriptions (continued)

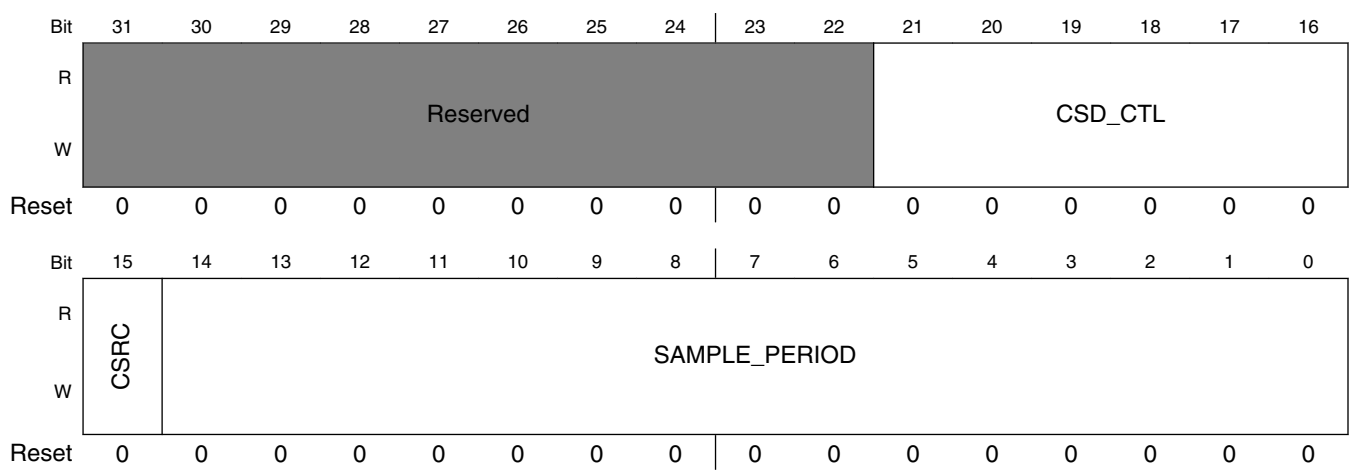
Field	Description
3 RR	RXFIFO Ready. This bit is set when one or more words are stored in the RXFIFO. 0 No valid data in RXFIFO. 1 More than 1 word in RXFIFO.
2 TF	TXFIFO Full. This bit is set when if the TXFIFO is full. 0 TXFIFO is not Full. 1 TXFIFO is Full.
1 TDR	TXFIFO Data Request. 0 Number of valid data slots in TXFIFO is greater than TX_THRESHOLD. 1 Number of valid data slots in TXFIFO is not greater than TX_THRESHOLD.
0 TE	TXFIFO Empty. This bit is set if the TXFIFO is empty. 0 TXFIFO contains one or more words. 1 TXFIFO is empty.

10.1.6.8 Sample Period Control Register (ECSPIx_PERIODREG)

The Sample Period Control Register (ECSPI_PERIODREG) provides software a way to insert delays (wait states) between consecutive SPI transfers. Control bits in this register select the clock source for the sample period counter and the delay count indicating the number of wait states to be inserted between data transfers.

The delay counts apply only when the current channel is operating in Master mode (ECSPI_CONREG[CHANNEL MODE] = 1).ECSPI_PERIODREG also contains the CSD CTRL field used to insert a delay between the Chip Select's active edge and the first SPI Clock edge.

Address: Base address + 1Ch offset



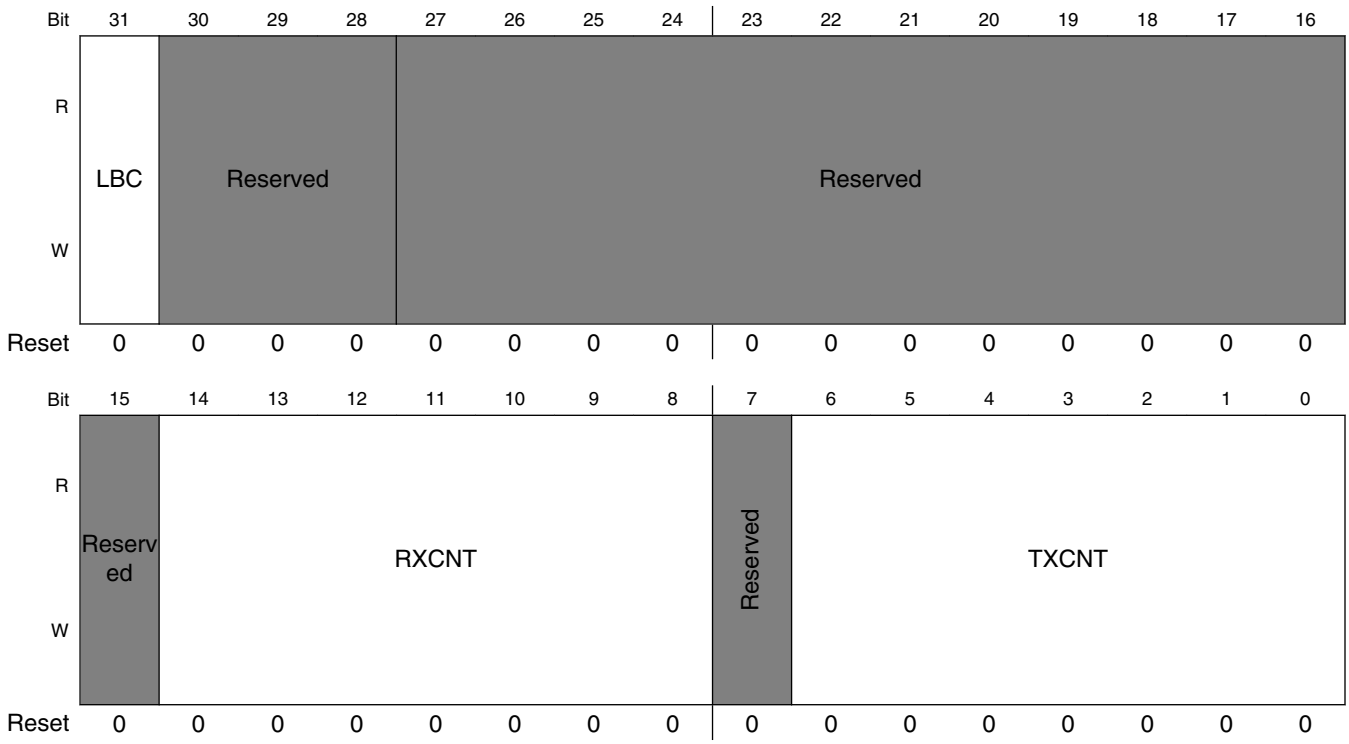
ECSPIx_PERIODREG field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–16 CSD_CTL	Chip Select Delay Control bits. This field defines how many SPI clocks will be inserted between the chip select's active edge and the first SPI clock edge. The range is from 0 to 63.
15 CSRC	Clock Source Control. This bit selects the clock source for the sample period counter. 0 SPI Clock (SCLK) 1 Low-Frequency Reference Clock (32.768 KHz)
SAMPLE_ PERIOD	Sample Period Control. These bits control the number of wait states to be inserted in data transfers. During the idle clocks, the state of the SS output will operate according to the SS_CTL control field in the ECSPI_CONREG register. 0x0000 0 wait states inserted 0x0001 1 wait state inserted ... ... 0x7FFE 32766 wait states inserted 0x7FFF 32767 wait states inserted

10.1.6.9 Test Control Register (ECSPIx_TESTREG)

The Test Control Register (ECSPI_TESTREG) provides software a mechanism to internally connect the receive and transmit devices of the ECSPI, and monitor the contents of the receive and transmit FIFOs.

Address: Base address + 20h offset



ECSPIx_TESTREG field descriptions

Field	Description
31 LBC	Loop Back Control. This bit is used in Master mode only. When this bit is set, the ECSPI connects the transmitter and receiver sections internally, and the data shifted out from the most-significant bit of the shift register is looped back into the least-significant bit of the Shift register. In this way, a self-test of the complete transmit/receive path can be made. The output pins continue to operate normally, however the input pins are ignored. 0 Not connected. 1 Transmitter and receiver sections internally connected for Loopback.
30–28 -	This field is reserved. Reserved, all bits should be ignored.
27–15 -	This field is reserved. Reserved
14–8 RXCNT	RXFIFO Counter. This field indicates the number of words in the RXFIFO.

Table continues on the next page...

ECSPIx_TESTREG field descriptions (continued)

Field	Description
7 -	This field is reserved. Reserved
TXCNT	TXFIFO Counter. This field indicates the number of words in the TXFIFO.

10.1.6.10 Message Data Register (ECSPIx_MSGDATA)

The Message Data Register (ECSPI_MSGDATA) forms the top word of the 16 x 32 MSG Data FIFO. Only word-size accesses are allowed for this register. Reading from this register returns zero, and writing to this register stores data in the MSG Data FIFO.

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																																
W	ECSPI_MSGDATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_MSGDATA field descriptions

Field	Description
ECSPI_ MSGDATA	ECSPI_MSGDATA holds the top word of MSG Data FIFO. The MSG Data FIFO is advanced for each write of this register. The data read is zero. The data written to this register is stored in the MSG Data FIFO.

10.2 FlexSPI Controller (FlexSPI)

10.2.1 Overview

FlexSPI supports two SPI channels and up to four external devices. Each channel supports Single/Dual/Quad/Octal mode data transfer (1/2/4/8 bidirectional data lines).

The FlexSPI configuration depends on the chip configuration. See the system-level section for boot information and pinmux for chip-specific information regarding the modes and number of devices supported.

FlexSPI supports communication with both serial flash memory and serial RAM devices. While many of the descriptions, registers, and fields specifically reference flash memory, almost all information can also be applied to serial RAM. Flash memory is used as an example in the tables and figures in this chapter.

NOTE

Terminology in this chapter has been updated to align with JEDEC standard *EXpanded Serial Peripheral Interface (xSPI) for Non Volatile Memory Devices, Version 1.0*.

Table 10-1. Updated terms

Updated term	Deprecated term
Controller	Master
Target	Slave

10.2.1.1 Block diagram

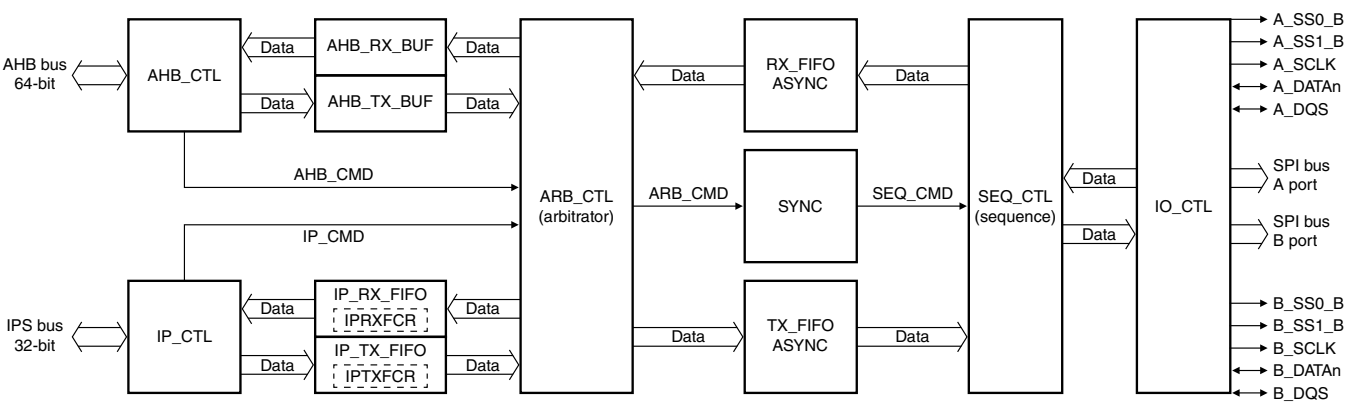


Figure 10-15. Block diagram

10.2.1.2 Features

FlexSPI supports:

- Flexible sequence engine (lookup table) to support various vendor devices
  - Serial NOR flash memory and other devices with SPI protocol similar to serial NOR flash memory
  - Serial NAND flash memory
  - HyperBus devices (HyperFlash/HyperRAM)
  - FPGA devices
- Flash memory access modes
  - Single, Dual, Quad, Octal mode
  - Single Data Transfer Rate (SDR) and Double Data Transfer Rate (DDR) mode
  - Individual/Parallel mode
- Sampling clock mode
  - Internal dummy read strobe looped back internally
  - Internal dummy read strobe looped back from pad
  - Flash-memory-provided read strobe
- Automatic data learning to select the correct sample clock phase
- Memory mapped read and write access by AHB bus
  - AHB receive buffer implemented to reduce read latency. Total AHB receive buffer size: 2048 bytes.
  - 16 AHB controllers with programmable priority for read access from each
  - 8 flexible and configurable buffers in AHB receive buffer
  - AHB transmit buffer implemented to buffer all write data from one AHB burst. AHB transmit buffer size is 64 bytes.

Note

All AHB controllers share this AHB transmit buffer. There is no AHB controller number limitation for write access.

- Software-triggered flash memory read and write access by IP bus
  - IP receive FIFO implemented to buffer all read data from external devices. FIFO size: 512 bytes
  - IP transmit FIFO implemented to buffer all write data to external devices. FIFO size: 1024 bytes
  - DMA support to read IP receive FIFO
  - DMA support to fill IP transmit FIFO
  - SCLK stops when IP receive FIFO is full during reading flash memory data
  - SCLK stops when IP transmit FIFO is empty during writing flash memory data

10.2.2 Functional description

10.2.2.1 Flash connection

There are two FlexSPI interface ports (port A and port B). Each port supports up to two flash devices by providing two chip select outputs.

NOTE

FlexSPI configuration depends on the chip configuration. See the chip-specific FlexSPI information regarding the number of devices supported.

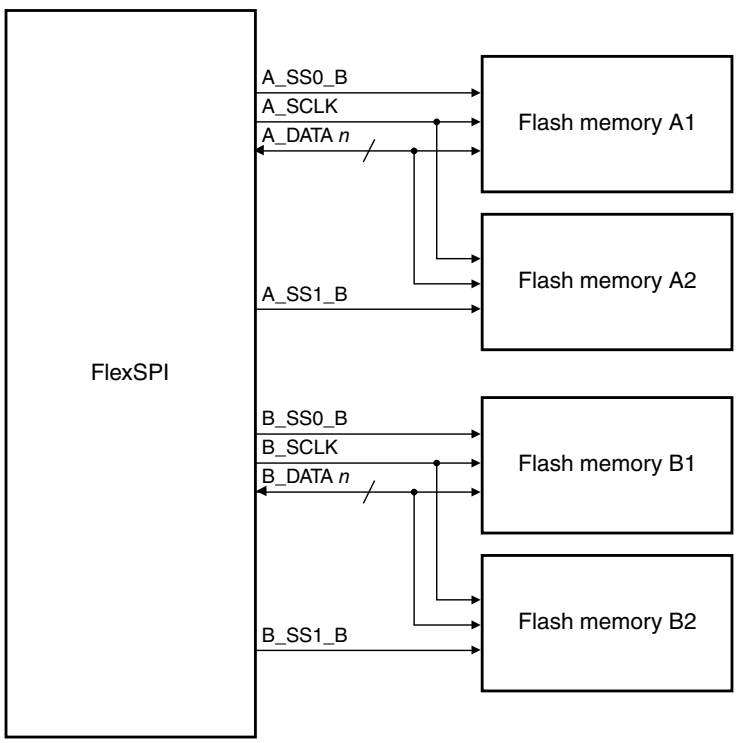


Figure 10-16. FlexSPI connection diagram with four devices.

NOTE

- Flash A1 and Flash A2 can be two flash chip packages or two flash dies in the same package. There is no difference to FlexSPI. This statement is also true for Flash B1 and Flash B2.
- Flash A1 and Flash B1 can be accessed in parallel, using parallel mode. FlexSPI merges or splits the flash read and

program data automatically. This statement is also true for Flash A2 and Flash B2.
- In parallel mode, Flash A1 and Flash A2 cannot be accessed at the same time. This statement is also true for Flash B1 and Flash B2.
- In individual mode, flash devices cannot be accessed at the same time. But these four devices can be accessed separately.

There is a combination mode (MCR0[COMBINATIONEN] = 1) used to provide octal flash memory support by combining Port A (A_DATA[3:0]) and Port B (B_DATA[3:0]) together. Normal mode can also support Octal mode by using all eight data lines from Port A or Port B.Figure 10-17 shows the connection diagram for combination mode.

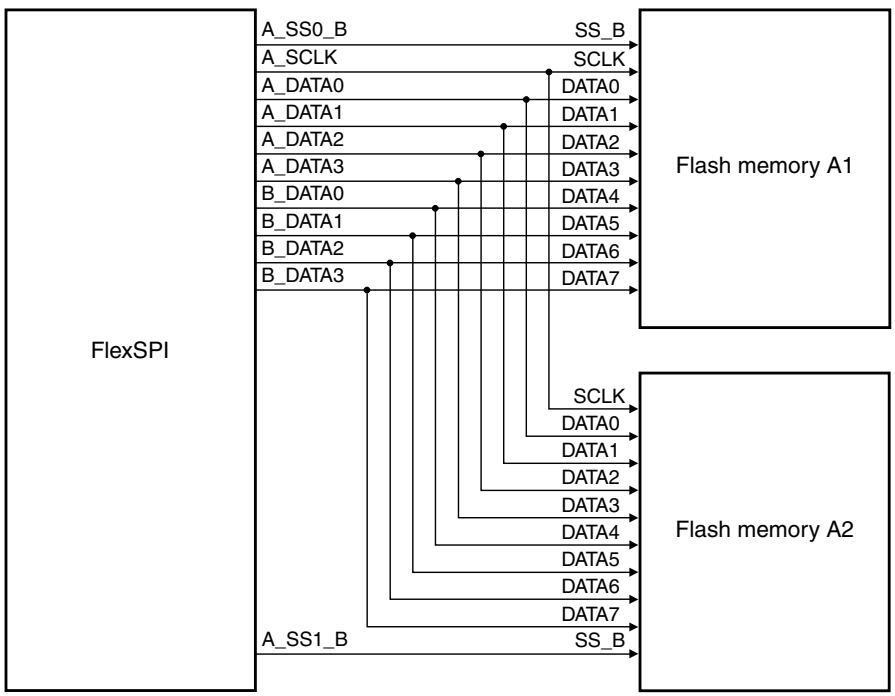


Figure 10-17. FlexSPI connection diagram for combination mode

10.2.2.2 Flash memory access mode

10.2.2.2.1 SPI clock mode

FlexSPI only supports SPI clock mode 0: clock polarity (CPOL) = 0 and clock phase (CPHA) = 0. When the SPI bus is idle, SCLK stays at a logic low state.

10.2.2.2.2 Individual mode and Parallel mode

In individual mode, read and write data are received and transmitted on port A or port B.

In parallel mode, read and write data are received and transmitted on port A and port B in parallel. FlexSPI merges or splits the read and program data automatically (READ/WRITE instruction). Only read and program data is merged or split (READ/WRITE instruction). For other instructions (such as Command, Address, Mode, and Data size), the same associated information is transmitted to the devices connected to port A and port B. For more details, see Executing instructions on SPI interface.

IPCR1[IPAREN] (for IP command) or AHBCR[APAREN] (for AHB command) statistically determines Individual and Parallel mode.

NOTE

FlexSPI does not support 16-bit read and write for the SPI interface.

10.2.2.2.3 SDR mode and DDR mode

In SDR mode, flash memory receives data on the rising edge of SCLK and transmits data on the falling edge of SCLK.

In DDR mode, flash memory receives and transmits data on both the rising and falling edges of SCLK.

The instruction (opcode) in the LUT sequence dynamically determines SDR and DDR modes. There is no static configuration register field setting for SDR and DDR modes. See FlexSPI input timing and FlexSPI output timing for details about input and output timing.

10.2.2.2.4 Single, Dual, Quad, and Octal mode

In Single mode, FlexSPI transmits and receives data on one data pin (DATA0 for transmitting, DATA1 for receiving).

In Dual mode, FlexSPI transmits and receives data on two data pins (DATA0–DATA1 for both transmitting and receiving).

In Quad mode, FlexSPI transmits and receives data on four data pins (DATA0–DATA3 for both transmitting and receiving).

In Octal mode, FlexSPI transmits and receives data on eight data pins (DATA0–DATA7 for both transmitting and receiving).

Instruction (num_pads) in the LUT sequence dynamically determine Single, Dual, Quad, and Octal modes. There is no static configuration register field setting for Single, Dual, Quad, and Octal modes.

10.2.2.3 Modes of operation

FlexSPI operates with these modes.

Table 10-2. Operating modes

Mode	Description
Module Disable	This mode is a low-power mode for FlexSPI. In Module Disable mode, the AHB clock and serial clock domains are gated off internally, but the IPS bus clock domain is not gated off. Read and write access to the control and status register are available, but the LUT, IP receive FIFO, and IP transmit FIFO cannot be accessed. Serial memory access is also not available. Write 1 to MCR0[MDIS] to enter this mode. Write 0 to MCR0[MDIS] to exit this mode.
Doze	This mode is a low-power mode for the chip. When the chip requires FlexSPI to enter Doze mode and MCR0[DOZEEN] = 1, FlexSPI enters Doze mode after all transactions are completed (STS0[ARBIDLE] = 1). In Doze mode, the AHB clock and serial clock domains are gated off internally, but the IPS bus clock is not gated off. Read and write access to the control and status register are available, but the LUT, IP receive FIFO, the IP transmit FIFO cannot be accessed. Serial memory access is also not available. This mode is entered via system request, and exited by deasserting this system request.
Stop	This mode is a low-power mode for the chip. When the chip requires the FlexSPI to enter Stop mode, FlexSPI waits for all transactions to complete (STS0[ARBIDLE] = 1) and return ACK handshake to the system. After the ACK handshake is returned, FlexSPI gates off the AHB clock and serial clock domains internally. The system can gate off the AHB bus clock, IPS bus clock, and serial clock at the system level. This mode is entered via system request. This mode is exited by deasserting this system request, and the ACK handshake message is also deasserted immediately.
Normal	No clock is gated off internally. Normal register access and serial memory access are available.

10.2.2.4 AHB access memory map

FlexSPI allocates AHB memory space for each memory device starting from the FlexSPI region base address of the system memory map. The FLSHxCR0[FLSHSZ] field determines the amount of memory allocated for each memory device in KB, Fx_SIZE. The memory is then allocated sequentially and contiguously.

NOTE

The maximum flash size supported for each device is 256 MB.The maximum total flash size supported (for all four devices) is also 256 MB.

When MCR2[SAMEDEVICEEN] = 1:

- FA1_SIZE = FLSHA1CR0[FLSHSZ] × 1 KB
- FA2_SIZE = FLSHA1CR0[FLSHSZ] × 1 KB
- FB1_SIZE = FLSHA1CR0[FLSHSZ] × 1 KB
- FB2_SIZE = FLSHA1CR0[FLSHSZ] × 1 KB

When MCR2[SAMEDEVICEEN] = 0:

- FA1_SIZE = FLSHSZ × 1 KB
- FA2_SIZE = FLSHA2CR0[FLSHSZ] × 1 KB
- FB1_SIZE = FLSHB1CR0[FLSHSZ] × 1 KB
- FB2_SIZE = FLSHB2CR0[FLSHSZ] × 1 KB

Flash B1 and Flash B2 size settings are ignored in parallel mode (FLSHB1CR0[FLSHSZ], FLSHB2CR0[FLSHSZ]). To execute in parallel mode, Flash B1 should be the same device as Flash A1, and Flash B2 should be the same device as Flash A2.

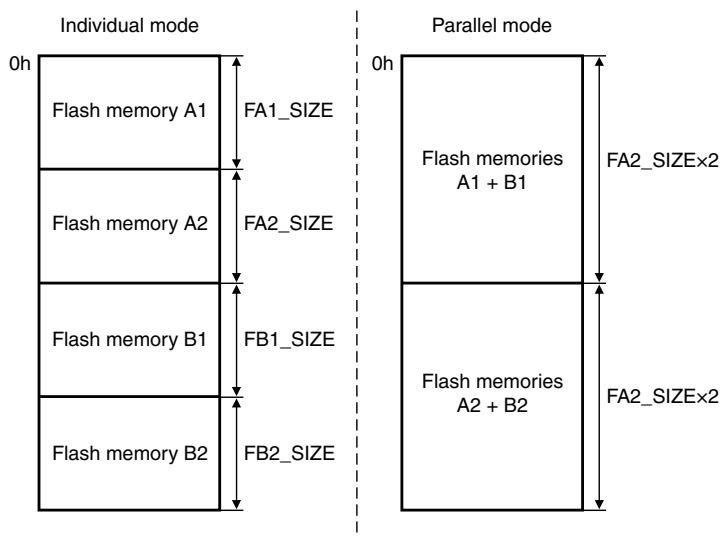


Figure 10-18. AHB access memory map in individual and parallel mode

AHB access memory map in individual mode:

- Flash A1 address range: 0000_0000h to FA1_SIZE
- Flash A2 address range: FA1_SIZE to (FA1_SIZE + FA2_SIZE)

- Flash B1 address range: (FA1_SIZE + FA2_SIZE) to (FA1_SIZE + FA2_SIZE + FB1_SIZE)
- Flash B2 address range: (FA1_SIZE + FA2_SIZE + FB1_SIZE) to (FA1_SIZE + FA2_SIZE + FB1_SIZE + FB2_SIZE)

AHB access memory map in parallel mode:

- Flash A1+B1 address range: 0000_0000h to FA1_SIZE × 2
- Flash A2+B2 address range: FA1_SIZE × 2 to (FA1_SIZE × 2 + FA2_SIZE × 2)

NOTE

The address used in Figure 10-18 and equations in this topic is the address presented to the memory. The base address for FlexSPI in the system memory map has already been removed, so it not used or shown here.

10.2.2.4.1 Dual image use case using HADDRSTART, HADDREND, and HADDROFFSET registers

The FlexSPI controller has a remap feature that supports the storage of dual images in memory. The HADDRSTART and HADDREND registers can be used to configure an address range to which an offset (that the HADDROFFSET register configures) is applied. This feature allows a single system bus address range to be mapped to two different memory ranges: the range that is remapped or the range that is not remapped. You can enable and disable the remapping to switch between accessing the two images at the same system memory address. This feature is used for booting.

NOTE

- Remapping in this table includes the REMAP register (HADDRSTART[0]) and SWAP register (HADDRSTART[2]).
- The swap function impacts both the AHB and IPS address remapping with the same mechanism. The remap function only impacts the AHB address.

Table 10-3. FlexSPI address translation

	AHB or IPS address	Memory address	Description
Enable remap			
AHB WR	ADDR	ADDR + OFFSET	The remap function is available if HADDRSTART <= ADDR < HADDREND, otherwise no change in address.
AHB RD	ADDR	ADDR + OFFSET	The remap function is available if HADDRSTART <= ADDR <HADDREND, otherwise no change in address.

Table continues on the next page...

Table 10-3. FlexSPI address translation (continued)

	AHB or IPS address	Memory address	Description
Disable remap			
AHB WR	ADDR	ADDR	No change in address.
AHB RD	ADDR	ADDR	No change in address.

NOTE

- HADDRSTART[0] is used to enable the remap feature.
- The ADDR above is the address from the IPS or AHB interface. The OFFSET is same as the HADDROFFSET register.

10.2.2.4.2 Flash address sent to flash memory devices

The AHB address (AHB command) or IPCR0[SFAR] (IP command) determines the flash memory access start address. See Flash memory access via AHB command and Flash memory access via IP command for more details.

For AHB commands, the FlexSPI controller removes the flash memory base address automatically when sending flash addresses to flash devices. The flash address is sent to devices in two parts: row address and column address. For flash devices that do not support the column address, set FLSHxCR1[CAS] to 0. This setting causes all flash address bits to be sent to flash devices as row addresses.

For word-addressable flash devices, the last bit of the address is not needed, because flash memory is read and programmed in terms of two bytes. For parallel mode, Flash A1 and Flash B1 (or Flash A2 and Flash B2) are accessed in parallel. As a result, the flash memory address sent to flash devices should be divided by 2. Table 10-4 indicates the relationship of row address, column address, and flash address (FA).

Table 10-4. Flash address, row address, and column address

Parallel mode	Word-addressable	Row address	Column address	Comment
0	0	FA[31:CAS]	FA[CAS-1:0]	There is no limitation on FA and data size alignment.
0	1	FA[31:CAS+1]	FA[CAS:1]	FA and data size should be two-byte aligned.
1	0	FA[31:CAS+1]	FA[CAS:1]	FA and data size should be two-byte aligned.
1	1	FA[31:CAS+2]	FA[CAS+1:2]	FA and data size should be four-byte aligned.

NOTE

- FA is the flash address without base address.
- If the bit number of the row or column flash address is more than the valid value, the high position bits are supplemented with zero. See Programmable sequence engine for details about row or column address instructions.

When parallel mode is enabled or word-addressable flash memory is used, there are limitations on the flash memory start address and data size. You can address this requirement by aligning the AHB bus access address (for AHB command) or IP command address IPCR0[SFAR] (for IP command) in software. There are two ways to avoid these limitations in the specified case.

1. For AHB Read Command only:

   When AHBCR[READADDROPT] and AHBCR[PREFETCHEN] are both 1, FlexSPI guarantees the start address and data size for the flash memory access are eight-byte aligned by hardware.

   When AHBCR[READADDROPT] is 1, FlexSPI fetches redundant data to guarantee the flash memory start address is eight-byte aligned. When prefetch is enabled (AHBCR[PREFETCHEN] is 1), the size of the eight-byte-aligned AHB receive buffer determines the flash memory read data size.

2. For AHB Write Command and Individual Mode only:

   By default, FlexSPI guarantees the flash memory write access start address and data size are two-byte aligned when using DQS as write mask (FLSHCR4[WMENB or WMENA] = 1).

   This feature is not applied in parallel mode and must be used if the external device supports write mask feature.

   For example, for odd start address:
   - To write 6 bytes, FlexSPI sends 8 bytes, the first and last byte are both masked.
   - To write 7 bytes, FlexSPI sends 8 bytes, only the first byte is masked.
   - To write 8 bytes, FlexSPI sends 10 bytes, the first and last byte are both masked.

   For even start address:
   - To write 6 bytes, FlexSPI sends 6 bytes, no bytes are masked.
   - To write 7 bytes, FlexSPI sends 8 bytes, last byte is masked.
   - To write 8 bytes, FlexSPI sends 8 bytes, no bytes are masked.

10.2.2.5 Lookup table (LUT)

The LUT is an internal memory that preserves a number of preprogrammed sequences. Each sequence consists of up to eight instructions which are executed sequentially. When an IP command or an AHB command triggers a flash memory access, the FlexSPI controller:

1. Fetches the sequence from LUT (sequence index or number).
2. Executes the flash memory access to generate a valid flash transaction on the SPI interface.

Figure 10-19 shows the LUT structures, sequences, and instructions.

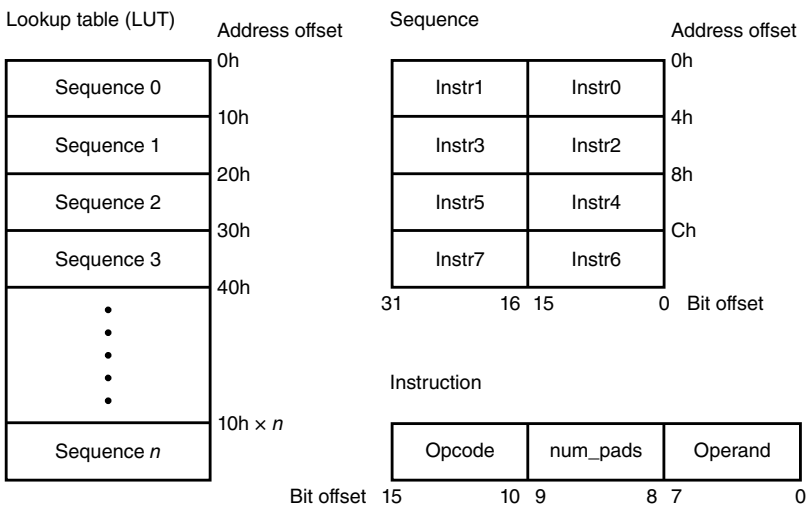


Figure 10-19. LUT structure, sequences, and instructions

NOTE

If the number of instructions needed for a flash transaction is less than eight, the STOP instruction (code 0000_0000h) must be programmed for unneeded instructions.

For IP and AHB write commands, the FlexSPI controller always executes from instruction pointer 0. For AHB read commands, the FlexSPI controller executes from a saved instruction start pointer. The FlexSPI controller saves the instruction start pointers separately for each flash device. All these saved instruction pointers are zero before the JMP_ON_CS instruction is executed. When the JMP_ON_CS instruction is executed, the operand in the JMP_ON_CS instruction is saved as instruction start pointer. See Execute-In-Place (XIP) enhanced mode.

The reset value of the LUT is unknown because it is implemented as internal memory. The LUT must be programmed according to the connected device. To protect its contents during a code runover, the LUT can be locked or unlocked to prevent unwanted changes after the LUT has been configured. The key to lock or unlock the LUT is **5AF05AF0h**.

To lock the LUT:

1. Write the key (5AF05AF0h) to LUT Key (LUTKEY).
2. Write 1 to LUTCR[LOCK] and write 0 to LUTCR[UNLOCK]. When there is another register write access to FlexSPI between these two write accesses, this LUT is not successfully locked.

To unlock the LUT:

1. Write the key (5AF05AF0h) to the LUT Key Register LUT Key (LUTKEY).
2. Write 0 to LUTCR[LOCK] and write 1 to LUTCR[UNLOCK]. When there is another register write access to FlexSPI between these two write accesses, this LUT is not successfully locked.

The lock status of the LUT can be read from LUTCR[LOCK] and LUTCR[UNLOCK].

10.2.2.6 Programmable sequence engine

The FlexSPI controller implements a programmable sequence engine that executes the sequence from the LUT. The FlexSPI controller executes the instructions sequentially, and generates flash transactions on the SPI interface accordingly. Table 10-5 is a complete list of the supported instructions.

Table 10-5. Instruction set

Name	Opcode	num_pads	Action on SPI interface	Transmit data	Bits/bytes/cycle number
CMD_SDR	00_0001h	00h - one pad (Single mode) 01h - two pads (Dual mode) 02h - four pads (Quad mode) 03h - eight pads (Octal mode)	Transmit command code to flash memory	Command code: Operand[7:0]	Bit number: 8
CMD_DDR	00_0021h				
RADDR_SDR	00_0002h		Transmit row address to flash memory See Flash address sent to flash memory devices	Row_Address[31:0] Row_Address comes from AHB bus (AHB command) or IPCR0[SFAR] (IP command).	Bit number: operand[7:0] Value of operand determines number of bits sent in Row_Address.
RADDR_DDR	00_0022h				
CADDR_SDR	00_0003h		Transmit column address to flash memory. See Flash address sent to	Column_Address[31:0]	Bit number: operand[7:0] Value of operand determines number of bits sent in Column_Address.
CADDR_DDR	00_0023h				

Table continues on the next page...

Table 10-5. Instruction set (continued)

Name	Opcode	num_pads	Action on SPI interface	Transmit data	Bits/bytes/cycle number
			flash memory devices.	Column_Address comes from AHB bus (AHB command) or IPCR0[SFAR] (IP command).	
MODE1_SDR	00_0004h		Transmit mode bits to flash memory	Mode bits: Operand[0]	Bit number: 1
MODE1_DDR	00_0024h				
MODE2_SDR	00_0005h			Mode bits: Operand[1:0]	Bit number: 2
MODE2_DDR	00_0025h				
MODE4_SDR	00_0006h			Mode bits: Operand[3:0]	Bit number: 4
MODE4_DDR	00_0026h				
MODE8_SDR	00_0007h			Mode bits: Operand[7:0]	Bit number: 8
MODE8_DDR	00_0027h				
WRITE_SDR	00_0008h		Transmit program data to flash device	Program data in IP_TX_FIFO or AHB_TX_BUF	AHB burst size and burst type (AHB command) or IPCR1[IDATSZ] (IP command) determines byte number (data size). For details about flash read or program data size, see Flash memory access via AHB command and Flash memory access via IP command.
WRITE_DDR	00_0028h				
READ_SDR	00_0009h		Receive read data from flash device Read data is put into AHB_RX_BUF or IP_RX_FIFO.	-	
READ_DDR	00_0029h				
DATSZ_SDR	00_000Bh		Transmit read or program data size (byte number) to flash device	Internal logic Read or program data size for current command sequence	Bit number: operand[7:0] Never set operand to zero or greater than 64 for DATSZ instruction.
DATSZ_DDR	00_002Bh				
DUMMY_SDR	00_000Ch		Leave data lines undriven by FlexSPI controller. Turnaround cycles are provided from host driving to device driving. num_pads determines number of pads in input mode.	-	Dummy cycle number (in serial root clock): Operand[7:0] Dummy cycle (N), described in data sheet of flash device, is in number of SCLK cycles. This number may be configurable. In SDR mode, SCLK cycle is same as serial root clock. Operand value must be set to N. In DDR mode, SCLK cycle is double serial root clock cycle. Operand value must be set to 2N, 2N-1 or 2N+1 depending on definition of dummy in data sheet of flash device. See Flash memory access
DUMMY_DDR	00_002Ch				

Table continues on the next page...

Table 10-5. Instruction set (continued)

Name	Opcode	num_pads	Action on SPI interface	Transmit data	Bits/bytes/cycle number
					sequence examples and dummy cycle definition in data sheet of external memory.
DUMMY_RWDS_SDR	00_000Dh		Similar to DUMMY_SDR/ DUMMY_DDR instruction. Difference is in dummy cycle number. DQS pin is called RWDS in HyperBus specification. See Dummy instruction for details. **Set operand to "Latency count" for HyperBus devices.**	-	For read command, dummy cycle number (in serial root clock): (operand[7:0] × 4 - 1) if RWDS (DQS pin) is high; (operand[7:0] × 2 - 1) if RWDS (DQS pin) is low; For write command, dummy cycle number (in serial root clock): (operand[7:0] × 4 - 2) if RWDS (DQS pin) is high; (operand[7:0] × 2 - 2) if RWDS (DQS pin) is low;
DUMMY_RWDS_DDR	00_002Dh				
LEARN_SDR	00_000Ah	00h - one pad (Single mode) 01h - two pads (Dual mode) 02h - four pads (Quad mode) 03h - eight pads (Octal mode)	Receive read data or preamble bit from flash device FlexSPI controller compares data line bits with the DLPR register to determine a correct sampling clock phase.	-	Bit number: operand[7:0] Never set operand to zero for LEARN instruction. Value of operand indicates number of bits to receive and compare to DLPR value. For example, 8-bit pattern 5Ah on each data line must set operand to 8.
LEARN_DDR	00_002Ah				
JMP_ON_CS	00_001Fh	Num_pads setting ignored.	Stop execution, deassert CS and save operand[7:0] as instruction start pointer for next sequence. Normally this instruction is used to support Execute-In-Place enhanced mode. See Execute-In-Place (XIP) enhanced mode. This instruction is only allowed for AHB read commands. When using this instruction in IP command or AHB write command,	-	No transaction on SPI interface.

Table continues on the next page...

Table 10-5. Instruction set (continued)

Name	Opcode	num_pads	Action on SPI interface	Transmit data	Bits/bytes/cycle number
			interrupt status flag set (INTR[IPCMDERR] or INTR[AHBCMDERR]).		
STOP	00_0000h		Stop execution and deassert CS. Next command sequence (to same flash device) starts from instruction pointer 0.	-	

The programmable sequence engine allows configuration of the LUT according to the connected external serial device. The flexible LUT structure easily adapts to new command or protocol changes from different vendors.

The DDR sequence is a flash memory access sequence that contains DDR instructions other than DUMMY_DDR, and may contain SDR instructions. The output and input timing on FlexSPI differs for SDR and DDR sequences. When included as part of a DDR sequence, SDR instructions execute differently from SDR instructions in an SDR sequence. See FlexSPI input timing and FlexSPI output timing.

10.2.2.6.1 Executing instructions on SPI interface

This section describes the execution of instructions on the SPI interface. For all instructions that receive bits from or transmit bits to flash devices:

- The bit order in one byte is higher on DATA3 than DATA0.
- The bit order is higher on port B than port A.

This order is shown in Figure 10-20.

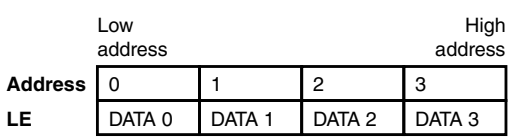


Figure 10-20. Bus bit order

Table 10-6. SPI instructions

Instructions	Description
Command (CMD_SDR, CMD_DDR)	Normally used to transmit a command code to the external device. The command code is the 8-bit operand in the instructions. Command code is sent to both port A and port B in parallel mode. See Flash memory access sequence examples.
Address (RADDR_SDR, RADDR_DDR, CADDR_SDR, CADDR_DDR)	Normally used to send the flash memory access start address (row address or column address) to an external device. FlexSPI determines the bits of the row address or column address according to the AHB access address or IP command address. See Flash address sent to flash memory devices. The operand value in the instruction code represents the number of address bits to send. The bits in the row address or column address are sent to both port A and port B in parallel mode. For example, when NUM_PADSn = 8, a memory reading the flash memory address on four SCK edges requires the sum of bits in CADDR and RADDR instructions to be 32. Otherwise, FlexSPI does not generate four SCK edges of address phase. See Flash memory access sequence examples.
Mode (MODEx_SDR, MODEx_DDR)	Normally used to send mode bits to external devices. Mode bits are the lower bits of an operand. For example, the bit number is 1 for MODE1_SDR and MODE1_DDR, 2 for MODE2_SDR and MODE2_DDR, 4 for MODE4_SDR and MODE4_DDR, and 8 for MODE8_SDR and MODE8_DDR. The pad number should not be greater than the mode bit number. For example, you cannot set NUM_PADSn to 11b (Octal mode) for MODE4_* instructions. Mode bits are sent to both port A and port B in parallel mode. See Flash memory access sequence examples for more details.
Data Size (DATSZ_SDR, DATSZ_DDR)	Used to send the size of program data or read data (byte number) to external devices. This instruction is normally used in FPGA applications where the memory space in external device acts like a FIFO. The external device requires data size information to determine how much data are popped from or pushed into the internal FIFO. The operation value in the data size instructions is the bit number. Data size is sent to both port A and port B in parallel mode. See Flash memory access sequence examples.
Write (WRITE_SDR, WRITE_DDR)	Normally used to send program data to external device. Programming data is fetched from IP_TX_FIFO (IP Command) or AHB_TX_Buffer (AHB command). For details about flash program data size, see Flash memory access via AHB command and Flash memory access via IP command. The byte order for program data is always from low to high. Odd bytes are sent on port A and even bytes are sent on port B in parallel mode. See Flash memory access sequence examples for more details.
Read (READ_SDR, READ_DDR)	Normally used to receive flash memory data from external devices. Received data is put into IP_RX_FIFO (IP Command) or AHB_RX_Buffer (AHB command). For information about flash memory read data size, see Flash memory access via AHB command and Flash memory access via IP command. The byte order for reading data is always from low to high. Odd bytes are received from port A and even bytes are received from port B in parallel mode. See Flash memory access sequence examples.
Dummy (DUMMY_SDR, DUMMY_DDR, DUMMY_RWDS_SDR, DUMMY_RWDS_DDR)	Used to provide turnaround cycles on the SPI interface. During dummy instruction, the FlexSPI controller and external devices do not drive the SPI interface. See Programmable sequence engine for details about dummy cycle number. DUMMY_RWDS_DDR can be used for a HyperBus device that uses RWDS pin to indicate whether extra latency is needed. DUMMY_RWDS_SDR is reserved. The FlexSPI controller checks the DQS pin input level at the fourth cycle after SCLK output toggling is enabled. The DQS pin is called RWDS in HyperBus specification. See Flash memory access sequence examples. **NOTE:** FlexSPI releases the bus after at least one cycle. To avoid data contention, the NUM_PADSn value for the dummy commands should be configured to match the number of data lines used for the external memory.

Table continues on the next page...

Table 10-6. SPI instructions (continued)

Instructions	Description
Learn (LEARN_SDR, LEARN_DDR)	Used to determine the correct sample clock phase for flash memory read data sampling. External device drives read data (or data learning pattern) bits on the FlexSPI interface. The FlexSPI controller compares the data line bits to the DLPR register to determine a correct sampling clock phase. Clock phase selection is automatically updated after executing learn instructions. See Data learning for more details. The operand value in learn instructions is the byte number. FlexSPI checks the same data pattern on each data line. The byte order is byte 0, byte 1, byte 2, byte 3, byte0, byte 1, and so on. • Byte 0 is DLPR register bits 7–0. • Byte 1 is DLPR register bits 15–8. • Byte 2 is DLPR register bits 23–16. • Byte 3 is DLPR register bit 31–24. The bit order is from high to low in each byte. See Flash memory access sequence examples.

10.2.2.6.2 Flash memory access sequence examples

Diagrams below all assume MCR0[SERCLKDIV] = 0.

Figure 10-21 shows an example SDR single I/O read sequence (Cypress Serial Nor Flash S25FS512S) in individual mode.

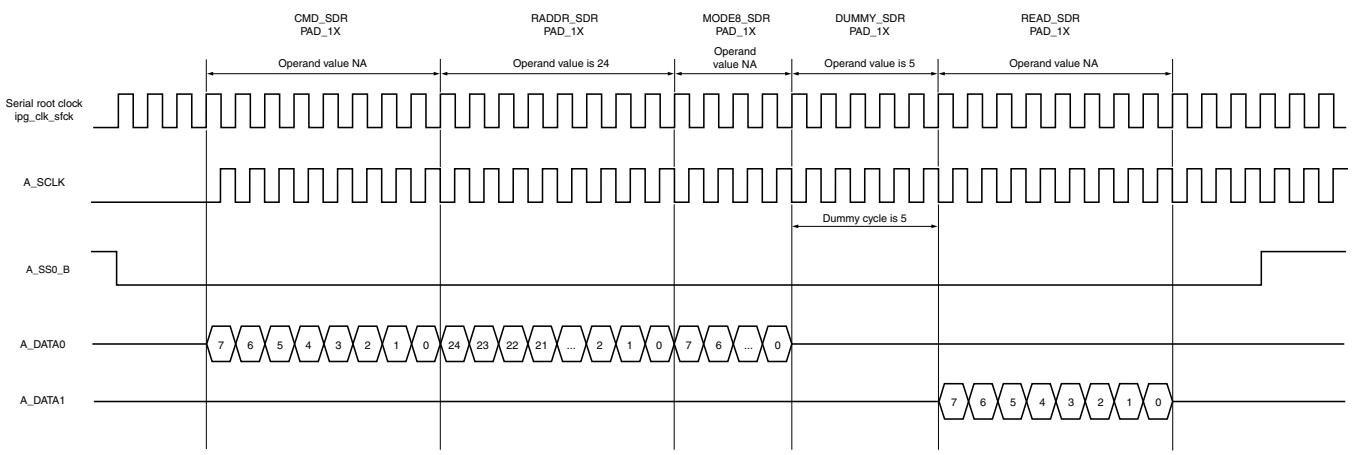


Figure 10-21. Flash memory access sequence example (SDR single I/O read sequence)

NOTE

- FlexSPI dummy instruction starts and ends at the rising edge of serial root clock.
- Device dummy cycle starts and ends at the falling edge of SCLK (on Cypress S25FS512S data sheet).

Figure 10-22 shows an example SDR quad I/O read sequence (Cypress Serial Nor Flash S25FS512S) in individual mode.

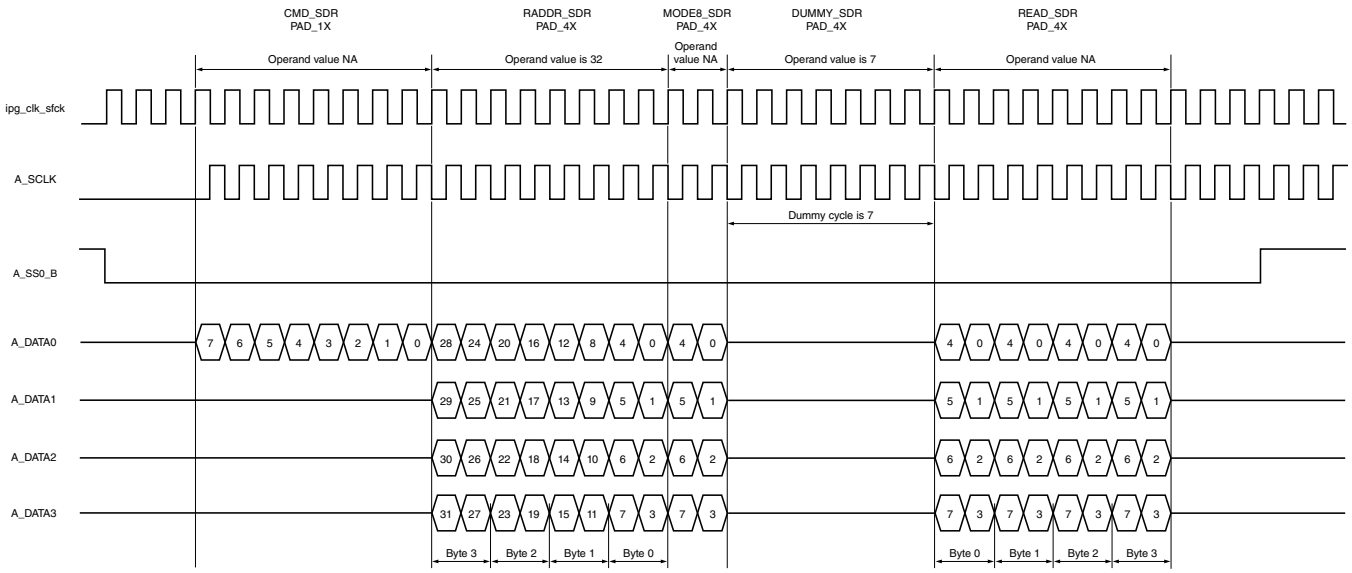


Figure 10-22. Flash memory access sequence example (SDR quad I/O read sequence)

NOTE

- FlexSPI dummy instruction starts and ends at the rising edge of serial root clock.
- Device dummy cycle starts and ends at the falling edge of SCLK (on Cypress S25FS512S data sheet).

Figure 10-23 shows an example DDR quad I/O read sequence (Cypress Serial Nor Flash S25FS512S) in parallel mode.

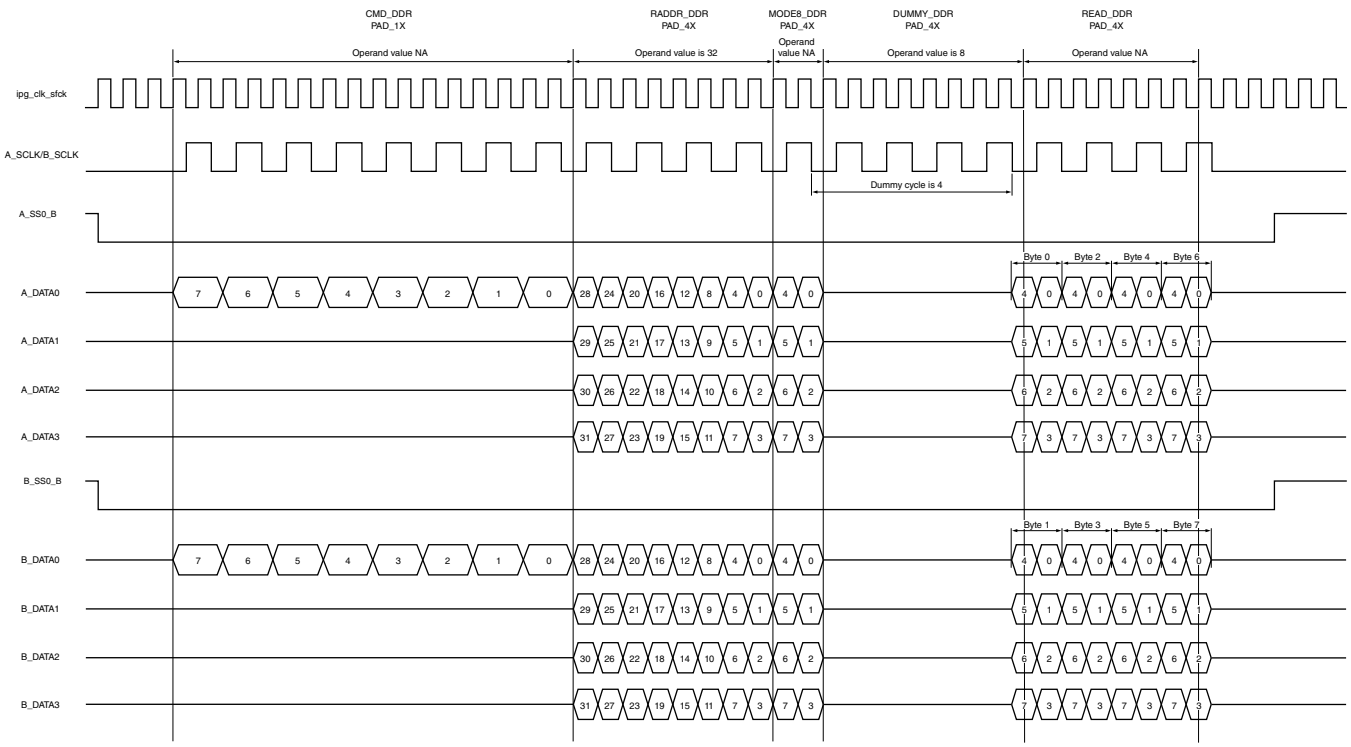


Figure 10-23. Flash memory access sequence example (DDR quad I/O read sequence)

Figure 10-24 shows an example learn instruction (not for a specified flash device) in individual mode.

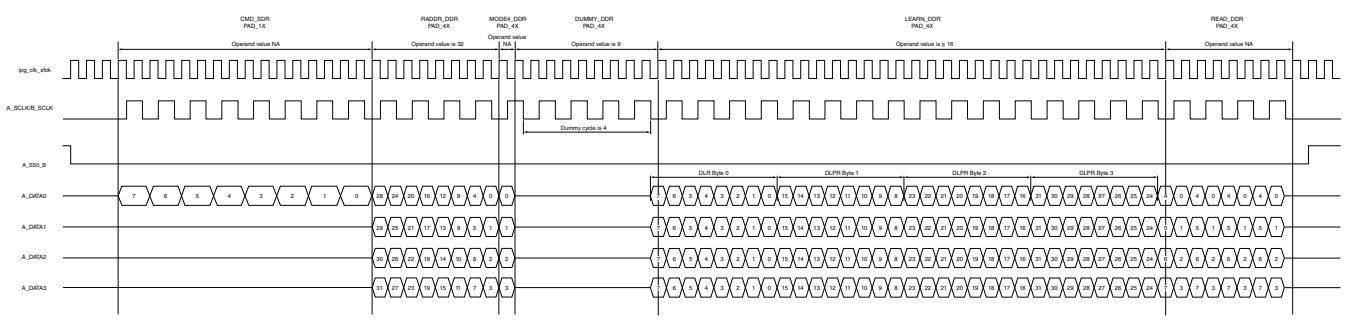


Figure 10-24. Flash memory access sequence example (data learning)

NOTE

- FlexSPI dummy instruction starts and ends at the rising edge of serial root clock.
- Device dummy cycle starts and ends at the falling edge of SCLK.
- The operand value of DUMMY_DDR instruction is odd because the total cycle number before DUMMY_DDR cycle is odd.

Figure 10-25 shows an example HyperBus device read transaction (single latency count) in individual mode.

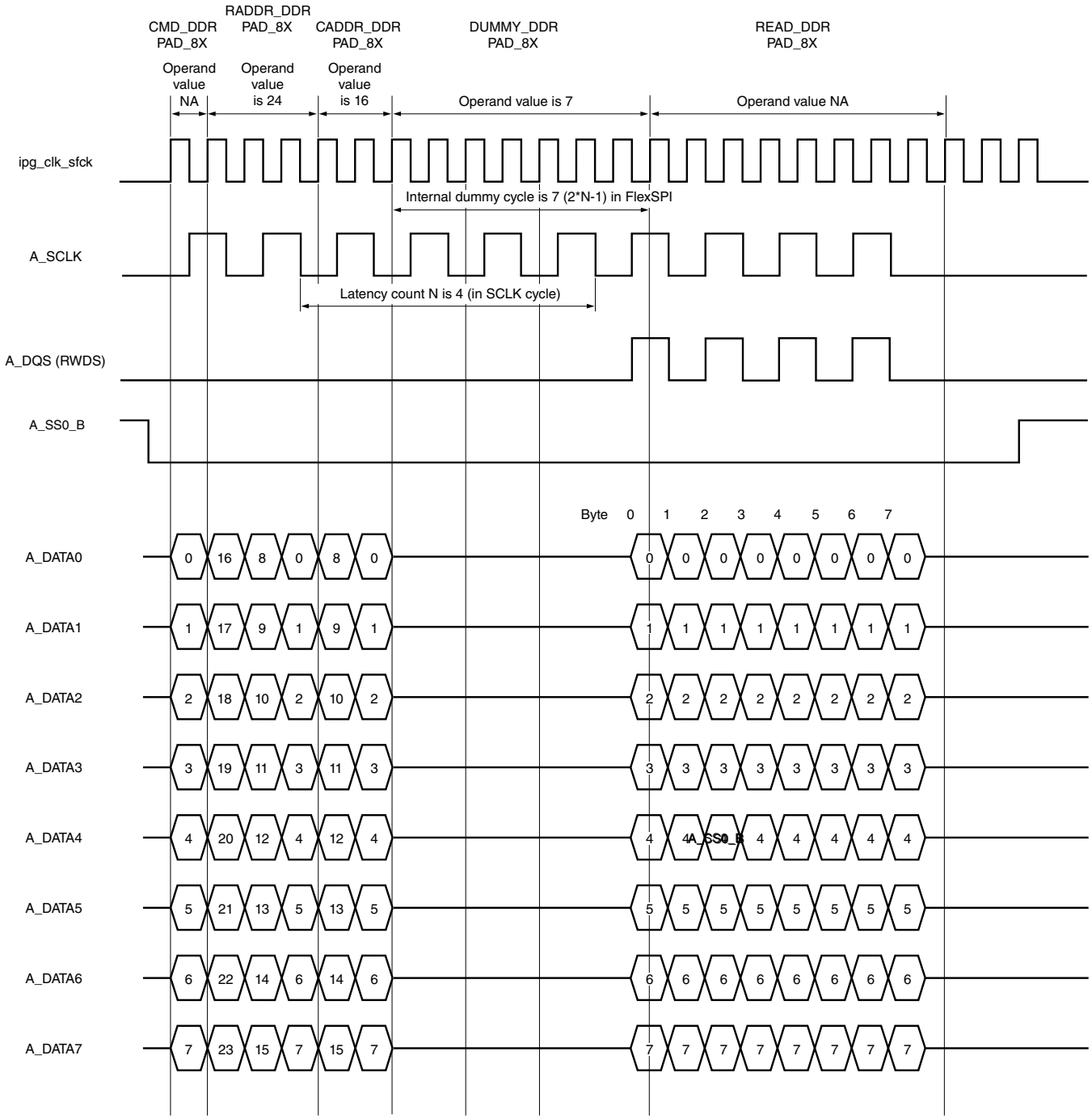


Figure 10-25. HyperBus device read transaction with single latency count

NOTE

FlexSPI continues to drive the clock until it has latched all of the read data it is expecting. On-chip delay for the data to reach FlexSPI can increase the number of SCK clock pulses.

Figure 10-26 shows an example HyperBus device read transaction (additional latency count) in individual mode.

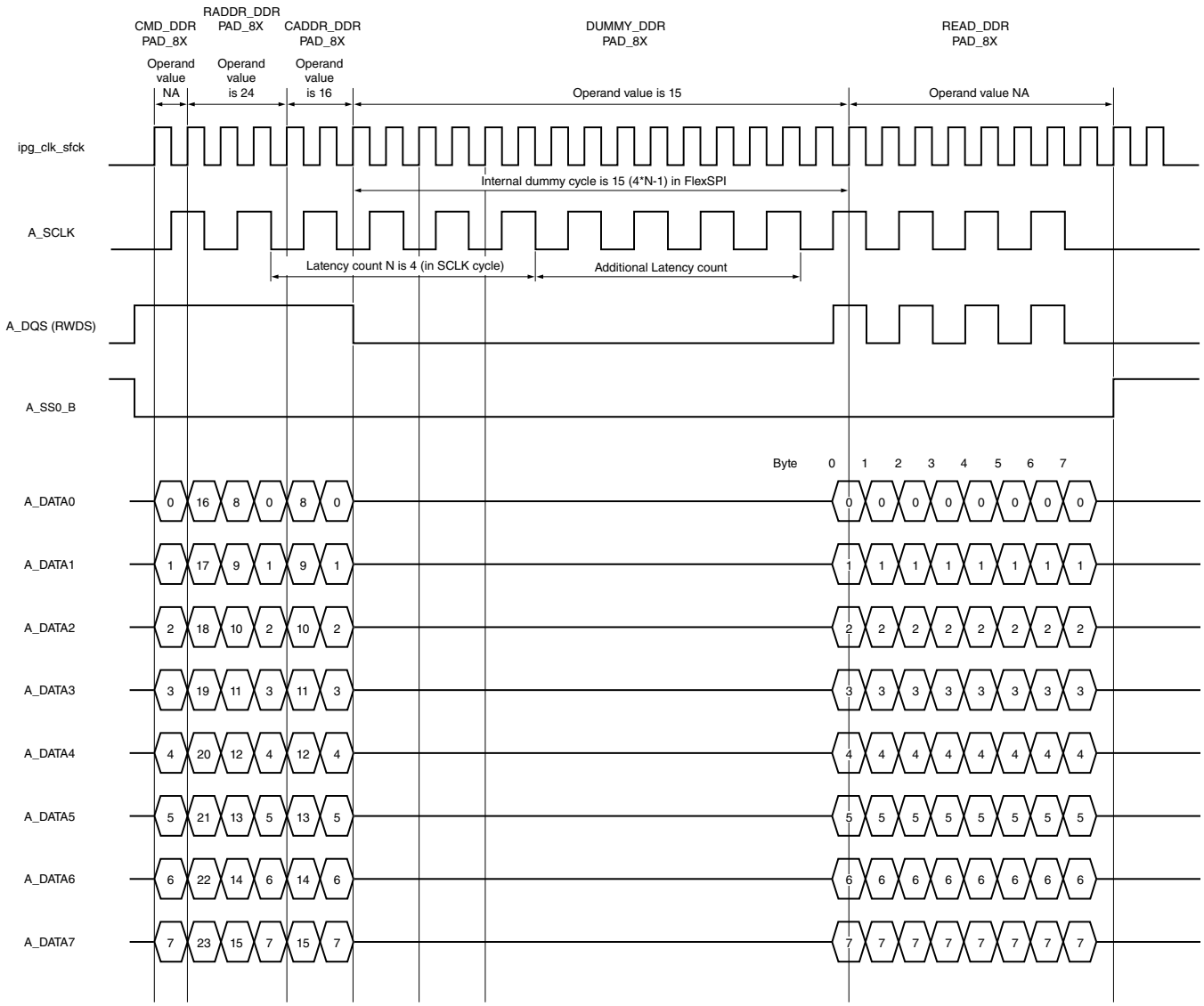


Figure 10-26. HyperBus device read transaction with additional latency count

Figure 10-27 shows an example HyperBus device write transaction (single latency count) in individual mode.

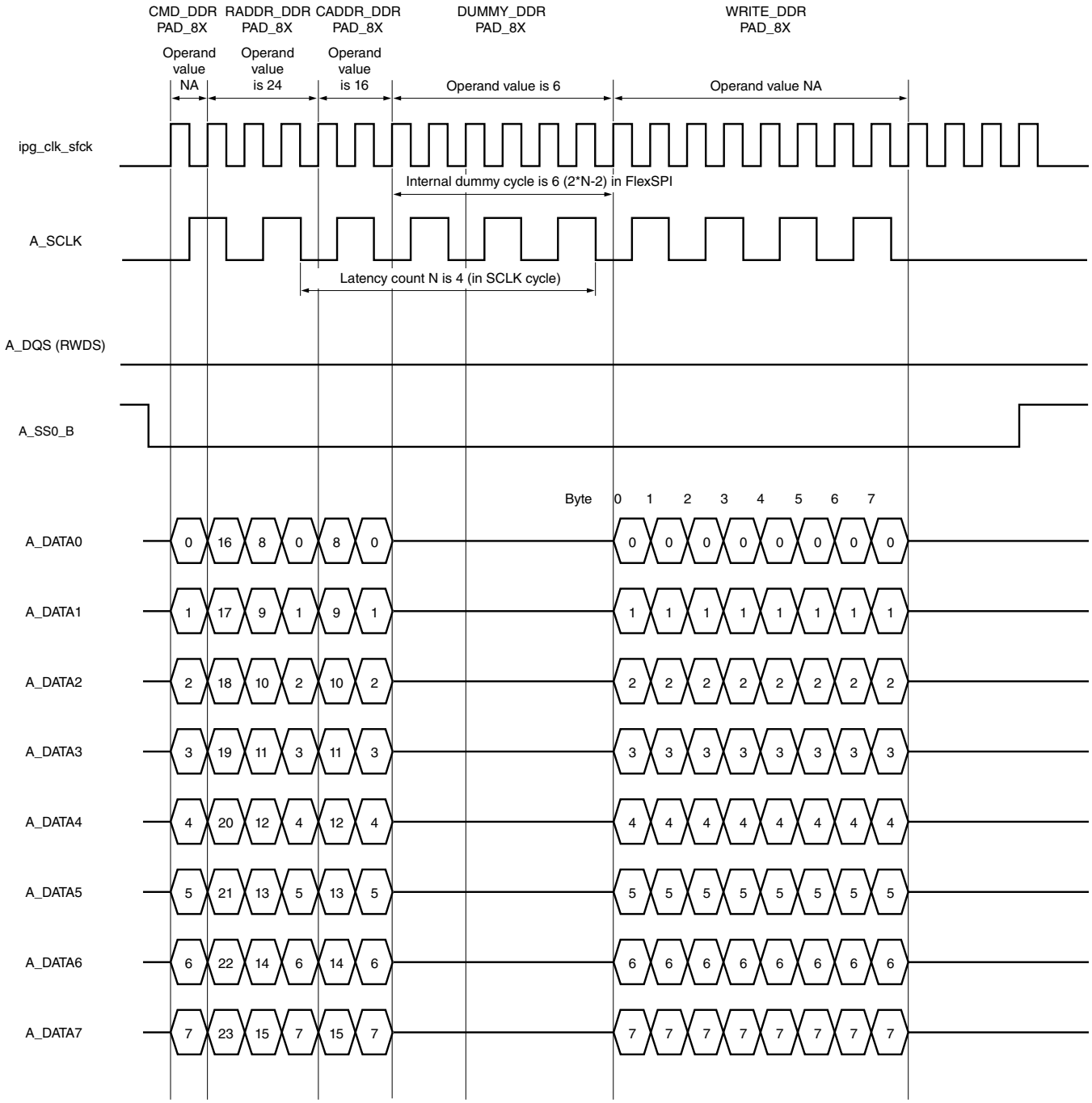


Figure 10-27. HyperBus device write transaction with single latency count

Figure 10-28 shows an example HyperBus device write transaction (additional latency count) in individual mode.

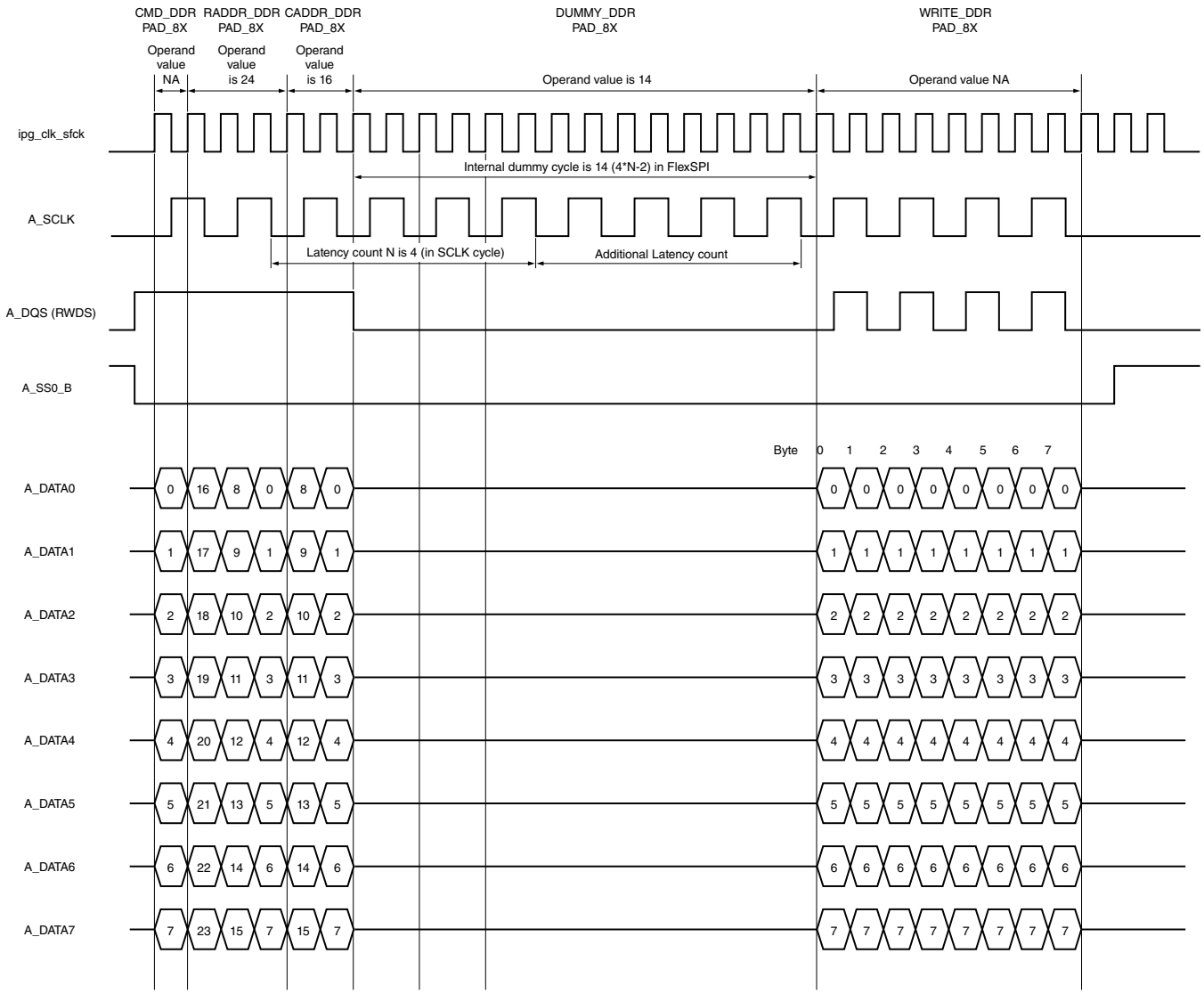


Figure 10-28. HyperBus device write transaction with additional latency count

10.2.2.7 Clocking

This section describes FlexSPI clocks and special clocking requirements.

Table 10-7. Clock usage

Clock name	Description	Comment
serial clock root (ipg_clk_sfck)	Root clock for Serial domain	FlexSPI core clock, used to generate the serial clock output. The clock is optionally divided using MCR0[SERCLKDIV] to obtain the SCLK.

Table continues on the next page...

Table 10-7. Clock usage (continued)

Clock name	Description	Comment
		Its frequency is same as SPI SCK frequency in SDR mode if MCR0[SERCLKDIV] = 0.
ahb clock (hclk)	AHB bus clock	hclk frequency is higher than 1/4 of serial root clock, so internal async FIFO is never full. This clock is used for internal logic in FlexSPI.
ipg clock (ipg_clk)	IPS bus clock	Register access clock. ipg_clk must be equal to hclk or an integer division of hclk. For example, ipg_clk = hclk/2.
SCLK	FlexSPI serial clock. Output clock on SCLK pin	Half clock frequency of serial clock root in DDR mode, and same frequency as serial clock root in SDR mode. Clock output toggles during the entire flash memory access sequence.
DQS_OUT	Dummy Read Strobe output	Same frequency as SCLK. Clock output toggles during READ and LEARN instructions. The DQS ensures that the data can be strobed correctly. It is only used for data strobe, instead of command or address strobe. The data includes useful read data and data learning patterns.
DQS_IN	Sample clock for receive data	Same frequency as SCLK. Sample clock comes from looped-back dummy read strobe, looped-back SCLK, or flash-memory-provided read strobe.

10.2.2.8 Interrupts

Table 10-8 describes all interrupts that FlexSPI generates.

Table 10-8. Flash memory address, row address, and column address

Interrupt	Enable	Condition
IP command done	INTEN[IPCMDDONEEN]	An IP command is finished.
IP command grant error	INTEN[IPCMDGEEN]	An IP command grant timeout occurs (bus not granted after MCR0[IPGRANTWAIT] × 1024 AHB clock cycles). See Overview of error categories, flags, and triggered sources.
AHB command grant error	INTEN[AHBCMDGEEN]	An AHB command grant timeout occurs (bus not granted after MCR0[AHBGRANTWAIT] × 1024 AHB clock cycles). See Overview of error categories, flags, and triggered sources.

Table continues on the next page...

Table 10-8. Flash memory address, row address, and column address (continued)

IP command error	INTEN[IPCMDERREN]	A command check error or a command execution error for an IP command occurs. See Overview of error categories, flags, and triggered sources.
AHB command error	INTEN[AHBCMDERREN]	A command check error or a command execution error for AHB command occurs. See Overview of error categories, flags, and triggered sources.
IP_RX_FIFO watermark available	INTEN[IPRXWAEN]	The fill level of IP_RX_FIFO reaches the watermark level (IPRXFCR[RXWMRK]).
IP_TX_FIFO watermark empty	INTEN[IPTXWEEN]	The empty level of IP_TX_FIFO reaches the watermark level (IPTXFCR[TXWMRK]).
Data learning failed	INTEN[DATALEARNFAILEN]	No valid sample clock phase found after LEARN instruction is executed. See Overview of error categories, flags, and triggered sources.
Sequence execution timeout	INTEN[SEQTIMEOUTEN]	A sequence execution time exceeds the timeout wait time (MCR1[SEQWAIT]). For example, the following flash memory read command sequence lasts about 800_0000h cycles (ipg_clk_sfck). If SEQWAIT is set to FFFFh, a sequence timeout interrupt is generated. • IP command triggers timeout. • Flash memory read data size is 100_0000h bytes. • Flash memory accesses in single mode and SDR mode.
AHB bus timeout	INTEN[AHBBUSTIMEOUTEN]	AHB bus response timeout occurs. For example, AHB read sequence is not configured properly in LUT (such as without a READ instruction). No data is read from an external device. As a result, FlexSPI cannot reach the read data in AHB_RX_Buffers for the AHB read command. See Overview of error categories, flags, and triggered sources.
Write command stops SCLK	INTEN[SCKSTOPBYWREN]	IP_TX_FIFO is empty while executing a write command sequence. FlexSPI stops SCLK output clock toggling and waits for write data filling.
Read command stops SCLK	INTEN[SCKSTOPBYRDEN]	IP_RX_FIFO is full while executing a read command sequence. FlexSPI stops SCLK output clock toggling and waits to read the data from IP_RX_FIFO.

10.2.2.9 Flash memory access via IP command

Follow these steps to trigger a flash memory access via IP command.

1. If this command is a programing command, fill the IP transmit FIFO with programming data (for instance, programming flash data and flash memory status registers.)
2. Write the flash memory access start address to IPCR0[SFAR].
3. Write the read or program data size to IPCR1[IDATSZ]. Write the sequence index to IPCR1[ISEQID]. Write the sequence number to IPCR1[ISEQNUM].
4. Write 1 to IPCMD[TRG] to trigger a flash memory access command.
5. Wait for the INTR[IPCMDDONE] flag to set or for the IP command done interrupt to fire, indicating the command has completed on the FlexSPI interface.

NOTE

- The IP transmit FIFO can be filled before or after writing the IPCR0, IPCR1, and IPCMD registers. If the SFM command starts with the IP transmit FIFO empty, FlexSPI stops SCLK toggling to wait for transmit data ready automatically.
- The IPCMD register must be written after writing the IPCR0 and IPCR1 registers.
- One IP command can issue up to eight command sequences.
- You cannot issue another IP command before the previous IP command is finished. Performing this action results in unknown behavior.

If this command is a read command, all read data from flash memory is put into the IP receive FIFO. Software must read data from the IP receive FIFO via AHB bus or IP bus. When the IP receive FIFO is full and the flash device still contains data to read, FlexSPI stops SCLK toggling until the FIFO has free space. See SCLK stop.

A triggered serial flash command contains:

- A flash memory access start address. IPCR0[SFAR] determines this address.
- A flash chip select. The flash memory access address and flash memory size setting (FLSHxCR0[FLSHSZ]) determine the chip select.
- A flash command sequence index and sequence number. FlexSPI sequentially executes the sequences indexed from IPCR1[ISEQID] to (IPCR1[ISEQID] + IPCR1[ISEQNUM]) in LUT.
- Flash individual/parallel memory access mode:

Determined by IPCR1[IPAREN].

- The flash memory read or program data size, in bytes.
  - If the value of IPCR1[IDATSZ] is nonzero, the data size is IPCR1[IDATSZ].
  - If the value of IPCR1[IDATSZ] is zero, the operand value in the read or write instruction determines the data size.

NOTE

- Software must ensure that the last sequence index never exceeds the LUT sequence number. That is, IPCR1[ISEQID] + IPCR1[ISEQNUM] must be less than 32.
- For sequence numbers greater than one, the data size is applied to every command sequence.
- If there is no write or read instruction in the command sequence, the data size is ignored.

The IP command request is sent to the arbitrator after software triggers it. It is not executed on the FlexSPI interface until the arbitrator grants it. See Command arbitration.

10.2.2.9.1 Reading data from IP receive FIFO

FlexSPI puts read data from the external device into the IP receive FIFO for IP commands.

The data stored in the IP receive FIFO can be read using IPS and register or system and AHB bus transactions:

- 100h–17Ch (by IPS bus)
- 34000000h–34000200h (by AHB bus)

If MCR0[ARDFEN] = 1, only the AHB bus can read the read data in the IP receive FIFO. Read access to the IP receive FIFO using IP bus always returns with data zero, and no bus error occurs.

If MCR0[ARDFEN] = 0, only the IPS bus can read the read data in the IP receive FIFO. Read access to the IP receive FIFO using the AHB bus triggers a bus error.

FlexSPI pushes read data into the IP receive FIFO in sets of 64 bits each time it receives 64 bits of data from an external device. When the number of read data bits is not 64-bit-aligned, FlexSPI pushes additional zero bits into the IP receive FIFO for the last push.

The processor or DMA can read the IP receive FIFO.

10.2.2.9.1.1 Reading via processor

To read data from the IP receive FIFO via processor, configure the items below.

- Write 0 to IPRXFCR[RXDMAEN].
- Write the watermark level to IPRXFCR[RXWMRK]. The watermark level is (IPRXFCR[RXWMRK] + 1) × 8 bytes.
- Write 1 to INTEN[IPRXWAEN] to enable the IP receive FIFO watermark available interrupt (optional).

The processor must poll INTR[IPRXWA] or wait for the IP receive FIFO watermark available interrupt before reading data from the FIFO. Waiting ensures that a watermark level amount of data is filled in the IP receive FIFO before reading.

After reading a watermark level amount of data from the FIFO, software must set the INTR[IPRXWA] flag to pop that data from the FIFO.

Figure 10-29 shows the reading flow from the IP receive FIFO via processor.

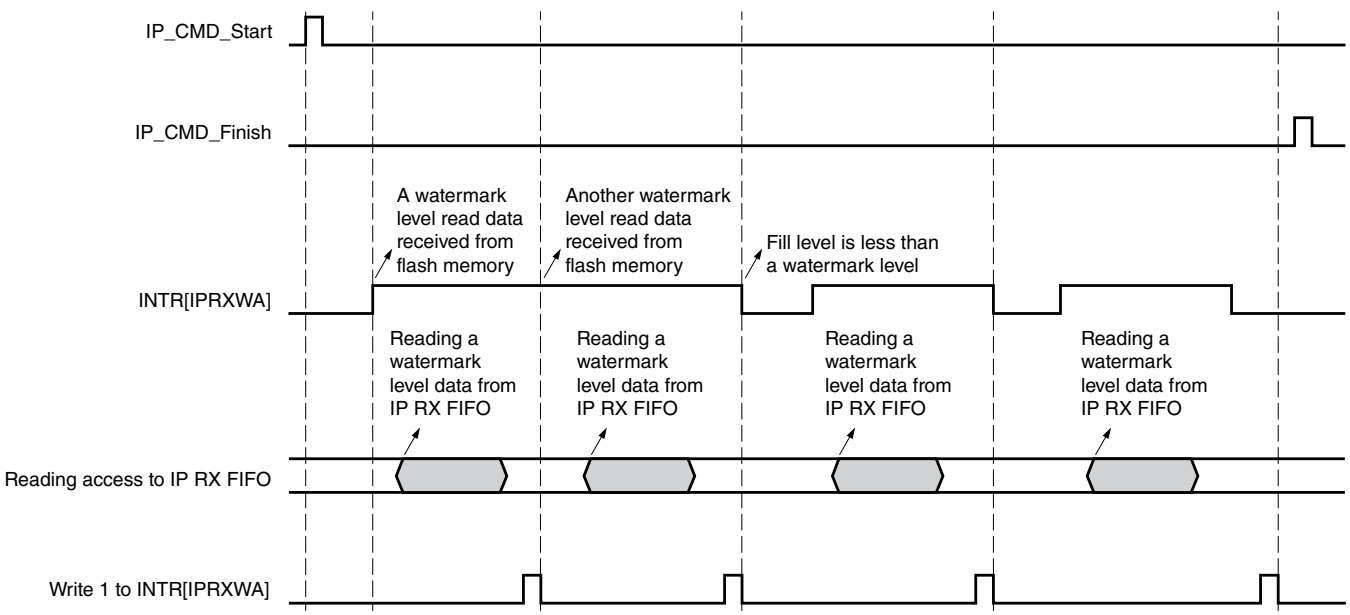


Figure 10-29. Reading IP receive FIFO via processor

NOTE

- The processor must read a watermark level of data from the IP receive FIFO before setting the INTR[IPRXWA] flag.
- The total flash read or program data size is not required to be a multiple of the watermark level. When the reading data size from the FIFO is less than a watermark level for the last time, software polls IPRXFSTS[FILL], not INTR[IPRXWA]. After all data from the FIFO are copied

and all command sequences to flash memory are finished (INTR[IPCMDDONE] = 1), software sets IPRXFCR[CLRIPRXF] to clear the FIFO. If the FIFO is not cleared, the reading data will be incorrect for the next reading command to flash memory.

- Setting the INTR[IPRXWA] flag pops the IP receive FIFO data. Each read access to the FIFO does not pop the data.

10.2.2.9.1.2 Reading via DMA

To read data from the IP receive FIFO via DMA, configure the items below.

- Write 1 to IPRXFCR[RXDMAEN].
- Write the watermark level to IPRXFCR[RXWMRK]. The watermark level is (IPRXFCR[RXWMRK] + 1) × 8 bytes.
- Configure the DMA transfer minor loop size to match the watermark level.

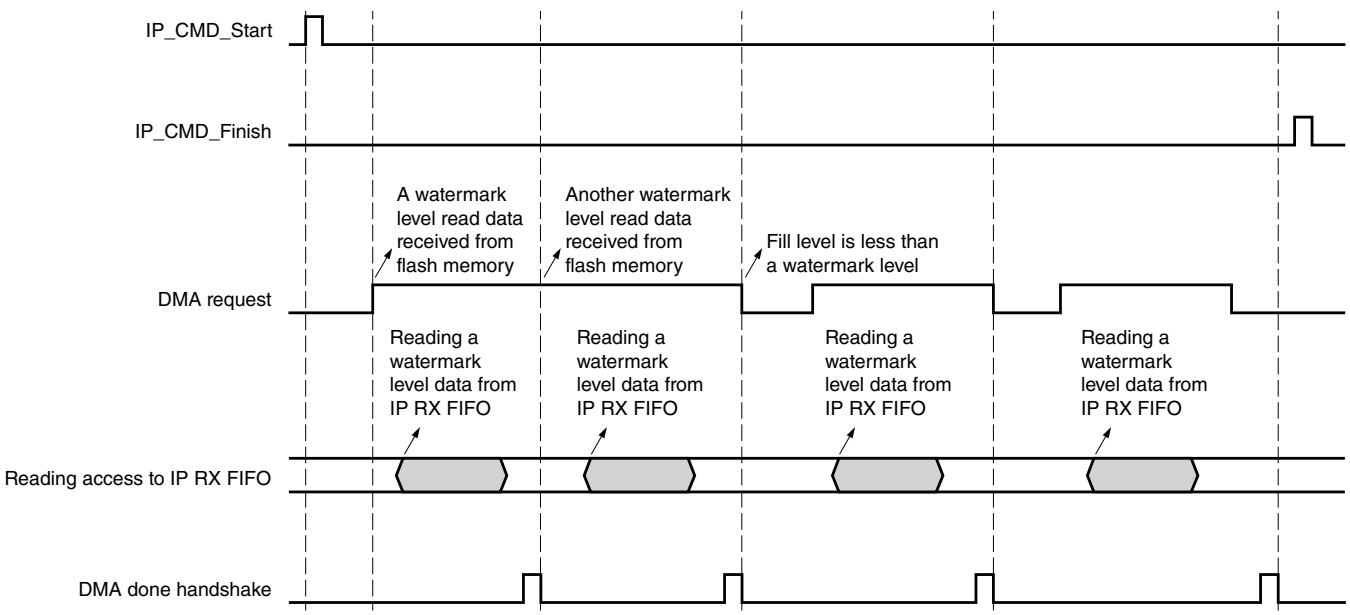


Figure 10-30. Reading from IP receive FIFO via DMA

NOTE

- When the fill level of the FIFO meets or exceeds the watermark level, a DMA request is generated. This request is level-valid, not pulse-valid.
- DMA reads a watermark level amount of data from the IP receive FIFO each time. (Configure the minor loop size to match the watermark level).

- DMA must return a done handshake (pulse valid) to FlexSPI each time it finishes reading a watermark level amount of data.
- The DMA done handshake (not each read access) pops the IP receive FIFO data.
- The total read and write data size (major loop size) must be a multiple of the watermark level. The DMA does not know when the data is ready for the last reading. When the data size is not a multiple of the watermark level, it is too complex for the DMA driver to poll IPRXFSTS[FILL].

10.2.2.9.2 Writing data to IP transmit FIFO

The programming data is put into the IP transmit FIFO, then FlexSPI transmits it to the flash memory. IPS or AHB access can write it:

- 400C_8180h–400C_81FCh (via IP bus)
- 33008000h–33008400h (via AHB bus)

To write the programming data into the IP transmit FIFO via AHB bus, write 1 to MCR0[ATDFEN]. The IP bus write access to the FIFO is ignored, but no bus error occurs.

To write the programming data into the IP transmit FIFO via IP bus, write 0 to MCR0[ATDFEN]. The AHB bus write access to the FIFO returns a bus error.

When FlexSPI fetches data to transmit, 64 bits of data from the IP transmit FIFO are popped. If the programming data size is not a multiple of 64 bits, the number of last popped valid bits is less than 64 bits. No error occurs; these invalid bits are not transmitted to the flash memory.

The processor or DMA can fill the IP transmit FIFO with programming data.

10.2.2.9.2.1 Writing via processor

To write data into the IP transmit FIFO via processor, configure the items below.

- Write 0 to IPTXFCR[TXDMAEN].
- Write the watermark level to IPTXFCR[TXWMRK]. The watermark level is (IPTXFCR[TXWMRK] + 1) × 8 bytes.
- Write 1 to INTEN[IPTXWEEN] to enable the IP transmit FIFO watermark available interrupt (optional).

The processor must poll INTR[IPTXWE] or wait for the IP transmit FIFO watermark empty interrupt before writing data to the FIFO. Waiting ensures that a watermark level amount of data is available in the IP transmit FIFO before writing.

After writing a watermark level amount of data to the FIFO, software must set the INTR[IPTXWE] flag to push that data into the FIFO. (The write pointer is incremented.)

NOTE

Setting the INTR[IPTXWE] flag pushes the IP transmit FIFO data. Each write access to the FIFO does not push the data.

Figure 10-31 shows the writing flow to the IP transmit FIFO via processor.

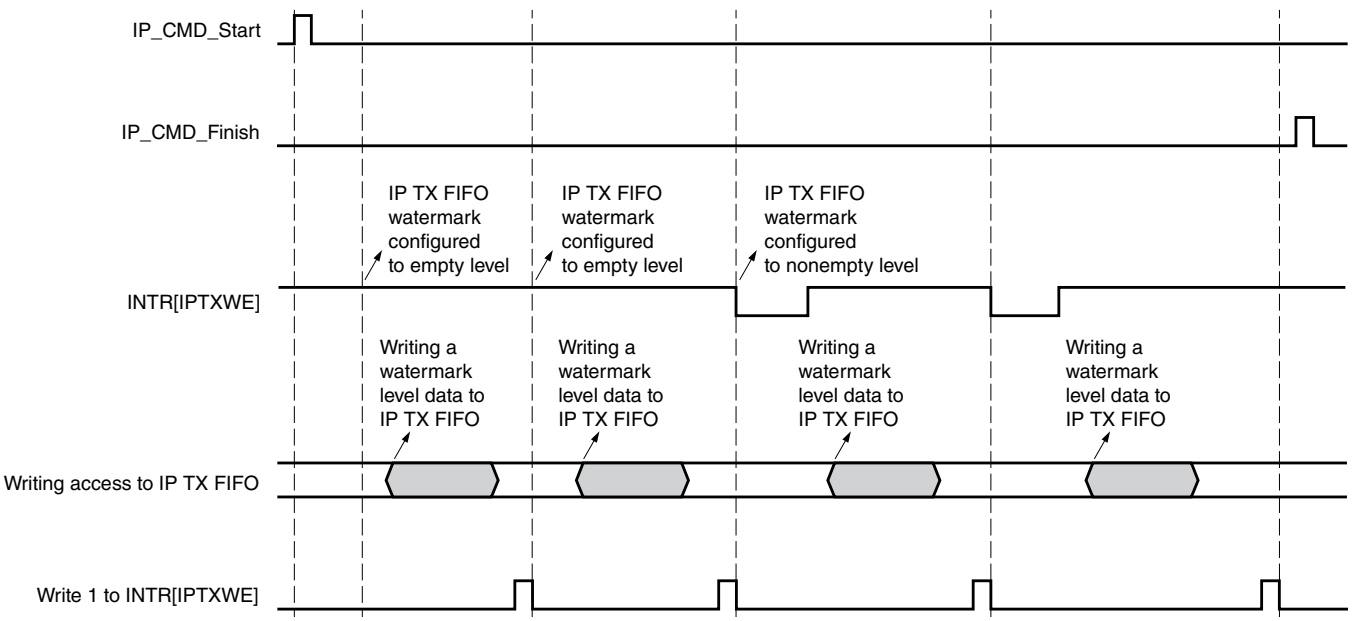


Figure 10-31. Writing to IP transmit FIFO via processor

NOTE

- The processor must write a watermark level amount of data into the IP transmit FIFO each time.
- The total flash write data size is not required to be a multiple of the watermark level. In this case, the writing data size to the FIFO is less than a watermark level for the last time. After all data to the FIFO are written and all command sequences to flash memory are finished (INTR[IPCMDDONE] = 1), the processor sets IPTXFCR[CLRIPTXF] to clear the FIFO. If the FIFO is not cleared, the programming data will be incorrect for the next programming command to flash memory.

10.2.2.9.2.2 Writing via DMA

To write data into the IP transmit FIFO via DMA, configure the items below.

- Write 1 to IPTXFCR[TXDMAEN].
- Write the watermark level to IPTXFCR[TXWMRK]. The watermark level is (IPTXFCR[TXWMRK] + 1) × 8 bytes.
- Configure the DMA transfer minor loop size to match the watermark level.

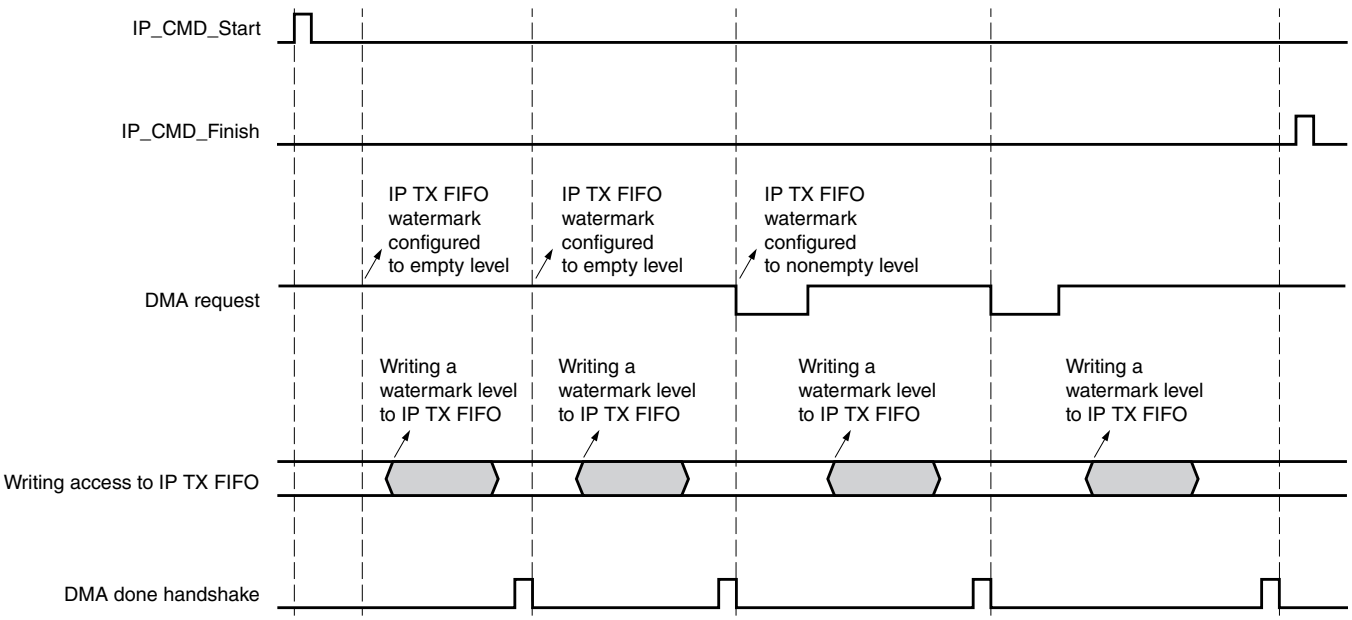


Figure 10-32. Writing into IP transfer FIFO via DMA

NOTE

- When there is empty space in the IP transmit FIFO greater than the watermark level, a DMA request is generated. This request is level-valid, not pulse-valid.
- DMA writes a watermark level amount of data into the FIFO each time. (Configure the minor loop size to match the watermark level.
- DMA returns a done handshake (pulse valid) to FlexSPI each time it finishes writing a watermark level amount of data.
- The DMA done handshake (not each write access) pushes the IP transmit FIFO data.
- The total program data size (major loop size) is not required to be a multiple of the watermark level. After writing all data into the IP transmit FIFO and all command

sequences to flash memory are finished (INTR[IPCMDDONE] = 1), write 1 to IPTXFCR[CLRIPTXF] to clear the FIFO. Failure to clear the FIFO results in incorrect programming data for the next programming command to flash memory.

10.2.2.10 Flash memory access via AHB command

Flash memory can be accessed via AHB bus directly on the AHB address space: 0h–10000000h. This address space is mapped to serial flash memory in FlexSPI. AHB bus accesses to this address space trigger flash memory access command sequences as needed.

For AHB read access to serial flash memory, FlexSPI fetches data from flash memory into the AHB receive buffer, then returns the data on the AHB bus. For AHB write access to serial flash memory, FlexSPI buffers AHB bus write data into the AHB transmit buffer, then transmits the data to serial flash memory.

No software configuration or polling is needed for AHB commands, except for FlexSPI initialization. As with a normal AHB target, the AHB controller accesses the external flash device transparently.

AHB commands are normally used to access serial flash memory space. IP commands must be used to access the control and status registers or other spaces, such as a one-time programmable (OTP) space in an external flash device.

The AHB commands for read and write access are discussed in detail below.

10.2.2.10.1 AHB write access to flash memory

For AHB write access to flash memory, FlexSPI buffers the write data from the AHB bus into the internal AHB transmit buffer. FlexSPI then transmits the data to flash memory. FlexSPI only buffers write data for one AHB burst. Figure 10-33 shows the hardware operation in response to an AHB write access to flash memory.

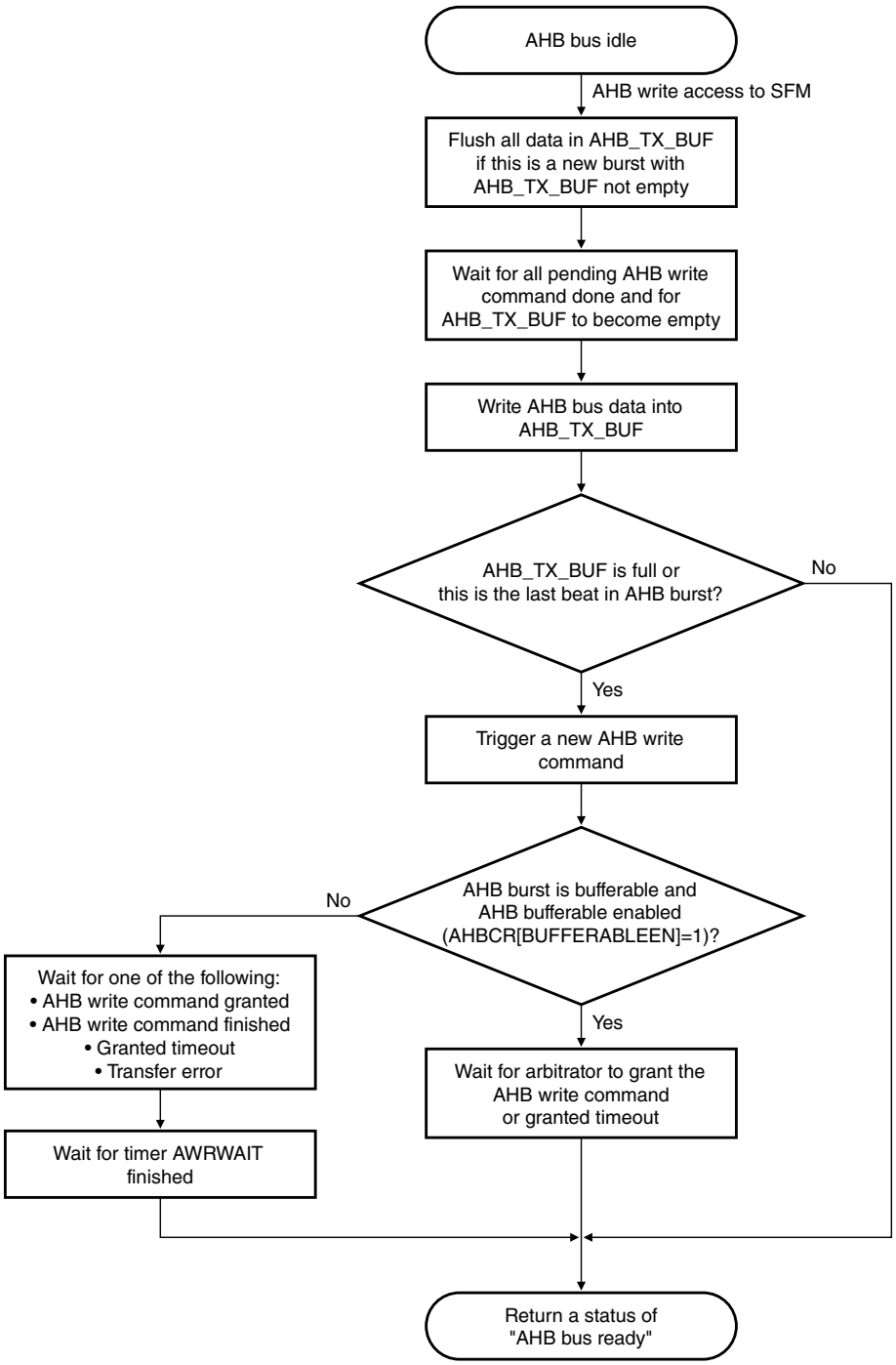


Figure 10-33. Hardware operation in response to AHB write access to flash memory

NOTE

1. If previous AHB burst is INCR write burst, AHB transmit buffer may not be empty when AHB bus is idle or new burst arrives. A new AHB write command is triggered to flush all data in AHB transmit buffer to external flash device. The new burst is held until this AHB write command finishes.

2. If the current access is bufferable, when the AHB write command is triggered, AHB ready is returned after the write command is granted. If the current access is non-bufferable, AHB ready is returned after the write command finishes.

FlexSPI triggers a new AHB write command in these cases:

- This beat is the last one in the current AHB burst, for any burst type except INCR.
- AHB transmit buffer is full after the current beat data is buffered.
- AHB bus becomes idle or a new burst arrives with the AHB transmit buffer not empty.

Table 10-9 describes the details of a triggered AHB write command.

Table 10-9. Triggered AHB write command

Item	Description
Flash memory access start address	Determined by AHB burst address. FlexSPI records the start address for the data in AHB transmit buffer, which is used as flash memory access start address.
Flash memory chip select	FlexSPI uses the settings of the flash memory access start address and flash memory size to determine the chip selection.
Flash memory command sequence index	Determined by FLSHxCR2[AWRSEQID].
Flash memory command sequence number	Determined by FLSHxCR2[AWRSEQNUM]. If AWRSEQNUM is not zero, multiple flash memory access command sequences are triggered every time for an AHB write command. The sequences indexed from AWRSEQID to (AWRSEQID + AWRSEQNUM) in LUT are executed sequentially.
Flash memory access mode (Individual/ Parallel)	Determined by AHBCR[APAREN].
Flash memory data size	Determined by byte number of buffered data in AHB transmit buffer.

The following examples indicate internal logic for AHB write access to flash memory, where AHB_TX_BUF is 64 bytes :

- AHB INCR 64-bit bufferable burst with address sequence 8h, 10h, 18h, 20h, ..., 50h (10 beats total).

  Two AHB write commands are triggered: the first command with flash memory start address 8h and data size 40h; the second command with flash memory start address 48h and data size 10h. See Figure 10-34.

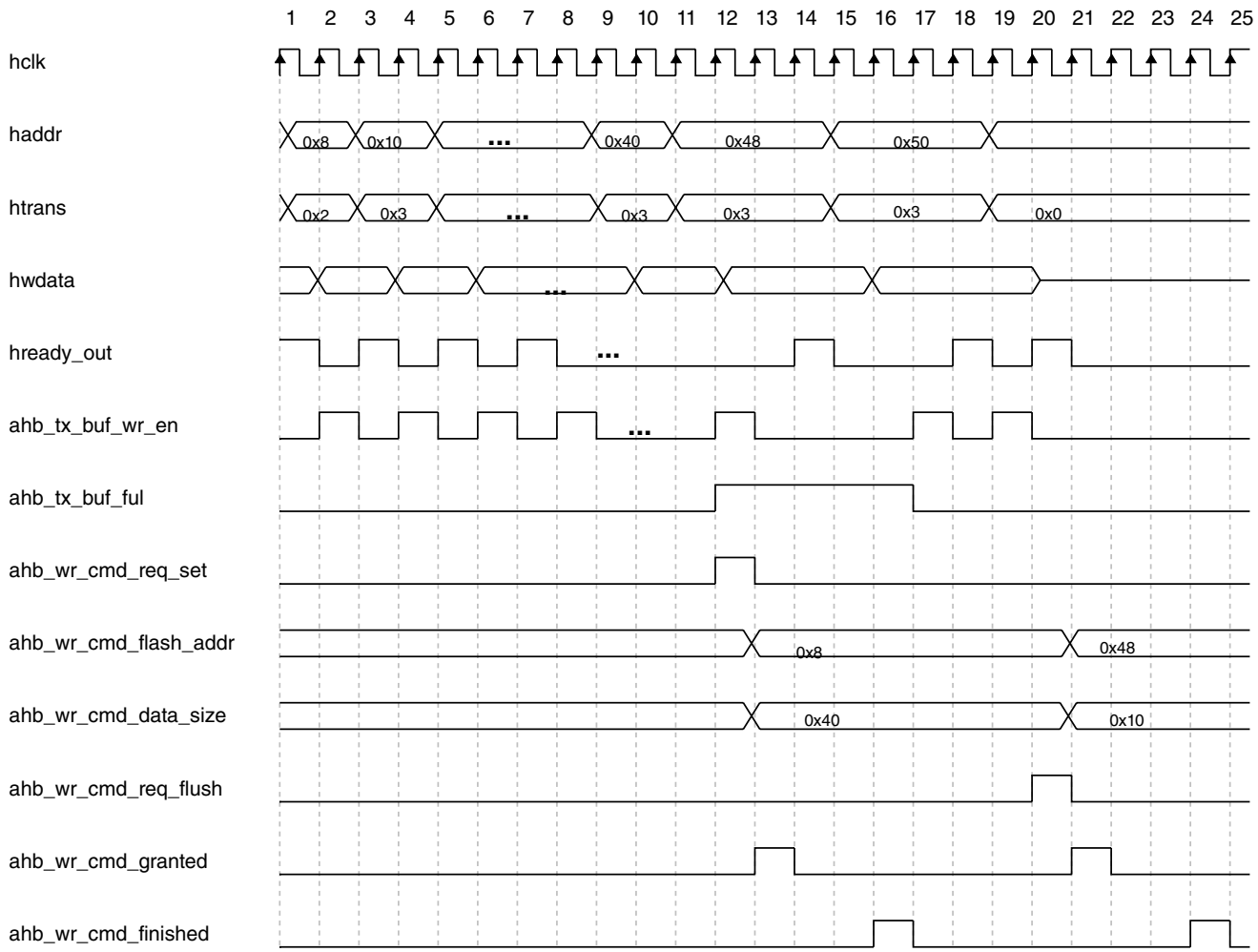


Figure 10-34. AHB write access (INCR, 64-bit, bufferable)

- AHB WRAP8 64-bit bufferable burst with address sequence 28h, 30h, 38h, 0h, 8h, 10h, 18h, 20h.

  One AHB write command is triggered with flash memory start address 0h and data size 40h. See Figure 10-35.

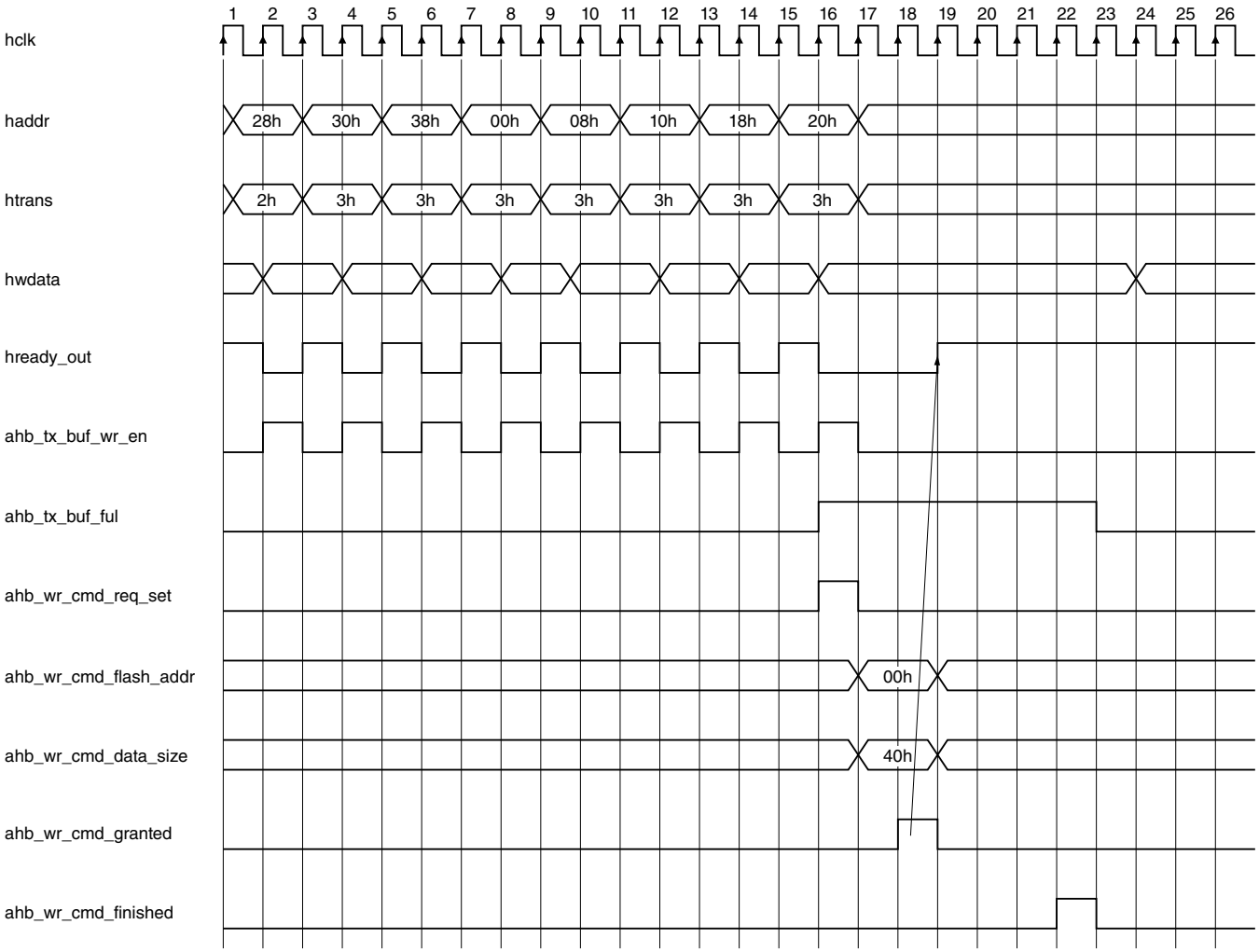


Figure 10-35. AHB write access (WRAP8, 64-bit, bufferable)

- AHB WRAP8 64-bit non-bufferable burst with address sequence 28h, 30h, 38h, 0h, 8h, 10h, 18h, 20h.

  One AHB write command is triggered with flash memory start address 0h and data size 40h.

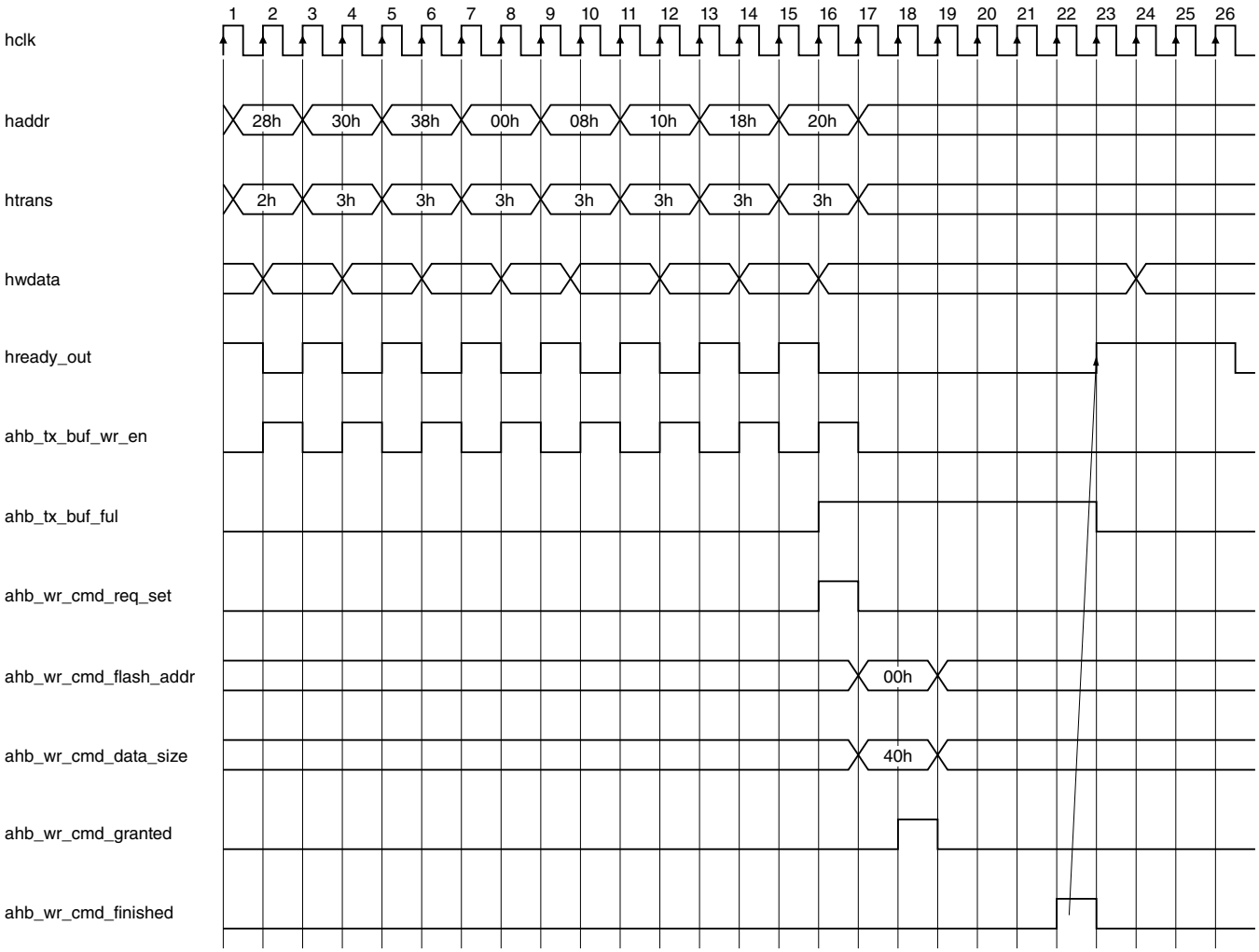


Figure 10-36. AHB write access (WRAP8, 64-bit, non-bufferable)

NOTE

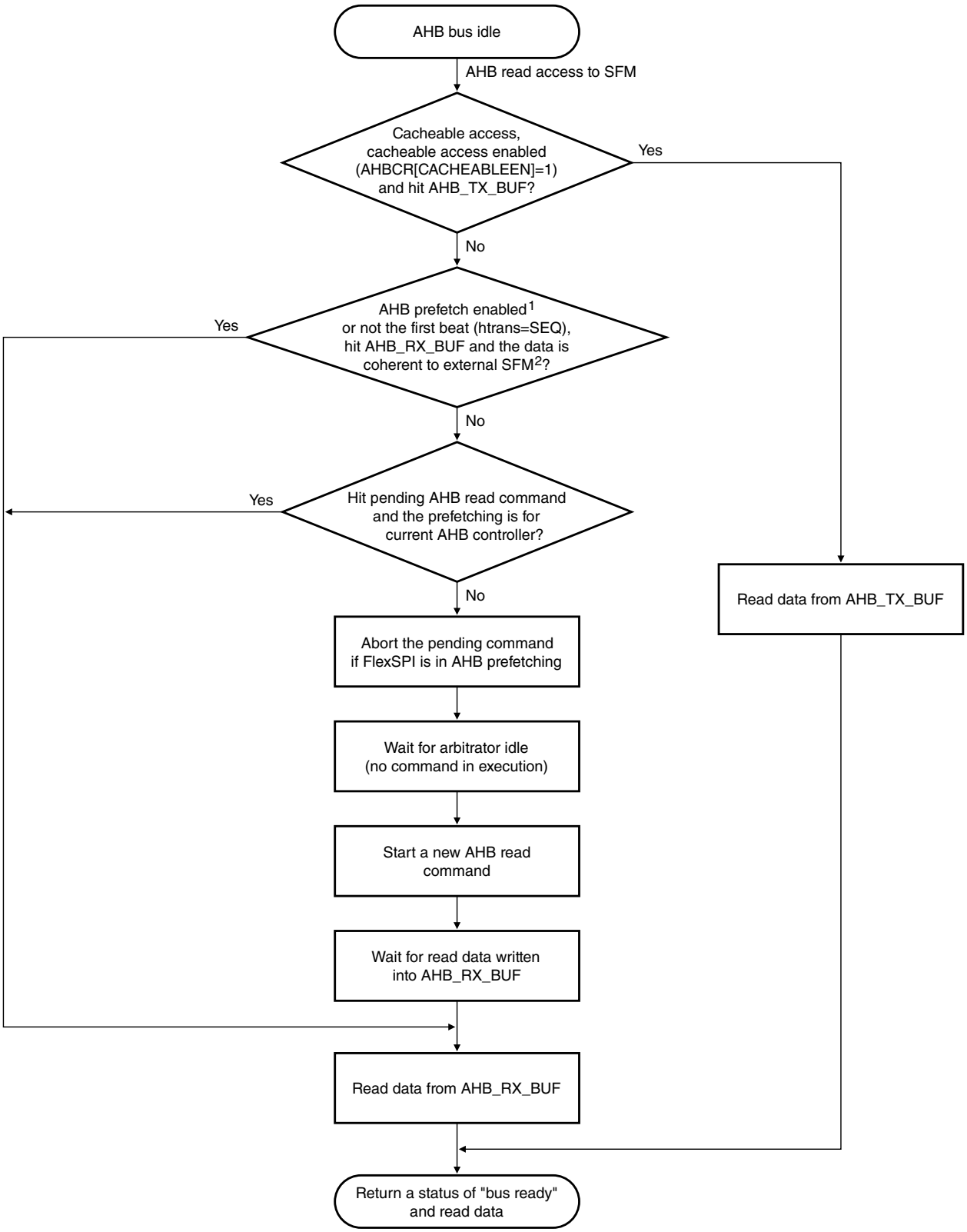
If the burst data size (in bytes) is greater than the AHB transmit buffer size, wrapper burst is not supported. For example, if AHB_TX_BUF is 64 bytes, AHB WRAP16 (16 × 64 bits = 128 bytes) write burst access is not supported.

10.2.2.10.2 AHB read access to flash memory

For AHB read access to flash memory, FlexSPI checks the burst access type and register setting to determine whether to access these locations:

- The AHB transmit buffer
- The AHB receive buffer
- A pending AHB read command

If all of these options are missed, FlexSPI triggers a new AHB read command to fetch data from flash memory. Figure 10-37 shows the hardware operation in response to AHB read access to flash memory.



Notes:
1) AHB prefetch is enabled when both AHBCR[PREFETCHEN]=1 and AHBRXBUFxCR0[PREFETCHEN]=1 (x is the AHB RX buffer ID for the current AHB controller).
2) The AHB RX buffer holds the data read from flash memory. This data may become incoherent if AHB writes to the same flash-memory address.

Figure 10-37. Hardware operation in response to AHB read access to flash memory

Table 10-10 describes the details of a triggered AHB read command.

Table 10-10. Triggered AHB read command

Item	Description
Flash memory access start address and data size	Determined by AHB address, burst type, and burst size. FlexSPI fetches read data from the start address for the current burst or beat. See Table 10-11 for details.
Flash memory chip select	FlexSPI uses the settings of the flash memory access start address and flash memory size to determine the chip select.
Flash memory command sequence index	Determined by FLSHxCR2[AWRSEQID].
Flash memory command sequence number	Determined by FLSHxCR2[AWRSEQNUM]. If AWRSEQNUM is not zero, multiple flash memory access command sequences are triggered every time for an AHB read command. The sequences indexed from AWRSEQID to (AWRSEQID + AWRSEQNUM) in LUT are executed sequentially.
Flash memory access mode (Individual/ Parallel)	Determined by AHBCR[APAREN].

NOTE

- FlexSPI automatically determines which ARDSEQNUM and ARDSEQID fields that the flash device uses as sequence ID for chip selection. See MCR2[SAMEDEVICEEN].
- FlexSPI determines which AHB receive buffer to use for the current AHB read access by controller ID. For details about the AHB receive buffer ID and AHB controller ID mapping, see AHB receive buffer management.
- You cannot allocate the size of the AHB receive buffer to be less than the AHB Burst size. This allocation results in unknown behavior.

Table 10-11. AHB read command flash memory start address and data size

Prefetch enable	Cross flash boundary	Burst type	Flash memory start address [27:0]	Data size
0	Never cross flash boundary, because AHB burst never crosses 1 KB boundary.	SINGLE, INCR4, INCR8, INCR16	hbeat_start_address	(hburst_end_address-hbeat_start_address)
		INCR	hbeat_start_address	byte size of current beat For INCR burst with prefetch disabled, each beat is handled same as SINGLE burst.
		WRAP4, WRAP8, WRAP16	hburst_start_address	(hburst_end_address-hburst_start_address)

Table continues on the next page...

Table 10-11. AHB read command flash memory start address and data size (continued)

Prefetch enable	Cross flash boundary	Burst type	Flash memory start address [27:0]	Data size
1	No	INCR, SINGLE, INCR4, INCR8, INCR16	hbeat_start_address	ahb_rx_buf_sz
	No	WRAP4, WRAP8, WRAP16	hburst_start_address	ahb_rx_buf_sz
	Yes	INCR, SINGLE, INCR4, INCR8, INCR16	hbeat_start_address	(flash_top_address-hbeat_start_address)
	Yes	WRAP4, WRAP8, WRAP16	hburst_start_address	(flash_top_address-hburst_start_address)

NOTE

- hbeat_start_address is HADDR input from the AHB controller for current beat.
- hburst_start_address (for example 00h) is the lowest address for current burst. hburst_end_address (for example 10h) is the highest address in current burst plus 1. For example, WRAP4 burst with HADDR 8h, Ch, 0h, 4h.
- ahb_rx_buf_sz is the buffer size in bytes of AHB receive buffer that the current AHB controller uses.
- flash_top_address is the top address of currently accessed flash memory.

10.2.2.10.3 AHB receive buffer management

There are 8 buffers (Buffer 0–7) in the AHB receive buffer. These buffers are transparent to AHB controllers. FlexSPI fetches flash memory data and returns it on the AHB bus automatically. No status register polling is needed for AHB read access to Serial Flash Memory (SFM).

The total size of AHB receive buffers is 2 KB. Use AHBRXBUF0CR0[BUFSZ]–AHBRXBUFCR7[BUFSZ] to configure the buffer size of each AHB receive buffer. The buffer size for Buffer 0 to Buffer 7 could be set to 0. If the buffer size is set to 0, FlexSPI ignores the related setting of AHBRXBUFxCR0[MSTRID]. Buffer 7 is used for all AHB controllers not assigned to Buffer 0–6. FlexSPI ignores the Buffer 7 size setting field (AHBRXBUF7CR0[BUFSZ]). Its buffer size is: (AHB receive buffer total size) - (the sum of the sizes of Buffer 0–6).

When an AHB read access to serial flash memory occurs, FlexSPI determines which buffer to use via this method:

1. If the controller ID equals AHBRXBUF0CR0[MSTRID] and AHBRXBUF0CR0[BUFSZ] is not zero, Buffer 0 is used.
2. If the controller ID equals AHBRXBUF1CR0[MSTRID] and AHBRXBUF1CR0[BUFSZ] is not zero, Buffer 1 is used.
3. If the controller ID equals AHBRXBUF2CR0[MSTRID] and AHBRXBUF2CR0[BUFSZ] is not zero, Buffer 2 is used.
4. If the controller ID equals AHBRXBUF3CR0[MSTRID] and AHBRXBUF3CR0[BUFSZ] is not zero, Buffer 3 is used.
5. If the controller ID equals AHBRXBUF4CR0[MSTRID] and AHBRXBUF4CR0[BUFSZ] is not zero, Buffer 4 is used.
6. If the controller ID equals AHBRXBUF5CR0[MSTRID] and AHBRXBUF5CR0[BUFSZ] is not zero, Buffer 5 is used.
7. If the controller ID equals AHBRXBUF6CR0[MSTRID] and AHBRXBUF6CR0[BUFSZ] is not zero, Buffer 6 is used.
8. If all the above cases are not met, Buffer 7 is used.

NOTE

- Software must ensure that the size of each buffer is not less than the maximum burst size of an AHB read access from the AHB controller using this buffer. Otherwise, the behavior is undefined. The minimum buffer size request depends on the chip implementation. For example, if the capability of the chip to send out a maximum burst is 64 bytes, the minimum AHB receive buffer is 64 bytes.
- Assigning multiple buffers to a single AHB controller is not supported, unless the prefetch buffer enhancement feature is being used.
- When AHB read prefetch is enabled (AHBCR[PREFETCHEN] = 1), the size of the buffer determines the prefetch data size. If it does not cross the flash boundary, FlexSPI fetches data from the external flash device with that buffer size.
- The priority setting of AHB controller (AHBRXBUFxCR0[PRIORITY] = 1) is used only for suspending control of AHB prefetching. See Command abort and suspend.

10.2.2.11 Command arbitration

There are four flash memory access command sources:

1. AHB command, that AHB write access to serial flash memory space triggers
2. AHB command, that AHB read access to serial flash memory space triggers
3. IP command, that writing IPCMD[TRG] triggers
4. Suspended command, an AHB read prefetch sequence which is suspended

An AHB bus access never triggers a write command and a read command at the same time.

The AHB prefetch sequence is an AHB command sequence triggered via AHB read access when AHB prefetch is enabled. After all read data is fetched for the current AHB read burst, FlexSPI prefetches more data to reduce read latency for the next AHB read access. An AHB command for a read operation is never aborted while fetching read data for the current AHB read burst. However, a new IP command or AHB command request while prefetching data (not for the current read burst) can abort an AHB read command.

When the arbitrator is idle (STS0[ARBIDLE] = 1), the granted priority of these command sources is:

1. AHB command (read or write)
2. IP command
3. Suspended command

NOTE

When the arbitrator is idle and there is no AHB or IP command request, a suspended command is not granted immediately. The arbitrator waits MCR2[RESUMEWAIT] AHB clock cycle idle states before resuming the suspended command to avoid AHB prefetch sequence being resumed and suspended frequently.

If the arbitrator is busy executing AHB or IP commands (not suspended commands), no new command requests are granted. If the grant times out, an AHB or IP command granted error occurs.

If a new AHB or IP command request comes while arbitrator is executing AHB read prefetch sequence, AHB read prefetch sequence is aborted (but not immediately). Arbitrator grants the AHB or IP command request after AHB read prefetch sequence is aborted on the FlexSPI interface and saved all internal data pointers.

10.2.2.11.1 Command abort and suspend

Table 10-12 describes the command abort and suspend mechanism.

Table 10-12. Command abort and suspend

Action	Description
Aborting a command	As mentioned above, if a new AHB or IP command request arrives, an AHB read prefetch sequence can be aborted.
Suspending a command	When an AHB read prefetch sequence is aborted on the FlexSPI interface, FlexSPI saves this suspended sequence in the cases listed below. FlexSPI resumes this sequence after the arbitrator has been idle for enough time. • There is no valid suspended command (AHBSPNDSTS[ACTIVE] = 0). This condition can occur when no command is suspended or when the suspended command is resumed. • The priority of aborted AHB read prefetch sequence is higher than the active suspend sequence. **NOTE:** FlexSPI ignores and never resumes the original suspended sequence.
Activating a suspended command	The suspended command (internal status) becomes active when an AHB prefetch command is aborted and suspended. It becomes inactive in the following cases: • The arbitrator resumes the suspended command. • The AHB controller triggers a new AHB read command request using the same AHB receive buffer (buffer ID). **NOTE:** MCR0[SWRESET] can be used to clear any suspended command. The saved suspended command is dropped and no resuming occurs. The suspended command must be cleared after the flash page program or erase. Otherwise, another command may lead to a period in which the flash memory cannot support a read command.

Figure 10-38 shows an example of the flow of command abort, suspend, or resume.

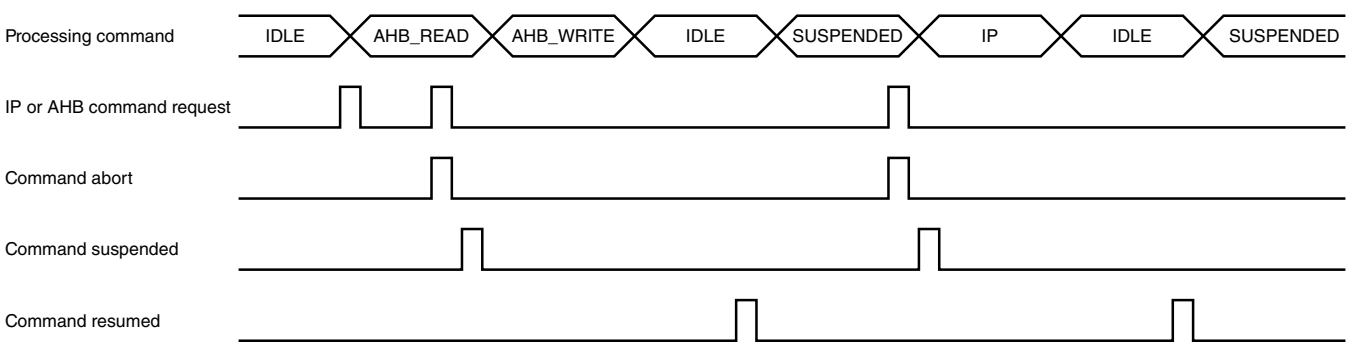


Figure 10-38. Command instruction execution on FlexSPI interface

NOTE

- The AHB prefetch sequence is aborted, suspended, or resumed twice in Figure 10-38. A new AHB write command request aborts the sequence the first time, and a new IP command request aborts the sequence the second time.
- The suspended sequence is resumed after a wait time of idle state. If there are frequent IP or AHB command

requests and the wait time is not set reasonably, the suspend and resume activities may be triggered frequently.

10.2.2.12 SCLK stop

FlexSPI stops SCLK output toggling:

- When programming data is not ready for a programming command sequence.
- When an internal FIFO has no space to receive data for reading command sequence.

Certain devices may not support the stopping of SCLK during a command sequence (chip select is valid). SCLK stopping can be avoided via the methods described below.

- For flash memory reading that an IP command triggers:
  - Never trigger a read command with data size larger than IP receive FIFO size.
  - Internal asynchronous FIFO for flash memory reading should never be full.

    FlexSPI pops 64 bits of data per AHB clock cycle from this FIFO, and receives data from FlexSPI interface on the serial root clock. The flash memory access mode (Single, Dual, Quad, or Octal mode and Individual or Parallel mode) determines the receiving speed. For example, in Octal mode and Parallel mode, FlexSPI receives 16 bits per serial root clock cycle. If the AHB clock frequency is faster than 1/4 of the serial root clock, this asynchronous FIFO is never full.
- For flash memory programming that an IP command triggers:
  - Never trigger a program command with data size larger than the IP transmit FIFO size. Write all programming data into the IP transmit FIFO before triggering the IP command.
  - The internal asynchronous FIFO for flash memory programming must never be empty.

    FlexSPI fetches 64 bits of programming data per AHB clock cycle into this FIFO, and transmits data to the FlexSPI interface on the serial root clock. The flash memory access mode (Single, Dual, Quad, or Octal mode and Individual or Parallel mode) determines the transmitting speed. For example, in Octal mode and Parallel mode, FlexSPI transmits 16 bits per serial root clock cycle. If the AHB clock frequency is faster than 1/4 of the serial root clock, this asynchronous FIFO is never empty.
- For flash memory reading or programming that an AHB command triggers:
  - The internal asynchronous FIFO for flash memory reading and programming must never be full or empty. The frequency ratio limitation is same as it is for flash memory reading or flash memory programming that an IP command triggers.

NOTE

- FlexSPI never triggers an AHB read command with data size larger than the internal AHB receive buffer size.
- FlexSPI never triggers an AHB program command with data size larger than the internal AHB transmit buffer size.
- All programming data is buffered into the AHB transmit buffer before triggering an AHB program command in FlexSPI.

10.2.2.13 FlexSPI output timing

10.2.2.13.1 Output timing between data and SCLK

This section describes the output timing relationship of data (on A_DATA and B_DATA) and SCLK. There are three cases.

- **SDR instruction in SDR sequence**
  An SDR sequence contains only SDR instructions. In this case, all data bits last one serial root clock cycle on the FlexSPI interface. Figure 10-39 shows the relationship between the serial root clock, data, and SCLK:

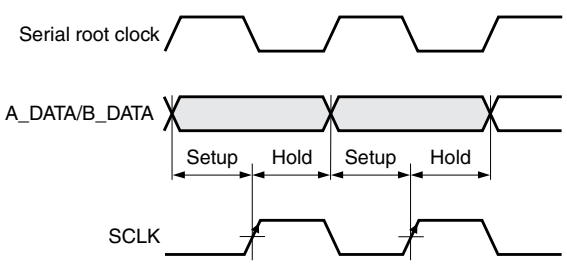


Figure 10-39. SDR instruction in SDR sequence

- **SDR instructions in a DDR sequence**

  A DDR sequence is a flash memory access command sequence that contains DDR instructions other than DUMMY_DDR. It may contain SDR instructions. If there are SDR instructions in a DDR sequence, all data bits last for two serial root clock cycles on the FlexSPI interface. Figure 10-40 shows the relationships between the serial root clock, data, and SCLK.

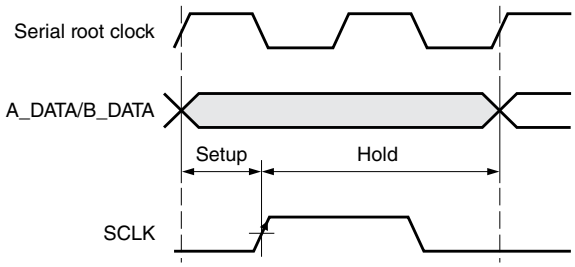


Figure 10-40. SDR instruction in DDR sequence

- **DDR instructions in a DDR sequence**

  For DDR instructions in a DDR sequence, all data bits last one serial root clock cycle on the FlexSPI interface. Figure 10-41 shows the relationships between the serial root clock, data, and SCLK.

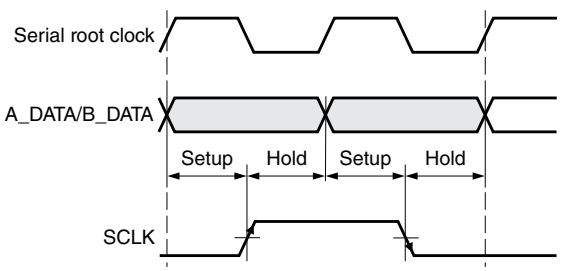


Figure 10-41. DDR instruction in DDR sequence

10.2.2.13.2 Output timing between chip select and SCLK

This section describes the output timing relationship between chip select (on A_SS0_B , A_SS1_B, B_SS0_B, and B_SS1_B) and SCLK. The timing relationship for the SDR sequence differs a little from the timing relationship for the DDR sequence.

- **Chip select timing in SDR sequence**

  For an SDR sequence, the delay from chip select assertion to the SCLK rising edge is (FLSHxCR1[TCSS] + 0.5) cycles of the serial root clock. The delay from SCLK falling edge to chip select deassertion is FLSHxCR1[TCSH] cycles of serial root clock. Figure 10-42 shows the timing relationship between the chip select and SCLK.

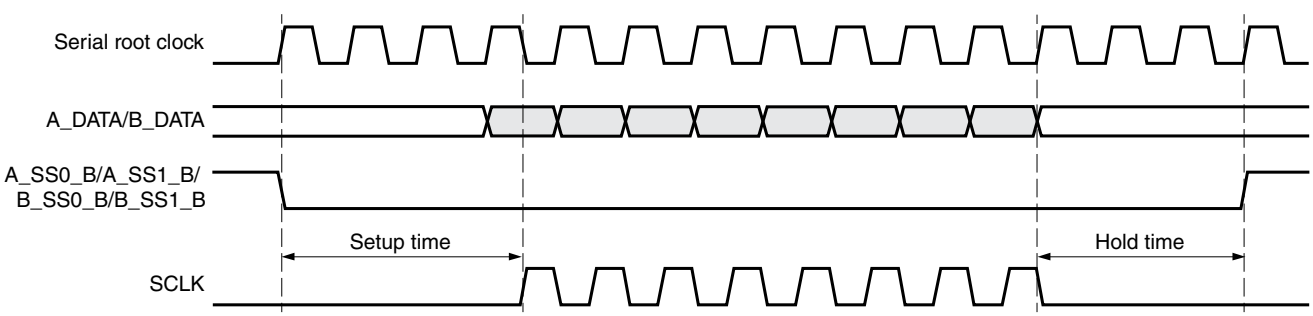


Figure 10-42. Chip select output timing for SDR sequence

- **Chip select timing in DDR sequence**

For a DDR sequence, the delay from chip select assertion to the SCLK rising edge is **(FLSHxCR1[TCSS] + 0.5)** cycles of the serial root clock. The delay from the SCLK falling edge to chip select deassertion is **(FLSHxCR1[TCSH] + 0.5)** cycles of serial root clock.

NOTE

When AHB receive prefetch is enabled, and prefetch can be aborted, configure TCSH to 1 or greater. Configuring TCSH in this way guarantees a positive chip select hold time after the SCLK falling edge.

Figure 10-43 shows the timing relationship between chip select and SCLK.

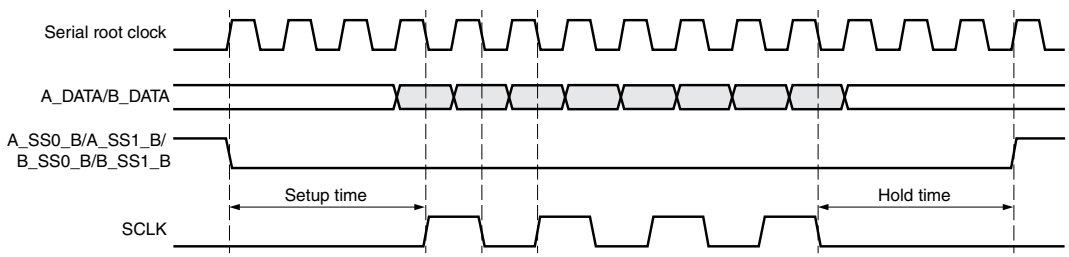


Figure 10-43. Chip select output timing for DDR sequence

For devices such as an FPGA device, a minimum chip select negation time (time between two chip select assertions) may be required. If the value of FLSHxCR1[CSINTERVAL] is nonzero, FlexSPI ensures a delay between chip select valid assertions. The delay time is CSINTERVAL × 256 cycles of serial root clock for an SDR or DDR sequence. If the external device has no limitation, configure the value of this field to zero. Figure 10-44 shows the timing of the chip select interval.

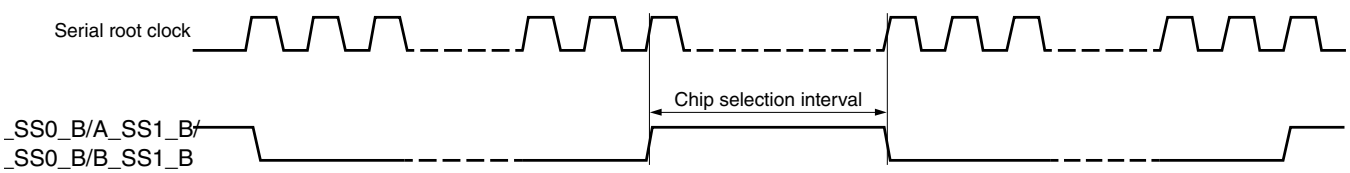


Figure 10-44. Chip select valid interval

10.2.2.14 FlexSPI input timing

This section describes the input timing of FlexSPI.

10.2.2.14.1 Receiving block

This section describes the receiving block in FlexSPI.

FlexSPI decodes instruction code to determine SDR mode or DDR mode. It also determines whether both the rising and falling edges are used for flash data sampling. FlexSPI samples the data line for learn instructions and read instructions.

MCR0[RXCLKSRC] selects the clock source from these four sources for the sampling clock:

- Internal dummy read strobe and looped back internally
- Internal dummy read strobe and looped back from DQS pad
- SCK output and looped back from SCK pad
- Flash-memory-provided read strobe

For details about DLL configuration, see DLL configuration for sampling.

Figure 10-45 shows the sampling block in FlexSPI.

FlexSPI uses both rising and falling edges to sample data from flash_A and flash_B. Two muxes with four clock source inputs generate the signals ipp_ind_io_fa, ipp_ind_io_fb. See Receive clock source features. ipp_ind_io_fa_int and ipp_ind_io_fb_int are reversed from ipp_ind_io_fa and ipp_ind_io_fb. Both sampling flip-flops are used when FlexSPI is in DDR mode. Only the falling edge sampling flip-flop is used in SDR mode.

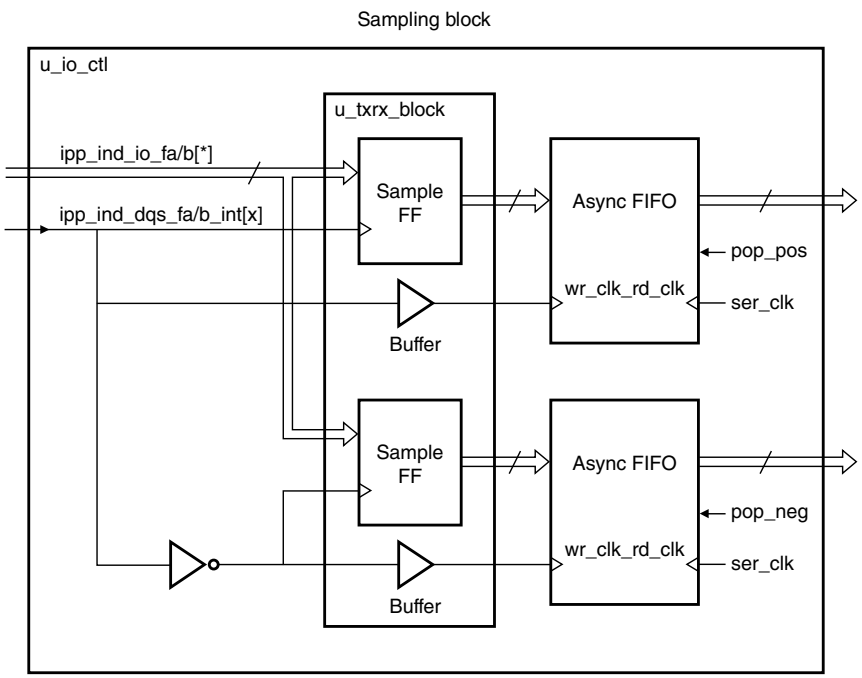


Figure 10-45. Sampling block in FlexSPI

Figure 10-46 is the receiving block diagram for data learning.

There are 16 clock phases for sampling data. An internal delay chain delays the clock phases.

See Data learning.

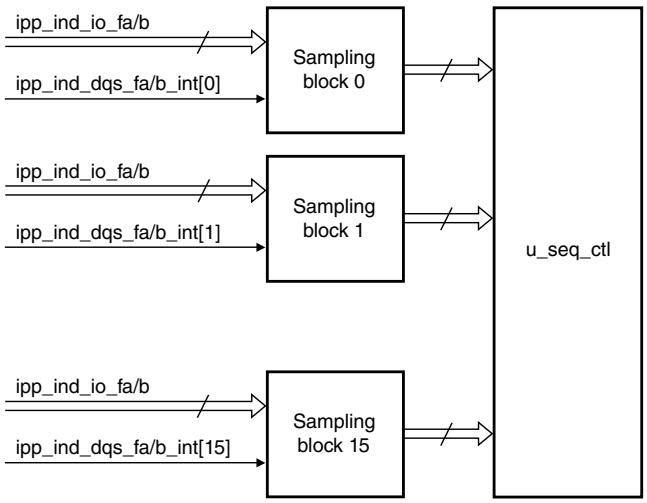


Figure 10-46. Sampling blocks for data learning

FlexSPI does not support read operations with a zero-turnaround cycle for direction switching on I/O pins. This read operation often occurs in QuadSPI configuration register reads. In Figure 10-47 and Figure 10-48, SIO[3:0] is an output from FlexSPI in the CMD phase, and is an input to FlexSPI in the DATA phase. No time remains for FlexSPI to switch the direction. This kind of read is not supported. FlexSPI and flash memory drive the I/O pins at the same time, which may cause a problem. To avoid this behavior on I/O pins, using SPI mode or reading with dummy cycles is required.

NOTE

The cycle numbers are not accurate in the examples below. See the device spec for accurate cycle numbers.

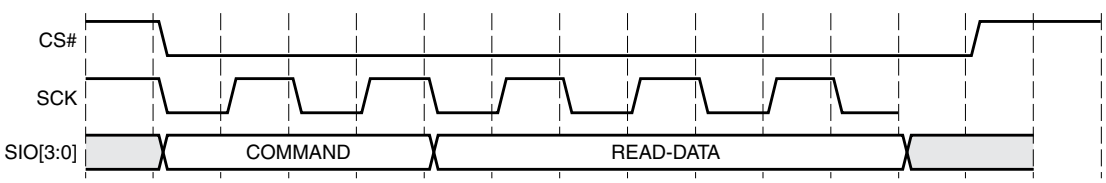


Figure 10-47. Unsupported read behavior

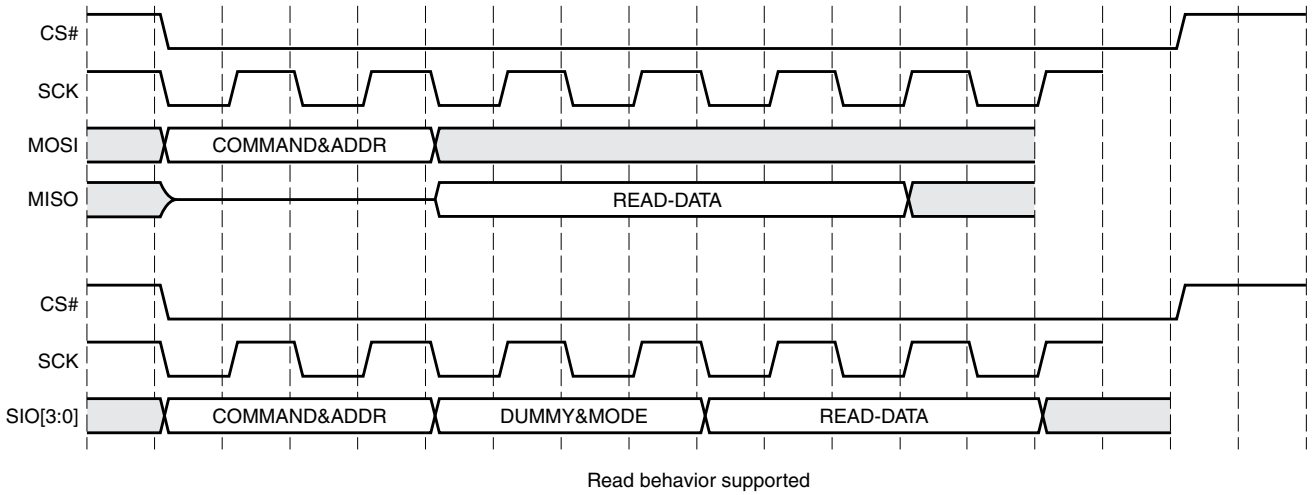


Figure 10-48. Supported read behavior

10.2.2.14.2 Receive clock source features

FlexSPI uses a read strobe to sample incoming data. There are several options for generating the read clock strobe that MCR0[RXCLKSRC] selects. Using the receive clock source method has implications for the board connections and can affect the maximum frequency that the interface supports. See the device data sheet for details on the maximum frequency supported for each RXCLKSRC option.

Table 10-13. RXCLKSRC options

MCR0[RXCLKSRC]	Description	Board connection	Max Frequency
0	Internal dummy read strobe and internal loopback	DQS pin is not used. The DQS pin can be configured for an alternate signal function.	Lowest
1	Internal dummy read strobe and loopback from DQS pad	FlexSPI uses the DQS pin, and it must be configured for its FlexSPI function. The internally generated read strobe is sent to the DQS pin and is sampled at the pin to match more closely the data pin timings. The DQS pin is typically left floating. External capacitance can be added to adjust timing.	Medium
3	Flash-memory-provided read strobe	DQS pin is connected to the DQS or RWDS signal that the memory provides.	Highest

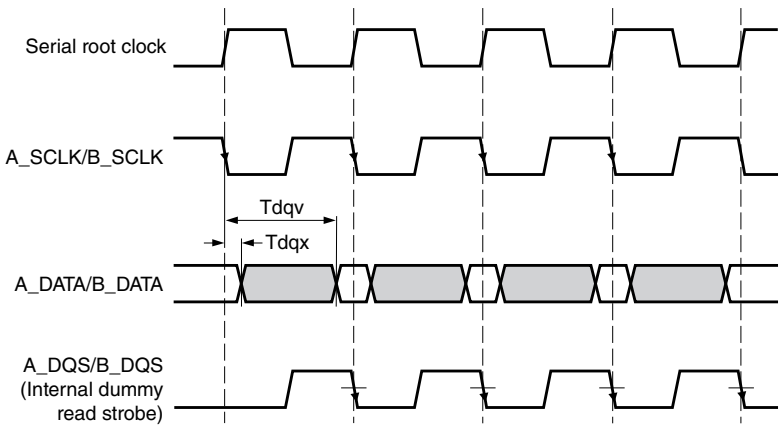
10.2.2.14.3 Input timing for sampling with dummy read strobe

This section describes the input timing when sampling with the internal dummy read strobe (MCR0[RXCLKSRC] = 0 or 1). The timing for sampling with an internal dummy read strobe loopback is very similar to the timing for loopback from pad. Sampling with a dummy read strobe loopback from the DQS pad can achieve a higher read frequency. It compensates for the delay of the SCLK output path and data pin input path.

The input timing is different for SDR mode and DDR mode.

- **Input timing for sampling with dummy read strobe in SDR mode**

  For SDR read or learn instructions, FlexSPI samples input data pins with the falling edge of the dummy read strobe. Figure 10-49 shows the input timing for sampling with dummy read strobe in SDR mode.



Tdqv is the setup time from SCLK to the next valid data.
Tdqx is the hold time from SCLK to the current valid data.

Figure 10-49. Input timing for sampling with dummy read strobe in SDR mode

- **Input timing for sampling with dummy read strobe in DDR mode**

  For DDR read or learn instructions, FlexSPI samples input data pins on rising and falling edges of the dummy read strobe. Figure 10-50 shows the input timing for sampling with dummy read strobe in DDR mode.

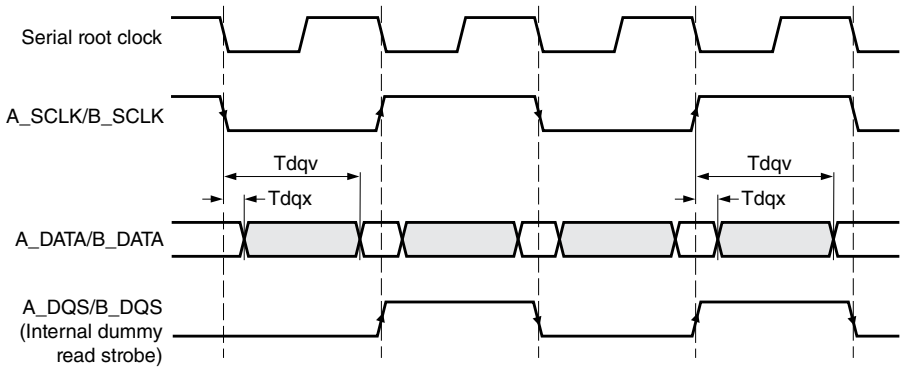


Figure 10-50. Input timing for sampling with dummy read strobe in DDR mode

10.2.2.14.4 Input timing for sampling with flash-memory-provided read strobe

This section describes the input timing when sampling with flash-memory-provided read strobe (MCR0[RXCLKSRC] = 3). The input timing is different for SDR mode and DDR mode.

NOTE

There are no known devices that provide a read strobe and support SDR mode operation.

- **Flash-memory-provided read strobe with SCLK**

  Certain flash devices provide both read data and read strobes with SCLK. Then the read strobe edge is aligned with the read data change. The FlexSPI controller delays the read strobe for one half cycle of the serial root clock (with DLL), then samples read data with the delayed strobe. See DLL configuration for sampling. Figure 10-51 and Figure 10-52 show the input timing for sampling with flash memory read strobe in SDR mode and DDR mode.

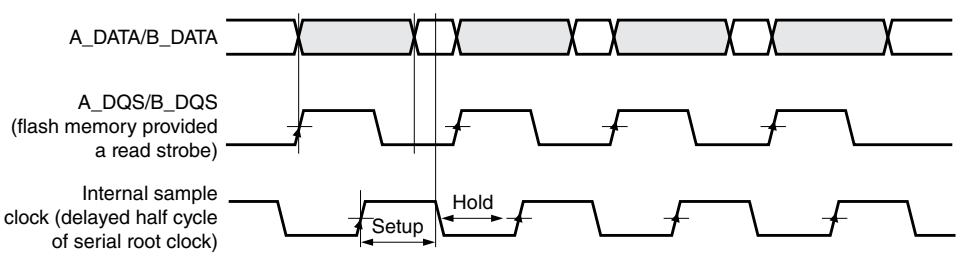


Figure 10-51. Input timing 2 for flash-memory-provided read strobe in SDR mode

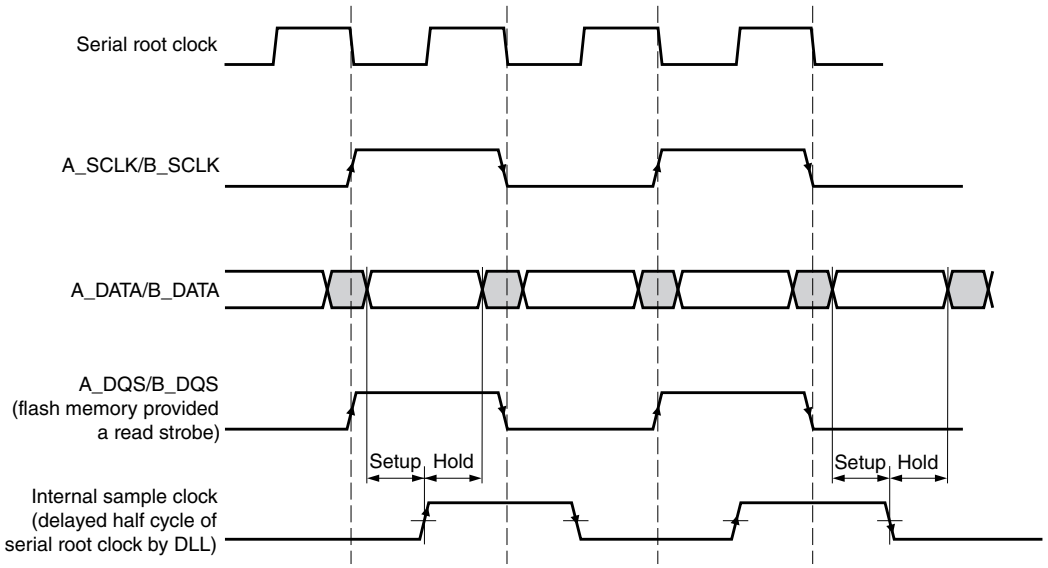


Figure 10-52. Input timing 2 for flash-memory-provided read strobe in DDR mode

10.2.2.14.5 DLL configuration for sampling

For the four sampling clock sources described previously, the input timing differs depending on the DLLxCR register configuration, according to the sampling clock source mode. The DLL is a delay-line chain that can be set to a fixed number of delay cells. It can also be auto-adjusted to lock on a certain phase delay to the reference clock.

- DLLxCR must be set to 0000_0100h (one fixed delay cell in DLL delay chain) in the following cases:
  - Sampling data with dummy read strobe looped back internally (MCR0[RXCLKSRC] = 0h)
  - Sampling data with dummy read strobe looped back from DQS pad (MCR0[RXCLKSRC] = 1h)
- When data is sampled with a flash-memory-provided read strobe (MCR0[RXCLKSRC] = 3h) and flash memory provides the read strobe with SCLK, DLL must be configured as follows (when serial root clock is over 100 Mhz. These settings ensure a lock of the reference clock (serial root clock).
  - DLLxCR[SLVDLYTARGET] is up to flash parameters used.
  - DLLxCR[REFPHASEGAP] = 2h
  - DLLxCR[DLLEN] = 1h
  - DLLxCR[OVRDEN] = 0h
  - Other fields in DLLxCR must be kept at their reset values (all zeroes). After the register settings above are completed, set the DLL with following steps.
    1. Reset the DLL(DLLxCR[DLLRESET] or MCR0[SWRESET]).
    2. Wait for STS2[xREFLOCK] and STS2[xSLVLOCK] being asserted.

3. Wait for equal or more than 512 serial root clock cycles.
4. Start read or write operation.

NOTE

If the serial root clock is slower than 100 MHz, DLL cannot lock on a half cycle of serial root clock. The delay cell number is limited in the delay chain. DLL must be configured as follows instead:

- DLLxCR[OVRDEN] = 1
- DLLxCR[OVRDVAL] = N

  Each delay cell in DLL is about 75 ps to 225 ps. The delay of DLL delay chain is (N × Delay_cell_delay). N is set based on the maximum DDR frequency (N ≤ equation). This value is calculated based on the MAX_DDR_FREQ parameter. N = 18. This value is a recommended value. If a failure occurs, the value might require adjustment in a real application.

  You can calculate the delay of one delay cell and obtain the value of N. If the serial root clock is lower than 100 MHz, the delay chain can't lock with DLLxCR[SLVDLYTARGET] = 0xF (half cycle of root clock delay). You need to set DLLxCR[SLVDLYTARGET] as 0xE (15/32 cycle delay) or less. When DLL is locked (read STS2[AREFLOCK] and STS2[ASLVLOCK]), the delay cell number = STS2[ASLVSEL] +1 and the delay value (one delay cell) = (total delay)/(delay cell number). Based on delay value, the N can be easily calculated.

- Other fields in DLLxCR must be kept at their reset values (all zeroes).

10.2.2.15 Data learning

The FlexSPI controller generates 16 clock phases (Phase 0–15) via delay cell line (DCL) with the selected sample clock. Clock phase 0 is the selected sample clock (DQS_IN) with no delay cell. 16 sampling blocks are implemented for both Port A and 16 sampling blocks are implemented for Port B.

During a learn instruction, FlexSPI compares the sampled data bits with the internal data learn pattern in the Data Learning Pattern (DLPR) register and determines the correct sampling clock phase. The phase selection is automatically updated after a learn instruction and applied to subsequent read instructions or sequences.

When data learning is disabled (MCR0[LEARNEN] = 0), FlexSPI always uses clock phase 0 to sample FlexSPI data lines. Attempting to execute a learn instruction results in an error flag (INTR[IPCMDERR] or INTR[AHBCMDERR]).

FlexSPI supports data learning in both SDR mode and DDR mode. Data learning is not supported if the sampling clock source is a flash-memory-provided read strobe (MCR0[RXCLKSRC] = 3h).

Data learning can achieve high read frequency, but the flash memory input timing still limits the highest frequency. Flash memory must receive the command code and flash address bits.

When data learning is enabled, FlexSPI uses clock phase 0 after reset and before executing a learn instruction. The clock phase selection is updated after FlexSPI executes the learn instruction. The selected sample clock phase can be polled via STS0[DATALEARNPHASEA] and STS0[DATALEARNPHASEB]. When data learning fails, the INTR[DATALEARNFAIL] interrupt flag is set, and the previous clock phase selection is kept. You can write 1 to MCR2[CLRLEARNPHASE] to reset the internal clock phase selection to clock phase 0.

10.2.2.15.1 Data learning with flash-memory-provided preamble bit

Certain flash devices support driving with preamble bits before driving read data in each read command sequence. Preamble bits are also called the data-learning pattern (DLP). The DLP is programmable via configuration register in the flash device.

For the specified read command sequence, flash memory returns preamble bits after dummy cycles and before returning read data.

For these flash devices, the operation flow with data learning is:

1. Configure the data learning pattern in Data Learning Pattern (DLPR) register.
2. Configure the same data learning pattern in the flash device by triggering an IP command.
3. Enable data learning in the flash device by triggering an IP command.
4. Configure LUT sequence with valid read command sequence containing the learn instruction.
5. Trigger a flash memory read command via AHB or IP command as normal.

NOTE

The first four steps are executed only once. They are not needed for every flash memory read command.

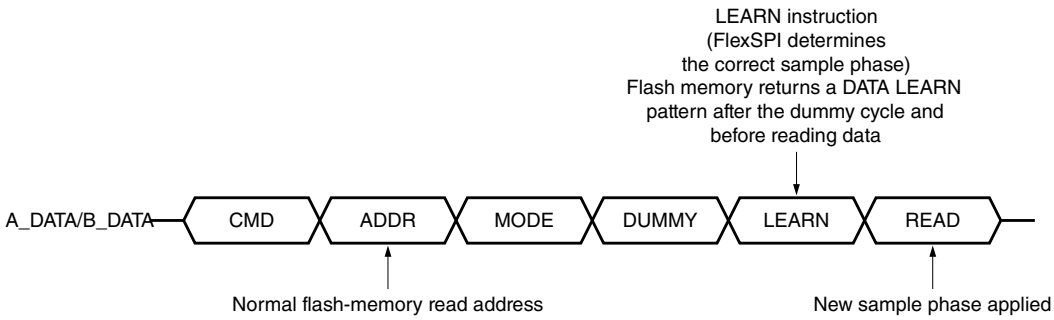


Figure 10-53. Data learning flow with flash-memory-provided preamble bit

10.2.2.15.2 Data learning without flash-memory-provided preamble bit

For flash devices that do not provide a preamble bit, the DLP cannot be returned automatically for each read command sequence. For these flash devices, the data learning sequence is:

1. Set data learning pattern in Data Learning Pattern (DLPR) register.
2. Reserve a certain flash memory area and program data (according to data learning pattern and flash memory access mode) to this area via IP command.
3. Configure LUT sequence with read sequence using a learn instruction instead of a read instruction. Trigger the read sequence to the reserved flash memory area via IP command.
4. Trigger the flash memory read command via AHB or IP command as normal. No learn instruction is needed in this read sequence.

NOTE

- The first three steps are executed only once. They are not needed for every flash memory read command.
- Step 4 should be executed with a certain interval to ensure that the internal sampling clock phase is adjusted in time.

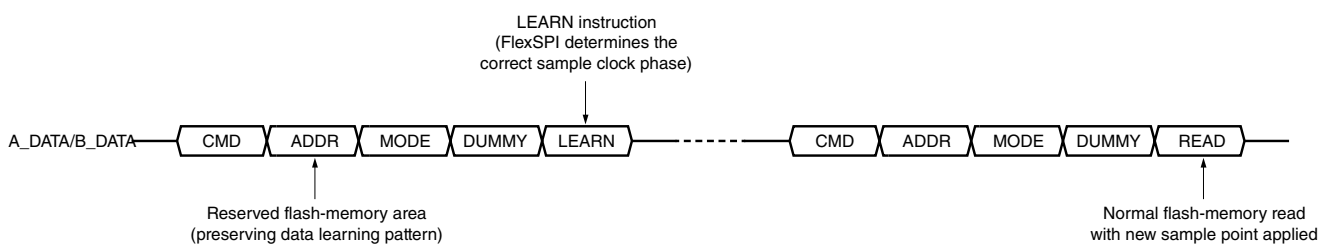


Figure 10-54. Data learning flow without flash-memory-provided preamble bit

NOTE

- For flash devices that do not provide a preamble bit, software overhead is required to read the reserved flash memory area at the interval.

These items determine the preprogramming data:

- Data Learning Pattern (DLPR register setting)
- DLP bit length
- Individual or Parallel mode for read command
- Octal, Quad, Dual, or Single mode for read command

Figure 10-55 shows a preprogramming data example where the DLPR value is 43h, eight bits, and flash memory is read in Quad mode and Parallel mode.

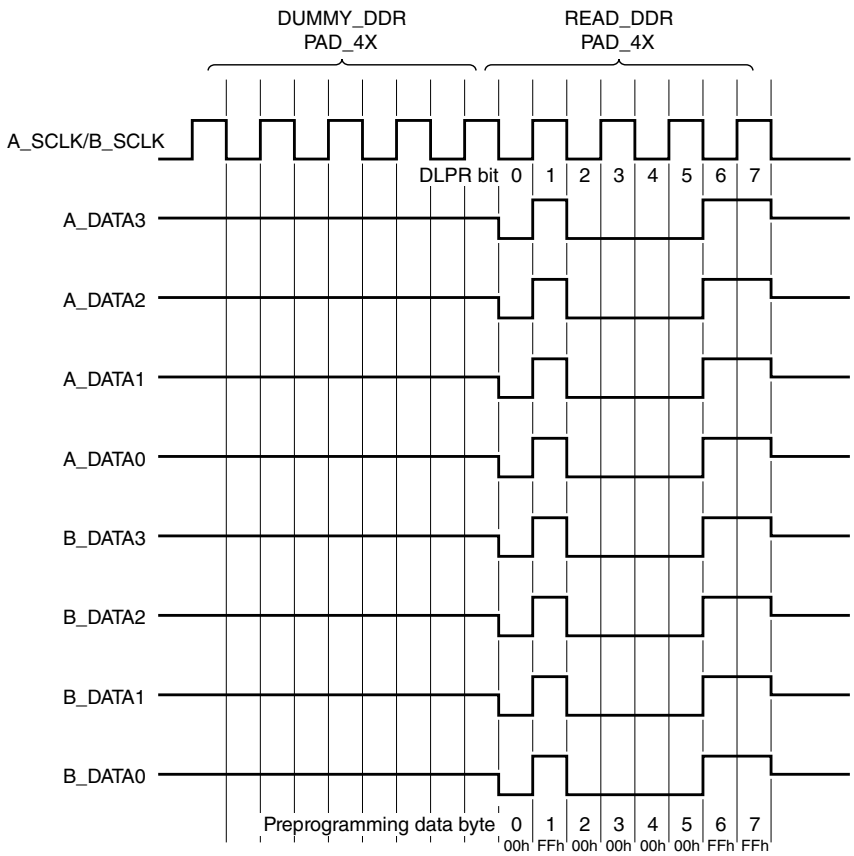


Figure 10-55. Preprogramming data for data learning

10.2.2.16 Execute-In-Place (XIP) enhanced mode

FlexSPI always supports XIP, regardless of whether external devices provide XIP enhanced mode. To support XIP, put program codes on an external device, then read or execute these codes directly via AHB read access to serial flash memory space. There is no configuration or status polling needed during AHB read access to the memory of the external device. The AHB receive buffer is fully transparent to software.

Certain devices provide XIP enhanced mode to improve code execution. In this mode, the command code is only sent in the first read instruction. It saves many cycles for command instructions and improves code execution. Devices enter or exit XIP enhanced mode via a special sequence that external devices specify. See the external device data sheet for more details.

Normally, a device enters XIP enhanced mode via this sequence:

1. Enable XIP enhanced mode in external flash device via IP command.
2. Send the first read sequence to external flash device with correct mode bits. Command code is included in this read sequence.
3. Send the subsequent read sequences to external flash device with correct mode bits. Command code is not included in these read sequences. Mode bits must be sent according to the flash specified; otherwise, the flash exits XIP enhanced mode.

The instruction JMP_ON_CS in FlexSPI must be used to support XIP enhanced mode in external devices. This instruction is only allowed in the AHB read command; otherwise, the instruction generates an IPCMDERR or AHBCDMERR interrupt (when the interrupt is enabled). When external devices do not support XIP enhanced mode, JMP_ON_CS must never be used. To support XIP enhanced mode:

- The first instruction in the read sequence must be the command instruction.
- The last valid instruction must be JMP_ON_CS (with operand 1h).

For the first AHB read command triggered, FlexSPI executes the instructions from the instruction pointer 0 in the sequence (which is the command instruction). After this sequence is executed, FlexSPI saves the operand in the JMP_ON_CS instruction as the start pointer for the next command to current device internally. For the next AHB read command triggered, FlexSPI executes from the instruction pointer 1h, bypassing the command instruction.

After XIP enhanced mode is started, it is possible to access flash via other commands like program commands. You must, however, use a special sequence to exit enhanced XIP mode before you can run a different command. Figure 10-56 indicates XIP operation with Flash XIP enhanced mode.

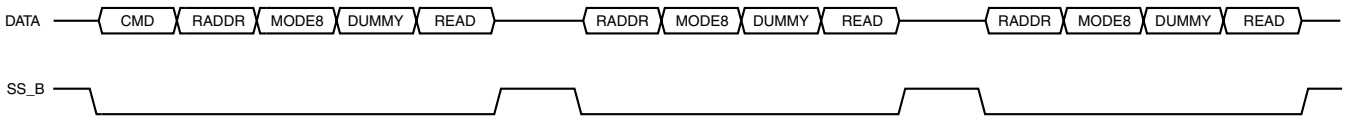


Figure 10-56. XIP Enhanced Mode Operation

10.2.3 External signals

This section provides the external signal information for the FlexSPI module.

Table 10-14. External Signal List

Signal name	Function	Direction	Description
A_SS0_B	Peripheral Chip Select Flash A1	O	Chip select for the serial flash device A1
A_SS1_B	Peripheral Chip Select Flash A2	O	Chip select for the serial flash device A2
B_SS0_B	Peripheral Chip Select Flash B1	O	Chip select for the serial flash device B1
B_SS1_B	Peripheral Chip Select Flash B2	O	Chip select for the serial flash device B2
A_SCLK	Serial Clock Flash A	O	Serial clock output to the serial flash device A. This clock runs at half the frequency of serial clock root in DDR mode, and at the same frequency as serial clock root in SDR mode. Clock output toggles during the entire flash memory access sequence.
B_SCLK	Serial Clock Flash B	O	Serial clock output to the serial flash device B. This clock runs at half the frequency of serial clock root in DDR mode, and at the same frequency as serial clock root in SDR mode. Clock output toggles during the entire flash memory access sequence. **NOTE:** B_SCLK may be used as the differential clock output of A_SCLK by writing 1 to MCR2[SCKBDIFFOPT].
A_DATAn	Serial I/O Flash A	I/O	Data I/O lines to and from the serial flash device A
B_DATAn	Serial I/O Flash B	I/O	Data I/O lines to and from the serial flash device B
A_DQS	Data Strobe Signal Flash A	I/O	Data strobe signal for port A. This signal has three functions: • Driven with read strobe by external device. Some flash devices provide the read strobe signal together with read data. In this case, if the external device drives this pad only when reading flash memory data, this pad may require a **pull-down** resistor. • Driven with latency information by external device. Some devices use this pin to indicate the dummy cycles needed (before

Table continues on the next page...

Table 10-14. External Signal List (continued)

Signal name	Function	Direction	Description
			program or read data transfer) such as HyperRAM or HyperFlash. • Loopback dummy read strobe. The FlexSPI controller provides internal dummy read strobe for flash memory read data. Higher read frequency can be achieved by looping back this dummy read strobe from the pad. This pin can be floated or have some capacitive load added at the board level to compensate for load on DATA or SCLK pins.
B_DQS	Data Strobe Signal Flash B	I/O	Similar to A_DQS

10.2.4 Initialization

The FlexSPI controller initialization sequence is:

1. Enable controller clocks (AHB clock, IP bus clock, or serial root clock) at system level.
2. To enter module stop mode, write 1 to MCR0[MDIS].
3. Configure module control registers: MCR0, MCR1, and MCR2. Do not change MCR0[MDIS].
4. When AHB commands are used, configure AHB Bus Control (AHBCR) and AHB Receive Buffer Control (AHBRXBUFxCR0) registers.
5. Configure flash control registers (FLSHxCR0, FLSHxCR1, or FLSHxCR2) according to external device type.
6. Configure DLL control register (DLLxCR) according to sample clock source selection.
7. To exit module stop mode, write 0 to MCR0[MDIS].
8. Configure LUT as needed for AHB command or IP command.
9. Optionally, to reset controller, write 1 to MCR0[SWRESET].

External devices must be configured via IP command normally after initialization. For example, the WRITE STATUS command performs device configuration for most serial NOR flash memory.

10.2.5 Application information

This section describes applications that FlexSPI supports.

10.2.5.1 Application on serial NOR flash device

This section provides the example LUT instruction sequences for serial NOR flash device (Cypress Flash S25FS128S).

10.2.5.1.1 Write enable command

Table 10-15 shows the WRITE ENABLE command sequence.

Table 10-15. WRITE ENABLE command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h	06h	Command name: WREN
1-7	STOP (00h)	0h	00h	

10.2.5.1.2 Write registers command

Table 10-16 shows the Write Registers command sequence.

Table 10-16. Write Registers command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	01h	Command name: WRR
1	WRITE_SDR	0h (Single)	1h or 2h	Data size is 1 byte or 2 bytes. Byte 0 is write data for Status Register 1; Byte 1 is write data for Configuration Register 1. IPCR1[IDATSZ] can override this value.
2-7	STOP (00h)	0h	00h	

10.2.5.1.3 Page Program command

Table 10-17 shows Page Program command sequence.

Table 10-17. PAGE PROGRAM command (Cypress serial NOR flash)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	02h or 12h	Command name: PP or 4PP PP is 3-byte address mode.

Table continues on the next page...

Table 10-17. PAGE PROGRAM command (Cypress serial NOR flash) (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
				4PP is 4-byte address mode.
1	ADDR_SDR	0h (Single)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	WRITE_SDR	0h (Single)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default programming data size. This value is ignored for AHB command.
3-7	STOP (00h)	0h	00h	

Table 10-18 shows Page Program command sequence (QPI mode).

Table 10-18. PAGE PROGRAM (QPI mode) command (Cypress serial NOR flash)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	2h (Quad)	02h or 12h	Command name: PP or 4PP PP is 3-byte address mode. 4PP is 4-byte address mode.
1	ADDR_SDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	WRITE_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default programming data size. This value is ignored for AHB command.
3-7	STOP (00h)	0h	00h	

10.2.5.1.4 Read Status 1 command

Table 10-19 shows READ STATUS 1 command sequence.

Table 10-19. READ STATUS 1 command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	05h	Command name: RDSR1
1	READ_SDR	0h (Single)	1h	1 byte for Status register 1
2-7	STOP (00h)	0h	00h	

10.2.5.1.5 Read command

Table 10-20 shows READ command sequence.

Table 10-20. READ command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	03h or 13h	Command name: READ or 4READ READ is 3-byte address mode. 4READ is 4-byte address mode.
1	ADDR_SDR	0h (Single)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	READ_SDR	0h (Single)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
3-7	STOP (00h)	0h	00h	

10.2.5.1.6 Fast Read command

Table 10-21 shows Fast Read command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[3:0] = 8h.

Table 10-21. FAST_READ command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	0Bh or 0Ch	Command name: FAST_READ or 4FAST_READ FAST_READ is 3-byte address mode. 4FAST_READ is 4-byte address mode.
1	ADDR_SDR	0h (Single)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	DUMMY_SDR	0h (Single)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
3	READ_SDR	0h (Single)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
4-7	STOP (00h)	0h	00h	

10.2.5.1.7 Dual IO Fast Read command

Table 10-22 shows Dual IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[3:0] = 8h, Continuous Read mode.

Table 10-22. Dual IO FAST_READ command (Continuous Read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	BBh or BCh	Command name: DIOR or 4DIOR DIOR is 3-byte address mode. 4DIOR is 4-byte address mode.
1	ADDR_SDR	1h (Dual)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	1h (Dual)	Axh	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_SDR	1h (Dual)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	1h (Dual)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
5	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
6-7	STOP (00h)	0h	00h	

Table 10-23 shows Dual IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[3:0] = 8h, Noncontinuous Read mode.

Table 10-23. Dual IO FAST_READ command (Noncontinuous Read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	BBh or BCh	Command name: DIOR or 4DIOR DIOR is 3-byte address mode. 4DIOR is 4-byte address mode.
1	ADDR_SDR	1h (Dual)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	1h (Dual)	Any value other than Axh	Exit continuous read mode or remain in noncontinuous read mode.
3	DUMMY_SDR	1h (Dual)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	1h (Dual)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading

Table continues on the next page...

Table 10-23. Dual IO FAST_READ command (Noncontinuous Read mode) (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
				data size. This value is ignored for AHB command.
5-7	STOP (00h)	0h	00h	

10.2.5.1.8 Quad IO Fast Read command

Table 10-24 shows Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 0, CR2V[3:0] = 8h, Non-QPI mode, Noncontinuous read mode.

Table 10-24. Quad IO FAST_READ command (Non-QPI mode, Noncontinuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	EBh or ECh	Command name: QIOR or 4QIOR QIOR is 3-byte address mode. 4QIOR is 4-byte address mode.
1	ADDR_SDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	2h (Quad)	Any value other than Axh	Exit continuous read mode or remain in noncontinuous read mode.
3	DUMMY_SDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
5-7	STOP (00h)	0h	00h	

Table 10-25 shows Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 0, CR2V[3:0] = 8h, Non-QPI mode, Continuous read mode.

Table 10-25. Quad IO FAST_READ command (Non-QPI mode, Continuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	EBh or ECh	Command name: QIOR or 4QIOR QIOR is 3-byte address mode. 4QIOR is 4-byte address mode.

Table continues on the next page...

Table 10-25. Quad IO FAST_READ command (Non-QPI mode, Continuous read mode) (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
1	ADDR_SDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	2h (Quad)	A0h	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_SDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
5	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
6-7	STOP (00h)	0h	00h	

Table 10-26 shows Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 1, CR2V[3:0] = 8h, QPI mode, Continuous read mode.

Table 10-26. Quad IO FAST_READ command (QPI mode, Continuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	2h (Quad)	EBh or ECh	Command name: QIOR or 4QIOR QIOR is 3-byte address mode. 4QIOR is 4-byte address mode.
1	ADDR_SDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	2h (Quad)	A0h	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_SDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
5	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
6-7	STOP (00h)	0h	00h	

10.2.5.1.9 DDR Quad IO Fast Read command

Table 10-27 shows DDR Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 0, CR2V[3:0] = 8h, Non-QPI mode, Non-continuous read mode.

Table 10-27. DDR Quad IO FAST_READ command (Non-QPI mode, Noncontinuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	EDh or EEh	Command name: QIOR_DDR or 4QIOR_DDR QIOR_DDR is 3-byte address mode. 4QIOR_DDR is 4-byte address mode.
1	ADDR_DDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_DDR	2h (Quad)	Any value other than Axh	Exit continuous read mode or keep in noncontinuous read mode.
3	DUMMY_DDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	LEARN_DDR	2h (Quad)	1h	DLP is 8 bits.
5	READ_DDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
6-7	STOP (00h)	0h	00h	

Table 10-28 shows DDR Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration register bits CR2V[7] = 0, CR2V[6] = 0, CR2V[3:0] = 8h, Non-QPI mode, Continuous read mode.

Table 10-28. DDR Quad IO FAST_READ command (Non-QPI mode, Continuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	EDh or EEh	Command name: QIOR_DDR or 4QIOR_DDR QIOR_DDR is 3-byte address mode. 4QIOR_DDR is 4-byte address mode.
1	ADDR_DDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_DDR	2h (Quad)	A0h	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_DDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.

Table continues on the next page...

Table 10-28. DDR Quad IO FAST_READ command (Non-QPI mode, Continuous read mode) (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
4	LEARN_DDR	2h (Quad)	1h	DLP is 8 bits.
5	READ_DDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
6	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
7	STOP (00h)	0h	00h	

Table 10-29 shows DDR Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 1, CR2V[3:0] = 8h, QPI mode, Continuous read mode.

Table 10-29. DDR Quad IO FAST_READ command (QPI mode, Continuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	2h (Quad)	EDh or EEh	Command name: QIOR_DDR or 4QIOR_DDR QIOR_DDR is 3-byte address mode. 4QIOR_DDR is 4-byte address mode.
1	ADDR_DDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_DDR	2h (Quad)	A0h	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_DDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	LEARN_DDR	2h (Quad)	1h	DLP is 8 bits.
5	READ_DDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
6	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
7	STOP (00h)	0h	00h	

10.2.5.2 Application on HyperBus device

This section provides the example LUT instruction sequences for HyperBus device (Cypress RPC flash, HyperRam, or HyperFlash).

10.2.5.2.1 HyperFlash

This section provides example sequences for HyperFlash devices (Cypress S26KS series).

Table 10-30 shows the read status command sequence.

Table 10-30. Read status command

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0 (Write - Addr = 555h, Data = 70h)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W# = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 0000AAh (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	AAh	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 05h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	05h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 0070h
	7	CMD_DDR	3h (Octal)	70h	
1 (Read - Addr = xxx, Data = Status register data)	0	CMD_DDR	3h (Octal)	A0h	CA bit 47: (R/W#) = 1h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: 24 bits
	2	CADDR_DDR	3h (Octal)	10h	Column Address: (13 zero bits + 3 valid bits)
	3	DUMMY_RWDS_DDR	3h (Octal)	0Bh	When latency count = 11
	4	READ_DDR	0h (Octal)	4h	4-Byte read
	5-7	STOP (00h)	0h	00h	

Table 10-31 shows the read (memory) command sequence.

Table 10-31. Read (memory) command

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0 (Read - Addr = xxx, Data = memory data)	0	CMD_DDR	3h (Octal)	A0h	CA bit 47: (R/W#) = 1h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h

Table continues on the next page...

Table 10-31. Read (memory) command (continued)

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
	1	RADDR_DDR	3h (Octal)	18h	Row Address: 24 bits
	2	CADDR_DDR	3h (Octal)	10h	Column Address: (13 zero bits + 3 valid bits)
	3	DUMMY_RWDS_DDR	3h (Octal)	0Bh	When latency count = 11
	4	READ_DDR	3h (Octal)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
	5-7	STOP (00h)	0h	00h	

Table 10-32 shows the word program command sequence.

Table 10-32. Word program command

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0 (Write - Addr = 555h, Data = AAh)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 0000AAh (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	AAh	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 05h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	05h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 00AAh
	7	CMD_DDR	3h (Octal)	AAh	
1 (Write - Addr = 2AAh, Data = 55h)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 000055h (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	55h	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 02h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	02h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 0055h
	7	CMD_DDR	3h (Octal)	55h	
2	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h

Table continues on the next page...

Table 10-32. Word program command (continued)

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
(Write - Addr = 555h, Data = A0h)					CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 0000AAh (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	AAh	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 05h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	55h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 00A0h
	7	CMD_DDR	3h (Octal)	A0h	
3 (Word Program)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: 24 bits
	2	CADDR_DDR	3h (Octal)	10h	Column Address: (13 zero bits + 3 valid bits)
	3	WRITE_DDR	3h (Octal)	02h	2-Byte written data
	4-7	STOP (0h)	0h	00h	

Table 10-33 shows Write-to-Buffer and Program-Buffer-to-Flash command sequence.

Table 10-33. Write-to-Buffer and Program-Buffer-to-Flash command

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0 (Write - Addr = 555h, Data = AAh)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 0000AAh (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	AAh	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 05h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	05h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 00AAh
	7	CMD_DDR	3h (Octal)	AAh	
1	0	CMD_DDR	0h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h

Table continues on the next page...

Table 10-33. Write-to-Buffer and Program-Buffer-to-Flash command (continued)

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
(Write - Addr = 2AAh, Data = 55h)					CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 000055h (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	55h	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 02h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	02h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 0055h
	7	CMD_DDR	3h (Octal)	55h	
2 (Write - Addr = SA, Data = 25h)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: SA (24 bit) SA is sector address. Write SA to IPCR0[SFAR].
	2	CADDR_DDR	3h (Octal)	10h	Column Address: 13 zero bits + 3 valid bits
	3	CMD_DDR	3h (Octal)	00h	Write Data: 0025h
	4	CMD_DDR	3h (Octal)	25h	
2 (Write - Addr = SA, Data = WC)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: SA (24 bit) SA is sector address. Write SA to IPCR0[SFAR].
	2	CADDR_DDR	3h (Octal)	10h	Column Address: 13 zero bits + 3 valid bits
	3	CMD_DDR	3h (Octal)	WC	Write Data: WC
	4	CMD_DDR	3h (Octal)		WC is word count.
3 - N (Write - Addr = WBL, Data = PD) N is the word count + 2	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: WBL (24 bit) WBL is write buffer location. Write WBL to IPCR0[SFAR].
	2	CADDR_DDR	3h (Octal)	10h	Column Address: 13 zero bits + 3 valid bits

Table continues on the next page...

Table 10-33. Write-to-Buffer and Program-Buffer-to-Flash command (continued)

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
	3	WRITE_DDR	3h (Octal)	02h	2-Byte write data
	4-7	STOP (0h)	0h	00h	
N + 1 (Write - Addr = SA, Data = 29) Program Buffer to Flash	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: SA (24 bit) SA is sector address. Write SA to IPCR0[SFAR].
	2	CADDR_DDR	3h (Octal)	10h	Column Address: 13 zero bits + 3 valid bits
	3	CMD_DDR	3h (Octal)	00h	Write Data: 29h
	4	CMD_DDR	3h (Octal)	29h	
	5-7	STOP (0h)	0h	00h	

10.2.5.2.2 HyperRAM

This section provides example sequences for HyperRAM (Cypress S27KL series).

The Read (memory) command sequence is same as HyperFlash. Table 10-34 shows the Write (memory) command sequence.

Table 10-34. Write (memory) command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
1	RADDR_DDR	3h (Octal)	18h	Row Address: 24 bits
2	CADDR_DDR	3h (Octal)	10h	Column Address: (13 zero bits + 3 valid bits)
3	DUMMY_RWDS_DDR	3h (Octal)	0Bh	When latency count = 11
4	WRITE_DDR	3h (Octal)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default writing data size. This value is ignored for AHB command.
5-7	STOP (00h)	0h	00h	

10.2.5.3 Application on Serial NAND flash device

This section provides example LUT instruction sequences for serial NAND flash device (Micron Flash MT29 series). The operation of serial NAND flash is similar to serial NOR flash.

The read operation sequence is:

- Page read. Transfer the data from the NAND flash array to the cache register.
- Get feature to read the status.
- Random data read

Table 10-35 shows the page read command sequence.

Table 10-35. Page Read command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	1h (Single)	13h	Command code: 13h
1	RADDR_SDR	1h (Single)	18h	Row address: 24 bit
2-7	STOP (0h)	0h	00h	

Table 10-36 shows Get Feature command sequence.

Table 10-36. Get Feature command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	0Fh	Command code: 0Fh
1	CMD_SDR	1h (Single)	C0h	Status register address (C0h)
2	READ_SDR	0x1 (Single)	02h	2 bytes read data
3-7	STOP (0h)	0h	00h	

Table 10-37 shows Random Data Read command sequence.

Table 10-37. Read from Cache x4 command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	1h (Single)	6Bh	Command code: 6Bh
1	MODE4_SDR	1h (Single)	0h or 1h	If plane selection is one, software should decode the flash address and write 1h to mode bits. If plane selection is zero, software should write 0h to mode bits.

Table continues on the next page...

Table 10-37. Read from Cache x4 command (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
				Plane selection bit is 18th bit of flash address. If NAND flash size is less than 4 Gbit, plane selection is always zero.
2	CADDR_SDR	1h (Single)	0Ch	Column address: 12 bit
3	DUMMY_SDR	2h (Quad)	08h	Dummy cycle number: 8 (serial root clock)
4	READ_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size.
5-7	STOP (0x0)	0x0	0x00	

The program operation sequence is:

- Write enable
- Program Load. Transfer the write data to the cache register.
- Program Execute
- Get Feature to read the status.

Table 10-38 shows Program Load command sequence.

Table 10-38. Program Load command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	1h (Single)	02h	Command code: 02h
1	MODE4_SDR	1h (Single)	0h or 1h	If plane selection is one, software should decode the flash address and write 1h to mode bits. If plane selection is zero, software should write 0h to mode bits. Plane selection bit is 18th bit of flash address. If NAND flash size is less than 4 Gbit, plane selection is always zero.
2	CADDR_SDR	1h (Single)	0Ch	Column address: 12 bit
3	WRITE_SDR	1h (Single)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default writing data size.
4-7	STOP (0h)	0h	00h	

Table 10-39 shows Program Execute command sequence.

Table 10-39. Program Execute command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	1h (Single)	10h	Command code: 10h
1	RADDR_SDR	1h (Single)	18h	Row Address: 24 bit
2-7	STOP (0h)	0h	00h	

10.2.5.4 Application on FPGA device

An FPGA device can be accessed via both AHB and IP commands. All AHB accesses to FPGA are transparent to the software driver (no software intervention). An FPGA device may have some special requirements.

Table 10-40. Special requirements for FPGA devices

Condition	How to manage the condition
Device type may be different on A1, A2, B1, or B2.	Write 0 to MCR2[SAMEDEVICEEN] and configure the FLSHxCR0 and FLSHxCR1 registers separately for up to four external devices.
Device requires different wait cycle for programming.	The AHB write wait cycle number can be set separately for these four external devices (via FLSHxCR2[AWRWAIT]). Software can configure the sequences in LUT with different dummy instructions (operand determines dummy cycle). FlexSPI holds the AHB bus ready for this wait time; AHB bus performance may decrease when this wait time is very long.
Device requires different wait cycle for reading.	The AHB read sequence index and sequence number can be set separately for these four external devices (via FLSHxCR2[ARDSEQID] and FLSHxCR2[ARDSEQNUM]). Software can configure the sequences in LUT with different dummy instructions (operand determines dummy cycle).
Device may be sensitive to read instruction clock cycle number.	If its internal memory is implemented similar to a FIFO, the device is sensitive to the read instruction clock cycle number. Software can send the data size information to the external device via DATSZ instruction. The FPGA device decodes the data size information and determines how many data bytes must be popped.
Device may require interval time between chip select valid.	This interval can be managed via FLSHxCR1[CSINTERVAL].
Device may use SCLK as reference clock for its internal PLL.	SCLK must be free-running and the clock frequency must be stable. To achieve this configuration, write 1 to MCR0[SCKFREERUNEN] and only use SDR sequences.

10.2.5.5 Overview of error categories, flags, and triggered sources

Table 10-41 provides an overview of the categories, flags, and triggered sources of errors in FlexSPI.

Table 10-41. Error category, triggered sources, and flags

Error Category	Triggered Sources	Description	Error Flags
Command grant error	AHB write command	Command grant timeout	INTR[AHBCMDGE] is set. AHB bus error response
	AHB read command		INTR[AHBCMDGE] is set. AHB bus error response
	IP command		INTR[IPCMDGE] is set.
Command check error	AHB write command	• AHB write command with JMP_ON_CS instruction used in the sequence • Unknown instruction opcode in the sequence. • Instruction DUMMY_SDR or DUMMY_RWDS_SDR used in DDR sequence. • Instruction DUMMY_DDR or DUMMY_RWDS_DDR used in SDR sequence.	INTR[AHBCMDERR] is set. Command is not executed when an error is detected in command check
	AHB read command	• Unknown instruction opcode in the sequence. • Instruction DUMMY_SDR or DUMMY_RWDS_SDR used in DDR sequence. • Instruction DUMMY_DDR or DUMMY_RWDS_DDR used in SDR sequence.	INTR[AHBCMDERR] is set. Command is not executed when an error is detected in command check
	IP command	• IP command with JMP_ON_CS instruction used in the sequence. • Unknown instruction opcode in the sequence • Instruction DUMMY_SDR or DUMMY_RWDS_SDR used in DDR sequence. • Instruction DUMMY_DDR or DUMMY_RWDS_DDR used in SDR sequence. • Flash boundary across	INTR[IPCMDERR] is set. Command is not executed when an error is detected in command check.
Command execution error	AHB write command	Command timeout during execution	INTR[AHBCMDERR] is set. INTR[SEQTIMEOUT] is set. An AHB bus error response occurs, except in following cases: • Flush triggers AHB write command (INCR

Table continues on the next page...

Table 10-41. Error category, triggered sources, and flags (continued)

Error Category	Triggered Sources	Description	Error Flags
			burst ended with AHB_TX_BUF not empty). • AHB bufferable write access and bufferable enabled (AHBCR[BUFFERABLEEN] = 1).
	AHB read command		INTR[AHBCMDERR] is set. INTR[SEQTIMEOUT] is set. AHB bus error response
	IP command		INTR[IPCMDERR] is set. INTR[SEQTIMEOUT] is set.
AHB bus timeout	AHB write command	AHB bus timeout (no bus ready return)	INTR[AHBBUSTIMEOUT] is set. AHB bus error response
	AHB read command		
Data learning failed	Any command	No valid sample clock phase found after learn instruction is executed.	INTR[DATALEARNFAIL] is set.

10.2.6 Memory map and register definition

This section includes the FlexSPI module memory map and detailed descriptions of all registers.

10.2.6.1 Register access

All registers can be accessed with 8-bit, 16-bit, and 32-bit width operations. Never change the values of reserved fields in control registers. Changing the values of reserved fields may impact the normal functioning of the controller.

NOTE

When FlexSPI can access sensitive memory contents, the FlexSPI controller should protect those contents using resource isolation such as XRDC or TrustZone. Ensure that only trusted

software is allowed to access the FlexSPI controller register interface.

10.2.6.2 FlexSPI register descriptions

10.2.6.2.1 FlexSPI memory map

FlexSPI base address: 30BB_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Module Control 0 (MCR0)	32	RW	FFFF_8002h
4h	Module Control 1 (MCR1)	32	RW	FFFF_FFFFh
8h	Module Control 2 (MCR2)	32	RW	2000_81F7h
Ch	AHB Bus Control (AHBCR)	32	RW	0000_0018h
10h	Interrupt Enable (INTEN)	32	RW	0000_0000h
14h	Interrupt (INTR)	32	RW	0000_0000h
18h	LUT Key (LUTKEY)	32	RW	5AF0_5AF0h
1Ch	LUT Control (LUTCR)	32	RW	0000_0002h
20h	AHB Receive Buffer 0 Control 0 (AHBRXBUF0CR0)	32	RW	8000_0020h
24h	AHB Receive Buffer 1 Control 0 (AHBRXBUF1CR0)	32	RW	8001_0020h
28h	AHB Receive Buffer 2 Control 0 (AHBRXBUF2CR0)	32	RW	8002_0020h
2Ch	AHB Receive Buffer 3 Control 0 (AHBRXBUF3CR0)	32	RW	8003_0020h
30h	AHB Receive Buffer 4 Control 0 (AHBRXBUF4CR0)	32	RW	8004_0020h
34h	AHB Receive Buffer 5 Control 0 (AHBRXBUF5CR0)	32	RW	8005_0020h
38h	AHB Receive Buffer 6 Control 0 (AHBRXBUF6CR0)	32	RW	8006_0020h
3Ch	AHB Receive Buffer 7 Control 0 (AHBRXBUF7CR0)	32	RW	8007_0020h
60h	Flash Control 0 (FLSHA1CR0)	32	RW	0001_0000h
64h	Flash Control 0 (FLSHA2CR0)	32	RW	0001_0000h
68h	Flash Control 0 (FLSHB1CR0)	32	RW	0001_0000h
6Ch	Flash Control 0 (FLSHB2CR0)	32	RW	0001_0000h
70h	Flash Control 1 (FLSHA1CR1)	32	RW	0000_0063h
74h	Flash Control 1 (FLSHA2CR1)	32	RW	0000_0063h
78h	Flash Control 1 (FLSHB1CR1)	32	RW	0000_0063h
7Ch	Flash Control 1 (FLSHB2CR1)	32	RW	0000_0063h
80h	Flash Control 2 (FLSHA1CR2)	32	RW	0000_0000h
84h	Flash Control 2 (FLSHA2CR2)	32	RW	0000_0000h
88h	Flash Control 2 (FLSHB1CR2)	32	RW	0000_0000h
8Ch	Flash Control 2 (FLSHB2CR2)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
94h	Flash Control 4 (FLSHCR4)	32	RW	0000_0000h
A0h	IP Control 0 (IPCR0)	32	RW	0000_0000h
A4h	IP Control 1 (IPCR1)	32	RW	0000_0000h
B0h	IP Command (IPCMD)	32	RW	0000_0000h
B4h	Data Learning Pattern (DLPR)	32	RW	0000_0000h
B8h	IP Receive FIFO Control (IPRXFCR)	32	RW	0000_0000h
BCh	IP Transmit FIFO Control (IPTXFCR)	32	RW	0000_0000h
C0h	DLL Control 0 (DLLACR)	32	RW	0000_0100h
C4h	DLL Control 0 (DLLBCR)	32	RW	0000_0100h
E0h	Status 0 (STS0)	32	R	0000_0002h
E4h	Status 1 (STS1)	32	R	0000_0000h
E8h	Status 2 (STS2)	32	R	0100_0100h
ECh	AHB Suspend Status (AHBSPNDSTS)	32	R	0000_0000h
F0h	IP Receive FIFO Status (IPRXFSTS)	32	R	0000_0000h
F4h	IP Transmit FIFO Status (IPTXFSTS)	32	R	0000_0000h
100h - 17Ch	IP Receive FIFO Data a (RFDR0 - RFDR31)	32	R	0000_0000h
180h - 1FCh	IP TX FIFO Data a (TFDR0 - TFDR31)	32	W	0000_0000h
200h - 3FCh	Lookup Table a (LUT0 - LUT127)	32	RW	Table 10-85
400h	AHB Controller ID 0 Control (HMSTR0CR)	32	RW	0000_0000h
404h	AHB Controller ID 1 Control (HMSTR1CR)	32	RW	0000_0000h
408h	AHB Controller ID 2 Control (HMSTR2CR)	32	RW	0000_0000h
40Ch	AHB Controller ID 3 Control (HMSTR3CR)	32	RW	0000_0000h
410h	AHB Controller ID 4 Control (HMSTR4CR)	32	RW	0000_0000h
414h	AHB Controller ID 5 Control (HMSTR5CR)	32	RW	0000_0000h
418h	AHB Controller ID 6 Control (HMSTR6CR)	32	RW	0000_0000h
41Ch	AHB Controller ID 7 Control (HMSTR7CR)	32	RW	0000_0000h
420h	HADDR REMAP Start Address (HADDRSTART)	32	RW	0000_0000h
424h	HADDR REMAP END ADDR (HADDREND)	32	RW	0000_0000h
428h	HADDR Remap Offset (HADDROFFSET)	32	RW	0000_0000h

10.2.6.2.2 Module Control 0 (MCR0)

10.2.6.2.2.1 Offset

Register	Offset
MCR0	0h

10.2.6.2.2.2 Function

Controls basic functions of FlexSPI module

10.2.6.2.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AHBGRANTWAIT								IPGRANTWAIT							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LEARNEN	SCKFREERUNEN	COMBINATIONEN	DOZEEN	HSEN	SERCLKDIV			ATDFEN	ARDFEN	RXCLKSRC		Reserved		MDIS	SWRESET
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

10.2.6.2.2.4 Fields

Field	Function
31-24 AHBGRANTWAIT	Timeouts Wait Cycle for AHB command Grant Sets duration of timeout wait cycle for AHB commands. If the arbitrator does not grant the AHB-triggered command, it times out after (AHBGRANTTIMEOUT × 1024) AHB clock cycles. When the pending command sequence is IP-triggered and the read or write data size is too large, this grant timeout may occur. When an AHB command grant timeout occurs, an INTR[AHBCMDGE] interrupt is generated. When INTEN[AHBCMDGEEN] = 1, the arbitrator ignores AHB commands. **NOTE:** This field is for debugging only. Do not change the value from its default.
23-16 IPGRANTWAIT	Timeout Wait Cycle for IP Command Grant Sets duration of timeout wait cycle for IP commands. If the arbitrator does not grant the IP-triggered command, it times out after (IPGRANTTIMEOUT × 1024) AHB clock cycles. When the pending command sequence is AHB-triggered and the read or write data size is too large, this grant timeout may occur. When IP command grant timeout occurs, an INTR[IPCMDGE] interrupt is generated. When INTEN[IPCMDGEEN] = 1, the arbitrator ignores IP commands. **NOTE:** This field is for debugging only. Do not change the value from its default.
15 LEARNEN	Data Learning Enable Enables data learning feature. When data learning is disabled, the sampling clock phase 0 is always used for receive data sampling, even when a learn instruction is correctly executed. See Data learning. 0b - Disable 1b - Enable
14	SCLK Free-running Enable

Table continues on the next page...

Field	Function
SCKFREERUNEN	Enables free-running SCLK output. For FPGA applications, the external device may use SCLK as a reference clock to its internal PLL. 0b - Disable 1b - Enable
13 COMBINATIONEN	Combination Mode Enable Enables combination mode, which supports flash Octal mode access. When Port A and Port B data are four bits wide, this mode combines Port A and B Data pins (A_DATA[3:0] and B_DATA[3:0]). When Port A and Port B data are eight bits wide, combination mode is not supported; write 0 to this field in this case. 0b - Disable 1b - Enable
12 DOZEEN	Doze Mode Enable Enables Doze mode. When enabled, AHB clock and serial clock are gated off when there is a Doze-mode request from the system. 0b - Disable 1b - Enable
11 HSEN	Half Speed Serial Flash Memory Access Enable Enables the clock divider to provide a half-speed clock to external serial flash devices (A_SCLK/B_SCLK) for all commands in SDR and DDR mode. Write 1 to MCR0[MDIS] before changing the value of this field. Failure to do so may cause issues in the internal logic or state machine. 0b - Disable 1b - Enable
10-8 SERCLKDIV	Serial Root Clock Divider Sets divider value for serial root clock. The serial root clock can be divided inside FlexSPI . See Clocking. **NOTE:** Do not modify this field while FlexSPI is active (MCR0[MDIS] = 0). 000b - Divided by 1 001b - Divided by 2 010b - Divided by 3 011b - Divided by 4 100b - Divided by 5 101b - Divided by 6 110b - Divided by 7 111b - Divided by 8
7 ATDFEN	AHB Write Access to IP Transmit FIFO Enable Enables AHB write access to IP transmit FIFO. See Writing data to IP transmit FIFO. 0b - AHB write access disabled. IP bus writes to IP transmit FIFO. AHB bus write access to IP transmit FIFO memory space produces bus error. 1b - AHB write access enabled. AHB bus writes to IP transmit FIFO. IP Bus write access to IP transmit FIFO memory space is ignored and causes no bus error.
6 ARDFEN	AHB Read Access to IP Receive FIFO Enable Enables AHB read access to IP receive FIFO. See Reading data from IP receive FIFO. 0b - AHB read access disabled. IP bus reads IP receive FIFO. AHB Bus read access to IP receive FIFO memory space produces bus error. 1b - AHB read access enabled. AHB bus reads IP receive FIFO. IP Bus read access to IP receive FIFO memory space returns data zero and causes no bus error.
5-4 RXCLKSRC	Sample Clock Source for Flash Reading Selects sampling clock source for flash memory reading. See Receive clock source features. 00b - Dummy Read strobe that FlexSPI generates, looped back internally 01b - Dummy Read strobe that FlexSPI generates, looped back from DQS pad 10b - Reserved

Table continues on the next page...

Field	Function
	11b - Flash-memory-provided read strobe and input from DQS pad
3-2 —	Reserved
1 MDIS	Module Disable Disables FlexSPI module. When the module is disabled, AHB and serial clock are gated off internally to save power. Only register access is allowed, except for the LUT, IP receive FIFO, and IP transmit FIFO. 0b - No impact 1b - Module disable
0 SWRESET	Software Reset Resets the internal FlexSPI state machine and aborts any transactions in progress. Hardware automatically writes 0 to this field after software reset is done. Configuration registers are not reset. 0b - No impact 1b - Software reset

10.2.6.2.3 Module Control 1 (MCR1)

10.2.6.2.3.1 Offset

Register	Offset
MCR1	4h

10.2.6.2.3.2 Function

Controls basic functions of FlexSPI module

10.2.6.2.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SEQWAIT															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AHBBUSWAIT															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

10.2.6.2.3.4 Fields

Field	Function
31-16 SEQWAIT	Command Sequence Wait Sets wait time for command sequence. Command sequence execution times out and aborts after (SEQWAIT × 1024) serial root clock cycles. When this timeout occurs, if the interrupt is enabled (INTEN[SEQTIMEOUTEN] = 1), an INTR[SEQTIMEOUT] interrupt is generated. Also, the arbitrator ignores AHB commands. **NOTE:** You cannot write 0 to this field.
15-0 AHBBUSWAIT	AHB Bus Wait Sets wait time for AHB bus. When data is not received from or transmitted to serial flash memory space after (AHBBUSWAIT × 1024) AHB clock cycles, the read or write access times out. When this timeout occurs, the AHB bus receives an error response, and an interrupt is generated (INTR[AHBBUSTIMEOUT]). When INTR[AHBBUSTIMEOUT] = 1, the arbitrator ignores AHB commands. **NOTE:** You cannot write 0 to this field.

10.2.6.2.4 Module Control 2 (MCR2)

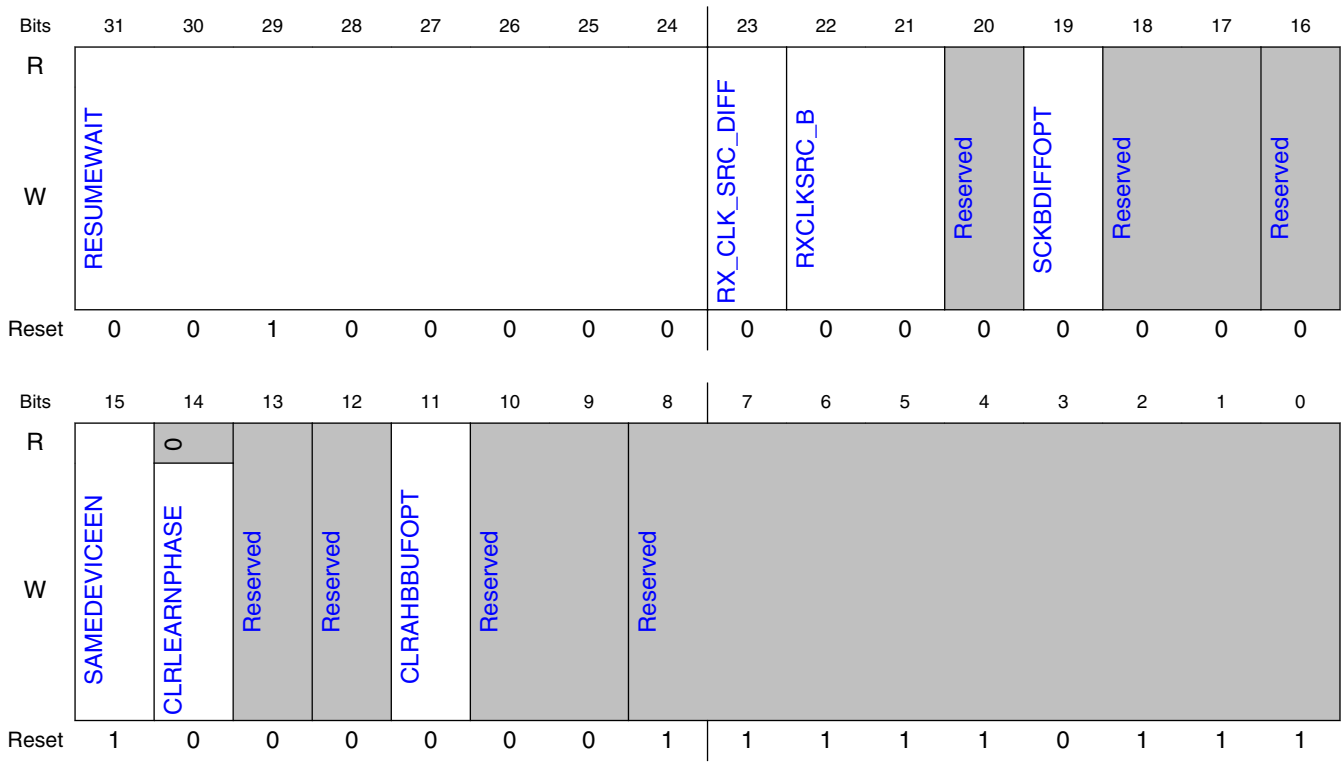
10.2.6.2.4.1 Offset

Register	Offset
MCR2	8h

10.2.6.2.4.2 Function

Controls basic functions of FlexSPI module.

10.2.6.2.4.3 Diagram



10.2.6.2.4.4 Fields

Field	Function
31-24 RESUMEWAIT	Resume Wait Duration Determines duration (in AHB clock cycles) to remain in idle state before suspended command sequence is resumed. See Command abort and suspend.
23 RX_CLK_SRC_DIFF	Sample Clock Source Different Selects whether to use the same clock source or a different clock source for Port A and Port B. 0b - Use MCR0[RXCLKSRC] for Port A and Port B. MCR2[RXCLKSRC_B] is ignored and MCR0[RXCLKSRC] selects the Sample Clock source for Flash Reading of both ports A and B. 1b - Use MCR0[RXCLKSRC] for Port A, and MCR2[RXCLKSRC_B] for Port B. MCR0[RXCLKSRC] selects the Sample Clock source for Flash Reading of port A (A_SCLK) and MCR2[RXCLKSRC_B] selects the Sample Clock source for Flash Reading of port B (B_SCLK).
22-21 RXCLKSRC_B	Port B Receiver Clock Source Selects Port B sample clock source selection for flash memory reading. 00b - Dummy read strobe that FlexSPI generates, looped back internally. 01b - Dummy read strobe that FlexSPI generates, looped back from DQS pad. 10b - SCLK output clock and looped back from SCLK padReserved 11b - Flash-memory-provided read strobe and input from DQS pad
20 —	Reserved
19	SCLK Port B Differential Output

Table continues on the next page...

Field	Function
SCKBDIFFOPT	Controls whether to use B_SCLK pad as the Port B SCLK output. Alternatively, B_SCLK pad can be used as A_SCLK differential clock output (inverted clock to A_SCLK). Before changing the value of this field, write 1 to MCR0[MDIS]. After changing the value of this field, write 1 to MCR0[SWRESET]. 0b - Use B_SCLK pad as port B SCLK clock output. Port B flash memory access is available. 1b - Use B_SCLK pad as port A SCLK inverted clock output (Differential clock to A_SCLK). Port B flash memory access is not available.
18-17 —	Reserved
16 —	Reserved
15 SAMEDEVICEEN	Same Device Enable Sets all external devices (A1, A2, B1, and B2) to be the same devices (in type and in size). 0b - In Individual mode, FLSHA1CRx and FLSHA2CRx, FLSHB1CRx and FLSHB2CRx settings are applied to Flash A1, A2, B1, B2 separately. In Parallel mode, FLSHA1CRx register setting is applied to Flash A1 and B1, FLSHA2CRx register setting is applied to Flash A2 and B2. FLSHB1CRx and FLSHB2CRx register settings are ignored. 1b - FLSHA1CR0, FLSHA1CR1, and FLSHA1CR2 register settings are applied to Flash A1, A2, B1, B2. FLSHA2CRx, FLSHB1CRx, and FLSHB2CRx settings are ignored.
14 CLRLEARNPHASE	Clear Learn Phase Selection Resets the sampling clock phase selection to 0. When 1 is written to this field, it becomes 0 again immediately. 0b - No impact 1b - Reset sample clock phase selection to 0
13 —	Reserved
12 —	Reserved
11 CLRAHBBUFOPT	Clear AHB Buffer Determines whether AHB receive and transmit buffers are cleared automatically when FlexSPI returns Stop mode ACK. If an AHB receive buffer or transmit buffer will be powered off in Stop mode, software should write 1 to this field. Otherwise, an AHB read access after exiting Stop mode may hit either buffer, but their data entries are invalid. 0b - Not cleared automatically 1b - Cleared automatically
10-9 —	Reserved
8-0 —	Reserved

10.2.6.2.5 AHB Bus Control (AHBCR)

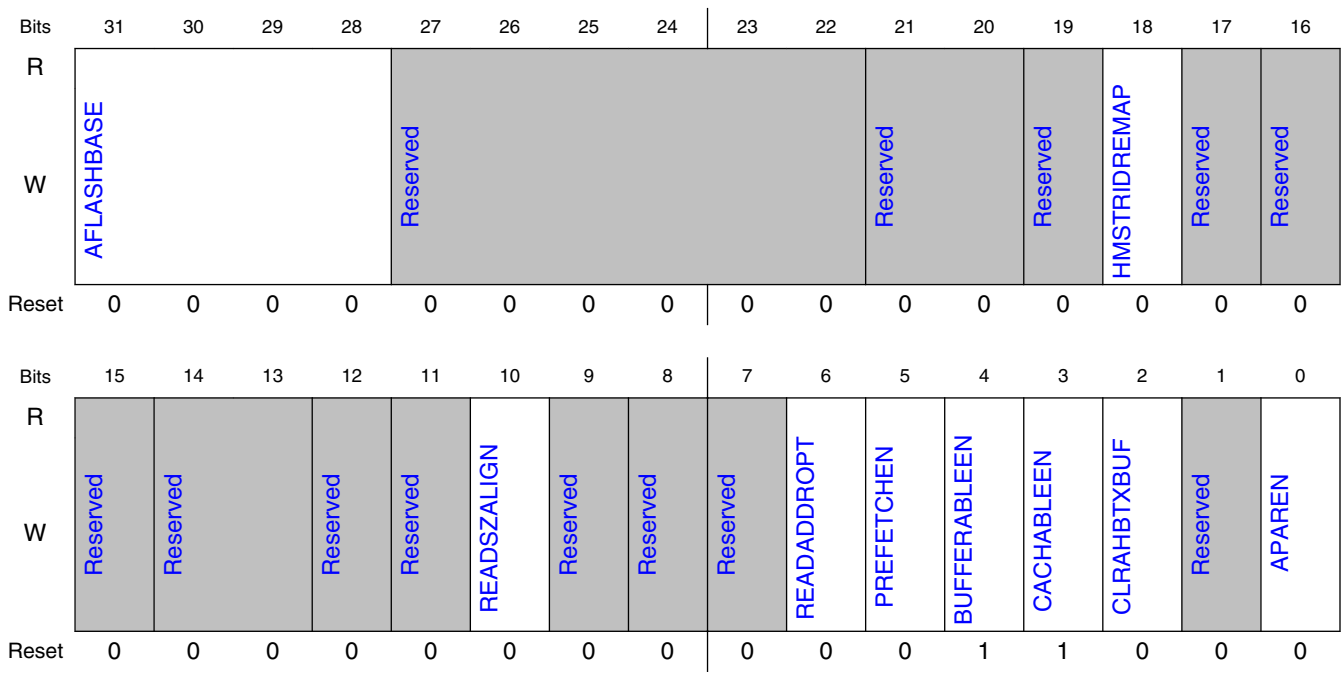
10.2.6.2.5.1 Offset

Register	Offset
AHBCR	Ch

10.2.6.2.5.2 Function

Controls AHB bus interface functions.

10.2.6.2.5.3 Diagram



10.2.6.2.5.4 Fields

Field	Function
31-28 AFLASHBASE	AHB Memory-Mapped Flash Base Address
27-22 —	Reserved
21-20 —	Reserved
19 —	Reserved

Table continues on the next page...

Field	Function
18 HMSTRIDREMAP	AHB Controller ID Remapping Enable Enables AHB controller ID remapping. See AHB controller ID remapping. 0b - Disable 1b - Enable
17 —	Reserved
16 —	Reserved
15 —	Reserved
14-13 —	Reserved
12 —	Reserved
11 —	Reserved
10 READSZALIGN	AHB Read Size Alignment Configures how AHB read size is determined. See Table 10-11. 0b - Register settings such as PREFETCH_EN determine AHB read size. 1b - AHB read size to up size to 8 bytes aligned, no prefetching
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 READADDROPT	AHB Read Address Option Removes AHB burst start address alignment limitation. When the FlexSPI controller is used for FPGA application, this field may be required for FlexSPI to fetch the exact byte number as an AHB burst. In this case, FPGA device should be non-word-addressable and this field must be 0. 0b - AHB read burst start address alignment is limited when flash memory is accessed in parallel mode or flash is word-addressable. 1b - AHB read burst start address alignment is not limited. FlexSPI fetches more data than the AHB burst requires for address alignment.
5 PREFETCHEN	AHB Read Prefetch Enable Enables AHB read prefetch. When enabled, FlexSPI fetches more flash read data than the current AHB burst requires. This action reduces the read latency for next AHB read access. **NOTE:** AHB read prefetch is enabled only when both this field and AHBRXBUFnCR0[PREFETCHEN] are 1. 0b - Disable 1b - Enable
4 BUFFERABLEEN	Bufferable Write Access Enable Enables AHB bus bufferable write access. This field affects the last beat of an AHB write access. See AHB write access to flash memory.

Table continues on the next page...

Field	Function
	0b - Disabled. For all AHB write accesses (bufferable or nonbufferable), FlexSPI returns AHB Bus Ready after transmitting all data and finishing command. 1b - Enabled. For AHB bufferable write access, FlexSPI returns AHB Bus Ready when the arbitrator grants the AHB command. FlexSPI does not wait for the AHB command to finish.
3 CACHABLEEN	Cacheable Read Access Enable Enables AHB bus cacheable read access. 0b - Disabled. When an AHB bus cacheable read access occurs, FlexSPI does not check whether it hit the AHB transmit buffer. 1b - Enabled. When an AHB bus cacheable read access occurs, FlexSPI first checks whether the access hit the AHB transmit buffer.
2 CLRAHBTXBUF	Clear AHB Transmit Buffer Clears status and pointers of AHB transmit buffer. When the clear operation completes, this field clears automatically, so it always reads as 0. 0b - No impact. 1b - Enable clear operation.
1 —	Reserved
0 APAREN	AHB Parallel Mode Enable Enables Parallel mode for AHB-triggered read and write commands. 0b - Flash is accessed in Individual mode. 1b - Flash is accessed in Parallel mode.

10.2.6.2.6 Interrupt Enable (INTEN)

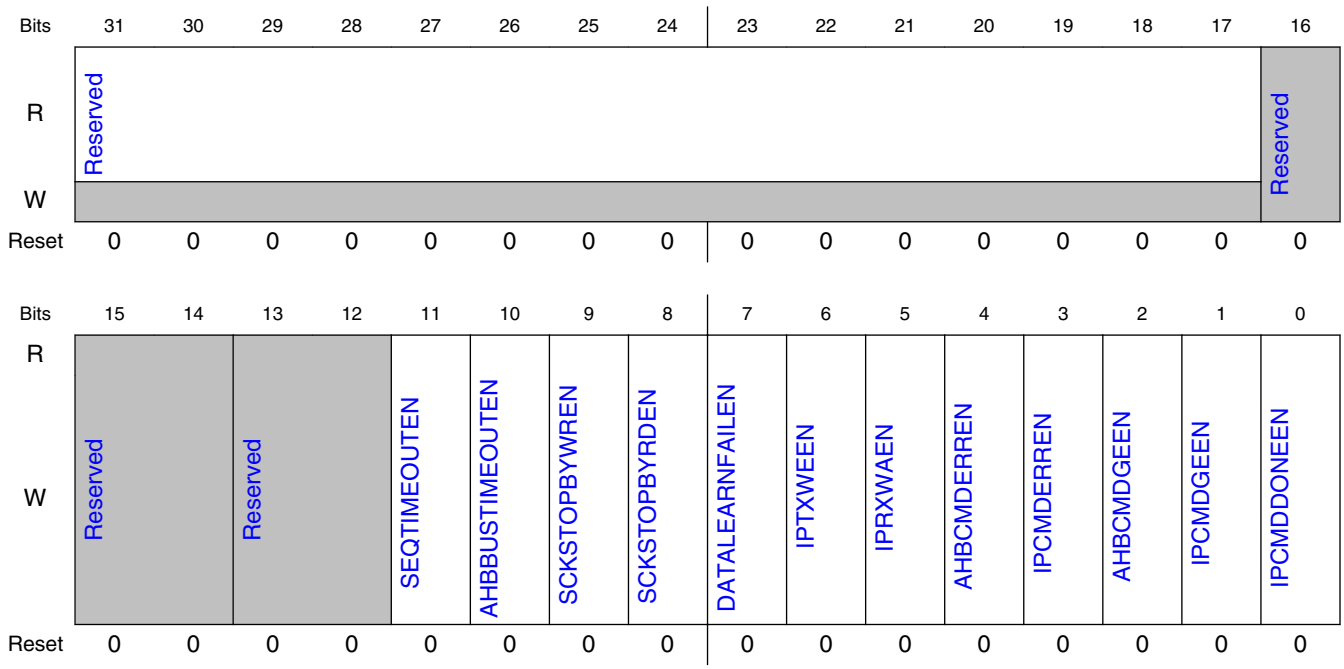
10.2.6.2.6.1 Offset

Register	Offset
INTEN	10h

10.2.6.2.6.2 Function

Includes AHB error response, IPS error response, and ECC error interrupt enables. See Interrupts.

10.2.6.2.6.3 Diagram



10.2.6.2.6.4 Fields

Field	Function
31-17 —	Reserved
16 —	Reserved
15-14 —	Reserved
13-12 —	Reserved
11 SEQTIMEOUTEN	Sequence execution Timeout Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
10 AHBBUSTIMEOUTEN	AHB Bus Timeout Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
9 SCKSTOPBYWREN	SCLK Stopped By Write Interrupt Enable Enables interrupt that indicates SCLK is stopped during command sequence because asynchronous transmit FIFO is empty. 0b - Disable interrupt or no impact 1b - Enable interrupt
8	SCLK Stopped By Read Interrupt Enable

Table continues on the next page...

Field	Function
SCKSTOPBYRDEN	Enables interrupt that indicates SCLK is stopped during command sequence because asynchronous receive FIFO is full. 0b - Disable interrupt or no impact 1b - Enable interrupt
7 DATALEARNFAILEN	Data Learning Failed Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
6 IPTXWEEN	IP Transmit FIFO Watermark Empty Interrupt Enable Enables interrupt that indicates IP transmit FIFO contains more empty space than watermark level. 0b - Disable interrupt or no impact 1b - Enable interrupt
5 IPRXWAEN	IP Receive FIFO Watermark Available Interrupt Enable Enables interrupt that indicates IP receive FIFO contains more valid data than the watermark level. 0b - Disable interrupt or no impact 1b - Enable interrupt
4 AHBCMDERREN	AHB-Triggered Command Sequences Error Detected Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
3 IPCMDERREN	IP-Triggered Command Sequences Error Detected Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
2 AHBCMDGEEN	AHB-Triggered Command Sequences Grant Timeout Interrupt Enable. 0b - Disable interrupt or no impact 1b - Enable interrupt
1 IPCMDGEEN	IP-Triggered Command Sequences Grant Timeout Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
0 IPCMDDONEEN	IP-Triggered Command Sequences Execution Finished Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt

10.2.6.2.7 Interrupt (INTR)

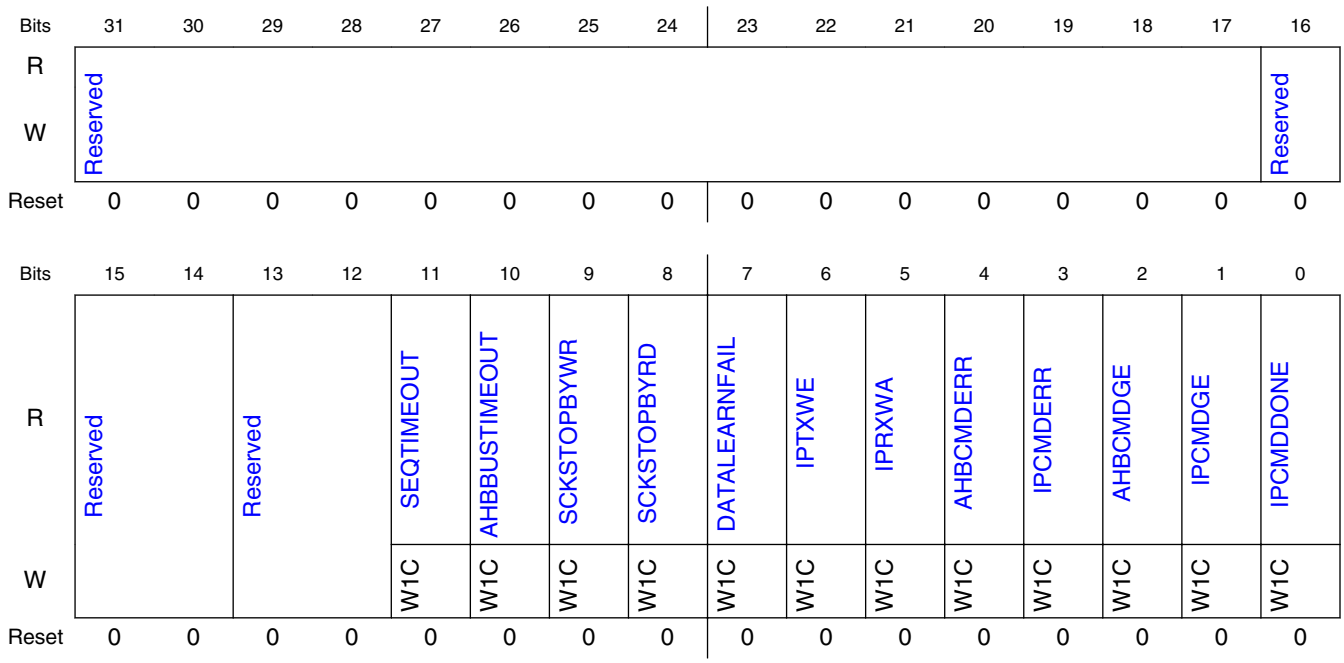
10.2.6.2.7.1 Offset

Register	Offset
INTR	14h

10.2.6.2.7.2 Function

Includes AHB error response, IPS error response, and ECC error interrupts. See Interrupts.

10.2.6.2.7.3 Diagram



10.2.6.2.7.4 Fields

Field	Function
31-17 —	Reserved
16 —	Reserved
15-14 —	Reserved
13-12 —	Reserved
11 SEQTIMEOUT	Sequence Execution Timeout **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
10 AHBBUSTIMEOUT	AHB Bus Timeout **NOTE:** This field behaves differently for read and write operations.

Table continues on the next page...

Field	Function
	When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
9 SCKSTOPBYWR	SCLK Stopped Due To Empty Transmit FIFO Generated when SCLK is stopped during command sequence because the asynchronous transmit FIFO is empty. **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
8 SCKSTOPBYRD	SCLK Stopped Due To Full Receive FIFO Generated when SCLK is stopped during command sequence because the asynchronous receive FIFO is full. **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
7 DATALEARNFAIL	Data Learning Failed **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
6 IPTXWE	IP Transmit FIFO Watermark Empty Generated when the IP transmit FIFO contains more empty space than the watermark level. **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred

Table continues on the next page...

Field	Function
	When writing 0b - No effect 1b - Clear the flag
5 IPRXWA	IP Receive FIFO Watermark Available Generated when the IP receive FIFO contains more valid data than the watermark level. **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
4 AHBCMDERR	AHB-Triggered Command Sequences Error Generated upon an AHB-triggered command sequence error. When an error is detected for an AHB command, the command is ignored and not executed. **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
3 IPCMDERR	IP-Triggered Command Sequences Error Generated upon an IP-triggered command sequence error. When an error is detected for an IP command, the command is ignored and not executed. **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
2 AHBCMDGE	AHB-Triggered Command Sequences Grant Timeout **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
1	IP-Triggered Command Sequences Grant Timeout

Table continues on the next page...

Field	Function
IPCMDGE	**NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
0 IPCMDDONE	IP-Triggered Command Sequences Execution Finished Generated upon completion of an IP-triggered command sequence. Also generated when IPCMDGE or IPCMDERR interrupt is generated **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag

10.2.6.2.8 LUT Key (LUTKEY)

10.2.6.2.8.1 Offset

Register	Offset
LUTKEY	18h

10.2.6.2.8.2 Function

Contains the key to lock and unlock LUT. See Lookup table (LUT).

10.2.6.2.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	KEY															
Reset	0	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KEY															
Reset	0	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0

10.2.6.2.8.4 Fields

Field	Function
31-0 KEY	LUT Key Contains key to lock or unlock LUT. The key is 5AF05AF0h. Read value is always 5AF05AF0h.

10.2.6.2.9 LUT Control (LUTCR)

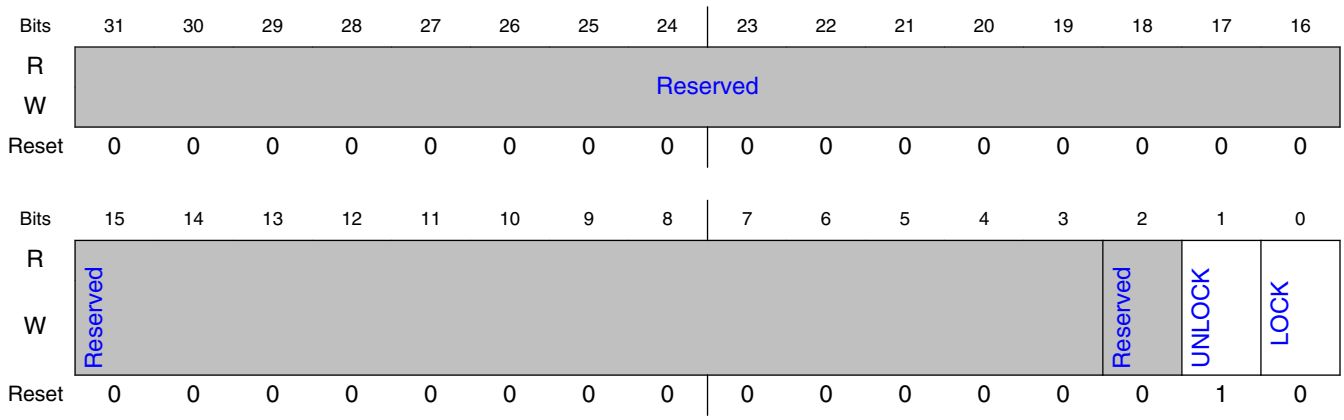
10.2.6.2.9.1 Offset

Register	Offset
LUTCR	1Ch

10.2.6.2.9.2 Function

Used with LUTKEY register to lock or unlock LUT. For the lock or unlock operation to be successful, this register must be written immediately after writing 5AF05AF0h to the LUTKEY register. See Lookup table (LUT) for details on locking and unlocking LUT. You cannot write 00 or 11 to the LOCK and UNLOCK fields.

10.2.6.2.9.3 Diagram



10.2.6.2.9.4 Fields

Field	Function
31-3 —	Reserved
2 —	Reserved
1 UNLOCK	Unlock LUT 0b - LUT is locked (LUTCR[LOCK] must be 1) 1b - LUT is unlocked and can be written
0 LOCK	Lock LUT 0b - LUT is unlocked (LUTCR[UNLOCK] must be 1) 1b - LUT is locked and cannot be written

10.2.6.2.10 AHB Receive Buffer n Control 0 (AHBRXBUF0CR0 - AHBRXBUF7CR0)

10.2.6.2.10.1 Offset

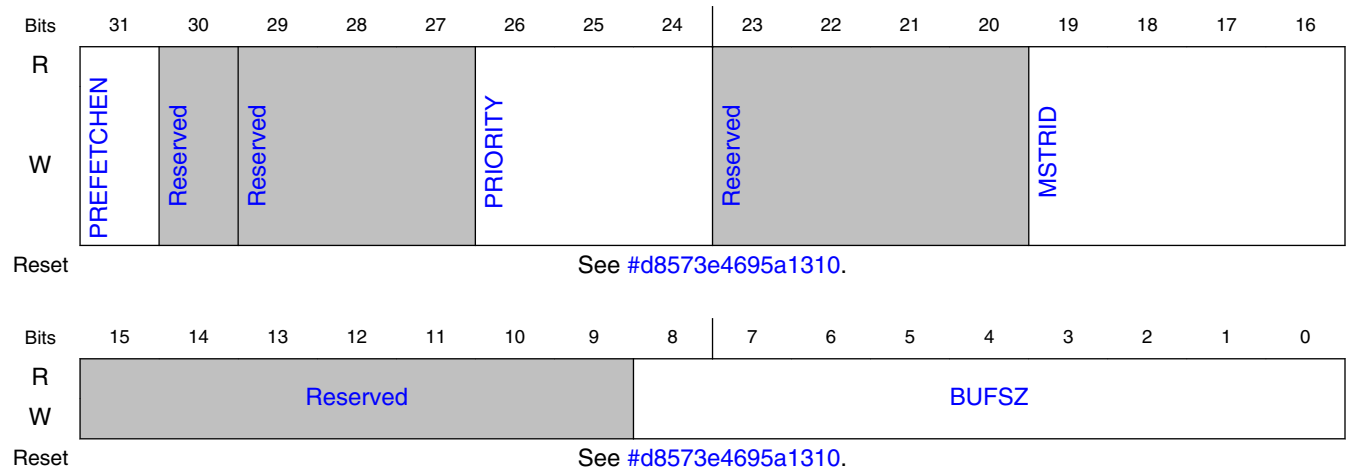
For n = 0 to 7:

Register	Offset
AHBRXBUFnCR0	20h + (n × 4h)

10.2.6.2.10.2 Function

Stores the read data from the SPI interface.

10.2.6.2.10.3 Diagram



10.2.6.2.10.4 Register reset values

Register	Reset value
AHBRXBUF0CR0	8000_0020h
AHBRXBUF1CR0	8001_0020h
AHBRXBUF2CR0	8002_0020h
AHBRXBUF3CR0	8003_0020h
AHBRXBUF4CR0	8004_0020h
AHBRXBUF5CR0	8005_0020h
AHBRXBUF6CR0	8006_0020h
AHBRXBUF7CR0	8007_0020h

10.2.6.2.10.5 Fields

Field	Function
31 PREFETCHEN	AHB Read Prefetch Enable Enables AHB read prefetch for the controller corresponding to the current AHB receive buffer. The prefetch feature is disabled when AHBCR[PREFETCHEN] is 0. You can use this field to enable or disable prefetch separately for each controller. 0b - Disabled 1b - Enabled when AHBCR[PREFETCHEN] is enabled.
30 —	Reserved
29-27	Reserved

Table continues on the next page...

Field	Function
—	
26-24 PRIORITY	AHB Controller Read Priority Configure the priority for the AHB Controller Read to which this AHB receive buffer is assigned. 7 is the highest priority, 0 the lowest. See Command abort and suspend.
23-20 —	Reserved
19-16 MSTRID	AHB Controller ID Configures the ID of the AHB controller to which this AHB receive buffer is assigned. See AHB receive buffer management.
15-9 —	Reserved
8-0 BUFSZ	AHB Receive Buffer Size Configures the size of the AHB receive buffer in multiples of 64 bits. See AHB receive buffer management.

10.2.6.2.11 Flash Control 0 (FLSHA1CR0 - FLSHB2CR0)

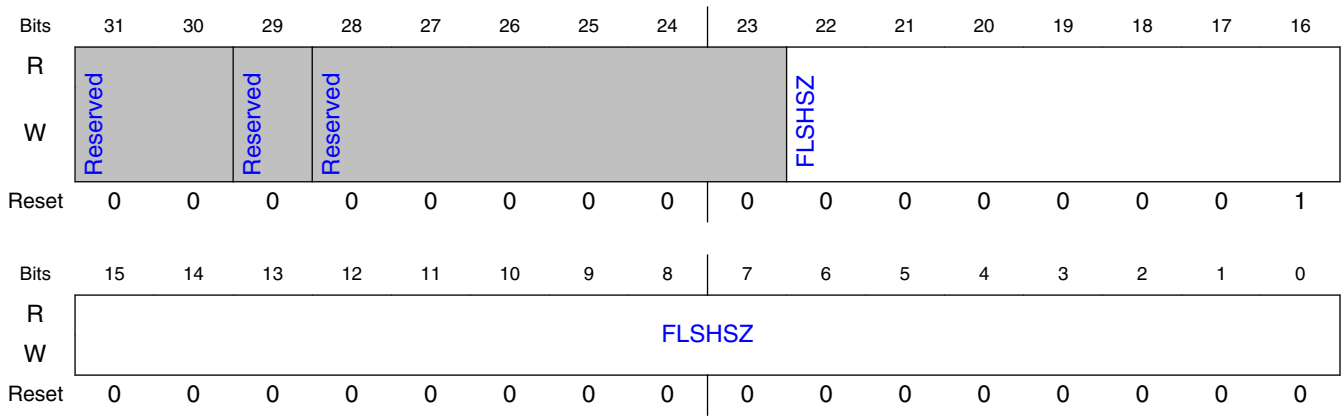
10.2.6.2.11.1 Offset

Register	Offset
FLSHA1CR0	60h
FLSHA2CR0	64h
FLSHB1CR0	68h
FLSHB2CR0	6Ch

10.2.6.2.11.2 Function

Contains flash memory size setting. FlexSPI determines which device is accessed (Chip Select) via this register.

10.2.6.2.11.3 Diagram



10.2.6.2.11.4 Fields

Field	Function
31-30 —	Reserved
29 —	Reserved
28-23 —	Reserved
22-0 FLSHSZ	Flash Size in KB Configures the maximum flash memory size. For AHB access: Maximum flash size supported for each device is 256 MB. When the value of this field is greater than the maximum size, the device flash size is interpreted as the maximum. The maximum flash size is the largest flash size supported for a single device. It is also the maximum total flash size supported for all devices (up to four). If the total flash size is larger than the maximum size, only the maximum flash size address space is accessible. For IPS access, The maximum flash size supported for each device is 4 GB. When the value of this field is greater than 400000h, the device flash size is taken as 4 GB. The max total flash size supported (for all 4 devices) is also 4 GB. If the total flash size is larger than 4 GB, only 4 GB of address space is accessible.

10.2.6.2.12 Flash Control 1 (FLSHA1CR1 - FLSHB2CR1)

10.2.6.2.12.1 Offset

Register	Offset
FLSHA1CR1	70h
FLSHA2CR1	74h

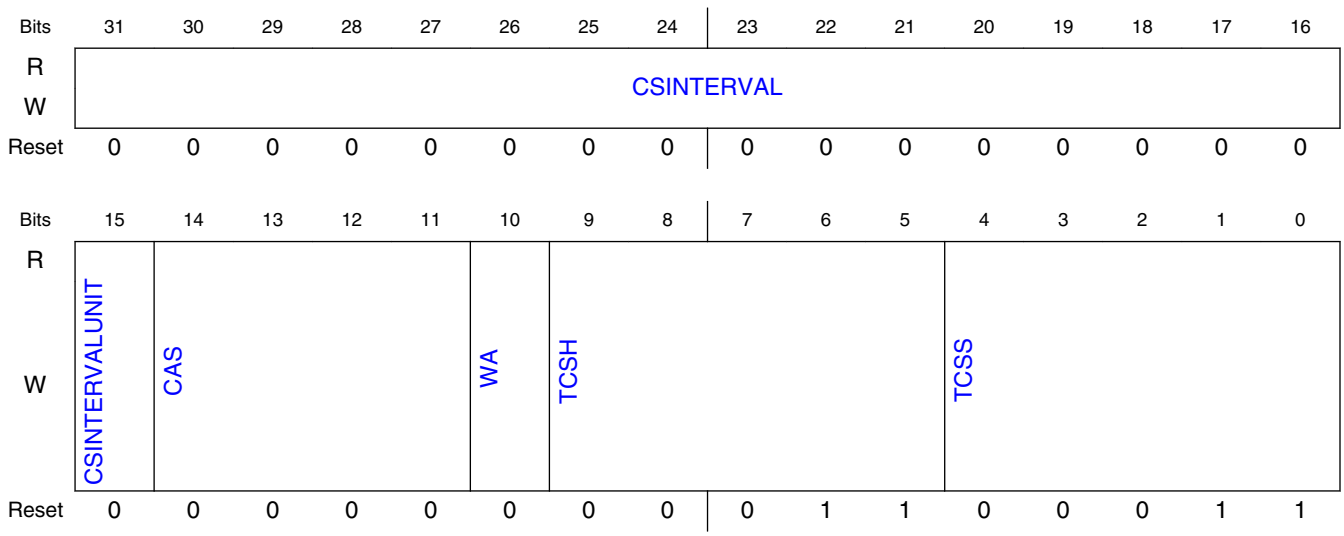
Table continues on the next page...

Register	Offset
FLSHB1CR1	78h
FLSHB2CR1	7Ch

10.2.6.2.12.2 Function

Contains settings for flash device-specific timings and flash internal address space.

10.2.6.2.12.3 Diagram



10.2.6.2.12.4 Fields

Field	Function
31-16 CSINTERVAL	Chip Select Interval Configures the minimum interval between flash device chip select deassertion and chip select assertion. If the external flash device has a limitation on the interval between command sequences, configure this field accordingly. If there is no limitation, write 0h to this field. When CSINTERVALUNIT = 0, the chip select invalid interval is: CSINTERVAL × 1 serial clock cycle; When CSINTERVALUNIT = 1, the chip select invalid interval is: CSINTERVAL × 256 serial clock cycles. **NOTE:** The minimum chip select interval is 2 cycles, even when the value of CSINTERVAL is less than 2.
15 CSINTERVALUNIT	Chip Select Interval Unit Configures the interval unit for chip select. 0b - 1 serial clock cycle 1b - 256 serial clock cycles
14-11 CAS	Column Address Size

Table continues on the next page...

Field	Function
	When external flash memory has a separate address field for rows and columns, this field configures the flash column address bit width. FlexSPI automatically splits a flash-mapped address into row address and column address according to the values of this field and the WA field. When the external flash memory does not support column address, write 0 to this field. FlexSPI transmits all flash address bits as Row address. For flash address mapping, see Flash address sent to flash memory devices.
10 WA	Word-Addressable Configures whether external flash memory is word-addressable or byte-addressable. If flash memory is word-addressable, it should be accessed in multiples of 16 bits. Currently, FlexSPI does not transmit flash address bit 0 to external flash memory. For flash address mapping, see Flash address sent to flash memory devices. 0b - Byte-addressable 1b - Word-addressable
9-5 TCSH	Serial Flash CS Hold Time Used to meet flash TCSH timing requirement. Serial flash CS Hold time that FlexSPI promises is: (TCSH + 1/2) serial clock cycles for DDR mode, and TCSH serial clock cycles for SDR mode. See Output timing between chip select and SCLK.
4-0 TCSS	Serial Flash CS Setup Time Used to meet flash TCSS timing requirement. Serial flash CS Setup time that FlexSPI promises is: (TCSS + 1/2) serial root clock cycles for SDR and DDR mode. See Output timing between chip select and SCLK.

10.2.6.2.13 Flash Control 2 (FLSHA1CR2 - FLSHB2CR2)

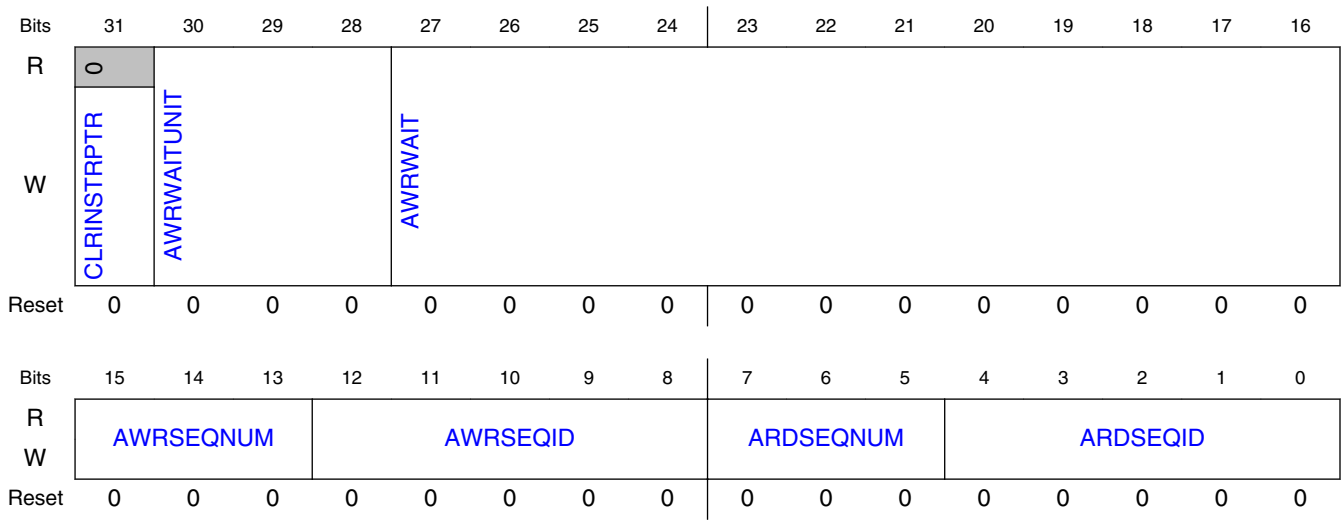
10.2.6.2.13.1 Offset

Register	Offset
FLSHA1CR2	80h
FLSHA2CR2	84h
FLSHB1CR2	88h
FLSHB2CR2	8Ch

10.2.6.2.13.2 Function

Contains fields to configure AHB bus access. If the four external devices are different types, AHB read and write commands may use different command sequences; AHB bus ready wait time may also differ.

10.2.6.2.13.3 Diagram



10.2.6.2.13.4 Fields

Field	Function
31 CLRINSTRPTR	Clear Instruction Pointer Clears the instruction pointer, which is the pointer that JMP_ON_CS saves internally. See Programmable sequence engine. This field is used for AHB read access to external flash memory supporting Execute-In-Place (XIP) mode.
30-28 AWRWAITUNIT	AWRWAIT Unit Configures the unit of AHB write wait time, as the value of AWRWAIT determines, in terms of AHB clock cycles. 000b - 2 001b - 8 010b - 32 011b - 128 100b - 512 101b - 2048 110b - 8192 111b - 32768
27-16 AWRWAIT	AHB Write Wait Configures AHB write wait time, with the AWRWAITUNIT field. Certain devices (such as FPGA) require time to write data into internal memory after the command sequences finished on the FlexSPI interface. If another read command sequence arrives before the current programming finishes internally, the read data may be wrong. This field is used to hold the AHB bus ready for AHB write access, waiting until the programming is finished in the external device. This hold ensures that no AHB read command is triggered before the programming finishes in the external device. The wait cycle between an AHB-triggered command sequence finishing on FlexSPI and the AHB return bus being ready is: AWRWAIT × AWRWAITUNIT.
15-13 AWRSEQNUM	Sequence Number for AHB Write-Triggered Command

Table continues on the next page...

Field	Function
	Configures the sequence number of an AHB read-triggered command. For certain flash devices (for example, HyperFlash, HyperRam, and Serial NAND flash), a flash programming access is done via several command sequences. An AHB write command triggers (AWRSEQNUM + 1) command sequences to external flash memory each time. FlexSPI executes the sequences in LUT incrementally. **NOTE:** • Software should ensure that the last sequence index never exceeds LUT sequence numbers: AWRSEQID + AWRSEQNUM < 32 • Software must ensure that the AWRSEQNUM and LUT fields are configured correctly according to the external device specification. FlexSPI does not check the sequence; it executes the sequences one by one.
12-8 AWRSEQID	Sequence Index for AHB Write-Triggered Command
7-5 ARDSEQNUM	Sequence Number for AHB Read-Triggered Command Configures the sequence number of an AHB read-triggered command in the lookup table. For certain flash devices (for example, HyperFlash, HyperRam, and Serial NAND flash), a flash read access is done via several command sequences. An AHB read command triggers (ARDSEQNUM + 1) command sequences to external flash memory each time. FlexSPI executes the sequences in LUT incrementally. **NOTE:** • Software should ensure that the last sequence index never exceeds LUT sequence numbers: ARDSEQID + ARDSEQNUM ≤ 32. • Software must ensure that the ARDSEQNUM and LUT fields are configured correctly according to the external device specification. FlexSPI does not check the sequence; it executes the sequences one by one.
4-0 ARDSEQID	Sequence Index for AHB Read-Triggered Command in LUT

10.2.6.2.14 Flash Control 4 (FLSHCR4)

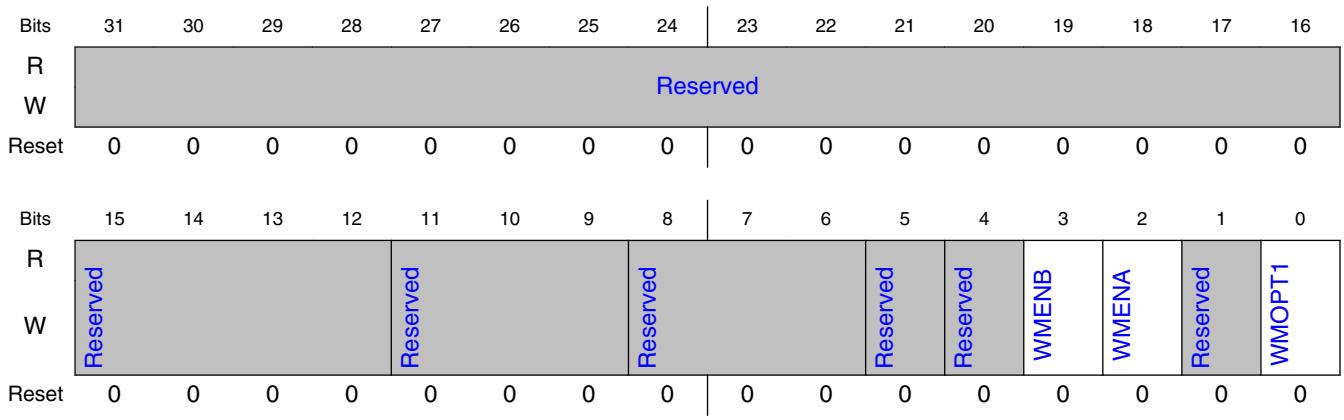
10.2.6.2.14.1 Offset

Register	Offset
FLSHCR4	94h

10.2.6.2.14.2 Function

Provides configuration for all external devices.

10.2.6.2.14.3 Diagram



10.2.6.2.14.4 Fields

Field	Function
31-12 —	Reserved
11-9 —	Reserved
8-6 —	Reserved
5 —	Reserved
4 —	Reserved
3 WMENB	Write Mask Enable for Port B Enables write mask for flash device on port B. 0b - Disabled. When writing to external device, DQS(RWDS) pin is not driven. 1b - Enabled. When writing to external device, FlexSPI drives DQS(RWDS) pin as write mask output.
2 WMENA	Write Mask Enable for Port A Enables write mask for flash device on port A. 0b - Disabled. When writing to external device, DQS(RWDS) pin is not driven. 1b - Enabled. When writing to external device, FlexSPI drives DQS(RWDS) pin as write mask output.
1 —	Reserved
0 WMOPT1	Write Mask Option 1 Used to remove AHB and IP write burst start address alignment limitation. 0b - When writing to an external device, DQS pin is used as write mask. When flash memory is accessed in individual mode, AHB or IP write burst start address alignment is not limited. 1b - When writing to an external device, DQS pin is not used as write mask. When flash memory is accessed in individual mode, AHB or IP write burst start address alignment is limited.

10.2.6.2.15 IP Control 0 (IPCR0)

10.2.6.2.15.1 Offset

Register	Offset
IPCR0	A0h

10.2.6.2.15.2 Function

Provides all configuration required for IP commands. Provides the start address for the flash device, instead of the chip address, to be accessed for IP command. FlexSPI determines the chip select automatically according to this start address.

NOTE

- You cannot issue an IP command that crosses flash device boundaries. If you do so, it generates an IPCMDERR interrupt.
- Configure this register before an IP command is triggered.
- Do not change the values in this register while an IP command is in progress.

10.2.6.2.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SFAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SFAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.2.6.2.15.4 Fields

Field	Function
31-0 SFAR	Serial Flash Address Configures the serial flash address for IP commands. The address should be the address of the flash device without the base address.

10.2.6.2.16 IP Control 1 (IPCR1)

10.2.6.2.16.1 Offset

Register	Offset
IPCR1	A4h

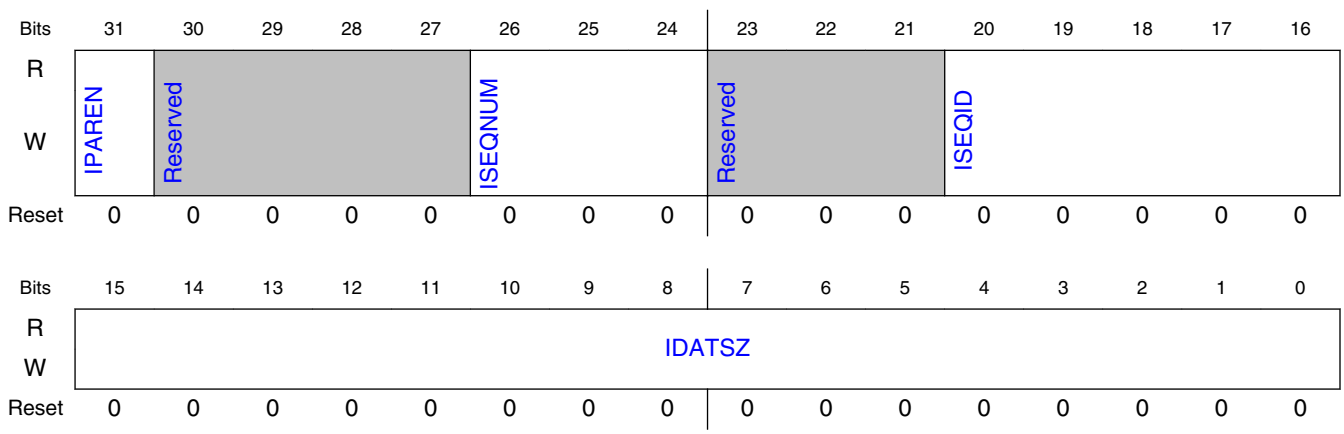
10.2.6.2.16.2 Function

Provides all configuration required for IP commands. Provides the flash read and program data size, sequence index in LUT, sequence number and individual/parallel mode settings for IP commands.

NOTE

- Configure this register before an IP command is triggered.
- Do not change the values in this register while an IP command is in progress.

10.2.6.2.16.3 Diagram



10.2.6.2.16.4 Fields

Field	Function
31 IPAREN	Parallel Mode Enable for IP Commands 0b - Disabled. Flash memory is accessed in Individual mode. 1b - Enabled. Flash memory is accessed in Parallel mode.
30-27	Reserved

Table continues on the next page...

Field	Function
—	
26-24 ISEQNUM	Sequence Number for IP command: ISEQNUM+1.
23-21 —	Reserved
20-16 ISEQID	Sequence Index in LUT for IP command.
15-0 IDATSZ	Flash Read/Program Data Size (in bytes) for IP command.

10.2.6.2.17 IP Command (IPCMD)

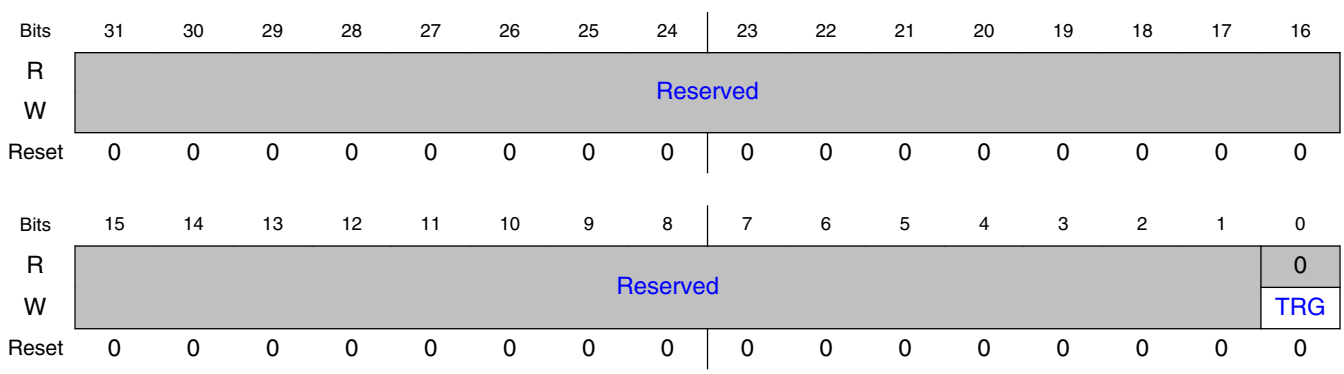
10.2.6.2.17.1 Offset

Register	Offset
IPCMD	B0h

10.2.6.2.17.2 Function

Used to trigger an IP command to access external flash device. When the arbitrator grants the IP command, the command is executed on the FlexSPI interface.

10.2.6.2.17.3 Diagram



10.2.6.2.17.4 Fields

Field	Function
31-1 —	Reserved
0 TRG	Command Trigger Triggers an IP command. This field clears automatically after it has been written and always reads as 0. **NOTE:** You cannot trigger another IP command before the previous IP command finishes on the FlexSPI interface. Software must poll INTR[IPCMDDONE] or wait for this interrupt. 0b - No action 1b - Start the IP command that the IPCR0 and IPCR1 registers define.

10.2.6.2.18 Data Learning Pattern (DLPR)

10.2.6.2.18.1 Offset

Register	Offset
DLPR	B4h

10.2.6.2.18.2 Function

Provides the pattern to be used during Data Learning in FlexSPI.

10.2.6.2.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.2.6.2.18.4 Fields

Field	Function
31-0	Data Learning Pattern

Field	Function
DLP	The operand in the LEARN_SDR or LEARN_DDR instruction code determines the data learning pattern bit number. This number never exceeds 32 bits. If the operand in the instruction code is greater than 32, a 32-bit pattern is used. See Data learning.

10.2.6.2.19 IP Receive FIFO Control (IPRXFCR)

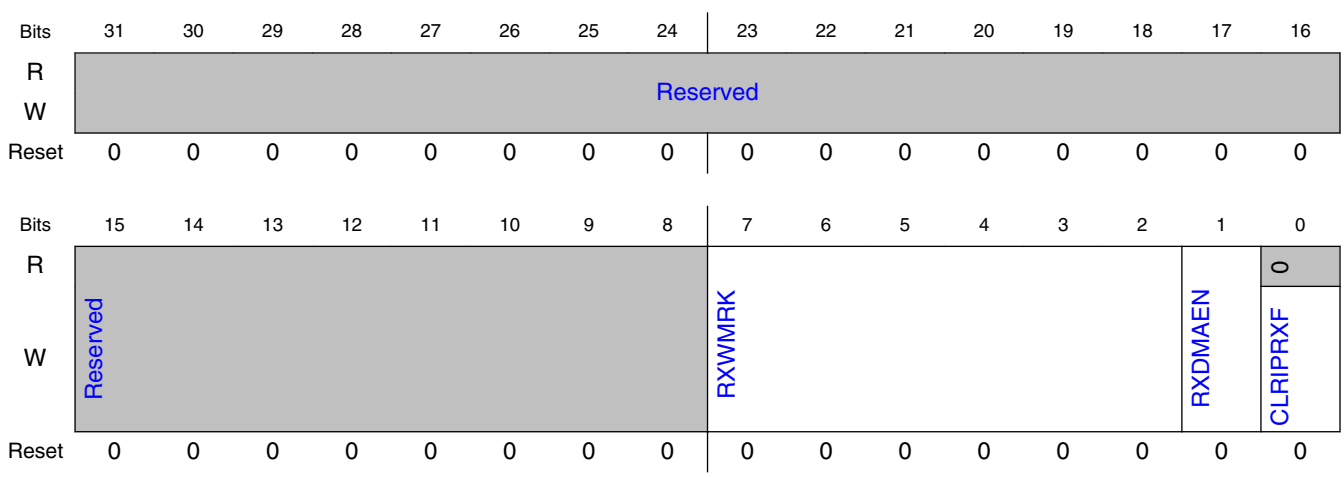
10.2.6.2.19.1 Offset

Register	Offset
IPRXFCR	B8h

10.2.6.2.19.2 Function

Provides the configuration fields for IP receive FIFO management.

10.2.6.2.19.3 Diagram



10.2.6.2.19.4 Fields

Field	Function
31-8 —	Reserved
7-2 RXWMRK	IP Receive FIFO Watermark Level Configures the watermark level for the IP receive FIFO. The watermark level is (RXWMRK + 1) × 64 bits.

Table continues on the next page...

Field	Function
	The INTR[IPRXWA] interrupt is set when FlexSPI fills the IP receive FIFO ≥ the watermark level. A DMA request occurs when the fill level is ≥ the watermark level and IPRXFCR[RXDMAEN] = 1. When the fill level is ≥ the watermark level and INTEN[IPRXWAEN] = 1, an INTR[IPRXWA] interrupt is generated. **NOTE:** After writing 1 to INTR[IPRXWA] to clear it, the read address should be rolled back to the start address (memory mapped). If the IP bus reads the IP receive FIFO, the read address should roll back to RFDR0. If the AHB bus reads the IP receive FIFO, the read address should roll back to ARDF_BASE.
1 RXDMAEN	IP Receive FIFO Reading by DMA Enable 0b - Disabled. The processor reads the FIFO. 1b - Enabled. DMA reads the FIFO.
0 CLRIPRXF	Clear IP Receive FIFO Clears all valid data entries in IP receive FIFO. Resets the read and write pointers in IP receive FIFO. 0b - No function 1b - A clock cycle pulse clears all valid data entries in IP receive FIFO.

10.2.6.2.20 IP Transmit FIFO Control (IPTXFCR)

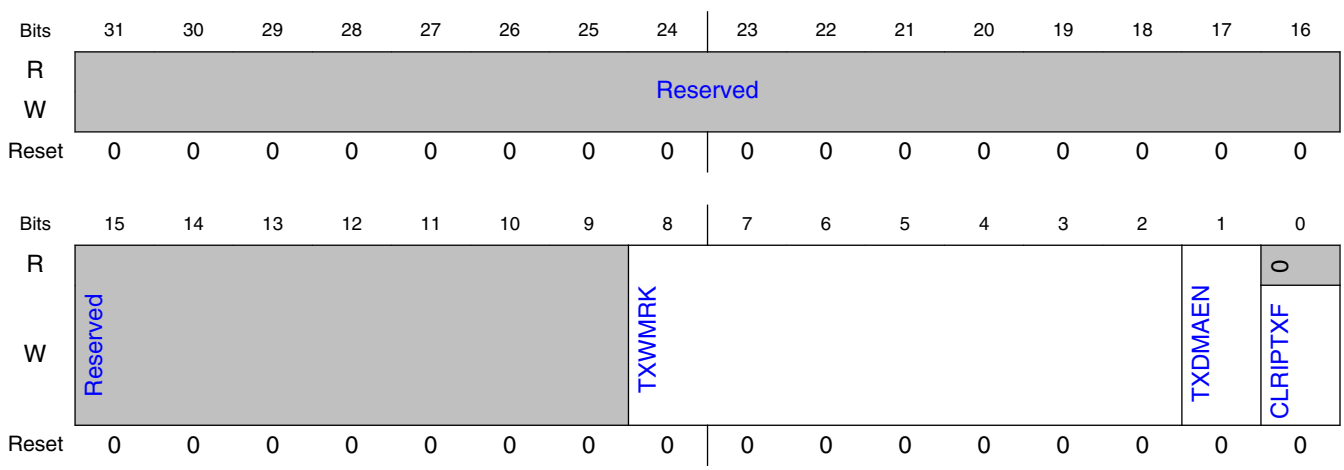
10.2.6.2.20.1 Offset

Register	Offset
IPTXFCR	BCh

10.2.6.2.20.2 Function

Provides the configuration fields for IP transmit FIFO management.

10.2.6.2.20.3 Diagram



10.2.6.2.20.4 Fields

Field	Function
31-9 —	Reserved
8-2 TXWMRK	Transmit Watermark Level Sets the transmit watermark level. The watermark level is (TXWMRK + 1) × 64 bits. INTR[IPTXWE] is set when FlexSPI empties the IP transmit FIFO ≥ the watermark level. When the empty level ≥ the watermark level and IPTXFCR[TXDMAEN] = 1, a DMA request occurs. When the empty level ≥ the watermark level and INTEN[IPTXWEEN] = 1, an INTR[IPTXWE] interrupt is generated. **NOTE:** • The watermark level should not be larger than the write window. • The watermark level should not be larger than the IP transmit FIFO size. • The write address to the IP receive FIFO should roll back to the start address of the write window. After pushing the IP transmit fifo, you should write 1 to INTR[IPTXWE] to clear it, which will perform the rollback.
1 TXDMAEN	Transmit FIFO DMA Enable Selects whether DMA or processor fills IP transmit FIFO. 0b - Processor 1b - DMA
0 CLRIPTXF	Clear IP Transmit FIFO Clears all valid data entries in IP transmit FIFO. The read and write pointers in IP transmit FIFO are reset. 0b - No function 1b - A clock cycle pulse clears all valid data entries in the IP transmit FIFO.

10.2.6.2.21 DLL Control 0 (DLLACR - DLLBCR)

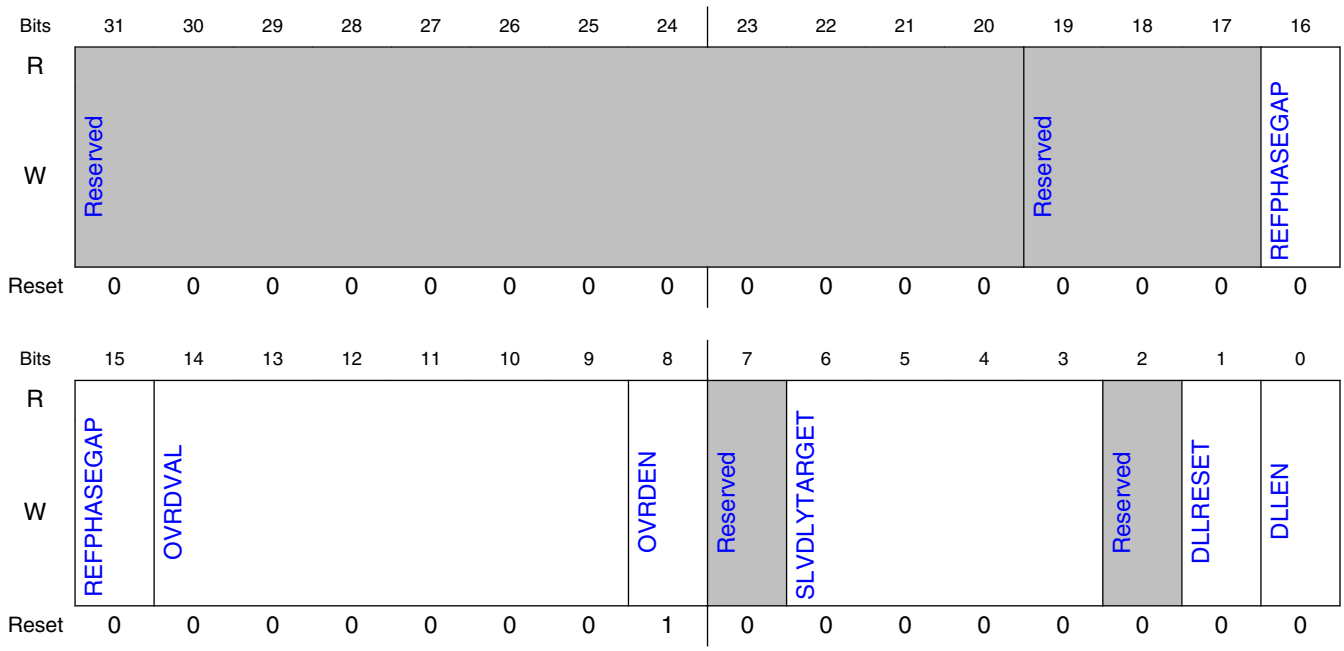
10.2.6.2.21.1 Offset

Register	Offset
DLLACR	C0h
DLLBCR	C4h

10.2.6.2.21.2 Function

Configures Flash A and B sample clock DLL.

10.2.6.2.21.3 Diagram



10.2.6.2.21.4 Fields

Field	Function
31-20 —	Reserved
19-17 —	Reserved
16-15 REFPHASEGAP	Reference Clock Delay Line Phase Adjust Gap. REFPHASEGAP setting of 2h is recommended if DLLEN is set.
14-9 OVRDVAL	Target Clock Delay Line Override Value Configures the override value for the target clock line delay cell number. When OVRDEN = 1, the delay cell number in DLL is OVRDVAL + 1. See DLL configuration for sampling.
8 OVRDEN	Target Clock Delay Line Override Value Enable Enables the override value for the target clock line delay cell number. 0b - Disable 1b - Enable
7 —	Reserved
6-3 SLVDLYTARGET	Target Delay Line Configures the target delay line for the target. The delay target for target delay line is: ((SLVDLYTARGET +1) × 1/32 × clock cycle of reference clock (serial root clock). If serial root clock is ≥ 100 MHz, DLLEN is set to 1, OVRDEN is set to 0, then SLVDLYTARGET setting is up to flash parameters used.
2	Reserved

Table continues on the next page...

Field	Function
—	
1 DLLRESET	DLL reset Forces a DLL reset. The forced reset causes the DLL to lose lock and recalibrate to detect a ref_clock half-period phase shift. The reset action is edge-triggered, so software must write 0 to this field after writing 1 to it (no delay limitation). 0b - No function 1b - Force DLL reset.
0 DLLEN	DLL Calibration Enable Enables DLL calibration. When DLL calibration is disabled, the delay cell number in the target delay line is always 1. When DLLxCR[OVRDEN] = 1, the target delay line is overridden and this field is ignored. 0b - Disable 1b - Enable

10.2.6.2.22 Status 0 (STS0)

10.2.6.2.22.1 Offset

Register	Offset
STS0	E0h

10.2.6.2.22.2 Function

Indicates the status of the internal state machine.

10.2.6.2.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				DATALEARNPHASEB				DATALEARNPHASEA				ARBCMDSRC		ARBIDLE	SEQIDLE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

10.2.6.2.22.4 Fields

Field	Function
31-12 —	Reserved
11-8 DATALEARNPHASEB	Data Learning Phase Selection on Port B Indicates the sampling clock phase selection on Port B after data learning. This field is similar to STS0[DATALEARNPHASEA]. The sampling clock phase is chosen separately for Port A and Port B.
7-4 DATALEARNPHASEA	Data Learning Phase Selection on Port A Indicates the sampling clock phase selection on Port A after data learning. There are 16 clock phases for the sampling clock that the delay cell line generates. When data learning is not enabled, the default clock phase 0 is used to sample flash read data. When data learning is enabled and a LEARN_SDR or LEARN_DDR instruction has executed correctly, FlexSPI determines the correct clock phase to sample read data. See Data learning.
3-2 ARBCMDSRC	ARB Command Source Indicates the trigger source of current command sequence that the arbitrator has granted. When ARB_CTL is not busy (STS0[ARBIDLE] = 1), the value of this field does not matter. 00b - Trigger source is AHB read command. 01b - Trigger source is AHB write command. 10b - Trigger source is IP command (by writing 1 to IPCMD[TRG]). 11b - Trigger source is a suspended command that has resumed.
1 ARBIDLE	ARB_CTL State Machine Idle Indicates whether the state machine in ARB_CTL is idle, or a command sequence that the arbitrator granted has not finished on the FlexSPI interface. When idle, no transaction is occurring on the FlexSPI interface (STS0[SEQIDLE] = 1). When waiting for the FlexSPI controller to become idle, poll this field instead of STS0[SEQIDLE]. 0b - Not idle 1b - Idle
0 SEQIDLE	SEQ_CTL State Machine Idle Indicates whether the state machine in SEQ_CTL is idle, or a command sequence is executing on the FlexSPI interface. 0b - Not idle 1b - Idle

10.2.6.2.23 Status 1 (STS1)

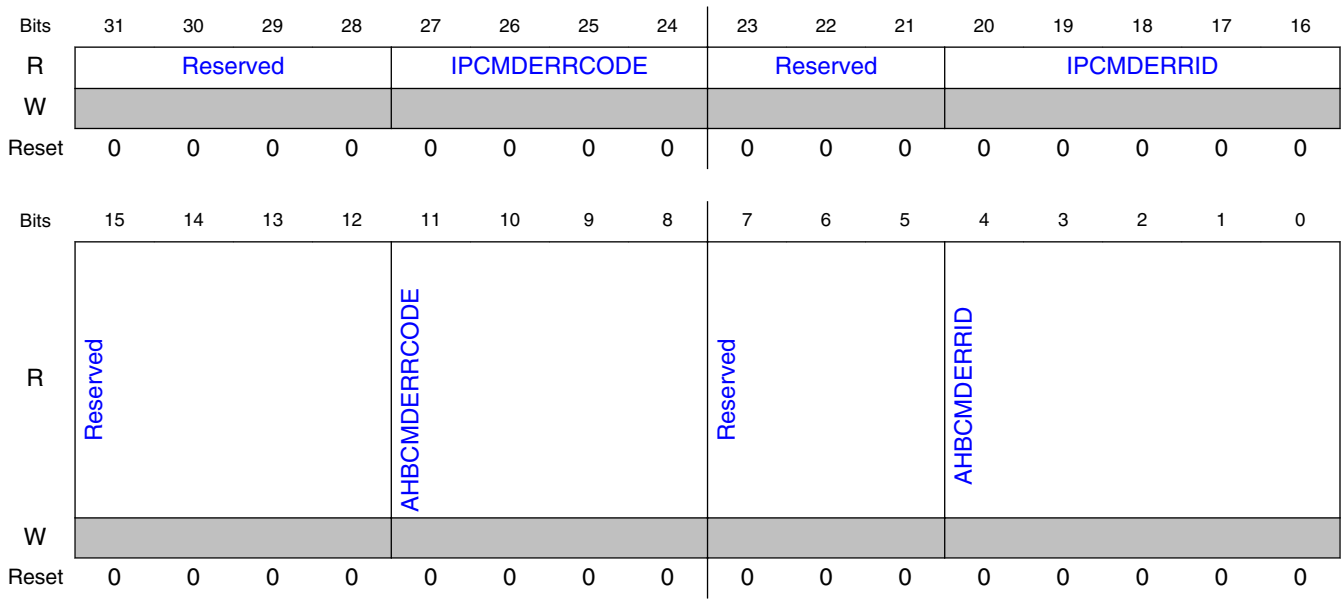
10.2.6.2.23.1 Offset

Register	Offset
STS1	E4h

10.2.6.2.23.2 Function

Indicates the error code of the AHB or IPS interface error response.

10.2.6.2.23.3 Diagram



10.2.6.2.23.4 Fields

Field	Function
31-28 —	Reserved
27-24 IPCMDERRCODE	IP Command Error Code When an IP command Error is detected, indicates the error code. This field returns to zero when INTR[IPCMDERR] is cleared. 0000b - No error 0010b - IP command with JMP_ON_CS instruction used in the sequence 0011b - Unknown instruction opcode in the sequence 0100b - DUMMY_SDR or DUMMY_RWDS_SDR instruction used in DDR sequence 0101b - DUMMY_DDR or DUMMY_RWDS_DDR instruction used in SDR sequence 0110b - Flash memory access start address exceeds entire flash address range (A1, A2, B1, and B2) 1110b - Sequence execution timeout 1111b - Flash boundary crossed
23-21 —	Reserved
20-16 IPCMDERRID	IP Command Error ID When an IP command error is detected, indicates the sequence index. This field returns to zero when INTR[IPCMDERR] is cleared.
15-12	Reserved

Table continues on the next page...

Field	Function
—	
11-8 AHBCMDERRCODE	AHB Command Error Code Contains the error code when an AHB command error is detected. This field returns to zero when INTR[AHBCMDERR] is cleared. 0000b - No error 0010b - AHB Write command with JMP_ON_CS instruction used in the sequence 0011b - Unknown instruction opcode in the sequence 0100b - DUMMY_SDR or DUMMY_RWDS_SDR instruction used in DDR sequence 0101b - DUMMY_DDR or DUMMY_RWDS_DDR instruction used in SDR sequence 1110b - Sequence execution timeout
7-5 —	Reserved
4-0 AHBCMDERRID	AHB Command Error ID When an AHB command error is detected, indicates the sequence index. This field returns to zero when INTR[AHBCMDERR] is cleared.

10.2.6.2.24 Status 2 (STS2)

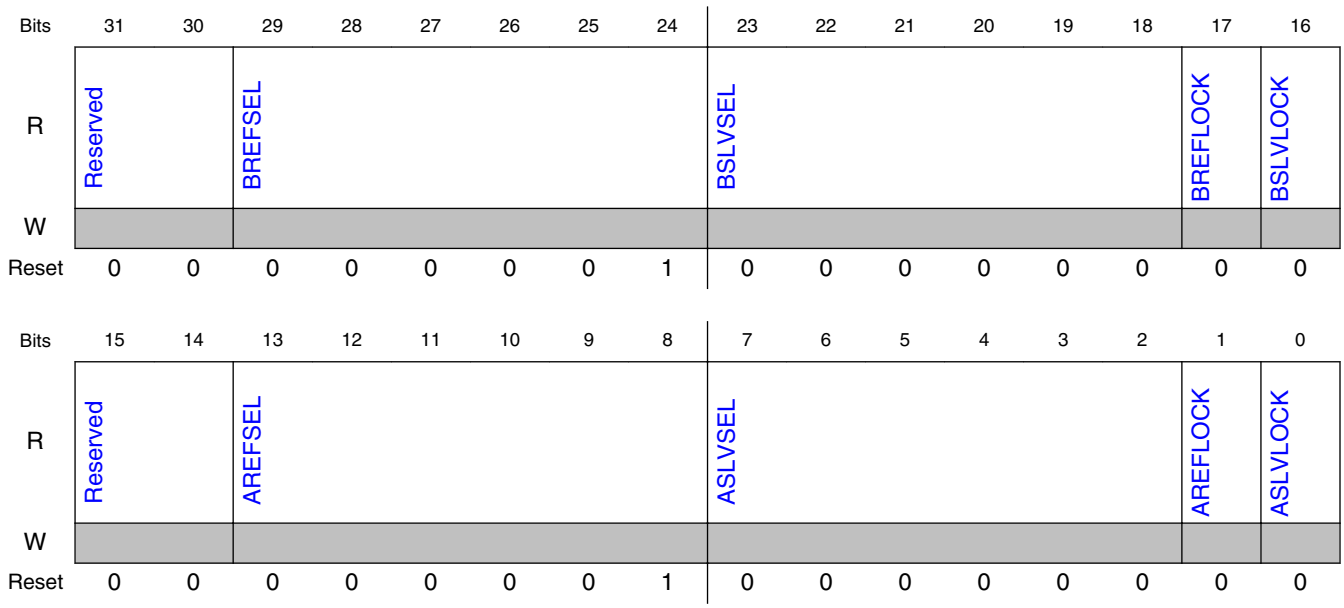
10.2.6.2.24.1 Offset

Register	Offset
STS2	E8h

10.2.6.2.24.2 Function

Indicates the status of Flash A and B sample clock DLLs.

10.2.6.2.24.3 Diagram



10.2.6.2.24.4 Fields

Field	Function
31-30 —	Reserved
29-24 BREFSEL	Flash B Sample Clock Reference Delay Line Delay Cell Number Indicates the sample clock reference delay line delay cell number for Flash B. There are 0-63 phases. • 000001b - Indicates lock phase 0. • 100000b - Indicates lock phase 63.
23-18 BSLVSEL	Flash B Sample Clock Target Delay Line Delay Cell Number Indicates the sample clock target delay line delay cell number for Flash B. There are 0-63 phases. • 000001b - Indicates lock phase 0. • 100000b - Indicates lock phase 63.
17 BREFLOCK	Flash B Sample Clock Reference Delay Line Locked 0b - Not locked 1b - Locked
16 BSLVLOCK	Flash B Sample Target Reference Delay Line Locked 0b - Not locked 1b - Locked
15-14 —	Reserved
13-8 AREFSEL	Flash A Sample Clock Reference Delay Line Delay Cell Number Indicates the sample clock reference delay line delay cell number for Flash A. There are 0-63 phases. • 000001b - Indicates lock phase 0. • 100000b - Indicates lock phase 63.

Table continues on the next page...

Field	Function
7-2 ASLVSEL	Flash A Sample Clock Target Delay Line Delay Cell Number Indicates the sample clock target delay line delay cell number for Flash A. There are 0-63 phases. • 000001b - Indicates lock phase 0. • 100000b - Indicates lock phase 63.
1 AREFLOCK	Flash A Sample Clock Reference Delay Line Locked 0b - Not locked 1b - Locked
0 ASLVLOCK	Flash A Sample Target Delay Line Locked 0b - Not locked 1b - Locked

10.2.6.2.25 AHB Suspend Status (AHBSPNDSTS)

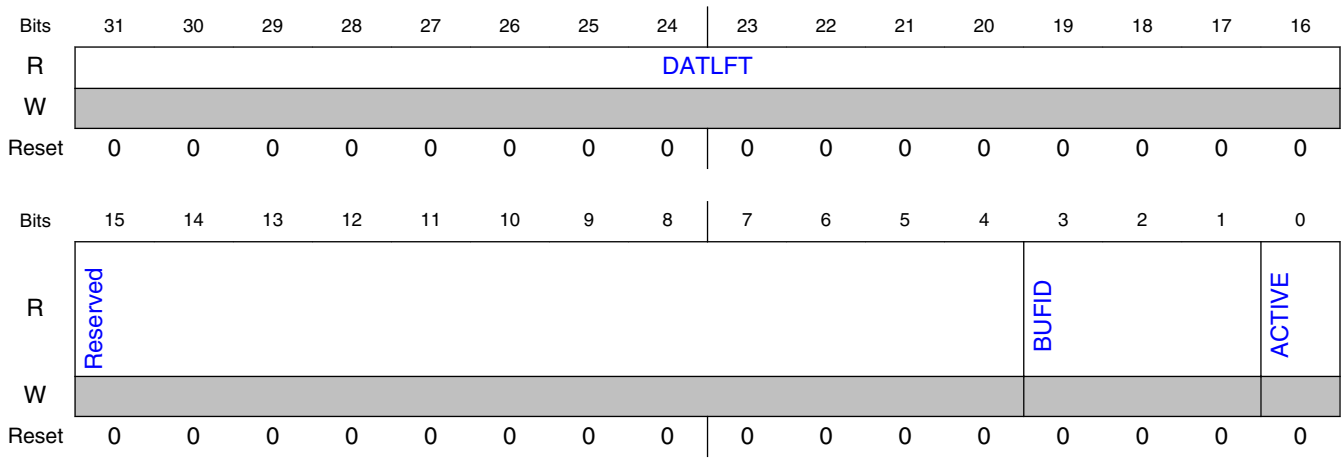
10.2.6.2.25.1 Offset

Register	Offset
AHBSPNDSTS	ECh

10.2.6.2.25.2 Function

Indicates the status of suspended AHB read prefetch command sequence. When an IP or AHB command is triggered while the arbitrator processes an AHB read sequence (prefetching additional data not for current AHB burst), the prefetch sequence is suspended. When there are no longer transactions on FlexSPI, the prefetch sequence may be resumed. FlexSPI saves only one AHB prefetch sequence. When a new prefetch sequence is suspended with an active sequence already suspended, the previous suspended sequence is removed and never resumed. See Command abort and suspend.

10.2.6.2.25.3 Diagram



10.2.6.2.25.4 Fields

Field	Function
31-16 DATLFT	Data Left Contains the data size remaining (in bytes) for a suspended command sequence.
15-4 —	Reserved
3-1 BUFID	AHB Receive Buffer ID for Suspended Command Sequence
0 ACTIVE	Active AHB Read Prefetch Suspended Indicates whether an AHB read prefetch command sequence has been suspended. 0b - No suspended AHB read prefetch command. 1b - An AHB read prefetch command sequence has been suspended.

10.2.6.2.26 IP Receive FIFO Status (IPRXFSTS)

10.2.6.2.26.1 Offset

Register	Offset
IPRXFSTS	F0h

10.2.6.2.26.2 Function

Indicates the status of IP receive FIFO.

10.2.6.2.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDCNTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								FILL							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.2.6.2.26.4 Fields

Field	Function
31-16 RDCNTR	Read Data Counter Contains counter for total data read. The data read is the value of this field × 64 bits.
15-8 —	Reserved
7-0 FILL	Fill Level of IP Receive FIFO Indicates how full the IP receive FIFO is. The fill level is the value of this field × 64 bits.

10.2.6.2.27 IP Transmit FIFO Status (IPTXFSTS)

10.2.6.2.27.1 Offset

Register	Offset
IPTXFSTS	F4h

10.2.6.2.27.2 Function

Indicates the status of IP transmit FIFO.

10.2.6.2.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WRCNTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								FILL							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.2.6.2.27.4 Fields

Field	Function
31-16 WRCNTR	Write Data Counter Contains counter for total data written. The data written is the value of this field × 64 bits.
15-8 —	Reserved
7-0 FILL	Fill Level of IP Transmit FIFO Indicates how full the IP transmit FIFO is. The fill level is the value of this field × 64 bits.

10.2.6.2.28 IP Receive FIFO Data a (RFDR0 - RFDR31)

10.2.6.2.28.1 Offset

For a = 0 to 31:

Register	Offset
RFDRa	100h + (a × 4h)

10.2.6.2.28.2 Function

Provides read access to IP receive FIFO via IPS bus. The read value is unknown for read access to invalid entries in the IP receive FIFO.

10.2.6.2.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.2.6.2.28.4 Fields

Field	Function
31-0 RXDATA	Receive Data Contains receive data. See Reading data from IP receive FIFO.

10.2.6.2.29 IP TX FIFO Data a (TFDR0 - TFDR31)

10.2.6.2.29.1 Offset

For a = 0 to 31:

Register	Offset
TFDRa	180h + (a × 4h)

10.2.6.2.29.2 Function

Provides write access to IP transmit FIFO via IPS bus.

10.2.6.2.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	TXDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	TXDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.2.6.2.29.4 Fields

Field	Function
31-0 TXDATA	Transmit Data Contains transmit data. See Writing data to IP transmit FIFO.

10.2.6.2.30 Lookup Table a (LUT0 - LUT127)

10.2.6.2.30.1 Offset

For a = 0 to 127:

Register	Offset
LUTa	200h + (a × 4h)

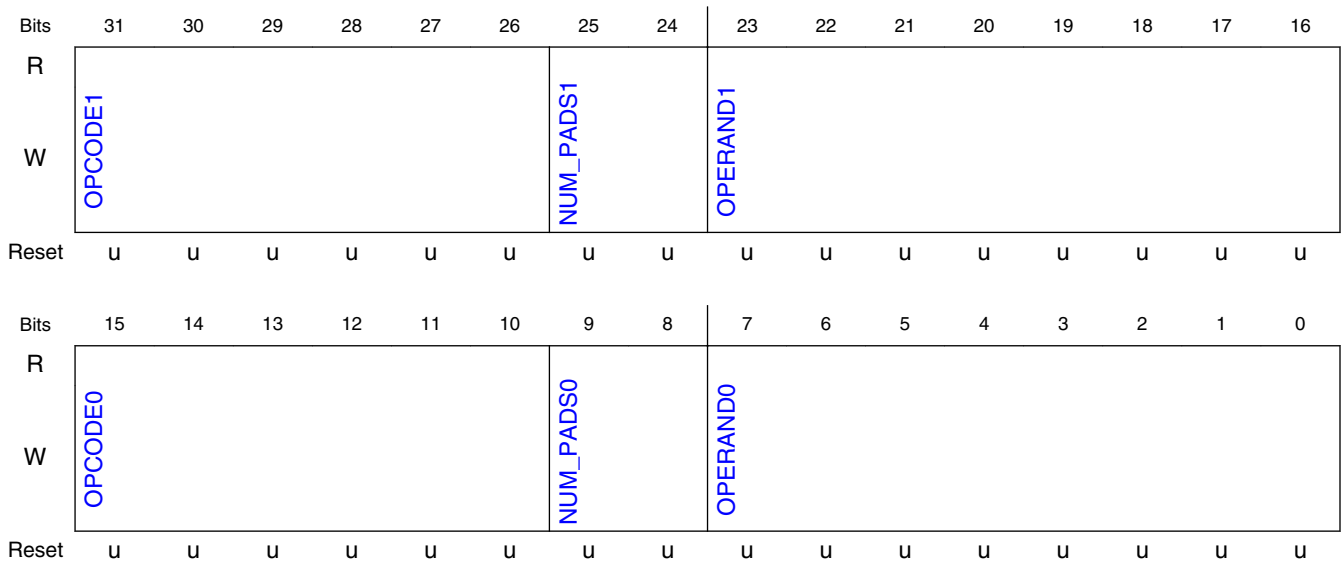
10.2.6.2.30.2 Function

Contains a lookup table for command sequences. Software should set the sequence index before triggering an IP command or AHB command. FlexSPI fetches the command sequence from LUT when an IP or AHB command is triggered. There are 32 command sequences in LUT. See Lookup table (LUT) and Programmable sequence engine.

NOTE

LUT is implemented as memory, so the reset value is unknown.

10.2.6.2.30.3 Diagram



10.2.6.2.30.4 Fields

Field	Function
31-26 OPCODE1	OPCODE1
25-24 NUM_PADS1	NUM_PADS1
23-16 OPERAND1	OPERAND1
15-10 OPCODE0	OPCODE
9-8 NUM_PADS0	NUM_PADS0
7-0 OPERAND0	OPERAND0

10.2.6.2.31 AHB Controller ID n Control (HMSTR0CR - HMSTR7CR)

10.2.6.2.31.1 Offset

For n = 0 to 7:

Register	Offset
HMSTRnCR	400h + (n × 4h)

10.2.6.2.31.2 Function

Contains the expected controller ID and the ID mask.

10.2.6.2.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSTRID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.2.6.2.31.4 Fields

Field	Function
31-16 MSTRID	Controller ID Contains the controller ID. This field is compared with the value of MASTER_ID ANDed with the MASK field. When there is a bit-by-bit match, the values are considered a match.
15-0 MASK	Mask bits for AHB Controller ID. Contains the mask for the AHB controller ID. This field is ANDed with the incoming controller ID on AHB user bits. 0000_0000_0000_0000b - Mask 0000_0000_0000_0001b - Unmask

10.2.6.2.32 HADDR REMAP Start Address (HADDRSTART)

10.2.6.2.32.1 Offset

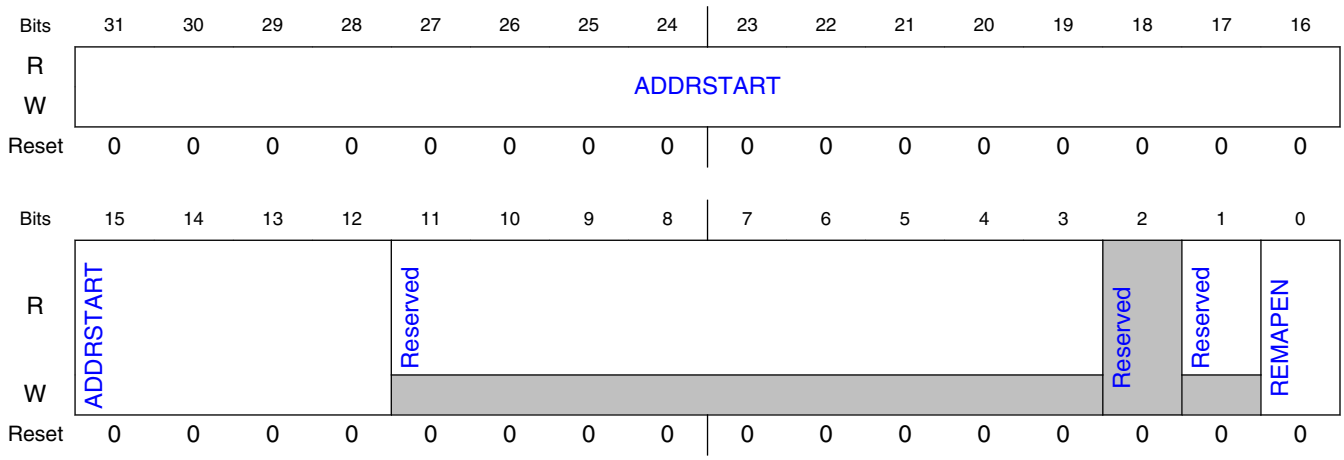
Register	Offset
HADDRSTART	420h

10.2.6.2.32.2 Function

Used to configure options for the dual-image remap feature.

See Dual image use case using HADDRSTART, HADDREND, and HADDROFFSET registers.

10.2.6.2.32.3 Diagram



10.2.6.2.32.4 Fields

Field	Function
31-12 ADDRSTART	HADDR Start Address Contains the start address of the HADDR remap range, aligned to 4 KB.
11-3 —	Reserved
2 —	Reserved
1 —	Reserved
0 REMAPEN	AHB Bus Address Remap Enable 0b - HADDR REMAP Disabled 1b - HADDR REMAP Enabled

10.2.6.2.33 HADDR REMAP END ADDR (HADDREND)

10.2.6.2.33.1 Offset

Register	Offset
HADDREND	424h

10.2.6.2.33.2 Function

Used to configure options for the dual-image remap feature.

See Dual image use case using HADDRSTART, HADDREND, and HADDROFFSET registers.

10.2.6.2.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ENDSTART															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ENDSTART				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.2.6.2.33.4 Fields

Field	Function
31-12 ENDSTART	End Address of HADDR Remap Range Contains the end address of the HADDR remap range, aligned to 4 KB.
11-0 —	Reserved

10.2.6.2.34 HADDR Remap Offset (HADDROFFSET)

10.2.6.2.34.1 Offset

Register	Offset
HADDROFFSET	428h

10.2.6.2.34.2 Function

Used to configure options for the dual-image remap feature.

See Dual image use case using HADDRSTART, HADDREND, and HADDROFFSET registers.

10.2.6.2.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDROFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDROFFSET				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.2.6.2.34.4 Fields

Field	Function
31-12 ADDROFFSET	HADDR Offset Contains the offset for HADDR. The remapped address is ADDR[31:12] = ADDR_original[31:12] + ADDROFFSET.
11-0 —	Reserved

10.2.7 AHB memory map definition

This section describes FlexSPI module AHB memory map in detail.

10.2.7.1 AHB memory map for serial flash memory and RAM access

AHB read and write access for serial flash and RAM memory are mapped to a specific address range. See the system memory map for specific address ranges supported.

AHB bus features supported for serial flash and RAM memory reading:

- Cacheable and non-cacheable access

- Prefetch enable and disable
- Burst size: 8, 16, 32, 64 bits
- All burst types: types: SINGLE, INCR, WRAP4, INCR4, WRAP8, INCR8, WRAP16, INCR16

AHB bus features for Serial RAM writing:

- Bufferable and Non-Bufferable access
- Burst size: 8, 16, 32, 64 bits
- All burst types: SINGLE, INCR, WRAP4, INCR4, WRAP8, INCR8, WRAP16, INCR16

See Flash memory access via AHB command for details about AHB access to serial flash memory and RAM.

10.2.7.2 AHB Memory Map for IP RX FIFO read access

The following address range is mapped for AHB read access to IP RX FIFO: 0x34000000 - 0x34000200.

AHB Bus feature supported for IP RX FIFO reading:

- Burst size: 8/16/32/64 bits
- All burst type: SINGLE/INCR/WRAP4/INCR4/WRAP8/INCR8/WRAP16/INCR16

Refer Reading data from IP receive FIFO for more details about IP RX FIFO reading.

10.2.7.3 AHB Memory Map for IP TX FIFO write access

The following address range is mapped for AHB write access to IP TX FIFO: 0x33008000 - 0x33008400.

AHB Bus feature supported for IP TX FIFO writing:

- Burst size: 8/16/32/64 bits
- All burst type: SINGLE/INCR/WRAP4/INCR4/WRAP8/INCR8/WRAP16/INCR16

Refer Writing data to IP transmit FIFO for more details about IP TX FIFO filling.

10.3 Ultra Secured Digital Host Controller (uSDHC)

10.3.1 Overview

The Ultra Secured Digital Host Controller (uSDHC) provides the interface between the host system and the SD/SDIO/MMC cards, as shown in Figure 10-58. The module acts as a bridge, passing host bus transactions to the SD/SDIO/MMC cards by sending commands and performing data accesses to/from the cards. It handles the SD/SDIO/MMC protocols at the transmission level.

10.3.1.1 Block Diagram

The following figure illustrates the block diagram for uSDHC.

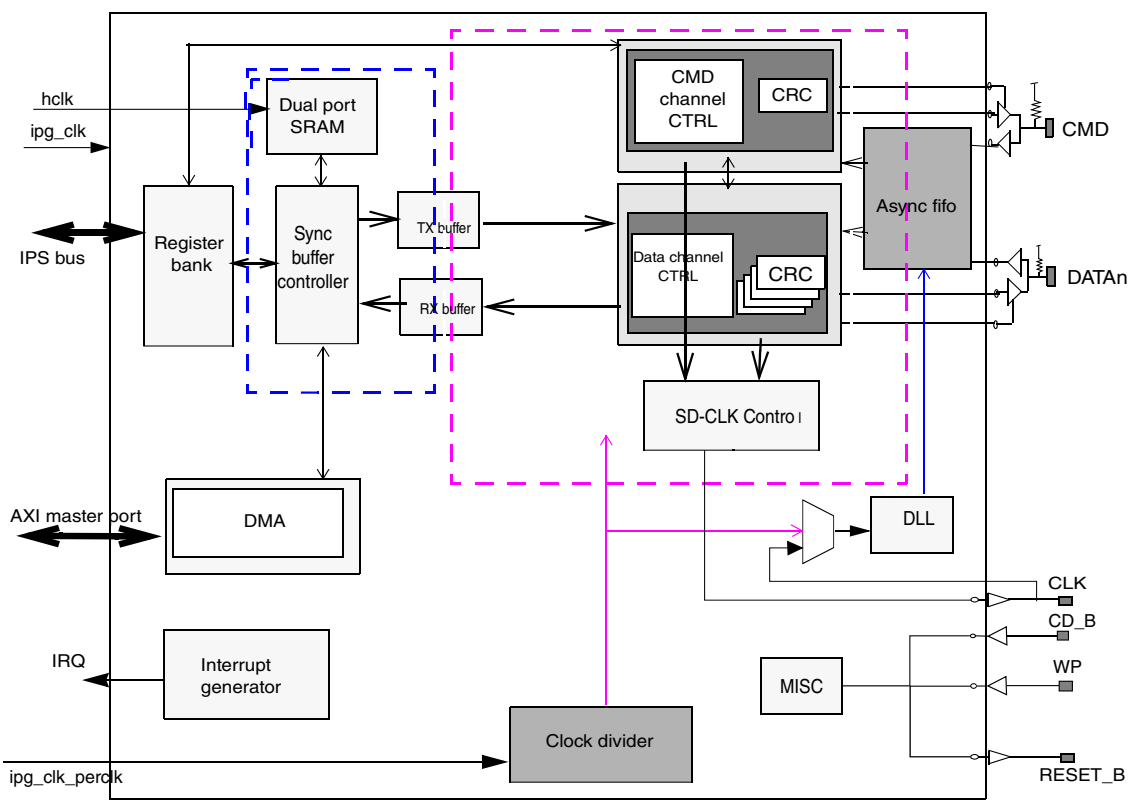


Figure 10-57. uSDHC block diagram

The figure below shows the System connection of uSDHC:

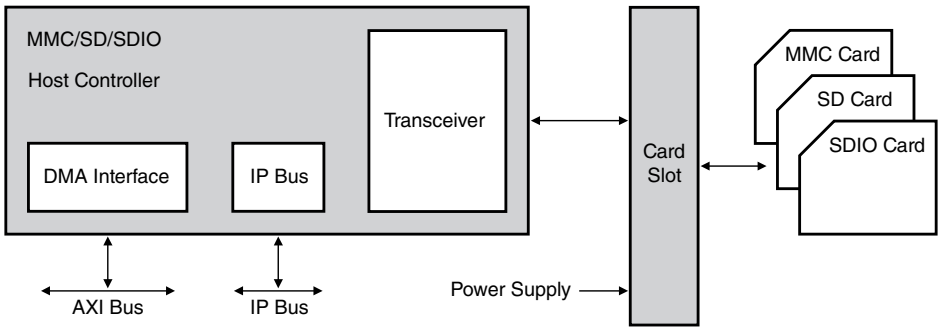


Figure 10-58. System connection of uSDHC

The following are brief descriptions of the cards supported by uSDHC:

- The Multi Media Card (MMC) is a universal low-cost data storage and communication media designed to cover a wide array of applications including mobile video and gaming. Previous MMC cards were based on a 7-pin serial bus with a single data pin, while the new high speed MMC communication is based on an advanced 11-pin serial bus designed to operate in the low-voltage range.
- The Secure Digital Card (SD) is an evolution of the old MMC technology. It is specifically designed to meet the security, capacity, performance, and environmental requirements inherent in newly emerging audio and video consumer electronic devices. The physical form factor, pin assignment, and data transfer protocol are backward/downward compatible with the old MMC (with some additions).
- Under the SD protocol, system connection can be categorized into memory card, I/O card, and combo card, which have both memory and I/O functions. The memory card invokes a copyright protection mechanism that complies with the security of the SDMI standard. The I/O card, which is also known as SDIO card, provides high-speed data I/O with low-power consumption for mobile electronic devices. For the sake of simplicity, the next figure does not show cards with reduced size or mini cards.

10.3.1.2 Features

The list given below shows the features of the uSDHC module:

- Conforms to the SD Host Controller Standard Specification version 2.0/3.0
- Compatible with the MMC System Specification version 4.2/4.3/4.4/4.41/4.5/5.0/5.1
- Compatible with the SD Memory Card Specification version 3.0 and supports the Extended Capacity SD Memory Card
- Compatible with the SDIO Card Specification version 2.0/3.0
- Designed to work with SD Memory, miniSD Memory, SDIO, miniSDIO, SD Combo, MMC, MMC plus, and MMC RS cards

- Card bus clock frequency up to 208 MHz
- Supports 1-bit/4-bit SD and SDIO modes, and 1-bit/4-bit/8-bit MMC modes
  - Up to 832 Mbps of data transfer for SDIO cards using four parallel data lines in the Single Data Rate (SDR) mode
  - Up to 400 Mbps of data transfer for SDIO card using four parallel data lines in the Dual Data Rate (DDR) mode
  - Up to 832 Mbps of data transfer for SDXC cards using four parallel data lines in the Single Data Rate (SDR) mode
  - Up to 400 Mbps of data transfer for SDXC card using four parallel data lines in the Dual Data Rate (DDR) mode
  - Up to 1600 Mbps of data transfer for MMC cards using eight parallel data lines in the Single Data Rate (SDR) mode
  - Up to 3200 Mbps of data transfer for MMC cards using eight parallel data lines in the Dual Data Rate (DDR) mode
- Supports single block/multi-block read and write
- Supports block sizes of 1 ~ 4096 bytes
- Supports the write protection switch for write operations
- Supports both synchronous and asynchronous abort
- Supports pause during the data transfer at block gap
- Supports SDIO Read Wait and Suspend Resume operations
- Supports Auto CMD12 for multi-block transfer
- Host can initiate non-data transfer command while data transfer is in progress
- Allows cards to interrupt the host in 1-bit and 4-bit SDIO modes; also supports interrupt period
- Embodies two fully configurable 256x32-bit FIFO for read/write data
- Supports internaland external DMA capabilities
- Support voltage selection by configuring vendor-specific register bit
- Supports Advanced DMA to perform linked memory access
- support Command queue mechanism

NOTE

This block can support all the above-listed speed mode and maximum data throughput. However, these may be specific to the device. See the corresponding chip-specific information or the device data sheet for accurate details.

10.3.2 Functional description

The following sections provide a brief functional description of the major system blocks, including the data buffer, DMA AXI interface, register bank as well as IP Bus interface, dual-port memory wrapper, data/command controller, clock and reset manager, and clock generator.

10.3.2.1 Modes and operations

10.3.2.1.1 Data transfer modes

The uSDHC module can select the following modes for data transfer:

- SD 1-bit
- SD 4-bit
- MMC 1-bit
- MMC 4-bit
- MMC 8-bit
- Identification mode (up to 400 kHz)
- MMC full-speed mode (up to 26 MHz)
- MMC high-speed mode (up to 52 MHz)
- MMC HS200 mode(up to 200 MHz)
- MMC HS400 mode (200 MHz both edges)
- MMC DDR mode (52 MHz both edges)
- SD/SDIO full-speed mode (up to 25 MHz)
- SD/SDIO high-speed mode (up to 50 MHz)
- SD/SDIO UHS-I mode(up to 208 MHz in SDR mode, up to 50 MHz in the DDR mode)

NOTE

This block can support all the above listed speed mode and maximum clock frequency. However, these may be specific to the device. See the corresponding chip-specific information or the device data sheet for accurate details.

10.3.2.2 Data buffer

The uSDHC module uses one configurable data buffer to transfer data between the system bus (IP bus or advanced extensible interface (AXI) bus) and the SD card in an optimized manner, maximizing throughput between the two clock domains (IP peripheral clock and the master clock).

The buffer is used as a temporary storage for transferring data between the host system and the card. The watermark levels for read and write are both configurable and can range between 1 to 128 words. When using internal DMA mode, The watermark level for read should be configured to 16 and watermark level for write should be configured to integral multiple of 16 to get max AXI bus throughput. The next figure provides the uSDHC buffer scheme.

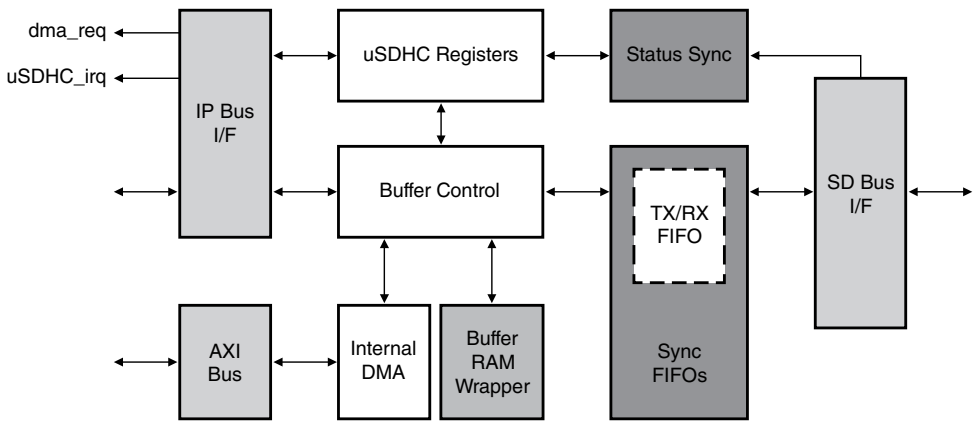


Figure 10-59. uSDHC buffer scheme

Here are 3 transfer modes to access the data buffer:

- CPU polling mode:
  - For a host-read operation, when the number of words received in the buffer meets or exceeds the RD_WML watermark value, by polling the BRR bit, the host driver can read the Buffer Data Port register to fetch the amount of words set in the RD_WML register from the buffer. The write operation is similar. For more information on the process of writing operation, see Write operation sequence.
- External DMA mode:
  - For a read operation, when there are more words received in the buffer than the amount set in the WTMK_LVL[RD_WML], a DMA request is sent out to inform the external DMA to fetch the data. The request will be immediately de-asserted when there is an access on the Buffer Data Port (DATA_BUFF_ACC_PORT) register. If the number of words in the buffer after the current burst meets or exceeds RD_WML value, the DMA request is asserted again. For instance, if there are twice as many words in the buffer as there are in

the WTMK_LVL[RD_WML] value, there are two successive DMA requests with only one cycle of de-assertion between. The write operation is similar. Note the accesses CPU polling mode and external DMA mode both use the IP bus, and if the external DMA is enabled, in both modes an external DMA request is sent when the buffer is ready.

- Internal DMA mode (includes simple and advanced DMA accesses):
  - The internal DMA access, either by simple or advanced DMA, is over the AXI bus. For internal DMA access mode, the external DMA request will never be sent out.

For a read operation, when there are more words in the buffer than the amount set in the RD_WML register, the internal DMA starts fetching data over the AXI bus.

The Write operation functions in a similar manner—sequential and contiguous access is necessary to ensure that the pointer address value is correct. Random or skipped access is not possible. The byte order, by reset, is little endian mode. The actual byte order is swapped inside the buffer, according to the endian mode configured by software (see the following figures). For a host write operation, the byte order is swapped after the data is fetched from the buffer and ready to send to the SD Bus. For a host read operation, the byte order is swapped before the data is stored in the buffer.

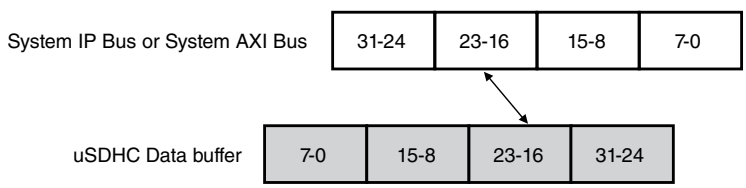


Figure 10-60. Data swap between system bus and uSDHC data buffer in the byte little endian mode

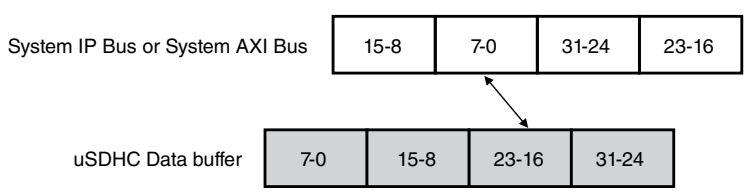


Figure 10-61. Data swap between system bus and uSDHC data buffer in half word big endian mode

10.3.2.2.1 Write operation sequence

There are 3 ways to write data into the buffer when the user transfers data to the card:

- External DMA through the uSDHC DMA request signal

- Processor core polling through the BWR bit in the Interrupt Status register (interrupt or polling)
- Internal DMA

When the internal DMA is not used, the DMAEN bit in the Transfer Type register is not set when the command is sent, uSDHC asserts a DMA request when the amount of buffer space exceeds the value set in the WTMK_LVL[WR_WML] bit field and is ready for receiving new data. At the same time, uSDHC sets the BWR bit. The buffer write ready interrupt is generated if it is enabled by software.

When internal DMA is used, uSDHC does not inform the system before all the required number of bytes are transferred (if no error is encountered). When an error occurs during the data transfer, uSDHC aborts the data transfer and abandons the current block. The host driver should read the contents of the DMA System Address register to obtain the starting address of the abandoned data block. If the current data transfer is in multi-block mode, uSDHC does not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The host driver sends CMD12 in this scenario and restarts the write operation from that address. It is recommended that a software reset for Data be applied before the transfer is restarted.

The uSDHC module does not start data transmission until the buffer has been filled with the number of words set in the WR_WML register. If the buffer is empty and the host system does not write data in time, uSDHC stops the CLK to avoid the data buffer underrun situation.

10.3.2.2.2 Read operation sequence

There are 3 ways to read data from the buffer when the data is received from the card:

- External DMA through uSDHC DMA request signal
- Processor core polling through the BRR bit in the Interrupt Status register (interrupt or polling)
- Internal DMA

When internal DMA is not used (DMAEN bit in Transfer Type register is not set when the command is sent), uSDHC asserts a DMA request when the amount of data exceeds the value set in the RD_WML register, which is available and ready for system fetching data. At the same time, uSDHC sets the BRR bit. The buffer read ready interrupt is generated if it is enabled by the software.

When internal DMA is used, uSDHC does not inform the system before all the required number of bytes are transferred (if no error is encountered). When an error occurs during the data transfer, uSDHC aborts the data transfer and abandons the current block. The host driver should read the content of the DMA System Address register to get the

starting address of the abandoned data block. If the current data transfer is in multi-block mode, uSDHC does not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The host driver sends CMD12 in this scenario and restarts the read operation from that address. It is recommended that a software reset for data be applied before the transfer is restarted.

For any write transfer mode, uSDHC does not start data transmission until the number of words set in the RD_WML register are in buffer. If the buffer is full and the host system does not read data in time, uSDHC stops the CLK to avoid the data buffer overrun situation.

10.3.2.2.3 Data buffer and block size

In the uSDHC module, the data buffer can hold up to 256 words (32-bit) and the watermark levels for write and read can be configured accordingly. The user needs to know the buffer size for the buffer operation during a data transfer to utilize it in the most optimized way.

The host driver may configure the value according to the system and requirement.

During a multi-block data transfer, the block length can be set to any value between 1 and 4096 bytes, satisfying the requirements of the external card. The only restriction is from the external card, which can be limited in size or support of a partial block access (which is not the integer times of 512 bytes).

As uSDHC treats each block individually, for block sizes which are not multiples of four (not word-aligned), stuffed bytes are required at the end of each block. For example, if the block size is 7 bytes and there are 12 blocks to write, the software side must write twice for each block. For each block, the ending byte is abandoned by uSDHC because it only sends 7 bytes to the card and picks data from the following system write, resulting in 24 beats of write access in total.

10.3.2.2.4 Dividing large data transfer

This SDIO command CMD53 definition limits the maximum data size of data transfers according to the following formula:

Max data size = Block size x Block count

The length of a multiple block transfer needs to be in multiples of block size. If the total data length cannot be divided evenly into a multiple of the block size, then there are two options to transfer the data. These two options depend on the function and the card design. Option 1 is for the host driver to split the transaction. The remainder of the block

size data is then transferred by using a single block command at the end. Option 2 is to add dummy data in the last block to fill the block size. For option 2, the card must manage the removal of the dummy data.

See the figure below for an example showing the division of large data transfers, assuming a kind of WLAN SDIO card that only supports a block size of up to 64 bytes. Although uSDHC supports a block size of up to 4096 bytes, the SDIO can only accept a block size of less than 64 bytes, so the data must be divided (see the example below).

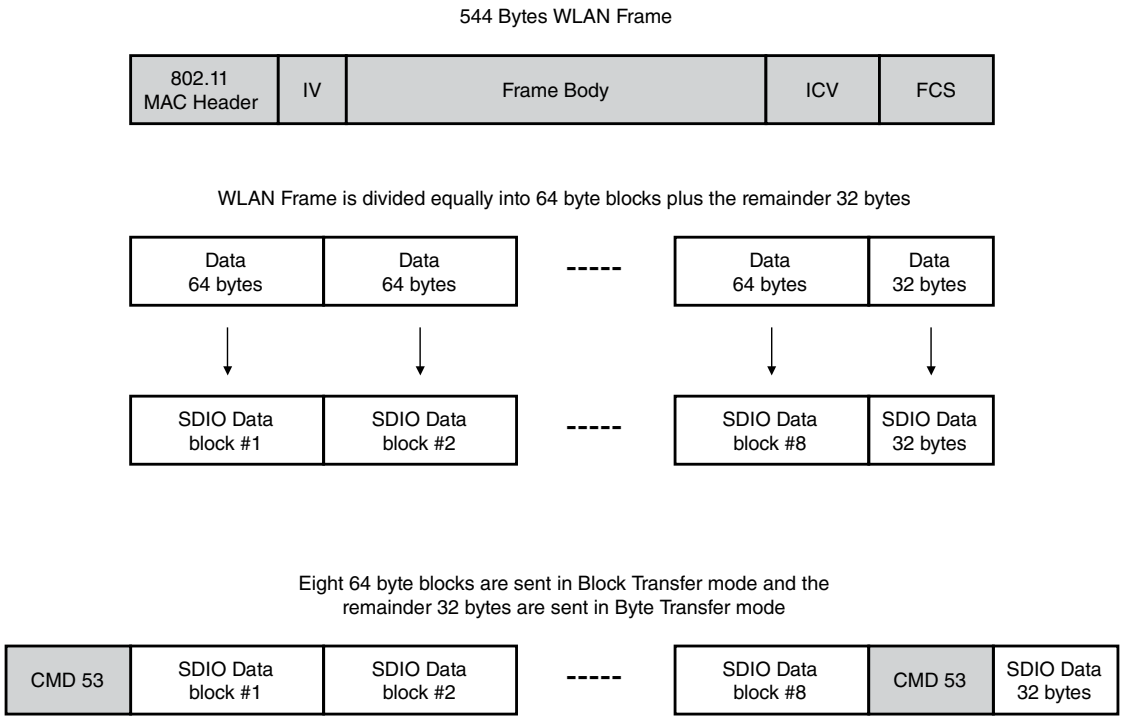


Figure 10-62. Example for dividing large data transfers

10.3.2.2.5 External DMA Request

When the internal DMA is not in use and external DMA is enabled, the Data Buffer will generate a DMA request to the system. During a write operation, when the number of WR_WML words can be held in the buffer free space, the signal uSDHC_dreq_b is asserted to 0, informing the Host System of a DMA write.

The BWR bit in the Interrupt Status register is also set, as long as the BWRSEN bit in the Interrupt Status Enable register is set. The DMA request is de-asserted after several accesses to the Data Port (DATA_BUFF_ACC_PORT) register are made while the buffer's free space can't meet the watermark condition (free space> write watermark level).

On read operation, when the number of RD_WML words are already in the buffer, the signal uSDHC_dreq_b is asserted to 0, informing the Host System for a DMA read. The BRR bit in the Interrupt Status register is also set, as long as the BRRSEN bit in the Interrupt Status Enable register is set. The DMA request is de-asserted after several accesses to the Data Port register are made while the buffer's data can't meet the watermark condition (the number of data in buffer > read watermark level).

If the DMA burst length can't change during a data transfer for an external DMA transfer, the watermark level (read or write) must be a divisor of the block size. If it is not, transferring the block may cause buffer under-run (read operation) or over-run (write operation). For example, if the block size is 512 bytes, the watermark level of read (or write) must be a power of two between 1 and 128. For processor core polling access there is no such issue, as the last access in the block transfer can be controlled by software. The watermark level can be any value, even larger than the block size (but no greater than 128 words) because the actual number of bytes transferred by the software can be controlled and does not exceed the block size in each transfer.

The uSDHC also supports non-word aligned block size, as long as the card supports that block size. In this case, the watermark level should be set as the number of words. For example, if the block size is 31 bytes, the watermark level can be set to any number of words. For this case, the BLKSIZE bits of the Block Attribute register will be set as 1fh. For the CPU polling access, the burst length can be 1 to 128 words, without restriction. This is because the software will transfer 8 words, and the uSDHC will also set the BWR or BRR bits when the remaining data does not violate data buffer. See for more details about the dynamic watermark level of the data buffer.

For the above example, even though 8 words are transferred via the DATA_BUFF_ACC_PORT register, the uSDHC will transfer only 31 bytes over the SD Bus, as required by the BLKSIZE bits. In this data transfer, with non-word aligned block size, the endian mode should be set cautiously or invalid data will be transferred to and from the card.

10.3.2.3 DMA AXI interface

The internal DMA implements a DMA engine and the AXI master. When the internal DMA is enabled, but the BWR and BRR bits are set if the BWRSEN and BRRSEN bits have been set in the Interrupt Status Enable register.

See the figure below for an illustration of the DMA AXI interface block.

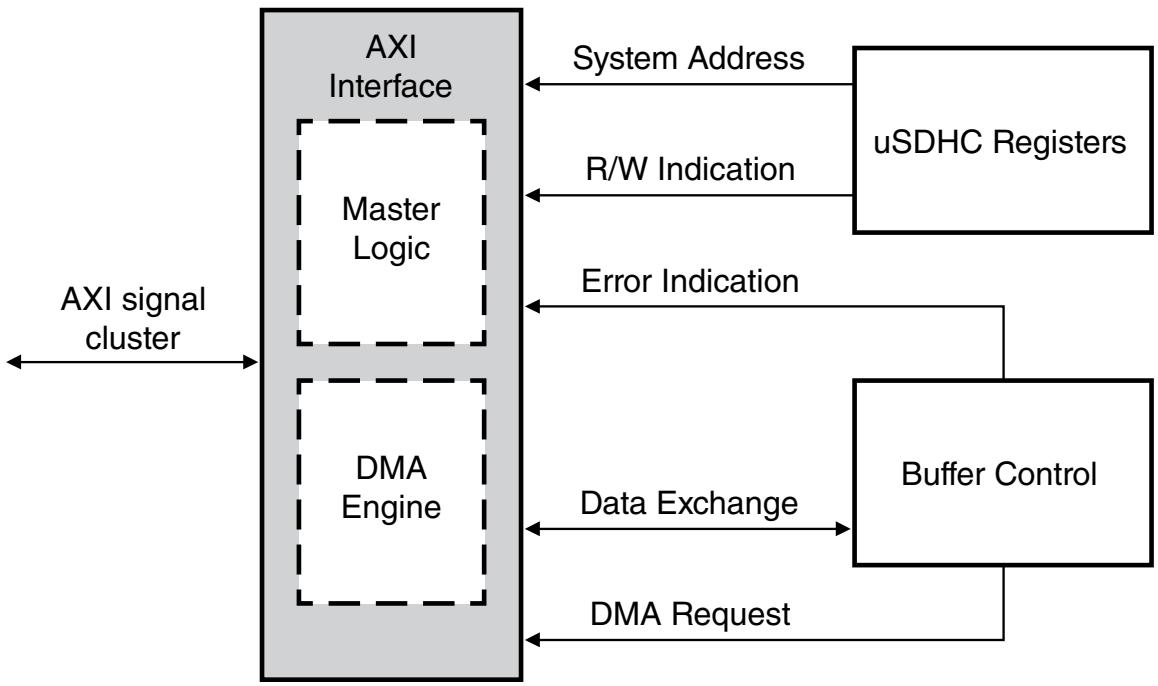


Figure 10-63. DMA AXI interface block

10.3.2.3.1 Internal DMA request

If the watermark level requirement is met in data transfer or if the last data of current block is ready in the data buffer, and the Internal DMA is enabled, the data buffer block sends a DMA request to AXI interface.Meanwhile, the external DMA request signal (uSDHC_dreq_b) is disabled.

The delay in response from the internal DMA engine depends on the system AXI bus loading and the priority assigned to uSDHC. The DMA engine does not respond to the request during its burst transfer, but is ready to serve as soon as the burst is over. The data buffer de-asserts the request if the data buffer space (for write) or bytes in data buffer is smaller than the watermark level. Upon access to the buffer by internal DMA, the data buffer updates its internal buffer pointer, and when the watermark level is satisfied or the last data of the current block is ready in the data buffer, another DMA request is sent.

The data transfer is in the block unit, and the subsequent watermark level is always set as the remaining number of words. For instance, for a multi block data read with each block size of 31 bytes, and the watermark level set to six words (24 bytes). After the first burst transfer, if there are greater than or equal to two words in the buffer, which might contain some data of the next block), another DMA request is sent. This is because the remaining number of words to send for the current block is ceiling((31 - 6 * 4) / 4) = 2. The uSDHC module reads two words out of the buffer, with seven valid bytes and one stuffed byte.

10.3.2.3.2 AXI master interface

It is possible that the internal AXI DMA engine could fail during the data transfer. Upon detection of an AXI bus error during DMA transfer, the DMA engine stops the transfer and goes to the idle state. At that point, the internal data buffer stops receiving incoming data and sending out data. The DMAE bit in the Interrupt Status register is generated to host CPU to report a bus error condition.

After the DMAE interrupt is received, the software sends a CMD12 to abort the current transfer and read the DS_ADDR bits of the DMA System Address register to get the starting address of the corrupted block. After the DMA error is fixed, the software should apply a data reset and restart the transfer from this address to recover the corrupted block. DMA operation resumes when the interrupt is serviced by the software.

10.3.2.3.3 ADMA engine

In the SD host controller standard, a new DMA transfer algorithm called the Advanced DMA (ADMA) is defined. For simple DMA, after the page boundary is reached, a DMA interrupt is generated and the new system address is programmed by the host driver.

The ADMA defines the programmable descriptor table in the system memory. The host driver can calculate the system address at the page boundary and program the descriptor table before executing ADMA. It reduces the frequency of interrupts to the host system. Therefore, higher speed DMA transfers could be realized because the host MCU intervention is not be needed during long DMA-based data transfers.

There are two types of ADMA in host controller: ADMA1 and ADMA2. ADMA1 can support data transfer of 4KB aligned data in system memory, and ADMA2 eliminates the restriction so that the data of any location and any size can be transferred in system memory. Their formats of Descriptor Table are different.

ADMA can recognize all kinds of descriptors defined in SD host controller Standard, and if the "End" flag is detected in the descriptor, ADMA stops after this descriptor is processed.

10.3.2.3.3.1 ADMA concept and descriptor format

ADMA1 includes the following descriptors:

- Valid/invalid descriptor
- Nop descriptor
- Set data length descriptor
- Set data address descriptor
- Link descriptor
- Interrupt flag and end flag in descriptor

ADMA2 includes the following descriptors:

- Valid/invalid descriptor
- Nop descriptor
- Rsv descriptor (new in ADMA2)
- Set data length and address descriptor
- Link descriptor
- Interrupt flag and end flag in descriptor

ADMA starts read/write operation after it reaches the tran state, using the data length and data address analyzed from most recent descriptor(s).

For ADMA1, the valid data length descriptor is the last set type descriptor before the tran type descriptor. Every tran type descriptor triggers a transfer, and the transfer data length is extracted from the most recent set type descriptor. If there is no set type descriptor after the previous Trans descriptor, the data length is the value for previous transfer, or 0 if no set descriptor is ever met.

For ADMA2, the tran type descriptor contains both data length and transfer data address, so the tran type descriptor itself can start a data transfer.

See the figure below for the format of the descriptor table for ADMA1.

Address/ Page Field		Address/ Page Field		Attribute Field					
31	12	11	6	5	4	3	2	1	0
Address or Data Length		000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	31- 28	27- 12
0	0	Nop	No Operation	Don't Care	
0	1	Set	Set Data Length	0000	Data Length
1	0	Tran	Transfer Data	Data Address	
1	1	Link	Link Descriptor	Descriptor Address	

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 10-64. Format of the ADMA1 descriptor table

See the figure below for the concept and access method of the descriptor table for ADMA1.

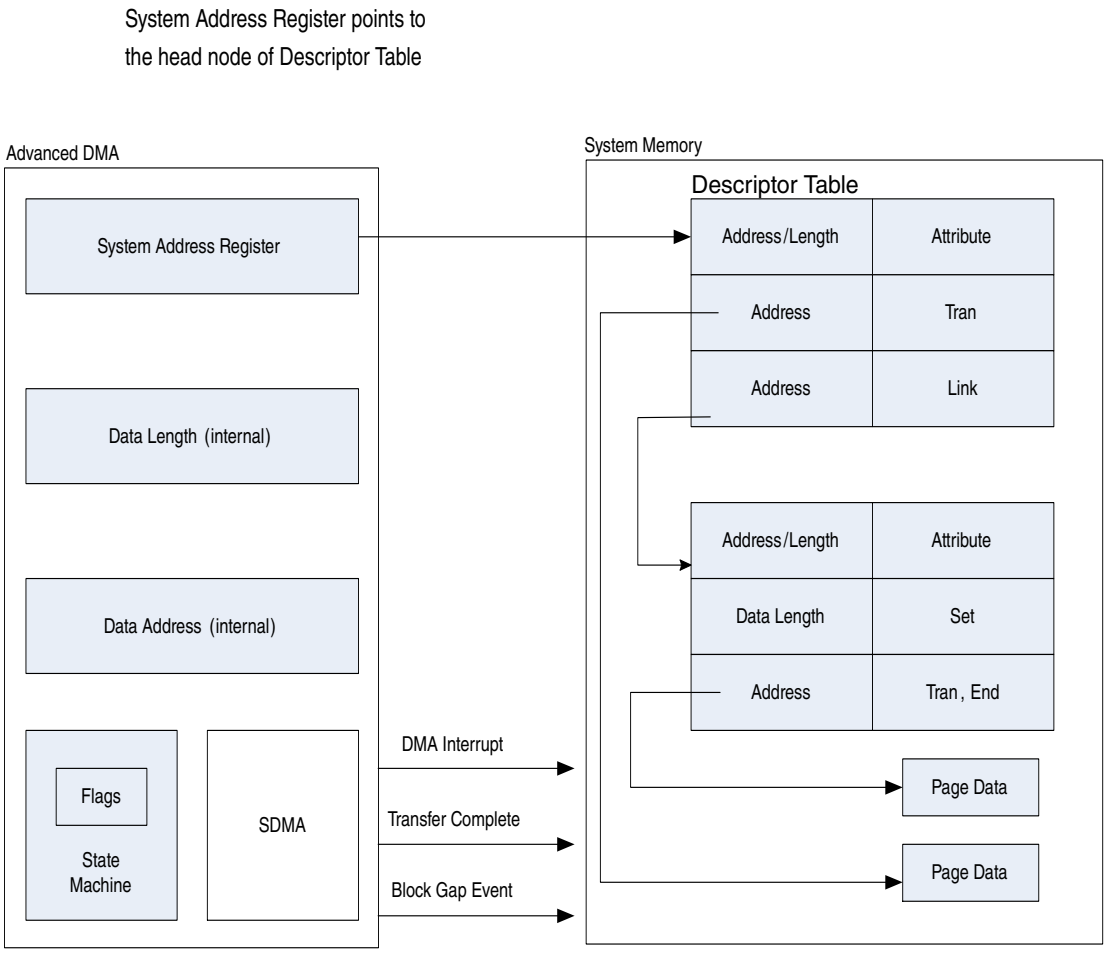


Figure 10-65. Concept and access method of the ADMA1 descriptor table

The figure below explains the ADMA2 format. ADMA2 deals with the lower 32-bit first, and then the higher 32-bit. If the 'Valid' flag of descriptor is 0, it ignores the higher 32-bit. The Address field should be set to word aligned (lower 2-bit is always set to 0). Data length is in byte unit.

Address Field		Length		Reserved		Attribute Field					
63	32	31	16	15	06	05	04	03	02	01	00
32-bit Address		16-bit length		0000000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	Operation
0	0	Nop	No Operation	Don't Care
0	1	Rsv	Reserved	Same as Nop. Read this line and go to next one
1	0	Tran	Transfer Data	Transfer data with address and length set in this descriptor line
1	1	Link	Link Descriptor	Link to another descriptor

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 10-66. Format of the ADMA2 descriptor table

See the figure below for the concept and access method of the descriptor table for ADMA2.

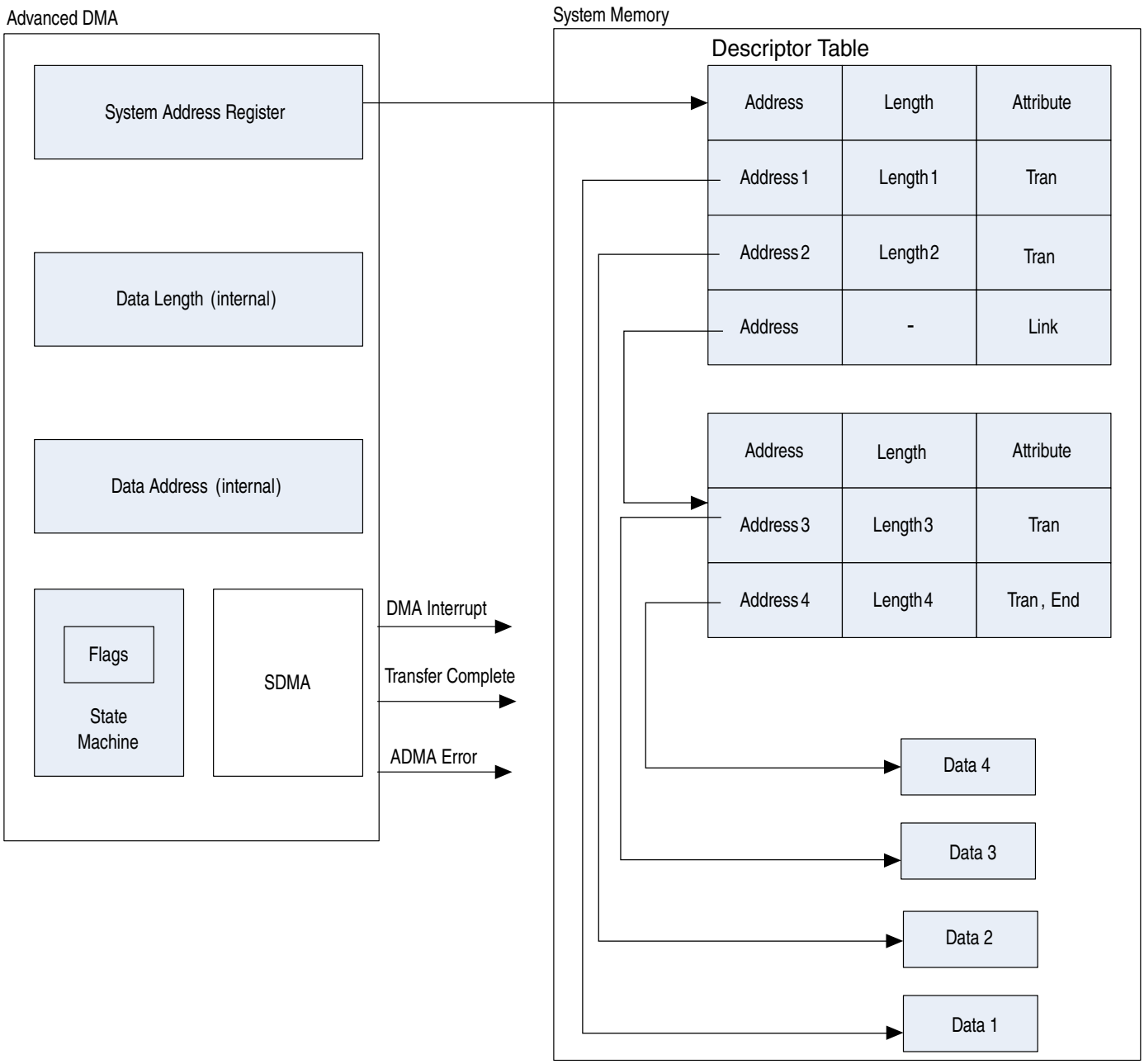


Figure 10-67. Concept and access method of the ADMA2 descriptor table

10.3.2.3.3.2 ADMA interrupt

If the interrupt flag descriptor is set, ADMA generates an interrupt according to various types of descriptors.

For ADMA1:

- Set type of descriptor: interrupt is generated when data length is set.
- Tran type descriptor: interrupt is generated when this transfer is complete.
- Link type of descriptor: interrupt is generated when new descriptor address is set.
- Nop type of descriptor: interrupt is generated just after this descriptor is fetched.

For ADMA2:

- Tran type of descriptor: interrupt is generated when this transfer is complete.
- Link type of descriptor: interrupt is generated when new descriptor address is set.
- Nop/Rsv type of descriptor: interrupt is generated just after this descriptor is fetched.

10.3.2.3.3.3 ADMA error

The ADMA stops whenever an error is encountered. These errors include:

- Fetching descriptor error
- AXI response error
- Data length mismatch error

An ADMA descriptor error is generated when it fails to detect a "valid" flag in the descriptor. If an ADMA descriptor error occurs, the interrupt is not generated even if the "Interrupt" flag of this descriptor is set.

When BLKCNTEN bit is set, data length set in buffer must be equal to the whole data length set in descriptor nodes, otherwise data length mismatch error is generated.

When BLKCNTEN bit is not set, then the whole data length set in descriptor is a multiple of block lengths; otherwise, when data set in the descriptor nodes are not performed at block boundaries, then data mismatch errors occur.

10.3.2.4 Register bank with IP bus interface

Register accesses through the IP bus interface are on the register bank.

See the figure below for the block diagram.

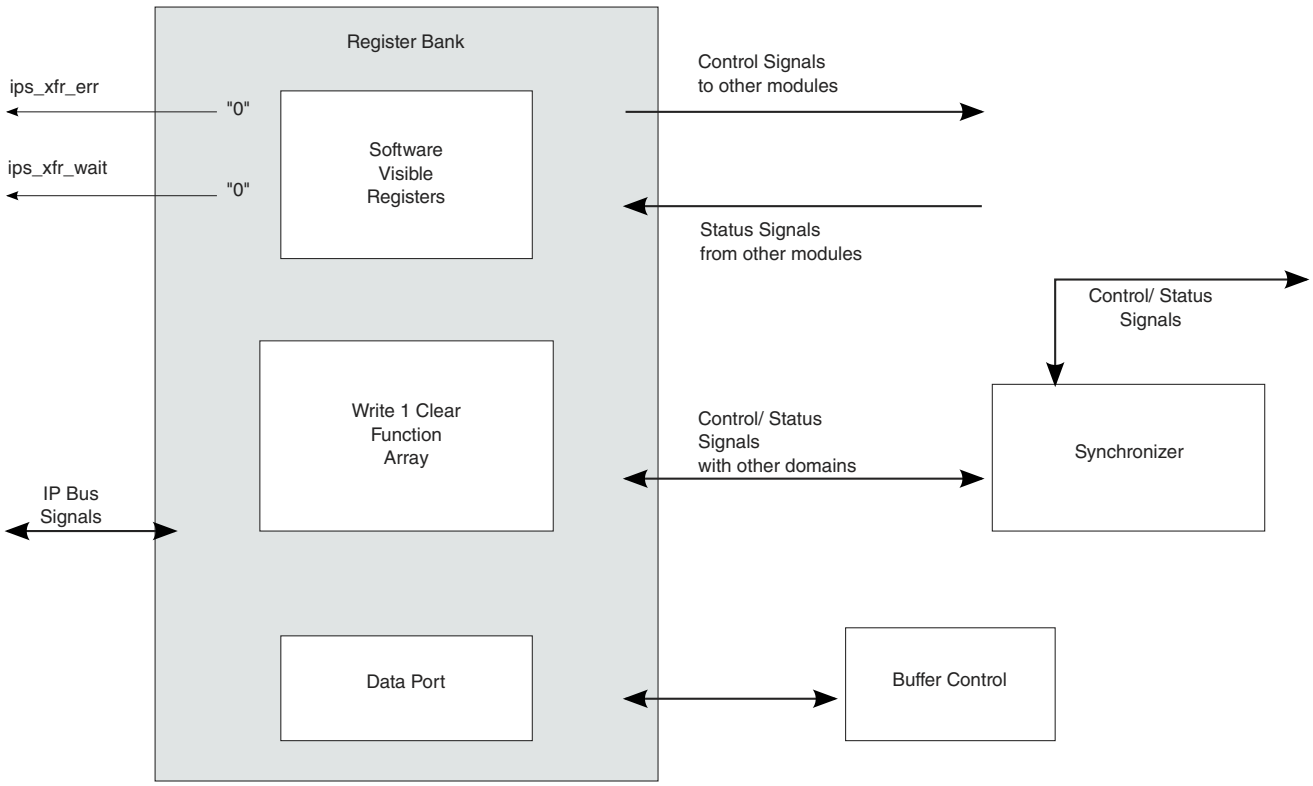


Figure 10-68. Register bank diagram

Only a 32-bit access is allowed, and no partial read/write is supported; therefore, all accesses are word aligned.

Access to an unimplemented address within the register memory map space does not generate a transfer error.

10.3.2.4.1 SD protocol unit

The SD protocol unit deals with all SD protocol affairs.

The SD protocol unit performs the following functions:

- Acts as the bridge between the internal buffer and the SD bus
- Sends the command data as well as its argument serially
- Stores the serial response bit stream into corresponding registers
- Detects the bus state on the CMD/DAT lines
- Monitors the interrupt from the SDIO card
- Asserts the read wait signal
- Gates off the SD clock when buffer is announcing danger status
- Detects the write protect state

The SD protocol unit consists of four submodules:

- SD control misc
- Command control
- Data control
- Clock control

10.3.2.4.2 SD control misc

In the SD control misc unit:

- The card detection (including CD_B and DATA3 for card detection) and card interrupt are implemented.
- The module monitors the signal level on all the eight data lines and the command lines. It directly routes the level values into the register bank.
- The module also detects the Write Protect (WP) line. If WP is active, writes to the register bank are ignored.

10.3.2.4.3 SD clock control

If the internal data buffer is near full (for read) or near empty (for write), the SD clock must be gated off to avoid buffer over/under-run. This module asserts the gate of the output SD clock to shut the clock off. After the buffer has space (for read) or has data (for write), the clock gate of this module opens, and the SD clock is active again.

10.3.2.4.4 Command control

The Command Control module deals with the transactions on the CMD line.

See the figure below for an illustration of the structure for the Command CRC shift register.

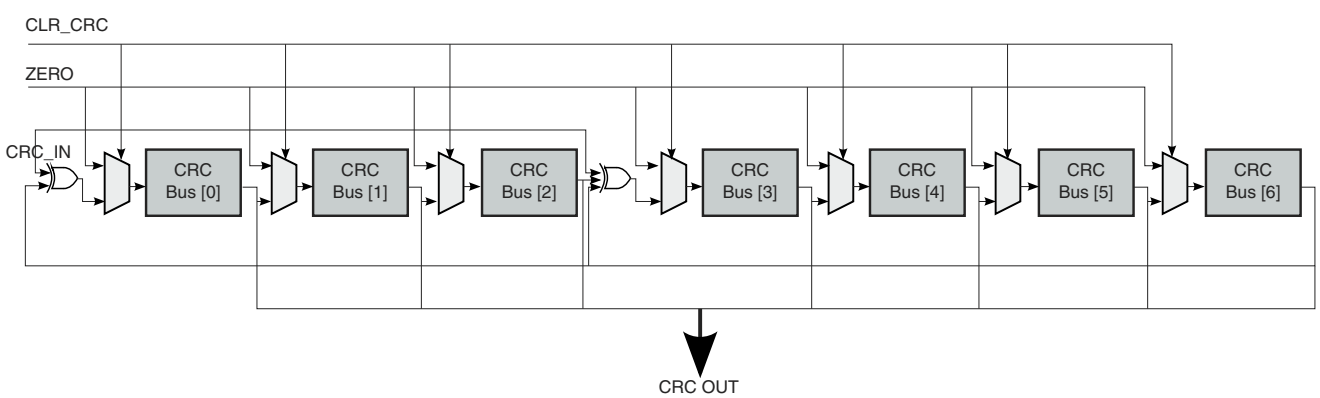


Figure 10-69. Command CRC shift register

The CRC polynomials for the CMD are as follows:

Generator polynomial: $G(x) = x^7 + x^3 + 1$
$M(x) = (\text{first bit}) * x^n + (\text{second bit}) * x^{n-1} + ... + (\text{last bit}) * x^0$
$CRC[6:0] = \text{Remainder } [(M(x) * x^7) / G(x)]$

10.3.2.4.5 Data control

The data agent deals with the transactions on the eight data lines. Moreover, this module also detects the busy state on the DATA0 line and generates the read wait state by the request from the transceiver.

The CRC polynomials for the data are as follows:

Generator polynomial: $G(x) = x^{16} + x^{12} + x^5 + 1$

$M(x) = (\text{first bit}) * x^n + (\text{second bit}) * x^{n-1} + ... + (\text{last bit}) * x^0$

$CRC[15:0] = \text{Remainder } [(M(x) * x^{16}) / G(x)]$

10.3.2.5 Card insertion and removal detection

The uSDHC module uses either the DATA3 pin or the CD_B pin to detect card insertion or removal. When there is no card on the MMC/SD bus, the DATA3 is pulled to a low voltage level by default.

When any card is inserted to or removed from the socket, uSDHC detects the logic value changes on the DATA3 pin and generates an interrupt. When the DATA3 pin is not used for card detection (for example, it is implemented in GPIO), the CD_B pin must be connected for card detection. Whether DATA3 is configured for card detection or not, the CD_B pin is always a reference for card detection. Whether the DATA3 pin or the CD_B pin is used to detect card insertion, uSDHC sends an interrupt (if enabled) to inform the Host system that a card is inserted.

10.3.2.6 Power management and wakeup events

When there is no operation between uSDHC and the card through the SD bus, the user can completely disable the peripheral clock and base clock in the chip-level clock control module to save power. When the user needs to use uSDHC to communicate with the card, it can enable the clock and start the operation.

In some circumstances, when the clocks to uSDHC are disabled, for instance, when the system is in low-power mode, there are some events for which the user needs to enable the clock and handle the event. These events are called wakeup interrupts. The uSDHC module can generate these interrupts even when there are no clocks enabled. The three interrupts that can be used as wakeup events are these:

- Card removal interrupt
- Card insertion interrupt
- Interrupt from SDIO card (not available in multiple block data transfers)

The uSDHC module offers a power management feature. By clearing the clock enabled bits in the System Control (SYS_CTRL) register, the clocks are gated in the low position to uSDHC. For maximum power saving, the user can disable all the clocks to uSDHC when there is no operation in progress.

These three wakeup events (or wakeup interrupts) can also be used to wakeup the system from low-power modes.

NOTE

To make the interrupt a wakeup event, when all the clocks to uSDHC are disabled or when the entire system is in low power mode, the corresponding wakeup enabled bit needs to be set. Refer to Protocol Control (PROT_CTRL) for more information on the uSDHC Protocol Control register.

10.3.2.6.1 Setting wakeup events

For uSDHC to respond to a wakeup event, the software must set the respective wakeup enable bit before the CPU enters the sleep mode.

Before the software disables the host clock, it should ensure that all the following conditions have been met:

- No read or write transfer is active
- Data and command lines are not active
- No interrupts are pending
- Internal data buffer is empty

10.3.2.7 MMC fast boot

The Embedded MultiMediaCard (eMMC4.3) specification adds a fast boot feature that requires hardware support. There are two types of fast boot modes in the eMMC4.3 specification: boot operation and alternative boot operation. Each type also has with-acknowledge and without-acknowledge modes.

In the boot operation mode, the master (MultiMediaCard host) can read boot data from the slave (MMC device) by keeping CMD line low after power-on, or sending CMD0 with argument + 0xFFFFFFFA (optional for slave), before issuing CMD1.

NOTE

For the eMMC4.3 card setting, please see the eMMC4.3 specification.

10.3.2.7.1 Boot operation

If the CMD line is held low for 74 clock cycles and more after power-up before the first command is issued, the slave recognizes that boot mode is being initiated and starts preparing boot data internally.

Within one second after the CMD line goes low, the slave starts to send the first boot data to the master on the DATA line(s). The master must keep the CMD line low to read all of the boot data.

NOTE

For the purposes of this documentation, fast boot is called "normal fast boot mode".

If boot acknowledge is enabled, the slave has to send acknowledge pattern '010' to the master within 50ms after the CMD line goes low. If boot acknowledge is disabled, the slave does not send out acknowledge pattern '010'.

The master can terminate the boot mode with the CMD line high.

The boot operation is terminated when all the contents of the enabled boot data are sent to the master. After the boot operation is executed, the slave gets ready for the CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.

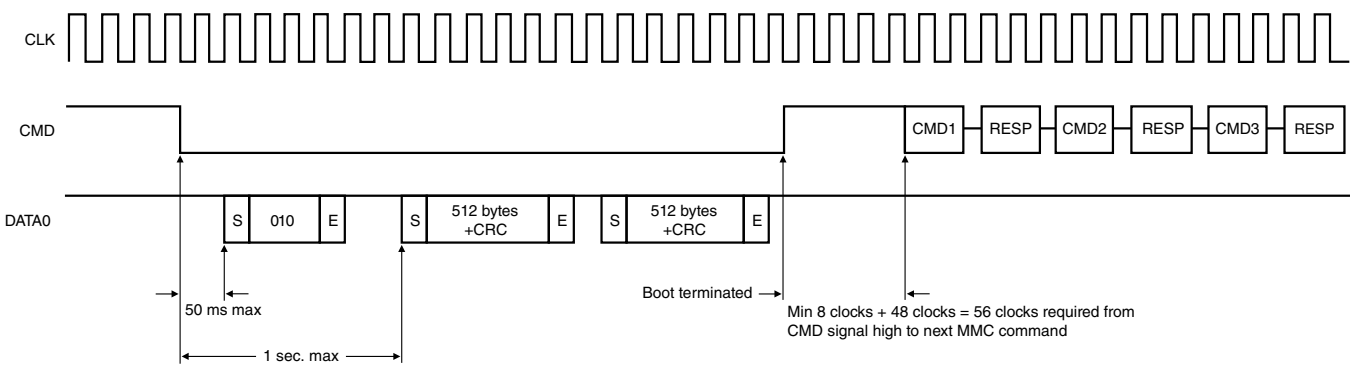


Figure 10-70. MultiMediaCard state diagram (normal boot mode)

10.3.2.7.2 Alternative boot operation

This boot function is optional for the device. If bit 0 in the extended CSD byte[228] is set to '1', the device supports the alternative boot operation.

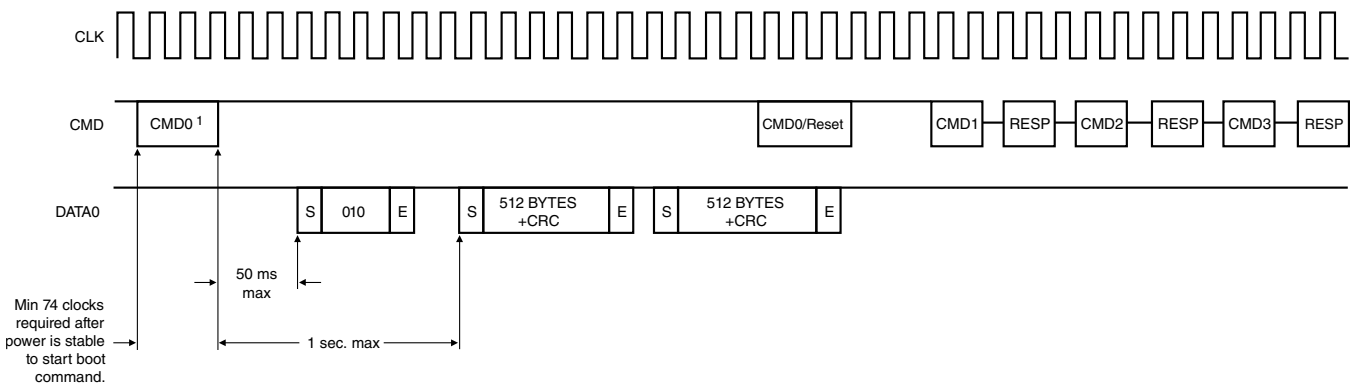
After power-up, if the host issues CMD0 with the argument of 0xFFFFFFFA after 74 clock cycles, before CMD1 is issued or the CMD line goes low, the slave recognizes that boot mode is being initiated and starts preparing boot data internally.

Within one second after CMD0 with the argument of 0xFFFFFFFA is issued, the slave starts to send the first boot data to the master on the DATA line(s).

If boot acknowledge is enabled, the slave must send the acknowledge pattern '010' to the master within 50ms after the CMD0 with the argument of 0xFFFFFFFA is received. If boot acknowledge is disabled, the slave does not send out acknowledge pattern '010'.

The master can terminate the boot mode by issuing CMD0 (Reset).

Boot operation is terminated when all the contents of the enabled boot data are sent to the master. After the boot operation is executed, the slave gets ready for the CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.



NOTE 1: CMD0 with argument 0xFFFFFFFA

Figure 10-71. MultiMediaCard state diagram (alternative boot mode)

10.3.2.8 Commands for MMC/SD/SDIO

A table containing the list of commands for the MMC/SD/SDIO cards is provided here.

Refer to the corresponding specifications for more details about the command information.

There are four kinds of commands defined to control the MultiMediaCard:

- Broadcast commands (bc), no response
- Broadcast commands with response (bcr), response from all cards simultaneously
- Addressed (point-to-point) commands (ac), no data transfer on the DATA
- Addressed (point-to-point) data transfer commands (adtc)

Response: A response is a token that is sent from the card to the host as an answer to a previously received command. A response is transferred serially on the CMD line.

Table 10-42. Commands for MMC/SD/SDIO cards

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all MMC and SD memory cards to idle state.
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all MMC and SD Memory cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.
CMD3[1]	ac	[31:6] RCA [15:0] stuff bits	R1 R6 (SDIO)	SET/ SEND_RELATIVE_ADDR	Assigns relative address to the card.
CMD4	bc	[31:0] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards.
CMD5	bc	[31:0] OCR without busy	R4	IO_SEND_OP_COND	Asks all SDIO cards in idle state to send them operation conditions register contents in the response on the CMD line.
CMD6[2]	adtc	[31] Mode 0: Check function 1: Switch function [30:8] Reserved for function groups 6 ~ 3 (All 0 or 0xFFFF) [7:4] Function group1 for command system [3:0] Function group2 for access mode	R1	SWITCH_FUNC	Checks switch ability (mode 0) and switch card function (mode 1). Refer to "SD Physical Specification V1.1" for more details.
CMD6[3]	ac	[31:26] Set to 0 [25:24] Access [23:16] Index [15:8] Value [7:3] Set to 0 [2:0] Cmd Set	R1b	SWITCH	Switches the mode of operation of the selected card or modifies the EXT_CSD registers. Refer to "The MultiMediaCard System Specification Version 4.0 Final draft 2" for more details.
CMD7	ac	[31:6] RCA [15:0] stuff bits	R1b	SELECT/ DESELECT_CARD	Toggles a card between the stand-by and transfer states or between the programming and disconnect states. In both cases, the card is selected by its own relative address and gets deselected by any other address. Address 0 deselects all.

Table continues on the next page...

Table 10-42. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD8	adtc	[31:0] stuff bits	R1	SEND_EXT_CSD	The card sends its EXT_CSD register as a block of data, with a block size of 512 bytes.
CMD9	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card-identification (CID) on the CMD line.
CMD11	adtc	[31:0] data address	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] stuff bits	R1b	STOP_TRANSMISSION	Forces the card to stop transmission.
CMD13	ac	[31:6] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:6] RCA [15:0] stuff bits	-	GO_INACTIVE_STATE	Sets the card to inactive state in order to protect the card stack against communication breakdowns.
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Sets the block length (in bytes) for all following block commands (read and write). Default block length is specified in the CSD.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a stop command.
CMD19	adtc	[31:0] reserved bits(all 0)	R1	SEND_TUNING_BLOCK	64 bytes tuning pattern is sent for SDR50 and SDR104.
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host, starting at the given address, until a STOP_TRANSMISION follows.
CMD21	adtc	[31:0] stuff bits	R1	SEND_TUNING_BLOCK	128 clocks of tuning pattern (64 byte in 4 bit mode or 128 byte in 8 bit mode) is sent for HS200 optimal sampling point detection.
CMD22-23	Reserved				
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.

Table continues on the next page...

Table 10-42. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command is issued only once per card. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
CMD31	Reserved				
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group.
CMD33	ac	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selection of a single sector to be selected for erase.
CMD34	ac	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	ac	[31:0] data address	R1	TAG_ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	ac	[31:0] data address	R1	TAG_ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase.
CMD37	ac	[31:0] data address	R1	UNTAG_ERASE_GROUP	Removes one previously selected erase group from the erase selection.
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erase all previously selected sectors.
CMD39	ac	[31:0] RCA [15] register write flag	R4	FAST_IO	Used to write and read 8-bit (register) data fields. The command addresses a card, and a register, and provides the data for writing if

Table continues on the next page...

Table 10-42. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
		[14:8] register address [7:0] register data			the write flag is set. The R4 response contains data read from the address register. This command accesses application dependent registers which are not defined in the MMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode.
CMD41	Reserved				
CDM42	adtc	[31:0] stuff bits	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43~51	Reserved				
CMD52	ac	[31:0] stuff bits	R5	IO_RW_DIRECT	Access a single register within the total 128k of register space in any I/O function.
CMD53	ac	[31:0] stuff bits	R5	IO_RW_EXTENDED	Accesses a multiple I/O register with a single command. Allows the reading or writing of a large number of I/O registers.
CMD54	Reserved				
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather than a standard command.
CMD56	adtc	[31:1] stuff bits [0]: RD/WR	R1b	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block is set by the SET_BLOCK_LEN command.
CMD57-59	Reserved				
CMD60	adtc	[31] WR [30:24] stuff bits [23:16] address [15:8] stuff bits [7:0] byte count	R1b	RW_MULTIPLE_REGISTER	These registers are used to control the behavior of the device and to retrieve status information regarding the operation of the device. All Status and Control registers are WORD (32-bit) in size and are WORD aligned. CMD60 is used to read and write these registers.
CMD61	adtc	[31] WR [30:16] stuff bits [15:0] data unit count	R1b	RW_MULTIPLE_BLOCK	The host issues a RW_MULTIPLE_BLOCK (CMD61) to begin the data transfer.
CMD62-63	Reserved				

Table continues on the next page...

Table 10-42. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
ACMD6[4]	ac	[31:2] stuff bits [1:0] bus width	R1	SET_BUS_WIDTH	Defines the data bus width ('00'=1bit or '10'=4bit bus) to be used for data transfer. The allowed data bus widths are given in SCR register.
ACMD13[4]	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD Memory Card status.
ACMD22[4]	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_SECTORS	Send the number of the written sectors (without errors). Responds with 32-bit plus the CRC data block.
ACMD23[4]	ac	[31:23] stuff bits [22:0] Number of blocks	R1	SET_WR_BLK_ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for fast Multiple Block WR command). "1"=default(one write block).
ACMD41[4]	bcr	[31:0] OCR	R3	SD_APP_OP_COND	Asks the accessed card to send its operating condition register (OCR) contents in the response on the CMD line.
ACMD42[4]	ac	[31:1] stuff bits [0] set_cd	R1	SET_CLR_CARD_DETECT	Connect(1)/Disconnect(0) the 50KOhm pull-up resistor on CD_B/DATA3 of the card.
ACMD51[4]	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

1. CMD3 differs for MMC and SD cards. For MMC cards, it is referred to as SET_RELATIVE_ADDR, with a response type of R1. For SD cards, it is referred to as SEND_RELATIVE_ADDR, with a response type of R6 (with RCA inside).
2. CMD6 differs completely between high speed MMC cards and high-speed SD cards. Command SWITCH_FUNC is for high-speed SD cards.
3. Command SWITCH is for high speed MMC cards. The Index field can contain any value from 0-255, but only values 0-191 are valid. If the Index value is in the 192-255 range the card does not perform any modification and the SWITCH_ERROR status bit in the EXT_CSD register is set. The Access Bits are shown in Table 2.
4. ACMDs is preceded with the APP_CMD command. Commands listed are used for SD only, other SD commands not listed are not supported on this module.

The access bits for the EXT_CSD access modes are listed in the following table.

Table 10-43. EXT_CSD access modes

Bits	Access name	Operation
00	Command set	The command set is changed according to the Cmd Set field of the argument.
01	Set bits	The bits in the pointed byte are set, according to the bits set to 1 in the Value field.
10	Clear bits	The bits in the pointed byte are cleared, according to the bits set to 1 in the Value field.
11	Write byte	The Value field is written into the pointed byte.

10.3.2.9 Clocks

The table found here describes the clock sources for uSDHC. For more information on clocking, see the Clocking chapter.

Table 10-44. Clocks

Clock name	Description
hclk	Bus clock
ipg_clk	Peripheral clock
ipg_clk_s	Peripheral access clock for register access
ipg_clk_perclk	Base clock

10.3.2.10 Clock and reset manager

This module controls all four kinds reset signals within uSDHC:

- Hardware reset
- Software reset for all logic
- Software reset for the data logic
- Software reset for the command logic

All these signals are fed into this module and stable signals are generated inside the module to reset all other modules. The module also gates off all the inside signals.

10.3.2.11 Clock generator

The clock generator generates the card CLK by peripheral source clock in two stages. The clock divisor can be configured through register SYS_CTRL [SDCLKFS] for prescaler configuration while the [DVS] for divisor configuration. Details can be found in the register function description. See the following figure for the structure of the divider. The term "Base" represents the frequency of peripheral source clock.

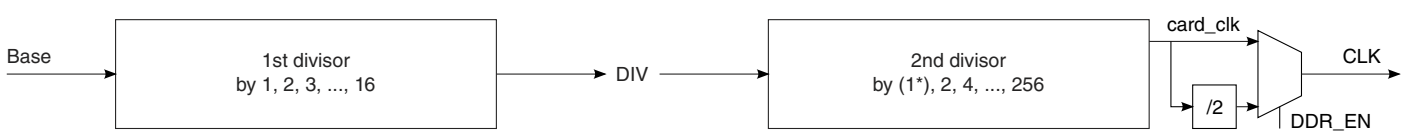


Figure 10-72. Two stages of the clock divider

The first stage outputs an intermediate clock (DIV) that can be Base, Base/2, Base/3, ..., or Base/16.

The second stage is a prescaler and outputs the actual internal working clock (card_clk). This clock is the driving clock for all the sub modules of the SD protocol unit, and helps in syncing FIFOs (see Figure 10-59) to synchronize with the data rate from the internal data buffer. The frequency of the clock output from this stage can be DIV, DIV/2, DIV/4,..., or DIV/256. Therefore, the highest frequency of the card_clk is base, and the next highest is Base/2, while the lowest frequency is Base/4096. If the duty cycle of the base clock is 50%, the duty cycle of card_clk is also 50%, even when the compound divisor is an odd value.

NOTE

CLK is different for the SDR and DDR modes.

- In the SDR mode, CLK is equal to the internal working clock (card_clk).

- In the DDR mode, CLK is equal to card_clk/2.

10.3.2.12 SDIO card interrupt

Information on interrupts in 1-bit mode, interrupts in 4-bit mode, and card interrupt handling are detailed in the sections below.

10.3.2.12.1 Interrupts in 1-bit mode

In this case, the DATA1 pin provides the interrupt function. An interrupt is asserted by pulling the DATA1 low from the SDIO card, until the interrupt service is finished to clear the interrupt.

10.3.2.12.2 Interrupt in 4-bit mode

As the interrupt and data line 1 share pin 8 in a 4-bit mode, an interrupt is only sent by the card and recognized by the host during a specific time. This is known as the interrupt period. The uSDHC module only provides samples the level on pin 8 during the interrupt period. At all other times, the host treats it as the data signal. The definition of the interrupt period is different for operations with single block and multiple block data transfers.

In the case of normal single data block transmissions, the interrupt period becomes active two clock cycles after the completion of a data packet. This interrupt period lasts until after the card receives the end bit of the next command that has a data block transfer associated with it.

For multiple block data transfers in a 4-bit mode, there is only a limited period of time that the interrupt period can be active because of the limited period of data line availability between the multiple blocks of data. This requires stricter definition of the interrupt period. For this case, the interrupt period is limited to two clock cycles. This begins two clocks after the end bit of the previous data block. During this 2-clock cycle interrupt period, if an interrupt is pending, the DATA1 line holds low for one clock cycle, then pulls high for the next clock cycle . On completion of the interrupt period, the card releases the DATA1 line into the high Z state. The uSDHC module provides sample of the DATA1 during the interrupt period when the IABG bit in the Protocol Control register is set.

Refer to SDIO Card Specification v1.10 for further information about the SDIO card interrupt.

10.3.2.12.3 Card interrupt handling

When the CINTIEN bit in the Interrupt Signal Enable Register is set to 0, uSDHC clears the interrupt request to the host system. The host driver should clear this bit before servicing the SDIO interrupt, and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.

The SDIO Card Interrupt Status can be cleared by writing 1 to this bit. But as the interrupt source from the SDIO card does not clear, this bit is set again. To clear this bit, it is required to reset the interrupt source from the external card followed by writing 1 to this bit. In a 1-bit mode, uSDHC detects the SDIO interrupt with or without the SD clock (to support wakeup). In a 4-bit mode, the interrupt signal is sampled during the interrupt period, so there are some sample delays between the interrupt signal from the SDIO card and the interrupt to the Host System Interrupt Controller. When the SDIO status is set and the host driver needs to service this interrupt, the SDIO bit in the Interrupt Control Register of SDIO card is cleared. This is required to clear the SDIO interrupt status latched in uSDHC and to stop driving the interrupt signal to the System Interrupt Controller. The host driver must issue a CMD52 to clear the card interrupt. After completion of the card interrupt service, the SDIO Interrupt Status Enable bit is set to 1 and uSDHC starts sampling the interrupt signal again.

See the figure below for an illustration of the SDIO card interrupt scheme and for the sequences of software and hardware events that take place during a card interrupt handling procedure.

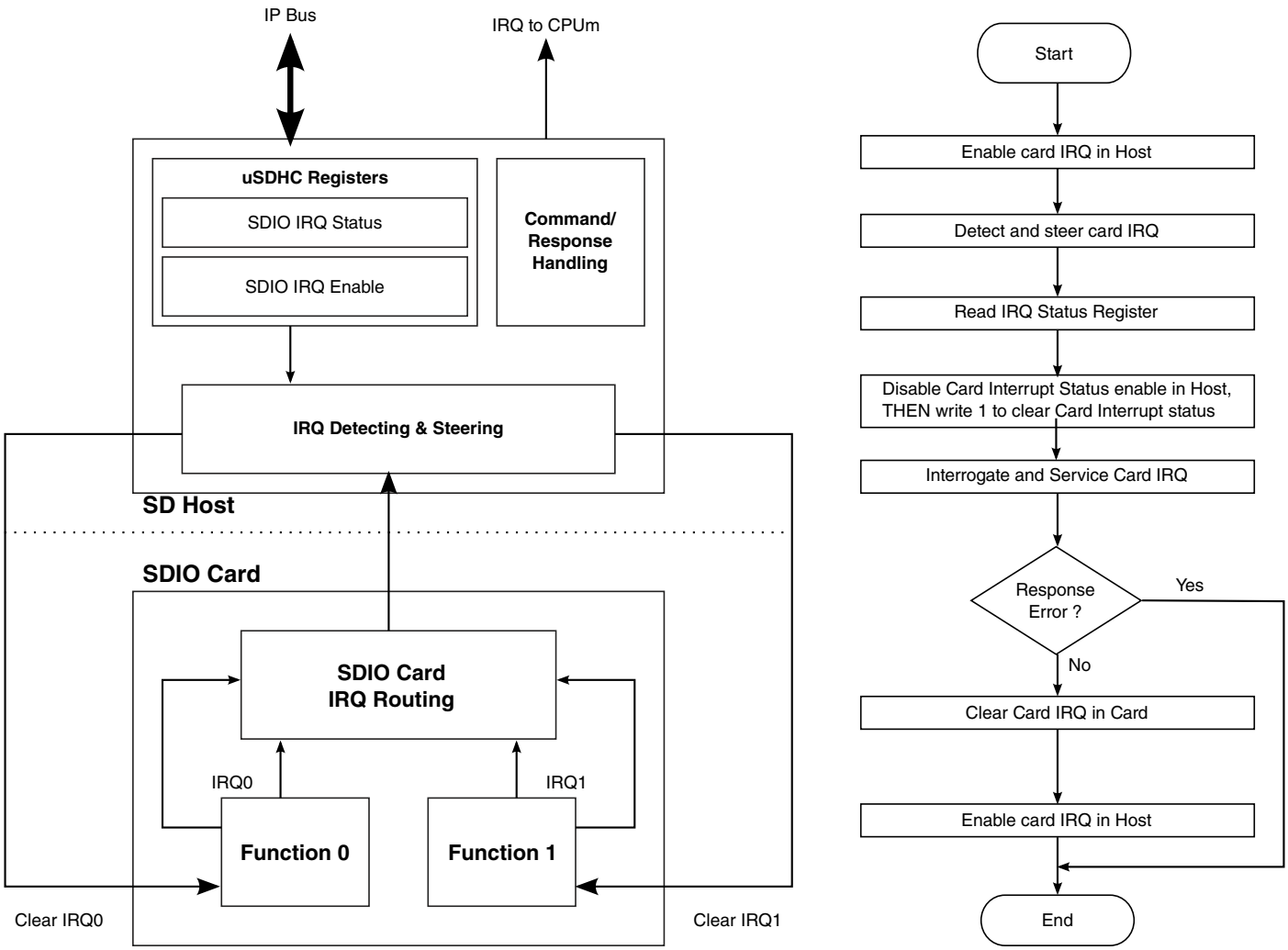


Figure 10-73. Card interrupt scheme, card interrupt detection, and handling procedure

10.3.3 External signals

The following table describes the uSDHC external signals:

Table 10-45. uSDHC external signals

Signal	Description	Direction
CLK	Clock for MMC/SD/SDIO card	O
CMD	CMD line connect to card	I/O
DATA7	DAT7 line in the 8-bit mode — Not used in other modes	I/O
DATA6	DAT6 line in the 8-bit mode — Not used in other modes	I/O

Table continues on the next page...

Table 10-45. uSDHC external signals (continued)

Signal	Description	Direction
DATA5	DAT5 line in the 8-bit mode — Not used in other modes	I/O
DATA4	DAT4 line in the 8-bit mode — Not used in other modes	I/O
DATA3	DAT3 line in the 4/8-bit mode or configured as card detection pin. The bit may be configured as card detection pin in the 1-bit mode.	I/O
DATA2	DAT2 line or Read Wait in the 4-bit mode Read Wait in 1-bit mode	I/O
DATA1	DAT1 line in the 4/8-bit mode Also, used to detect interrupt in 1/4-bit mode	I/O
DATA0	DAT0 line in all the modes Also, used to detect busy state	I/O
CD_B	Card detection pin If not used (for the embedded memory), tie low to indicate that there is a card attached.	I
WP	Card write protect detect If not used (for the embedded memory), tie low to indicate that it is not write protected.	I
RESET_B	Card hardware reset signal, active low	O
VSELECT	IO power voltage selection signal	O
STROBE	Input clock for eMMC HS400 mode	I

10.3.3.1 Signals overview

uSDHC has 15 associated I/O signals.

- The CLK is an internally generated clock used to drive the MMC, SD, and SDIO cards.
- The CMD I/O is used to send commands and receive responses to and from the card. Eight data lines (DAT7~DAT0) are used to perform data transfers between the uSDHC module and the card.
- The CD is card detection signal directly routed from the socket. This signal is active low (0). A low on CD# means that a card is inserted. It is optional for system implementation.
- The WP is write protection signals directly routed from the socket. This signal is active low (0). A high on WP means that the write protect switch is active. It is optional for system implementation.

- RST is an output signal used to reset the MMC card.
- VSELECT is an output signal used to change the voltage of the external power supplier. It is optional for system implementation.
- STROBE is input clock signal for eMMC HS400 mode.

If uSDHC needs to support a 4-bit data transfer, DAT7~DAT4 can also be optional and tied to high. If the uSDHC does not support the HS400 mode, STROBE can also be optional and tied to low.

10.3.4 Application of uSDHC

All communication between the system and cards are controlled by the host. The host sends commands of two types: broadcast and addressed (point-to-point).

Broadcast commands are intended for all cards, such as GO_IDLE_STATE, SEND_OP_COND, and ALL_SEND_CID. In the Broadcast mode, all cards are in the open-drain mode to avoid bus contention. Refer to Commands for MMC/SD/SDIO for the commands of bc and bcr categories.

After the broadcast command CMD3 is issued, the cards enter the standby mode. Addressed type commands are used from this point. In this mode, the CMD/DATA I/O pads turns to the push-pull mode to have the driving capability for maximum frequency operation. Refer to Commands for MMC/SD/SDIO for the commands of ac and adtc categories.

A uSDHC FIFO control problem exists with a software reset for a single block read. See MIX_CTRL[DTDSEL] for more information.

10.3.4.1 Command send & response receive basic operation

Assuming that the data type WORD is an unsigned 32-bit integer, the flow indicated below presents a guideline for sending a command to the card(s):

```
send_command(cmd_index, cmd_arg, other requirements)
{
WORD wCmd; // 32-bit integer to make up the data to write into Transfer Type register, it is
recommended to implement in a bit-field manner
wCmd = (<cmd_index> & 0x3f) << 24; // set the first 8 bits as '00'+<cmd_index>
set CMDTYP, DPSEL, CICEN, CCCEN, RSTTYP, DTDSEL accorind to the command index;
if (internal DMA is used) wCmd |= 0x1;
if (multi-block transfer) {
```

```
            set MSBSEL bit;
            if (finite block number) {
                        set BCEN bit;
                        if (auto12 command is to use) set AC12EN bit;
            }
}
write_reg(CMDARG, <cmd_arg>); // configure the command argument
write_reg(XFERTYP, wCmd); // set Transfer Type register as wCmd value to issue the command
}
wait_for_response(cmd_index)
{
while (CC bit in IRQ Status register is not set); // wait until Command Complete bit is set
read IRQ Status register and check if any error bits about Command are set
if (any error bits are set) report error;
write 1 to clear CC bit and all Command Error bits;
}
```

For the sake of simplicity, the function wait_for_response is implemented here by means of polling. For an effective and formal way, the response is usually checked after the Command Complete Interrupt is received. When doing this, make sure that the corresponding interrupt status bits are enabled.

For some scenarios, the response time-out is expected. For instance, after all cards respond to CMD3 and move to the Standby state no response to the Host when CMD2 is sent. The host driver deals with "fake" errors like this with caution.

10.3.4.2 Card identification mode

When a card is inserted to the socket or the card is reset by the host, the host needs to validate the operation voltage range, identify the cards, request the cards to publish the Relative Card Address (RCA) or to set the RCA for MMC cards.

10.3.4.2.1 Card detect

See the figure below for a flow diagram showing the detection of MMC, SD, and SDIO cards using uSDHC.

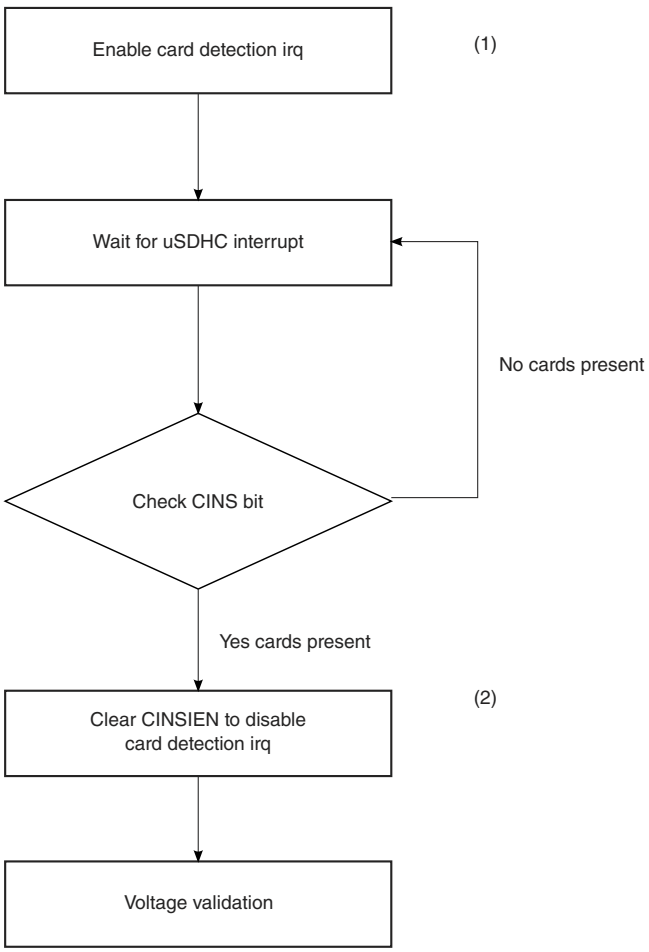


Figure 10-74. Flow diagram for card detection

Here is the card detect sequence:

- Set the CINSIEN bit to enable card detection interrupt.
- When an interrupt from uSDHC is received, check the CINS bit in the Interrupt Status register to see if it was caused by card insertion.
- Clear the CINSIEN bit to disable the card detection interrupt and ignore all card insertion interrupts afterwards.

10.3.4.2.2 Reset

The host consists of three types of resets:

- Hardware reset (Card and Host) that is driven by Power On Reset (POR).

- Software reset (Host only) is initiated by the write operation on the RSTD, RSTC, or RSTA bits of the System Control register to reset the data part, command part, or all parts of the host controller, respectively.
- Card reset (Card only): The command, "Go_Idle_State" (CMD0), is the software reset command for all types of MMC cards and SD memory cards. This command sets each card into the idle state regardless of the current card state. For an SDIO card, CMD52 is used to write an I/O reset in CCCR. The cards are initialized with a default relative card address (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

After the card is reset, the host needs to validate the voltage range of the card. See the figure below for the software flow to reset both uSDHC and the card.

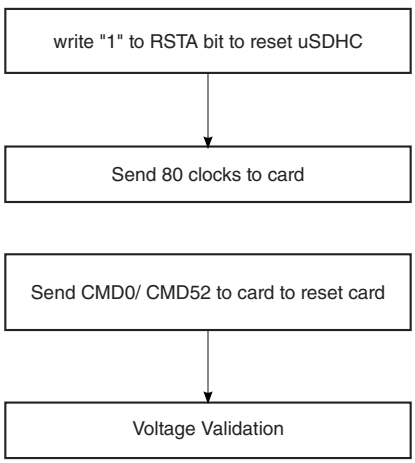


Figure 10-75. Flow chart for resetting uSDHC and SD I/O card

```
software_reset() { set_bit(SYSCTRL, RSTA); // software reset the Host set DTOCV and
      SDCLKFS bit fields to get the CLK of frequency around 400kHz configure IO pad to set
the power
      voltage of external card to around 3.0V poll bits CIHB and CDIHB bits of PRSSTAT to
wait both
      bits are cleared set_bit(SYSCTRL, INTIA); // send 80 clock ticks for card to power up
      send_command(CMD_GO_IDLE_STATE, <other parameters>); // reset the card with CMD0 or
      send_command(CMD_IO_RW_DIRECT, <other parameters>); }
```

10.3.4.2.3 Voltage validation

All cards should be able to establish communication with the host using any operation voltage in the maximum allowed voltage range specified in the card specification. However, the supported minimum and maximum values for VDD are defined in the Operation Conditions Register (OCR) and may not cover the whole range.

Cards that store the CID and CSD data in the preload memory are only able to communicate this information under data transfer VDD conditions. This means that if the host and card have non-common VDD ranges, the card is neither able to complete the identification cycle nor able to send CSD data.

Therefore, special commands Send_Op_Cont (CMD1 for MMC), SD_Send_Op_Cont (ACMD41 for SD Memory), and IO_Send_Op_Cont (CMD5 for SD I/O) are used. The voltage validation procedure is designed to provide a mechanism to identify and reject cards that do not match the VDD range(s) desired by the host. This is accomplished when the host sends the desired VDD voltage window as the operand of this command. Cards that cannot perform the data transfer in the specified range must discard themselves from further bus operations and go into the Inactive state. By omitting the voltage range in the command, the host can query each card and determine the common voltage range before sending out-of-range cards into the Inactive state. This query should be used if the host is able to select a common voltage range or if a notification is sent to the system when a non-usable card in the stack is detected.

The following steps show how to perform voltage validation when a card is inserted:

```
voltage_validation(voltage_range_arguement) { label the card as UNKNOWN;
      send_command(IO_SEND_OP_COND, 0x0, <other parameters are omitted>); // CMD5, check SDIO
      operation voltage, command argument is zero if (RESP_TIMEOUT !=
      wait_for_response(IO_SEND_OP_COND)) { // SDIO command is accepted if (0 < number of IO
      functions) { label the card as SDIO; IORDY = 0; while (!(IORDY in IO OCR response))
{ // set
      voltage range for each IO function send_command(IO_SEND_OP_COND, <voltage range>,
      <other parameter>); wait_for_response(IO_SEND_OP_COND); } // end of while ... } // end
      of if (0 < ... if (memory part is present inside SDIO card) Label the card as
SDCombo; //
      this is an SD-Combo card } // end of if (RESP_TIMEOUT ... if (the card is labelled as
SDIO
      card) return; // card type is identified and voltage range is set, so exit the
function;
      send_command(APP_CMD, 0x0, <other parameters are omitted>); // CMD55, Application
      specific CMD prefix if (no error calling wait_for_response(APP_CMD, <...>) { // CMD55
is
      accepted send_command(SD_APP_OP_COND, <voltage range>, <...>); // ACMD41, to set
      voltage range for memory part or SD card wait_for_response(SD_APP_OP_COND); // voltage
range
      is set if (card type is UNKNOWN) label the card as SD; return; // } // end of if (no
error ...
      else if (errors other than time-out occur) { // command/response pair is corrupted
deal with
      it by program specific manner; } // of else if (response time-out else { // CMD55 is
refuse,
      it must be MMC card if (card is already labelled as SDCombo) { // change label re-
label the
      card as SDIO; ignore the error or report it; return; // card is identified as SDIO
card } //
      of if (card is ... send_command(SEND_OP_COND, <voltage range>, <...>); if
      (RESP_TIMEOUT == wait_for_response(SEND_OP_COND)) { // CMD1 is not accepted, either
label the
      card as UNKNOWN; return; } // of if (RESP_TIMEOUT ... } // of else }
```

10.3.4.2.4 Card registry

Card registry for the MMC and SD/SDIO/SD combo cards are different. For the SD card, the identification process starts at a clock rate lower than 400 kHz and the power voltage higher than 2.7 V (as defined by the card spec). Currently, the CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated, the host requests the card to send their valid operation conditions.

The response to ACMD41 is the operation condition register of the card. The same command is sent to all the new cards in the system. Incompatible cards are put into the Inactive state. The host then issues the command, All_Send_CID (CMD2), to each card to get its unique card identification (CID) number. Cards that are currently unidentified (in the Ready state), send their CID number as the response. After the CID is sent by the card, the card goes into the Identification state.

The host then issues Send_Relative_Addr (CMD3), requesting the card to publish a new relative card address (RCA) that is shorter than the CID. This RCA is used to address the card for future data transfer operations. After the RCA is received, the card changes its state to the Standby state. At this point, if the host wants the card to have an alternative RCA number, it may ask the card to publish a new number by sending another Send_Relative_Addr command to the card. The last published RCA is the actual RCA of the card.

The host repeats the identification process with CMD2 and CMD3 for each card in the system until the last CMD2 gets no response from any of the cards in the system.

For MMC operation, the host starts the card identification process in the open-drain mode with the identification clock rate lower than 400 kHz and the power voltage higher than 2.7 V. The open drain driver stages on the CMD line to allow parallel card operation during card identification. After the bus is activated, the host requests the cards to send their valid operation conditions (CMD1). The response to CMD1 is the "wired AND " operation on the condition restrictions of all cards in the system. Incompatible cards are sent into the Inactive state. The host then issues the broadcast command All_Send_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards (the cards in the Ready state) simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods, stop sending their CID immediately and must wait for the next identification cycle. As the CID is unique for each card, only one card can be successfully sent its full CID to the host. This card then goes into the Identification state. Thereafter, the host issues Set_Relative_Addr (CMD3) to assign a relative card address (RCA) to the card. After the RCA is received, the state of the card changes to standby, and the card does not react in further identification cycles. Also, its output driver switches

from open-drain to push-pull. The host repeats the process, mainly CMD2 and CMD3, until the host receives a time-out condition to recognize the completion of the identification process.

The following steps show how to perform an operation using MMC cards:

```
card_registry() { do { // decide RCA for each card until response time-out
            if(card is labelled as SDCombo or SDIO) { // for SDIO card like device
            send_command(SET_RELATIVE_ADDR, 0x00, <...>); // ask SDIO card to publish its RCA
            retrieve RCA from response; } // end if (card is labelled as SDCombo ... else if
(card
            is labelled as SD) { // for SD card send_command(ALL_SEND_CID, <...>); if
            (RESP_TIMEOUT == wait_for_response(ALL_SEND_CID)) break;
send_command(SET_RELATIVE_ADDR,
            <...>); retrieve RCA from response; } // else if (card is labelled as SD ... else
            if (card is labelled as MMC) { send_command(ALL_SEND_CID, <...>); rca = 0x1; //
            arbitrarily set RCA, 1 here for example send_command(SET_RELATIVE_ADDR, 0x1 <<
16,
            <...>); // send RCA at upper 16 bits } // end of else if (card is labelled as MMC
            ... } while (response is not time-out); }
```

10.3.4.3 Card access

Information about Block Write, Block Read, Suspense Resume, ADMA Usage, Transfer Error, and Card Interrupt are detailed in the sections below.

10.3.4.3.1 Block write

Information on Normal Write, DDR Write, and Write with Pause are detailed in the sections below.

10.3.4.3.1.1 Normal Write

During a block write (CMD24 - 27, CMD60, CMD61), one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. If the CRC fails, the card indicates that the failure on the DATA line. The transferred data is discarded and not written, and all further transmitted blocks (in multiple block write mode) are ignored.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card detects the block misalignment error and aborts the programming before the beginning of the first misaligned block. The card sets the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer, waits in the Receive-data-State for a stop command. The write operation is also aborted if the host tries to write over a write-protected area.

For MMC and SD cards, programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card reports an error and does not change any register contents.

For all types of cards, some may require long and unpredictable periods of time to write a block of data. After receiving a block of data and completing the CRC check, the card begins writing and holds the DATA line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) or other means for SDIO cards at any time, and the card responds with its status. The responded status indicates whether the card can accept new data or whether the write process is still in progress. The host may deselect the card by issuing a CMD7 (to select a different card) to place the card into the Standby state and release the DATA line without interrupting the write operation. When re-selecting the card, it reactivates the busy indication by pulling data to low if the programming is still in progress and the write buffer is unavailable.

The software flow to write to a card incorporates the internal DMA and the write operation is a multi-block write with the Auto CMD12 enabled. For the other two methods (by means of external DMA orCPU polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to write to a card is described below:

1. Check the card status, wait until the card is ready for data.
2. Set the card block length/size:
   - For SD/MMC cards, use SET_BLOCKLEN (CMD16).
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT (CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred, or another error that occurred during the auto12 command sending and response receiving.

10.3.4.3.1.2 DDR write

uSDHC supports the dual data rate mode.

The software flow to write to a card in the DDR mode is described as below:

1. Check the card status and wait until the card is ready for data.

NOTE

For eMMC card, block length only can be set to 512byte.

2. Set the uSDHC number block register (NOB), where nob is 5, for instance.
3. Set the eMMC card to high-speed mode and use SWITCH(CMD6).
4. Set the eMMC card bus with 4-bit/8-bit DDR mode and use SWITCH(CMD6).
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. The DDR_EN and AC12EN bits should be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred or another error that occurred during the auto12 command sending and response receiving.

10.3.4.3.1.3 Write with Pause

The write operation can be paused during the transfer. Instead of stopping the CLK at any time to pause all the operations, which is also inaccessible to the host driver, the driver can set the Stop At Block Gap Request (SABGREQ) bit in the Protocol Control register to pause the transfer between the data blocks. As there is no time-out condition in a write operation during the data blocks, a write to all types of cards can be paused in this way, and if the DATA0 line is not required to de-assert to release the busy state, no suspend command is needed.

Similar to the flow described in Normal Write, the write with pause is shown with the same kind of write operation:

1. Check the card status and wait until the card is ready for data.
2. Set the card block length/size:
   - For SD/MMC, use SET_BLOCKLEN (CMD16).
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.

5. Disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Set the SABGREQ bit.
7. Wait for the Transfer Complete interrupt.
8. Clear the SABGREQ bit.
9. Check the status bit to see if a write CRC error occurred.
10. Set the CREQ bit to continue the write operation.
11. Wait for the Transfer Complete interrupt.
12. Check the status bit to see if a write CRC error occurred or some another error that occurred during the auto12 command sending and response receiving.

The number of blocks left during the data transfer is accessible by reading the contents of the BLKCNT field in the Block Attribute register. As the data transfer and the setting of the SABGREQ bit are concurrent, and the delay of register read and the register setting, the actual number of blocks left may not be exactly the value read earlier. The host driver reads the value of BLKCNT after the transfer is paused and the Transfer Complete interrupt is received.

It is also possible that the last block has begun when the Stop At Block Gap Request is sent to the buffer. In this case, the next block gap is the end of the transfer. These types of requests are ignored, the driver should treat these as a non-pause transfer, and deal with it as a common write operation.

When the write operation is paused, the data transfer inside the host system is not stopped, and the transfer is active until the data buffer is full. Because of this, it is recommended to avoid using the Suspend command for the SDIO card. This is because when such a command is sent, uSDHC interprets the system that switches to another function on the SDIO card and flush the data buffer. uSDHC takes the Resume command as a normal command with data transfer, and it is left for the host driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set as well as the AC12EN bit. However, the uSDHC module automatically sends a CMD12 to mark the end of the multi-block transfer.

10.3.4.3.2 Block read

Information about Normal read, DDR read, Read with Pause, and Delay Line (DLL) in Read Path are detailed in the sections below.

10.3.4.3.2.1 Normal read

For block reads, the basic unit of data transfer is a block whose maximum size is stored in areas defined by the corresponding card specification. A CRC is appended to the end of each block, ensuring data transfer integrity. The CMD17, CMD18, CMD53, CMD60, CMD61, and so on, can initiate a block read. After completing the transfer, the card returns to the Transfer state. For multi blocks read, data blocks are continuously transferred until a stop command is issued.

The software flow to read from a card incorporates the internal DMA and the read operation is a multi-block read with the Auto CMD12 enabled. For the other two methods (by means of external DMA orCPU polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to read from a card is described below:

1. Check the card status and wait until card is ready for data.
2. Set the card block length/size:
   - For SD/MMC, use SET_BLOCKLEN (CMD16).
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.
5. Disable the buffer read ready interrupt, configure the DMA settings, and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a read CRC error occurred, or another error occurred during the auto12 command sending and response receiving.

10.3.4.3.2.2 DDR read

The uSDHC module supports dual data rate mode.

The software flow to write to a card in the DDR mode is described below:

1. Check the card status and wait until the card is ready for data.

NOTE

For eMMC card, block length can be set to only 512 bytes.

2. Set the uSDHC number block register (NOB) where nob is 5, for instance.

3. Set the eMMC card to high-speed mode and use SWITCH (CMD6).
4. Set the eMMC card bus with 4-bit /8-bit DDR mode and use SWITCH(CMD6).
5. Disable the buffer write ready interrupt, configure the DMA settings, and enable the uSDHC DMA when sending the command with data transfer. The DDR_EN and AC12EN bits should be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred, or another error that occurred during the auto12 command sending and response receiving.

10.3.4.3.2.3 Read with Pause

The read operation is not generally able to pause. Only the SDIO card (and SDCombo card working under I/O mode) supporting the Read Wait feature can pause during the read operation. If the SDIO card supports Read Wait (SRW bit in CCCR register is 1), the host driver can set the SABGREQ bit in the Protocol Control register to pause the transfer between the data blocks.

Before setting the SABGREQ bit, ensure that the RWCTL bit in the Protocol Control register is set, otherwise uSDHC does not assert the Read Wait signal during the block gap and data corruption occurs. It is recommended to set the RWCTL bit after the Read Wait capability of the SDIO card is recognized.

Similar to the flow described in Normal read, the read with pause is shown with the same kind of read operation:

1. Check the SRW bit in the CCR register on the SDIO card to confirm that the card supports the Read Wait mode.
2. Set the RWCTL bit.
3. Check the card status and wait until the card is ready for data.
4. Set the card block length/size:
   - For SD/MMC, use SET_BLOCKLEN (CMD16).
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
5. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
6. Set the uSDHC number block register (NOB), where nob is 5, for instance.
7. Disable the buffer read ready interrupt, configure the DMA setting, and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
8. Set the SABGREQ bit.
9. Wait for the Transfer Complete interrupt.

10. Clear the SABGREQ bit.
11. Check the status bit to see if read CRC error occurred.
12. Set the CREQ bit to continue the read operation.
13. Wait for the Transfer Complete interrupt.
14. Check the status bit to see if a read CRC error occurred, or another error occurred during the auto12 command sending and response receiving.

Similar to the Write operation, it is possible to meet the ending block of the transfer when paused. In this case, uSDHC ignores the Stop At Block Gap Request and treats it as a command read operation.

Unlike the write operation, there is no remaining data inside the buffer when the transfer is paused. All data received before the pause is transferred to the host system. No matter if the Suspend command is sent or not, the internal data buffer is not flushed.

If the Suspend command is sent and the transfer is later resumed by means of a Resume command, uSDHC takes the command as a normal one accompanied with data transfer. It is left for the host driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set, as well as the AC12EN bit. However, the uSDHC automatically sends CMD12 to mark the end of multi-block transfer.

10.3.4.3.2.4 Delay Line (DLL) in read path

The DLL is newly added to assist in sampling read data. The DLL provides the ability to programmatically select a quantized delay (in fractions of the clock period) regardless of on-chip variations such as process, voltage, and temperature (PVT).

The reasons why DLL is needed for uSDHC are these:

- The path of read data traveling from card to host varies.
- In the SD/MMC DDR mode, the minimum input setup and hold time are both at 2.5 ns.

The data sampling window is so small that the delay of loopback clock needs to be accurate and consistent regardless of PVT. The DLL takes the divided card_clk as the reference clock and loopback clock as the input clock. It then generates a delayed version of the input clock according to the programmed target delay.

The DLL can be disabled or bypassed, and it can also be manually set for a fixed delay in the override mode. The override value set is the number of delay cells. In the override mode, there is no need to set the DLL_enable. Another DLL mode is target value mode. In this mode, the DLL automatically adjusts the number of delay cells according to the target value set by the user and PVT changes. Be aware that the target value is in units of

1/32 of the clock reference period. If the card_clk is 100Mhz, then the reference clock period is 10ns; setting target value of 16 means 5ns =(16/32)*10ns. The software can disable automatic update by the setting dll_gate_update bit.

As the user may change the frequency of card_clk from time to time by changing SDCLKFS[7:0]/DVS[3:0], the software must adjust the delay value to ensure it works correctly when the reference clock (card_clk) is changed. The following is the correct flow, which should be ignored if DLL_CTRL_ENABLE is not set.

Step 1: Set the DLL_CTRL_RESET bit

Step 2: Configure the SDCLKFS[7:0] and DVS[3:0]

Step 3: Wait until SDSTB is asserted

Step 4: Clear the DLL_CTRL_RESET bit

Step 5: Wait until both the DLL_STS_SLV_LOCK and DLL_STS_REF_LOCK are asserted

Step 6: Set the DLL_CTRL_SLV_FORCE_UPD

Step 7: Clear the DLL_CTRL_SLV_FORCE_UPD

NOTE

The software should make sure that the DLL_CTRL_SLV_FORCE_UPD lasts for at least one card_clk. So, the software may need to add some delay between step 6 and step 7.

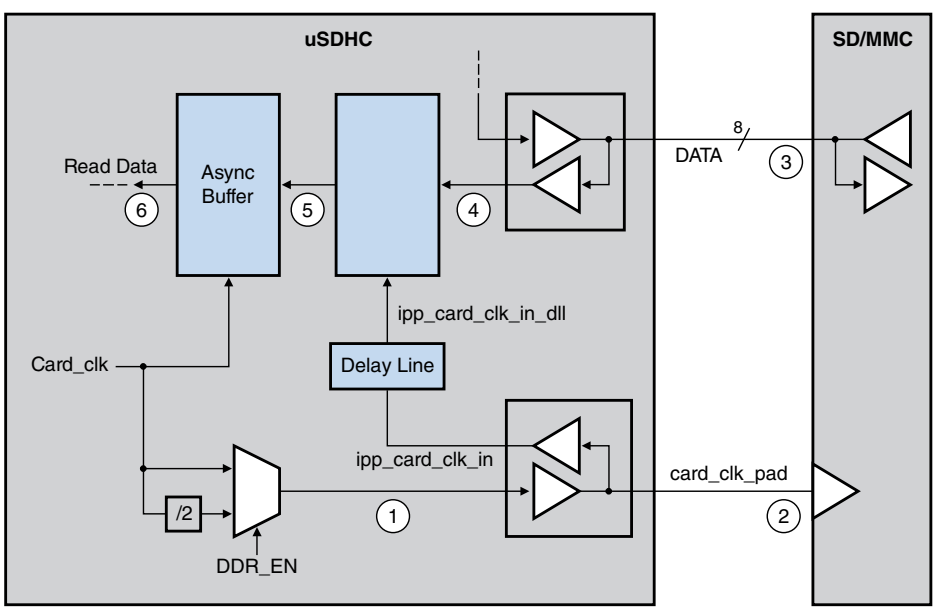


Figure 10-76. DLL in read path

10.3.4.3.3 Suspend Resume

The uSDHC module supports the Suspend Resume operations of SDIO cards, although slightly different than the suggested implementation of Suspend in the SDIO card specification.

10.3.4.3.3.1 Suspend

After setting the SABGREQ bit, the host driver may send a Suspend command to switch to another function of the SDIO card. The uSDHC module does not monitor the content of the response, so it does not know if the Suspend command succeeded or not. Accordingly, it does not de-assert Read Wait for read pause. To solve this problem, the host driver does not mark the Suspend command as "Suspend", (that is, setting the CMDTYP bits to 01). Instead, the driver sends this command as if it were a normal command, and only when the command succeeds, and the BS bit is set in the response, can the Driver send another command marked as "Suspend" to inform uSDHC that the current transfer is suspended. Here is the sequence for the Suspend operation:

1. Set the SABREQ bit to pause the current data transfer at block gap.
2. After the BGE bit is set, send the Suspend command to suspend the active function. The CMDTYP bit field must be 2'b00.
3. Check the BS bit of the CCCR in the response. If it is 1, repeat this step until the BS bit is cleared or abandon the suspend operation according to the Driver strategy.
4. Send another normal I/O command to the suspended function. The CMDTYP of this command must be 2'b01, so uSDHC can detect this special setting and be informed that the paused operation has successfully suspended. If the paused transfer is a read operation, uSDHC stops driving DATA2 and goes to the idle state.
5. Save the context registers in the system memory for later use, including the DMA System Address Register (for internal DMA operation), and the Block Attribute Register.
6. Begin operation for another function on the SDIO card.

10.3.4.3.3.2 Resume

To resume the data transfer, a Resume command is issued:

1. To resume the suspended function, restore the context register with the saved value in step #5 of the Suspend operation.
2. Send the Resume command: In the Transfer Type register, all the bit fields are set to the value as if this were another ordinary data transfer instead of a transfer resume (except the CMDTYP is set to 2'b10).

3. If the Resume command has responded, the data transfer is resumed.

10.3.4.3.4 ADMA usage

To use the ADMA in a data transfer, the host driver must prepare the correct descriptor chain prior to sending the read/write command.

The steps to prepare the correct descriptor chain are these:

1. Create a descriptor to set the data length that the current descriptor group is about to transfer. The data length should be even numbers of the block size.
2. Create another descriptor to transfer the data from the address setting in this descriptor. The data address must be at a page boundary (4KB address aligned).
3. If necessary, create a Link descriptor containing the address of the next descriptor. The descriptor group is created in steps 1 ~ 3.
4. Repeat steps 1 ~ 3 until all descriptors are created.
5. In the last descriptor, set the End flag to 1 and make sure the total length of all descriptors matches the product of the block size and block number configured in the Block Attribute Register.
6. Set the ADMA System Address Register to the address of the first descriptor and set the DMAS field in the Protocol Control Register to 01 to select the ADMA.
7. Issue a write or read command with the DMAEN bit set to 1 in the Transfer Type Register.

Steps 1 ~ 5 are independent of step 6, so step 6 can finish before steps 1 ~ 5. Regarding the descriptor configuration, it is recommended not to use the Link descriptor as it requires extra system memory access.

10.3.4.3.5 Transfer error

Information about CRC, Internal DMA, Transfer ADMA, and Auto CMD12 errors are detailed in the sections below.

10.3.4.3.5.1 CRC error

It is possible at the end of a block transfer that a write CRC status error or read CRC error occurs. For this type of error, the latest block received is discarded. This is because the integrity of the data block is not guaranteed. It is recommended to discard the following data blocks and re-transfer the block from the corrupted one.

For a multi-block transfer, the host driver issues a CMD12 to abort the current process and start the transfer by a new data command. In this scenario, even when the AC12EN and BCEND bits are set, uSDHC does not automatically send a CMD12 because the last

block is not transferred. On the other hand, if it is within the last block that the CRC error occurs, an Auto CMD12 is sent by uSDHC. In this case, the host driver re-sends or re-obtains the last block with a single block transfer.

10.3.4.3.5.2 Internal DMA error

During the data transfer with internal Simple DMA, if the DMA engine encounters an error on the AXI bus, the DMA operation is aborted, and the DMA error interrupt is sent to the host system. When acknowledged by such an interrupt, the host driver calculates the start address of data block in which the error occurs.

The start address can be calculated by either:

- Reading the DMA System Address register: The error occurs during the previous burst. Considering the block size, and the start address of the next burst transfer, it is straight forward to obtain the start address of the corrupted block.
- Reading the BLKCNT field of the Block Attribute register: Considering the number of blocks left, the total number to transfer, the start address of transfer, and the size of each block, the start address of corrupted block can be determined.To use this method, MIX_CTRL[BCEN] bit has to be set to enable Block Attribute register update.

When a DMA error occurs, it is recommended to abort the current transfer by means of a CMD12 (for multi block transfer), apply a reset for data, and restart the transfer from the corrupted block to recover from the error.

10.3.4.3.5.3 Transfer ADMA error

There are three kinds of possible ADMA errors: The AXI transfer, invalid descriptor, and data-length mismatch. Whenever these errors occur, the DMA transfer stops and the corresponding error status bit is set.

For acknowledging the status, the host driver should recover the error as shown below and re-transfer from the place of interruption.

- AXI transfer error: Such errors may occur during data transfer or descriptor fetch. For either scenario, it is recommended to retrieve the transfer context, reset for the data part and re-transfer the block that was corrupted, or to the next block if no block is corrupted.
- Invalid descriptor error: For such errors, it is recommended to retrieve the transfer context, reset for the data part and recreate the descriptor chain from the invalid descriptor and issue a new transfer. As the data to transfer now may be less than the

previous setting, the data length configured in the new descriptor chain should match the new value.
- Data-length mismatch error: It is similar to recover from this error. The Host Driver polls relating registers to retrieve the transfer context, apply a reset for the data part, configure a new descriptor chain and make another transfer if there is data left. Like the previous scenario of the invalid descriptor error, the data length must match the new transfer.

10.3.4.3.5.4 Auto CMD12 error

After the last block of the multi-block transfer is sent or received, and the AC12EN bit is set when the data transfer is initiated by the data command, uSDHC automatically sends a CMD12 to the card to stop the transfer.

When errors with this command occur, it is recommended to the host driver to deal with the situations in the following manner:

- Auto CMD12 response time-out: It is not certain whether the command is accepted by the card or not. The host driver clears the auto CMD12 error status bits and re-send CMD12 until it is accepted by the card.
- Auto CMD12 response CRC error: As the card responds to CMD12, it aborts the transfer. The host driver may ignore the error and clear the error status bit.
- Auto CMD12 conflict error or not sent: The command is not sent; therefore, the host driver sends a CMD12 manually.

10.3.4.3.6 Card interrupt

The external cards can inform the host controller by means of some special signals. For the SDIO card, it can be the low-level on the DATA1 line during some special period. The uSDHC module only monitors the DATA1 line and supports the SDIO interrupt.

When the SDIO interrupt is captured by uSDHC, and the host system is informed by uSDHC asserting the uSDHC interrupt line, the interrupt service from the host driver is called.

As the interrupt source is controlled by the external card, the interrupt from the SDIO card must be serviced before the CINT bit is cleared by writing 1. Refer to Card interrupt handling for the card interrupt handling flow.

10.3.4.4 Switch function

A switch command is issued by the host driver to enable new features added to the SD/MMC spec. The SD/MMC cards can transfer data at bus widths other than 1-bit. Different speed modes are also defined. To enable these features, a switch command is issued by the host driver.

For SDIO cards, the high-speed mode/DDR50/SDR50/SDR104 are enabled by writing the EHS bit in the CCCR register after the SHS bit is confirmed. For SD cards, the high-speed mode/DDR50/SDR50/SDR104 are queried and enabled by a CMD6 (with the mnemonic symbol as SWITCH_FUNC). For MMC cards, the high-speed mode/HS200/HS400 are queried by a CMD8 and enabled by a CMD6 (with the mnemonic symbol as SWITCH).

The SDR4-bit, SDR8-bit, DDR4-bit, and DDR8-bit width of MMC is also enabled by the SWITCH command, but with a different argument.

These new functions can also be disabled by a software reset. For SDIO cards, it can be done by setting the RES bit in the CCCR register. For other cards, it can be accomplished by issuing a CMD0. This method of restoring to the normal mode is not recommended because a complete identification process is needed before the card is ready for data transfer.

For the sake of simplicity, the following pseudocode examples do not show current capability check, which is recommended in the function switch process.

10.3.4.4.1 Query, enable, and disable SDIO high-speed mode

```
enable_sdio_high_speed_mode(void)
{
send CMD52 to query bit SHS at address 0x13;
if (SHS bit is '0') report the SDIO card does not support high speed mode and return;
send CMD52 to set bit EHS at address 0x13 and read after write to confirm EHS bit is set;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 50MHz;
(data transactions like normal peers)
}
disable_sdio_high_speed_mode(void)
{
send CMD52 to clear bit EHS at address 0x13 and read after write to confirm EHS bit is
cleared;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

10.3.4.4.2 Query, enable, and disable SD high-speed mode/DDR50/SDR50/SDR104

```
enable_sd_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
```

```
send CMD6, with argument 0xFFFFFx and read 64 bytes of data accompanying the R1 response;
(high speed mode,x=1; SDR50,x=2; SDR104,x=3; DDR50,x=4;)
wait data transfer done bit is set;
check if the bit x of received 512 bits is set;
if (bit 401 is '0') report the SD card does not support high speed mode and return;
if (bit 402 is '0') report the SD card does not support SDR50 mode and return;
if (bit 403 is '0') report the SD card does not support SDR104 mode and return;
if (bit 404 is '0') report the SD card does not support DDR50 mode and return;
send CMD6, with argument 0x80FFFFFx and read 64 bytes of data accompanying the R1 response;
(high speed mode,x=1; SDR50,x=2; SDR104 x=3; DDR50 x=4;)
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 50MHz for high speed mode, 100MHz for SDR50, 200MHz for SDR104, 50MHz for
DDR50;
(data transactions like normal peers)
}
disable_sd_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0x80FFFFF0 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

10.3.4.4.3 Query, enable, and disable MMC high-speed mode

```
enable_mmc_high_speed_mode(void)
{
send CMD9 to get CSD value of MMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support high speed mode and
return;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
extract the value of CARD_TYPE field to check the 'high speed mode' in this MMC is 26MHz or
52MHz;
send CMD6 with argument 0x1B90100;
send CMD13 to wait card ready (busy line released);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 1;
if (HS_TIMING is not 1) report MMC switching to high speed mode failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 26MHz or 52MHz according to the CARD_TYPE;
(data transactions like normal peers)
}
disable_mmc_high_speed_mode(void)
{
send CMD6 with argument 0x2B90100;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 0;
if (HS_TIMING is not 0) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 20MHz;
(data transactions like normal peers)
}
```

10.3.4.4.4 Set MMC bus width

```
change_mmc_bus_width(void)
{
send CMD9 to get CSD value of MMC;
```

```
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support multiple bit width and
return;
send CMD6 with argument 0x3B70x00; (8-bit(dual data rate), x=6; 4-bit(dual data rate), x=5;8-
bit, x=2; 4-bit, x=1; 1-bit, x=0)
send CMD13 to wait card ready (busy line released);
(data transactions like normal peers)
}
```

10.3.4.5 ADMA operation

Here are the codes for the ADMA1 and ADMA2 operations.

10.3.4.5.1 ADMA1 operation

```
Set_adma1_descriptor
{
if (to start data transfer) {
// Make sure the address is 4KB align.
Set 'Set' type descriptor;
{
Set Act bits to 01;
Set [31:12] bits data length (byte unit);
}
Set 'Tran' type descriptor;
{
Set Act bits to 10;
Set [31:12] bits address (4KB align);
}
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to 11;
Set [31:12] bits the next descriptor address (4KB aligned);
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set End bit to 1;
}
if (to generate interrupt for this descriptor) {
Set Int bit to 1;
}
Set Valid bit to 1;
}
```

10.3.4.5.2 ADMA2 operation

```
Set_adma2_descriptor
{
if (to start data transfer) {
// Make sure the address is a 32-bit boundary (lower 2-bit are always '00').
Set higher 32-bit of descriptor for this data transfer initial address;
Set [31:16] bits data length (byte unit);
Set Act bits to '10';
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to '11';
// Make sure the address is 32-bit boundary (lower 2-bit are always set to '00').
Set higher 32-bit of descriptor for the next descriptor address;
}
else { // other types of descriptor
```

```
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set 'End' bit '1';
}
if (to generate interrupt for this descriptor) {
Set 'Int' bit '1';
}
Set the 'Valid' bit to '1';
}
```

10.3.4.6 Fast boot operation

10.3.4.6.1 Normal fast boot flow

Here are the steps for normal fast boot flow:

1. Software must configure the SYS_CTRL[INITA] bit to make sure that 74 card clocks are finished.
2. Software must configure the MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 0 (normal fast boot), and bit 4 to select the ack mode. If the data is sent through the DMA mode, the software should configure bit 7 to enable the automatic stop at block gap feature, and configure bit 3-bit 0 to select the ack timeout value according to the SD CLK frequency.
3. Software then needs to configure the Block Attributes Register to set the block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes.
4. Software must configure the Protocol control register to set Data Transfer Width (DTW). If in the DDR fast boot mode, DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software needs to configure the Command Argument Register to set argument if needed (no need in normal fast boot).
6. Software must configure the Transfer Type Register to start the boot process. In normal boot mode, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN retain the default value, where DPSEL bit is set to 1, DTDSEL is set to 1 and MSBSEL is set to 1.
7. DMAEN should be configured as 0 in the polling mode and if BCEN is configured as 1, it is recommended to configure the number of blocks in the Block Attributes Register to the maximum value. If in DDR fast boot mode, DDR_EN needs to be set to 1.
8. When step 6 is configured, the boot process begins. The software needs to poll the data buffer ready status to read the data from the buffer in time. If a boot timeout happens (ack times out or the first data read times out), an interrupt is triggered, and the software must configure MMC Boot Register to bit 6 to 0 to disable boot. This

makes CMD high, then after at least 56 clocks, it is ready to begin a normal initialization process.

9. If there is no timeout, software needs to determine when the data read is finished and then configure MMC Boot Register bit 6 to 0 to disable boot. This render CMD line high and command completed asserted. After at least 56 clocks, it is ready to begin the normal initialization process.
10. You must reset the host and begin the normal process.

10.3.4.6.2 Alternative fast boot flow

Here are the steps for alternative fast boot flow:

1. Software needs to configure SYS_CTRL[INITA] to make sure 74 card clocks are finished.
2. Software needs to configure MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 1 (alternative boot), and bit 4 to select the ack mode or not. If data needs to be sent through the DMA mode, then configure bit 7 to enable the automatic stop at block gap feature. Software should also configure bit 3-bit 0 to select the ack timeout value according to the SD clock frequency.
3. Software then needs to configure the Block Attributes Register to set the block size and count. If in the DDR fast boot mode, the block size only can be configured to 512 bytes.
4. Software needs to configure the Protocol control register to set the data transfer width (DTW). If in the DDR fast boot mode, DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software needs to configure Command Argument Register to set argument to 0xFFFFFFFA.
6. Software needs to configure the Transfer Type Register to start the boot process by CMD0 with the 0xFFFFFFFA argument. In alternative boot, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN, and DMAEN retain the default value. DPSEL bit is set to 1, DTDSEL is set to 1, and MSBSEL is set to 1. Note DMAEN should be configured as 0 in the polling mode, and if BCEN is configured as 1 in polling mode, it is recommended to configure the block count in the Block Attributes Register to the maximum value. If in the DDR fast boot mode, DDR_EN needs to be set to 1.
7. When step 6 is configured, the boot process begins. Software needs to poll the data buffer ready status to read the data from the buffer in time. If there is a boot timeout (ack data timeout in 50ms or data timeout in 1s), the host sends out the interrupt and the software needs to send CMD0 with reset and then configure the boot enable bit to 0 to stop this process.
8. If there is no time out, the software needs to decide when to stop the boot process, and send out the CMD0 with reset and then after the command is completed,

configure the MMC Boot Register bit 6 to stop the process. After 8 clocks from the command completion, the slave (card) is ready for the identification step.
9. You must reset the host and begin the normal process.

10.3.4.6.3 Fast boot application case (in DMA mode)

In the boot application case, because the image destination and the image size are contained in the beginning of the image, it is necessary to switch DMA parameters on the fly during MMC fast boot.

In fast boot, the host can use Advanced DMA2 (ADMA2) with two destinations.

The detailed flow is described below:

1. The software needs to configure INIT_ACTIVE bit (system control register bit 27) to make sure that 74 card clocks are finished.
2. The software needs to configure the MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot); and bit 5 to 0 (normal fast boot) or 1 (alternative boot); and bit 4 to select the ack mode. In DMA mode, configure bit 7 to 1 to enable the automatic stop at block gap feature. Also configure bits[31-16] to set the (BLK_CNT - VALUE1). Here VALUE1 is the value of the block count that needs to transfer the first time, so that the host stops at the block gap when the uSDHC controller gets VAULE1 blocks from the device. Also, configure bits[3-0] to select the ack timeout value according to the SD clock frequency.
3. The software then needs to configure the Block Attributes Register to set block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes. In DMA mode, it is recommended to set the block count (BLK_CNT) to the max value (16'hffff).
4. The software needs to configure Protocol Control Register to set DTW (data transfer width). If in DDR fast boot mode, the DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software enable ADMA2 by configuring Protocol Control Register bits [9-8].
6. The software needs to set at least three pairs of ADMA2 descriptor in boot memory (that is, in IRAM, at least six words). The first pair descriptor defines the start address (that is, IRAM) and data length (that is,512byte*VALUE1) of the first part boot code. The software also needs to set the second pair descriptor, the second start address (any value that is writable), and data length is suggested to set 1~2word (record as VALUE2). Note that the second couple desc also transfers useful data even at lease 1 word, because our ADMA2 cannot support 0 data_length data transfer descriptor.
7. The software needs to configure Command Argument Register to set argument to 0xFFFFFFFA in alternative fast boot and do not need to be set in normal fast boot.

8. The software needs to configure Transfer Type Register to start the boot process. CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN, and DMAEN retain the default value. DPSEL bit is set to 1, DTDSEL is set to 1, and MSBSEL is set to 1. DMAEN is configured as 1 in the DMA mode. And, if BCEN is configured as 1, then configure blk no in Bock Attributes Register to the max value. And, if in the DDR fast boot mode, DDR_EN needs to be set to 1.
9. When step 8 is configured, boot process begins, the first VALUE1 block number data gets transferred. The software needs to poll the TC bit (bit1 in Interrupt Status Register) to determine first transfer is ended. Also, the software needs to polling the BGE bit (bit2 in Interrupt Status Register) to determine if the first transfer stops at the block gap.
10. When TC and BGE bits are set to 1, the software can analyze the first code of VALUE1 block, initializes the new memory device, if required, and sets the third pair of descriptors to define the start address and length of the remaining part of the boot code (VALUE3, the remain boot code block). Remember to set the last descriptor with END.
11. The software needs to configure the MMC Boot Register (offset 0xc4) again. Set bit 6 to 1 (enable boot); and bit 5 to 0 (normal fast boot), to 1 (alternative boot); and bit 4 to select the ack mode or not. In the DMA mode, configure bit 7 to 1 for enabling the automatically stop at block gap feature. Also, configure bit31-bit16 to set the (BLK_CNT - (VALUE1+1+VALUE3)), that host stops at block gap when the uSDHC controller gets (VALUE1+1+VALUE3)) blocks from device totally include the blocks received in step 9. And need to configure bit 3-bit0 to select the ack timeout value according to the sd clk frequency. Note that the software does not need to configure the BLK_CNT again, because it is counted down automatically by the uSDHC controller.
12. The software needs to clear the TC and BGE bits and the software needs to clear SABGREQ (bit 16 in the Protocol control register) and set CREQ (bit17 in the Protocol control register) to 1 to resume the data transfer. Host transfers the VALUE2 and VALUE3 data to the destination that is set by descriptor.
13. The software needs to do poll BGE bit to determine if the fast boot is over.

Note:

- When ADMA boot flow starts, for uSDHC, it is like a normal ADMA read operation. So, set ADMA2 descriptor as the normal ADMA2 transfer.
- Need a few words length memory to keep descriptor.
- For the 1~2-word data in second descriptor setting, it is a useful data, so the software needs to deal the data because of the application case.

10.3.5 Software restrictions

10.3.5.1 Initialization active

The driver cannot set INITA bit in System Control register when any of the command line or data lines are active, so the driver must ensure both CDIHB and CIHB bits are cleared.

10.3.5.2 Software polling procedure

For polling read or write, after the software begins a buffer read or write, it must access exactly the number of times as the values set in the Watermark Level Register; moreover, if the block size is not a multiple of the value in the Watermark Level Register (read and write respectively), the software must access exactly the remaining number of words at the end of each block.

For example, for a read operation, if the RD_WML is 4, indicating the watermark level is 16 bytes, block size is 40 bytes, and the block number is 2, then the access times for the burst sequence in the whole transfer process must be 4, 4, 2, 4, 4, 2.

10.3.5.3 Suspend operation

To suspend the data transfer, the software must inform uSDHC that the suspend command is successfully accepted. To achieve this, after the Suspend command is accepted by the SDIO card, software must send another normal command marked as suspend command (CMDTYP bits set as '01') to inform uSDHC that the transfer is suspended.

If software needs to resume the suspended transfer, it should read the value in BLKCNT register to save the remaining number of blocks before sending the normal command marked as suspend, otherwise on sending such a 'suspend' command, uSDHC treats the current transfer is aborted and change the BLKCNT register to its original value, instead of retaining the remaining number of blocks.

10.3.5.4 Data length setting

For either ADMA (ADMA1 or ADMA2) transfer, the data in the data buffer must be word aligned, so the data length set in the descriptor must be a multiple of 4.

ADMA1 is 4KB aligned.

10.3.5.5 (A)DMA address setting

To configure the ADMA1/ADMA2/DMA address register, when TC bit is set, the register always updates itself with the internal address value to support dynamic address synchronization, so the software must ensure that the TC bit is cleared prior to configuring the ADMA1/ADMA2/DMA address register.

10.3.5.6 Data port access

Data port does not support parallel access. For example, during an internal DMA access, it is not allowed to write any data to the data port by CPU; or during a CPU read operation, it is also prohibited to write any data to the data port, by either CPU or internal DMA. Otherwise the data is corrupted inside the uSDHC buffer.

10.3.5.7 Change clock frequency

The uSDHC module does not automatically gate off the card clock when the host driver changes the clock frequency. To prevent possible glitch on the card clock, clear the FRC_SDCLK_ON bit when changing the clock divisor value (SDCLKFS or DVS in System Control Register) or setting the RSTA bit.

Also, before changing the clock divisor value, the host driver should make sure that the SDSTB bit is high.

10.3.5.8 Multi-block read

For pre-defined multi-block read operation, that is, the number of blocks to read has been defined by previous CMD23 for MMC, or pre-defined number of blocks in CMD53 for SDIO/SDCombo, or whatever multi-block read without abort command at card side, an abort command, either automatic or manual CMD12/CMD52, is still required by uSDHC after the pre-defined number of blocks are done, to drive the internal state machine to idle mode.

In this case, the card may not respond to this extra abort command and uSDHC gets response timeout. It is recommended to manually send an abort command with RSPTYP[1:0] both bits cleared.

10.3.6 uSDHC memory map/register definition

10.3.6.1 uSDHC register descriptions

This section includes the module memory map and detailed descriptions of all registers.

See the table below for the register memory map of uSDHC. All these registers only support 32-bit accesses.

NOTE

The uSDHC registers are 32-bit wide and only support 32-bit access.

10.3.6.1.1 uSDHC memory map

USDHC1 base address: 30B4_0000h

USDHC2 base address: 30B5_0000h

USDHC3 base address: 30B6_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	DMA System Address (DS_ADDR)	32	RW	0000_0000h
4h	Block Attributes (BLK_ATT)	32	RW	0001_0000h
8h	Command Argument (CMD_ARG)	32	RW	0000_0000h
Ch	Command Transfer Type (CMD_XFR_TYP)	32	RW	0000_0000h
10h	Command Response0 (CMD_RSP0)	32	RO	0000_0000h
14h	Command Response1 (CMD_RSP1)	32	RO	0000_0000h
18h	Command Response2 (CMD_RSP2)	32	RO	0000_0000h
1Ch	Command Response3 (CMD_RSP3)	32	RO	0000_0000h
20h	Data Buffer Access Port (DATA_BUFF_ACC_PORT)	32	RW	0000_0000h
24h	Present State (PRES_STATE)	32	RO	Table 10-92
28h	Protocol Control (PROT_CTRL)	32	RW	0880_0020h
2Ch	System Control (SYS_CTRL)	32	RW	0080_800Fh
30h	Interrupt Status (INT_STATUS)	32	W1C	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
34h	Interrupt Status Enable (INT_STATUS_EN)	32	RW	0000_0000h
38h	Interrupt Signal Enable (INT_SIGNAL_EN)	32	RW	0000_0000h
3Ch	Auto CMD12 Error Status (AUTOCMD12_ERR_STATUS)	32	RW	0000_0000h
40h	Host Controller Capabilities (HOST_CTRL_CAP)	32	RW	07F3_B407h
44h	Watermark Level (WTMK_LVL)	32	RW	0810_0810h
48h	Mixer Control (MIX_CTRL)	32	RW	8000_0000h
50h	Force Event (FORCE_EVENT)	32	WORZ	0000_0000h
54h	ADMA Error Status (ADMA_ERR_STATUS)	32	RO	0000_0000h
58h	ADMA System Address (ADMA_SYS_ADDR)	32	RW	0000_0000h
60h	DLL (Delay Line) Control (DLL_CTRL)	32	RW	0000_0000h
64h	DLL Status (DLL_STATUS)	32	RO	0000_0200h
68h	CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS)	32	RW	0000_0000h
70h	Strobe DLL control (STROBE_DLL_CTRL)	32	RW	0000_0000h
74h	Strobe DLL status (STROBE_DLL_STATUS)	32	RO	0000_0200h
C0h	Vendor Specific Register (VEND_SPEC)	32	RW	3000_7809h
C4h	MMC Boot (MMC_BOOT)	32	RW	0000_0000h
C8h	Vendor Specific 2 Register (VEND_SPEC2)	32	RW	0001_9006h
CCh	Tuning Control (TUNING_CTRL)	32	RW	0021_2800h
100h	Command Queuing Version (CQVER)	32	RO	0000_0510h
104h	Command Queuing Capabilities (CQCAP)	32	RO	0000_310Ah
108h	Command Queuing Configuration (CQCFG)	32	RW	0000_0000h
10Ch	Command Queuing Control (CQCTL)	32	RW	0000_0000h
110h	Command Queuing Interrupt Status (CQIS)	32	W1C	0000_0000h
114h	Command Queuing Interrupt Status Enable (CQISTE)	32	RW	0000_0000h
118h	Command Queuing Interrupt Signal Enable (CQISGE)	32	RW	0000_0000h
11Ch	Command Queuing Interrupt Coalescing (CQIC)	32	RW	0000_0000h
120h	Command Queuing Task Descriptor List Base Address (CQTDLBA)	32	RW	0000_0000h
124h	Command Queuing Task Descriptor List Base Address Upper 32 Bits (CQTDLBAU)	32	RW	0000_0000h
128h	Command Queuing Task Doorbell (CQTDBR)	32	RW	0000_0000h
12Ch	Command Queuing Task Completion Notification (CQTCN)	32	W1C	0000_0000h
130h	Command Queuing Device Queue Status (CQDQS)	32	RO	0000_0000h
134h	Command Queuing Device Pending Tasks (CQDPT)	32	RO	0000_0000h
138h	Command Queuing Task Clear (CQTCLR)	32	RW	0000_0000h
140h	Command Queuing Send Status Configuration 1 (CQSSC1)	32	RW	0001_1000h
144h	Command Queuing Send Status Configuration 2 (CQSSC2)	32	RW	0000_0000h
148h	Command Queuing Command Response for Direct-Command Task (CQCRDCT)	32	RO	0000_0000h
150h	Command Queuing Response Mode Error Mask (CQRMEM)	32	RW	FDF9_A080h
154h	Command Queuing Task Error Information (CQTERRI)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
158h	Command Queuing Command Response Index (CQCRI)	32	RO	0000_0000h
15Ch	Command Queuing Command Response Argument (CQCRA)	32	RO	0000_0000h

10.3.6.1.2 DMA System Address (DS_ADDR)

10.3.6.1.2.1 Offset

Register	Offset
DS_ADDR	0h

10.3.6.1.2.2 Function

This register contains the physical system memory address used for DMA transfers.

10.3.6.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DS_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DS_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.2.4 Fields

Field	Function
31-0 DS_ADDR	System address **DMA system address / argument 2** When ACMD23_ARGU2_EN is set to 0, SDMA uses this register as system address and supports only 32-bit addressing mode. Auto CMD23 cannot be used with SDMA. When ACMD23_ARGU2_EN is set to 1, SDMA uses ADMA System Address register (05Fh – 058h) instead of this register to support both 32-bit and 64-bit addressing. This register is used only for Argument2 and SDMA may use Auto CMD23. 1. **SDMA system address**

Field	Function
	Because the address must be word (4 bytes) aligned, the least two bits are reserved and always set to 0. When uSDHC stops a DMA transfer, this register points out the system address of the next contiguous data position. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operation during transfers may return an invalid value. The host driver initializes this register before starting a DMA transaction. After DMA has stopped, the system address of the next contiguous data position can be read from this register. This register is protected during a data transfer. When data lines are active, write to this register is ignored. The host driver waits until the DLA field in the Present State register is cleared, before writing to this register. The uSDHC internal DMA does not support a virtual memory system. It only supports continuous physical memory access. Because this register supports dynamic address reflecting, when TC field is set, it automatically alters the value of internal address counter, so the software cannot change this register when TC field is set. Such restriction is also listed in Software restrictions. 2. **Argument 2** This register is used with the Auto CMD23 to set a 32-bit block count value to the argument of the CMD23 while executing Auto CMD23. If Auto CMD23 is used with ADMA, the full 32-bit block count value can be used. If Auto CMD23 is used without ADMA, the available block count value is limited by the Block Count register. 65535 blocks is the maximum value in this case.

10.3.6.1.3 Block Attributes (BLK_ATT)

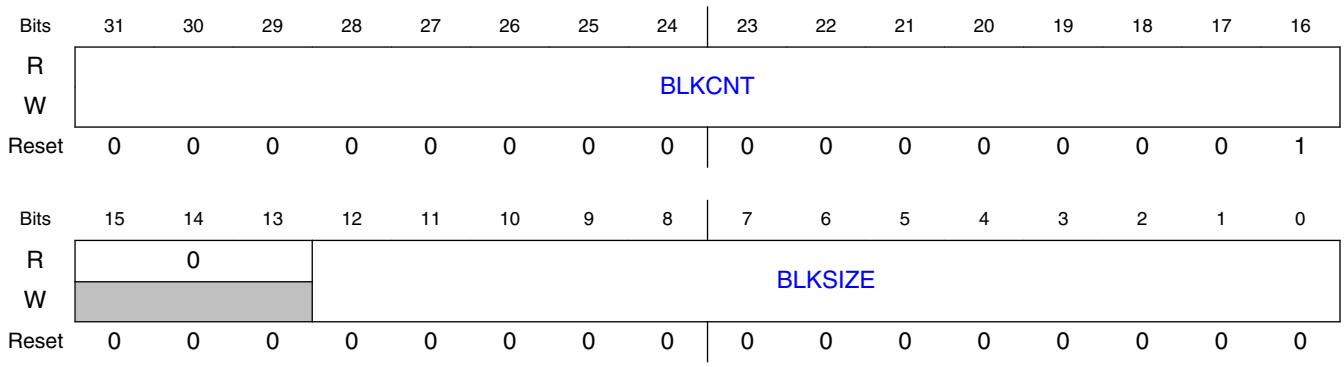
10.3.6.1.3.1 Offset

Register	Offset
BLK_ATT	4h

10.3.6.1.3.2 Function

This register is used to configure the number of data blocks and the number of bytes in each block.

10.3.6.1.3.3 Diagram



10.3.6.1.3.4 Fields

Field	Function
31-16 BLKCNT	Blocks count for current transfer This field is enabled when the Block Count Enable field in the Transfer Mode register is set to 1 and is valid only for multiple block transfers. For single block transfer, this field always reads as 1. The host driver sets this field to a value between 1 and the maximum block count. The uSDHC module decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to zero results in no data blocks being transferred. This field should be accessed only when no transaction is executing (that is, after transactions are stopped). During data transfer, read operations on this field may return an invalid value and write operations are ignored. When saving transfer content because of a Suspend command, the number of blocks yet to be transferred can be determined by reading this field. The reading of this field should be applied after transfer is paused by stop at block gap operation and before sending the command marked as suspend. This is because when the Suspend command is sent out, uSDHC treats the current transfer as aborted and change the BLKCNT field back to its original value instead of keeping the dynamical indicator of the remaining block count. When restoring transfer content prior to issuing a Resume command, the host driver restores the previously saved block count. **NOTE:** Although the BLKCNT field is 0 after reset, the read of reset value is 0x1. This is because when MSBSEL field is indicating a single block transfer, the read value of BLKCNT is always 1. 0000000000000000b - Stop count 0000000000000001b - 1 block 0000000000000010b - 2 blocks 1111111111111111b - 65535 blocks
15-13 —	Reserved
12-0 BLKSIZE	Transfer block size This field specifies the block size for block data transfers. Values ranging from 1 byte up to the maximum buffer size can be set. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations are ignored. 0000000000000b - No data transfer 0000000000001b - 1 byte

Field	Function
	0000000000010b - 2 bytes 0000000000011b - 3 bytes 0000000000100b - 4 bytes 0000111111111b - 511 bytes 0001000000000b - 512 bytes 0100000000000b - 2048 bytes 1000000000000b - 4096 bytes

10.3.6.1.4 Command Argument (CMD_ARG)

10.3.6.1.4.1 Offset

Register	Offset
CMD_ARG	8h

10.3.6.1.4.2 Function

This register contains the SD/MMC command argument.

10.3.6.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CMDARG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMDARG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.4.4 Fields

Field	Function
31-0 CMDARG	Command argument The SD/MMC command argument is specified as bits 39-8 of the command format in the SD or MMC specification. This field is write protected when the Command Inhibit (CMD) field in the Present State register is set.

10.3.6.1.5 Command Transfer Type (CMD_XFR_TYP)

10.3.6.1.5.1 Offset

Register	Offset
CMD_XFR_TYP	Ch

10.3.6.1.5.2 Function

This register is used to control the operation of data transfers. The host driver sets this register before issuing a command followed by a data transfer or before issuing a Resume command. To prevent data loss, uSDHC prevents writing to the bits, which are involved in the data transfer of this register, when data transfer is active. These fields are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN, and DMAEN.

The host driver checks the Command Inhibit DAT field (PRES_STATE[CDIHB]) and the Command Inhibit CMD field (PRES_STATE[CIHB]) in the Present State register before writing to this register. When the CDIHB field in the Present State register is set, any attempt to send a command with data by writing to this register is ignored; when the CIHB field is set, any write to this register is ignored.

On sending commands with data transfer involved, it is mandatory that the block size is non-zero. Block count must also be non-zero, or indicated as a single block transfer (bit 5 of this register is '0' when written), or block count is disabled (bit 1 of this register is '0' when written), otherwise uSDHC ignores the sending of this command and do nothing. For write command, with all above restrictions, it is also mandatory that the write protect switch is not active (PRES_STATE[WPSPL] field of Present State register is '1'); otherwise, uSDHC also ignores the command.

If the commands with data transfer do not receive the response in 64 clock cycles, that is, if response time-out happens, uSDHC treats the external device, does not accept the command, and aborts the data transfer. In this scenario, the driver should issue the command again to retry the transfer. It is also possible that for some reason the card responds to the command but uSDHC does not receive the response, and if it is internal DMA (either simple DMA or ADMA) read operation, the external system memory is over-written by the internal DMA with data sent back from the card.

The table below shows the summary of how register settings determine the type of data transfer.

Table 10-46. Transfer type register setting for various transfer types

Multi/single block select	Block count enable	Block count	Function
0	Do not care	Do not care	Single transfer
1	0	Do not care	Infinite transfer
1	1	Positive number	Multiple transfer
1	1	Zero	No data transfer

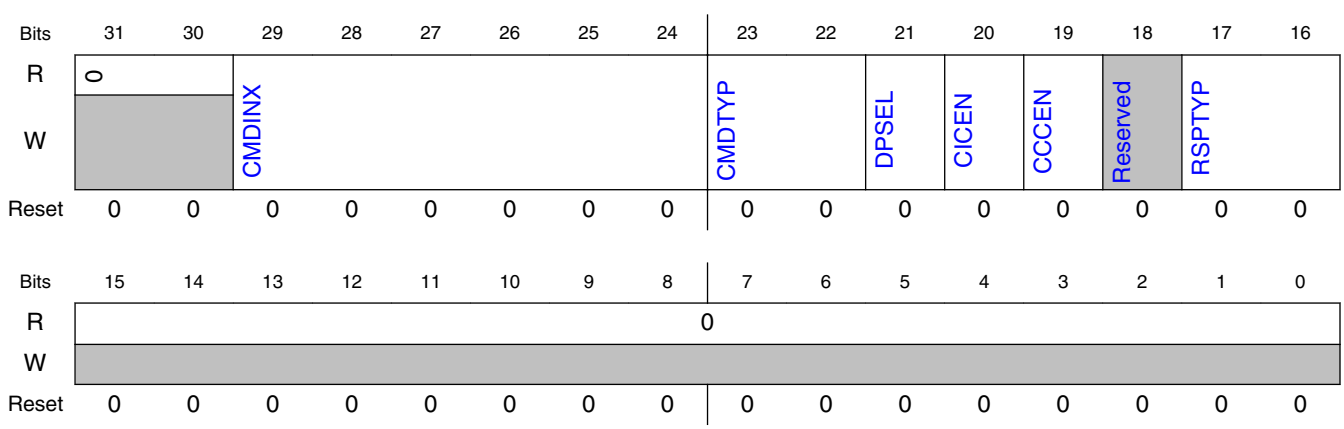
The table below shows the relationship between the Command Index Check Enable and the Command CRC Check Enable, regarding the Response Type bits as well as the name of the response type.

Table 10-47. Relationship between parameters and the name of the response type

Response type	Index check enable	CRC check enable	Name of response type
00	0	0	No response
01	0	1	R2
10	0	0	R3, R4
10	1	1	R1, R5, R6
11	1	1	R1b,R5b

- In the SDIO specification, response type notation for R5b is not defined. R5 includes R5b in the SDIO specification, but R5b is defined in this specification to specify that uSDHC checks the busy status after receiving a response. For example, usually CMD52 is used with R5, but the I/O abort command is used with R5b.
- The CRC fields for R3 and R4 are expected to be all 1 bits. The CRC check is disabled for these response types.

10.3.6.1.5.3 Diagram



10.3.6.1.5.4 Fields

Field	Function
31-30 —	Reserved
29-24 CMDINX	Command index These fields are set to the command number that is specified in bits 45-40 of the command-format in the SD Memory Card Physical Layer Specification and SDIO Card Specification.
23-22 CMDTYP	Command type There are three types of special commands: Suspend, Resume, and Abort. These fields are set to 00b for all other commands. • Suspend command: If the Suspend command succeeds, uSDHC assumes that the card bus has been released and that it is possible to issue the next command that uses the DATA line. Because uSDHC does not monitor the content of command response, it does not know if the Suspend command succeeded or not. It is the host driver's responsibility to check the status of the Suspend command and send another command marked as Suspend to inform uSDHC that a Suspend command was successfully issued. See Suspend Resume for more details. After the end bit of command is sent, uSDHC deasserts Read Wait for read transactions and stops checking busy for write transactions. In a 4-bit mode, the interrupt cycle starts. If the Suspend command fails, uSDHC maintains its current state, and the host driver restarts the transfer by setting the Continue Request field in the Protocol Control register. • Resume command: The host driver re-starts the data transfer by restoring the registers saved before sending the Suspend command and then sends the Resume command. The uSDHC module checks for a pending busy state before starting write transfers. • Abort command: If this command is set when executing a read transfer, uSDHC stops reads to the buffer. If this command is set when executing a write transfer, uSDHC stops driving the DATA line. After issuing the Abort command, the host driver should issue a software reset (Abort Transaction). 00b - Normal other commands 01b - Suspend CMD52 for writing bus suspend in CCCR 10b - Resume CMD52 for writing function select in CCCR 11b - Abort CMD12, CMD52 for writing I/O Abort in CCCR
21 DPSEL	Data present select This field is set to 1 to indicate that data is present and is transferred using the DATA line. It is set to 0 for the following: • Commands using only the CMD line (for example, CMD52) • Commands with no data transfer, but using the busy signal on DATA0 line (R1b or R5b (for example, CMD38)) **NOTE:** In resume command, this field is set, and other bits in this register is set the same as when the transfer was initially launched. When the write protect switch is on, (that is, the WPSPL field is active as '0'), any command with a write operation ignored. When this field is set, while the DTDSEL field is 0, writes to the register Transfer Type are ignored. 0b - No data present 1b - Data present
20 CICEN	Command index check enable If this field is set to 1, uSDHC checks the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this field is set to 0, the Index field is not checked. 0b - Disable command index check 1b - Enables command index check

Table continues on the next page...

Field	Function
19 CCCEN	Command CRC check enable If this field is set to 1, uSDHC checks the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this field is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. See RSPTYP[1:0] and Command Transfer Type (CMD_XFR_TYP). 0b - Disables command CRC check 1b - Enables command CRC check
18 —	Reserved
17-16 RSPTYP	Response type select 00b - No response 01b - Response length 136 10b - Response length 48 11b - Response length 48, check busy after response
15-0 —	Reserved

10.3.6.1.6 Command Response0 (CMD_RSP0)

10.3.6.1.6.1 Offset

Register	Offset
CMD_RSP0	10h

10.3.6.1.6.2 Function

This register is used to store part 0 of the response bits from the card.

10.3.6.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.6.4 Fields

Field	Function
31-0 CMDRSP0	Command response 0 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

10.3.6.1.7 Command Response1 (CMD_RSP1)

10.3.6.1.7.1 Offset

Register	Offset
CMD_RSP1	14h

10.3.6.1.7.2 Function

This register is used to store part 1 of the response bits from the card.

10.3.6.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.7.4 Fields

Field	Function
31-0 CMDRSP1	Command response 1 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

10.3.6.1.8 Command Response2 (CMD_RSP2)

10.3.6.1.8.1 Offset

Register	Offset
CMD_RSP2	18h

10.3.6.1.8.2 Function

This register is used to store part 2 of the response bits from the card.

10.3.6.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.8.4 Fields

Field	Function
31-0 CMDRSP2	Command response 2 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

10.3.6.1.9 Command Response3 (CMD_RSP3)

10.3.6.1.9.1 Offset

Register	Offset
CMD_RSP3	1Ch

10.3.6.1.9.2 Function

This register is used to store part 3 of the response bits from the card.

The table below describes the mapping of command responses from the SD bus to Command Response registers for each response type. In this table, R[ ] refers to a bit range within the response data as transmitted on the SD bus.

Table 10-48. Response bit definition for each response type

Response type	Meaning of response	Response field	Response register
R1,R1b (normal response)	Card status	R[39:8]	CMDRSP0
R1b (auto CMD12 response)	Card status for auto CMD12	R[39:8]	CMDRSP3
R2 (CID, CSD register)	CID/CSD register [127:8]	R[127:8]	{CMDRSP3[23:0], CMDRSP2, CMDRSP1, CMDRSP0}
R3 (OCR register)	OCR register for memory	R[39:8]	CMDRSP0
R4 (OCR register)	OCR register for I/O etc.	R[39:8]	CMDRSP0
R5, R5b	SDIO response	R[39:8]	CMDRSP0
R6 (publish RCA)	New published RCA[31:16] and card status[15:0]	R[39:9]	CMDRSP0

This table shows that most responses with a length of 48 (R[47:0]) have 32-bits of the response data (R[39:8]) stored in the CMDRSP0 register. Responses of type R1b (Auto CMD12 responses) have response data bits (R[39:8]) stored in the CMDRSP3 register. Responses with length 136 (R[135:0]) have 120-bits of the response data (R[127:8]) stored in the CMDRSP0, 1, 2, and 3 registers.

To be able to read the response status efficiently, uSDHC only stores part of the response data in the Command Response registers. This enables the host driver to efficiently read 32-bits of response data in one read cycle on a 32-bit bus system. Parts of the response, the Index field and the CRC, are checked by uSDHC (as specified by the Command Index Check Enable and the Command CRC Check Enable bits in the Transfer Type register) and generate an error interrupt if any error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, uSDHC checks R[47:1], and if the response length is 136 the uSDHC checks R[119:1].

Because uSDHC may have a multiple block data transfer executing concurrently with a CMD_wo_DAT command, uSDHC stores the Auto CMD12 response in the CMDRSP3 register. The CMD_wo_DAT response is stored in CMDRSP0. This allows uSDHC to

avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa. When uSDHC modifies part of the Command Response registers, as shown in the table above, it preserves the unmodified bits.

10.3.6.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.9.4 Fields

Field	Function
31-0 CMDRSP3	Command response 3 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

10.3.6.1.10 Data Buffer Access Port (DATA_BUFF_ACC_PORT)

10.3.6.1.10.1 Offset

Register	Offset
DATA_BUFF_ACC_PORT	20h

10.3.6.1.10.2 Function

The Buffer Data Port register is for 32-bit data access by the Arm platform or the external DMA. When the internal DMA is enabled, any write to this field is ignored, and any read from this field always yields 0s.

10.3.6.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATCONT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATCONT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.10.4 Fields

Field	Function
31-0 DATCONT	Data content This field is used to access the internal buffer.

10.3.6.1.11 Present State (PRES_STATE)

10.3.6.1.11.1 Offset

Register	Offset
PRES_STATE	24h

10.3.6.1.11.2 Function

The host driver can get status of uSDHC from this 32-bit read only register.

The host driver can issue CMD0, CMD12, CMD13 (for memory) and CMD52 (for SDIO) when the DATA lines are busy during a data transfer. These commands can be issued when Command Inhibit (CMD) is set to zero. Other commands are issued when Command Inhibit (DATA) is set to zero. Possible changes to the SD Physical Specification may add other commands to this list in the future.

NOTE

The reset value of Present State register depends on board connectivity.

10.3.6.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLSL								CLSL	0			WPSPL	CDPL	0	CINST
W																
Reset	u	u	u	u	u	u	u	u	u	0	0	0	u	u	0	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSCD	0		RTR	BREN	BWEN	RTA	WTA	SDOFF	PEROFF	HCKOFF	IPGOFF	SDSTB	DLA	CDIHB	CIHB
W																
Reset	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

10.3.6.1.11.4 Fields

Field	Function
31-24 DLSL	DATA[7:0] line signal level This field is used to check the DATA line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from DATA0. The reset value is affected by the external pull-up / pull-down resistors. By default, the read value of this field after reset is 8'b11110111, when DATA3 is pulled down and the other lines are pulled up. 00000000b - Data 0 line signal level 00000001b - Data 1 line signal level 00000010b - Data 2 line signal level 00000011b - Data 3 line signal level 00000100b - Data 4 line signal level 00000101b - Data 5 line signal level 00000110b - Data 6 line signal level 00000111b - Data 7 line signal level
23 CLSL	CMD line signal level This field is used to check the CMD line level to recover from errors, and for debugging. The reset value is affected by the external pull-up / pull-down resistor, by default, the read value of this field after reset is 1'b1, when the command line is pulled up.
22-20 —	Reserved
19 WPSPL	Write protect switch pin level The Write Protect switch is supported for memory and combo cards. This field reflects the inverted value of the WP pin of the card socket. A software reset does not affect this field. The reset value is affected by the external write protect switch. If the WP pin is not used, it should be tied low, so that the reset value of this field is high and write is enabled. 0b - Write protected (WP = 1) 1b - Write enabled (WP = 0)
18	Card detect pin level

Table continues on the next page...

Field	Function
CDPL	This field reflects the inverse value of the CD_B pin for the card socket. Debouncing is not performed on this field. This field may be valid, but is not guaranteed, because of propagation delay. Use of this field is limited to testing because it must be debounced by software. A software reset does not affect this field. A write to the Force Event Register does not affect this field. The reset value is affected by the external card detection pin. This field shows the value on the CD_B pin (that is, when a card is inserted in the socket, it is 0 on the CD_B input, and consequently, the CDPL reads 1. 0b - No card present (CD_B = 1) 1b - Card present (CD_B = 0)
17 —	Reserved
16 CINST	Card inserted This field indicates whether a card has been inserted. The uSDHC module debounces this signal so that the host driver does not need to wait for it to stabilize. Changing from a 0 to 1 generates a Card Insertion interrupt in the Interrupt Status register. Changing from a 1 to 0 generates a Card Removal interrupt in the Interrupt Status register. A write to the Force Event Register does not affect this field. The Software Reset For All in the System Control register does not affect this field. A software reset does not affect this field. 0b - Power on reset or no card 1b - Card inserted
15 TSCD	Tap select change done This field indicates the delay setting is effective after write CLK_TUNE_CTRL_STATUS register. 0b - Delay cell select change is not finished. 1b - Delay cell select change is finished.
14-13 —	Reserved
12 RTR	Re-Tuning Request (only for SD3.0 SDR104 mode,and EMMC HS200 mode) Host controller may request host driver to execute re-tuning sequence by setting this field when the data window is shifted by temperature drift and a tuned sampling point does not have a good margin to receive correct data. This field is cleared when a command is issued with setting Execute Tuning field in MIX_CTRL register. Changing of this field from 0 to 1 generates Re-Tuning Event. See Interrupt status registers for more detail. This field isn't set to 1 if Sampling Clock Select in the MIX_CTRL register is set to 0 (using fixed sampling clock). 0b - Fixed or well tuned sampling clock 1b - Sampling clock needs re-tuning
11 BREN	Buffer read enable This status field is used for non-DMA read transfers. The uSDHC module implements an internal buffer to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer. If this field is high, valid data greater than the watermark level exist in the buffer. A change of this field from 1 to 0 occurs when some reads from the buffer (read DATPORT (Base + 0x20)) are made and the buffer hasn't valid data greater than the watermark level. A change of this field from 0 to1 occurs when there is enough valid data ready in the buffer and the Buffer Read Ready interrupt has been generated and enabled. 0b - Read disable 1b - Read enable
10	Buffer write enable

Table continues on the next page...

Field	Function
BWEN	This status field is used for non-DMA write transfers. The uSDHC module implements an internal buffer to transfer data efficiently. This read only flag indicates if space is available for write data. If this field is 1, valid data greater than the watermark level can be written to the buffer. A change of this field from 1 to 0 occurs when some writes to the buffer (write DATPORT (Base + 0x20)) are made and the buffer hasn't valid space greater than the watermark level. A change of this field from 0 to 1 occurs when the buffer can hold valid data greater than the write watermark level and the Buffer Write Ready interrupt is generated and enabled. 0b - Write disable 1b - Write enable
9 RTA	Read transfer active This status field is used for detecting completion of a read transfer. This field is set for either of the following conditions: • After the end field of the read command • When writing a 1 to the Continue Request field in the Protocol Control register to restart a read transfer A transfer complete interrupt is generated when this field changes to 0. This field is cleared for either of the following conditions: • When the last data block as specified by block length is transferred to the System, that is, all data are read away from uSDHC internal buffer. • When all valid data blocks have been transferred from uSDHC internal buffer to the System and no current block transfers are being sent because of the Stop At Block Gap Request being set to 1. 0b - No valid data 1b - Transferring data
8 WTA	Write transfer active This status field indicates a write transfer is active. If this field is 0, it means no valid write data exists in uSDHC. This field is set in either of the following cases: • After the end field of the write command • When writing 1 to the Continue Request field in the Protocol Control register to restart a write transfer This field is cleared in either of the following cases: • After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple) • After getting the CRC status of any block where data transmission is about to be stopped by a Stop At Block Gap Request During a write transaction, a Block Gap Event interrupt is generated when this field is changed to 0, as result of the Stop At Block Gap Request being set. This status is useful for the host driver in determining when to issue commands during Write Busy state. 0b - No valid data 1b - Transferring data
7 SDOFF	SD clock gated off internally This status field indicates that the SD clock is internally gated off, because of buffer over / under-run or read pause without read wait assertion, or the driver set FRC_SDCLK_ON field is 0 to stop the SD clock in idle status. This field is for the host driver to debug data transaction on the SD bus. 0b - SD clock is active. 1b - SD clock is gated off.

Table continues on the next page...

Field	Function
6 PEROFF	IPG_PERCLK gated off internally This status field indicates that the IPG_PERCLK is internally gated off. This field is for the host driver to debug transaction on the SD bus. When IPG_CLK_SOFT_EN is cleared, IPG_PERCLK is gated off, otherwise IPG_PERCLK is always active. 0b - IPG_PERCLK is active. 1b - IPG_PERCLK is gated off.
5 HCKOFF	HCLK gated off internally This status field indicates that the HCLK is internally gated off. This field is for the host driver to debug during a data transfer. 0b - HCLK is active. 1b - HCLK is gated off.
4 IPGOFF	Peripheral clock gated off internally This status field indicates that the peripheral clock is internally gated off. This field is for the host driver to debug. 0b - Peripheral clock is active. 1b - Peripheral clock is gated off.
3 SDSTB	SD clock stable This status field indicates that the internal card clock is stable. This field is for the host driver to poll clock status when changing the clock frequency. It is recommended to clear FRC_SDCLK_ON field in System Control register to remove glitches on the card clock when the frequency is changing. Before changing clock divisor value (SDCLKFS or DVS), host driver should make sure the SDSTB field is high. 0b - Clock is changing frequency and not stable. 1b - Clock is stable.
2 DLA	Data line active This status field indicates whether one of the DATA lines on the SD bus is in use. In the case of read transactions: This status indicates if a read transfer is executing on the SD bus. Changes in this value from 1 to 0, between data blocks, generates a Block Gap Event interrupt in the Interrupt Status register. This field is set in either of the following cases: • After the end field of the read command • When writing a 1 to the Continue Request field in the Protocol Control register to restart a read transfer This field is cleared in either of the following cases: • When the end field of the last data block is sent from the SD bus to uSDHC. • When the Read Wait state is stopped by a Suspend command and the DATA2 line is released. The uSDHC module waits at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), uSDHC can wait for a current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait to use the suspend / resume function. This field remains 1 during Read Wait. In the case of write transactions: This status indicates that a write transfer is executing on the SD bus. Changes in this value from 1 to 0 generate a Transfer Complete interrupt in the Interrupt Status register. This field is set in either of the following cases:

Table continues on the next page...

Field	Function
	• After the end field of the write command • When writing to 1 to the Continue Request field in the Protocol Control register to continue a write transfer This field is cleared in either of the following cases: • When the SD card releases Write Busy of the last data block, uSDHC also detects if the output is not busy. If the SD card does not drive the busy signal after the CRC status is received, uSDHC assumes the card drive "Not Busy". • When the SD card releases write busy, prior to waiting for write transfer, and because of a Stop At Block Gap Request. In the case of command with busy pending: This status indicates that a busy state follows the command and the data line is in use. This field is cleared when the DATA0 line is released. 0b - DATA line inactive 1b - DATA line active
1 CDIHB	Command Inhibit Data (DATA) This status field is generated if either the DAT Line Active or the Read Transfer Active is set to 1. If this field is 0, it indicates that uSDHC can issue the next SD / MMC Command. Commands with a busy signal belong to Command Inhibit (DATA) (for example. R1b, R5b type). Changing from 1 to 0 generates a Transfer Complete interrupt in the Interrupt Status register. **NOTE:** The SD host driver can save registers for a suspend transaction after this field has changed from 1 to 0. 0b - Can issue command that uses the DATA line 1b - Cannot issue command that uses the DATA line
0 CIHB	Command inhibit (CMD) If this status bit is 0, it indicates that the CMD line is not in use and uSDHC can issue a SD / MMC command using the CMD line. This field is set also immediately after the Transfer Type register is written. This field is cleared when the command response is received. Even if the Command Inhibit (DATA) is set to 1, commands using only the CMD line can be issued if this field is 0. Changing from 1 to 0 generates a Command Complete interrupt in the Interrupt Status register. If uSDHC cannot issue the command because of a command conflict error (see Command CRC Error) or because of a Command Not Issued By Auto CMD12 Error, this field remains 1 and the Command Complete is not set. The Status of issuing an auto CMD12 does not show on this field. 0b - Can issue command using only CMD line 1b - Cannot issue command

10.3.6.1.12 Protocol Control (PROT_CTRL)

10.3.6.1.12.1 Offset

Register	Offset
PROT_CTRL	28h

10.3.6.1.12.2 Function

This register controls three cases to restart the transfer after stop at the block gap. Which case is appropriate depends on whether uSDHC issues a Suspend command or the SD card accepts the Suspend command.

- If the host driver does not issue a Suspend command, the Continue request is used to restart the transfer.
- If the host driver issues a Suspend command and the SD card accepts it, a Resume command is used to restart the transfer.
- If the host driver issues a Suspend command and the SD card does not accept it, the Continue request is used to restart the transfer.

Any time stop at block gap request stops the data transfer, the host driver waits for a Transfer Complete (in the Interrupt Status register), before attempting to restart the transfer. When restarting the data transfer by Continue Request, the host driver clears the Stop At Block Gap Request before or simultaneously.

10.3.6.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	NON_EXACT_BLK_RD	Reserved			WECRM	WECINS	WECINT	Reserved			RD_DONE_NO_8CLK	IABG	RWCTL	CREQ	SABGREQ
W																
Reset	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						DMASEL		CDSS	CDTL	EMODE		D3CD	DTW		Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

10.3.6.1.12.4 Fields

Field	Function
31	Reserved
—	Always write as 0

Table continues on the next page...

Field	Function
30 NON_EXACT_BLK_RD	Non-exact block read Current block read is non-exact block read. It is only used for SDIO. 0b - The block read is exact block read. Host driver does not need to issue abort command to terminate this multi-block read. 1b - The block read is non-exact block read. Host driver needs to issue abort command to terminate this multi-block read.
29-27 —	Reserved
26 WECRM	Wakeup event enable on SD card removal This field enables a wakeup event, via a card removal, in the Interrupt Status register. FN_WUS (Wakeup Support) in CIS does not affect this field. When this field is set, the Card Removal Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Removal Status and uSDHC interrupt. 0b - Disables wakeup event enable on SD card removal 1b - Enables wakeup event enable on SD card removal
25 WECINS	Wakeup event enable on SD card insertion This field enables a wakeup event, via a card insertion, in the Interrupt Status register. FN_WUS (Wakeup Support) in CIS does not affect this field. When this field is set, the Card Insertion Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Insertion Status and uSDHC interrupt. 0b - Disable wakeup event enable on SD card insertion 1b - Enable wakeup event enable on SD card insertion
24 WECINT	Wakeup event enable on card interrupt This field enables a wakeup event, via a card interrupt, in the Interrupt Status register. This field can be set to 1 if FN_WUS (Wakeup Support) in CIS is set to 1. When this field is set, the Card Interrupt Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Interrupt Status and uSDHC interrupt. 0b - Disables wakeup event enable on card interrupt 1b - Enables wakeup event enable on card interrupt
23-21 —	Reserved Always write as 3'b100
20 RD_DONE_NO_8CLK	Read performed number 8 clock According to the SD/MMC spec, for read data transaction, 8 clocks are needed after the end field of the last data block. So, by default(RD_DONE_NO_8CLK=0), eight clocks are active after the end field of the last read data transaction. However, these 8 clocks should not be active if user wants to use stop at block gap (include the auto stop at block gap in boot mode) feature for read and the RWCTL field (bit18) is not enabled. In this case, software should set RD_DONE_NO_8CLK to avoid these 8 clocks. Otherwise, the device might send extra data to uSDHC while uSDHC ignores these data. In a summary, this field should be set only if the use case needs to use stop at block gap feature while the device can't support the read wait feature.
19 IABG	Interrupt at block gap This field is valid only in 4-bit mode, of the SDIO card, and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SDIO card cannot signal an interrupt during a multiple block transfer, this field should be set to 0 to avoid an inadvertent interrupt. When the host driver detects an SDIO card insertion, it sets this field according to the CCCR of the card.

Table continues on the next page...

Field	Function
	0b - Disables interrupt at block gap 1b - Enables interrupt at block gap
18 RWCTL	Read wait control The read wait function provided by this field is optional for SDIO cards. If the card supports read wait, set this field to enable use of the read wait protocol to stop read data using the DATA2 line. Otherwise, uSDHC has to stop the SD clock to hold read data, which restricts commands generation. When the host driver detects an SDIO card insertion, it sets this field according to the CCCR of the card. If the card does not support read wait, this field should never be set to 1; otherwise, DATA line conflicts might occur. If this field is set to 0, stop at block gap during read operation is also supported, but uSDHC stops the SD clock to pause reading operation. 0b - Disables read wait control and stop SD clock at block gap when SABGREQ field is set 1b - Enables read wait control and assert read wait without stopping SD clock at block gap when SABGREQ field is set
17 CREQ	Continue request This field is used to restart a transaction which was stopped using the stop at block gap request. When a suspend operation is not accepted by the card, it is also by setting this field to restart the paused transfer. To cancel stop at the block gap, set stop at block gap request to 0 and set this field to 1 to restart the transfer. The uSDHC module automatically clears this field, therefore it is not necessary for the host driver to set this field to 0. If both stop at block gap request and this field are 1, the continue request is ignored. 0b - No effect 1b - Restart
16 SABGREQ	Stop at block gap request This field is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the transfer complete is set to 1, indicating a transfer completion, the host driver leaves this field set to 1. Clearing both the stop at block gap request and continue request does not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The uSDHC module supports the stop at block gap request for write transfers, but for read transfers it requires that the SDIO card support read wait. Therefore, the host driver does not set this field during read transfers unless the SDIO card supports Read Wait and has set the read wait control to 1; otherwise, uSDHC stops the SD bus clock to pause the read operation during block gap. In the case of write transfers in which the host driver writes data to the Data Port register, the host driver sets this field after all block data is written. If this field is set to 1, the host driver does not write data to the Data Port register after a block is sent. Once this field is set, the host driver does not clear this field before the Transfer Complete field in Interrupt Status register is set, otherwise uSDHC's behavior is undefined. This field effects read transfer active, write transfer active, DATA Line Active and Command Inhibit (DATA) in the Present State register. 0b - Transfer 1b - Stop
15-10 —	Reserved
9-8 DMASEL	DMA select This field is valid while DMA (SDMA or ADMA) is enabled and selects the DMA operation. 00b - No DMA or simple DMA is selected. 01b - ADMA1 is selected. 10b - ADMA2 is selected. 11b - Reserved
7 CDSS	Card detect signal selection This field selects the source for the card detection.

Table continues on the next page...

Field	Function
	0b - Card detection level is selected (for normal purpose). 1b - Card detection test level is selected (for test purpose).
6 CDTL	Card detect test level This field is enabled while the card detection signal selection is set to 1 and it indicates card insertion. 0b - Card detect test level is 0, no card inserted 1b - Card detect test level is 1, card inserted
5-4 EMODE	Endian mode This field supports all three endian modes in data transfer. See Data buffer for more details. 00b - Big endian mode 01b - Half word big endian mode 10b - Little endian mode 11b - Reserved
3 D3CD	DATA3 as card detection pin If this field is set, DATA3 should be pulled down to act as a card detection pin. Be cautious when using this feature, because DATA3 is also a chip-select for the SPI mode. A pull-down on this pin and CMD0 might set the card into the SPI mode, which uSDHC does not support. 0b - DATA3 does not monitor card insertion 1b - DATA3 as card detection pin
2-1 DTW	Data transfer width This field selects the data width of the SD bus for a data transfer. The host driver sets it to match the data width of the card. Possible data transfer width is 1-bit, 4-bits or 8-bits. 00b - 1-bit mode 01b - 4-bit mode 10b - 8-bit mode 11b - Reserved
0 —	Reserved

10.3.6.1.13 System Control (SYS_CTRL)

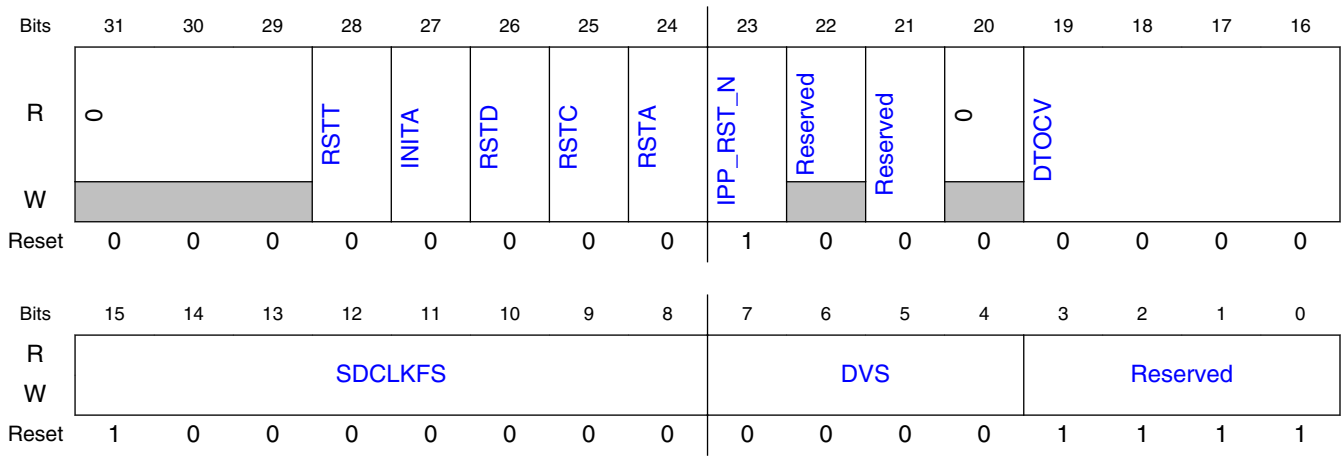
10.3.6.1.13.1 Offset

Register	Offset
SYS_CTRL	2Ch

10.3.6.1.13.2 Function

This register provides control of the system. See detail in the field description.

10.3.6.1.13.3 Diagram



10.3.6.1.13.4 Fields

Field	Function
31-29 —	Reserved
28 RSTT	Reset tuning When set this field to 1, it resets tuning circuit. After tuning circuits are reset, bit value is 0. Clearing execute_tuning field in AUTOCMD12_ERR_STATUS also sets this field to 1 to reset tuning circuit
27 INITA	Initialization active When this field is set, 80 SD-clocks are sent to the card. After the 80 clocks are sent, this field is self cleared. This field is very useful during the card power-up period when 74 SD-clocks are needed and the clock auto gating feature is enabled. Writing 1 to this bit when this field is already 1 has no effect. Writing 0 to this field at any time has no effect. When either of the CIHB and CDIHB fields in the Present State register are set, writing 1 to this field is ignored (that is, when command line or data lines are active, write to this field is not allowed). On the other-hand, when this field is set, that is, during initialization active period, it is allowed to issue command, and the command bit stream appears on the CMD pad after all 80 clock cycles are done. So, when this command ends, the driver can make sure the 80 clock cycles are sent out. This is very useful when the driver needs to send 80 cycles to the card and does not want to wait till this field is self cleared.

26
RSTD
	Software reset for data line
	Only part of the data circuit is reset. DMA circuit is also reset. After this field is set, the software waits for self-clear.
	The following registers and bits are cleared by this field:
	• Data Port register
	  • Buffer is cleared and initialized
	• Present State register
	  • Buffer read enable
	  • Buffer write enable
	  • Read transfer active
	  • Write transfer active
	  • DATA line active
	  • Command Inhibit (DATA)

Table continues on the next page...

Field	Function

	• Protocol Control register
	  • Continue request
	• Interrupt Status register
	  • Buffer read ready
	  • Buffer write ready
	  • DMA interrupt
	  • Block gap event
	  • Transfer complete
	
	NOTE: When reset, the software must make sure there is no incomplete data transferring. If there is data transfer going on, the software needs to wait TC or DC INT_STATUS register is set.
	0b - No reset
	1b - Reset

25
RSTC
	Software reset for CMD line
	Only part of the command circuit is reset. After this field is set, the software waits for self-clear.
	The following registers and bits are cleared by this field:
	• Present State Register
	  • Command Inhibit (CMD)
	• Interrupt Status Register
	  • Command Complete
	
	0b - No reset
	1b - Reset

24 RSTA	Software reset for all This reset affects the entire host controller except for the card detection circuit. RSTA resets all the registers that can be reset by RSTC/RSTD. During its initialization, the host driver is set this field to 1 to reset uSDHC. The uSDHC module resets this field to 0 when the capabilities registers are valid and the host driver can read them. Additional use of Software Reset For All does not affect the value of the capabilities registers. After this field is set, it is recommended that the host driver reset the external card and re-initialize it. After this field is set, the software should wait for self-clear. In tuning process, after every CMD19 is finished, this field is set to retest uSDHC. **NOTE:** When reset, the software must make sure there is no incomplete data transferring. If there is data transfer going on, the software needs to wait TC or DC INT_STATUS register is set. 0b - No reset 1b - Reset
23 IPP_RST_N	Hardware reset This field's value is output to card through pad directly to hardware reset pin of the card if the card supports this feature.
22 —	Reserved
21 —	Reserved
20 —	Reserved
19-16 DTOCV	Data timeout counter value This value determines the interval by which DAT line timeouts are detected. See the Data Timeout Error field in the Interrupt Status register for information on factors that dictate time-out generation. Time-out clock frequency is generated by dividing the base clock SDCLK value by this value.

Table continues on the next page...

Field	Function
	The host driver can clear the Data Timeout Error Status Enable (in the Interrupt Status Enable register) to prevent inadvertent time-out events. 0000b - SDCLK x 2^{32} 0001b - SDCLK x 2^{33} 0010b - SDCLK x 2^{18} 0011b - SDCLK x 2^{19} 1101b - SDCLK x 2^{29} recommend to use for other speed mode except HS400/HS200/SDR104 mode 1110b - SDCLK x 2^{30} recommend to use for HS200/SDR104 mode 1111b - SDCLK x 2^{31} recommend to use for HS400 mode
15-8 SDCLKFS	SDCLK frequency select This field is used to select the frequency of the SDCLK pin. The frequency is not programmed directly, rather this field holds the prescaler (of this register) and divisor (next register) of the Base Clock Frequency register. In Single Data Rate mode (DDR_EN field of MIXERCTRL is '0') Only the following settings are allowed: 80h) Base clock divided by 256 40h) Base clock divided by 128 20h) Base clock divided by 64 10h) Base clock divided by 32 08h) Base clock divided by 16 04h) Base clock divided by 8 02h) Base clock divided by 4 01h) Base clock divided by 2 00h) Base clock divided by 1 While in Dual Data Rate mode (DDR_EN field of MIXERCTRL is '1') Only the following settings are allowed: 80h) Base clock divided by 512 40h) Base clock divided by 256 20h) Base clock divided by 128 10h) Base clock divided by 64 08h) Base clock divided by 32 04h) Base clock divided by 16 02h) Base clock divided by 8 01h) Base clock divided by 4 00h) Base clock divided by 2 When the software changes the DDR_EN field, SDCLKFS might need to be changed also. In Single Data Rate mode, setting 00h bypasses the frequency prescaler of the SD clock. Multiple bits must not be set, or the behavior of this prescaler is undefined. The two default divider values can be calculated by the frequency of ipg_perclk and the following Divisor bits. The frequency of SDCLK is set by the following formula: Clock Frequency = (Base Clock) / (prescaler x divisor)

Table continues on the next page...

Field	Function
	For example, in Single Data Rate mode, if the Base Clock Frequency is 96 MHz, and the target frequency is 25 MHz, then choosing the prescaler value of 01h and divisor value of 1h yields 24 MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400 kHz, the prescaler value of 08h and divisor value of eh yields the exact clock value of 400 kHz. The reset value of this field is 80h, so if the input Base Clock (ipg_perclk) is about 96 MHz, the default SD clock after reset is 375 kHz. According to the SD Physical Specification Version 1.1 and the SDIO Card Specification Version 1.2, the maximum SD clock frequency is 50 MHz and is never exceed this limit. Before changing clock divisor value (SDCLKFS or DVS), host driver should make sure the SDSTB field is high. If setting SDCLKFS and DVS can generate the same clock frequency,(for example, in SDR mode, SDCLKFS = 01h is same as DVS = 01h.), SDCLKFS is highly recommended.
7-4 DVS	Divisor This field is used to provide a more exact divisor to generate the desired SD clock frequency. Note the divider can even support odd divisors without deterioration of duty cycle. Before changing clock divisor value (SDCLKFS or DVS), Host driver should make sure the SDSTB field is high. The settings are as follows: 0000b - Divide-by-1 0001b - Divide-by-2 1110b - Divide-by-15 1111b - Divide-by-16
3-0 —	Reserved Always write as 1.

10.3.6.1.14 Interrupt Status (INT_STATUS)

10.3.6.1.14.1 Offset

Register	Offset
INT_STATUS	30h

10.3.6.1.14.2 Function

An interrupt is generated when the normal interrupt signal enable is enabled and at least one of the status fields is set to 1. For all fields, writing 1 to a bit clears it; writing to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. For card interrupt, before writing 1 to clear, it is required that the card stops asserting the interrupt, meaning that when the card driver services the interrupt condition; otherwise, the CINT field is asserted again.

The table below shows the relationship between the command timeout error and the command complete.

Table 10-49. uSDHC status for command timeout error/command complete bit combinations

Command CRC Error	Command timeout error	Meaning of the status
0	0	X
X	1	Response not received within 64 SDCLK cycles
1	0	Response received

The table below shows the relationship between the transfer complete and the data timeout error.

Table 10-50. uSDHC status for data timeout error/transfer complete bit combinations

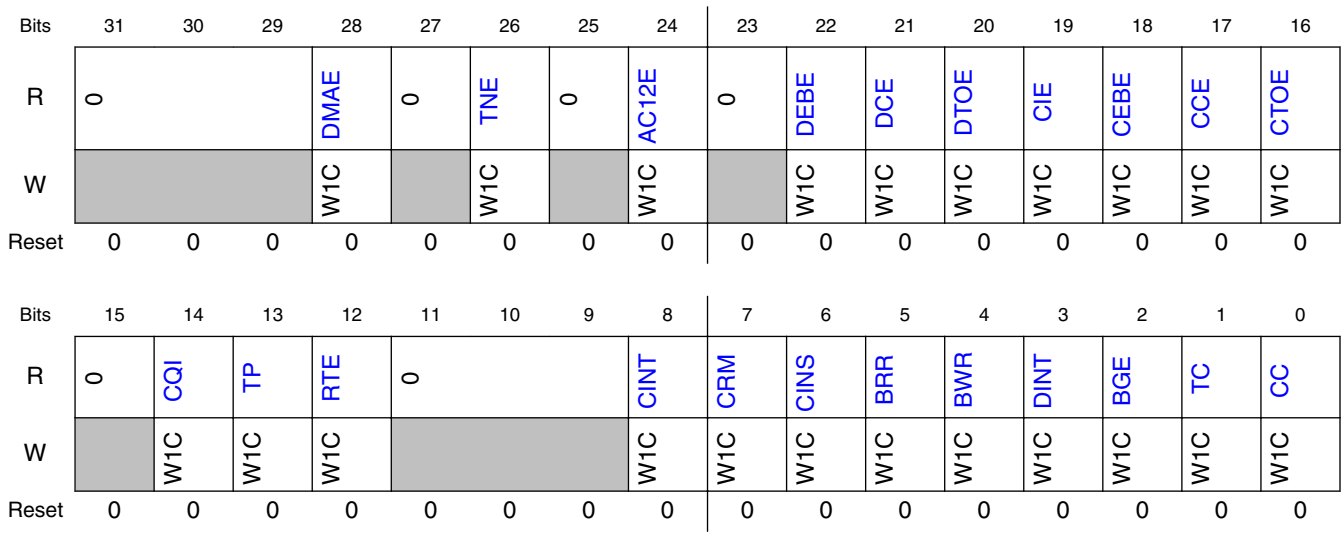
Transfer complete	Data timeout error	Meaning of the status
0	0	X
0	1	Timeout occurred during transfer
1	X	Data transfer complete

The table below shows the relationship between the command CRC error and command timeout error.

Table 10-51. uSDHC status for command CRC error/command timeout error bit combinations

Command complete	Command timeout error	Meaning of the status
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

10.3.6.1.14.3 Diagram



10.3.6.1.14.4 Fields

Field	Function
31-29 —	Reserved
28 DMAE	DMA error Occurs when an Internal DMA transfer has failed. This field is set to 1, when some error occurs in the data transfer. This error can be caused by either simple DMA or ADMA, depending on which DMA is in use. The value in DMA System Address register is the next fetch address where the error occurs. Because any error corrupts the whole data block, the host driver restarts the transfer from the corrupted block boundary. The address of the block boundary can be calculated either from the current DS_ADDR value or from the remaining number of blocks and the block size. 0b - No error 1b - Error
27 —	Reserved
26 TNE	Tuning error: (only for SD3.0 SDR104 mode and EMMC HS200 mode) This field is set when an unrecoverable error is detected in a tuning circuit. By detecting Tuning Error, host driver needs to abort a command executing and perform tuning.
25 —	Reserved
24 AC12E	Auto CMD12 error Occurs when detecting that one of the fields in the Auto CMD12 Error Status register has changed from 0 to 1. This field is set to 1, not only when the errors in Auto CMD12 occur, but also, when the Auto CMD12 is not executed due to the previous command error. 0b - No error 1b - Error
23	Reserved

Table continues on the next page...

Field	Function
—	
22 DEBE	Data end bit error Occurs either when detecting 0 at the end field position of read data that uses the DATA line, or at the end field position of the CRC. This field is not asserted in tuning process. 0b - No error 1b - Error
21 DCE	Data CRC error Occurs when detecting a CRC error when transferring read data that uses the DATA line, or when detecting the Write CRC status having a value other than 010. This field is not asserted in tuning process. 0b - No error 1b - Error
20 DTOE	Data timeout error Occurs when detecting one of following time-out conditions. • Busy time-out for R1b, R5b type • Busy time-out after Write CRC status • Read Data time-out. This field is not asserted in tuning process. 0b - No error 1b - Time out
19 CIE	Command index error Occurs if a command index error occurs in the command response. This field is not asserted in tuning process. 0b - No error 1b - Error
18 CEBE	Command end bit error Occurs when detecting that the end field of a command response is 0. This field is not asserted in tuning process. 0b - No error 1b - End bit error generated
17 CCE	Command CRC error Command CRC Error is generated in two cases. • If a response is returned and the Command Timeout Error is set to 0 (indicating no time-out), this field is set when detecting a CRC error in the command response. • The uSDHC module detects a CMD line conflict by monitoring the CMD line when a command is issued. If uSDHC drives the CMD line to 1, but detects 0 on the CMD line at the next SDCLK edge, then uSDHC aborts the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error should also be set to 1 to distinguish CMD line conflict. This field is not asserted in tuning process. 0b - No error 1b - CRC error generated
16	Command timeout error

Table continues on the next page...

Field	Function
CTOE	Occurs only if no response is returned within 64 SDCLK cycles from the end field of the command. If uSDHC detects a CMD line conflict, in which case a Command CRC Error is also be set (as shown in Interrupt Status (INT_STATUS)), this field is set without waiting for 64 SDCLK cycles. This is because the command is aborted by uSDHC. This field is not asserted in tuning process. 0b - No error 1b - Time out
15 —	Reserved
14 CQI	Command queuing interrupt This interrupt is asserted when at least one of the bits in CQIS register is set. This interrupt is cleared only by clearing the source interrupt in CQIS register.
13 TP	Tuning pass:(only for SD3.0 SDR104 mode and EMMC HS200 mode) Current CMD19 transfer is done successfully. That is, current sampling point is correct.
12 RTE	Re-tuning event: (only for SD3.0 SDR104 mode and EMMC HS200 mode) This status is set if Re-Tuning Request in the Present State register changes from 0 to 1. host controller requests host driver to perform re-tuning for next data transfer. Current data transfer (not large block count) can be completed without re-tuning. 0b - Re-tuning is not required. 1b - Re-tuning should be performed.
11-9 —	Reserved
8 CINT	Card interrupt This status field is set when an interrupt signal is detected from the external card. In 1-bit mode, uSDHC detects the Card Interrupt without the SD clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so the interrupt from card can only be sampled during interrupt cycle, introducing some delay between the interrupt signal from the SDIO card and the interrupt to the host system. Writing this field to 1 can clear this field, but as the interrupt source from the SDIO card does not clear, this field is set again. to clear this field, it is required to reset the interrupt source from the external card followed by a writing 1 to this field. When this status has been set, and the host driver needs to service this interrupt, the Card Interrupt Signal Enable in the Interrupt Signal Enable register should be 0 to stop driving the interrupt signal to the host system. After completion of the card interrupt service (It should reset the interrupt sources in the SDIO card and the interrupt signal might not be asserted), write 1 to clear this field, set the Card Interrupt Signal Enable to 1, and start sampling the interrupt signal again. 0b - No card interrupt 1b - Generate card interrupt
7 CRM	Card removal This status field is set if the Card Inserted field in the Present State register changes from 1 to 0. When the host driver writes this field to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state might possibly be changed when the host driver clears this field and the interrupt event might not be generated. When this field is cleared, it is set again if no card is inserted. to leave it cleared, clear the Card Removal Status Enable field in Interrupt Status Enable register. 0b - Card state unstable or inserted 1b - Card removed
6	Card insertion

Table continues on the next page...

Field	Function
CINS	This status field is set if the Card Inserted field in the Present State register changes from 0 to 1. When the host driver writes this field to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state might possibly be changed when the host driver clears this field and the interrupt event might not be generated. When this field is cleared, it is set again if a card is inserted. to leave it cleared, clear the Card Inserted Status Enable field in Interrupt Status Enable register. 0b - Card state unstable or removed 1b - Card inserted
5 BRR	Buffer read ready This status field is set if the Buffer Read Enable field, in the Present State register, changes from 0 to 1. See the Buffer Read Enable field in the Present State register for additional information. This field indicates that cmd19 is finished in tuning process. 0b - Not ready to read buffer 1b - Ready to read buffer
4 BWR	Buffer write ready This status field is set if the Buffer Write Enable field, in the Present State register, changes from 0 to 1. See the Buffer Write Enable field in the Present State register for additional information. 0b - Not ready to write buffer 1b - Ready to write buffer
3 DINT	DMA interrupt Occurs only when the internal DMA finishes the data transfer successfully. Whenever errors occur during data transfer, this field does not be set. Instead, the DMAE field is set. Either Simple DMA or ADMA finishes data transferring, this field is set. 0b - No DMA interrupt 1b - DMA interrupt is generated.
2 BGE	Block gap event If the Stop At Block Gap Request field in the Protocol Control register is set, this field is set when a read or write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this field is not set to 1. In the case of a Read Transaction: This field is set at the falling edge of the DATA Line Active Status (When the transaction is stopped at SD bus timing). The Read Wait must be supported to use this function. In the case of Write Transaction: This field is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing). 0b - No block gap event 1b - Transaction stopped at block gap
1 TC	Transfer complete This field is set when a read or write transfer is completed. In the case of a Read Transaction: This field is set at the falling edge of the Read Transfer Active Status. There are two cases in which this interrupt is generated. The first is when a data transfer is completed as specified by the data length (after the last data has been read to the host system). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request field in the Protocol Control register (after valid data has been read to the host system). In the case of a Write Transaction: This field is set at the falling edge of the DATA Line Active Status. There are two cases in which this interrupt is generated. The first is when the last data is written to the SD card as specified by the data length and the busy signal is released. The second is when data

Table continues on the next page...

Field	Function
	transfers are stopped at the block gap, by setting the Stop At Block Gap Request field in the Protocol Control register, and the data transfers are completed. (after valid data is written to the SD card and the busy signal released). In the case of a command with busy, this field is set when busy is deasserted. This field is not asserted in tuning process. 0b - Transfer does not complete 1b - Transfer complete
0 CC	Command complete This field is set when you receive the end field of the command response (except auto CMD12). See the Command Inhibit (CMD) in the Present State register. This field is not asserted in tuning process. 0b - Command not complete 1b - Command complete

10.3.6.1.15 Interrupt Status Enable (INT_STATUS_EN)

10.3.6.1.15.1 Offset

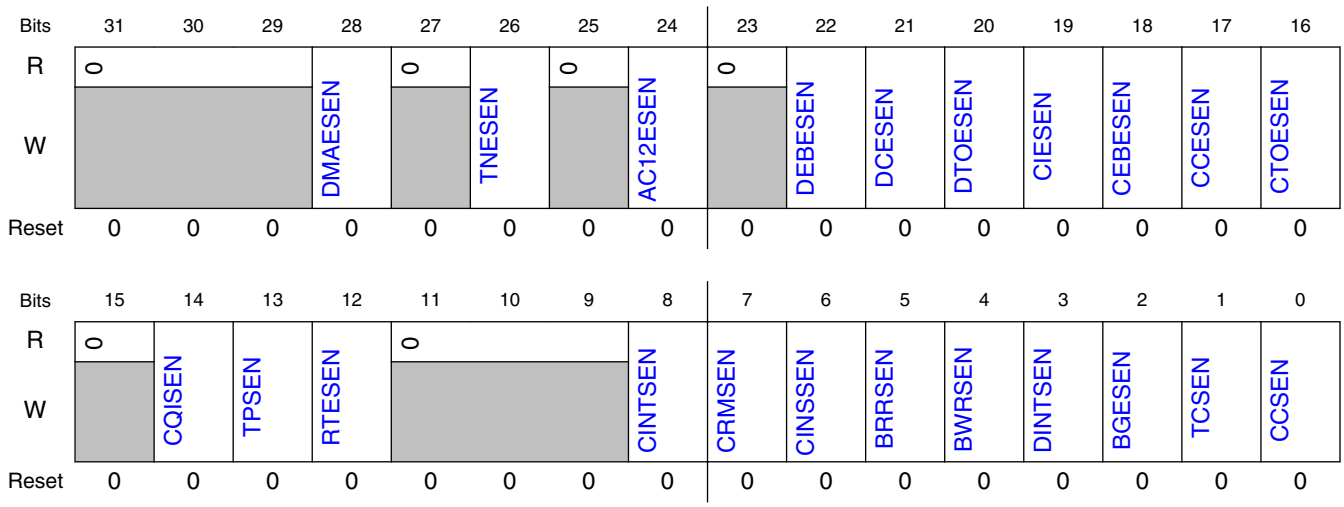
Register	Offset
INT_STATUS_EN	34h

10.3.6.1.15.2 Function

Setting any bit in this register to 1 enables the corresponding interrupt status be requested to the system. If any bit is set to 0, the corresponding interrupt request gets blocked.

- Depending on IABG field setting, uSDHC might be programmed to sample the card interrupt signal during the interrupt period and hold its value in the flip-flop. There are some delays on the Card Interrupt, asserted from the card, to the time the host system is informed.
- To detect a CMD line conflict, the host driver must set both Command Timeout Error Status Enable and Command CRC Error Status Enable to 1.

10.3.6.1.15.3 Diagram



10.3.6.1.15.4 Fields

Field	Function
31-29 —	Reserved
28 DMAESEN	DMA error status enable 0b - Masked 1b - Enabled
27 —	Reserved
26 TNESEN	Tuning error status enable 0b - Masked 1b - Enabled
25 —	Reserved
24 AC12ESEN	Auto CMD12 error status enable 0b - Masked 1b - Enabled
23 —	Reserved
22 DEBESEN	Data end bit error status enable 0b - Masked 1b - Enabled
21 DCESEN	Data CRC error status enable 0b - Masked 1b - Enabled
20 DTOESEN	Data timeout error status enable 0b - Masked 1b - Enabled

Table continues on the next page...

Field	Function
19 CIESEN	Command index error status enable 0b - Masked 1b - Enabled
18 CEBESEN	Command end bit error status enable 0b - Masked 1b - Enabled
17 CCESEN	Command CRC error status enable 0b - Masked 1b - Enabled
16 CTOESEN	Command timeout error status enable 0b - Masked 1b - Enabled
15 —	Reserved
14 CQISEN	Command queuing status enable 0b - Masked 1b - Enabled
13 TPSEN	Tuning pass status enable 0b - Masked 1b - Enabled
12 RTESEN	Re-tuning event status enable 0b - Masked 1b - Enabled
11-9 —	Reserved
8 CINTSEN	Card interrupt status enable If this field is set to 0, uSDHC clears the interrupt request to the system. The Card Interrupt detection is stopped when this field is cleared and restarted when this field is set to 1. The host driver should clear the Card Interrupt Status Enable before servicing the Card Interrupt and should set this field again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. 0b - Masked 1b - Enabled
7 CRMSEN	Card removal status enable 0b - Masked 1b - Enabled
6 CINSSEN	Card insertion status enable 0b - Masked 1b - Enabled
5 BRRSEN	Buffer read ready status enable 0b - Masked 1b - Enabled
4 BWRSEN	Buffer write ready status enable 0b - Masked 1b - Enabled
3 DINTSEN	DMA interrupt status enable 0b - Masked 1b - Enabled
2	Block gap event status enable 0b - Masked

Table continues on the next page...

Field	Function
BGESEN	1b - Enabled
1 TCSEN	Transfer complete status enable 0b - Masked 1b - Enabled
0 CCSEN	Command complete status enable 0b - Masked 1b - Enabled

10.3.6.1.16 Interrupt Signal Enable (INT_SIGNAL_EN)

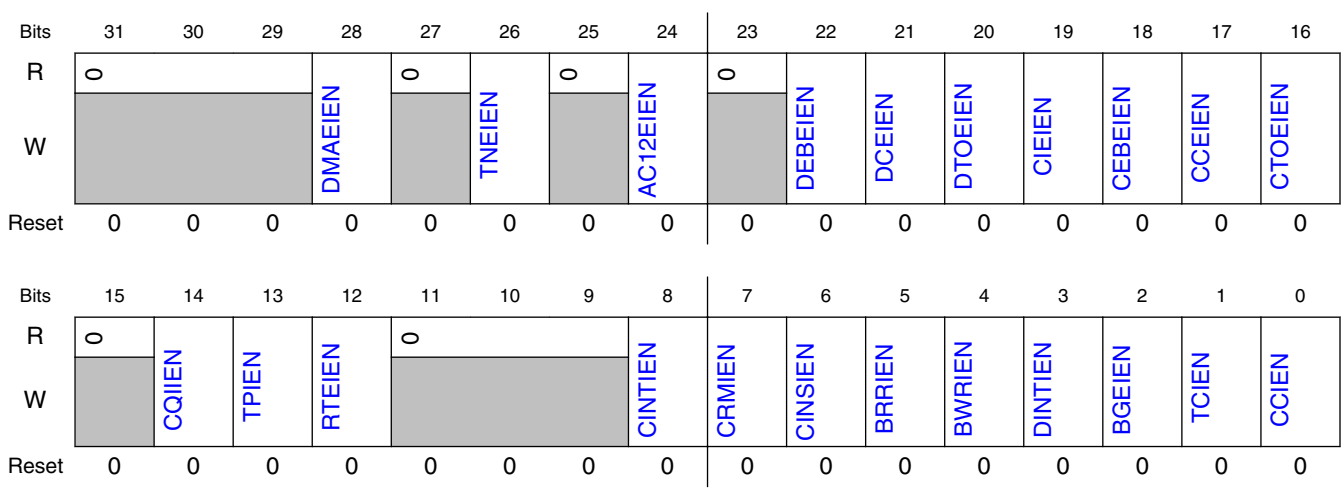
10.3.6.1.16.1 Offset

Register	Offset
INT_SIGNAL_EN	38h

10.3.6.1.16.2 Function

This register is used to select which interrupt status is indicated to the host system as the interrupt. These status fields all share the same interrupt lines. Setting any of these fields to 1 enables interrupt generation. The corresponding Status register field generates an interrupt when the corresponding interrupt signal enable field is set.

10.3.6.1.16.3 Diagram



10.3.6.1.16.4 Fields

Field	Function
31-29 —	Reserved
28 DMAEIEN	DMA error interrupt enable 0b - Masked 1b - Enable
27 —	Reserved
26 TNEIEN	Tuning error interrupt enable 0b - Masked 1b - Enabled
25 —	Reserved
24 AC12EIEN	Auto CMD12 error interrupt enable 0b - Masked 1b - Enabled
23 —	Reserved
22 DEBEIEN	Data end bit error interrupt enable 0b - Masked 1b - Enabled
21 DCEIEN	Data CRC error interrupt enable 0b - Masked 1b - Enabled
20 DTOEIEN	Data timeout error interrupt enable 0b - Masked 1b - Enabled
19 CIEIEN	Command index error interrupt enable 0b - Masked 1b - Enabled
18 CEBEIEN	Command end bit error interrupt enable 0b - Masked 1b - Enabled
17 CCEIEN	Command CRC error interrupt enable 0b - Masked 1b - Enabled
16 CTOEIEN	Command timeout error interrupt enable 0b - Masked 1b - Enabled
15 —	Reserved
14 CQIIEN	Command queuing signal enable 0b - Masked 1b - Enabled
13 TPIEN	Tuning pass interrupt enable 0b - Masked 1b - Enabled

Table continues on the next page...

Field	Function
12 RTEIEN	Re-tuning event interrupt enable 0b - Masked 1b - Enabled
11-9 —	Reserved
8 CINTIEN	Card interrupt enable 0b - Masked 1b - Enabled
7 CRMIEN	Card removal interrupt enable 0b - Masked 1b - Enabled
6 CINSIEN	Card insertion interrupt enable 0b - Masked 1b - Enabled
5 BRRIEN	Buffer read ready interrupt enable 0b - Masked 1b - Enabled
4 BWRIEN	Buffer write ready interrupt enable 0b - Masked 1b - Enabled
3 DINTIEN	DMA interrupt enable 0b - Masked 1b - Enabled
2 BGEIEN	Block gap event interrupt enable 0b - Masked 1b - Enabled
1 TCIEN	Transfer complete interrupt enable 0b - Masked 1b - Enabled
0 CCIEN	Command complete interrupt enable 0b - Masked 1b - Enabled

10.3.6.1.17 Auto CMD12 Error Status (AUTOCMD12_ERR_STATUS)

10.3.6.1.17.1 Offset

Register	Offset
AUTOCMD12_ERR_STATUS	3Ch

10.3.6.1.17.2 Function

When the Auto CMD12 Error Status field in the Status register is set, the host driver checks this register to identify what kind of error the Auto CMD12 / CMD 23 indicated. Auto CMD23 errors are indicated in field 04-01. This register is valid only when the Auto CMD12 Error status field is set.

The table below shows the relationship between the Auto CMGD12 CRC Error and the Auto CMD12 Command Timeout Error.

Table 10-52. Relationship between command CRC error and command timeout error for auto CMD12

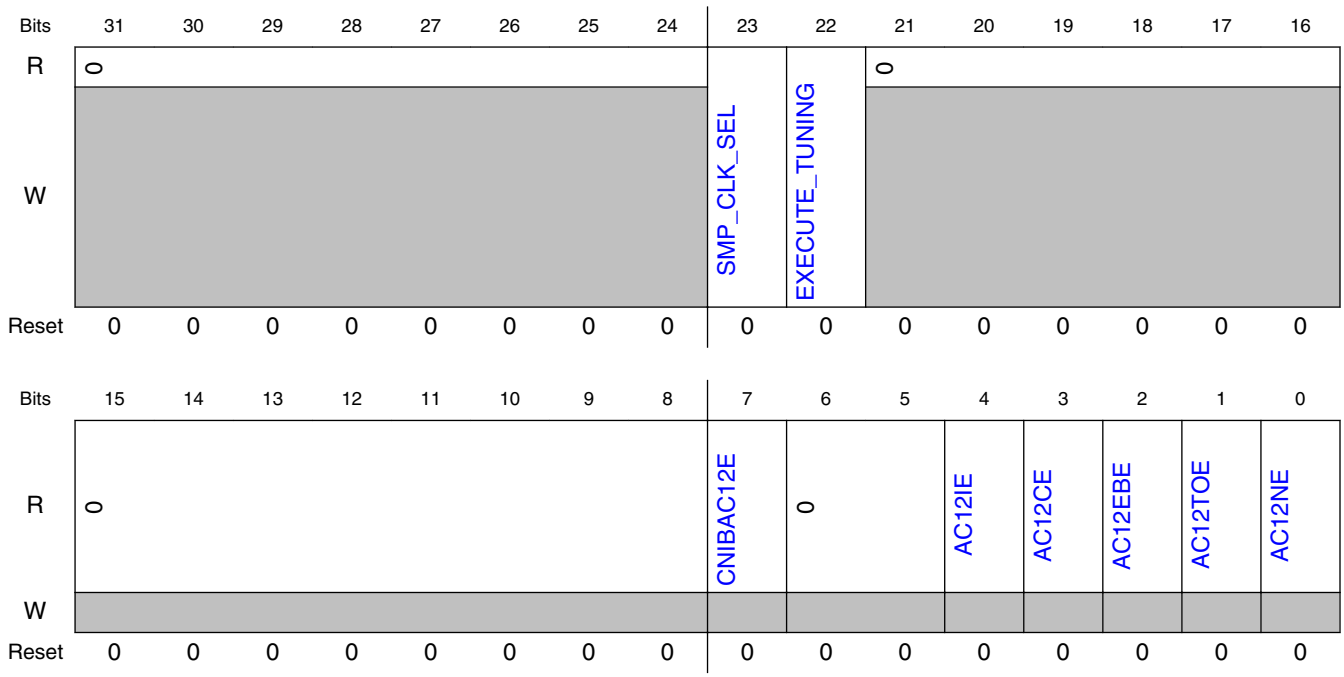
Auto CMD12 CRC error	Auto CMD12 timeout error	Type of error
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

Changes in Auto CMD12 Error Status register can be classified in three scenarios:

- When uSDHC is going to issue an Auto CMD12
  - Set field 0 to 1 if the Auto CMD12 cannot be issued due to an error in the previous command
  - Set field 0 to 0 if the Auto CMD12 is issued
- At the end field of an Auto CMD12 response
  - Check errors correspond to fields 1-4
  - Set fields 1-4 corresponding to detected errors
  - Clear fields 1-4 corresponding to detected errors
- Before reading the Auto CMD12 Error Status field 7
  - Set field 7 to 1 if there is a command that can't be issued
  - Clear field 7 if there is no command to issue

The timing for generating the Auto CMD12 Error and writing to the Command register are asynchronous. After that, field 7 is sampled when the driver is not writing to the Command register. So, it is suggested to read this register only when the AC12E field in Interrupt Status register is set. An Auto CMD12 Error Interrupt is generated when one of the error fields (0-4) is set to 1. The Command Not Issued By Auto CMD12 Error does not generate an interrupt.

10.3.6.1.17.3 Diagram



10.3.6.1.17.4 Fields

Field	Function
31-24 —	Reserved
23 SMP_CLK_SEL	Sample clock select This field is used to select sampling clock to receive CMD and DATA. This field is functional only when TUNING_CTRL[STD_TUNING_EN] is set. Otherwise, this field is reserved. This field is controlled by re-tuning procedure and is valid after the tuning procedure is complete (AUTOCMD12_ERR_STATUS[EXECUTE_TUNING]=0). A value of '1' means that tuning has been completed successfully and a value of '0' implies the tuning has failed. Writing '1' to this field is meaningless and ignored when the tuning has failed. The tuning circuit is reset by writing '0' to this field. This field can be cleared by setting AUTOCMD12_ERR_STATUS[EXECUTE_TUNING]=1. Once the tuning circuit is reset, it takes time to complete tuning sequence. Therefore, host driver should keep this field as 1 to perform the re-tuning sequence in a short time. This field cannot be changed while the Host controller is receiving response or a read data block. 0b - Fixed clock is used to sample data 1b - Tuned clock is used to sample data
22 EXECUTE_TUNING	Execute tuning This field is used to start the tuning procedure. This field is functional only when TUNING_CTRL[STD_TUNING_EN] is set. Otherwise, this field is reserved. This field is set to start tuning procedure and automatically cleared when tuning procedure is completed. The result of tuning is indicated to AUTOCMD12_ERR_STATUS[SMP_CLK_SEL]. 0b - Tuning procedure is aborted 1b - Start tuning procedure

Table continues on the next page...

Field	Function
21-8 —	Reserved
7 CNIBAC12E	Command not issued by Auto CMD12 error Setting this field to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 error (D04-D01) in this register. 0b - No error 1b - Not issued
6-5 —	Reserved
4 AC12IE	Auto CMD12 / 23 index error Occurs if the command index error occurs in response to a command. 0b - No error 1b - Error, the CMD index in response is not CMD12/23
3 AC12CE	Auto CMD12 / 23 CRC error Occurs when detecting a CRC error in the command response. 0b - No CRC error 1b - CRC error met in Auto CMD12/23 response
2 AC12EBE	Auto CMD12 / 23 end bit error Occurs when detecting that the end field of command response is 0 which should be 1. 0b - No error 1b - End bit error generated
1 AC12TOE	Auto CMD12 / 23 timeout error Occurs if no response is returned within 64 SDCLK cycles from the end field of the command. If this field is set to1, the other error status fields (2-4) have no meaning. 0b - No error 1b - Time out
0 AC12NE	Auto CMD12 not executed If memory multiple block data transfer is not started, due to a command error, this field is not set because it is not necessary to issue an Auto CMD12. Setting this field to 1 means uSDHC cannot issue the Auto CMD12 to stop a memory multiple block data transfer due to some error. If this field is set to 1, other error status fields (1-4) have no meaning. 0b - Executed 1b - Not executed

10.3.6.1.18 Host Controller Capabilities (HOST_CTRL_CAP)

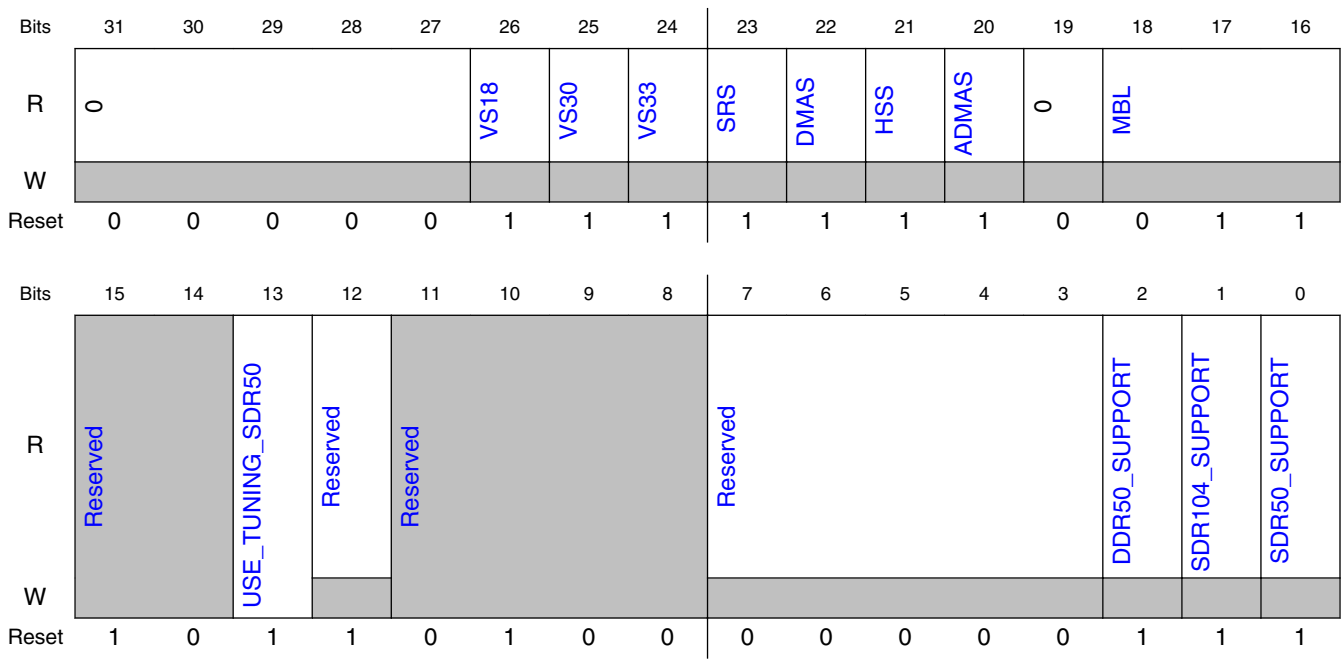
10.3.6.1.18.1 Offset

Register	Offset
HOST_CTRL_CAP	40h

10.3.6.1.18.2 Function

This register provides the host driver with information specific to uSDHC implementation. The value in this register is the power-on-reset value and does not change with a software reset.

10.3.6.1.18.3 Diagram



10.3.6.1.18.4 Fields

Field	Function
31-27 —	Reserved
26 VS18	Voltage support 1.8 V This field depends on the host system ability. 0b - 1.8 V not supported 1b - 1.8 V supported
25 VS30	Voltage support 3.0 V This field depends on the host system ability. 0b - 3.0 V not supported 1b - 3.0 V supported
24 VS33	Voltage support 3.3 V This field depends on the host system ability.

Table continues on the next page...

Field	Function
	0b - 3.3 V not supported 1b - 3.3 V supported
23 SRS	Suspend / resume support This field indicates whether uSDHC supports Suspend / Resume functionality. If this field is 0, the Suspend and Resume mechanism, as well as the read wait, are not supported, and the host driver does not issue either Suspend or Resume commands. 0b - Not supported 1b - Supported
22 DMAS	DMA support This field indicates whether uSDHC can use the internal DMA to transfer data between system memory and the data buffer directly. 0b - DMA not supported 1b - DMA supported
21 HSS	High speed support This field indicates whether uSDHC supports High Speed mode and the host system can supply a SD clock frequency from 25 MHz to 50 MHz. 0b - High speed not supported 1b - High speed supported
20 ADMAS	ADMA support This field indicates whether uSDHC supports the ADMA feature. 0b - Advanced DMA not supported 1b - Advanced DMA supported
19 —	Reserved
18-16 MBL	Max block length This field indicates the maximum block size that the host driver can read and write to the buffer in uSDHC. The buffer transfers block size without wait cycles. 000b - 512 bytes 001b - 1024 bytes 010b - 2048 bytes 011b - 4096 bytes
15-14 —	Reserved
13 USE_TUNING_SDR50	Use Tuning for SDR50 This field is used to enable tuning for SDR50 mode 0b - SDR50 does not support tuning 1b - SDR50 supports tuning
12 —	Reserved
11-8 —	Reserved
7-3 —	Reserved
2	DDR50 support

Table continues on the next page...

Field	Function
DDR50_SUPPORT	This field indicates support of DDR50 mode.
1 SDR104_SUPPORT	SDR104 support This field indicates support of SDR104 mode.
0 SDR50_SUPPORT	SDR50 support This field indicates support of SDR50 mode.

10.3.6.1.19 Watermark Level (WTMK_LVL)

10.3.6.1.19.1 Offset

Register	Offset
WTMK_LVL	44h

10.3.6.1.19.2 Function

This register indicates configurability of write and read watermark levels (FIFO threshold). Their value can range from 1 to 128 words.

10.3.6.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			Reserved					WR_WML							
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			Reserved					RD_WML							
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0

10.3.6.1.19.4 Fields

Field	Function
31-29	Reserved

Table continues on the next page...

Field	Function
—	
28-24 —	Reserved
23-16 WR_WML	Write watermark level This field indicates the number of words used as the watermark level (FIFO threshold) in a DMA write operation. Also, the number of words as a sequence of write bursts in back-to-back mode. The maximum legal value for the write watermark level is 128.When using internal DMA, WR_WML can be set any value between 1 and 128. To get max AXI throughput, we recommend to use integer multiple of 16.
15-13 —	Reserved
12-8 —	Reserved
7-0 RD_WML	Read watermark level This field indicates the number of words used as the watermark level (FIFO threshold) in a DMA read operation. Also, the number of words as a sequence of read bursts in back-to-back mode. The maximum legal value for the read water mark level is 128.When using internal DMA mode, RD_WML must be set to the value less than 16. To get max AXI throughput, recommend to use 16.

10.3.6.1.20 Mixer Control (MIX_CTRL)

10.3.6.1.20.1 Offset

Register	Offset
MIX_CTRL	48h

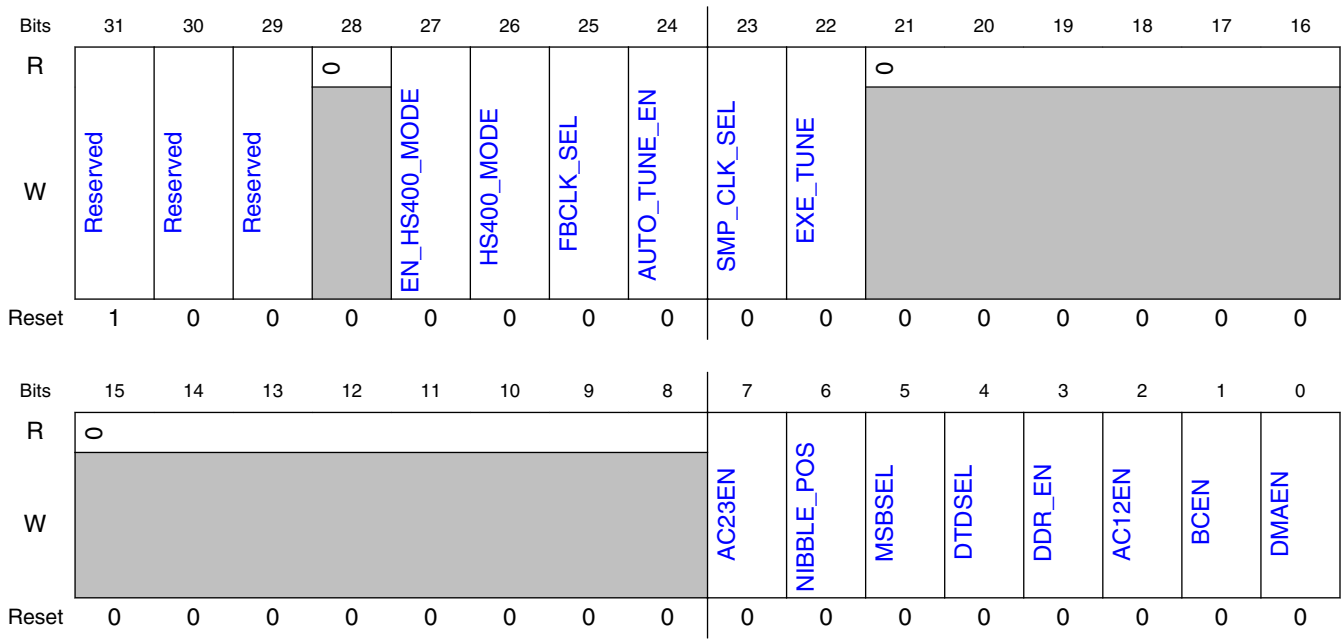
10.3.6.1.20.2 Function

This register is used to DMA and data transfer. To prevent data loss, the software should check if data transfer is active before writing this register. These fields are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN, and DMAEN.

Table 10-53. Transfer type register setting for various transfer types

Multi/single block select	Block count enable	Block count	Function
0	Do not care	Do not care	Single transfer
1	0	Do not care	Infinite transfer
1	1	Positive number	Multiple transfer
1	1	Zero	No data transfer

10.3.6.1.20.3 Diagram



10.3.6.1.20.4 Fields

Field	Function
31 —	Reserved Always write as 1.
30 —	Reserved Always write as 0.
29 —	Reserved Always write as 0.
28 —	Reserved
27 EN_HS400_MODE	Enable enhance HS400 mode Enhance HS400 enable
26 HS400_MODE	Enable HS400 mode HS400 Enable
25 FBCLK_SEL	Feedback clock source selection (Only used for SD3.0, SDR104 mode and EMMC HS200 mode) 0b - Feedback clock comes from the loopback CLK 1b - Feedback clock comes from the ipp_card_clk_out
24 AUTO_TUNE_EN	Auto tuning enable (Only used for SD3.0, SDR104 mode and and EMMC HS200 mode) 0b - Disable auto tuning 1b - Enable auto tuning
23	Clock selection

Table continues on the next page...

Field	Function
SMP_CLK_SEL	When TUNING_CTRL[STD_TUNING_EN] is 0, this field is used to select Tuned clock or Fixed clock to sample data / cmd (Only used for SD3.0, SDR104 mode and EMMC HS200 mode) 0b - Fixed clock is used to sample data / cmd 1b - Tuned clock is used to sample data / cmd
22 EXE_TUNE	Execute tuning: (Only used for SD3.0, SDR104 mode and EMMC HS200 mode) When TUNING_CTRL[STD_TUNING_EN] is 0, this field is set to 1 to indicate the host driver is starting tuning procedure. Tuning procedure is aborted by writing 0. 0b - Not tuned or tuning completed 1b - Execute tuning
21-8 —	Reserved
7 AC23EN	Auto CMD23 enable When this field is set to 1, the host controller issues a CMD23 automatically before issuing a command specified in the Command Register.
6 NIBBLE_POS	Nibble position indication This field indicates the nibble position in the DDR 4-bit mode. 0- the sequence is 'odd high nibble -> even high nibble -> odd low nibble -> even low nibble'; 1- the sequence is 'odd high nibble -> odd low nibble -> even high nibble -> even low nibble'.
5 MSBSEL	Multi / Single block select This field enables multiple block DATA line data transfers. For any other commands, this field can be set to 0. If this field is 0, it is not necessary to set the Block Count register. See Command Transfer Type (CMD_XFR_TYP). 0b - Single block 1b - Multiple blocks
4 DTDSEL	Data transfer direction select This field defines the direction of DATA line data transfers. The field is set to 1 by the host driver to transfer data from the SD card to uSDHC and is set to 0 for all other commands. If a CMD with the response data size cannot be divided by 64 bytes, is given without a STOP CMD, and is followed by a software reset, then you must write a 0 to DTDSEL before inserting a software reset. This is necessary to avoid any issue during the SD card reinitialization. Read multiple blocks will not be affected as it is required to have a STOP CMD. 0b - Write (Host to card) 1b - Read (Card to host)
3 DDR_EN	Dual data rate mode selection
2 AC12EN	Auto CMD12 enable Multiple block transfers for memory require a CMD12 to stop the transaction. When this field is set to 1, uSDHC issues a CMD12 automatically when the last block transfer has completed. The host driver is not set this field to issue commands that do not require CMD12 to stop a multiple block data transfer. In particular, secure commands defined in File Security Specification (see reference list) do not require CMD12. In single block transfer, uSDHC ignores this field no matter it is set or not. 0b - Disable 1b - Enable
1 BCEN	Block count enable

Table continues on the next page...

Field	Function
	This field is used to enable the Block Count register, which is only relevant for multiple block transfers. When this field is 0, the internal counter for block is disabled, which is useful in executing an infinite transfer. 0b - Disable 1b - Enable
0 DMAEN	DMA enable This field enables DMA functionality. If this field is set to 1, a DMA operation begins when the host driver sets the DPSEL field of this register. Whether the simple DMA or the advanced DMA is active depends on the DMA Select field of the Protocol Control register. 0b - Disable 1b - Enable

10.3.6.1.21 Force Event (FORCE_EVENT)

10.3.6.1.21.1 Offset

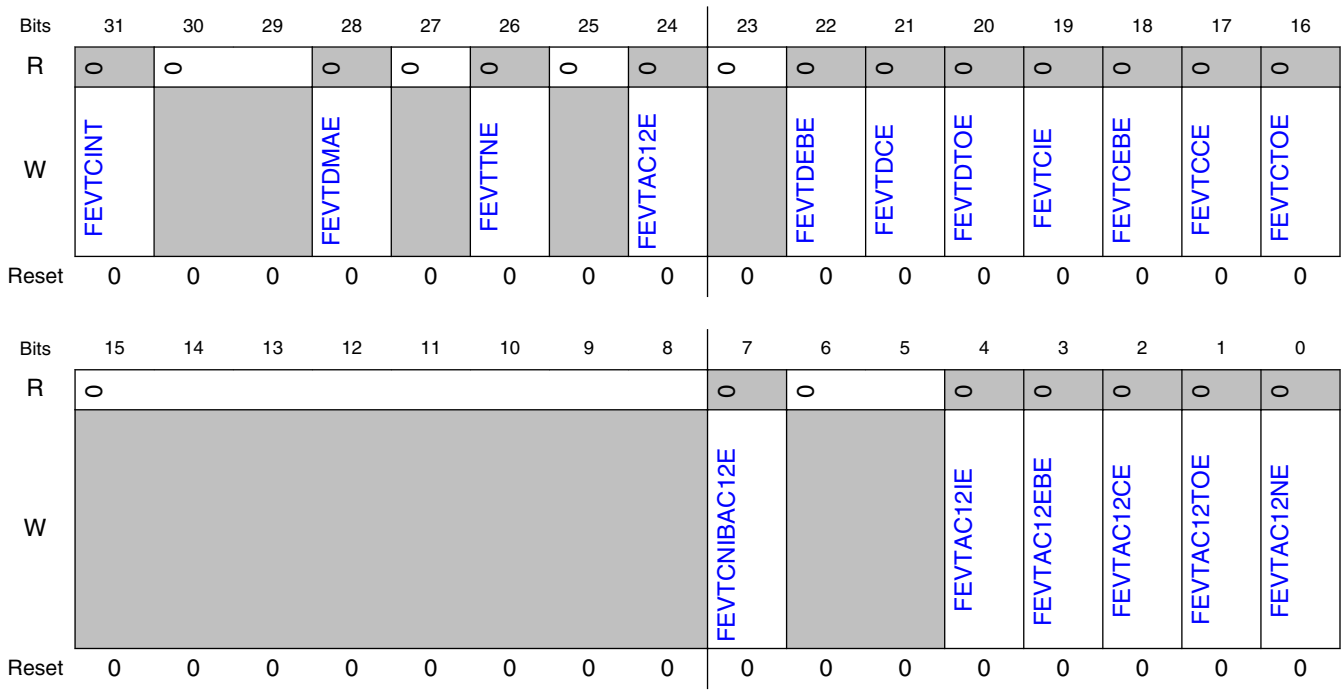
Register	Offset
FORCE_EVENT	50h

10.3.6.1.21.2 Function

This register is not a physically implemented register. Rather, it is an address at which the Interrupt Status register can be written if the corresponding field of the Interrupt Status Enable Register is set. This register is a write only register and writing 0 to it has no effect. Writing 1 to this register sets the corresponding field of Interrupt Status register. A read from this register always results in 0's. to change the corresponding status fields in the Interrupt Status register, make sure to set IPGEN field in System Control Register so that peripheral clock is always active.

Forcing a card interrupt generates a short pulse on the DATA1 line, and the driver might treat this interrupt as a normal interrupt. The interrupt service routine might skip polling the card interrupt factor as the interrupt is self cleared.

10.3.6.1.21.3 Diagram



10.3.6.1.21.4 Fields

Field	Function
31 FEVTCINT	Force event card interrupt Writing 1 to this field generates a short low-level pulse on the internal DATA1 line, as if a self-clearing interrupt was received from the external card. If enabled, the CINT field is set and the interrupt service routine might treat this interrupt as a normal interrupt from the external card.
30-29 —	Reserved
28 FEVTDMAE	Force event DMA error Forces the DMAE field of Interrupt Status register to be set
27 —	Reserved
26 FEVTTNE	Force tuning error Forces the TNE field of Interrupt Status register to be set
25 —	Reserved
24 FEVTAC12E	Force event Auto Command 12 error Forces the AC12E field of Interrupt Status register to be set
23 —	Reserved

Table continues on the next page...

Field	Function
22 FEVTDEBE	Force event data end bit error Forces the DEBE field of Interrupt Status register to be set
21 FEVTDCE	Force event data CRC error Forces the DCE field of Interrupt Status register to be set
20 FEVTDTOE	Force event data time out error Force the DTOE field of Interrupt Status register to be set
19 FEVTCIE	Force event command index error Forces the CCE field of Interrupt Status register to be set
18 FEVTCEBE	Force event command end bit error Forces the CEBE field of Interrupt Status register to be set
17 FEVTCCE	Force event command CRC error Forces the CCE field of Interrupt Status register to be set
16 FEVTCTOE	Force event command time out error Forces the CTOE field of Interrupt Status register to be set
15-8 —	Reserved
7 FEVTCNIBAC12E	Force event command not executed by Auto Command 12 error Forces the CNIBAC12E field in the Auto Command12 Error Status register to be set
6-5 —	Reserved
4 FEVTAC12IE	Force event Auto Command 12 index error Forces the AC12IE field in the Auto Command12 Error Status register to be set
3 FEVTAC12EBE	Force event Auto Command 12 end bit error Forces the AC12EBE field in the Auto Command12 Error Status register to be set
2 FEVTAC12CE	Force event auto command 12 CRC error Forces the AC12CE field in the Auto Command12 Error Status register to be set
1 FEVTAC12TOE	Force event auto command 12 time out error Forces the AC12TOE field in the Auto Command12 Error Status register to be set
0 FEVTAC12NE	Force event auto command 12 not executed Forces the AC12NE field in the Auto Command12 Error Status register to be set

10.3.6.1.22 ADMA Error Status (ADMA_ERR_STATUS)

10.3.6.1.22.1 Offset

Register	Offset
ADMA_ERR_STATUS	54h

10.3.6.1.22.2 Function

When an ADMA Error Interrupt has occurred, the ADMA error sates field in this register holds the ADMA state and the ADMA System Address register holds the address around the error descriptor.

For recovering from this error, the host driver requires the ADMA state to identify the error descriptor address as follows:

- ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address.
- ST_FDS: Current location set in the ADMA System Address register is the error descriptor address.
- ST_CADR: This state is never set because it only increments the descriptor pointer and does not generate an ADMA error.
- ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address.

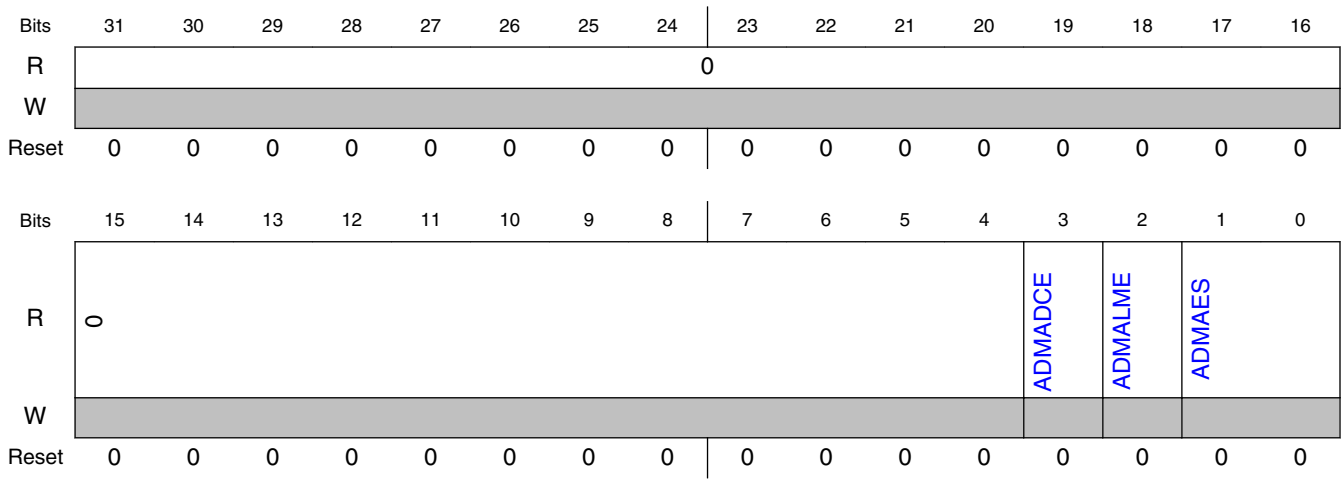
In case of a write operation, the host driver should use the ACMD22 to get the number of the written block, rather than using this information, because unwritten data might exist in the host controller.

The host controller generates the ADMA Error Interrupt when it detects invalid descriptor data (Valid=0) in the ST_FDS state. The host driver can distinguish this error by reading the Valid field of the error descriptor.

Table 10-54. ADMA error state coding

D01-D00	ADMA error state (when error has occurred)	Contents of ADMA System Address register
00	ST_STOP (Stop DMA)	Holds the address of the next executable descriptor command
01	ST_FDS (Fetch descriptor)	Holds the valid descriptor address
10	ST_CADR (Change address)	No ADMA error is generated
11	ST_TFR (Transfer data)	Holds the address of the next executable descriptor command

10.3.6.1.22.3 Diagram



10.3.6.1.22.4 Fields

Field	Function
31-4 —	Reserved
3 ADMADCE	ADMA descriptor error This error occurs when invalid descriptor fetched by ADMA. 0b - No error 1b - Error
2 ADMALME	ADMA length mismatch error This error occurs in the following two cases: • While the block count enable is being set, the total data length specified by the descriptor table is different from that specified by the block count and block length. • Total data length cannot be divided by the block length. 0b - No error 1b - Error
1-0 ADMAES	ADMA error state (when ADMA error is occurred) This field indicates the state of the ADMA when an error has occurred during an ADMA data transfer. See ADMA Error Status (ADMA_ERR_STATUS) for more details.

10.3.6.1.23 ADMA System Address (ADMA_SYS_ADDR)

10.3.6.1.23.1 Offset

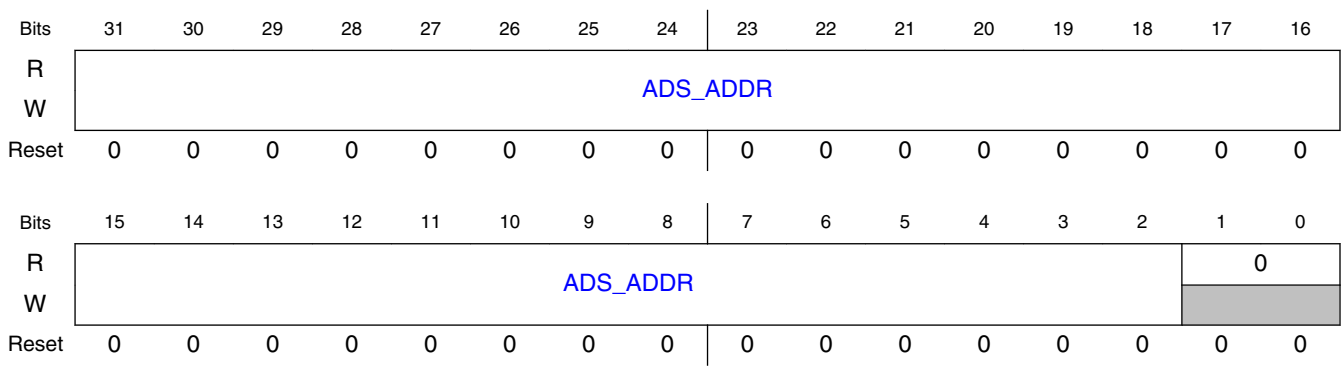
Register	Offset
ADMA_SYS_ADDR	58h

10.3.6.1.23.2 Function

This register holds the word address of the executing command in the Descriptor table. At the start of ADMA, the host driver sets the start address of the Descriptor table. The ADMA engine increments this register address whenever fetching a Descriptor command. When the ADMA is stopped at the Block Gap, this register indicates the address of the next executable Descriptor command. When the ADMA Error Interrupt is generated, this register holds the valid Descriptor address depending on the ADMA state. The lower 2 bits of this register is tied to '0' so the ADMA address is always word aligned.

Because this register supports dynamic address reflecting, when TC field is set, it automatically alters the value of internal address counter, so the software cannot change this register when TC field is set. Such restriction is also listed in Software restrictions.

10.3.6.1.23.3 Diagram



10.3.6.1.23.4 Fields

Field	Function
31-2 ADS_ADDR	ADMA system address This field contains the physical system memory address used for ADMA transfers.
1-0 —	Reserved

10.3.6.1.24 DLL (Delay Line) Control (DLL_CTRL)

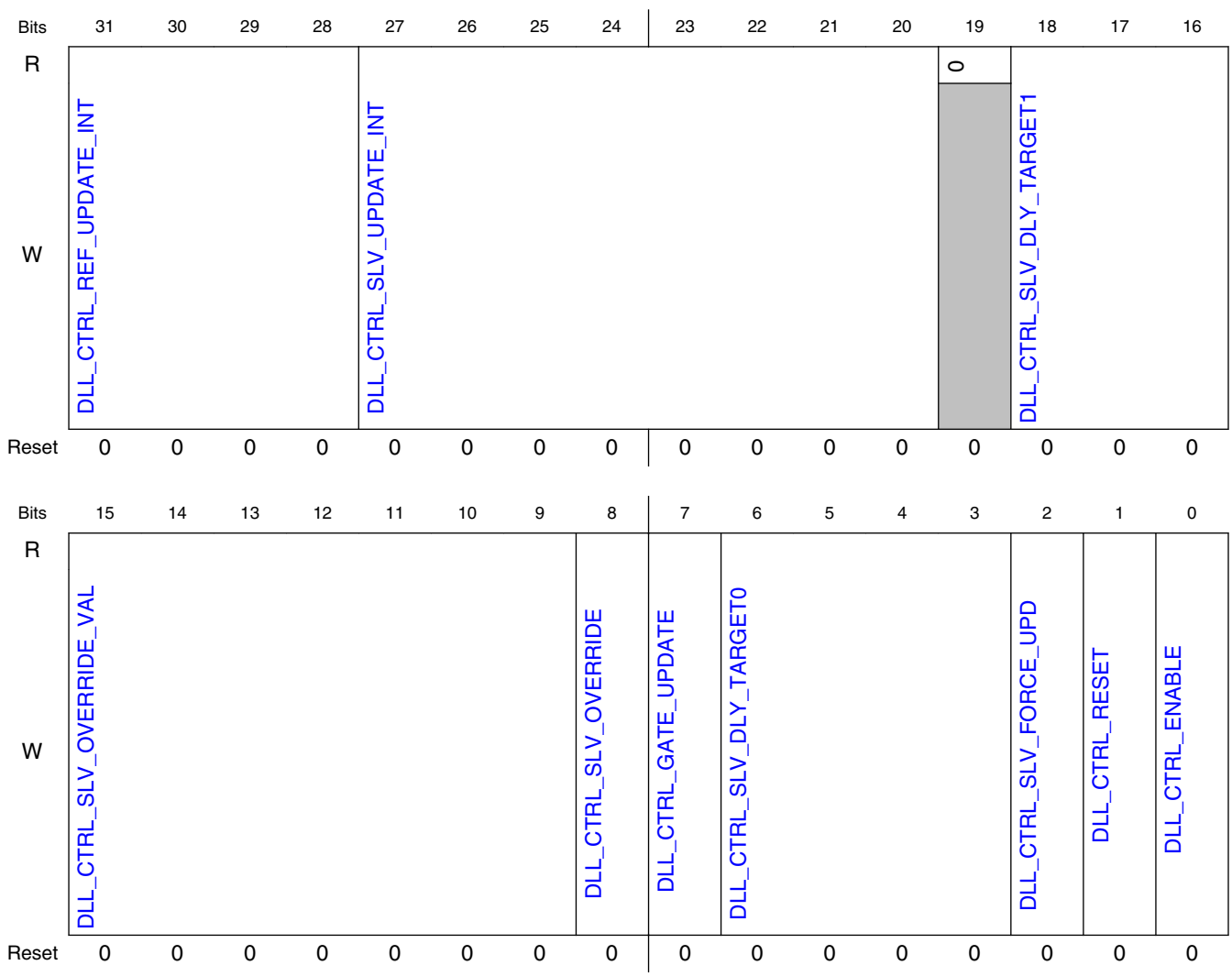
10.3.6.1.24.1 Offset

Register	Offset
DLL_CTRL	60h

10.3.6.1.24.2 Function

This register contains control fields for DLL.

10.3.6.1.24.3 Diagram



10.3.6.1.24.4 Fields

Field	Function
31-28 DLL_CTRL_REF_UPDATE_INT	DLL control loop update interval The interval cycle is (2 + REF_UPDATE_INT) * REF_CLOCK. By default, the DLL control loop updates every two REF_CLOCK cycles. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that might effect the delay (such as voltage and temperature)
27-20 DLL_CTRL_SLV_UPDATE_INT	Slave delay line update interval If default 0 is used, it means 256 cycles of REF_CLOCK. A value of 0x0f results in 15 cycles and so on. Note that software can always cause an update of the slave-delay line using the SLV_FORCE_UPDATE register. Note that the slave delay line also updates automatically when the reference DLL transitions to a locked state (from an un-locked state).
19 —	Reserved
18-16 DLL_CTRL_SLV_DLY_TARGET1	DLL slave delay target1 See DLL_CTRL_SLV_DLY_TARGET0 below.
15-9 DLL_CTRL_SLV_OVERRIDE_VAL	DLL slave override val When SLV_OVERRIDE = 1, this field is used to select 1 of 128 physical taps manually. A value of 0 selects tap 1, and a value of 0x7f selects tap 128.
8 DLL_CTRL_SLV_OVERRIDE	DLL slave override Set this field to 1 to Enable manual override for slave delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to be enabled using the ENABLE field. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE = 0
7 DLL_CTRL_GATE_UPDATE	DLL gate update Set this field to 1 to prevent the DLL from updating (because when clock_in exists, glitches might appear during DLL updates). This field might be used by software if such a condition occurs. Clear the bit to 0 to allow the DLL to update automatically.
6-3 DLL_CTRL_SLV_DLY_TARGET0	DLL slave delay target0 The delay target for uSDHC loopback read clock can be programmed in 1/16th increments of an ref_clock half-period. The delay is (({DLL_CTRL_SLV_DLY_TARGET1,DLL_CTRL_SLV_DLY_TARGET0} +1)* REF_CLOCK / 2) / 16 So the input read-clock can be delayed relative input data from (REF_CLOCK / 2) / 16 to REF_CLOCK * 4. **NOTE:** For the restrictions of delay cell implementation, the delay target must be set between REF_CLOCK/16 and REF_CLOCK*2 when REF_CLOCK is running at 200 MHz. When REF_CLOCK frequency is slower than 100 MHz, the maximum delay target might not reach REF_CLOCK*2.
2 DLL_CTRL_SLV_FORCE_UPD	DLL slave delay line Setting this field to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line updates automatically based on the SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered). Be sure to use it when uSDHC is idle. This function might not work when uSDHC is working on data / cmd / response.
1	DLL reset

Table continues on the next page...

Field	Function
DLL_CTRL_RESET	Setting this field to 1 force a reset on DLL. This causes the DLL to lose lock and re-calibrate to detect an REF_CLOCK half period phase shift. This signal is used by the DLL as edge-sensitive, so to create a subsequent reset, RESET must be taken low and then asserted again.
0 DLL_CTRL_ENABLE	DLL and delay chain Set this field to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. Note that using the slave delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled.

10.3.6.1.25 DLL Status (DLL_STATUS)

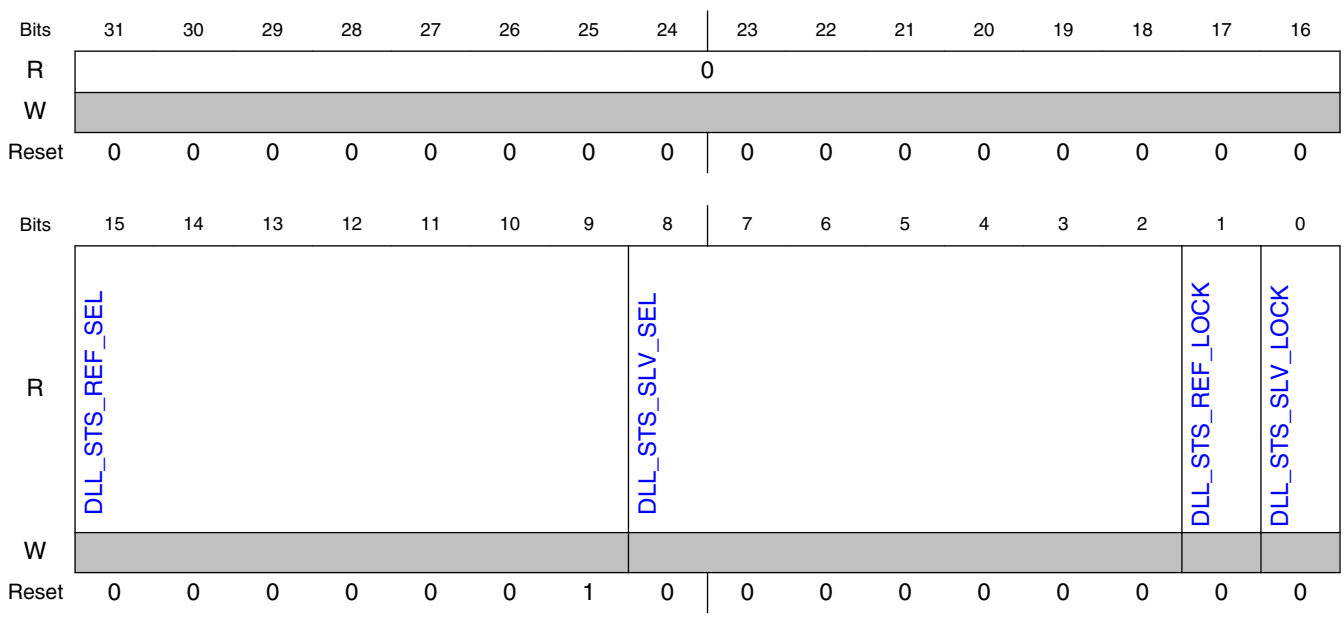
10.3.6.1.25.1 Offset

Register	Offset
DLL_STATUS	64h

10.3.6.1.25.2 Function

This register contains the DLL status information. All fields are read only and reads the same as the power-reset value.

10.3.6.1.25.3 Diagram



10.3.6.1.25.4 Fields

Field	Function
31-16 —	Reserved
15-9 DLL_STS_REF_SEL	Reference delay line select taps This is encoded by 7 fields for 127 taps.
8-2 DLL_STS_SLV_SEL	Slave delay line select status This is the instant value generated from reference chain. Because the reference chain can only be updated when REF_CLOCK is detected, this value should be the right value to be updated when the reference is locked.
1 DLL_STS_REF_LOCK	Reference DLL lock status This signifies that the DLL has detected and locked to a half-phase ref_clock shift, allowing the slave delay-line to perform programmed clock delays
0 DLL_STS_SLV_LOCK	Slave delay-line lock status This signifies that a valid calibration has been set to the slave-delay line and that the slave-delay line is implementing the programmed delay value

10.3.6.1.26 CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS)

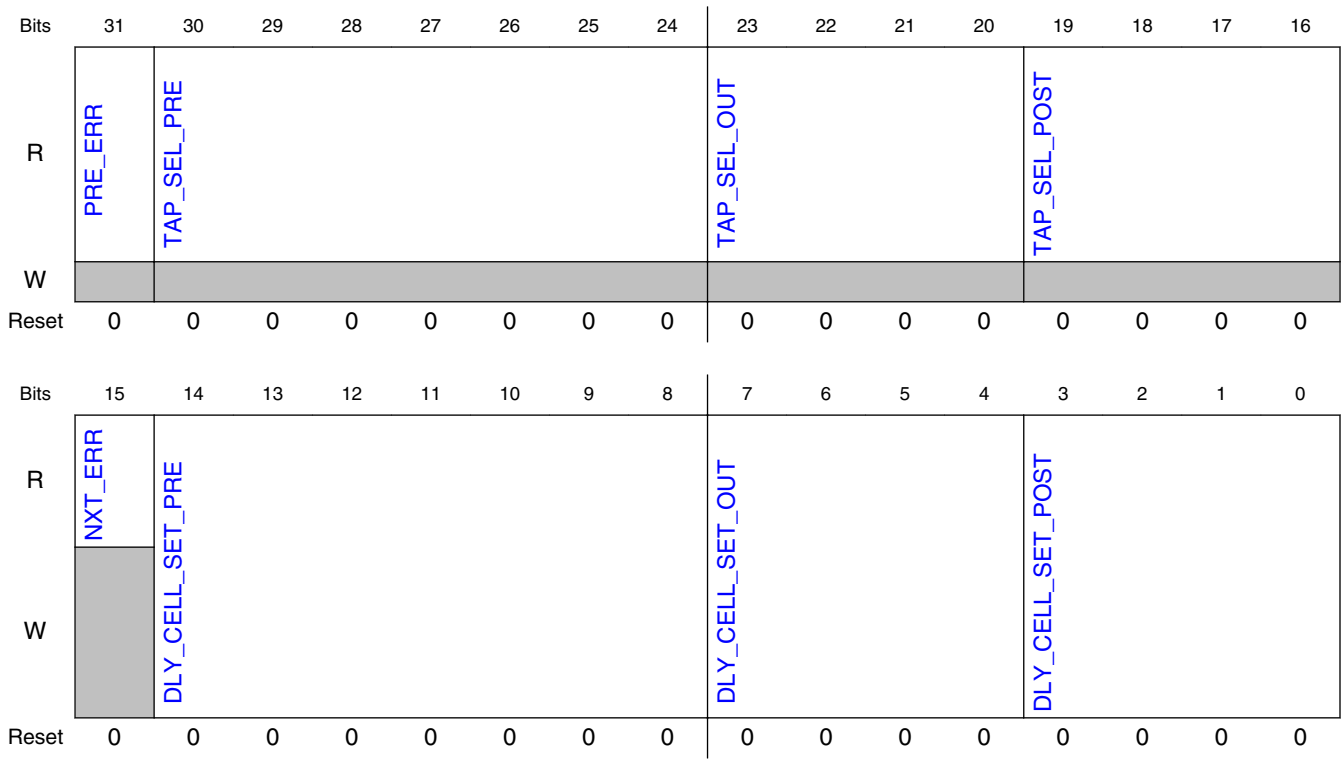
10.3.6.1.26.1 Offset

Register	Offset
CLK_TUNE_CTRL_STATUS	68h

10.3.6.1.26.2 Function

This register contains the Clock Tuning Control status information. All fields are read only and reads the same as the power-reset value. This register is added to support SD3.0 UHS-I SDR104 mode and EMMC HS200 mode.

10.3.6.1.26.3 Diagram



10.3.6.1.26.4 Fields

Field	Function
31 PRE_ERR	PRE error PRE error which means the number of delay cells added on the feedback clock is too small. It is valid only when SMP_CLK_SEL of Mix control register (bit23 of 0x48) is enabled.
30-24 TAP_SEL_PRE	TAP_SEL_PRE Reflects the number of delay cells added on the feedback clock between the feedback clock and CLK_PRE. When AUTO_TUNE_EN (bit24 of 0x48) is disabled, TAP_SEL_PRE is always equal to DLY_CELL_SET_PRE. When AUTO_TUNE_EN (bit24 of 0x48) is enabled, TAP_SEL_PRE is updated automatically according to the status of the auto tuning circuit to adjust the sample clock phase.
23-20 TAP_SEL_OUT	Delay cells added on the feedback clock between CLK_PRE and CLK_OUT Reflects the number of delay cells added on the feedback clock between CLK_PRE and CLK_OUT.
19-16 TAP_SEL_POST	Delay cells added on the feedback clock between CLK_OUT and CLK_POST Reflects the number of delay cells added on the feedback clock between CLK_OUT and CLK_POST.
15 NXT_ERR	NXT error NXT error which means the number of delay cells added on the feedback clock is too large. It's valid only when SMP_CLK_SEL of Mix control register (bit23 of 0x48) is enabled.

Table continues on the next page...

Field	Function
14-8 DLY_CELL_SET_PRE	delay cells on the feedback clock between the feedback clock and CLK_PRE Set the number of delay cells on the feedback clock between the feedback clock and CLK_PRE.
7-4 DLY_CELL_SET_OUT	Delay cells on the feedback clock between CLK_PRE and CLK_OUT Set the number of delay cells on the feedback clock between CLK_PRE and CLK_OUT.
3-0 DLY_CELL_SET_POST	Delay cells on the feedback clock between CLK_OUT and CLK_POST Set the number of delay cells on the feedback clock between CLK_OUT and CLK_POST.

10.3.6.1.27 Strobe DLL control (STROBE_DLL_CTRL)

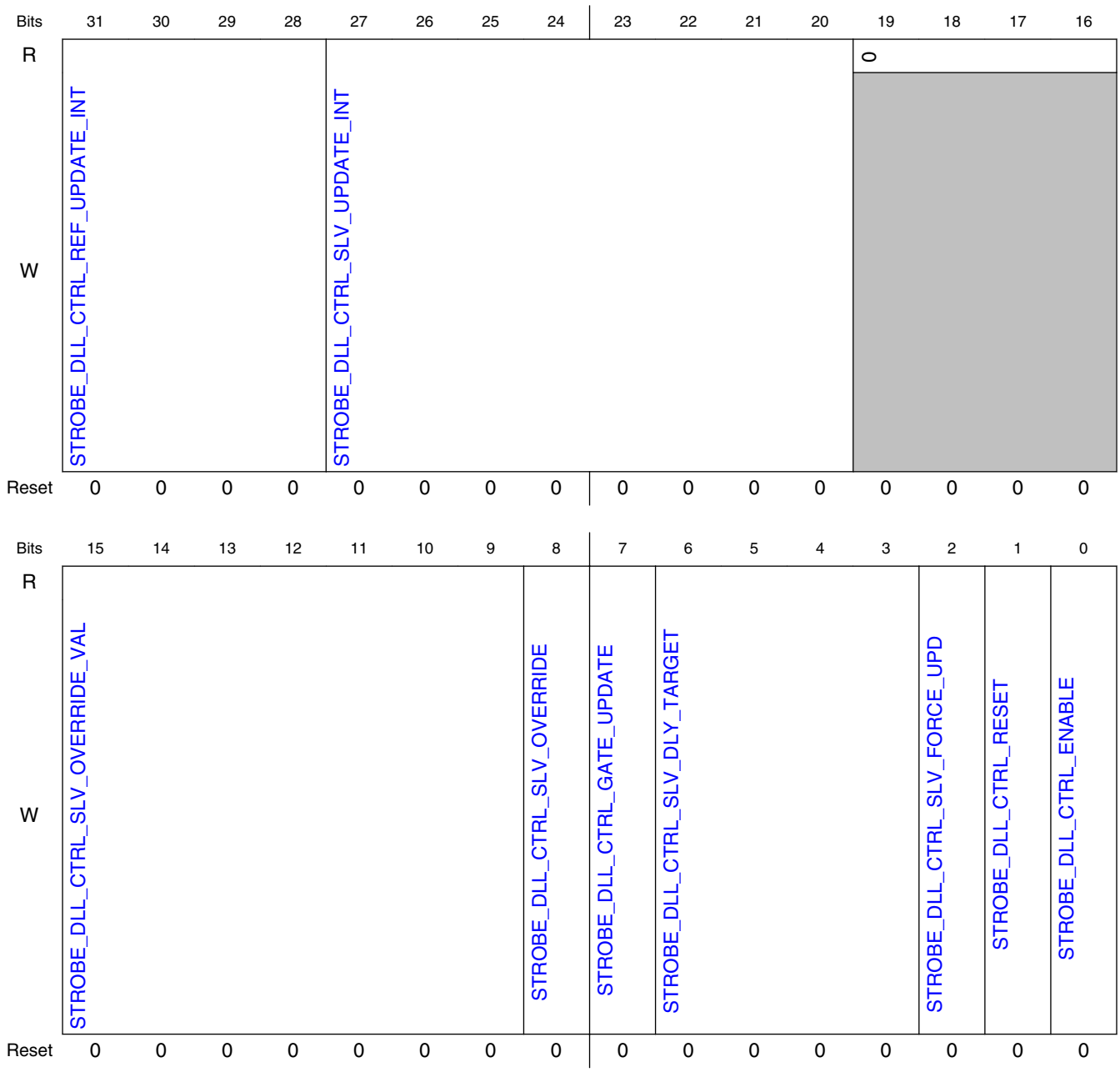
10.3.6.1.27.1 Offset

Register	Offset
STROBE_DLL_CTRL	70h

10.3.6.1.27.2 Function

This register contains the strobe DLL control information.

10.3.6.1.27.3 Diagram



10.3.6.1.27.4 Fields

Field	Function
31-28 STROBE_DLL_CTRL_REF_UPDATE_INT	Strobe DLL control reference update interval The interval cycle is (2 + STROBE_REF_UPDATE_INT) * STROBE_REF_CLOCK. By default, the DLL control loop updates every two STROBE_REF_CLOCK cycles.

Table continues on the next page...

Field	Function
	NOTE: Increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that might effect the delay (such as voltage and temperature)
27-20 STROBE_DLL_CTRL_SLV_UPDATE_INT	Strobe DLL control slave update interval Slave delay line update interval. If default 0 is used, it means 256 cycles of STROBE_REF_CLOCK. A value of 0x0F results in 15 cycles and so on. **NOTE:** software can always cause an update of the slave-delay line using the STROBE_SLV_FORCE_UPDATE register. The slave delay line also updates automatically when the reference DLL transitions to a locked state (from an un-locked state).
19-16 —	This field is reserved. This read-only field is reserved and always has the value 0.
15-9 STROBE_DLL_CTRL_SLV_OVERRIDE_VAL	Strobe DLL control slave Override value When STROBE_SLV_OVERRIDE = 1, this field is used to manually select one of 128 physical taps. A value of 0 selects tap 1, and a value of 0x7F selects tap 128.
8 STROBE_DLL_CTRL_SLV_OVERRIDE	Strobe DLL control slave override Set this field to 1 to Enable manual override for slave delay chain using STROBE_SLV_OVERRIDE_VAL; set this field to 0 to disable manual override. This feature does not require the DLL to be enabled using the ENABLE field. In fact to reduce power, if STROBE_SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE = 0.
7 STROBE_DLL_CTRL_GATE_UPDATE	Strobe DLL control gate update Set this field to 1 to prevent the DLL from updating (because when STROBE_CLOCK_IN exists, glitches might appear during DLL updates). This field can be used by software if such a condition occurs. Clear the field to 0 to allow the DLL to update automatically.
6-3 STROBE_DLL_CTRL_SLV_DLY_TARGET	Strobe DLL Control Slave Delay Target The delay target for uSDHC loopback read clock can be programmed in 1/16th increments of an STROBE_REF_CLOCK half-period. The delay is ((STROBE_DLL_CTRL_SLV_DLY_TARGET + 1) * STROBE_REF_CLOCK / 2) / 16, So the input read-clock can be delayed relative input data from (STROBE_REF_CLOCK / 2) / 16 to (STROBE_REF_CLOCK *4)/16. **NOTE:** For the restrictions of delay cell implementation, the minimum delay target that can be set is REF_CLOCK/16.
2 STROBE_DLL_CTRL_SLV_FORCE_UPD	Strobe DLL control slave force updated Setting this field to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line should automatically update the STROBE_SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if STROBE_SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered). Be sure to use it when uSDHC is idle. This function might not work when uSDHC is working on data / cmd / response.
1 STROBE_DLL_CTRL_RESET	Strobe DLL control reset Setting this field to 1 to force a reset on DLL. This causes the DLL to lose lock and re-calibrate to detect an REF_CLOCK half period phase shift. This signal is used by the DLL as edge-sensitive, to create a subsequent reset, RESET must be taken low and then asserted again.
0 STROBE_DLL_CTRL_ENABLE	Strobe DLL control enable Set this field to 1 to enable the DLL and delay chain; otherwise, set to 0 to bypasses DLL. **NOTE:** Using the slave delay line override feature with STROBE_SLV_OVERRIDE and STROBE_SLV_OVERRIDE VAL, the DLL does not need to be enabled.

10.3.6.1.28 Strobe DLL status (STROBE_DLL_STATUS)

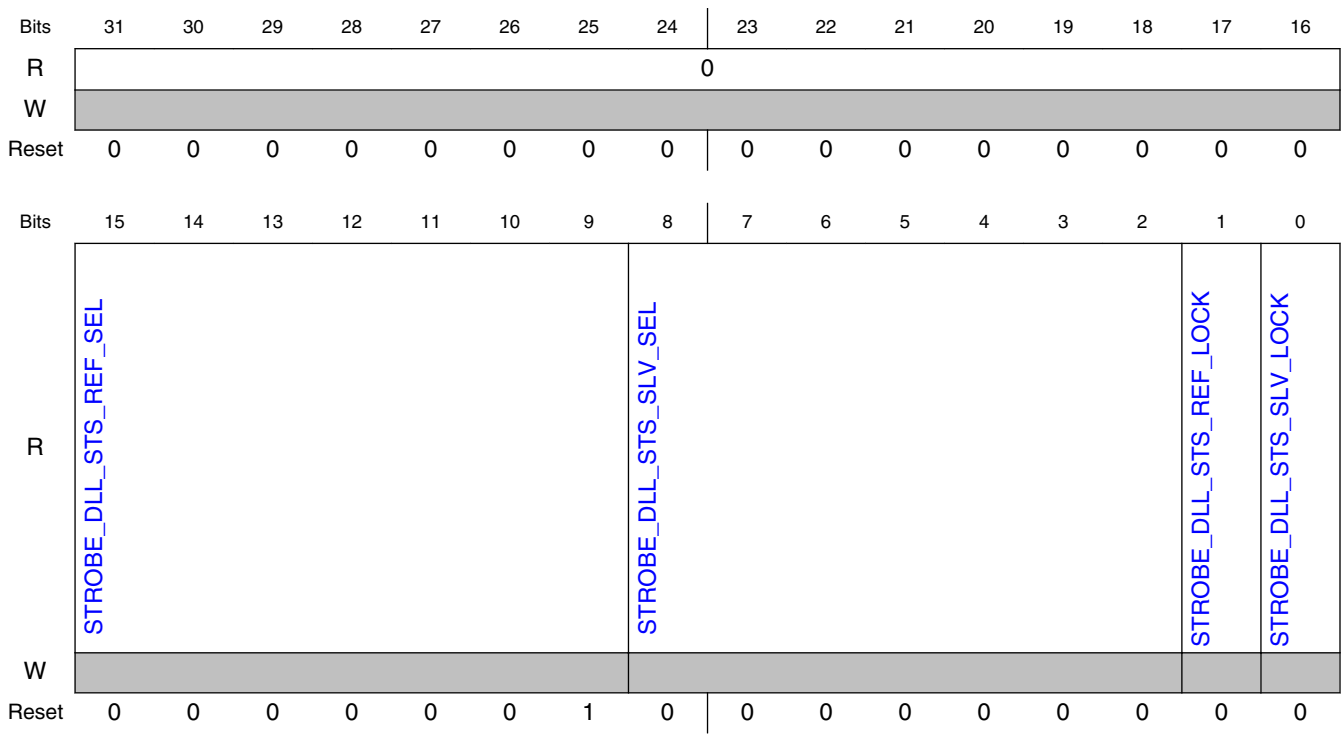
10.3.6.1.28.1 Offset

Register	Offset
STROBE_DLL_STATUS	74h

10.3.6.1.28.2 Function

This register contains the strobe DLL status information. All fields are read only and read the same as the power-reset value.

10.3.6.1.28.3 Diagram



10.3.6.1.28.4 Fields

Field	Function
31-16	This field is reserved.
—	This read-only field is reserved and always has the value 0.
15-9	Strobe DLL status reference select

Table continues on the next page...

Field	Function
STROBE_DLL_STS_REF_SEL	Reference delay line select taps. This is encoded by 7 fields for 127 taps.
8-2 STROBE_DLL_STS_SLV_SEL	Strobe DLL status slave select Slave delay line select status. This is the instant value generated from reference chain. Because the reference chain can only be updated when STROBE_REF_CLOCK is detected, this value can be updated with the right value when the reference is locked.
1 STROBE_DLL_STS_REF_LOCK	Strobe DLL status reference lock This signifies that the DLL has detected and locked to a half-phase REF_CLOCK shift, it allows the slave delay-line to perform programmed clock delays.
0 STROBE_DLL_STS_SLV_LOCK	Strobe DLL status slave lock Slave delay-line lock status. This signifies that a valid calibration has been set to the slave-delay line, and the slave-delay line is implementing the programmed delay value.

10.3.6.1.29 Vendor Specific Register (VEND_SPEC)

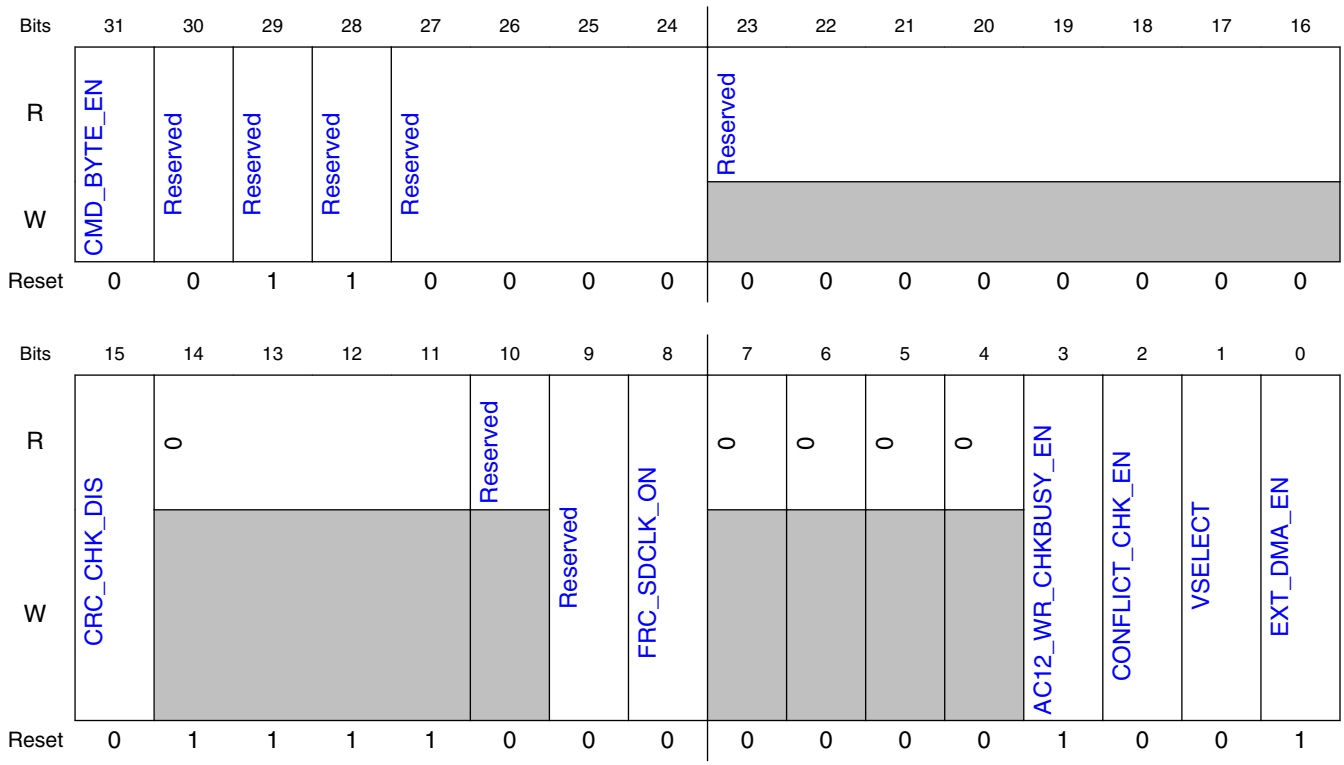
10.3.6.1.29.1 Offset

Register	Offset
VEND_SPEC	C0h

10.3.6.1.29.2 Function

This register contains the vendor specific control/status register.

10.3.6.1.29.3 Diagram



10.3.6.1.29.4 Fields

Field	Function
31 CMD_BYTE_EN	Byte access This field controls the byte access. 0b - Disable 1b - Enable
30 —	Reserved Always write as 0.
29 —	Reserved Always write as 1
28 —	Reserved Always write as 0.
27-24 —	Reserved Always write as 4'b0000.
23-16 —	Reserved Always write as 8'h00.
15 CRC_CHK_DIS	CRC Check Disable 0b - Check CRC16 for every read data packet and check CRC fields for every write data packet

Table continues on the next page...

Field	Function
	1b - Ignore CRC16 check for every read data packet and ignore CRC fields check for every write data packet
14-11 —	Reserved Reserved
10 —	Reserved Always write as 0.
9 —	Reserved Always write as 0.
8 FRC_SDCLK_ON	Force CLK Force CLK output active Do not set this bit to 1 unless it is necessary. Also, make sure that this bit is cleared when uSDHC's clock is about to be changed (frequency change, clock source change, or delay chain tuning). 0b - CLK active or inactive is fully controlled by the hardware. 1b - Force CLK active
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 AC12_WR_CHKBUSY_EN	Check busy enable Check busy enable after auto CMD12 for write data packet 0b - Do not check busy after auto CMD12 for write data packet 1b - Check busy after auto CMD12 for write data packet
2 CONFLICT_CHK_EN	Conflict check enable It is not implemented in uSDHC IP. 0b - Conflict check disable 1b - Conflict check enable
1 VSELECT	Voltage selection Change the value of output signal VSELECT to control the voltage on pads for external card. There must be a control circuit out of uSDHC to change the voltage on pads. 0b - Change the voltage to high voltage range, around 3.0 V 1b - Change the voltage to low voltage range, around 1.8 V
0 EXT_DMA_EN	External DMA request enable Enable the request to external DMA. When the internal DMA (either Simple DMA or Advanced DMA) is not in use, and this field is set, uSDHC sends out DMA request when the internal buffer is ready. This field is particularly useful when transferring data by Arm platform polling mode, and it is not allowed to send out the external DMA request. By default, this field is set. 0b - In any scenario, uSDHC does not send out external DMA request. 1b - When internal DMA is not active, the external DMA request is sent out.

10.3.6.1.30 MMC Boot (MMC_BOOT)

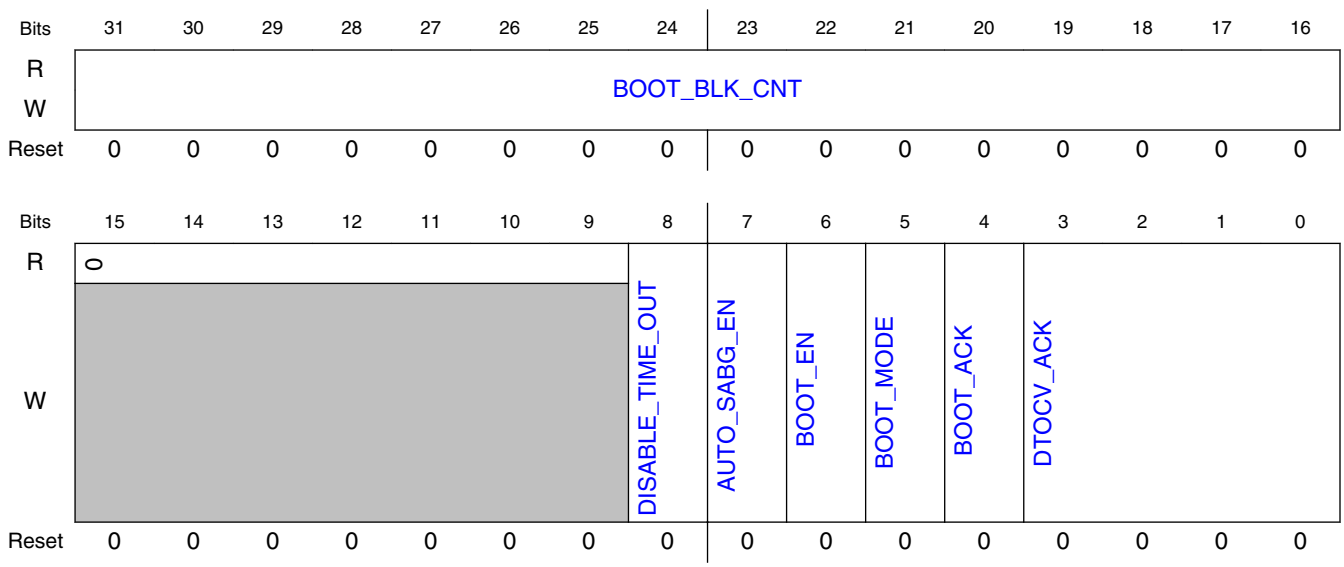
10.3.6.1.30.1 Offset

Register	Offset
MMC_BOOT	C4h

10.3.6.1.30.2 Function

This register contains the MMC Fast Boot control register.

10.3.6.1.30.3 Diagram



10.3.6.1.30.4 Fields

Field	Function
31-16 BOOT_BLK_CNT	Stop At Block Gap value of automatic mode The value defines the Stop At Block Gap value of automatic mode. When received, card block cnt is equal to (BLK_CNT - BOOT_BLK_CNT) and AUTO_SABG_EN is 1, then Stop At Block Gap. Here, BLK_CNT is defined in the Block Attributes Register, field 31 - 16 of 0x04.
15-9 —	Reserved
8	Time out

Table continues on the next page...

Field	Function
DISABLE_TIME_OUT	**NOTE:** When this field is set, there is no timeout check no matter whether BOOT_EN is set or not. 0b - Enable time out 1b - Disable time out
7 AUTO_SABG_EN	Auto stop at block gap During boot, enable auto stop at block gap function. This function is triggered, and host stops at block gap when received card block cnt is equal to (BLK_CNT - BOOT_BLK_CNT).
6 BOOT_EN	Boot enable Boot mode enable 0b - Fast boot disable 1b - Fast boot enable
5 BOOT_MODE	Boot mode Boot mode select 0b - Normal boot 1b - Alternative boot
4 BOOT_ACK	BOOT ACK Boot ACK mode select 0b - No ack 1b - Ack
3-0 DTOCV_ACK	DTOCV_ACK Boot ACK time out counter value 0000b - SDCLK x 2^32 0001b - SDCLK x 2^33 0010b - SDCLK x 2^18 0011b - SDCLK x 2^19 0100b - SDCLK x 2^20 0101b - SDCLK x 2^21 0110b - SDCLK x 2^22 0111b - SDCLK x 2^23 1110b - SDCLK x 2^30 1111b - SDCLK x 2^31

10.3.6.1.31 Vendor Specific 2 Register (VEND_SPEC2)

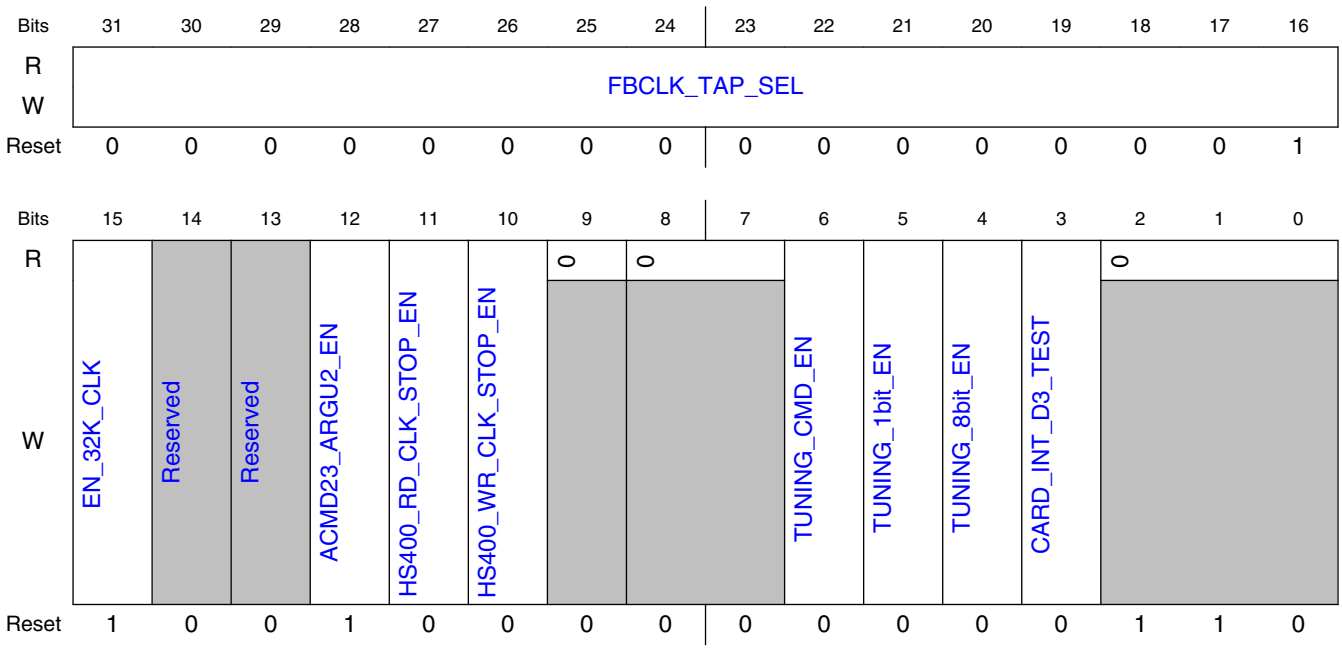
10.3.6.1.31.1 Offset

Register	Offset
VEND_SPEC2	C8h

10.3.6.1.31.2 Function

This register contains the vendor specific control 2 register.

10.3.6.1.31.3 Diagram



10.3.6.1.31.4 Fields

Field	Function
31-16 FBCLK_TAP_SEL	Enable extra delay on internal feedback clock
15 EN_32K_CLK	Enable 32khz clock for card detection Use 32khz clock for card detection
14 —	Reserved
13 —	Reserved
12 ACMD23_ARGU2_EN	Argument2 register enable for ACMD23 0b - Disable 1b - Argument2 register enable for ACMD23 sharing with SDMA system address register. Default is enabled.
11 HS400_RD_CLK_STOP_EN	HS400 read clock stop enable Only stop clock at read block gap.
10 HS400_WR_CLK_STOP_EN	HS400 write clock stop enable Only stop clock at write block gap.
9 —	Reserved

Table continues on the next page...

Field	Function
8-7 —	Reserved
6 TUNING_CMD_EN	Tuning command enable Enable the auto tuning circuit to check the CMD line. 0b - Auto tuning circuit does not check the CMD line. 1b - Auto tuning circuit checks the CMD line.
5 TUNING_1bit_EN	Tuning 1bit enable Enable the auto tuning circuit to check the DATA0 only. It is used with the TUNING_8bit_EN together.
4 TUNING_8bit_EN	Tuning 8bit enable Enable the auto tuning circuit to check the DATA[7:0]. It is used with the TUNING_1bit_EN together. 0b00 - Tuning circuit only checks the DATA[3:0] 0b01 - Tuning circuit only checks the DATA0 0b10 - Tuning circuit checks the whole DATA[7:0] 0b11 - Invalid **NOTE:** The format of these two fields are [TUNNING_8bit_EN:TUNNING_1bit_EN].
3 CARD_INT_D3_TEST	Card interrupt detection test This field is used only for debugging. 0b - Check the card interrupt only when DATA3 is high. 1b - Check the card interrupt by ignoring the status of DATA3.
2-0 —	Reserved

10.3.6.1.32 Tuning Control (TUNING_CTRL)

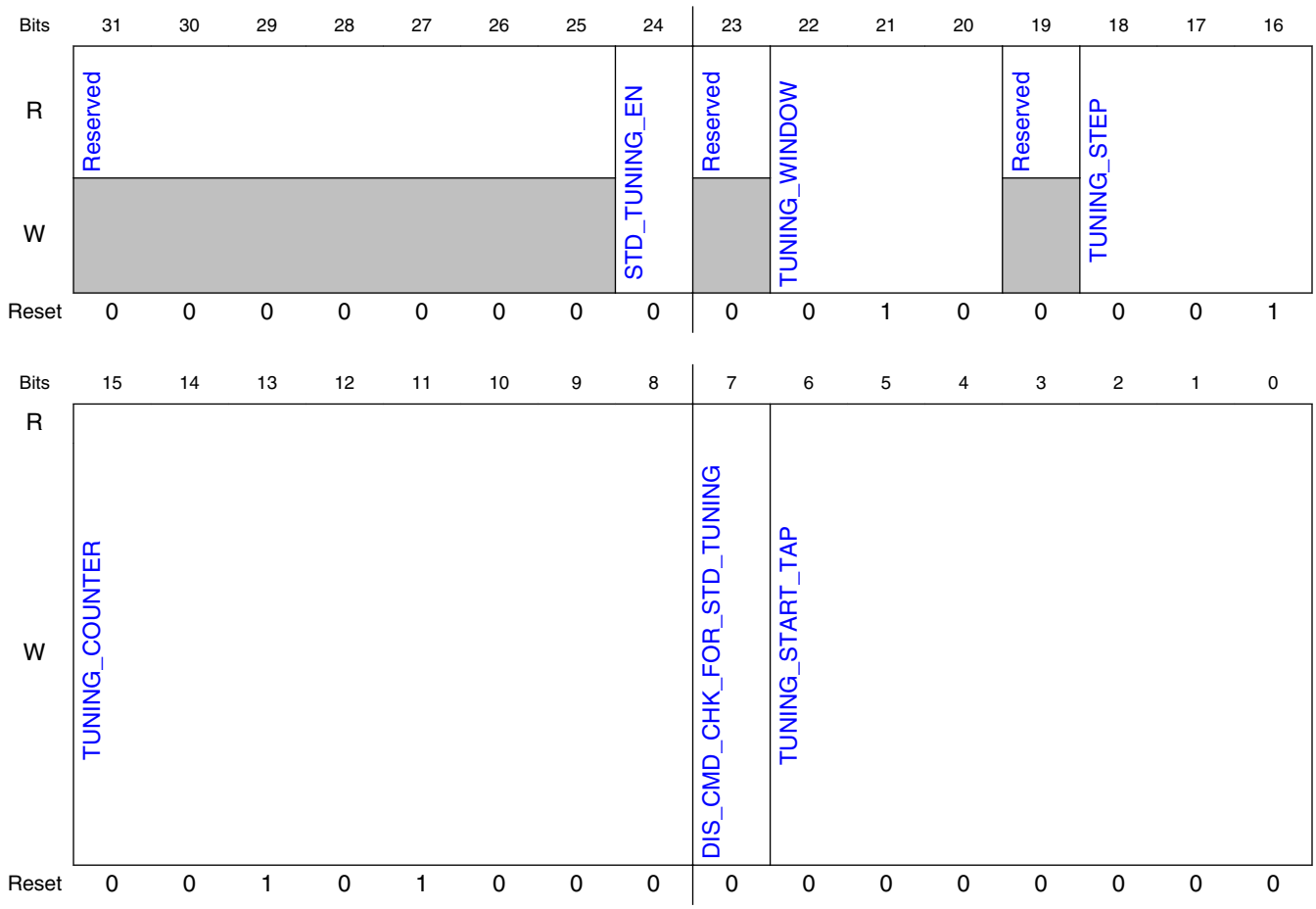
10.3.6.1.32.1 Offset

Register	Offset
TUNING_CTRL	CCh

10.3.6.1.32.2 Function

The register contains configuration of tuning circuit.

10.3.6.1.32.3 Diagram



10.3.6.1.32.4 Fields

Field	Function
31-25 —	Reserved
24 STD_TUNING_EN	Standard tuning circuit and procedure enable This field is used to enable standard tuning circuit and procedure.
23 —	Reserved
22-20 TUNING_WINDOW	Data window Select data window value for auto tuning
19 —	Reserved
18-16	TUNING_STEP

Table continues on the next page...

Field	Function
TUNING_STEP	The increasing delay cell steps in tuning procedure.
15-8 TUNING_COUNTER	Tuning counter The MAX repeat CMD19 times in tuning procedure.
7 DIS_CMD_CHK_FOR_STD_TUNING	Disable command check for standard tuning Writing 1 to this field disables command check (command CRC, CMD end bit error, and CMD index error or CMD timeout error) for standard tuning flow after each tuning command is sent.
6-0 TUNING_START_TAP	Tuning start The start delay cell point when send first CMD19 in tuning procedure.

10.3.6.1.33 Command Queuing Version (CQVER)

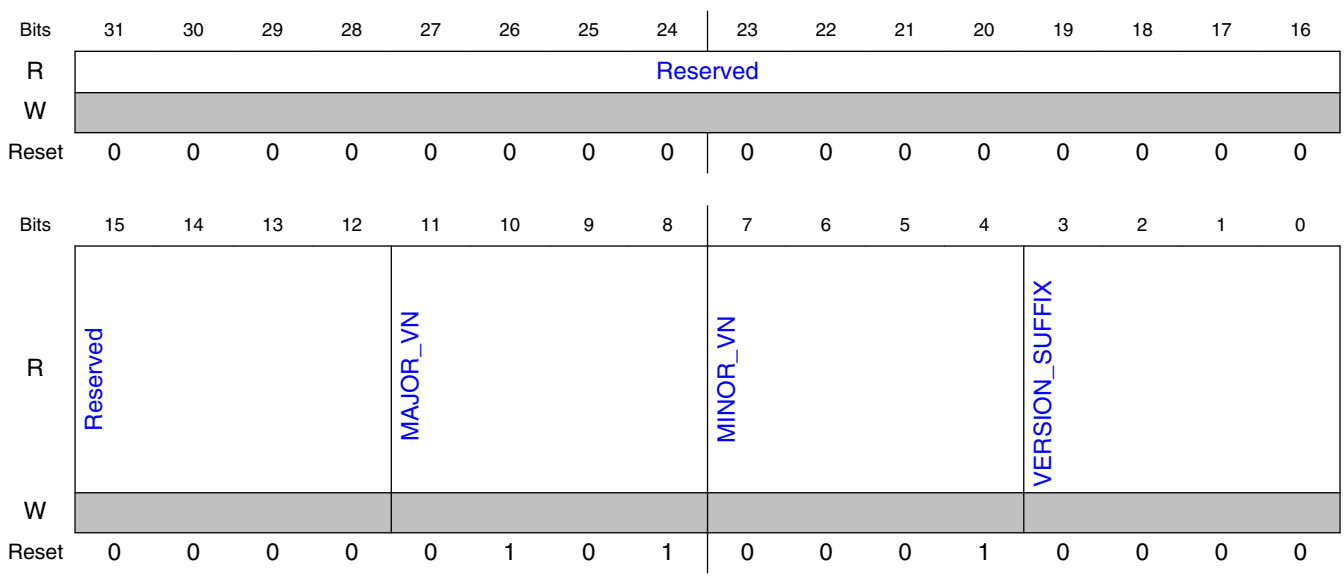
10.3.6.1.33.1 Offset

Register	Offset
CQVER	100h

10.3.6.1.33.2 Function

This register provides information about the version of the e•MMC CQ standard that is implemented by CQE, in BCD format. The current version is 5.1.

10.3.6.1.33.3 Diagram



10.3.6.1.33.4 Fields

Field	Function
31-12 —	Reserved
11-8 MAJOR_VN	e •MMC major version number Digit right of decimal point, in BCD format
7-4 MINOR_VN	e •MMC minor version number Digit right of decimal point, in BCD format
3-0 VERSION_SUFFIX	e •MMC version suffix The second digit right of decimal point, in BCD format

10.3.6.1.34 Command Queuing Capabilities (CQCAP)

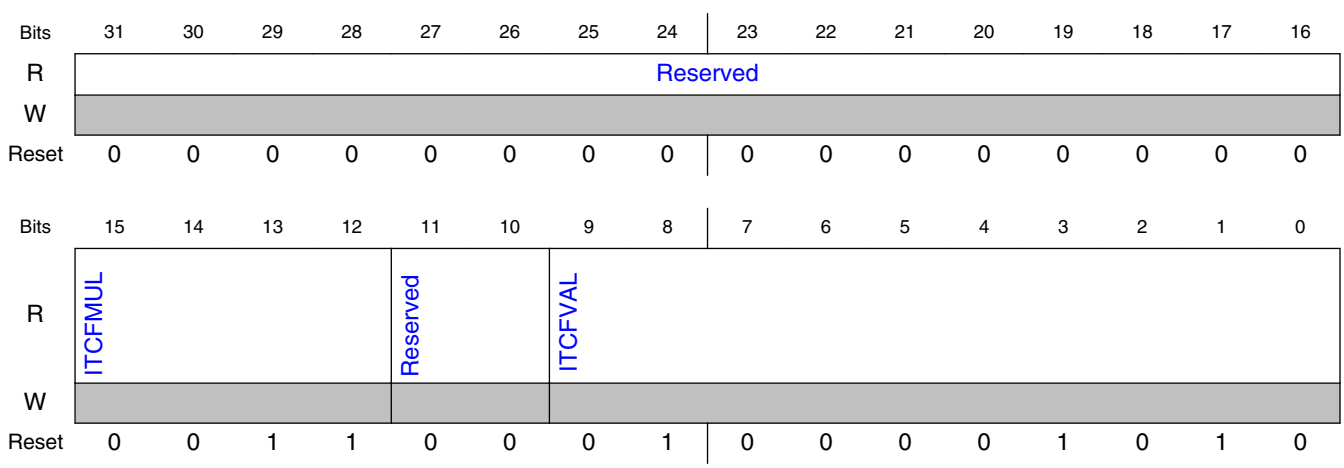
10.3.6.1.34.1 Offset

Register	Offset
CQCAP	104h

10.3.6.1.34.2 Function

This register indicates hardware capabilities.

10.3.6.1.34.3 Diagram



10.3.6.1.34.4 Fields

Field	Function
31-16 —	Reserved
15-12 ITCFMUL	Internal timer clock frequency multiplier ITCFMUL and ITCFVAL indicate the frequency of the clock used for interrupt coalescing timer and for determining the SQS polling period. See ITCFVAL definition for details. 0001b - 0.001 MHz 0010b - 0.01 MHz 0011b - 0.1 MHz 0100b - 1 MHz 0101b - 10 MHz 0110-1001b - Reserved
11-10 —	Reserved
9-0 ITCFVAL	Internal timer clock frequency value ITCFMUL and ITCFVAL indicate the frequency of the clock used for interrupt coalescing timer and for determining the polling period when using periodic SEND_QUEUE_ STATUS (CMD13) polling. The clock frequency is calculated as ITCFVAL* ITCFMUL. For example, to encode 19.2 MHz, ITCFVAL is C0h (= 192 decimal) and ITCFMUL is 2h (0.1 MHz): 192 * 0.1 MHz = 19.2 MHz.

10.3.6.1.35 Command Queuing Configuration (CQCFG)

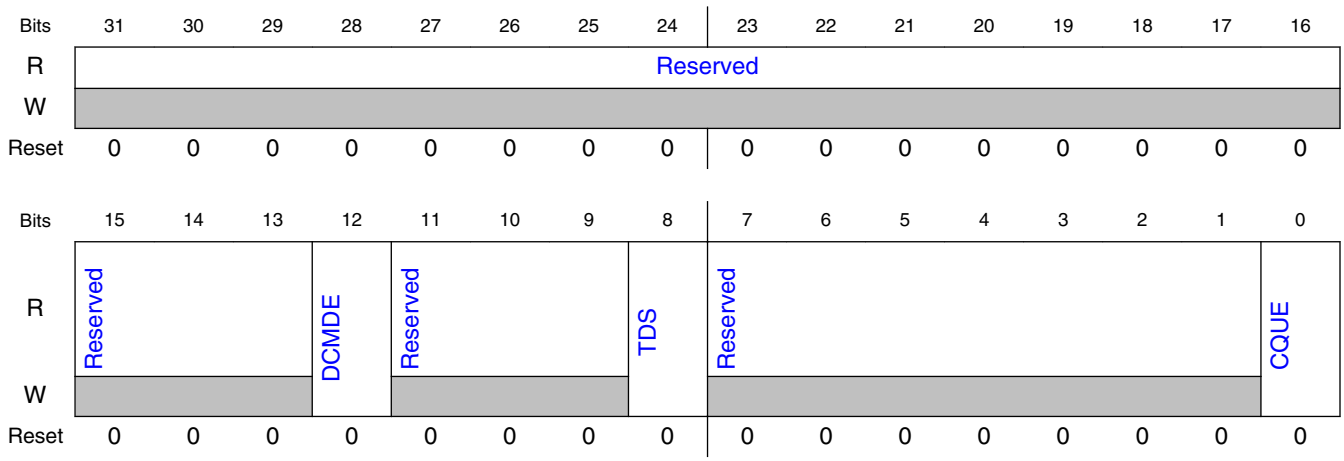
10.3.6.1.35.1 Offset

Register	Offset
CQCFG	108h

10.3.6.1.35.2 Function

This register controls CQE behavior affecting the general operation of command queuing module or operation of multiple tasks in the same time.

10.3.6.1.35.3 Diagram



10.3.6.1.35.4 Fields

Field	Function
31-13 —	Reserved
12 DCMDE	Direct command (DCMD) enable This bit indicates to the hardware whether the Task Descriptor in slot #31 of the TDL is a Data Transfer Task Descriptor, or a Direct Command Task Descriptor. CQE uses this bit when a task is issued in slot #31, to determine how to decode the Task Descriptor. 0b - Task descriptor in slot #31 is a Data Transfer Task Descriptor 1b - Task descriptor in slot #31 is a DCMD Task Descriptor
11-9 —	Reserved
8 TDS	Task descriptor size This bit indicates whether the task descriptor size is 128 bits or 64 bits as detailed in Data Structures section. This bit can only be configured when Command Queuing Enable bit is ‘0’ (command queuing is disabled) 0b - Task descriptor size is 64 bits 1b - Task descriptor size is 128 bits
7-1 —	Reserved
0 CQUE	Command queuing enable Software shall write '1' this bit when in order to enable command queuing mode (for example, enable CQE). When this bit is 0, CQE is disabled and software controls the e•MMC bus using the legacy e•MMC host controller. Before software writes '1' to this bit, software shall verify that the e•MMC host controller is in idle state and there are no commands or data transfers ongoing. When software wants to exit command queuing mode, it shall clear all previous tasks if such exist before setting this bit to 0.

10.3.6.1.36 Command Queuing Control (CQCTL)

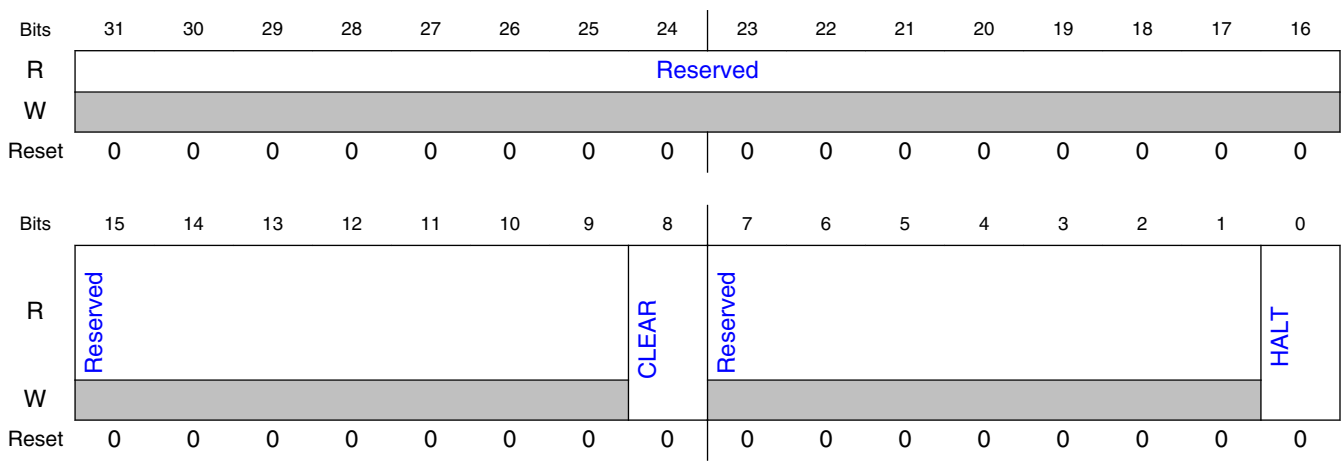
10.3.6.1.36.1 Offset

Register	Offset
CQCTL	10Ch

10.3.6.1.36.2 Function

This register controls CQE behavior affecting the general operation of command queuing module or operation of multiple tasks in the same time.

10.3.6.1.36.3 Diagram



10.3.6.1.36.4 Fields

Field	Function
31-9 —	Reserved
8 CLEAR	Clear all tasks Software shall write '1' this bit when it wants to clear all the tasks sent to the chip. This bit can only be written when CQE is in halt state (for example, Halt bit is 1). When software writes 1, the value of the register is updated to '1', and CQE shall reset CQTDBR register and all other context information for all unfinished tasks. Then CQE will clear this bit.

Table continues on the next page...

Field	Function
	Software should poll on this bit until it is set to back 0 and may then resume normal operation, by clearing the Halt bit. CQE does not communicate to the chip that the tasks were cleared. It is software's responsibility to order the chip to discard the tasks in its queue using CMDQ_TASK_MGMT command. Writing '0' to this register shall have no effect.
7-1 —	Reserved
0 HALT	Halt Host software shall write '1' to the bit when it wants to acquire software control over the e•MMC bus and disable CQE from issuing commands on the bus. For example, issuing a Discard Task command (CMDQ_TASK_MGMT) When software writes '1', CQE shall complete the ongoing task if such a task is in progress. Once the task is completed and CQE is in idle state, CQE shall not issue new commands and shall indicate so to software by setting this bit to 1. Software may poll on this bit until it is set to 1, and may only then send commands on the e•MMC bus. In order to exit halt state (for example, resume CQE activity), software shall clear this bit (write '0'). Writing '0' when the value is already '0' shall have no effect.

10.3.6.1.37 Command Queuing Interrupt Status (CQIS)

10.3.6.1.37.1 Offset

Register	Offset
CQIS	110h

10.3.6.1.37.2 Function

This register indicates pending interrupts that require service. Each bit in this registers is asserted in response a specific event, only if the respective bit is set in CQISTE register.

10.3.6.1.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												TCL	RED	TCC	HAC
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.37.4 Fields

Field	Function
31-4 —	Reserved
3 TCL	Task cleared This status bit is asserted (if CQISTE[TCL_STE] =1) when a task clear operation is completed by CQE. The completed task clear operation is either an individual task clear (CQTCLR) or clearing of all tasks (CQCTL).
2 RED	Response error detected interrupt This status bit is asserted (if CQISTE[RED_STE] =1) when a response is received with an error bit set in the device status field. The contents of the device status field are listed in 6.13. Software uses CQRMEM register to configure which device status bit fields may trigger an interrupt, and which are masked.
1 TCC	Task complete interrupt This status bit is asserted (if CQISTE[TCC_STE] =1) when at least one of the following two conditions are met: • A task is completed and the INT bit is set in its Task Descriptor • Interrupt caused by Interrupt Coalescing logic (see 0)
0 HAC	Halt complete interrupt This status bit is asserted (if CQISTE[HAC_STE] =1) when halt bit in CQCTL register transitions from 0 to 1 indicating that host controller has completed its current ongoing task and has entered halt state.

10.3.6.1.38 Command Queuing Interrupt Status Enable (CQISTE)

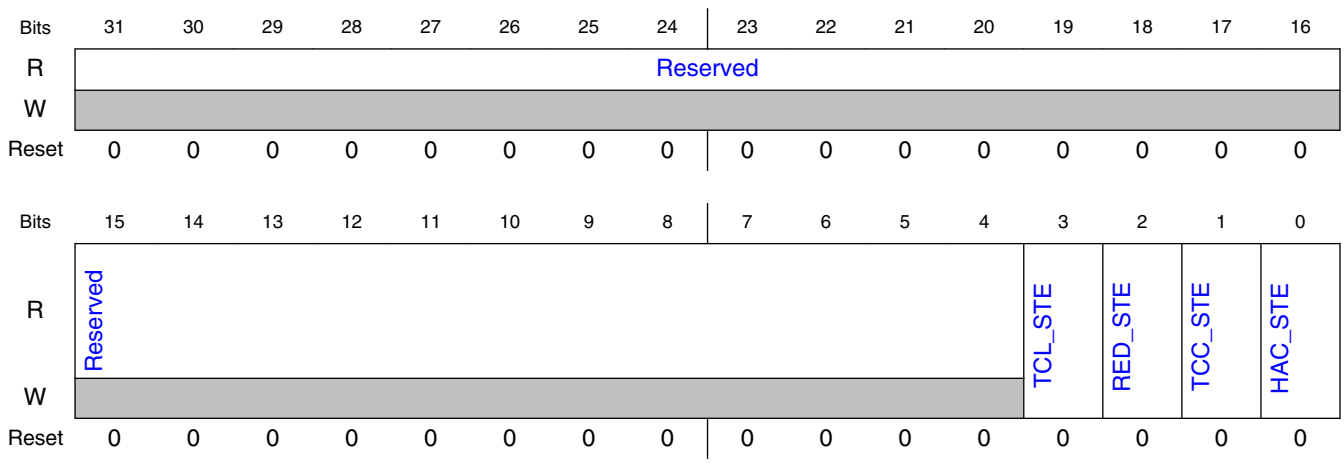
10.3.6.1.38.1 Offset

Register	Offset
CQISTE	114h

10.3.6.1.38.2 Function

This register enables and disables the reporting of the corresponding interrupt to host software in CQIS register. When a bit is set ('1') and the corresponding interrupt condition is active, then the bit is CQIS is asserted. Interrupt sources that are disabled (‘0’) are not indicated in the CQIS register. This register is bit-index matched to CQIS register.

10.3.6.1.38.3 Diagram



10.3.6.1.38.4 Fields

Field	Function
31-4 —	Reserved
3 TCL_STE	Task cleared status enable 0b - CQIS[TCL] is disabled 1b - CQIS[TCL] is set when its interrupt condition is active
2 RED_STE	Response error detected status enable 0b - CQIS[RED] is disabled 1b - CQIS[RED] is set when its interrupt condition is active
1 TCC_STE	Task complete status enable 0b - CQIS[TCC] is disabled 1b - CQIS[TCC] is set when its interrupt condition is active
0 HAC_STE	Halt complete status enable 0b - CQIS[HAC] is disabled 1b - CQIS[HAC] is set when its interrupt condition is active

10.3.6.1.39 Command Queuing Interrupt Signal Enable (CQISGE)

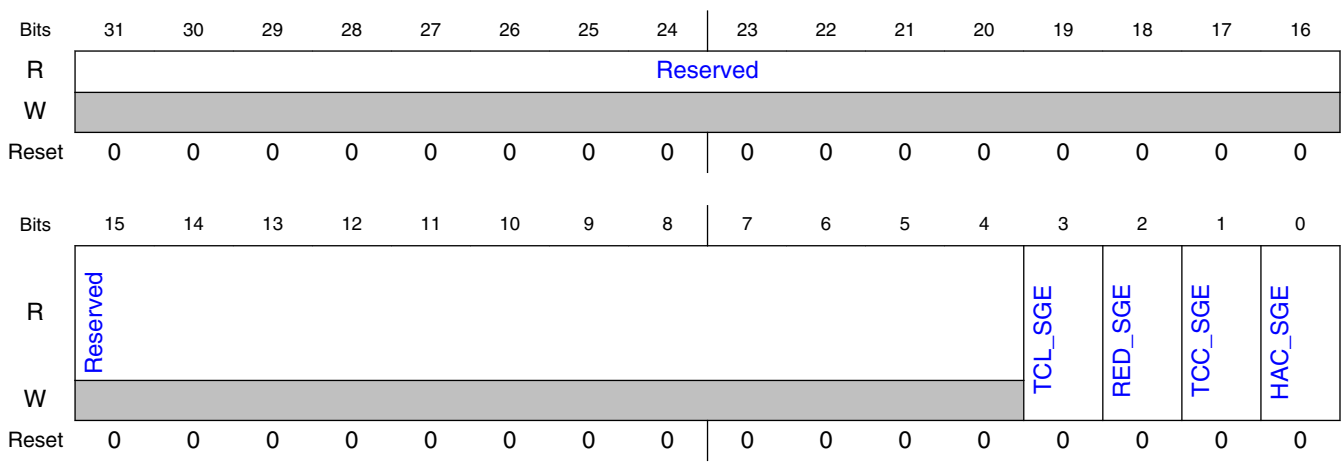
10.3.6.1.39.1 Offset

Register	Offset
CQISGE	118h

10.3.6.1.39.2 Function

This register enables and disables the generation of interrupts to host software. When a bit is set ('1') and the corresponding bit in CQIS is set, then an interrupt is generated. Interrupt sources that are disabled (‘0’) are still indicated in the CQIS register. This register is bit-index matched to CQIS register.

10.3.6.1.39.3 Diagram



10.3.6.1.39.4 Fields

Field	Function
31-4 —	Reserved
3 TCL_SGE	Task cleared signal enable When set and CQIS[TCL] is asserted, the CQE generates an interrupt.
2 RED_SGE	Response error detected signal enable When set and CQIS[RED] is asserted, the CQE generates an interrupt.
1	Task complete signal enable

Table continues on the next page...

Field	Function
TCC_SGE	When set and CQIS[TCC] is asserted, the CQE generates an interrupt.
0	Halt complete signal enable
HAC_SGE	When set and CQIS[HAC] is asserted, the CQE generates an interrupt.

10.3.6.1.40 Command Queuing Interrupt Coalescing (CQIC)

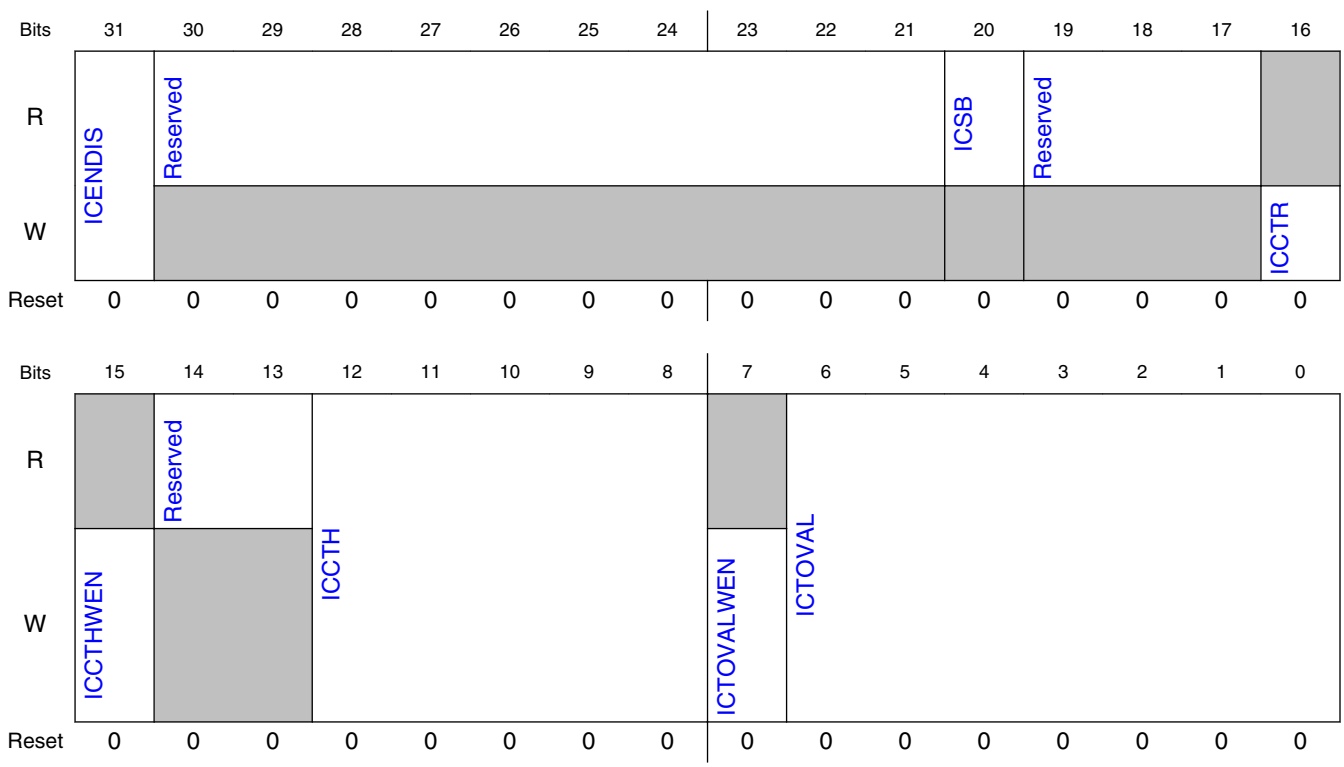
10.3.6.1.40.1 Offset

Register	Offset
CQIC	11Ch

10.3.6.1.40.2 Function

This register controls the interrupt coalescing feature.

10.3.6.1.40.3 Diagram



10.3.6.1.40.4 Fields

Field	Function
31 ICENDIS	Interrupt coalescing enable/disable When host driver writes ‘1’, the interrupt coalescing timer and counter are reset.
30-21 —	Reserved
20 ICSB	Interrupt coalescing status This bit indicates to software whether any tasks (with INT=0) have completed and counted towards interrupt coalescing (for example, ICSB is set if and only if IC counter > 0). 0b - No task completions have occurred since last counter reset (IC counter =0) 1b - At least one task completion has been counted (IC counter >0)
19-17 —	Reserved
16 ICCTR	Counter and timer reset When host driver writes ‘1’, the interrupt coalescing timer and counter are reset.
15 ICCTHWEN	Interrupt coalescing counter threshold write enable When software writes ‘1’, the value ICCTH is updated with the contents written at the same cycle. When software writes ‘0’, the value in ICCTH is not updated. **NOTE:** Write operations to ICCTH are only allowed when the task queue is empty.
14-13 —	Reserved
12-8 ICCTH	Interrupt coalescing counter threshold Software uses this field to configure the number of task completions (only tasks with INT=0 in the Task Descriptor) which are required in order to generate an interrupt. Counter Operation: As data transfer tasks with INT=0 complete, they are counted by CQE. The counter is reset by software during the interrupt service routine. The counter stops counting when it reaches the value configured in ICCTH. The maximum allowed value is 31. **NOTE:** When ICCTH is 0, task completions are not counted, and counting-based interrupts are not generated. In order to write to this field, the ICCTHWEN field must be set at the same write operation.
7 ICTOVALWEN	Interrupt coalescing timeout value write enable When software writes ‘1’, the value ICTOVAL is updated with the contents written at the same cycle. When software writes ‘0’, the value in ICTOVAL is not updated. **NOTE:** Write operations to ICTOVAL are only allowed when the task queue is empty.
6-0 ICTOVAL	Interrupt coalescing timeout value Software uses this field to configure the maximum time allowed between the completion of a task on the bus and the generation of an interrupt. Timer Operation: The timer is reset by software during the interrupt service routine. It starts running when a data transfer task with INT=0 is completed, after the timer was reset. When the timer reaches the value configured in ICTOVAL field it generates an interrupt and stops.

Field	Function
	The timer's unit is equal to 1024 clock periods of the clock whose frequency is specified in the Internal Timer Clock Frequency field CQCAP register. The minimum value is 01h (1024 clock periods) and the maximum value is 7Fh (127*1024 clock periods). For example, a CQCAP field value of 0 indicates a 19.2 MHz clock frequency (period = 52.08 ns). If the setting in ICTOVAL is 10h, the calculated polling period is 16*1024*52.08 ns= 853.33 us **NOTE:** When ICTOVAL is 0, the timer is not running, and timer-based interrupts are not generated. In order to write to this field, the ICTOVALWEN bit must be set at the same write operation.

10.3.6.1.41 Command Queuing Task Descriptor List Base Address (CQTDLBA)

10.3.6.1.41.1 Offset

Register	Offset
CQTDLBA	120h

10.3.6.1.41.2 Function

This register is used for configuring the lower 32 bits of the byte address of the head of the Task Descriptor List in the host memory.

10.3.6.1.41.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDLBA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDLBA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.41.4 Fields

Field	Function
31-0	Task descriptor list base address

Field	Function
TDLBA	This register stores the LSB bits (bits 31:0) of the byte address of the head of the Task Descriptor List in system memory. The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by Host driver. This address shall be set on 1 KByte boundary: The lower 10 bits of this register shall be set to 0 by software and shall be ignored by CQE.

10.3.6.1.42 Command Queuing Task Descriptor List Base Address Upper 32 Bits (CQTDLBAU)

10.3.6.1.42.1 Offset

Register	Offset
CQTDLBAU	124h

10.3.6.1.42.2 Function

This register is used for configuring the upper 32 bits of the byte address of the head of the Task Descriptor List in the host memory.

10.3.6.1.42.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDLBAU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDLBAU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.42.4 Fields

Field	Function
31-0 TDLBAU	Task descriptor list base address This register stores the MSB bits (bits 63:32) of the byte address of the head of the Task Descriptor List in system memory.

Field	Function
	The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by Host driver. This register is reserved when using 32-bit addressing mode.

10.3.6.1.43 Command Queuing Task Doorbell (CQTDBR)

10.3.6.1.43.1 Offset

Register	Offset
CQTDBR	128h

10.3.6.1.43.2 Function

Using this register, software triggers CQE to process a new task.

10.3.6.1.43.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.43.4 Fields

Field	Function
31-0 TDBR	Task doorbell Software configures TDLBA and TDLBAU, and enable CQE in CQCFG before using this register. Writing 1 to bit n of this register triggers CQE to start processing the task encoded in slot n of the TDL. CQE always processes tasks in-order according to the order submitted to the list by CQTDBR write transactions. CQE processes Data Transfer tasks by reading the Task Descriptor and sending QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) commands to the device.

Field	Function
	CQE processes DCMD tasks (in slot #31, when enabled) by reading the Task Descriptor, and generating the command encoded by its index and argument. The corresponding bit is cleared to '0' by CQE in one of the following events: 1. When a task execution is completed (with success or error) 2. The task is cleared using CQTCLR register 3. All tasks are cleared using CQCTL register 4. CQE is disabled using CQCFG register Software may initiate multiple tasks at the same time (batch submission) by writing 1 to multiple bits of this register in the same transaction. In the case of batch submission: CQE shall process the tasks in order of the task index, starting with the lowest index. If one or more tasks in the batch are marked with QBR, the ordering of execution will be based on said processing order. Writing 0 by software shall have no impact on the hardware, and will not change the value of the register bit.

10.3.6.1.44 Command Queuing Task Completion Notification (CQTCN)

10.3.6.1.44.1 Offset

Register	Offset
CQTCN	12Ch

10.3.6.1.44.2 Function

This register is used by CQE to notify software about completed tasks.

10.3.6.1.44.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TCN															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TCN															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.44.4 Fields

Field	Function
31-0 TCN	Task complete notification CQE shall set bit n of this register (at the same time it clears bit n of CQTDBR) when a task execution is completed (with success or error). When receiving interrupt for task completion, software may read this register to know which tasks have finished. After reading this register, software may clear the relevant bit fields by writing 1 to the corresponding bits.

10.3.6.1.45 Command Queuing Device Queue Status (CQDQS)

10.3.6.1.45.1 Offset

Register	Offset
CQDQS	130h

10.3.6.1.45.2 Function

This register stores the most recent value of the device's queue status.

10.3.6.1.45.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DQS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DQS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.45.4 Fields

Field	Function
31-0 DQS	Device queue status

Field	Function
	Every time the Host controller receives a queue status register (QSR) from the device, it updates this register with the response of status command (For example, the chip's queue status).

10.3.6.1.46 Command Queuing Device Pending Tasks (CQDPT)

10.3.6.1.46.1 Offset

Register	Offset
CQDPT	134h

10.3.6.1.46.2 Function

This register indicates to software which tasks are queued in the device, awaiting execution.

10.3.6.1.46.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DPT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DPT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.46.4 Fields

Field	Function
31-0 DPT	Device pending tasks Bit n of this register is set if and only if QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) were sent for this specific task and if this task hasn't been executed yet. CQE shall set this bit after receiving a successful response for CMD45. CQE shall clear this bit after the task has completed execution.

Field	Function
	Software needs to read this register in the task-discard procedure, when the controller is halted, to determine if the task is queued in the device. If the task is queued, the driver sends a CMDQ_TASK_MGMT (CMD48) to the device ordering it to discard the task. Then software clears the task in the CQE. Only then the software orders CQE to resume its operation using CQCTL register.

10.3.6.1.47 Command Queuing Task Clear (CQTCLR)

10.3.6.1.47.1 Offset

Register	Offset
CQTCLR	138h

10.3.6.1.47.2 Function

This register is used for removing an outstanding task in the CQE. The register should be used only when CQE is in Halt state.

10.3.6.1.47.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TCLR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TCLR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.47.4 Fields

Field	Function
31-0 TCLR	Task clear Writing 1 to bit n of this register orders CQE to clear a task which software has previously issued. This bit can only be written when CQE is in Halt state as indicated in CQCFG register Halt bit. When software writes ‘1’ to a bit in this register, CQE updates the value to ‘1’, and starts clearing the data structures related to the task. CQE clears the bit fields (sets a value of 0) in CQTCLR and in CQTDBR once clear operation is complete.

Field	Function
	Software should poll on the CQTCLR until it is cleared to verify clear operation was complete. Writing to this register only clears the task in the CQE and does not have impact on the device. In order to discard the task in the device, host software shall send CMDQ_TASK _MGMT while CQE is still in Halt state. Host driver is not allowed to use this register to clear multiple tasks at the same time. Clearing multiple tasks can be done using CQCTL register. Writing 0 to a register bit shall have no impact.

10.3.6.1.48 Command Queuing Send Status Configuration 1 (CQSSC1)

10.3.6.1.48.1 Offset

Register	Offset
CQSSC1	140h

10.3.6.1.48.2 Function

The register controls the when SEND_QUEUE_STATUS commands are sent.

10.3.6.1.48.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved												CBC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CIT															
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.48.4 Fields

Field	Function
31-20 —	Reserved
19-16	Send status command block counter

Table continues on the next page...

Field	Function
CBC	This field indicates to CQE when to send SEND_QUEUE_STATUS (CMD13) command to inquire the status of the device's task queue. A value of *n* means CQE sends status command on the CMD line, during the transfer of data block BLOCK_CNT-n, on the data lines, where BLOCK_CNT is the number of blocks in the current transaction. A value of 0 means that SEND_QUEUE_STATUS (CMD13) command is not sent during the transaction. Instead it will be sent only when the data lines are idle. A value of 1 means that STATUS command is to be sent during the last block of the transaction.
15-0 CIT	Send status command idle timer This field indicates to CQE the polling period to use when using periodic SEND_QUEUE_STATUS (CMD13) polling. Periodic polling is used when tasks are pending in the device, but no data transfer is in progress. When a SEND_QUEUE_STATUS response indicating that no task is ready for execution, CQE counts the configured time until it issues the next SEND_QUEUE_STATUS. Timer units are clock periods of the clock whose frequency is specified in the Internal Timer Clock Frequency field CQCAP register. The minimum value is 0001h (1 clock period) and the maximum value is FFFFh (65535 clock periods). Default interval is: 4096 clock periods. For example, a CQCAP field value of 0 indicates a 19.2 MHz clock frequency (period = 52.08 ns). If the setting in CQSSC1.CIT is 1000h, the calculated polling period is 4096*52.08 ns= 213.33 us

10.3.6.1.49 Command Queuing Send Status Configuration 2 (CQSSC2)

10.3.6.1.49.1 Offset

Register	Offset
CQSSC2	144h

10.3.6.1.49.2 Function

This register is used for configuring RCA field in SEND_QUEUE_STATUS command argument.

10.3.6.1.49.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SSC2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.49.4 Fields

Field	Function
31-16 —	Reserved
15-0 SSC2	Send queue status RCA This field provides CQE with the contents of the 16-bit RCA field in SEND_QUEUE_ STATUS (CMD13) command argument. CQE copies this field to bits 31:16 of the argument when transmitting SEND_ QUEUE_STATUS (CMD13) command.

10.3.6.1.50 Command Queuing Command Response for Direct-Command Task (CQCRDCT)

10.3.6.1.50.1 Offset

Register	Offset
CQCRDCT	148h

10.3.6.1.50.2 Function

This register is used for passing the response of a DCMD task to software.

10.3.6.1.50.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CRDCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CRDCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.50.4 Fields

Field	Function
31-0 CRDCT	Direct command last response This register contains the response of the command generated by the last direct-command (DCMD) task which was sent. CQE updates this register when it receives the response for a DCMD task. This register is considered valid only after bit 31 of CQTDBR register is cleared by CQE.

10.3.6.1.51 Command Queuing Response Mode Error Mask (CQRMEM)

10.3.6.1.51.1 Offset

Register	Offset
CQRMEM	150h

10.3.6.1.51.2 Function

This register controls the generation of Response Error Detection (RED) interrupt.

10.3.6.1.51.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RMEM															
Reset	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RMEM															
Reset	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0

10.3.6.1.51.4 Fields

Field	Function
31-0 RMEM	Response mode error mask This bit is used as in interrupt mask on the device status filed which is received in R1/R1b responses. The reset value of this register is set to trigger an interrupt on all “Error” type bits in the device status, as defined in 6.13. **NOTE:** Responses to CMD13 (SQS) encode the QSR, so they are ignored by this logic. 00000000000000000000000000000000b - When a R1/R1b response is received, bit i in the device status is ignored 00000000000000000000000000000001b - When a R1/R1b response is received, with bit i in the device status set, a RED interrupt is generated

10.3.6.1.52 Command Queuing Task Error Information (CQTERRI)

10.3.6.1.52.1 Offset

Register	Offset
CQTERRI	154h

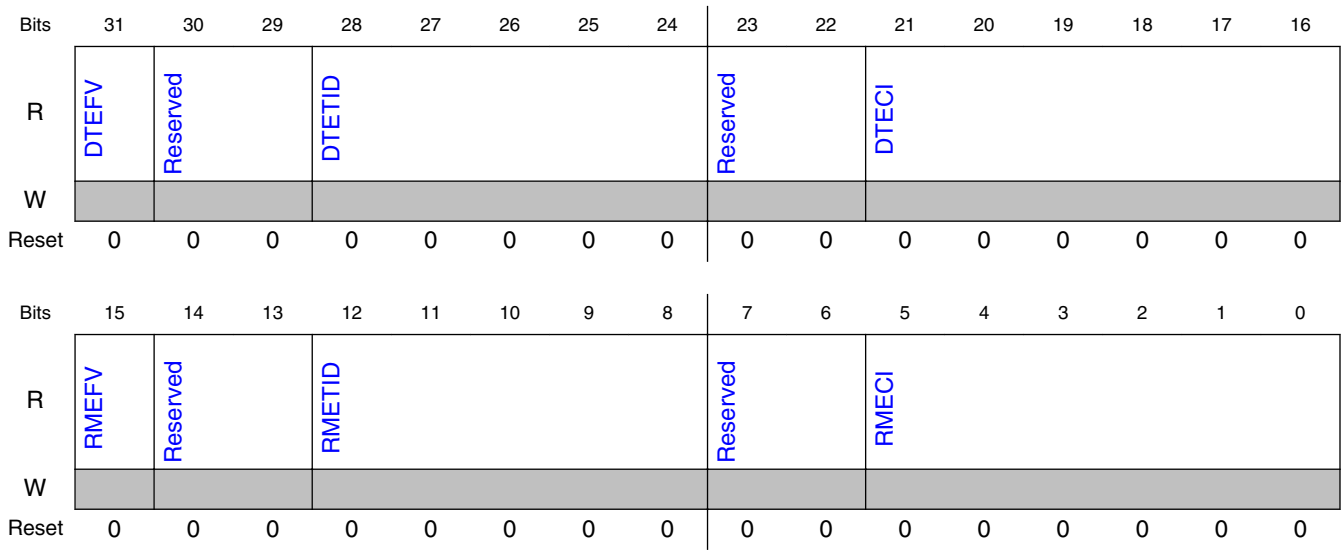
10.3.6.1.52.2 Function

This register is updated by CQE when an error occurs on data or command related to a task activity.

When such error is detected by CQE or indicated by the e•MMC controller CQE stores in CQTERRI the task IDs and the command indices of the commands which were executed on the command line and data lines when the error occurred.

Software is expected to use this information in the error recovery procedure. See B.2.8 for more details.

10.3.6.1.52.3 Diagram



10.3.6.1.52.4 Fields

Field	Function
31 DTEFV	Data transfer error fields valid This bit is updated when an error is detected by CQE, or indicated by e•MMC controller. If a data transfer is in progress when the error is detected/indicated, the bit is set to 1. If a no data transfer is in progress when the error is detected/indicated, the bit is cleared to 0.
30-29 —	Reserved
28-24 DTETID	Data transfer error task ID This field indicates the ID of the task which was executed on the data lines when an error occurred. The field is updated if a data transfer is in progress when an error is detected by CQE, or indicated by e•MMC controller.
23-22 —	Reserved
21-16 DTECI	Data transfer error command index This field indicates the index of the command which was executed on the data lines when an error occurred. The index shall be set to EXECUTE_READ_TASK (CMD46) or EXECUTE_WRITE_TASK (CMD47) according to the data direction. The field is updated if a data transfer is in progress when an error is detected by CQE, or indicated by e•MMC controller.

Table continues on the next page...

Field	Function
15 RMEFV	Response mode error fields valid This bit is updated when an error is detected by CQE, or indicated by e•MMC controller. If a command transaction is in progress when the error is detected/indicated, the bit is set to 1. If a no command transaction is in progress when the error is detected/indicated, the bit is cleared to 0.
14-13 —	Reserved
12-8 RMETID	Response mode error task ID This field indicates the index of the command which was executed on the command line when an error occurred. The field is updated if a command transaction is in progress when an error is detected by CQE, or indicated by e•MMC controller.
7-6 —	Reserved
5-0 RMECI	Response mode error command index This field indicates the index of the command which was executed on the command line when an error occurred. The field is updated if a command transaction is in progress when an error is detected by CQE, or indicated by e•MMC controller.

10.3.6.1.53 Command Queuing Command Response Index (CQCRI)

10.3.6.1.53.1 Offset

Register	Offset
CQCRI	158h

10.3.6.1.53.2 Function

This register stores the index of the last received command response.

10.3.6.1.53.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										LCMDRI					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.53.4 Fields

Field	Function
31-6 —	Reserved
5-0 LCMDRI	Last command response index This field stores the index of the last received command response. CQE shall update the value every time a command response is received..

10.3.6.1.54 Command Queuing Command Response Argument (CQCRA)

10.3.6.1.54.1 Offset

Register	Offset
CQCRA	15Ch

10.3.6.1.54.2 Function

This register stores the argument of the last received command response.

10.3.6.1.54.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LCMDRA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LCMDRA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.6.1.54.4 Fields

Field	Function
31-0 LCMDRA	Last command response argument This field stores the argument of the last received command. CQE shall update the value every time a command response is received.

Chapter 11
Connectivity

11.1 HSIO BLK_CTRL

11.1.1 Overview

The HSIOMIX domain contains control and status registers known as HSIO Block Control (HSIO BLK_CTRL). These registers have miscellaneous top-level controls for peripherals within the HSIOMIX domain. Some of these controls includes clocking, interrupt, and reset sources. Please see the clock control chapter and interrupt chapter for more information regarding the clock input sources and IRQ event mapping to HSIOMIX.

11.1.1.1 Block Diagram

The HSIO BLK_CTRL interface connects to the HSIOMIX domain peripherals. A block diagram illustrating the HSIOMIX domain is shown below.

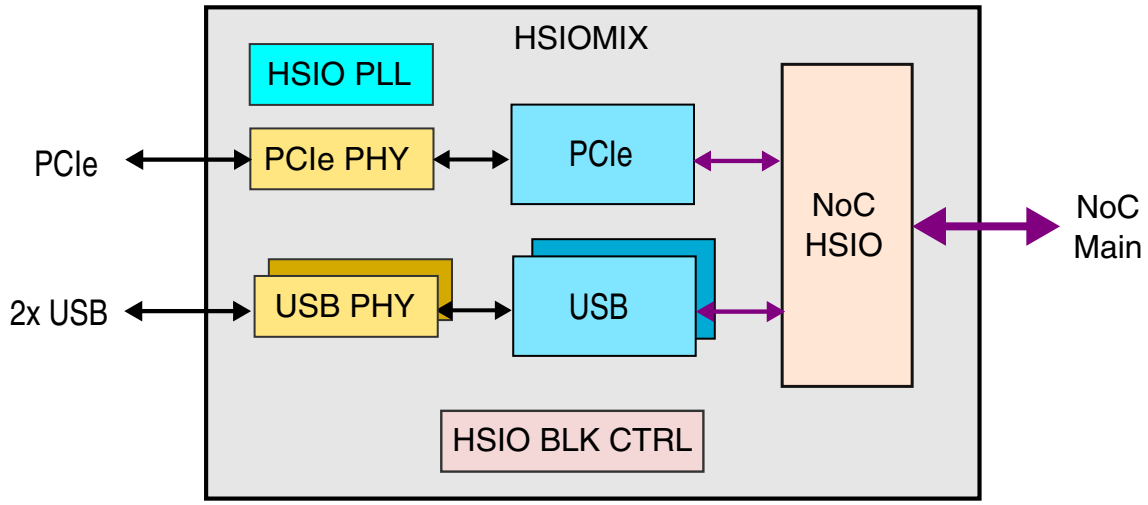


Figure 11-1. HSIOMIX Block Diagram

11.1.2 Clocks and Resets

The HSIO BLK_CTRL contains clock distribution and gating controls, as well as reset handling to several of the HSIOMIX peripherals. Please see related clock and reset controls in the BLK_CTRL register description. For more information about the input clocks and resets to the domain, please see the clock control chapter, system reset chapter, and system-level IRQ assignments.

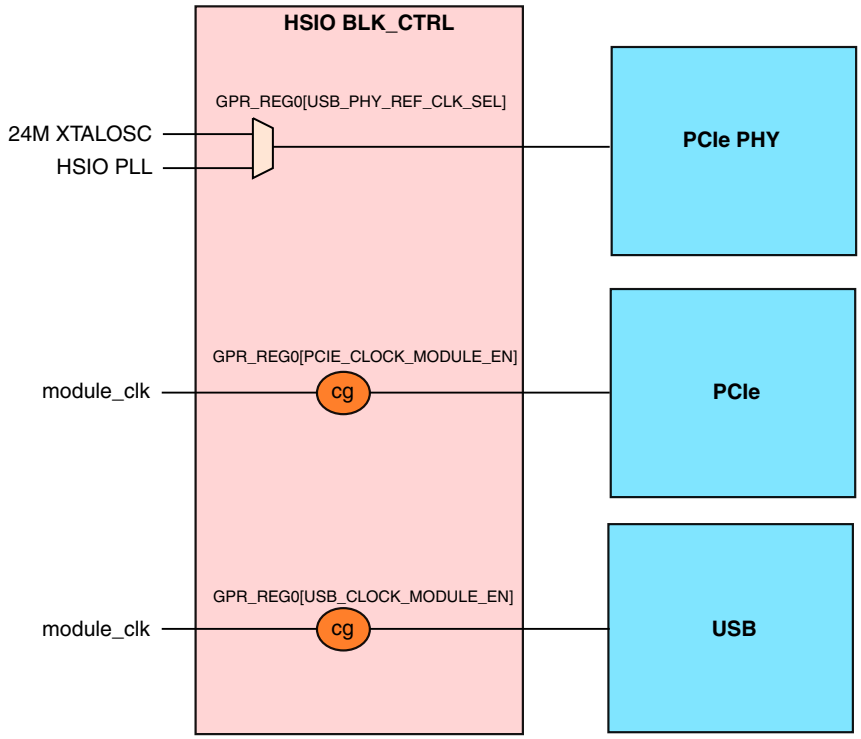


Figure 11-2. HSIO BLK_CTRL Clocks

11.1.3 Memory Map and register definition

This section includes the HSIO BLK_CTRL module memory map and descriptions of all registers.

11.1.3.1 register descriptions

11.1.3.1.1 HSIO_BLK_CTL memory map

HSIO_BLK_CTRL base address: 32F1_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Clock select reset and debug info select (GPR_REG0)	32	RW	0000_0000h
4h	PCIE controller status (GPR_REG1)	32	RO	0000_0000h
8h	PLL configuration 0 (GPR_REG2)	32	RW	1FC8_0000h
Ch	PLL configuration 1 (GPR_REG3)	32	RW	0000_1000h
10h	PCIE PME message and error detect register (GPR_REG4)	32	W1C	0000_0000h

Table continues on the next page...

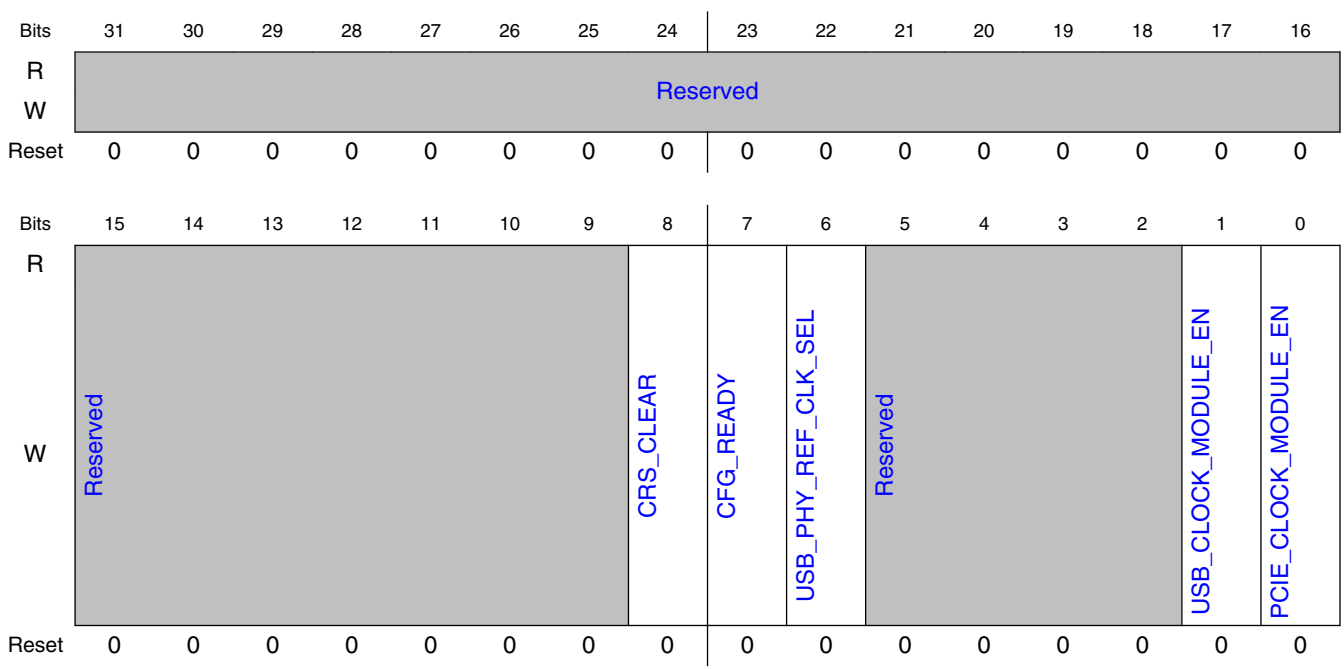
Offset	Register	Width (In bits)	Access	Reset value
14h	PCIE PME message and error detect interrupt enable register (GPR_REG5)	32	RW	0000_0000h
18h	PCIE PME message and error detect interrupt detect disable register (GPR_REG6)	32	RW	0000_0000h
1Ch	USB1 beat limit and enable (GPR_REG7)	32	RW	0000_0000h
20h	USB2 beat limit and enable (GPR_REG8)	32	RW	0000_0000h
24h	PCIE beat limit and enable (GPR_REG9)	32	RW	0000_0000h
100h	Register for USB1 wakeup (USB1_WAKEUP_CTRL)	32	RW	0000_0000h
104h	Status of USB1 wakeup (USB1_WAKEUP_STATUS)	32	RO	0000_1000h
108h	Register for USB2 wakeup (USB2_WAKEUP_CTRL)	32	RW	0000_0000h
10Ch	Status of USB2 wakeup (USB2_WAKEUP_STATUS)	32	RO	0000_1000h

11.1.3.1.2 Clock select reset and debug info select (GPR_REG0)

11.1.3.1.2.1 Offset

Register	Offset
GPR_REG0	0h

11.1.3.1.2.2 Diagram



11.1.3.1.2.3 Fields

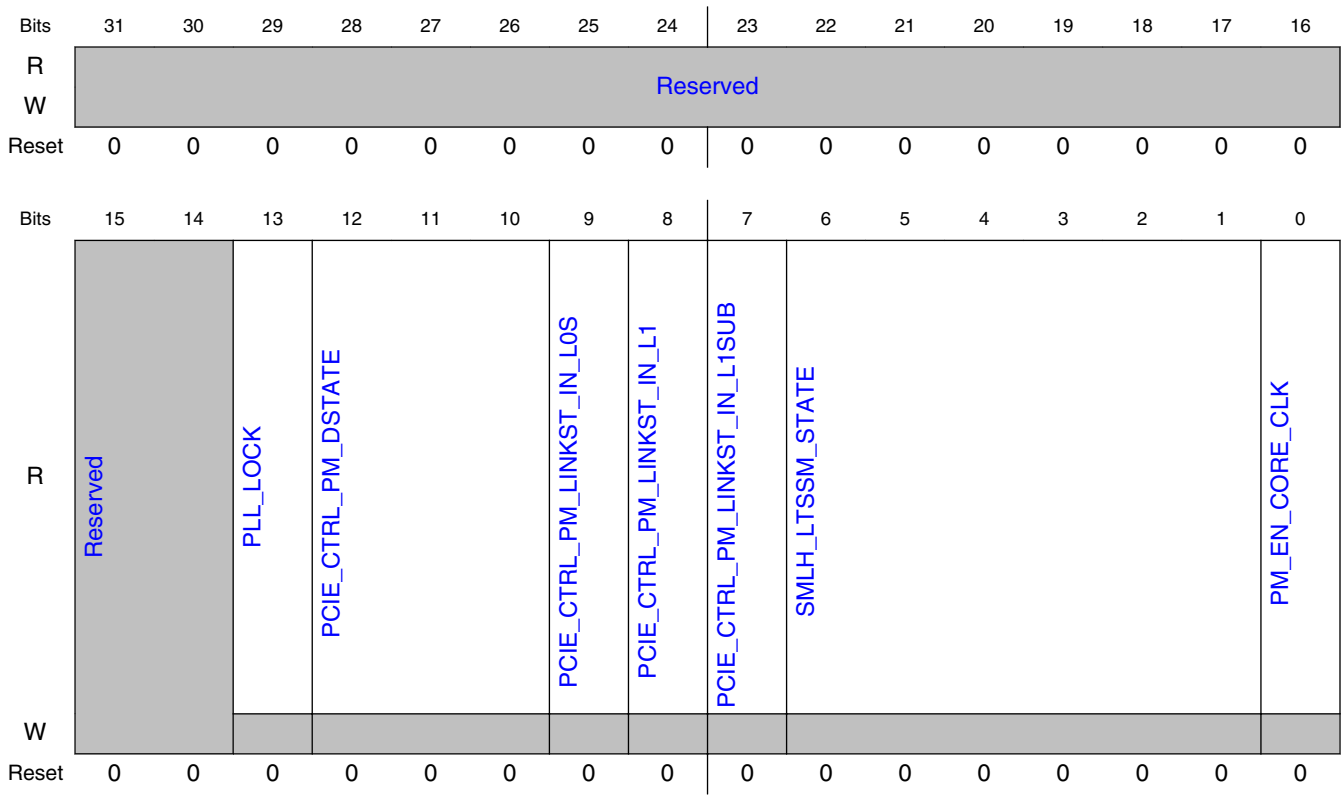
Field	Function
31-9 —	Reserved
8 CRS_CLEAR	Clear CSR interrupt
7 CFG_READY	Configuration ready This bit is cleared by default to enable EP application software configure the PCIe controller. Any config attempts from the Root Complex will be returned with config retry status (CRS). After application Software finishes configuring the PCIe controller it sets the CFG_READY bit to enable accesses from the Root Complex.
6 USB_PHY_REF_CLK_SEL	USB PHY ref clock selection 0b - 24Mhz exteral osc 1b - 100Mhz high performace PLL
5-2 —	Reserved
1 USB_CLOCK_MODULE_EN	USB related clock enable
0 PCIE_CLOCK_MODULE_EN	PCIE related clock enable

11.1.3.1.3 PCIE controller status (GPR_REG1)

11.1.3.1.3.1 Offset

Register	Offset
GPR_REG1	4h

11.1.3.1.3.2 Diagram



11.1.3.1.3.3 Fields

Field	Function
31-14 —	Reserved
13 PLL_LOCK	High performance PLL lock status
12-10 PCIE_CTRL_PM_DSTATE	PCIE ctrl's pm dstate
9 PCIE_CTRL_PM_LINKST_IN_L0S	PCIE ctrl link in l0s state
8 PCIE_CTRL_PM_LINKST_IN_L1	PCIE ctrl link in l1 state
7	PCIE ctrl link in l1sub state

Table continues on the next page...

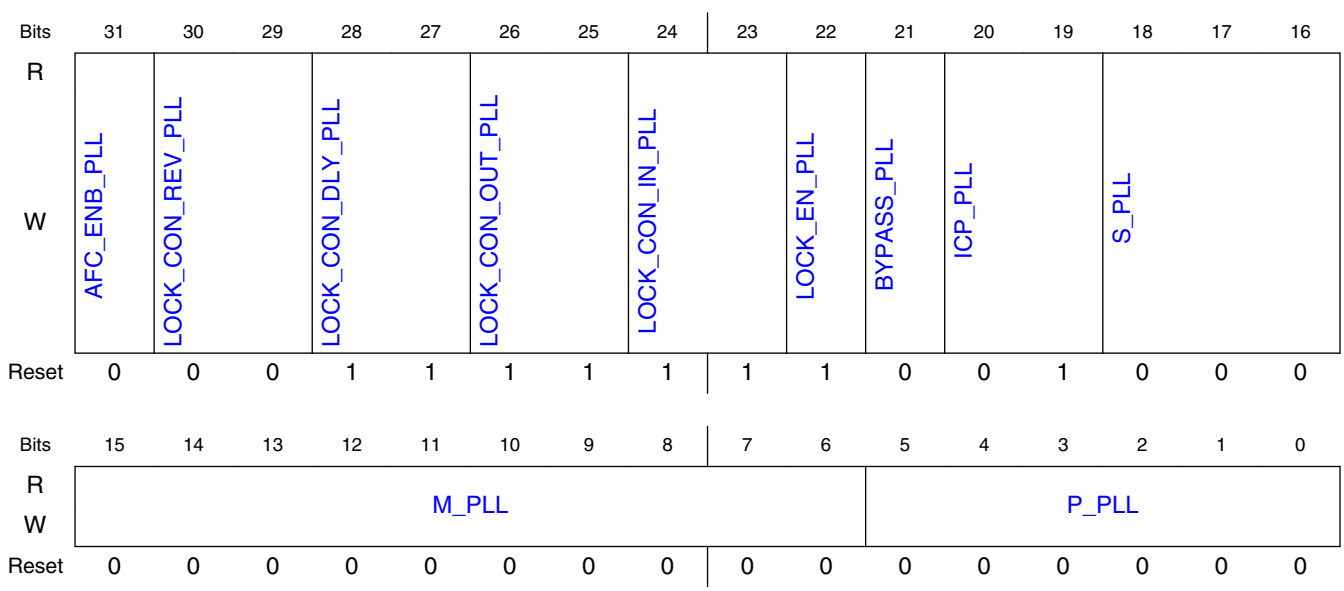
Field	Function
PCIE_CTRL_PM_LINKST_IN_L1SUB	
6-1 SMLH_LTSSM_STATE	PCIE link state
0 PM_EN_CORE_CLK	pm_en_core_clk pin status of pcie ctrl

11.1.3.1.4 PLL configuration 0 (GPR_REG2)

11.1.3.1.4.1 Offset

Register	Offset
GPR_REG2	8h

11.1.3.1.4.2 Diagram



11.1.3.1.4.3 Fields

Field	Function
31	AFC_ENB input of high performance PLL

Table continues on the next page...

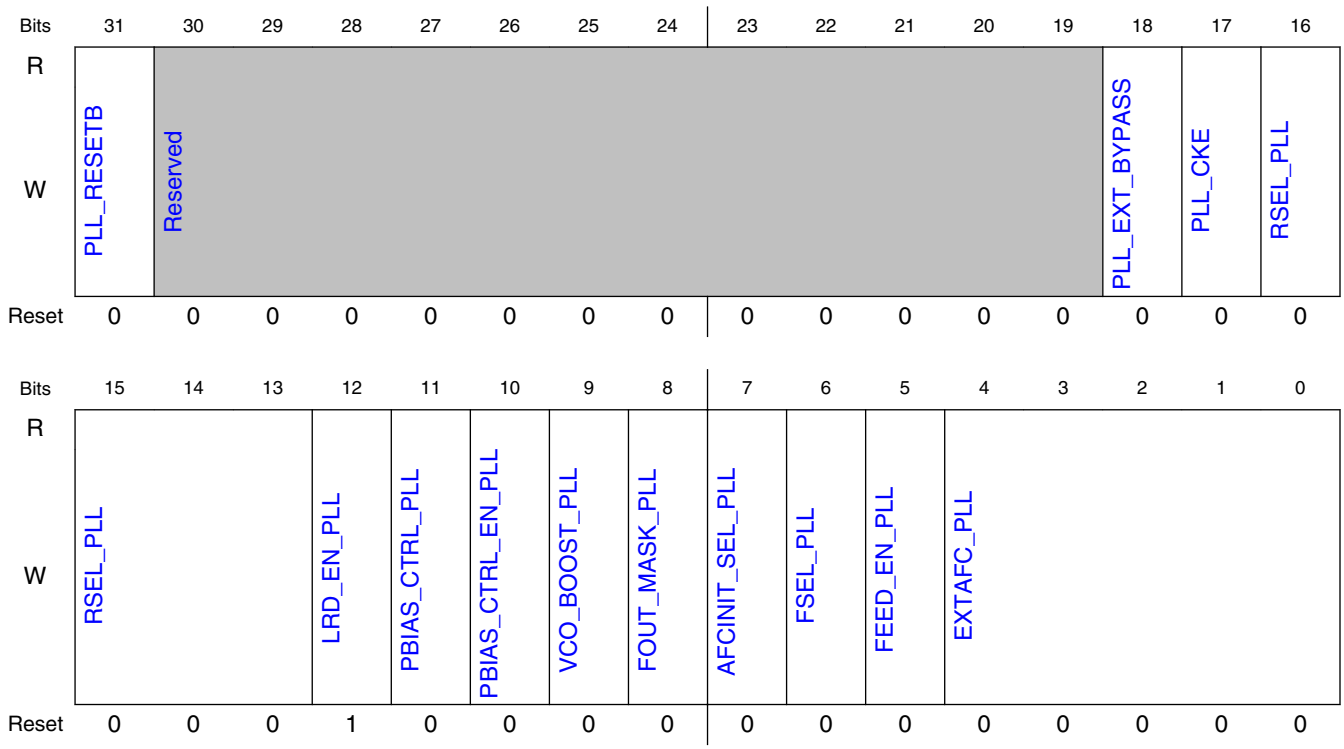
Field	Function
AFC_ENB_PLL	
30-29 LOCK_CON_REV_PLL	Lock con rev pin input of high performance PLL
28-27 LOCK_CON_DLY_PLL	Lock con delay input of high performance PLL
26-25 LOCK_CON_OUT_PLL	Lock con input of high performance PLL
24-23 LOCK_CON_IN_PLL	Lock con in pin input of high performance PLL
22 LOCK_EN_PLL	locken pin input of high performance PLL
21 BYPASS_PLL	Bypass pin input of high performance PLL
20-19 ICP_PLL	ICP pin input of high performance PLL
18-16 S_PLL	S pin input of high performance PLL
15-6 M_PLL	M pin input of high performance PLL
5-0 P_PLL	P pin input of high performance PLL

11.1.3.1.5 PLL configuration 1 (GPR_REG3)

11.1.3.1.5.1 Offset

Register	Offset
GPR_REG3	Ch

11.1.3.1.5.2 Diagram



11.1.3.1.5.3 Fields

Field	Function
31 PLL_RESETB	reset pin input of high performance PLL
30-19 —	Reserved
18 PLL_EXT_BYPASS	PLL ext bypass pin input of high performance PLL
17 PLL_CKE	PLL cke pin input of high performance PLL
16-13 RSEL_PLL	RSEL pin input of high performance PLL
12 LRD_EN_PLL	LRD EN pin input of high performance PLL
11 PBIAS_CTRL_PLL	PBIAS CTRL pin input of high performance PLL
10	PBIAS CTRL EN pin input of high performance PLL

Table continues on the next page...

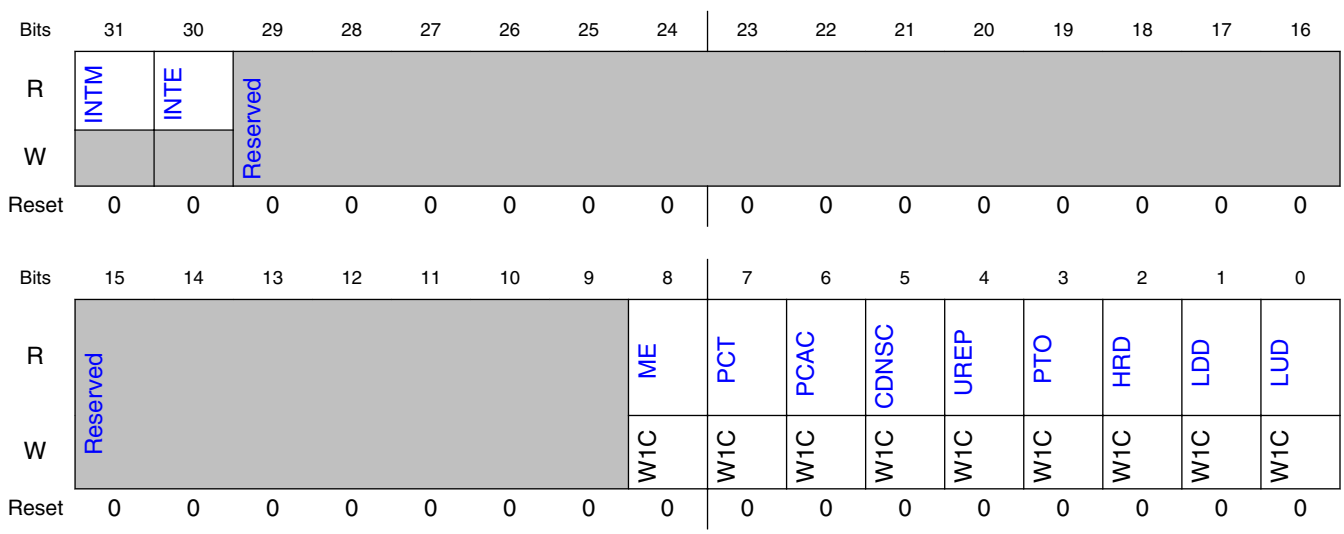
Field	Function
PBIAS_CTRL_EN_PLL	
9 VCO_BOOST_PLL	VCO BOOST pin input of high performance PLL
8 FOUT_MASK_PLL	FOUT MASK pin input of high performance PLL
7 AFCINIT_SEL_PLL	AFCINT SEL input of high performance PLL
6 FSEL_PLL	FSEL pin input of high performance PLL
5 FEED_EN_PLL	Feed en pin input of high performance PLL
4-0 EXTAFC_PLL	Extafc pin input of high performance PLL

11.1.3.1.6 PCIE PME message and error detect register (GPR_REG4)

11.1.3.1.6.1 Offset

Register	Offset
GPR_REG4	10h

11.1.3.1.6.2 Diagram



11.1.3.1.6.3 Fields

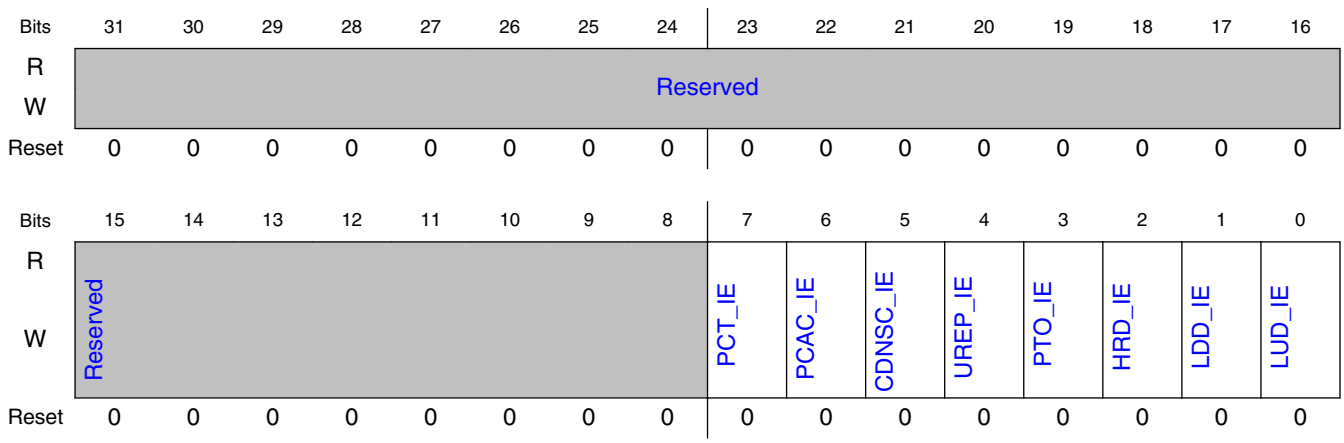
Field	Function
31 INTM	Per PF dependent message interrupt is pending
30 INTE	Per PF dependent error interrupt is pending.
29-9 —	Reserved
8 ME	Indicates Multiple errors of same type. If any of the detectable errors in PEX_ERR_DET register is detected more than one time the ME bit will be set 1'b1: Multiple errors of same type was detected. 1'b0: Multiple errors of same type was not detected
7 PCT	Indicates completion timeout 1'b1: Completion timeout was detected 1'b0: Completion timeout was not detected
6 PCAC	Completer abort was detected. 1'b1: Completer abort was detected 1'b0: Completer abort was not detected
5 CDNSC	Completion with data not succsessful was detected. 1'b1: Completion with data not succsessful was detected 1'b0: Completion with data not succsessful was not detected
4 UREP	Indicates an unsupported request completion was detected 1'b1: Unsupported request in EP mode was detected 1'b0: Unsupported request in EP mode not detected
3 PTO	Indicates that PME turn off was detected 1'b1: PME turn off was detected 1'b0: PME turn off was not detected
2 HRD	Indicates a hot reset was detected 1'b1: hot reset was detected 1'b0: hot reset was not detected
1 LDD	Indicates a link down was detected 1'b1: link down was detected. 1'b0: link down was not detected
0 LUD	Indicates a link up was detected 1'b1: link up was detected. 1'b0: link up was not detected

11.1.3.1.7 PCIE PME message and error detect interrupt enable register (GPR_REG5)

11.1.3.1.7.1 Offset

Register	Offset
GPR_REG5	14h

11.1.3.1.7.2 Diagram



11.1.3.1.7.3 Fields

Field	Function
31-8 —	Reserved
7 PCT_IE	completion timeout interrupt enable 1'b1: Completion timeout interrupt enabled 1'b0: Completion timeout interrupt disabled.
6 PCAC_IE	Completer abort interrupt enable. 1'b1: Completer abort interrupt enabled 1'b0: Completer abort interrupt disabled.
5 CDNSC_IE	Completion with data not succsessful interrupt enable 1'b1: Completion with data not succsessful interrupt enabled. 1'b0: Completion with data not succsessful was not detected.
4 UREP_IE	Unsupported request in EP mode interrupt enable. 1'b1: Unsupported request in EP mode interrupt enabled. 1'b0: Unsupported request in EP mode interrupt disabled

Table continues on the next page...

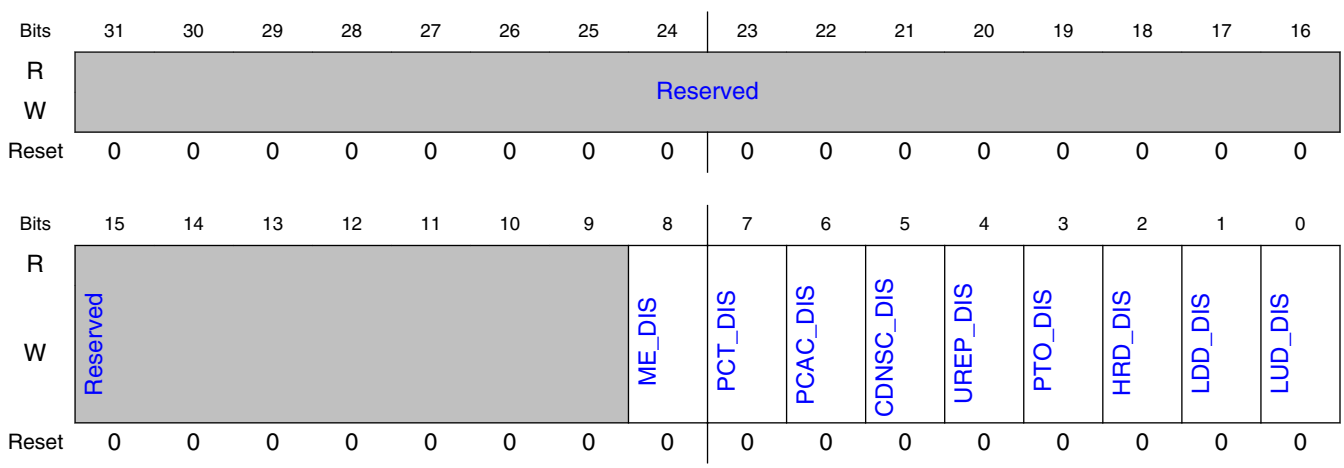
Field	Function
3 PTO_IE	PME turn off detect interrupt enable 1'b1: PME turn off detect interrupt enabled 1'b0: PME turn off detect interrupt disabled.
2 HRD_IE	Hot reset detect interrupt enable. 1'b1: hot reset detect interrupt enabled. 1'b0: hot reset detect interrupt disabled.
1 LDD_IE	Link down detect interrupt enable 1'b1: link down detect interrupt enabled 1'b0: link down detect interrupt disabled
0 LUD_IE	Link up detect interrupt enable 1'b1: link up detect interrupt enabled. 1'b0: link up detect interrupt disabled

11.1.3.1.8 PCIE PME message and error detect interrupt detect disable register (GPR_REG6)

11.1.3.1.8.1 Offset

Register	Offset
GPR_REG6	18h

11.1.3.1.8.2 Diagram



11.1.3.1.8.3 Fields

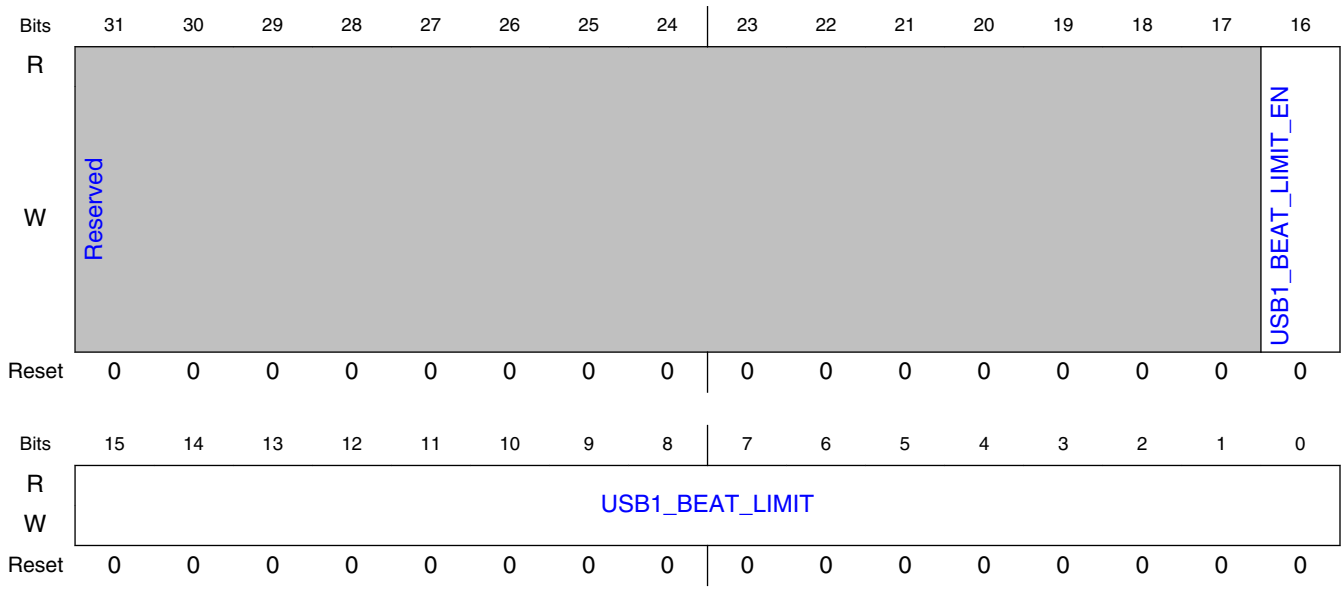
Field	Function
31-9 —	Reserved
8 ME_DIS	Multiple errors of same type detection disable 1'b1: Multiple errors of same type detection disabled. 1'b0: Multiple errors of same type detection enabled
7 PCT_DIS	completion detection disable 1'b1: Completion timeout detection disabled. 1'b0: Completion timeout detection enabled.
6 PCAC_DIS	Completer abort detection disable. 1'b1: Completer abort detection disabled. 1'b0: Completer abort detection enabled
5 CDNSC_DIS	Completion with data not succsessful detection disable 1'b1: Completion with data not succsessful detection disabled. 1'b0: Completion with data not succsessful detection enabled.
4 UREP_DIS	Unsupported request in EP mode detection disable 1'b1: Unsupported request in EP mode detection disabled. 1'b0: Unsupported request in EP mode detection enabled.
3 PTO_DIS	PME turn off detect disabled. 1'b1: PME turn off detect disabled. 1'b0: PME turn off detect enabled.
2 HRD_DIS	Hot reset detect disable 1'b1: hot reset detect disabled. 1'b0: hot reset detect enabled.
1 LDD_DIS	Link down detect disable 1'b1: link down detect disabled. 1'b0: link down detect
0 LUD_DIS	Link up detect disable 1'b1: link up detect disabled. 1'b0: link up detect enabled

11.1.3.1.9 USB1 beat limit and enable (GPR_REG7)

11.1.3.1.9.1 Offset

Register	Offset
GPR_REG7	1Ch

11.1.3.1.9.2 Diagram



11.1.3.1.9.3 Fields

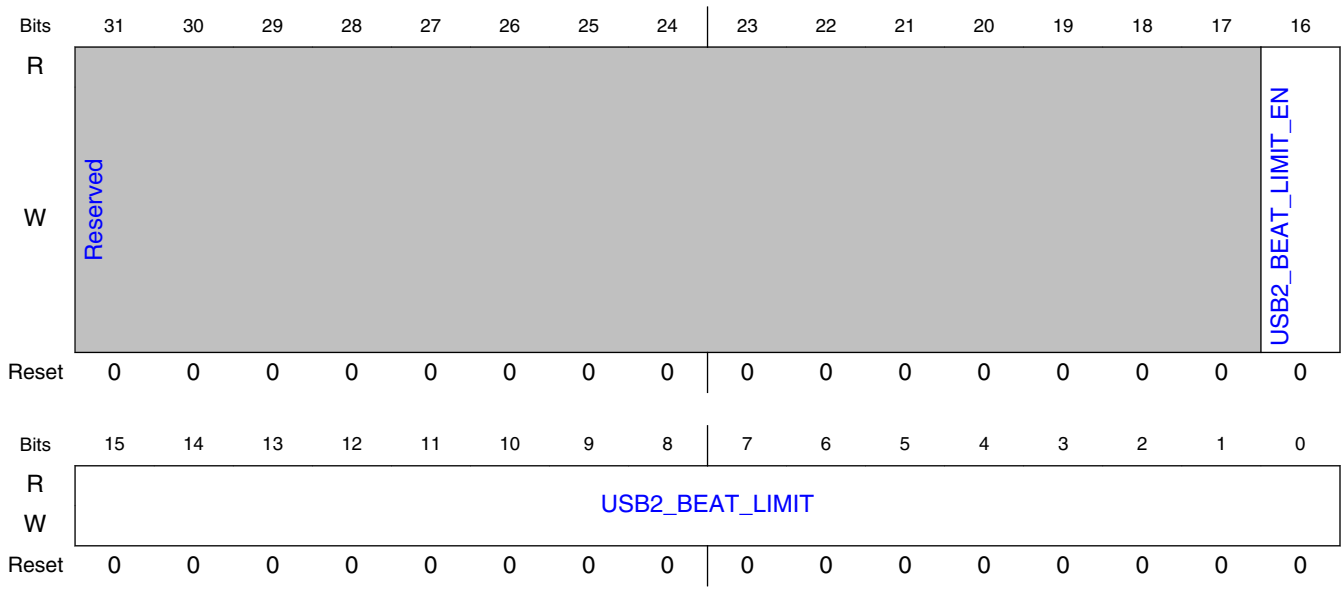
Field	Function
31-17 —	Reserved
16 USB1_BEAT_LIMIT_EN	USB1 beat limit enable
15-0 USB1_BEAT_LIMIT	USB1 beat limit number

11.1.3.1.10 USB2 beat limit and enable (GPR_REG8)

11.1.3.1.10.1 Offset

Register	Offset
GPR_REG8	20h

11.1.3.1.10.2 Diagram



11.1.3.1.10.3 Fields

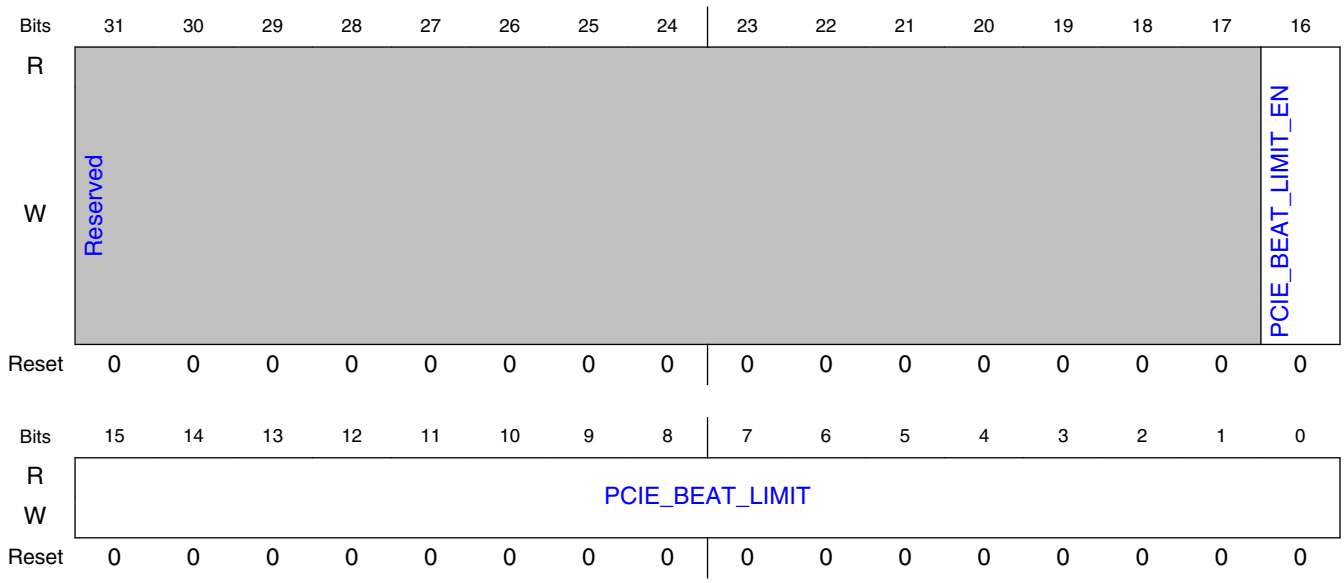
Field	Function
31-17 —	Reserved
16 USB2_BEAT_LIMIT_EN	USB2 beat limit enable
15-0 USB2_BEAT_LIMIT	USB2 beat limit number

11.1.3.1.11 PCIE beat limit and enable (GPR_REG9)

11.1.3.1.11.1 Offset

Register	Offset
GPR_REG9	24h

11.1.3.1.11.2 Diagram



11.1.3.1.11.3 Fields

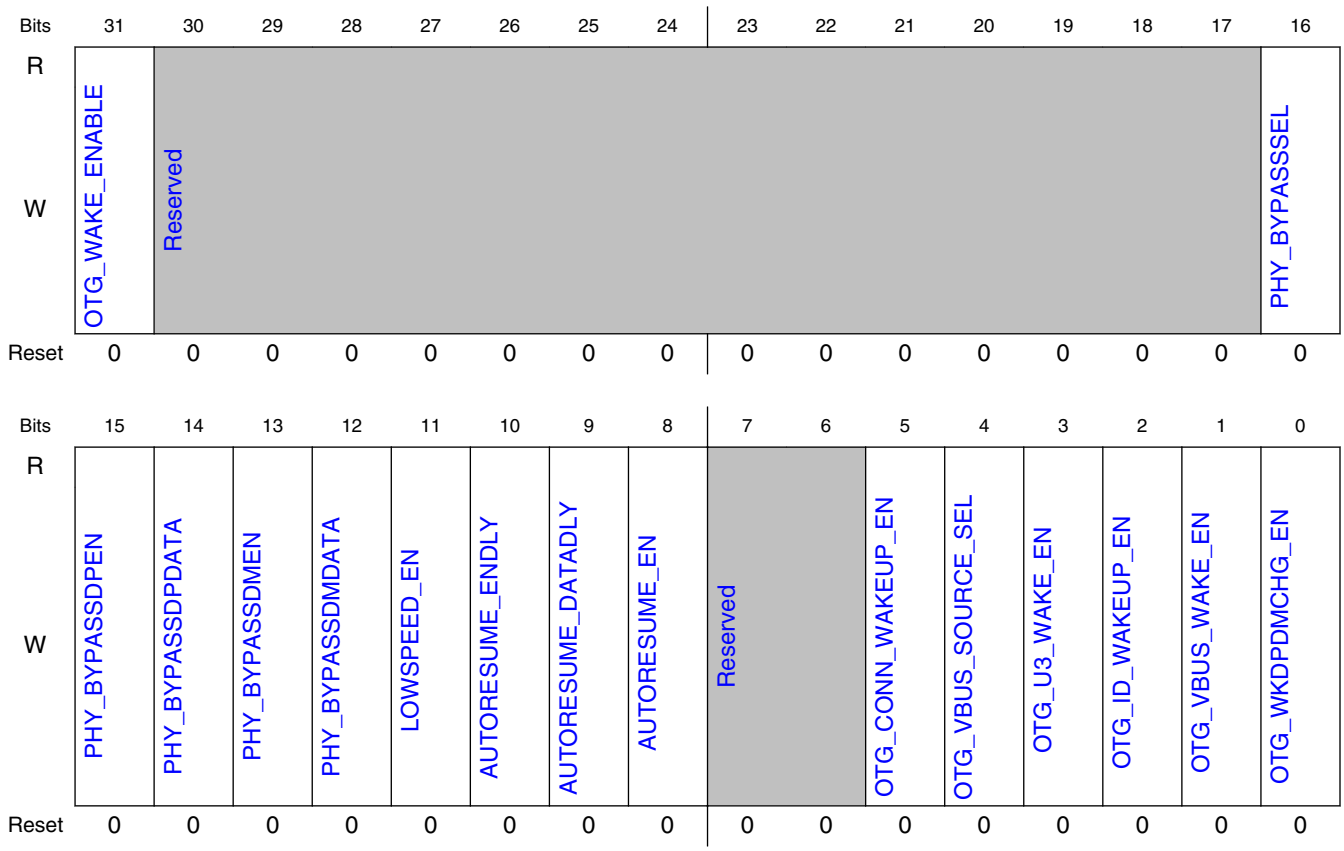
Field	Function
31-17 —	Reserved
16 PCIE_BEAT_LIMIT_EN	PCIE beat limit enable
15-0 PCIE_BEAT_LIMIT	PCIE beat limit number

11.1.3.1.12 Register for USBa wakeup (USB1_WAKEUP_CTRL - USB2_WAKEUP_CTRL)

11.1.3.1.12.1 Offset

Register	Offset
USB1_WAKEUP_CTRL	100h
USB2_WAKEUP_CTRL	108h

11.1.3.1.12.2 Diagram



11.1.3.1.12.3 Fields

Field	Function
31 OTG_WAKE_ENABLE	Global wakeup interrupt enable. Used to clear interrupt. Default 0, same as WIE before.
30-17 —	Reserved
16 PHY_BYPASSSEL	Transmitter Digital Bypass Select
15 PHY_BYPASSDPEN	DP Transmitter Digital Bypass Enable
14 PHY_BYPASSDPDATA	Data for DP Transmitter Digital Bypass
13	DM Transmitter Digital Bypass Enable

Table continues on the next page...

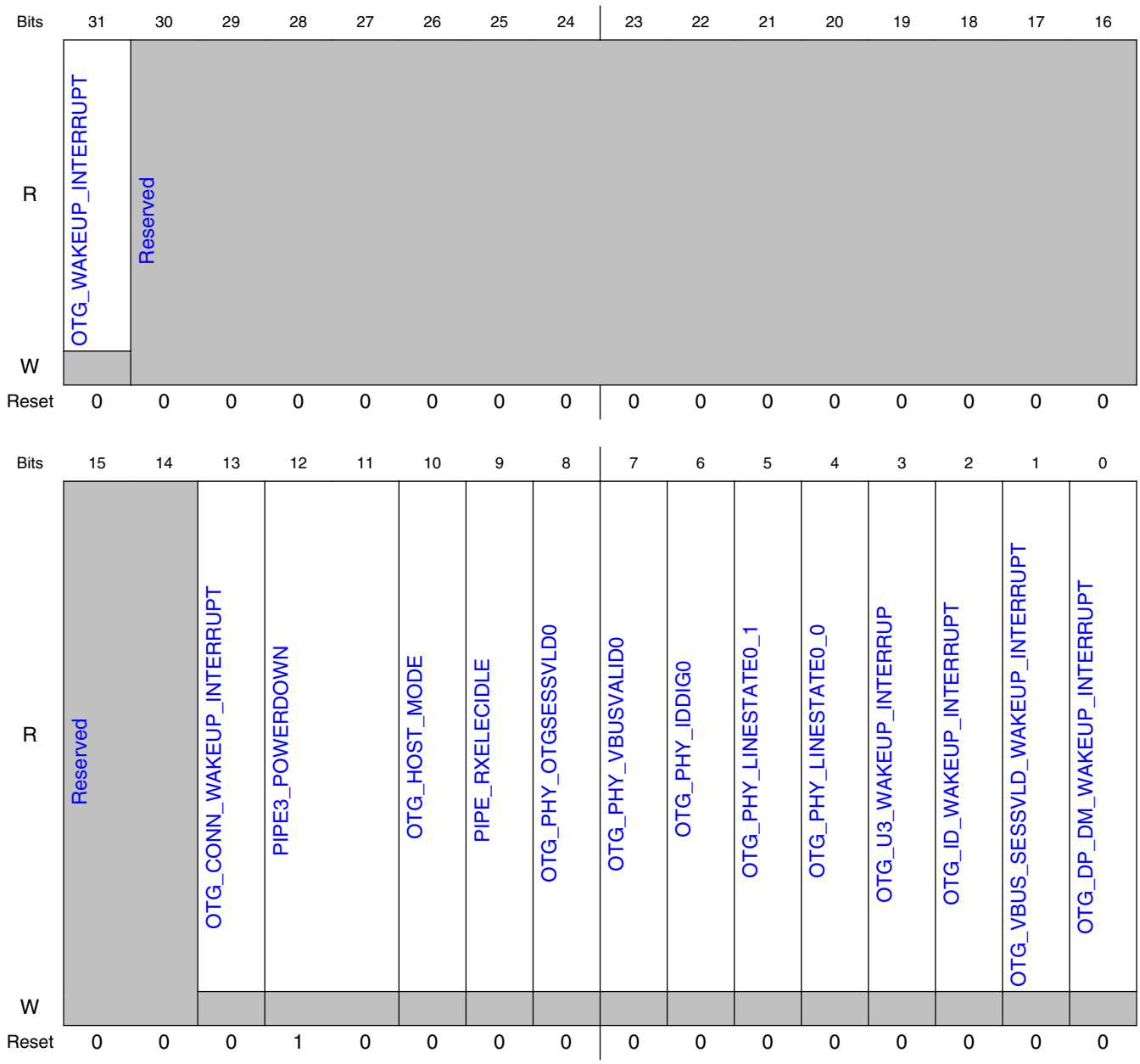
Field	Function
PHY_BYPASSDMEN	
12 PHY_BYPASSDMDATA	Data for DM Transmitter Digital Bypass
11 LOWSPEED_EN	Enable lowspeed autoresume feature 0b - full/high speed 1b - lowspeed
10 AUTORESUME_ENDLY	Tuning the timing between dp/dm data and enable signal when autoresume finish. {autoresume_endly, autoresume_datadly}= 00 or 11: data and enable pulldown together 01: enable signal pull down later than data pin one 32k clock cycle 10: enable signal pull down earlier than data pin one 32k clock cycle
9 AUTORESUME_DATADLY	Please refer to the bitfield description for AUTORESUME_ENDLY.
8 AUTORESUME_EN	Enable autoresume feature. When using phy_bypass signal, it should be 1'b0. 0b - disable. 1b - enable.
7-6 —	Reserved
5 OTG_CONN_WAKEUP_EN	Enable signal for wakeup from connection or disconnection, only for superspeed.
4 OTG_VBUS_SOURCE_SEL	otg_vbus_source_sel 0b - select vbus_valid 1b - select sessvld
3 OTG_U3_WAKE_EN	Enable signal for wake up from u3 state, only for superspeed 0b - disable 1b - enable
2 OTG_ID_WAKEUP_EN	Enable signal for wake up from id change 0b - disable 1b - enable
1 OTG_VBUS_WAKE_EN	Enable signal for wake up from vbus valid or session valid changes 0b - disable 1b - enable
0 OTG_WKDPDMCHG_EN	Enable signal for wake up from dp/dm change. 0b - disable 1b - enable

11.1.3.1.13 Status of USBa wakeup (USB1_WAKEUP_STATUS - USB2_WAKEUP_STATUS)

11.1.3.1.13.1 Offset

Register	Offset
USB1_WAKEUP_STATUS	104h
USB2_WAKEUP_STATUS	10Ch

11.1.3.1.13.2 Diagram



11.1.3.1.13.3 Fields

Field	Function
31 OTG_WAKEUP_INTERRUPT	Same as WIR before, it's OR of all wakeup interrupt, then AND with otg_wakeup_enable. All separate wakeup interrupt status need to be registered by wakeup enable, so clear wakeup enable, the related wakeup interrupt will be cleared tens of us later.
30-14 —	Reserved

Table continues on the next page...

Field	Function
13 OTG_CONN_WAKEUP_INTERRUPT	Interrupt status of connection
12-11 PIPE3_POWERDOWN	Pipe powerdown
10 OTG_HOST_MODE	USB drd mode indicator 0b - device mode 1b - host mode
9 PIPE_RXELECIDLE	pipe_rxelecidel, wakeup source
8 OTG_PHY_OTGSESSVLD0	session valid, wakeup source
7 OTG_PHY_VBUSVALID0	vbus valid, wakeup source
6 OTG_PHY_IDDIG0	ID status, wakeup source
5 OTG_PHY_LINESTATE0_1	linestate[1], wakeup source
4 OTG_PHY_LINESTATE0_0	linestate[0], wakeup source
3 OTG_U3_WAKEUP_INTERRUP	Interrupt status of wakeup from u3 state
2 OTG_ID_WAKEUP_INTERRUPT	Interrupt status of wakeup from id
1 OTG_VBUS_SESSVLD_WAKEUP_INTERRUPT	Interrupt status of vbus or session valid signal. otg_vbus_wakeup_interrupt | otg_sessvld_wakeup_interrupt
0 OTG_DP_DM_WAKEUP_INTERRUPT	Interrupt status of dm or dp, otg_dp_wakeup_interrupt | otg_dm_wakeup_interrupt

11.2 Universal Serial Bus Controller (USB)

11.2.1 Overview

The USB module is a USB 3.0-compliant serial interface engine for implementing a USB interface. This module may be connected to an external port. Collectively the module and external port are called the USB 3.0 interface. USB 3.0 supports Super-speed (SS), High-speed (HS), Full-speed (FS), and Low-speed (LS) operations.

- The upper layer is common for USB 2.0 and USB 3.0 operation. This has the bus interface, buffer management block, list processor for scheduling, and control and status register (CSR) functions
- USB 2.0 PHY and MAC layers
- USB 3.0 PHY, LINK, and MAC layers

11.2.1.1 Block diagram

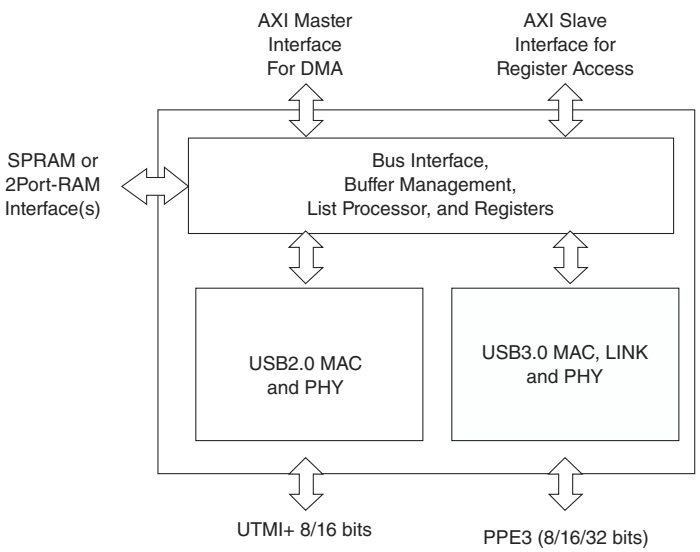


Figure 11-3. Block diagram

11.2.1.2 Features

The USB 3.0 module includes the following features:

- Complies with USB specification rev 3.0 (xHCI compatible)

- Supports operation as a standalone USB host controller
- USB dual-role operation and can be configured as host or device
- Super-speed (5 Gbit/s), high-speed (480 Mbit/s), full-speed (12 Mbit/s), and low-speed (1.5 Mbit/s) operations.
- Supports operation as a standalone single port USB
- Supports eight programmable, bidirectional USB endpoints
- Supports system memory interface with 40-bit addressing capability

11.2.2 Functional description

11.2.2.1 Modes of operation

The USB 3.0 module operates in following modes.

- Host Mode: SS/HS/FS/LS
- Device Mode: SS/HS/FS

11.2.2.2 System memory descriptor and data buffers

The software creates transfer request blocks (TRBs), and four DWORDs each, that point to the data buffers. Normally, the TRBs are allocated consecutively in system memory; only the data buffers can be scattered. In the case of a circular buffer, the link-TRB points to the next TRB. Once the TRBs and data buffers are set up in the system memory, the software driver issues a start transfer command that points to the location of the first TRB in the system memory to start the DMA operation. TRBs, though small (only four DWORDs), provide a rich set of features for the software to schedule transfers, isochronous, control, interrupt moderation, and so on.

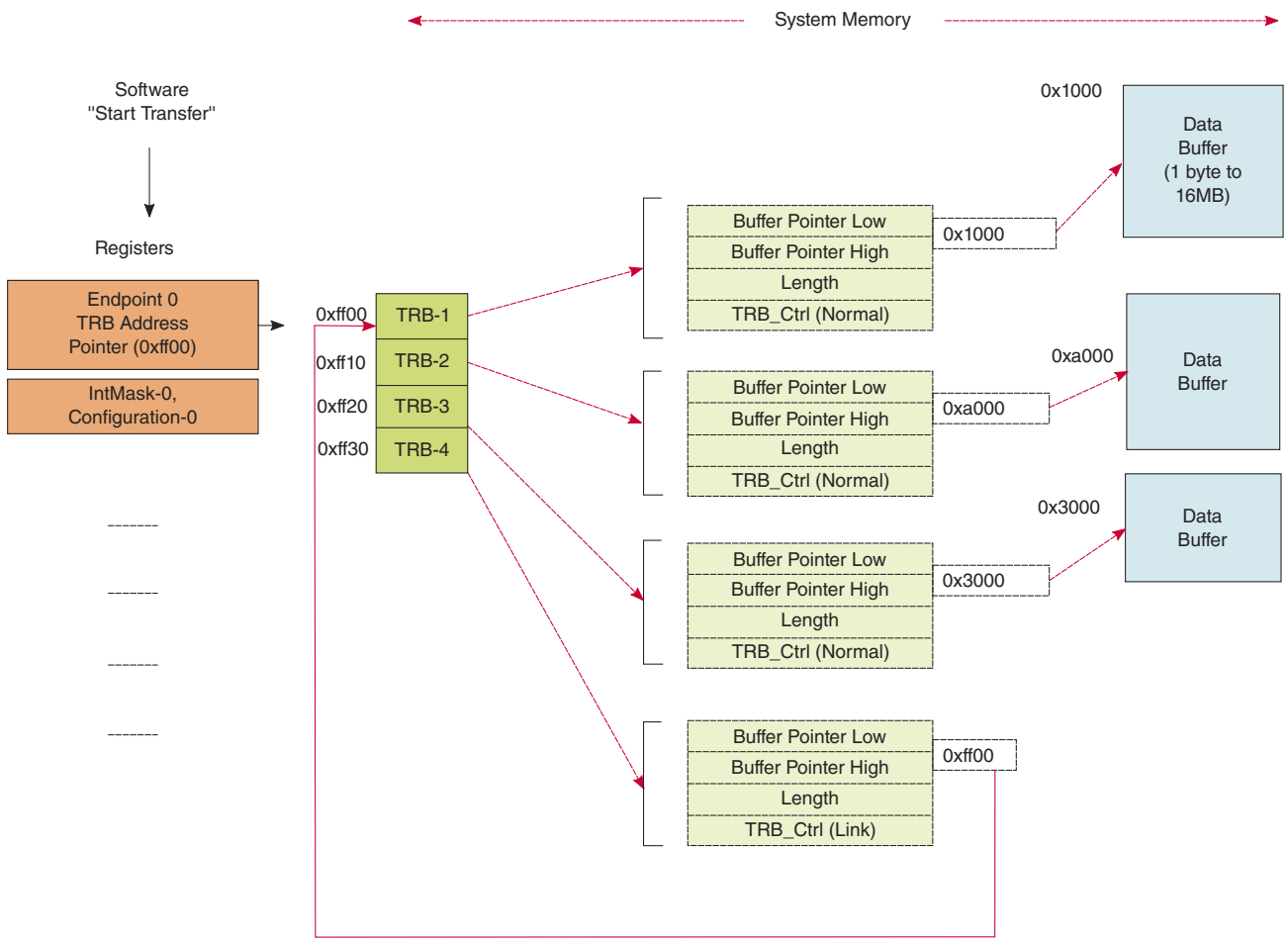


Figure 11-4. System Memory Descriptor and Data Buffers

11.2.2.3 Device descriptor structures

Device mode transfer request blocks (TRBs) are small 4 DWORDs and at the same time provide a rich set of features so that the software can efficiently manage the USB core, memory buffers, and MIPS requirements.

The following is a list of device mode TRB characteristics:

- TRBs provide support for scattered data structures. The scattered data buffers can be zero bytes to 16 MB in length.
- TRBs must be placed in system memory aligned to a 16-byte boundary.
- TRBs are kept in linear memory to enhance descriptor caching performance. If the data buffers are small (as in an Ethernet application) the core can efficiently collect the scattered data to build USB packets without wasting bus efficiency that collecting scattered descriptors requires.

- Supports TRB linking. This allows software to set up a circular ring of TRBs with the last TRB linking to the first one.
- Supports system memory interface with 40-bit addressing capability
- Supports byte-aligned buffers for each TRB of a transfer. This feature prevents the need for buffer copying in cases where the USB device driver receives unaligned data buffers from other applications (for example, Ethernet). Whenever the application controls buffer allocation, it must allocate SoC bus width and burst-aligned buffers to facilitate efficient bus and memory utilization. For transmitting, the core supports byte-aligned buffers on all TRBs.

  For example: To use 16 bursts in a 64-bit system, you must try to allocate buffers that are 16 * 8 = 128 bytes aligned. This is the normal buffer structure because Linux-like OSs allocate 4 KB buffers. SDR/DDR memory controllers also provide better performance when requests are burst aligned.

- Supports software queueing of multiple USB transfers (LST bit for IN/OUT transfers and CSP bit for OUT transfers control this function).
- Provides interrupt moderation capability, allowing software to selectively enable events on TRB completion, as controlled by the IOC (Interrupt on completion) bit. The IOC event is also used by software to reallocate the released buffers, allowing software to re-use just a few buffers in a circular fashion. This helps when the memory is limited but have enough MIPS to process interrupts. Larger buffers can be allocated to reduce the number of interrupts.
  - In USB 3.0 larger transfers are recommended because the raw transfer rate is almost 10 times faster, unlike USB 2.0, where drivers set up only 64 KB or 128 KB transfers. For example, a 64 KB transfer that takes 1.3 ms in USB 2.0 requires only 164 µS in USB 3.0. If you set up buffers of 64 KB and enable the transfer completion event, then you receive an interrupt every 164 µS.
- Supports streaming (Stream ID field used for this purpose).

11.2.2.3.1 Structures

This figure shows the control and status field of a transfer request block.

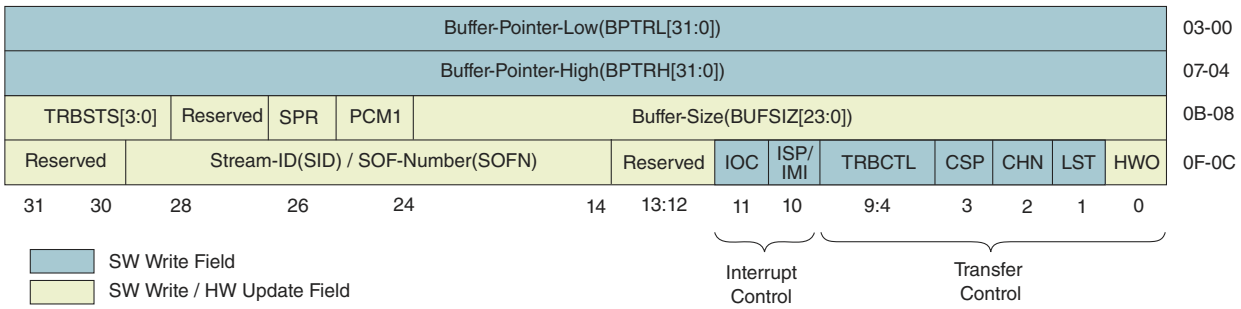


Figure 11-5. TRB control and status fields

Table 11-1. Device descriptor structure field definitions

Field	Description	Hardware access
DW 03-00		
31:0	Buffer pointer low (BPTRL) Data buffer pointer to low 32 bit address (BPTR[31:0]). Hardware may also update this field (implementation specific).	R/W
DW 07-04		
31:0	Buffer pointer high (BPTRH) Data buffer pointer to high 32-bit address (BPTR[63:32]).	R/W
DW 0B-08		
31:28	TRB status (TRBSTS) Hardware updates this field with transfer status information before releasing the TRB. • 4'h0: OK • 4'h1: MissedIsoc: Isochronous interval missed or incomplete • 4'h2: SetupPending - During the current control transfer data/status phase, another SETUP was received. • 4'h4: TransferInProgress - During the current transfer, an end transfer command was received.	R/W
27	Reserved. Hardware may update these bits for internal usage.	R/W
26	SPR • If GUCTL1[DEV_TRB_OUT_SPR_IND] is 1'b1, this field indicates the short packet received for the last processed TRB for the transfer. If a short packet has been received, this TRB field is set during an OUT transfer TRB write back if this is the last TRB used for that transfer descriptor. • If GUCTL1[DEV_TRB_OUT_SPR_IND] is 1'b0, this field is reserved. Hardware may update these bits for internal usage.	R/W
25:24	Packet count M1 (PCM1) For high-speed, High bandwidth isochronous IN endpoints, this field in an Isoc-First TRB represents the total number of packets in the Buffer Descriptor minus 1.	R/W
23:0	Buffer size (BUFSIZ) If CHN=0 and HWO=0 for the TRB, this field represents the total remaining Buffer Descriptor buffer size in bytes. Valid Range: 0 bytes to (16 MB - 1 byte). The hardware decrements this field to represent the remaining buffer size after data is transferred.	R/W

Table continues on the next page...

Table 11-1. Device descriptor structure field definitions (continued)

Field	Description	Hardware access
	For a Link TRB, the buffer size should be "0".	
DW 0F-0C		
31:30	Reserved.	R/W
29:14	Stream ID / SOF Number For stream-based bulk endpoints: The Stream ID of the transfer this TRB is associated with. Stream ID must be the same in all TRBs (R/W). For isochronous endpoints: The (micro)frame number in which the last packet of this TRB's buffer was transmitted or received (debug purposes only) (RO).	R/W
13:12	Reserved.	R/W
11	Interrupt on Complete (IOC) When IOC is set in a TRB, and once the transfer for this buffer is completed, the core will issue XferInProgress event with IOC bit set in the event's status. This indicates the buffer is available for software to reuse or release.	R
10	Interrupt on Short Packet / Interrupt on Missed ISOC (ISP/IMI) Applicable to OUT endpoints when a short packet is received, and CSP=1 and LST=0. If this bit is 1, the core generates an XferInProgress event. For Isochronous endpoints: If this bit is 1, the core generates an XferInProgress event when the interval represented by the Buffer Descriptor completes with a "Missed Isoc" status.	R
9:4	TRB Control (TRBCTL) Indicates the type of TRB: 1: Normal (Control-Data-2+ / Bulk / Interrupt) - Set TRBCTL to 1 for all TRBs used in data stage except the first TRB 2: Control-Setup 3: Control-Status-2 - Set TRBCTL to 3 for a SETUP request without data stage 4: Control-Status-3 - Set TRBCTL to 4 for a SETUP request with data stage 5: Control-Data - Set TRBCTL to 5 for the first TRB of a data stage 6: Isochronous-First - Set TRBCTL to 6 for the first TRB of a Service Interval 7: Isochronous 8: Link TRB Others: Reserved	R
3	Continue on Short Packet (CSP) Applicable to OUT endpoints only when a short packet is received. If this bit is 1, the core will continue to the next Buffer Descriptor. This setting is required for isochronous endpoints. If this bit is 0, the core will generate an XferComplete event and remove the stream.	R
2	Chain Buffers (CHN) Applicable to IN and OUT endpoints. Set to 1 by software to associate this TRB with the next TRB. A Buffer Descriptor is defined as one or more TRBs. The CHN bit is used to identify the TRBs that comprise a Buffer Descriptor. The CHN bit is always 0 in the last TRB of a Buffer Descriptor and when the LST field is set to 1.	R

Table continues on the next page...

Table 11-1. Device descriptor structure field definitions (continued)

Field	Description	Hardware access
1	Last TRB (LST) Indicates this is the last TRB in a list. After completing the transfer for the associated buffer the core will stop the transfer for the endpoint / bulk-stream and issues an XferComplete event. The stream is automatically removed by the hardware.	R
0	Hardware Owner of Descriptor (HWO) Indicates that hardware owns the TRB. Software sets this bit to 1 when it creates the TRB, and cannot modify it until hardware resets this bit to 0. However, there are exceptions for short packets on OUT endpoints and Link TRBs. Because the hardware autonomously checks this bit to determine if the entire TRB is valid, software must set this field to '1' after preparing the other three DWORDs of the TRB with valid information.	R/W

11.2.2.3.1.1 Normal (Control-Data/Bulk/Interrupt), Isochronous, and Status Transfer Request Block Structure

The Normal TRB is used for Bulk/Control-Data/Interrupt endpoint transfers. The data buffers can be scattered anywhere and each may have different sizes. The TRB Buffer Pointer and the Buffer Size fields point to buffer address and size respectively. The Stream ID for bulk endpoints will be programmed by the application.

For isochronous endpoints, the first TRB in a service interval must have the Isoc-First type, the last TRB in a service interval must have CHN=0, and any other TRBs have CHN=1. The starting (micro)frame time is communicated via the Start Transfer command, and the core tracks the (micro)frame times of the subsequent Buffer Descriptors.

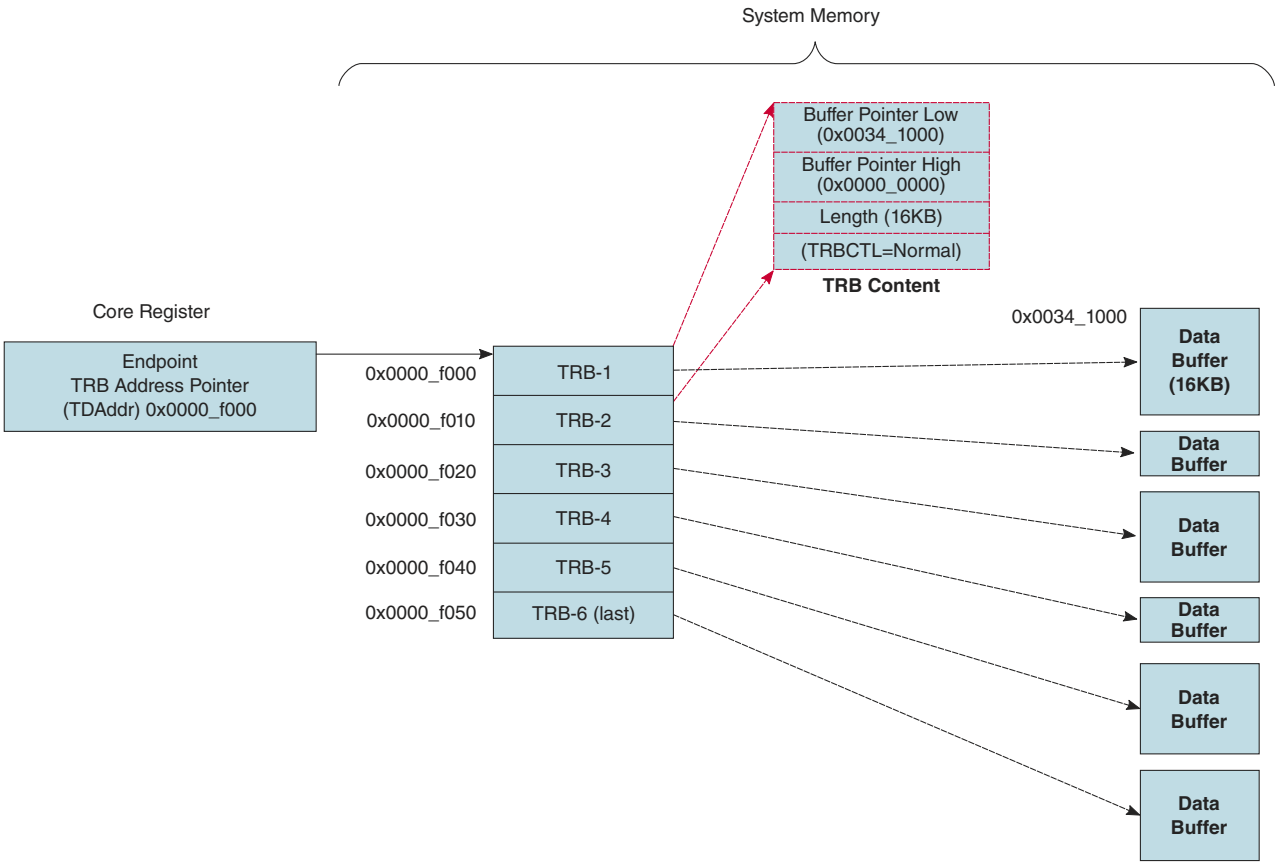


Figure 11-6. Normal (Control-Data/Bulk/Interrupt) Descriptor Structure

11.2.2.3.1.2 Setup and Status TRB Structure

To receive a SETUP packet, the driver queues up a single Setup TRB, whose buffer pointer value may be set to any address, including the address of the TRB. The buffer size must be set to 8. The core will write the 8 bytes of the received SETUP to the address requested. If the address of the TRB is used, there is no need for a separate data buffer to receive a SETUP packet. After completing Setup stage, driver will schedule Data stage and Status stage transfers. For more information, see Control transfer programming model.

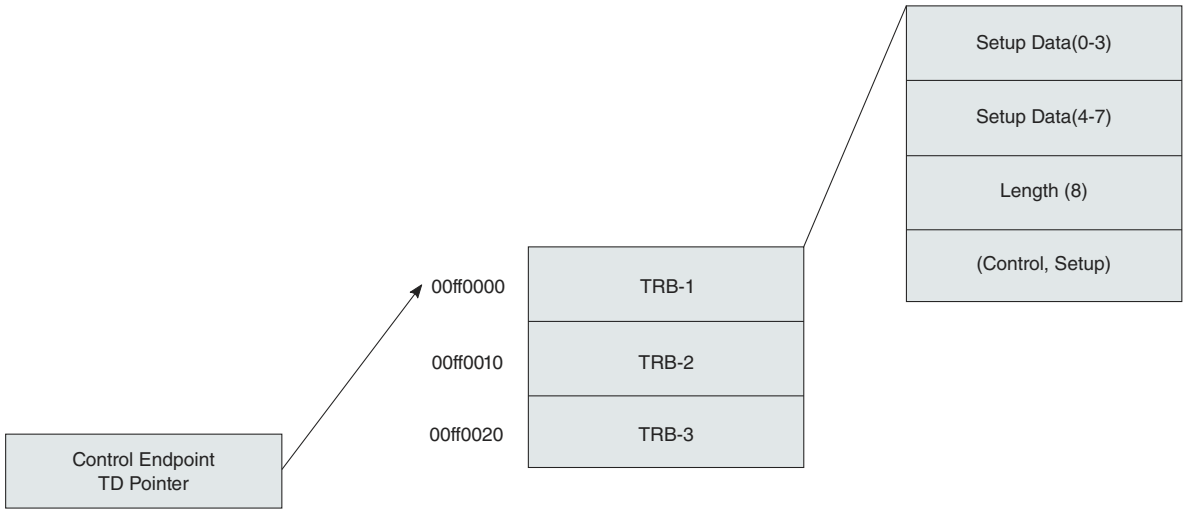


Figure 11-7. SETUP Descriptor Structure with Buffer Pointing to Setup TRB

After interpreting the SETUP bytes, software will determine if the next stage of the control transfer is a data stage or status stage.

If the SETUP bytes require a 3-stage control transfer, the TRB type used in the Data stage must be Control- Data for the first TRB of the Buffer Descriptor. When the host moves on to the Status stage, the TRB Type must be Control-Status-3.

If the SETUP bytes require a 2-stage control transfer, the TRB Type must be Control-Status-2.

The Status TRB is a zero-length TRB, with an unspecified Buffer Pointer value and a Buffer Size of zero. There is no data buffer associated with a Status TRB.

11.2.2.3.1.3 Link TRB Structure

The Link TRB is used to link back to the starting TRB for reusing TRBs in a circular fashion.

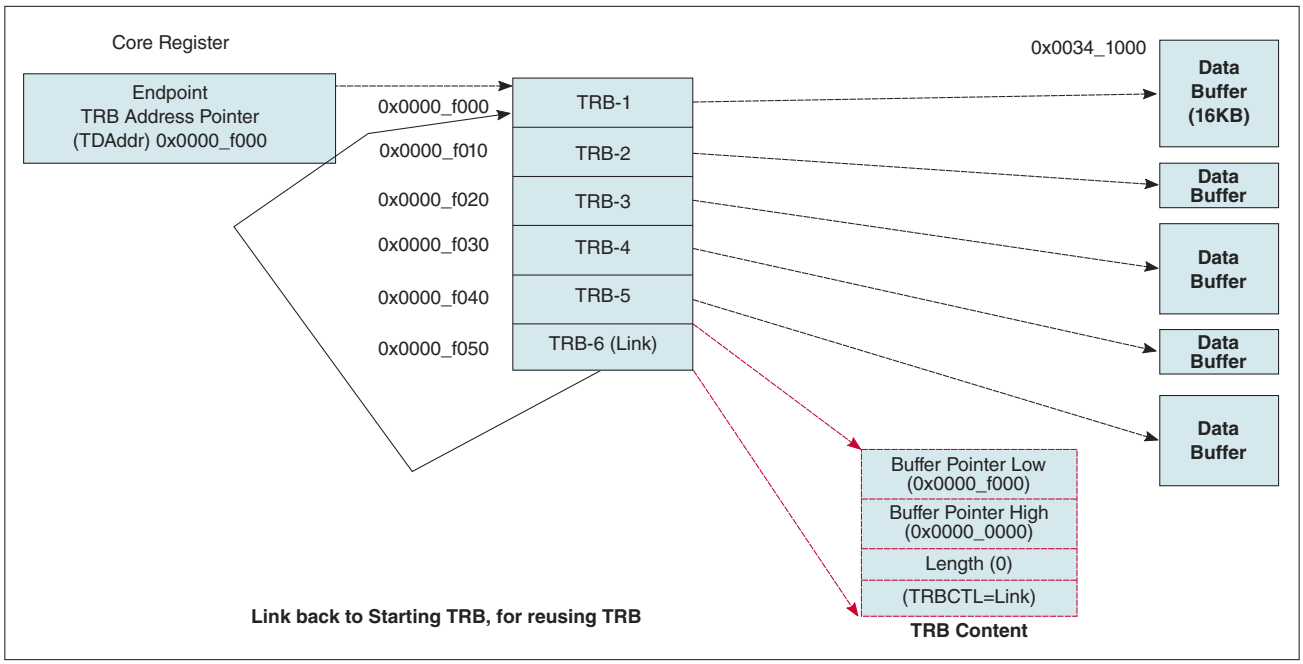


Figure 11-8. Link TRB Structure

If software prepares a circular TRB list which is shorter than the number of cached TRBs (four), the core automatically detects the loop and will not re-fetch a TRB that has already been fetched. For example, for four cached TRBs the software setup three TRBs in the following way:

- TRB-1: Normal, HWO=1
- TRB-2: Normal, HWO=1
- TRB-3: Link to TRB 1

The core will fetch TRBs 1, 2, and 3. Then the core will follow the link to TRB-1, note that its address is the same as the TRB-1 that has already been fetched into the cache, and will temporarily stop fetching TRBs. When TRB-1 completes due to traffic on the USB, the core will write TRB-1 back to memory with the HWO field set to '0', generate a XferInProgress event if necessary, and will automatically attempt to fetch TRB-1 again. In most cases, the core will see the HWO field set to '0' and will stop fetching until software updates the TRB, sets the HWO field back to '1', and issues an Update Transfer command.

11.2.2.3.1.4 Chaining buffers (CHN) and interrupt on completion (IOC) usage

This figure shows a chaining buffer example for an isochronous IN and a bulk IN transaction.

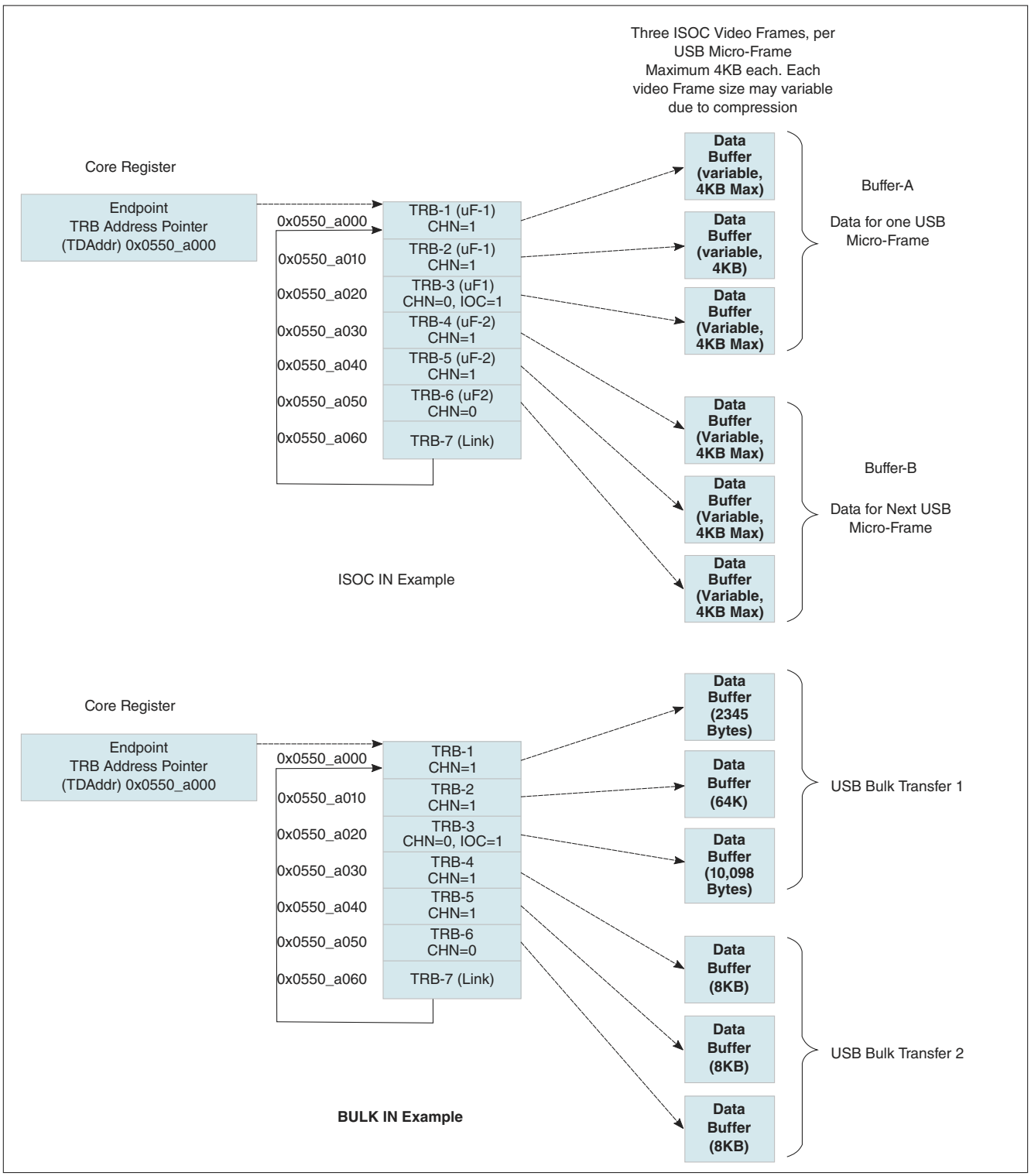


Figure 11-9. Chaining buffers for isochronous and bulk IN

In the isochronous IN application shown in the above figure, there are three video frames to be sent to the host in each microframe. Each video buffer size is maximum of 4 KB and the size may vary for each microframe depending on the compression. The CHN bit

in the TRB-3 is 0, which indicates this is the last buffer of a transfer for the given microframe. The device schedules TRB-1, 2, and 3 in the first bInterval. Similarly, TRB-6 indicates (CHN=0) microframe boundary, and TRB-4, 5, and 6 will be sent during the next bInterval.

For the last packet of last transfer in a microframe, the device internally sets last packet flag when responding at SuperSpeed for isochronous IN transfers.

The application can also use the IOC bit to receive an interrupt when the buffer-A transfer is completed, so it can use buffer-A to send data on the bInterval following the next bInterval. As long as the driver can service the interrupt and set up data before another bInterval, the USB transfers can continue without interruption. Depending upon the system interrupt latency, the buffer size can be adjusted.

If interrupt latency is high occasionally, the software may not have time to set up the linked TRB before the next bInterval. In this case, the core, on seeing HWO bit set to 0 in the TRB, stops processing further TRBs of this endpoint. When hardware receives the Update Transfer command from software, it will re-fetch the TRB. In the case of bulk application, the core will issue the NRDY signal. In isochronous transfers, zero- length packets are sent to the host until software enables the transfer again.

In the bulk IN example, the CHN bit indicates USB Transfer boundary. During a TRB transfer, CHN indicates whether to send the bytes from next TRB buffer as part of the current transfer. For example, even though TRB-1 has 2,345 bytes, the last 297 (2345 - 2 * 1024 = 297) bytes will be combined with the data in TRB-2 and sent as a 1,024 byte packet on the USB since CHN=1. On the other hand, when there is a short packet left in TRB-3, the short packet will be sent separately, since CHN=0.

Figure 11-10 and Figure 11-11 shows examples of chaining buffers, with the fetch on the SoC bus and the transfer on the USB.

Figure 11-12 shows a bulk OUT application where two transfers are set up. The device writes data into TRB-1 and TRB-2 for the first transfer. The CHN bit (CHN=0) in TRB-2 indicates that this is the last buffer of the transfer, then TRB-3 and TRB-4 are written for the second transfer. Similarly, the CHN bit (CHN=0) in TRB-4 indicates the second transfer boundary.

If an interrupt latency is high, software may not have time to set up the TRB. In this case, the core, on seeing the HWO bit set to 0 in the TRB, will stop processing further TRBs for the endpoint. Hardware will re-fetch the TRB when software issues the update transfer command. In the case of bulk, control, or interrupt endpoints, the core will issue NRDY. In the case of isochronous endpoints, packets will be dropped until software enables the transfer again.

In the bulk OUT example below, when a short packet is received for TRB-1, TRB-3 will be used for the next OUT transfer and TRB-2 will be closed.

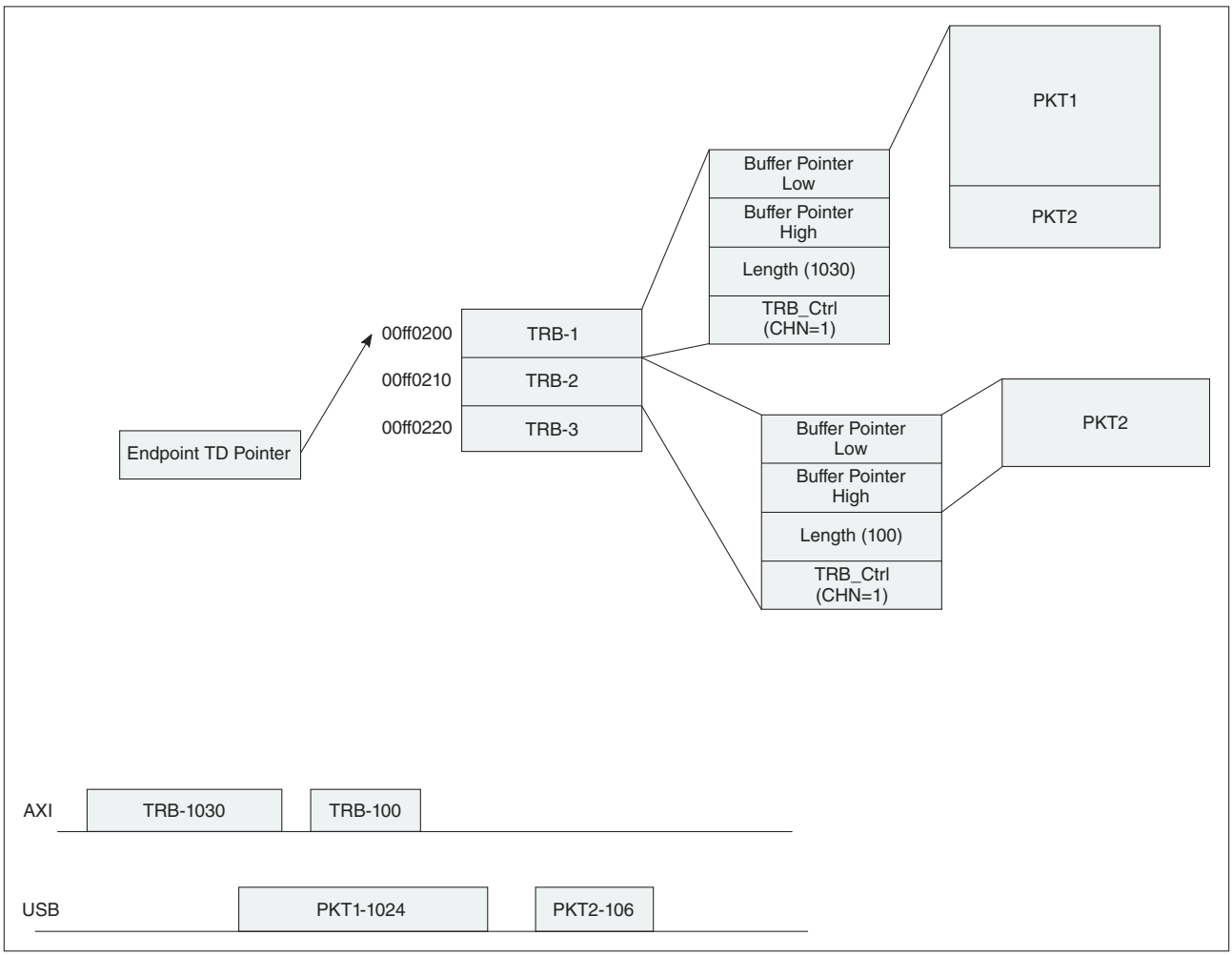


Figure 11-10. Chaining buffers, example 1

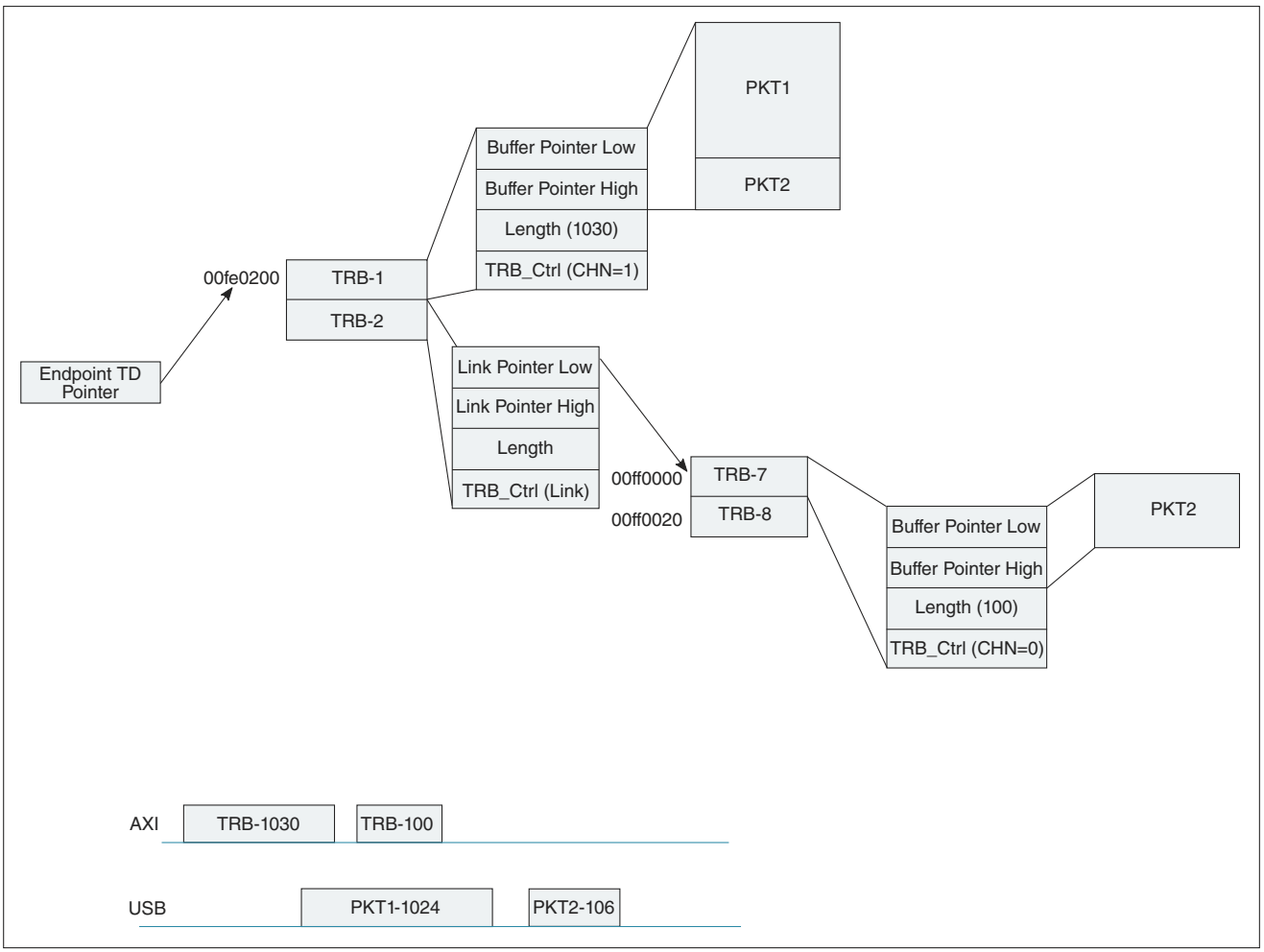


Figure 11-11. Chaining buffers, example 2

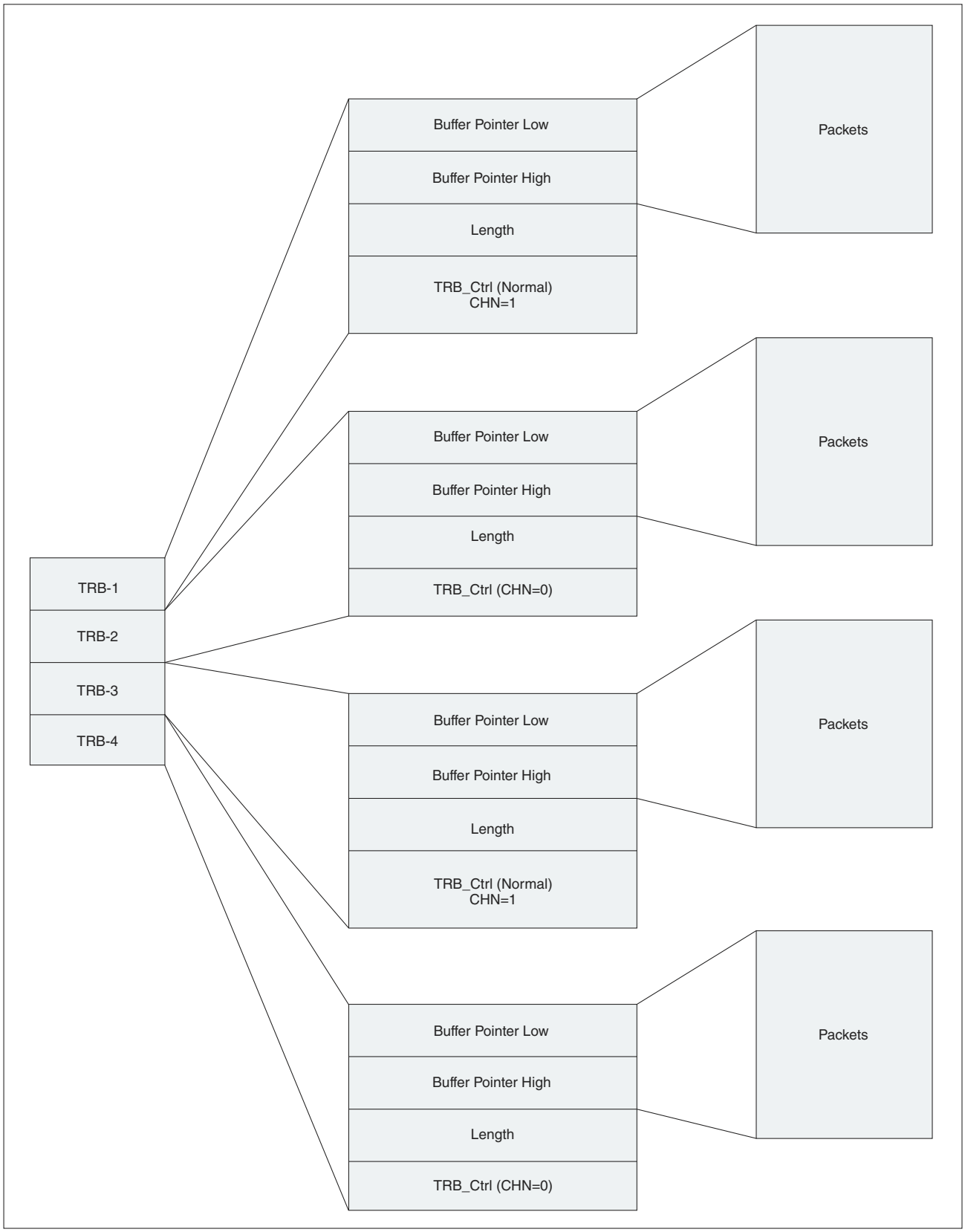


Figure 11-12. Bulk OUT with two transfers

11.2.2.3.1.5 Interrupt on Short Packet (ISP) and Continue on Short Packet (CSP) Usage

These two bits are used to schedule single or multiple OUT transfer. In most applications where only one OUT transfer per endpoint is scheduled, the software always sets ISP=1 and CSP=0. If the device receives a short packet, then it indicates a USB transfer completion through XferComplete events.

In an Ethernet-over-USB application, where software knows it is going to receive short packets, it can set up multiple transfer size buffers in one step by setting CSP=1 in all the TRBs. On a short packet, the device will update the byte count and move to next TRB. Software can also set ISP once in n number of TRBs to receive an XferInProgress event so it can process the previous short packets.

The following example shows software setting multiple 1500-byte Ethernet transfers and enabling the XferInProgress event once for four Ethernet packets to reduce the number of interrupts.

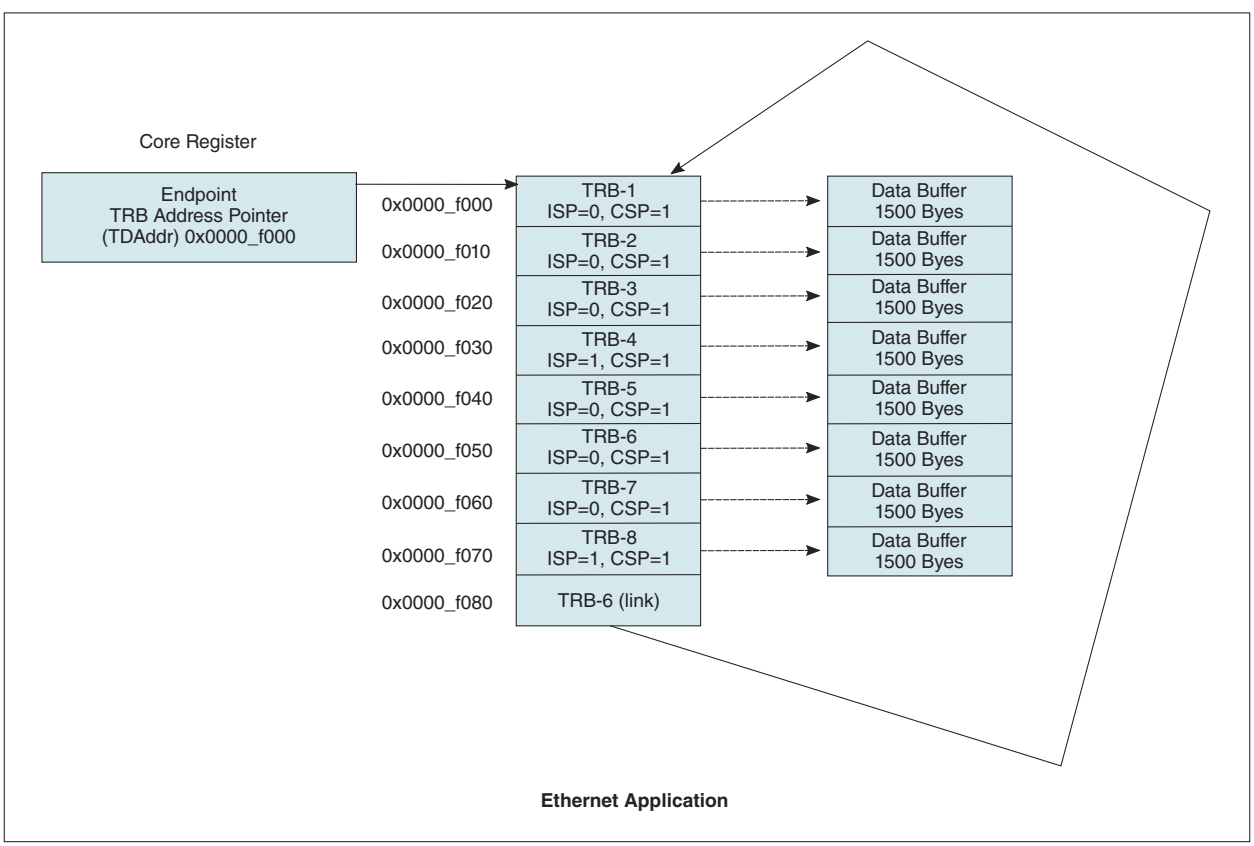


Figure 11-13. ISP and CSP Usage

11.2.2.3.1.6 Example of Setting Up TRBs

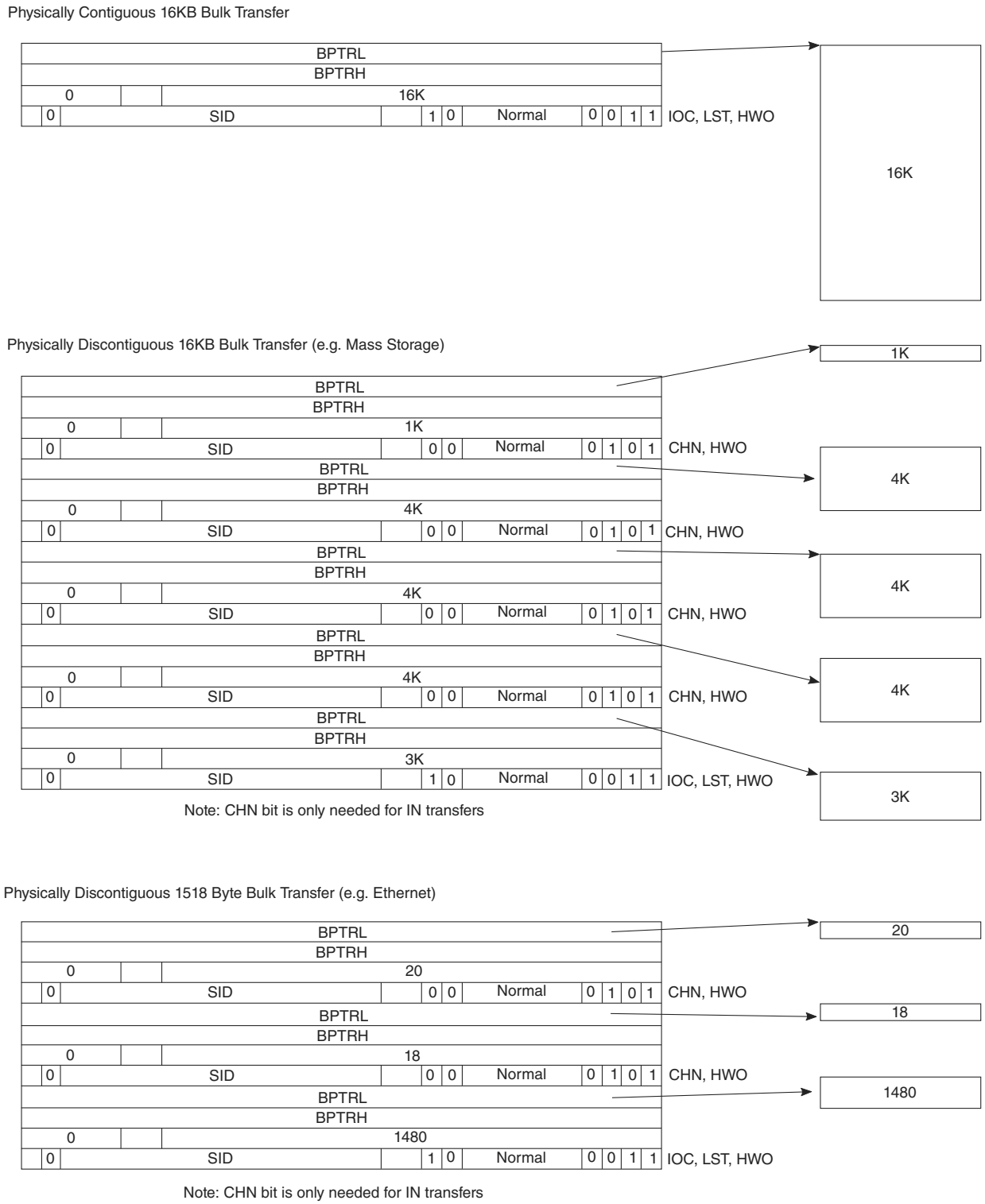


Figure 11-14. Bulk IN TRB Examples

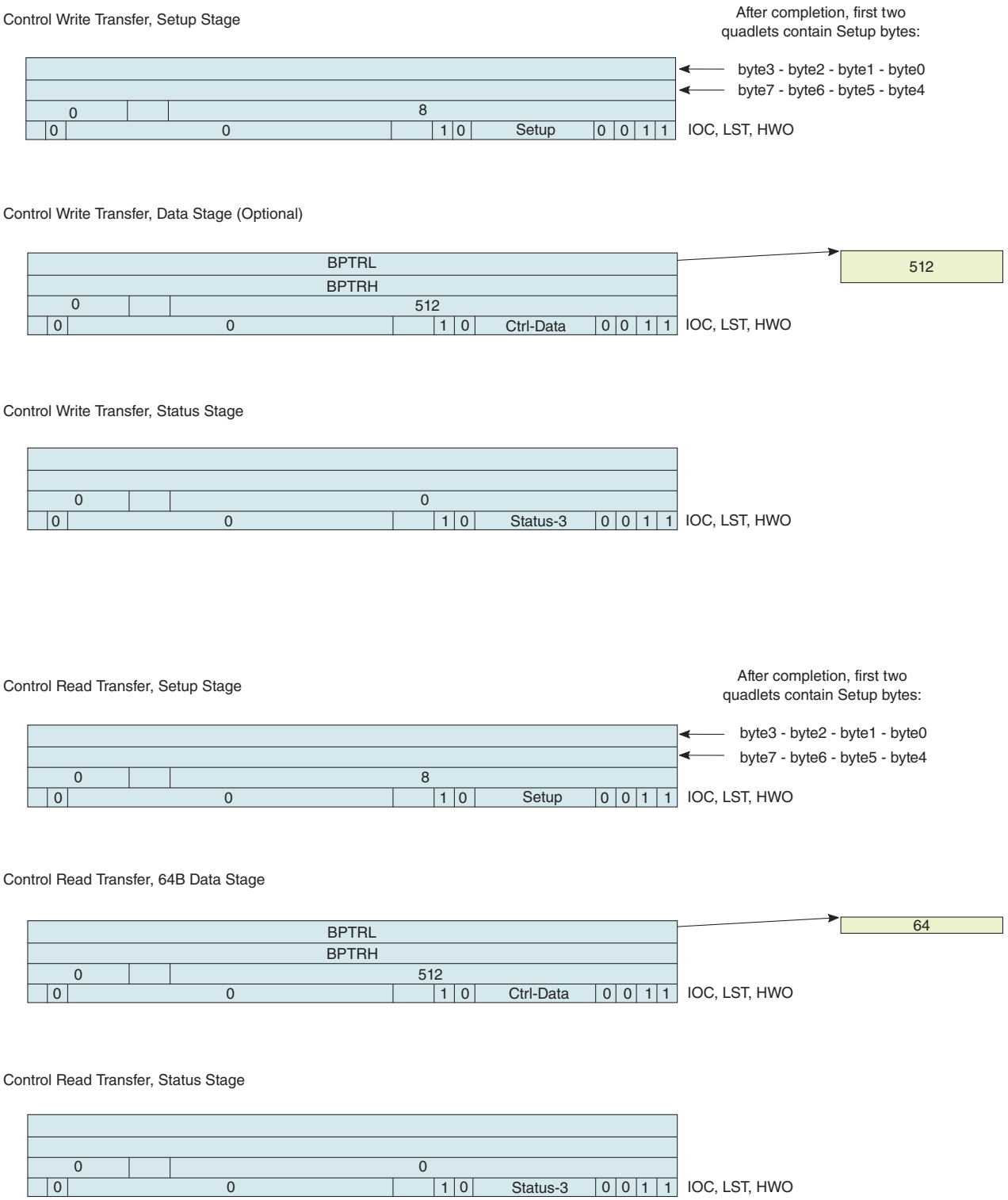


Figure 11-15. Setup/Control/Status TRB Examples

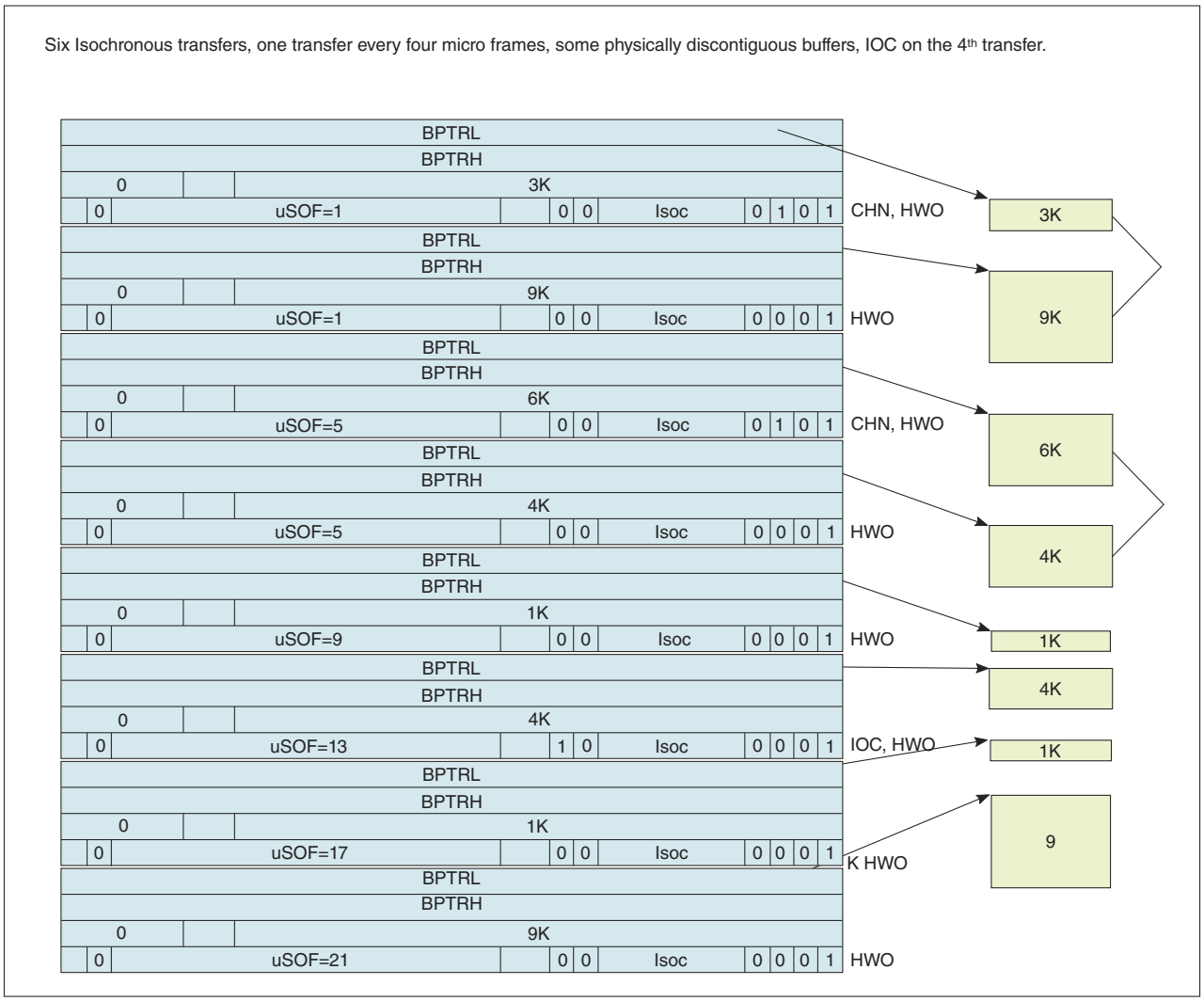


Figure 11-16. Isochronous IN TRB, Example 2

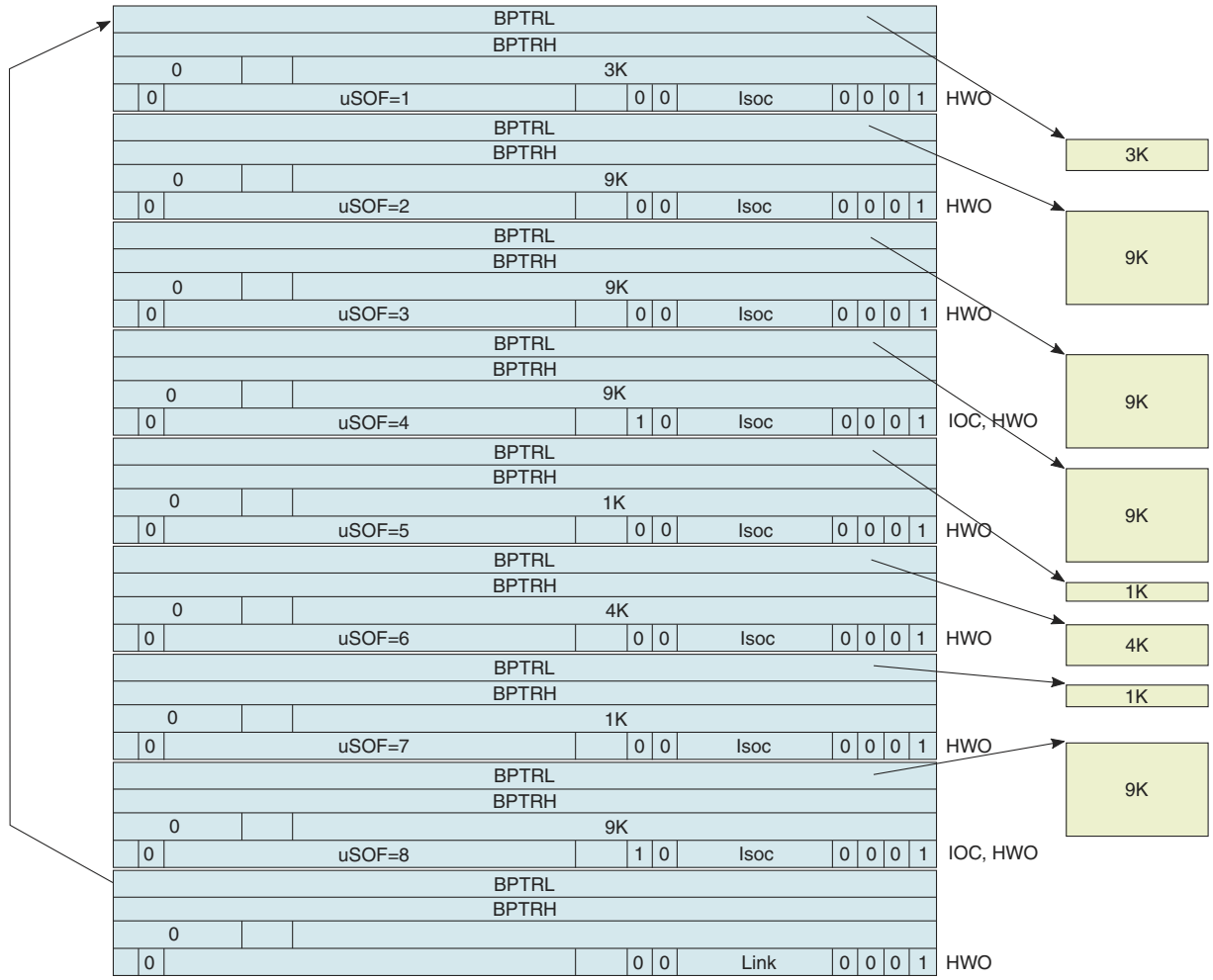


Figure 11-17. Isochronous IN TRB, Example 1

11.2.2.4 Device generic command structure

Software can program the controller using DGCMD and DGCMDPAR registers to send link management packets and notifications.

DGCMD[7:0] specifies the type of generic command that the software needs the controller to perform. This section discusses the commands present in the following table.

Any command that is not present in the table is considered Reserved.

Table 11-2. device generic command types

Command	Description
02h	Set Periodic Parameters

Table continues on the next page...

Table 11-2. device generic command types (continued)

Command	Description
	Parameter[9:0] (SystemExitLatency): Software should set this to the same value programmed by the host through the Set SEL device request, in microseconds. The Set SEL control transfer has six bytes of data and contains four values; Refer to Section 9.4.12 of the *USB 3.0 Specification*. Offset: Name (Meaning) • 0: U1SEL (Time in µs for U1 System Exit Latency) • 1: U1PEL (Time in µs for U1 Device to Host Exit Latency) • 2: U2SEL (Time in µs for U2 System Exit Latency) • 4: U2PEL (Time in µs for U2 Device to Host Exit Latency) If the device is enabled for U1 and U2, then the U2PEL should be programmed. If the device is enabled only for U1, then U1PEL should be programmed into this parameter. If the value is greater than 125 µs, then the software must program a value of zero into this register. coreConsultant configuration provides the default value, assuring the register contains a valid setting when software does not set the register value. The coreConsultant parameter is DWC_USB3_PERIODIC_PARAMS. **NOTE:** Currently, the controller does not use the programmed value.
04h	Set Scratchpad Buffer Array Address Lo This command sets bits [31:0] of the external address of the scratchpad buffer array used for save/restore. If either this command or command 05h is issued while the controller is stopped (RunStop=0), the CmdIOC bit must be set to '0'. The device scratchpad buffer array has the same format as the xHCI scratchpad buffer array; it contains an array of 64-bit pointers to data buffers that will be used to save the controller state.
05h	Set Scratchpad Buffer Array Address Hi This command sets bits [63:32] of the external address of the scratchpad array buffer used for save/restore. If either this command or command 04h is issued while the controller is stopped (RunStop=0), the CmdIOC bit must be set to '0'. The device scratchpad buffer array has the same format as the xHCI scratchpad buffer array; it contains an array of 64-bit pointers to data buffers that will be used to save the controller's state.
07h	Transmit Device Notification

Table continues on the next page...

Table 11-2. device generic command types (continued)

Command	Description
	This command allows any device notification to be transmitted, using the notification type and notification parameters specified in the DGCMDPAR register. DGCMDPAR[3:0] = Notification Type DGCMDPAR[31:4] = Notification parameters, depends on the notification type For example, to transmit a Function Wake, software sets DGCMDPAR[3:0] to 1, and DGCMDPAR[10:4] to the Interface Number. This field relates to the “Notification Type Specific” field in a Device Notification Transaction Packet as described in Section 8.5.6 of the *USB 3.0 Specification*. The following bits of the DGCMDPAR register have been put into the corresponding DWORD described in Section 8.5.6 of the *USB 3.0 Specification*: DGCMDPAR[3:0] into DWORD 1[7:4] (Notification Type) DGCMDPAR[27:4] into DWORD 1[31:8] (Notification Type Specific) DGCMDPAR[31:28] into DWORD 2[3:0] (Notification Type Specific) There is one exception for the Bus Interval Adjustment Device Notification: DGCMDPAR[19:4] represents the Bus Interval Adjustment field; however, in the *USB 3.0 Specification*, the Bus Interval Adjustment field is actually at 31:16 of DWORD 1. In the case of Host Role Request, DGCMDPAR[3:0] = 4, and DGCMDPAR[5:4] = RSP Phase.
09h	Selected FIFO Flush • Parameter[4:0] = FIFO Number • Paramer[5] = ‘1’ for TX FIFO or ‘0’ for RX FIFO
0Ah	All FIFO Flush No Parameter
0Ch	Set Endpoint NRDY Issuing this command will make the controller think that the given endpoint is in an NRDY state. If there are buffers available in that endpoint, the controller will immediately transmit an ERDY. Parameter[4:0]: Physical Endpoint Number
10h	Run SoC Bus LoopBack Test When enabled, executes an SoC Bus Loopback test, which allows the data flow from transmit to receive to be tested without any connection to a PHY. 1. Configure Physical EP0 as a non-stream capable Bulk OUT endpoint (USB Endpoint Number field set to 5'h0) with the desired MaxPacketSize. 2. Configure Physical EP1 as a non-stream capable Bulk IN endpoint (USB Endpoint Number field set to 5'h1) assigned to TxFIFO 0 with the desired MaxPacketSize.

Table 11-2. device generic command types

Command	Description
	3. Issue this command with Parameter[0] set to '1', enabling Loopback mode. 4. Issue Start Transfer to EP0 (Command 6: Start transfer (DEPSTRTXFER)). 5. Issue Start Transfer to EP1. 6. The controller reads data from the IN buffers and writes it back to the OUT buffers. The Tx-FIFO 0 default value of ((512/'DWC_USB3_MBYTES) + 2) should be changed to ((1024/'DWC_USB3_MBYTES) + 2) for loopback mode. Parameter[0]: When 1, enables Loopback mode. When 0, disables Loopback mode.

11.2.2.5 Device programming model

11.2.2.5.1 Register initialization

The USB3 core contains global registers (prefixed by G*) and device registers (prefixed by D*) that are programmed to start operation and handle certain events. This section describes which registers must be accessed depending on the event that software is attempting to handle:

- Power-On or Soft Reset
- USB Reset Event
- Connect Done Event
- SetAddress Device Request
- SetConfiguration Device Request
- Disconnect Event
- Device-Initiated Disconnect and Reconnect
- Device U3 Exit

11.2.2.5.1.1 Device power-on or soft reset

This section explains device core initialization after power-on or soft reset. The application must follow this initialization sequence for device mode operation.

When the core is first powered on, software initializes the following registers. The order of operations is not important, except for the first and last steps (DCTL[CSFTRST]=1 and DCTL[RUN_STOP]=1).

Table 11-3. Power-on or soft reset register initialization

Register	Description
DCTL	Set the CSftRst field to '1' and wait for a read to return '0'. This resets the device core.
GSBUSCFG0/1	Leave the default values, refer Global SoC Bus Configuration Register 0 (GSBUSCFG0) and Global SoC Bus Configuration Register 1 (GSBUSCFG1)
GTXTHRCFG/ GRXTHRCFG	This is required only if threshold is enabled.
GUSB2PHYCFG	Program PHY as per Global USB2 PHY Configuration Register (GUSB2PHYCFG) or leave the default values if the correct power-on values were selected. Note: The PHY must not be enabled for auto-resume in device mode.
GUSB3PIPECTL	Program the following PHY [DatWidth] or leave the default values if the correct power-on values were selected.
GTXFIFOSIZn	Write these registers to allocate prefetch buffers for each Tx endpoint. Unless the packet sizes of the endpoints are application-specific, it is recommended to use the default value.
GRXFIFOSIZ0	Write this register to allocate the receive buffer for all endpoints. Unless the packet sizes of the endpoints are application-specific, it is recommended to use the default value.
GEVNTADR/ GEVNTSIZ/ GEVNTCOUNT	Depending on the number of interrupts allocated, program the Event Buffer Address and Size registers to point to the Event Buffer locations in system memory, the sizes of the buffers, and unmask the interrupt. Note: USB operation stops if the Event Buffer memory is insufficient, because the core stops receiving/ transmitting packets.
GCTL	Program this register to override scaledown, RAM clock select, and clock gating parameters.
DCFG	Program device speed and periodic frame interval
DEVTEN	At a minimum, enable USB Reset, Connection Done, and USB/Link State Change events
DEPCMD0	Issue a DEPSTARTCFG command with DEPEVT.XferRscIdx set to 0 and CmdIOC set to 0 to initialize the transfer resource allocation. Poll CmdAct for completion.
DEPCMD0/ DEPCMD1	Issue a DEPCFG command for physical endpoints 0 & 1 with the following characteristics, and poll CmdAct for completions: USB Endpoint Number = 0 or 1 (for physical endpoint 0 or 1) FIFONum= 0 XferNRdyEn and XferCmplEn = 1 Maximum Packet Size = 512 Burst Size = 0 EPType = 2'b00 (Control)
DEPCMD0/ DEPCMD1	Issue a DEPXFERCFG command for physical endpoints 0 & 1 with DEPCMDPAR0_0/1 set to 1, and poll CmdAct for completions
DEPCMD0	Prepare a buffer for a setup packet, initialize a setup TRB, and issue a DEPSTRTXFER command for physical endpoint 0, pointing to the setup TRB. Poll CmdAct for completion. Note: The core attempts to fetch the setup TRB via the master interface after this command completes.
DALEPENA	Enable physical endpoints 0 & 1 by writing 0x3 to this register
DCTL	Set DCTL[RunStop] to '1' to allow the device to attach to the host. At this point, the device is ready to receive SOF packets, respond to control transfers on control endpoint 0, and generate events.

After the controller has been started, wait for the following events:

- Wait for a DEVT.USBReset event. This indicates that a reset is detected on the USB.
- Wait for a DEVT.ConnectionDone event. This event indicates the end of the reset on the USB. On this event, read the DSTS register to get the connection speed.

11.2.2.5.1.2 Initialization on USB reset

To initialize the core as a device, during USB Reset, the application must perform the following steps:

Table 11-4. Initialization on USB reset

Register	Description
DEPCMD0	If a control transfer is still in progress, complete it and get the core into the "Setup a Control-Setup TRB / Start Transfer" state
DEPCMDn	Issue a DEPENDXFER command for any active transfers (except for the default control endpoint 0)
DEPCMDn	Issue a DEPCSTALL (ClearStall) command for any endpoint in STALL mode prior to the USB Reset (excluding control endpoints)
DCFG	Set DevAddr to '0'

NOTE

- Special Reset Considerations for Default Control Endpoint 0: The default control endpoint is not affected by a USB Reset, so software must continue following the software flow for control transfers explained in "Control Transfer Programming Model" even across a USB Reset event. Resources can only be assigned to the default control endpoint once after a power on or soft reset.

11.2.2.5.1.3 Initialization on connect done

When this event is received, software must perform the following steps:

Table 11-5. Initialization on connect done

Register	Description
DSTS	Read this register to obtain the connection speed
GCTL	Program the RAMClkSel field to select the correct clock for the RAM clock domain. This field is reset to 0 after USB reset, so it must be reprogrammed each time on Connect Done.
DEPCMD0/ DEPCMD1	Issue a DEPCFG command (with Config Action set to "Modify") for physical endpoints 0 & 1 using the same endpoint characteristics from Power-On Reset, but set MaxPacketSize to 512 (SuperSpeed), 64 (High-Speed), 8/16/32/64 (Full-Speed), or 8 (Low-Speed).
GTXFIFOSIZn	(optional) Based on the new MaxPacketSize of IN endpoint 0, software may choose to re-allocate the TXFIFO sizes by writing to these registers.

11.2.2.5.1.4 Initialization on SetAddress request

When the application receives a SetAddress request in a SETUP packet, it performs the following steps:

Table 11-6. Initialization on SetAddress request

Register	Description
DCFG	Program the DCFG register with the device address received as part of the SetAddress request when SETUP packet is decoded.
DEPCMD1	After receiving the XferNotReady(Status) event, acknowledge the status stage by issuing a DEPSTRTXFER command pointing to a Status TRB. This step must be done after the DCFG register is programmed with the new device address.

At this point, the device is ready to receive micro-SOF/ITP and is configured to receive control transfers on control endpoint 0 with a new address assigned.

11.2.2.5.1.5 Initialization on SetConfiguration or SetInterface Request

When the application receives a SetConfiguration or SetInterface request in a SETUP packet, it performs the following steps:

Table 11-7. Initialization on SetConfiguration or SetInterface Request

Register	Description
DALEPENA	Set this register to 0x3 to disable all endpoints other than the default control endpoint 0.
DEPCMDn	Issue a DEPENDXFER command for any active transfers (except for the default control endpoint 0).
DEPCMD1	Issue a DEPCFG command (with Config Action field set to "Modify") for physical endpoint 1 using the current endpoint characteristics to re-initialize the TXFIFO allocation.
DEPCMD0	Issue a DEPSTARTCFG command with DEPCMD0.XferRscIdx set to 2 to re-initialize the transfer resource allocation.
DEPCMDn	Issue a DEPCFG command (with the Config Action field set to "Initialize") for each endpoint that is present in the new configuration (except for the default control endpoint). Note: Control endpoints are bi-directional and must use consecutive even/odd physical endpoint numbers for the OUT/IN direction (such as 2/3, or 4/5), and the FIFONum must be configured to the same value in both directions.
DEPCMDn	Issue a DEPXFERCFG command for each endpoint that is present in the new configuration (except for the default control endpoint 0).
GTXFIFOSIZn	(optional) Based on the new configuration of IN endpoints, software may choose to re-allocate the TXFIFO sizes by writing to these registers.
DALEPENA	Enable the logical endpoints that are active in the new configuration.
DEPCMD1	After receiving the XferNotReady(Status) event, acknowledge the status stage by issuing a DEPSTRTXFER command pointing to a Status TRB. This step must be done after the previous steps to ensure the host does not access any other endpoints that are being set up.

At this point, the core is prepared to accept Start Transfer commands for the newly-configured endpoints. For information on how to set up transfers, see the Operational model.

11.2.2.5.1.6 Alternate initialization on SetInterface request

When handling a SetInterface device request, another possibility is that software may reconfigure existing endpoints instead of starting the configuration from the beginning. This is only possible if no TXFIFOs need to be reassigned. This flow also assumes that the endpoints that are being removed or reconfigured in the new interface have halted their traffic prior to the host issuing the SetInterface request.

Table 11-8. Alternate initialization on SetInterface request

Register	Description
DEPCMDn	Make sure there are no transfers still active on the endpoints that are changing in the new interface. This is normally ensured by the host prior to issuing the SetInterface, but if not, issue an End Transfer command for the transfer on each endpoint that is changing.
DEPCMDn	Issue a DEPCFG command (with the Config Action field set to "Initialize") for each endpoint that is changing in the new configuration. The side effect of the DEPCFG command is that the endpoint's sequence number is automatically set to 0.
DALEPENA	Write this register with all the endpoints that are enabled (including the ones that are not changing).
DEPCMDn	Using the StartXfer command, acknowledge the status stage response for the SetInterface request.

11.2.2.5.1.7 Initialization on disconnect event

When the application receives a Disconnect event, it must set DCTL[ULSTCHNGREQ] to 5. Other than this, the core does not require any initialization. Because the DCTL[RUN_STOP] bit is still '1', the device attempts to reconnect to the host, at which time a USB Reset and Connect Done event occurs. However, if the application does not want to attempt to reconnect to the host, it should perform the steps in the next section.

NOTE

When DCFG[DEVSPD] is programmed for 2.0 only mode (such as, High-Speed or Full-Speed), and if the application wants to issue any commands to clear any pending transfers during a Disconnect interrupt, then it has to disable USB_GUSB2PHYCFG[SUSPENDUSB20] before issuing any commands and re-enable it after the commands have completed.

11.2.2.5.1.8 Device-initiated disconnect

If the application wants to disconnect from the host, it should perform the following actions:

Table 11-9. Initialization on device-initiated disconnect

Register	Description
DEPCMD0	If a control transfer is still in progress, complete it and get the core into the "Setup a Control-Setup TRB / Start Transfer" state
DEPCMDn	Issue a DEPENDXFER command for any active transfers (except for the default control endpoint 0)
DCTL	Set DCTL[RUN_STOP] to '0' to disconnect from the host
DSTS	Poll [DEVCTRLHLT] until it is '1'

At this point, the device is disconnected from the host and will not attempt to reconnect.

11.2.2.5.1.9 Reconnect after device-initiated disconnect

If the application decides to reconnect to the host, it must follow the steps in Device power-on or soft reset.

11.2.2.5.1.10 Initialization after U3 exit

When a device exits from U3 to U0, if there is an active transfer on an endpoint prior to the entry into U3, then software needs to issue a ‘Set Endpoint NRDY’ command. This makes the device endpoint send ERDY.

NOTE

The Set Endpoint NRDY command should be used only for device initialization after U3 exit. It should not to be used in any other scenario.

Typically, the host does not transition the device to U3 during transfers. However, when the following conditions occur, the host may send LGO_U3 request:

- Endpoint enters flow control because transfer is not ready, and
- Host decides to enter a low power state due to user interaction

In such a scenario, the device accepts the LGO_U3 request and enters the U3 state. Because SS is full duplex, the device may have sent ERDY while receiving the LGO_U3 request. As a result,

- On the host side, this ERDY is not processed
- On device side, there is no information stored that this corner case occurred and the host did not process ERDY

Therefore, when coming out of U3, the device needs to send ERDY again so that the host continues the transfer for that endpoint. This is achieved by the device software driver issuing Set Endpoint NRDY command.

When the device software receives a link state change event with the link indicating that the device is entering U3, then the device software tracks and waits for the link state change event with the link indicating U0 entry. When the device software detects this condition and also detects that there are transfers still active on the endpoint, it issues the Set EndPoint NRDY command.

11.2.2.5.2 Operational model

The following sections describes the processes and data structures that the core uses to implement the USB 3.0 specification.

11.2.2.5.2.1 USB and physical endpoints

Endpoints are referred to in two ways:

- As a USB endpoint number: USB endpoints are defined in the USB specification.
- As a physical endpoint resource number: The hardware has a fixed number of physical endpoint resources. Each resource is unidirectional and can be configured to refer to either direction of any USB endpoint.

The software always works on physical endpoints and it knows how physical endpoint corresponds to USB endpoints. DALEPENA is the only register that has one enable bit per USB endpoint.

During SetConfiguration, software maps physical endpoint resources to the required USB endpoints. When a USB request comes in, the USB endpoint number gets converted to a physical endpoint number in the core. Similarly, when a USB packet is sent out, the physical endpoint number is converted to the USB endpoint and sent out.

A USB control endpoint requires two physical endpoints. One physical endpoint is mapped to the OUT direction of the Control endpoint, and the other one is mapped to the IN direction of the Control endpoint. Specifically the two physical endpoints are used as the following:

- The Setup stage of any control transfer uses the OUT direction physical endpoint.

- For a control write or 2-stage transfer, the Data stage (if present) uses the OUT direction physical endpoint and the Status stage uses the IN direction physical endpoint.
- For a control read transfer, the Data stage uses the IN direction physical endpoint and the Status stage uses the OUT direction physical endpoint.

11.2.2.5.2.2 Event buffers

Hardware passes command completion, transfer progress, and asynchronous events to software through one or more Event Buffers.

To configure an Event Buffer, the software performs the following steps:

1. Sets up an empty buffer in system memory.
2. Writes the address of the beginning of the buffer into GEVNTADR. This address must be aligned to the Event Buffer size.
3. Writes the size of the buffer and interrupt mask into GEVNTSIZ. Depending on your system interrupt latency, enough Event Buffer space must be allocated to avoid lost interrupts or reduced performance.
4. Write a 0 into the GEVNTCOUNT register. This must be the last step, as it enables the Event Buffer. After the Event Buffer has been configured, software must not change the size or address.

There is one interrupt line per Event Buffer that indicates there are one or more events present. Software reads one or more events out of the buffer and indicates to hardware how many events it processed by writing the byte count to the GEVNTCOUNT register.

Clock crossing delays may result in the interrupt's continual assertion after software acknowledges the last event. Therefore, when the interrupt line is asserted, software must read the GEVNTCOUNT register and only process events if the GEVNTCOUNT is greater than 0.

The first event produced by the core after the Event Buffer is configured will be written to the address specified in GEVNTADR. Most events are 32 bits, and subsequent events will be written to the address (PreviousEventAddress + 4). When that address exceeds the sum of GEVNTADR and GEVNTSIZ, the core wraps around to the first GEVNTADR value. In this way, the Event Buffer operates like a circular buffer with hardware writing to the "tail" of the buffer and software reading from the "head."

Most events are exactly 4 bytes in size, but there is one exception: The Vendor Device Test LMP Received Event (VndrDevTstRcved) is a 12 byte event that includes a header in the first 4 bytes and the contents of the LMP in the following 8 bytes.

When an event occurs within the core, hardware checks the enable bit that corresponds to the event to decide whether the event will be written to the Event Buffer or not. The Event Buffer contains one of the following types of information, depending on the value of the lower bits of the event:

- Device Endpoint-Specific Event (DEPEVT) (Event[0] = 0x0)
  - The DEPCFG endpoint-specific command specifies the bits that are enabled and which Event Buffer to use for these events.
- Device-Specific Event (DEVT) (Event[0] = 0x1, Event[7:1] = 0x00)
  - The Generic Command Complete event is enabled through the DGCMD[CmdIOC] field when the command is issued.
  - The Event Buffer Overflow Event cannot be disabled and is written to the Event Buffer that encounters the overflow.
  - The rest of the Device-Specific events are enabled through the DEVTEN register.
  - Except for the Event Buffer Overflow Event, these events are written to the Event Buffer specified in the DCFG[IntrNum] field.

The core always leaves one entry free in each Event Buffer. When the Event Buffer is almost full, hardware writes the Event Buffer Overflow event and the USB will eventually get stalled when endpoints start responding NRDY or the link layer will stop returning credits (in SuperSpeed). This event is an indication to software that it is not processing events quickly enough. During this time, events will be queued up internally. When software frees up Event Buffer space, the queued up events will be written out and the USB will return to normal operation.

Event Buffer Content for Device Endpoint-Specific Events (DEPEVT)

Table 11-10. Device Endpoint-n Events: DEPEVT

Field	Description
31–16	Event Parameters (EventParam) For XferNotReady, XferComplete, and Stream events on Bulk Endpoints: [31-16]StreamID. Applies only to bulk endpoints that support streams. This indicates the Stream ID [1] of the transfer for which the event is generated For XferInProgress: [31-16] Isochronous Microframe Number (IsocMicroFrameNum). Indicates the microframe number of the beginning of the interval that generated the XferInProgress event (debug purposes only) For XferNotReady events on Isochronous Endpoints: [31-16] Isochronous Microframe Number (IsocMicroFrameNum). Indicates the microframe number during which the endpoint was not ready **NOTE:** USB 3.0 core represents USB bus time as a 14-bit value on the bus and also in the DSTS register (DSTS[SOFFN]), but as a 16-bit value in the XferNotReady event. Use the 16-bit value to interact with Isochronous endpoints via the StartXfer command. The extra two bits that the USB 3.0 core produces will be necessary for handling wrap-around conditions in the interaction between software and hardware.

Table continues on the next page...

Table 11-10. Device Endpoint-n Events: DEPEVT (continued)

Field	Description
	EPCmdCmplt events For all EPCmdCmplt events [27-24]- Command Type. The command type that completed (Valid only in a DEPEVT event. Undefined when read from the DEPCMD.EventParam field). For EPCmdCmplt event in response to Start Transfer command. [22-16] Transfer Resource Index (XferRscIdx). The internal hardware transfer resource index assigned to this transfer. This index must be used in all Update Transfer and End Transfer commands.
15–12	Event Status (EventStatus) Within an XferNotReady event- [15]: Indicates the reason why the XferNotReady event is generated: 0- XferNotActive- Host initiated a transfer, but the requested transfer is not present in the hardware 1- XferActive- Host initiated a transfer, the transfer is present, but no valid TRBs are available [14]: Not Used [13:12]: For control endpoints, indicates what stage was requested when the transfer was not ready: 2'b01- Control Data Request 2'b10- Control Status Request Within an XferComplete or XferInProgress event- [15]: LST bit of the completed TRB (XferComplete only) [15]: MissedIsoc: Indicates the interval did not complete successfully (XferInProgress only) [14]: IOC bit of the TRB that completed [13]: Indicates the TRB completed with a short packet reception or the last packet of an isochronous interval [12]: Reserved If the host aborts the data stage of a control transfer, software may receive a XferComplete event with the EventStatus field equal to '0'. This is a valid event that must be processed as a part of the Control transfer programming model. Within a Stream Event- [15:12]: - 4'h2- StreamNotFound- This stream event is issued when the stream-capable endpoint performed a search in its transfer resource cache, but could not find an active and ready stream. - 4'h1- StreamFound- This stream event is issued when the stream-capable endpoint found an active and ready stream in its transfer resource cache, and initiated traffic for that stream to the host. The ID of the selected Stream is in the EventParam field. In response to a Start Transfer command- 4'h2- Indicates expiry of the bus time reflected in the Start Transfer command. 4'h1- Indicates there is no transfer resource available on the endpoint. In response to a Set Transfer Resource (DEPXFERCFG) command- 4'h1- Indicates an error has occurred because software is requesting more transfer resources to be assigned than have been configured in the hardware. In response to a End Transfer command-

Table continues on the next page...

Table 11-10. Device Endpoint-n Events: DEPEVT (continued)

Field	Description
	4'h1- Indicates an invalid transfer resource was specified.
11–10	This field is reserved.
9–6	4'h7- Endpoint Command Complete (EPCmdCmplt) Indicates software may issue another Device Endpoint command to the endpoint. When issued in response to an End Transfer command, indicates that DMA stopped for the endpoint. For all other commands, this event does not imply that all effects of the commandtook place. The DEPCMD register contains the same status information present in the Event Status Bits field. 4'h6- Stream Event (StreamEvt) Indicates that a stream-capable endpoint initiated a search within its transfer resource cache. The result of the search is in the EventStatus field (Found or NotFound). 4'h5- Reserved 4'h4 - Reserved 4'h3- XferNotReady Event (XferNotReady) Indicates receipt of a transaction when no TRBs are available for the endpoint. For isochronous IN endpoints, a zero-length packet is automatically sent by hardware and NRDY for non-isochronous endpoints. The application must enable this event if it plans to issue Start Transfer on demand. This event can happen when software issues a Start Transfer or Update Transfer. In this case, software must ignore this event because it has already issued the Start Transfer or Update Transfer. XferNotReady is generated when the core responds NRDY to the host on the USB. It is useful in the beginning of a transfer when software wants to wait for the host to start polling the endpoint before setting up TRBs, but it is not efficient to use this event to determine when the core has run out of TRBs. For determining when the core has processed TRBs and needs software to setup more, use the IOC field in a TRB along with the XferInProgress event that is generated when the core completes a TRB with IOC=1. For non-stream-capable endpoints, the hardware filters multiple events if the host continues transactions, even after the core responds with NRDY. This internal filter is reset after software issues any endpoint command to this endpoint. For stream-capable endpoints, this event is generated each time the host attempts a transaction, even if the core responds with NRDY. For isochronous endpoints, this event is generated only once prior to the Start Transfer command to communicate the current bus time. For additional information, see the Event Status field. 4'h2- XferInProgress Event (XferInProgress) Applies to IN and OUT endpoints. Indicates an EP/Stream specific event happened and it is continuing the transfer. For additional information, see the Device programming model. 4'h1- XferComplete Event (XferComplete) Applies to IN and OUT endpoints. Indicates that a EP/Stream transfer completed and the core stopped the transfer. For additional information, see the Device programming model. 4'h0- Reserved
5–1	Physical Endpoint Number (0-31)

Table continues on the next page...

Table 11-10. Device Endpoint-n Events: DEPEVT (continued)

Field	Description
0	1'b0- Indicates that this is an endpoint-specific event.

1. Note that term Stream ID used in USB chapter is different from Arm SMMU terminology.

Event Buffer Content for Device-Specific Events (DEVT)

Table 11-11. Device-specific events: DEVT

Field	Description
31-25	Reserved
24-16	Event Information Bits (EvtInfo) For a USB/Link State change Event, this field indicates the state of the link- EvtInfo[8:5] - HIRD value received from the LPM token (valid for a Hibernation Request Event) EvtInfo[4] - SuperSpeed event. Set to 1 for SS; Set to 0 for non-SS. EvtInfo[3:0] - Link State. Indicates link state at the time of the event. Follows same encoding as DSTS link state bits.
15-12	Reserved
11-8	11- Event Buffer Overflow Event (EvntOverflow) The core writes this event to indicate there is no space in the event buffer, and one or more Device-specific events may have been dropped after this event. Endpoint-Specific events will not be dropped, but they will be delayed which can cause a drop in performance on the USB. Software uses this event as a warning that the Event Buffer is too small and the core should be reconfigured with a larger Event Buffer or software needs to process events more quickly. In order to avoid repeated Event Buffer Overflow Events, software must free up space in the Event Buffer by acknowledging more than 1 event (writing a value greater than 4 to the GEVNTCOUNT register). This event cannot be disabled, and is always written to the Event Buffer that encounters the overflow. 10- Generic Command Complete Event (CmdCmplt) The core writes this event to indicate the generic command is complete. 9- Erratic Error Event (ErrticErr) The core writes this event to report erratic errors (phy_rxvalid_i/phy_rxvldh_i or phy_rxactive_i is asserted for at least 2 ms, due to PHY error) seen on the UTMI+. Due to erratic errors, the USB 3.0 core goes into Suspended state and a USB/Link state change event (ULStChng) is generated to the application. If the early suspend is asserted due to an erratic error, the application can only perform a soft disconnect recover. For SuperSpeed operation, the PHY erratic error event indicates that the PIPE is not responding to the PHY command (for example, pipe3_PowerDown change or receiver detection). After the core requested any PHY command, if it did not receive pipe3_PhyStatus within 100ms, the core will timeout and generate the erratic error event. After generating the erratic error, the core assumes that the PHY command is completed successfully and proceeds with the next pending PHY command. Software must reset the controller on receiving the erratic error. 8- Reserved 7- Start of (micro)Frame (Sof) 6- USB Suspend Entry Event

Table continues on the next page...

Table 11-11. Device-specific events: DEVT (continued)

Field	Description
	This event provides a notification that the link has gone to a suspend state (L2, U3, or L1). This event is generated when hibernation mode is disabled. The existing Link State Change event (3) provides the same information, but is generated for every link state change. 5- Hibernation Request Event This event provides a notification that the link has gone to a suspend state where hibernation is supported (L2, U3, or L1) and requires software intervention - When DCTL.KeepConnect=1 and L1HibernationEn=0, this event is generated only when the link goes to L2 or U3 - When DCTL.KeepConnect=1, L1HibernationEn=1, and the core is configured to support LPM (DCTL.LPMCap=1 and DCTL.HIRD_Thres[4]=1), this event is generated when the link goes to L2, U3, or L1 with an HIRD value greater than the DCTL.HIRD_Thres[3-0] value The existing Link State Change event (3) provides the same information, but is generated for every link state change. If both events are enabled, only the Hibernation Request event will be generated if the conditions are met. When this event occurs, software is required to start the hibernation process EvtInfo[4:0] indicates the Link State and Speed which caused the Hibernation Event (using the same encoding as the Link State Change event. Suspend is 3, and L1 is 2)EvtInfo[8:5] provides the HIRD value from the host's LPM tokenFor L2 and U3, this event is only generated when DCTL.KeepConnect=1For L1, this event is only generated when DCTL.L1HibernationEn=1 and DCTL.KeepConnect=1In addition to the suspend states that trigger this event, hibernation is also supported when the link is disconnected. See 'Device-Initiated Disconnect and Low Power While Disconnected' for details. 4- Resume/Remote Wakeup Detected Event (WkUpEvt) This event is generated when the host initiates a resume condition on the USB bus. This event is not generated by the device initiating a remote wakeup to the host. 3- USB/Link State Change (ULStChng) The core writes this event to indicate a change of USB or Link state. Bits 23-16 (EvtInfo) of this event indicate the Link Status. The event is generated in SuperSpeed when the link LTSSM state changes, except when LTSSM exits from HOT_RESET, POLL or LTSSM enters into RECOVERY. 2- Connection Done (ConnectDone) 1- USB Reset (USBRst) 0- Disconnect Detected Event (DisconnEvt) Note: This event is generated only if VBUS is off.
7-1	7’h00 - Device Specific Event
0	1’b1 - Non-Endpoint-Specific Event

11.2.2.5.2.3 Transfer and buffer rules

Buffers that are used to transfer data to and from an endpoint are defined using a Buffer Descriptor, which consists of one or more Transfer Request Blocks (TRBs). A CHN flag in the TRB is used to identify the TRBs that comprise a Buffer Descriptor. Therefore, a

Buffer Descriptor refers to a consecutive set of TRB data structures where the CHN flag is set in all TRBs, except the last. Note that a Buffer Descriptor may consist of a single TRB, whose CHN flag will not be set.

Software communicates the location of the first TRB by using the Start Transfer command on an endpoint. Even endpoints that are not stream-capable use this command.The core fetches TRBs from external memory starting at the address provided in the Start Transfer command parameters, continuing in a linear fashion and following the Link TRB until a TRB is encountered that has its LST bit set to '1' or HWO bit set to '0'. Therefore, software must ensure that it has valid TRBs prepared before issuing the Start Transfer command to prevent the core from reading uninitialized memory.

Descriptor fetch requests are buffered within the hardware and handled separately from the progress of the current transfer. Therefore, it is possible that the core completes a transfer with XferComplete but still continues reading TRBs that it has not cached yet. If software is immediately de-allocating the memory for TRBs based on the XferComplete event, it is recommended that software issue an End Transfer command for the endpoint/ transfer resource prior to de-allocating the memory. The completion of the End Transfer command flushes out any pipelined descriptor fetches and avoids a potential bus error.

While processing TRBs, two conditions may cause the core to write out an event and raise an interrupt line:

- TRB Complete:
  - For OUT endpoints, a packet is received which reduces the remaining byte count in the TRB buffer to zero.
  - For IN endpoints, an acknowledgement is received for a transmitted packet which reduces the remaining byte count in the TRB buffer to zero.
- Short Packet Received:
  - For OUT endpoints only. While writing to a TRB buffer, the endpoint receives a packet that is smaller than the endpoint's MaxPacketSize.

This table shows the USB 3.0 Core Actions Based on TRB Control Bits.

Table 11-12. USB 3.0 Core Actions Based on TRB Control Bits

Direction	TRB Complete Packet	ISP	IOC	CHN	LST	CSP	Action
IN	Yes	-	X	0	1	-	XferComplete event
IN	Yes	-	0	X	0	-	No event
IN	Yes	-	1	X	0	-	XferInProgress event
OUT	Yes No	-	X	0	1	-	XferComplete event
OUT	Yes No	-	0	X	0	-	No event

Table continues on the next page...

Table 11-12. USB 3.0 Core Actions Based on TRB Control Bits (continued)

OUT	Yes No	-	1	X	0	-	XferInProgress event
OUT	X Yes	X	X	X	X	0	XferComplete event [1]
OUT	X Yes	X	X	0	1	1	XferComplete event
OUT	X Yes	X	X	1	0	1	Search for CHN=0, accumulate IOC and ISP, then follow CHN=0 rules [2]
OUT	X Yes	X	1	0	0	1	XferInProgress event
OUT	X Yes	0	0	0	0	1	No event
OUT	X Yes	1	0	0	0	1	XferInProgress event

1. When a TRB receives whose CSP bit is 0 and CHN bit is 1 receives a short packet, the chained TRBs that follow it are not written back (for example, the BUFSIZ and HWO fields remain the same as the software-prepared value)
2. In the case of an OUT endpoint, if the CHN bit is set (and CSP is also set), and a short packet is received, the core retires the TRB in progress and skip past the TRB where CHN=0, accumulating the ISP and IOC bits from each TRB. If ISP or IOC is set in any TRB, the core generates an XferInProgress event. Hardware does not set the HWO bit to 0 in skipped TRBs. If the endpoint type is isochronous, the CHN=0 TRB will also be retired and its buffer size field updated with the total number of bytes remaining in the BD.

On XferComplete and XferInProgress events, status bits in the event indicate LST, IOC, Short Packet Received, or Bus Error status. In the "fast-forward" case, the IOC status is accumulated from the skipped TRBs, not just the TRB that received the short packet.

When the hardware writes back the TRBs, it updates the BUFSIZ field to represent the remaining unused buffer.

11.2.2.5.2.3.1 Number of TRBs rule

- Software must set up only one TRB for a control setup or status stage.
- If software is preparing multiple transfers for an IN endpoint, it may be necessary to place a 0-length TRB between transfers that are MaxPacketSize aligned to indicate transfer boundaries to the host. This is not necessary, if the class driver and the host can handle bursting between transfers.
- If software is preparing multiple transfers for an OUT endpoint, it needs to place a MaxPacketSize TRB between transfers, if it expects the host to transmit a 0-length packet between transfers.

11.2.2.5.2.3.2 TRB control bit rules

Transfer control bits must conform to the following restrictions:

For OUT endpoints, the CSP bit must be the same in every TRB within a Buffer Descriptor (either set or clear).

- The core autonomously checks the HWO field of a TRB to determine if the entire TRB is valid.
- Therefore, software must ensure that the rest of the TRB is valid before setting the HWO field to '1'. In most systems, this means that software must update the fourth

DWORD of a TRB last. Every time software validates a TRB by setting HWO=1, it must also issue an Update Transfer command to the core.

- Software sets the HWO bit to 1 when it creates the TRB, and cannot modify it until hardware resets it to 0. However,
  - Software must detect when a "fast-forward" occurs on an OUT endpoint that receives a short packet, since some TRBs in a chain may still have their HWO bit set to 1 while belonging to software.
  - Hardware will not clear the HWO bit of a Link TRB. Therefore, software can only modify a Link TRB if the TRB prior to the Link TRB has its HWO bit set to 0.
- The LST bit must not be set to 1 for isochronous endpoints.
- For a Setup or Status TRB, set CHN=0, LST=1, and CSP=0.
- For the data stage of a control transfer, set CSP=0 in all TRBs and LST=1 in the last TRB of the data stage.
- For Link TRBs, the LST, CHN, IOC, ISP, CSP, and Stream ID fields are ignored and must be set to 0.
- The Link TRB's chain bit is implicitly equal to the chain bit of the TRB before it. If the TRB before it has CHN=1, then the Link acts as if it's CHN=1. If the TRB before it has CHN=0, then the Link acts as if it's CHN=0. A Link TRB cannot be the last TRB in a Buffer Descriptor.
- When CSP=1 and the core receives a short packet, it searches for the end of the Buffer Descriptor by finding the Normal TRB that has CHN=0. Therefore, it is illegal to setup a circular buffer with all Normal TRBs with CHN=1.

11.2.2.5.2.3.3 Buffer size rules and zero-length packets

The hardware contains a cache that holds four TRBs per transfer.

For IN endpoints, the following rules apply:

- The number of chained TRBs necessary to construct a single packet must never exceed 3 . A maximum of one link TRB can be present in the chain.
- If software wants to indicate a transfer completion to the host by sending a zero-length packet after a multiple of MAXPACKETSIZE, it must set up a zero-length TRB following the last TRB in the transfer.

For OUT endpoints, the following rules apply:

- If the first TRB has CHN=1, its buffer size must be ≥ 1 byte
- The total size of a buffer descriptor must be a multiple of MAXPACKETSIZE
- A received zero-length packet still requires a MAXPACKETSIZE buffer. Therefore, if the expected amount of data to be received is a multiple of MAXPACKETSIZE,

software should add MAXPACKETSIZE bytes to the buffer to sink a possible zero-length packet at the end of the transfer.

For IN and OUT endpoints, the following rule applies:

- The BUFSIZ field in a link TRB must be set to 0.

11.2.2.5.2.3.4 Transfer setup recommendations

Software can either set up transfers before the host attempts to move data on an endpoint ("preset" transfers) or can set up transfers on demand ("on-demand" transfers). When using preset transfers, software can safely disable the XferNotReady event in the endpoint configuration. However, when using on-demand transfers, the XferNotReady event must be enabled and software may not use the "No Response" variant of the Update Transfer command. The XferNotReady event is issued when the host attempts to move data on an endpoint when one of the following conditions is present:

- No previous transfer was started with Start Transfer.
- Not enough hardware-owned (HWO=1) TRBs are available to handle the requested data movement. The XferNotReady event must not be disabled for control endpoints because the event is an integral part ofcontrol transfer handling.

Although there are many valid ways to set up transfers, it is recommended that you choose one of three general mechanisms:

- When software wants to set up one transfer at a time and has the entire buffer available for transfer, it must set up TRBs that point to the data buffers and in the last TRB it must set the LST bit and issue Start Transfer, which points to the first TRB location.
  - When the USB transfer completes, the core will notify the software through the XferComplete event. The LST bit will be set in the status field of the event. The XferComplete will also release the Transfer Resource.
  - A premature XferComplete event can happen before all the data buffers are exhausted, if there is a Bus Error or, in an OUT transfer a short packet has been received and CSP=0. During these conditions the TRB will have an updated BUFSIZ field which represents the amount of buffer remaining after the successful part of data transfer.
  - Software can also set up an IOC bit TRB, so that it gets notified when data buffers are used up and can free them up sooner than waiting for the entire transfer to complete. On completing a TRB with IOC set, and not the last one, the core will issue an XferInProgress event with IOC set in the status field of the event.
- When software wants to set up one transfer at a time, but it has fewer data buffers available than the full transfer size, it must set up circular TRBs (using a Link TRB),

and also set up IOC bits in the TRBs. Depending on buffer allocation and interrupt frequency it can set IOC for once in "x" number of TRBs.
    - As soon as the USB transfer for a TRB is completed, and if IOC is set, the core will generate an XferInProgress event with IOC set in the status field.
    - On seeing the event, software reuses the TRB and updates it with the next data buffer. It also issues an Update Transfer command, indicating it has updated a TRB.
    - If software is slow and hardware finds a TRB with HWO reset to 0, it waits for an Update Transfer command. Upon seeing the Update Transfer, it prefetches the TRB and continues the transfer.
    - When software reaches the end of the transfer, it sets the TRB LST bit. When hardware completes the TRB, it issues an XferComplete event and releases the Transfer Resource.
- When the Device software has multiple transfers to set up, it must set up circular TRBs and also set up CHN bits in all TRBs, except the last of each transfer. For OUT endpoints, multiple transfers can be supported when the CSP field is set to '0' if each transfer has a multiple of MaxPacketSize bytes, since a short packet will end the transfer. If multiple transfers may contain short packets, the CSP field must be set to '1' to enable the next transfer to continue even if a short packet is received. Depending on buffer allocation and interrupt frequency, it can either set the IOC for once in "x" number of TRBs or in the last TRB of each transfer.
    - For OUT endpoints, the transfer can finish prematurely due to a short packet from the host. In this case, the core processes the remaining TRBs, skipping the updates to these TRBs until it reaches TRB of the next transfer. While skipping these TRBs, if any of the TRB it encounters the interrupt setting of ISP or IOC, it will generate the corresponding event once and ignore the interrupt settings of the remaining TRBs until a TRB of next transfer is reached.
    - Device software can reclaim these skipped TRBs even though the HWO still indicates 1 (hardware-owned).
    - For IN endpoints, software cannot stop providing transfers while it is ending transfers with CHN=0 and LST=0, otherwise it is possible that the endpoint is left in a flow-controlled state on the USB.

11.2.2.5.2.4 Transfer resource usage and transfer state

Software allocates one Transfer Resource for an endpoint during initialization. When software issues a Start Transfer command, this Transfer Resource is used. When an XferComplete event happens, the Transfer Resource is released back to software. Similarly, when End Transfer completes, the Transfer Resource is released.

Following is a typical transfer usage summary (not complete usage model):

- Software sets up data buffer(s) and TRB(s) in external memory with the TRB(s) pointing to the buffer(s).
- Software issues Start Transfer with the pointer to the first TRB in the command parameters.
- If software sets up all the buffers and TRBs at the same time (Host and Device Software negotiate the transfer size), then just one Start Transfer is enough, and software waits for a XferComplete event.
- If software did not set up all the TRBs, then every time software adds a new TRB (by setting HWO=1) it must issue an Update Transfer command. The hardware will fetch the TRB again and continue the transfer.
- The End Transfer command is used only during error conditions and not used during normal transfers. For example, if software has set up multiple transfers and if a USB Reset event happens, it must remove all the transfers from the queue using End Transfer with the ForceRM bit set 1.
- A transfer is completed when all the data has been transferred or a short packet has been received. If software is queuing in only single transfers at a time (normal method for mass storage - for ethernet over USB, software can set up multiple transfers using the Chain bit and Continue on Short Packet bit), once XferComplete interrupt has happened, the transfer is completed. When starting a new transfer, software now needs to issue the Start Transfer command.
- An OUT transfer's transfer size (Total TRB buffer allocation) must be a multiple of MaxPacketSize even if software is expecting a fixed non-multiple of MaxPacketSize transfer from the Host.

The following is the usage flow when the device does not know the transfer size and the host is not expected to send short packets to end the transfer. Note that this is not a normal scenario, but some device applications require this flow.

1. The device sets up the TRB(s) with the maximum transfer size expected, and enables all of them (HWO=1).
2. The host sends the OUT transactions, and completes the transfer without sending any short or zero-length packet.
3. The device controller keeps updating the TRBs as required but does not give XferComplete event.
4. The device software implements a timer to count inactivity on the OUT endpoint to detect the end of transfer.
5. When the timer expires, software issues End Transfer command with ForceRM=1, and waits for the command to complete.
6. Software reads the RAM0 cache (stream scratchpad bytecount location) through the slave interface to know the remaining byte count and calculates the received byte count.g.Software calculates the byte address using the following formula:

dut_bytecount_addr = dut_ram0_base + `DWC_USB3_ADDR_CACHE_STREAMS + (stream_id *(32 + (`DWC_USB3_CACHE_TRBS_PER_TRANSFER * 16))) + (7*4);

where, stream_id is the Transfer Resource Index (XferRscIdx) of the OUT endpoint, and

dut_ram0_base = `DUT_REGISTER_BASE_ADDRESS+32'h4_0000; // RAM0 starting address

7. Software sets up the new transfer.

The following figure displays the Transfer Usage State Machine.

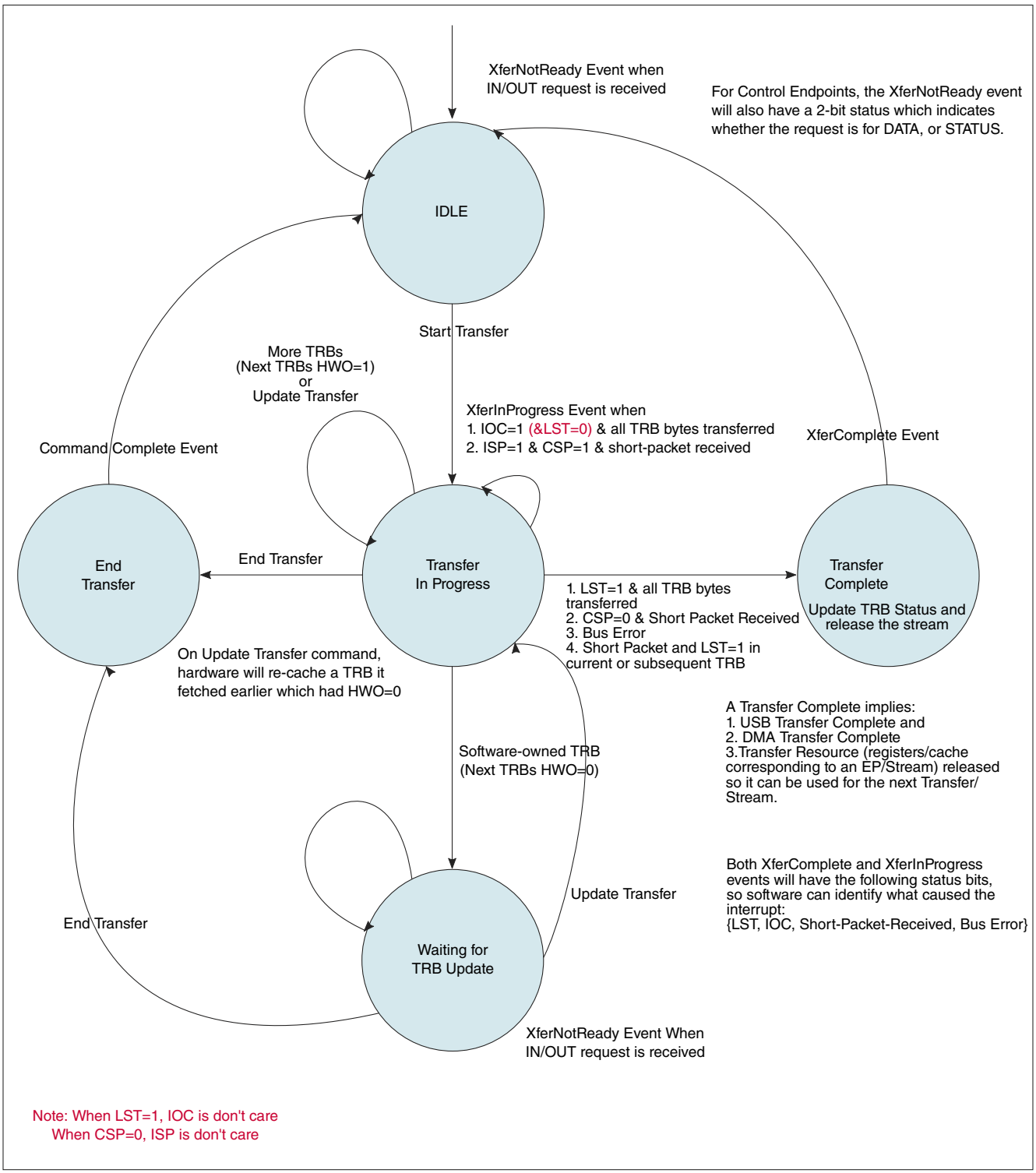


Figure 11-18. Transfer usage state machine

11.2.2.5.2.5 Transfer descriptions

The following sections describe typical transfer processes.

11.2.2.5.2.5.1 Non-isochronous OUT transfers

The sequence below describes a regular, non-isochronous OUT transfer (control-data/ bulk/interrupt).

Table 11-13. Non-isochronous OUT transfer sequence

Steps	Host	Device Core	Device Software
Software sets up data buffers before a host request			
Step 1	-	-	Software sets up Normal TRBs and enables DMA by issuing Start Transfer.
Step 2	-	Device fetches the TRB and caches it.	-
Step 3	Host sends data packets	-	-
Step 4	-	Core receives the packets in RXFIFO and stores them in to TRB data buffers through DMAs. While the RXFIFO has space and the BUS DMA keeps up with the host data transfers, the device receives all the packets. Once all the data is received, the core updates the TRB status and then generates a XferComplete event. In SS mode, if GRXTHRCFG.USBRxPktCntSel is set to, • 0: the NumP sent depends on the DCFG.NumP and bMaxBurstSize • 1: the NumP sent depends on the RXFIFO space available when the packet is received When DMA is slow or CRC fails: • If the RXFIFO overruns or CRC error is detected, device times out in Non-SS mode or asks for the same data packet (through ACK/SeqNumber/Retry Bit) in SS mode. • In HS mode, on NAK the host will issue PING and device will ACK it (if there is enough space in the RXFIFO) and the host resumes with data again. In SS mode, the host will send the requested data. If the device receives any packet with other than the requested data sequence, it will ignore it.	-
Step 5	-	-	Software processes the XferComplete event.
Software sets up data buffers after the host request.			
Step 6	Host sends data packets.	-	-
Step 7	-	Since DMA is not set up, the device will send NAK in Non-SS mode and will send NRDY in SS mode to the host. It also generates an XferNotReady event.	-
Step 8	-	-	On seeing the XferNotReady event, Software sets up TRBs and enables DMA by issuing Start Transfer.
Step 9	-	Steps 2-5 are followed.	-

Table continues on the next page...

Table 11-13. Non-isochronous OUT transfer sequence (continued)

Steps	Host	Device Core	Device Software
Step 10	-	If the core encounters a short packet before the transfer size data programmed in the TRB is received, it skips the remaining TRBs of the transfer. If the core encounters control bits LST, IOC, or ISP it will generate events and, in the case of an LST TRB, the stream will be completed. Software has to reclaim any TRBs skipped with HWO=1.	On detecting a skipped TRB condition, software reclaims the TRBs with HWO=1.

11.2.2.5.2.5.2 Isochronous OUT transfers

This section describes the difference between isochronous and non-isochronous OUT transfers referring to the sequence in Non-isochronous OUT transfers

- In Step 2, if the TRB is not fetched in time, the core discards the packet. In addition subsequent packets in the service interval also will be dropped.
- In Step 4, if DMA is enabled and if RXFIFO overrun happens or CRC fails, the core discards the packet. In addition subsequent packets in the service interval also will be dropped.
- In Step 7, if DMA is not enabled, the core drops the data and generates XferNotReady event.

In general, the TRB chain of a service interval will be released if the core receives DATA0 PID in HS USB 2.0 mode or receives a packet with LPF set in SuperSpeed mode.

Special scenarios are:

- If the core misses this last packet indication, it releases TRB at the service interval boundary.
- If the core encounters a short packet before the transfer size data programmed in TRB is received, it skips the remaining TRBs of the transfer.
- If the core encounters control bits 'LST' or 'IOC' or "ISP" it will generate event and, in the case of LST TRB, the stream will be completed, but software has to reclaim TRBs if any TRBs skipped with HWO=1.

11.2.2.5.2.5.3 Non-isochronous IN transfers

The sequence below describes a regular, non-isochronous IN data transfer (control-data/ bulk/interrupt).

Table 11-14. Non-isochrnous IN transfers

Steps	Host	Device Core	Device Software

Table continues on the next page...

Table 11-14. Non-isochrnous IN transfers (continued)

	Software sets up data buffers before a host request		
Step 1	-	-	Software sets up TRBs and enables DMA by issuing Start Transfer.
Step 2	-	Device fetches the TRB, caches it, and prefetches the data into TXFIFO. Only in SS mode, the core sends ERDY (Async ERDY, even before host requesting data provided endpoint is in flow controlled state. LSP flow is same regardless of if the host asked for it.)	-
Step 3	Non-SS host issues a token request, SS host issues ACK TP to request data.	-	-
Step 4	-	If DMA fetches keep up with host requests, the device sends all the data, updates the TRB status, and then generates an XferComplete event when ACKs for all the packets are received. When DMA is slow: • If the TXFIFO becomes empty on a host data request, the device sends a NAK in Non-SS mode. In SS mode, the device sends NRDY, and when data is available in TXFIFO it sends ERDY and waits for the host request again. • If the TXFIFO becomes empty in the middle of a packet transfer (happens only in Threshold mode), in Non-SS mode the device will invert the CRC. In SS mode, it terminates with EDB after appending the corrupted CRC. The core waits for the host to retry the packet before fetching the data again. If the host retries a packet: • In Non-SS mode, a transmitted packet is kept in the TXFIFO until the ACK arrives (this is immediate in Non-SS mode). If the host retries a packet, it is immediately sent out from the TXFIFO. • In SS mode, before ACK can be received, the core could have sent additional packets in a burst. When a retry happens, NRDY is sent to the host and the TXFIFO is flushed. A DMA command is issued on the master bus to prefetch data from where the host is asking. Once data is fetched into the TXFIFO, the device sends ERDY and waits for host request again. • In Threshold mode, when the TX threshold amount of data is fetched, the device sends ERDY to the host.	-
Step 5	-	-	Software processes XferComplete event.
	Software sets up data buffers after a host request.		

Table continues on the next page...

Table 11-14. Non-isochrnous IN transfers (continued)

Step 6	Non-SS host issues a token request. SS host issues ACK TP to request data.	-	-
Step 7	-	Since DMA is not set up, the device sends s NAK in Non- SS mode, or NRDY in SS mode, to the host. The core also generates an XferNotReady event	-
Step 8	-	-	On seeing the XferNotReady event, software sets up Normal TRBs and enables DMA by issuing Start Transfer.
Step 9	-	Steps 2-5 are followed.	-
Step 10	-	When software knows it is talking to a host that always ends transfers with short packets, and therefore always requires a zero-byte packet to end a transfer, it includes a zero-byte TRB (CHN=0) after the last TRB which is an even multiple of MaxPacketSize. When software does not know if the host is going to do another IN token after receiving the exact number of bytes it expected (and it was a multiple of MPS), it does not include a zero-byte TRB, and sets LST in the last TRB. If the host returns with another IN for a zero-byte packet, it uses the XferNotReady/Start Transfer mechanism to add one zero-byte TRB. This is called "On- Demand."	-

11.2.2.5.2.5.4 Isochronous IN transfers

This section describes the difference between isochronous and non-isochronous IN transfers referring to the sequence in Non-isochronous IN transfers.

- In Step 4, if the TXFIFO is empty when host IN request arrives, the core returns a zero length packet in both Non-SS and SS modes.
- In Step 7, the device returns zero length packet and generates an XferNotReady event.

Special scenarios:

- When data is fetched and there is no request from the host (possibly due to a missing request or corrupted request) the data will be flushed at the end of the service interval (corresponding Micro- Frame) boundary and the core moves onto fetching next service interval data. If there are some TRBs that are not serviced in the service interval, the core will skip these TRBs.
- If the core encounters control bits LST or IOC, it will generate an event, and in the case of LST TRB, the stream will be completed, but has to reclaim TRBs if any TRBs skipped with HWO=1.

11.2.2.5.2.6 Handling ENDPOINT_HALT

On receiving a SetFeature (ENDPOINT_HALT) control transfer, software issues a Set Stall command on the endpoint. Because of the delay from when software issues the command and when it is recognized by the core, other transfer events (XferInProgress, XferComplete) may be received prior to the endpoint being halted. For a description of the Set Stall endpoint command and its effects, see Commands 4 and 5: Set Stall and Clear Stall (DEPSSTALL, DEPCSTALL).

On ClearFeature (ENDPOINT_HALT), software must first remove all pending transfers for the endpoint through the End Transfer command. It may then issue a Clear Stall command on the endpoint followed by Start Transfer to start transfers again. For a description of the Clear Stall endpoint command and its effects, see Commands 4 and 5: Set Stall and Clear Stall (DEPSSTALL, DEPCSTALL).

11.2.2.5.2.7 Handling L1 event during a transfer

This section discusses the scenario in which the software already issued the Start Transfer command but the LPM occurred before or during the data transfer.

When an IN transfer is in progress, (that is, started and not completed yet), and the software sees a link state change event to L1, then the software can use the GDBGFIFOSPACE register (write followed by a read) to determine if the TXFIFO is empty or not. If the TXFIFO is not empty, it can initiate a remote wakeup.

NOTE

- The GDBGFIFOSPACE register is a debug register which gives the 'Space Available" for the endpoint selected with a write prior to the read. If this space available is less than the particular TXFIFO's maximum size, then the TXFIFO is not empty.
- Software can enable the Link State change event for USB 2.0 as there are not a lot of events compared to SS.

11.2.2.5.3 Isochronous transfer programming model

Isochronous endpoints get guaranteed service by the host during a "Service Interval". The service interval for an endpoint is communicated to the host via the endpoint descriptor, and it is configured in the hardware through the DEPCFG command. However, unexpected behavior on the bus may prevent the host from servicing the endpoint as frequently as expected.

Although the endpoint receives "guaranteed" service during the interval, there are no bus-level acknowledgments, which means that there is no guarantee that the host (IN endpoints) or the device (OUT endpoints) receives correct data.

Therefore, the isochronous programming model differs from the bulk programming model to accommodate these two factors:

- No bus-level acknowledgement of packet transmission or reception
- No guarantee that all the data setup for an interval has been transmitted or received

For isochronous endpoints, software has the following responsibilities:

- Set up enough TRBs to keep up with the rate of data transmission or reception
- For FIFO-based IN endpoints, guarantee the validity of the TX data at least 1 micro Frame before the beginning of the interval that the data will be transmitted

Hardware has the following responsibilities:

- Transmit or receive data at the appropriate bus time
- For FIFO-based IN endpoints, delay fetching until 1 micro Frame before the beginning of the interval that the data will be transmitted
- Maintain a 1-to-1 correspondence between Buffer Descriptors and the intervals on the bus
- Report missed isoc intervals and update the buffer sizes in TRBs to reflect the amount of data moved

11.2.2.5.3.1 Definitions

- bInterval: The field in an endpoint descriptor used to communicate the service interval of the device. Its value ranges from 1 to 16, however, the USB 3.0 core supports a range of 1 to 14.
- Service interval: An integral number of microframes (2^[bInterval-1]) during which the host will poll the device to move data.
- Beginning of interval: The interval begins when the least significant (bInterval-1) bits of the bus time are all 0's.
- End of interval: The interval ends when the least significant (bInterval-1) bits of the bus time are all 1's.
- Data payload: The number of bytes to be moved during the service interval. It will be transmitted using MaxPacketSize packets.
- Buffer descriptor: A set of one or more TRBs with CHN=1 and the last one having CHN=0. The group of TRBs represents the data buffers for one service interval.
- Last TRB of a Buffer Descriptor: The TRB within a Buffer Descriptor that has CHN=0. When an interval has completed, this TRB will contain the total remaining buffer size of the Buffer Descriptor andus also the completion status of the interval.

- Microframe (μF): A unit of time on the USB that lasts 125 μs.
- Start of Frame (SOF): The beginning of a microframe.
- FIFO-Based Isoc IN: A sub-type of Isoc IN endpoints that has the characteristic of always assuming the HWO bit in a TRB is '1' and delays fetching of transmit data.
- Prefetch delta: For FIFO-based IN endpoints, TX data may be prefetched as early as 1 μF before the beginning of a Service Interval, but never earlier.
- Retire a TRB: When HW sets the HWO bit to 0 in a TRB and writes it back.
- Retire a Buffer Descriptor: When HW retires at least the first and last TRB of a Buffer Descriptor.
  - Other TRBs of the Buffer Descriptor may not be retired if an interval was missed (IN or OUT) or if a short packet/last packet is received (OUT). In this case, software may reclaim those TRBs even though HWO=1.
- Missed interval: When the device does not move all the data in an interval. This may occur when the host does not poll for all the data, or because of application-side delays that prevent all the data from being moved.
  - For IN endpoints, this occurs when the host does not request all the data prepared in the Buffer Descriptor, or not all the data is fetched from the system bus in time.
  - For OUT endpoints, this occurs when the host does not send a packet, a packet is dropped due to CRC error, or the system bus is too busy to accept the data.

11.2.2.5.3.2 Endpoint configuration

An isochronous endpoint is setup in much the same way as a bulk endpoint, with the following exceptions:

- The bInterval_m1 value in DEPCFG Parameter 1 must be set to the value reported in the endpoint descriptor minus 1. When the core is operating in Full Speed, bInterval_m1 must be set to 0.
- The MaxPacketSize value in DEPCFG Parameter 0 must be set to the same value reported in the endpoint descriptor.
- The "FIFO-based" bit in DEPCFG Parameter 1, bit 31, must be set for FIFO-based isochronous endpoints.
- For the best performance, set the TXFIFOs corresponding to Isochronous IN endpoints to a high priority.

After the endpoint is configured, it can be enabled by setting the appropriate bit in the DALEPENA register.

11.2.2.5.3.3 Transfer configuration

Software describes isochronous transfers through a series of Buffer Descriptors. Each Buffer Descriptor corresponds directly to one service interval of data. The fields within an isochronous TRB are the same as bulk TRBs with the following exceptions:

- The SOF field in an IN endpoint TRB is a don't care. For an OUT endpoint, the core will write this field with the timestamp of the last packet received into the TRB's buffer. This information is not needed for normal operation, only for debug purposes.
- For High-Speed, High-Bandwidth IN endpoints, a maximum of 3 packets can be sent during an interval. The PktCntM1 field ([25:24] of the 3rd DWORD) must be set to the (number of packets in the Buffer Descriptor - 1). For example, if 3 packets are to be transmitted during the interval, this field must be set to 2. For Super-Speed and OUT endpoints, this field is a don't care.
- The first TRB in a Buffer Descriptor must have the TRBCTL field set to the "Isochronous-First" type while all others have this field set to "Isochronous".
- The ISP bit is renamed IMI (Interrupt on Missed Interval) and should be set if software wants to receive an XferInProgress event when at least 1 packet is missed within an interval.
- The IMI bit should be set to the same value in all TRBs of a Buffer Descriptor.
- The CSP bit must be set to 1 (short packets cause the hardware to move to the next Buffer Descriptor, they do not end an isochronous transfer).
- The LST bit should be set to 0 (isochronous transfers normally continue until the endpoint is removed entirely, at which time an End Transfer command is used to stop the transfer).

All other bits and fields (IOC, HWO, CHN, BPTR, BUFSIZ) retain the same behavior as they have for bulk endpoints. If software needs to receive an interrupt after every service interval, it should set the IOC bit to '1' in each TRB. However, if software wants to reduce the interrupt frequency and the application can tolerate some latency, the IOC bit can be set to '0' and the core will not generate a XferInProgress event when the TRB is completed.

For OUT endpoints, the Buffer Descriptor must describe a total buffer size of:

```
(Mult+1)*MaxBurst*MaxPacketSize
```

11.2.2.5.3.4 Starting a transfer

The hardware will report the bus time that the host starts polling the endpoint inside the XferNotReady event. Software will use this value as a reference when issuing the Start Transfer command to serve as time synchronization with the host.

1. Set up TRBs and data buffer for the endpoint.

2. After configuring and enabling the endpoint, wait for an XferNotReady event.

   The XferNotReady event will contain the time that the host started polling the endpoint in the upper 16 bits.

3. Issue the Start Transfer command to the endpoint with a future microframe time written into the upper 16 bits of the DEPCMD register. The future microframe time must be a value that is an integral multiple of intervals after the time reported in the XferNotReady event and aligned to the beginning of an interval. For example, if bInterval is 3 (4 microframes), and the XferNotReady time is 2, the value can be 4, 8, 12, and so on. The future microframe time must also be no greater than 4 seconds past the time reported in the XferNotReady event.
4. If the future microframe time has already passed when the command is received, the core will respond with an error (bit 13 in the Command Complete event).

In this case, software must issue End Transfer, then wait for another XferNotReady event and attempt the command again with a time that is further in the future.Otherwise, the first Buffer Descriptor will be used for the interval starting with the microframe time specified in the Start Transfer command.

11.2.2.5.3.5 Core behavior during an interval

After a transfer has been started, the hardware will perform the following functions for IN endpoints:

1. Fetch TX data as early as one interval prior to the beginning of the interval (call it A) if the HWO bit is set to one in the TRB.
2. Decrement the buffer size of each TRB as packets are transmitted.
3. Retire TRBs when their buffer size has reached 0, issuing an XferInProgress event if the IOC bit is set.
4. If the next interval (B) starts before all the packets have been transmitted for interval A:
   a. Flush the TXFIFO.
   b. Retire the Buffer Descriptor of interval A with a "Missed Isochronous" status.
   c. Retire the Buffer Descriptor of interval B with a "Missed Isochronous" status.
   d. See Checking interval status for a description of how software can determine that an interval ended unexpectedly.
   e. Go to step 1 to prepare for interval C
5. Otherwise, if all the TRBs of interval A are completed, the hardware will prepare for interval B.
6. If the host completes an interval by polling for all the data, but then it polls the endpoint again during the same interval, the hardware will respond with a zero-length packet and no interrupt will be made to software.

7. If the host polls the endpoint prior to the time specified in the Start Transfer command, the hardware will respond with a zero-length packet and no interrupt will be made to software.
8. If the host polls the endpoint during the expected interval and the hardware has no data prepared, the core will respond with a zero-length packet, but no XferNotReady event will be generated because the transfer is active. When the next interval starts, the XferInProgress event is generated (based on ISP and IMI) with the "Missed Isochronous" status, which is the way software finds out that this occurred.

For OUT endpoints, the hardware performs the following functions:

1. If a packet is received for the correct interval represented by the Buffer Descriptor (call it A).
   a. Write the packet into the buffer.
   b. Decrement the buffer size of the TRB.
   c. Write the timestamp of the received packet into the TRB.
   d. Retire a TRB when its buffer size reaches 0, issuing an XferInProgress event if the IOC bit is set.
2. If a short packet or packet with the last packet flag (lpf) is received for the correct interval.
   a. Write the packet into the buffer.
   b. Retire the Buffer Descriptor of interval A with the TRBSTS set to 0.
   c. If the IOC bit is set in the last TRB of the Buffer Descriptor, issue an XferInProgress event with bit [13] set.
   d. Go to step 1 to prepare for interval B.
3. If a packet is received for the correct interval but it has an error (CRC error, RXFIFO overflow), the core will not increment its expected sequence number value which causes future packets within the same interval to be dropped.
4. If a packet is received at a bus time prior to the time specified in the Start Transfer command, it will be dropped.
5. If a packet is received for the correct interval (A), but not enough buffer space is available for the core to write the packet (due to the HWO bit still '0' in one or more of the TRBs of the Buffer Descriptor), no XferNotReady event will be generated. The core will behave as follows:
   a. Wait until software sets the HWO bit to '1' and issues an Update Transfer command
   b. Retire the Buffer Descriptor of interval A with a "Missed Isochronous" status
   c. Go to step 1 to prepare for interval B.
6. If a packet is received for the next interval (B) before all the packets have been received for interval A:
   a. Retire the Buffer Descriptor of interval A with a "Missed Isochronous" status.
   b. Retire the Buffer Descriptor of interval B with a "Missed Isochronous" status.

c. See section Checking interval status for a description of how software can determine that an interval ended unexpectedly.
d. Go to step 1 to prepare for interval C.

Special Considerations for Isochronous OUT Endpoints

For OUT isochronous endpoints, the hardware detects a missed interval when the host sends a data packet in a future interval. It does not detect the missed interval at the exact interval boundary. Therefore, if the host abruptly stops sending isochronous OUT packets, there will be no interrupt or event indicating this to software. Software should use one of the following mechanisms to detect the interruption of isochronous OUT traffic:

1. The ultimate consumer of the isochronous OUT data will detect an underflow.
2. The device driver can use a timer detect that no XferInProgress events have been received for multiple intervals.

If this occurs, software should issue an End Transfer command to the endpoint and wait for a XferNotReady event which signals that the host is ready to resume isochronous traffic.

11.2.2.5.3.6 Checking interval status

As packets are transmitted or received during an interval, the buffer size of each TRB will be decremented. When the host is operating normally and polling for all the interval data, the buffer size of all the TRBs of an IN endpoint Buffer Descriptor will be zero after the interval has completed. For OUT endpoints, if the host sends less data than the Buffer Descriptor was setup for, the remaining buffer size may be greater than 0.

When the interval completes, the final remaining buffer size and missed isoc status is written to the last TRB of the Buffer Descriptor. If MissedIsoc is set, then it means the BUFSIZ is not accurate and may indicate that more data was transmitted or received than in reality. If the MissedIsoc is not set, it means the BUFSIZ field is correct.

When hardware detects the end of an interval (including normal and abnormal ends), it performs the following:

- If it has not already been retired, the first TRB of the Buffer Descriptor will be retired.
- Any non-first TRBs with CHN=1 that had not already been retired will not be written back. HWO will still be 1, and software can reclaim them for another transfer.
- The last TRB of the Buffer Descriptor will be retired with:
  - HWO = 0.

    - BUFSIZ = The total remaining buffer size of the Buffer Descriptor.
    - TRBSTS = "Missed Isoc" if any packets were missed, zero otherwise.
- If the IOC bit is set in the last TRB, or the IMI bit is set and packets were missed, hardware will issue an XferInProgress event

Software can get notification of an interval completing by setting the IOC bit in the last TRB of the Buffer Descriptor. The IOC bit may also be set in any other TRB of the Buffer Descriptor if software wants earlier notification that a TRB has completed. Software can also get only a notification of an interval completing unexpectedly by setting the IMI (Interrupt on Missed Isoc) bit in the last TRB of the Buffer Descriptor. When an interval completes unexpectedly and either the IOC or IMI bit is set in the TRB, the MissedIsoc bit (15) of the XferInProgress event will be set. The following table shows which events are generated in each scenario:

Table 11-15. TRB event generation events

Scenario	IOC	IMI	Action
TRB completed with a MaxPacketSize packet	0	0	No interrupt
	0	1	No interrupt
	1	0	XferInProgress (IOC=1)
	1	1	XrefInProgress (IOC=1)
TRB completed with a short packet	0	0	No interrupt
	0	1	No interrupt
	1	0	XferInProgress (Short=1, IOC=1)
	1	1	XrefInProgress (Short=1, IOC=1)
Interval completed due to missed packet (missed isoc)	0	0	No interrupt
	0	1	XferInProgress (MissedIsoc=1, IOC=1)
	1	0	XferInProgress (MissedIsoc=1, IOC=1)
	1	1	XrefInProgress (MissedIsoc=1, IOC=1)

It is normal to lose two intervals at a time when an error occurs during one interval. One interval is lost because of the host (which may not be polling enough) and the second interval is dropped because the device needs time to synchronize up to the next (third) interval.

11.2.2.5.3.7 Adding intervals to a transfer

Because isochronous endpoints represent a stream of data, the TRBs of an isochronous endpoint will normally be setup in a circular list, such as:

- TRB 1 (CHN=0, IOC=1)

- TRB 2 (CHN=0, IOC=1)
- Link (to TRB 1)

Example: Assume TRB 1 represents the data for the first interval and TRB 2 represents the data for the second interval. Because IOC=1, when the core completes the first interval, it will issue an XferInProgress event which indicates to software that TRB 1 can be analyzed to retrieve the results from interval 1. TRB 1 can the be re-used to represent the data for the third interval by setting HWO=1 and issuing an Update Transfer command.

Each time software sets up TRBs for a new interval, it must follow the guidelines in the Transfer Configuration section. Software will set up TRBs for future intervals when it obtains those buffers from another software layer or when TRBs are retired by the core.

11.2.2.5.3.8 Moderating events

During a transfer, software may enable or disable any endpoint-specific event by re-issuing the DEPCFG command with the "Config Action" set to "Modify" and a modified DEPEVTEN field. All other field. All other fields in all other parameters must remain the same as the initial endpoint configuration.

11.2.2.5.3.9 Other types of isochronous endpoints

The above description of the isochronous endpoint programming model assumes that software is setting up buffers in external memory that apply to certain intervals of data. This is called a "buffer-based" isochronous endpoint.

However, there is another mode that the hardware supports to accommodate other mechanisms of sourcing or sinking isochronous data, called "FIFO-based" and the types are defined by bits [31:30] of DEPCFG Parameter 1. This table illustrates the differences between the models:

Table 11-16. Isochronous endpoints models

Type	FIFO-Based [31]	Bulk-based [30]	TRB Fetch	TRB Write back	TX DMA	Retire Buffer Descriptor at passed interval
Buffer-based	0	0	When HWO=1	Yes	When HWO=1, no earlier than one interval ahead of time.	Yes
FIFO-based	1	0	When internal cache space available	No	No earlier than 1 µF before the interval	Yes
Invalid	1	1				

11.2.2.5.3.9.1 FIFO-based isochronous IN endpoints

The software of some applications is unable to keep up with the short latency and high bandwidth of isochronous traffic and require the data to be sourced and done a sync through external FIFOs. One example is that there are two external TXFIFOs, one for each interval, and an address-to-FIFO-pop logic is implemented so that when the core attempts to read from interval 1's address, the translation logic converts this into a pop signal for interval 1's TXFIFO. When the core attempts to read from interval 2's address, the translation logic converts this into a pop signal for interval 2's TXFIFO.

To describe a FIFO-based implementation, software sets up an endpoint with the "FIFO-based" configuration bit set. This type of endpoint ignores the HWO bit in all TRBs, assuming that the TRB is always valid, and that the core should never write it back. Software chooses the buffer pointers within the TRBs to correspond to the address needed by the translation logic to specify which FIFO should be popped. By also using the CHN bit, headers and payload can be concatenated from different FIFOs. For example, if there are 4 external FIFOs: HA, PA, HB, PB, where HA/HB contain the 4 byte header for 2 intervals and PA/PB contain the 512 byte payload for 2 intervals, this can be described by using the following 5 TRBs:

1. BUFPTR=HA, IOC=0, CHN=1, BUFSIZ=4
2. BUFPTR=PA, IOC=1, CHN=0, BUFSIZ=512
3. BUFPTR=HB, IOC=0, CHN=1, BUFSIZ=4
4. BUFPTR=PB, IOC=1, CHN=0, BUFSIZ=512
5. Link to (1)

Every interval, the core will be creating a 516 byte packet that consists of 4 bytes from the header FIFO and 512 bytes from the payload FIFO. However, if the host does not poll for the packet, the core will skip 1 or 2 intervals of popping, depending on whether it has already started reading the next interval. External logic (or software) is responsible for flushing and refilling alternate FIFOs so that the core is always reading the correct data for the next interval. In normal intervals, the FIFOs will be empty, but when an interval is missed, there may still be data present in the FIFOs.

Software Requirements:

- The "FIFO-based" bit must be set in the DEPCFG when configuring the endpoint.
- No field within any TRB may be changed after the Start Transfer command is issued.
- Data must be valid in the external FIFO at least 1 uF before the beginning of the interval for which the data is intended.

Core Behavior:

- The earliest the core will read from the FIFO will be 1 µF before the beginning of the interval for which the data is intended.
- The core will not write back the TRB after it has completed it.
- The core will re-read the TRB from external memory even though it is not allowed to change.
- The only indication of missed intervals is the XferInProgress event if IOC or IMI is set in the TRBs. No indication will be made of how much data was actually transmitted.

11.2.2.5.3.9.2 FIFO-based isochronous OUT endpoints

The FIFO-based model is also supported for OUT endpoints. In this case, core writes to specific addresses will be translated into external FIFO pushes. The same example from above (4 byte headers and 512 byte payloads) can be applied to OUT endpoints where the headers are split from the payload as the core receives 516 byte packets. The same software requirements and core behavior apply (for example, no writebacks and no indication of how much data was received). External logic assumes that if less than the expected amount of data is pushed into an interval's FIFO, the interval is invalid.

11.2.2.5.3.10 End a transfer

Since isochronous endpoints have no bus-level acknowledgements, it is not necessary to set LST=1 in a TRB to gracefully end an isochronous transfer. Software can issue an End Transfer command to end an isochronous transfer. The core will wait until it can complete operations for the endpoint before returning the Command Complete event in response.

11.2.2.5.4 Control transfer programming model

Control transfers follow a flow of setup/data/status. On the USB bus, there are many error scenarios that can disrupt this flow. The hardware has a special mechanism that automatically recovers from these scenarios as long as software follows this single control transfer programming model.

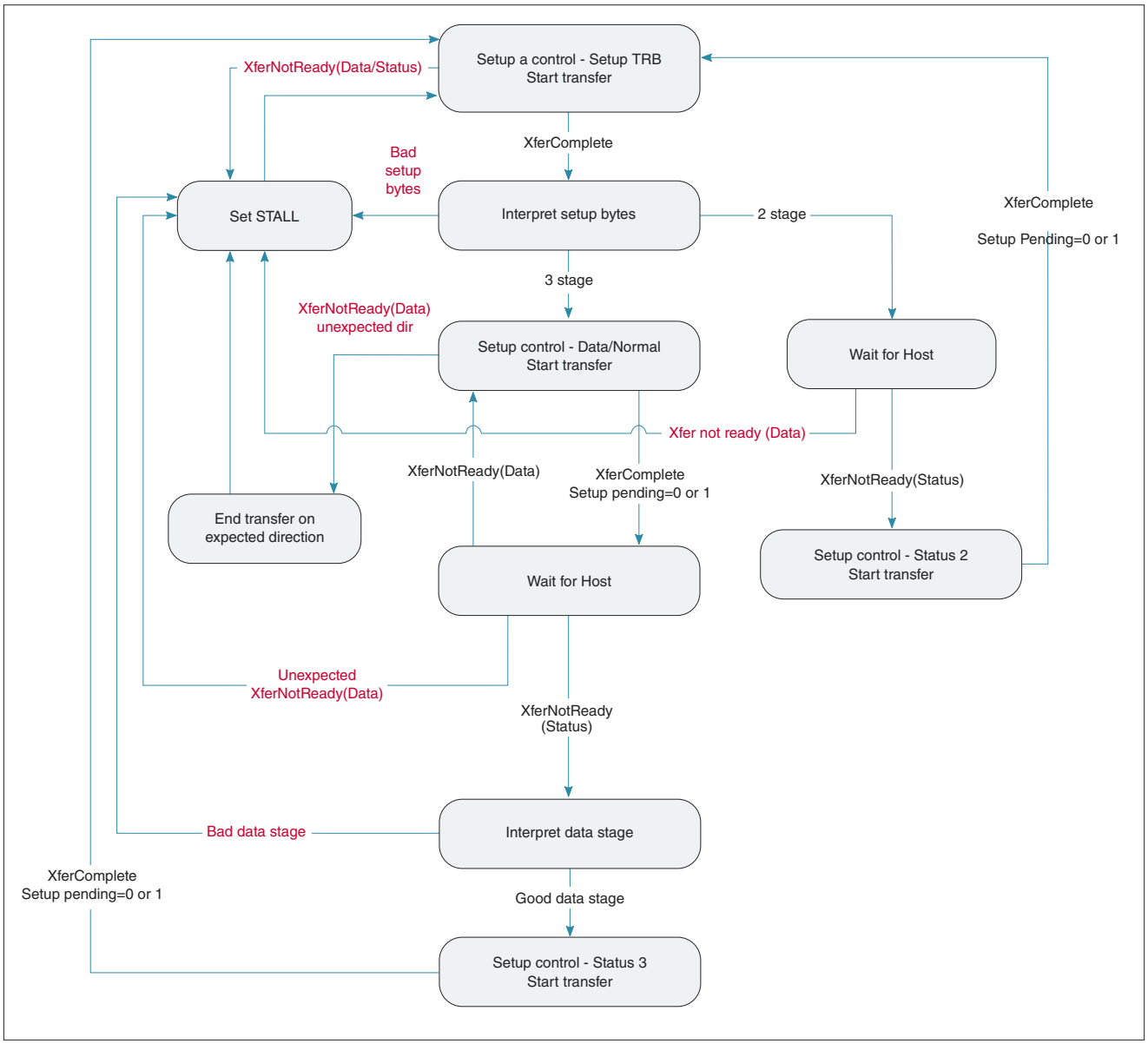


Figure 11-19. Software flow for control transfers

If hardware does not receive a setup packet, it discards the unexpected data and status stages. It does not issue XferNotReady to the application. The hardware will also not generate a XferNotReady event if it receives a setup packet before a TRB is prepared for it. As control endpoints are bi-directional, the IN and OUT directions of the endpoint are used for the different stages as follows:

Control read:

- EP0 - Setup OUT
- EP1 - Data IN
- EP0 - Status OUT/Status TP

Control write or 2-stage transfer:

- EP0 - Setup OUT
- EP0 - Data OUT (if present)
- EP1 - Status IN/Status TP

Set STALL is always issued on EP0, and the XferComplete/XferNotReady events are generated on the correct direction. For other control endpoints, substitute EP0 with the OUT direction (such as, 2, 4, 6) and EP1 with the IN direction (such as, 3, 5, 7). The same programming model is used for USB 2.0 and USB 3.0 control transfers.

11.2.2.5.4.1 Two-stage control transfer programming model

1. Software sets up a Setup TRB and issues Start Transfer on EP0 pointing to the Setup TRB.
2. After XferComplete is received, software interprets the setup bytes. If they are bad, issue Set Stall on EP0 and go back to the step 1. If a XferNotReady (Data/Status) event is received before the XferComplete event for the Setup stage, issue Set Stall. This is an error case where the host is attempting to move data or start the status stage for a previous control transfer that has already completed.
3. Wait for an XferNotReady event for the Status stage which will occur on EP1. If an XferNotReady event for the Data stage is received (either direction), issue Set Stall on EP0 and go back to the step 1. This is an error case where the host is attempting to start the data stage when the setup bytes did not indicate a data stage was present.
4. After XferNotReady (Status) is received, start the status stage by issuing Start Transfer on EP1 pointing to a valid Status-2 TRB.
5. After XferComplete is received on EP1, go back to the step 1.

Although not required, software may look at the Status TRB and examine the Setup Pending bit in the TRB status field. If it is set, it means that this control transfer was aborted on the USB bus and that the host did not receive the Status stage ACK.

11.2.2.5.4.2 Three-stage control transfer programming model

1. Software sets up a Setup TRB and issues Start Transfer on EP0 pointing to the Setup TRB.
2. After XferComplete is received, software interprets the setup bytes. If they are bad, issue Set Stall on EP0 and go back to the step 1. If a XferNotReady (Data/Status) event is received before the XferComplete event for the Setup stage, issue Set Stall. This is an error case where the host is attempting to move data or start the status stage for a previous control transfer that has already completed.
3. Start the data stage by issuing Start Transfer on EP0 (control write) or EP1 (control read) pointing to a valid Control-Data TRB.

   a. If an XferNotReady (Data) event is received for the incorrect direction, software must issue an End Transfer for the data stage it has already started, then issue Set Stall. This is an error case where the host is attempting to move data in the wrong direction.
   b. If a XferNotReady (Data) event is received for the correct direction, ignore the event and continue to wait for a XferComplete event.
4. After the XferComplete event is received, software waits for an XferNotReady event. Although not required, software may look at the completed Data TRB(s) and examine the Setup Pending bit in the TRB status field and the BUFSIZ. If the Setup Pending bit is set or the BUFSIZ is non-zero, it means that this control transfer was aborted on the USB bus and that the host did not complete the data stage.
5. If a XferNotReady (Data) event is received after the XferComplete for the Data stage, it could mean one of two things:
   a. This host is trying to complete the data stage by moving a 0-length packet. This can occur if the data stage was an exact multiple of max packet size. If this is the case, software sets up an extra TRB (with a BUFSIZ of max packet size for control writes, or 0 for control reads), issues Start Transfer, and goes back to the step 4.
   b. This host is trying to move more data than specified in the wLength field of the setup bytes. In this case, software issues Set Stall on EP0 and goes back to the step 1.
6. When the XferNotReady (Status) event is received, software interprets the data stage (for example, number of bytes transferred, content of the control write buffer) and decides if the data stage was successful. If not, it issues Set Stall on EP0 and goes back to the step 1.
7. Start the status stage by issuing Start Transfer on EP1 (control write) or EP0 (control read), pointing to a valid Status-3 TRB.
8. After XferComplete is received, go back to the step 1.

Although not required, software may look at the Status TRB and examine the Setup Pending bit in the TRB status field. If it is set, it means that this control transfer was aborted on the USB bus and that the host did not actually receive the Status stage ACK. When device software sets STALL with the Set Stall command, it is possible that this command can be delayed on the chip bus due to latency or the chip bus servicing other agents. In this case, software can receive an XferNotReady event if the core detects no TRB active before it detected the Set Stall. Software must service these events knowing that it stalled the control command.

11.2.2.5.4.3 Handling fewer requests than wLength

In a control IN transfer, if the host asks for less data than wLength, the core skips the TRBs of the data stage and generates an XferComplete event after encountering the TRB that has LST=1. Hardware sets HWO to 0 in those TRBs that it skipped and in the last TRB. In a control OUT transfer, if the host sends less data than wLength, the core treats this like a short packet received into a TRB with CSP=0. It immediately generates a XferComplete event, not setting the HWO bit in the remaining TRBs to 0. Software needs to reclaim those TRBs that still have HWO=1.

11.2.2.5.4.4 Control OUT transfer examples

The sequence in the following table describes two and three stage control OUT transfers.

Table 11-17. Control OUT transfer

Steps	Host	Device core	Device software
	Setup stage		
1	-	-	Software sets up the Setup TRB and enables DMA by issuing Start Transfer to EP0 (endpoint 0, OUT direction).
2	-	Device fetches the TRB and caches it.	-
3	Host sends SETUP packet.	-	-
4	-	SETUP packet is received in the RXFIFO. There are 24 bytes allocated for each control endpoint, which guarantees reception of the three back-to-back SETUP packets. An ACK handshake is returned in non-SS mode. An ACK TP with NumP=1 is returned in SS mode. After SETUP, bytes are transferred through the DMA, and the core issues an XferComplete event.	-
5	-	If the host starts a Data/Status stage, the core returns NAK in Non-SS mode or NRDY TP in SS mode. If the SETUP request is invalid, core returns STALL (when software sets STALL) in non-SS mode. In SS mode the core returns ERDY, and when the Data/Status stage is received it returns STALL.	Software processes the XferComplete event and decodes the SETUP bytes. If a Data stage is required, it will set up one Control-Data TRB and possibly subsequent Normal TRBs, then issue Start Transfer to EP0 (endpoint 0, OUT direction). If a Data stage is not needed, it will skip to the Status stage. If the SETUP request is invalid, software instead sets the STALL bit by issuing Set Stall command for EP0 (OUT direction).
	Data stage		
6	-	Device fetches the TRB and caches it. In SS mode only, the core will send ERDY.	-

Table continues on the next page...

Table 11-17. Control OUT transfer (continued)

Steps	Host	Device core	Device software
7	Host sends data packets.	-	-
8	-	The NumP for control transfer will be 1. In addition, if another SETUP is received before the data transfer happens on USB, the core will skip the data transfer that software sets up without waiting for it to happen on the USB bus, and send XferComplete (when LST=1 TRB encountered) with the SetupPndg status bit set. Software has to reclaim the TRBs with HWO=1 in the skipped TRBs.	-
9	-	If the host starts a Status stage, the core returns NAK in Non-SS mode or NRDY TP in SS mode.	On seeing the event: • Software decodes the control write data. If the Data stage is OK, then it will wait for XferNotReady with "Control Status Request" set, set up a Control-Status-3 TRB (0-bytes), and enable DMA by issuing Start Transfer to EP1 (endpoint 0, IN direction). • If the Data stage is not OK, then software sets STALL by issuing Set Stall command to EP0.
	Status stage		
10	-	Device fetches the TRB and caches it. In SS mode only, the core will send ERDY	Software sets up a Status stage TRB only after the Data stage of a three-stage control transfer is completed and an XferNotReady event occurred.
11	Host requests status packets.	-	-
12	-	In Non-SS mode, the device will send a zero-length packet if STALL is not set; otherwise, it will send STALL. In SS mode, if STALL is not set, then it sends ACK TP with NUMP=1 else it sends STALL. If another SETUP is received before the status transfer on USB, the core will send XferComplete with the SetupPndg status bit set. The STALL bit will be cleared by the core whenever it receives a SETUP packet. SETUPs are always accepted.	-
13	-	-	On seeing the event: • If there are no other errors, then the transfer has completed successfully.

11.2.2.5.4.5 Control IN transfer examples

The sequence in the following table describes two and three stage control IN transfers.

Table 11-18. Control IN transfer

Steps	Host	Device Core	Device Software
	Setup stage		
1-5	Steps 1-5 are same as Control OUT transfer examples, except that software sets up Tx buffers and Start Transfer is issued to EP1 (endpoint 0, IN direction).		
	Data stage		
6	-	Device fetches the TRB and caches it. It prefetches the data and puts it in the TXFIFO. The core sends ERDY only in SS mode.	-
7	Host requests data packets	-	-
8	-	This step is similar to the step 4 of Non-isochronous IN transfers. The NumP for control transfer is 1. In addition, if another SETUP is received before the data transfer happens on USB, the core will skip the data transfer software has set up without waiting for it to happen on the USB bus and send XferComplete (when LST=1 TRB encountered) with the SetupPndg status bit set. Software has to reclaim the TRBs with HWO=1 in the skipped TRBs and flush the TXFIFO.	-
9	-	If the host starts the Status stage, the core returns NAK in non-SS mode or NRDY TP in SS mode.	On seeing the event: • If the Data stage is OK, then it sets up Control-Status-3 TRB (zero-bytes) and enables DMA by issuing Start Transfer to EP0 (endpoint 0, OUT direction). • If the Data stage is not OK, then software will set a STALL by issuing Set Stall command to EP0 (OUT direction).
	Status stage		
	Same as in Control OUT transfer examples.		

11.2.2.5.5 Stream handling in superspeed

The SuperSpeed stream protocol consists of a negotiation between the host and device concerning the selection of a stream, followed by data movement on the selected stream. The core automatically handles the stream protocol by initiating stream selection and also taking into account streams initiated by the host.

NOTE

This section is not applicable for USB 2.0 mode.

11.2.2.5.5.1 Stream IDs and transfer resources

A stream ID is a 16-bit value which represents an individual flow of data between host and device. For stream-capable endpoints, the stream ID must be non-zero and is used in:

- The Command Parameter field in the DEPCMD register used when issuing a Start Transfer command.
- The Stream ID field in a TRB.
- The EventParam field in an endpoint-specific event (DEPEVT): XferComplete, XferInProgress, XferNotReady, and StreamEvt.

It is necessary to allocate 1 Transfer Resource per endpoint. The cost of each transfer resource is (32 + N_TRBS_PER_XFER x 16) bytes of memory in the descriptor cache. Core can cache maximum four TRBs per transfer.

After the Start Transfer command completes, the hardware returns the 7-bit Transfer Resource Index in the command complete event (and). This index is used only for the Update Transfer and End Transfer commands. In all other places, the Stream ID is used.

11.2.2.5.5.2 Stream selection and stream programming model

Referring to the USB 3.0 Device Stream Protocol State Machine, the Idle state is the state where a device does not have a current stream to work on. When the hardware detects that it is in the Idle state, it will attempt to initiate and move data on a stream by transmitting ERDY to the host.

In order for this to occur, the endpoint must be "primed" and the stream must be both active and ready.

- **Primed**: After a stream-capable endpoint is configured, it is in the disabled state, which prevents the device from initiating any stream selection. After the host performs a Prime transaction, the endpoint is moved into the "Primed" state where it is eligible to initiate stream selection.
- **Active**: Software makes a stream active by issuing a Start Transfer command for the stream. A stream becomes inactive after an XferComplete event or after software issues an End Transfer command.
- **Ready**: After a stream is added via the Start Transfer command, its default state is "ready." The hardware will label the stream as "not ready" if the host gives a NoStream rejection to the hardware's attempt to initiate the stream. If the host

performs a Prime, all currently active streams are automatically set to the "ready" state by hardware.

In the Idle state, the hardware selects a stream from its cache to initiate. Once the core has chosen an active stream, it attempts to initiate and move data on that stream until the endpoint enters the Idle state, which occurs when the host rejects the stream selection (NoStream) or the stream is terminated (PP=0) by the host or device. Therefore, software is required to prepare enough HWO=1 TRBs for at least one packet prior to issuing the StartXfer command.

In device-oriented stream selection class drivers (such as UASP), the host will not reject the device's stream selection. However, other class drivers may be developed in which the host initiates stream selection. In this situation, software will not start any transfers and will wait for a XferNotReady event. When that event is received, software will look at the StreamID in the event and start the transfer associated with the given Stream ID. If for some reason the class driver allows the host to reject a stream that has already been initiated by the host or device, software will receive a Stream (NotFound) event. If this occurs, software may replace the existing stream by issuing End Transfer for it and then issuing Start Transfer for another stream.

Some host implementation scenarios exist where the host and device become out of sync in the stream protocol, creating a deadlock condition where the device waits for the host to issue a Prime transaction while the host waits for the device to issue an ERDY. To resolve potential deadlock conditions, software should perform the following:

1. Implement a time out for stalled transfers on stream-capable endpoints.
2. When the timeout elapses with no data transfer for any stream after a StreamEvent (NotFound) event, restart one of the streams by issuing an End Transfer command followed by a Start Transfer command.

This places the stream in the Ready state and causes the core to transmit an ERDY to the host.

11.2.2.5.5.3 Data movement within a stream

Data movement within a stream is the same as data movement using the Bulk endpoint type (Start Transfer, Update Transfer, and updating TRBs). The XferInProgress, XferComplete, and XferNotReady events are issued as data is transferred between host and device. The Stream ID field in the events indicate which stream within the endpoint is generating the events. If software receives an XferNotReady event with the EventStatus field indicating XferNotActive, it means that the host is attempting to initiate a stream that has not been added to the hardware's cache through Start Transfer. In response to this event, software should add the requested stream if it is available.

11.2.2.5.6 Multiple device interrupt support

The DWC_USB3_DEVICE_NUM_INT parameter allows you to instantiate up to 32 interrupt lines. Each interrupt line is associated with an event buffer described by the GEVNT*n registers. The interrupt number and the event buffer number are the same. When the endpoint is configured with the DEPCFG command, it is mapped to a specific interrupt such that all endpoint-specific events are written to the event buffer selected.

One time static interrupt mapping during the device endpoint configuration is recommended. In some circumstances, software may want to change the interrupt number of an endpoint between transfers. For example, the Bulk Only Transport (BOT) class driver uses a single endpoint for command (CBW), Data IN/OUT, and status (CSW), and you may be interested in routing CBW and CSW to one interrupt and Data IN/OUT to another interrupt. Software can issue a DEPCFG command with the “Config Action” set to “Modify” to change the IntrNum field without affecting the flow control status of the endpoint. This ability to dynamically map interrupt comes with the following restrictions:

- The XferNotReady event must always be disabled on the endpoint
- The interrupt number can only be changed on Non-Stream Bulk endpoints
- The interrupt number can only be changed between transfers (i.e. after a XferComplete event or before issuing a StartXfer command)
- The EPCmdComplete event for the DEPCFG command changing the interrupt number will come on the new interrupt number
- All other fields of the DEPCFG command must match the original configuration of the endpoint, except the event enable bits

11.2.2.5.7 Worst-case response time

The DWC_usb3 controller, as a high-speed device, needs a minimum of three PHY (UTMI/ULPI) clocks to complete processing of the OUT data packet. Therefore, it cannot accept new token for these three clocks. However, for an isochronous transaction, there is no handshake and the next token could come within three PHY clocks after the previous RxActive de-assertion[1]. If this worst-case condition occurs, then the controller

sends NAK for bulk and interrupt tokens and drops isochronous and SETUP tokens. The host handles this as a timeout condition for SETUP packet and retries the SETUP packet, but the ISOC OUT token is dropped.

11.2.2.5.8 Low power operation

1. This can happen theoretically when operating in UTMI 16-bit mode only with worst-case timing/delays with five hubs/cables. The host provides a minimum of 88 HS bit-times (typically, more than that) of Inter Packet Gap between packets and generally will not result in less than three clocks on the device-end UTMI PHY.

11.2.2.5.8.1 Low power operation of USB

The USB 3.0 protocol allows a SuperSpeed host and device to negotiate with each other and decide when to bring their link into a lower power state (U1 or U2). This is accomplished by each side making individual decisions about whether they have any traffic currently pending or if they expect any traffic soon. The device requests low power entry by transmitting an LGO_U1 (or LGO_U2) Link Command and exits low power by using an LFPS handshake. The USB 3.0 Device Controller uses information from the host (bus side) and from software (application side) to decide when the best time to request low power entry is and when to exit from low power. The same calculation is used when the controller decides whether to accept a low power request from the host. The controller maintains state information about each endpoint and whether it is "active" or "paused". If all enabled endpoints are paused from either the application side or the bus side, low power entry is allowed. If at least one endpoint is not paused, low power entry is not allowed. To indicate that it does not have any pending traffic and allow the link to go into a low power state, the software may pause all enabled endpoints using one of the following mechanisms:

- Allow transfers to complete and not start any new ones
- Not setting the HWO bit in the TRBs to '1'

Exception: Software may prepare a Setup TRB for control endpoints without affecting the ability of the link going into low power. Otherwise, an endpoint is considered active on the application side. For these endpoints, the host indicates that it is paused in one of the following ways:

- Setting the Packet Pending (PP) flag to '0' in the DPH
- Setting the PP flag to '0' and NumP to '0' in the ACK TP
- For interrupt endpoints, by not polling the endpoint for two service intervals
- Note: After an endpoint is configured using the DEPCFG command and before the host sends any packets to it, the endpoint is considered to be paused on the bus side

Exception: If the data packet payload has a CRC error, the core responds with an ACK (Retry=1), and the endpoint is not paused. Otherwise, if PP=1 or Deferred=1, the endpoint is considered to be active on the bus side. Control endpoints have an additional condition as follows:

The core does not allow low power entry during a control transfer (starting from the reception of setup packet and ending at the completion of the status stage).If the core is configured for isochronous support, the following additional rules apply:

- The controller enters U0 at SystemExitLatency microseconds prior to the beginning of every microframe to receive the Isochronous Timestamp Packet (ITP).

- A PING causes every isochronous endpoint to wake up and be considered active, preventing low power entry.
- An isochronous endpoint is paused automatically when it transmits or receives the last packet of its interval (lpf=1) or if two intervals pass and the endpoint has not transmitted or received its last packet. In this way, after the PING, the core does not enter low power until all isochronous endpoints have moved their last packet or two intervals pass without another PING.

If all enabled endpoints are paused by software or the host, the core issues an LGO_U1 (or LGO_U2) link command or it responds with an LAU link command to the low power request of the host or hub. Otherwise, the core rejects the low power requests with an LXU link command.

Additionally,

- For bulk endpoints, the USB 3.0 specification says that the device may go to U1/U2 500ms (tERDYTimeout) after transmitting ERDY. It is the responsibility of the software to detect no activity on a bulk endpoint for 500ms or more and issue End Transfer if there is no activity. When the transfer is removed, the endpoint allows U1/U2 entry again.
- Because U1/U2 entry is based on the presence of packets in the TX FIFO and not based on the TRBs that are available to the core, it is possible that an IN endpoint responds with a NRDY, goes to U1, then U0, then transmits an ERDY if the timing and TXFIFO size works out such that the core cannot keep at least one packet in the TXFIFO at all times.
- In addition, before the final decision made to either initiate or accept low power state, device checks the state of DCTL[INITU2ENA][ACCEPTU2ENA][INITU1ENA] [ACCEPTU1ENA]. If the device receives Set Link Function LMP with Force_LinkPM_Accept bit asserted, the device accepts low power request from its link partner independent of endpoints state and DCTL Power control settings.

11.2.2.5.8.2 Low power operation of core

When the link is in the U1, U2, U3, SS.Inactive, SS.Rxdetect, or SS.Disabled state (for USB 3.0) or the USB is in Sleep, Suspend, or V_{BUS}-off mode (for USB 2.0), the core can be put into a low power mode to save power. The core supports the following low power mode:

- **Clock-gating mode**: In this mode, the clocks connected to most of the core modules are gated off. Modules that still get clocks in the low power mode detect wakeup conditions and remove clock gating to other modules when a wakeup condition is detected. Clock-Gating mode can be used when the USB is in any low power state. Clock-Gating mode is enabled by setting the Disable Clock Gating bit in GCTL to 0.

11.2.2.5.8.2.1 Clock-gating mode

Clock-Gating mode is enabled if the Disable Clock Gating bit in the GCTL is set to 0. When the USB/link is in a low power state, the clocks to most of the core modules are gated off automatically by the core. When a wakeup condition is detected by the core, or when the software schedules new work to the core (in U1 or U2 states) or changes the USB/Link state, the core turns on clocks and the entire core goes into active mode.

11.2.2.6 Host programming model

Please refer xHCI specification for detailed information.

11.2.2.6.1 Initializing host registers

To initialize the controller as host, the application must perform the steps described in the xHCI Specification. If intermediate firmware needs to override a coreConsultant parameter or reset value, it can write to the controller-specific global (G*) registers at power-on.

Table 11-19. Overriding global registers in host configuration

Register	Description
GSBUSCFG0/1	Leave the default values if the correct power-on values were selected during coreConsultant configuration.
GTXTHRCFG/ GRXTHRCFG	This is required only if you are planning to enable thresholding. Leave the default values if the correct power-on values were selected during coreConsultant configuration.
GSNPSID	The software can read the Synopsys ID register to find the controller version and configure the firmware for any version-specific features.
GUID	Optionally, the software can program the User ID GUID register, if this register is selected for implementation in coreConsultant.
GUSB2PHYCFG	Program the following PHY configuration fields: USBTrdTim, FSIntf, PHYIf, TOUTCal, or leave the default values if the correct power-on values were selected during coreConsultant configuration.
GUSB3PIPECTL	Program the following PHY configuration fields: DatWidth, PrtOpDir, or leave the default values if the correct power-on values were selected during coreConsultant configuration.
GTXFIFOSIZn/ GRXFIFOSIZn	Program these registers based on the speed used for the FIFO. For details, see the “Architecture” chapter in the *DWC SuperSpeed USB 3.0 Controller Databook*. Unless the packet sizes of the endpoints are application-specific, it is recommended that you use the default value.

Table continues on the next page...

Table 11-19. Overriding global registers in host configuration (continued)

Register	Description
GCTL	Program this register to override scaledown, RAM clock select, and clock gating parameters.
GUCTL	If you want to improve the interoperability with different devices, program this register to override the behavior of the controller in Host mode.
DCTL	If you want to improve interoperability as a Debug Capability Target, program the InitU1/U2Ena and AcceptU1/U2Ena fields to '0' to disable U1/U2 entry.

11.2.2.6.2 Host controller capability registers

For register definition, refer to Capability Registers Length and Host Controller Operational Registers (CAPLENGTH) to Runtime Register Space Offset Register (RTSOFF).

11.2.2.6.3 xHCI implementation details

This section discusses xHCI implementation details specific to the USB 3.0 controller.

11.2.2.6.3.1 LHCRST behavior

The xHCI Specification does not describe the programming model of light reset. It only specifies that the Operational and Runtime Registers that are not contained in the Aux Power well are at their default values.

Light reset must only be applied when USB_DCTL[Run_Stop] is equal to '0' to ensure that there is no discrepancy between hardware and software states. If light reset is applied during a transfer, the behavior on the USB is undefined, and may cause a packet to be terminated abruptly.

On light reset, the hardware resets all non-Aux registers to their default values which means that the port state (which is contained in PORTSC) does not change, but USB_DCTL[Run_Stop] is immediately set to '0' and all the context information about connected devices is lost. It is the responsibility of software to re-initialize the controller.

11.2.2.6.3.2 ENT requirements

In the xHCI Specification, the rules about when the ENT (Evaluate Next TRB) flag in a TRB must be set or cleared are not very strict. However, it states that the ENT flag must be set to '1' in the last Normal TRB when a TD ends with an Event Data TRB.

The controller requires software to follow these rules:

1. Regardless of whether the endpoint is stream-capable or not, if a TD ends with an Event Data TRB, the Normal TRB that precedes it must have ENT set to '1'. This Normal TRB may be separated from the Event Data TRB by a Link TRB
2. In all other cases, the ENT flag must be set to '0' Failure to follow these rules results in undefined behavior.

11.2.2.6.3.3 Behavior on babble error

If a babble error is detected and the received data passes all integrity checks, the host controller may write the received data (up to the expected data length) to the data buffer, and the value of the TRB transfer length field in the babble detected error transfer event shall be consistent with the number of data bytes written to the buffer.

The controller does not write the received data to the data buffer. The entire packet is discarded and the residual byte count is written to the TRB transfer length field.

11.2.2.6.3.4 Max_exit_latency_too_large message

If periodic transfers are on every microframe (Binterval 1), the controller reports max_exit_latency_too_large if PING scheduling is enabled. The xHCI scheduler must prefetch data from the system memory ahead of next microframe, manage non-periodic transfer, and schedule PING before ISOC. Because the system memory access and USB responses are not predictable Quality of Service (QoS) is not guaranteed if the xHCI host needs to perform all of the above for every microframe when U1/U2 is enabled. Therefore, the expectation is to disable U1/U2 if periodic data needs to be scheduled every microframe. In addition, because the U1 exit latency is in the order of 10usec, only very low system-level power saving is achieved even if the scheduler allows U1 transition by reducing QoS.

U1, U2, and U3 link states and exit time observed during lab testing:

- U1: Lowest power saving
  - In this state the chip is active, USB controller is inactive, only the Tx transmitter of the PHY is inactive.
  - U1 exit time per link partner: ~10 - 19 usec
- U2: Higher power saving than U1
  - In this state the chip is active, USB controller is inactive, both the Rx and Tx transceivers of the PHY are inactive.
  - U2 exit time per link partner: ~85 - 115 usec
- U3: Highest power saving

- In this state, the chip and USB controller/PHY are inactive.
- U3 exit time per link partner: 200 - 450 usec

Following are the recommendations for driver to handle periodic endpoints and "max_exit_latency_too_large" message:

- If Binterval of any endpoint is '1', then disable U1/U2 and do not schedule PING ("max_exit_latency" set to 0).
- If periodic scheduling is expected on every microframe, then disable U1/U2 and do not schedule PING.
- If periodic scheduling is expected only on every other microframe, then disable U2 and use U1 exit latency for "max_exit_latency" calculation.
- If ((number of hubs +1) * U2 exit latency) >= (2**(Binterval-1) * 125 usec), then disable U2.
- For all other periodic scheduling use U2 exit latency for "max_exit_latency" calculation.
- If the "max_exit_latency_too_large" error is sent to software, then disable U2 and re-issue ep- config with U1 exit latency.
- If "max_exit_latency_too_large" still happens, then disable U1 and re-issue ep_config with max_exit_latency set to 0 (no PING).

11.2.2.6.3.5 Disabled checks

The xHCI specification requires the host controller to perform some checks to ensure that the xHCI driver does not inadvertently set a parameter to an incorrect value. However, some of those checks are incompatible with existing software and so they are currently disabled.

1. Disabled check for EP State in Input Context

   Section “Configure Endpoint” in the *xHCI Specification* states:

   “else // Not all Input Endpoint Contexts identified by Add Context flag fields = ‘1’ are valid Completion Code =Parameter Error.”

   NOTE

   The xHC must check that all referenced contexts are valid before executing the command. If aninvalid context is detected, the state of the Output Device Context shall not change and the a Command Completion Event is generated with the Completion Code set to Parameter Error."

   Section “Configure Endpoint Command Usage” in the *xHCI Specification* states:

“An Input Endpoint Context is considered “valid” by the Configure Endpoint Command if the Add Context flag is ‘1’ and: 1) the values of the Max Packet Size, Max Burst Size, and the Interval are considered within range for endpoint type and the speed of the device, 2) if MaxPStreams > 0, then the TR Dequeue Pointer field points to an array of valid Stream Contexts, or if MaxPStreams = 0, then the TR Dequeue Pointer field points to a Transfer Ring, 3) the EP State field = Disabled, and 4) all other fields are within their valid range of values.”

Therefore, it suggests the controller must verify the EP State field is 'Disabled'. However, the xHCI CV fails on this check and so it is currently disabled.

2. Disabled check for DCS field in Input Context

   Section “Address Device” in the *xHCI Specification* states:

   “The xHC must check that all referenced contexts are valid before executing the command. If an invalid context is detected, the state of the Output Device Context shall not change and the a Command Completion Event shall be generated with the Completion Code set to Parameter Error.”

   Section “Address Device Command Usage” in the *xHCI Specification* states:

   “The Input Endpoint 0 Context is considered “valid” by the Address Device Command if: 1) the EP Type field = Control, 2) the values of the Max Packet Size, Max Burst Size, and the Interval are considered within range for endpoint type and the speed of the device, 3) the TR Dequeue Pointer field points to a valid Transfer Ring, 4) the DCS field = ‘1’, 5) the MaxPStreams field = ‘0’, and 6) all other fields are within the valid range of values.”

   Therefore, it suggests the controller must verify the DCS field is '1'. However, it was found in lab testing that some drivers do not guarantee DCS is always '1', so this check is disabled.

11.2.2.6.4 xHCI debug capability

NOTE

- The controller supports byte-aligned buffers for each TRB of a transfer. This feature prevents the need for buffer copying in cases where the USB device driver receives unaligned data buffers from another application (for example, Ethernet). Whenever the application controls buffer allocation, however, it must allocate SoC bus width

and burst-aligned buffers to facilitate efficient bus and memory utilization.
- The Host controller does not support zero-length TRB in a Transfer Descriptor (TD) that has multiple chain TRBs. However, if the TD has only a single transfer TRB, then it is supports zero-length TRB.

The xHCI Debug Capability (DbC) allows one of the host ports to negotiate as an upstream (device) port. When this happens, the debug port is called the "Debug Target," and the connected host is called the "Debug Host."

NOTE

- To implement the xHCI Debug Capability, select "Yes" for "Enable xHCI Debug Capability?" in coreConsultant (parameter DWC_USB3_EN_DBC = 1)
- For details on the supported configurations and limitations, see "Supported Configurations ".

Section 7.6 of the xHCI Specification explains the functionality of the Debug Capability. However, it explains mostly the software interface, the operation of the DbC at the lower layers, and the handling of the control transfers. The operation of the bulk endpoints for a Debug Target is explained in the rest of the specification because the TRB structure, TD descriptions, Event Ring descriptions, and Endpoint Context are described in other parts of the xHCI Specification.

The DbC feature merges the host (xHCI Specification) software-side programming model with the device USB-side operations. Within the DbC hardware, this translates to a logic that interprets the DbC/xHCI programming model at the upper layers and interacts with the lower layers in the same way the device hardware interacts.

11.2.2.6.4.1 Debug support in a multi-port configuration

Figure 11-20 shows a multi-port xHCI USB 3.0 host controller in which any one of the ports can be used for debugging. A multi-port xHCI host controller supports concurrent host and debug operation.

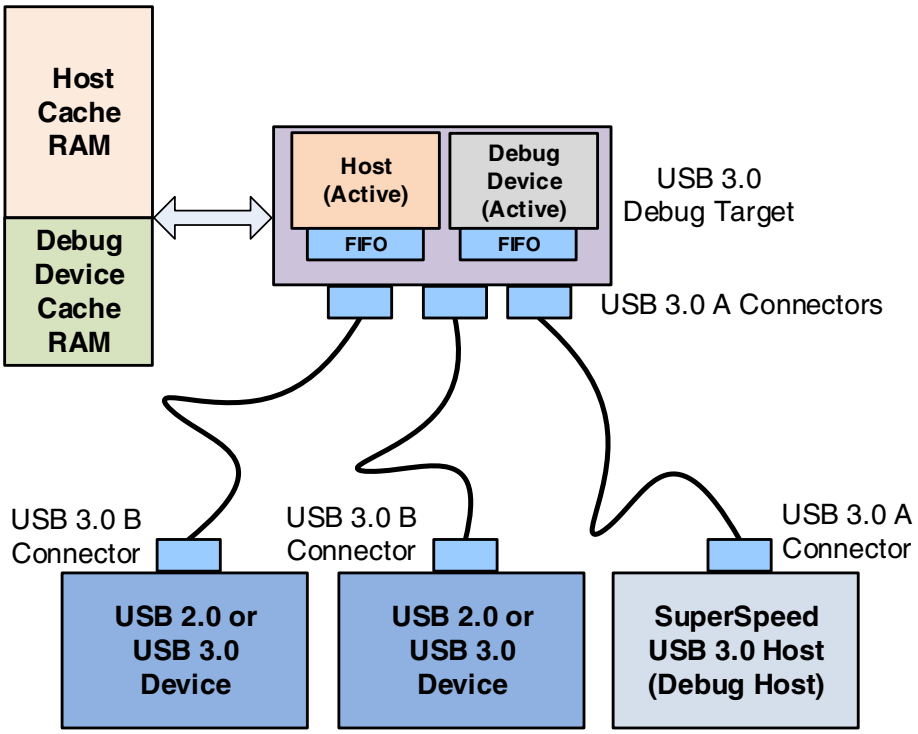


Figure 11-20. Multi-port USB3.0 xHCI-host (Host and debug-device concurrent mode)

11.2.2.6.4.2 Debug support in a one-port configuration supporting either host or debug operation

Figure 11-21 illustrates the operation of a normal one-port xHCI host controller that supports either host or debug operation at any given time. You may need an additional Ethernet, or UART, or USB device controller to support debug operation to support concurrent host and debug operation if you have only a one-port host controller. This adds additional development cost and silicon cost. Also, an additional Ethernet, UART, or USB connector is not desirable in small devices such as mobile phones.

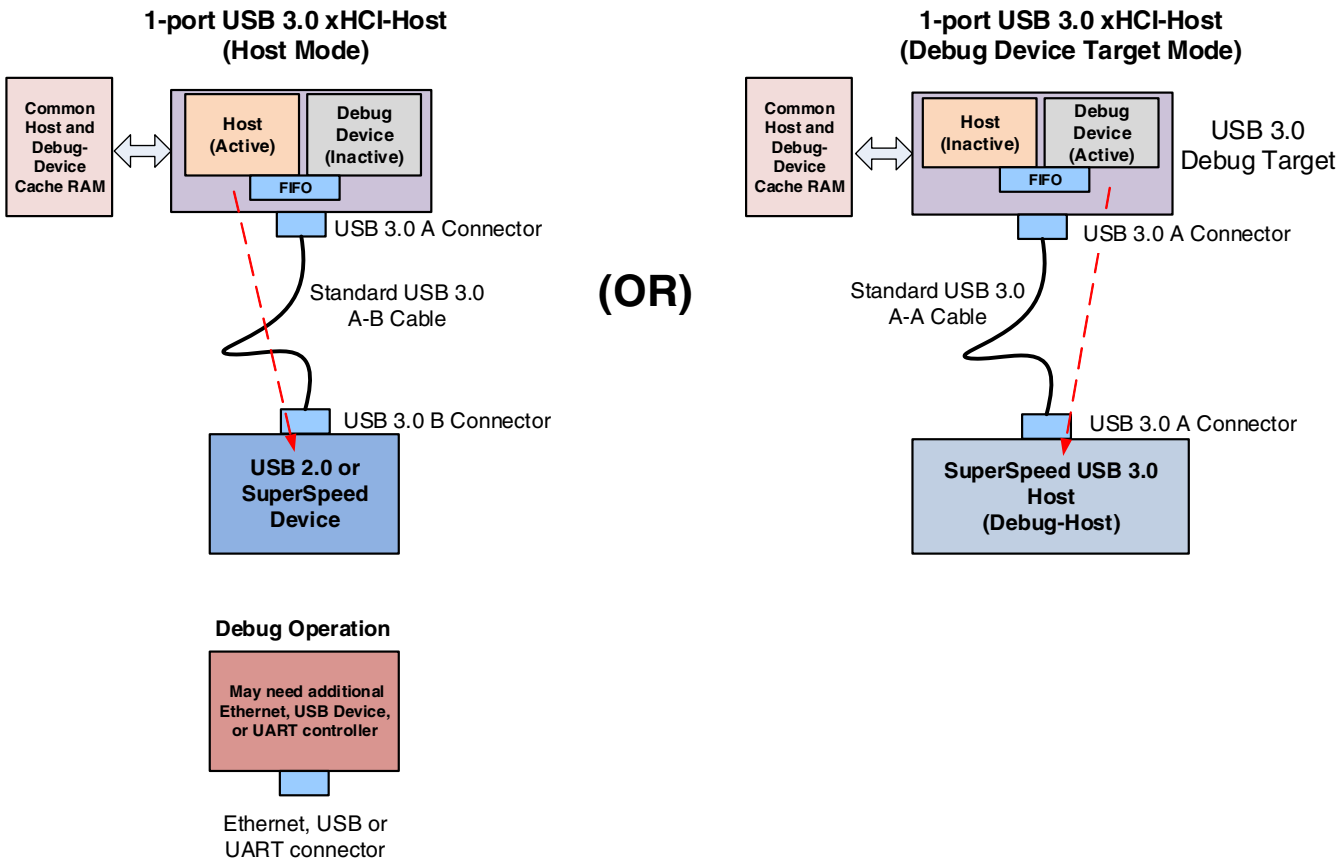


Figure 11-21. One-port USB3.0 xHCI-host (Host or debug mode)

11.2.2.6.4.3 Debug support in a one-port configuration supporting concurrent host and debug operation

The Synopsys xHCI host controller supports concurrent host and debug operation even in a one-port configuration (patent pending). Figure 11-22 illustrates a one-port configuration in which a split USB cable mechanism is used for concurrent host and debug support. The Synopsys xHCI controller supports separate data paths (FIFOs and DMA) for host and debug operation even in a one-port configuration. Similarly, the cache memory also has separate host and debug data structures. The increase in area because of this additional logic is much smaller than adding a separate Ethernet, UART, or USB device controller.

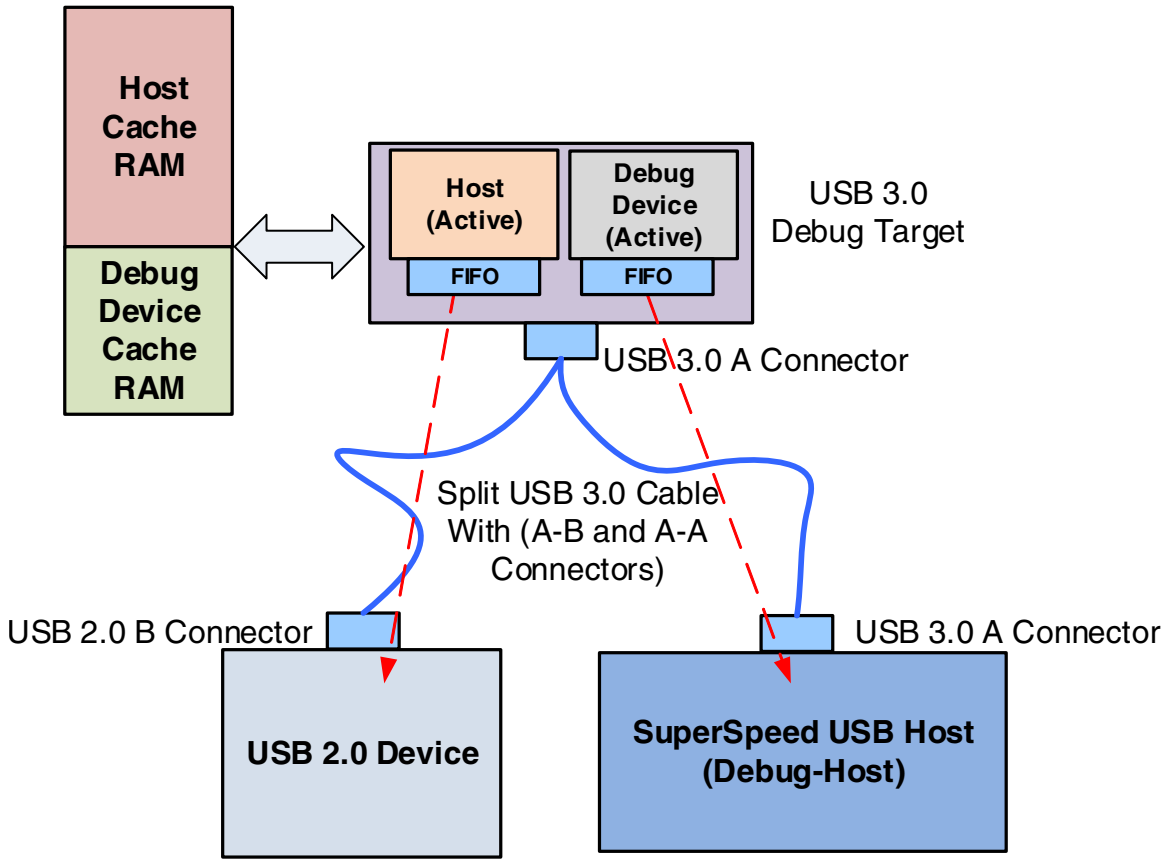


Figure 11-22. One-port usb 3.0 xhci-host (Host and debug-device concurrent mode)

The split cable includes three plugs and two cables.

The details of the plugs are as follows:

- The first plug is a USB 3.0 A plug,
- The second plug is a USB 2.0 B plug, and
- The third plug is a USB 3.0 A plug The details of the cables are as follows:
- The first cable couples the first and second plugs, and includes VBUS, D-, D+, and GND wires.
- The second cable couples the first and third plugs, and includes GND, StdA_SSRX-, StdA_SSRX+, GND_DRAIN, StdA_SSTX-, and StdA_SSTX+ wires.

The first plug is connected to the Synopsys xHCI host controller and the second plug is connected to a USB 2.0 or USB 3.0 device, and the third plug is connected to a USB 3.0 debug host. This provides concurrent USB 2.0 traffic between the Synopsys xHCI host controller and the USB device and SuperSpeed traffic between the Synopsys xHCI host controller and the USB 3.0 debug host.

11.2.2.6.4.4 Operational model

The operational model of the Debug Capability is described in the xHCI Specification. However, further details about the actual implementation of the operational model and its affect on the hardware are detailed in this section.

11.2.2.6.4.4.1 Capability discovery

The software discovers that the DWC_usb3 controller supports the Debug Capability feature by reading through the extended capabilities list. ID 10 is used to indicate that the Debug Capability extended feature is supported, and the software then knows the location of the DbC registers. It can then initialize the DbC registers as described in the xHCI Specification.

11.2.2.6.4.4.2 Port training

The software enables the Debug Capability by setting the `DCCTRL.DCE` field to '1'. This causes the lower layer to advertise that it supports being an upstream port in its Port Capability LMP's. If there are multiple ports exposed, the controller does this across all of the USB 3.0 ports. If the user plugs a host (the Debug Host) into any of these ports, the Debug Host gets trained as a downstream port, and the port that is enabled for Debug Capability gets trained as an upstream port.

When the debug port is selected, the BMU reconfigures TX/RX FIFOs to match the sizes required by the Debug Capability. In addition, the TR switches to select the correct Bus Interfaces to connect the upper and lower layers.

When a port status change occurs in the `DCPORTSC` register for the debug port, a Port Status Change Event is written to the DbC Event Ring, and the `DCST.ER` field is set to '1' to indicate that an event is present.

11.2.2.6.4.4.3 Standard USB request handling

Once the port is trained as an upstream port, the Debug Host initializes the Debug Target (as it normally does for any other device) by issuing a USB reset, `SetAddress`, `GetDescriptor(s)`, and `SetConfiguration`. The Debug Capability LSP module handles each of the control transfers without involving the debug capability driver. After the `SetConfiguration` request, the Debug Target exposes EP1 with an IN and OUT endpoint. The response of the DWC_usb3 controller to each of the possible control transfers is discussed in Control transfers.

11.2.2.6.4.4.4 Transfer handling

The debug capability driver on the Debug Target prepares the TRBs for the OUT endpoint, expecting to receive commands from the host. It rings the “IN TR” doorbell (which corresponds to the OUT endpoint), and this enables the DWC_usb3 controller to receive packets on the OUT endpoint and generate Transfer Events. The software polls the DCST.ER field to determine whether there are events present in the DbC Event Ring. When it sees a transfer event, it knows that the buffers pointed at by the TRBs are now valid. It can interpret the commands sent, and prepare a data or status response on the IN endpoint.

For the IN endpoint, the debug capability driver prepares the TRBs on the “OUT TR”, and then rings the OUT TR doorbell (which corresponds to the IN endpoint). The DWC_usb3 controller starts prefetching the data to be transmitted and transmits ERDY to the host if necessary (that is, if the endpoint is flow controlled on the USB). Retries are automatically handled by the controller which re-fetches the data to be transmitted. There is no communication to the debug capability driver if a retry happens on the USB.

The DWC_usb3 controller supports two transfer rings and assumes that only the following types of TRBs can be on the transfer rings:

- Normal
- Link
- No-Op
- Vendor-Defined: The cores ignores these TRBs as described in section Vendor defined TRB types of the xHCI Specification. The controller generates a “TRB Error” transfer event for any other TRB type.

The controller inherits the following transfer behaviors from the xHCI operational model:

- Waits for a doorbell before fetching TRBs
- Generates Transfer Events when IOC=1
- Fast-forwards to the next TD when the OUT endpoint receives a short packet
- Skips and ignores Vendor-Defined TRBs

11.2.2.6.4.4.5 Endpoint context state

Figure 11-23 shows the Endpoint Context state diagram as illustrated in section 4.8.3 of the xHCI Specification.

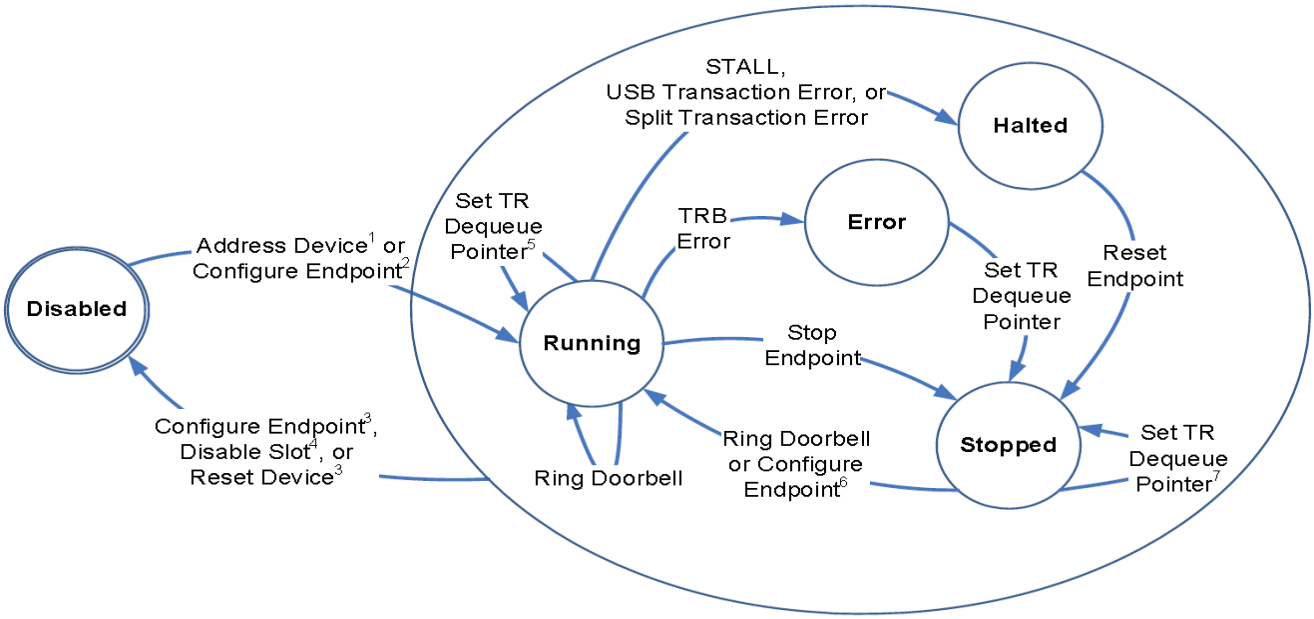


Figure 11-23. xHCI endpoint state transitions

Figure 11-23 is applicable to the endpoints as referenced by the host controller, but for debug capability, the endpoint state diagram is different (see Figure 11-24).

Figure 11-24. DbC bulk endpoint state transitions

The main differences are as follows:

- DbC does not have any “Set TR Dequeue Pointer,” “Stop Endpoint,” or “Reset Endpoint” command
- Address Device and Configure Endpoint are commands issued by the hardware when it receives the

`SetAddress` or `SetConfiguration` control transfer. They are not issued by the software

- The Running state has four substates: Stopped, GetContext, Paused, and Fetching. These substates are not visible to the software, but they are used internally to determine whether the endpoint needs to fetch the Endpoint Context, and whether it is fetching TRBs or not.

The DbC writes the endpoint state to the Endpoint Context during the transitions between the Disabled, Running, Halted, Stopped, and Error states. Even when the transition to the Disabled state is caused by the software setting `DCCTRL.DCE` to '0', the controller still writes the new state to the Endpoint Context despite the fact that the context is not owned by the controller.

11.2.2.6.4.4.6 Halting endpoints

The software can halt one of the bulk endpoints by setting the DCCTRL.HOT or HIT field to '1'. HOT applies to the OUT transfer ring, which is the IN endpoint. HIT applies to the IN transfer ring, which is the OUT endpoint.

When the software writes HOT or HIT with ‘1’, the controller performs the following operations:

1. Propagates the STALL endpoint setting request to the USB clock domain, and waits for the end of the current transaction on the USB

Meanwhile, if the software reads the DCCTRL register, it sees the HOT or HIT field is still '0' which indi- cates that the halt request is not propagated to the USB yet.

1. Starts returning STALL on the USB to any transaction to that endpoint
2. Sets the DCCTRL.HOT or HIT read value to ‘1’
3. Writes out the Endpoint Context with the current value of the TR Dequeue Pointer (after the current transaction is completed on the USB), and “Halted” in the EP State field
4. Generates a Transfer Event if there is an incomplete Normal TRB present in the transfer ring
    a. The TRB pointer references the TRB that the STALL occurred on.
    b. The Transfer Length field may indicate that the TRB is partially completed.
    c. The Completion Code is “Stall Error.”
    d. This event is generated regardless of the value of IOC in the error TRB.
    e. If the TRB ring is empty, or does not contain any Normal TRBs, a “Stall Error” transfer event is not generated because it does not have any TRB to reference.

If the controller is in the process of responding to a transfer on the USB when the software sets the HOT or HIT field to ‘1,’ it completes the current transaction before performing the previous-listed operations. For example, if the controller already responded with an ACK to an OUT packet, that packet is written to the system memory before the transfer event is generated, but any transaction after the HOT/HIT setting receives a STALL response.

When the host sends a ClearFeature(Halt) request to clear the halt on an endpoint, the controller performs the following operations:

1. Clears the STALL condition, and resets the data sequence number to ‘0’.

After this point, any new token on the USB does not receive a STALL response. If the host attempts a transaction to the endpoint before the software prepares a new transfer, the host receives an NRDY response.

1. Writes the Endpoint Context with “Stopped” in the EP State field.

This indicates that the endpoint is not halted on the USB, and the software can update the TR Dequeue Pointer, and ring the doorbell to restart the transfer.

1. Sets the DCCTRL.HOT or HIT read value to ‘0’.

When the software needs to resume transfers on the endpoint, it must wait for the EP State in the Endpoint Context to be “Stopped,” then program the new TR Dequeue Pointer in the Endpoint Context before ringing the doorbell. The controller fetches the new TR Dequeue Pointer from the Endpoint Context the first time the doorbell is rung after the endpoint halted.

The USB endpoint state can be determined by the EP State in the Endpoint Context and the value of the HOT/HIT field as shown in the following table.

Table 11-20. USB endpoint state Vs. HOT/HIT

EP state	HOT/HIT	USB endpoint state
Running	0	Not Stalled
Running	1	Stalled
Halted	0	<Invalid>
Halted	1	Unknown (maybe Not Stalled if the host just sent a ClearFeature (Halt))
Stopped	1	Not Stalled
Stopped	0	Not Stalled

11.2.2.6.4.4.7 TRB error

If an unknown TRB type is encountered on the transfer ring, the controller handles this similar to the HOT/HIT setting described in Halting endpoints. In particular, the controller performs the following operations:

1. Propagates the STALL endpoint setting request to the USB clock domain, and waits for the end of the current transaction on the USB. Meanwhile, if the software reads the DCCTRL register, it sees the HOT or HIT field is still '0' which indi- cates that the halt request is not propagated to the USB yet.
2. Starts returning STALL on the USB to any transaction to that endpoint.
3. Sets the DCCTRL.HOT or HIT read value to ‘1’.
4. Writes out the Endpoint Context with the current value of the TR Dequeue Pointer (after the current transaction is completed on the USB), and “Error” in the EP State field.
5. Generates a Transfer Event
   a. The TRB pointer references to the TRB with the unsupported TRB type.
   b. The Transfer Length field is '0'.
   c. The Completion Code is “TRB Error”.

d. This event is generated regardless of the value of IOC in the unsupported TRB.

Upon receiving the TRB Error transfer event, the software can reset the controller by toggling the DCE from 1->0->1, or can wait for the host to send a `ClearFeature(Halt)` request. When the host sends a `ClearFeature(Halt)` request to clear the halt on the endpoint, the controller performs the following operations:

1. Clears the STALL condition and resets the data sequence number to 0

   After this point, any new tokens on the USB do not receive a STALL response. If the host attempts a transaction to the endpoint before the software prepares a new transfer, the host receives an NRDY response.

2. Writes the Endpoint Context with “Stopped” in the EP State field

   This indicates that the endpoint is not halted on the USB, and the software can update the TR Dequeue Pointer and ring the doorbell to restart the transfer.

3. Sets the `DCCTRL.HOT` or `HIT` read value to ‘0’

   When the software needs to resume transfers on the endpoint, it must wait for the EP State in the Endpoint Context to be “Stopped,” then program the new TR Dequeue Pointer in the Endpoint Context before ringing the doorbell. The controller fetches the new TR Dequeue Pointer from the Endpoint Context the first time the doorbell is rung after the endpoint halted.

11.2.2.6.4.4.8 Event ring

The Debug Capability has its own Event Ring where it places transfer events and port status change events. However, there is no interrupt line that is asserted when there is an event on the Event Ring. The software polls the `DSTS.ER` field to determine when there is an event to be processed.

When there is one location left in the Event Ring before it is full, the controller writes the “Event Ring Full” transfer event to indicate that the Event Ring is too full to write additional events. However, this does not

mean that the events are lost, only that the events are piling up inside the controller and possibly causing back pressure to the Debug Host which eventually causes the controller to stop releasing credits on the USB 3.0 link.

11.2.2.6.4.4.9 Disabled endpoint handling (DCCTRL.DRC/DCR)

The xHCI Specification does not precisely describe the handshake between the hardware and software when an endpoint becomes Disabled and the DbC exits the Configured state, leaving open several different interpretations. The DWC_usb3 controller adheres to the following description of the Endpoint Disabled handling. This text has been proposed as an errata to the xHCI Specification.

There are several conditions which cause an endpoint to transition from the Running to the Disabled state (DbC exits the Configured state). This causes the DCCTRL.DCR field to transition from a '1' to '0' and DCCTRL.DRC to be set to '1'.

- The Debug host initiates a USB Reset.
- The Debug host issues a SetConfiguration(0) device request.
- Timeout occurs in the Port State Machine.
- The USB cable is disconnected.
- The Debug Capability driver writes '0' to the DCPORTSC.PED field. When the bulk endpoint becomes Disabled, the following occurs:
- If there is a valid Transfer TRB on the Transfer Ring, a Transfer Event is generated.
  - The TRB Pointer field of the Transfer Event references the Transfer TRB that the event occurred on.
  - The TRB Transfer Length field of the Transfer Event indicates that the Transfer TRB is partially completed.
  - The Completion Code field of the Transfer Event indicates a USB Transaction Error.
  - This Transfer Event is generated irrespective of the IOC flag being set or not in the associated Transfer TRB.
  - The DbC increments its TR Dequeue Pointer.
- The Endpoint Context is written with
  - TR Dequeue Pointer set to its current value
  - Endpoint State set to Disabled

The software detects this condition by reading the DCCTRL.DRC field as '1' and by reading the Endpoint State as Disabled. In response, the software reads the TR Dequeue Pointer field in the Endpoint Context and updates it to point to the next TRB to be executed after it clears the DCCTRL.DRC field and rings the doorbell.

NOTE

The DbC is not required to advance the Dequeue Pointer of an endpoint to the next TD boundary when a Disabled Endpoint condition occurs.

11.2.2.6.4.4.10 Resets and asynchronous operations

This section discusses the several asynchronous operations which affect the debug capability.

11.2.2.6.4.4.10.1 Resets

- When the software sets DCE from 1 to 0, it invokes a “DCE Reset.”
  - The controller sets the DCPORTSC status change bits to '0.'
  - If the controller is previously in the DbC state, the port exits the DbC state.
- When the Debug Host performs a Hot or Warm Reset, it invokes a “USB Reset.”
  - The controller stops the transfer rings for endpoint 1, and stops responding on this endpoint.
  - The controller does not resume transfers until the Debug Host reconfigures the Debug Target and the software rings the doorbell again.
- When the software sets HCRST or GCTL.CoreSoftReset, it invokes a “Chip Hardware Reset.”

  This causes all asynchronous resets in the controller to be asserted. DCPORTSC is set to 0, DCCTRL is set to 0 (DCE=0), all state machines go back to IDLE, the sticky PORTSC registers are reset, and the PHYs are reset.

- When the software sets LHCRST, it invokes a “Light Reset.”

  This resets the upper layer, the debug capability, and its registers, but not the sticky PORTSC registers. If the debug capability is in an “Upstream Mode” state, this reset causes the DCE bit to go to '0' which in turn causes the root hub (which does not reset on light reset) to go back to downstream mode only.

- GUSB2PHYCFG.SoftReset

  This resets only the individual USB 2.0 PHY.

- GUSB3PIPECTL.SoftReset

  This resets only the individual USB 3.0 PHY.

11.2.2.6.4.4.10.2 Other asynchronous operations

- When the software sets DCE from 0 to 1, it invokes a “DCE Enable.”

  The LSP fetches the DbCIC, enables endpoint 0, and allows upstream connections on its ports.

- When the host sends the device a `SetConfiguration(1)`, it invokes a “Device Configure.”

  The LSP configures EP1 IN and OUT by clearing any residual state, such as STALL, Data Sequence Number, and the SPR flag (synchronization between LSP and PTL).

- When the host sends the device a `SetConfiguration(0)`, it invokes a “Device Deconfigure.”

  The controller stops transfer rings, and stops responding on endpoint 1.

- When the software sets `DCPORTSC.PED` to '1', it invokes a “Port Enable.”
- When the software sets `DCPORTSC.PED` to '0', it invokes a “Port Disable.”

  The port still belongs to the debug capability, but it disconnects from the Debug Host if it was already connected.

11.2.2.6.4.4.11 U1/U2/U3 Entry

The Debug Host enables the Debug Target to initiate U1/U2 entry by sending the `SetFeature(U1_ENABLE)` or `SetFeature(U2_ENABLE)` requests. The U1/U2 entry can be forcefully disabled by the software by setting `DCTL.InitU1/U2Ena` and `DCTL.AcceptU1/U2Ena` to '0'. This may be necessary to improve interoperability with the Debug Hosts and Hubs.

The Debug Device, similar to DWC_usb3 device, accepts and initiates U1/U2 entry. This means that the Debug Device keeps track of its endpoints and the Packet Pending (PP) flag from the Debug Host. If all the endpoints are idle, the controller accepts U1/U2 entry. If the endpoints are idle, and the host has enabled the U1_ENABLE/U2_ENABLE feature, then the Debug Target initiates U1/U2 entry on the link.

The USB 3.0 protocol allows a SuperSpeed host and device to negotiate with each other and decide when to bring their link into a lower power state (U1 or U2). This is accomplished by each side making their individual decisions based on whether they have any traffic currently pending or expect any traffic soon. The device requests low power entry by transmitting an LGO_U1 (or LGO_U2) Link Command and exits low power by using a LFPS handshake.

The Debug Capability uses the information from the host (bus side) and the software (the application side) to decide the best time to request a low power entry and exit from the low power state. The same calculation is used by the controller to decide whether to accept a low power request from the host. The controller maintains the state information about each endpoint and whether it is "active" or "paused". If all enabled endpoints are paused from either the application side or the bus side, the low power entry is allowed. If at least one endpoint is not paused, the low power entry is not allowed.

To indicate that there is no pending traffic and allow the link to go into a low power state, the software pauses all enabled endpoints. Otherwise, an endpoint is considered active on the application side. For these endpoints, the host indicates that it is paused using one of the following ways:

- By setting the Packet Pending (PP) flag to '0' in a DPH
- Setting PP to '0' and NumP to '0' in an ACK TP

Exception: If the data packet payload has a CRC error, the DbC responds with an ACK (Retry=1), and the endpoint is not paused. Otherwise, if PP=1 or Deferred=1, the endpoint is considered to be active on the bus side.

Control endpoints have the following additional condition:

The controller does not allow low power entry during a control transfer (starting from the reception of a setup packet and ending at the completion of the status stage).

If all the enabled endpoints are paused by the software or the host, the controller issues an LGO_U1 (or LGO_U2) link command or it responds with an LAU link command to the low power request of the host or hub. Otherwise, the controller rejects the low power requests with an LXU link command.

Other related information:

- For bulk endpoints, the USB 3.0 Specification says that the device may go to U1/U2 500ms (tERDY- Timeout) after transmitting ERDY. However, it is not possible to force the DbC to enter U1/U2 in this scenario where the host is unresponsive to the ERDY.
- Because U1/U2 entry is based on the presence of packets in the TX FIFO and not based on the TRBs that are available to the controller, it is possible that an IN endpoint responds NRDY, goes to U1, then U0, then transmits an ERDY if the timing and TX FIFO size work out such that the controller cannot keep at least 1 packet in the TX FIFO at all times.
- In addition, before the final decision is made to either initiate or accept a low power state, the DbC checks the state of `DCTL.InitU1Ena/RejectU1Dis/InitU2Ena/RejectU2Dis` bits. If the device receives Set Link Function LMP with `Force_LinkPM_Accept` bit asserted, the DbC accepts a low power request from the host independent of endpoints state and DCTL Power control settings.

For U3, the Debug Device, similar to the normal device, enters and exits U3, as required in the USB 3.0 specification. The Debug Device does not transmit a Function Wake Device Notification because it is not Remote Wakeup capable.

11.2.2.6.4.5 Data structures

The xHCI data structures that are produced and consumed by the Debug Capability are:

- Debug Capability Info Context Data Structure (DbCIC)
- Endpoint Context
- Transfer TRB
- Event TRB

Each of these data structures are described in the xHCI Specification, but details specific to the implementation are described in this section.

11.2.2.6.4.5.1 Debug capability info context data structure

The Debug Target fetches the Debug Capability Info Context data structure when the software sets DCE=1, and uses the information to respond to `GetDescriptor(STRING)` requests from the host.

11.2.2.6.4.5.2 Endpoint context

In Debug Capability, the Endpoint Contexts are fixed to 64 bytes each. Figure 11-25 shows the Endpoint Context.

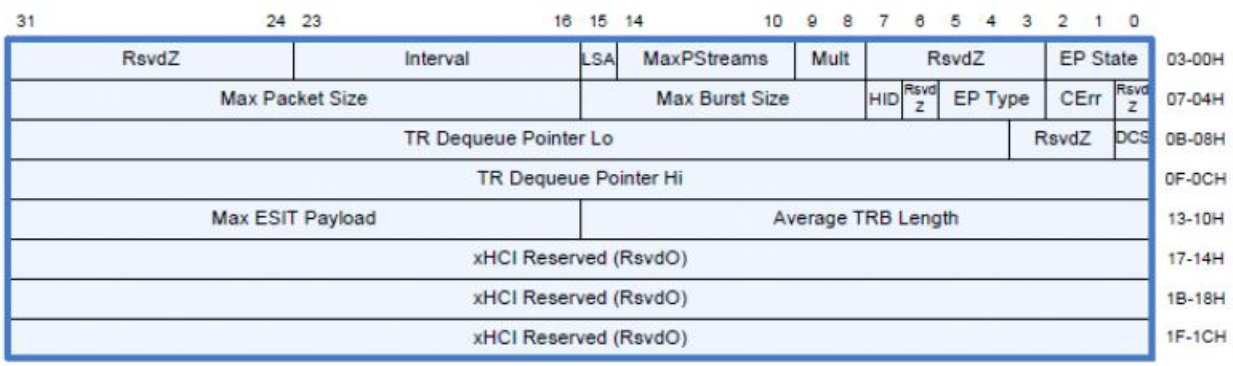


Figure 11-25. Endpoint Context

Even though the Endpoint Context format is the same, only EP State, TR Dequeue Pointer, and DCS fields of the EP Context are applicable to a Debug Target. The following fields are ignored:

- Max Packet Size: Always assumed to be 1024 bytes, unless `GCTL.FRMSCLDWN` is set to a different value, in which case the MPS is 1024 << `GCTL.FRMSCLDWN`
- EP Type: Always assumed to be 2 (Bulk)
- Max Burst Size: Assumed to be 15 (`DWC_USB3_DBC_MAX_BURST_SIZE`) for both IN and OUT endpoints. However, for the OUT endpoint the controller fills in the NumP (number

of receive buffers) field of the ACK TP based on the value in the DCFG.NumP register, which is 4 (DWC_USB3_DEVDB- C_OUT_NUMP) by default.

The other fields (Mult, MaxPStreams, LSA, Interval, CErr, HID, Average TRB Length, and Max ESIT Payload) are not applicable.

11.2.2.6.4.5.3 Transfer TRBs

The following are the only supported Transfer TRBs types:

- Normal
- Link
- No-Op
- Vendor-Defined

NOTE

- The Interrupter Target and BEI fields are ignored because the debug capability does not have an interrupt line.
- The IDT flag in Transfer TRBs must not be set to '1.'
- The ENT flag is ignored, as it is primarily used for stream-based endpoints.

11.2.2.6.4.5.4 Event TRBs

The controller produces the following types of Event TRBs:

- Port Status Change Event
- Transfer Event

The Completion Codes that are generated are:

- Success
- TRB Error
- Stall Error
- Short Packet
- Event Ring Full Error
- USB Transaction Error (see Disabled endpoint handling (DCCTRL.DRC/DCR))

The Babble Detected completion code is not generated, because the controller responds with ACK(Retry) to a TD Babble scenario. The Endpoint ID field in transfer events is ‘1’ for OUT Transfer Ring and ‘2’ for IN Transfer Ring.

11.2.2.6.4.6 Requirements, assumptions, and limitations

This section discusses the various requirements, assumptions and limitations of the xHCI Debug Capability Table 11-21 gives a summary of the limitations.

Table 11-21. xHCI debug capability limitations

Feature	Comments
Incompatible Features	DbC can be configured along with Hibernation and Clock Gating. However, these features are turned off automatically when Debug Capability is enabled by the DbC driver.
U1/U2 Support	The controller supports autonomous entry into U1 or U2, but it is recommended to disable this support via the DCTL register to improve interoperability with the Debug Hosts and Hubs.
Event Data TRB	The controller does not support the use of Event Data TRBs on the Debug Capability Transfer Rings. The Microsoft Debug Capability driver does not use them.
SetFeature(Halt) on EP1	If the debug host issues a SetFeature(ENDPOINT_HALT) request to the debug target's bulk endpoint 1, it receives a STALL response. This request is not supported.
ERDY	The controller transmits an ERDY when it is ready to transmit or receive a packet on the corresponding bulk endpoint, not explicitly when the endpoint's doorbell is rung by the software.
Chained TRBs greater than TRB cache size	The debug capability driver must not create a multi-TRB TD that describes smaller than a 1K packet that spreads across 8 or more TRBs on either the IN TR or the OUT TR.
TD Babble	If the host transmits a packet larger than the TD that the Debug Capability Driver has been setup for it, the debug target respondd repeatedly with ACK(Retry). No Transfer Event is generated for this scenario.
Latency Tolerance Message	The Debug Target does not transmit LTM Device Notifications.
HOT/HIT and Transfer Events	When the Debug Capability Driver sets HOT or HIT to '1', the controller generates a Transfer Event with "Stall Error" only if a Normal TRB is valid in the Transfer Ring. For more information, see Halting endpoints.
Reading DCCTRL.HOT/HIT	Immediately after setting HOT or HIT to '1', the field returns '0' when read until the STALL condition has internally propagated. When the controller is prepared to return STALL on the USB, the HOT/HIT field returns '1' when read. For more information, see Halting endpoints.
Residual DMA after DCE=0	The Debug Capability specification has no feedback after the software sets DCE to '0' to know when the DbC has halted (ala USBSTS.HCHalted). Therefore, it is possible that the controller continues DMA on the system bus to and from data buffers, TRBs, or Endpoint Contexts even after the software has set DCE=0. There is no definition for how long the DMA persists.
DCE Toggling	The software must wait at least 1 millisecond after setting DCE to '0' before setting it back to '1'. For more information, see DCE toggling.

Table continues on the next page...

Table 11-21. xHCI debug capability limitations (continued)

Feature	Comments
Overcurrent on Debug Port	If an overcurrent occurs on the debug port, the controller generates an xHCI port status change event and set the OCA and OCC bits in the associated PORTSC register. For more information, see Overcurrent on debug port.
Zero Length TD's for the OUT endpoint	Zero-length TD's are not allowed on the IN Transfer Ring (OUT endpoint). If the Debug Capability Driver is expecting a zero-length packet, it must prepare a TD that has at least 1 byte.
HCRST behavior	The xHCI Specification mentions that the DbC updates the Endpoint Context TR Dequeue Pointer if HCRST is set to '1'. However, because the debug capability is reset on HCRST, the controller does not update the Endpoint Context TR Dequeue Pointer.

11.2.2.6.4.6.1 Supported configurations

The DWC_usb3 controller can be configured with the Debug Capability hardware in any configuration supported by a host configuration. In particular, the Debug Capability logic can co-exist with the following other configurations:

- Host Only (`DWC_USB3_MODE=1`)
- Dual Role Device (`DWC_USB3_MODE=2`)
- Clock Gating (`DWC_USB3_EN_PWROPT > 0`)
- Hibernation (`DWC_USB3_EN_PWROPT=2`)

However, when the debug capability is enabled (`DCCTRL.DCE=1`), the following features (if configured) are disabled by the hardware:

- Clock gating (`GCTL.DsblClkGtng=1`)
- Hibernation (`GCTL.GblHibernationEn=0`)

11.2.2.6.4.6.2 RAM requirements

When the Debug Capability is enabled (that is, when parameter DWC_USB3_EN_DBC is 1), additional internal RAM is needed due to:

- Extra TxFIFOs and RxFIFOs
- Extra space for the DbC LSP Cache
- DbC RAM-based CSRs

11.2.2.6.4.6.2.1 Additional TxFIFOs and RxFIFOs

ITo support simultaneous host and debug target operation, most configurations require additional RAM and FIFO controllers. One exception is a DRD configuration with a large device TxFIFO RAM requirement.

Table 11-22 indicates the increased RAM requirement, which depends on the number of Host U3 root ports and SS USB instances as specified through coreConsultant. The last column indicates the presence of an additional set of sink/source 2clkfifo buffers. Each set consists of three flop-based FIFOs that are MDWIDTH wide and 8 deep if MDWIDTH is 32 or 64, or 4 deep if MDWIDTH is 128.

Table 11-22. DbC TxFIFO RAM requirements based on configuration

Num ber of U3 Ports	Number of SS USBInstances	TxFIFO RAM +	TxFIFO Controller +	RxFIFO RAM +			Cache RAM +	Sink/ Source Buffer +	
				MDW 32,64	MDW 128	RxFIFO Controller +	MDW 32,64		MDW 128
1	1	256B+2W[1]	1	0	0	0	64B	128B	0
2	1	3.25KB+6W	2	3KB+40B	3KB+64B	1	4*64B	4*128B	1
2	2	256B+2W	1	0	0	0	64B	128B	0
4	2	3.25KB+6W	2	3KB+40B	3KB+64B	1	4*64B	4*128B	1
4	4	256B+2W	1	0	0	0	64B	128B	0

1. W is the number of bytes in a FIFO word. If MDWIDTH is 32, then W is 4; if MDWIDTH is 64, then W is 8; if MDWIDTH is 128, then W is 16.

In all cases in which an additional TxFIFO controller is necessary, an additional TXQ FIFO controller is also necessary along with additional space in RAM0 (64B if MDWIDTH is 32 or 64, or 128B if MDWIDTH is 128). Similarly, in cases in which an additional RxFIFO is necessary, an additional RXQ FIFO controller and RIQ FIFO controller are necessary along with RAM0 space as indicated previously.

Examples:

1. In the single-port configuration (see first row in Table 11-22), one additional TxFIFO controller is necessary along with 256 bytes plus two FIFO words of TxFIFO RAM. If MDWIDTH is 64, then the TxFIFO RAM requires an additional 272 bytes (256 + 2 * 8), and the Cache RAM requires 64 bytes for the extra TXQ.
2. In a two-port configuration with one SS USB instance (see second row in Table 11-22), two additional TxFIFO controllers are necessary along with 3.25KB + 6W TxFIFO RAM, assuming the default three packet buffering requirement for U3 burst operation. One additional RxFIFO is necessary along with 3KB + 40B of RxFIFO RAM, and 4*64B of Cache RAM for the four additional FIFO controllers for the DMA queues (TXQ (2), RXQ, RIQ).
3. In a single-port DRD configuration with four device IN endpoints and a total device TxFIFO RAM requirement larger than the host TxFIFO RAM requirement plus DBC requirement in Table 11-22 (similar to last row but with specified device configuration enabled), no additional FIFO controllers are necessary and no additional FIFO RAM is necessary.

11.2.2.6.4.6.2.2 DbC LSP cache

The DbC LSP uses a part of the internal RAM to track transfers to and from the Debug Host. This is called the DbC LSP Cache. The DbC LSP Cache consists of the following sections:

- 12 DWORDs for the DbCIC (String Descriptor addresses and lengths plus padding)
- 4 DWORDs for EP1 OUT Endpoint Context (referred to in the xHCI specification as the “IN Transfer Ring”)
- 8 DWORDs for EP1 OUT Endpoint Scratchpad
- 4*N DWORDs for EP1 OUT TRB cache
- 4 DWORDs for EP1 IN Endpoint Context (referred to in the xHCI specification as the “OUT Transfer Ring”)
- 8 DWORDs for EP1 IN Endpoint Scratchpad
- 4*N DWORDs for EP1 IN TRB cache

The “N” in the previous equations is the number of TRBs per endpoint to be cached. The default value is 16, so the default space required for the DbC LSP Cache is 12+4+8+4*16+4+8+4*16 = 164 DWORDs, or 656 bytes. This space is added to RAM0.

11.2.2.6.4.6.2.3 DbC RAM-based CSRs

Several of the DbC registers are stored in the internal RAM. These types of CSRs are called “RAM-based CSRs,” and they increase the internal RAM requirement. For DbC, the following registers are RAM-based and increase the internal RAM requirement by 32 bytes:

- DCERSTBA_LO
- DCERSTBA_HI
- DCERDP_LO
- DCERDP_HI
- DCCP_LO
- DCCP_HI
- DCDDI1
- DCDDI2

This space is added to RAM0.

11.2.2.6.4.6.3 Port state machine/root hub

Multiple Debug Target Ports

- Any port may become a debug port, but only one port can be the debug port at a point of time.

- After one port is selected as the debug port, the others stop reporting upstream capable in their Port Capability LMP's.
- Even though the controller prevents two ports from training as upstream ports, exhaustive testing is not done for this scenario because it is assumed that the user does not attempt to simultaneously plug two Debug Hosts into two different ports on the Debug Target.

Port power

- The `vbus_ctrl` is not driven to 0 when the port becomes a debug port. The `vbus_ctrl` is based on the `PORTSC` register.

Overcurrent

- If an overcurrent occurs on a port that is acting as a Debug Target, the `PORTSC` register associated with that port reports the overcurrent.

11.2.2.6.4.6.4 Control transfers

When the controller handles control transfers autonomously, it always responds with NRDY for the first time it receives a STATUS TP, then sends ERDY, and ACKs the status phase when the host repeats it. The reason the controller never ACKs the STATUS TP for the first time is to account for the misbehaving hosts that might continue the data phase for more transactions, or might initiate the data phase in the incorrect direction (IN vs. OUT). By waiting for the status phase, the controller is guaranteed that the host has concluded the data phase.

Table 11-23 lists the controller'd responses to the standard device requests.

NOTE

- As shown in Table 11-23, the Debug Target responds STALL to a SetFea- ture(ENDPOINT_HALT) request from the host. The debug capability cannot be halted from the host, it can only be halted from the debug capability driver.
- The Debug Target responds STALL to a SetInterface(0) request from the host. The USB 2.0 and USB 3.0 specifications allow this behavior when a device supports only one interface.

Table 11-23. Control transfer responses

Request	Index/Value	State	Response
SetFeature	ENDPOINT_HALT (EP1)	All states	STALL
	ENDPOINT_HALT (EP0)	All states	ACK

Table continues on the next page...

Table 11-23. Control transfer responses (continued)

Request	Index/Value	State	Response
	FUNCTION_SUSPEND	Configured	ACK
	U1_ENABLE	Configured	ACK
	U2_ENABLE	Configured	ACK
	LTM_ENABLE	All states	STALL
	B3_NTF_HOST_REL	All states	STALL
	TEST_MODE	All states	STALL
ClearFeature	ENDPOINT_HALT (EP1)	Configured	ACK
	ENDPOINT_HALT (EP0)	All states	ACK
	FUNCTION_SUSPEND	Configured	ACK
	U1_ENABLE	Configured	ACK
	U2_ENABLE	Configured	ACK
	LTM_ENABLE	All states	STALL
	B3_NTF_HOST_REL	All states	STALL
	TEST_MODE	All states	STALL
GetConfiguration		Addressed/Configured	Return 0 (unconfigured) or 1 (configured) ACK
SetDescriptor		All states	STALL
GetInterface		Configured	Return 0 ACK
SetInterface		All states	STALL&
GetStatus	Device	Addressed/Configured	ACK
	Interface	Configured	ACK
	Endpoint (EP0)	Addressed/Configured	ACK
	Endpoint (EP1)	Configured	ACK
GetDescriptor	DEVICE	All states	ACK
	CONFIGURATION	All states	Fixed value + cC fields
	BOS	All states	Fixed value + cC fields
	STRING (supported desc_index 0 to 3)	All states	Read from external memory
	STRING (unsupported desc_index > 3)	All states	STALL
	INTERFACE	All states	STALL
	ENDPOINT	All states	STALL
	INTERFACE_POWER	All states	STALL
	DEBUG	All states	STALL
	INTERFACE_ASSOCIATION	All states	STALL
	DEVICE_CAPABILITY	All states	STALL
	SUPERSPEED_USB_ ENDPOINT_COMPANION	All states	STALL
SetAddress	non-zero	Default/Addressed	ACK
SetConfiguration		Addressed/Configured	ACK

Table continues on the next page...

Table 11-23. Control transfer responses (continued)

Request	Index/Value	State	Response
SetIsochDelay		All states	ACK
SetSEL		Configured	ACK
SynchFrame		All states	STALL
*	Wrong Index/Value or wrong wLength	*	STALL
*	non-standard bmRequest Type (such as, Class/Vendor/ Reserved)	*	STALL
*	*	Non-supported states	STALL

11.2.2.6.4.6.5 Flow control

The controller has the following behavior:

- When an IN endpoint is flow controlled, the Debug Device transmits an ERDY when it has pre-fetched a packet into the TX FIFO.
- When an OUT endpoint is flow controlled, the Debug Device transmits an ERDY when it has fetched enough TRBs to receive a `MaxPacketSize` packet or the last packet of a TD.

The controller does not transmit an ERDY on a doorbell ring.

11.2.2.6.4.6.6 Bursting behavior for OUT and IN transfers

When the controller, as a Debug Target, responds to OUT and IN transfers, it uses the following mechanisms to throttle the Debug Host:

- For OUT transfers, setting the NumP field in the ACK TP
  - It is not possible or practical for the controller to predict the exact rate of packets entering and exiting the receive FIFO, so the `NumP` field is set based on the `DCFG.NumP` field, which defaults to the `DWC_USB3_DEVDBC_OUT_NUMP` parameter (a value of 4) on reset.
  - You can change this field during controller configuration based on your system simulations to allow or prevent the host from initiating large bursts.
  - This is the same behavior as the Synopsys USB 3.0 Device controller.
- For IN transfers, setting the EOB field in the DPH
  - The controller sets EOB as 1 during an IN transfer burst if the transmit FIFO does not have another packet after the one that is being transmitted to force flow control on the endpoint.

- Once there is a full packet in the transmit FIFO, the controller automatically transmits an ERDY to the Debug Host.
- This is the same behavior as the Synopsys USB 3.0 Device controller.

11.2.2.6.4.6.7 TRB cache size and transfers

The controller contains a TRB cache, one for the IN endpoint and one for the OUT endpoint, that holds a number of TRBs (configured by `DWC_USB3_DBC_TRBS_PER_TRANSFER`, default is 16).

For the bulk IN endpoint, the controller constructs a full packet with the TRBs that are present in the internal cache. This means that the debug capability driver must not create a multi-TRB TD that describes less than a `MaxPacketSize` packet (1K) that spreads across `DWC_USB3_DBC_TRBS_PER_TRANSFER`-1 or more TRBs.

For the bulk OUT endpoint, the controller receives a full packet or a short packet with the TRBs that are present in the internal cache. This means the debug capability driver must not create a multi-TRB TD that describes a buffer less than 1K that spreads across `DWC_USB3_DBC_TRBS_PER_TRANSFER`-1 or more TRBs.

In each of the previous equations, the "-1" is added to account for at most one Link TRB, which may be present in a TD. If more than one Link TRB is present in a TD, the requirement shrinks by the maximum number of Link TRBs that can be in a TD.

11.2.2.6.4.6.8 DCE=0 restrictions

When the software sets `DCCTRL.DCE=0`, several operations take place inside the controller. This section describes some of the assumptions and restrictions when this occurs.

11.2.2.6.4.6.8.1 Data structure ownership

Section 7.6.9 of the xHCI Specification states "While the Debug Capability Enable (DCE) bit in the Debug Capability Control Register (DCCTRL) is '1', the xHC maintains ownership of the data structures." This also implies that when DCE=0, the controller does not own the data structures (DbC Info Context, OUT EP, and IN EP Context). However, due to residual DMA (see Residual DMA), the controller can be in the process of writing to these data structures and therefore there is no guarantee that the controller prevents these writes immediately when DCE is set to '0'.

11.2.2.6.4.6.8.2 Residual DMA

Because the controller may be busy performing a read or write DMA operation on the system bus when the software sets DCE=0, it is not possible to immediately abort those operations. Therefore, the controller completes the DMA operations on the system bus,

and finds a graceful point to finish the DbC halt operation. Unfortunately, there is no defined “Halted” bit in the xHCI Specification, so the software cannot determine when the DMA operations have completed.

11.2.2.6.4.6.8.3 DCE toggling

Because there are a set of operations performed within the controller when the software sets DCE from '1' to '0', and there is no feedback when the controller has completed its residual DbC operations, the software must wait at least 1 millisecond after setting DCE to '0' before setting it back to '1.'

11.2.2.6.4.6.9 TD babble response

A “TD Babble” is when the debug capability driver prepares a TD that is smaller than the amount of data sent by the Debug Host to the OUT endpoint. For example, the debug capability driver prepares an 8 byte TRB to receive an 8 byte packet from the host, but the host instead sends a 1K packet.

The xHCI Specification allows several different responses in this situation. The DWC_usb3 controller responds on the USB with an ACK(Retry) if this happens. If the host continues to send the same packet three times, the host will timeout the transfer and likely reset the Debug Target. There is no indication of the retries to the debug capability driver, but when the reset occurs, the `DCCTRL.DCR` field is set to '0' and DRC is set to '1' to indicate that the Debug Target exited the Configured state, and requires the debug driver to restart the bulk transfers.

11.2.2.6.4.6.10 TRB Types

The controller supports Normal, Link, No-Op, and Vendor Defined TRB Types. Event Data TRBs are not supported, as they are not required by the Microsoft Debug Capability driver.

11.2.2.6.4.6.11 Latency tolerance message

The controller, when acting as a Debug Target, does not transmit LTM Device Notifications.

11.2.2.6.4.6.12 Overcurrent on debug port

Because some systems use a single overcurrent signal for all ports, the controller cannot determine if an overcurrent on the debug port is on the port itself or whether it is from another downstream port. Therefore, if an overcurrent is detected on the debug port, the controller still sets the OCA and OCC bit in the PORTSC register associated with the port, and generates a port status change event to be consumed by the xHCI driver.

11.2.2.6.4.6.13 Vendor device test LMP

The controller, acting as a Debug Target, has the capability of receiving the Vendor Device Test LMP without error. However, the LMP is dropped with no indication to the software.

11.2.2.7 Device physical endpoint-specific commands

11.2.2.7.1 Command 1: Set endpoint configuration: DEPCFG

This command sets the physical endpoint configuration information.

NOTE

If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

Table 11-24. Command 1: Set endpoint configuration parameters: DEPCFG

Field	Description
Parameter 2	
31-0	Set to endpoint state when the Config Action field of Parameter 0 is 1. Otherwise, this is Reserved.
Parameter 1	
31	FIFO-based Set to '1' if this isochronous endpoint represents a FIFO-based data stream where TRBs have fixed values and are never written back by the core.
30	Bulk-based Set to '1' if this isochronous endpoint represents a bulk data stream that ignores the relationship of bus time to the intervals programmed in TRBs.
29-25	USB Endpoint Number bit[29:26]: Endpoint number bit[25]: Endpoint direction: 0 OUT 1 IN

Table continues on the next page...

Table 11-24. Command 1: Set endpoint configuration parameters: DEPCFG (continued)

Field	Description
	Physical endpoint 0 (EP0) must be allocated for control endpoint 0 OUT. Physical endpoint 1 (EP1) must be allocated for control endpoint 0 IN.
24	Stream Capable (StrmCap) Indicates this endpoint is stream-capable (MaxStreams != 0).
23-16	bInterval_m1 Set to the bInterval value minus 1. The valid values for this field are 0 through 13. The bInterval value is reported in the endpoint descriptor. When the core is operating in Full-Speed mode, this field must be set to 0.
15	Set to '1' if this bulk endpoint utilizes the External Buffer Control (EBC) mode. This limits the number of outstanding DMA transfers to one read and one write. This is a requirement for the EBC to operate properly. Using EBC without setting this bit to '1' results in undefined behavior.
14	Reserved
13-8	Bits 13-8 are used for Device Endpoint Specific Event Enable (DEPEVTEN). If an enable bit is set to 0, the corresponding event is not generated. Bit 13 Stream Event Enable (StreamEvtEn) Bit 12 Reserved Bit 11 Reserved Bit 10 XferNotReady Enable (XferNRdyEn) Bit 9 XferInProgress Enable (XferInProgEn) Bit 8 XferComplete Enable (XferCmplEn)
7-5	Reserved
4:0	Interrupt number (IntrNum) Applies to IN and OUT endpoints. Indicates interrupt/Event Buffer number on which endpoint related interrupts for this endpoint are generated. This must be set to 0.
Parameter 0	
31-30	Config Action 0 Initialize endpoint state: This action is used when an endpoint is configured the first time. It will cause the data sequence number and flow control state to be reset. DEPCMDPAR2 will be ignored. The encoding of this action is backward compatible with software that previously set "Ignore Sequence Number" to 0. 1 Restore endpoint state: This action is used when reconfiguring an endpoint with saved state after hibernation. It will cause the data sequence number and flow control state to be restored from DEPCMDPAR2. 2 Modify endpoint state: This action is used when modifying an existing endpoint configuration, such as changing the DEPEVTEN event enable bits, interrupt number, or MaxPacketSize. The data sequence number and flow control state will be unchanged, and DEPCMDPAR2 will be ignored. The encoding of this action is backward compatible with software that previously set "Ignore Sequence Number" to 1. When issuing a Endpoint Configuration command with Config Action=Restore, Parameter 2 must be filled in with the same value returned by the Get Endpoint State command prior to hibernation.
29-26	Reserved
25-22	Burst Size (BrstSiz) When field is set to- 0 Burst length = 1

Table continues on the next page...

Table 11-24. Command 1: Set endpoint configuration parameters: DEPCFG (continued)

Field	Description
	1 Burst length = 2, and so on, up to, 15 Burst length = 16 For IN transfers, this value represents the maximum length of the burst that the device attempts when the Host initiates an IN transfer with a TP_ACK. If BrstSiz < the NumP value in the initiating TP_ACK, then the device controller limits the burst length to BrstSiz. If BrstSiz >= the NumP value in the initiating TP_ACK, then the device controller attempts a burst length of NumP. For OUT transfers, this value represents the NumP value utilized in the response TP_ACK from the device controller. The NumP value indicates (to the host) the burst length the device desires. Note: In the special case of BrstSiz=0 (burst length = 1), the TP_ACK from the device controller has NumP=0; which is a flow-control condition. If this value is utilized, then the endpoint enters flow control for each DP received and a subsequent ERDY is transmitted (according to the USB Specification). However, this may have an undesirable impact on the system throughput. To avoid this impact, it is recommended to set BrstSize=1 to prevent the flow-control condition. The trade-off is that the host could potentially burst two OUT data packets to the device, resulting in an ACK for the first packet, and an NRDY for the second packet.
21-17	FIFO Number (FIFONum) Indicates which transmit FIFO is assigned to this endpoint. For control endpoints, the FIFONum value in the OUT direction must be programmed to the same value as the IN direction. This field should be set to 0 for all other OUT endpoints. Even though there may be more than 16 TXFIFOs in DRD mode, the device mode must use lower 16 TXFIFOs.
16-14	Reserved
13-3	Maximum Packet Size (MPS) Applies to IN and OUT endpoints. The application must program this field with the maximum packet size (in bytes) for the current USB endpoint. USB 3.0 supports up to 1024 bytes.
2-1	Endpoint Type (EPType) This is the transfer type supported by this USB endpoint. 00 Control 01 Isochronous 10 Bulk 11 Interrupt
0	Reserved

11.2.2.7.2 Command 2: Set endpoint transfer resource configuration (DEPXFERCFG)

There must be only one transfer resource allocated per endpoint. Start transfer causes the use of the transfer resource. End Transfer or an XferComplete event releases the transfer resource. If software attempts to allocate more transfer resources than have been configured in the hardware, this command will return an error in the CmdStatus/ EventStatus field.

NOTE

If GUSB2PHYCFG[SUSPENDUSB20] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

Table 11-25. Command 2: Set endpoint transfer resource configuration parameters: DEPXFERCFG

Field	Description
Parameter 2	
31-0	Reserved
Parameter 1	
31-0	Reserved
Parameter 0	
31-16	Reserved
15-0	Number of Transfer Resources (NumXferRes) Defines the number of Transfer Resources allocated to this endpoint. This field must be set to 1.

11.2.2.7.3 Command 3: Get endpoint state (DEPGETSTATE)

After Get Endpoint State (DEPGETSTATE) command completes, the 32 bits in DEPCMDPAR2 represent the endpoint state to be saved. This includes the current data sequence number, flow control state, and control transfer state (for control endpoints).

This command is only used when the controller enters hibernation. The controller requires the 32 bits of state information when exiting hibernation as part of the DEPCFG command with the Config Action field set to "Restore endpoint state".

NOTE

If GUSB2PHYCFG[6] is set to ‘1’, it must be set to '0' prior to issuing this command and may be set to ‘1’ after the command completes.

11.2.2.7.4 Commands 4 and 5: Set Stall and Clear Stall (DEPSSTALL, DEPCSTALL)

These commands apply to non-isochronous IN and OUT endpoints only.

The application issues a Set Stall command to stall all tokens from the USB host to this endpoint. If the endpoint is in the NRDY state, the STALL state takes priority.

If a transaction is currently in progress when the software issues Set Stall, the behavior depends on the direction of the endpoint:

- For OUT endpoints, the current packet will complete. The core will respond with STALL to the next OUT DP or token.
- For IN endpoints in SuperSpeed, the current burst transaction will complete, and the core will respond with STALL to the next ACK TP. For other speeds, the core will complete the current packet and respond STALL to the next IN token.

For non-control endpoints, the application is responsible for both setting and clearing STALL via the Set Stall/Clear Stall commands. When the application clears the STALL, the endpoint's data sequence number is reset to zero.

For control endpoints, the application issues only the Set Stall command, and only on the OUT direction of the control endpoint. The controller automatically clears the STALL when it receives a SETUP token for the endpoint.

The application must not issue the Clear Stall command on a control endpoint. If the endpoint is in flow control (NRDY) and also in the STALL state, it responds with a STALL for any packet. The only exception is that the controller always responds to SETUP data packets with an ACK handshake, independent of the STALL state.

NOTE

- If GUSB2PHYCFG[6] is '1', it must be set to '0' prior to issuing this command and may be set to ‘1’ after the command completes.
- When issuing Clear Stall command for IN endpoints in SuperSpeed mode, the software must set the "ClearPendIN" bit to '1' to clear any pending IN transcations, so that the device does not expect any ACK TP from the host for the data sent earlier.

11.2.2.7.5 Command 6: Start transfer (DEPSTRTXFER)

This command applies to IN and OUT endpoints.

A Transfer Descriptor (TD) in Device mode is a list of TRBs that comprises one or more transfers. The pointer to a TD is added to the core's cache by the Start Transfer command and removed by an XferComplete event or by the End Transfer command.

Software issues this command indicating that a descriptor is ready to be processed and DMA can start for this endpoint/stream combination. For IN endpoints, this causes the descriptor to be processed, and data is moved from the corresponding memory buffer to corresponding transmit FIFO when the transfer is started. For OUT endpoints, this causes the descriptor to be processed and data is moved from the receive FIFO to corresponding memory buffer.

In response to the Start Transfer command, the hardware assigns this transfer a resource index number (XferRscIdx) and returns the index in the DEPCMDn register and in the Command Complete event. This index must be used in subsequent Update and End Transfer commands.

It is illegal to issue a Start Transfer command for the same TD if it is already present in the core's cache.

Non-stream capable endpoints rules:

- The Stream ID field must be set to 0.
- The HighPri field is reserved.

Stream-capable endpoint rules:

- The Stream ID field must be set to non-zero and match the Stream ID passed into the Start Transfer endpoint command associated with this transfer. In all the TRBs of a transfer, the Stream ID fields must be the same.
- The HighPri field is reserved.

NOTE

- If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.
- Before issuing a start transfer command, software needs to verify whether the link is in U0. If the link is not in U0, software needs to bring the link to U0 by performing a remote wakeup. This is applicable to both SS and USB 2.0 speeds.
- When an IN transfer is in progress, (that is, started and not completed yet), and the software sees a link state change event to L1, then the software can use the GDBGFIFOSPACE register (write followed by a read) to

determine if the TXFIFO is empty or not. If the TXFIFO is not empty, it can initiate a remote wakeup.

Table 11-26. Command 6: Start transfer parameters: DEPSTRTXFER

Field	Description
Parameter 2	
31-0	Reserved
Parameter 1	
31-0	Transfer Descriptor Address (TDAddr Low) Indicates the lower 32 bits of the external memory's start address for the transfer descriptor. Because TRBs must be aligned to a 16-byte boundary, the lower 4 bits of this address must be 0.
Parameter 0	
31-0	Transfer Descriptor Address (TDAddr High) Indicates the higher 32 bits of the external memory's start address for the transfer descriptor.

11.2.2.7.6 Command 7: Update transfer (DEPUPDXFER)

If software uses circular TRB buffers and updates a TRB, whose Hardware Owner (HWO) bit was 0, by setting HWO=1, it must execute the Update Transfer command, specifying the transfer resource index of the TRB in the DEPCMD register. The hardware uses this information to re-cache the TRB.

Software may issue a special “No Response Update Transfer” command by setting CmdAct=0 and CmdIOC=0. In this case, the hardware does not generate a Command Complete event, does not set the CmdAct bit to '0' (because it will be '0'), and software may immediately issue another command to the same endpoint following this one. This special type of Update Transfer may not be used when software depends on the XferNotReady event to setup TRBs (see “On Demand” transfers in Transfer setup recommendations).

Software may issue an Update Transfer command for a transfer resource that has already completed (either due to XferComplete event or an End Transfer command), and the core will detect that the Update Transfer is unnecessary. However, software must not issue an Update Transfer command for a transfer resource index that has never been started.

NOTE

If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

11.2.2.7.7 Command 8: End transfer (DEPENDXFER)

This command applies to IN and OUT endpoints. Software issues this command requesting DMA to stop for the endpoint/stream specifying the transfer resource index of the TRB and the ForceRM parameter to be set to 1 in the DEPCMD register.

When issuing an End Transfer command, software must set the CmdIOC bit (field 8) so that an Endpoint Command Complete event is generated after the transfer ends. This is necessary to synchronize the conclusion of system bus traffic before the End Transfer command is completed.

For IN endpoints, this command causes descriptor processing to stop and the core stops fetching new data for the endpoint and stops transfers on the USB. The core may truncate a packet being transmitted with the DPPABORT ordered set, does not wait for any pending ACKs from the USB, and does not update the TRB status. For OUT endpoints, this command causes descriptor processing to stop. If there is currently data, it is moved from the receive FIFO to corresponding memory buffer and completed at packet boundary. The core does not update the TRB status.

Use this command under the following conditions:

- When handling USBReset or SetConfiguration, endpoints are closed using this command.
- After receiving a ClearFeature (STALL) control transfer, software issues End Transfer followed by Clear Stall, followed by Start Transfer.
- After an XferInProgress event when the TRB after the one that caused the XferInProgress event has its HWO bit set to '0', this command can be used.
- For isochronous endpoints, if the host stops moving data for many intervals, software may force the end of the transfer and wait for the host to restart.

The hardware does not issue a XferComplete event on End Transfer, but only issues a CommandComplete event.

NOTE

If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

Table 11-27. Command 8: End Transfer Parameters: DEPENDXFER

Field	Description
Parameter 2	
31-0	Reserved
Parameter 1	
31-0	Reserved

Table continues on the next page...

Table 11-27. Command 8: End Transfer Parameters: DEPENDXFER (continued)

Field	Description
Parameter 0	
31-0	Reserved

End Transfer Command With Hibernation Enabled The following applies only when hibernation is enabled. Prior to hibernation, software will set ForceRM to '0' when it issues End Transfer commands for active transfers. This command may cause a TRB to be written back with intermediate state.When software issues an End Transfer command with the ForceRM bit set to 0, and the core has at least one valid current TRB, it will perform the following:

- Write back the TRB updating the Buffer Pointer and Buffer Size fields to reflect the current state of the TRB.
- Write the value '4' (TRBInProgress) into the TRBSTS field to indicate that this TRB may have been partially used and is still active. This is an indication to software that the original buffer pointer field may have been changed by hardware.
- Set the HWO bit to '0' unless it is a Link TRB. The HWO bit in a Link TRB will be unchanged.

After the state is restored, software may restart a transfer from a partially completed TRB by setting TRBSTS to '0', setting the HWO bit back to '1', and issuing the Start Transfer command.Software is not required to perform any special manipulation on partially completed TRBs. When the core completes a restarted partially complete TRB, it will set TRBSTS back to '0'.

11.2.2.7.8 Command 9: Start new configuration (DEPSTARTCFG)

Software issues this command under the following conditions:

- After power-on-reset with XferRscIdx=0 before starting to configure Endpoints 0 and 1. CmdIOC must be set to '0' and software must poll the CmdAct bit to determine when the command is complete because Endpoint 0 is not yet configured with a valid interrupt number.
- With XferRscIdx=2 when it receives SetConfiguration before starting to configure Endpoints > 1. CmdIOC may be set to '0' or '1'.

This command should always be issued to Endpoint 0 (DEPCMD0).

Hardware resets the transfer resource allocation to the value in the XferRscIdx parameter (must be 0 or 2) upon receiving this command.

NOTE

If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

11.2.2.8 Power management overview

The following power management features are available in the USB 3.0 core:

- Hibernation (save/restore) in device (L1, L2, U3, Disconnected) and host (U3, Disconnected)
- Hardware-controlled LPM in USB 2.0 host
- Clock gating in device (U1/U2/U3) and host (U1/U2/U3)

When hibernation is supported, the core saves some of its internal state to an external location and allows the USB3 core to be powered off. Small modules, namely the PMUs, detect wakeup conditions and request power be reapplied to the core, at which time the external state is brought back into the core.

Since the PMU modules are not inside the USB 3.0 core, the PMUs can easily be placed anywhere in the chip during floor-planning, adding isolation cell/voltage shifter cells. The PMU modules can be placed in the always ON power domain and the power domain is 0.8 V which could be operating at a different voltage than the USB 3.0 power domain.

Hibernation is described in the xHCI Specification as an optional feature that can be supported.

In addition, this feature is also added to the device controller.

NOTE

- The PMU does not support the following features in Hibernation mode:
  - BC
- It is required that the SuperSpeed PHY must support receiver detection in P3 PowerDown mode.
- In the host mode, the USB 3.0 core does not support clock gating and hibernation when the Debug Capability is enabled during runtime (DCCTRL.DCE=1).
- During the normal host mode of operation, the USB 3.0 core supports clock gating and hibernation features when the Debug Capability is not enabled by the software (DCCTRL.DCE=0).

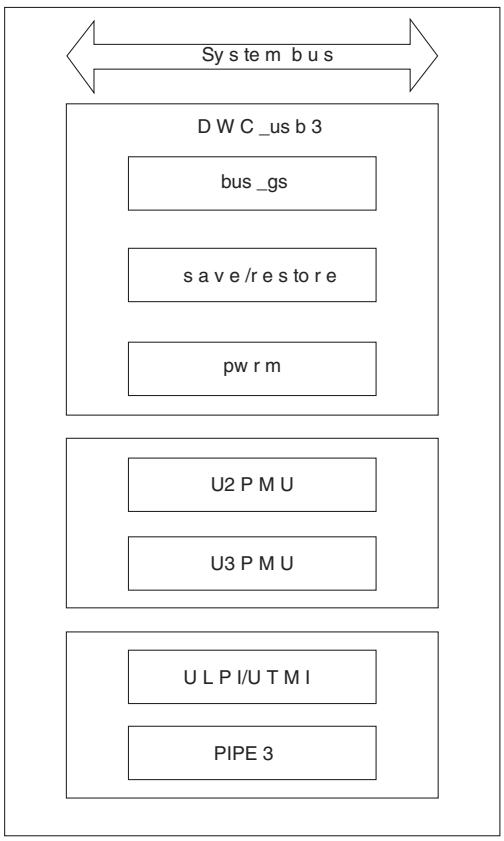


Figure 11-26. Power management hibernation architecture

The second power saving mechanism is hardware-controlled LPM. This is a USB 2.0 host-only feature where the host controller automatically detects the downstream port is idle for a certain amount of time and autonomously initiates an LPM token to the connected device. If the device accepts, the link is put into the standard USB 2.0 Suspend state.

Finally, clock gating is also supported in the host controller as well as in the device controller. The following table gives the list of power saving mechanisms available in USB 3.0 core.

Table 11-28. Power savings support

USB 3.0 State	USB 2.0 State	Device Power Savings	Host Power Savings
U1 (Hardware Initiated)	None	Core: Clock Gating PHY: P1	Core: Clock Gating PHY: P1
U2 (Hardware Initiated)	LPM-L1 (Hardware Initiated)	Core: Clock Gating, LPM-L1 Hibernation PHY: P2/Sleep	Core: Clock Gating, PHY: P2/ Sleep
U3 (Software Initiated)	Suspend (Software Initiated)	Core: Clock Gating, Hibernation PHY: P3/Suspend	Core: Clock Gating, Hibernation PHY: P3/Suspend

Table continues on the next page...

Table 11-28. Power savings support (continued)

USB 3.0 State	USB 2.0 State	Device Power Savings	Host Power Savings
Rx.Detect (Software Initiated)	None	None	Core: Clock Gating, Hibernation PHY: P3/Suspend
SS.Disabled (Software Initiated)	None	Core: Clock Gating, Hibernation PHY: P3/Suspend	None

11.2.2.8.1 Clock gating

The core implements clock gating in the following situations:

- In USB 2.0 device mode
  - When the UTMI suspend or l1_suspend signal is asserted and the core is idle
- In USB 3.0 device mode, when the link is in U1, U2 or U3 state and the core is idle.

In host mode, clock gating is enabled in the following situations:

- When all USB 2.0 ports are in the suspend or L1 Suspend state, and all USB 3.0 ports are in the U1/U2/U3 state, and the core is idle.

Internal clock gating will apply to:

- Modules that use the ram_clk
- Some modules that use bus_clk

Internal clock gating will not apply to:

- Modules that use mac3_clk: When the USB 3.0 PHY is suspended, these modules switch to using suspend_clk, so their power consumption is reduced but the clock is not completely stopped.
- Modules that use mac2_clk: When the USB 2.0 PHY is suspended, these modules switch to using suspend_clk, so their power consumption is reduced but the clock is not completely stopped.
- The bus_gs module, which uses bus_clk. This module detects wakeup on the slave interface and thus needs a non-gated clock.

Clock gating can be enabled or disabled using the GCTL register. The GCTL register is "sticky," it retains its value across soft resets and hibernation.

11.2.2.8.2 Hardware-Controlled LPM

In USB 2.0, the main link power management mechanism was for the host to suspend the link. However, suspending the link requires a long period of idle and was wasteful of power, especially because the host knows ahead of time whether it is going to schedule any traffic. It also required a long period of resume signaling.

The USB 2.0 Link Power Management Addendum [LPM-ECN] added a new token to the USB 2.0 specification, the LPM token. The host has the option of suspending the link quickly instead of waiting for a long period of idle, and communicates its desire to the device by sending an LPM token. The ECN also introduced a terminology for different link states. This new link state was called L1. The traditional USB 2.0 suspend was called L2. The L1 state requires a much shorter resume signaling period too.

An xHCI driver can initiate LPM by writing '2' to the PORTSC.PLS field. However, there is a lot of overhead for an xHCI driver to be constantly trying to keep the link in a low power state. Because of this, the xHCI specification also allows a compatible xHC to be enabled with "Hardware-Controlled LPM" by setting PORTPMSC.HLE as 1. When this happens, the xHC is expected to automatically send LPM tokens when it has no pending transfers and thinks that the link is idle enough to enter L1. This feature is supported by the host core.

11.2.2.8.3 USB PHY power gating

The USB PHY power gating is used to save the leakage power of PHY if it is not required. Follow the steps given below to take PHY into power gating state:

1. Assert PHY_RESET by writing 1'b0 to the PHY_CTRL1[PHY_RESET] bit.
2. Set the PHY_CTRL1[PHY_TEST_POWERDOWN_HSP] and PHY_CTRL1[PHY_TEST_POWERDOWN_SSP] to 2'b11.

NOTE

Both the above steps must have separate writes.

11.2.2.9 Clock domains

The DWC_usb3 has the following clock domains which describes clock usage during SS Only or SS and HS (Host mode) operation:

- USB 3.0 PIPE3 PHY clock
  - This clock is 125 MHz with 32-bit PHY, 250 MHz with 16-bit PHY, or 500 MHz with 8-bit PHY.

  - Separate Rx and Tx clock input pins are provided to make external PHY integration easier. The Tx clock can be an early version of the PHY clock to compensate for output PAD delays.
  - During the lowest SS PHY power-down state (P3), the PHY does not drive this signal; therefore, a suspend clock signal is provided in its place.
- USB 2.0 PHY/MAC clock

  This clock is 30 or 60 MHz in UTMI+ PHY mode, or 60 MHz in ULPI mode

- MAC3 clock (125 MHz MAC3 clock, which is divided PIPE3 clock)

  The MAC3 clock is a divided version of the PIPE3 PHY clock. In Device or single-port Host mode, the MAC3 clock is synchronous to the PIPE3 clock, but may be delayed to tolerate the larger MAC3 clock insertion delay. In multi-port Host and Hub mode, the MAC3 clock and the PIPE3 clock may be asynchronous.

- Bus clock
  The bus clock is system-dependent. The SoC bus bandwidth must be at least the same as USB 3.0 bandwidth. For example, the bus clock needs to be at least 125 MHz with a 64-bit SoC bus to satisfy the device duplex performance requirement of 8 Gbps (125 MHz * 64 = 8 Gbps). For example, in the case of a four-port concurrent host, the bus clock must be at least 250 MHz at 128 bitd to meet 32 Gbps (250 MHz * 128 = 32 Gbps).

NOTE

A system bus interface can have different master and slave types, but they must use the same bus clock.

- RAM clock to meet both MAC and BUS domain bandwidth

  The RAM needs to support twice the USB 3.0 bandwidth: one to service the USB side and the other to service the bus side. In the case of a single-port RAM, the recommended clock speed is at least 250 MHz. In the case of 2-Port mode, it needs to be at least 125 MHz. This is typically connected to PIPE3 or MAC3 clock. If your bus clock is faster than the PIPE3/MAC3 clock, you can connect the bus clock to the RAM.

- Suspend Clock

  During suspend, in order to achieve low power, the PIPE3/MAC3/MAC2/MAC clock is switched to a lower frequency variable clock.

The following clock conditions apply during High Speed (High Speed device connection or only High Speed Devices in Host mode)

- The USB 3.0 PIPE3 PHY clock is gated/shut-down.

- The USB 2.0 PHY/MAC clock is 30 or 60 MHz in UTMI+ PHY mode, or 60 MHz in ULPI mode.
- The MAC3 clock, which is a divided PIPE3 clock, is gated/shut-down.
- The bus clock is system dependent and should be at least 60 MHz to meet a five-clock USB 2.0 turnaround time.
- The RAM clock, to meet both MAC and BUS domain bandwidth, must support twice the USB 2.0 bandwidth; one to service the USB side and one to service the bus side. Minimum 60 MHz; typically, the bus clock is connected to RAM.

11.2.2.10 Reset generation

The following resets can be controlled by registers (GUSB2PHYCFG, GUSB3PIPECTL, GCTL, and USBCMD), and the signal vcc_reset_n:

- USB 2.0 PHY reset
- USB 3.0 PHY reset
- Controller internal reset
- Light reset

The following figure shows the Reset Generation unit.

NOTE

When Hibernation is enabled, PHY reset behavior is enabled only when both vcc_reset_n and vaux_reset_n are asserted.

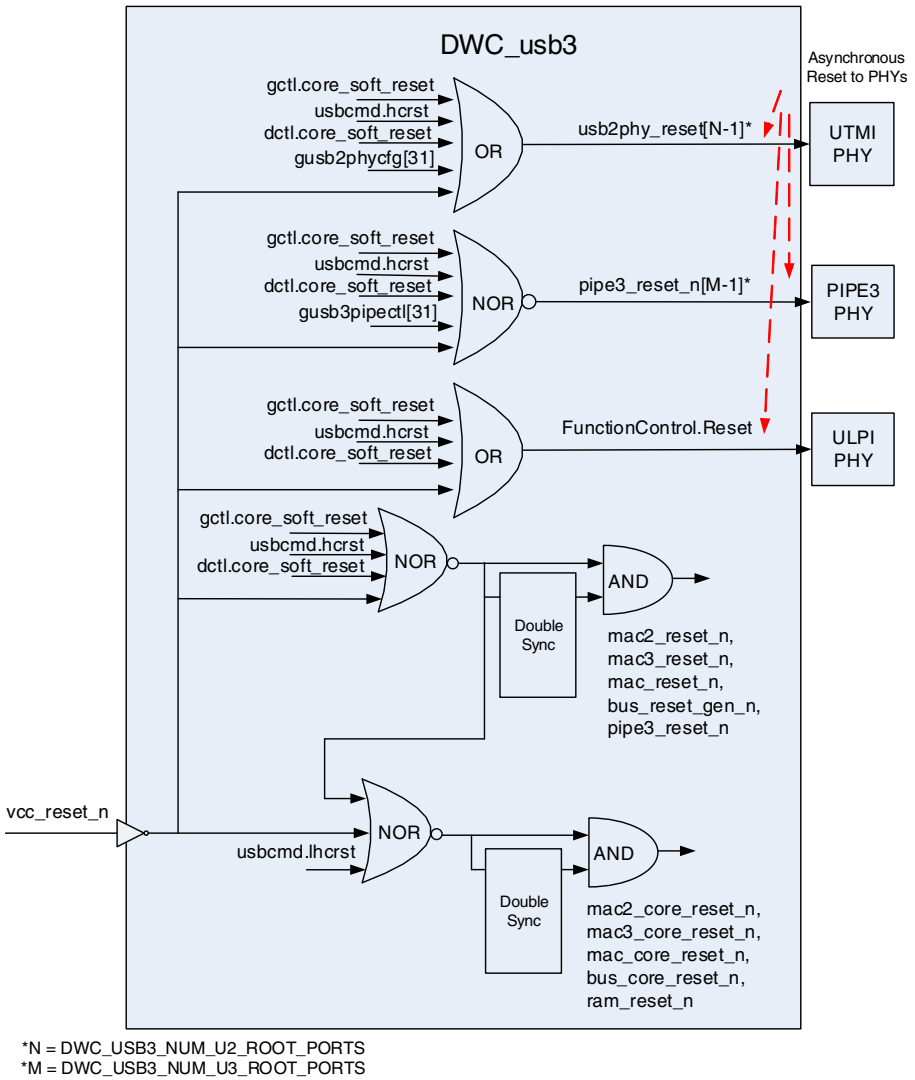


Figure 11-27. Reset Generation

Because there are multiple clock domains in the design, and toggle interfaces are used between clock domain, reset generation requires the following conditions to be met:

- All internal resets are active simultaneously for some period to initialize all the flops. The reset-module (DWC_usb3_rst.v) of the controller takes care of this by asynchronously propagating vcc_reset_n to all the internal resets in the design.
- The external USB 3.0 and USB 2.0 PHY reset requirements must be met. The controller provides pipe3_reset_n and usb2phy_reset signals to achieve these requirements.

11.2.2.10.1 Asynchronous reset

In asynchronous reset mode, the controller generates the PHY resets. You must ensure that vcc_reset_n is active to meet the PHY reset timing. Asynchronous reset mode requires the following conditions to be met:

- Connect pipe3_reset_n to USB 3.0 PHY.
- Connect usb2phy_reset to USB 2.0 PHY.
- Assert the vcc_reset_n signal for a minimum period necessary to reset the USB 3.0 and USB 2.0 PHYs.

Sequence of Asynchronous Resets

Asynchronous resets must be generated in the following sequence:

1. Drive vcc_reset_n for at least a minimum period required to reset USB 3.0 and USB 2.0 PHYs.
2. The controller drives the pipe3_reset_n and usb2phy_reset outputs active as long as vcc_reset_ n input is active. In addition, the controller also drives all the internal resets in the controller.
3. When vcc_reset_n is de-asserted, the controller also de-asserts pipe3_reset_n and usb2- phy_reset outputs. In addition, the controller also removes the internal bus and RAM clock domain resets. However, the controller keeps the internal MAC3 clock domain reset.
4. When pipe3_reset_n is de-asserted, the USB 3.0 PHY drives pipe3_PhyStatus to “0” once the PHY is reset and the pipe3_pclk clock is stable. The controller waits until pipe3_PhyStatus turns “0” before removing the internal reset to the PHY and MAC domains (mac3_reset_n). This ensures that the controller starts Rx detection only after the PHY clock is stable.
5. In the case of the USB 2.0 PHY, after usb2phy_reset is de-asserted, the USB 2.0 PHY drives a stable valid clock based on the PHY specification.

In addition to power-on reset, you can use the Software reset feature. Following figure shows the asynchronous reset and PHY clock sequencing and requirements.

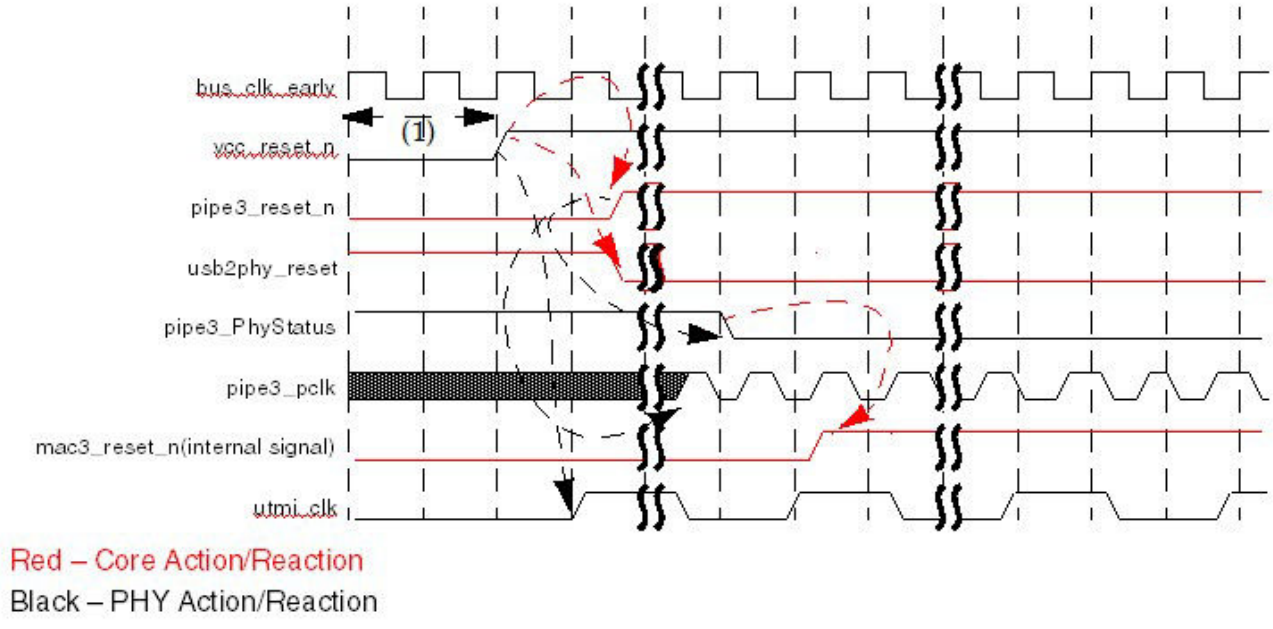


Figure 11-28. DWC_usb3 clock domains and data flow

11.2.2.10.2 Software reset

Software reset is an optional feature that allows the software to reset the controller and the PHY whenever it is required. For example, while developing drivers you can use this feature to reset the controller and PHY before you load modules into the driver. This ensures that you do not have to do a hardware reset of your development platform if the driver stops functioning, thereby saving some time.

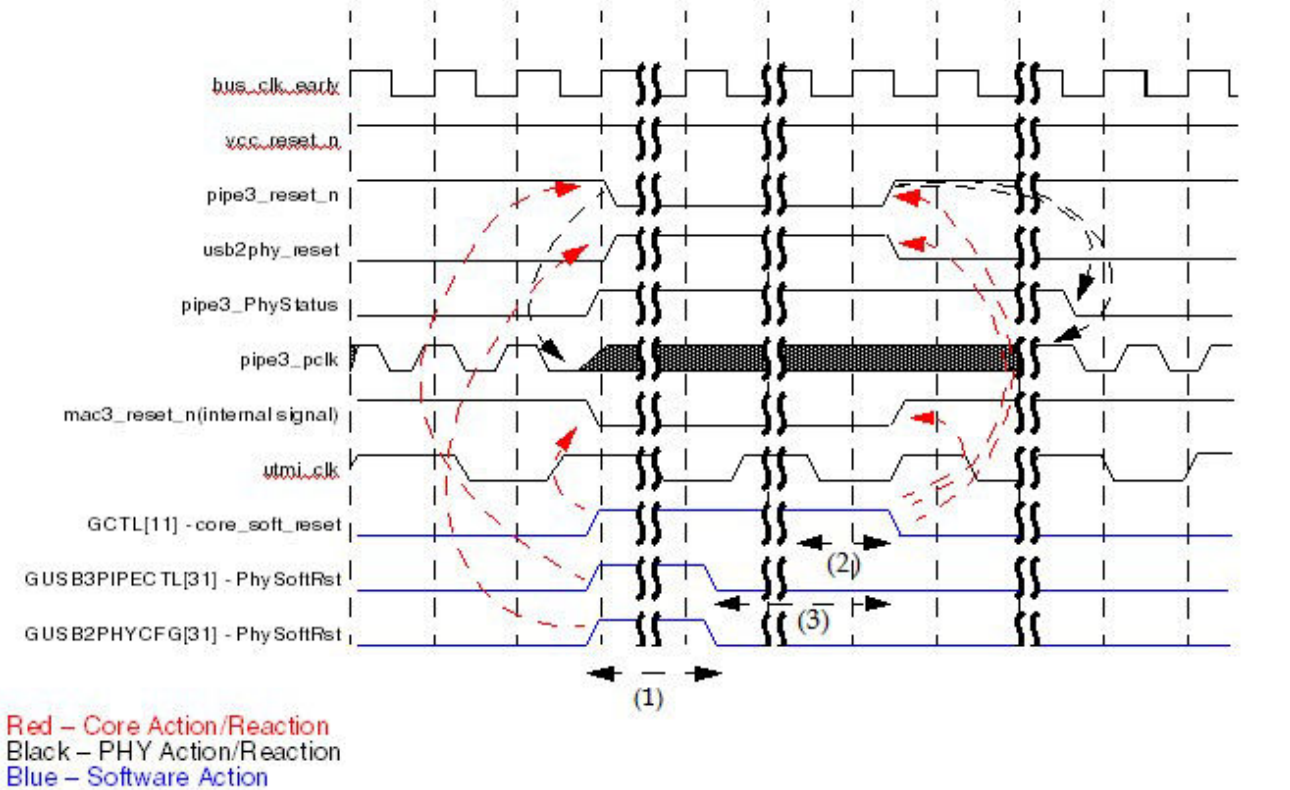
NOTE

The PHYs can be reset only if you are using the pipe3_reset_n and usb2phy_reset signals from the controller.

The GUSB2PHYCFG, GUSB3PIPECTL, and GCTL registers control the USB 2.0 PHY reset, USB 3.0 PHY reset, and internal reset of the controller. Software resets must be generated in the following sequence:

1. Set GUSB2PHYCFG[31], GUSB3PIPECTL[31], and GCTL[11] to reset the PHYs and keep the controller in a reset state.
2. Reset GUSB2PHYCFG[31] and GUSB3PIPECTL[31] after they meet PHY reset duration. This removes the reset to the PHYs.
3. Wait for the PHY clock to stabilize and reset GCTL[11] bit. This ensures that the reset to all the internal blocks are asserted when all the clocks are stable.

The following figure shows the software reset and PHY clock sequencing and requirements.



(1) – PhySoftRst assertion timing must meet the minimum USB 3.0 and USB 2.0 PHY reset timing.
(2) – The core_soft_reset signal must be de-asserted only after at least 16 clock cycles of the slowest clock; if the suspend_clk is the slowest clock, then use the suspend_clk.
(3) – The core_soft_reset de-assertion after the PhySoftRst de-assertion must be at least after 16 clocks of the slowest clock; if the suspend_clk is the slowest clock, then use the suspend_clk

Figure 11-29. Software resets and PHY clock sequencing and requirements

11.2.2.11 Interrupts

NOTE

This section is for applicable only for non-SSIC device configurations.

The DWC_usb3 controller generates the interrupt[(DWC_USB3_NUM_INT-1):0] signal to indicate the occurrence of an event needing application intervention.

While configuring the controller using coreConsultant, you can select the number of Event Buffers by choosing the “Number of Device Mode Event Buffers” (DWC_USB3_NUM_INT). Each Event Buffer is associated with one interrupt line.

Software allocates memory for the Event Buffers, to receive interrupts, using the GEVNTADR and GEVNTSIZ registers. The GEVNTCOUNT register holds the number of valid bytes in the Event Buffer.

When an event occurs within the controller, the hardware checks the enable bit that corresponds with the event to determine if the event needs to be generated. If it is enabled, the controller writes the event to the Event Buffer, increments the GEVNTCOUNT[15:0] (Event Count) register field in the GEVNTCOUNT register, and generates the hardware interrupt (if not masked and is the correct time to assert the interrupt based on the DEV_IMOD register).

For information on the structure of the Event Buffers, see Device programming model. By reading the Event Buffer, software can decode the type of event that occurred. For information on how to use multiple interrupts, see Multiple device interrupt support.

The DWC_usb3 controller generates following types of interrupt events:

- Command interrupt event
- Device-specific interrupt event
- Device physical endpoint-specific interrupt event
- Interrupt based on TRB settings
- Device interrupt (GSTS[6])
- Battery charger interrupt event

11.2.2.11.1 Command interrupt event

Command interrupt event are generated based on completion of device generic commands or endpoint-specific commands.

Based on whether the command is issued to the generic device or a specific endpoint, there are two types of command interrupt events:

- Device generic interrupt events

  If you set DGCMD[8] (CMDIOC) register bit while issuing a device generic command, you get Generic Command Complete event for that command.

  In the Event Buffer, this event shows up with DEVT[11:8] bits set to 10 (see Table 11-11).

- Device endpoint-specific interrupt events

  If you set DEPCMD[8] (CMDIOC) register bit while issuing a device endpoint-specific command, you get Endpoint Command Complete event for that command.

  In the Event Buffer, this event shows up with DEPEVT[9:6] bits set to 7 (see Table 11-10).

11.2.2.11.2 Device-specific interrupt event

You can use DEVTEN register fields to enable/disable the device-specific events. (See the “DEVTEN” section of the Registers chapter).

If the last bit of the event in the Event Buffer is 1, it indicates a device-specific event. Bits [11:8] indicate the type of device-specific event that occurred. For different types of device-specific events, see bits [11:8] in Table 11-11.

11.2.2.11.3 Device physical endpoint-specific interrupt event

Unlike the case of device-specific commands, for device endpoint-specific commands, you need to issue a DEPCFG command with field [13:8] (DEPEVTEN) set to enable/disable the device endpoint-specific events (Refer bits [13:8] in Table 11-24).

If the last bit of the event in the Event Buffer is 0, it indicates a device endpoint-specific event. Bits [9:6] indicate the type of endpoint-specific event that occurred. For the different types of device endpoint-specific events, see Table 11-10.

11.2.2.11.4 Interrupt based on TRB settings

Based on the TRB control fields (DW 0F-0C), the DWC_usb3 controller generates events, and writes them to the Event Buffer. Note that you must have enabled the required events using the DEPCFG command prior to this step.

For example, when you set the Interrupt on Complete (IOC) bit in a TRB, once the transfer indicated by that TRB (buffer) is completed, the controller issues XferInProgress event with IOC bit set in the event status field of the DEPEVT event. This indicates the buffer is available for software to reuse or release. This helps in freeing up the buffer sooner than waiting for the entire transfer to complete. Note that you must have set XferInProgEn to 1 using DEPCFG command for that specific endpoint for XferInProgress event to occur.

11.2.2.11.5 Device interrupt (GSTS[6])

The GSTS[6] (Device_IP) field indicates that there is a pending interrupt pertaining to the device operation in the Device event queue. This can be used to know if any interrupt is pending before halting the controller.

11.2.2.11.6 Battery charger interrupt event

Battery charger events are captured in the BCEVT register if the corresponding BC event enable bits in the BCEVTEN register is set accordingly.

The GSTS[9] (BC_IP) register bit indicates that there is a pending interrupt pertaining to BC in the BCEVT register.

The DWC_usb3 controller also drives the bc_interrupt signal to indicate a BC event has occurred and it needs application intervention. This signal is driven high as long as any event is set in BCEVT register. Once the application services the interrupt/event, it can clear it by writing a 1 to the corresponding event bit in the BCEVT register.

NOTE

Unlike other interrupt events, all battery charger-related events (those captured in the BCEVT register) are not mapped to the "interrupt" output signal. They are mapped to a separate interrupt line called "bc_interrupt".

11.2.2.11.7 USB core interrupts

Each USB core uses one dedicated vector in the Interrupt Table. The vector numbers associated with each of the cores can be found in the Interrupt section.

With the exception of the wake-up interrupts, all of the interrupt sources are controlled in the USB Cores.

11.2.2.11.8 Wake-up interrupts

Each USB has an associated wake-up interrupt. The wake-up interrupts are generated outside of the USB controller cores, and using dedicated vector. These interrupts are generated by the Power Control blocks which run on the 32 kHz standby clock. The wake-up interrupt is designed to work even when the USB and Arm platform clocks are disabled, such that a wake-up condition on the USB bus can re-activate the Arm platform clocks. Because the wake-up interrupt is generated and cleared on a 32 kHz clock, this interrupt request responds very slowly to clear actions. For this reason, the software must disable the wake-up interrupt to clear the request flag. Disabling the interrupt masks the request instantaneously as this is clocked by the Arm platform clock. The software should wait for at least three 32 kHz clock cycles before re-enabling this interrupt to allow sufficient time for the request flag to clear. Because this interrupt is only used during low power modes of the USB, it is sufficient to enable the wake-up interrupt just prior to entering the USB suspend mode.

11.2.3 External signals

This section contains detailed descriptions of all the USB 3.0 controller signals. Many of the signals for the PHY interfaces are muxed onto the same pins to reduce pin count. The following table shows the USB signals, indicating which interface supports each signal.

Table 11-29. USB 3.0 External signals

Signal	I/O	Description
USB1_D_P	IO	USB PHY Data Plus
USB1_D_M	IO	USB PHY Data Minus
USB1_VBUS	IO	USB PHY VBUS
USB1_TX_P	O	USB PHY 3.0 Transmit Data (positive)
USB1_TX_M	O	USB PHY 3.0 Transmit Data (negative)
USB1_RX_P	I	USB PHY 3.0 Receive Data (positive)
USB1_RX_M	I	USB PHY 3.0 Receive Data (negative)
USB1_RESREF	IO	USB PHY Impedance Calibration
USB1_DRVVBUS	O	VBus power enable.
USB1_PWRFAULT	I	Indicates that a Vbus fault has occurred.

11.2.4 Initialization/application information

The PHY can be configured for fixed equalization by programming relevant control registers in the USB 3.0 PHY.

In order to initialize the USB 3.0 PHY, software should perform the following steps:

1. Write 1'b0 to RX_OVRD_IN_HI.RX_EQ_EN [address 16'h1006: bit 6].
2. Write 1'b1 to RX_OVRD_IN_HI.RX_EQ_EN_OVRD [address 16'h1006: bit 7].
3. Write a fixed value to RX_OVRD_IN_HI.RX_EQ [address 16'h1006: bits 10–8] [equalization setting] (generally 2–4, based on testing in customer environment).
4. Write 1'b1 to RX_OVRD_IN_HI.RX_EQ_OVRD [address 16'h1006: bit 11].

11.2.5 Memory map/register definition

This section describes the registers and data structures in the USB3 module.

NOTE

For USB Wakeup Registers see HSIO BLK_CTRL register description.

11.2.5.1 USB3 register descriptions

USB3.0 registers are 32-bit wide, and the addresses are 32-bit block aligned. To avoid hardware or software incompatibility, the driver must access the registers as 32-bit units. The driver must not access these registers as 8 bit or 16 bit.

11.2.5.1.1 DWC_usb3 memory map

USB1 base address: 3810_0000h

USB2 base address: 3820_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Capability Registers Length and Host Controller Operational Registers (CAPLENGTH)	32	RO	Table 11-29
4h	Structural Parameters 1 Register (HCSPARAMS1)	32	RO	Table 11-29
8h	Structural Parameters 2 Register (HCSPARAMS2)	32	RO	Table 11-29
Ch	Structural Parameters 3 Register (HCSPARAMS3)	32	RO	Table 11-29
10h	Capability Parameters 1 Register (HCCPARAMS1)	32	RO	0220_FE6Ch
14h	Doorbell Offset Register (DBOFF)	32	RO	Table 11-29
18h	Runtime Register Space Offset Register (RTSOFF)	32	RO	Table 11-29
1Ch	Host Controller Capability Parameters 2 (HCCPARAMS2)	32	RO	0000_002Fh
20h	USB Command Register (USBCMD)	32	RW	Table 11-29
24h	USB Status Register (USBSTS)	32	RW	Table 11-29
28h	Page Size Register (PAGESIZE)	32	RO	Table 11-29
34h	Device Notification Register (DNCTRL)	32	RW	0000_0000h
38h	Command Ring Control Register - Low (CRCR_LO)	32	RW	Table 11-29
3Ch	Command Ring Control Register - High (CRCR_HI)	32	RW	0000_0000h
50h	Device Context Base Address Array Pointer Register - Low (DCBAAP_LO)	32	RW	0000_0000h
54h	Device Context Base Address Array Pointer Register -High (DCBAAP_HI)	32	RW	0000_0000h
58h	Configuration Register (CONFIG)	32	RW	Table 11-29
100h	USB1 Wakeup Register (USB1_WAKEUP_CTRL)	32	RW	0000_0000h
104h	USB1 Wakeup Status Register (USB1_WAKEUP_STATUS)	32	RO	0000_1000h
108h	USB2 Wakeup Register (USB2_WAKEUP_CTRL)	32	RW	0000_0000h
10Ch	USB2 Wakeup Status Register (USB2_WAKEUP_STATUS)	32	RO	0000_1000h
420h	USB2 Port Status and Control Register (PORTSC_20)	32	RW	Table 11-29
424h	USB2 Port Power Management Status and Control Register (PORTPMSC_20)	32	RW	Table 11-29
428h	USB2 Port Link Information Register (PORTLI_20)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
42Ch	USB2 Port Hardware LPM Control Register (PORTHLPMC_20)	32	RW	Table 11-29
430h	USB3 Port Status and Control Register (PORTSC_30)	32	RW	Table 11-29
434h	USB3 Port Power Management Status and Control Register (PORTPMSC_30)	32	RW	Table 11-29
438h	USB3 Port Link Information Register (PORTLI_30)	32	RO	Table 11-29
43Ch	USB3 Port Hardware LPM Control Register (PORTHLPMC_30)	32	RO	0000_0000h
440h	Microframe Index Register (MFINDEX)	32	RO	Table 11-29
460h	Interrupter Management Register (IMAN)	32	RW	Table 11-29
464h	Interrupter Moderation Register (IMOD)	32	RW	0000_0FA0h
468h	Event Ring Segment Table Size Register (ERSTSZ)	32	RW	Table 11-29
470h	Event Ring Segment Table Base Address Register - Low (ERSTBA_LO)	32	RW	0000_0000h
474h	Event Ring Segment Table Base Address Register -Low (ERSTBA_HI)	32	RW	0000_0000h
478h	Event Ring Dequeue Register - Low (ERDP_LO)	32	RW	0000_0000h
47Ch	Event Ring Dequeue Register - Low (ERDP_HI)	32	RW	0000_0000h
480h	Doorbell Register (DB)	32	RW	Table 11-29
880h	USB Legacy Support Capability Register (USBLEGSUP)	32	RW	Table 11-29
884h	USB Legacy Support Control/Status Register (USBLEGCTLSTS)	32	RW	Table 11-29
890h	USB2 xHCI Supported Protocol Capability Data Word 0 Register (SUPTPRT2_DW0)	32	RO	0200_0402h
894h	USB2 xHCI Supported Protocol Capability Data Word 1 Register (SUPTPRT2_DW1)	32	RO	2042_5355h
898h	USB2 xHCI Supported Protocol Capability Data Word 2 Register (SUPTPRT2_DW2)	32	RO	Table 11-29
89Ch	USB2 xHCI Supported Protocol Capability Data Word 3 Register (SUPTPRT2_DW3)	32	RO	Table 11-29
8A0h	USB3 xHCI Supported Protocol Capability Data Word 0 Register (SUPTPRT3_DW0)	32	RO	0300_0402h
8A4h	USB3 xHCI Supported Protocol Capability Data Word 1 Register (SUPTPRT3_DW1)	32	RO	2042_5355h
8A8h	USB3 xHCI Supported Protocol Capability Data Word 2 Register (SUPTPRT3_DW2)	32	RO	0000_0102h
8ACh	USB3 xHCI Supported Protocol Capability Data Word 3 Register (SUPTPRT3_DW3)	32	RO	Table 11-29
C100h	Global SoC Bus Configuration Register 0 (GSBUSCFG0)	32	RW	Table 11-29
C104h	Global SoC Bus Configuration Register 1 (GSBUSCFG1)	32	RW	Table 11-29
C108h	Global Tx Threshold Control Register (GTXTHRCFG)	32	RW	Table 11-29
C10Ch	Global Rx Threshold Control Register (GRXTHRCFG)	32	RW	Table 11-29
C110h	Global Core Control Register (GCTL)	32	RW	30C1_3004h
C118h	Global Status Register (GSTS)	32	RW	Table 11-29
C11Ch	Global User Control Register 1 (GUCTL1)	32	RW	Table 11-29
C128h	Global User ID Register (GUID)	32	RW	A000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
C12Ch	Global User Control Register (GUCTL)	32	RW	Table 11-29
C130h	Gobal SoC Bus Error Address Register - Low (GBUSERRADDRLO)	32	RO	0000_0000h
C134h	Gobal SoC Bus Error Address Register - High (GBUSERRADDRHI)	32	RO	0000_0000h
C138h	Global SS Port to Bus Instance Mapping Register - Low (GPRTBIMAPLO)	32	RW	0000_0000h
C13Ch	Global SS Port to Bus Instance Mapping Register - High (GPRTBIMAPHI)	32	RW	Table 11-29
C140h	Global Hardware Parameters Register 0 (GHWPARAMS0)	32	RO	2020_400Ah
C144h	Global Hardware Parameters Register 1 (GHWPARAMS1)	32	RO	8160_C93Bh
C148h	Global Hardware Parameters Register 2 (GHWPARAMS2)	32	RO	A000_0000h
C14Ch	Global Hardware Parameters Register 3 (GHWPARAMS3)	32	RO	0821_0085h
C150h	Global Hardware Parameters Register 4 (GHWPARAMS4)	32	RO	4782_2004h
C154h	Global Hardware Parameters Register 5 (GHWPARAMS5)	32	RO	0420_4108h
C158h	Global Hardware Parameters Register 6 (GHWPARAMS6)	32	RO	0A58_DC20h
C15Ch	Global Hardware Parameters Register 7 (GHWPARAMS7)	32	RO	0608_11ECh
C180h	Global High-Speed Port to Bus Instance Mapping Register - Low (GPRTBIMAP_HSLO)	32	RW	0000_0000h
C184h	Global High-Speed Port to Bus Instance Mapping Register - High (GPRTBIMAP_HSHI)	32	RW	Table 11-29
C188h	Global Full-Speed Port to Bus Instance Mapping Register - Low (GPRTBIMAP_FSLO)	32	RW	0000_0000h
C18Ch	Global Full-Speed Port to Bus Instance Mapping Register - High (GPRTBIMAP_FSHI)	32	RW	Table 11-29
C194h	Global Error Injection 1 Control Register (GERRINJCTL_1)	32	RW	0000_0000h
C198h	Global Error Injection 2 Control Register (GERRINJCTL_2)	32	RW	0000_0000h
C19Ch	Global User Control Register 2 (GUCTL2)	32	RW	Table 11-29
C200h	Global USB2 PHY Configuration Register (GUSB2PHYCFG)	32	RW	Table 11-29
C280h	Global USB 2.0 ULPI PHY Vendor Control Register (GUSB2PHYACC_ULPI)	32	RO	Table 11-29
C2C0h	Global USB 3.0 PIPE Control Register (GUSB3PIPECTL)	32	RW	010C_0002h
C300h - C31Ch	Global Transmit FIFO Size Register (GTXFIFOSIZ0 - GTXFIFOSIZ7)	32	RW	0000_0042h
C380h - C388h	Global Receive FIFO Size Register (GRXFIFOSIZ0 - GRXFIFOSIZ2)	32	RW	0000_0285h
C400h	Global Event Buffer Address Low Register (GEVNTADRLO)	32	RW	0000_0000h
C404h	Global Event Buffer Address High Register (GEVNTADRHI)	32	RW	0000_0000h
C408h	Global Event Buffer Size Register (GEVNTSIZ)	32	RW	Table 11-29
C40Ch	Global Event Buffer Count Register (GEVNTCOUNT)	32	RW	0000_0000h
C600h	Global Hardware Parameters Register 8 (GHWPARAMS8)	32	RO	0000_0A58h
C60Ch	Global User Control Register 3 (GUCTL3)	32	RW	0001_0000h
C610h	Global Device TX FIFO DMA Priority Register (GTXFIFOPRIDEV)	32	RW	Table 11-29
C618h	Global Host TX FIFO DMA Priority Register (GTXFIFOPRIHST)	32	RW	Table 11-29

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
C61Ch	Global Host RX FIFO DMA Priority Register (GRXFIFOPRIHST)	32	RW	Table 11-29
C620h	Global Host Debug Capability DMA Priority Register (GFIFOPRIDBC)	32	RW	Table 11-29
C624h	Global Host FIFO DMA High-Low Priority Ratio Register (GDMAHLRATIO)	32	RW	Table 11-29
C630h	Global Frame Length Adjustment Register (GFLADJ)	32	RW	Table 11-29
C640h	Global USB 2.0 Root Hub Control Register (GUSB2RHBCTL)	32	RW	0000_0000h
C700h	Device Configuration Register (DCFG)	32	RW	Table 11-29
C704h	Device Control Register (DCTL)	32	RW	Table 11-29
C708h	Device Event Enable Register (DEVTEN)	32	RW	Table 11-29
C70Ch	Device Status Register (DSTS)	32	RW	Table 11-29
C710h	Device Generic Command Parameter Register (DGCMDPAR)	32	RW	0000_0000h
C714h	Device Generic Command Register (DGCMD)	32	RW	Table 11-29
C720h	Device Active USB Endpoint Enable Register (DALEPENA)	32	RW	0000_0000h
C800h	Device Physical Endpoint-n Command Parameter 2 Register (DEPCMDPAR2)	32	RW	0000_0000h
C804h	Device Physical Endpoint-n Command Parameter 1 Register (DEPCMDPAR1)	32	RW	0000_0000h
C808h	Device Physical Endpoint-n Command Parameter 0 Register (DEPCMDPAR0)	32	RW	0000_0000h
C80Ch	Device Physical Endpoint-n Command Register (DEPCMD)	32	RW	0000_0000h
CA00h	Device Interrupt Moderation Register (DEV_IMOD)	32	RW	0000_0000h
CC30h	BC Configuration Register (BCFG)	32	RW	Table 11-29
CC38h	BC Event Register (BCEVT)	32	RW	0000_0000h
CC3Ch	BC Event Enable Register (BCEVTEN)	32	RW	Table 11-29
D000h	U1/U2 LFPS Rx Timer Register (LU1LFPSRXTIM0)	32	RW	0000_1F1Fh
D020h	Link Setting Register (LINK_SETTINGS0)	32	RW	4590_0000h
D024h	Link User Control Register (LLUCTL0)	32	RW	0080_0080h
D028h	Link Datapath Delay Register (LPTMDPDELAY0)	32	RW	0000_0C00h
D800h	U3 Root Hub Debug Register (BU3RHBDBG)	32	RW	0000_0000h
D850h	Block RAM Single Bit Error Count Register (BRSERRCNT)	32	RO	0000_0000h
D854h	Block RAM Multiple Bit Error Count Register (BRMERRCNT)	32	RO	0000_0000h
D858h	Block RAM ECC Error Vector Register (BRMECCERR)	32	RO	0000_0000h
D85Ch	Block RAM ECC Error Control Register (BRERRCTL)	32	RW	0000_0000h
D860h	Block RAM0 Address Error Register (BRAM0ADDRERR)	32	RO	0000_0000h
D864h	Block RAM1 Address Error Register (BRAM1ADDRERR)	32	RO	0000_0000h
D868h	Block RAM2 Address Error Register (BRAM2ADDRERR)	32	RO	0000_0000h

11.2.5.1.2 Capability Registers Length and Host Controller Operational Registers (CAPLENGTH)

11.2.5.1.2.1 Offset

Register	Offset
CAPLENGTH	0h

11.2.5.1.2.2 Function

Capability Registers Length = Base address + CAPLENGTH where CAPLENGTH is `DWC_USB3_HOST_CAP_REG_LEN whose default value is 20h.

Reset Mask:0xFF00

11.2.5.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HCIVERSION															
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								CAPLENGTH							
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

11.2.5.1.2.4 Fields

Field	Function
31-16 HCIVERSION	Host Controller Interface Version Number This is a two-byte register containing a BCD encoding of the xHCI specification revision number supported by this host controller. The most significant byte of this register represents a major revision and the least significant byte is the minor revision. e.g. 0100h corresponds to xHCI version 1.0.
15-8 —	Reserved
7-0 CAPLENGTH	Capability Registers Length It is used as an offset to add to register base to find the beginning of the Operational Register Space.

11.2.5.1.3 Structural Parameters 1 Register (HCSPARAMS1)

11.2.5.1.3.1 Offset

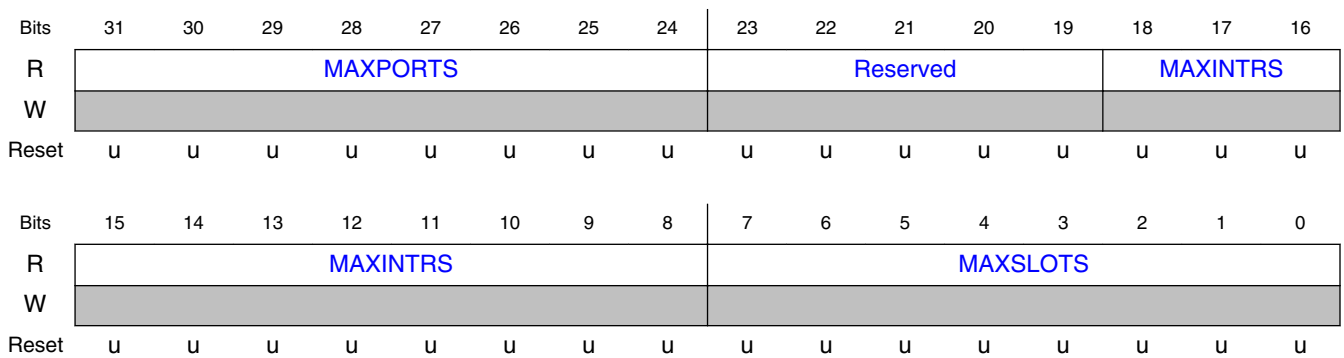
Register	Offset
HCSPARAMS1	4h

11.2.5.1.3.2 Function

For register definitions, refer to the xHCI specification.

Reset Mask:0xF80000

11.2.5.1.3.3 Diagram



11.2.5.1.3.4 Fields

Field	Function
31-24 MAXPORTS	Number of Ports • Number of ports implemented is defined by the parameter (`DWC_USB3_HOST_NUM_U2_ROOT_PORTS + `DWC_USB3_HOST_NUM_U3_ROOT_PORTS) • Number of ports enabled is controlled by the core input signals host_num_u2_port[3:0]+host_num_u3_port[3:0] **NOTE:** In USB 2.0-only mode, the host_num_u3_port signal is zero.
23-19 —	Reserved
18-8 MAXINTRS	Number of Interrupters Defined by the configurable parameter `DWC_USB3_HOST_NUM_INTERRUPTER_SUPT
7-0 MAXSLOTS	Number of Device Slots Defined by configurable parameter `DWC_USB3_NUM_DEVICE_SUPT

11.2.5.1.4 Structural Parameters 2 Register (HCSPARAMS2)

11.2.5.1.4.1 Offset

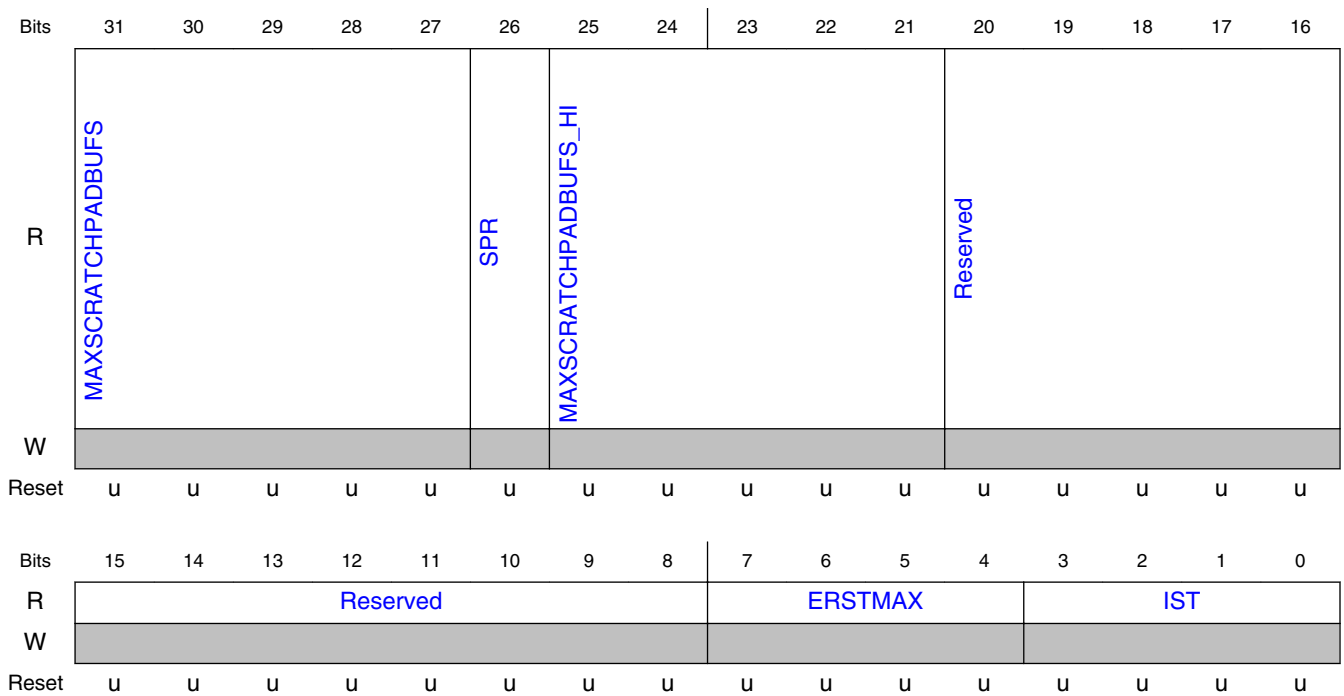
Register	Offset
HCSPARAMS2	8h

11.2.5.1.4.2 Function

For register definitions, refer to the xHCI specification.

Reset Mask:0x1FFF00

11.2.5.1.4.3 Diagram



11.2.5.1.4.4 Fields

Field	Function
31-27 MAXSCRATCHPADBUFS	Max Scratchpad Bufs Lo The value is calculated based on chosen configuration parameter values. Possible values are 1-4.
26	Scratchpad Restore

Table continues on the next page...

Field	Function
SPR	
25-21 MAXSCRATCHPADBUFS_HI	Max Scratchpad Bufs HI The core automatically updates this field.
20-8 —	Reserved
7-4 ERSTMAX	Event Ring Segment Table Max
3-0 IST	Isochronous Scheduling Threshold

11.2.5.1.5 Structural Parameters 3 Register (HCSPARAMS3)

11.2.5.1.5.1 Offset

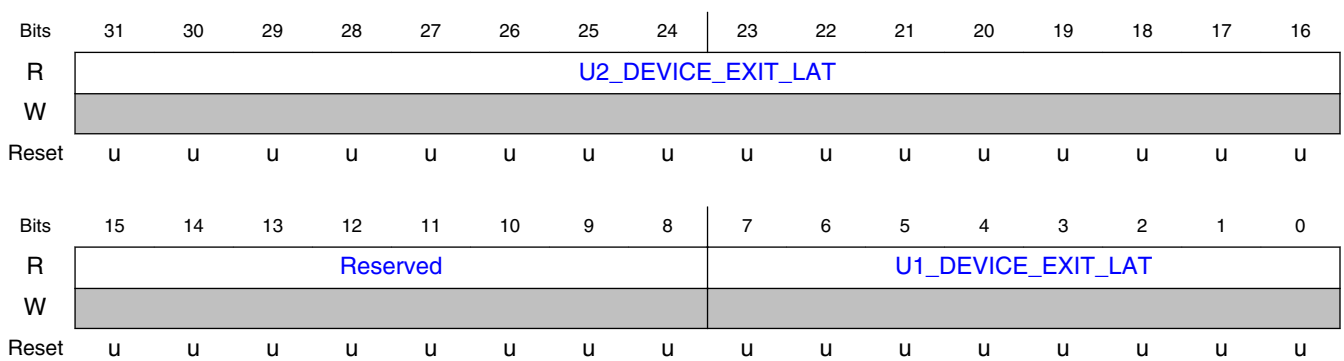
Register	Offset
HCSPARAMS3	Ch

11.2.5.1.5.2 Function

For register definitions, refer to the xHCI specification.

Reset Mask:0xFF00

11.2.5.1.5.3 Diagram



11.2.5.1.5.4 Fields

Field	Function
31-16 U2_DEVICE_EXIT_LAT	U2 Device Exit Latency
15-8 —	Reserved
7-0 U1_DEVICE_EXIT_LAT	U1 Device Exit Latency

11.2.5.1.6 Capability Parameters 1 Register (HCCPARAMS1)

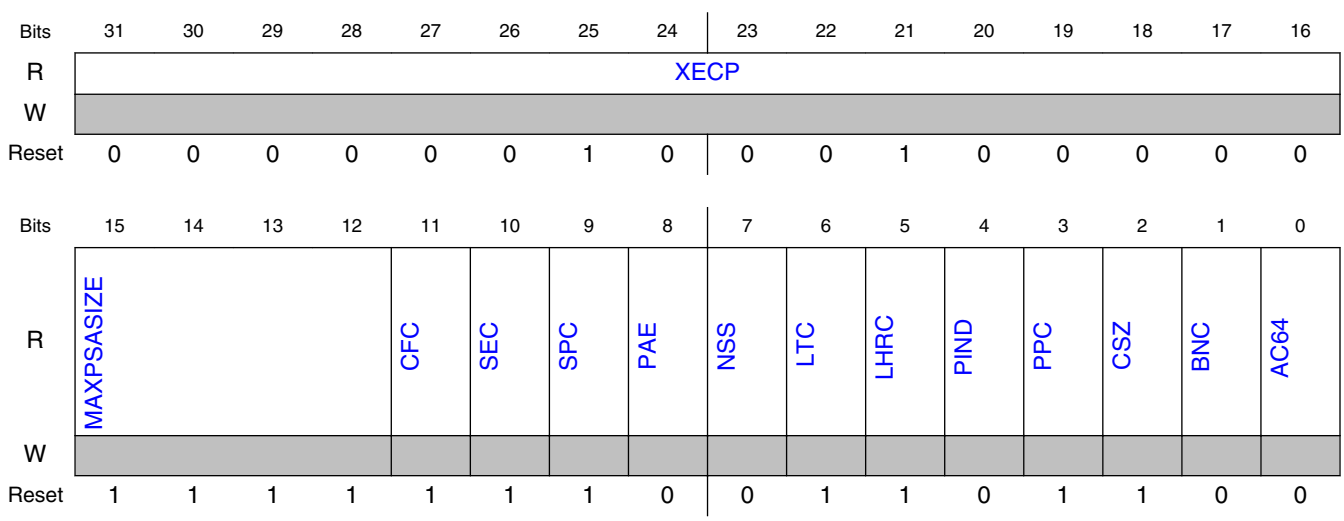
11.2.5.1.6.1 Offset

Register	Offset
HCCPARAMS1	10h

11.2.5.1.6.2 Function

For register definitions, refer to the xHCI specification.

11.2.5.1.6.3 Diagram



11.2.5.1.6.4 Fields

Field	Function
31-16 XECP	xHCI Extended Capabilities Pointer Based on configuration, core automatically updates it. For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
15-12 MAXPSASIZE	Maximum Primary Stream Array Size For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
11 CFC	Contiguous Frame ID Capability For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
10 SEC	Stopped EDLTA Capability For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
9 SPC	Short Packet Capability For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
8 PAE	Parse All Event Data For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
7 NSS	No Secondary SID Support For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
6 LTC	Latency Tolerance Messaging Capability For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
5 LHRC	Light HC Reset Capability For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
4 PIND	Port Indicators For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
3 PPC	Port Power Control For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
2 CSZ	Context Size For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
1 BNC	BW Negotiation Capability For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
0 AC64	64-bit Addressing Capability For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.7 Doorbell Offset Register (DBOFF)

11.2.5.1.7.1 Offset

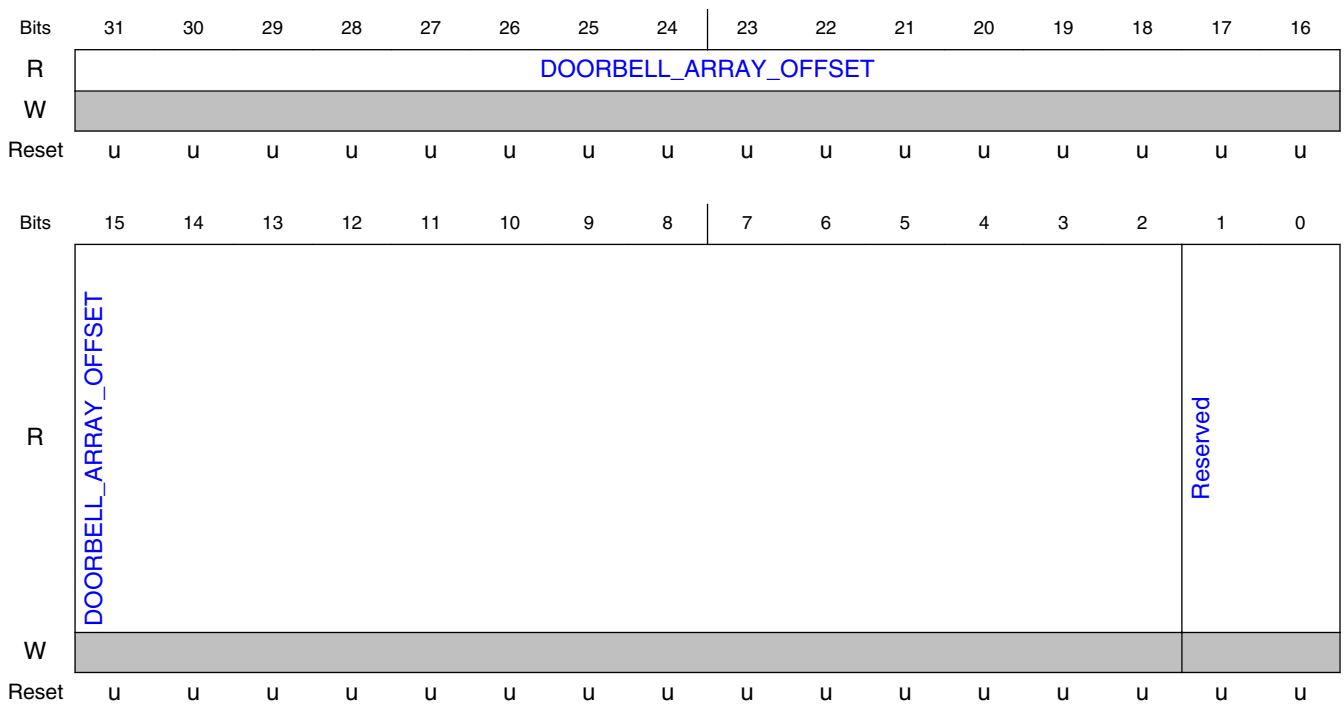
Register	Offset
DBOFF	14h

11.2.5.1.7.2 Function

For register definitions, refer to the xHCI specification.

Reset Mask:0x3

11.2.5.1.7.3 Diagram



11.2.5.1.7.4 Fields

Field	Function
31-2 DOORBELL_ARRAY_OFFSET	Doorbell Array Offset RO Based on configuration, core automatically updates it. For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Table continues on the next page...

Field	Function
1-0 —	Reserved

11.2.5.1.8 Runtime Register Space Offset Register (RTSOFF)

11.2.5.1.8.1 Offset

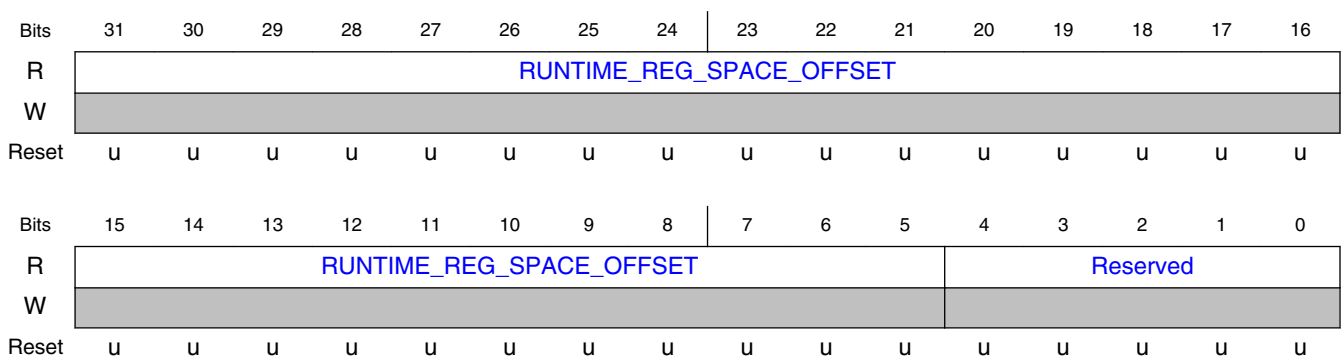
Register	Offset
RTSOFF	18h

11.2.5.1.8.2 Function

For register definitions, refer to the xHCI specification.

Reset Mask:0x1F

11.2.5.1.8.3 Diagram



11.2.5.1.8.4 Fields

Field	Function
31-5 RUNTIME_REG_SPACE_OFFSET	Runtime Register Space Offset Based on configuration, core automatically updates it. For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
4-0 —	Reserved

11.2.5.1.9 Host Controller Capability Parameters 2 (HCCPARAMS2)

11.2.5.1.9.1 Offset

Register	Offset
HCCPARAMS2	1Ch

11.2.5.1.9.2 Function

For register definitions, refer to the xHCI specification.

11.2.5.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										CIC	LEC	CTC	FSC	CMC	U3C
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1

11.2.5.1.9.4 Fields

Field	Function
31-6 —	Reserved
5 CIC	Configuration Information Capability For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
4 LEC	Large ESIT Payload Capability For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
3 CTC	Compliance Transition Capability For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
2 FSC	Force Save Context Capability

Table continues on the next page...

Field	Function
	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
1 CMC	Configure Endpoint Command Max Exit Latency Too Large Capability For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
0 U3C	U3 Entry Capability For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.10 USB Command Register (USBCMD)

11.2.5.1.10.1 Offset

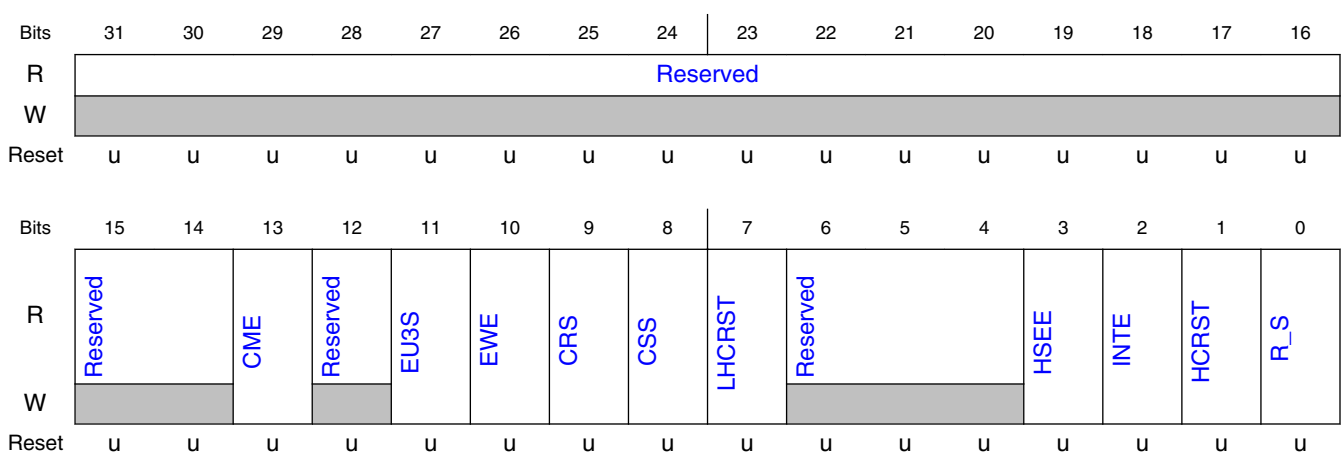
Register	Offset
USBCMD	20h

11.2.5.1.10.2 Function

For a description of this standard USB register field see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

Reset Mask:0xFFFFC070

11.2.5.1.10.3 Diagram



11.2.5.1.10.4 Fields

Field	Function
31-14 —	Reserved
13 CME	CEM Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
12 —	Reserved
11 EU3S	Enable U3 MFINDEX Stop For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
10 EWE	Enable Wrap Event For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
9 CRS	Controller Restore State This command is similar to the USBCMD.CRS bit in host mode and initiates the restore process. When software sets this bit to '1', the controller immediately sets DSTS.RSS to '1'. When the controller has finished the restore process, it sets DSTS.RSS to '0'. **NOTE:** When read, this field always returns '0'.
8 CSS	Controller Save State This command is similar to the USBCMD.CSS bit in host mode and initiates the save process. When software sets this bit to '1', the controller immediately sets DSTS.SSS to '1'. When the controller has finished the save process, it sets DSTS.SSS to '0'. **NOTE:** When read, this field always returns '0'.
7 LHCRST	Light Host Controller Reset For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0. The following bits reset the internal logic of the host controller. Under soft reset, some CSR accesses may fail (Timeout). - HCRST - LHCRST Bit Bash register testing is not recommended.
6-4 —	Reserved
3 HSEE	Host System Error Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
2 INTE	Interrupter Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
1 HCRST	Host Controller Reset The following bits reset the internal logic of the host controller. Under soft reset, some CSR accesses may fail (Timeout). - HCRST - LHCRST Bit Bash register testing is not recommended.
0 R_S	Run/Stop For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0. Due to side-effects this reguster field is not recommended for Bit-Bash testing.

11.2.5.1.11 USB Status Register (USBSTS)

11.2.5.1.11.1 Offset

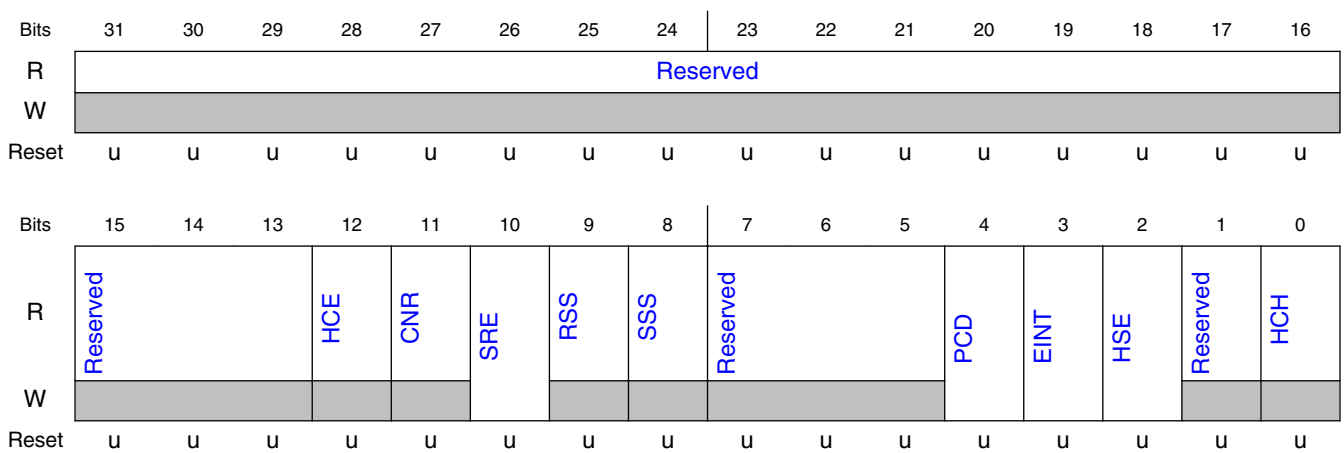
Register	Offset
USBSTS	24h

11.2.5.1.11.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

Reset Mask:0xFFFFE0E2

11.2.5.1.11.3 Diagram



11.2.5.1.11.4 Fields

Field	Function
31-13 —	Reserved
12 HCE	Host Controller Error RO Default = 0. '0' = No internal xHC error conditions exist and '1' = Internal xHC error condition. This flag must be set to indicate that an internal error condition has been detected which requires software to reset and reinitialize the xHC. Refer to section 4.24.1 of the DesignWare Cores SuperSpeed USB 3.0 Controller Databookfor more information.
11	Controller Not Ready

Table continues on the next page...

Field	Function
CNR	RO Default = '1, '0' = Ready, and '1' = Not Ready. Software must not write to thes Doorbell or Operational register of the xHC, other than the USBSTS register, until CNR = '0'. This flag is set by the xHC after a Chip Hardware Reset and cleared when the xHC is ready to begin accepting register writes. This flag remains cleared ('0') until the next Chip Hardware Reset.
10 SRE	Save/Restore Error This bit is currently not supported.
9 RSS	Restore State Status This bit is similar to the USBSTS.RSS in host mode. When the controller has finished the restore process, it completes the command by setting DSTS.RSS to '0'.
8 SSS	Save State Status This bit is similar to the USBSTS.SSS in host mode. When the controller has finished the save process, it completes the command by setting DSTS.SSS to '0'.
7-5 —	Reserved
4 PCD	Port Change Detect For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
3 EINT	Event Interrupt For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
2 HSE	Host System Error For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
1 —	Reserved
0 HCH	HC Halted For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.12 Page Size Register (PAGESIZE)

11.2.5.1.12.1 Offset

Register	Offset
PAGESIZE	28h

11.2.5.1.12.2 Function

Use this register to enable or disable the reporting of specific USB Device Notification Transaction Packets being received.

Reset Mask:0xFFFF0000

A Notification Enable (Nx, where x = 0 to 15) flag is defined for each of the 16 possible device notification types. If a flag is set for a specific notification type, a Device Notification Event is generated when the respective notification packet is received. After reset, all notifications are disabled. Refer to section 6.4.2.7 of the DesignWare Cores SuperSpeed USB 3.0 Controller Databookfor more information. This register is written as a Dword. Byte writes produce undefined results.

11.2.5.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PAGE_SIZE															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

11.2.5.1.12.4 Fields

Field	Function
31-16 —	Reserved
15-0 PAGE_SIZE	Page Size For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.13 Device Notification Register (DNCTRL)

11.2.5.1.13.1 Offset

Register	Offset
DNCTRL	34h

11.2.5.1.13.2 Function

For a description of this standard USB register field see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.2.5.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	N0_N15															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.13.4 Fields

Field	Function
31-16 —	Reserved
15-0 N0_N15	Notification Enable For a description of this standard USB register field see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.14 Command Ring Control Register - Low (CRCR_LO)

11.2.5.1.14.1 Offset

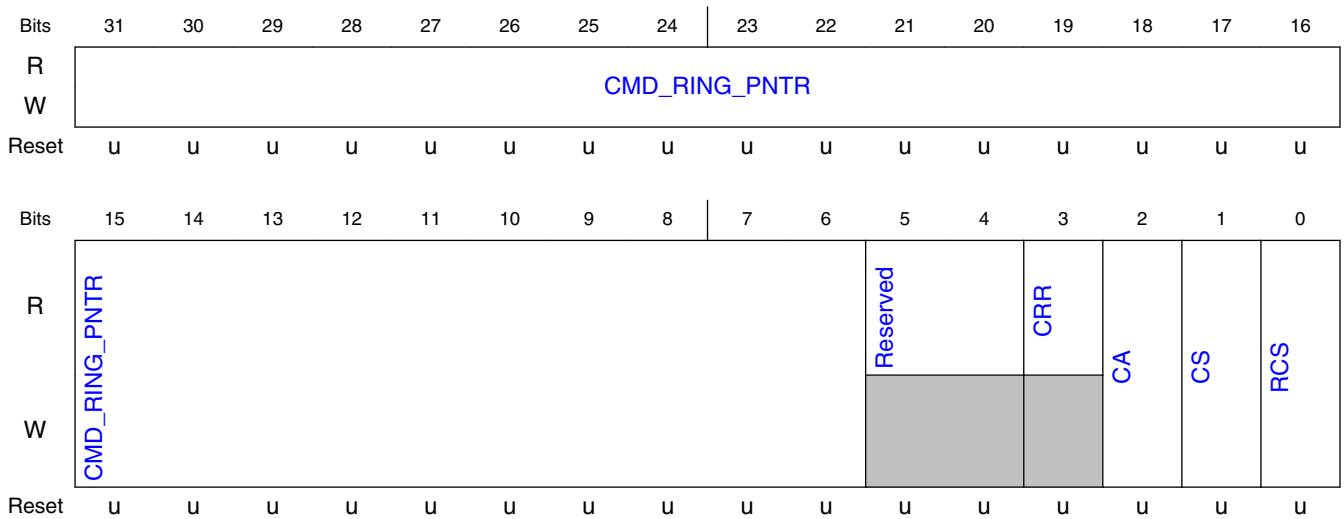
Register	Offset
CRCR_LO	38h

11.2.5.1.14.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

Reset Mask:0x30

11.2.5.1.14.3 Diagram



11.2.5.1.14.4 Fields

Field	Function
31-6 CMD_RING_PNTR	Command Ring Pointer For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
5-4 —	Reserved
3 CRR	Command Ring Running For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
2 CA	Command Abort For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
1 CS	Command Stop For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
0 RCS	Ring Cycle State For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.15 Command Ring Control Register - High (CRCR_HI)

11.2.5.1.15.1 Offset

Register	Offset
CRCR_HI	3Ch

11.2.5.1.15.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.2.5.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CMD_RING_PNTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMD_RING_PNTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.15.4 Fields

Field	Function
31-0 CMD_RING_PNTR	Command Ring Pointer Reading this field always returns '0'. For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.16 Device Context Base Address Array Pointer Register - Low (DCBAAP_LO)

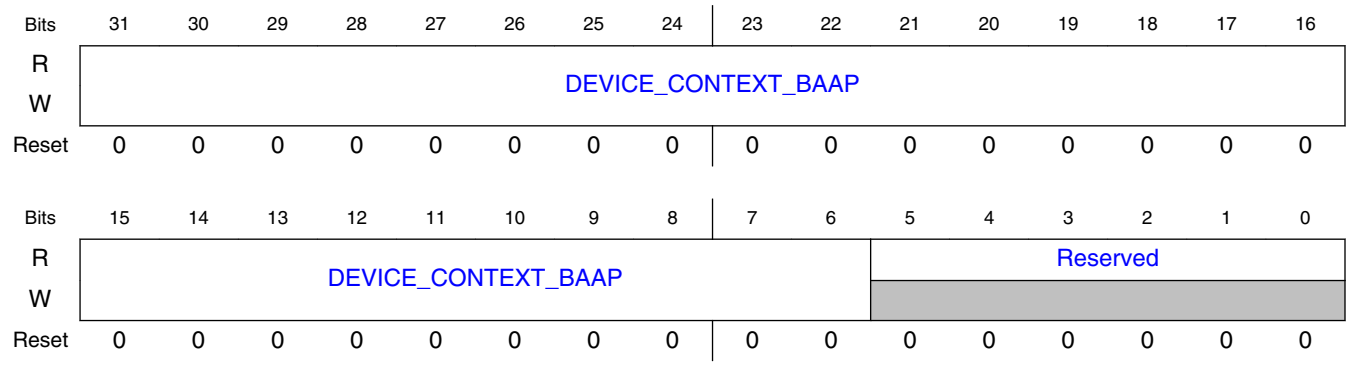
11.2.5.1.16.1 Offset

Register	Offset
DCBAAP_LO	50h

11.2.5.1.16.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.2.5.1.16.3 Diagram



11.2.5.1.16.4 Fields

Field	Function
31-6 DEVICE_CONTEXT_BAAP	Device Context Base Address Array Pointer For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
5-0 —	Reserved

11.2.5.1.17 Device Context Base Address Array Pointer Register -High (DCBAAP_HI)

11.2.5.1.17.1 Offset

Register	Offset
DCBAAP_HI	54h

11.2.5.1.17.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.2.5.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DEVICE_CONTEXT_BAAP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DEVICE_CONTEXT_BAAP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.17.4 Fields

Field	Function
31-0 DEVICE_CONTEXT_BAAP	Device Context Base Address Array Pointer For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.18 Configuration Register (CONFIG)

11.2.5.1.18.1 Offset

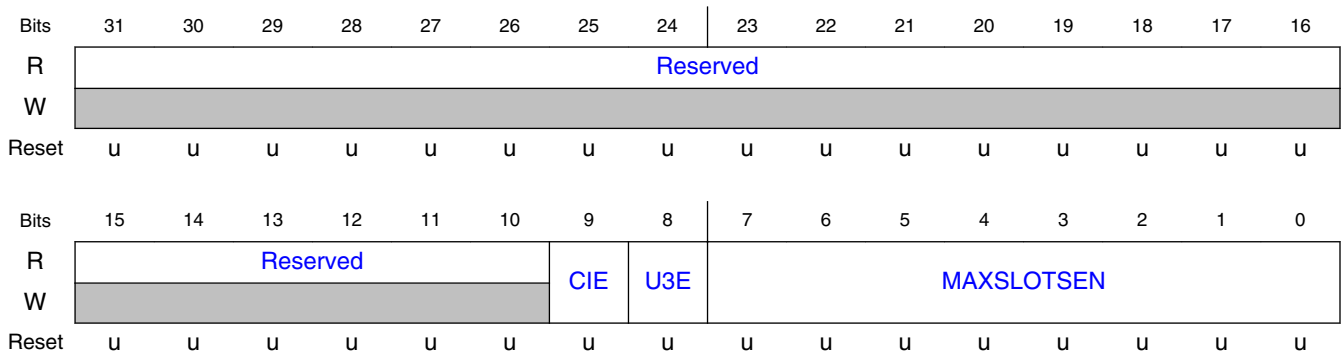
Register	Offset
CONFIG	58h

11.2.5.1.18.2 Function

This register is in the Aux Power well.

Reset Mask:0xFFFFFC00

11.2.5.1.18.3 Diagram



11.2.5.1.18.4 Fields

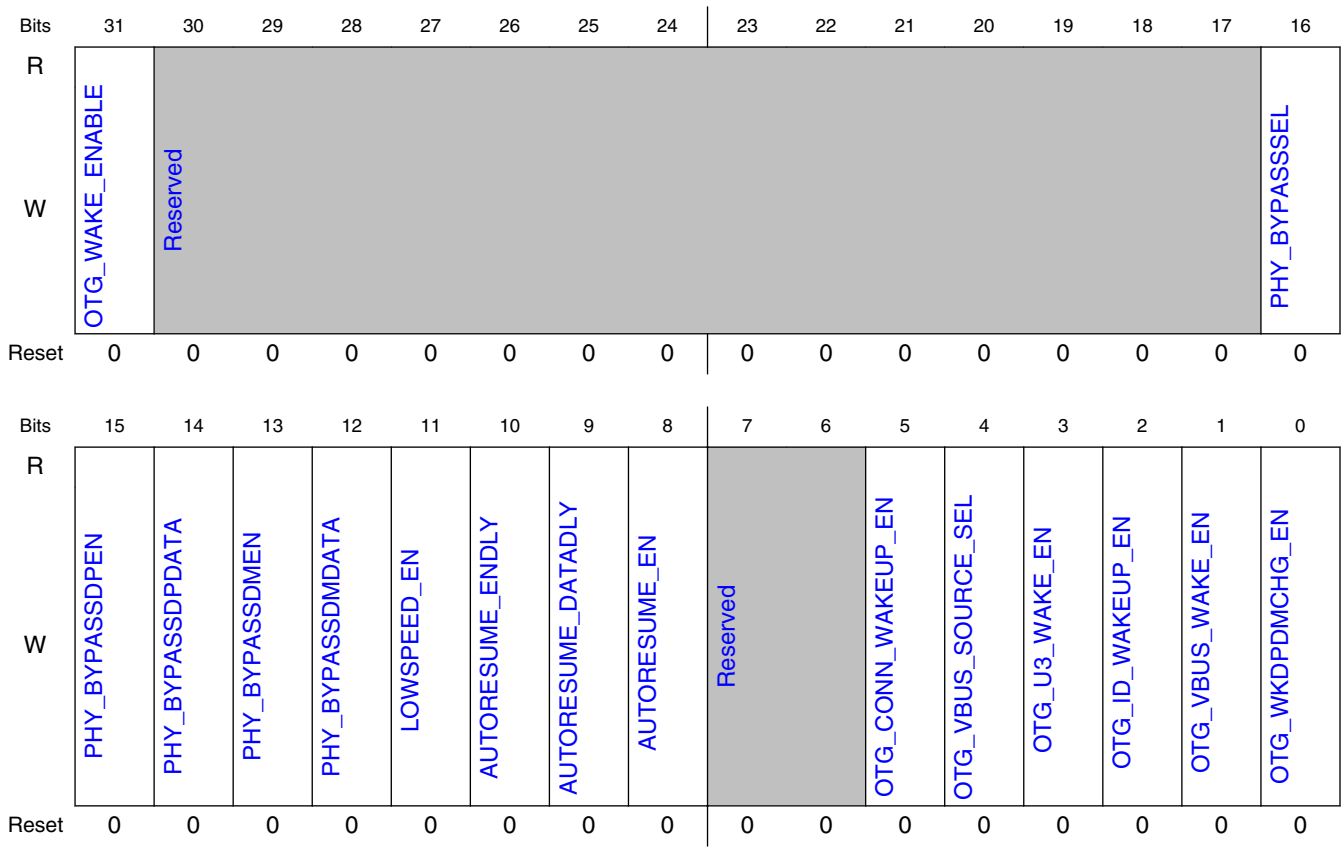
Field	Function
31-10 —	Reserved
9 CIE	Configuration Information Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
8 U3E	U3 Entry Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
7-0 MAXSLOTSEN	Max Device Slots Enabled For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.19 USB1 Wakeup Register (USB1_WAKEUP_CTRL)

11.2.5.1.19.1 Offset

Register	Offset
USB1_WAKEUP_CTRL	100h

11.2.5.1.19.2 Diagram



11.2.5.1.19.3 Fields

Field	Function
31 OTG_WAKE_ENABLE	Global Wakeup Interrupt Enable Used to clear interrupt. Default 0.
30-17 —	Reserved
16 PHY_BYPASSSEL	Transmitter Digital Bypass Select
15 PHY_BYPASSDPEN	DP Transmitter Digital Bypass Enable
14 PHY_BYPASSDPDATA	Data for DP Transmitter Digital Bypass
13	DM Transmitter Digital Bypass Enable

Table continues on the next page...

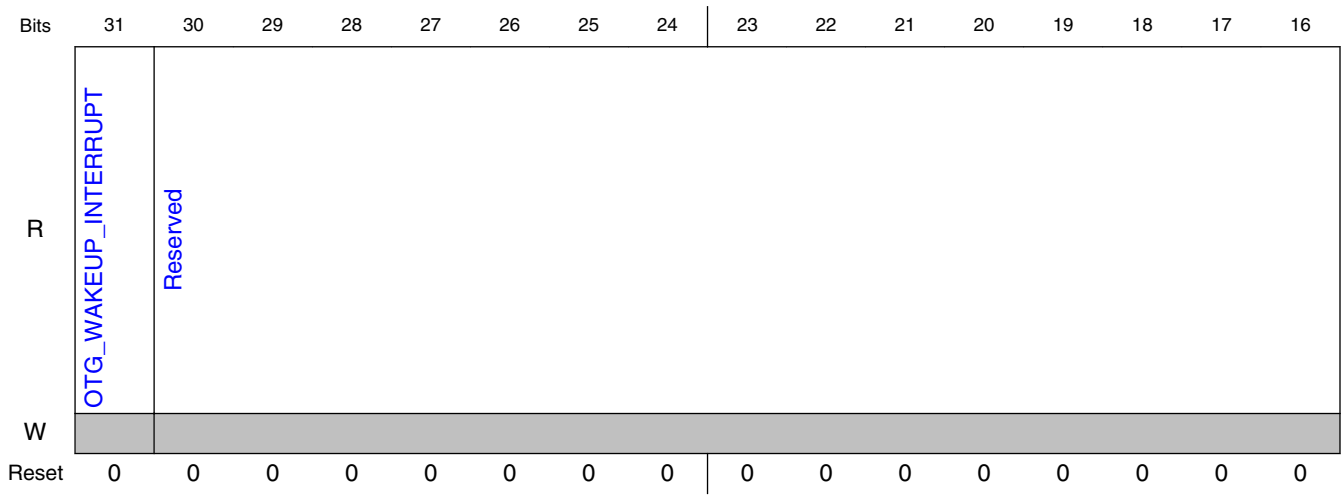
Field	Function
PHY_BYPASSDMEN	
12 PHY_BYPASSDMDATA	Data for DM Transmitter Digital Bypass
11 LOWSPEED_EN	Enable lowspeed autoresume feature. 0b - Full/high speed 1b - Lowspeed
10 AUTORESUME_ENDLY	Tuning the timing between dp/dm data and enable signal when autoresume finish. {autoresume_endly, autoresume_datadly}= 00 or 11: data and enable pulldown together 01: enable signal pull down later than data pin one 32k clock cycle 10: enable signal pull down earlier than data pin one 32k clock cycle
9 AUTORESUME_DATADLY	See to the bitfield description for AUTORESUME_ENDLY. 0b - Disable 1b - Enable
8 AUTORESUME_EN	Enable autoresume feature. When using phy_bypass signal, it should be 1'b0. 0b - Disable 1b - Enable
7-6 —	Reserved
5 OTG_CONN_WAKEUP_EN	Enable signal for wakeup from connection or disconnection, only for superspeed.
4 OTG_VBUS_SOURCE_SEL	0b - Select vbus_valid 1b - Select sessvld
3 OTG_U3_WAKE_EN	Enable signal for wake up from u3 state, only for superspeed. 0b - Disable 1b - Enable
2 OTG_ID_WAKEUP_EN	Enable signal for wake up from id change. 0b - Disable 1b - Enable
1 OTG_VBUS_WAKE_EN	Enable signal for wake up from vbus valid or session valid changes. 0b - Disable 1b - Enable
0 OTG_WKDPDMCHG_EN	Enable signal for wake up from dp/dm change. 0b - Disable 1b - Enable

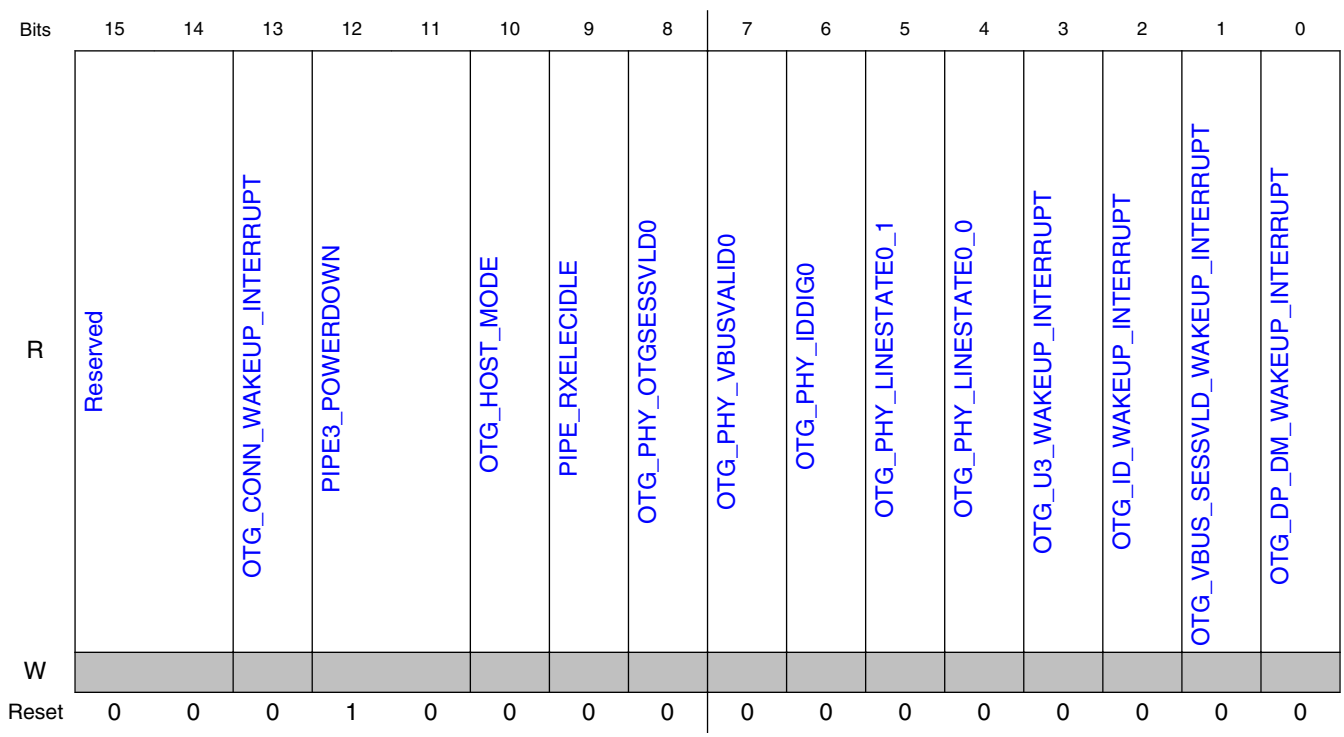
11.2.5.1.20 USB1 Wakeup Status Register (USB1_WAKEUP_STATUS)

11.2.5.1.20.1 Offset

Register	Offset
USB1_WAKEUP_STATUS	104h

11.2.5.1.20.2 Diagram





11.2.5.1.20.3 Fields

Field	Function
31 OTG_WAKEUP_INTERRUPT	It is OR of all wakeup interrupt, then AND with otg_wakeup_enable. All separate wakeup interrupt status need to be registered by wakeup enable, so clear wakeup enable, the related wakeup interrupt will be cleared tens of us later..
30-14 —	Reserved
13 OTG_CONN_WAKEUP_INTERRUPT	Interrupt status of connection
12-11 PIPE3_POWERDOWN	Pipe powerdown
10 OTG_HOST_MODE	USB drd mode indicator 0b - Device mode 1b - Host mode
9 PIPE_RXELECIDLE	Pipe_rxelecidel, wakeup source
8 OTG_PHY_OTGSESSVLD0	Session valid, wakeup source
7 OTG_PHY_VBUSVALID0	Vbus valid, wakeup source
6 OTG_PHY_IDDIG0	ID status, wakeup source
5 OTG_PHY_LINESTATE0_1	Linestate[1], wakeup source
4 OTG_PHY_LINESTATE0_0	Linestate[0], wakeup source
3 OTG_U3_WAKEUP_INTERRUPT	Interrupt status of wakeup from u3 state
2 OTG_ID_WAKEUP_INTERRUPT	Interrupt status of wakeup from id.
1	Interrupt status of vbus or session valid signal. otg_vbus_wakeup_interrupt | otg_sessvld_wakeup_interrupt

Table continues on the next page...

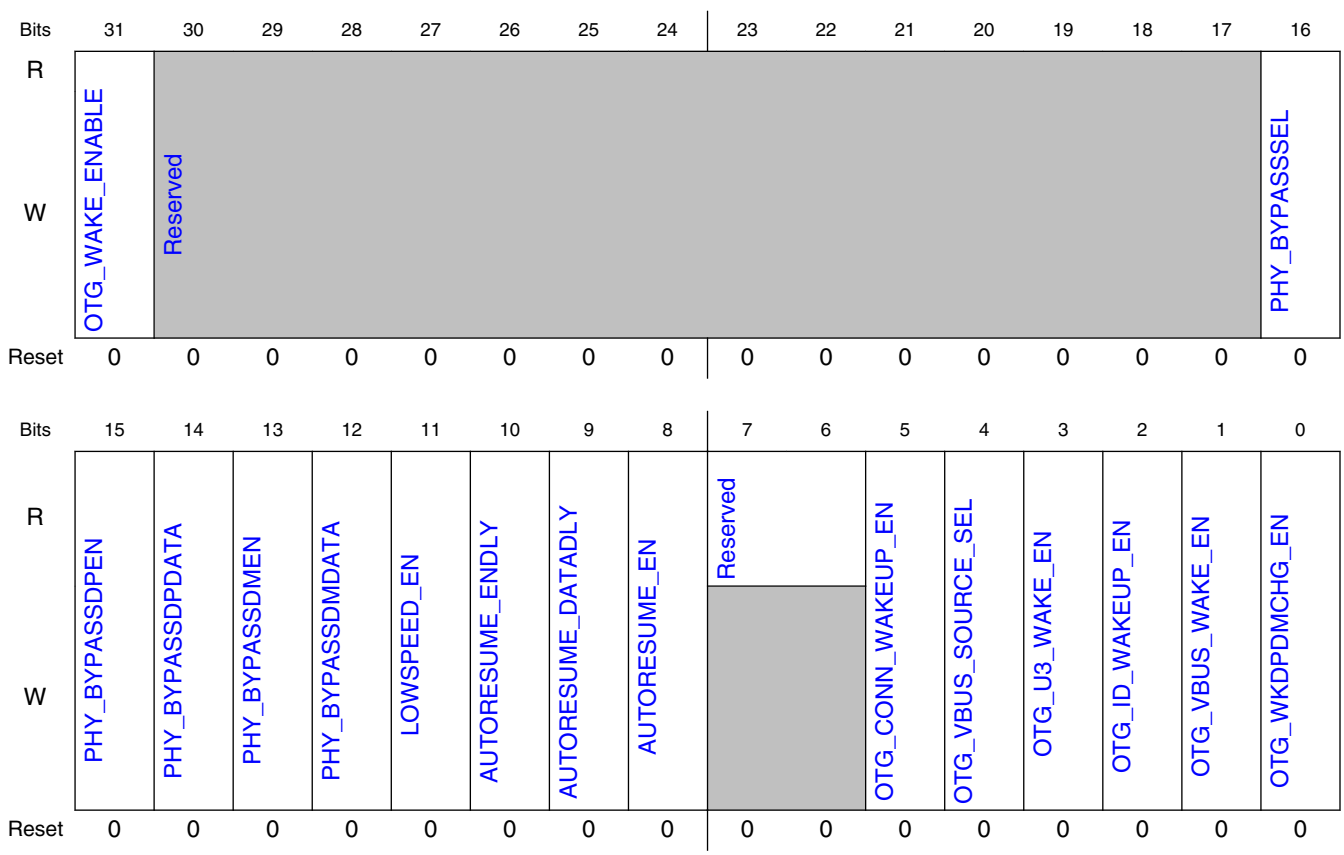
Field	Function
OTG_VBUS_SESSVLD_WAKEUP_INTERRUPT	
0 OTG_DP_DM_WAKEUP_INTERRUPT	Interrupt status of dm or dp, otg_dp_wakeup_interrupt | otg_dm_wakeup_interrupt

11.2.5.1.21 USB2 Wakeup Register (USB2_WAKEUP_CTRL)

11.2.5.1.21.1 Offset

Register	Offset
USB2_WAKEUP_CTRL	108h

11.2.5.1.21.2 Diagram



11.2.5.1.21.3 Fields

Field	Function
31 OTG_WAKE_ENABLE	Global Wakeup Interrupt Enable Used to clear interrupt. Default 0.
30-17 —	Reserved
16 PHY_BYPASSSEL	Transmitter Digital Bypass Select
15 PHY_BYPASSDPEN	DP Transmitter Digital Bypass Enable
14 PHY_BYPASSDPDATA	Data for DP Transmitter Digital Bypass
13 PHY_BYPASSDMEN	DM Transmitter Digital Bypass Enable
12 PHY_BYPASSDMDATA	Data for DM Transmitter Digital Bypass
11 LOWSPEED_EN	Enable lowspeed autoresume feature. 0b - Full/high speed 1b - Lowspeed
10 AUTORESUME_ENDLY	Tuning the timing between dp/dm data and enable signal when autoresume finish. {autoresume_endly, autoresume_datadly}= 00 or 11: data and enable pulldown together 01: enable signal pull down later than data pin one 32k clock cycle 10: enable signal pull down earlier than data pin one 32k clock cycle
9 AUTORESUME_DATADLY	See to the bitfield description for AUTORESUME_ENDLY. 0b - Disable 1b - Enable
8 AUTORESUME_EN	Enable autoresume feature. When using phy_bypass signal, it should be 1'b0. 0b - Disable 1b - Enable
7-6 —	Reserved
5 OTG_CONN_WAKEUP_EN	Enable signal for wakeup from connection or disconnection, only for superspeed.
4	0b - Select vbus_valid 1b - Select sessvld

Table continues on the next page...

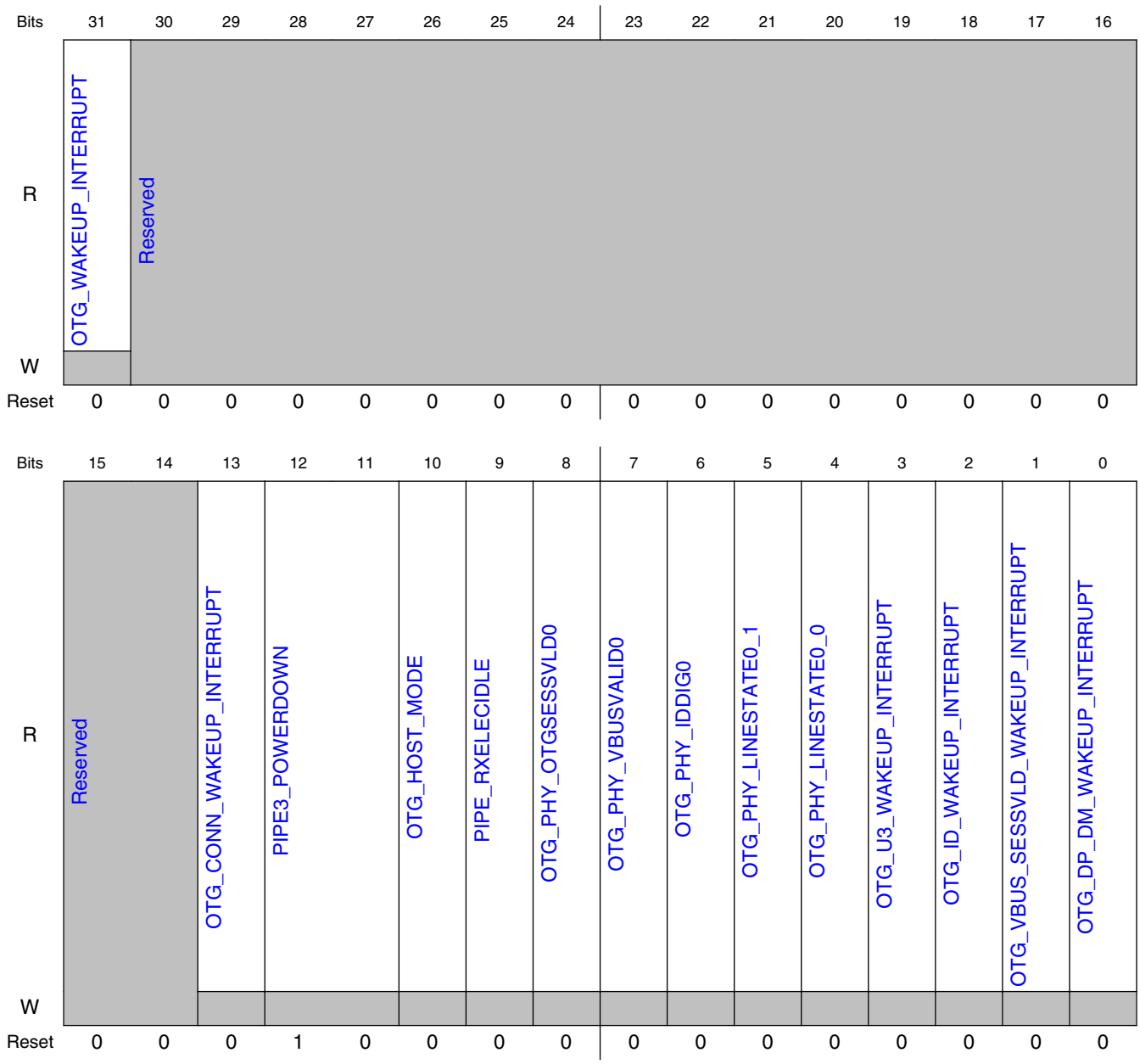
Field	Function
OTG_VBUS_SOURCE_SEL	
3 OTG_U3_WAKE_EN	Enable signal for wake up from u3 state, only for superspeed. 0b - Disable 1b - Enable
2 OTG_ID_WAKEUP_EN	Enable signal for wake up from id change. 0b - Disable 1b - Enable
1 OTG_VBUS_WAKE_EN	Enable signal for wake up from vbus valid or session valid changes. 0b - Disable 1b - Enable
0 OTG_WKDPDMCHG_EN	Enable signal for wake up from dp/dm change. 0b - Disable 1b - Enable

11.2.5.1.22 USB2 Wakeup Status Register (USB2_WAKEUP_STATUS)

11.2.5.1.22.1 Offset

Register	Offset
USB2_WAKEUP_STATUS	10Ch

11.2.5.1.22.2 Diagram



11.2.5.1.22.3 Fields

Field	Function
31 OTG_WAKEUP_INTERRUPT	It is OR of all wakeup interrupt, then AND with otg_wakeup_enable. All separate wakeup interrupt status need to be registered by wakeup enable, so clear wakeup enable, the related wakeup interrupt will be cleared tens of us later..
30-14 —	Reserved

Table continues on the next page...

Field	Function
13 OTG_CONN_WAKEUP_INTERRUPT	Interrupt status of connection
12-11 PIPE3_POWERDOWN	Pipe powerdown
10 OTG_HOST_MODE	USB drd mode indicator 0b - Device mode 1b - Host mode
9 PIPE_RXELECIDLE	Pipe_rxelecidel, wakeup source
8 OTG_PHY_OTGSESSVLD0	Session valid, wakeup source
7 OTG_PHY_VBUSVALID0	Vbus valid, wakeup source
6 OTG_PHY_IDDIG0	ID status, wakeup source
5 OTG_PHY_LINESTATE0_1	Linestate[1], wakeup source
4 OTG_PHY_LINESTATE0_0	Linestate[0], wakeup source
3 OTG_U3_WAKEUP_INTERRUPT	Interrupt status of wakeup from u3 state
2 OTG_ID_WAKEUP_INTERRUPT	Interrupt status of wakeup from id.
1 OTG_VBUS_SESSVLD_WAKEUP_INTERRUPT	Interrupt status of vbus or session valid signal. otg_vbus_wakeup_interrupt | otg_sessvld_wakeup_interrupt
0 OTG_DP_DM_WAKEUP_INTERRUPT	Interrupt status of dm or dp, otg_dp_wakeup_interrupt | otg_dm_wakeup_interrupt

11.2.5.1.23 USB2 Port Status and Control Register (PORTSC_20)

11.2.5.1.23.1 Offset

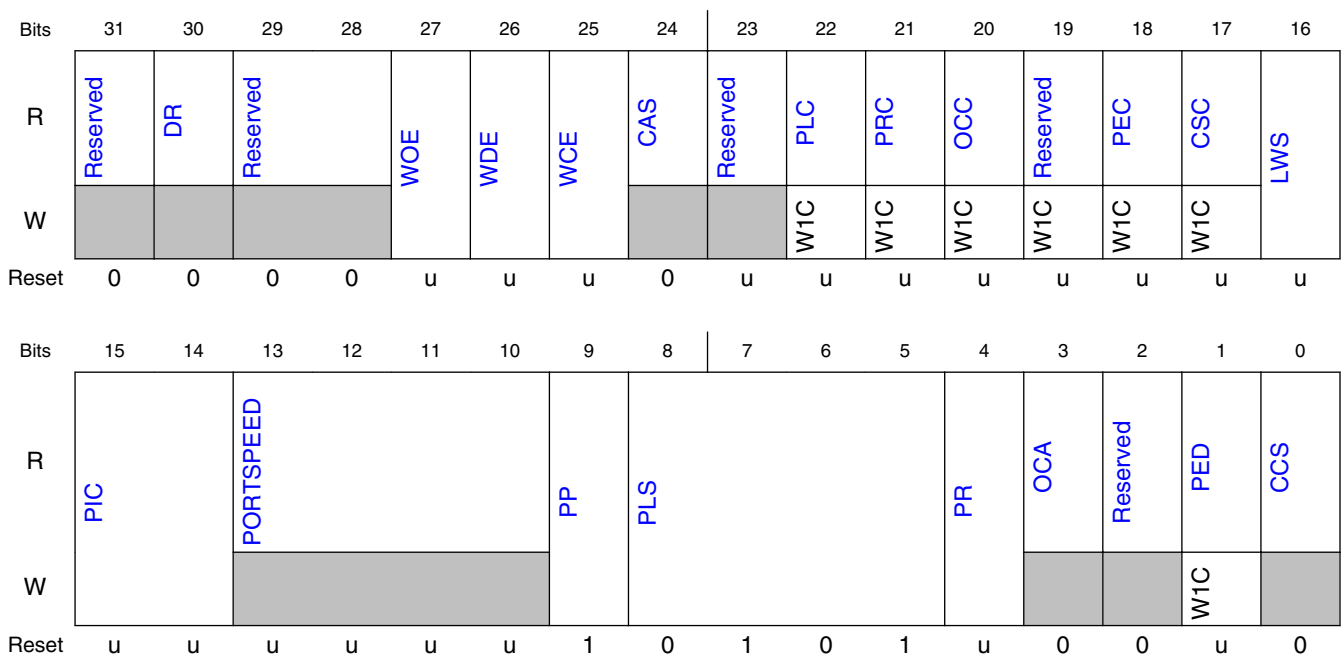
Register	Offset
PORTSC_20	420h

11.2.5.1.23.2 Function

Access fails (Timeout) if the UTMI/ULPI clock is not running or one of the following bits is asserted.

Reset Mask:0xF10003ED

11.2.5.1.23.3 Diagram



11.2.5.1.23.4 Fields

Field	Function
31 —	Reserved

Table continues on the next page...

Field	Function
30 DR	Device Removable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
29-28 —	Reserved
27 WOE	Wake on Overcurrent Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
26 WDE	Wake on Disconnect Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
25 WCE	Wake on Connect Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
24 CAS	Cold Attach Status For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
23 —	Reserved
22 PLC	Port Link State Change For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
21 PRC	Port Reset Change For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
20 OCC	Over Current Change For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
19 —	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
18 PEC	Port Enabled/Disabled Change For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
17 CSC	Connect Status Change For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
16 LWS	Port Link State Write Strobe For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
15-14 PIC	Port Indicator Control For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
13-10	Port Speed

Table continues on the next page...

Field	Function
PORTSPEED	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
9 PP	Port Power For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
8-5 PLS	Port Link State For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
4 PR	Port Reset DWC_usb3_map/DWC_usb3_block_Host_Cntrl_Port_Reg_Set/PORTSC_20_REGS/PORTSC_20/PR VolatileMemory 1 Programming this field with random data will cause side effect.
3 OCA	Over Current Active For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
2 —	Reserved
1 PED	Port Enabled/Disabled For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
0 CCS	Current Connect Status For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.24 USB2 Port Power Management Status and Control Register (PORTPMSC_20)

11.2.5.1.24.1 Offset

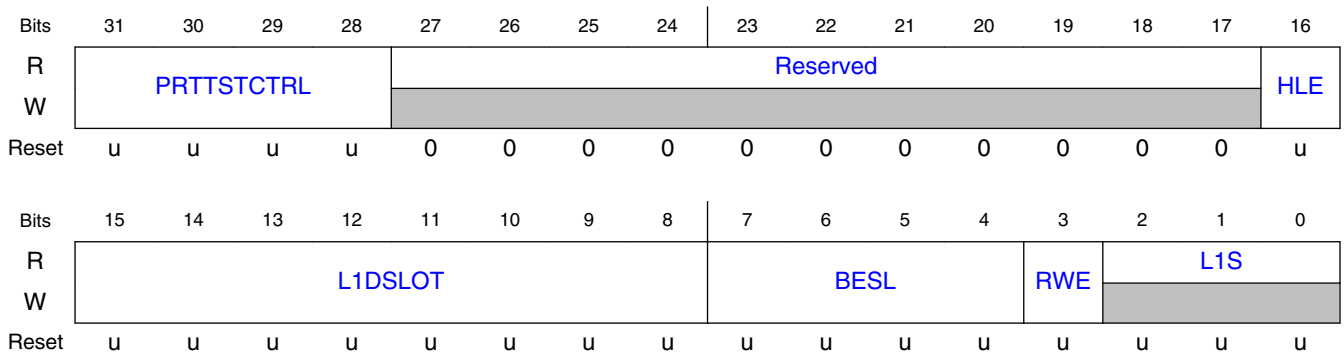
Register	Offset
PORTPMSC_20	424h

11.2.5.1.24.2 Function

This register is in the Aux Power well.

Reset Mask:0xFFE0000

11.2.5.1.24.3 Diagram



11.2.5.1.24.4 Fields

Field	Function
31-28 PRTTSTCTRL	Port Test Control When this field is '0', the port is NOT operating in a test mode. A non-zero value indicates that it is operating in test mode and the specific test mode is indicated by the specific value. A non-zero Port Test Control value is only valid to a port that is in the Powered-Off state (PLS = Disabled). If the port is not in this state, the xHC shall respond with the Port Test Control field set to Port Test Control Error. Refer to 'Operational Model' topic for using these test modes. The encoding of the Test Mode bits for a USB2 protocol port are: 0: Test mode not enabled 1: Test J_STATE 2: Test K_STATE 3: Test SE0_NAK 4: Test Packet 5: Test FORCE_ENABLE 6-14: Reserved. 15: Port Test Control Error.
27-17 —	Reserved
16 HLE	Hardware LPM Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
15-8 L1DSLOT	L1 Device Slot For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
7-4 BESL	Best Effort Service Latency System software sets this field to indicate to the recipient device how long the xHC will drive resume if it initiates an exit from L1.
3 RWE	Remote Wake Enable

Table continues on the next page...

Field	Function
	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
2-0 L1S	L1 Status For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.25 USB2 Port Link Information Register (PORTLI_20)

11.2.5.1.25.1 Offset

Register	Offset
PORTLI_20	428h

11.2.5.1.25.2 Function

Port Link Information Register Programming this field with random data will cause side effect i.e. Register Access will fail (Timeout) if the pipe clock is not running or reset is asserted . Bit Bash register testing is not recommended

11.2.5.1.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.25.4 Fields

Field	Function
31-0 —	Reserved

11.2.5.1.26 USB2 Port Hardware LPM Control Register (PORTHLPMC_20)

11.2.5.1.26.1 Offset

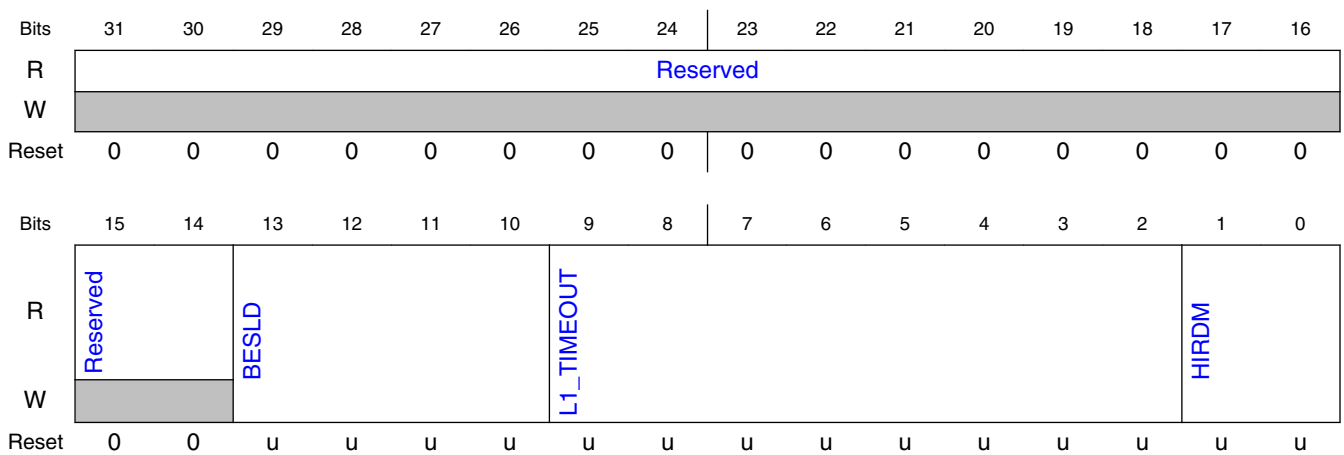
Register	Offset
PORTHLPMC_20	42Ch

11.2.5.1.26.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Reset Mask:0xFFFFC000

11.2.5.1.26.3 Diagram



11.2.5.1.26.4 Fields

Field	Function
31-14 —	Reserved
13-10 BESLD	Best Effort Service Latency Deep System software sets this field to indicate to the recipient device how long the xHC will drive resume on an exit from U2.
9-2 L1_TIMEOUT	L1 Timeout For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus(USB) Specification 3.0.

Table continues on the next page...

Field	Function
1-0 HIRDM	Host Initiated Resume Duration Mode

11.2.5.1.27 USB3 Port Status and Control Register (PORTSC_30)

11.2.5.1.27.1 Offset

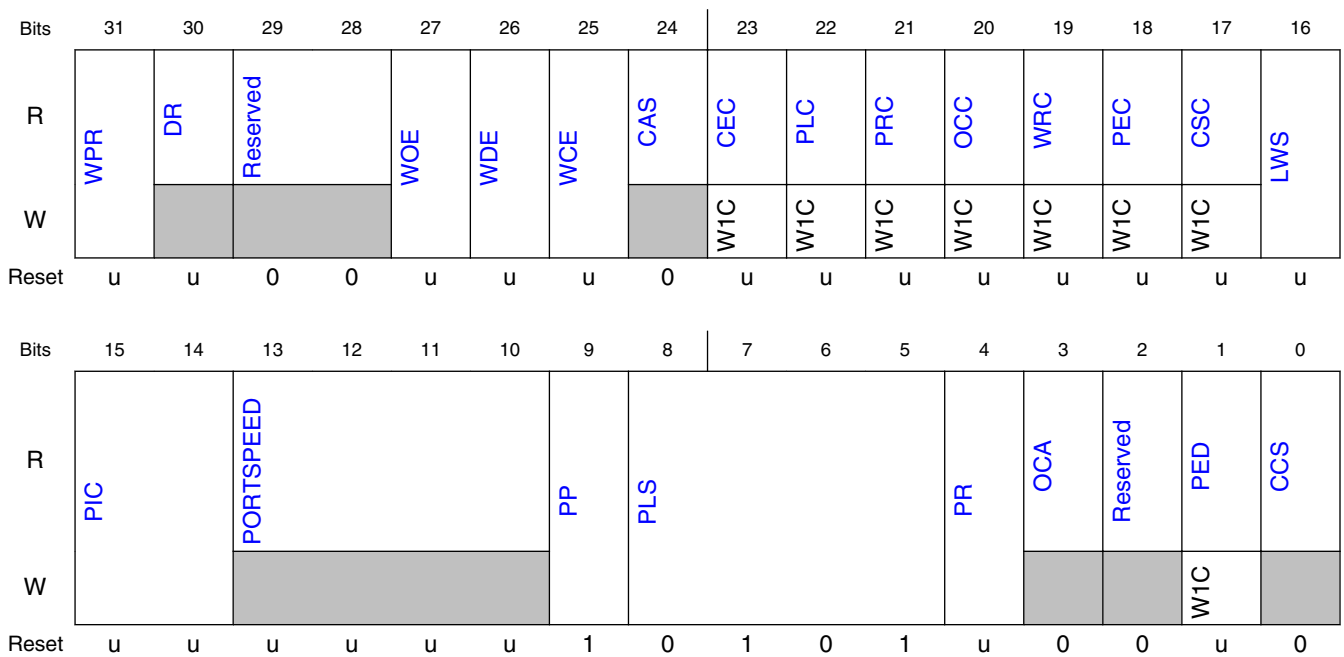
Register	Offset
PORTSC_30	430h

11.2.5.1.27.2 Function

The PORTSC Register Access fails (Timeout) if the UTMI/ULPI clock is not running or one of the following bits is asserted.

Reset Mask:0x310003ED

11.2.5.1.27.3 Diagram



11.2.5.1.27.4 Fields

Field	Function
31 WPR	Warm Port Reset For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
30 DR	Device Removable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
29-28 —	Reserved
27 WOE	Wake on Overcurrent Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
26 WDE	Wake on Disconnect Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
25 WCE	Wake on Connect Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
24 CAS	Cold Attach Status For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
23 CEC	Port Config Error Change For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
22 PLC	Port Link State Change For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
21 PRC	Port Reset Change For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
20 OCC	Over Current Change For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
19 WRC	Warm Port Reset Change For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
18 PEC	Port Enabled/Disabled Change For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
17 CSC	Connect Status Change For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
16	Port Link State Write Strobe

Table continues on the next page...

Field	Function
LWS	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
15-14 PIC	Port Indicator Control For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
13-10 PORTSPEED	Port Speed For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
9 PP	Port Power For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
8-5 PLS	Port Link State For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
4 PR	Port Reset PR set_register_field_attribute DWC_usb3_map/DWC_usb3_block_Host_Cntrl_Port_Reg_Set/PORTSC_30_REGS/PORTSC_30/PR VolatileMemory 1 Programming this field with random data will cause side effect.
3 OCA	Over Current Active For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
2 —	Reserved
1 PED	Port Enabled/Disabled For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
0 CCS	Current Connect Status For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.28 USB3 Port Power Management Status and Control Register (PORTPMSC_30)

11.2.5.1.28.1 Offset

Register	Offset
PORTPMSC_30	434h

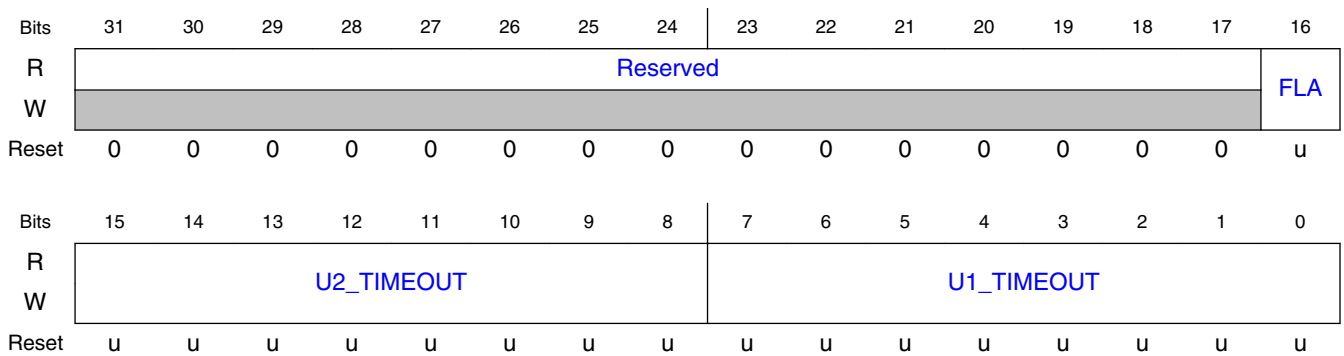
11.2.5.1.28.2 Function

USB3 Port Power Management Status and Control Register Bit Definitions

This register is in the Aux Power well.

Reset Mask:0xFFFE0000

11.2.5.1.28.3 Diagram



11.2.5.1.28.4 Fields

Field	Function
31-17 —	Reserved
16 FLA	Force Link PM Accept For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
15-8 U2_TIMEOUT	U2 Timeout For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
7-0 U1_TIMEOUT	U1 Timeout For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.29 USB3 Port Link Information Register (PORTLI_30)

11.2.5.1.29.1 Offset

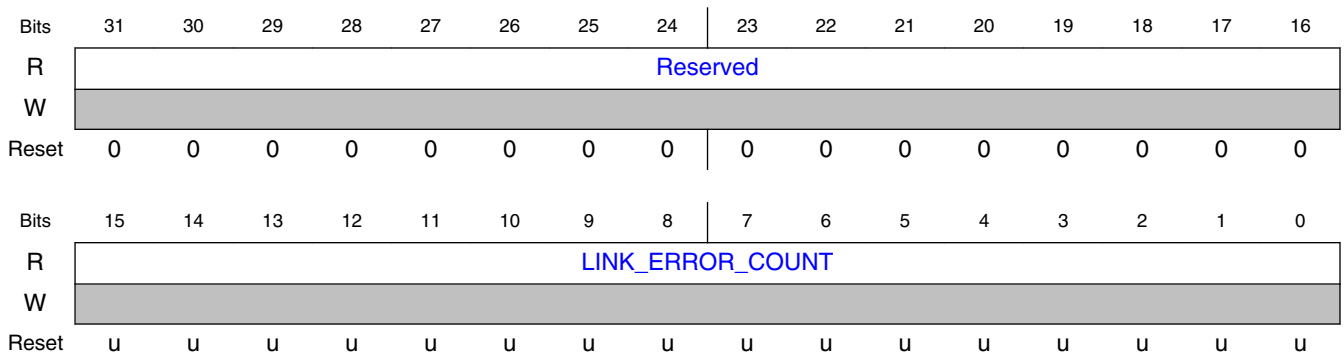
Register	Offset
PORTLI_30	438h

11.2.5.1.29.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Reset Mask:0xFFFF0000

11.2.5.1.29.3 Diagram



11.2.5.1.29.4 Fields

Field	Function
31-16 —	Reserved
15-0 LINK_ERROR_COUNT	Link Error Count For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.30 USB3 Port Hardware LPM Control Register (PORTHLPMC_30)

11.2.5.1.30.1 Offset

Register	Offset
PORTHLPMC_30	43Ch

11.2.5.1.30.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.30.4 Fields

Field	Function
31-0 —	Reserved

11.2.5.1.31 Microframe Index Register (MFINDEX)

11.2.5.1.31.1 Offset

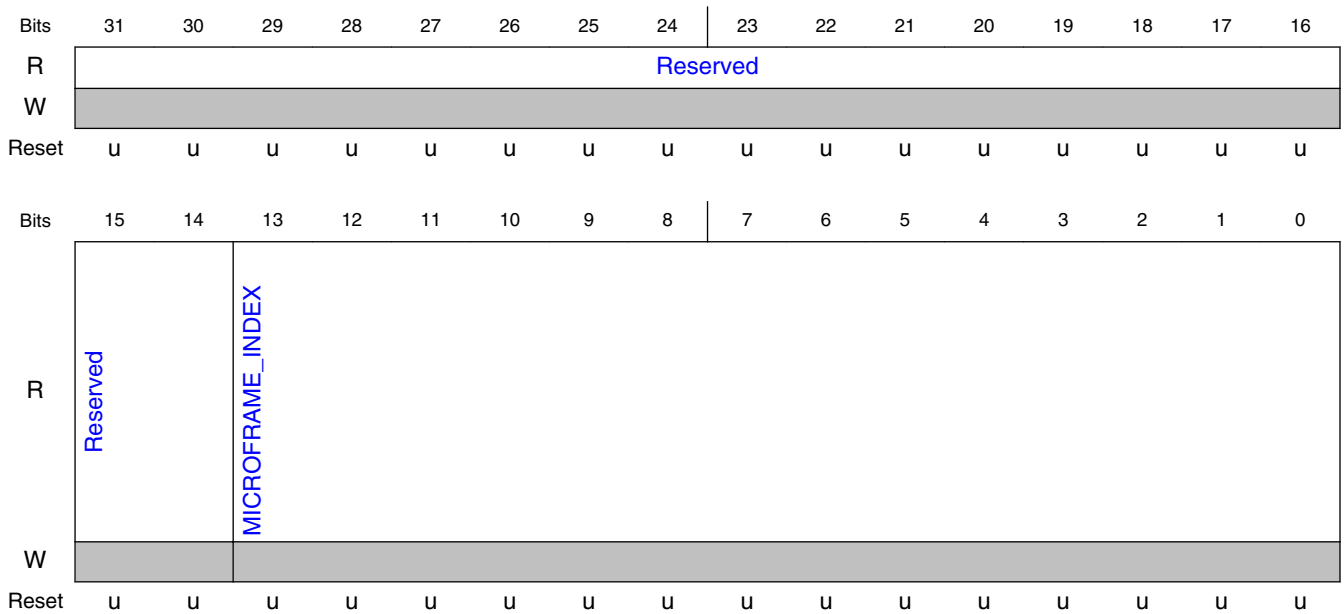
Register	Offset
MFINDEX	440h

11.2.5.1.31.2 Function

See the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Reset Mask:0xFFFFC000

11.2.5.1.31.3 Diagram



11.2.5.1.31.4 Fields

Field	Function
31-14 —	Reserved
13-0 MICROFRAME_INDEX	Microframe Index For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.32 Interrupter Management Register (IMAN)

11.2.5.1.32.1 Offset

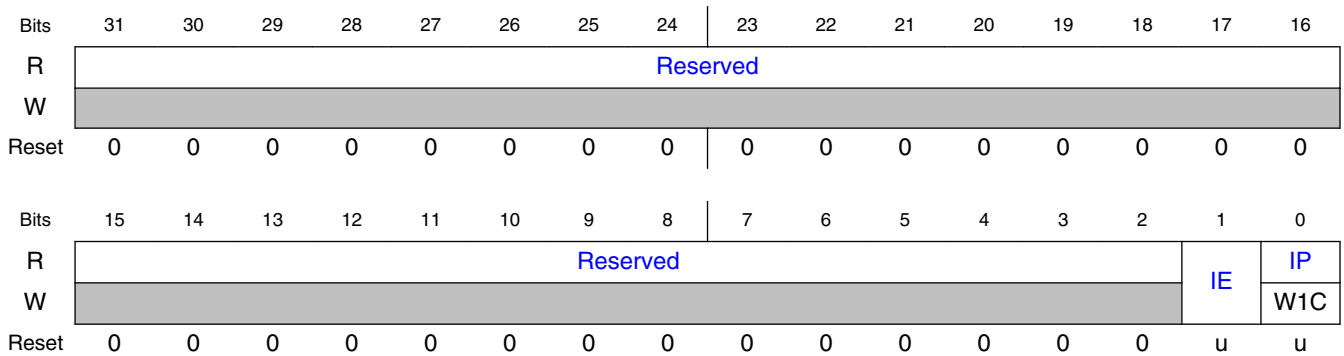
Register	Offset
IMAN	460h

11.2.5.1.32.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Reset Mask:0xFFFFFFFC

11.2.5.1.32.3 Diagram



11.2.5.1.32.4 Fields

Field	Function
31-2 —	Reserved
1 IE	Interrupt Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
0 IP	Interrupt Pending For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.33 Interrupter Moderation Register (IMOD)

11.2.5.1.33.1 Offset

Register	Offset
IMOD	464h

11.2.5.1.33.2 Function

The software may use this register to pace (or even out) the delivery of interrupts to the host CPU.

11.2.5.1.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IMODC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMODI															
Reset	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0

11.2.5.1.33.4 Fields

Field	Function
31-16 IMODC	Interrupt Moderation Counter For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
15-0 IMODI	Interrupt Moderation Interval For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.34 Event Ring Segment Table Size Register (ERSTSZ)

11.2.5.1.34.1 Offset

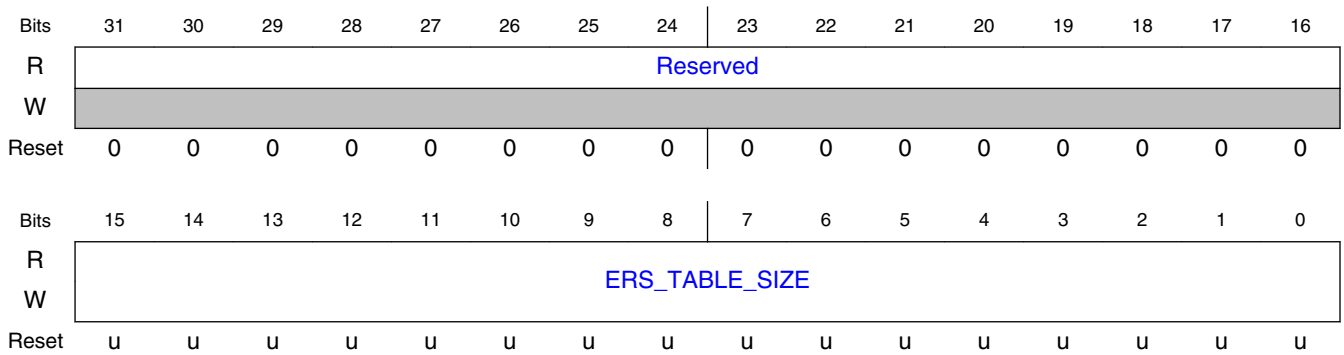
Register	Offset
ERSTSZ	468h

11.2.5.1.34.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Reset Mask:0xFFFF0000

11.2.5.1.34.3 Diagram



11.2.5.1.34.4 Fields

Field	Function
31-16 —	Reserved
15-0 ERS_TABLE_SIZE	Event Ring Segment Table Size For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.35 Event Ring Segment Table Base Address Register - Low (ERSTBA_LO)

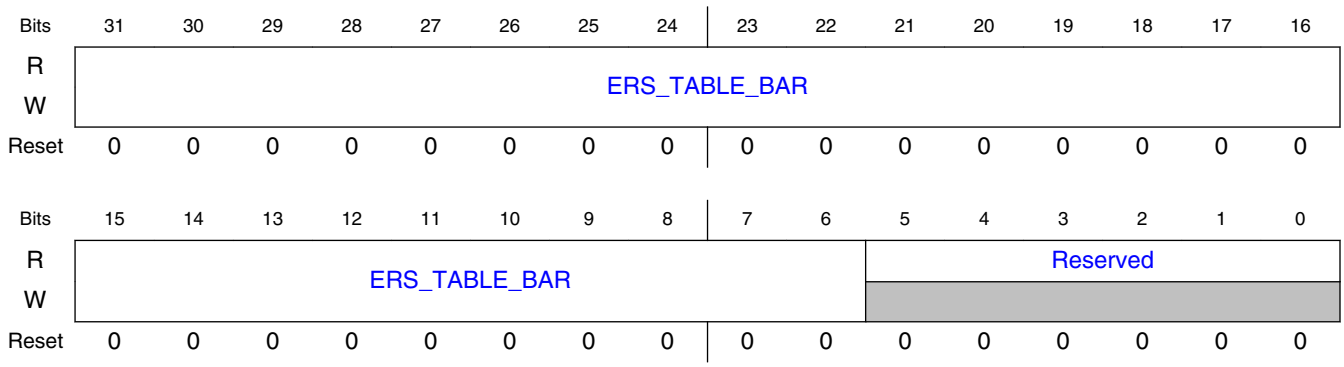
11.2.5.1.35.1 Offset

Register	Offset
ERSTBA_LO	470h

11.2.5.1.35.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.35.3 Diagram



11.2.5.1.35.4 Fields

Field	Function
31-6 ERS_TABLE_BAR	Event Ring Segment Table Base Address For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
5-0 —	Reserved

11.2.5.1.36 Event Ring Segment Table Base Address Register -Low (ERSTBA_HI)

11.2.5.1.36.1 Offset

Register	Offset
ERSTBA_HI	474h

11.2.5.1.36.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.36.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERS_TABLE_BAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERS_TABLE_BAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.36.4 Fields

Field	Function
31-0 ERS_TABLE_BAR	Event Ring Segment Table Base Address For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.37 Event Ring Dequeue Register - Low (ERDP_LO)

11.2.5.1.37.1 Offset

Register	Offset
ERDP_LO	478h

11.2.5.1.37.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.2.5.1.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERD_PNTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERD_PNTR												EHB	DESI		
W													W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.37.4 Fields

Field	Function
31-4 ERD_PNTR	Event Ring Dequeue Pointer For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
3 EHB	Event Handler Busy For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
2-0 DESI	Dequeue ERST Segment Index For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.38 Event Ring Dequeue Register - Low (ERDP_HI)

11.2.5.1.38.1 Offset

Register	Offset
ERDP_HI	47Ch

11.2.5.1.38.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.2.5.1.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERD_PNTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERD_PNTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.38.4 Fields

Field	Function
31-0 ERD_PNTR	Event Ring Dequeue Pointer For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.39 Doorbell Register (DB)

11.2.5.1.39.1 Offset

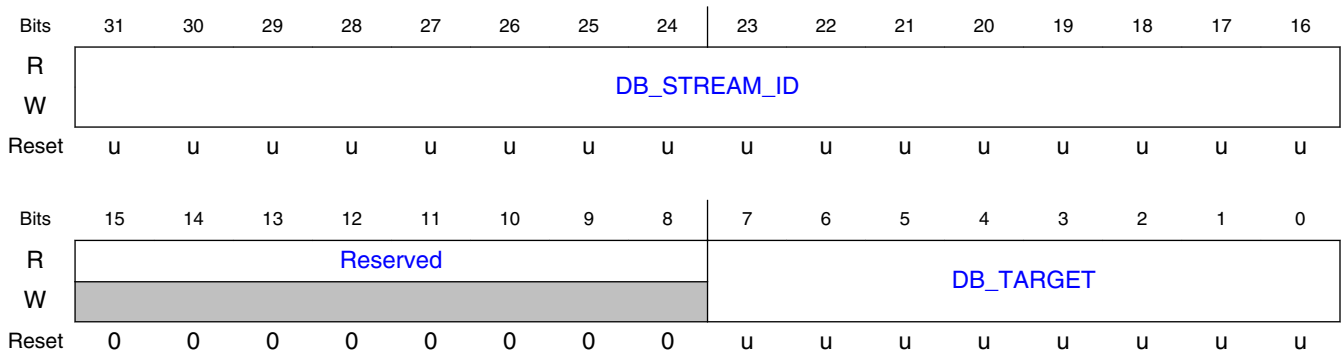
Register	Offset
DB	480h

11.2.5.1.39.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

Reset Mask:0xFF00

11.2.5.1.39.3 Diagram



11.2.5.1.39.4 Fields

Field	Function
31-16 DB_STREAM_ID	Doorbell Stream ID For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
15-8 —	Reserved
7-0 DB_TARGET	Doorbell Target For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.40 USB Legacy Support Capability Register (USBLEGSUP)

11.2.5.1.40.1 Offset

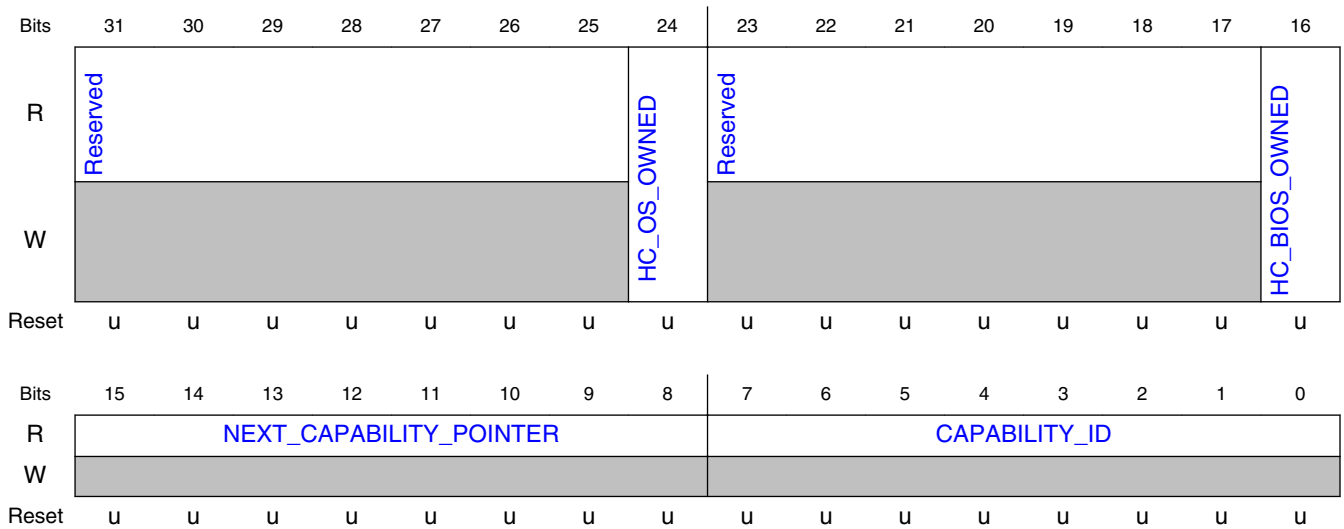
Register	Offset
USBLEGSUP	880h

11.2.5.1.40.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Reset Mask:0x7E7E0000

11.2.5.1.40.3 Diagram



11.2.5.1.40.4 Fields

Field	Function
31-25 —	Reserved
24 HC_OS_OWNED	HC OS Owned Semaphore For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
23-17 —	Reserved
16 HC_BIOS_OWNED	HC BIOS Owned Semaphore For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
15-8 NEXT_CAPABILITY_POINTER	Next Capability Pointer For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
7-0 CAPABILITY_ID	Capability ID set_register_field_attribute DWC_usb3_map/DWC_usb3_block_HC_Extended_Capability_Register/ USBLEGSUP/CAPABILITY_ID RegisterResetValue 0x1 For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.41 USB Legacy Support Control/Status Register (USBLEGCTLSTS)

11.2.5.1.41.1 Offset

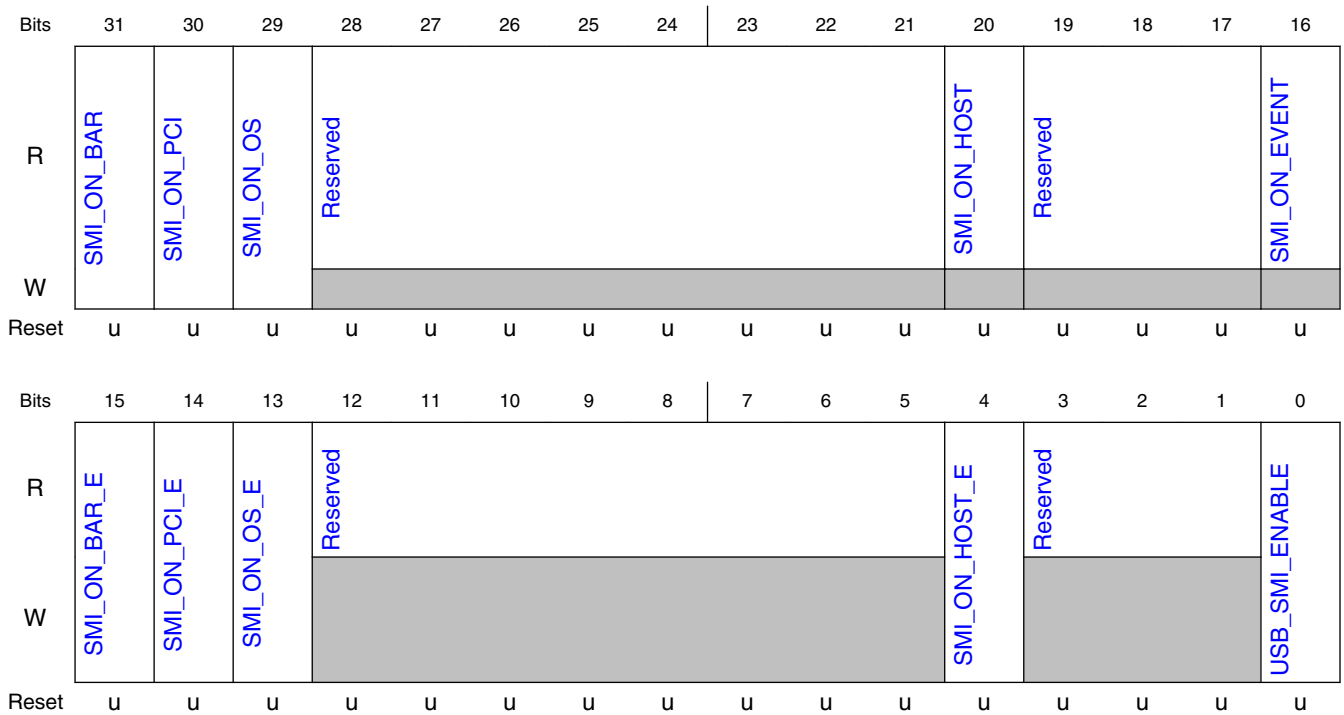
Register	Offset
USBLEGCTLSTS	884h

11.2.5.1.41.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Reset Mask:0x1FEE1FEE

11.2.5.1.41.3 Diagram



11.2.5.1.41.4 Fields

Field	Function
31 SMI_ON_BAR	SMI on BAR For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
30 SMI_ON_PCI	SMI on PCI Command

Table continues on the next page...

Field	Function
	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
29 SMI_ON_OS	SMI on Ownership Change For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
28-21 —	Reserved
20 SMI_ON_HOST	SMI on Host System Error For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
19-17 —	Reserved
16 SMI_ON_EVENT	SMI on Event Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
15 SMI_ON_BAR_E	SMI on BAR Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
14 SMI_ON_PCI_E	SMI on PCI Command Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
13 SMI_ON_OS_E	SMI on OS Ownership Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
12-5 —	Reserved
4 SMI_ON_HOST_E	SMI on Host Sysytem Error Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
3-1 —	Reserved
0 USB_SMI_ENABLE	USB SMI Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.42 USB2 xHCI Supported Protocol Capability Data Word 0 Register (SUPTPRT2_DW0)

11.2.5.1.42.1 Offset

Register	Offset
SUPTPRT2_DW0	890h

11.2.5.1.42.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.42.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR_REVISION								MINOR_REVISION							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NEXT_CAPABILITY_POINTER								CAPABILITY_ID							
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0

11.2.5.1.42.4 Fields

Field	Function
31-24 MAJOR_REVISION	Major Revision For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
23-16 MINOR_REVISION	Minor Revision For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
15-8 NEXT_CAPABILITY_POINTER	Next Capability Pointer For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
7-0 CAPABILITY_ID	Capability ID For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.43 USB2 xHCI Supported Protocol Capability Data Word 1 Register (SUPTPRT2_DW1)

11.2.5.1.43.1 Offset

Register	Offset
SUPTPRT2_DW1	894h

11.2.5.1.43.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.43.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NAME_STRING															
W																
Reset	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NAME_STRING															
W																
Reset	0	1	0	1	0	0	1	1	0	1	0	1	0	1	0	1

11.2.5.1.43.4 Fields

Field	Function
31-0 NAME_STRING	Name String For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.44 USB2 xHCI Supported Protocol Capability Data Word 2 Register (SUPTPRT2_DW2)

11.2.5.1.44.1 Offset

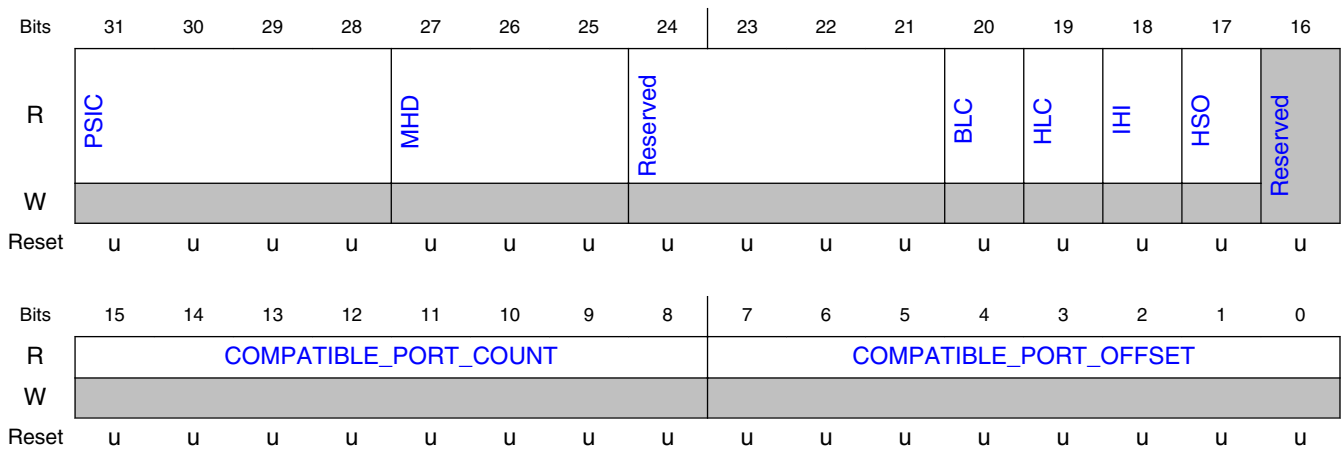
Register	Offset
SUPTPRT2_DW2	898h

11.2.5.1.44.2 Function

For a description of other register fields, see section 7.2 of the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Reset Mask:0x1E00000

11.2.5.1.44.3 Diagram



11.2.5.1.44.4 Fields

Field	Function
31-28 PSIC	Protocol Speed ID Count For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
27-25 MHD	Hub Depth For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
24-21 —	Reserved
20 BLC	BESL LPM Capability When this bit is set to: - 1: The ports described by this xHCI Supported Protocol Capability applies BESL timing to the BESL and BESLD fields of the PORTPMSC and PORTHLPMC registers. - 0: The ports described by this xHCI Supported Protocol Capability applies HIRD timing to the BESL and BESLD fields of the PORTPMSC and PORTHLPMC registers.
19 HLC	Compatible Port Offset. Compatible Port Count Refer to Table 7-3 in the DWC Cores SuperSpeed USB 3.0 Controller Databook.
18 IHI	IHI

Table continues on the next page...

Field	Function
17 HSO	HSO
16 —	Reserved
15-8 COMPATIBLE_PORT_COUNT	Compatible Port Count For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
7-0 COMPATIBLE_PORT_OFFSET	Compatible Port Offset For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.45 USB2 xHCI Supported Protocol Capability Data Word 3 Register (SUPTPRT2_DW3)

11.2.5.1.45.1 Offset

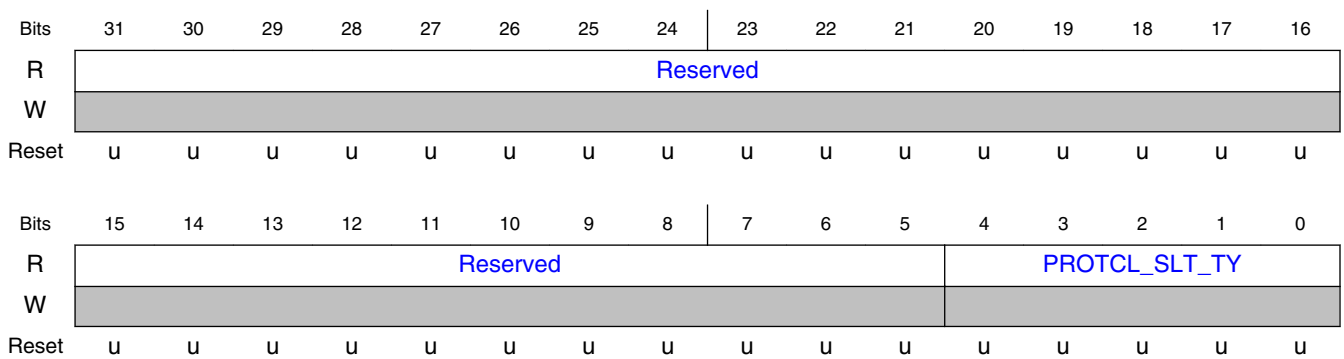
Register	Offset
SUPTPRT2_DW3	89Ch

11.2.5.1.45.2 Function

For a description of other register fields, see section 7.2 of the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Reset Mask:0xFFFFFFE0

11.2.5.1.45.3 Diagram



11.2.5.1.45.4 Fields

Field	Function
31-5 —	Reserved
4-0 PROTCL_SLT_TY	Protocol Slot Type For a description of other register fields, see section 7.2 of the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.46 USB3 xHCI Supported Protocol Capability Data Word 0 Register (SUPTPRT3_DW0)

11.2.5.1.46.1 Offset

Register	Offset
SUPTPRT3_DW0	8A0h

11.2.5.1.46.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.46.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR_REVISION								MINOR_REVISION							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NEXT_CAPABILITY_POINTER								CAPABILITY_ID							
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0

11.2.5.1.46.4 Fields

Field	Function
31-24	Major Revision

Table continues on the next page...

Field	Function
MAJOR_REVISION	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
23-16 MINOR_REVISION	Minor Revision For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
15-8 NEXT_CAPABILITY_POINTER	Next Capability Pointer For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
7-0 CAPABILITY_ID	Capability ID For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.47 USB3 xHCI Supported Protocol Capability Data Word 1 Register (SUPTPRT3_DW1)

11.2.5.1.47.1 Offset

Register	Offset
SUPTPRT3_DW1	8A4h

11.2.5.1.47.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.47.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NAME_STRING															
W																
Reset	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NAME_STRING															
W																
Reset	0	1	0	1	0	0	1	1	0	1	0	1	0	1	0	1

11.2.5.1.47.4 Fields

Field	Function
31-0 NAME_STRING	Name String For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.48 USB3 xHCI Supported Protocol Capability Data Word 2 Register (SUPTPRT3_DW2)

11.2.5.1.48.1 Offset

Register	Offset
SUPTPRT3_DW2	8A8h

11.2.5.1.48.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.2.5.1.48.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PSIC				MHD				Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COMPATIBLE_PORT_COUNT								COMPATIBLE_PORT_OFFSET							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0

11.2.5.1.48.4 Fields

Field	Function
31-28 PSIC	Protocol Speed ID Count For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
27-25	Hub Depth

Table continues on the next page...

Field	Function
MHD	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
24-16 —	Reserved
15-8 COMPATIBLE_PORT_COUNT	Compatible Port Count For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
7-0 COMPATIBLE_PORT_OFFSET	Compatible Port Offset For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.49 USB3 xHCI Supported Protocol Capability Data Word 3 Register (SUPTPRT3_DW3)

11.2.5.1.49.1 Offset

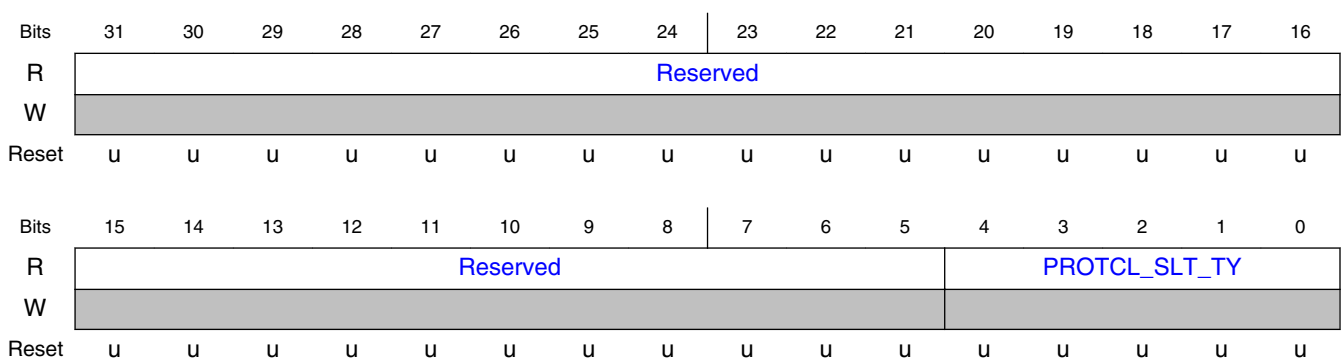
Register	Offset
SUPTPRT3_DW3	8ACh

11.2.5.1.49.2 Function

For a description of other register fields, see section 7.2 of the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Reset Mask:0xFFFFFFE0

11.2.5.1.49.3 Diagram



11.2.5.1.49.4 Fields

Field	Function
31-5 —	Reserved
4-0 PROTCL_SLT_TY	Protocol Slot Type For a description of other register fields, see section 7.2 of the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.2.5.1.50 Global SoC Bus Configuration Register 0 (GSBUSCFG0)

11.2.5.1.50.1 Offset

Register	Offset
GSBUSCFG0	C100h

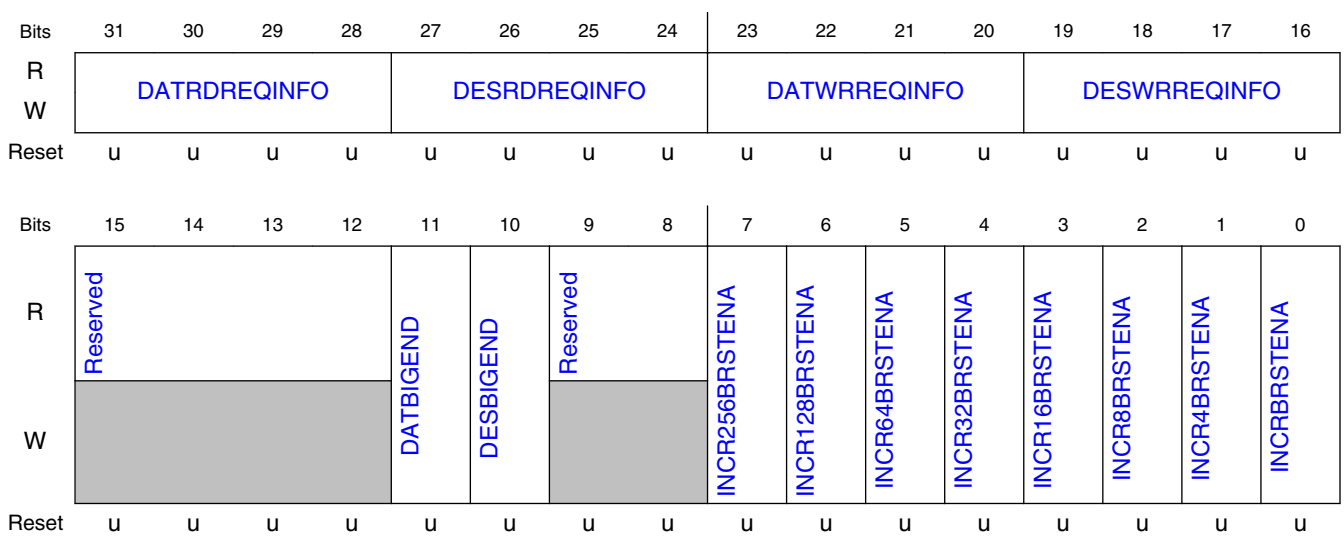
11.2.5.1.50.2 Function

This register configures system bus DMA options for the master bus, which may be configured as AHB, AXI, or Native.

Reset Mask:0xF300

Reset Default Value:0x1

11.2.5.1.50.3 Diagram



11.2.5.1.50.4 Fields

Field	Function
31-28 DATRDREQINFO	DATRDREQINFO AHB-prot/AXI-cache/OCP-ReqInfo for Data Read (DatRdReqInfo) Input to BUS-GM.
27-24 DESRDREQINFO	DESRDREQINFO AHB-prot/AXI-cache/OCP-ReqInfo for Descriptor Read (DesRdReqInfo). Input to BUS-GM.
23-20 DATWRREQINFO	DATWRREQINFO AHB-prot/AXI-cache/OCP-ReqInfo for Data Write (DatWrReqInfo). Input to BUS-GM.
19-16 DESWRREQINFO	DESWRREQINFO AHB-prot/AXI-cache/OCP-ReqInfo for Descriptor Write (DesWrReqInfo) Input to BUS-GM.
15-12 —	Reserved for future use
11 DATBIGEND	Data Access is Big Endian This bit controls the endian mode for data accesses. - Little-endian (default); - Big-endian; In big-endian mode, DMA access (both read and write) for packet data a Byte Invariant Big-Endian mode (see "Little-Endian and Big-Endian" section in the DWC SuperSpeed USB 3.0 Controller User Guide). Note: Since AXI requires byte invariant endianness, setting DescBigend and DatBigEnd to one causes an address invariant transform to be applied, which is not appropriate. See section 9.3 and 9.4 of the AMBA AXI Specification. Hence for an AXI master (DWC_USB3_MBUS_TYPE=1), this bit must be set to zero.
10 DESBIGEND	Descriptor Access is Big Endian This bit controls the endian mode for descriptor accesses. - Little-endian (default) - Big-endian In big-endian mode, DMA access (both read and write) for descriptors uses a Byte Invariant Big-Endian mode (see "Little-Endian and Big-Endian" section in the DWC SuperSpeed USB 3.0 Controller User Guide. Data is considered as 'embedded data' in the descriptors in the following cases: - Device mode: The buffer pointer of a Setup TRB points to the Setup TRB itself. - Host mode: The Immediate Data (IDT) bit in a Transfer TRB is set to 1. In device mode, if the system uses different endian modes for descriptor and data, software must not use 'embedded' data. In host mode, if the system uses different endian modes for data and descriptors, the controller treats 'embedded data' as descriptor (not as data) in terms of endian mode handling. If this is not the expectation of the system, the software must manipulate the 'embedded data' accordingly. Note: Since AXI requires byte invariant endianness, setting DescBigend and DatBigEnd to one causes an address invariant transform to be applied, which is not appropriate. See section 9.3 and 9.4 of the AMBA AXI Specification. Hence for an AXI master (DWC_USB3_MBUS_TYPE=1), this bit must be set to zero.
9-8 —	Reserved
7 INCR256BRSTENA	INCR256 Burst Type Enable Input to BUS-GM For the AHB/AXI configuration, if software set this bit to 1, the AHB/AXI master uses INCR to do the 256-beat burst.
6 INCR128BRSTENA	INCR128 Burst Type Enable Input to BUS-GM For the AHB/AXI configuration, if software set this bit to 1, the AHB/AXI master uses INCR to do the 128-beat burst.
5	INCR64 Burst Type Enable Input to BUS-GM

Table continues on the next page...

Field	Function
INCR64BRSTENA	For the AHB/AXI configuration, if software set this bit to 1, the AHB/AXI master uses INCR to do the 64-beat burst.
4 INCR32BRSTENA	INCR32 Burst Type Enable Input to BUS-GM For the AHB/AXI configuration, if software set this bit to 1, the AHB/AXI master uses INCR to do the 32-beat burst.
3 INCR16BRSTENA	INCR16 Burst Type Enable Input to BUS-GM For the AHB/AXI configuration, if software set this bit to '1', the AHB/AXI master uses INCR to do the 16-beat burst.
2 INCR8BRSTENA	INCR8 Burst Type Enable Input to BUS-GM For the AHB/AXI configuration, if software set this bit to "1", the AHB/AXI master uses INCR to do the 8-beat burst.
1 INCR4BRSTENA	INCR4 Burst Type Enable Input to BUS-GM For the AXI configuration, when this bit is enabled the controller is allowed to do bursts of beat length 1, 2, 3, and 4. It is highly recommended that this bit is enabled to prevent descriptor reads and writes from being broken up into separate transfers.
0 INCRBRSTENA	Undefined Length INCR Burst Type Enable Input to BUS-GM This bit determines the set of burst lengths the master interface uses. It works in conjunction with the GSBUSCFG0[7:1] enables (INCR256/128/64/32/16/8/4). 0: INCRX burst mode HBURST (for AHB configurations) and ARLEN/AWLEN (for AXI configurations) do not use INCR except in case of non-aligned burst transfers. In the case of address-aligned transfers, they use only the following burst lengths: - 1 - 4 (if GSBUSCFG0.INCR4BrstEna = 1) - 8 (if GSBUSCFG0.INCR8BrstEna = 1) - 16 (if GSBUSCFG0.INCR16BrstEna = 1) - 32 (if GSBUSCFG0.INCR32BrstEna = 1) - 64 (if GSBUSCFG0.INCR64BrstEna = 1) - 128 (if GSBUSCFG0.INCR128BrstEna = 1) - 256 (if GSBUSCFG0.INCR256BrstEna = 1) Note: - In case of non-address-aligned transfers, INCR may get generated at the beginning and end of the transfers to align the address boundaries, even though INCR is disabled. - In AHB mode, if INCRX burst mode is enabled, but none of the supported INCRx bursts bits are enabled, then the controller will perform (undefined length) INCR bursts. 1: INCR (undefined length) burst mode - AHB configurations: HBURST uses SINGLE or INCR of any length with handling 1KB boundary breakup. - AXI configurations: ARLEN/AWLEN uses any length less than or equal to the largest-enabled burst length of INCR32/64/128/256. For cache line-aligned applications, this bit is typically set to 0 to ensure that the master interface uses only power-of-2 burst lengths (as enabled via GSBUSCFG0[7:0]).

11.2.5.1.51 Global SoC Bus Configuration Register 1 (GSBUSCFG1)

11.2.5.1.51.1 Offset

Register	Offset
GSBUSCFG1	C104h

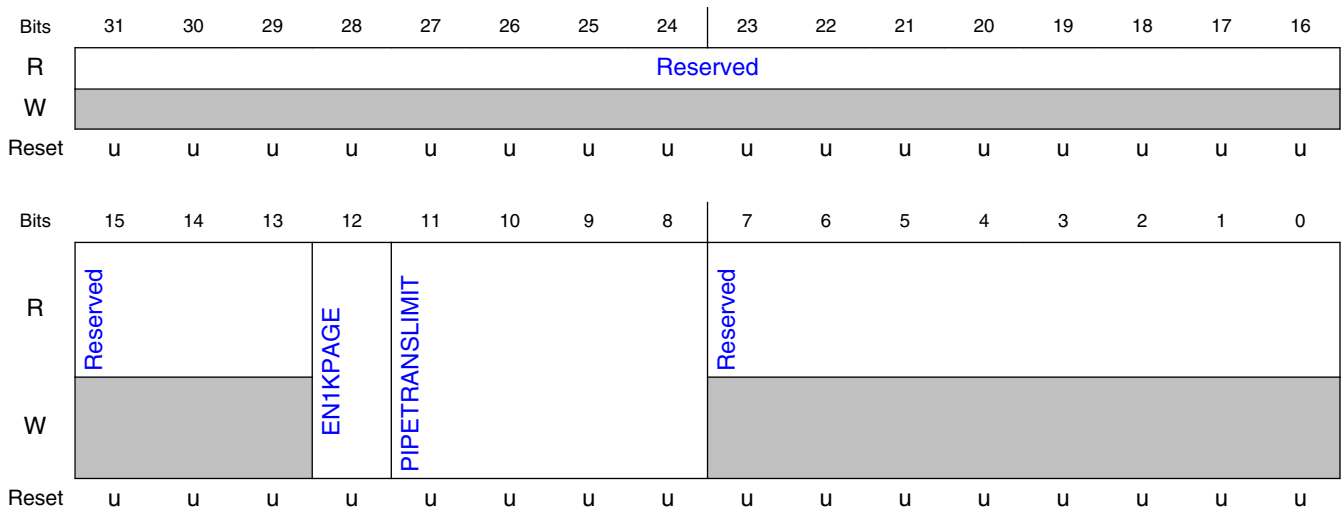
11.2.5.1.51.2 Function

xHCI Register Power-On Value: If you are using a standard xHCI host driver, make sure to set the register's power-on value during coreConsultant configuration (DWC_USB3_GSBUSCFG1_INIT parameter) because the standard xHCI driver does not access this register.

Reset Mask:0xFFFFE0FF

Reset Default Value:0x300

11.2.5.1.51.3 Diagram



11.2.5.1.51.4 Fields

Field	Function
31-13 —	Reserved
12 EN1KPAGE	1k Page Boundary Enable By default (this bit is disabled) the AXI breaks transfers at the 4k page boundary. When this bit is enabled, the AXI master (DMA data) breaks transfers at the 1k page boundary.
11-8 PIPETRANSLIMIT	AXI Pipelined Transfers Burst Request Limit The field controls the number of outstanding pipelined transfer requests the AXI master pushes to the AXI slave. When the AXI master reaches this limit, it does not make any more requests on the AXI ARADDR and AWADDR buses until the associated data phases complete. This field is encoded as follows: - 'h0: 1 request - 'h1: 2 requests - 'h2: 3 requests - 'h3: 4 requests - ... - 'hF: 16 requests
7-0 —	Reserved

11.2.5.1.52 Global Tx Threshold Control Register (GTXTHRCFG)

11.2.5.1.52.1 Offset

Register	Offset
GTXTHRCFG	C108h

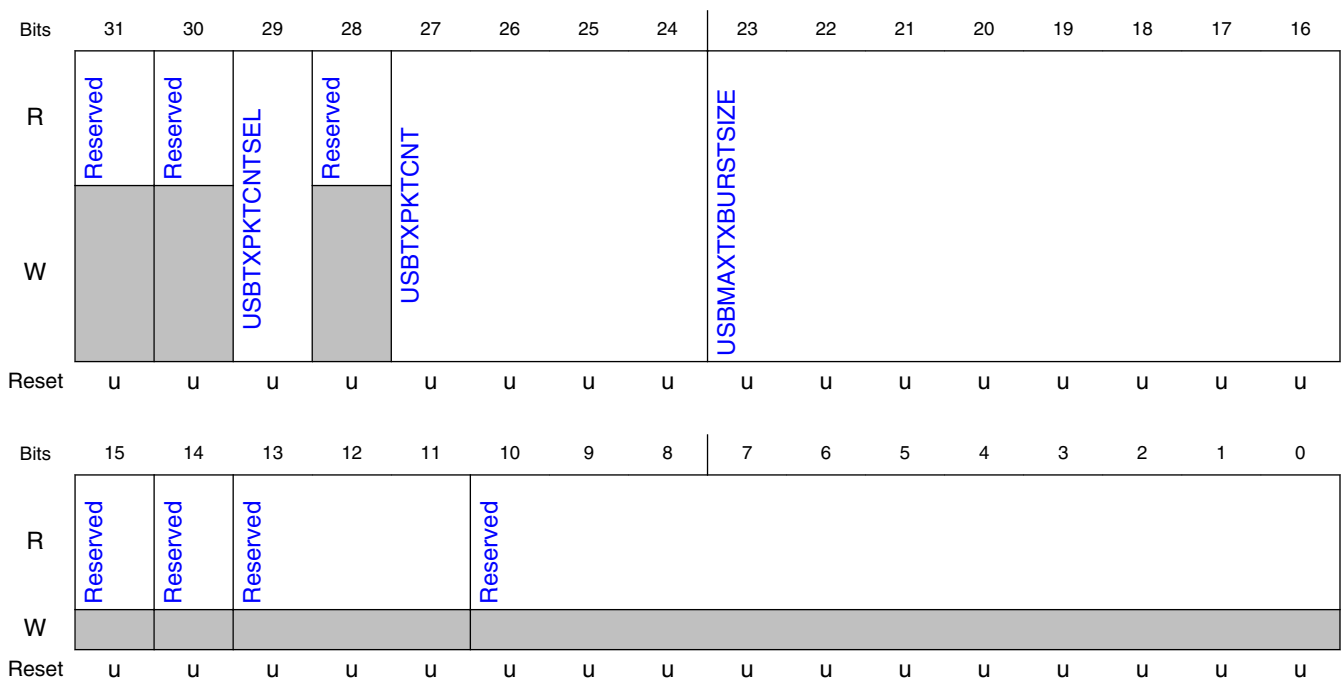
11.2.5.1.52.2 Function

All bits in the GTXTHRCFG register are valid only in host mode. This register is not applicable for debug target and USB 2.0-only mode.

Reset Mask:0xD000FFFF

Reset Default Value:0x0

11.2.5.1.52.3 Diagram



11.2.5.1.52.4 Fields

Field	Function
31	Reserved

Table continues on the next page...

Field	Function
—	
30 —	Reserved
29 USBTXPKTCNTSEL	USB Transmit Packet Count Enable This field enables/disables the USB transmission multi-packet thresholding: - 0: USB transmission multi-packet thresholding is disabled; the core can start transmission on the USB after the entire (one full) packet has been fetched into the corresponding TXFIFO. - 1: USB transmission multi-packet thresholding is enabled. The core can only start transmission on the USB after USB Transmit Packet Count amount of packets for the USB transaction (burst) are already in the corresponding TXFIFO. This mode is valid in both host and device modes. It is only used for SuperSpeed operation.
28 —	Reserved
27-24 USBTXPKTCNT	USB Transmit Packet Count This field specifies the number of packets that must be in the TXFIFO before the core can start transmission for the corresponding USB transaction (burst). This field is only valid when the USB Transmit Packet Count Enable field is set to one. Valid values are from 1 to 15. Note: - In device mode, if device controller does not have the TRBs for the number of packets or if it cannot fetch the TRBs because of high latency or switching between other endpoints, then it does not wait for the threshold number of packets. The threshold number of packets will be honored only when the TRBs are available in the controller for the number of packets before it starts the data fetch. - This field must be less than or equal to the USB Maximum TX Burst Size field.
23-16 USBMAXTXBURSTSIZE	USB Maximum TX Burst Size When UsbTxPktCntSel is one, this field specifies the Maximum Bulk OUT burst the core can do. When the system bus is slower than the USB, TX FIFO can underrun during a long burst. User can program a smaller value to this field to limit the TX burst size that the core can do. Host mode: It only applies to SS Bulk, Isochronous, and Interrupt OUT endpoints. Device mode: This value is not used in device mode, but users need to program a value when using the TX threshold feature to make sure that the value programmed in UsbTxPktCnt is less than this value. Valid values are from 1 to 16.
15 —	Reserved
14 —	Reserved1 Register field must write only 0 by the application. The read value must be treated as X (unknown).
13-11 —	Reserved The register field must write only 0 by the application. The read value must be treated as X (unknown).
10-0 —	Reserved for future use

11.2.5.1.53 Global Rx Threshold Control Register (GRXTHRCFG)

11.2.5.1.53.1 Offset

Register	Offset
GRXTHRCFG	C10Ch

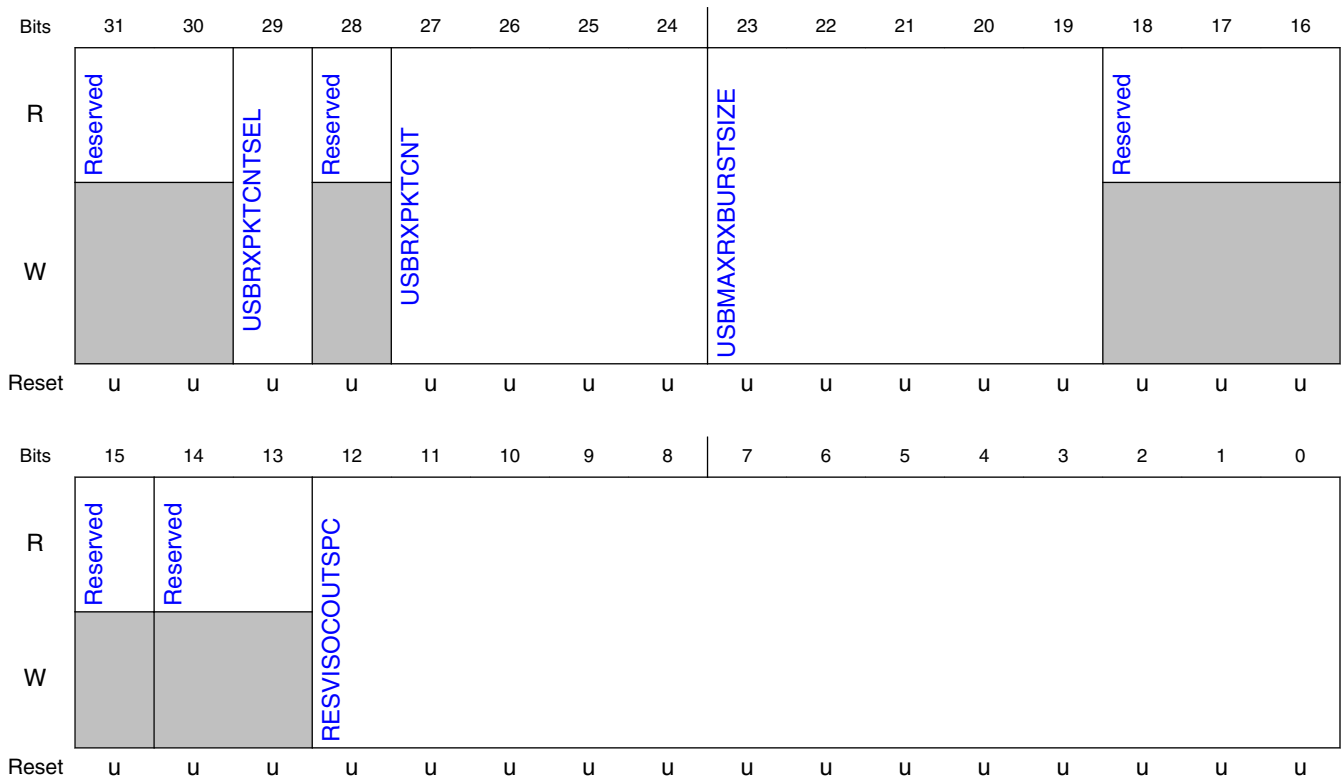
11.2.5.1.53.2 Function

This register is not applicable for debug target and USB 2.0-only mode. In a normal case, a Tx burst starts as soon as one packet is prefetched; an Rx burst starts as soon as 1-packet space is available.

Reset Mask:0xD007E000

Reset Default Value:0x0

11.2.5.1.53.3 Diagram



11.2.5.1.53.4 Fields

Field	Function
31-30 —	Reserved
29 USBRXPKTCNTSEL	USB Receive Packet Count Enable This field enables/disables the USB reception multi-packet thresholding: - 0: The core can only start reception on the USB when the RX FIFO has space for at least one packet. - 1: The core can only start reception on the USB when the RX FIFO has space for at least UsbRxPktCnt amount of packets. This

Table continues on the next page...

Field	Function
	mode is valid in both host and device mode. It is only used for SuperSpeed. In device mode, - Setting this bit to 1 also enables the functionality of reporting NUMP in the ACK TP based on the RX FIFO space instead of reporting a fixed NUMP derived from DCFG.NUMP for non-control endpoints. - If you are using external buffer control (EBC) feature, disable this mode by setting UsbRxPktCntSel to 0.
28 —	Reserved
27-24 USBRXPKTCNT	USB Receive Packet Count In host mode, this field specifies the space (in terms of the number of packets) that must be available in the RX FIFO before the core can start the corresponding USB RX transaction (burst). In device mode, this field specifies the space (in terms of the number of packets) that must be available in the RX FIFO before the core can send ERDY for a flow-controlled endpoint. This field is valid only when the USB Receive Packet Count Enable field is set to 1. The valid values for this field are from 1 to 15. Note: This field must be less than or equal to the USB Maximum Receive Burst Size field.
23-19 USBMAXRXBURSTSIZE	USB Maximum Receive Burst Size In host mode, this field specifies the Maximum Bulk IN burst the DWC_usb3 core can perform. When the system bus is slower than the USB, RX FIFO can overrun during a long burst. You can program a smaller value to this field to limit the RX burst size that the core can perform. It only applies to SS Bulk, Isochronous, and Interrupt IN endpoints in the host mode. In device mode, this field specifies the NUMP value that is sent in ERDY for an OUT endpoint. The programmed value should not exceed the RXFIFO size. This field is valid only when UsbRxPktCntSel is one. The valid values for this field are from 1 to 16.
18-16 —	Reserved
15 —	Reserved
14-13 —	Reserved
12-0 RESVISOCOUTSPC	Space Reserved in Rx FIFO for ISOC OUT In host mode, this field is not applicable and must be programmed to 0. In device mode, this value represents the amount of space to be reserved for ISOC OUT packets. The value to be programmed should be chosen so as to ensure that non ISOC packets are not completely dropped. If no space needs to be reserved for ISOC OUT packets, program this field to 0. This field is valid only in device mode. The maximum configurable depth of RX FIFO is 8192. Therefore, this field is 13 bits wide. The value of space reserved is in terms of DWC_USB3_MDWIDTH. For SS, the space reservation is always rounded off to the nearest packet boundary. Therefore, it is always recommended to program a value corresponding to MPS or its multiples. For HS/FS, the space reservation is the actual value. Note: For SS, reserve space for ISOC when the Rx FIFO space can accommodate two MPS or more. Otherwise, this may result in degraded performance for non-ISOC packets. If the space is entirely allocated for ISOC, the non-ISOC packets will be completely dropped. To help you decide during the time of configuring the core, refer to the section "Device-Mode Receive Path" in Chapter 3.

11.2.5.1.54 Global Core Control Register (GCTL)

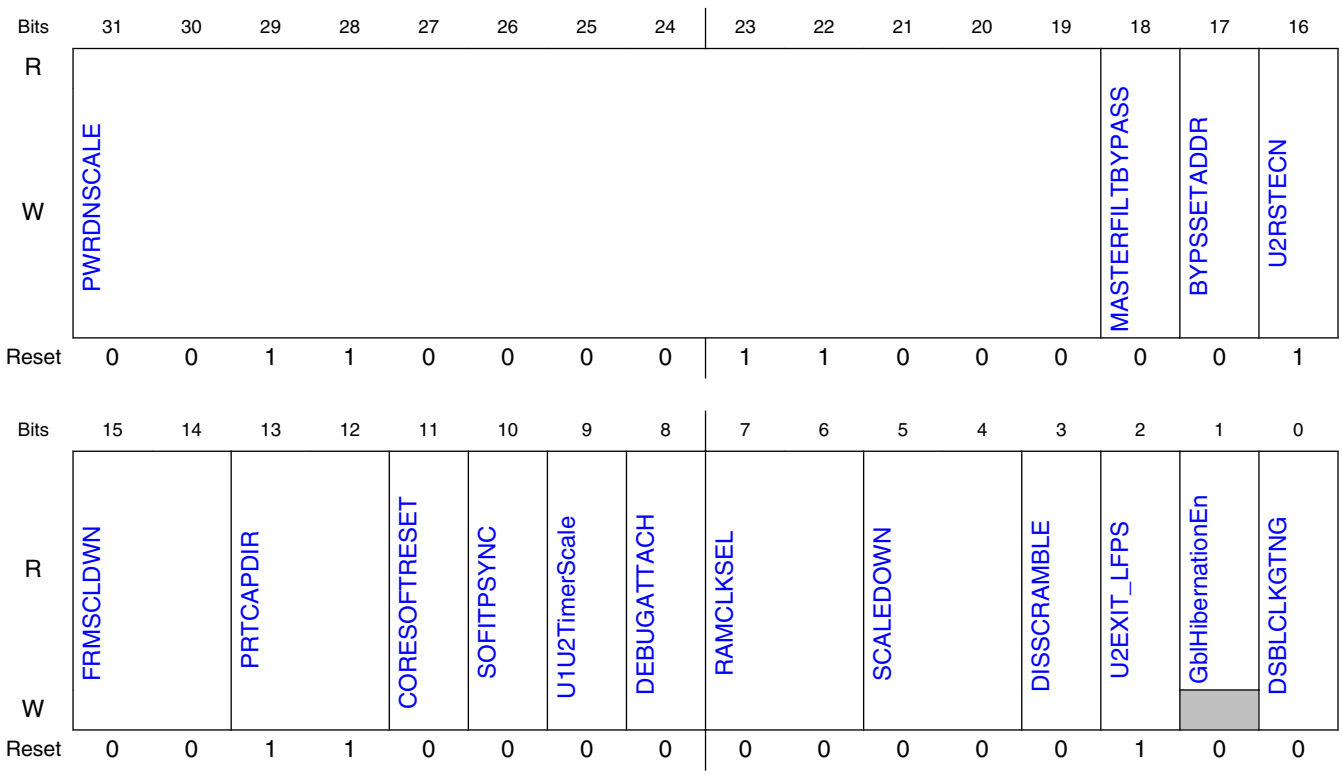
11.2.5.1.54.1 Offset

Register	Offset
GCTL	C110h

11.2.5.1.54.2 Function

Note: When Hibernation is not enabled, you can write any value to GblHibernationEn. It always returns 0 when read.

11.2.5.1.54.3 Diagram



11.2.5.1.54.4 Fields

Field	Function
31-19 PWRDNSCALE	Power Down Scale The USB3 suspend_clk input replaces pipe3_rx_pclk as a clock source to a small part of the USB3 core that operates when the SS PHY is in its lowest power (P3) state, and therefore does not provide a clock. The Power Down Scale field specifies how many suspend_clk periods fit into a 16 kHz clock period. When performing the division, round up the remainder. For example, when using an 8-bit/16-bit/32-bit PHY and 25-MHz Suspend clock, Power Down Scale = 25000 kHz/16 kHz = 13'd1563 (rounder up) Note: - Minimum Suspend clock frequency is 32 kHz - Maximum Suspend clock frequency is 125 MHz The LTSSM uses Suspend clock for 12-ms and 100-ms timers during suspend mode. According to the USB 3.0 specification, the accuracy on these timers is 0% to +50%. - 12 ms + 0~+50% accuracy = 18 ms (Range is 12 ms - 18 ms) - 100 ms + 0~+50% accuracy = 150 ms (Range is 100 ms - 150 ms). The suspend clock accuracy requirement is: - (12,000/62.5) * (GCTL[31:19]) * actual suspend_clk_period must be between 12,000 and 18,000 - (100,0000/62.5) * (GCTL[31:19]) * actual suspend_clk_period must be between 100,000 and 150,000 For example, if your suspend_clk frequency varies from 7.5 MHz

Table continues on the next page...

Field	Function
	to 10.5MHz, then the value needs to programmed is: Power Down Scale = 10500/16 = 657 (rounded up; and fastest frequency used).
18 MASTERFILTBYPASS	Master Filter Bypass When this bit is set to 1'b1, irrespective of the parameter `DWC_USB3_EN_BUS_FILTERS chosen, all the filters in the DWC_usb3_filter module are bypassed. The double synchronizers to mac_clk preceding the filters are also bypassed. For enabling the filters, this bit must be 1'b0.
17 BYPSSETADDR	Bypass SetAddress in Device Mode When BYPSSETADDR bit is set, the device core uses the value in the DCFG[DevAddr] bits directly for comparing the device address in the tokens. For simulation, you can use this feature to avoid sending an actual SET ADDRESS control transfer on the USB, and make the device core respond to a new address. When the xHCI Debug capability is enabled and this bit is set, the Debug Target immediately enters the configured state without requiring the Debug Host to send a SetAddress or SetConfig request. Note: You can set this bit for simulation purposes only. In the actual hardware, this bit must be set to 1'b0.
16 U2RSTECN	If the SuperSpeed connection fails during POLL or LMP exchange, the device connects at non-SS mode. If this bit is set, then device attempts three more times to connect at SS, even if it previously failed to operate in SS mode. For each attempt, the device checks receiver termination eight times. From 2.60a release, this bit controls whether to check for Rx.Detect eight times or one time for every attempt. Device controller on USB 2.0 reset checks for receiver termination eight times per attempt if this bit is set to zero, or only once per attempt if the bit is set to one. Note: This bit is applicable only in device mode.
15-14 FRMSCLDWN	This field scales down device view of a SOF/USOF/ITP duration. For SS/HS mode: - Value of 2'h3 implements interval to be 15.625 us - Value of 2'h2 implements interval to be 31.25 us - Value of 2'h1 implements interval to be 62.5 us - Value of 2'h0 implements interval to be 125us For FS mode, the scale-down value is multiplied by 8. When xHCI Debug Capability is enabled, this field also scales down the MaxPacketSize of the IN and OUT bulk endpoint to allow more traffic during simulation. It can only be changed from a non-zero value during simulation. - 2'h0: 1024 bytes - 2'h1: 512 bytes - 2'h2: 256 bytes - 2'h3: 128 bytes
13-12 PRTCAPDIR	Port Capability Direction 2'b01: for Host configurations - 2'b10: for Device configurations . The sequence for switching modes in DRD configuration is as follows: Switching from Device to Host: 1. Reset the controller using GCTL[11] (CoreSoftReset). 2. Set GCTL[13:12] (PrtCapDir) to 2'b01 (Host mode). 3. Reset the host using USBCMD.HCRESET. 4. Follow the steps in "Initializing Host Registers" section of the databook. Switching from Host to Device: 1. Reset the controller using GCTL[11] (CoreSoftReset). 2. Set GCTL[13:12] (PrtCapDir) to 2'b10 (Device mode). 3. Reset the device by setting DCTL[30] (CSftRst). 4. Follow the steps in "Register Initialization" section of the databook. Programming this field with random data causes the core to keep toggling between the host mode and the device mode . Bit Bash register testing is not recommended.
11 CORESOFTRESET	Core Soft Reset 1'b0 - No soft reset - 1'b1 - Soft reset to core Clears the interrupts and all the CSRs except the following registers: - GCTL - GUCTL - GSTS - GSNPSID - GGPIO - GUID - GUSB2PHYCFGn registers - GUSB3PIPECTLn registers - DCFG - DCTL - DEVTEN - DSTS When you reset PHYs (using GUBS3PHYCFG or GUSB3PIPECTL registers), you must keep the core in reset state until PHY clocks are stable. This controls the bus, ram, and mac domain resets. Refer to the "Reset Generation" section in the Architecture Details chapter in the DesignWare Cores SuperSpeed USB 3.0 Controller Databook. Note: This bit is for debug purposes only. Use USBCMD.HCRESET in xHCI Mode and DCTL.SoftReset in device mode for soft reset. Programming this field with random data will reset the internal logic of the host controller. Due to this side effect Bit Bash register testing is not recommended.
10 SOFITPSYNC	If this bit is set to '0' operating in host mode, the core keeps the UTMI/ULPI PHY on the first port in a non-suspended state whenever there is a SuperSpeed port that is not in Rx.Detect, SS.Disable and U3. If this bit is set to '1' operating in host mode, the core keeps the UTMI/ULPI PHY on the first port in a non-suspended state whenever the other non-SuperSpeed ports are not in a suspended state. This feature is useful because it saves power by suspending UTMI/ULPI when SuperSpeed only is active, and it helps resolve when the PHY does not transmit a host resume unless it is placed in suspend state. This bit must

Table continues on the next page...

Field	Function
	be programmed as a part of initialization at power-on reset, and must not be dynamically changed afterwards. Note: - USB2PHYCFGn[6].PhySusp eventually decides to put the UTMI/ULPI PHY in to suspend state. In addition, when this bit is set to '1', the core generates ITP from the ref_clk based counter. Otherwise, ITP and SOF are generated from utmi/ulpi_clk[0] based counter. To program the reference clock period inside the core, refer to GUCTL[31:22].REFCLKPER. - This feature is valid in Host and DRD configurations and used only in Host mode operation. - If you never use this feature or the GFLADJ.GFLADJ_REFCLK_LPM_SEL, the minimum frequency for the ref_clk can be as low as 32KHz. You can connect the suspend_clk (as low as 32 KHz) to the ref_clk. - If you plan to enable hardware-based LPM or software-based LPM (PORTPMSC. HLE=1), then you cannot use this feature. Turn off this feature by setting this bit to '0' and use the GFLADJ.GFLADJ_REFCLK_LPM_SEL feature. - If you set this bit to '1', the GUSB2PHYCFG.U2_FREECLK_EXISTS bit and the DWC_USB3_FREECLK_USB2_EXIST parameter must be set to '0'. Program this bit to 0 if the core is intended to be operated in USB 3.0 mode.
9 U1U2TimerScale	Disable U1/U2 timer Scaledown If set to '1' along with GCTL[5:4] (ScaleDown) = 2'bX1, disables the scale down of U1/U2 inactive timer values. This is for simulation mode only.
8 DEBUGATTACH	Debug Attach When this bit is set, - SS Link proceeds directly to the Polling link state (after RUN/STOP in the DCTL register is asserted) without checking remote termination; - Link LFPS polling timeout is infinite; - Polling timeout during TS1 is infinite (in case link is waiting for TXEQ to finish).
7-6 RAMCLKSEL	RAM Clock Select 2'b00: bus clock - 2'b01: pipe clock (Only used in device mode) - 2'b10: In device mode , pipe/2 clock.In Host mode, controller switches ram_clk between pipe/2 clock, mac2_clk and bus_clk based on the status of the U2/U3 ports - 2'b11: In device mode, selects mac2_clk as ram_clk (when 8-bit UTMI or ULPI used. Not supported in 16-bit UTMI mode) In Host mode, controller switches ram_clk between pipe_clk, mac2_clk and bus_clk based on the status of the U2/U3 ports. In device mode, upon a USB reset and USB disconnect, the hardware clears these bits to 2'b00. For more information on how to select the RAM clock, see "Clock Generation and Clock Tree Synthesis (CTS) Requirements" in the Architectural Details chapter in the DesignWare Cores SuperSpeed USB 3.0 Controller Databook. Note: - In device mode, if you set RAMClkSel to 2'b11 (mac2_clk), the controller internally switches the ram_clk to bus_clk when the link state changes to Suspend (L2 or L3), and switches the ram_clk back to mac2_clk when the link state changes to resume or U2. - In host mode, if a value of 2/3 is chosen, then controller switches ram_clk between bus_clk, mac2_clk and pipe_clk, pipe_clk/2, based on the state of the U2/U3 ports. For example if only the U2 port is active and the U3 ports are suspended, then the ram_clk is swicthed to mac2_clk. When only the U3 ports are active and the U2 ports are suspended, then the core internally swicth the ram_clk to pipe3 clock and when all U2 and U3 ports are suspended, it switch the ram_clk to bus_clk. This allows de coupling the ram_clk from the bus_clk and depending on the bandwidth requiredmnet allows the bus_clk to be run at a lower frequency than the ram_clk requirements. bus_clk frequency still cannot go below 60Mhz in host mode, and this is not verified. A value of 2 can be chosen only if the pipe data width is 8 or 16 bits. In this case the when the ram_clk is switched to pipe_clk, it uses pipe_clk/2 instead of pipe_clk. If a value of 3 is chosen for RAMClkSel, then when ram_clk is switched to pipe_clk, then pipe_clk is used without any divider. - In device mode, when RAMClkSel != 2'b00, the bus_clk_early frequency can be a minimum of 1 MHz. This is tested in simulation and also in hardware with Linux, Microsoft Windows 8, and MCCI Windows7 host drivers. Only control and non periodic transfers are supported when bus_clk is 1 MHz. For periodic applications, the bus_clk_early minimum frequency is higher depending on your application and SoC bus. Even though 1 MHz has been tested with standard host drivers, Synopsys recommends 5 MHz minimum for ASIC designs to provide a margin or at least have a backup option to increase the bus_clk frequency to 5 MHz if needed. Programming this field with random data will cause side effect. Bit Bash register testing is not recommended.
5-4 SCALEDOWN	Scale-Down Mode When Scale-Down mode is enabled for simulation, the core uses scaled-down timing values, resulting in faster simulations. When Scale-Down mode is disabled, actual timing values are used. This is required for hardware operation. HS/FS/LS Modes - 2'b00: Disables all scale-downs. Actual timing values are used. - 2'b01: Enables scale-down of all timing values except Device mode suspend and resume. These

Table continues on the next page...

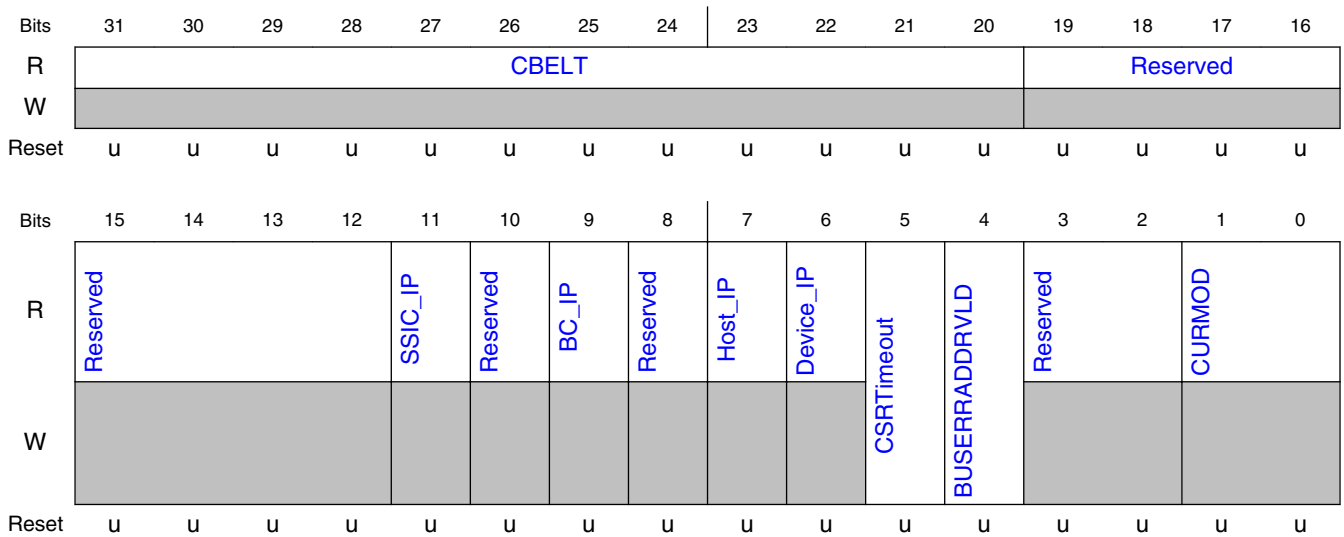
Field	Function
	include Speed enumeration, and Host mode suspend and resume - 2'b10: Enables scale-down of Device mode suspend and resume timing values only. - 2'b11: Enables bit 0 and bit 1 scale-down timing values. SS Mode - 2'b00: Disables all scale-downs. Actual timing values are used. - 2'b01: Enables scaled down SS timing and repeat values including: (1) Number of TxEq training sequences reduce to 8; (2) LFPS polling burst time reduce to 256 nS; (3) LFPS warm reset receive reduce to 30 uS. Refer to the rtl_vip_scaledown_mapping.xls file under <workspace>/sim/SoC_sim directory for the complete list. - 2'b10: No TxEq training sequences are sent. Overrides Bit 4. - 2'b11: Enables bit 0 and bit 1 scale-down timing values.
3 DISSCRAMBLE	Disable Scrambling Transmit request to Link Partner on next transition to Recovery or Polling.
2 U2EXIT_LFPS	If this bit is, - 0: the link treats 248ns LFPS as a valid U2 exit. - 1: the link waits for 8us of LFPS before it detects a valid U2 exit. This bit is added to improve interoperability with a third-party host/device controller. This host/device controller in U2 state while performing receiver detection generates an LFPS glitch of about 4ms duration. This causes the host/device to exit from U2 state because the LFPS filter value is 248ns. With the new functionality enabled, the host/device can stay in U2 while ignoring this glitch from the host/device controller. This bit is applicable for both host and device controller. This bit is added to improve interoperability with a third party host controller. This host controller in U2 state while performing receiver detection generates an LFPS glitch of about 4ms duration. This causes the device to exit from U2 state because the LFPS filter value is 248ns. With the new functionality enabled, the device can stay in U2 while ignoring this glitch from the host controller.
1 GblHibernationEn	Global Hibernation Enable This bit enables hibernation at the global level. If hibernation is not enabled through this bit, the PMU immediately accepts the D0->D3 and D3->D0 power state change requests, but does not save or restore any core state. In addition, the PMUs never drive the PHY interfaces and let the core continue to drive the PHY interfaces.
0 DSBLCLKGTNG	Disable Clock Gating This bit is set to 1 and the core is in Low Power mode, internal clock gating is disabled. You can set this bit to 1'b1 after Power On Reset.

11.2.5.1.55 Global Status Register (GSTS)

11.2.5.1.55.1 Offset

Register	Offset
GSTS	C118h

11.2.5.1.55.2 Diagram



11.2.5.1.55.3 Fields

Field	Function
31-20 CBELT	Current BELT Value In Host mode, this field indicates the minimum value of all received device BELT values and the BELT value that is set by the Set Latency Tolerance Value command.
19-12 —	Reserved
11 SSIC_IP	SSIC Interrupt Pending This field indicates that there is a pending interrupt related to SSIC in the SEVT register. Note: When the DWC_USB3_NUM_SSIC_PORTS parameter is set to zero, this bit is reserved.
10 —	Reserved
9 BC_IP	Battery Charger Interrupt Pending This field indicates that there is a pending interrupt pertaining to BC in BCEVT register.
8 —	Reserved
7 Host_IP	Host Interrupt Pending This field indicates that there is a pending interrupt pertaining to xHC in the Host event queue.
6 Device_IP	Device Interrupt Pending This field indicates that there is a pending interrupt pertaining to peripheral (device) operation in the Device event queue.
5 CSRTimeout	CSR Timeout When this bit is 1'b1, it indicates that the software performed a write or read to a core register that could not be completed within `DWC_USB3_CSR_ACCESS_TIMEOUT bus clock cycles (default: h1FFFF).

Table continues on the next page...

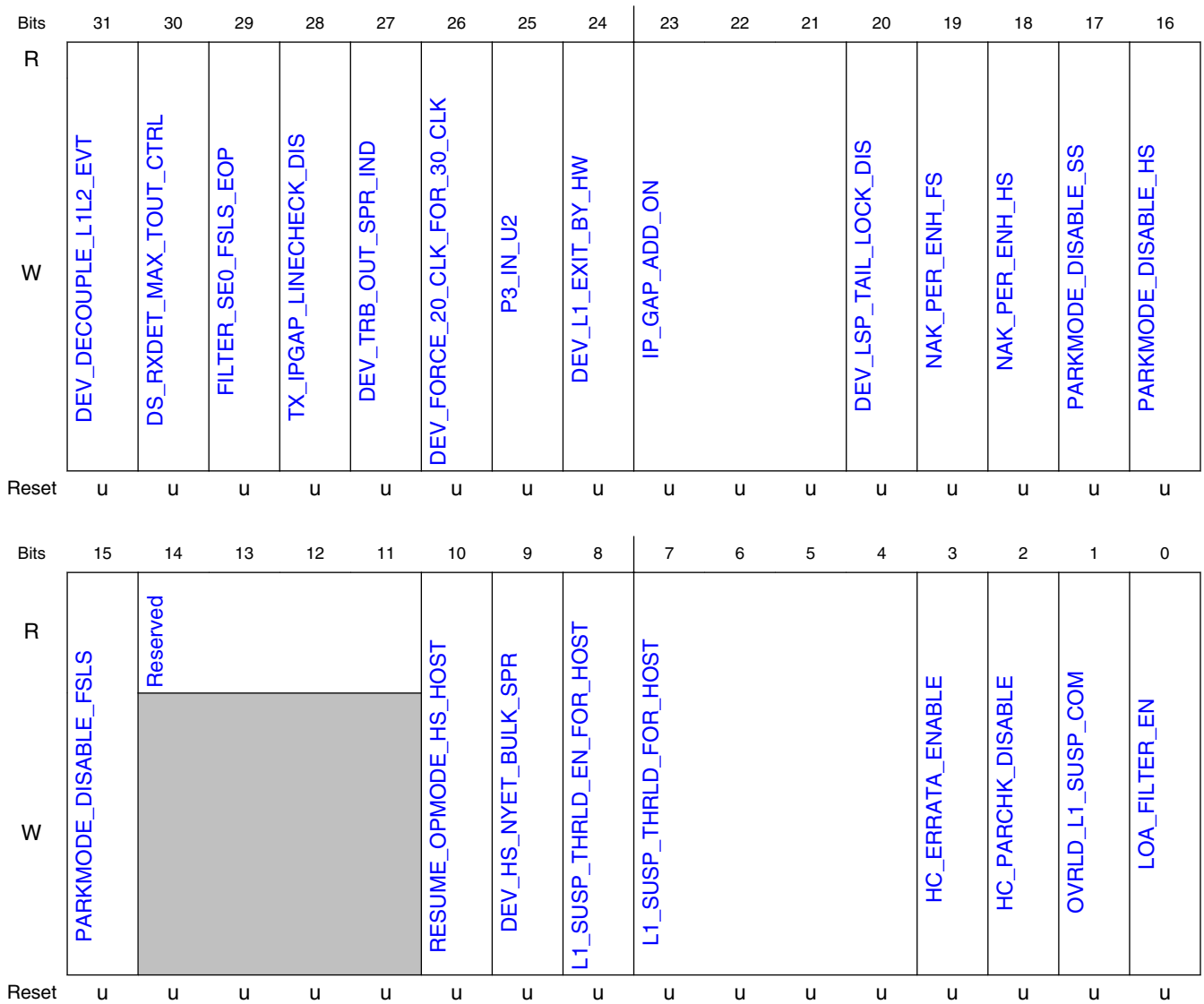
Field	Function
4 BUSERRADDRVLD	Bus Error Address Valid Indicates the GBUSERRADDR register is valid and reports the first bus address that encounters a bus error. Note: Only supported in AHB and AXI configurations.
3-2 —	Reserved
1-0 CURMOD	Current Mode of Operation Indicates the current mode of operation: - 2'b00: Device mode - 2'b01: Host mode

11.2.5.1.56 Global User Control Register 1 (GUCTL1)

11.2.5.1.56.1 Offset

Register	Offset
GUCTL1	C11Ch

11.2.5.1.56.2 Diagram



11.2.5.1.56.3 Fields

Field	Function
31 DEV_DECOUPLE_L1L2_EVT	DEV_DECOUPLE_L1L2_EVT - 0: Default behavior, no change in device events L1/L2U3 events are not decoupled (old behavior of v2.90a and before) - 1: Feature enabled, L1 and L2 events are separated when operating in 2.0 mode. Separate event enable bits for L1 suspend and wake events. This bit is applicable for device mode only. If this feature is enabled, L1 suspend and wake events have individual controls to enable/mask them. Enable this feature if you want to get L1 (LPM) events separtely and not combined with L2 events when operating in 2.0 speeds.
30 DS_RXDET_MAX_TOUT_CTRL	DS_RXDET_MAX_TOUT_CTRL This bit is used to control the tRxDetectTimeoutDFP timer for the SuperSpeed link. - 0: Default behavior; 12ms is used as tRxDetectTimeoutDFP. - 1: 120ms is used as the tRxDetectTimeoutDFP. This bit is used only in host mode. For more details, refer to ECN020 for USB 3.0 Specification.

Table continues on the next page...

Field	Function
29 FILTER_SE0_FSLS_EOP	FILTER_SE0_FSLS_EOP - 0: Default behavior, no change in Linestate check for SE0 detection in FS/LS - 1: Feature enabled, FS/LS SE0 is filtered for 2 clocks for detecting EOP This bit is applicable for FS/LS operation. If this feature is enabled, then SE0 on the linestate is validated for 2 consecutive utmi/ulpi clock edges for EOP detection. This feature is applicable only in FS in device mode and FS/LS mode of operation in host mode. Device mode: FS - If GUCTL1.FILTER_SE0_FSLS_EOP is set, then for device LPM handshake, the core will ignore single SE0 glitch on the linestate during transmit. Only 2 or more SE0 is considered as a valid EOP on FS. Host mode: FS/LS - If GUCTL1.FILTER_SE0_FSLS_EOP is set, then the core will ignore single SE0 glitch on the linestate during transmit. Only 2 or more SE0 is considered as a valid EOP on FS/LS port. Enable this feature if the LineState has SE0 glitches during transmission. This bit is quasi-static, that is, it must not be changed during device operation.
28 TX_IPGAP_LINECHECK_DIS	TX_IPGAP_LINECHECK_DIS - 0: Default behavior, no change in Linestate check - 1: Feature enabled, 2.0 MAC disables Linestate check during HS transmit This bit is applicable for HS operation of u2mac. If this feature is enabled, then the 2.0 mac operating in HS ignores the UTMI/ULPI Linestate during the transmit of a token (during token-to-token and token-to-data IPGAP). When enabled, the controller implements a fixed 40-bit TxEndDelay after the packet is given on UTMI and ignores the Linestate during this time. This feature is applicable only in HS mode of operation. Device mode: If GUCTL1.TX_IPGAP_LINECHECK_DIS is set, then for device LPM handshake, the core will ignore the linestate after TX and wait for fixed clocks (40 bit times equivalent) after transmitting ACK on utmi. Host mode: If GUCTL1.TX_IPGAP_LINECHECK_DIS is set, then the ipgap between (tkn to tkn/data) is added by 40 bit times of TXENDDELAY, and linestate is ignored during this 40 bit times delay. Enable this bit if the LineState will not reflect the expected line state (J) during transmission. This bit is quasi-static, that is, it must not be changed during device operation.
27 DEV_TRB_OUT_SPR_IND	DEV_TRB_OUT_SPR_IND - 0: Default behavior, no change in TRB status dword - 1: Feature enabled, OUT TRB status indicates Short Packet This bit is applicable for device mode only (and ignored in host mode). If the device application (software/hardware) wants to know if a short packet was received for an OUT in the TRB status itself, then this feature can be enabled, so that a bit is set in the TRB writeback in the buf_size dword. Bit[26] - SPR of the {trbstatus, RSVD, SPR, PCM1, bufsize} dword will be set during an OUT transfer TRB write back if this is the last TRB used for that transfer descriptor. This bit is quasi-static, that is, it must not be changed during device operation.
26 DEV_FORCE_20_CLK_FOR_30_CLK	DEV_FORCE_20_CLK_FOR_30_CLK - 0: Default behavior, Uses 3.0 clock when operating in 2.0 mode - 1: Feature enabled This bit is applicable (and to be set) for device mode (DCFG.Speed != SS) only. In the 3.0 device core, if the core is programmed to operate in 2.0 only (that is, Device Speed is programmed to 2.0 speeds in DCFG[Speed]), then setting this bit makes the internal 2.0 (utmi/ulpi) clock to be routed as the 3.0 (pipe) clock. Enabling this feature allows the pipe3 clock to be not-running when forcibly operating in 2.0 device mode. Note: - When using this feature, all pipe3 inputs must be in inactive mode. In particular, the pipe3 clocks must not be running and the pipe3_phystatus_async must be tied to 0. This bit should not be set if the core is programmed to operate in SuperSpeed mode (even when it falls back to 2.0). - This bit is quasi-static, that is, it must not be changed during operation. - If the parameter "DWC_USB3_REMOVE_PIPE_CLK_MUX_FOR_20_MODE" is enabled, then muxing 2.0 clock/signals to pipe_clk/signals have to be done outside the controller for this feature to work.
25 P3_IN_U2	P3_IN_U2 - 0: Default behavior, When SuperSpeed link is in U2, PowerState P2 is attempted on the PIPE Interface. - 1: When SuperSpeed link is in U2, PowerState P3 is attempted if GUSB3PIPECTL[17] is set. Setting this bit enables P3 Power State when the SuperSpeed link is in U2. Another Power Saving option. Check with your PHY vendor before enabling this option.When setting this bit to 1 to enable P3 in P2, GUSB3PIPECTL[27] should be set to 0 to make sure that the U2 exit is attempted in P0. This bit should be set only when GCTL.SOFITPSYNC=1 or GFLADJ.GFLADJ_REFCLK_LPM_SEL=1.
24 DEV_L1_EXIT_BY_HW	DEV_L1_EXIT_BY_HW - 0: Default behavior, disables device L1 hardware exit logic - 1: feature enabled This bit is applicable for device mode (2.0) only. This field enables device controller sending remote wakeup for L1 if the device becomes ready for sending/accepting data when in L1 state. If the host expects the device to send remote wakeup signaling to resume after going into L1 in flow controlled state, then this bit can be set to send the remote wake signal automatically when the device controller becomes ready. This hardware remote wake feature is applicable only to bulk and interrupt transfers, and not for Isoch/Control - When control transfers are in progress, the LPM will be rejected (NYET response). Only after control transfers are completed (either with ACK/STALL), LPM will be accepted - For Isoch transfers, the host needs to do the wake-up and start the transfer. Device controller will not do remote-

Table continues on the next page...

Field	Function
	wakeup when Isoch endpoints get ready. The device SW needs to keep the GUSB2PHYCFG[EnblSlpM] reset in order to keep the PHY clock to be running for keeping track of SOF intervals. - When L1 hibernation is enabled, the controller will not do automatic exit for hibernation requests thru L1. This bit is quasi-static, that is, it must not be changed during device operation.
23-21 IP_GAP_ADD_ON	This register field is used to add on to the default inter packet gap setting in the USB 2.0 MAC. This should be programmed to a non zero value only in case where you need to increase the default inter packet delay calculations in the USB 2.0 MAC module DWC_usb3_u2mac.v
20 DEV_LSP_TAIL_LOCK_DIS	DEV_LSP_TAIL_LOCK_DIS - 0: Default behavior, enables device lsp lock logic for tail TRB update - 1: Fix disabled This is a bug fix for STAR 9000716195 that affects the CSP mode for OUT endpoints in device mode. The issue is that tail TRB index is not synchronized with the cache Scratchpad bytecount update. If the fast-forward request comes in-between the bytecount update on a newly fetched TRB and the tail-index write update in TPF, the RDP works on an incorrect tail index and misses the byte count decrement for the newly fetched TRB in the fast-forwarding process. This fix needs to be present all the times.
19 NAK_PER_ENH_FS	NAK_PER_ENH_FS - 1: Enables performance enhancement for FS async endpoints in the presence of NAKs - 0: Enhancement not applied If a periodic endpoint is present , and if a bulk endpoint which is also active is being NAKed by the device, then this could result in a decrease in performance of other Full Speed bulk endpoint which is ACked by the device. Setting this bit to 1, will enable the host controller to schedule more transactions to the async endpoints (bulk/ control) and hence will improve the performance of the bulk endpoint. This control bit should be enabled only if the existing performance with the default setting is not sufficient for your FullSpeed application. Setting this bit will only control, and is only required for Full Speed transfers.
18 NAK_PER_ENH_HS	NAK_PER_ENH_HS - 1: Enables performance enhancement for HS async endpoints in the presence of NAKs - 0: Enhancement not applied If a periodic endpoint is present, and if a bulk endpoint which is also active is being NAKed by the device, then this could result in decrease in performance of other High Speed bulk endpoint which is ACked by the device. Setting this bit to 1, will enable the host controller to schedule more transactions to the async endpoints (bulk/ control) and hence will improve the performance of the bulk endpoint. This control bit should be enabled only if the existing performance with the default setting is not sufficient for your HighSpeed application. Setting this bit will only control, and is only required for High Speed transfers.
17 PARKMODE_DISABLE_SS	PARKMODE_DISABLE_SS This bit is used only in host mode, and is for debug purpose only. When this bit is set to '1' all SS bus instances in park mode are disabled.
16 PARKMODE_DISABLE_HS	PARKMODE_DISABLE_HS This bit is used only in host mode. When this bit is set to '1' all HS bus instances park mode are disabled. To improve performance in park mode, the xHCI scheduler queues in three requests of 4 packets each for High Speed asynchronous endpoints in a micro-frame. But if a device is slow and if it NAKs more than 3 times, then it is rescheduled only in the next micro-frame. This could decrease the performance of a slow device even further. In a few high speed devices (such as Sandisk Cruzer Blade 4GB VID:1921, PID:21863 and Flex Drive VID:3744, PID:8552) when an IN request is sent within 900ns of the ACK of the previous packet, these devices send a NAK. When connected to these devices, if required, the software can disable the park mode if you see performance drop in your system. When park mode is disabled, pipelining of multiple packet is disabled and instead one packet at a time is requested by the scheduler. This allows up to 12 NAKs in a micro-frame and improves performance of these slow devices.
15 PARKMODE_DISABLE_FSLS	PARKMODE_DISABLE_FSLS This bit is used only in host mode, and is for debug purpose only. When this bit is set to '1' all FS/LS bus instances in park mode disabled.
14-11 —	Reserved
10	RESUME_OPMODE_HS_HOST This bit is used only in host mode, and is for USB 2.0 opmode behavior in HS Resume. - When this bit is set to '1', the UTMI/ULPI opmode will be changed to "normal" along with HS terminations after EOR. This option is to support certain legacy UTMI/ULPI PHYs. - When this bit is

Table continues on the next page...

Field	Function
RESUME_OPMODE_HS_HOST	set to '0', the UTMI/ULPI opmode will be changed to "normal" 2us after HS terminations change after EOR. This is the default behavior.
9 DEV_HS_NYET_BULK_SPR	DEV_HS_NYET_BULK_SPR - 0: Default behavior, no change in device response - 1: Feature enabled, HS bulk OUT short packet gets NYET response This bit is applicable for device mode only (and ignored in host mode) to be used in 2.0 operation. If this bit is set, the device core sends NYET response instead of ACK response for a successfully received bulk OUT short packet. If NYET is sent after receiving short packet, then the host would PING before sending the next OUT; this improves the performance as well as clears up the buffer/cache on the host side. Internal to the device core, short packet (SPR=1) processing takes some time, and during this time, the USB is flow controlled. With NYET response instead of ACK on short packet, the host does not send another OUT-DATA without pinging in HS mode. This bit is quasi-static, that is, it must not be changed during device operation.
8 L1_SUSP_THRLD_EN_FOR_HOST	L1_SUSP_THRLD_EN_FOR_HOST This bit is used only in host mode. The host controller asserts the utmi_l1_suspend_n and utmi_sleep_n output signals (see "LPM Interface Signals" table in the Databook) as follows: The controller asserts the utmi_l1_suspend_n signal to put the PHY into deep low-power mode in L1 when both of the following are true: - The HIRD/BESL value used is greater than or equal to the value in L1_SUSP_THRLD_FOR_HOST field. - The L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1'b1. The controller asserts utmi_sleep_n on L1 when one of the following is true: - The HIRD/BESL value used is less than the value in L1_SUSP_THRLD_FOR_HOST field. - The L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1'b0.
7-4 L1_SUSP_THRLD_FOR_HOST	L1_SUSP_THRLD_FOR_HOST This field is effective only when the L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1. For more details, refer to the description of the L1_SUSP_THRLD_EN_FOR_HOST bit.
3 HC_ERRATA_ENABLE	Host ELD Enable When this bit is set to 1, it enables the Exit Latency Delta (ELD) support defined in the xHCI 1.0 Errata. This bit is used only in the host mode. This bit has to be set to 1 in Host mode.
2 HC_PARCHK_DISABLE	Host Parameter Check Disable When this bit is set to '0' (by default), the xHC checks that the input slot/EP context fields comply to the xHCI Specification. Upon detection of a parameter error during command execution, the xHC generates an event TRB with completion code indicating 'PARAMETER ERROR'. When the bit is set to '1', the xHC does not perform parameter checks and does not generate 'PARAMETER ERROR' completion code.
1 OVRLD_L1_SUSP_COM	OVRLD_L1_SUSP_COM If this bit is set, the utmi_l1_suspend_com_n is overloaded with the utmi_sleep_n signal. This bit is usually set if the PHY stops the port clock during L1 sleep condition. Note: The recommended connection for the SUSPENDM/SLEEPM signals to the PHY with respect to this bit is as follows. For non-zero ports: Connect: - utmi_sleep_n[n] to SLEEPM[n] - (utmi_suspend_n[n] & utmi_l1_suspend_n[n]) to SUSPENDM[n] - USB2 PHYCLK[n] to utmi_clk[n] GUCTL1.OVRLD_L1_SUSP_COM impacts only Port0. For Port0: For Synopsys PHY, GUSB2PHYCFGn.U2_FREECLK_EXISTS=1; With this connection, the PHY keeps PLL active so that FREECLK is always available irrespective of suspend/sleep. - Connect USB2 PHY COMMONONN to 0. - Connect utmi_sleep_n[0] to SLEEPM[0]. - Connect (utmi_suspend_n[0] & utmi_l1_suspend_n[0]) to SUSPENDM[0]. - Connect USB2 PHY FREECLK to utmi_clk[0]. - Leave utmi_suspend_com_n, utmi_l1_suspend_com_n unconnected. - GUCTL1.OVRLD_L1_SUSP_COM can be set to any value. For Third Party PHY, GUSB2PHYCFGn.U2_FREECLK_EXISTS=0; With this connection the PHY can shut off all the clocks when the required conditions are met (like, GUSB2PHYCFGn[8,6], GUCTL1[1], GFLADJ[23], GCTL[10], Suspend condition, HW LPM enable etc). - Connect -utmi_suspend_com_n to SUSPENDM[0] (or equivalent). - Connect -utmi_l1_suspend_com_n to SLEEPM[0] (or equivalent). - Connect PHYCLK0 (first port clock) to utmi_clk[0]. - Leave utmi_suspend_n[0], utmi_l1_suspend_ n[0], utmi_sleep_n[0] unconnected. - Set GUCTL1.OVRLD_L1_SUSP_COM to 1'b1.
0 LOA_FILTER_EN	LOA_FILTER_EN If this bit is set, the USB 2.0 port babble is checked at least three consecutive times before the port is disabled. This prevents false triggering of the babble condition when using low quality cables. Note: This bit is valid only in host mode.

11.2.5.1.57 Global User ID Register (GUID)

11.2.5.1.57.1 Offset

Register	Offset
GUID	C128h

11.2.5.1.57.2 Function

This is a read/write register containing the User ID.

11.2.5.1.57.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	USERID															
Reset	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	USERID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.57.4 Fields

Field	Function
31-0 USERID	User ID Application-programmable ID field.

11.2.5.1.58 Global User Control Register (GUCTL)

11.2.5.1.58.1 Offset

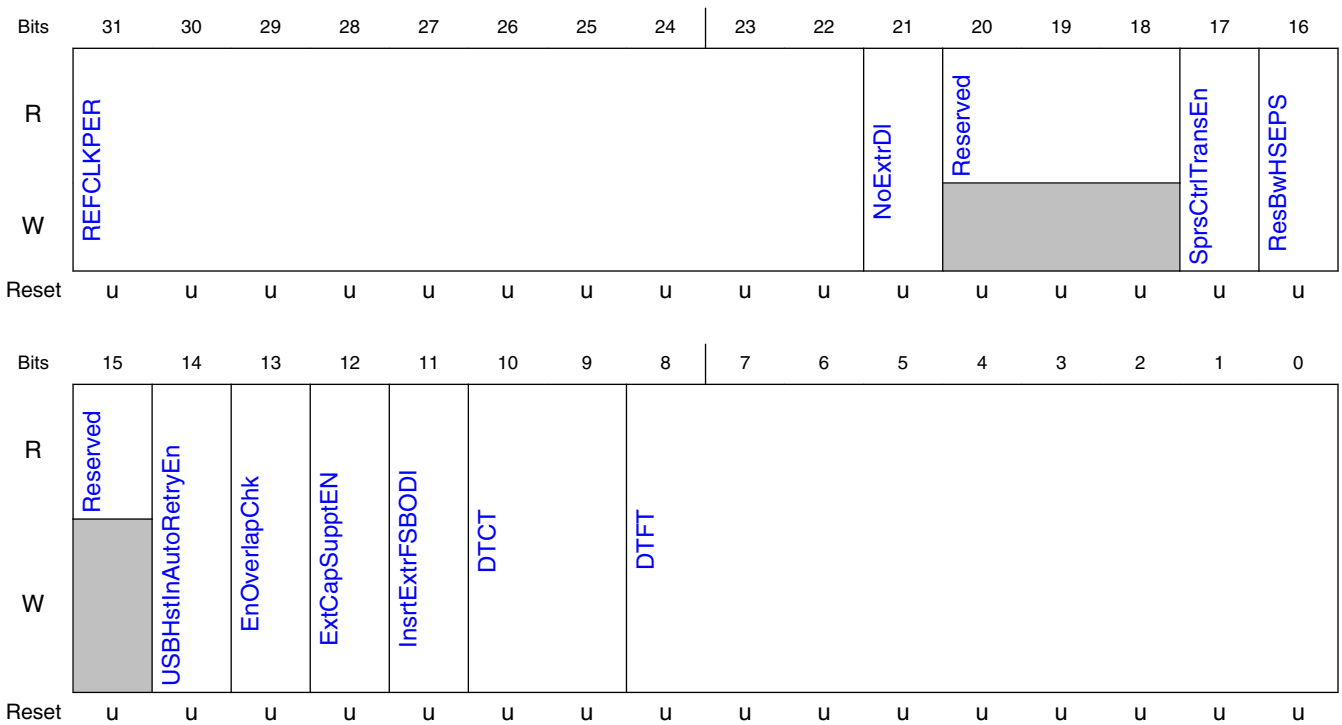
Register	Offset
GUCTL	C12Ch

11.2.5.1.58.2 Function

This register provides a few options for the software to control the core behavior in the Host mode.

Reset Mask: 0x1C8000

11.2.5.1.58.3 Diagram



11.2.5.1.58.4 Fields

Field	Function
31-22 REFCLKPER	This field indicates in terms of nano seconds the period of ref_clk. The default value of this register is set to 'h8 (8ns/125 MHz). This field needs to be updated during power-on initialization, if GCTL.SOFITPSYNC or GFLADJ.GFLADJ_REFCLK_LPM_SEL is set to '1'. The programmable maximum value is 62ns, and the minimum value is 8ns. You must use a reference clock with a period that is an integer multiple, so that ITP can meet the jitter margin of 32ns. The allowable ref_clk frequencies whose period is not integer multiples are 16/17/19.2/24/39.7MHz. This field must not be set to '0' at any time. If you never plan to use this feature, then set this field to 'h8, the default value.
21 NoExtrDl	No Extra Delay Between SOF and the First Packet Some HS devices misbehave when the host sends a packet immediately after a SOF. However, adding an extra delay between a SOF and the first packet can reduce the USB data rate and performance. This bit is used to control whether the host must wait for 2 microseconds before it sends the first packet after a SOF, or not. User can set this bit to one to improve the performance if those problematic devices are not a concern in the user's host environment. - 1'b0: Host waits for 2 microseconds after a SOF before it sends the first USB packet. - 1'b1: Host doesn't wait after a SOF before it sends the first USB packet.

Table continues on the next page...

Field	Function
20-18 —	Reserved
17 SprsCtrlTransEn	Sparse Control Transaction Enable Some devices are slow in responding to Control transfers. Scheduling multiple transactions in one microframe/frame can cause these devices to misbehave. If this bit is set to 1'b1, the host controller schedules transactions for a Control transfer in different microframes/frames.
16 ResBwHSEPS	Reserving 85% Bandwidth for HS Periodic EPs By default, HC reserves 80% of the bandwidth for periodic EPs. If this bit is set, the bandwidth is relaxed to 85% to accommodate two high speed, high bandwidth ISOC EPs. USB 2.0 required 80% bandwidth allocated for ISOC traffic. If two High-bandwidth ISOC devices (HD Webcams) are connected, and if each requires 1024-bytes X 3 packets per Micro-Frame, then the bandwidth required is around 82%. If this bit is set, then it is possible to connect two Webcams of 1024bytes X 3 paylod per Micro-Frame each. Otherwise, you may have to reduce the resolution of the Webcams. This bit is valid in Host and DRD configuration and is used in host mode operation only. Ignore this bit in device mode.
15 —	Reserved
14 USBHstInAutoRetryEn	Host IN Auto Retry When set, this field enables the Auto Retry feature. For IN transfers (non-isochronous) that encounter data packets with CRC errors or internal overrun scenarios, the auto retry feature causes the Host core to reply to the device with a non-terminating retry ACK (that is, an ACK transaction packet with Retry = 1 and NumP != 0). If the Auto Retry feature is disabled (default), the core will respond with a terminating retry ACK (that is, an ACK transaction packet with Retry = 1 and NumP = 0). - 1'b0: Auto Retry Disabled - 1'b1: Auto Retry Enabled Note: When enabling Auto Retry feature, if the system latency is large enough to cause the internal PSQ full (PSQ can be full as the result of messages not being processed because of pending fetches before flushing the TxQ due to NRDY/ERDY conditions), then the host controller can generate a transaction error.
13 EnOverlapChk	Enable Check for LFPS Overlap During Remote Ux Exit: If this bit is set to, - 1'b1: The SuperSpeed link when exiting U1/U2/U3 waits for either the remote link LFPS or TS1/TS2 training symbols before it confirms that the LFPS handshake is complete. This is done to handle the case where the LFPS glitch causes the link to start exiting from the low power state. Looking for the LFPS overlap makes sure that the link partner also sees the LFPS. - 1'b0: When the link exists U1/U2/U3 because of a remote exit, it does not look for an LFPS overlap.
12 ExtCapSupptEN	External Extended Capability Support Enable When set, this field enables extended capabilities to be implemented outside the core. When the ExtCapSupEN is set and the Debug Capability is enabled, the Next Capability pointer in "Debug Capability" returns 16. A read to the first DWORD of the last internal extended capability (the "xHCI Supported Protocol Capability for USB 3.0" when the Debug Capability is not enabled) returns a value of 4 in the Next Capability Pointer field. This indicates to software that there is another capability four DWORDs after this capability (for example, at address N+16 where N is the address of this DWORD). If enabled, an external address decoder that snoops the xHC slave interface must be implemented. If it sees an access to N+16 or greater, the slave access is re-routed to a piece of hardware which returns the external capability pointer register of the new capability and also handles reads/writes to this new capability and the side effects. If disabled, a read to the first DWORD of the last internal extended capability returns 0 in the 'Next Capability Pointer field. This indicates there are no more capabilities.
11 InsrtExtrFSBODI	Insert Extra Delay Between FS Bulk OUT Transactions Some FS devices are slow to receive Bulk OUT data and can get stuck when there are consecutive Bulk OUT transactions with short inter-transaction delays. This bit is used to control whether the host inserts extra delay between consecutive Bulk OUT transactions to a FS Endpoint. - 1'b0: Host doesn't insert extra delay between consecutive Bulk OUT transactions to a FS Endpoint. - 1'b1: Host inserts about 12us extra delay between consecutive Bulk OUT transactions to a FS Endpoint to work around the device

Table continues on the next page...

Field	Function
	issue. Note: Setting this bit to one will reduce the Bulk OUT transfer performance for most of the FS devices.
10-9 DTCT	Device Timeout Coarse Tuning This field is a Host mode parameter which determines how long the host waits for a response from device before considering a timeout. The core first checks the DTCT value. If it is 0, then the timeout value is defined by the DTFT. If it is non-zero, then it uses the following timeout values: - 2'b00: 0 usec -> use DTFT value instead - 2'b01: 500 usec - 2'b10: 1.5 msec - 2'b11: 6.5 msec Note: When the system latency is larger than the programmed DTCT/DTFT value, if the host controller is not able to accept certain transactions on the bus (because of system bus delays), the controller may not release header credits which in turn can cause the host to report a transaction error. Therefore, program this value to be larger than your system delay.
8-0 DTFT	Device Timeout Fine Tuning This field is a Host mode parameter which determines how long the host waits for a response from device before considering a timeout. For the DTFT field to take effect, DTCT must be set to 2'b00. The DTFT value is the number of 125 MHz clocks * 256 to count before considering a device timeout. The minimum value of DTFT is 2. For example, if the mac3_clk is 125 MHz clk (8 ns period), this is calculated as follows: (DTFT value) * 256 * (8 ns) Quick Reference: - if DTFT = 0x2, 2*256*8 = 4usec timeout - if DTFT = 0x5, 5*256*8 = 10usec timeout - if DTFT = 0xA, 10*256*8 = 20usec timeout - if DTFT = 0x10, 16*256*8 = 32usec timeout - if DTFT = 0x19, 25*256*8 = 51usec timeout - if DTFT = 0x31, 49*256*8 = 100usec timeout - if DTFT = 0x62, 98*256*8 = 200usec timeout Note: - When SSIC is enabled, in HS_G1_G2_G3 mode when ssic_soc_pa_clk_freq = 2'b01, then DTFT value must be calculated for 156.25 MHz clock, for example, for equivalent delay of 32usec DTFT value must be 9'h14. - When the system latency is larger than the programmed DTCT/DTFT value, if the host controller is not able to accept certain transactions on the bus (because of system bus delays), the controller may not release header credits which in turn can cause the host to report a transaction error. Therefore, program this value to be larger than your system delay.

11.2.5.1.59 Gobal SoC Bus Error Address Register - Low (GBUSERRADDRLO)

11.2.5.1.59.1 Offset

Register	Offset
GBUSERRADDRLO	C130h

11.2.5.1.59.2 Function

When the AHB or AXI Master Bus returns an "Error" response, the "SoC Bus Error" is generated. In the Host mode, the host_system_err port indicates this condition. In addition, it is also indicated in the USBSTS.HSE field. In the Device mode, the GSTS.BusErrAddrVld field is the only indication of the SoC Bus Error

11.2.5.1.59.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUSERRADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUSERRADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.59.4 Fields

Field	Function
31-0 BUSERRADDR	Bus Address - Low This register contains the lower 32 bits of the first bus address that encountered a SoC bus error. It is valid when the GSTS.BusErrAddrVld field is 1. It can only be cleared by resetting the core. Note: Only supported in AHB and AXI configurations.

11.2.5.1.60 Gobal SoC Bus Error Address Register - High (GBUSERRADDRHI)

11.2.5.1.60.1 Offset

Register	Offset
GBUSERRADDRHI	C134h

11.2.5.1.60.2 Function

When the AHB or AXI Master Bus returns an "Error" response, the "SoC Bus Error" is generated. In the Host mode, the host_system_err port indicates this condition. In addition, it is also indicated in the USBSTS.HSE field. In the Device mode, the GSTS.BusErrAddrVld field is the only indication of the SoC Bus Error

11.2.5.1.60.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUSERRADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUSERRADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.60.4 Fields

Field	Function
31-0 BUSERRADDR	Bus Address - High This register contains the higher 32 bits of the first bus address that encountered a SoC bus error. It is valid when the GSTS.BusErrAddrVld field is 1. It can only be cleared by resetting the core. Note: Only supported in AHB and AXI configurations.

11.2.5.1.61 Global SS Port to Bus Instance Mapping Register - Low (GPRTBIMAPLO)

11.2.5.1.61.1 Offset

Register	Offset
GPRTBIMAPLO	C138h

11.2.5.1.61.2 Function

This register specifies the SuperSpeed USB instance number to which each USB 3.0 port is connected. By default, USB 3.0 ports are evenly distributed among all SS USB instances. The software can program this register to specify how USB 3.0 ports are connected to SS USB instances. Register fields are read-write with respect to number of port instantiated. writeAsRead constraint is added to limit side effects for unused fields.

11.2.5.1.61.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BINUM8				BINUM7				BINUM6				BINUM5			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BINUM4				BINUM3				BINUM2				BINUM1			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.61.4 Fields

Field	Function
31-28 BINUM8	SS USB Instance Number for Port 8. Application-programmable ID field.
27-24 BINUM7	SS USB Instance Number for Port 7. Application-programmable ID field.
23-20 BINUM6	SS USB Instance Number for Port 6. Application-programmable ID field.
19-16 BINUM5	SS USB Instance Number for Port 5. Application-programmable ID field.
15-12 BINUM4	SS USB Instance Number for Port 4. Application-programmable ID field.
11-8 BINUM3	SS USB Instance Number for Port 3. Application-programmable ID field.
7-4 BINUM2	SS USB Instance Number for Port 2. Application-programmable ID field.
3-0 BINUM1	SS USB Instance Number for Port 1. Application-programmable ID field.

11.2.5.1.62 Global SS Port to Bus Instance Mapping Register - High (GPRTBIMAPHI)

11.2.5.1.62.1 Offset

Register	Offset
GPRTBIMAPHI	C13Ch

11.2.5.1.62.2 Function

This register specifies the SuperSpeed USB instance number to which each USB 3.0 port is connected. By default, USB 3.0 ports are evenly distributed among all SS USB instances. The software can program this register to specify how USB 3.0 ports are connected to SS USB instances. Register fields are read-write with respect to number of port instantiated. writeAsRead constraint is added to limit side effects for unused fields.

Reset Mask: 0xF0000000

11.2.5.1.62.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				BINUM15				BINUM14				BINUM13			
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BINUM12				BINUM11				BINUM10				BINUM9			
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

11.2.5.1.62.4 Fields

Field	Function
31-28 —	Reserved
27-24 BINUM15	SS USB Instance Number for Port 15. Application-programmable ID field.
23-20 BINUM14	SS USB Instance Number for Port 14. Application-programmable ID field.
19-16 BINUM13	SS USB Instance Number for Port 13. Application-programmable ID field.
15-12 BINUM12	SS USB Instance Number for Port 12. Application-programmable ID field.
11-8 BINUM11	SS USB Instance Number for Port 11. Application-programmable ID field.
7-4 BINUM10	SS USB Instance Number for Port 10. Application-programmable ID field.
3-0 BINUM9	SS USB Instance Number for Port 9. Application-programmable ID field.

11.2.5.1.63 Global Hardware Parameters Register 0 (GHWPARAMS0)

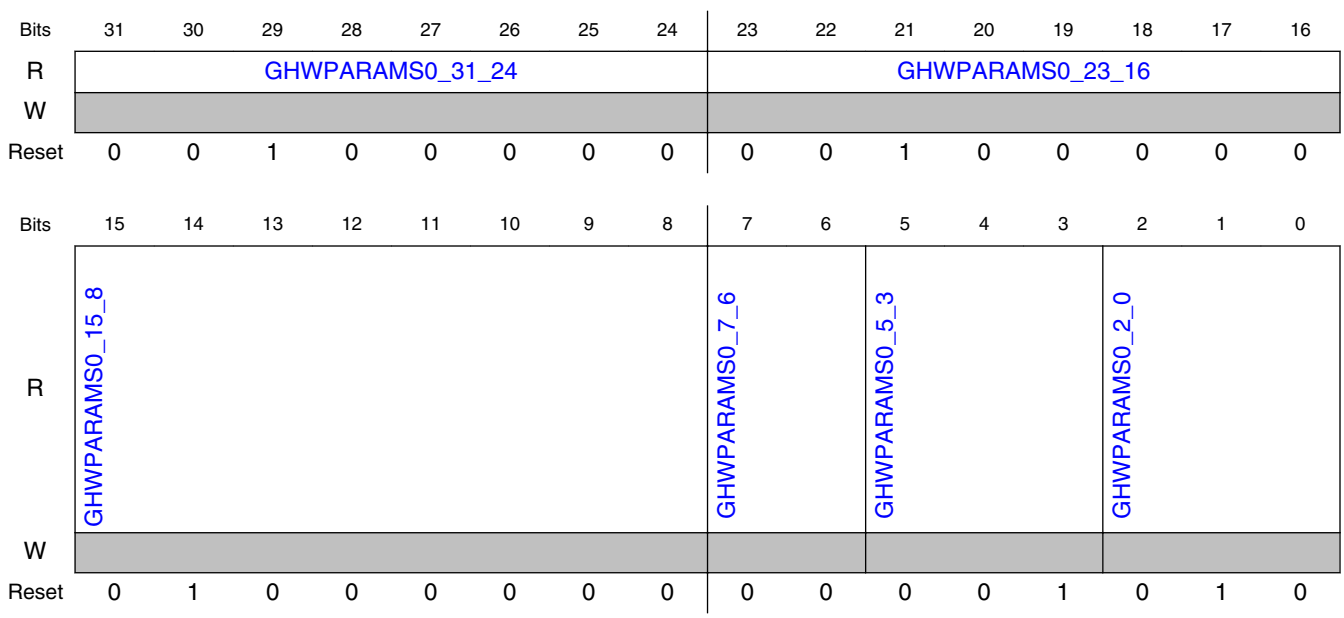
11.2.5.1.63.1 Offset

Register	Offset
GHWPARAMS0	C140h

11.2.5.1.63.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.2.5.1.63.3 Diagram



11.2.5.1.63.4 Fields

Field	Function
31-24 GHWPARAMS0_31_24	`DWC_USB3_AWIDTH
23-16	`DWC_USB3_SDWIDTH

Table continues on the next page...

Field	Function
GHWPARAMS0_23_16	
15-8 GHWPARAMS0_15_8	`DWC_USB3_MDWIDTH
7-6 GHWPARAMS0_7_6	`DWC_USB3_SBUS_TYPE
5-3 GHWPARAMS0_5_3	`DWC_USB3_MBUS_TYPE
2-0 GHWPARAMS0_2_0	`DWC_USB3_MODE

11.2.5.1.64 Global Hardware Parameters Register 1 (GHWPARAMS1)

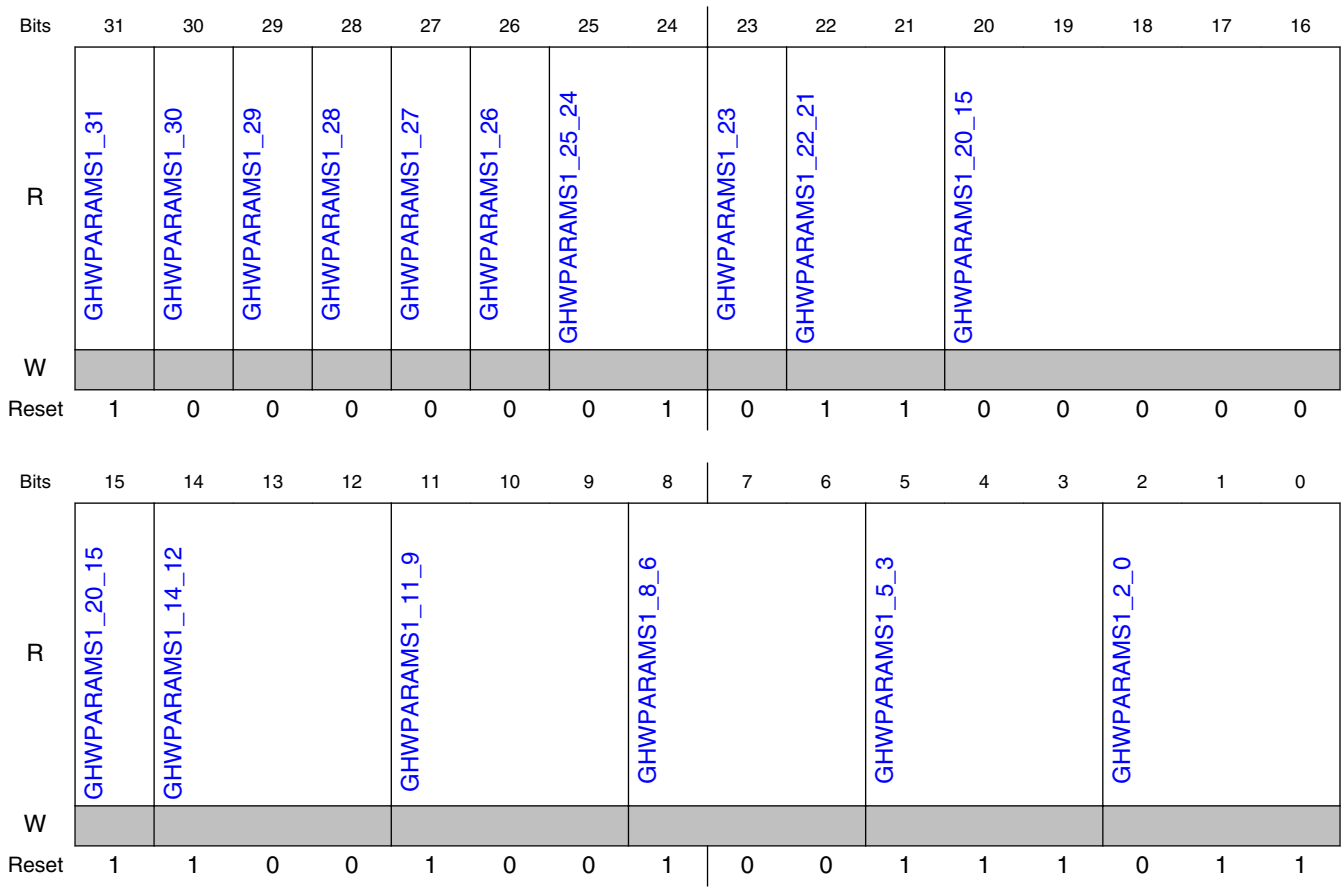
11.2.5.1.64.1 Offset

Register	Offset
GHWPARAMS1	C144h

11.2.5.1.64.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.2.5.1.64.3 Diagram



11.2.5.1.64.4 Fields

Field	Function
31 GHWPARAMS1_31	`DWC_USB3_EN_DBC
30 GHWPARAMS1_30	`DWC_USB3_RM_OPT_FEATURES
29 GHWPARAMS1_29	Reserved1
28 GHWPARAMS1_28	`DWC_USB3_RAM_BUS_CLKS_SYNC
27 GHWPARAMS1_27	`DWC_USB3_MAC_RAM_CLKS_SYNC

Table continues on the next page...

Field	Function
26 GHWPARAMS1_26	`DWC_USB3_MAC_PHY_CLKS_SYNC
25-24 GHWPARAMS1_25_24	`DWC_USB3_EN_PWROPT
23 GHWPARAMS1_23	`DWC_USB3_SPRAM_TYP
22-21 GHWPARAMS1_22_21	`DWC_USB3_NUM_RAMS
20-15 GHWPARAMS1_20_15	`DWC_USB3_DEVICE_NUM_INT For details on `DWC_USB3_DEVICE_NUM_INT, refer to <workspace>/src/DWC_usb3_params.v file.
14-12 GHWPARAMS1_14_12	`DWC_USB3_ASPACEWIDTH
11-9 GHWPARAMS1_11_9	`DWC_USB3_REQINFOWIDTH
8-6 GHWPARAMS1_8_6	`DWC_USB3_DATAINFOWIDTH
5-3 GHWPARAMS1_5_3	`DWC_USB3_BURSTWIDTH-1
2-0 GHWPARAMS1_2_0	`DWC_USB3_IDWIDTH-1

11.2.5.1.65 Global Hardware Parameters Register 2 (GHWPARAMS2)

11.2.5.1.65.1 Offset

Register	Offset
GHWPARAMS2	C148h

11.2.5.1.65.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.2.5.1.65.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GHWPARAMS2_31_0															
W																
Reset	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GHWPARAMS2_31_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.65.4 Fields

Field	Function
31-0 GHWPARAMS2_31_0	`DWC_USB3_USERID

11.2.5.1.66 Global Hardware Parameters Register 3 (GHWPARAMS3)

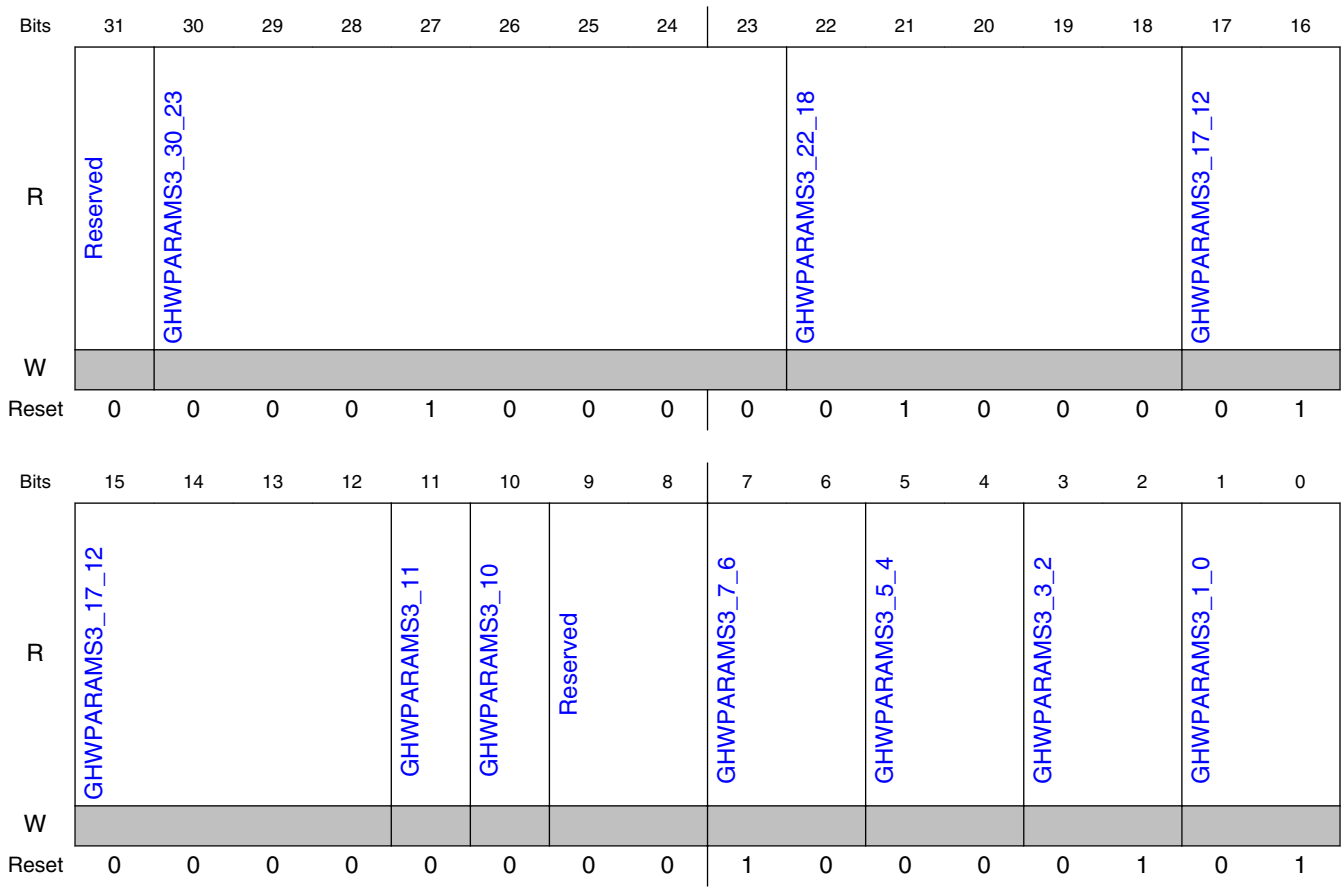
11.2.5.1.66.1 Offset

Register	Offset
GHWPARAMS3	C14Ch

11.2.5.1.66.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.2.5.1.66.3 Diagram



11.2.5.1.66.4 Fields

Field	Function
31 —	Reserved
30-23 GHWPARAMS3_30_23	`DWC_USB3_CACHE_TOTAL_XFER_RESOURCES
22-18 GHWPARAMS3_22_18	`DWC_USB3_NUM_IN_EPS
17-12 GHWPARAMS3_17_12	`DWC_USB3_NUM_EPS
11 GHWPARAMS3_11	`DWC_USB3_ULPI_CARKIT

Table continues on the next page...

Field	Function
10 GHWPARAMS3_10	`DWC_USB3_VENDOR_CTL_INTERFACE
9-8 —	Reserved
7-6 GHWPARAMS3_7_6	`DWC_USB3_HSPHY_DWIDTH
5-4 GHWPARAMS3_5_4	`DWC_USB3_FSPHY_INTERFACE
3-2 GHWPARAMS3_3_2	`DWC_USB3_HSPHY_INTERFACE
1-0 GHWPARAMS3_1_0	`DWC_USB3_SSPHY_INTERFACE

11.2.5.1.67 Global Hardware Parameters Register 4 (GHWPARAMS4)

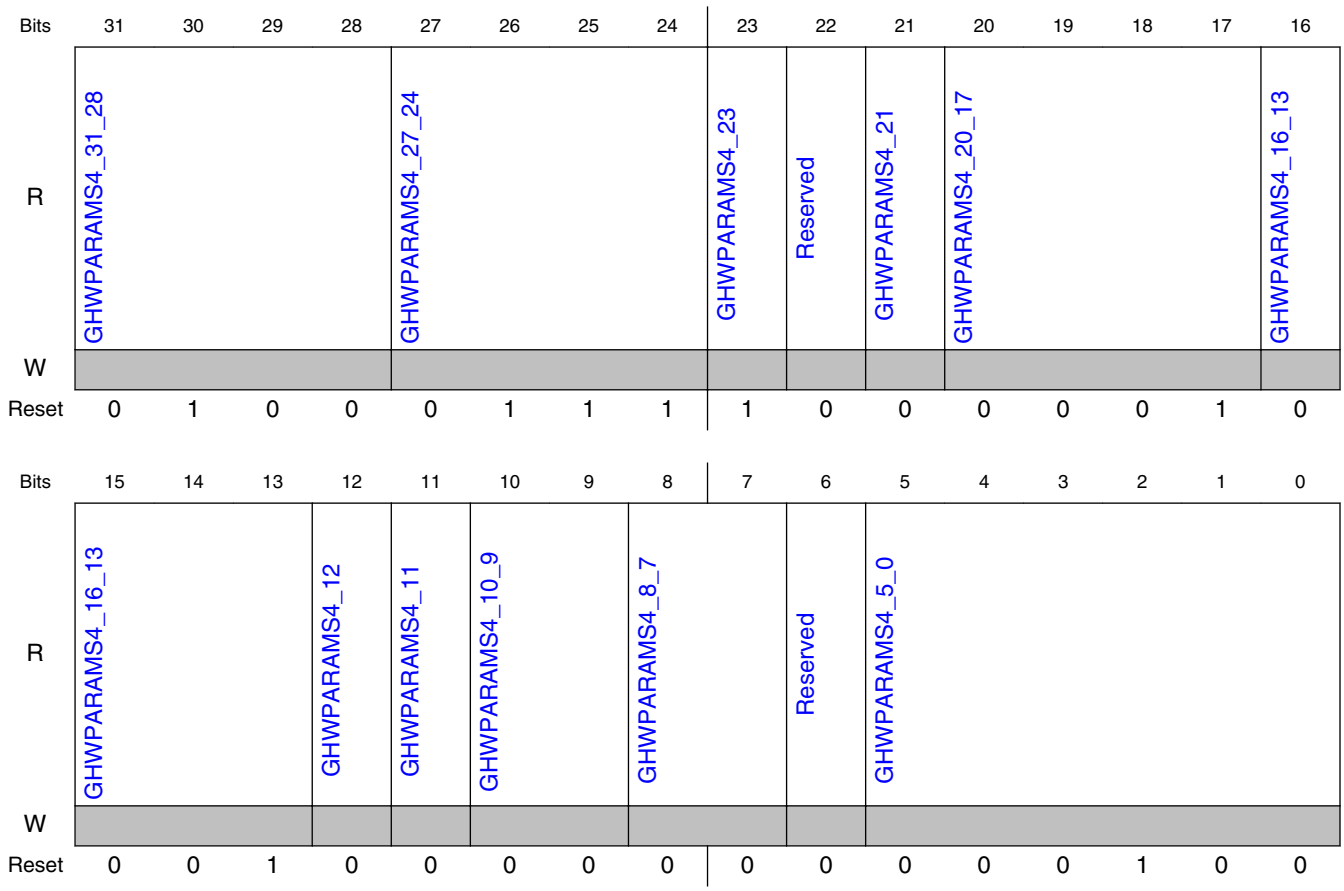
11.2.5.1.67.1 Offset

Register	Offset
GHWPARAMS4	C150h

11.2.5.1.67.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.2.5.1.67.3 Diagram



11.2.5.1.67.4 Fields

Field	Function
31-28 GHWPARAMS4_31_28	`DWC_USB3_BMU_LSP_DEPTH
27-24 GHWPARAMS4_27_24	`DWC_USB3_BMU_PTL_DEPTH-1
23 GHWPARAMS4_23	`DWC_USB3_EN_ISOC_SUPT
22 —	Reserved
21 GHWPARAMS4_21	`DWC_USB3_EXT_BUFF_CONTROL

Table continues on the next page...

Field	Function
20-17 GHWPARAMS4_20_17	`DWC_USB3_NUM_SS_USB_INSTANCES
16-13 GHWPARAMS4_16_13	`DWC_USB3_HIBER_SCRATCHBUFS Number of external scratchpad buffers the core requires to save its internal state in the device mode. Each buffer is assumed to be 4 KB. The scratchpad buffer array must have this many buffer pointers.
12 GHWPARAMS4_12	`DWC_USB3_EN_SSIC - 1'b0: if DWC_USB3_EN_SSIC == 0 - 1'b1: if DWC_USB3_EN_SSIC != 0 Note: When the DWC_USB3_NUM_SSIC_PORTS parameter is set to zero, this bit is Reserved.
11 GHWPARAMS4_11	`DWC_USB3_SSIC_NON_SNPS_MPHY This field indicates whether Synopsys M-PHY or a third-party M-PHY is used with SSIC ports. - 1'b0: Synopsys M-PHY - 1'b1: Third-party M-PHY Note: When the DWC_USB3_NUM_SSIC_PORTS parameter is set to zero, this bit is Reserved.
10-9 GHWPARAMS4_10_9	`DWC_USB3_SSIC_GEAR This field indicates DWC_USB3_SSIC_GEAR parameter value chosen by the user - 2'b00: Reserved - 2'b01: HS-G1 - 2'b10: HS-G2 - 2'b11: HS-G3 Note: When the DWC_USB3_NUM_SSIC_PORTS parameter is set to zero, this field is Reserved.
8-7 GHWPARAMS4_8_7	`DWC_USB3_NUM_SSIC_NUM_LANE This bit indicates `DWC_USB3_SSIC_NUM_LANE parameter value chosen by the user - 2'b00: 4 lane - 2'b01: 1 lane - 2'b10: 2 lane - 2'b11: Reserved Note: When the DWC_USB3_NUM_SSIC_PORTS parameter is set to zero, this field is Reserved.
6 —	Reserved
5-0 GHWPARAMS4_5_0	DWC_USB3_CACHE_TRBS_PER_TRANSFER

11.2.5.1.68 Global Hardware Parameters Register 5 (GHWPARAMS5)

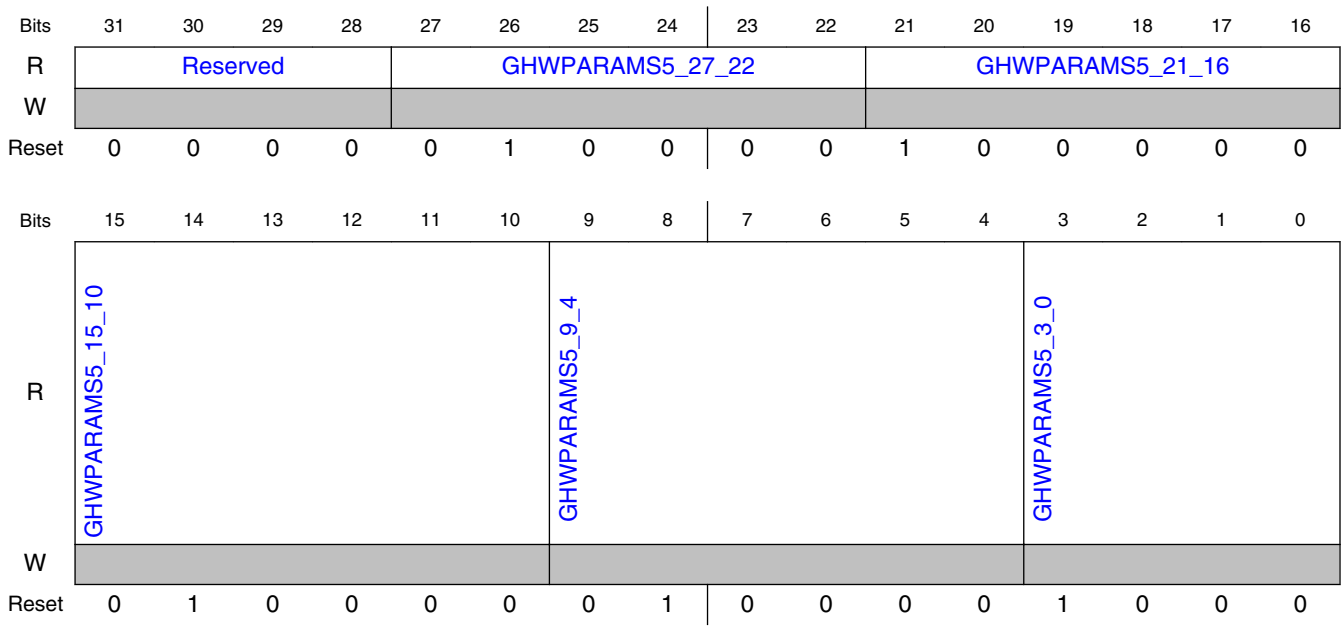
11.2.5.1.68.1 Offset

Register	Offset
GHWPARAMS5	C154h

11.2.5.1.68.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.2.5.1.68.3 Diagram



11.2.5.1.68.4 Fields

Field	Function
31-28 —	Reserved
27-22 GHWPARAMS5_27_22	`DWC_USB3_DFQ_FIFO_DEPTH
21-16 GHWPARAMS5_21_16	`DWC_USB3_DWQ_FIFO_DEPTH
15-10 GHWPARAMS5_15_10	`DWC_USB3_TXQ_FIFO_DEPTH
9-4 GHWPARAMS5_9_4	`DWC_USB3_RXQ_FIFO_DEPTH
3-0 GHWPARAMS5_3_0	`DWC_USB3_BMU_BUSGM_DEPTH

11.2.5.1.69 Global Hardware Parameters Register 6 (GHWPARAMS6)

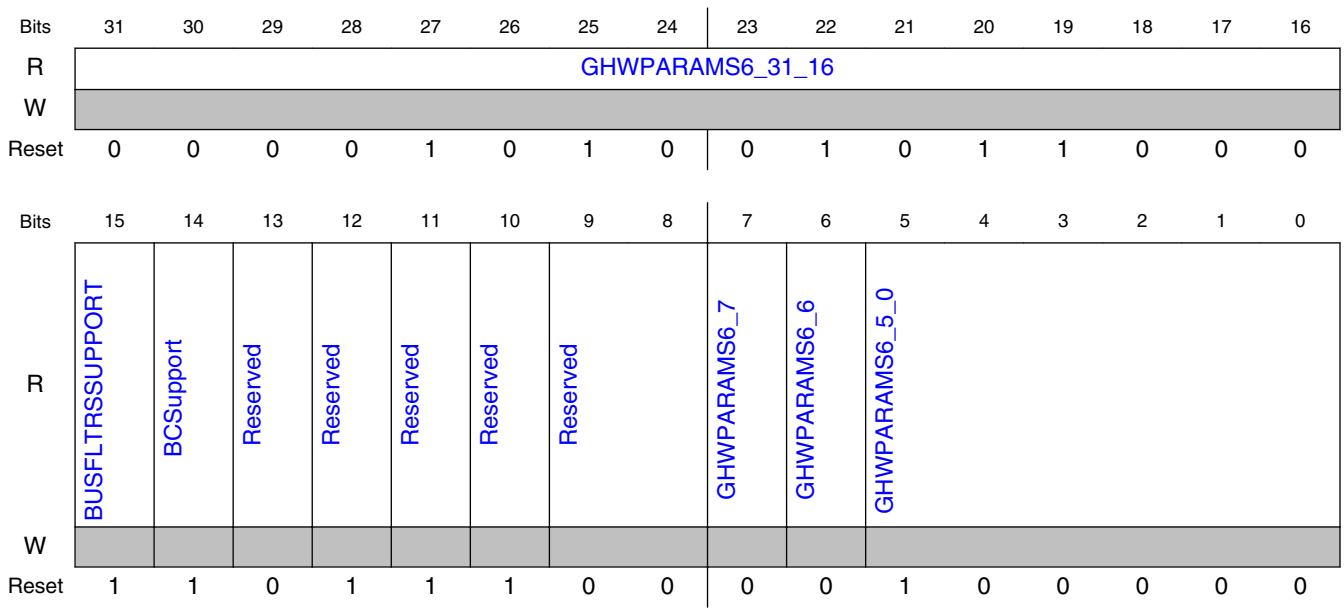
11.2.5.1.69.1 Offset

Register	Offset
GHWPARAMS6	C158h

11.2.5.1.69.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.2.5.1.69.3 Diagram



11.2.5.1.69.4 Fields

Field	Function
31-16 GHWPARAMS6_31_16	`DWC_USB3_RAM0_DEPTH
15 BUSFLTRSSUPPORT	`DWC_USB3_EN_BUS_FILTERS
14 BCSupport	`DWC_USB3_EN_BC
13 —	Reserved

Table continues on the next page...

Field	Function
12 —	Reserved
11 —	Reserved
10 —	Reserved
9-8 —	Reserved
7 GHWPARAMS6_7	`DWC_USB3_EN_FPGA
6 GHWPARAMS6_6	`DWC_USB3_EN_DBG_PORTS
5-0 GHWPARAMS6_5_0	`DWC_USB3_PSQ_FIFO_DEPTH

11.2.5.1.70 Global Hardware Parameters Register 7 (GHWPARAMS7)

11.2.5.1.70.1 Offset

Register	Offset
GHWPARAMS7	C15Ch

11.2.5.1.70.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.2.5.1.70.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GHWPARAMS7_31_16															
W																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GHWPARAMS7_15_0															
W																
Reset	0	0	0	1	0	0	0	1	1	1	1	0	1	1	0	0

11.2.5.1.70.4 Fields

Field	Function
31-16 GHWPARAMS7_31_16	`DWC_USB3_RAM2_DEPTH
15-0 GHWPARAMS7_15_0	`DWC_USB3_RAM1_DEPTH

11.2.5.1.71 Global High-Speed Port to Bus Instance Mapping Register - Low (GPRTBIMAP_HSLO)

11.2.5.1.71.1 Offset

Register	Offset
GPRTBIMAP_HSLO	C180h

11.2.5.1.71.2 Function

This register specifies the High Speed USB instance number to which each USB 3.0 port is connected. By default, USB 3.0 ports are evenly distributed among all HS USB instances. The software can program this register to specify how USB 3.0 ports are connected to HS USB instances. Register fields are read-write with respect to number of port instantiated. writeAsRead constraint is added to limit side effects for unused fields.

11.2.5.1.71.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BINUM8				BINUM7				BINUM6				BINUM5			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BINUM4				BINUM3				BINUM2				BINUM1			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.71.4 Fields

Field	Function
31-28 BINUM8	HS USB Instance Number for Port 8. Application-programmable ID field.
27-24 BINUM7	HS USB Instance Number for Port 7. Application-programmable ID field.
23-20 BINUM6	USB Instance Number for Port 6. Application-programmable ID field.
19-16 BINUM5	HS USB Instance Number for Port 5. Application-programmable ID field.
15-12 BINUM4	HS USB Instance Number for Port 4. Application-programmable ID field.
11-8 BINUM3	HS USB Instance Number for Port 3. Application-programmable ID field.
7-4 BINUM2	HS USB Instance Number for Port 2. Application-programmable ID field.
3-0 BINUM1	HS USB Instance Number for Port 1. Application-programmable ID field.

11.2.5.1.72 Global High-Speed Port to Bus Instance Mapping Register - High (GPRTBIMAP_HSHI)

11.2.5.1.72.1 Offset

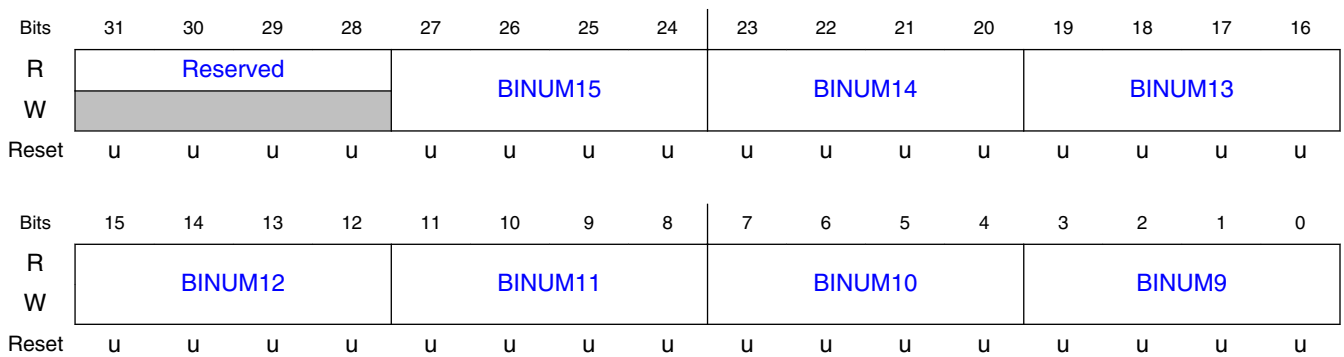
Register	Offset
GPRTBIMAP_HSHI	C184h

11.2.5.1.72.2 Function

This register specifies the High Speed USB instance number to which each USB 3.0 port is connected. By default, USB 3.0 ports are evenly distributed among all HS USB instances. The software can program this register to specify how USB 3.0 ports are connected to HS USB instances. Register fields are read-write with respect to number of port instantiated. writeAsRead constraint is added to limit side effects for unused fields.

Reset Mask: 0xF0000000

11.2.5.1.72.3 Diagram



11.2.5.1.72.4 Fields

Field	Function
31-28 —	Reserved
27-24 BINUM15	HS USB Instance Number for Port 15. Application-programmable ID field.
23-20 BINUM14	HS USB Instance Number for Port 14. Application-programmable ID field.
19-16 BINUM13	HS USB Instance Number for Port 13. Application-programmable ID field.
15-12 BINUM12	HS USB Instance Number for Port 12. SApplication-programmable ID field.
11-8 BINUM11	HS USB Instance Number for 11. Application-programmable ID field.
7-4 BINUM10	HS USB Instance Number for Port 10. Application-programmable ID field.
3-0 BINUM9	HS USB Instance Number for Port 9. Application-programmable ID field.

11.2.5.1.73 Global Full-Speed Port to Bus Instance Mapping Register - Low (GPRTBIMAP_FSLO)

11.2.5.1.73.1 Offset

Register	Offset
GPRTBIMAP_FSLO	C188h

11.2.5.1.73.2 Function

This register specifies the Full Speed USB instance number to which each USB 3.0 port is connected. By default, USB 3.0 ports are evenly distributed among all FS USB instances. Software can program this register to specify how USB 3.0 ports are connected to FS USB instances. - Register fields are read-write with respect to number of port instantiated. writeAsRead constraint is added to limit side effects for unused fields.

11.2.5.1.73.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BINUM8				BINUM7				BINUM6				BINUM5			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BINUM4				BINUM3				BINUM2				BINUM1			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.73.4 Fields

Field	Function
31-28 BINUM8	FS USB Instance Number for Port 8. Application-programmable ID field.
27-24 BINUM7	FS USB Instance Number for Port 7. Application-programmable ID field.
23-20 BINUM6	FS USB Instance Number for Port 6. Application-programmable ID field.
19-16	FS USB Instance Number for Port 5. Application-programmable ID field.

Table continues on the next page...

Field	Function
BINUM5	
15-12 BINUM4	FS USB Instance Number for Port 4. Application-programmable ID field.
11-8 BINUM3	FS USB Instance Number for Port 3. Application-programmable ID field.
7-4 BINUM2	FS USB Instance Number for Port 2. Application-programmable ID field.
3-0 BINUM1	FS USB Instance Number for Port 1. Application-programmable ID field.

11.2.5.1.74 Global Full-Speed Port to Bus Instance Mapping Register - High (GPRTBIMAP_FSHI)

11.2.5.1.74.1 Offset

Register	Offset
GPRTBIMAP_FSHI	C18Ch

11.2.5.1.74.2 Function

This register specifies the Full Speed USB instance number to which each USB 3.0 port is connected. By default, USB 3.0 ports are evenly distributed among all FS USB instances. Software can program this register to specify how USB 3.0 ports are connected to FS USB instances. - Register fields are read-write with respect to number of port instantiated. writeAsRead constraint is added to limit side effects for unused fields.

Reset Mask: 0xF0000000

11.2.5.1.74.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				BINUM15				BINUM14				BINUM13			
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BINUM12				BINUM11				BINUM10				BINUM9			
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

11.2.5.1.74.4 Fields

Field	Function
31-28 —	Reserved
27-24 BINUM15	FS USB Instance Number for Port 15. Application-programmable ID field
23-20 BINUM14	FS USB Instance Number for Port 14. Application-programmable ID field
19-16 BINUM13	FS USB Instance Number for Port 13. Application-programmable ID field
15-12 BINUM12	FS USB Instance Number for Port 12. Application-programmable ID field
11-8 BINUM11	FS USB Instance Number for Port 11. Application-programmable ID field
7-4 BINUM10	FS USB Instance Number for Port 10. Application-programmable ID field
3-0 BINUM9	FS USB Instance Number for Port 9. Application-programmable ID field.

11.2.5.1.75 Global Error Injection 1 Control Register (GERRINJCTL_1)

11.2.5.1.75.1 Offset

Register	Offset
GERRINJCTL_1	C194h

11.2.5.1.75.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.75.3 Fields

Field	Function
31-0 —	Reserved

11.2.5.1.76 Global Error Injection 2 Control Register (GERRINJCTL_2)

11.2.5.1.76.1 Offset

Register	Offset
GERRINJCTL_2	C198h

11.2.5.1.76.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.76.3 Fields

Field	Function
31-0 —	Reserved

11.2.5.1.77 Global User Control Register 2 (GUCTL2)

11.2.5.1.77.1 Offset

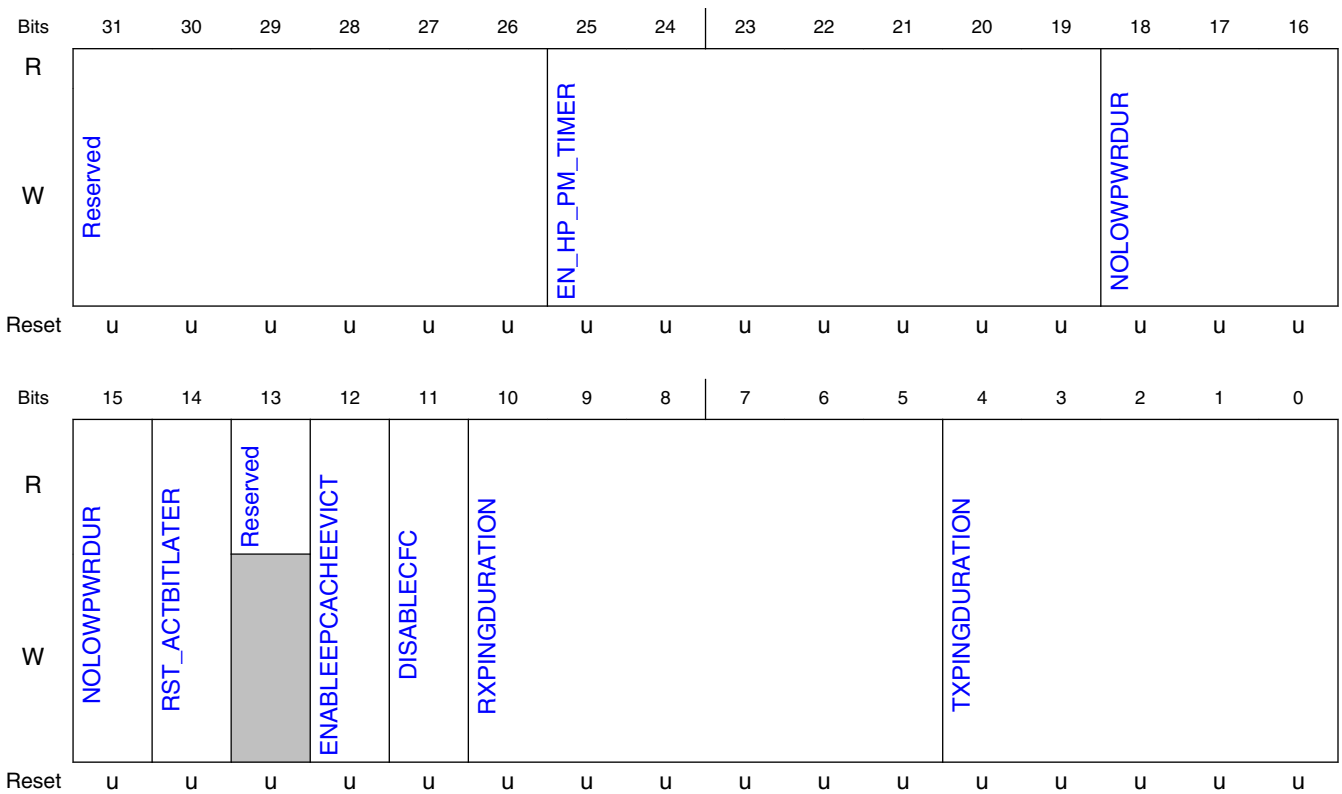
Register	Offset
GUCTL2	C19Ch

11.2.5.1.77.2 Function

This register provides a few options for the software to control the core behavior in the Host and device mode.

Reset Mask: 0x3FC008000

11.2.5.1.77.3 Diagram



11.2.5.1.77.4 Fields

Field	Function
31-26 —	Reserved

Table continues on the next page...

Field	Function
25-19 EN_HP_PM_TIMER	This register field is used to set new HP and PM timers. - To enable PM timer, set GUCTL2[19] bit as 1. - To enable HP timer, set GUCTL2[20] bit as 1. Default value is 4us, when new HP timer is enabled default value is 11us. Use GUCTL2[25:21] to specify HP timer value in microseconds.
18-15 NOLOWPWRDUR	No Low Power Duration This bit is applicable for device mode only and is ignored in host mode. After starting a transfer on an ESS ISOC endpoint, the application must program these bits. Each count represents the duration in terms of milli seconds. For example, a value of 3 represents 3 ms. **NOTE:** • To disable this feature, set this field to 4'b0. • These bits are applicable only in device mode and ignored in host mode. • Some xHCI hosts do not send ITPs when performing ISOC transfers when the link enters U1/U2 low power states. This causes the device to lose frame synchronization over a period of time resulting in ISOC packets being dropped.
14 RST_ACTBITLATER	Enable clearing of the command active bit for the ENDXFER command after the command execution is completed. This bit is valid in device mode only.
13 —	Reserved for future use
12 ENABLEEPCACHEEVICT	Enable Evicting Endpoint Cache after Flow Control for bulk endpoints In 3.00a release, a performance enhancement was done to keep the non-stream capable bulk IN endpoint in cache after flow control. Setting this bit will disable this enhancement. This should be set only for debug purpose.
11 DISABLECFC	Disable xHCI Errata Feature Contiguous Frame ID Capability This field controls the xHCI Errata feature Contiguous FrameID capability. When set, the xHCI HCCPARAMS1 bit 11 will be set to 0 indicating that CFC is not supported. Disable this feature only if your application cannot tolerate Misssed Service Error events for Isoc transfers, and your system latencies are large to cause Missed Service errors even if the software is following the Isochronous Thresholding rules.
10-5 RXPINGDURATION	Recieve Ping Maximum Duration This field is relevant to Host mode and controls the maximum duration of received LFPS to be treated as a Ping LFPS. The Max duration of the Ping LFPS is controlled by programming this value and is in terms of 8 ns granularity. Eg: A value of 32 indicates 256 ns.
4-0 TXPINGDURATION	Transmit Ping Maximum Duration This field is relevant to Device mode and controls the maximum duration for which the controller should instruct the PHY to transmit a Ping LFPS. The duration of the Ping LFPS is controlled by programming this value and is in terms of 8 ns granularity. Eg: A value of 13 indicates 104 ns.

11.2.5.1.78 Global USB2 PHY Configuration Register (GUSB2PHYCFG)

11.2.5.1.78.1 Offset

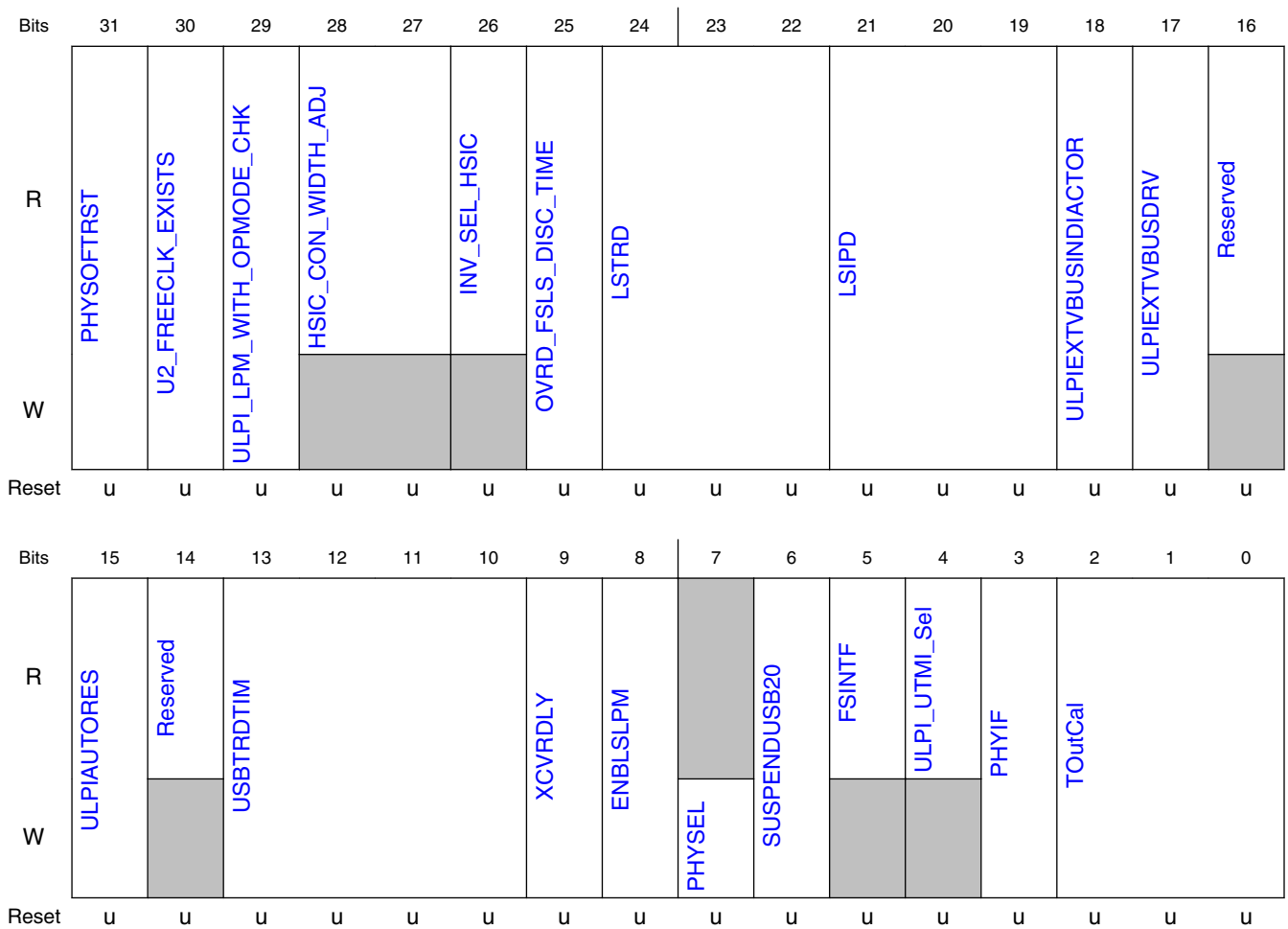
Register	Offset
GUSB2PHYCFG	C200h

11.2.5.1.78.2 Function

The application must program this register before starting any transactions on either the SoC bus or the USB.

Reset Mask:0x2014000

11.2.5.1.78.3 Diagram



11.2.5.1.78.4 Fields

Field	Function
31 PHYSOFTRST	UTMI PHY Soft Reset Causes the usb2phy_reset signal to be asserted to reset a UTMI PHY.
30 U2_FREECLK_EXISTS	U2_FREECLK_EXISTS Specifies whether your USB 2.0 PHY provides a free-running PHY clock, which is active when the clock control input is active. Specifies whether your USB 2.

Table continues on the next page...

Field	Function
	free-running PHY clock, which is active when the clock control input is active. If your USB 2.0 PHY provides a free-running PHY clock, it must be connected to the utmi_clk[0] input. The remaining utmi_clk[n] must be connected to the respective port clocks. The core uses the Port-0 clock for generating the internal mac2 clock. - 1'b0: USB 2.0 free clock does not exist - 1'b1: USB 2.0 free clock exists Note: When the core is configured as device-only (DWC_USB3_MODE = 0), do not set this bit to 1.
29 ULPI_LPM_WITH_OPMODE_CHK	Support the LPM over ULPI without NOPID token to the ULPI PHY. If this bit is set, the ULPI PHY is expected to qualify the EXT PID with OPMODE=2'b00 for LPM and not treat it as a NOPID. Check with your PHY vendor about your PHY behavior. This bit is valid only when the DWC_USB3_HSPHY_INTERFACE parameter is 2 or 3. - 1'b0: A NOPID is sent before sending an EXTPID for LPM; - 1'b1: An EXTPID is sent without previously sending a NOPID; Note: This bit is valid only in host mode. This bit should be '0' for Synopsys PHY.
28-27 HSIC_CON_WIDTH_ADJ	This bit is used in the HSIC device mode of operation.
26 INV_SEL_HSIC	The application driver uses this bit to control the HSIC enable/disable function.
25 OVRD_FSLS_DISC_TIME	Overriding the FS/LS Disconnect Time • If this value is 0, the FS/LS disconnect time is set to 2.5us as per the USB specification. • If this value is non-0, the disconnect detection time is set to 32us. Normally, this value is set to 0. However, if the USB 2.0 PHYs introduce noise on the UTMI linestate and cause SE0 glitches longer than 2.5us, then a false disconnect condition may get triggered. To avoid interoperability issues with these PHYs, this bit can be set to 1.
24-22 LSTRD	LS Turnaround Time This field indicates the value of the Rx-to-Tx packet gap for LS devices.
21-19 LSIPD	LS Inter-Packet Time This field indicates the value of Tx-to-Tx packet gap for LS devices.
18 ULPIEXTVBUSINDIACTOR	ULPI External VBUS Indicator Indicates the ULPI PHY VBUS over-current indicator.
17 ULPIEXTVBUSDRV	ULPI External VBUS Drive Selects supply source to drive 5V on VBUS, in the ULPI PHY.
16 —	Reserved
15 ULPIAUTORES	ULPI Auto Resume Sets the AutoResume bit in Interface Control register on the ULPI PHY.
14 —	Reserved
13-10 USBTRDTIM	USB 2.0 Turnaround Time
9 XCVRDLY	Transceiver Delay Enables a delay between the assertion of the UTMI/ULPI Transceiver Select signal (for HS) and the assertion of the TxValid signal during a HS Chirp.
8	Enable utmi_sleep_n and utmi_l1_suspend_

Table continues on the next page...

Field	Function
ENBLSLPM	The application uses this bit to control utmi_sleep_n and utmi_l1_suspend_n assertion to the PHY in the L1 state..
7 PHYSEL	USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver Select
6 SUSPENDUSB20	Suspend USB2.0 HS/FS/LS PHY
5 FSINTF	Full-Speed Serial Interface Select The application uses this bit to select a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface.
4 ULPI_UTMI_Sel	ULPI or UTMI+ Select The application uses this bit to select a UTMI+ or ULPI Interface.
3 PHYIF	PHY Interface If UTMI+ is selected, the application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface.
2-0 TOutCal	HS/FS Timeout Calibration The number of PHY clocks, as indicated by the application in this field, is multiplied by a bit-time factor; this factor is added to the high-speed/full-speed interpacket timeout duration in the core to account for additional delays introduced by the PHY.

11.2.5.1.79 Global USB 2.0 ULPI PHY Vendor Control Register (GUSB2PHYACC_ULPI)

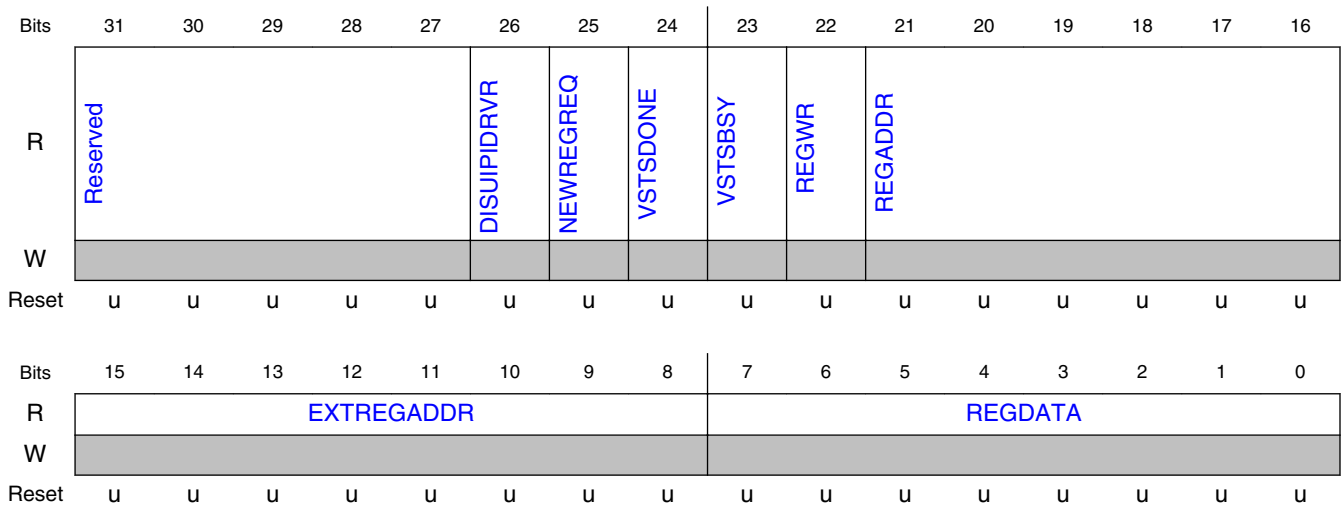
11.2.5.1.79.1 Offset

Register	Offset
GUSB2PHYACC_ULPI	C280h

11.2.5.1.79.2 Function

Reset Mask:0xF8000000

11.2.5.1.79.3 Diagram



11.2.5.1.79.4 Fields

Field	Function
31-27 —	Reserved
26 DISUIPIDRVR	DISUIPIDRVR
25 NEWREGREQ	New Register Request New Register Request The application sets this bit for a new vendor control access..
24 VSTSDONE	VSTSDONE
23 VSTSBSY	VSTSBSY
22 REGWR	Register Write The application sets this bit for register writes and clears it for register reads. **NOTE:** This bit is applicable for ULPI register read/write access only.
21-16 REGADDR	Register Address The 6-bit PHY register address for immediate PHY Register Set access. Set to 6'h2F for Extended PHY Register Set access. **NOTE:** These bits are applicable for ULPI only.
15-8 EXTREGADDR	EXTREGADDR
7-0 REGDATA	REGDATA

11.2.5.1.80 Global USB 3.0 PIPE Control Register (GUSB3PIPECTL)

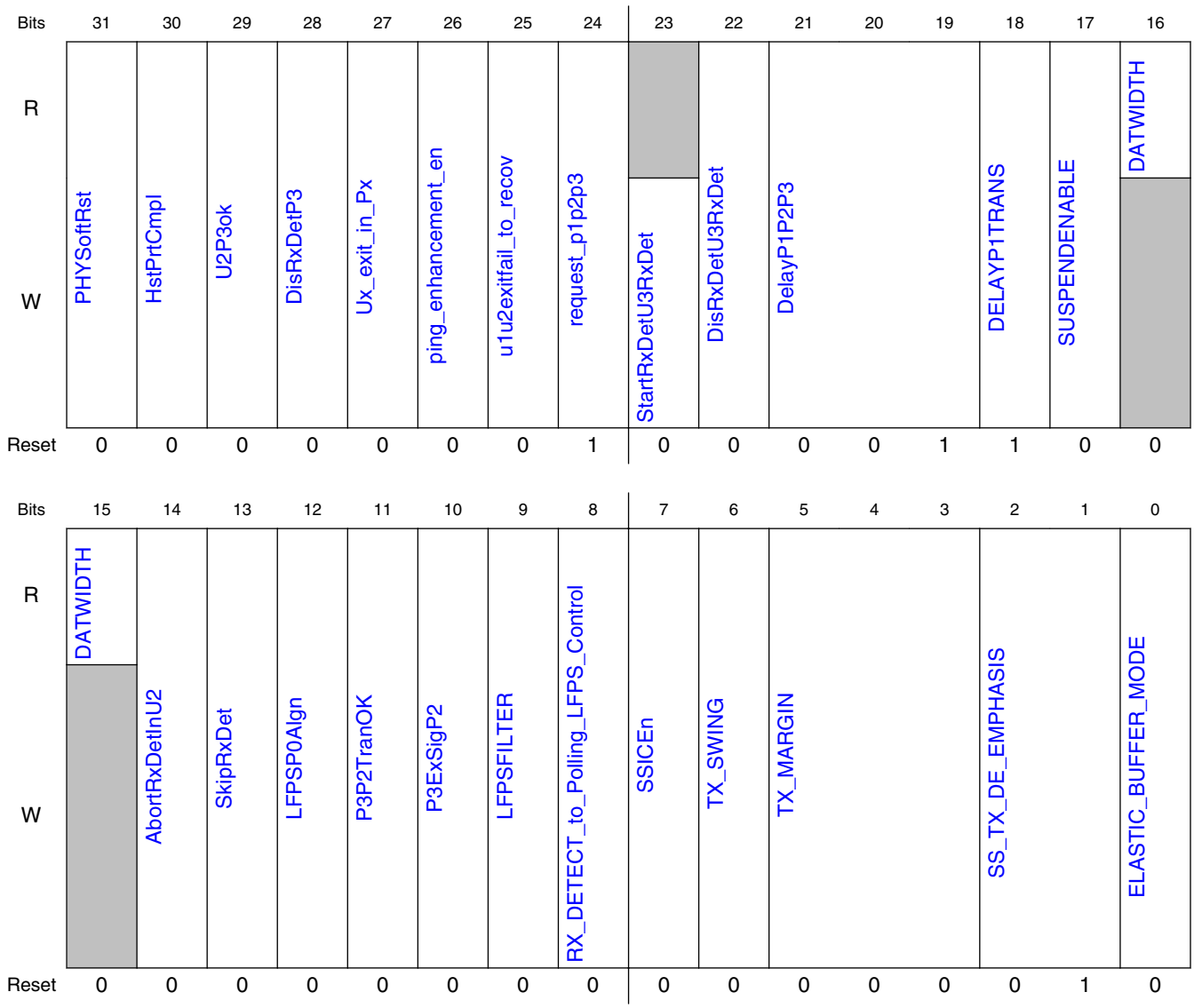
11.2.5.1.80.1 Offset

Register	Offset
GUSB3PIPECTL	C2C0h

11.2.5.1.80.2 Function

The application uses this register to configure the USB3 PHY and PIPE interface. Device-only configuration requires only one register. In Host mode, registers are implemented for each port. Note: - GUSB3PIPECTLn registers are not applicable for USB 2.0-only mode.

11.2.5.1.80.3 Diagram



11.2.5.1.80.4 Fields

Field	Function
31 PHYSoftRst	USB3 PHY Soft Reset After setting this bit to '1', the software needs to clear this bit.
30 HstPrtCmpl	This feature tests the PIPE PHY compliance patterns without having to have a test fixture on the USB 3.
29 U2P3ok	P3 OK for U2 0: During link state U2/SS.Inactive, put PHY in P2 (Default) 1: During link state U2/SS.Inactive, put PHY in P3.

Table continues on the next page...

Field	Function
28 DisRxDetP3	Disabled Receiver Detection in P3 0:If PHY is in P3 and controller needs to perform receiver detection, The controller performs receiver detection in P3. (Default) 1: If PHY is in P3 and controller needs to perform receiver detection, The controller changes the PHY power state to P2 and then performs receiver detection. After receiver detection, the cores changes PHY power state to P3.
27 Ux_exit_in_Px	Ux Exit in Px 0: The core does U1/U2/U3 exit in PHY power state P0 (default behavior). 1: The controller does U1/U2/U3 exit in PHY power state P1/P2/P3 respectively.
26 ping_enhancement_en	Ping Enhancement Enable When set, the Downstream port U1 ping receive timeout becomes 500 ms instead of 300 ms. Minimum Ping.LFPS receive duration is 8 ns (one mac3_clk). This field is valid for the downstream port only.
25 u1u2exitfail_to_recov	U1U2exitfail to Recovery When set, and U1/U2 LFPS handshake fails, the LTSSM transitions from U1/U2 to Recovery instead of SS Inactive. If Recovery fails, then the LTSSM can enter SS.Inactive. This is an enhancement only. It prevents interoperability issue if the remote link does not do proper handshake.
24 request_p1p2p3	Always Request P1/P2/P3 for U1/U2/U3 When set, the core always requests PHY power change from P0 to P1/P2/P3 during U0 to U1/U2/U3 transition. If this bit is 0, and immediate Ux exit (remotely initiated, or locally initiated) happens, the controller does not request P1/P2/P3 power state change.
23 StartRxDetU3RxDet	Start Receiver Detection in U3/Rx.Detect If DWC_USB3_GUSB3PIPECTL_INIT[22] is set, and the link is in either U3 or Rx.Detect state, the core starts receiver detection on the rising edge of this bit. This can only be used for Downstream ports. This bit must be set to '0' for Upstream ports. This feature must not be enabled for normal operation.
22 DisRxDetU3RxDet	Disable Receiver Detection in U3/Rx.Detect Disable Receiver Detection in U3/Rx.Det When set, the core does not handle receiver detection in either U3 or Rx.Detect states. DWC_USB3_GUSB3PIPECTL_INIT[23] must be used to start receiver detection manually. This bit can only be used for the downstream port. This bit must be set to "0" for Upstream ports. This feature must not be enabled for normal operation.
21-19 DelayP1P2P3	Delay P1P2P3 Delay P0 to P1/P2/P3 request when entering U1/U2/U3 until (DWC_USB3_GUSB3PIPECTL_INIT[21:19]*8) 8B10B error occurs, or Pipe3_RxValid drops to 0.
18 DELAYP1TRANS	Delay PHY Power It change from P0 to P1/P2/P3 when link state changing from U0 to U1/U2/U3 respectively. 1'b1: When entering U1/U2/U3, delay the transition to P1/P2/P3 until the pipe3 signals, Pipe3_RxElecIdle is 1 and pipe3_RxValid is 0. 1'b0: When entering U1/U2/U3, transition to P1/P2/P3 without checking for Pipe3_RxElecIdle and pipe3_RxValid.
17 SUSPENDENABLE	Suspend USB3 SS PHY When set, and if Suspend conditions are valid, the USB 3.0 PHY enters Suspend mode. For DRD configurations, it is recommended that this bit is set to '0' during coreConsultant configuration. If it is set to '1', then the application must clear this bit after power-on reset. Application needs to set it to '1' after the controller initialization is completed. For all other configurations, this bit can be set to '1' during controller configuration.
16-15 DATWIDTH	PIPE Data Width 2'b00: 32 bits

Table continues on the next page...

Field	Function
	2'b01: 16 bits - One clock after reset, these bits receive the value seen on the pipe3_DataBusWidth. The simulation testbench uses the coreConsultant parameter to configure the VIP. These bits in the coreConsultant parameter must match your PHY data width and the pipe3_DataBusWidth port.
14 AbortRxDetInU2	Abort Rx Detect in U2 When set, and the link state is U2, then the core will abort receiver detection if it receives U2 exit LFPS from the remote link partner. This bit is for the downstream port only.
13 SkipRxDet	Skip Rx Detect When set, the core skips Rx Detection if pipe3_RxElecIdle is low.
12 LFPSP0Algn	LFPS P0 Align When set, the core deasserts LFPS transmission on the clock edge that it requests Phy power state 0 when exiting U1, U2, or U3 low power states. Otherwise, LFPS transmission is asserted one clock earlier. The controller requests symbol transmission two pipe3_rx_pclks periods after the PHY asserts PhyStatus as a result of the PHY switching from P1 or P2 state to P0 state.
11 P3P2TranOK	P3 P2 Transitions OK When set, the core transitions directly from Phy power state P2 to P3 or from state P3 to P2. When not set, P0 is always entered as an intermediate state during transitions between P2 and P3, as defined in the PIPE3 Specification. According to the PIPE3 Specification, any direct transition between P3 and P2 is illegal
10 P3ExSigP2	P3 Exit Signal in P2 When this bit is set, the core always changes the PHY power state to P2, before attempting a U3 exit handshake.
9 LFPSFILTER	LFPS Filter When set, filter LFPS reception with pipe3_RxValid in PHY power state P0, that is, ignore LFPS reception from the PHY unless both pipe3_Rxelecidle and pipe3_RxValid are deasserted.
8 RX_DETECT_to_Polling_LFPS_Control	RX_DETECT to Polling LFPS Control 1'b0 (Default): Enables a 400us delay to start Polling LFPS after RX_DETECT. This allows VCM offset to settle to a proper level. 1'b1: Disables the 400us delay to start Polling LFPS after RX_DETECT. During controller certification with third party PHY it is observed that the PHY is not able to meet the Tx AC common mode voltage active (VTX-CM-ACPP_ACTIVE <100mv) if the link starts polling within 80us from the time rx.detect is performed. To meet this VTX-CM-ACPP_ACTIVE specification, the polling must be delayed further. If the PHY does not have issue then they can set this bit to 1 which allows polling to start within 80us.
7 SSICEn	USB3 SSIC Enable This bit is valid only when coreConsultant parameter DWC_USB3_EN_SSIC=1 else this bit needs to be set to 1'b0.
6 TX_SWING	Tx Swing
5-3 TX_MARGIN	Tx Margin [2:0]
2-1 SS_TX_DE_EMPHASIS	Tx Deemphasis The value driven to the PHY is controlled by the LTSSM during USB3 Compliance mode.

Table continues on the next page...

Field	Function
0 ELASTIC_BUFFER_MODE	Elastic Buffer Mode

11.2.5.1.81 Global Transmit FIFO Size Register (GTXFIFOSIZ0 - GTXFIFOSIZ7)

11.2.5.1.81.1 Offset

For a = 0 to 7:

Register	Offset
GTXFIFOSIZa	C300h + (a × 4h)

11.2.5.1.81.2 Function

This register specifies the RAM start address and depth (both in MDWIDTH-bit words) for each implemented TxFIFO.

11.2.5.1.81.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TXFSTADDR_N															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TXFDEP_N															
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0

11.2.5.1.81.4 Fields

Field	Function
31-16 TXFSTADDR_N	Transmit FIFOn RAM Start Address This field contains the memory start address for TxFIFOn in MDWIDTH-bit words.
15-0 TXFDEP_N	TXFIFO depth This field contains the depth of TxFIFOn in MDWIDTH-bit words.

11.2.5.1.82 Global Receive FIFO Size Register (GRXFIFOSIZ0 - GRXFIFOSIZ2)

11.2.5.1.82.1 Offset

Register	Offset
GRXFIFOSIZ0	C380h
GRXFIFOSIZ1	C384h
GRXFIFOSIZ2	C388h

11.2.5.1.82.2 Function

This register specifies the RAM start address and depth (both in MDWIDTH-bit words) for each implemented RxFIFO.

11.2.5.1.82.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RXFSTADDR_N															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RXFDEP_N															
Reset	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1

11.2.5.1.82.4 Fields

Field	Function
31-16 RXFSTADDR_N	RxFIFOn RAM Start Address This field contains the memory start address for RxFIFOn in MDWIDTH-bit words.
15-0 RXFDEP_N	RxFIFO Depth This field contains the depth of RxFIFOn in MDWIDTH-bit words.

11.2.5.1.83 Global Event Buffer Address Low Register (GEVNTADRLO)

11.2.5.1.83.1 Offset

Register	Offset
GEVNTADRLO	C400h

11.2.5.1.83.2 Function

This is an alternate register for the GEVNTADRn register.

11.2.5.1.83.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EVNTADRLO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EVNTADRLO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.83.4 Fields

Field	Function
31-0 EVNTADRLO	Event Buffer Address Holds the lower 32 bits of start address of the external memory for the Event Buffer.

11.2.5.1.84 Global Event Buffer Address High Register (GEVNTADRHI)

11.2.5.1.84.1 Offset

Register	Offset
GEVNTADRHI	C404h

11.2.5.1.84.2 Function

This is an alternate register for the GEVNTADRn register.

11.2.5.1.84.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EVNTADRHI															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EVNTADRHI															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.84.4 Fields

Field	Function
31-0 EVNTADRHI	Event Buffer Address Holds the higher 32 bits of start address of the external memory for the Event Buffer.

11.2.5.1.85 Global Event Buffer Size Register (GEVNTSIZ)

11.2.5.1.85.1 Offset

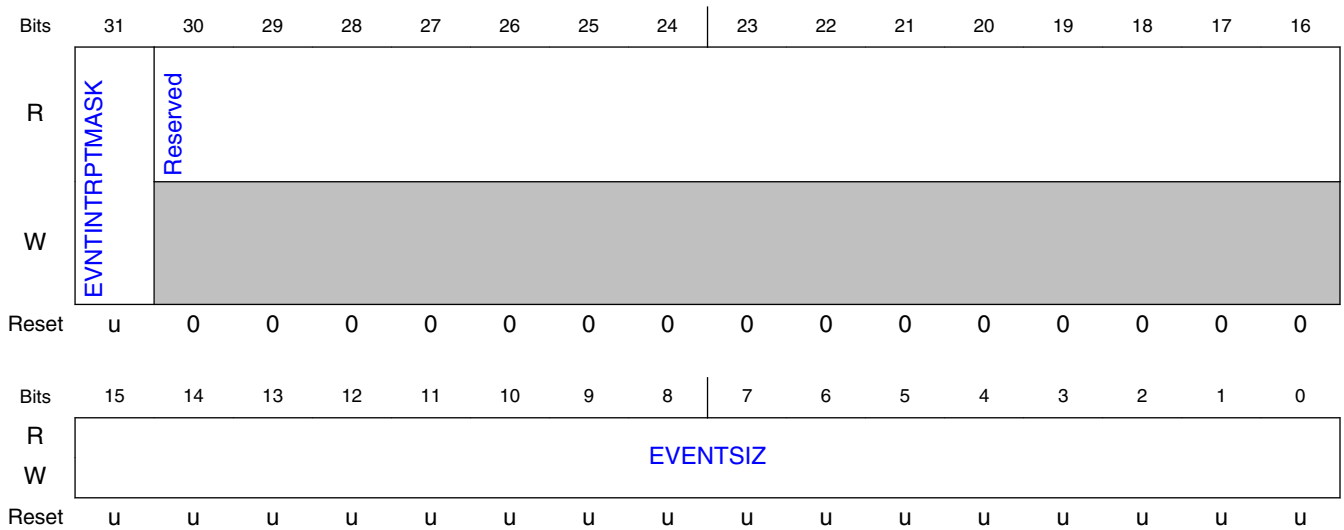
Register	Offset
GEVNTSIZ	C408h

11.2.5.1.85.2 Function

This register holds the Event Buffer Size and the Event Interrupt Mask bit. During power-on initialization, software must initialize the size with the number of bytes allocated for the Event Buffer. The Event Interrupt Mask will mask the interrupt, but events are still queued. After configuration, software must preserve the Event Buffer Size value when changing the Event Interrupt Mask.

Reset Mask:0x7FFF0000

11.2.5.1.85.3 Diagram



11.2.5.1.85.4 Fields

Field	Function
31 EVNTINTRPTMASK	Event Interrupt Mask.
30-16 —	Reserved
15-0 EVENTSIZ	Event Buffer Size in Bytes Holds the size of the Event Buffer in bytes; must be a multiple of four.

11.2.5.1.86 Global Event Buffer Count Register (GEVNTCOUNT)

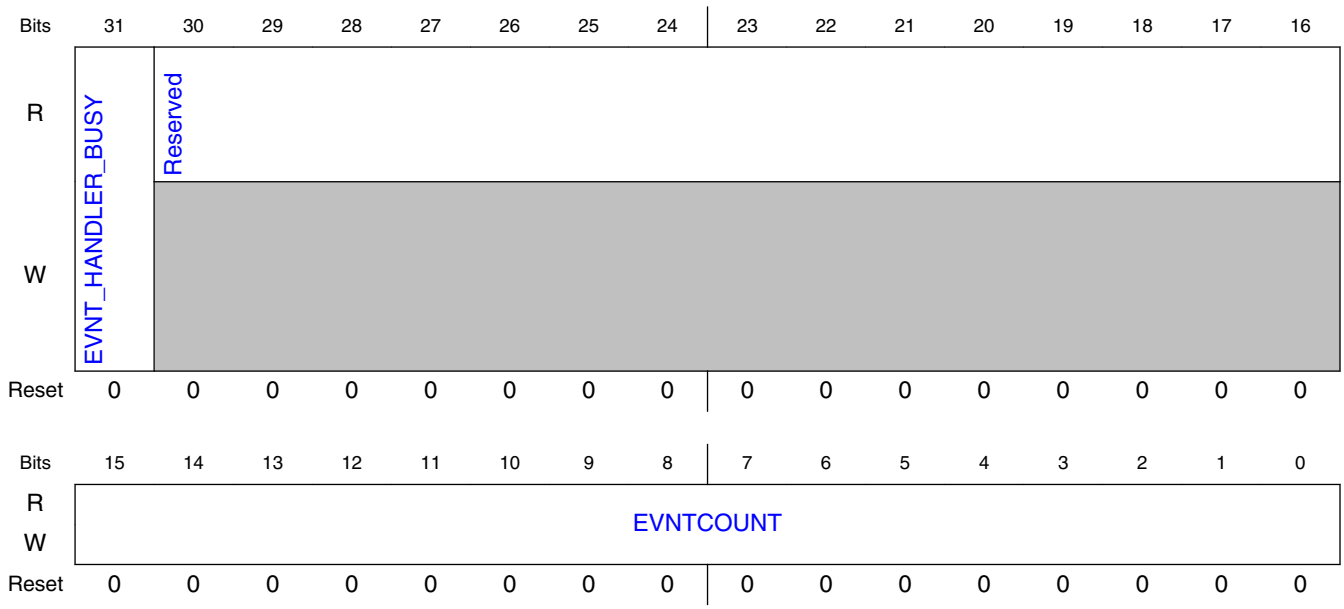
11.2.5.1.86.1 Offset

Register	Offset
GEVNTCOUNT	C40Ch

11.2.5.1.86.2 Function

This register holds the number of valid bytes in the Event Buffer.

11.2.5.1.86.3 Diagram



11.2.5.1.86.4 Fields

Field	Function
31 EVNT_HANDLER_BUSY	Event Handler Busy Device software event handler busy indication.
30-16 —	Reserved
15-0 EVNTCOUNT	Event Count When read, returns the number of valid events in the Event Buffer (in bytes).

11.2.5.1.87 Global Hardware Parameters Register 8 (GHWPARAMS8)

11.2.5.1.87.1 Offset

Register	Offset
GHWPARAMS8	C600h

11.2.5.1.87.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.2.5.1.87.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GHWPARAMS8_32_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GHWPARAMS8_32_0															
W																
Reset	0	0	0	0	1	0	1	0	0	1	0	1	1	0	0	0

11.2.5.1.87.4 Fields

Field	Function
31-0 GHWPARAMS8_32_0	`DWC_USB3_DCACHE_DEPTH_INFO

11.2.5.1.88 Global User Control Register 3 (GUCTL3)

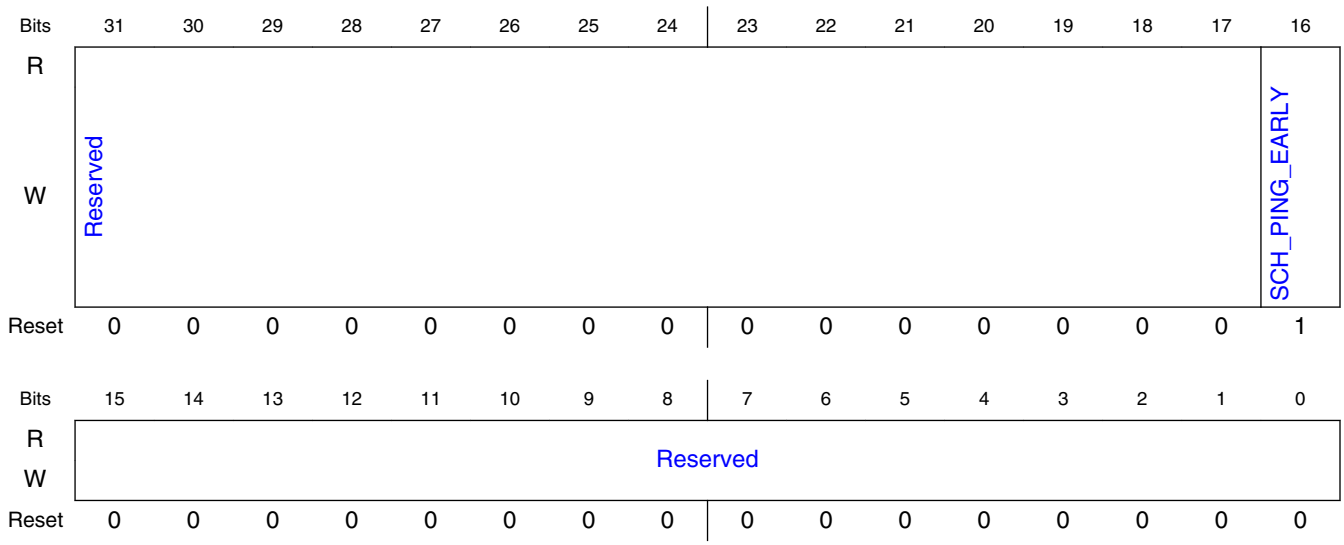
11.2.5.1.88.1 Offset

Register	Offset
GUCTL3	C60Ch

11.2.5.1.88.2 Function

This register provides a few options for the software to control the controller behavior in the Host mode. Most of the options are used to improve host inter-operability with different devices.

11.2.5.1.88.3 Diagram



11.2.5.1.88.4 Fields

Field	Function
31-17 —	Reserved
16 SCH_PING_EARLY	Enable SuperSpeed Ping Transaction Packet scheduling early in the microframe. This bit is valid in Host mode only.
15-0 —	Reserved

11.2.5.1.89 Global Device TX FIFO DMA Priority Register (GTXFIFOPRIDEV)

11.2.5.1.89.1 Offset

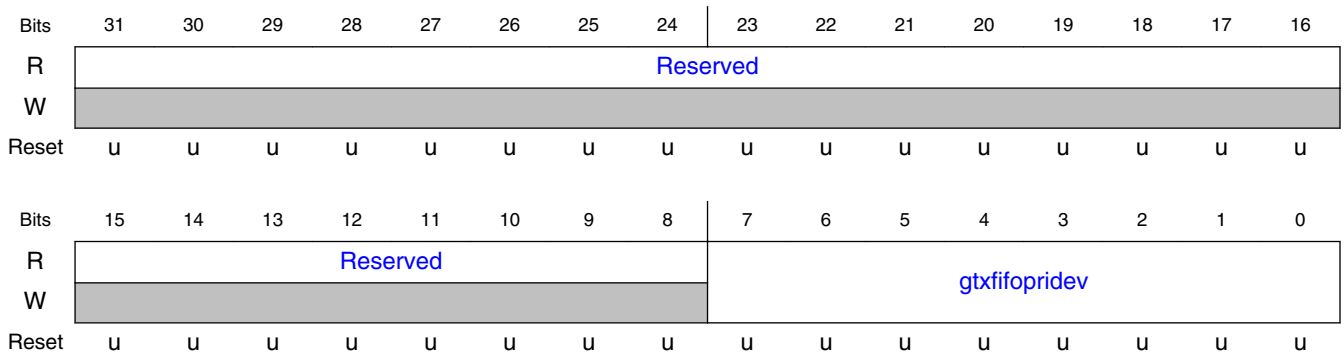
Register	Offset
GTXFIFOPRIDEV	C610h

11.2.5.1.89.2 Function

This register specifies the relative DMA priority level among the Device TXFIFOs (one per IN endpoint).

Reset Mask: 0xFFFF00

11.2.5.1.89.3 Diagram



11.2.5.1.89.4 Fields

Field	Function
31-8 —	Reserved
7-0 gtxfifopridev	Device TxFIFO priority

11.2.5.1.90 Global Host TX FIFO DMA Priority Register (GTXFIFOPRIHST)

11.2.5.1.90.1 Offset

Register	Offset
GTXFIFOPRIHST	C618h

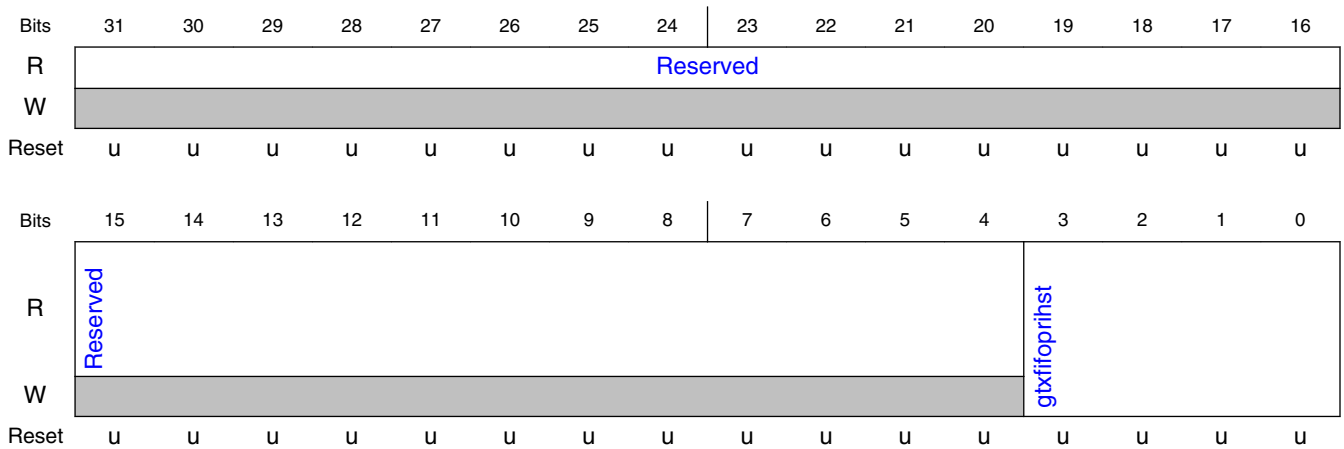
11.2.5.1.90.2 Function

This register specifies the relative DMA priority level among the Host TXFIFOs (one per USB bus instance) within the associated speed group (SS or HS/FSLS).

Reset Mask: 0xFFFF0

Reset Default Value: 0x0

11.2.5.1.90.3 Diagram



11.2.5.1.90.4 Fields

Field	Function
31-4 —	Reserved
3-0 gtxfifoprihst	Host TxFIFO priority

11.2.5.1.91 Global Host RX FIFO DMA Priority Register (GRXFIFOPRIHST)

11.2.5.1.91.1 Offset

Register	Offset
GRXFIFOPRIHST	C61Ch

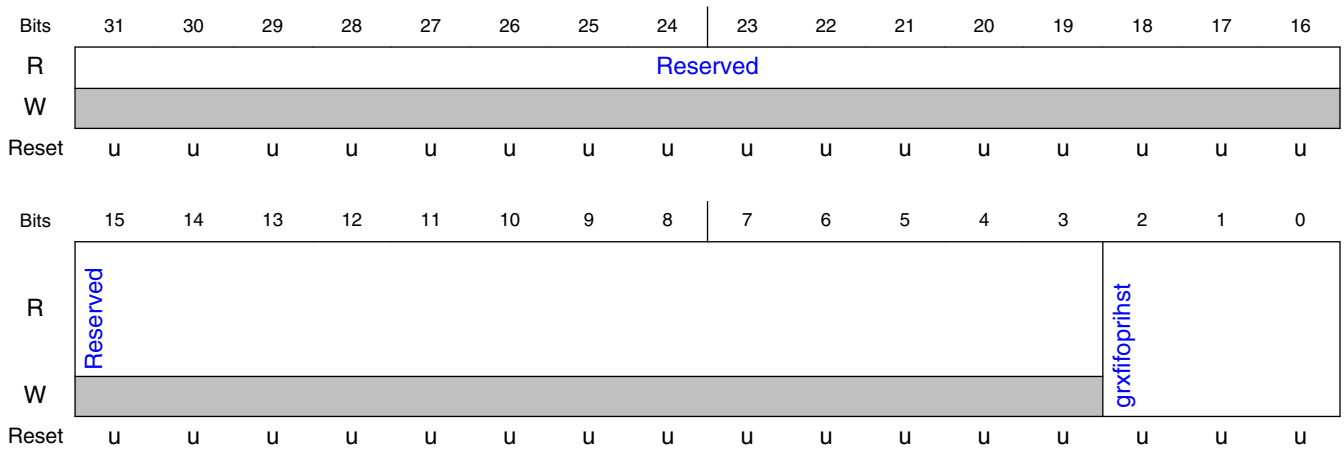
11.2.5.1.91.2 Function

This register specifies the relative DMA priority level among the Host RXFIFOs (one per USB bus instance) within the associated speed group (SS or HS/FSLS).

Reset Mask: 0x7FFF8

Reset Default Value: 0x0

11.2.5.1.91.3 Diagram



11.2.5.1.91.4 Fields

Field	Function
31-3 —	Reserved
2-0 grxfifoprihst	Host RxFIFO priority

11.2.5.1.92 Global Host Debug Capability DMA Priority Register (GFIFOPRIDBC)

11.2.5.1.92.1 Offset

Register	Offset
GFIFOPRIDBC	C620h

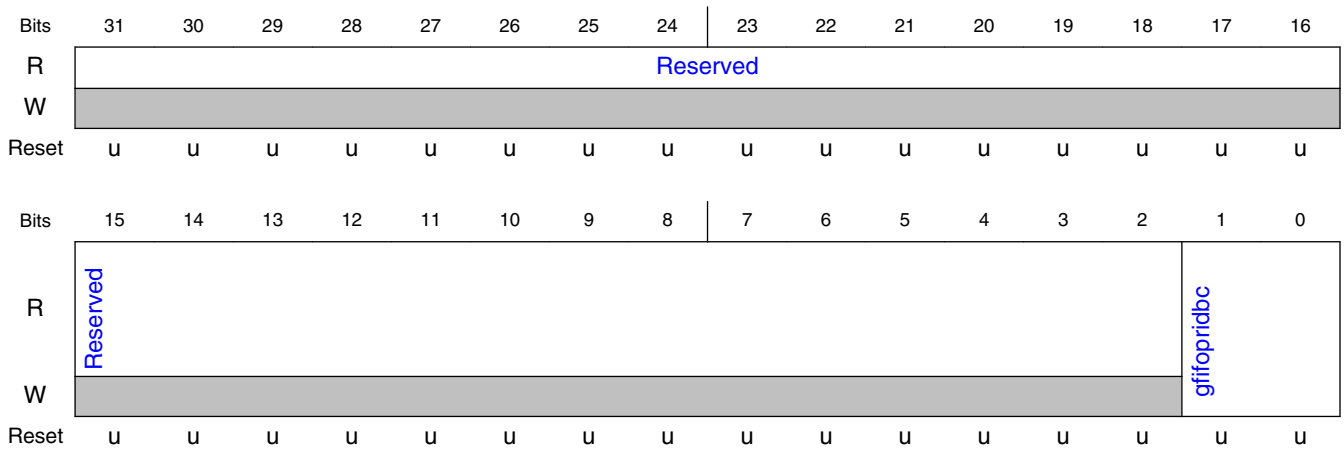
11.2.5.1.92.2 Function

This register specifies the relative priority of the RXFIFOs and TXFIFOs associated with the DbC mode.

Reset Mask: 0xFFFFFFFC

Reset Default Value: 0x0

11.2.5.1.92.3 Diagram



11.2.5.1.92.4 Fields

Field	Function
31-2 —	Reserved
1-0 gfifopridbc	Host DbC DMA priority

11.2.5.1.93 Global Host FIFO DMA High-Low Priority Ratio Register (GDMAHLRATIO)

11.2.5.1.93.1 Offset

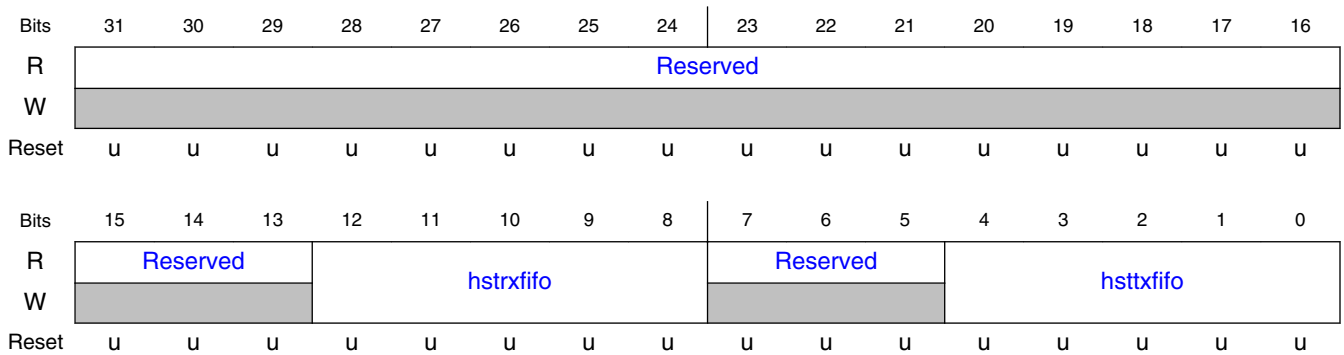
Register	Offset
GDMAHLRATIO	C624h

11.2.5.1.93.2 Function

This register specifies the relative priority of the SS FIFOs with respect to the HS/FSLS FIFOs.

Reset Mask:0xFFFFE0E0

11.2.5.1.93.3 Diagram



11.2.5.1.93.4 Fields

Field	Function
31-13 —	Reserved
12-8 hstrxfifo	Host RXFIFO DMA High-Low Priority
7-5 —	Reserved
4-0 hsttxfifo	Host TXFIFO DMA High-Low Priority

11.2.5.1.94 Global Frame Length Adjustment Register (GFLADJ)

11.2.5.1.94.1 Offset

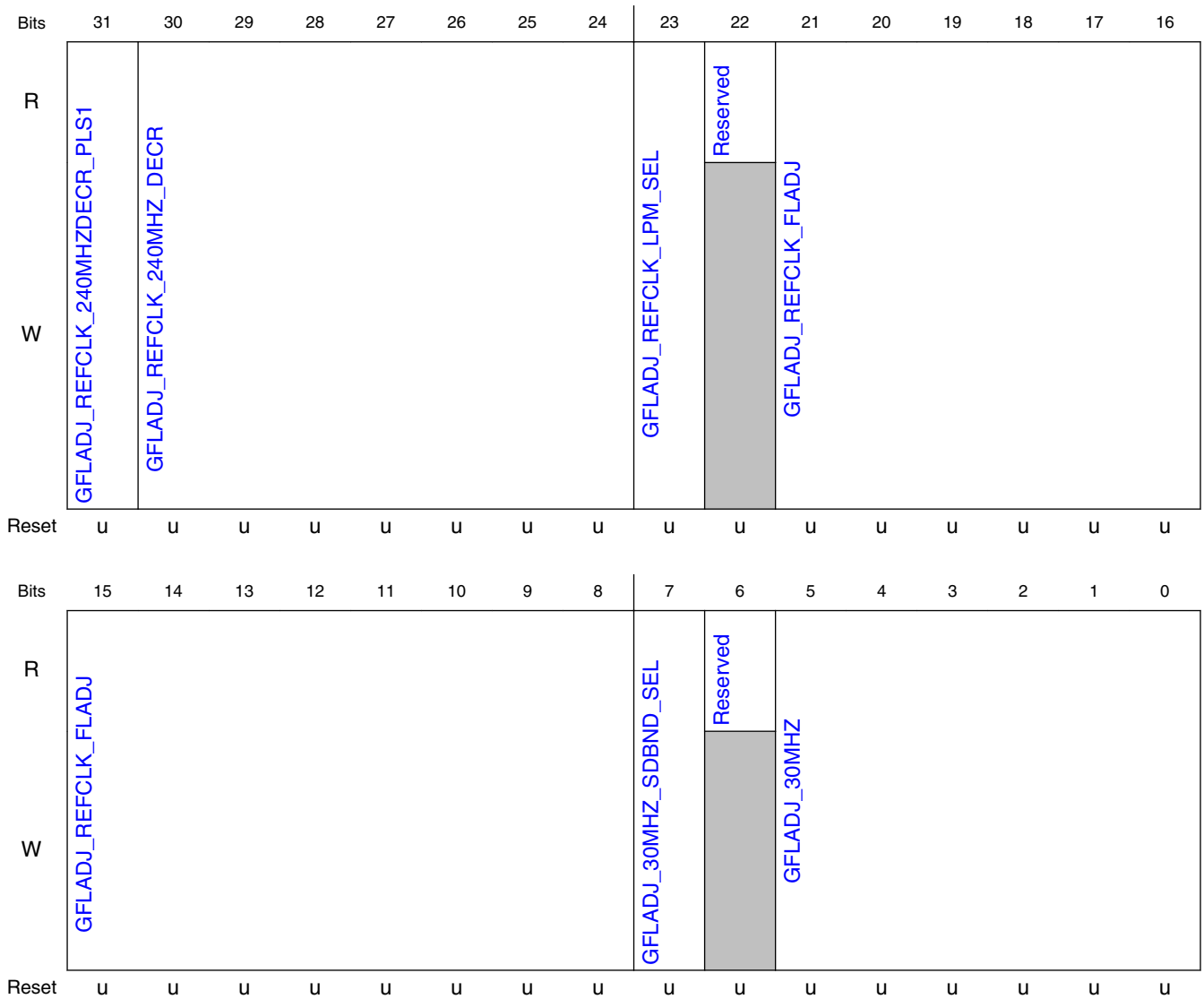
Register	Offset
GFLADJ	C630h

11.2.5.1.94.2 Function

This register provides options for the software to control the core behavior with respect to SOF (Start of Frame) and ITP (Isochronous Timestamp Packet) timers and frame timer functionality.

Reset Mask:0x400040

11.2.5.1.94.3 Diagram



11.2.5.1.94.4 Fields

Field	Function
31 GFLADJ_REFCLK_240MHZDECR_PLS1	This field indicates that the decrement value that the controller applies for each ref_clk must be GFLADJ_REFCLK_240MHZ_DECR and GFLADJ_REFCLK_240MHZ_DECR +1 alternatively on each ref_clk. Set this bit to a '1' only if GFLADJ_REFCLK_LPM_SEL is set to '1' and the fractional component of 240/ref_frequency is greater than or equal to 0.5. Examples: If the ref_clk is 19.2 MHz then - GUCTL.REF_CLK_PERIOD = 52 - GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = (240/19.2) = 12.5 - GFLADJ.GFLADJ_REFCLK_240MHZDECR_PLS1 = 1 If the ref_clk is 24 MHz then - GUCTL.REF_CLK_PERIOD = 41 - GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = (240/24) = 10 - GFLADJ.GFLADJ_REFCLK_240MHZDECR_PLS1 = 0

Table continues on the next page...

Field	Function
30-24 GFLADJ_REFCLK_240MHZ_DECR	This field indicates the decrement value that the controller applies for each ref_clk in order to derive a frame timer in terms of a 240-MHz clock. This field must be programmed to a non-zero value only if GFLADJ_REFCLK_LPM_SEL is set to '1'. The value is derived as follows: GFLADJ_REFCLK_240MHZ_DECR = 240/ref_clk_frequency Examples: If the ref_clk is 24 MHz then - GUCTL.REF_CLK_PERIOD = 41 - GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = 240/24 = 10 If the ref_clk is 48 MHz then - GUCTL.REF_CLK_PERIOD = 20 - GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = 240/48 = 5 If the ref_clk is 17 MHz then - GUCTL.REF_CLK_PERIOD = 58 - GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = 240/17 = 14
23 GFLADJ_REFCLK_LPM_SEL	This bit enables the functionality of running SOF/ITP counters on the ref_clk. This bit must not be set to '1' if GCTL.SOFITPSYNC bit is set to '1'. Similarly, if GFLADJ_REFCLK_LPM_SEL set to '1', GCTL.SOFITPSYNC must not be set to '1'. When GFLADJ_REFCLK_LPM_SEL is set to '1' the overloading of the suspend control of the USB 2.0 first port PHY (UTMI/ULPI) with USB 3.0 port states is removed. For example, for Synopsys PHY, the COMMONONN signal can be tied to '1'. Note that the ref_clk frequencies supported in this mode are 16/17/19.2/20/24/39.7/40 MHz. The utmi_clk[0] signal of the core must be connected to the FREECLK of the PHY. Note: If you set this bit to '1', the GUSB2PHYCFG.U2_FREECLK_EXISTS bit must be set to '0'.
22 —	Reserved for future use
21-8 GFLADJ_REFCLK_FLADJ	This field indicates the frame length adjustment to be applied when SOF/ITP counter is running on the ref_clk. This register value is used to adjust the ITP interval when GCTL[SOFITPSYNC] is set to '1'; SOF and ITP interval when GLADJ.GFLADJ_REFCLK_LPM_SEL is set to '1'. This field must be programmed to a non-zero value only if GFLADJ_REFCLK_LPM_SEL is set to '1' or GCTL.SOFITPSYNC is set to '1'. The value is derived as follows: FLADJ_REF_CLK_FLADJ=((125000/ref_clk_period_integer)-(125000/ref_clk_period)) * ref_clk_period where - the ref_clk_period_integer is the integer value of the ref_clk period got by truncating the decimal (fractional) value that is programmed in the GUCTL.REF_CLK_PERIOD field. - the ref_clk_period is the ref_clk period including the fractional value. Examples: If the ref_clk is 24 MHz then - GUCTL.REF_CLK_PERIOD = 41 - GFLADJ.GLADJ_REFCLK_FLADJ = ((125000/41)-(125000/41.6666))*41.6666 = 2032 (ignoring the fractional value) If the ref_clk is 48 MHz then - GUCTL.REF_CLK_PERIOD = 20 - GFLADJ.GLADJ_REFCLK_FLADJ = ((125000/20)-(125000/20.8333))*20.8333 = 5208 (ignoring the fractional value)
7 GFLADJ_30MHZ_SDBND_SEL	This field selects whether to use the input signal fladj_30mhz_reg or the GFLADJ.GFLADJ_30MHZ to adjust the frame length for the SOF/ITP. When this bit is set to, - 1, the controller uses the register field GFLADJ.GFLADJ_30MHZ value - 0, the controller uses the input signal fladj_30mhz_reg value
6 —	Reserved for future use
5-0 GFLADJ_30MHZ	This field indicates the value that is used for frame length adjustment instead of considering from the sideband input signal fladj_30mhz_reg. This enables post-silicon frame length adjustment in case the input signal fladj_30mhz_reg is connected to a wrong value or is not valid. For details on how to set this value, refer to section 5.2.4, "Frame Length Adjustment Register (FLADJ)," of the xHCI Specification.

11.2.5.1.95 Global USB 2.0 Root Hub Control Register (GUSB2RHBCTL)

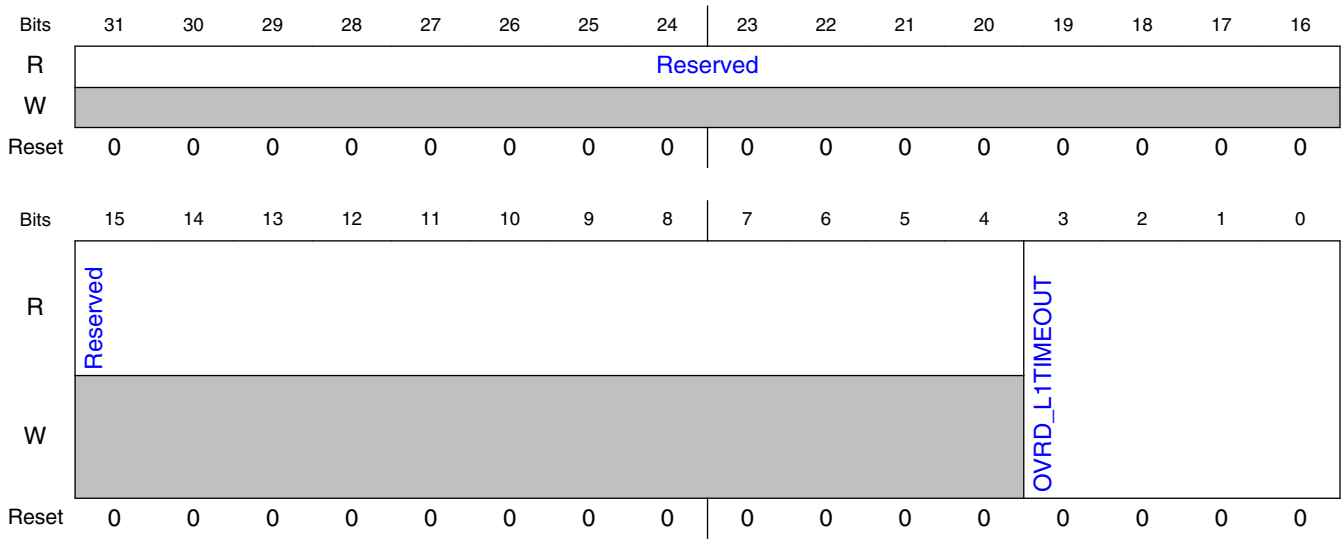
11.2.5.1.95.1 Offset

Register	Offset
GUSB2RHBCTL	C640h

11.2.5.1.95.2 Function

The application must program this register before starting any transactions on the USB if a non-default value is desired. In Device-only configurations, only one register is needed. In Host mode, per-port registers are implemented.

11.2.5.1.95.3 Diagram



11.2.5.1.95.4 Fields

Field	Function
31-4 —	Reserved
3-0 OVRD_L1TIMEOUT	Overriding Driver Programmed L1Timeout Value If this value is 0, the L1 Timeout value is taken from the xHCI PORTHLPMC register. If this value is non-0, then this will override the L1 Timeout value programmed in the xHCI PORTHLPMC register.

11.2.5.1.96 Device Configuration Register (DCFG)

11.2.5.1.96.1 Offset

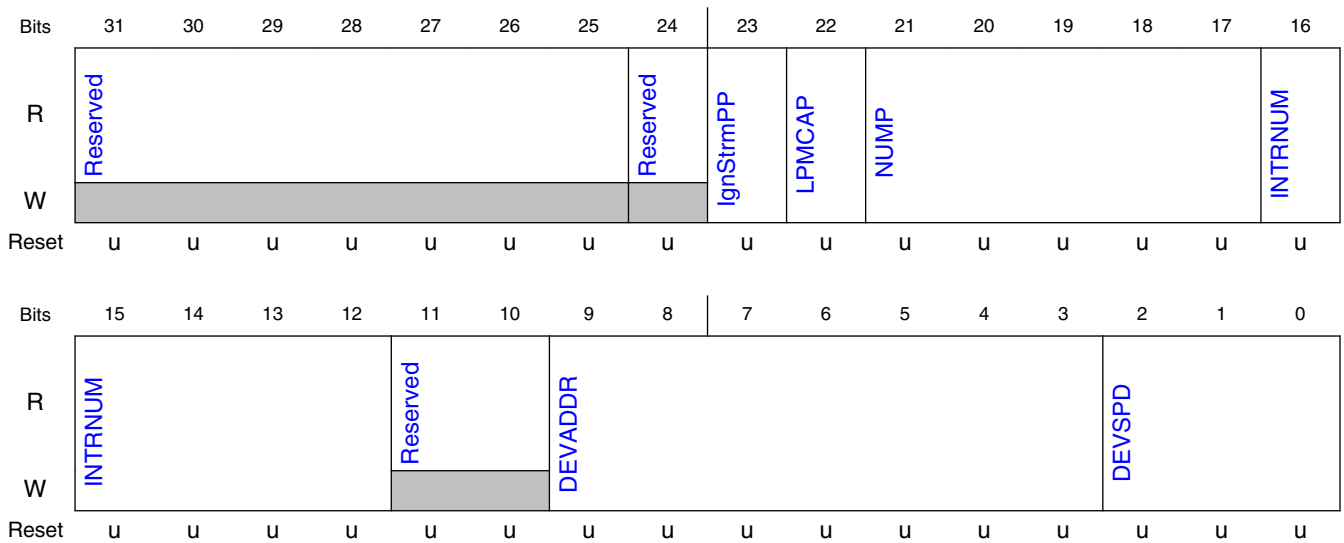
Register	Offset
DCFG	C700h

11.2.5.1.96.2 Function

This register configures the core in device mode after power-on or after certain control commands or enumeration. Do not make changes to this register after initial programming.

Reset Mask:0x7F000C00

11.2.5.1.96.3 Diagram



11.2.5.1.96.4 Fields

Field	Function
31-25 —	Reserved
24 —	Reserved
23 IgnStrmPP	Ignore Stream PP This bit only affects stream-capable bulk endpoints. When this bit is set to '0' and the controller receives a Data Packet with the Packet Pending (PP) bit set to 0 for OUT endpoints, or it receives an ACK with the NumP field set to 0 and PP set to 0 for IN endpoints, the core attempts to search for another stream (CStream) to initiate to the host. However, there are two situations where this behavior is not optimal: - When the host is setting PP=0 even though it has not finished the stream, or - When the endpoint on the device is configured with one transfer resource and therefore does not have any other streams to initiate to the host. When this bit is set to '1', the core ignores the Packet Pending bit for the purposes of stream selection and does not search for another stream when it receives DP(PP=0) or ACK(NumP=0, PP=0). This can enhance the performance when the device system bus bandwidth is low or the host responds to the core's ERDY transmission very quickly.
22	LPM Capable

Table continues on the next page...

Field	Function
LPMCAP	The application uses this bit to control the DWC_usb3 core LPM capabilities. If the core operates as a non-LPM-capable device, it cannot respond to LPM transactions. - 1'b0: LPM capability is not enabled. - 1'b1: LPM capability is enabled.
21-17 NUMP	Number of Receive Buffers This bit indicates the number of receive buffers to be reported in the ACK TP. The DWC_usb3 controller uses this field for non-control endpoints if GRXTHRCFG.UsbRxPktCntSel is set to '0'. The application can program this value based on RxFIFO size, buffer sizes programmed in descriptors, and system latency. For an OUT endpoint, this field controls the number of receive buffers reported in the NumP field of the ACK TP transmitted by the core. Note: This bit is used in host mode when Debug Capability is enabled.
16-12 INTRNUM	Interrupt Number Indicates interrupt/EventQ number on which non-endpoint-specific device-related interrupts (see DEVT) are generated.
11-10 —	Reserved
9-3 DEVADDR	Device Address The application must perform the following: - Program this field after every SetAddress request. - Reset this field to zero after USB reset.
2-0 DEVSPD	Device Speed Indicates the speed at which the application requires the core to connect, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. - 3'b100: SuperSpeed (USB 3.0 PHY clock is 125 MHz or 250 MHz) - 3'b000: High-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) - 3'b001: Full-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 000b - 001b - 100b -

11.2.5.1.97 Device Control Register (DCTL)

11.2.5.1.97.1 Offset

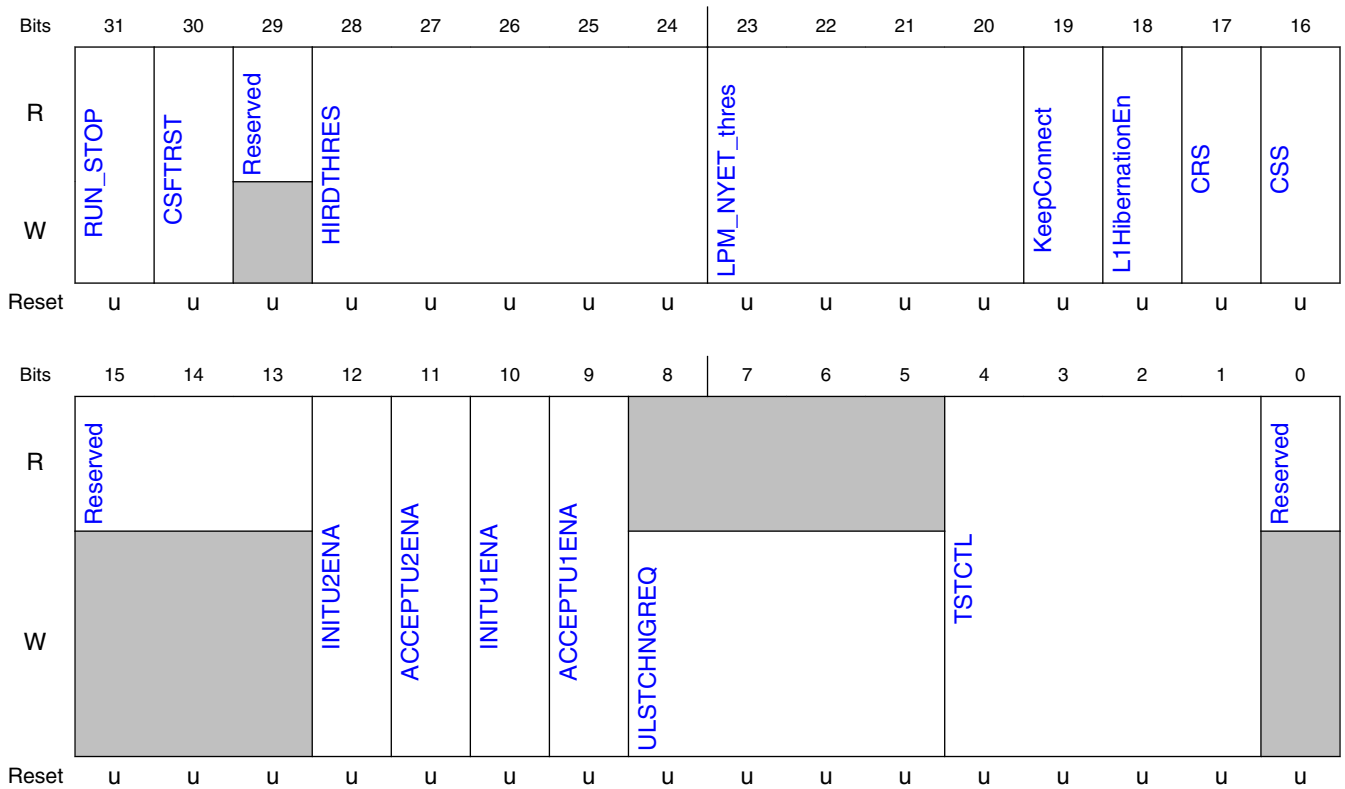
Register	Offset
DCTL	C704h

11.2.5.1.97.2 Function

When hibernation is not enabled using GCTL[GBLHIBERNATIONEN] bit

- Any value can be written to CSS, CRS, L1HIBERNATIONEN, and KEEPCONNECT bits
- L1HIBERNATIONEN and KEEPCONNET bits always return 0 when read in this hibernation-disabled state

11.2.5.1.97.3 Diagram



11.2.5.1.97.4 Fields

Field	Function
31 RUN_STOP	Run/Stop The software writes 1 to this bit to start the device controller operation. To stop the device controller operation, the software must remove any active transfers and write 0 to this bit. When the controller is stopped, it sets the DSTS.DevCtrlHlt bit when the core is idle and the lower layer finishes the disconnect process. The Run/Stop bit must be used in following cases as specified: - After power-on reset and CSR initialization, the software must write 1 to this bit to start the device controller. The controller does not signal connect to the host until this bit is set. - The software uses this bit to control the device controller to perform a soft disconnect. When the software writes 0 to this bit, the host does not see that the device is connected. The device controller stays in the disconnected state until the software writes 1 to this bit. The minimum duration of keeping this bit cleared is specified in the Note below. If the software attempts a connect after the soft disconnect or detects a disconnect event, it must set DCTL[8:5] to 5 before reasserting the Run/Stop bit. - When the USB or Link is in a lower power state and the Two Power Rails configuration is selected, software writes 0 to this bit to indicate that it is going to turn off the Core Power Rail. After the software turns on the Core Power Rail again and re-initializes the device controller, it must set this bit to start the device controller. For more details, see "Low Power Operation" section of the Databook. Note: The following is the minimum duration under various conditions for which the soft disconnect (SftDiscon) bit must be set for the USB host to detect a device disconnect: 30ms: - For SuperSpeed, when the device state is Suspended, Idle, Transmit, or Receive. 10ms: - For high-speed, when the device state is Suspended, Idle, or not Idle/Suspended (performing transactions) - For full-speed/low-speed, when the device state is Suspended, Idle, or not Idle/Supended (performing

Table continues on the next page...

Field	Function
	transactions) To accommodate clock jitter, it is recommended that the application add extra delay to the specified minimum duration.
30 CSFTRST	Core Soft Reset Resets the all clock domains as follows: - This bit clears the interrupts and all the CSRs except GSTS, GSNPSID, GGPIO, GUID, GUSB2PHYCFGn registers, GUSB3PIPECTLn registers, DCFG, DCTL, DEVTEN, and DSTS registers. - All module state machines (except the SoC Bus Slave Unit) are reset to the IDLE state, and all the TxFIFOs and the RxFIFO are flushed. - Any transactions on the SoC bus Master are terminated as soon as possible, after gracefully completing the last data phase of a SoC bus transfer. Any transactions on the USB are terminated immediately. The application can write this bit at any time to reset the core. This is a self-clearing bit; the core clears this bit after all necessary logic is reset in the core, which may take several clocks depending on the core's current state. Once this bit is cleared, the software must wait at least 3 PHY clocks before accessing the PHY domain (synchronization delay). Typically, software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain must be reset for proper operation. Note: Programming this field with random data causes side effect . Bit Bash register testing is not recommended.
29 —	Reserved
28-24 HIRDTHRES	HIRD Threshold The core asserts output signals utmi_l1_suspend_n and utmi_sleep_n (see "LPM Interface Signals" table in the Databook) on the basis of this signal: The core asserts utmi_l1_suspend_n to put the PHY into Deep Low-Power mode in L1 when both of the following are true: - HIRD value is greater than or equal to the value in DCTL.HIRD_Thres[3:0] - HIRD_Thres[4] is set to 1'b1. The core asserts utmi_sleep_n on L1 when one of the following is true: - If the HIRD value is less than HIRD_Thres[3:0] or - HIRD_Thres[4] is set to 1'b0. Note: This field must be set to '0' during SuperSpeed mode of operation.
23-20 LPM_NYET_thres	LPM NYET Threshold When LPM Errata is enabled: Bits [23:20]: LPM NYET Response Threshold (LPM_NYET_thres) Handshake response to LPM token specified by device application. Response depends on DCFG.LPMCap. - DCFG.LPMCap is 1'b0 - The core always responds with Timeout (that is, no response). - DCFG.LPMCap is 1'b1 - The core responds with an ACK on successful LPM transaction, which requires that all of the following are satisfied: - There are no PID or CRC5 errors in both the EXT token and the LPM token (if not true, inactivity results in a timeout ERROR). - No data is pending in the TxFIFO and RxFIFO is empty (else NYET). - The BESL value in the LPM token is less than or equal to LPM_NYET_thres[3:0]
19 KeepConnect	Keep Connect When '1', this bit enables the save and restore programming model by preventing the core from disconnecting from the host when DCTL.RunStop is set to '0'. It also enables the Hibernation Request Event to be generated when the link goes to U3 or L2. The device core disconnects from the host when DCTL.RunStop is set to '0'. This bit indicates whether to preserve this behavior ('0'), or if the core must not disconnect when RunStop is set to 0 ('1'). This bit also prevents the LTSSM from automatically going to U0/L0 when the host requests resume from U3/L2. Note: If Hibernation is disabled, that is, GCTL[1].GblHibernationEn = 0, this bit is tied to zero.
18 L1HibernationEn	L1 Hibernation Enable When this bit is set along with KeepConnect, the device core generates a Hibernation Request Event if L1 is enabled and the HIRD value in the LPM token is larger than the threshold programmed in DCTL.HIRD_Thres. The core does not exit the LPM L1 state until software writes Recovery into the DCTL.ULStChngReq field. This prevents corner cases where the device is entering hibernation at the same time the host is attempting to exit L1. Note: If Hibernation is disabled, that is, GCTL[1].GblHibernationEn = 0, this bit is tied to zero.
17	Controller Restore State

Table continues on the next page...

Field	Function
CRS	This command is similar to the USBCMD.CRS bit in host mode and initiates the restore process. When software sets this bit to '1', the controller immediately sets DSTS.RSS to '1'. When the controller has finished the restore process, it sets DSTS.RSS to '0'. Note: When read, this field always returns '0'.
16 CSS	Controller Save State This command is similar to the USBCMD.CSS bit in host mode and initiates the save process. When software sets this bit to '1', the controller immediately sets DSTS.SSS to '1'. When the controller has finished the save process, it sets DSTS.SSS to '0'. Note: When read, this field always returns '0'.
15-13 —	Reserved
12 INITU2ENA	Initiate U2 Enable 1'b0: May not initiate U2 (default) - 1'b1: May initiate U2 On USB reset, hardware clears this bit to 0. Software sets this bit after receiving SetFeature(U2_ENABLE), and clears this bit when ClearFeature(U2_ENABLE) is received. If DCTL[11] (AcceptU2Ena) is 0, the link immediately exits U2 state.
11 ACCEPTU2ENA	Accept U2 Enable 1'b0: Reject U2 except when Force_LinkPM_Accept bit is set (default) - 1'b1: Core accepts transition to U2 state if nothing is pending on the application side. On USB reset, hardware clears this bit to 0. Software sets this bit after receiving a SetConfiguration command.
10 INITU1ENA	Initiate U1 Enable 1'b0: May not initiate U1 (default); - 1'b1: May initiate U1. On USB reset, hardware clears this bit to 0. Software sets this bit after receiving SetFeature(U1_ENABLE), and clears this bit when ClearFeature(U1_ENABLE) is received. If DCTL[9] (AcceptU1Ena) is 0, the link immediately exits U1 state.
9 ACCEPTU1ENA	Accept U1 Enable 1'b0: Core rejects U1 except when Force_LinkPM_Accept bit is set (default) - 1'b1: Core accepts transition to U1 state if nothing is pending on the application side. On USB reset, hardware clears this bit to 0. Software sets this bit after receiving a SetConfiguration command.
8-5 ULSTCHNGREQ	Software writes this field to issue a USB/Link state change request. A change in this field indicates a new request to the core. If software wants to issue the same request back-to-back, it must write a 0 to this field between the two requests. The result of the state change request is reflected in the USB/Link State in DSTS. These bits are self-cleared on the MAC Layer exiting suspended state. If software is updating other fields of the DCTL register and not intending to force any link state change, then it must write a 0 to this field. SS Compliance mode is normally entered and controlled by the remote link partner. Refer to the USB 3.0 specification. Alternatively, you can force the local link directly into compliance mode, by resetting the SS link with the RUN/STOP bit set to zero. If you then write '10' to the USB/Link State Change field and '1' to RUN/STOP, the link goes to compliance mode. Once you are in compliance, you may alternately write zero and '10' to this field to advance the compliance pattern. In SS mode: - Value Requested Link State Transition/Action - 0 No Action - 4 SS.Disabled - 5 Rx.Detect - 6 SS.Inactive - 8 Recovery - 10 Compliance - Others: Reserved In HS/FS/LS mode: - ValueRequested USB state transition - 8 Remote wakeup request - Others: Reserved The Remote wakeup request must be issued 2us after the device goes into suspend state (DSTS[21:18] is 3 - refer to Table "Fields for Register: DSTS" in the Databook). Note: After coming out of hibernation, software must write 8 (Recovery) into this field to confirm exit from the suspended state.
4-1 TSTCTL	Test Control 4'b000: Test mode disabled - 4'b001: Test_J mode - 4'b010: Test_K mode - 4'b011: Test_SE0_NAK mode - 4'b100: Test_Packet mode - 4'b101: Test_Force_Enable - Others: Reserved
0 —	Reserved

11.2.5.1.98 Device Event Enable Register (DEVTEN)

11.2.5.1.98.1 Offset

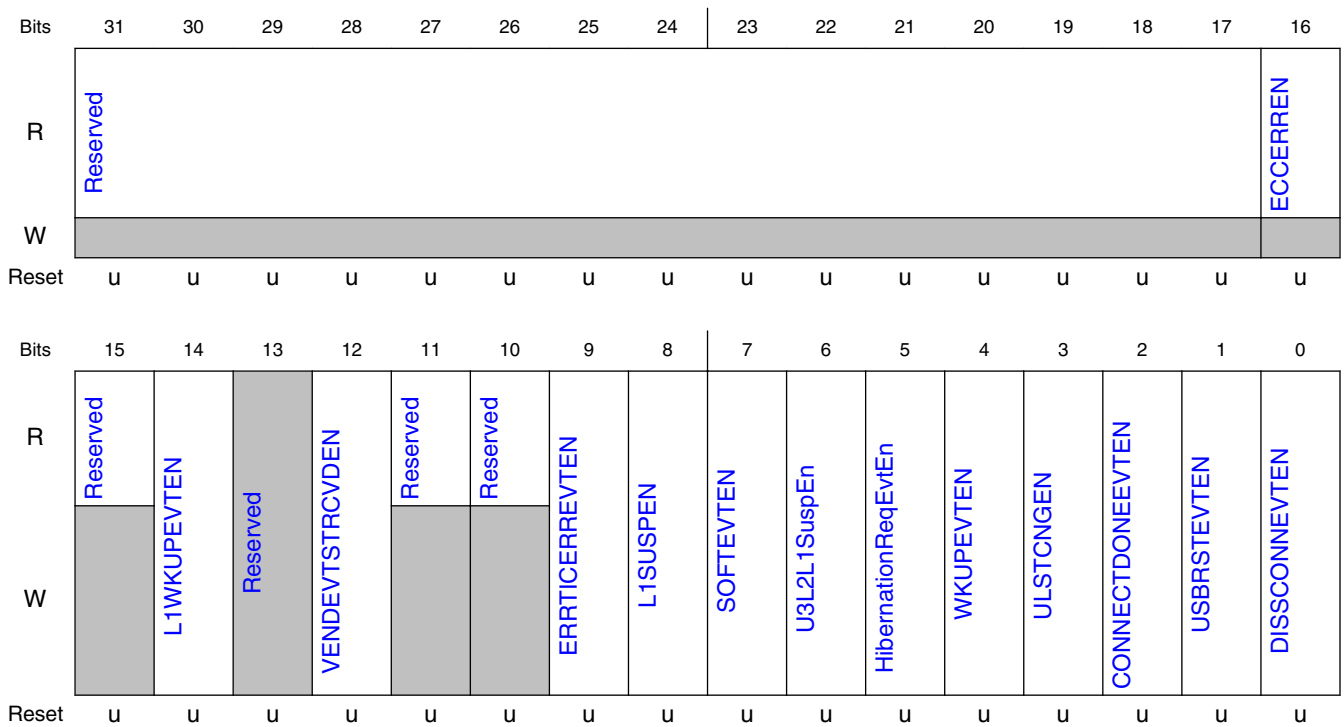
Register	Offset
DEVTEN	C708h

11.2.5.1.98.2 Function

This register controls the generation of device-specific events (see "Event Buffer Content for Device-Specific Events (DEVT)").

Reset Mask:0xFFFE0C00

11.2.5.1.98.3 Diagram



11.2.5.1.98.4 Fields

Field	Function
31-17	Reserved

Table continues on the next page...

Field	Function
—	
16 ECCERREN	ECC Error Enable If this bit is set to 1, the controller reports an ECC error to the software when an uncorrectable ECC occurs internally.
15 —	Reserved
14 L1WKUPEVTEN	L1 Resume Detected Event Enable Note: If GUCTL1[DEV_DECOUPLE_L1L2_EVT] is enabled, then this bit is for L1 Resume Detected Event Enable.
13 —	Reserved
12 VENDEVTSTRCVDEN	Vendor Device Test LMP Received Event
11 —	Reserved
10 —	Reserved
9 ERRTICERREVTEN	Erratic Error Event Enable
8 L1SUSPEN	L1 Suspend Event Enable Note: Only if GUCTL1[DEV_DECOUPLE_L1L2_EVT] is enabled, this bit is for L1 Suspend Event Enable.
7 SOFTEVTEN	Start of (u)frame
6 U3L2L1SuspEn	U3/L2 or U3/L2L1 Suspend Event Enable Note: - If GUCTL1[DEV_DECOUPLE_L1L2_EVT] is enabled, then this bit is for U3/L2 Suspend Event Enable. - If GUCTL1[DEV_DECOUPLE_L1L2_EVT] is not enabled, then this bit is for U3/L2L1 Suspend Event Enable.
5 HibernationReqEvtEn	Hibernation Request Event This bit enables/disables the generation of the Hibernation Request Event.
4 WKUPEVTEN	U3/L2 or U3/L2L1 Resume Detected Event Enable Note: - If GUCTL1[DEV_DECOUPLE_L1L2_EVT] is enabled, then this bit is for U3/L2 Resume Detected Event Enable. - If GUCTL1[DEV_DECOUPLE_L1L2_EVT] is not enabled, then this bit is for U3/L2L1 Resume Detected Event Enable.
3 ULSTCNGEN	USB/Link State Change Event Enable
2 CONNECTDONEEVTEN	Connection Done Enable
1 USBRSTEVTEN	USB Reset Enable

Table continues on the next page...

Field	Function
0 DISSCONNEVTEN	Disconnect Detected Event Enable

11.2.5.1.99 Device Status Register (DSTS)

11.2.5.1.99.1 Offset

Register	Offset
DSTS	C70Ch

11.2.5.1.99.2 Function

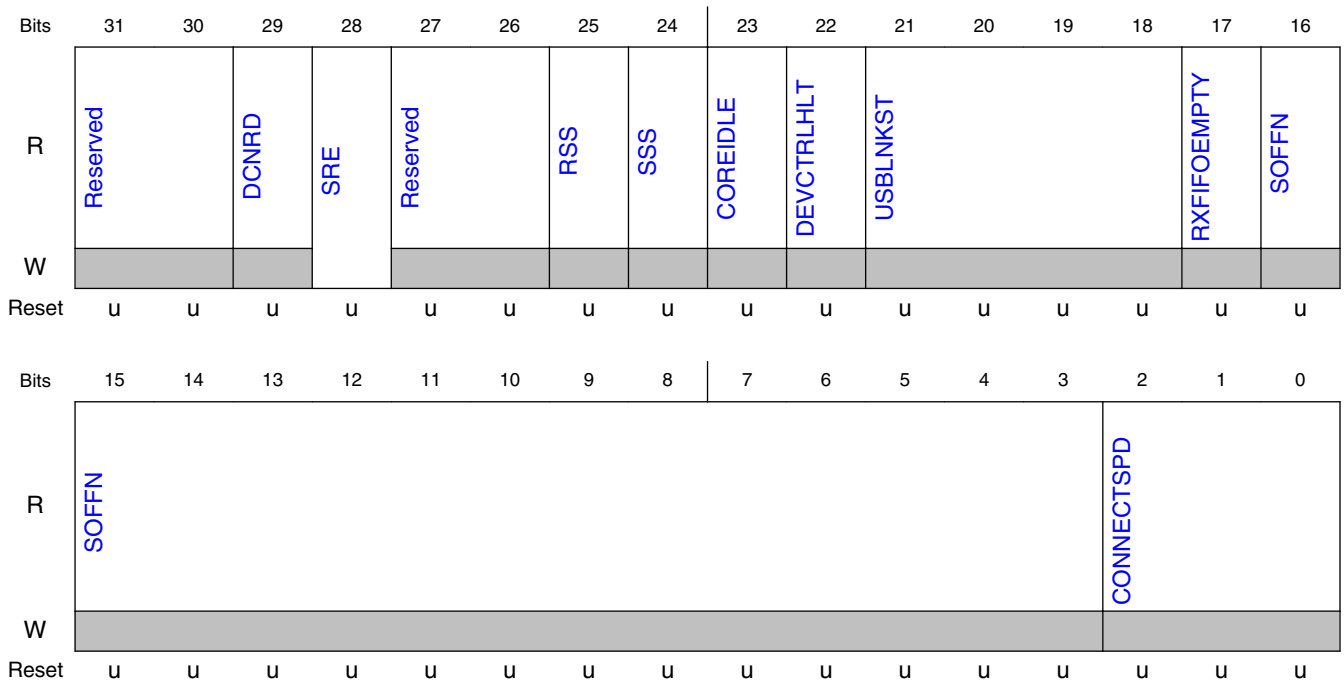
This register indicates the status of the device controller with respect to USB-related events.

Reset Mask:0xDC000000

NOTE

When hibernation is not enabled, RSS and SSS bits always return 0 when read.

11.2.5.1.99.3 Diagram



11.2.5.1.99.4 Fields

Field	Function
31-30 —	Reserved
29 DCNRD	Device Controller Not Ready The bit indicates that the core is in the process of completing the state transitions after exiting from hibernation. To complete the state transitions, it takes 256 bus clock cycles from the time DCTL[31].Run/Stop is set. During hibernation, if the UTMI/ULPI PHY is in suspended state, then the 256-bus clock cycle delay starts after the PHY exited suspended state. Software must set DCTL[31].Run/Stop to '1' and wait for this bit to be de-asserted to zero before processing DSTS.USBLnkSt. This bit is valid only when DWC_USB3_EN_PWROPT is set to 2 and GCTL[1].GblHibernationEn =1.
28 SRE	Save Restore Error Currently not supported.
27-26 —	Reserved
25 RSS	Restore State Status This bit is similar to the USBSTS.RSS in host mode. When the controller finishes the restore process, it completes the command by setting DSTS.RSS to '0'.
24 SSS	Save State Status This bit is similar to the USBSTS.SSS in host mode. When the controller has finished the save process, it completes the command by setting DSTS.SSS to '0'.
23	Core Idle

Table continues on the next page...

Field	Function
COREIDLE	The bit indicates that the core finished transferring all RxFIFO data to system memory, writing out all completed descriptors, and all Event Counts are zero. Note: While testing for Reset values, mask out the read value. This bit represents the changing state of the core and does not hold a static value.
22 DEVCTRLHLT	Device Controller Halted This bit is set to 0 when the Run/Stop bit in the DCTL register is set to 1. The core sets this bit to 1 when, after SW sets Run/Stop to 0, the core is idle and the lower layer finishes the disconnect process. When Halted=1, the core does not generate Device events. Note: - The core does not set this bit to 1 if GEVNTCOUNTn has some valid value. Software needs to acknowledge the events that are generated (by writing to GEVNTCOUNTn) while it is waiting for this bit to be set to 1. - When Interrupt Moderation is enabled, there could be delay in raising the interrupt line when the event count is non-zero. Software should read the GEVNTCOUNT register directly and acknowledge them.
21-18 USBLNKST	USB/Link State In SS mode: LTSSM State - 4'h0: U0 - 4'h1: U1 - 4'h2: U2 - 4'h3: U3 - 4'h4: SS_DIS - 4'h5: RX_DET - 4'h6: SS_INACT - 4'h7: POLL - 4'h8: RECOV - 4'h9: HRESET - 4'ha: CMPLY - 4'hb: LPBK - 4'hf: Resume/Reset In HS/FS/LS mode: - 4'h0: On state - 4'h2: Sleep (L1) state - 4'h3: Suspend (L2) state - 4'h4: Disconnected state (Default state) - 4'h5: Early Suspend state (valid only when Hibernation is disabled, GCTL[1].GblHibernationEn = 0) - 4'he: Reset (valid only when Hibernation is enabled, GCTL[1].GblHibernationEn = 1) - 4'hf: Resume (valid only when Hibernation is enabled, GCTL[1].GblHibernationEn = 1) The link state Resume/Reset indicates that the core received a resume or USB reset request from the host while the link was in hibernation. Software must write '8' (Recovery) to the DCTL.ULStChngReq field to acknowledge the resume/reset request. When Hibernation is enabled, GCTL[1].GblHibernationEn = 1, this field USBLnkSt is valid only when DCTL[31].Run/Stop set to '1' and DSTS[29].DCNRD = 0. Note: If SSIC is enabled, while exiting a low power state, the USBLnkSt field indicates Resume/Reset even for a disconnect condition because a resume precedes the disconnect.
17 RXFIFOEMPTY	RxFIFO Empty.
16-3 SOFFN	Frame/Microframe Number of the Received SOF When the core is operating at SuperSpeed, - [16:3] indicates the uframe/ITP number When the core is operating at high-speed, - [16:6] indicates the frame number - [5:3] indicates the microframe number When the core is operating at full-speed, - [16:14] is not used. Software can ignore these 3 bits - [13:3] indicates the frame number
2-0 CONNECTSPD	Connected Speed Indicates the speed at which the DWC_usb3 core has come up after speed detection through a chirp sequence. - 3'b100: SuperSpeed (PHY clock is running at 125 or 250 MHz) - 3'b000: High-speed (PHY clock is running at 30 or 60 MHz) - 3'b001: Full-speed (PHY clock is running at 30 or 60 MHz) Low-speed is not supported for devices using a UTMI+ PHY. 000b - 001b - 100b -

11.2.5.1.100 Device Generic Command Parameter Register (DGCMDPAR)

11.2.5.1.100.1 Offset

Register	Offset
DGCMDPAR	C710h

11.2.5.1.100.2 Function

This register indicates the device command parameter.

11.2.5.1.100.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.100.4 Fields

Field	Function
31-0 PARAMETER	PARAMETER

11.2.5.1.101 Device Generic Command Register (DGCMD)

11.2.5.1.101.1 Offset

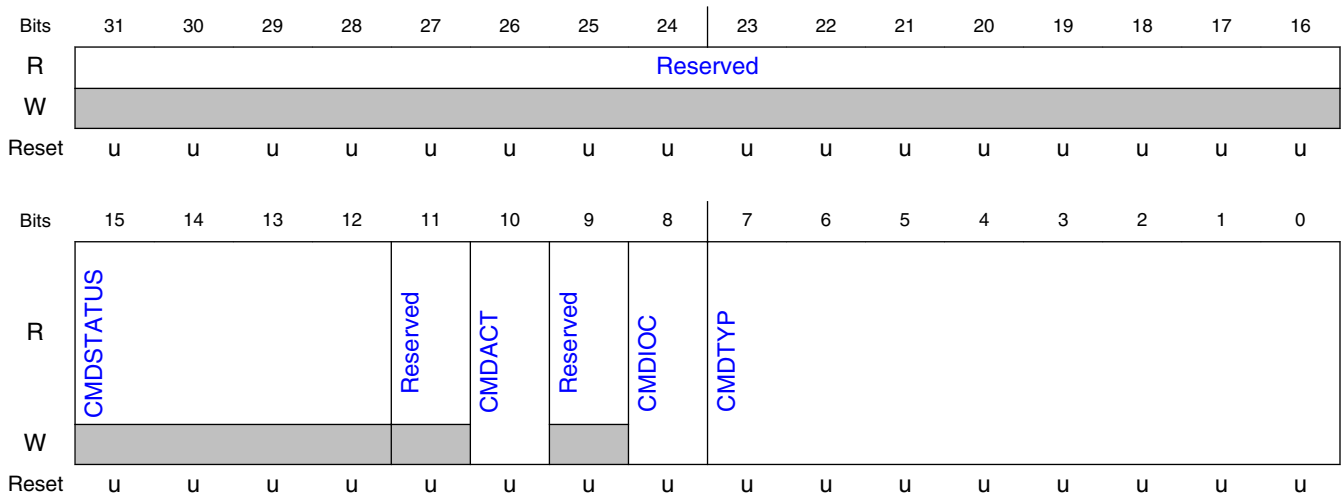
Register	Offset
DGCMD	C714h

11.2.5.1.101.2 Function

This register enables software to program the core using a single generic command interface to send link management packets and notifications.

Reset Mask:0xFFFF0A00

11.2.5.1.101.3 Diagram



11.2.5.1.101.4 Fields

Field	Function
31-16 —	Reserved
15-12 CMDSTATUS	Command Status 1: CmdErr: Indicates that the device controller encountered an error while processing the command. - 0: Indicates command success
11 —	Reserved
10 CMDACT	Command Active The software sets this bit to 1 to enable the device controller to execute the generic command. The device controller sets this bit to 0 after executing the command.
9 —	Reserved
8 CMDIOC	Command Interrupt on Complete When this bit is set, the device controller issues a Generic Command Completion event after executing the command. Note that this interrupt is mapped to DCFG.IntrNum. Note: This field must not set to '1' if the DCTL.RunStop field is '0'.
7-0 CMDTYP	Generic Command Type Specifies the type of generic command the software driver is requesting the controller to perform. - 02h: Set Periodic Parameters - 04h: Set Scratchpad Buffer Array Address Lo - 05h: Set Scratchpad Buffer Array Address Hi - 07h: Transmit Device Notification - 09h: Selected FIFO Flush - 0Ah: All FIFO Flush - 0Ch: Set Endpoint NRDY - 10h: Run SoC Bus LoopBack Test - 11h: Restart After Disconnect All other values are reserved.

11.2.5.1.102 Device Active USB Endpoint Enable Register (DALEPENA)

11.2.5.1.102.1 Offset

Register	Offset
DALEPENA	C720h

11.2.5.1.102.2 Function

This register indicates whether a USB endpoint is active in a given configuration or interface.

11.2.5.1.102.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	USBACTEP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	USBACTEP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.102.4 Fields

Field	Function
31-0 USBACTEP	USB active endpoints This bit indicates if a USB endpoint is active in the current configuration and interface. Bit[0]: USB EP0-OUT Bit[1]: USB EP0-IN Bit[2]: USB EP1-OUT Bit[3]: USB EP1-IN Bit[4]: USB EP2-OUT Bit[5]: USB EP2-IN Bit[6]: USB EP3-OUT Bit[7]: USB EP3-IN The entity programming this register must set bits 0 and 1 because they enable control endpoints that map to physical endpoints (resources) after USB reset.

Field	Function
	Hardware clears these bits for all endpoints (other than EP0-OUT and EP0-IN) after detecting a USB reset event. After receiving SetConfiguration and SetInterface requests, the application must program endpoint registers accordingly and set these bits.

11.2.5.1.103 Device Physical Endpoint-n Command Parameter 2 Register (DEPCMDPAR2)

11.2.5.1.103.1 Offset

Register	Offset
DEPCMDPAR2	C800h

11.2.5.1.103.2 Function

This register indicates the physical endpoint command Parameter 2.

11.2.5.1.103.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.103.4 Fields

Field	Function
31-0 PARAMETER	PARAMETER

11.2.5.1.104 Device Physical Endpoint-n Command Parameter 1 Register (DEPCMDPAR1)

11.2.5.1.104.1 Offset

Register	Offset
DEPCMDPAR1	C804h

11.2.5.1.104.2 Function

This register indicates the physical endpoint command parameter 1. It must be programmed before issuing the command.

11.2.5.1.104.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.104.4 Fields

Field	Function
31-0 PARAMETER	PARAMETER

11.2.5.1.105 Device Physical Endpoint-n Command Parameter 0 Register (DEPCMDPAR0)

11.2.5.1.105.1 Offset

Register	Offset
DEPCMDPAR0	C808h

11.2.5.1.105.2 Function

This register indicates the physical endpoint command parameter 0. This must be programmed before or along with the command. For commands needing only one 32-bit parameter, this register must be programmed with the command register.

11.2.5.1.105.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.105.4 Fields

Field	Function
31-0 PARAMETER	PARAMETER

11.2.5.1.106 Device Physical Endpoint-n Command Register (DEPCMD)

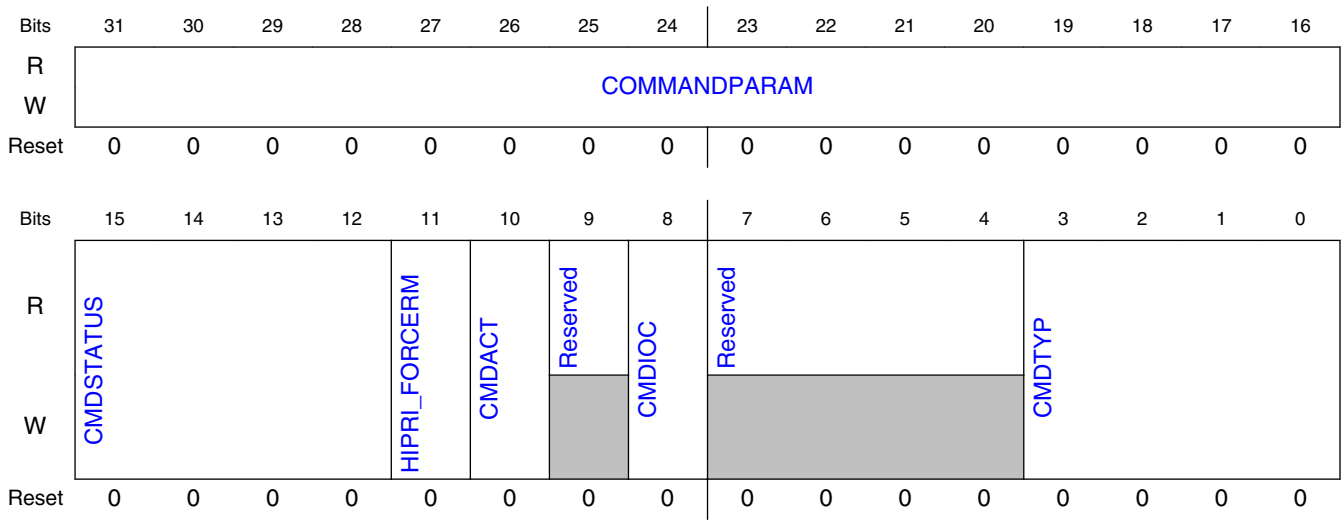
11.2.5.1.106.1 Offset

Register	Offset
DEPCMD	C80Ch

11.2.5.1.106.2 Function

This register enables software to issue physical endpoint-specific commands.

11.2.5.1.106.3 Diagram



11.2.5.1.106.4 Fields

Field	Function
31-16 COMMANDPARAM	Command Parameters or Event Parameters Command Parameters When this register is written for Start Transfer command: - [31:16]: StreamID. The USB StreamID assigned to this transfer
15-12 CMDSTATUS	Command Completion Status Additional information about the completion of this command is available in this field.
11 HIPRI_FORCERM	High Priority/ForceRM HighPriority: Only valid for Start Transfer command ForceRM: Only valid for End Transfer command ClearPendIN:Only valid for Clear Stall command .
10 CMDACT	Command Active Software sets this bit to 1 to enable the device endpoint controller to execute the generic command. The device controller sets this bit to 0 when the CmdStatus field is valid and the endpoint is ready to accept another command. This does not imply that all the effects of the previously-issued command have taken place.
9 —	Reserved
8 CMDIOC	CMDIOC Command Interrupt on Complete When this bit is set, the device controller issues a generic Endpoint Command Complete event after executing the command. Note that this interrupt is mapped to DEPCFG.IntrNum. When the DEPCFG command is executed, the command interrupt on completion goes to the interrupt pointed by the DEPCFG.IntrNum in the current command. **NOTE:** This field must not set to 1 if the DCTL.RunStop field is 0.
7-4 —	Reserved

Table continues on the next page...

Field	Function
3-0 CMDTYP	Command Type Specifies the type of command the software driver is requesting the core to perform. 01h: Set Endpoint Configuration 64 or 96-bit Parameter

11.2.5.1.107 Device Interrupt Moderation Register (DEV_IMOD)

11.2.5.1.107.1 Offset

Register	Offset
DEV_IMOD	CA00h

11.2.5.1.107.2 Function

This register controls the Interrupt Moderation feature that allows the device software to throttle the interrupt rate.

11.2.5.1.107.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DEVICE_IMODC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DEVICE_IMODI															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.107.4 Fields

Field	Function
31-16 DEVICE_IMODC	Interrupt Moderation Down Counter Loaded with the DEVICE_IMODI value, whenever the hardware interrupt(n) line is de-asserted from the asserted state, counts down to 0, and stops.
15-0 DEVICE_IMODI	Moderation Interval This field holds the minimum inter-interrupt interval between events.

11.2.5.1.108 BC Configuration Register (BCFG)

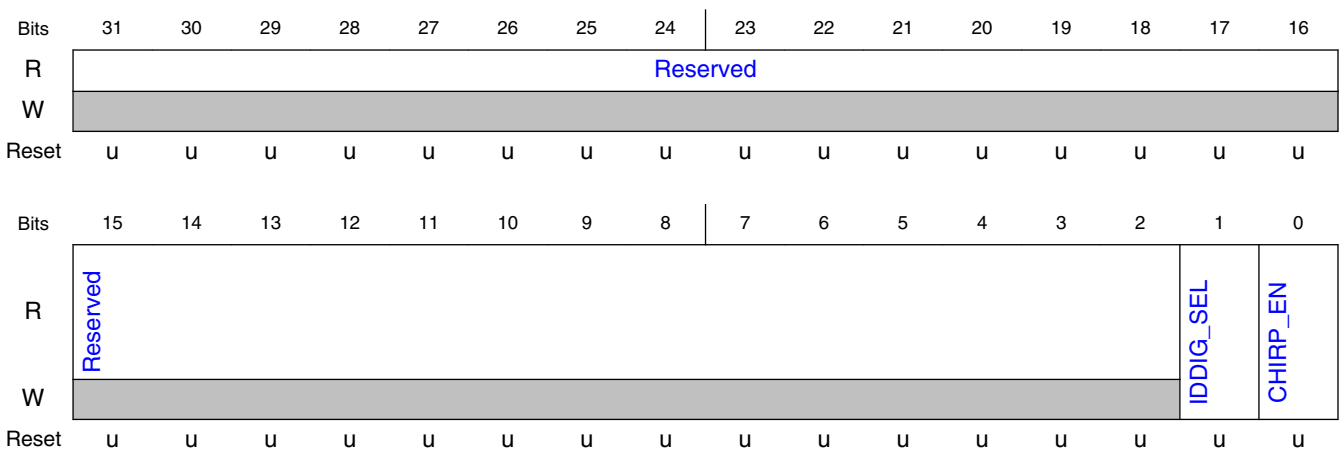
11.2.5.1.108.1 Offset

Register	Offset
BCFG	CC30h

11.2.5.1.108.2 Function

Reset Mask:0xFFFFFFFC

11.2.5.1.108.3 Diagram



11.2.5.1.108.4 Fields

Field	Function
31-2 —	Reserved
1 IDDIG_SEL	IDDIG Select When this bit is programmed as: 1'b1: The core determines the ID pin value from RID_A and RID_GND inputs of the Battery Charger. 1'b0: The core determines ID pin value from utmiotg_iddig input. When ULPI PHY is used, IDDIG_SEL needs to be programmed to 1'b0 always, so that the core uses the ID value obtained from the RX_CMD.
0 CHIRP_EN	Chirp Enable When this bit is 1'b1, the core asserts the bc_chirp_on output signal to indicate an imminent chirp.

11.2.5.1.109 BC Event Register (BCEVT)

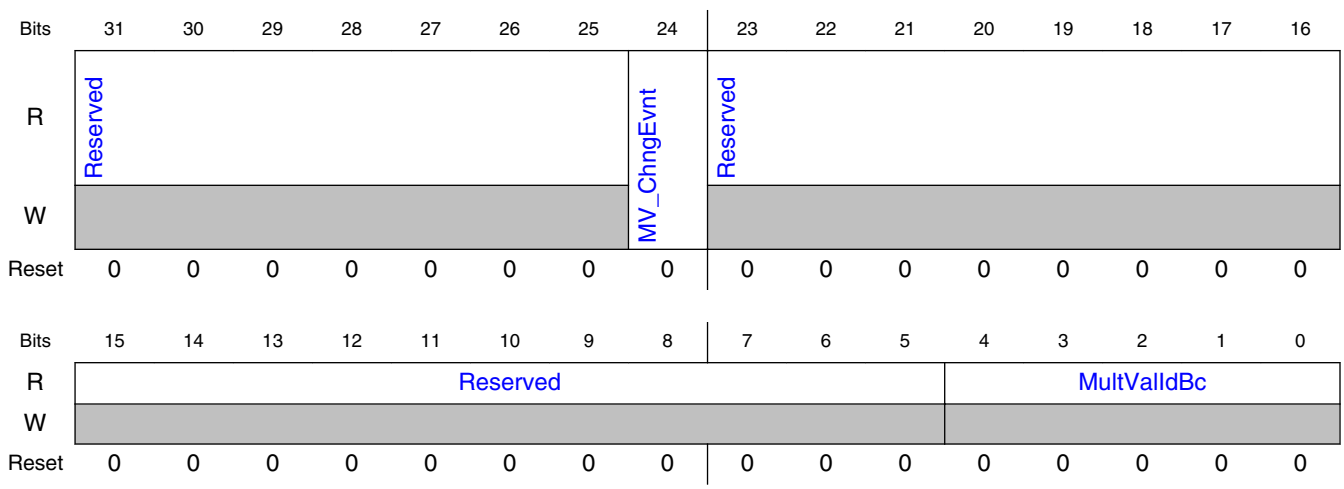
11.2.5.1.109.1 Offset

Register	Offset
BCEVT	CC38h

11.2.5.1.109.2 Function

Writing 1 to the info bit in this register clears the register bit and the associated interrupt.

11.2.5.1.109.3 Diagram



11.2.5.1.109.4 Fields

Field	Function
31-25 —	Reserved
24 MV_ChngEvnt	Multi-Valued Input Changed Event This event indicates that there is a change in the value of at least one ACA pin value. This bit is present only if DWC_USB3_EN_BC = 1, otherwise it is Reserved.
23-5 —	Reserved
4-0 MultValIdBc	Multi Valued ID pin Indicates the Battery Charger ACA inputs. - MultValIdBc[4]: bc_rid_float - MultValIdBc[3]: bc_rid_gnd - MultValIdBc[2]: bc_rid_a - MultValIdBc[1]: bc_rid_b - MultValIdBc[0]: bc_rid_c

11.2.5.1.110 BC Event Enable Register (BCEVTEN)

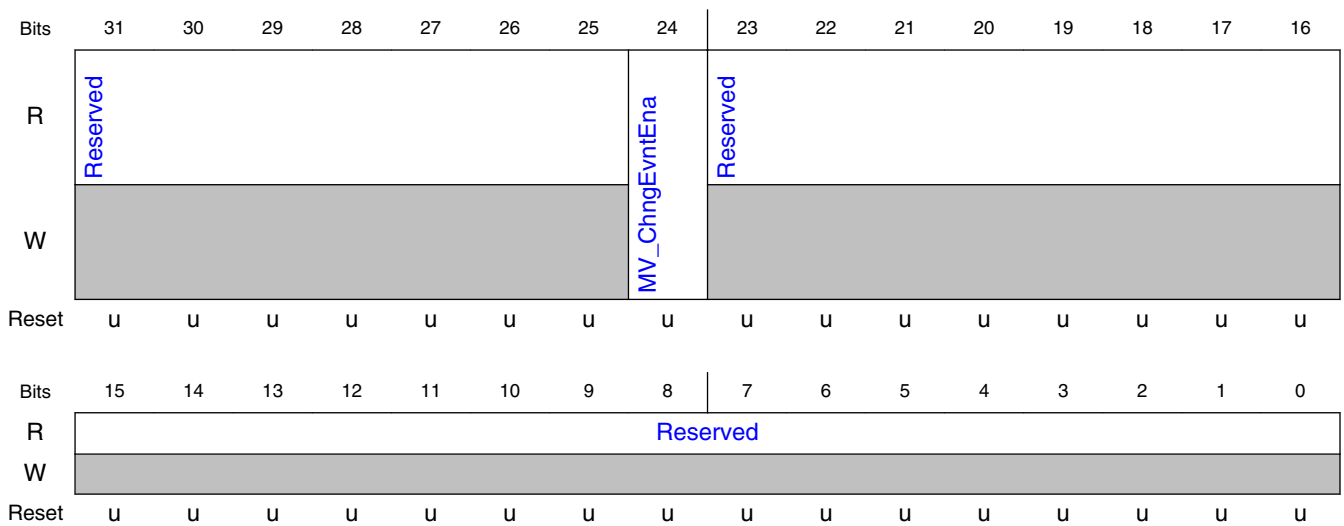
11.2.5.1.110.1 Offset

Register	Offset
BCEVTEN	CC3Ch

11.2.5.1.110.2 Function

Reset Mask:0xFEFFFFFF

11.2.5.1.110.3 Diagram



11.2.5.1.110.4 Fields

Field	Function
31-25 —	Reserved
24 MV_ChngEvntEna	Multi-Valued Input changed Event Enable When this bit is set, MV_ChngEvnt in BCEVT register is enabled.
23-0 —	Reserved

11.2.5.1.111 U1/U2 LFPS Rx Timer Register (LU1LFPSRXTIM0)

11.2.5.1.111.1 Offset

Register	Offset
LU1LFPSRXTIM0	D000h

11.2.5.1.111.2 Function

Link Layer Register for U1/U2 LFPS RX Timers. This register is common for all SS ports.

11.2.5.1.111.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	U1U2_LFPS_EXIT_RX_CLK								U1U2_EXIT_RSP_RX_CLK							
Reset	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

11.2.5.1.111.4 Fields

Field	Function
31-16 —	Reserved
15-8 U1U2_LFPS_EXIT_RX_CLK	Programmable U1U2 LFPS EXIT RX CLKS Applicable to Remote Partner initiated Ux exit: Time to recognize valid Ux exit request from the remote partner. This field is encoded as the pipe clk (8ns) count for the LFPS. • 1: 8ns • 2: 16ns • 3: 24ns, and so on
7-0 U1U2_EXIT_RSP_RX_CLK	Programmable U1U2 EXIT RESP RX CLKS Applicable to locally initiated Ux exit: Minimum LFPS reception from remote to consider Ux exit handshake is successful. This field is encoded as the pipe clk (8ns) count for the LFPS. • 1: 8ns • 2: 16ns • 3: 24ns, and so on

11.2.5.1.112 Link Setting Register (LINK_SETTINGS0)

11.2.5.1.112.1 Offset

Register	Offset
LINK_SETTINGS0	D020h

11.2.5.1.112.2 Function

Link Layer User Control Register for enabling Link/PHY-specific options. This register is common for all SS ports.

11.2.5.1.112.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	U1_RESID_TIMER_US			Reserved	PM_LC_TIMER_US			PM_ENTRY_TIMER_US				Reserved			
W																
Reset	0	1	0	0	0	1	0	1	1	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.112.4 Fields

Field	Function
31 —	Reserved
30-28 U1_RESID_TIMER_US	Programmable U1 MIN RESIDENCY TIMER This field specifies U1 MIN RESIDENCY TIMER value in us. Set to 0 to disable the timer.
27	Reserved

Table continues on the next page...

Field	Function
—	
26-24 PM_LC_TIMER_US	Programmable PM_LC_TIMER This field specifies PM_LC_TIMER value in us.
23-20 PM_ENTRY_TIMER_US	Programmable PM_ENTRY_TIMER This field specifies PM_ENTRY_TIMER value in us.
19-0 —	Reserved

11.2.5.1.113 Link User Control Register (LLUCTL0)

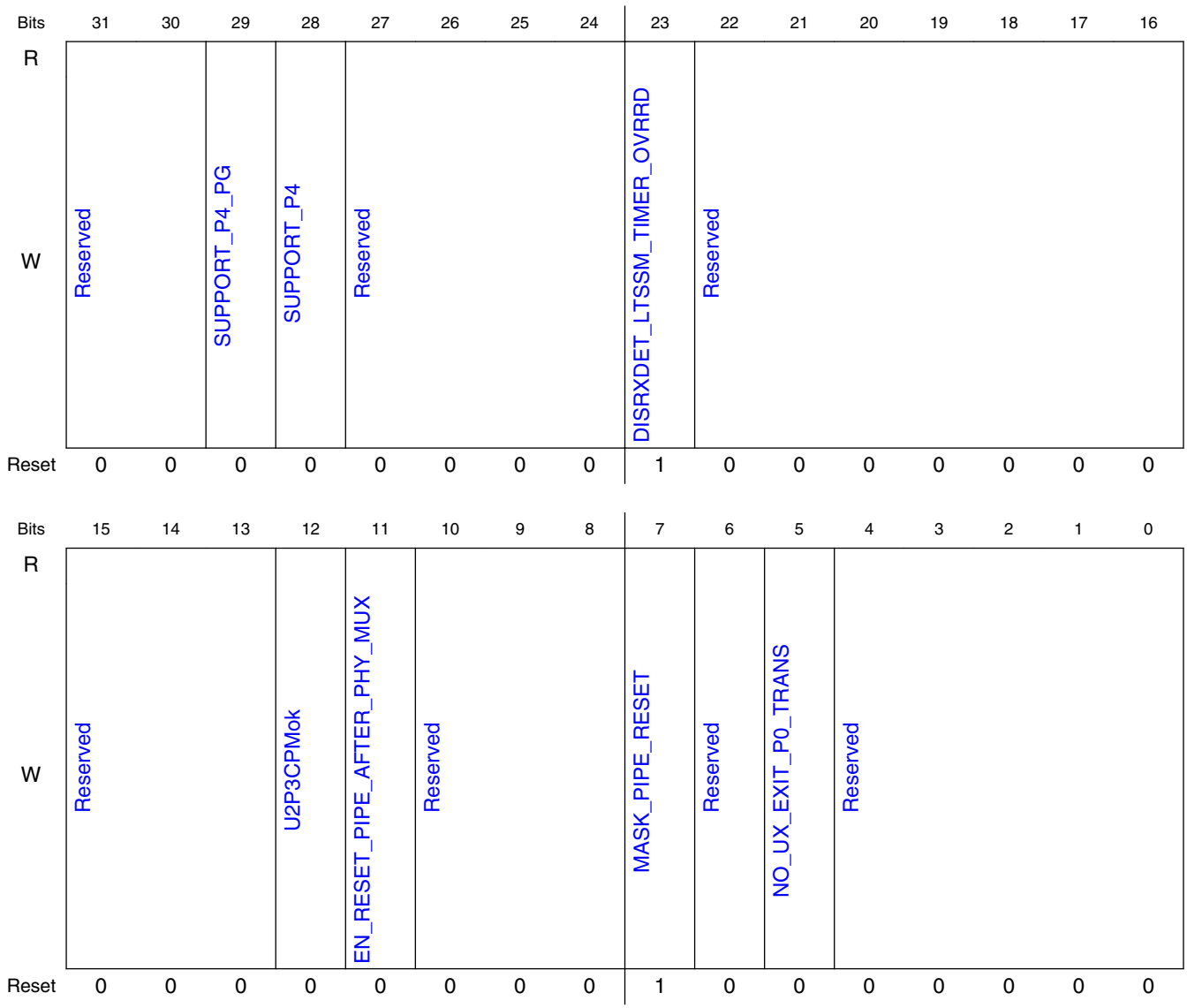
11.2.5.1.113.1 Offset

Register	Offset
LLUCTL0	D024h

11.2.5.1.113.2 Function

Link Layer User Control Register for enabling Link/PHY-specific options. This register is common for all SS ports.

11.2.5.1.113.3 Diagram



11.2.5.1.113.4 Fields

Field	Function
31-30 —	Reserved
29 SUPPORT_P4_PG	PHY P4 Power Gate Mode (PG) Enabled Set this bit if the PHY supports PG mode in P4.
28 SUPPORT_P4	Support PHY P3.CPM and P4 Power States When this bit is set, the controller puts the PHY in P3.CPM or P4 in certain states.
27-24	Reserved

Table continues on the next page...

Field	Function
—	
23 DISRXDET_LTSSM_TIMER_OVRRD	When DisRxDetU3RxDet is asserted in Polling or U1, the timeout expires immediately.
22-13 —	Reserved
12 U2P3CPMok	P3CPM OK for U2/SSInactive • 0: During link state U2/SS.Inactive, put PHY in P2 (Default) • 1: During link state U2/SS.Inactive, put PHY in P3CPM.
11 EN_RESET_PIPE_AFTER_PHY_MUX	The controller issues USB 3.0 PHY reset after DisRxDetU3RxDet is de-asserted.
10-8 —	Reserved
7 MASK_PIPE_RESET	Mask Pipe Reset If this bit is set, controller blocks pipe_reset_n from going to the PHY when DisRxDetU3RxDet=1.
6 —	Reserved
5 NO_UX_EXIT_P0_TRANS	Link LTSSM detects Ux_exit LFPS when P0 transition is on-going by default. If this bit is set, Link LTSSM may miss Ux_exit LFPS when P0 transition is happening.
4-0 —	Reserved

11.2.5.1.114 Link Datapath Delay Register (LPTMDPDELAY0)

11.2.5.1.114.1 Offset

Register	Offset
LPTMDPDELAY0	D028h

11.2.5.1.114.2 Function

Link Layer Timer Register for P3CPM/P4 Residency. This register is common for all SS ports.

11.2.5.1.114.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved										P3CPMP4_RESIDENCY					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	P3CPMP4_RESIDENCY						Reserved									
Reset	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0

11.2.5.1.114.4 Fields

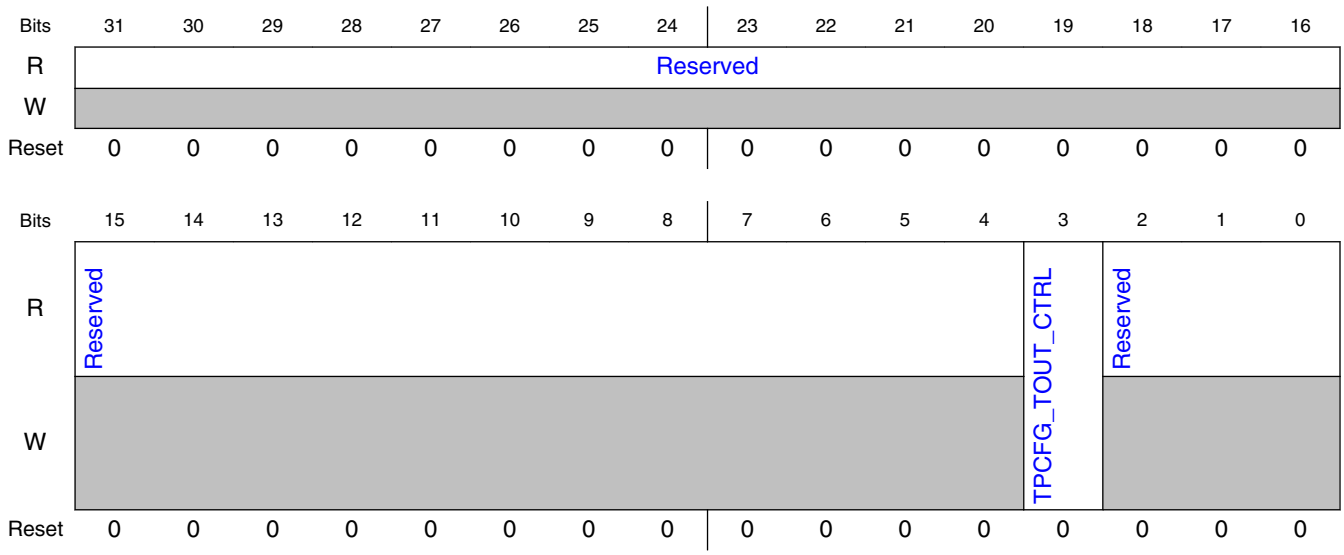
Field	Function
31-22 —	Reserved
21-10 P3CPMP4_RESIDENCY	Programmable PM_ENTRY_TIMER This field specifies PM_ENTRY_TIMER value in us.
9-0 —	Reserved

11.2.5.1.115 U3 Root Hub Debug Register (BU3RHBDBG)

11.2.5.1.115.1 Offset

Register	Offset
BU3RHBDBG	D800h

11.2.5.1.115.2 Diagram



11.2.5.1.115.3 Fields

Field	Function
31-4 —	Reserved
3 TPCFG_TOUT_CTRL	This bit controls the USB 3.0 port configuration timeout duration. • 1: The port configuration timeout counter resets when the link is not in U0. • 0: The port configuration timeout counter does not reset if the link enters recovery or exits U0.
2-0 —	Reserved

11.2.5.1.116 Block RAM Single Bit Error Count Register (BRSERRCNT)

11.2.5.1.116.1 Offset

Register	Offset
BRSERRCNT	D850h

11.2.5.1.116.2 Function

8 bit count for each of the 3 RAMs.

11.2.5.1.116.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								RAM2SERRCNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RAM1SERRCNT								RAM0SERRCNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.116.4 Fields

Field	Function
31-24 —	Reserved
23-16 RAM2SERRCNT	RAM2 Single bit Error Count
15-8 RAM1SERRCNT	RAM1 Single bit Error Count
7-0 RAM0SERRCNT	RAM0 Single bit Error Count

11.2.5.1.117 Block RAM Multiple Bit Error Count Register (BRMERRCNT)

11.2.5.1.117.1 Offset

Register	Offset
BRMERRCNT	D854h

11.2.5.1.117.2 Function

8 bit count for each of the 3 RAMs.

11.2.5.1.117.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								RAM2MERRCNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RAM1MERRCNT								RAM0MERRCNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.117.4 Fields

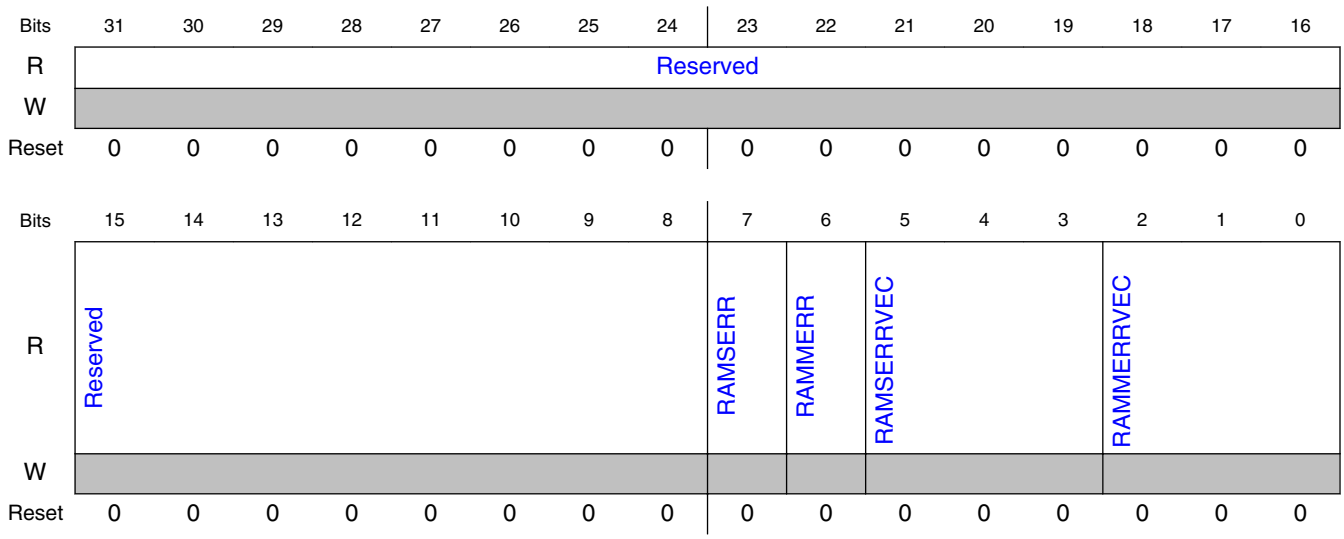
Field	Function
31-24 —	Reserved
23-16 RAM2MERRCNT	RAM2 Multiple bit Error Count
15-8 RAM1MERRCNT	RAM1 Multiple bit Error Count
7-0 RAM0MERRCNT	RAM0 Multiple bit Error Count

11.2.5.1.118 Block RAM ECC Error Vector Register (BRMECCERR)

11.2.5.1.118.1 Offset

Register	Offset
BRMECCERR	D858h

11.2.5.1.118.2 Diagram



11.2.5.1.118.3 Fields

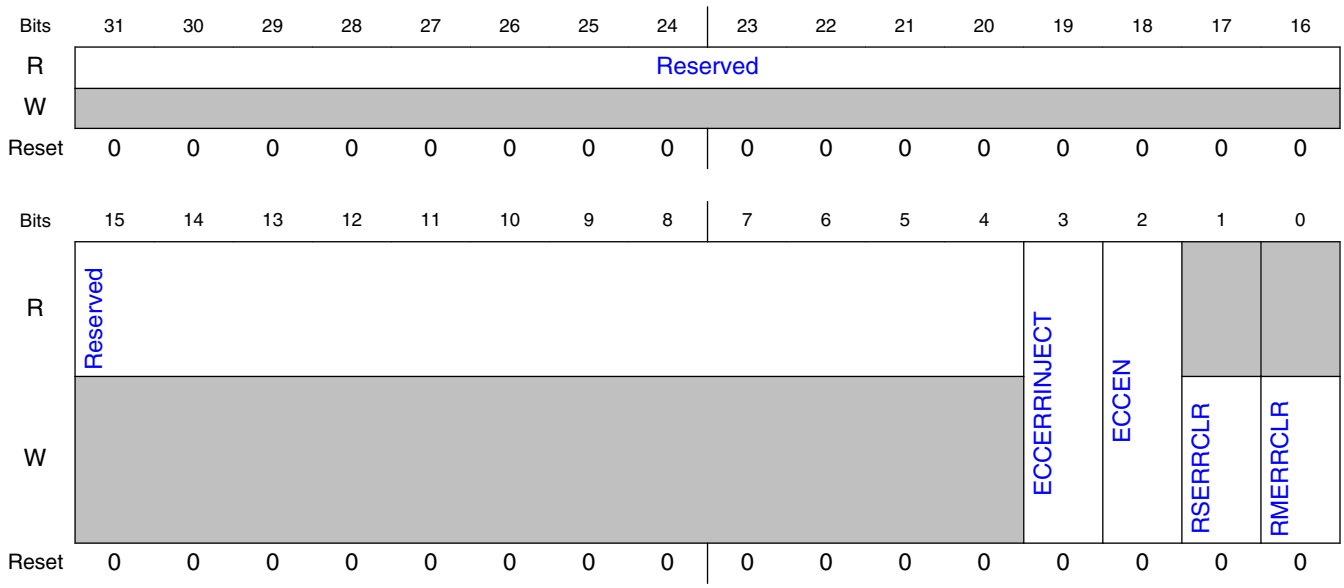
Field	Function
31-8 —	Reserved
7 RAMSERR	RAM Single bit Error Status
6 RAMMERR	RAM Multiple bit Error Status
5-3 RAMSERRVEC	RAM Single Error Vector
2-0 RAMMERRVEC	RAM Multiple Error Vector

11.2.5.1.119 Block RAM ECC Error Control Register (BRERRCTL)

11.2.5.1.119.1 Offset

Register	Offset
BRERRCTL	D85Ch

11.2.5.1.119.2 Diagram



11.2.5.1.119.3 Fields

Field	Function
31-4 —	Reserved
3 ECCERRINJECT	This bit should be used for testing purposes only. 1: Inject ECC error 0: No ECC error injected
2 ECCEN	ECC Enable (1) or Disable (0)
1 RSERRCLR	RAM Single bit Error Clear
0 RMERRCLR	RAM Multiple bit Error Clear

11.2.5.1.120 Block RAM0 Address Error Register (BRAM0ADDRERR)

11.2.5.1.120.1 Offset

Register	Offset
BRAM0ADDRERR	D860h

11.2.5.1.120.2 Function

Block RAM Address of RAM0 having uncorrectable error

11.2.5.1.120.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RAM0ADDRLOC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.120.4 Fields

Field	Function
31-16 —	Reserved
15-0 RAM0ADDRLOC	RAM0 Address Location

11.2.5.1.121 Block RAM1 Address Error Register (BRAM1ADDRERR)

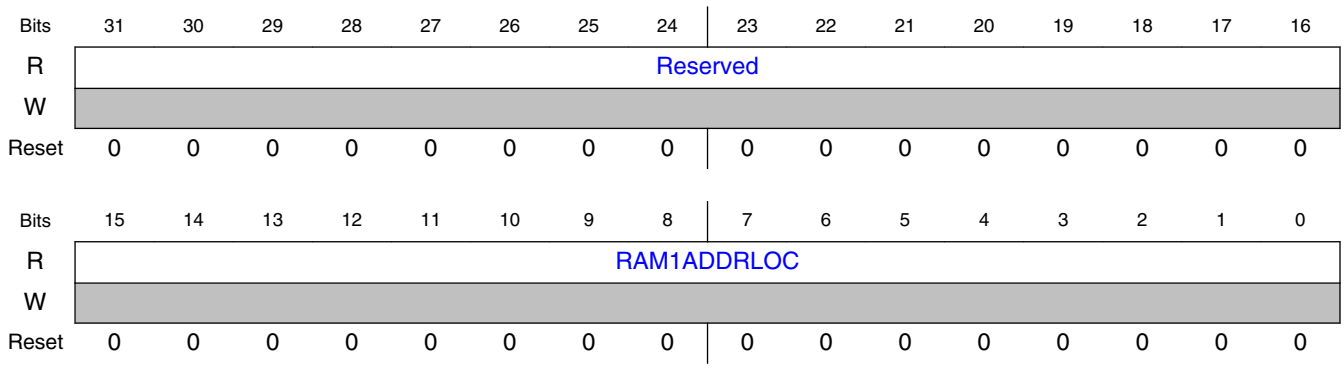
11.2.5.1.121.1 Offset

Register	Offset
BRAM1ADDRERR	D864h

11.2.5.1.121.2 Function

Block RAM1 Address of RAM1 having uncorrectable error

11.2.5.1.121.3 Diagram



11.2.5.1.121.4 Fields

Field	Function
31-16 —	Reserved
15-0 RAM1ADDRLOC	RAM1 Address Location

11.2.5.1.122 Block RAM2 Address Error Register (BRAM2ADDRERR)

11.2.5.1.122.1 Offset

Register	Offset
BRAM2ADDRERR	D868h

11.2.5.1.122.2 Function

Block RAM Address of RAM2 having uncorrectable error

11.2.5.1.122.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RAM2ADDRLOC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.5.1.122.4 Fields

Field	Function
31-16 —	Reserved
15-0 RAM2ADDRLOC	RAM2 Address Location

11.2.5.2 USB3_GLUE register descriptions

Between USB3.0 Core and USB PHY, there is a GLUE architecture to make the connection more configurable.

11.2.5.2.1 GLUE_usb3 memory map

USB GLUE architecture makes the connection between Core and PHY more configurable. USB GLUE contains some configurable registers and status signal monitoring to Core/PHY. There are two GLUE blocks and each has its own configurable registers.

USB1_GLUE base address: 381F_0000h

USB2_GLUE base address: 382F_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	USB3 Controller 0 (USB3_CTRL0)	32	RW	0020_F003h
4h	USB3 Controller 1 (USB3_CTRL1)	32	RW	0000_0000h
20h	USB3 Status 0 (USB3_STS0)	32	RO	0000_07E8h
40h	PHY Controller 0 (PHY_CTRL0)	32	RW	0000_CE10h
44h	PHY Controller 1 (PHY_CTRL1)	32	RW	0018_0006h
48h	PHY Controller 2 (PHY_CTRL2)	32	RW	0000_0300h
4Ch	PHY Controller 3 (PHY_CTRL3)	32	RW	88D4_66EBh
50h	PHY Controller 4 (PHY_CTRL4)	32	RW	042B_9E09h
54h	PHY Controller 5 (PHY_CTRL5)	32	RW	0000_0078h
58h	PHY Controller 6 (PHY_CTRL6)	32	RW	0000_0003h
80h	PHY Status 0 (PHY_STS0)	32	W1C	0000_0000h

11.2.5.2.2 USB3 Controller 0 (USB3_CTRL0)

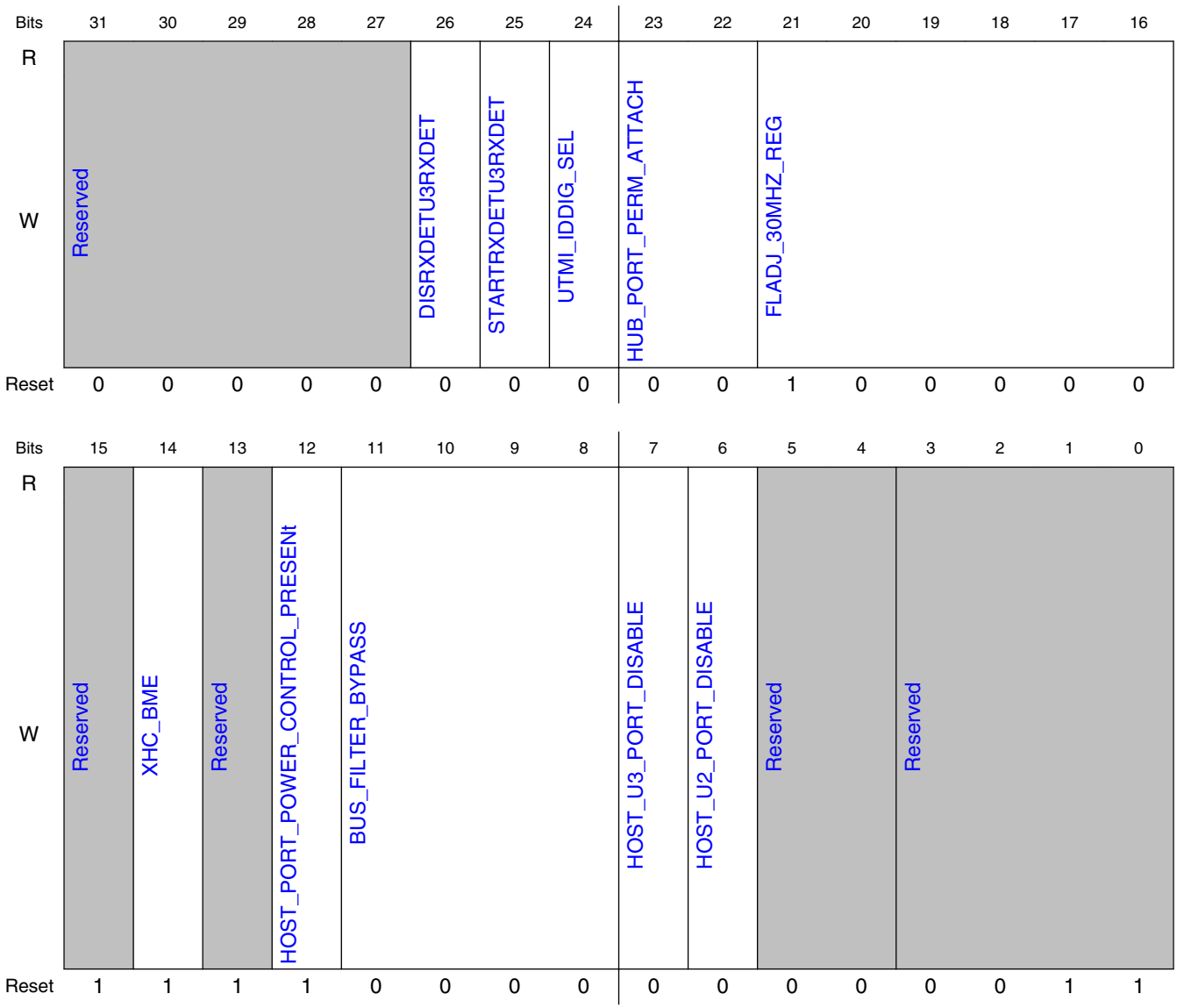
11.2.5.2.2.1 Offset

Register	Offset
USB3_CTRL0	0h

11.2.5.2.2.2 Function

USB Core Configurable bits Register

11.2.5.2.2.3 Diagram



11.2.5.2.2.4 Fields

Field	Function
31-27 —	Reserved
26 DISRXDETU3RXDET	DisRxDetU3RxDet of USB 3.0 SS Ports This input signal requests the controller to stop issuing more PHY commands and release the PIPE ownership. The controller acknowledges this request by asserting the DisRxDetU3RxDet_ack signal. This signal is for Type-C support.
25	StartRxdetU3RxDet of USB 3.0 SS Ports

Table continues on the next page...

Field	Function
STARTRXDETU3RXDET	This signal is the Start Receiver Detection in U3/Rx.Detect (StartRxdetU3RxDet). If DisRxDetU3RxDet is set and the link is in either U3 or Rx.Detect state, the controller starts receiver detection on the rising edge of this bit. This signal can only be used for downstream ports. It must be set to '0' for upstream ports.
24 UTMI_IDDIG_SEL	Iddig Source Select Signal 0b - USB PHY ID0 1b - GPIO PAD
23-22 HUB_PORT_PERM_ATTACH	Indicates if the device attached to a downstream port is permanently attached or not. bit[22] for usb2 port, bit[23] for usb3_SS port. 00b - Not permanently attached 01b - Permanently attached
21-16 FLADJ_30MHZ_REG	HS Jitter Adjustment. Indicates the correction required to accommodate mac3 clock and utmi clock jitter to measure 125 's duration. Default 0x20, each count is equal to 16 high speed bit times.
15 —	Reserved
14 XHC_BME	Disable the bus mastering capability of the xHC This signal is used to disable the bus mastering capability of the xHC. In a PCI system, it comes from the Bus Master Enable (BME) bit of the Device Control Register in the PCI Configuration register space. **NOTE:** In Host mode, For a non-PCI system, the xhc_bme is always tied to 1'b1 for the controller master to work. For a PCI system, connect the BME register bit of the PCI to this xhc_bme input. In Device mode, the xhc_bme can be any value, but it is recommended to tie this signal to 1. 0b - Bus mastering capability is disabled. The host controller cannot use the bus master interface. 1b - Bus mastering capability is enabled.
13 —	Reserved
12 HOST_PORT_POWER_CONTROL_PRESENt	This port defines the bit [3] of Capability Parameters (HCCPARAMS). Change the PPC value through the pin Port Power Control (PPC). This indicates whether the host controller implementation includes port power control. 0b - Indicates that the port does not have port power switches. 1b - Indicates that the port has port power switches.
11-8 BUS_FILTER_BYPASS	Bus Filter Bypass. It is expected that this signal is set or reset at power-on reset and is not changed during the normal operation of the controller. The function of each bit is: bus_filter_bypass[3]: Reserved bus_filter_bypass[2]: Bypass the filter for utmisrp_bvalid bus_filter_bypass[1]: Bypass the filter for pipe3_PowerPresent all U3 ports bus_filter_bypass[0]: Bypass the filter for utmi_vbusvalid all U2 ports In Host-only mode, internal bus filters are not needed. Therefore, bus_filter_bypass[2:0] must be connected to logic high value (3'b111). The reserved bit can be tied to 0 or 1, but should not be floating.
7 HOST_U3_PORT_DISABLE	USB 3.0 SS Port Disable Control. This signal, when '1', stops reporting connect/disconnect events the port and keeps the port in disabled state. This could be used for security reasons where hardware can disable a port irrespective of whether xHCI driver enables a port or not.

Table continues on the next page...

Field	Function
	The 'Number of Ports' field of the HCSPARAMS1 register is not affected by this signal. This signal should be either static (should not change during operation), or change only once from 0 to 1 during operation and stay at 1 after that. **NOTE:** This signal is not present in USB 2.0-only mode. **NOTE:** The SuperSpeed Port0 (first SS port) must not be disabled if the other SuperSpeed ports are operational. 0b - Port Enabled 1b - Port Disabled
6 HOST_U2_PORT_DISABLE	USB 2.0 Port Disable Control. This could be used for security reasons where hardware can disable a port irrespective of whether xHCI driver enables a port or not. The 'Number of Ports' field of the HCSPARAMS1 register is not affected by this signal. This signal should be either static (should not change during operation), or change only once from 0 to 1 during operation and stay at 1 after that. Note: Port-0 (the first U2 port) must not be disabled if any other U2 port is operational. 0b - Port Enabled 1b - Port Disabled. When '1', this signal stops reporting connect/disconnect events the port and keeps the port in disabled state.
5-4 —	Reserved
3-0 —	Reserved

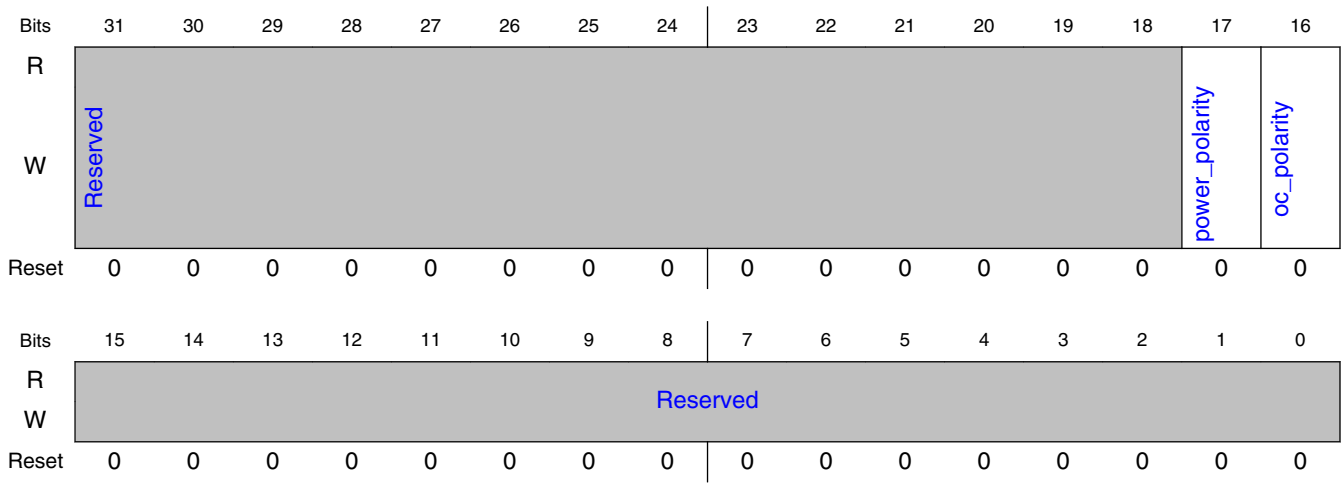
11.2.5.2.3 USB3 Controller 1 (USB3_CTRL1)

11.2.5.2.3.1 Offset

Register	Offset
USB3_CTRL1	4h

11.2.5.2.3.2 Function

11.2.5.2.3.3 Diagram



11.2.5.2.3.4 Fields

Field	Function
31-18 —	Reserved
17 power_polarity	Power polarity Used to control external PWR pad 0b - Active high 1b - Active low
16 oc_polarity	Overcurrent polarity 0b - Active high 1b - Active low
15-0 —	Reserved

11.2.5.2.4 USB3 Status 0 (USB3_STS0)

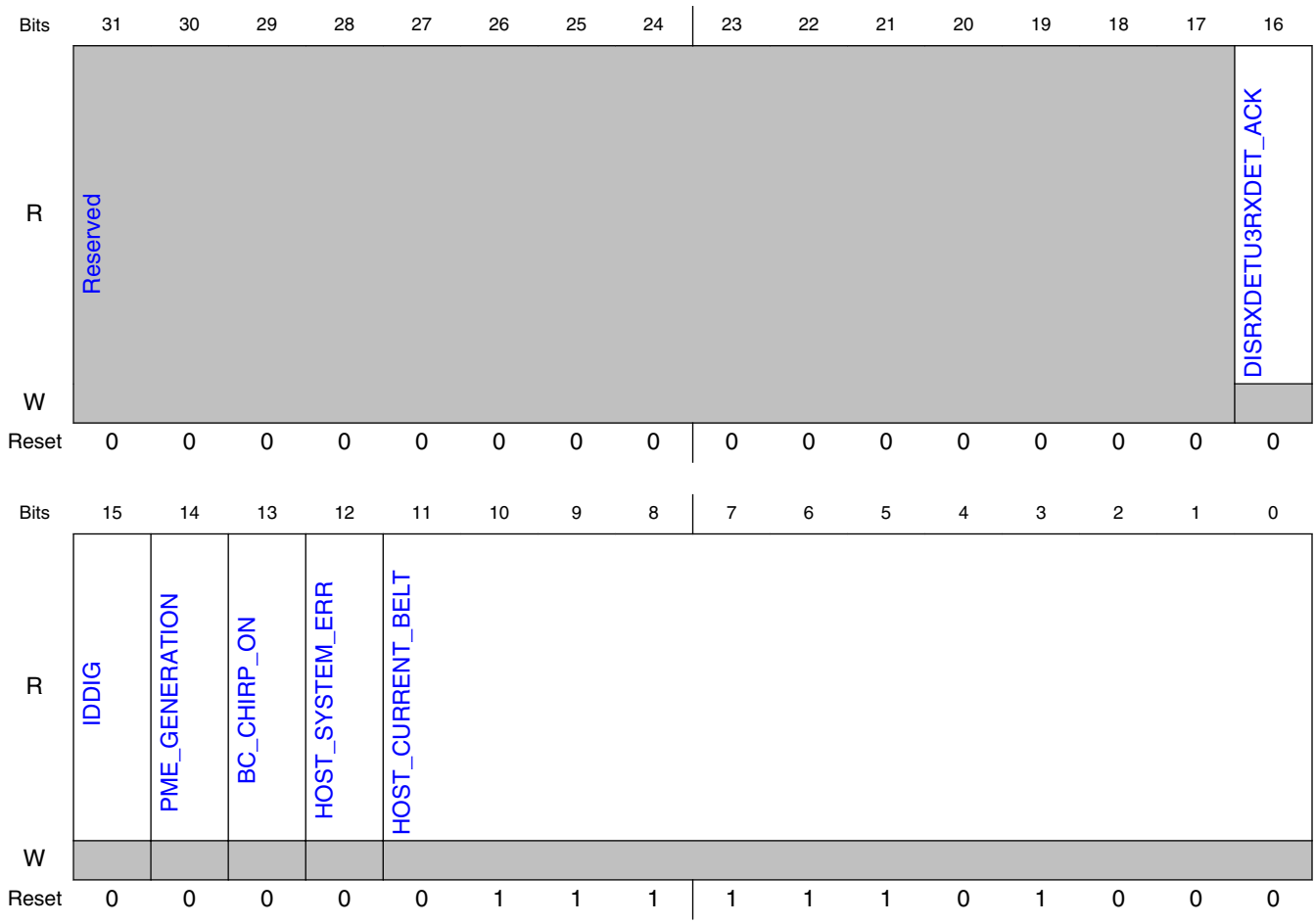
11.2.5.2.4.1 Offset

Register	Offset
USB3_STS0	20h

11.2.5.2.4.2 Function

USB3.0 Core Status bits Register

11.2.5.2.4.3 Diagram



11.2.5.2.4.4 Fields

Field	Function
31-17 —	Reserved
16 DISRXDETU3RXDET_ACK	DisRxDetU3RxDet_ack of USB 3.0 SS Ports This signal is an acknowledgment of DisRxDetU3RxDet. When this signal is 1'b1, the controller releases the ownership of the the PIPE interface and does not issue PHY commands to the PIPE interface.
15 IDDIG	This controller signal indicates whether the connected plug is a mini-A or mini-B plug. This signal must be valid within 20 ms after the IDPULLUP0 signal is set to 1’b1. 0b - Connected plug is a mini-A plug. 1b - Connected plug is a mini-B plug.
14 PME_GENERATION	PME# Generation. This signal is used to generate a PME# (Power Management Event). When the Run/Stop bit of the USB Command Register is cleared during USB suspend mode, the controller cannot generate an event and

Table continues on the next page...

Field	Function
	cannot assert a regular interrupt. In this case, the controller asserts a pme_generation signal to report any wakeup condition if pme_en is high. This is a level signal that gets cleared when the software clears CSC, OCC and PLC bits in all the PORTSC registers.
13 BC_CHIRP_ON	When asserted indicates an imminent chirp signal.
12 HOST_SYSTEM_ERR	Host System Error This signal indicates that a Host System Error has occurred as reflected in the USBSTS[HSE] field. This signal is asserted only if the USBCMD[HSEE] field is set to '1'. It can occur when the host controller encounters an 'Error' response in the AHB, the AXI, or the Native Master Bus. When the USBSTS[HSE] field is cleared by software, this signal is deasserted unless the master continues to assert its bus error output. The typical software response to an HSE is to reset the controller.
11-0 HOST_CURRENT_BELT	Current BELT Value This signal indicates the minimum value of all received BELT values and the BELT that is set by the Set LTV command. This signal is valid only in Host mode.

11.2.5.2.5 PHY Controller 0 (PHY_CTRL0)

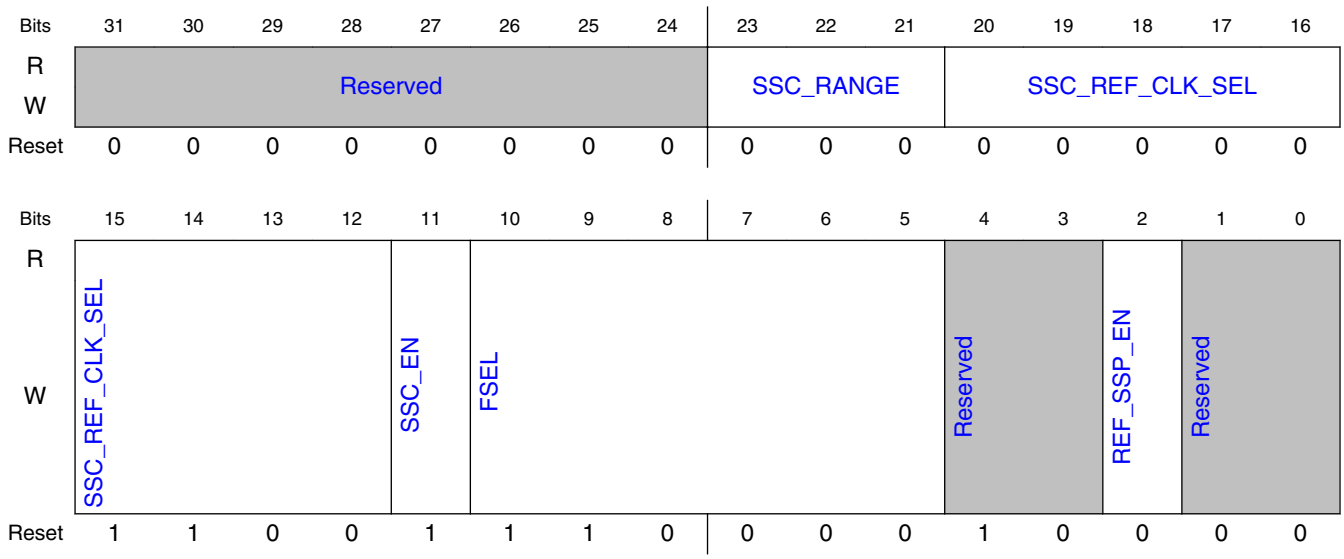
11.2.5.2.5.1 Offset

Register	Offset
PHY_CTRL0	40h

11.2.5.2.5.2 Function

USB3.0 PHY Status bits Register

11.2.5.2.5.3 Diagram



11.2.5.2.5.4 Fields

Field	Function
31-24 —	Reserved
23-21 SSC_RANGE	Spread Spectrum Clock Range Selects the range of spread spectrum modulation when ssc_en is asserted and the PHY is spreading the high-speed transmit clocks. Applies a fixed offset to the phase accumulator. ssc_range[2:0] Down Spread of Clock (ppm) 000b - 4980 001b - 4492 010b - 4003
20-12 SSC_REF_CLK_SEL	Spread Spectrum Reference Clock Shifting Enables non-standard oscillator frequencies to generate targeted MPLL output rates. Input corresponds to frequency-synthesis coefficient. ssc_ref_clk_sel[8:6] = modulous – 1 ssc_ref_clk_sel[5:0] = 2's complement push amount
11 SSC_EN	Spread Spectrum Enable Enables spread spectrum clock production (0.5% down-spread at ~ 31.5 KHz) in the USB 3.0 femtoPHY. Set this signal to 1'b1.
10-5 FSEL	100111b - 100 MHz ref clock 101010b - 24 MHz ref clock
4-3 —	Reserved
2	Reference Clock Enable for SS function.

Table continues on the next page...

Field	Function
REF_SSP_EN	Enables the reference clock to the prescaler. The ref_ssp_en signal must remain deasserted until the reference clock is running at the appropriate frequency, at which point ref_ssp_en can be asserted. For lower power states, ref_ssp_en can also be deasserted.
1-0 —	Reserved

11.2.5.2.6 PHY Controller 1 (PHY_CTRL1)

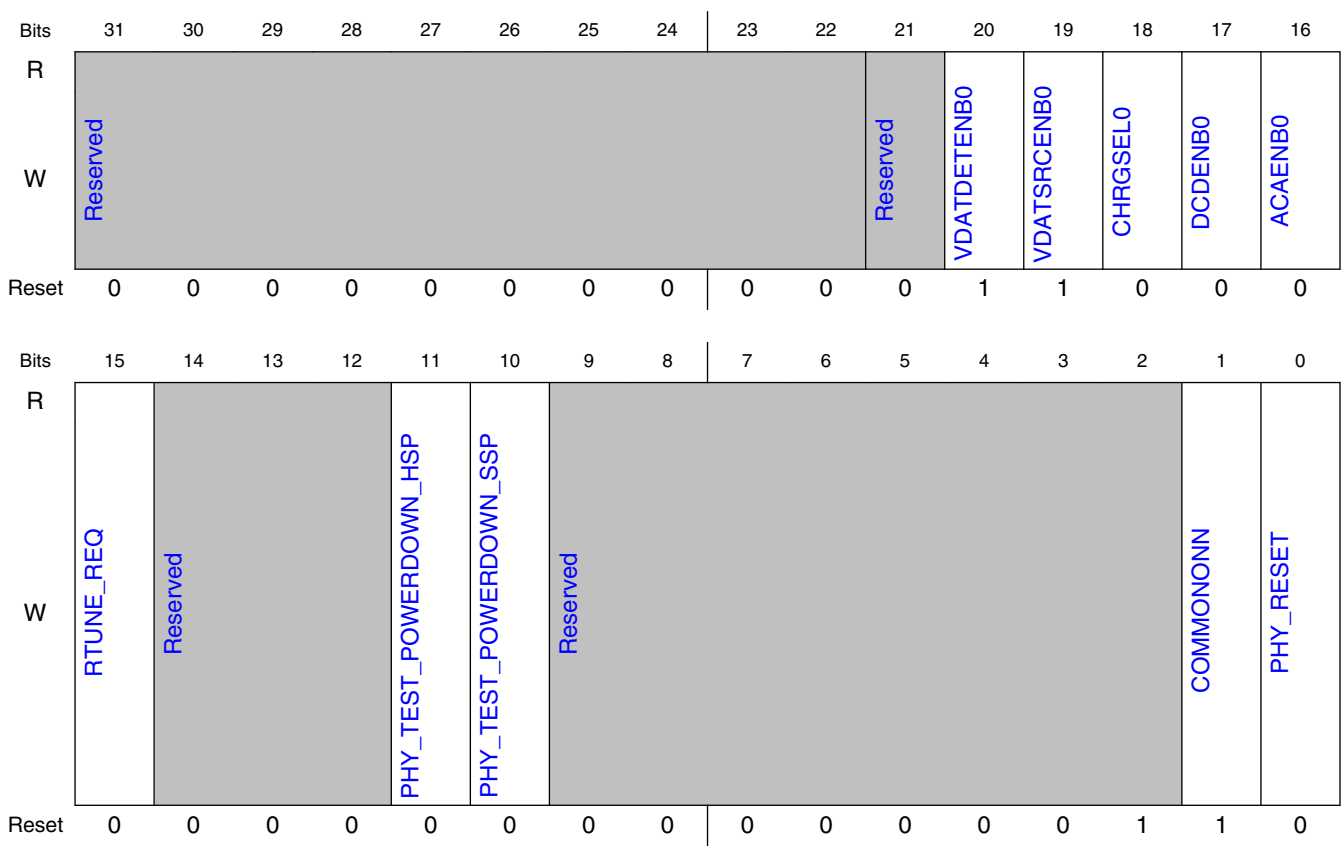
11.2.5.2.6.1 Offset

Register	Offset
PHY_CTRL1	44h

11.2.5.2.6.2 Function

USB3.0 PHY Status bits Register

11.2.5.2.6.3 Diagram



11.2.5.2.6.4 Fields

Field	Function
31-22 —	Reserved
21 —	Reserved
20 VDATDETENB0	Battery Charging Attach/Connect Detection Enable Enables or disables attach/connect detection. 1: Data detect voltage (CHG_DET) is enabled. 0: Data detect voltage (CHG_DET) is disabled
19 VDATSRCENB0	Battery Charging Sourcing Select Enables or disables sourcing for battery charging. 1: Data source voltage (VDAT_SRC) is enabled. 0: Data source voltage (VDAT_SRC) is disabled.
18 CHRGSEL0	Battery Charging Source Select Determines whether current is sourced onto or sunk from DPn or DMn. 1: Data source voltage (VDAT_SRC) is sourced onto DMn and sunk from DPn. 0: Data source voltage (VDAT_SRC) is sourced onto DPn and sunk from DMn.
17 DCDENB0	Data Contact Detection Enable Enables current sourcing on the D+ line. For special applications, the IDP_SRC control and the RDM_DWN control are separated. For standard DCD operation in accordance with the Battery Charger v1.2 specification,simultaneously set DCDENBn = 1'b1 (to enable IDP_SRC) and set DMPULLDOWNn = 1'b1 (to enable RDM_DWN). Set both signals to 1'b0 when DCD is completed. 1: IDP_SRC current is sourced onto DPn, pull-down resistance on DMn is enabled. 0: IDP_SRC current is disabled, pull-down resistance on DMn is disabled. **NOTE:** During USB 3.0 femtoPHY normal operation, set this signal low
16 ACAENB0	Battery Charging Source Select Determines whether current is sourced onto or sunk from DPN or DMN. 1: Data source voltage (VDAT_SRC) is sourced onto DMN and sunk from DPN. 0: Data source voltage (VDAT_SRC) is sourced onto DPN and sunk from DMN.
15 RTUNE_REQ	Resistor Tune Request Assertion triggers a resistor tune request (if one is not already in progress). When this signal is asserted, rtune_ack goes high until calibration of the termination impedances is complete. Note that termination resistance tuning disrupts the normal flow of data; therefore, assert rtune_req only when the PHY is inactive. Because the PHY automatically performs a tune after coming out of a reset, rtune_req can be tied low unless you want to be able to manually force a tuning.
14-12 —	Reserved
11	USB3.0 PHY Signal

Table continues on the next page...

Field	Function
PHY_TEST_POWERDOWN_HSP	Powers down all HS function circuitry in the PHY for IDDQ testing. Before asserting this signal, ensure that VDATSRCENBn, VDATDETENBn, DCDENBn, BYPASSSELn, and test_burnin are set to 1'b0 now it's 1 if ioring_ipp_ind_test_mode.
10 PHY_TEST_POWERDOWN_SSP	USB3.0 PHY Signal Powers down all SS function circuitry in the PHY for IDDQ testing ioring_test_mode 0.
9-2 —	Reserved
1 COMMONONN	Common Block Power-Down Control Controls the power-down signals in the PLL block when the USB 3.0 femtoPHY is in Suspend or Sleep mode. While the PHY is not in either Suspend mode nor Sleep state, COMMONONN can be dynamically controlled. 1: In Suspend or Sleep mode, the PLL block is powered down. 0: In Suspend or Sleep mode, the PLL block remains powered and continues to draw current.
0 PHY_RESET	USB3.0 PHY Signal Resets the core and all state machines with the exception of the reference clock buffer and the JTAG interface. Asserting phy_reset triggers the assertion of the tx<#>_reset and rx<#>_reset signals.

11.2.5.2.7 PHY Controller 2 (PHY_CTRL2)

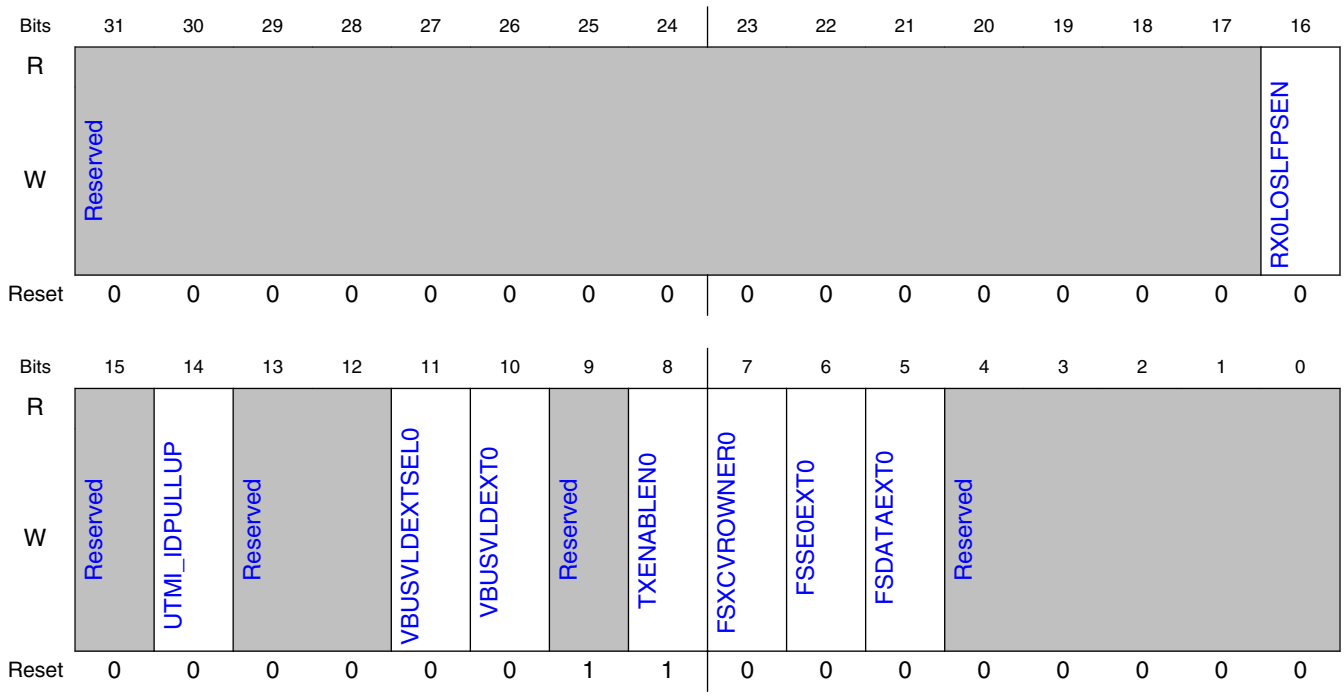
11.2.5.2.7.1 Offset

Register	Offset
PHY_CTRL2	48h

11.2.5.2.7.2 Function

USB3.0 PHY Status bits Register

11.2.5.2.7.3 Diagram



11.2.5.2.7.4 Fields

Field	Function
31-17 —	Reserved
16 RX0LOSLFPSEN	RX LOS LFPS Filter Enable Enables the RX LOS LFPS filter. During USB 3.0 operation, set to 1'b1. Change this pin setting only during a reset. **NOTE:** In Scan mode, tie this pin to 0 1: Enables RX LOS LFPS filter. 0: Disables RX LOS LFPS filter.
15 —	Reserved
14 UTMI_IDPULLUP	Analog ID Input Sample Enable This controller signal controls IDN line sampling. 1: IDN pin sampling is enabled, and the IDDIG0 output is valid. 0: IDN pin sampling is disabled, and the IDDIG0 output is not valid.
13-12 —	Reserved
11	Selects the VBUSVLDEXTn input or the internal Session Valid comparator to indicate when the VBUSn signal on the USB cable is valid.

Table continues on the next page...

Field	Function
VBUSVLDEXTSEL0	This signal is a strapping option that must be set prior to a power-on reset and remain static during normal operation. Strapping options are not critical for STA,and any other timings or loading limits for the pin are specified in the .lib timing model included in the product deliverables. 1: VBUSVLDEXTn input is used. 0: Internal Session Valid comparator is used.
10 VBUSVLDEXT0	External VBUS Valid Indicator Valid in Device mode and only when the VBUSVLDEXTSEL0 signal is set to 1'b1. VBUSVLDEXTn indicates whether the VBUSn signal on the USB cable is valid. In addition, VBUSVLDEXTn enables the pull- up resistor on the D+ line. In Host mode, this input is not used and can be tied to 1'b0. 1: VBUSn signal is valid, and the pull-up resistor on D+ is enabled. 0: VBUSn signal is not valid, and the pull-up resistor on D+ is disabled.
9 —	Reserved
8 TXENABLEN0	USB 1.1 Data Enable This controller signal enables the FSDATAEXTn and FSSE0EXTn inputs. TXENABLENn is valid only when the FSXCVROWNERn signal is set to 1'b1. 1: FSDATAEXTn and FSSE0EXTn are disabled. 0: FSDATAEXTn and FSSE0EXTn are enabled.
7 FSXCVROWNER0	UTMI+/Serial Interface Select This controller signal enables the UTMI+ or serial interface. 1: TXENABLENn, FSDATAEXTn, and FSSE0EXTn inputs drive USB 2.0 data output onto the D+ and D– lines. Data that the USB 3.0 femtoPHY receives from the D+ and D– lines appears on the FSVMINUSn and FSVPLUSn outputs. 0: Data on the D+ and D– lines is transmitted and received through the UTMI+.
6 FSSE0EXT0	USB 1.1 Transmit Data This controller signal sets the USB (2.0) to either a J or K state. This signal is valid only if FSXCVROWNERn is set to 1'b1, TXENABLENn is set to 1'b0, and FSSE0EXTn is set to 1'b0. 1: D+ and D– lines are driven to a Differential 1. 0: D+ and D– lines are driven to a Differential 0.
5 FSDATAEXT0	USB 1.1 SE0 Generation This controller signal sets the USB (2.0) to an SE0 state. This signal is valid only if FSXCVROWNERn is set to 1'b1 and TXENABLENn is set to 1'b0.
4-0 —	Reserved

11.2.5.2.8 PHY Controller 3 (PHY_CTRL3)

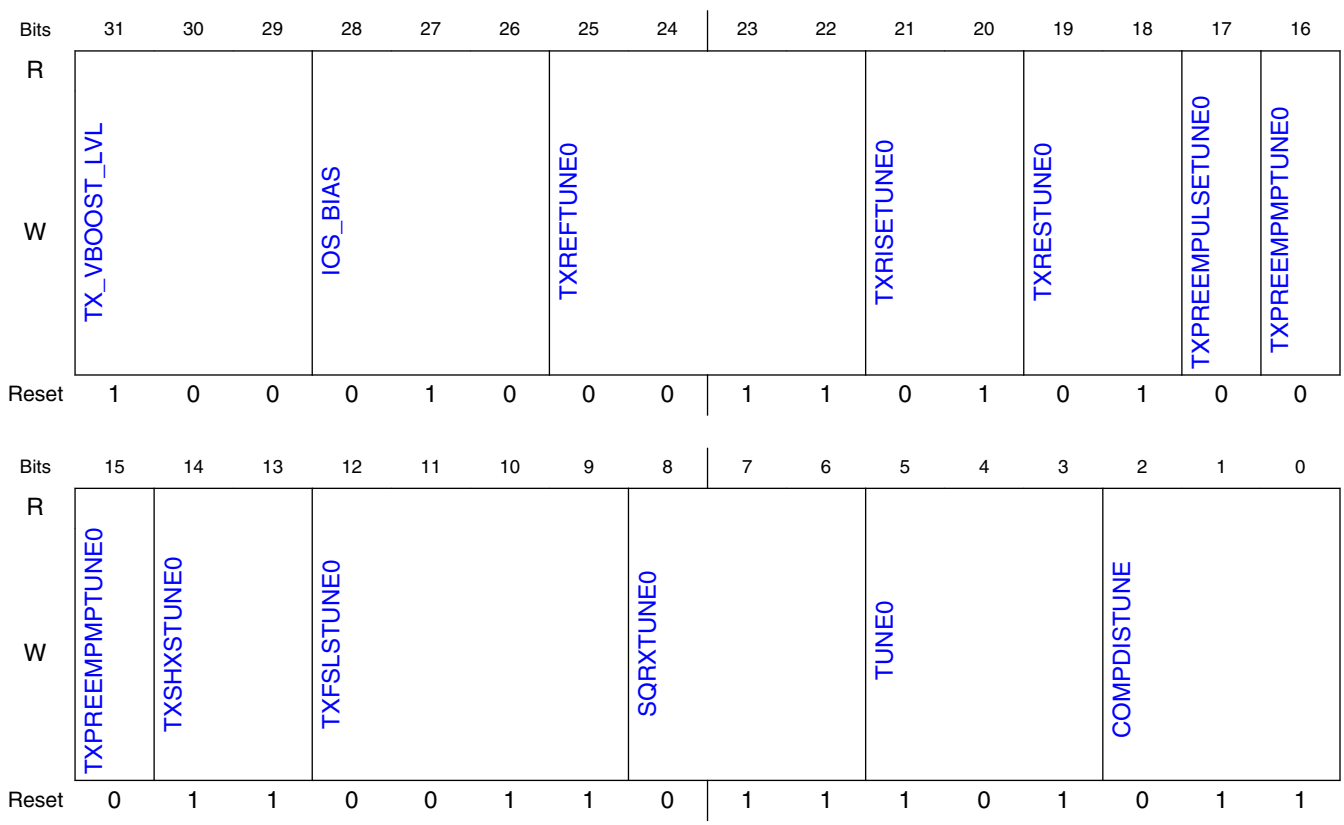
11.2.5.2.8.1 Offset

Register	Offset
PHY_CTRL3	4Ch

11.2.5.2.8.2 Function

USB3.0 PHY Status bits Register

11.2.5.2.8.3 Diagram



11.2.5.2.8.4 Fields

Field	Function
31-29 TX_VBOOST_LVL	TX Voltage Boost Level Sets the boosted transmit launch amplitude (mVppd). To enable tuning at the board level, connect this bus to a register. Valid settings: 011: Corresponds to a launch amplitude of 1.12 V. 010: Corresponds to a launch amplitude of 1.04 V.

Table continues on the next page...

Field	Function
	000: Corresponds to a launch amplitude of 0.88 V.
28-26 IOS_BIAS	Loss-of-Signal Detector Threshold Level Control Sets the LOS detection threshold level. To enable tuning at the board level, connect this bus to a register. 111: 135 mV 110: 120 mV 101: 105 mV 100: 90 mV 011: 75 mV 010: 60 mV (default) 001: 45 mV 000: Invalid
25-22 TXREFTUNE0	HS DC Voltage Level Adjustment Adjusts the high-speed DC level voltage. To enable tuning at the board level, connect this bus to a register. 0000b - - 6% 0001b - - 4% 0010b - - 2% 0011b - 0 (default) 0100b - + 2% 0101b - + 4% 0110b - + 6% 0111b - + 8% 1000b - + 10 1001b - + 12% 1010b - + 14% 1011b - + 16% 1100b - + 18% 1101b - + 20% 1110b - + 22% 1111b - + 24%
21-20 TXRISETUNE0	HS Transmitter Rise/Fall Time Adjustment Adjusts the rise/fall times of the high-speed waveform. To enable tuning at the board level, connect this bit to a register. 00b - + 3% 01b - 0 (default) 10b - - 1% 11b - - 3%
19-18 TXRESTUNE0	USB Source Impedance Adjustment Some applications require additional devices to be added on the USB, such as a series switch, which can add significant series resistance. This bus adjusts the driver source impedance to compensate for added series resistance on the USB. **NOTE:** Any setting other than the default can result in source impedance variation across process, voltage, and temperature conditions that does not meet USB 2.0 specification limits. If this bus is not used, leave it at the default setting 00b - Source impedance is increased by approximately 2 Ω 01b - 0 (Design default) 10b - Source impedance is decreased by approximately 3 Ω 11b - Source impedance is decreased by approximately 4.5 Ω

Table continues on the next page...

Field	Function
17 TXPREEMPULSETUNE0	HS Transmitter Pre-Emphasis Duration Control This signal controls the duration for which the HS pre-emphasis current is sourced onto DP or DM. The HS Transmitter pre-emphasis duration is defined in terms of unit amounts. One unit of pre-emphasis duration is approximately 580 ps and is defined as 1X pre-emphasis duration. This signal is valid only if either TXPREEMPMPTUNE[1] or TXPREEMPMPTUNE[0] is set to 1'b1. 0b - long pre-emphasis current duration (design default). If TXPREEMPULSETUNE is not used, set it to 1'b0 1b - Short pre-emphasis current duration
16-15 TXPREEMPMPTUNE0	HS Transmitter Pre-Emphasis Current Control This signal controls the amount of current sourced to DPn and DMn after a J-to-K or K-to-J transition. The HS Transmitter pre-emphasis current is defined in terms of unit amounts. One unit amount is approximately 600 uA and is defined as 1X pre-emphasis current. 00b - Disabled (default) 01b - 1x pre-emphasis current 10b - 2x pre-emphasis current 11b - 3x pre-emphasis current
14-13 TXSHXSTUNE0	Transmitter High-Speed Crossover Adjustment. This bus adjusts the voltage at which the DPn and DMn signals cross while transmitting in HS mode. 00b - Reserved 01b - - 16 mV 10b - + 14 mV 11b - 0 (default)
12-9 TXFSLSTUNE0	FS/LS Source Impedance Adjustment Adjusts the low- and full-speed single-ended source impedance while driving high. To enable tuning at the board level, connect this bus to a register. This parameter control is encoded in thermometer code. Any non-thermometer code setting (that is, 1001) is reserved and not supported. 0000b - + 3.5% 0001b - + 1.8% 0011b - 0 (default) 0111b - - 1.7% 1111b - - 3.5%
8-6 SQRXTUNE0	Squelch Threshold Adjustment Adjusts the voltage level for the threshold used to detect valid high-speed data. To enable tuning at the board level, connect this bus to a register. 000b - + 16.07% 001b - + 10.71% 010b - + 5.36% 011b - 0 (default) 100b - - 5.36% 101b - - 10.71% 110b - - 16.07% 111b - - 22.32%
5-3 TUNE0	VBUS Valid Threshold Adjustment This bus adjusts the voltage level for the VBUS valid threshold. To enable tuning at the board level, connect this bus to a register. 111: + 5.8% 110: + 2.9%

Table continues on the next page...

Field	Function
	101: 0 (default) 100: – 2.9% 011: – 5.8% 010: – 8.7% 001: – 11.6% 000: – 14.5%
2-0 COMPDISTUNE	Disconnect Threshold Adjustment. Adjusts the voltage level for the threshold used to detect a disconnect event at the host. To enable tuning at the board level, connect this bus to a register. 000b - - 9.01% 001b - - 6.5% 010b - - 3.04% 011b - 0 (default) 100b - + 4.08 101b - + 7.52% 110b - + 11.86% 111b - + 15.54%

11.2.5.2.9 PHY Controller 4 (PHY_CTRL4)

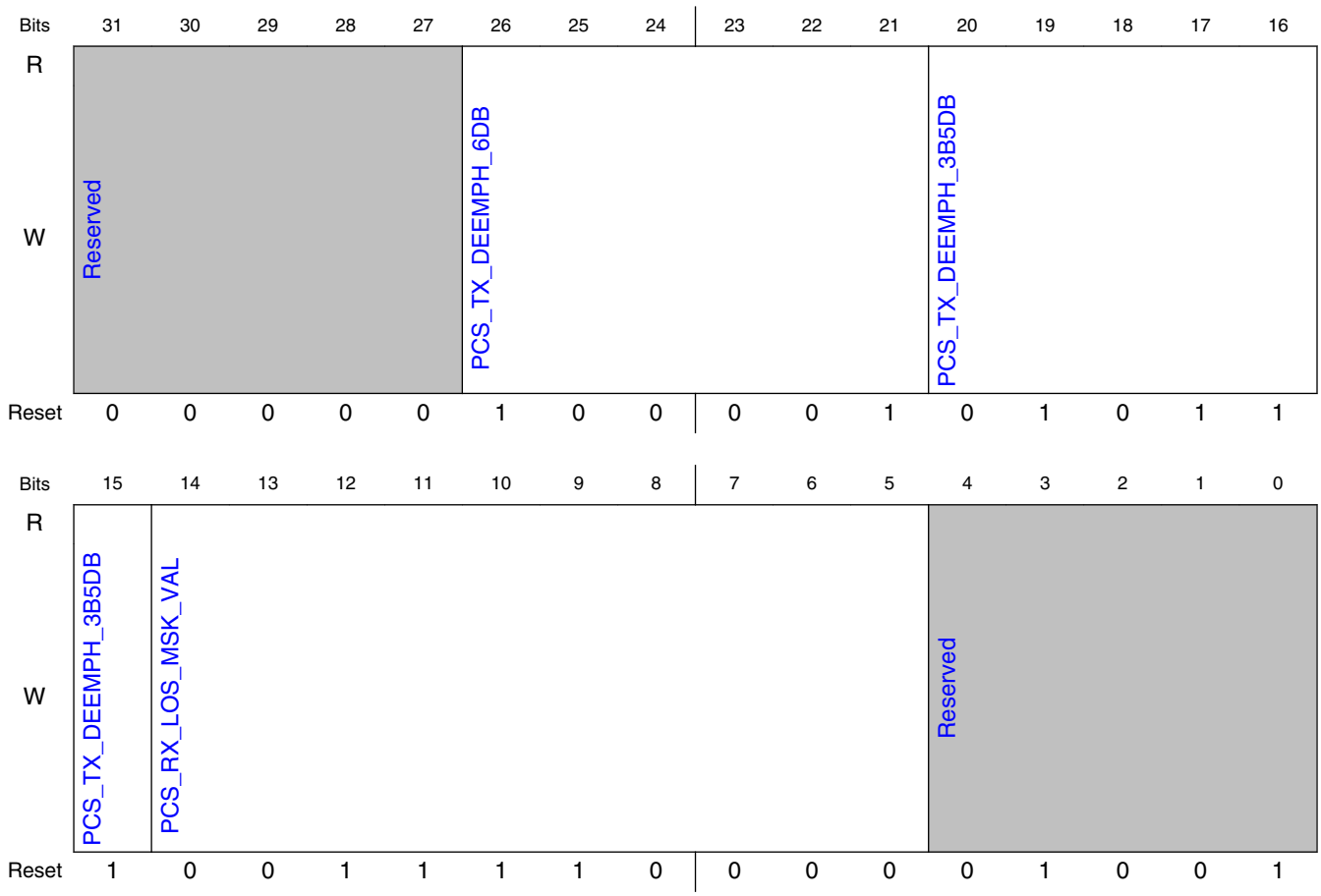
11.2.5.2.9.1 Offset

Register	Offset
PHY_CTRL4	50h

11.2.5.2.9.2 Function

USB3.0 PHY Status bits Register

11.2.5.2.9.3 Diagram



11.2.5.2.9.4 Fields

Field	Function
31-27 —	Reserved
26-21 PCS_TX_DEEMPH_6DB	TX De-Emphasis at 6 dB This static value sets the TX driver de-emphasis value when pipeP_tx_deemph[1:0] is set to 2’b10 (according to the PIPE3 specification). This bus is provided for completeness and as a second potential launch amplitude.
20-15 PCS_TX_DEEMPH_3B5DB	TX De-Emphasis at 3.5 dB This static value sets the TX driver de-emphasis value when pipeP_tx_deemph[1:0] is set to 2’b01 (according to the PIPE3 specification). To enable tuning at the board level for RX eye compliance, connect this signal to a register.
14-5 PCS_RX_LOS_MSK_VAL	Configurable Loss-of-Signal Mask Width Sets the number of reference clock cycles to mask the incoming LFPS in U3 and U2 states. Masks the incoming LFPS for the number of reference clock cycles equal to the value of pcs_rx_los_mask_val[9:0]. This control filters out short, non-compliant LFPS glitches sent by a non-compliant host. For normal operation, set to a targeted mask interval of 10 µs (value = 10 µs / Tref_clk). If the ref_clkdiv2 signal is

Table continues on the next page...

Field	Function
	used, the value = 10 µs / (2 * Tref_clk). Setting this bus to 10'b0 disables masking. This bus should be accessible to general configuration registers for system testing and debug. The value should be defined when in reset. Changing this value during operation might disrupt normal operation of the link.Sets the number of reference clock cycles to mask the incoming LFPS. For normal operation, set to a targeted mask interval of 10 µs (value = 10 µs / Tref_clk). If the ref_clkdiv2 signal is used, the value = 10 µs / (2 * Tref_clk).
4-0 —	Reserved

11.2.5.2.10 PHY Controller 5 (PHY_CTRL5)

11.2.5.2.10.1 Offset

Register	Offset
PHY_CTRL5	54h

11.2.5.2.10.2 Function

USB3.0 PHY Status bits Register

11.2.5.2.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PCS_TX_SWING_FULL							
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0

11.2.5.2.10.4 Fields

Field	Function
31-7 —	Reserved
6-0	TX Amplitude (Full Swing Mode)

Field	Function
PCS_TX_SWING_FULL	This static value sets the launch amplitude of the transmitter. To enable tuning at the board level for RX eye compliance, connect this signal to a register.

11.2.5.2.11 PHY Controller 6 (PHY_CTRL6)

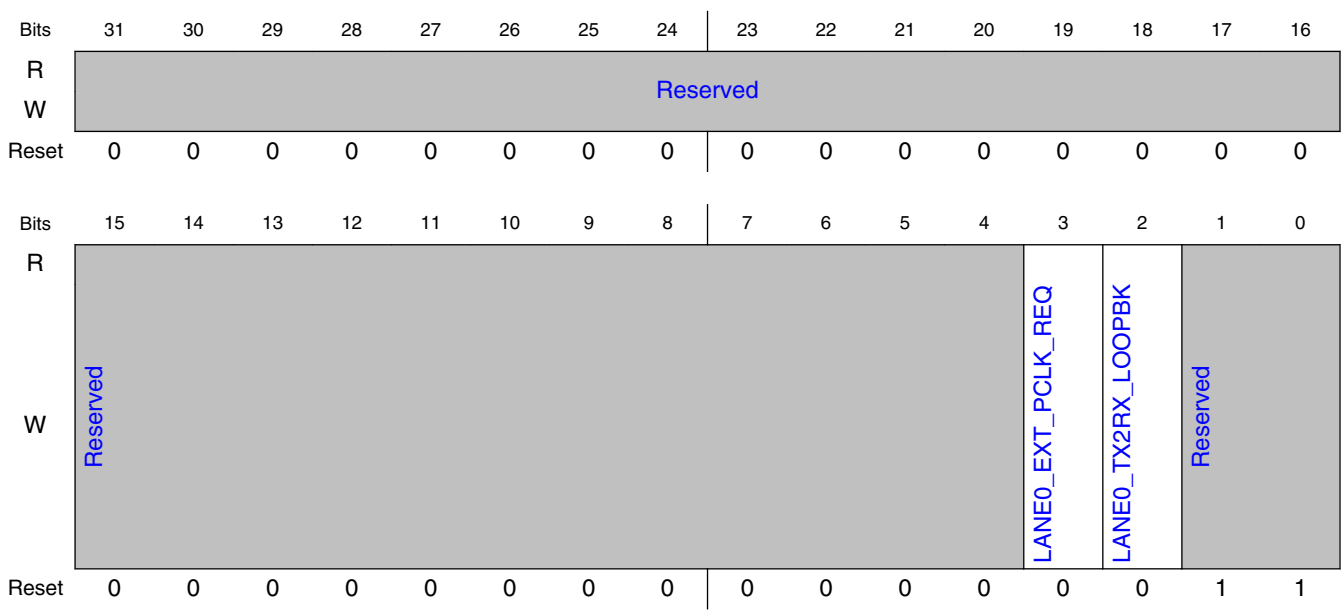
11.2.5.2.11.1 Offset

Register	Offset
PHY_CTRL6	58h

11.2.5.2.11.2 Function

USB3.0 PHY Status bits Register

11.2.5.2.11.3 Diagram



11.2.5.2.11.4 Fields

Field	Function
31-4 —	Reserved

Table continues on the next page...

Field	Function
3 LANE0_EXT_PCLK_REQ	External PIPE Clock Enable Request When asserted, this signal enables the pipeP_pclk output regardless of power state (along with the associated increase in power consumption). You can use this input to enable pipeP_pclk in the P3 state without going through a complete boot sequence. Theoretically, this signal enables you to disable the RX terminations and use the PHY as a high-speed clock source.
2 LANE0_TX2RX_LOOPBK	Loopback When this signal is asserted, data from the transmit predriver is looped back to the receive slicers. LOS is bypassed and based on the tx0_en input so that rx0_los = !tx0_data_en.
1-0 —	Reserved

11.2.5.2.12 PHY Status 0 (PHY_STS0)

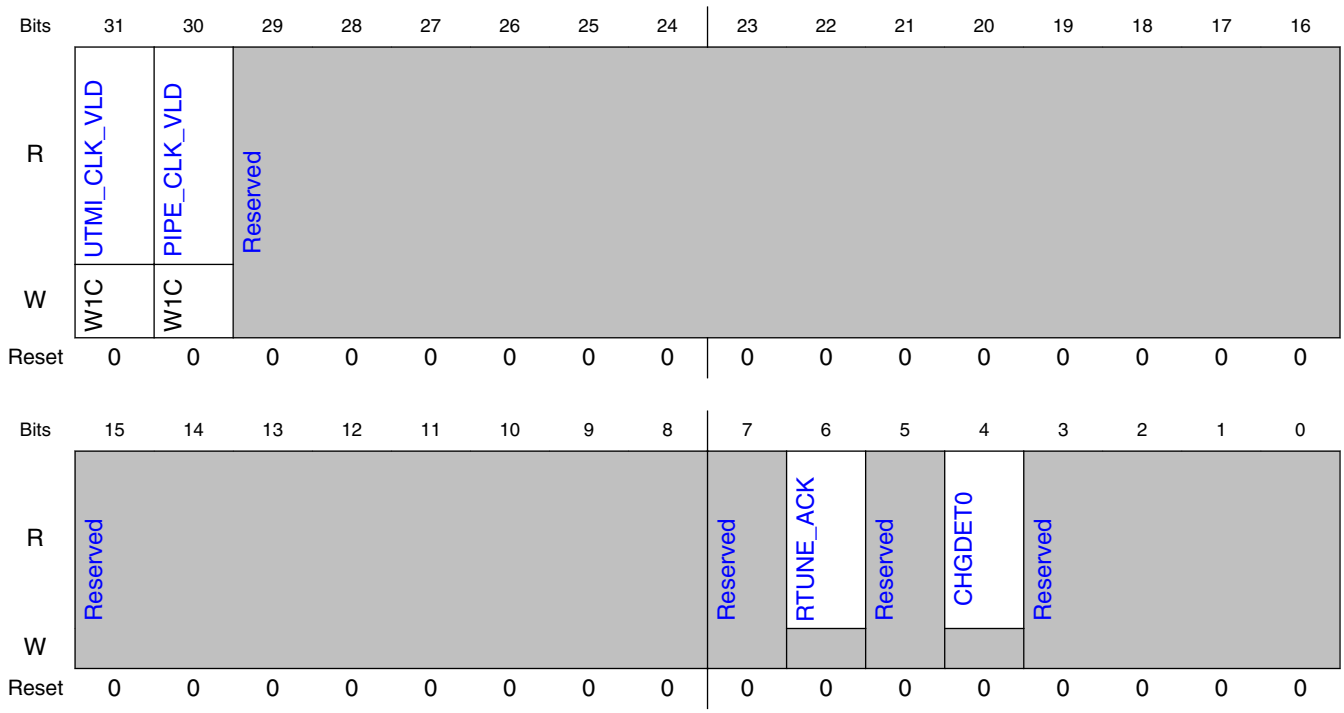
11.2.5.2.12.1 Offset

Register	Offset
PHY_STS0	80h

11.2.5.2.12.2 Function

USB3.0 PHY Status bits Register

11.2.5.2.12.3 Diagram



11.2.5.2.12.4 Fields

Field	Function
31 UTMI_CLK_VLD	USB3.0 PHY Signal synchronised by USB bus clock.After PHY and core reset pipe clock is stable if this bit is set.
30 PIPE_CLK_VLD	USB3.0 PHY Signal synchronised by USB bus clock.After PHY and core reset pipe clock is stable if this bit is set.
29-8 —	Reserved
7 —	Reserved
6 RTUNE_ACK	Resistor Tune Acknowledge. While asserted, indicates that a resistor tune is still in progress.
5 —	Reserved
4 CHGDET0	Battery Charger Detection Output Indicates whether the voltage level on DP or DM is greater than VDAT_REF as defined in the Battery Charger specification. 1: VDP > VDAT_REF (for CHRGSELn = 1’b1) or VDM > VDAT_REF (for CHRGSELn = 1’b0) 0: VDP < VDAT_REF (for CHRGSELn = 1’b1) or VDM < VDAT_REF (for CHRGSELn = 1’b0)
3-0	Reserved

Field	Function
—	

11.3 Universal Serial Bus PHY (USB_PHY)

11.3.1 Overview

The USB 3.0 PHY is designed for USB 3.0 SuperSpeed applications connecting to a USB controller. The USB 3.0 PHY supports the USB 3.0 SuperSpeed (5 Gbps) protocol and data rate and is backward compatible with USB 2.0 high-speed (480 Mbps), full-speed (12 Mbps), and low-speed (1.5 Mbps) protocols and data rates. This block guide provides an introduction to the USB 3.0 PHY and its features.

NOTE

The information within this block guide is Synopsys Proprietary. Used with permission.

11.3.1.1 Block Diagram

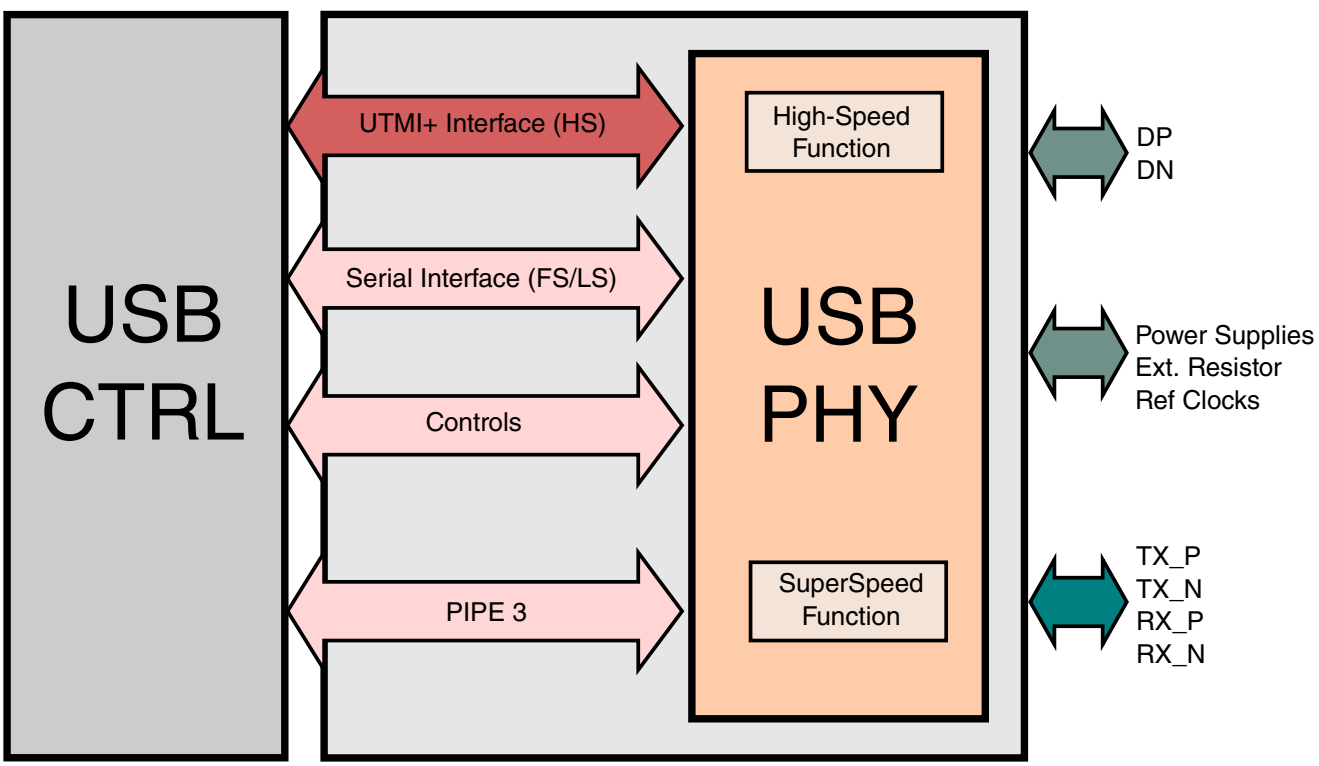


Figure 11-30. USB PHY Block Diagram

11.3.1.2 Features

The USB 3.0 PHY provides the following general features:

- Built-in VBUS threshold comparators;
- Single reference clock input for both USB 3.0 and USB 2.0 operating modes
- Integrated 3.3-V-to-1.8-V regulator in the PHY

The USB 3.0 PHY provides the following USB 3.0 features:

- 5-Gbps SuperSpeed data transmission rate through 3-m USB 3.0 cable
- PIPE 3-compliant SuperSpeed USB 3.0 Transceiver interface
- Integrated PHY includes transmitter, receiver, SSC generation, PLL, digital core, and ESD
- Adaptive Rx equalization
- Low-jitter PLL technology with excellent supply isolation

The USB 3.0 PHY provides the following USB 2.0 features:

- Built-in VBUS threshold comparators;
- 480-Mbps high-speed, 12-Mbps full-speed, and 1.5-Mbps low-speed serial (Host mode only) data transmission rates
- Supports high-speed power modes: suspend, resume, and remote wakeup
- 45-Ω termination, 1.5-kΩ pull-up and 15-kΩ pull-down resistors with support for independence control of the pull-down resistors
- 8/16-bit unidirectional parallel interfaces for HS, FS, and LS (Host mode only) modes of operation in accordance with the UTMI+ specification
- Dual (HS/FS) mode host/device support
- Data recovery from serial data on the USB connector
- SYNC/End-of-Packet (EOP) generation and checking
- Bit stuffing and unstuffing, and bit-stuffing error detection
- Non Return to Zero Invert (NRZI) encoding and decoding
- Bit serialization and deserialization
- VBUS threshold comparators

11.3.1.3 Standards Compliance

The USB 3.0 PHY is fully compliant with all the required features of the following standards:

- Universal Serial Bus 3.0 Specification, Revision 1.0

    - Supports 5.0-Gbps SuperSpeed (SS) through 3-m cable, 480-Mbps high-speed (HS), 12-Mbps fullspeed (FS), and 1.5-Mbps low-speed (LS) serial (Host mode only) data transmission rates, with a high-speed (HS) transmit-to-receive interpacket delay requirement of 38 bit-times as implied by the UTMI specification, version 1.05.
    - Supports beaconing, receiver detection, and electrical idle
    - Four-wire SuperSpeed connection (two Tx and two Rx unidirectional) with built-in Tx and Rx equalization
- PIPE 3-compliant USB 3.0 SuperSpeed interface
- UTMI+ Specification, Revision 1.0 (Level 3)
- Battery Charging Specification, Revision 1.2

11.3.2 Functional Description

11.3.2.1 PIPE Interface

The datapath of the USB 3.0 PHY is through the standard PIPE 3 interface. This interface can connect directly to any USB 3.0 controller. For more information about the PIPE 3 interface signals, refer to PHY Interface For the PCI Express and USB 3.0 Architectures.

11.3.2.2 Termination Tuning

The PHY uses an external resistor to calibrate the termination impedances of the PHY's high-speed inputs and outputs. The resistor is shared between the USB 2.0 high-speed outputs and the SuperSpeed I/O. Each time the PHY is taken out of a reset, a termination calibration is performed. The termination calibration is continuous for USB 2.0 operation except when the USB 2.0 port is suspended. The termination calibration is performed only when coming out of a reset in SuperSpeed operation.

A resistor calibration on the SuperSpeed link cannot be performed while the link is operational. At no point is a request for calibrating the SuperSpeed link using rtune_req required; this feature is provided only for completeness.

11.3.2.3 Battery Charging

The USB 3.0 PHY supports battery charger detection to enable devices to draw increased current from VBUS for charging and/or powering up from dedicated charging ports or charging downstream ports.

11.3.2.4 Dead Battery Provision

The dead battery provision enables a device to draw increased current (up to 100 mA) from VBUS in certain conditions. This function can be used only if all power supplies are stable and at their respective, normal, valid operating levels.

11.3.2.5 Data Contact Detect

Prior to a device performing battery charger detection, DCD is performed to determine if the D+ and D– pins in the device have made contact with those in the host or charger.

11.3.2.6 UART / Autoresume

The USB 3.0 PHY's single-ended receivers are used for UART and Autoresume data reception, while the full-speed drivers are used for data transmission. UART and Autoresume signaling levels are the same as signaling levels for USB 2.0 FS mode.

In this mode of operation, the driver source impedance is not calibrated; however, the driver retains the settings of the last calibration. In addition, the rise/fall time control is disabled, so changes in the state of DP and DN driven by this mode are not expected to meet the rise and fall times as defined in the USB 2.0 specification.

11.3.2.7 Serial Interface

The serial interface supports full-speed and low-speed data transmission rates to and from a USB 1.1 controller.

11.3.2.8 Clocks

The USB 3.0 PHY supports a wide range of input reference clocks from both external and on-chip clock sources. To support both SuperSpeed and high-speed operations, one of the following must be provided:

- A compatible, shared reference clock frequency
- Separate clock sources to support SuperSpeed operation and high-speed operation

11.3.2.9 Fixed RX Equalization

The USB 3.0 femtoPHY's adaptive RX equalization function does not always provide an optimal setting for equalization; the result can be marginal jitter tolerance curves. It is recommended to use fixed equalization instead of adaptive equalization. System-level simulations and hardware testing have proven that fixed equalization is sufficient for both short and long channels. The PHY can be configured for fixed equalization by programming relevant control registers in the USB 3.0 femtoPHY as follows.

1. Write 1'b0 to RX_OVRD_IN_HI.RX_EQ_EN [address 16'h1006: bit 6].
2. Write 1'b1 to RX_OVRD_IN_HI.RX_EQ_EN_OVRD [address 16'h1006: bit 7].
3. Write a fixed value to RX_OVRD_IN_HI.RX_EQ [address 16'h1006: bits 8–10] [equalization setting] (generally 2–4, based on testing in customer environment).
4. Write 1'b1 to RX_OVRD_IN_HI.RX_EQ_OVRD [address 16'h1006: bit 11].

The RX_OVRD_IN_HI is a PHY internal register and accessed through the register in the USB controller. See PHY Register Control (PHY_CRCTL) in the Universal Serial Bus Controller (USB) chapter for details of how to update a USB PHY register. In the above steps, the register RX_OVRD_IN_HI will be written four times. The software must also read the RX_OVRD_IN_HI register first and keep other bits the same while updating specific bits.

11.3.3 External signals

This section describes the USB 3.0 PHY signals that go to pins:

Table 11-30. USB PHY external signals

Signal	I/O	Description
DM0	IO	USB D- Signal. This bidirectional pin carries USB2.0 data to and from the USB 3.0 femtoPHY.
DP0	IO	USB D+ Signal. This bidirectional pin carries USB2.0 data to and from the USB 3.0 femtoPHY.
ID0	IO	USB Mini-Receptacle Identifier and Test Point for DC Points Probes Inside USB 3.0 femtoPHY.
VBUS0	IO	USB 5-V Power Supply Pin. USB power supply pin. A charge pump external to the USB 3.0 femtoPHY must provide power to this pin.
resref	IO	Reference Resistor Connection. Attach a precision resistor-to-ground on the board, with the following specifications: • Resistance: 200 Ohm • Temperature: Coefficient +/-100 ppm/degree C • Tolerance: +/-1%
rxX_m	I	rxX_m is High-Speed Differential Receive Pair. Receive differential pair.
rxX_p	I	rxX_p is High-Speed Differential Receive Pair. Receive differential pair.

Table continues on the next page...

Table 11-30. USB PHY external signals (continued)

Signal	I/O	Description
txX_m	O	txX_m is High-Speed Differential Transmit Pair. Transmit differential pair.
txX_p	O	txX_p is High-Speed Differential Transmit Pair. Transmit differential pair.

11.4 PCI Express (PCIe)

11.4.1 Overview

The PCI Express interface is compatible with the PCI Express™ Base Specification, Revision 4.0, Version 0.7 (available from http://www.pcisig.com). It is beyond the scope of this manual to document the intricacies of the PCI Express protocol. This chapter describes the PCI Express controller of this device and provides a basic description of the PCI Express protocol. The specific emphasis is directed at how the device implements the PCI Express specification. Designers of systems incorporating PCI Express devices should refer to the specification for a thorough description of PCI Express.

NOTE

The information within this block guide is Synopsys Proprietary. Used with permission.

11.4.1.1 Block Diagram

The following figure is the PCIe Application diagram.

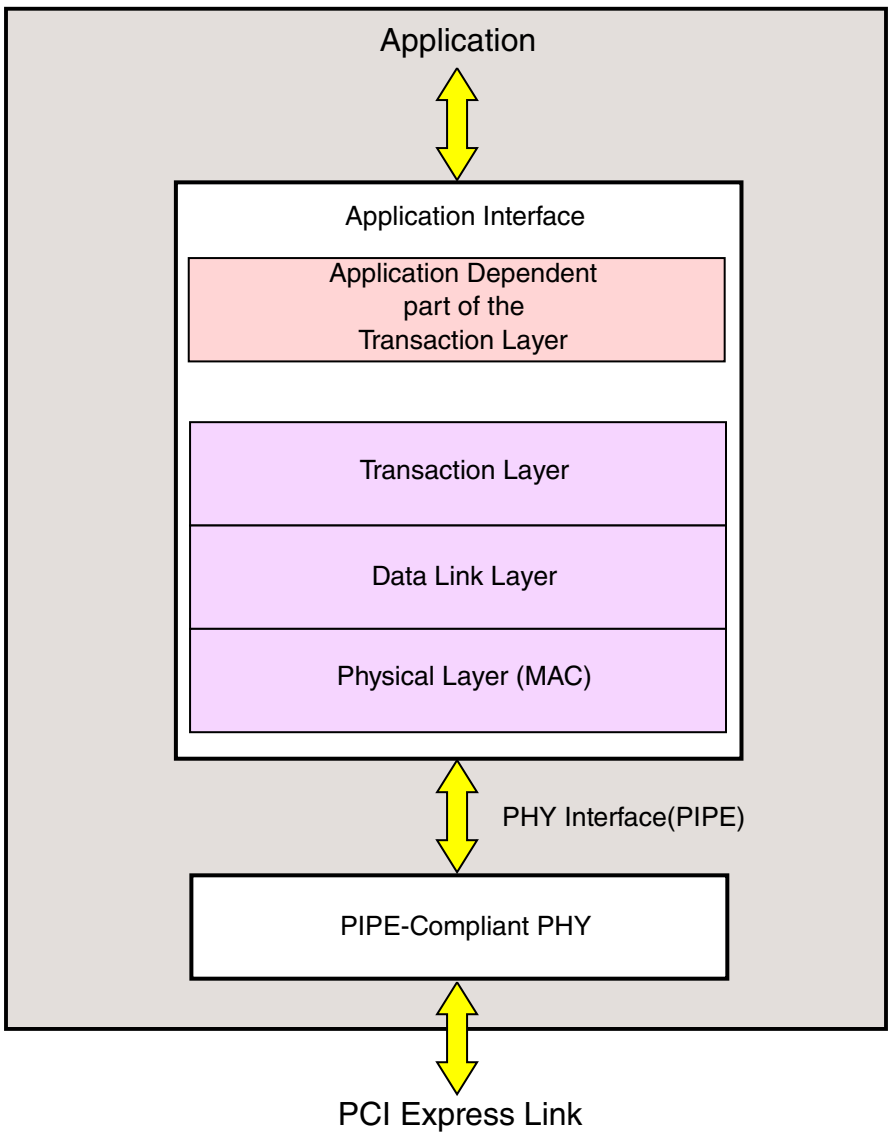


Figure 11-31. PCIe Application

11.4.1.2 Features

The following is a list of features supported by the PCI Express controller:

- Supports Root Complex (RC) and Endpoint (EP) configurations
- Maximum link speed up to Gen3 (8 GT/s)
- x1 link width
- 128-byte maximum payload size (MAX_PAYLOAD_SIZE)
- 32- and 64-bit PCI Express address support
- Supports accesses to all PCI Express memory and I/O address spaces (RC mode requestor only)
- Supports posting of processor-to-PCI Express and PCI Express-to-memory writes

- Embedded DMA
- AXI3 bridge
- Credit-based flow control management handled by PCI Express core
- One Virtual Channel (VC0)
- PCI Express configuration space registers with Type 0 Header in EP mode and Type 1 Header in RC mode
- Internal Address Translation Unit
- Resizable BAR (RBAR) with Expanded RBAR support
- Configurable BAR Filtering, I/O Filtering, Configuration Filtering and Completion Lookup/Timeout
- Supports Completion Timeout Ranges
- Extended Tag Support
- Supports strong and relaxed transaction ordering rules
- Configurable Filtering Rules for Posted, Non-posted, and Completion Traffic
- ECRC Generation and Checking
- Baseline and Advanced Error Reporting (AER) support
- PCI Express Advanced Error Reporting (AER) with Multiple Header Logging
- Supports PCI Express messages and interrupts
- Supports 64-bit MSI interrupts
- Internal MSI-X Generation Module
- MSI and MSI-X with Per-Vector Masking (PVM), Extended message data for MSI
- Latency Tolerance Reporting (LTR)
- ID-Based Ordering (IDO)
- PASID
- L1 Substates (L1SS)
- Separate Refclk with Independent Spread Spectrum Clocking (SRIS)
- Lightweight Notifications (LN)
- Readiness Notifications (RN)
- PCI Express Active State Power Management (ASPM)
- Vital Product Data (VPD)
- Power Gating (UPF) Support
- Advanced Power and Clock Management

The PCIe controller implementation is compatible with the following standards:

- *PCI Express Base Specification, Revision 4.0, Version 0.7*
- PCI Local Bus Specification, Revision 3.0
- PCI Bus Power Management Specification, Revision 1.2
- PCI Express Card Electromechanical Specification, Revision 1.1

11.4.2 Functional Description

The PCI Express controller connects the internal platform to a serial interface. As both an initiator and a target device, the PCI Express interface is capable of high-bandwidth data transfer and is designed to support next generation I/O devices. Upon coming out of reset, the PCI Express interface performs link speed and width negotiation and exchanges flow control credits with its link partner. Once link auto negotiation is successful, the controller is in operation.

Internally, the design contains queues to keep track of inbound and outbound transactions. There is control logic that handles buffer management, bus protocol, transaction spawning and tag generation. In addition, there are memory blocks used to store inbound and outbound data.

The PCI Express controller can be configured to operate as either a PCI Express Root Complex (RC) or Endpoint (EP) device. An RC device connects the host CPU/memory subsystem to I/O devices while an EP device typically denotes a peripheral or I/O device. In RC mode, a PCI Express Type 1 configuration header is used; in EP mode, a PCI Express Type 0 configuration header is used.

A complete PCI Express port solution includes the controller, an analog PHY macro, and internal platform as the application logic to source and sink data.

See the high-level block diagram of the PCI Express controller and analog PHY macro.

As an initiator, the PCI Express controller supports memory read and write operations. In addition, configuration and I/O transactions are supported if the PCI Express controller is in RC mode. As a target interface, the PCI Express controller accepts read and write operations to local memory space. As an EP device, the PCI Express controller accepts configuration transactions to the internal PCI Express configuration registers. Message generation and acceptance are supported in both RC and EP modes. Locked transactions and inbound I/O transactions are not supported.

11.4.2.1 Outbound Transactions

Outbound internal platform transactions to PCI Express are first mapped to a translation window to determine what PCI Express transactions are to be issued. A transaction from the internal platform can become a PCI Express Memory, I/O, Message, or Configuration transaction depending on the window attributes.

A transaction may be broken up into smaller sized transactions depending on the original request size, transaction type, and either the PCI Express Device Control register [PCIE_CAP_MAX_PAYLOAD_SIZE_CS] field for write requests or the PCI Express Device Control register [PCIE_CAP_MAX_READ_REQ_SIZE] field for read requests. The controller performs PCI Express ordering rule checking to determine which transaction is to be sent on the PCI Express link.

In general, transactions are serviced in the order that they are received from the internal platform . Only when there is a stalled condition does the controller apply PCI Express ordering rules to outstanding transactions. For posted write transactions, once all data has been received from the internal platform , the data is forwarded to the PCI Express link and the transaction is considered as done. For non-posted write transactions, the controller waits for the completion packets to return before considering the transaction finished. For non-posted read transactions, the controller waits for all completion packets to return and then forwards all data back to the internal platform before terminating the transaction

NOTE

After reset or when recovering from a link down condition, external transactions should not be attempted until the link has successfully trained. Software can poll the LTSSM state status to check the status of link training before issuing external requests.

In EP mode, after reset or when recovering from a link down condition, the SoC must not generate any outbound memory or I/O transactions until the remote host has configured the Bus Master Enable bit in the PCI Command register.

11.4.2.2 Inbound Transactions

Inbound PCI Express transactions to internal platform are first mapped to a translation window to determine what internal platform transactions are to be issued.

A transaction may be broken up into smaller sized transactions when sending to the internal platform depending on the original request size, byte enables and starting/ending addresses. The controller performs PCI Express ordering rule checking to determine what transaction is to be sent next to the internal platform.

In general, transactions are serviced in the order that they are received from the PCI Express link. Only when there is a stalled condition does the controller apply PCI Express ordering to outstanding transactions. For posted write transactions, once all data

has been received from the PCI Express link, the data is forwarded to the internal platform and the transaction is considered as done. For non-posted read transactions, the controller forwards internal platform data back to the PCI Express link.

NOTE

The controller splits transactions at the crossing of every 128-byte-aligned boundary when sending completion data back to the PCI Express link.

11.4.2.3 Link Establishment

The controller implements the LTSSM function according to the *PCI Express Base Specification, Revision 4.0, Version 0.7. .*

11.4.2.3.1 Transmit TLP Processing

The following section describe the flow of transmit TLPs through the controller.

NOTE

The controller does not check the TLP for errors.

The PCIe controller ensures transmitted TLP Payload sizes are less than or equal to the maximum payload size limit.

11.4.2.3.1.1 Transmit TLP Arbitration

TLPs and DLLPs have equal priorities during transmit arbitration as shown in the figure below. The priority of TLP types and DLLP types are shown in tables below.

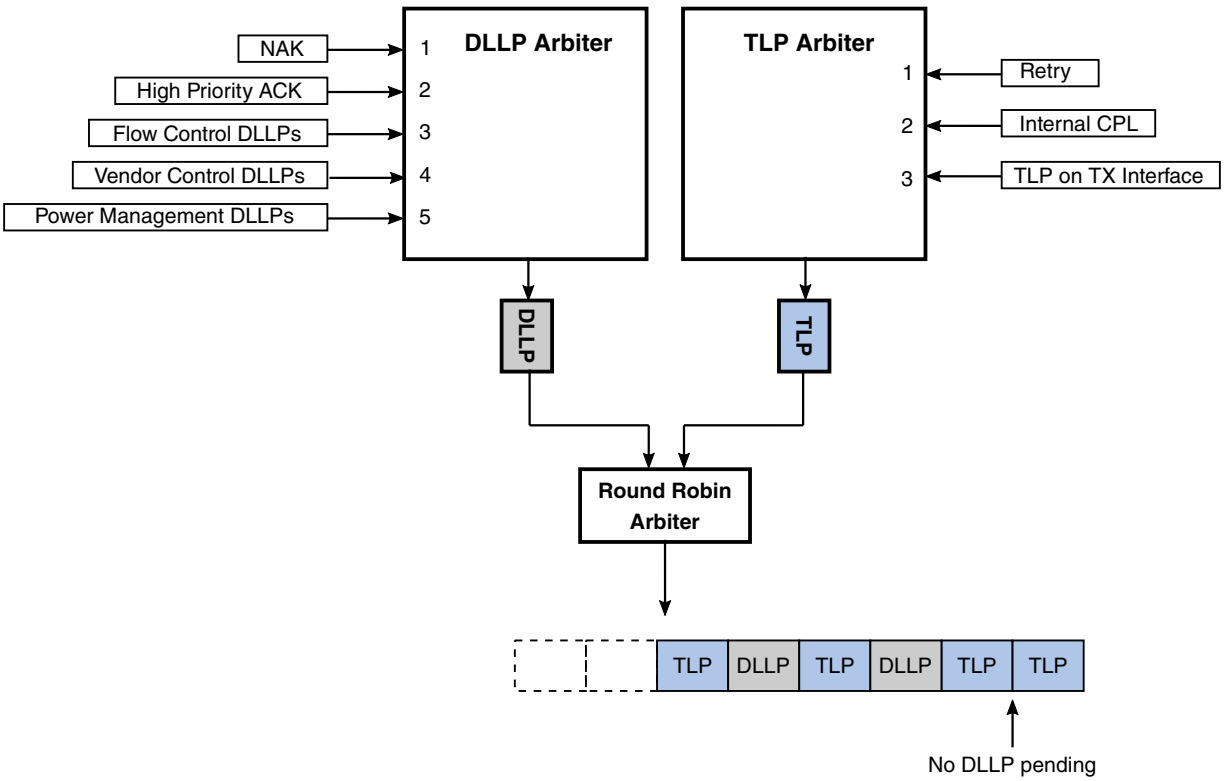


Figure 11-32. Transmit TLP/DLLP Arbitration

Table 11-31. TLP Arbitration Priority (1= Highest, 4= Lowest)

Priority	TLP Type
1	Completion of any TLP currently in progress
2	Retry buffer retransmissions
3	TLPs from the transaction layer (in the following order of priority): • Messages generated by the controller (including by your application through the MSI interface) • Upstream ports: Completions generated by the controller (including memory or I/O mapped application register space) for Type 0 configuration read and write requests, or responses to error conditions (unsupported requests) • Downstream ports: Completions generated by the controller for unsupported requests or completer aborts
4	TLPs on the AXI Master/Slave interface

Table 11-32. DLLP Arbitration Priority (1= Highest, 6 = Lowest)

Priority	DLLP Type

Table continues on the next page...

Table 11-32. DLLP Arbitration Priority (1= Highest, 6 = Lowest) (continued)

1	Completion of any DLLP currently in progress
2	NAK DLLP
3	High Priority ACK DLLP
4	Flow Control DLLP
5	Vendor Specific Message DLLP
6	Power management or any other Low Priority DLLP

11.4.2.3.1.1.1 Effects of Flow Control Credits On Transmit Client Arbitration

The controller checks that enough flow control (FC) credits are available in the remote device for the specific type of transaction (posted, non-posted, completion) before allowing a transmission of a TLP. TLPs that passed the credit check are arbitrated according to the supported arbitration method. Internally generated completions and messages are also gated by the arbitration logic, though at highest priority, and must also pass the FC credit test before they are accepted for transmission.

11.4.2.3.1.2 Transmit Replay

You can modify the time-out value of the replay buffer as shown in the figure below. For a typical system, you do not have to modify the default settings, unless the remote device is executing unexpected replays.

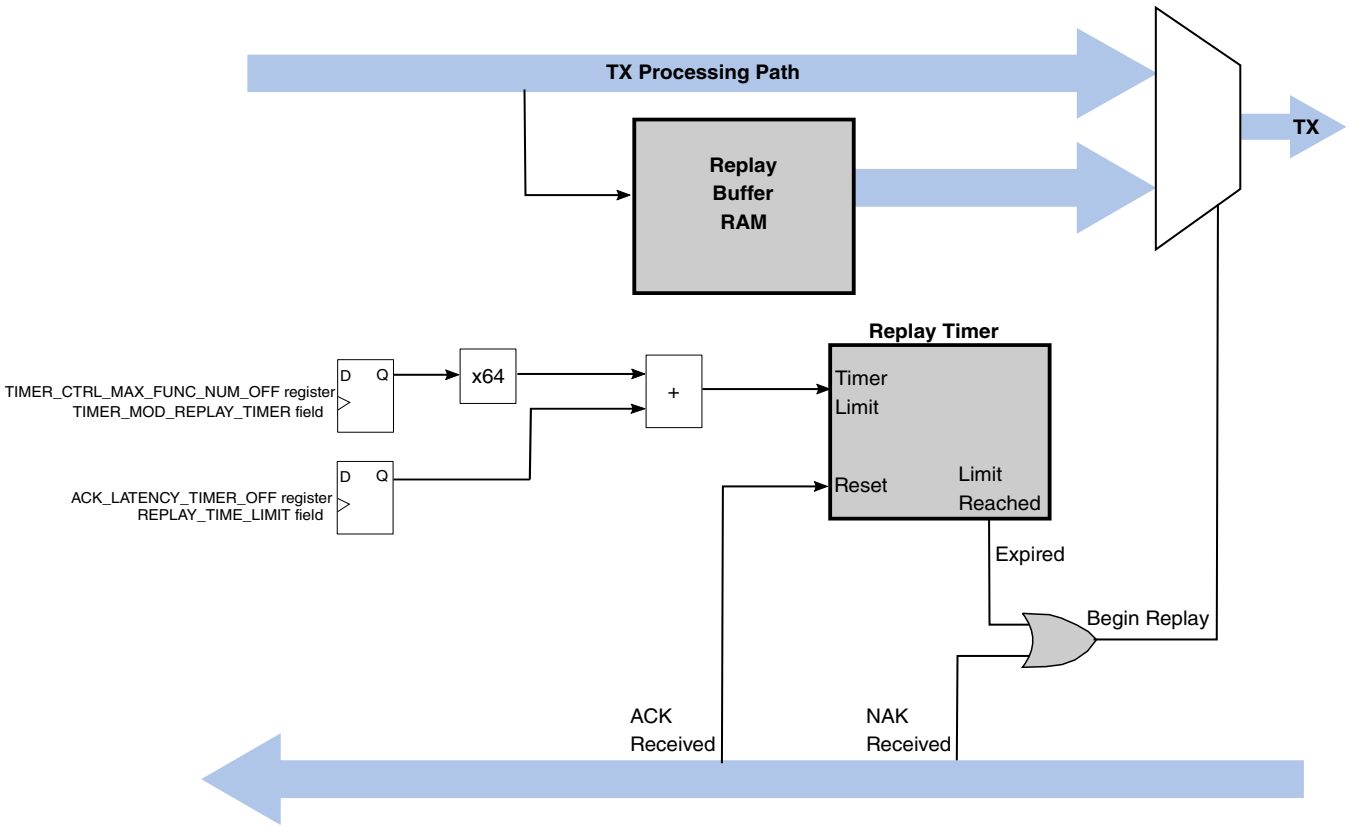


Figure 11-33. Replay Buffer And Time (Representative)

11.4.2.3.2 Receive TLP Processing

This section describes the flow of received TLPs through the controller.

11.4.2.3.2.1 Receive Filtering

The controller contains a filter module that is responsible for the following tasks:

- Determine the status of a received TLP using filtering rules.
- Determine the destination of a received TLP based on the filtering status.
- Indicate the status of the received TLP using outputs.
- Report Errors to AER registers based on filter results. When more than one type of error is detected, Error Pollution, of the *PCI Express Base Specification, Revision 4.0, Version 0.7* is followed.
- The controller filters and routes received TLPs according to a set of rules determined by the TLP type based on the *PCI Express Base Specification, Revision 4.0, Version 0.7* and user-configurable filtering options.

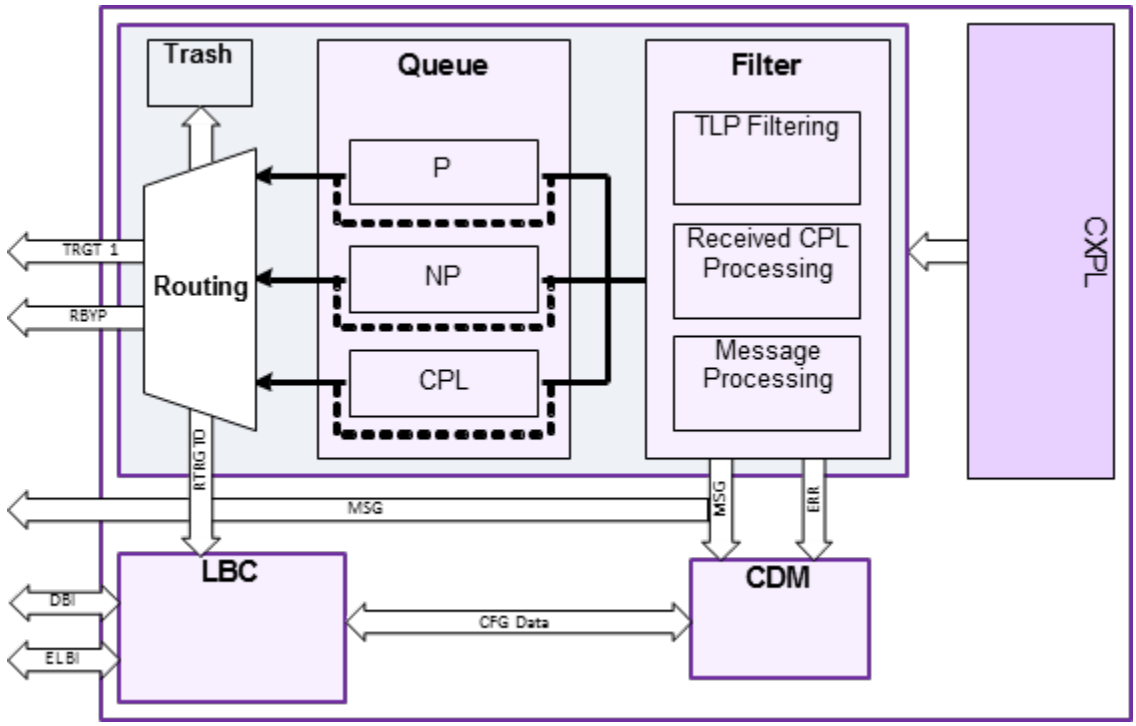


Figure 11-34. RADM Block Diagram

The following general rules apply to all incoming TLPs that are not malformed. For details on what happens to malformed TLPs, see "Error Detection for Received TLPs" By default:

- For a function in device power states D1, D2, and D3hot, the controller only accepts CFG and MSG requests TLPs for that function. All other incoming request types for that function are treated as unsupported requests (UR).
- When the controller detects an error in a received TLP, it normally performs the following:
  - Discards the TLP
  - Generates a completion (for non-posted requests) with the completion status set to CA or UR
  - Sets the status in the PCI-compatible Status register
  - Sets the status in the AER registers
  - Generates an error message (upstream port only)

All error-free MSG requests are decoded internally, signaled on the SII interface and then dropped. When you want to have the decoded message also sent to the application interface, then see “Routing of Received Messages”.

11.4.2.3.2.2 Receive Routing

This section discusses how the RADM routes different TLP types to different receive interfaces depending on TLP type and filter result. TRGT1 refers to the AXI interface. TRGT0 refers to the internal interface used to access the CDM registers or the ELBI in an upstream port.

11.4.2.3.2.2.1 EP Mode Routing Overview

The possible destinations of a posted or non-posted request TLP are TRGT1 interface, TRGT0 interface and Discard. By default:

- CFG requests are routed to TRGT0 and then to CDM through the LBC.
- BAR-matched MEM and I/O requests are routed to TRGT1.
- MSG requests are decoded internally, signaled on the SII interface, and then terminated.

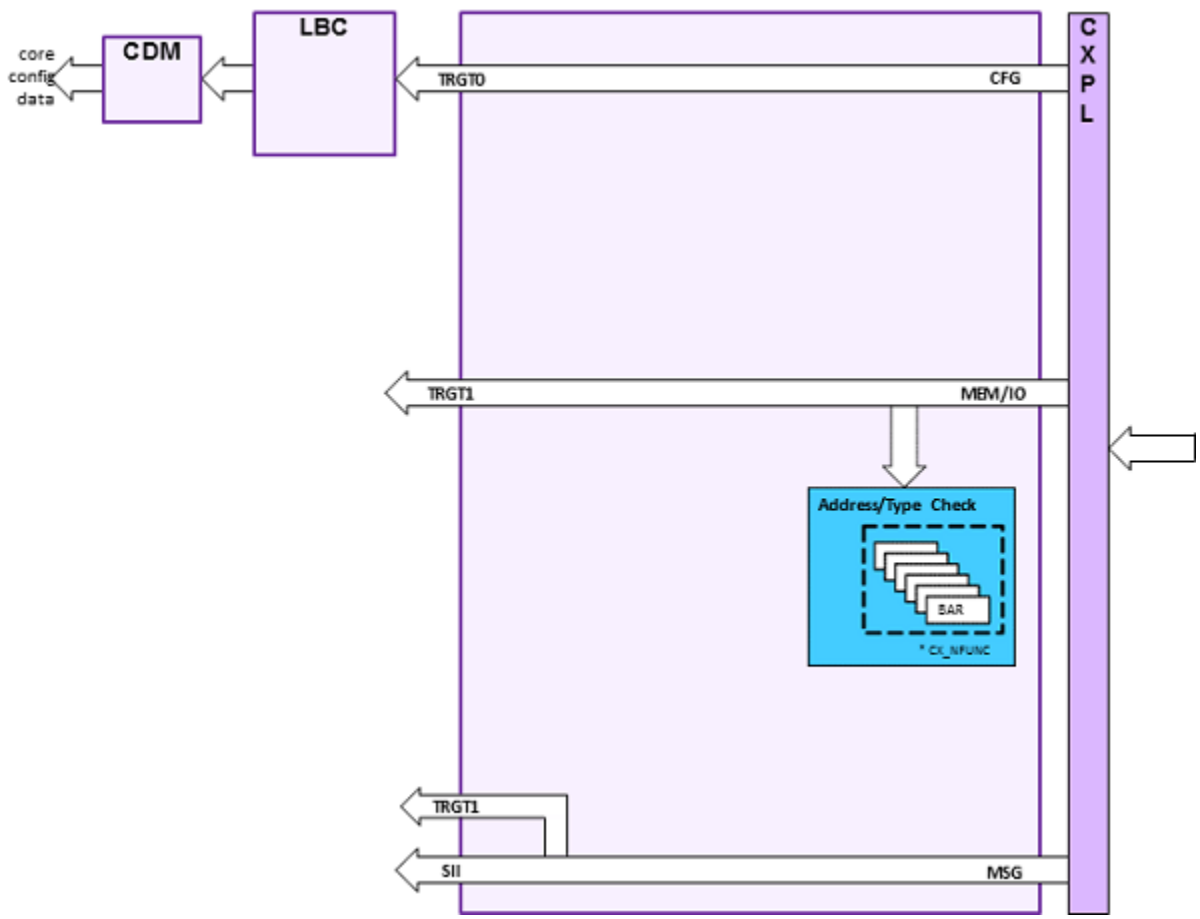


Figure 11-35. EP Mode Routing

11.4.2.3.2.2.2 RC Mode Routing Overview

The possible destinations of a posted or non-posted request are TRGT1 and Discard. By default:

- MEM requests outside of the memory range and prefetchable memory range as determined by the corresponding Base and Limit fields in the Type-1 header are routed to TRGT1.
- MSG requests are decoded internally, signaled on the SII interface, and then terminated.
- An RC does not expect to receive CFG or I/O requests.
- BARs should be disabled and not used.

The possible destinations of a completion TLP are TRGT1 and Discard.

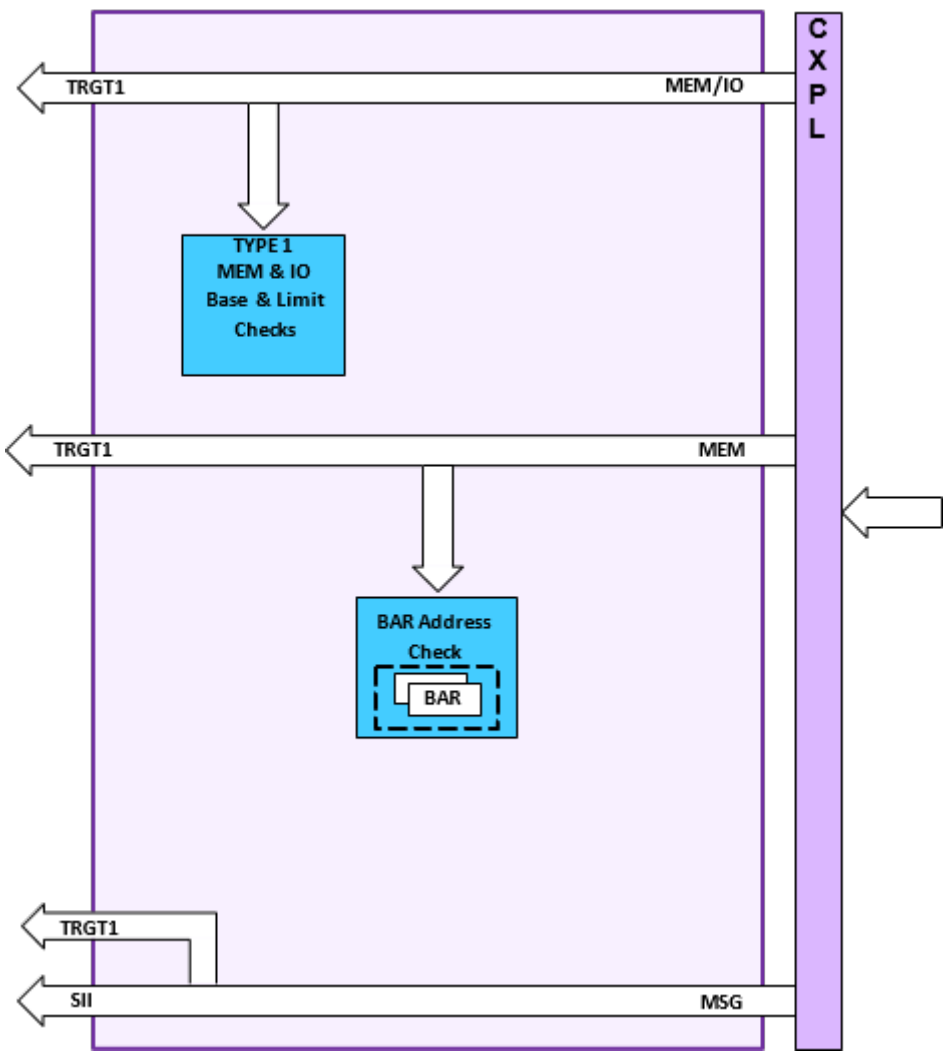


Figure 11-36. Default Request TLP Routing (Assuming no TLPs with CA/CRS/UR Error Status)

11.4.2.3.2.3 Error Handling

The controller supports the Baseline Capabilities, AER capabilities, and Advisory Non-Fatal Error Messaging as specified in section "Error Signaling and Logging" of the *PCI Express Base Specification, Revision 4.0, Version 0.7*. These include Correctable and Uncorrectable (Fatal and Non-Fatal) errors. For an RC port, the reporting of most errors is internal to the root port. No external error notifications are generated. One exception to this (for example) is unsupported request (UR) completion status.

11.4.2.3.2.3.1 Error Detection for Received TLPs

The controller performs all mandatory error detections, the error reporting mechanism based on the PCI Express Base Specification, Revision 4.0, Version 0.7. When the controller detects an error in a received TLP, it normally performs the following:

- Discards the TLP
- Generates a completion (for non-posted requests) with the completion status set to CA or UR
- Sets the status in the PCI-compatible status register
- Sets the status in the AER registers
- Generates an error MSG (upstream port only)

11.4.2.3.2.3.2 AER Multiple Header Logging

The controller supports the AER multiple header logging as specified in Section, "Multiple Error Handling (Advanced Error Reporting Capability)" of the *PCI Express Base Specification, Revision 4.0, Version 0.7*.

11.4.2.3.2.3.3 Physical Functions

By default, each physical function only logs the first header that caused an error. Per physical function, the controller implements a FIFO queue that interfaces with the AER Header Log registers (HDR_LOG_i_OFF). A valid queue entry is never overwritten. When the queue is full and your software does not read the header log registers fast enough, new header log information will be missed.

11.4.2.3.3 Register Module and Data Bus Interface (DBI)

11.4.2.3.3.1 Overview

The Local Bus Controller (LBC) module provides a mechanism for a link partner (in upstream mode) or a local CPU (through the DBI) to access:

- Internal registers (in the CDM)
- External application registers connected externally to the ELBI
- In RC mode:
- There is no PCIe wire access to the CDM registers and ELBI.
- There is full DBI access to the CDM registers and ELBI.
- The controller has 4096 bytes of register PCIe configuration and port logic space per function. The controller also has a common iATU and DMA configuration register space for all functions. The controller register space is fully accessible from the DBI without any restrictions.

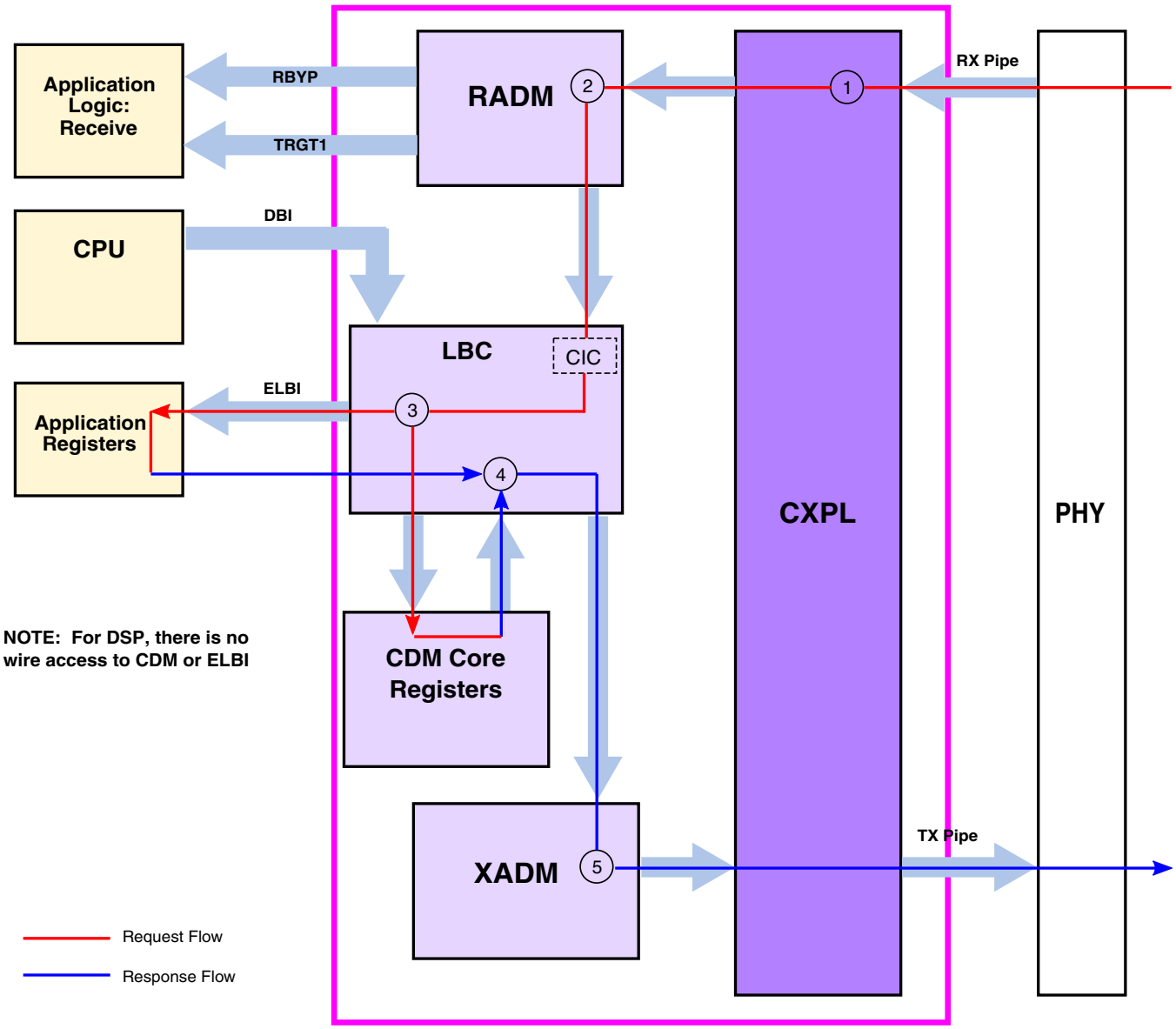
NOTE

Please refer to the Memory Map for the assigned space for the PCIe controller.

Table 11-33. EP Mode Port Region Access

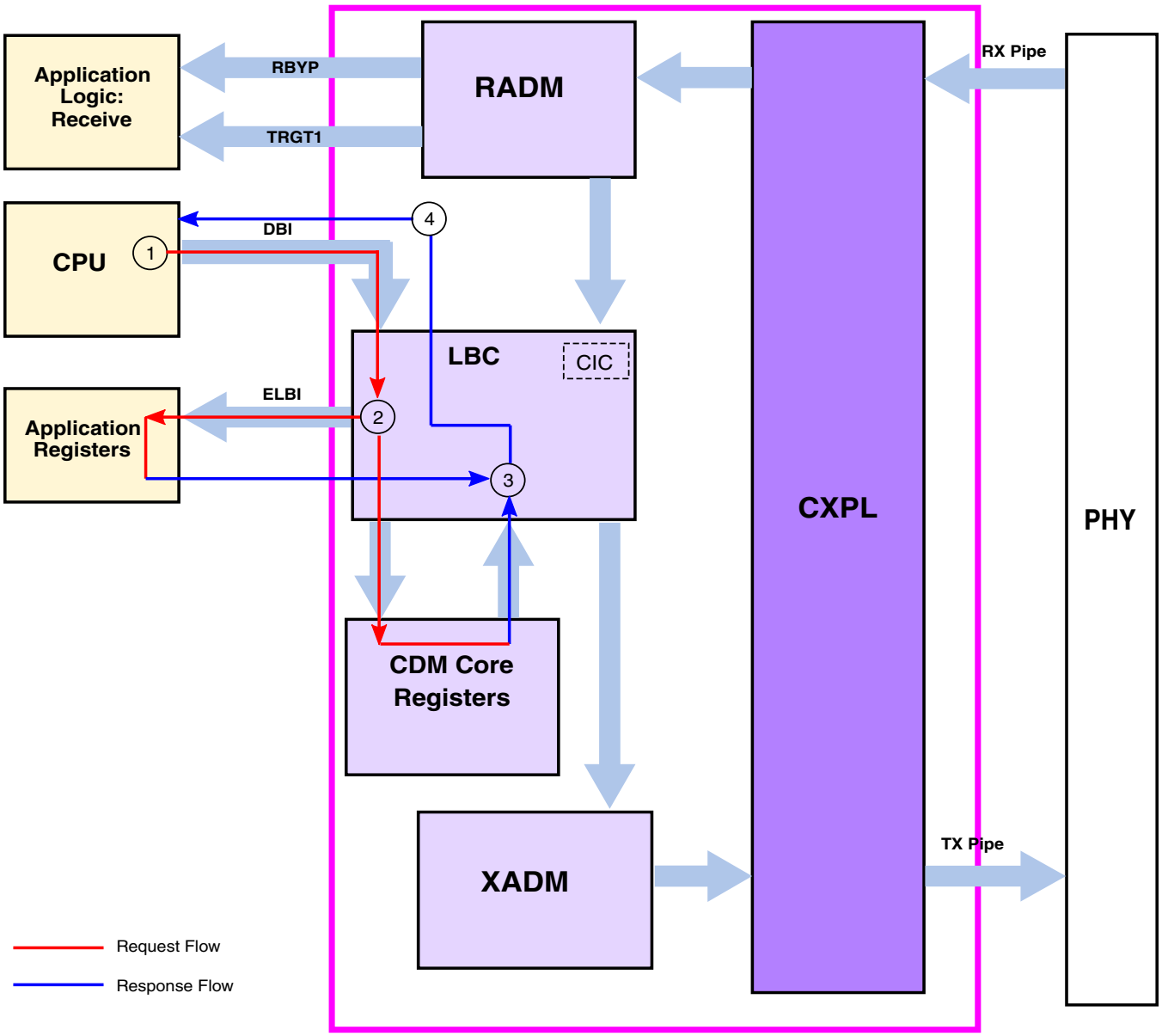
Register Location		CDM					ELBI
Register Type		PCI-SIG		Port Logic			User
		Normal	Shadow	Misc	IATU	DMA	User
Access Details	CDM/ELBI Selector Bit[1] Value	0	0	0	1	1	1
	CS2 Selector Bit Value	0	1	0	1	1	0
DBI/Wire Access Allowed							
DBI		Y	Y	Y	Y	Y	Y
Wire	**CFG Request**	Y	-	Y	-	-	Y
	BAR Matched MEM Request	-	-	Y[2]	Y	Y	Y[-1]

1. Selector bit locations are configuration-dependent.
2. IO request also supported.



1. Incoming request from PCIe remote link partner
2. Request is filtered and routed by RADM and through TRGT0 to LBC
3. LBC forwards the request to external registers (through ELBI) or internal registers in CDM
4. LBC forms completion TLPs with response received from ELBI or internal registers in CDM
5. PCIe controller transmits response completion to remote link partner

Figure 11-37. PCIe Wire Access to CDM Registers or ELBI



1. Request from Local CPU through DBI
2. LBC forwards the request to external registers (through ELBI) or internal registers in CDM
3. LBC forms completion TLPs with response received from ELBI or internal registers in CDM
4. PCIe controller presents response to Local CPU through DBI

Figure 11-38. DBI Access to CDM Registers or ELBI

11.4.2.3.3.2 CDM Register Space Layout

11.4.2.3.3.2.1 Overview

The controller has 4096(0x1000) bytes of register PCIe configuration and port logic space per function. The controller also has a common iATU and DMA configuration register space for all functions. The controller register space is fully accessible from the DBI without any restrictions. Please see the Memory Map for the register base address assignment.

Table 11-34. Port Region Access

Register Location		CDM					ELBI
Register Type		PCI-SIG		Port Logic			User
		Normal	Shadow	Misc	IATU	DMA	User
Access Details	CDM/ELBI Selector Bit[1] Value	0	0	0	1	1	1
	CS2 Selector Bit Value	0	1	0	1	1	0
DBI/Wire Access Allowed							
DBI		Y	Y	Y	Y	Y	Y
Wire	**CFG Request**	Y	-	Y	-	-	Y
	BAR Matched MEM Request	-	-	Y[2]	Y	Y	Y[-1]

1. Selector bit locations are configuration-dependent.
2. IO request also supported.

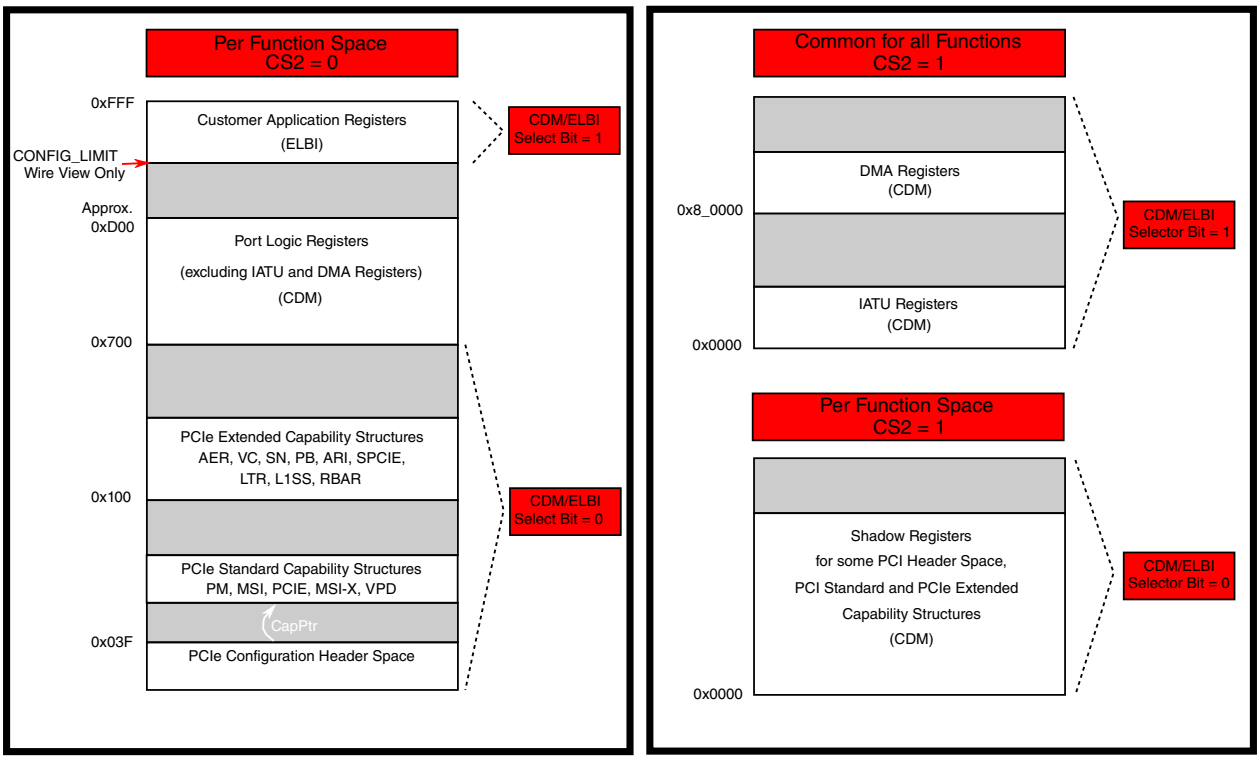


Figure 11-39. Controller Configuration Space Layout (EP Mode)

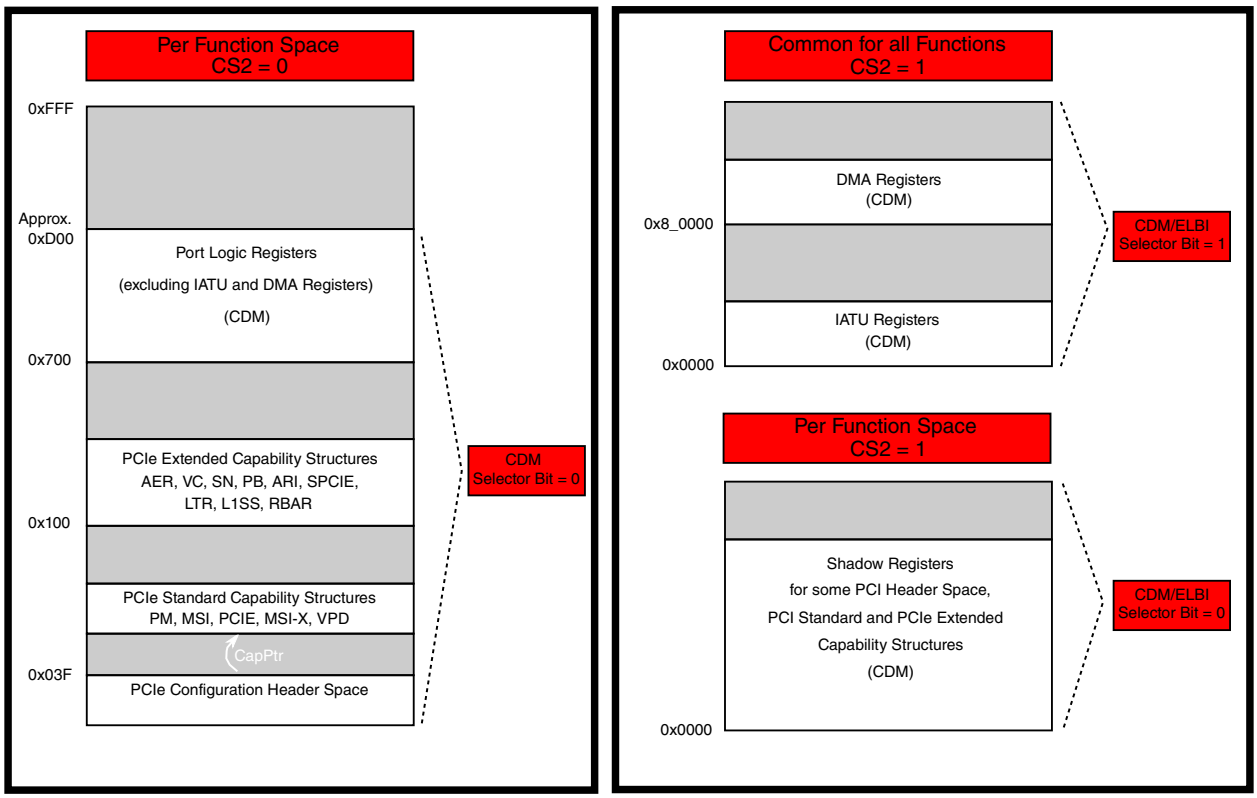


Figure 11-40. Controller Configuration Space Layout: (RC Mode)

Wire Access (Upstream Ports Only):

- Any CFG > CONFIG_LIMIT goes to TRGT1 or ELBI
- Any MEM/IO captured by a BAR whose interface target is TRGT0 goes to ELBI
- Any MEM captured by BAR*n* (*n* =UNROLL_BAR_NUM) when
- ENABLE_MEM_MAP_UNROLL_{DMA|ATU}_REG=1, whose interface target is TRGT0, and whose address falls into the range defined in "Memory Mapping iATU and DMA Port Logic Registers" goes to the IATU/DMA registers.

DBI Access (Upstream and Downstream Ports):

- No concept of MEM/IO/CFG
- Uses the specified address bit to route the access to ELBI or CDM
- No restrictions regarding what spaces can be accessed or not

11.4.2.3.3.2.2 Core Configuration Space Registers

- CDM Registers
  - PCI Configuration Header and Capability Registers:
    - The *PCI configuration header and capability registers* in the previous diagrams are PCIe controller configuration registers specified by the *PCI Express Base Specification, Revision 4.0, Version 0.7*. Access from the PCIe

wire is possible with CFG requests (upstream ports). These registers are fully accessible from the DBI without any restrictions.

- Port Logic Registers:
  - The port logic (PL) registers are configuration registers which are not specified by the *PCI Express Base Specification, Revision 4.0, Version 0.7*. Access from the PCIe wire is normally with CFG requests. These registers are fully accessible from the DBI without any restrictions.
- iATU and DMA Port Logic Registers:
  - This memory space is dedicated for iATU and DMA configuration registers. Access from the PCIe wire is with specific BAR matched MEM requests. These registers are fully accessible from the DBI without any restrictions.

11.4.2.3.3.2.3 PCIe Wire Access (EP Mode)

By default all requests to the standard PCI configuration space or to the port logic registers (except iATU and DMA registers) are routed to TRGT0 and then to the CDM through the LBC. To access the iATU/DMA registers, you must use MEM requests that target the BAR you have setup to capture these addresses. For more details, "Memory Mapping iATU and DMA Port Logic Registers."

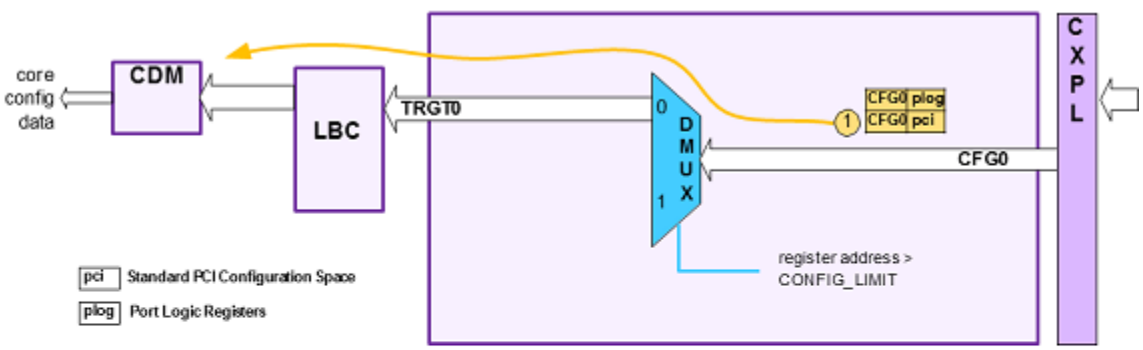


Figure 11-41. Request TLP Routing: Typical Use Model

11.4.2.3.3.3 Data Bus Interface (DBI)

11.4.2.3.3.3.1 Overview

The DBI can access all 4096 bytes (1024 DWORDs) of the PCI Express configuration space per function. DBI can also access iATU and DMA configuration space. This address space is fully accessible from the DBI without any restrictions.

11.4.2.3.3.4 Features and Limitations

The following limitations exist for the LBC.

- CDM Access Restrictions:
  - Maximum payload Length =1 DWORD.
  - PCIe MEM requests with Length > 1 are processed as completer abort.
  - PCIe MEM requests with Length =0 are processed as completer abort.
  - PCIe IO or CFG requests with Length !=1 are processed as malformed TLPs.
- ELBI Access Restrictions:
  - CS2 bit must be 1’b0.
  - PCIe MEM requests with Length =0 are processed as completer abort.
  - PCIe IO or CFG requests with Length !=1 are processed as malformed TLPs.
  - Does not support "Atomic Operations (AtomicOps)"
  - Does not support TPH. Core does not forward the TPH bits to the ELBI.
  - Data (Read)
- DBI Data/Address Port Widths (with no AXI bridge):
  - Data: 32 bit
  - Address: 32 bit

11.4.2.3.3.4.1 Additional AXI DBI Limitations

The following limitations exist when you configure the controller to use the AXI bridge module.

- The DBI address must be a DWORD address.
- AXI transfer widths of 8, 16, and 32 bits are supported. The AXI dedicated/shared DBI interface returns an error response to requests with width greater than 32 bits. The type of error response is programmable through AMBA_ERROR_RESPONSE_DEFAULT_OFF register1.
- The CDM only supports 32-bit data access. The AXI dedicated/shared DBI interface returns an error response to multi beat requests. The type of error response is programmable through AMBA_ERROR_RESPONSE_DEFAULT_OFF register.

A zero-byte write over DBI is processed as a read and should never be issued. The AXI dedicated/shared DBI slave drops a zero-byte write before it reaches the internal (native controller) DBI and returns an error response. The type of error response is programmable through AMBA_ERROR_RESPONSE_DEFAULT_OFF register.

The type of error response returned by the AXI dedicated/shared DBI slave is the same as that programmed for UR completions in the AMBA_ERROR_RESPONSE_DEFAULT_OFF register.

11.4.2.3.4 Flow Control

The flow control mechanism is divided into two phases: initialization and update. The controller automatically performs both of these phases with minimal support required from your application. By default the RADM is responsible for returning flow control credits when it reads data out of the receive queues.

11.4.2.3.4.1 Initial Credit Advertisement by the Controller

To prevent overflow of the link partner's receiver buffers and for order-compliance purposes, the transmitter cannot send transactions unless it has enough flow control (FC) credits to send. Each device maintains an FC credit pool. The FC information is conveyed between the two link partners by DLLPs during link training (initial credit advertisement). The transaction layer performs the FC accounting functions. One FC unit is four DWs (16-bytes) of data.

Table 11-35. Initial Credit Advertisement By The Controller

Credit Type	Initial Credit Advertisement
PH (Memory Write, Message Write)	12
PD (Memory Write, Message Write)	25 (or 400 bytes)[1]
NPH (Memory Read, IO Read, Cfg Read, Cfg Write)	12
NPD (IO Write, Cfg Write)	3 (or 48 bytes)
CPLH (Memory Read Completion, IO R/W Completion, Cfg R/W Completion)	Infinite
CPLD (Memory Read Completion, IO Read Completion, Cfg Read Completion)	Infinite

1. When the outstanding inbound posted write data transfer size exceeds 400 Bytes, the number of inbound posted write transactions that the i.MX 8M Plus PCI Express Controller can support is up to 12 headers or 400 bytes of data payload, whichever is consumed first. Refer to application note, AN13164, iMX8MP PCIe Bandwidth Analysis for more detailed information.

NOTE

When the outstanding inbound write data transfer size exceeds 400Bytes, all inbound MWr (Memory Write) transactions to the

i.MX 8M Plus PCIe Controller support up to a maximum of 12 PCIe write packet headers and up to 25 (400 Byte) write packet data payload.

11.4.2.3.4.2 Calculation of Initial Credits and Receive Buffer Sizes For The Link Partner

The default buffer sizes and credits for each TLP type are automatically calculated from the number of lanes, the controller datapath width, the maximum PCIe payload , flow control update latencies, internal delays, and the PHY latency.

You can determine the transmit posted, non-posted, and completion credits that are advertised by the receiver at the other end of the link by reading the following port logic registers:

- Transmit Posted FC Credit Status (TX_P_FC_CREDIT_STATUS)
- Transmit Non-Posted FC Credit Status (TX_NP_FC_CREDIT_STATUS)
- Transmit Completion FC Credit Status (TX_CPL_FC_CREDIT_STATUS)

NOTE

You can override the default flow control update latency timer value using the FC Latency Timer Override Value field (TIMER_MOD_FLOW_CONTROL) in the Queue Status register (Q_STATUS).

11.4.2.3.5 Completion Timeout Ranges

Timeout ranges are supported as defined in the *PCI Express Base Specification, Revision 4.0, Version 0.7*. The controller supports all ranges. The Device Control 2 Register has a default equal to the default in the specification: 0000b Default range: 50 us to 50 ms. When the default is used, then the timeout is in Range B: 0101b: 16 ms to 55 ms. This range was chosen for the default because the *PCI Express Base Specification, Revision 4.0, Version 0.7* states, "It is strongly recommended that the completion Timeout mechanism not expire in less than 10 ms". The table below illustrates the specification values versus the PCI Express controller values for the ranges.

NOTE

As the PCIe specification states, This mechanism is intended to be activated only when there is no reasonable expectation that the completion is returned, and should never occur under normal operating conditions.

Table 11-36. Comparison of PCIe Specification and PCIe Core Completion Timeout Ranges

Range	Encoding	Spec Minimum	Spec Maximum	PCIe controller Minimum	PCIe controller Maximum
Default	0000b	50 us	50 ms	28 ms	45 ms
A	0001b	50 us	100 us	65 us	99 ms
A	0010b	1 ms	10 ms	4.1 ms	6.2 ms
B	0101b	16 ms	55 ms	28 ms	44 ms
B	0110b	65 ms	210 ms	86 ms	131 ms
C	1001b	260 ms	900 ms	260 ms	390 ms
C	1010b	1s	3.5 s	1.8 s	2.8 s
D	1101b	4s	13 s	5.4 s	8.2 s
D	1110b	17s	64 s	38 s	58 s

11.4.2.4 DMA Controller

This section describes the embedded multichannel DMA controller.

11.4.2.4.1 DMA Overview

The RC system CPU, or the EP application CPU, can offload the transferring of large blocks of data to the embedded DMA controller, leaving the CPU free to perform other tasks. It can simultaneously perform the following types of memory transactions (as shown in the figure 'System Level View of DMA'):

DMA write: Transfer (copy) of a block of data from local (application) memory to remote (link partner) memory.

DMA read: Transfer (copy) of a block of data from remote (link partner) memory to local (application) memory.

Therefore the DMA supports full duplex operation, processing read and write transfers at the same time, and in parallel with normal (non-DMA) traffic. Upon completion of a DMA transfer or an error, the DMA optionally interrupts the local CPU or sends an interrupt MWr (IMWr) to the remote CPU.

In linked list mode, the DMA fetches the transfer control information (called channel context) for each transfer (block), from a list of DMA elements that you have constructed in local memory.

11.4.2.4.2 DMA Architecture

This section describes the architecture of the DMA.

11.4.2.4.2.1 Architecture Overview

A DMA write and read channel operate independently to maximize the performance of the DMA read and write data transfers over the PCIe link. DMA with multiple read channels uses a weighted round robin (WRR) arbitration scheme to select the next read channel to be serviced. The same applies for multiple write channels.

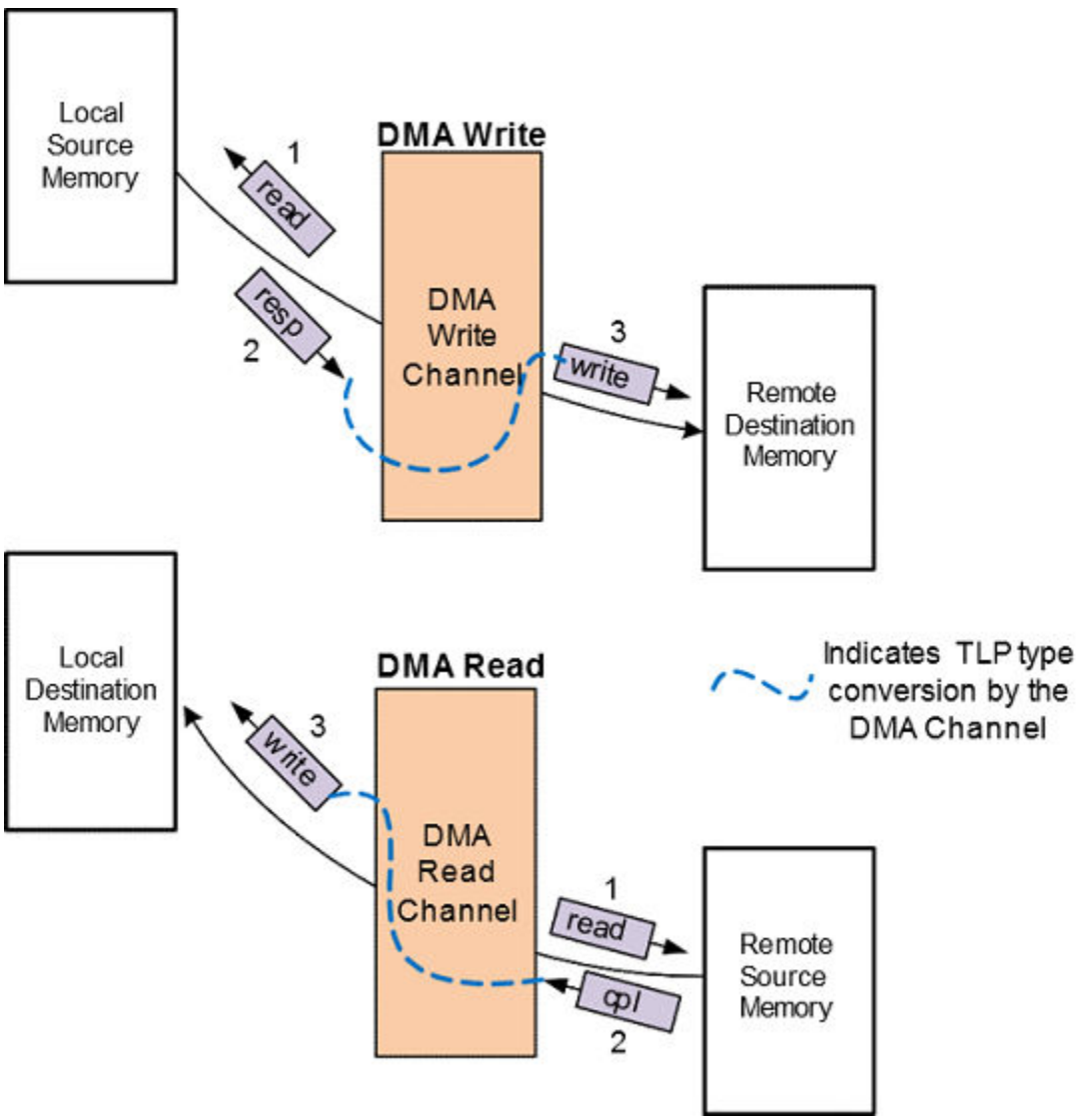


Figure 11-42. System Level View of DMA

11.4.2.4.2.1.1 DMA Read Transfer

The DMA injects multiple MRd requests of size less than or equal to the minimum of {Max Read Request Size, Max Payload Size} into the outbound request path, directed toward the remote link partner. The DMA converts the read responses into write requests, that it then transmits to the local application. When the DMA data transfer is complete, the CPU is notified. For a read transfer, the SAR is the address of the remote memory, and the DAR is the address of the local memory.

11.4.2.4.2.1.2 DMA Write Transfer

The DMA injects multiple MRd requests of size less than or equal to Max_Payload_Size into the inbound request path, directed toward the local application. The DMA converts the read responses into MWr TLPs, that it then transmits to the remote link partner. When the DMA data transfer is complete, the CPU is notified. For a write transfer, the SAR is the address of the local memory, and the DAR is the address of the remote memory.

11.4.2.4.2.2 Interrupts and Error Handling

11.4.2.4.2.2.1 Interrupt Handling

The DMA generates two interrupts per channel:

Done: The DMA successfully completes the transfer.

Abort: The DMA fails to complete the transfer, or an error occurs during the transfer.

The interrupt handling mechanism is different for linked list (LL) mode (than non LL mode), and there are also some differences between the read and write channels.

11.4.2.4.2.2.2 Non Linked List Mode

You enable the local and remote interrupts through the local and remote interrupt enable (LIE and RIE) bits:

DMA_CH_CONTROL1_OFF_WRCH_0.lie and
DMA_CH_CONTROL1_OFF_WRCH_0.rie.

In the write channel, there is only one error condition that results in an abort interrupt. For more details, see "Linked List Mode". You mask, clear, and read the status of each of the two interrupts (done and abort) through the DMA interrupt registers as indicated in figure 'Write Interrupt Generation - Non Linked List Mode - Shown For Write Channel #0'.

In the read channel, there are five error conditions that results in an abort interrupt. For more details, see "Linked List Mode". You mask and clear each of the two interrupts (done and abort) through the DMA interrupt registers as indicated in figure 'Read Interrupt Generation - Non Linked List Mode - Shown For Read Channel #0'. However, you can read the status of each of the five abort errors (that contribute to the abort interrupt) through DMA_READ_ERR_STATUS_LOW_OFF and DMA_READ_ERR_STATUS_HIGH_OFF.

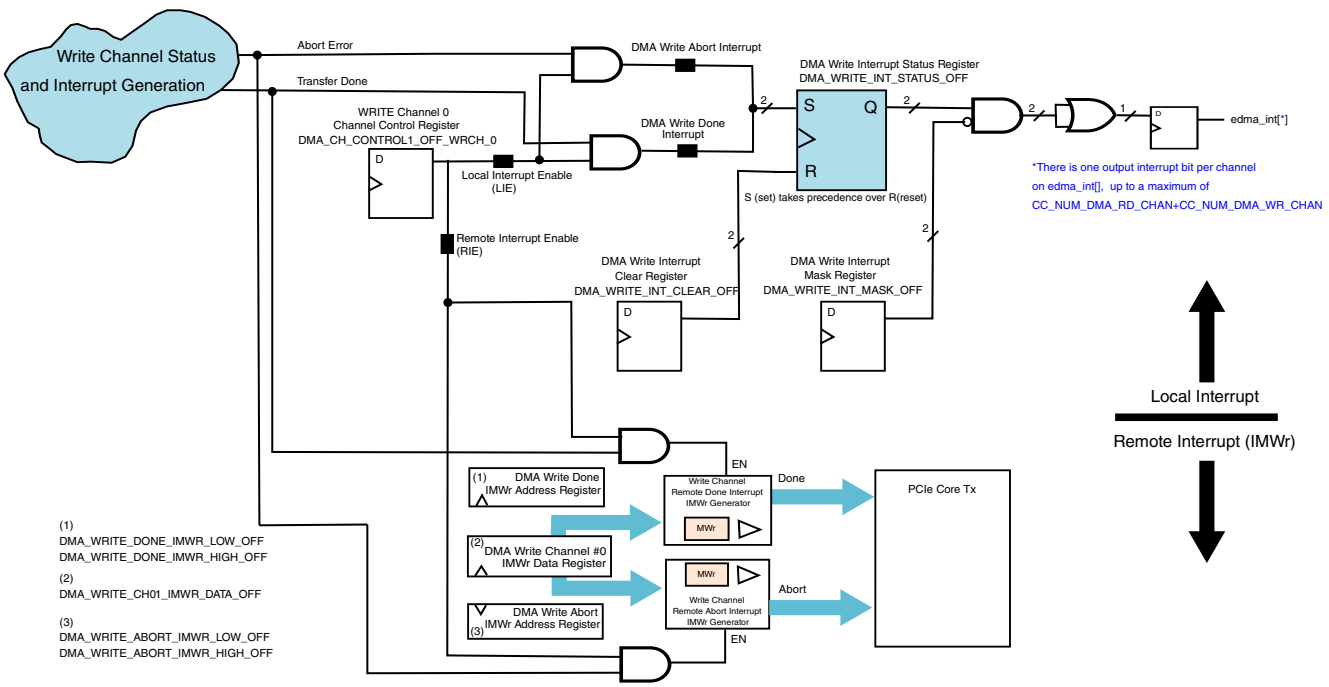


Figure 11-43. Write Interrupt Generation - Non Linked List Mode - Shown For Write Channel #0

If you want a remote interrupt and not a local interrupt then:

- Set LIE and RIE
- Mask the local interrupt using the mask register

This will allow you to poll the status register to distinguish a DONE from an ABORT

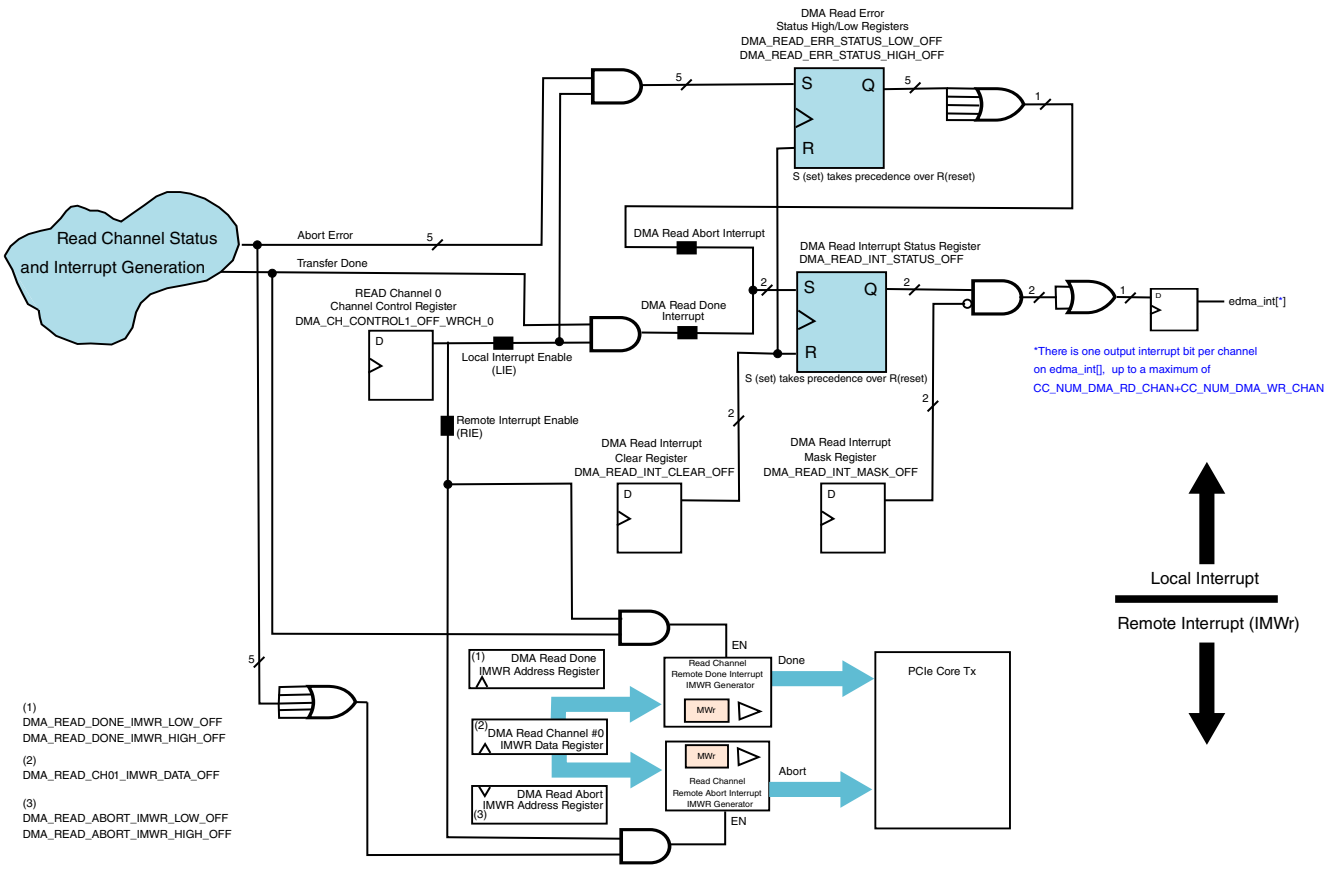


Figure 11-44. Read Interrupt Generation - Non Linked List Mode - Shown For Read Channel #0

11.4.2.4.2.2.3 Linked List Mode

The LIE and RIE bits in the LL element enable the channel done interrupts (local and remote). The LLLAIE and LLRAIE bits of the DMA_WRITE_LINKED_LIST_ERR_EN_OFF/ DMA_READ_LINKED_LIST_ERR_EN_OFF registers enable the channel abort interrupts (local and remote). In the write channel, there are two error conditions that results in an abort interrupt. For more details, see "Linked List Mode". You mask and clear each of the two interrupts (done and abort) through the DMA interrupt registers as indicated in the figure below. You can read the status of each of the two abort errors (that contribute to the abort interrupt) through the DMA_WRITE_ERR_STATUS_OFF register.

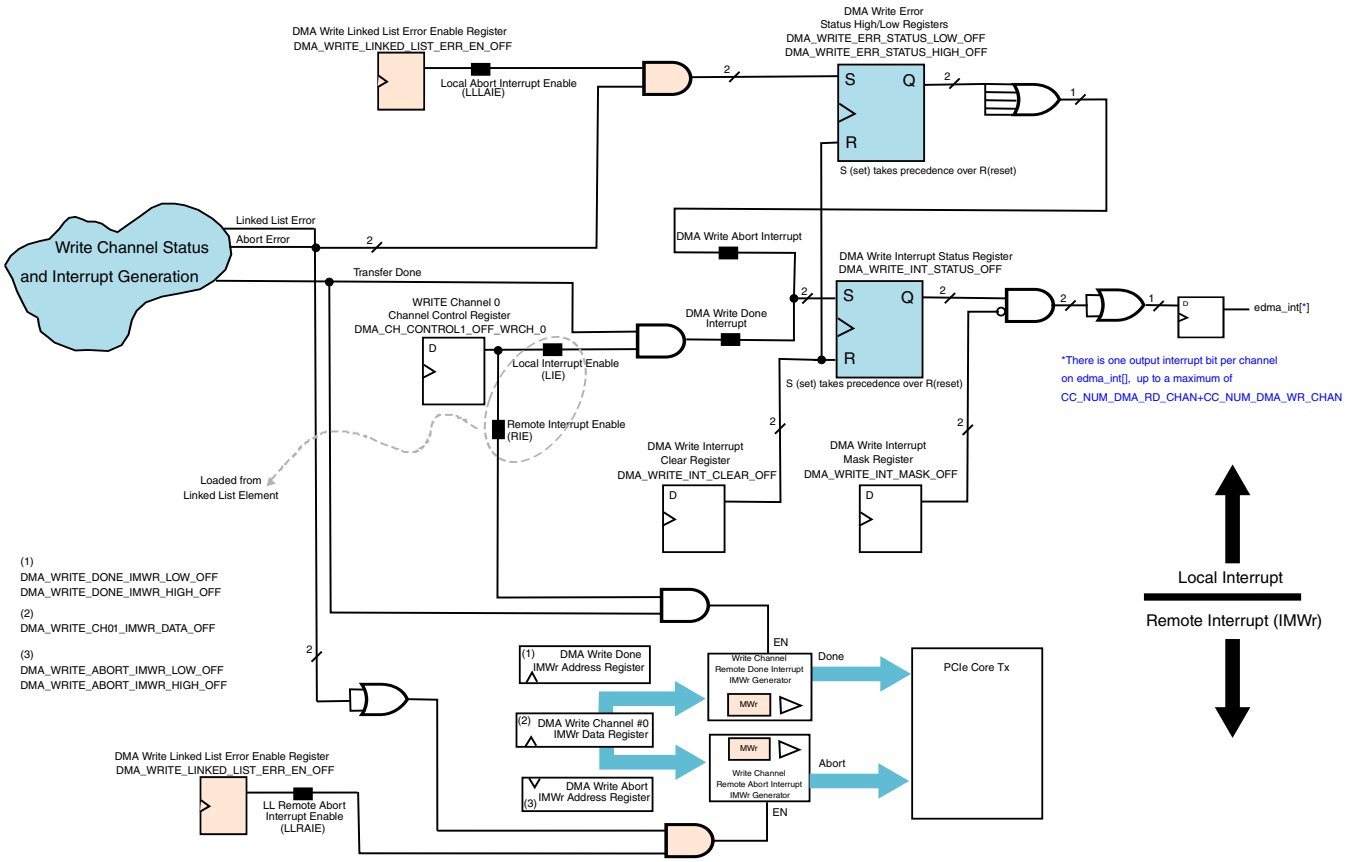


Figure 11-45. Write Interrupt Generation - Linked List Mode - Shown For Write Channel #0

In the read channel, there are six error conditions that results in an abort interrupt. For more details, see "Linked List Mode". You mask and clear each of the two interrupts (done and abort) through the DMA interrupt registers as indicated in the figure below. You can read the status of each of the six abort errors (that contribute to the abort interrupt) through the DMA_READ_ERR_STATUS_LOW_OFF and DMA_READ_ERR_STATUS_HIGH_OFF registers.

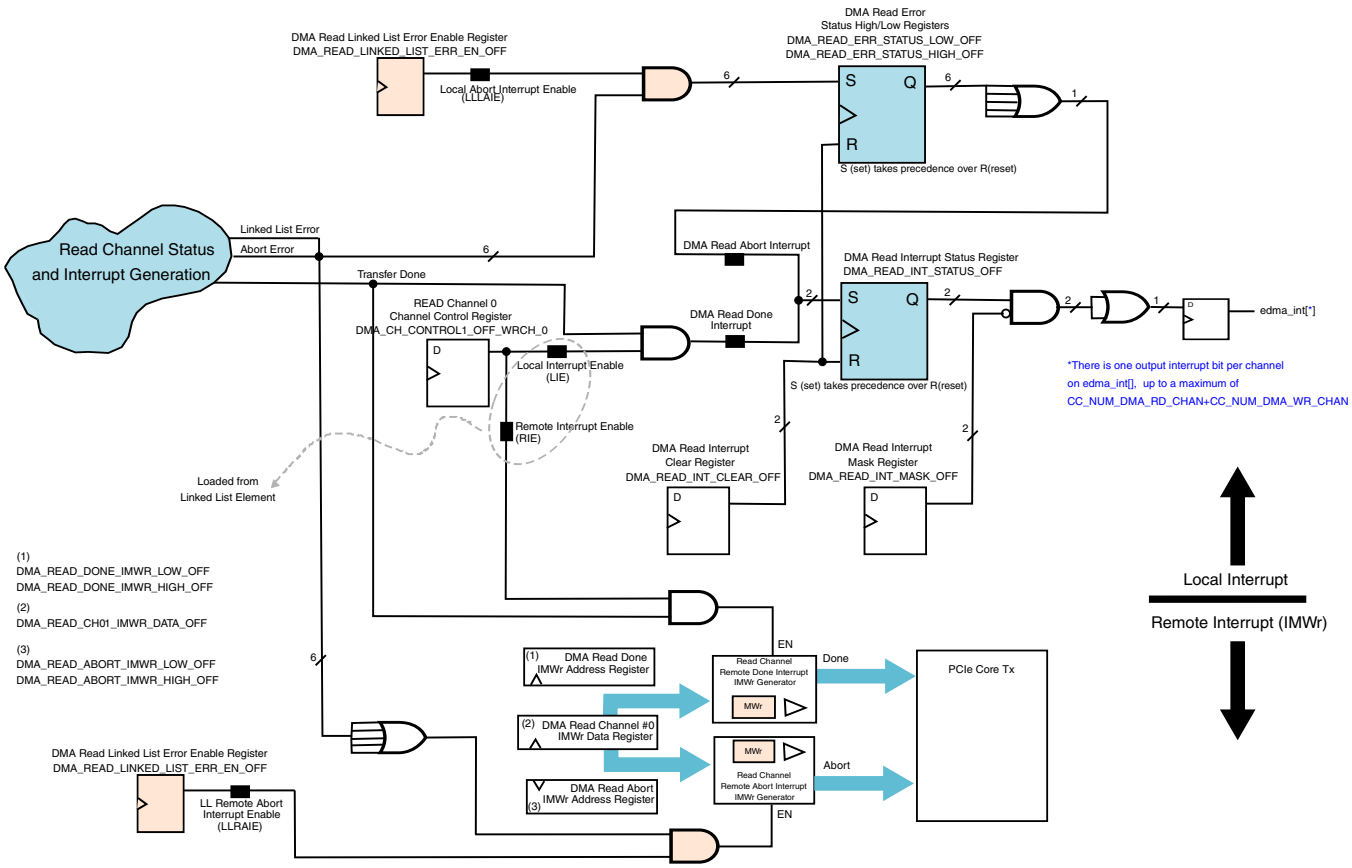


Figure 11-46. Read Interrupt Generation - Linked List Mode - Shown For Read Channel #0

NOTE

In non-linked list mode, LIE acts as a global switch. However when in linked list mode, LIE is just local to the current linked list element and the global switch is LLLAIE.

NOTE

If the DMA driver is running on the host and the interrupt service routine is reading local interrupts to determine if the transfer is successful, then you must set LIE and RIE in the same element and you should mask or ignore the local interrupt pin. Setting RIE and LIE in element A followed by RIE (only) in element B is not a verified usage scenario.

11.4.2.4.2.3 Linked List Mode

The DMA provides a linked list (LL) mode to efficiently move data from source to destination with minimal intervention from the local CPU. This mode provides an alternative to programming the DMA multiple times to transfer multiple blocks of data.

The programming information (address, size, and so on) for each block of memory is pre-programmed by your software into a LL element (also known as a descriptor) in local memory. Each element (called a data element) in the LL structure (called a transfer list) can transfer up to 4 GB of data. You enable LL operation for a channel, by setting the LLE field of the DMA_CH_CONTROL1_OFF_WRCH_0 (Replace WRCH with RDCH for a read channel) register to 1. You can enable LL mode independently for each channel. When you enable LL for more than one channel, then you must have a separate LL structure in local memory for each channel. Your application must produce the LL element structure in local memory as shown in the figure below. Normally, all of the elements are contiguous (one after the other) in memory, and each element has six DWORDs containing the information about the block of data to be transferred. You program the channel context registers (DMA_LLP_LOW_REG_WRCH_0 and DMA_LLP_HIGH_OFF_WRCH_0) with the location of where you have placed the LL element structure in local memory.

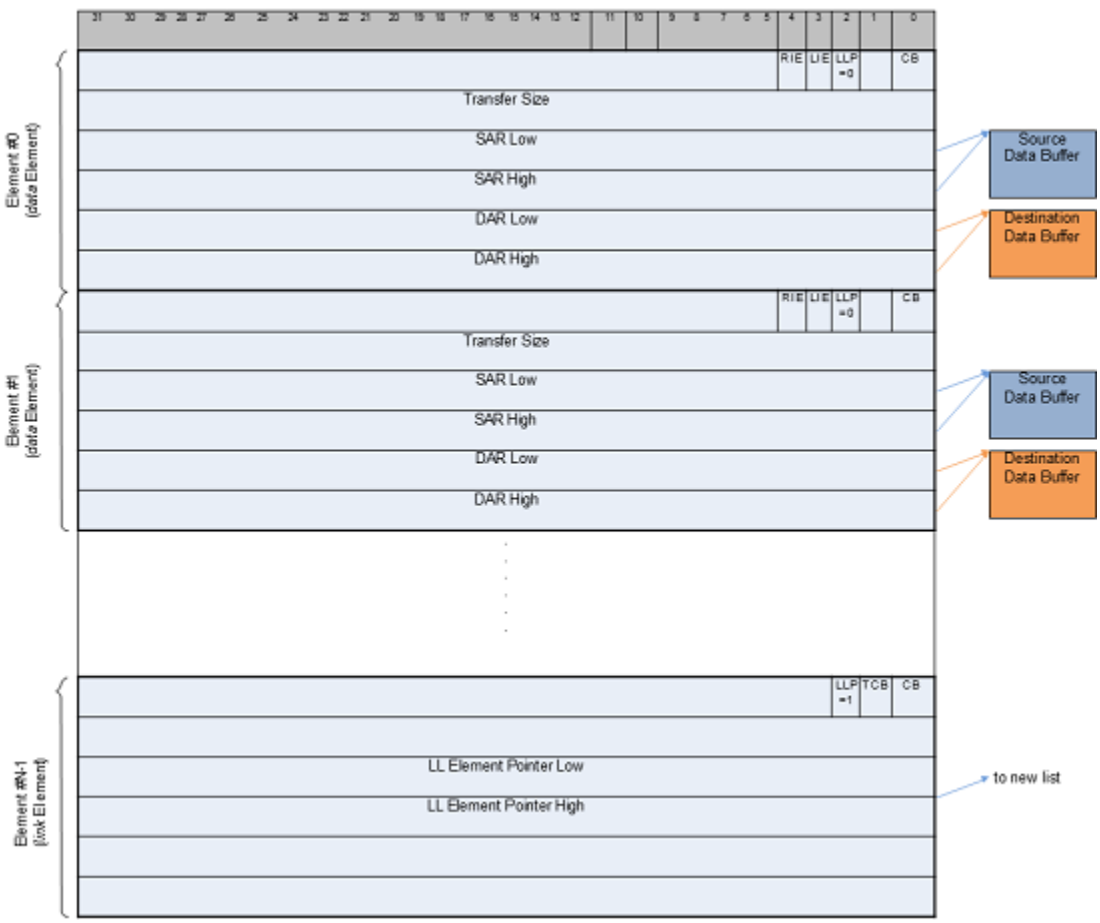


Figure 11-47. Linked List Element/Descriptor Structure in Local Memory with N Elements

When you start the DMA transfer (by writing to the DMA Write Doorbell Register DMA_WRITE_DOORBELL_OFF or DMA Read Doorbell Register DMA_READ_DOORBELL_OFF), the DMA reads (consumes) each element from local memory, and loads the information (SAR, DAR, size, and so on) from that element into the channel context registers in the DMA. These channel context registers determine the operation of the channel that the DMA controller is currently servicing. The DMA then proceeds to transfer the block of data (as defined by the element), and when it is finished, reads the next element from local memory. Normally, all of the elements are contiguous (one after the other) in memory, with the starting address defined in the channel context DMA Linked List Pointer Low Register DMA_LLP_LOW_OFF_WRCH_0. When you want to jump in local memory to another element list (or recycle the consumed elements), then you set the LLP bit in the element (for example, link element #N-1 in figure shown)), specify the location of the next element structure using the LL Element Pointer DWORDs (as indicated in the figure), and, set TCB to 1 (for recycling) or to 0 (to jump to another list).

11.4.2.4.2.3.1 Relationship Between Element DWORDs and Channel Context Registers

Notice the similarity between a data element and the DMA Channel Context registers for each channel. Each element has six DWORDs as in figure 'Linked List Element/ Descriptor Structure in Local Memory with N Elements'. There are eight channel context registers (DWORDs) for a channel. The DMA loads the six element DWORDs into the following channel context:

- CB, LLP, LIE, and RIE fields of the DMA Channel Control 1 register.
- DMA Transfer Size
- DMA SAR Low and DMA SAR High
- DMA DAR Low and DMA DAR High

The definitions of the element DWORD bit fields are the same as the DMA Channel Context registers described in the "DMA Software Register Map", with the exception of the LIE and RIE bits. The LIE and RIE bits in a LL element, only enable the done interrupt. In non-LL mode, the RIE and LIE bits (in the channel context registers) enable the done and abort interrupts.

When the LLP field of the first DWORD in the element is set to 1, the element is a link element. The DMA only loads the following information into the channel context registers:

- CB, TCB, and LLP bits of the first DWORD. The LIE and RIE bits are not defined in a link element.
- LL Element Pointer (3rd and 4th DWORDs) into the DMA Linked List Pointer registers (DMA_LL- P_LOW_REG_WRCH_0 and DMA_LLP_HIGH_OFF_WRCH_0).

Linked List Operation

- When you enable linked list operation (DMA_CH_CONTROL1_OFF_WRCH_0.lle =1b1), then the DMA overwrites the following DMA channel context registers with the following information from linked list data elements:
  - The CB, LLP, LIE, and RIE fields of the DMA Channel Control 1 register.
  - DMA Transfer Size
  - DMA SAR Low & High
  - DMA DAR Low & High
- The structure of a link element is different to that of the data element. A data element has no TCB field. A link element has no LIE or RIE fields. It has no SAR, DAR, or Transfer Size DWORDs, but has LL Element Pointer DWORDs instead of the SAR DWORDs.
- The LIE and RIE bits in a LL element, only enable the done interrupt. In non-LL mode, the RIE and LIE bits enable the done and abort interrupts. For more information, see "Interrupts and Error Handling".

11.4.2.4.3 Using the DMA

This section describes how to use the DMA. It can simultaneously perform the following types of memory transactions:

- DMA write : Transfer (copy) of a block of data from local memory to remote memory
- DMA read :Transfer (copy) of a block of data from remote memory to local memory

Therefore the DMA supports full duplex operation, processing read and write transfers at the same time, and in parallel with normal (non-DMA) traffic. Upon completion of a DMA transfer or an error, the DMA sends an interrupt MWr (IMWr).

The channels are named read channel 7..0 and write channel 7..0. Each channel is programmed using a number of registers.

11.4.2.4.3.1 Source and Destination Address Registers (SAR, DAR)

The DMA channel context SAR and DAR registers (DMA_SAR_LOW_OFF_WRCH_01 / DMA_SAR_HIGH_OFF_WRCH_0 / DMA_DAR_LOW_OFF_WRCH_0 / DMA_DAR_HIGH_OFF_WRCH_0) provide support for remote-to-local, and local-to-remote PCIe address mapping. You program the start of the local and remote data buffers using these registers, and the DMA increments the SAR and DAR as the DMA transfer progresses. For a write transfer, the SAR is the address of the local memory, and the DAR is the address of the remote memory, as shown in Figure 'Write Transfer: SAR and DAR for Write Channel 0'.

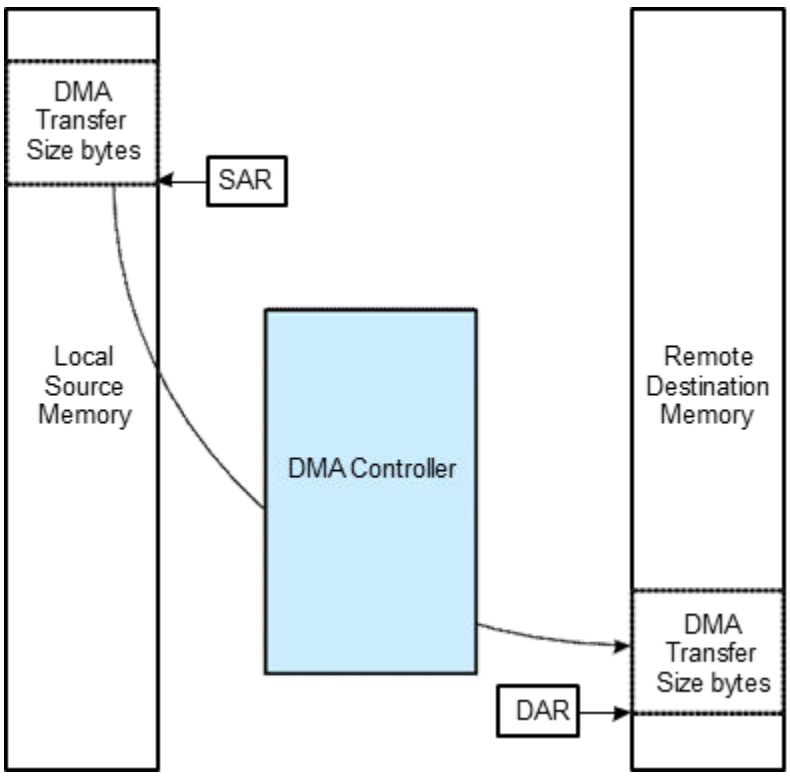


Figure 11-48. Write Transfer: SAR and DAR for Write Channel 0

For a read transfer, the SAR is the address of the remote memory, and the DAR is the address of the local memory, as shown in the figure below.

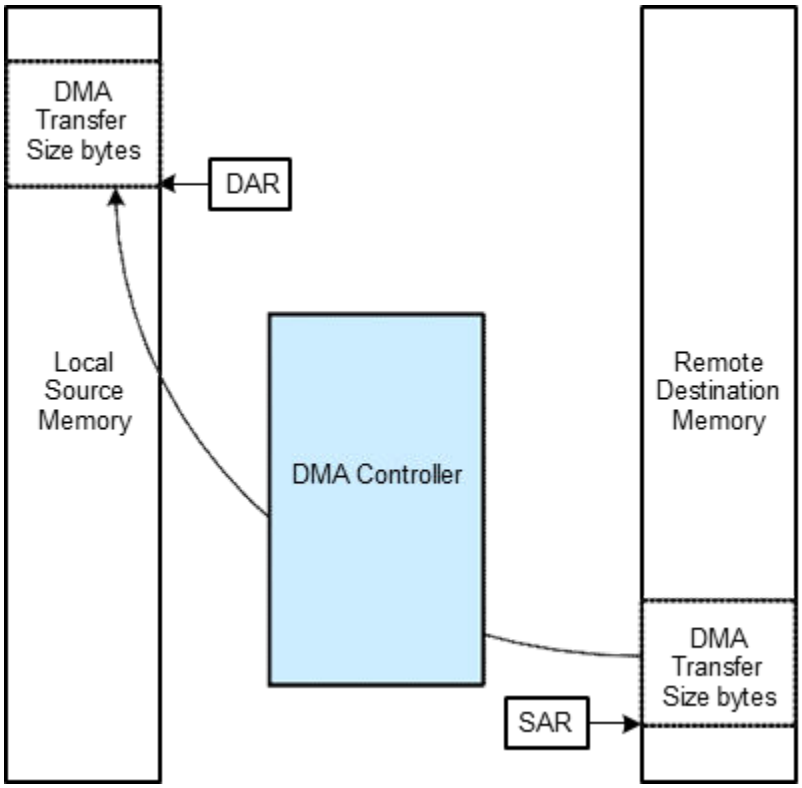


Figure 11-49. Read Transfer: SAR and DAR for Read Channel 0

The presence of the DMA controller does not affect:

- Normal filtering rules for inbound TLPs (for example, BAR checking in an Endpoint, as described in as described in “Receive Filtering”
- The operation of any internal or external address translation as described in “Internal Address Translation Unit (iATU)”).

NOTE

When you do not want the iATU to translate outbound requests that are generated by the internal DMA module, then you must implement one of the following approaches:

- Ensure that the combination of DMA channel programming and iATU control register programming, causes no translation of DMA traffic to be done in the iATU.
- Activate the DMA bypass mode to allow request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. You can activate DMA bypass mode by setting the DMA Bypass field of the iATU Control 2 Register (IATU_REGION_C-TRL_OFF_2_OUTBOUND_0).

11.4.2.4.3.2 DMA Transfer Size Register

You program the DMA transfer size using the DMA Transfer Size Register (DMA_TRANSFER_SIZE_OFF_WRCH_0). The DMA decrements the value in this register as the DMA transfer progresses. You can read this register at any time to determine the number of bytes remaining to be transferred. When all bytes are successfully transferred, the result is zero. The DMA transfer size can be of any value from one byte, up to a maximum of 4 GB.

11.4.2.4.3.3 Starting The DMA Transfer

After you program the DMA controller registers (including writing to the DMA Read Engine Enable or DMA Write Engine Enable register), you start a DMA transfer by writing the channel number to the Doorbell Number field of the DMA Write Doorbell Register (DMA_WRITE_DOORBELL_OFF) or DMA Read Doorbell Register (DMA_READ_DOORBELL_OFF). You can program and start both a read and a write transfer at the same time. The DMA supports full duplex operation, processing read and write transfers at the same time and in parallel with normal (non-DMA) traffic. You can program and start any number of channels in the DMA controller sequentially or simultaneously.

Sequentially: configure and start a channel, and then configure and start another channel

Simultaneously: configure multiple channels, and then start multiple channels

NOTE

You must not write to any of the context registers for a particular channel after you start the channel by writing the channel number to the Doorbell Number field of the DMA Write Doorbell Register (DMA_WRITE_DOORBELL_OFF) or DMA Read Doorbell Register (DMA_READ_DOORBELL_OFF).

11.4.2.4.3.4 Detecting the End of The DMA Transfer

If a DMA transfer proceeds without any errors, it stops automatically when finished.

11.4.2.4.3.4.1 Detecting End of Transfer without Errors

The normal end of a DMA transfer is detected by any of the following methods:

- Done local interrupt (pin) asserted.
- Done remote interrupt (IMWr) received.

- Channel status field of the Channel Control 1 register is Stopped, and the DMA Transfer Size register is 0x0.
- Polling of the DMA Write Interrupt Status Register (DMA_WRITE_INT_STATUS_OFF) or DMA Read Interrupt Status Register (DMA_READ_INT_STATUS_OFF).

11.4.2.4.3.4.2 Detecting End of Transfer with Errors

The abnormal end of a DMA transfer is detected by any of the following methods:

- Abort local interrupt (pin) asserted.
- Abort remote interrupt (IMWr) received.
- Polling of the DMA Write Interrupt Status Register (DMA_WRITE_INT_STATUS_OFF) or DMA Read Interrupt Status Register (DMA_READ_INT_STATUS_OFF).
- Channel Status field of the Channel Control 1 register is halted. When the DMA controller detects an error, it forces the DMA to stop issuing requests for the channel. It also sets the channel status field of the Channel Control 1 register to halted, generates an abort interrupt (if enabled), and sends an abort IMWr (if enabled). The DMA Transfer Size register indicates the remaining number bytes to be transferred, except when there is an AXI write error during a DMA read transfer. For more details, see table 'Possible Sources of an Error during a DMA Read Transfer'.
- Channel Status field of the Channel Control 1 register is Stopped, and the DMA Transfer Size register is not 0x0. You have prematurely stopped this channel as described in “Stopping the DMA Transfer (Software Stop)”.

11.4.2.4.3.5 Stopping the DMA Transfer (Software Stop)

You can manually abort (stop) the DMA transfer by writing the channel number to the Doorbell Number field and writing 1 to the Stop field in DMA_WRITE_DOORBELL_OFF. This causes the DMA to:

- Place the channel in a Stopped state.
- The channel Status field in DMA_CH_CONTROL1_OFF_WRCH_0 is Stopped and the value in DMA_TRANS- FER_SIZE_OFF_WRCH_0 will not be 0x0.
- Wait for all outstanding pending transactions.
- Assert the abort interrupt (if it is enabled) in DMA_WRITE_INT_STATUS_OFF.

You might do this as part of error handling which is described in “Error Handling Assistance by Remote Software”, which is only necessary during software development if you incorrectly program the DMA write channel DAR. You might also do this as part of a function level reset (FLR). FLR does not directly affect the DMA transfer so you must manually stop the DMA transfer before initiating an FLR.

Use This Feature With Caution in Linked List Mode

You should not use the STOP feature in operational mode. It is only a debug feature and has associated hazards.

- Before setting the Stop bit, you must read the channel status field (CS) of the DMA Channel Control 1 register to ensure that the corresponding channel is Running (transferring data). To eliminate the possibility of a race condition between these two actions (read and write), you should confirm the presence of the abort bit in the DMA Write Interrupt Status Register (DMA_WRITE_INT_STATUS_OFF) or DMA Read Interrupt Status Register (DMA_READ_INT_STATUS_OFF) and check the status of any fatal error if some other event has occurred.
- After a Stop event, you cannot seamlessly resume the transfer again because the DMA will not carry on exactly from the point that it was stopped at. Therefore you must setup and start the complete channel context again, including the
  - linked list pointer
  - linked list structure in memory
  - related PCS and CCS values

11.4.2.4.3.6 TLP Generator Header Control

You must program bits [31:12] of the DMA Channel Control 1 Register (DMA_CH_CONTROL1_REG_ WRCH_0/DMA_CH_CONTROL1_OFF_RDCH_0), so that the DMA TLP generator correctly sets the TC, RO, NS, AT, and FN TLP header fields for any TLPs that it generates. When SRIOV is enabled, then you must also program DMA_CH_CONTROL2_OFF_WRCH_0/ DMA_CH_CONTROL2_OFF_RDCH_0.

11.4.2.4.3.7 DMA Software Register Map

This section discusses the grouping and access methods of the DMA registers. The following registers are present when the DMA is enabled.

The DMA Global Registers control the global settings

Table 11-37. DMA Global Registers

Name	Description
0x8_0000 +0x000	DMA Arbitration Scheme for TRGT1 Interface
+0x008	Number of DMA Channels
+0x00C	DMA Write Engine Enable
+0x010	DMA Write Doorbell
+0x018	DMA Write Engine Channel Arbitration Weight Low
+0x01C	DMA Write Engine Channel Arbitration Weight High
+0x02C	DMA Read Engine Enable
+0x030	DMA Read Doorbell
+0x034	DMA Global Read Control
+0x038	DMA Read Engine Channel Arbitration Weight Low
+0x03C	DMA Read Engine Channel Arbitration Weight High

The DMA Interrupt Registers control the generation of local interrupts and remote IMWr. The IMWr addresses and data are stored in RAM.

Table 11-38. DMA Interrupt Registers

Byte Offset	Description
0x8_0000 +0x4C	DMA Write Interrupt Status
+0x50	Not used (RsvdP).
+0x54	DMA Write Interrupt Mask
+0x58	DMA Write Interrupt Clear
+0x5C	DMA Write Error Status
+0x60	DMA Write Done IMWr Address Low
+0x64	DMA Write Done IMWr Address High
+0x68	DMA Write Abort IMWr Address Low
+0x6C	DMA Write Abort IMWr Address High
+0x70	DMA Write Channels 1 and 0 IMWr Data
+0x84	DMA Write Channels 3 and 2 IMWr Data
+0x88	DMA Write Channels 5 and 4 IMWr Data
+0x8C	DMA Write Channels 7 and 6 IMWr Data
+0x90	DMA Write Linked List Error Enable
up to +0x9C	Not used (RsvdP).
+0xA0	DMA Read Interrupt Status
+0xA4	Not used (RsvdP).
+0xA8	DMA Read Interrupt Mask
+0xAC	DMA Read Interrupt Clear
+0xB0	Not used (RsvdP).
+0xB4	DMA Read Error Status Low
+0xB8	DMA Read Error Status High

Table continues on the next page...

Table 11-38. DMA Interrupt Registers (continued)

up to +0xC0	Not used (RsvdP).
+0xC4	DMA Read Linked List Error Enable
+0xC8	Not used (RsvdP).
+0xCC	DMA Read Done IMWr Address Low
+0xD0	DMA Read Done IMWr Address High
+0xD4	DMA Read Abort IMWr Address Low
+0xD8	DMA Read Abort IMWr Address High
+0xDC	DMA Read Channels 1 and 0 IMWr Data
+0xE0	DMA Read Channels 3 and 2 IMWr Data
+0xE4	DMA Read Channels 5 and 4 IMWr Data
+0xE8	DMA Read Channels 7 and 6 IMWr Data

DMA Channel Context Registers: these registers (which are stored in RAM) are for the control and status information that is specific to each channel. There is a set of registers for each channel.

Table 11-39. DMA Channel Context Registers for Each Channel

Byte Offset	Description
DMA Channel Context	
0x8_0000 + 0x200*ia + 0x100*(WR|RD)b	DMA Channel Control 1 Register
+0x004*i + 0x200*i + 0x100 *(WR|RD)	DMA Channel Control 2 Register
+0x008*i + 0x200*i + 0x100 *(WR|RD)	DMA Transfer Size
+0x00C*i +0x200*i + 0x100 *(WR|RD)	DMA SAR Low
+0x010*i + 0x200*i + 0x100 *(WR|RD)	DMA SAR High
+0x014*i + 0x200*i + 0x100 *(WR|RD)	DMA DAR Low
+0x018*i + 0x200*i + 0x100 *(WR|RD)	DMA DAR High
+0x01C*i + 0x200*i + 0x100 *(WR|RD)	DMA Linked List Pointer Low
+0x020*i + 0x200*i + 0x100 *(WR|RD)	DMA Linked List Pointer High

a. i is the number of channels

b. WR=0 | RD=1

11.4.2.4.3.7.1 Implementation of Channel Context Registers

- The channel context registers are stored in RAM whose contents are automatically initialized after power-on.
  - Therefore, you must program every “register” value as the default is undefined

  - You must program all fields marked Reserved to 1’b0
  - These registers are not affected by any of the reset signals
- You must not write to any of the context registers for a particular channel after you start the channel by writing the channel number to the Doorbell Number field of the DMA Write Door- bell Register (DMA_WRITE_DOORBELL_OFF) or DMA Read Doorbell Register (DMA_READ_DOORBELL_OFF).

Linked List Operation

When you enable linked list operation (Linked List Enable field of DMA Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_0) t o 1b1), the DMA overwrites the following DMA channel context registers with the following information from linked list data elements:

- The CB, LLP, LIE, and RIE fields of the DMA Channel Control 1 register.
- DMA Transfer Size
- DMA SAR Low
- DMA SAR High
- DMA DAR Low
- DMA DAR High

NOTE

- The structure of a link element is different to that of the data element. A data element has no TCB field. A link element has no LIE or RIE fields. It has no SAR, DAR, or Transfer Size DWORDs, but has LL Element Pointer DWORDs instead of the SAR DWORDs. For more details, see Relationship Between Element DWORDs and Channel Context Registers.
- The LIE and RIE bits in a LL element, only enable the Done interrupt. In non-LL mode, the RIE and LIE bits enable the Done and Abort interrupts. For more information, see Interrupts and Error Handling.

11.4.2.4.3.8 Programming Examples

This section provides a programming and operation flow for each of the following scenarios:

- 1 MB Write Transfer On Write Channel #0 Initiated by Local CPU (Non LL mode)
- 1 MB Read Transfer On Read Channel #0 Initiated by Local CPU (Non LL mode)
- 1 MB Write Transfer On Write Channel #5 Initiated by Remote CPU (Non LL mode)

- 1 MB Read Transfer On Read Channel #5 Initiated by Remote CPU (Non LL mode)
- Linked List Mode Programming Example (Write Channel #0)

NOTE

The order in which the registers in the following examples are programmed is not important, except for the Doorbell register, which initiates the DMA transfer. Therefore, to start both a "1 MB Write Transfer On Write Channel #0 Initiated by Local CPU (Non LL mode)" and a "1 MB Read Transfer On Read Channel #0 Initiated by Local CPU (Non LL mode)" by executing the commands on the table. If, as mentioned in "Starting The DMA Transfer" , you want both channels to start at (approximately) the same time, then do not execute the last command (at 0x280) in the table 'DMA Register Setup' until after you have executed all the commands in table.

NOTE

You must enable the DMA before you start a DMA channel. Therefore you must write ‘1’ to DMA Read Engine Enable or DMA Write Engine Enable register before you write ‘1’ to the DMA Read Doorbell or DMA Write Doorbell register.

11.4.2.4.3.8.1 1 MB Write Transfer On Write Channel #0 Initiated by Local CPU (Non LL mode)

In this example, the IMWr generation is disabled, as the local CPU initiates the DMA transfer. The local CPU is interrupted using the edma_int bus. The SAR is the address of the local memory, and the DAR is the address of the remote memory, as shown in the figure below. The table below provides the programming details for this example transfer.

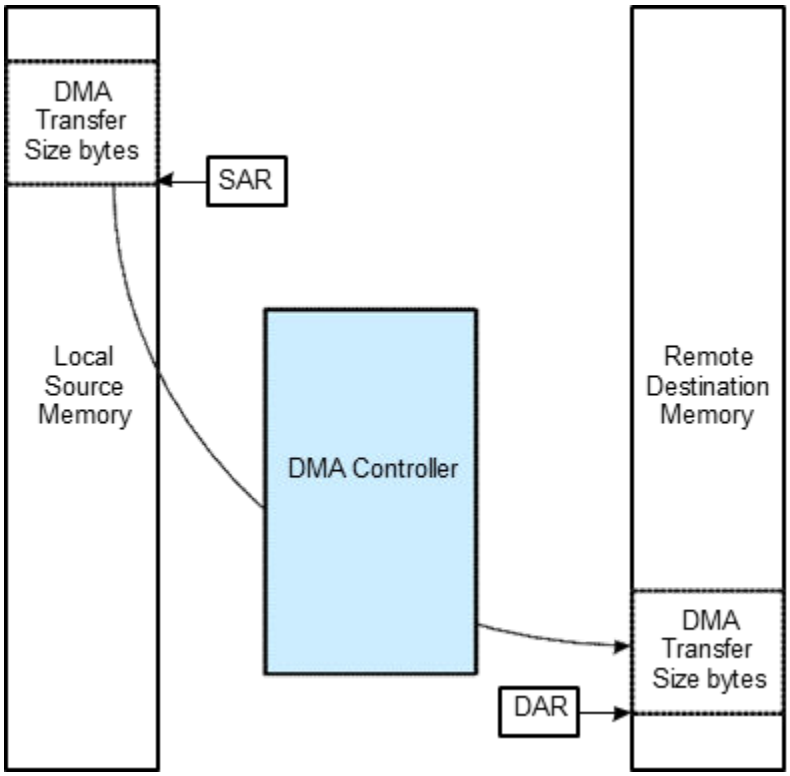


Figure 11-50. 1 MB Write DMA Transfer from 0xBEEF_BEE0 to 0xCAFE_CAF0

Table 11-40. 1 MB Write DMA Transfer from 0xBEEF_BEE0 to 0xCAFE_CAF0

Address (0x8_0000+)	Name	Value
0x00C	DMA Write Engine Enable You must not write 0 to this register. Even temporarily writing 0 to this register resets the DMA logic. For more details, see DMA Write Engine Enable Register (DMA_WRITE_ENGINE_EN_OFF).	0x1
0x054	DMA Write Interrupt Mask	0x0
0x200	DMA Channel Control 1 register • Local Interrupt Enable (LIE) =1 • Remote Interrupt Enable (RIE) =0 • AT, • RO, NS, TC, Function Number =0	0x04000008
0x208	DMA Transfer Size	0x00100000
0x20C	DMA SAR Low	0xBEEF_BEE0
0x210	DMA SAR High	0x0000_0000
0x214	DMA DAR Low	0xCAFE_CAF0
0x218	DMA DAR High	0x0000_0000
0x010	DMA Write Doorbell	0x0

11.4.2.4.3.8.2 1 MB Read Transfer On Read Channel #0 Initiated by Local CPU (Non LL mode)

In this example, the IMWr generation is disabled, as the local CPU initiates the DMA transfer. The local CPU is interrupted using the edma_int bus. The SAR is the address of the remote memory, and the DAR is the address of the local memory, as shown in the figure below. The table below provides the programming details for this example transfer.

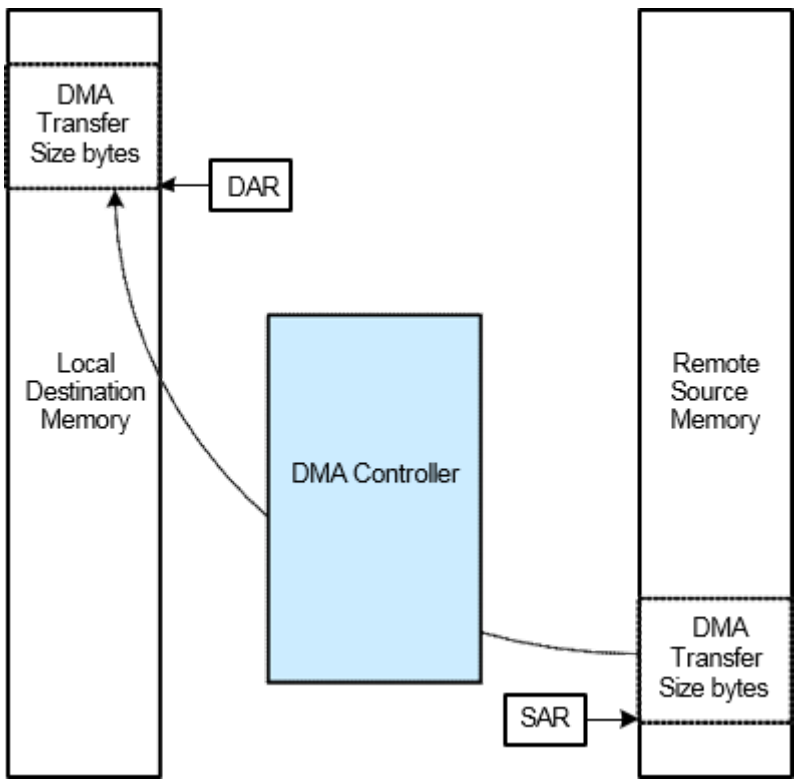


Figure 11-51. 1 MB Read DMA Transfer from 0xBEEF_BEE0 to 0xCAFE_CAF0

Table 11-41. DMA Register Setup

Address (0x8_0000+)	Name	Value
0x02C	DMA Read Engine Enable You must not write 0 to this register. Even temporarily writing 0 to this register resets the DMA logic. For more details, see DMA Read Engine Enable Register (DMA_READ_ENGINE_EN_OFF).	0x1
0x0A8	DMA Read Interrupt Mask	0x0
0x300	DMA Channel Control 1 register • Local Interrupt Enable (LIE) =1	0x04000008

Table continues on the next page...

Table 11-41. DMA Register Setup (continued)

	• Remote Interrupt Enable (RIE) =0 • AT, • RO, NS, TC, Function Number =0	
0x308	DMA Transfer Size	0x00100000
0x30C	DMA SAR Low	0xBEEF_BEE0
0x310	DMA SAR High	0x0000_0000
0x314	DMA DAR Low	0xCAFE_CAF0
0x318	DMA DAR High	0x0000_0000
0x030	DMA Read Doorbell	0x0

11.4.2.4.3.8.3 1 MB Write Transfer On Write Channel #5 Initiated by Remote CPU (Non LL mode)

In this example, the local interrupt generation is disabled, as the remote CPU initiates the DMA transfer. The remote CPU is interrupted using an IMWr. The SAR is the address of the local memory, and the DAR is the address of the remote memory, as shown in the figure below. The table below provides the programming details for this example transfer.

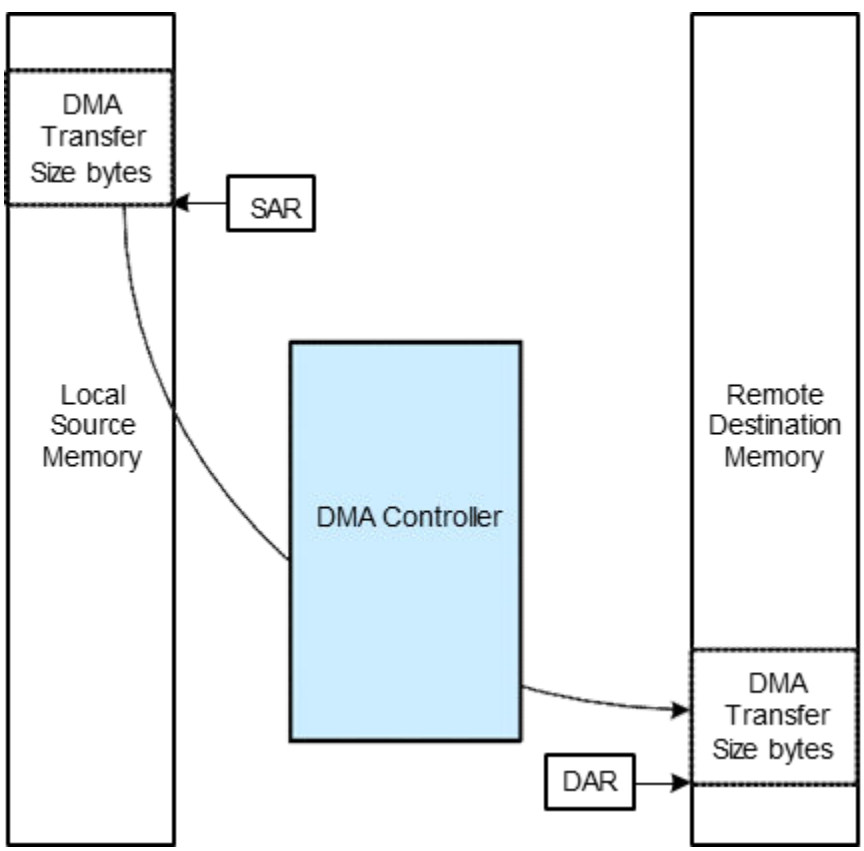


Figure 11-52. 1 MB Write DMA Transfer from 0xBEEF_BEE0 to 0xCAFE_CAF0

Table 11-42. DMA Register Setup

Address (0x8_0000+)	Name	Value
0x00C	DMA Write Engine Enable You must not write 0 to this register. Even temporarily writing 0 to this register resets the DMA logic. For more details, see DMA Write Engine Enable Register (DMA_WRITE_ENGINE_EN_OFF).	0x1
0x060 / 0x064	DMA Write Done IMWr Address Low and High	your IMWr Address #1
0x068 / 0x06C	DMA Write Abort IMWr Address Low and High	your IMWr Address #2
0x06C	DMA Write Channel 0 IMWr Data	your IMWr Data
0xA00	DMA Channel Control 1 register • Local Interrupt Enable (LIE) =0 • Remote Interrupt Enable (RIE) =1 • AT, • RO, NS, TC, Function Number =0	0x04000010
0xA08	DMA Transfer Size	0x00100000
0xA0C	DMA SAR Low	0xBEEF_BEE0
0xA10	DMA SAR High	0x0000_0000
0xA14	DMA DAR Low	0xCAFE_CAF0
0xA18	DMA DAR High	0x0000_0000

11.4.2.4.3.8.4 1 MB Read Transfer On Read Channel #5 Initiated by Remote CPU (Non LL mode)

In this example, the local interrupt generation is disabled, as the remote CPU initiates the DMA transfer. The remote CPU is interrupted using an IMWr. The SAR is the address of the remote memory, and the DAR is the address of the local memory, as shown in Figure below. The table below provides the programming details for this example transfer.

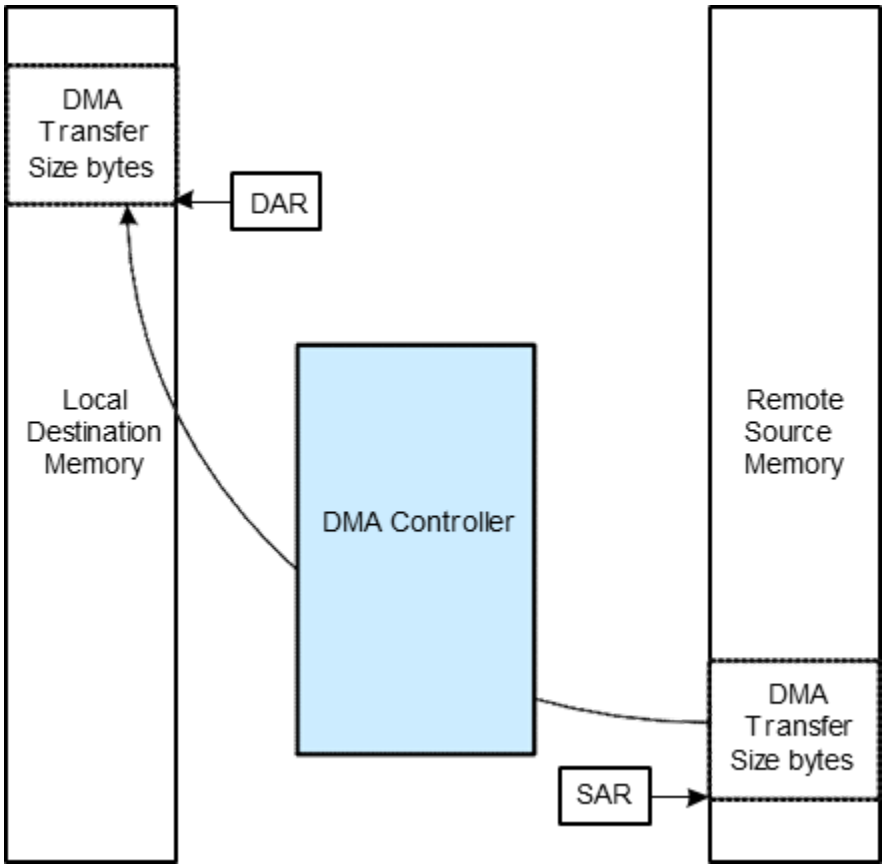


Figure 11-53. 1 MB Read DMA Transfer from 0xBEEF_BEE0 to 0xCAFE_CAF0

Table 11-43. DMA Register Setup

Address (0x700+)	Name	Value
0x29C	DMA Read Engine Enable You must not write “0” to this register. Even temporarily writing 0 to this register resets the DMA logic. For more details, see “DMA Read Engine Enable Register” (DMA_READ_ENGINE_EN).	0x1
0x33C / 0x340	DMA Read Done IMWr Address Low and high	your IMWr Address #1
0x344 / 0x348	DMA Read Abort IMWr Address Low and High	your IMWr Address #2
0x34C	DMA Read Channel 0 IMWr Data	your IMWr Data
0x36C	DMA Channel Context Index	0x8000_0005
0x370	DMA Channel Control 1 register • Local Interrupt Enable (LIE) =0 • Remote Interrupt Enable (RIE) =1 • AT, RO, NS, TC, Function Number =0 • TD =1	0x04000010
0x378	DMA Transfer Size	0x00100000
0x37C	DMA SAR Low	0xBEEF_BEE0
0x380	DMA SAR High	0x0000_0000
0x384	DMA DAR Low	0xCAFE_CAF0
0x388	DMA DAR High	0x0000_0000

Table continues on the next page...

Table 11-43. DMA Register Setup (continued)

0x2A0	DMA Read Doorbell	0x5

11.4.2.4.3.8.5 Linked List Mode Programming Example (Write Channel #0)

This write channel example corresponds to figure 'Example Producer (Software) - Consumer (DMA) Synchronization Flow', with the following features:

- Recycled linked list (LL) of nine data elements1and one link element.
- Transfer is initiated by local CPU.
- Each data element corresponds to a 1 MB (0x0010_0000) write transfer.
- The SARs for each element are separated by 2 MB (0x0020_0000). The DAR for each element are sepa- rated by 16 MB (0x0100_0000). The SAR is the address of the local memory, and the DAR is the address of the remote memory, as shown in the figure below.
- The LL transfer list is located at address 0x0000_0200 in local memory.
- No remote interrupt is generated so the IMWr generation is disabled, as the local CPU initiates the DMA transfer. The local CPU is interrupted using the edma_int bus.
- There is a Watermark interrupt at element #5, and an Empty interrupt at element #8. Both of these interrupts are done interrupts (as opposed to abort interrupts).

You should familiarize yourself with the Linked List Flow for Producer and Consumer as shown in Figure 'Linked List Flow for Producer and Consumer'. These are the main steps for this process in this example:

Step 1: Create TL in local memory:

- Create the 10 elements as per Table 'Linked List Element Initial Setup'.
  - For elements 0, 1, 2, 3, 4, 6, 7: RIE, LIE, LLP, CB =0,0,0,1
  - For element 5: RIE, LIE, LLP, CB =0,1,0,1 ❑For element 8: RIE, LIE, LLP, CB =0,1,0,1 ❑For element 9: LLP, TCB, CB =1,1,0
- For more details, see “PCS-CCS-CB-TCB Producer-Consumer Synchronization” and “Linked List Mode”

Step 2: Program and Start DMA:

- Configure and start the DMA registers as per Table 'DMA Register Setup'.

Step 4: Wait for done interrupt and recycle TL

- For more details, see “Using Interrupts for Linked List Producer-Consumer Synchronization”
- When recycling elements, consider “Element Recycling”

NOTE

If the DMA driver is running on the host and the interrupt service routine is reading local interrupts to determine if the transfer is successful, then you must set LIE and RIE in the same element and you should mask or ignore the local interrupt pin. Setting RIE and LIE in element A followed by RIE (only) in element B is not a verified usage scenario.

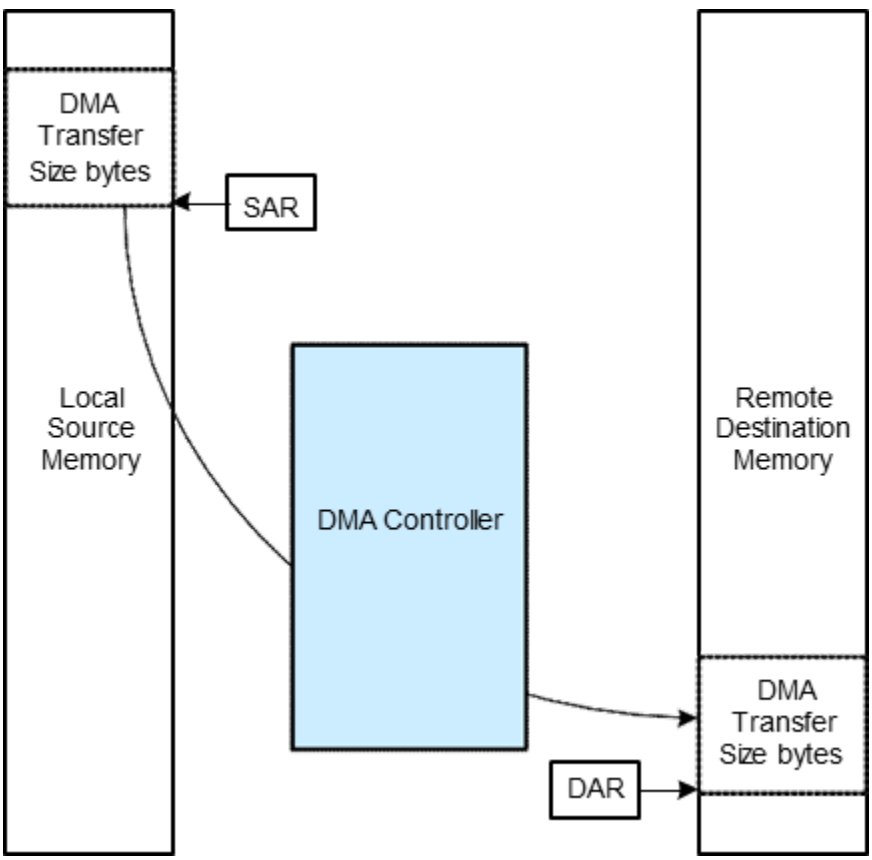


Figure 11-54. 1 MB Write DMA Transfer from SAR to DAR

Table 11-44. Linked List Element Initial Setup

Element #	Address	Data	Label
0	0x0000_0200	0x00000001	Channel Control
0	0x0000_0204	0x00100000	Transfer Size
0	0x0000_0208	0xBEEF_BEE0	SAR Low
0	0x0000_020C	0x0	SAR High
0	0x0000_0210	0xCAFE_CAF0	DAR Low
0	0x0000_0214	0x0	DAR High
1	0x0000_0218	0x00000001	Channel Control
1	0x0000_021C	0x00100000	Transfer Size
1	0x0000_0220	0xBF0F_BEE0	SAR Low

Table continues on the next page...

Table 11-44. Linked List Element Initial Setup (continued)

1	0x0000_0224	0x0	SAR High
1	0x0000_0228	0xCBFE_CAF0	DAR Low
1	0x0000_022C	0x0	DAR High
2	0x0000_0230	0x00000001	Channel Control
2	0x0000_0234	0x00100000	Transfer Size
2	0x0000_0238	0xBF2F_BEE0	SAR Low
2	0x0000_023C	0x0	SAR High
2	0x0000_0240	0xCCFE_CAF0	DAR Low
2	0x0000_0244	0x0	DAR High
3	0x0000_0248	0x00000001	Channel Control
3	0x0000_024C	0x00100000	Transfer Size
3	0x0000_0250	0xBF4F_BEE0	SAR Low
3	0x0000_0254	0x0	SAR High
3	0x0000_0258	0xCDFE_CAF0	DAR Low
3	0x0000_025C	0x0	DAR High
4	0x0000_0260	0x00000001	Channel Control
4	0x0000_0264	0x00100000	Transfer Size
4	0x0000_0268	0xBF6F_BEE0	SAR Low
4	0x0000_026C	0x0	SAR High
4	0x0000_0270	0xCEFE_CAF0	DAR Low
4	0x0000_0274	0x0	DAR High
5	0x0000_0278	0x00000009	Channel Control
5	0x0000_027C	0x00100000	Transfer Size
5	0x0000_0280	0xBF8F_BEE0	SAR Low
5	0x0000_0284	0x0	SAR High
5	0x0000_0288	0xCFFE_CAF0	DAR Low
5	0x0000_028C	0x0	DAR High
6	0x0000_0290	0x00000001	Channel Control
6	0x0000_0294	0x00100000	Transfer Size
6	0x0000_0298	0xBFAF_BEE0	SAR Low
6	0x0000_029C	0x0	SAR High
6	0x0000_0290	0xD0FE_CAF0	DAR Low
6	0x0000_0294	0x0	DAR High
7	0x0000_0298	0x00000001	Channel Control
7	0x0000_029C	0x00100000	Transfer Size
7	0x0000_02A0	0xBFCF_BEE0	SAR Low
7	0x0000_02A4	0x0	SAR High
7	0x0000_02A8	0xD1FE_CAF0	DAR Low
7	0x0000_02AC	0x0	DAR High
8	0x0000_02B0	0x00000009	Channel Control

Table continues on the next page...

Table 11-44. Linked List Element Initial Setup (continued)

8	0x0000_02B4	0x00100000	Transfer Size
8	0x0000_02B8	0xBFEF_BEE0	SAR Low
8	0x0000_02BC	0x0	SAR High
8	0x0000_02C0	0xD2FE_CAF0	DAR Low
8	0x0000_02C4	0x0	DAR High
9	0x0000_02C8	0x00000006	Channel Control
9	0x0000_02CC	0x0	Reserved
9	0x0000_02D0	0x0000_0200	Linked List Element Pointer Low
9	0x0000_02D4	0x0	Linked List Element Pointer High

Table 11-45. DMA Register Setup

Address **(0x8_0000+)**	**Name**	**Value**
0x030	DMA Write Engine Enable You must not write 0 to this register. Even temporarily writing 0 to this register resets the DMA logic. For more details, see DMA Write Engine Enable Register (DMA_WRITE_ENGINE_EN_OFF).	0x1
0x054	DMA Write Interrupt Mask	0x0
0x090	DMA Write Linked List Error Enable • LLLAIE =1	0x1
0x200	DMA Channel Control 1 register • Linked List Enable (LLE) =1 • Consumer Cycle Status (CCS) =PCS =1 • AT, • RO, NS, TC, Function Number =0	0x04000300
0x21C	DMA Linked List Pointer Low	0x0000_0200
0x220	DMA Linked List Pointer High	0x0000_0000
0x010	DMA Write Doorbell	0x0

11.4.2.4.4 Advanced DMA Information and Operation

This section discusses advanced features and operation of the DMA.

11.4.2.4.4.1 Overview

The DMA write and read channels operate independently to maximize the performance of the DMA read and write data transfers over the PCIe link.

The DMA can simultaneously perform the following types of memory transactions:

- DMA write: transfer (copy) of a block of data from local memory to remote memory
- DMA read: transfer (copy) of a block of data from remote memory to local memory

After you have programmed and started a DMA transfer (see "Using the DMA"), the DMA transfers the data as described in "**DMA Write Transfer"** and "**DMA Read Transfer"**

11.4.2.4.4.2 DMA Ordering

The DMA assumes that all DMA read channel data must be written to the local application memory in strict address order. For a DMA read transfer, each original outgoing request has a unique PCIe tag. The remote link partner or intermediate switch does not have to preserve the order of CPLs corresponding to such individual requests. When the DMA receives the CPLs out of order, it reorders that data in the "Read Buffer" as described in "**DMA Read Transfer"** . DMA write transfer data is sourced from the local application memory and sent to the remote memory in address order.

11.4.2.4.4.3 Error Handling

The DMA supports full AXI and PCIe error handling in conjunction with the PCIe controller. The tables below list the possible sources of an error for DMA read and write transfers.

Table 11-46. Possible Sources of an Error during a DMA Read Transfer

Source	Type	Description
Application Write Error Detected	Fatal	The DMA read channel has received an error response from the AXI bus at the bridge master interface (or TRGT1 interface when the AXI bridge is not used) while writing posted data to it. This error is fatal. • You must restart the transfer from the beginning because the channel context is corrupted.

Table continues on the next page...

Table 11-46. Possible Sources of an Error during a DMA Read Transfer (continued)

		• The MWr is not rolled back. • The AXI bridge master does not report this error to software through error logging or MSG. **NOTE:** During linked list mode, the Application Write Error only corrupts the current element in the linked list. This is because the DMA does not fetch the next element from local memory until it has completely finished transferring the current block of data (as defined by the current linked list element). Finished implies that your application has successfully acknowledged (without ERROR) all of the MWr requests.
Unsupported Request (UR)	Non-fatal	The DMA read channel has received a PCIe UR CPL status from the remote device in response to the MRd request.
Completer Abort (CA)	Non-fatal	The DMA read channel has received a PCIe CA CPL status from the remote device in response to the MRd request.
CPL Time Out	Non-fatal	The DMA read channel has timed-out while waiting for the remote device to respond to the MRd request, or a malformed CplD has been received.
Data Poisoning	Fatal	The DMA read channel has detected data poisoning in the CPL from the remote device in response to the MRd request. The DMA read channel will drop the completion and then be halted.
Linked List Element Fetch Error Detected	Fatal	The DMA read channel has received an error response from the AXI bus (or TRGT1 interface when the AXI bridge is not used) while reading a linked list element from local memory.

Table 11-47. Possible Sources of an Error during a DMA Write Transfer

Source	Type	Description
Application Read Error Detected	Fatal	The DMA write channel has received an error response from the AXI bus at the bridge master interface (or TRGT1 interface when the AXI bridge is not used) while reading data from it. • The response TLP is discarded by the DMA and is not converted to an outbound MWr TLP.

Table continues on the next page...

Table 11-47. Possible Sources of an Error during a DMA Write Transfer (continued)

		• The UR or CA CPL status TLP generated by the bridge is discarded by the DMA. • The AXI bridge master does not report this error to software through error logging or MSG.
Linked List Element Fetch Error Detected	Fatal	The DMA write channel has received an error response from the AXI bus at the bridge master interface (or TRGT1 interface when the AXI bridge is not used) while reading a linked list element from local memory. • The UR or CA CPL status TLP generated by the bridge is discarded by the DMA. • The AXI bridge master1 does not report this (or for non-DMA traffic) error to software through error logging or MSG.

When the DMA detects an error, it performs the following actions (for the channel with the error):

- Stops issuing new requests to the remote link partner.
- Stops issuing new requests.
  - All data in the “Read Buffer”, that was received before the error was detected, is valid. In the case of a non-fatal error, this is reflected in the DMA Transfer Size Register DMA_TRANSFER_SIZE_OFF_WRCH_0.
  - All data in the read buffer (for that channel), that was received after the error was detected, is discarded.
- Places the DMA channel in a halted state (channel status field of the DMA Channel Control 1 Register is halted).
- Waits for all outstanding requests and CPLs to complete.
- Generates the abort interrupt or IMWr, if enabled, as outlined in “Interrupt Handling”.

For a DMA read transfer, you can read the DMA Read Error Status High Register (DMA_READ_ERR_ STATUS_HIGH_OFF) and DMA Read Error Status Low Register (DMA_READ_ERR_STATUS_LOW_OFF) to identify the source of the error. For a DMA write transfer (in linked list mode only), you can read the DMA Write Error Status Register (DMA_WRITE_ERR_STATUS_OFF) to identify the source of the error. After your software has finished error and interrupt routine handling for Non-Fatal errors, you can request the DMA to continue processing (for the channel with the error) by:

1. Reprogramming the SAR based on the number of bytes remaining in the DMA Transfer Size register
2. Writing the channel number to the Doorbell Number field of the DMA Write Doorbell Register (DMA_WRITE_DOORBELL_OFF) or DMA Read Doorbell Register (DMA_READ_DOORBELL_REG).

NOTE

You cannot continue after a Fatal Error is detected, as the channel context is corrupted. Any actions that were in progress are not rolled back. To exit from this condition, you need to Soft Reset the complete DMA controller, as described in DMA Write Engine Enable Register (DMA_WRITE_ENGINE_EN_OFF) and DMA Read Engine Enable Register (DMA_READ_ENGINE_EN_OFF).

When an error occurs in linked list mode, the next LL element is not fetched from local memory. DMA Linked List Pointer Low Register (DMA_LLP_LOW_OFF_WRCH_0) is not incremented by the DMA, but remains pointing to the element that caused the error.

11.4.2.4.4.3.1 Error Handling Assistance by Remote Software

During a DMA write transfer, the DMA is not aware of MWr errors that are detected in the remote device, because an MWr is a posted request and has no corresponding completion TLP. In these cases, your remote CPU must manually abort the DMA transfer by writing the channel number to the Doorbell Number field, and 1 to the Stop field of the DMA Write Doorbell Register (DMA_WRITE_DOORBELL_OFF). This causes the DMA to:

- Place the channel in a stopped state. The channel Status field of the DMA Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_0) is stopped and the DMA Transfer Size register is not 0x0.
- Wait for all outstanding pending transactions.
- Assert the abort interrupt if it is enabled

11.4.2.4.4.4 Linked List Operation

This section describes the detailed operation of the DMA in linked list (LL) mode. It also discusses how the software should produce LL elements1, and how the DMA and software maintain synchronization.

11.4.2.4.4.4.1 LL Operation Overview

You should first read "Linked List Mode" for an overview of LL mode. In this section, a normal LL operation called recycling2 is described in detail. The process is described for a write channel, but it is possible to have duplicate processes running in parallel for other write or read channels. In this process interrupts are used to trigger the producer (software) to recycle elements. Typically a Watermark interrupt (positioned near the middle of the transfer list (TL)) and an Empty interrupt (in the last data element) are used by setting LIE in these two elements to ‘1’.

The steps in this process are:

1. Software creates a LL element structure called a TL in local memory consisting of N-1 data elements and one link element.

   The link elements "LL Element Pointer Low/High DWORDs"(as shown in Linked List Element figure in Linked List Mode topic) are programmed to point back to the beginning of the first data element.

   One of the data elements (near the middle of the TL) is programmed to generate a "Watermark" interrupt.

   The last data element is programmed to generate an "Empty" interrupt.

2. Software programs the DMA with the location of the TL. This is done by writing to the DMA Linked List Pointer Low Register (DMA_LLP_LOW_OFF_WRCH_0) and DMA Linked List Pointer High Register (DMA_LLP_HIGH_OFF_WRCH_0).
3. Software starts the DMA process by ringing the write channel Doorbell. This is done by writing to the DMA Write Doorbell Register (DMA_WRITE_DOORBELL_OFF).

   The solid purple loop L1 in Figure 11-55 corresponds to steps 1-3.

4. The DMA reads (**consumes**) each data element from local memory, and if the CB and CCS bits match, loads the information (SAR, DAR, size, and so on) from that data element into the channel context registers. These context registers determine the operation of the channel that the DMA is currently servicing. The DMA executes the check for the CB/CCS match after it has evaluated TCB and toggled CCS. The DMA always loads link elements into the channel context.

NOTE

If the DMA stops on a link element, then your software must set the DMA Linked List Pointer registers, and restart the DMA process by ringing the doorbell.

With recycling, the same LL structure is used over and over again. Another mode of operation would be to jump to a new LL element structure when the current one is consumed.

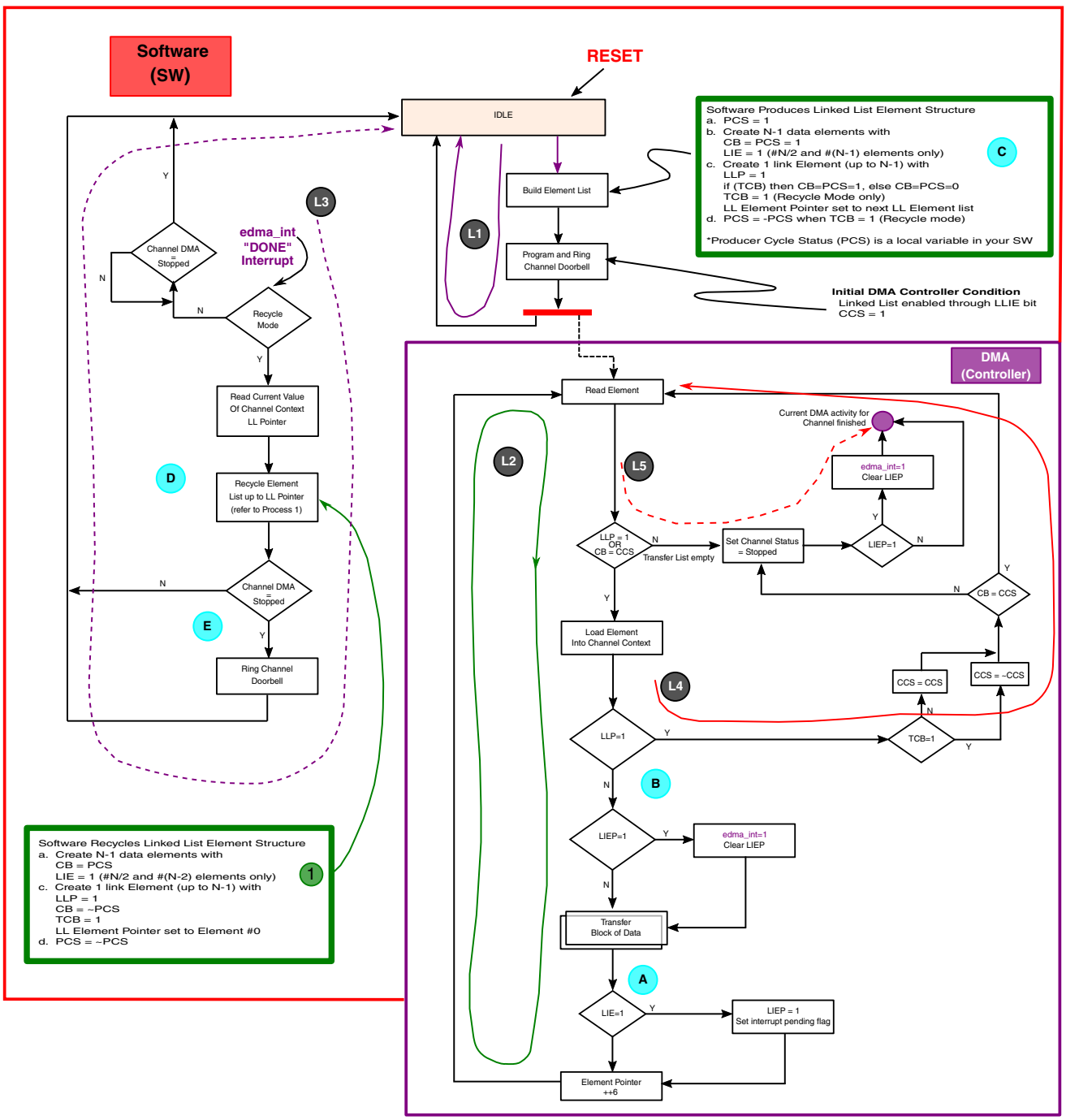


Figure 11-55. Linked List Flow for Producer and Consumer

5. The DMA then proceeds to transfer the block of data (as defined by the element), and when it is finished, reads the next element from local memory.

The solid green loop L2 in Figure 11-55 corresponds to steps 4-5.

6. The last element in the list (called a link element, and indicated by having its LLP field set to 1) is not used to transfer a block of data.

   It causes the DMA to effectively repeat the task in step 4.

   Its TCB is typically set to 1. This causes the DMA to toggle its CCS bit, and the software to toggle its PCS bit. The software and DMA use the "PCS-CCS-CB-TCB" producer-consumer synchronization mechanism to ensure that:

   Software does not recycle elements that have not yet being consumed by the DMA

   DMA correctly recognizes and consumes recycled elements. The solid red loop L4 in Figure 11-55 corresponds to step 6.

7. Upon reception of the interrupt (mentioned in step 1), the software starts to recycle the TL.

   Software reprograms each data element with new DMA transfer information.

   The software and DMA use the "PCS-CCS-CB-TCB" producer-consumer synchronization mecha- nism described later in this section.

   The dashed purple loop L3 in Figure 11-55 corresponds to step 7.

8. As some point, the software wants to terminate the complete DMA process, by not recycling any more elements. The DMA recognizes this condition when CB !=CCS, and sets the channel status to stopped

   The dashed red loop L5 in Figure 11-55 corresponds to step 8.

11.4.2.4.4.4.2 Using Interrupts for Linked List Producer-Consumer Synchronization

When the DMA is finished with an element, the DMA checks the LIE bit, before reading the next LL element. For more details, see circled step A in Figure 11-55. When the LIE is set, then the DMA does not assert the "done" interrupt immediately, but sets an internal interrupt pending flag (LIEP). It asserts the actual done interrupt after the next data element has been read from system memory. For more details, see circled step B in Figure 11-55. This automatic internal process of delaying the interrupt avoids a race condition between the DMA and software when the LIE bit is set in an element.

Your software should set the "Watermark" interrupt early to schedule the recycling of the consumed elements. Placing the Watermark interrupt too far down in the list, combined with a slow element recycling process in your application; ensures that the DMA returns to the start of the LL before your software has recycled it. The DMA channel STOPs, and you have to restart it by writing to its Doorbell. Typically a "Watermark interrupt"

(positioned near the middle of the TL) and an Empty interrupt (in the last data element) are used by setting LIE in these two elements to 1b1. Upon reception of the interrupt, the software starts to recycle the TL.

Operational interrupts are always "done" interrupts. For more details, see "Interrupts and Error Handling"

NOTE

The DMA processes the remote interrupt enable (RIE) bit in the LL element in the same way as the LIE. Therefore, everywhere that LIE and edma_int are mentioned in this section can be interpreted to mean LIE and edma_int, or RIE and IMWr.

11.4.2.4.4.4.3 PCS-CCS-CB-TCB Producer-Consumer Synchronization

The software and DMA use the "PCS-CCS-CB-TCB" producer-consumer synchronization mechanism to ensure that software does not recycle elements that have not yet being consumed by the DMA, and that the DMA correctly recognizes and consumes recycled elements. This process, which is shared between the DMA and the software, is illustrated through an example producer-consumer flow in the figure below.

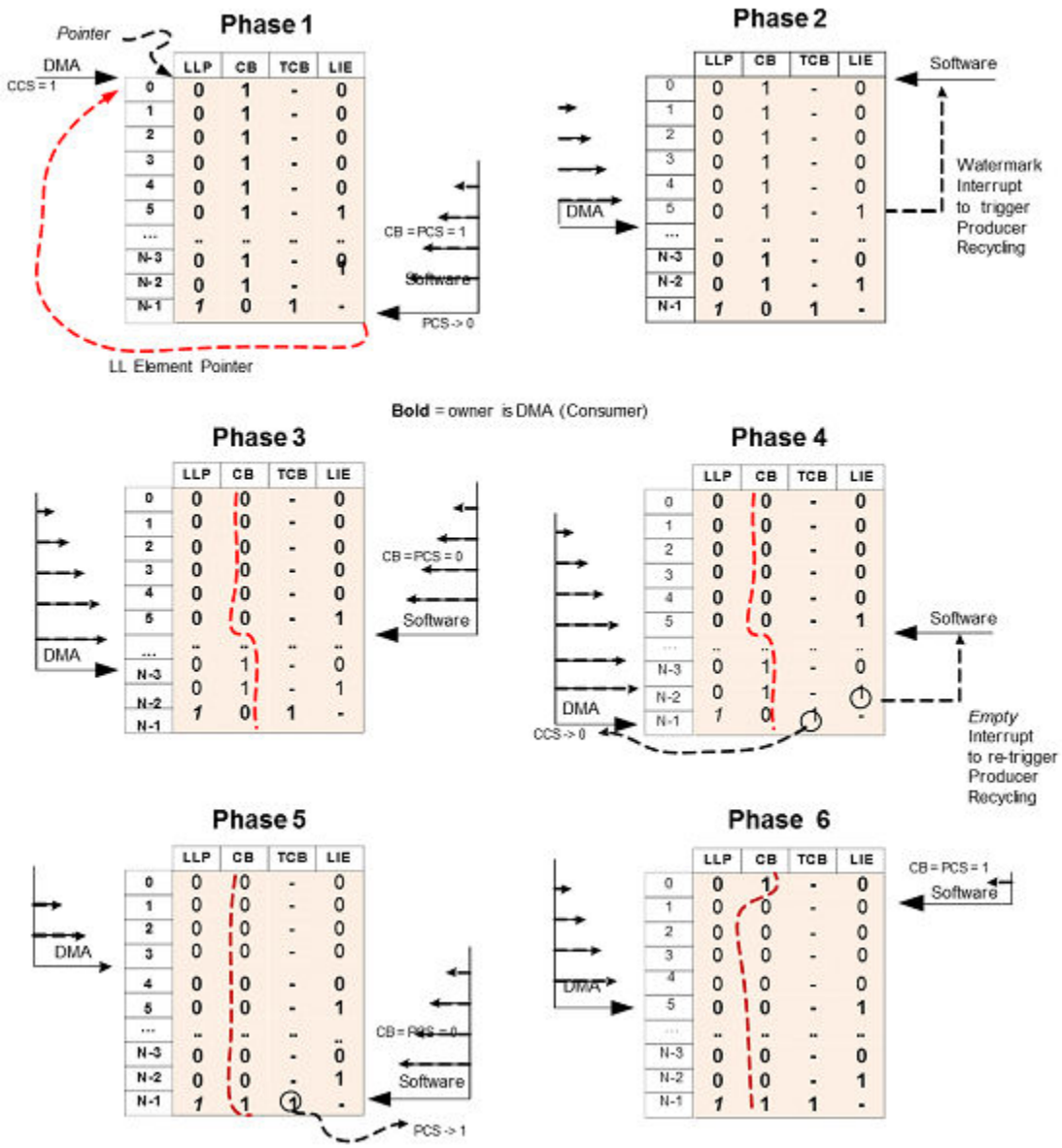


Figure 11-56. Example Producer (Software) - Consumer (DMA) Synchronization Flow

Before looking at the example in detail, it is useful to note that the DMA performs the following two tests as part of this process:

- Consumer-Owned Element, or Transfer List (TL) Empty Test

  The solid red loop L4 in the Figure 11-55 corresponds to this test. The Cycle Bit (CB) of DMA Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_0) (which was loaded from the LL element) is tested against the DMA Consumer Cycle State (CCS) bit of the same register. When CB =CCS, the element is owned by the consumer (DMA) and the data transfer

is executed. When CB !=CCS, the TL is empty and the DMA sets channel Status (CS) in DMA Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_0) to "Stopped".

- Toggle Cycle Bit (TCB) Test

  The Dashed Red Loop Marked L5 in Figure 11-55 corresponds to this test. When the Toggle Cycle Bit (TCB) of the DMA Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_0) (which was loaded from the LL element) is set to 1, the DMA toggles the CCS bit of the DMA Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_0) to be ~CB. The producer (software) also performs a similar action on its local variable "Producer Cycle Status" (PCS), as it sets PCS =~PCS. The TCB field is only valid in the last LL element (the link element). Toggling the PCS and CCS flags in this manner synchronizes the consumer to the producer.

The example in Figure 11-56 has the following six phases:

1. The software builds the transfer list for the write (or read) channel, according to the steps outlined in the circled step C in Figure 11-55. For each element, CB =PCS =CCS =1. The exception to this is the link element where CB is set to the next iteration value because the link element is always loaded into the channel context.
   - The link element points to the start of the TL and its TCB bit is 1. Software toggles its PCS variable.
   - Software writes the location of the TL into the channel context LL Pointer register.
   - Software rings the write channel Doorbell.
2. The DMA starts processing elements. After processing element 5 (and fetching the next element), the DMA asserts a Watermark “done” interrupt to signal the software to start recycling elements.
3. The software is recycling elements. It reads the channel context DMA Linked List Pointer Low Register (DMA_LLP_LOW_OFF_WRCH_0) and DMA Linked List Pointer High Register (DMA_LL- P_HIGH_OFF_WRCH_0), and recycles elements up to this location1in the TL. It sets CB for each element to be PCS. The Dashed Purple Loop Marked L3 in Figure 11-55 corresponds to this activity. The DMA continues processing (consuming) elements as before.
4. The DMA then asserts an Empty “done” interrupt (from the LIE bit in the last data element at the end of TL that was set in the previous phase) to indicate to the software, to start recycling elements again. The DMA reaches the end of the TL, and as the TCB is set to 1, it toggles its CCS bit to 0 (~CB). For more details, see circled step D in Figure 11-55.
5. The software reads the channel context DMA Linked List Pointer Low Register (DMA_LLP_LOW_REG_WRCH_0) and DMA Linked List Pointer High Register

(DMA_LLP_HIGH_OFF_WRCH_0), and recycles elements up to this location in the TL. It sets CB for each element to be PCS. As it recycles the link element, it detects that TCB is set to 1. It toggles its internal variable PCS to 0, and the initial TL has been fully recycled. The DMA starts processing the recycled elements.

6. The DMA continues processing (consuming) elements as before. Software did not have to ring the Doorbell, as the first element was already recycled before the DMA started processing it. The software might continue recycling from the top of the list, if the DMA had already returned to the top of the list when the software read the channel context DMA Linked List Pointer Low Register (DMA_LLP_LOW_OFF_WRCH_0) and DMA Linked List Pointer High Register (DMA_LL-P_HIGH_OFF_WRCH_0), in step 5. If not, it starts recycling again when the DMA processes element 4 and the DMA asserts a Watermark “done” interrupt.

11.4.2.4.4.4.4 Element Recycling

When your software is recycling elements, it should program the first DWORD (control bits) in the Figure 11-47 only after the other DWORDs have been programmed. In addition, it should set the CB bit (to ~PCS) within this DWORD only after the other bits in this DWORD have been set. This programming sequence avoids the DMA incorrectly fetching an element that software is currently recycling.

When the software is finished recycling it must check the channel context status (CS). When CS is "Stopped", the software must ring the channel Doorbell again. For more details, see circled step E in Figure 11-43. If your element recycling process was fast enough, then the DMA will seamlessly move from processing the last old element to the first new element, without the need for the software to ring the channel Doorbell again. To eliminate the possibility of a race condition between these two actions (read and write), you should first confirm the presence of the "done" interrupt bit in the DMA Write Interrupt Status Register (DMA_WRITE_INT_STATUS_OFF) or DMA Read Interrupt Status Register (DMA_READ_INT_ STATUS_OFF) and also check the DMA Linked List Pointer Low Register (DMA_LLP_LOW_REG_WRCH_0) to confirm that it is pointing to the correct location.

11.4.2.4.4.4.5 Link Down Recovery

During normal operation when the link goes down, the controller generates a "link reset" request. A

high-to-low transition on link_req_rst_not indicates that the PCIe controller is requesting external logic to reset the PCIe controller because the PHY link is down. For more details, see "Hot Reset". To determine the progress of the linked list operation when the

link went down, you can read the contents of the channel context DMA Linked List Pointer Low Register (DMA_LLP_LOW_OFF_WRCH_0) and DMA Linked List Pointer High Register (DMA_LLP_HIGH_OFF_WRCH_0).

11.4.2.4.4.5 Multichannel Arbitration

A DMA write and read channel operate independently in full duplex mode. This maximizes the performance of the DMA read and write data transfers over the PCIe link. When DMA is configured with multiple read channels, it uses a weighted round robin (WRR) arbitration scheme to select the next read channel to be serviced. The same applies when you have multiple write channels. The DMA handles the multiple channels in a time-multiplexed way. From a software point of view every channel operates independently, so each one has its own instance of linked list and context. The bandwidth for all channels is shared in time and is controlled by a weighted round-robin arbiter.

The arbitration weight for each channel can be set using the DMA_WRITE_CHANNEL_ARB_WEIGHT_LOW_OFF, DMA_WRITE_CHANNEL_ARB_WEIGHT_HIGH_OFF), DMA_READ_CHANNEL_ARB_WEIGHT_LOW_REG, and DMA_READ_CHANNEL_ARB_WEIGHT_HIGH_OFF registers. Each register has a 5-bit field (representing a weight from 1 to 32) for up to four channels. The weights must be set before "ringing" the doorbell.

The channel weight specifies the number of TLP requests that the DMA can issue for that channel before it must return to the arbitration routine. When the channel weight count is reached, the WWR arbiter selects the next channel to be processed. For example, consider two DMA write channels (write channel #0 and write channel #1) where you want to assign the bandwidth for write channel #1 to be two times that of write channel #3. Therefore set:

- Write channel #0 weight to 32
- Write channel #1 weight to 16

Depending on the transfer size and the Maximum Payload Size, the DMA will issue 32 MRd requests for write channel #0, followed by 16 MRd requests for write channel #1, followed by 32 MRd requests for write channel #0 and so on, until the current transfer (or element when using LL mode) is finished. Therefore in this example, the weights reserve 33% bandwidth for write channel #1 and 67% bandwidth for write channel #0.

NOTE

- Max tag constraints the maximum number of outstanding DMA write engine MRd requests to the application, so the

sum of weights should not exceed this value. The Max tag for this chip is 8.
- If these limits are exceeded, then the respective DMA engine requester thread is back back pressured, defeating purpose of the round-robin bandwidth sharing mechanism.
- When the ratio (transfer-size / TLP-size) is smaller than the round-robin arbiter setting (say, DMA_WRITE_CHANNEL_ARB_WEIGHT_LOW_OFF), the DMA issues less requests than the value in DMA_WRITE_CHANNEL_ARB_WEIGHT_LOW_OFF. The TLP size is determined by the DMA using each function's Maximum Payload Size and Maximum Read Request Size rules. This means that the arbitration value in DMA_WRITE_CHANNEL_ARB_WEIGHT_LOW_OFF determines the number of TLPs send and not the absolute data bandwidth.

11.4.2.4.4.5.1 Request, Response, and Descriptor-Fetch Threading In LL Mode For A Read Channel

For each read channel there are two threads (1) the request thread and (2) the response thread which are running in opposite directions. The DMA processes these threads independently. The request thread does not wait for the individual responses to arrive before issuing the next request. The request thread for the current channel i continues to issue requests until one of the following is true:

- The DMA exhausts its pool of available tags and the logic in the “Read Buffer” back-pres-sures the DMA read engine controller.
- The DMA completes its current DMA transfer block for the current channel. It has requested all of the data specified by the currently-loaded LL element.
- The current channel has issued Wi MRd requests, where Wi is the round robin weight associated with that channel.

A channel arbitration event now occurs and the DMA loads the context for the winning channel j. When the DMA is loading the new context, the response path for the previous channel i will be still utilizing the wire because the response thread lags the request thread. As soon as the DMA has loaded the context (and LL element if in LL mode) for the current channel j, it starts to issue requests. Therefore, in multichannel operation, when the block for channel i completes and a new LL element is been fetched, the DMA can be issuing requests belonging to other DMA channels. The currently inactive requester thread for channel i needs to synchronize with the response thread (that is, wait for all completions) before fetching its next LL element from local memory.

11.4.3 Memory Map and Register Definition

This section includes the PCIe module memory map and detailed descriptions of all registers.

NOTE

Synopsys Proprietary. Used with permission.

11.4.3.1 register descriptions

11.4.3.1.1 PCIe memory map

PCIe base address: 3380_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Device ID and Vendor ID Register. (TYPE1_DEV_ID_VEND_ID_REG)	32	RW	ABCD_16C3h
4h	Status and Command Register. (TYPE1_STATUS_COMMAND_REG)	32	RW	0010_0000h
8h	Class Code and Revision ID Register. (TYPE1_CLASS_CODE_REV_ID_REG)	32	RW	0000_0001h
Ch	Header Type, Latency Timer, and Cache Line Size Register. (TYPE1_HDR_TYPE_LAT_CACHE_LINE_SIZE_REG)	32	RW	0001_0000h
18h	Secondary Latency Timer, Subordinate Bus Number, Secondary Bus Number, and Primary Bus Number Register. (SEC_LAT_TIMER_SUB_BUS_SEC_BUS_PRI_BUS_REG)	32	RW	0000_0000h
1Ch	Secondary Status, and I/O Limit and Base Register. (SEC_STAT_IO_LIMIT_IO_BASE_REG)	32	RW	0000_0000h
20h	Memory Limit and Base Register. (MEM_LIMIT_MEM_BASE_REG)	32	RW	0000_0000h
24h	Prefetchable Memory Limit and Base Register. (PREF_MEM_LIMIT_PREF_MEM_BASE_REG)	32	RW	0000_0000h
28h	Prefetchable Base Upper 32 Bits Register. (PREF_BASE_UPPER_REG)	32	RO	0000_0000h
2Ch	Prefetchable Limit Upper 32 Bits Register. (PREF_LIMIT_UPPER_REG)	32	RO	0000_0000h
30h	I/O Limit and Base Upper 16 Bits Register. (IO_LIMIT_UPPER_IO_BASE_UPPER_REG)	32	RO	0000_0000h
34h	Capabilities Pointer Register. (TYPE1_CAP_PTR_REG)	32	RW	0000_0040h
38h	Expansion ROM Base Address Register. (TYPE1_EXP_ROM_BASE_REG)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
3Ch	Bridge Control, Interrupt Pin, and Interrupt Line Register. (BRIDGE_CTRL_INT_PIN_INT_LINE_REG)	32	RW	0000_01FFh
40h	Power Management Capabilities Register. (CAP_ID_NXT_PTR_REG)	32	RW	DBC3_5001h
44h	Power Management Control and Status Register. (CON_STATUS_REG)	32	RW	0000_0000h
50h	MSI Capability ID, Next Pointer, Capability/Control Registers. (PCI_MSI_CAP_ID_NEXT_CTRL_REG)	32	RW	0180_7005h
54h	MSI Message Lower Address Register. (MSI_CAP_OFF_04H_REG)	32	RW	0000_0000h
58h	For a 32 bit MSI Message, this register contains Data. (MSI_CAP_OFF_08H_REG)	32	RW	0000_0000h
5Ch	For a 64 bit MSI Message, this register contains Data. (MSI_CAP_OFF_0CH_REG)	32	RW	0000_0000h
60h	Used for MSI when Vector Masking Capable. (MSI_CAP_OFF_10H_REG)	32	RW	0000_0000h
64h	Used for MSI 64 bit messaging when Vector Masking Capable. (MSI_CAP_OFF_14H_REG)	32	RO	0000_0000h
70h	PCI Express Capabilities, ID, Next Pointer Register. (PCIE_CAP_ID_PCIE_NEXT_CAP_PTR_PCIE_CAP_REG)	32	RW	0042_0010h
74h	Device Capabilities Register. (DEVICE_CAPABILITIES_REG)	32	RW	0000_8000h
78h	Device Control and Status Register. (DEVICE_CONTROL_DEVICE_STATUS)	32	RW	0000_2010h
7Ch	Link Capabilities Register. (LINK_CAPABILITIES_REG)	32	RW	0073_CC12h
80h	Link Control and Status Register. (LINK_CONTROL_LINK_STATUS_REG)	32	RW	1011_0000h
84h	Slot Capabilities Register. (SLOT_CAPABILITIES_REG)	32	RW	0000_0000h
88h	Slot Control and Status Register. (SLOT_CONTROL_SLOT_STATUS)	32	RW	0040_03C0h
8Ch	Root Control and Capabilities Register. (ROOT_CONTROL_ROOT_CAPABILITIES_REG)	32	RW	0001_0000h
90h	Root Status Register. (ROOT_STATUS_REG)	32	W1C	0000_0000h
94h	Device Capabilities 2 Register. (DEVICE_CAPABILITIES2_REG)	32	RO	0000_041Fh
98h	Device Control 2 and Status 2 Register. (DEVICE_CONTROL2_DEVICE_STATUS2_REG)	32	RW	0000_0000h
9Ch	Link Capabilities 2 Register. (LINK_CAPABILITIES2_REG)	32	RO	0000_0006h
A0h	Link Control 2 and Status 2 Register. (LINK_CONTROL2_LINK_STATUS2_REG)	32	RW	0001_0002h
100h	Advanced Error Reporting Extended Capability Header. (AER_EXT_CAP_HDR_OFF)	32	RW	1482_0001h
104h	Uncorrectable Error Status Register. (UNCORR_ERR_STATUS_OFF)	32	W1C	0000_0000h
108h	Uncorrectable Error Mask Register. (UNCORR_ERR_MASK_OFF)	32	RW	0040_0000h
10Ch	Uncorrectable Error Severity Register. (UNCORR_ERR_SEV_OFF)	32	RW	0046_2030h
110h	Correctable Error Status Register. (CORR_ERR_STATUS_OFF)	32	W1C	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
114h	Correctable Error Mask Register. (CORR_ERR_MASK_OFF)	32	RW	0000_E000h
118h	Advanced Error Capabilities and Control Register. (ADV_ERR_CAP_CTRL_OFF)	32	RW	0000_00A0h
11Ch	Header Log Register 0. (HDR_LOG_0_OFF)	32	RO	0000_0000h
120h	Header Log Register 1. (HDR_LOG_1_OFF)	32	RO	0000_0000h
124h	Header Log Register 2. (HDR_LOG_2_OFF)	32	RO	0000_0000h
128h	Header Log Register 3. (HDR_LOG_3_OFF)	32	RO	0000_0000h
12Ch	Root Error Command Register. (ROOT_ERR_CMD_OFF)	32	RW	0000_0000h
130h	Root Error Status Register. (ROOT_ERR_STATUS_OFF)	32	RW	0000_0000h
134h	Error Source Identification Register. (ERR_SRC_ID_OFF)	32	RO	0000_0000h
138h	TLP Prefix Log Register 1. (TLP_PREFIX_LOG_1_OFF)	32	RO	0000_0000h
13Ch	TLP Prefix Log Register 2. (TLP_PREFIX_LOG_2_OFF)	32	RO	0000_0000h
140h	TLP Prefix Log Register 3. (TLP_PREFIX_LOG_3_OFF)	32	RO	0000_0000h
144h	TLP Prefix Log Register 4. (TLP_PREFIX_LOG_4_OFF)	32	RO	0000_0000h
158h	L1 Substates Extended Capability Header. (L1SUB_CAP_HEADER_REG)	32	RW	0001_001Eh
15Ch	L1 Substates Capability Register. (L1SUB_CAPABILITY_REG)	32	RW	0028_0A1Bh
160h	L1 Substates Control 1 Register. (L1SUB_CONTROL1_REG)	32	RW	0000_0A00h
164h	L1 Substates Control 2 Register. (L1SUB_CONTROL2_REG)	32	RW	0000_0028h
700h	Ack Latency Timer and Replay Timer Register. (ACK_LATENCY_TIMER_OFF)	32	RW	1846_0817h
704h	Vendor Specific DLLP Register. (VENDOR_SPEC_DLLP_OFF)	32	RW	FFFF_FFFFh
708h	Port Force Link Register. (PORT_FORCE_OFF)	32	RW	0000_0004h
70Ch	Ack Frequency and L0-L1 ASPM Control Register. (ACK_F_ASPM_CTRL_OFF)	32	RW	1B2C_2C00h
710h	Port Link Control Register. (PORT_LINK_CTRL_OFF)	32	RW	0001_0120h
714h	Lane Skew Register. (LANE_SKEW_OFF)	32	RW	0000_0000h
718h	Timer Control and Max Function Number Register. (TIMER_CTRL_MAX_FUNC_NUM_OFF)	32	RW	0000_C000h
71Ch	Symbol Timer Register and Filter Mask 1 Register. (SYMBOL_TIMER_FILTER_1_OFF)	32	RW	0000_0280h
720h	Filter Mask 2 Register. (FILTER_MASK_2_OFF)	32	RW	0000_0000h
724h	AMBA Multiple Outbound Decomposed NP SubRequests Control Register. (AMBA_MUL_OB_DECOMP_NP_SUB_REQ_CTRL_OFF)	32	RW	0000_0001h
728h	Debug Register 0 (PL_DEBUG0_OFF)	32	RO	Table 11-47
72Ch	Debug Register 1 (PL_DEBUG1_OFF)	32	RO	Table 11-47
730h	Transmit Posted FC Credit Status (TX_P_FC_CREDIT_STATUS_OFF)	32	RO	0000_0000h
734h	Transmit Non-Posted FC Credit Status (TX_NP_FC_CREDIT_STATUS_OFF)	32	RO	0000_0000h
738h	Transmit Completion FC Credit Status (TX_CPL_FC_CREDIT_STATUS_OFF)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
73Ch	Queue Status (QUEUE_STATUS_OFF)	32	RW	0000_0000h
740h	VC Transmit Arbitration Register 1 (VC_TX_ARBI_1_OFF)	32	RO	0000_000Fh
748h	Segmented-Buffer VC0 Posted Receive Queue Control. (VC0_P_RX_Q_CTRL_OFF)	32	RW	4520_C019h
74Ch	Segmented-Buffer VC0 Non-Posted Receive Queue Control. (VC0_NP_RX_Q_CTRL_OFF)	32	RW	0520_C003h
750h	Segmented-Buffer VC0 Completion Receive Queue Control. (VC0_CPL_RX_Q_CTRL_OFF)	32	RW	0580_0000h
80Ch	Link Width and Speed Change Control Register. (GEN2_CTRL_OFF)	32	RW	0002_012Ch
810h	PHY Status Register. (PHY_STATUS_OFF)	32	RO	Table 11-47
814h	PHY Control Register. (PHY_CONTROL_OFF)	32	RW	0000_0000h
81Ch	Programmable Target Map Control Register. (TRGT_MAP_CTRL_OFF)	32	RW	0000_007Fh
820h	Integrated MSI Reception Module (iMRM) Address Register. (MSI_CTRL_ADDR_OFF)	32	RW	0000_0000h
824h	Integrated MSI Reception Module Upper Address Register. (MSI_CTRL_UPPER_ADDR_OFF)	32	RW	0000_0000h
828h	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_0_EN_OFF)	32	RW	0000_0000h
82Ch	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_0_MASK_OFF)	32	RW	0000_0000h
830h	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_0_STATUS_OFF)	32	W1C	0000_0000h
834h	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_1_EN_OFF)	32	RW	0000_0000h
838h	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_1_MASK_OFF)	32	RW	0000_0000h
83Ch	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_1_STATUS_OFF)	32	W1C	0000_0000h
840h	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_2_EN_OFF)	32	RW	0000_0000h
844h	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_2_MASK_OFF)	32	RW	0000_0000h
848h	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_2_STATUS_OFF)	32	W1C	0000_0000h
84Ch	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_3_EN_OFF)	32	RW	0000_0000h
850h	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_3_MASK_OFF)	32	RW	0000_0000h
854h	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_3_STATUS_OFF)	32	W1C	0000_0000h
858h	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_4_EN_OFF)	32	RW	0000_0000h
85Ch	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_4_MASK_OFF)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
860h	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_4_STATUS_OFF)	32	W1C	0000_0000h
864h	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_5_EN_OFF)	32	RW	0000_0000h
868h	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_5_MASK_OFF)	32	RW	0000_0000h
86Ch	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_5_STATUS_OFF)	32	W1C	0000_0000h
870h	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_6_EN_OFF)	32	RW	0000_0000h
874h	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_6_MASK_OFF)	32	RW	0000_0000h
878h	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_6_STATUS_OFF)	32	W1C	0000_0000h
87Ch	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_7_EN_OFF)	32	RW	0000_0000h
880h	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_7_MASK_OFF)	32	RW	0000_0000h
884h	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_7_STATUS_OFF)	32	W1C	0000_0000h
888h	Integrated MSI Reception Module General Purpose IO Register. (MSI_GPIO_IO_OFF)	32	RW	0000_0000h
88Ch	RADM clock gating enable control register. (CLOCK_GATING_CTRL_OFF)	32	RW	0000_0001h
8B4h	Order Rule Control Register. (ORDER_RULE_CTRL_OFF)	32	RW	0000_0000h
8B8h	PIPE Loopback Control Register. (PIPE_LOOPBACK_CONTROL_OFF)	32	RW	0000_0001h
8BCh	DBI Read-Only Write Enable Register. (MISC_CONTROL_1_OFF)	32	RW	0000_0001h
8C0h	UpConfigure Multi-lane Control Register. (MULTI_LANE_CONTROL_OFF)	32	RW	0000_0000h
8C4h	PHY Interoperability Control Register. (PHY_INTEROP_CTRL_OFF)	32	RW	0000_0237h
8C8h	TRGT_CPL_LUT Delete Entry Control register. (TRGT_CPL_LUT_DELETE_ENTRY_OFF)	32	RW	0000_0000h
8CCh	Link Reset Request Flush Control Register. (LINK_FLUSH_CONTROL_OFF)	32	RW	FF00_0001h
8D0h	AXI Bridge Slave Error Response Register. (AMBA_ERROR_RESPONSE_DEFAULT_OFF)	32	RW	0000_9C00h
8D4h	Link Down AXI Bridge Slave Timeout Register. (AMBA_LINK_TIMEOUT_OFF)	32	RW	0000_0032h
8D8h	AMBA Ordering Control. (AMBA_ORDERING_CTRL_OFF)	32	RW	0000_0000h
8F0h	Lower 20 bits of the programmable AXI address where Messages coming from wire are mapped to. (AXI_MSTR_MSG_ADDR_LOW_OFF)	32	RW	0000_0000h
8F4h	Upper 32 bits of the programmable AXI address where Messages coming from wire are mapped to. (AXI_MSTR_MSG_ADDR_HIGH_OFF)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8F8h	PCIe Controller IIP Release Version Number. (PCIE_VERSION_NUMBER_OFF)	32	RO	3530_302Ah
8FCh	PCIe Controller IIP Release Version Type. (PCIE_VERSION_TYPE_OFF)	32	RO	6761_2A2Ah
B40h	Auxiliary Clock Frequency Control Register. (AUX_CLK_FREQ_OFF)	32	RW	0000_000Ah
B44h	L1 Substates Timing Register. (L1_SUBSTATES_OFF)	32	RW	0000_00D2h
30_0000h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_0)	32	RW	0000_0000h
30_0004h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_0)	32	RW	0000_0000h
30_0008h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
30_000Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
30_0010h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_0)	32	RW	0000_FFFFh
30_0014h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
30_0018h	iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
30_0100h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_0)	32	RW	0000_0000h
30_0104h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_0)	32	RW	0000_0000h
30_0108h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
30_010Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
30_0110h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_0)	32	RW	0000_FFFFh
30_0114h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
30_0200h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_1)	32	RW	0000_0000h
30_0204h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_1)	32	RW	0000_0000h
30_0208h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_1)	32	RW	0000_0000h
30_020Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_1)	32	RW	0000_0000h
30_0210h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_1)	32	RW	0000_FFFFh
30_0214h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_1)	32	RW	0000_0000h
30_0218h	iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_1)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
30_0300h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_1)	32	RW	0000_0000h
30_0304h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_1)	32	RW	0000_0000h
30_0308h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_1)	32	RW	0000_0000h
30_030Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_1)	32	RW	0000_0000h
30_0310h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_1)	32	RW	0000_FFFFh
30_0314h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_1)	32	RW	0000_0000h
30_0400h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_2)	32	RW	0000_0000h
30_0404h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_2)	32	RW	0000_0000h
30_0408h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_2)	32	RW	0000_0000h
30_040Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_2)	32	RW	0000_0000h
30_0410h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_2)	32	RW	0000_FFFFh
30_0414h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_2)	32	RW	0000_0000h
30_0418h	iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_2)	32	RW	0000_0000h
30_0500h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_2)	32	RW	0000_0000h
30_0504h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_2)	32	RW	0000_0000h
30_0508h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_2)	32	RW	0000_0000h
30_050Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_2)	32	RW	0000_0000h
30_0510h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_2)	32	RW	0000_FFFFh
30_0514h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_2)	32	RW	0000_0000h
30_0600h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_3)	32	RW	0000_0000h
30_0604h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_3)	32	RW	0000_0000h
30_0608h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_3)	32	RW	0000_0000h
30_060Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_3)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
30_0610h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_3)	32	RW	0000_FFFFh
30_0614h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_3)	32	RW	0000_0000h
30_0618h	iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_3)	32	RW	0000_0000h
30_0700h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_3)	32	RW	0000_0000h
30_0704h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_3)	32	RW	0000_0000h
30_0708h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_3)	32	RW	0000_0000h
30_070Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_3)	32	RW	0000_0000h
30_0710h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_3)	32	RW	0000_FFFFh
30_0714h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_3)	32	RW	0000_0000h
38_0000h	DMA Arbitration Scheme for TRGT1 Interface. (DMA_CTRL_DATA_ARB_PRIOR_OFF)	32	RW	0000_0688h
38_0008h	DMA Number of Channels Register. (DMA_CTRL_OFF)	32	RW	0001_0001h
38_000Ch	DMA Write Engine Enable Register. (DMA_WRITE_ENGINE_EN_OFF)	32	RW	0000_0000h
38_0010h	DMA Write Doorbell Register. (DMA_WRITE_DOORBELL_OFF)	32	RW	0000_0000h
38_0018h	DMA Write Engine Channel Arbitration Weight Low Register. (DMA_WRITE_CHANNEL_ARB_WEIGHT_LOW_OFF)	32	RW	0000_8421h
38_001Ch	DMA Write Engine Channel Arbitration Weight High Register. (DMA_WRITE_CHANNEL_ARB_WEIGHT_HIGH_OFF)	32	RW	0000_8421h
38_002Ch	DMA Read Engine Enable Register. (DMA_READ_ENGINE_EN_OFF)	32	RW	0000_0000h
38_0030h	DMA Read Doorbell Register. (DMA_READ_DOORBELL_OFF)	32	RW	0000_0000h
38_0038h	DMA Read Engine Channel Arbitration Weight Low Register. (DMA_READ_CHANNEL_ARB_WEIGHT_LOW_OFF)	32	RW	0000_8421h
38_003Ch	DMA Read Engine Channel Arbitration Weight High Register. (DMA_READ_CHANNEL_ARB_WEIGHT_HIGH_OFF)	32	RW	0000_8421h
38_004Ch	DMA Write Interrupt Status Register. (DMA_WRITE_INT_STATUS_OFF)	32	RW	0000_0000h
38_0054h	DMA Write Interrupt Mask Register. (DMA_WRITE_INT_MASK_OFF)	32	RW	0001_0001h
38_0058h	DMA Write Interrupt Clear Register. (DMA_WRITE_INT_CLEAR_OFF)	32	W1C	0000_0000h
38_005Ch	DMA Write Error Status Register (DMA_WRITE_ERR_STATUS_OFF)	32	RO	0000_0000h
38_0060h	DMA Write Done IMWr Address Low Register. (DMA_WRITE_DONE_IMWR_LOW_OFF)	32	RW	0000_0000h
38_0064h	DMA Write Done IMWr Interrupt Address High Register. (DMA_WRITE_DONE_IMWR_HIGH_OFF)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
38_0068h	DMA Write Abort IMWr Address Low Register. (DMA_WRITE_ABORT_IMWR_LOW_OFF)	32	RW	0000_0000h
38_006Ch	DMA Write Abort IMWr Address High Register. (DMA_WRITE_ABORT_IMWR_HIGH_OFF)	32	RW	0000_0000h
38_0070h	DMA Write Channel 1 and 0 IMWr Data Register. (DMA_WRITE_CH01_IMWR_DATA_OFF)	32	RW	0000_0000h
38_0074h	DMA Write Channel 3 and 2 IMWr Data Register. (DMA_WRITE_CH23_IMWR_DATA_OFF)	32	RW	0000_0000h
38_0078h	DMA Write Channel 5 and 4 IMWr Data Register. (DMA_WRITE_CH45_IMWR_DATA_OFF)	32	RW	0000_0000h
38_007Ch	DMA Write Channel 7 and 6 IMWr Data Register. (DMA_WRITE_CH67_IMWR_DATA_OFF)	32	RW	0000_0000h
38_0090h	DMA Write Linked List Error Enable Register. (DMA_WRITE_LINKED_LIST_ERR_EN_OFF)	32	RW	0000_0000h
38_00A0h	DMA Read Interrupt Status Register. (DMA_READ_INT_STATUS_OFF)	32	RW	0000_0000h
38_00A8h	DMA Read Interrupt Mask Register. (DMA_READ_INT_MASK_OFF)	32	RW	0001_0001h
38_00ACh	DMA Read Interrupt Clear Register. (DMA_READ_INT_CLEAR_OFF)	32	WO	0000_0000h
38_00B4h	DMA Read Error Status Low Register. (DMA_READ_ERR_STATUS_LOW_OFF)	32	RO	0000_0000h
38_00B8h	DMA Read Error Status High Register. (DMA_READ_ERR_STATUS_HIGH_OFF)	32	RO	0000_0000h
38_00C4h	DMA Read Linked List Error Enable Register. (DMA_READ_LINKED_LIST_ERR_EN_OFF)	32	RW	0000_0000h
38_00CCh	DMA Read Done IMWr Address Low Register. (DMA_READ_DONE_IMWR_LOW_OFF)	32	RW	0000_0000h
38_00D0h	DMA Read Done IMWr Address High Register. (DMA_READ_DONE_IMWR_HIGH_OFF)	32	RW	0000_0000h
38_00D4h	DMA Read Abort IMWr Address Low Register. (DMA_READ_ABORT_IMWR_LOW_OFF)	32	RW	0000_0000h
38_00D8h	DMA Read Abort IMWr Address High Register. (DMA_READ_ABORT_IMWR_HIGH_OFF)	32	RW	0000_0000h
38_00DCh	DMA Read Channel 1 and 0 IMWr Data Register. (DMA_READ_CH01_IMWR_DATA_OFF)	32	RW	0000_0000h
38_00E0h	DMA Read Channel 3 and 2 IMWr Data Register. (DMA_READ_CH23_IMWR_DATA_OFF)	32	RW	0000_0000h
38_00E4h	DMA Read Channel 5 and 4 IMWr Data Register. (DMA_READ_CH45_IMWR_DATA_OFF)	32	RW	0000_0000h
38_00E8h	DMA Read Channel 7 and 6 IMWr Data Register. (DMA_READ_CH67_IMWR_DATA_OFF)	32	RW	0000_0000h
38_0200h	DMA Write Channel Control 1 Register. (DMA_CH_CONTROL1_OFF_WRCH_0)	32	RW	0000_0000h
38_0208h	DMA Write Transfer Size Register. (DMA_TRANSFER_SIZE_OFF_WRCH_0)	32	RW	0000_0000h
38_020Ch	DMA Write SAR Low Register. (DMA_SAR_LOW_OFF_WRCH_0)	32	RW	0000_0000h

Table continues on the next page...

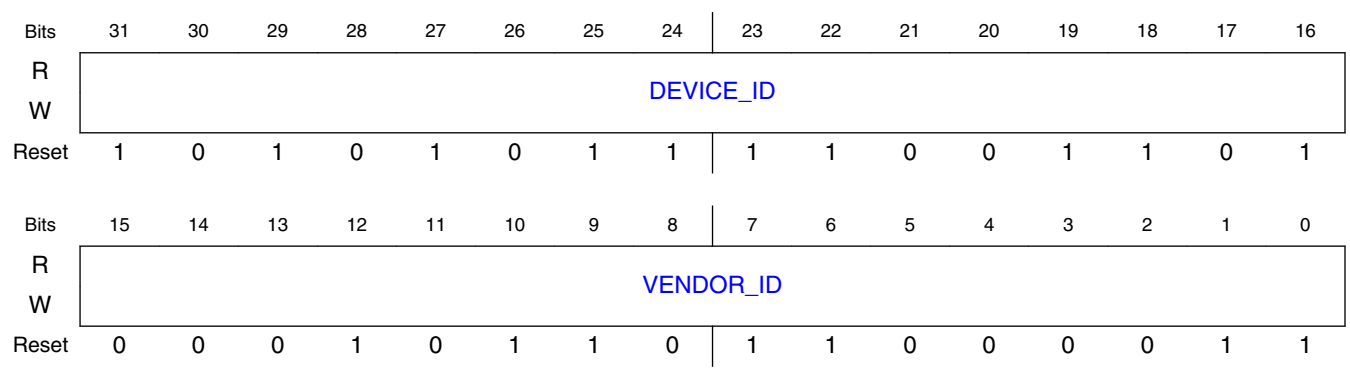
Offset	Register	Width (In bits)	Access	Reset value
38_0210h	DMA Write SAR High Register. (DMA_SAR_HIGH_OFF_WRCH_0)	32	RW	0000_0000h
38_0214h	DMA Write DAR Low Register. (DMA_DAR_LOW_OFF_WRCH_0)	32	RW	0000_0000h
38_0218h	DMA Write DAR High Register. (DMA_DAR_HIGH_OFF_WRCH_0)	32	RW	0000_0000h
38_021Ch	DMA Write Linked List Pointer Low Register. (DMA_LLP_LOW_OFF_WRCH_0)	32	RW	0000_0000h
38_0220h	DMA Write Linked List Pointer High Register. (DMA_LLP_HIGH_OFF_WRCH_0)	32	RW	0000_0000h
38_0300h	DMA Read Channel Control 1 Register. (DMA_CH_CONTROL1_OFF_RDCH_0)	32	RW	0000_0000h
38_0308h	DMA Read Transfer Size Register. (DMA_TRANSFER_SIZE_OFF_RDCH_0)	32	RW	0000_0000h
38_030Ch	DMA Read SAR Low Register. (DMA_SAR_LOW_OFF_RDCH_0)	32	RW	0000_0000h
38_0310h	DMA Read SAR High Register. (DMA_SAR_HIGH_OFF_RDCH_0)	32	RW	0000_0000h
38_0314h	DMA Read DAR Low Register. (DMA_DAR_LOW_OFF_RDCH_0)	32	RW	0000_0000h
38_0318h	DMA Read DAR High Register. (DMA_DAR_HIGH_OFF_RDCH_0)	32	RW	0000_0000h
38_031Ch	DMA Read Linked List Pointer Low Register. (DMA_LLP_LOW_OFF_RDCH_0)	32	RW	0000_0000h
38_0320h	DMA Read Linked List Pointer High Register. (DMA_LLP_HIGH_OFF_RDCH_0)	32	RW	0000_0000h

11.4.3.1.2 Device ID and Vendor ID Register. (TYPE1_DEV_ID_VEND_ID_REG)

11.4.3.1.2.1 Offset

Register	Offset
TYPE1_DEV_ID_VEND_ID_REG	0h

11.4.3.1.2.2 Diagram



11.4.3.1.2.3 Fields

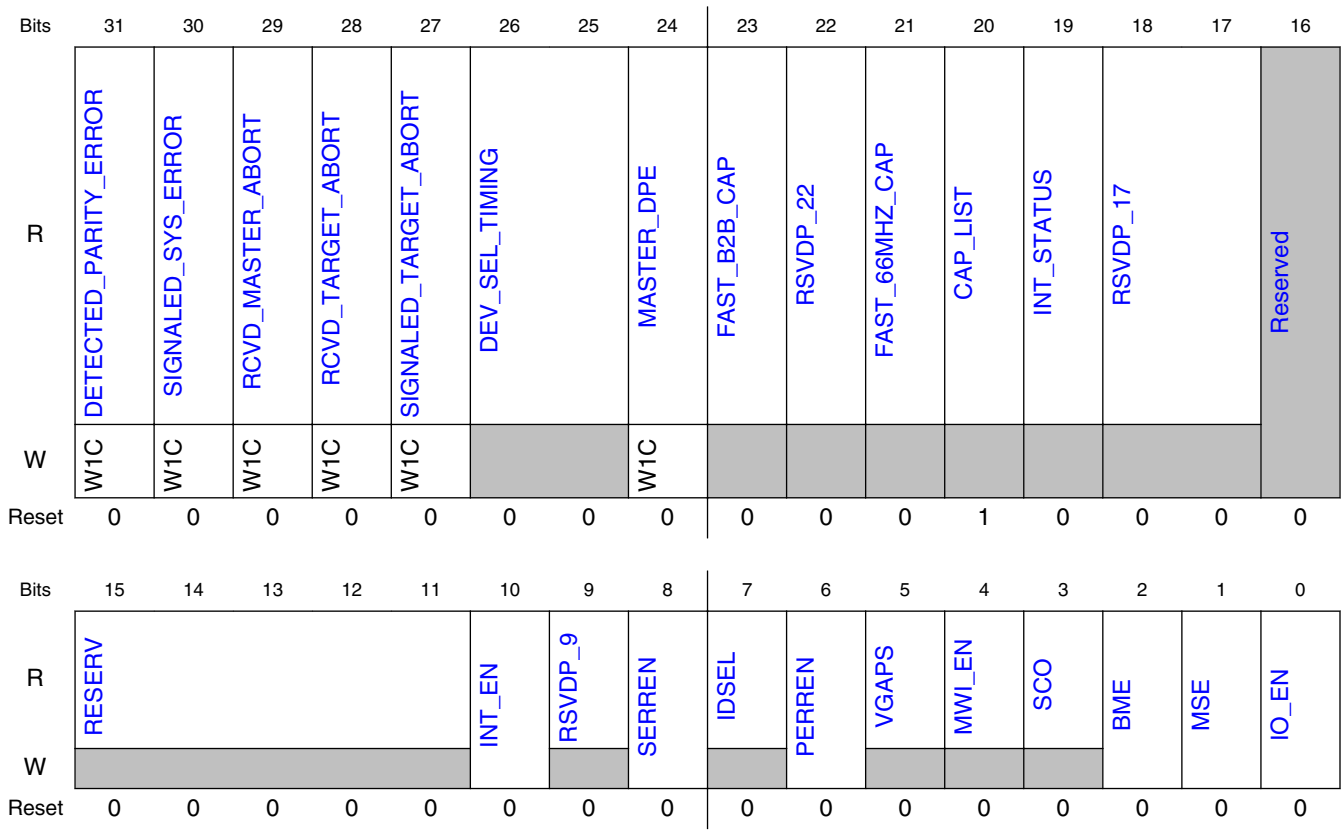
Field	Function
31-16 DEVICE_ID	Device ID. The Device ID register identifies the particular Function. This identifier is allocated by the vendor. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15-0 VENDOR_ID	Vendor ID. The Vendor ID register identifies the manufacturer of the Function. Valid vendor identifiers are allocated by the PCI-SIG to ensure uniqueness. It is not permitted to populate this register with a value of FFFFh, which is an invalid value for Vendor ID. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

11.4.3.1.3 Status and Command Register. (TYPE1_STATUS_COMMAND_REG)

11.4.3.1.3.1 Offset

Register	Offset
TYPE1_STATUS_COMMAND_REG	4h

11.4.3.1.3.2 Diagram



11.4.3.1.3.3 Fields

Field	Function
31 DETECTED_PARITY_ERROR	Detected Parity Error. This bit is set by a Function whenever it receives a Poisoned TLP, regardless of the state the Parity Error Response bit in the Command register. The bit is set when the Poisoned TLP is received by a Function's primary side.
30 SIGNALED_SYS_ERROR	Signaled System Error. This bit is set when a Function sends an ERR_FATAL or ERR_NONFATAL Message, and the SERR# Enable bit in the Command register is 1b.
29 RCVD_MASTER_ABORT	Received Master Abort. This bit is set when a Requester receives a Completion with Unsupported Request Completion status. The bit is set when the Unsupported Request is received by a Function's primary side.
28 RCVD_TARGET_ABORT	Received Target Abort. This bit is set when a Requester receives a Completion with Completer Abort Completion status. The bit is set when the Completer Abort is received by a Function's primary side.
27 SIGNALED_TARGET_ABORT	Signaled Target Abort. This bit is set when a Function completes a Posted or Non-Posted Request as a Completer Abort error. This applies to a Function when the Completer Abort was generated by its primary side.

Table continues on the next page...

Field	Function
26-25 DEV_SEL_TIMING	DEVSEL Timing. This field was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires it to 00b.
24 MASTER_DPE	Master Data Parity Error. This bit is set by a Function if the Parity Error Response bit in the Command register is 1b and either of the following two conditions occurs: - Port receives a Poisoned Completion going downstream - Port transmits a Poisoned Request upstream If the Parity Error Response bit is 0b, this bit is never set.
23 FAST_B2B_CAP	Fast Back-to-Back Transactions Capable. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
22 RSVDP_22	Reserved for future use.
21 FAST_66MHZ_CAP	66 MHz Capable. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
20 CAP_LIST	Capabilities List. Indicates the presence of an Extended Capability list item. Since all PCI Express device Functions are required to implement the PCI Express Capability structure, the controller hardwires this bit to 1b.
19 INT_STATUS	Interrupt Status. When set, indicates that an INTx emulation interrupt is pending internally in the Function. INTx emulation interrupts forwarded by Functions from the secondary side are not reflected in this bit. Setting the Interrupt Disable bit has no effect on the state of this bit. For Functions that do not generate INTx interrupts, the controller hardwires this bit to 0b.
18-17 RSVDP_17	Reserved for future use.
16 —	Reserved
15-11 RESERV	Reserved
10 INT_EN	Interrupt Disable. Controls the ability of a Function to generate INTx emulation interrupts. When set, Functions are prevented from asserting INTx interrupts. Note: - Any INTx emulation interrupts already asserted by the Function must be deasserted when this bit is set. INTx interrupts use virtual wires that must, if asserted, be deasserted using the appropriate Deassert_INTx message(s) when this bit is set. - Only the INTx virtual wire interrupt(s) associated with the Function(s) for which this bit is set are affected. - For Functions that generate INTx interrupts on their own behalf, this bit is required. This bit has no effect on interrupts forwarded from the secondary side. For Functions that do not generate INTx interrupts on their own behalf this bit is optional. If this bit is not implemented, the controller hardwires it to 0b.
9 RSVDP_9	Reserved for future use.
8 SERREN	SERR# Enable. When set, this bit enables reporting upstream of Non-fatal and Fatal errors detected by the Function. Note: The errors are reported if enabled either through this bit or through the PCI Express specific bits in the Device Control register. For more details see the "Error Registers" section of the PCI Express Specification. In addition, this bit controls transmission by the primary interface of ERR_NONFATAL and ERR_FATAL error Messages forwarded from the secondary interface. This bit does not affect the transmission of forwarded ERR_COR messages.
7 IDSEL	IDSEL Stepping/Wait Cycle Control. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.

Table continues on the next page...

Field	Function
6 PERREN	Parity Error Response. This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Status register. For more details see the "Error Registers" section of the PCI Express Specification.
5 VGAPS	VGA Palette Snoop. This bit was originally described in the PCI Local Bus Specification and the PCI-to-PCI Bridge Architecture Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
4 MWI_EN	Memory Write and Invalidate. This bit was originally described in the PCI Local Bus Specification and the PCI-to-PCI Bridge Architecture Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b. For PCI Express to PCI/PCI-X Bridges, refer to the PCI Express to PCI/PCI-X Bridge Specification for requirements for this register.
3 SCO	Special Cycle Enable. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
2 BME	Bus Master Enable. This bit controls forwarding of Memory or I/O requests by a port in the Upstream direction. When this bit is 0b, Memory and I/O Requests received at a Root Port must be handled as Unsupported Requests (UR) For Non-Posted Requests a Completion with UR completion status must be returned. This bit does not affect forwarding of Completions in either the Upstream or Downstream direction. The forwarding of Requests other than Memory or I/O Requests is not controlled by this bit.
1 MSE	Memory Space Enable. This bit controls a Function's response to Memory Space accesses received on its primary side. - When set, the Function is enabled to decode the address and further process Memory Space accesses. - When clear, all received Memory Space accesses are caused to be handled as Unsupported Requests. You cannot write to this register if your configuration has no MEM bars; that is, the internal signal has_mem_bar =0. Note: The access attributes of this field are as follows: - Dbi: ! has_mem_bar ? RO : RW
0 IO_EN	IO Space Enable. This bit controls a Function's response to I/O Space accesses received on its primary side. - When set, the Function is enabled to decode the address and further process I/O Space accesses. - When clear, all received I/O accesses are caused to be handled as Unsupported Requests. You cannot write to this register if your configuration has no IO bars; that is, the internal signal has_io_bar =0. Note: The access attributes of this field are as follows: - Dbi: !has_io_bar ? RO : RW

11.4.3.1.4 Class Code and Revision ID Register. (TYPE1_CLASS_CODE_REV_ID_REG)

11.4.3.1.4.1 Offset

Register	Offset
TYPE1_CLASS_CODE_REV_ID_REG	8h

11.4.3.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_CLASS_CODE								SUBCLASS_CODE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PROGRAM_INTERFACE								REVISION_ID							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

11.4.3.1.4.3 Fields

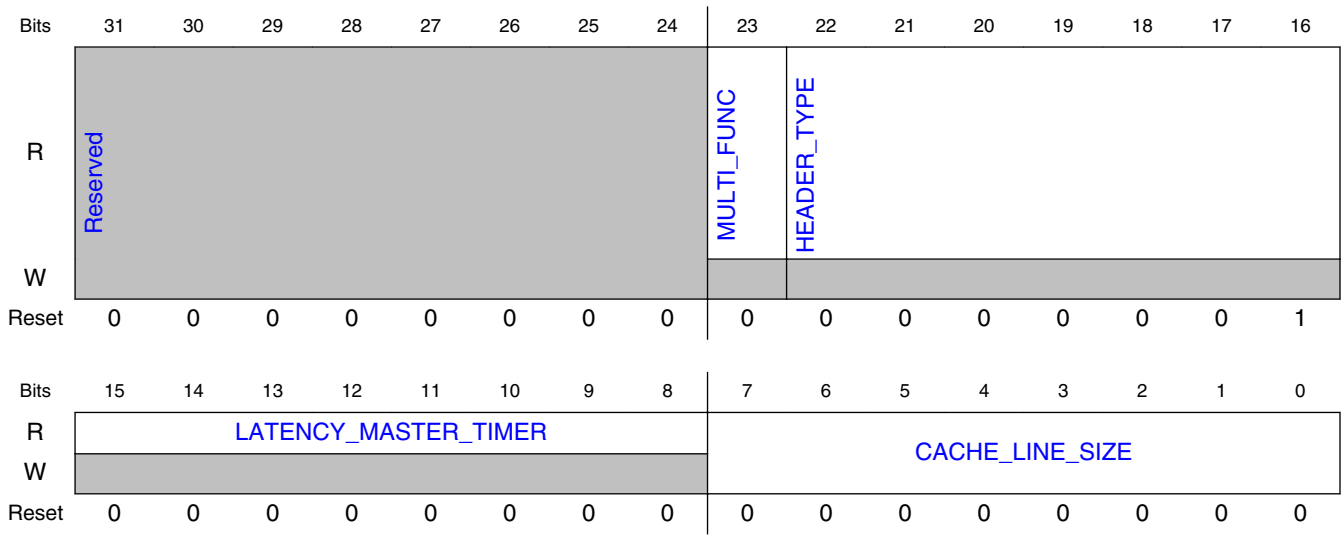
Field	Function
31-24 BASE_CLASS_CODE	Base Class Code. A code that broadly classifies the type of operation the Function performs. Encodings for base class, are provided in the PCI Code and ID Assignment Specification. All unspecified encodings are reserved. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
23-16 SUBCLASS_CODE	Sub-Class Code. Specifies a base class sub-class, which identifies more specifically the operation of the Function. Encodings for sub-class are provided in the PCI Code and ID Assignment Specification. All unspecified encodings are reserved. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15-8 PROGRAM_INTERFACE	Programming Interface. This field identifies a specific register level programming interface (if any) so that device independent software can interact with the Function. Encodings for interface are provided in the PCI Code and ID Assignment Specification. All unspecified encodings are reserved. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
7-0 REVISION_ID	Revision ID. The value of this field specifies a Function specific revision identifier. The value is chosen by the vendor. Zero is an acceptable value. The Revision ID should be viewed as a vendor defined extension to the Device ID. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

11.4.3.1.5 Header Type, Latency Timer, and Cache Line Size Register. (TYPE1_HDR_TYPE_LAT_CACHE_LINE_SIZE_REG)

11.4.3.1.5.1 Offset

Register	Offset
TYPE1_HDR_TYPE_LAT_CACHE_LINE_SIZE_REG	Ch

11.4.3.1.5.2 Diagram



11.4.3.1.5.3 Fields

Field	Function
31-24 —	Reserved
23 MULTI_FUNC	Multi-Function Device. - When set, indicates that the device may contain multiple Functions, but not necessarily. Software is permitted to probe for Functions other than Function 0. - When clear, software must not probe for Functions other than Function 0 unless explicitly indicated by another mechanism, such as an ARI or SR-IOV Capability structure. Except where stated otherwise, it is recommended that this bit be set if there are multiple Functions, and clear if there is only one Function. Note: This register field is sticky.
22-16 HEADER_TYPE	Header Layout. This field identifies the layout of the second part of the predefined header. The controller uses 000 0001b encoding. The encoding 000 0010b is reserved. This encoding was originally described in the PC Card Standard Electrical Specification and is used in previous versions of the programming model. Careful consideration should be given to any attempt to repurpose it.
15-8 LATENCY_MASTER_TIMER	Latency Timer. This register is also referred to as Primary Latency Timer. The Latency Timer was originally described in the PCI Local Bus Specification and the PCI-to-PCI Bridge Architecture Specification. Its functionality does not apply to PCI Express. The controller hardwires this register to 00h.
7-0 CACHE_LINE_SIZE	Cache Line Size. The Cache Line Size register is programmed by the system firmware or the operating system to system cache line size. However, legacy conventional PCI software may not always be able to program this register correctly especially in the case of Hot-Plug devices. This read-write register is implemented for legacy compatibility purposes but has no effect on any PCI Express device behavior.

11.4.3.1.6 Secondary Latency Timer, Subordinate Bus Number, Secondary Bus Number, and Primary Bus Number Register. (SEC_LAT_TIMER_SUB_BUS_SEC_BUS_PRI_BUS_REG)

11.4.3.1.6.1 Offset

Register	Offset
SEC_LAT_TIMER_SUB_BUS_SEC_BUS_PRI_BUS_REG	18h

11.4.3.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SEC_LAT_TIMER								SUB_BUS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SEC_BUS								PRIM_BUS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.6.3 Fields

Field	Function
31-24 SEC_LAT_TIMER	Secondary Latency Timer. This register does not apply to PCI Express. The controller hardwires it to 00h.
23-16 SUB_BUS	Subordinate Bus Number. The Subordinate Bus Number register is used to record the bus number of the highest numbered PCI bus segment which is behind (or subordinate to) the bridge. Configuration software programs the value in this register. The bridge uses this register in conjunction with the Secondary Bus Number register to determine when to respond to and pass on a Type 1 configuration transaction on the primary interface to the secondary interface.
15-8 SEC_BUS	Secondary Bus Number. The Secondary Bus Number register is used to record the bus number of the PCI bus segment to which the secondary interface of the bridge is connected. Configuration software programs the value in this register. The bridge uses this register to determine when to respond to and convert a Type 1 configuration transaction on the primary interface into a Type 0 transaction on the secondary interface.
7-0 PRIM_BUS	Primary Bus Number. This register is not used by PCI Express Functions. It is implemented for compatibility with legacy software.

11.4.3.1.7 Secondary Status, and I/O Limit and Base Register. (SEC_STAT_IO_LIMIT_IO_BASE_REG)

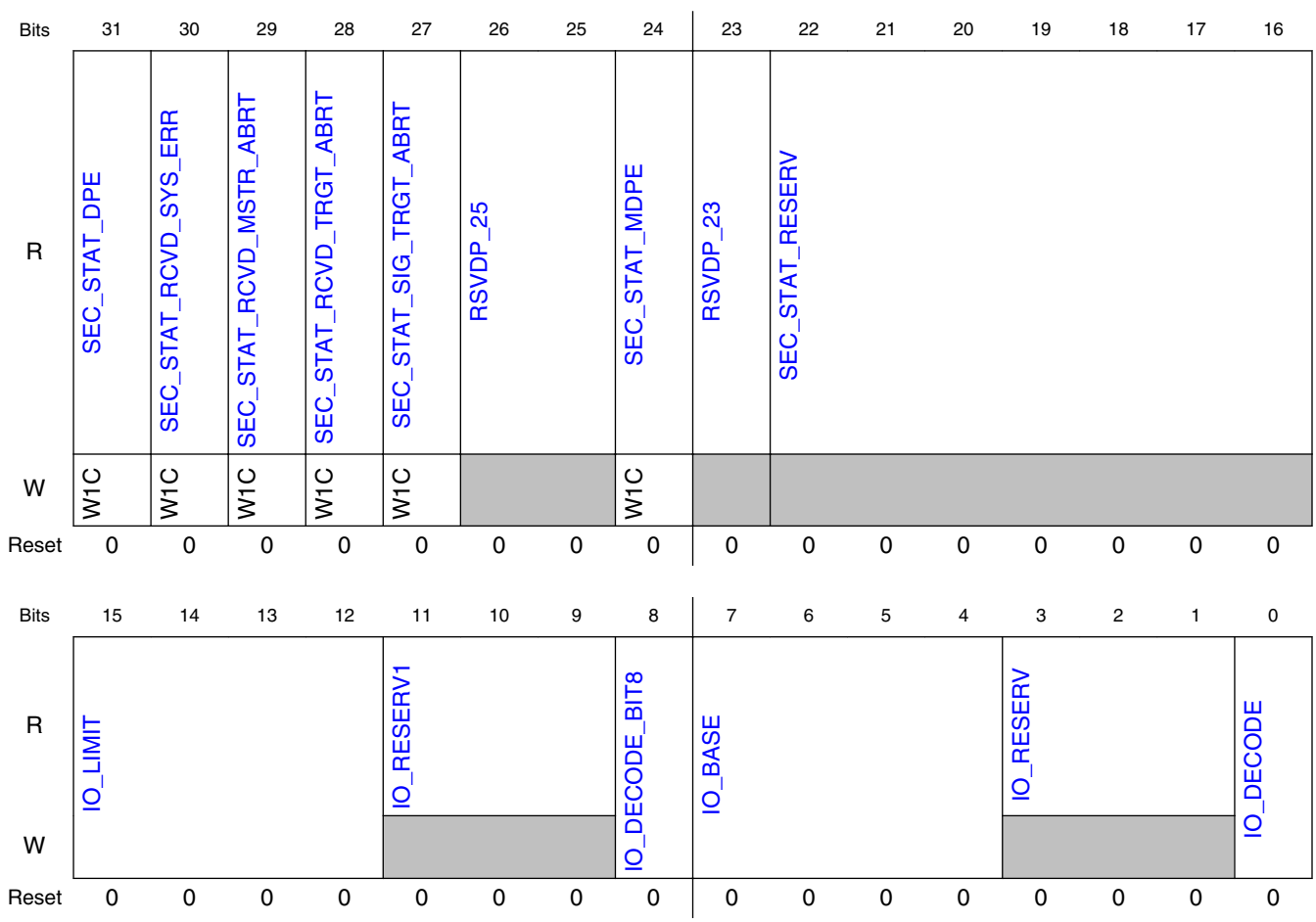
11.4.3.1.7.1 Offset

Register	Offset
SEC_STAT_IO_LIMIT_IO_BASE_REG	1Ch

11.4.3.1.7.2 Function

Secondary Status, and I/O Limit and Base Register. The I/O Limit and I/O Base registers are optional and define an address range that is used by the bridge to determine when to forward I/O transactions from one interface to the other. If a bridge does not implement an I/O address range, then both the I/O Limit and I/O Base registers must be implemented as read-only registers that return zero when read. If a bridge supports an I/O address range, then these registers must be initialized by configuration software so default states are not specified.

11.4.3.1.7.3 Diagram



11.4.3.1.7.4 Fields

Field	Function
31 SEC_STAT_DPE	Detected Parity Error. This bit is set by a Function when a Poisoned TLP is received by its secondary side, regardless of the state the Parity Error Response Enable bit in the Bridge Control register.
30 SEC_STAT_RCVD_SYS_ERR	Received System Error. This bit is set when the secondary side of a Function receives an ERR_FATAL or ERR_NONFATAL message.
29 SEC_STAT_RCVD_MSTR_ABRT	Received Master Abort. This bit is set when the secondary side of a Function (for requests initiated by the Type 1 header Function itself) receives a Completion with Unsupported Request Completion status.
28 SEC_STAT_RCVD_TRGT_ABRT	Received Target Abort. This bit is set when the secondary side of a Function (for requests initiated by the Type 1 header Function itself) receives a Completion with Completer Abort Completion status.
27 SEC_STAT_SIG_TRGT_ABRT	Signaled Target Abort. This bit is set when the secondary side of the Function (for Requests completed by the Type 1 header Function itself) completes a Posted or Non-Posted request as a Completer Abort error.
26-25 RSVDP_25	Reserved for future use.
24 SEC_STAT_MDPE	Master Data Parity Error. This bit is set by a Function if the Parity Error Response Enable bit in the Bridge Control register is set, and either of the following two conditions occurs: - Port receives a Poisoned Completion coming Upstream - Port transmits a Poisoned Request Downstream If the Parity Error Response Enable bit is clear, this bit is never set.
23 RSVDP_23	Reserved for future use.
22-16 SEC_STAT_RESERV	Reserved
15-12 IO_LIMIT	I/O Limit Address. These bits correspond to the address[15:12] of IO address range. For the purpose of address decoding, the bridge assumes that the lower 12 address bits, address[11:0], of the I/O limit address (not implemented in the I/O Limit register) are FFFh. The I/O Limit register can be programmed to a smaller value than the I/O Base register, if there are no I/O addresses on the secondary side of the bridge. In this case, the bridge will not forward any I/O transactions from the primary bus to the secondary and will forward all I/O transactions from the secondary bus to the primary bus.
11-9 IO_RESERV1	Reserved
8 IO_DECODE_BIT8	I/O Addressing Encode (IO Limit Address). This bit encodes the IO addressing capability of the bridge. IO_DECODE_BIT8 indicates the following: - 0h: The bridge supports only 16-bit I/O addressing (for ISA compatibility). For the purpose of address decoding, the bridge assumes that the upper 16 address bits, Address[31:16], of the I/O limit address (not implemented in I/O Limit register) are zero. Note: The bridge must still perform a full 32-bit decode of the I/O address (that is, check that Address[31:16] are 0000h). In this case, the I/O address range supported by the bridge will be restricted to the first 64 KB of I/O Space (0000 0000h to 0000 FFFFh). - 01h: The bridge supports 32-bit I/O address decoding, and the I/O Limit Upper 16 Bits hold the upper 16 bits, corresponding to Address[31:16], of the 32-bit Limit address. In this

Table continues on the next page...

Field	Function
	case, system configuration software is permitted to locate the I/O address range supported by the bridge anywhere in the 4-GB I/O Space. Note: The 4-KB alignment and granularity restrictions still apply when the bridge supports 32-bit I/O addressing. Note: The access attributes of this field are as follows: - Dbi: R
7-4 IO_BASE	I/O Base Address. These bits correspond to the address[15:12] of IO address range. For the purpose of address decoding, the bridge assumes that the lower 12 address bits, address[11:0], of the I/O base address (not implemented in the I/O Base register) are zero.
3-1 IO_RESERV	Reserved
0 IO_DECODE	I/O Addressing Encode (IO Base Address) This bit encodes the IO addressing capability of the bridge. IO_DECODE indicates the following: - 0h: The bridge supports only 16-bit I/O addressing (for ISA compatibility). For the purpose of address decoding, the bridge assumes that the upper 16 address bits, Address[31:16], of the I/O base address (not implemented in I/O base register) are zero. Note: The bridge must still perform a full 32-bit decode of the I/O address (that is, check that Address[31:16] are 0000h). In this case, the I/O address range supported by the bridge will be restricted to the first 64 KB of I/O Space (0000 0000h to 0000 FFFFh). - 01h: The bridge supports 32-bit I/O address decoding, and the I/O Base Upper 16 Bits hold the upper 16 bits, corresponding to Address[31:16], of the 32-bit Base address. In this case, system configuration software is permitted to locate the I/O address range supported by the bridge anywhere in the 4-GB I/O Space. Note: The 4-KB alignment and granularity restrictions still apply when the bridge supports 32-bit I/O addressing. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

11.4.3.1.8 Memory Limit and Base Register. (MEM_LIMIT_MEM_BASE_REG)

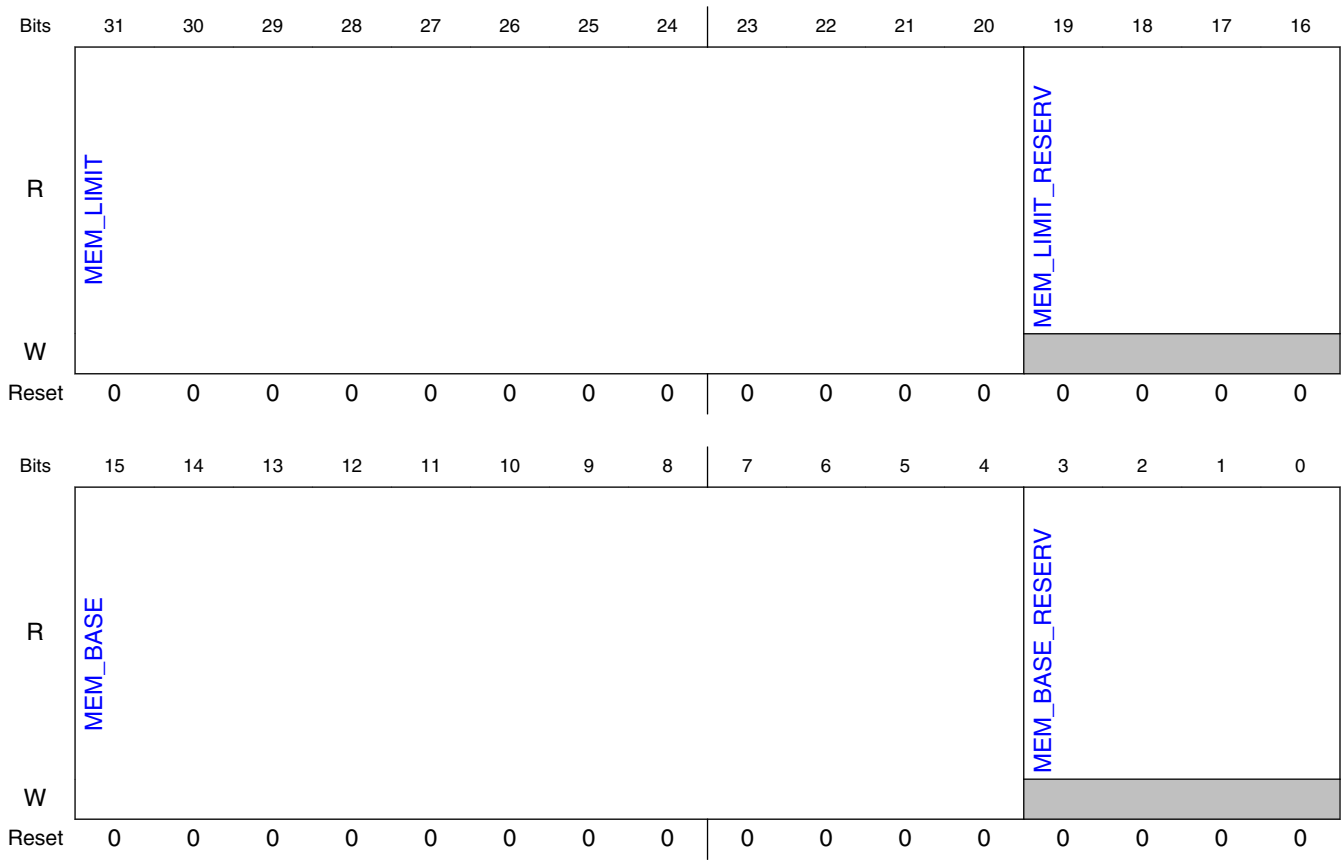
11.4.3.1.8.1 Offset

Register	Offset
MEM_LIMIT_MEM_BASE_REG	20h

11.4.3.1.8.2 Function

Memory Limit and Base Register. The Memory Limit and Memory Base registers define a memory mapped address range which is used by the bridge to determine when to forward memory transactions from one interface to the other. If there is no prefetchable memory space, and there is no memory-mapped space on the secondary side of the bridge, then the bridge will not forward any memory transactions from the primary bus to the secondary bus and will forward all memory transactions from the secondary bus to the primary bus.

11.4.3.1.8.3 Diagram



11.4.3.1.8.4 Fields

Field	Function
31-20 MEM_LIMIT	Memory Limit Address. These bits correspond to the upper 12 address bits, Address[31:20], of 32-bit addresses. For the purpose of address decoding, the bridge assumes that the lower 20 address bits, Address[19:0], of the memory limit address (not implemented in the Memory Limit register) are F FFFFh. The Memory Limit register must be programmed to a smaller value than the Memory Base register if there is no memory-mapped address space on the secondary side of the bridge.
19-16 MEM_LIMIT_RESERV	Reserved
15-4 MEM_BASE	Memory Base Address. These bits correspond to the upper 12 address bits, Address[31:20], of 32-bit addresses. For the purpose of address decoding, the bridge assumes that the lower 20 address bits, Address[19:0], of the memory base address (not implemented in the Memory Base register) are zero.
3-0 MEM_BASE_RESERV	Reserved

11.4.3.1.9 Prefetchable Memory Limit and Base Register. (PREF_MEM_LIMIT_PREF_MEM_BASE_REG)

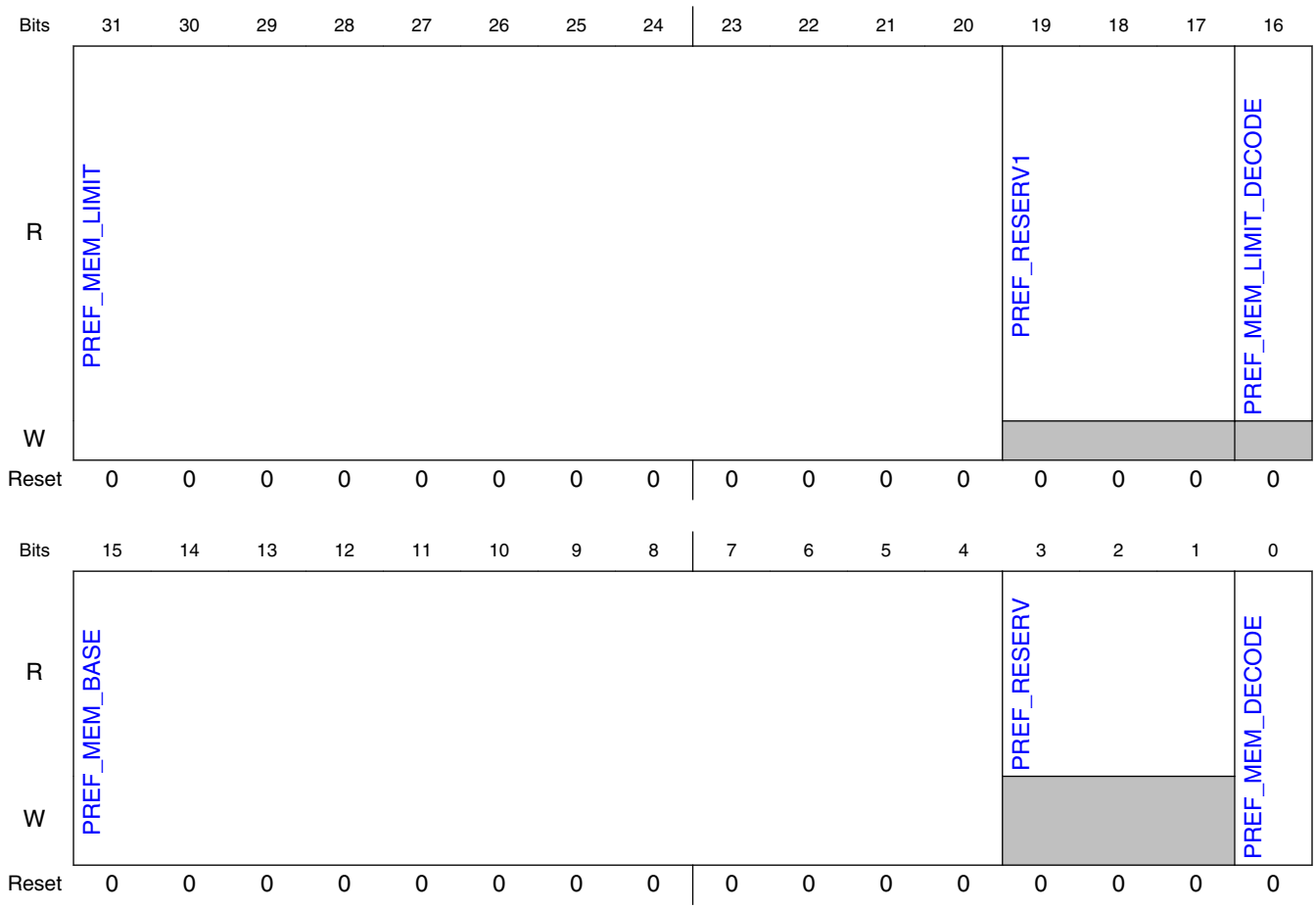
11.4.3.1.9.1 Offset

Register	Offset
PREF_MEM_LIMIT_PREF_MEM_BASE_REG	24h

11.4.3.1.9.2 Function

Prefetchable Memory Limit and Base Register. The Prefetchable Memory Limit and Prefetchable Memory Base registers must indicate that 64-bit addresses are supported, as defined in PCI-to-PCI Bridge Architecture Specification. The Prefetchable Memory Limit and Prefetchable Memory Base registers are optional. They define a prefetchable memory address range which is used by the bridge to determine when to forward memory transactions from one interface to the other (see the PCI-to-PCI Bridge Architecture Specification for additional details).

11.4.3.1.9.3 Diagram



11.4.3.1.9.4 Fields

Field	Function
31-20 PREF_MEM_LIMIT	Prefetchable Memory Limit Address. If the Prefetchable Memory Limit register indicates support for 32-bit addressing, then the Prefetchable Limit Upper 32 Bits register is implemented as a read-only register that returns zero when read. If the Prefetchable Memory Limit registers indicate support for 64-bit addressing, then the Prefetchable Limit Upper 32 Bits register is implemented as a read/write register which must be initialized by configuration software. If a 64-bit prefetchable memory address range is supported, the Prefetchable Limit Upper 32 Bits register specifies the upper 32 bits, corresponding to Address[63:32], of the 64-bit limit addresses which specify the prefetchable memory address range.
19-17 PREF_RESERV1	Reserved
16 PREF_MEM_LIMIT_DECODE	Prefetchable Memory Limit Decode. This bit encodes whether or not the bridge supports 64-bit addresses. The value of PREF_MEM_LIMIT_DECODE indicates the following: - 0b: Indicates that the bridge supports only 32 bit addresses - 1b: Indicates that the bridge supports 64 bit addresses. Prefetchable Limit Upper 32 Bits registers holds the rest of the 64-bit prefetchable limit address.
15-4	Prefetchable Memory Base Address. If the Prefetchable Memory Base register indicates support for 32-bit addressing, then the Prefetchable Base Upper 32 Bits register is implemented as a read-only register

Table continues on the next page...

Field	Function
PREF_MEM_BASE	that returns zero when read. If the Prefetchable Memory Base register indicates support for 64-bit addressing, then the Prefetchable Limit Upper 32 Bits register is implemented as a read/write register which must be initialized by configuration software. If a 64-bit prefetchable memory address range is supported, the Prefetchable Base Upper 32 Bits register specifies the upper 32 bits, corresponding to Address[63:32], of the 64-bit base addresses which specify the prefetchable memory address range.
3-1 PREF_RESERV	Reserved
0 PREF_MEM_DECODE	Prefetchable Memory Base Decode. This bit encodes whether or not the bridge supports 64-bit addresses. The value of PREF_MEM_DECODE indicates the following: - 0b: Indicates that the bridge supports only 32 bit addresses. - 1b: Indicates that the bridge supports 64 bit addresses. Prefetchable Base Upper 32 Bits registers holds the rest of the 64-bit prefetchable base address. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

11.4.3.1.10 Prefetchable Base Upper 32 Bits Register. (PREF_BASE_UPPER_REG)

11.4.3.1.10.1 Offset

Register	Offset
PREF_BASE_UPPER_REG	28h

11.4.3.1.10.2 Function

Prefetchable Base Upper 32 Bits Register. The Prefetchable Base Upper 32 Bits register is an optional extension to the Prefetchable Memory Base register.

11.4.3.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PREF_MEM_BASE_UPPER															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PREF_MEM_BASE_UPPER															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.10.4 Fields

Field	Function
31-0 PREF_MEM_BASE_UPPER	Prefetchable Base Upper 32 Bit. If the Prefetchable Memory Base register indicates support for 32-bit addressing, then this register is implemented as read-only register that returns zero when read. If the Prefetchable Memory Base register indicate support for 64-bit addressing, then this register is implemented as read/write register which must be initialized by configuration software. This register specifies the upper 32 bits, corresponding to Address[63:32], of the 64-bit base addresses which specify the prefetchable memory address range. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

11.4.3.1.11 Prefetchable Limit Upper 32 Bits Register. (PREF_LIMIT_UPPER_REG)

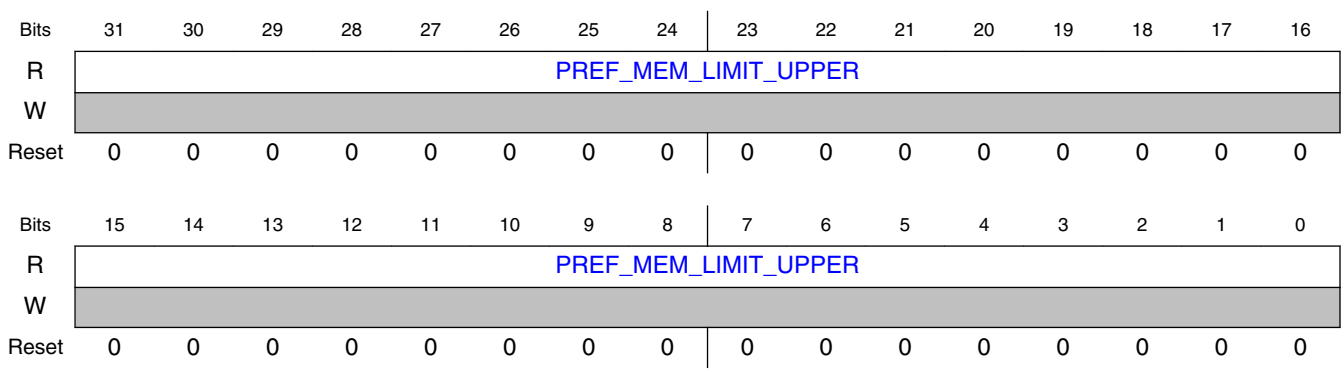
11.4.3.1.11.1 Offset

Register	Offset
PREF_LIMIT_UPPER_REG	2Ch

11.4.3.1.11.2 Function

Prefetchable Limit Upper 32 Bits Register. The Prefetchable Limit Upper 32 Bits register is an optional extension to the Prefetchable Memory Limit register.

11.4.3.1.11.3 Diagram



11.4.3.1.11.4 Fields

Field	Function
31-0 PREF_MEM_LIMIT_UPPER	Prefetchable Limit Upper 32 Bit. If the Prefetchable Memory Limit register indicate support for 64-bit addressing, then this register is implemented as read/write register which must be initialized by configuration software. This register specifies the upper 32 bits, corresponding to Address[63:32], of the 64-bit base addresses which specify the prefetchable memory address range. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

11.4.3.1.12 I/O Limit and Base Upper 16 Bits Register. (IO_LIMIT_UPPER_IO_BASE_UPPER_REG)

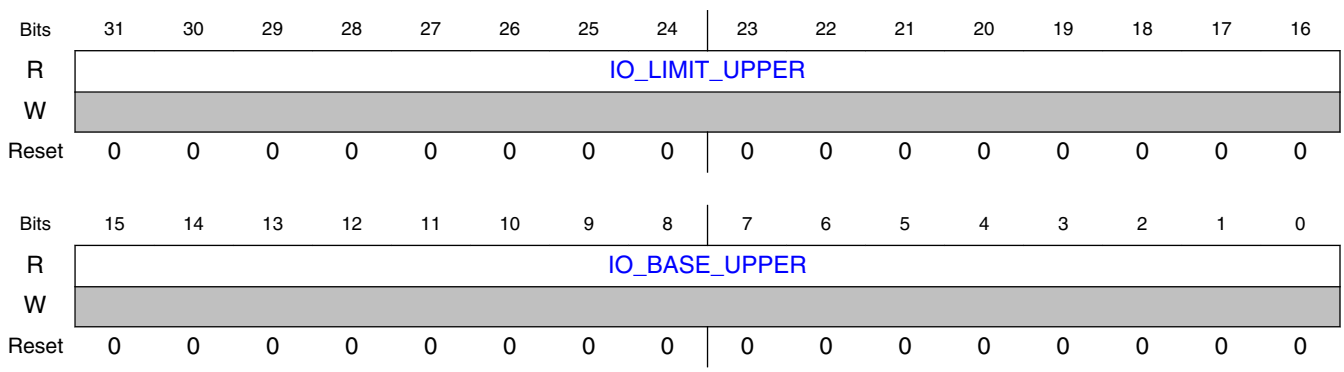
11.4.3.1.12.1 Offset

Register	Offset
IO_LIMIT_UPPER_IO_BASE_UPPER_REG	30h

11.4.3.1.12.2 Function

I/O Limit and Base Upper 16 Bits Register. The I/O Limit Upper 16 Bits and I/O Base Upper 16 Bits registers are optional extensions to the I/O Limit and I/O Base registers.

11.4.3.1.12.3 Diagram



11.4.3.1.12.4 Fields

Field	Function
31-16 IO_LIMIT_UPPER	I/O Limit Upper 16 Bits. If the I/O Limit register indicates support for 16-bit I/O address decoding, then this register is implemented as a read-only register which return zero when read. If the I/O Limit register indicates support for 32-bit I/O addressing, then this register must be initialized by configuration software.

Table continues on the next page...

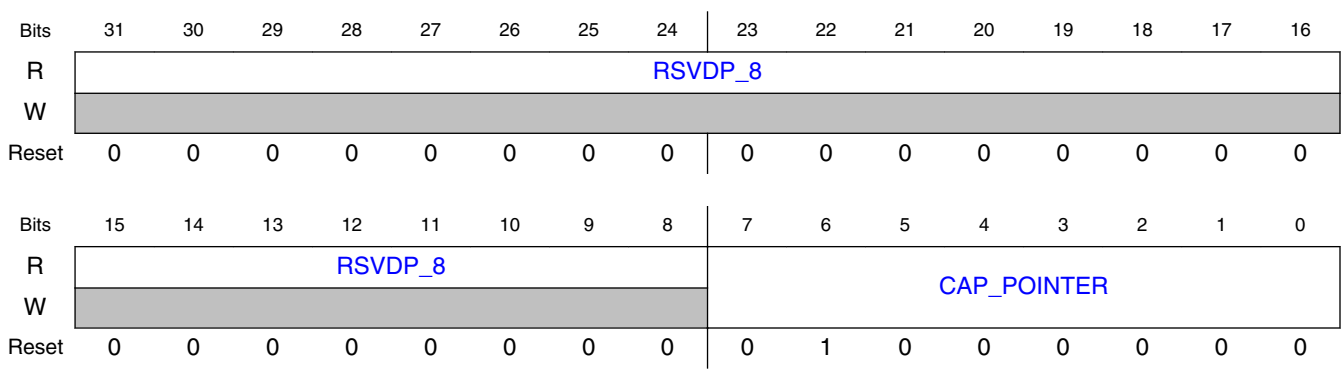
Field	Function
	If 32-bit I/O address decoding is supported, this register specifies the upper 16 bits, corresponding to Address[31:16], of the 32-bit limit address, that specify the I/O address range. See the PCI-to-PCI Bridge Architecture Specification for additional details). Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
15-0 IO_BASE_UPPER	I/O Base Upper 16 Bits. If the I/O Base register indicates support for 16-bit I/O address decoding, then this register is implemented as a read-only register which return zero when read. If the I/O base register indicates support for 32-bit I/O addressing, then this register must be initialized by configuration software. If 32-bit I/O address decoding is supported, this register specifies the upper 16 bits, corresponding to Address[31:16], of the 32-bit base address, that specify the I/O address range. See the PCI-to-PCI Bridge Architecture Specification for additional details. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

11.4.3.1.13 Capabilities Pointer Register. (TYPE1_CAP_PTR_REG)

11.4.3.1.13.1 Offset

Register	Offset
TYPE1_CAP_PTR_REG	34h

11.4.3.1.13.2 Diagram



11.4.3.1.13.3 Fields

Field	Function
31-8 RSVDP_8	Reserved for future use.
7-0 CAP_POINTER	Capabilities Pointer. This register is used to point to a linked list of capabilities implemented by this Function. Since all PCI Express Functions are required to implement the PCI Express Capability structure, this register must point to a valid capability structure and either this structure is the PCI Express Capability structure, or a subsequent list item points to the PCI Express Capability structure. The bottom

Field	Function
	two bits are Reserved and must be set to 00b. Software must mask these bits off before using this register as a pointer in Configuration Space to the first entry of a linked list of new capabilities. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

11.4.3.1.14 Expansion ROM Base Address Register. (TYPE1_EXP_ROM_BASE_REG)

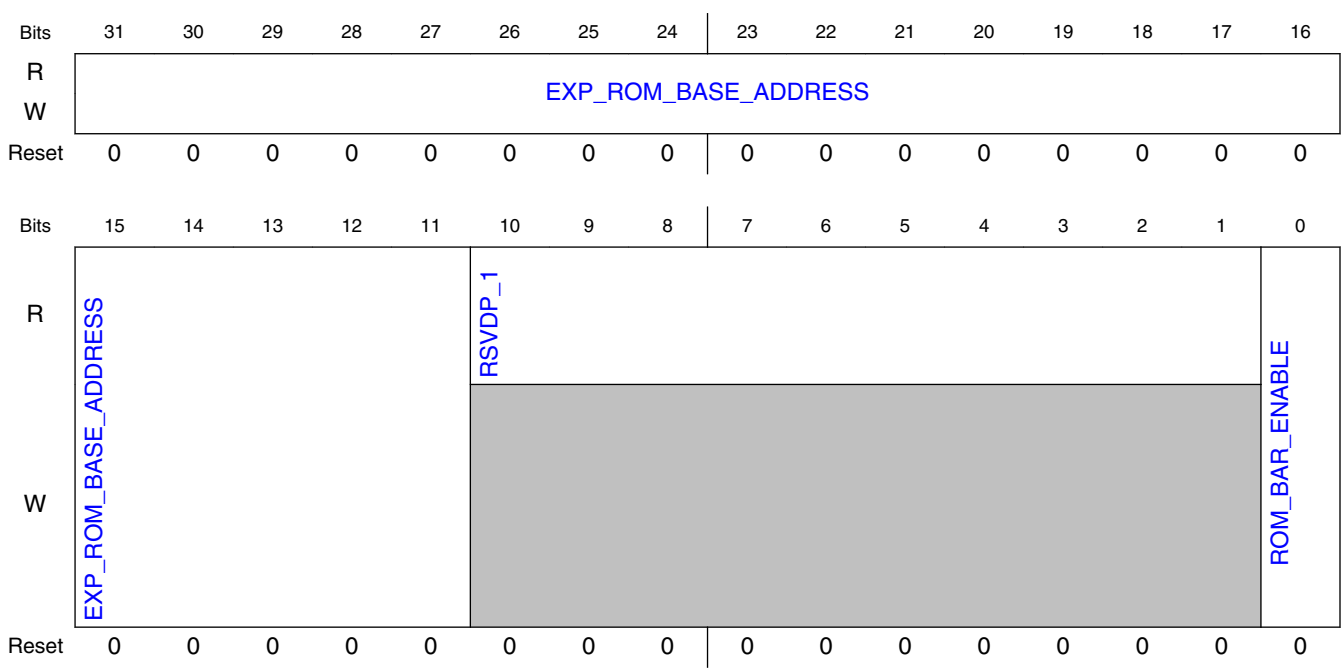
11.4.3.1.14.1 Offset

Register	Offset
TYPE1_EXP_ROM_BASE_REG	38h

11.4.3.1.14.2 Function

Expansion ROM Base Address Register. This register is defined to handle the base address and size information for this expansion ROM. The mask for this ROM BAR exists (if implemented) as a shadow register at this address. The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the second register at this address.

11.4.3.1.14.3 Diagram



11.4.3.1.14.4 Fields

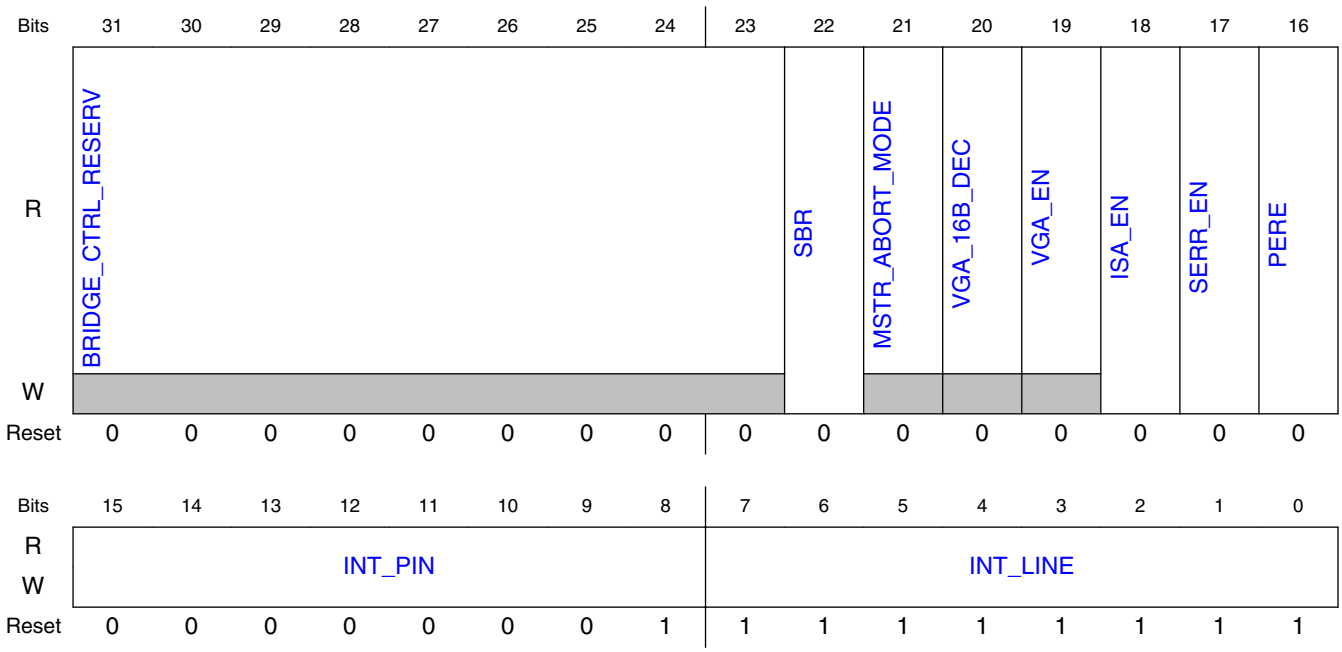
Field	Function
31-11 EXP_ROM_BASE_ADDRESS	Expansion ROM Base Address. Upper 21 bits of the Expansion ROM base address. The number of bits (out of these 21) that a Function actually implements depends on how much address space the Function requires. The mask for this ROM BAR exists (if implemented) as a shadow register at this address. The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the second register at this address. Note: The access attributes of this field are as follows: - Dbi: R/W
10-1 RSVDP_1	Reserved for future use.
0 ROM_BAR_ENABLE	Expansion ROM Enable. This bit controls whether or not the Function accepts accesses to its expansion ROM. When this bit is 0b, the Function's expansion ROM address space is disabled. When the bit is 1b, address decoding is enabled using the parameters in the other part of the Expansion ROM Base Address register. The Memory Space Enable bit in the Command register has precedence over the Expansion ROM Enable bit. A Function must claim accesses to its expansion ROM only if both the Memory Space Enable bit and the Expansion ROM Enable bit are set. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.15 Bridge Control, Interrupt Pin, and Interrupt Line Register. (BRIDGE_CTRL_INT_PIN_INT_LINE_REG)

11.4.3.1.15.1 Offset

Register	Offset
BRIDGE_CTRL_INT_PIN_INT_LINE_REG	3Ch

11.4.3.1.15.2 Diagram



11.4.3.1.15.3 Fields

Field	Function
31-23 BRIDGE_CTRL_RESERV	Reserved
22 SBR	Secondary Bus Reset. Setting this bit triggers a hot reset on the corresponding PCI Express Port. Software must ensure a minimum reset duration (Trst) as defined in the PCI Local Bus Specification. Software and systems must honor first-access-following-reset timing requirements, unless the Readiness Notifications mechanism is used or if the Immediate Readiness bit in the relevant Function's Status Register register is set. Port configuration registers must not be changed, except as required to update Port status.
21 MSTR_ABORT_MODE	Master Abort Mode. This bit was originally described in the PCI-to-PCI Bridge Architecture Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Dbi: R/W
20 VGA_16B_DEC	VGA 16 bit decode. This bit only has meaning if VGA Enable bit is set. This bit enables system configuration software to select between 10-bit and 16-bit I/O address decoding for all VGA I/O register accesses that are forwarded from primary to secondary. The following actions are taken based on the value of the VGA_16B_DEC bit: - 0b: Execute 10-bit address decodes on VGA I/O accesses - 1b: Execute 16-bit address decodes on VGA I/O accesses For Functions that do not support VGA, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Dbi: R
19 VGA_EN	VGA Enable. Modifies the response by the bridge to VGA compatible addresses. If the VGA Enable bit is set, the bridge will positively decode and forward the following accesses on the primary interface to the secondary interface (and, conversely, block the forwarding of these addresses from the secondary to primary interface): - Memory accesses in the range 000A 0000h to 000B FFFFh - I/O addresses in the first 64 KB of the I/O address space (Address[31:16] are 0000h) where Address[9:0] are in the ranges 3B0h to 3BBh and 3C0h to 3DFh (inclusive of ISA address aliases determined by the setting of VGA 16-

Table continues on the next page...

Field	Function
	bit Decode) If the VGA Enable bit is set, forwarding of these accesses is independent of the I/O address range and memory address ranges defined by the I/O Base and Limit registers, the Memory Base and Limit registers, and the Prefetchable Memory Base and Limit registers of the bridge. (Forwarding of these accesses is also independent of the setting of the ISA Enable bit (in the Bridge Control register) when the VGA Enable bit is set. Forwarding of these accesses is qualified by the I/O Space Enable and Memory Space Enable bits in the Command register.) The following actions are taken based on the value of the VGA_EN bit: - 0b: Do not forward VGA compatible memory and I/O addresses from the primary to the secondary interface (addresses defined above) unless they are enabled for forwarding by the defined I/O and memory address ranges - 1b: Forward VGA compatible memory and I/O addresses (addresses defined above) from the primary interface to the secondary interface (if the I/O Space Enable and Memory Space Enable bits are set) independent of the I/O and memory address ranges and independent of the ISA Enable bit For Functions that do not support VGA, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Dbi: R
18 ISA_EN	ISA Enable. Modifies the response by the bridge to ISA I/O addresses. This applies only to I/O addresses that are enabled by the I/O Base and I/O Limit registers and are in the first 64 KB of I/O address space (0000 0000h to 0000 FFFFh). If this bit is set, the bridge will block any forwarding from primary to secondary of I/O transactions addressing the last 768 bytes in each 1-KB block. In the opposite direction (secondary to primary), I/O transactions will be forwarded if they address the last 768 bytes in each 1-KB block. The following actions are taken based on the value of the ISA_EN bit: - 0b: Forward downstream all I/O addresses in the address range defined by the I/O Base and I/O Limit registers - 1b: Forward upstream ISA I/O addresses in the address range defined by the I/O Base and I/O Limit registers that are in the first 64 KB of PCI I/O address space (top 768 bytes of each 1-KB block.
17 SERR_EN	SERR# Enable. This bit controls forwarding of ERR_COR, ERR_NONFATAL and ERR_FATAL from secondary to primary.
16 PERE	Parity Error Response Enable. This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Secondary Status register.
15-8 INT_PIN	Interrupt PIN. The Interrupt Pin register register that identifies the legacy interrupt Message(s) the Function uses. Valid values are: - 01h, 02h, 03h, and 04h: map to legacy interrupt Messages for INTA, INTB, INTC, and INTD respectively. - 00h: indicates that the Function uses no legacy interrupt Message(s). - 05h through FFh: Reserved PCI Express defines one legacy interrupt Message for a single Function device and up to four legacy interrupt Messages for a multi-Function device. For a single Function device, only INTA may be used. Any Function on a multi-Function device can use any of the INTx Messages. If a device implements a single legacy interrupt Message, it must be INTA; if it implements two legacy interrupt Messages, they must be INTA and INTB; and so forth. For a multi-Function device, all Functions may use the same INTx Message or each may have its own (up to a maximum of four Functions) or any combination thereof. A single Function can never generate an interrupt request on more than one INTx Message. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7-0 INT_LINE	Interrupt Line. The Interrupt Line register communicates interrupt line routing information. The register must be implemented by any Function that uses an interrupt pin. Values in this register are programmed by system software and are system architecture specific. The Function itself does not use this value; rather the value in this register is used by device drivers and operating systems.

11.4.3.1.16 Power Management Capabilities Register. (CAP_ID_NXT_PTR_REG)

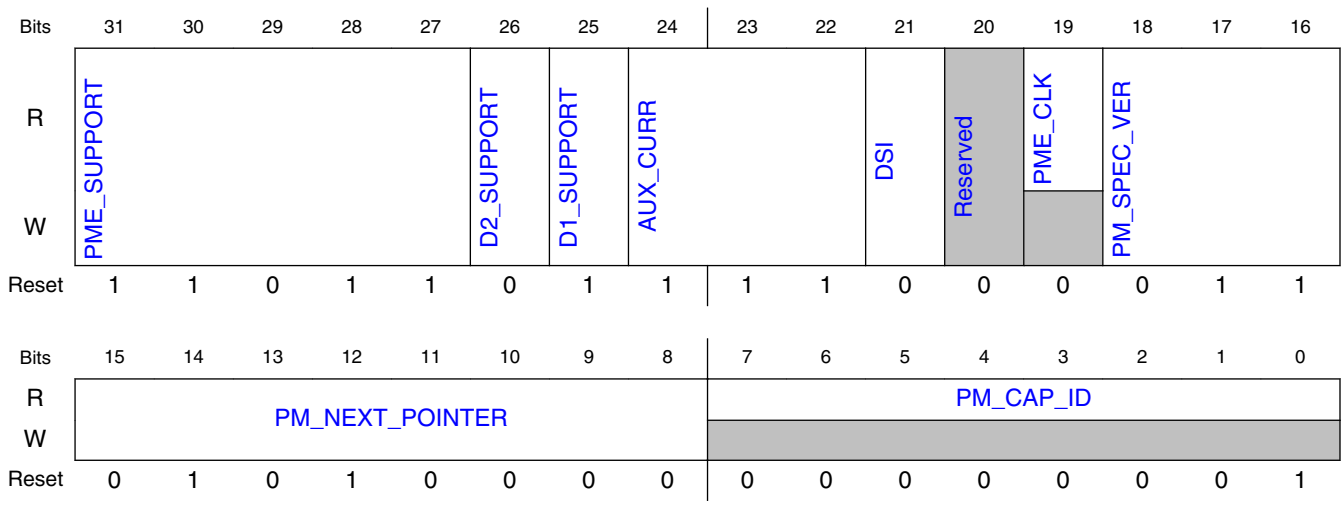
11.4.3.1.16.1 Offset

Register	Offset
CAP_ID_NXT_PTR_REG	40h

11.4.3.1.16.2 Function

Power Management Capabilities Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.16.3 Diagram



11.4.3.1.16.4 Fields

Field	Function
31-27 PME_SUPPORT	Power Management Event Support. For a description of this standard PCIe register field, see the PCI Express Specification. The read value from this field is the write value && (sys_aux_pwr_det, 1'b1, D2_SUPPORT, D1_SUPPORT, 1'b1), where D1_SUPPORT and D2_SUPPORT are fields in this register. The reset value PME_SUPPORT_n && (sys_aux_pwr_det, 1'b1, D2_SUPPORT, D1_SUPPORT, 1'b1), where PME_SUPPORT_n is a configuration parameter. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
26 D2_SUPPORT	D2 State Support. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
25 D1_SUPPORT	D1 State Support. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

Table continues on the next page...

Field	Function
24-22 AUX_CURR	Auxiliary Current Requirements. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
21 DSI	Device Specific Initialization Bit. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
20 —	Reserved
19 PME_CLK	PCI Clock Requirement. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
18-16 PM_SPEC_VER	Power Management Spec Version. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
15-8 PM_NEXT_POINTER	Next Capability Pointer. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7-0 PM_CAP_ID	Power Management Capability ID. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.17 Power Management Control and Status Register. (CON_STATUS_REG)

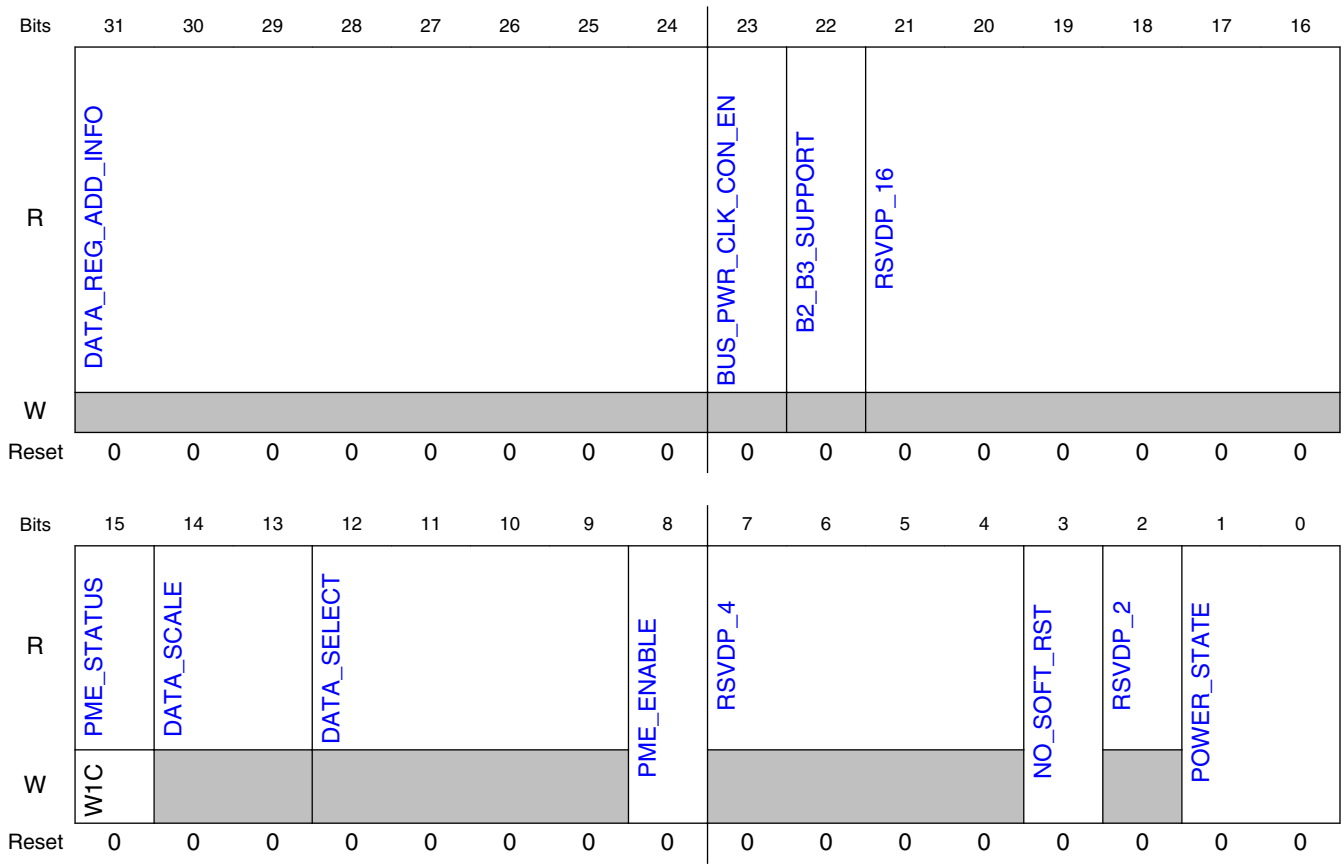
11.4.3.1.17.1 Offset

Register	Offset
CON_STATUS_REG	44h

11.4.3.1.17.2 Function

Power Management Control and Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.17.3 Diagram



11.4.3.1.17.4 Fields

Field	Function
31-24 DATA_REG_ADD_INFO	Power Data Information Register. For a description of this standard PCIe register field, see the PCI Express Specification.
23 BUS_PWR_CLK_CON_EN	Bus Power/Clock Control Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
22 B2_B3_SUPPORT	B2B3 Support for D3hot. For a description of this standard PCIe register field, see the PCI Express Specification.
21-16 RSVDP_16	Reserved for future use.
15 PME_STATUS	PME Status. For a description of this standard PCIe register field, see the PCI Express Specification.
14-13 DATA_SCALE	Data Scaling Factor. For a description of this standard PCIe register field, see the PCI Express Specification.

Table continues on the next page...

Field	Function
12-9 DATA_SELECT	Data Select. For a description of this standard PCIe register field, see the PCI Express Specification.
8 PME_ENABLE	PME Enable. For a description of this standard PCIe register field, see the PCI Express Specification. The PMC registers this value under aux power. Sometimes it might remember the old value, even if you try to clear it by writing '0'. Note: This register field is sticky.
7-4 RSVDP_4	Reserved for future use.
3 NO_SOFT_RST	No soft Reset. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2 RSVDP_2	Reserved for future use.
1-0 POWER_STATE	Power State. For a description of this standard PCIe register field, see the PCI Express Specification. You can write to this register. However, the read-back value is the actual power state, not the write value. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.18 MSI Capability ID, Next Pointer, Capability/Control Registers. (PCI_MSI_CAP_ID_NEXT_CTRL_REG)

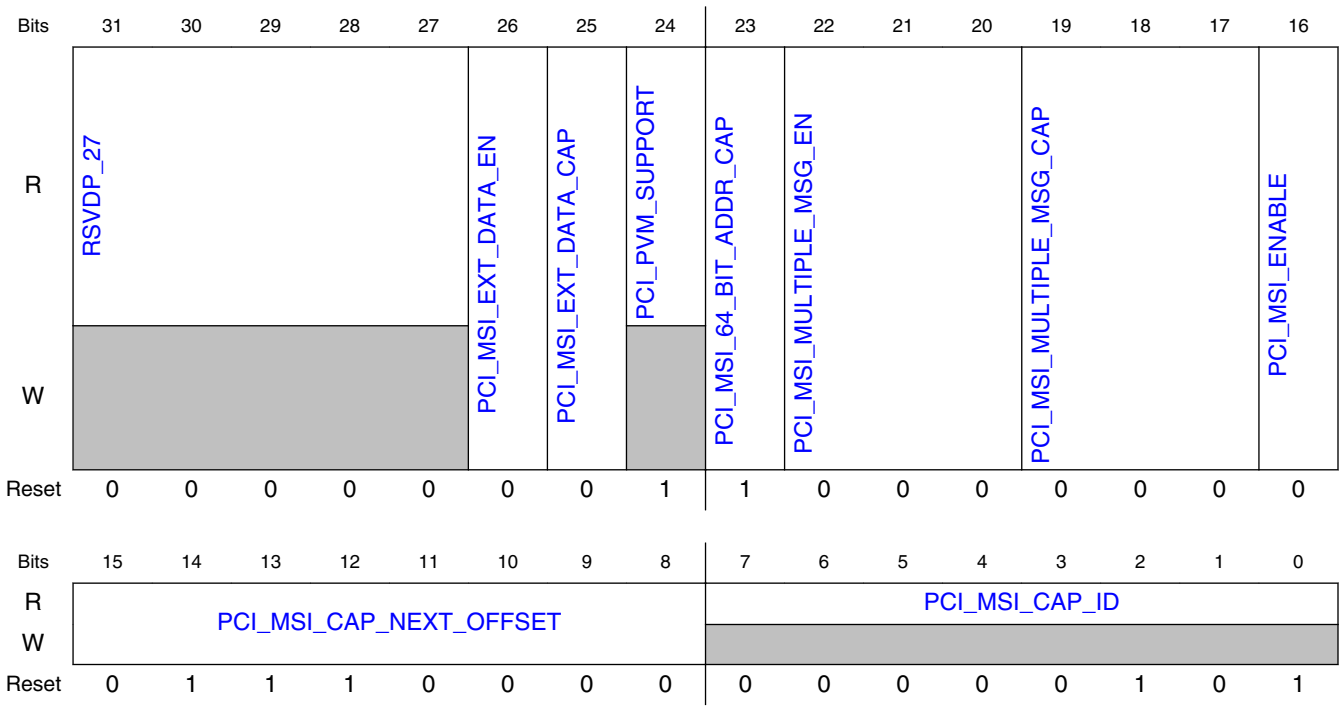
11.4.3.1.18.1 Offset

Register	Offset
PCI_MSI_CAP_ID_NEXT_CTRL_REG	50h

11.4.3.1.18.2 Function

MSI Capability ID, Next Pointer, Capability/Control Registers. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.18.3 Diagram



11.4.3.1.18.4 Fields

Field	Function
31-27 RSVDP_27	Reserved for future use.
26 PCI_MSI_EXT_DATA_EN	Extended Message Data Enable. For a description of this standard PCIe register, see the PCI-SIG ECN for Extended MSI Data, Feb 24, 2016, affecting PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: PCI_MSI_CAP_ID_NEXT_CTRL_REG.PCI_MSI_EXT_DATA_CAP ? RW : RO
25 PCI_MSI_EXT_DATA_CAP	Extended Message Data Capable. For a description of this standard PCIe register, see the PCI-SIG ECN for Extended MSI Data, Feb 24, 2016, affecting PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
24 PCI_PVM_SUPPORT	MSI Per Vector Masking Capable. For a description of this standard PCIe register field, see the PCI Express Specification.
23 PCI_MSI_64_BIT_ADDR_CAP	MSI 64-bit Address Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
22-20 PCI_MSI_MULTIPLE_MSG_EN	MSI Multiple Message Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

Table continues on the next page...

Field	Function
19-17 PCI_MSI_MULTIPLE_MSG_CAP	MSI Multiple Message Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
16 PCI_MSI_ENABLE	MSI Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
15-8 PCI_MSI_CAP_NEXT_OFFSET	MSI Capability Next Pointer. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7-0 PCI_MSI_CAP_ID	MSI Capability ID. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.19 MSI Message Lower Address Register. (MSI_CAP_OFF_04H_REG)

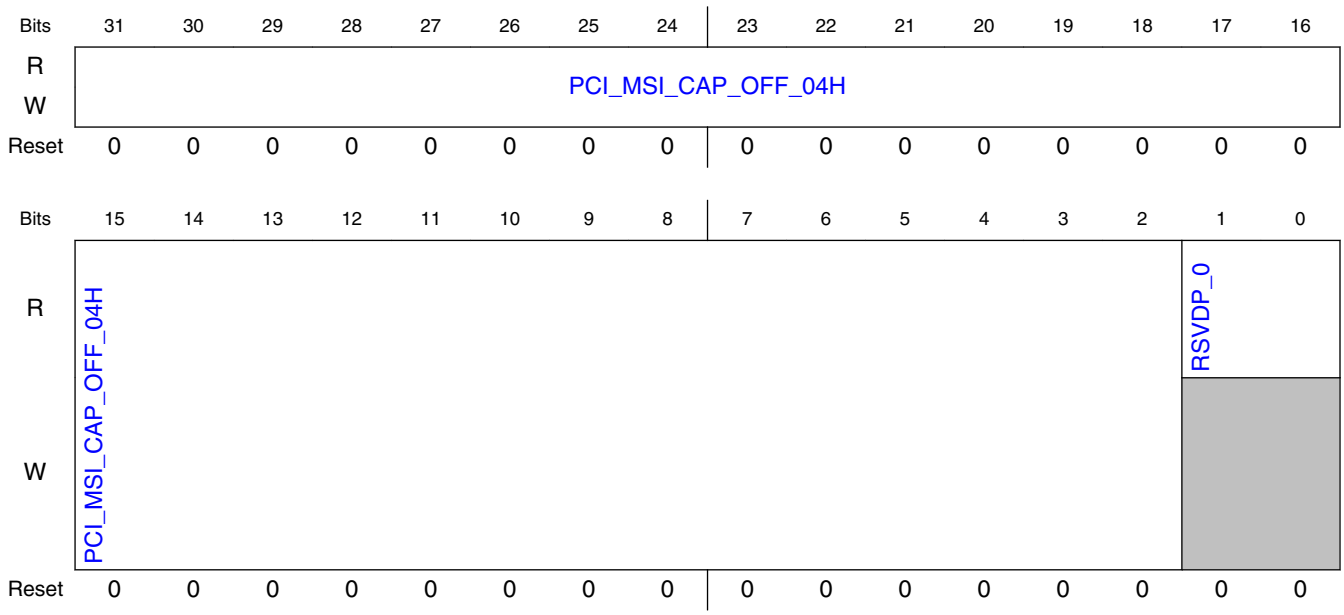
11.4.3.1.19.1 Offset

Register	Offset
MSI_CAP_OFF_04H_REG	54h

11.4.3.1.19.2 Function

MSI Message Lower Address Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.19.3 Diagram



11.4.3.1.19.4 Fields

Field	Function
31-2 PCI_MSI_CAP_OFF_04H	MSI Message Lower Address Field. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
1-0 RSVDP_0	Reserved for future use.

11.4.3.1.20 For a 32 bit MSI Message, this register contains Data. (MSI_CAP_OFF_08H_REG)

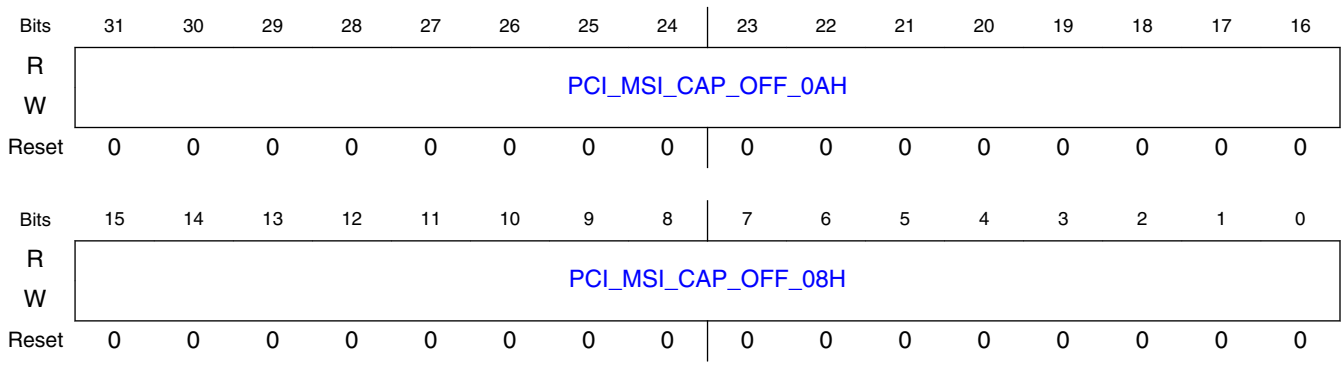
11.4.3.1.20.1 Offset

Register	Offset
MSI_CAP_OFF_08H_REG	58h

11.4.3.1.20.2 Function

For a 32 bit MSI Message, this register contains Data. For 64 bit it contains the Upper Address. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.20.3 Diagram



11.4.3.1.20.4 Fields

Field	Function
31-16 PCI_MSI_CAP_OFF_0AH	For a 32 bit MSI Message, this field contains Ext MSI Data. For 64-bit it contains upper 16 bits of the Upper Address. For a description of this standard PCIe register field, see the PCI Express Specification Note: The access attributes of this field are as follows: - Dbi: PCI_MSI_64_BIT_ADDR_CAP || `DEFAULT_EXT_MSI_DATA_CAPABLE ? R/W : R
15-0 PCI_MSI_CAP_OFF_08H	For a 32-bit MSI Message, this field contains Data. For 64-bit it contains lower 16 bits of the Upper Address. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: PCI_MSI_64_BIT_ADDR_CAP ? R/W : R

11.4.3.1.21 For a 64 bit MSI Message, this register contains Data. (MSI_CAP_OFF_0CH_REG)

11.4.3.1.21.1 Offset

Register	Offset
MSI_CAP_OFF_0CH_REG	5Ch

11.4.3.1.21.2 Function

For a 64 bit MSI Message, this register contains Data. For 32 bit, it contains Mask Bits if PVM enabled. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PCI_MSI_CAP_OFF_0EH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCI_MSI_CAP_OFF_0CH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.21.4 Fields

Field	Function
31-16 PCI_MSI_CAP_OFF_0EH	For a 64-bit MSI Message, this field contains Data. For 32-bit, it contains the upper Mask Bits if PVM is enabled. For a description of this standard PCIe register field, see the PCI Express Specification Note: The access attributes of this field are as follows: - Dbi: (!MSI_64_EN && MSI_PVM_EN_VALUE) ? RW: MSI_64_EN && DEFAULT_EXT_MSI_DATA_CAPABLE ? RW : RO
15-0 PCI_MSI_CAP_OFF_0CH	For a 64-bit MSI Message, this field contains Data. For 32-bit, it contains the lower Mask Bits if PVM is enabled. For a description of this standard PCIe register field, see the PCI Express Specification Note: The access attributes of this field are as follows: - Dbi: PCI_MSI_64_BIT_ADDR_CAP || MSI_PVM_EN ? R/W : R

11.4.3.1.22 Used for MSI when Vector Masking Capable. (MSI_CAP_OFF_10H_REG)

11.4.3.1.22.1 Offset

Register	Offset
MSI_CAP_OFF_10H_REG	60h

11.4.3.1.22.2 Function

Used for MSI when Vector Masking Capable. For 32 bit contains Pending Bits. For 64 bit, contains Mask Bits. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PCI_MSI_CAP_OFF_10H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCI_MSI_CAP_OFF_10H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.22.4 Fields

Field	Function
31-0 PCI_MSI_CAP_OFF_10H	Used for MSI when Vector Masking Capable. For 32-bit contains Pending Bits. For 64-bit, contains Mask Bits. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: PCI_MSI_64_BIT_ADDR_CAP && MSI_PVM_EN ? R/W : R

11.4.3.1.23 Used for MSI 64 bit messaging when Vector Masking Capable. (MSI_CAP_OFF_14H_REG)

11.4.3.1.23.1 Offset

Register	Offset
MSI_CAP_OFF_14H_REG	64h

11.4.3.1.23.2 Function

Used for MSI 64 bit messaging when Vector Masking Capable. Contains Pending Bits. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PCI_MSI_CAP_OFF_14H															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PCI_MSI_CAP_OFF_14H															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.23.4 Fields

Field	Function
31-0 PCI_MSI_CAP_OFF_14H	Used for MSI 64-bit messaging when Vector Masking Capable. Contains Pending Bits. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.24 PCI Express Capabilities, ID, Next Pointer Register. (PCIE_CAP_ID_PCIE_NEXT_CAP_PTR_PCIE_CAP_REG)

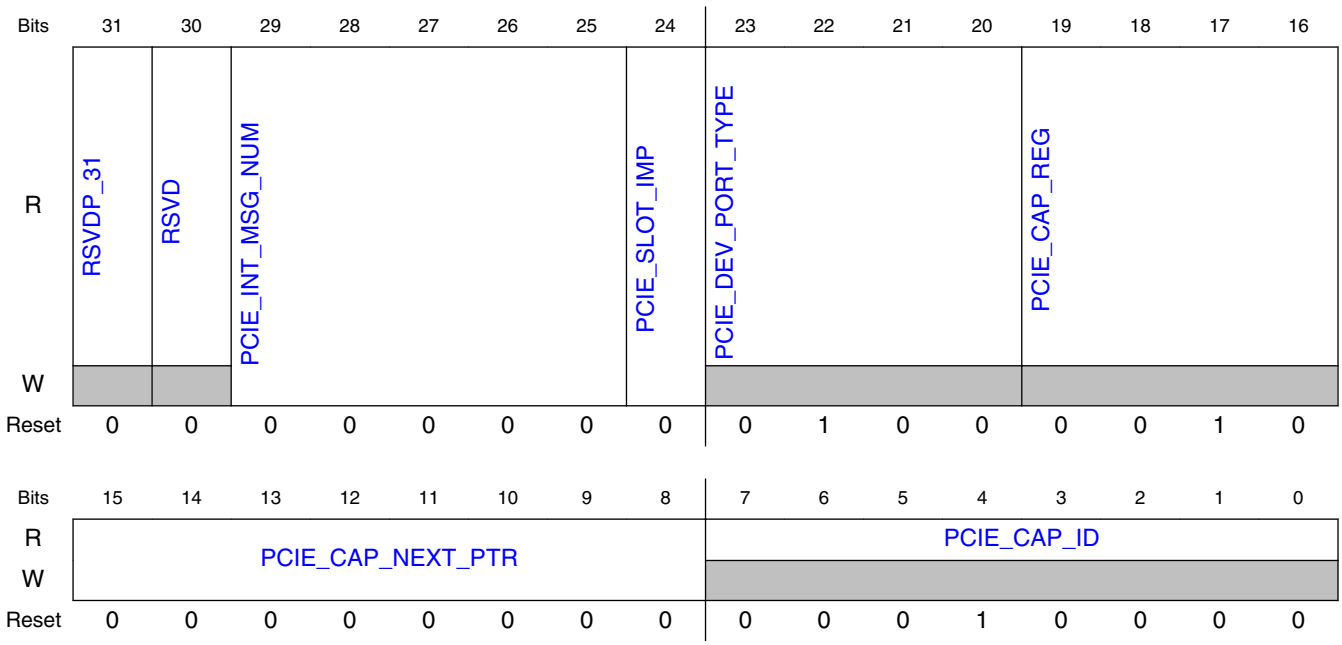
11.4.3.1.24.1 Offset

Register	Offset
PCIE_CAP_ID_PCIE_NEXT_CAP_PTR_PCIE_CAP_REG	70h

11.4.3.1.24.2 Function

PCI Express Capabilities, ID, Next Pointer Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.24.3 Diagram



11.4.3.1.24.4 Fields

Field	Function
31 RSVDP_31	Reserved for future use.
30 RSVD	Reserved For a description of this standard PCIe register field, see the PCI Express Specification.
29-25 PCIE_INT_MSG_NUM	PCIE Interrupt Message Number. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
24 PCIE_SLOT_IMP	PCIe Slot Implemented Valid. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
23-20 PCIE_DEV_PORT_TYPE	PCIE Device/PortType. For a description of this standard PCIe register field, see the PCI Express Specification.
19-16 PCIE_CAP_REG	PCIE Capability Version Number. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 PCIE_CAP_NEXT_PTR	PCIE Next Capability Pointer. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7-0 PCIE_CAP_ID	PCIE Capability ID. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.25 Device Capabilities Register. (DEVICE_CAPABILITIES_REG)

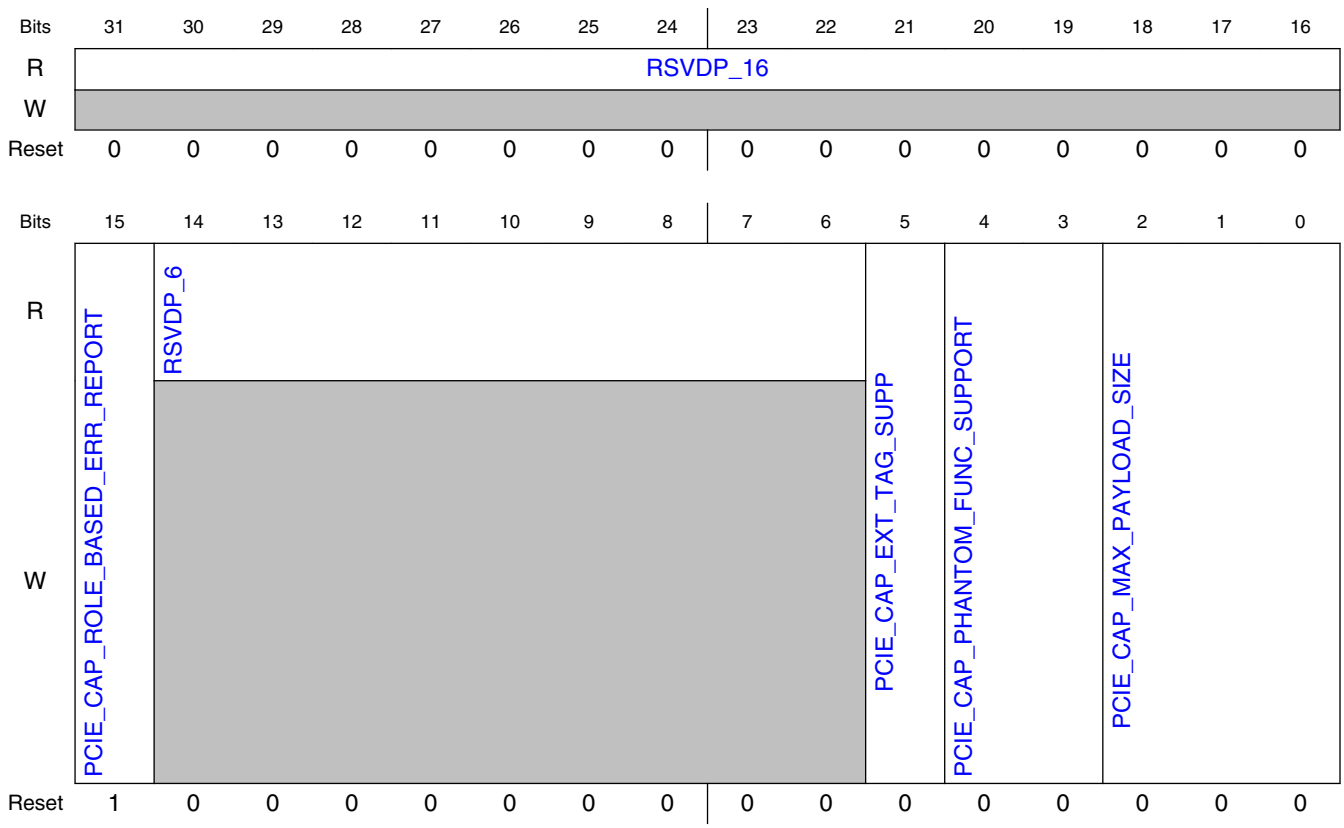
11.4.3.1.25.1 Offset

Register	Offset
DEVICE_CAPABILITIES_REG	74h

11.4.3.1.25.2 Function

Device Capabilities Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.25.3 Diagram



11.4.3.1.25.4 Fields

Field	Function
31-16 RSVDP_16	Reserved for future use.
15 PCIE_CAP_ROLE_BASED_ERR_REPORT	Role-based Error Reporting Implemented. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
14-6 RSVDP_6	Reserved for future use.
5 PCIE_CAP_EXT_TAG_SUPP	Extended Tag Field Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
4-3 PCIE_CAP_PHANTOM_FUNC_SUPPORT	Phantom Functions Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2-0 PCIE_CAP_MAX_PAYLOAD_SIZE	Max Payload Size Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

11.4.3.1.26 Device Control and Status Register. (DEVICE_CONTROL_DEVICE_STATUS)

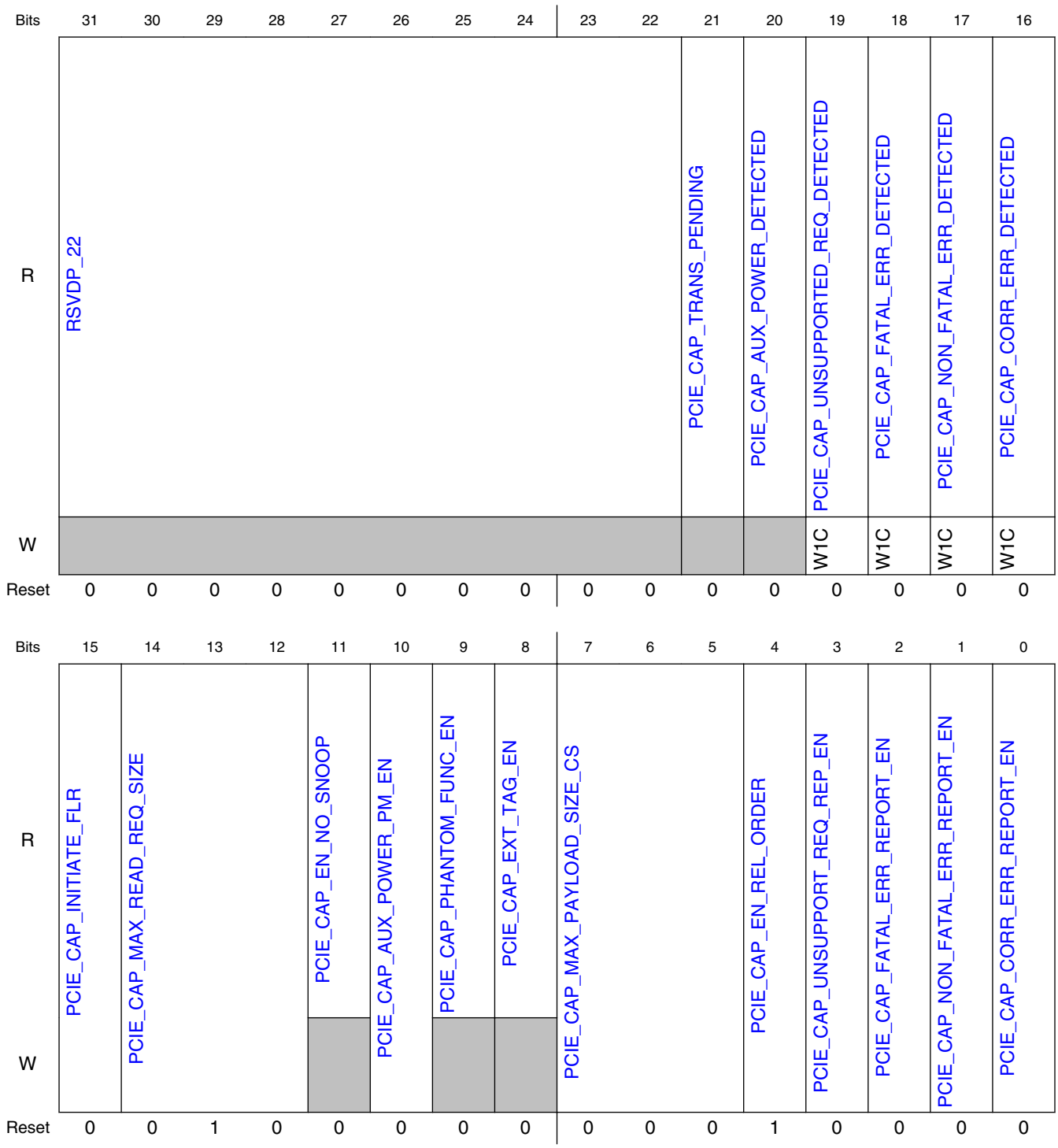
11.4.3.1.26.1 Offset

Register	Offset
DEVICE_CONTROL_DEVICE_STATUS	78h

11.4.3.1.26.2 Function

Device Control and Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.26.3 Diagram



11.4.3.1.26.4 Fields

Field	Function
31-22	Reserved for future use.

Table continues on the next page...

Field	Function
RSVDP_22	
21 PCIE_CAP_TRANS_PENDING	Transactions Pending Status. For a description of this standard PCIe register field, see the PCI Express Specification.
20 PCIE_CAP_AUX_POWER_DETECTED	Aux Power Detected Status. For a description of this standard PCIe register field, see the PCI Express Specification. This bit is derived by sampling the sys_aux_pwr_det input.
19 PCIE_CAP_UNSUPPORTED_REQ_DETECTED	Unsupported Request Detected Status. For a description of this standard PCIe register field, see the PCI Express Specification.
18 PCIE_CAP_FATAL_ERR_DETECTED	Fatal Error Detected Status. For a description of this standard PCIe register field, see the PCI Express Specification.
17 PCIE_CAP_NON_FATAL_ERR_DETECTED	Non-Fatal Error Detected Status. For a description of this standard PCIe register field, see the PCI Express Specification.
16 PCIE_CAP_CORR_ERR_DETECTED	Correctable Error Detected Status. For a description of this standard PCIe register field, see the PCI Express Specification.
15 PCIE_CAP_INITIATE_FLR	Initiate Function Level Reset (for endpoints). For a description of this standard PCIe register field, see the PCI Express Specification.
14-12 PCIE_CAP_MAX_READ_REQ_SIZE	Max Read Request Size. For a description of this standard PCIe register field, see the PCI Express Specification.
11 PCIE_CAP_EN_NO_SNOOP	Enable No Snoop. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R
10 PCIE_CAP_AUX_POWER_PM_EN	Aux Power PM Enable. For a description of this standard PCIe register field, see the PCI Express Specification. This bit is derived by sampling the sys_aux_pwr_det input. Note: This register field is sticky.
9 PCIE_CAP_PHANTOM_FUNC_EN	Phantom Functions Enable. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_PHANTOM_FUNC_SUPPORT field of DEVICE_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Dbi: DEVICE_CAPABILITIES_REG.PCIE_CAP_PHANTOM_FUNC_SUPPORT ? RW : RO
8 PCIE_CAP_EXT_TAG_EN	Extended Tag Field Enable. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_EXT_TAG_SUPP field of DEVICE_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Dbi: DEVICE_CAPABILITIES_REG.PCIE_CAP_EXT_TAG_SUPP ? RW : RO

Table continues on the next page...

Field	Function
7-5 PCIE_CAP_MAX_PAYLOAD_SIZE_CS	Max Payload Size. Max_Payload_Size . This field sets maximum TLP payload size for the Function. Permissible values that can be programmed are indicated by the Max_Payload_Size Supported field (PCIE_CAP_MAX_PAYLOAD_SIZE) in the Device Capabilities register (DEVICE_CAPABILITIES_REG).
4 PCIE_CAP_EN_REL_ORDER	Enable Relaxed Ordering. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
3 PCIE_CAP_UNSUPPORT_REQ_REP_EN	Unsupported Request Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
2 PCIE_CAP_FATAL_ERR_REPORT_EN	Fatal Error Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
1 PCIE_CAP_NON_FATAL_ERR_REPORT_EN	Non-fatal Error Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
0 PCIE_CAP_CORR_ERR_REPORT_EN	Correctable Error Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.27 Link Capabilities Register. (LINK_CAPABILITIES_REG)

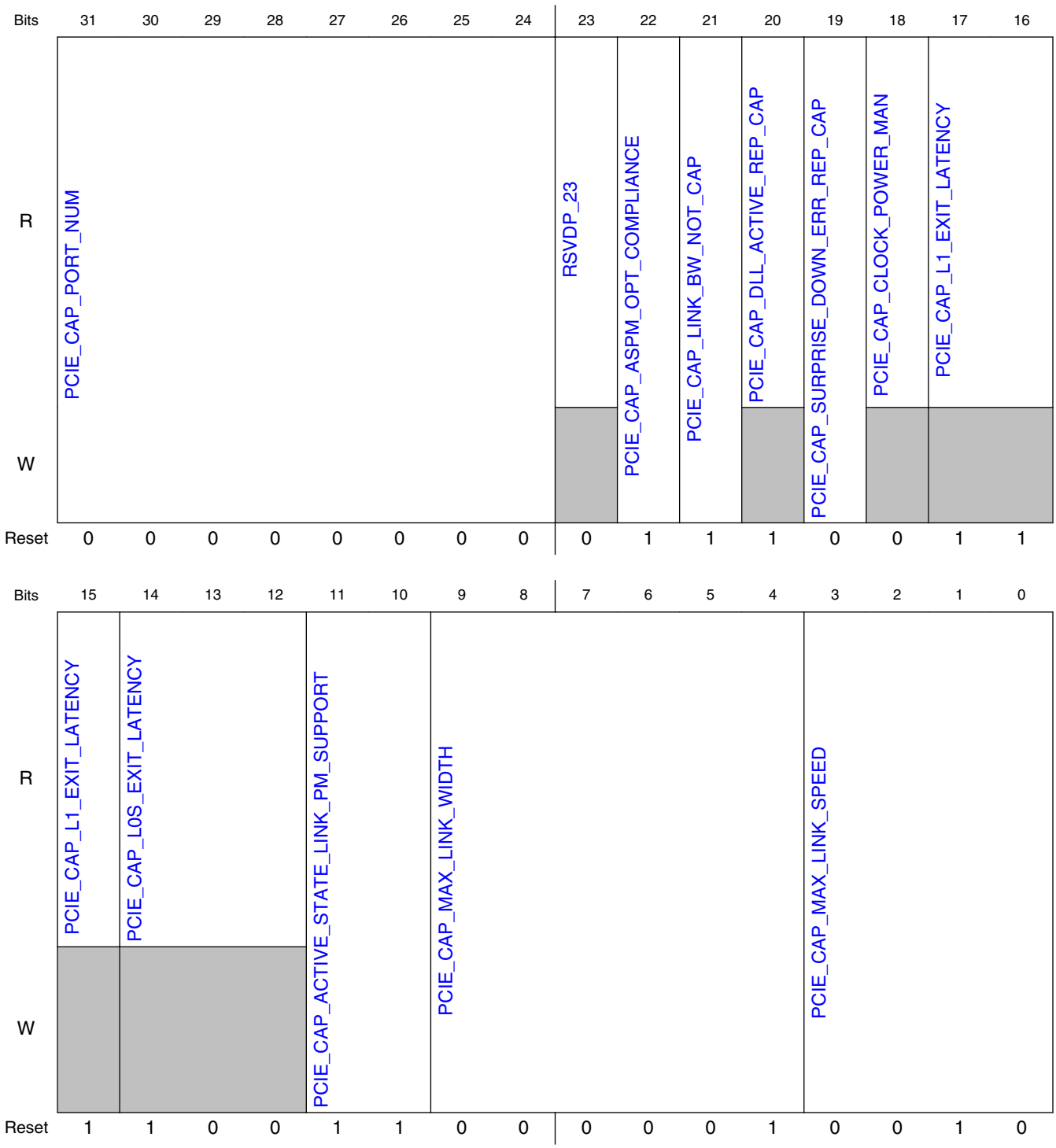
11.4.3.1.27.1 Offset

Register	Offset
LINK_CAPABILITIES_REG	7Ch

11.4.3.1.27.2 Function

Link Capabilities Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.27.3 Diagram



11.4.3.1.27.4 Fields

Field	Function
31-24 PCIE_CAP_PORT_NUM	Port Number. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
23 RSVDP_23	Reserved for future use.
22 PCIE_CAP_ASPM_OPT_COMPLIANCE	ASPM Optionality Compliance. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
21 PCIE_CAP_LINK_BW_NOT_CAP	Link Bandwidth Notification Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
20 PCIE_CAP_DLL_ACTIVE_REP_CAP	Data Link Layer Link Active Reporting Capable. For a description of this standard PCIe register field, see the PCI Express Specification.
19 PCIE_CAP_SURPRISE_DOWN_ERR_REP_CAP	Surprise Down Error Reporting Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
18 PCIE_CAP_CLOCK_POWER_MAN	Clock Power Management. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
17-15 PCIE_CAP_L1_EXIT_LATENCY	L1 Exit Latency. For a description of this standard PCIe register field, see the PCI Express Specification. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is supported (possible) in the controller when one or more of the following expressions is true: - CX_NFTS !=CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY !=DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY !=DEFAULT_COMM_L1_EXIT_LATENCY Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
14-12 PCIE_CAP_L0S_EXIT_LATENCY	LOs Exit Latency. For a description of this standard PCIe register field, see the PCI Express Specification. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is supported (possible) in the controller when one or more of the following expressions is true: - CX_NFTS !=CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY !=DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY !=DEFAULT_COMM_L1_EXIT_LATENCY Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of

Table continues on the next page...

Field	Function
	the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
11-10 PCIE_CAP_ACTIVE_STATE_LINK_PM_SUPPORT	Level of ASPM (Active State Power Management) Support. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
9-4 PCIE_CAP_MAX_LINK_WIDTH	Maximum Link Width. For a description of this standard PCIe register field, see the PCI Express Specification. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
3-0 PCIE_CAP_MAX_LINK_SPEED	Maximum Link Speed. For a description of this standard PCIe register field, see the PCI Express Specification. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

11.4.3.1.28 Link Control and Status Register. (LINK_CONTROL_LINK_STATUS_REG)

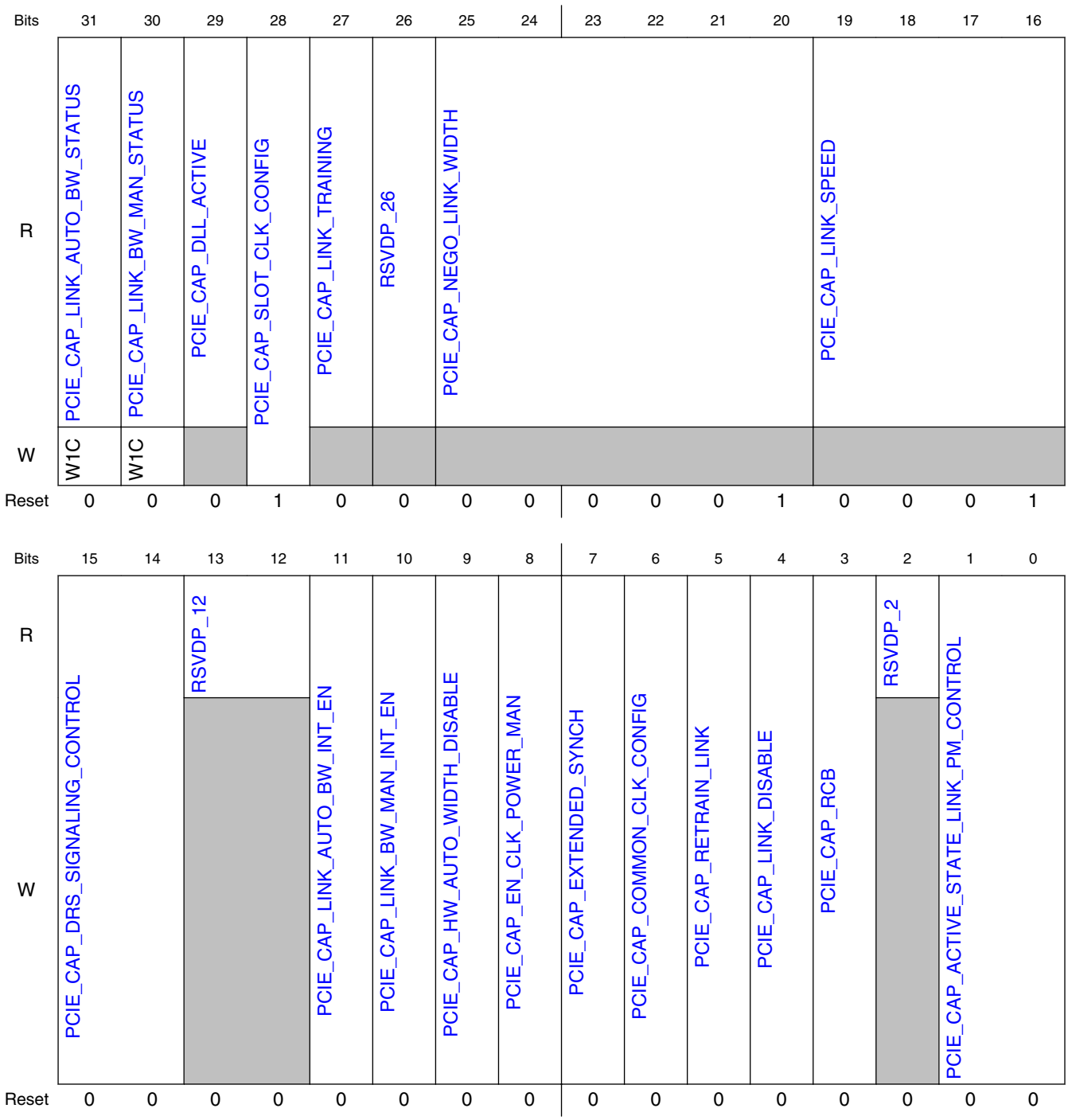
11.4.3.1.28.1 Offset

Register	Offset
LINK_CONTROL_LINK_STATUS_REG	80h

11.4.3.1.28.2 Function

Link Control and Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.28.3 Diagram



11.4.3.1.28.4 Fields

Field	Function
31 PCIE_CAP_LINK_AUTO_BW_STATUS	Link Autonomous Bandwidth Status. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG.
30 PCIE_CAP_LINK_BW_MAN_STATUS	Link Bandwidth Management Status. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG.
29 PCIE_CAP_DLL_ACTIVE	Data Link Layer Active. For a description of this standard PCIe register field, see the PCI Express Specification.
28 PCIE_CAP_SLOT_CLK_CONFIG	Slot Clock Configuration. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
27 PCIE_CAP_LINK_TRAINING	LTSSM is in Configuration or Recovery State. For a description of this standard PCIe register field, see the PCI Express Specification.
26 RSVDP_26	Reserved for future use.
25-20 PCIE_CAP_NEGO_LINK_WIDTH	Negotiated Link Width. For a description of this standard PCIe register field, see the PCI Express Specification.
19-16 PCIE_CAP_LINK_SPEED	Current Link Speed. For a description of this standard PCIe register field, see the PCI Express Specification.
15-14 PCIE_CAP_DRS_SIGNALING_CONTROL	DRS Signaling Control. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: LINK_CAPABILITIES2_REG.DRS_SUPPORTED ? RW : RO
13-12 RSVDP_12	Reserved for future use.
11 PCIE_CAP_LINK_AUTO_BW_INT_EN	Link Autonomous Bandwidth Management Interrupt Enable. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_LINK_BW_NOT_CAP ? RW : RO
10 PCIE_CAP_LINK_BW_MAN_INT_EN	Link Bandwidth Management Interrupt Enable. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_LINK_BW_NOT_CAP ? RW : RO
9	Hardware Autonomous Width Disable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W

Table continues on the next page...

Field	Function
PCIE_CAP_HW_AUTO_WIDTH_DISABLE	
8 PCIE_CAP_EN_CLK_POWER_MAN	Enable Clock Power Management. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_CLOCK_POWER_MAN field in LINK_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_CLOCK_POWER_MAN ? RWS : ROS Note: This register field is sticky.
7 PCIE_CAP_EXTENDED_SYNCH	Extended Synch. For a description of this standard PCIe register field, see the PCI Express Specification.
6 PCIE_CAP_COMMON_CLK_CONFIG	Common Clock Configuration. For a description of this standard PCIe register field, see the PCI Express Specification.
5 PCIE_CAP_RETRAIN_LINK	Initiate Link Retrain. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: see description
4 PCIE_CAP_LINK_DISABLE	Initiate Link Disable. For a description of this standard PCIe register field, see the PCI Express Specification. In a DSP that supports crosslink, the controller gates the write value with the CROSS_LINK_EN field in PORT_LINK_CTRL_OFF. Note: The access attributes of this field are as follows: - Dbi: CX_CROSSLINK_ENABLE=1 && PORT_LINK_CTRL_OFF.CROSS_LINK_EN=1|| CX_CROSSLINK_ENABLE=0 && dsp=1? RW : RO
3 PCIE_CAP_RCB	Read Completion Boundary (RCB). Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
2 RSVDP_2	Reserved for future use.
1-0 PCIE_CAP_ACTIVE_STATE_LINK_PM_CONTROL	Active State Power Management (ASPM) Control. Software must not enable L0s in either direction on a given Link unless components on both sides of the Link each support L0s; otherwise, the result is undefined. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.29 Slot Capabilities Register. (SLOT_CAPABILITIES_REG)

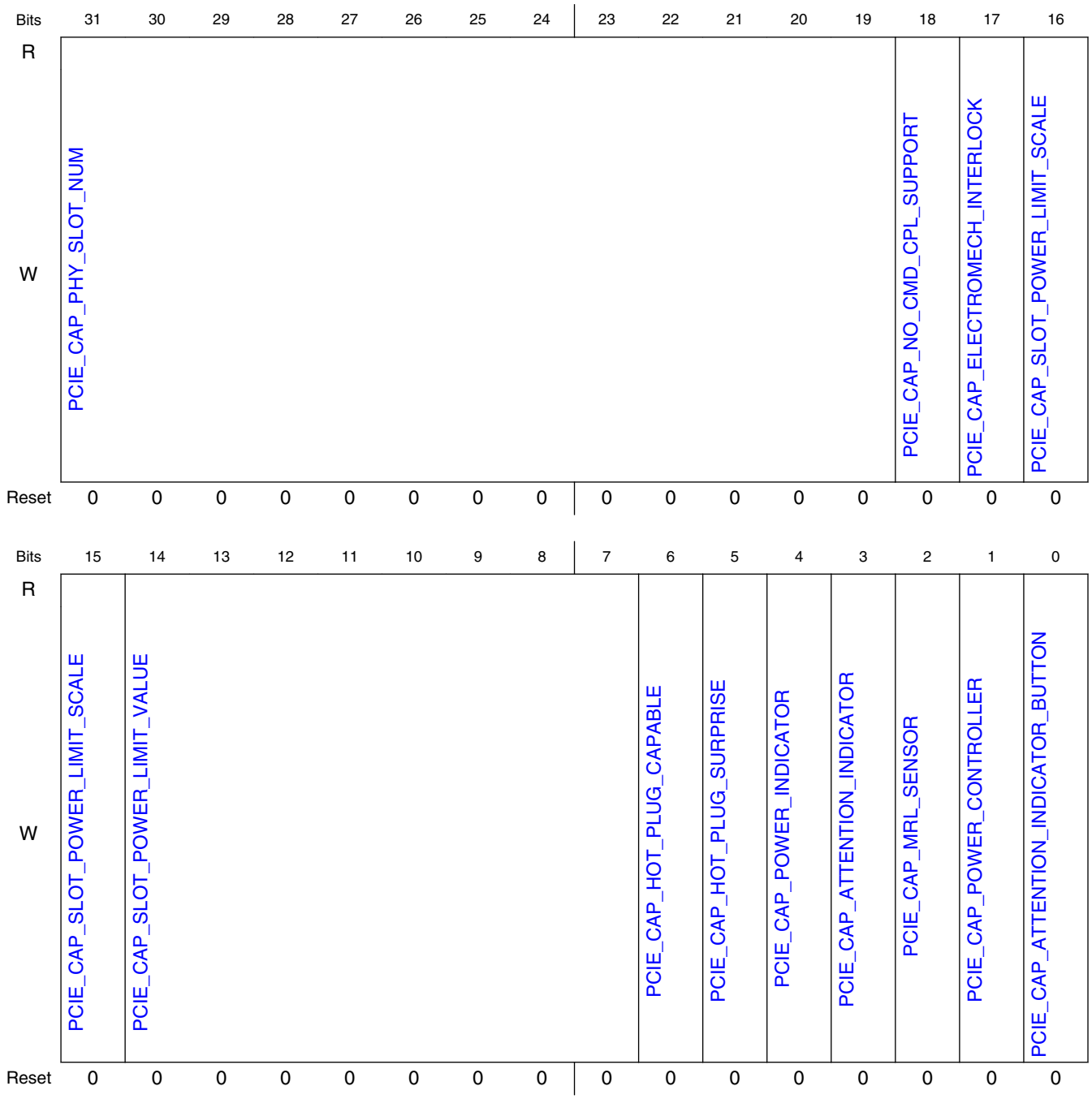
11.4.3.1.29.1 Offset

Register	Offset
SLOT_CAPABILITIES_REG	84h

11.4.3.1.29.2 Function

Slot Capabilities Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.29.3 Diagram



11.4.3.1.29.4 Fields

Field	Function
31-19 PCIE_CAP_PHY_SLOT_NUM	Physical Slot Number. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
18 PCIE_CAP_NO_CMD_CPL_SUPPORT	No Command Completed Support. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
17 PCIE_CAP_ELECTROMECH_INTERLOCK	Electromechanical Interlock Present. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
16-15 PCIE_CAP_SLOT_POWER_LIMIT_SCALE	Slot Power Limit Scale. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
14-7 PCIE_CAP_SLOT_POWER_LIMIT_VALUE	Slot Power Limit Value. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
6 PCIE_CAP_HOT_PLUG_CAPABLE	Hot Plug Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
5 PCIE_CAP_HOT_PLUG_SURPRISE	Hot Plug Surprise possible. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
4 PCIE_CAP_POWER_INDICATOR	Power Indicator Present. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
3 PCIE_CAP_ATTENTION_INDICATOR	Attention Indicator Present. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
2 PCIE_CAP_MRL_SENSOR	MRL Present. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
1 PCIE_CAP_POWER_CONTROLLER	Power Controller Present. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
0	Attention Button Present. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

Field	Function
PCIE_CAP_ATTENTION_INDICATOR_BUTTON	

11.4.3.1.30 Slot Control and Status Register. (SLOT_CONTROL_SLOT_STATUS)

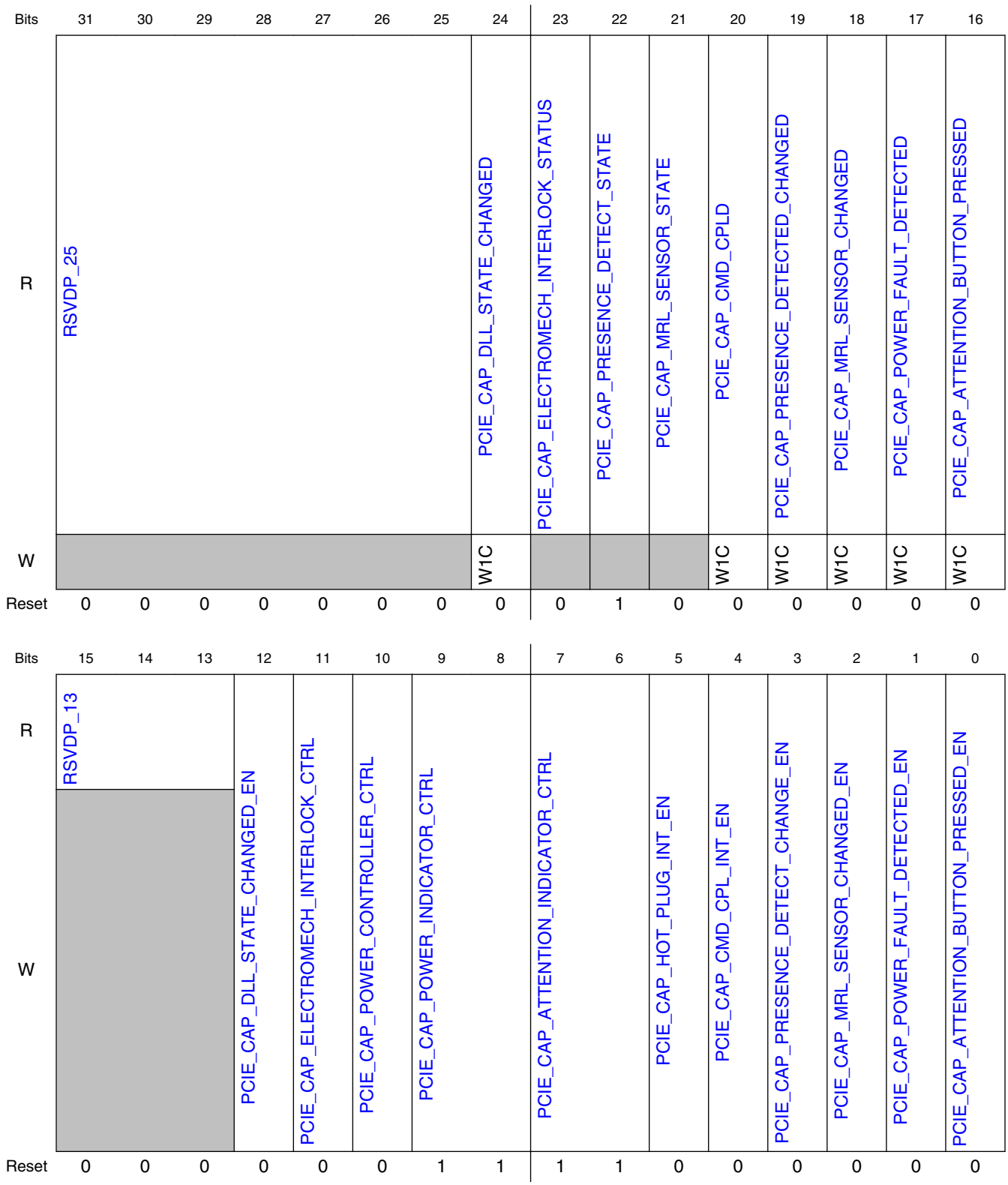
11.4.3.1.30.1 Offset

Register	Offset
SLOT_CONTROL_SLOT_STATUS	88h

11.4.3.1.30.2 Function

Slot Control and Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.30.3 Diagram



11.4.3.1.30.4 Fields

Field	Function
31-25 RSVDP_25	Reserved for future use.
24 PCIE_CAP_DLL_STATE_CHANGED	Data Link Layer State Changed. For a description of this standard PCIe register field, see the PCI Express Specification.
23 PCIE_CAP_ELECTROMECH_INTERLOCK_STATUS	Electromechanical Interlock Status. For a description of this standard PCIe register field, see the PCI Express Specification.
22 PCIE_CAP_PRESENCE_DETECT_STATE	Presence Detect State. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R
21 PCIE_CAP_MRL_SENSOR_STATE	MRL Sensor State. For a description of this standard PCIe register field, see the PCI Express Specification.
20 PCIE_CAP_CMD_CPLD	Command Completed. For a description of this standard PCIe register field, see the PCI Express Specification.
19 PCIE_CAP_PRESENCE_DETECTED_CHANGED	Presence Detect Changed. For a description of this standard PCIe register field, see the PCI Express Specification.
18 PCIE_CAP_MRL_SENSOR_CHANGED	MRL Sensor Changed. For a description of this standard PCIe register field, see the PCI Express Specification.
17 PCIE_CAP_POWER_FAULT_DETECTED	Power Fault Detected. For a description of this standard PCIe register field, see the PCI Express Specification.
16 PCIE_CAP_ATTENTION_BUTTON_PRESSED	Attention Button Pressed. For a description of this standard PCIe register field, see the PCI Express Specification.
15-13 RSVDP_13	Reserved for future use.
12	Data Link Layer State Changed Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

Table continues on the next page...

Field	Function
PCIE_CAP_DLL_STATE_CHANGED_EN	
11 PCIE_CAP_ELECTROMECH_INTERLOCK_CTRL	Electromechanical Interlock Control. For a description of this standard PCIe register field, see the PCI Express Specification.
10 PCIE_CAP_POWER_CONTROLLER_CTRL	Power Controller Control. For a description of this standard PCIe register field, see the PCI Express Specification.
9-8 PCIE_CAP_POWER_INDICATOR_CTRL	Power Indicator Control. For a description of this standard PCIe register field, see the PCI Express Specification.
7-6 PCIE_CAP_ATTENTION_INDICATOR_CTRL	Attention Indicator Control. For a description of this standard PCIe register field, see the PCI Express Specification.
5 PCIE_CAP_HOT_PLUG_INT_EN	Hot Plug Interrupt Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
4 PCIE_CAP_CMD_CPL_INT_EN	Command Completed Interrupt Enable. For a description of this standard PCIe register field, see the PCI Express Specification. Write value is gated with PCIE_CAP_NO_CMD_CPL_SUPPORT field in SLOT_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Dbi: SLOT_CAPABILITIES_REG.PCIE_CAP_NO_CMD_CPL_SUPPORT ? RO : RW
3 PCIE_CAP_PRESENCE_DETECT_CHANGE_EN	Presence Detect Changed Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
2 PCIE_CAP_MRL_SENSOR_CHANGED_EN	MRL Sensor Changed Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
1 PCIE_CAP_POWER_FAULT_DETECTED_EN	Power Fault Detected Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
0 PCIE_CAP_ATTENTION_BUTTON_PRESSED_EN	Attention Button Pressed Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.31 Root Control and Capabilities Register. (ROOT_CONTROL_ROOT_CAPABILITIES_REG)

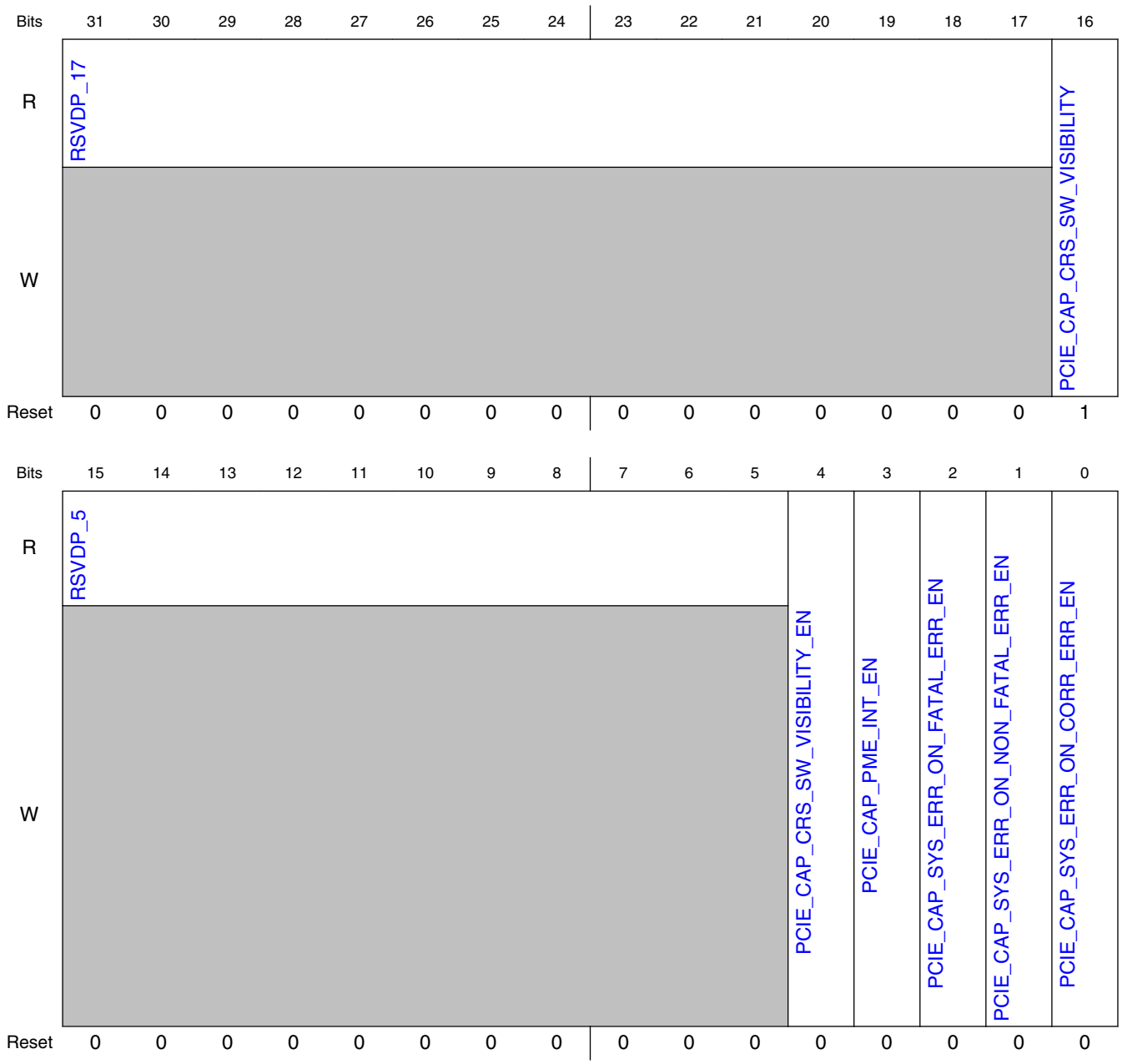
11.4.3.1.31.1 Offset

Register	Offset
ROOT_CONTROL_ROOT_CAPABILITIES_REG	8Ch

11.4.3.1.31.2 Function

Root Control and Capabilities Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.31.3 Diagram



11.4.3.1.31.4 Fields

Field	Function
31-17 RSVDP_17	Reserved for future use.
16	CRS Software Visibility Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W (Sticky) else R (Sticky) Note: This register field is sticky.

Table continues on the next page...

Field	Function
PCIE_CAP_CRS_SW_VISIBILITY	
15-5 RSVDP_5	Reserved for future use.
4 PCIE_CAP_CRS_SW_VISIBILITY_EN	Configuration Request Retry Status (CRS) Software Visibility Enable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: ROOT_CONTROL_ROOT_CAPABILITIES_REG.PCIE_CAP_CRS_SW_VISIBILITY ? RW : RO
3 PCIE_CAP_PME_INT_EN	PME Interrupt Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
2 PCIE_CAP_SYS_ERR_ON_FATAL_ERR_EN	System Error on Fatal Error Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
1 PCIE_CAP_SYS_ERR_ON_NON_FATAL_ERR_EN	System Error on Non-fatal Error Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
0 PCIE_CAP_SYS_ERR_ON_CORR_ERR_EN	System Error on Correctable Error Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.32 Root Status Register. (ROOT_STATUS_REG)

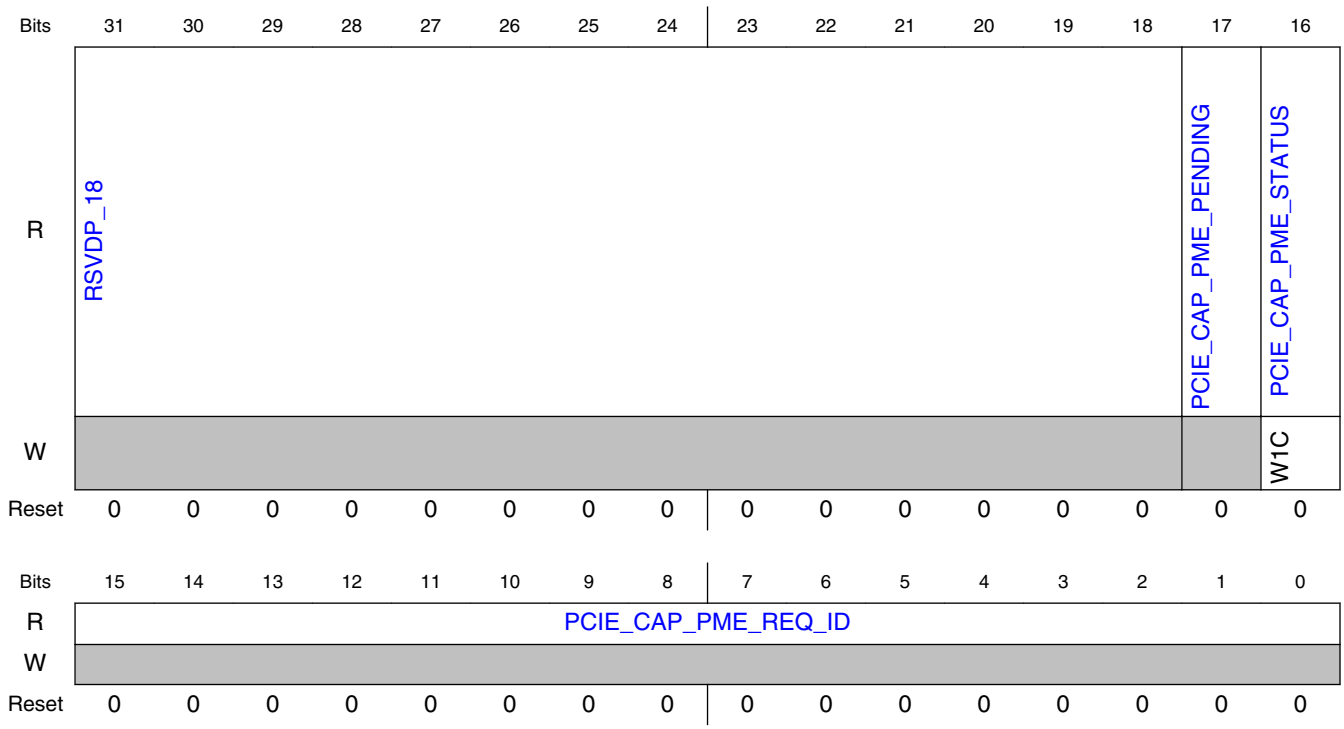
11.4.3.1.32.1 Offset

Register	Offset
ROOT_STATUS_REG	90h

11.4.3.1.32.2 Function

Root Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.32.3 Diagram



11.4.3.1.32.4 Fields

Field	Function
31-18 RSVDP_18	Reserved for future use.
17 PCIE_CAP_PME_PENDING	PME Pending. For a description of this standard PCIe register field, see the PCI Express Specification.
16 PCIE_CAP_PME_STATUS	PME Status. For a description of this standard PCIe register field, see the PCI Express Specification.
15-0 PCIE_CAP_PME_REQ_ID	PME Requester ID. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.33 Device Capabilities 2 Register. (DEVICE_CAPABILITIES2_REG)

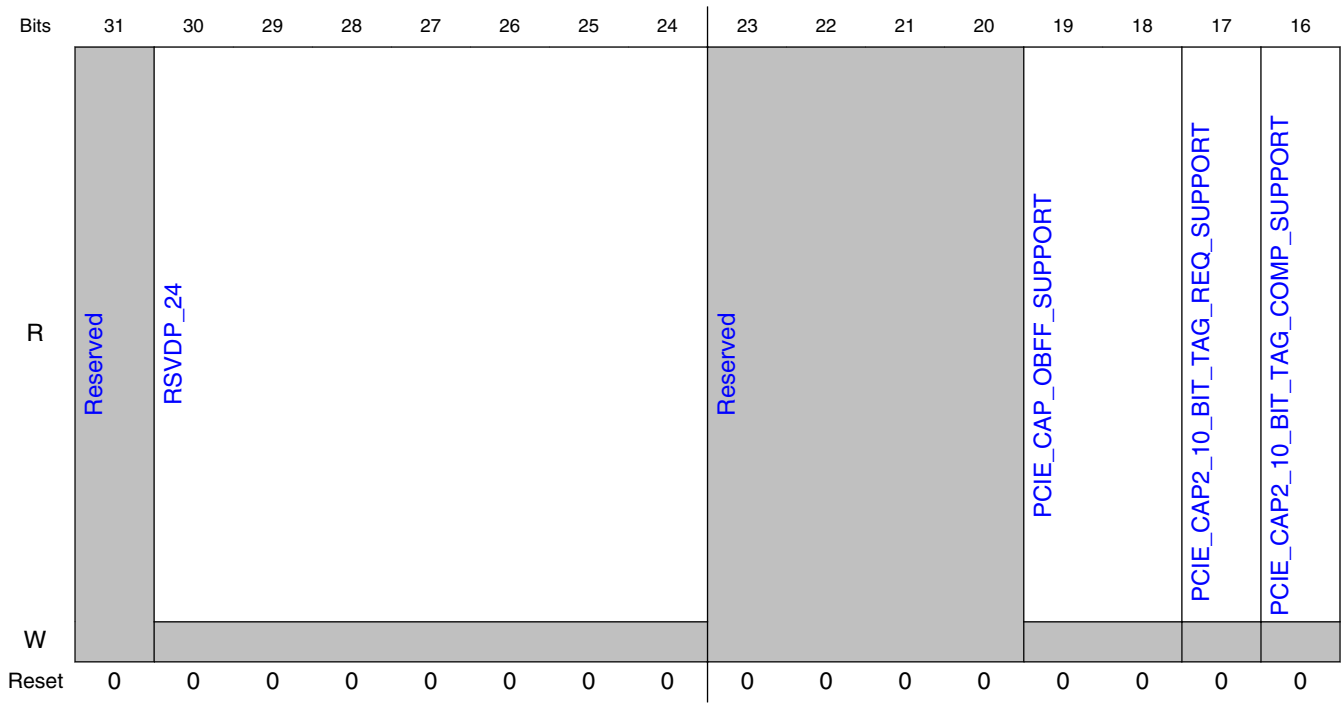
11.4.3.1.33.1 Offset

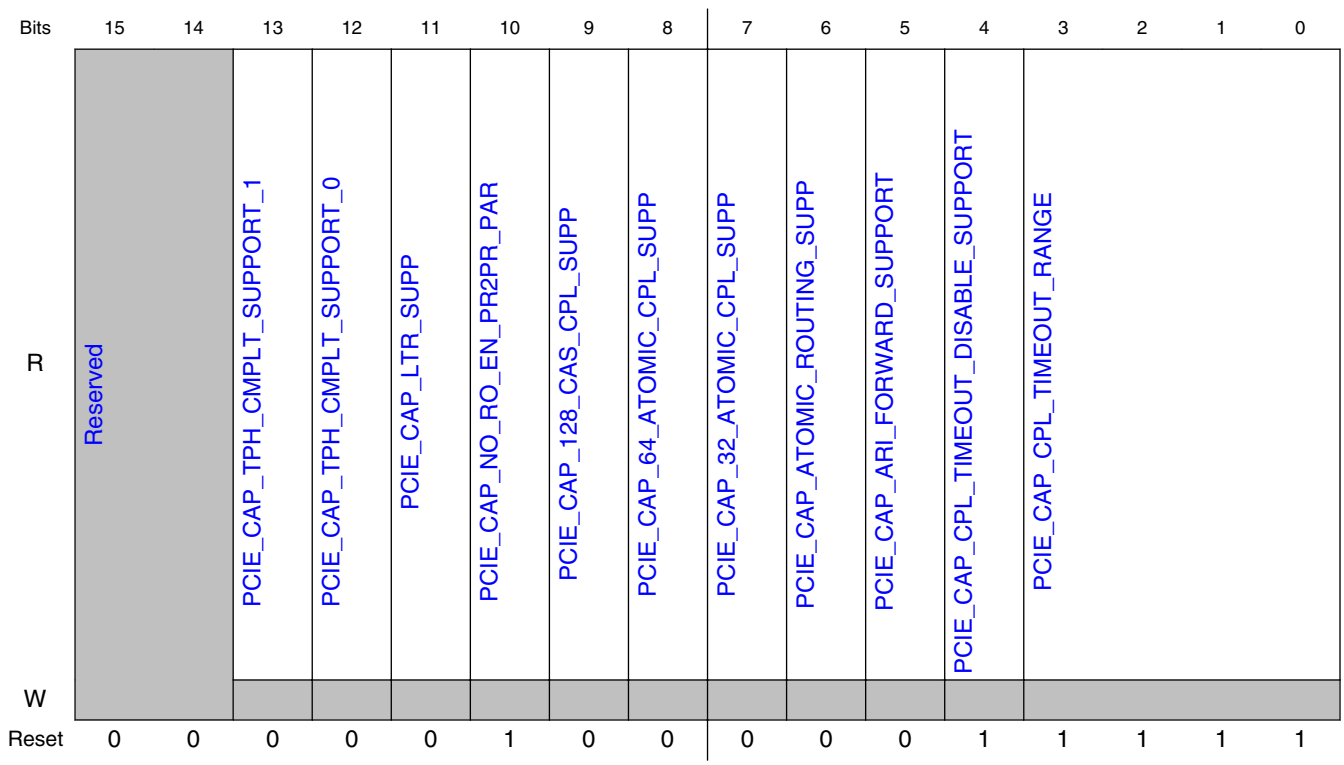
Register	Offset
DEVICE_CAPABILITIES 2_REG	94h

11.4.3.1.33.2 Function

Device Capabilities 2 Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.33.3 Diagram





11.4.3.1.33.4 Fields

Field	Function
31 —	Reserved
30-24 RSVDP_24	Reserved for future use.
23-20 —	Reserved
19-18 PCIE_CAP_OBFF_SUPPORT	(OBFF) Optimized Buffer Flush/fill Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
17 PCIE_CAP2_10_BIT_TAG_REQ_SUPPORT	10-Bit Tag Requester Supported. For a description of this standard PCIe register field, see the PCI Express Base Specification 4.0.
16 PCIE_CAP2_10_BIT_TAG_COMP_SUPPORT	10-Bit Tag Completer Supported. For a description of this standard PCIe register field, see the PCI Express Base Specification 4.0.
15-14 —	Reserved
13 PCIE_CAP_TPH_CMPLT_SUPPORT_1	TPH Completer Supported Bit 1. For a description of this standard PCIe register field, see the PCI Express Specification.
12 PCIE_CAP_TPH_CMPLT_SUPPORT_0	TPH Completer Supported Bit 0. For a description of this standard PCIe register field, see the PCI Express Specification.
11 PCIE_CAP_LTR_SUPP	LTR Mechanism Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
10 PCIE_CAP_NO_RO_EN_PR2PR_PAR	No Relaxed Ordering Enabled PR-PR Passing. For a description of this standard PCIe register field, see the PCI Express Specification.
9 PCIE_CAP_128_CAS_CPL_SUPP	128 Bit CAS Completer Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
8 PCIE_CAP_64_ATOMIC_CPL_SUPP	64 Bit AtomicOp Completer Supported. For a description of this standard PCIe register field, see the PCI Express Specification.

Table continues on the next page...

Field	Function
7 PCIE_CAP_32_ATOMIC_CPL_SUPP	32 Bit AtomicOp Completer Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
6 PCIE_CAP_ATOMIC_ROUTING_SUPP	Atomic Operation Routing Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
5 PCIE_CAP_ARI_FORWARD_SUPPORT	ARI Forwarding Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
4 PCIE_CAP_CPL_TIMEOUT_DISABLE_SUPPORT	Completion Timeout Disable Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
3-0 PCIE_CAP_CPL_TIMEOUT_RANGE	Completion Timeout Ranges Supported. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.34 Device Control 2 and Status 2 Register. (DEVICE_CONTROL2_DEVICE_STATUS2_REG)

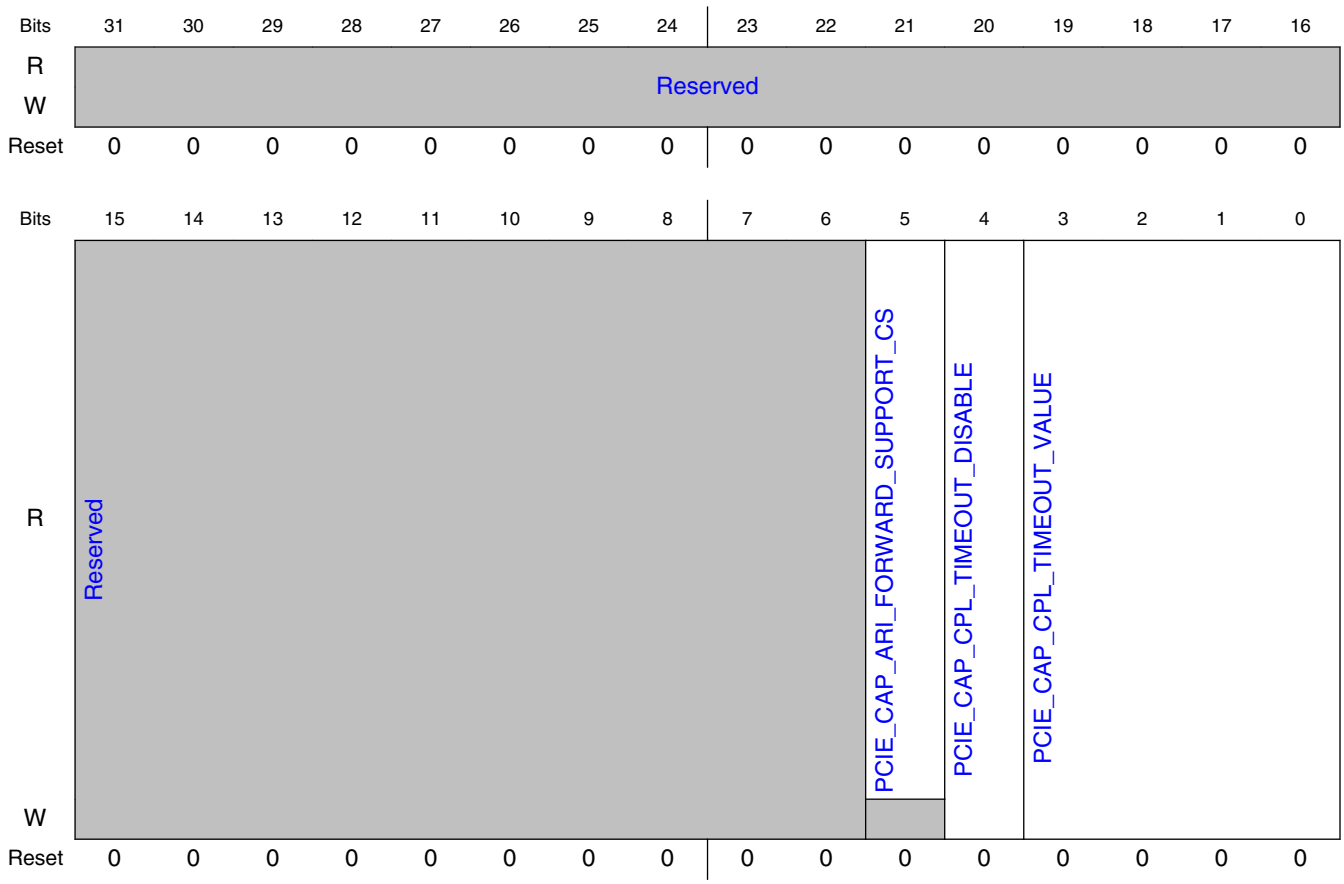
11.4.3.1.34.1 Offset

Register	Offset
DEVICE_CONTROL2_DEVICE_STATUS2_REG	98h

11.4.3.1.34.2 Function

Device Control 2 and Status 2 Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.34.3 Diagram



11.4.3.1.34.4 Fields

Field	Function
31-6 —	Reserved
5 PCIE_CAP_ARI_FORWARD_SUPPORT_CS	ARI Forwarding Enable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
4 PCIE_CAP_CPL_TIMEOUT_DISABLE	Completion Timeout Disable. For a description of this standard PCIe register field, see the PCI Express Specification.
3-0 PCIE_CAP_CPL_TIMEOUT_VALUE	Completion Timeout Value. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.35 Link Capabilities 2 Register. (LINK_CAPABILITIES2_REG)

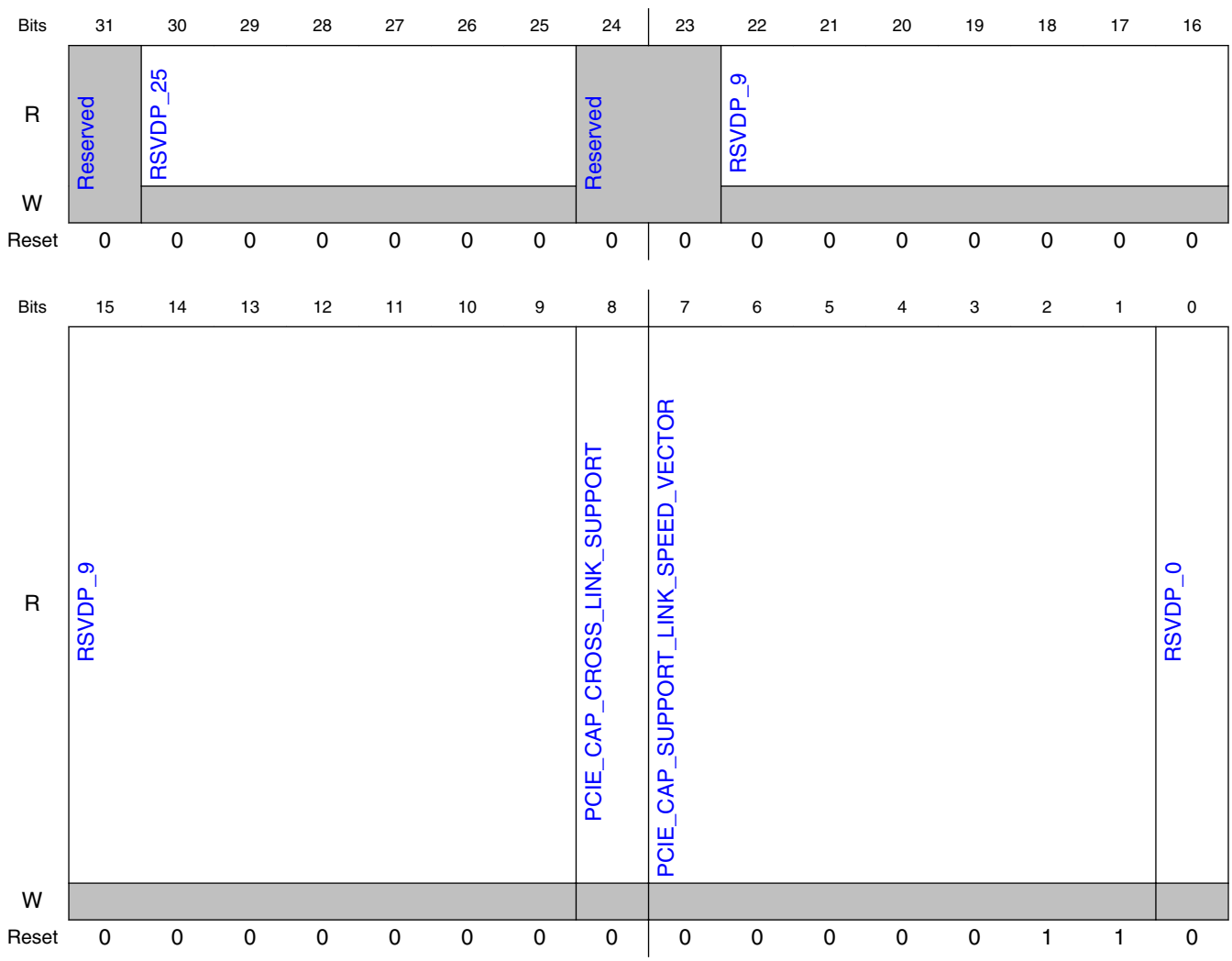
11.4.3.1.35.1 Offset

Register	Offset
LINK_CAPABILITIES2_REG	9Ch

11.4.3.1.35.2 Function

Link Capabilities 2 Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.35.3 Diagram



11.4.3.1.35.4 Fields

Field	Function
31 —	Reserved
30-25 RSVDP_25	Reserved for future use.
24-23 —	Reserved
22-9 RSVDP_9	Reserved for future use.
8 PCIE_CAP_CROSS_LINK_SUPPORT	Cross Link Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
7-1 PCIE_CAP_SUPPORT_LINK_SPEED_VECTOR	Supported Link Speeds Vector. For a description of this standard PCIe register field, see the PCI Express Specification. This field has a default of (PCIE_CAP_MAX_LINK_SPEED == 0100) ? 0001111 : (PCIE_CAP_MAX_LINK_SPEED == 0011) ? 0000111 : (PCIE_CAP_MAX_LINK_SPEED == 0010) ? 0000011 : 0000001 where PCIE_CAP_MAX_LINK_SPEED is a field in the LINK_CAPABILITIES_REG register.
0 RSVDP_0	Reserved for future use.

11.4.3.1.36 Link Control 2 and Status 2 Register. (LINK_CONTROL2_LINK_STATUS2_REG)

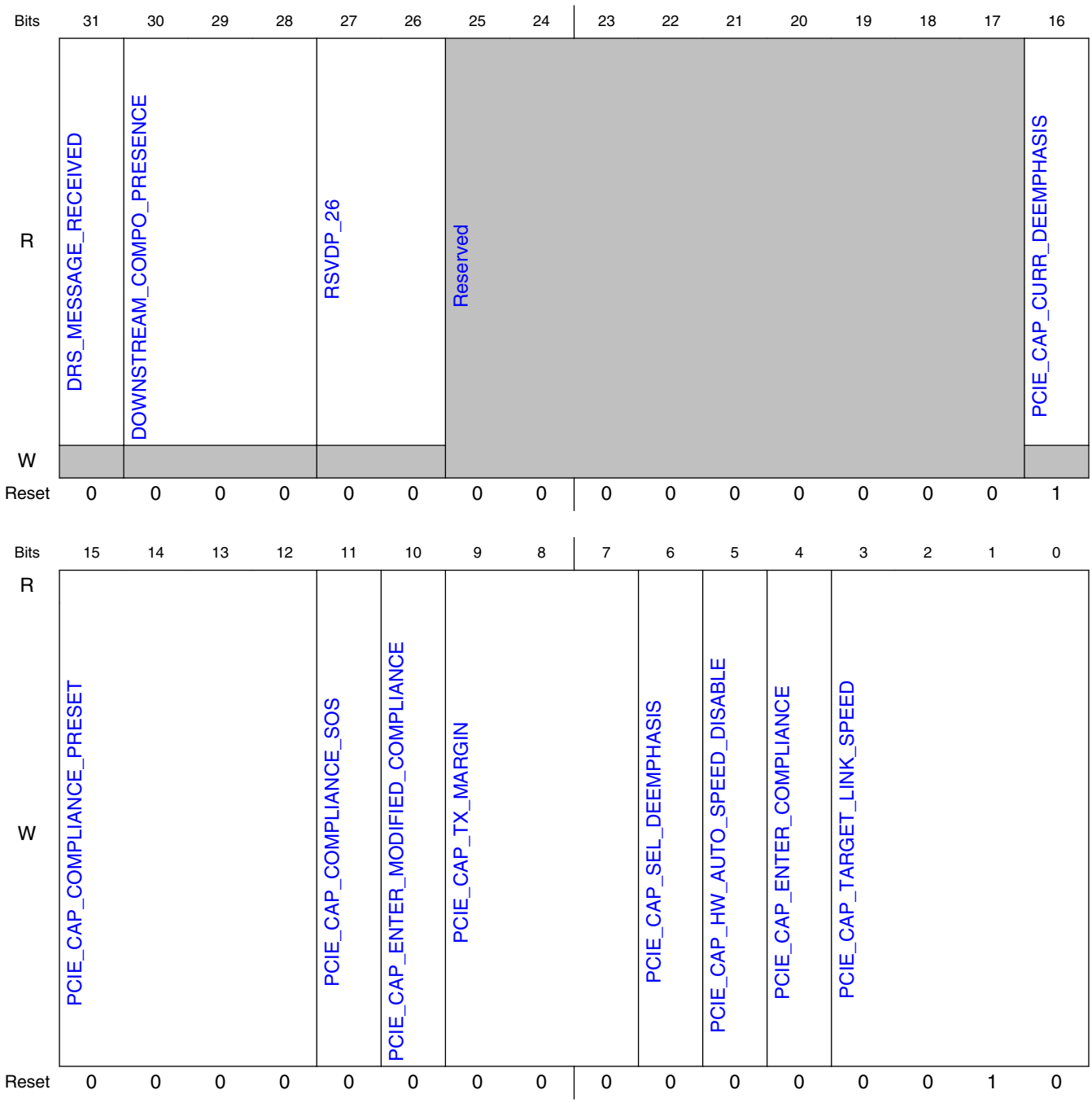
11.4.3.1.36.1 Offset

Register	Offset
LINK_CONTROL2_LINK_STATUS2_REG	A0h

11.4.3.1.36.2 Function

Link Control 2 and Status 2 Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.36.3 Diagram



11.4.3.1.36.4 Fields

Field	Function
31 DRS_MESSAGE_RECEIVED	DRS Message Received. For a description of this standard PCIe register field, see the PCI Express Base Specification 4.0. Note: The access attributes of this field are as follows: - Dbi: RW1C

Table continues on the next page...

Field	Function
30-28 DOWNSTREAM_COMPO_PRESENCE	Downstream Component Presence. For a description of this standard PCIe register field, see the PCI Express Base Specification 4.0.
27-26 RSVDP_26	Reserved for future use.
25-17 —	Reserved
16 PCIE_CAP_CURR_DEEMPHASIS	Current De-emphasis Level. For a description of this standard PCIe register field, see the PCI Express Specification. In M-PCIe mode this register is always 0x0. In C-PCIe mode, its contents are derived by sampling the PIPE
15-12 PCIE_CAP_COMPLIANCE_PRESET	Sets Compliance Preset/De-emphasis for 5 GT/s and 8 GT/s. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
11 PCIE_CAP_COMPLIANCE_SOS	Sets Compliance Skip Ordered Sets transmission. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
10 PCIE_CAP_ENTER_MODIFIED_COMPLIANCE	Enter Modified Compliance. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
9-7 PCIE_CAP_TX_MARGIN	Controls Transmit Margin for Debug or Compliance. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
6 PCIE_CAP_SEL_DEEMPHASIS	Controls Selectable De-emphasis for 5 GT/s. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
5 PCIE_CAP_HW_AUTO_SPEED_DISABLE	Hardware Autonomous Speed Disable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
4 PCIE_CAP_ENTER_COMPLIANCE	Enter Compliance Mode. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
3-0 PCIE_CAP_TARGET_LINK_SPEED	Target Link Speed. For a description of this standard PCIe register field, see the PCI Express Specification. In M-PCIe mode, the contents of this field are derived from other registers. Note: This register field is sticky.

11.4.3.1.37 Advanced Error Reporting Extended Capability Header. (AER_EXT_CAP_HDR_OFF)

11.4.3.1.37.1 Offset

Register	Offset
AER_EXT_CAP_HDR_OFF	100h

11.4.3.1.37.2 Function

Advanced Error Reporting Extended Capability Header. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NEXT_OFFSET												CAP_VERSION			
Reset	0	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CAP_ID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

11.4.3.1.37.4 Fields

Field	Function
31-20 NEXT_OFFSET	Next Capability Offset. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19-16 CAP_VERSION	Capability Version. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15-0 CAP_ID	AER Extended Capability ID. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

11.4.3.1.38 Uncorrectable Error Status Register. (UNCORR_ERR_STATUS_OFF)

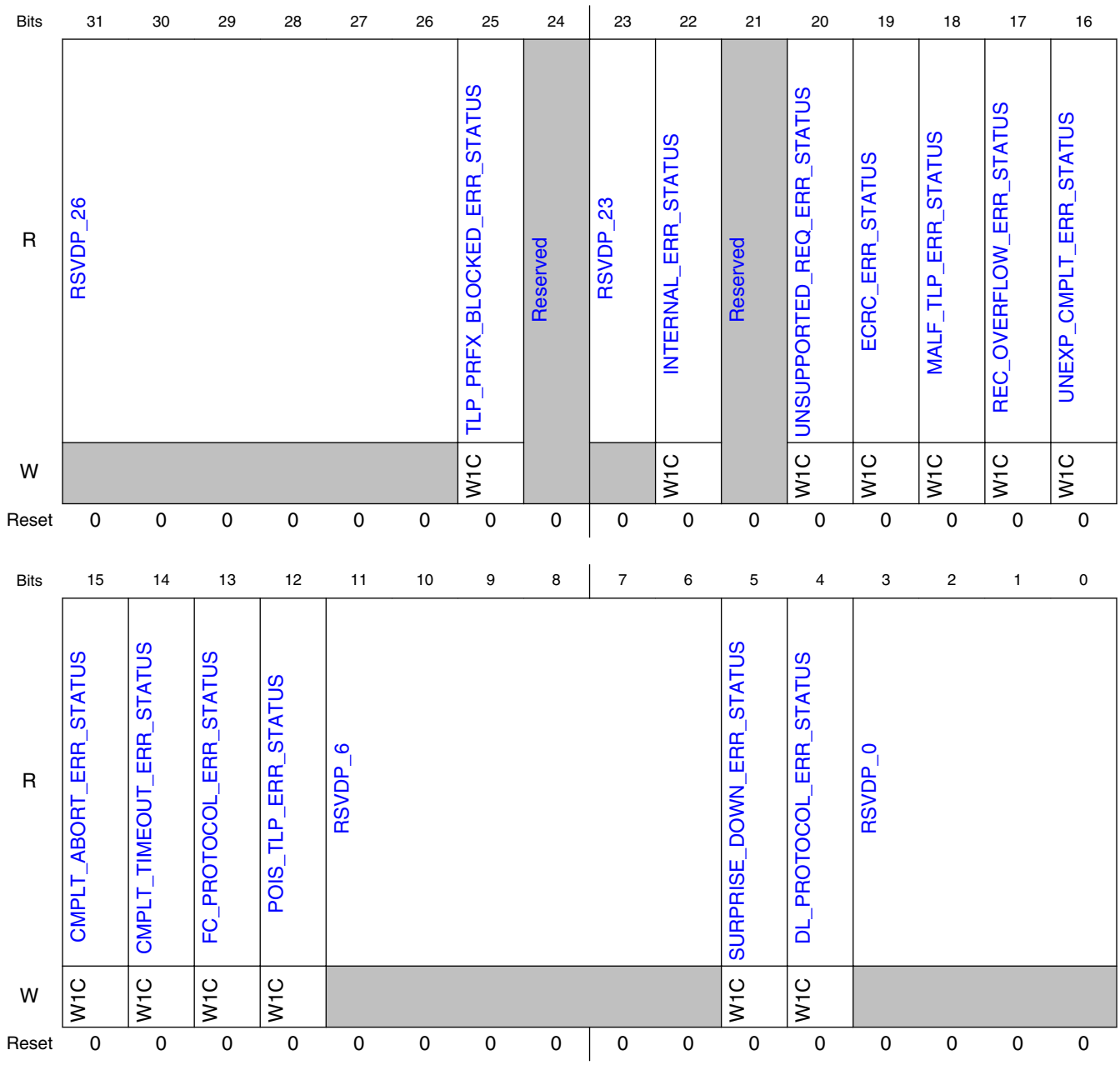
11.4.3.1.38.1 Offset

Register	Offset
UNCORR_ERR_STATUS_OFF	104h

11.4.3.1.38.2 Function

Uncorrectable Error Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.38.3 Diagram



11.4.3.1.38.4 Fields

Field	Function
31-26 RSVDP_26	Reserved for future use.
25	TLP Prefix Blocked Error Status. For a description of this standard PCIe register field, see the PCI Express Specification. Note: Not supported.

Table continues on the next page...

Field	Function
TLP_PRFX_BLOCKED_ERR_STATUS	
24 —	Reserved
23 RSVDP_23	Reserved for future use.
22 INTERNAL_ERR_STATUS	Uncorrectable Internal Error Status. For a description of this standard PCIe register field, see the PCI Express Specification. The controller sets this bit when your application asserts app_err_bus[9]. It does not set this bit when it detects internal uncorrectable internal errors such as parity and ECC failures. You should use the outputs from these errors to drive the app_err_bus[9] input. For more details, see the "Data Integrity (Wire, Datapath, and RAM Protection)" section in the Databook.
21 —	Reserved
20 UNSUPPORTED_REQ_ERR_STATUS	Unsupported Request Error Status. For a description of this standard PCIe register field, see the PCI Express Specification.
19 ECRC_ERR_STATUS	ECRC Error Status. For a description of this standard PCIe register field, see the PCI Express Specification.
18 MALF_TLP_ERR_STATUS	Malformed TLP Status. For a description of this standard PCIe register field, see the PCI Express Specification.
17 REC_OVERFLOW_ERR_STATUS	Receiver Overflow Status. For a description of this standard PCIe register field, see the PCI Express Specification.
16 UNEXP_CMPLT_ERR_STATUS	Unexpected Completion Status. For a description of this standard PCIe register field, see the PCI Express Specification.
15 CMPLT_ABORT_ERR_STATUS	Completer Abort Status. For a description of this standard PCIe register field, see the PCI Express Specification.
14 CMPLT_TIMEOUT_ERR_STATUS	Completion Timeout Status. For a description of this standard PCIe register field, see the PCI Express Specification.
13 FC_PROTOCOL_ERR_STATUS	Flow Control Protocol Error Status. For a description of this standard PCIe register field, see the PCI Express Specification.
12 POIS_TLP_ERR_STATUS	Poisoned TLP Status. For a description of this standard PCIe register field, see the PCI Express Specification.
11-6	Reserved for future use.

Table continues on the next page...

Field	Function
RSVDP_6	
5 SURPRISE_DOWN_ERR_STATUS	Surprise Down Error Status (Optional). For a description of this standard PCIe register field, see the PCI Express Specification.
4 DL_PROTOCOL_ERR_STATUS	Data Link Protocol Error Status. For a description of this standard PCIe register field, see the PCI Express Specification.
3-0 RSVDP_0	Reserved for future use.

11.4.3.1.39 Uncorrectable Error Mask Register. (UNCORR_ERR_MASK_OFF)

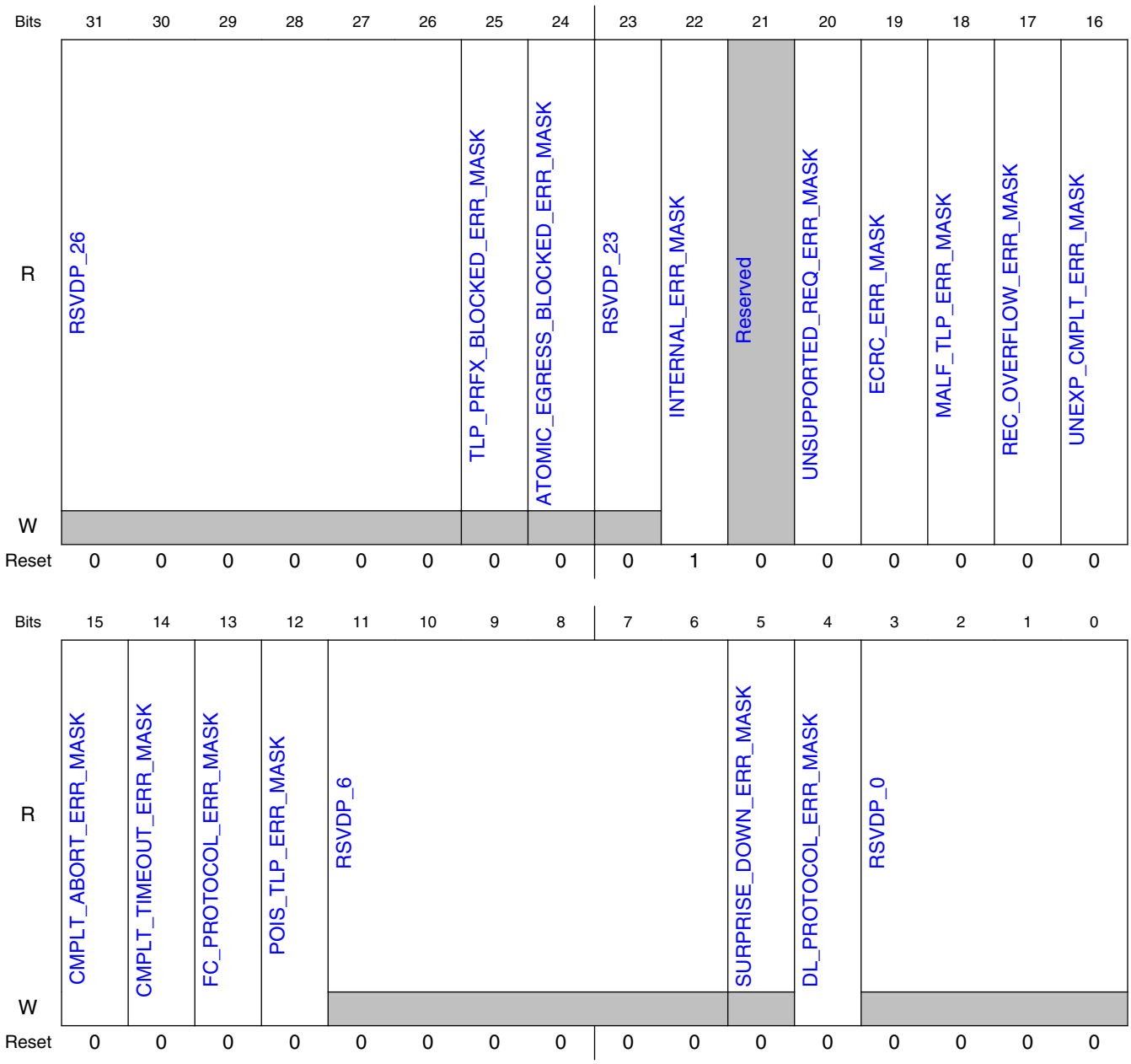
11.4.3.1.39.1 Offset

Register	Offset
UNCORR_ERR_MASK_OFF	108h

11.4.3.1.39.2 Function

Uncorrectable Error Mask Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.39.3 Diagram



11.4.3.1.39.4 Fields

Field	Function
31-26 RSVDP_26	Reserved for future use.
25	TLP Prefix Blocked Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: Not supported. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.

Table continues on the next page...

Field	Function
TLP_PRFX_BLOCKED_ERR_MASK	
24 ATOMIC_EGRESS_BLOCKED_ERR_MASK	AtomicOp Egress Block Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
23 RSVDP_23	Reserved for future use.
22 INTERNAL_ERR_MASK	Uncorrectable Internal Error Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
21 —	Reserved
20 UNSUPPORTED_REQ_ERR_MASK	Unsupported Request Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
19 ECRC_ERR_MASK	ECRC Error Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
18 MALF_TLP_ERR_MASK	Malformed TLP Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
17 REC_OVERFLOW_ERR_MASK	Receiver Overflow Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
16 UNEXP_CMPLT_ERR_MASK	Unexpected Completion Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15 CMPLT_ABORT_ERR_MASK	Completer Abort Error Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
14 CMPLT_TIMEOUT_ERR_MASK	Completion Timeout Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
13 FC_PROTOCOL_ERR_MASK	Flow Control Protocol Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
12 POIS_TLP_ERR_MASK	Poisoned TLP Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
11-6 RSVDP_6	Reserved for future use.

Table continues on the next page...

Field	Function
5 SURPRISE_DOWN_ERR_MASK	Surprise Down Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_SURPRISE_DOWN_ERR_REP_CAP ? RW : RO Note: This register field is sticky.
4 DL_PROTOCOL_ERR_MASK	Data Link Protocol Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
3-0 RSVDP_0	Reserved for future use.

11.4.3.1.40 Uncorrectable Error Severity Register. (UNCORR_ERR_SEV_OFF)

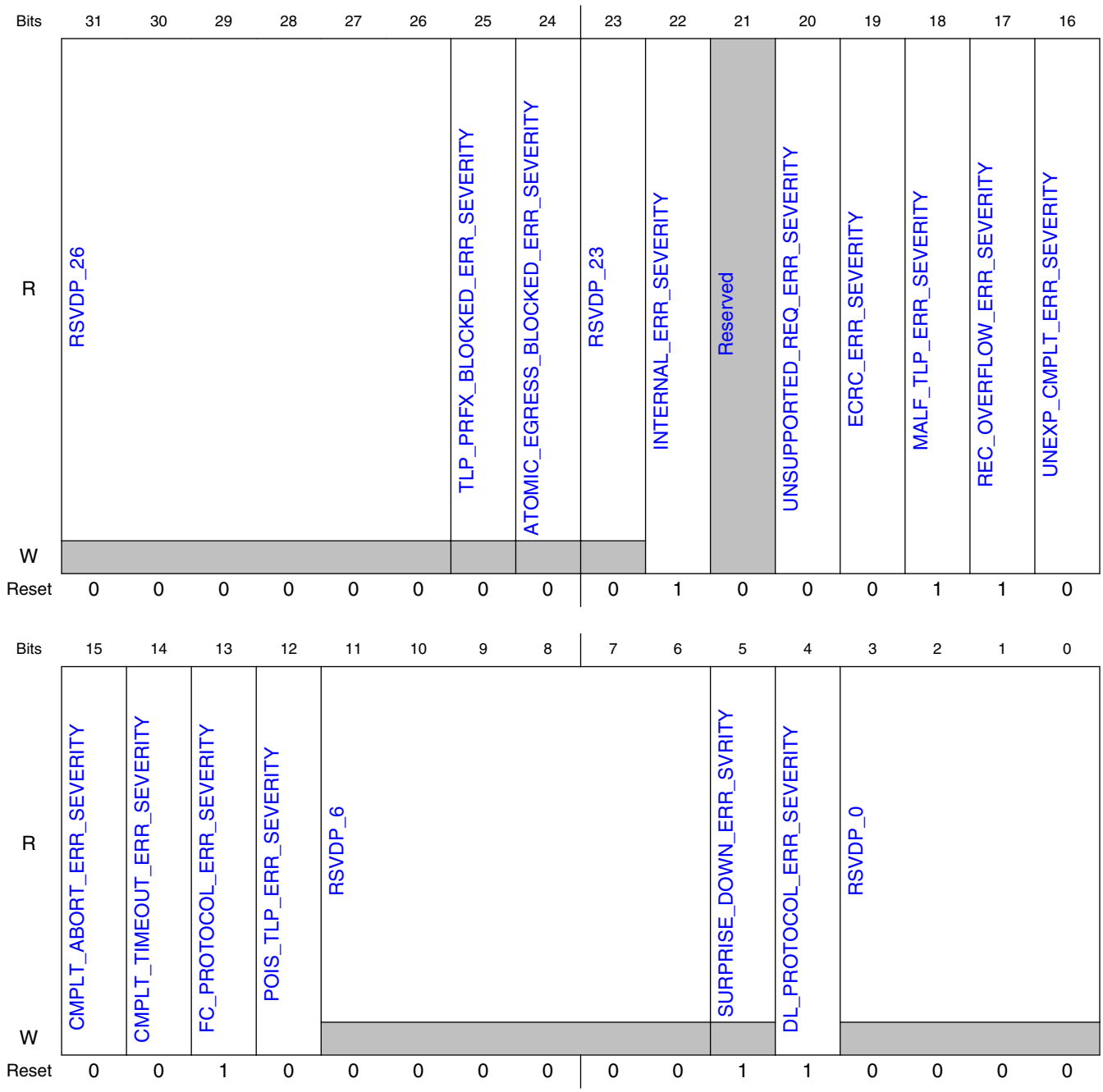
11.4.3.1.40.1 Offset

Register	Offset
UNCORR_ERR_SEV_OFF	10Ch

11.4.3.1.40.2 Function

Uncorrectable Error Severity Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.40.3 Diagram



11.4.3.1.40.4 Fields

Field	Function
31-26 RSVDP_26	Reserved for future use.

Table continues on the next page...

Field	Function
25 TLP_PRFX_BLOCKED_ERR_SEVERITY	TLP Prefix Blocked Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: Not supported. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
24 ATOMIC_EGRESS_BLOCKED_ERR_SEVERITY	AtomicOp Egress Blocked Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
23 RSVDP_23	Reserved for future use.
22 INTERNAL_ERR_SEVERITY	Uncorrectable Internal Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
21 —	Reserved
20 UNSUPPORTED_REQ_ERR_SEVERITY	Unsupported Request Error Severity. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
19 ECRC_ERR_SEVERITY	ECRC Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
18 MALF_TLP_ERR_SEVERITY	Malformed TLP Severity. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
17 REC_OVERFLOW_ERR_SEVERITY	Receiver Overflow Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
16 UNEXP_CMPLT_ERR_SEVERITY	Unexpected Completion Error Severity. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15 CMPLT_ABORT_ERR_SEVERITY	Completer Abort Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
14 CMPLT_TIMEOUT_ERR_SEVERITY	Completion Timeout Error Severity. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
13 FC_PROTOCOL_ERR_SEVERITY	Flow Control Protocol Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
12 POIS_TLP_ERR_SEVERITY	Poisoned TLP Severity. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
11-6 RSVDP_6	Reserved for future use.
5 SURPRISE_DOWN_ERR_SVRITY	Surprise Down Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_SURPRISE_DOWN_ERR_REP_CAP ? RW : RO Note: This register field is sticky.
4 DL_PROTOCOL_ERR_SEVERITY	Data Link Protocol Error Severity. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
3-0 RSVDP_0	Reserved for future use.

11.4.3.1.41 Correctable Error Status Register. (CORR_ERR_STATUS_OFF)

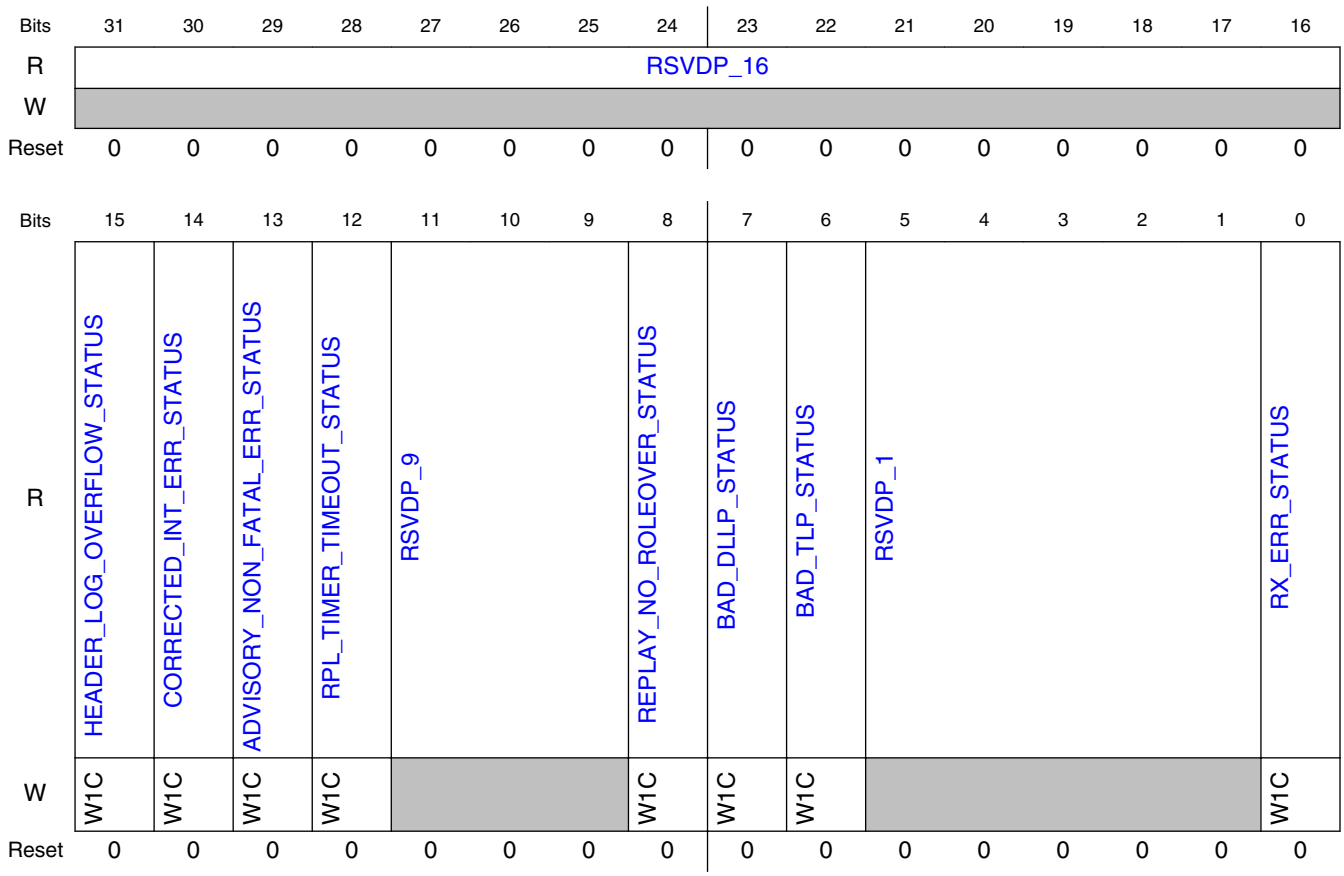
11.4.3.1.41.1 Offset

Register	Offset
CORR_ERR_STATUS_OFF	110h

11.4.3.1.41.2 Function

Correctable Error Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.41.3 Diagram



11.4.3.1.41.4 Fields

Field	Function
31-16 RSVDP_16	Reserved for future use.
15 HEADER_LOG_OVERFLOW_STATUS	Header Log Overflow Error Status (Optional). For a description of this standard PCIe register field, see the PCI Express Specification.
14 CORRECTED_INT_ERR_STATUS	Corrected Internal Error Status (Optional). For a description of this standard PCIe register field, see the PCI Express Specification.
13 ADVISORY_NON_FATAL_ERR_STATUS	Advisory Non-Fatal Error Status. For a description of this standard PCIe register field, see the PCI Express Specification.
12	Replay Timer Timeout Status. For a description of this standard PCIe register field, see the PCI Express Specification.

Table continues on the next page...

Field	Function
RPL_TIMER_TIMEOUT_STATUS	
11-9 RSVDP_9	Reserved for future use.
8 REPLAY_NO_ROLEOVER_STATUS	REPLAY_NUM Rollover Status. For a description of this standard PCIe register field, see the PCI Express Specification.
7 BAD_DLLP_STATUS	Bad DLLP Status. For a description of this standard PCIe register field, see the PCI Express Specification.
6 BAD_TLP_STATUS	Bad TLP Status. For a description of this standard PCIe register field, see the PCI Express Specification.
5-1 RSVDP_1	Reserved for future use.
0 RX_ERR_STATUS	Receiver Error Status (Optional). For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.42 Correctable Error Mask Register. (CORR_ERR_MASK_OFF)

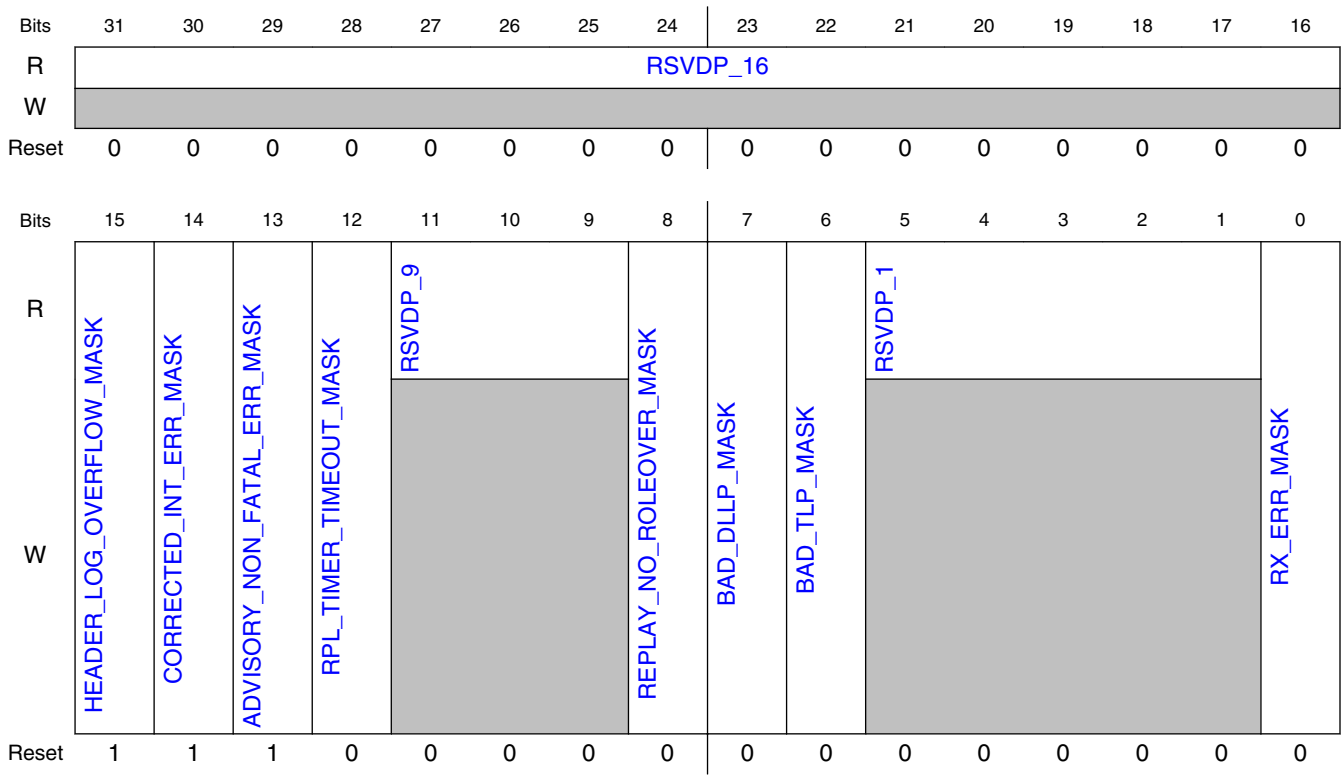
11.4.3.1.42.1 Offset

Register	Offset
CORR_ERR_MASK_OFF	114h

11.4.3.1.42.2 Function

Correctable Error Mask Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.42.3 Diagram



11.4.3.1.42.4 Fields

Field	Function
31-16 RSVDP_16	Reserved for future use.
15 HEADER_LOG_OVERFLOW_MASK	Header Log Overflow Error Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
14 CORRECTED_INT_ERR_MASK	Corrected Internal Error Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
13 ADVISORY_NON_FATAL_ERR_MASK	Advisory Non-Fatal Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
12 RPL_TIMER_TIMEOUT_MASK	Replay Timer Timeout Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
11-9 RSVDP_9	Reserved for future use.

Table continues on the next page...

Field	Function
8 REPLAY_NO_ROLEOVER_MASK	REPLAY_NUM Rollover Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7 BAD_DLLP_MASK	Bad DLLP Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
6 BAD_TLP_MASK	Bad TLP Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
5-1 RSVDP_1	Reserved for future use.
0 RX_ERR_MASK	Receiver Error Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.4.3.1.43 Advanced Error Capabilities and Control Register. (ADV_ERR_CAP_CTRL_OFF)

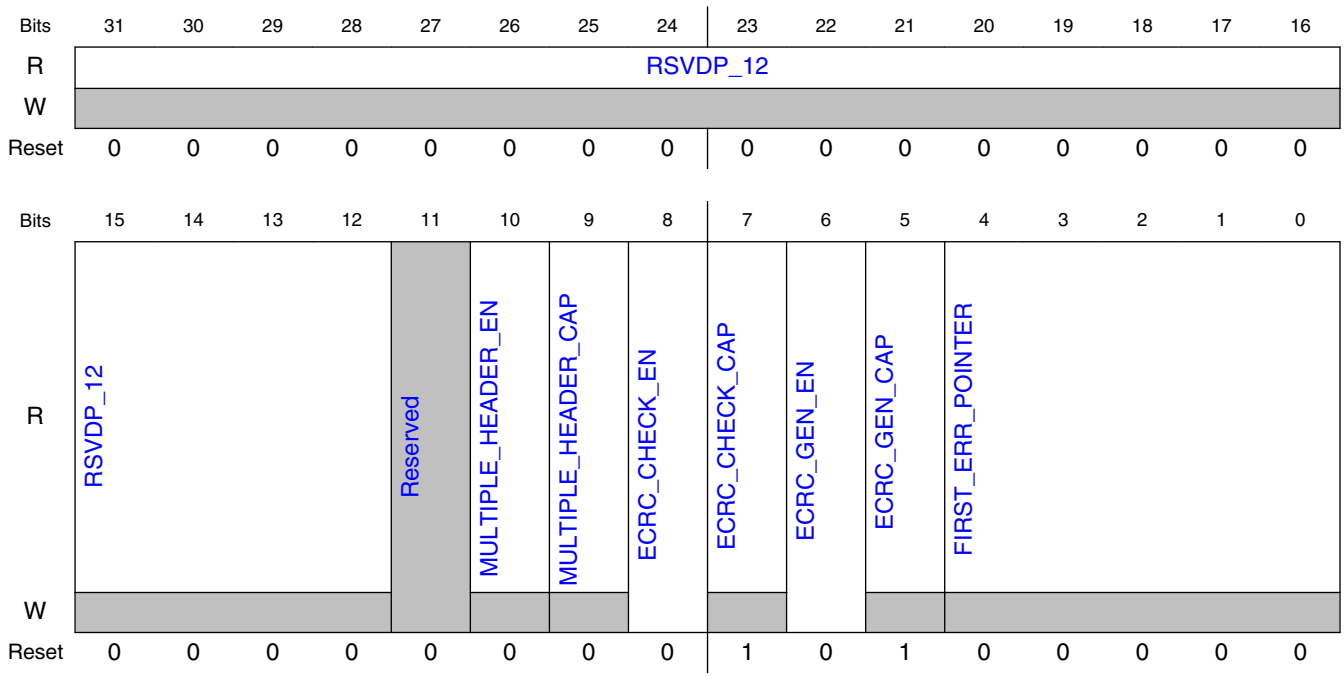
11.4.3.1.43.1 Offset

Register	Offset
ADV_ERR_CAP_CTRL_OFF	118h

11.4.3.1.43.2 Function

Advanced Error Capabilities and Control Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.43.3 Diagram



11.4.3.1.43.4 Fields

Field	Function
31-12 RSVDP_12	Reserved for future use.
11 —	Reserved
10 MULTIPLE_HEADER_EN	Multiple Header Recording Enable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
9 MULTIPLE_HEADER_CAP	Multiple Header Recording Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
8 ECRC_CHECK_EN	ECRC Check Enable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7 ECRC_CHECK_CAP	ECRC Check Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
6 ECRC_GEN_EN	ECRC Generation Enable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
5 ECRC_GEN_CAP	ECRC Generation Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
4-0 FIRST_ERR_POINTER	First Error Pointer. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.4.3.1.44 Header Log Register 0. (HDR_LOG_0_OFF)

11.4.3.1.44.1 Offset

Register	Offset
HDR_LOG_0_OFF	11Ch

11.4.3.1.44.2 Function

Header Log Register 0. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.44.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FIRST_DWORD_FOURTH_BYTE								FIRST_DWORD_THIRD_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIRST_DWORD_SECOND_BYTE								FIRST_DWORD_FIRST_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.44.4 Fields

Field	Function
31-24 FIRST_DWORD_FOURTH_BYTE	Byte 3 of Header log register of First 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
23-16 FIRST_DWORD_THIRD_BYTE	Byte 2 of Header log register of First 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 FIRST_DWORD_SECOND_BYTE	Byte 1 of Header log register of First 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 FIRST_DWORD_FIRST_BYTE	Byte 0 of Header log register of First 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.4.3.1.45 Header Log Register 1. (HDR_LOG_1_OFF)

11.4.3.1.45.1 Offset

Register	Offset
HDR_LOG_1_OFF	120h

11.4.3.1.45.2 Function

Header Log Register 1. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.45.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SECOND_DWORD_FOURTH_BYTE								SECOND_DWORD_THIRD_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SECOND_DWORD_SECOND_BYTE								SECOND_DWORD_FIRST_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.45.4 Fields

Field	Function
31-24 SECOND_DWORD_FOURTH_BYTE	Byte 3 of Header log register of Second 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
23-16 SECOND_DWORD_THIRD_BYTE	Byte 2 of Header log register of Second 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 SECOND_DWORD_SECOND_BYTE	Byte 1 of Header log register of Second 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 SECOND_DWORD_FIRST_BYTE	Byte 0 of Header log register of Second 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.4.3.1.46 Header Log Register 2. (HDR_LOG_2_OFF)

11.4.3.1.46.1 Offset

Register	Offset
HDR_LOG_2_OFF	124h

11.4.3.1.46.2 Function

Header Log Register 2. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.46.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	THIRD_DWORD_FOURTH_BYTE								THIRD_DWORD_THIRD_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	THIRD_DWORD_SECOND_BYTE								THIRD_DWORD_FIRST_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.46.4 Fields

Field	Function
31-24 THIRD_DWORD_FOURTH_BYTE	Byte 3 of Header log register of Third 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
23-16 THIRD_DWORD_THIRD_BYTE	Byte 2 of Header log register of Third 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 THIRD_DWORD_SECOND_BYTE	Byte 1 of Header log register of Third 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 THIRD_DWORD_FIRST_BYTE	Byte 0 of Header log register of Third 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.4.3.1.47 Header Log Register 3. (HDR_LOG_3_OFF)

11.4.3.1.47.1 Offset

Register	Offset
HDR_LOG_3_OFF	128h

11.4.3.1.47.2 Function

Header Log Register 3. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.47.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FOURTH_DWORD_FOURTH_BYTE								FOURTH_DWORD_THIRD_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FOURTH_DWORD_SECOND_BYTE								FOURTH_DWORD_FIRST_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.47.4 Fields

Field	Function
31-24 FOURTH_DWORD_FOURTH_BYTE	Byte 3 of Header log register of Fourth 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
23-16 FOURTH_DWORD_THIRD_BYTE	Byte 2 of Header log register of Fourth 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 FOURTH_DWORD_SECOND_BYTE	Byte 1 of Header log register of Fourth 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 FOURTH_DWORD_FIRST_BYTE	Byte 0 of Header log register of Fourth 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.4.3.1.48 Root Error Command Register. (ROOT_ERR_CMD_OFF)

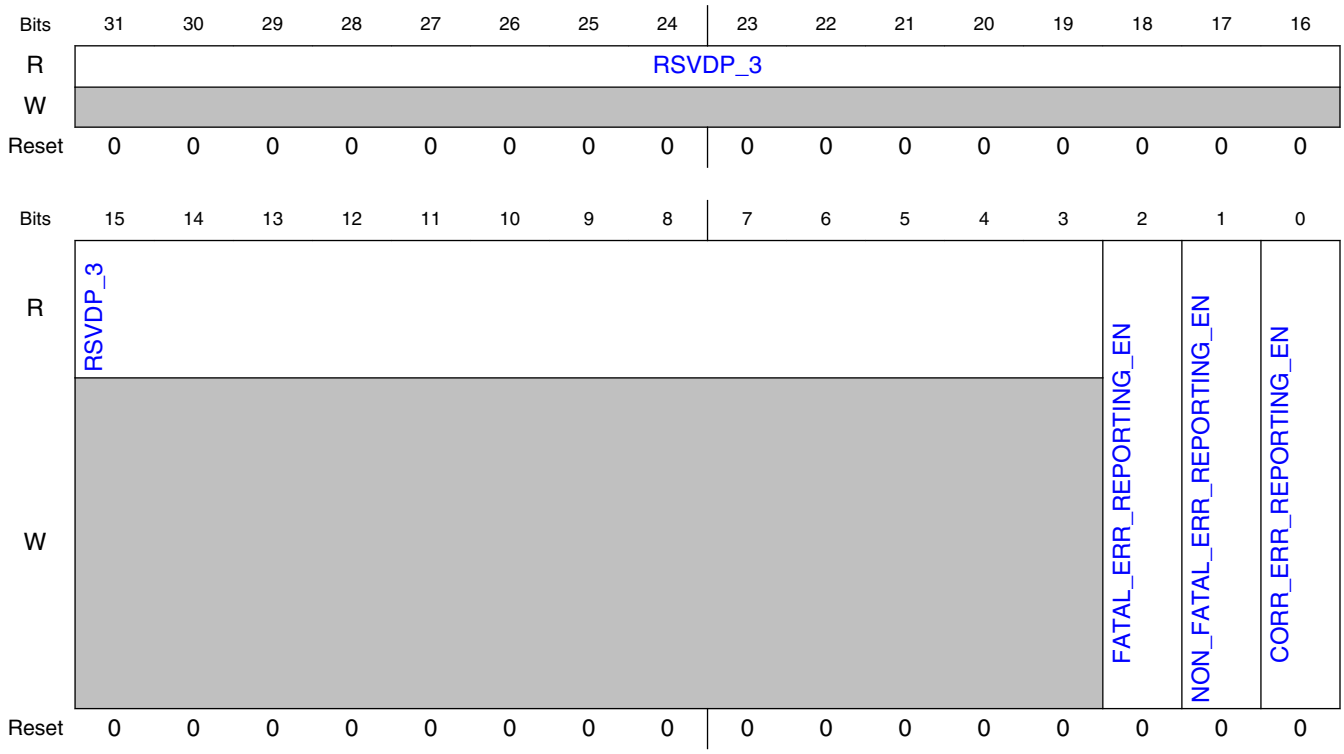
11.4.3.1.48.1 Offset

Register	Offset
ROOT_ERR_CMD_OFF	12Ch

11.4.3.1.48.2 Function

Root Error Command Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.48.3 Diagram



11.4.3.1.48.4 Fields

Field	Function
31-3 RSVDP_3	Reserved for future use.
2 FATAL_ERR_REPORTING_EN	Fatal Error Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
1 NON_FATAL_ERR_REPORTING_EN	Non-Fatal Error Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
0 CORR_ERR_REPORTING_EN	Correctable Error Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.49 Root Error Status Register. (ROOT_ERR_STATUS_OFF)

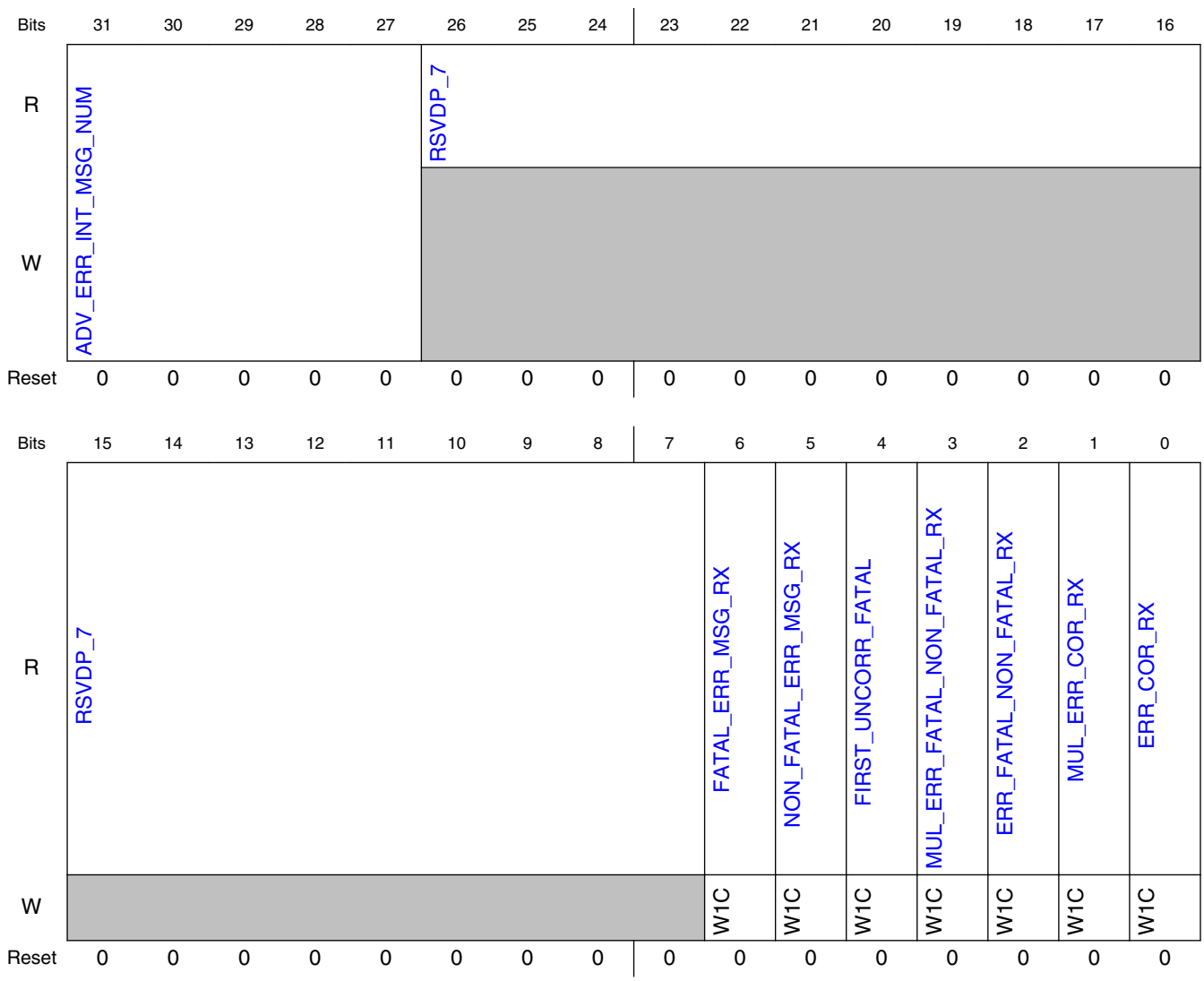
11.4.3.1.49.1 Offset

Register	Offset
ROOT_ERR_STATUS_OFF	130h

11.4.3.1.49.2 Function

Root Error Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.49.3 Diagram



11.4.3.1.49.4 Fields

Field	Function
31-27 ADV_ERR_INT_MSG_NUM	Advanced Error Interrupt Message Number. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
26-7 RSVDP_7	Reserved for future use.
6 FATAL_ERR_MSG_RX	One or more Fatal Error Messages Received. For a description of this standard PCIe register field, see the PCI Express Specification.
5 NON_FATAL_ERR_MSG_RX	One or more Non-Fatal Error Messages Received. For a description of this standard PCIe register field, see the PCI Express Specification.
4 FIRST_UNCOR R_FATAL	First Uncorrectable Error is Fatal. For a description of this standard PCIe register field, see the PCI Express Specification.
3 MUL_ERR_FATAL_NON_FATAL_RX	Multiple Fatal or Non-Fatal Errors Received. For a description of this standard PCIe register field, see the PCI Express Specification.
2 ERR_FATAL_NON_FATAL_RX	Fatal or Non-Fatal Error Received. For a description of this standard PCIe register field, see the PCI Express Specification.
1 MUL_ERR_COR_RX	Multiple Correctable Errors Received. For a description of this standard PCIe register field, see the PCI Express Specification.
0 ERR_COR_RX	Correctable Error Received. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.50 Error Source Identification Register. (ERR_SRC_ID_OFF)

11.4.3.1.50.1 Offset

Register	Offset
ERR_SRC_ID_OFF	134h

11.4.3.1.50.2 Function

Error Source Identification Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.50.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERR_FATAL_NON_FATAL_SOURCE_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_COR_SOURCE_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.50.4 Fields

Field	Function
31-16 ERR_FATAL_NON_FATAL_SOURCE_ID	Source of Fatal/Non-Fatal Error. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-0 ERR_COR_SOURCE_ID	Source of Correctable Error. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.4.3.1.51 TLP Prefix Log Register 1. (TLP_PREFIX_LOG_1_OFF)

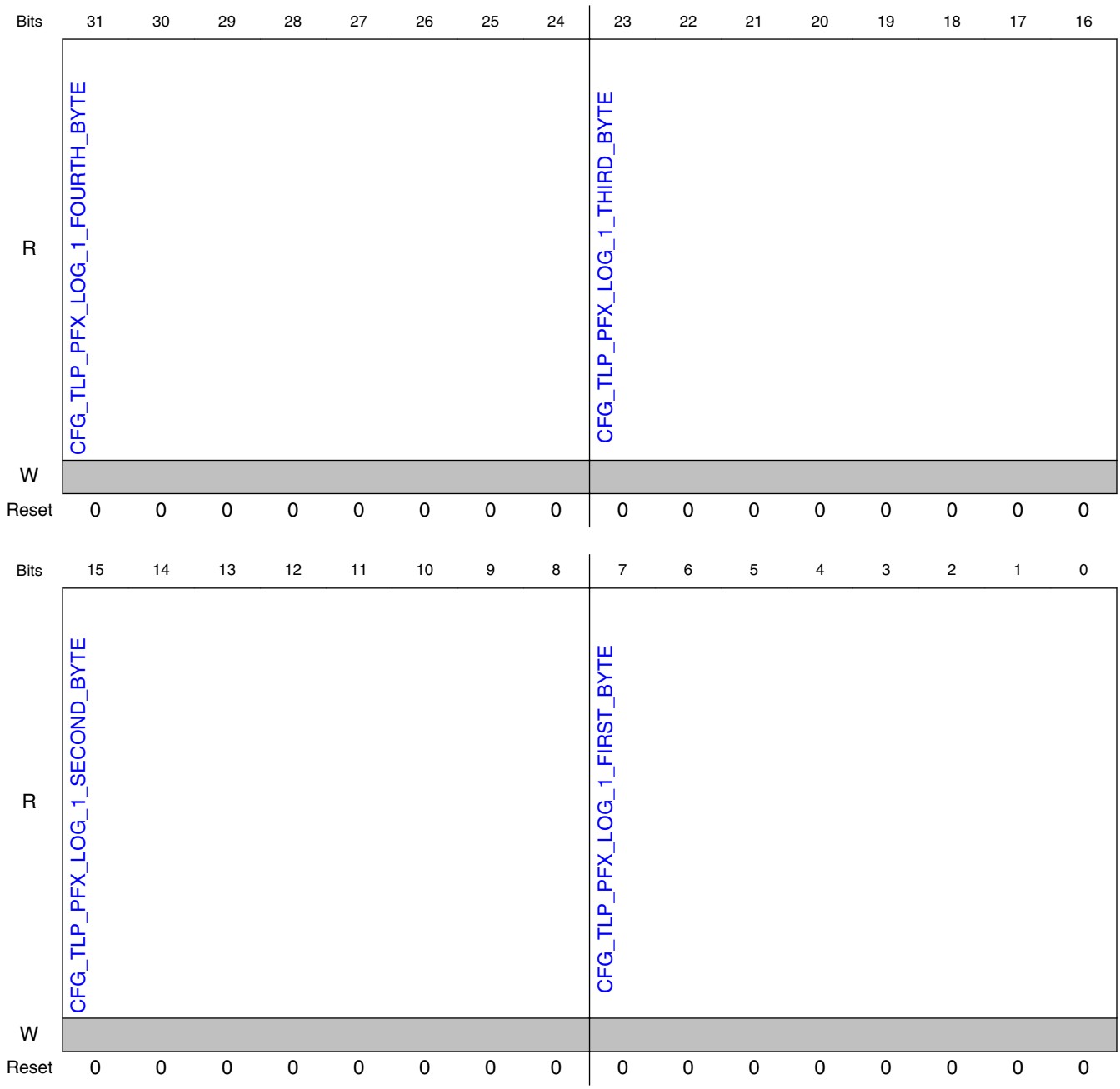
11.4.3.1.51.1 Offset

Register	Offset
TLP_PREFIX_LOG_1_OFF	138h

11.4.3.1.51.2 Function

TLP Prefix Log Register 1. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.51.3 Diagram



11.4.3.1.51.4 Fields

Field	Function
31-24 CFG_TLP_PFX_LOG_1_FOURTH_BYTE	Byte 3 of Error TLP Prefix Log 1. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
23-16 CFG_TLP_PFX_LOG_1_THIRD_BYTE	Byte 2 of Error TLP Prefix Log 1. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 CFG_TLP_PFX_LOG_1_SECOND_BYTE	Byte 1 of Error TLP Prefix Log 1. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 CFG_TLP_PFX_LOG_1_FIRST_BYTE	Byte 0 of Error TLP Prefix Log 1. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.4.3.1.52 TLP Prefix Log Register 2. (TLP_PREFIX_LOG_2_OFF)

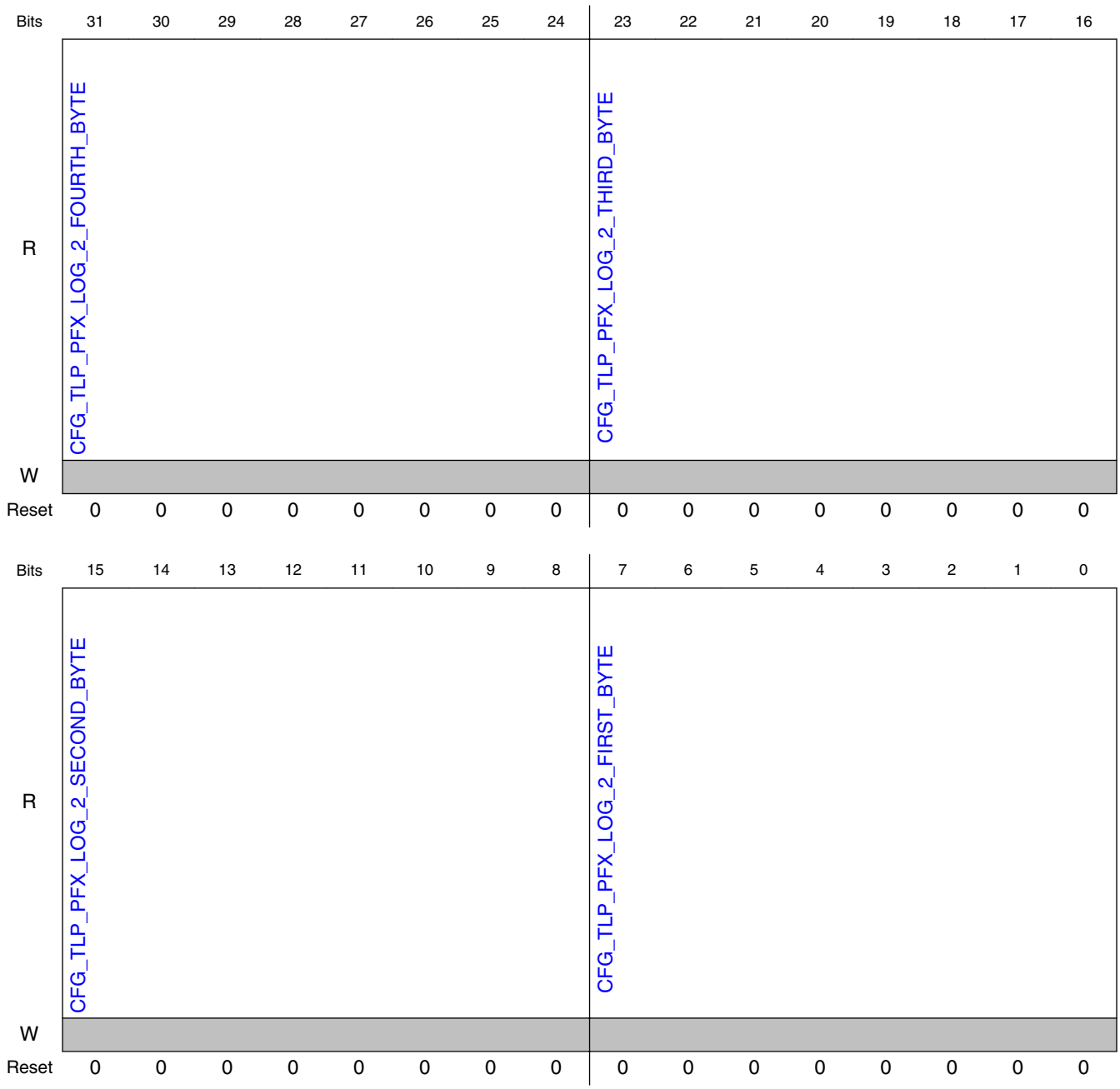
11.4.3.1.52.1 Offset

Register	Offset
TLP_PREFIX_LOG_2_OFF	13Ch

11.4.3.1.52.2 Function

TLP Prefix Log Register 2. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.52.3 Diagram



11.4.3.1.52.4 Fields

Field	Function
31-24 CFG_TLP_PFX_LOG_2_FOURTH_BYTE	Byte 3 Error TLP Prefix Log 2. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
23-16 CFG_TLP_PFX_LOG_2_THIRD_BYTE	Byte 2 Error TLP Prefix Log 2. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 CFG_TLP_PFX_LOG_2_SECOND_BYTE	Byte 1 Error TLP Prefix Log 2. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 CFG_TLP_PFX_LOG_2_FIRST_BYTE	Byte 0 Error TLP Prefix Log 2. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.4.3.1.53 TLP Prefix Log Register 3. (TLP_PREFIX_LOG_3_OFF)

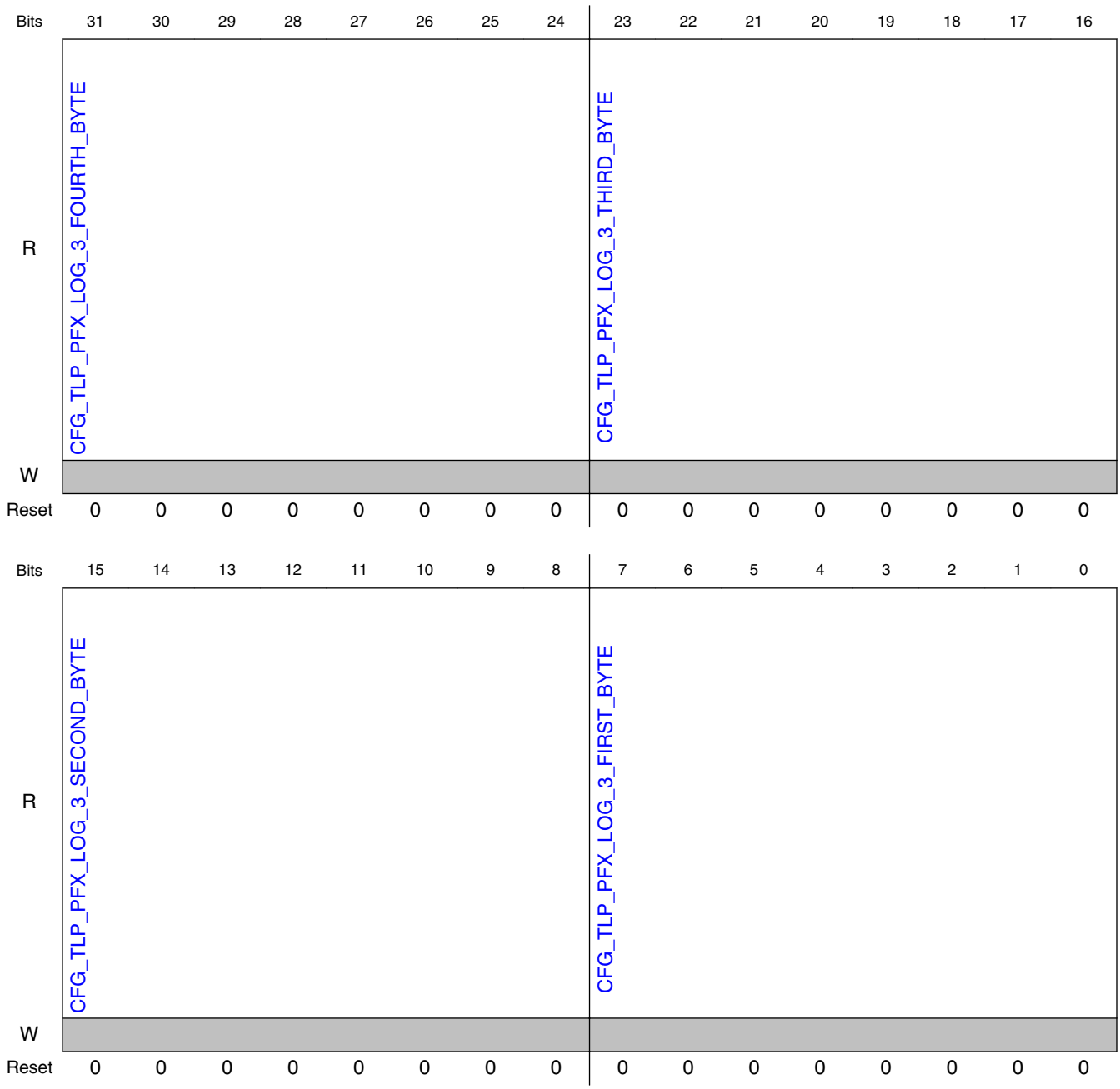
11.4.3.1.53.1 Offset

Register	Offset
TLP_PREFIX_LOG_3_OFF	140h

11.4.3.1.53.2 Function

TLP Prefix Log Register 3. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.53.3 Diagram



11.4.3.1.53.4 Fields

Field	Function
31-24 CFG_TLP_PFX_LOG_3_FOURTH_BYTE	Byte 3 Error TLP Prefix Log 3. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
23-16 CFG_TLP_PFX_LOG_3_THIRD_BYTE	Byte 2 Error TLP Prefix Log 3. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 CFG_TLP_PFX_LOG_3_SECOND_BYTE	Byte 1 Error TLP Prefix Log 3. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 CFG_TLP_PFX_LOG_3_FIRST_BYTE	Byte 0 Error TLP Prefix Log 3. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.4.3.1.54 TLP Prefix Log Register 4. (TLP_PREFIX_LOG_4_OFF)

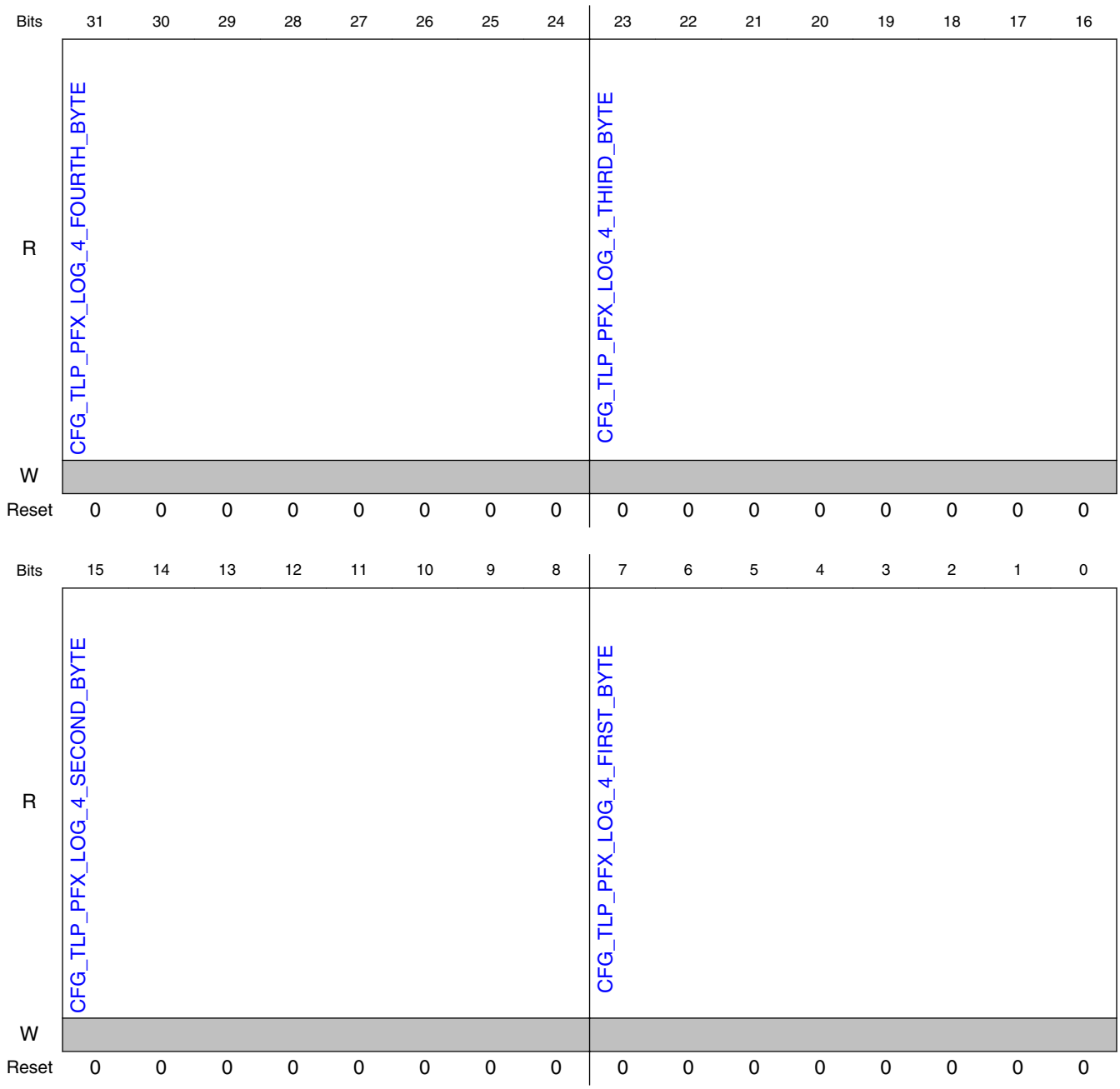
11.4.3.1.54.1 Offset

Register	Offset
TLP_PREFIX_LOG_4_OFF	144h

11.4.3.1.54.2 Function

TLP Prefix Log Register 4. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.54.3 Diagram



11.4.3.1.54.4 Fields

Field	Function
31-24 CFG_TLP_PFX_LOG_4_FOURTH_BYTE	Byte 3 Error TLP Prefix Log 4. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
23-16 CFG_TLP_PFX_LOG_4_THIRD_BYTE	Byte 2 Error TLP Prefix Log 4. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 CFG_TLP_PFX_LOG_4_SECOND_BYTE	Byte 1 Error TLP Prefix Log 4. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 CFG_TLP_PFX_LOG_4_FIRST_BYTE	Byte 0 Error TLP Prefix Log 4. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.4.3.1.55 L1 Substates Extended Capability Header. (L1SUB_CAP_HEADER_REG)

11.4.3.1.55.1 Offset

Register	Offset
L1SUB_CAP_HEADER_REG	158h

11.4.3.1.55.2 Function

L1 Substates Extended Capability Header. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.55.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NEXT_OFFSET												CAP_VERSION			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EXTENDED_CAP_ID															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0

11.4.3.1.55.4 Fields

Field	Function
31-20 NEXT_OFFSET	Next Capability Offset. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19-16 CAP_VERSION	Capability Version. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15-0 EXTENDED_CAP_ID	L1SUB Extended Capability ID. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

11.4.3.1.56 L1 Substates Capability Register. (L1SUB_CAPABILITY_REG)

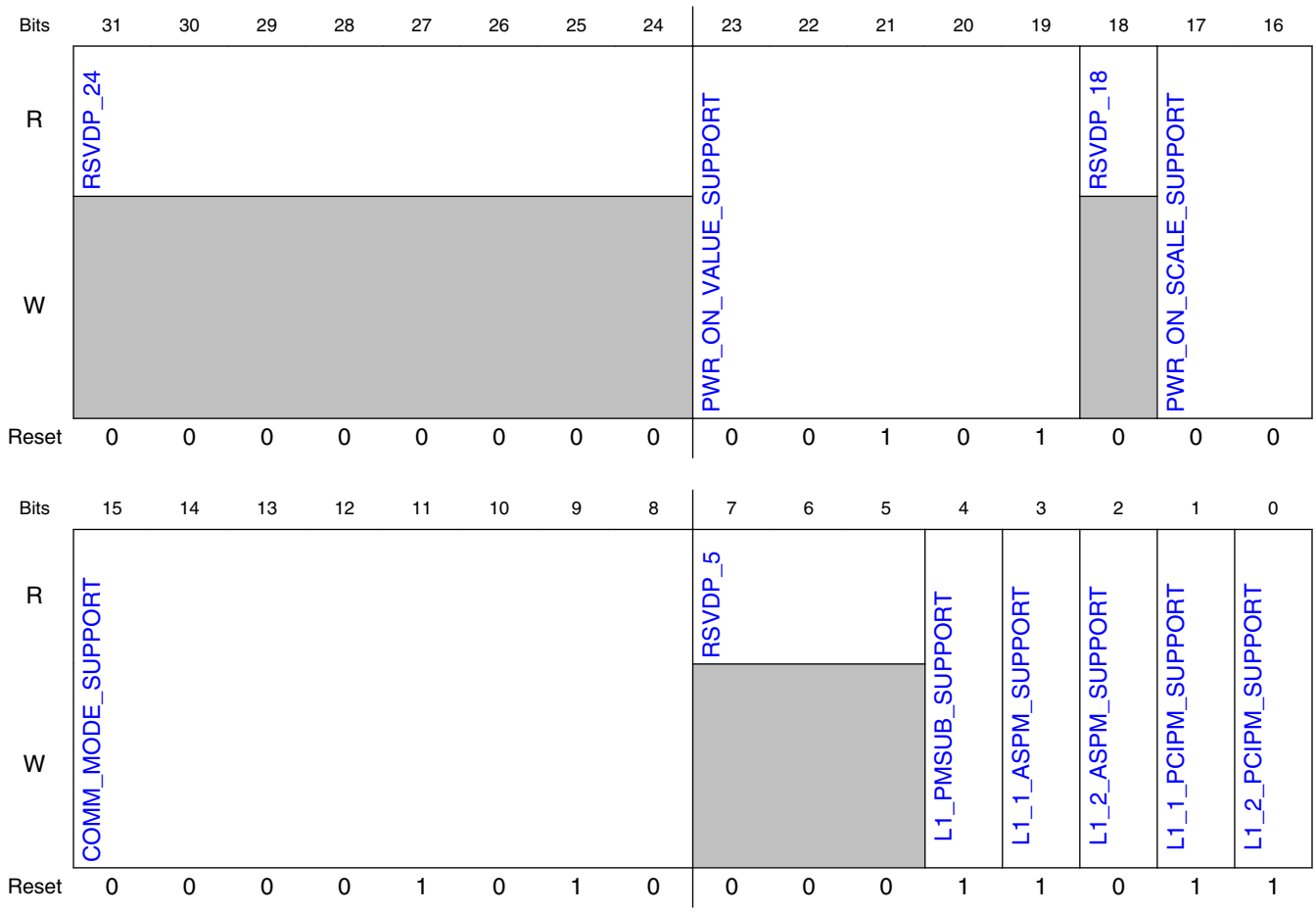
11.4.3.1.56.1 Offset

Register	Offset
L1SUB_CAPABILITY_REG	15Ch

11.4.3.1.56.2 Function

L1 Substates Capability Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.56.3 Diagram



11.4.3.1.56.4 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-19 PWR_ON_VALUE_SUPPORT	Port T Power On Value. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
18 RSVDP_18	Reserved for future use.
17-16 PWR_ON_SCALE_SUPPORT	Port T Power On Scale. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
15-8 COMM_MODE_SUPPORT	Port Common Mode Restore Time. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W

Table continues on the next page...

Field	Function
7-5 RSVDP_5	Reserved for future use.
4 L1_PMSUB_SUPPORT	L1 PM Substates ECN Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
3 L1_1_ASPM_SUPPORT	ASPM L11 Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
2 L1_2_ASPM_SUPPORT	ASPM L12 Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
1 L1_1_PCIPM_SUPPORT	PCI-PM L11 Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
0 L1_2_PCIPM_SUPPORT	PCI-PM L12 Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.57 L1 Substates Control 1 Register. (L1SUB_CONTROL1_REG)

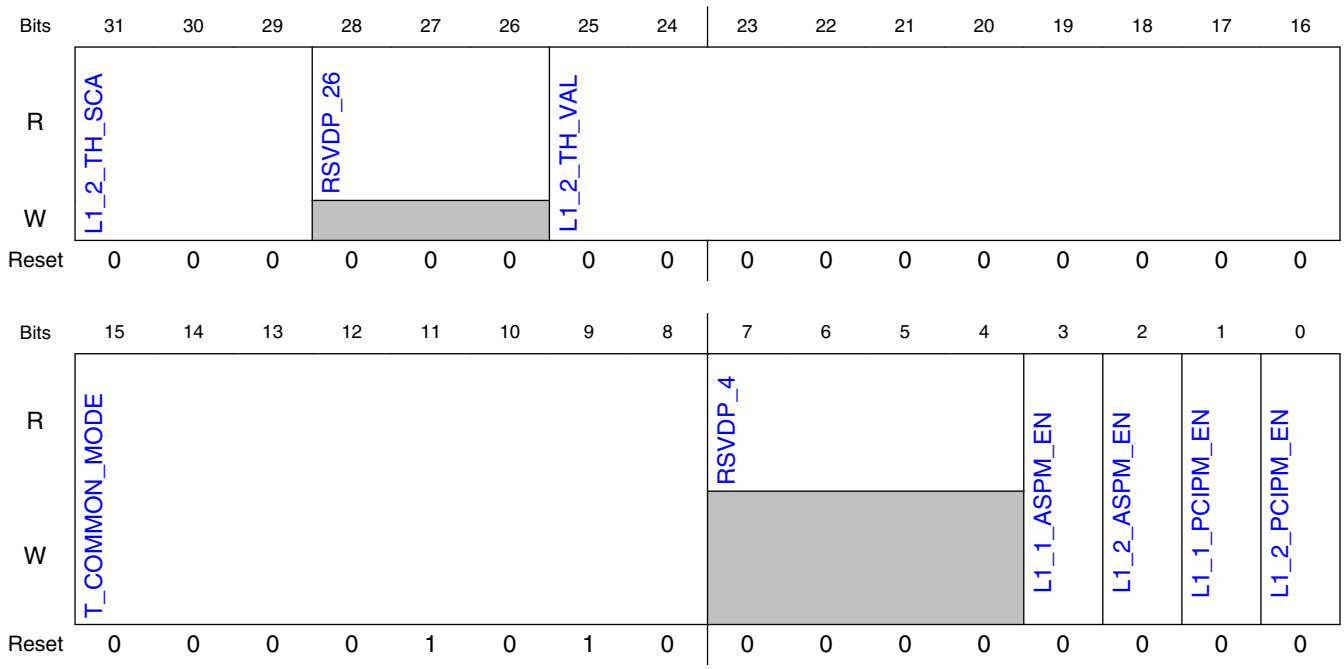
11.4.3.1.57.1 Offset

Register	Offset
L1SUB_CONTROL1_REG	160h

11.4.3.1.57.2 Function

L1 Substates Control 1 Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.57.3 Diagram



11.4.3.1.57.4 Fields

Field	Function
31-29 L1_2_TH_SCA	LTR L12 Threshold Scale. For a description of this standard PCIe register field, see the PCI Express Specification.
28-26 RSVDP_26	Reserved for future use.
25-16 L1_2_TH_VAL	LTR L12 Threshold Value. For a description of this standard PCIe register field, see the PCI Express Specification.
15-8 T_COMMON_MODE	Common Mode Restore Time. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
7-4 RSVDP_4	Reserved for future use.
3 L1_1_ASPM_EN	ASPM L11 Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
2 L1_2_ASPM_EN	ASPM L12 Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
1 L1_1_PCIPM_EN	PCI-PM L11 Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

Table continues on the next page...

Field	Function
0 L1_2_PCIPM_EN	PCI-PM L12 Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.58 L1 Substates Control 2 Register. (L1SUB_CONTROL2_REG)

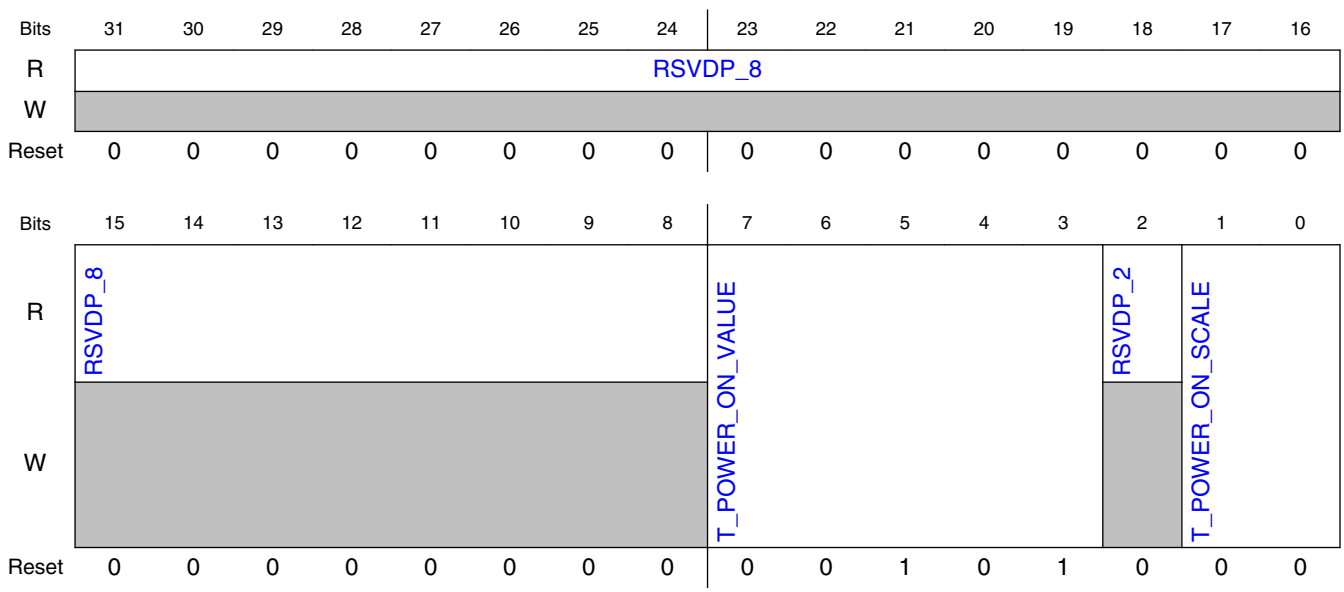
11.4.3.1.58.1 Offset

Register	Offset
L1SUB_CONTROL2_REG	164h

11.4.3.1.58.2 Function

L1 Substates Control 2 Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.4.3.1.58.3 Diagram



11.4.3.1.58.4 Fields

Field	Function
31-8	Reserved for future use.

Table continues on the next page...

Field	Function
RSVDP_8	
7-3 T_POWER_ON_VALUE	T Power On Value. For a description of this standard PCIe register field, see the PCI Express Specification.
2 RSVDP_2	Reserved for future use.
1-0 T_POWER_ON_SCALE	T Power On Scale. For a description of this standard PCIe register field, see the PCI Express Specification.

11.4.3.1.59 Ack Latency Timer and Replay Timer Register. (ACK_LATENCY_TIMER_OFF)

11.4.3.1.59.1 Offset

Register	Offset
ACK_LATENCY_TIMER_OFF	700h

11.4.3.1.59.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REPLAY_TIME_LIMIT															
Reset	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ROUND_TRIP_LATENCY_TIME_LIMIT															
Reset	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	1

11.4.3.1.59.3 Fields

Field	Function
31-16 REPLAY_TIME_LIMIT	Replay Timer Limit. The replay timer expires when it reaches this limit. The controller initiates a replay upon reception of a NAK or when the replay timer expires. For more details, see "Transmit Replay". You can modify the effective timer limit with the TIMER_MOD_REPLAY_TIMER field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register. After reset, the controller updates the default according to the Negotiated Link Width, Max_Payload_Size, and speed. The value is determined from Tables 3-4,

Table continues on the next page...

Field	Function
	3-5, and 3-6 of the PCIe 3.0 specification. If there is a change in the payload size or link speed, the controller will override any value that you have written to this register field, and reset the field back to the specification-defined value. It will not change the value in the TIMER_MOD_REPLAY_TIMER field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register.
15-0 ROUND_TRIP_LATENCY_TIME_LIMIT	Ack Latency Timer Limit. The Ack latency timer expires when it reaches this limit. For more details, see "Ack Scheduling". You can modify the effective timer limit with the TIMER_MOD_ACK_NAK field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register. After reset, the controller updates the default according to the Negotiated Link Width, Max_Payload_Size, and speed. The value is determined from Tables 3-7, 3-8, and 3-9 of the PCIe 3.0 specification. The limit must reflect the round trip latency from requester to completer. If there is a change in the payload size or link width, the controller will override any value that you have written to this register field, and reset the field back to the specification-defined value. It will not change the value in the TIMER_MOD_ACK_NAK field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register.

11.4.3.1.60 Vendor Specific DLLP Register. (VENDOR_SPEC_DLLP_OFF)

11.4.3.1.60.1 Offset

Register	Offset
VENDOR_SPEC_DLLP_OFF	704h

11.4.3.1.60.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VENDOR_SPEC_DLLP															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VENDOR_SPEC_DLLP															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.4.3.1.60.3 Fields

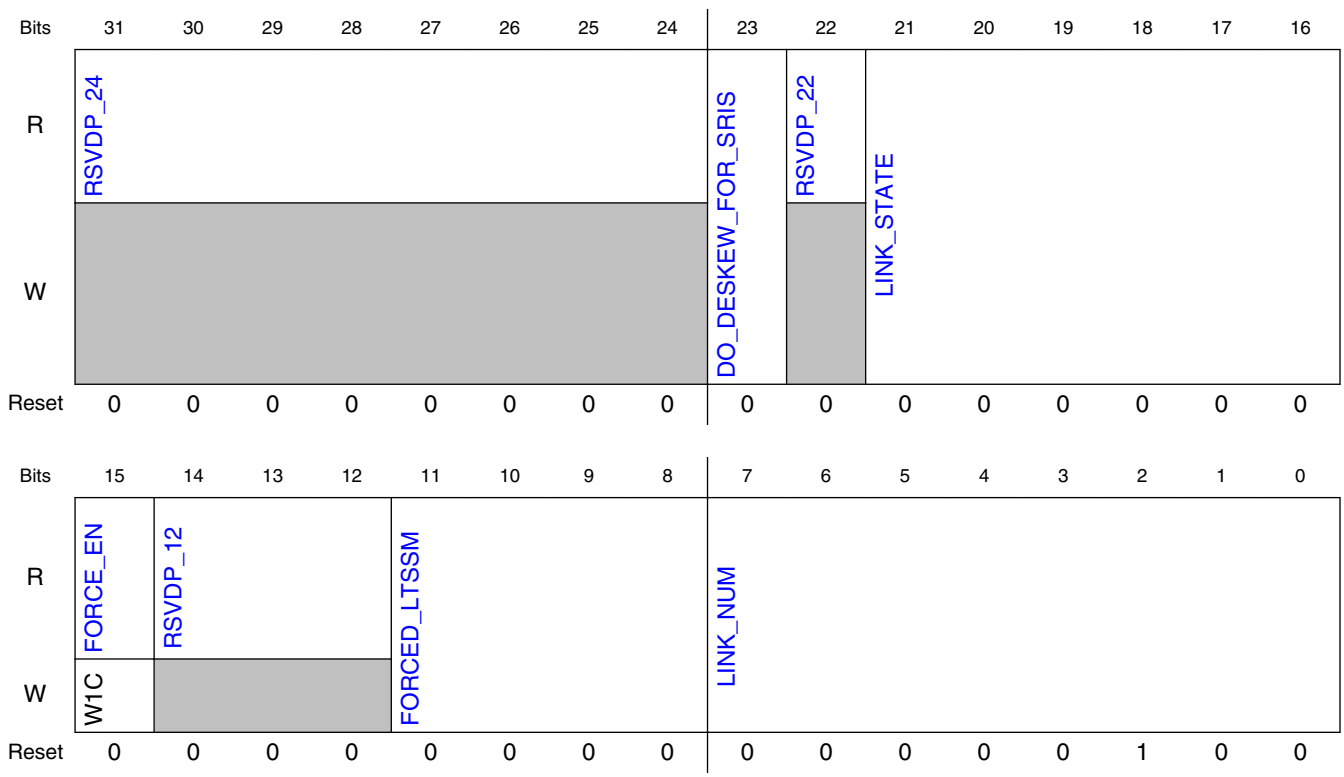
Field	Function
31-0 VENDOR_SPEC_DLLP	Vendor Specific DLLP Register. Used to send a specific PCI Express DLLP. Your application writes the 8-bit DLLP Type and 24-bits of Payload data into this register, then sets the field VENDOR_SPECIFIC_DLLP_REQ of PORT_LINK_CTRL_OFF to send the DLLP. - [7:0] = Type - [31:8] = Payload (24 bits) The dllp type is in bits [7:0] while the remainder is the vendor defined payload. Note: This register field is sticky.

11.4.3.1.61 Port Force Link Register. (PORT_FORCE_OFF)

11.4.3.1.61.1 Offset

Register	Offset
PORT_FORCE_OFF	708h

11.4.3.1.61.2 Diagram



11.4.3.1.61.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23 DO_DESKEW_FOR_SRIS	Use the transitions from TS2 to Logical Idle Symbol, SKP OS to Logical Idle Symbol, and FTS Sequence to SKP OS to do deskew for SRIS instead of using received SKP OS if DO_DESKEW_FOR_SRIS is set to 1. Note: This register field is sticky.
22	Reserved for future use.

Table continues on the next page...

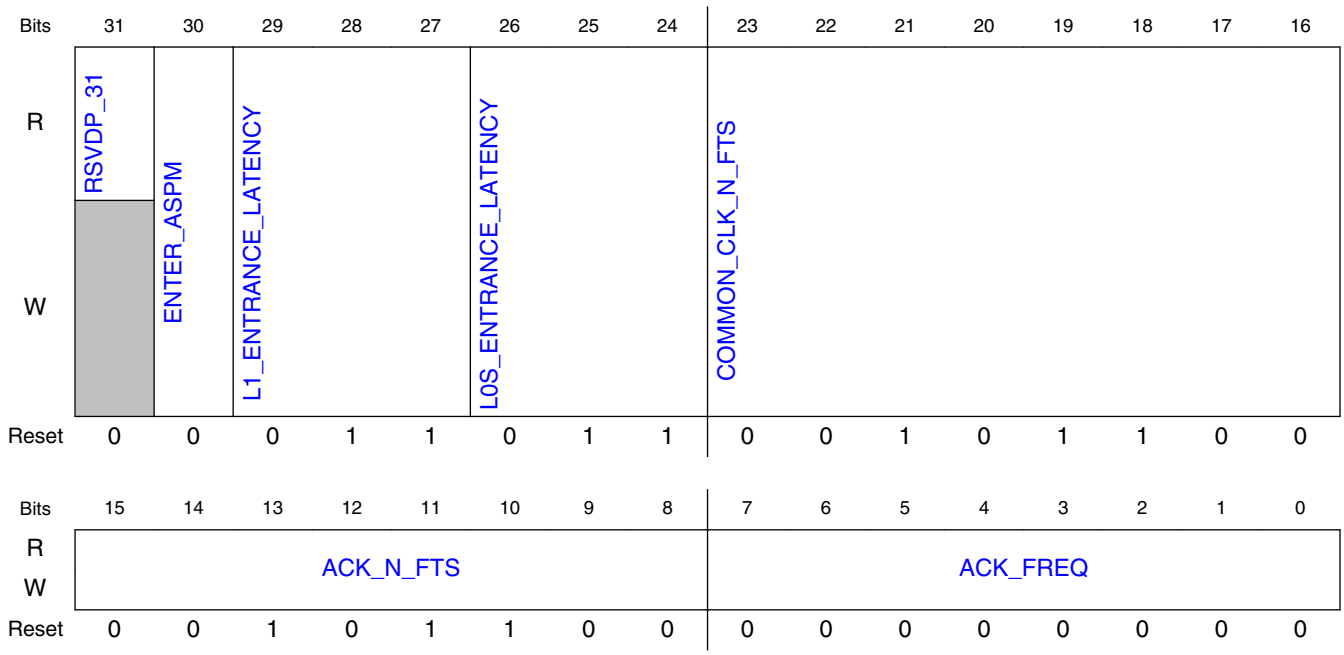
Field	Function
RSVDP_22	
21-16 LINK_STATE	Forced LTSSM State. The LTSSM state that the controller is forced to when you set the FORCE_EN bit (Force Link). LTSSM state encoding is defined by the lts_state variable in workspace/src/Layer1/smlh_ltssm.v. Note: This register field is sticky.
15 FORCE_EN	Force Link. The controller supports a testing and debug capability to allow your software to force the LTSSM state machine into a specific state, and to force the controller to transmit a specific Link Command. Asserting this bit triggers the following actions: - Forces the LTSSM to the state specified by the Forced LTSSM State field. - Forces the controller to transmit the command specified by the Forced Link Command field. This is a self-clearing register field. Reading from this register field always returns a "0".
14-12 RSVDP_12	Reserved for future use.
11-8 FORCED_LTSSM	Forced Link Command. The link command that the controller is forced to transmit when you set FORCE_EN bit (Force Link). Link command encoding is defined by the ltssm_cmd variable in workspace/src/Layer1/smlh_ltssm.v. Note: This register field is sticky.
7-0 LINK_NUM	Link Number. Not used for endpoint. Not used for M-PCIe. Note: This register field is sticky.

11.4.3.1.62 Ack Frequency and L0-L1 ASPM Control Register. (ACK_F_ASPM_CTRL_OFF)

11.4.3.1.62.1 Offset

Register	Offset
ACK_F_ASPM_CTRL_OFF	70Ch

11.4.3.1.62.2 Diagram



11.4.3.1.62.3 Fields

Field	Function
31 RSVDP_31	Reserved for future use.
30 ENTER_ASPM	ASPM L1 Entry Control. - 1: Core enters ASPM L1 after a period in which it has been idle. - 0: Core enters ASPM L1 only after idle period during which both receive and transmit are in L0s. Note: This register field is sticky.
29-27 L1_ENTRANCE_LATENCY	L1 Entrance Latency. Value range is: - 000: 1 us - 001: 2 us - 010: 4 us - 011: 8 us - 100: 16 us - 101: 32 us - 110 or 111: 64 us Note: Programming this timer with a value greater that 32us has no effect unless extended sync is used, or all of the credits are infinite. Note: This register field is sticky.
26-24 L0S_ENTRANCE_LATENCY	L0s Entrance Latency. Values correspond to: - 000: 1 us - 001: 2 us - 010: 3 us - 011: 4 us - 100: 5 us - 101: 6 us - 110 or 111: 7 us This field is applicable to STALL while in L0 for M-PCIe. Note: This register field is sticky.
23-16 COMMON_CLK_N_FTS	Common Clock N_FTS. This is the N_FTS when common clock is used. The number of Fast Training Sequence ordered sets to be transmitted when transitioning from L0s to L0. The maximum number of FTS ordered-sets that a component can request is 255. This field is only writable (sticky) when all of the following configuration parameter equations are true: - CX_NFTS !=CX_COMM_NFTS - DEFAULT_L0S_EXIT_LATENCY !=DEFAULT_COMM_L0S_EXIT_LATENCY - DEFAULT_L1_EXIT_LATENCY !=DEFAULT_COMM_L1_EXIT_LATENCY The controller does not support a value of zero; a value of zero can cause the LTSSM to go into the recovery state when exiting from L0s. This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R
15-8 ACK_N_FTS	N_FTS. The number of Fast Training Sequence ordered sets to be transmitted when transitioning from L0s to L0. The maximum number of FTS ordered-sets that a component can request is 255. The controller does not support a value of zero; a value of zero can cause the LTSSM to go into the recovery

Table continues on the next page...

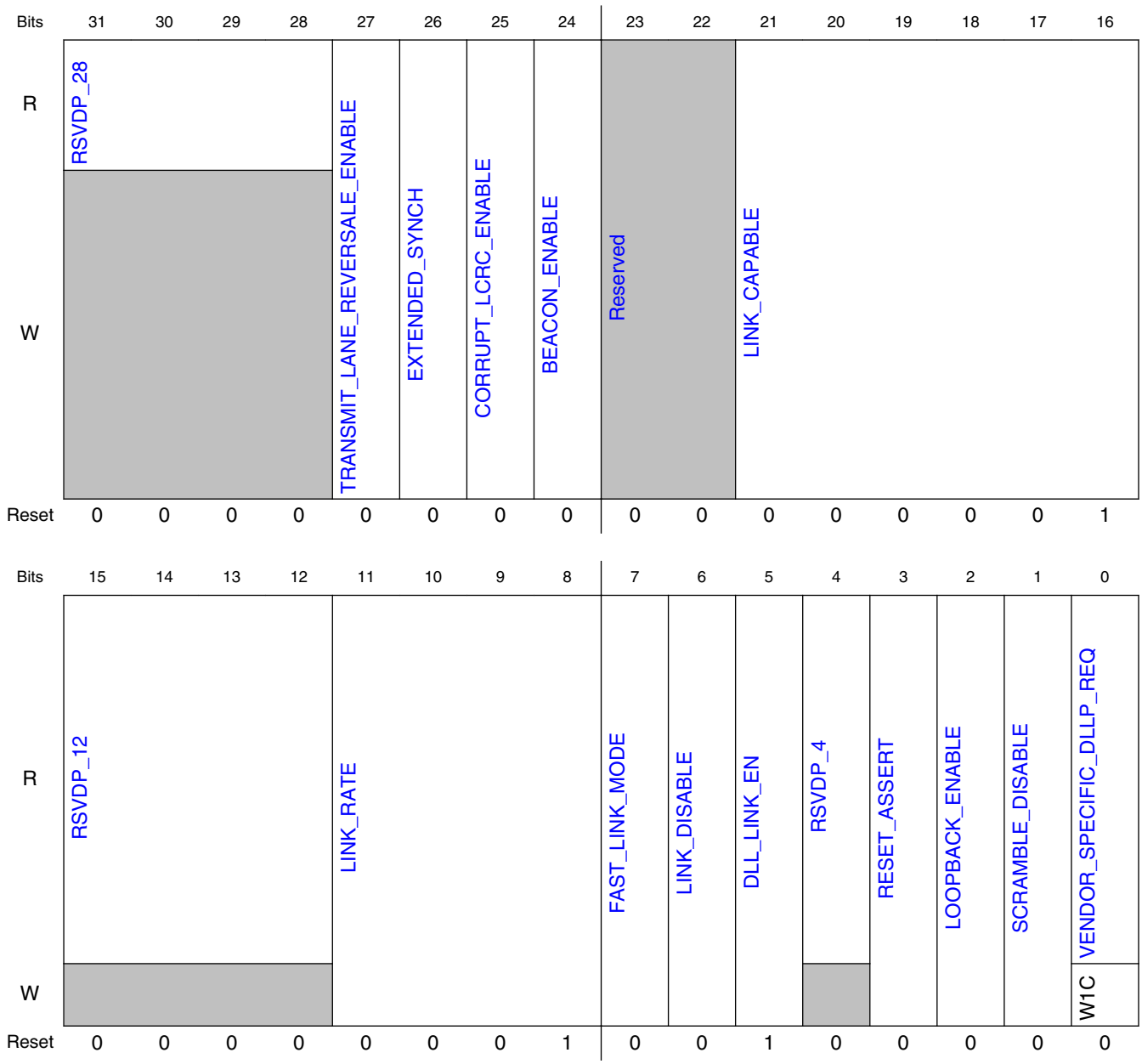
Field	Function
	state when exiting from L0s. This field is reserved (fixed to '0') for M-PCIe. Note: This register field is sticky.
7-0 ACK_FREQ	Ack Frequency. The controller accumulates the number of pending ACKs specified here (up to 255) before sending an ACK DLLP. - 0: Indicates that this Ack frequency control feature is turned off. The controller schedules a low-priority ACK DLLP for every TLP that it receives. - 1-255: Indicates that the controller will schedule a high-priority ACK after receiving this number of TLPs. It might schedule the ACK before receiving this number of TLPs, but never later. For a typical system, you do not have to modify the default setting. For more details, see "ACK/NAK Scheduling". Note: This register field is sticky.

11.4.3.1.63 Port Link Control Register. (PORT_LINK_CTRL_OFF)

11.4.3.1.63.1 Offset

Register	Offset
PORT_LINK_CTRL_OFF	710h

11.4.3.1.63.2 Diagram



11.4.3.1.63.3 Fields

Field	Function
31-28 RSVDP_28	Reserved for future use.
27	TRANSMIT_LANE_REVERSALE_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.

Table continues on the next page...

Field	Function
TRANSMIT_LANE_REVERSALE_ENABLE	
26 EXTENDED_SYNCH	EXTENDED_SYNCH is an internally reserved field. Do not use. Note: This register field is sticky.
25 CORRUPT_LCRC_ENABLE	CORRUPT_LCRC_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
24 BEACON_ENABLE	BEACON_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
23-22 —	Reserved
21-16 LINK_CAPABLE	Link Mode Enable. Sets the number of lanes in the link that you want to connect to the link partner. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Predetermined Number of Lanes" field of the "Link Width and Speed Change Control Register". For more information, see "How to Tie Off Unused Lanes". For information on upsizing and downsizing the link width, see "Link Establishment". - 000001: x1 - 000011: x2 - 000111: x4 - 001111: x8 - 011111: x16 - 111111: x32 (not supported) This field is reserved (fixed to '0') for M-PCIe. Note: This register field is sticky.
15-12 RSVDP_12	Reserved for future use.
11-8 LINK_RATE	LINK_RATE is an internally reserved field. Do not use. Note: This register field is sticky.
7 FAST_LINK_MODE	Fast Link Mode. Sets all internal LTSSM millisecond timers to Fast Mode for speeding up simulation. Forces the LTSSM training (link initialization) to use shorter time-outs and to link up faster. The default scaling factor can be changed using the DEFAULT_FAST_LINK_SCALING_FACTOR parameter or through the FAST_LINK_SCALING_FACTOR field in the TIMER_CTRL_MAX_FUNC_NUM_OFF register. Fast Link Mode can also be activated by setting the diag_ctrl_bus[2] pin to '1'. For more details, see the "Fast Link Simulation Mode" section in the "Integrating the Core with the PHY or Application RTL or Verification IP" chapter of the User Guide. For M-PCIe, this field also affects Remain Hibern8 Time, Minimum Activate Time, and RRAP timeout. If this bit is set to '1', tRRAPInitiatorResponse is set to 1.88 ms(60 ms/32). Note: This register field is sticky.
6 LINK_DISABLE	LINK_DISABLE is an internally reserved field. Do not use. Note: This register field is sticky.
5 DLL_LINK_EN	DLL Link Enable. Enables link initialization. When DLL Link Enable =0, the controller does not transmit InitFC DLLPs and does not establish a link. Note: This register field is sticky.
4 RSVDP_4	Reserved for future use.
3 RESET_ASSERT	Reset Assert. Triggers a recovery and forces the LTSSM to the hot reset state (downstream port only). Note: This register field is sticky.
2 LOOPBACK_ENABLE	Loopback Enable. Turns on loopback. For more details, see "Loopback". For M-PCIe, to force the master to enter Digital Loopback mode, you must set this field to "1" during Configuration.start state(initial discovery/configuration). M-PCIe doesn't support loopback mode from L0 state - only from Configuration.start. Note: This register field is sticky.

Table continues on the next page...

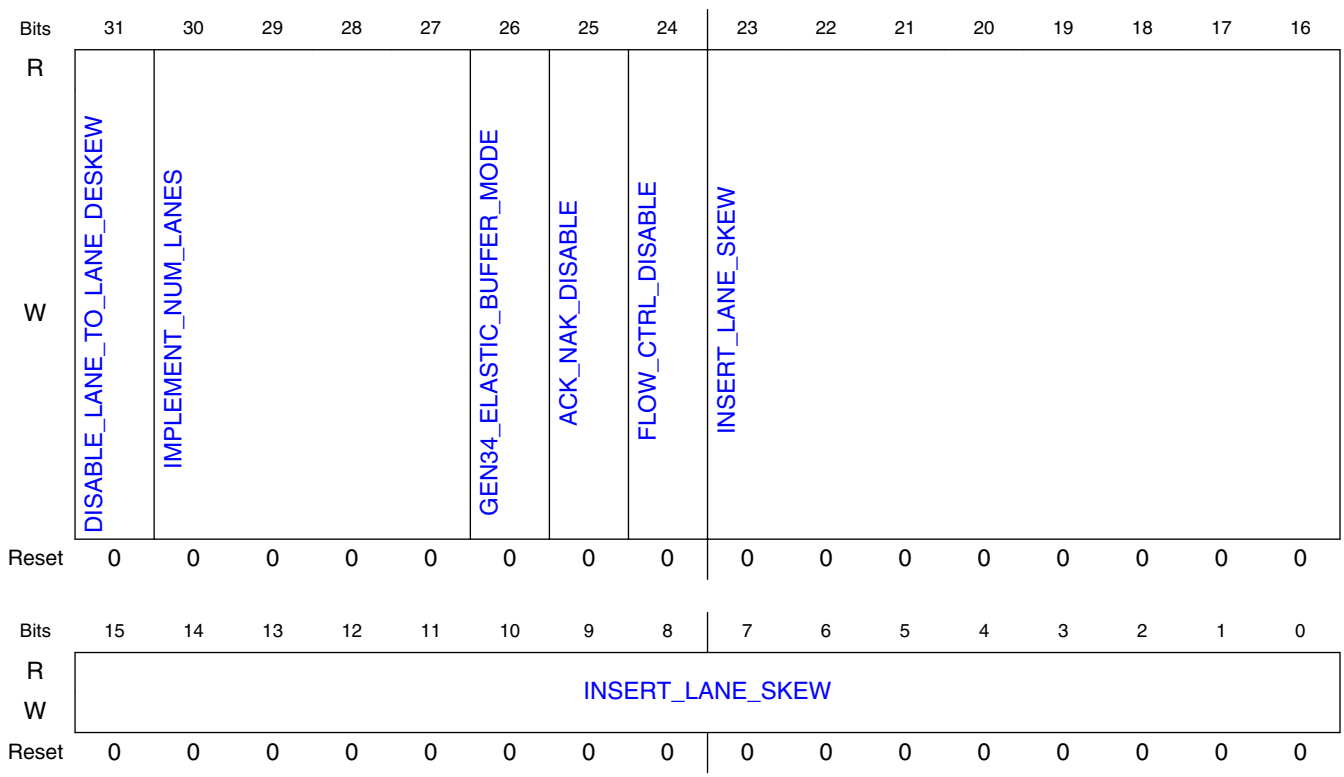
Field	Function
1 SCRAMBLE_DISABLE	Scramble Disable. Turns off data scrambling. Note: This register field is sticky.
0 VENDOR_SPECIFIC_DLLP_REQ	Vendor Specific DLLP Request. When software writes a '1' to this bit, the controller transmits the DLLP contained in the VENDOR_SPEC_DLLP field of VENDOR_SPEC_DLLP_OFF. Reading from this self-clearing register field always returns a '0'.

11.4.3.1.64 Lane Skew Register. (LANE_SKEW_OFF)

11.4.3.1.64.1 Offset

Register	Offset
LANE_SKEW_OFF	714h

11.4.3.1.64.2 Diagram



11.4.3.1.64.3 Fields

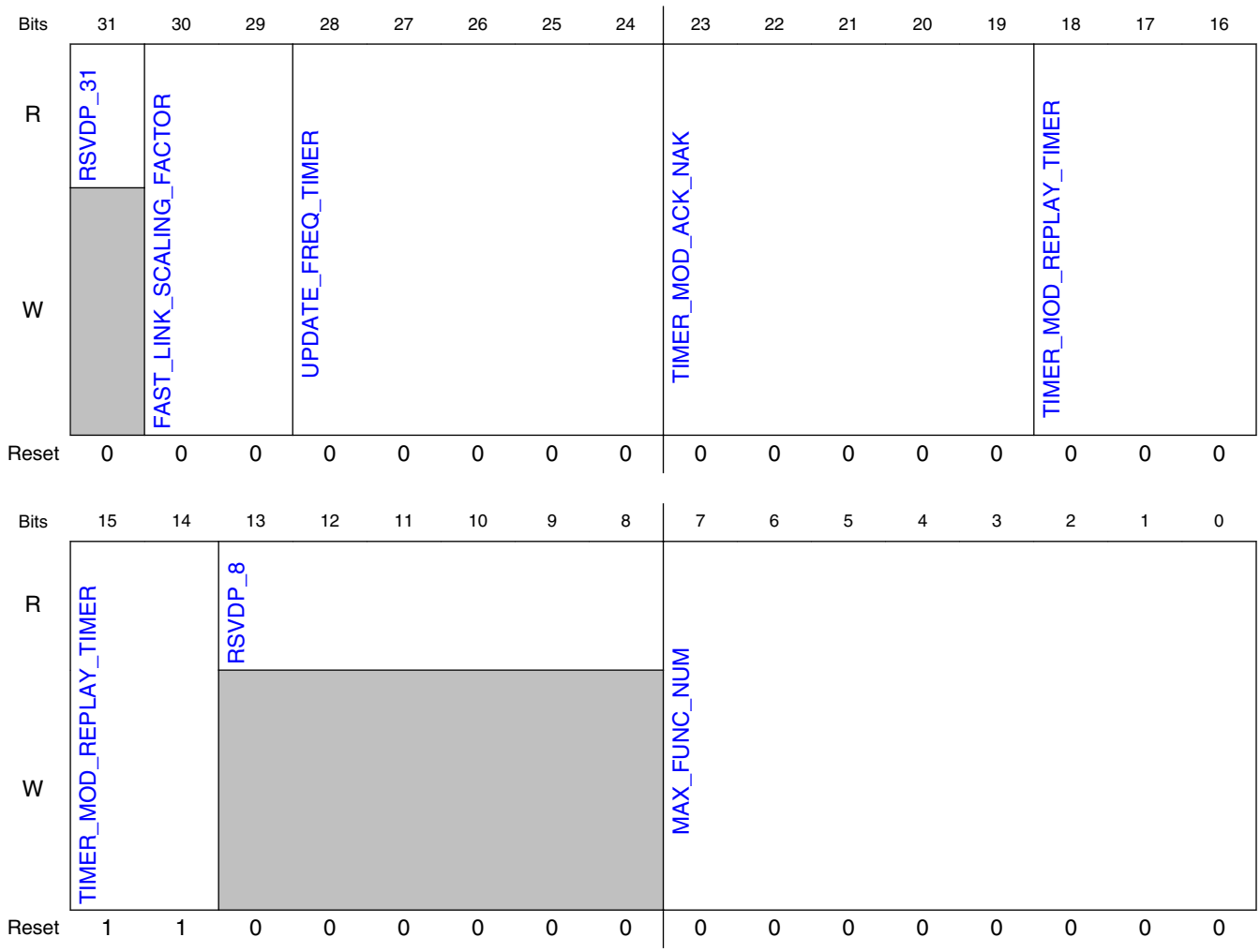
Field	Function
31 DISABLE_LANE_TO_LANE_DESKEW	Disable Lane-to-Lane Deskew. Causes the controller to disable the internal Lane-to-Lane deskew logic. Note: This register field is sticky.
30-27 IMPLEMENT_NUM_LANES	Implementation-specific Number of Lanes. Set the implementation-specific number of lanes. Allowed values are: - 4'b0000: 1 lane - 4'b0001: 2 lanes - 4'b0011: 4 lanes - 4'b0111: 8 lanes - 4'b1111: 16 lanes The number of lanes to be used when in Loopback Master. The number of lanes programmed must be equal to or less than the valid number of lanes set in LINK_CAPABLE field. You must configure this field before initiating Loopback by writing in the LOOPBACK_ENABLE field. The controller will transition from Loopback.Entry to Loopback.Active after receiving two consecutive TS1 Ordered Sets with the Loopback bit asserted on the implementation specific number of lanes configured in this field. Note: This register field is sticky.
26 GEN34_ELASTIC_BUFFER_MODE	Selects Elasticity Buffer operating mode in Gen3 or Gen4 rate: 0: Nominal Half Full Buffer mode 1: Nominal Empty Buffer Mode This register bit only affects Gen3 or Gen4 operating rate. For Gen1 or Gen2 operating rate the Elasticity Buffer operating mode is always the Nominal Half Full Buffer mode. Note: This register field is sticky.
25 ACK_NAK_DISABLE	Ack/Nak Disable. Prevents the controller from sending ACK and NAK DLLPs. Note: This register field is sticky.
24 FLOW_CTRL_DISABLE	Flow Control Disable. Prevents the controller from sending FC DLLPs. Note: This register field is sticky.
23-0 INSERT_LANE_SKEW	INSERT_LANE_SKEW is an internally reserved field. Do not use. Note: This register field is sticky.

11.4.3.1.65 Timer Control and Max Function Number Register. (TIMER_CTRL_MAX_FUNC_NUM_OFF)

11.4.3.1.65.1 Offset

Register	Offset
TIMER_CTRL_MAX_FUNC_NUM_OFF	718h

11.4.3.1.65.2 Diagram



11.4.3.1.65.3 Fields

Field	Function
31 RSVDP_31	Reserved for future use.
30-29 FAST_LINK_SCALING_FACTOR	Fast Link Timer Scaling Factor. Sets the scaling factor of LTSSM timer when FAST_LINK_MODE field in PORT_LINK_CTRL_OFF is set to '1'. - 0: Scaling Factor is 1024 (1ms is 1us) - 1: Scaling Factor is 256 (1ms is 4us) - 2: Scaling Factor is 64 (1ms is 16us) - 3: Scaling Factor is 16 (1ms is 64us) Default is set by the hidden configuration parameter DEFAULT_FAST_LINK_SCALING_FACTOR which defaults to '0'. Not used for M-PCIe. Note: This register field is sticky.
28-24 UPDATE_FREQ_TIMER	UPDATE_FREQ_TIMER is an internally reserved field. Do not use. Note: This register field is sticky.
23-19	Ack Latency Timer Modifier. Increases the timer value for the Ack latency timer in increments of 64 clock cycles. A value of "0" represents no modification to the timer value. For more details, see the

Table continues on the next page...

Field	Function
TIMER_MOD_ACK_NAK	ROUND_TRIP_LATENCY_TIME_LIMIT field of the ACK_LATENCY_TIMER_OFF register. Note: This register field is sticky.
18-14 TIMER_MOD_REPLAY_TIMER	Replay Timer Limit Modifier. Increases the time-out value for the replay timer in increments of 64 clock cycles at Gen1 or Gen2 speed, and in increments of 256 clock cycles at Gen3 speed. A value of "0" represents no modification to the timer limit. For more details, see the REPLAY_TIME_LIMIT field of the ACK_LATENCY_TIMER_OFF register. At Gen3 speed, the controller automatically changes the value of this field to DEFAULT_GEN3_REPLAY_ADJ. For M-PCIe, this field increases the time-out value for the replay timer in increments of 64 clock cycles at HS-Gear1, HS-Gear2, or HS-Gear3 speed. Note: This register field is sticky.
13-8 RSVDP_8	Reserved for future use.
7-0 MAX_FUNC_NUM	Maximum function number that can be used in a request. Configuration requests targeted at function numbers above this value are returned with UR (unsupported request). Note: This register field is sticky.

11.4.3.1.66 Symbol Timer Register and Filter Mask 1 Register. (SYMBOL_TIMER_FILTER_1_OFF)

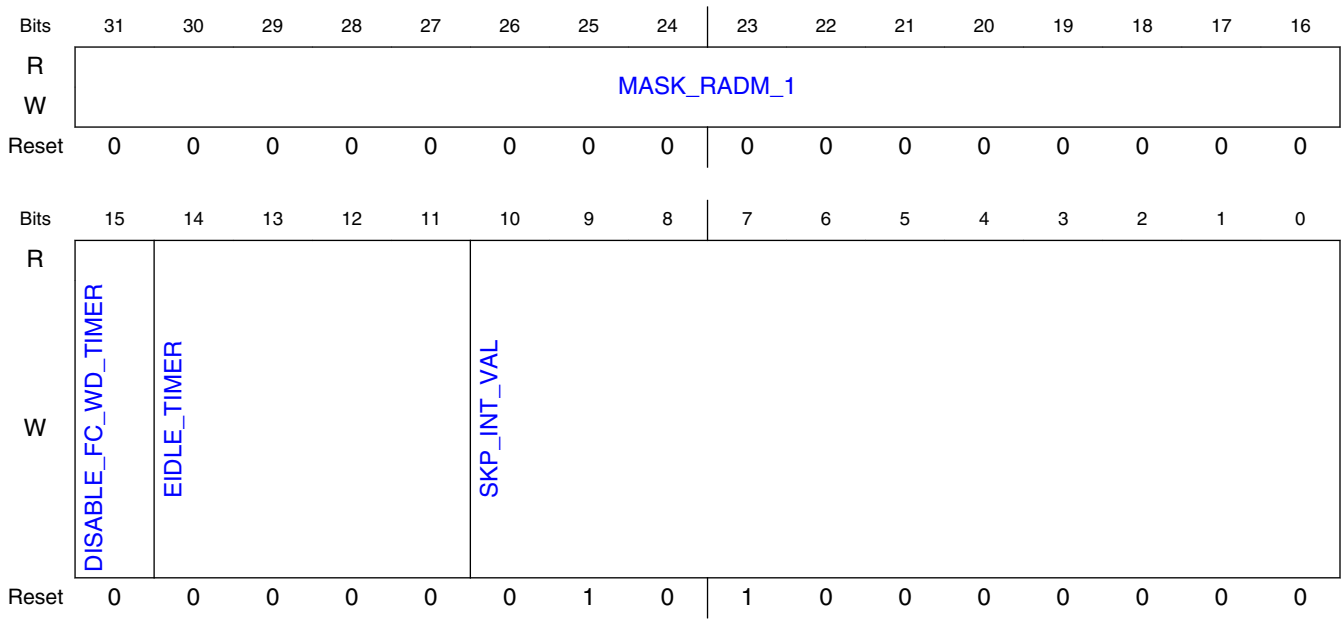
11.4.3.1.66.1 Offset

Register	Offset
SYMBOL_TIMER_FILTER_1_OFF	71Ch

11.4.3.1.66.2 Function

Symbol Timer Register and Filter Mask 1 Register. The Filter Mask 1 Register modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule.

11.4.3.1.66.3 Diagram



11.4.3.1.66.4 Fields

Field	Function
31-16 MASK_RADM_1	Filter Mask 1. The Filter Mask 1 Register modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule. [31]: CX_FLT_MASK_RC_CFG_DISCARD - 0: For RADM RC filter to not allow CFG transaction being received - 1: For RADM RC filter to allow CFG transaction being received [30]: CX_FLT_MASK_RC_IO_DISCARD - 0: For RADM RC filter to not allow IO transaction being received - 1: For RADM RC filter to allow IO transaction being received [29]: CX_FLT_MASK_MSG_DROP - 0: Drop MSG TLP (except for Vendor MSG). Send decoded message on the SII. - 1: Do not Drop MSG (except for Vendor MSG). Send message TLPs to your application on TRGT1 and send decoded message on the SII. - The default for this bit is the inverse of FLT_DROP_MSG. That is, if FLT_DROP_MSG =1, then the default of this bit is "0" (drop message TLPs). This bit only controls message TLPs other than Vendor MSGs. Vendor MSGs are controlled by Filter Mask Register 2, bits [1:0]. The controller never passes ATS Invalidate messages to the SII interface regardless of this filter rule setting. The controller passes all ATS Invalidate messages to TRGT1 (or AXI bridge master), as they are too big for the SII. [28]: CX_FLT_MASK_CPL_ECRC_DISCARD - Only used when completion queue is advertised with infinite credits and is in store-and-forward mode. - 0: Discard completions with ECRC errors - 1: Allow completions with ECRC errors to be passed up - Reserved field for SW. [27]: CX_FLT_MASK_ECRC_DISCARD - 0: Discard TLPs with ECRC errors - 1: Allow TLPs with ECRC errors to be passed up [26]: CX_FLT_MASK_CPL_LEN_MATCH - 0: Enforce length match for completions; a violation results in cpl_abort, and possibly AER of unexp_cpl_err - 1: MASK length match for completions [25]: CX_FLT_MASK_CPL_ATTR_MATCH - 0: Enforce attribute match for completions; a violation results in a malformed TLP error, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask attribute match for completions [24]: CX_FLT_MASK_CPL_TC_MATCH - 0: Enforce Traffic Class match for completions; a violation results in a malformed TLP error, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask Traffic Class match for completions [23]: CX_FLT_MASK_CPL_FUNC_MATCH - 0: Enforce function match for completions; a violation results in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask function match for completions [22]: CX_FLT_MASK_CPL_REQID_MATCH - 0: Enforce Req. Id match for completions; a violation result in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask Req.

Table continues on the next page...

Field	Function
	Id match for completions [21]: CX_FLT_MASK_CPL_TAGERR_MATCH - 0: Enforce Tag Error Rules for completions; a violation result in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca - 1: Mask Tag Error Rules for completions [20]: CX_FLT_MASK_LOCKED_RD_AS_UR - 0: Treat locked Read TLPs as UR for EP; Supported for RC - 1: Treat locked Read TLPs as Supported for EP; UR for RC [19]: CX_FLT_MASK_CFG_TYPE1_REQ_AS_UR - 0: Treat CFG type1 TLPs as UR for EP; Supported for RC - 1: Treat CFG type1 TLPs as Supported for EP; UR for RC - When CX_SRIOV_ENABLE is set then this bit is set to allow the filter to process Type 1 Config requests if the EP consumes more than one bus number. [18]: CX_FLT_MASK_UR_OUTSIDE_BAR - 0: Treat out-of-bar TLPs as UR - 1: Do not treat out-of-bar TLPs as UR [17]: CX_FLT_MASK_UR_POIS - 0: Treat poisoned request TLPs as UR - 1: Do not treat poisoned request TLPs as UR - The native controller always passes poisoned completions to your application except when you are using the DMA read channel. [16]: CX_FLT_MASK_UR_FUNC_MISMATCH - 0: Treat Function MisMatched TLPs as UR - 1: Do not treat Function MisMatched TLPs as UR Note: This register field is sticky.
15 DISABLE_FC_WD_TIMER	Disable FC Watchdog Timer. Note: This register field is sticky.
14-11 EIDLE_TIMER	EIDLE_TIMER is an internally reserved field. Do not use. Note: This register field is sticky.
10-0 SKP_INT_VAL	SKP Interval Value. The number of symbol times to wait between transmitting SKP ordered sets. Note that the controller actually waits the number of symbol times in this register plus 1 between transmitting SKP ordered sets. Your application must program this register accordingly. For example, if 1536 were programmed into this register (in a 250 MHz controller), then the controller actually transmits SKP ordered sets once every 1537 symbol times. The value programmed to this register is actually clock ticks and not symbol times. In a 125 MHz controller, programming the value programmed to this register should be scaled down by a factor of 2 (because one clock tick = two symbol times in this case). Note: This value is not used at Gen3 speed; the skip interval is hardcoded to 370 blocks. For M-PCIe configurations, if the 2K_PPM_DISABLED field in the M-PCIe Configuration Attribute is changed, then this field is changed automatically as follows. - 2K_PPM_DISABLED=1: 1280 / CX_NB - 2K_PPM_DISABLED=0: 228/CX_NB You need to set this field again if necessary when 2K_PPM_DISABLED is changed. Note: This register field is sticky.

11.4.3.1.67 Filter Mask 2 Register. (FILTER_MASK_2_OFF)

11.4.3.1.67.1 Offset

Register	Offset
FILTER_MASK_2_OFF	720h

11.4.3.1.67.2 Function

Filter Mask 2 Register. This register modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule.

11.4.3.1.67.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MASK_RADM_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_RADM_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.67.4 Fields

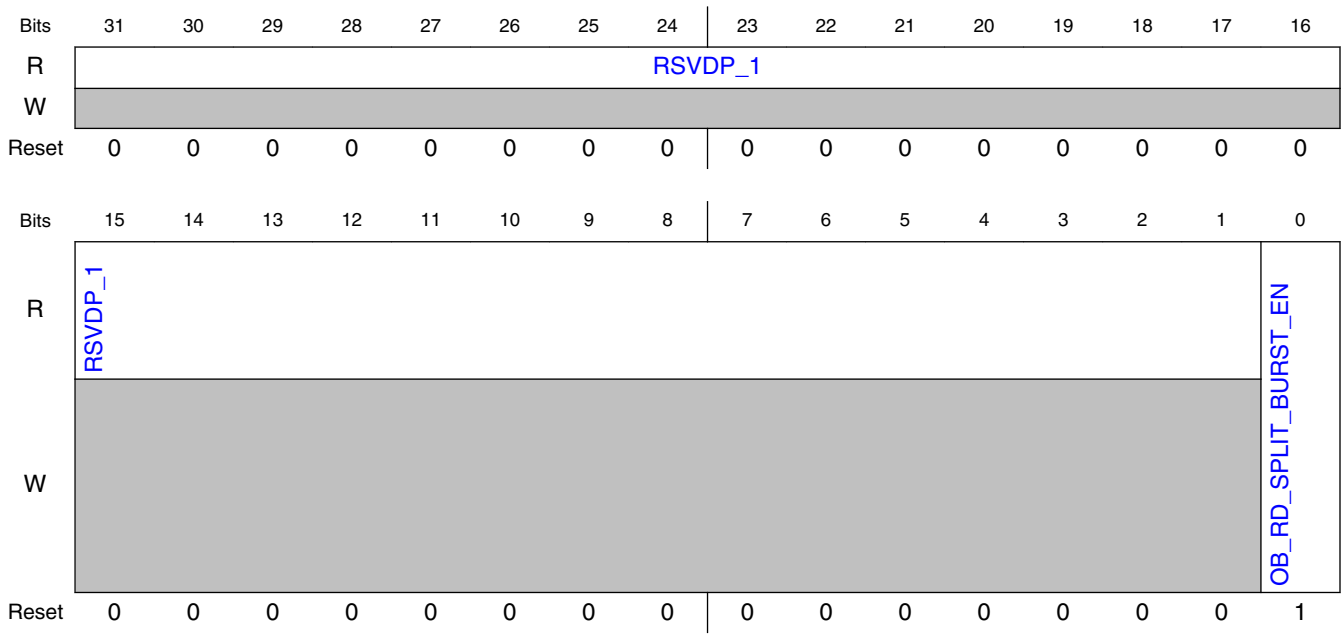
Field	Function
31-0 MASK_RADM_2	Filter Mask 2. This field modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule. [31:8]: Reserved [7]: CX_FLT_MASK_PRS_DROP - 0: Allow PRS message to pass through - 1: Drop PRS Messages silently - This bit is ignored when the CX_FLT_MASK_MSG_DROP bit in the MASK_RADM_1 field of the SYMBOL_TIMER_FILTER_1_OFF register is set to '1'. [6]: CX_FLT_UNMASK_TD - 0: Disable unmask TD bit if CX_STRIP_ECRC_ENABLE - 1: Enable unmask TD bit if CX_STRIP_ECRC_ENABLE [5]: CX_FLT_UNMASK_UR_POIS_TRGT0 - 0: Disable unmask CX_FLT_MASK_UR_POIS with TRGT0 destination - 1: Enable unmask CX_FLT_MASK_UR_POIS with TRGT0 destination [4]: CX_FLT_MASK_LN_VENMSG1_DROP - 0: Allow LN message to pass through - 1: Drop LN Messages silently [3]: CX_FLT_MASK_HANDLE_FLUSH - 0: Disable controller Filter to handle flush request - 1: Enable controller Filter to handle flush request [2]: CX_FLT_MASK_DABORT_4UCPL - 0: Enable DLLP abort for unexpected completion - 1: Do not enable DLLP abort for unexpected completion [1]: CX_FLT_MASK_VENMSG1_DROP - 0: Vendor MSG Type 1 dropped silently - 1: Vendor MSG Type 1 not dropped [0]: CX_FLT_MASK_VENMSG0_DROP - 0: Vendor MSG Type 0 dropped with UR error reporting - 1: Vendor MSG Type 0 not dropped Note: This register field is sticky.

11.4.3.1.68 AMBA Multiple Outbound Decomposed NP SubRequests Control Register. (AMBA_MUL_OB_DECOMP_NP_SUB_REQ_CTRL_OFF)

11.4.3.1.68.1 Offset

Register	Offset
AMBA_MUL_OB_DECOMP_NP_SUB_REQ_CTRL_OFF	724h

11.4.3.1.68.2 Diagram



11.4.3.1.68.3 Fields

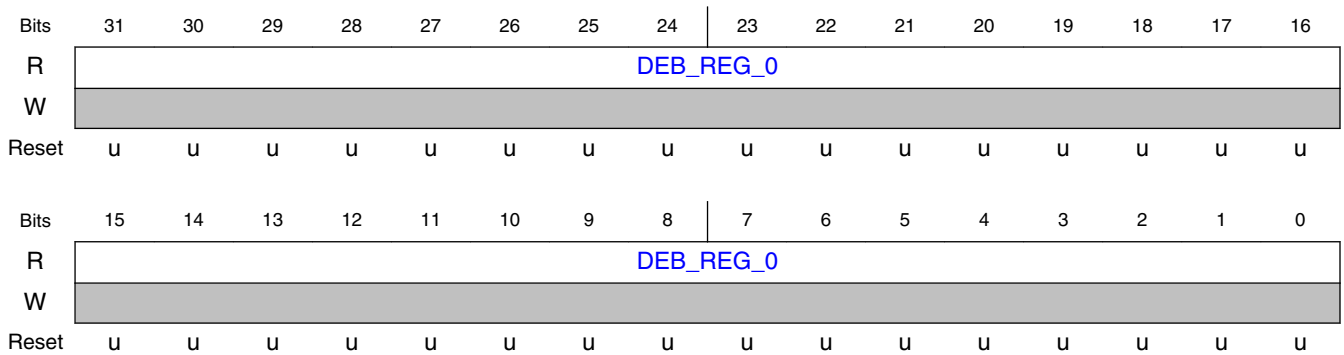
Field	Function
31-1 RSVDP_1	Reserved for future use.
0 OB_RD_SPLIT_BURST_EN	Enable AMBA Multiple Outbound Decomposed NP SubRequests. This bit when set to "0" disables the possibility of having multiple outstanding non-posted requests that were derived from decomposition of an outbound AMBA request. For more details, see "AXI Bridge Ordering" in the AXI chapter of the Databook. You should not clear this register unless your application master is requesting an amount of read data greater than Max_Read_Request_Size, and the remote device (or switch) is reordering completions that have different tags. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.

11.4.3.1.69 Debug Register 0 (PL_DEBUG0_OFF)

11.4.3.1.69.1 Offset

Register	Offset
PL_DEBUG0_OFF	728h

11.4.3.1.69.2 Diagram



11.4.3.1.69.3 Fields

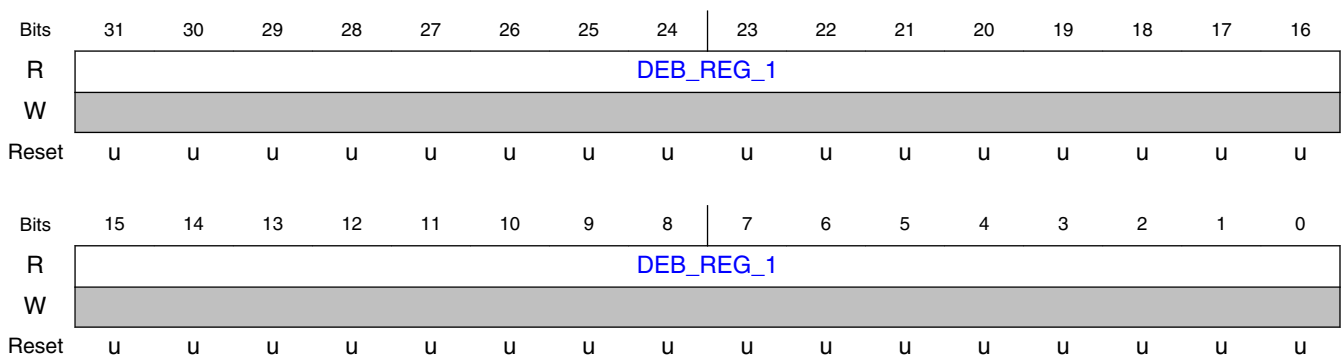
Field	Function
31-0 DEB_REG_0	The value on cxpl_debug_info[31:0].

11.4.3.1.70 Debug Register 1 (PL_DEBUG1_OFF)

11.4.3.1.70.1 Offset

Register	Offset
PL_DEBUG1_OFF	72Ch

11.4.3.1.70.2 Diagram



11.4.3.1.70.3 Fields

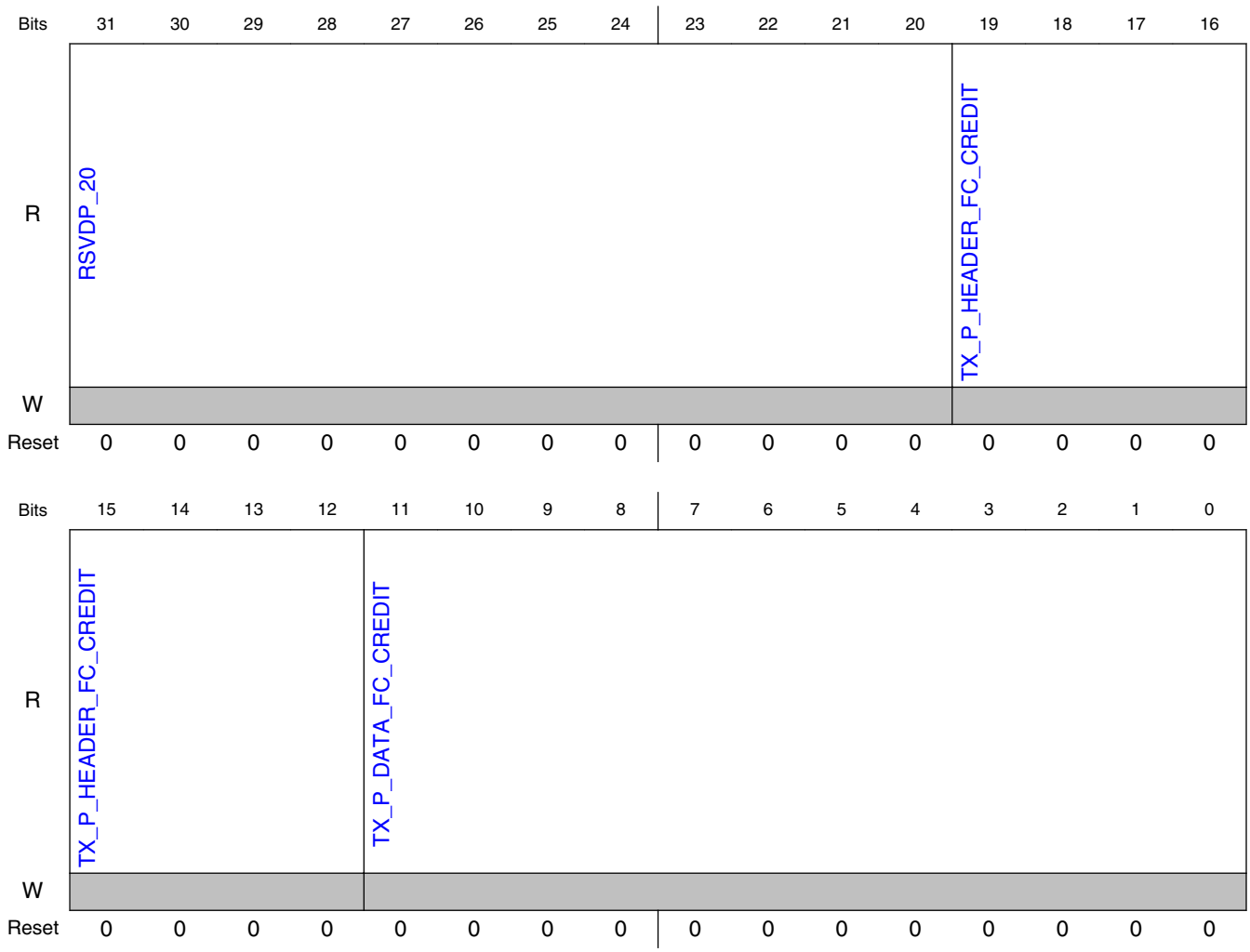
Field	Function
31-0 DEB_REG_1	The value on cxpl_debug_info[63:32].

11.4.3.1.71 Transmit Posted FC Credit Status (TX_P_FC_CREDIT_STATUS_OFF)

11.4.3.1.71.1 Offset

Register	Offset
TX_P_FC_CREDIT_STATUS_OFF	730h

11.4.3.1.71.2 Diagram



11.4.3.1.71.3 Fields

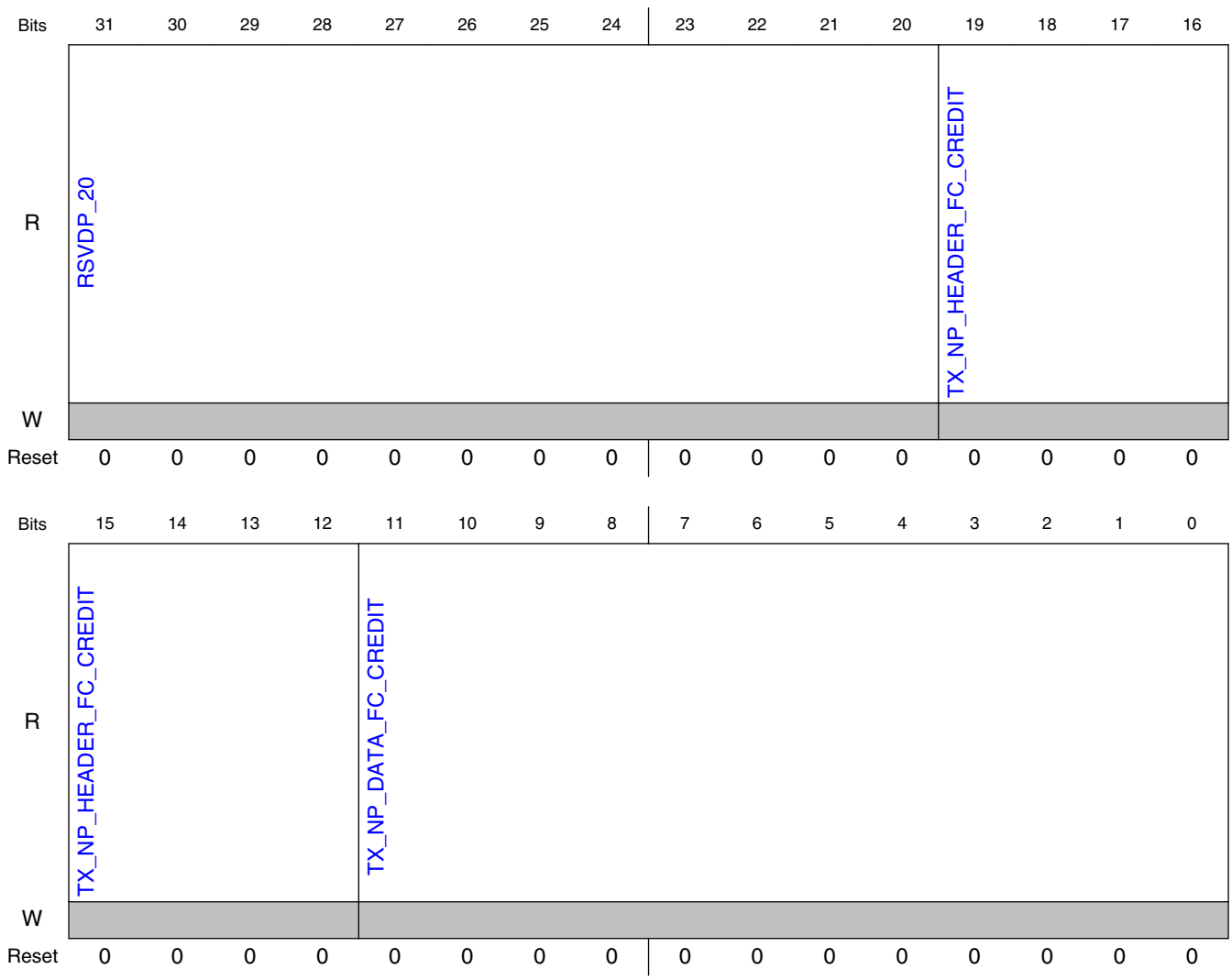
Field	Function
31-20 RSVDP_20	Reserved for future use.
19-12 TX_P_HEADER_FC_CREDIT	Transmit Posted Header FC Credits. The posted Header credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. Default value depends on the number of advertised credits for header and data [12'b0, xtlh_xadm_ph_cdts, xtlh_xadm_pd_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].
11-0 TX_P_DATA_FC_CREDIT	Transmit Posted Data FC Credits. The posted Data credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. Default value depends on the number of advertised credits for header and data [12'b0, xtlh_xadm_ph_cdts, xtlh_xadm_pd_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

11.4.3.1.72 Transmit Non-Posted FC Credit Status (TX_NP_FC_CREDIT_STATUS_OFF)

11.4.3.1.72.1 Offset

Register	Offset
TX_NP_FC_CREDIT_STATUS_OFF	734h

11.4.3.1.72.2 Diagram



11.4.3.1.72.3 Fields

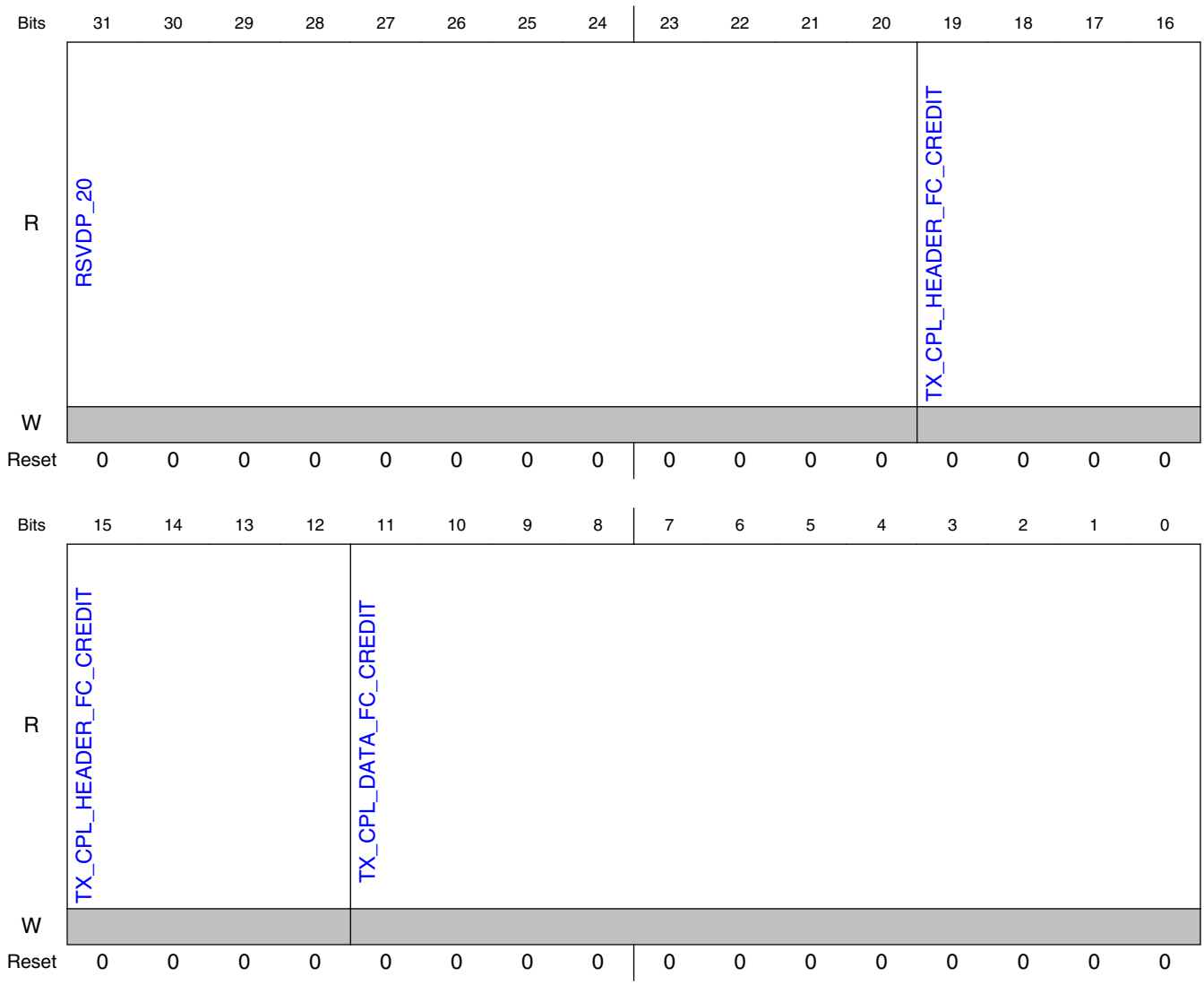
Field	Function
31-20 RSVDP_20	Reserved for future use.
19-12 TX_NP_HEADER_FC_CREDIT	Transmit Non-Posted Header FC Credits. The non-posted Header credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. Default value depends on the number of advertised credits for header and data [12'b0, xtlh_xadm_nph_cdts, xtlh_xadm_npd_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].
11-0 TX_NP_DATA_FC_CREDIT	Transmit Non-Posted Data FC Credits. The non-posted Data credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. Default value depends on the number of advertised credits for header and data [12'b0, xtlh_xadm_nph_cdts, xtlh_xadm_npd_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

11.4.3.1.73 Transmit Completion FC Credit Status (TX_CPL_FC_CREDIT_STATUS_OFF)

11.4.3.1.73.1 Offset

Register	Offset
TX_CPL_FC_CREDIT_STATUS_OFF	738h

11.4.3.1.73.2 Diagram



11.4.3.1.73.3 Fields

Field	Function
31-20 RSVDP_20	Reserved for future use.
19-12 TX_CPL_HEADER_FC_CREDIT	Transmit Completion Header FC Credits. The Completion Header credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. Default value depends on the number of advertised credits for header and data [12'b0, xtlh_xadm_cplh_cdts, xtlh_xadm_cpld_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].
11-0 TX_CPL_DATA_FC_CREDIT	Transmit Completion Data FC Credits. The Completion Data credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. Default value depends on the number of advertised credits for header and data [12'b0, xtlh_xadm_cplh_cdts, xtlh_xadm_cpld_cdts]; When the

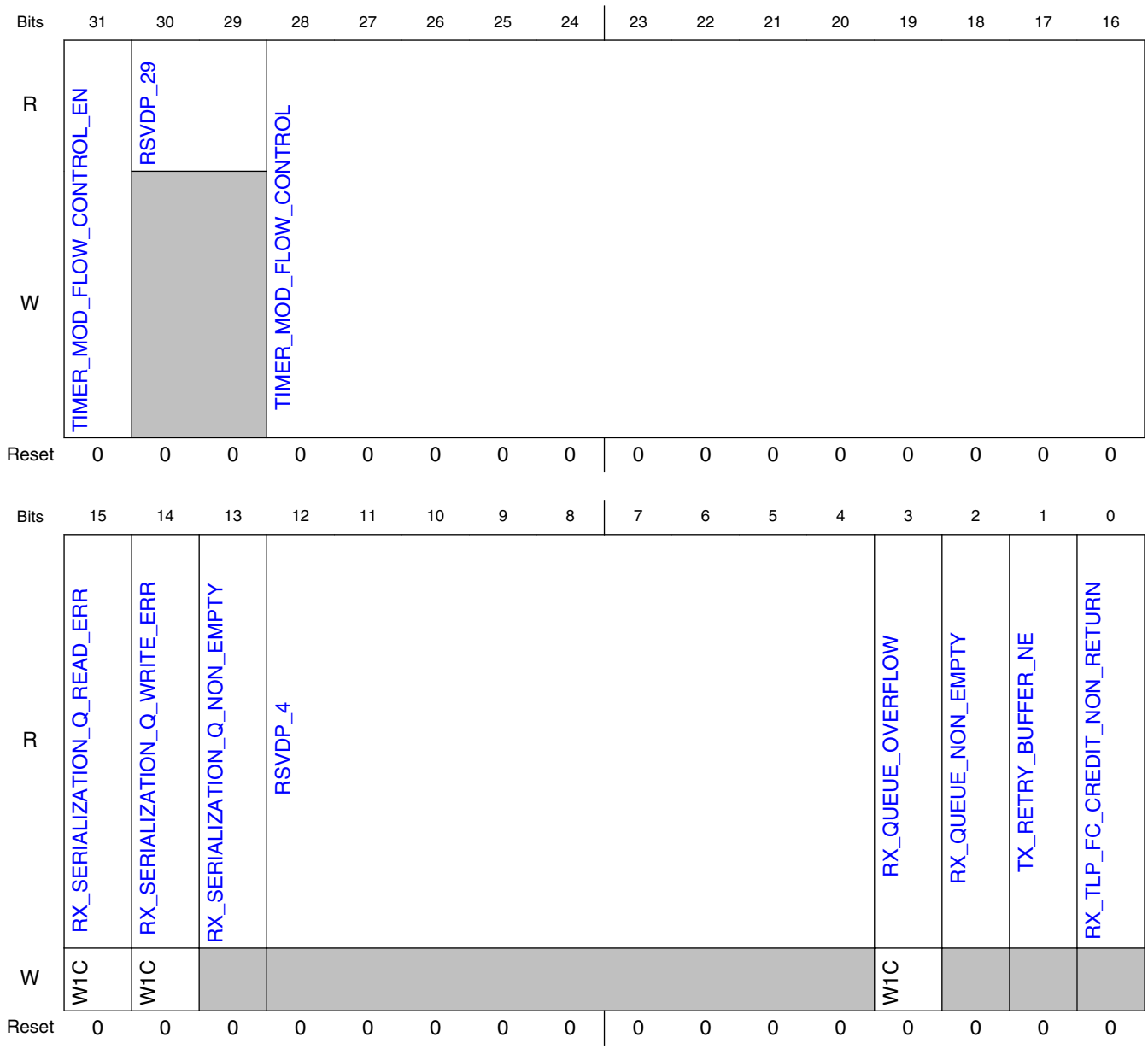
Field	Function
	number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

11.4.3.1.74 Queue Status (QUEUE_STATUS_OFF)

11.4.3.1.74.1 Offset

Register	Offset
QUEUE_STATUS_OFF	73Ch

11.4.3.1.74.2 Diagram



11.4.3.1.74.3 Fields

Field	Function
31 TIMER_MOD_FLOW_CONTROL_EN	FC Latency Timer Override Enable. When this bit is set, the value from the "FC Latency Timer Override Value" field in this register will override the FC latency timer value that the controller calculates according to the PCIe specification. Note: This register field is sticky.
30-29 RSVDP_29	Reserved for future use.

Table continues on the next page...

Field	Function
28-16 TIMER_MOD_FLOW_CONTROL	FC Latency Timer Override Value. When you set the "FC Latency Timer Override Enable" in this register, the value in this field will override the FC latency timer value that the controller calculates according to the PCIe specification. For more details, see "Flow Control". Note: This register field is sticky.
15 RX_SERIALIZATION_Q_READ_ERR	Receive Serialization Read Error. Indicates the serialization queue has attempted to read an incorrectly formatted TLP.
14 RX_SERIALIZATION_Q_WRITE_ERR	Receive Serialization Queue Write Error. Indicates insufficient buffer space available to write to the serialization queue.
13 RX_SERIALIZATION_Q_NON_EMPTY	Receive Serialization Queue Not Empty. Indicates there is data in the serialization queue.
12-4 RSVDP_4	Reserved for future use.
3 RX_QUEUE_OVERFLOW	Receive Credit Queue Overflow. Indicates insufficient buffer space available to write to the P/NP/CPL credit queue.
2 RX_QUEUE_NON_EMPTY	Receive Credit Queue Not Empty. Indicates there is data in one or more of the receive buffers.
1 TX_RETRY_BUFFER_NE	Transmit Retry Buffer Not Empty. Indicates that there is data in the transmit retry buffer.
0 RX_TLP_FC_CREDIT_NON_RETURN	Received TLP FC Credits Not Returned. Indicates that the controller has received a TLP but has not yet sent an UpdateFC DLLP indicating that the credits for that TLP have been restored by the receiver at the other end of the link.

11.4.3.1.75 VC Transmit Arbitration Register 1 (VC_TX_ARBI_1_OFF)

11.4.3.1.75.1 Offset

Register	Offset
VC_TX_ARBI_1_OFF	740h

11.4.3.1.75.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								WRR_WEIGHT_VC_0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

11.4.3.1.75.3 Fields

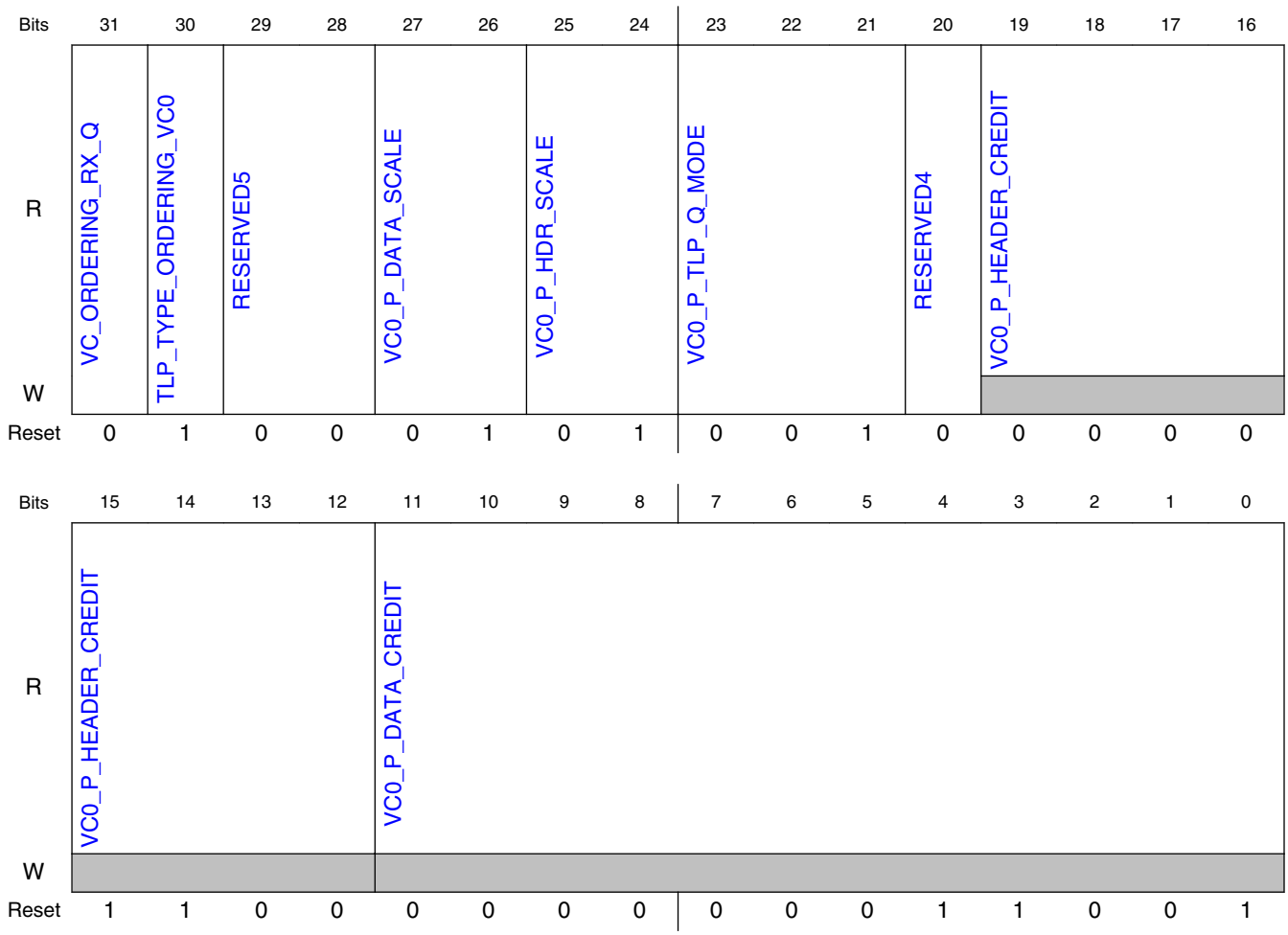
Field	Function
31-24 —	Reserved
23-16 —	Reserved
15-8 —	Reserved
7-0 WRR_WEIGHT_VC_0	WRR Weight for VC0. Note: The access attributes of this field are as follows: - Dbi: R

11.4.3.1.76 Segmented-Buffer VC0 Posted Receive Queue Control. (VC0_P_RX_Q_CTRL_OFF)

11.4.3.1.76.1 Offset

Register	Offset
VC0_P_RX_Q_CTRL_OFF	748h

11.4.3.1.76.2 Diagram



11.4.3.1.76.3 Fields

Field	Function
31 VC_ORDERING_RX_Q	VC Ordering for Receive Queues. Determines the VC ordering rule for the receive queues, used only in the segmented-buffer configuration: - 1: Strict ordering, higher numbered VCs have higher priority - 0: Round robin Note: This register field is sticky.
30 TLP_TYPE_ORDERING_VC0	TLP Type Ordering for VC0. Determines the TLP type ordering rule for VC0 receive queues, used only in the segmented-buffer configuration: - 1: PCIe ordering rules (recommended) - 0: Strict ordering: posted, completion, then non-posted Note: This register field is sticky.
29-28 RESERVED5	Reserved Note: This register field is sticky.
27-26 VC0_P_DATA_SCALE	VC0 Scale Posted Data Credites. Note: This register field is sticky.
25-24	VC0 Scale Posted Header Credites. Note: This register field is sticky.

Table continues on the next page...

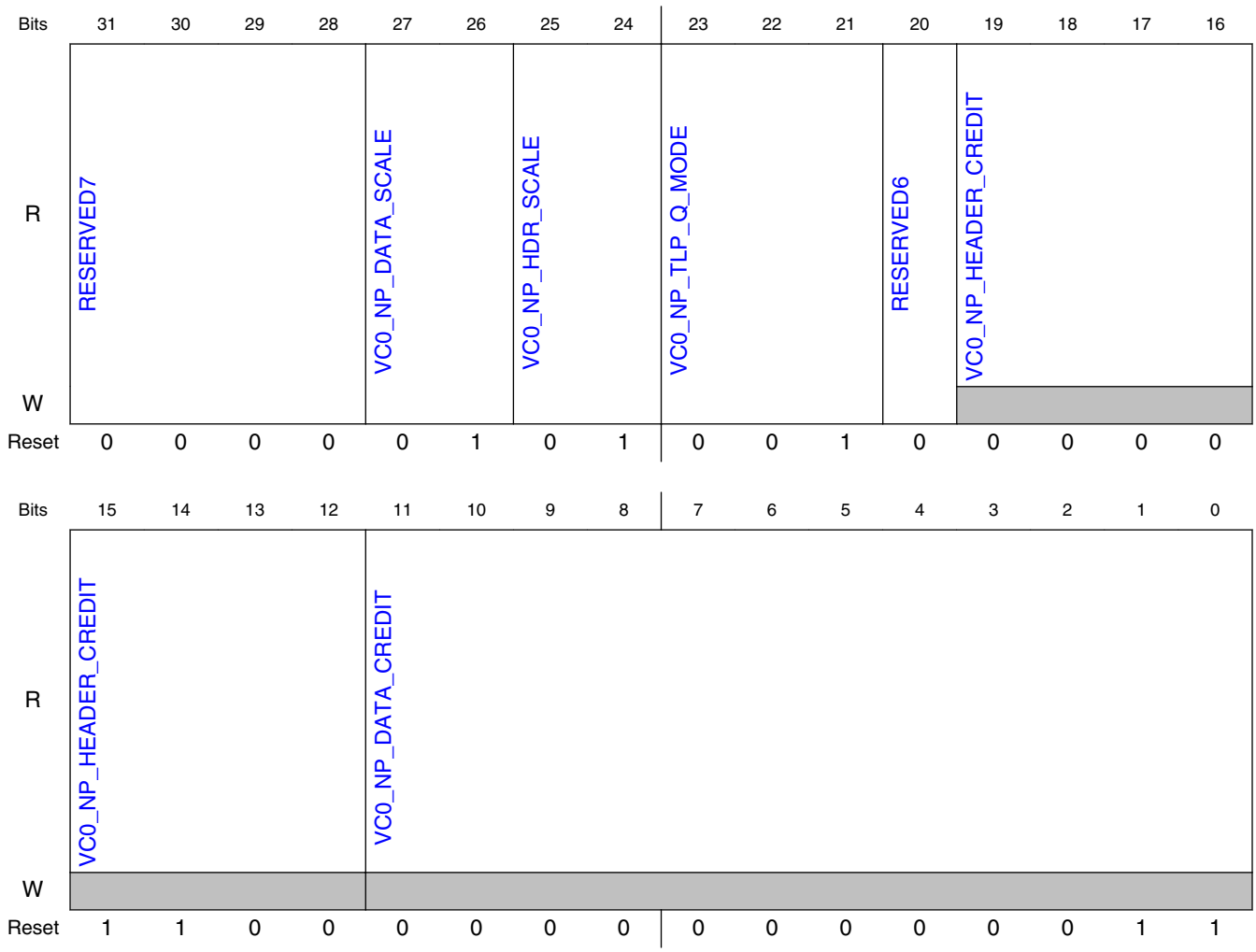
Field	Function
VC0_P_HDR_SCALE	
23-21 VC0_P_TLP_Q_MODE	Reserved Note: This register field is sticky.
20 RESERVED4	Reserved Note: This register field is sticky.
19-12 VC0_P_HEADER_CREDIT	VC0 Posted Header Credits. The number of initial posted header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Dbi: R (sticky) Note: This register field is sticky.
11-0 VC0_P_DATA_CREDIT	VC0 Posted Data Credits. The number of initial posted data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Dbi: R (sticky) Note: This register field is sticky.

11.4.3.1.77 Segmented-Buffer VC0 Non-Posted Receive Queue Control. (VC0_NP_RX_Q_CTRL_OFF)

11.4.3.1.77.1 Offset

Register	Offset
VC0_NP_RX_Q_CTRL_OFF	74Ch

11.4.3.1.77.2 Diagram



11.4.3.1.77.3 Fields

Field	Function
31-28 RESERVED7	Reserved Note: This register field is sticky.
27-26 VC0_NP_DATA_SCALE	VC0 Scale Non-Posted Data Credites. Note: This register field is sticky.
25-24 VC0_NP_HDR_SCALE	VC0 Scale Non-Posted Header Credites. Note: This register field is sticky.
23-21 VC0_NP_TLP_Q_MODE	Reserved Note: This register field is sticky.

Table continues on the next page...

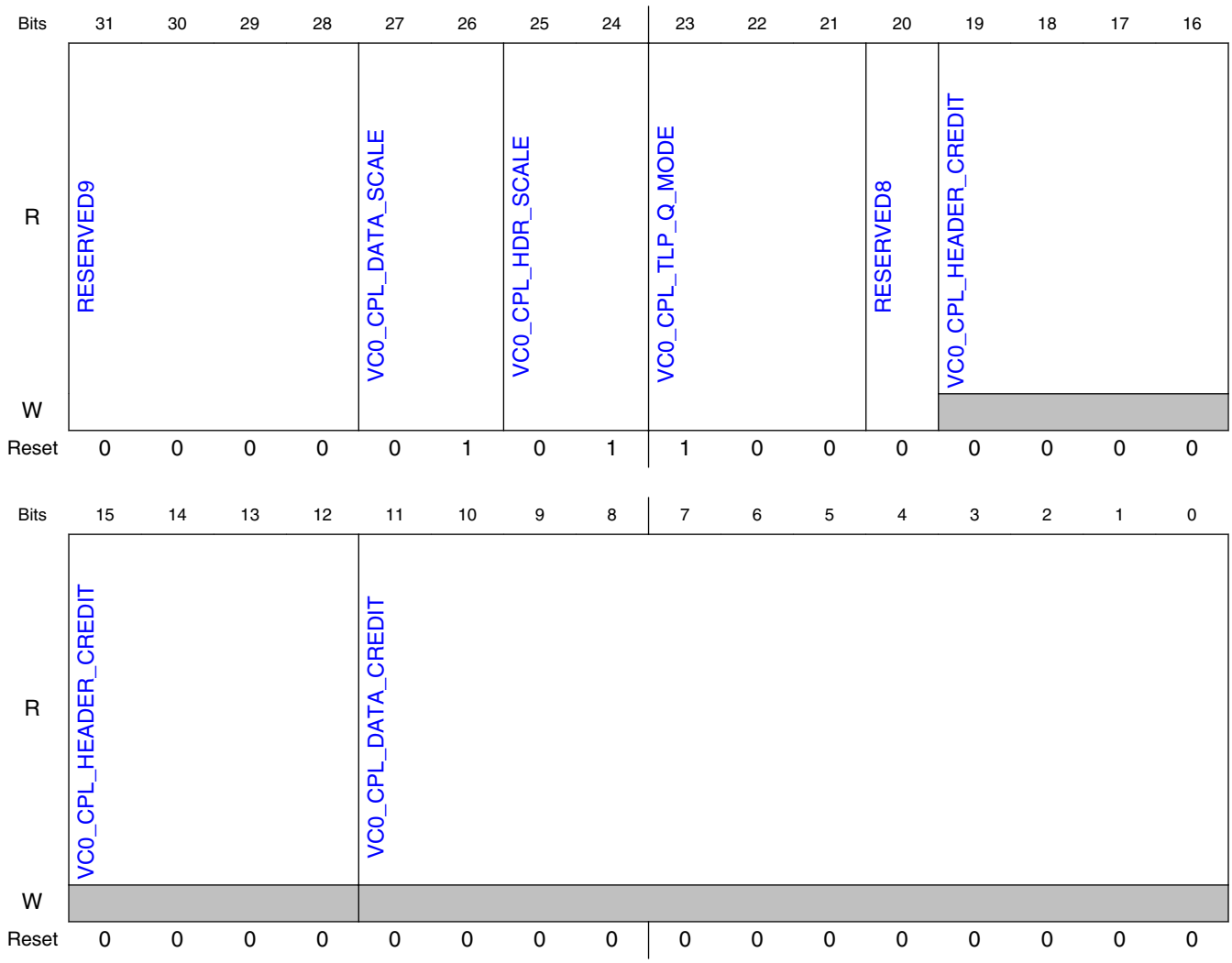
Field	Function
20 RESERVED6	Reserved Note: This register field is sticky.
19-12 VC0_NP_HEADER_CREDIT	VC0 Non-Posted Header Credits. The number of initial non-posted header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Dbi: R (sticky) Note: This register field is sticky.
11-0 VC0_NP_DATA_CREDIT	VC0 Non-Posted Data Credits. The number of initial non-posted data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Dbi: R (sticky) Note: This register field is sticky.

11.4.3.1.78 Segmented-Buffer VC0 Completion Receive Queue Control. (VC0_CPL_RX_Q_CTRL_OFF)

11.4.3.1.78.1 Offset

Register	Offset
VC0_CPL_RX_Q_CTRL_OFF	750h

11.4.3.1.78.2 Diagram



11.4.3.1.78.3 Fields

Field	Function
31-28 RESERVED9	Reserved Note: This register field is sticky.
27-26 VC0_CPL_DATA_SCALE	VC0 Scale CPL Data Credites. Note: This register field is sticky.
25-24 VC0_CPL_HDR_SCALE	VC0 Scale CPL Header Credites. Note: This register field is sticky.
23-21	Reserved Note: This register field is sticky.

Table continues on the next page...

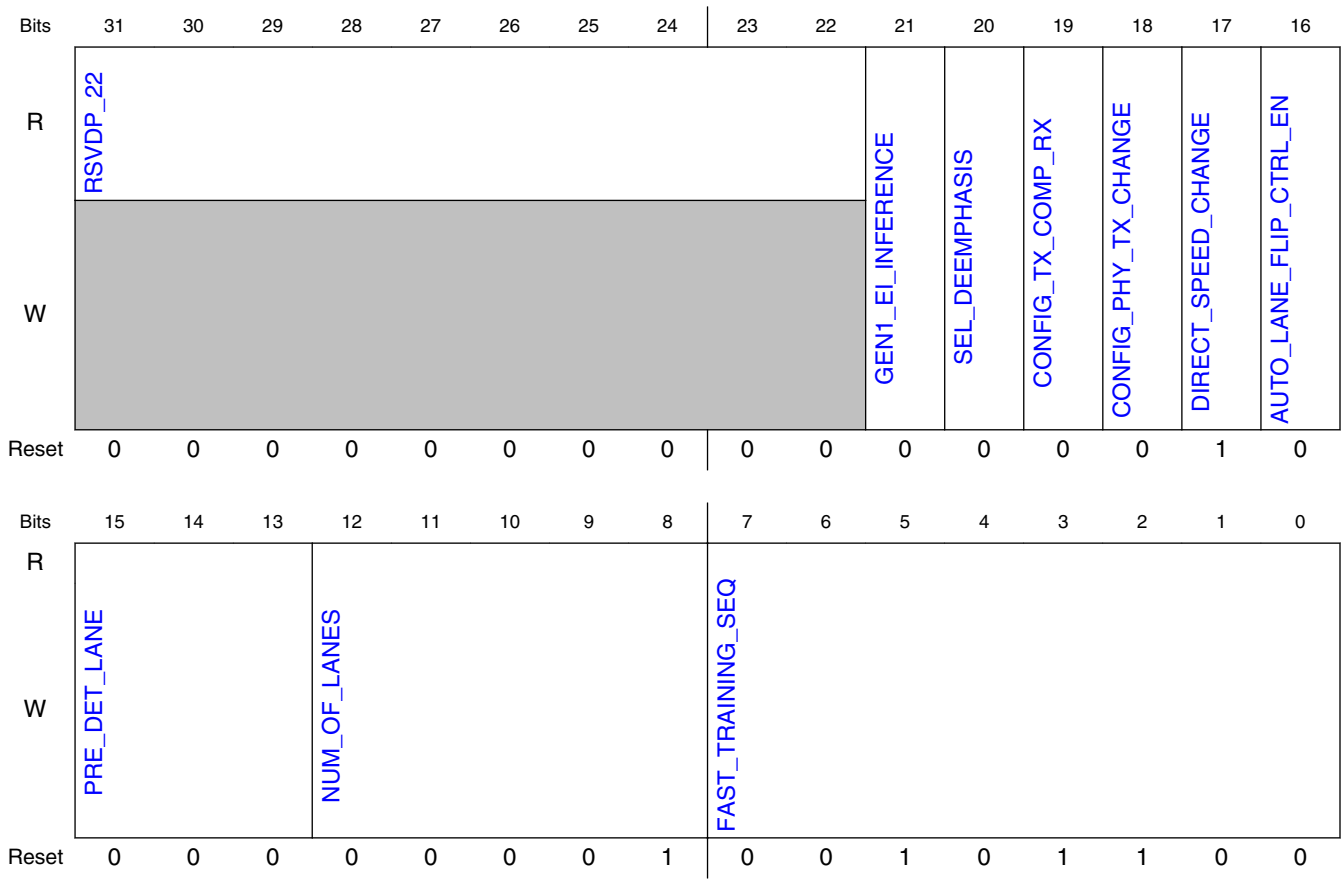
Field	Function
VC0_CPL_TLP_Q_MODE	
20 RESERVED8	Reserved Note: This register field is sticky.
19-12 VC0_CPL_HEADER_CREDIT	VC0 Completion Header Credits. The number of initial Completion header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Dbi: R (sticky) Note: This register field is sticky.
11-0 VC0_CPL_DATA_CREDIT	VC0 Completion Data Credits. The number of initial Completion data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Dbi: R (sticky) Note: This register field is sticky.

11.4.3.1.79 Link Width and Speed Change Control Register. (GEN2_CTRL_OFF)

11.4.3.1.79.1 Offset

Register	Offset
GEN2_CTRL_OFF	80Ch

11.4.3.1.79.2 Diagram



11.4.3.1.79.3 Fields

Field	Function
31-22 RSVDP_22	Reserved for future use.
21 GEN1_EI_INFERENCE	Electrical Idle Inference Mode at Gen1 Rate. Programmable mode to determine inferred electrical idle (EI) in Recovery.Speed or Loopback.Active (as slave) state at Gen1 speed by looking for a "1" value on RxElecIdle instead of looking for a "0" on RxValid. If the PHY fails to deassert the RxValid signal in Recovery.Speed or Loopback.Active (because of corrupted EIOS for example), then EI cannot be inferred successfully in the controller by just detecting the condition RxValid=0. - 0: Use RxElecIdle signal to infer Electrical Idle - 1: Use RxValid signal to infer Electrical Idle Note: This register field is sticky.
20 SEL_DEEMPHASIS	Used to set the de-emphasis level for upstream ports. This bit selects the level of de-emphasis the link operates at. - 0: -6 dB - 1: -3.5 dB This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
19 CONFIG_TX_COMP_RX	Config Tx Compliance Receive Bit. When set to 1, signals LTSSM to transmit TS ordered sets with the compliance receive bit assert (equal to "1"). This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
18	Config PHY Tx Swing. Controls the PHY transmitter voltage swing level. The controller drives the mac_phy_txswing output from this register bit field. - 0: Full Swing - 1: Low Swing This field is reserved

Table continues on the next page...

Field	Function
CONFIG_PHY_TX_CHANGE	(fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
17 DIRECT_SPEED_CHANGE	Directed Speed Change. Writing "1" to this field instructs the LTSSM to initiate a speed change to Gen2 or Gen3 after the link is initialized at Gen1 speed. When the speed change occurs, the controller will clear the contents of this field; and a read to this field by your software will return a "0". To manually initiate the speed change: - Write to LINK_CONTROL2_LINK_STATUS2_REG . PCIE_CAP_TARGET_LINK_SPEED in the local device - Deassert this field - Assert this field If you set the default of this field using the DEFAULT_GEN2_SPEED_CHANGE configuration parameter to "1", then the speed change is initiated automatically after link up, and the controller clears the contents of this field. If you want to prevent this automatic speed change, then write a lower speed value to the Target Link Speed field of the Link Control 2 register (LINK_CONTROL2_LINK_STATUS2_OFF . PCIE_CAP_TARGET_LINK_SPEED) through the DBI before link up. This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W
16 AUTO_LANE_FLIP_CTRL_EN	Enable Auto flipping of the lanes. You must set the CX_AUTO_LANE_FLIP_CTRL_EN configuration parameter to include the hardware for this feature in the controller. For more details, see the 'Lane Reversal' appendix in the Databook. This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
15-13 PRE_DET_LANE	Predetermined Lane for Auto Flip. This field defines which physical lane is connected to logical Lane0 by the flip operation performed in Detect. Allowed values are: - 3'b000: Connect logical Lane0 to physical lane 0 or CX_NL-1 or CX_NL/2-1 or CX_NL/4-1 or CX_NL/8-1, depending on which lane is detected - 3'b001: Connect logical Lane0 to physical lane 1 - 3'b010: Connect logical Lane0 to physical lane 3 - 3'b011: Connect logical Lane0 to physical lane 7 - 3'b100: Connect logical Lane0 to physical lane 15 This field is used to restrict the receiver detect procedure to a particular lane when the default detect and polling procedure performed on all lanes cannot be successful. A notable example of when it is useful to program this field to a value different from the default, is when a lane is asymmetrically broken, that is, it is detected in Detect LTSSM state but it cannot exit Electrical Idle in Polling LTSSM state. Note: This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
12-8 NUM_OF_LANES	Predetermined Number of Lanes. Defines the number of lanes which are connected and not bad. Used to limit the effective link width to ignore 'broken" or "unused" lanes that detect a receiver. Indicates the number of lanes to check for exit from Electrical Idle in Polling.Active and L2.Idle. It is possible that the LTSSM might detect a receiver on a bad or broken lane during the Detect Substate. However, it is also possible that such a lane might also fail to exit Electrical Idle and therefore prevent a valid link from being configured. This value is referred to as the "Predetermined Number of Lanes" in section 4.2.6.2.1 of the PCI Express Base 3.0 Specification, revision 1.0. Encoding is as follows: - 0x01: 1 lane - 0x02: 2 lanes - 0x03: 3 lanes - .. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Link Mode Enable" field of PORT_LINK_CTRL_OFF. The value in this register is normally the same as the encoded value in PORT_LINK_CTRL_OFF. If you find that one of your used lanes is bad then you must reduce the value in this register. For more information, see "How to Tie Off Unused Lanes." For information on upsizing and downsizing the link width, see "Link Establishment." This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
7-0 FAST_TRAINING_SEQ	Sets the Number of Fast Training Sequences (N_FTS) that the controller advertises as its N_FTS during Gen2 or Gen3 link training. This value is used to inform the link partner about the PHY's ability to recover synchronization after a low power state. The number should be provided by the PHY vendor. Do not set N_FTS to zero; doing so can cause the LTSSM to go into the recovery state when exiting from L0s. This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.

11.4.3.1.80 PHY Status Register. (PHY_STATUS_OFF)

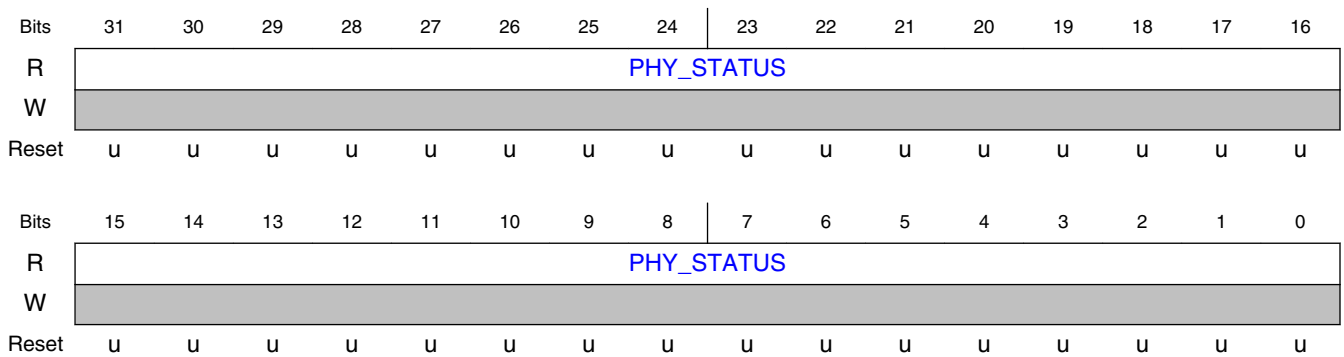
11.4.3.1.80.1 Offset

Register	Offset
PHY_STATUS_OFF	810h

11.4.3.1.80.2 Function

PHY Status Register. Memory mapped register from phy_cfg_status GPIO input pins.

11.4.3.1.80.3 Diagram



11.4.3.1.80.4 Fields

Field	Function
31-0 PHY_STATUS	PHY Status. Data received directly from the phy_cfg_status bus. These is a GPIO register reflecting the values on the static phy_cfg_status input signals. The usage is left completely to the user and does not in any way influence controller functionality. You can use it for any static sideband status signalling requirements that you have for your PHY. This field is reserved (fixed to '0') for M-PCIe. Note: This register field is sticky.

11.4.3.1.81 PHY Control Register. (PHY_CONTROL_OFF)

11.4.3.1.81.1 Offset

Register	Offset
PHY_CONTROL_OFF	814h

11.4.3.1.81.2 Function

PHY Control Register. Memory mapped register to cfg_phy_control GPIO output pins.

11.4.3.1.81.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PHY_CONTROL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHY_CONTROL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.81.4 Fields

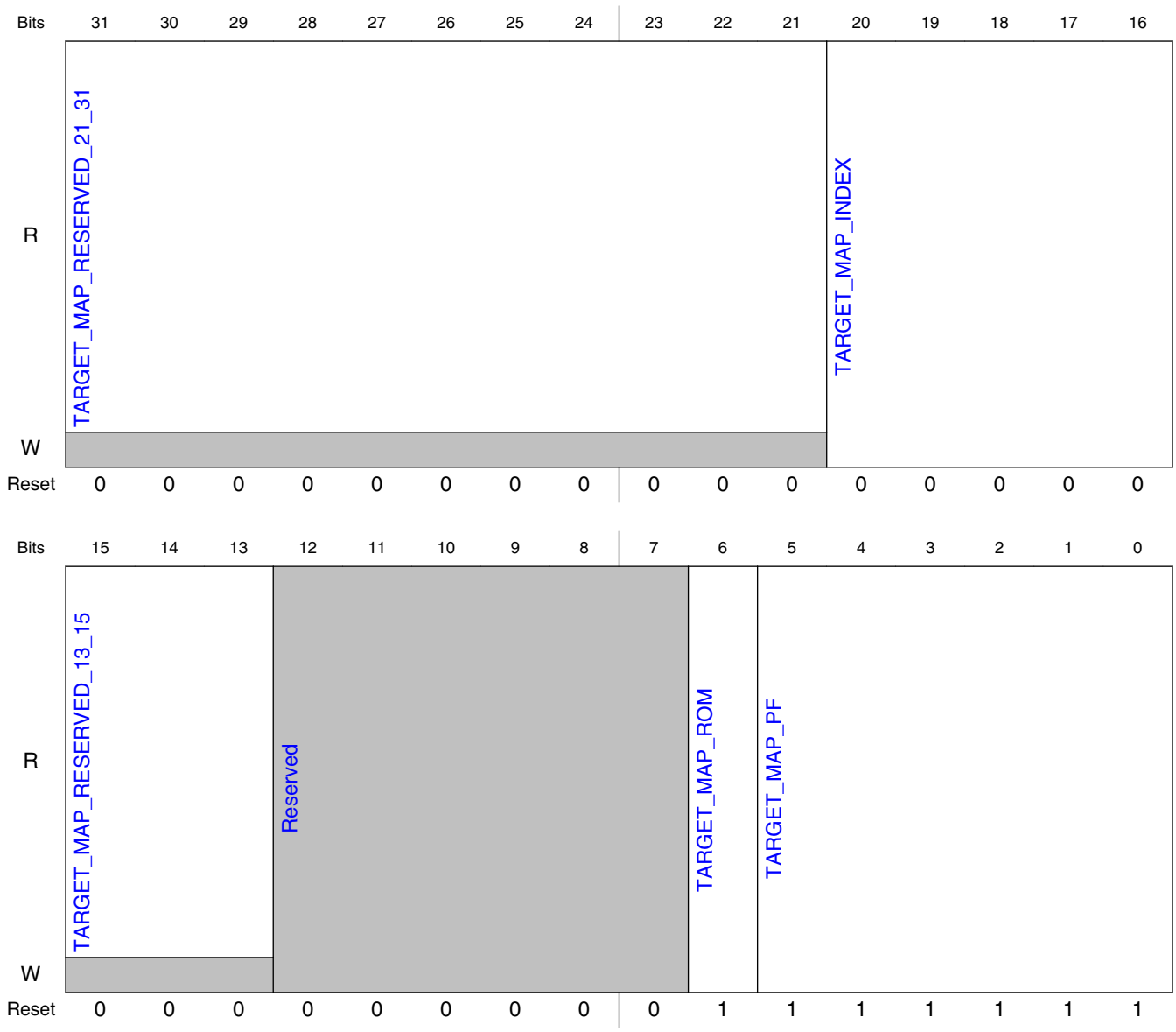
Field	Function
31-0 PHY_CONTROL	PHY Control. Data sent directly to the cfg_phy_control bus. These is a GPIO register driving the values on the static cfg_phy_control output signals. The usage is left completely to the user and does not in any way influence controller functionality. You can use it for any static sideband control signalling requirements that you have for your PHY. This field is reserved (fixed to '0') for M-PCIe. Note: This register field is sticky.

11.4.3.1.82 Programmable Target Map Control Register. (TRGT_MAP_CTRL_OFF)

11.4.3.1.82.1 Offset

Register	Offset
TRGT_MAP_CTRL_OFF	81Ch

11.4.3.1.82.2 Diagram



11.4.3.1.82.3 Fields

Field	Function
31-21 TARGET_MAP_RESERVED_21_31	Reserved Note: The access attributes of this field are as follows: - Dbi: R (sticky)
20-16 TARGET_MAP_INDEX	The number of the PF Function on which the Target Values are set. This register does not respect the Byte Enable setting. any write will affect all register bits.

Table continues on the next page...

Field	Function
15-13 TARGET_MAP_RESERVED_13_15	Reserved Note: The access attributes of this field are as follows: - Dbi: R (sticky)
12-7 —	Reserved
6 TARGET_MAP_ROM	Target Value for the ROM page of the PF Function selected by the index number. This register does not respect the Byte Enable setting. any write will affect all register bits.
5-0 TARGET_MAP_PF	Target Values for each BAR on the PF Function selected by the index number. This register does not respect the Byte Enable setting. any write will affect all register bits.

11.4.3.1.83 Integrated MSI Reception Module (iMRM) Address Register. (MSI_CTRL_ADDR_OFF)

11.4.3.1.83.1 Offset

Register	Offset
MSI_CTRL_ADDR_OFF	820h

11.4.3.1.83.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.83.3 Fields

Field	Function
31-0	Integrated MSI Reception Module Address. System specified address for MSI memory write transaction termination. Within the AXI Bridge, every received Memory Write request is examined to see if it targets the MSI Address that has been specified in this register; and also to see if it satisfies the definition of an

Field	Function
MSI_CTRL_ADDR	MSI interrupt request. When these conditions are satisfied the Memory Write request is marked as an MSI request. Note: This register field is sticky.

11.4.3.1.84 Integrated MSI Reception Module Upper Address Register. (MSI_CTRL_UPPER_ADDR_OFF)

11.4.3.1.84.1 Offset

Register	Offset
MSI_CTRL_UPPER_ADDR_OFF	824h

11.4.3.1.84.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_UPPER_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_UPPER_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.84.3 Fields

Field	Function
31-0 MSI_CTRL_UPPER_ADDR	Integrated MSI Reception Module Upper Address. System specified upper address for MSI memory write transaction termination. Allows functions to support a 64-bit MSI address. Note: This register field is sticky.

11.4.3.1.85 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_0_EN_OFF)

11.4.3.1.85.1 Offset

Register	Offset
MSI_CTRL_INT_0_EN_OFF	828h

11.4.3.1.85.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_0_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_0_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.85.3 Fields

Field	Function
31-0 MSI_CTRL_INT_0_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.86 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_0_MASK_OFF)

11.4.3.1.86.1 Offset

Register	Offset
MSI_CTRL_INT_0_MASK_OFF	82Ch

11.4.3.1.86.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_0_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_0_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.86.3 Fields

Field	Function
31-0 MSI_CTRL_INT_0_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.87 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_0_STATUS_OFF)

11.4.3.1.87.1 Offset

Register	Offset
MSI_CTRL_INT_0_STATUS_OFF	830h

11.4.3.1.87.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_0_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_0_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.87.3 Fields

Field	Function
31-0 MSI_CTRL_INT_0_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.4.3.1.88 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_1_EN_OFF)

11.4.3.1.88.1 Offset

Register	Offset
MSI_CTRL_INT_1_EN_OFF	834h

11.4.3.1.88.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_1_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_1_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.88.3 Fields

Field	Function
31-0 MSI_CTRL_INT_1_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.89 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_1_MASK_OFF)

11.4.3.1.89.1 Offset

Register	Offset
MSI_CTRL_INT_1_MASK_OFF	838h

11.4.3.1.89.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_1_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_1_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.89.3 Fields

Field	Function
31-0 MSI_CTRL_INT_1_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.90 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_1_STATUS_OFF)

11.4.3.1.90.1 Offset

Register	Offset
MSI_CTRL_INT_1_STATUS_OFF	83Ch

11.4.3.1.90.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_1_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_1_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.90.3 Fields

Field	Function
31-0 MSI_CTRL_INT_1_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.4.3.1.91 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_2_EN_OFF)

11.4.3.1.91.1 Offset

Register	Offset
MSI_CTRL_INT_2_EN_OFF	840h

11.4.3.1.91.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_2_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_2_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.91.3 Fields

Field	Function
31-0 MSI_CTRL_INT_2_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.92 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_2_MASK_OFF)

11.4.3.1.92.1 Offset

Register	Offset
MSI_CTRL_INT_2_MASK_OFF	844h

11.4.3.1.92.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_2_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_2_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.92.3 Fields

Field	Function
31-0 MSI_CTRL_INT_2_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.93 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_2_STATUS_OFF)

11.4.3.1.93.1 Offset

Register	Offset
MSI_CTRL_INT_2_STATUS_OFF	848h

11.4.3.1.93.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_2_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_2_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.93.3 Fields

Field	Function
31-0 MSI_CTRL_INT_2_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.4.3.1.94 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_3_EN_OFF)

11.4.3.1.94.1 Offset

Register	Offset
MSI_CTRL_INT_3_EN_OFF	84Ch

11.4.3.1.94.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_3_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_3_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.94.3 Fields

Field	Function
31-0 MSI_CTRL_INT_3_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.95 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_3_MASK_OFF)

11.4.3.1.95.1 Offset

Register	Offset
MSI_CTRL_INT_3_MASK_OFF	850h

11.4.3.1.95.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_3_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_3_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.95.3 Fields

Field	Function
31-0 MSI_CTRL_INT_3_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.96 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_3_STATUS_OFF)

11.4.3.1.96.1 Offset

Register	Offset
MSI_CTRL_INT_3_STATUS_OFF	854h

11.4.3.1.96.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_3_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_3_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.96.3 Fields

Field	Function
31-0 MSI_CTRL_INT_3_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.4.3.1.97 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_4_EN_OFF)

11.4.3.1.97.1 Offset

Register	Offset
MSI_CTRL_INT_4_EN_OFF	858h

11.4.3.1.97.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_4_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_4_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.97.3 Fields

Field	Function
31-0 MSI_CTRL_INT_4_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.98 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_4_MASK_OFF)

11.4.3.1.98.1 Offset

Register	Offset
MSI_CTRL_INT_4_MASK_OFF	85Ch

11.4.3.1.98.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_4_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_4_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.98.3 Fields

Field	Function
31-0 MSI_CTRL_INT_4_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.99 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_4_STATUS_OFF)

11.4.3.1.99.1 Offset

Register	Offset
MSI_CTRL_INT_4_STATUS_OFF	860h

11.4.3.1.99.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_4_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_4_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.99.3 Fields

Field	Function
31-0 MSI_CTRL_INT_4_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.4.3.1.100 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_5_EN_OFF)

11.4.3.1.100.1 Offset

Register	Offset
MSI_CTRL_INT_5_EN_OFF	864h

11.4.3.1.100.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_5_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_5_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.100.3 Fields

Field	Function
31-0 MSI_CTRL_INT_5_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.101 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_5_MASK_OFF)

11.4.3.1.101.1 Offset

Register	Offset
MSI_CTRL_INT_5_MASK_OFF	868h

11.4.3.1.101.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_5_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_5_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.101.3 Fields

Field	Function
31-0 MSI_CTRL_INT_5_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.102 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_5_STATUS_OFF)

11.4.3.1.102.1 Offset

Register	Offset
MSI_CTRL_INT_5_STATUS_OFF	86Ch

11.4.3.1.102.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_5_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_5_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.102.3 Fields

Field	Function
31-0 MSI_CTRL_INT_5_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.4.3.1.103 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_6_EN_OFF)

11.4.3.1.103.1 Offset

Register	Offset
MSI_CTRL_INT_6_EN_OFF	870h

11.4.3.1.103.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_6_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_6_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.103.3 Fields

Field	Function
31-0 MSI_CTRL_INT_6_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.104 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_6_MASK_OFF)

11.4.3.1.104.1 Offset

Register	Offset
MSI_CTRL_INT_6_MASK_OFF	874h

11.4.3.1.104.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_6_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_6_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.104.3 Fields

Field	Function
31-0 MSI_CTRL_INT_6_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.105 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_6_STATUS_OFF)

11.4.3.1.105.1 Offset

Register	Offset
MSI_CTRL_INT_6_STATUS_OFF	878h

11.4.3.1.105.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_6_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_6_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.105.3 Fields

Field	Function
31-0 MSI_CTRL_INT_6_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.4.3.1.106 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_7_EN_OFF)

11.4.3.1.106.1 Offset

Register	Offset
MSI_CTRL_INT_7_EN_OFF	87Ch

11.4.3.1.106.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_7_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_7_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.106.3 Fields

Field	Function
31-0 MSI_CTRL_INT_7_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.107 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_7_MASK_OFF)

11.4.3.1.107.1 Offset

Register	Offset
MSI_CTRL_INT_7_MASK_OFF	880h

11.4.3.1.107.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_7_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_7_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.107.3 Fields

Field	Function
31-0 MSI_CTRL_INT_7_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.4.3.1.108 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_7_STATUS_OFF)

11.4.3.1.108.1 Offset

Register	Offset
MSI_CTRL_INT_7_STATUS_OFF	884h

11.4.3.1.108.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_7_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_7_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.108.3 Fields

Field	Function
31-0 MSI_CTRL_INT_7_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.4.3.1.109 Integrated MSI Reception Module General Purpose IO Register. (MSI_GPIO_IO_OFF)

11.4.3.1.109.1 Offset

Register	Offset
MSI_GPIO_IO_OFF	888h

11.4.3.1.109.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_GPIO_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_GPIO_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.109.3 Fields

Field	Function
31-0 MSI_GPIO_REG	MSI GPIO Register. The contents of this register drives the top-level GPIO msi_ctrl_io[31:0] Note: This register field is sticky.

11.4.3.1.110 RADM clock gating enable control register. (CLOCK_GATING_CTRL_OFF)

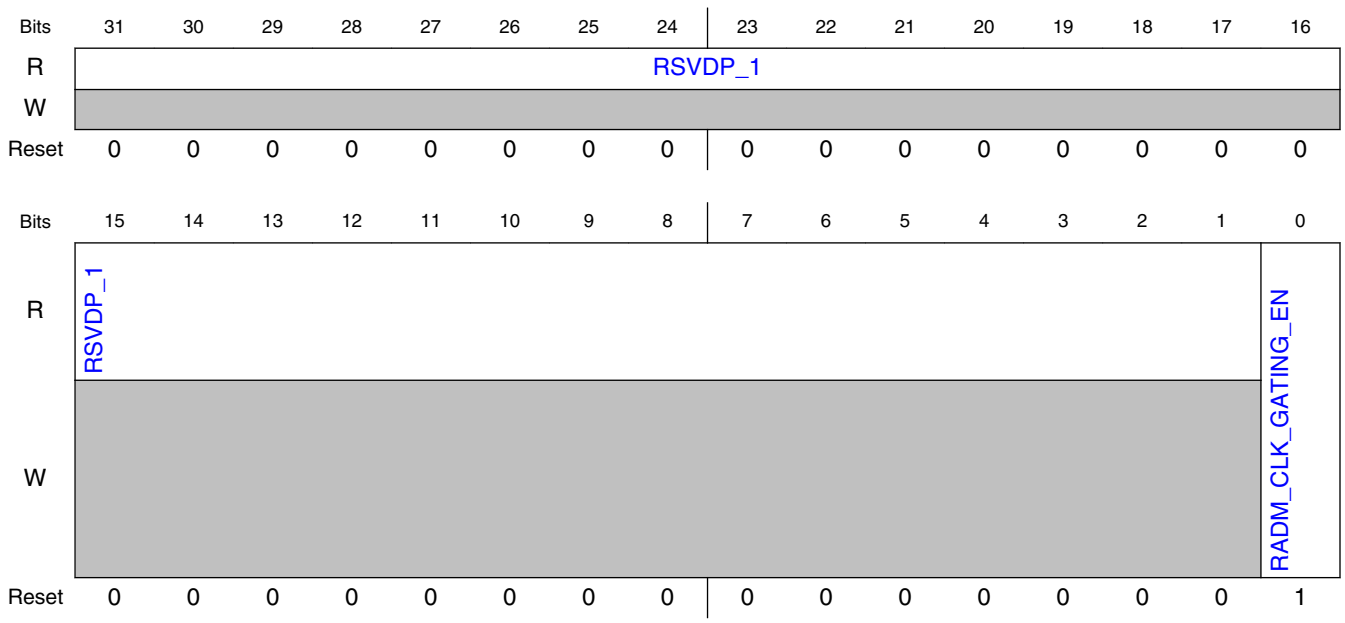
11.4.3.1.110.1 Offset

Register	Offset
CLOCK_GATING_CTRL_OFF	88Ch

11.4.3.1.110.2 Function

RADM clock gating enable control register. Using this register you can disable the RADM clock gating feature. The DWC_pcie_clk_rst.v modules uses the en_radm_clk_g output to gate core_clk and create the radm_clk_g clock input. The controller de-asserts the en_radm_clk_g output when there is no Rx traffic, Rx queues and pre/post-queue pipelines are empty, RADM completion LUT is empty, and there are no FLR actions pending. You must set the RADM_CLK_GATING_EN field to enable this functionality; otherwise the en_radm_clk_g output will always be set to '1'.

11.4.3.1.110.3 Diagram



11.4.3.1.110.4 Fields

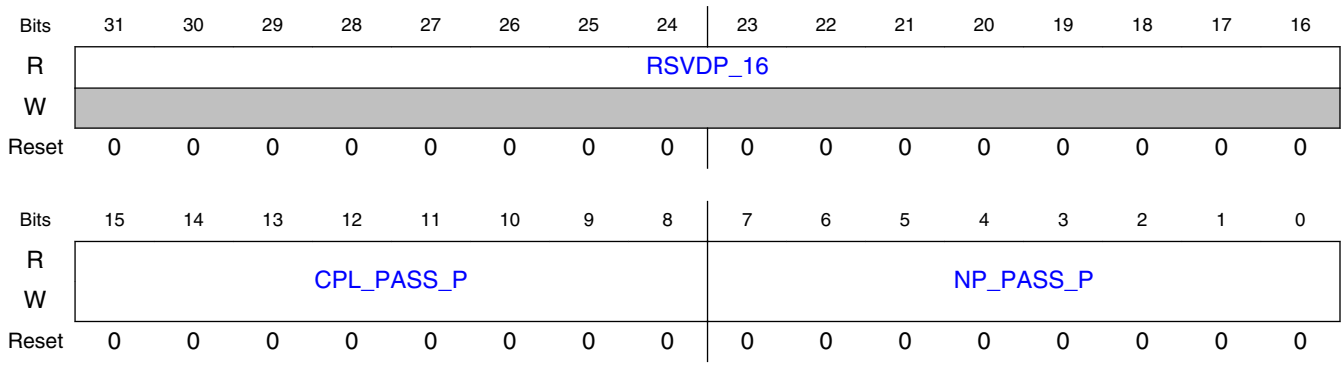
Field	Function
31-1 RSVDP_1	Reserved for future use.
0 RADM_CLK_GATING_EN	Enable Radm clock gating feature. - 0: Disable - 1: Enable(default)

11.4.3.1.111 Order Rule Control Register. (ORDER_RULE_CTRL_OFF)

11.4.3.1.111.1 Offset

Register	Offset
ORDER_RULE_CTRL_OFF	8B4h

11.4.3.1.111.2 Diagram



11.4.3.1.111.3 Fields

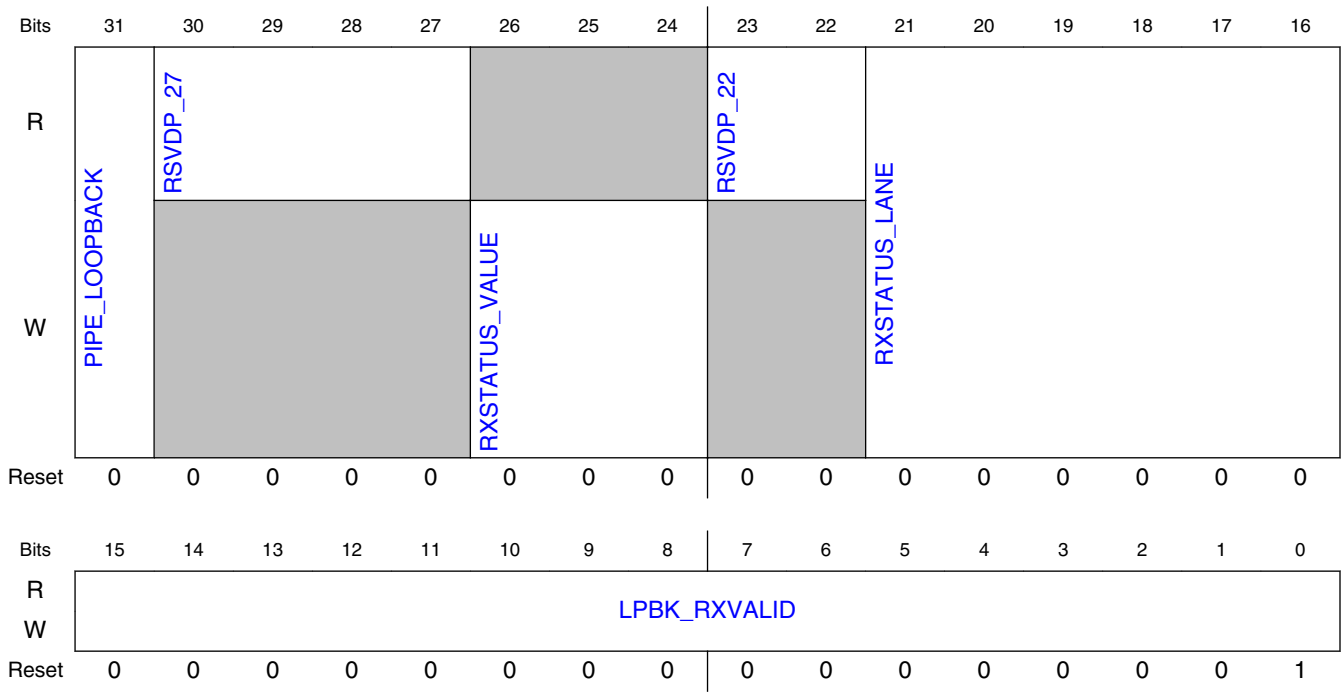
Field	Function
31-16 RSVDP_16	Reserved for future use.
15-8 CPL_PASS_P	Completion Passing Posted Ordering Rule Control. Determines if CPL can pass halted P queue. - 0: CPL can not pass P (recommended) - 1: CPL can pass P
7-0 NP_PASS_P	Non-Posted Passing Posted Ordering Rule Control. Determines if NP can pass halted P queue. - 0 : NP can not pass P (recommended). - 1 : NP can pass P

11.4.3.1.112 PIPE Loopback Control Register. (PIPE_LOOPBACK_CONTROL_OFF)

11.4.3.1.112.1 Offset

Register	Offset
PIPE_LOOPBACK_CONTROL_OFF	8B8h

11.4.3.1.112.2 Diagram



11.4.3.1.112.3 Fields

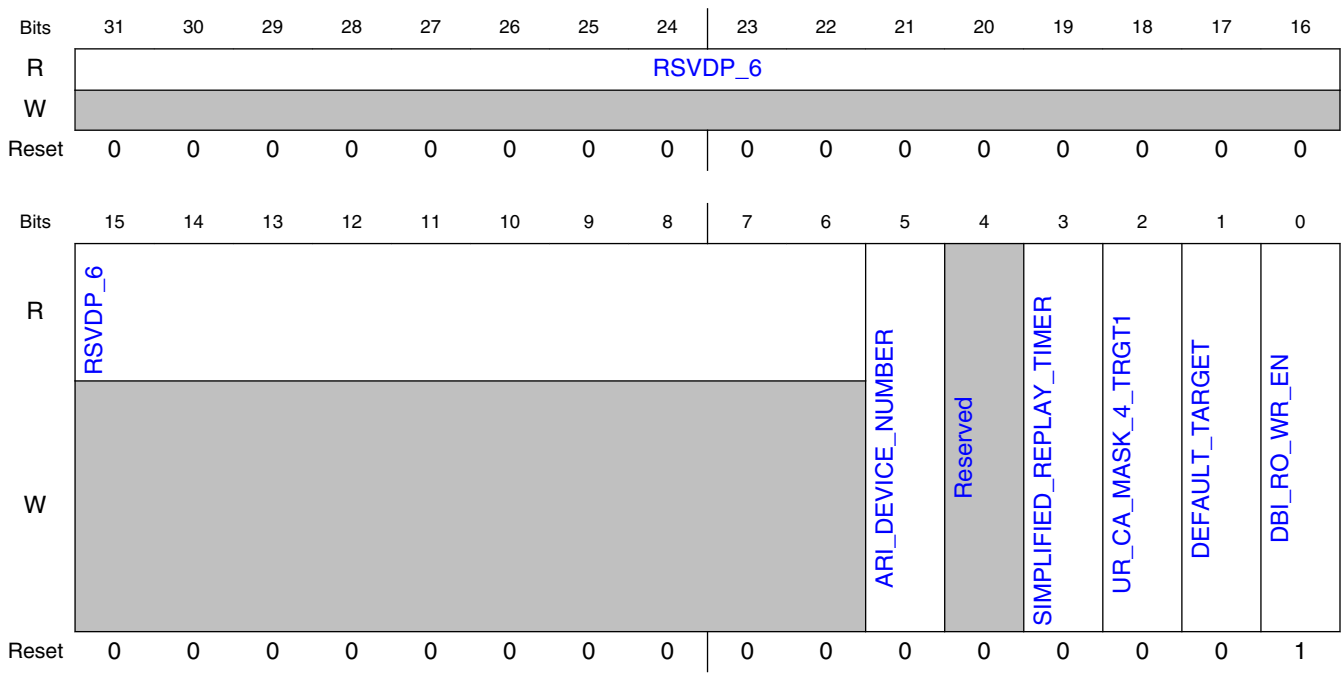
Field	Function
31 PIPE_LOOPBACK	PIPE Loopback Enable. Indicates RMMI Loopback if M-PCIe. Note: This register field is sticky.
30-27 RSVDP_27	Reserved for future use.
26-24 RXSTATUS_VALUE	RXSTATUS_VALUE is an internally reserved field. Do not use.
23-22 RSVDP_22	Reserved for future use.
21-16 RXSTATUS_LANE	RXSTATUS_LANE is an internally reserved field. Do not use. Note: This register field is sticky.
15-0 LPBK_RXVALID	LPBK_RXVALID is an internally reserved field. Do not use. Note: This register field is sticky.

11.4.3.1.113 DBI Read-Only Write Enable Register. (MISC_CONTROL_1_OFF)

11.4.3.1.113.1 Offset

Register	Offset
MISC_CONTROL_1_OFF	8BCh

11.4.3.1.113.2 Diagram



11.4.3.1.113.3 Fields

Field	Function
31-6 RSVDP_6	Reserved for future use.
5 ARI_DEVICE_NUMBER	When ARI is enabled, this field enables use of the device ID. Note: This register field is sticky.
4 —	Reserved

Table continues on the next page...

Field	Function
3 SIMPLIFIED_REPLAY_TIMER	Enables Simplified Replay Timer (Gen4). For more details, see "Transmit Replay" in the Controller Operations chapter of the Databook. Simplified Replay Timer Values are: - A value from 24,000 to 31,000 Symbol Times when Extended Synch is 0b. - A value from 80,000 to 100,000 Symbol Times when Extended Synch is 1b. Must not be changed while link is in use. Note: This register field is sticky.
2 UR_CA_MASK_4_TRGT1	This field only applies to request TLPs (with UR filtering status) that you have chosen to forward to the application (when you set DEFAULT_TARGET in this register). - When you set this field to '1', the core suppresses error logging, Error Message generation, and CPL generation (for non-posted requests). - For more details, refer to the "Advanced Error Handling For Received TLPs" chapter of the Databook. You should set this if you have set the Default Target port logic register to '1'. Default is CX_MASK_UR_CA_4_TRGT1 configuration parameter. Note: This register field is sticky.
1 DEFAULT_TARGET	Default target a received IO or MEM request with UR/CA/CRS is sent to by the controller. - 0: The controller drops all incoming I/O or MEM requests (after corresponding error reporting). A completion with UR status will be generated for non-posted requests. - 1: The controller forwards all incoming I/O or MEM requests with UR/CA/CRS status to your application For more details, see "ECRC Handling" and "Request TLP Routing Rules" in "Receive Routing" section of the "Controller Operations" chapter of the Databook. Default value is DEFAULT_TARGET configuration parameter. Note: This register field is sticky.
0 DBI_RO_WR_EN	Write to RO Registers Using DBI. When you set this field to "1", then some RO and HwInit bits are writable from the local application through the DBI. For more details, see "Writing to Read-Only Registers." Note: This register field is sticky.

11.4.3.1.114 UpConfigure Multi-lane Control Register. (MULTI_LANE_CONTROL_OFF)

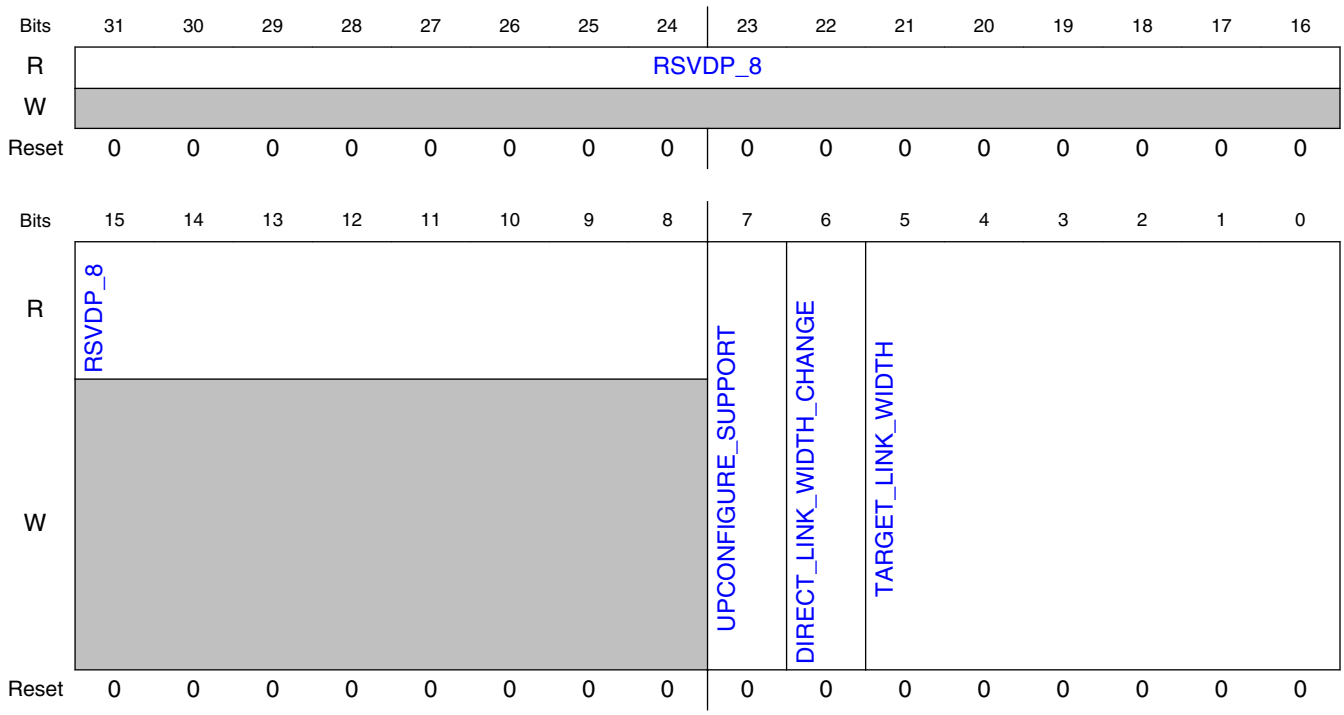
11.4.3.1.114.1 Offset

Register	Offset
MULTI_LANE_CONTROL_OFF	8C0h

11.4.3.1.114.2 Function

UpConfigure Multi-lane Control Register. Used when upsizing or downsizing the link width through Configuration state without bringing the link down. For more details, see the "Link Establishment" section in the "Controller Operations" chapter of the Databook.

11.4.3.1.114.3 Diagram



11.4.3.1.114.4 Fields

Field	Function
31-8 RSVDP_8	Reserved for future use.
7 UPCONFIGURE_SUPPORT	Upconfigure Support. The controller sends this value as the Link Upconfigure Capability in TS2 Ordered Sets in Configuration.Complete state. This field is reserved (fixed to '0') for M-PCIe. Note: This register field is sticky.
6 DIRECT_LINK_WIDTH_CHANGE	Directed Link Width Change. The controller always moves to Configuration state through Recovery state when this bit is set to '1'. - If the upconfigure_capable variable is '1' and the PCIE_CAP_HW_AUTO_WIDTH_DISABLE bit in LINK_CONTROL_LINK_STATUS_REG is '0', the controller starts upconfigure or autonomous width downsizing (to the TARGET_LINK_WIDTH value) in the Configuration state. - If TARGET_LINK_WIDTH value is 0x0, the controller does not start upconfigure or autonomous width downsizing in the Configuration state. The controller self-clears this field when the controller accepts this request. This field is reserved (fixed to '0') for M-PCIe.
5-0 TARGET_LINK_WIDTH	Target Link Width. Values correspond to: - 6'b000000: Core does not start upconfigure or autonomous width downsizing in the Configuration state. - 6'b000001: x1 - 6'b000010: x2 - 6'b000100: x4 - 6'b001000: x8 - 6'b010000: x16 - 6'b100000: x32 This field is reserved (fixed to '0') for M-PCIe.

11.4.3.1.115 PHY Interoperability Control Register. (PHY_INTEROP_CTRL_OFF)

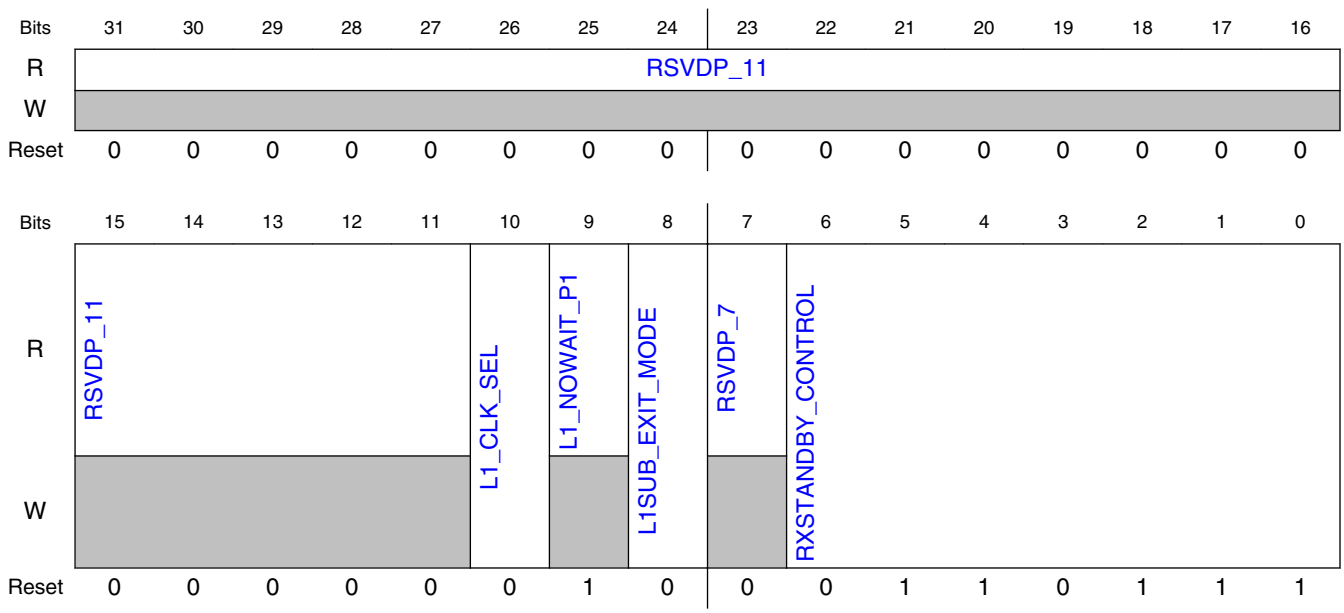
11.4.3.1.115.1 Offset

Register	Offset
PHY_INTEROP_CTRL_OFF	8C4h

11.4.3.1.115.2 Function

PHY Interoperability Control Register. This register is reserved for internal use. You should not write to this register and change the default unless specifically instructed by Synopsys support.

11.4.3.1.115.3 Diagram



11.4.3.1.115.4 Fields

Field	Function
31-11 RSVDP_11	Reserved for future use.
10 L1_CLK_SEL	L1 Clock control bit. - 1: Controller does not request aux_clk switch and core_clk gating in L1. - 0: Controller requests aux_clk switch and core_clk gating in L1. Note: This register field is sticky.
9 L1_NOWAIT_P1	L1 entry control bit. - 1: Core does not wait for PHY to acknowledge transition to P1 before entering L1. - 0: Core waits for the PHY to acknowledge transition to P1 before entering L1. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.

Table continues on the next page...

Field	Function
8 L1SUB_EXIT_MODE	L1 Exit Control Using phy_mac_pclkack_n. - 1: Core exits L1 without waiting for the PHY to assert phy_mac_pclkack_n. - 0: Core waits for the PHY to assert phy_mac_pclkack_n before exiting L1. Note: This register field is sticky.
7 RSVDP_7	Reserved for future use.
6-0 RXSTANDBY_CONTROL	Rxstandby Control. Bits 0..5 determine if the controller asserts the RxStandby signal (mac_phy_rxstandby) in the indicated condition. Bit 6 enables the controller to perform the RxStandby/ RxStandbyStatus handshake. - [0]: Rx EIOS and subsequent T TX-IDLE-MIN - [1]: Rate Change - [2]: Inactive lane for upconfigure/downconfigure - [3]: PowerDown=P1orP2 - [4]: RxL0s.Idle - [5]: EI Infer in L0 - [6]: Execute RxStandby/RxStandbyStatus Handshake This field is reserved (fixed to '0') for M-PCIe. Note: This register field is sticky.

11.4.3.1.116 TRGT_CPL_LUT Delete Entry Control register. (TRGT_CPL_LUT_DELETE_ENTRY_OFF)

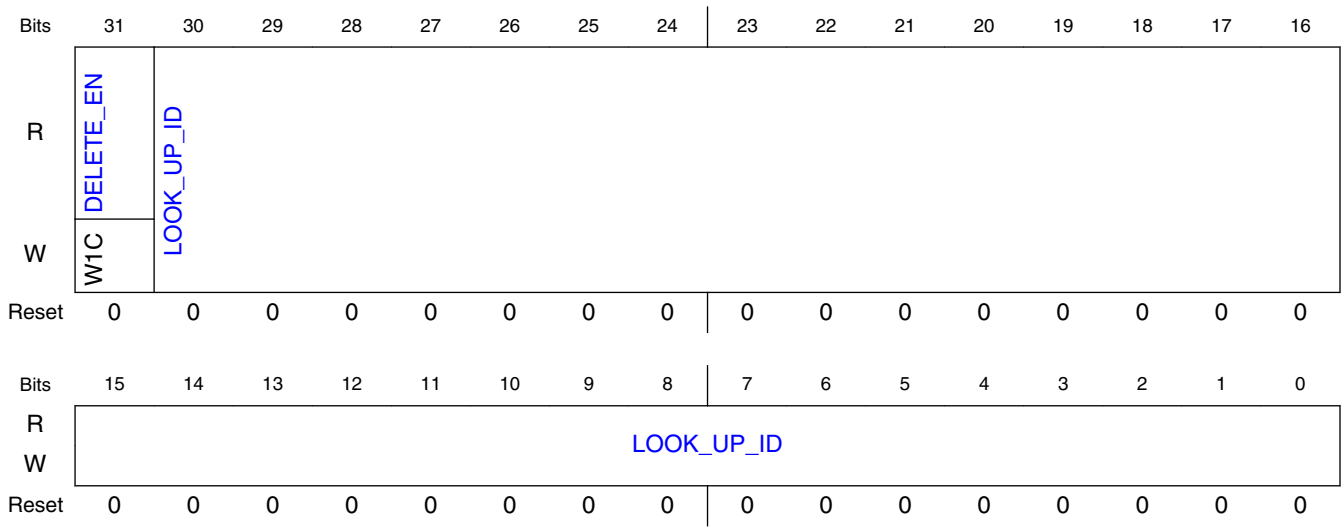
11.4.3.1.116.1 Offset

Register	Offset
TRGT_CPL_LUT_DELETE_ENTRY_OFF	8C8h

11.4.3.1.116.2 Function

TRGT_CPL_LUT Delete Entry Control register. Using this register you can delete one entry in the target completion LUT. You should only use this register when you know that your application will never send the completion because of an FLR or any other reason. Note:: The target completion LUT (and associated target completion timeout event) is watching for completions (from your application on XALI0/1/2 or AXI master read channel) corresponding to previously received non-posted requests from the PCIe wire.

11.4.3.1.116.3 Diagram



11.4.3.1.116.4 Fields

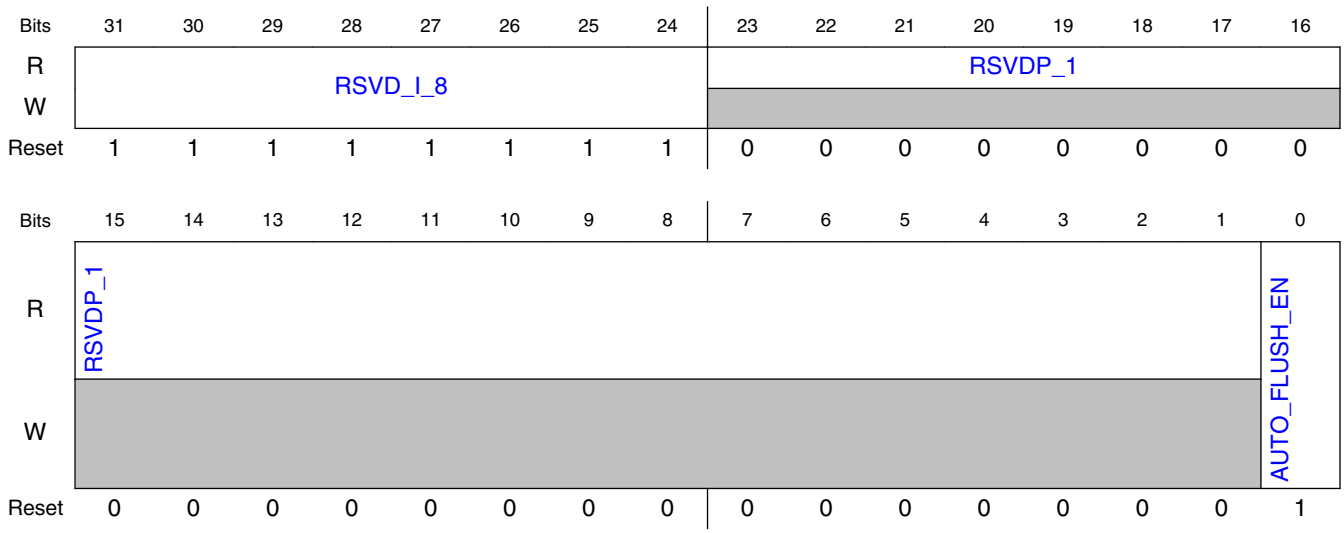
Field	Function
31 DELETE_EN	This is a one-shot bit. A '1' write to this bit triggers the deletion of the target completion LUT entry that is specified in the LOOK_UP_ID field. This is a self-clearing register field. Reading from this register field always returns a '0'.
30-0 LOOK_UP_ID	This number selects one entry to delete of the TRGT_CPL_LUT.

11.4.3.1.117 Link Reset Request Flush Control Register. (LINK_FLUSH_CONTROL_OFF)

11.4.3.1.117.1 Offset

Register	Offset
LINK_FLUSH_CONTROL_OFF	8CCh

11.4.3.1.117.2 Diagram



11.4.3.1.117.3 Fields

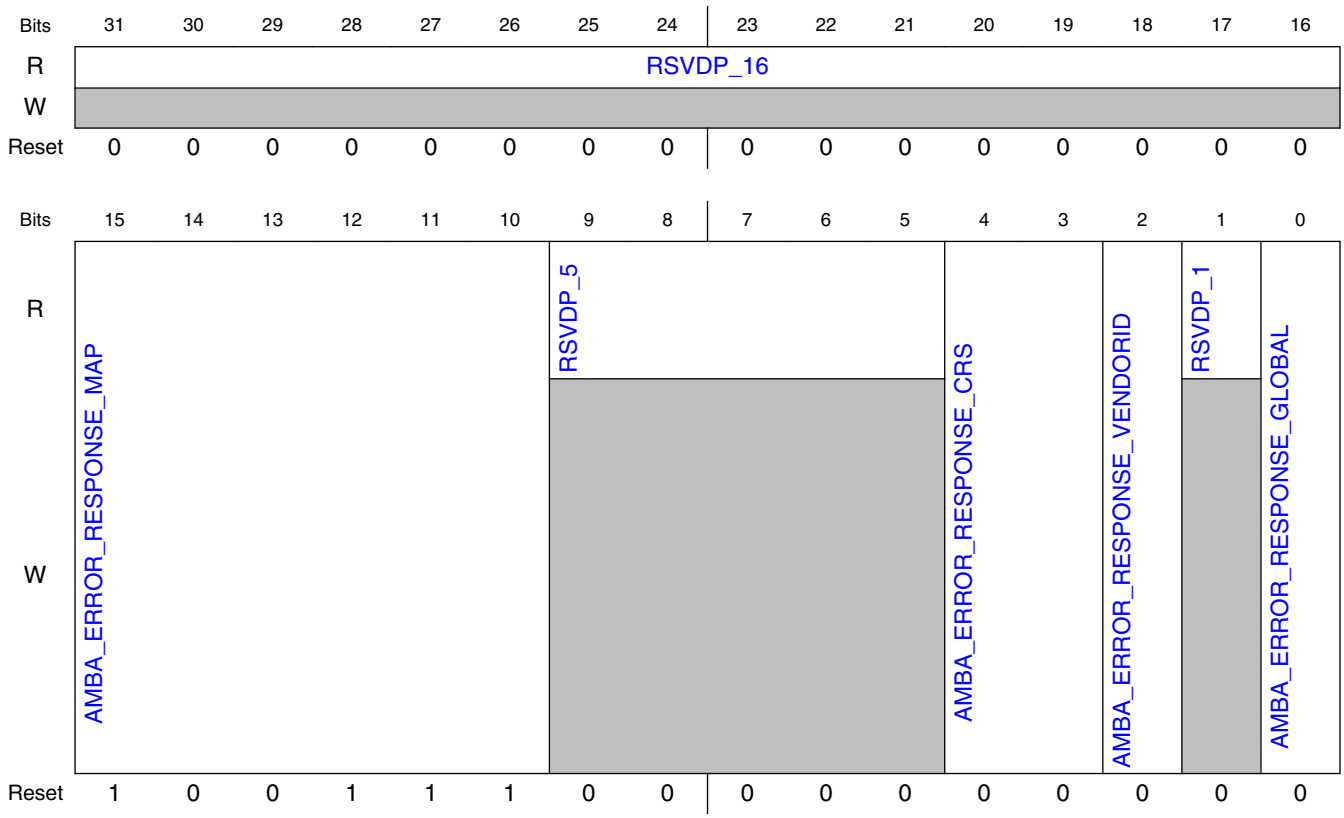
Field	Function
31-24 RSVD_I_8	This is an internally reserved field. Do not use. Note: This register field is sticky.
23-1 RSVDP_1	Reserved for future use.
0 AUTO_FLUSH_EN	Enables automatic flushing of pending requests before sending the reset request to the application logic to reset the PCIe controller and the AXI Bridge. The flushing process is initiated if any of the following events occur: - Hot reset request. A downstream port (DSP) can "hot reset" an upstream port (USP) by sending two consecutive TS1 ordered sets with the hot reset bit asserted. - Warm (Soft) reset request. Generated when exiting from D3 to D0 and cfg_pm_no_soft_rst=0. - Link down reset request. A high to low transition on smlh_req_rst_not indicates the link has gone down and the controller is requesting a reset. If you disable automatic flushing, your application is responsible for resetting the PCIe controller and the AXI Bridge. For more details see "Warm and Hot Resets" section in the Architecture chapter of the Databook. Note: This register field is sticky.

11.4.3.1.118 AXI Bridge Slave Error Response Register. (AMBA_ERROR_RESPONSE_DEFAULT_OFF)

11.4.3.1.118.1 Offset

Register	Offset
AMBA_ERROR_RESPONSE_DEFAULT_OFF	8D0h

11.4.3.1.118.2 Diagram



11.4.3.1.118.3 Fields

Field	Function
31-16 RSVDP_16	Reserved for future use.
15-10 AMBA_ERROR_RESPONSE_MAP	AXI Slave Response Error Map. Allows you to selectively map the errors received from the PCIe completion (for non-posted requests) to the AXI slave responses, slv_rresp or slv_bresp. The recommended setting is SLVERR. CRS is always mapped to OKAY. - [0] - 0: UR (unsupported request) -> DECERR - 1: UR (unsupported request) -> SLVERR - [1] - 0: CRS (configuration retry status) -> DECERR - 1: CRS (configuration retry status) -> SLVERR - [2] - 0: CA (completer abort) -> DECERR - 1: CA (completer abort) -> SLVERR - [3]: Reserved - [4]: Reserved - [5]: - 0: Completion Timeout -> DECERR - 1: Completion Timeout -> SLVERR The AXI bridge internally drops (processes internally but not passed to your application) a completion that has been marked by the Rx filter as UC or MLF, and does not pass its status directly down to the slave interface. It waits for a timeout and then signals "Completion Timeout" to the slave interface. The controller sets the AXI slave read databus to 0xFFFF for all error responses. Note: This register field is sticky.
9-5 RSVDP_5	Reserved for future use.
4-3	CRS Slave Error Response Mapping. Determines the AXI slave response for CRS completions. For more details see "Error Handling" in the AXI chapter of the Databook. AHB: - always returns OKAY AXI: - 00: OKAY - 01: OKAY with all FFFF_FFFF data for all CRS completions - 10: OKAY with FFFF_0001 data for

Table continues on the next page...

Field	Function
AMBA_ERROR_RESPONSE_CRS	CRS completions to vendor ID read requests, OKAY with FFFF_FFFF data for all other CRS completions - 11: SLVERR/DECERR (the AXI_ERROR_RESPONSE_MAP field determines the PCIe-to-AXI Slave error response mapping) Note: This register field is sticky.
2 AMBA_ERROR_RESPONSE_VENDORID	Vendor ID Non-existent Slave Error Response Mapping. Determines the AXI slave response for errors on reads to non-existent Vendor ID register. For more details see "Error Handling" in the AXI chapter of the Databook. AHB: - 0: OKAY (with FFFF data). The controller ignores the setting in the bit when bit 0 of this register is '0'. - 1: ERROR AXI: - 0: OKAY (with FFFF data). - 1: SLVERR/DECERR (the AXI_ERROR_RESPONSE_MAP field determines the PCIe-to-AXI Slave error response mapping) Note: This register field is sticky.
1 RSVDP_1	Reserved for future use.
0 AMBA_ERROR_RESPONSE_GLOBAL	Global Slave Error Response Mapping. Determines the AXI slave response for all error scenarios on non-posted requests. For more details see "Error Handling" in the AXI chapter of the Databook. AHB: - 0: OKAY (with FFFF data for non-posted requests) and ignore the setting in bit [2] of this register. - 1: ERROR for normal link (data) accesses and look at bit [2] for other scenarios. AXI: - 0: OKAY (with FFFF data for non-posted requests) - 1: SLVERR/DECERR (the AXI_ERROR_RESPONSE_MAP field determines the PCIe-to-AXI Slave error response mapping) The error response mapping is not applicable to Non-existent Vendor ID register reads. Note: This register field is sticky.

11.4.3.1.119 Link Down AXI Bridge Slave Timeout Register. (AMBA_LINK_TIMEOUT_OFF)

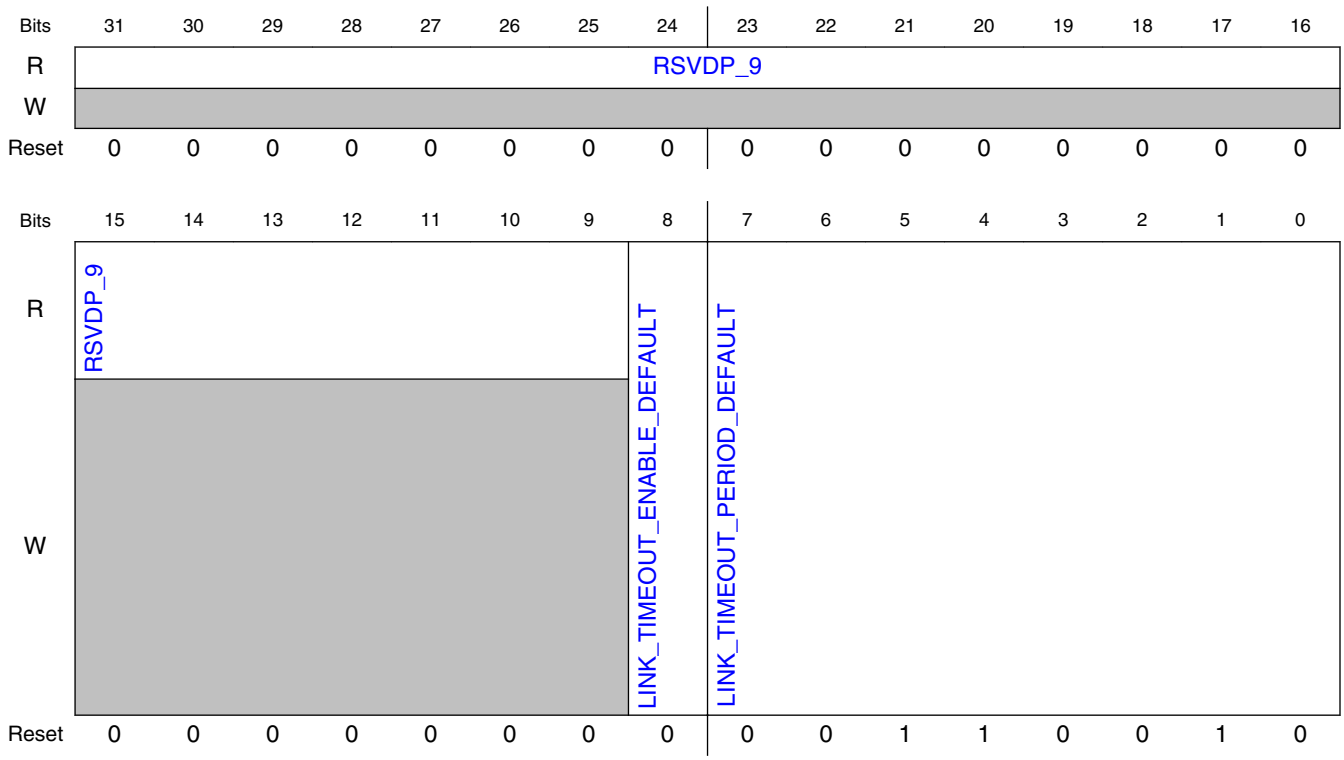
11.4.3.1.119.1 Offset

Register	Offset
AMBA_LINK_TIMEOUT_OFF	8D4h

11.4.3.1.119.2 Function

Link Down AXI Bridge Slave Timeout Register. If your application AXI master issues outbound requests to the AXI bridge slave interface before the PCIe link is operational, the controller starts a "flush" timer. The timeout value of the timer is set by this register. The timer will timeout and then flush the bridge TX request queues after this amount of time. The timer counts when there are pending outbound AXI slave interface (or DMA) requests and the PCIe TX link is not transmitting any of these requests. For more details, see the "AXI Bridge Initialization, Clocking and Reset" section in the AXI chapter of the Databook.

11.4.3.1.119.3 Diagram



11.4.3.1.119.4 Fields

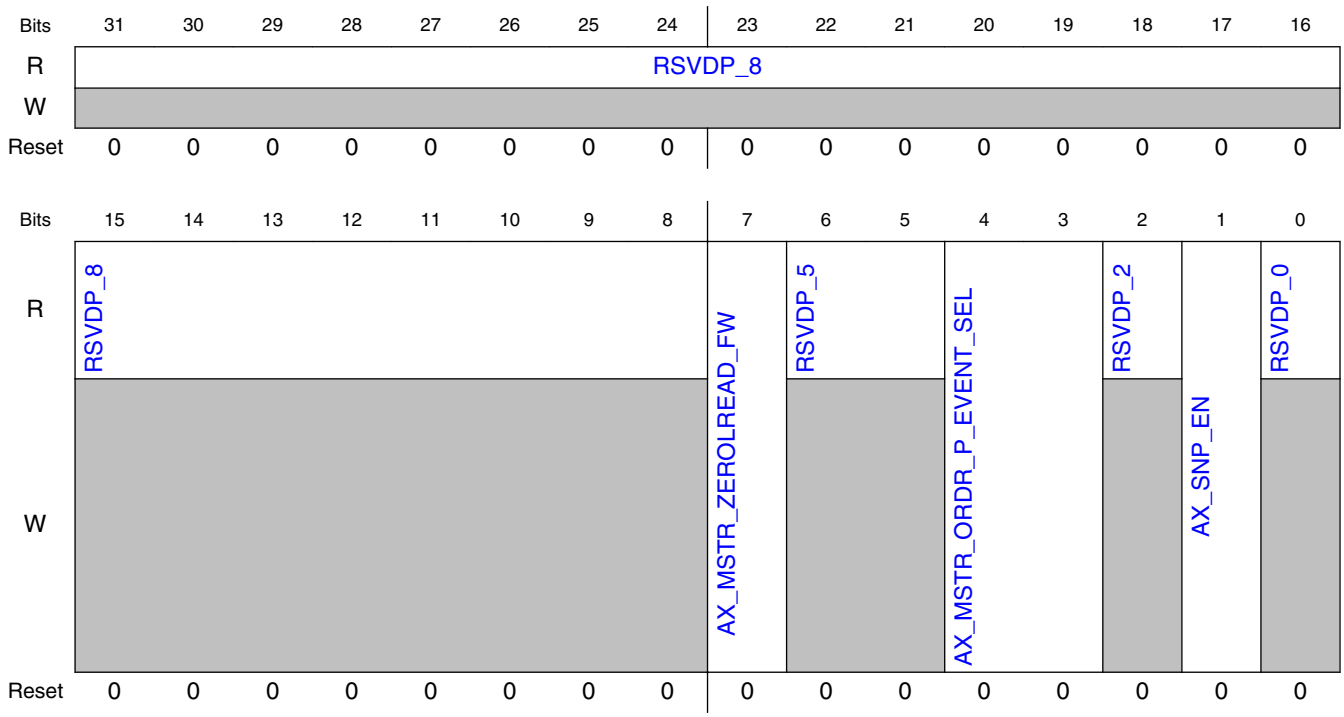
Field	Function
31-9 RSVDP_9	Reserved for future use.
8 LINK_TIMEOUT_ENABLE_DEFAULT	Disable Flush. You can disable the flush feature by setting this field to "1". Note: This register field is sticky.
7-0 LINK_TIMEOUT_PERIOD_DEFAULT	Timeout Value (ms). The timer will timeout and then flush the bridge TX request queues after this amount of time. The timer counts when there are pending outbound AXI slave interface requests and the PCIe TX link is not transmitting any of these requests. The timer is clocked by core_clk. For an M-PCIe configuration: - Time unit of this field is 4 ms. - Margin of error for RateA clock is < 1%. - Margin of error for RateB clock is between 16% and 17%. Note: This register field is sticky.

11.4.3.1.120 AMBA Ordering Control. (AMBA_ORDERING_CTRL_OFF)

11.4.3.1.120.1 Offset

Register	Offset
AMBA_ORDERING_CTRL_OFF	8D8h

11.4.3.1.120.2 Diagram



11.4.3.1.120.3 Fields

Field	Function
31-8 RSVDP_8	Reserved for future use.
7 AX_MSTR_ZEROLREAD_FW	AXI Master Zero Length Read Forward to the application. The DW PCIe controller AXI bridge is able to terminate in order with the Posted transactions the zero length read, implementing the PCIe express flush semantics of the Posted transactions. - 0x0: The zero length Read is terminated at the DW PCIe AXI bridge master - 0x1: The zero length Read is forward to the application.
6-5 RSVDP_5	Reserved for future use.
4-3 AX_MSTR_ORDR_P_EVENT_SEL	AXI Master Posted Ordering Event Selector. This field selects how the master interface determines when a P write is completed when enforcing the PCIe ordering rule, "NP must not pass P" at the AXI Master Interface. The AXI protocol does not support ordering between channels. Therefore, NP reads can pass P on your AXI bus fabric. This can result in an ordering violation when the read overtakes a P that is

Table continues on the next page...

Field	Function
	going to the same address. Therefore, the bridge master does not issue any NP requests until all outstanding P writes reach their destination. It does this by waiting for the all of the write responses on the B channel. This can affect the performance of the master read channel. For scenarios where the interconnect serializes the AXI master "AW", "W" and "AR" channels,you can increase the performance by reducing the need to wait until the complete Posted transaction has effectively reached the application slave. - 00: B'last event: wait for the all of the write responses on the B channel thereby ensuring that the complete Posted transaction has effectively reached the application slave (default). - 01: AW'last event: wait until the complete Posted transaction has left the AXI address channel at the bridge master. - 10: W'last event: wait until the complete Posted transaction has left the AXI data channel at the bridge master. - 11: Reserved Note 2: This setting will not affect: - MSI interrupt catcher and P data ordering. This is always driven by the B'last event. - DMA read engine TLP ordering. This is always driven by the B'last event. - NP write transactions which are always serialized with P write transactions.
2 RSVDP_2	Reserved for future use.
1 AX_SNP_EN	AXI Serialize Non-Posted Requests Enable. This field enables the AXI Bridge to serialize same ID Non-Posted Read/Write Requests on the wire. Serialization implies one outstanding same ID NP Read or Write on the wire and used to avoid AXI RAR and WAW hazards at the remote link partner. For more details, see the "Optional Serialization of AXI Slave Non-posted Requests" section in the AXI chapter of the Databook.
0 RSVDP_0	Reserved for future use.

11.4.3.1.121 Lower 20 bits of the programmable AXI address where Messages coming from wire are mapped to. (AXI_MSTR_MSG_ADDR_LOW_OFF)

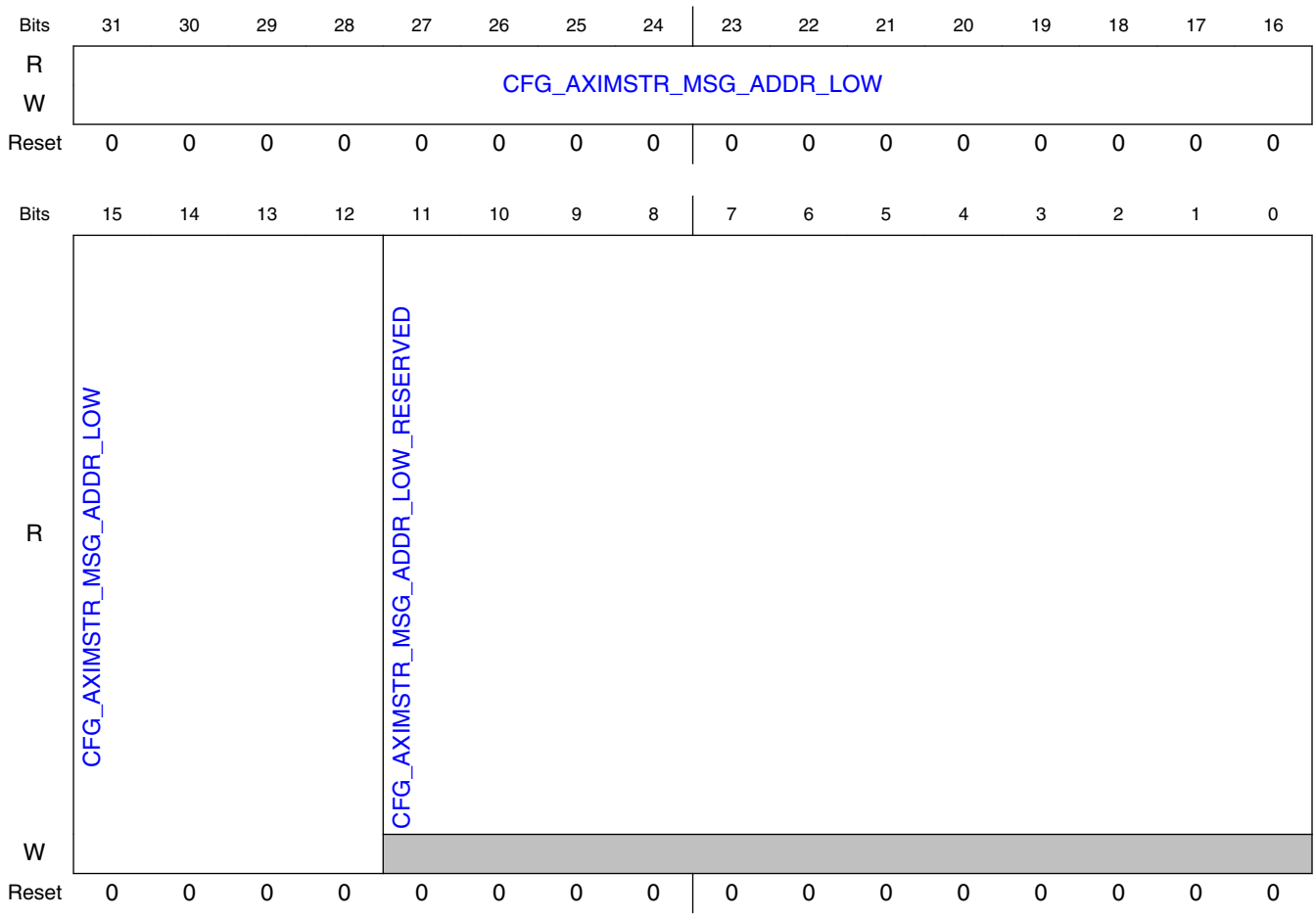
11.4.3.1.121.1 Offset

Register	Offset
AXI_MSTR_MSG_ADDR_LOW_OFF	8F0h

11.4.3.1.121.2 Function

Lower 20 bits of the programmable AXI address where Messages coming from wire are mapped to. Bits [11:0] of the register are tied to zero for the address to be 4k-aligned. In previous releases, the third and fourth DWORDs of a message (Msg/MsgD) TLP header were delivered though the AXI master address bus (mstr_awaddr). These DWORDS are now supplied through the mstr_awmisc_info_hdr_34dw[63:0] output; and the value on mstr_awaddr is driven to the value you have programmed into the AXI_MSTR_MSG_ADDR_LOW_OFF and AXI_MSTR_MSG_ADDR_HIGH_OFF registers.

11.4.3.1.121.3 Diagram



11.4.3.1.121.4 Fields

Field	Function
31-12 CFG_AXIMSTR_MSG_ADDR_LOW	Lower 20 bits of the programmable AXI address for Messages. Note: This register field is sticky.
11-0 CFG_AXIMSTR_MSG_ADDR_LOW_RESERVED	Reserved for future use. Note: This register field is sticky.

11.4.3.1.122 Upper 32 bits of the programmable AXI address where Messages coming from wire are mapped to. (AXI_MSTR_MSG_ADDR_HIGH_OFF)

11.4.3.1.122.1 Offset

Register	Offset
AXI_MSTR_MSG_ADDR_HIGH_OFF	8F4h

11.4.3.1.122.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CFG_AXIMSTR_MSG_ADDR_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CFG_AXIMSTR_MSG_ADDR_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.122.3 Fields

Field	Function
31-0 CFG_AXIMSTR_MSG_ADDR_HIGH	Upper 32 bits of the programmable AXI address for Messages. Note: This register field is sticky.

11.4.3.1.123 PCIe Controller IIP Release Version Number. (PCIE_VERSION_NUMBER_OFF)

11.4.3.1.123.1 Offset

Register	Offset
PCIE_VERSION_NUMBER_OFF	8F8h

11.4.3.1.123.2 Function

PCIe Controller IIP Release Version Number. The version number is given in hex format. You should convert each pair of hex characters to ASCII to interpret. Using 4.70a (GA) as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x67612a2a which translates to ga** Using 4.70a-ea01 as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x65613031 which translates to ea01 GA is a general release available on www.designware.com EA is an early release available on a per-customer basis.

11.4.3.1.123.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VERSION_NUMBER															
W																
Reset	0	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VERSION_NUMBER															
W																
Reset	0	0	1	1	0	0	0	0	0	0	1	0	1	0	1	0

11.4.3.1.123.4 Fields

Field	Function
31-0 VERSION_NUMBER	Version Number.

11.4.3.1.124 PCIe Controller IIP Release Version Type. (PCIE_VERSION_TYPE_OFF)

11.4.3.1.124.1 Offset

Register	Offset
PCIE_VERSION_TYPE_OFF	8FCh

11.4.3.1.124.2 Function

PCIe Controller IIP Release Version Type. The type is given in hex format. You should convert each pair of hex characters to ASCII to interpret. Using 4.70a (GA) as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x67612a2a which translates to ga** Using 4.70a-ea01 as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x65613031 which translates to ea01 GA is a general release available on www.designware.com EA is an early release available on a per-customer basis.

11.4.3.1.124.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VERSION_TYPE															
W																
Reset	0	1	1	0	0	1	1	1	0	1	1	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VERSION_TYPE															
W																
Reset	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0

11.4.3.1.124.4 Fields

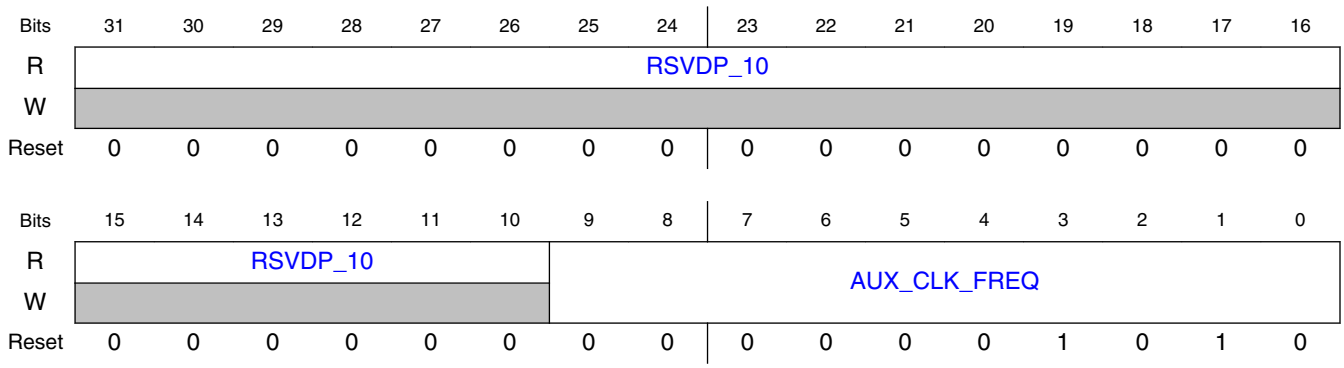
Field	Function
31-0 VERSION_TYPE	Version Type.

11.4.3.1.125 Auxiliary Clock Frequency Control Register. (AUX_CLK_FREQ_OFF)

11.4.3.1.125.1 Offset

Register	Offset
AUX_CLK_FREQ_OFF	B40h

11.4.3.1.125.2 Diagram



11.4.3.1.125.3 Fields

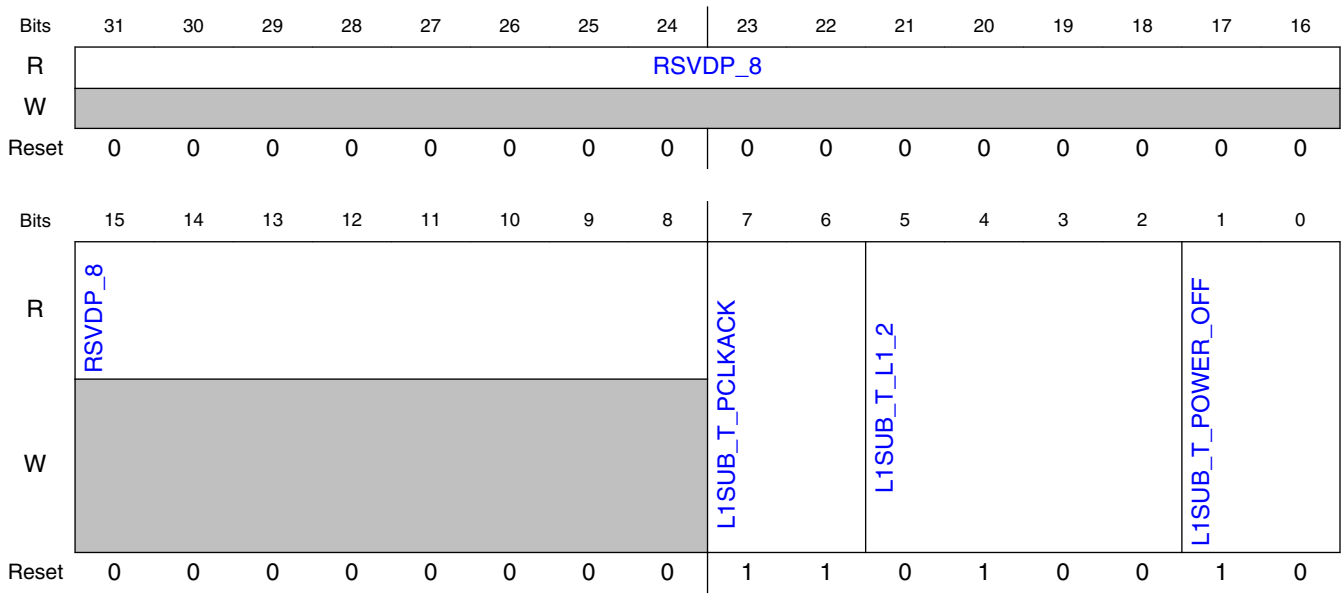
Field	Function
31-10 RSVDP_10	Reserved for future use.
9-0 AUX_CLK_FREQ	The aux_clk frequency in MHz. This value is used to provide a 1 us reference for counting time during low-power states with aux_clk when the PHY has removed the pipe_clk. Frequencies lower than 1 MHz are possible but with a loss of accuracy in the time counted. If the actual frequency (f) of aux_clk does not exactly match the programmed frequency (f_prog), then there is an error in the time counted by the controller that can be expressed in percentage as: err% = (f_prog/f-1)*100. For example if f=2.5 MHz and f_prog=3 MHz, then err% =(3/2.5-1)*100 =20%, meaning that the time counted by the controller on aux_clk will be 20% greater than the time in us programmed in the corresponding time register (for example T_POWER_ON). Note: This register field is sticky.

11.4.3.1.126 L1 Substates Timing Register. (L1_SUBSTATES_OFF)

11.4.3.1.126.1 Offset

Register	Offset
L1_SUBSTATES_OFF	B44h

11.4.3.1.126.2 Diagram



11.4.3.1.126.3 Fields

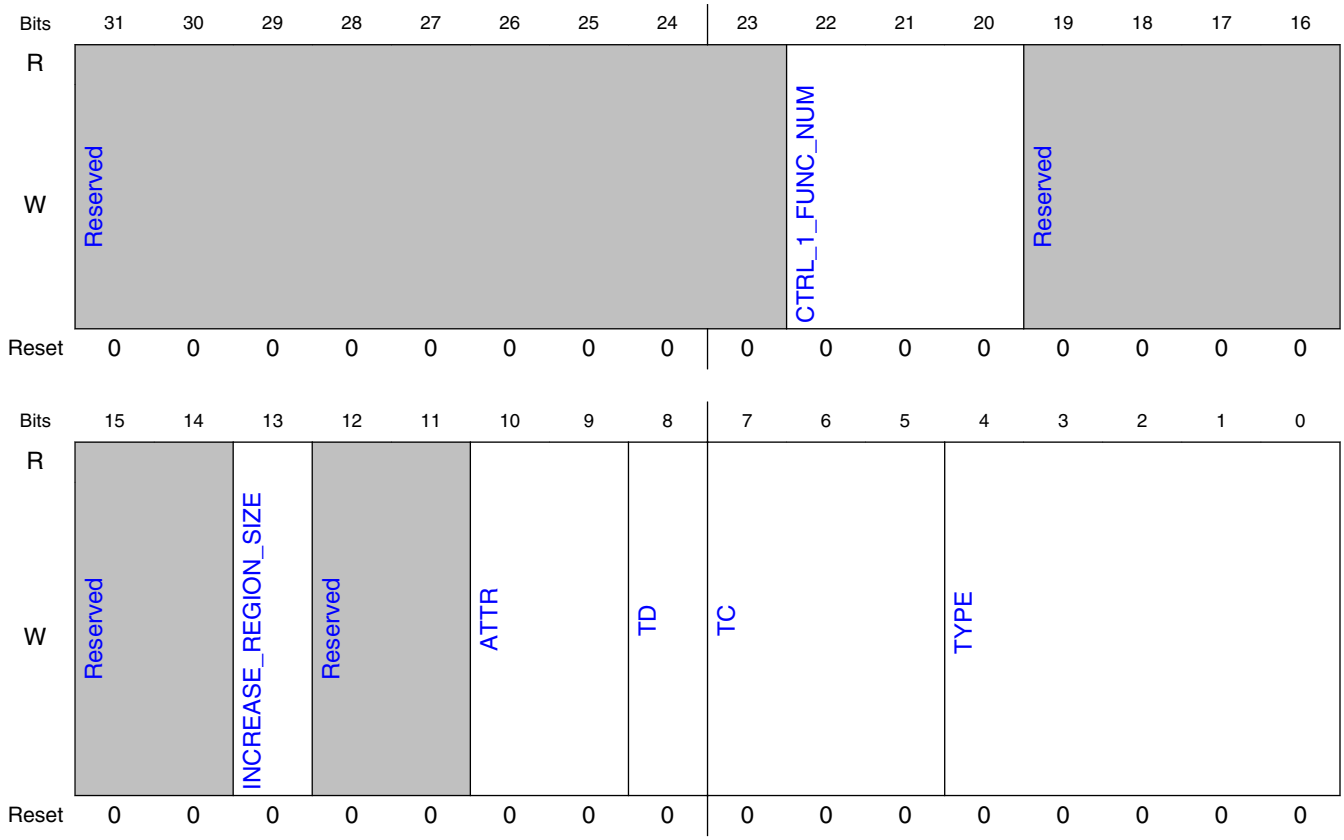
Field	Function
31-8 RSVDP_8	Reserved for future use.
7-6 L1SUB_T_PCLKACK	Max delay (in 1us units) between a MAC request to remove the clock on mac_phy_pclkreq_n and a PHY response on phy_mac_pclkack_n. If the PHY does not respond within this time the request is aborted. Range is 0..3 Note: This register field is sticky.
5-2 L1SUB_T_L1_2	Duration (in 1us units) of L1.2. Range is 0.15. Note: The timeout value can vary by 50%. Note: This register field is sticky.
1-0 L1SUB_T_POWER_OFF	Duration (in 1us units) of L1.2.Entry. Range is 0.3. Note: The timeout value can vary by 50%. Note: This register field is sticky.

11.4.3.1.127 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_0)

11.4.3.1.127.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_OUTBOUND_0	30_0000h

11.4.3.1.127.2 Diagram



11.4.3.1.127.3 Fields

Field	Function
31-23 —	Reserved
22-20 CTRL_1_FUNC_NUM	Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). - When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
19-14 —	Reserved
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by Maximum Size of iATU Region When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11	Reserved

Table continues on the next page...

Field	Function
—	
10-9 ATTR	When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8 TD	When the address of an outbound TLP is matched to this region, then the TD field of the TLP is changed to the value in this register. Note: This register field is sticky.
7-5 TC	When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4-0 TYPE	When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

11.4.3.1.128 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_0)

11.4.3.1.128.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_OUTBOUND_0	30_0004h

11.4.3.1.128.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REGION_EN	Reserved	INVERT_MODE	CFG_SHIFT_MODE	DMA_BYPASS	Reserved			HEADER_SUBSTITUTE_EN	INHIBIT_PAYLOAD	Reserved	SNP	FUNC_BYPASS	Reserved		TAG_SUBSTITUTE_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TAG								MSG_CODE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.128.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 —	Reserved
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27 DMA_BYPASS	DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
26-24 —	Reserved
23 HEADER_SUBSTITUTE_EN	Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. - 1: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register is used to fill bytes 8-to-11 (for 3 DWORD header) or bytes 12-to-15 (for 4 DWORD header) of the translated TLP header. - 0: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register forms the new address of the translated region. Note: This register field is sticky.
22 INHIBIT_PAYLOAD	Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. - 1: Fmt[1] =0 so that only TLP type without data is sent. For example, a Msg instead of MsgD will be sent. - 0: Fmt[1] =0/1 so that TLPs with or without data can be sent. Note: This register field is sticky.
21 —	Reserved
20 SNP	Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19 FUNC_BYPASS	Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18-17 —	Reserved
16 TAG_SUBSTITUTE_EN	TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note (CX_TPH_ENABLE=1): TAG substitution for MWr will not occur because this field (byte 6) in the TLP header is the ST field. ST substitution can still take place using the MSG_CODE field in IATU_REGION_CTRL_2_OFF_OUTBOUND_i. Note: This register field is sticky.

Table continues on the next page...

Field	Function
15-8 TAG	TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST;Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

11.4.3.1.129 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0)

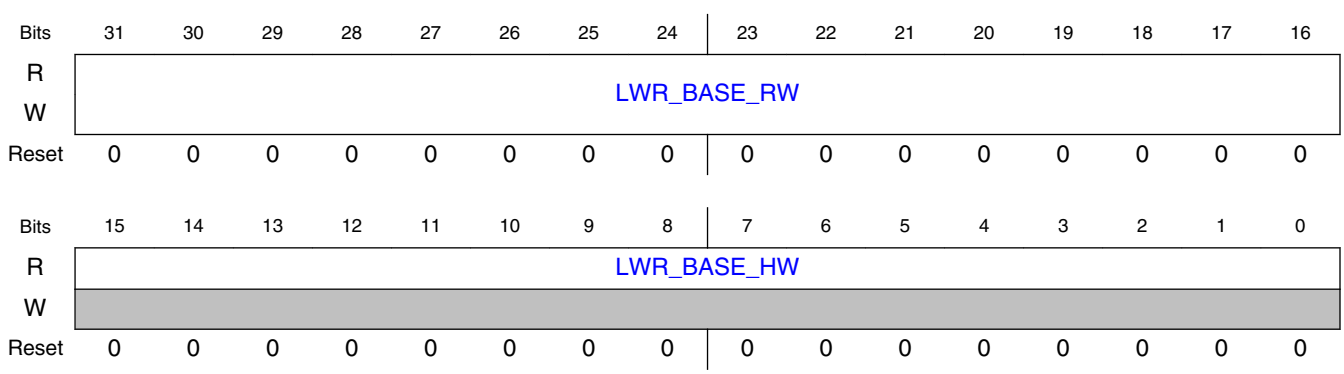
11.4.3.1.129.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0	30_0008h

11.4.3.1.129.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.4.3.1.129.3 Diagram



11.4.3.1.129.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.4.3.1.130 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0)

11.4.3.1.130.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0	30_000Ch

11.4.3.1.130.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.130.3 Fields

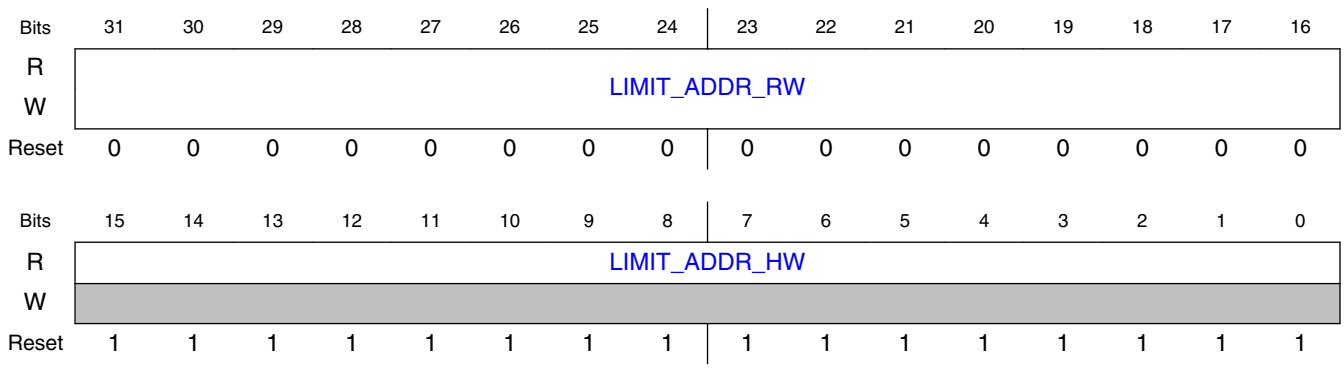
Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

11.4.3.1.131 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_0)

11.4.3.1.131.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_OUTBOUND_0	30_0010h

11.4.3.1.131.2 Diagram



11.4.3.1.131.3 Fields

Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.4.3.1.132 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0)

11.4.3.1.132.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0	30_0014h

11.4.3.1.132.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.132.3 Fields

Field	Function
31-0 LWR_TARGET_RW_OUTBOUND	When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to a Minimum Size of iATU Region kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(Minimum Size of iATU Region). When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

11.4.3.1.133 iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0)

11.4.3.1.133.1 Offset

Register	Offset
IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0	30_0018h

11.4.3.1.133.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.133.3 Fields

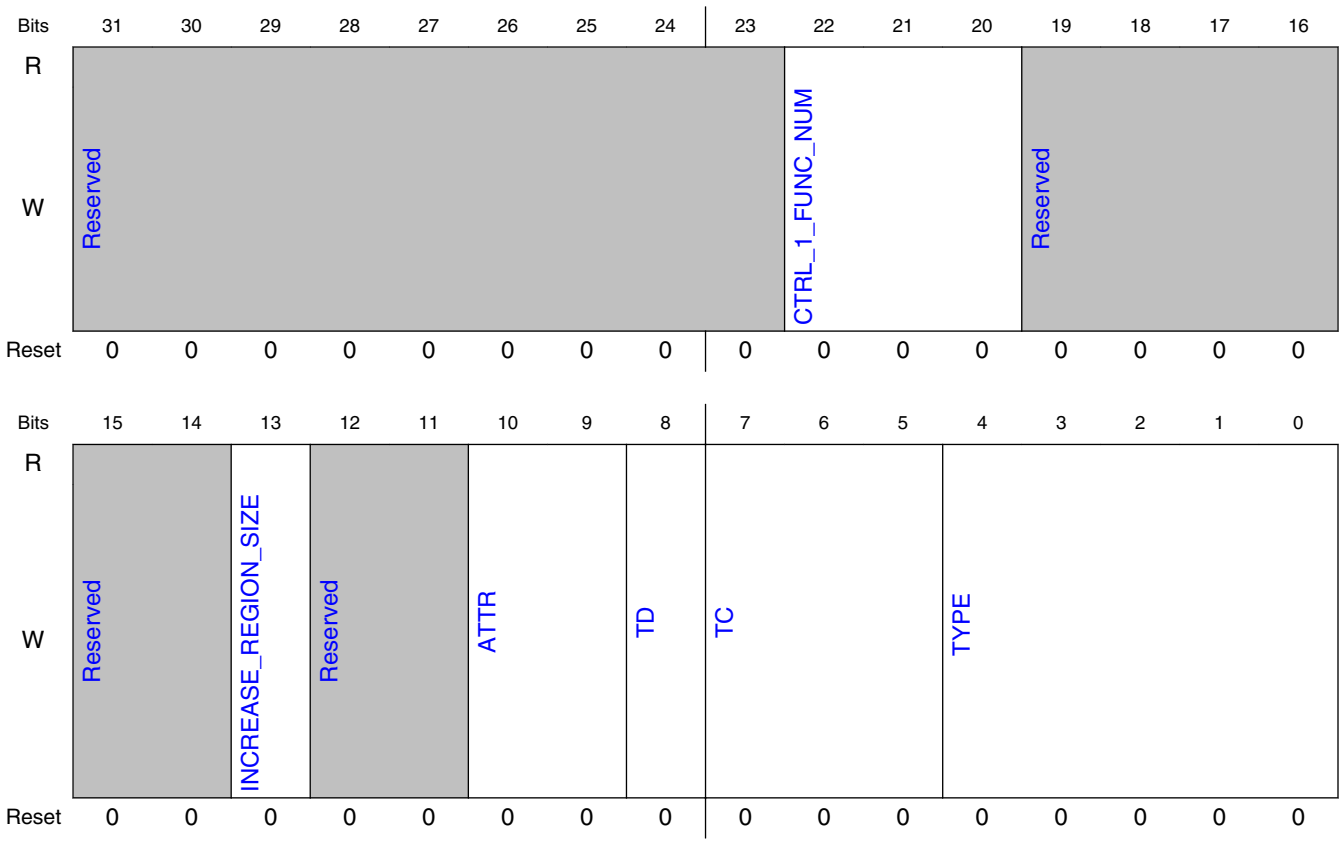
Field	Function
31-0 UPPER_TARGET_RW	Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

11.4.3.1.134 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_0)

11.4.3.1.134.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_INBOUND_0	30_0100h

11.4.3.1.134.2 Diagram



11.4.3.1.134.3 Fields

Field	Function
31-23 —	Reserved
22-20 CTRL_1_FUNC_NUM	Function Number. - MEM-I/O: When the Address and BAR matching logic in the controller indicate that a MEM-I/O transaction matches a BAR in the function corresponding to this value, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. - CFG0/CFG1: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
19-14 —	Reserved
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by Maximum Size of iATU Region When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved

Table continues on the next page...

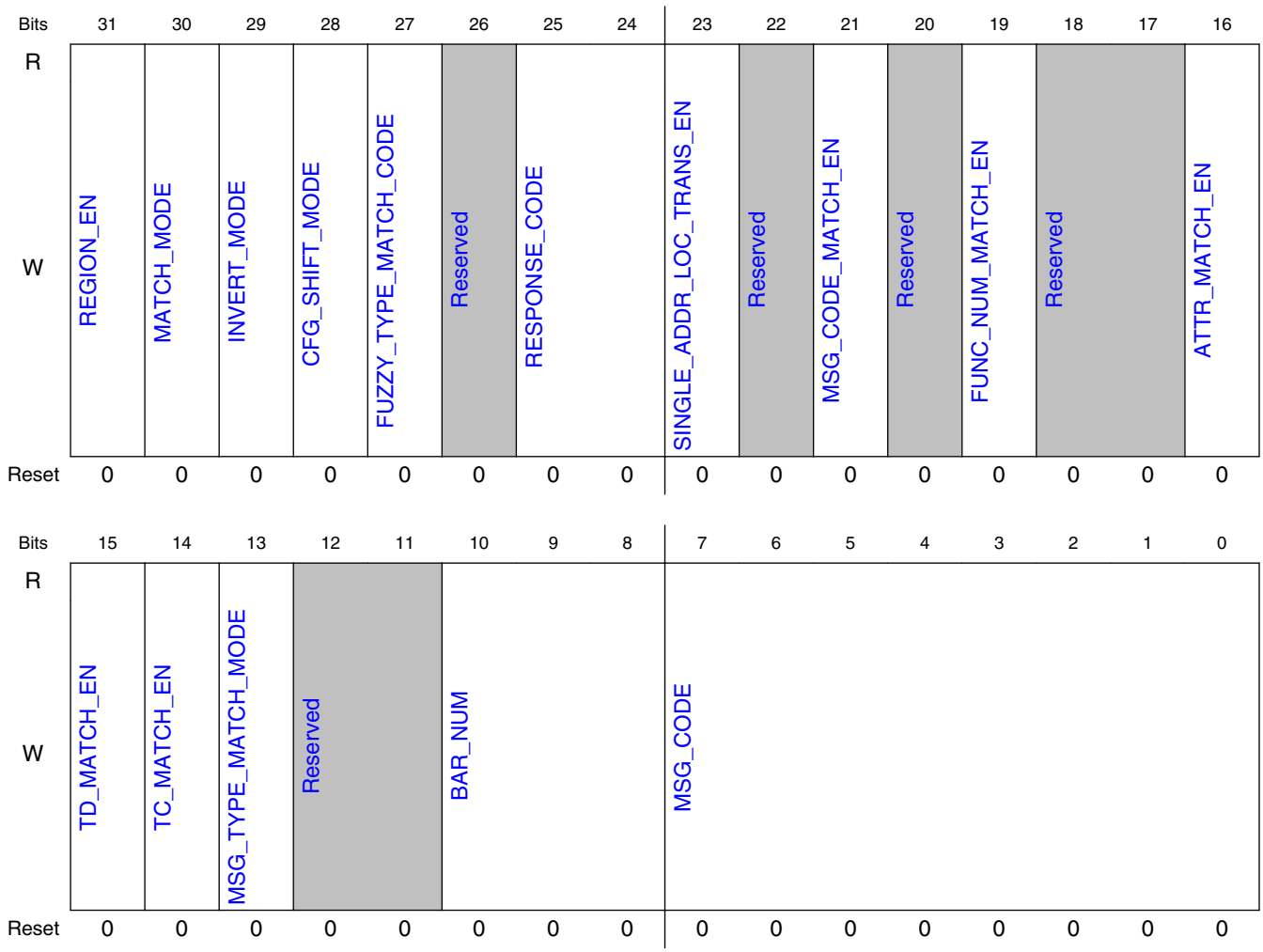
Field	Function
10-9 ATTR	When the ATTR field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "ATTR Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
8 TD	When the TD field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TD Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
7-5 TC	When the TC field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TC Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
4-0 TYPE	When the TYPE field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). Note: This register field is sticky.

11.4.3.1.135 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_0)

11.4.3.1.135.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_INBOUND_0	30_0104h

11.4.3.1.135.2 Diagram



11.4.3.1.135.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 MATCH_MODE	Match Mode. Determines Inbound matching mode for TLPs. The mode depends on the type of TLP that is received as follows: For MEM-I/O TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU operates using addresses as in the outbound direction. The Region Base and Limit Registers must be setup. - 1:BAR Match Mode. BAR matching is used. The "BAR Number" field is relevant. Not used for RC. For CFG0 TLPs, this field is interpreted as follows: - 0: Routing ID Match Mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM-I/O transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. - 1: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. For MSG/MSGD TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU treats the third dword and fourth dword of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers. - 1: Vendor ID

Table continues on the next page...

Field	Function
	Match Mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the third dword of the TLP header, but matches against the Vendor ID in bits [15:0] of the third dword of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the fourth dword of the TLP header. - If SINGLE_ADDRESS_LOCATION_TRANSLATE_EN = 1 AND MSG_TYPE_MATCH_MODE = 1, then Match Mode is ignored. Note: This register field is sticky.
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of your application memory space using a 28-bit effective address. Shifts bits [31:16] of the untranslated address to form bits [27:12] of the translated address. Note: This register field is sticky.
27 FUZZY_TYPE_MATCH_CODE	Fuzzy Type Match Mode. When enabled, the iATU relaxes the matching of the TLP TYPE field against the expected TYPE field so that - CfgRd0 and CfgRd1 TLPs are seen as identical. Similarly with CfgWr0 and CfgWr1. - MWr, MRd and MRdLk TLPs are seen as identical - The Routing field of Msg/MsgD TLPs is ignored - FetchAdd, Swap and CAS are seen as identical. For example, CFG0 in the TYPE field in the "iATU Control 1 Register" matches against an inbound CfgRd0, CfgRd1, CfgWr0 or CfgWr1 TLP. Note: This register field is sticky.
26 —	Reserved
25-24 RESPONSE_CODE	Response Code. Defines the type of response to give for accesses matching this region. This overrides the normal RADM filter response. Note that this feature is not available for any region where Single Address Location Translate is enabled. - 00 - Normal RADM filter response is used. - 01 - Unsupported request (UR) - 10 - Completer abort (CA) - 11 - Not used / undefined / reserved. Note: This register field is sticky.
23 SINGLE_ADDR_LOC_TRANS_EN	Single Address Location Translate Enable. When enabled, Rx TLPs can be translated to a single address location as determined by the target address register of the iATU region. The main usage scenario is translation of Messages (such as Vendor Defined or ATS Messages) to MWr TLPs when the AXI bridge is enabled. Note: This register field is sticky.
22 —	Reserved
21 MSG_CODE_MATCH_EN	Message Code Match Enable (Msg TLPS). Ensures that a successful message Code TLP field comparison match (see Message Code field of the "iATU Control 2 Register") occurs (in MSG transactions) for address translation to proceed. ST Match Enable (Mem TLPs). Ensures that a successful ST TLP field comparison match (see ST field of the "iATU Control 2 Register") occurs (in MEM transactions) for address translation to proceed. Only Valid when the CX_TPH_ENABLE configuration parameter is 1 Note: This register field is sticky.
20 —	Reserved
19 FUNC_NUM_MATCH_EN	Function Number Match Enable. Ensures that a successful Function Number TLP field comparison match (see Function Number field of the "iATU Control 1 Register") occurs (in MEM-I/O and CFG0/CFG1 transactions) for address translation to proceed. Note: This register field is sticky.
18-17 —	Reserved
16	ATTR Match Enable. Ensures that a successful ATTR TLP field comparison match (see ATTR field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.

Table continues on the next page...

Field	Function
ATTR_MATCH_EN	
15 TD_MATCH_EN	TD Match Enable. Ensures that a successful TD TLP field comparison match (see TD field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
14 TC_MATCH_EN	TC Match Enable. Ensures that a successful TC TLP field comparison match (see TC field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
13 MSG_TYPE_MATCH_MODE	Message Type Match Mode. When enabled, and if single address location translate enable is set, then inbound TLPs of type MSG/MSGd which match the type field of the iatu_region_ctrl_1_OFF_inbound register (=>TYPE[4:3]=2'b10) will be translated. Message type match mode overrides any value of MATCH_MODE field in this register. Usage scenarios for this are translation of VDM or ATS messages when AXI bridge is configured on client interface. Note: This register field is sticky.
12-11 —	Reserved
10-8 BAR_NUM	BAR Number. When the BAR number of an inbound MEM or IO TLP " that is matched by the normal internal BAR address matching mechanism " is the same as this field, address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Match Mode" bit of the "iATU Control 2 Register" is set. - 000b - BAR0 - 001b - BAR1 - 010b - BAR2 - 011b - BAR3 - 100b - BAR4 - 101b - BAR5 - 110b - ROM - 111b - reserved - IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b - 101b and that BAR configured as an IO BAR. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs: (Message Code). When the TYPE field of an inbound Msg/MsgD TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Message Code Match Enable" bit of the "iATU Control 2 Register" is set. Memory TLPs: (ST;Steering Tag). When the ST field of an inbound TLP is matched to this value, then address translation proceeds. This check is only performed if the "ST Match Enable" bit of the "iATU Control2 Register" is set. The setting is independent of the setting of the TH field. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

11.4.3.1.136 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_0)

11.4.3.1.136.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_INBOUND_0	30_0108h

11.4.3.1.136.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned

on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.4.3.1.136.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_BASE_HW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.136.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.4.3.1.137 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_0)

11.4.3.1.137.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_INBOUND_0	30_010Ch

11.4.3.1.137.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.137.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. Note: This register field is sticky.

11.4.3.1.138 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_0)

11.4.3.1.138.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_INBOUND_0	30_0110h

11.4.3.1.138.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.4.3.1.138.3 Fields

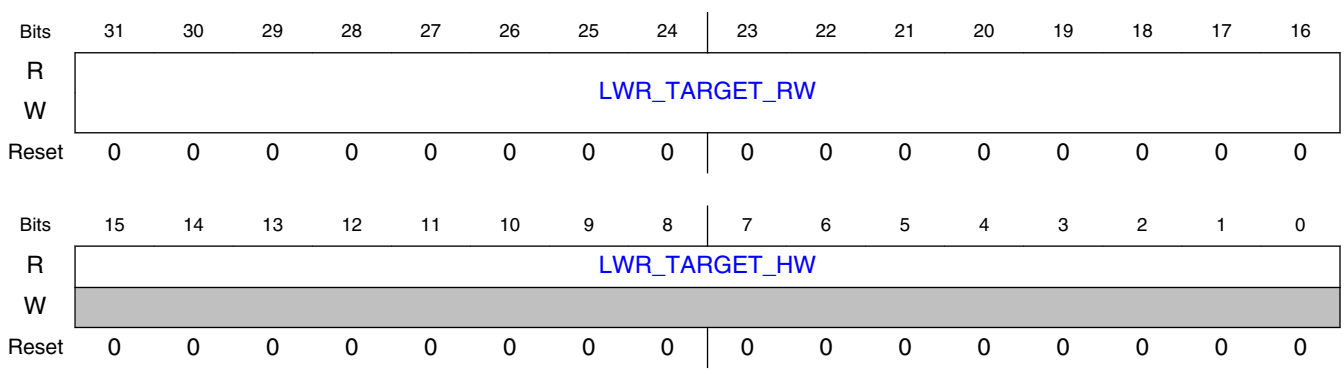
Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.4.3.1.139 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_0)

11.4.3.1.139.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_INBOUND_0	30_0114h

11.4.3.1.139.2 Diagram



11.4.3.1.139.3 Fields

Field	Function
31-16 LWR_TARGET_RW	Forms MSB's of the Lower Target part of the new address of the translated region. These bits are always '0'. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode. Note: This register field is sticky.

Table continues on the next page...

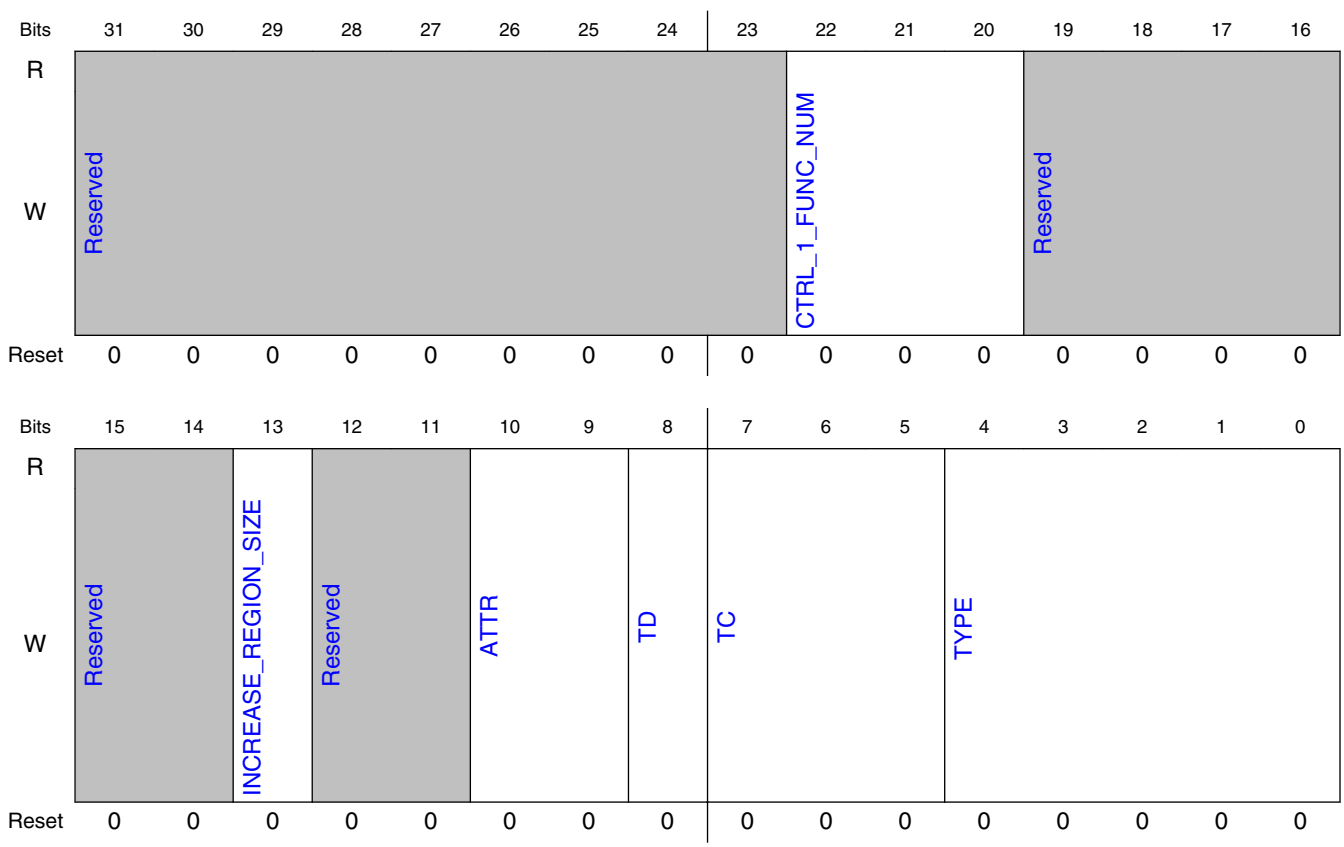
Field	Function
15-0 LWR_TARGET_HW	Forms the LSB's of the Lower Target part of the new address of the translated region. The start address must be aligned to a Minimum Size of iATU Region kB boundary (in address match mode); and to the Bar size boundary (in BAR match mode) so that these bits are always '0'. If the BAR is smaller than the iATU region size, then the iATU target address must align to the iATU region size; otherwise it must align to the BAR size. A write to this location is ignored by the PCIe controller. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode.

11.4.3.1.140 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_1)

11.4.3.1.140.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_OUTBOUND_1	30_0200h

11.4.3.1.140.2 Diagram



11.4.3.1.140.3 Fields

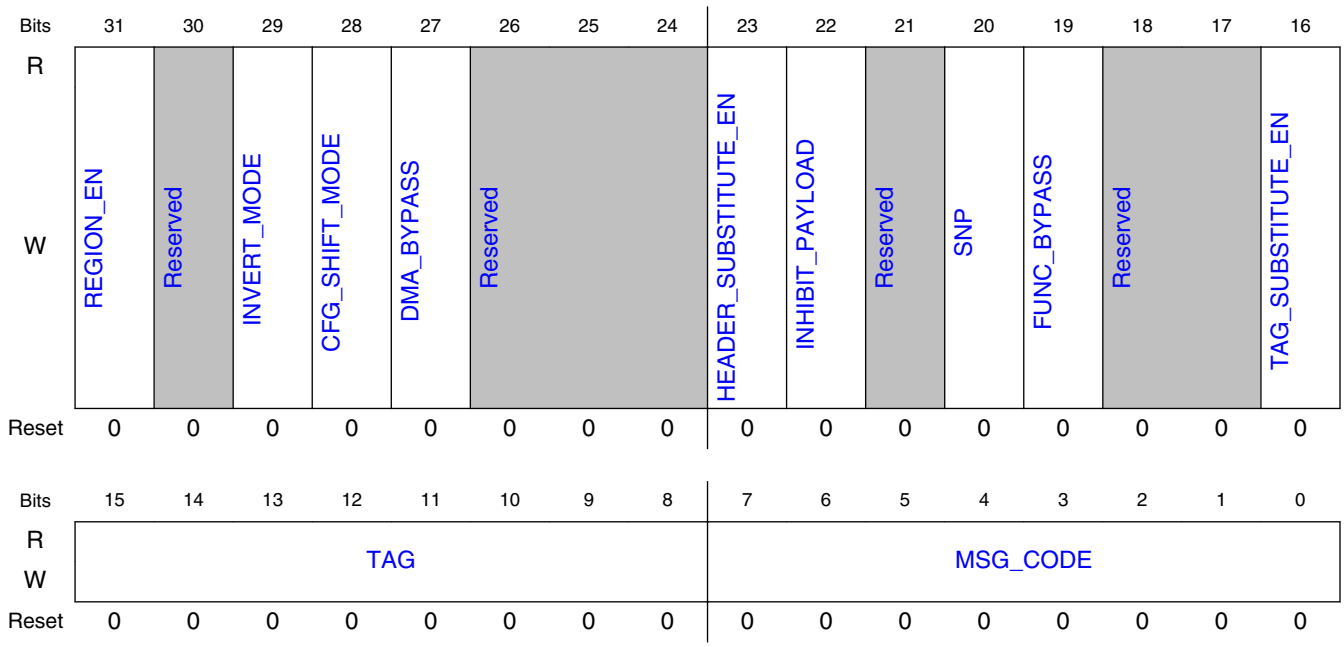
Field	Function
31-23 —	Reserved
22-20 CTRL_1_FUNC_NUM	Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). - When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
19-14 —	Reserved
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by Maximum Size of iATU Region When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved
10-9 ATTR	When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8 TD	When the address of an outbound TLP is matched to this region, then the TD field of the TLP is changed to the value in this register. Note: This register field is sticky.
7-5 TC	When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4-0 TYPE	When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

11.4.3.1.141 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_1)

11.4.3.1.141.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_OUTBOUND_1	30_0204h

11.4.3.1.141.2 Diagram



11.4.3.1.141.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 —	Reserved
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27 DMA_BYPASS	DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
26-24 —	Reserved
23 HEADER_SUBSTITUTE_EN	Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. - 1: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register is used to fill bytes 8-to-11 (for 3 DWORD header) or bytes 12-to-15 (for 4 DWORD header) of the translated TLP

Table continues on the next page...

Field	Function
	header. - 0: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register forms the new address of the translated region. Note: This register field is sticky.
22 INHIBIT_PAYLOAD	Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. - 1: Fmt[1] =0 so that only TLP type without data is sent. For example, a Msg instead of MsgD will be sent. - 0: Fmt[1] =0/1 so that TLPs with or without data can be sent. Note: This register field is sticky.
21 —	Reserved
20 SNP	Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19 FUNC_BYPASS	Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18-17 —	Reserved
16 TAG_SUBSTITUTE_EN	TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note (CX_TPH_ENABLE=1): TAG substitution for MWr will not occur because this field (byte 6) in the TLP header is the ST field. ST substitution can still take place using the MSG_CODE field in IATU_REGION_CTRL_2_OFF_OUTBOUND_i. Note: This register field is sticky.
15-8 TAG	TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST;Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

11.4.3.1.142 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_1)

11.4.3.1.142.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_OUTBOUND_1	30_0208h

11.4.3.1.142.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies

the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.4.3.1.142.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LWR_BASE_RW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_BASE_HW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.142.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.4.3.1.143 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_1)

11.4.3.1.143.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_1	30_020Ch

11.4.3.1.143.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.143.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

11.4.3.1.144 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_1)

11.4.3.1.144.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_OUTBOUND_1	30_0210h

11.4.3.1.144.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.4.3.1.144.3 Fields

Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.4.3.1.145 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_1)

11.4.3.1.145.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_1	30_0214h

11.4.3.1.145.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.145.3 Fields

Field	Function
31-0 LWR_TARGET_RW_OUTBOUND	When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to a Minimum Size of iATU Region kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(Minimum Size of iATU Region). When

Field	Function
	HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

11.4.3.1.146 iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_1)

11.4.3.1.146.1 Offset

Register	Offset
IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_1	30_0218h

11.4.3.1.146.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.146.3 Fields

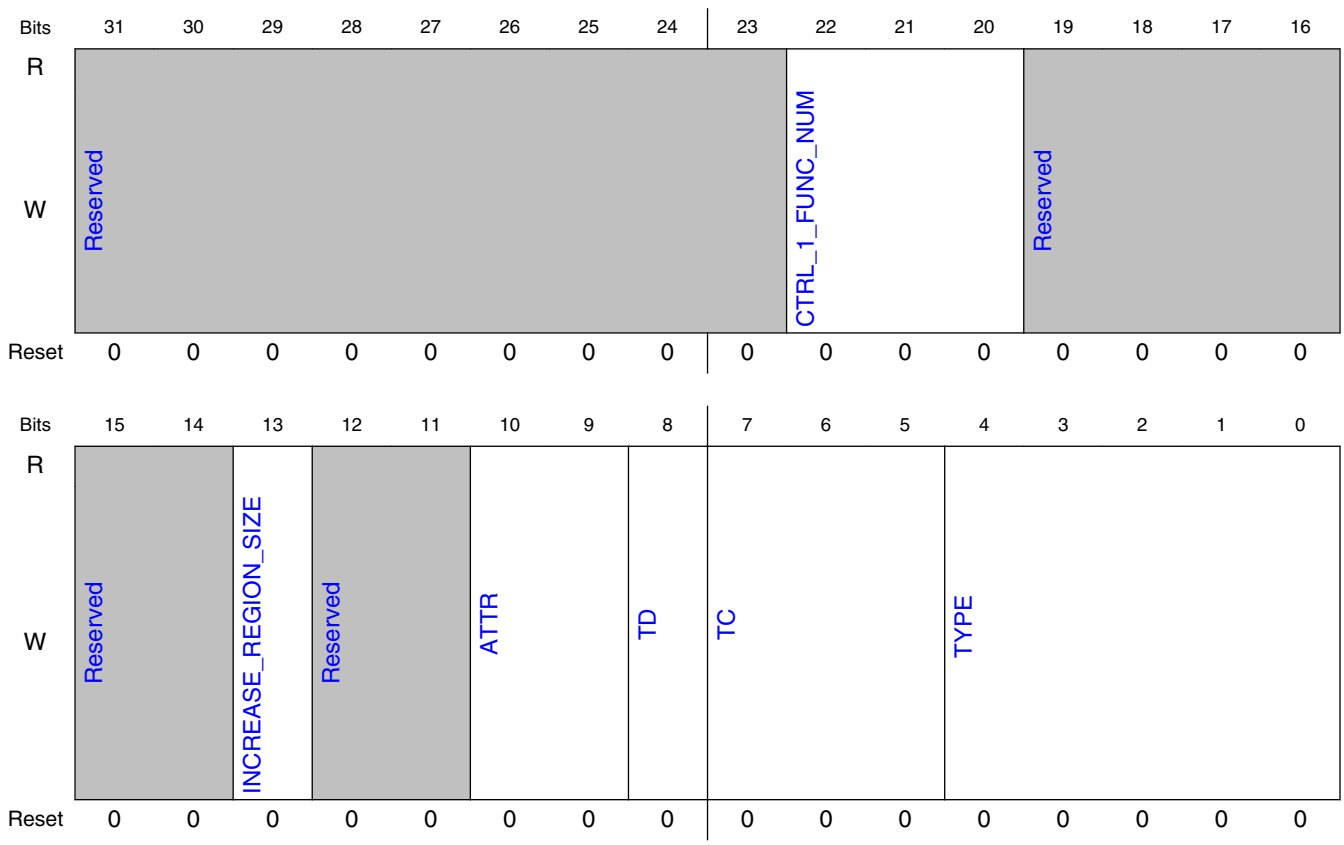
Field	Function
31-0 UPPER_TARGET_RW	Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

11.4.3.1.147 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_1)

11.4.3.1.147.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_INBOUND_1	30_0300h

11.4.3.1.147.2 Diagram



11.4.3.1.147.3 Fields

Field	Function
31-23 —	Reserved
22-20	Function Number. - MEM-I/O: When the Address and BAR matching logic in the controller indicate that a MEM-I/O transaction matches a BAR in the function corresponding to this value, then address translation

Table continues on the next page...

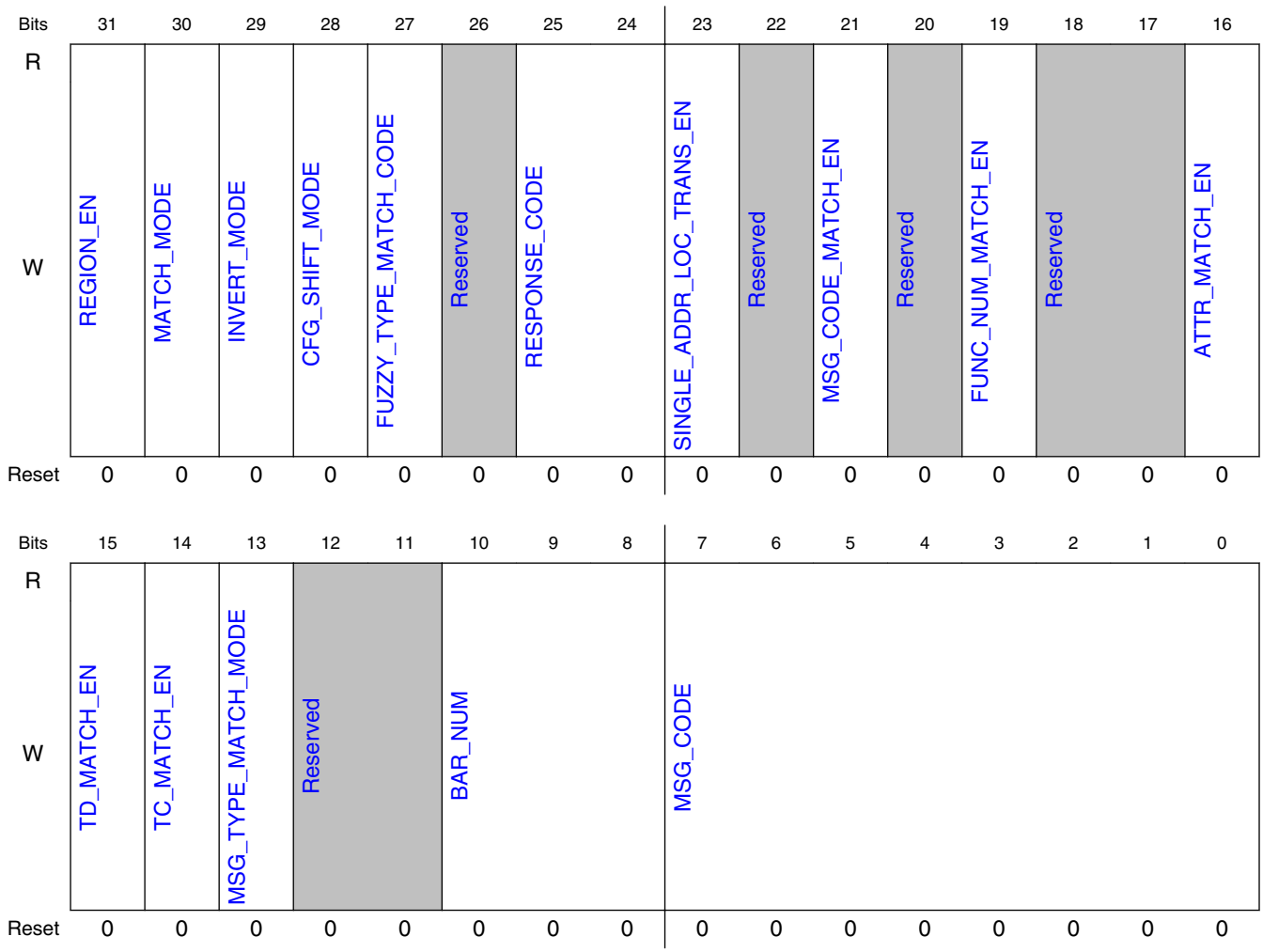
Field	Function
CTRL_1_FUNC_NUM	proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. - CFG0/CFG1: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
19-14 —	Reserved
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by Maximum Size of iATU Region When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved
10-9 ATTR	When the ATTR field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "ATTR Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
8 TD	When the TD field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TD Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
7-5 TC	When the TC field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TC Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
4-0 TYPE	When the TYPE field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). Note: This register field is sticky.

11.4.3.1.148 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_1)

11.4.3.1.148.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_INBOUND_1	30_0304h

11.4.3.1.148.2 Diagram



11.4.3.1.148.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 MATCH_MODE	Match Mode. Determines Inbound matching mode for TLPs. The mode depends on the type of TLP that is received as follows: For MEM-I/O TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU operates using addresses as in the outbound direction. The Region Base and Limit Registers must be setup. - 1:BAR Match Mode. BAR matching is used. The "BAR Number" field is relevant. Not used for RC. For CFG0 TLPs, this field is interpreted as follows: - 0: Routing ID Match Mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM-I/O transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. - 1: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. For MSG/MSGD TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU treats the third dword and fourth dword of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers. - 1: Vendor ID

Table continues on the next page...

Field	Function
	Match Mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the third dword of the TLP header, but matches against the Vendor ID in bits [15:0] of the third dword of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the fourth dword of the TLP header. - If SINGLE_ADDRESS_LOCATION_TRANSLATE_EN = 1 AND MSG_TYPE_MATCH_MODE = 1, then Match Mode is ignored. Note: This register field is sticky.
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of your application memory space using a 28-bit effective address. Shifts bits [31:16] of the untranslated address to form bits [27:12] of the translated address. Note: This register field is sticky.
27 FUZZY_TYPE_MATCH_CODE	Fuzzy Type Match Mode. When enabled, the iATU relaxes the matching of the TLP TYPE field against the expected TYPE field so that - CfgRd0 and CfgRd1 TLPs are seen as identical. Similarly with CfgWr0 and CfgWr1. - MWr, MRd and MRdLk TLPs are seen as identical - The Routing field of Msg/MsgD TLPs is ignored - FetchAdd, Swap and CAS are seen as identical. For example, CFG0 in the TYPE field in the "iATU Control 1 Register" matches against an inbound CfgRd0, CfgRd1, CfgWr0 or CfgWr1 TLP. Note: This register field is sticky.
26 —	Reserved
25-24 RESPONSE_CODE	Response Code. Defines the type of response to give for accesses matching this region. This overrides the normal RADM filter response. Note that this feature is not available for any region where Single Address Location Translate is enabled. - 00 - Normal RADM filter response is used. - 01 - Unsupported request (UR) - 10 - Completer abort (CA) - 11 - Not used / undefined / reserved. Note: This register field is sticky.
23 SINGLE_ADDR_LOC_TRANS_EN	Single Address Location Translate Enable. When enabled, Rx TLPs can be translated to a single address location as determined by the target address register of the iATU region. The main usage scenario is translation of Messages (such as Vendor Defined or ATS Messages) to MWr TLPs when the AXI bridge is enabled. Note: This register field is sticky.
22 —	Reserved
21 MSG_CODE_MATCH_EN	Message Code Match Enable (Msg TLPS). Ensures that a successful message Code TLP field comparison match (see Message Code field of the "iATU Control 2 Register") occurs (in MSG transactions) for address translation to proceed. ST Match Enable (Mem TLPs). Ensures that a successful ST TLP field comparison match (see ST field of the "iATU Control 2 Register") occurs (in MEM transactions) for address translation to proceed. Only Valid when the CX_TPH_ENABLE configuration parameter is 1 Note: This register field is sticky.
20 —	Reserved
19 FUNC_NUM_MATCH_EN	Function Number Match Enable. Ensures that a successful Function Number TLP field comparison match (see Function Number field of the "iATU Control 1 Register") occurs (in MEM-I/O and CFG0/CFG1 transactions) for address translation to proceed. Note: This register field is sticky.
18-17 —	Reserved
16	ATTR Match Enable. Ensures that a successful ATTR TLP field comparison match (see ATTR field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.

Table continues on the next page...

Field	Function
ATTR_MATCH_EN	
15 TD_MATCH_EN	TD Match Enable. Ensures that a successful TD TLP field comparison match (see TD field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
14 TC_MATCH_EN	TC Match Enable. Ensures that a successful TC TLP field comparison match (see TC field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
13 MSG_TYPE_MATCH_MODE	Message Type Match Mode. When enabled, and if single address location translate enable is set, then inbound TLPs of type MSG/MSGd which match the type field of the iatu_region_ctrl_1_OFF_inbound register (=>TYPE[4:3]=2'b10) will be translated. Message type match mode overrides any value of MATCH_MODE field in this register. Usage scenarios for this are translation of VDM or ATS messages when AXI bridge is configured on client interface. Note: This register field is sticky.
12-11 —	Reserved
10-8 BAR_NUM	BAR Number. When the BAR number of an inbound MEM or IO TLP " that is matched by the normal internal BAR address matching mechanism " is the same as this field, address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Match Mode" bit of the "iATU Control 2 Register" is set. - 000b - BAR0 - 001b - BAR1 - 010b - BAR2 - 011b - BAR3 - 100b - BAR4 - 101b - BAR5 - 110b - ROM - 111b - reserved - IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b - 101b and that BAR configured as an IO BAR. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs: (Message Code). When the TYPE field of an inbound Msg/MsgD TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Message Code Match Enable" bit of the "iATU Control 2 Register" is set. Memory TLPs: (ST;Steering Tag). When the ST field of an inbound TLP is matched to this value, then address translation proceeds. This check is only performed if the "ST Match Enable" bit of the "iATU Control2 Register" is set. The setting is independent of the setting of the TH field. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

11.4.3.1.149 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_1)

11.4.3.1.149.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_INBOUND_1	30_0308h

11.4.3.1.149.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned

on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.4.3.1.149.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_BASE_HW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.149.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.4.3.1.150 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_1)

11.4.3.1.150.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_INBOUND_1	30_030Ch

11.4.3.1.150.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.150.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. Note: This register field is sticky.

11.4.3.1.151 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_1)

11.4.3.1.151.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_INBOUND_1	30_0310h

11.4.3.1.151.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.4.3.1.151.3 Fields

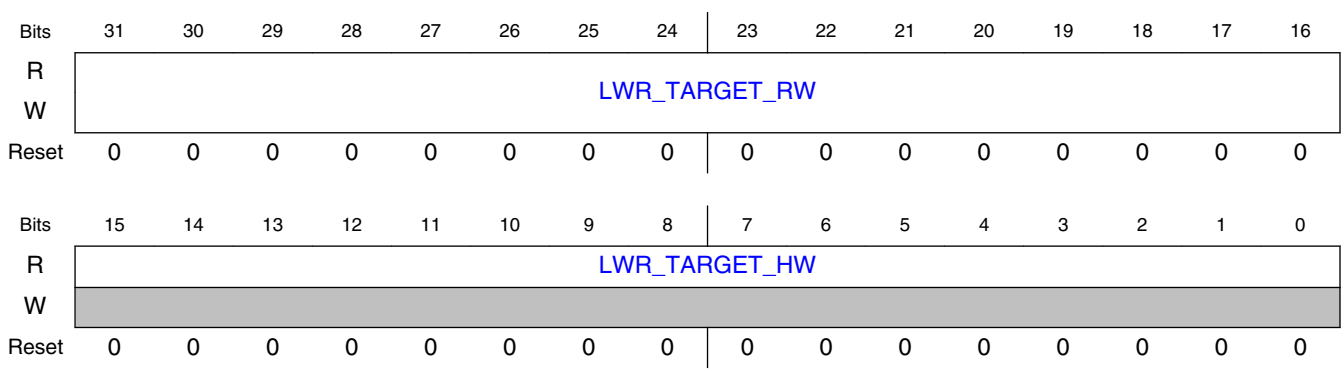
Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.4.3.1.152 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_1)

11.4.3.1.152.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_INBOUND_1	30_0314h

11.4.3.1.152.2 Diagram



11.4.3.1.152.3 Fields

Field	Function
31-16 LWR_TARGET_RW	Forms MSB's of the Lower Target part of the new address of the translated region. These bits are always '0'. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode. Note: This register field is sticky.

Table continues on the next page...

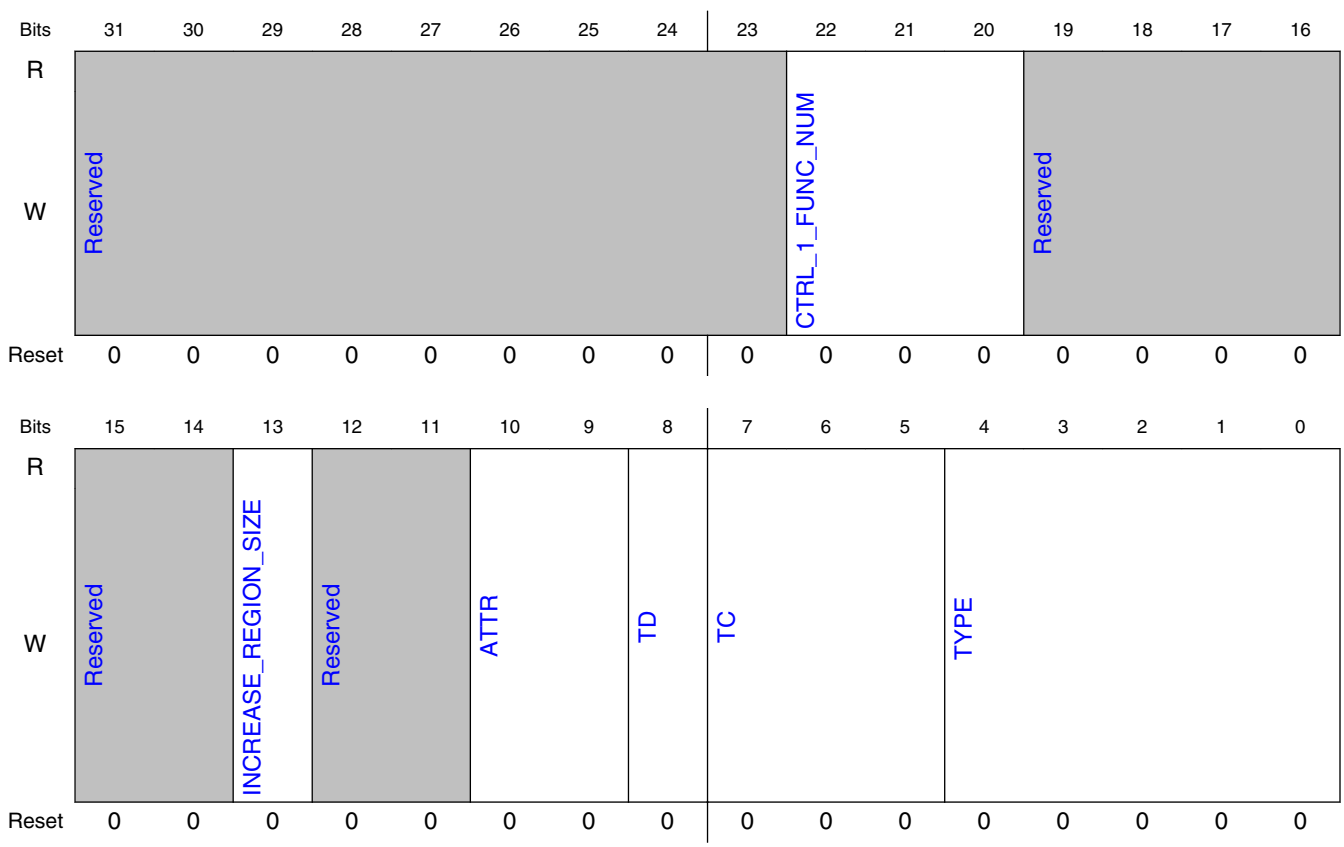
Field	Function
15-0 LWR_TARGET_HW	Forms the LSB's of the Lower Target part of the new address of the translated region. The start address must be aligned to a Minimum Size of iATU Region kB boundary (in address match mode); and to the Bar size boundary (in BAR match mode) so that these bits are always '0'. If the BAR is smaller than the iATU region size, then the iATU target address must align to the iATU region size; otherwise it must align to the BAR size. A write to this location is ignored by the PCIe controller. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode.

11.4.3.1.153 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_2)

11.4.3.1.153.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_OUTBOUND_2	30_0400h

11.4.3.1.153.2 Diagram



11.4.3.1.153.3 Fields

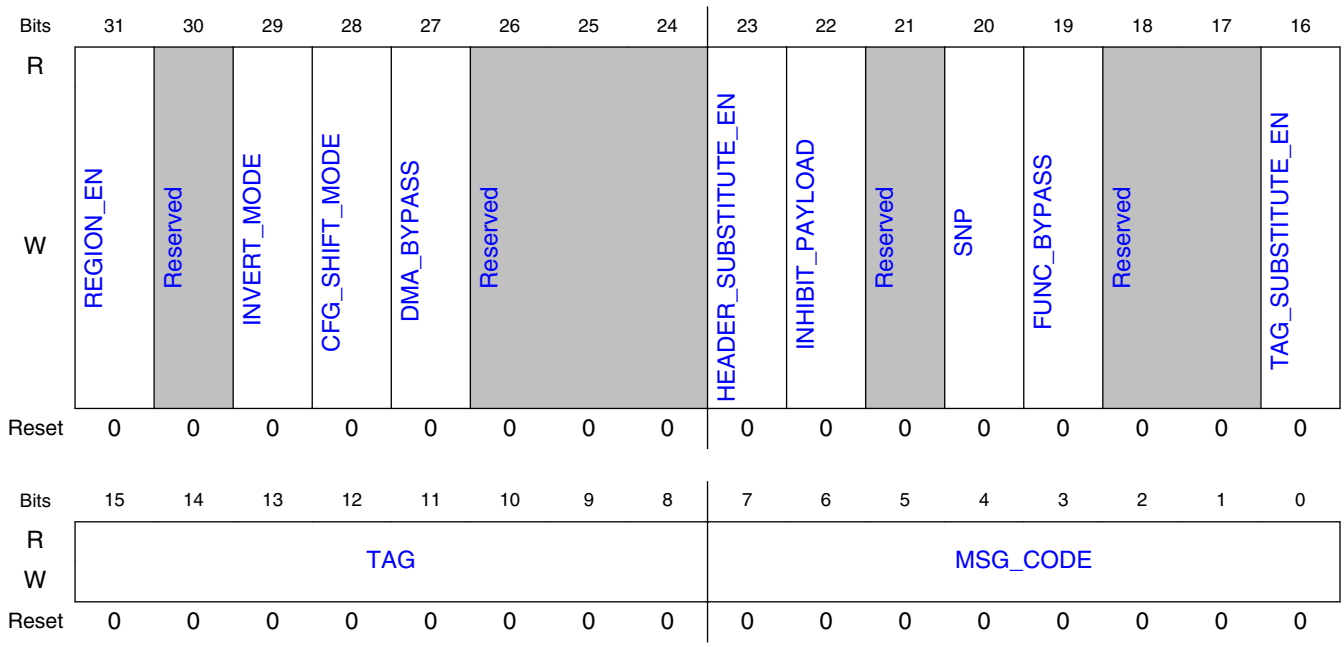
Field	Function
31-23 —	Reserved
22-20 CTRL_1_FUNC_NUM	Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). - When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
19-14 —	Reserved
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by Maximum Size of iATU Region When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved
10-9 ATTR	When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8 TD	When the address of an outbound TLP is matched to this region, then the TD field of the TLP is changed to the value in this register. Note: This register field is sticky.
7-5 TC	When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4-0 TYPE	When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

11.4.3.1.154 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_2)

11.4.3.1.154.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_OUTBOUND_2	30_0404h

11.4.3.1.154.2 Diagram



11.4.3.1.154.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 —	Reserved
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27 DMA_BYPASS	DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
26-24 —	Reserved
23 HEADER_SUBSTITUTE_EN	Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. - 1: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register is used to fill bytes 8-to-11 (for 3 DWORD header) or bytes 12-to-15 (for 4 DWORD header) of the translated TLP

Table continues on the next page...

Field	Function
	header. - 0: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register forms the new address of the translated region. Note: This register field is sticky.
22 INHIBIT_PAYLOAD	Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. - 1: Fmt[1] =0 so that only TLP type without data is sent. For example, a Msg instead of MsgD will be sent. - 0: Fmt[1] =0/1 so that TLPs with or without data can be sent. Note: This register field is sticky.
21 —	Reserved
20 SNP	Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19 FUNC_BYPASS	Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18-17 —	Reserved
16 TAG_SUBSTITUTE_EN	TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note (CX_TPH_ENABLE=1): TAG substitution for MWr will not occur because this field (byte 6) in the TLP header is the ST field. ST substitution can still take place using the MSG_CODE field in IATU_REGION_CTRL_2_OFF_OUTBOUND_i. Note: This register field is sticky.
15-8 TAG	TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST;Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

11.4.3.1.155 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_2)

11.4.3.1.155.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_OUTBOUND_2	30_0408h

11.4.3.1.155.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies

the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.4.3.1.155.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LWR_BASE_RW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_BASE_HW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.155.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.4.3.1.156 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_2)

11.4.3.1.156.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_2	30_040Ch

11.4.3.1.156.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.156.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

11.4.3.1.157 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_2)

11.4.3.1.157.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_OUTBOUND_2	30_0410h

11.4.3.1.157.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.4.3.1.157.3 Fields

Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.4.3.1.158 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_2)

11.4.3.1.158.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_2	30_0414h

11.4.3.1.158.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.158.3 Fields

Field	Function
31-0 LWR_TARGET_RW_OUTBOUND	When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to a Minimum Size of iATU Region kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(Minimum Size of iATU Region). When

Field	Function
	HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

11.4.3.1.159 iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_2)

11.4.3.1.159.1 Offset

Register	Offset
IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_2	30_0418h

11.4.3.1.159.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.159.3 Fields

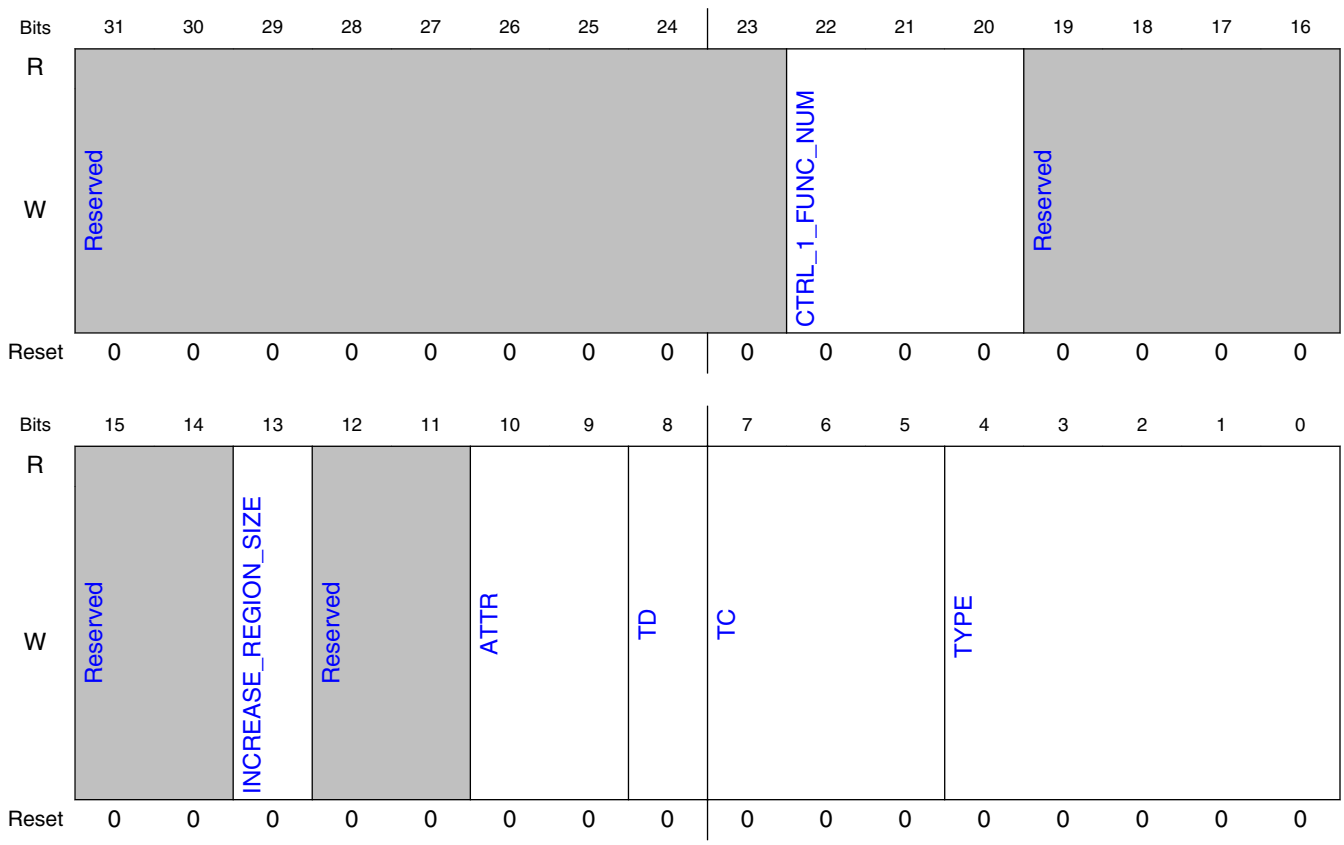
Field	Function
31-0 UPPER_TARGET_RW	Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

11.4.3.1.160 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_2)

11.4.3.1.160.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_INBOUND_2	30_0500h

11.4.3.1.160.2 Diagram



11.4.3.1.160.3 Fields

Field	Function
31-23 —	Reserved
22-20	Function Number. - MEM-I/O: When the Address and BAR matching logic in the controller indicate that a MEM-I/O transaction matches a BAR in the function corresponding to this value, then address translation

Table continues on the next page...

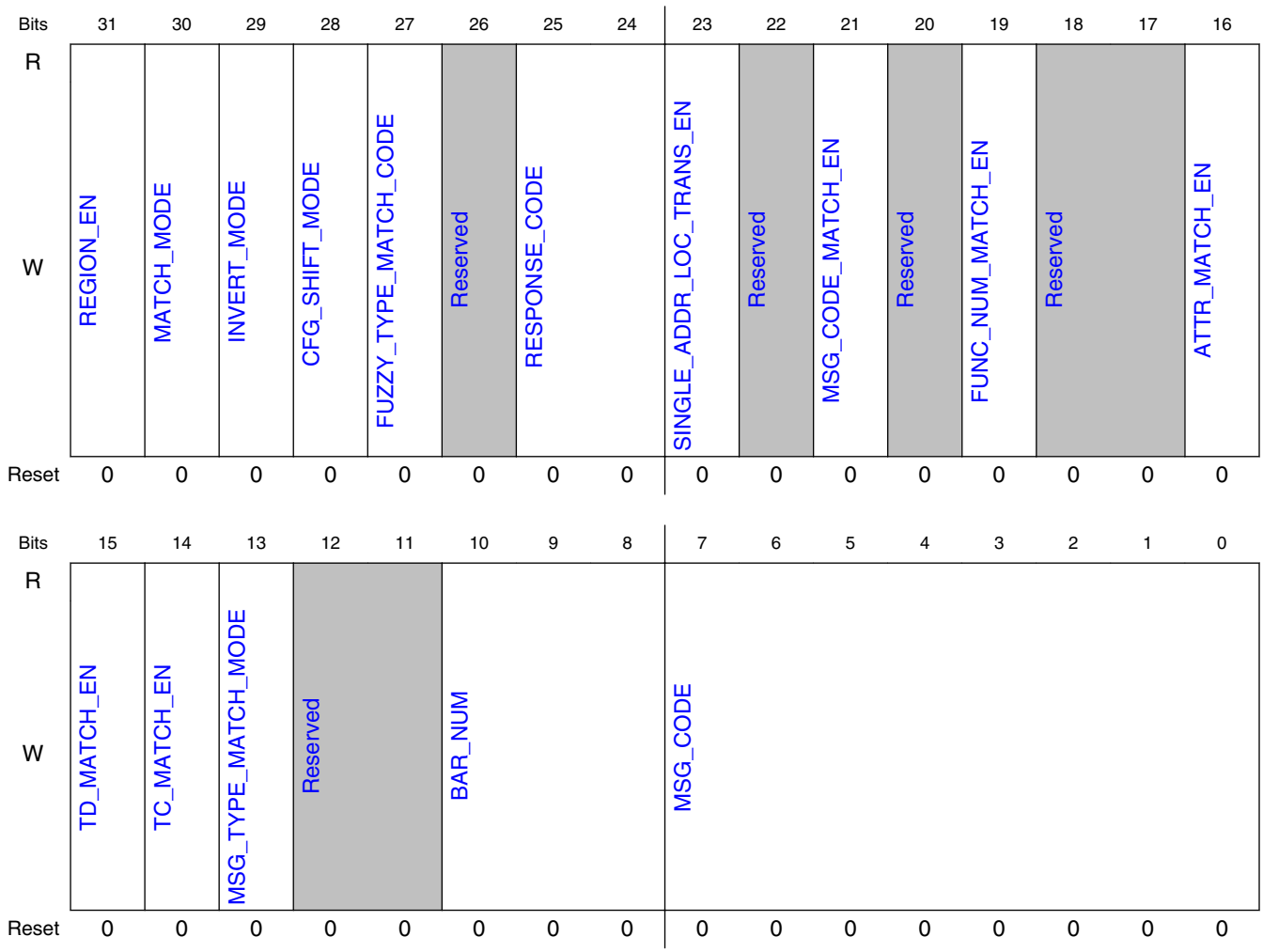
Field	Function
CTRL_1_FUNC_NUM	proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. - CFG0/CFG1: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
19-14 —	Reserved
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by Maximum Size of iATU Region When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved
10-9 ATTR	When the ATTR field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "ATTR Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
8 TD	When the TD field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TD Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
7-5 TC	When the TC field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TC Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
4-0 TYPE	When the TYPE field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). Note: This register field is sticky.

11.4.3.1.161 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_2)

11.4.3.1.161.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_INBOUND_2	30_0504h

11.4.3.1.161.2 Diagram



11.4.3.1.161.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 MATCH_MODE	Match Mode. Determines Inbound matching mode for TLPs. The mode depends on the type of TLP that is received as follows: For MEM-I/O TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU operates using addresses as in the outbound direction. The Region Base and Limit Registers must be setup. - 1:BAR Match Mode. BAR matching is used. The "BAR Number" field is relevant. Not used for RC. For CFG0 TLPs, this field is interpreted as follows: - 0: Routing ID Match Mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM-I/O transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. - 1: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. For MSG/MSGD TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU treats the third dword and fourth dword of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers. - 1: Vendor ID

Table continues on the next page...

Field	Function
	Match Mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the third dword of the TLP header, but matches against the Vendor ID in bits [15:0] of the third dword of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the fourth dword of the TLP header. - If SINGLE_ADDRESS_LOCATION_TRANSLATE_EN = 1 AND MSG_TYPE_MATCH_MODE = 1, then Match Mode is ignored. Note: This register field is sticky.
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of your application memory space using a 28-bit effective address. Shifts bits [31:16] of the untranslated address to form bits [27:12] of the translated address. Note: This register field is sticky.
27 FUZZY_TYPE_MATCH_CODE	Fuzzy Type Match Mode. When enabled, the iATU relaxes the matching of the TLP TYPE field against the expected TYPE field so that - CfgRd0 and CfgRd1 TLPs are seen as identical. Similarly with CfgWr0 and CfgWr1. - MWr, MRd and MRdLk TLPs are seen as identical - The Routing field of Msg/MsgD TLPs is ignored - FetchAdd, Swap and CAS are seen as identical. For example, CFG0 in the TYPE field in the "iATU Control 1 Register" matches against an inbound CfgRd0, CfgRd1, CfgWr0 or CfgWr1 TLP. Note: This register field is sticky.
26 —	Reserved
25-24 RESPONSE_CODE	Response Code. Defines the type of response to give for accesses matching this region. This overrides the normal RADM filter response. Note that this feature is not available for any region where Single Address Location Translate is enabled. - 00 - Normal RADM filter response is used. - 01 - Unsupported request (UR) - 10 - Completer abort (CA) - 11 - Not used / undefined / reserved. Note: This register field is sticky.
23 SINGLE_ADDR_LOC_TRANS_EN	Single Address Location Translate Enable. When enabled, Rx TLPs can be translated to a single address location as determined by the target address register of the iATU region. The main usage scenario is translation of Messages (such as Vendor Defined or ATS Messages) to MWr TLPs when the AXI bridge is enabled. Note: This register field is sticky.
22 —	Reserved
21 MSG_CODE_MATCH_EN	Message Code Match Enable (Msg TLPS). Ensures that a successful message Code TLP field comparison match (see Message Code field of the "iATU Control 2 Register") occurs (in MSG transactions) for address translation to proceed. ST Match Enable (Mem TLPs). Ensures that a successful ST TLP field comparison match (see ST field of the "iATU Control 2 Register") occurs (in MEM transactions) for address translation to proceed. Only Valid when the CX_TPH_ENABLE configuration parameter is 1 Note: This register field is sticky.
20 —	Reserved
19 FUNC_NUM_MATCH_EN	Function Number Match Enable. Ensures that a successful Function Number TLP field comparison match (see Function Number field of the "iATU Control 1 Register") occurs (in MEM-I/O and CFG0/CFG1 transactions) for address translation to proceed. Note: This register field is sticky.
18-17 —	Reserved
16	ATTR Match Enable. Ensures that a successful ATTR TLP field comparison match (see ATTR field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.

Table continues on the next page...

Field	Function
ATTR_MATCH_EN	
15 TD_MATCH_EN	TD Match Enable. Ensures that a successful TD TLP field comparison match (see TD field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
14 TC_MATCH_EN	TC Match Enable. Ensures that a successful TC TLP field comparison match (see TC field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
13 MSG_TYPE_MATCH_MODE	Message Type Match Mode. When enabled, and if single address location translate enable is set, then inbound TLPs of type MSG/MSGd which match the type field of the iatu_region_ctrl_1_OFF_inbound register (=>TYPE[4:3]=2'b10) will be translated. Message type match mode overrides any value of MATCH_MODE field in this register. Usage scenarios for this are translation of VDM or ATS messages when AXI bridge is configured on client interface. Note: This register field is sticky.
12-11 —	Reserved
10-8 BAR_NUM	BAR Number. When the BAR number of an inbound MEM or IO TLP " that is matched by the normal internal BAR address matching mechanism " is the same as this field, address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Match Mode" bit of the "iATU Control 2 Register" is set. - 000b - BAR0 - 001b - BAR1 - 010b - BAR2 - 011b - BAR3 - 100b - BAR4 - 101b - BAR5 - 110b - ROM - 111b - reserved - IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b - 101b and that BAR configured as an IO BAR. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs: (Message Code). When the TYPE field of an inbound Msg/MsgD TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Message Code Match Enable" bit of the "iATU Control 2 Register" is set. Memory TLPs: (ST;Steering Tag). When the ST field of an inbound TLP is matched to this value, then address translation proceeds. This check is only performed if the "ST Match Enable" bit of the "iATU Control2 Register" is set. The setting is independent of the setting of the TH field. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

11.4.3.1.162 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_2)

11.4.3.1.162.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_INBOUND_2	30_0508h

11.4.3.1.162.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned

on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.4.3.1.162.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_BASE_HW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.162.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.4.3.1.163 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_2)

11.4.3.1.163.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_INBOUND_2	30_050Ch

11.4.3.1.163.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.163.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. Note: This register field is sticky.

11.4.3.1.164 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_2)

11.4.3.1.164.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_INBOUND_2	30_0510h

11.4.3.1.164.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.4.3.1.164.3 Fields

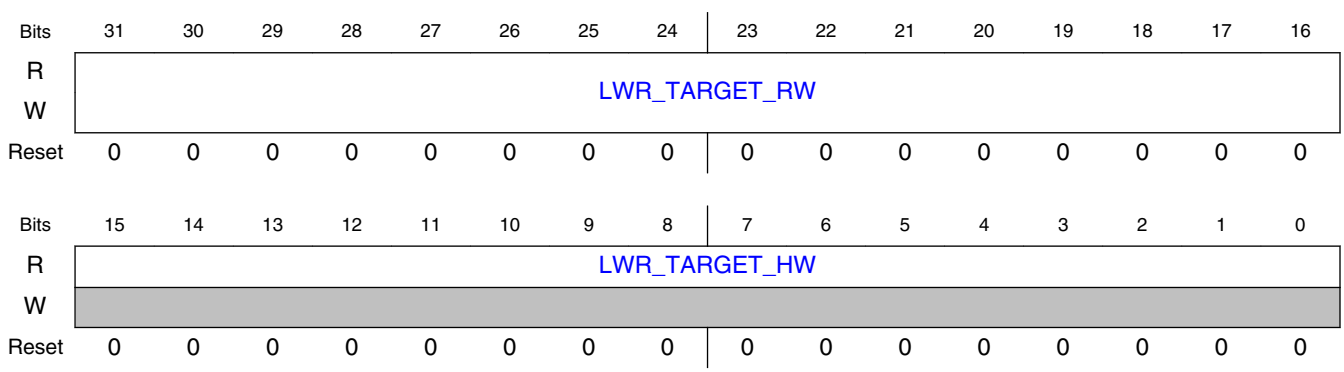
Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.4.3.1.165 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_2)

11.4.3.1.165.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_INBOUND_2	30_0514h

11.4.3.1.165.2 Diagram



11.4.3.1.165.3 Fields

Field	Function
31-16 LWR_TARGET_RW	Forms MSB's of the Lower Target part of the new address of the translated region. These bits are always '0'. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode. Note: This register field is sticky.

Table continues on the next page...

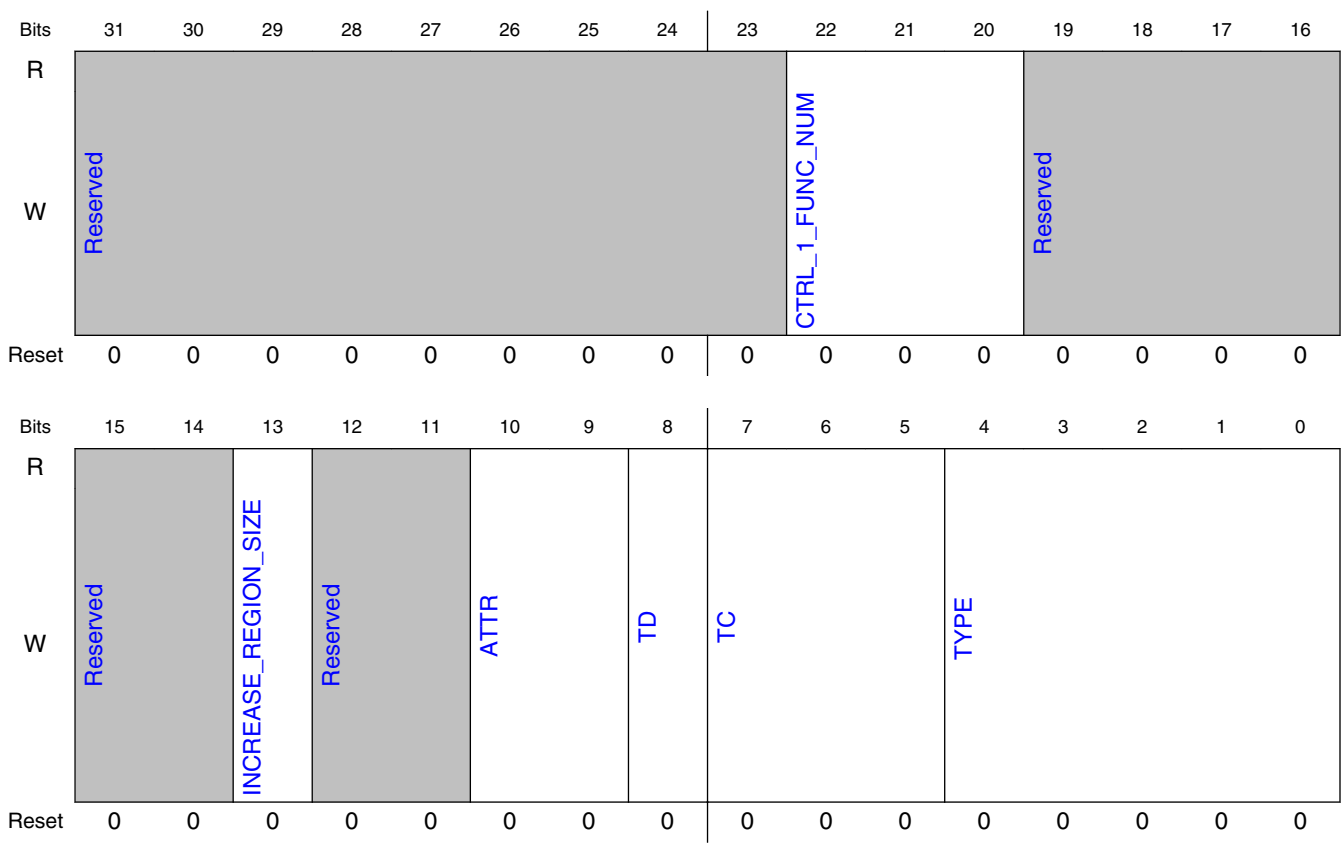
Field	Function
15-0 LWR_TARGET_HW	Forms the LSB's of the Lower Target part of the new address of the translated region. The start address must be aligned to a Minimum Size of iATU Region kB boundary (in address match mode); and to the Bar size boundary (in BAR match mode) so that these bits are always '0'. If the BAR is smaller than the iATU region size, then the iATU target address must align to the iATU region size; otherwise it must align to the BAR size. A write to this location is ignored by the PCIe controller. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode.

11.4.3.1.166 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_3)

11.4.3.1.166.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_OUTBOUND_3	30_0600h

11.4.3.1.166.2 Diagram



11.4.3.1.166.3 Fields

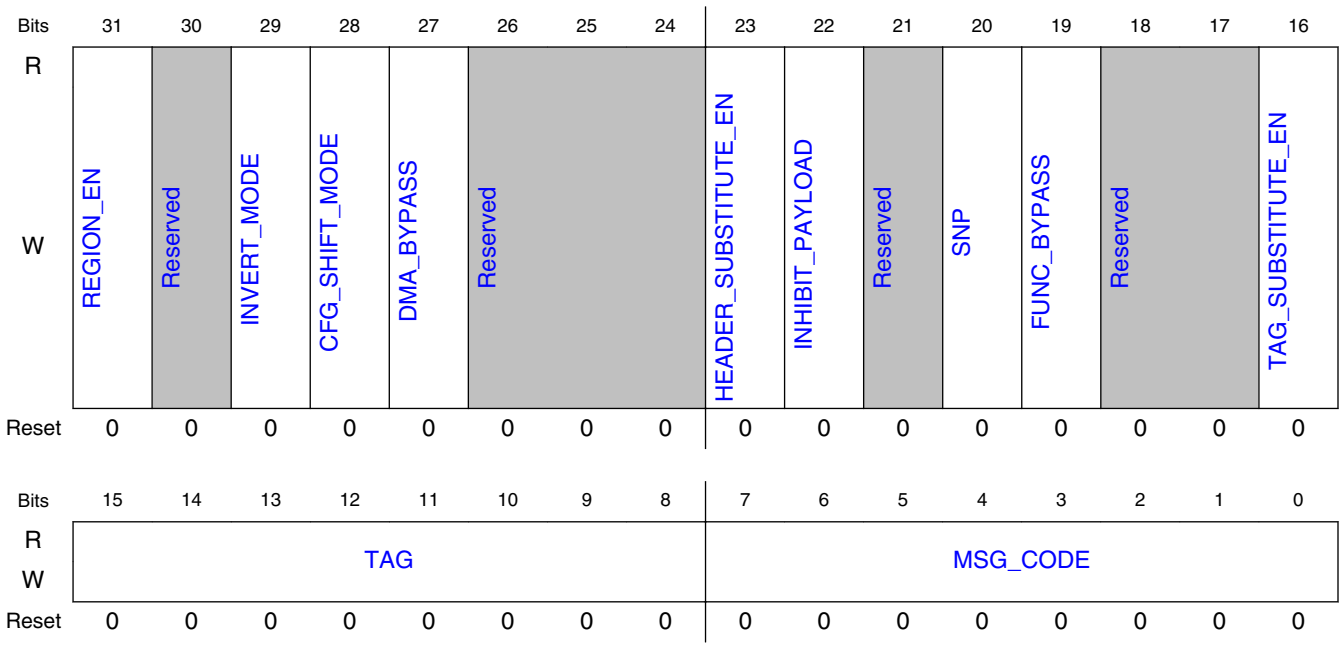
Field	Function
31-23 —	Reserved
22-20 CTRL_1_FUNC_NUM	Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). - When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
19-14 —	Reserved
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by Maximum Size of iATU Region When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved
10-9 ATTR	When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8 TD	When the address of an outbound TLP is matched to this region, then the TD field of the TLP is changed to the value in this register. Note: This register field is sticky.
7-5 TC	When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4-0 TYPE	When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

11.4.3.1.167 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_3)

11.4.3.1.167.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_OUTBOUND_3	30_0604h

11.4.3.1.167.2 Diagram



11.4.3.1.167.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 —	Reserved
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27 DMA_BYPASS	DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
26-24 —	Reserved
23 HEADER_SUBSTITUTE_EN	Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. - 1: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register is used to fill bytes 8-to-11 (for 3 DWORD header) or bytes 12-to-15 (for 4 DWORD header) of the translated TLP

Table continues on the next page...

Field	Function
	header. - 0: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register forms the new address of the translated region. Note: This register field is sticky.
22 INHIBIT_PAYLOAD	Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. - 1: Fmt[1] =0 so that only TLP type without data is sent. For example, a Msg instead of MsgD will be sent. - 0: Fmt[1] =0/1 so that TLPs with or without data can be sent. Note: This register field is sticky.
21 —	Reserved
20 SNP	Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19 FUNC_BYPASS	Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18-17 —	Reserved
16 TAG_SUBSTITUTE_EN	TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note (CX_TPH_ENABLE=1): TAG substitution for MWr will not occur because this field (byte 6) in the TLP header is the ST field. ST substitution can still take place using the MSG_CODE field in IATU_REGION_CTRL_2_OFF_OUTBOUND_i. Note: This register field is sticky.
15-8 TAG	TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST;Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

11.4.3.1.168 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_3)

11.4.3.1.168.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_OUTBOUND_3	30_0608h

11.4.3.1.168.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies

the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.4.3.1.168.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LWR_BASE_RW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_BASE_HW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.168.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.4.3.1.169 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_3)

11.4.3.1.169.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_3	30_060Ch

11.4.3.1.169.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.169.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

11.4.3.1.170 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_3)

11.4.3.1.170.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_OUTBOUND_3	30_0610h

11.4.3.1.170.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.4.3.1.170.3 Fields

Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.4.3.1.171 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_3)

11.4.3.1.171.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_3	30_0614h

11.4.3.1.171.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.171.3 Fields

Field	Function
31-0 LWR_TARGET_RW_OUTBOUND	When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to a Minimum Size of iATU Region kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(Minimum Size of iATU Region). When

Field	Function
	HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

11.4.3.1.172 iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_3)

11.4.3.1.172.1 Offset

Register	Offset
IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_3	30_0618h

11.4.3.1.172.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.172.3 Fields

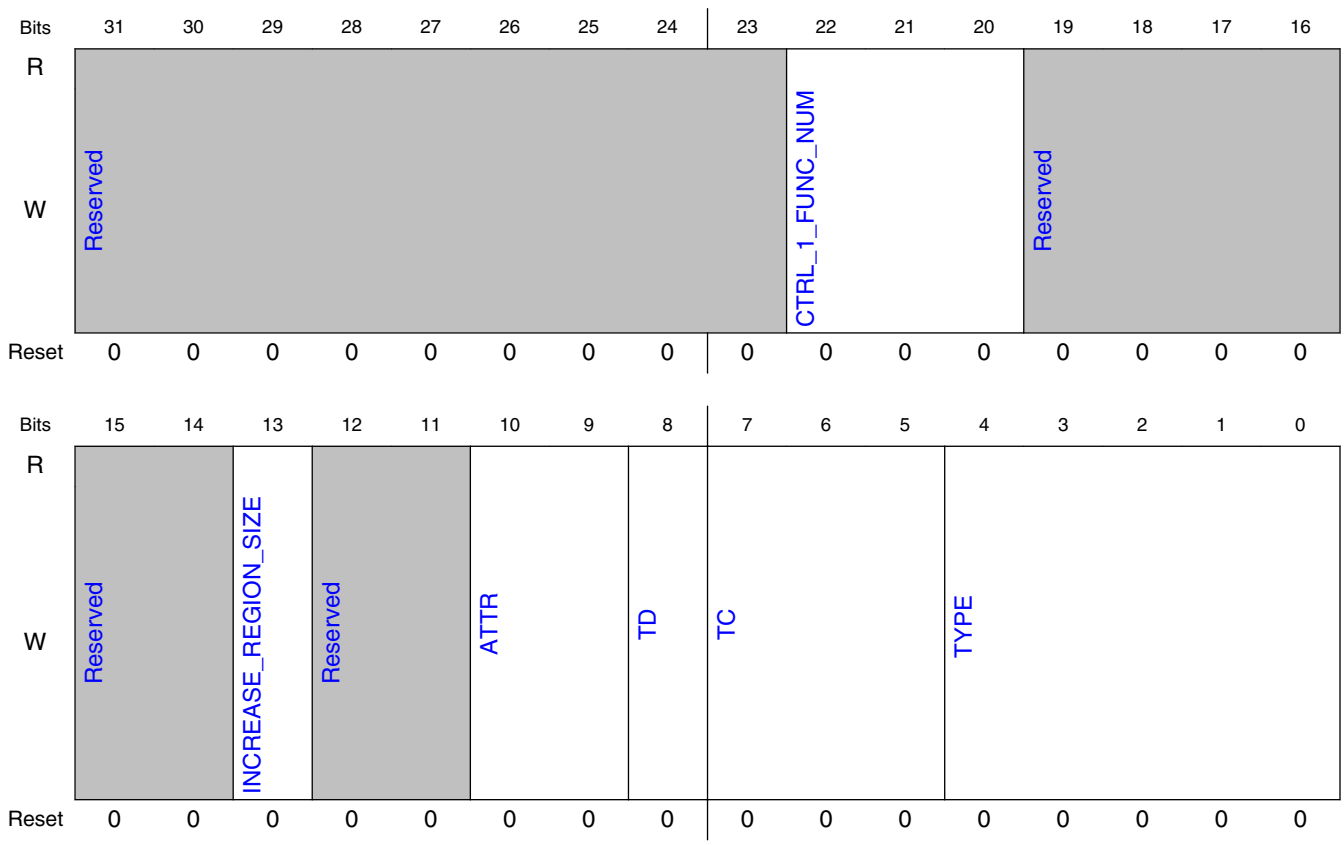
Field	Function
31-0 UPPER_TARGET_RW	Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

11.4.3.1.173 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_3)

11.4.3.1.173.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_INBOUND_3	30_0700h

11.4.3.1.173.2 Diagram



11.4.3.1.173.3 Fields

Field	Function
31-23 —	Reserved
22-20	Function Number. - MEM-I/O: When the Address and BAR matching logic in the controller indicate that a MEM-I/O transaction matches a BAR in the function corresponding to this value, then address translation

Table continues on the next page...

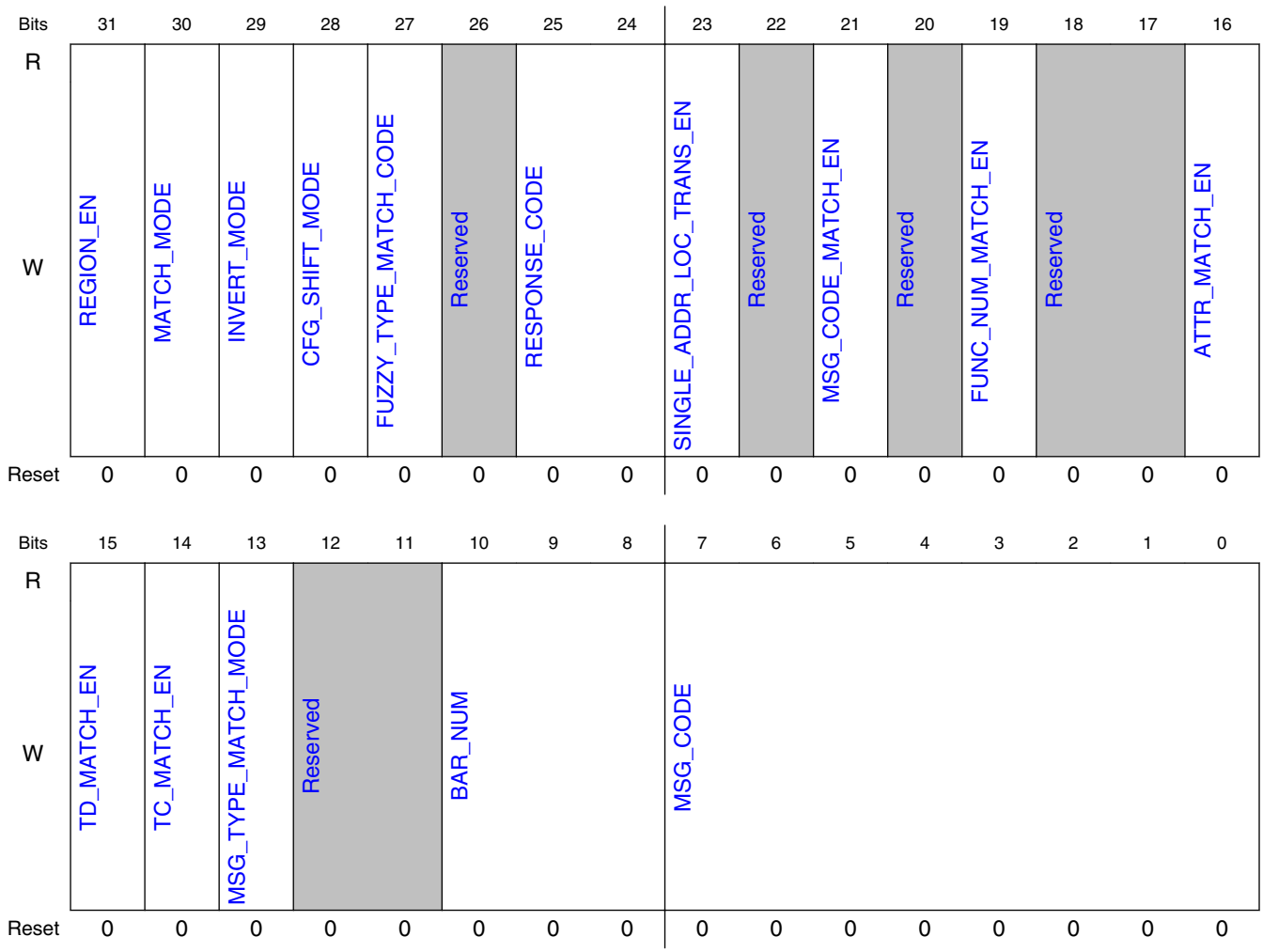
Field	Function
CTRL_1_FUNC_NUM	proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. - CFG0/CFG1: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
19-14 —	Reserved
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by Maximum Size of iATU Region When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved
10-9 ATTR	When the ATTR field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "ATTR Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
8 TD	When the TD field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TD Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
7-5 TC	When the TC field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TC Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
4-0 TYPE	When the TYPE field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). Note: This register field is sticky.

11.4.3.1.174 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_3)

11.4.3.1.174.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_INBOUND_3	30_0704h

11.4.3.1.174.2 Diagram



11.4.3.1.174.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 MATCH_MODE	Match Mode. Determines Inbound matching mode for TLPs. The mode depends on the type of TLP that is received as follows: For MEM-I/O TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU operates using addresses as in the outbound direction. The Region Base and Limit Registers must be setup. - 1:BAR Match Mode. BAR matching is used. The "BAR Number" field is relevant. Not used for RC. For CFG0 TLPs, this field is interpreted as follows: - 0: Routing ID Match Mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM-I/O transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. - 1: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. For MSG/MSGD TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU treats the third dword and fourth dword of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers. - 1: Vendor ID

Table continues on the next page...

Field	Function
	Match Mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the third dword of the TLP header, but matches against the Vendor ID in bits [15:0] of the third dword of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the fourth dword of the TLP header. - If SINGLE_ADDRESS_LOCATION_TRANSLATE_EN = 1 AND MSG_TYPE_MATCH_MODE = 1, then Match Mode is ignored. Note: This register field is sticky.
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of your application memory space using a 28-bit effective address. Shifts bits [31:16] of the untranslated address to form bits [27:12] of the translated address. Note: This register field is sticky.
27 FUZZY_TYPE_MATCH_CODE	Fuzzy Type Match Mode. When enabled, the iATU relaxes the matching of the TLP TYPE field against the expected TYPE field so that - CfgRd0 and CfgRd1 TLPs are seen as identical. Similarly with CfgWr0 and CfgWr1. - MWr, MRd and MRdLk TLPs are seen as identical - The Routing field of Msg/MsgD TLPs is ignored - FetchAdd, Swap and CAS are seen as identical. For example, CFG0 in the TYPE field in the "iATU Control 1 Register" matches against an inbound CfgRd0, CfgRd1, CfgWr0 or CfgWr1 TLP. Note: This register field is sticky.
26 —	Reserved
25-24 RESPONSE_CODE	Response Code. Defines the type of response to give for accesses matching this region. This overrides the normal RADM filter response. Note that this feature is not available for any region where Single Address Location Translate is enabled. - 00 - Normal RADM filter response is used. - 01 - Unsupported request (UR) - 10 - Completer abort (CA) - 11 - Not used / undefined / reserved. Note: This register field is sticky.
23 SINGLE_ADDR_LOC_TRANS_EN	Single Address Location Translate Enable. When enabled, Rx TLPs can be translated to a single address location as determined by the target address register of the iATU region. The main usage scenario is translation of Messages (such as Vendor Defined or ATS Messages) to MWr TLPs when the AXI bridge is enabled. Note: This register field is sticky.
22 —	Reserved
21 MSG_CODE_MATCH_EN	Message Code Match Enable (Msg TLPS). Ensures that a successful message Code TLP field comparison match (see Message Code field of the "iATU Control 2 Register") occurs (in MSG transactions) for address translation to proceed. ST Match Enable (Mem TLPs). Ensures that a successful ST TLP field comparison match (see ST field of the "iATU Control 2 Register") occurs (in MEM transactions) for address translation to proceed. Only Valid when the CX_TPH_ENABLE configuration parameter is 1 Note: This register field is sticky.
20 —	Reserved
19 FUNC_NUM_MATCH_EN	Function Number Match Enable. Ensures that a successful Function Number TLP field comparison match (see Function Number field of the "iATU Control 1 Register") occurs (in MEM-I/O and CFG0/CFG1 transactions) for address translation to proceed. Note: This register field is sticky.
18-17 —	Reserved
16	ATTR Match Enable. Ensures that a successful ATTR TLP field comparison match (see ATTR field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.

Table continues on the next page...

Field	Function
ATTR_MATCH_EN	
15 TD_MATCH_EN	TD Match Enable. Ensures that a successful TD TLP field comparison match (see TD field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
14 TC_MATCH_EN	TC Match Enable. Ensures that a successful TC TLP field comparison match (see TC field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
13 MSG_TYPE_MATCH_MODE	Message Type Match Mode. When enabled, and if single address location translate enable is set, then inbound TLPs of type MSG/MSGd which match the type field of the iatu_region_ctrl_1_OFF_inbound register (=>TYPE[4:3]=2'b10) will be translated. Message type match mode overrides any value of MATCH_MODE field in this register. Usage scenarios for this are translation of VDM or ATS messages when AXI bridge is configured on client interface. Note: This register field is sticky.
12-11 —	Reserved
10-8 BAR_NUM	BAR Number. When the BAR number of an inbound MEM or IO TLP " that is matched by the normal internal BAR address matching mechanism " is the same as this field, address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Match Mode" bit of the "iATU Control 2 Register" is set. - 000b - BAR0 - 001b - BAR1 - 010b - BAR2 - 011b - BAR3 - 100b - BAR4 - 101b - BAR5 - 110b - ROM - 111b - reserved - IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b - 101b and that BAR configured as an IO BAR. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs: (Message Code). When the TYPE field of an inbound Msg/MsgD TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Message Code Match Enable" bit of the "iATU Control 2 Register" is set. Memory TLPs: (ST;Steering Tag). When the ST field of an inbound TLP is matched to this value, then address translation proceeds. This check is only performed if the "ST Match Enable" bit of the "iATU Control2 Register" is set. The setting is independent of the setting of the TH field. Only Valid when the CX_TPH_ENABLE configuration parameter is 1. Note: This register field is sticky.

11.4.3.1.175 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_3)

11.4.3.1.175.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_INBOUND_3	30_0708h

11.4.3.1.175.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned

on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.4.3.1.175.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_BASE_HW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.175.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.4.3.1.176 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_3)

11.4.3.1.176.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_INBOUND_3	30_070Ch

11.4.3.1.176.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.176.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. Note: This register field is sticky.

11.4.3.1.177 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_3)

11.4.3.1.177.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_INBOUND_3	30_0710h

11.4.3.1.177.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.4.3.1.177.3 Fields

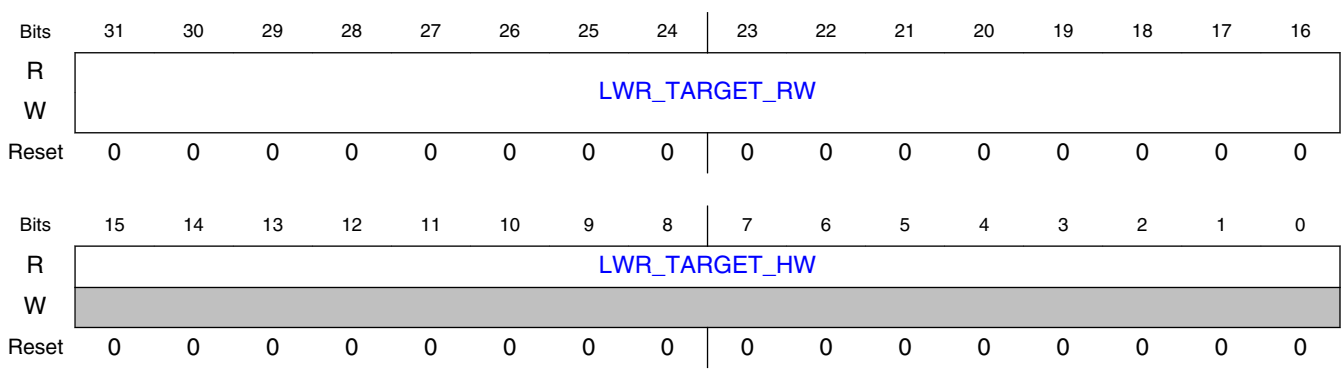
Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.4.3.1.178 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_3)

11.4.3.1.178.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_INBOUND_3	30_0714h

11.4.3.1.178.2 Diagram



11.4.3.1.178.3 Fields

Field	Function
31-16 LWR_TARGET_RW	Forms MSB's of the Lower Target part of the new address of the translated region. These bits are always '0'. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode. Note: This register field is sticky.

Table continues on the next page...

Field	Function
15-0 LWR_TARGET_HW	Forms the LSB's of the Lower Target part of the new address of the translated region. The start address must be aligned to a Minimum Size of iATU Region kB boundary (in address match mode); and to the Bar size boundary (in BAR match mode) so that these bits are always '0'. If the BAR is smaller than the iATU region size, then the iATU target address must align to the iATU region size; otherwise it must align to the BAR size. A write to this location is ignored by the PCIe controller. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode.

11.4.3.1.179 DMA Arbitration Scheme for TRGT1 Interface. (DMA_CTRL_DATA_ARB_PRIOR_OFF)

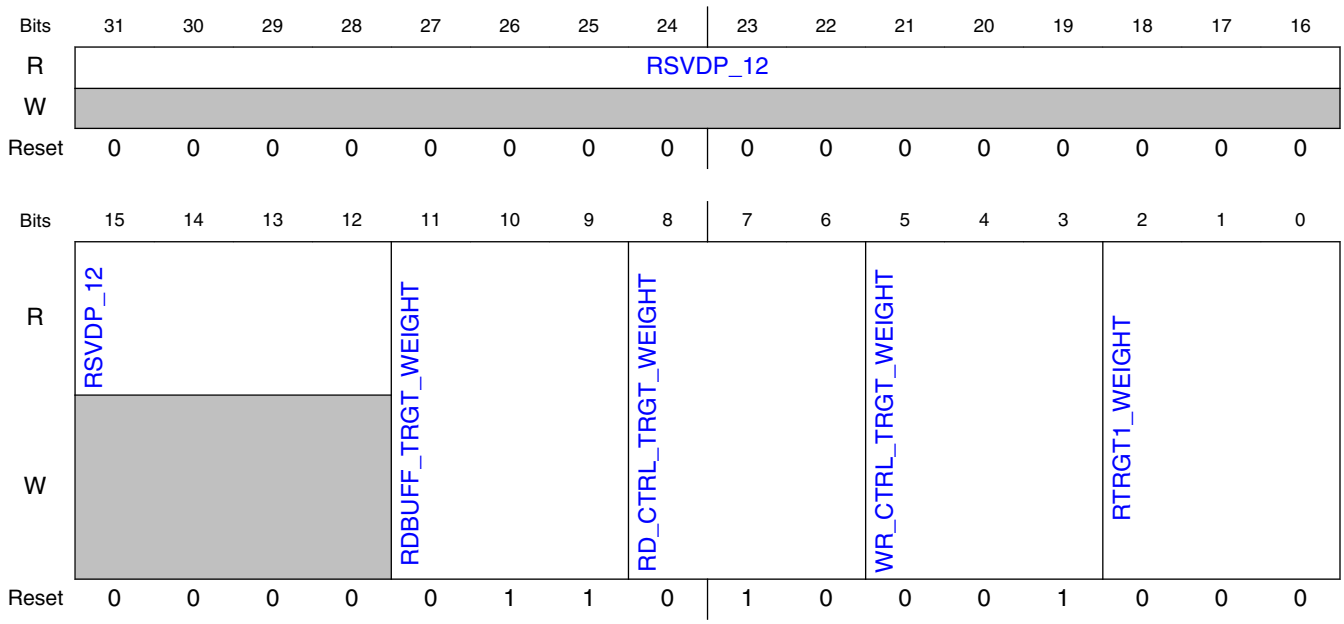
11.4.3.1.179.1 Offset

Register	Offset
DMA_CTRL_DATA_ARB_PRIOR_OFF	38_0000h

11.4.3.1.179.2 Function

DMA Arbitration Scheme for TRGT1 Interface. This register is used to control traffic priorities among various sources that are delivered to your application through TRGT1 where 0x0 represents the highest priority. - Non-DMA Rx Requests - DMA Write Channel MRd Requests (DMA data requests and LL element/descriptor access) - DMA Read Channel MRd Requests (LL element/descriptor access) - DMA Read Channel MWr Requests Concurrent traffic from channels with same priority are sorted according to Round-Robin arbitration rules. The arbitration priority defaults to Non-DMA requests (highest), Write Channel MRd, Read Channel MRd, Read Channel MWr. For more details, see the For more details, see the Internal Architecture section in the DMA chapter of the Databook.

11.4.3.1.179.3 Diagram



11.4.3.1.179.4 Fields

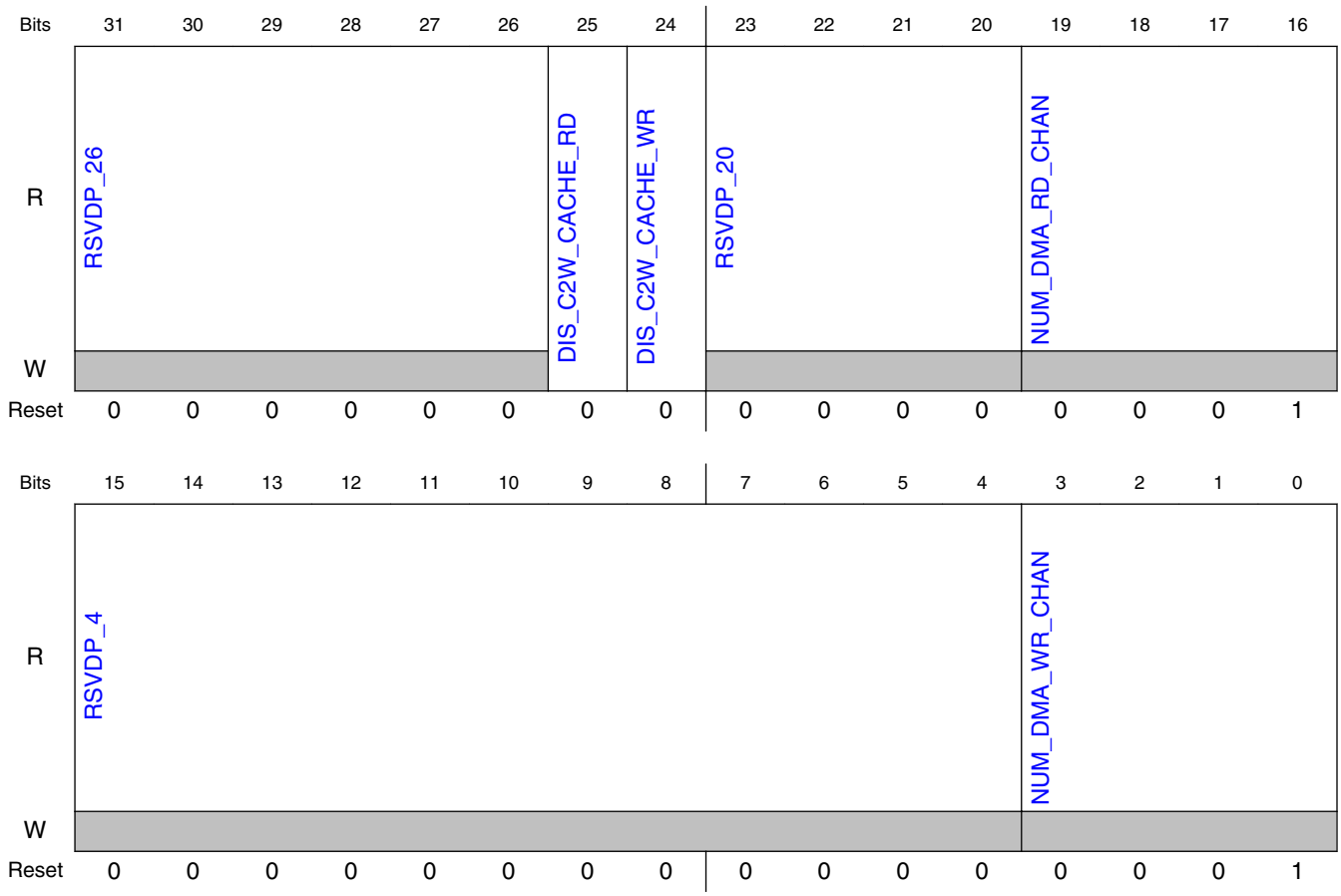
Field	Function
31-12 RSVDP_12	Reserved for future use.
11-9 RDBUFF_TRGT_WEIGHT	DMA Read Channel MWr Requests. Note: The access attributes of this field are as follows: - Dbi: R/W
8-6 RD_CTRL_TRGT_WEIGHT	DMA Read Channel MRd Requests. For LL element/descriptor access. Note: The access attributes of this field are as follows: - Dbi: R/W
5-3 WR_CTRL_TRGT_WEIGHT	DMA Write Channel MRd Requests. For DMA data requests and LL element/descriptor access. Note: The access attributes of this field are as follows: - Dbi: R/W
2-0 RTRGT1_WEIGHT	Non-DMA Rx Requests. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.180 DMA Number of Channels Register. (DMA_CTRL_OFF)

11.4.3.1.180.1 Offset

Register	Offset
DMA_CTRL_OFF	38_0008h

11.4.3.1.180.2 Diagram



11.4.3.1.180.3 Fields

Field	Function
31-26 RSVDP_26	Reserved for future use.
25 DIS_C2W_CACHE_RD	Disable DMA Read Channels "completion to memory write" context cache pre-fetch function. Note: For internal debugging only. Note: The access attributes of this field are as follows: - Dbi: R/W
24	Disable DMA Write Channels "completion to memory write" context cache pre-fetch function. Note: For internal debugging only. Note: The access attributes of this field are as follows: - Dbi: R/W

Table continues on the next page...

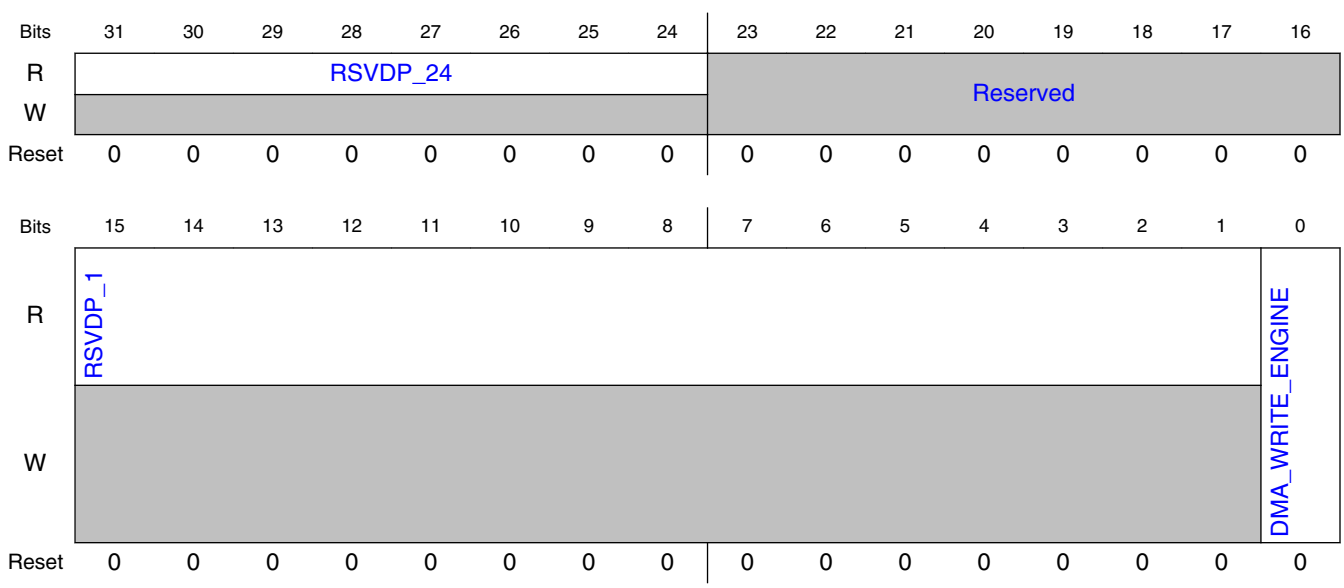
Field	Function
DIS_C2W_CACHE_WR	
23-20 RSVDP_20	Reserved for future use.
19-16 NUM_DMA_RD_CHAN	Number of Read Channels. You can read this register to determine the number of read channels the DMA controller has been configured to support.
15-4 RSVDP_4	Reserved for future use.
3-0 NUM_DMA_WR_CHAN	Number of Write Channels. You can read this register to determine the number of write channels the DMA controller has been configured to support.

11.4.3.1.181 DMA Write Engine Enable Register. (DMA_WRITE_ENGINE_EN_OFF)

11.4.3.1.181.1 Offset

Register	Offset
DMA_WRITE_ENGINE_EN_OFF	38_000Ch

11.4.3.1.181.2 Diagram



11.4.3.1.181.3 Fields

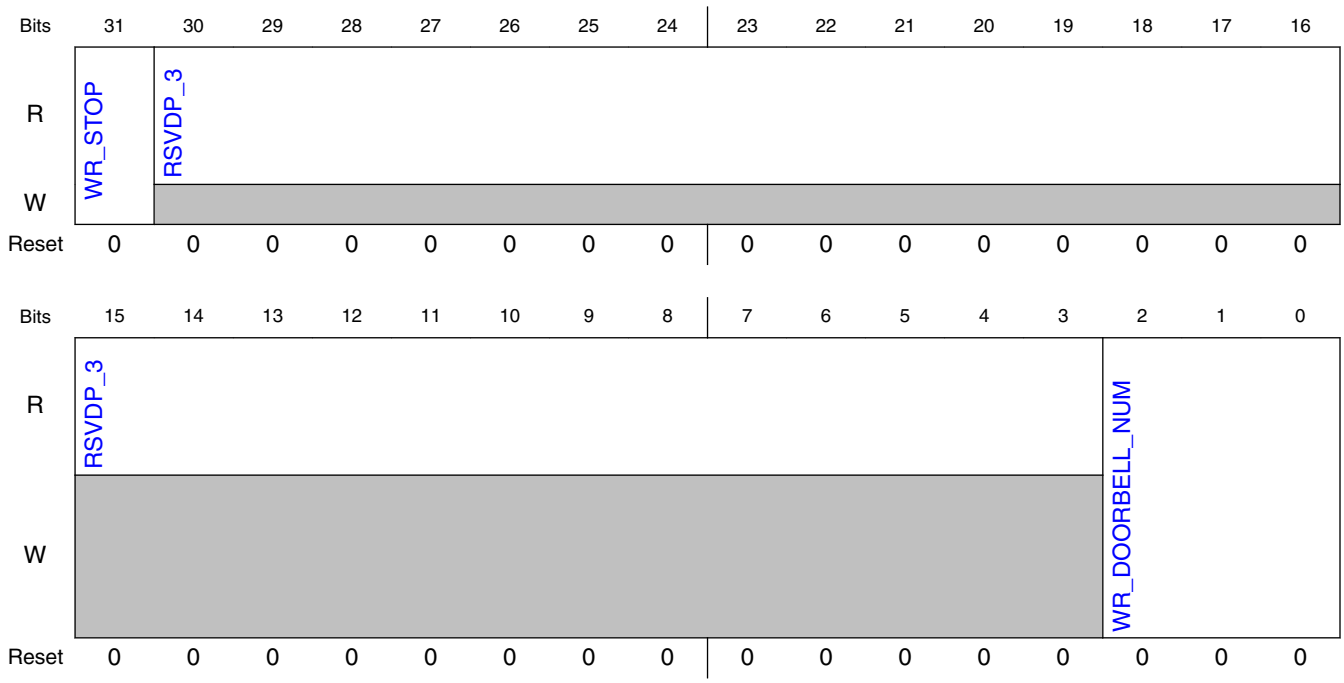
Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16 —	Reserved
15-1 RSVDP_1	Reserved for future use.
0 DMA_WRITE_ENGINE	DMA Write Engine Enable. - 1: Enable - 0: Disable (Soft Reset) For normal operation, you must initially set this bit to "1", before any other software setup actions. You do not need to toggle or rewrite to this bit during normal operation. You should set this bit to "0" when you want to "Soft Reset" the DMA controller write logic. There are three possible reasons for resetting the DMA controller write logic: - The "Abort Interrupt Status" bit is set (in the "DMA Write Interrupt Status Register" DMA_WRITE_INT_STATUS_OFF), and any of the bits is in the "DMA Write Error Status Register" (DMA_WRITE_ERR_STATUS_OFF) are set. Resetting the DMA controller write logic re-initializes the control logic, ensuring that the next DMA write transfer is executed successfully. - You have executed the procedure outlined in "Stop Bit" , after which, the "Abort Interrupt Status" bit is set and the Channel Status field (CS) of the DMA write "DMA Channel Control 1 Register " (DMA_CH_CONTROL1_OFF_WRCH_0) is set to "Stopped." Resetting the DMA controller write logic re-initializes the control logic ensuring that the next DMA write transfer is executed successfully. - During software development, when you incorrectly program the DMA write engine. To "Soft Reset" the DMA controller write logic, you must: - De-assert the DMA write engine enable bit. - Wait for the DMA to complete any in-progress TLP transfer, by waiting until a read on the DMA write engine enable bit returns a "0". - Assert the DMA write engine enable bit. This "Soft Reset" does not clear the DMA configuration registers. The DMA write transfer does not start until you write to the "DMA Write Doorbell Register" (DMA_WRITE_DOORBELL_OFF). Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.182 DMA Write Doorbell Register. (DMA_WRITE_DOORBELL_OFF)

11.4.3.1.182.1 Offset

Register	Offset
DMA_WRITE_DOORBELL_OFF	38_0010h

11.4.3.1.182.2 Diagram



11.4.3.1.182.3 Fields

Field	Function
31 WR_STOP	Stop. Set in conjunction with the Doorbell Number field. The DMA write channel stops issuing requests, sets the channel status to "Stopped", and asserts the "Abort" interrupt if it is enabled. Before setting the Stop bit, you must read the channel Status field (CS) of the "DMA Channel Control 1 Register " (DMA_CH_CONTROL1_OFF_WRCH_0) to ensure that the write channel is "Running" (transferring data). For more information, see "Stopping the DMA Transfer (Software Stop)." Note: The access attributes of this field are as follows: - Dbi: R/W
30-3 RSVDP_3	Reserved for future use.
2-0 WR_DOORBELL_NUM	Doorbell Number. You must write the channel number to this register to start the DMA write transfer for that channel. The DMA detects a write to this register field even if the value of this field does not change. You do not need to toggle or write any other value to this register to start a new transfer. The range of this field is 0x0 to 0x7, and 0x0 corresponds to channel 0. Also note that a write to this field triggers the controller to exit L1 substates. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.183 DMA Write Engine Channel Arbitration Weight Low Register. (DMA_WRITE_CHANNEL_ARB_WEIGHT_LOW_OFF)

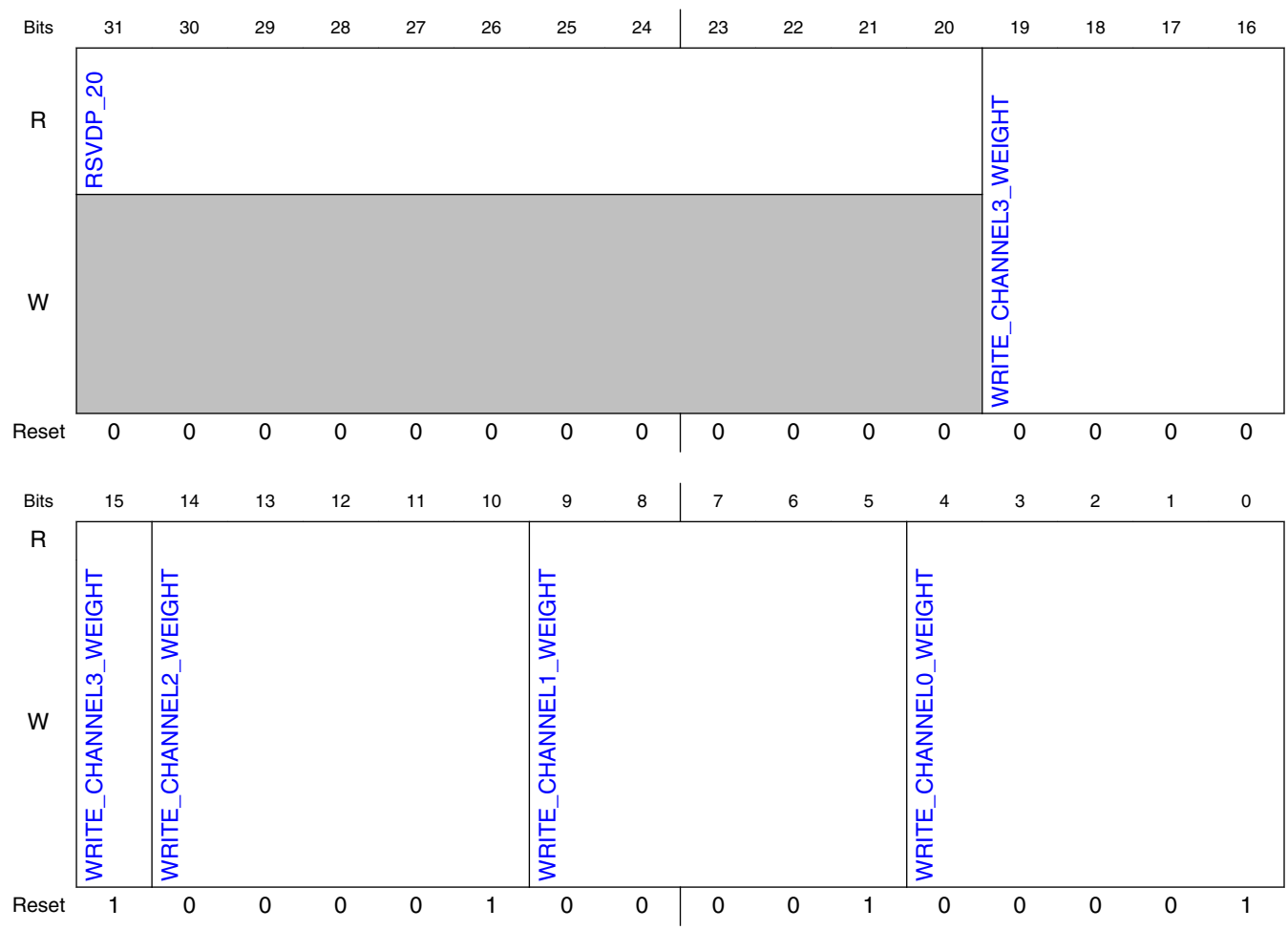
11.4.3.1.183.1 Offset

Register	Offset
DMA_WRITE_CHANNEL_ARB_WEIGHT_LOW_OFF	38_0018h

11.4.3.1.183.2 Function

DMA Write Engine Channel Arbitration Weight Low Register. The 5-bit channel weight (for write channels 0-3) specifies the maximum number of TLP requests that the DMA can issue for that channel before it must return to the arbitration routine. When the channel weight count is reached or DMA channel request transfer size reaches zero, the WWR arbiter selects the next channel to be processed. Your software must initialize this register before ringing the doorbell. For more details, see "Multichannel Arbitration". Value range is (0-0x1F) corresponding to (1-32) transaction requests.

11.4.3.1.183.3 Diagram



11.4.3.1.183.4 Fields

Field	Function
31-20 RSVDP_20	Reserved for future use.
19-15 WRITE_CHANNEL3_WEIGHT	Channel 3 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W
14-10 WRITE_CHANNEL2_WEIGHT	Channel 2 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W
9-5 WRITE_CHANNEL1_WEIGHT	Channel 1 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W
4-0 WRITE_CHANNEL0_WEIGHT	Channel 0 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.184 DMA Write Engine Channel Arbitration Weight High Register. (DMA_WRITE_CHANNEL_ARB_WEIGHT_HIGH_OFF)

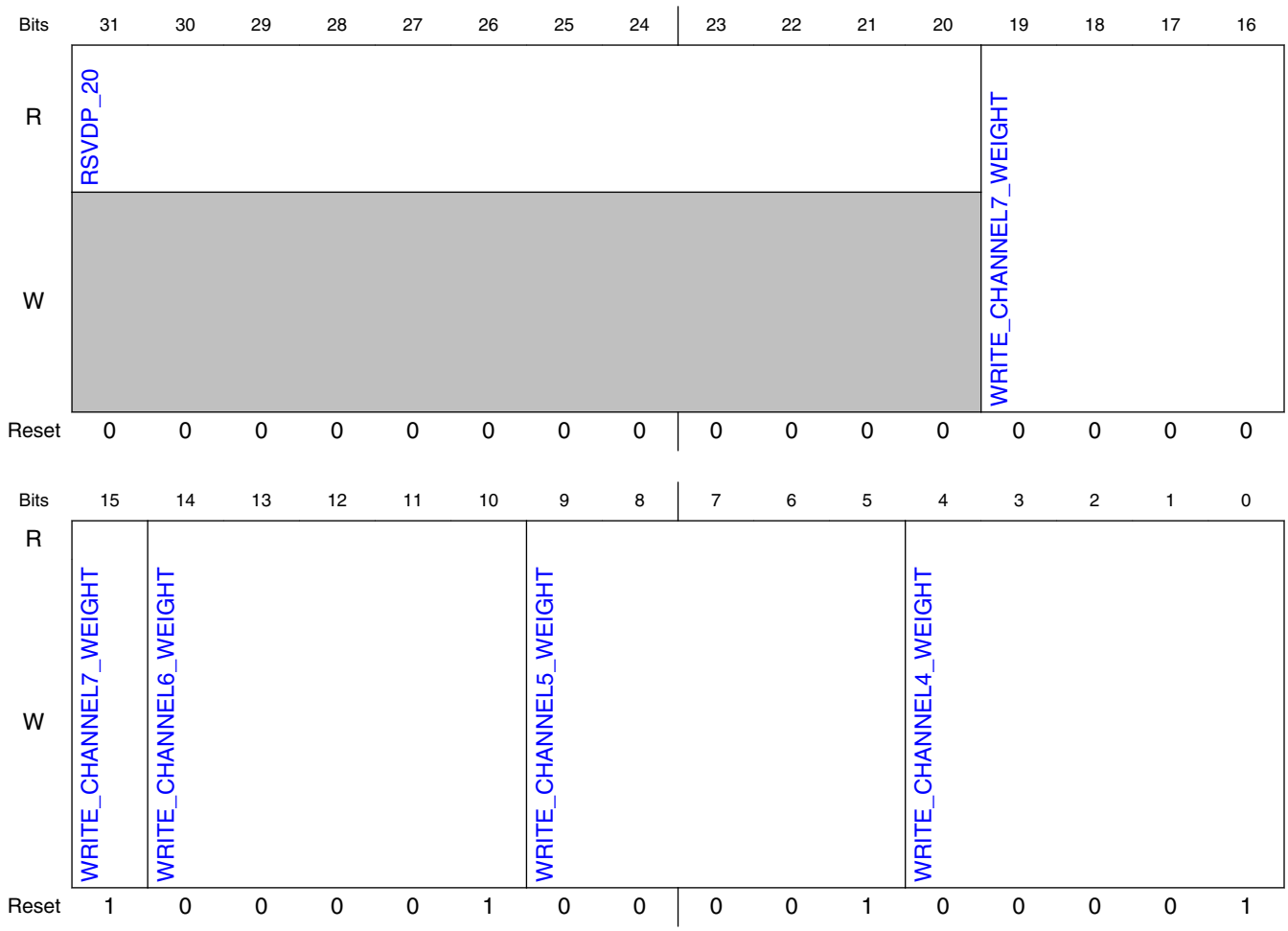
11.4.3.1.184.1 Offset

Register	Offset
DMA_WRITE_CHANNEL_ARB_WEIGHT_HIGH_OFF	38_001Ch

11.4.3.1.184.2 Function

DMA Write Engine Channel Arbitration Weight High Register. The 5-bit channel weight (for write channels 4-7) specifies the maximum number of TLP requests that the DMA can issue for that channel before it must return to the arbitration routine. When the channel weight count is reached or DMA channel request transfer size reaches zero, the WWR arbiter selects the next channel to be processed. Your software must initialize this register before ringing the doorbell. For more details, see "Multichannel Arbitration". Value range is (0-0x1F) corresponding to (1-32) transaction requests.

11.4.3.1.184.3 Diagram



11.4.3.1.184.4 Fields

Field	Function
31-20 RSVDP_20	Reserved for future use.
19-15 WRITE_CHANNEL7_WEIGHT	Channel 7 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W
14-10 WRITE_CHANNEL6_WEIGHT	Channel 6 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W
9-5 WRITE_CHANNEL5_WEIGHT	Channel 5 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W

Table continues on the next page...

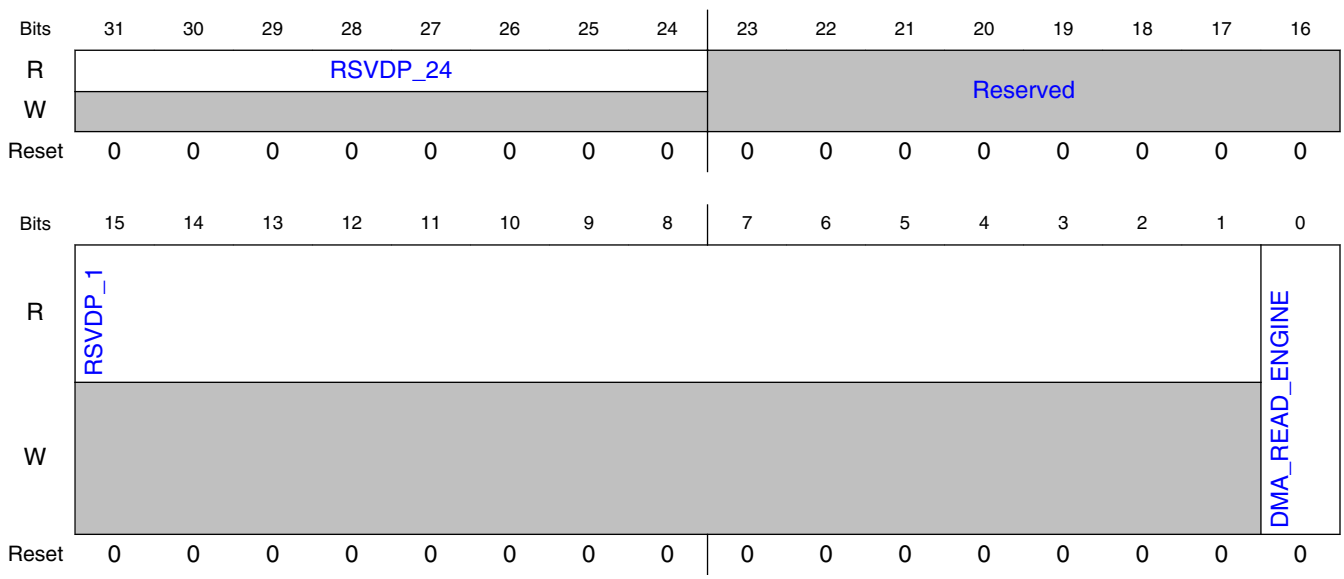
Field	Function
4-0 WRITE_CHANNEL4_WEIGHT	Channel 4 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.185 DMA Read Engine Enable Register. (DMA_READ_ENGINE_EN_OFF)

11.4.3.1.185.1 Offset

Register	Offset
DMA_READ_ENGINE_EN_OFF	38_002Ch

11.4.3.1.185.2 Diagram



11.4.3.1.185.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16	Reserved

Table continues on the next page...

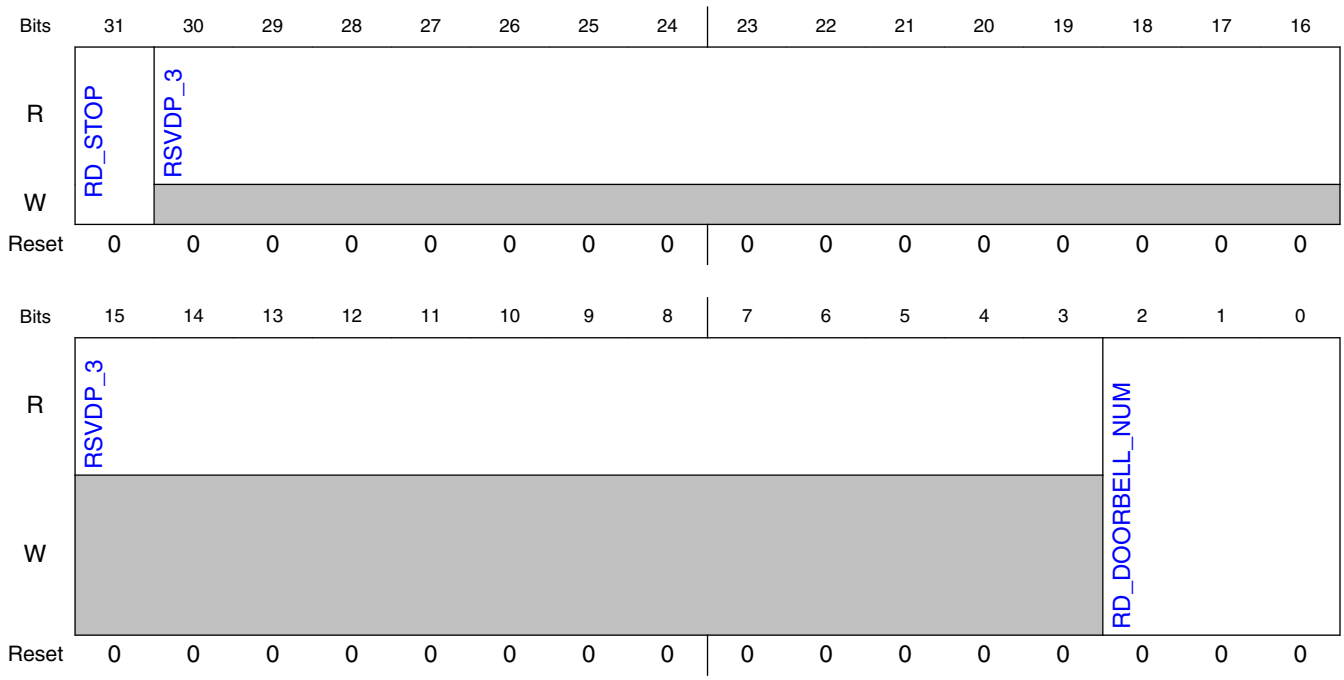
Field	Function
—	
15-1 RSVDP_1	Reserved for future use.
0 DMA_READ_ENGINE	DMA Read Engine Enable. - 1: Enable - 0: Disable (Soft Reset) For normal operation, you must initially set this bit to "1", before any other software setup actions. You do not need to toggle or rewrite to this bit during normal operation. You should set this field to "0" when you want to "Soft Reset" the DMA controller read logic. There are three possible reasons for resetting the DMA controller read logic: - The "Abort Interrupt Status" bit is set (in the "DMA Read Interrupt Status Register" (DMA_READ_INT_STATUS_OFF), and any of the bits in the "DMA Read Error Status Low Register" (DMA_READ_ERR_STATUS_LOW_OFF) is set. Resetting the DMA controller read logic re-initializes the control logic, ensuring that the next DMA read transfer is executed successfully. - You have executed the procedure outlined in "Stop Bit", after which, the "Abort Interrupt Status" bit is set and the channel Status field (CS) of the DMA read "DMA Channel Control 1 Register " (DMA_CH_CONTROL1_OFF_WRCH_0) is set to "Stopped". Resetting the DMA controller read logic re-initializes the control logic ensuring that the next DMA read transfer is executed successfully. - During software development, when you incorrectly program the DMA read engine. To "Soft Reset" the DMA controller read logic, you must: - De-assert the DMA read engine enable bit. - Wait for the DMA to complete any in-progress TLP transfer, by waiting until a read on the DMA read engine enable bit returns a "0". - Assert the DMA read engine enable bit. This "Soft Reset" does not clear the DMA configuration registers. The DMA read transfer does not start until you write to the "DMA Read Doorbell Register" (DMA_READ_DOORBELL_OFF). Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.186 DMA Read Doorbell Register. (DMA_READ_DOORBELL_OFF)

11.4.3.1.186.1 Offset

Register	Offset
DMA_READ_DOORBELL_OFF	38_0030h

11.4.3.1.186.2 Diagram



11.4.3.1.186.3 Fields

Field	Function
31 RD_STOP	Stop. Set in conjunction with the Doorbell Number field. The DMA read channel stops issuing requests, sets the channel status to "Stopped", and asserts the "Abort" interrupt if it is enabled. Before setting the Stop bit, you must read the channel Status field (CS) of the "DMA Channel Control 1 Register " (DMA_CH_CONTROL1_OFF_RDCH_0) to ensure that the read channel is "Running" (transferring data). For more information, see "Stopping the DMA Transfer (Software Stop)". Note: The access attributes of this field are as follows: - Dbi: R/W
30-3 RSVDP_3	Reserved for future use.
2-0 RD_DOORBELL_NUM	Doorbell Number. You must write 0x0 to this register to start the DMA read transfer for that channel. The DMA detects a write to this register field even if the value of this field does not change. The range of this field is 0x0 to 0x7, and 0x0 corresponds to channel 0. Also note that a write to this field triggers the controller to exit L1 substates. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.187 DMA Read Engine Channel Arbitration Weight Low Register. (DMA_READ_CHANNEL_ARB_WEIGHT_LOW_OFF)

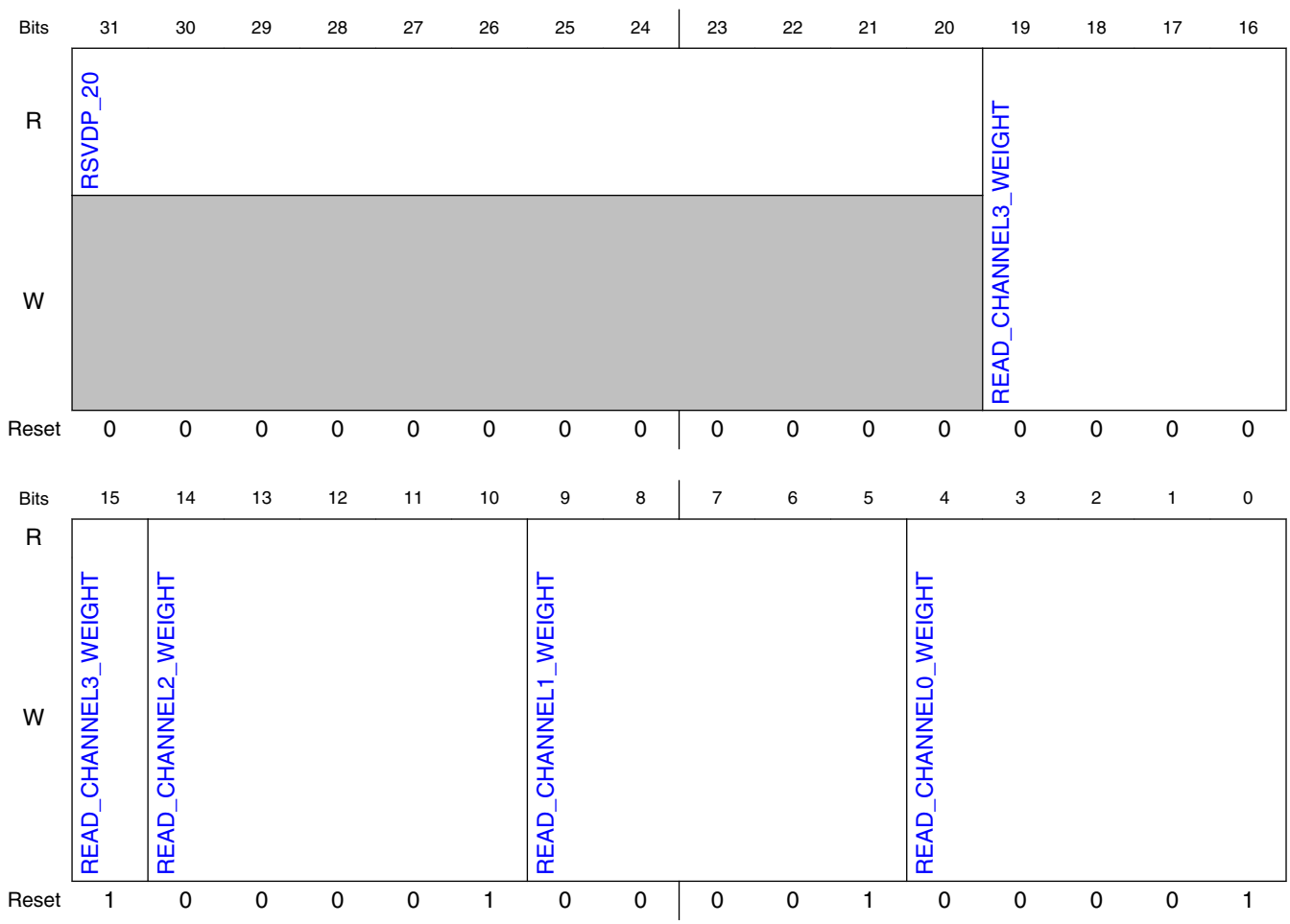
11.4.3.1.187.1 Offset

Register	Offset
DMA_READ_CHANNEL_ARB_WEIGHT_LOW_OFF	38_0038h

11.4.3.1.187.2 Function

DMA Read Engine Channel Arbitration Weight Low Register. The 5-bit channel weight (for read channels 0-3) specifies the maximum number of TLP requests that the DMA can issue for that channel before it must return to the arbitration routine. When the channel weight count is reached or DMA channel request transfer size reaches zero, the WWR arbiter selects the next channel to be processed. Your software must initialize this register before ringing the doorbell. For more details, see "Multichannel Arbitration". Value range is (0-0x1F) corresponding to (1-32) transaction requests.

11.4.3.1.187.3 Diagram



11.4.3.1.187.4 Fields

Field	Function
31-20 RSVDP_20	Reserved for future use.
19-15 READ_CHANNEL3_WEIGHT	Channel 3 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W
14-10 READ_CHANNEL2_WEIGHT	Channel 2 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W
9-5 READ_CHANNEL1_WEIGHT	Channel 1 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W
4-0 READ_CHANNEL0_WEIGHT	Channel 0 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.188 DMA Read Engine Channel Arbitration Weight High Register. (DMA_READ_CHANNEL_ARB_WEIGHT_HIGH_OFF)

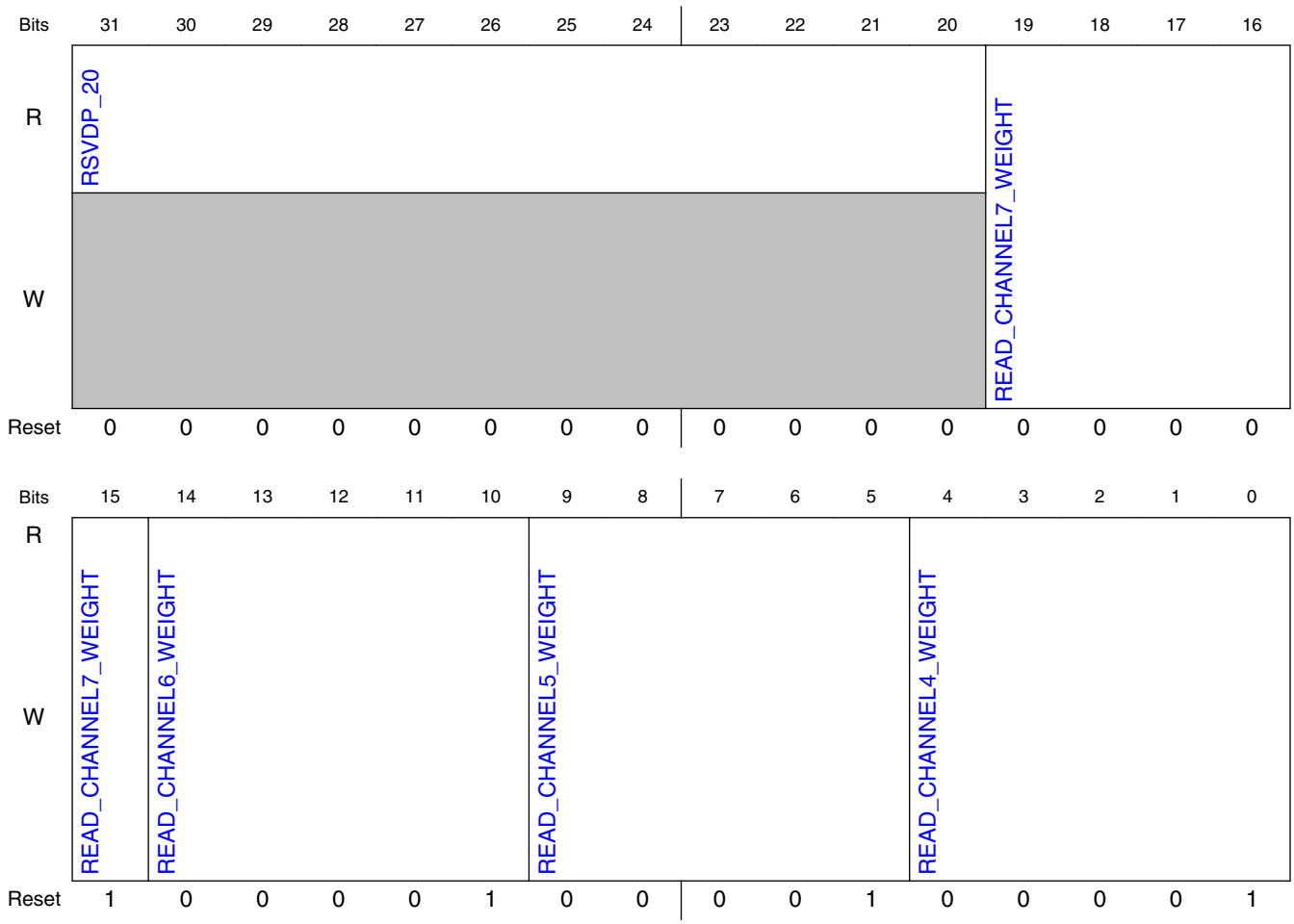
11.4.3.1.188.1 Offset

Register	Offset
DMA_READ_CHANNEL_ARB_WEIGHT_HIGH_OFF	38_003Ch

11.4.3.1.188.2 Function

DMA Read Engine Channel Arbitration Weight High Register. The 5-bit channel weight (for read channels 4-7) specifies the maximum number of TLP requests that the DMA can issue for that channel before it must return to the arbitration routine. When the channel weight count is reached or DMA channel request transfer size reaches zero, the WWR arbiter selects the next channel to be processed. Your software must initialize this register before ringing the doorbell. For more details, see "Multichannel Arbitration". Value range is (0-0x1F) corresponding to (1-32) transaction requests.

11.4.3.1.188.3 Diagram



11.4.3.1.188.4 Fields

Field	Function
31-20 RSVDP_20	Reserved for future use.
19-15 READ_CHANNEL7_WEIGHT	Channel 7 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W
14-10 READ_CHANNEL6_WEIGHT	Channel 6 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W
9-5 READ_CHANNEL5_WEIGHT	Channel 5 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W

Table continues on the next page...

Field	Function
4-0 READ_CHANNEL4_WEIGHT	Channel 4 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.189 DMA Write Interrupt Status Register. (DMA_WRITE_INT_STATUS_OFF)

11.4.3.1.189.1 Offset

Register	Offset
DMA_WRITE_INT_STATUS_OFF	38_004Ch

11.4.3.1.189.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVDP_24								WR_ABORT_INT_STATUS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVDP_8								WR_DONE_INT_STATUS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.189.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16 WR_ABORT_INT_STATUS	Abort Interrupt Status. The DMA write channel has detected an error, or you manually stopped the transfer as described in "Error Handling Assistance by Remote Software". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA write interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the DMA write interrupt Clear register to clear this interrupt bit. Note: You can write to this register to emulate interrupt generation, during software or hardware testing. A write to the address triggers an interrupt, but the DMA does not set the Done or Abort bits in this register. Note: The access attributes of this field are as follows: - Dbi: R/W
15-8	Reserved for future use.

Table continues on the next page...

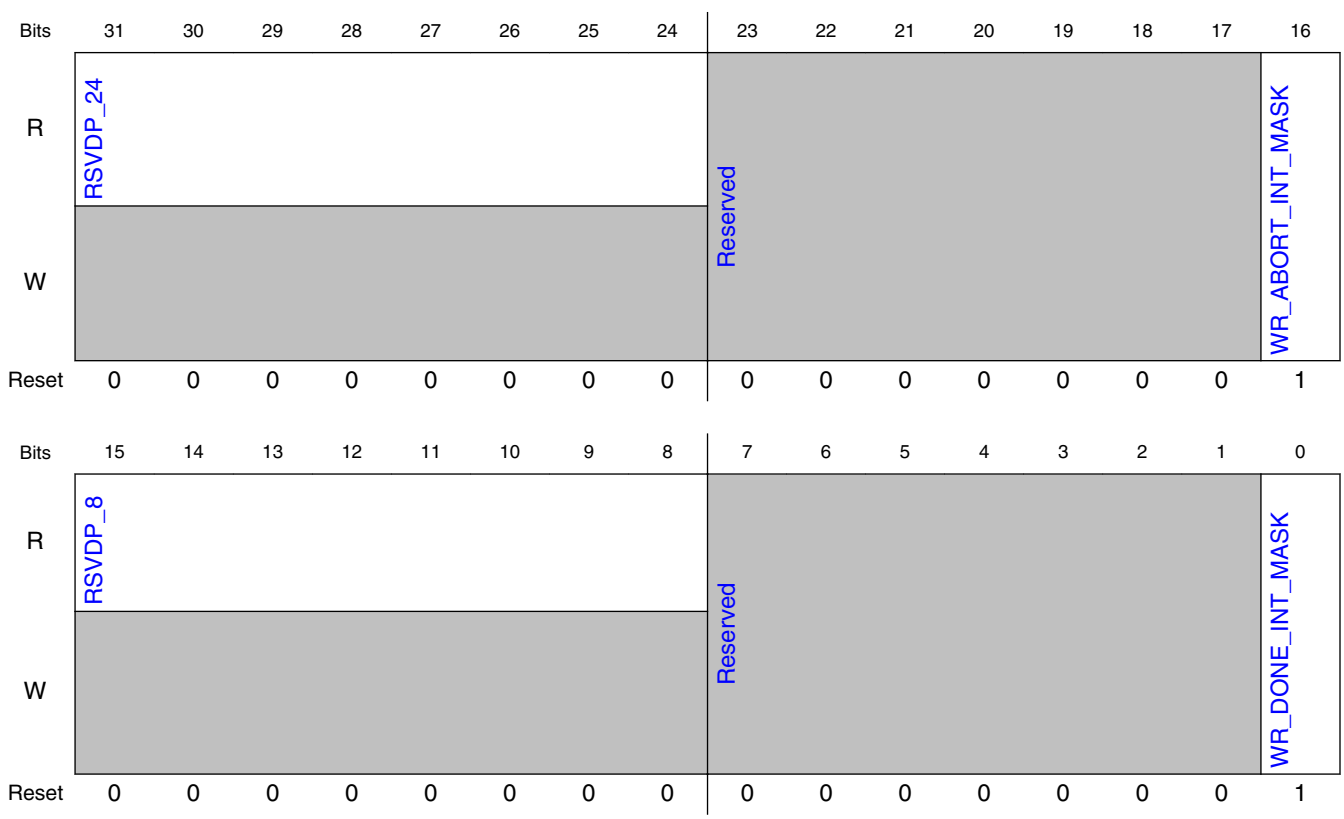
Field	Function
RSVDP_8	
7-0 WR_DONE_INT_STATUS	Done Interrupt Status. The DMA write channel has successfully completed the DMA transfer. For more details, see "Interrupts and Error Handling". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA write interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the DMA write interrupt Clear register to clear this interrupt bit. Note: You can write to this register to emulate interrupt generation, during software or hardware testing. A write to the address triggers an interrupt, but the DMA does not set the Done or Abort bits in this register. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.190 DMA Write Interrupt Mask Register. (DMA_WRITE_INT_MASK_OFF)

11.4.3.1.190.1 Offset

Register	Offset
DMA_WRITE_INT_MASK_OFF	38_0054h

11.4.3.1.190.2 Diagram



11.4.3.1.190.3 Fields

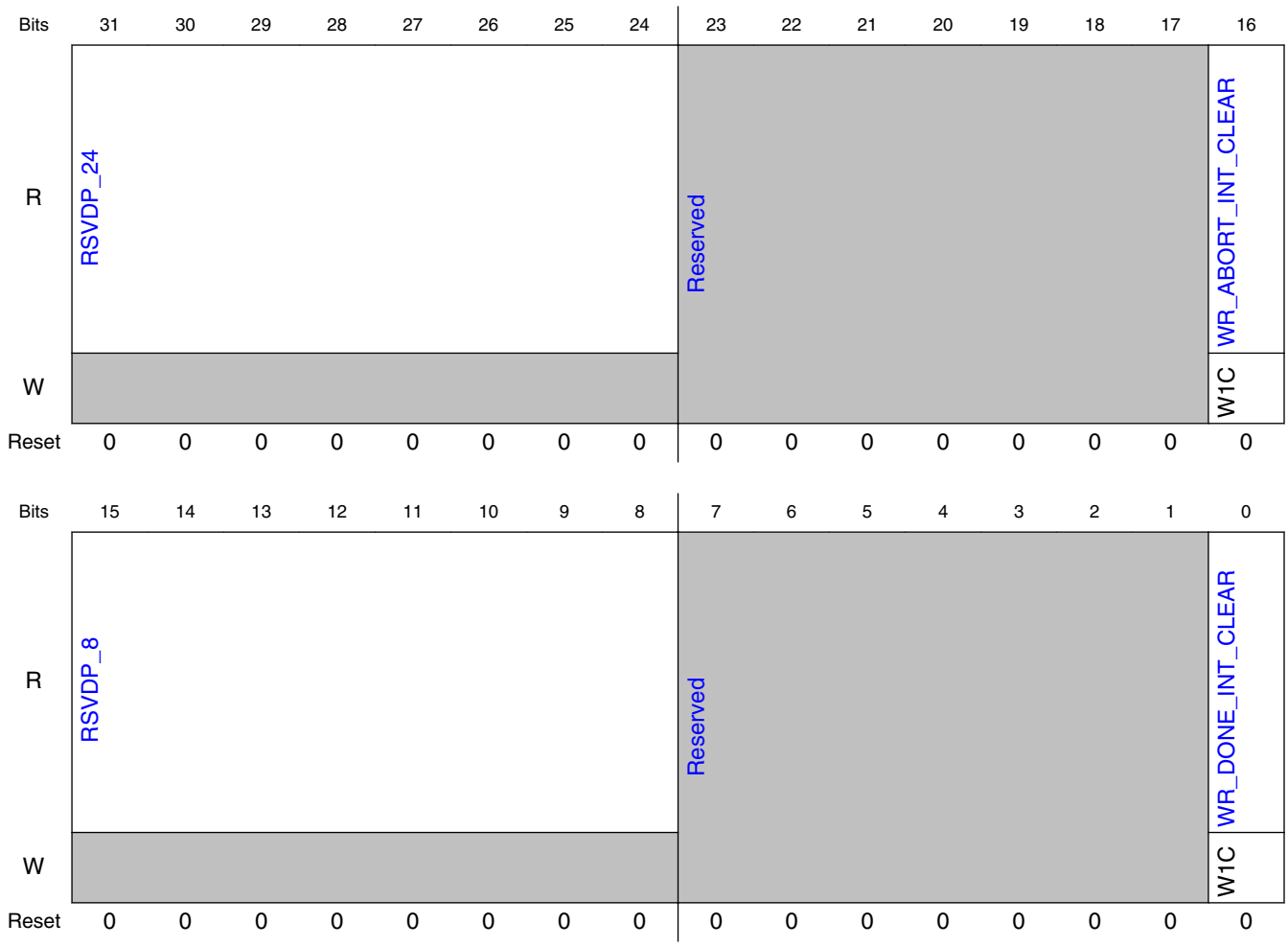
Field	Function
31-24 RSVDP_24	Reserved for future use.
23-17 —	Reserved
16 WR_ABORT_INT_MASK	Abort Interrupt Mask. Prevents the Abort interrupt status field in the DMA write interrupt status register from asserting the edma_int output. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: The access attributes of this field are as follows: - Dbi: R/W
15-8 RSVDP_8	Reserved for future use.
7-1 —	Reserved
0 WR_DONE_INT_MASK	Done Interrupt Mask. Prevents the Done interrupt status field in the DMA write interrupt status register from asserting the edma_int output. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.191 DMA Write Interrupt Clear Register. (DMA_WRITE_INT_CLEAR_OFF)

11.4.3.1.191.1 Offset

Register	Offset
DMA_WRITE_INT_CLEAR_OFF	38_0058h

11.4.3.1.191.2 Diagram



11.4.3.1.191.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-17 —	Reserved
16 WR_ABORT_INT_CLEAR	Abort Interrupt Clear. You must write a 1'b1 to clear the corresponding bit in the Abort interrupt status field of the DMA write interrupt status register. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: Reading from this self-clearing register field always returns a "0".
15-8 RSVDP_8	Reserved for future use.
7-1 —	Reserved

Table continues on the next page...

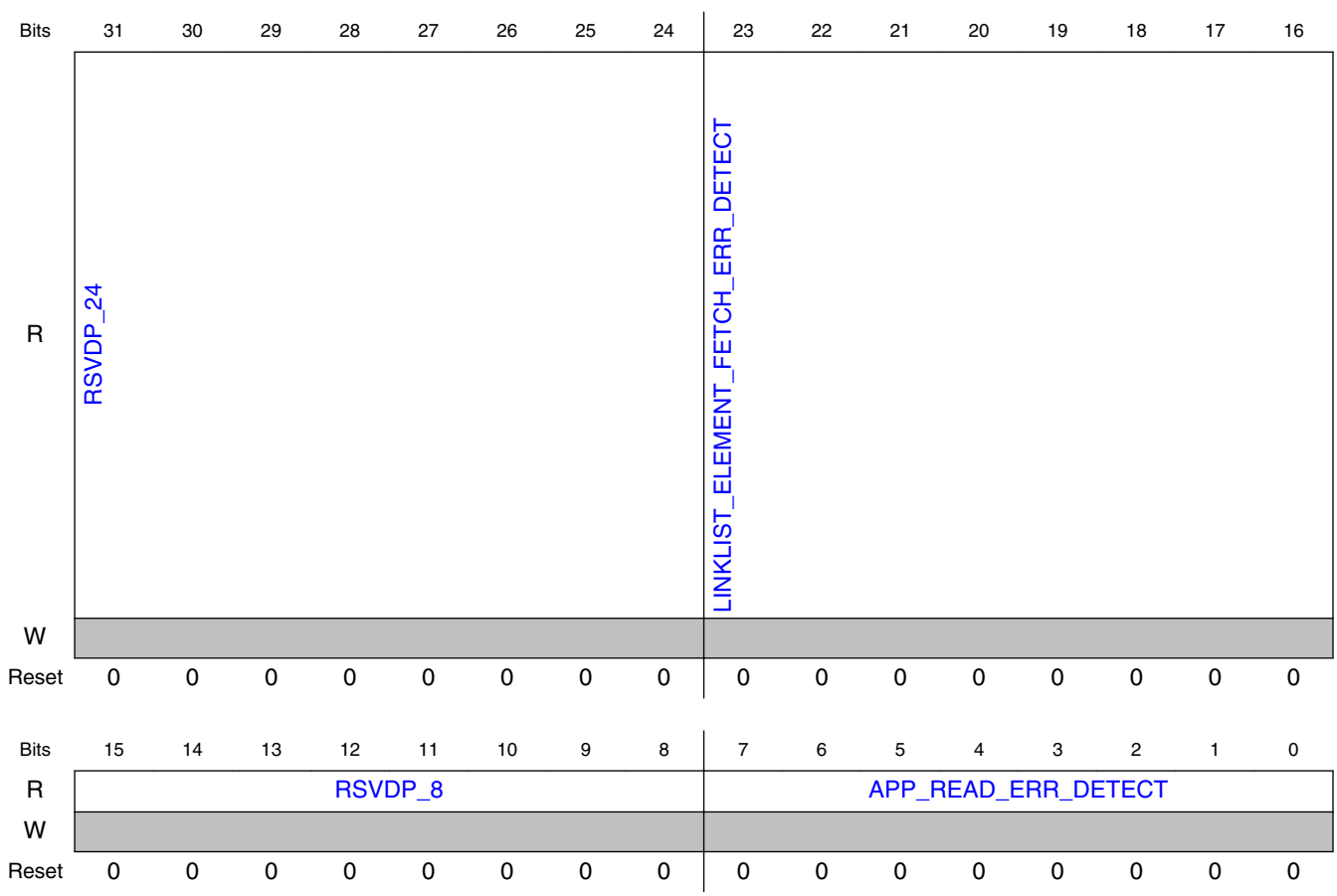
Field	Function
0 WR_DONE_INT_CLEAR	Done Interrupt Clear. You must write a 1'b1 to clear the corresponding bit in the Done interrupt status field of the DMA write interrupt status register. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: Reading from this self-clearing register field always returns a "0".

11.4.3.1.192 DMA Write Error Status Register (DMA_WRITE_ERR_STATUS_OFF)

11.4.3.1.192.1 Offset

Register	Offset
DMA_WRITE_ERR_STATUS_OFF	38_005Ch

11.4.3.1.192.2 Diagram



11.4.3.1.192.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16 LINKLIST_ELEMENT_FETCH_ERR_DETECT	Linked List Element Fetch Error Detected. The DMA write channel has received an error response from the AXI bus (or TRGT1 interface when the AXI Bridge is not used) while reading a linked list element from local memory. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA write interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Write Interrupt Clear Register" (DMA_WRITE_INT_CLEAR_OFF) to clear this error bit.
15-8 RSVDP_8	Reserved for future use.
7-0 APP_READ_ERR_DETECT	Application Read Error Detected. The DMA write channel has received an error response from the AXI bus (or TRGT1 interface when the AXI Bridge is not used) while reading data from it. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA write interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Write Interrupt Clear Register" (DMA_WRITE_INT_CLEAR_OFF) to clear this error bit.

11.4.3.1.193 DMA Write Done IMWr Address Low Register. (DMA_WRITE_DONE_IMWR_LOW_OFF)

11.4.3.1.193.1 Offset

Register	Offset
DMA_WRITE_DONE_IMWR_LOW_OFF	38_0060h

11.4.3.1.193.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_WRITE_DONE_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_WRITE_DONE_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.193.3 Fields

Field	Function
31-0 DMA_WRITE_DONE_LOW_REG	The DMA uses this field to generate bits [31:0] of the address field for the Done IMWr TLP. Bits [1:0] must be "00" as this address must be dword aligned. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.194 DMA Write Done IMWr Interrupt Address High Register. (DMA_WRITE_DONE_IMWR_HIGH_OFF)

11.4.3.1.194.1 Offset

Register	Offset
DMA_WRITE_DONE_IMWR_HIGH_OFF	38_0064h

11.4.3.1.194.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_WRITE_DONE_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_WRITE_DONE_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.194.3 Fields

Field	Function
31-0 DMA_WRITE_DONE_HIGH_REG	The DMA uses this field to generate bits [63:32] of the address field for the Done IMWr TLP. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.195 DMA Write Abort IMWr Address Low Register. (DMA_WRITE_ABORT_IMWR_LOW_OFF)

11.4.3.1.195.1 Offset

Register	Offset
DMA_WRITE_ABORT_IMWR_LOW_OFF	38_0068h

11.4.3.1.195.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_WRITE_ABORT_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_WRITE_ABORT_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.195.3 Fields

Field	Function
31-0 DMA_WRITE_ABORT_LOW_REG	The DMA uses this field to generate bits [31:0] of the address field for the Abort IMWr TLP it generates. Bits [1:0] must be "00" as this address must be dword aligned. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.196 DMA Write Abort IMWr Address High Register. (DMA_WRITE_ABORT_IMWR_HIGH_OFF)

11.4.3.1.196.1 Offset

Register	Offset
DMA_WRITE_ABORT_IMWR_HIGH_OFF	38_006Ch

11.4.3.1.196.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_WRITE_ABORT_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_WRITE_ABORT_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.196.3 Fields

Field	Function
31-0 DMA_WRITE_ABORT_HIGH_REG	The DMA uses this field to generate bits [63:32] of the address field for the Abort IMWr TLP. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.197 DMA Write Channel 1 and 0 IMWr Data Register. (DMA_WRITE_CH01_IMWR_DATA_OFF)

11.4.3.1.197.1 Offset

Register	Offset
DMA_WRITE_CH01_IMWR_DATA_OFF	38_0070h

11.4.3.1.197.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WR_CHANNEL_1_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WR_CHANNEL_0_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.197.3 Fields

Field	Function
31-16 WR_CHANNEL_1_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 1. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 WR_CHANNEL_0_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 0. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.198 DMA Write Channel 3 and 2 IMWr Data Register. (DMA_WRITE_CH23_IMWR_DATA_OFF)

11.4.3.1.198.1 Offset

Register	Offset
DMA_WRITE_CH23_IMWR_DATA_OFF	38_0074h

11.4.3.1.198.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WR_CHANNEL_3_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WR_CHANNEL_2_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.198.3 Fields

Field	Function
31-16 WR_CHANNEL_3_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 3. Note: The access attributes of this field are as follows: - Dbi: R/W

Table continues on the next page...

Field	Function
15-0 WR_CHANNEL_2_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 2. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.199 DMA Write Channel 5 and 4 IMWr Data Register. (DMA_WRITE_CH45_IMWR_DATA_OFF)

11.4.3.1.199.1 Offset

Register	Offset
DMA_WRITE_CH45_IMWR_DATA_OFF	38_0078h

11.4.3.1.199.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WR_CHANNEL_5_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WR_CHANNEL_4_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.199.3 Fields

Field	Function
31-16 WR_CHANNEL_5_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 5. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 WR_CHANNEL_4_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 4. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.200 DMA Write Channel 7 and 6 IMWr Data Register. (DMA_WRITE_CH67_IMWR_DATA_OFF)

11.4.3.1.200.1 Offset

Register	Offset
DMA_WRITE_CH67_IMWR_DATA_OFF	38_007Ch

11.4.3.1.200.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WR_CHANNEL_7_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WR_CHANNEL_6_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.200.3 Fields

Field	Function
31-16 WR_CHANNEL_7_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 7. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 WR_CHANNEL_6_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 6. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.201 DMA Write Linked List Error Enable Register. (DMA_WRITE_LINKED_LIST_ERR_EN_OFF)

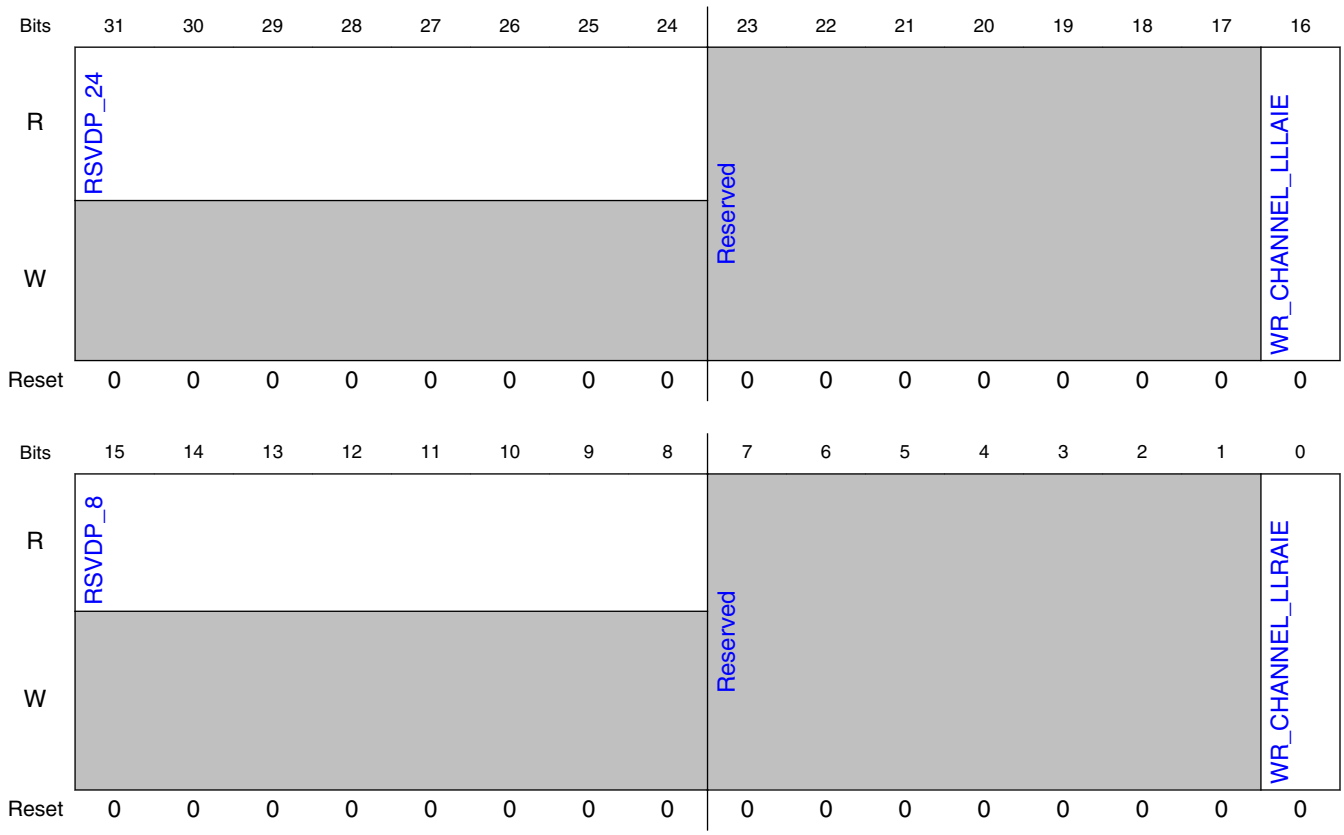
11.4.3.1.201.1 Offset

Register	Offset
DMA_WRITE_LINKED_LIST_ERR_EN_OFF	38_0090h

11.4.3.1.201.2 Function

DMA Write Linked List Error Enable Register. The LIE and RIE bits in the LL element enable the channel "done" interrupts (local and remote). The LLLAIE and LLRAIE bits enable the channel "abort" interrupts (local and remote).

11.4.3.1.201.3 Diagram



11.4.3.1.201.4 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-17 —	Reserved
16 WR_CHANNEL_LLLAIE	Write Channel LL Local Abort Interrupt Enable (LLLAIE). You enable the write channel local abort interrupt through this bit. The LIE and RIE bits in the LL element enable the write channel done interrupts. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Used in linked list mode only. For more details, see "Interrupt Handling". Note: The access attributes of this field are as follows: - Dbi: R/W

Table continues on the next page...

Field	Function
15-8 RSVDP_8	Reserved for future use.
7-1 —	Reserved
0 WR_CHANNEL_LLRAIE	Write Channel LL Remote Abort Interrupt Enable (LLRAIE). You enable the write channel remote abort interrupt through this bit. The LIE and RIE bits in the LL element enable the write channel done interrupts. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Used in linked list mode only. For more details, see "Interrupt Handling". Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.202 DMA Read Interrupt Status Register. (DMA_READ_INT_STATUS_OFF)

11.4.3.1.202.1 Offset

Register	Offset
DMA_READ_INT_STATUS_OFF	38_00A0h

11.4.3.1.202.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVDP_24								RD_ABORT_INT_STATUS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVDP_8								RD_DONE_INT_STATUS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.202.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16	Abort Interrupt Status. The DMA read channel has detected an error, or you manually stopped the transfer as described in "Stopping the DMA Transfer (Software Stop)". Each bit corresponds to a DMA

Table continues on the next page...

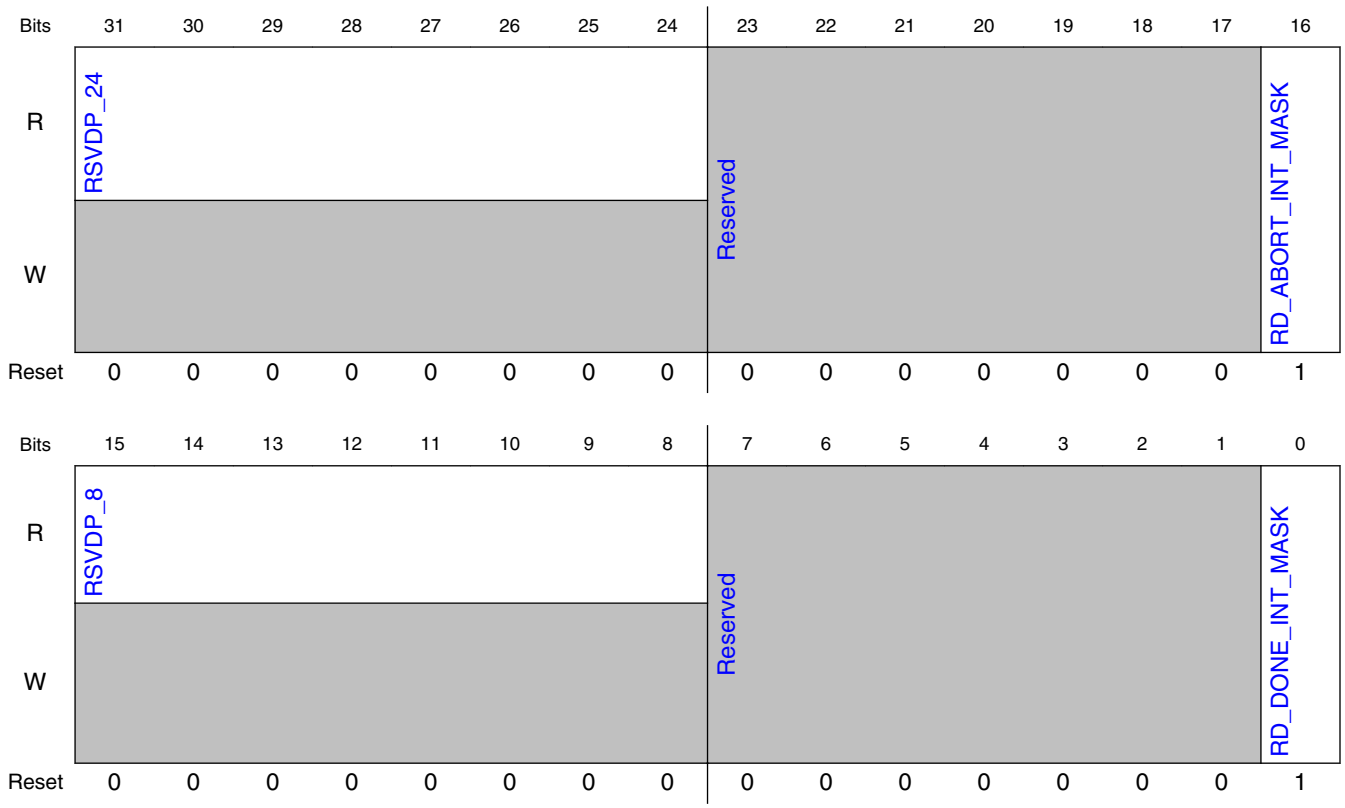
Field	Function
RD_ABORT_INT_STATUS	channel. Bit [0] corresponds to channel 0. You can read the "DMA Read Error Status Low Register" (DMA_READ_ERR_STATUS_LOW_OFF) and "DMA Read Error Status High Register" (DMA_READ_ERR_STATUS_HIGH_OFF) to determine the source of the error. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the DMA read interrupt Clear register to clear this interrupt bit. Note: You can write to this register to emulate interrupt generation, during software or hardware testing. A write to the address triggers an interrupt, but the DMA does not set the Done or Abort bits in this register. Note: The access attributes of this field are as follows: - Dbi: R/W
15-8 RSVDP_8	Reserved for future use.
7-0 RD_DONE_INT_STATUS	Done Interrupt Status. The DMA read channel has successfully completed the DMA read transfer. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the DMA read interrupt Clear register to clear this interrupt bit. Note: You can write to this register to emulate interrupt generation, during software or hardware testing. A write to the address triggers an interrupt, but the DMA does not set the Done or Abort bits in this register. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.203 DMA Read Interrupt Mask Register. (DMA_READ_INT_MASK_OFF)

11.4.3.1.203.1 Offset

Register	Offset
DMA_READ_INT_MASK_OFF	38_00A8h

11.4.3.1.203.2 Diagram



11.4.3.1.203.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-17 —	Reserved
16 RD_ABORT_INT_MASK	Abort Interrupt Mask. Prevents the Abort interrupt status field in the DMA read interrupt status register from asserting the edma_int output. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: The access attributes of this field are as follows: - Dbi: R/W
15-8 RSVDP_8	Reserved for future use.
7-1 —	Reserved
0 RD_DONE_INT_MASK	Done Interrupt Mask. Prevents the Done interrupt status field in the DMA read interrupt status register from asserting the edma_int output. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.204 DMA Read Interrupt Clear Register. (DMA_READ_INT_CLEAR_OFF)

11.4.3.1.204.1 Offset

Register	Offset
DMA_READ_INT_CLEAR_OFF	38_00ACh

11.4.3.1.204.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVDP_24															
W									RD_ABORT_INT_CLEAR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVDP_8															
W									RD_DONE_INT_CLEAR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.204.3 Fields

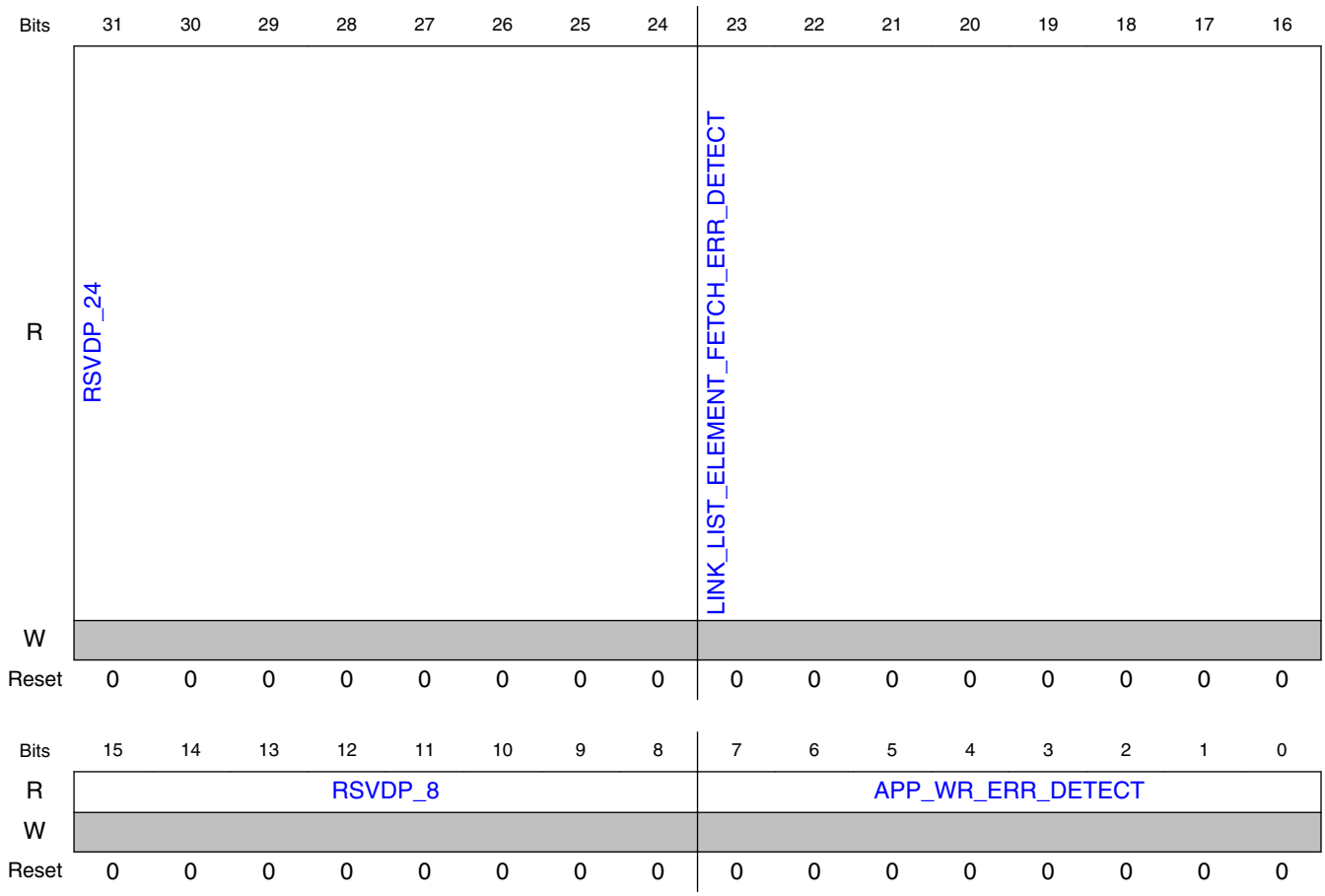
Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16 RD_ABORT_INT_CLEAR	Abort Interrupt Clear. You must write a 1'b1 to clear the corresponding bit in the Abort interrupt status field of the DMA read interrupt status register. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: Reading from this self-clearing register field always returns a "0".
15-8 RSVDP_8	Reserved for future use.
7-0 RD_DONE_INT_CLEAR	Done Interrupt Clear. You must write a 1'b1 to clear the corresponding bit in the Done interrupt status field of the DMA read interrupt status register. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: Reading from this self-clearing register field always returns a "0".

11.4.3.1.205 DMA Read Error Status Low Register. (DMA_READ_ERR_STATUS_LOW_OFF)

11.4.3.1.205.1 Offset

Register	Offset
DMA_READ_ERR_STATUS_LOW_OFF	38_00B4h

11.4.3.1.205.2 Diagram



11.4.3.1.205.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16 LINK_LIST_ELEMENT_FETCH_ERR_DETECT	Linked List Element Fetch Error Detected. - The DMA read channel has received an error response from the AXI bus while reading a linked list element from local memory. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Read Interrupt Clear

Table continues on the next page...

Field	Function
	Register" (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this clears all bits in this register, and also the DMA Read Error Status High register (DMA_READ_ERR_STATUS_HIGH_OFF).
15-8 RSVDP_8	Reserved for future use.
7-0 APP_WR_ERR_DETECT	Application Write Error Detected. The DMA read channel has received an error response from the AXI bus (or TRGT1 interface when the AXI Bridge is not used) while writing data to it. This error is fatal. You must restart the transfer from the beginning, as the channel context is corrupted, and the transfer is not rolled back. For more details, see "Linked List Mode". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Read Interrupt Clear Register" (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this clears all bits in this register, and also the DMA Read Error Status High register (DMA_READ_ERR_STATUS_HIGH_OFF).

11.4.3.1.206 DMA Read Error Status High Register. (DMA_READ_ERR_STATUS_HIGH_OFF)

11.4.3.1.206.1 Offset

Register	Offset
DMA_READ_ERR_STATUS_HIGH_OFF	38_00B8h

11.4.3.1.206.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DATA_POISIONING								CPL_TIMEOUT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CPL_ABORT								UNSUPPORTED_REQ							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.206.3 Fields

Field	Function
31-24	Data Poisoning. The DMA read channel has detected data poisoning in the completion from the remote device (in response to the MRd request). The DMA read channel will drop the completion and then be halted. The CX_FLT_MASK_UR_POIS filter rule does not affect this behavior. Each bit corresponds to a

Table continues on the next page...

Field	Function
DATA_POISIONING	DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Read Interrupt Clear Register" (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this also clears the other error bits for the same channel in this register and in the DMA Read Error Status Low register.
23-16 CPL_TIMEOUT	Completion Time Out. The DMA read channel has timed-out while waiting for the remote device to respond to the MRd request, or a malformed CplD has been received. For more details, see "Linked List Mode". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling" . - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Read Interrupt Clear Register" (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this also clears the other error bits for the same channel in this register and in the DMA Read Error Status Low register.
15-8 CPL_ABORT	Completer Abort. The DMA read channel has received a PCIe completer abort completion status from the remote device in response to the MRd request. For more details, see "Linked List Mode". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Read Interrupt Clear Register" (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this also clears the other error bits for the same channel in this register and in the DMA Read Error Status Low register.
7-0 UNSUPPORTED_REQ	Unsupported Request. The DMA read channel has received a PCIe unsupported request completion status from the remote device in response to the MRd request. For more details, see "Linked List Mode". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Read Interrupt Clear Register" (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this also clears the other error bits for the same channel in this register and in the DMA Read Error Status Low register.

11.4.3.1.207 DMA Read Linked List Error Enable Register. (DMA_READ_LINKED_LIST_ERR_EN_OFF)

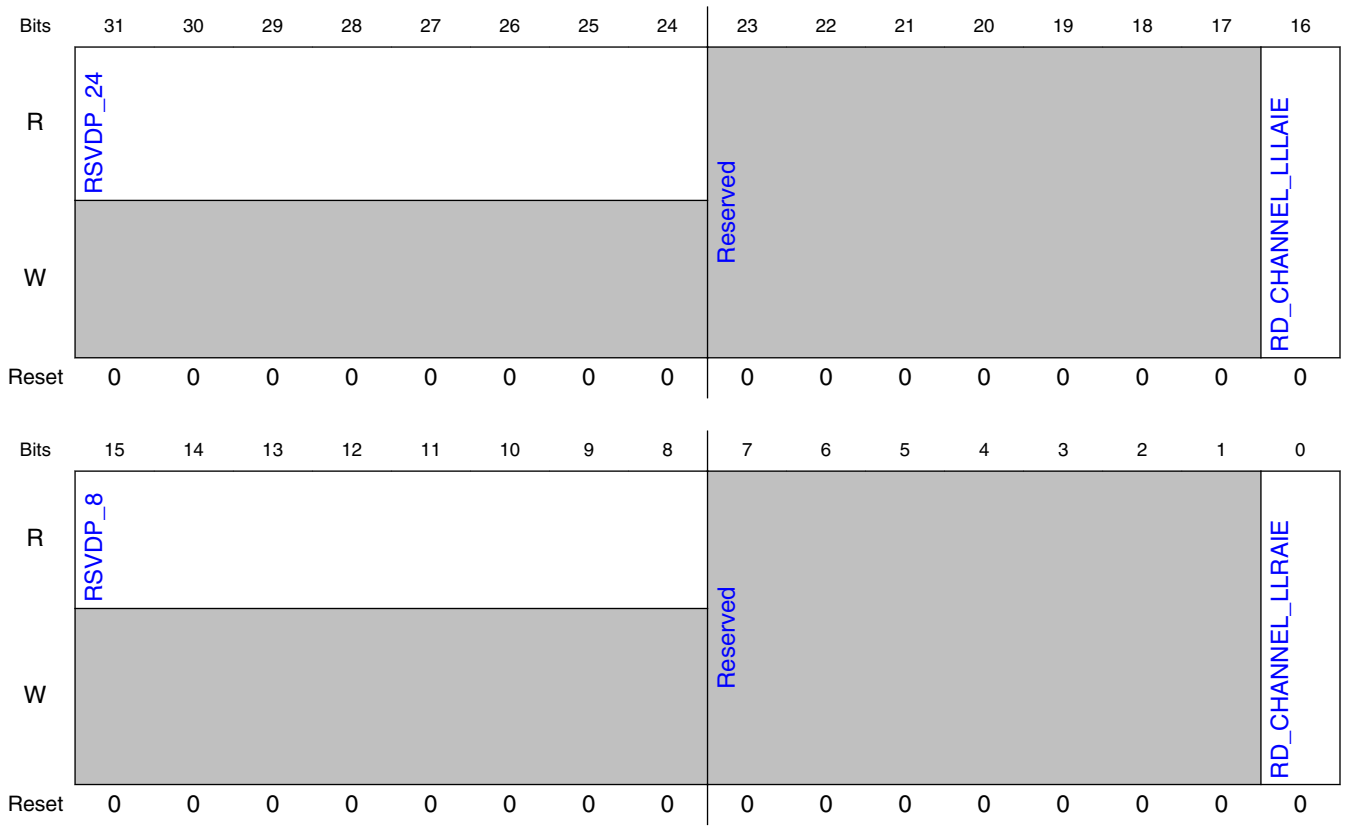
11.4.3.1.207.1 Offset

Register	Offset
DMA_READ_LINKED_LIST_ERR_EN_OFF	38_00C4h

11.4.3.1.207.2 Function

DMA Read Linked List Error Enable Register. The LIE and RIE bits in the LL element enable the channel "done" interrupts (local and remote). The LLLAIE and LLRAIE bits enable the channel "abort" interrupts (local and remote).

11.4.3.1.207.3 Diagram



11.4.3.1.207.4 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-17 —	Reserved
16 RD_CHANNEL_LLLAIE	Read Channel LL Local Abort Interrupt Enable (LLLAIE). You enable the read channel Local Abort interrupt through this bit. The LIE and RIE bits in the LL element enable the read channel done interrupts. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Used in linked list mode only. For more details, see "Interrupt Handling". Note: The access attributes of this field are as follows: - Dbi: R/W
15-8 RSVDP_8	Reserved for future use.
7-1 —	Reserved
0 RD_CHANNEL_LLRAIE	Read Channel LL Remote Abort Interrupt Enable (LLRAIE). You enable the read channel Remote Abort interrupt through this bit. The LIE and RIE bits in the LL element enable the read channel done interrupts. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Used in linked list mode only. For more details, see "Interrupt Handling". Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.208 DMA Read Done IMWr Address Low Register. (DMA_READ_DONE_IMWR_LOW_OFF)

11.4.3.1.208.1 Offset

Register	Offset
DMA_READ_DONE_IMWR_LOW_OFF	38_00CCh

11.4.3.1.208.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_READ_DONE_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_READ_DONE_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.208.3 Fields

Field	Function
31-0 DMA_READ_DONE_LOW_REG	The DMA uses this field to generate bits [31:0] of the address field for the Done IMWr TLP. Bits [1:0] must be "00" as this address must be dword aligned. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.209 DMA Read Done IMWr Address High Register. (DMA_READ_DONE_IMWR_HIGH_OFF)

11.4.3.1.209.1 Offset

Register	Offset
DMA_READ_DONE_IMWR_HIGH_OFF	38_00D0h

11.4.3.1.209.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_READ_DONE_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_READ_DONE_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.209.3 Fields

Field	Function
31-0 DMA_READ_DONE_HIGH_REG	The DMA uses this field to generate bits [63:32] of the address field for the Done IMWr TLP. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.210 DMA Read Abort IMWr Address Low Register. (DMA_READ_ABORT_IMWR_LOW_OFF)

11.4.3.1.210.1 Offset

Register	Offset
DMA_READ_ABORT_IMWR_LOW_OFF	38_00D4h

11.4.3.1.210.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_READ_ABORT_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_READ_ABORT_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.210.3 Fields

Field	Function
31-0 DMA_READ_ABORT_LOW_REG	The DMA uses this field to generate bits [31:0] of the address field for the Abort IMWr TLP. Bits [1:0] must be "00" as this address must be dword aligned. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.211 DMA Read Abort IMWr Address High Register. (DMA_READ_ABORT_IMWR_HIGH_OFF)

11.4.3.1.211.1 Offset

Register	Offset
DMA_READ_ABORT_IMWR_HIGH_OFF	38_00D8h

11.4.3.1.211.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_READ_ABORT_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_READ_ABORT_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.211.3 Fields

Field	Function
31-0 DMA_READ_ABORT_HIGH_REG	The DMA uses this field to generate bits [63:32] of the address field for the Abort IMWr TLP. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.212 DMA Read Channel 1 and 0 IMWr Data Register. (DMA_READ_CH01_IMWR_DATA_OFF)

11.4.3.1.212.1 Offset

Register	Offset
DMA_READ_CH01_IMWR_DATA_OFF	38_00DCh

11.4.3.1.212.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RD_CHANNEL_1_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RD_CHANNEL_0_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.212.3 Fields

Field	Function
31-16 RD_CHANNEL_1_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 1. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 RD_CHANNEL_0_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 0. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.213 DMA Read Channel 3 and 2 IMWr Data Register. (DMA_READ_CH23_IMWR_DATA_OFF)

11.4.3.1.213.1 Offset

Register	Offset
DMA_READ_CH23_IMWR_DATA_OFF	38_00E0h

11.4.3.1.213.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RD_CHANNEL_3_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RD_CHANNEL_2_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.213.3 Fields

Field	Function
31-16 RD_CHANNEL_3_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 3. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 RD_CHANNEL_2_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 2. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.214 DMA Read Channel 5 and 4 IMWr Data Register. (DMA_READ_CH45_IMWR_DATA_OFF)

11.4.3.1.214.1 Offset

Register	Offset
DMA_READ_CH45_IMWR_DATA_OFF	38_00E4h

11.4.3.1.214.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RD_CHANNEL_5_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RD_CHANNEL_4_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.214.3 Fields

Field	Function
31-16 RD_CHANNEL_5_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 5. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 RD_CHANNEL_4_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 4. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.215 DMA Read Channel 7 and 6 IMWr Data Register. (DMA_READ_CH67_IMWR_DATA_OFF)

11.4.3.1.215.1 Offset

Register	Offset
DMA_READ_CH67_IMWR_DATA_OFF	38_00E8h

11.4.3.1.215.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RD_CHANNEL_7_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RD_CHANNEL_6_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.215.3 Fields

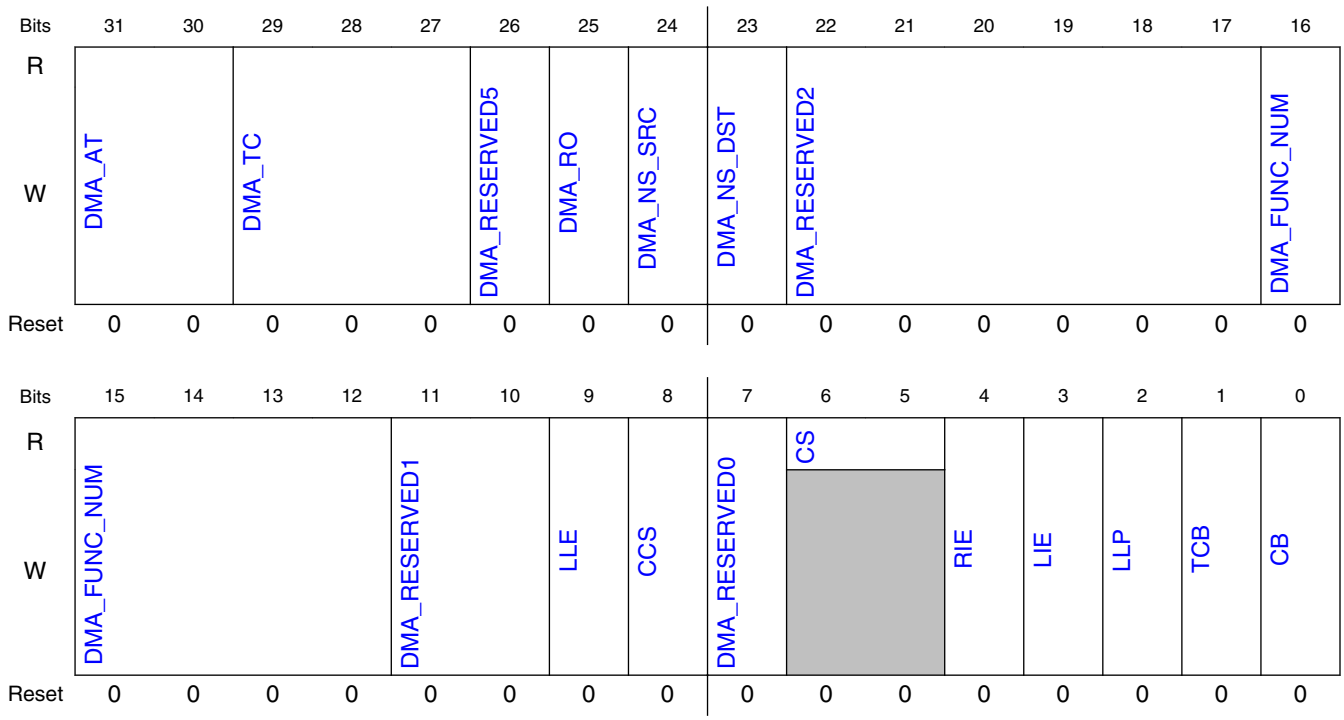
Field	Function
31-16 RD_CHANNEL_7_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 7. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 RD_CHANNEL_6_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 6. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.216 DMA Write Channel Control 1 Register. (DMA_CH_CONTROL1_OFF_WRCH_0)

11.4.3.1.216.1 Offset

Register	Offset
DMA_CH_CONTROL1_OFF_WRCH_0	38_0200h

11.4.3.1.216.2 Diagram



11.4.3.1.216.3 Fields

Field	Function
31-30 DMA_AT	Address Translation TLP Header Bit (AT) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
29-27 DMA_TC	Traffic Class TLP Header Bit (TC) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
26 DMA_RESERVED5	Reserved Note: The access attributes of this field are as follows: - Dbi: R/W
25 DMA_RO	Relaxed Ordering TLP Header Bit (RO) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
24 DMA_NS_SRC	Source No Snoop TLP Header Bit (DMA_NS_SRC). The DMA uses this TLP header field when generating MRd (SAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
23 DMA_NS_DST	Destination No Snoop TLP Header Bit (DMA_NS_DST). The DMA uses this TLP header field when generating MWr (DAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
22-17 DMA_RESERVED2	Reserved Note: The access attributes of this field are as follows: - Dbi: R/W

Table continues on the next page...

Field	Function
16-12 DMA_FUNC_NUM	Function Number (FN). The controller uses this field when generating the requester ID for the MRd/MWr DMA TLP. When you have enabled SR-IOV, then this field is ignored if you have set the VFE field in the "DMA Write Channel Control 2 Register" (DMA_CH_CONTROL2_OFF_WRCH_0). Note: The access attributes of this field are as follows: - Dbi: R/W
11-10 DMA_RESERVED1	Reserved Note: The access attributes of this field are as follows: - Dbi: R/W
9 LLE	Linked List Enable (LLE). - 0: Disable linked list operation - 1: Enable linked list operation Note: The access attributes of this field are as follows: - Dbi: R/W
8 CCS	Consumer Cycle State (CCS). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". You must initialize this bit. The DMA updates this bit during linked list operation. Note: The access attributes of this field are as follows: - Dbi: R/W
7 DMA_RESERVED0	Reserved Note: The access attributes of this field are as follows: - Dbi: R/W
6-5 CS	Channel Status (CS). The channel status bits identify the current operational state of the DMA channel. The operation state encoding for each DMA channel is a s follows: - 00: Reserved - 01: Running. This channel is active and transferring data. - 10: Halted. An error condition has been detected, and the DMA has stopped this channel. - 11: Stopped. The DMA has transferred all data for this channel, or you have prematurely stopped this channel by writing to the Stop field of the "DMA Write Doorbell Register" (DMA_WRITE_DOORBELL_OFF) or "DMA Read Doorbell Register" (DMA_READ_DOORBELL_OFF).
4 RIE	Remote Interrupt Enable (RIE). You must set this bit to enable the generation of the Done or Abort Remote interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the RIE of the LL element. The RIE bit in a LL element only enables the Done interrupt. In non-LL mode, the RIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Dbi: R/W
3 LIE	Local Interrupt Enable (LIE). You must set this bit to enable the generation of the Done or Abort Local interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the LIE of the LL element. The LIE bit in a LL element only enables the Done interrupt. In non-LL mode, the LIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Dbi: R/W
2 LLP	Load Link Pointer (LLP). Used in linked list mode only. Indicates that this linked list element is a link element, and its LL element pointer dwords are pointing to the next (non-contiguous) element. The DMA loads this field with the LLP of the linked list element. Note: The access attributes of this field are as follows: - Dbi: R/W
1 TCB	Toggle Cycle Bit (TCB). Indicates to the DMA to toggle its interpretation of the CB. Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the TCB of the linked list element. this field is not defined in a data LL element. Note: The access attributes of this field are as follows: - Dbi: R/W
0 CB	Cycle Bit (CB). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the CB of the linked list element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.217 DMA Write Transfer Size Register. (DMA_TRANSFER_SIZE_OFF_WRCH_0)

11.4.3.1.217.1 Offset

Register	Offset
DMA_TRANSFER_SIZE_OFF_WRCH_0	38_0208h

11.4.3.1.217.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_TRANSFER_SIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_TRANSFER_SIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.217.3 Fields

Field	Function
31-0 DMA_TRANSFER_SIZE	DMA Transfer Size. You program this register with the size of the DMA transfer. The maximum DMA transfer size is 4Gbytes. The minimum transfer size is one byte (0x1). This field is automatically decremented by the DMA as the DMA write channel transfer progresses. This field indicates the number bytes remaining to be transferred. When all bytes are successfully transferred the current transfer size is zero. In LL mode, the DMA overwrites this register with the corresponding dword of the LL element. You can read this register to monitor the transfer progress, however in some scenarios this register is updated after a delay. For example, when less than 3 channels are doorbelled, this register is updated only after a descriptor finishes(linked list mode), or the transfer ends (non-linked list mode). Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.218 DMA Write SAR Low Register. (DMA_SAR_LOW_OFF_WRCH_0)

11.4.3.1.218.1 Offset

Register	Offset
DMA_SAR_LOW_OFF_WRCH_0	38_020Ch

11.4.3.1.218.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SRC_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRC_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.218.3 Fields

Field	Function
31-0 SRC_ADDR_REG_LOW	Source Address Register (Lower 32 bits). Indicates the next address to be read from. The DMA increments the SAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The SAR is the address of the remote memory. - DMA Write: The SAR is the address of the local memory. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.219 DMA Write SAR High Register. (DMA_SAR_HIGH_OFF_WRCH_0)

11.4.3.1.219.1 Offset

Register	Offset
DMA_SAR_HIGH_OFF_WRCH_0	38_0210h

11.4.3.1.219.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SRC_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRC_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.219.3 Fields

Field	Function
31-0 SRC_ADDR_REG_HIGH	Source Address Register (Higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.220 DMA Write DAR Low Register. (DMA_DAR_LOW_OFF_WRCH_0)

11.4.3.1.220.1 Offset

Register	Offset
DMA_DAR_LOW_OFF_WRCH_0	38_0214h

11.4.3.1.220.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DST_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DST_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.220.3 Fields

Field	Function
31-0 DST_ADDR_REG_LOW	Destination Address Register (Lower 32 bits). Indicates the next address to be written to. The DMA increments the DAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The DAR is the address of the local memory. - DMA Write: The DAR is the address of the remote memory. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.221 DMA Write DAR High Register. (DMA_DAR_HIGH_OFF_WRCH_0)

11.4.3.1.221.1 Offset

Register	Offset
DMA_DAR_HIGH_OFF_WRCH_0	38_0218h

11.4.3.1.221.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DST_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DST_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.221.3 Fields

Field	Function
31-0 DST_ADDR_REG_HIGH	Destination Address Register (Higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.222 DMA Write Linked List Pointer Low Register. (DMA_LLP_LOW_OFF_WRCH_0)

11.4.3.1.222.1 Offset

Register	Offset
DMA_LLP_LOW_OFF_WRCH_0	38_021Ch

11.4.3.1.222.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LLP_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LLP_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.222.3 Fields

Field	Function
31-0 LLP_LOW	Lower bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list after the previous element is consumed. - When the current element is a data element; this field is incremented by 6. - When the current element is a link element; this field is overwritten by the LL Element Pointer of the element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.223 DMA Write Linked List Pointer High Register. (DMA_LLP_HIGH_OFF_WRCH_0)

11.4.3.1.223.1 Offset

Register	Offset
DMA_LLP_HIGH_OFF_WRCH_0	38_0220h

11.4.3.1.223.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LLP_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LLP_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.223.3 Fields

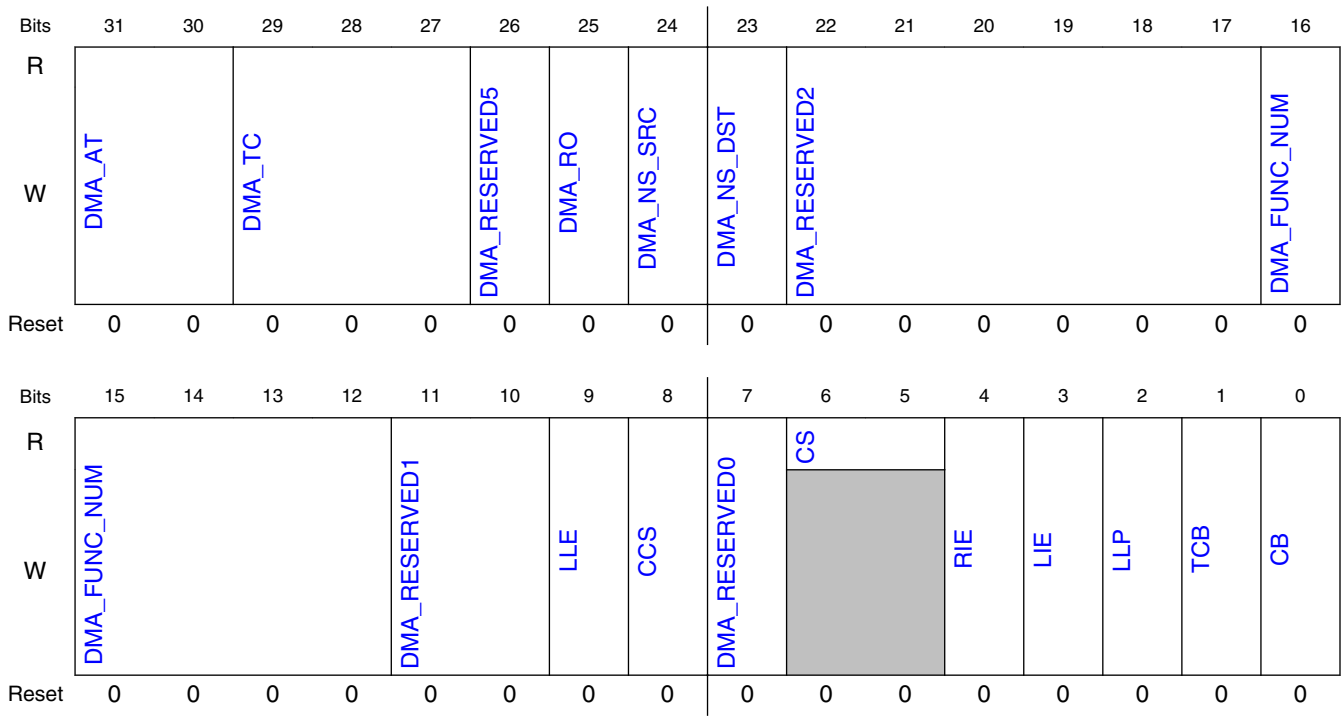
Field	Function
31-0 LLP_HIGH	Upper 32 bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list as elements are consumed. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.224 DMA Read Channel Control 1 Register. (DMA_CH_CONTROL1_OFF_RDCH_0)

11.4.3.1.224.1 Offset

Register	Offset
DMA_CH_CONTROL1_OFF_RDCH_0	38_0300h

11.4.3.1.224.2 Diagram



11.4.3.1.224.3 Fields

Field	Function
31-30 DMA_AT	Address Translation TLP Header Bit (AT) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
29-27 DMA_TC	Traffic Class TLP Header Bit (TC) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
26 DMA_RESERVED5	Reserved Note: The access attributes of this field are as follows: - Dbi: R/W
25 DMA_RO	Relaxed Ordering TLP Header Bit (RO) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
24 DMA_NS_SRC	Source No Snoop TLP Header Bit (DMA_NS_SRC). The DMA uses this TLP header field when generating MRd (SAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
23 DMA_NS_DST	Destination No Snoop TLP Header Bit (DMA_NS_DST). The DMA uses this TLP header field when generating MWr (DAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
22-17 DMA_RESERVED2	Reserved Note: The access attributes of this field are as follows: - Dbi: R/W

Table continues on the next page...

Field	Function
16-12 DMA_FUNC_NUM	Function Number (FN). The controller uses this field when generating the requester ID for the MRd/MWr DMA TLP. When you have enabled SR-IOV, then this field is ignored if you have set the VFE field in the "DMA Read Channel Control 2 Register" (DMA_CH_CONTROL2_OFF_RDCH_0). Note: The access attributes of this field are as follows: - Dbi: R/W
11-10 DMA_RESERVED1	Reserved Note: The access attributes of this field are as follows: - Dbi: R/W
9 LLE	Linked List Enable (LLE). - 0: Disable linked list operation - 1: Enable linked list operation Note: The access attributes of this field are as follows: - Dbi: R/W
8 CCS	Consumer Cycle State (CCS). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". You must initialize this bit. The DMA updates this bit during linked list operation. Note: The access attributes of this field are as follows: - Dbi: R/W
7 DMA_RESERVED0	Reserved Note: The access attributes of this field are as follows: - Dbi: R/W
6-5 CS	Channel Status (CS). The channel status bits identify the current operational state of the DMA channel. The operation state encoding for each DMA channel is a s follows: - 00: Reserved - 01: Running. This channel is active and transferring data. - 10: Halted. An error condition has been detected, and the DMA has stopped this channel. - 11: Stopped. The DMA has transferred all data for this channel, or you have prematurely stopped this channel by writing to the Stop field of the "DMA Read Doorbell Register" (DMA_WRITE_DOORBELL_OFF) or "DMA Read Doorbell Register" (DMA_READ_DOORBELL_OFF).
4 RIE	Remote Interrupt Enable (RIE). You must set this bit to enable the generation of the Done or Abort Remote interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the RIE of the LL element. The RIE bit in a LL element only enables the Done interrupt. In non-LL mode, the RIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Dbi: R/W
3 LIE	Local Interrupt Enable (LIE). You must set this bit to enable the generation of the Done or Abort Local interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the LIE of the LL element. The LIE bit in a LL element only enables the Done interrupt. In non-LL mode, the LIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Dbi: R/W
2 LLP	Load Link Pointer (LLP). Used in linked list mode only. Indicates that this linked list element is a link element, and its LL element pointer dwords are pointing to the next (non-contiguous) element. The DMA loads this field with the LLP of the linked list element. Note: The access attributes of this field are as follows: - Dbi: R/W
1 TCB	Toggle Cycle Bit (TCB). Indicates to the DMA to toggle its interpretation of the CB. Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the TCB of the linked list element. this field is not defined in a data LL element. Note: The access attributes of this field are as follows: - Dbi: R/W
0 CB	Cycle Bit (CB). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the CB of the linked list element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.225 DMA Read Transfer Size Register. (DMA_TRANSFER_SIZE_OFF_RDCH_0)

11.4.3.1.225.1 Offset

Register	Offset
DMA_TRANSFER_SIZE_OFF_RDCH_0	38_0308h

11.4.3.1.225.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_TRANSFER_SIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_TRANSFER_SIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.225.3 Fields

Field	Function
31-0 DMA_TRANSFER_SIZE	DMA Transfer Size. You program this register with the size of the DMA transfer. The maximum DMA transfer size is 4Gbytes. The minimum transfer size is one byte (0x1). This field is automatically decremented by the DMA as the DMA read channel transfer progresses. This field indicates the number bytes remaining to be transferred. When all bytes are successfully transferred the current transfer size is zero. In LL mode, the DMA overwrites this register with the corresponding dword of the LL element. You can read this register to monitor the transfer progress, however in some scenarios this register is updated after a delay. For example, when less than 3 channels are doorbelled, this register is updated only after a descriptor finishes(linked list mode), or the transfer ends (non-linked list mode). Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.226 DMA Read SAR Low Register. (DMA_SAR_LOW_OFF_RDCH_0)

11.4.3.1.226.1 Offset

Register	Offset
DMA_SAR_LOW_OFF_RDCH_0	38_030Ch

11.4.3.1.226.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SRC_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRC_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.226.3 Fields

Field	Function
31-0 SRC_ADDR_REG_LOW	Source Address Register (Lower 32 bits). Indicates the next address to be read from. The DMA increments the SAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The SAR is the address of the remote memory. - DMA Read: The SAR is the address of the local memory. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.227 DMA Read SAR High Register. (DMA_SAR_HIGH_OFF_RDCH_0)

11.4.3.1.227.1 Offset

Register	Offset
DMA_SAR_HIGH_OFF_RDCH_0	38_0310h

11.4.3.1.227.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SRC_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRC_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.227.3 Fields

Field	Function
31-0 SRC_ADDR_REG_HIGH	Source Address Register (Higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.228 DMA Read DAR Low Register. (DMA_DAR_LOW_OFF_RDCH_0)

11.4.3.1.228.1 Offset

Register	Offset
DMA_DAR_LOW_OFF_RDCH_0	38_0314h

11.4.3.1.228.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DST_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DST_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.228.3 Fields

Field	Function
31-0 DST_ADDR_REG_LOW	Destination Address Register (Lower 32 bits). Indicates the next address to be written to. The DMA increments the DAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The DAR is the address of the local memory. - DMA Read: The DAR is the address of the remote memory. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.229 DMA Read DAR High Register. (DMA_DAR_HIGH_OFF_RDCH_0)

11.4.3.1.229.1 Offset

Register	Offset
DMA_DAR_HIGH_OFF_RDCH_0	38_0318h

11.4.3.1.229.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DST_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DST_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.229.3 Fields

Field	Function
31-0 DST_ADDR_REG_HIGH	Destination Address Register (Higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.230 DMA Read Linked List Pointer Low Register. (DMA_LLP_LOW_OFF_RDCH_0)

11.4.3.1.230.1 Offset

Register	Offset
DMA_LLP_LOW_OFF_RDCH_0	38_031Ch

11.4.3.1.230.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LLP_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LLP_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.230.3 Fields

Field	Function
31-0 LLP_LOW	Lower bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list after the previous element is consumed. - When the current element is a data element; this field is incremented by 6. - When the current element is a link element; this field is overwritten by the LL Element Pointer of the element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4.3.1.231 DMA Read Linked List Pointer High Register. (DMA_LLP_HIGH_OFF_RDCH_0)

11.4.3.1.231.1 Offset

Register	Offset
DMA_LLP_HIGH_OFF_RDCH_0	38_0320h

11.4.3.1.231.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LLP_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LLP_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.3.1.231.3 Fields

Field	Function
31-0 LLP_HIGH	Upper 32 bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list as elements are consumed. Note: The access attributes of this field are as follows: - Dbi: R/W

11.5 PCI Express PHY (PCIe_PHY)

11.5.1 Overview

The PCIe PHY IP core is used for PCI-Express (PCIe) applications. The transceiver in the core performs these functions:

- Serializes the 8b/10b encoded data for transmission for Gen1 and Gen2 operation, and 128b/130b encoded data for Gen3
- De-serializes the received code groups

When transmitting the data, the transceiver performs these functions:

- Accepts four 10-bit 8b/10b encoded transmit characters or four 8-bit 128b/130b encoded data
- Attaches and serializes the data to the TXP/TXN differential outputs to a maximum value of 8.0 Gb/s

When receiving the data, the transceiver performs these functions:

- Samples the received serial data on the RXP/RXN differential inputs
- Deserializes it into four 10-bit(or 8b-bit) received characters

PCIe PHY core has on-chip PLL circuitry for synthesis of the baud-rate transmitting clocks, and extraction of the retimed clocks from the received serial stream.

11.5.1.1 Block Diagram

The figure below illustrates the block diagram of a 4-lane PCIe PHY.

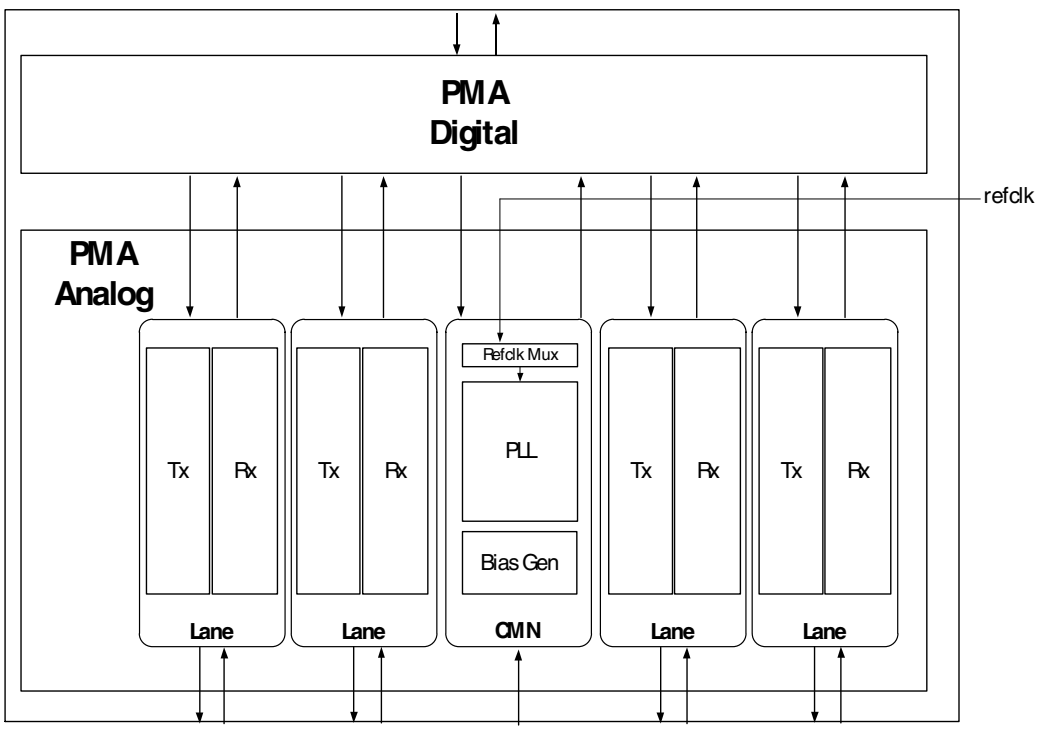


Figure 11-57. Block Diagram of PCIe PHY

The PCIe PHY comprises of one Common, four lanes of transceiver, one external reference clock in/output buffer, and PMA digital.

11.5.1.2 Features

The supported features in a Samsung PCIe PHY IP core are:

- PCI Express Base Specification 4.0 compliance
- 2.5Gb/s, 5.0Gb/s, and 8.0Gb/s Serializer/Deserializer
- PHY Interface for the PCI Express Architecture, Version 4.2 compliance
- Receiver Detection

- Spread Spectrum Clocking in Transmitter and Receiver
- Separate Refclk Independent SSC (SRIS) Architecture
- Continuous-Time Linear Equalizer and 5-tap adaptive Decision-Feedback Equalizer

11.5.2 Functional Description

The PCIe PHY comprises of a PMA analog and PMA digital. The PMA consists of:

- One CMN block
- Four lane blocks including a full-duplex transceiver,
- One external reference clock input/output (bidirectional) buffer.

In the transmitter side, the PMA Analog executes these actions:

- PMA Analog receives parallel data from PMA digital.
- The TX driver characteristics, such as amplitude and 3-tap FIR filter, are both presettable and user- controllable.

In the receiver side, the PMA executes these actions:

- Distortion on received signal due to channel loss is compensated by Continuous Time Linear Equalizer (CTLE) and Decision Feedback Equalizer (DFE).
- Clock and Data Recovery (CDR) generates frequency- and phase-aligned clock and equalized signal is deserialized to parallel data which is delivered to the PHY digital.

11.5.2.1 Common (CMN)

The CMN consists of a band-gap reference (BGR), bias generator, and PLL. Detailed information on each block is described in the following sections

11.5.2.1.1 BGR and bias generator

The BGR circuit generates high-accuracy reference voltage across PVT variation.

A bias generator makes two kinds of current, external- and internal-resistor (RMRES)-referred current, and these currents are distributed to each block in CMN and lanes. The external resistor-referred (REXT) current is generated using an 8.2KΩ resistor which is connected between the rext pad and ground outside the chip. On the other hand, the internal resistor-referred current is generated using internally integrated replica resistor to compensate resistance error across PVT variation.

11.5.2.1.2 Phase Locked Loop (PLL)

The PLL in the CMN synthesizes high-speed clock, which is used for TX serializer and RX CDR lock, from a reference clock. The reference clock can be selected from two clock sources; internal SoC reference clock and external differential reference clock.

In the PCIe PHY, the high frequency clock from PLL is used in the serialization of TX data and CDR lock acquisition and maintenance. The following are the main features of the PLL:

- Supports wide-range LC VCO from 8GHz to 10GHz
- Supports pre-defined and programmable divider setting for each PCIe specification
- Supports spread spectrum clocking (SSC) with sigma-delta modulated fractional divider.
- Supports Automatic Frequency Calibrated (AFC) oscillator setting for the desired frequency of operation
- Supports programmable charge pump current and loop filter resistance allowing for wide range of programmable loop bandwidth and peaking

11.5.2.2 Transmitter

The main features of the TX are provided below:

- Supports high voltage swing with auxiliary current-mode driver.
- Supports 3-tap FIR filter with resolution of 1/48.
- Supports 10 FIR presets (P0 to P9)
- Supports automatic TX termination resistance calibration.
- Supports electrical idle mode with either fast or low-power common-mode keep.
- Supports receiver detection.

11.5.3 Memory Map and register definition

This section includes the PCIE_PHY module memory map and descriptions of the registers.

11.5.3.1 PCIe PHY register descriptions

11.5.3.1.1 PCIe_PHY Memory map

PCIE_PHY base address: 32F0_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	(CMN_REG000)	8	RW	00h
4h	(CMN_REG001)	8	RW	0Ah
8h	(CMN_REG002)	8	RW	00h
Ch	(CMN_REG003)	8	RW	15h
10h	(CMN_REG004)	8	RW	01h
14h	(CMN_REG005)	8	RW	F0h
18h	(CMN_REG006)	8	RW	08h
1Ch	(CMN_REG007)	8	RW	83h
20h	(CMN_REG008)	8	RW	06h
24h	(CMN_REG009)	8	RW	20h
28h	(CMN_REG00A)	8	RW	A9h
2Ch	(CMN_REG00B)	8	RW	0Fh
30h	(CMN_REG00C)	8	RW	46h
34h	(CMN_REG00D)	8	RW	46h
38h	(CMN_REG00E)	8	RW	46h
3Ch	(CMN_REG00F)	8	RW	46h
40h	(CMN_REG010)	8	RW	3Fh
44h	(CMN_REG011)	8	RW	3Fh
48h	(CMN_REG012)	8	RW	24h
4Ch	(CMN_REG013)	8	RW	24h
50h	(CMN_REG014)	8	RW	FFh
54h	(CMN_REG015)	8	RW	FFh
58h	(CMN_REG016)	8	RW	77h
5Ch	(CMN_REG017)	8	RW	77h
60h	(CMN_REG018)	8	RW	24h
64h	(CMN_REG019)	8	RW	24h
68h	(CMN_REG01A)	8	RW	00h
6Ch	(CMN_REG01B)	8	RW	00h
70h	(CMN_REG01C)	8	RW	3Fh
74h	(CMN_REG01D)	8	RW	3Fh
78h	(CMN_REG01E)	8	RW	DDh
7Ch	(CMN_REG01F)	8	RW	AAh
80h	(CMN_REG020)	8	RW	11h
84h	(CMN_REG021)	8	RW	11h
88h	(CMN_REG022)	8	RW	1Eh
8Ch	(CMN_REG023)	8	RW	A8h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
90h	(CMN_REG024)	8	RW	00h
94h	(CMN_REG025)	8	RW	00h
98h	(CMN_REG026)	8	RW	72h
9Ch	(CMN_REG027)	8	RW	00h
A0h	(CMN_REG028)	8	RW	00h
A4h	(CMN_REG029)	8	RW	FFh
A8h	(CMN_REG02A)	8	RW	FFh
ACh	(CMN_REG02B)	8	RW	00h
B0h	(CMN_REG02C)	8	RW	32h
B4h	(CMN_REG02D)	8	RW	32h
B8h	(CMN_REG02E)	8	RW	28h
BCh	(CMN_REG02F)	8	RW	28h
C0h	(CMN_REG030)	8	RW	32h
C4h	(CMN_REG031)	8	RW	32h
C8h	(CMN_REG032)	8	RW	28h
CCh	(CMN_REG033)	8	RW	28h
D0h	(CMN_REG034)	8	RW	01h
D4h	(CMN_REG035)	8	RW	04h
D8h	(CMN_REG036)	8	RW	32h
DCh	(CMN_REG037)	8	RW	10h
E0h	(CMN_REG038)	8	RW	05h
E4h	(CMN_REG039)	8	RW	02h
E8h	(CMN_REG03A)	8	RW	00h
ECh	(CMN_REG03B)	8	RW	00h
F0h	(CMN_REG03C)	8	RW	00h
F4h	(CMN_REG03D)	8	RW	00h
F8h	(CMN_REG03E)	8	RW	00h
FCh	(CMN_REG03F)	8	RW	00h
100h	(CMN_REG040)	8	RW	00h
104h	(CMN_REG041)	8	RW	00h
108h	(CMN_REG042)	8	RW	00h
10Ch	(CMN_REG043)	8	RW	00h
110h	(CMN_REG044)	8	RW	00h
114h	(CMN_REG045)	8	RW	00h
118h	(CMN_REG046)	8	RW	00h
11Ch	(CMN_REG047)	8	RW	00h
120h	(CMN_REG048)	8	RW	00h
124h	(CMN_REG049)	8	RW	00h
128h	(CMN_REG04A)	8	RW	00h
12Ch	(CMN_REG04B)	8	RW	00h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
130h	(CMN_REG04C)	8	RW	00h
134h	(CMN_REG04D)	8	RW	00h
138h	(CMN_REG04E)	8	RW	00h
13Ch	(CMN_REG04F)	8	RW	00h
140h	(CMN_REG050)	8	RW	00h
144h	(CMN_REG051)	8	RW	00h
148h	(CMN_REG052)	8	RW	00h
14Ch	(CMN_REG053)	8	RW	30h
150h	(CMN_REG054)	8	RW	30h
154h	(CMN_REG055)	8	RW	30h
158h	(CMN_REG056)	8	RW	30h
15Ch	(CMN_REG057)	8	RW	1Fh
160h	(CMN_REG058)	8	RW	1Fh
164h	(CMN_REG059)	8	RW	1Fh
168h	(CMN_REG05A)	8	RW	1Fh
16Ch	(CMN_REG05B)	8	RW	00h
170h	(CMN_REG05C)	8	RW	10h
174h	(CMN_REG05D)	8	RW	1Bh
178h	(CMN_REG05E)	8	RW	0Eh
17Ch	(CMN_REG05F)	8	RW	11h
180h	(CMN_REG060)	8	RW	23h
184h	(CMN_REG061)	8	RW	01h
188h	(CMN_REG062)	8	RW	00h
18Ch	(CMN_REG063)	8	RW	00h
190h	(CMN_REG064)	8	RW	0Ah
194h	(CMN_REG065)	8	RW	99h
198h	(CMN_REG066)	8	RW	C2h
19Ch	(CMN_REG067)	8	RW	00h
1A0h	(CMN_REG068)	8	RW	00h
1A4h	(CMN_REG069)	8	RW	00h
1A8h	(CMN_REG06A)	8	RW	05h
1ACh	(CMN_REG06B)	8	RW	10h
1B0h	(CMN_REG06C)	8	RW	14h
1B4h	(CMN_REG06D)	8	RW	00h
1B8h	(CMN_REG06E)	8	RW	00h
1BCh	(CMN_REG06F)	8	RW	00h
1C0h	(CMN_REG070)	8	RW	00h
1C4h	(CMN_REG071)	8	RW	00h
1C8h	(CMN_REG072)	8	RW	00h
1CCh	(CMN_REG073)	8	RW	00h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1D0h	(CMN_REG074)	8	RW	00h
1D4h	(CMN_REG075)	8	RW	00h
200h	(CMN_REG076)	8	RW	00h
204h	(CMN_REG077)	8	RW	1Fh
208h	(CMN_REG078)	8	RW	55h
20Ch	(CMN_REG079)	8	RW	00h
210h	(CMN_REG080)	8	RW	00h
214h	(CMN_REG081)	8	RW	18h
218h	(CMN_REG082)	8	RW	00h
400h	(TRSV_REG000)	8	RW	00h
404h	(TRSV_REG001)	8	RW	1Fh
408h	(TRSV_REG002)	8	RW	1Fh
40Ch	(TRSV_REG003)	8	RW	1Fh
410h	(TRSV_REG004)	8	RW	1Fh
414h	(TRSV_REG005)	8	RW	00h
418h	(TRSV_REG006)	8	RW	00h
41Ch	(TRSV_REG007)	8	RW	00h
420h	(TRSV_REG008)	8	RW	00h
424h	(TRSV_REG009)	8	RW	00h
428h	(TRSV_REG00A)	8	RW	00h
42Ch	(TRSV_REG00B)	8	RW	05h
430h	(TRSV_REG00C)	8	RW	93h
434h	(TRSV_REG00D)	8	RW	00h
438h	(TRSV_REG00E)	8	RW	55h
43Ch	(TRSV_REG00F)	8	RW	00h
440h	(TRSV_REG010)	8	RW	00h
444h	(TRSV_REG011)	8	RW	00h
448h	(TRSV_REG012)	8	RW	01h
44Ch	(TRSV_REG013)	8	RW	77h
450h	(TRSV_REG014)	8	RW	77h
454h	(TRSV_REG015)	8	RW	77h
458h	(TRSV_REG016)	8	RW	77h
45Ch	(TRSV_REG017)	8	RW	00h
460h	(TRSV_REG018)	8	RW	30h
464h	(TRSV_REG019)	8	RW	00h
468h	(TRSV_REG01A)	8	RW	3Ch
46Ch	(TRSV_REG01B)	8	RW	30h
470h	(TRSV_REG01C)	8	RW	00h
474h	(TRSV_REG01D)	8	RW	10h
478h	(TRSV_REG01E)	8	RW	80h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
47Ch	(TRSV_REG01F)	8	RW	00h
480h	(TRSV_REG020)	8	RW	01h
484h	(TRSV_REG021)	8	RW	37h
488h	(TRSV_REG022)	8	RW	00h
48Ch	(TRSV_REG023)	8	RW	00h
490h	(TRSV_REG024)	8	RW	33h
494h	(TRSV_REG025)	8	RW	37h
498h	(TRSV_REG026)	8	RW	31h
49Ch	(TRSV_REG027)	8	RW	00h
4A0h	(TRSV_REG028)	8	RW	04h
4A4h	(TRSV_REG029)	8	RW	6Dh
4A8h	(TRSV_REG02A)	8	RW	48h
4ACh	(TRSV_REG02B)	8	RW	00h
4B0h	(TRSV_REG02C)	8	RW	01h
4B4h	(TRSV_REG02D)	8	RW	1Bh
4B8h	(TRSV_REG02E)	8	RW	70h
4BCh	(TRSV_REG02F)	8	RW	0Fh
4C0h	(TRSV_REG030)	8	RW	5Ch
4C4h	(TRSV_REG031)	8	RW	5Ch
4C8h	(TRSV_REG032)	8	RW	7Fh
4CCh	(TRSV_REG033)	8	RW	7Fh
4D0h	(TRSV_REG034)	8	RW	00h
4D4h	(TRSV_REG035)	8	RW	00h
4D8h	(TRSV_REG036)	8	RW	03h
4DCh	(TRSV_REG037)	8	RW	0Fh
4E0h	(TRSV_REG038)	8	RW	0Fh
4E4h	(TRSV_REG039)	8	RW	0Fh
4E8h	(TRSV_REG03A)	8	RW	0Fh
4ECh	(TRSV_REG03B)	8	RW	00h
4F0h	(TRSV_REG03C)	8	RW	02h
4F4h	(TRSV_REG03D)	8	RW	04h
4F8h	(TRSV_REG03E)	8	RW	00h
4FCh	(TRSV_REG03F)	8	RW	02h
500h	(TRSV_REG040)	8	RW	02h
504h	(TRSV_REG041)	8	RW	00h
508h	(TRSV_REG042)	8	RW	11h
50Ch	(TRSV_REG043)	8	RW	00h
510h	(TRSV_REG044)	8	RW	5Fh
514h	(TRSV_REG045)	8	RW	00h
518h	(TRSV_REG046)	8	RW	EEh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
51Ch	(TRSV_REG047)	8	RW	FFh
520h	(TRSV_REG048)	8	RW	00h
524h	(TRSV_REG049)	8	RW	08h
528h	(TRSV_REG04A)	8	RW	01h
52Ch	(TRSV_REG04B)	8	RW	00h
530h	(TRSV_REG04C)	8	RW	00h
534h	(TRSV_REG04D)	8	RW	00h
538h	(TRSV_REG04E)	8	RW	00h
53Ch	(TRSV_REG04F)	8	RW	E0h
540h	(TRSV_REG050)	8	RW	00h
544h	(TRSV_REG051)	8	RW	05h
548h	(TRSV_REG052)	8	RW	00h
54Ch	(TRSV_REG053)	8	RW	24h
550h	(TRSV_REG054)	8	RW	00h
554h	(TRSV_REG055)	8	RW	00h
558h	(TRSV_REG056)	8	RW	00h
55Ch	(TRSV_REG057)	8	RW	00h
560h	(TRSV_REG058)	8	RW	00h
564h	(TRSV_REG059)	8	RW	00h
568h	(TRSV_REG05A)	8	RW	00h
56Ch	(TRSV_REG05B)	8	RW	00h
570h	(TRSV_REG05C)	8	RW	00h
574h	(TRSV_REG05D)	8	RW	00h
578h	(TRSV_REG05E)	8	RW	00h
57Ch	(TRSV_REG05F)	8	RW	00h
580h	(TRSV_REG060)	8	RW	00h
584h	(TRSV_REG061)	8	RW	00h
588h	(TRSV_REG062)	8	RW	00h
58Ch	(TRSV_REG063)	8	RW	00h
590h	(TRSV_REG064)	8	RW	00h
594h	(TRSV_REG065)	8	RW	01h
598h	(TRSV_REG066)	8	RW	00h
59Ch	(TRSV_REG067)	8	RW	00h
5A0h	(TRSV_REG068)	8	RW	3Fh
5A4h	(TRSV_REG069)	8	RW	04h
5A8h	(TRSV_REG06A)	8	RW	0Ch
5ACh	(TRSV_REG06B)	8	RW	18h
5B0h	(TRSV_REG06C)	8	RW	00h
5B4h	(TRSV_REG06D)	8	RW	04h
5B8h	(TRSV_REG06E)	8	RW	02h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
5BCh	(TRSV_REG06F)	8	RW	56h
5C0h	(TRSV_REG070)	8	RW	00h
5C4h	(TRSV_REG071)	8	RW	00h
5C8h	(TRSV_REG072)	8	RW	00h
5CCh	(TRSV_REG073)	8	RW	00h
5D0h	(TRSV_REG074)	8	RW	00h
5D4h	(TRSV_REG075)	8	RW	00h
5D8h	(TRSV_REG076)	8	RW	10h
5DCh	(TRSV_REG077)	8	RW	00h
5E0h	(TRSV_REG078)	8	RW	00h
5E4h	(TRSV_REG079)	8	RW	00h
5E8h	(TRSV_REG07A)	8	RW	00h
5ECh	(TRSV_REG07B)	8	RW	21h
5F0h	(TRSV_REG07C)	8	RW	00h
5F4h	(TRSV_REG07D)	8	RW	00h
5F8h	(TRSV_REG07E)	8	RW	C0h
5FCh	(TRSV_REG07F)	8	RW	00h
600h	(TRSV_REG080)	8	RW	00h
604h	(TRSV_REG081)	8	RW	00h
608h	(TRSV_REG082)	8	RW	00h
60Ch	(TRSV_REG083)	8	RW	00h
610h	(TRSV_REG084)	8	RW	00h
614h	(TRSV_REG085)	8	RW	17h
618h	(TRSV_REG086)	8	RW	FDh
61Ch	(TRSV_REG087)	8	RW	30h
620h	(TRSV_REG088)	8	RW	00h
624h	(TRSV_REG089)	8	RW	C8h
628h	(TRSV_REG08A)	8	RW	00h
62Ch	(TRSV_REG08B)	8	RW	C8h
630h	(TRSV_REG08C)	8	RW	00h
634h	(TRSV_REG08D)	8	RW	A0h
638h	(TRSV_REG08E)	8	RW	01h
63Ch	(TRSV_REG08F)	8	RW	40h
640h	(TRSV_REG090)	8	RW	03h
644h	(TRSV_REG091)	8	RW	03h
648h	(TRSV_REG092)	8	RW	03h
64Ch	(TRSV_REG093)	8	RW	05h
650h	(TRSV_REG094)	8	RW	0Ah
654h	(TRSV_REG095)	8	RW	10h
658h	(TRSV_REG096)	8	RW	10h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
65Ch	(TRSV_REG097)	8	RW	05h
660h	(TRSV_REG098)	8	RW	00h
664h	(TRSV_REG099)	8	RW	06h
668h	(TRSV_REG09A)	8	RW	FFh
66Ch	(TRSV_REG09B)	8	RW	FFh
670h	(TRSV_REG09C)	8	RW	FFh
674h	(TRSV_REG09D)	8	RW	CCh
678h	(TRSV_REG09E)	8	RW	F7h
67Ch	(TRSV_REG09F)	8	RW	44h
680h	(TRSV_REG0A0)	8	RW	FFh
684h	(TRSV_REG0A1)	8	RW	FFh
688h	(TRSV_REG0A2)	8	RW	FFh
68Ch	(TRSV_REG0A3)	8	RW	A4h
690h	(TRSV_REG0A4)	8	RW	F5h
694h	(TRSV_REG0A5)	8	RW	20h
698h	(TRSV_REG0A6)	8	RW	03h
69Ch	(TRSV_REG0A7)	8	RW	02h
6A0h	(TRSV_REG0A8)	8	RW	25h
6A4h	(TRSV_REG0A9)	8	RW	03h
6A8h	(TRSV_REG0AA)	8	RW	13h
6ACh	(TRSV_REG0AB)	8	RW	24h
6B0h	(TRSV_REG0AC)	8	RW	40h
6B4h	(TRSV_REG0AD)	8	RW	80h
6B8h	(TRSV_REG0AE)	8	RW	33h
6BCh	(TRSV_REG0AF)	8	RW	01h
6C0h	(TRSV_REG0B0)	8	RW	00h
6C4h	(TRSV_REG0B1)	8	RW	00h
6C8h	(TRSV_REG0B2)	8	RW	00h
6CCh	(TRSV_REG0B3)	8	RW	00h
6D0h	(TRSV_REG0B4)	8	RW	00h
6D4h	(TRSV_REG0B5)	8	RW	07h
6D8h	(TRSV_REG0B6)	8	RW	04h
6DCh	(TRSV_REG0B7)	8	RW	00h
6E0h	(TRSV_REG0B8)	8	RW	00h
6E4h	(TRSV_REG0B9)	8	RW	00h
6E8h	(TRSV_REG0BA)	8	RW	00h
6ECh	(TRSV_REG0BB)	8	RW	00h
6F0h	(TRSV_REG0BC)	8	RW	08h
6F4h	(TRSV_REG0BD)	8	RW	00h
6F8h	(TRSV_REG0BE)	8	RW	00h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
6FCh	(TRSV_REG0BF)	8	RW	00h
700h	(TRSV_REG0C0)	8	RW	00h
704h	(TRSV_REG0C1)	8	RW	00h
708h	(TRSV_REG0C2)	8	RW	00h
70Ch	(TRSV_REG0C3)	8	RW	00h
710h	(TRSV_REG0C4)	8	RW	00h
714h	(TRSV_REG0C5)	8	RW	00h
718h	(TRSV_REG0C6)	8	RW	00h
71Ch	(TRSV_REG0C7)	8	RW	00h
720h	(TRSV_REG0C8)	8	RW	00h
724h	(TRSV_REG0C9)	8	RW	00h
728h	(TRSV_REG0CA)	8	RW	00h
72Ch	(TRSV_REG0CB)	8	RW	00h
730h	(TRSV_REG0CC)	8	RW	00h
734h	(TRSV_REG0CD)	8	RW	00h
738h	(TRSV_REG0CE)	8	RW	31h
73Ch	(TRSV_REG0CF)	8	RW	00h
740h	(TRSV_REG0D0)	8	RW	00h
744h	(TRSV_REG0D1)	8	RW	01h
748h	(TRSV_REG0D2)	8	RW	00h
74Ch	(TRSV_REG0D3)	8	RW	00h
750h	(TRSV_REG0D4)	8	RW	00h
754h	(TRSV_REG0D5)	8	RW	00h
758h	(TRSV_REG0D6)	8	RW	00h
75Ch	(TRSV_REG0D7)	8	RW	00h
760h	(TRSV_REG0D8)	8	RW	00h
764h	(TRSV_REG0D9)	8	RW	00h
768h	(TRSV_REG0DA)	8	RW	00h
76Ch	(TRSV_REG0DB)	8	RW	00h
770h	(TRSV_REG0DC)	8	RW	00h
774h	(TRSV_REG0DD)	8	RW	00h
778h	(TRSV_REG0DE)	8	RW	00h
77Ch	(TRSV_REG0DF)	8	RW	00h
780h	(TRSV_REG0E0)	8	RW	00h
784h	(TRSV_REG0E1)	8	RW	00h
788h	(TRSV_REG0E2)	8	RW	00h
78Ch	(TRSV_REG0E3)	8	RW	00h
790h	(TRSV_REG0E4)	8	RW	00h
794h	(TRSV_REG0E5)	8	RW	00h
798h	(TRSV_REG0E6)	8	RW	00h

Table continues on the next page...

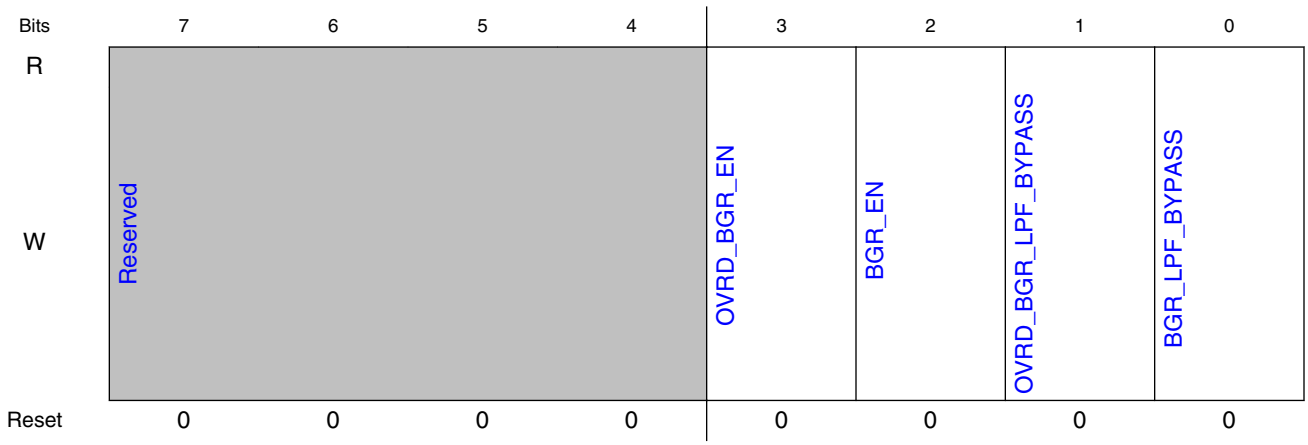
Offset	Register	Width (In bits)	Access	Reset value
79Ch	(TRSV_REG0E7)	8	RW	00h
7A0h	(TRSV_REG0E8)	8	RW	00h
7A4h	(TRSV_REG0E9)	8	RW	00h
7A8h	(TRSV_REG0EA)	8	RW	00h
7ACh	(TRSV_REG0EB)	8	RW	00h
7B0h	(TRSV_REG0EC)	8	RW	00h
7B4h	(TRSV_REG0ED)	8	RW	00h
7B8h	(TRSV_REG0EE)	8	RW	00h
7BCh	(TRSV_REG0EF)	8	RW	00h
7C0h	(TRSV_REG0F0)	8	RW	00h
7C4h	(TRSV_REG0F1)	8	RW	00h
7C8h	(TRSV_REG0F2)	8	RW	00h
7CCh	(TRSV_REG0F3)	8	RW	00h
7D0h	(TRSV_REG0F4)	8	RW	00h
7D4h	(TRSV_REG0F5)	8	RW	00h
7D8h	(TRSV_REG0F6)	8	RW	00h
7DCh	(TRSV_REG0F7)	8	RW	00h
7E0h	(TRSV_REG0F8)	8	RW	00h
7E4h	(TRSV_REG0F9)	8	RW	00h
7E8h	(TRSV_REG0FA)	8	RW	00h
7ECh	(TRSV_REG0FB)	8	RW	00h
7F0h	(TRSV_REG0FC)	8	RW	00h
7F4h	(TRSV_REG0FD)	8	RW	00h
7F8h	(TRSV_REG0FE)	8	RW	1Fh
7FCh	(TRSV_REG0FF)	8	RW	18h

11.5.3.1.2 (CMN_REG000)

11.5.3.1.2.1 Offset

Register	Offset
CMN_REG000	0h

11.5.3.1.2.2 Diagram



11.5.3.1.2.3 Fields

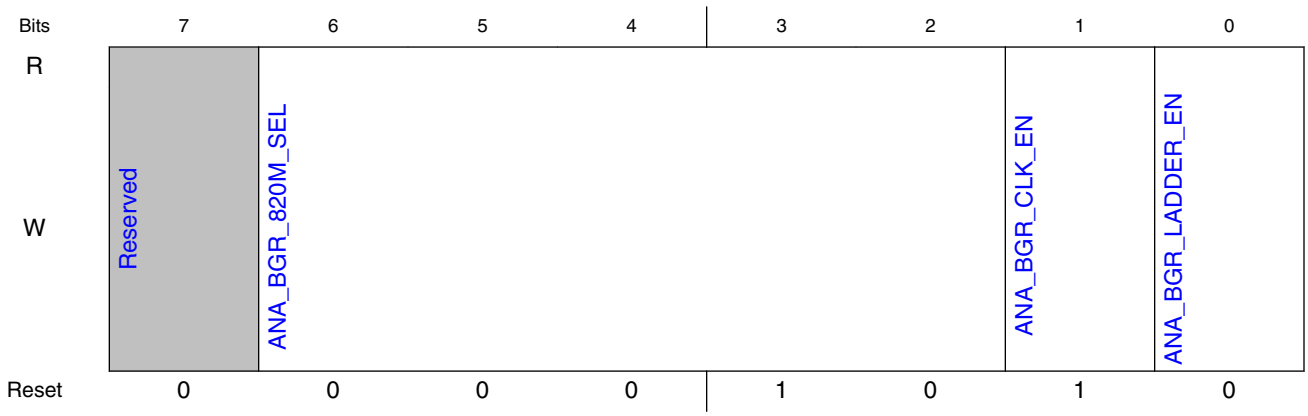
Field	Function
7-4 —	Reserved
3 OVRD_BGR_EN	Override enable for bgr_en
2 BGR_EN	BGR enable 0: Disable, 1: Enable
1 OVRD_BGR_LPF_BYPASS	Override enable for bgr_lpf_bypass
0 BGR_LPF_BYPASS	BGR LPF bypass to reduce BGR settle time 0: LPF enable, 1: LPF bypass

11.5.3.1.3 (CMN_REG001)

11.5.3.1.3.1 Offset

Register	Offset
CMN_REG001	4h

11.5.3.1.3.2 Diagram



11.5.3.1.3.3 Fields

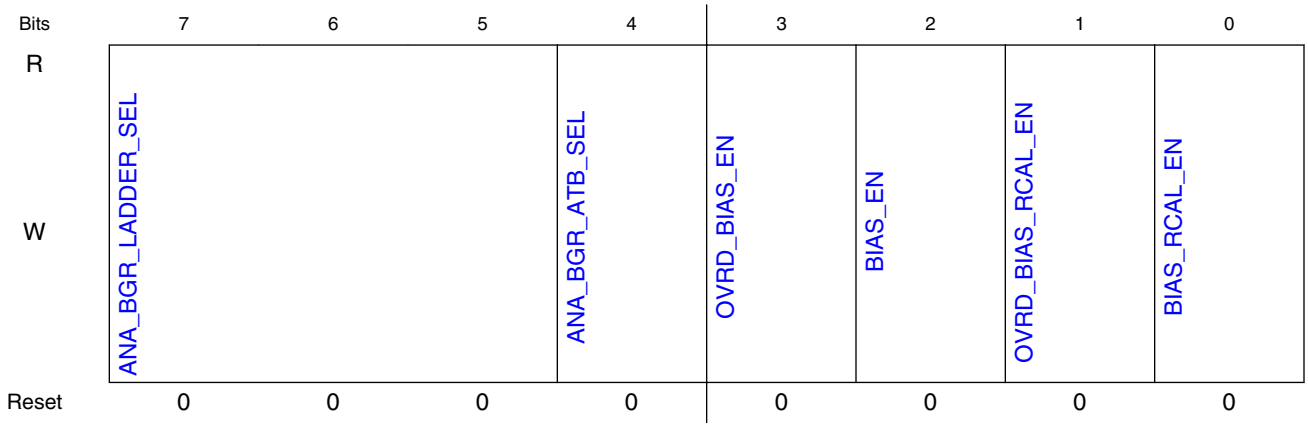
Field	Function
7 —	Reserved
6-2 ANA_BGR_820M_SEL	BGR 820mV selection (for current bias) 00001: 800mV 00010: 820mV 00100: 840mV 01000: 860mV 10000: 900mV
1 ANA_BGR_CLK_EN	BGR chopper clock enable 0: Disable, 1: Enable
0 ANA_BGR_LADDER_EN	BGR output voltage selection 0: BGR output, 1: Resistor ladder output

11.5.3.1.4 (CMN_REG002)

11.5.3.1.4.1 Offset

Register	Offset
CMN_REG002	8h

11.5.3.1.4.2 Diagram



11.5.3.1.4.3 Fields

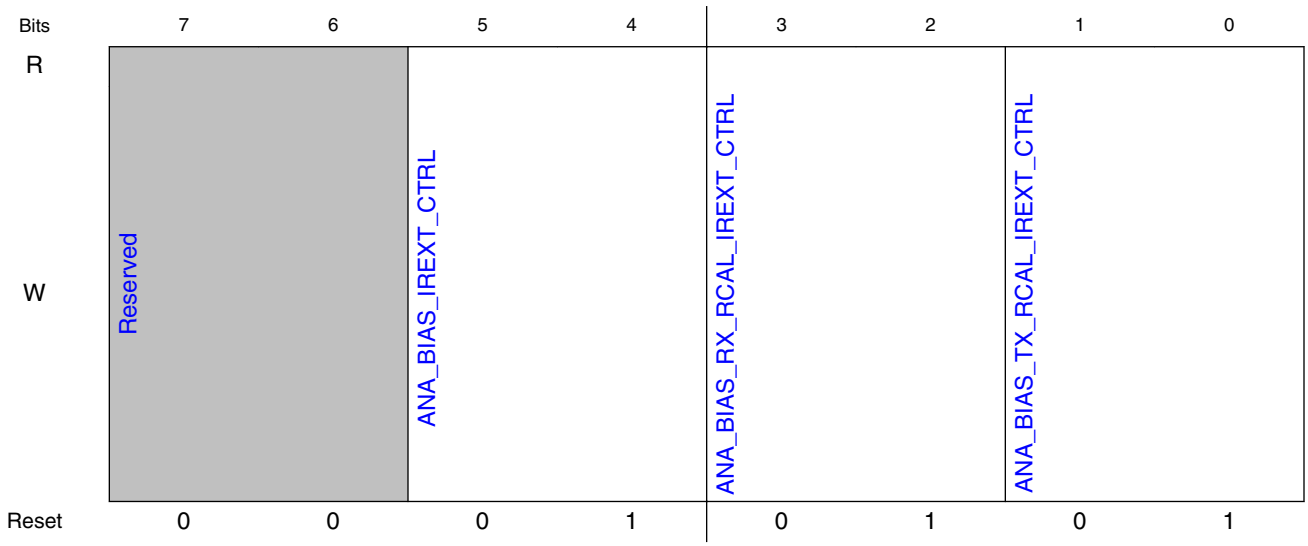
Field	Function
7-5 ANA_BGR_LADDER_SEL	Resistor ladder voltage selection 000: 1000mV, 001: 960mV, 010: 920mV, 011: 880mV, 100: 840mV, 101: 800mV, 110: 760mV, 111: 720mV
4 ANA_BGR_ATB_SEL	BGR ATB select
3 OVRD_BIAS_EN	Override enable for bias_en
2 BIAS_EN	Bias current enable 0: Disable, 1: Enable
1 OVRD_BIAS_RCAL_EN	Override enable for bias_rcal_en
0 BIAS_RCAL_EN	RX RCAL bias current enable 0: Disable, 1: Enable

11.5.3.1.5 (CMN_REG003)

11.5.3.1.5.1 Offset

Register	Offset
CMN_REG003	Ch

11.5.3.1.5.2 Diagram



11.5.3.1.5.3 Fields

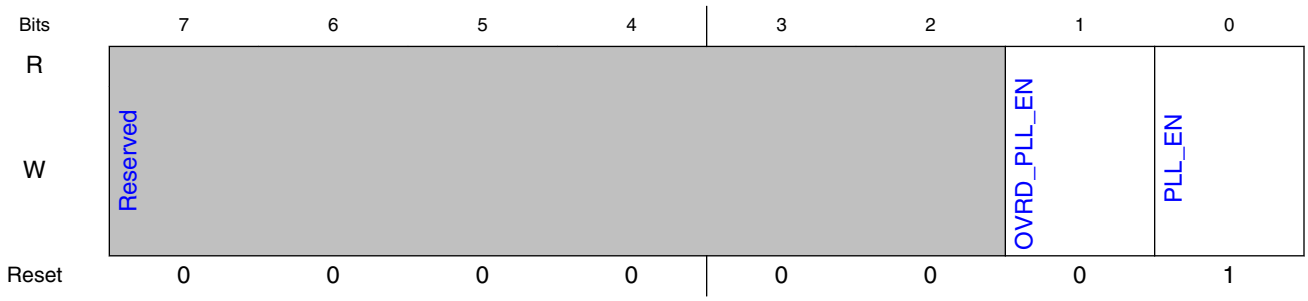
Field	Function
7-6 —	Reserved
5-4 ANA_BIAS_IREXT_CTRL	REXT-refered bias current controll for overall IP 00: 80uA 01: 100uA 10: 132uA
3-2 ANA_BIAS_RX_RCAL_IREXT_CTRL	REXT-refered bias current controll for RX RCAL 00: 85uA 01: 100uA 11: 115uA
1-0 ANA_BIAS_TX_RCAL_IREXT_CTRL	REXT-refered bias current control MSB for TX RCAL

11.5.3.1.6 (CMN_REG004)

11.5.3.1.6.1 Offset

Register	Offset
CMN_REG004	10h

11.5.3.1.6.2 Diagram



11.5.3.1.6.3 Fields

Field	Function
7-2 —	Reserved
1 OVRD_PLL_EN	Override enable for pll_en
0 PLL_EN	PLL enable 0: Disable, 1: Enable

11.5.3.1.7 (CMN_REG005)

11.5.3.1.7.1 Offset

Register	Offset
CMN_REG005	14h

11.5.3.1.7.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	PLL_VCO_MODE_G1	PLL_VCO_MODE_G2	PLL_VCO_MODE_G3	PLL_VCO_MODE_G4	OVRD_PLL_AFC_INIT_RSTN	PLL_AFC_INIT_RSTN	OVRD_PLL_AFC_RSTN	PLL_AFC_RSTN
W								
Reset	1	1	1	1	0	0	0	0

11.5.3.1.7.3 Fields

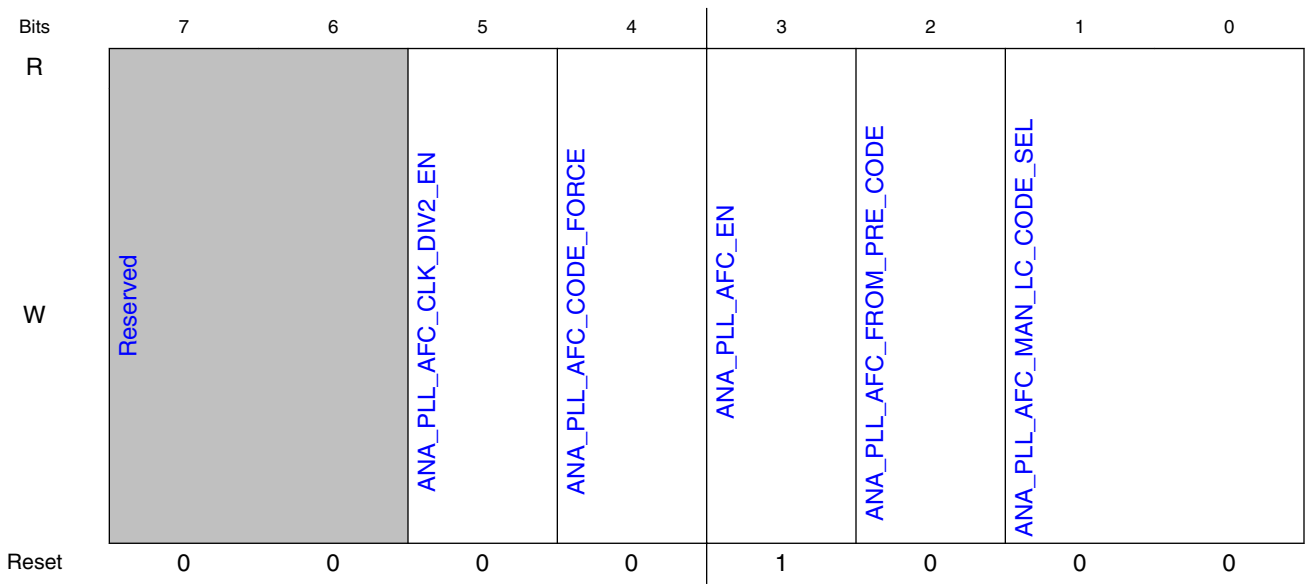
Field	Function
7 PLL_VCO_MODE_G1	[GEN1] PLL VCO selection 0: Ring VCO, 1: LC VCO
6 PLL_VCO_MODE_G2	[GEN2]
5 PLL_VCO_MODE_G3	[GEN3]
4 PLL_VCO_MODE_G4	[GEN4]
3 OVRD_PLL_AFC_INIT_RSTN	Override enable for pll_afc_init_rstn
2 PLL_AFC_INIT_RSTN	PLL AFC initial reset. When initial reset is asserted, the previous AFC result is reset. When initial reset is released, AFC starts from the initial AFC code given in i_rx_cdr_afc_sel_logic[3:0]. 0: Reset, 1: Released
1 OVRD_PLL_AFC_RSTN	Override enable for pll_afc_rstn
0 PLL_AFC_RSTN	PLL AFC reset. When AFC reset is asserted, the previous AFC result is held. When AFC reset is released, AFC starts from the previous AFC code stored in internal memory. 0: Reset, 1: Released

11.5.3.1.8 (CMN_REG006)

11.5.3.1.8.1 Offset

Register	Offset
CMN_REG006	18h

11.5.3.1.8.2 Diagram



11.5.3.1.8.3 Fields

Field	Function
7-6 —	Reserved
5 ANA_PLL_AFC_CLK_DIV2_EN	PLL AFC clock frequency selection 0: fVCO, 1: fVCO/2
4 ANA_PLL_AFC_CODE_FORCE	PLL AFC code manual selection enable 0: AFC result, 1: Manual selection
3 ANA_PLL_AFC_EN	PLL AFC enable; if enabled, VCO frequency is automatically calibrated. If disabled, VCO starts to oscillate with fixed AFC code of i_pll_man_bsel_m and _l. 0: Disable, 1: Enable
2	PLL AFC option in restart case

Table continues on the next page...

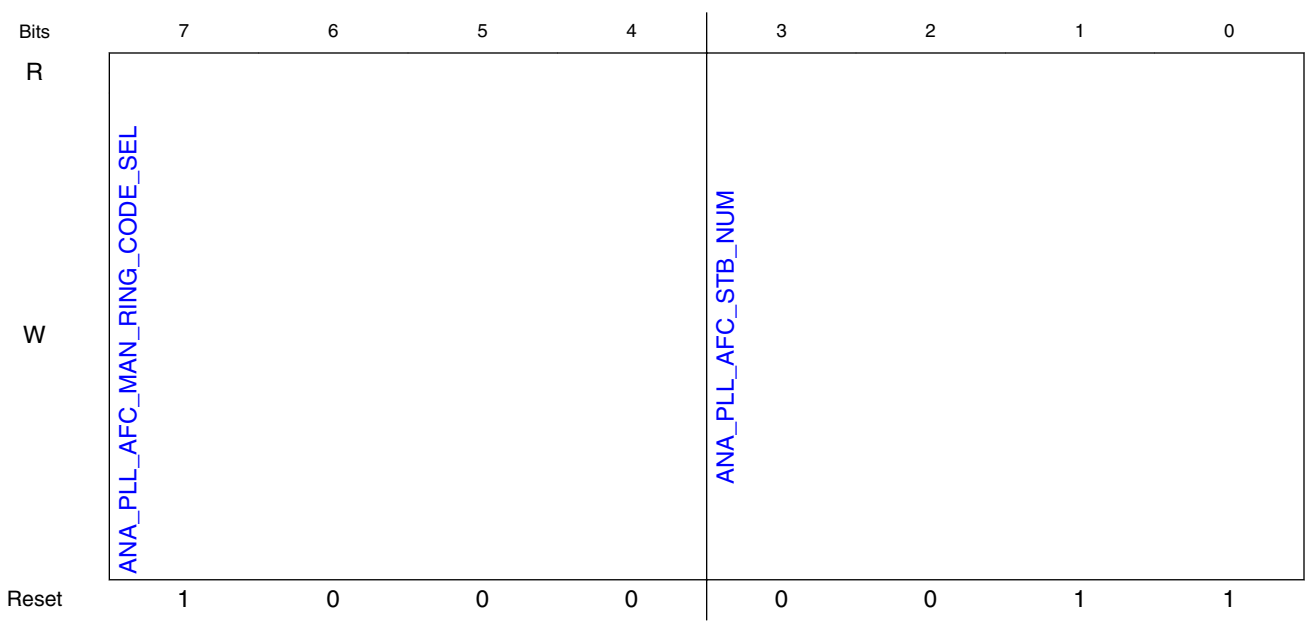
Field	Function
ANA_PLL_AFC_FROM_PRE_CODE	0: Restart from initial AFC code, 1: Restart from previous AFC code
1-0 ANA_PLL_AFC_MAN_LC_CODE_SEL	Manual PLL AFC code selection (MSB)

11.5.3.1.9 (CMN_REG007)

11.5.3.1.9.1 Offset

Register	Offset
CMN_REG007	1Ch

11.5.3.1.9.2 Diagram



11.5.3.1.9.3 Fields

Field	Function
7-4	Manual PLL AFC code selection (LSB)

Table continues on the next page...

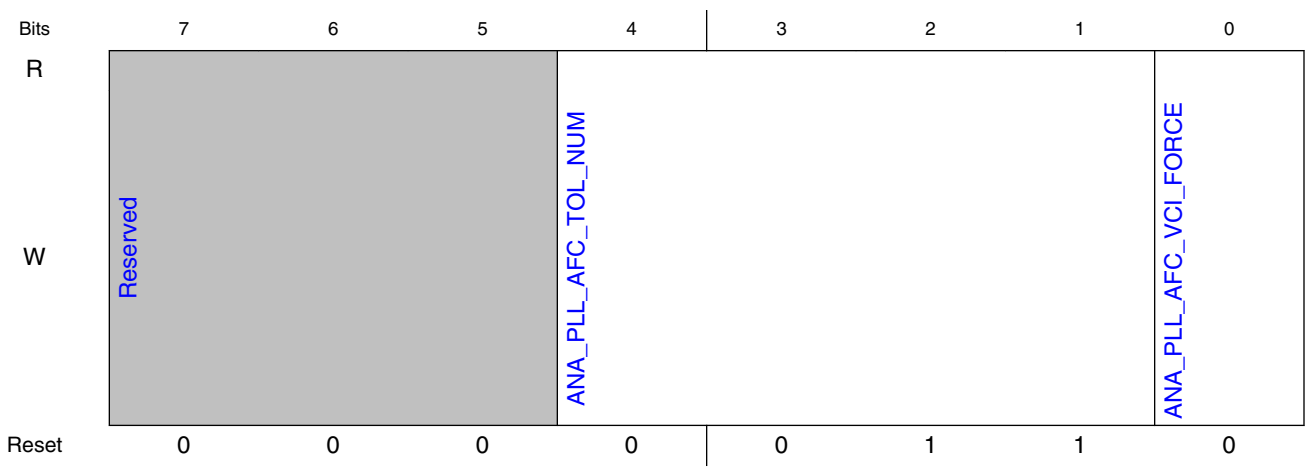
Field	Function
ANA_PLL_AFC_MAN_RING_CODE_SEL	
3-0 ANA_PLL_AFC_STB_NUM	Number of reference clock cycle to check VCO stabilization during PLL AFC start

11.5.3.1.10 (CMN_REG008)

11.5.3.1.10.1 Offset

Register	Offset
CMN_REG008	20h

11.5.3.1.10.2 Diagram



11.5.3.1.10.3 Fields

Field	Function
7-5 —	Reserved
4-1 ANA_PLL_AFC_TOL_NUM	PLL VCO stabilization tolerance; VCO is considered as settled-down if |counter difference|<i_pll_afc_tol during i_pll_afc_stb_num
0	PLL control voltage force for open-loop test purpose

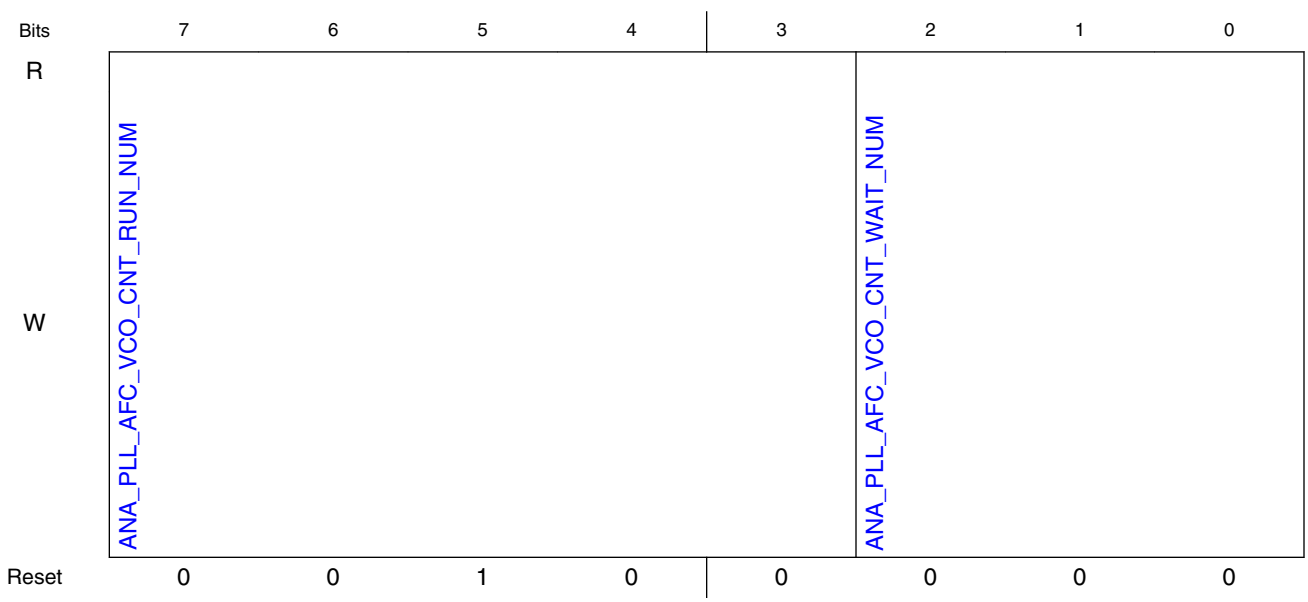
Field	Function
ANA_PLL_AFC_VCI_FORCE	0: Released, 1: Forced

11.5.3.1.11 (CMN_REG009)

11.5.3.1.11.1 Offset

Register	Offset
CMN_REG009	24h

11.5.3.1.11.2 Diagram



11.5.3.1.11.3 Fields

Field	Function
7-3 ANA_PLL_AFC_VCO_CNT_RUN_NUM	Number of reference clock cycle to wait VCO clock during AFC
2-0 ANA_PLL_AFC_VCO_CNT_WAIT_NUM	Number of reference clock cycle to count VCO clock during AFC

11.5.3.1.12 (CMN_REG00A)

11.5.3.1.12.1 Offset

Register	Offset
CMN_REG00A	28h

11.5.3.1.12.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	ANA_PLL_AFC_VCO_START_CRITERION			ANA_PLL_AGMC_COMP_EN	ANA_PLL_AGMC_FROM_MAX_GM	ANA_PLL_AGMC_GM_ADD		ANA_PLL_AGMC_MAN_GM_SEL_EN
W								
Reset	1	0	1	0	1	0	0	1

11.5.3.1.12.3 Fields

Field	Function
7-5 ANA_PLL_AFC_VCO_START_CRITERION	Minimum PLL VCO counter value to start AFC (Criterion to suppose PLL VCO successfully start to oscillate)
4 ANA_PLL_AGMC_COMP_EN	Comparator enable for PLL LC VCO automatic gm search 0: Disable, 1: Enable
3	PLL LC VCO automatic gm search initial condition 0: Start from min gm condition, 1:Start from max gm condition

Table continues on the next page...

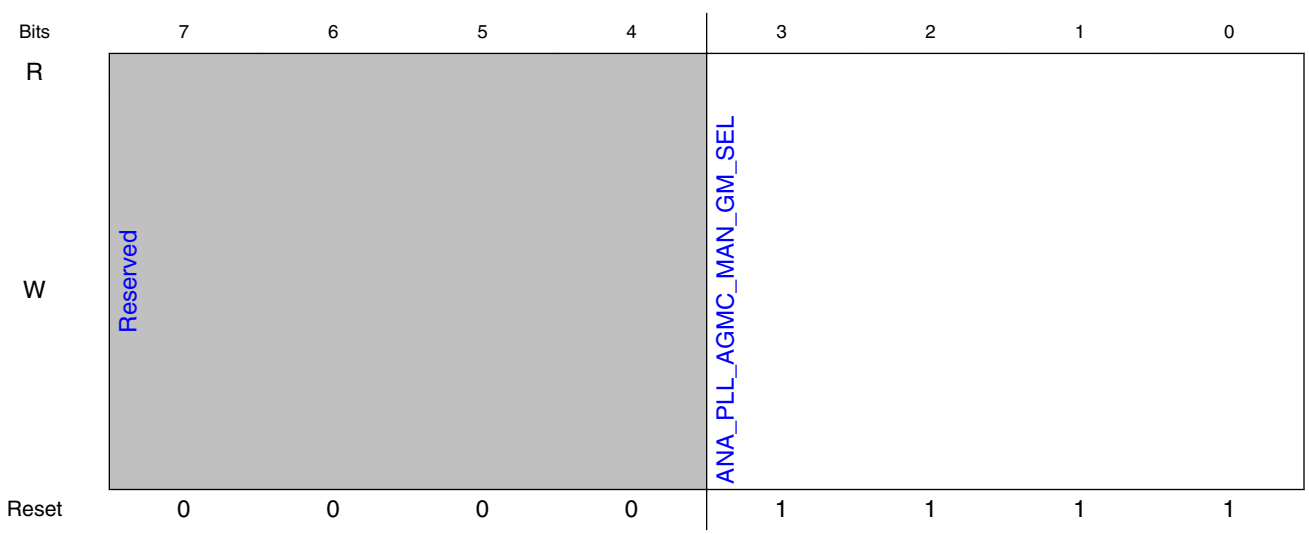
Field	Function
ANA_PLL_AGMC_FROM_MAX_GM	
2-1 ANA_PLL_AGMC_GM_ADD	Offset code to be added to final gm code 00: Min, … , 11: Max
0 ANA_PLL_AGMC_MAN_GM_SEL_EN	PLL LC VCO GM code selection 0: AGMC result, 1: Manual selection

11.5.3.1.13 (CMN_REG00B)

11.5.3.1.13.1 Offset

Register	Offset
CMN_REG00B	2Ch

11.5.3.1.13.2 Diagram



11.5.3.1.13.3 Fields

Field	Function
7-4	Reserved

Table continues on the next page...

Field	Function
—	
3-0 ANA_PLL_AGMC_MAN_GM_SEL	Manual GM code selection for LC VCO 0000: GM min (ampitude min), …, 1111: GM max (amplitude max)

11.5.3.1.14 (CMN_REG00C)

11.5.3.1.14.1 Offset

Register	Offset
CMN_REG00C	30h

11.5.3.1.14.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_AGMC_TG_CODE_G1							
Reset	0	1	0	0	0	1	1	0

11.5.3.1.14.3 Fields

Field	Function
7-0 PLL_AGMC_TG_CODE_G1	[GEN1] Target counter value for automatic GM search of LC VCO 4GHz:8'd40 or 8'b00101000 or 8'h28 (Default) 7GHz: 8'd70 or 8'b1000110 or 8'h46

11.5.3.1.15 (CMN_REG00D)

11.5.3.1.15.1 Offset

Register	Offset
CMN_REG00D	34h

11.5.3.1.15.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_AGMC_TG_CODE_G2							
Reset	0	1	0	0	0	1	1	0

11.5.3.1.15.3 Fields

Field	Function
7-0 PLL_AGMC_TG_CODE_G2	[GEN2]

11.5.3.1.16 (CMN_REG00E)

11.5.3.1.16.1 Offset

Register	Offset
CMN_REG00E	38h

11.5.3.1.16.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_AGMC_TG_CODE_G3							
Reset	0	1	0	0	0	1	1	0

11.5.3.1.16.3 Fields

Field	Function
7-0 PLL_AGMC_TG_CODE_G3	[GEN3]

11.5.3.1.17 (CMN_REG00F)

11.5.3.1.17.1 Offset

Register	Offset
CMN_REG00F	3Ch

11.5.3.1.17.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_AGMC_TG_CODE_G4							
Reset	0	1	0	0	0	1	1	0

11.5.3.1.17.3 Fields

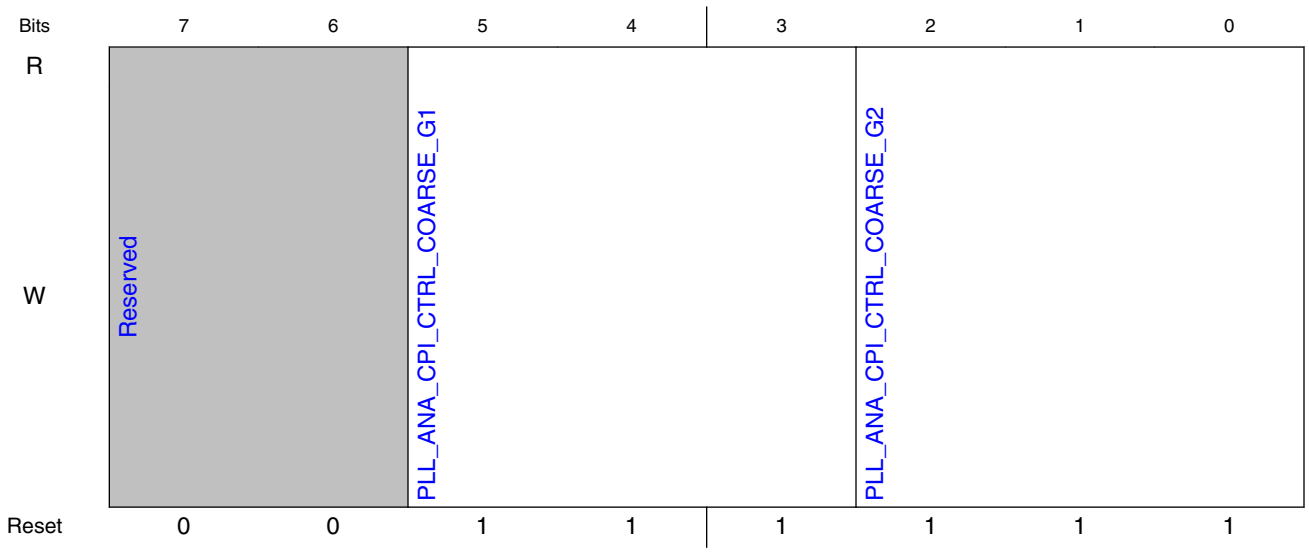
Field	Function
7-0 PLL_AGMC_TG_CODE_G4	[GEN4]

11.5.3.1.18 (CMN_REG010)

11.5.3.1.18.1 Offset

Register	Offset
CMN_REG010	40h

11.5.3.1.18.2 Diagram



11.5.3.1.18.3 Fields

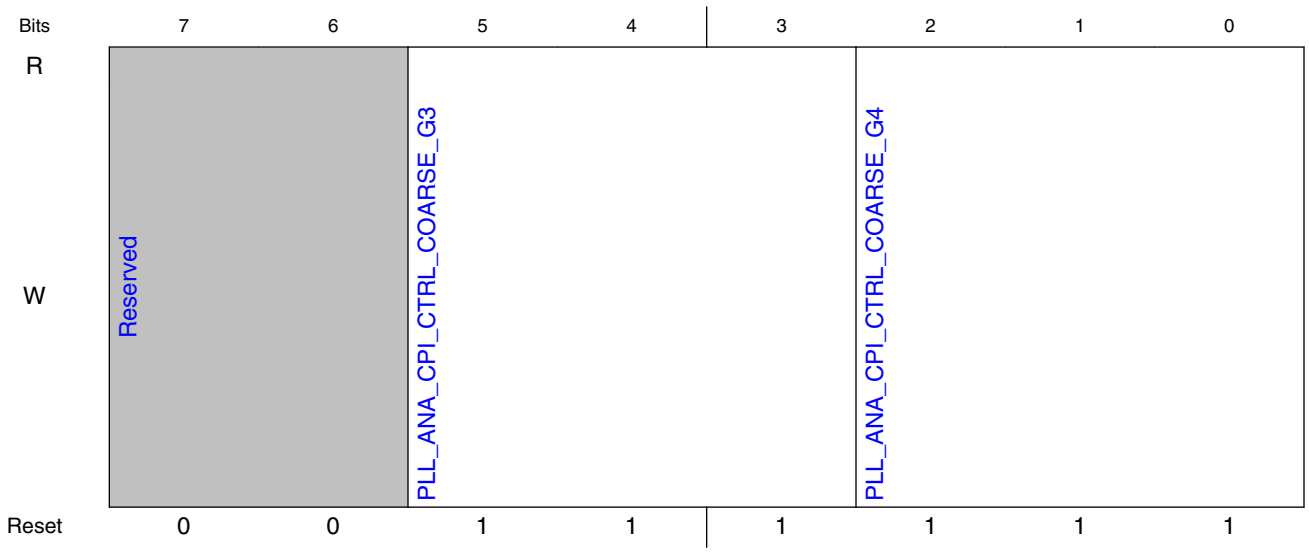
Field	Function
7-6 —	Reserved
5-3 PLL_ANA_CPI_CTRL_COARSE_G1	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA
2-0 PLL_ANA_CPI_CTRL_COARSE_G2	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA

11.5.3.1.19 (CMN_REG011)

11.5.3.1.19.1 Offset

Register	Offset
CMN_REG011	44h

11.5.3.1.19.2 Diagram



11.5.3.1.19.3 Fields

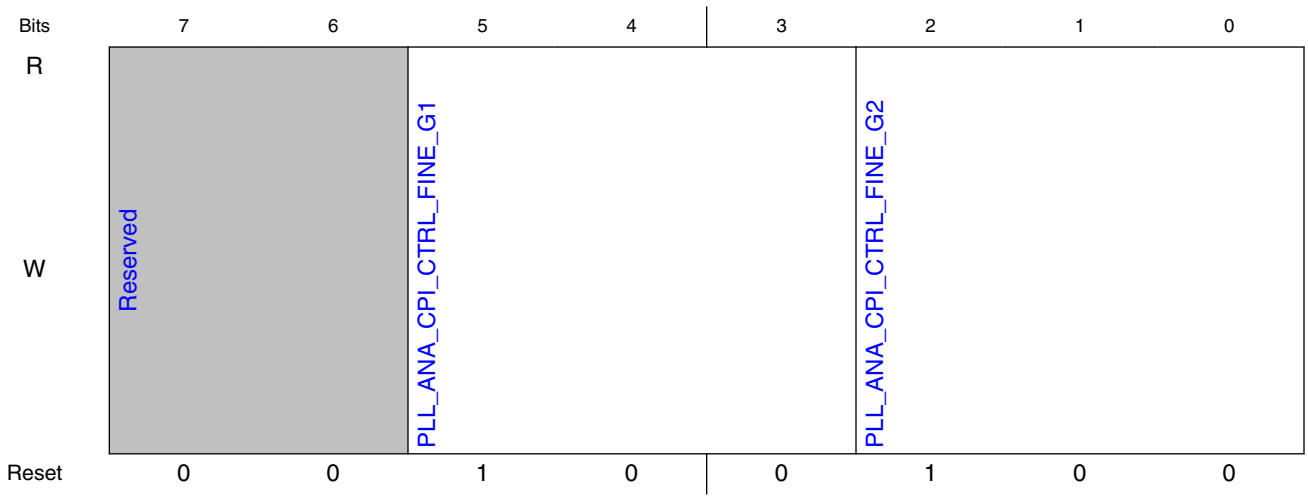
Field	Function
7-6 —	Reserved
5-3 PLL_ANA_CPI_CTRL_COARSE_G3	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA
2-0 PLL_ANA_CPI_CTRL_COARSE_G4	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA

11.5.3.1.20 (CMN_REG012)

11.5.3.1.20.1 Offset

Register	Offset
CMN_REG012	48h

11.5.3.1.20.2 Diagram



11.5.3.1.20.3 Fields

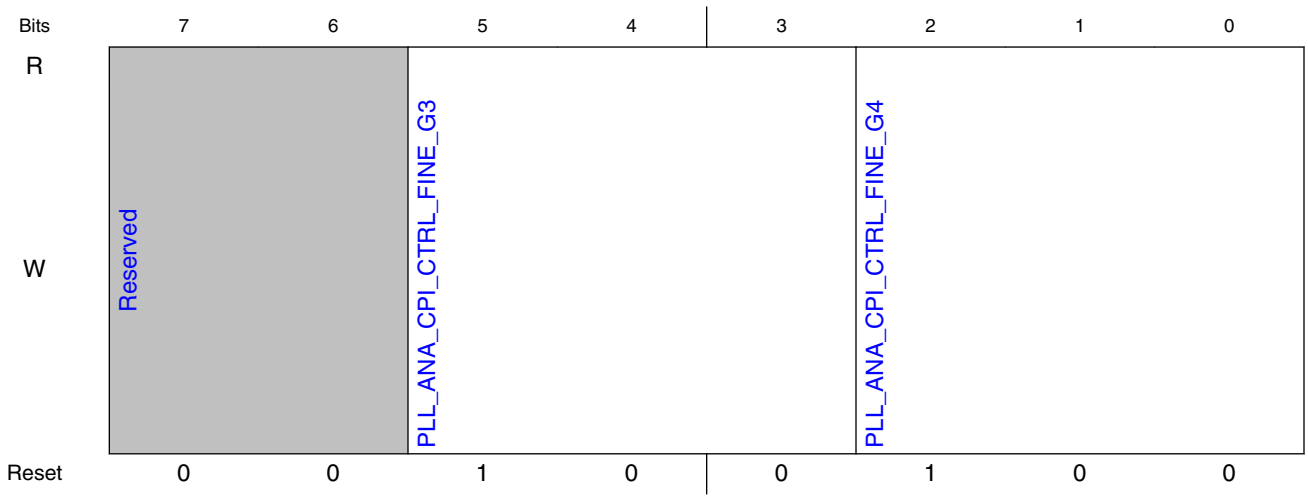
Field	Function
7-6 —	Reserved
5-3 PLL_ANA_CPI_CTRL_FINE_G1	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA
2-0 PLL_ANA_CPI_CTRL_FINE_G2	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA

11.5.3.1.21 (CMN_REG013)

11.5.3.1.21.1 Offset

Register	Offset
CMN_REG013	4Ch

11.5.3.1.21.2 Diagram



11.5.3.1.21.3 Fields

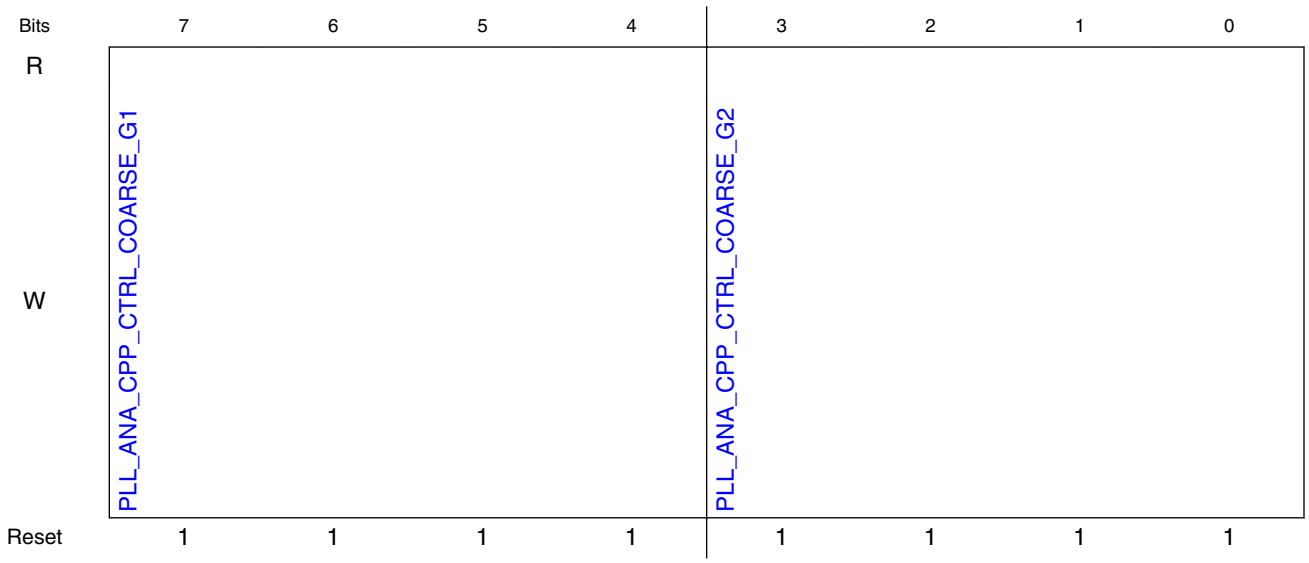
Field	Function
7-6 —	Reserved
5-3 PLL_ANA_CPI_CTRL_FINE_G3	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA
2-0 PLL_ANA_CPI_CTRL_FINE_G4	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA

11.5.3.1.22 (CMN_REG014)

11.5.3.1.22.1 Offset

Register	Offset
CMN_REG014	50h

11.5.3.1.22.2 Diagram



11.5.3.1.22.3 Fields

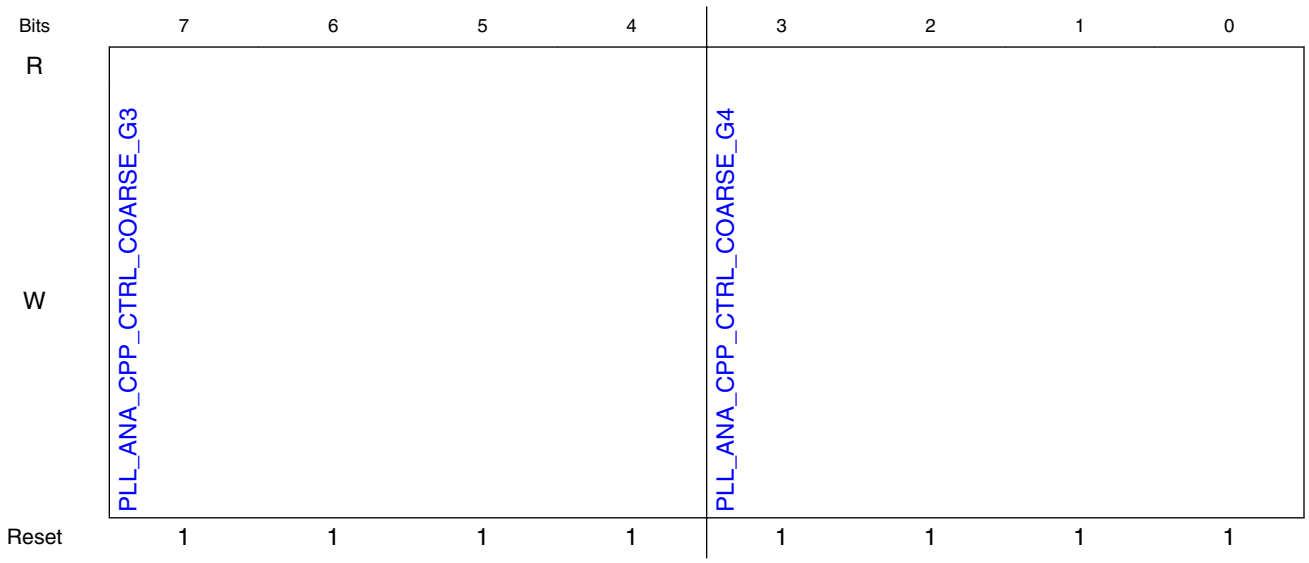
Field	Function
7-4 PLL_ANA_CPP_CTRL_COARSE_G1	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA
3-0 PLL_ANA_CPP_CTRL_COARSE_G2	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA

11.5.3.1.23 (CMN_REG015)

11.5.3.1.23.1 Offset

Register	Offset
CMN_REG015	54h

11.5.3.1.23.2 Diagram



11.5.3.1.23.3 Fields

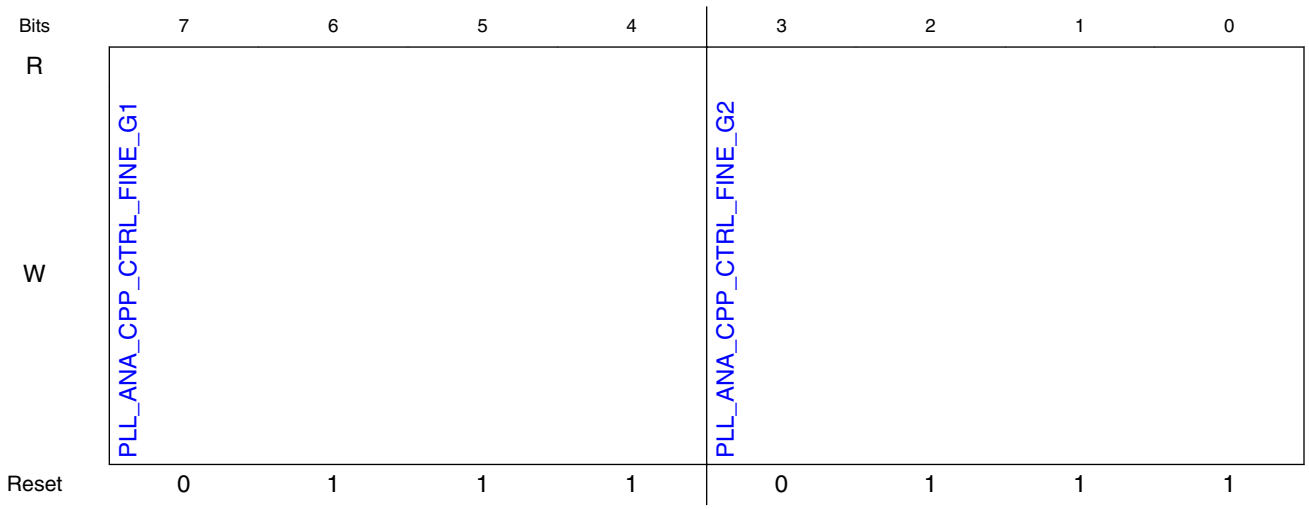
Field	Function
7-4 PLL_ANA_CPP_CTRL_COARSE_G3	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA
3-0 PLL_ANA_CPP_CTRL_COARSE_G4	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA

11.5.3.1.24 (CMN_REG016)

11.5.3.1.24.1 Offset

Register	Offset
CMN_REG016	58h

11.5.3.1.24.2 Diagram



11.5.3.1.24.3 Fields

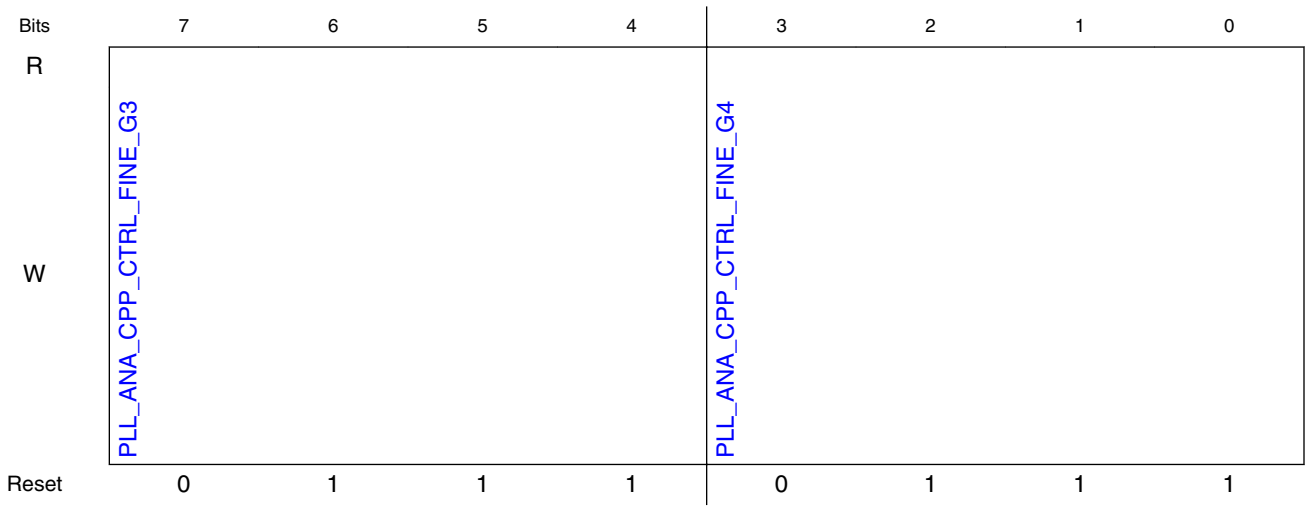
Field	Function
7-4 PLL_ANA_CPP_CTRL_FINE_G1	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA
3-0 PLL_ANA_CPP_CTRL_FINE_G2	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA

11.5.3.1.25 (CMN_REG017)

11.5.3.1.25.1 Offset

Register	Offset
CMN_REG017	5Ch

11.5.3.1.25.2 Diagram



11.5.3.1.25.3 Fields

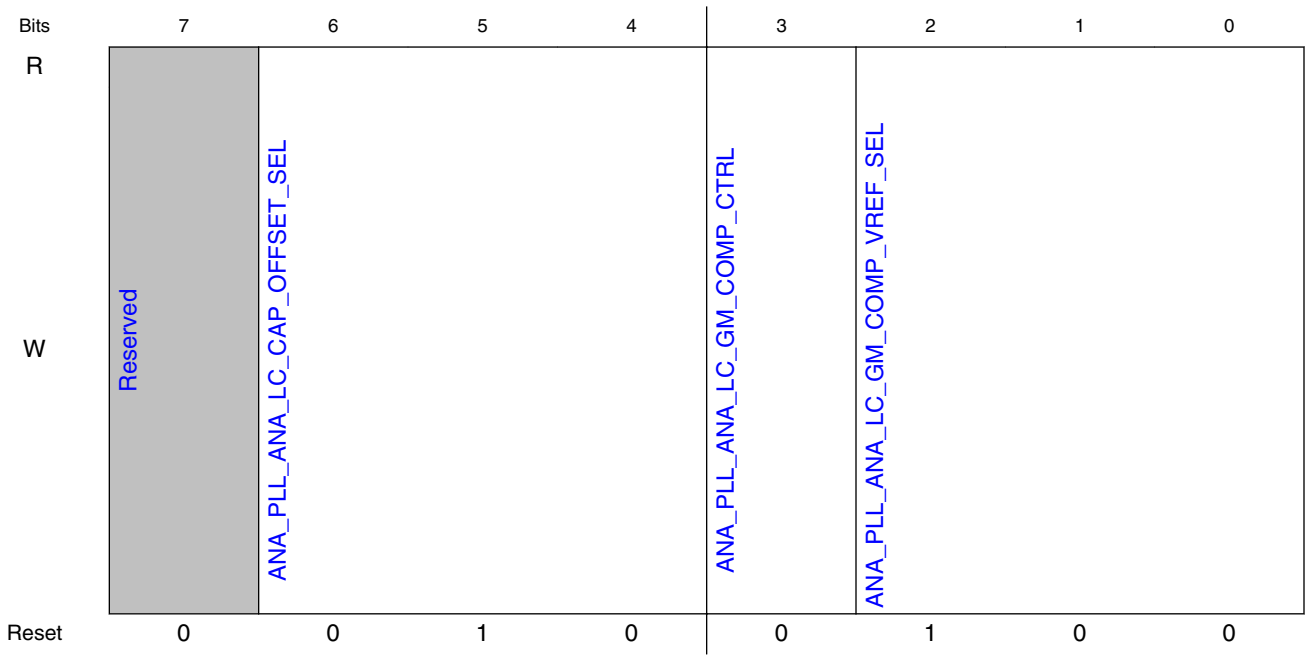
Field	Function
7-4 PLL_ANA_CPP_CTRL_FINE_G3	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA
3-0 PLL_ANA_CPP_CTRL_FINE_G4	[GEN1] PLL integral path charge-pump current contorl 0xx: half current (not used) 100: 1.25uA, 101:1.5625uA, 110: 20833uA, 111: 3.125uA

11.5.3.1.26 (CMN_REG018)

11.5.3.1.26.1 Offset

Register	Offset
CMN_REG018	60h

11.5.3.1.26.2 Diagram



11.5.3.1.26.3 Fields

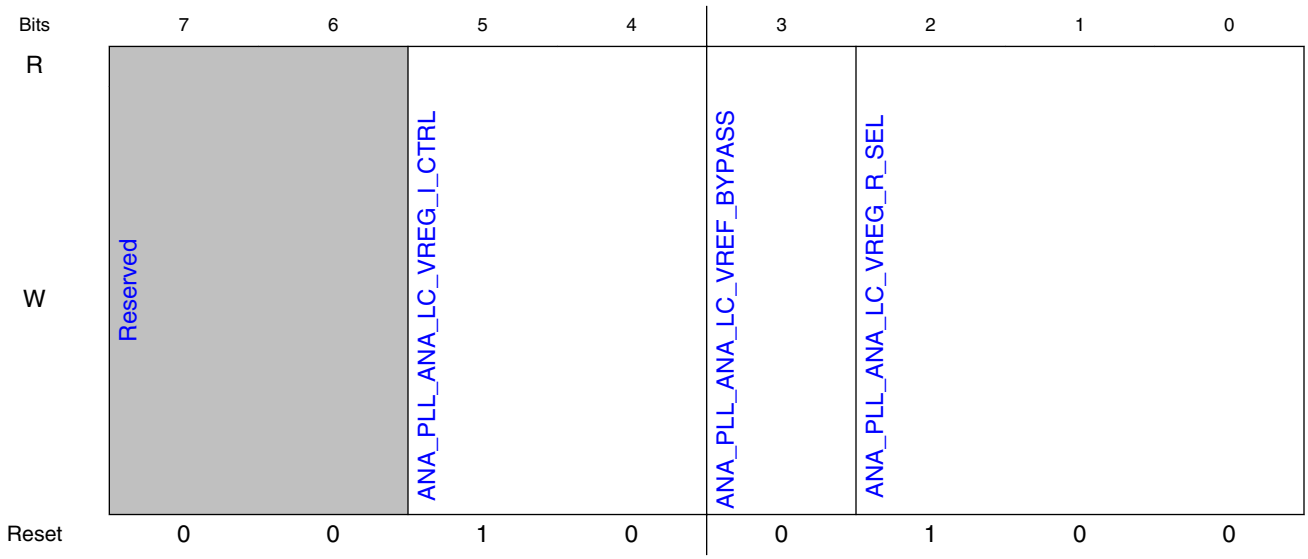
Field	Function
7 —	Reserved
6-4 ANA_PLL_ANA_LC_CAP_OFFSET_SEL	LC VCO varactor bias voltage selection 000: 1.3V, 001: 1.2V, 010: 1.1V, 011: 1.0V, 100: 0.9V, 101: 0.8V, 110: 0.7V, 111: 0.6V:
3 ANA_PLL_ANA_LC_GM_COMP_CTRL	
2-0 ANA_PLL_ANA_LC_GM_COMP_VREF_SEL	PLL GM comparator reference voltage selection 000: 769mV, 001: 724mV, 010: 679mV, 011: 634mV, 100: 588mV, 101: 543mV, 110: 498mV, 111: 454mV

11.5.3.1.27 (CMN_REG019)

11.5.3.1.27.1 Offset

Register	Offset
CMN_REG019	64h

11.5.3.1.27.2 Diagram



11.5.3.1.27.3 Fields

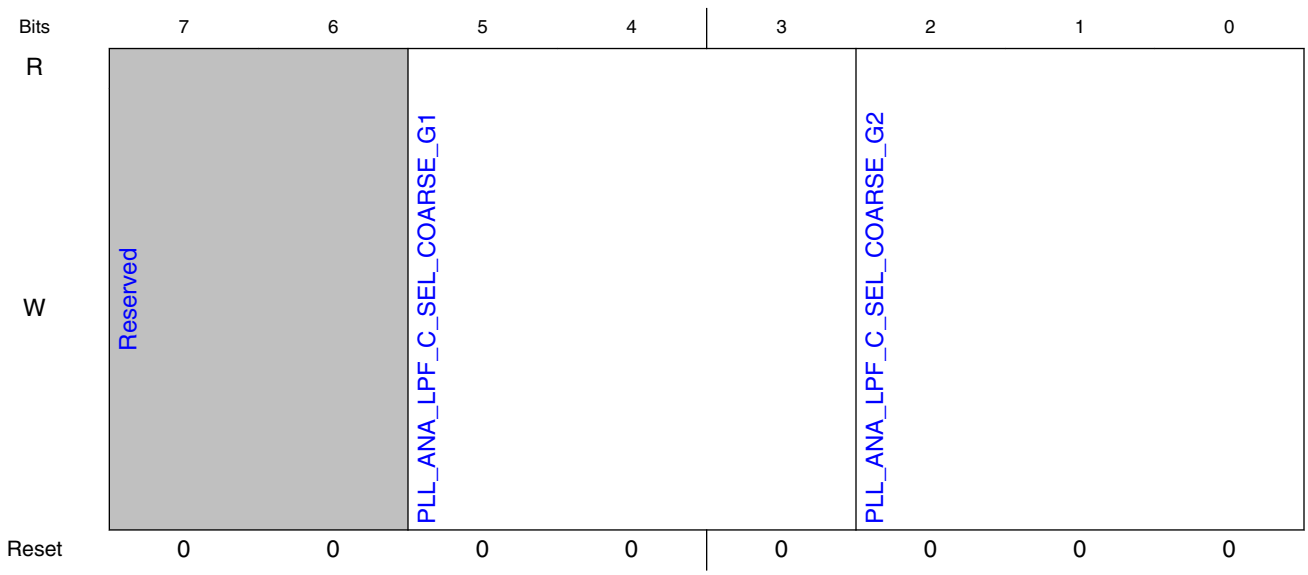
Field	Function
7-6 —	Reserved
5-4 ANA_PLL_ANA_LC_VREG_I_CTRL	LC VCO Vreg current branch enable (1+1/20 or 1+ 1/33)
3 ANA_PLL_ANA_LC_VREF_BYPASS	LPF on reference voltage for PLL LC VCO bypass 0: LPF enable, 1: LPF bypass
2-0 ANA_PLL_ANA_LC_VREG_R_SEL	LC VCO voltage regulator output control 000: 853mV, 001: 863mV, 010: 878mV, 011: 888mV 100: 911mV, 101: 921mV, 110: 936mV, 111: 946mV

11.5.3.1.28 (CMN_REG01A)

11.5.3.1.28.1 Offset

Register	Offset
CMN_REG01A	68h

11.5.3.1.28.2 Diagram



11.5.3.1.28.3 Fields

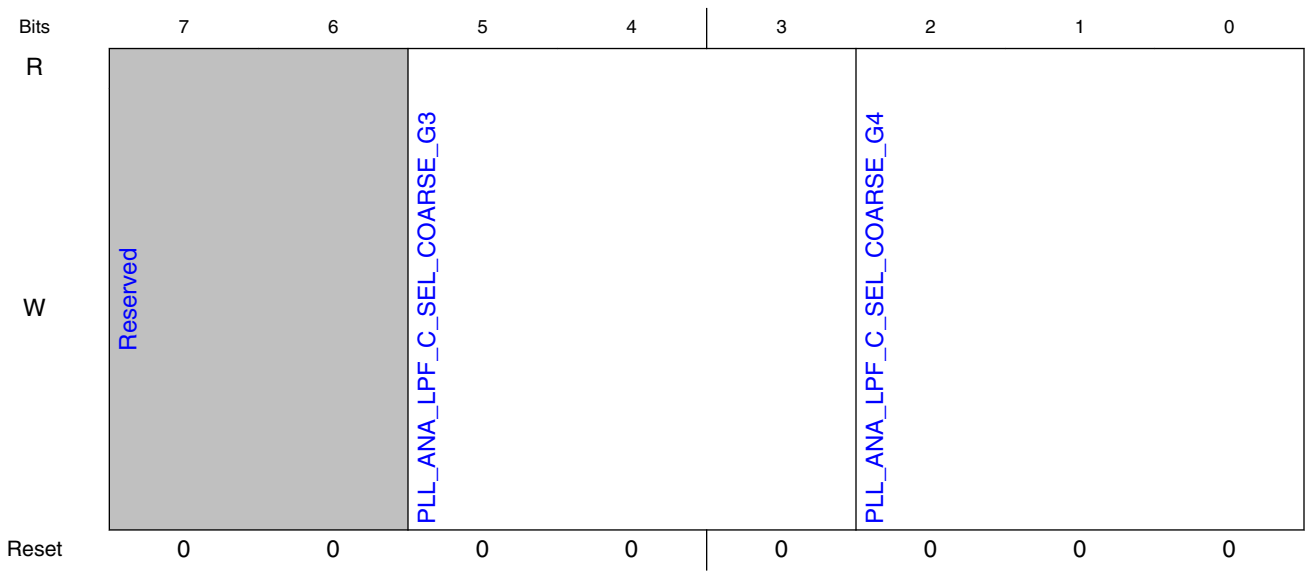
Field	Function
7-6 —	Reserved
5-3 PLL_ANA_LPF_C_SEL_COARSE_G1	[GEN1] [Coarse Mode] PLL loop filter capacitor control 000: 0pF, 001: 1pF, 010: 2pF, 011: 3pF, 100: 4pF, 101: 5pF,110: 6pF, 111: 7pF
2-0 PLL_ANA_LPF_C_SEL_COARSE_G2	[GEN2] [Coarse Mode]

11.5.3.1.29 (CMN_REG01B)

11.5.3.1.29.1 Offset

Register	Offset
CMN_REG01B	6Ch

11.5.3.1.29.2 Diagram



11.5.3.1.29.3 Fields

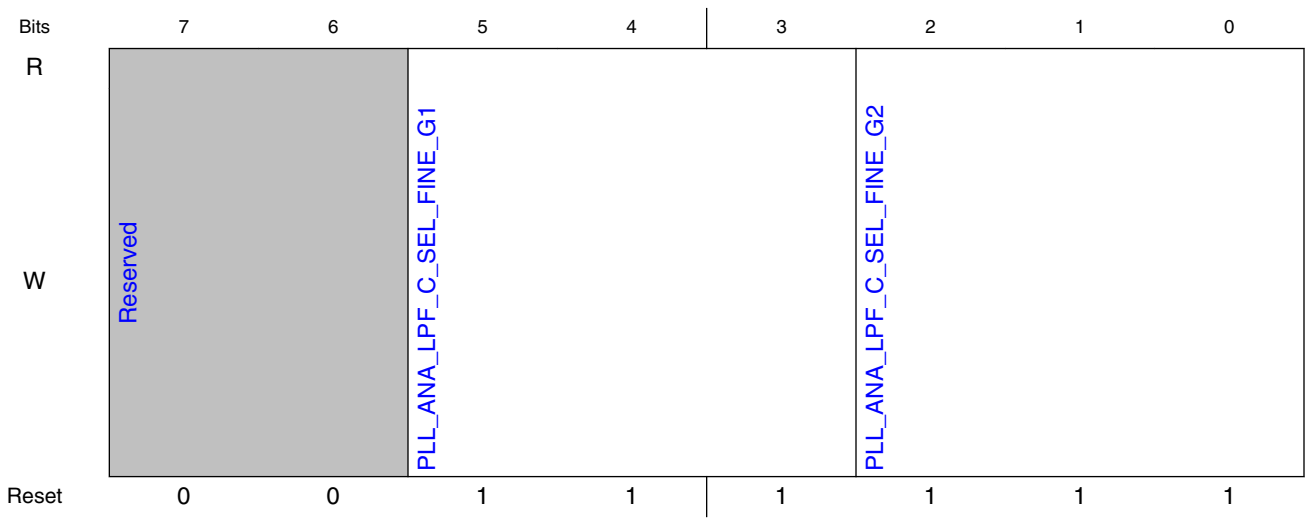
Field	Function
7-6 —	Reserved
5-3 PLL_ANA_LPF_C_SEL_COARSE_G3	[GEN3] [Coarse Mode]
2-0 PLL_ANA_LPF_C_SEL_COARSE_G4	[GEN4] [Coarse Mode]

11.5.3.1.30 (CMN_REG01C)

11.5.3.1.30.1 Offset

Register	Offset
CMN_REG01C	70h

11.5.3.1.30.2 Diagram



11.5.3.1.30.3 Fields

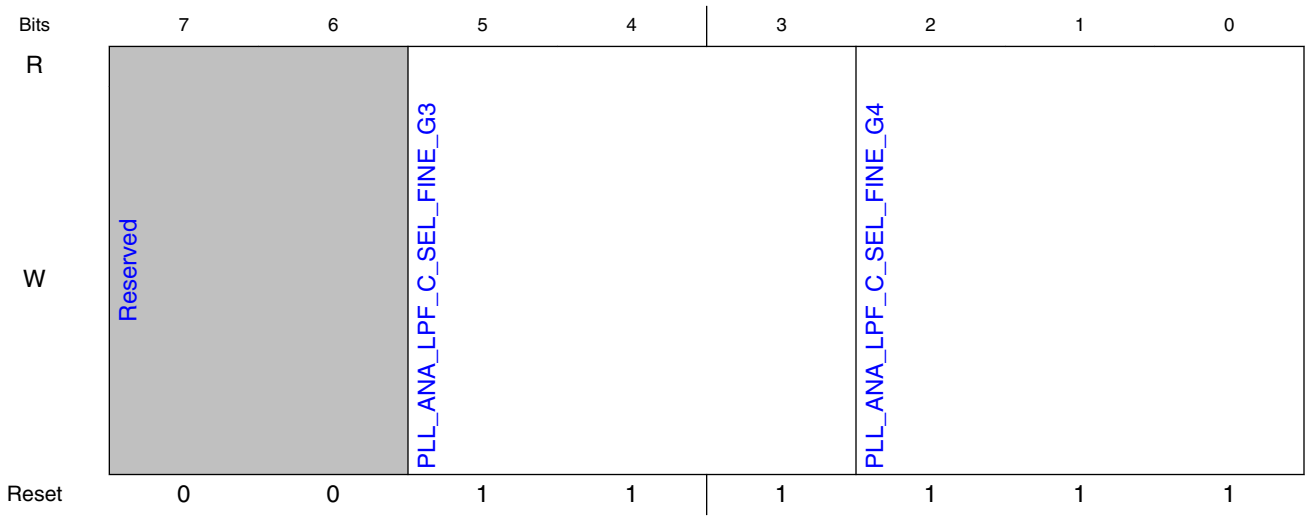
Field	Function
7-6 —	Reserved
5-3 PLL_ANA_LPF_C_SEL_FINE_G1	[GEN1] [Fine Mode] PLL loop filter capacitor control 000: Min, … 111: Max
2-0 PLL_ANA_LPF_C_SEL_FINE_G2	[GEN2] [Fine Mode]

11.5.3.1.31 (CMN_REG01D)

11.5.3.1.31.1 Offset

Register	Offset
CMN_REG01D	74h

11.5.3.1.31.2 Diagram



11.5.3.1.31.3 Fields

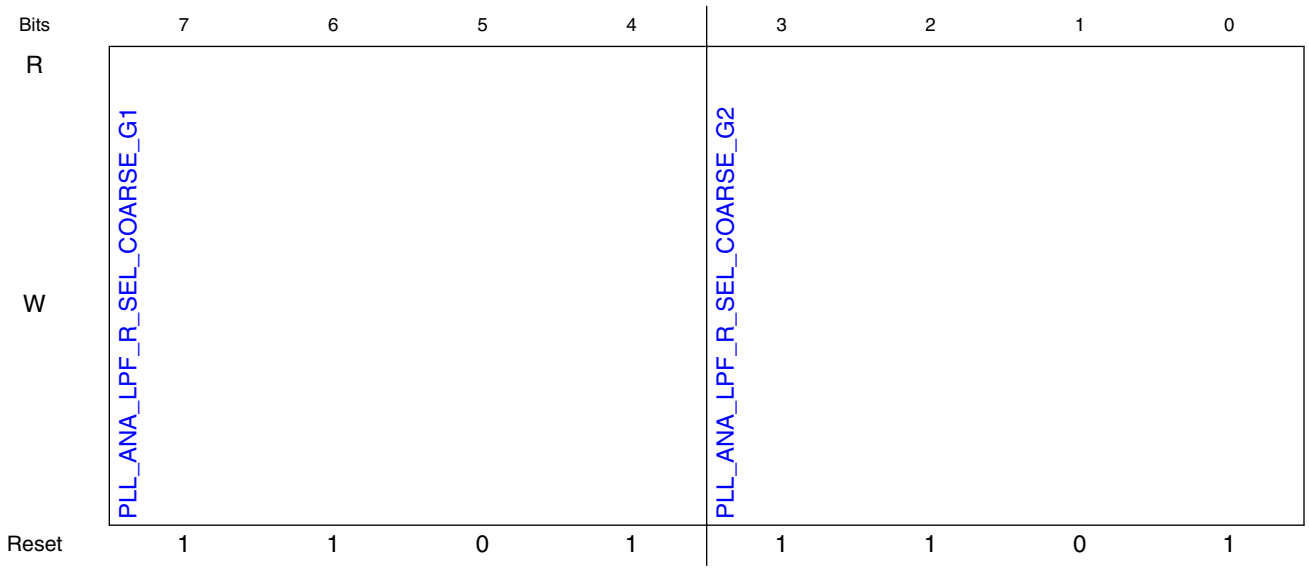
Field	Function
7-6 —	Reserved
5-3 PLL_ANA_LPF_C_SEL_FINE_G3	[GEN3] [Fine Mode]
2-0 PLL_ANA_LPF_C_SEL_FINE_G4	[GEN4] [Fine Mode]

11.5.3.1.32 (CMN_REG01E)

11.5.3.1.32.1 Offset

Register	Offset
CMN_REG01E	78h

11.5.3.1.32.2 Diagram



11.5.3.1.32.3 Fields

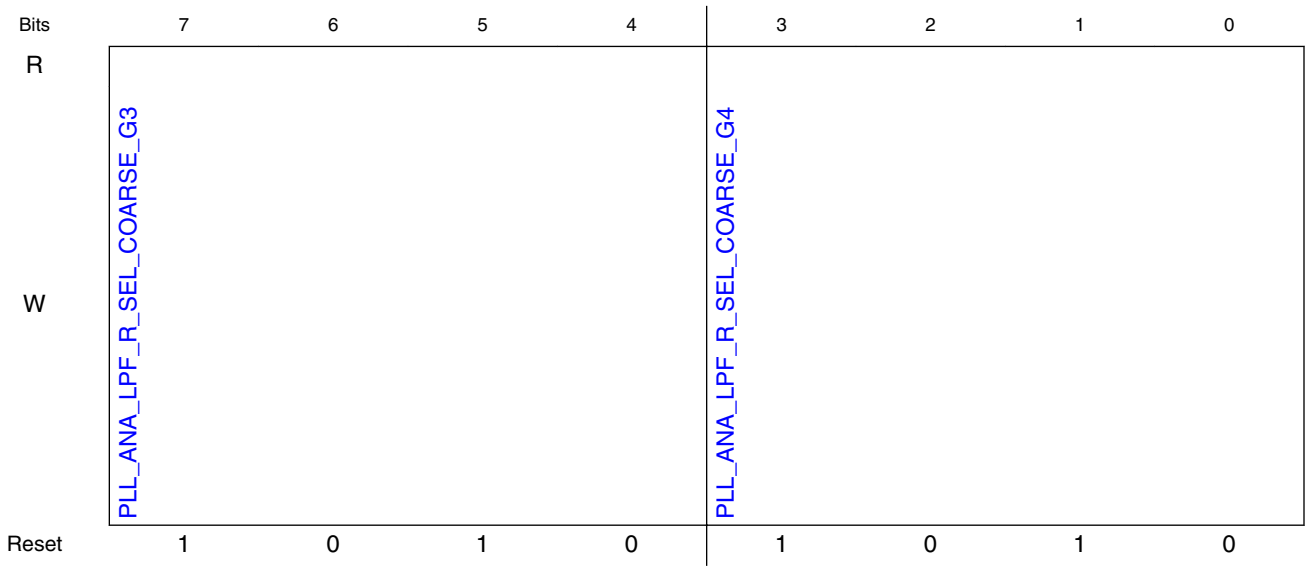
Field	Function
7-4 PLL_ANA_LPF_R_SEL_COARSE_G1	[GEN1] [Coarse Mode] PLL loop filter resistor control 0000: Min, … 1111: Max
3-0 PLL_ANA_LPF_R_SEL_COARSE_G2	[GEN2] [Coarse Mode]

11.5.3.1.33 (CMN_REG01F)

11.5.3.1.33.1 Offset

Register	Offset
CMN_REG01F	7Ch

11.5.3.1.33.2 Diagram



11.5.3.1.33.3 Fields

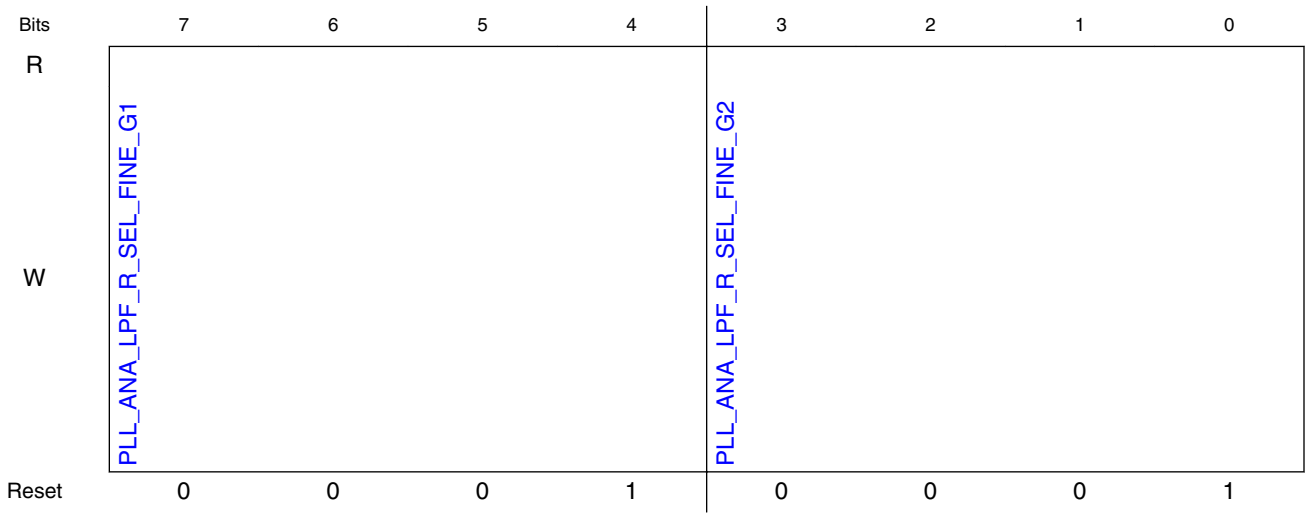
Field	Function
7-4 PLL_ANA_LPF_R_SEL_COARSE_G3	[GEN3] [Coarse Mode]
3-0 PLL_ANA_LPF_R_SEL_COARSE_G4	[GEN4] [Coarse Mode]

11.5.3.1.34 (CMN_REG020)

11.5.3.1.34.1 Offset

Register	Offset
CMN_REG020	80h

11.5.3.1.34.2 Diagram



11.5.3.1.34.3 Fields

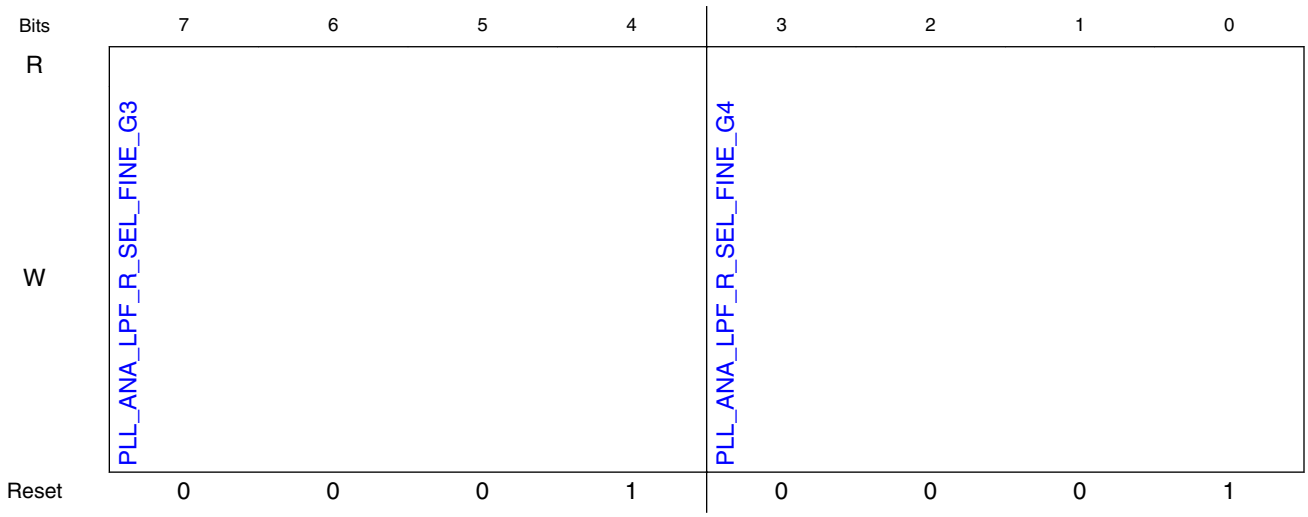
Field	Function
7-4 PLL_ANA_LPF_R_SEL_FINE_G1	[GEN1] [Fine Mode] PLL loop filter resistor control 0000: Min, … 1111: Max
3-0 PLL_ANA_LPF_R_SEL_FINE_G2	[GEN2] [Fine Mode]

11.5.3.1.35 (CMN_REG021)

11.5.3.1.35.1 Offset

Register	Offset
CMN_REG021	84h

11.5.3.1.35.2 Diagram



11.5.3.1.35.3 Fields

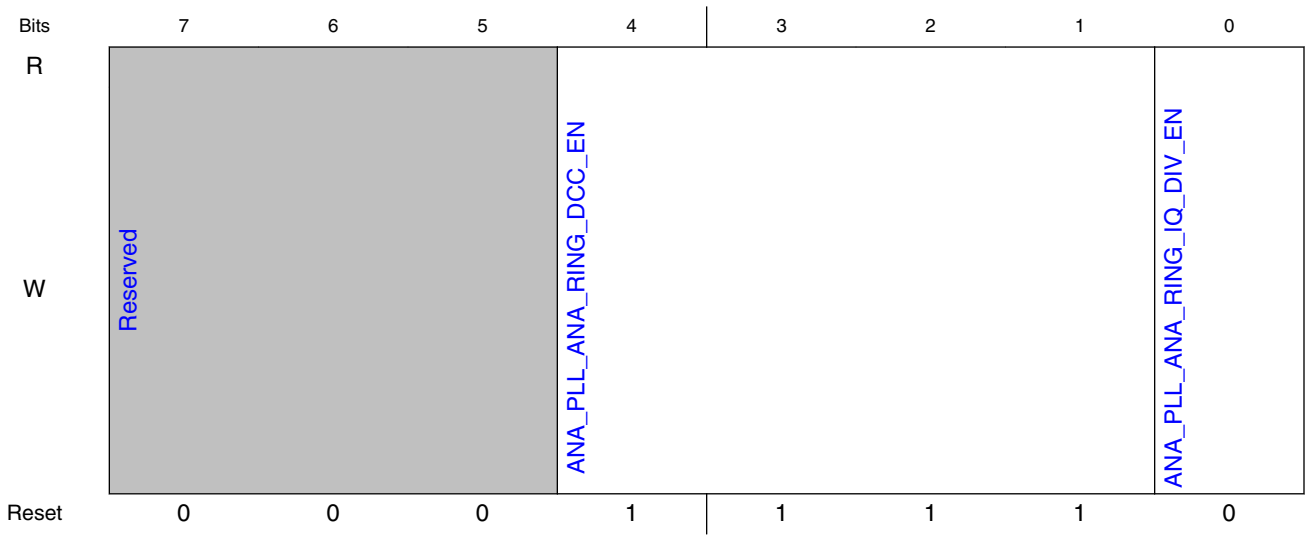
Field	Function
7-4 PLL_ANA_LPF_R_SEL_FINE_G3	[GEN3] [Fine Mode]
3-0 PLL_ANA_LPF_R_SEL_FINE_G4	[GEN4] [Fine Mode]

11.5.3.1.36 (CMN_REG022)

11.5.3.1.36.1 Offset

Register	Offset
CMN_REG022	88h

11.5.3.1.36.2 Diagram



11.5.3.1.36.3 Fields

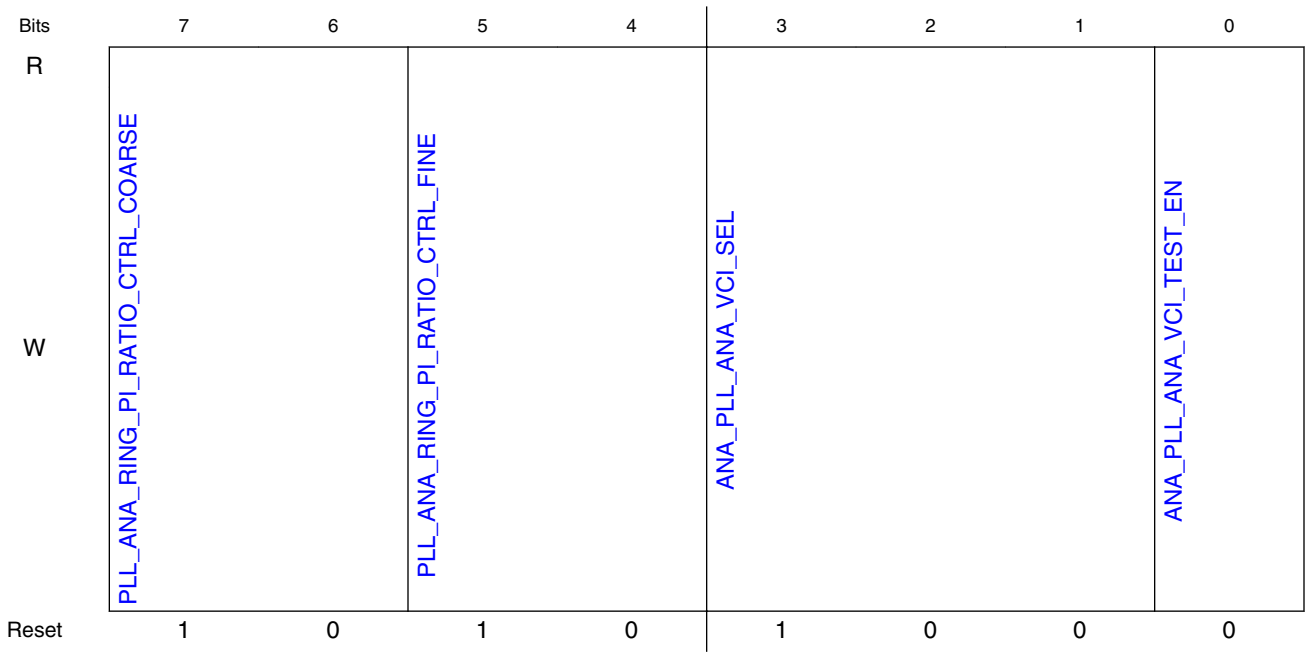
Field	Function
7-5 —	Reserved
4-1 ANA_PLL_ANA_RING_DCC_EN	PLL ring VCO DCC enable for each phase 0: Disable, 1: Enable
0 ANA_PLL_ANA_RING_IQ_DIV_EN	I/Q divider enable for PLL ring VCO clock 0: Disable, 1: Enable

11.5.3.1.37 (CMN_REG023)

11.5.3.1.37.1 Offset

Register	Offset
CMN_REG023	8Ch

11.5.3.1.37.2 Diagram



11.5.3.1.37.3 Fields

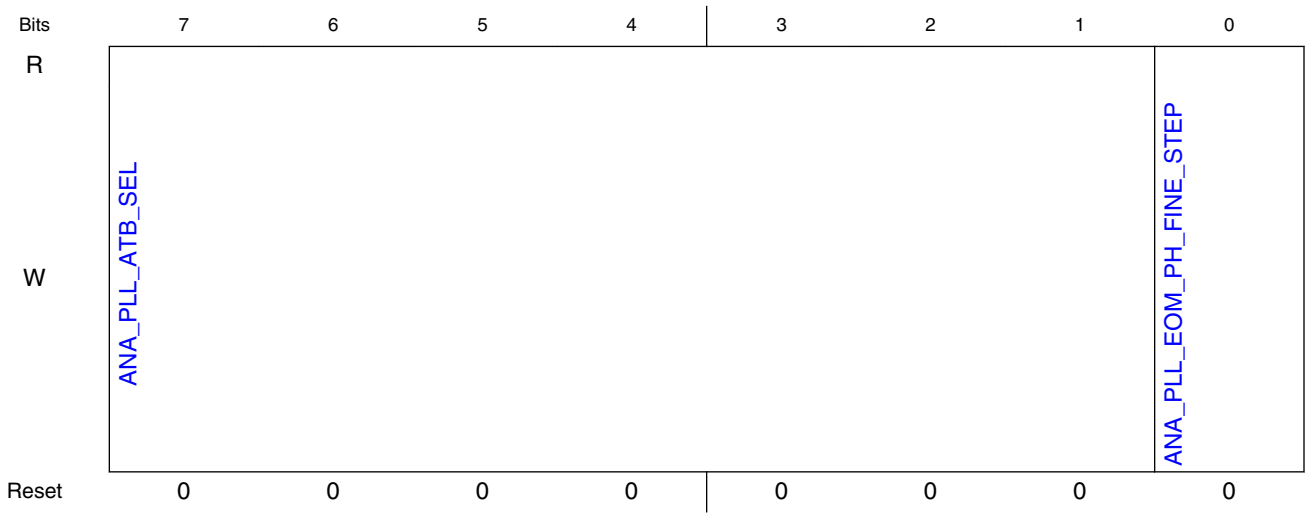
Field	Function
7-6 PLL_ANA_RING_PI_RATIO_CTRL_COARSE	Ratio between proportional and integral path gain 00: N/A, 01: 1/20, 10: 1/33, 11: N/A
5-4 PLL_ANA_RING_PI_RATIO_CTRL_FINE	Ratio between proportional and integral path gain 00: N/A, 01: 1/20, 10: 1/33, 11: N/A
3-1 ANA_PLL_ANA_VCI_SEL	PLL control voltage selection in AFC or open-loop test 000: 1.69V, 001: 1.59V, 010: 1.48V, 011: 1.38V, 100: 1.27V, 101: 1.16V, 110: 1.06V, 111: 0.95V
0 ANA_PLL_ANA_VCI_TEST_EN	PLL VCO test mode enable 0: Disable, 1: Enable

11.5.3.1.38 (CMN_REG024)

11.5.3.1.38.1 Offset

Register	Offset
CMN_REG024	90h

11.5.3.1.38.2 Diagram



11.5.3.1.38.3 Fields

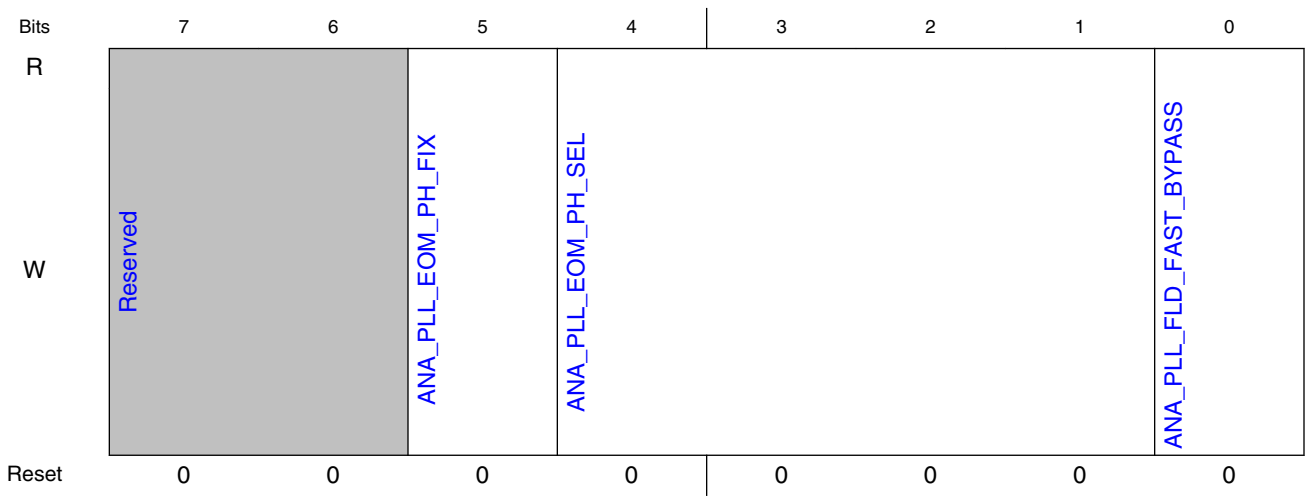
Field	Function
7-1 ANA_PLL_ATB_SEL	Select PLL ATB 0000000: No ATB (OPEN) 0000001: VCO_VCI 0000010: LF_VCI_BUF 0000100: GM_VCI 0001000: CAP_OFFSET_VCI 0010000: LC_MON_VDD_LDO 0100000: VDDH 1000000: RING_MON_VCM
0 ANA_PLL_EOM_PH_FINE_STEP	EOM phase resolution enhancement 0: Disable, 1: Double resolution

11.5.3.1.39 (CMN_REG025)

11.5.3.1.39.1 Offset

Register	Offset
CMN_REG025	94h

11.5.3.1.39.2 Diagram



11.5.3.1.39.3 Fields

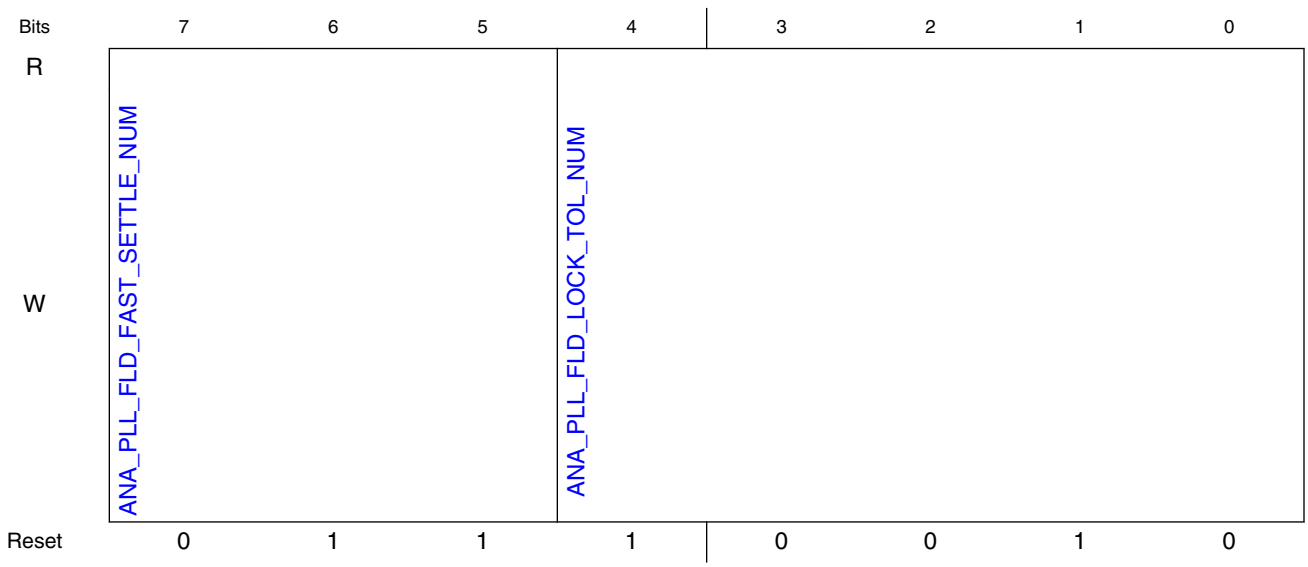
Field	Function
7-6 —	Reserved
5 ANA_PLL_EOM_PH_FIX	Phase shift enable for EOM 0: Disable, 1: Enable
4-1 ANA_PLL_EOM_PH_SEL	EOM phase selection 0000: 0°, 0001: 22.5°, 0010: 45°, 0011: 67.5° … , 1110: 315°, 1111: 337.5°
0 ANA_PLL_FLD_FAST_BYPASS	PLL fast frequency lock detection bypass 0: no bypass, 1: bypass

11.5.3.1.40 (CMN_REG026)

11.5.3.1.40.1 Offset

Register	Offset
CMN_REG026	98h

11.5.3.1.40.2 Diagram



11.5.3.1.40.3 Fields

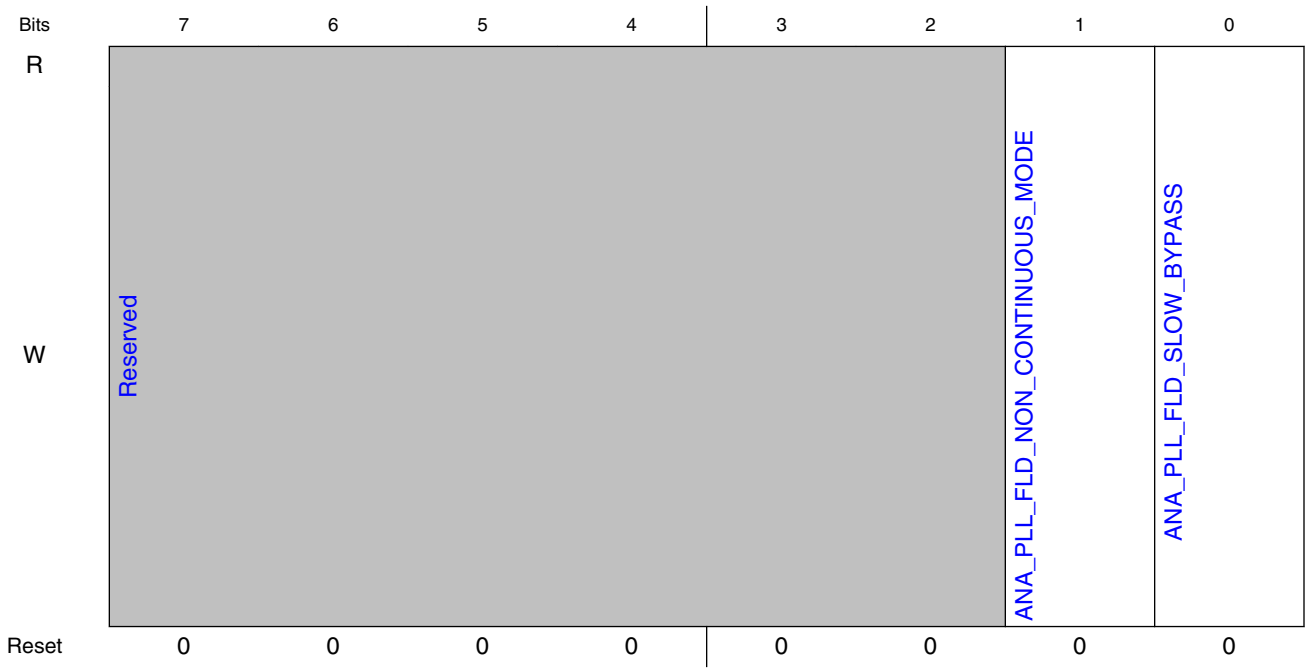
Field	Function
7-5 ANA_PLL_FLD_FAST_SETTLE_NUM	Number of reference clock cycle to check VCO stabilization in fast FLD
4-0 ANA_PLL_FLD_LOCK_TOL_NUM	FLD lock tolerance setting Lock detector activates lock signal when VCO counter value is smaller than i_pll_fld_ppm_set/(Min(1024<(i_pll_pms_m* 2^n)<2048)) * 1000000 where n=1~10, in fast FLD, and i_pll_fld_ppm_set/1024) * 1000000, In slow FLD, respectively.

11.5.3.1.41 (CMN_REG027)

11.5.3.1.41.1 Offset

Register	Offset
CMN_REG027	9Ch

11.5.3.1.41.2 Diagram



11.5.3.1.41.3 Fields

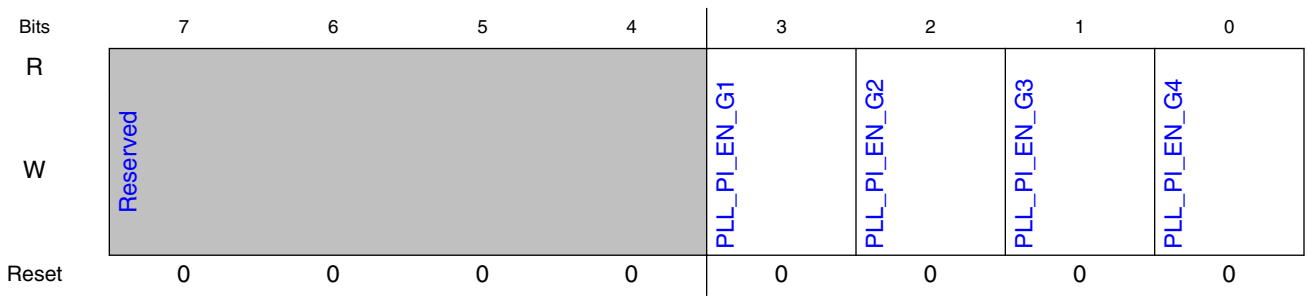
Field	Function
7-2 —	Reserved
1 ANA_PLL_FLD_NON_CONTINUOUS_MODE	Check frequency lock detection of PLL 0: once, 1: continuously
0 ANA_PLL_FLD_SLOW_BYPASS	PLL slow frequency lock detection bypass 0: no bypass, 1: bypass

11.5.3.1.42 (CMN_REG028)

11.5.3.1.42.1 Offset

Register	Offset
CMN_REG028	A0h

11.5.3.1.42.2 Diagram



11.5.3.1.42.3 Fields

Field	Function
7-4 —	Reserved
3 PLL_PI_EN_G1	[GEN1] PLL phase interpolator enable 0: Disable, 1: Enable
2 PLL_PI_EN_G2	[GEN2]
1 PLL_PI_EN_G3	[GEN3]
0 PLL_PI_EN_G4	[GEN4]

11.5.3.1.43 (CMN_REG029)

11.5.3.1.43.1 Offset

Register	Offset
CMN_REG029	A4h

11.5.3.1.43.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	PLL_PI_STR_G1				PLL_PI_STR_G2			
W								
Reset	1	1	1	1	1	1	1	1

11.5.3.1.43.3 Fields

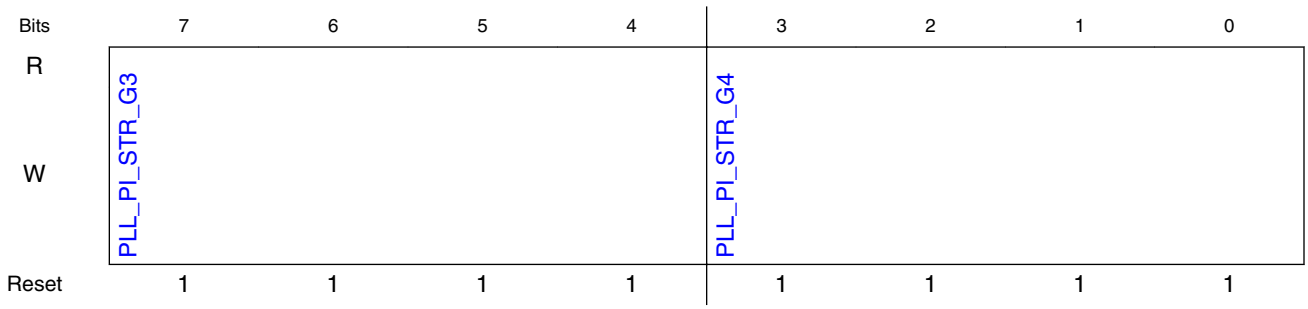
Field	Function
7-4 PLL_PI_STR_G1	[GEN1] PLL phase interpolator input buffer strength control for Gen3 and Gen4 (for 8GHz VCO) 0000: Min. strength …. 1111: Max. strength
3-0 PLL_PI_STR_G2	[GEN2]

11.5.3.1.44 (CMN_REG02A)

11.5.3.1.44.1 Offset

Register	Offset
CMN_REG02A	A8h

11.5.3.1.44.2 Diagram



11.5.3.1.44.3 Fields

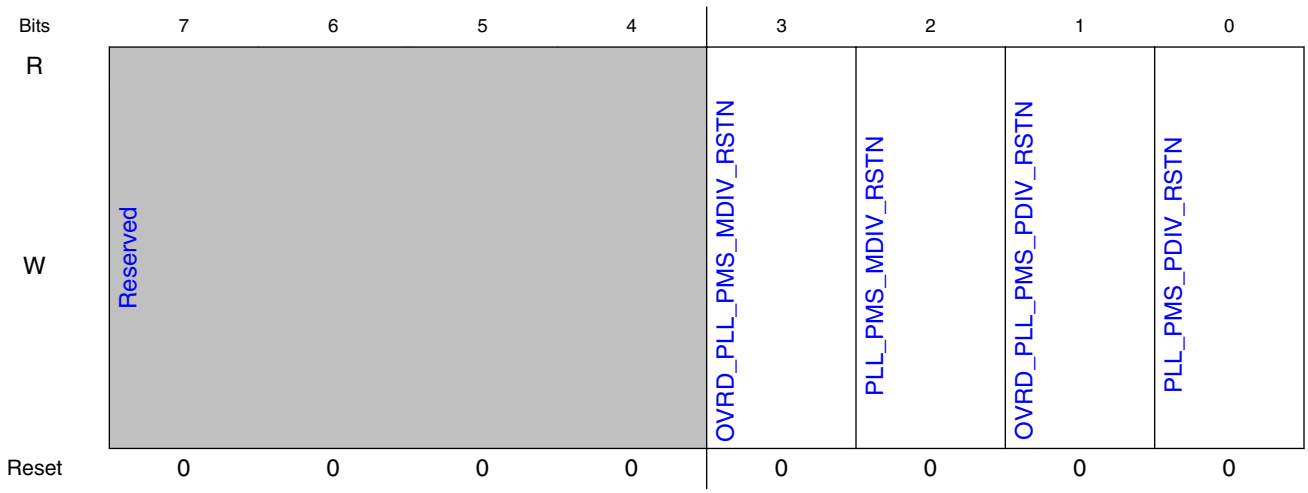
Field	Function
7-4 PLL_PI_STR_G3	[GEN3]
3-0 PLL_PI_STR_G4	[GEN4]

11.5.3.1.45 (CMN_REG02B)

11.5.3.1.45.1 Offset

Register	Offset
CMN_REG02B	ACh

11.5.3.1.45.2 Diagram



11.5.3.1.45.3 Fields

Field	Function
7-4 —	Reserved
3 OVRD_PLL_PMS_MDIV_RSTN	Override enable for pll_pms_mdiv_rstn
2 PLL_PMS_MDIV_RSTN	PLL main divider reset 0: Reset, 1: Released
1 OVRD_PLL_PMS_PDIV_RSTN	Override enable for pll_pms_pdiv_rstn
0 PLL_PMS_PDIV_RSTN	PLL pre-divider reset 0: Reset, 1: Released

11.5.3.1.46 (CMN_REG02C)

11.5.3.1.46.1 Offset

Register	Offset
CMN_REG02C	B0h

11.5.3.1.46.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_PMS_MDIV_AFC_G1							
Reset	0	0	1	1	0	0	1	0

11.5.3.1.46.3 Fields

Field	Function
7-0 PLL_PMS_MDIV_AFC_G1	[GEN1] PLL AFC target value (fVCO/fREF) setting (Same as i_pll_pms_mdiv)

11.5.3.1.47 (CMN_REG02D)

11.5.3.1.47.1 Offset

Register	Offset
CMN_REG02D	B4h

11.5.3.1.47.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_PMS_MDIV_AFC_G2							
Reset	0	0	1	1	0	0	1	0

11.5.3.1.47.3 Fields

Field	Function
7-0 PLL_PMS_MDIV_AFC_G2	[GEN2]

11.5.3.1.48 (CMN_REG02E)

11.5.3.1.48.1 Offset

Register	Offset
CMN_REG02E	B8h

11.5.3.1.48.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_PMS_MDIV_AFC_G3							
Reset	0	0	1	0	1	0	0	0

11.5.3.1.48.3 Fields

Field	Function
7-0 PLL_PMS_MDIV_AFC_G3	[GEN3]

11.5.3.1.49 (CMN_REG02F)

11.5.3.1.49.1 Offset

Register	Offset
CMN_REG02F	BCh

11.5.3.1.49.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_PMS_MDIV_AFC_G4							
Reset	0	0	1	0	1	0	0	0

11.5.3.1.49.3 Fields

Field	Function
7-0 PLL_PMS_MDIV_AFC_G4	[GEN4]

11.5.3.1.50 (CMN_REG030)

11.5.3.1.50.1 Offset

Register	Offset
CMN_REG030	C0h

11.5.3.1.50.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_PMS_MDIV_G1							
Reset	0	0	1	1	0	0	1	0

11.5.3.1.50.3 Fields

Field	Function
7-0 PLL_PMS_MDIV_G1	[GEN1] PLL main divider setting Divider value = Setting code

11.5.3.1.51 (CMN_REG031)

11.5.3.1.51.1 Offset

Register	Offset
CMN_REG031	C4h

11.5.3.1.51.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_PMS_MDIV_G2							
Reset	0	0	1	1	0	0	1	0

11.5.3.1.51.3 Fields

Field	Function
7-0 PLL_PMS_MDIV_G2	[GEN2]

11.5.3.1.52 (CMN_REG032)

11.5.3.1.52.1 Offset

Register	Offset
CMN_REG032	C8h

11.5.3.1.52.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_PMS_MDIV_G3							
Reset	0	0	1	0	1	0	0	0

11.5.3.1.52.3 Fields

Field	Function
7-0 PLL_PMS_MDIV_G3	[GEN3]

11.5.3.1.53 (CMN_REG033)

11.5.3.1.53.1 Offset

Register	Offset
CMN_REG033	CCh

11.5.3.1.53.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_PMS_MDIV_G4							
Reset	0	0	1	0	1	0	0	0

11.5.3.1.53.3 Fields

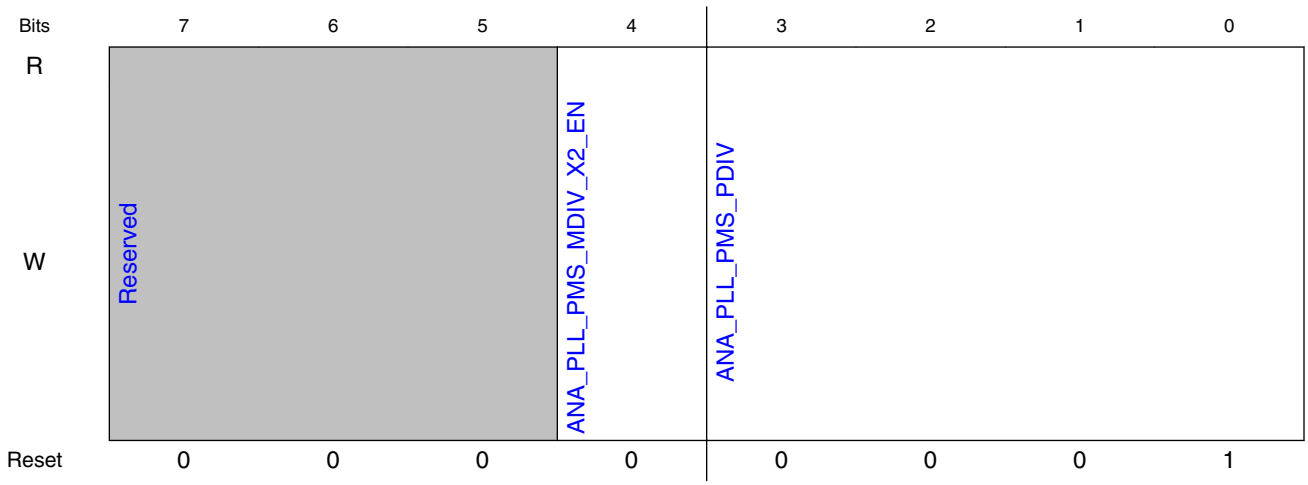
Field	Function
7-0 PLL_PMS_MDIV_G4	[GEN4]

11.5.3.1.54 (CMN_REG034)

11.5.3.1.54.1 Offset

Register	Offset
CMN_REG034	D0h

11.5.3.1.54.2 Diagram



11.5.3.1.54.3 Fields

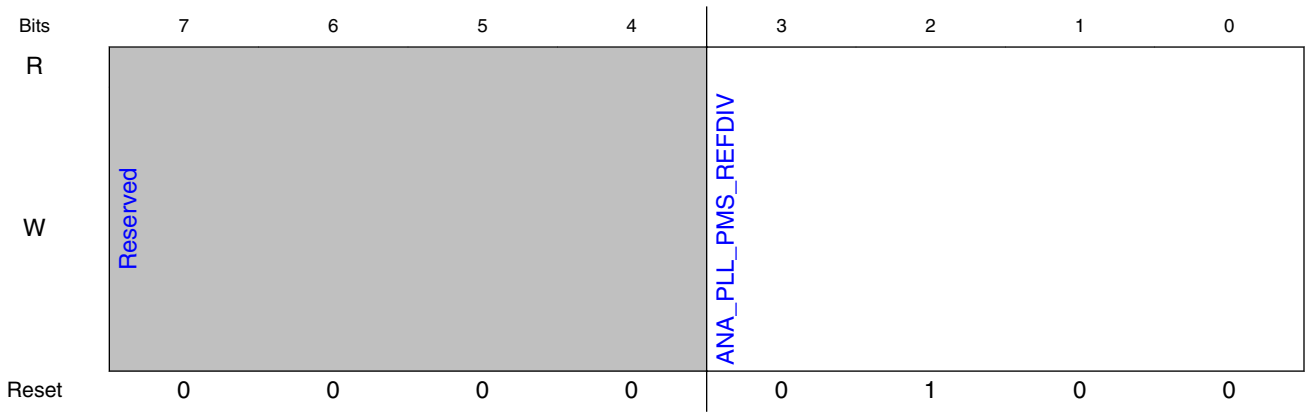
Field	Function
7-5 —	Reserved
4 ANA_PLL_PMS_MDIV_X2_EN	PLL main divider extra X2 enable 0: /1, 1: /2
3-0 ANA_PLL_PMS_PDIV	PLL pre-divider setting 0000: N/A, 0001: /1, 0010: /2, 0011: /3, … 1111: /15

11.5.3.1.55 (CMN_REG035)

11.5.3.1.55.1 Offset

Register	Offset
CMN_REG035	D4h

11.5.3.1.55.2 Diagram



11.5.3.1.55.3 Fields

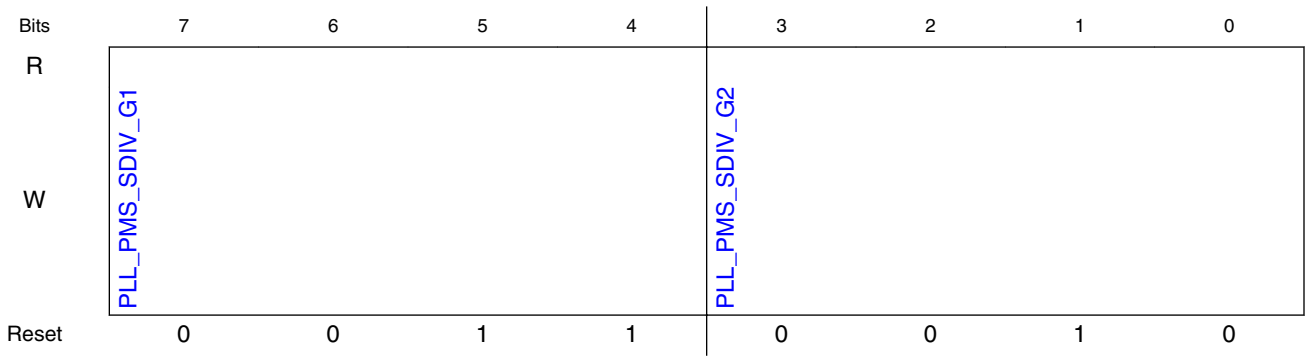
Field	Function
7-4 —	Reserved
3-0 ANA_PLL_PMS_REFDIV	PLL reference clock divider setting 0000: N/A, 0001: /1, 0010: /2, 0011: /3, … 1111: /15

11.5.3.1.56 (CMN_REG036)

11.5.3.1.56.1 Offset

Register	Offset
CMN_REG036	D8h

11.5.3.1.56.2 Diagram



11.5.3.1.56.3 Fields

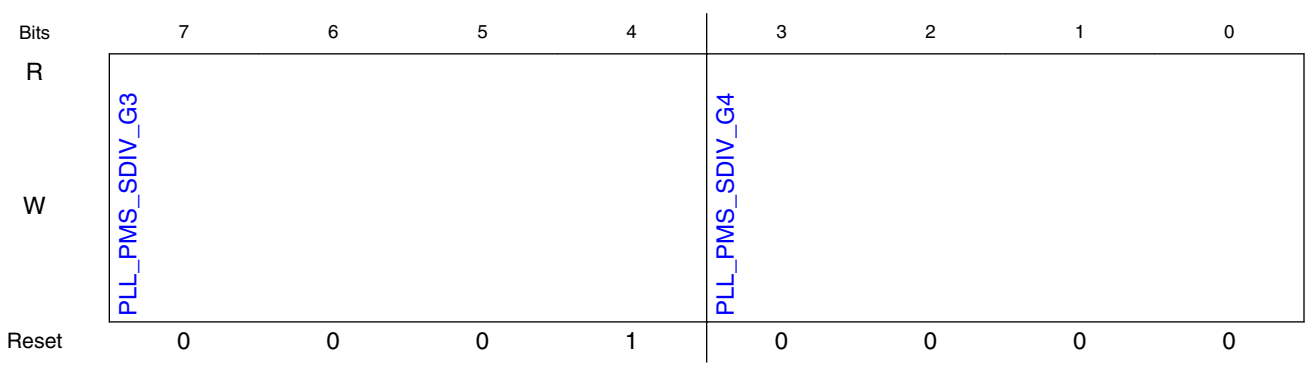
Field	Function
7-4 PLL_PMS_SDIV_G1	[GEN1] PLL post divider setting 0000: /1, 0001: /2, 0010: /4, 0011: /8
3-0 PLL_PMS_SDIV_G2	[GEN2]

11.5.3.1.57 (CMN_REG037)

11.5.3.1.57.1 Offset

Register	Offset
CMN_REG037	DCh

11.5.3.1.57.2 Diagram



11.5.3.1.57.3 Fields

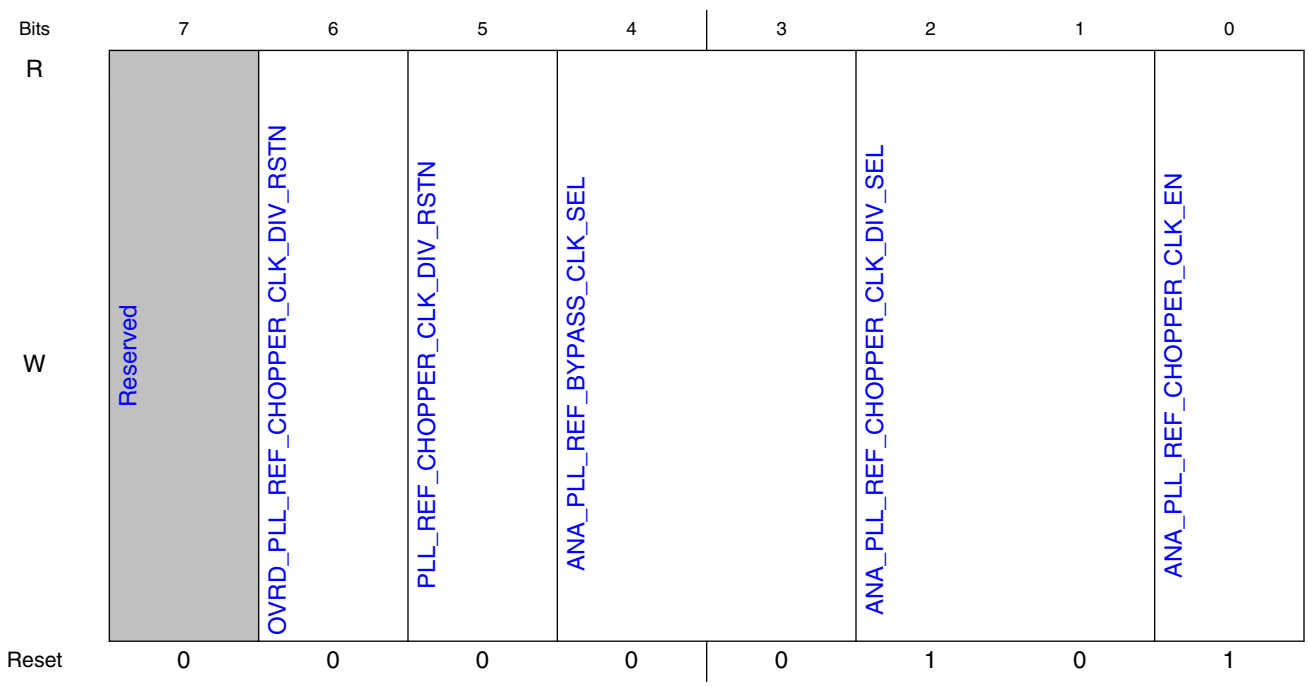
Field	Function
7-4 PLL_PMS_SDIV_G3	[GEN3]
3-0 PLL_PMS_SDIV_G4	[GEN4]

11.5.3.1.58 (CMN_REG038)

11.5.3.1.58.1 Offset

Register	Offset
CMN_REG038	E0h

11.5.3.1.58.2 Diagram



11.5.3.1.58.3 Fields

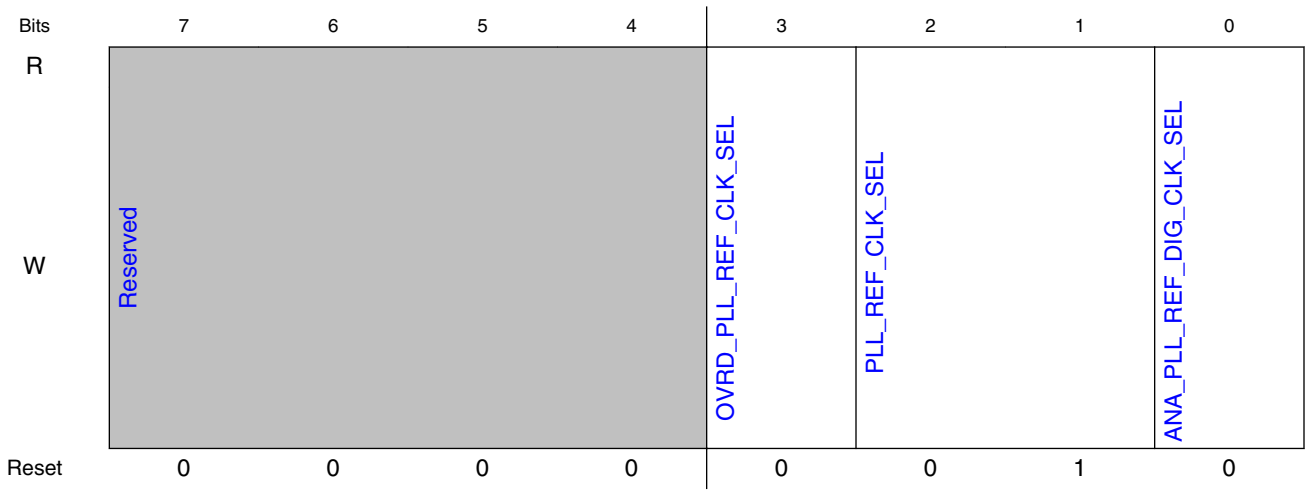
Field	Function
7 —	Reserved
6 OVRD_PLL_REF_CHOPPER_CLK_DIV_RSTN	Override enable for pll_ref_chopper_clk_div_rstn
5 PLL_REF_CHOPPER_CLK_DIV_RSTN	Chopper clk divider reset 0: Reset, 1: Released
4-3 ANA_PLL_REF_BYPASS_CLK_SEL	PLL Bypass clock selection 00: VSS 01: XO 10: External reference clock 11: System PLL clock
2-1 ANA_PLL_REF_CHOPPER_CLK_DIV_SEL	Chopper clk divider value 00: /20 (5MHz) , 01: /40 (2.5MHz), 10: /50 (2MHz), 11: /100 (1MHz)
0 ANA_PLL_REF_CHOPPER_CLK_EN	Chopper clk enable 0: Disable, 1: Enable

11.5.3.1.59 (CMN_REG039)

11.5.3.1.59.1 Offset

Register	Offset
CMN_REG039	E4h

11.5.3.1.59.2 Diagram



11.5.3.1.59.3 Fields

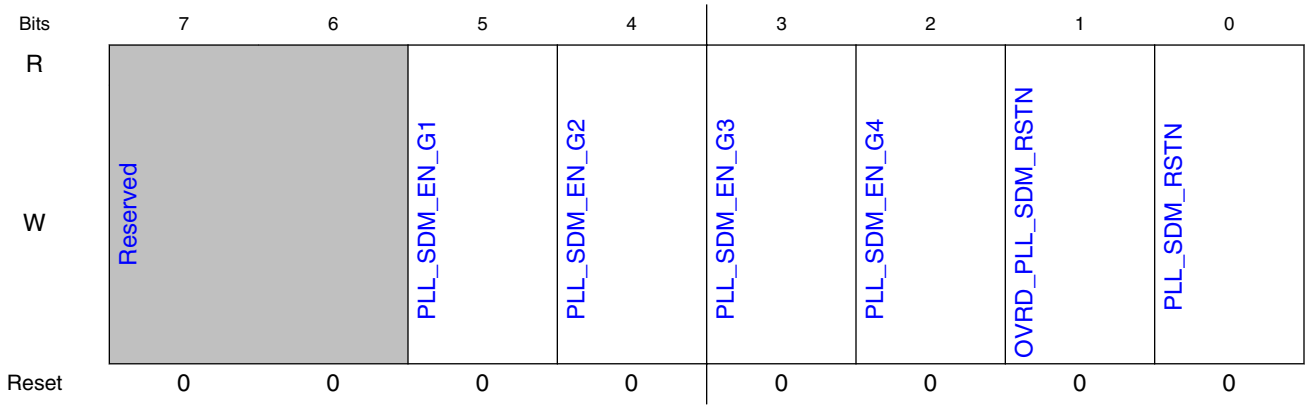
Field	Function
7-4 —	Reserved
3 OVRD_PLL_REF_CLK_SEL	Override enable for pll_ref_clk_sel
2-1 PLL_REF_CLK_SEL	PLL reference clock selection 00: VSS 01: XO 10: External reference clock 11: System PLL clock
0 ANA_PLL_REF_DIG_CLK_SEL	

11.5.3.1.60 (CMN_REG03A)

11.5.3.1.60.1 Offset

Register	Offset
CMN_REG03A	E8h

11.5.3.1.60.2 Diagram



11.5.3.1.60.3 Fields

Field	Function
7-6 —	Reserved
5 PLL_SDM_EN_G1	[GEN1] PLL SDM enable 0: Disable, 1: Enable
4 PLL_SDM_EN_G2	[GEN2]
3 PLL_SDM_EN_G3	[GEN3]
2 PLL_SDM_EN_G4	[GEN4]
1 OVRD_PLL_SDM_RSTN	Override enable for pll_sdm_rstn
0 PLL_SDM_RSTN	PLL SDM reset 0: reset, 1: released

11.5.3.1.61 (CMN_REG03B)

11.5.3.1.61.1 Offset

Register	Offset
CMN_REG03B	ECh

11.5.3.1.61.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Reserved		PLL_SDC_FRACTIONAL_EN_G1	PLL_SDC_FRACTIONAL_EN_G2	PLL_SDC_FRACTIONAL_EN_G3	PLL_SDC_FRACTIONAL_EN_G4	OVRD_PLL_SDC_RSTN	PLL_SDC_RSTN
Reset	0	0	0	0	0	0	0	0

11.5.3.1.61.3 Fields

Field	Function
7-6 —	Reserved
5 PLL_SDC_FRACTIONAL_EN_G1	[GEN1] Fractional clock divide in SDM clock generation 0: Integer division, 1: Fractional division
4 PLL_SDC_FRACTIONAL_EN_G2	[GEN2]
3 PLL_SDC_FRACTIONAL_EN_G3	[GEN3]
2 PLL_SDC_FRACTIONAL_EN_G4	[GEN4]

Table continues on the next page...

Field	Function
1 OVRD_PLL_SDC_RSTN	Override enable for pll_sdc_rstn
0 PLL_SDC_RSTN	PLL SDM clock generation (SDC) reset 0: Reset, 1: Release

11.5.3.1.62 (CMN_REG03C)

11.5.3.1.62.1 Offset

Register	Offset
CMN_REG03C	F0h

11.5.3.1.62.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_SDM_DENOMINATOR_G1							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.62.3 Fields

Field	Function
7-0 PLL_SDM_DENOMINATOR_G1	[GEN1] Denominator of SDM (Max. 255) Constraint: i_pll_sdm_lc > |signed i_pll_sdm_k-i_pll_ssc_fm_deviation-i_pll_ssc_fm_freq-1|

11.5.3.1.63 (CMN_REG03D)

11.5.3.1.63.1 Offset

Register	Offset
CMN_REG03D	F4h

11.5.3.1.63.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_SDM_DENOMINATOR_G2							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.63.3 Fields

Field	Function
7-0 PLL_SDM_DENOMINATOR_G2	[GEN2]

11.5.3.1.64 (CMN_REG03E)

11.5.3.1.64.1 Offset

Register	Offset
CMN_REG03E	F8h

11.5.3.1.64.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_SDM_DENOMINATOR_G3							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.64.3 Fields

Field	Function
7-0 PLL_SDM_DENOMINATOR_G3	[GEN3]

11.5.3.1.65 (CMN_REG03F)

11.5.3.1.65.1 Offset

Register	Offset
CMN_REG03F	FCh

11.5.3.1.65.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_SDM_DENOMINATOR_G4							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.65.3 Fields

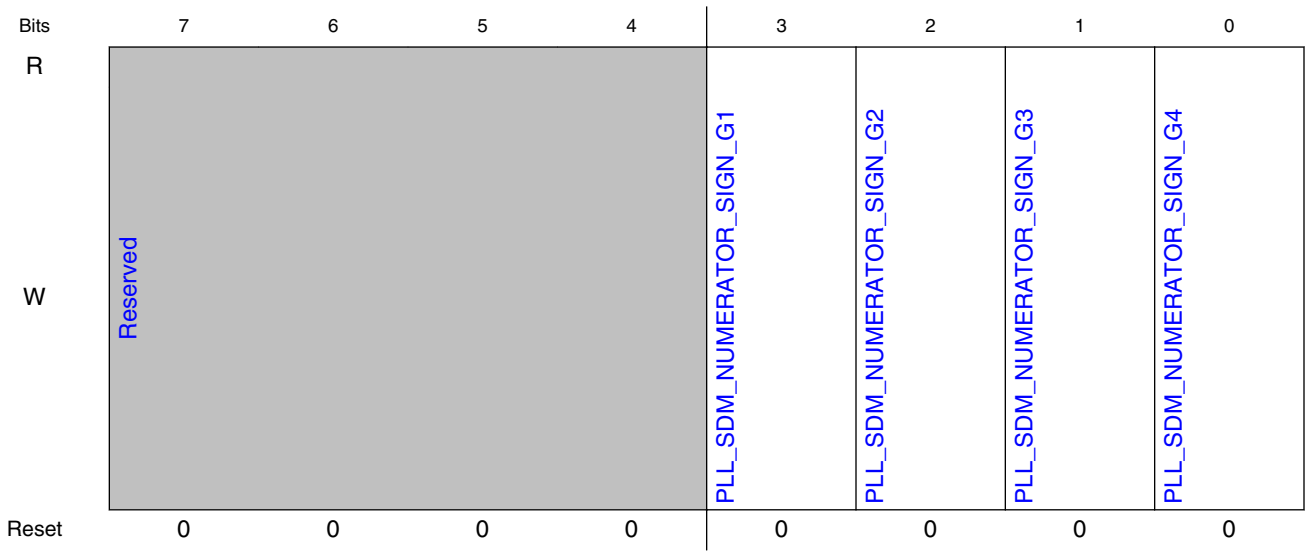
Field	Function
7-0 PLL_SDM_DENOMINATOR_G4	[GEN4]

11.5.3.1.66 (CMN_REG040)

11.5.3.1.66.1 Offset

Register	Offset
CMN_REG040	100h

11.5.3.1.66.2 Diagram



11.5.3.1.66.3 Fields

Field	Function
7-4 —	Reserved
3 PLL_SDM_NUMERATOR_SIGN_G1	[GEN1] Sign of SDM numerator 0: positive, 1: negative
2 PLL_SDM_NUMERATOR_SIGN_G2	[GEN2]
1 PLL_SDM_NUMERATOR_SIGN_G3	[GEN3]
0 PLL_SDM_NUMERATOR_SIGN_G4	[GEN4]

11.5.3.1.67 (CMN_REG041)

11.5.3.1.67.1 Offset

Register	Offset
CMN_REG041	104h

11.5.3.1.67.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_SDM_NUMERATOR_G1							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.67.3 Fields

Field	Function
7-0 PLL_SDM_NUMERATOR_G1	[GEN1] Numerator of SDM with i_pll_sdm_k_sign (-255~255)

11.5.3.1.68 (CMN_REG042)

11.5.3.1.68.1 Offset

Register	Offset
CMN_REG042	108h

11.5.3.1.68.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_SDM_NUMERATOR_G2							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.68.3 Fields

Field	Function
7-0 PLL_SDM_NUMERATOR_G2	[GEN2]

11.5.3.1.69 (CMN_REG043)

11.5.3.1.69.1 Offset

Register	Offset
CMN_REG043	10Ch

11.5.3.1.69.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_SDM_NUMERATOR_G3							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.69.3 Fields

Field	Function
7-0 PLL_SDM_NUMERATOR_G3	[GEN3]

11.5.3.1.70 (CMN_REG044)

11.5.3.1.70.1 Offset

Register	Offset
CMN_REG044	110h

11.5.3.1.70.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PLL_SDM_NUMERATOR_G4							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.70.3 Fields

Field	Function
7-0 PLL_SDM_NUMERATOR_G4	[GEN4]

11.5.3.1.71 (CMN_REG045)

11.5.3.1.71.1 Offset

Register	Offset
CMN_REG045	114h

11.5.3.1.71.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Reserved				PLL_SDM_PH_NUM_SEL_G1	PLL_SDM_PH_NUM_SEL_G2	PLL_SDM_PH_NUM_SEL_G3	PLL_SDM_PH_NUM_SEL_G4
Reset	0	0	0	0	0	0	0	0

11.5.3.1.71.3 Fields

Field	Function
7-4 —	Reserved
3 PLL_SDM_PH_NUM_SEL_G1	[GEN1] PLL PI input clock phase number 0: 8-phase, 1: 4-phase
2 PLL_SDM_PH_NUM_SEL_G2	[GEN2]
1 PLL_SDM_PH_NUM_SEL_G3	[GEN3]
0 PLL_SDM_PH_NUM_SEL_G4	[GEN4]

11.5.3.1.72 (CMN_REG046)

11.5.3.1.72.1 Offset

Register	Offset
CMN_REG046	118h

11.5.3.1.72.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	PLL_SDM_PI_STEP_G1		PLL_SDM_PI_STEP_G2		PLL_SDM_PI_STEP_G3		PLL_SDM_PI_STEP_G4	
W								
Reset	0	0	0	0	0	0	0	0

11.5.3.1.72.3 Fields

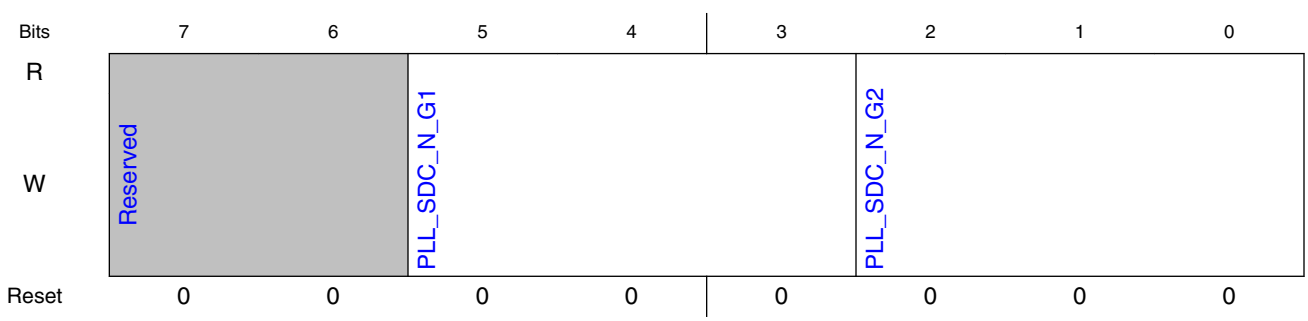
Field	Function
7-6 PLL_SDM_PI_STEP_G1	[GEN1] PLL phase interpolstor step 00: 1-step (16-phase), 01: 2-step (8-phase), 1x: 0.5-step
5-4 PLL_SDM_PI_STEP_G2	[GEN2]
3-2 PLL_SDM_PI_STEP_G3	[GEN3]
1-0 PLL_SDM_PI_STEP_G4	[GEN4]

11.5.3.1.73 (CMN_REG047)

11.5.3.1.73.1 Offset

Register	Offset
CMN_REG047	11Ch

11.5.3.1.73.2 Diagram



11.5.3.1.73.3 Fields

Field	Function
7-6 —	Reserved

Table continues on the next page...

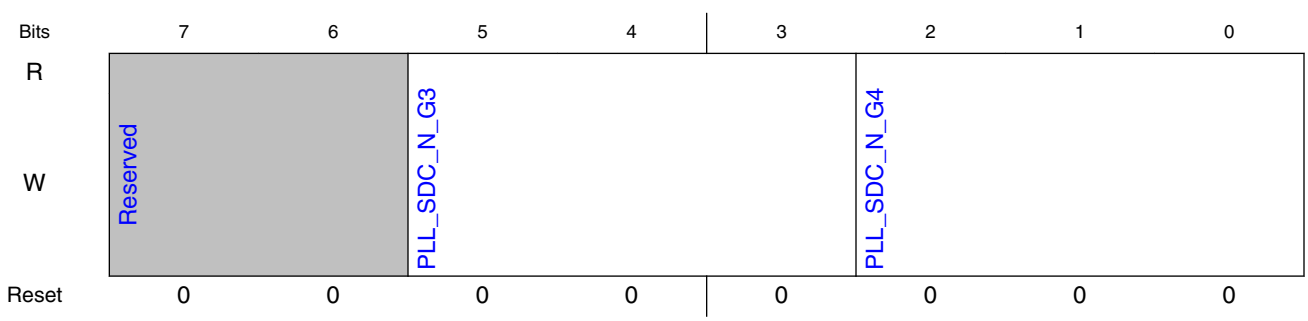
Field	Function
5-3 PLL_SDC_N_G1	[GEN1] PLL SDC divide-ratio selection 000: /3, 001: /4, 010: /5, 011: /6, 100: /7, 101: /8
2-0 PLL_SDC_N_G2	[GEN2]

11.5.3.1.74 (CMN_REG048)

11.5.3.1.74.1 Offset

Register	Offset
CMN_REG048	120h

11.5.3.1.74.2 Diagram



11.5.3.1.74.3 Fields

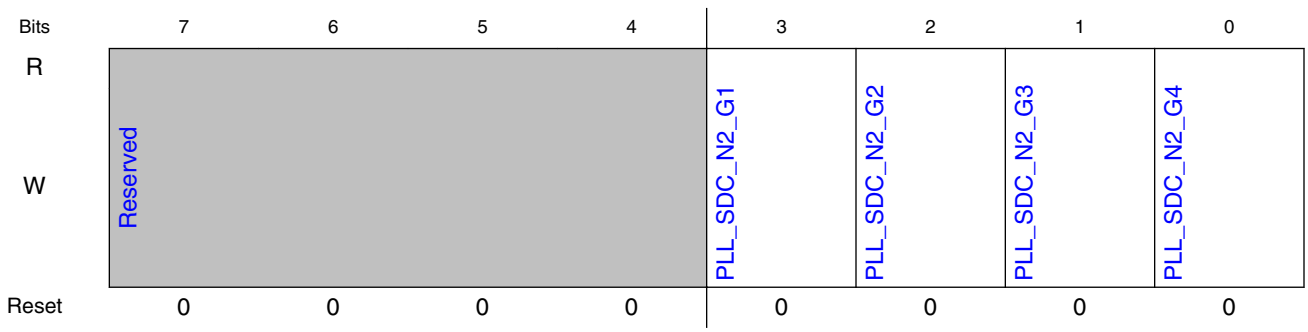
Field	Function
7-6 —	Reserved
5-3 PLL_SDC_N_G3	[GEN3]
2-0 PLL_SDC_N_G4	[GEN4]

11.5.3.1.75 (CMN_REG049)

11.5.3.1.75.1 Offset

Register	Offset
CMN_REG049	124h

11.5.3.1.75.2 Diagram



11.5.3.1.75.3 Fields

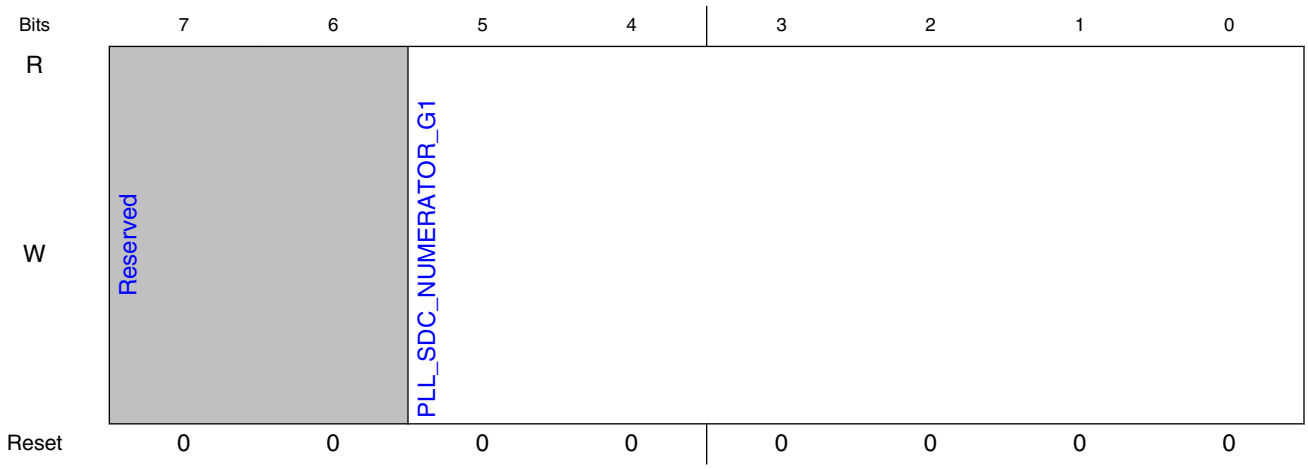
Field	Function
7-4 —	Reserved
3 PLL_SDC_N2_G1	[GEN1] PLL SDC divide-ratio selection 0: No additional /2, 1: Additional /2
2 PLL_SDC_N2_G2	[GEN2]
1 PLL_SDC_N2_G3	[GEN3]
0 PLL_SDC_N2_G4	[GEN4]

11.5.3.1.76 (CMN_REG04A)

11.5.3.1.76.1 Offset

Register	Offset
CMN_REG04A	128h

11.5.3.1.76.2 Diagram



11.5.3.1.76.3 Fields

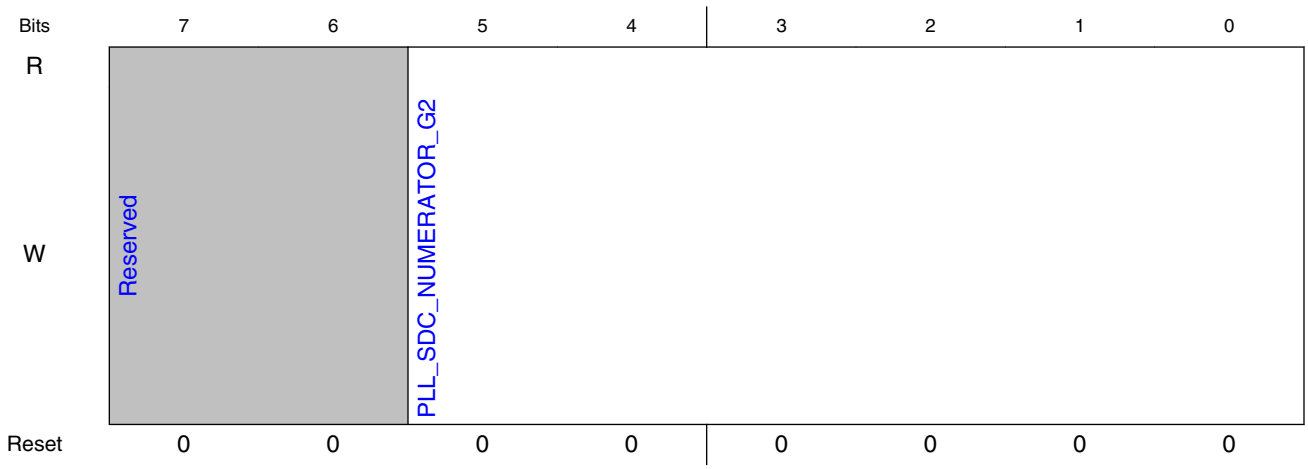
Field	Function
7-6 —	Reserved
5-0 PLL_SDC_NUMERATOR_G1	[GEN1] Numerator of SDC (Max 65)

11.5.3.1.77 (CMN_REG04B)

11.5.3.1.77.1 Offset

Register	Offset
CMN_REG04B	12Ch

11.5.3.1.77.2 Diagram



11.5.3.1.77.3 Fields

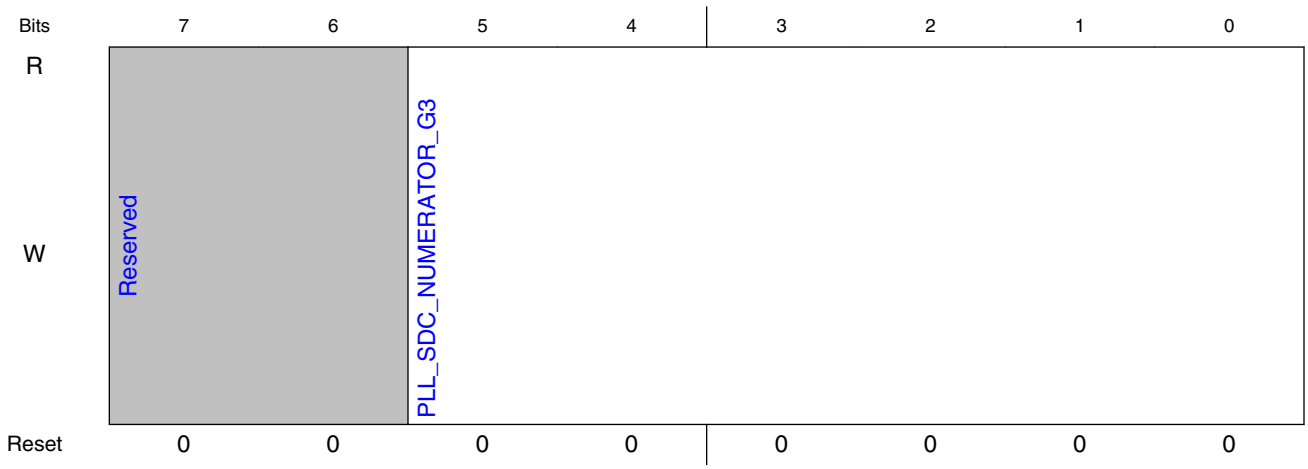
Field	Function
7-6 —	Reserved
5-0 PLL_SDC_NUMERATOR_G2	[GEN2]

11.5.3.1.78 (CMN_REG04C)

11.5.3.1.78.1 Offset

Register	Offset
CMN_REG04C	130h

11.5.3.1.78.2 Diagram



11.5.3.1.78.3 Fields

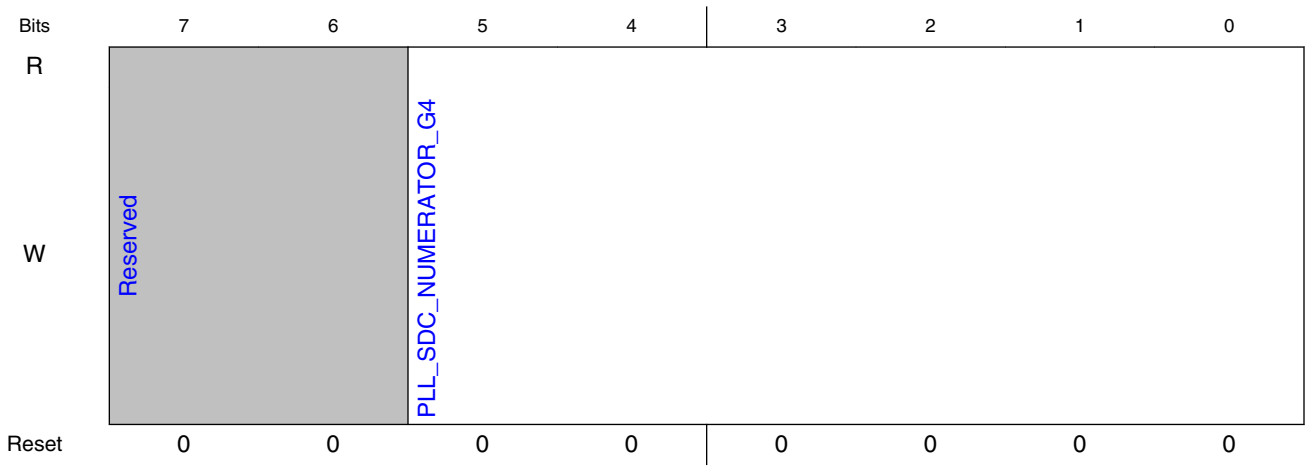
Field	Function
7-6 —	Reserved
5-0 PLL_SDC_NUMERATOR_G3	[GEN3]

11.5.3.1.79 (CMN_REG04D)

11.5.3.1.79.1 Offset

Register	Offset
CMN_REG04D	134h

11.5.3.1.79.2 Diagram



11.5.3.1.79.3 Fields

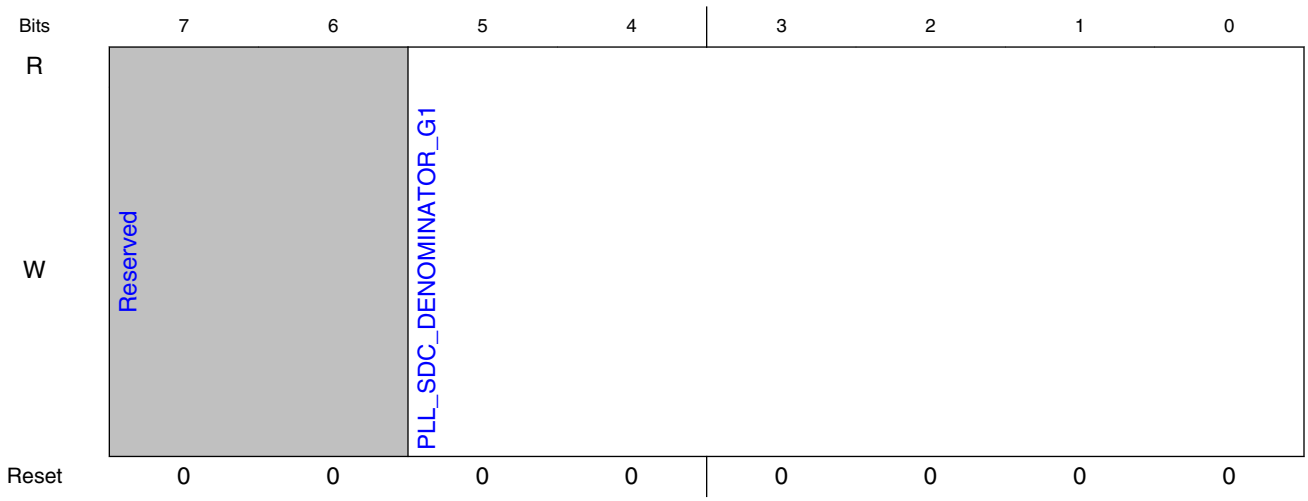
Field	Function
7-6 —	Reserved
5-0 PLL_SDC_NUMERATOR_G4	[GEN4]

11.5.3.1.80 (CMN_REG04E)

11.5.3.1.80.1 Offset

Register	Offset
CMN_REG04E	138h

11.5.3.1.80.2 Diagram



11.5.3.1.80.3 Fields

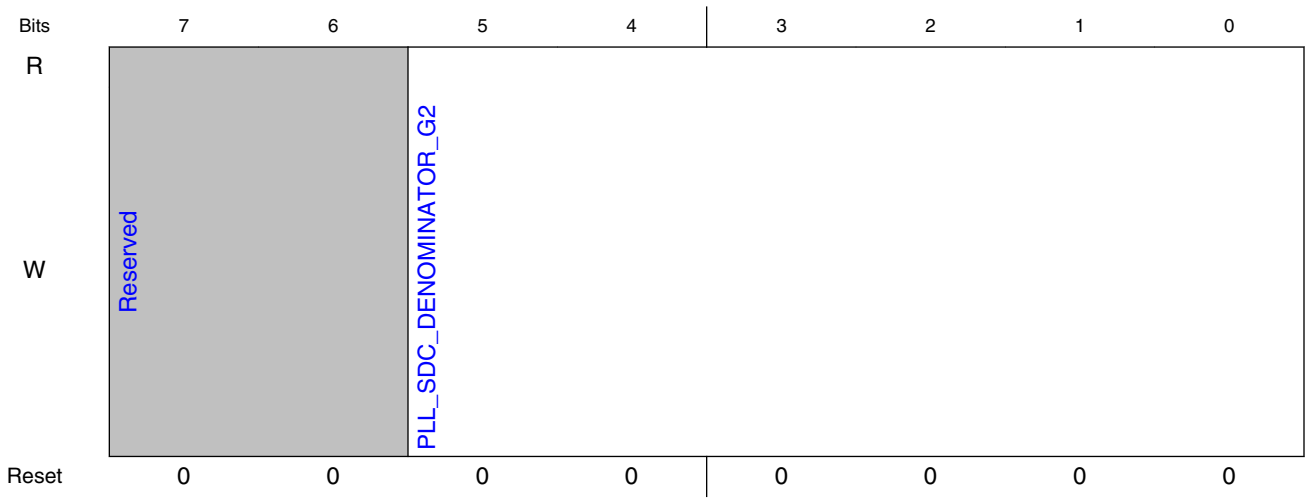
Field	Function
7-6 —	Reserved
5-0 PLL_SDC_DENOMINATOR_G1	[GEN1] Denominator of SDC (Max 65)

11.5.3.1.81 (CMN_REG04F)

11.5.3.1.81.1 Offset

Register	Offset
CMN_REG04F	13Ch

11.5.3.1.81.2 Diagram



11.5.3.1.81.3 Fields

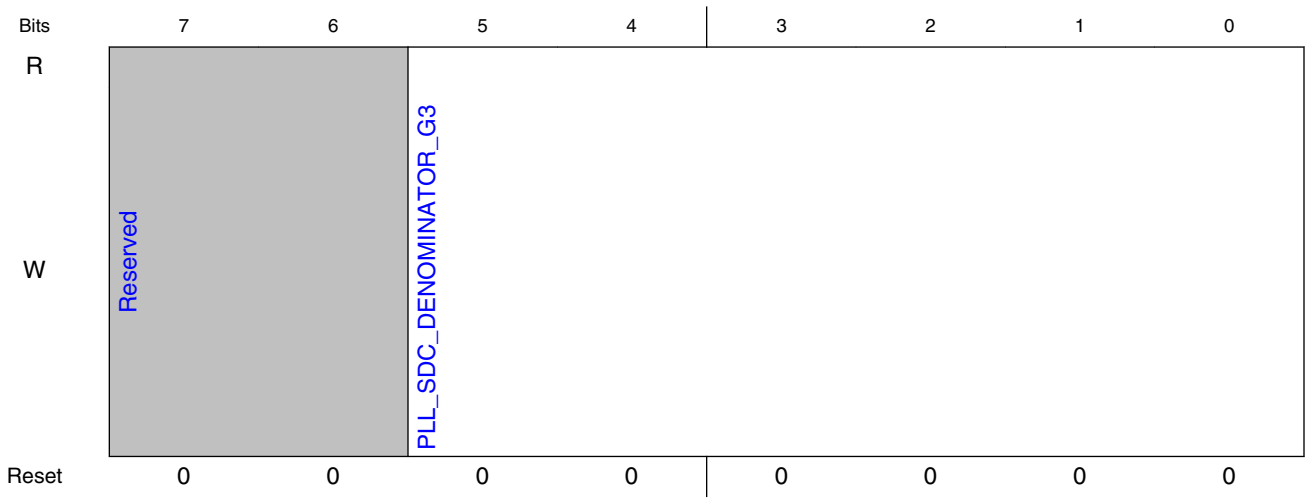
Field	Function
7-6 —	Reserved
5-0 PLL_SDC_DENOMINATOR_G2	[GEN2]

11.5.3.1.82 (CMN_REG050)

11.5.3.1.82.1 Offset

Register	Offset
CMN_REG050	140h

11.5.3.1.82.2 Diagram



11.5.3.1.82.3 Fields

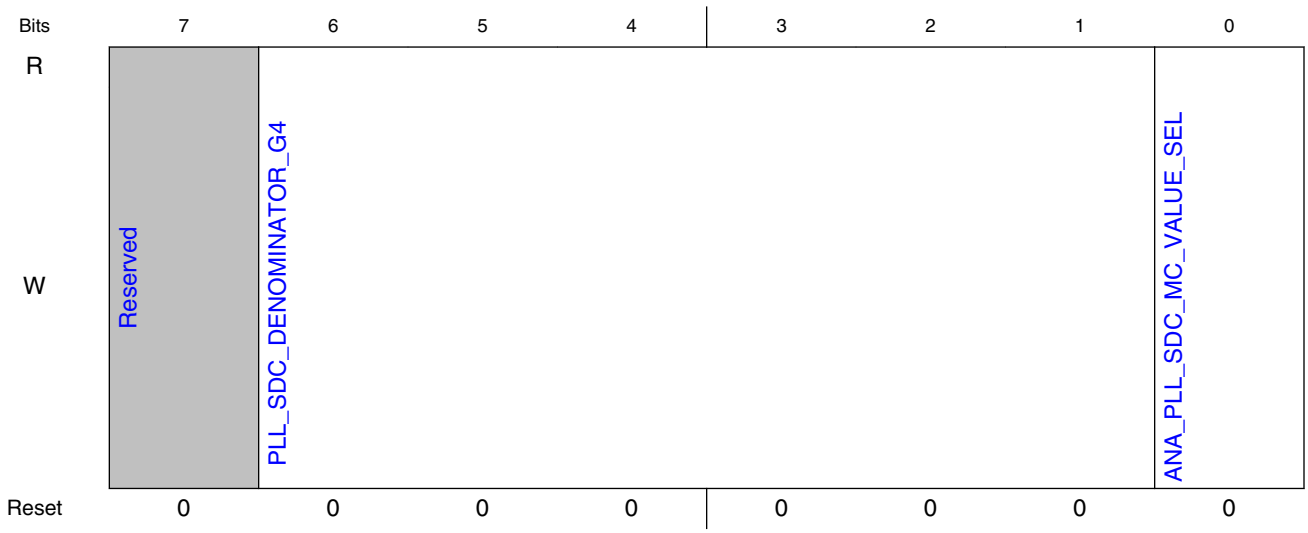
Field	Function
7-6 —	Reserved
5-0 PLL_SDC_DENOMINATOR_G3	[GEN3]

11.5.3.1.83 (CMN_REG051)

11.5.3.1.83.1 Offset

Register	Offset
CMN_REG051	144h

11.5.3.1.83.2 Diagram



11.5.3.1.83.3 Fields

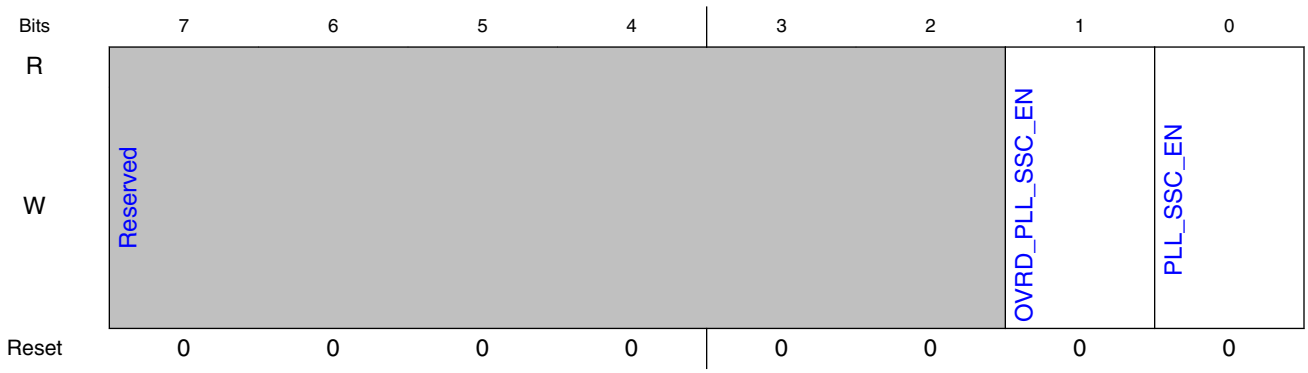
Field	Function
7 —	Reserved
6-1 PLL_SDC_DENOMINATOR_G4	[GEN4]
0 ANA_PLL_SDC_MC_VALUE_SEL	PLL SDC value force 0: SDC MC = sdc_n, 1: SDC_MC = sdc_n-1

11.5.3.1.84 (CMN_REG052)

11.5.3.1.84.1 Offset

Register	Offset
CMN_REG052	148h

11.5.3.1.84.2 Diagram



11.5.3.1.84.3 Fields

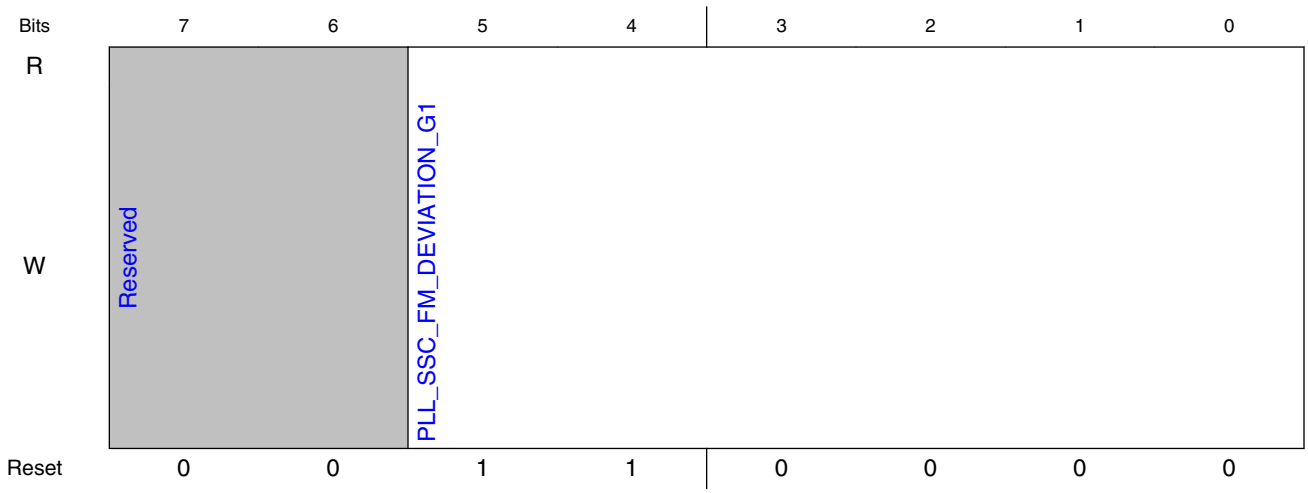
Field	Function
7-2 —	Reserved
1 OVRD_PLL_SSC_EN	Override enable for pll_ssc_en
0 PLL_SSC_EN	PLL SSC enable 0: Disable, 1: Enable

11.5.3.1.85 (CMN_REG053)

11.5.3.1.85.1 Offset

Register	Offset
CMN_REG053	14Ch

11.5.3.1.85.2 Diagram



11.5.3.1.85.3 Fields

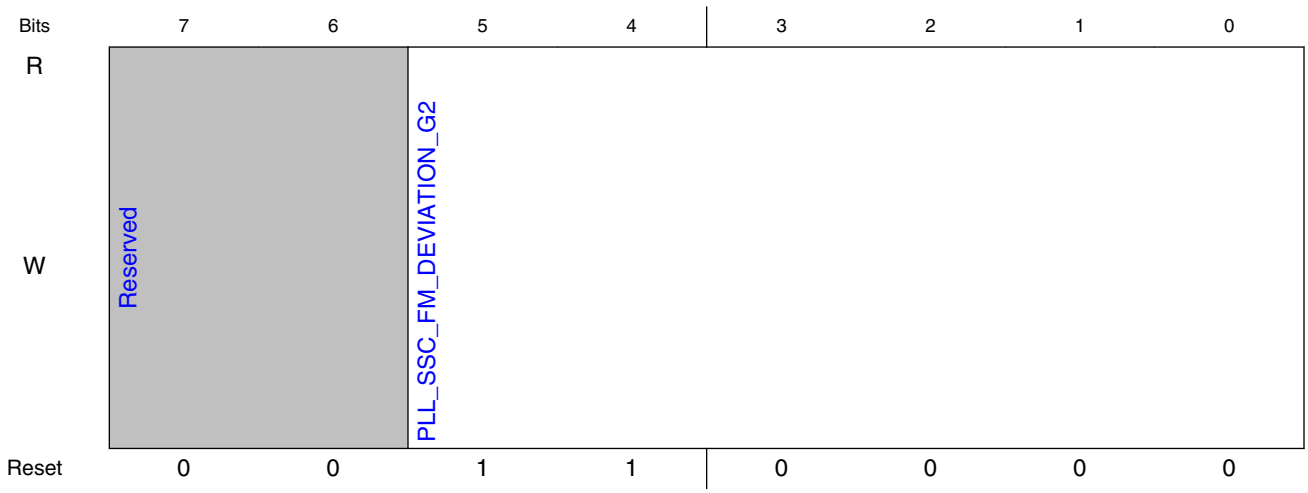
Field	Function
7-6 —	Reserved
5-0 PLL_SSC_FM_DEVIATION_G1	[GEN1] PLL SSC modulation deviation

11.5.3.1.86 (CMN_REG054)

11.5.3.1.86.1 Offset

Register	Offset
CMN_REG054	150h

11.5.3.1.86.2 Diagram



11.5.3.1.86.3 Fields

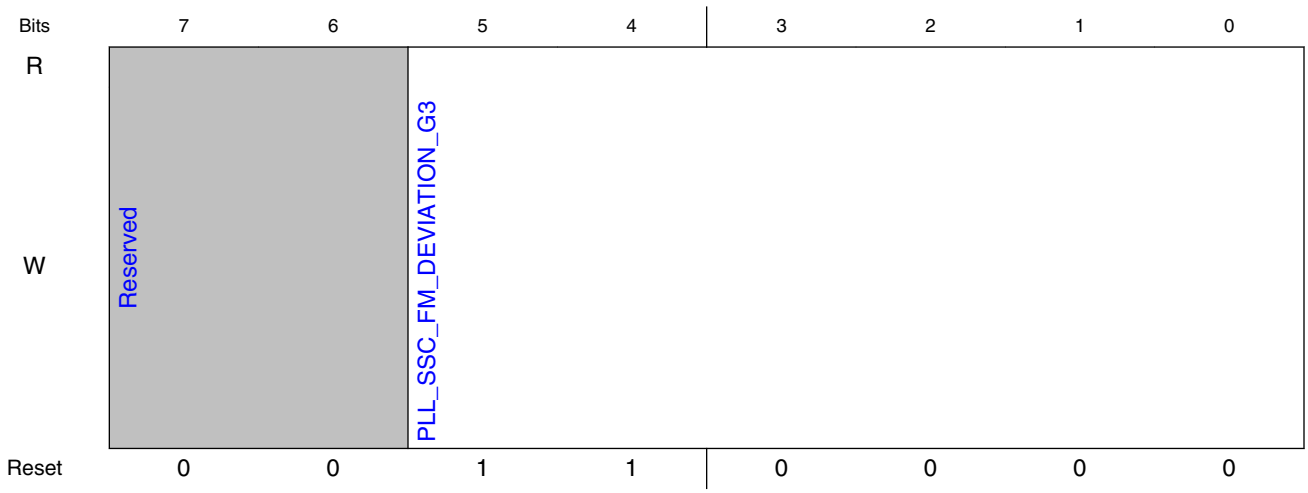
Field	Function
7-6 —	Reserved
5-0 PLL_SSC_FM_DEVIATION_G2	[GEN2]

11.5.3.1.87 (CMN_REG055)

11.5.3.1.87.1 Offset

Register	Offset
CMN_REG055	154h

11.5.3.1.87.2 Diagram



11.5.3.1.87.3 Fields

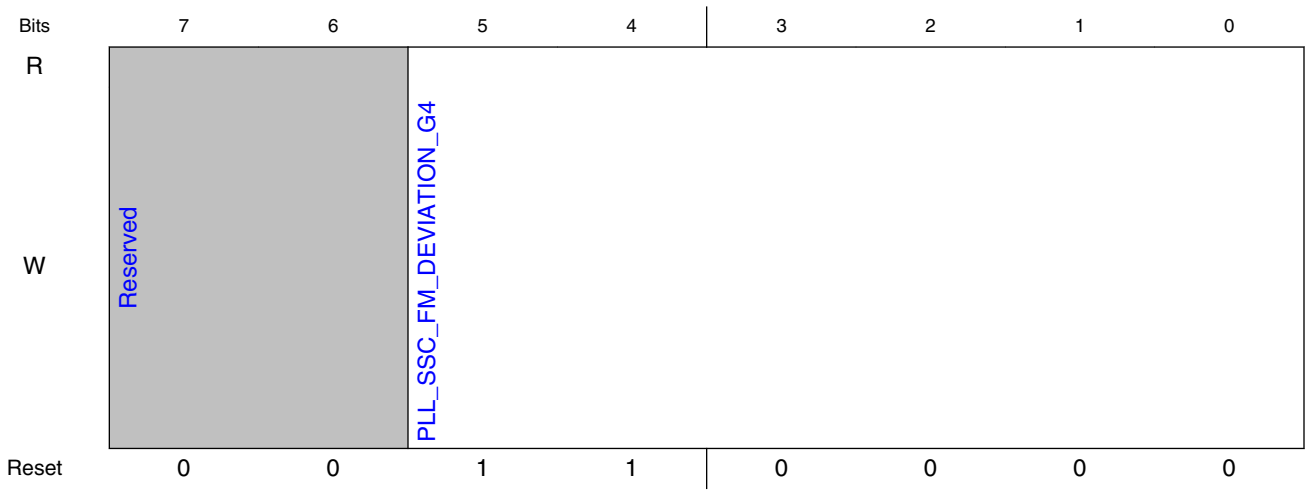
Field	Function
7-6 —	Reserved
5-0 PLL_SSC_FM_DEVIATION_G3	[GEN3]

11.5.3.1.88 (CMN_REG056)

11.5.3.1.88.1 Offset

Register	Offset
CMN_REG056	158h

11.5.3.1.88.2 Diagram



11.5.3.1.88.3 Fields

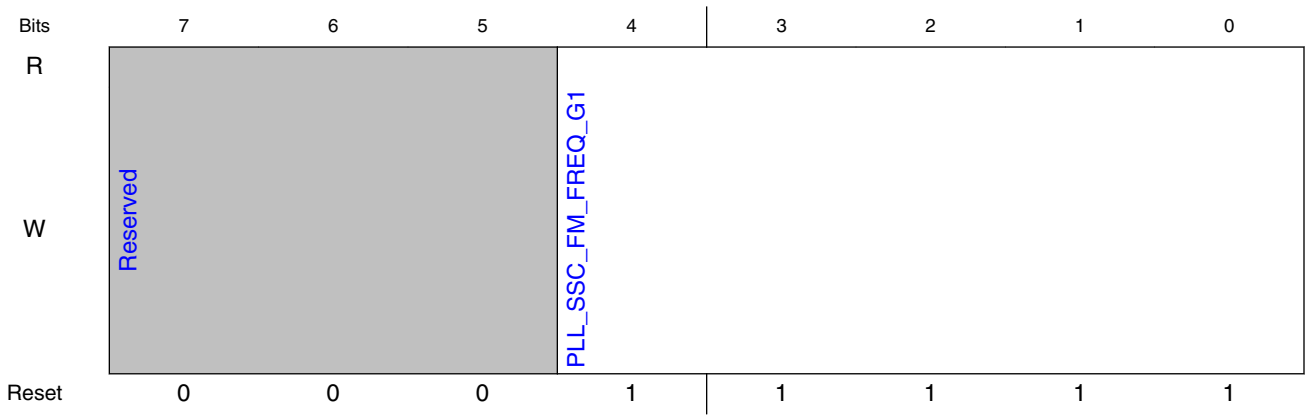
Field	Function
7-6 —	Reserved
5-0 PLL_SSC_FM_DEVIATION_G4	[GEN4]

11.5.3.1.89 (CMN_REG057)

11.5.3.1.89.1 Offset

Register	Offset
CMN_REG057	15Ch

11.5.3.1.89.2 Diagram



11.5.3.1.89.3 Fields

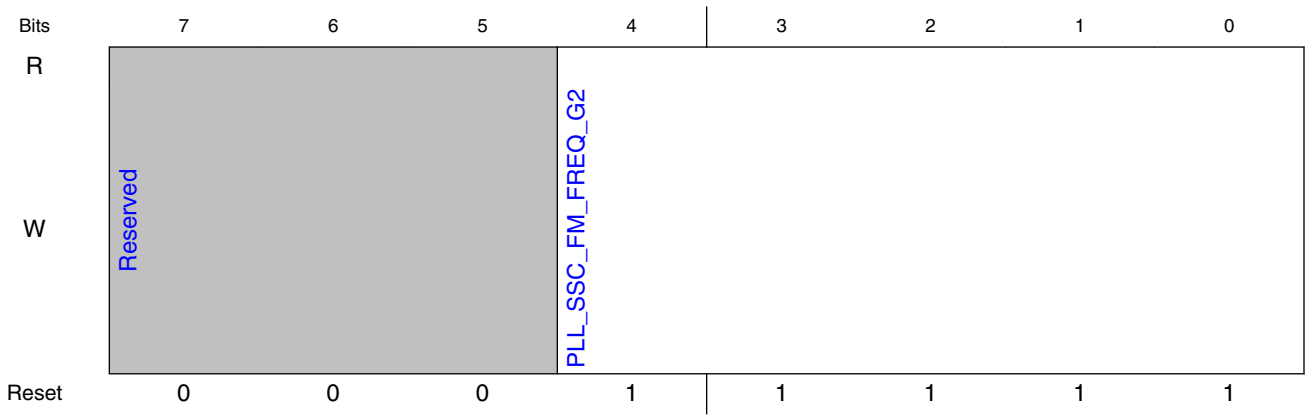
Field	Function
7-5 —	Reserved
4-0 PLL_SSC_FM_FREQ_G1	[GEN1] PLL SSC modulation frequency

11.5.3.1.90 (CMN_REG058)

11.5.3.1.90.1 Offset

Register	Offset
CMN_REG058	160h

11.5.3.1.90.2 Diagram



11.5.3.1.90.3 Fields

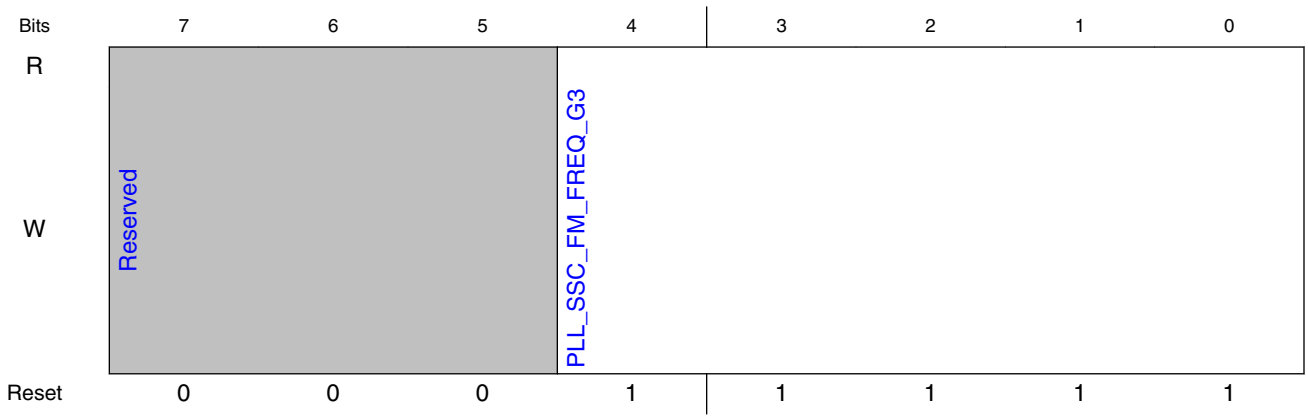
Field	Function
7-5 —	Reserved
4-0 PLL_SSC_FM_FREQ_G2	[GEN2]

11.5.3.1.91 (CMN_REG059)

11.5.3.1.91.1 Offset

Register	Offset
CMN_REG059	164h

11.5.3.1.91.2 Diagram



11.5.3.1.91.3 Fields

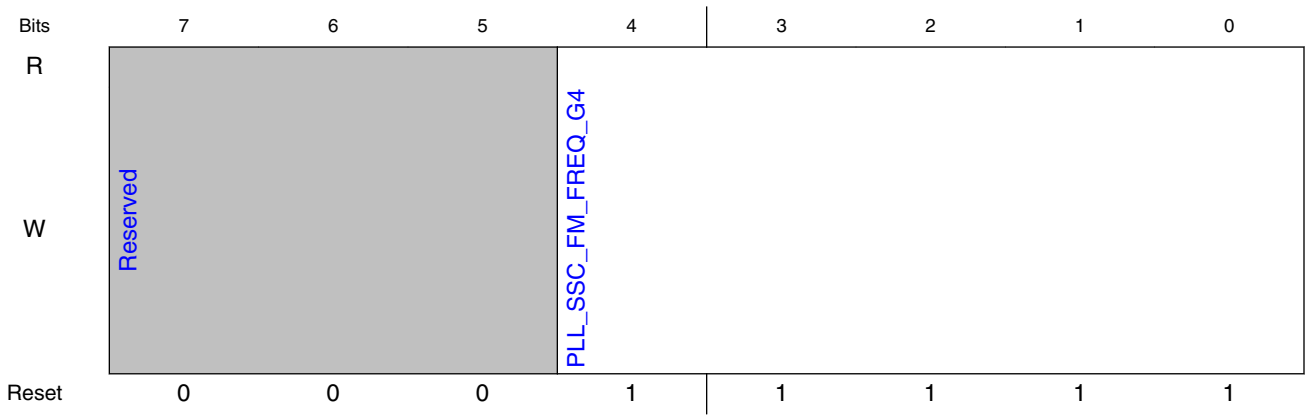
Field	Function
7-5 —	Reserved
4-0 PLL_SSC_FM_FREQ_G3	[GEN3]

11.5.3.1.92 (CMN_REG05A)

11.5.3.1.92.1 Offset

Register	Offset
CMN_REG05A	168h

11.5.3.1.92.2 Diagram



11.5.3.1.92.3 Fields

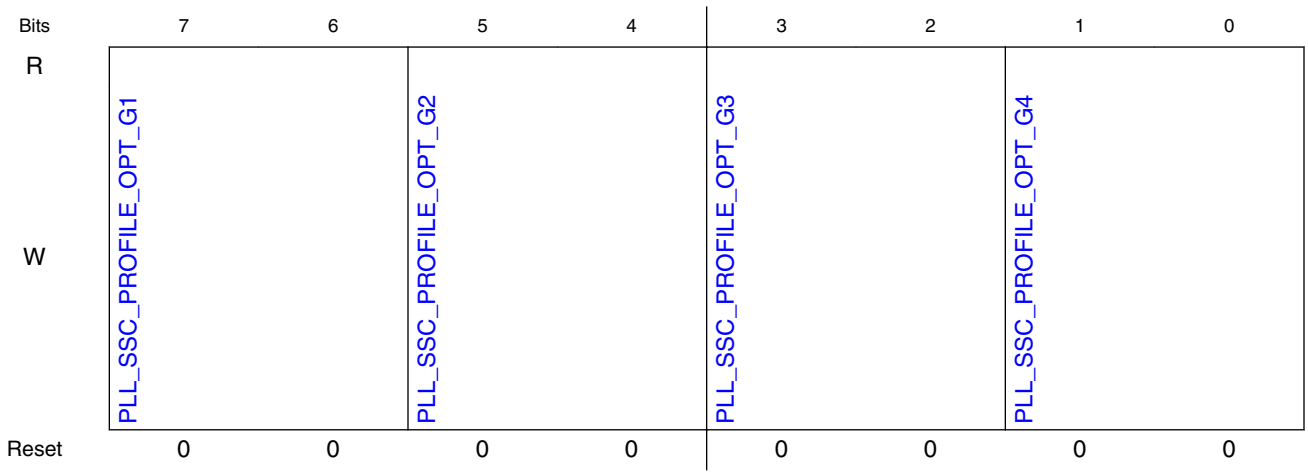
Field	Function
7-5 —	Reserved
4-0 PLL_SSC_FM_FREQ_G4	[GEN4]

11.5.3.1.93 (CMN_REG05B)

11.5.3.1.93.1 Offset

Register	Offset
CMN_REG05B	16Ch

11.5.3.1.93.2 Diagram



11.5.3.1.93.3 Fields

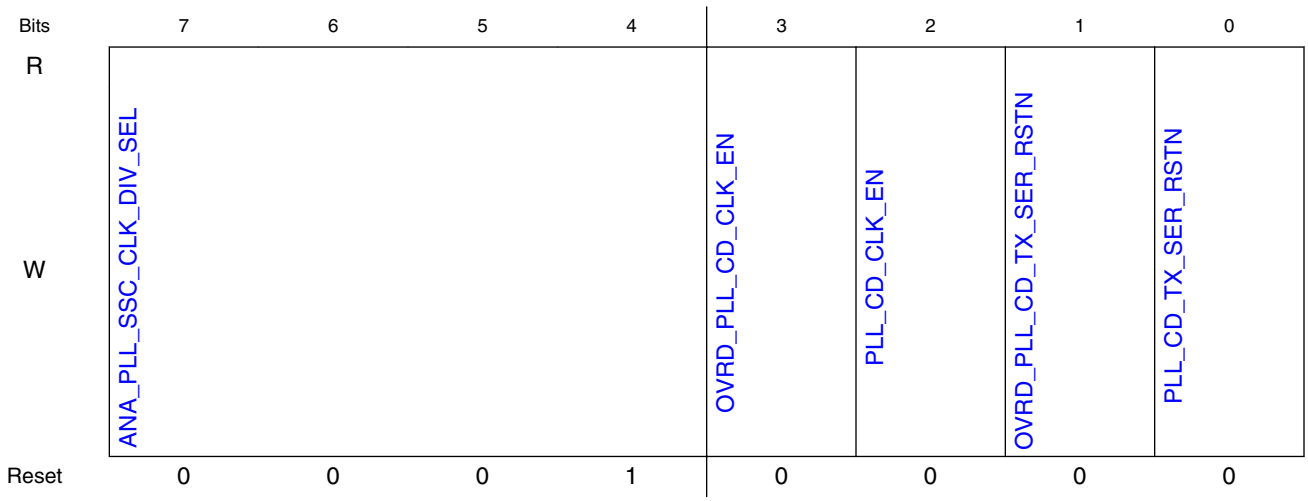
Field	Function
7-6 PLL_SSC_PROFILE_OPT_G1	[GEN1] PLL SSC modulation profile shape control 00: No jump, x1:2-level jump high edge , 1x: 2-level jump low edge
5-4 PLL_SSC_PROFILE_OPT_G2	[GEN2]
3-2 PLL_SSC_PROFILE_OPT_G3	[GEN3]
1-0 PLL_SSC_PROFILE_OPT_G4	[GEN4]

11.5.3.1.94 (CMN_REG05C)

11.5.3.1.94.1 Offset

Register	Offset
CMN_REG05C	170h

11.5.3.1.94.2 Diagram



11.5.3.1.94.3 Fields

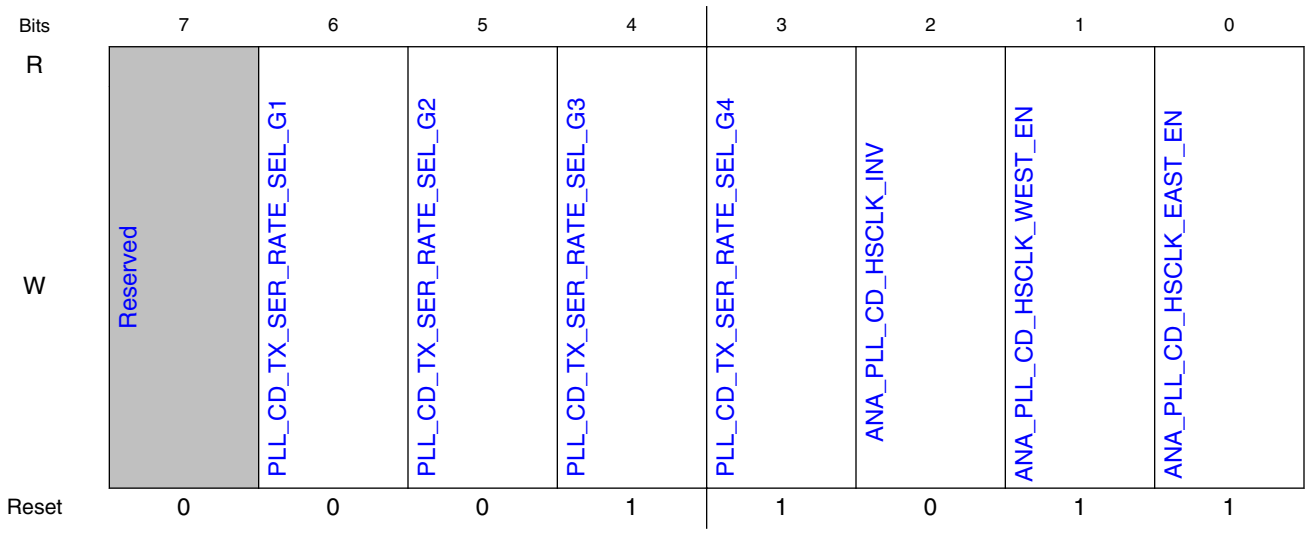
Field	Function
7-4 ANA_PLL_SSC_CLK_DIV_SEL	PLL SSC clock divide ratio 0000: N/A, 0001: /1, 0010: /2, …, 1111: /15
3 OVRD_PLL_CD_CLK_EN	Override enable for pll_cd_clk_en
2 PLL_CD_CLK_EN	CD enable 0: Disable, 1: Enable
1 OVRD_PLL_CD_TX_SER_RSTN	Override enable for pll_cd_tx_ser_rstn
0 PLL_CD_TX_SER_RSTN	TX_SER resetn

11.5.3.1.95 (CMN_REG05D)

11.5.3.1.95.1 Offset

Register	Offset
CMN_REG05D	174h

11.5.3.1.95.2 Diagram



11.5.3.1.95.3 Fields

Field	Function
7 —	Reserved
6 PLL_CD_TX_SER_RATE_SEL_G1	[GEN1] TX serializer data rate selection for Gen4 (Need to be controlled with i_tx_en_40bit) 0: 20/40-bit 1: 16/32-bit
5 PLL_CD_TX_SER_RATE_SEL_G2	[GEN2]
4 PLL_CD_TX_SER_RATE_SEL_G3	[GEN3]
3 PLL_CD_TX_SER_RATE_SEL_G4	[GEN4]
2 ANA_PLL_CD_HSCLK_INV	CD output clock polarity inversion 0: No swap, 1: P/N swap
1	CD driver nmos strength control N/A

Table continues on the next page...

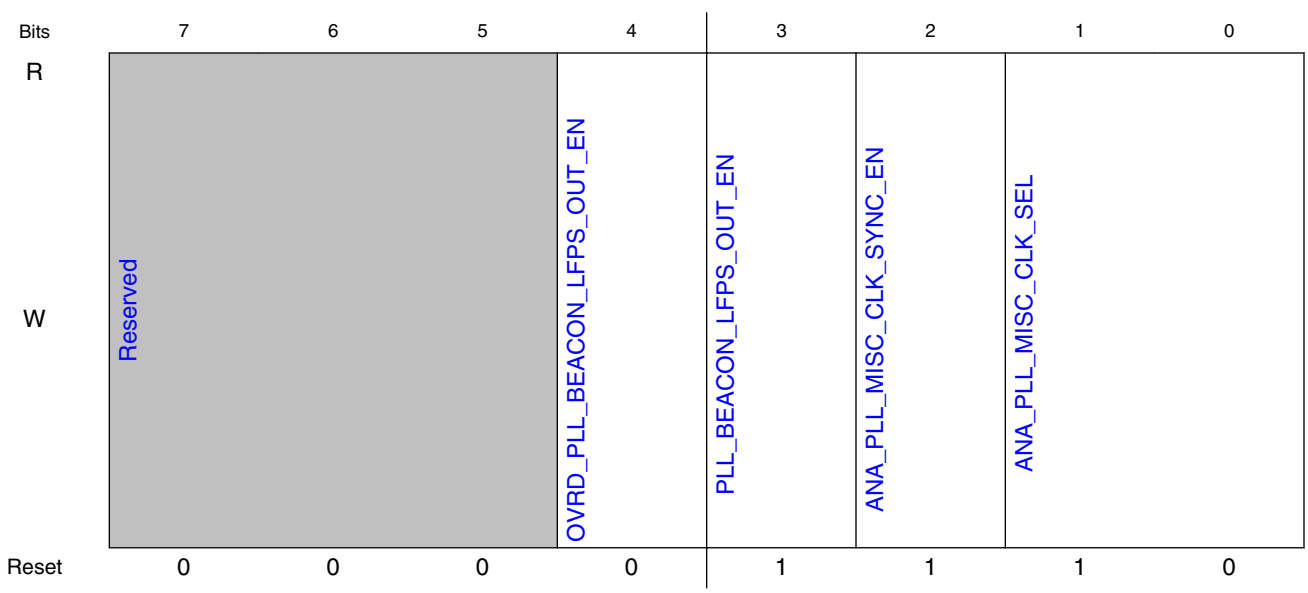
Field	Function
ANA_PLL_CD_HSCLK_WEST_EN	
0 ANA_PLL_CD_HSCLK_EAST_EN	CD driver pmos strength control N/A

11.5.3.1.96 (CMN_REG05E)

11.5.3.1.96.1 Offset

Register	Offset
CMN_REG05E	178h

11.5.3.1.96.2 Diagram



11.5.3.1.96.3 Fields

Field	Function
7-5 —	Reserved

Table continues on the next page...

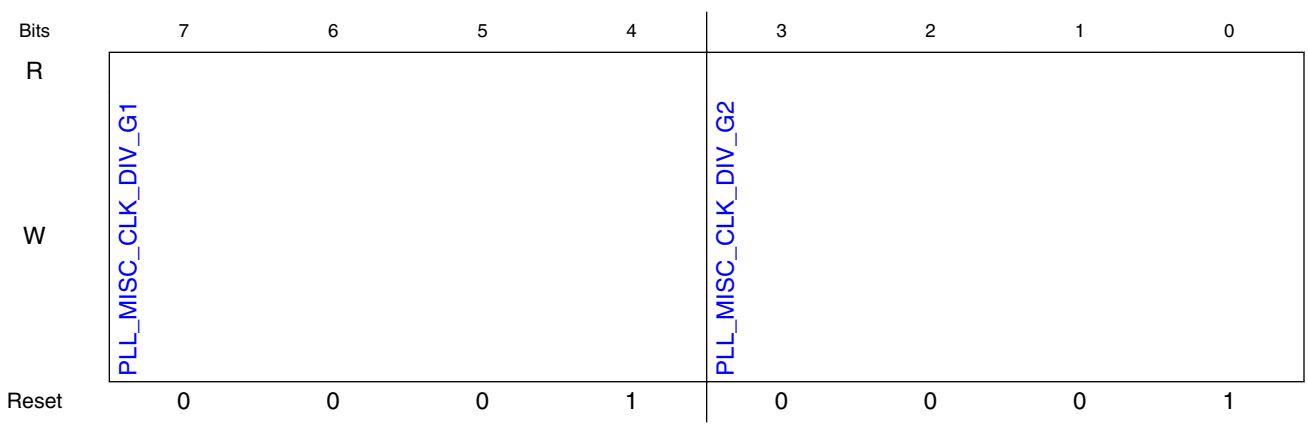
Field	Function
4 OVRD_PLL_BEACON_LFPS_OUT_EN	Override enable for pll_beacon_lfps_out_en
3 PLL_BEACON_LFPS_OUT_EN	TX beacon clock enable 0: Disable, 1: Enable
2 ANA_PLL_MISC_CLK_SYNC_EN	PLL miscellaneous clock synchronization enable If enabled, PLL misc clock output is self-synchronized so that clock shape is not affedted by LFPS/ beacon enable timing. 0: Disable (async), 1: Enalbe (sync)
1-0 ANA_PLL_MISC_CLK_SEL	PLL low-frequency clock output source selection 00: Reference clock, 01: Config clock from SoC, 10: Internal oscillator clock, 11: Divided PLL clock

11.5.3.1.97 (CMN_REG05F)

11.5.3.1.97.1 Offset

Register	Offset
CMN_REG05F	17Ch

11.5.3.1.97.2 Diagram



11.5.3.1.97.3 Fields

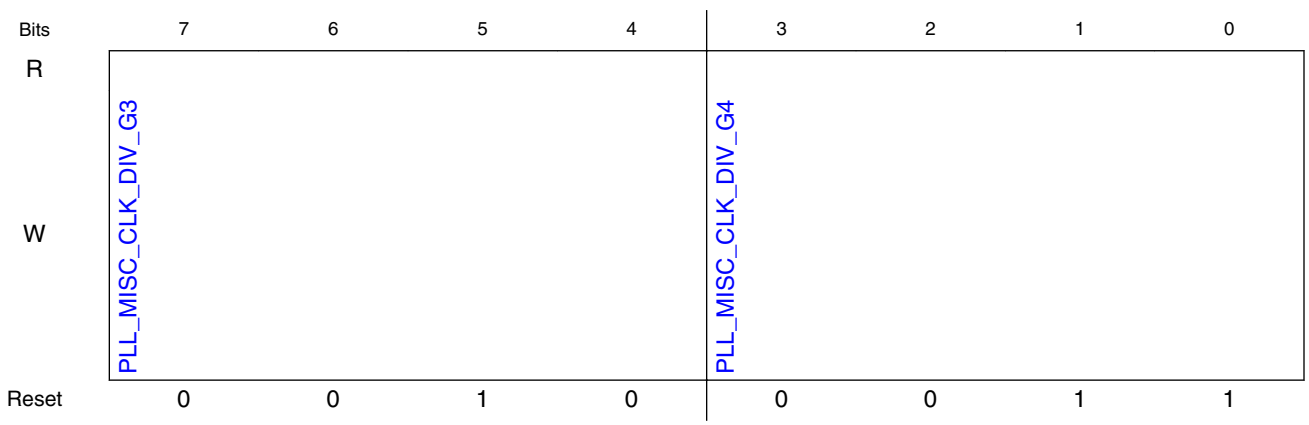
Field	Function
7-4 PLL_MISC_CLK_DIV_G1	[GEN1] PLL miscellaneous clock divider ratio 0000:
3-0 PLL_MISC_CLK_DIV_G2	[GEN2]

11.5.3.1.98 (CMN_REG060)

11.5.3.1.98.1 Offset

Register	Offset
CMN_REG060	180h

11.5.3.1.98.2 Diagram



11.5.3.1.98.3 Fields

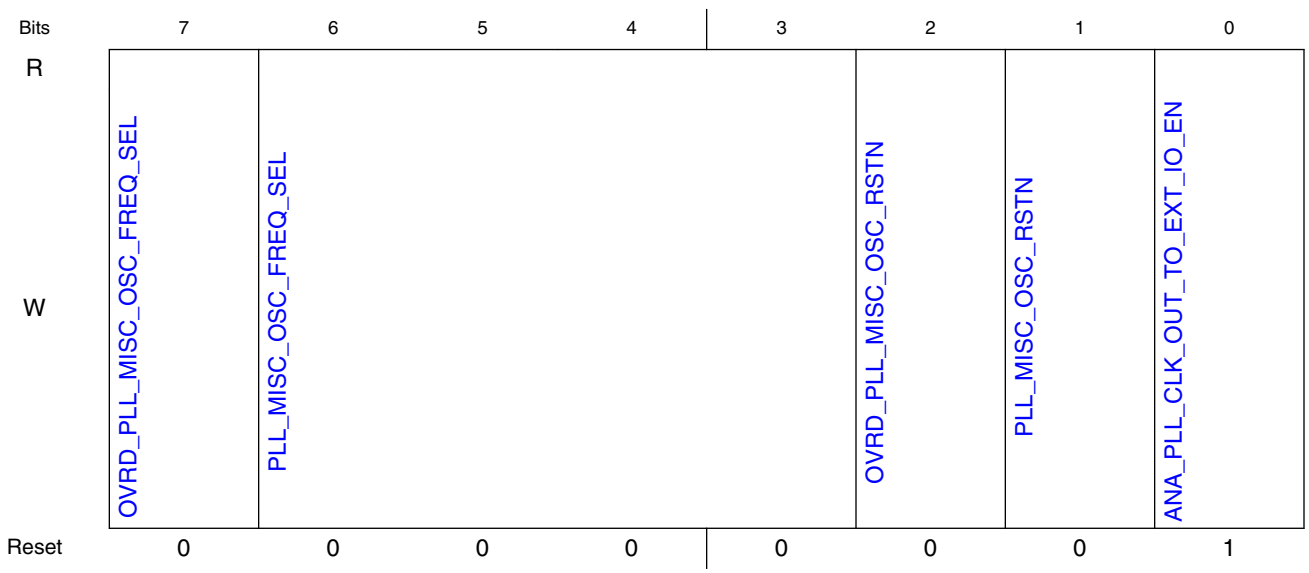
Field	Function
7-4 PLL_MISC_CLK_DIV_G3	[GEN3]
3-0 PLL_MISC_CLK_DIV_G4	[GEN4]

11.5.3.1.99 (CMN_REG061)

11.5.3.1.99.1 Offset

Register	Offset
CMN_REG061	184h

11.5.3.1.99.2 Diagram



11.5.3.1.99.3 Fields

Field	Function
7 OVRD_PLL_MISC_OSC_FREQ_SEL	Override enable for pll_misc_osc_freq_sel
6-3 PLL_MISC_OSC_FREQ_SEL	PLL miscellaneous clock frequency selection Since it is usually automatically determined by oscillator calibration, this is used to force the frequency w/o calibration. 0000: lowest frequency, 1111: highest frequency
2 OVRD_PLL_MISC_OSC_RSTN	Override enable for pll_misc_osc_rstn

Table continues on the next page...

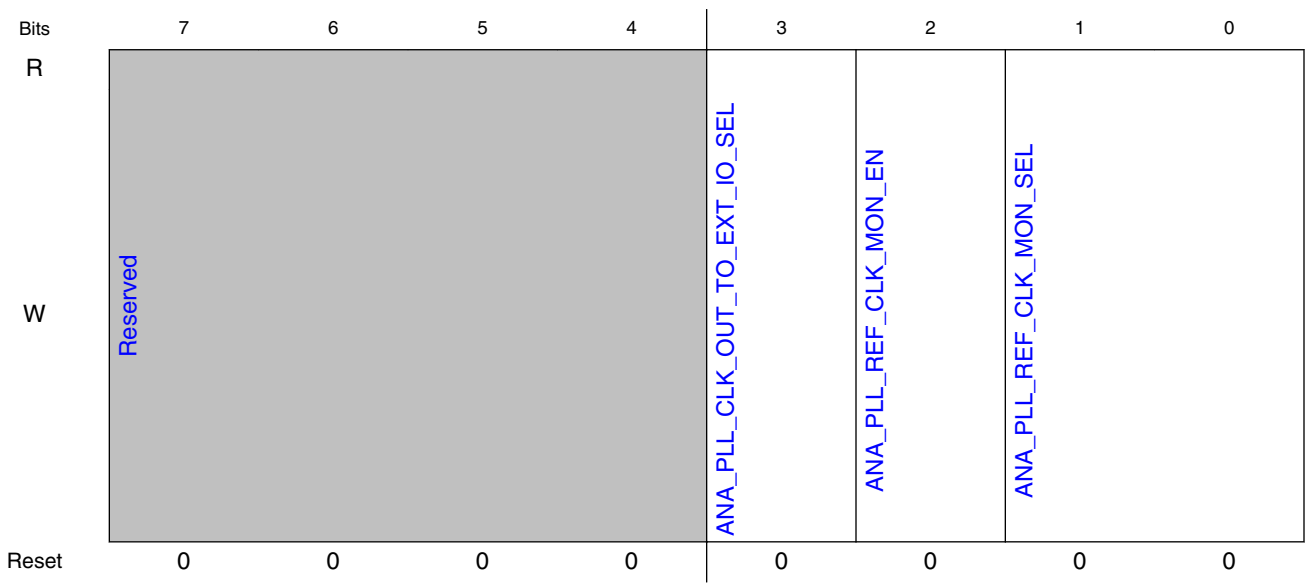
Field	Function
1 PLL_MISC_OSC_RSTN	PLL miscellaneous clock oscillator reset 0: Reset, 1: Released
0 ANA_PLL_CLK_OUT_TO_EXT_IO_EN	PLL low-frequency clock output to external I/O enable 0: Disable, 1: Enable

11.5.3.1.100 (CMN_REG062)

11.5.3.1.100.1 Offset

Register	Offset
CMN_REG062	188h

11.5.3.1.100.2 Diagram



11.5.3.1.100.3 Fields

Field	Function
7-4 —	Reserved

Table continues on the next page...

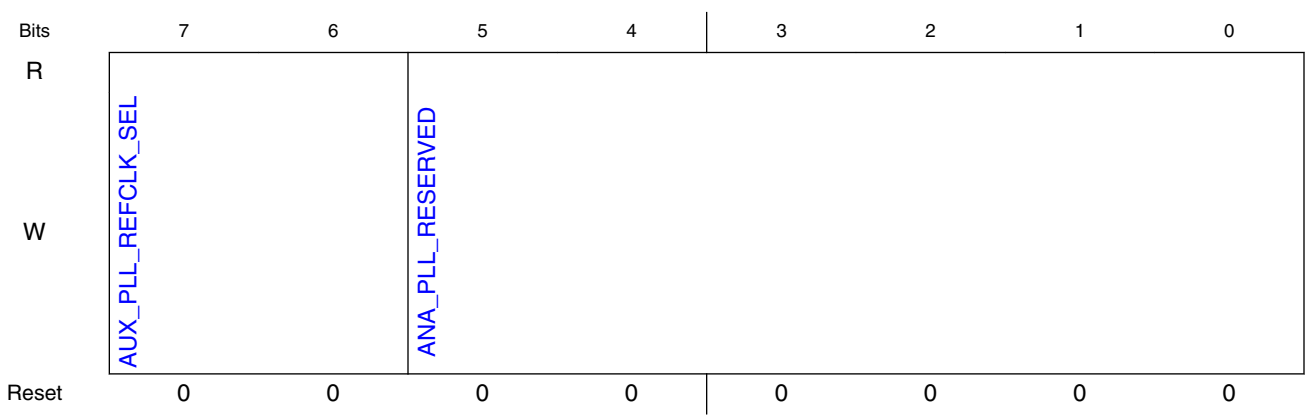
Field	Function
3 ANA_PLL_CLK_OUT_TO_EXT_IO_SEL	PLL low-frequency clock output to external I/O source selection 0: 100MHz clock generated by PLL (for PCIe), 1: Reference clock bypass (for MPHY)
2 ANA_PLL_REF_CLK_MON_EN	PLL reference clock monitor enable 0: Disable, 1: Enable
1-0 ANA_PLL_REF_CLK_MON_SEL	PLL reference clock selection for monitor 00: FB clock 01: REF clock 10: Chopper clock 11: Bypass clock

11.5.3.1.101 (CMN_REG063)

11.5.3.1.101.1 Offset

Register	Offset
CMN_REG063	18Ch

11.5.3.1.101.2 Diagram



11.5.3.1.101.3 Fields

Field	Function
7-6	0X: AUX_IN (PLL clock) 10: I_PLL_REFCLK_FROM_IO

Table continues on the next page...

Field	Function
AUX_PLL_REFCLK_SEL	11: I_PLL_REFCLK_FROM_SYSPLL
5-0 ANA_PLL_RESERVED	PLL Reserved pins

11.5.3.1.102 (CMN_REG064)

11.5.3.1.102.1 Offset

Register	Offset
CMN_REG064	190h

11.5.3.1.102.2 Diagram

Bits	7	6	5	4	3	2	1	0
R								
W	ANA_AUX_RX_TX_SEL	OVRD_AUX_EN	AUX_EN	ANA_AUX_RX_CAP_BYPASS	ANA_AUX_RX_TERM_GND_EN	ANA_AUX_TX_TERM		
Reset	0	0	0	0	1	0	1	0

11.5.3.1.102.3 Fields

Field	Function
7 ANA_AUX_RX_TX_SEL	Select mode (TX or RX) 0: RX, 1:TX
6 OVRD_AUX_EN	Override enable for aux_en
5	AUX Enable

Table continues on the next page...

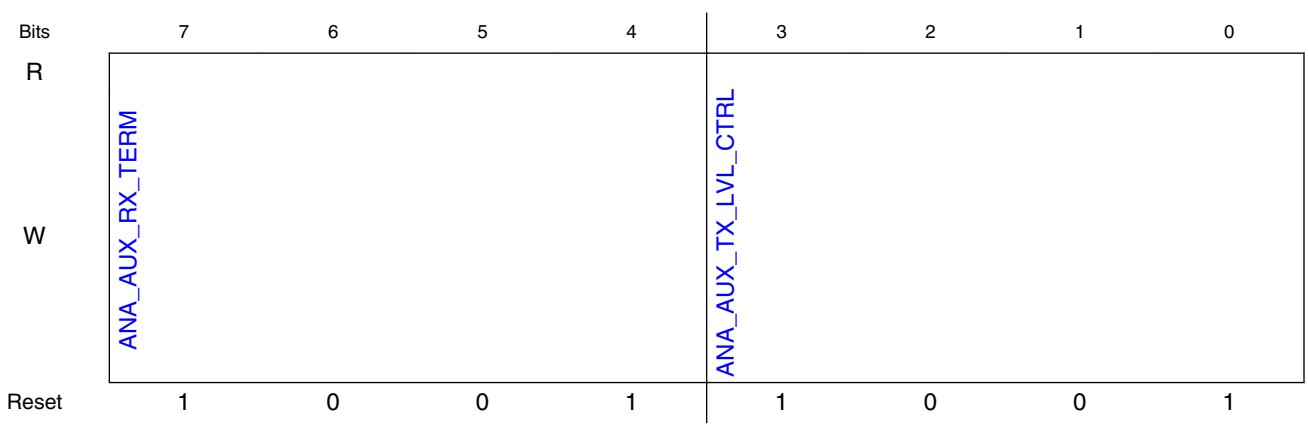
Field	Function
AUX_EN	0:Disable 1:Enable
4 ANA_AUX_RX_CAP_BYPASS	External reference clock I/O AC-coupling capacitor bypass enable 0: Bypass AC coupling cap, 1: Use AC coupling cap
3 ANA_AUX_RX_TERM_GND_EN	External reference clock I/O termination to ground 0: Disable termination to GND , 1: Enable termination to GND
2-0 ANA_AUX_TX_TERM	TX termination resistor control. Default code : 010, 50.7Ω 30Ω (000) ~ 79Ω (111)

11.5.3.1.103 (CMN_REG065)

11.5.3.1.103.1 Offset

Register	Offset
CMN_REG065	194h

11.5.3.1.103.2 Diagram



11.5.3.1.103.3 Fields

Field	Function
7-4	RX termination resistor control. Default code : 1001, 99.6Ω 77Ω (0000) ~ High impedance state, Hi-Z (1111)

Table continues on the next page...

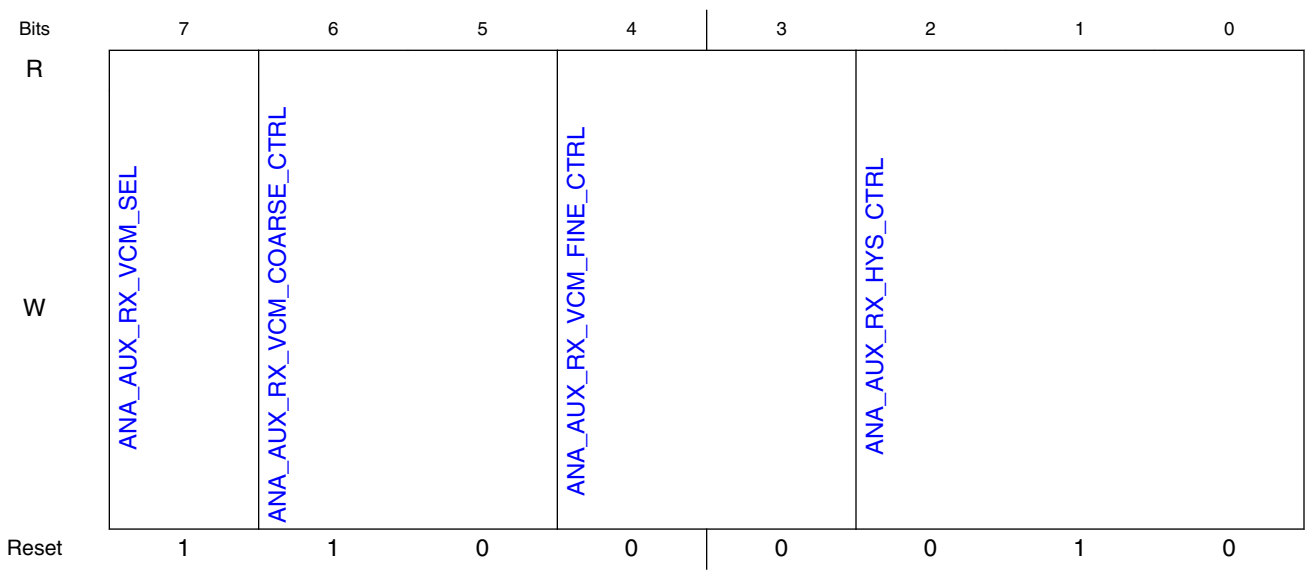
Field	Function
ANA_AUX_RX_TERM	
3-0 ANA_AUX_TX_LVL_CTRL	TX Amplitude resistor control. Default code : 101, 375mVpp 187.5mVpp (000) ~ 750mVpp (111)

11.5.3.1.104 (CMN_REG066)

11.5.3.1.104.1 Offset

Register	Offset
CMN_REG066	198h

11.5.3.1.104.2 Diagram



11.5.3.1.104.3 Fields

Field	Function
7 ANA_AUX_RX_VCM_SEL	Input common mode voltage control. 0: Disalbe, 1: Enable
6-5	VCM of RX control. 171mV ~ 680mV at typical condition.

Table continues on the next page...

Field	Function
ANA_AUX_RX_VCM_COARSE_CTRL	Default code : 10 (637mV)
4-3 ANA_AUX_RX_VCM_FINE_CTRL	VCM of RX control. 171mV ~ 680mV at typical condition. Default code : 00 (637mV)
2-0 ANA_AUX_RX_HYS_CTRL	Hysteresis for RX noise blocking control. 0mv ~ 103.6mv (must sense more than 140mv)

11.5.3.1.105 (CMN_REG067)

11.5.3.1.105.1 Offset

Register	Offset
CMN_REG067	19Ch

11.5.3.1.105.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	ANA_AUX_RESERVED							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.105.3 Fields

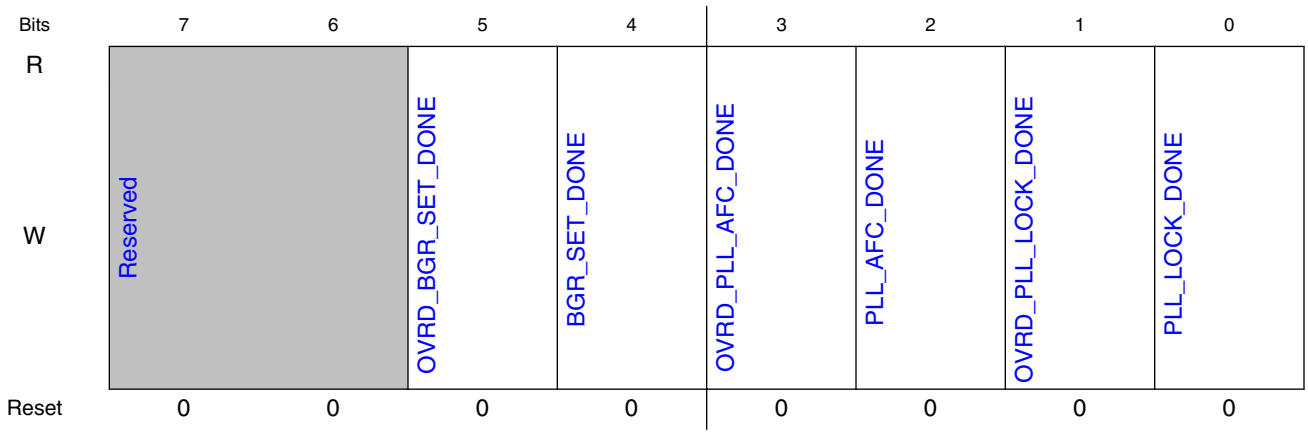
Field	Function
7-0 ANA_AUX_RESERVED	Reserved port

11.5.3.1.106 (CMN_REG068)

11.5.3.1.106.1 Offset

Register	Offset
CMN_REG068	1A0h

11.5.3.1.106.2 Diagram



11.5.3.1.106.3 Fields

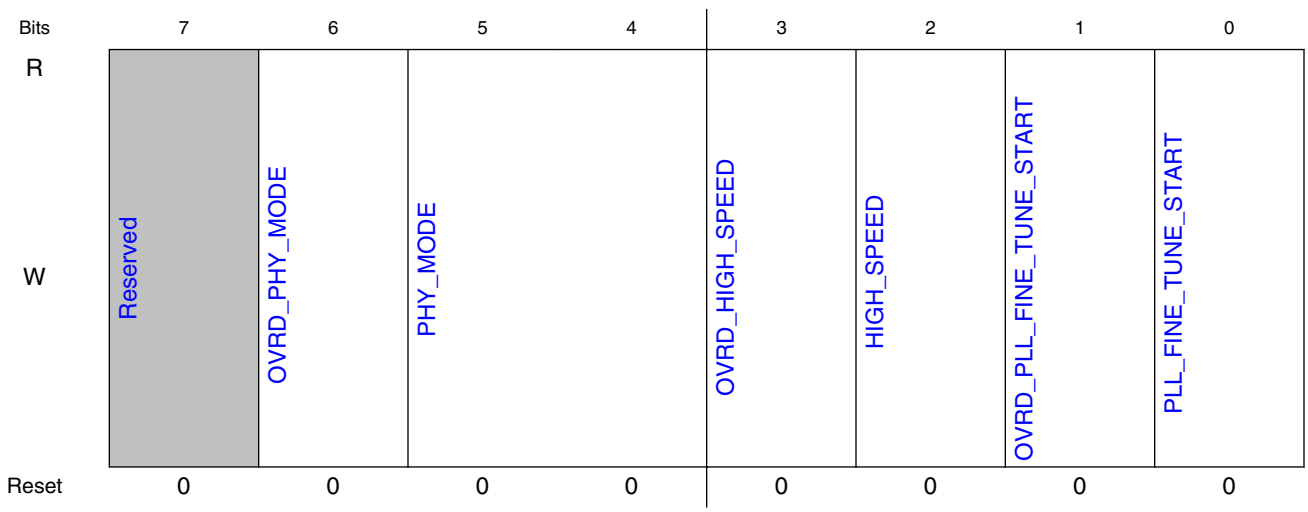
Field	Function
7-6 —	Reserved
5 OVRD_BGR_SET_DONE	Override enable for bgr_set_done
4 BGR_SET_DONE	BGR set done
3 OVRD_PLL_AFC_DONE	Override enable for pll_afc_done
2 PLL_AFC_DONE	PLL AFC done overide value
1 OVRD_PLL_LOCK_DONE	Override enable for pll_lock_done
0 PLL_LOCK_DONE	PLL lock done overide value

11.5.3.1.107 (CMN_REG069)

11.5.3.1.107.1 Offset

Register	Offset
CMN_REG069	1A4h

11.5.3.1.107.2 Diagram



11.5.3.1.107.3 Fields

Field	Function
7 —	Reserved
6 OVRD_PHY_MODE	Override enable for phy_mode
5-4 PHY_MODE	
3 OVRD_HIGH_SPEED	Override enable for high_speed
2 HIGH_SPEED	HIGH SPEED indicator by operating LC VCO 0: Disable, 1: Enable

Table continues on the next page...

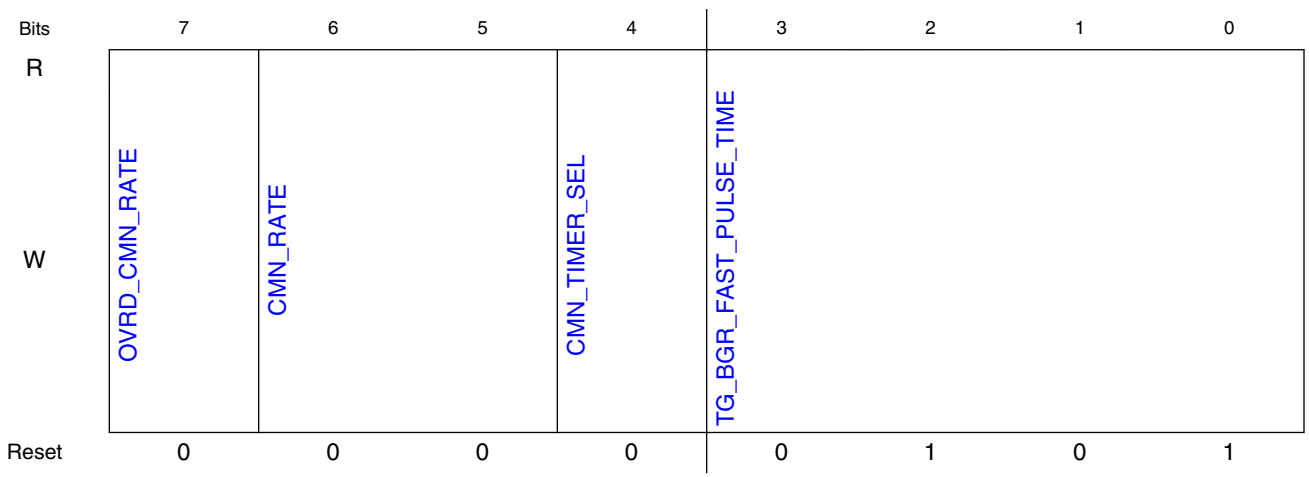
Field	Function
1 OVRD_PLL_FINE_TUNE_START	Override enable for pll_fine_tune_start
0 PLL_FINE_TUNE_START	

11.5.3.1.108 (CMN_REG06A)

11.5.3.1.108.1 Offset

Register	Offset
CMN_REG06A	1A8h

11.5.3.1.108.2 Diagram



11.5.3.1.108.3 Fields

Field	Function
7 OVRD_CMN_RATE	Override enable for cmn_rate
6-5 CMN_RATE	TX Data Rate manual setting 00: Gen1, 01: Gen2, 10: Gen3, 11:Gen4

Table continues on the next page...

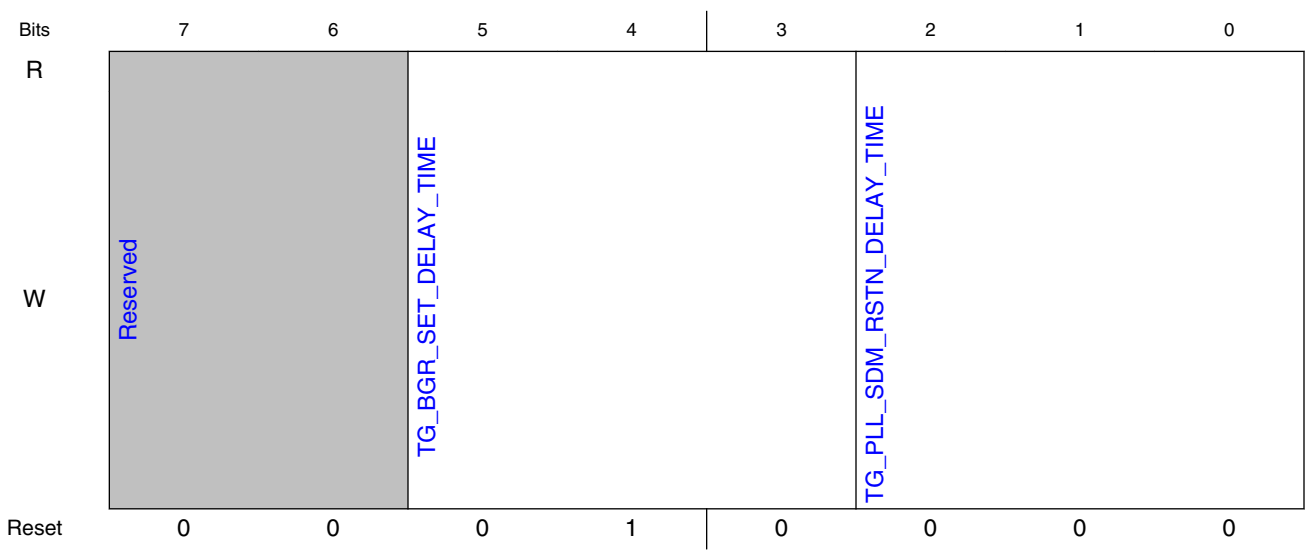
Field	Function
4 CMN_TIMER_SEL	
3-0 TG_BGR_FAST_PULSE_TIME	BGR LPF bypass duration after BGR_EN = 1

11.5.3.1.109 (CMN_REG06B)

11.5.3.1.109.1 Offset

Register	Offset
CMN_REG06B	1ACh

11.5.3.1.109.2 Diagram



11.5.3.1.109.3 Fields

Field	Function
7-6 —	Reserved
5-3	

Table continues on the next page...

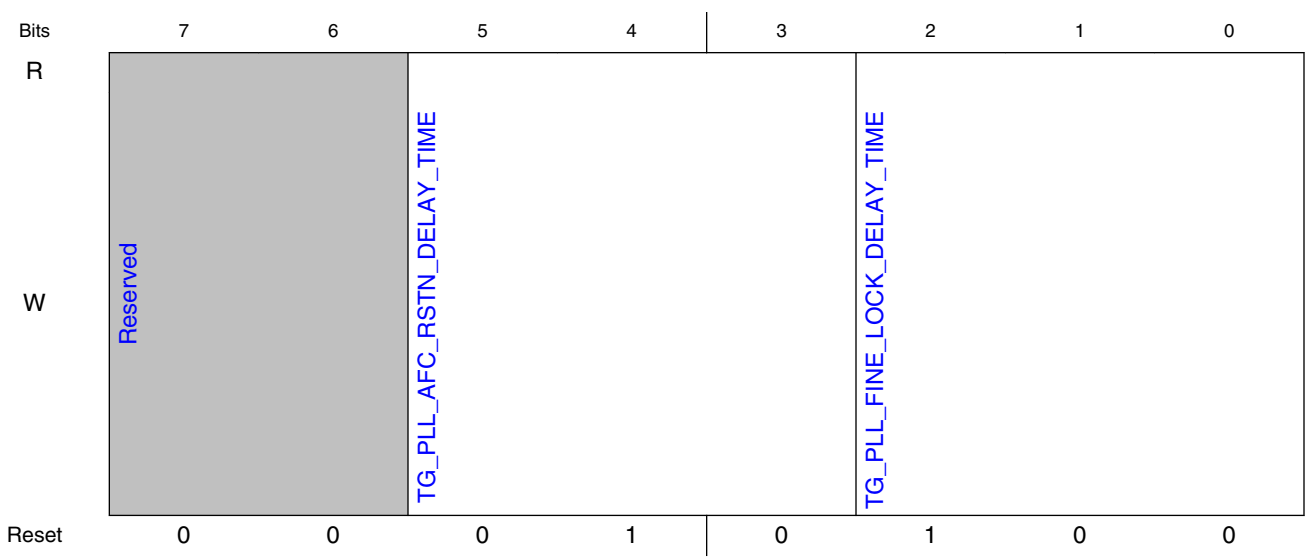
Field	Function
TG_BGR_SET_DELAY_TIME	
2-0 TG_PLL_SDM_RSTN_DELAY_TIME	PLL SDM start delay after PLL integer-mode lock(PLL lock)

11.5.3.1.110 (CMN_REG06C)

11.5.3.1.110.1 Offset

Register	Offset
CMN_REG06C	1B0h

11.5.3.1.110.2 Diagram



11.5.3.1.110.3 Fields

Field	Function
7-6 —	Reserved
5-3	PLL AFC reset delay time after bypassing BGR LPF

Table continues on the next page...

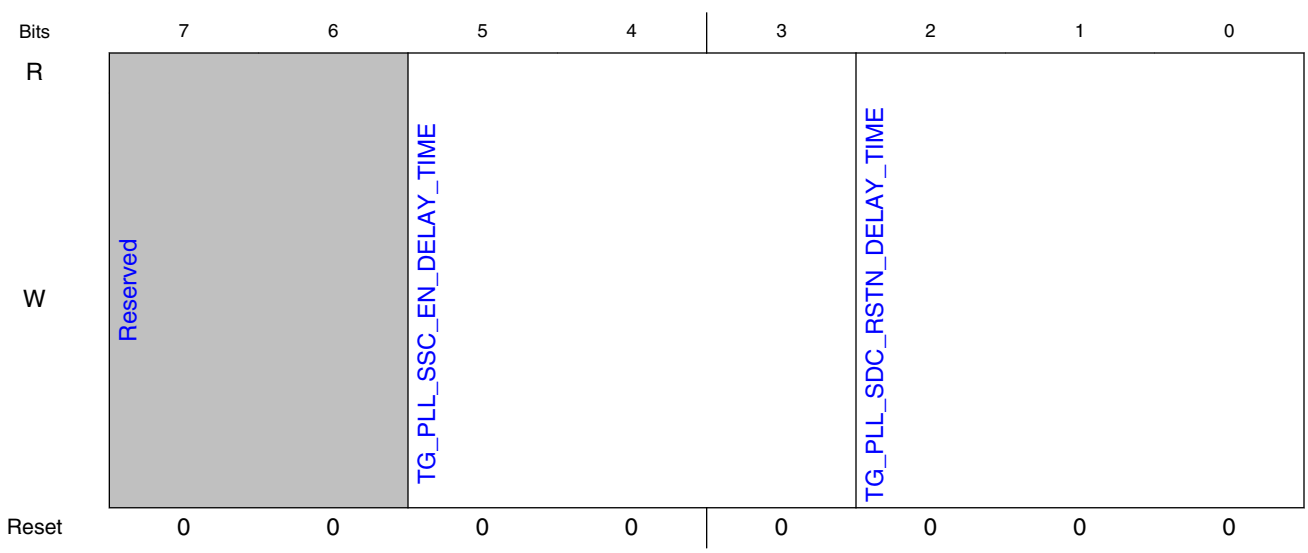
Field	Function
TG_PLL_AFC_RSTN_DELAY_TIME	
2-0 TG_PLL_FINE_LOCK_DELAY_TIME	PLL Fine LOCK DLY CODE

11.5.3.1.111 (CMN_REG06D)

11.5.3.1.111.1 Offset

Register	Offset
CMN_REG06D	1B4h

11.5.3.1.111.2 Diagram



11.5.3.1.111.3 Fields

Field	Function
7-6 —	Reserved
5-3	PLL SSC start delay time after

Table continues on the next page...

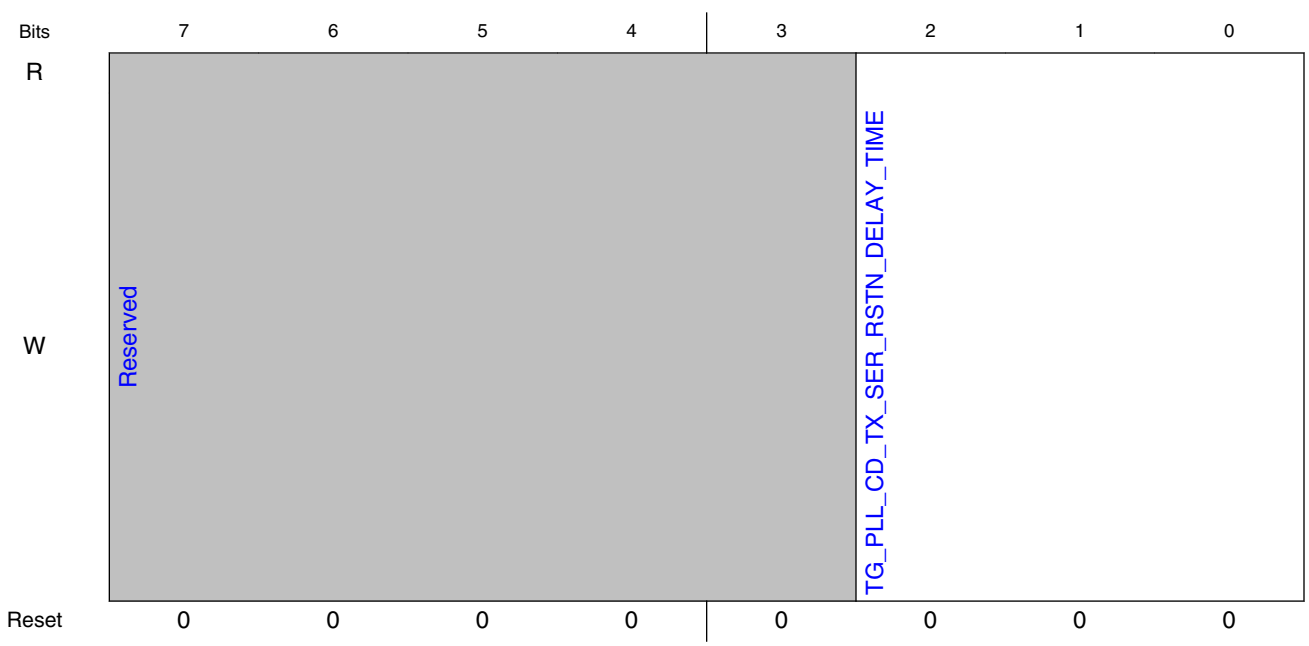
Field	Function
TG_PLL_SSC_EN_DELAY_TIME	
2-0 TG_PLL_SDC_RSTN_DELAY_TIME	PLL SDM RESET STABLE DLY CODE

11.5.3.1.112 (CMN_REG06E)

11.5.3.1.112.1 Offset

Register	Offset
CMN_REG06E	1B8h

11.5.3.1.112.2 Diagram



11.5.3.1.112.3 Fields

Field	Function
7-3	Reserved

Table continues on the next page...

Field	Function
—	
2-0 TG_PLL_CD_TX_SER_RSTN_DELAY_TIME	

11.5.3.1.113 (CMN_REG06F)

11.5.3.1.113.1 Offset

Register	Offset
CMN_REG06F	1BCh

11.5.3.1.113.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	DTB_SEL							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.113.3 Fields

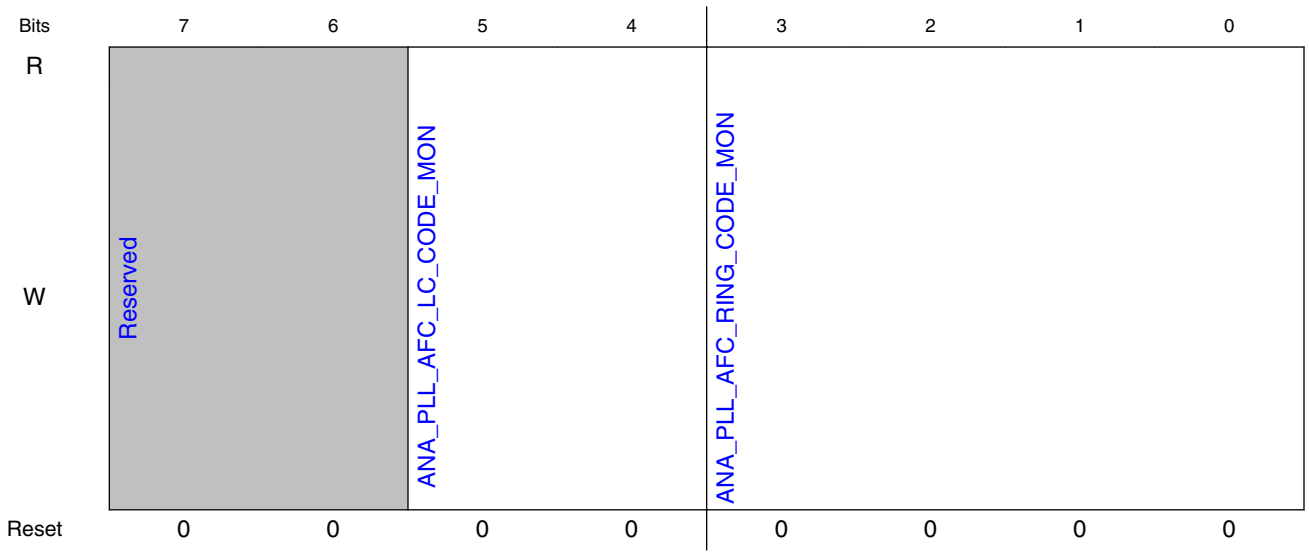
Field	Function
7-0 DTB_SEL	Digital Test Bus (DTB) selection

11.5.3.1.114 (CMN_REG070)

11.5.3.1.114.1 Offset

Register	Offset
CMN_REG070	1C0h

11.5.3.1.114.2 Diagram



11.5.3.1.114.3 Fields

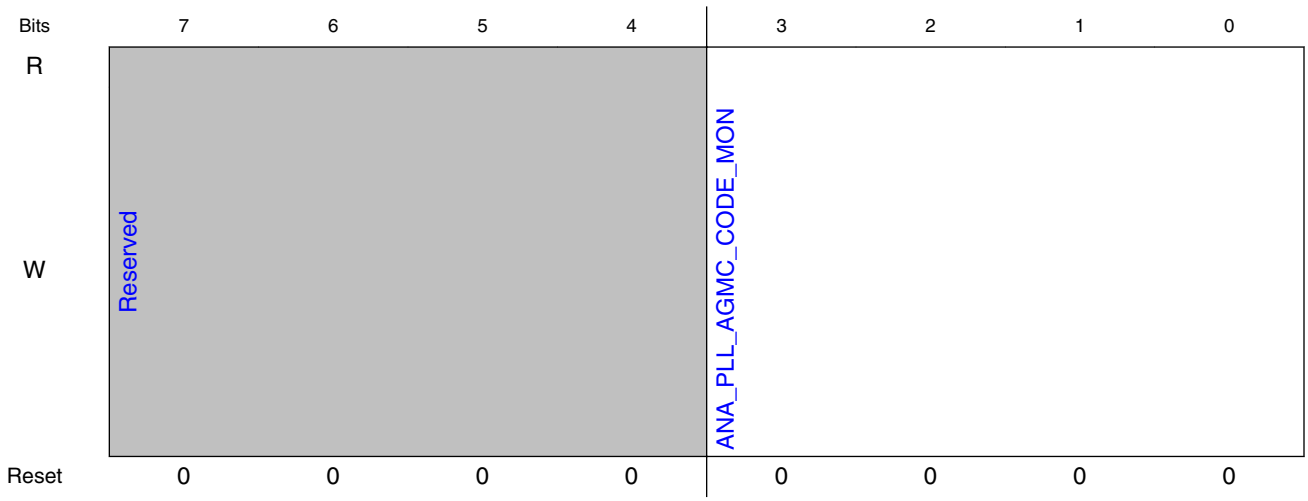
Field	Function
7-6 —	Reserved
5-4 ANA_PLL_AFC_LC_CODE_MON	LC AFC code MSB monitor
3-0 ANA_PLL_AFC_RING_CODE_MON	LC AFC code MSB monitor

11.5.3.1.115 (CMN_REG071)

11.5.3.1.115.1 Offset

Register	Offset
CMN_REG071	1C4h

11.5.3.1.115.2 Diagram



11.5.3.1.115.3 Fields

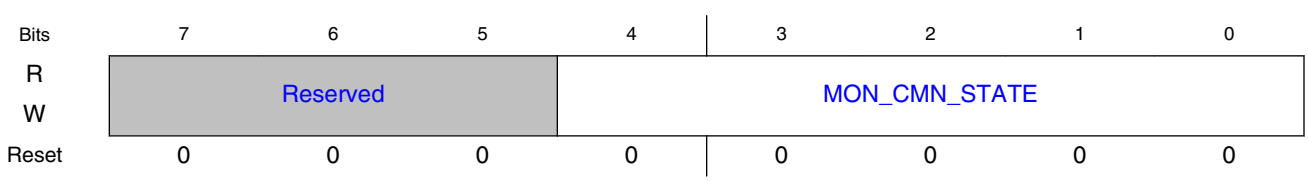
Field	Function
7-4 —	Reserved
3-0 ANA_PLL_AGMC_CODE_MON	LC VCO GM code monitor

11.5.3.1.116 (CMN_REG072)

11.5.3.1.116.1 Offset

Register	Offset
CMN_REG072	1C8h

11.5.3.1.116.2 Diagram



11.5.3.1.116.3 Fields

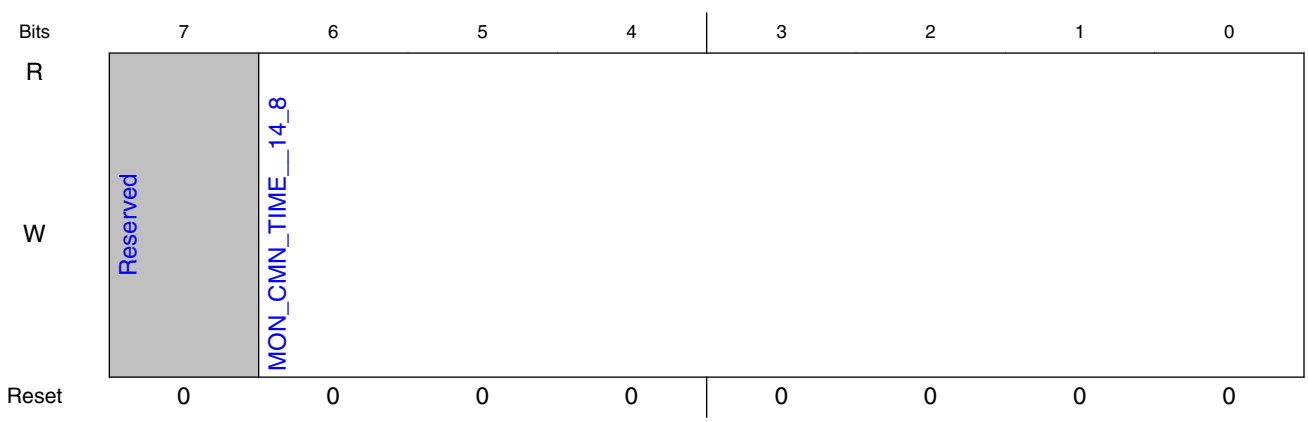
Field	Function
7-5 —	Reserved
4-0 MON_CMN_STATE	CMN state monitor

11.5.3.1.117 (CMN_REG073)

11.5.3.1.117.1 Offset

Register	Offset
CMN_REG073	1CCh

11.5.3.1.117.2 Diagram



11.5.3.1.117.3 Fields

Field	Function
7 —	Reserved
6-0 MON_CMN_TIME__14_8	CMN timer monitor

11.5.3.1.118 (CMN_REG074)

11.5.3.1.118.1 Offset

Register	Offset
CMN_REG074	1D0h

11.5.3.1.118.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	MON_CMN_TIME__7_0							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.118.3 Fields

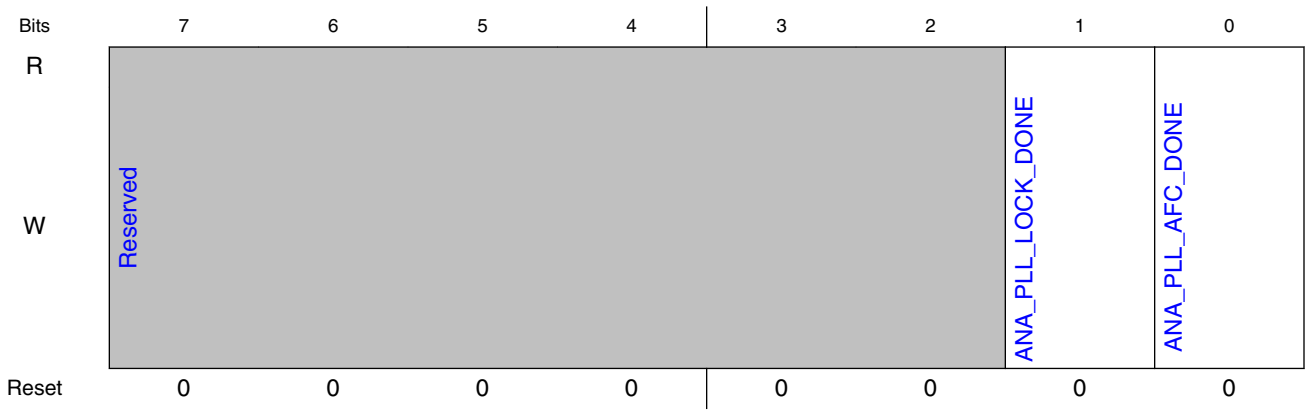
Field	Function
7-0 MON_CMN_TIME__7_0	CMN timer monitor

11.5.3.1.119 (CMN_REG075)

11.5.3.1.119.1 Offset

Register	Offset
CMN_REG075	1D4h

11.5.3.1.119.2 Diagram



11.5.3.1.119.3 Fields

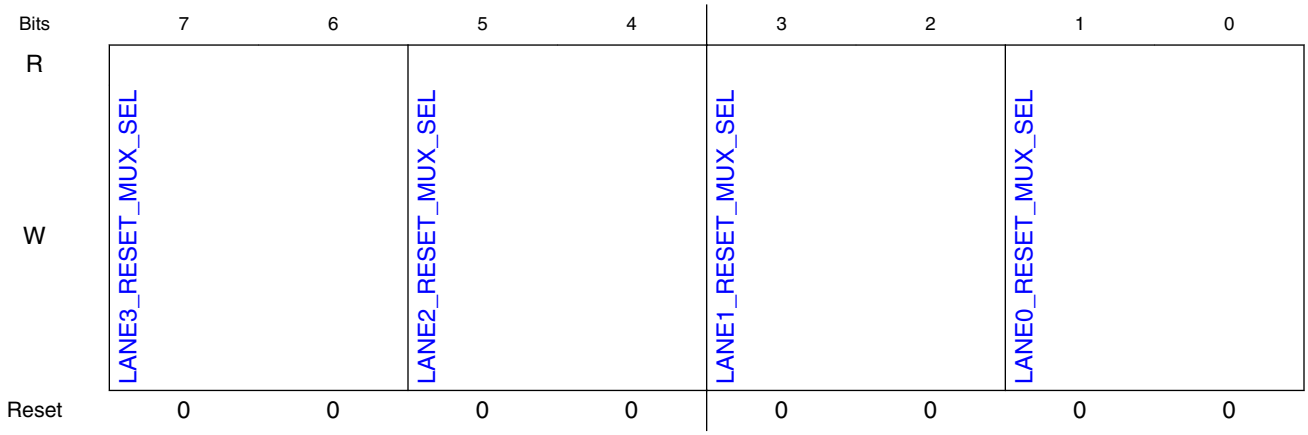
Field	Function
7-2 —	Reserved
1 ANA_PLL_LOCK_DONE	PLL Lock Done
0 ANA_PLL_AFC_DONE	PLL AFC Done

11.5.3.1.120 (CMN_REG076)

11.5.3.1.120.1 Offset

Register	Offset
CMN_REG076	200h

11.5.3.1.120.2 Diagram



11.5.3.1.120.3 Fields

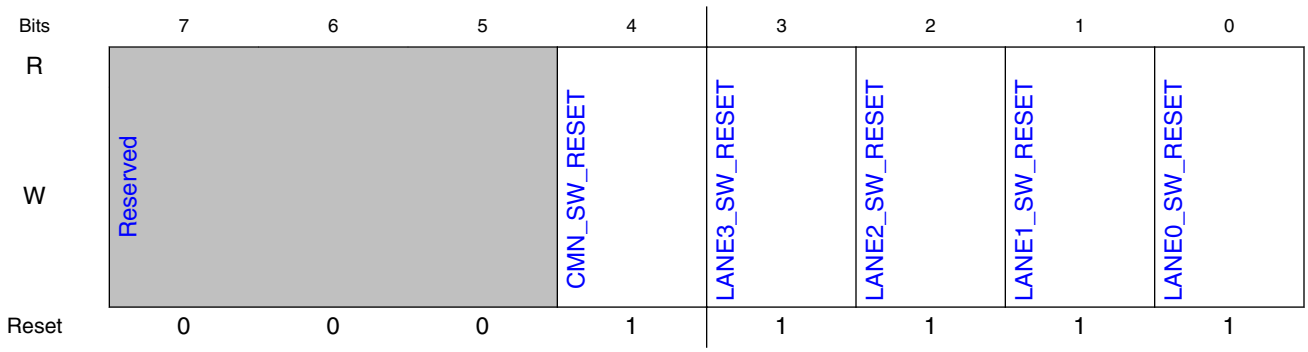
Field	Function
7-6 LANE3_RESET_MUX_SEL	0x0 : Lane3 reset signal from Port 0 0x1 : Lane3 reset signal from Port 1
5-4 LANE2_RESET_MUX_SEL	0x0 : Lane2 reset signal from Port 0 0x1 : Lane2 reset signal from Port 1
3-2 LANE1_RESET_MUX_SEL	0x0 : Lane1 reset signal from Port 0 0x1 : Lane1 reset signal from Port 1
1-0 LANE0_RESET_MUX_SEL	0x0 : Lane0 reset signal from Port 0 0x1 : Lane0 reset signal from Port 1

11.5.3.1.121 (CMN_REG077)

11.5.3.1.121.1 Offset

Register	Offset
CMN_REG077	204h

11.5.3.1.121.2 Diagram



11.5.3.1.121.3 Fields

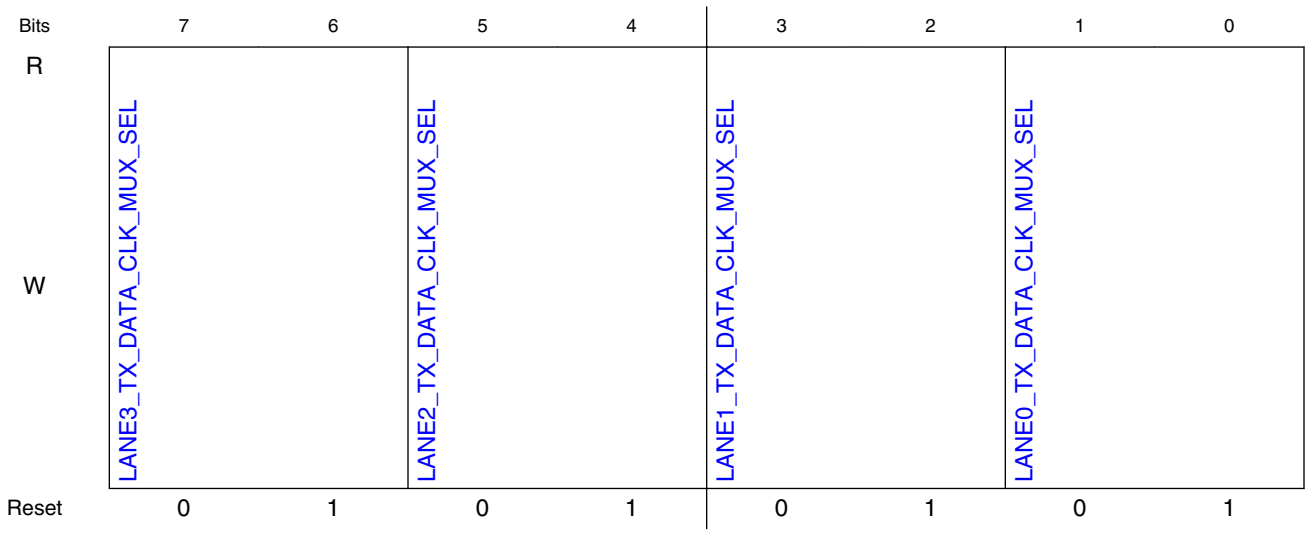
Field	Function
7-5 —	Reserved
4 CMN_SW_RESET	0x0 : cmn reset 0x1 : cmn reset release
3 LANE3_SW_RESET	0x0 : Lane3 reset 0x1 : Lane3 reset release
2 LANE2_SW_RESET	0x0 : Lane2 reset 0x1 : Lane2 reset release
1 LANE1_SW_RESET	0x0 : Lane1 reset 0x1 : Lane1 reset release
0 LANE0_SW_RESET	0x0 : Lane0 reset 0x1 : Lane0 reset release

11.5.3.1.122 (CMN_REG078)

11.5.3.1.122.1 Offset

Register	Offset
CMN_REG078	208h

11.5.3.1.122.2 Diagram



11.5.3.1.122.3 Fields

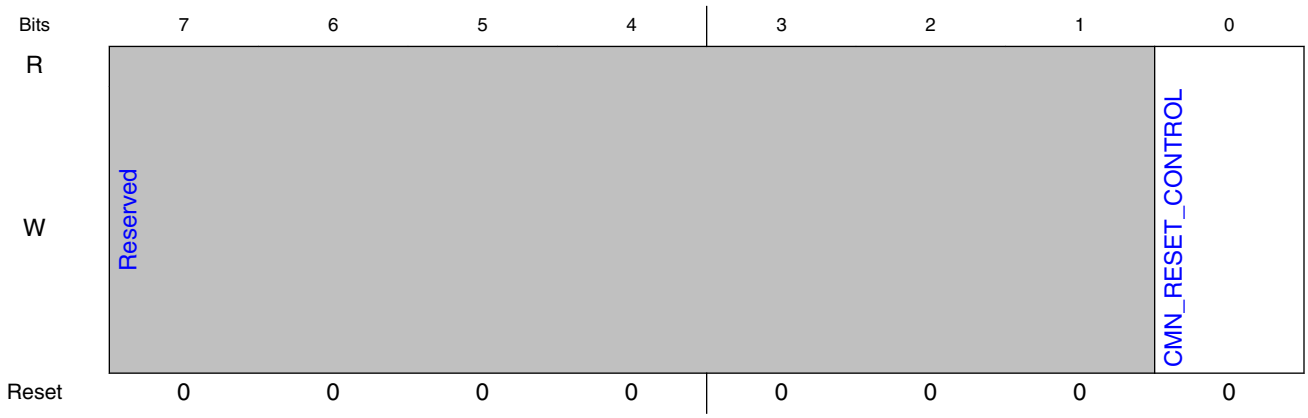
Field	Function
7-6 LANE3_TX_DATA_CLK_MUX_SEL	0x0 : tx data clock from lane0 0x1 : tx data clock from lane1 0x2 : tx data clock from lane2 0x3 : tx data clock from lane3
5-4 LANE2_TX_DATA_CLK_MUX_SEL	
3-2 LANE1_TX_DATA_CLK_MUX_SEL	
1-0 LANE0_TX_DATA_CLK_MUX_SEL	

11.5.3.1.123 (CMN_REG079)

11.5.3.1.123.1 Offset

Register	Offset
CMN_REG079	20Ch

11.5.3.1.123.2 Diagram



11.5.3.1.123.3 Fields

Field	Function
7-1 —	Reserved
0 CMN_RESET_CONTROL	

11.5.3.1.124 (CMN_REG080)

11.5.3.1.124.1 Offset

Register	Offset
CMN_REG080	210h

11.5.3.1.124.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	RATE_CHANGE_DELAY							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.124.3 Fields

Field	Function
7-0 RATE_CHANGE_DELAY	

11.5.3.1.125 (CMN_REG081)

11.5.3.1.125.1 Offset

Register	Offset
CMN_REG081	214h

11.5.3.1.125.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	RX_EFOM_ERROR_TH_7_0							
Reset	0	0	0	1	1	0	0	0

11.5.3.1.125.3 Fields

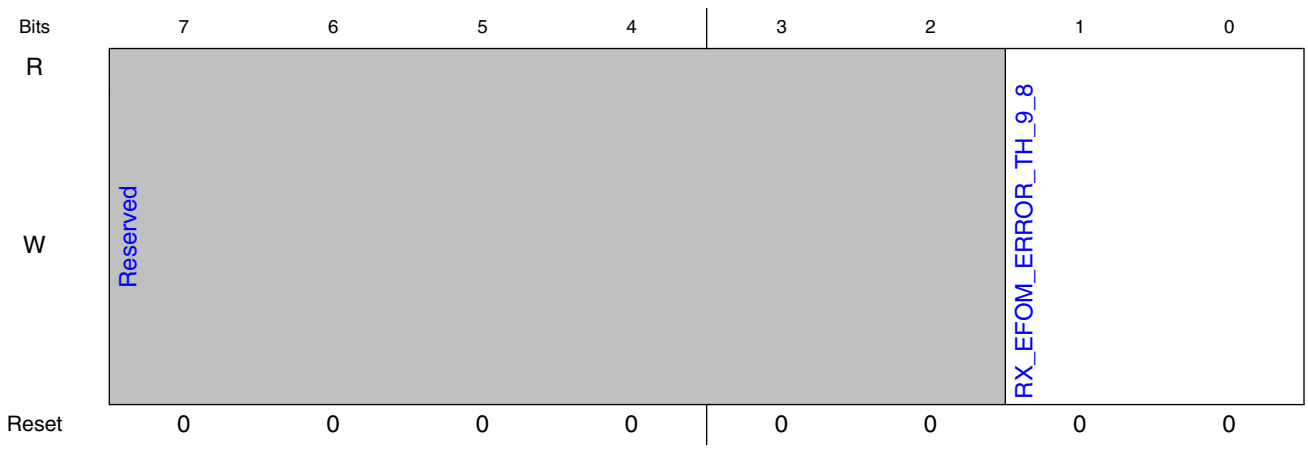
Field	Function
7-0 RX_EFOM_ERROR_TH_7_0	

11.5.3.1.126 (CMN_REG082)

11.5.3.1.126.1 Offset

Register	Offset
CMN_REG082	218h

11.5.3.1.126.2 Diagram



11.5.3.1.126.3 Fields

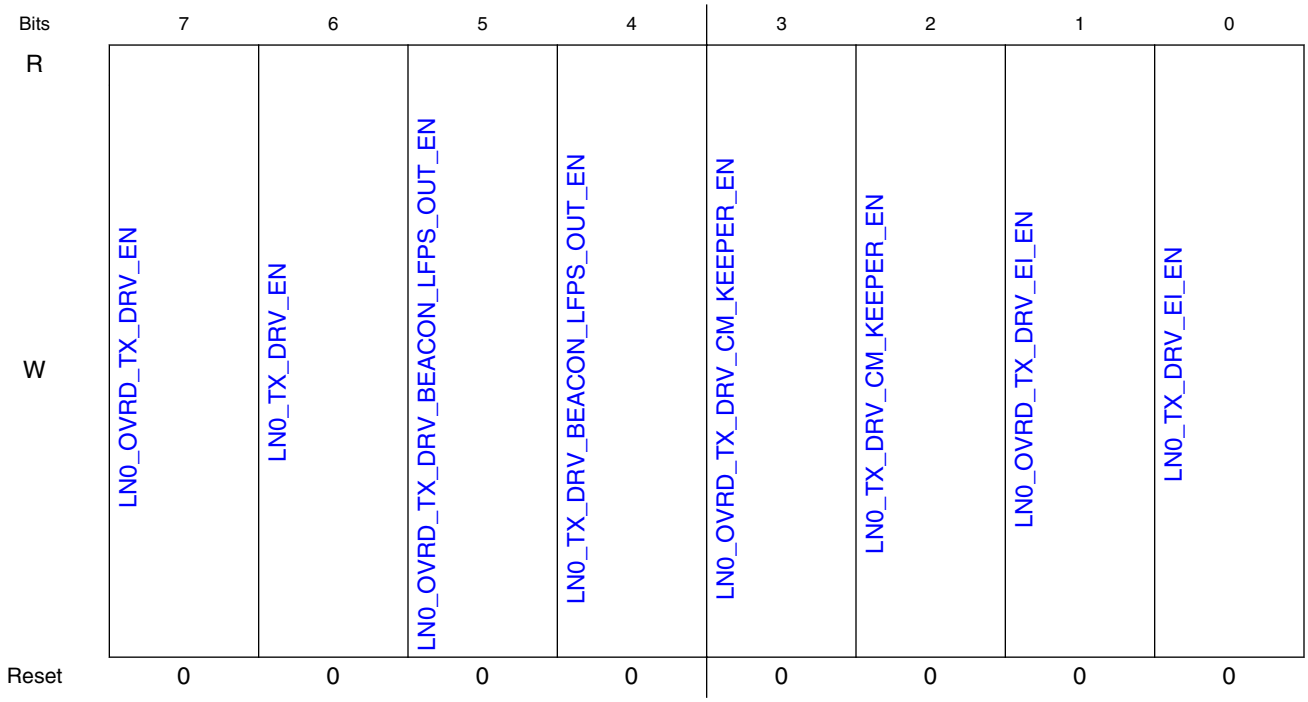
Field	Function
7-2 —	Reserved
1-0 RX_EFOM_ERROR_TH_9_8	

11.5.3.1.127 (TRSV_REG000)

11.5.3.1.127.1 Offset

Register	Offset
TRSV_REG000	400h

11.5.3.1.127.2 Diagram



11.5.3.1.127.3 Fields

Field	Function
7 LN0_OVRD_TX_DRV_EN	Override enable for tx_drv_en
6 LN0_TX_DRV_EN	TX driver enable 0: Disable, 1: Enable
5 LN0_OVRD_TX_DRV_BEACON_LFPS_OUT_EN	Override enable for tx_drv_beacon_lfps_out_en
4 LN0_TX_DRV_BEACON_LFPS_OUT_EN	TX beacon or LFPS enable 0: Disable, 1: Enable
3 LN0_OVRD_TX_DRV_CM_KEEPER_EN	Override enable for tx_drv_cm_keeper_en
2	TX driver common-mode keep enable

Table continues on the next page...

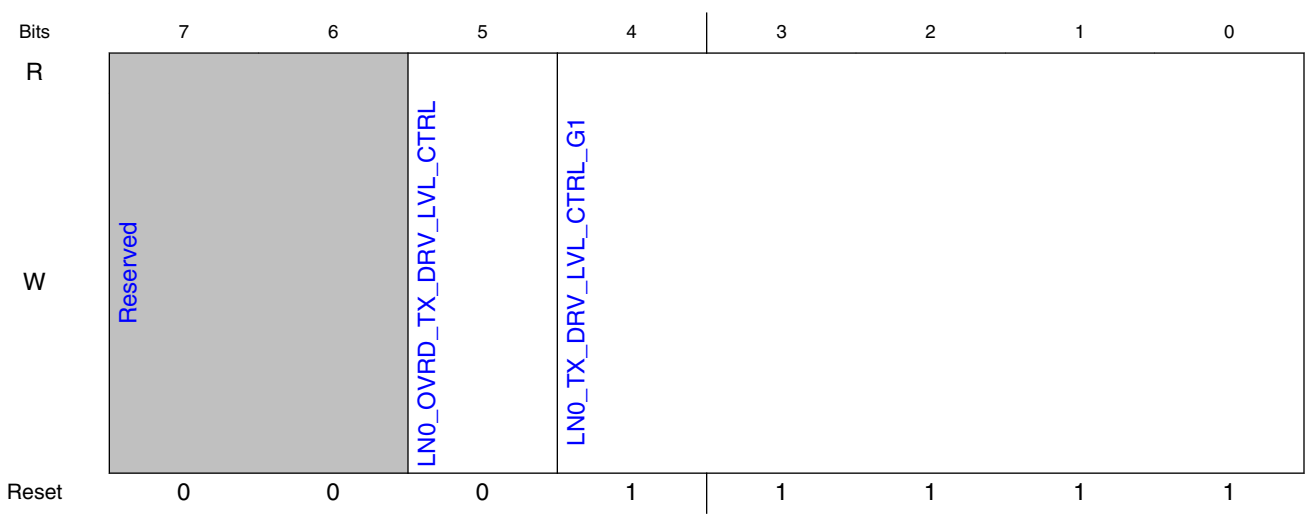
Field	Function
LN0_TX_DRV_CM_KEEPER_EN	0: Disable, 1: Common-mode voltage keep
1 LN0_OVRD_TX_DRV_EI_EN	Override enable for tx_drv_ei_en
0 LN0_TX_DRV_EI_EN	TX driver electrical-idle state enable 0: Disable, 1: Electrical-idle enable

11.5.3.1.128 (TRSV_REG001)

11.5.3.1.128.1 Offset

Register	Offset
TRSV_REG001	404h

11.5.3.1.128.2 Diagram



11.5.3.1.128.3 Fields

Field	Function
7-6 —	Reserved

Table continues on the next page...

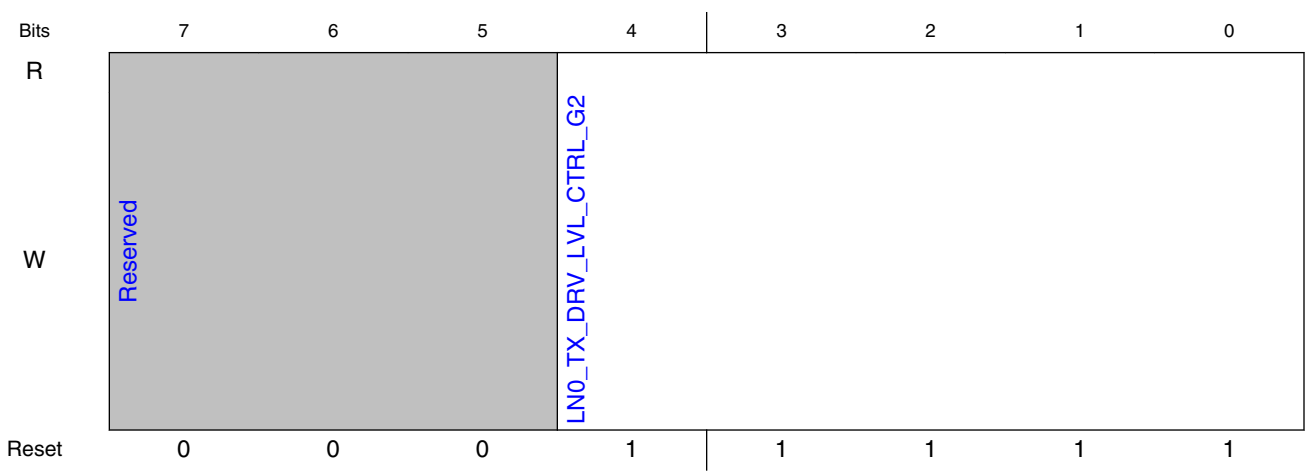
Field	Function
5 LN0_OVRD_TX_DRV_LVL_CTRL	Override enable for tx_drv_lvl_ctrl_g1
4-0 LN0_TX_DRV_LVL_CTRL_G1	[GEN1] TX driver main-tap level 1111: max main-tap level … 0000: min main-tap level Others: N/A

11.5.3.1.129 (TRSV_REG002)

11.5.3.1.129.1 Offset

Register	Offset
TRSV_REG002	408h

11.5.3.1.129.2 Diagram



11.5.3.1.129.3 Fields

Field	Function
7-5 —	Reserved

Table continues on the next page...

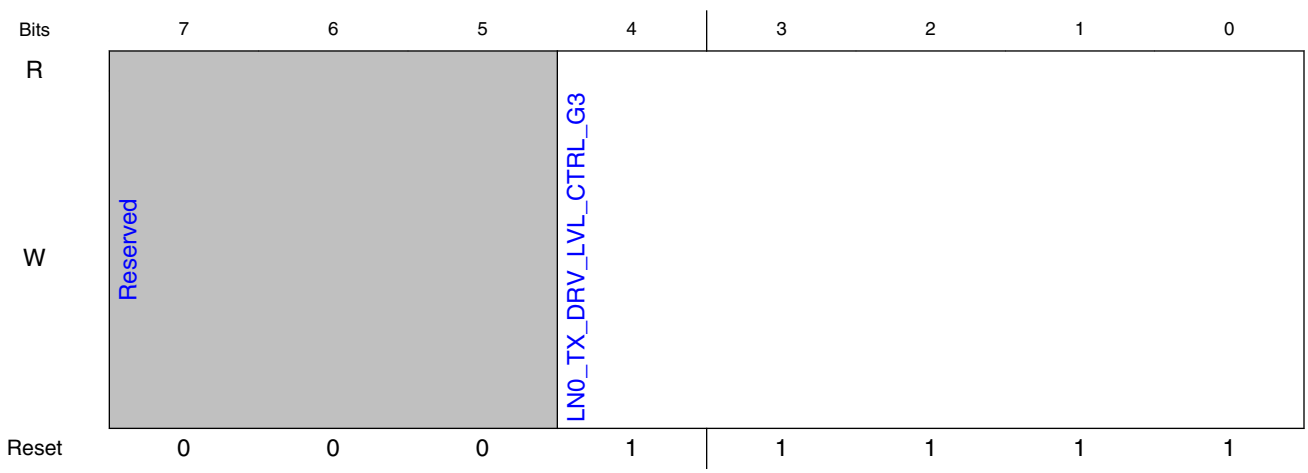
Field	Function
4-0 LN0_TX_DRV_LVL_CTRL_G2	[GEN2]

11.5.3.1.130 (TRSV_REG003)

11.5.3.1.130.1 Offset

Register	Offset
TRSV_REG003	40Ch

11.5.3.1.130.2 Diagram



11.5.3.1.130.3 Fields

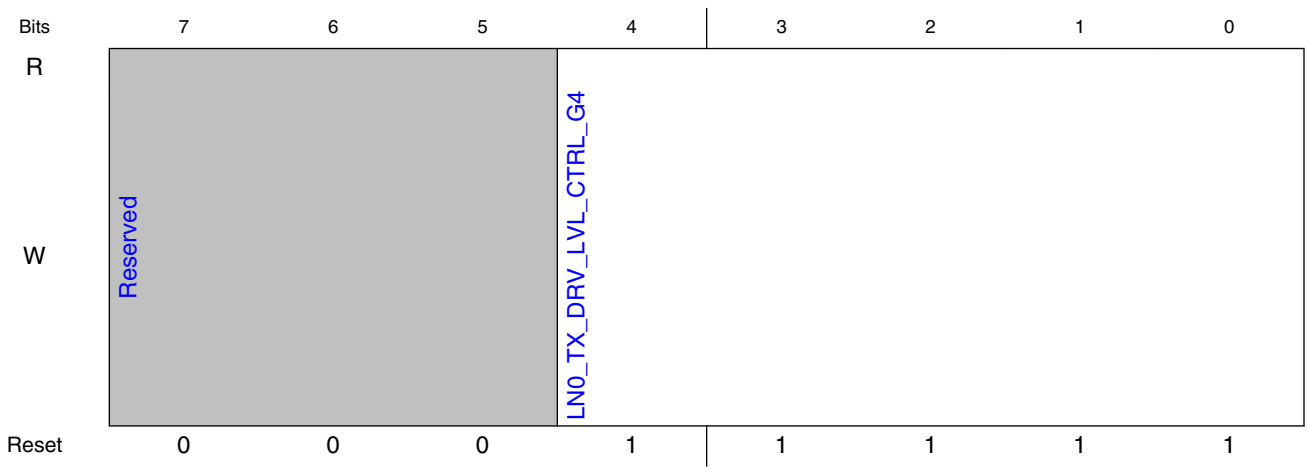
Field	Function
7-5 —	Reserved
4-0 LN0_TX_DRV_LVL_CTRL_G3	[GEN3]

11.5.3.1.131 (TRSV_REG004)

11.5.3.1.131.1 Offset

Register	Offset
TRSV_REG004	410h

11.5.3.1.131.2 Diagram



11.5.3.1.131.3 Fields

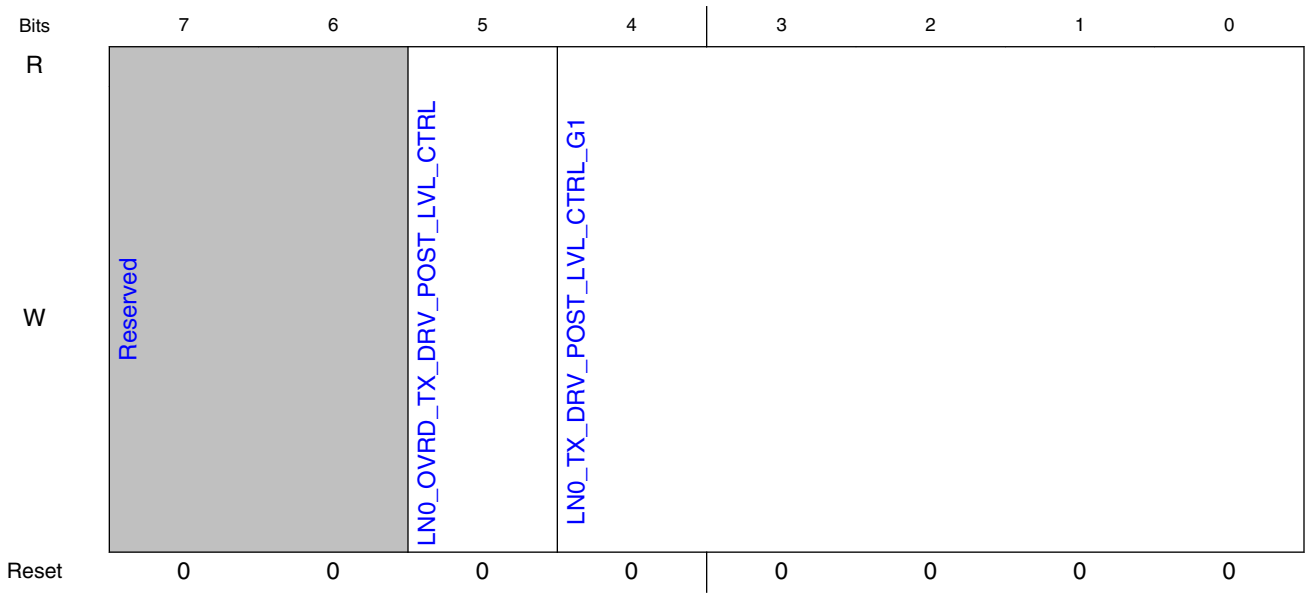
Field	Function
7-5 —	Reserved
4-0 LN0_TX_DRV_LVL_CTRL_G4	[GEN4]

11.5.3.1.132 (TRSV_REG005)

11.5.3.1.132.1 Offset

Register	Offset
TRSV_REG005	414h

11.5.3.1.132.2 Diagram



11.5.3.1.132.3 Fields

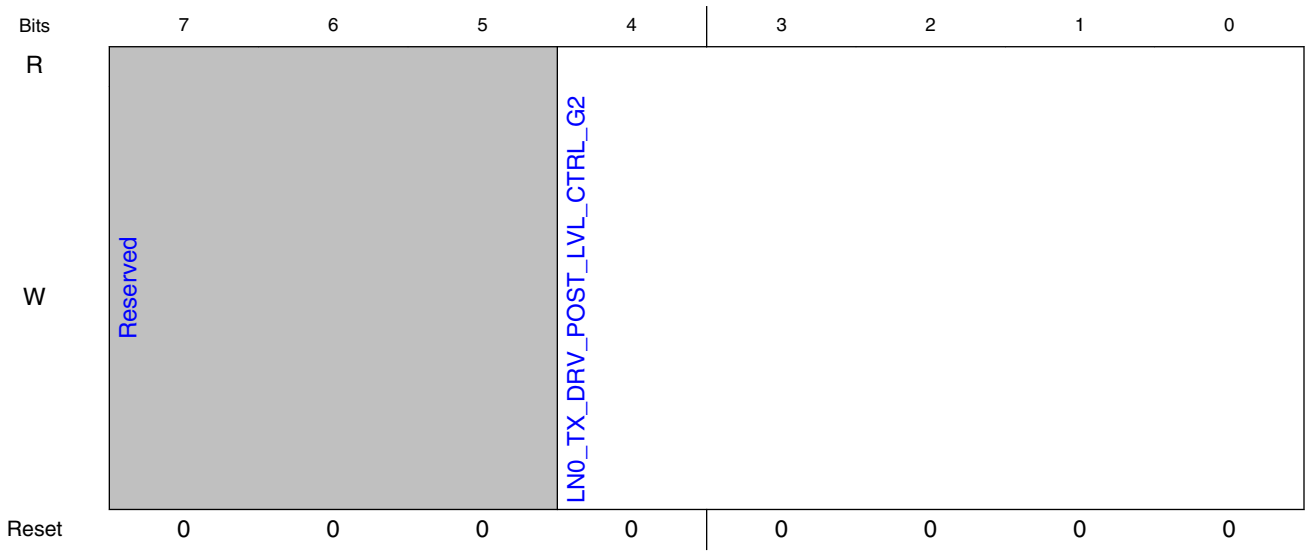
Field	Function
7-6 —	Reserved
5 LN0_OVRD_TX_DRV_POST_LVL_CTRL	Override enable for tx_drv_post_lvl_ctrl_g1
4-0 LN0_TX_DRV_POST_LVL_CTRL_G1	[GEN1] TX driver de-emphasis level 00000: min de-emphasis level … 10100: max de-emphasis level Others: N/A

11.5.3.1.133 (TRSV_REG006)

11.5.3.1.133.1 Offset

Register	Offset
TRSV_REG006	418h

11.5.3.1.133.2 Diagram



11.5.3.1.133.3 Fields

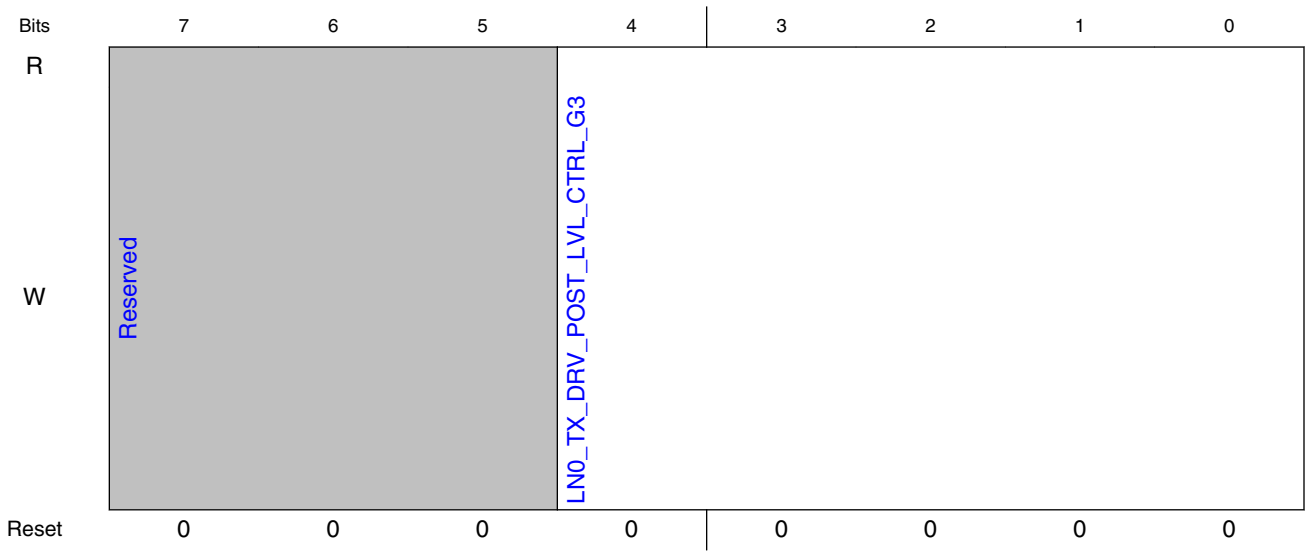
Field	Function
7-5 —	Reserved
4-0 LN0_TX_DRV_POST_LVL_CTRL_G2	[GEN2]

11.5.3.1.134 (TRSV_REG007)

11.5.3.1.134.1 Offset

Register	Offset
TRSV_REG007	41Ch

11.5.3.1.134.2 Diagram



11.5.3.1.134.3 Fields

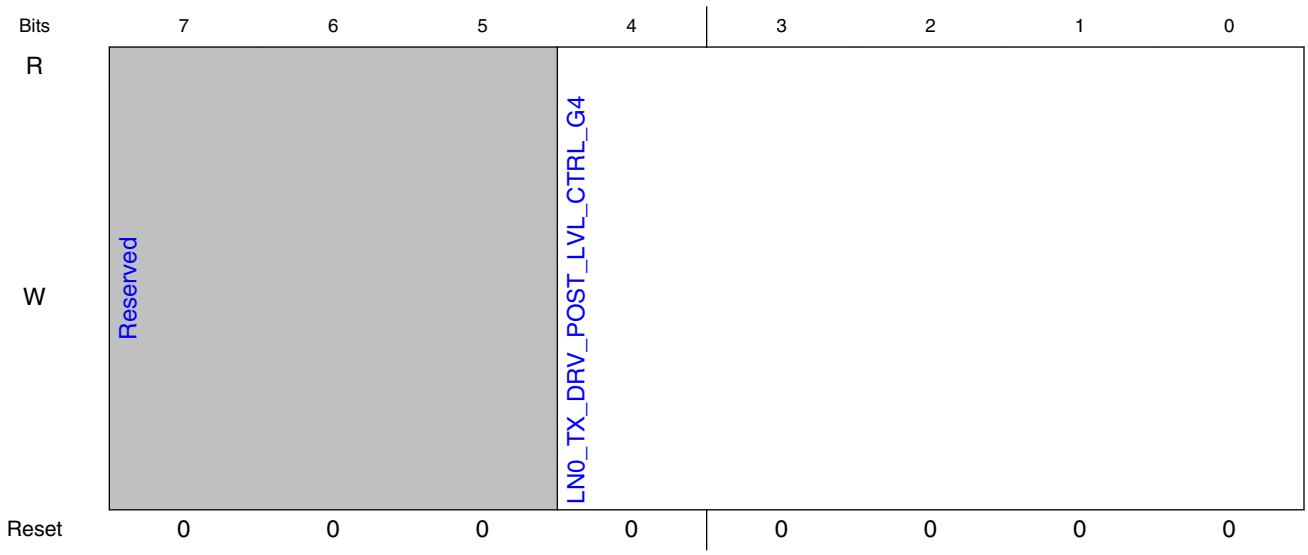
Field	Function
7-5 —	Reserved
4-0 LN0_TX_DRV_POST_LVL_CTRL_G3	[GEN3]

11.5.3.1.135 (TRSV_REG008)

11.5.3.1.135.1 Offset

Register	Offset
TRSV_REG008	420h

11.5.3.1.135.2 Diagram



11.5.3.1.135.3 Fields

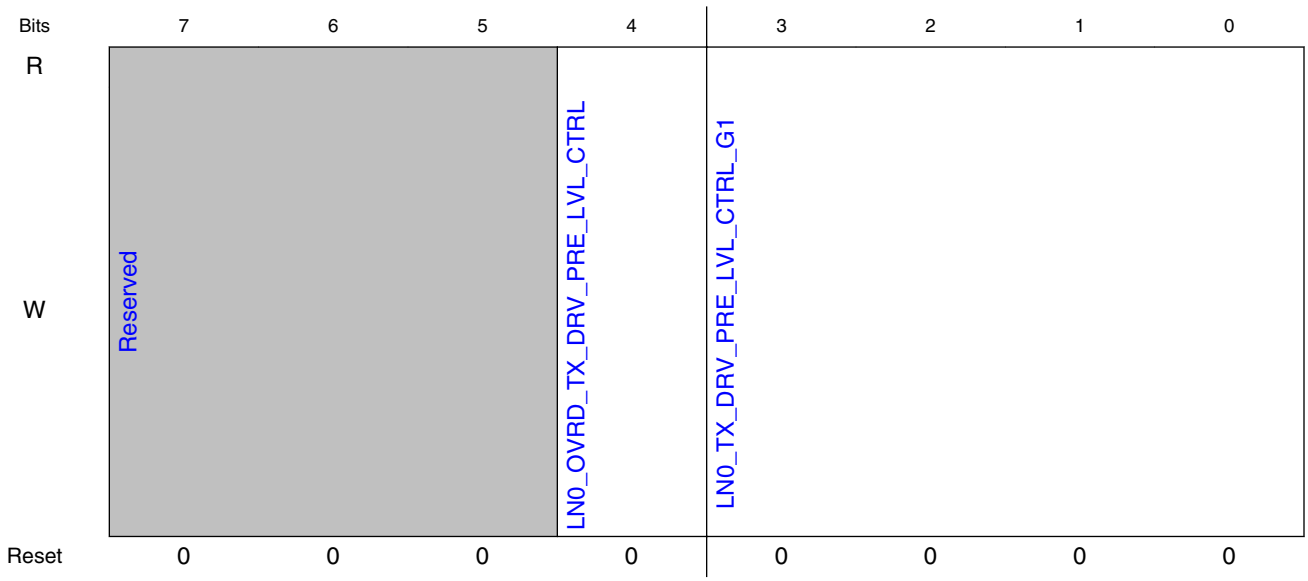
Field	Function
7-5 —	Reserved
4-0 LN0_TX_DRV_POST_LVL_CTRL_G4	[GEN4]

11.5.3.1.136 (TRSV_REG009)

11.5.3.1.136.1 Offset

Register	Offset
TRSV_REG009	424h

11.5.3.1.136.2 Diagram



11.5.3.1.136.3 Fields

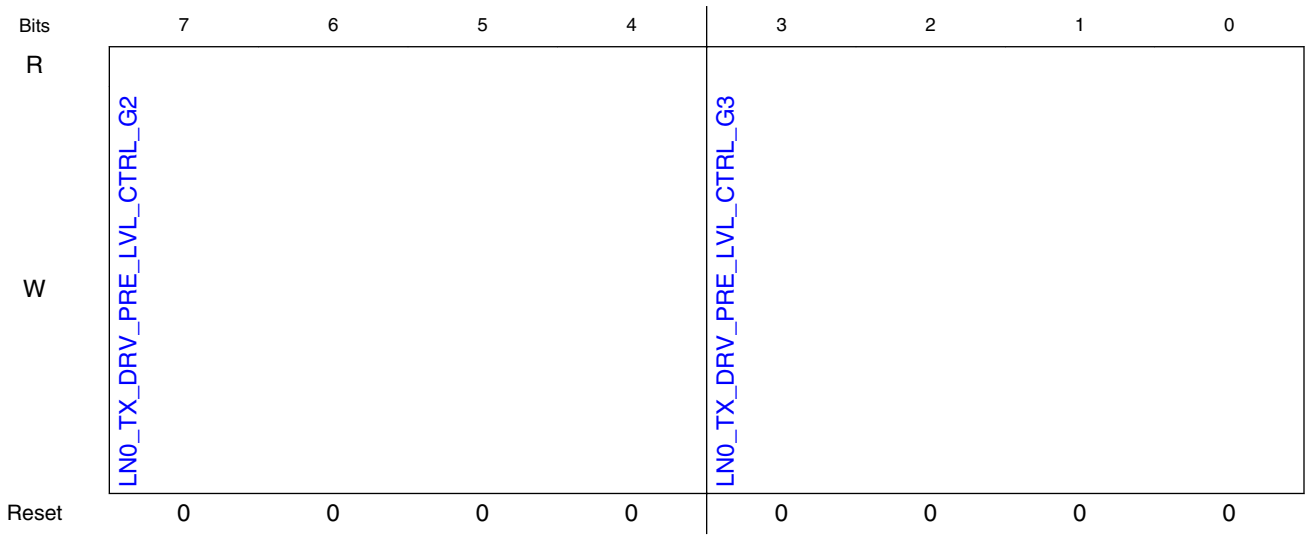
Field	Function
7-5 —	Reserved
4 LN0_OVRD_TX_DRV_PRE_LVL_CTRL	Override enable for tx_drv_pre_lvl_ctrl_g1
3-0 LN0_TX_DRV_PRE_LVL_CTRL_G1	[GEN1] TX driver pre-shoot level 0000: min pre-shoot boosting … 1011: max pre-shoot boosting Others: N/A

11.5.3.1.137 (TRSV_REG00A)

11.5.3.1.137.1 Offset

Register	Offset
TRSV_REG00A	428h

11.5.3.1.137.2 Diagram



11.5.3.1.137.3 Fields

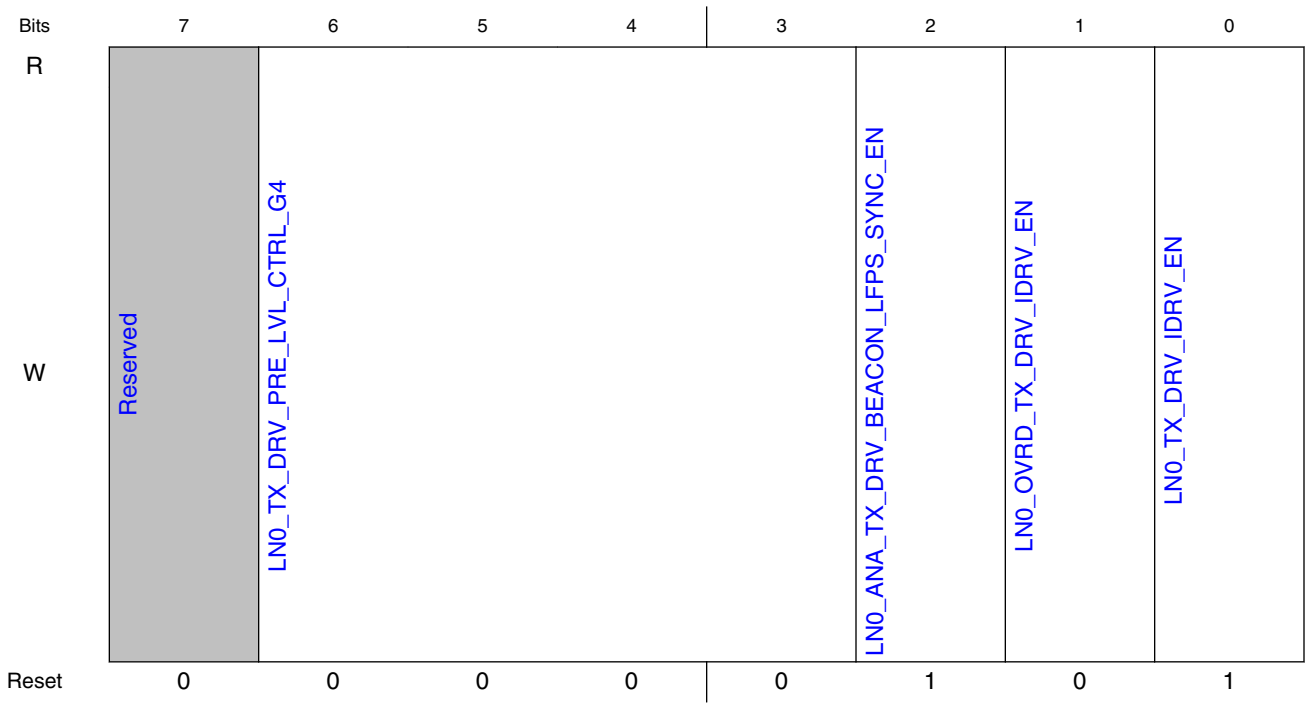
Field	Function
7-4 LN0_TX_DRV_PRE_LVL_CTRL_G2	[GEN2]
3-0 LN0_TX_DRV_PRE_LVL_CTRL_G3	[GEN3]

11.5.3.1.138 (TRSV_REG00B)

11.5.3.1.138.1 Offset

Register	Offset
TRSV_REG00B	42Ch

11.5.3.1.138.2 Diagram



11.5.3.1.138.3 Fields

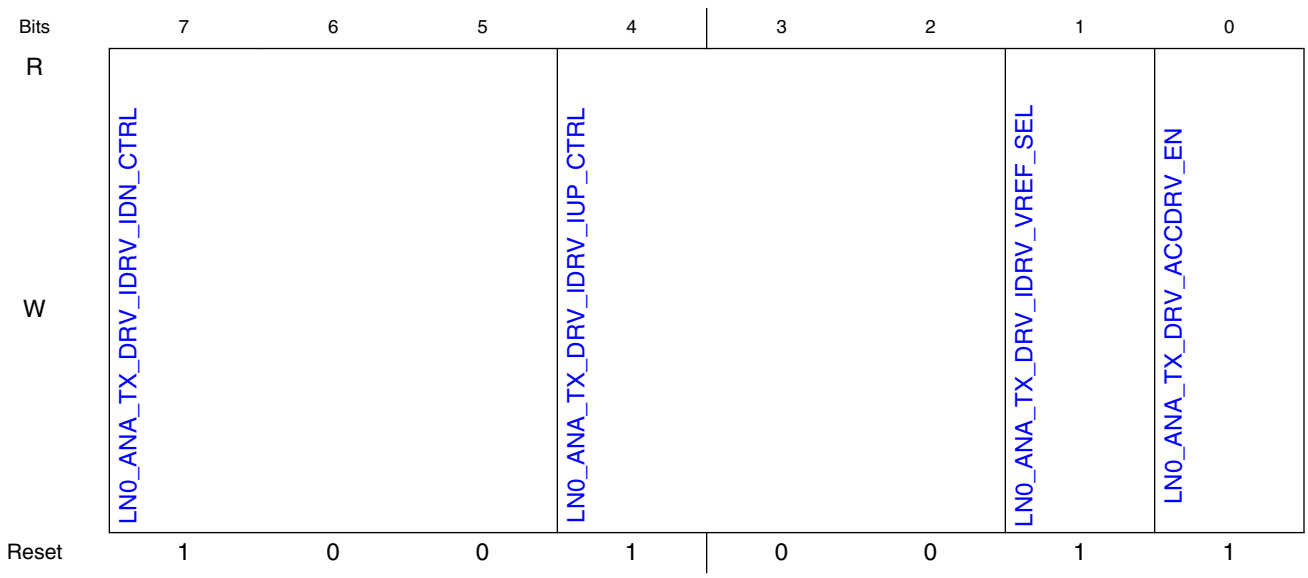
Field	Function
7 —	Reserved
6-3 LN0_TX_DRV_PRE_LVL_CTRL_G4	[GEN4]
2 LN0_ANA_TX_DRV_BEACON_LFPS_SYNC_EN	TX LFPS/BEACON synchronization enable If enabled, LFPS/BEACON output is self-synchronized so that waveform is not affedted by LFPS/beacon enable timing. 0: Disable (async), 1: Enalbe (sync)
1 LN0_OVRD_TX_DRV_IDRV_EN	Override enable for tx_drv_idrv_en
0 LN0_TX_DRV_IDRV_EN	TX current-driver enable 0: disable, 1: enable

11.5.3.1.139 (TRSV_REG00C)

11.5.3.1.139.1 Offset

Register	Offset
TRSV_REG00C	430h

11.5.3.1.139.2 Diagram



11.5.3.1.139.3 Fields

Field	Function
7-5 LN0_ANA_TX_DRV_IDRV_IDN_CTRL	TX current driver pmos current control 000: Min … 111: Max
4-2 LN0_ANA_TX_DRV_IDRV_IUP_CTRL	TX current driver pmos current control 000: Min … 111: Max
1 LN0_ANA_TX_DRV_IDRV_VREF_SEL	TX current driver reference selection 1: Fixed reference, 0: Unity-gain buffer(Default)

Table continues on the next page...

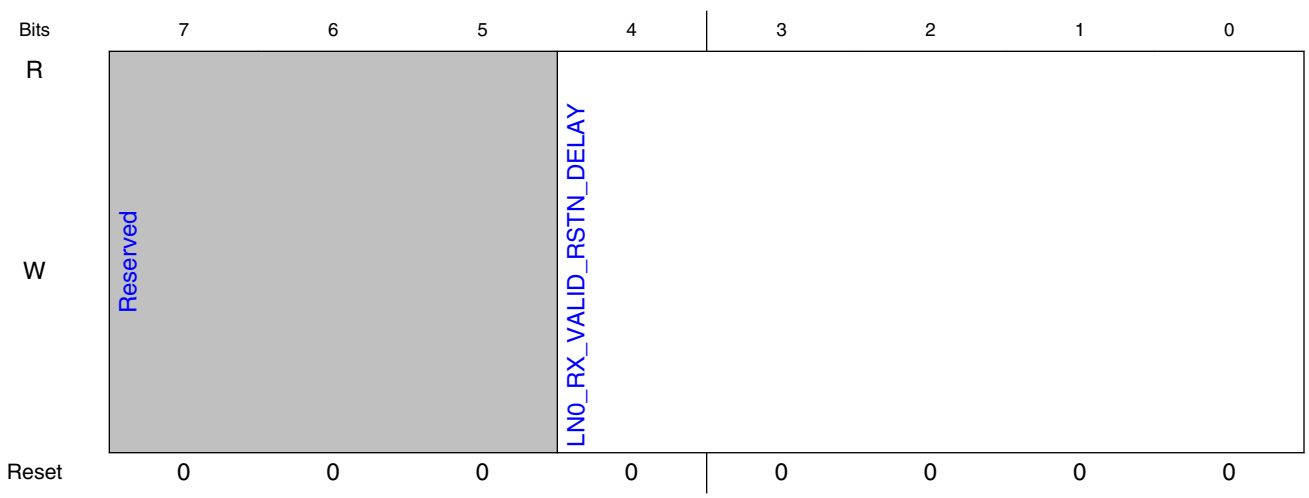
Field	Function
0 LN0_ANA_TX_DRV_ACCDRV_EN	Enable of Cap. Peaking block. 0: Disable, 1: Enable

11.5.3.1.140 (TRSV_REG00D)

11.5.3.1.140.1 Offset

Register	Offset
TRSV_REG00D	434h

11.5.3.1.140.2 Diagram



11.5.3.1.140.3 Fields

Field	Function
7-5 —	Reserved
4-0 LN0_RX_VALID_RSTN_DELAY	

11.5.3.1.141 (TRSV_REG00E)

11.5.3.1.141.1 Offset

Register	Offset
TRSV_REG00E	438h

11.5.3.1.141.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_TX_DRV_EI_EN_DELAY_SEL_G1		LN0_TX_DRV_EI_EN_DELAY_SEL_G2		LN0_TX_DRV_EI_EN_DELAY_SEL_G3		LN0_TX_DRV_EI_EN_DELAY_SEL_G4	
W								
Reset	0	1	0	1	0	1	0	1

11.5.3.1.141.3 Fields

Field	Function
7-6 LN0_TX_DRV_EI_EN_DELAY_SEL_G1	[GEN1] TX EI enable latency control Time delay from EI enable from PCS to real 00: , 01: , 10: , 11:
5-4 LN0_TX_DRV_EI_EN_DELAY_SEL_G2	[GEN2]
3-2 LN0_TX_DRV_EI_EN_DELAY_SEL_G3	[GEN3]
1-0	[GEN4]

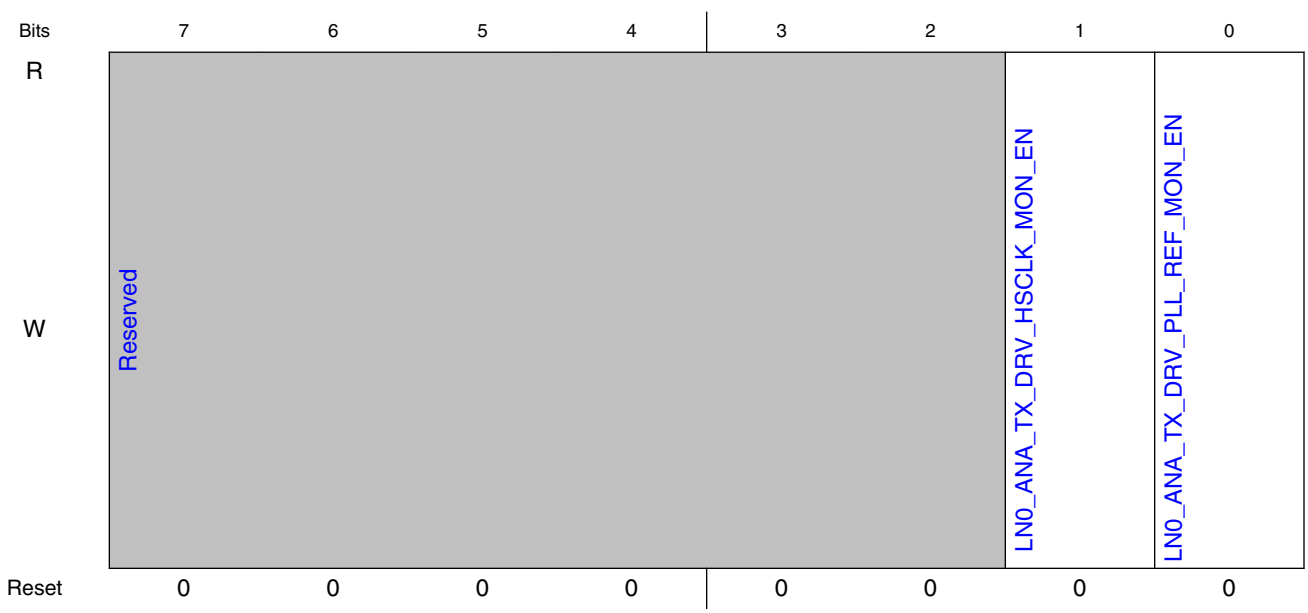
Field	Function
LN0_TX_DRV_EI_EN_DELAY_SEL_G4	

11.5.3.1.142 (TRSV_REG00F)

11.5.3.1.142.1 Offset

Register	Offset
TRSV_REG00F	43Ch

11.5.3.1.142.2 Diagram



11.5.3.1.142.3 Fields

Field	Function
7-2 —	Reserved
1 LN0_ANA_TX_DRV_HSCLK_MON_EN	Enable of high-speed clock monitor through Tx driver 0: Disable, 1: Enable

Table continues on the next page...

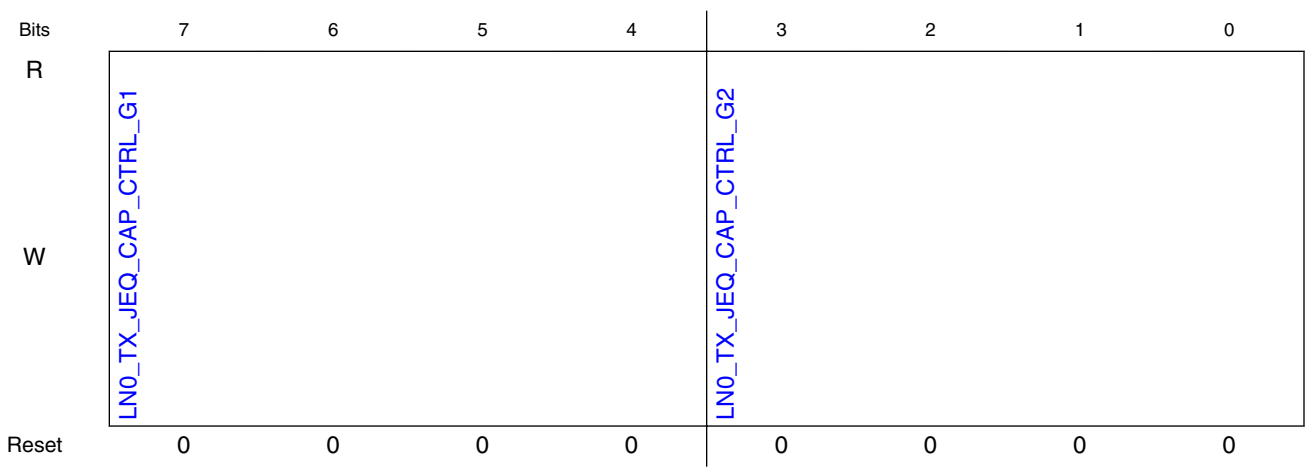
Field	Function
0 LN0_ANA_TX_DRV_PLL_REF_MON_EN	Enable of PLL reference clock monitor through Tx driver 0: Disable, 1: Enable

11.5.3.1.143 (TRSV_REG010)

11.5.3.1.143.1 Offset

Register	Offset
TRSV_REG010	440h

11.5.3.1.143.2 Diagram



11.5.3.1.143.3 Fields

Field	Function
7-4 LN0_TX_JEQ_CAP_CTRL_G1	[GEN1] TX jitter EQ loding capacitance control in thermomether 0000: default 0001: 1 unit cap loading 0011: 2 unit cap loading 0111: 3 unit cap loading 1111: 4 unit cap loading
3-0	[GEN2]

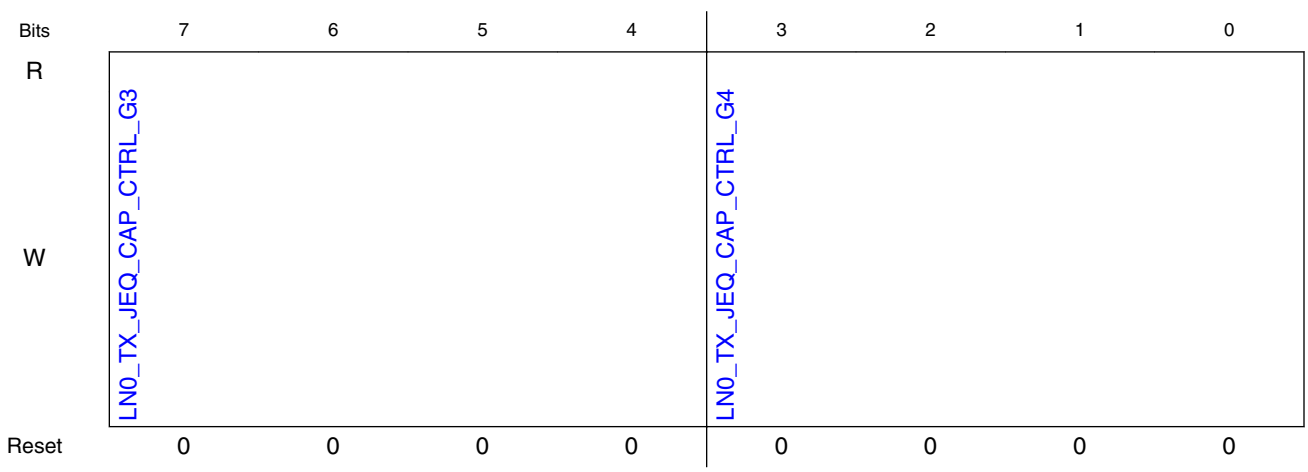
Field	Function
LN0_TX_JEQ_CAP_CTRL_G2	

11.5.3.1.144 (TRSV_REG011)

11.5.3.1.144.1 Offset

Register	Offset
TRSV_REG011	444h

11.5.3.1.144.2 Diagram



11.5.3.1.144.3 Fields

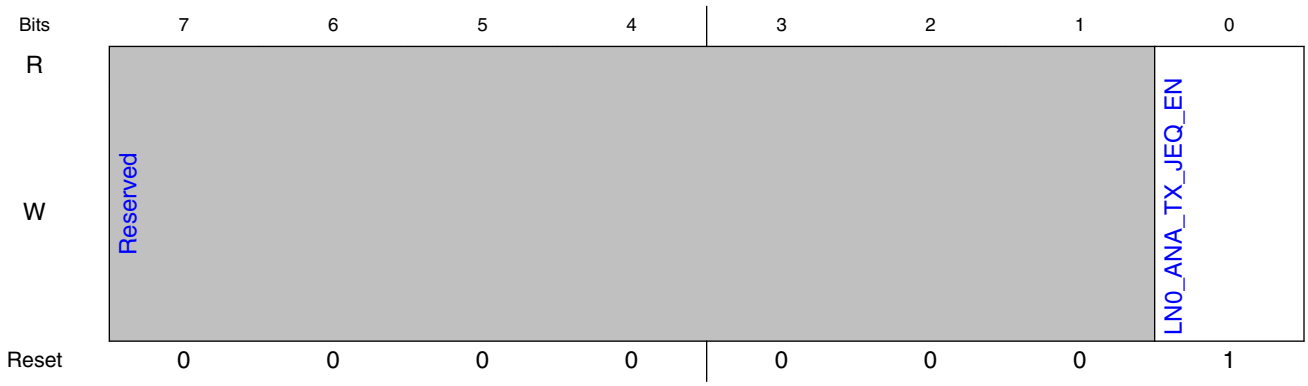
Field	Function
7-4 LN0_TX_JEQ_CAP_CTRL_G3	[GEN3]
3-0 LN0_TX_JEQ_CAP_CTRL_G4	[GEN4]

11.5.3.1.145 (TRSV_REG012)

11.5.3.1.145.1 Offset

Register	Offset
TRSV_REG012	448h

11.5.3.1.145.2 Diagram



11.5.3.1.145.3 Fields

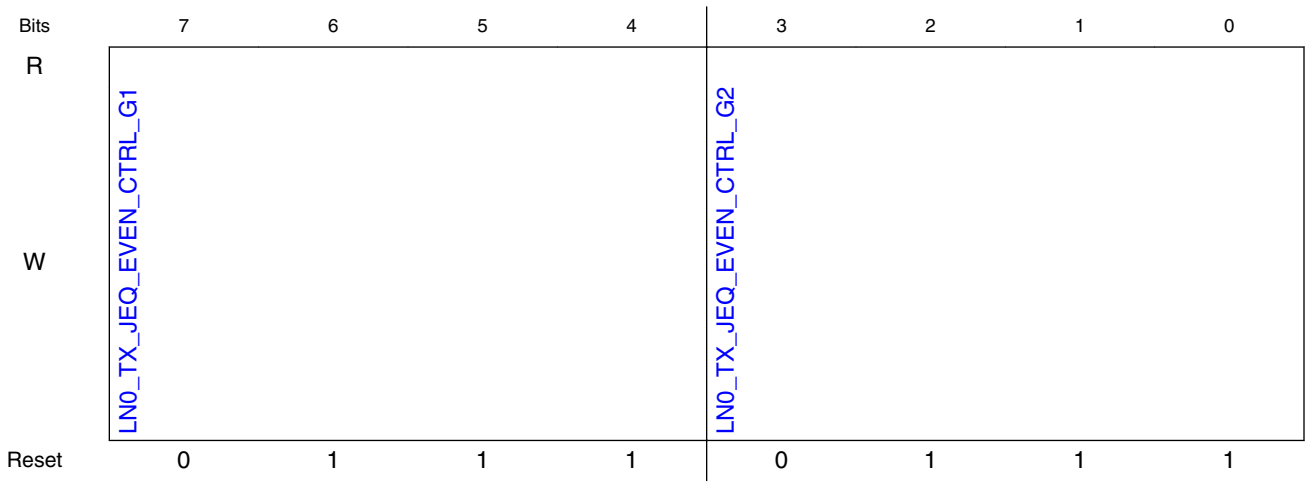
Field	Function
7-1 —	Reserved
0 LN0_ANA_TX_JEQ_EN	TX jitter EQ enable 0: disable 1: enable

11.5.3.1.146 (TRSV_REG013)

11.5.3.1.146.1 Offset

Register	Offset
TRSV_REG013	44Ch

11.5.3.1.146.2 Diagram



11.5.3.1.146.3 Fields

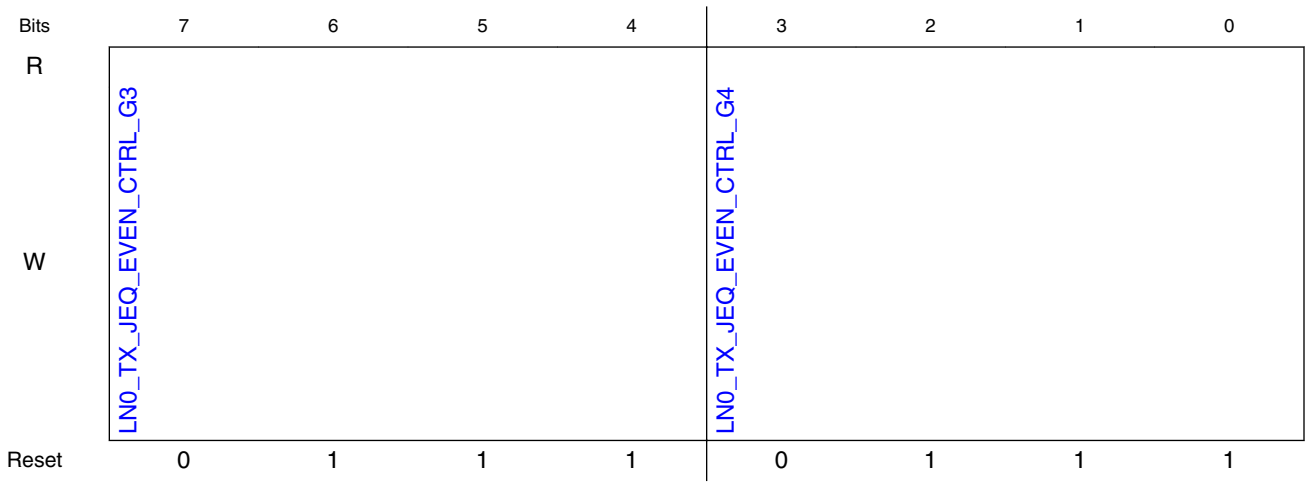
Field	Function
7-4 LN0_TX_JEQ_EVEN_CTRL_G1	[GEN1] TX jitter EQ driver (even) strength control 0000: 0 legs on (jitter EQ driver off) 0100: 2 legs on 1000: 4 legs on 1111: 8 legs on
3-0 LN0_TX_JEQ_EVEN_CTRL_G2	[GEN2]

11.5.3.1.147 (TRSV_REG014)

11.5.3.1.147.1 Offset

Register	Offset
TRSV_REG014	450h

11.5.3.1.147.2 Diagram



11.5.3.1.147.3 Fields

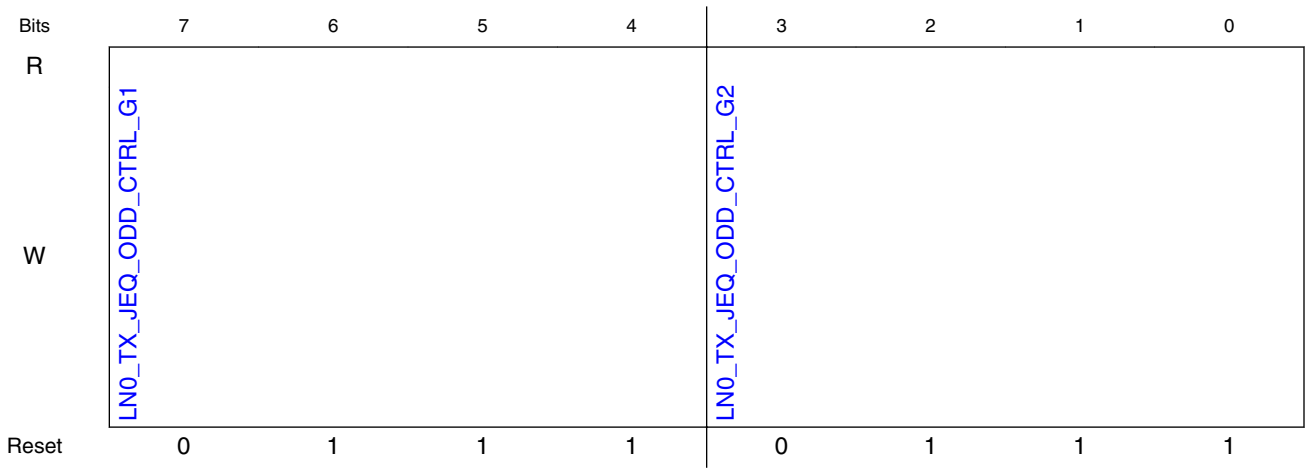
Field	Function
7-4 LN0_TX_JEQ_EVEN_CTRL_G3	[GEN3]
3-0 LN0_TX_JEQ_EVEN_CTRL_G4	[GEN4]

11.5.3.1.148 (TRSV_REG015)

11.5.3.1.148.1 Offset

Register	Offset
TRSV_REG015	454h

11.5.3.1.148.2 Diagram



11.5.3.1.148.3 Fields

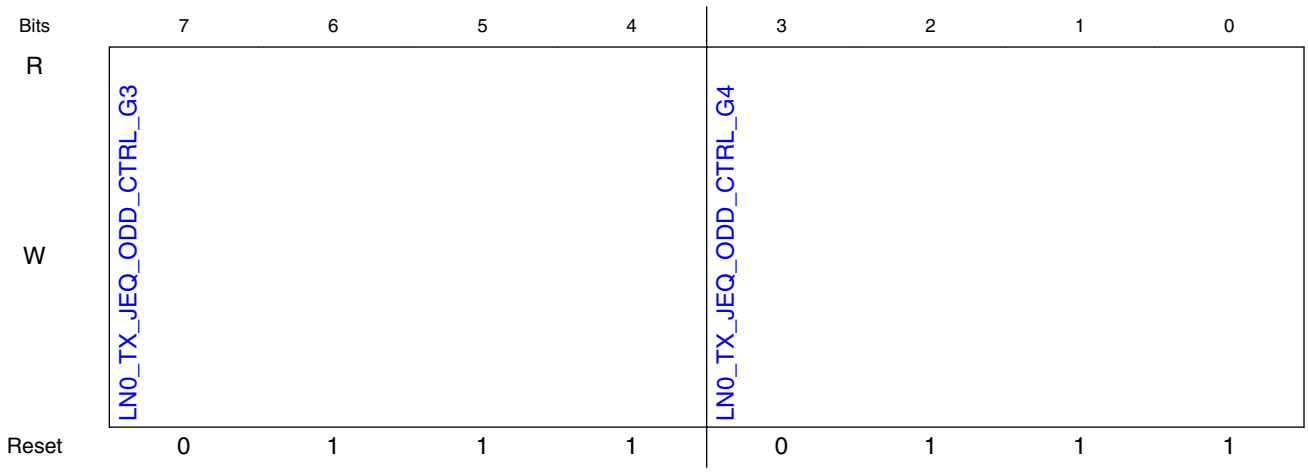
Field	Function
7-4 LN0_TX_JEQ_ODD_CTRL_G1	[GEN1] TX jitter EQ driver (odd) strength control
3-0 LN0_TX_JEQ_ODD_CTRL_G2	[GEN2]

11.5.3.1.149 (TRSV_REG016)

11.5.3.1.149.1 Offset

Register	Offset
TRSV_REG016	458h

11.5.3.1.149.2 Diagram



11.5.3.1.149.3 Fields

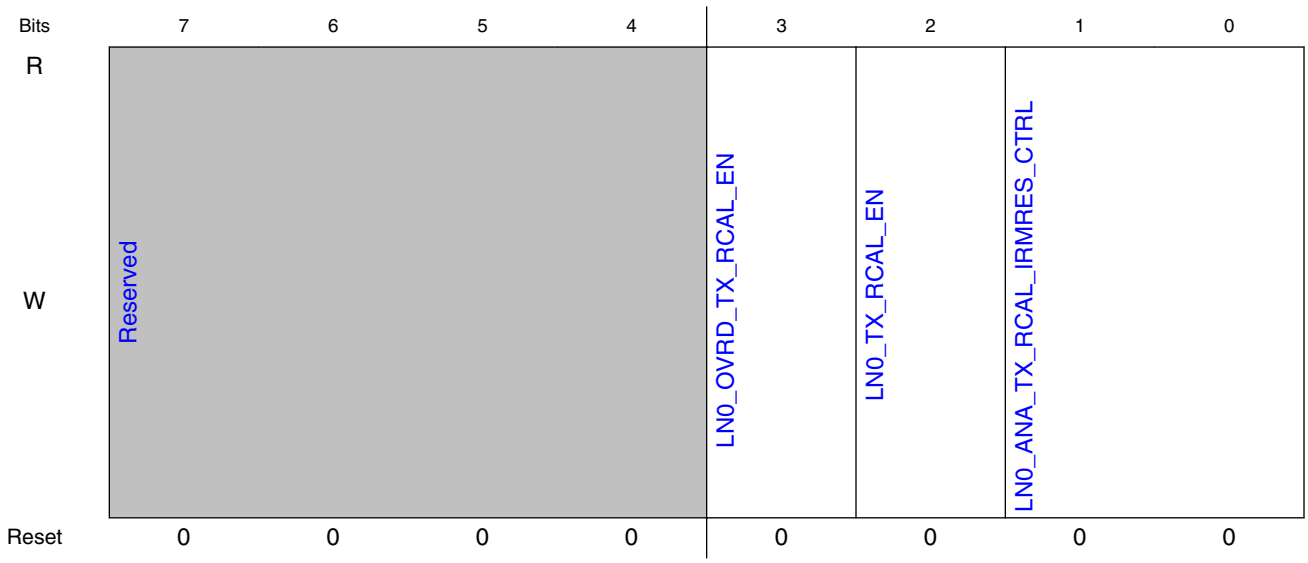
Field	Function
7-4 LN0_TX_JEQ_ODD_CTRL_G3	[GEN3]
3-0 LN0_TX_JEQ_ODD_CTRL_G4	[GEN4]

11.5.3.1.150 (TRSV_REG017)

11.5.3.1.150.1 Offset

Register	Offset
TRSV_REG017	45Ch

11.5.3.1.150.2 Diagram



11.5.3.1.150.3 Fields

Field	Function
7-4 —	Reserved
3 LN0_OVRD_TX_RCAL_EN	Override enable for tx_rcal_en
2 LN0_TX_RCAL_EN	TX RCAL enable 0: Disable, 1: Enable
1-0 LN0_ANA_TX_RCAL_IRMRES_CTRL	TX RCAL rmres bias current control 00: 85uA, 01:100uA, 10:100uA, 11:115uA T-coil + 42.5ohm cal: 2'b00 T-coil + 50ohm cal: 2'b01 w/o T-coil + 42.5ohm cal: 2'b01 w/o T-coil + 50ohm cal: 2'b11

11.5.3.1.151 (TRSV_REG018)

11.5.3.1.151.1 Offset

Register	Offset
TRSV_REG018	460h

11.5.3.1.151.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_TX_RTERM_42P5_EN_G1	LN0_TX_RTERM_42P5_EN_G2	LN0_TX_RTERM_42P5_EN_G3	LN0_TX_RTERM_42P5_EN_G4	LN0_OVRD_TX_RXD_COMP_EN	LN0_TX_RXD_COMP_EN	LN0_OVRD_TX_RXD_EN	LN0_TX_RXD_EN
W								
Reset	0	0	1	1	0	0	0	0

11.5.3.1.151.3 Fields

Field	Function
7 LN0_TX_RTERM_42P5_EN_G1	[GEN1]
6 LN0_TX_RTERM_42P5_EN_G2	[GEN2]
5 LN0_TX_RTERM_42P5_EN_G3	[GEN3]
4 LN0_TX_RTERM_42P5_EN_G4	[GEN4]
3	Override enable for tx_rxd_comp_en

Table continues on the next page...

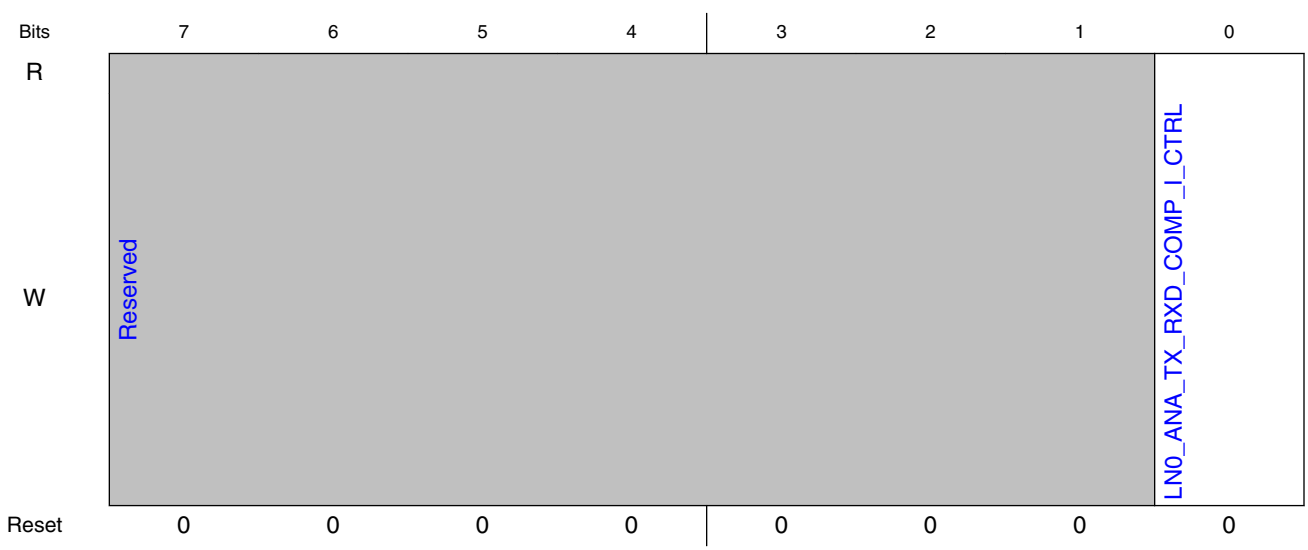
Field	Function
LN0_OVRD_TX_RXD_COMP_EN	
2 LN0_TX_RXD_COMP_EN	TX receiver detector comparator enable 0: Disable, 1: Enable
1 LN0_OVRD_TX_RXD_EN	Override enable for tx_rxd_en
0 LN0_TX_RXD_EN	TX receiver detector enable. Drives a transition on the serial data and measures the charge time of the line in order to determine whether a receiver is connected. 0: normal operation, 1: initiate a receiver detect sequence

11.5.3.1.152 (TRSV_REG019)

11.5.3.1.152.1 Offset

Register	Offset
TRSV_REG019	464h

11.5.3.1.152.2 Diagram



11.5.3.1.152.3 Fields

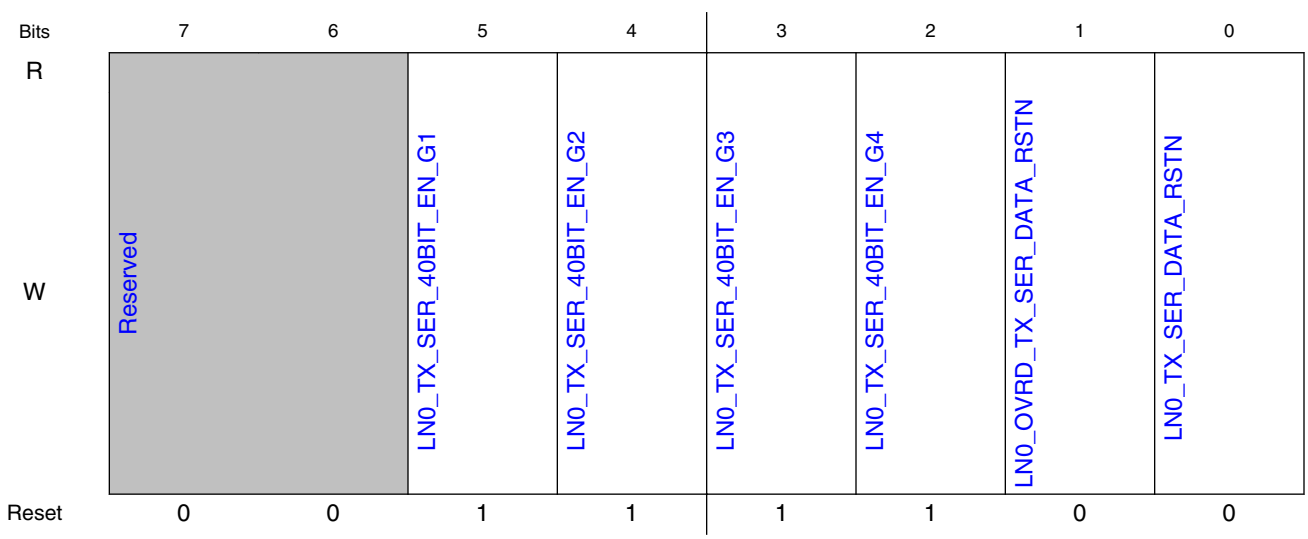
Field	Function
7-1 —	Reserved
0 LN0_ANA_TX_RXD_COMP_I_CTRL	TX receiver detector comparator bias control 0: 1x, 1: 3x

11.5.3.1.153 (TRSV_REG01A)

11.5.3.1.153.1 Offset

Register	Offset
TRSV_REG01A	468h

11.5.3.1.153.2 Diagram



11.5.3.1.153.3 Fields

Field	Function
7-6 —	Reserved

Table continues on the next page...

Field	Function
5 LN0_TX_SER_40BIT_EN_G1	[GEN1] TX serializer data width selection 0: 20/16-bit, 1: 40/32-bit
4 LN0_TX_SER_40BIT_EN_G2	[GEN2]
3 LN0_TX_SER_40BIT_EN_G3	[GEN3]
2 LN0_TX_SER_40BIT_EN_G4	[GEN4]
1 LN0_OVRD_TX_SER_DATA_RSTN	Override enable for tx_ser_data_rstn
0 LN0_TX_SER_DATA_RSTN	TX serializer data-path resetn 0: Reset, 1: Released

11.5.3.1.154 (TRSV_REG01B)

11.5.3.1.154.1 Offset

Register	Offset
TRSV_REG01B	46Ch

11.5.3.1.154.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_TX_SER_RATE_SEL_G1	LN0_TX_SER_RATE_SEL_G2	LN0_TX_SER_RATE_SEL_G3	LN0_TX_SER_RATE_SEL_G4	LN0_OVRD_TX_SER_CLK_RSTN	LN0_TX_SER_CLK_RSTN	LN0_ANA_TX_CDR_CLK_MON_EN	LN0_ANA_TX_SER_TXCLK_INV
W								
Reset	0	0	1	1	0	0	0	0

11.5.3.1.154.3 Fields

Field	Function
7 LN0_TX_SER_RATE_SEL_G1	[GEN1] TX serializer data rate selection for Gen4 (Need to be controlled with i_tx_en_40bit) 0: 20/40-bit 1: 16/32-bit
6 LN0_TX_SER_RATE_SEL_G2	[GEN2]
5 LN0_TX_SER_RATE_SEL_G3	[GEN3]
4 LN0_TX_SER_RATE_SEL_G4	[GEN4]
3 LN0_OVRD_TX_SER_CLK_RSTN	Override enable for tx_ser_clk_rstn
2 LN0_TX_SER_CLK_RSTN	TX serializer clock-path resetn 0: Reset, 1: Released
1 LN0_ANA_TX_CDR_CLK_MON_EN	TX serializer clock selection 0: PLL clock, 1: CDR clock
0	TX byte clock polarity inversion

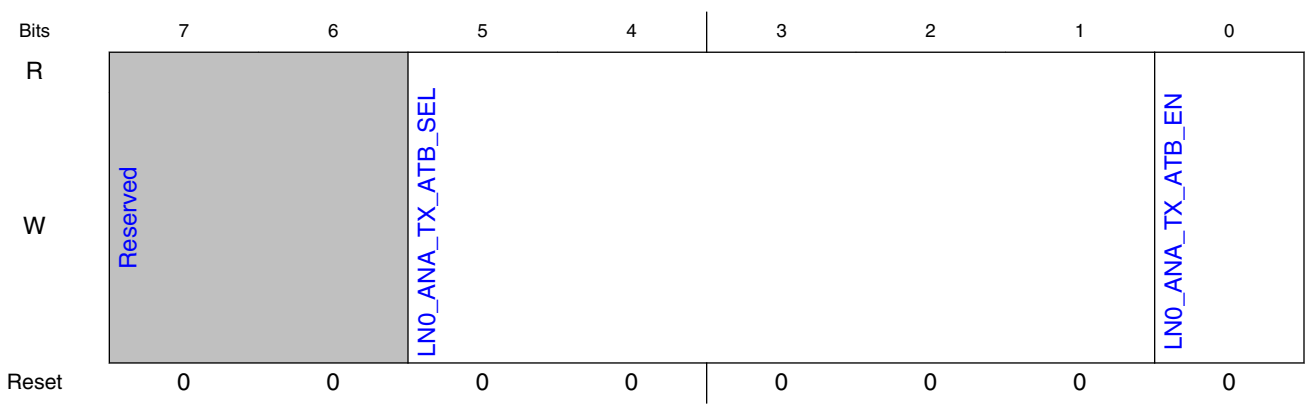
Field	Function
LN0_ANA_TX_SER_TXCLK_INV	0: No inversion, 1: Inversion

11.5.3.1.155 (TRSV_REG01C)

11.5.3.1.155.1 Offset

Register	Offset
TRSV_REG01C	470h

11.5.3.1.155.2 Diagram



11.5.3.1.155.3 Fields

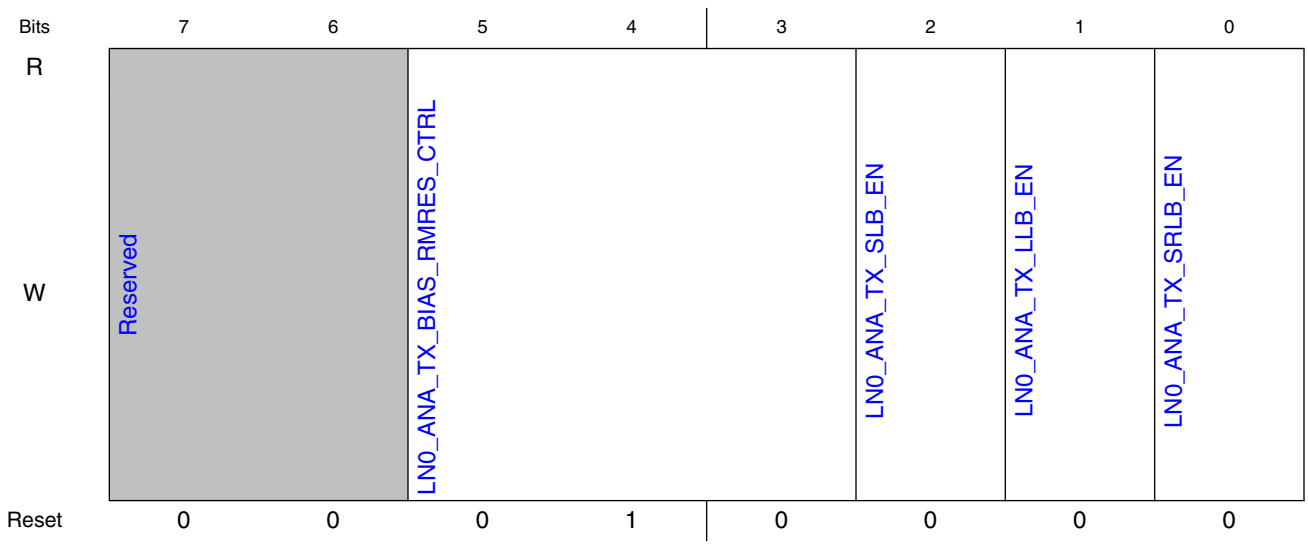
Field	Function
7-6 —	Reserved
5-1 LN0_ANA_TX_ATB_SEL	0000: Serailizer VDD, 0001: Pre Driver VDD, 0010: Driver VDD, 0011: Driver VDDH, 0100: Driver VDD, 0101: Serailizer VSS, 0110: Pre Driver VSS, 0111: Driver VSS, 1000: RCVD detected, 1001: VINM_FB_REPLICA (AUX BIAS), 1010: VBP_BIAS (AUX BIAS), 1011: VBN_BIAS (AUX BIAS), 1XXX: N/A
0 LN0_ANA_TX_ATB_EN	TX ATB enable

11.5.3.1.156 (TRSV_REG01D)

11.5.3.1.156.1 Offset

Register	Offset
TRSV_REG01D	474h

11.5.3.1.156.2 Diagram



11.5.3.1.156.3 Fields

Field	Function
7-6 —	Reserved
5-3 LN0_ANA_TX_BIAS_RMRES_CTRL	RX RMRES bias current control 000: 62%, 001: 87%, 010: 100% (Default), 100: 138%, 111: 156%
2 LN0_ANA_TX_SLB_EN	Serial loopback enable 0: Disable, 1: Enable (PMA digital → TX SER → RX CDR (DES) → PMA digital)
1 LN0_ANA_TX_LLB_EN	Line loopback enalble 0: Disable, 1: Enable (RX pad → RX AFE (CTLE) → TX driver → TX PAD)
0 LN0_ANA_TX_SRLB_EN	Serial retimed loopback enable 0: Disable 1: Enable (RX pad → RX AFE → RX CDR (BBPD) → TX driver → TX pad)

11.5.3.1.157 (TRSV_REG01E)

11.5.3.1.157.1 Offset

Register	Offset
TRSV_REG01E	478h

11.5.3.1.157.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_TX_EQ_2UI_DELAY_EN_G1	LN0_TX_EQ_2UI_DELAY_EN_G2	LN0_TX_EQ_2UI_DELAY_EN_G3	LN0_TX_EQ_2UI_DELAY_EN_G4	LN0_ANA_TX_RESERVED			
W								
Reset	1	0	0	0	0	0	0	0

11.5.3.1.157.3 Fields

Field	Function
7 LN0_TX_EQ_2UI_DELAY_EN_G1	[GEN1] TX FIR filter delay control when bit-duplication 0: 1UI delay, 1: 2UI delay
6 LN0_TX_EQ_2UI_DELAY_EN_G2	[GEN2]
5 LN0_TX_EQ_2UI_DELAY_EN_G3	[GEN3]
4	[GEN4]

Table continues on the next page...

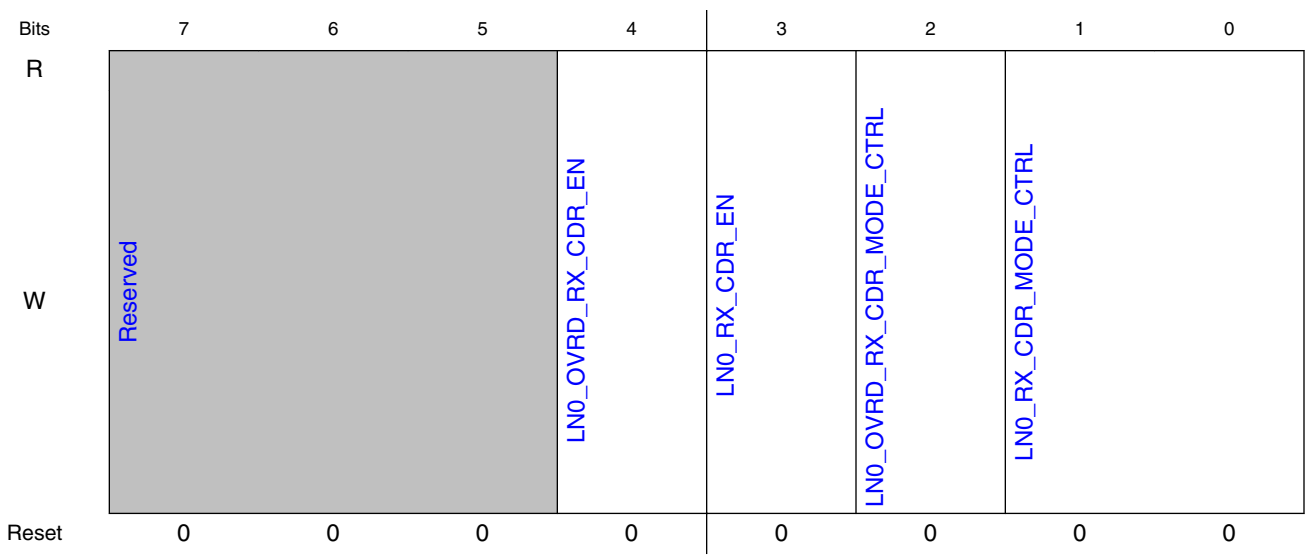
Field	Function
LN0_TX_EQ_2UI_DELAY_EN_G4	
3-0 LN0_ANA_TX_RESERVED	Reserved port

11.5.3.1.158 (TRSV_REG01F)

11.5.3.1.158.1 Offset

Register	Offset
TRSV_REG01F	47Ch

11.5.3.1.158.2 Diagram



11.5.3.1.158.3 Fields

Field	Function
7-5 —	Reserved
4	Override enable for rx_cdr_en

Table continues on the next page...

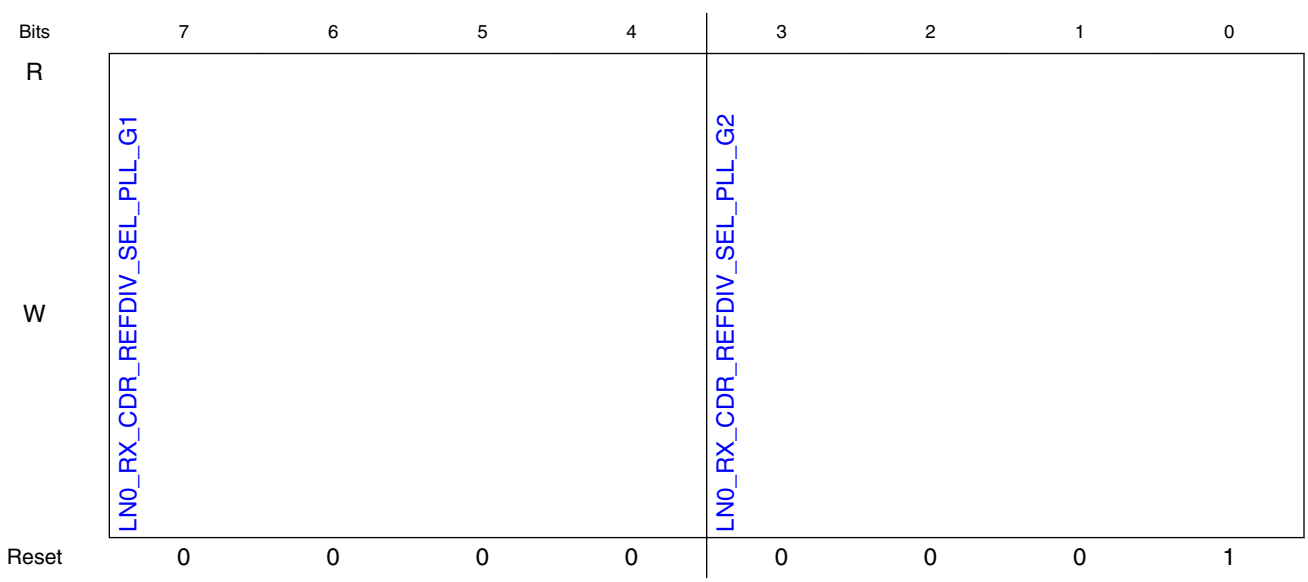
Field	Function
LN0_OVRD_RX_CDR_EN	
3 LN0_RX_CDR_EN	RX CDR enable 0: Disable, 1: Enable
2 LN0_OVRD_RX_CDR_MODE_CTRL	Override enable for rx_cdr_mode_ctrl
1-0 LN0_RX_CDR_MODE_CTRL	RX CDR mode select 01 : Rx AFC& Calibaration mode, 00 : PLL mode, 10 : CDR clock mode, 11 : CDR data mode

11.5.3.1.159 (TRSV_REG020)

11.5.3.1.159.1 Offset

Register	Offset
TRSV_REG020	480h

11.5.3.1.159.2 Diagram



11.5.3.1.159.3 Fields

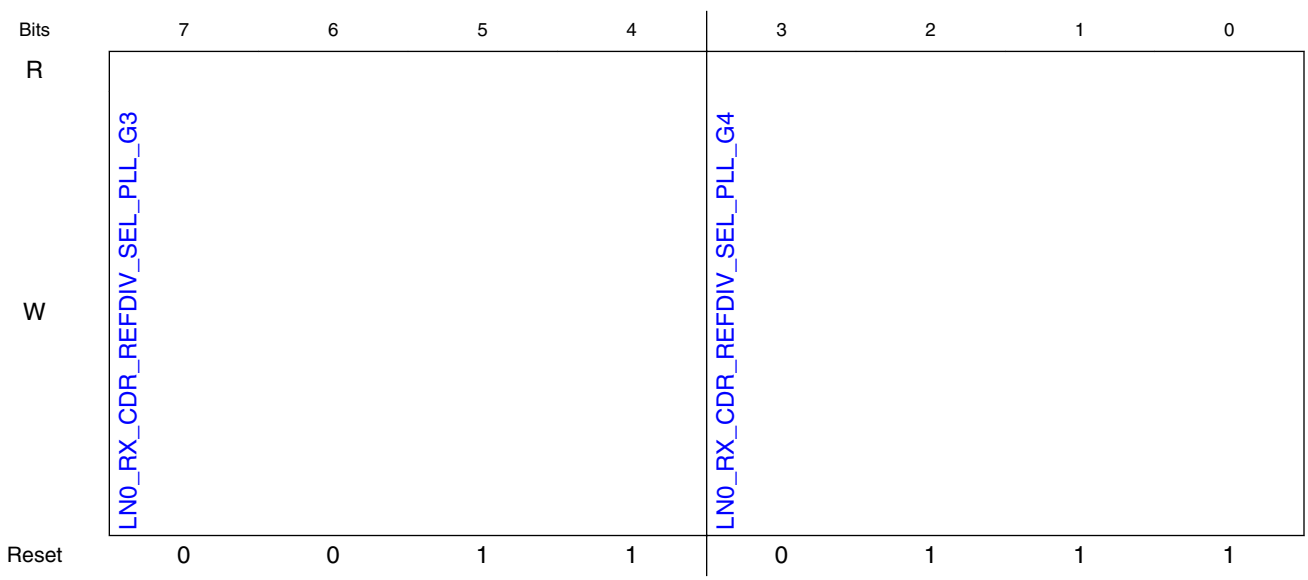
Field	Function
7-4 LN0_RX_CDR_REFDIV_SEL_PLL_G1	[GEN1] [PLL mode] Decision of CDR ref. clock dividing-rate 1111 : div16, 0111 : div8, 0011 : div4, 0001 : div2, 0000: div1
3-0 LN0_RX_CDR_REFDIV_SEL_PLL_G2	[GEN2] [PLL mode]

11.5.3.1.160 (TRSV_REG021)

11.5.3.1.160.1 Offset

Register	Offset
TRSV_REG021	484h

11.5.3.1.160.2 Diagram



11.5.3.1.160.3 Fields

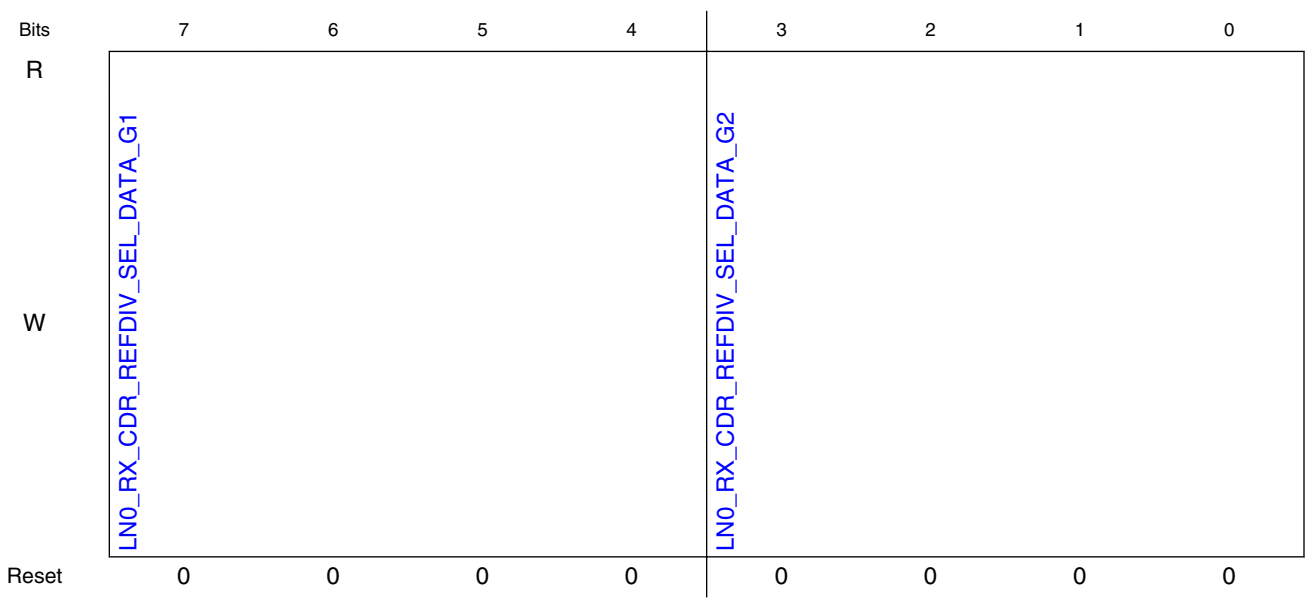
Field	Function
7-4 LN0_RX_CDR_REFDIV_SEL_PLL_G3	[GEN3] [PLL mode]
3-0 LN0_RX_CDR_REFDIV_SEL_PLL_G4	[GEN4] [PLL mode]

11.5.3.1.161 (TRSV_REG022)

11.5.3.1.161.1 Offset

Register	Offset
TRSV_REG022	488h

11.5.3.1.161.2 Diagram



11.5.3.1.161.3 Fields

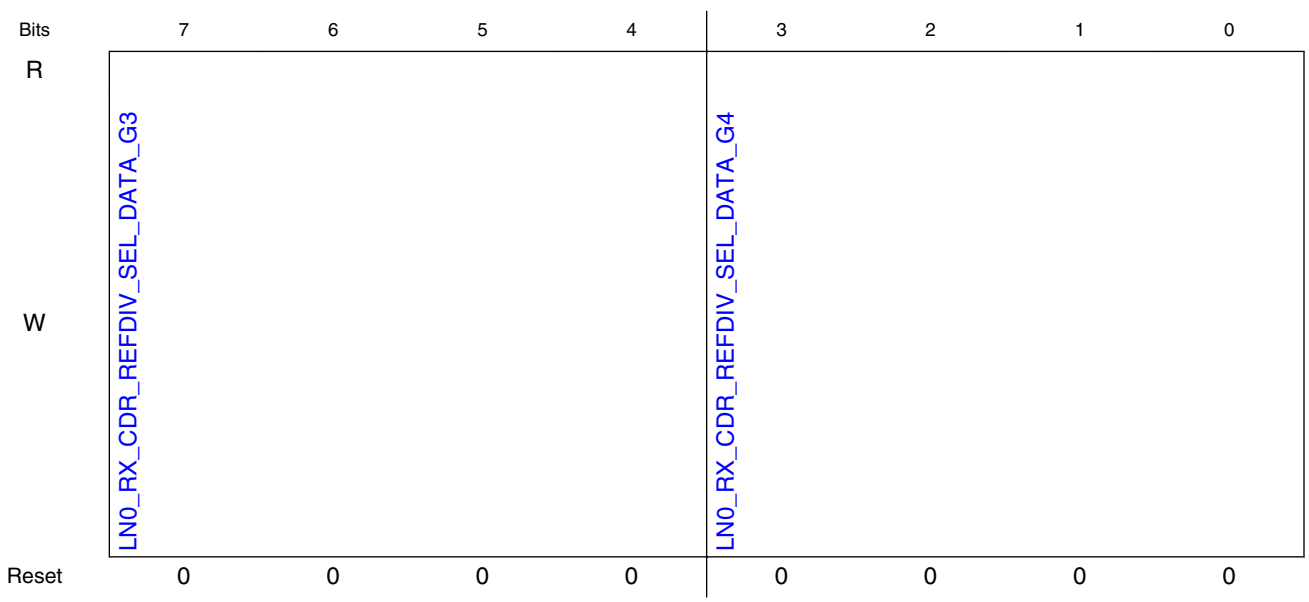
Field	Function
7-4 LN0_RX_CDR_REFDIV_SEL_DATA_G1	[GEN1] [Data mode] Decision of CDR ref. divider ratio 1111 : div16, 0111 : div8, 0011 : div4, 0001 : div2, 0000: div1
3-0 LN0_RX_CDR_REFDIV_SEL_DATA_G2	[GEN2] [Data mode]

11.5.3.1.162 (TRSV_REG023)

11.5.3.1.162.1 Offset

Register	Offset
TRSV_REG023	48Ch

11.5.3.1.162.2 Diagram



11.5.3.1.162.3 Fields

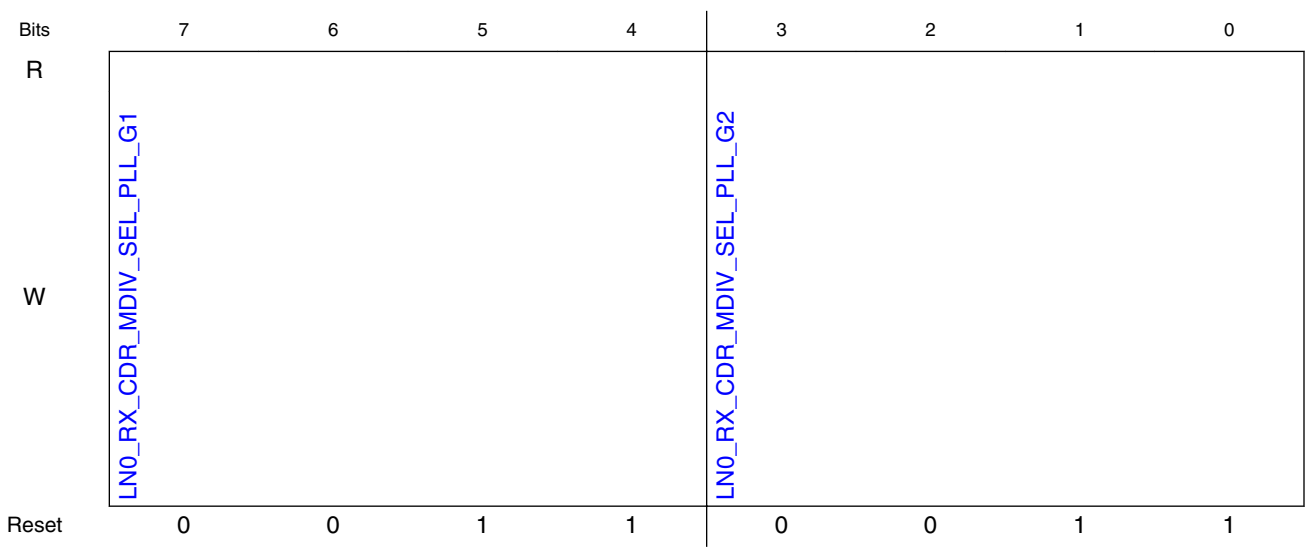
Field	Function
7-4 LN0_RX_CDR_REFDIV_SEL_DATA_G3	[GEN3] [Data mode]
3-0 LN0_RX_CDR_REFDIV_SEL_DATA_G4	[GEN4] [Data mode]

11.5.3.1.163 (TRSV_REG024)

11.5.3.1.163.1 Offset

Register	Offset
TRSV_REG024	490h

11.5.3.1.163.2 Diagram



11.5.3.1.163.3 Fields

Field	Function
7-4	[GEN1] [PLL mode] Decision of CDR main-divider ratio

Table continues on the next page...

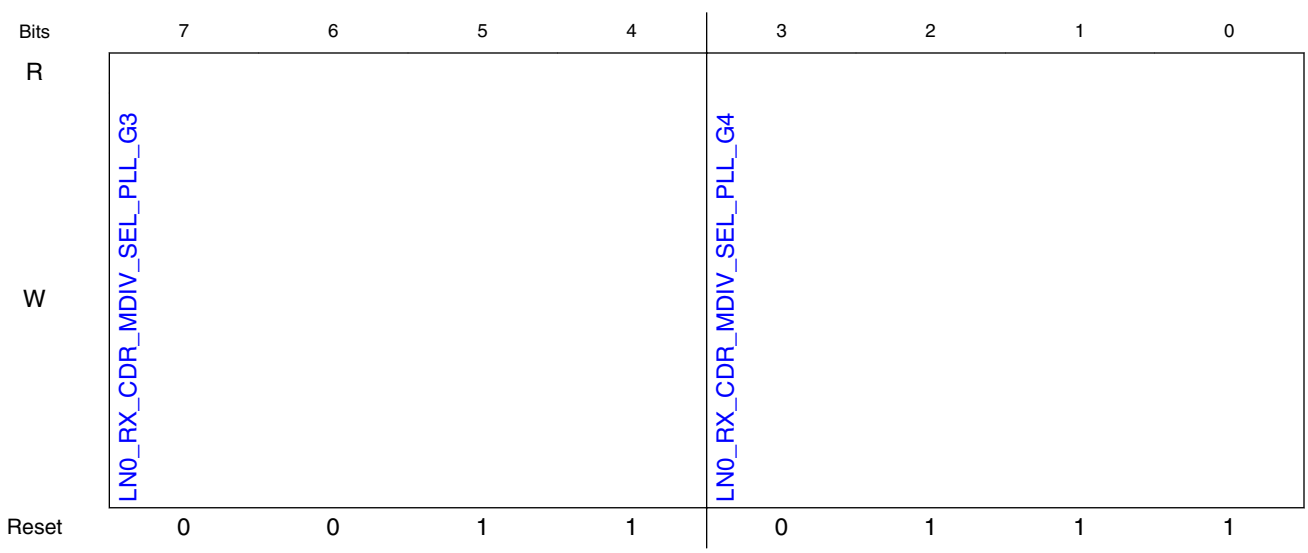
Field	Function
LN0_RX_CDR_MDIV_SEL_PLL_G1	1111 : div16, 0111 : div8, 0011 : div4, 0001 : div2, 0000: div1
3-0 LN0_RX_CDR_MDIV_SEL_PLL_G2	[GEN2] [PLL mode]

11.5.3.1.164 (TRSV_REG025)

11.5.3.1.164.1 Offset

Register	Offset
TRSV_REG025	494h

11.5.3.1.164.2 Diagram



11.5.3.1.164.3 Fields

Field	Function
7-4 LN0_RX_CDR_MDIV_SEL_PLL_G3	[GEN3] [PLL mode]

Table continues on the next page...

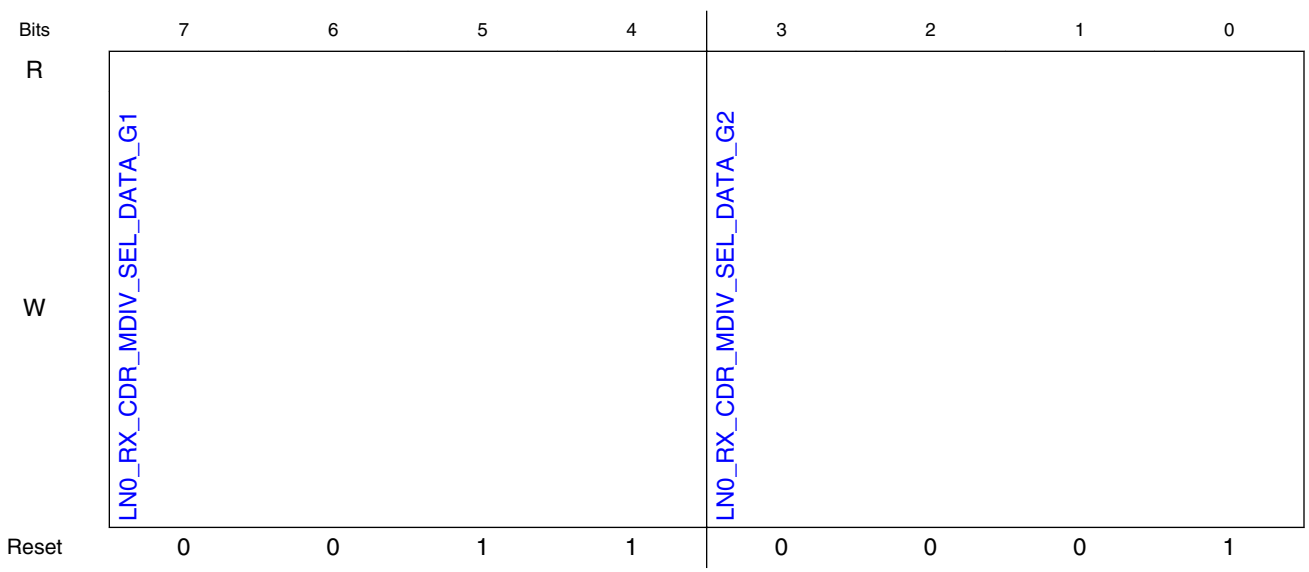
Field	Function
3-0 LN0_RX_CDR_MDIV_SEL_PLL_G4	[GEN4] [PLL mode]

11.5.3.1.165 (TRSV_REG026)

11.5.3.1.165.1 Offset

Register	Offset
TRSV_REG026	498h

11.5.3.1.165.2 Diagram



11.5.3.1.165.3 Fields

Field	Function
7-4 LN0_RX_CDR_MDIV_SEL_DATA_G1	[GEN1] [Data mode] Decision of CDR main-divider ratio 1111 : div16, 0111 : div8, 0011 : div4, 0001 : div2, 0000: div1
3-0	[GEN2] [Data mode]

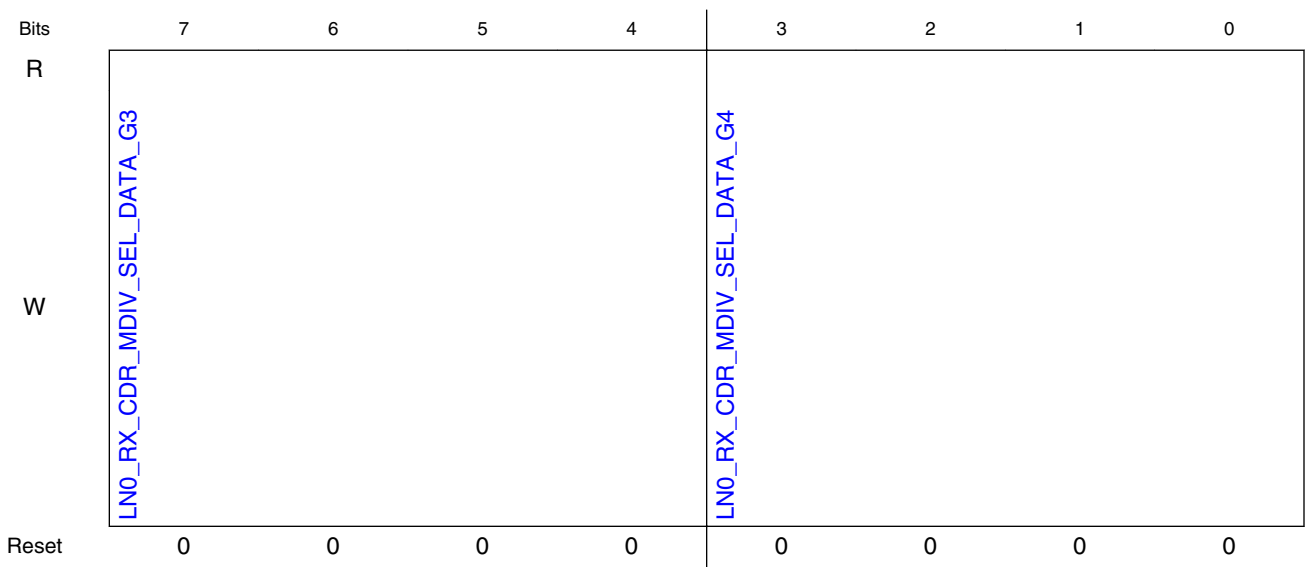
Field	Function
LN0_RX_CDR_MDIV_SEL_DATA_G2	

11.5.3.1.166 (TRSV_REG027)

11.5.3.1.166.1 Offset

Register	Offset
TRSV_REG027	49Ch

11.5.3.1.166.2 Diagram



11.5.3.1.166.3 Fields

Field	Function
7-4 LN0_RX_CDR_MDIV_SEL_DATA_G3	[GEN3] [Data mode]
3-0 LN0_RX_CDR_MDIV_SEL_DATA_G4	[GEN4] [Data mode]

11.5.3.1.167 (TRSV_REG028)

11.5.3.1.167.1 Offset

Register	Offset
TRSV_REG028	4A0h

11.5.3.1.167.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	Reserved		LN0_OVRD_RX_CDR_BW_CTRL	LN0_RX_CDR_BW_CTRL	LN0_ANA_RX_CDR_DES_RXCLK_INV	LN0_ANA_RX_CDR_AFC_EN	LN0_ANA_RX_CDR_AFC_TEST_EN	LN0_ANA_RX_CDR_AFC_VCI_FORCE
W								
Reset	0	0	0	0	0	1	0	0

11.5.3.1.167.3 Fields

Field	Function
7-6 —	Reserved
5 LN0_OVRD_RX_CDR_BW_CTRL	Override enable for rx_cdr_bw_ctrl
4 LN0_RX_CDR_BW_CTRL	RX CDR bandwidth control 0: wide bandwidth, 1: narror bandwidth
3	RX byte clock polarity inversion 0: No inversion, 1: Inversion

Table continues on the next page...

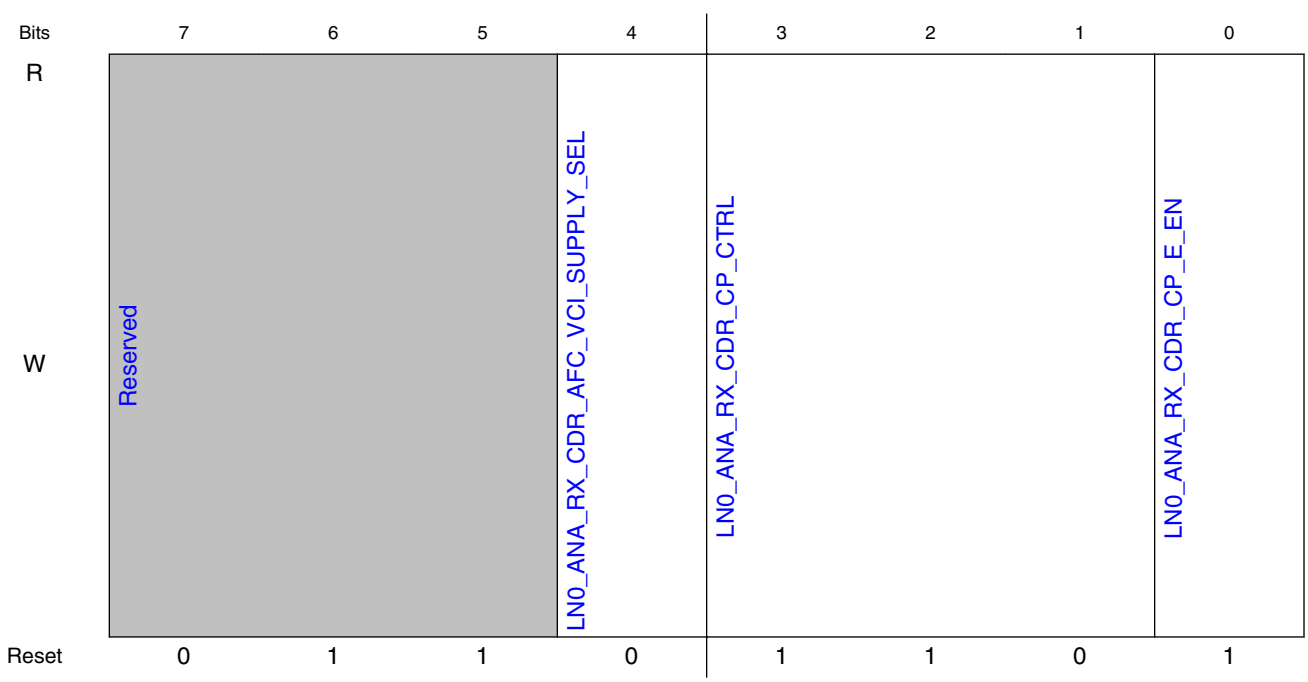
Field	Function
LN0_ANA_RX_CDR_DES_RXCLK_INV	
2 LN0_ANA_RX_CDR_AFC_EN	RX CDR AFC enable 0: Disable, 1: Enable
1 LN0_ANA_RX_CDR_AFC_TEST_EN	RX CDR test mode enable 0: Disable, 1: Enable
0 LN0_ANA_RX_CDR_AFC_VCI_FORCE	RX CDR control voltage force

11.5.3.1.168 (TRSV_REG029)

11.5.3.1.168.1 Offset

Register	Offset
TRSV_REG029	4A4h

11.5.3.1.168.2 Diagram



11.5.3.1.168.3 Fields

Field	Function
7-5 —	Reserved
4 LN0_ANA_RX_CDR_AFC_VCI_SUPPLY_SEL	RX CDR VCI reference voltage selection 0: Regulated 820mV or VDD after LPF (depending on i_rx_cdr_cp_reg_in_sel), 1: VDD
3-1 LN0_ANA_RX_CDR_CP_CTRL	RX CDR charge pump current control (Ieven + Iodd) 000: 2.5uA, 001: 3.2uA, 010: 4.2uA, 011: 6.3uA 100: 1.4uA, 101: 1.6uA, 110: 1.8uA, 111: 2.1uA
0 LN0_ANA_RX_CDR_CP_E_EN	RX CDR even charge-pump enable 0: Disable, 1: Enable

11.5.3.1.169 (TRSV_REG02A)

11.5.3.1.169.1 Offset

Register	Offset
TRSV_REG02A	4A8h

11.5.3.1.169.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	Reserved	LN0_ANA_RX_CDR_CP_O_EN	LN0_ANA_RX_CDR_CP_VREG_IN_SEL	LN0_ANA_RX_CDR_CP_VREG_LPF_EN	LN0_OVRD_RX_CDR_FBB_CAL_EN	LN0_RX_CDR_FBB_CAL_EN	LN0_OVRD_RX_CDR_VCO_STARTUP	LN0_RX_CDR_VCO_STARTUP
W								
Reset	0	1	0	0	1	0	0	0

11.5.3.1.169.3 Fields

Field	Function
7 —	Reserved
6 LN0_ANA_RX_CDR_CP_O_EN	RX CDR odd charge-pump enable 0: Disable, 1: Enable
5 LN0_ANA_RX_CDR_CP_VREG_IN_SEL	RX CDR charge pump regulator reference voltage selection 0: BGR 820mV, 1: VDD
4 LN0_ANA_RX_CDR_CP_VREG_LPF_EN	LPF enable for RX CDR charge pump regualtor 0: Disable, 1: Enable
3 LN0_OVRD_RX_CDR_FBB_CAL_EN	Override enable for rx_cdr_fbb_cal_en
2 LN0_RX_CDR_FBB_CAL_EN	RX CDR FBB calibration enable 0: Disable, 1: Enable
1	Override enable for rx_cdr_vco_startup

Table continues on the next page...

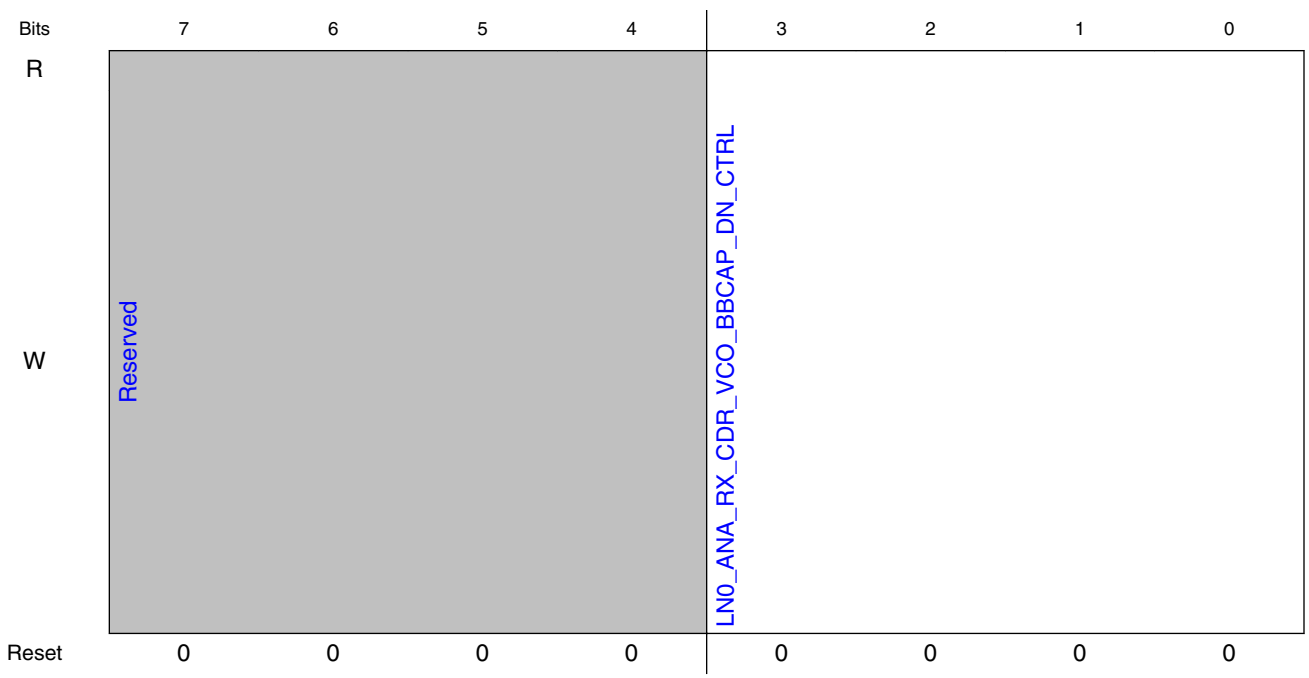
Field	Function
LN0_OVRD_RX_CDR_VCO_STARTUP	
0 LN0_RX_CDR_VCO_STARTUP	RX CDR VCO startup signal, low to high transition

11.5.3.1.170 (TRSV_REG02B)

11.5.3.1.170.1 Offset

Register	Offset
TRSV_REG02B	4ACh

11.5.3.1.170.2 Diagram



11.5.3.1.170.3 Fields

Field	Function
7-4	Reserved

Table continues on the next page...

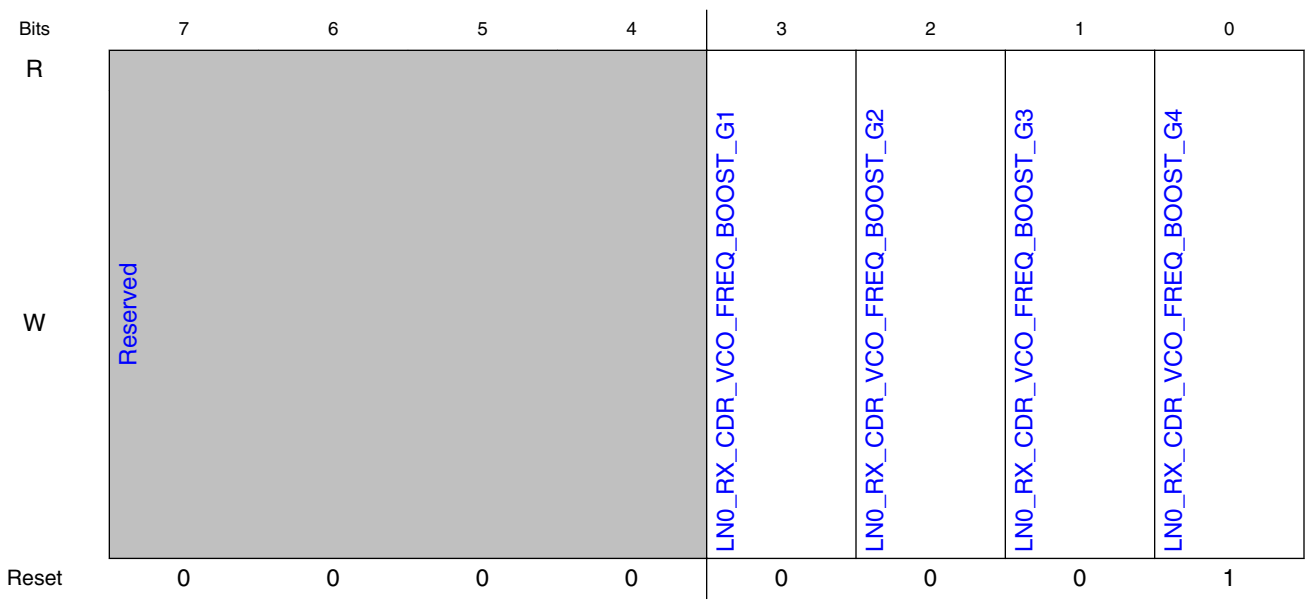
Field	Function
—	
3-0 LN0_ANA_RX_CDR_VCO_BBCAP_DN_CTRL	RX CDR BBVCO dummy cap control to decrease frequency 0000: Min freq, 1111: Max freq.

11.5.3.1.171 (TRSV_REG02C)

11.5.3.1.171.1 Offset

Register	Offset
TRSV_REG02C	4B0h

11.5.3.1.171.2 Diagram



11.5.3.1.171.3 Fields

Field	Function
7-4 —	Reserved
3	[GEN1] RX CDR VCO frequency boost enable

Table continues on the next page...

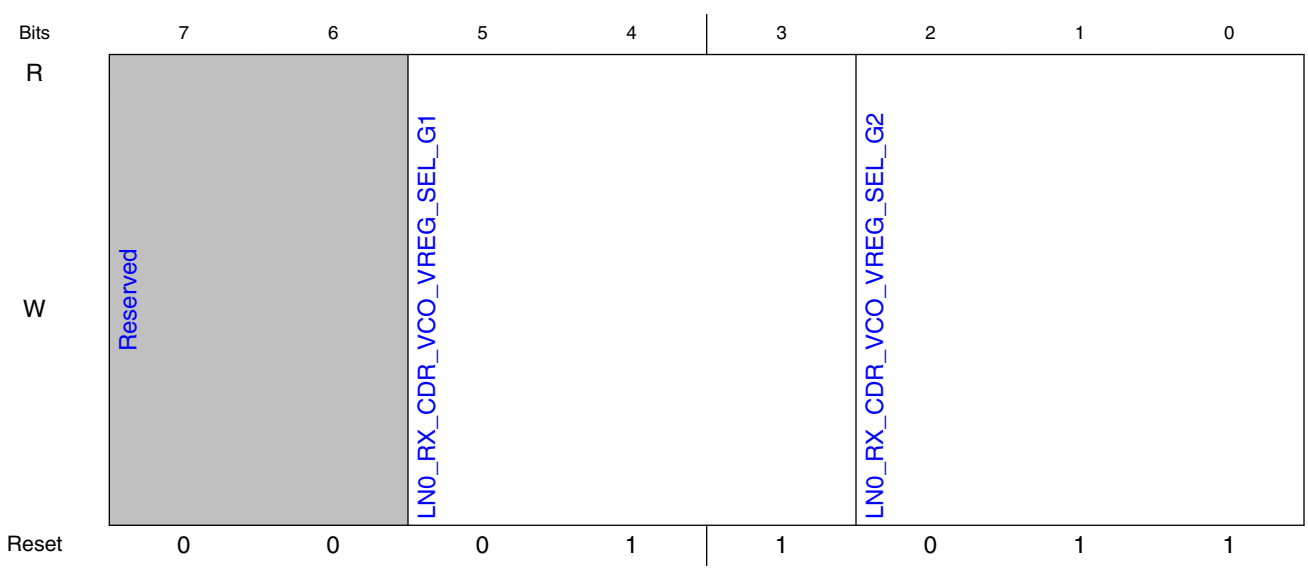
Field	Function
LN0_RX_CDR_VCO_FREQ_BOOST_G1	0: Disable, 1: Enable (High freqency)
2 LN0_RX_CDR_VCO_FREQ_BOOST_G2	[GEN2]
1 LN0_RX_CDR_VCO_FREQ_BOOST_G3	[GEN3]
0 LN0_RX_CDR_VCO_FREQ_BOOST_G4	[GEN4]

11.5.3.1.172 (TRSV_REG02D)

11.5.3.1.172.1 Offset

Register	Offset
TRSV_REG02D	4B4h

11.5.3.1.172.2 Diagram



11.5.3.1.172.3 Fields

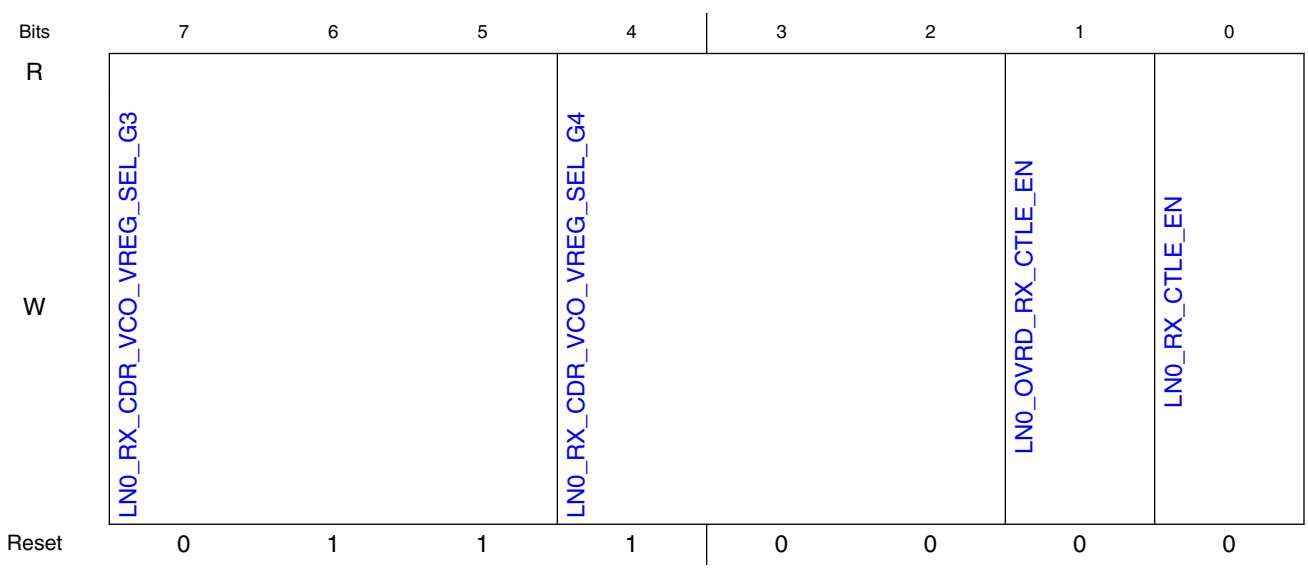
Field	Function
7-6 —	Reserved
5-3 LN0_RX_CDR_VCO_VREG_SEL_G1	[GEN1] RX CDR voltage regualtor selection 000: 1.0 X Vctrl, 001: 1.2 X Vctrl, 010: 1.4 X Vctrl, 011: 1.6 X Vctrl, 100: 1.8 X Vctrl, 101: 2.0 X Vctrl, 110: 2.2 X Vctrl, 111: 2.4 X Vctrl
2-0 LN0_RX_CDR_VCO_VREG_SEL_G2	[GEN2]

11.5.3.1.173 (TRSV_REG02E)

11.5.3.1.173.1 Offset

Register	Offset
TRSV_REG02E	4B8h

11.5.3.1.173.2 Diagram



11.5.3.1.173.3 Fields

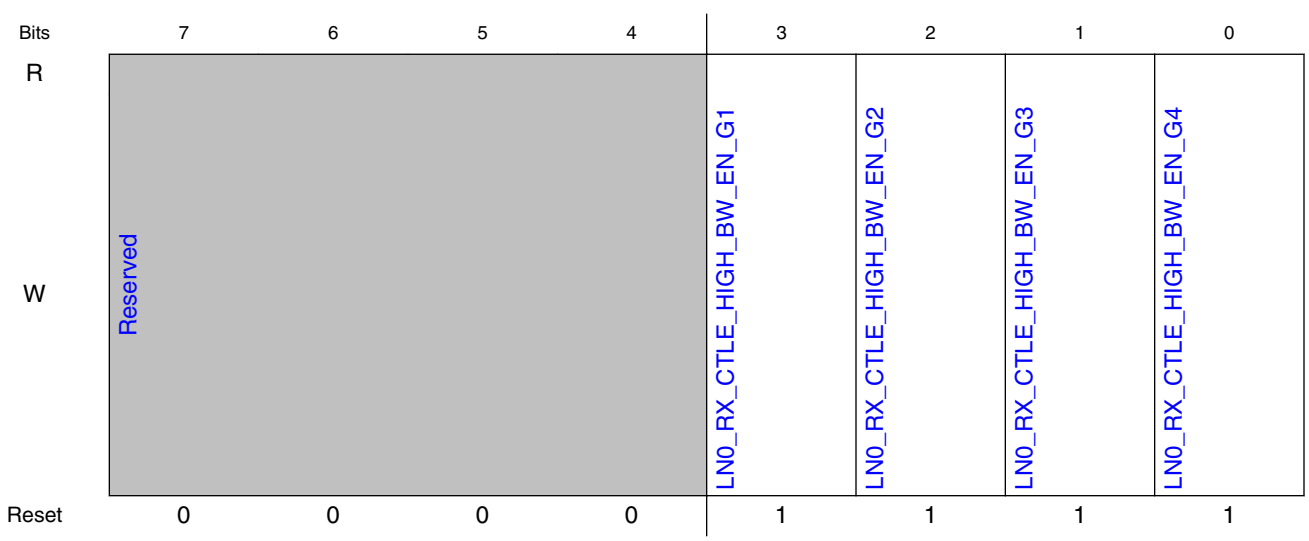
Field	Function
7-5 LN0_RX_CDR_VCO_VREG_SEL_G3	[GEN3]
4-2 LN0_RX_CDR_VCO_VREG_SEL_G4	[GEN4]
1 LN0_OVRD_RX_CTLE_EN	Override enable for rx_ctle_en
0 LN0_RX_CTLE_EN	RX CTLE enable 0: Disable, 1: Enable

11.5.3.1.174 (TRSV_REG02F)

11.5.3.1.174.1 Offset

Register	Offset
TRSV_REG02F	4BCh

11.5.3.1.174.2 Diagram



11.5.3.1.174.3 Fields

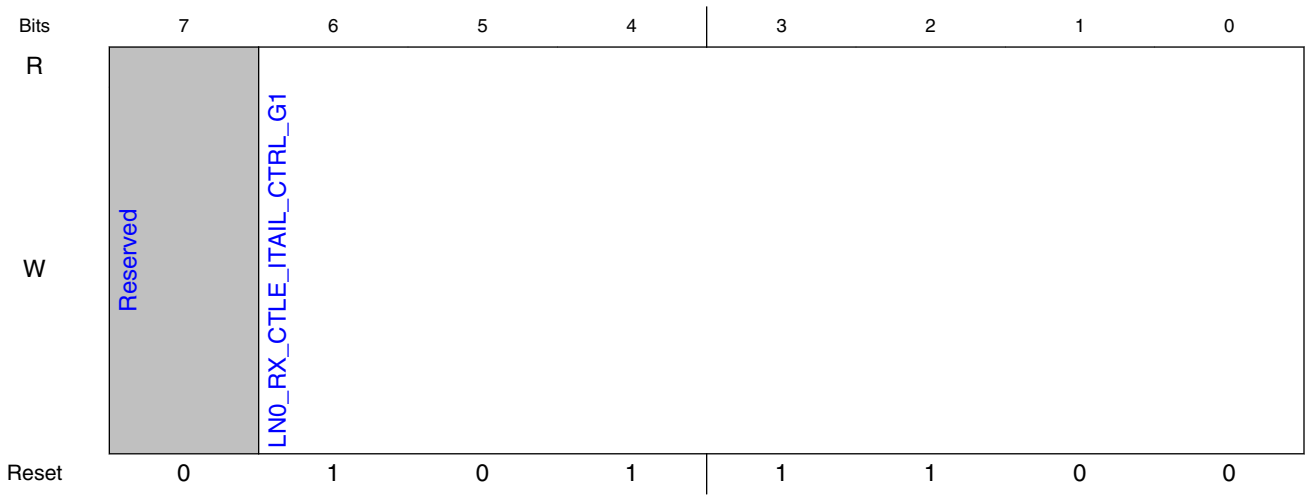
Field	Function
7-4 —	Reserved
3 LN0_RX_CTLE_HIGH_BW_EN_G1	[GEN1] RX CTLE bandwidth enhancement by boosting up current 0: Nominal BW, 1: Higher BW
2 LN0_RX_CTLE_HIGH_BW_EN_G2	[GEN2]
1 LN0_RX_CTLE_HIGH_BW_EN_G3	[GEN3]
0 LN0_RX_CTLE_HIGH_BW_EN_G4	[GEN4]

11.5.3.1.175 (TRSV_REG030)

11.5.3.1.175.1 Offset

Register	Offset
TRSV_REG030	4C0h

11.5.3.1.175.2 Diagram



11.5.3.1.175.3 Fields

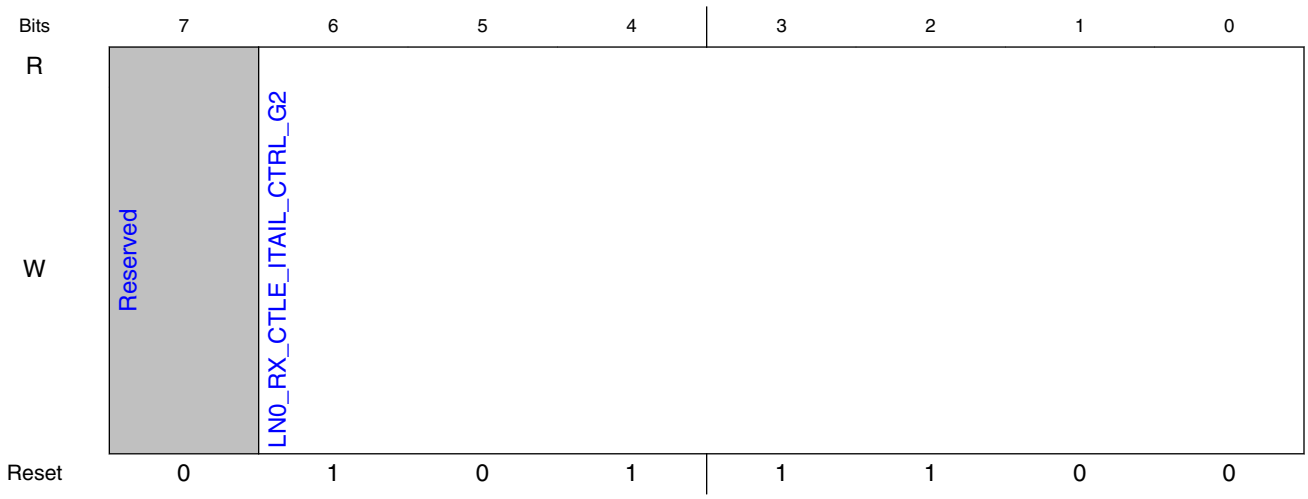
Field	Function
7 —	Reserved
6-0 LN0_RX_CTLE_ITAIL_CTRL_G1	[GEN1] RX CTLE main tail current

11.5.3.1.176 (TRSV_REG031)

11.5.3.1.176.1 Offset

Register	Offset
TRSV_REG031	4C4h

11.5.3.1.176.2 Diagram



11.5.3.1.176.3 Fields

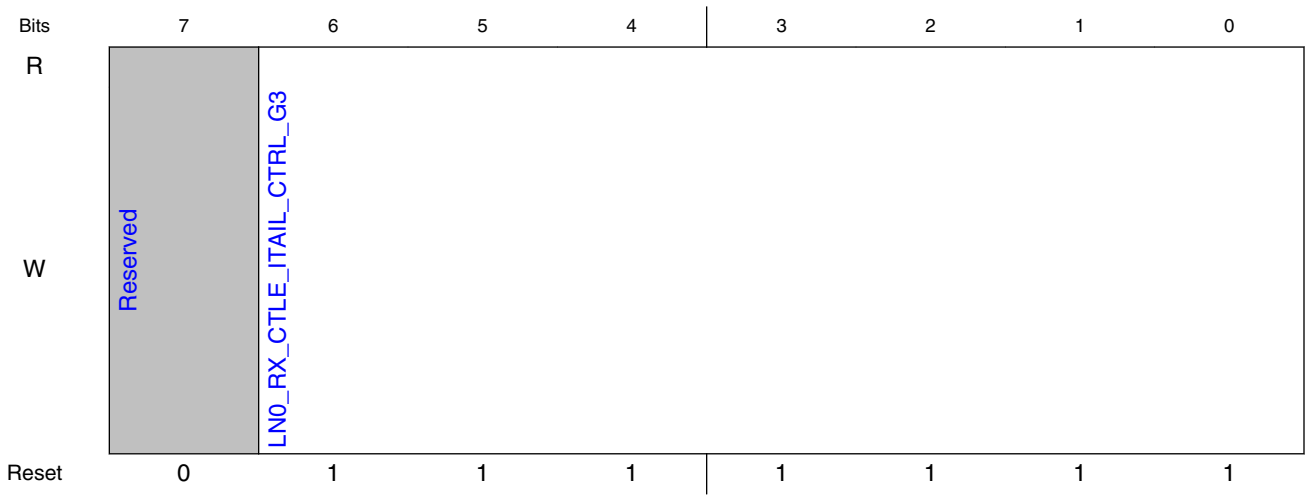
Field	Function
7 —	Reserved
6-0 LN0_RX_CTLE_ITAIL_CTRL_G2	[GEN2]

11.5.3.1.177 (TRSV_REG032)

11.5.3.1.177.1 Offset

Register	Offset
TRSV_REG032	4C8h

11.5.3.1.177.2 Diagram



11.5.3.1.177.3 Fields

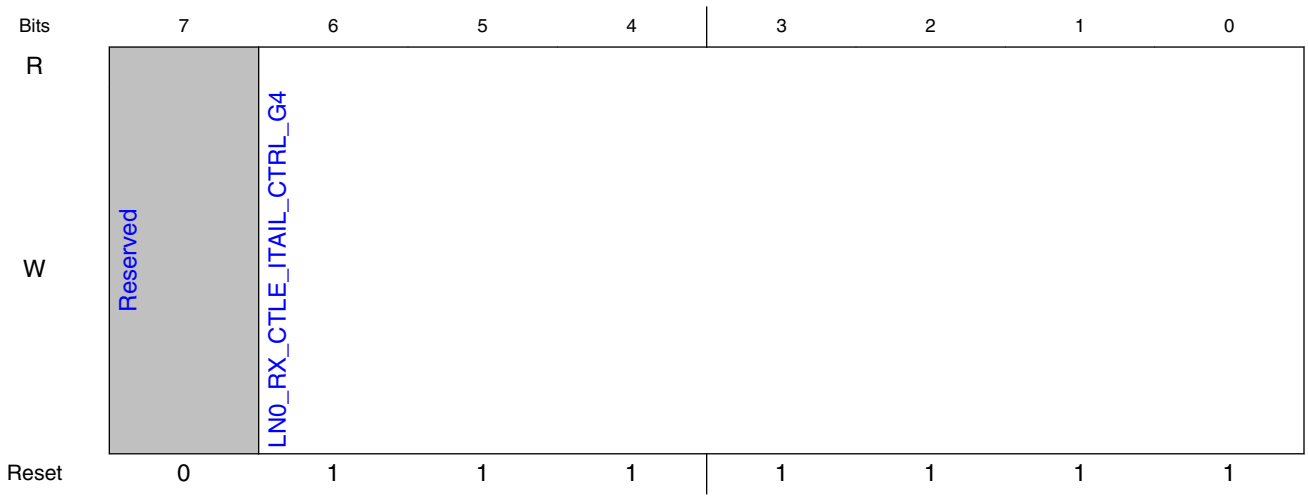
Field	Function
7 —	Reserved
6-0 LN0_RX_CTLE_ITAIL_CTRL_G3	[GEN3]

11.5.3.1.178 (TRSV_REG033)

11.5.3.1.178.1 Offset

Register	Offset
TRSV_REG033	4CCh

11.5.3.1.178.2 Diagram



11.5.3.1.178.3 Fields

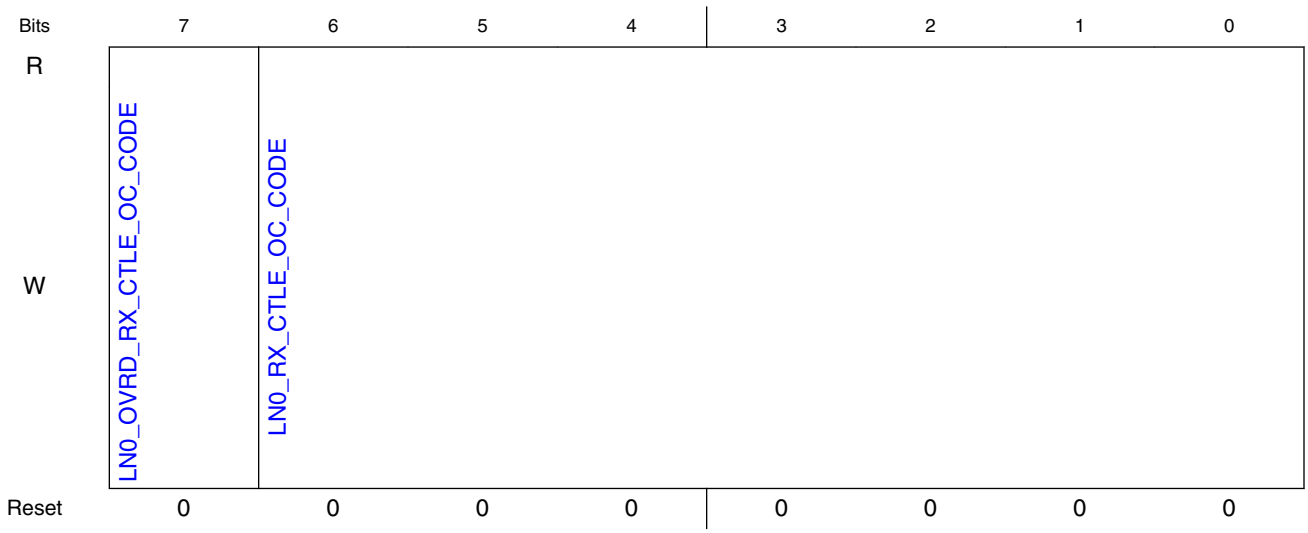
Field	Function
7 —	Reserved
6-0 LN0_RX_CTLE_ITAIL_CTRL_G4	[GEN4]

11.5.3.1.179 (TRSV_REG034)

11.5.3.1.179.1 Offset

Register	Offset
TRSV_REG034	4D0h

11.5.3.1.179.2 Diagram



11.5.3.1.179.3 Fields

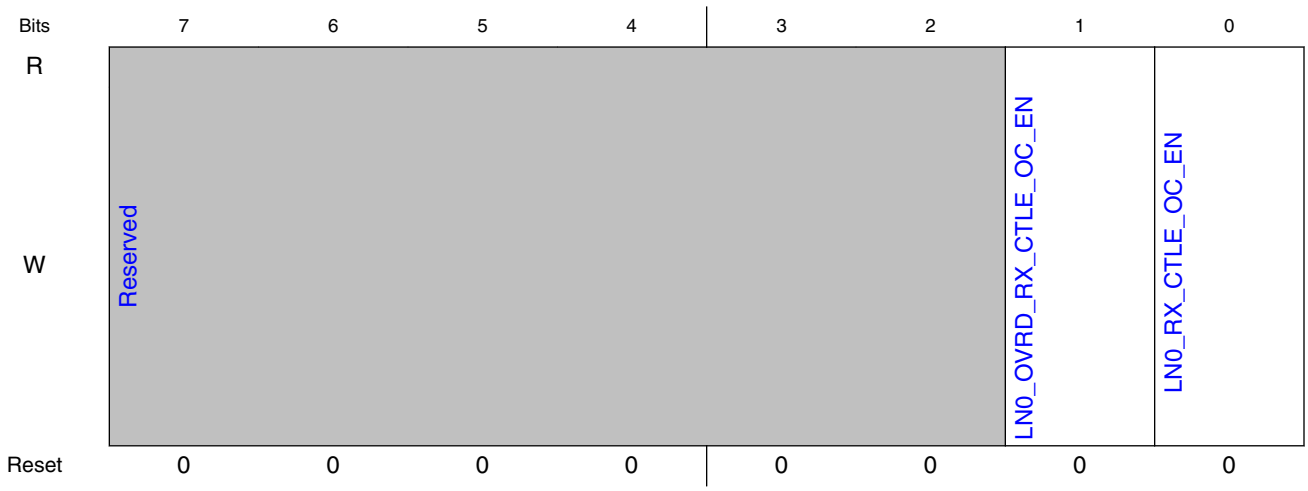
Field	Function
7 LN0_OVRD_RX_CTLE_OC_CODE	Override enable for rx_ctle_oc_code
6-0 LN0_RX_CTLE_OC_CODE	RX CTLE manual offset code 2 MSB: Range control - 00: Disable, 01: Min, 10: Medium, 11: Max range 5 LSB: Offset magnitude - binary weighted and 5'b10000 is center code.

11.5.3.1.180 (TRSV_REG035)

11.5.3.1.180.1 Offset

Register	Offset
TRSV_REG035	4D4h

11.5.3.1.180.2 Diagram



11.5.3.1.180.3 Fields

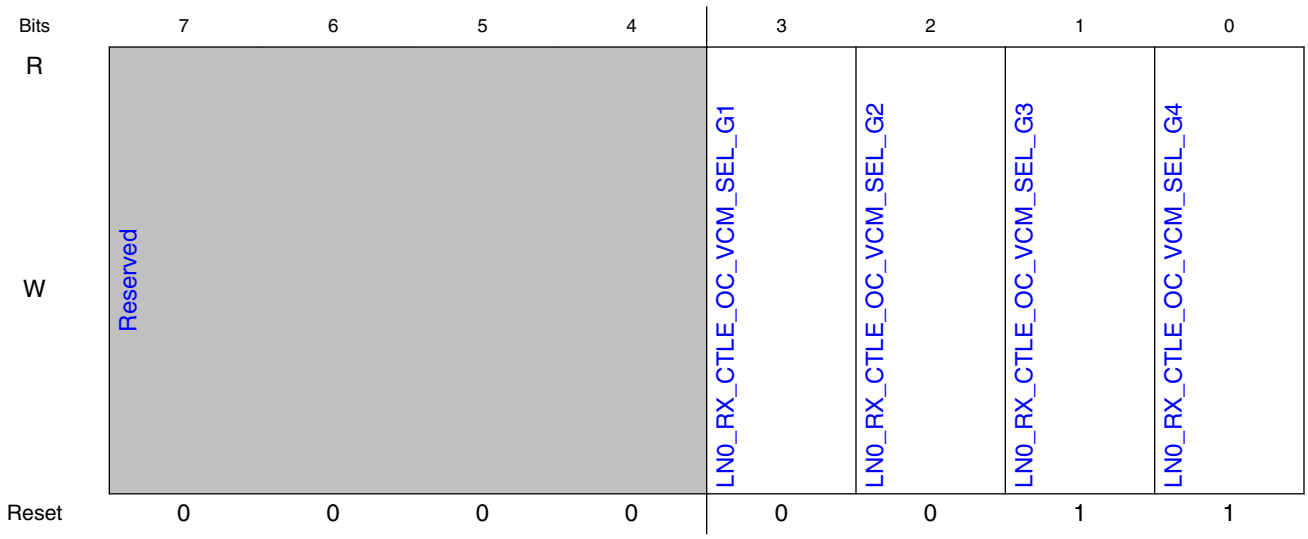
Field	Function
7-2 —	Reserved
1 LN0_OVRD_RX_CTLE_OC_EN	Override enable for rx_ctle_oc_en
0 LN0_RX_CTLE_OC_EN	RX CTLE offset calibration enable 0: Disable, 1: Enable

11.5.3.1.181 (TRSV_REG036)

11.5.3.1.181.1 Offset

Register	Offset
TRSV_REG036	4D8h

11.5.3.1.181.2 Diagram



11.5.3.1.181.3 Fields

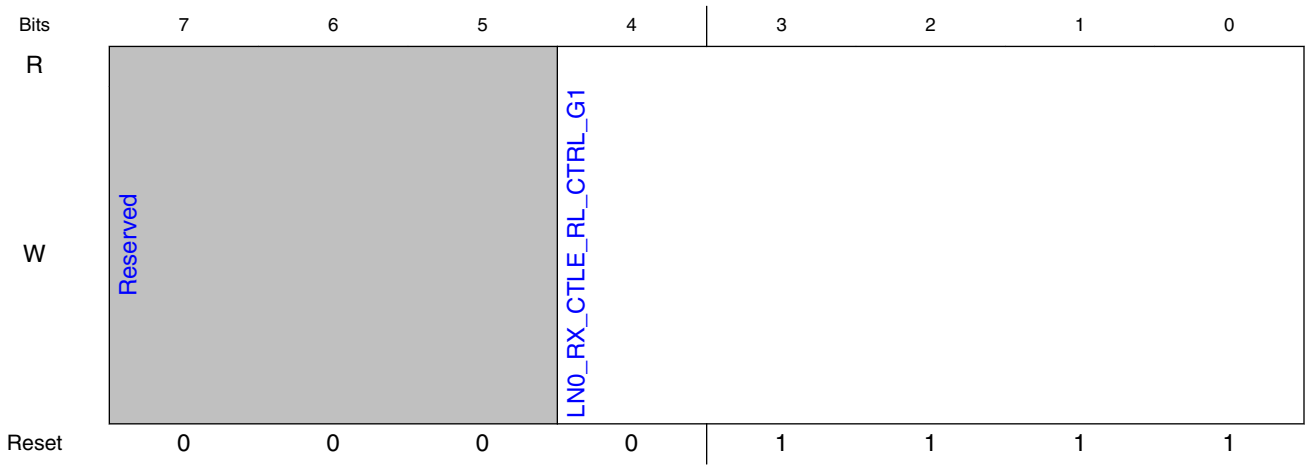
Field	Function
7-4 —	Reserved
3 LN0_RX_CTLE_OC_VCM_SEL_G1	[GEN1] RX CTLE input common-mode selection in offset calibration 0: GND, 1: VCM (~200mV)
2 LN0_RX_CTLE_OC_VCM_SEL_G2	[GEN2]
1 LN0_RX_CTLE_OC_VCM_SEL_G3	[GEN3]
0 LN0_RX_CTLE_OC_VCM_SEL_G4	[GEN4]

11.5.3.1.182 (TRSV_REG037)

11.5.3.1.182.1 Offset

Register	Offset
TRSV_REG037	4DCh

11.5.3.1.182.2 Diagram



11.5.3.1.182.3 Fields

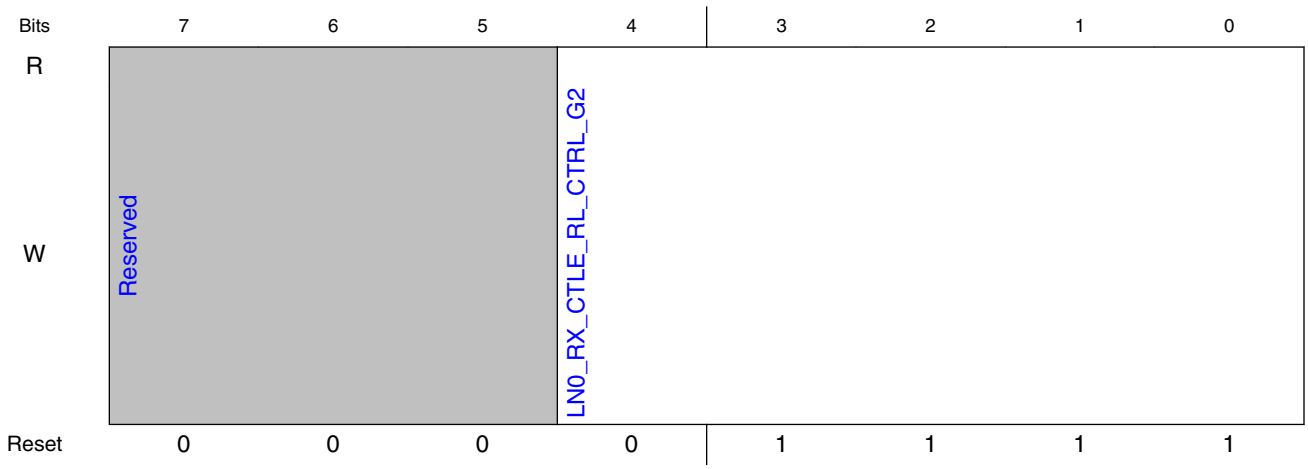
Field	Function
7-5 —	Reserved
4-0 LN0_RX_CTLE_RL_CTRL_G1	[GEN1] RX CTLE load resistance control for Gen1 0000: Max resistance, 1111: Min resistance 3'b001 for Gen1 and 2, 3'b011 for Gen3, and 3'b111 for Gen4 is recommended

11.5.3.1.183 (TRSV_REG038)

11.5.3.1.183.1 Offset

Register	Offset
TRSV_REG038	4E0h

11.5.3.1.183.2 Diagram



11.5.3.1.183.3 Fields

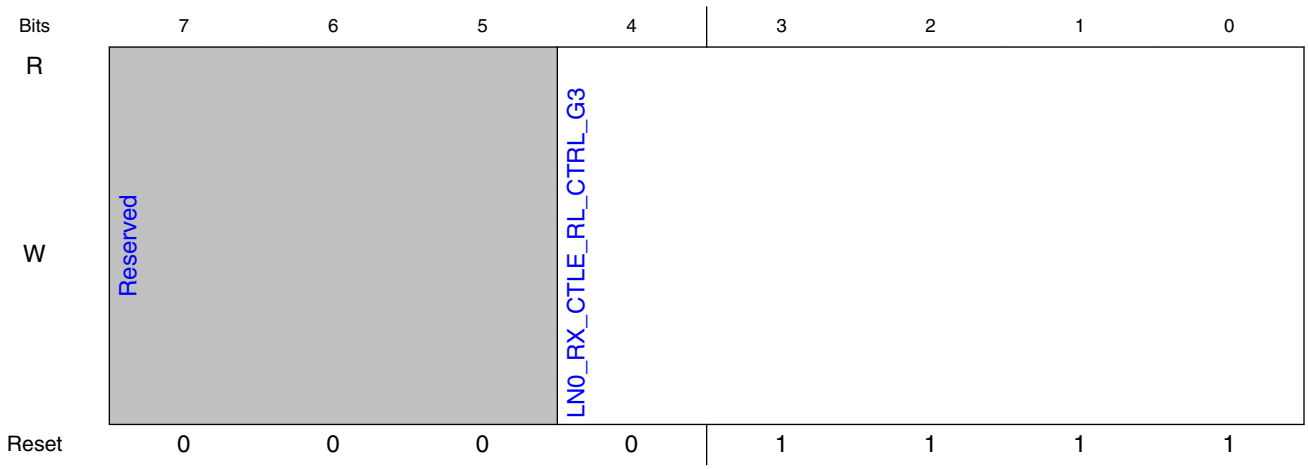
Field	Function
7-5 —	Reserved
4-0 LN0_RX_CTLE_RL_CTRL_G2	[GEN2]

11.5.3.1.184 (TRSV_REG039)

11.5.3.1.184.1 Offset

Register	Offset
TRSV_REG039	4E4h

11.5.3.1.184.2 Diagram



11.5.3.1.184.3 Fields

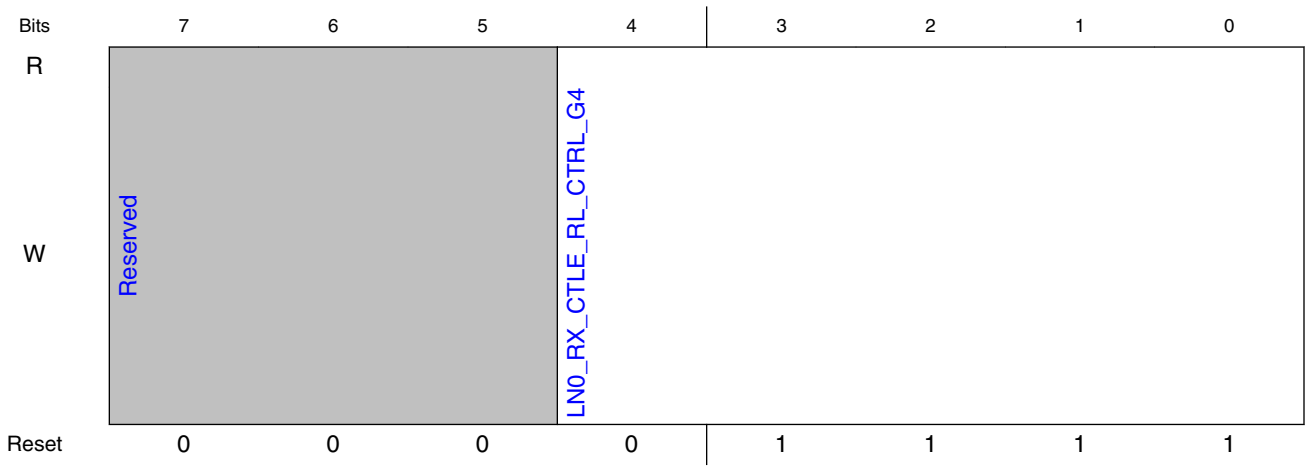
Field	Function
7-5 —	Reserved
4-0 LN0_RX_CTLE_RL_CTRL_G3	[GEN3]

11.5.3.1.185 (TRSV_REG03A)

11.5.3.1.185.1 Offset

Register	Offset
TRSV_REG03A	4E8h

11.5.3.1.185.2 Diagram



11.5.3.1.185.3 Fields

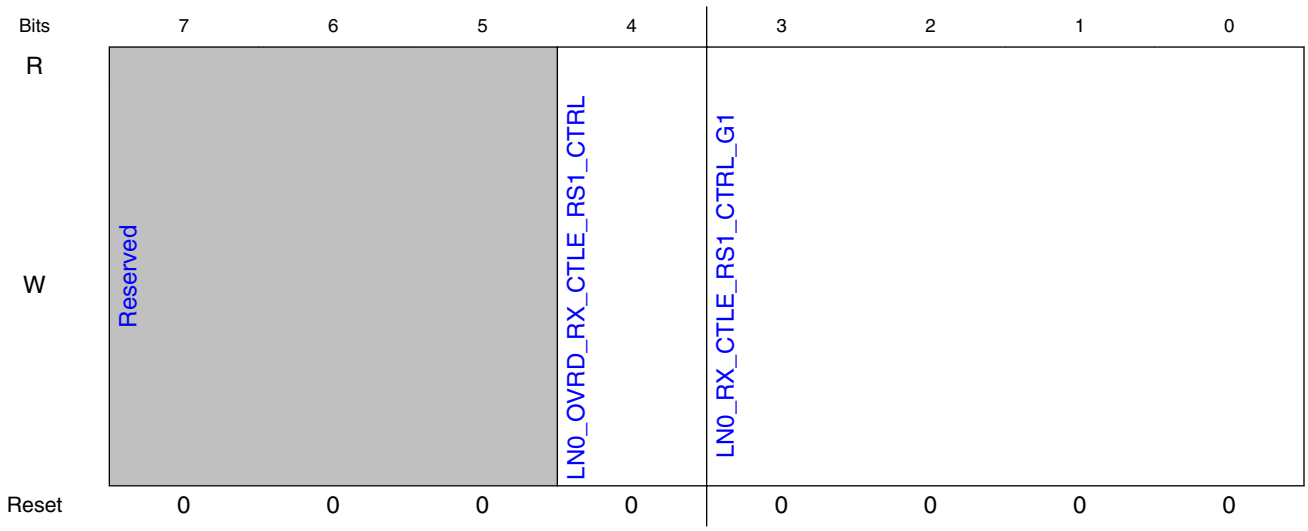
Field	Function
7-5 —	Reserved
4-0 LN0_RX_CTLE_RL_CTRL_G4	[GEN4]

11.5.3.1.186 (TRSV_REG03B)

11.5.3.1.186.1 Offset

Register	Offset
TRSV_REG03B	4ECh

11.5.3.1.186.2 Diagram



11.5.3.1.186.3 Fields

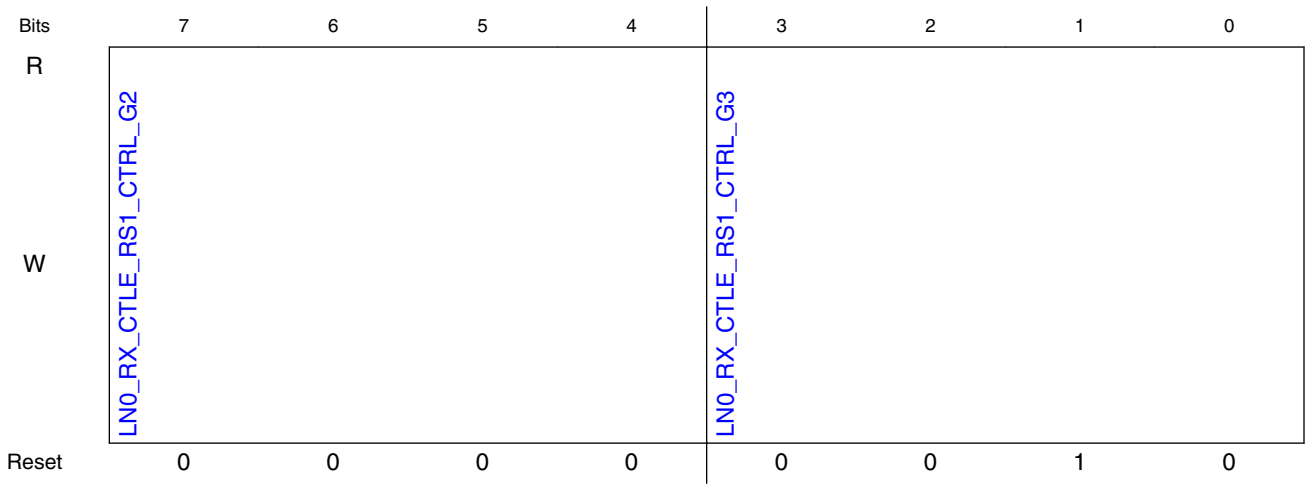
Field	Function
7-5 —	Reserved
4 LN0_OVRD_RX_CTLE_RS1_CTRL	Override enable for rx_ctle_rs1_ctrl_g1
3-0 LN0_RX_CTLE_RS1_CTRL_G1	[GEN1] RX CTLE 1st stage source series resistance control 000: Min resistance, … , 111: Max resistance

11.5.3.1.187 (TRSV_REG03C)

11.5.3.1.187.1 Offset

Register	Offset
TRSV_REG03C	4F0h

11.5.3.1.187.2 Diagram



11.5.3.1.187.3 Fields

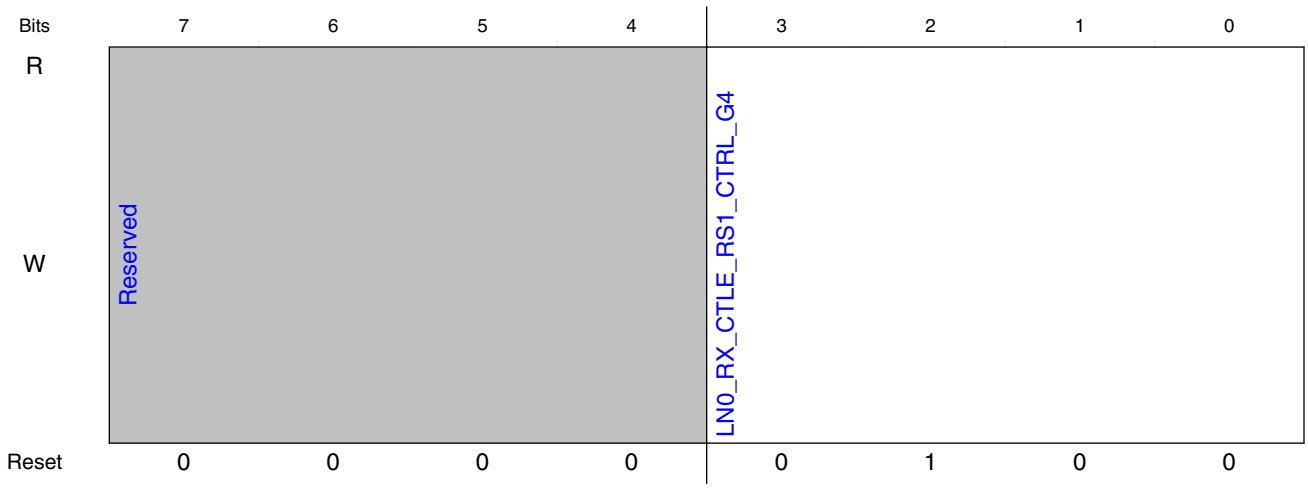
Field	Function
7-4 LN0_RX_CTLE_RS1_CTRL_G2	[GEN2]
3-0 LN0_RX_CTLE_RS1_CTRL_G3	[GEN3]

11.5.3.1.188 (TRSV_REG03D)

11.5.3.1.188.1 Offset

Register	Offset
TRSV_REG03D	4F4h

11.5.3.1.188.2 Diagram



11.5.3.1.188.3 Fields

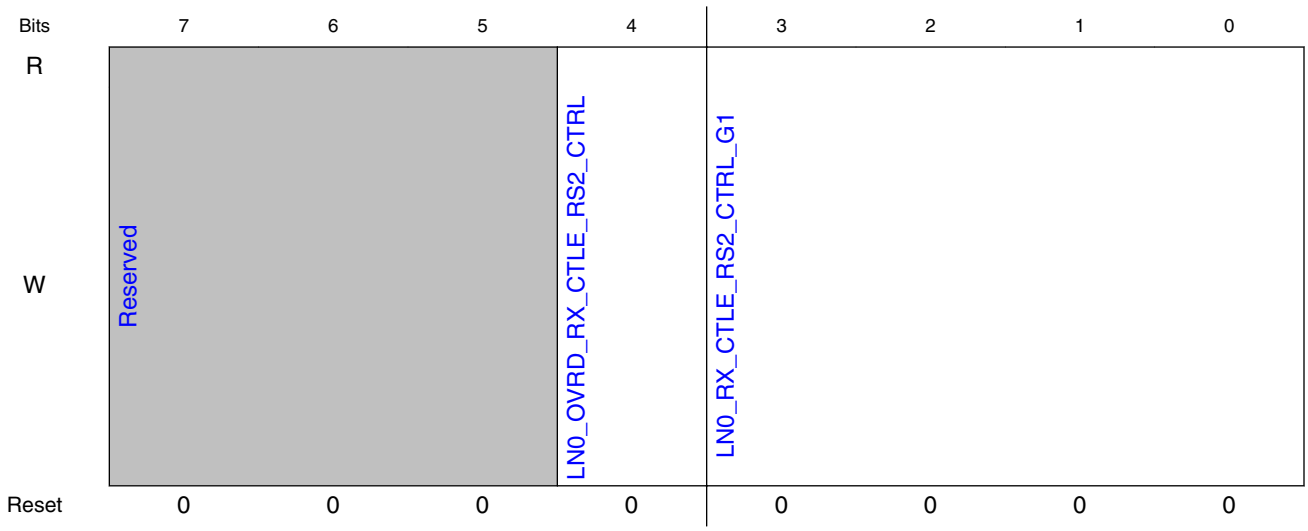
Field	Function
7-4 —	Reserved
3-0 LN0_RX_CTLE_RS1_CTRL_G4	[GEN4]

11.5.3.1.189 (TRSV_REG03E)

11.5.3.1.189.1 Offset

Register	Offset
TRSV_REG03E	4F8h

11.5.3.1.189.2 Diagram



11.5.3.1.189.3 Fields

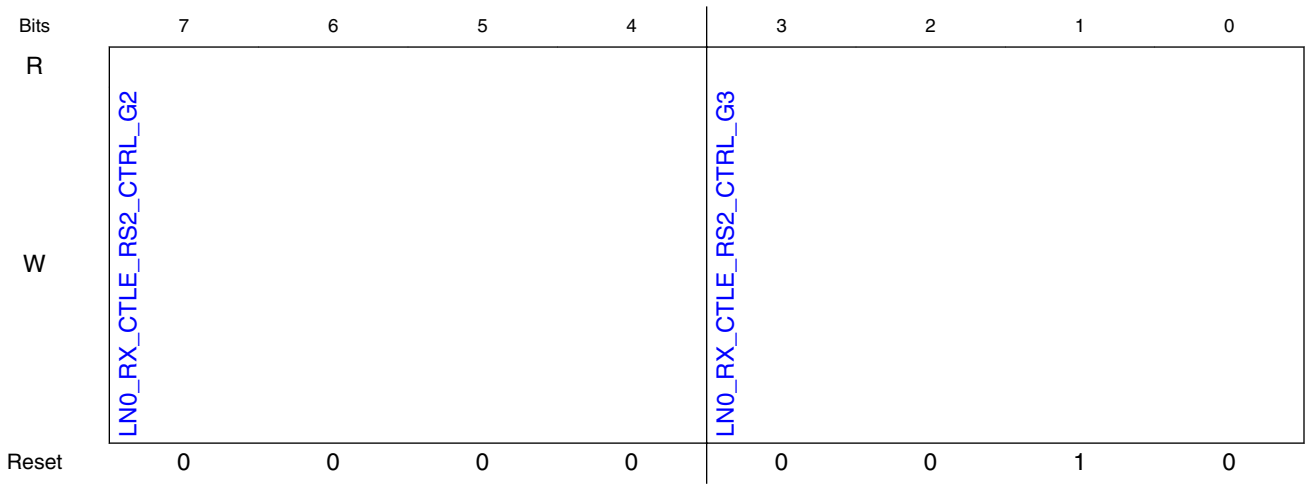
Field	Function
7-5 —	Reserved
4 LN0_OVRD_RX_CTLE_RS2_CTRL	Override enable for rx_ctle_rs2_ctrl_g1
3-0 LN0_RX_CTLE_RS2_CTRL_G1	[GEN1] RX CTLE 2nd stage source series resistance control

11.5.3.1.190 (TRSV_REG03F)

11.5.3.1.190.1 Offset

Register	Offset
TRSV_REG03F	4FCh

11.5.3.1.190.2 Diagram



11.5.3.1.190.3 Fields

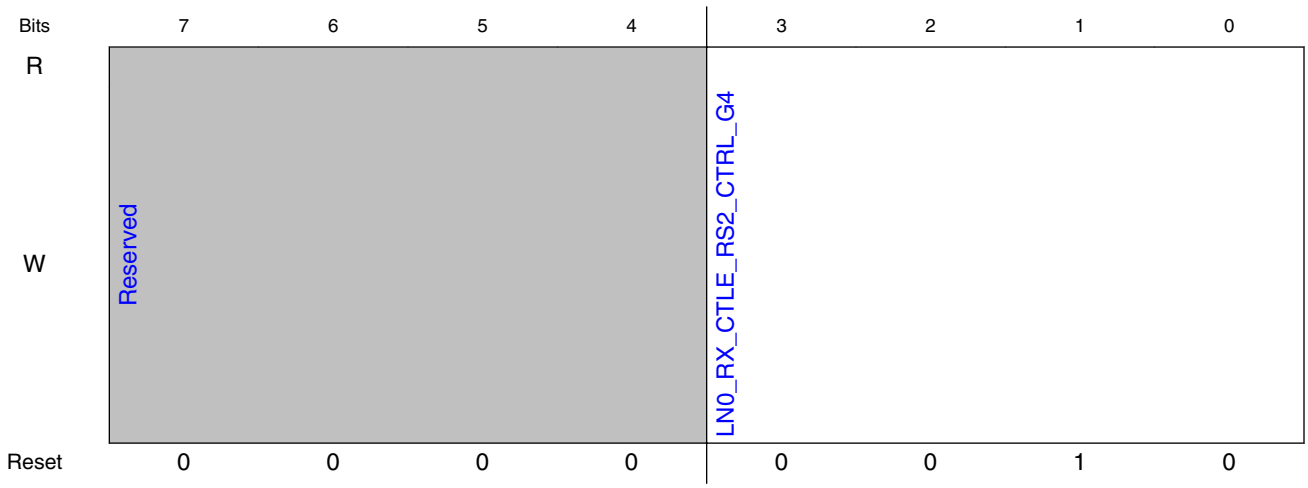
Field	Function
7-4 LN0_RX_CTLE_RS2_CTRL_G2	[GEN2]
3-0 LN0_RX_CTLE_RS2_CTRL_G3	[GEN3]

11.5.3.1.191 (TRSV_REG040)

11.5.3.1.191.1 Offset

Register	Offset
TRSV_REG040	500h

11.5.3.1.191.2 Diagram



11.5.3.1.191.3 Fields

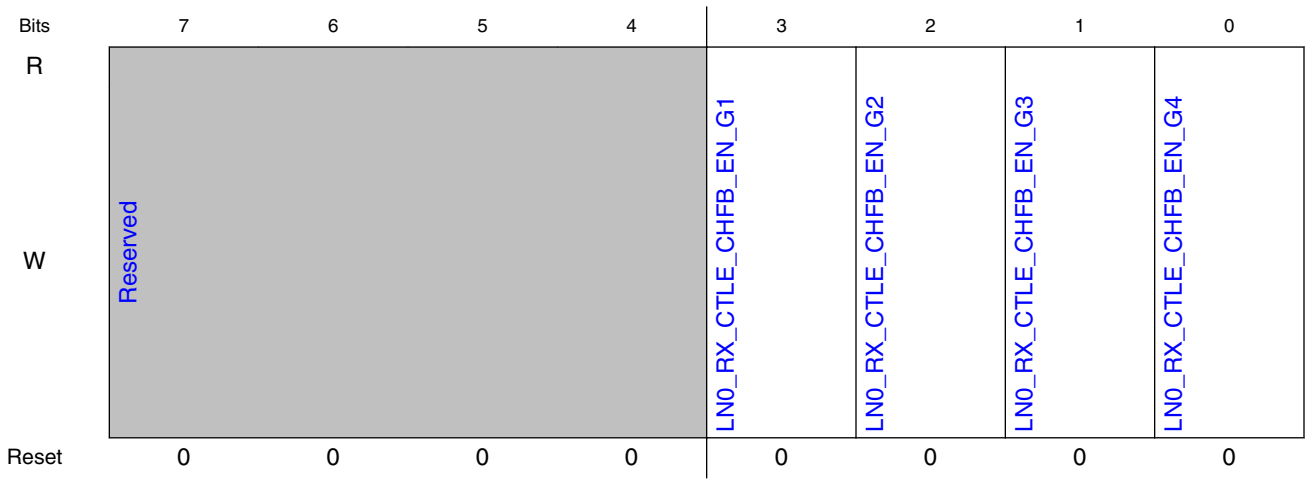
Field	Function
7-4 —	Reserved
3-0 LN0_RX_CTLE_RS2_CTRL_G4	[GEN4]

11.5.3.1.192 (TRSV_REG041)

11.5.3.1.192.1 Offset

Register	Offset
TRSV_REG041	504h

11.5.3.1.192.2 Diagram



11.5.3.1.192.3 Fields

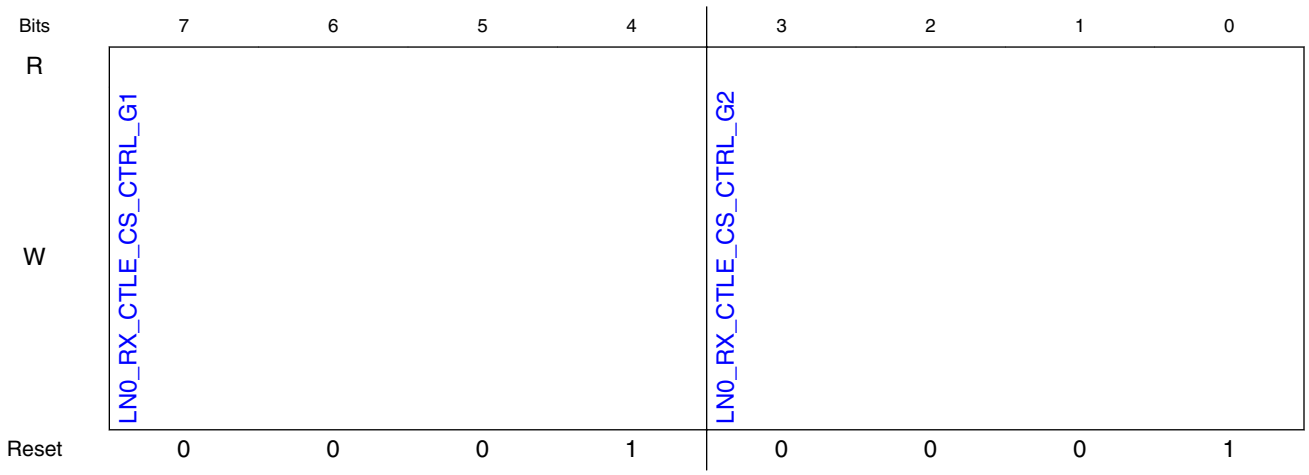
Field	Function
7-4 —	Reserved
3 LN0_RX_CTLE_CHFB_EN_G1	[GEN1] RX CTLE Cherry-Hooper feedback amplifier enable 0: Disable, 1: Enable
2 LN0_RX_CTLE_CHFB_EN_G2	[GEN2]
1 LN0_RX_CTLE_CHFB_EN_G3	[GEN3]
0 LN0_RX_CTLE_CHFB_EN_G4	[GEN4]

11.5.3.1.193 (TRSV_REG042)

11.5.3.1.193.1 Offset

Register	Offset
TRSV_REG042	508h

11.5.3.1.193.2 Diagram



11.5.3.1.193.3 Fields

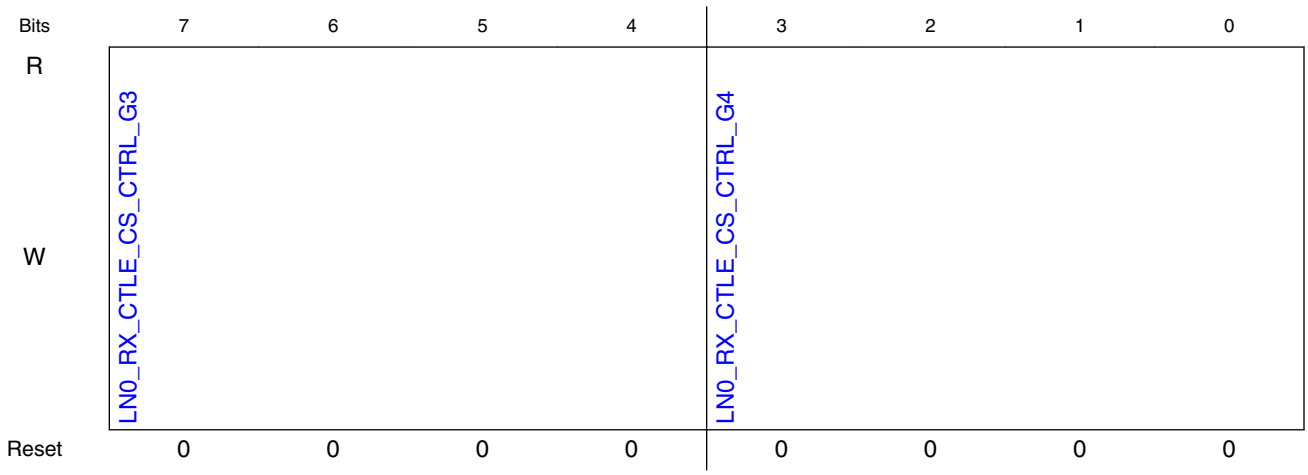
Field	Function
7-4 LN0_RX_CTLE_CS_CTRL_G1	[GEN1] CTLE capacitance control. 4'h0=Gen4,3 4'h3=Gen2 4'h7=Gen1
3-0 LN0_RX_CTLE_CS_CTRL_G2	[GEN2]

11.5.3.1.194 (TRSV_REG043)

11.5.3.1.194.1 Offset

Register	Offset
TRSV_REG043	50Ch

11.5.3.1.194.2 Diagram



11.5.3.1.194.3 Fields

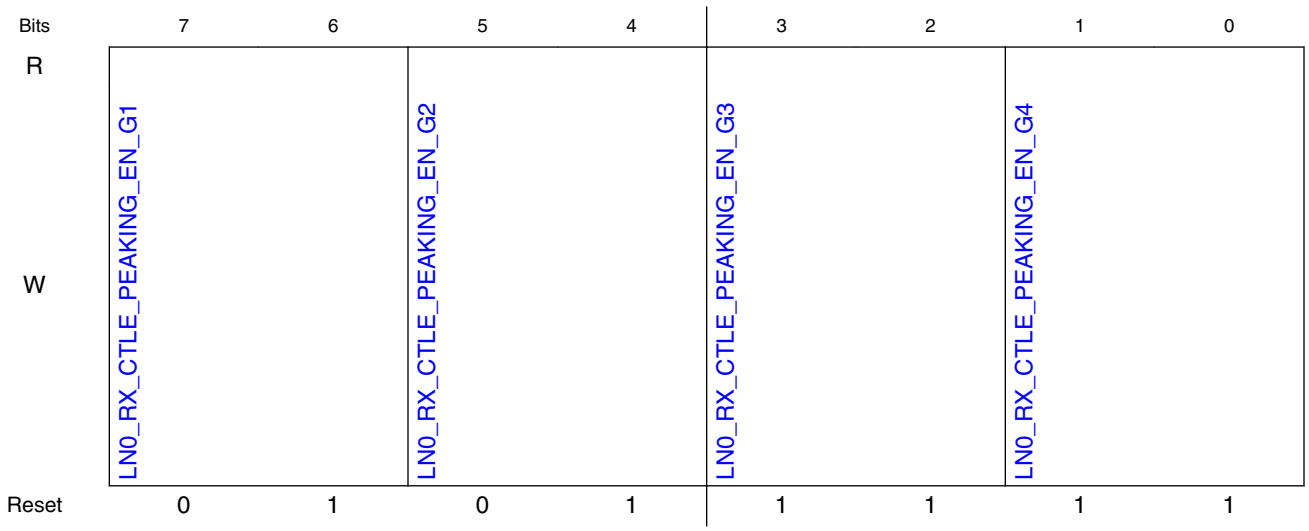
Field	Function
7-4 LN0_RX_CTLE_CS_CTRL_G3	[GEN3]
3-0 LN0_RX_CTLE_CS_CTRL_G4	[GEN4]

11.5.3.1.195 (TRSV_REG044)

11.5.3.1.195.1 Offset

Register	Offset
TRSV_REG044	510h

11.5.3.1.195.2 Diagram



11.5.3.1.195.3 Fields

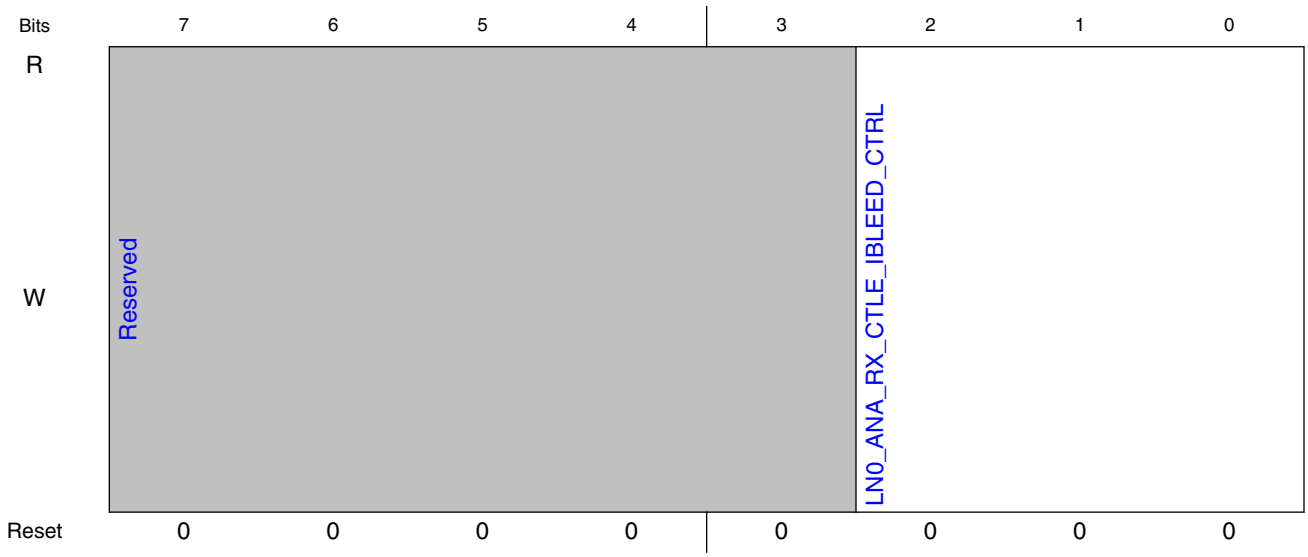
Field	Function
7-6 LN0_RX_CTLE_PEAKING_EN_G1	[GEN1] RX CTLE stage enable for Gen1 [0]: 1st stage, [1]: 2nd stage, [3]=3rd stage 0: Disable, 1: Enable
5-4 LN0_RX_CTLE_PEAKING_EN_G2	[GEN2]
3-2 LN0_RX_CTLE_PEAKING_EN_G3	[GEN3]
1-0 LN0_RX_CTLE_PEAKING_EN_G4	[GEN4]

11.5.3.1.196 (TRSV_REG045)

11.5.3.1.196.1 Offset

Register	Offset
TRSV_REG045	514h

11.5.3.1.196.2 Diagram



11.5.3.1.196.3 Fields

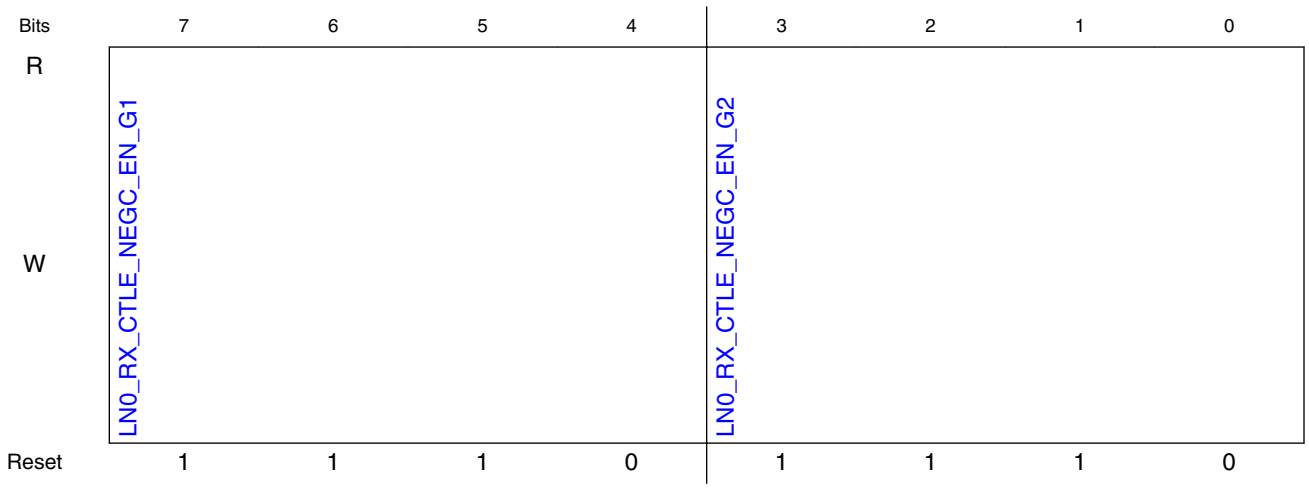
Field	Function
7-3 —	Reserved
2-0 LN0_ANA_RX_CTLE_IBLEED_CTRL	RX CTLE bleeder current control 000: No bleeder current, … , 111: Max bleeder current

11.5.3.1.197 (TRSV_REG046)

11.5.3.1.197.1 Offset

Register	Offset
TRSV_REG046	518h

11.5.3.1.197.2 Diagram



11.5.3.1.197.3 Fields

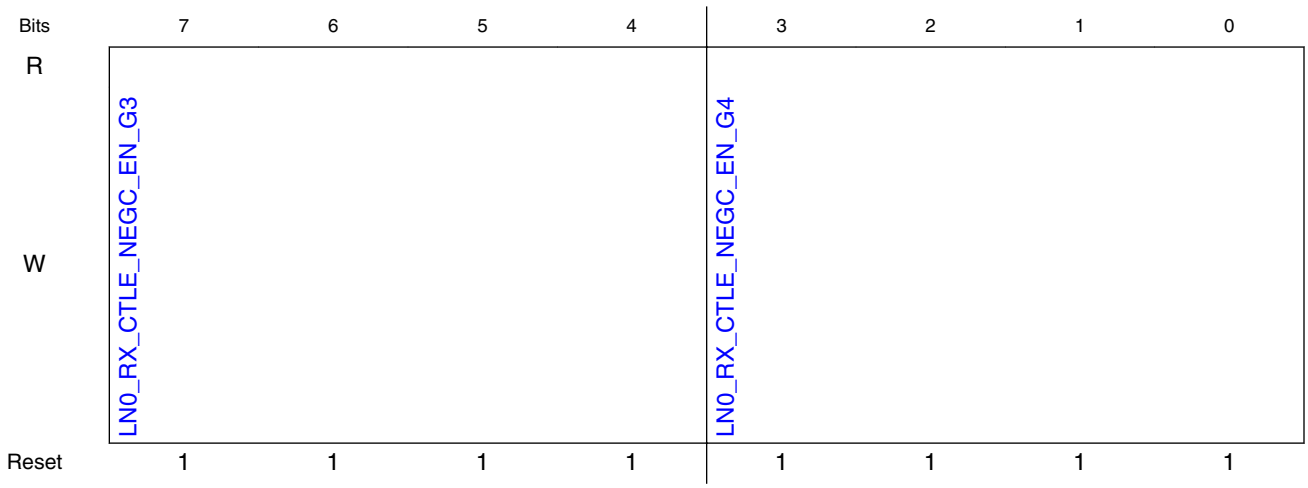
Field	Function
7-4 LN0_RX_CTLE_NEGC_EN_G1	[GEN1] RX CTLE negative-C enable 0: Disable, 1: Enable
3-0 LN0_RX_CTLE_NEGC_EN_G2	[GEN2]

11.5.3.1.198 (TRSV_REG047)

11.5.3.1.198.1 Offset

Register	Offset
TRSV_REG047	51Ch

11.5.3.1.198.2 Diagram



11.5.3.1.198.3 Fields

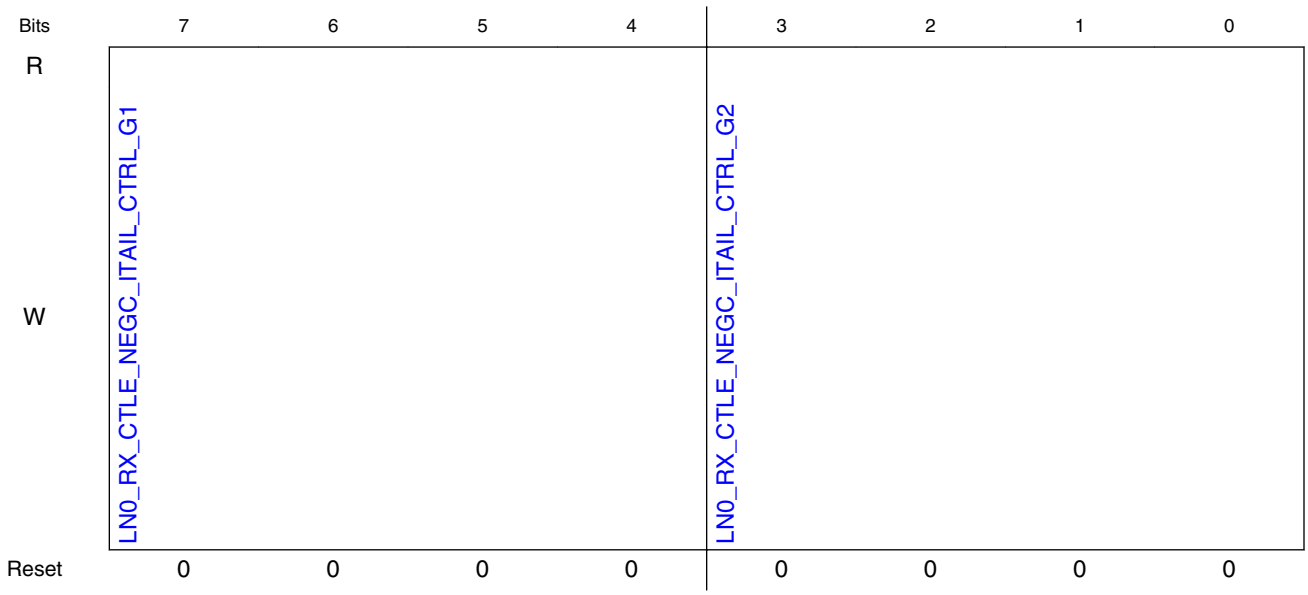
Field	Function
7-4 LN0_RX_CTLE_NEGC_EN_G3	[GEN3]
3-0 LN0_RX_CTLE_NEGC_EN_G4	[GEN4]

11.5.3.1.199 (TRSV_REG048)

11.5.3.1.199.1 Offset

Register	Offset
TRSV_REG048	520h

11.5.3.1.199.2 Diagram



11.5.3.1.199.3 Fields

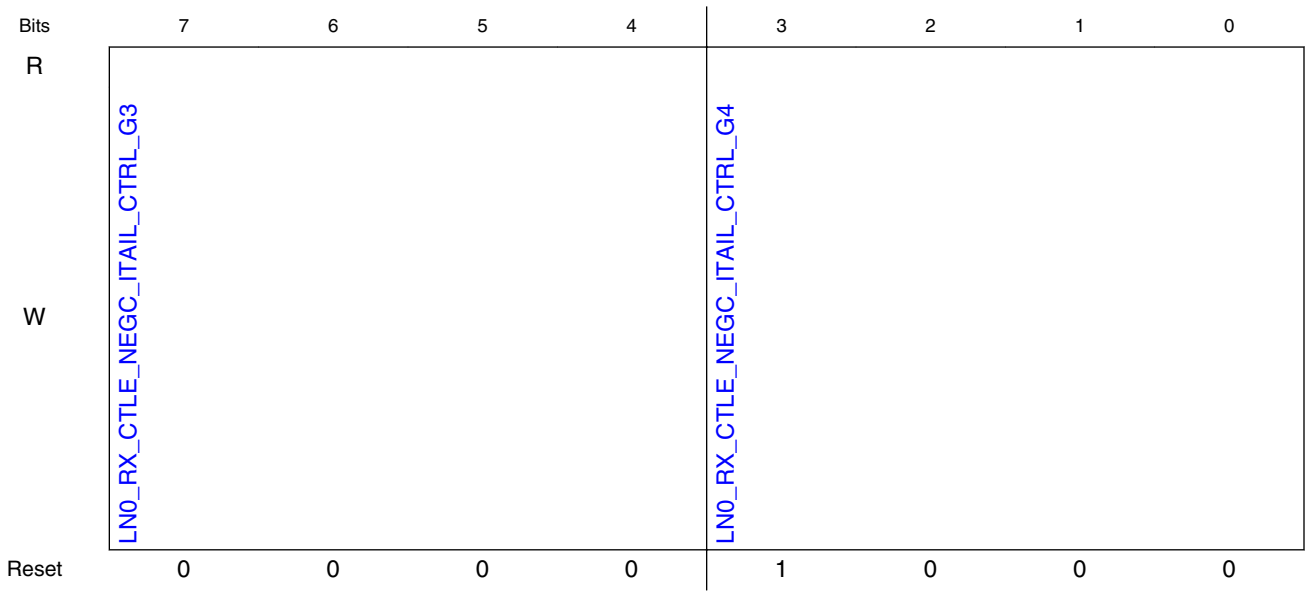
Field	Function
7-4 LN0_RX_CTLE_NEGC_ITAIL_CTRL_G1	[GEN1] RX CTLE negative-C tail current control 0000:, 1111:
3-0 LN0_RX_CTLE_NEGC_ITAIL_CTRL_G2	[GEN2]

11.5.3.1.200 (TRSV_REG049)

11.5.3.1.200.1 Offset

Register	Offset
TRSV_REG049	524h

11.5.3.1.200.2 Diagram



11.5.3.1.200.3 Fields

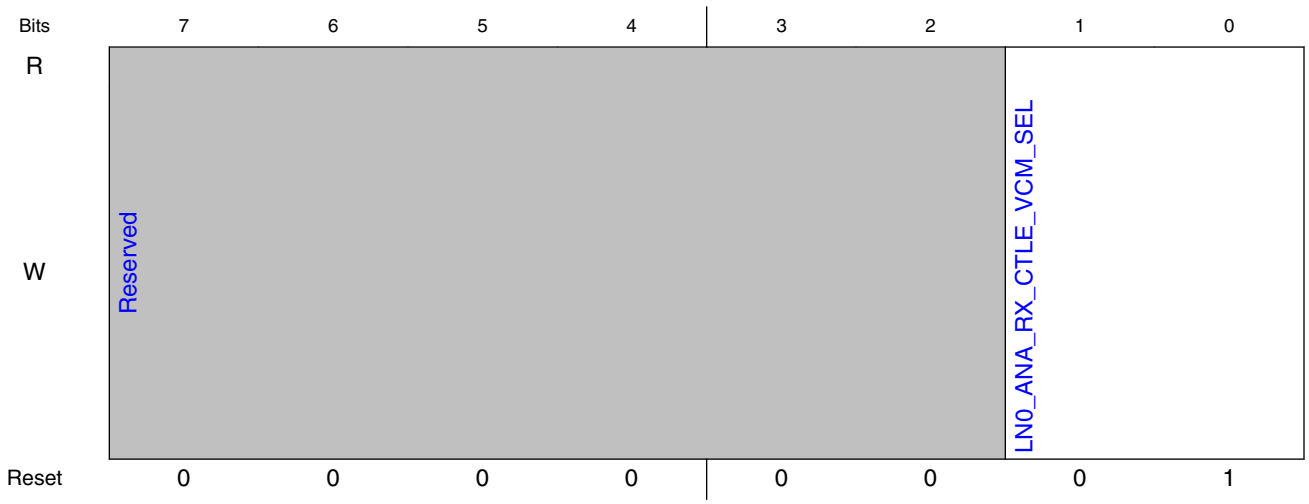
Field	Function
7-4 LN0_RX_CTLE_NEGC_ITAIL_CTRL_G3	[GEN3]
3-0 LN0_RX_CTLE_NEGC_ITAIL_CTRL_G4	[GEN4]

11.5.3.1.201 (TRSV_REG04A)

11.5.3.1.201.1 Offset

Register	Offset
TRSV_REG04A	528h

11.5.3.1.201.2 Diagram



11.5.3.1.201.3 Fields

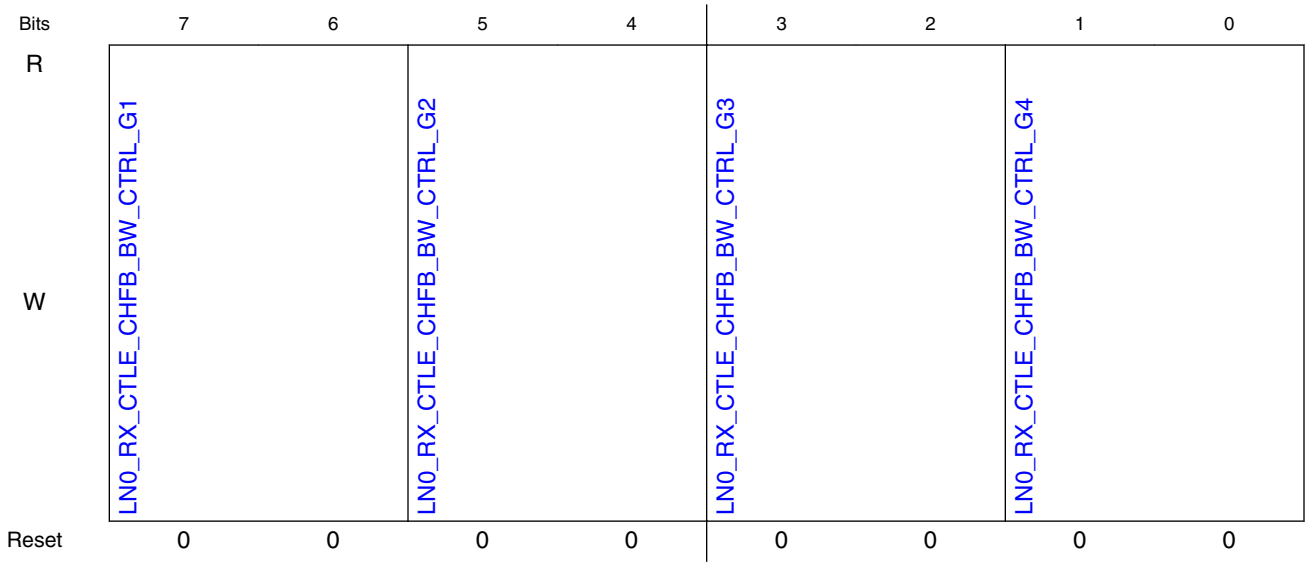
Field	Function
7-2 —	Reserved
1-0 LN0_ANA_RX_CTLE_VCM_SEL	RX AFE (CTLE output) common-mode voltage selection 00=150mV, 01: 180mV, 10: 210mV, 11: 230mV

11.5.3.1.202 (TRSV_REG04B)

11.5.3.1.202.1 Offset

Register	Offset
TRSV_REG04B	52Ch

11.5.3.1.202.2 Diagram



11.5.3.1.202.3 Fields

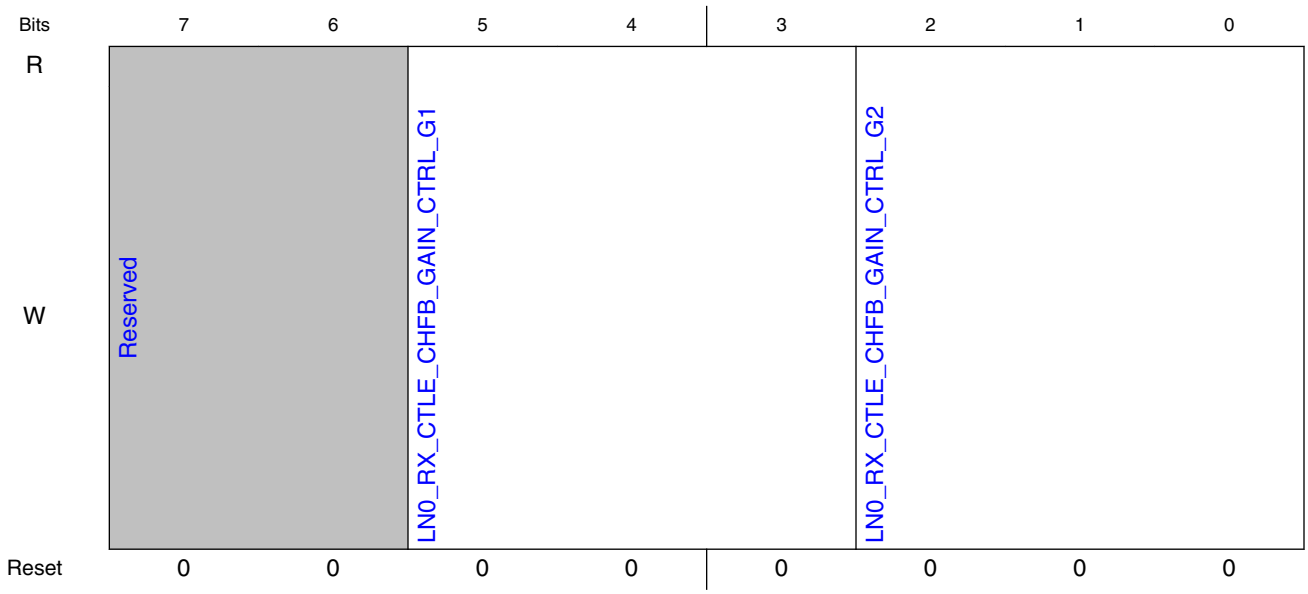
Field	Function
7-6 LN0_RX_CTLE_CHFB_BW_CTRL_G1	[GEN1] RX CTLE Cherry-Hooper feedback amplifier bandwidth control 00: , 01: , 10: , 11
5-4 LN0_RX_CTLE_CHFB_BW_CTRL_G2	[GEN2]
3-2 LN0_RX_CTLE_CHFB_BW_CTRL_G3	[GEN3]
1-0 LN0_RX_CTLE_CHFB_BW_CTRL_G4	[GEN4]

11.5.3.1.203 (TRSV_REG04C)

11.5.3.1.203.1 Offset

Register	Offset
TRSV_REG04C	530h

11.5.3.1.203.2 Diagram



11.5.3.1.203.3 Fields

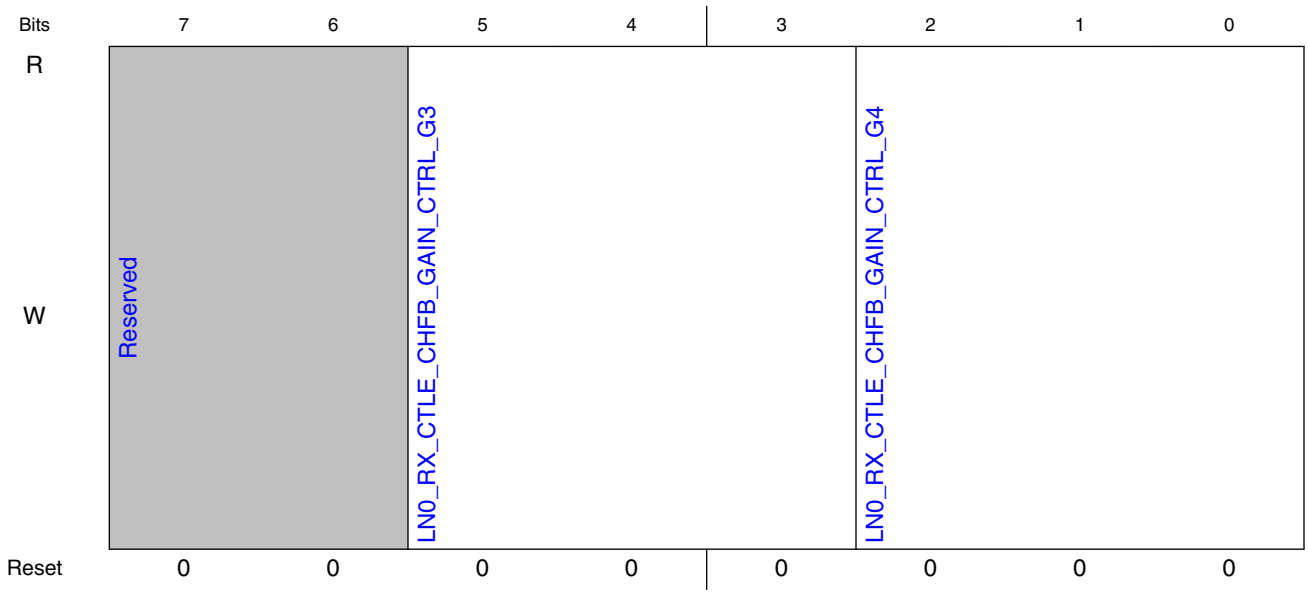
Field	Function
7-6 —	Reserved
5-3 LN0_RX_CTLE_CHFB_GAIN_CTRL_G1	[GEN1] RX CTLE Cherry-Hooper feedback amplifier gain control 000: , 001: , 010: , 011, 100: , 101: , 110: , 111,
2-0 LN0_RX_CTLE_CHFB_GAIN_CTRL_G2	[GEN2]

11.5.3.1.204 (TRSV_REG04D)

11.5.3.1.204.1 Offset

Register	Offset
TRSV_REG04D	534h

11.5.3.1.204.2 Diagram



11.5.3.1.204.3 Fields

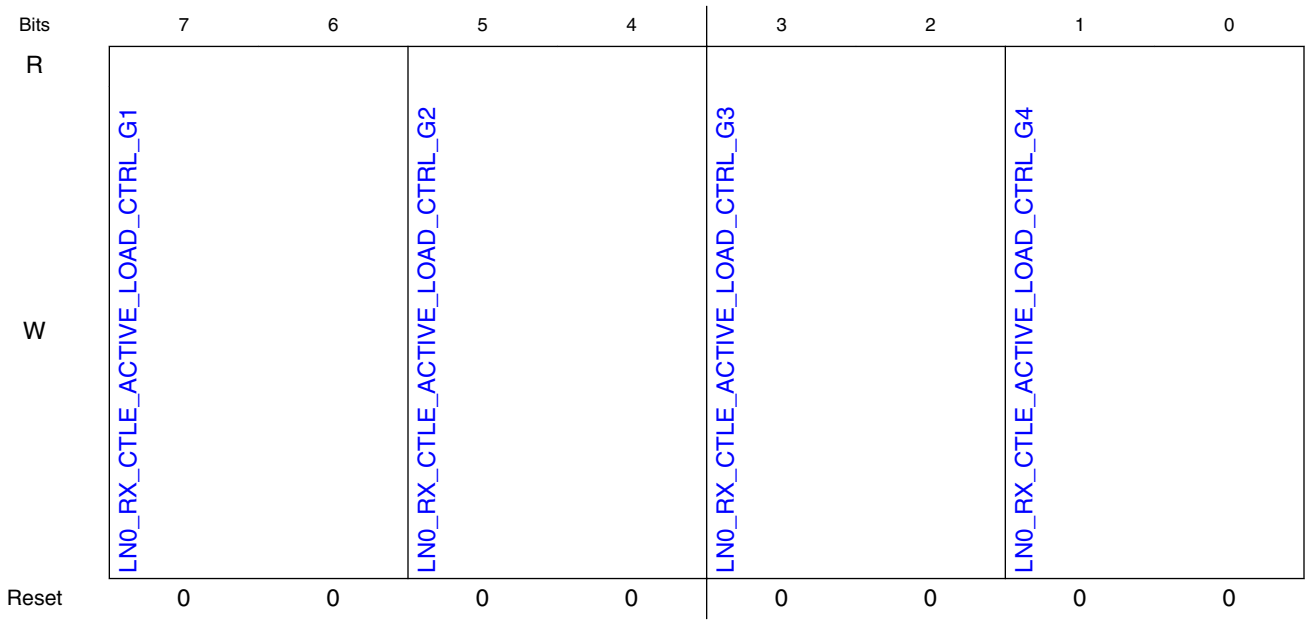
Field	Function
7-6 —	Reserved
5-3 LN0_RX_CTLE_CHFB_GAIN_CTRL_G3	[GEN3]
2-0 LN0_RX_CTLE_CHFB_GAIN_CTRL_G4	[GEN4]

11.5.3.1.205 (TRSV_REG04E)

11.5.3.1.205.1 Offset

Register	Offset
TRSV_REG04E	538h

11.5.3.1.205.2 Diagram



11.5.3.1.205.3 Fields

Field	Function
7-6 LN0_RX_CTLE_ACTIVE_LOAD_CTRL_G1	[GEN1] RX CTLE active load control 00: , 01: , 10: , 11
5-4 LN0_RX_CTLE_ACTIVE_LOAD_CTRL_G2	[GEN2]
3-2 LN0_RX_CTLE_ACTIVE_LOAD_CTRL_G3	[GEN3]
1-0 LN0_RX_CTLE_ACTIVE_LOAD_CTRL_G4	[GEN4]

11.5.3.1.206 (TRSV_REG04F)

11.5.3.1.206.1 Offset

Register	Offset
TRSV_REG04F	53Ch

11.5.3.1.206.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_ANA_RX_CTLE_VGA_CTRL			LN0_ANA_RX_CTLE_VREG_SEL				LN0_ANA_RX_CTLE_PTAT_EN
W								
Reset	1	1	1	0	0	0	0	0

11.5.3.1.206.3 Fields

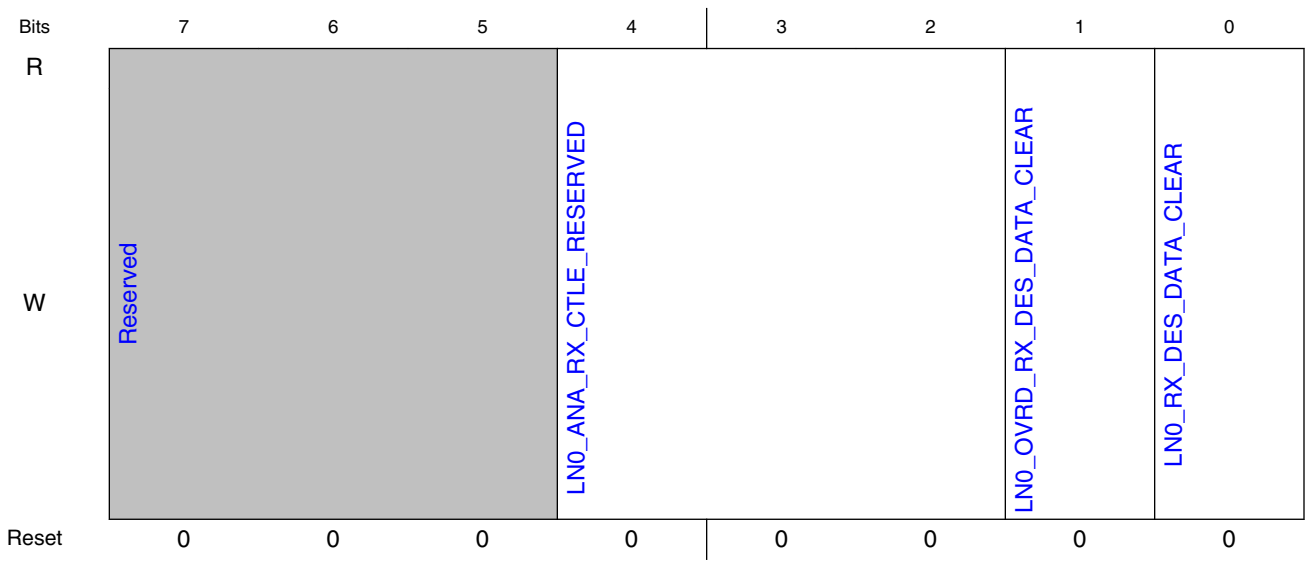
Field	Function
7-5 LN0_ANA_RX_CTLE_VGA_CTRL	RX CTLE stage3 gain control 000:Max. gain , 111: Max. gain
4-1 LN0_ANA_RX_CTLE_VREG_SEL	RX CTLE voltage regulator output voltage 0000: 820mV, 1111: Max voltage
0 LN0_ANA_RX_CTLE_PTAT_EN	RX CTLE PTAT current enable 0: Disable, 1: Enable

11.5.3.1.207 (TRSV_REG050)

11.5.3.1.207.1 Offset

Register	Offset
TRSV_REG050	540h

11.5.3.1.207.2 Diagram



11.5.3.1.207.3 Fields

Field	Function
7-5 —	Reserved
4-2 LN0_ANA_RX_CTLE_RESERVED	Reserved port
1 LN0_OVRD_RX_DES_DATA_CLEAR	Override enable for rx_des_data_clear
0 LN0_RX_DES_DATA_CLEAR	RX deserializer data clear to prevent garbage data 0: Normal operation, 1: Clear (all 0)

11.5.3.1.208 (TRSV_REG051)

11.5.3.1.208.1 Offset

Register	Offset
TRSV_REG051	544h

11.5.3.1.208.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_RX_DES_DATA_WIDTH_SEL_G1		LN0_RX_DES_DATA_WIDTH_SEL_G2		LN0_RX_DES_DATA_WIDTH_SEL_G3		LN0_RX_DES_DATA_WIDTH_SEL_G4	
W								
Reset	0	0	0	0	0	1	0	1

11.5.3.1.208.3 Fields

Field	Function
7-6 LN0_RX_DES_DATA_WIDTH_SEL_G1	[GEN1] RX deserializer data width selection 00: 2:40 mode, 01: 2:32 mode, 10: 2:20 mode, Others: N/A
5-4 LN0_RX_DES_DATA_WIDTH_SEL_G2	[GEN2]
3-2 LN0_RX_DES_DATA_WIDTH_SEL_G3	[GEN3]
1-0	[GEN4]

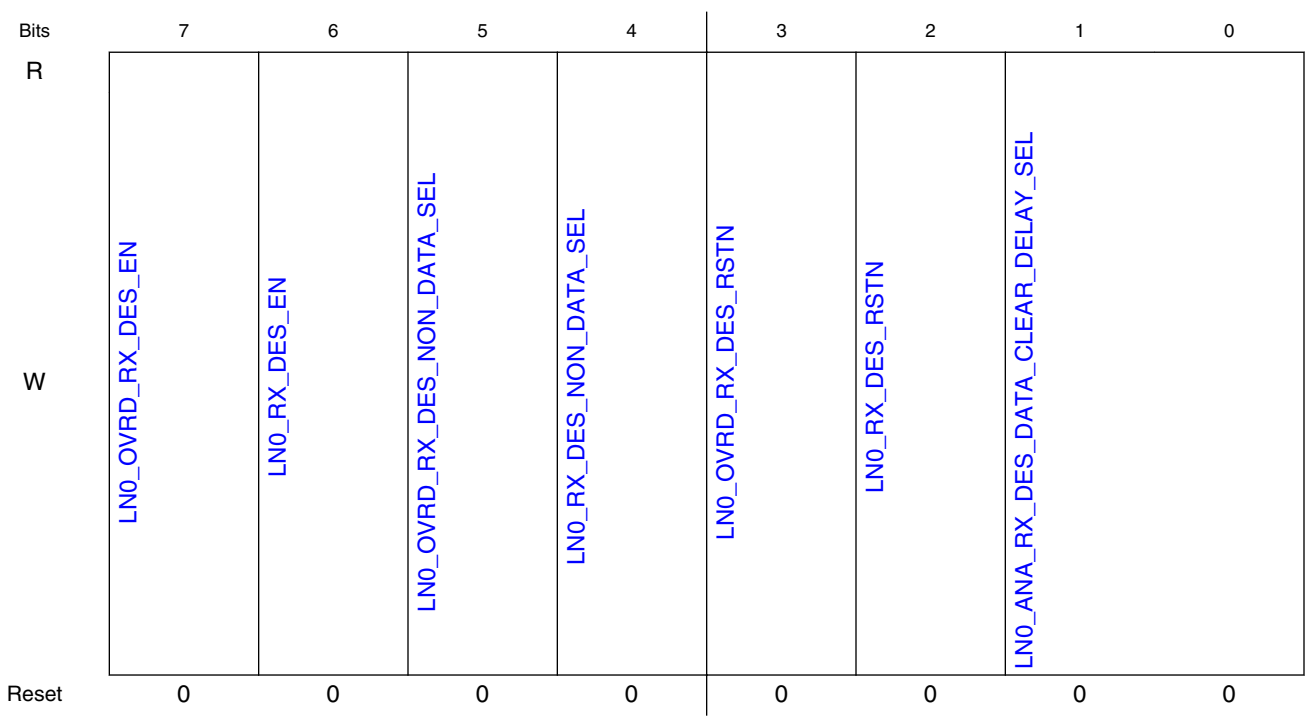
Field	Function
LN0_RX_DES_DATA_WIDTH_SEL_G4	

11.5.3.1.209 (TRSV_REG052)

11.5.3.1.209.1 Offset

Register	Offset
TRSV_REG052	548h

11.5.3.1.209.2 Diagram



11.5.3.1.209.3 Fields

Field	Function
7 LN0_OVRD_RX_DES_EN	Override enable for rx_des_en

Table continues on the next page...

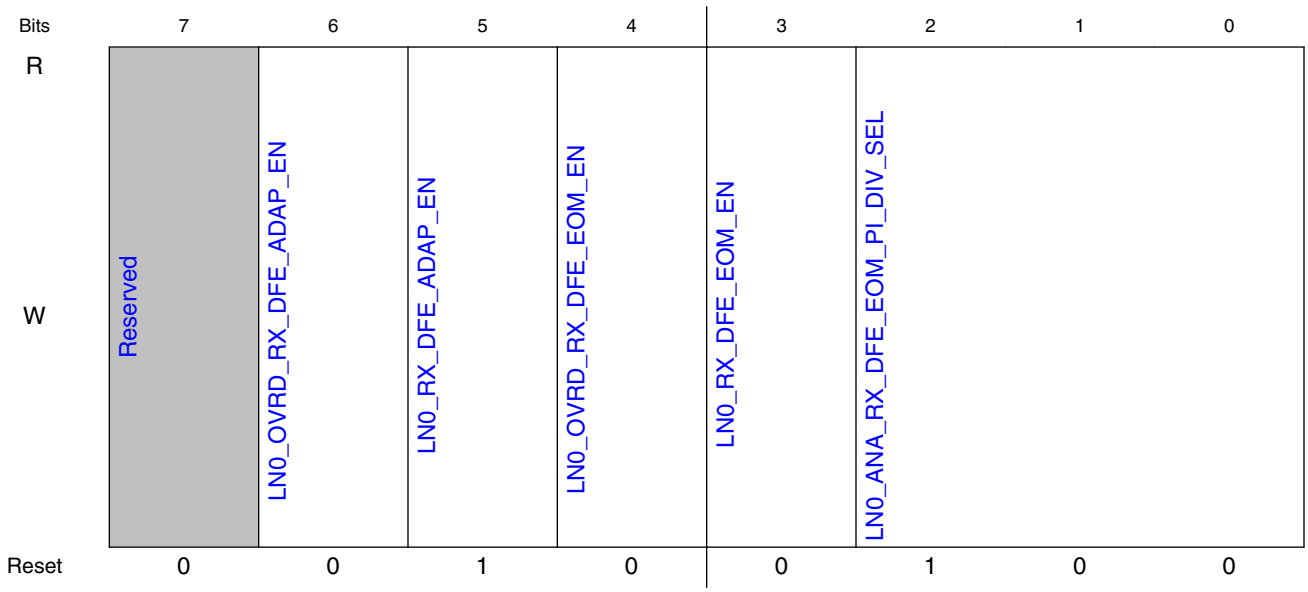
Field	Function
6 LN0_RX_DES_EN	RX deserializer enable 0: Disable, 1: Enable
5 LN0_OVRD_RX_DES_NON_DATA_SEL	Override enable for rx_des_non_data_sel
4 LN0_RX_DES_NON_DATA_SEL	RX deserializer non-data selection for edge/error sampler calibration 0: Error, 1: Edge
3 LN0_OVRD_RX_DES_RSTN	Override enable for rx_des_rstn
2 LN0_RX_DES_RSTN	RX deserializer reset 0: Reset, 1: Released
1-0 LN0_ANA_RX_DES_DATA_CLEAR_DELAY_SEL	

11.5.3.1.210 (TRSV_REG053)

11.5.3.1.210.1 Offset

Register	Offset
TRSV_REG053	54Ch

11.5.3.1.210.2 Diagram



11.5.3.1.210.3 Fields

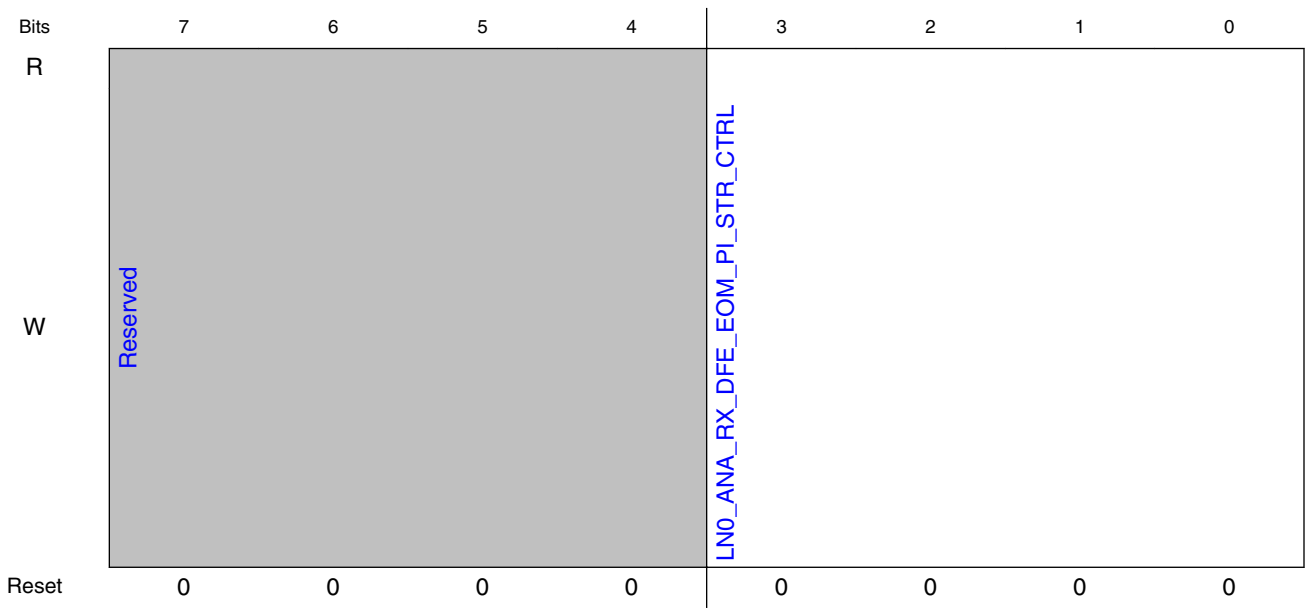
Field	Function
7 —	Reserved
6 LN0_OVRD_RX_DFE_ADAP_EN	Override enable for rx_dfe_adap_en
5 LN0_RX_DFE_ADAP_EN	RX DFE adaptation path enable. Only one of i_rx_dfe_adap_en and i_rx_eom_en should be "1". 0: Disable, 1: Enable
4 LN0_OVRD_RX_DFE_EOM_EN	Override enable for rx_dfe_eom_en
3 LN0_RX_DFE_EOM_EN	RX EOM enable 0: Disable, 1: Enable
2-0 LN0_ANA_RX_DFE_EOM_PI_DIV_SEL	Clock divider control before RX EOM phase interpolator 000 : div8, 001 : div4, 01x : div2, 1xx: div1

11.5.3.1.211 (TRSV_REG054)

11.5.3.1.211.1 Offset

Register	Offset
TRSV_REG054	550h

11.5.3.1.211.2 Diagram



11.5.3.1.211.3 Fields

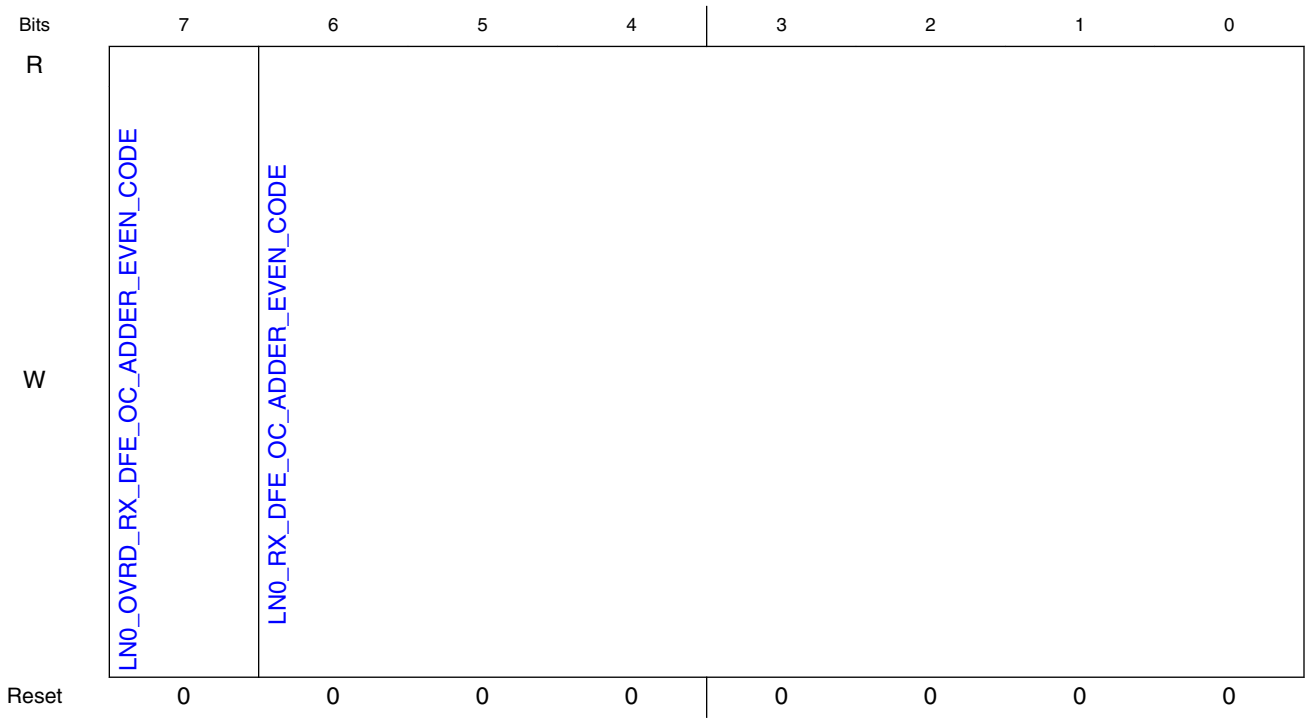
Field	Function
7-4 —	Reserved
3-0 LN0_ANA_RX_DFE_EOM_PI_STR_CTRL	RX EOM PI drive strengh in pre-buffer stage 0000: slower slope … 1111: steeper slope

11.5.3.1.212 (TRSV_REG055)

11.5.3.1.212.1 Offset

Register	Offset
TRSV_REG055	554h

11.5.3.1.212.2 Diagram



11.5.3.1.212.3 Fields

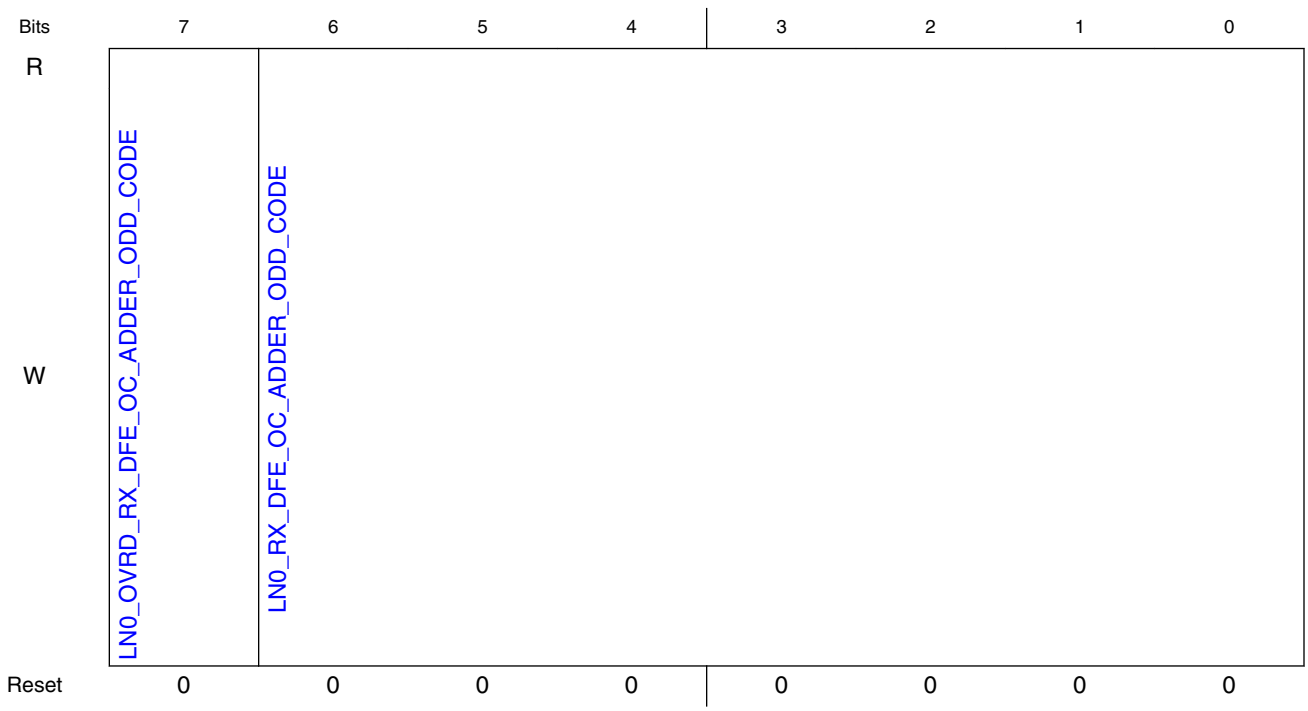
Field	Function
7 LN0_OVRD_RX_DFE_OC_ADDER_EVEN_CODE	Override enable for rx_dfe_oc_adder_even_code
6-0 LN0_RX_DFE_OC_ADDER_EVEN_CODE	RX DFE even data path offset calibration code

11.5.3.1.213 (TRSV_REG056)

11.5.3.1.213.1 Offset

Register	Offset
TRSV_REG056	558h

11.5.3.1.213.2 Diagram



11.5.3.1.213.3 Fields

Field	Function
7 LN0_OVRD_RX_DFE_OC_ADDER_ODD_CODE	Override enable for rx_dfe_oc_adder_odd_code
6-0 LN0_RX_DFE_OC_ADDER_ODD_CODE	RX DFE odd data path offset calibration code

11.5.3.1.214 (TRSV_REG057)

11.5.3.1.214.1 Offset

Register	Offset
TRSV_REG057	55Ch

11.5.3.1.214.2 Diagram

Bits	7	6	5	4	3	2	1	0
R								
W	LN0_OVRD_RX_DFE_OC_DAC_ADDER_EVEN_CODE	LN0_RX_DFE_OC_DAC_ADDER_EVEN_CODE	LN0_OVRD_RX_DFE_OC_DAC_ADDER_ODD_CODE	LN0_RX_DFE_OC_DAC_ADDER_ODD_CODE	LN0_OVRD_RX_DFE_OC_DAC_EDGE_EVEN_CODE	LN0_RX_DFE_OC_DAC_EDGE_EVEN_CODE		
Reset	0	0	0	0	0	0	0	0

11.5.3.1.214.3 Fields

Field	Function
7 LN0_OVRD_RX_DFE_OC_DAC_ADDER_EVEN_CODE	Override enable for rx_dfe_oc_dac_adder_even_code
6	Fine control of zero-crossing in RX DFE DAC for even adder offset calibration

Table continues on the next page...

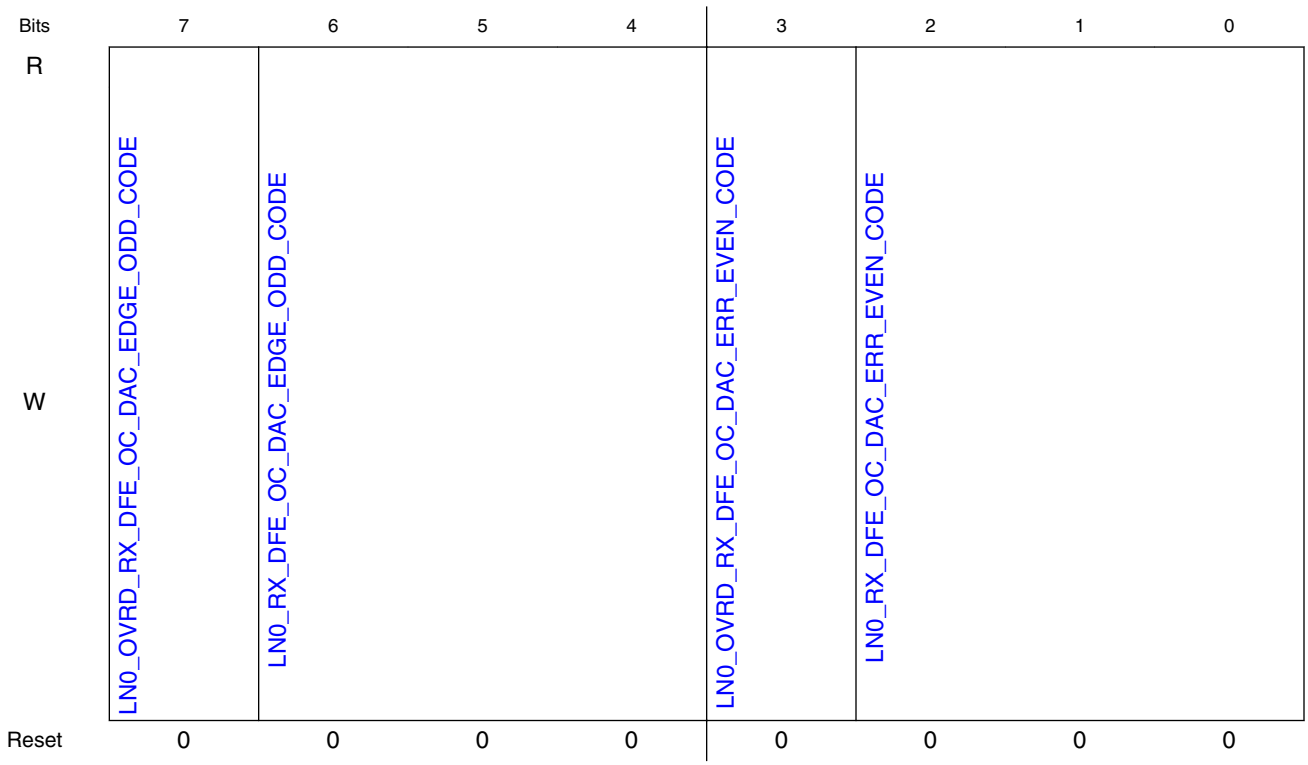
Field	Function
LN0_RX_DFE_OC_DAC_ADDER_EVEN_CODE	
5 LN0_OVRD_RX_DFE_OC_DAC_ADDER_ODD_CODE	Override enable for rx_dfe_oc_dac_adder_odd_code
4 LN0_RX_DFE_OC_DAC_ADDER_ODD_CODE	Fine control of zero-crossing in RX DFE DAC for odd adder offset calibration
3 LN0_OVRD_RX_DFE_OC_DAC_EDGE_EVEN_CODE	Override enable for rx_dfe_oc_dac_edge_even_code
2-0 LN0_RX_DFE_OC_DAC_EDGE_EVEN_CODE	Fine control of zero-crossing in RX DFE DAC for even edge path offset calibration

11.5.3.1.215 (TRSV_REG058)

11.5.3.1.215.1 Offset

Register	Offset
TRSV_REG058	560h

11.5.3.1.215.2 Diagram



11.5.3.1.215.3 Fields

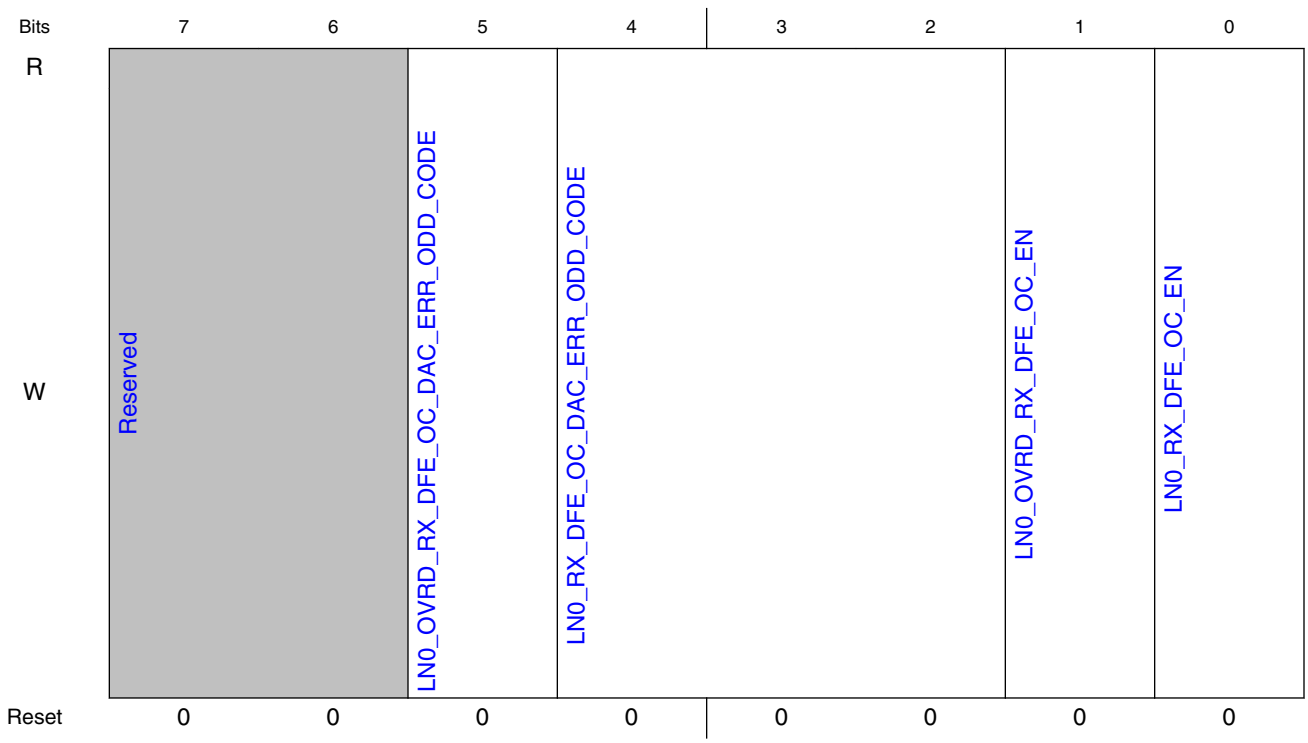
Field	Function
7 LN0_OVRD_RX_DFE_OC_DAC_EDGE_ODD_CODE	Override enable for rx_dfe_oc_dac_edge_odd_code
6-4 LN0_RX_DFE_OC_DAC_EDGE_ODD_CODE	Fine control of zero-crossing in RX DFE DAC for odd edge path offset calibration
3 LN0_OVRD_RX_DFE_OC_DAC_ERR_EVEN_CODE	Override enable for rx_dfe_oc_dac_err_even_code
2-0 LN0_RX_DFE_OC_DAC_ERR_EVEN_CODE	Fine control of zero-crossing in RX DFE DAC for even error path offset calibration

11.5.3.1.216 (TRSV_REG059)

11.5.3.1.216.1 Offset

Register	Offset
TRSV_REG059	564h

11.5.3.1.216.2 Diagram



11.5.3.1.216.3 Fields

Field	Function
7-6 —	Reserved
5 LN0_OVRD_RX_DFE_OC_DAC_ERR_ODD_CODE	Override enable for rx_dfe_oc_dac_err_odd_code
4-2	Fine control of zero-crossing in RX DFE DAC for odd error path offset calibration

Table continues on the next page...

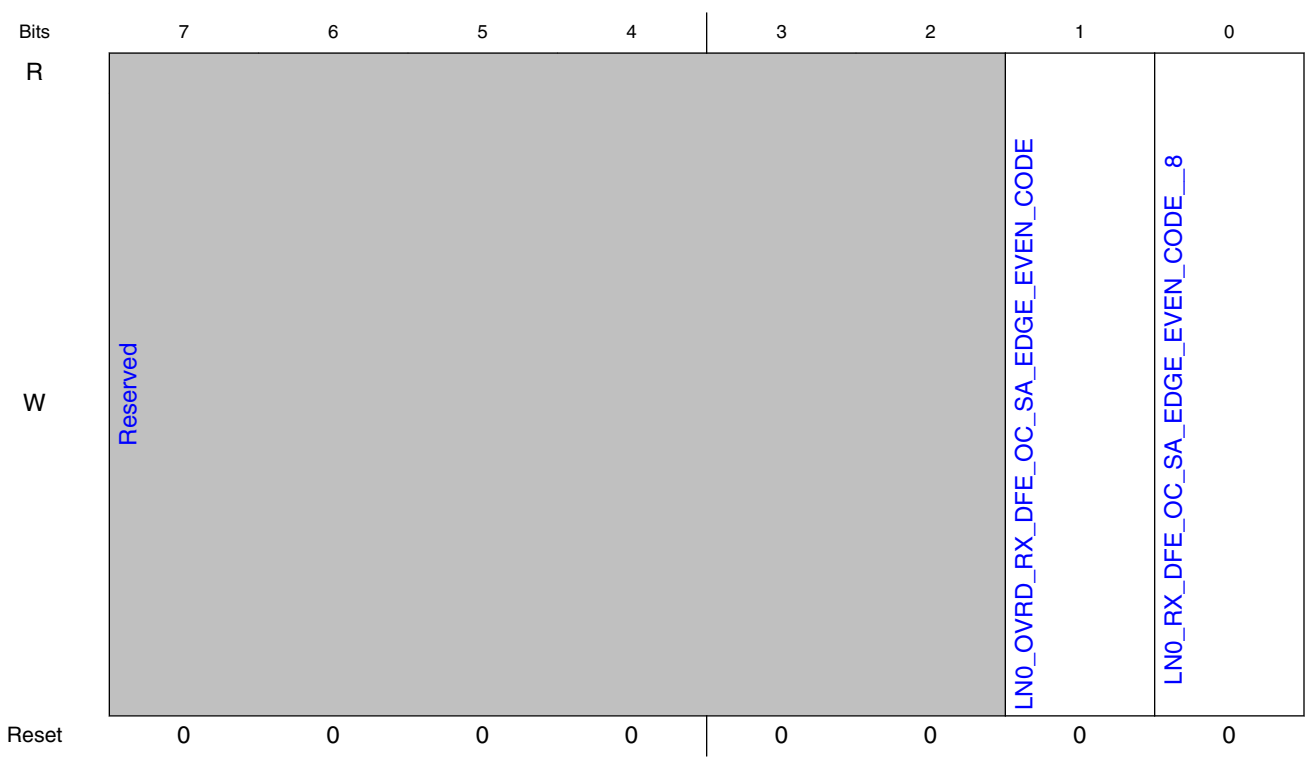
Field	Function
LN0_RX_DFE_OC_DAC_ERR_ODD_CODE	
1 LN0_OVRD_RX_DFE_OC_EN	Override enable for rx_dfe_oc_en
0 LN0_RX_DFE_OC_EN	RX DFE offset calibration progress enable 0: Disable (Normal operation of CTLE), 1: Enable (CTLE output forced to VCM)

11.5.3.1.217 (TRSV_REG05A)

11.5.3.1.217.1 Offset

Register	Offset
TRSV_REG05A	568h

11.5.3.1.217.2 Diagram



11.5.3.1.217.3 Fields

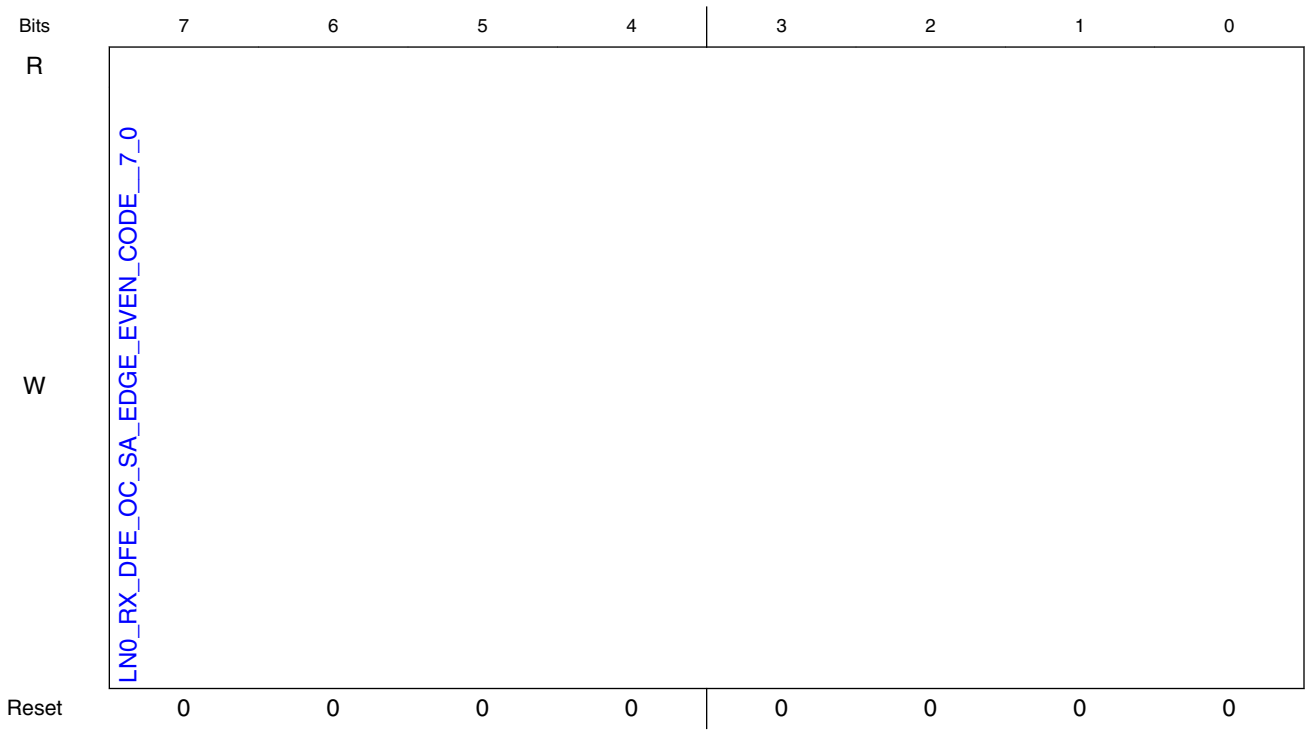
Field	Function
7-2 —	Reserved
1 LN0_OVRD_RX_DFE_OC_SA_EDGE_EVEN_CODE	Override enable for rx_dfe_oc_sa_edge_even_code
0 LN0_RX_DFE_OC_SA_EDGE_EVEN_CODE__8	RX DFE even edge path offset calibration code

11.5.3.1.218 (TRSV_REG05B)

11.5.3.1.218.1 Offset

Register	Offset
TRSV_REG05B	56Ch

11.5.3.1.218.2 Diagram



11.5.3.1.218.3 Fields

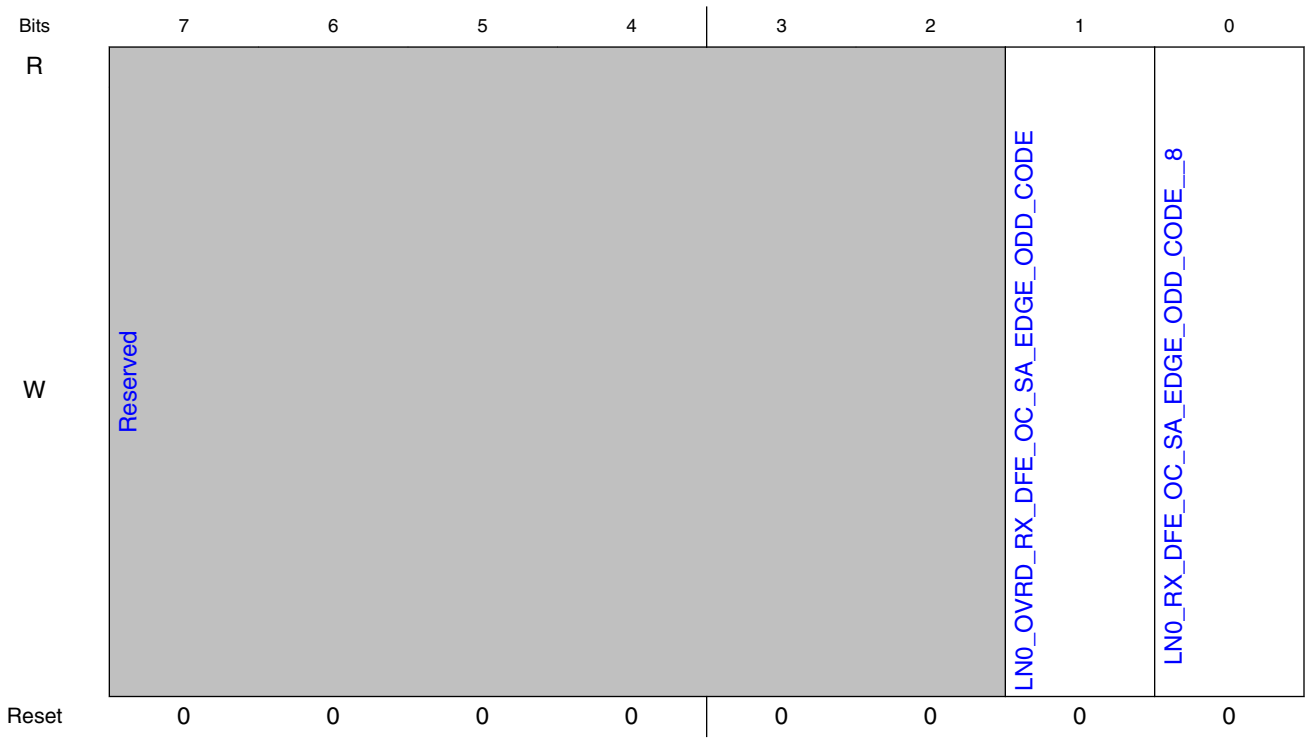
Field	Function
7-0 LN0_RX_DFE_OC_SA_EDGE_EVEN_CODE__7_0	RX DFE even edge path offset calibration code

11.5.3.1.219 (TRSV_REG05C)

11.5.3.1.219.1 Offset

Register	Offset
TRSV_REG05C	570h

11.5.3.1.219.2 Diagram



11.5.3.1.219.3 Fields

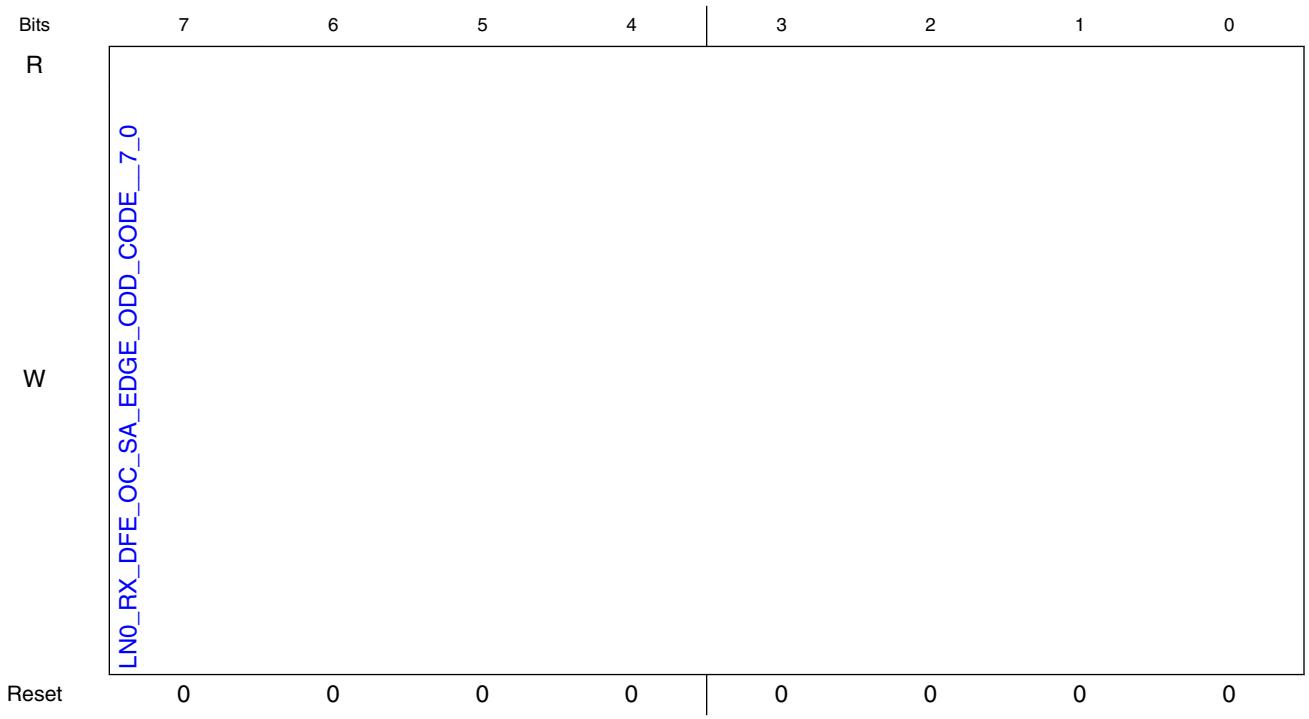
Field	Function
7-2 —	Reserved
1 LN0_OVRD_RX_DFE_OC_SA_EDGE_ODD_CODE	Override enable for rx_dfe_oc_sa_edge_odd_code
0 LN0_RX_DFE_OC_SA_EDGE_ODD_CODE__8	RX DFE odd edge path offset calibration code

11.5.3.1.220 (TRSV_REG05D)

11.5.3.1.220.1 Offset

Register	Offset
TRSV_REG05D	574h

11.5.3.1.220.2 Diagram



11.5.3.1.220.3 Fields

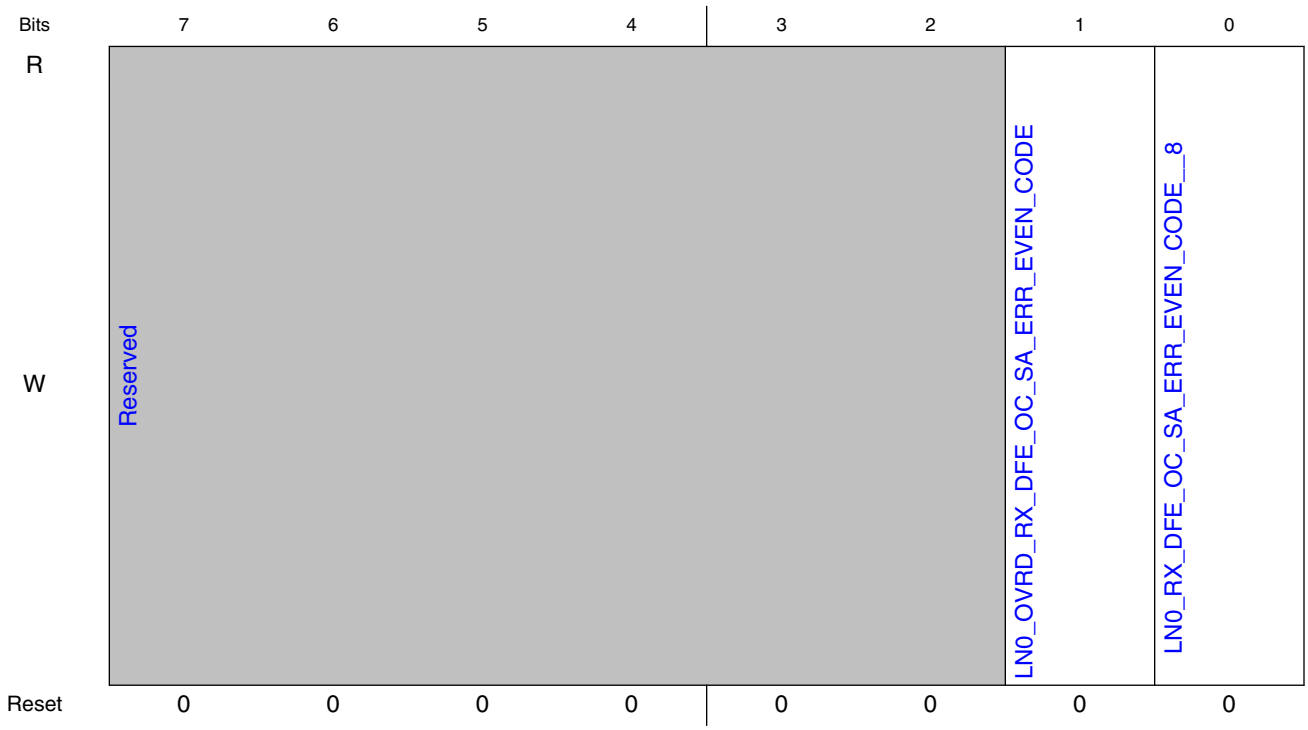
Field	Function
7-0 LN0_RX_DFE_OC_SA_EDGE_ODD_CODE__7_0	RX DFE odd edge path offset calibration code

11.5.3.1.221 (TRSV_REG05E)

11.5.3.1.221.1 Offset

Register	Offset
TRSV_REG05E	578h

11.5.3.1.221.2 Diagram



11.5.3.1.221.3 Fields

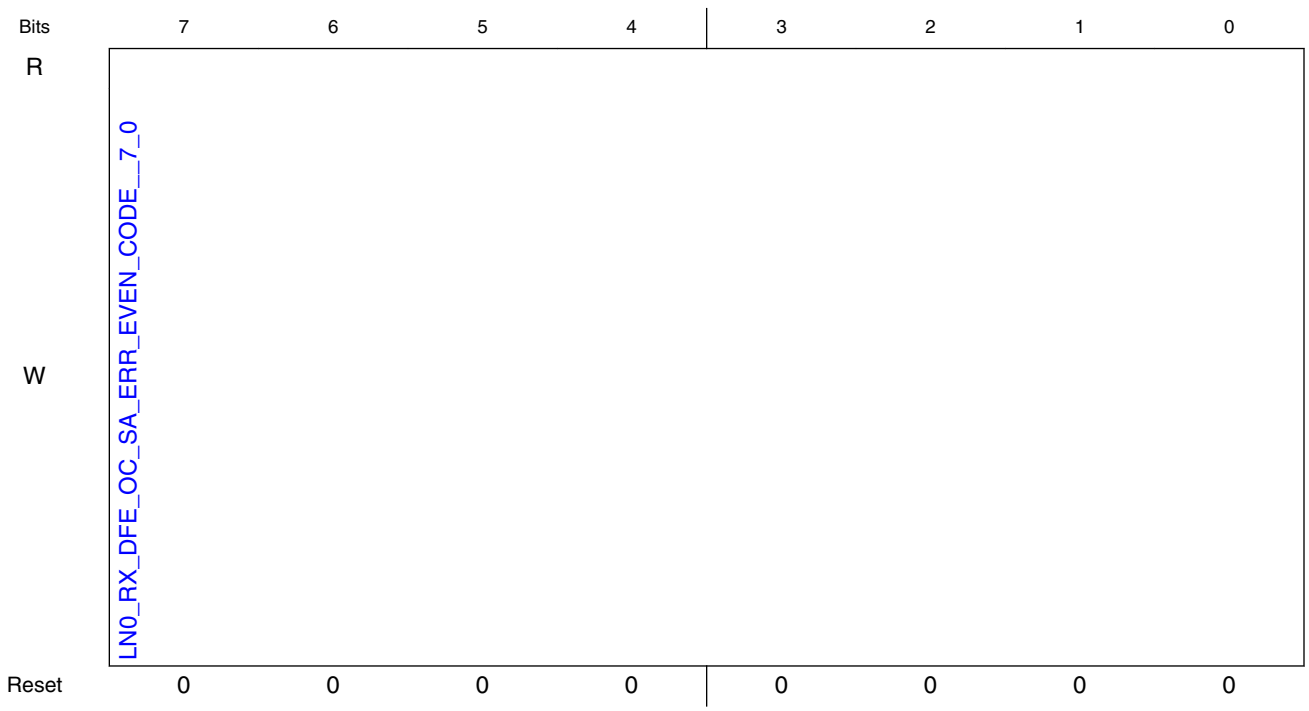
Field	Function
7-2 —	Reserved
1 LN0_OVRD_RX_DFE_OC_SA_ERR_EVEN_CODE	Override enable for rx_dfe_oc_sa_err_even_code
0 LN0_RX_DFE_OC_SA_ERR_EVEN_CODE__8	RX DFE even error path offset calibration code

11.5.3.1.222 (TRSV_REG05F)

11.5.3.1.222.1 Offset

Register	Offset
TRSV_REG05F	57Ch

11.5.3.1.222.2 Diagram



11.5.3.1.222.3 Fields

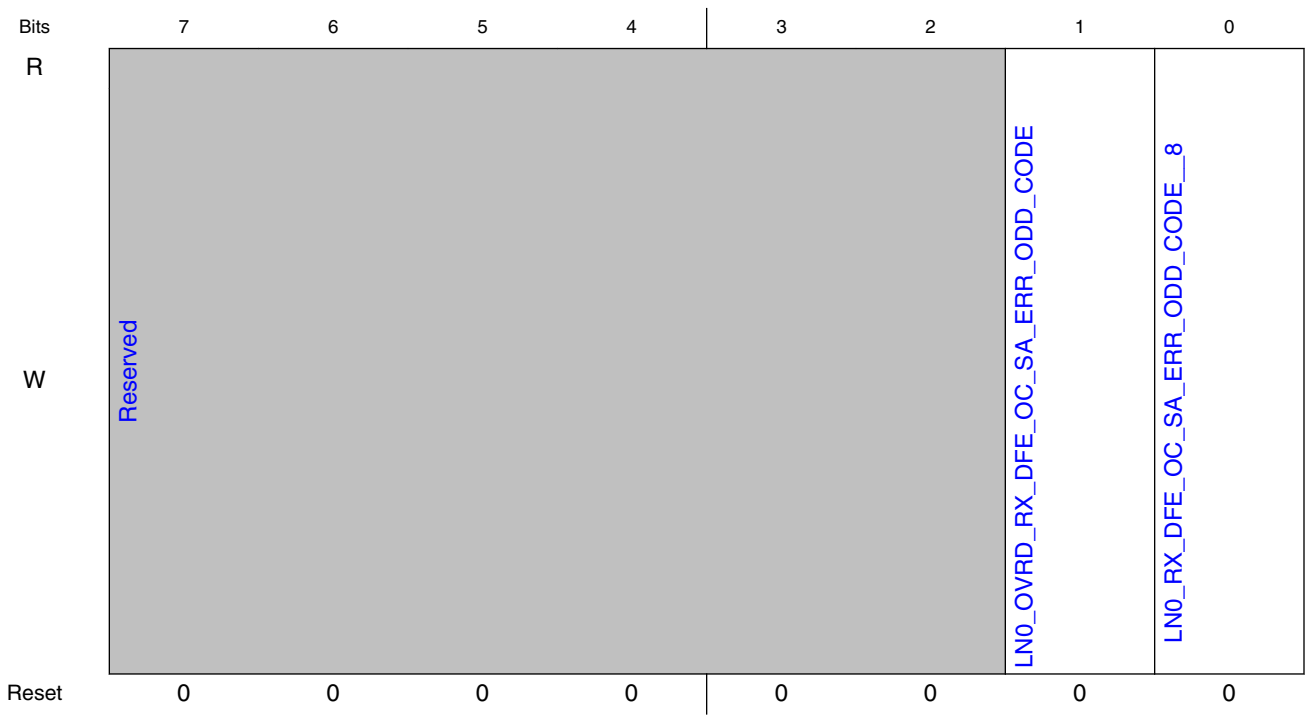
Field	Function
7-0 LN0_RX_DFE_OC_SA_ERR_EVEN_CODE__7_0	RX DFE even error path offset calibration code

11.5.3.1.223 (TRSV_REG060)

11.5.3.1.223.1 Offset

Register	Offset
TRSV_REG060	580h

11.5.3.1.223.2 Diagram



11.5.3.1.223.3 Fields

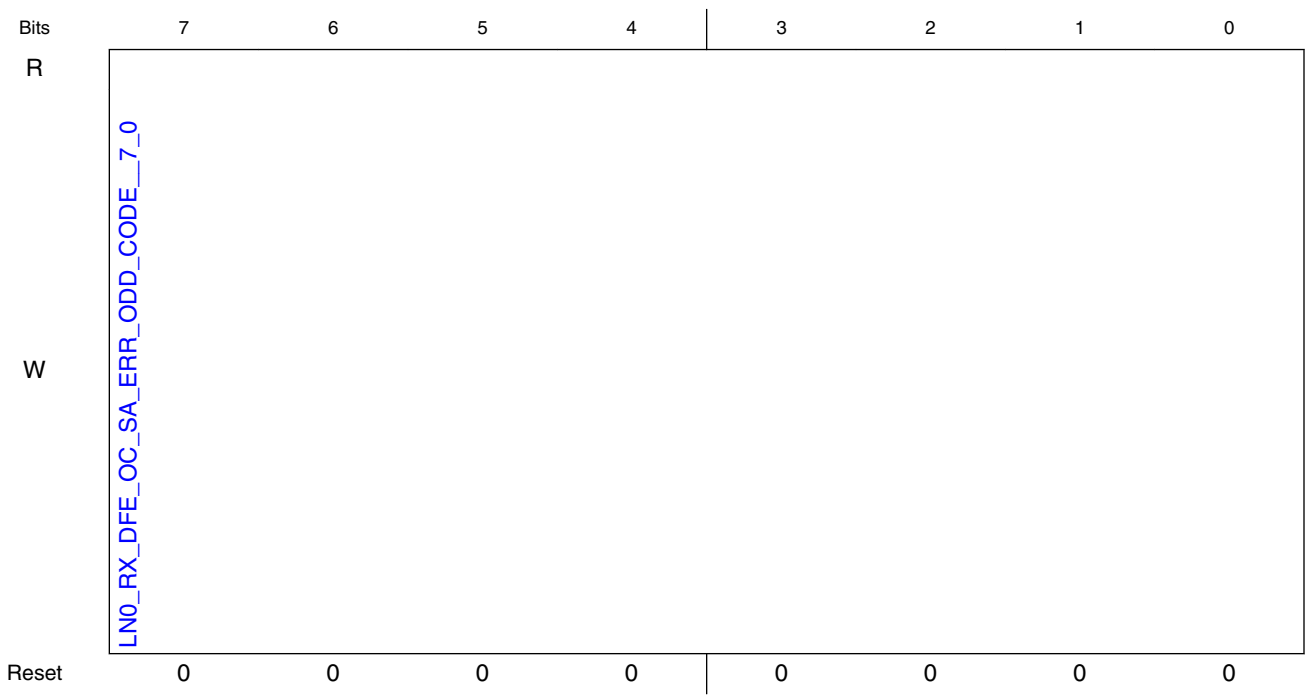
Field	Function
7-2 —	Reserved
1 LN0_OVRD_RX_DFE_OC_SA_ERR_ODD_CODE	Override enable for rx_dfe_oc_sa_err_odd_code
0 LN0_RX_DFE_OC_SA_ERR_ODD_CODE__8	RX DFE odd error path offset calibration code

11.5.3.1.224 (TRSV_REG061)

11.5.3.1.224.1 Offset

Register	Offset
TRSV_REG061	584h

11.5.3.1.224.2 Diagram



11.5.3.1.224.3 Fields

Field	Function
7-0 LN0_RX_DFE_OC_SA_ERR_ODD_CODE__7_0	RX DFE odd error path offset calibration code

11.5.3.1.225 (TRSV_REG062)

11.5.3.1.225.1 Offset

Register	Offset
TRSV_REG062	588h

11.5.3.1.225.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_OVRD_RX_DFE_SA_DATA_EVEN_OC_EN	LN0_RX_DFE_SA_DATA_EVEN_OC_EN	LN0_OVRD_RX_DFE_SA_DATA_ODD_OC_EN	LN0_RX_DFE_SA_DATA_ODD_OC_EN	LN0_OVRD_RX_DFE_SA_EDGE_OC_EN	LN0_RX_DFE_SA_EDGE_OC_EN	LN0_OVRD_RX_DFE_SA_ERR_OC_EN	LN0_RX_DFE_SA_ERR_OC_EN
W								
Reset	0	0	0	0	0	0	0	0

11.5.3.1.225.3 Fields

Field	Function
7 LN0_OVRD_RX_DFE_SA_DATA_EVEN_OC_EN	Override enable for rx_dfe_sa_data_even_oc_en
6 LN0_RX_DFE_SA_DATA_EVEN_OC_EN	RX DFE data odd path enable in offset calibration (If all of rx_dfe_sa_*_oc_en are 0, all path are autimatically activated as normal mode) 0: Disable, 1: only data odd path alive
5	Override enable for rx_dfe_sa_data_odd_oc_en

Table continues on the next page...

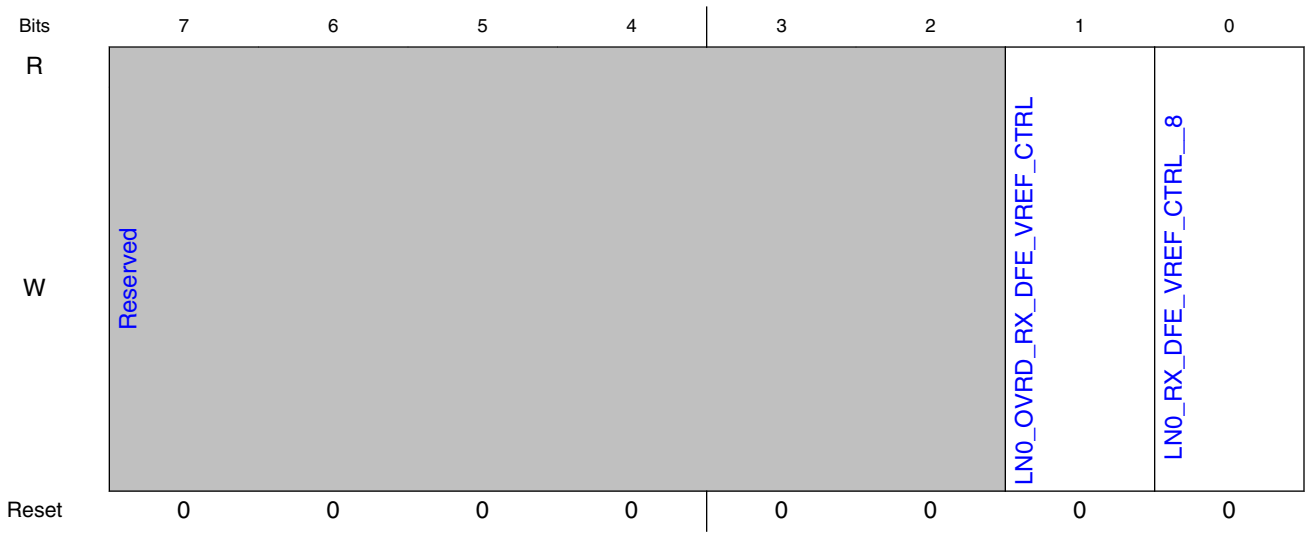
Field	Function
LN0_OVRD_RX_DFE_SA_DATA_ODD_OC_EN	
4 LN0_RX_DFE_SA_DATA_ODD_OC_EN	RX DFE data even path enable in offset calibration (If all of rx_dfe_sa_*_oc_en are 0, all path are autimatically activated as normal mode) 0: Disable, 1: only data even path alive
3 LN0_OVRD_RX_DFE_SA_EDGE_OC_EN	Override enable for rx_dfe_sa_edge_oc_en
2 LN0_RX_DFE_SA_EDGE_OC_EN	RX DFE edge path enable in offset calibration (If all of rx_dfe_sa_*_oc_en are 0, all path are autimatically activated as normal mode) 0: Disable, 1: only edge path alive
1 LN0_OVRD_RX_DFE_SA_ERR_OC_EN	Override enable for rx_dfe_sa_err_oc_en
0 LN0_RX_DFE_SA_ERR_OC_EN	RX DFE error path enable in offset calibration (If all of rx_dfe_sa_*_oc_en are 0, all path are autimatically activated as normal mode) 0: Disable, 1: only error path alive

11.5.3.1.226 (TRSV_REG063)

11.5.3.1.226.1 Offset

Register	Offset
TRSV_REG063	58Ch

11.5.3.1.226.2 Diagram



11.5.3.1.226.3 Fields

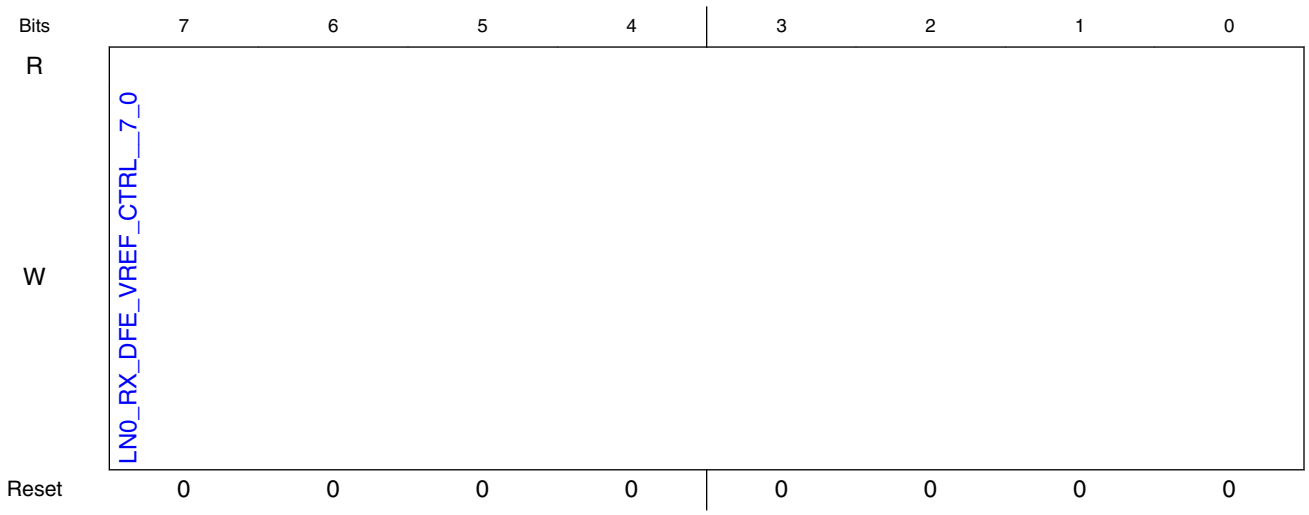
Field	Function
7-2 —	Reserved
1 LN0_OVRD_RX_DFE_VREF_CTRL	Override enable for rx_dfe_vref_ctrl
0 LN0_RX_DFE_VREF_CTRL__8	RX DFE Vref control 0000_0000: -300mVdiff ... 1111_1111: +300mVdiff

11.5.3.1.227 (TRSV_REG064)

11.5.3.1.227.1 Offset

Register	Offset
TRSV_REG064	590h

11.5.3.1.227.2 Diagram



11.5.3.1.227.3 Fields

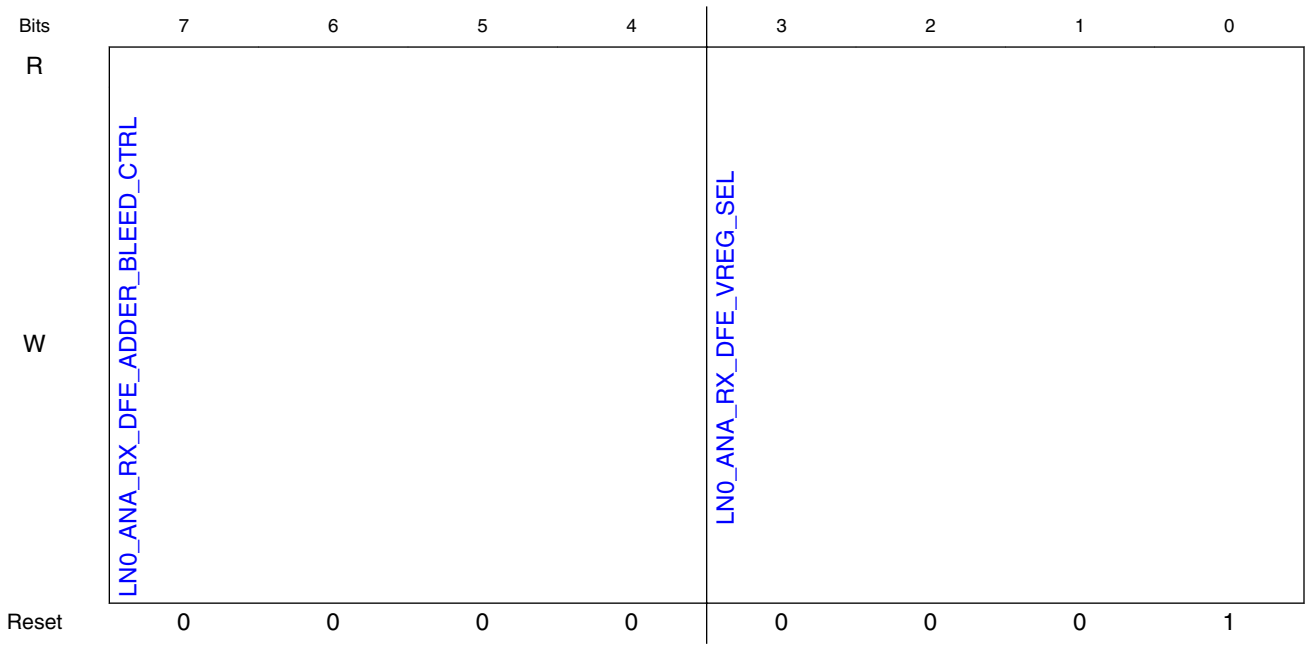
Field	Function
7-0 LN0_RX_DFE_VREF_CTRL__7_0	RX DFE Vref control 0000_0000: -300mVdiff ... 1111_1111: +300mVdiff

11.5.3.1.228 (TRSV_REG065)

11.5.3.1.228.1 Offset

Register	Offset
TRSV_REG065	594h

11.5.3.1.228.2 Diagram



11.5.3.1.228.3 Fields

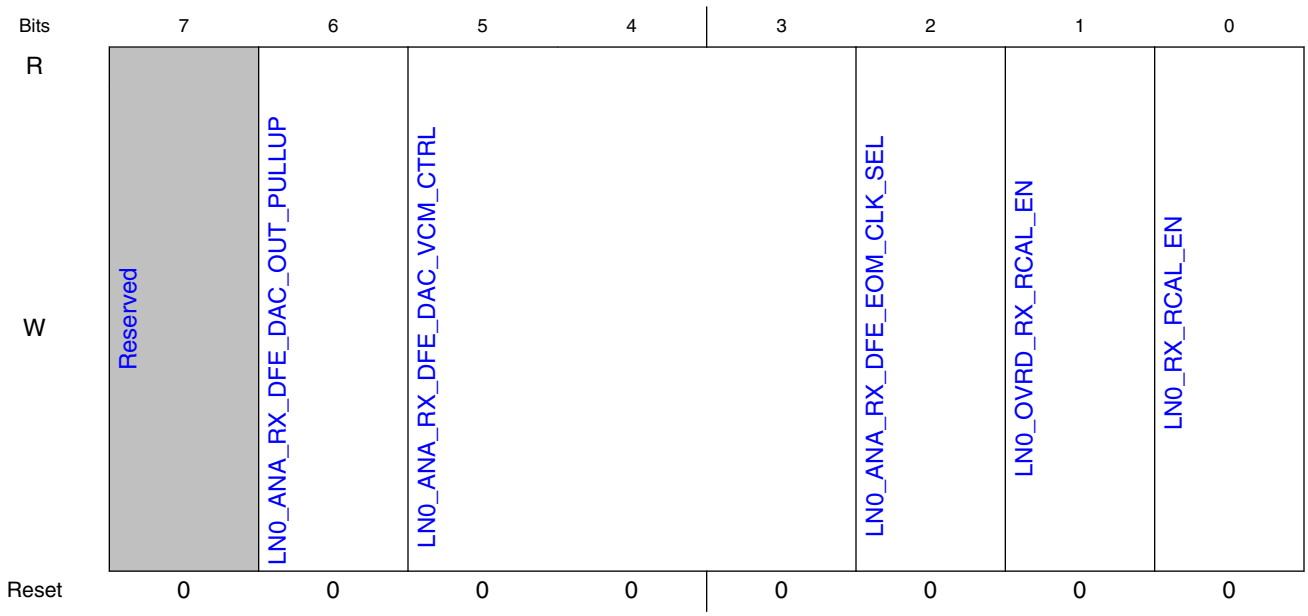
Field	Function
7-4 LN0_ANA_RX_DFE_ADDER_BLEED_CTRL	
3-0 LN0_ANA_RX_DFE_VREG_SEL	N/A, DFE bias current control [3:2]: DFE adder bias current, [1:0]: OC DAC bias current. 00: 37.5uA, 01: 50uA, 10: 62.5uA, 11: 75uA

11.5.3.1.229 (TRSV_REG066)

11.5.3.1.229.1 Offset

Register	Offset
TRSV_REG066	598h

11.5.3.1.229.2 Diagram



11.5.3.1.229.3 Fields

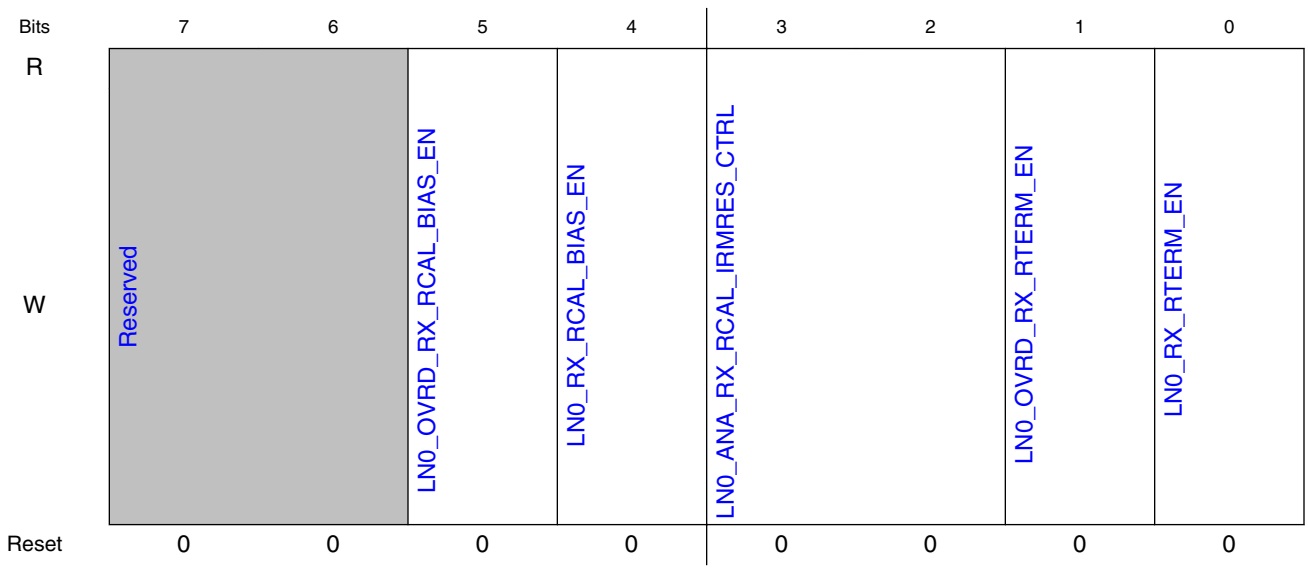
Field	Function
7 —	Reserved
6 LN0_ANA_RX_DFE_DAC_OUT_PULLUP	Pull-up all DAC output in RX DFE to disable all offset code effect 0: DAC output enable (offset code effect on), 1: DAC output pull-up (offset code effect off)
5-3 LN0_ANA_RX_DFE_DAC_VCM_CTRL	
2 LN0_ANA_RX_DFE_EOM_CLK_SEL	RX EOM clock selection 0: CDR clock, 1: PLL clock
1 LN0_OVRD_RX_RCAL_EN	Override enable for rx_rcal_en
0 LN0_RX_RCAL_EN	RX RCAL enable 0: Disable, 1: Enable

11.5.3.1.230 (TRSV_REG067)

11.5.3.1.230.1 Offset

Register	Offset
TRSV_REG067	59Ch

11.5.3.1.230.2 Diagram



11.5.3.1.230.3 Fields

Field	Function
7-6 —	Reserved
5 LN0_OVRD_RX_RCAL_BIAS_EN	Override enable for rx_rcal_bias_en
4 LN0_RX_RCAL_BIAS_EN	RX RCAL bias current enable 0: Disable, 1: Enable
3-2 LN0_ANA_RX_RCAL_IRMRES_CTRL	

Table continues on the next page...

Field	Function
1 LN0_OVRD_RX_RTERM_EN	Override enable for rx_rterm_en
0 LN0_RX_RTERM_EN	RX RTERM enable 0: Disable, 1: Enable

11.5.3.1.231 (TRSV_REG068)

11.5.3.1.231.1 Offset

Register	Offset
TRSV_REG068	5A0h

11.5.3.1.231.2 Diagram

Bits	7	6	5	4	3	2	1	0
R								
W	LN0_RX_RTERM_42P5_EN_G1	LN0_RX_RTERM_42P5_EN_G2	LN0_RX_RTERM_42P5_EN_G3	LN0_RX_RTERM_42P5_EN_G4	LN0_ANA_RX_RTERM_INCM_ITAIL_CTRL		LN0_ANA_RX_RTERM_INCM_SW_CTRL	
Reset	0	0	1	1	1	1	1	1

11.5.3.1.231.3 Fields

Field	Function
7	[GEN1] RX RTERM resistance shift 0: 50ohm, 1: 42.5ohm

Table continues on the next page...

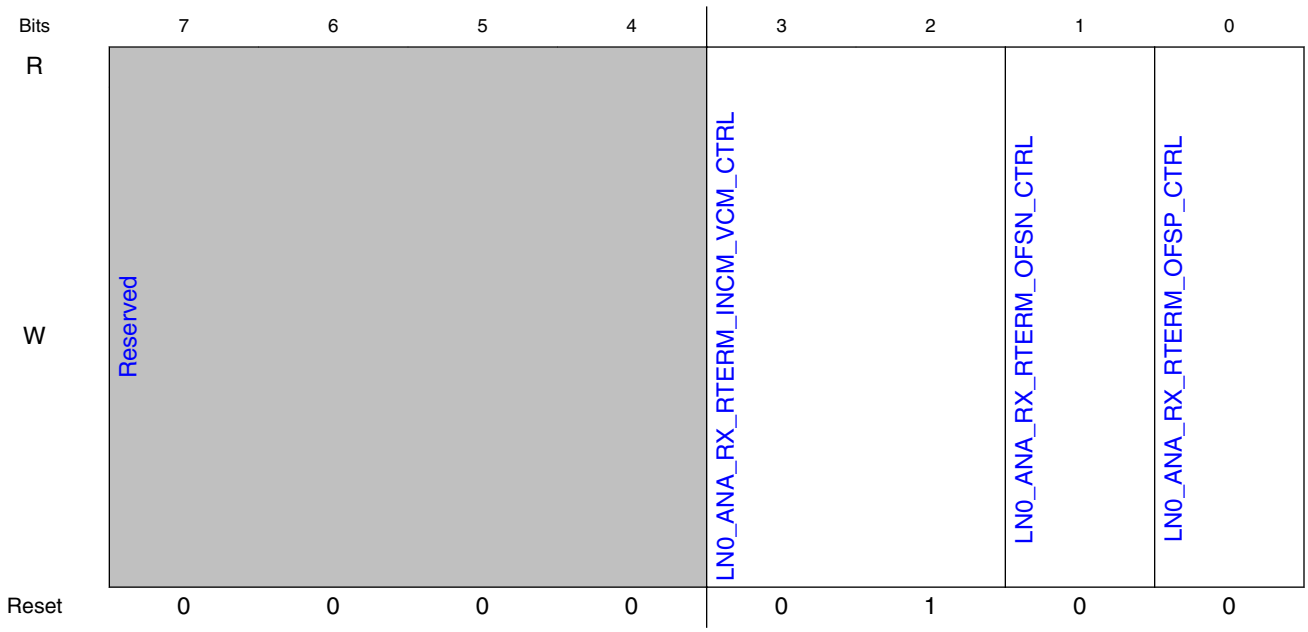
Field	Function
LN0_RX_RTERM_42P5_EN_G1	
6 LN0_RX_RTERM_42P5_EN_G2	[GEN2]
5 LN0_RX_RTERM_42P5_EN_G3	[GEN3]
4 LN0_RX_RTERM_42P5_EN_G4	[GEN4]
3-2 LN0_ANA_RX_RTERM_INCM_ITAIL_CTRL	RX RTERM single-ended impedance control by current control 00: Max, .. , 11: Min
1-0 LN0_ANA_RX_RTERM_INCM_SW_CTRL	RX RTERM single-ended impedance control by switch control 00: Max, .. , 11: Min

11.5.3.1.232 (TRSV_REG069)

11.5.3.1.232.1 Offset

Register	Offset
TRSV_REG069	5A4h

11.5.3.1.232.2 Diagram



11.5.3.1.232.3 Fields

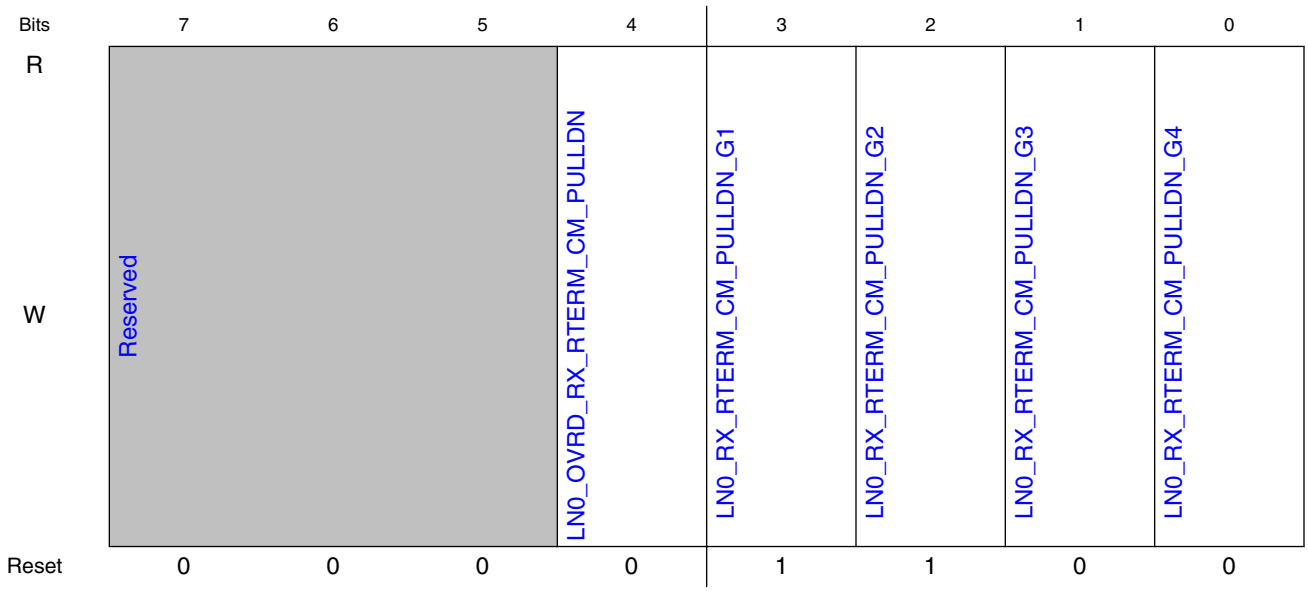
Field	Function
7-4 —	Reserved
3-2 LN0_ANA_RX_RTERM_INCM_VCM_CTRL	RX RTERM output common-mode voltage control 00: 180mV, 01: 225mV, 10: 270mV, 11: 300mV
1 LN0_ANA_RX_RTERM_OFSN_CTRL	Offset code for RX RTERM N node
0 LN0_ANA_RX_RTERM_OFSP_CTRL	Offset code for RX RTERM P node 0: No offset added, 1: +1 offset code added

11.5.3.1.233 (TRSV_REG06A)

11.5.3.1.233.1 Offset

Register	Offset
TRSV_REG06A	5A8h

11.5.3.1.233.2 Diagram



11.5.3.1.233.3 Fields

Field	Function
7-5 —	Reserved
4 LN0_OVRD_RX_RTERM_CM_PULLDN	Override enable for rx_rterm_cm_pulldn_g1
3 LN0_RX_RTERM_CM_PULLDN_G1	[GEN1] RX RTERM termination voltage pull-down 0: No pull-down, 1: Pull-down
2 LN0_RX_RTERM_CM_PULLDN_G2	[GEN2]
1	[GEN3]

Table continues on the next page...

Field	Function
LN0_RX_RTERM_CM_PULLDN_G3	
0 LN0_RX_RTERM_CM_PULLDN_G4	[GEN4]

11.5.3.1.234 (TRSV_REG06B)

11.5.3.1.234.1 Offset

Register	Offset
TRSV_REG06B	5ACh

11.5.3.1.234.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_OVRD_RX_RTERM_VCM_EN	LN0_RX_RTERM_VCM_EN_G1	LN0_RX_RTERM_VCM_EN_G2	LN0_RX_RTERM_VCM_EN_G3	LN0_RX_RTERM_VCM_EN_G4	LN0_OVRD_RX_SQ_BMR_EN	LN0_RX_SQ_BMR_EN	LN0_ANA_RX_SQ_VREF_820M_LPF_BYPASS
Reset	0	0	0	1	1	0	0	0

11.5.3.1.234.3 Fields

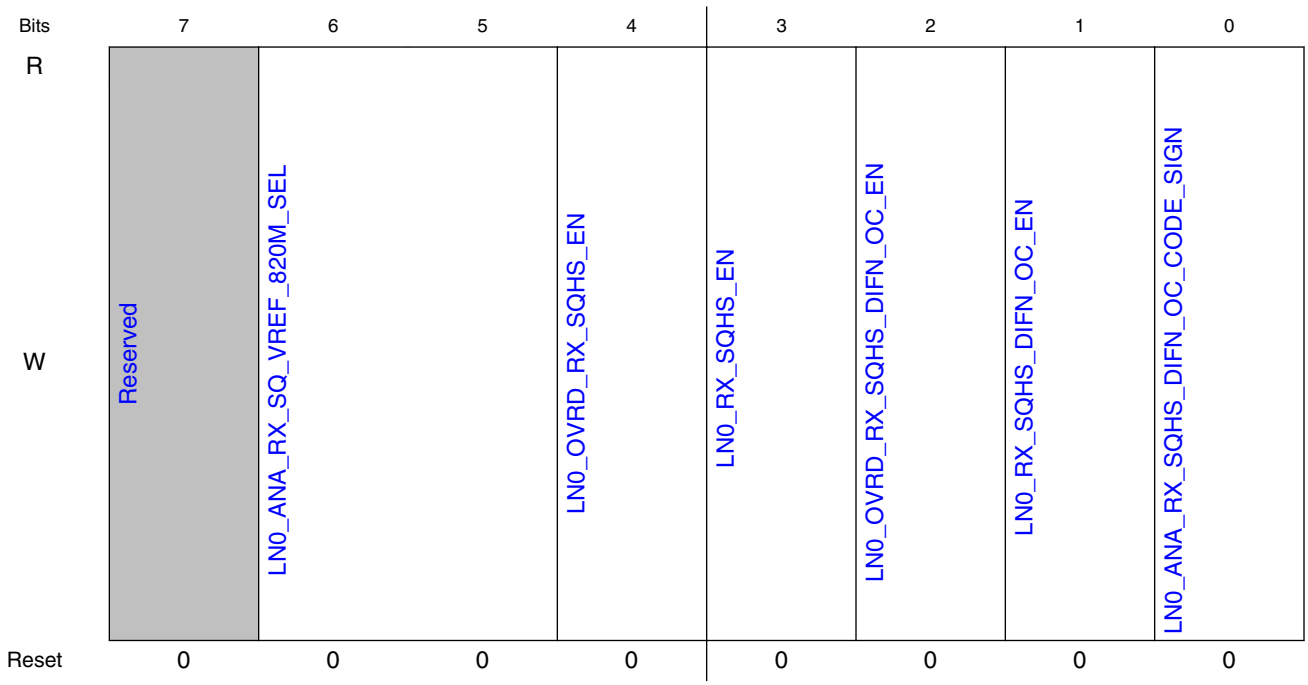
Field	Function
7 LN0_OVRD_RX_RTERM_VCM_EN	Override enable for rx_rterm_vcm_en_g1
6 LN0_RX_RTERM_VCM_EN_G1	[GEN1]
5 LN0_RX_RTERM_VCM_EN_G2	[GEN2]
4 LN0_RX_RTERM_VCM_EN_G3	[GEN3]
3 LN0_RX_RTERM_VCM_EN_G4	[GEN4]
2 LN0_OVRD_RX_SQ_BMR_EN	Override enable for rx_sq_bmr_en
1 LN0_RX_SQ_BMR_EN	
0 LN0_ANA_RX_SQ_VREF_820M_LPF_BYPASS	

11.5.3.1.235 (TRSV_REG06C)

11.5.3.1.235.1 Offset

Register	Offset
TRSV_REG06C	5B0h

11.5.3.1.235.2 Diagram



11.5.3.1.235.3 Fields

Field	Function
7 —	Reserved
6-5 LN0_ANA_RX_SQ_VREF_820M_SEL	
4 LN0_OVRD_RX_SQHS_EN	Override enable for rx_sqhs_en
3 LN0_RX_SQHS_EN	RX high-speed squelch enable 0: Disable, 1: Enable
2 LN0_OVRD_RX_SQHS_DIFN_OC_EN	Override enable for rx_sqhs_difn_oc_en
1 LN0_RX_SQHS_DIFN_OC_EN	RX high-squelch diff-N offset calibration enable
0	RX high-squelch diff-N offset sign

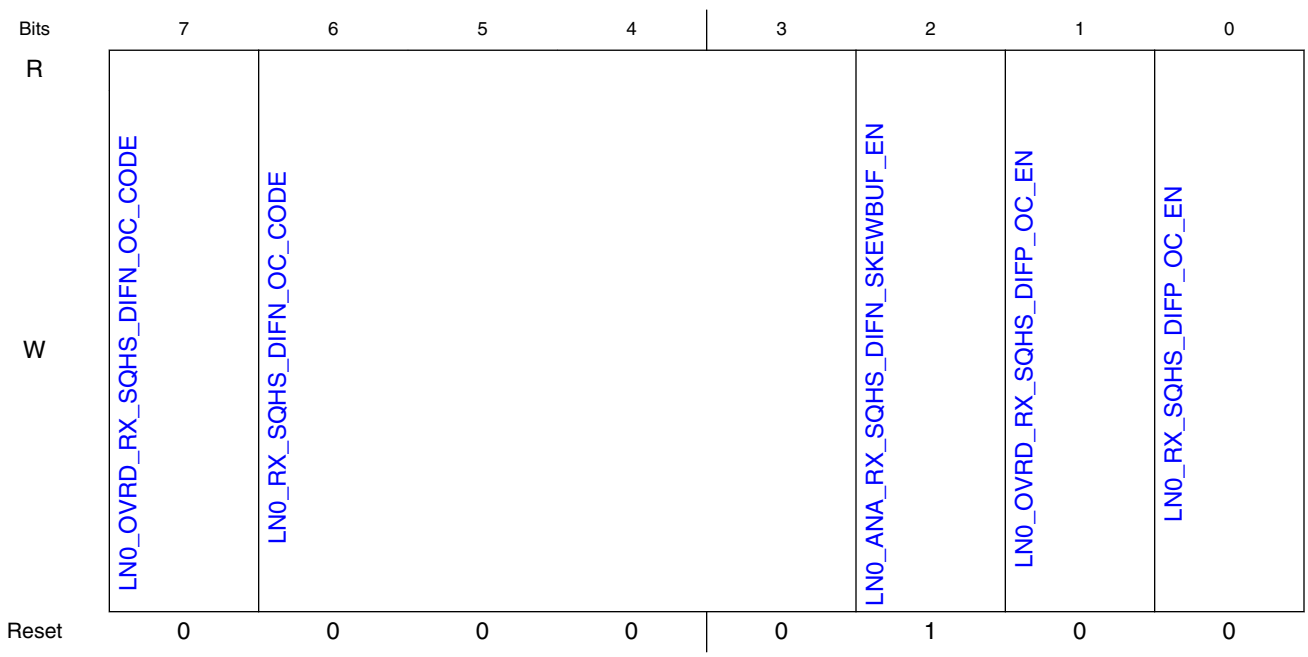
Field	Function
LN0_ANA_RX_SQHS_DIFN_OC_CODE_SIGN	0: Positive, 1: Negative

11.5.3.1.236 (TRSV_REG06D)

11.5.3.1.236.1 Offset

Register	Offset
TRSV_REG06D	5B4h

11.5.3.1.236.2 Diagram



11.5.3.1.236.3 Fields

Field	Function
7 LN0_OVRD_RX_SQHS_DIFN_OC_CODE	Override enable for rx_sqhs_difn_oc_code
6-3	RX high-squelch diff-N manual offset code

Table continues on the next page...

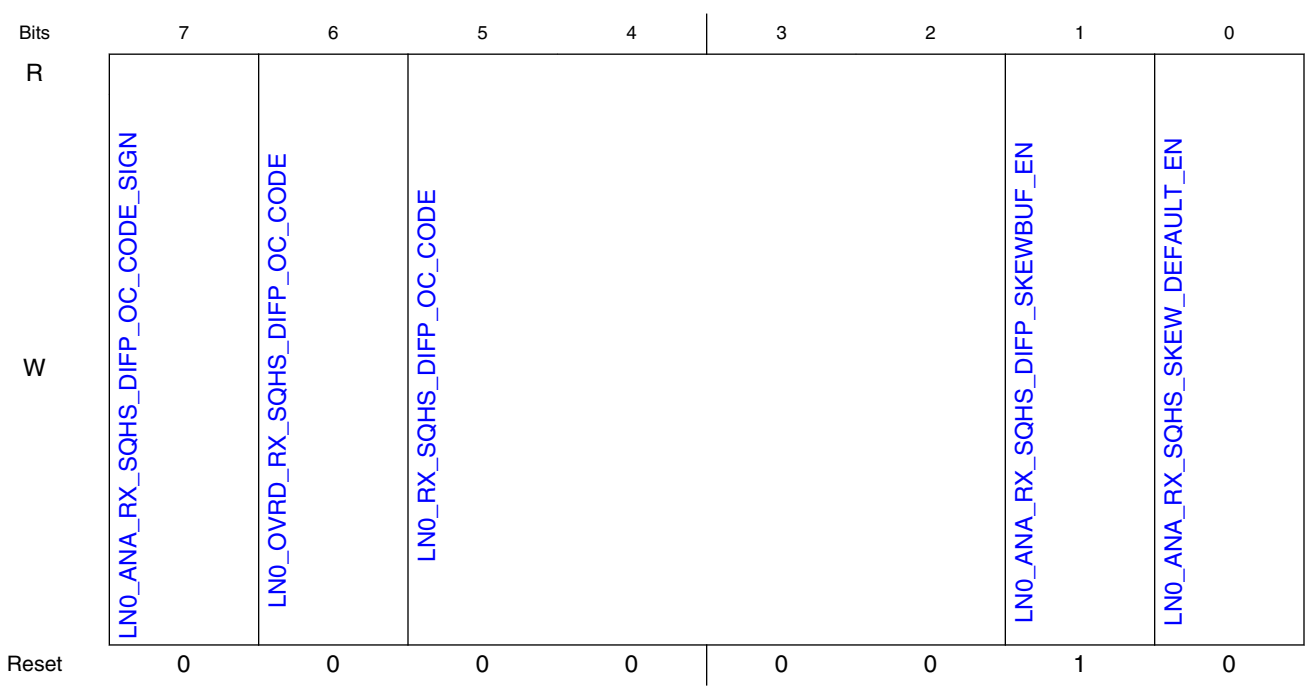
Field	Function
LN0_RX_SQHS_DIFN_OC_CODE	
2 LN0_ANA_RX_SQHS_DIFN_SKEWBUF_EN	
1 LN0_OVRD_RX_SQHS_DIFP_OC_EN	Override enable for rx_sqhs_difp_oc_en
0 LN0_RX_SQHS_DIFP_OC_EN	RX high-squelch diff-P offset calibration enable

11.5.3.1.237 (TRSV_REG06E)

11.5.3.1.237.1 Offset

Register	Offset
TRSV_REG06E	5B8h

11.5.3.1.237.2 Diagram



11.5.3.1.237.3 Fields

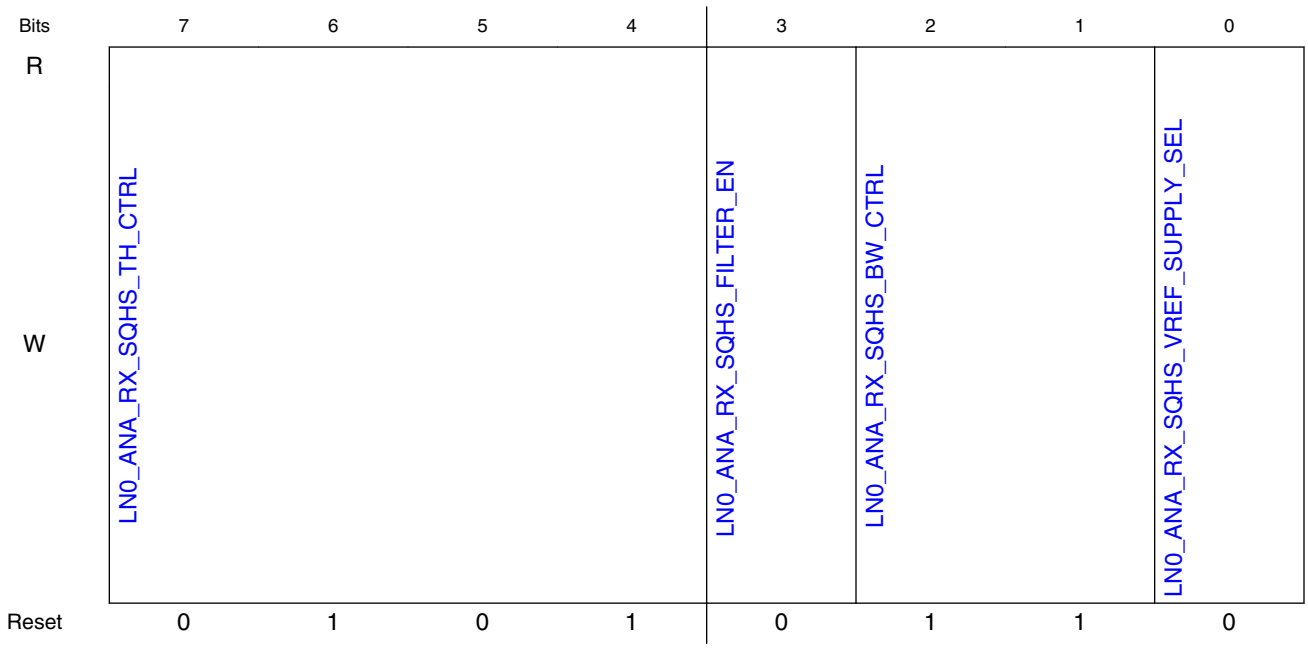
Field	Function
7 LN0_ANA_RX_SQHS_DIFP_OC_CODE_SIGN	RX high-squelch diff-P offset sign 0: Positive, 1: Negative
6 LN0_OVRD_RX_SQHS_DIFP_OC_CODE	Override enable for rx_sqhs_difp_oc_code
5-2 LN0_RX_SQHS_DIFP_OC_CODE	RX squelch diff-P manual offset code 0000: (-) Max, … 1111: (+) Max
1 LN0_ANA_RX_SQHS_DIFP_SKEWBUF_EN	Enable the high speed Squelch DIFP SKEW BUFFER
0 LN0_ANA_RX_SQHS_SKEW_DEFAULT_EN	Fixed skew for PCIe/SATA Squelch

11.5.3.1.238 (TRSV_REG06F)

11.5.3.1.238.1 Offset

Register	Offset
TRSV_REG06F	5BCh

11.5.3.1.238.2 Diagram



11.5.3.1.238.3 Fields

Field	Function
7-4 LN0_ANA_RX_SQHS_TH_CTRL	RX squelch threshold voltage selection 0000: 0mV, 0001: 45mV, 0010: 60mV, 0011: 75mV 0100: 90mV, 0101: 105mV, 0110: 120mV, 0111: 135mV 1000: 150mV, 1001: 165mV, 1010: 180mV, 1011: 195mV 1100: 210mV, 1101: 225mV, 1110: 240mV, 1111: 255mV
3 LN0_ANA_RX_SQHS_FILTER_EN	SQHS loss detector enable
2-1 LN0_ANA_RX_SQHS_BW_CTRL	
0 LN0_ANA_RX_SQHS_VREF_SUPPLY_SEL	Selection of supply voltage of reference voltage for threshold calibration of HS SQ 0: VDD, 1: VREG

11.5.3.1.239 (TRSV_REG070)

11.5.3.1.239.1 Offset

Register	Offset
TRSV_REG070	5C0h

11.5.3.1.239.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_OVRD_RX_SQLS_DIFN_DET_EN	LN0_RX_SQLS_DIFN_DET_EN	LN0_OVRD_RX_SQLS_DIFP_DET_EN	LN0_RX_SQLS_DIFP_DET_EN	LN0_ANA_RX_SQLS_DIFP_FAST_ANA_SEL	LN0_ANA_RX_SQLS_DIFN_TH_CTRL		
W								
Reset	0	0	0	0	0	0	0	0

11.5.3.1.239.3 Fields

Field	Function
7 LN0_OVRD_RX_SQLS_DIFN_DET_EN	Override enable for rx_sqls_difn_det_en
6 LN0_RX_SQLS_DIFN_DET_EN	RX low-speed DIFN squelch enable 0: Disable, 1: Enable
5 LN0_OVRD_RX_SQLS_DIFP_DET_EN	Override enable for rx_sqls_difp_det_en

Table continues on the next page...

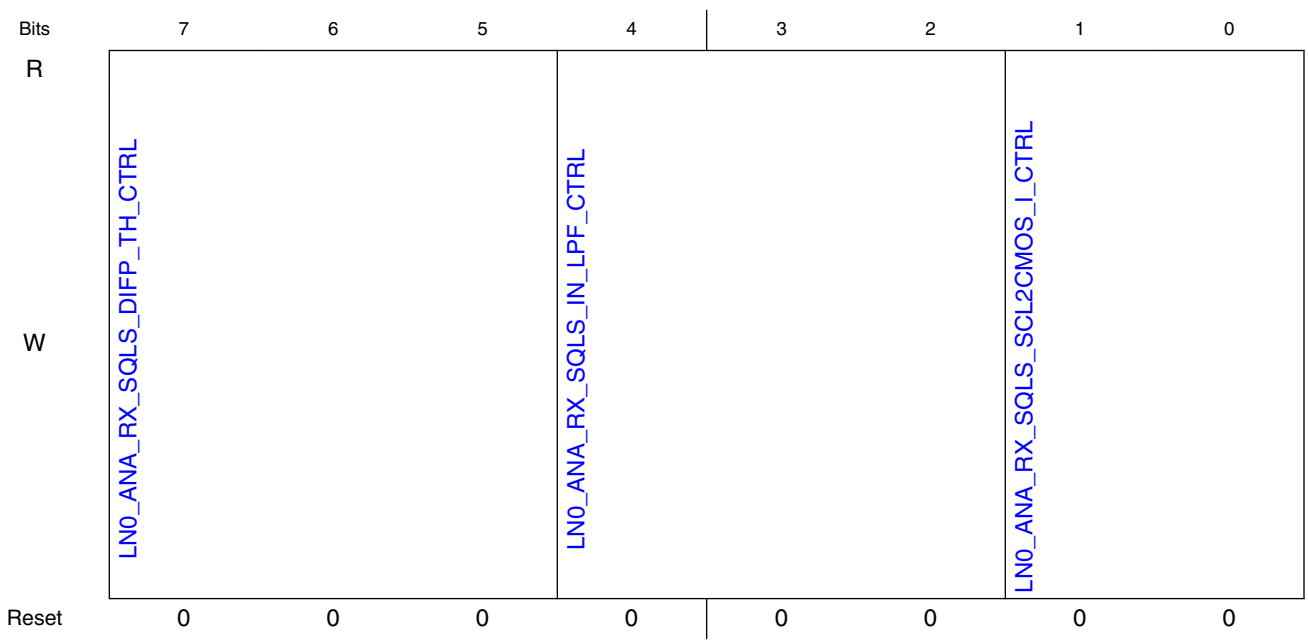
Field	Function
4 LN0_RX_SQLS_DIFP_DET_EN	RX low-speed DIFP squelch enable 0: Disable, 1: Enable
3 LN0_ANA_RX_SQLS_DIFP_FAST_ANA_SEL	RX DIFP detect signal selection for PRE_DATA_VALID and DATA_VALID signal 0: low-speed squelch detect signal, 1: low-speed squelch detect signal detect signal
2-0 LN0_ANA_RX_SQLS_DIFN_TH_CTRL	DIFN in MPHY, LFPS in USB: 0:40m, 1:60m, 2:80m, 3:96m, 4:116m, 5:135m, 6: 154m, 7:174m

11.5.3.1.240 (TRSV_REG071)

11.5.3.1.240.1 Offset

Register	Offset
TRSV_REG071	5C4h

11.5.3.1.240.2 Diagram



11.5.3.1.240.3 Fields

Field	Function
7-5 LN0_ANA_RX_SQLS_DIFP_TH_CTRL	DIFP in MPHY 0:40m, 1:60m, 2:80m, 3:96m, 4:116m, 5:135m, 6: 154m, 7:174m
4-2 LN0_ANA_RX_SQLS_IN_LPF_CTRL	Low pass filter resistor control for Squelch input : 00:30MHz,01:60MHz,, 10:100MHz, 11: 150MHz
1-0 LN0_ANA_RX_SQLS_SCL2CMOS_I_CTRL	Current controls for low-speed Squelch comparator current source 00 (Min BW) - 11:Max BW

11.5.3.1.241 (TRSV_REG072)

11.5.3.1.241.1 Offset

Register	Offset
TRSV_REG072	5C8h

11.5.3.1.241.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_OVRD_RX_PWM_OSC_EN	LN0_RX_PWM_OSC_EN	LN0_OVRD_RX_PWM_CNT_EN	LN0_RX_PWM_CNT_EN	LN0_OVRD_RX_PWM_RSTN	LN0_RX_PWM_RSTN	LN0_OVRD_RX_PWM_AFC_RSTN	LN0_RX_PWM_AFC_RSTN
W								
Reset	0	0	0	0	0	0	0	0

11.5.3.1.241.3 Fields

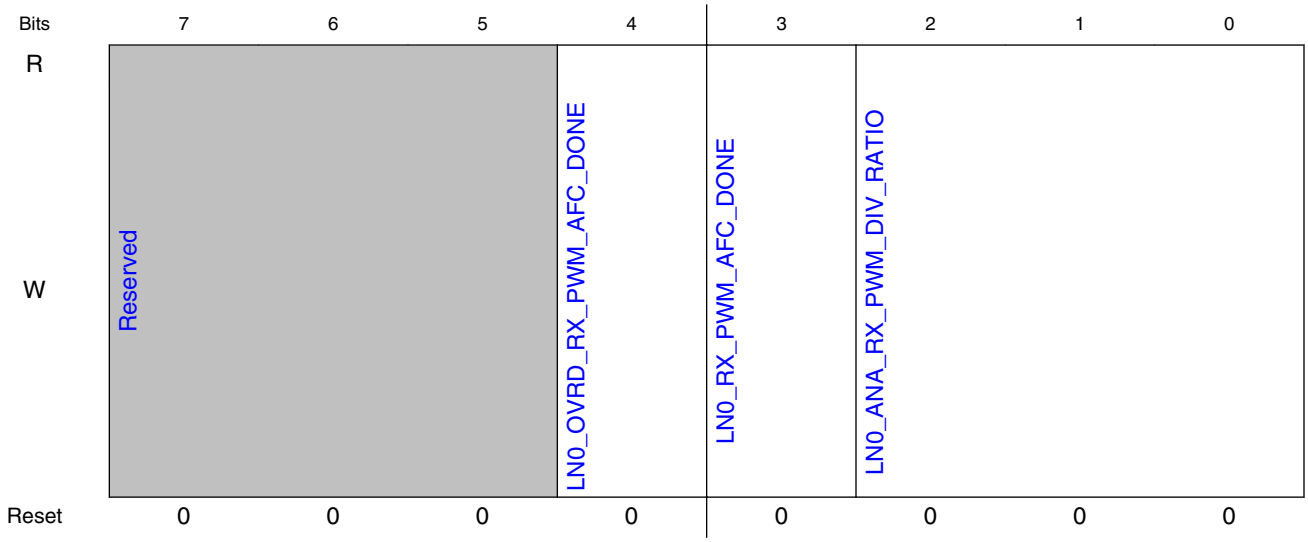
Field	Function
7 LN0_OVRD_RX_PWM_OSC_EN	Override enable for rx_pwm_osc_en
6 LN0_RX_PWM_OSC_EN	RX MPHY PWM oscillator enable which is used in analog RX block in order to oversample 0: Disable, 1: Enable
5 LN0_OVRD_RX_PWM_CNT_EN	Override enable for rx_pwm_cnt_en
4 LN0_RX_PWM_CNT_EN	Enalbe counter clock for PWM over sampling 0: Disable, 1: Enable
3 LN0_OVRD_RX_PWM_RSTN	Override enable for rx_pwm_rstn
2 LN0_RX_PWM_RSTN	RX MPHY PWM reset
1 LN0_OVRD_RX_PWM_AFC_RSTN	Override enable for rx_pwm_afc_rstn
0 LN0_RX_PWM_AFC_RSTN	RX MPHY PWM AFC reset

11.5.3.1.242 (TRSV_REG073)

11.5.3.1.242.1 Offset

Register	Offset
TRSV_REG073	5CCh

11.5.3.1.242.2 Diagram



11.5.3.1.242.3 Fields

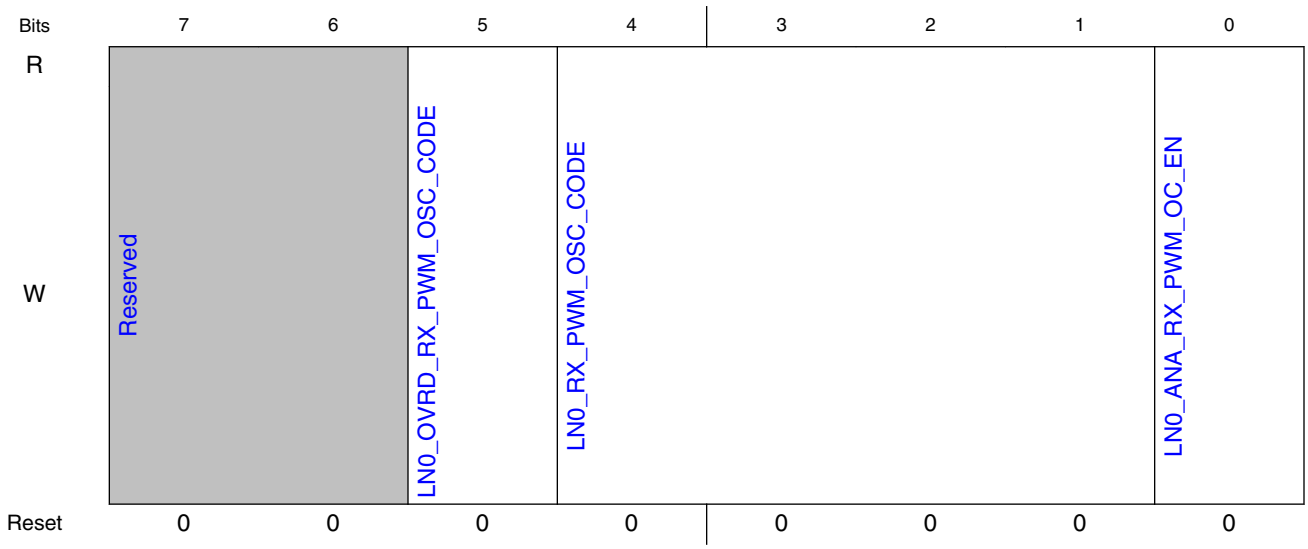
Field	Function
7-5 —	Reserved
4 LN0_OVRD_RX_PWM_AFC_DONE	Override enable for rx_pwm_afc_done
3 LN0_RX_PWM_AFC_DONE	RX MPHY PWM AFC done signal
2-0 LN0_ANA_RX_PWM_DIV_RATIO	RX MPHY PWM oversampling clock divide ratio from PWM oscillator 000: 001:

11.5.3.1.243 (TRSV_REG074)

11.5.3.1.243.1 Offset

Register	Offset
TRSV_REG074	5D0h

11.5.3.1.243.2 Diagram



11.5.3.1.243.3 Fields

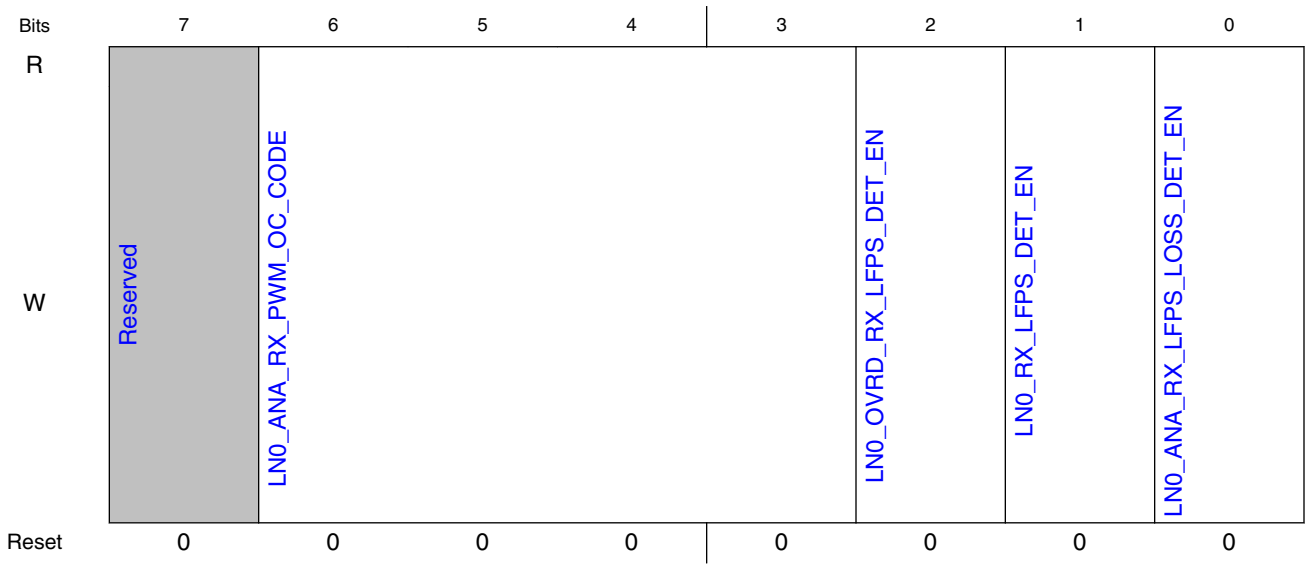
Field	Function
7-6 —	Reserved
5 LN0_OVRD_RX_PWM_OSC_CODE	Override enable for rx_pwm_osc_code
4-1 LN0_RX_PWM_OSC_CODE	RX MPHY PWM AFC code for oscillator
0 LN0_ANA_RX_PWM_OC_EN	

11.5.3.1.244 (TRSV_REG075)

11.5.3.1.244.1 Offset

Register	Offset
TRSV_REG075	5D4h

11.5.3.1.244.2 Diagram



11.5.3.1.244.3 Fields

Field	Function
7 —	Reserved
6-3 LN0_ANA_RX_PWM_OC_CODE	min(-7 or 0_111){maximum negative offset} - max(+7 or 1_111) {maximum positive offset} <3>: Sign bit <2:0> Offset magnitude bits{000 no offset, 111=max offset} 1_000(+0) or 0000(-0) are center code
2 LN0_OVRD_RX_LFPS_DET_EN	Override enable for rx_lfps_det_en
1 LN0_RX_LFPS_DET_EN	LFPS detector enable 0 : Disable, 1:Enable
0 LN0_ANA_RX_LFPS_LOSS_DET_EN	LFPS loss detector enable 0 : Disable, 1:Enable

11.5.3.1.245 (TRSV_REG076)

11.5.3.1.245.1 Offset

Register	Offset
TRSV_REG076	5D8h

11.5.3.1.245.2 Diagram

Bits	7	6	5	4	3	2	1	0
R								
W	LN0_OVRD_RX_BIAS_EN	LN0_RX_BIAS_EN	LN0_ANA_RX_BIAS_RMRES_CTRL			LN0_ANA_RX_SLB_EN	LN0_ANA_RX_LLB_EN	LN0_ANA_RX_SRLB_EN
Reset	0	0	0	1	0	0	0	0

11.5.3.1.245.3 Fields

Field	Function
7 LN0_OVRD_RX_BIAS_EN	Override enable for rx_bias_en
6 LN0_RX_BIAS_EN	RX bias current enable 0: Disable all RX bias current, 1: Enable
5-3 LN0_ANA_RX_BIAS_RMRES_CTRL	RX RMRES bias current control 000: 62%, 001: 87%, 010: 100% (Default), 100: 138%, 111: 156%
2 LN0_ANA_RX_SLB_EN	Serial loopback enable 0: Disable, 1: Enable (PMA digital → TX SER → RX CDR (DES) → PMA digital)
1 LN0_ANA_RX_LLB_EN	Line loopback enalble 0: Disable, 1: Enable (RX pad → RX AFE (CTLE) → TX driver → TX PAD)
0	Serial retimed loopback enable

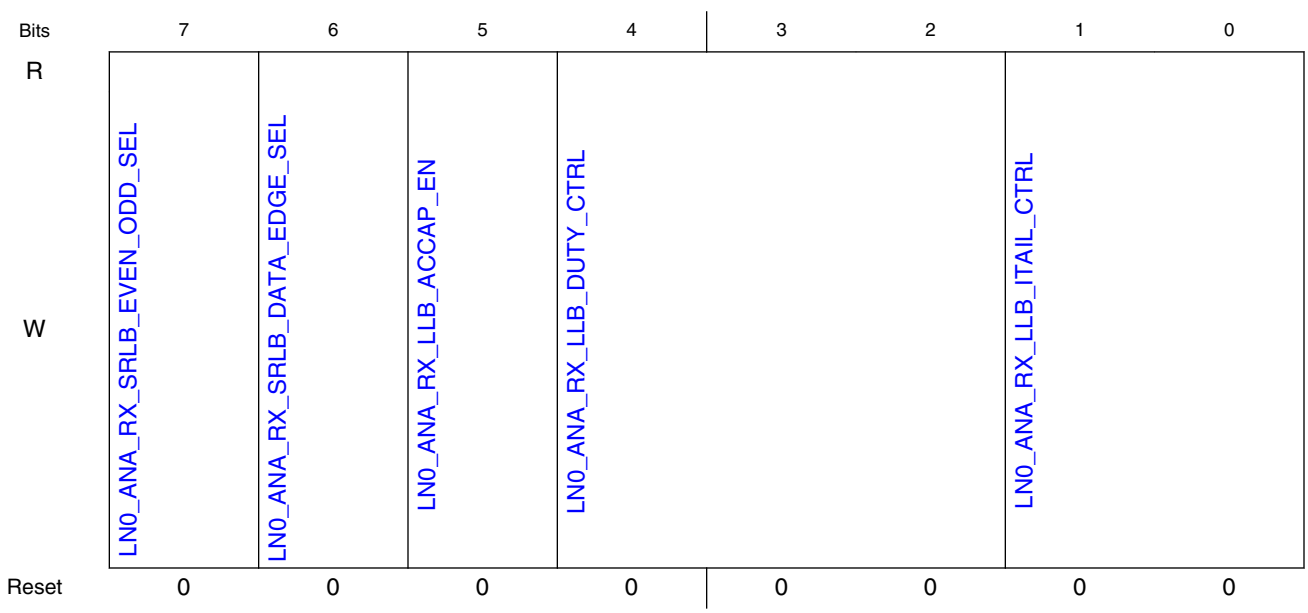
Field	Function
LN0_ANA_RX_SRLB_EN	0: Disable 1: Enable (RX pad → RX AFE → RX CDR (BBPD) → TX driver → TX pad)

11.5.3.1.246 (TRSV_REG077)

11.5.3.1.246.1 Offset

Register	Offset
TRSV_REG077	5DCh

11.5.3.1.246.2 Diagram



11.5.3.1.246.3 Fields

Field	Function
7 LN0_ANA_RX_SRLB_EVEN_ODD_SEL	Serial retimed loopback path selection 0: Even path, 1: Odd path
6	Serial retimed loopback path selection 0: Data path, 1: Edge path

Table continues on the next page...

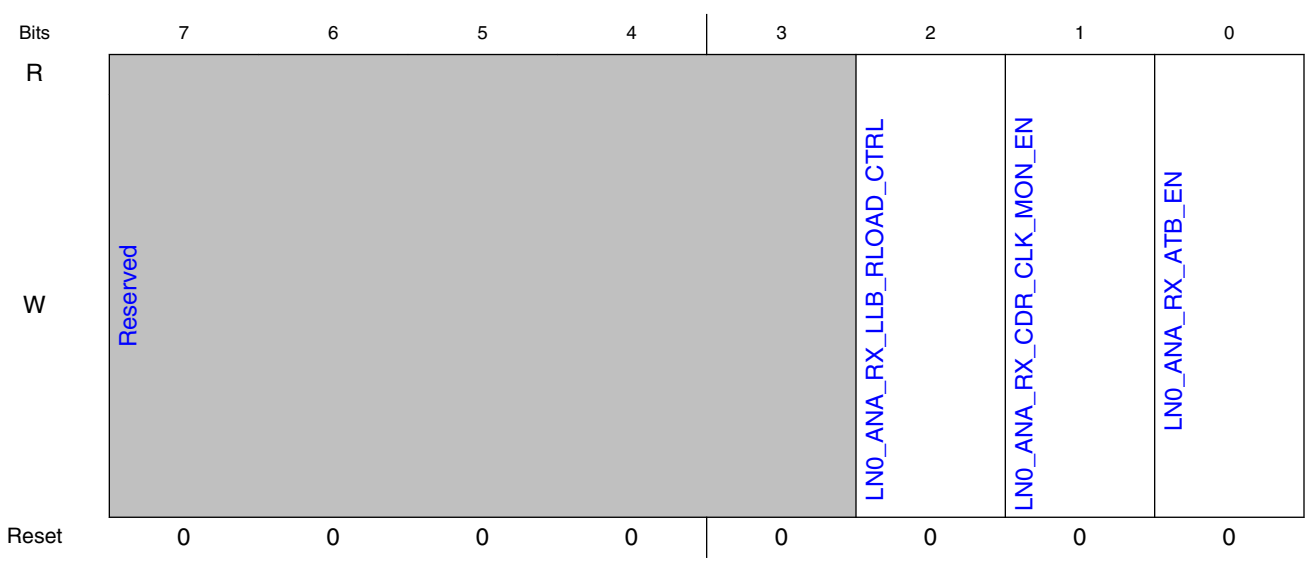
Field	Function
LN0_ANA_RX_SRLB_DATA_EDGE_SEL	
5 LN0_ANA_RX_LLB_ACCAP_EN	Line loopback path selection 0: SCL-to-CMOS, 1: AC-coupled
4-2 LN0_ANA_RX_LLB_DUTY_CTRL	Line loopback duty-ratio control 000:, 111:
1-0 LN0_ANA_RX_LLB_ITAIL_CTRL	Line loopback tail-current control 00:, 11:

11.5.3.1.247 (TRSV_REG078)

11.5.3.1.247.1 Offset

Register	Offset
TRSV_REG078	5E0h

11.5.3.1.247.2 Diagram



11.5.3.1.247.3 Fields

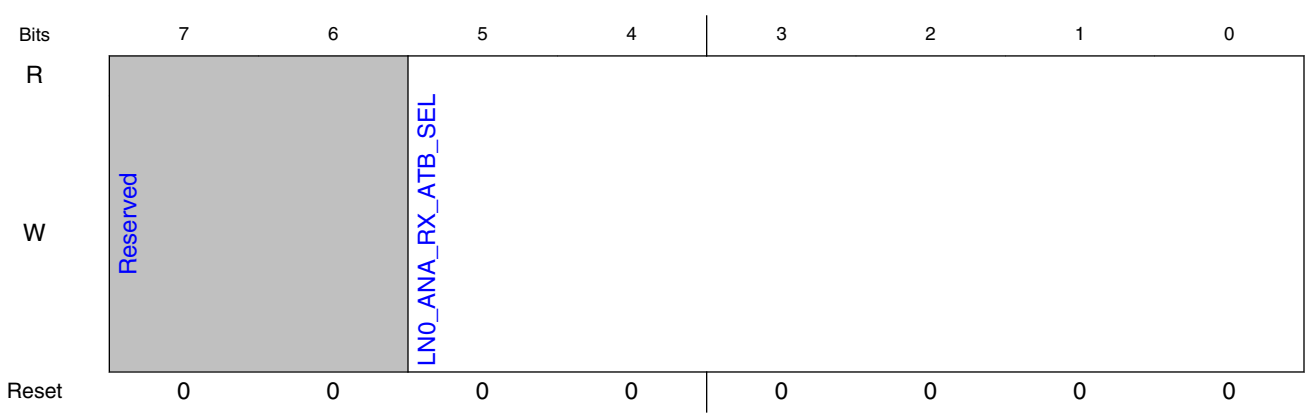
Field	Function
7-3 —	Reserved
2 LN0_ANA_RX_LLB_RLOAD_CTRL	Line loopback load resistance control 0:, 1:
1 LN0_ANA_RX_CDR_CLK_MON_EN	
0 LN0_ANA_RX_ATB_EN	RX ATB enable

11.5.3.1.248 (TRSV_REG079)

11.5.3.1.248.1 Offset

Register	Offset
TRSV_REG079	5E4h

11.5.3.1.248.2 Diagram



11.5.3.1.248.3 Fields

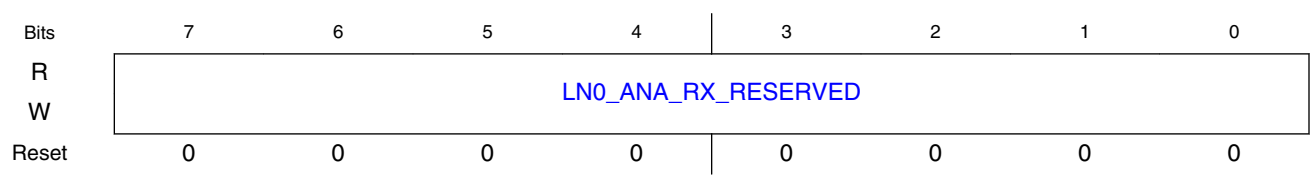
Field	Function
7-6 —	Reserved
5-0 LN0_ANA_RX_ATB_SEL	When i_sfr_rx_atb_en=1 and i_sfr_rx_atb_sel<5>=0, RX AFE nodes are under monitoring. According to i_sfr_rx_atb_sel<4:0>, 00000: LLB_SCL_INN, 00001: LLB_SCL_INP, 00010: OUTN, 00011: OUTP, 00100: VSS, 00101: VCM_INCMGENOUT, 00110: CTLE_VCM, 00111: CTLE_NBIAS, 01000: CTLE_PCAS, 01001: CTLE_PBIAS, 01010: SQOFSN, 01011: SQOFSP, 01100: SQ_PBIAS, 01101: VSS, 01110: VSS, 01111: VDD When i_sfr_rx_atb_en=1 and i_sfr_rx_atb_sel<5>=0, RX CDR nodes are under monitoring. According to i_sfr_rx_atb_sel<4:0>, 00000: VCI SET, 00001: VCM, 00010: VREG 820mV LPF, 00011: CP VDDH, 00100: CP FB, 00101: DCC VDD, 00110:DCC VSS, 00111: IQ-DIV VDD, 01000: IQ-DIV VSS, 01001: BBPD/DFE VDD, 01010: BBPD/DFE VSS, 01011: DFE VREFP, 01100: DFE VREFN, 01101: DFE VOFSP DE, 01110: DFE VOFSN DE, 01111: N/A

11.5.3.1.249 (TRSV_REG07A)

11.5.3.1.249.1 Offset

Register	Offset
TRSV_REG07A	5E8h

11.5.3.1.249.2 Diagram



11.5.3.1.249.3 Fields

Field	Function
7-0	Reserved port

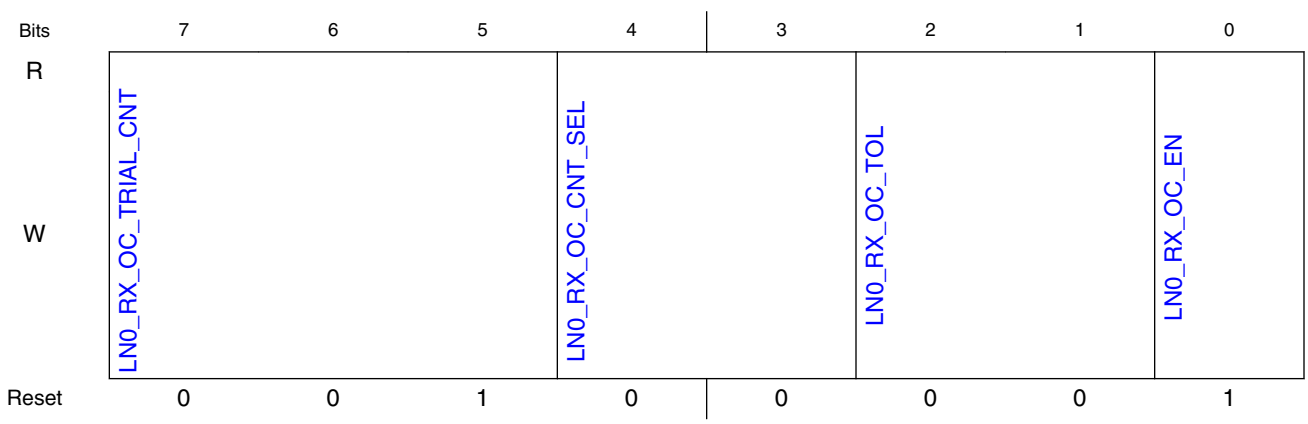
Field	Function
LN0_ANA_RX_RESERVED	

11.5.3.1.250 (TRSV_REG07B)

11.5.3.1.250.1 Offset

Register	Offset
TRSV_REG07B	5ECh

11.5.3.1.250.2 Diagram



11.5.3.1.250.3 Fields

Field	Function
7-5 LN0_RX_OC_TRIAL_CNT	RX offset calibration trial number selection 00: 2, 01: 4, 10: 6, 11: 8
4-3 LN0_RX_OC_CNT_SEL	RX SQ offset calibraiton counter selection for waiting time 00: 10, 01: 15, 20: 4, 11: 25
2-1 LN0_RX_OC_TOL	RX offset calibration enable 0: Disable, 1: Enable
0 LN0_RX_OC_EN	RX offset calibration tolerance for average value 00: 2, 01: 3, 10: 4, 11: 5

11.5.3.1.251 (TRSV_REG07C)

11.5.3.1.251.1 Offset

Register	Offset
TRSV_REG07C	5F0h

11.5.3.1.251.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_RX_OC_UPD_CNT_SEL		LN0_RX_OC_BYPASS_DFE_SA_ERR_ODD	LN0_RX_OC_BYPASS_DFE_SA_ERR_EVEN	LN0_RX_OC_BYPASS_DFE_SA_EDGE_ODD	LN0_RX_OC_BYPASS_DFE_SA_EDGE_EVEN	LN0_RX_OC_BYPASS_DFE_ADDER_ODD_FINAL	LN0_RX_OC_BYPASS_DFE_ADDER_EVEN_FINAL
W								
Reset	0	0	0	0	0	0	0	0

11.5.3.1.251.3 Fields

Field	Function
7-6 LN0_RX_OC_UPD_CNT_SEL	RX offset calibration code wating time selection for SA & CTLE only 00: 4, 01: 8, 10: 12, 11:16
5 LN0_RX_OC_BYPASS_DFE_SA_ERR_ODD	Bypass

Table continues on the next page...

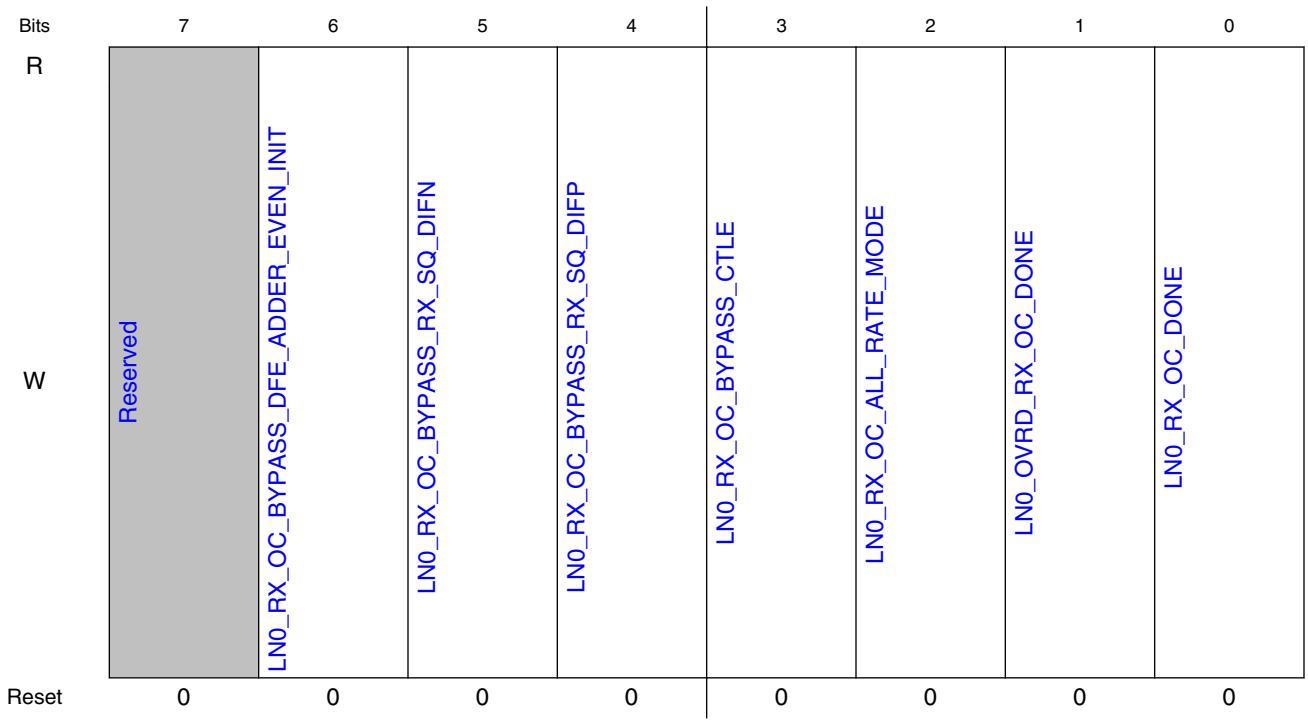
Field	Function
4 LN0_RX_OC_BYPASS_DFE_SA_ERR_EVEN	
3 LN0_RX_OC_BYPASS_DFE_SA_EDGE_ODD	
2 LN0_RX_OC_BYPASS_DFE_SA_EDGE_EVEN	
1 LN0_RX_OC_BYPASS_DFE_ADDER_ODD_FINAL	
0 LN0_RX_OC_BYPASS_DFE_ADDER_EVEN_FINAL	

11.5.3.1.252 (TRSV_REG07D)

11.5.3.1.252.1 Offset

Register	Offset
TRSV_REG07D	5F4h

11.5.3.1.252.2 Diagram



11.5.3.1.252.3 Fields

Field	Function
7 —	Reserved
6 LN0_RX_OC_BYPASS_DFE_ADDER_EVEN_INIT	
5 LN0_RX_OC_BYPASS_RX_SQ_DIFN	
4 LN0_RX_OC_BYPASS_RX_SQ_DIFP	
3 LN0_RX_OC_BYPASS_CTLE	Bypass offset calibration for CTLE
2	

Table continues on the next page...

Field	Function
LN0_RX_OC_ALL_RATE_MODE	
1 LN0_OVRD_RX_OC_DONE	Override enable for rx_oc_done
0 LN0_RX_OC_DONE	RX offset calibration overide value

11.5.3.1.253 (TRSV_REG07E)

11.5.3.1.253.1 Offset

Register	Offset
TRSV_REG07E	5F8h

11.5.3.1.253.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_RX_SSLMS_C0_INIT							
Reset	1	1	0	0	0	0	0	0

11.5.3.1.253.3 Fields

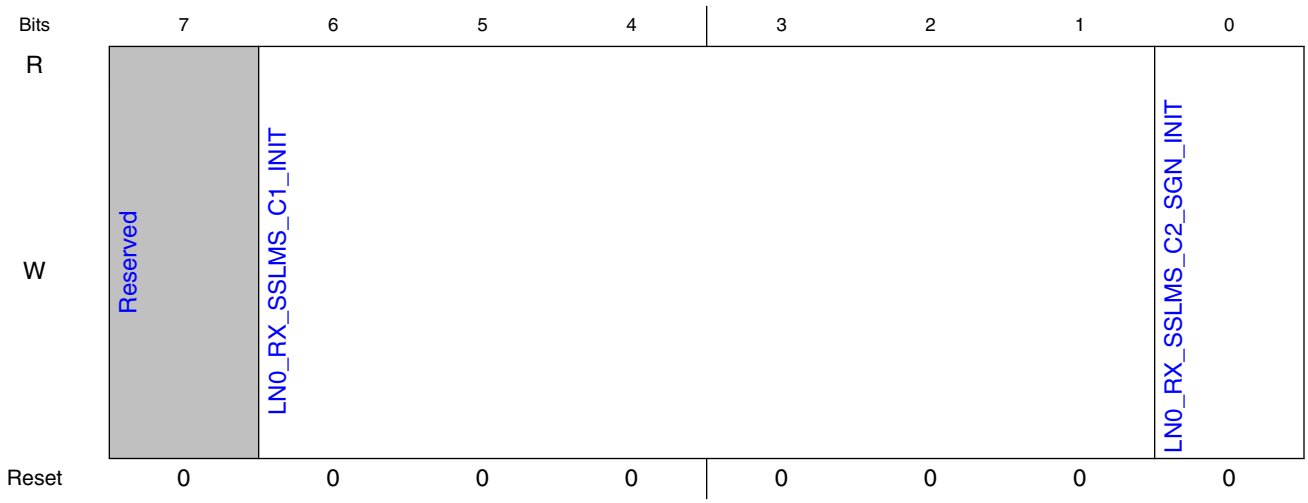
Field	Function
7-0 LN0_RX_SSLMS_C0_INIT	

11.5.3.1.254 (TRSV_REG07F)

11.5.3.1.254.1 Offset

Register	Offset
TRSV_REG07F	5FCh

11.5.3.1.254.2 Diagram



11.5.3.1.254.3 Fields

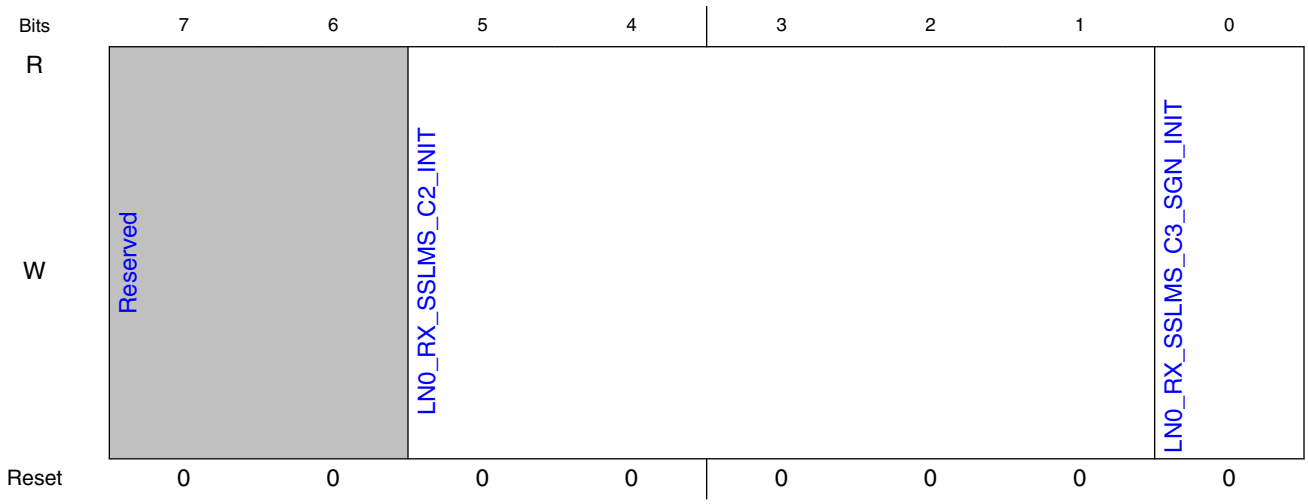
Field	Function
7 —	Reserved
6-1 LN0_RX_SSLMS_C1_INIT	
0 LN0_RX_SSLMS_C2_SGN_INIT	

11.5.3.1.255 (TRSV_REG080)

11.5.3.1.255.1 Offset

Register	Offset
TRSV_REG080	600h

11.5.3.1.255.2 Diagram



11.5.3.1.255.3 Fields

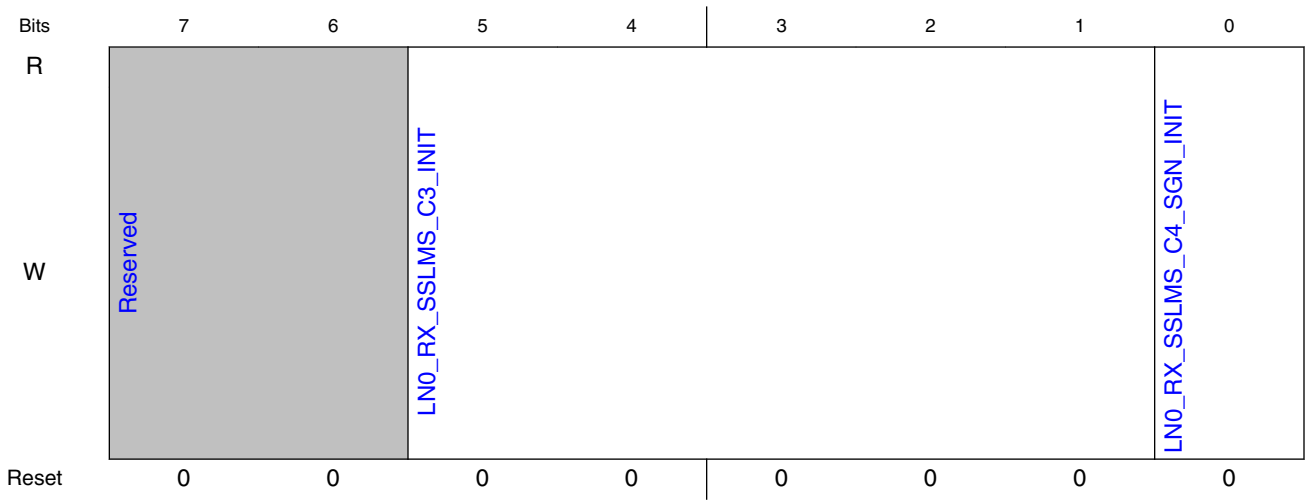
Field	Function
7-6 —	Reserved
5-1 LN0_RX_SSLMS_C2_INIT	
0 LN0_RX_SSLMS_C3_SGN_INIT	

11.5.3.1.256 (TRSV_REG081)

11.5.3.1.256.1 Offset

Register	Offset
TRSV_REG081	604h

11.5.3.1.256.2 Diagram



11.5.3.1.256.3 Fields

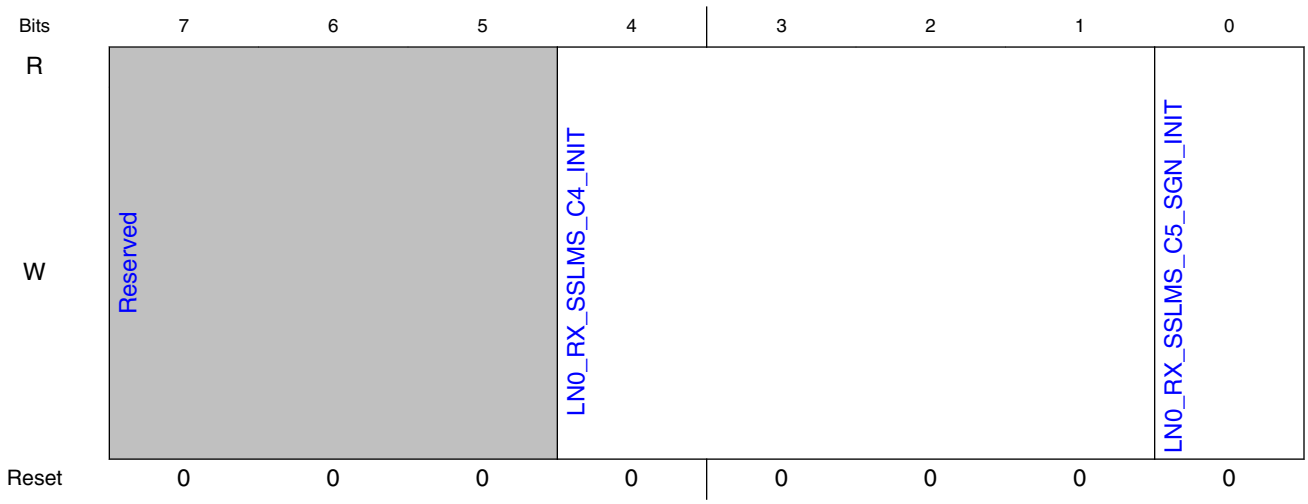
Field	Function
7-6 —	Reserved
5-1 LN0_RX_SSLMS_C3_INIT	
0 LN0_RX_SSLMS_C4_SGN_INIT	

11.5.3.1.257 (TRSV_REG082)

11.5.3.1.257.1 Offset

Register	Offset
TRSV_REG082	608h

11.5.3.1.257.2 Diagram



11.5.3.1.257.3 Fields

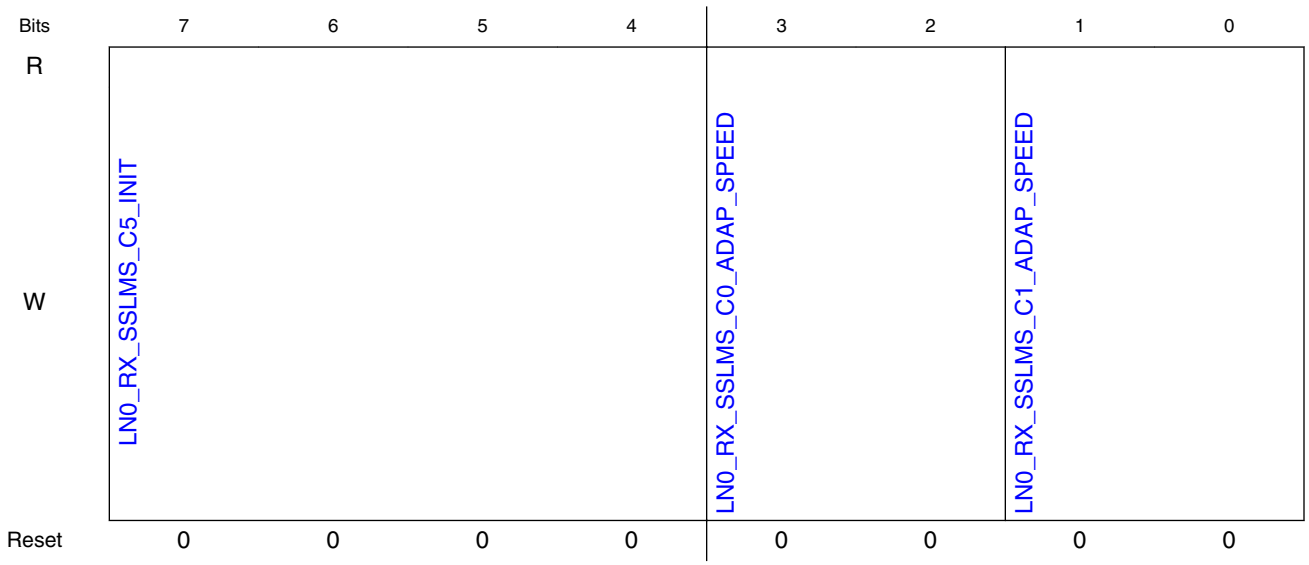
Field	Function
7-5 —	Reserved
4-1 LN0_RX_SSLMS_C4_INIT	
0 LN0_RX_SSLMS_C5_SGN_INIT	

11.5.3.1.258 (TRSV_REG083)

11.5.3.1.258.1 Offset

Register	Offset
TRSV_REG083	60Ch

11.5.3.1.258.2 Diagram



11.5.3.1.258.3 Fields

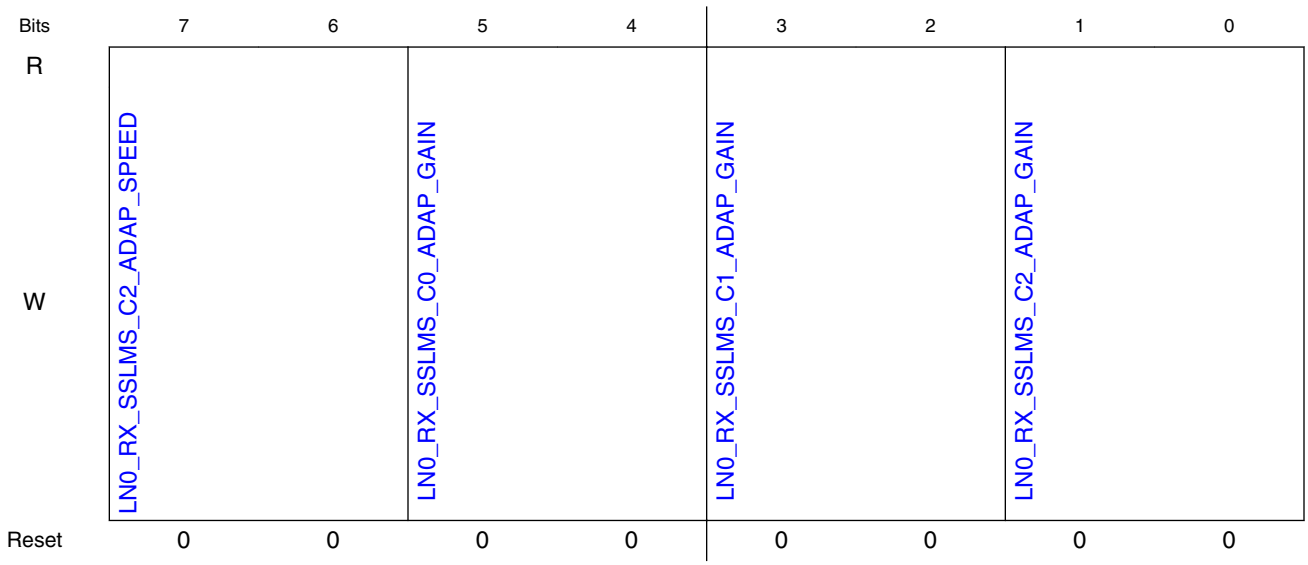
Field	Function
7-4 LN0_RX_SSLMS_C5_INIT	
3-2 LN0_RX_SSLMS_C0_ADAP_SPEED	RX DFE SSLMS c0 adaptation speed selection 00: 1/1024, 01: 1/2048, 10: 1/4096, 11: 1/8192
1-0 LN0_RX_SSLMS_C1_ADAP_SPEED	RX DFE SSLMS c1 adaptation speed selection 00: 1/1024, 01: 1/2048, 10: 1/4096, 11: 1/8192

11.5.3.1.259 (TRSV_REG084)

11.5.3.1.259.1 Offset

Register	Offset
TRSV_REG084	610h

11.5.3.1.259.2 Diagram



11.5.3.1.259.3 Fields

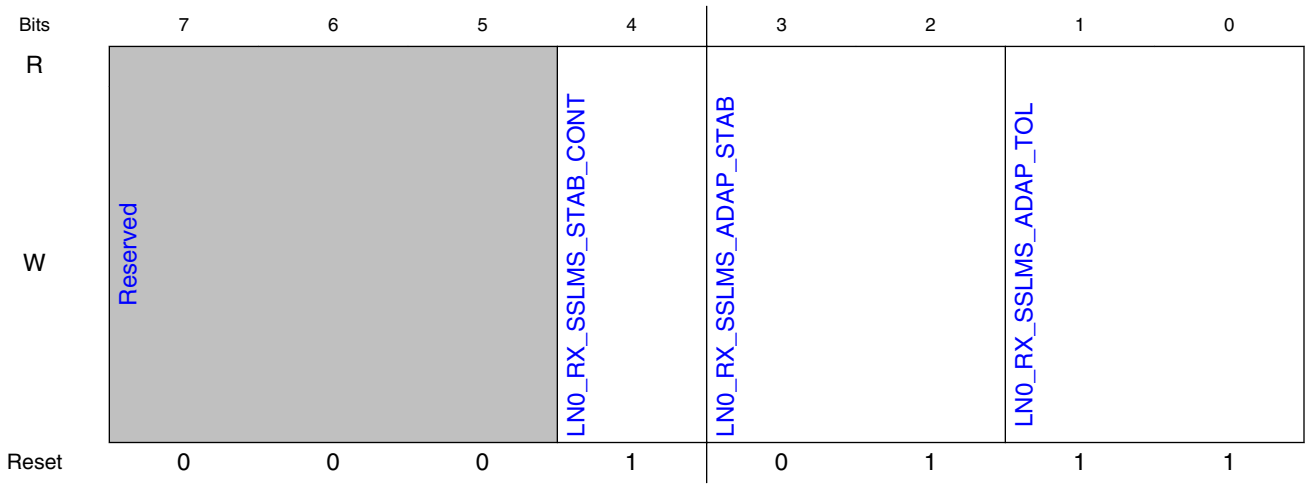
Field	Function
7-6 LN0_RX_SSLMS_C2_ADAP_SPEED	RX DFE SSLMS c2 adaptation speed selection 00: 1/1024, 01: 1/2048, 10: 1/4096, 11: 1/8192
5-4 LN0_RX_SSLMS_C0_ADAP_GAIN	
3-2 LN0_RX_SSLMS_C1_ADAP_GAIN	
1-0 LN0_RX_SSLMS_C2_ADAP_GAIN	

11.5.3.1.260 (TRSV_REG085)

11.5.3.1.260.1 Offset

Register	Offset
TRSV_REG085	614h

11.5.3.1.260.2 Diagram



11.5.3.1.260.3 Fields

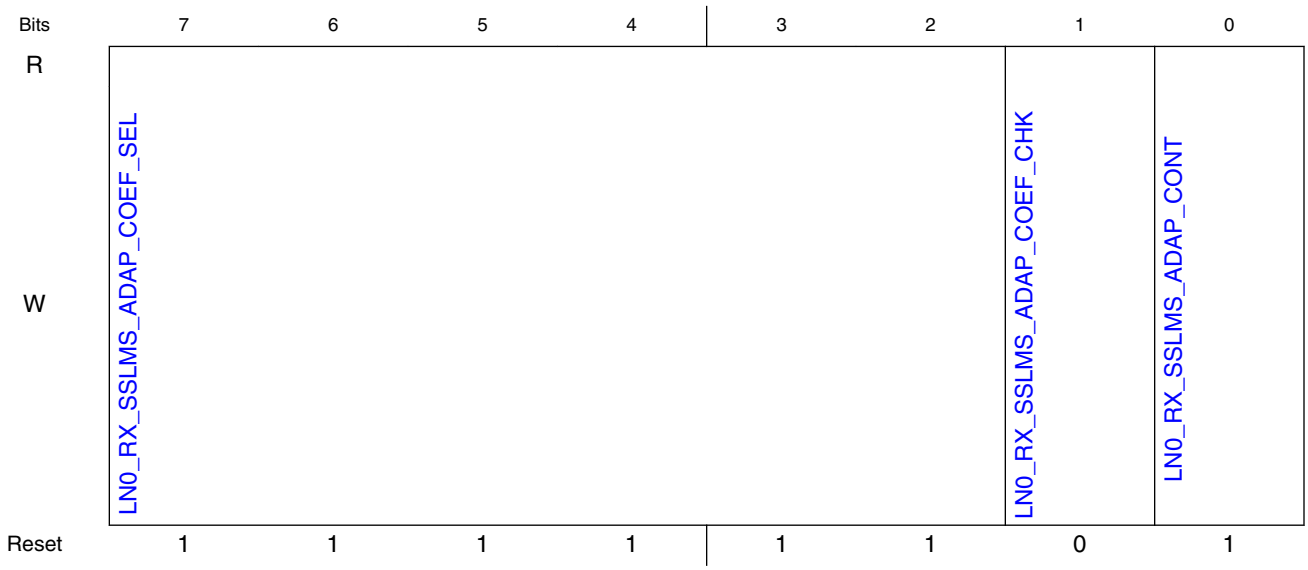
Field	Function
7-5 —	Reserved
4 LN0_RX_SSLMS_STAB_CONT	
3-2 LN0_RX_SSLMS_ADAP_STAB	
1-0 LN0_RX_SSLMS_ADAP_TOL	

11.5.3.1.261 (TRSV_REG086)

11.5.3.1.261.1 Offset

Register	Offset
TRSV_REG086	618h

11.5.3.1.261.2 Diagram



11.5.3.1.261.3 Fields

Field	Function
7-2 LN0_RX_SSLMS_ADAP_COEF_SEL	
1 LN0_RX_SSLMS_ADAP_COEF_CHK	
0 LN0_RX_SSLMS_ADAP_CONT	

11.5.3.1.262 (TRSV_REG087)

11.5.3.1.262.1 Offset

Register	Offset
TRSV_REG087	61Ch

11.5.3.1.262.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_RX_SSLMS_ADAP_TIMEOUT_SEL		LN0_RX_SSLMS_ADAP_TIMEOUT_EN	LN0_RX_SSLMS_RSTN	LN0_OVRD_RX_SSLMS_ADAP_EN	LN0_RX_SSLMS_ADAP_EN	LN0_OVRD_RX_SSLMS_ADAP_HOLD	LN0_RX_SSLMS_ADAP_HOLD
W								
Reset	0	0	1	1	0	0	0	0

11.5.3.1.262.3 Fields

Field	Function
7-6 LN0_RX_SSLMS_ADAP_TIMEOUT_SEL	
5 LN0_RX_SSLMS_ADAP_TIMEOUT_EN	
4 LN0_RX_SSLMS_RSTN	
3 LN0_OVRD_RX_SSLMS_ADAP_EN	Override enable for rx_sslms_adap_en
2	

Table continues on the next page...

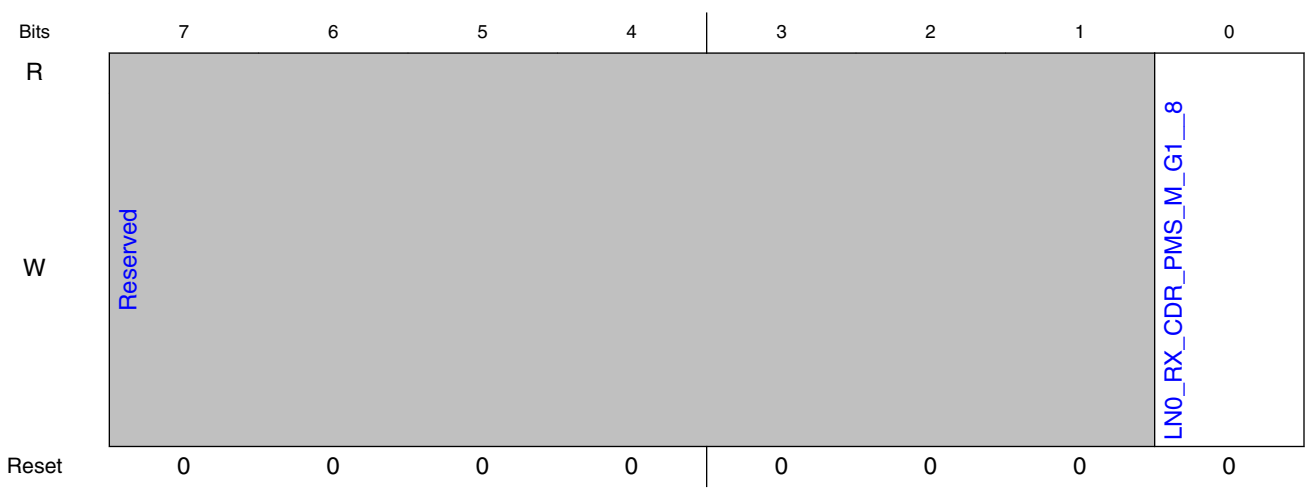
Field	Function
LN0_RX_SSLMS_ADAP_EN	
1 LN0_OVRD_RX_SSLMS_ADAP_HOLD	Override enable for rx_sslms_adap_hold
0 LN0_RX_SSLMS_ADAP_HOLD	

11.5.3.1.263 (TRSV_REG088)

11.5.3.1.263.1 Offset

Register	Offset
TRSV_REG088	620h

11.5.3.1.263.2 Diagram



11.5.3.1.263.3 Fields

Field	Function
7-1 —	Reserved
0	

Field	Function
LN0_RX_CDR_PMS_M_G1__8	

11.5.3.1.264 (TRSV_REG089)

11.5.3.1.264.1 Offset

Register	Offset
TRSV_REG089	624h

11.5.3.1.264.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_RX_CDR_PMS_M_G1__7_0							
Reset	1	1	0	0	1	0	0	0

11.5.3.1.264.3 Fields

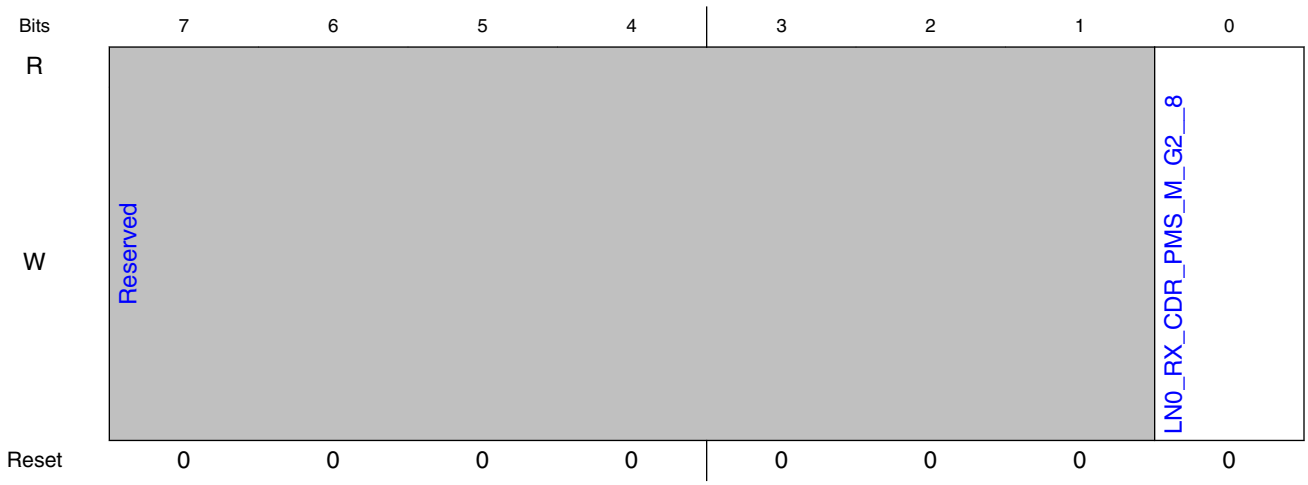
Field	Function
7-0 LN0_RX_CDR_PMS_M_G1__7_0	

11.5.3.1.265 (TRSV_REG08A)

11.5.3.1.265.1 Offset

Register	Offset
TRSV_REG08A	628h

11.5.3.1.265.2 Diagram



11.5.3.1.265.3 Fields

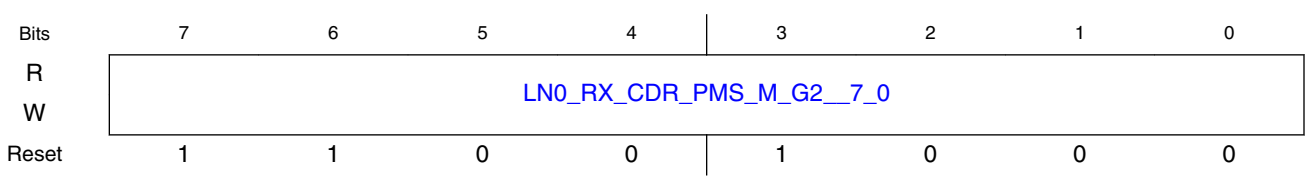
Field	Function
7-1 —	Reserved
0 LN0_RX_CDR_PMS_M_G2__8	

11.5.3.1.266 (TRSV_REG08B)

11.5.3.1.266.1 Offset

Register	Offset
TRSV_REG08B	62Ch

11.5.3.1.266.2 Diagram



11.5.3.1.266.3 Fields

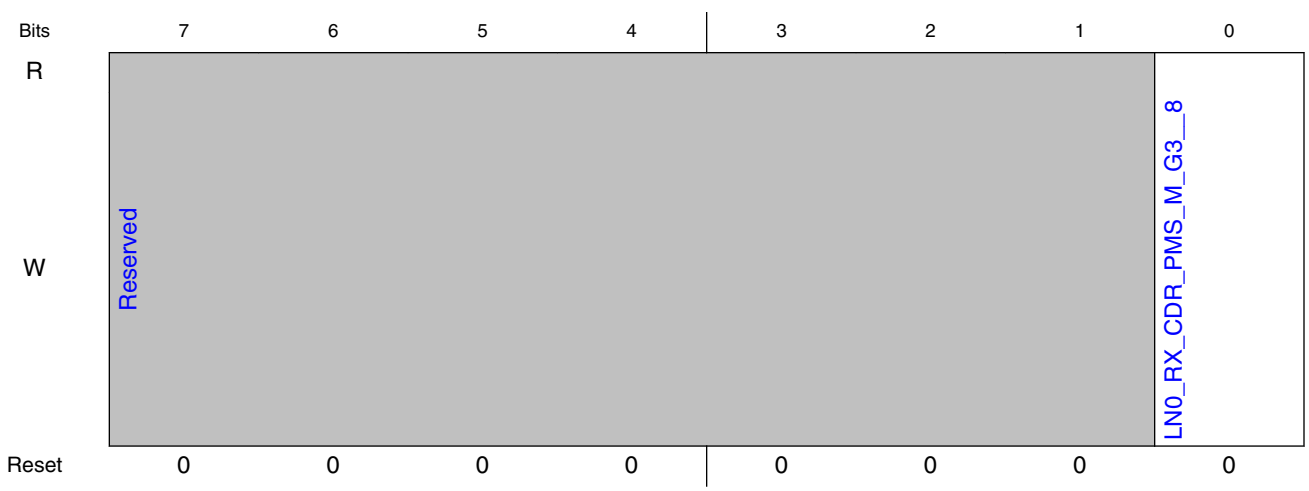
Field	Function
7-0 LN0_RX_CDR_PMS_M_G2__7_0	

11.5.3.1.267 (TRSV_REG08C)

11.5.3.1.267.1 Offset

Register	Offset
TRSV_REG08C	630h

11.5.3.1.267.2 Diagram



11.5.3.1.267.3 Fields

Field	Function
7-1 —	Reserved
0 LN0_RX_CDR_PMS_M_G3__8	

11.5.3.1.268 (TRSV_REG08D)

11.5.3.1.268.1 Offset

Register	Offset
TRSV_REG08D	634h

11.5.3.1.268.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_RX_CDR_PMS_M_G3__7_0							
Reset	1	0	1	0	0	0	0	0

11.5.3.1.268.3 Fields

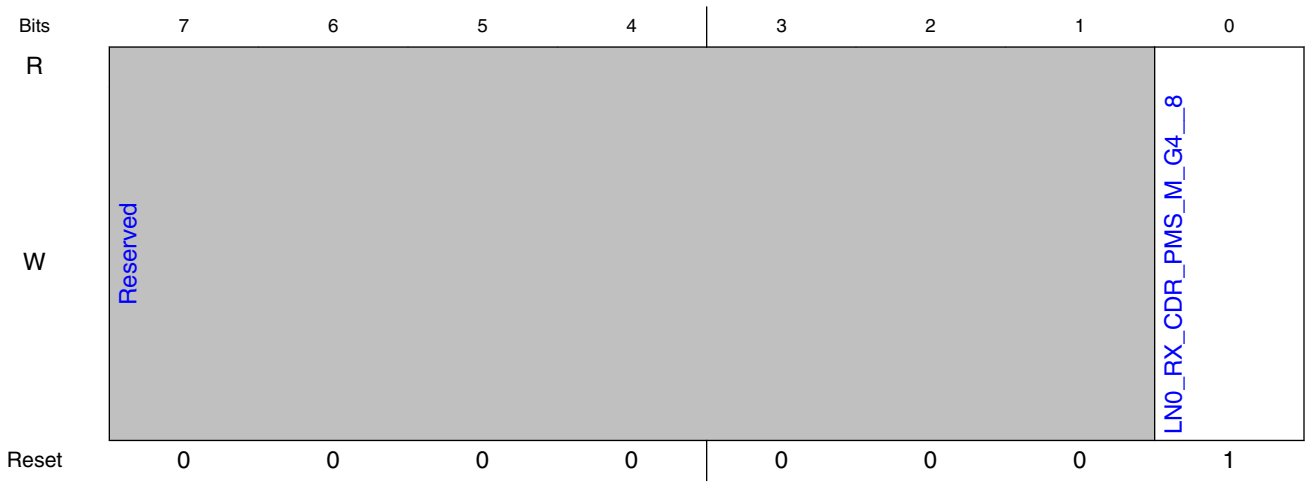
Field	Function
7-0 LN0_RX_CDR_PMS_M_G3__7_0	

11.5.3.1.269 (TRSV_REG08E)

11.5.3.1.269.1 Offset

Register	Offset
TRSV_REG08E	638h

11.5.3.1.269.2 Diagram



11.5.3.1.269.3 Fields

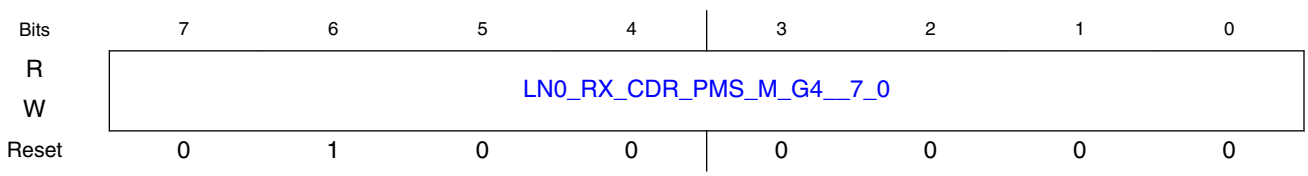
Field	Function
7-1 —	Reserved
0 LN0_RX_CDR_PMS_M_G4__8	

11.5.3.1.270 (TRSV_REG08F)

11.5.3.1.270.1 Offset

Register	Offset
TRSV_REG08F	63Ch

11.5.3.1.270.2 Diagram



11.5.3.1.270.3 Fields

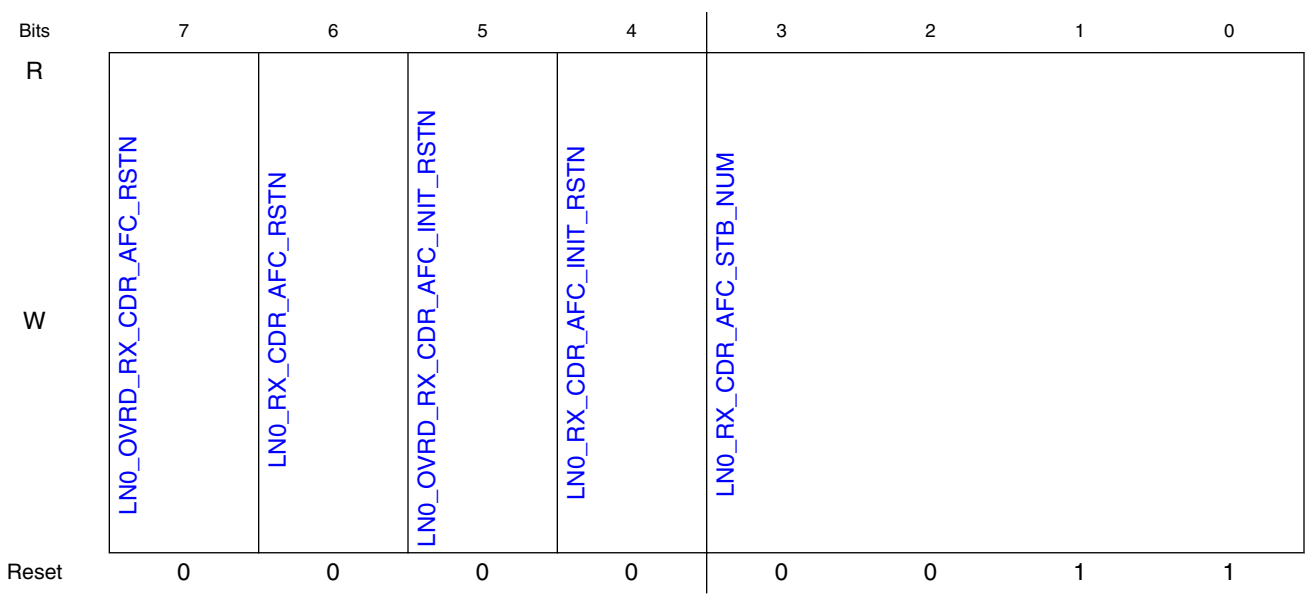
Field	Function
7-0 LN0_RX_CDR_PMS_M_G4__7_0	

11.5.3.1.271 (TRSV_REG090)

11.5.3.1.271.1 Offset

Register	Offset
TRSV_REG090	640h

11.5.3.1.271.2 Diagram



11.5.3.1.271.3 Fields

Field	Function
7	Override enable for rx_cdr_afc_rstn

Table continues on the next page...

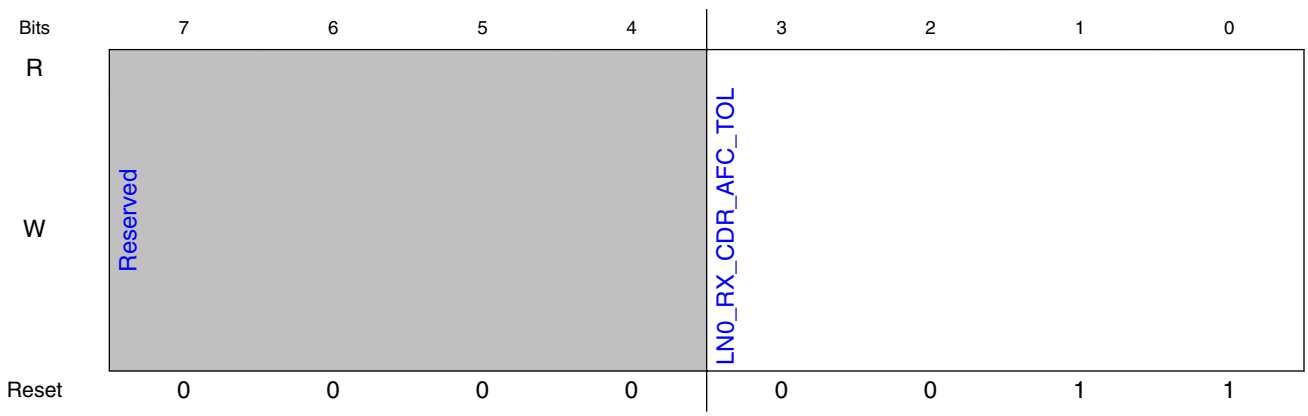
Field	Function
LN0_OVRD_RX_CDR_AFC_RSTN	
6 LN0_RX_CDR_AFC_RSTN	
5 LN0_OVRD_RX_CDR_AFC_INIT_RSTN	Override enable for rx_cdr_afc_init_rstn
4 LN0_RX_CDR_AFC_INIT_RSTN	
3-0 LN0_RX_CDR_AFC_STB_NUM	

11.5.3.1.272 (TRSV_REG091)

11.5.3.1.272.1 Offset

Register	Offset
TRSV_REG091	644h

11.5.3.1.272.2 Diagram



11.5.3.1.272.3 Fields

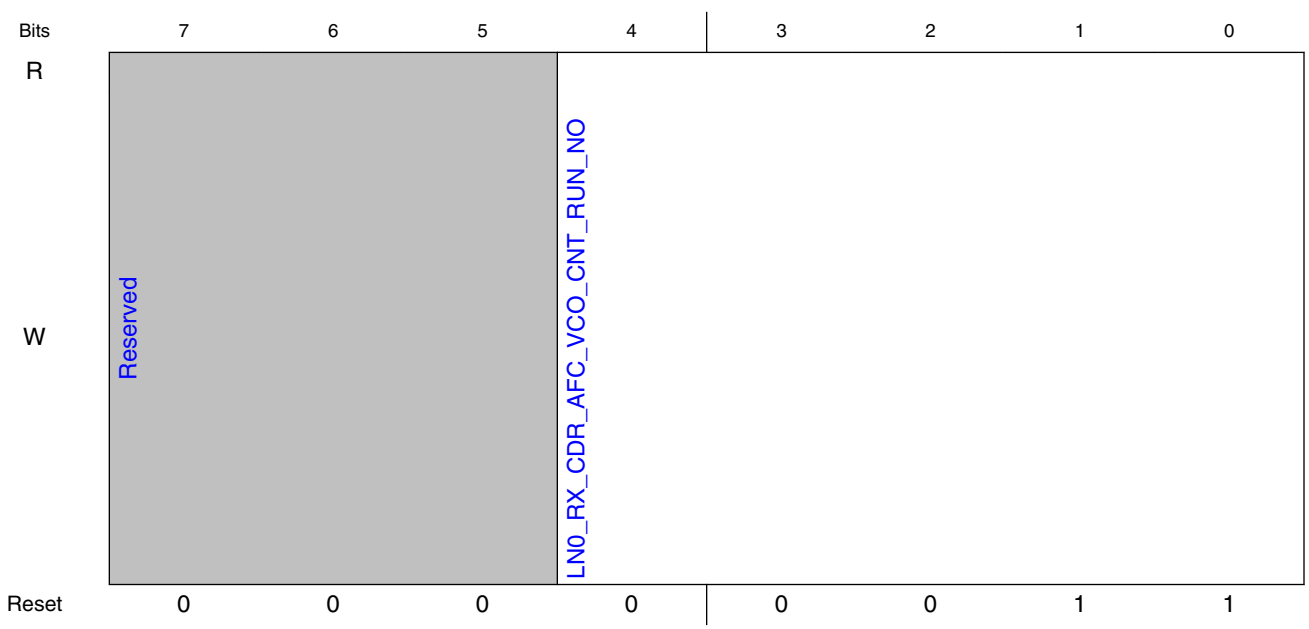
Field	Function
7-4 —	Reserved
3-0 LN0_RX_CDR_AFC_TOL	

11.5.3.1.273 (TRSV_REG092)

11.5.3.1.273.1 Offset

Register	Offset
TRSV_REG092	648h

11.5.3.1.273.2 Diagram



11.5.3.1.273.3 Fields

Field	Function
7-5	Reserved

Table continues on the next page...

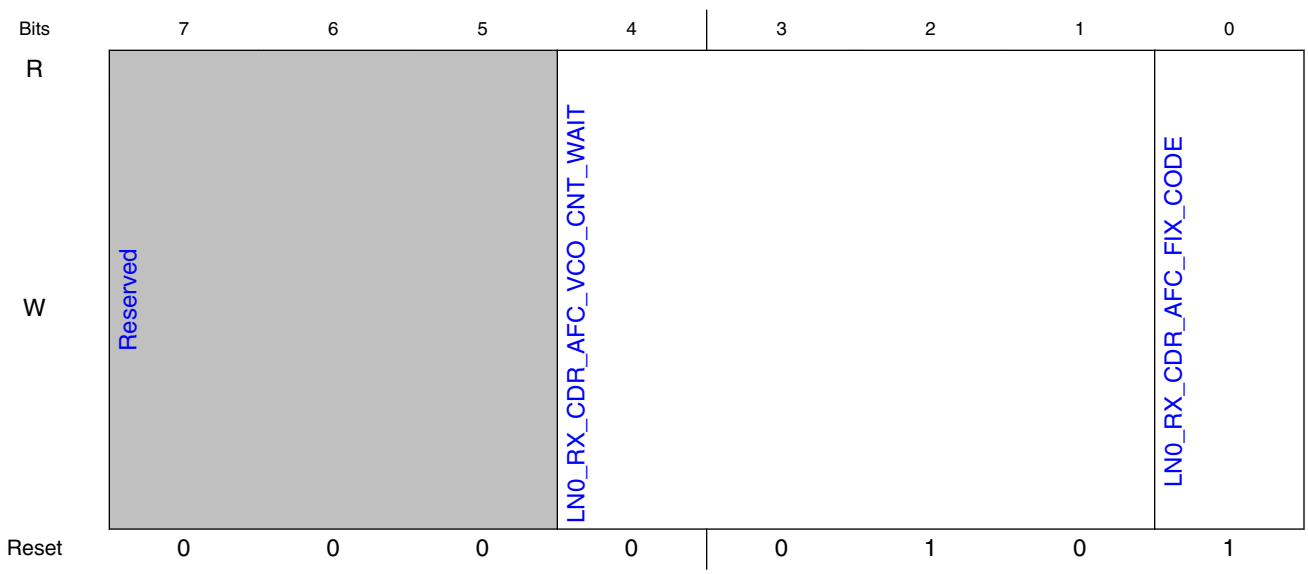
Field	Function
—	
4-0 LN0_RX_CDR_AFC_VCO_CNT_RUN_NO	

11.5.3.1.274 (TRSV_REG093)

11.5.3.1.274.1 Offset

Register	Offset
TRSV_REG093	64Ch

11.5.3.1.274.2 Diagram



11.5.3.1.274.3 Fields

Field	Function
7-5 —	Reserved
4-1	

Table continues on the next page...

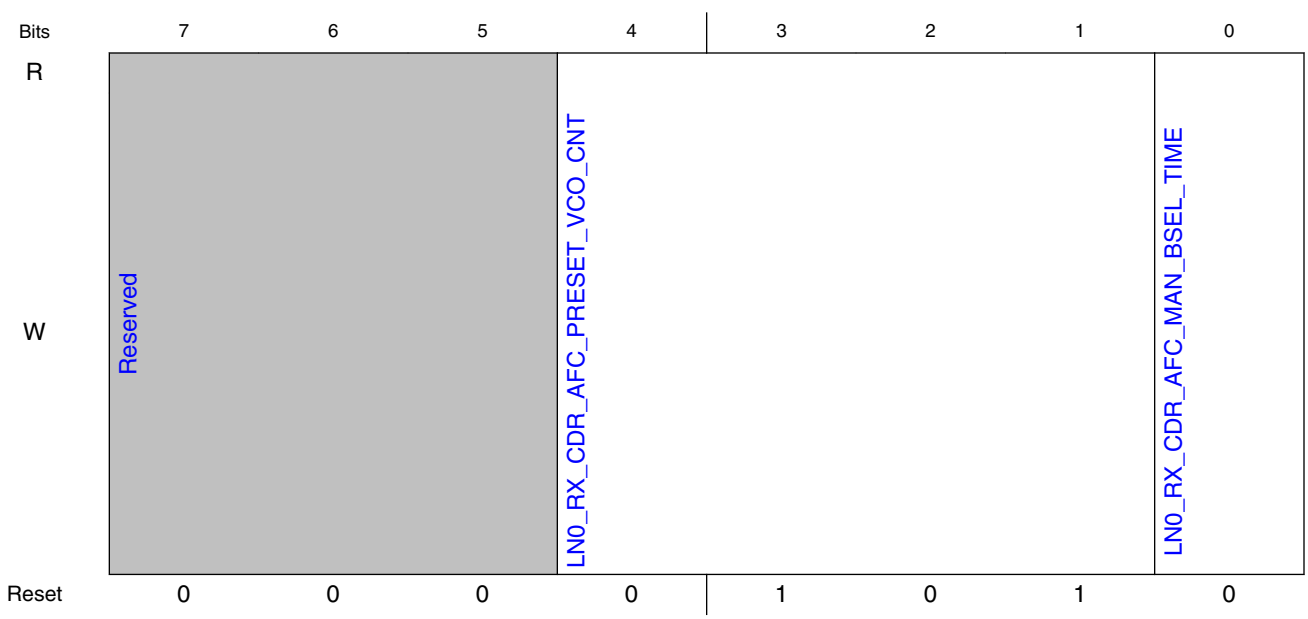
Field	Function
LN0_RX_CDR_AFC_VCO_CNT_WAIT	
0 LN0_RX_CDR_AFC_FIX_CODE	

11.5.3.1.275 (TRSV_REG094)

11.5.3.1.275.1 Offset

Register	Offset
TRSV_REG094	650h

11.5.3.1.275.2 Diagram



11.5.3.1.275.3 Fields

Field	Function
7-5 —	Reserved

Table continues on the next page...

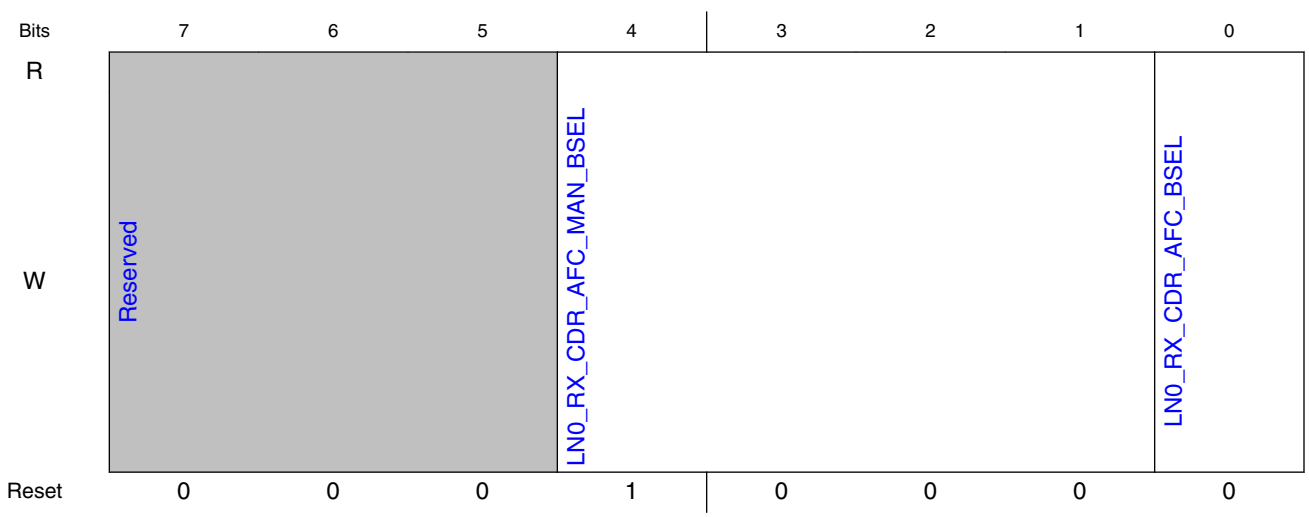
Field	Function
4-1 LN0_RX_CDR_AFC_PRESET_VCO_CNT	
0 LN0_RX_CDR_AFC_MAN_BSEL_TIME	

11.5.3.1.276 (TRSV_REG095)

11.5.3.1.276.1 Offset

Register	Offset
TRSV_REG095	654h

11.5.3.1.276.2 Diagram



11.5.3.1.276.3 Fields

Field	Function
7-5 —	Reserved
4-1	

Table continues on the next page...

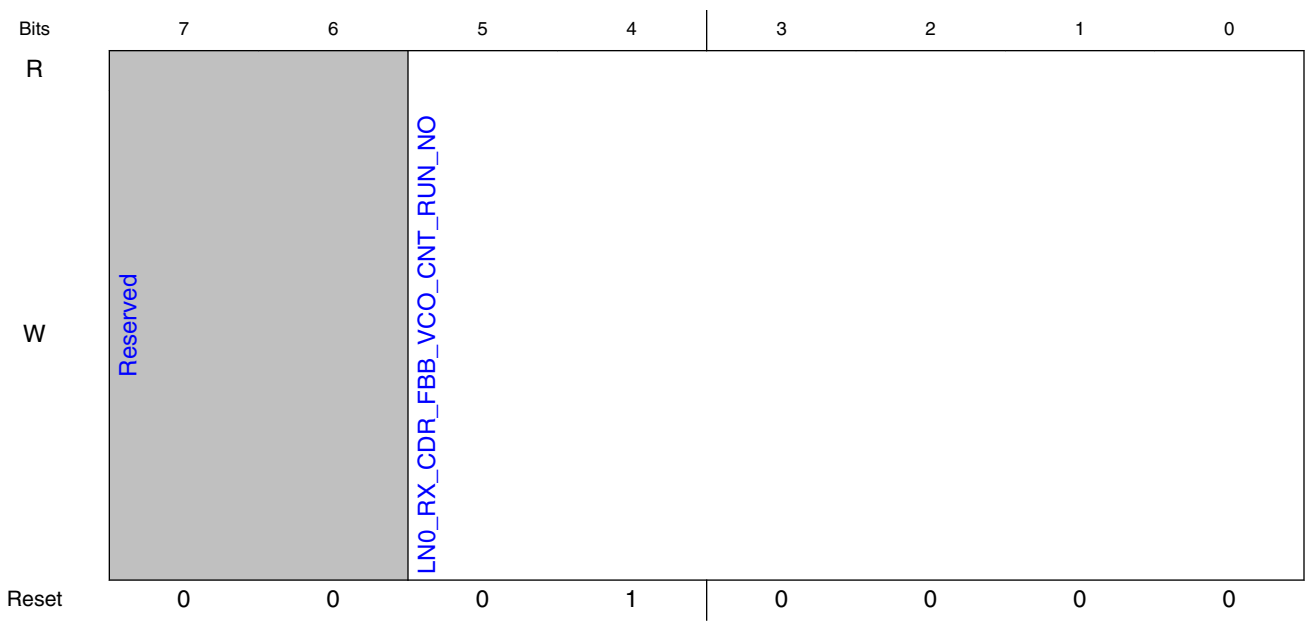
Field	Function
LN0_RX_CDR_AFC_MAN_BSEL	
0 LN0_RX_CDR_AFC_BSEL	

11.5.3.1.277 (TRSV_REG096)

11.5.3.1.277.1 Offset

Register	Offset
TRSV_REG096	658h

11.5.3.1.277.2 Diagram



11.5.3.1.277.3 Fields

Field	Function
7-6 —	Reserved

Table continues on the next page...

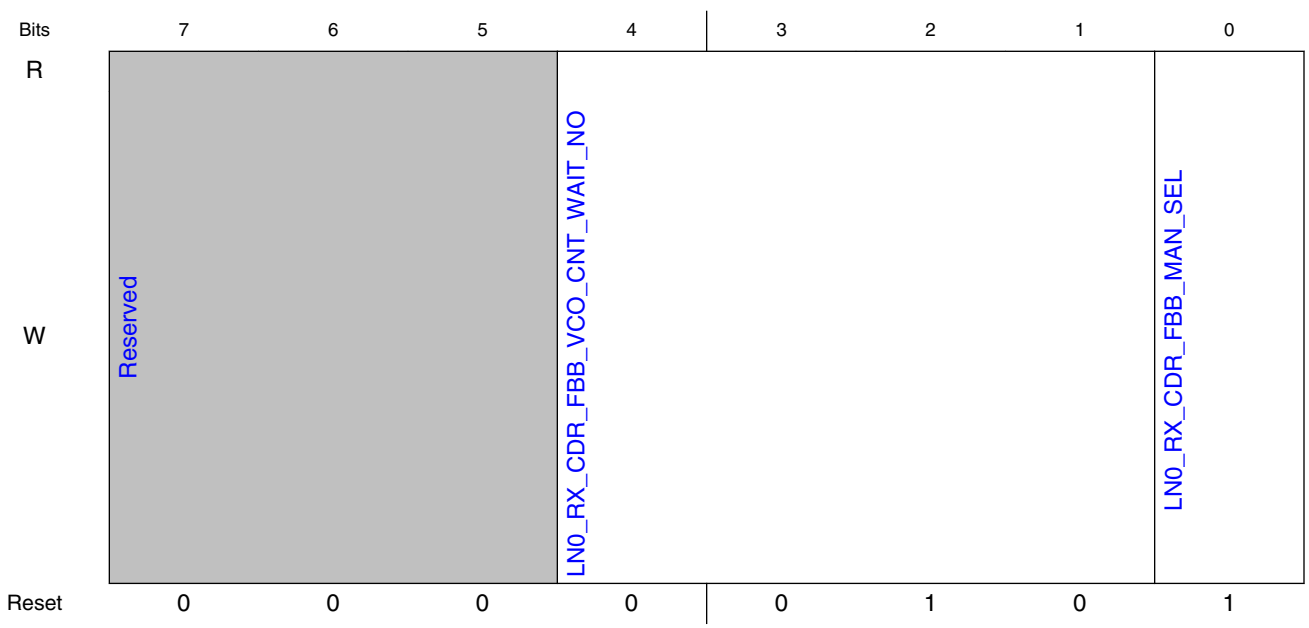
Field	Function
5-0 LN0_RX_CDR_FBB_VCO_CNT_RUN_NO	

11.5.3.1.278 (TRSV_REG097)

11.5.3.1.278.1 Offset

Register	Offset
TRSV_REG097	65Ch

11.5.3.1.278.2 Diagram



11.5.3.1.278.3 Fields

Field	Function
7-5 —	Reserved
4-1	

Table continues on the next page...

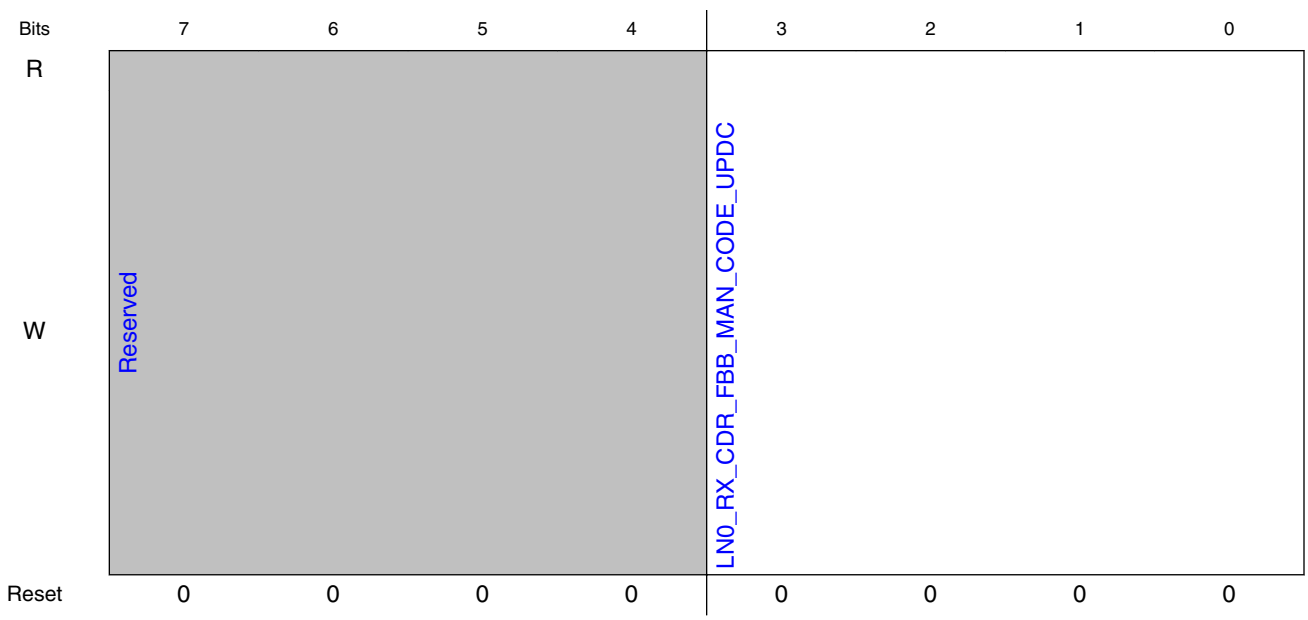
Field	Function
LN0_RX_CDR_FBB_VCO_CNT_WAIT_NO	
0 LN0_RX_CDR_FBB_MAN_SEL	

11.5.3.1.279 (TRSV_REG098)

11.5.3.1.279.1 Offset

Register	Offset
TRSV_REG098	660h

11.5.3.1.279.2 Diagram



11.5.3.1.279.3 Fields

Field	Function
7-4 —	Reserved

Table continues on the next page...

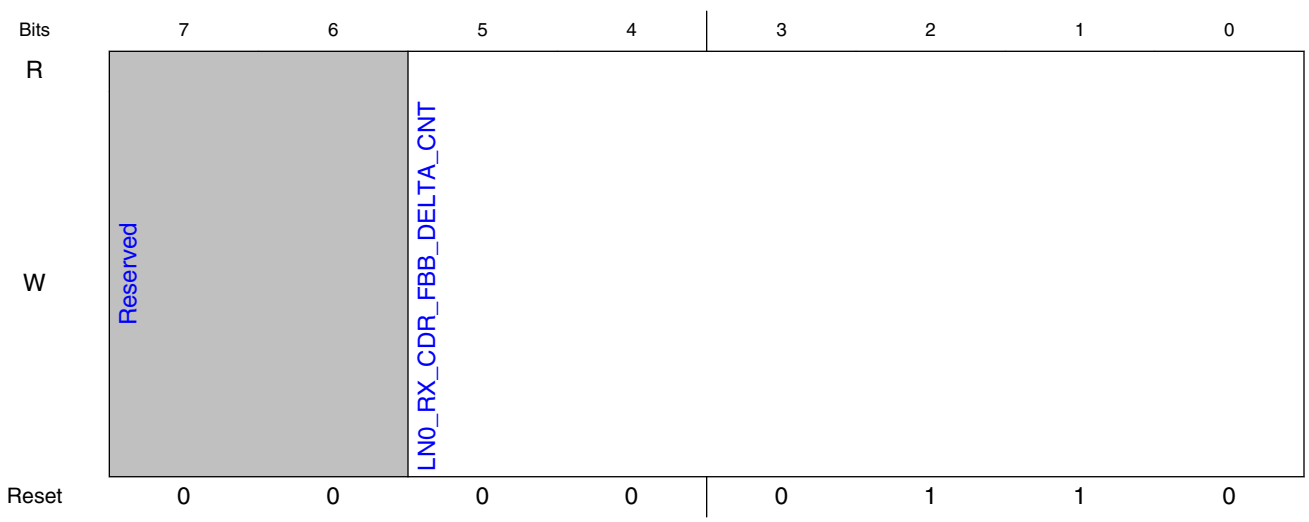
Field	Function
3-0 LN0_RX_CDR_FBB_MAN_CODE_UPDC	

11.5.3.1.280 (TRSV_REG099)

11.5.3.1.280.1 Offset

Register	Offset
TRSV_REG099	664h

11.5.3.1.280.2 Diagram



11.5.3.1.280.3 Fields

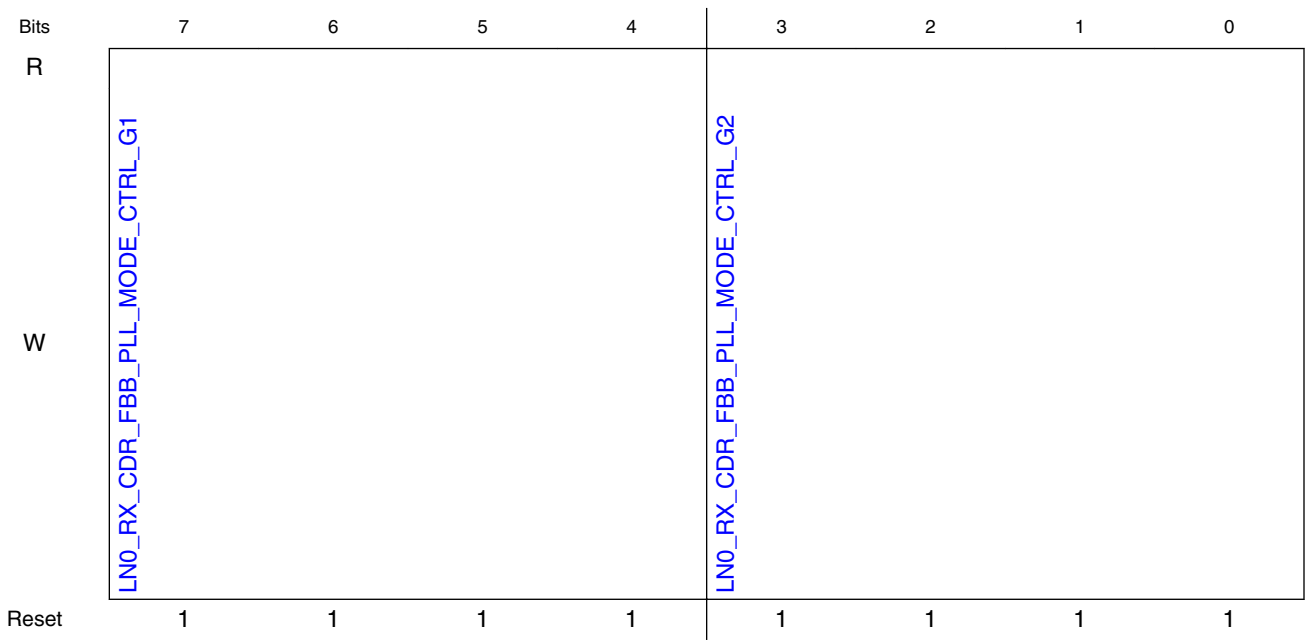
Field	Function
7-6 —	Reserved
5-0 LN0_RX_CDR_FBB_DELTA_CNT	Target delta number in VCO counter in CDR FBB cal mode Delta = fFBB(target) / fref * NRUN If Fvco=5GHz, fref=25MHz, target fFBB=9MHz, and NRUN=16, then delta = 9/25*17~=6

11.5.3.1.281 (TRSV_REG09A)

11.5.3.1.281.1 Offset

Register	Offset
TRSV_REG09A	668h

11.5.3.1.281.2 Diagram



11.5.3.1.281.3 Fields

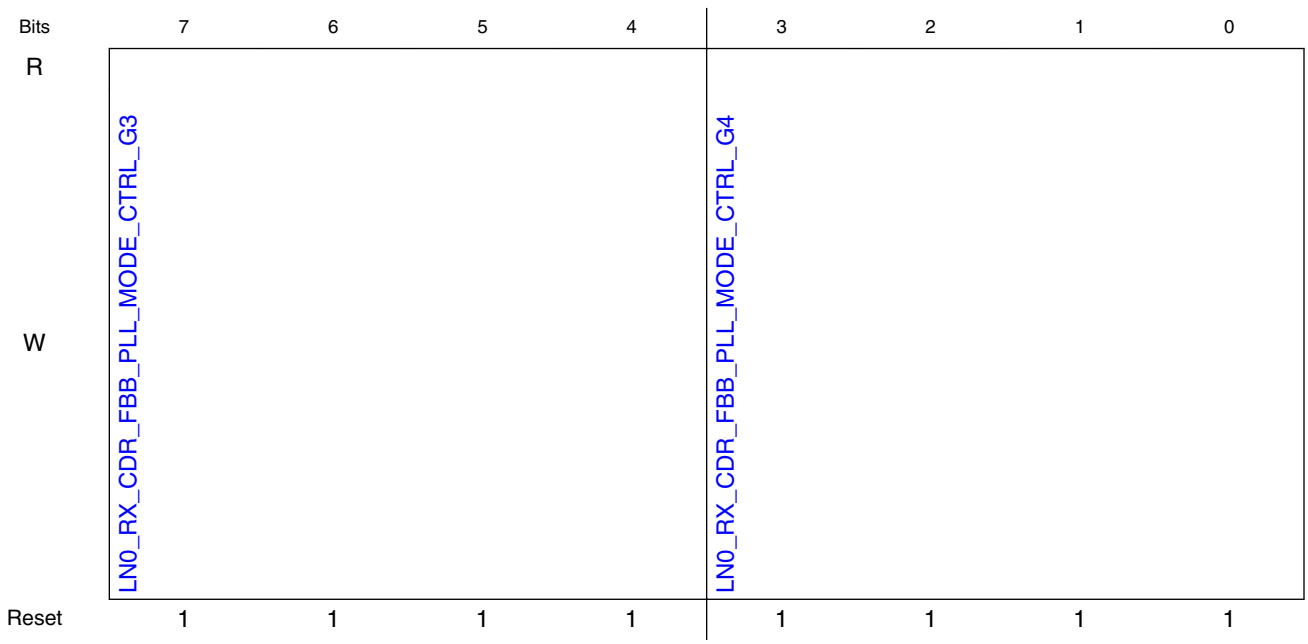
Field	Function
7-4 LN0_RX_CDR_FBB_PLL_MODE_CTRL_G1	[GEN1] RX CDR BBVCO FBB gain control in PLL mode 0000: Lowest gain ... 1111: Highest gain
3-0 LN0_RX_CDR_FBB_PLL_MODE_CTRL_G2	[GEN2]

11.5.3.1.282 (TRSV_REG09B)

11.5.3.1.282.1 Offset

Register	Offset
TRSV_REG09B	66Ch

11.5.3.1.282.2 Diagram



11.5.3.1.282.3 Fields

Field	Function
7-4 LN0_RX_CDR_FBB_PLL_MODE_CTRL_G3	[GEN3]
3-0 LN0_RX_CDR_FBB_PLL_MODE_CTRL_G4	[GEN4]

11.5.3.1.283 (TRSV_REG09C)

11.5.3.1.283.1 Offset

Register	Offset
TRSV_REG09C	670h

11.5.3.1.283.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_RX_CDR_FBB_COARSE_CTRL_G1				LN0_RX_CDR_FBB_COARSE_CTRL_G2			
W								
Reset	1	1	1	1	1	1	1	1

11.5.3.1.283.3 Fields

Field	Function
7-4 LN0_RX_CDR_FBB_COARSE_CTRL_G1	[GEN1] RX CDR BBVCO FBB gain control in coarse mode (high bandwidth) 0000: Lowest gain ... 1111: Highest gain
3-0 LN0_RX_CDR_FBB_COARSE_CTRL_G2	[GEN2]

11.5.3.1.284 (TRSV_REG09D)

11.5.3.1.284.1 Offset

Register	Offset
TRSV_REG09D	674h

11.5.3.1.284.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_RX_CDR_FBB_COARSE_CTRL_G3				LN0_RX_CDR_FBB_COARSE_CTRL_G4			
W								
Reset	1	1	0	0	1	1	0	0

11.5.3.1.284.3 Fields

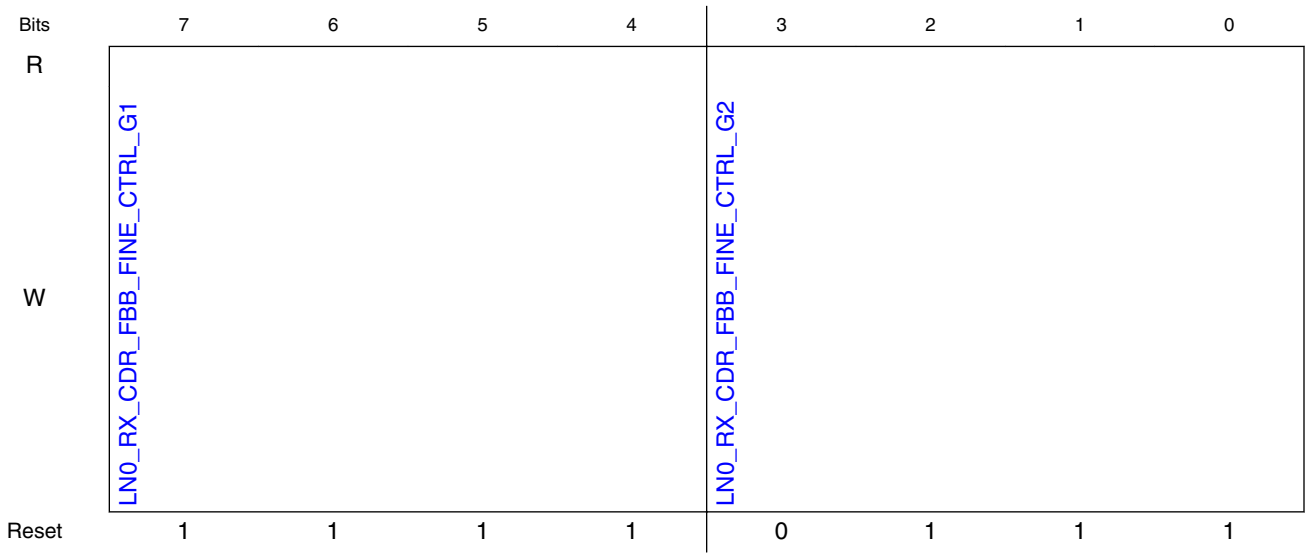
Field	Function
7-4 LN0_RX_CDR_FBB_COARSE_CTRL_G3	[GEN3]
3-0 LN0_RX_CDR_FBB_COARSE_CTRL_G4	[GEN4]

11.5.3.1.285 (TRSV_REG09E)

11.5.3.1.285.1 Offset

Register	Offset
TRSV_REG09E	678h

11.5.3.1.285.2 Diagram



11.5.3.1.285.3 Fields

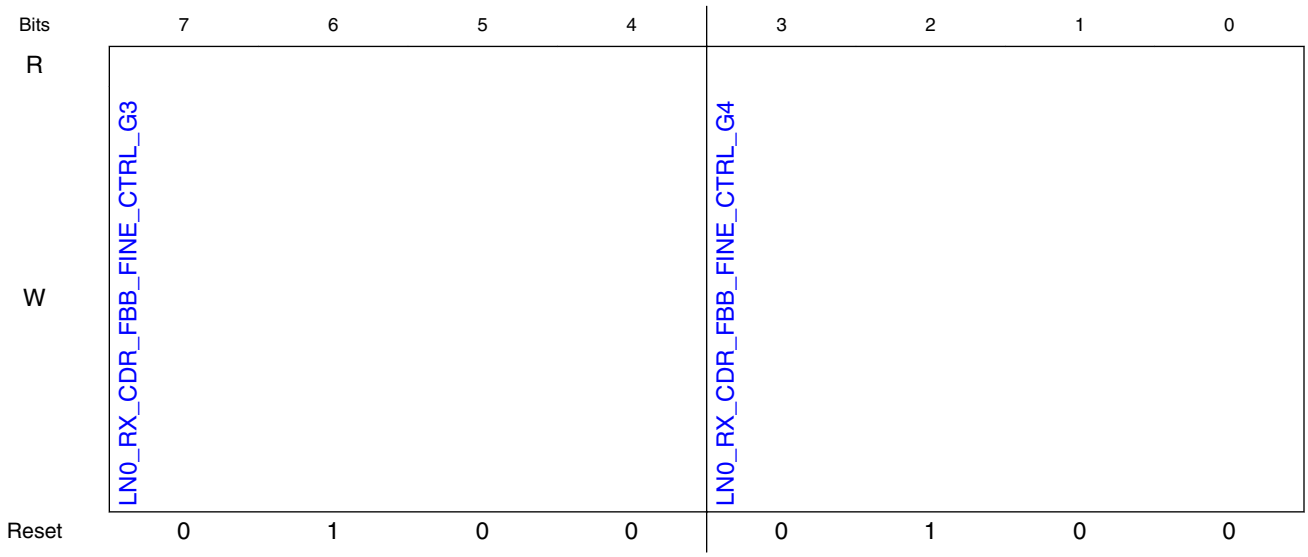
Field	Function
7-4 LN0_RX_CDR_FBB_FINE_CTRL_G1	[GEN1] RX CDR BBVCO FBB gain control in fine mode (low bandwidth) 0000: Lowest gain ... 1111: Highest gain
3-0 LN0_RX_CDR_FBB_FINE_CTRL_G2	[GEN2]

11.5.3.1.286 (TRSV_REG09F)

11.5.3.1.286.1 Offset

Register	Offset
TRSV_REG09F	67Ch

11.5.3.1.286.2 Diagram



11.5.3.1.286.3 Fields

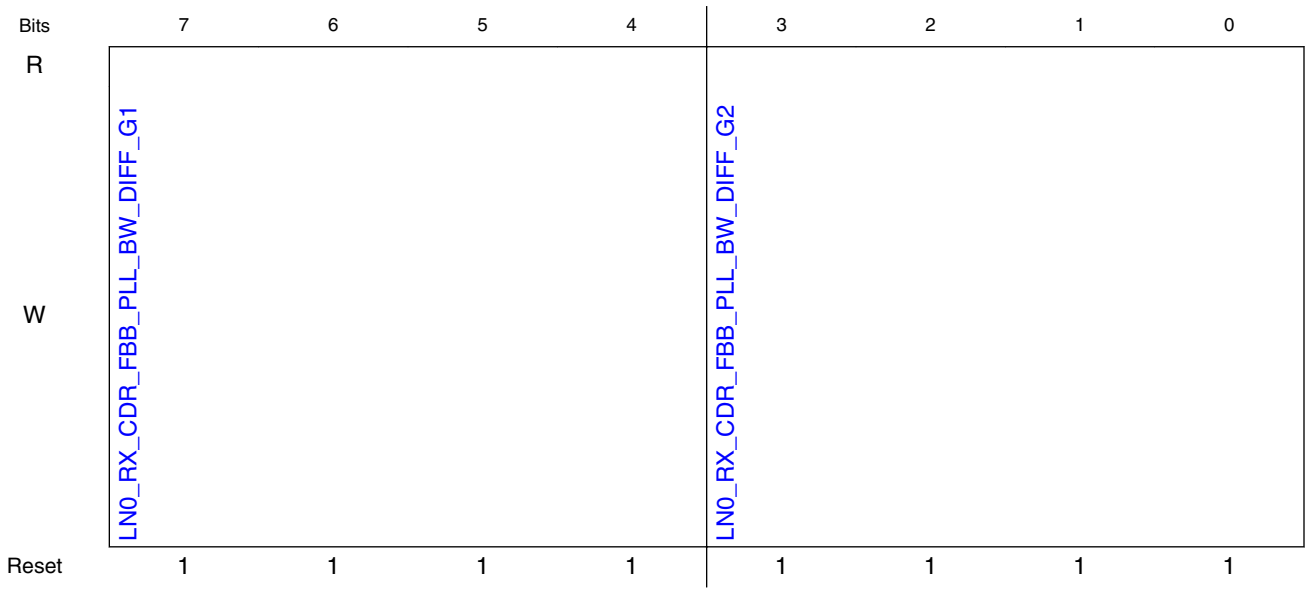
Field	Function
7-4 LN0_RX_CDR_FBB_FINE_CTRL_G3	[GEN3]
3-0 LN0_RX_CDR_FBB_FINE_CTRL_G4	[GEN4]

11.5.3.1.287 (TRSV_REG0A0)

11.5.3.1.287.1 Offset

Register	Offset
TRSV_REG0A0	680h

11.5.3.1.287.2 Diagram



11.5.3.1.287.3 Fields

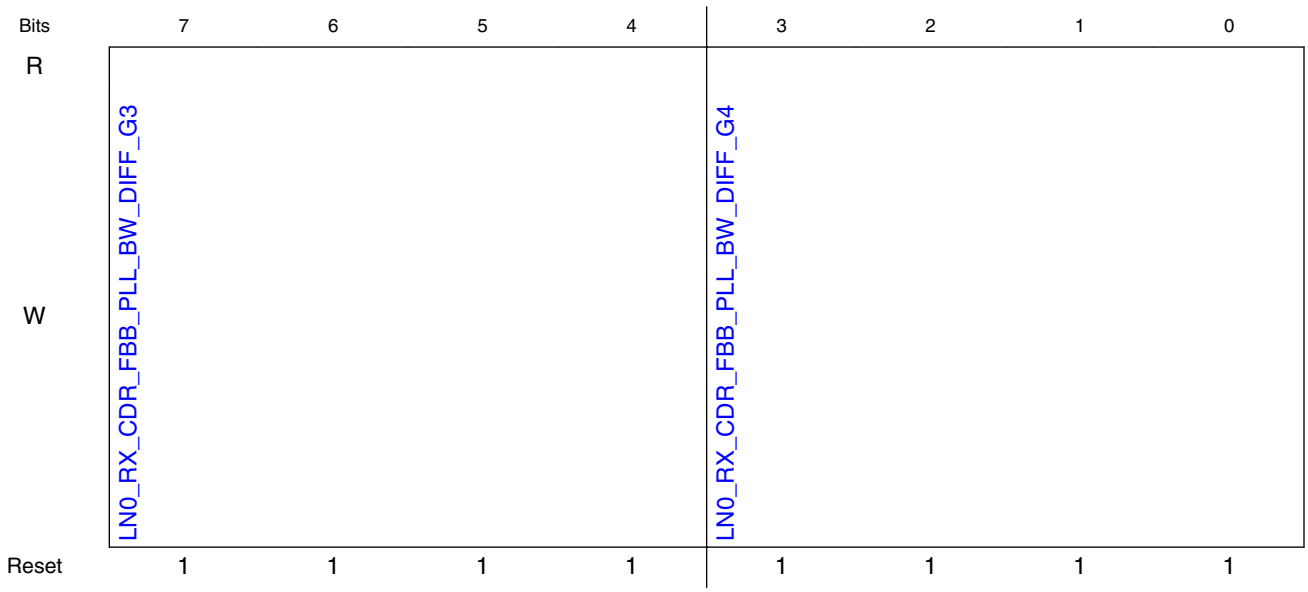
Field	Function
7-4 LN0_RX_CDR_FBB_PLL_BW_DIFF_G1	[GEN4]
3-0 LN0_RX_CDR_FBB_PLL_BW_DIFF_G2	[GEN4]

11.5.3.1.288 (TRSV_REG0A1)

11.5.3.1.288.1 Offset

Register	Offset
TRSV_REG0A1	684h

11.5.3.1.288.2 Diagram



11.5.3.1.288.3 Fields

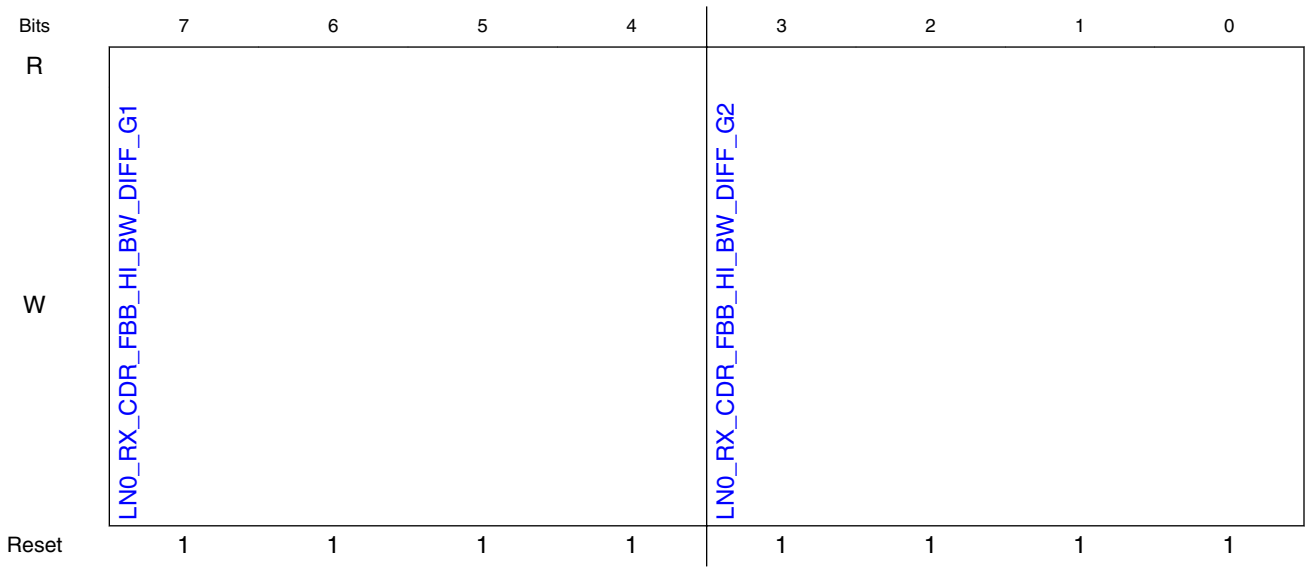
Field	Function
7-4 LN0_RX_CDR_FBB_PLL_BW_DIFF_G3	[GEN4]
3-0 LN0_RX_CDR_FBB_PLL_BW_DIFF_G4	[GEN4]

11.5.3.1.289 (TRSV_REG0A2)

11.5.3.1.289.1 Offset

Register	Offset
TRSV_REG0A2	688h

11.5.3.1.289.2 Diagram



11.5.3.1.289.3 Fields

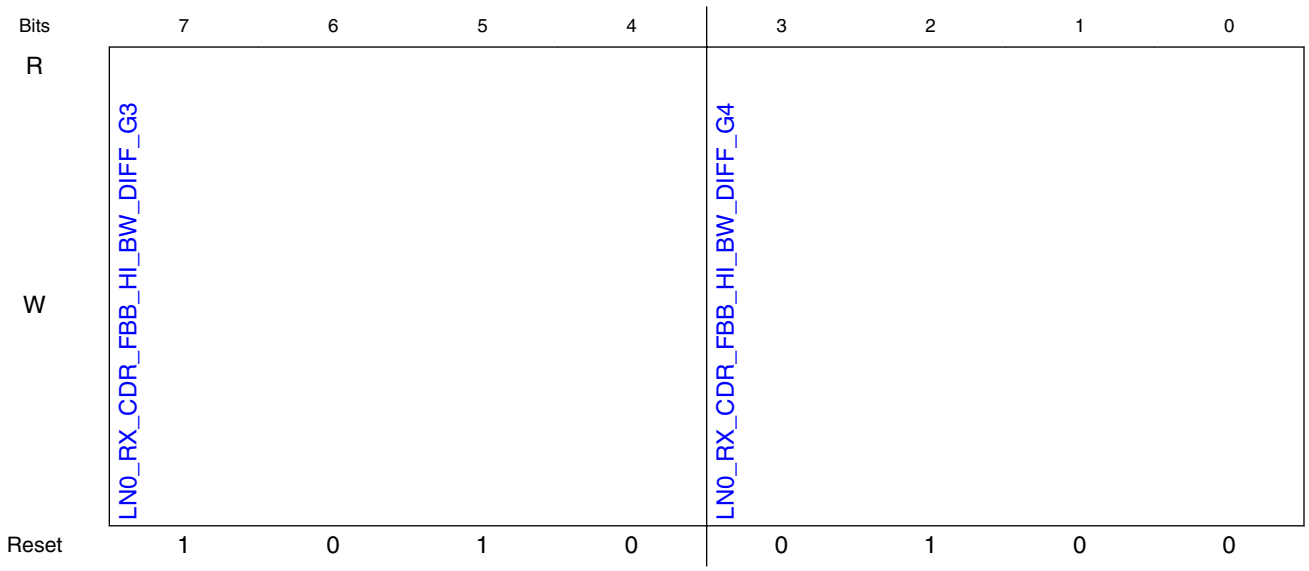
Field	Function
7-4 LN0_RX_CDR_FBB_HI_BW_DIFF_G1	[GEN4]
3-0 LN0_RX_CDR_FBB_HI_BW_DIFF_G2	[GEN4]

11.5.3.1.290 (TRSV_REG0A3)

11.5.3.1.290.1 Offset

Register	Offset
TRSV_REG0A3	68Ch

11.5.3.1.290.2 Diagram



11.5.3.1.290.3 Fields

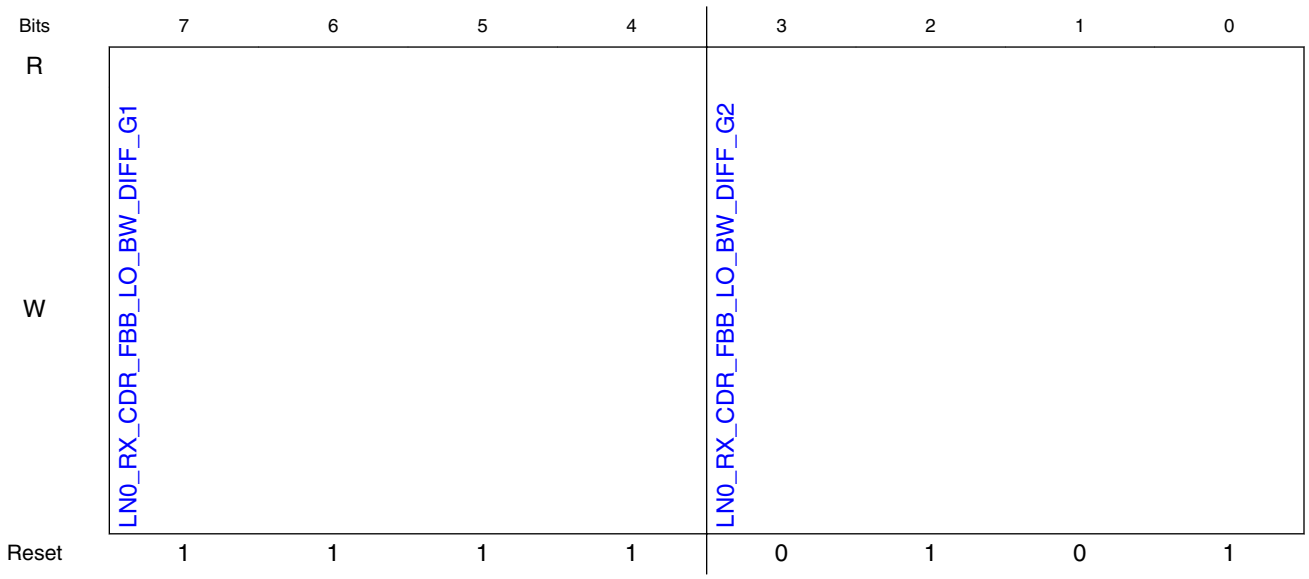
Field	Function
7-4 LN0_RX_CDR_FBB_HI_BW_DIFF_G3	[GEN4]
3-0 LN0_RX_CDR_FBB_HI_BW_DIFF_G4	[GEN4]

11.5.3.1.291 (TRSV_REG0A4)

11.5.3.1.291.1 Offset

Register	Offset
TRSV_REG0A4	690h

11.5.3.1.291.2 Diagram



11.5.3.1.291.3 Fields

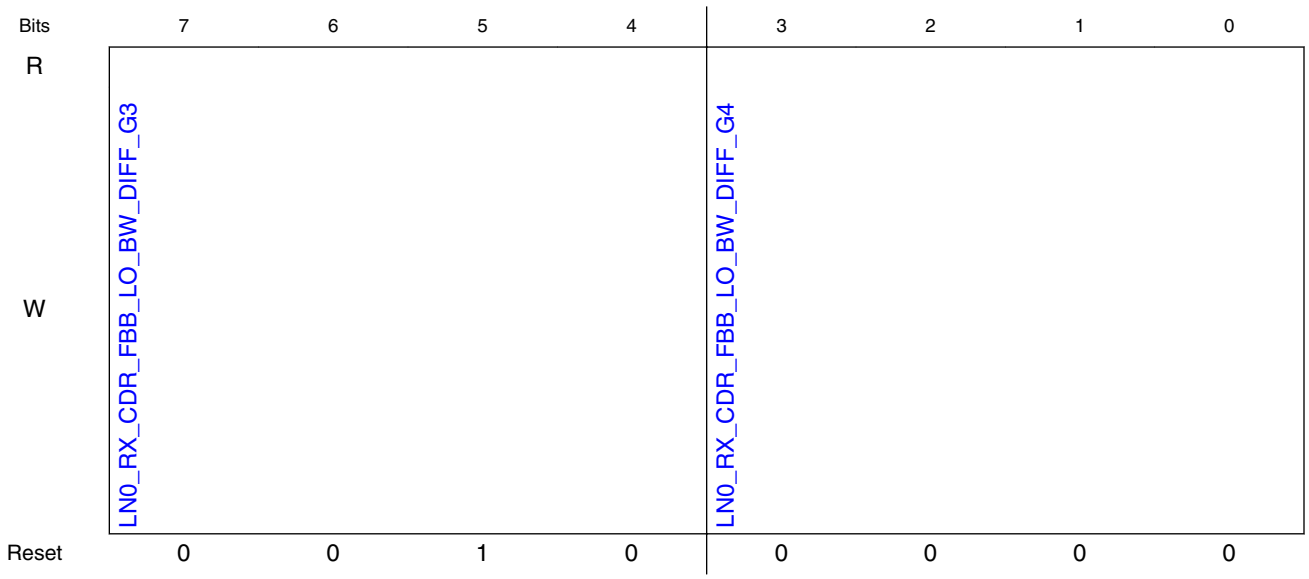
Field	Function
7-4 LN0_RX_CDR_FBB_LO_BW_DIFF_G1	[GEN4]
3-0 LN0_RX_CDR_FBB_LO_BW_DIFF_G2	[GEN4]

11.5.3.1.292 (TRSV_REG0A5)

11.5.3.1.292.1 Offset

Register	Offset
TRSV_REG0A5	694h

11.5.3.1.292.2 Diagram



11.5.3.1.292.3 Fields

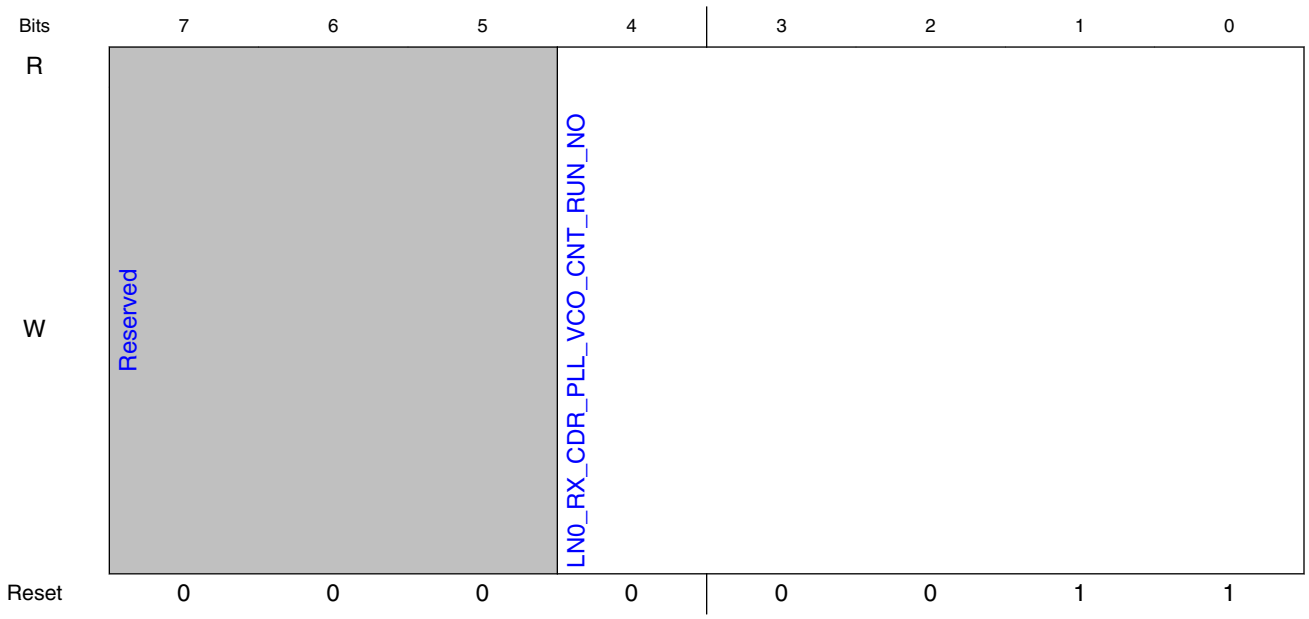
Field	Function
7-4 LN0_RX_CDR_FBB_LO_BW_DIFF_G3	[GEN4]
3-0 LN0_RX_CDR_FBB_LO_BW_DIFF_G4	[GEN4]

11.5.3.1.293 (TRSV_REG0A6)

11.5.3.1.293.1 Offset

Register	Offset
TRSV_REG0A6	698h

11.5.3.1.293.2 Diagram



11.5.3.1.293.3 Fields

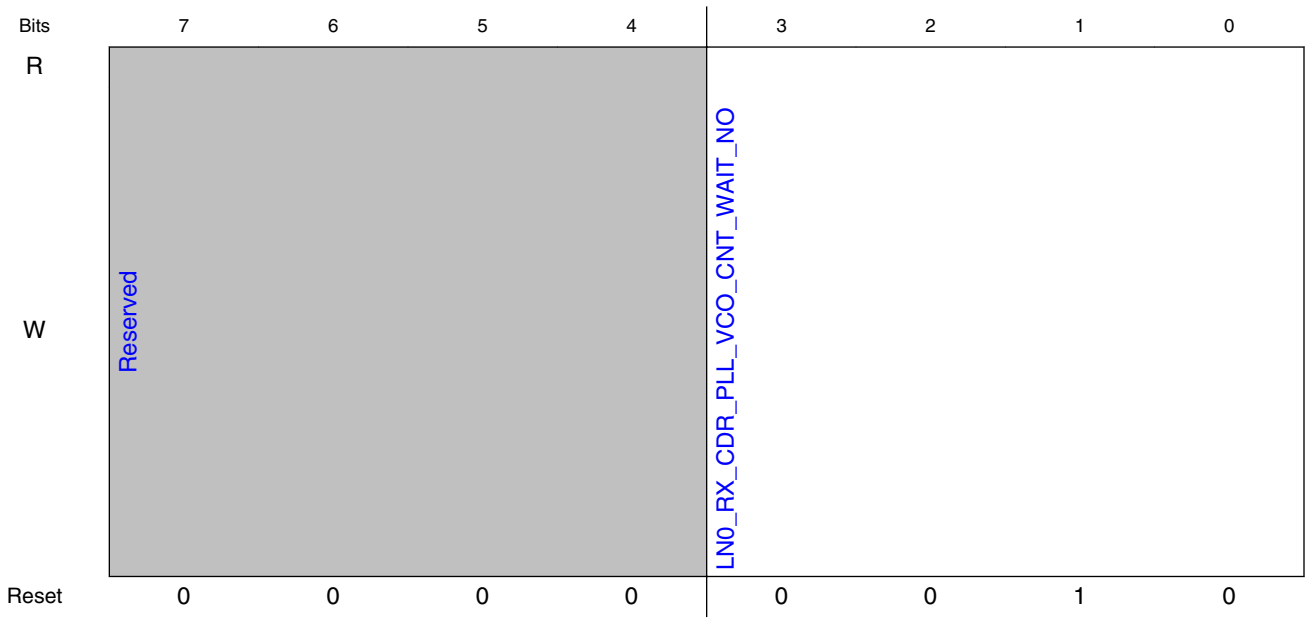
Field	Function
7-5 —	Reserved
4-0 LN0_RX_CDR_PLL_VCO_CNT_RUN_NO	

11.5.3.1.294 (TRSV_REG0A7)

11.5.3.1.294.1 Offset

Register	Offset
TRSV_REG0A7	69Ch

11.5.3.1.294.2 Diagram



11.5.3.1.294.3 Fields

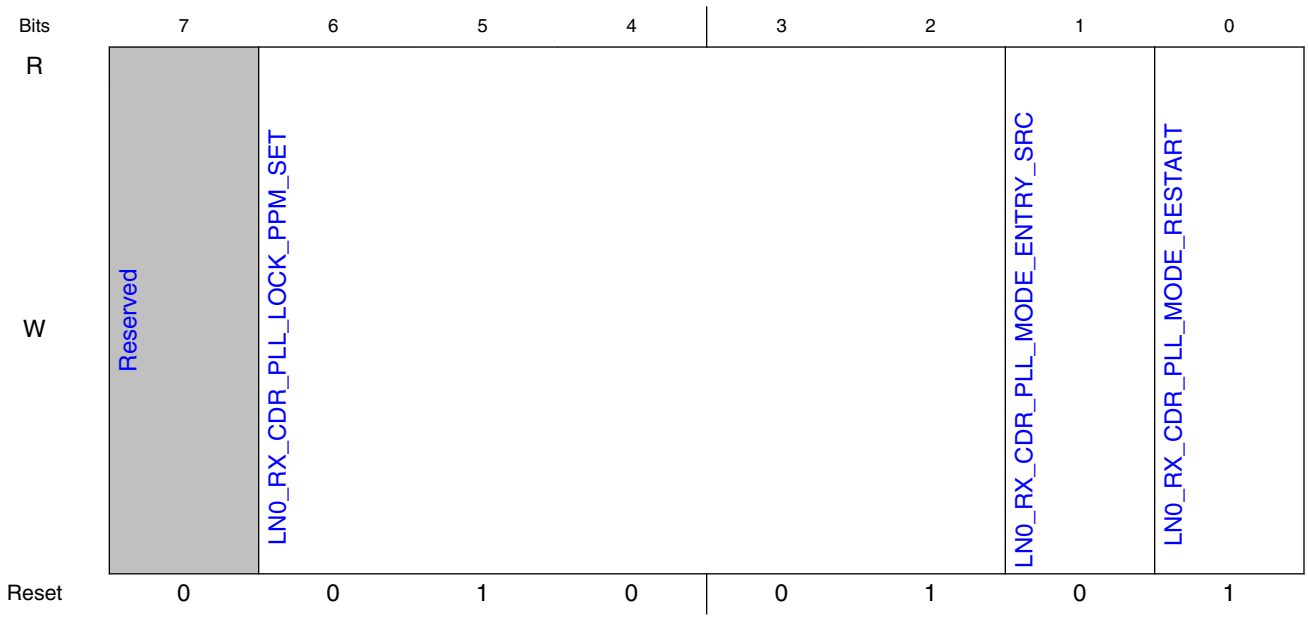
Field	Function
7-4 —	Reserved
3-0 LN0_RX_CDR_PLL_VCO_CNT_WAIT_NO	

11.5.3.1.295 (TRSV_REG0A8)

11.5.3.1.295.1 Offset

Register	Offset
TRSV_REG0A8	6A0h

11.5.3.1.295.2 Diagram



11.5.3.1.295.3 Fields

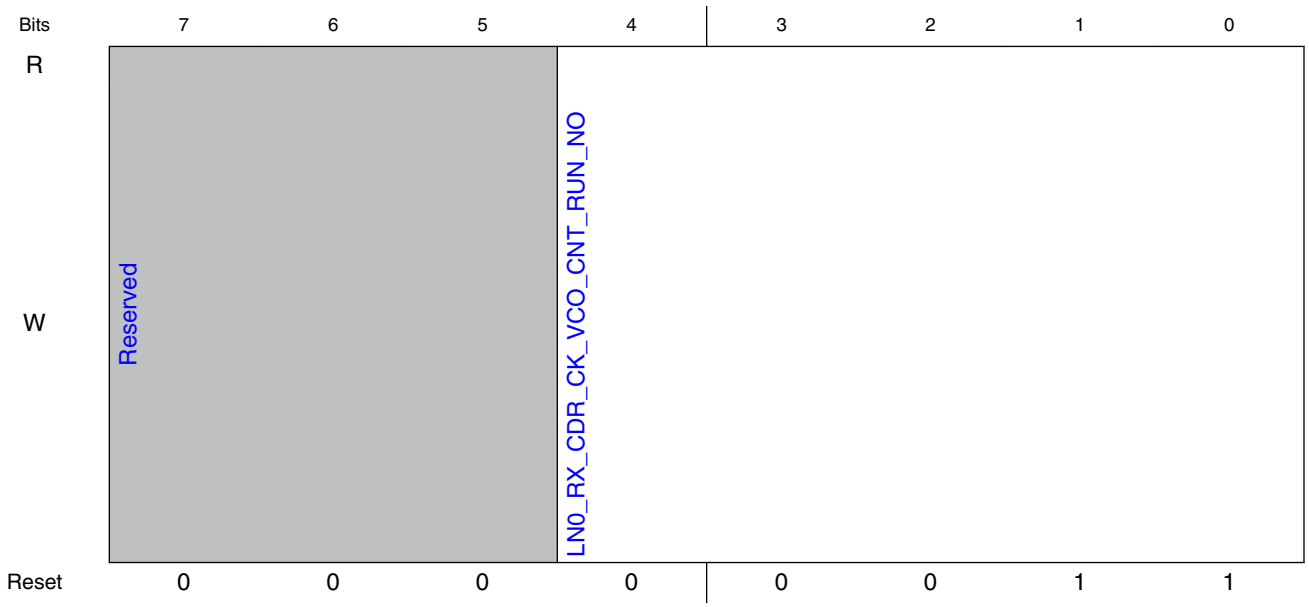
Field	Function
7 —	Reserved
6-2 LN0_RX_CDR_PLL_LOCK_PPM_SET	
1 LN0_RX_CDR_PLL_MODE_ENTRY_SRC	
0 LN0_RX_CDR_PLL_MODE_RESTART	

11.5.3.1.296 (TRSV_REG0A9)

11.5.3.1.296.1 Offset

Register	Offset
TRSV_REG0A9	6A4h

11.5.3.1.296.2 Diagram



11.5.3.1.296.3 Fields

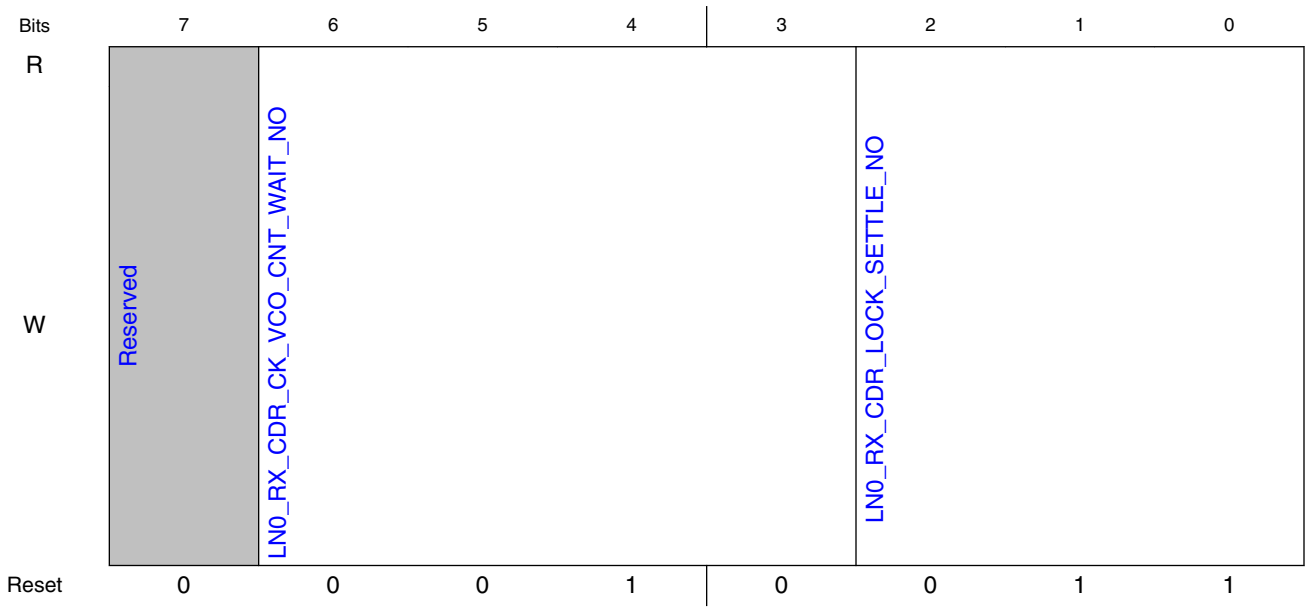
Field	Function
7-5 —	Reserved
4-0 LN0_RX_CDR_CK_VCO_CNT_RUN_NO	

11.5.3.1.297 (TRSV_REG0AA)

11.5.3.1.297.1 Offset

Register	Offset
TRSV_REG0AA	6A8h

11.5.3.1.297.2 Diagram



11.5.3.1.297.3 Fields

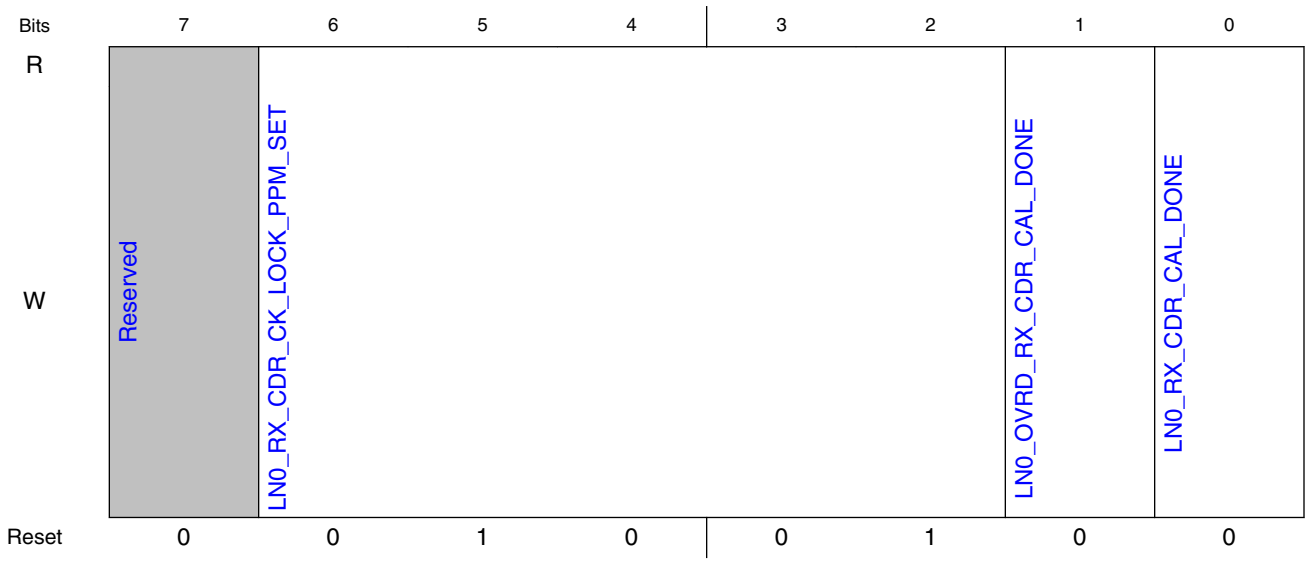
Field	Function
7 —	Reserved
6-3 LN0_RX_CDR_CK_VCO_CNT_WAIT_NO	
2-0 LN0_RX_CDR_LOCK_SETTLE_NO	

11.5.3.1.298 (TRSV_REG0AB)

11.5.3.1.298.1 Offset

Register	Offset
TRSV_REG0AB	6ACh

11.5.3.1.298.2 Diagram



11.5.3.1.298.3 Fields

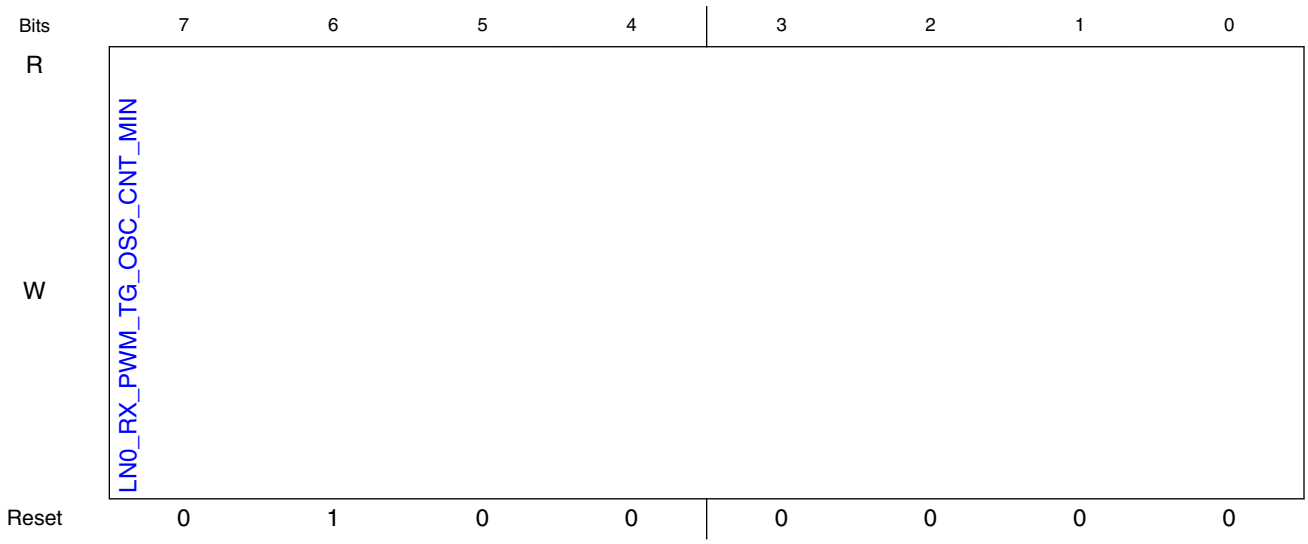
Field	Function
7 —	Reserved
6-2 LN0_RX_CDR_CK_LOCK_PPM_SET	
1 LN0_OVRD_RX_CDR_CAL_DONE	Override enable for rx_cdr_cal_done
0 LN0_RX_CDR_CAL_DONE	

11.5.3.1.299 (TRSV_REG0AC)

11.5.3.1.299.1 Offset

Register	Offset
TRSV_REG0AC	6B0h

11.5.3.1.299.2 Diagram



11.5.3.1.299.3 Fields

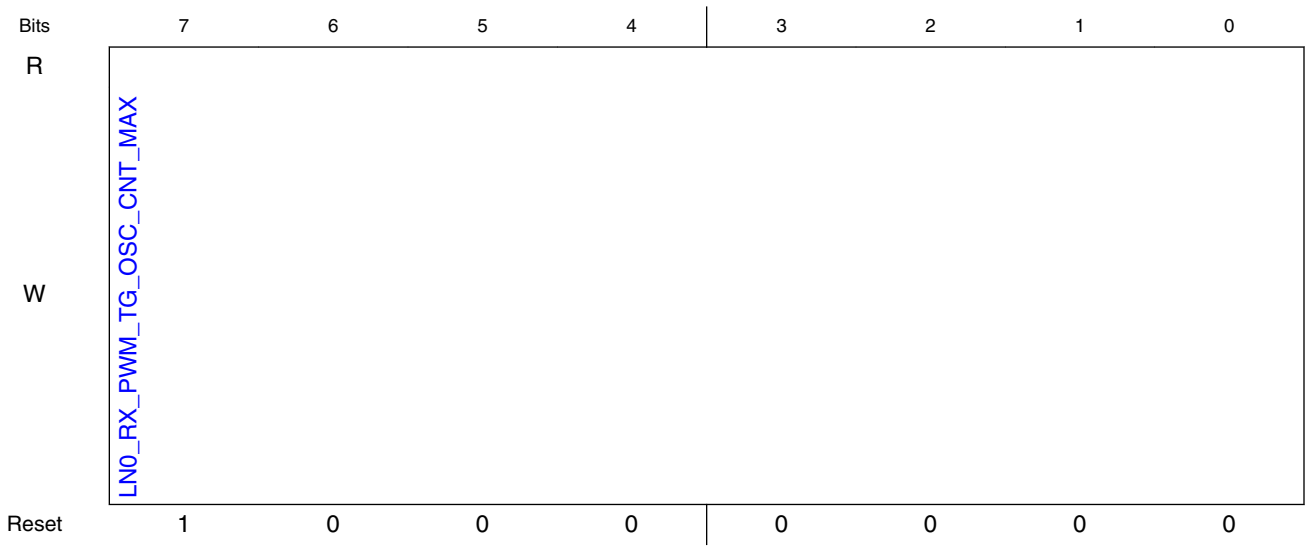
Field	Function
7-0 LN0_RX_PWM_TG_OSC_CNT_MIN	

11.5.3.1.300 (TRSV_REG0AD)

11.5.3.1.300.1 Offset

Register	Offset
TRSV_REG0AD	6B4h

11.5.3.1.300.2 Diagram



11.5.3.1.300.3 Fields

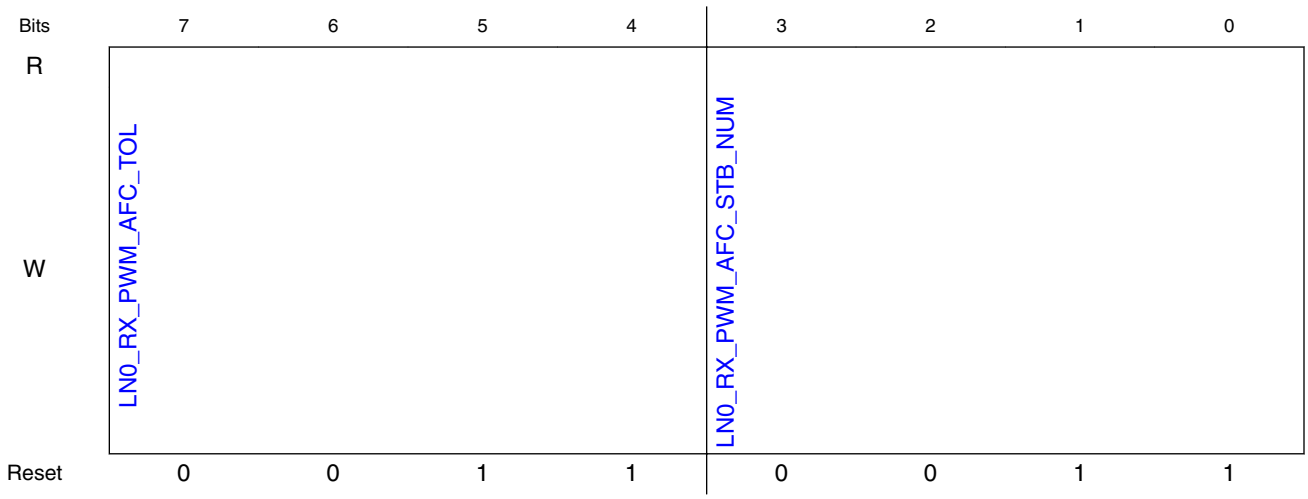
Field	Function
7-0 LN0_RX_PWM_TG_OSC_CNT_MAX	

11.5.3.1.301 (TRSV_REG0AE)

11.5.3.1.301.1 Offset

Register	Offset
TRSV_REG0AE	6B8h

11.5.3.1.301.2 Diagram



11.5.3.1.301.3 Fields

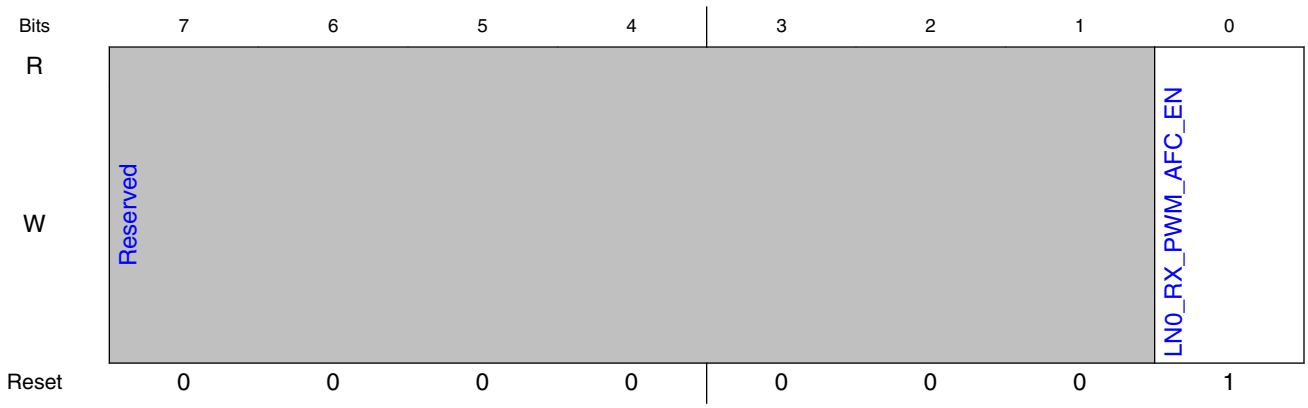
Field	Function
7-4 LN0_RX_PWM_AFC_TOL	
3-0 LN0_RX_PWM_AFC_STB_NUM	

11.5.3.1.302 (TRSV_REG0AF)

11.5.3.1.302.1 Offset

Register	Offset
TRSV_REG0AF	6BCh

11.5.3.1.302.2 Diagram



11.5.3.1.302.3 Fields

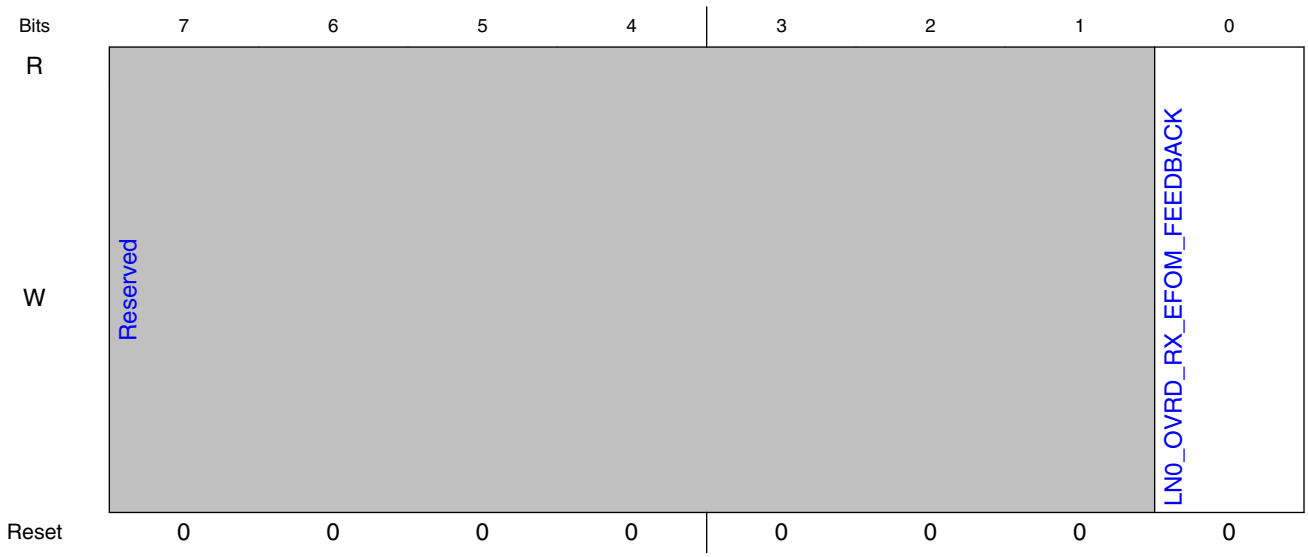
Field	Function
7-1 —	Reserved
0 LN0_RX_PWM_AFC_EN	

11.5.3.1.303 (TRSV_REG0B0)

11.5.3.1.303.1 Offset

Register	Offset
TRSV_REG0B0	6C0h

11.5.3.1.303.2 Diagram



11.5.3.1.303.3 Fields

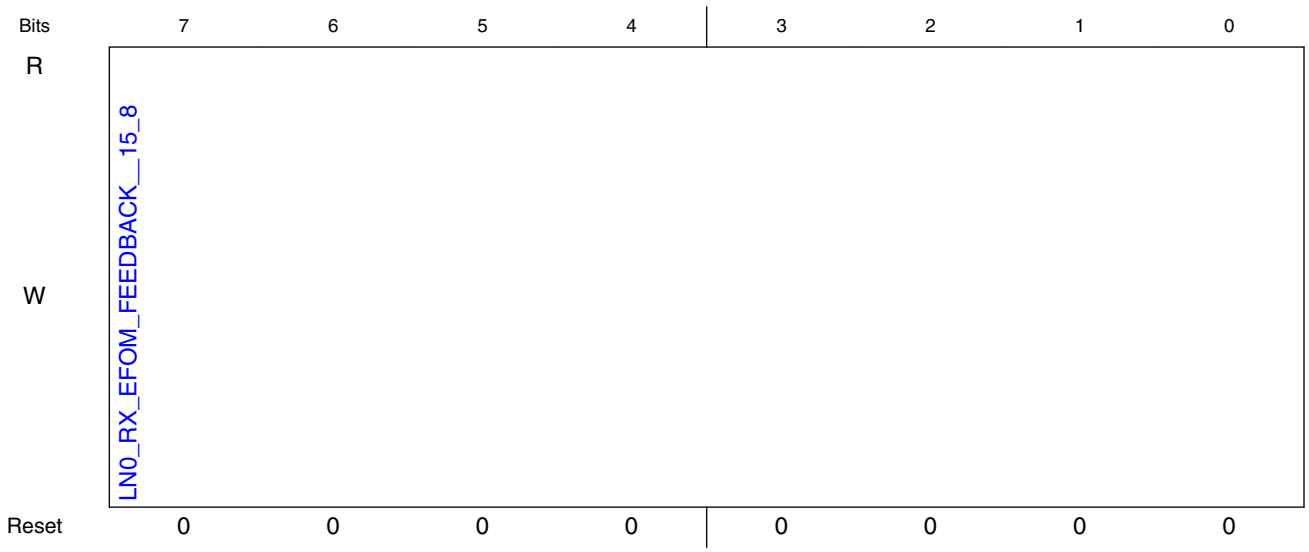
Field	Function
7-1 —	Reserved
0 LN0_OVRD_RX_EFOM_FEEDBACK	Override enable for

11.5.3.1.304 (TRSV_REG0B1)

11.5.3.1.304.1 Offset

Register	Offset
TRSV_REG0B1	6C4h

11.5.3.1.304.2 Diagram



11.5.3.1.304.3 Fields

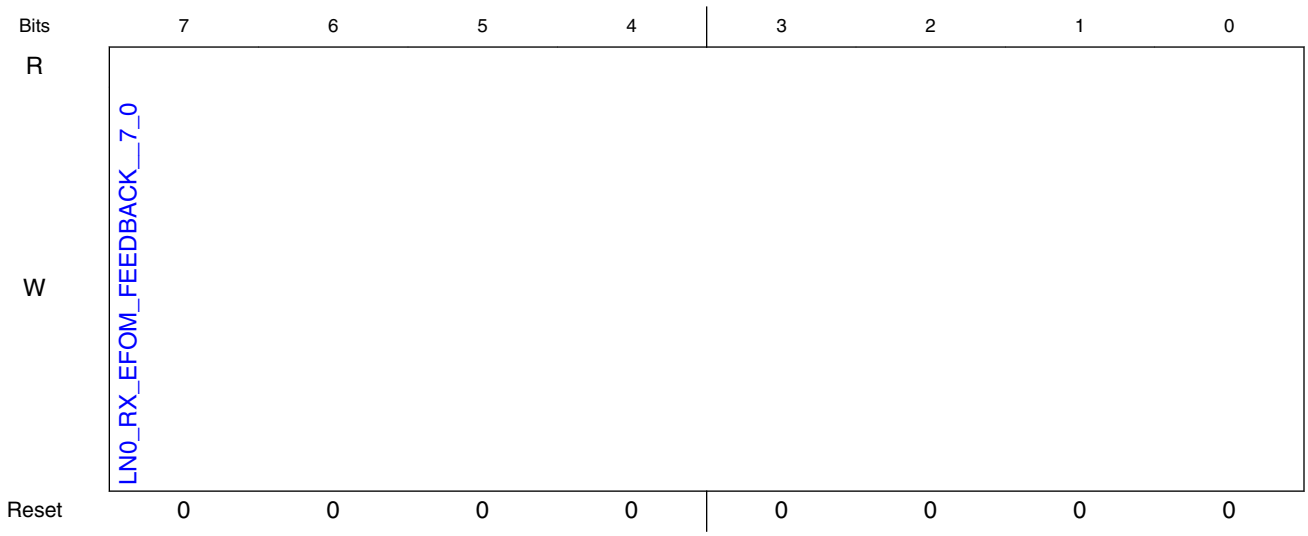
Field	Function
7-0 LN0_RX_EFOM_FEEDBACK__15_8	

11.5.3.1.305 (TRSV_REG0B2)

11.5.3.1.305.1 Offset

Register	Offset
TRSV_REG0B2	6C8h

11.5.3.1.305.2 Diagram



11.5.3.1.305.3 Fields

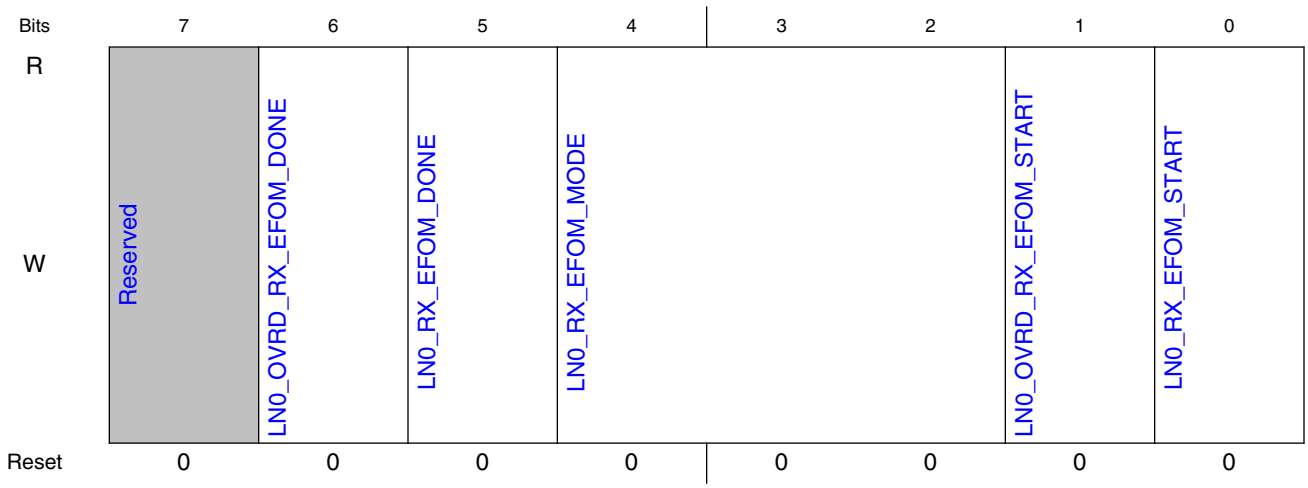
Field	Function
7-0 LN0_RX_EFOM_FEEDBACK__7_0	

11.5.3.1.306 (TRSV_REG0B3)

11.5.3.1.306.1 Offset

Register	Offset
TRSV_REG0B3	6CCh

11.5.3.1.306.2 Diagram



11.5.3.1.306.3 Fields

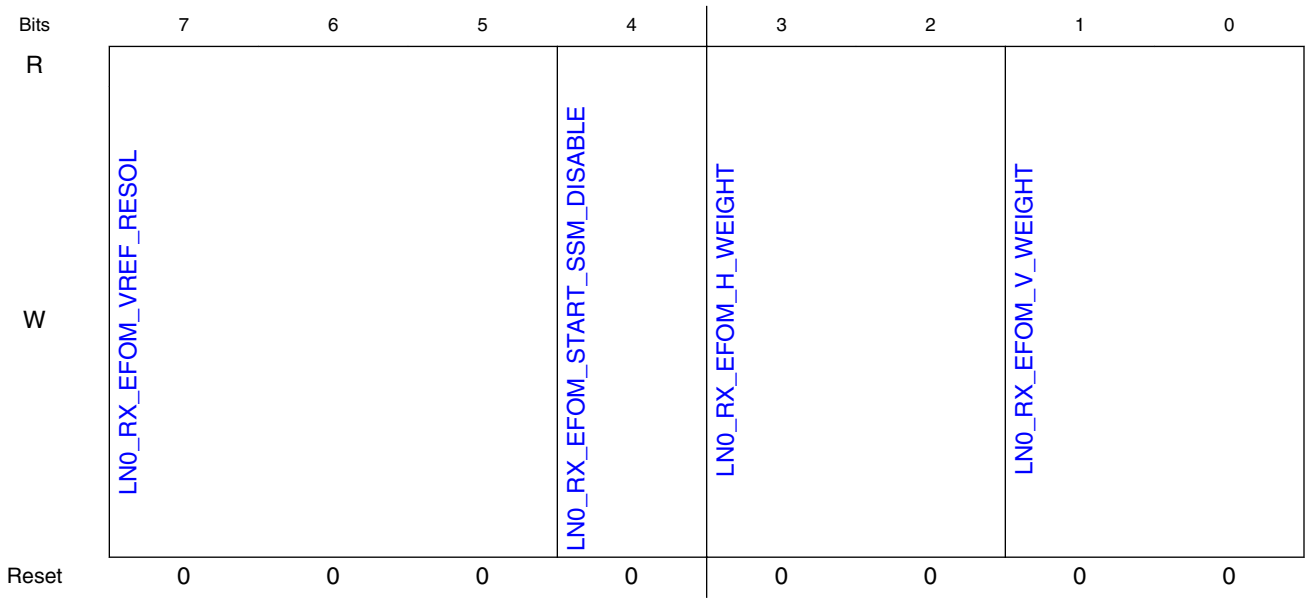
Field	Function
7 —	Reserved
6 LN0_OVRD_RX_EFOM_DONE	Override enable for rx_efom_done
5 LN0_RX_EFOM_DONE	
4-2 LN0_RX_EFOM_MODE	
1 LN0_OVRD_RX_EFOM_START	Override enable for rx_efom_start
0 LN0_RX_EFOM_START	

11.5.3.1.307 (TRSV_REG0B4)

11.5.3.1.307.1 Offset

Register	Offset
TRSV_REG0B4	6D0h

11.5.3.1.307.2 Diagram



11.5.3.1.307.3 Fields

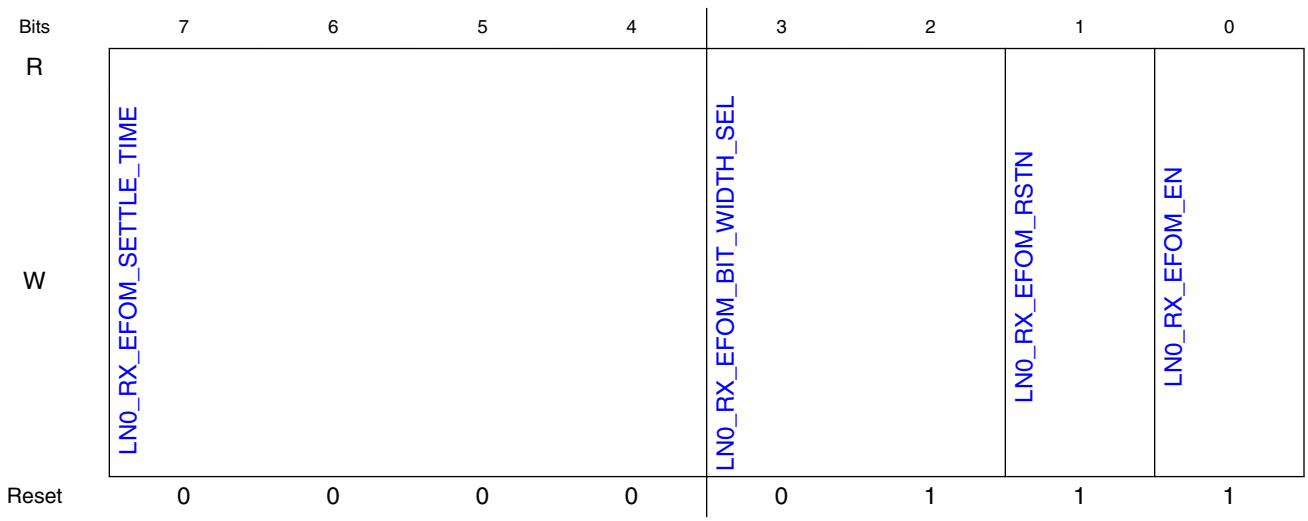
Field	Function
7-5 LN0_RX_EFOM_VREF_RESOL	
4 LN0_RX_EFOM_START_SSM_DISABLE	
3-2 LN0_RX_EFOM_H_WEIGHT	
1-0 LN0_RX_EFOM_V_WEIGHT	

11.5.3.1.308 (TRSV_REG0B5)

11.5.3.1.308.1 Offset

Register	Offset
TRSV_REG0B5	6D4h

11.5.3.1.308.2 Diagram



11.5.3.1.308.3 Fields

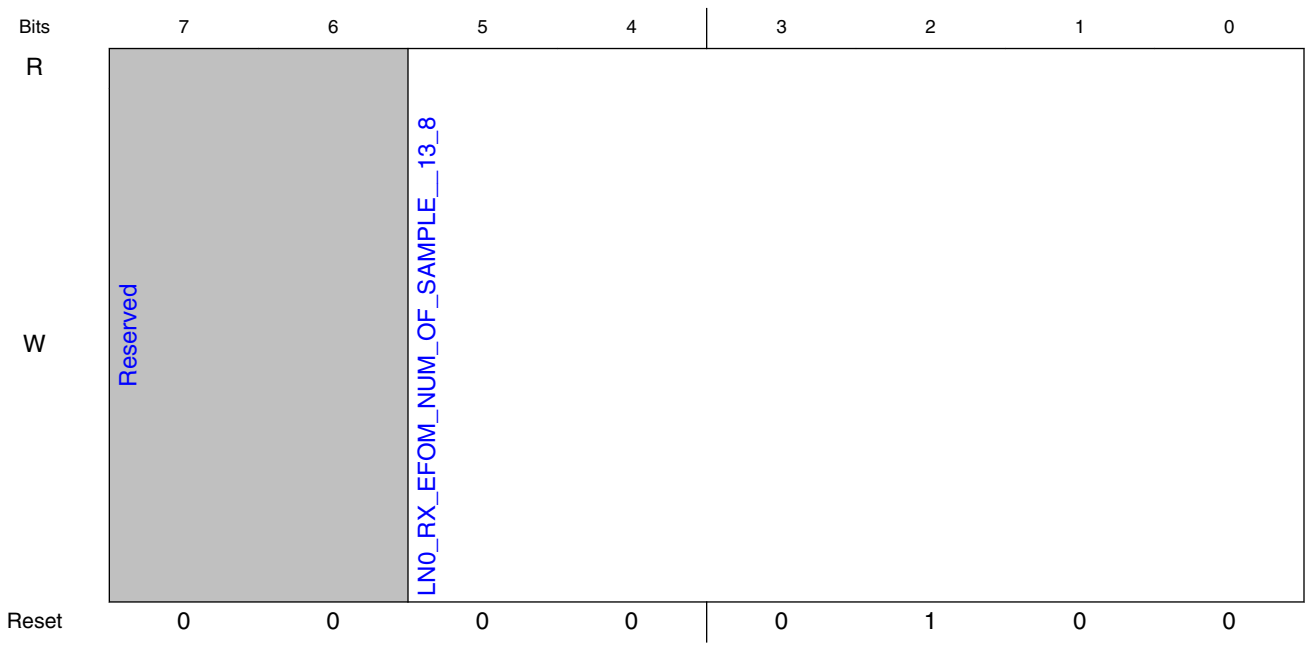
Field	Function
7-4 LN0_RX_EFOM_SETTLE_TIME	
3-2 LN0_RX_EFOM_BIT_WIDTH_SEL	
1 LN0_RX_EFOM_RSTN	
0 LN0_RX_EFOM_EN	

11.5.3.1.309 (TRSV_REG0B6)

11.5.3.1.309.1 Offset

Register	Offset
TRSV_REG0B6	6D8h

11.5.3.1.309.2 Diagram



11.5.3.1.309.3 Fields

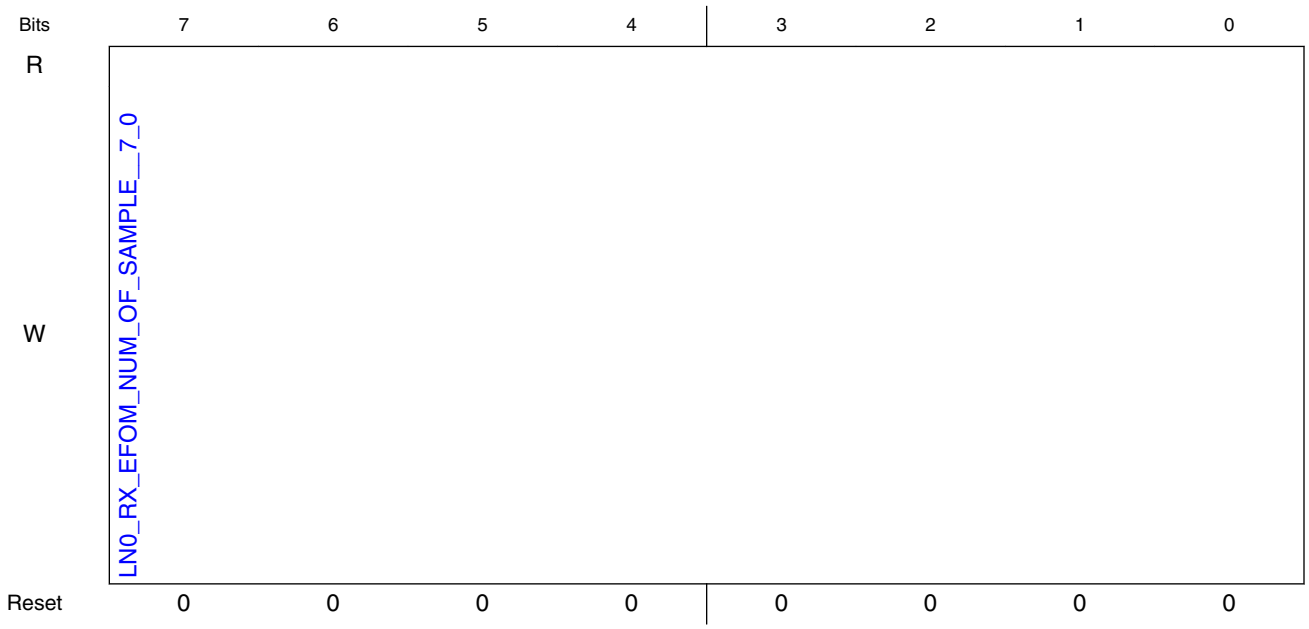
Field	Function
7-6 —	Reserved
5-0 LN0_RX_EFOM_NUM_OF_SAMPLE__13_8	

11.5.3.1.310 (TRSV_REG0B7)

11.5.3.1.310.1 Offset

Register	Offset
TRSV_REG0B7	6DCh

11.5.3.1.310.2 Diagram



11.5.3.1.310.3 Fields

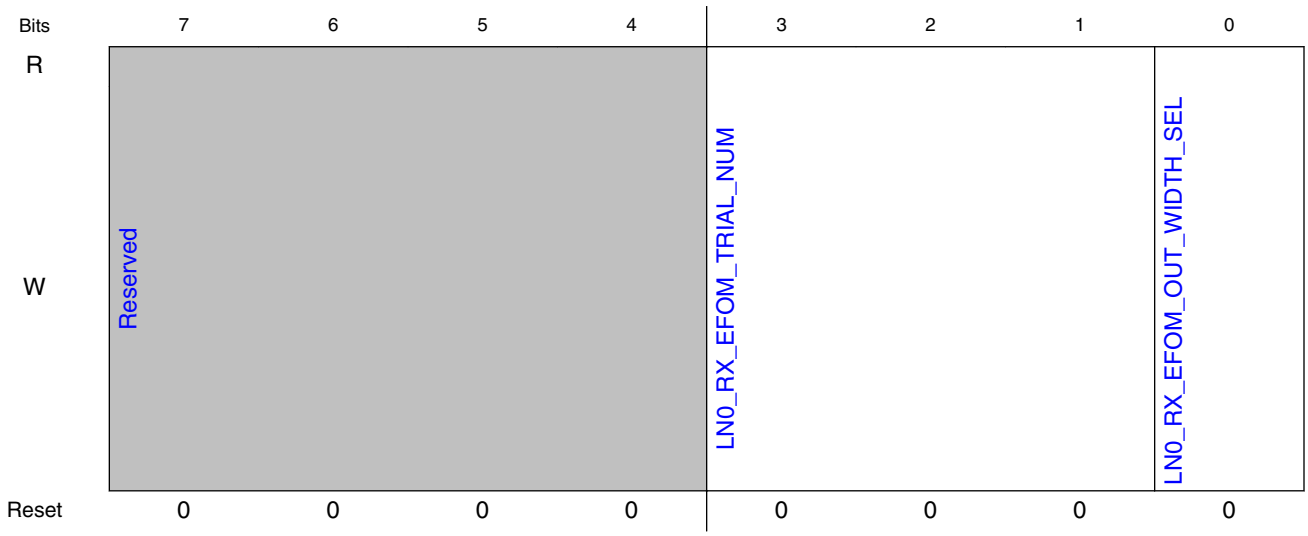
Field	Function
7-0 LN0_RX_EFOM_NUM_OF_SAMPLE__7_0	

11.5.3.1.311 (TRSV_REG0B8)

11.5.3.1.311.1 Offset

Register	Offset
TRSV_REG0B8	6E0h

11.5.3.1.311.2 Diagram



11.5.3.1.311.3 Fields

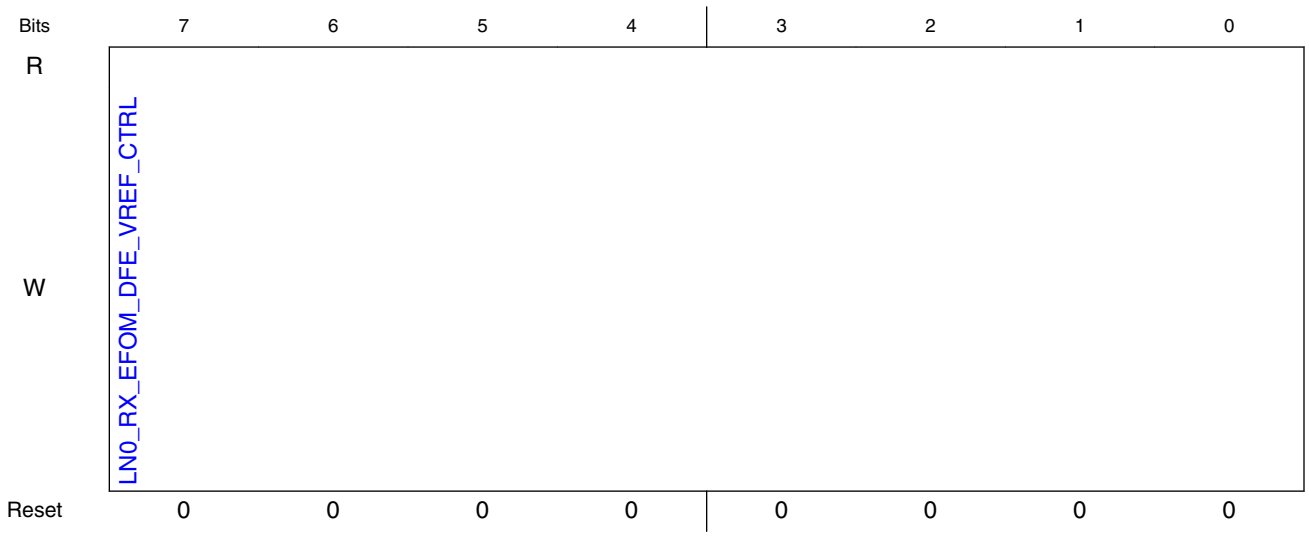
Field	Function
7-4 —	Reserved
3-1 LN0_RX_EFOM_TRIAL_NUM	
0 LN0_RX_EFOM_OUT_WIDTH_SEL	

11.5.3.1.312 (TRSV_REG0B9)

11.5.3.1.312.1 Offset

Register	Offset
TRSV_REG0B9	6E4h

11.5.3.1.312.2 Diagram



11.5.3.1.312.3 Fields

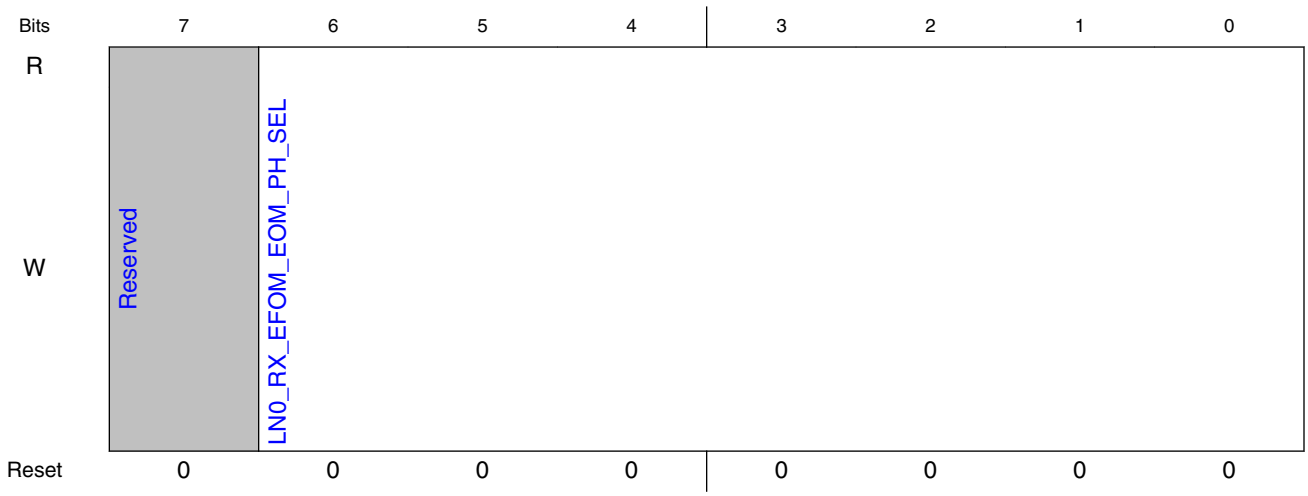
Field	Function
7-0 LN0_RX_EFOM_DFE_VREF_CTRL	

11.5.3.1.313 (TRSV_REG0BA)

11.5.3.1.313.1 Offset

Register	Offset
TRSV_REG0BA	6E8h

11.5.3.1.313.2 Diagram



11.5.3.1.313.3 Fields

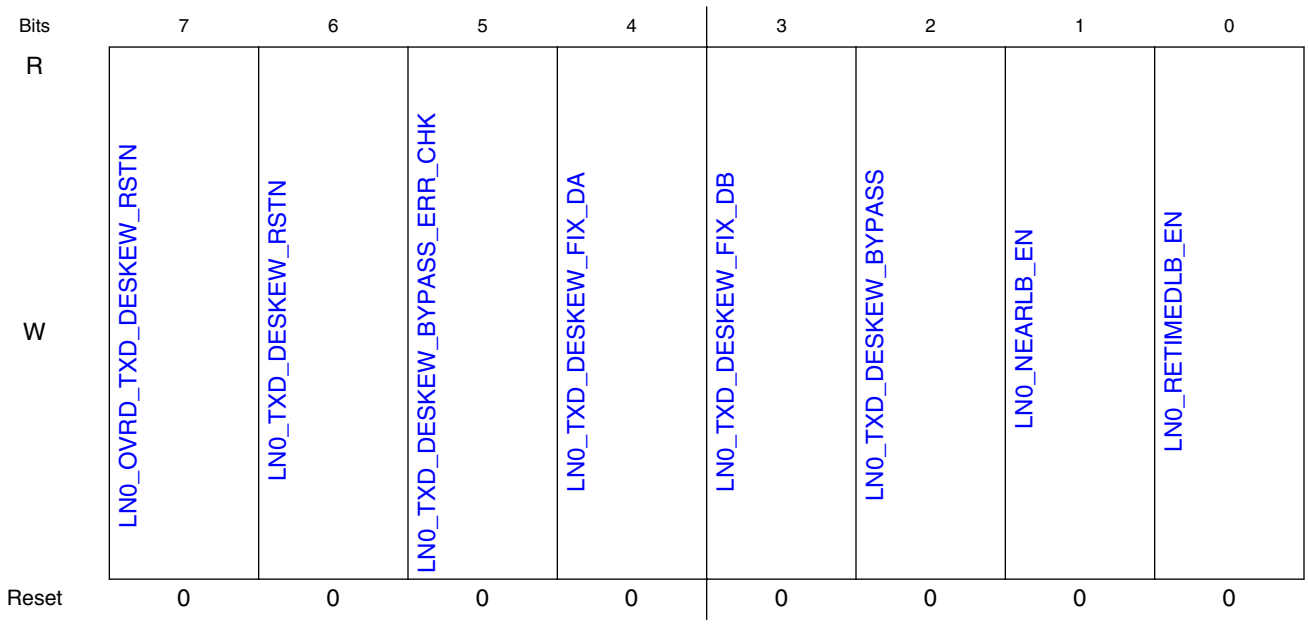
Field	Function
7 —	Reserved
6-0 LN0_RX_EFOM_EOM_PH_SEL	

11.5.3.1.314 (TRSV_REG0BB)

11.5.3.1.314.1 Offset

Register	Offset
TRSV_REG0BB	6ECh

11.5.3.1.314.2 Diagram



11.5.3.1.314.3 Fields

Field	Function
7 LN0_OVRD_TXD_DESKEW_RSTN	Override enable for txd_deskew_rstn
6 LN0_TXD_DESKEW_RSTN	
5 LN0_TXD_DESKEW_BYPASS_ERR_CHK	
4 LN0_TXD_DESKEW_FIX_DA	
3 LN0_TXD_DESKEW_FIX_DB	
2 LN0_TXD_DESKEW_BYPASS	
1	

Table continues on the next page...

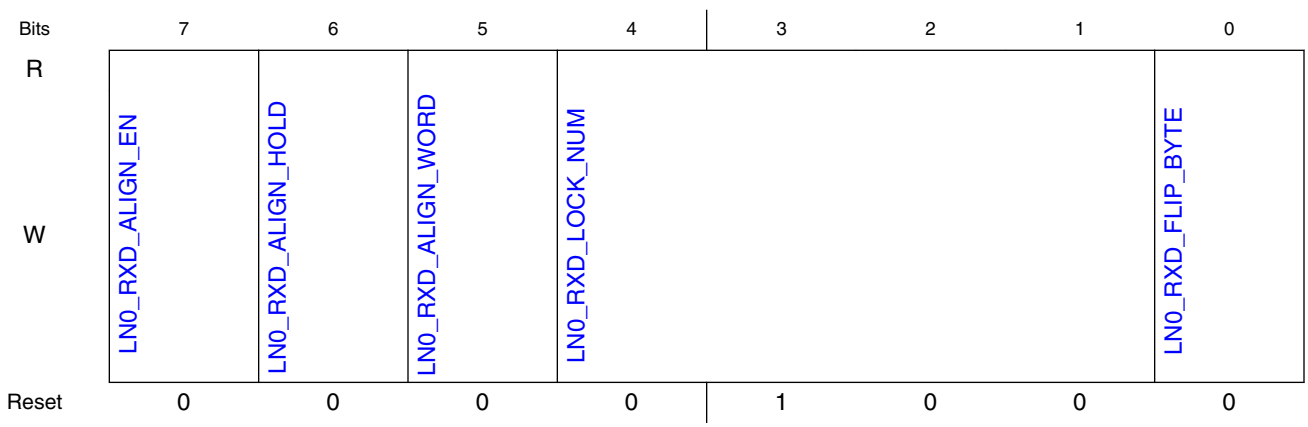
Field	Function
LN0_NEARLB_EN	
0 LN0_RETIMEDLB_EN	

11.5.3.1.315 (TRSV_REG0BC)

11.5.3.1.315.1 Offset

Register	Offset
TRSV_REG0BC	6F0h

11.5.3.1.315.2 Diagram



11.5.3.1.315.3 Fields

Field	Function
7 LN0_RXD_ALIGN_EN	
6 LN0_RXD_ALIGN_HOLD	
5	

Table continues on the next page...

Field	Function
LN0_RXD_ALIGN_WORD	
4-1 LN0_RXD_LOCK_NUM	
0 LN0_RXD_FLIP_BYTE	

11.5.3.1.316 (TRSV_REG0BD)

11.5.3.1.316.1 Offset

Register	Offset
TRSV_REG0BD	6F4h

11.5.3.1.316.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_RXD_FLIP_BIT	LN0_RXD_POLARITY	LN0_OVRD_TX_RCAL_RSTN	LN0_TX_RCAL_RSTN	LN0_TX_RCAL_UP_OPT_CODE		LN0_TX_RCAL_DN_OPT_CODE	
W								
Reset	0	0	0	0	0	0	0	0

11.5.3.1.316.3 Fields

Field	Function
7 LN0_RXD_FLIP_BIT	

Table continues on the next page...

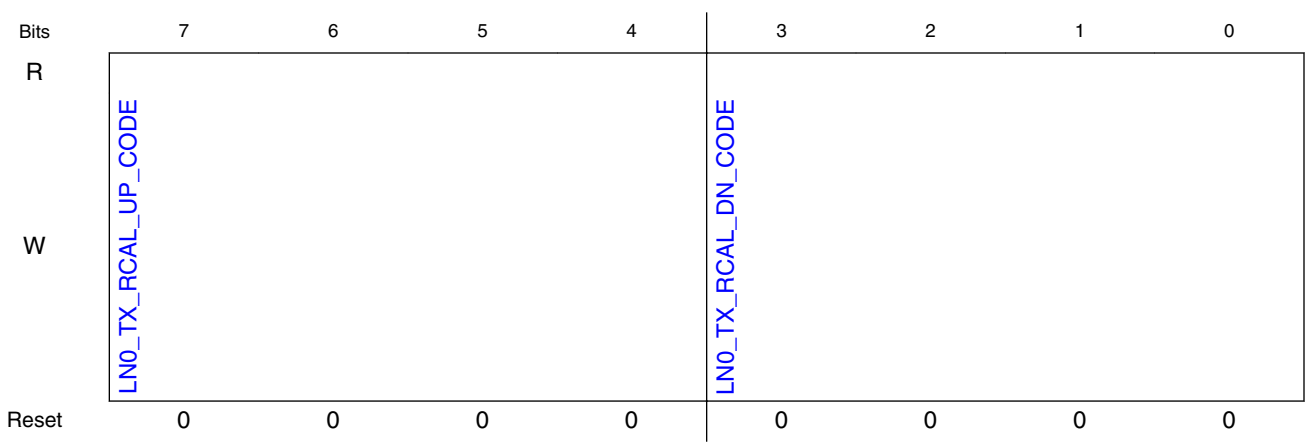
Field	Function
6 LN0_RXD_POLARITY	
5 LN0_OVRD_TX_RCAL_RSTN	Override enable for tx_rcal_rstn
4 LN0_TX_RCAL_RSTN	
3-2 LN0_TX_RCAL_UP_OPT_CODE	
1-0 LN0_TX_RCAL_DN_OPT_CODE	

11.5.3.1.317 (TRSV_REG0BE)

11.5.3.1.317.1 Offset

Register	Offset
TRSV_REG0BE	6F8h

11.5.3.1.317.2 Diagram



11.5.3.1.317.3 Fields

Field	Function
7-4 LN0_TX_RCAL_UP_CODE	Termination up control bits. <3> bit is reserved bit., Default code = 011,Min code =000 and Max code is 111
3-0 LN0_TX_RCAL_DN_CODE	Termination down control bits. <3>bit is reserved bit., Default code= 011,Min code =000 and Max code is 111

11.5.3.1.318 (TRSV_REG0BF)

11.5.3.1.318.1 Offset

Register	Offset
TRSV_REG0BF	6FCh

11.5.3.1.318.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_OVRD_TX_RCAL_COMP_OUT	LN0_TX_RCAL_COMP_OUT	LN0_OVRD_TX_RCAL_DONE	LN0_TX_RCAL_DONE	LN0_OVRD_RX_RCAL_RSTN	LN0_RX_RCAL_RSTN	LN0_RX_RCAL_OPT_CODE	
W								
Reset	0	0	0	0	0	0	0	0

11.5.3.1.318.3 Fields

Field	Function
7	Override enable for tx_rcal_comp_out

Table continues on the next page...

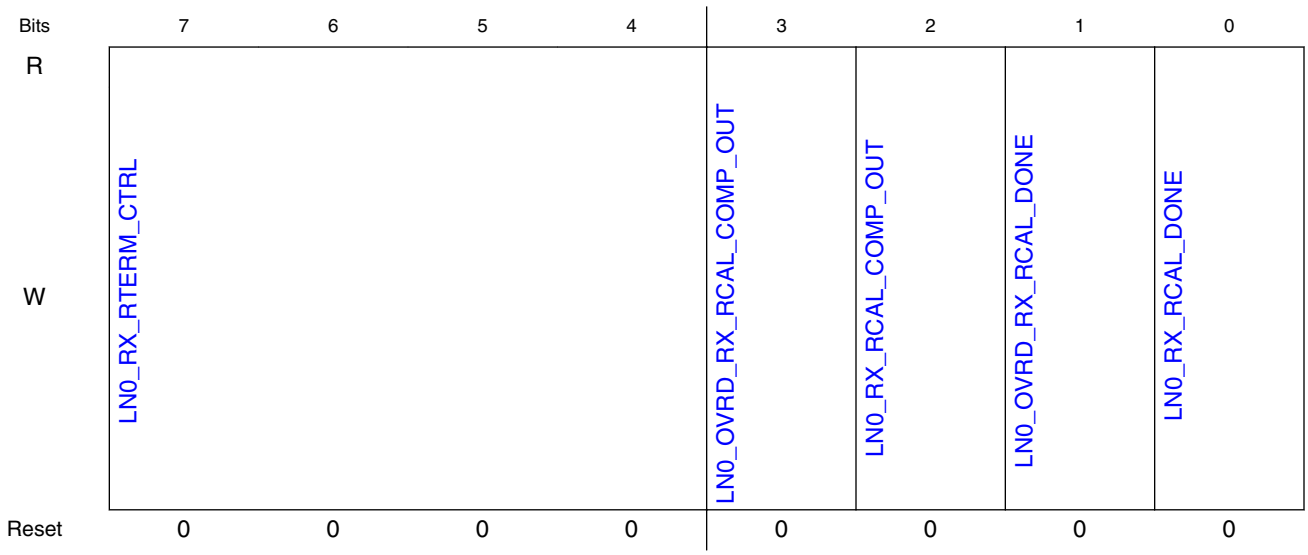
Field	Function
LN0_OVRD_TX_RCAL_COMP_OUT	
6 LN0_TX_RCAL_COMP_OUT	
5 LN0_OVRD_TX_RCAL_DONE	Override enable for tx_rcal_done
4 LN0_TX_RCAL_DONE	Monitoring for TX RCAL done
3 LN0_OVRD_RX_RCAL_RSTN	Override enable for rx_rcal_rstn
2 LN0_RX_RCAL_RSTN	
1-0 LN0_RX_RCAL_OPT_CODE	

11.5.3.1.319 (TRSV_REG0C0)

11.5.3.1.319.1 Offset

Register	Offset
TRSV_REG0C0	700h

11.5.3.1.319.2 Diagram



11.5.3.1.319.3 Fields

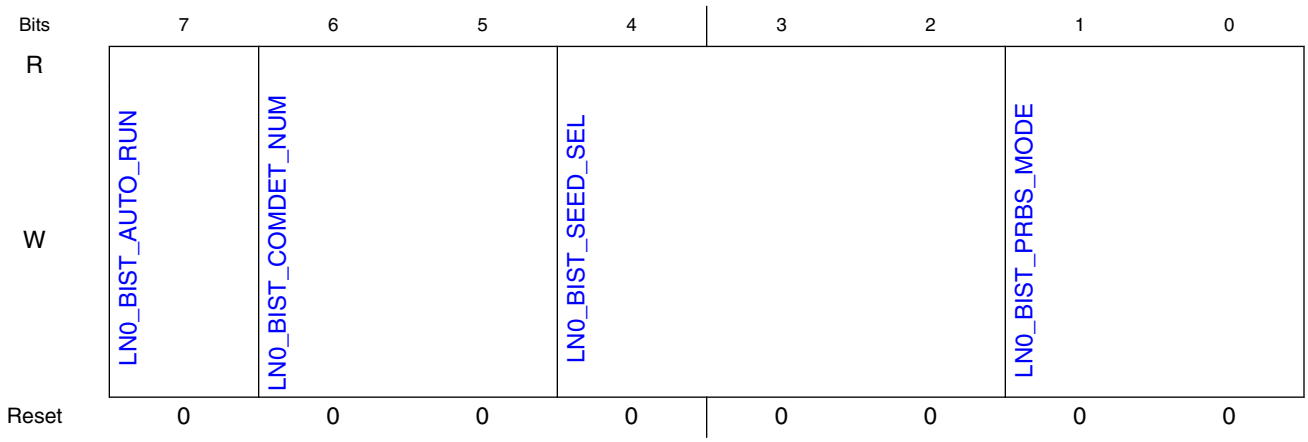
Field	Function
7-4 LN0_RX_RTERM_CTRL	Termination Calibration will send control signals to make 42.5 ohms. MSB<4>=> 1'b0=50,1'b1=42.5(default). Others<3:0> Binary weighted control for RTERM
3 LN0_OVRD_RX_RCAL_COMP_OUT	Override enable for rx_rcal_comp_out
2 LN0_RX_RCAL_COMP_OUT	
1 LN0_OVRD_RX_RCAL_DONE	Override enable for rx_rcal_done
0 LN0_RX_RCAL_DONE	RX RCAL done

11.5.3.1.320 (TRSV_REG0C1)

11.5.3.1.320.1 Offset

Register	Offset
TRSV_REG0C1	704h

11.5.3.1.320.2 Diagram



11.5.3.1.320.3 Fields

Field	Function
7 LN0_BIST_AUTO_RUN	
6-5 LN0_BIST_COMDET_NUM	
4-2 LN0_BIST_SEED_SEL	
1-0 LN0_BIST_PRBS_MODE	

11.5.3.1.321 (TRSV_REG0C2)

11.5.3.1.321.1 Offset

Register	Offset
TRSV_REG0C2	708h

11.5.3.1.321.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_BIST_EN	LN0_BIST_DATA_EN	LN0_BIST_RX_EN	LN0_BIST_RX_HOLD	LN0_BIST_RX_START	LN0_BIST_TX_EN	LN0_BIST_TX_ERRINJ	LN0_BIST_TX_START
W								
Reset	0	0	0	0	0	0	0	0

11.5.3.1.321.3 Fields

Field	Function
7 LN0_BIST_EN	
6 LN0_BIST_DATA_EN	
5 LN0_BIST_RX_EN	
4 LN0_BIST_RX_HOLD	
3 LN0_BIST_RX_START	
2 LN0_BIST_TX_EN	
1 LN0_BIST_TX_ERRINJ	
0	

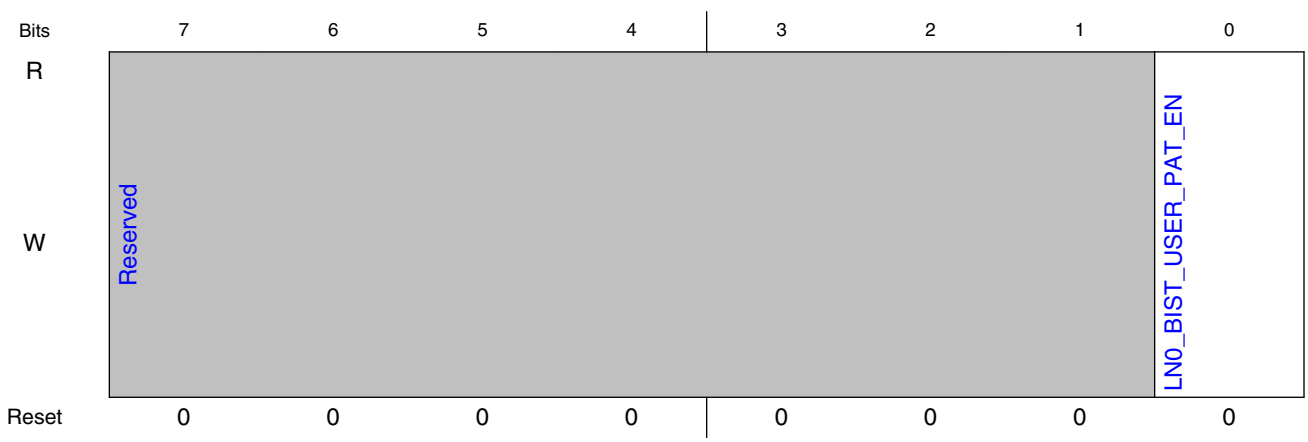
Field	Function
LN0_BIST_TX_START	

11.5.3.1.322 (TRSV_REG0C3)

11.5.3.1.322.1 Offset

Register	Offset
TRSV_REG0C3	70Ch

11.5.3.1.322.2 Diagram



11.5.3.1.322.3 Fields

Field	Function
7-1 —	Reserved
0 LN0_BIST_USER_PAT_EN	

11.5.3.1.323 (TRSV_REG0C4)

11.5.3.1.323.1 Offset

Register	Offset
TRSV_REG0C4	710h

11.5.3.1.323.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_BIST_USER_PAT__79_72							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.323.3 Fields

Field	Function
7-0 LN0_BIST_USER_PAT__79_72	

11.5.3.1.324 (TRSV_REG0C5)

11.5.3.1.324.1 Offset

Register	Offset
TRSV_REG0C5	714h

11.5.3.1.324.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_BIST_USER_PAT__71_64							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.324.3 Fields

Field	Function
7-0 LN0_BIST_USER_PAT__71_64	

11.5.3.1.325 (TRSV_REG0C6)

11.5.3.1.325.1 Offset

Register	Offset
TRSV_REG0C6	718h

11.5.3.1.325.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_BIST_USER_PAT__63_56							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.325.3 Fields

Field	Function
7-0 LN0_BIST_USER_PAT__63_56	

11.5.3.1.326 (TRSV_REG0C7)

11.5.3.1.326.1 Offset

Register	Offset
TRSV_REG0C7	71Ch

11.5.3.1.326.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_BIST_USER_PAT__55_48							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.326.3 Fields

Field	Function
7-0 LN0_BIST_USER_PAT__55_48	

11.5.3.1.327 (TRSV_REG0C8)

11.5.3.1.327.1 Offset

Register	Offset
TRSV_REG0C8	720h

11.5.3.1.327.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_BIST_USER_PAT__47_40							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.327.3 Fields

Field	Function
7-0 LN0_BIST_USER_PAT__47_40	

11.5.3.1.328 (TRSV_REG0C9)

11.5.3.1.328.1 Offset

Register	Offset
TRSV_REG0C9	724h

11.5.3.1.328.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_BIST_USER_PAT__39_32							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.328.3 Fields

Field	Function
7-0 LN0_BIST_USER_PAT__39_32	

11.5.3.1.329 (TRSV_REG0CA)

11.5.3.1.329.1 Offset

Register	Offset
TRSV_REG0CA	728h

11.5.3.1.329.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_BIST_USER_PAT__31_24							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.329.3 Fields

Field	Function
7-0 LN0_BIST_USER_PAT__31_24	

11.5.3.1.330 (TRSV_REG0CB)

11.5.3.1.330.1 Offset

Register	Offset
TRSV_REG0CB	72Ch

11.5.3.1.330.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_BIST_USER_PAT__23_16							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.330.3 Fields

Field	Function
7-0 LN0_BIST_USER_PAT__23_16	

11.5.3.1.331 (TRSV_REG0CC)

11.5.3.1.331.1 Offset

Register	Offset
TRSV_REG0CC	730h

11.5.3.1.331.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_BIST_USER_PAT__15_8							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.331.3 Fields

Field	Function
7-0 LN0_BIST_USER_PAT__15_8	

11.5.3.1.332 (TRSV_REG0CD)

11.5.3.1.332.1 Offset

Register	Offset
TRSV_REG0CD	734h

11.5.3.1.332.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_BIST_USER_PAT__7_0							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.332.3 Fields

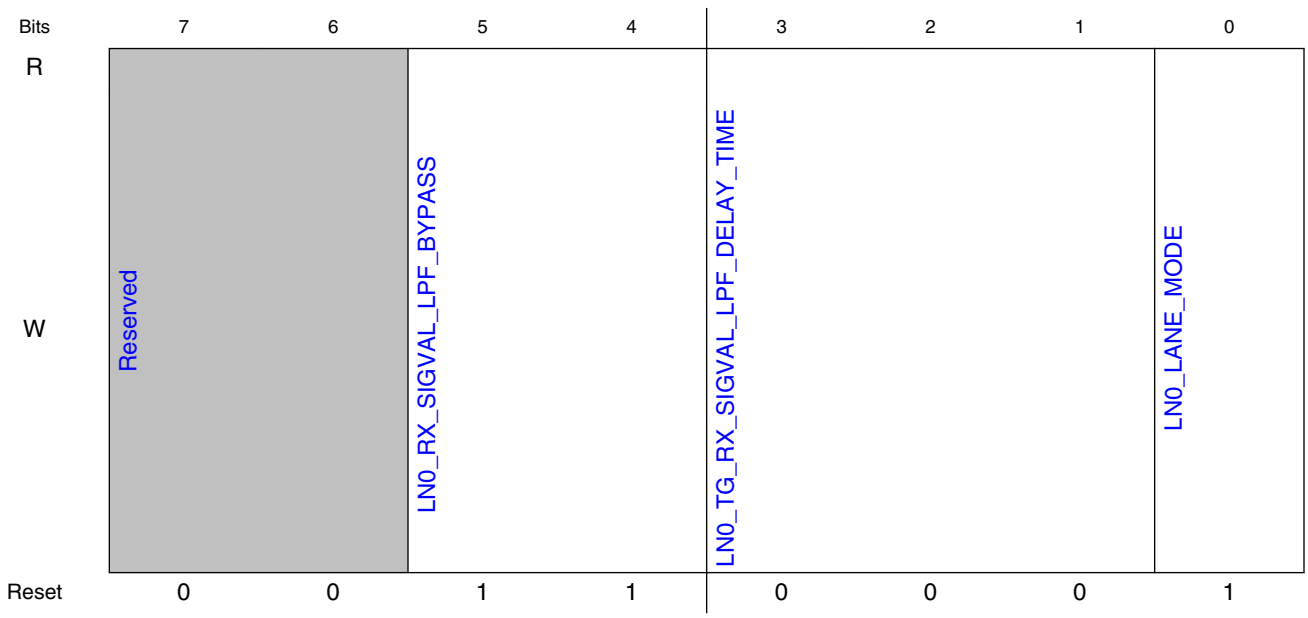
Field	Function
7-0 LN0_BIST_USER_PAT__7_0	

11.5.3.1.333 (TRSV_REG0CE)

11.5.3.1.333.1 Offset

Register	Offset
TRSV_REG0CE	738h

11.5.3.1.333.2 Diagram



11.5.3.1.333.3 Fields

Field	Function
7-6 —	Reserved
5-4 LN0_RX_SIGVAL_LPF_BYPASS	
3-1 LN0_TG_RX_SIGVAL_LPF_DELAY_TIME	
0 LN0_LANE_MODE	Lane operation mode 1 : Aggregation Mode 0 : Lane Bifurcation mode

11.5.3.1.334 (TRSV_REG0CF)

11.5.3.1.334.1 Offset

Register	Offset
TRSV_REG0CF	73Ch

11.5.3.1.334.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	Reserved	LN0_OVRD_LANE_RATE	LN0_LANE_RATE		LN0_LANE_TIMER_SEL	LN0_MISC_TX_CLK_SRC	LN0_MISC_RX_CLK_SRC	LN0_MISC_RX_CLK_INV
W								
Reset	0	0	0	0	0	0	0	0

11.5.3.1.334.3 Fields

Field	Function
7 —	Reserved
6 LN0_OVRD_LANE_RATE	Override enable for lane_rate
5-4 LN0_LANE_RATE	
3 LN0_LANE_TIMER_SEL	
2 LN0_MISC_TX_CLK_SRC	
1	

Table continues on the next page...

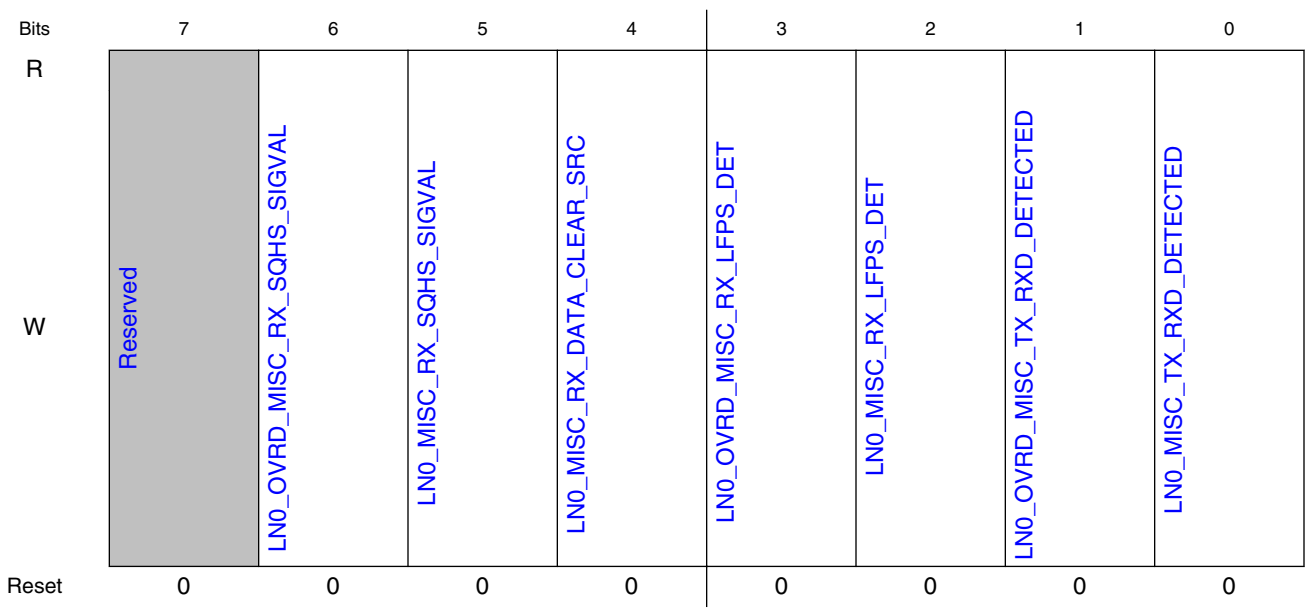
Field	Function
LN0_MISC_RX_CLK_SRC	
0 LN0_MISC_RX_CLK_INV	

11.5.3.1.335 (TRSV_REG0D0)

11.5.3.1.335.1 Offset

Register	Offset
TRSV_REG0D0	740h

11.5.3.1.335.2 Diagram



11.5.3.1.335.3 Fields

Field	Function
7 —	Reserved
6	Override enable for misc_rx_sqhs_sigval

Table continues on the next page...

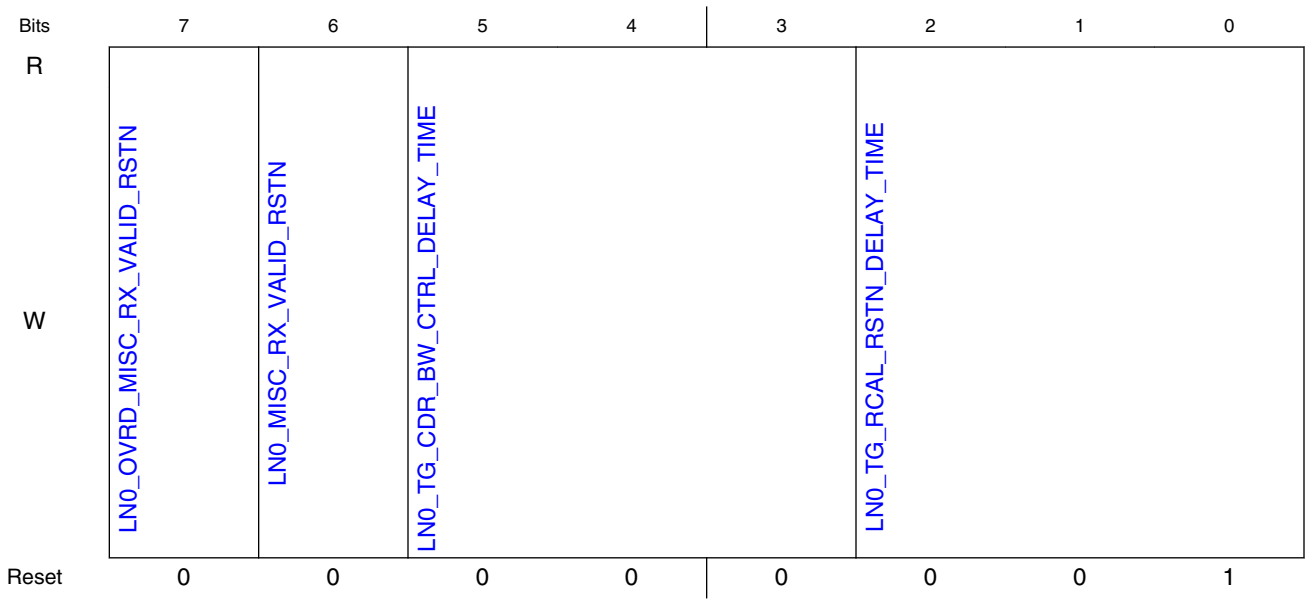
Field	Function
LN0_OVRD_MISC_RX_SQHS_SIGVAL	
5 LN0_MISC_RX_SQHS_SIGVAL	
4 LN0_MISC_RX_DATA_CLEAR_SRC	
3 LN0_OVRD_MISC_RX_LFPS_DET	Override enable for misc_rx_lfps_det
2 LN0_MISC_RX_LFPS_DET	
1 LN0_OVRD_MISC_TX_RXD_DETECTED	Override enable for misc_tx_rxd_detected
0 LN0_MISC_TX_RXD_DETECTED	

11.5.3.1.336 (TRSV_REG0D1)

11.5.3.1.336.1 Offset

Register	Offset
TRSV_REG0D1	744h

11.5.3.1.336.2 Diagram



11.5.3.1.336.3 Fields

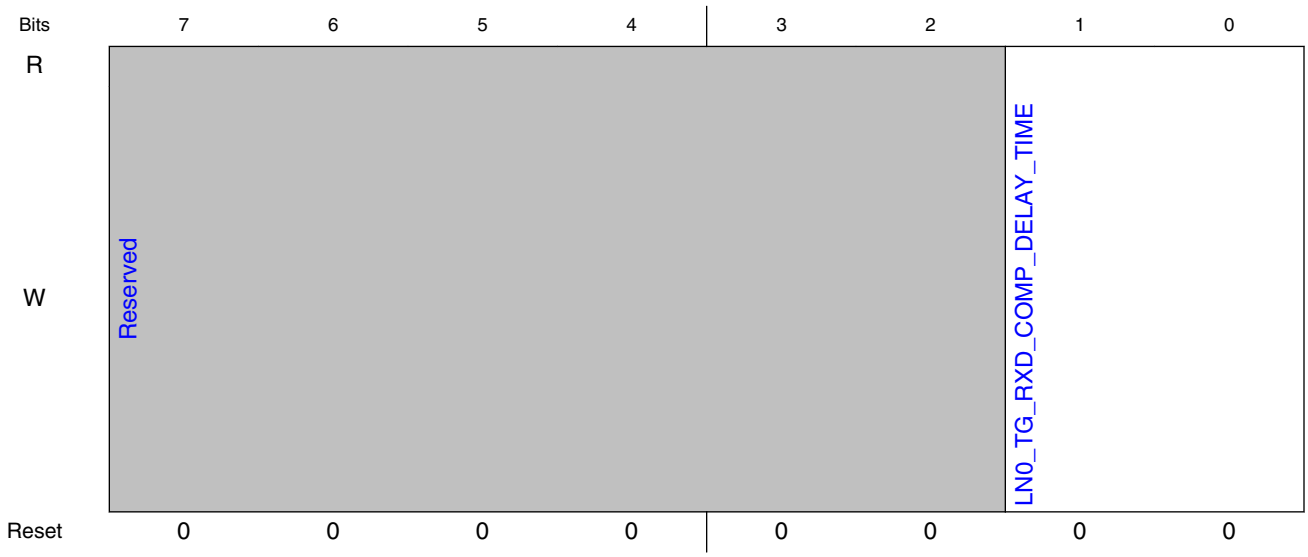
Field	Function
7 LN0_OVRD_MISC_RX_VALID_RSTN	Override enable for misc_rx_valid_rstn
6 LN0_MISC_RX_VALID_RSTN	
5-3 LN0_TG_CDR_BW_CTRL_DELAY_TIME	RX CDR bandwidth change time control
2-0 LN0_TG_RCAL_RSTN_DELAY_TIME	Rx Rcal reset delay time after PLL AFC done

11.5.3.1.337 (TRSV_REG0D2)

11.5.3.1.337.1 Offset

Register	Offset
TRSV_REG0D2	748h

11.5.3.1.337.2 Diagram



11.5.3.1.337.3 Fields

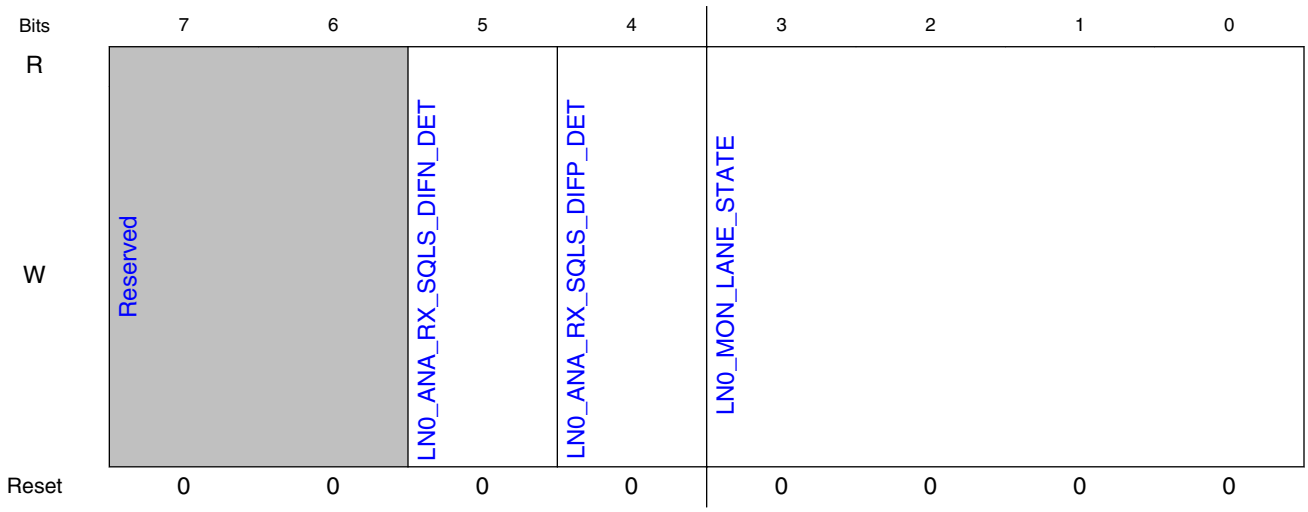
Field	Function
7-2 —	Reserved
1-0 LN0_TG_RXD_COMP_DELAY_TIME	

11.5.3.1.338 (TRSV_REG0D3)

11.5.3.1.338.1 Offset

Register	Offset
TRSV_REG0D3	74Ch

11.5.3.1.338.2 Diagram



11.5.3.1.338.3 Fields

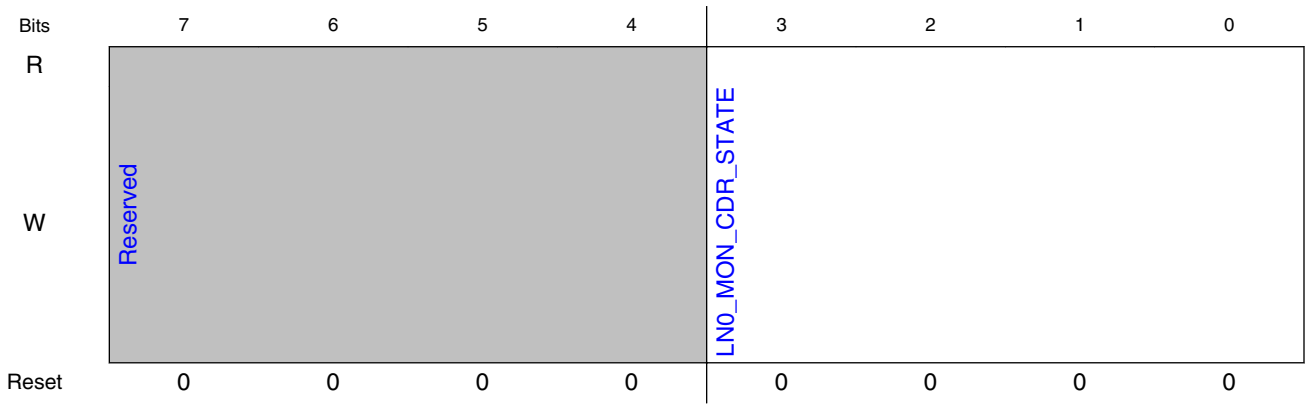
Field	Function
7-6 —	Reserved
5 LN0_ANA_RX_SQLS_DIFN_DET	DIFN Detection signal
4 LN0_ANA_RX_SQLS_DIFP_DET	DIFP Detection signal
3-0 LN0_MON_LANE_STATE	

11.5.3.1.339 (TRSV_REG0D4)

11.5.3.1.339.1 Offset

Register	Offset
TRSV_REG0D4	750h

11.5.3.1.339.2 Diagram



11.5.3.1.339.3 Fields

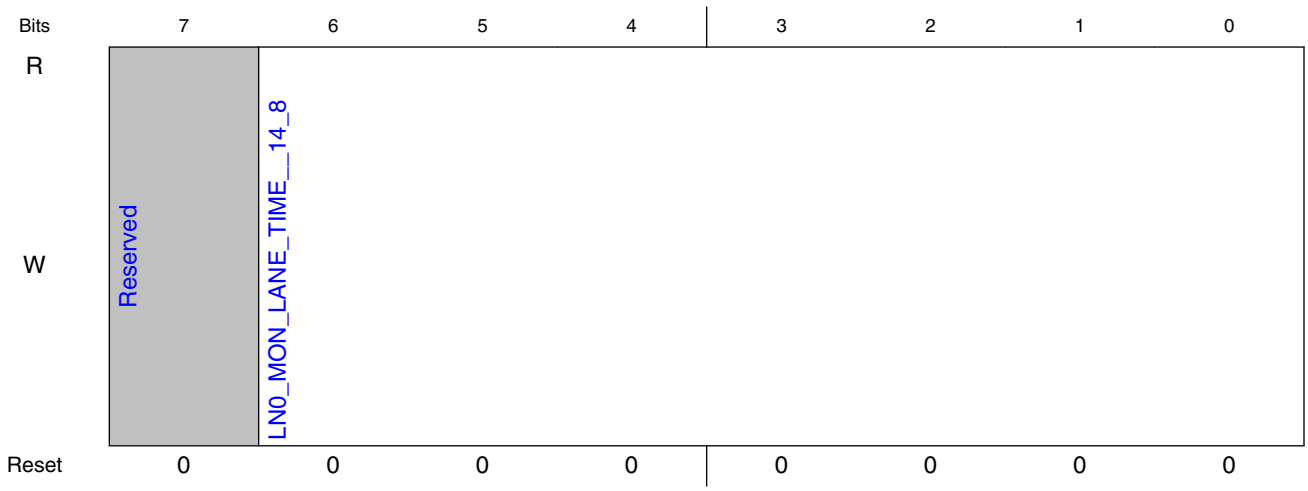
Field	Function
7-4 —	Reserved
3-0 LN0_MON_CDR_STATE	

11.5.3.1.340 (TRSV_REG0D5)

11.5.3.1.340.1 Offset

Register	Offset
TRSV_REG0D5	754h

11.5.3.1.340.2 Diagram



11.5.3.1.340.3 Fields

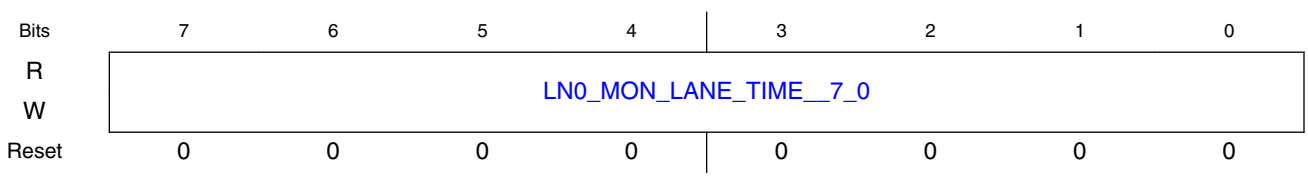
Field	Function
7 —	Reserved
6-0 LN0_MON_LANE_TIME__14_8	

11.5.3.1.341 (TRSV_REG0D6)

11.5.3.1.341.1 Offset

Register	Offset
TRSV_REG0D6	758h

11.5.3.1.341.2 Diagram



11.5.3.1.341.3 Fields

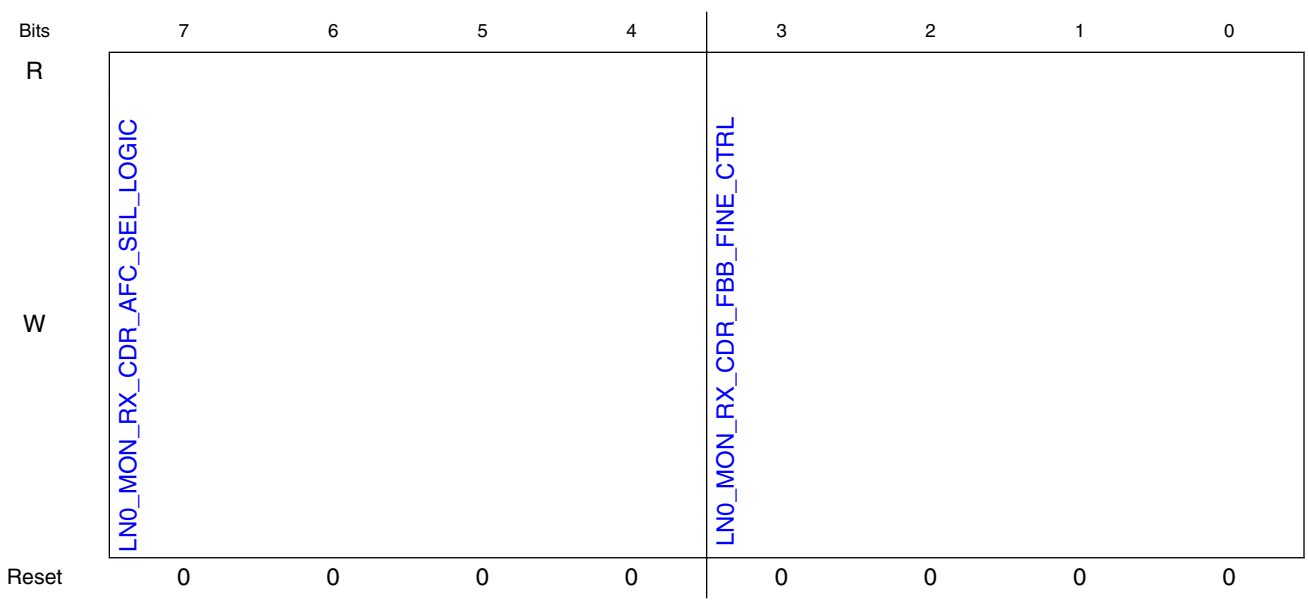
Field	Function
7-0 LN0_MON_LANE_TIME__7_0	

11.5.3.1.342 (TRSV_REG0D7)

11.5.3.1.342.1 Offset

Register	Offset
TRSV_REG0D7	75Ch

11.5.3.1.342.2 Diagram



11.5.3.1.342.3 Fields

Field	Function
7-4 LN0_MON_RX_CDR_AFC_SEL_LOGIC	

Table continues on the next page...

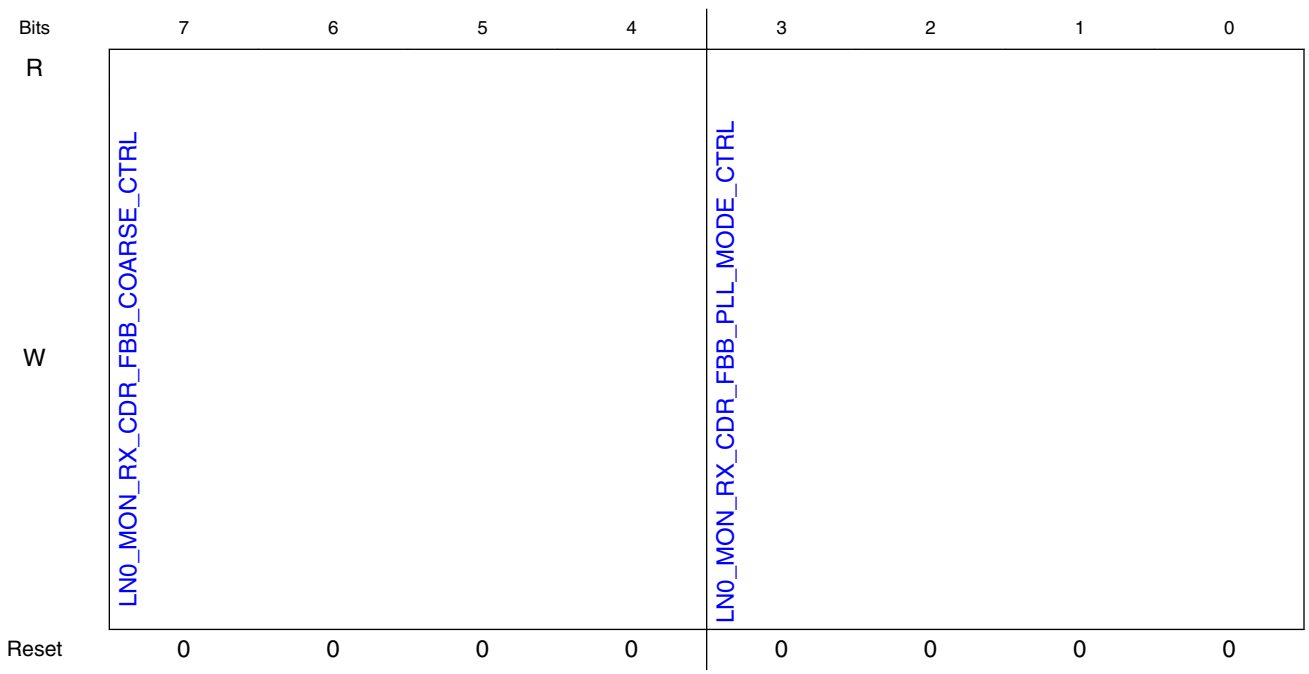
Field	Function
3-0 LN0_MON_RX_CDR_FBB_FINE_CTRL	

11.5.3.1.343 (TRSV_REG0D8)

11.5.3.1.343.1 Offset

Register	Offset
TRSV_REG0D8	760h

11.5.3.1.343.2 Diagram



11.5.3.1.343.3 Fields

Field	Function
7-4	

Table continues on the next page...

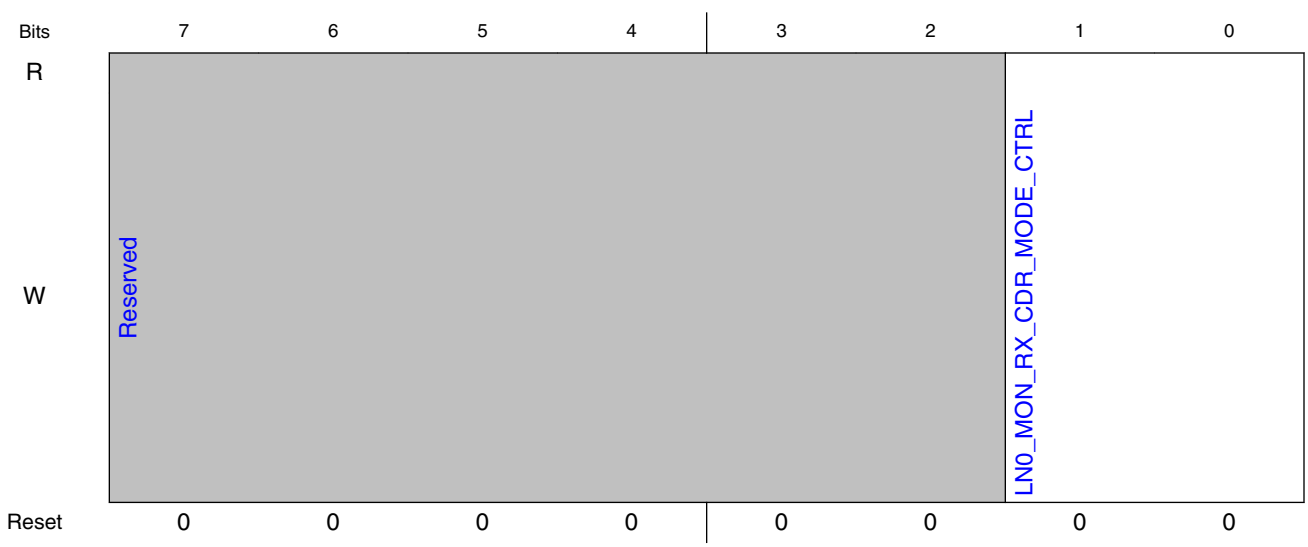
Field	Function
LN0_MON_RX_CDR_FBB_COARSE_CTRL	
3-0 LN0_MON_RX_CDR_FBB_PLL_MODE_CTRL	

11.5.3.1.344 (TRSV_REG0D9)

11.5.3.1.344.1 Offset

Register	Offset
TRSV_REG0D9	764h

11.5.3.1.344.2 Diagram



11.5.3.1.344.3 Fields

Field	Function
7-2 —	Reserved
1-0	

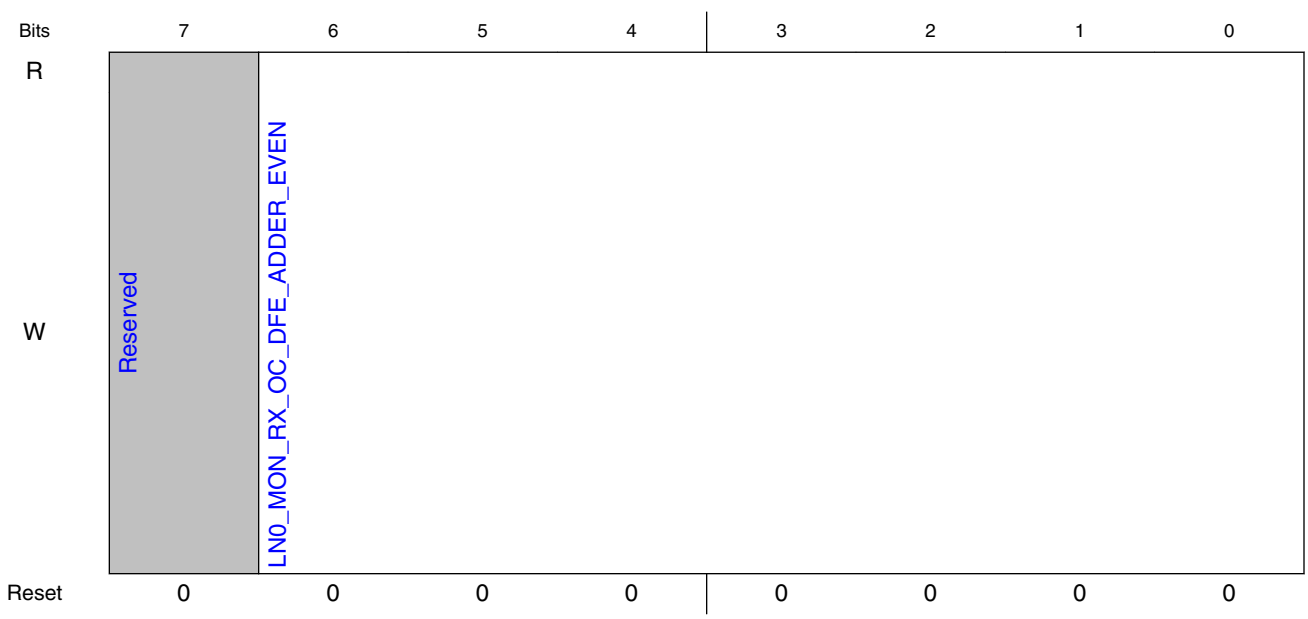
Field	Function
LN0_MON_RX_CDR_MODE_CTRL	

11.5.3.1.345 (TRSV_REG0DA)

11.5.3.1.345.1 Offset

Register	Offset
TRSV_REG0DA	768h

11.5.3.1.345.2 Diagram



11.5.3.1.345.3 Fields

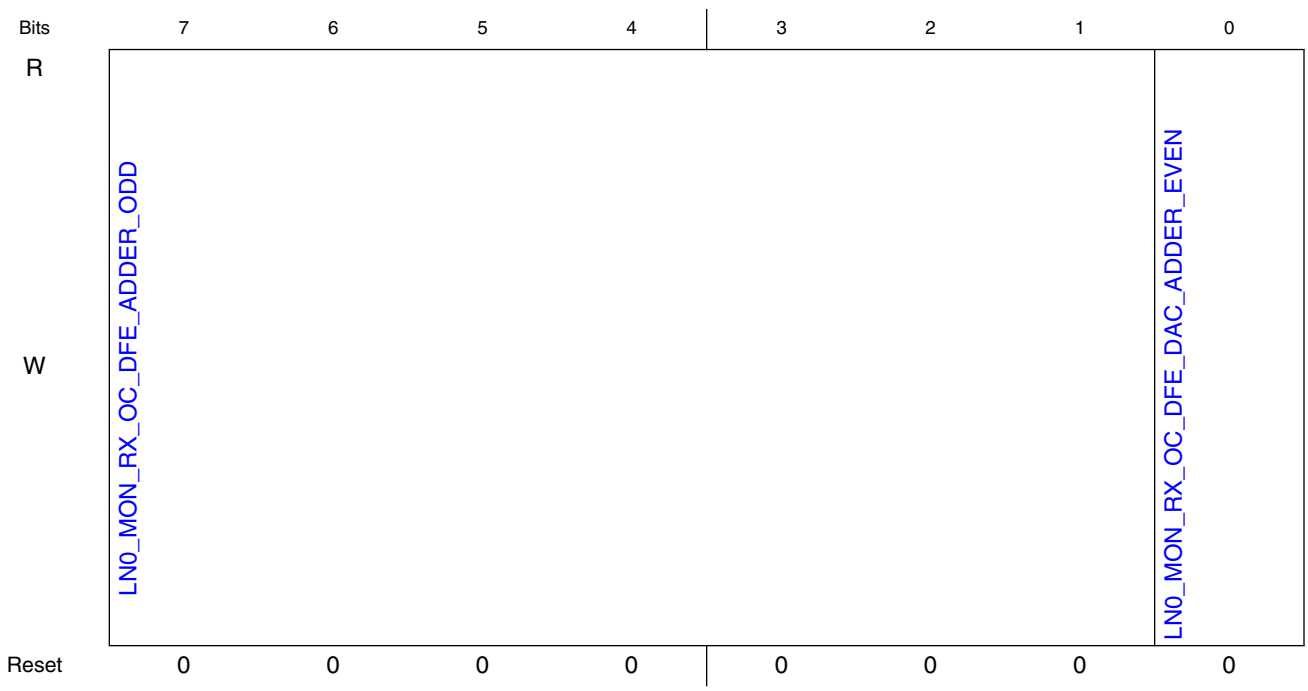
Field	Function
7 —	Reserved
6-0 LN0_MON_RX_OC_DFE_ADDER_EVEN	

11.5.3.1.346 (TRSV_REG0DB)

11.5.3.1.346.1 Offset

Register	Offset
TRSV_REG0DB	76Ch

11.5.3.1.346.2 Diagram



11.5.3.1.346.3 Fields

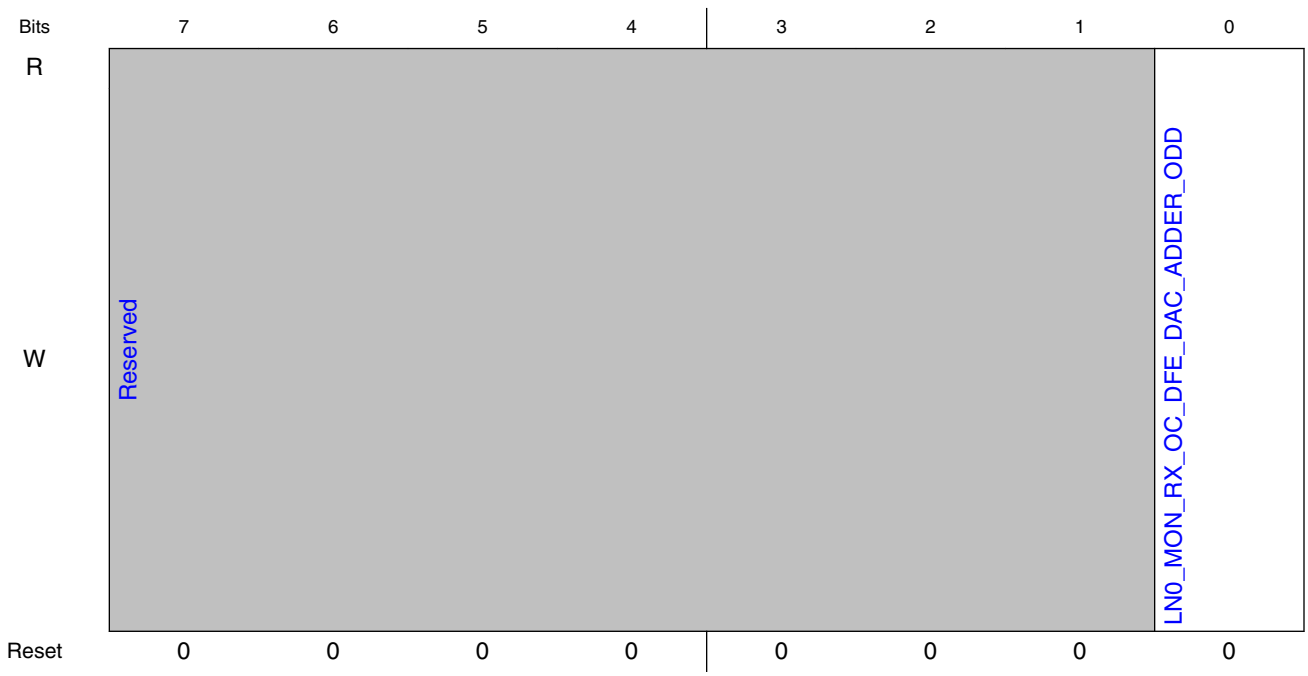
Field	Function
7-1 LN0_MON_RX_OC_DFE_ADDER_ODD	
0 LN0_MON_RX_OC_DFE_DAC_ADDER_EVEN	

11.5.3.1.347 (TRSV_REG0DC)

11.5.3.1.347.1 Offset

Register	Offset
TRSV_REG0DC	770h

11.5.3.1.347.2 Diagram



11.5.3.1.347.3 Fields

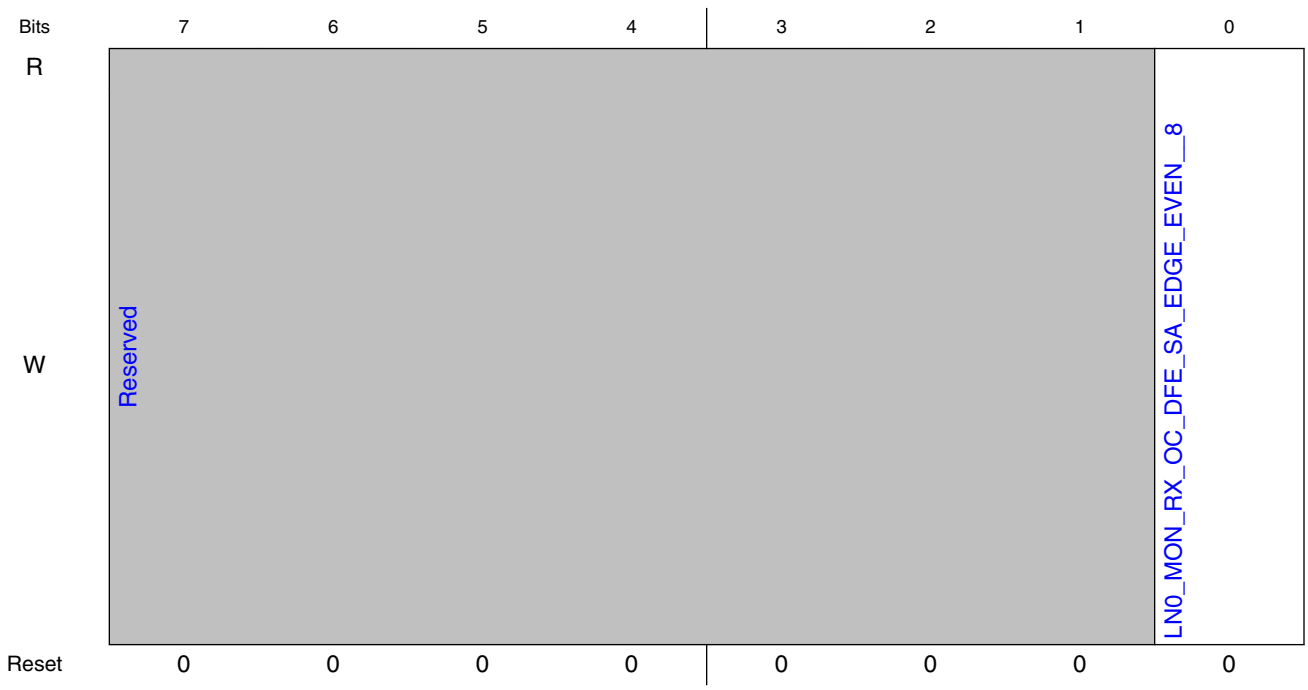
Field	Function
7-1 —	Reserved
0 LN0_MON_RX_OC_DFE_DAC_ADDER_ODD	

11.5.3.1.348 (TRSV_REG0DD)

11.5.3.1.348.1 Offset

Register	Offset
TRSV_REG0DD	774h

11.5.3.1.348.2 Diagram



11.5.3.1.348.3 Fields

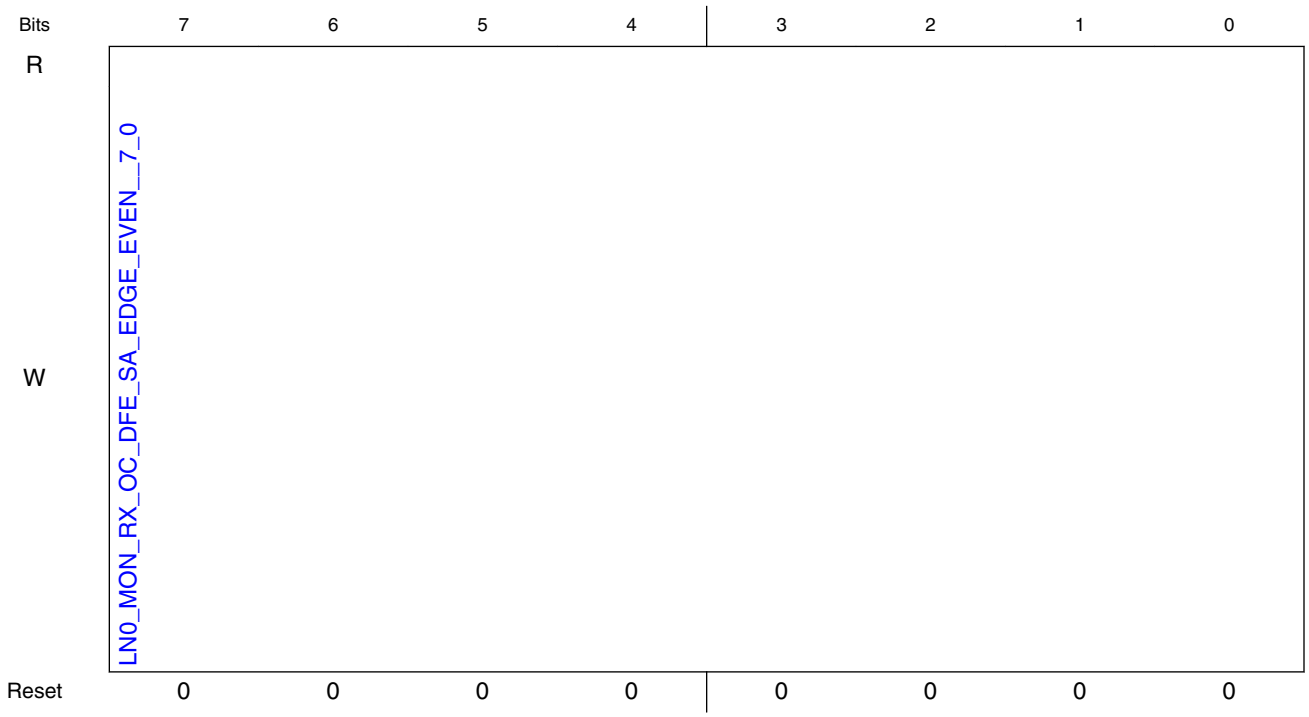
Field	Function
7-1 —	Reserved
0 LN0_MON_RX_OC_DFE_SA_EDGE_EVEN__8	

11.5.3.1.349 (TRSV_REG0DE)

11.5.3.1.349.1 Offset

Register	Offset
TRSV_REG0DE	778h

11.5.3.1.349.2 Diagram



11.5.3.1.349.3 Fields

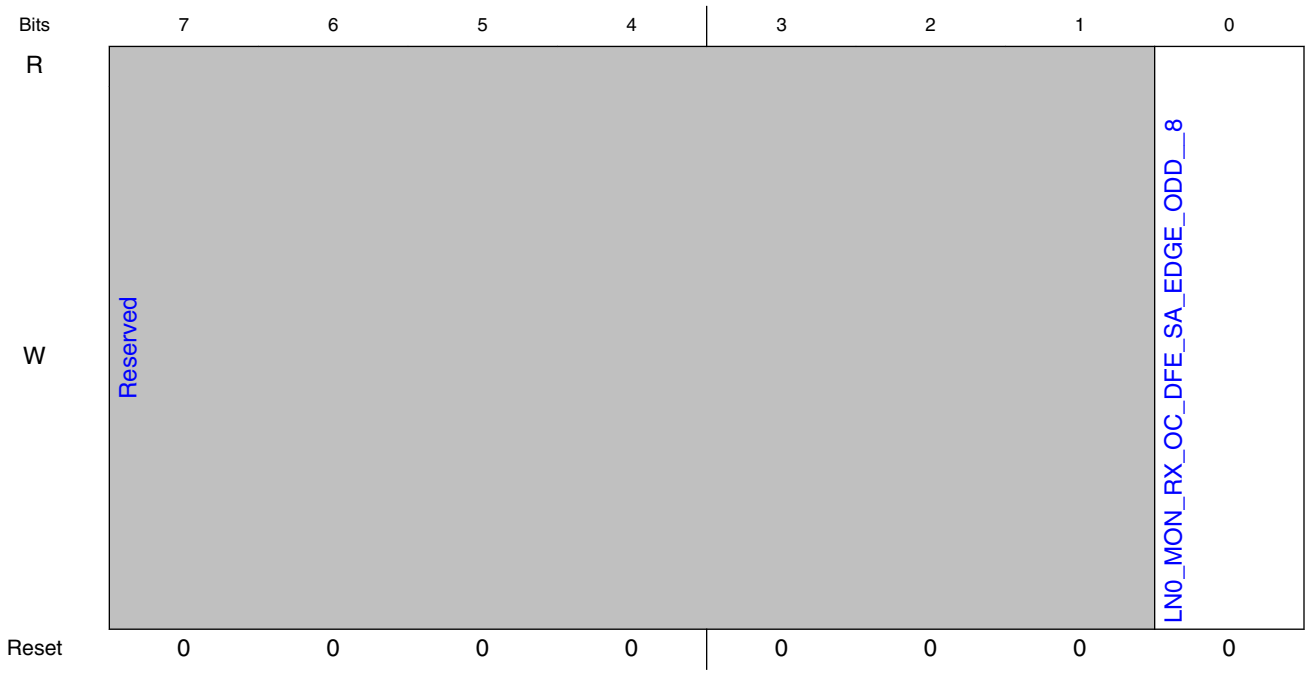
Field	Function
7-0 LN0_MON_RX_OC_DFE_SA_EDGE_EVEN__7_0	

11.5.3.1.350 (TRSV_REG0DF)

11.5.3.1.350.1 Offset

Register	Offset
TRSV_REG0DF	77Ch

11.5.3.1.350.2 Diagram



11.5.3.1.350.3 Fields

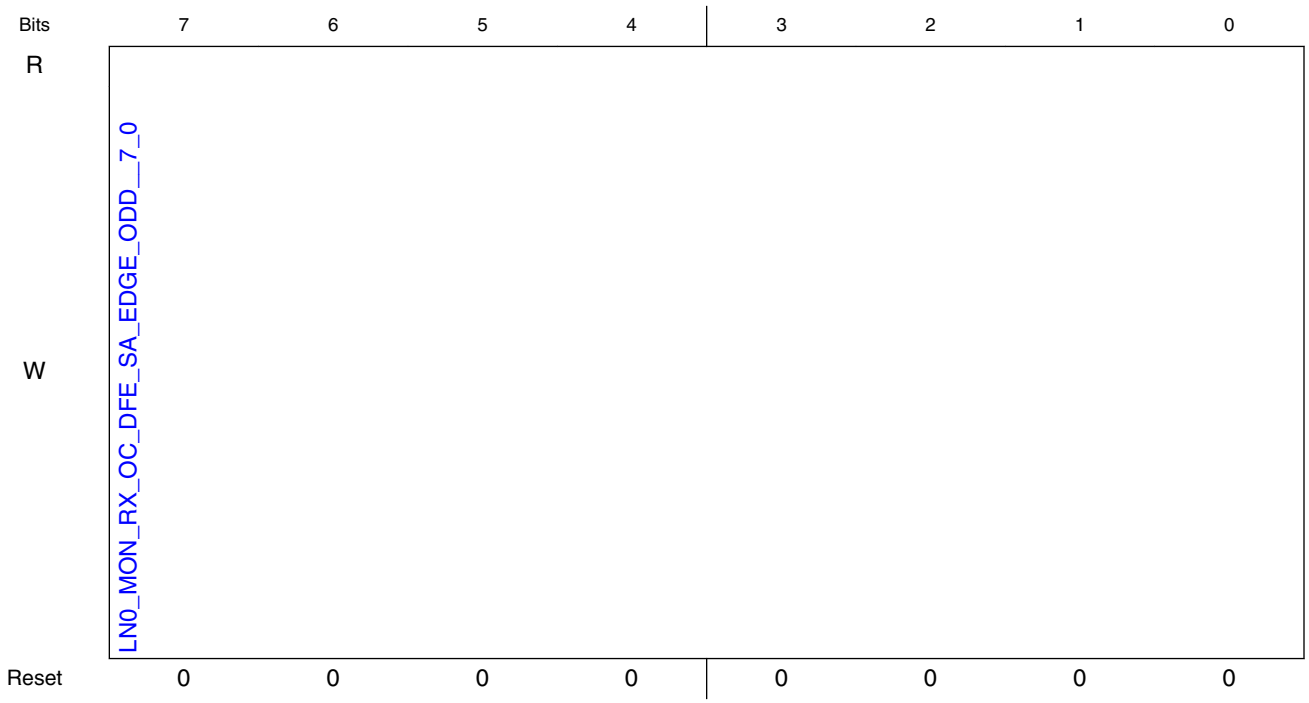
Field	Function
7-1 —	Reserved
0 LN0_MON_RX_OC_DFE_SA_EDGE_ODD__8	

11.5.3.1.351 (TRSV_REG0E0)

11.5.3.1.351.1 Offset

Register	Offset
TRSV_REG0E0	780h

11.5.3.1.351.2 Diagram



11.5.3.1.351.3 Fields

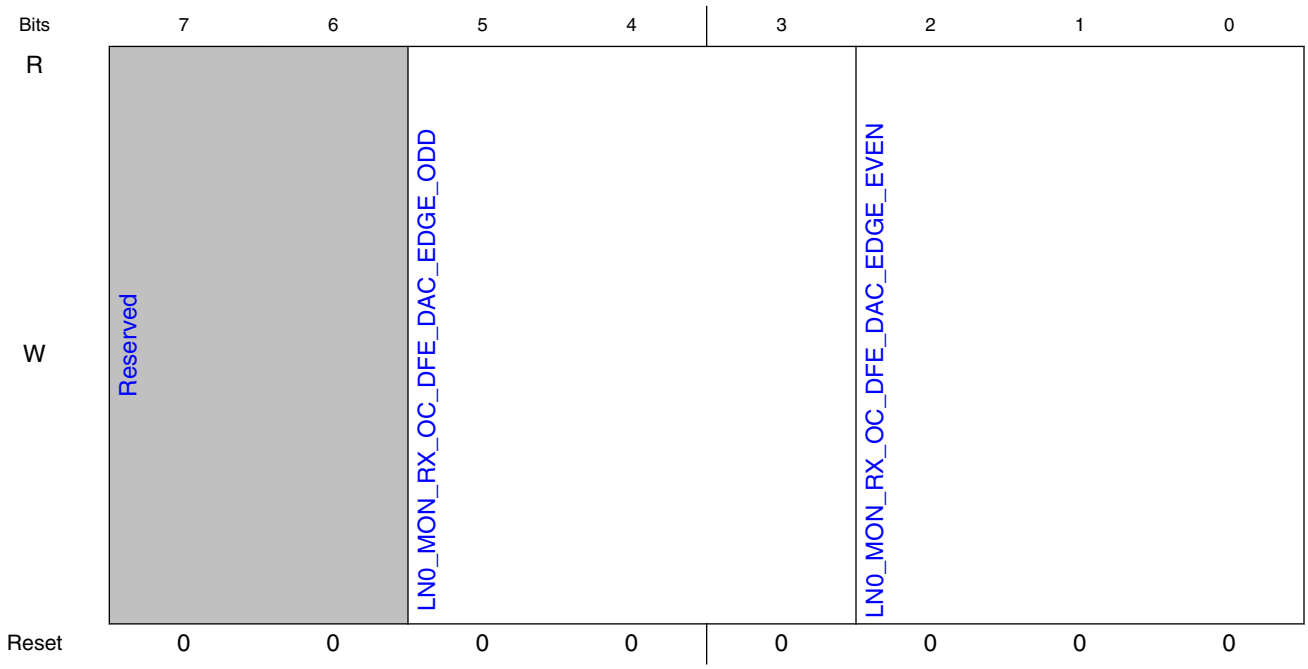
Field	Function
7-0 LN0_MON_RX_OC_DFE_SA_EDGE_ODD__7_0	

11.5.3.1.352 (TRSV_REG0E1)

11.5.3.1.352.1 Offset

Register	Offset
TRSV_REG0E1	784h

11.5.3.1.352.2 Diagram



11.5.3.1.352.3 Fields

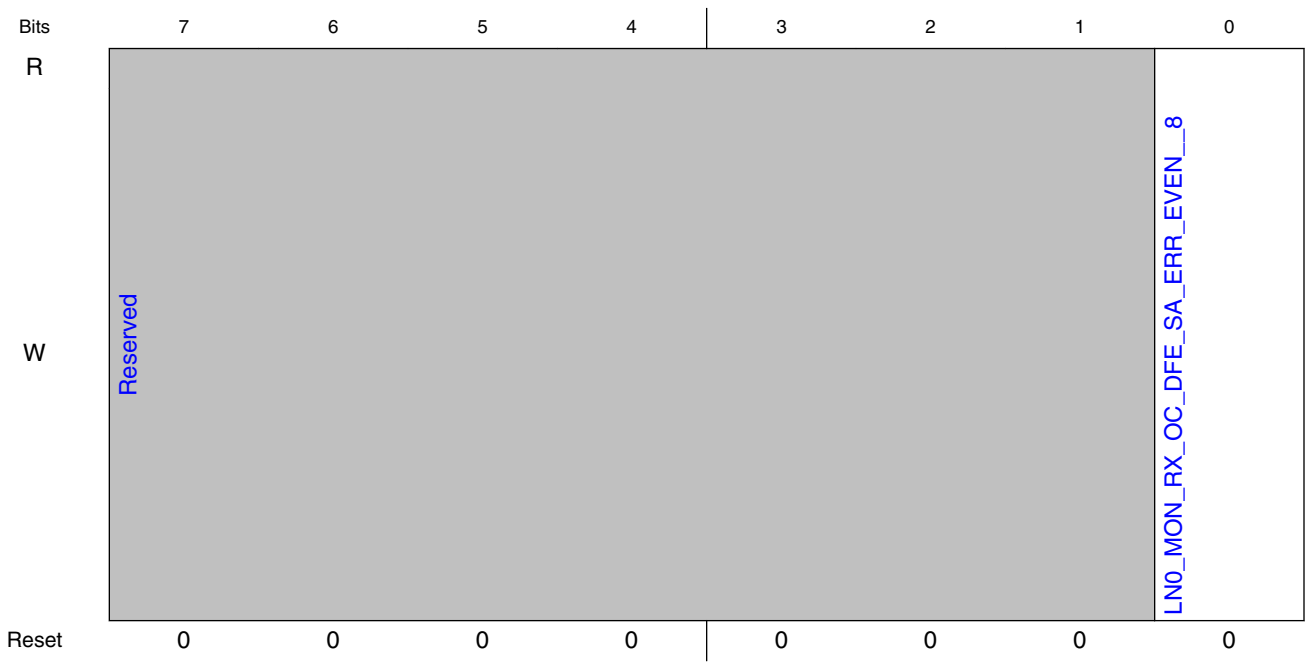
Field	Function
7-6 —	Reserved
5-3 LN0_MON_RX_OC_DFE_DAC_EDGE_ODD	
2-0 LN0_MON_RX_OC_DFE_DAC_EDGE_EVEN	

11.5.3.1.353 (TRSV_REG0E2)

11.5.3.1.353.1 Offset

Register	Offset
TRSV_REG0E2	788h

11.5.3.1.353.2 Diagram



11.5.3.1.353.3 Fields

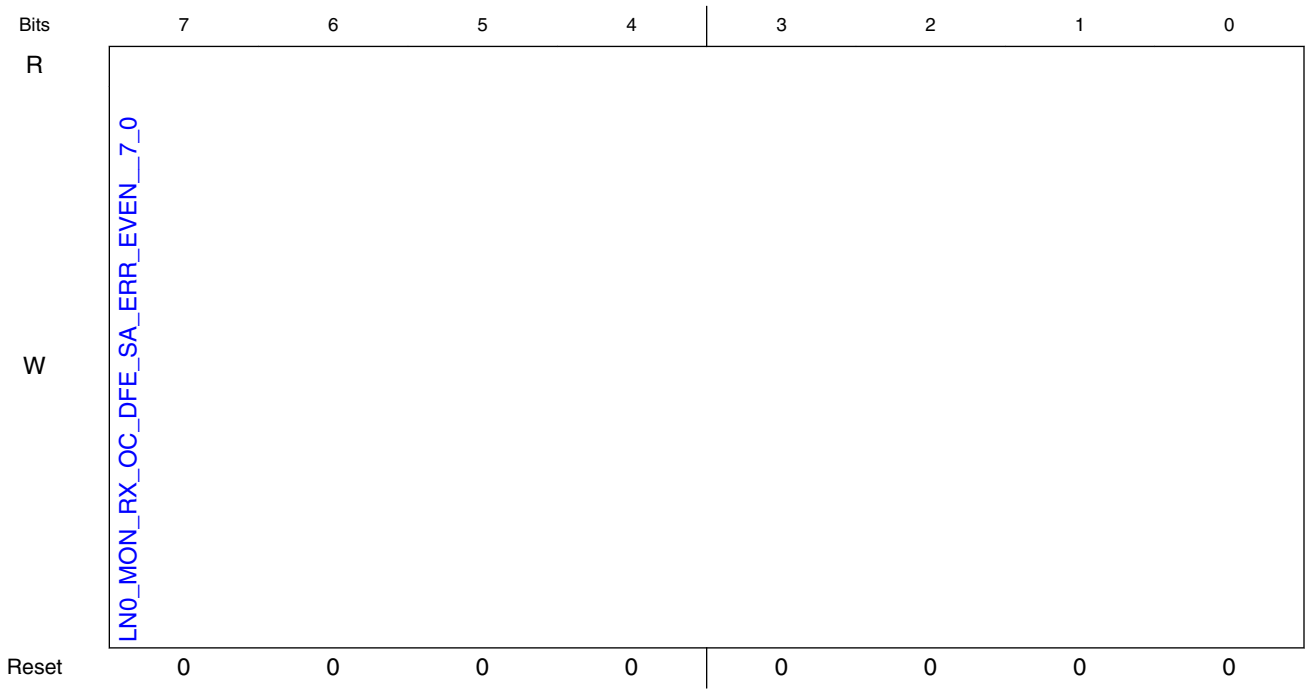
Field	Function
7-1 —	Reserved
0 LN0_MON_RX_OC_DFE_SA_ERR_EVEN__8	

11.5.3.1.354 (TRSV_REG0E3)

11.5.3.1.354.1 Offset

Register	Offset
TRSV_REG0E3	78Ch

11.5.3.1.354.2 Diagram



11.5.3.1.354.3 Fields

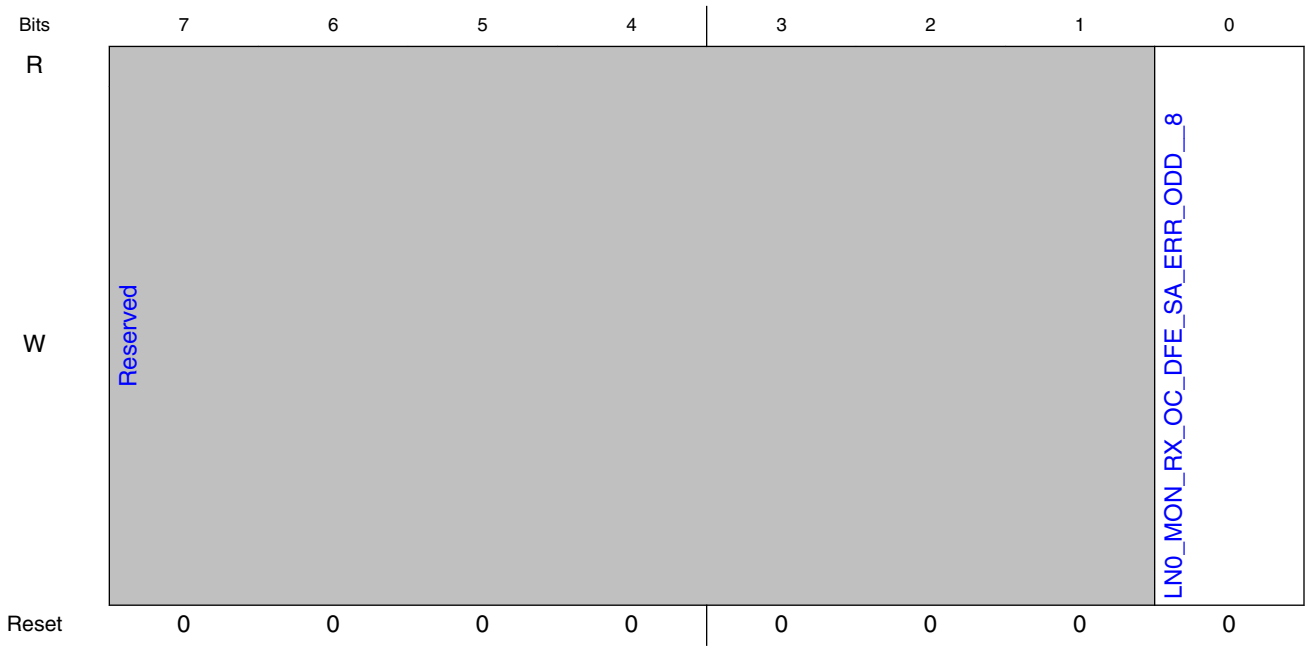
Field	Function
7-0 LN0_MON_RX_OC_DFE_SA_ERR_EVEN__7_0	

11.5.3.1.355 (TRSV_REG0E4)

11.5.3.1.355.1 Offset

Register	Offset
TRSV_REG0E4	790h

11.5.3.1.355.2 Diagram



11.5.3.1.355.3 Fields

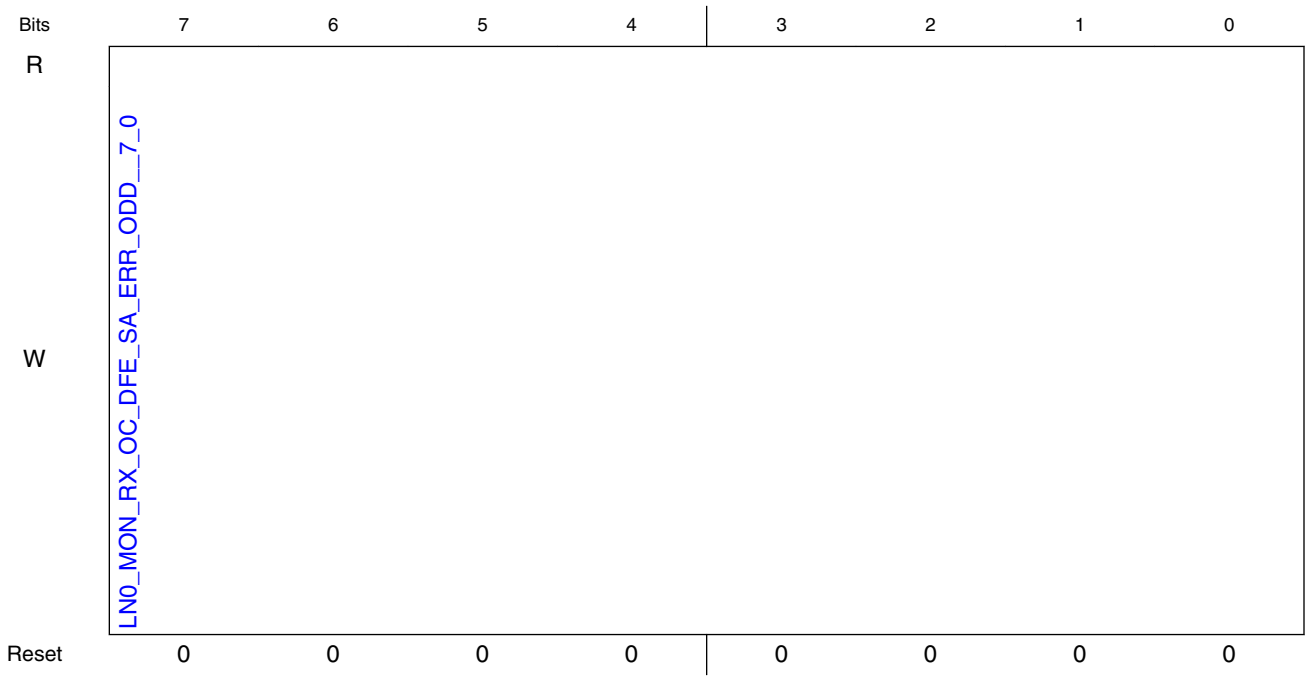
Field	Function
7-1 —	Reserved
0 LN0_MON_RX_OC_DFE_SA_ERR_ODD__8	

11.5.3.1.356 (TRSV_REG0E5)

11.5.3.1.356.1 Offset

Register	Offset
TRSV_REG0E5	794h

11.5.3.1.356.2 Diagram



11.5.3.1.356.3 Fields

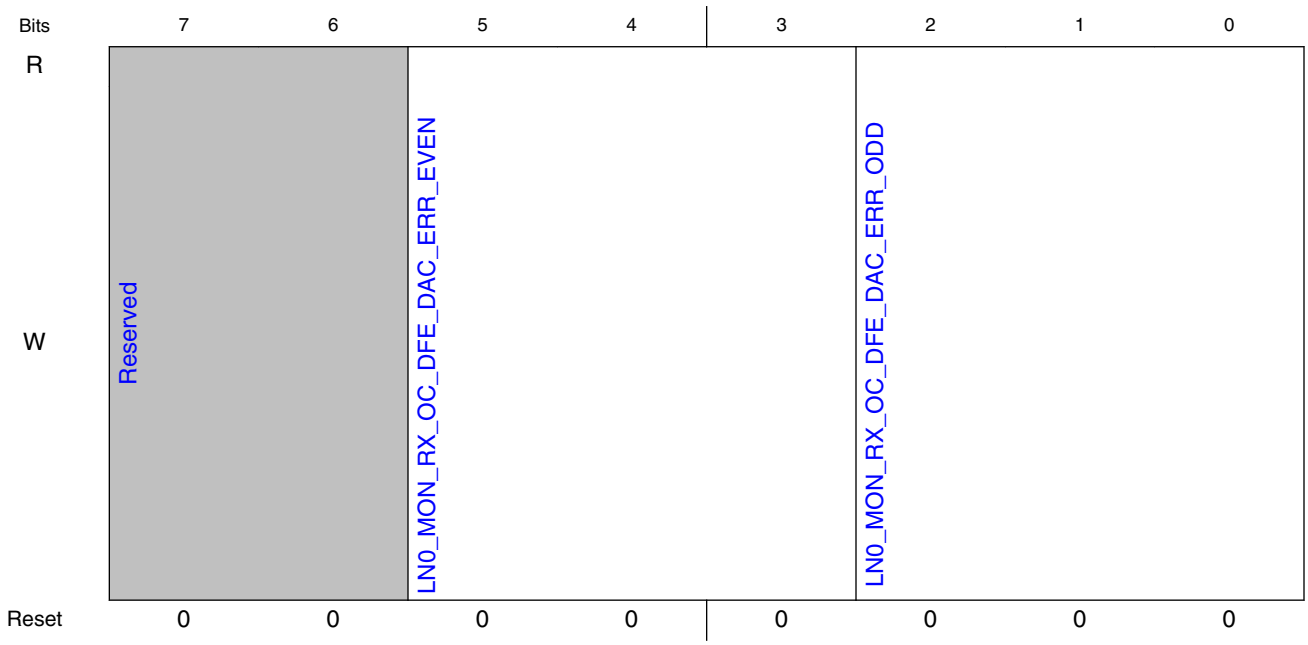
Field	Function
7-0 LN0_MON_RX_OC_DFE_SA_ERR_ODD__7_0	

11.5.3.1.357 (TRSV_REG0E6)

11.5.3.1.357.1 Offset

Register	Offset
TRSV_REG0E6	798h

11.5.3.1.357.2 Diagram



11.5.3.1.357.3 Fields

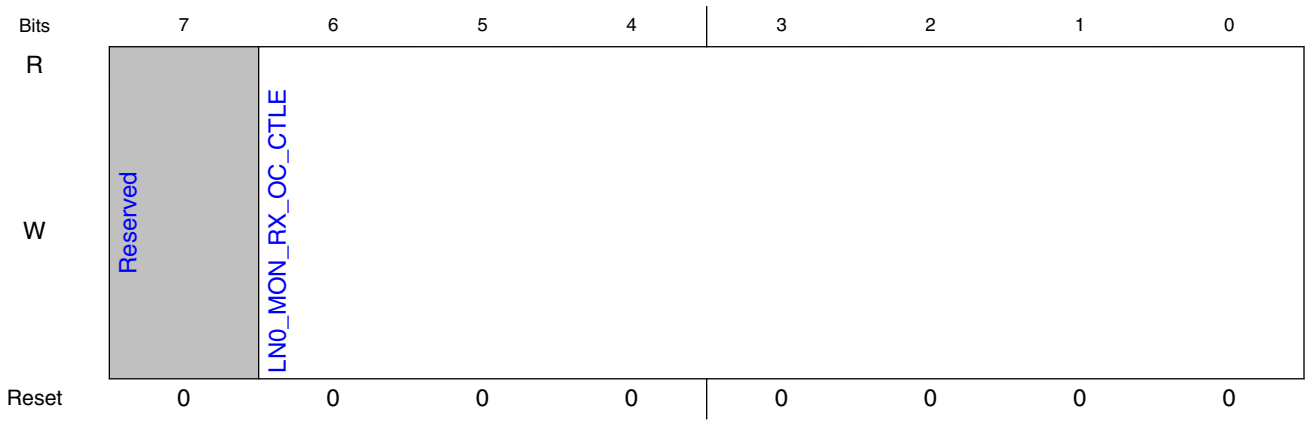
Field	Function
7-6 —	Reserved
5-3 LN0_MON_RX_OC_DFE_DAC_ERR_EVEN	
2-0 LN0_MON_RX_OC_DFE_DAC_ERR_ODD	

11.5.3.1.358 (TRSV_REG0E7)

11.5.3.1.358.1 Offset

Register	Offset
TRSV_REG0E7	79Ch

11.5.3.1.358.2 Diagram



11.5.3.1.358.3 Fields

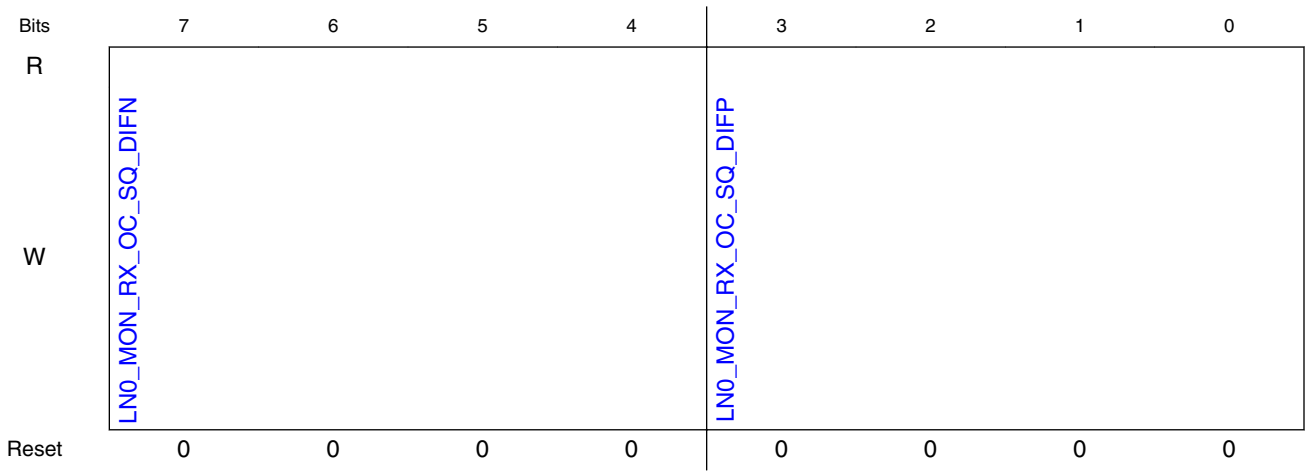
Field	Function
7 —	Reserved
6-0 LN0_MON_RX_OC_CTLE	

11.5.3.1.359 (TRSV_REG0E8)

11.5.3.1.359.1 Offset

Register	Offset
TRSV_REG0E8	7A0h

11.5.3.1.359.2 Diagram



11.5.3.1.359.3 Fields

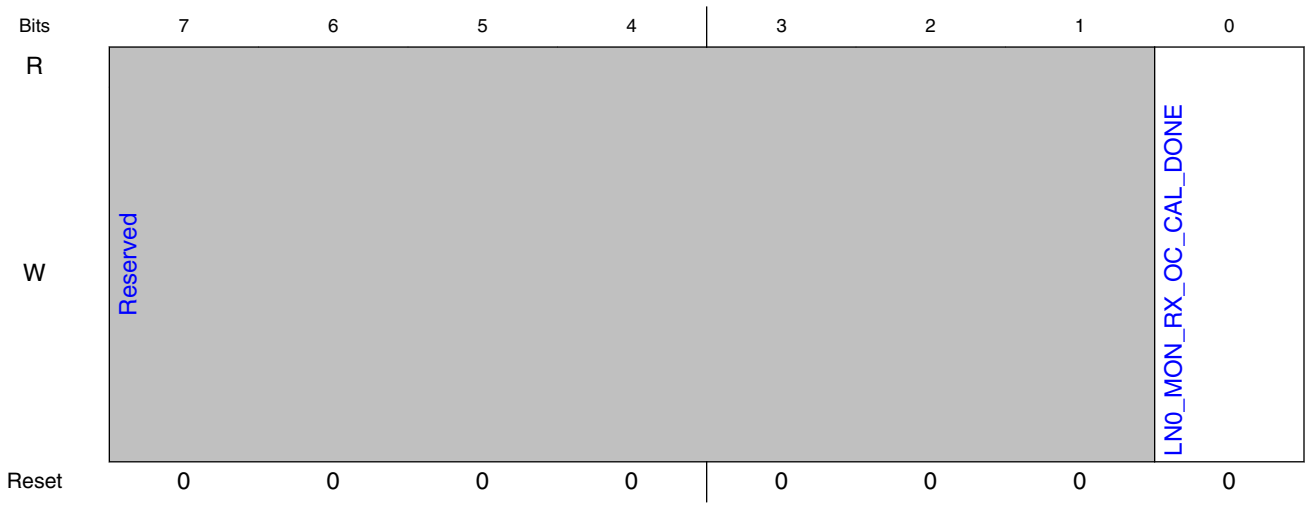
Field	Function
7-4 LN0_MON_RX_OC_SQ_DIFN	
3-0 LN0_MON_RX_OC_SQ_DIFP	

11.5.3.1.360 (TRSV_REG0E9)

11.5.3.1.360.1 Offset

Register	Offset
TRSV_REG0E9	7A4h

11.5.3.1.360.2 Diagram



11.5.3.1.360.3 Fields

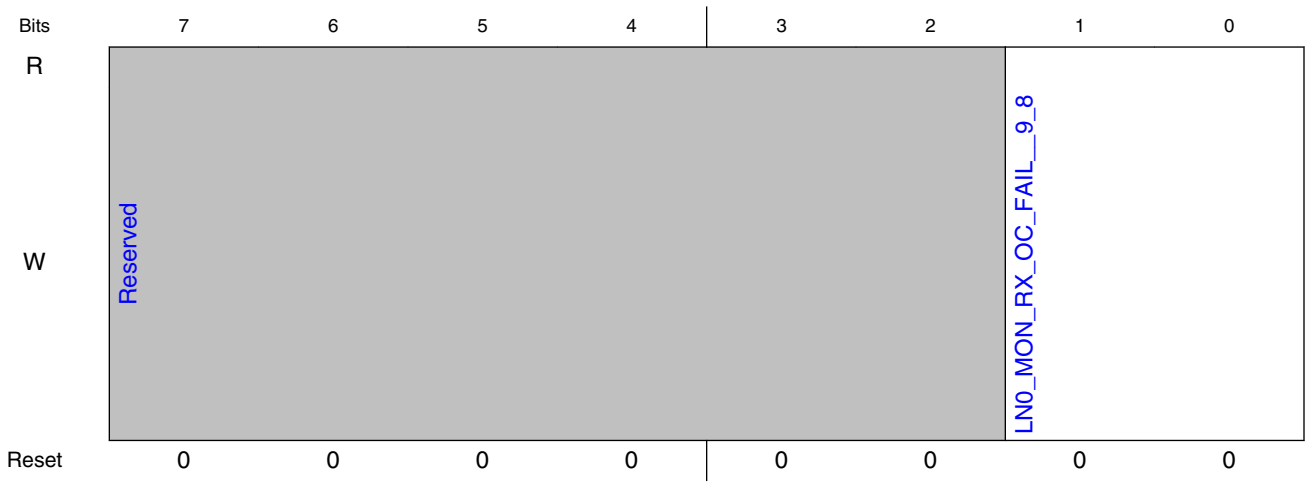
Field	Function
7-1 —	Reserved
0 LN0_MON_RX_OC_CAL_DONE	

11.5.3.1.361 (TRSV_REG0EA)

11.5.3.1.361.1 Offset

Register	Offset
TRSV_REG0EA	7A8h

11.5.3.1.361.2 Diagram



11.5.3.1.361.3 Fields

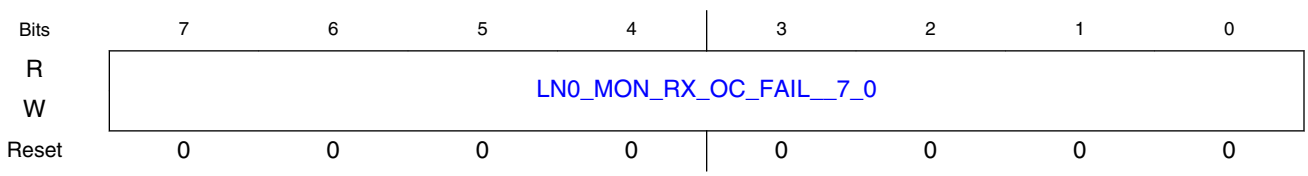
Field	Function
7-2 —	Reserved
1-0 LN0_MON_RX_OC_FAIL__9_8	

11.5.3.1.362 (TRSV_REG0EB)

11.5.3.1.362.1 Offset

Register	Offset
TRSV_REG0EB	7ACh

11.5.3.1.362.2 Diagram



11.5.3.1.362.3 Fields

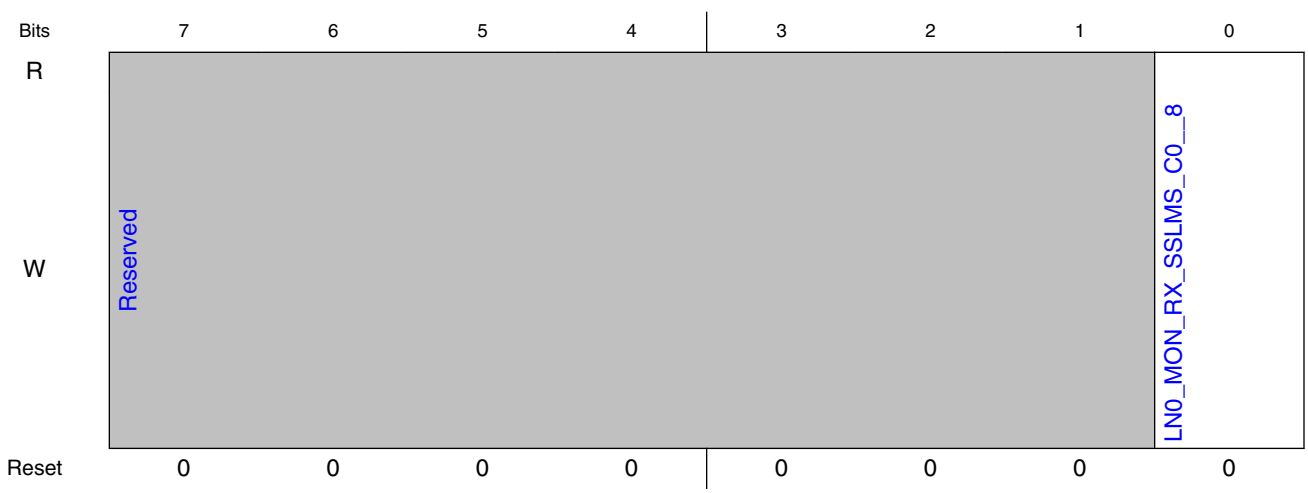
Field	Function
7-0 LN0_MON_RX_OC_FAIL__7_0	

11.5.3.1.363 (TRSV_REG0EC)

11.5.3.1.363.1 Offset

Register	Offset
TRSV_REG0EC	7B0h

11.5.3.1.363.2 Diagram



11.5.3.1.363.3 Fields

Field	Function
7-1 —	Reserved
0 LN0_MON_RX_SSLMS_C0__8	

11.5.3.1.364 (TRSV_REG0ED)

11.5.3.1.364.1 Offset

Register	Offset
TRSV_REG0ED	7B4h

11.5.3.1.364.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	LN0_MON_RX_SSLMS_C0__7_0							
Reset	0	0	0	0	0	0	0	0

11.5.3.1.364.3 Fields

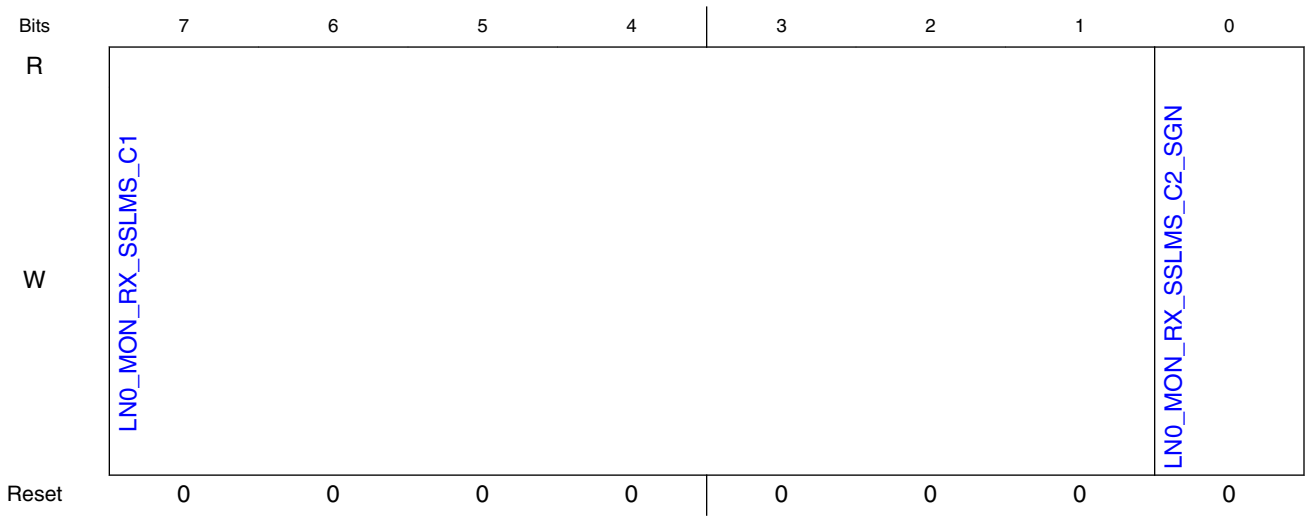
Field	Function
7-0 LN0_MON_RX_SSLMS_C0__7_0	

11.5.3.1.365 (TRSV_REG0EE)

11.5.3.1.365.1 Offset

Register	Offset
TRSV_REG0EE	7B8h

11.5.3.1.365.2 Diagram



11.5.3.1.365.3 Fields

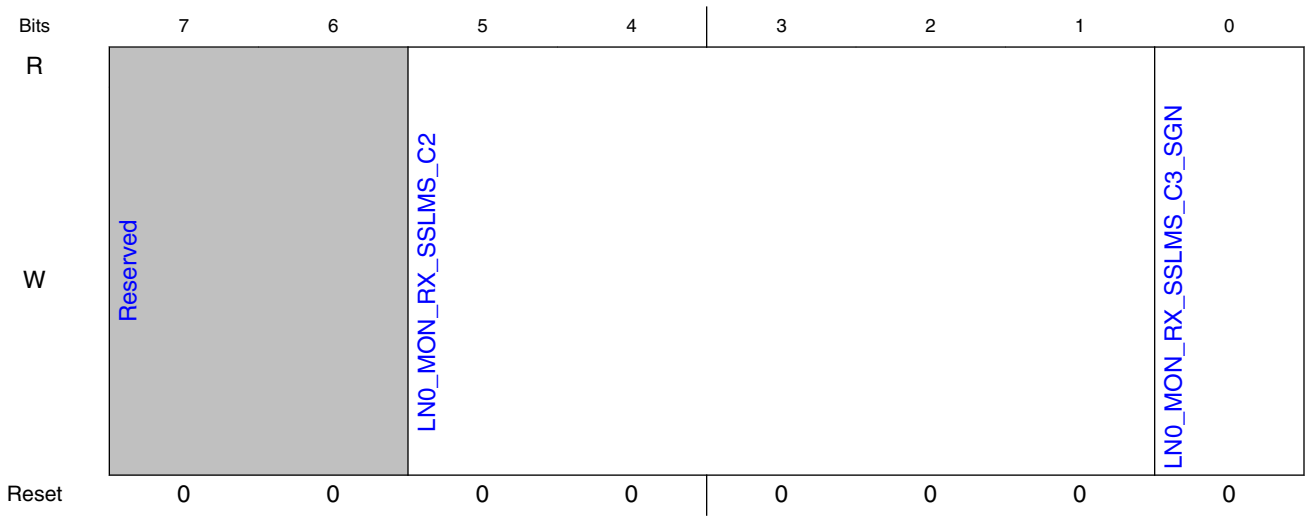
Field	Function
7-1 LN0_MON_RX_SSLMS_C1	
0 LN0_MON_RX_SSLMS_C2_SGN	

11.5.3.1.366 (TRSV_REG0EF)

11.5.3.1.366.1 Offset

Register	Offset
TRSV_REG0EF	7BCh

11.5.3.1.366.2 Diagram



11.5.3.1.366.3 Fields

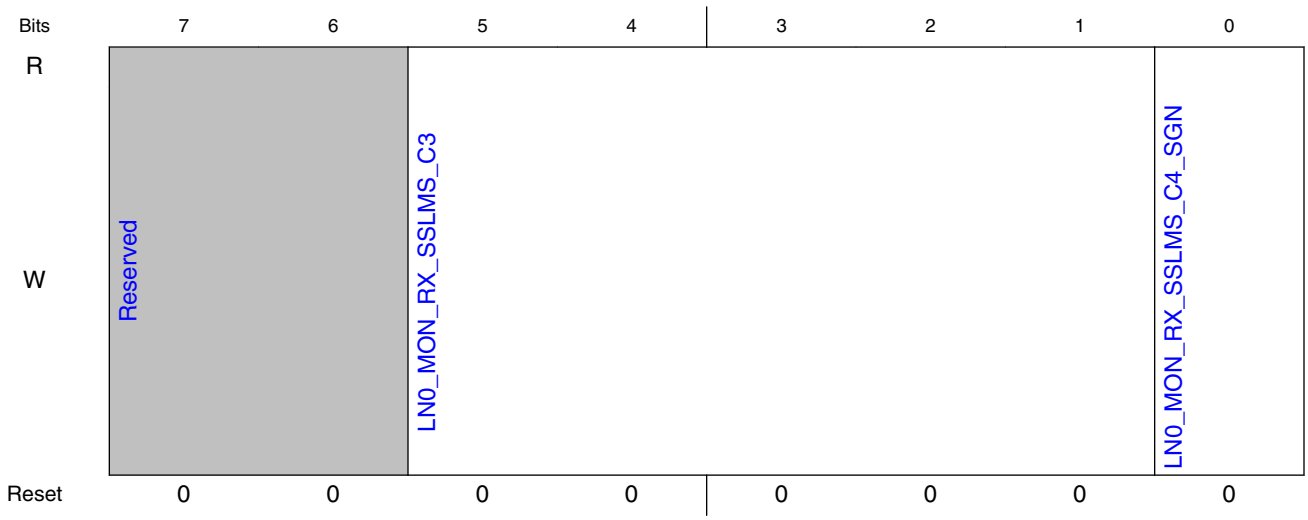
Field	Function
7-6 —	Reserved
5-1 LN0_MON_RX_SSLMS_C2	
0 LN0_MON_RX_SSLMS_C3_SGN	

11.5.3.1.367 (TRSV_REG0F0)

11.5.3.1.367.1 Offset

Register	Offset
TRSV_REG0F0	7C0h

11.5.3.1.367.2 Diagram



11.5.3.1.367.3 Fields

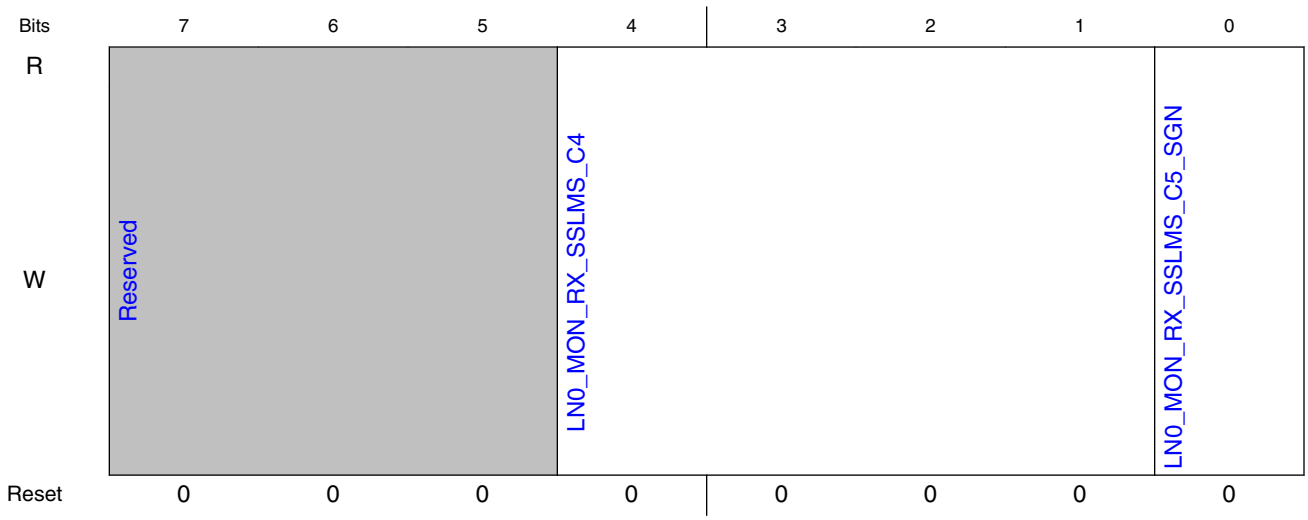
Field	Function
7-6 —	Reserved
5-1 LN0_MON_RX_SSLMS_C3	
0 LN0_MON_RX_SSLMS_C4_SGN	

11.5.3.1.368 (TRSV_REG0F1)

11.5.3.1.368.1 Offset

Register	Offset
TRSV_REG0F1	7C4h

11.5.3.1.368.2 Diagram



11.5.3.1.368.3 Fields

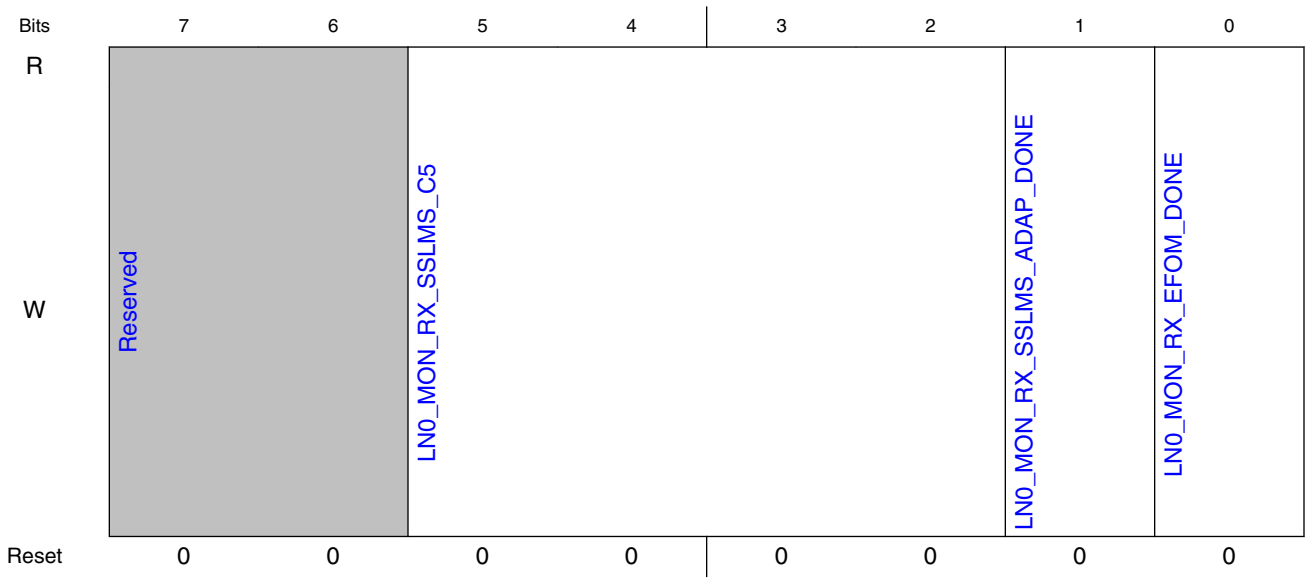
Field	Function
7-5 —	Reserved
4-1 LN0_MON_RX_SSLMS_C4	
0 LN0_MON_RX_SSLMS_C5_SGN	

11.5.3.1.369 (TRSV_REG0F2)

11.5.3.1.369.1 Offset

Register	Offset
TRSV_REG0F2	7C8h

11.5.3.1.369.2 Diagram



11.5.3.1.369.3 Fields

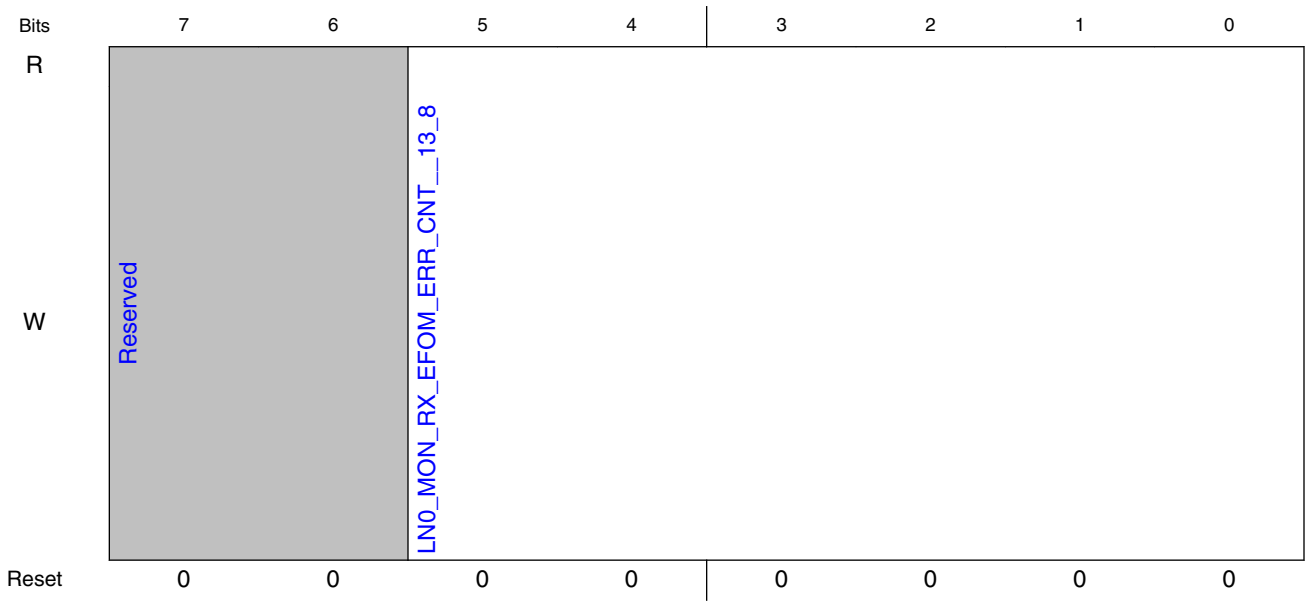
Field	Function
7-6 —	Reserved
5-2 LN0_MON_RX_SSLMS_C5	
1 LN0_MON_RX_SSLMS_ADAP_DONE	
0 LN0_MON_RX_EFOM_DONE	

11.5.3.1.370 (TRSV_REG0F3)

11.5.3.1.370.1 Offset

Register	Offset
TRSV_REG0F3	7CCh

11.5.3.1.370.2 Diagram



11.5.3.1.370.3 Fields

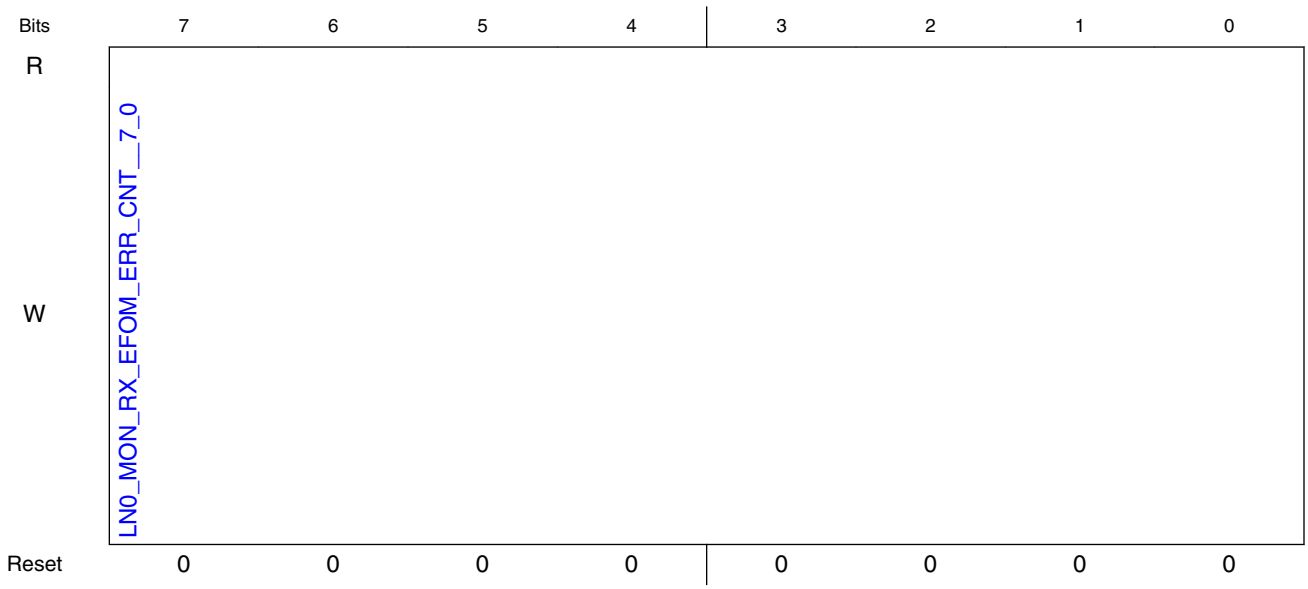
Field	Function
7-6 —	Reserved
5-0 LN0_MON_RX_EFOM_ERR_CNT__13_8	

11.5.3.1.371 (TRSV_REG0F4)

11.5.3.1.371.1 Offset

Register	Offset
TRSV_REG0F4	7D0h

11.5.3.1.371.2 Diagram



11.5.3.1.371.3 Fields

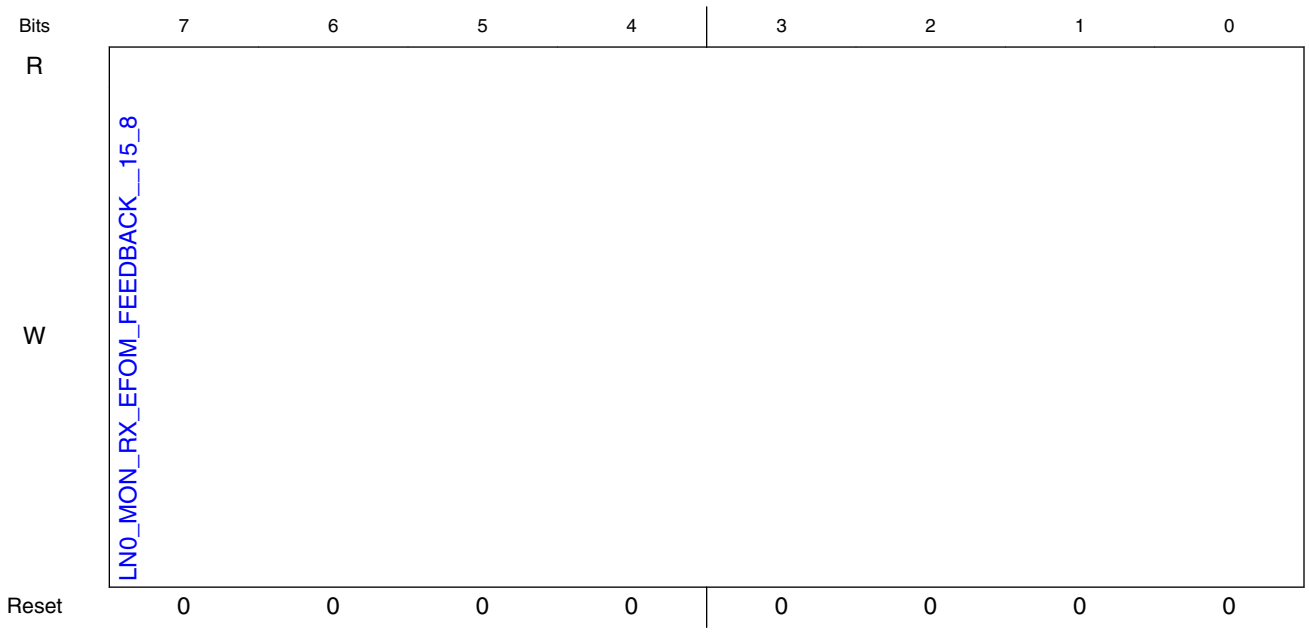
Field	Function
7-0 LN0_MON_RX_EFOM_ERR_CNT__7_0	

11.5.3.1.372 (TRSV_REG0F5)

11.5.3.1.372.1 Offset

Register	Offset
TRSV_REG0F5	7D4h

11.5.3.1.372.2 Diagram



11.5.3.1.372.3 Fields

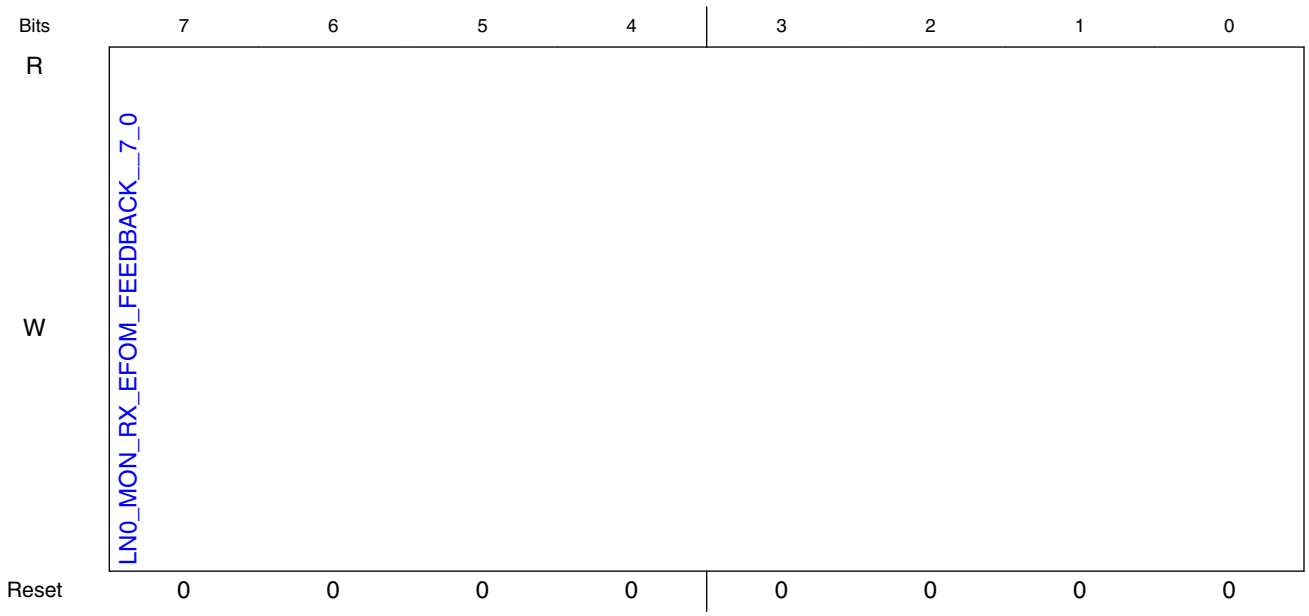
Field	Function
7-0 LN0_MON_RX_EFOM_FEEDBACK__15_8	

11.5.3.1.373 (TRSV_REG0F6)

11.5.3.1.373.1 Offset

Register	Offset
TRSV_REG0F6	7D8h

11.5.3.1.373.2 Diagram



11.5.3.1.373.3 Fields

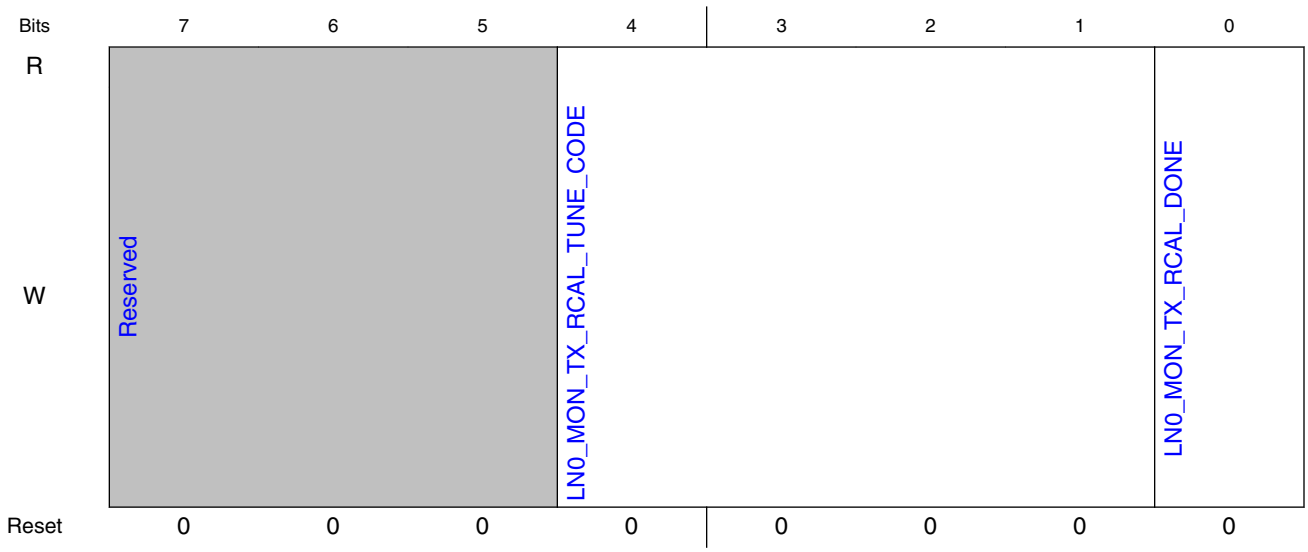
Field	Function
7-0 LN0_MON_RX_EFOM_FEEDBACK__7_0	

11.5.3.1.374 (TRSV_REG0F7)

11.5.3.1.374.1 Offset

Register	Offset
TRSV_REG0F7	7DCh

11.5.3.1.374.2 Diagram



11.5.3.1.374.3 Fields

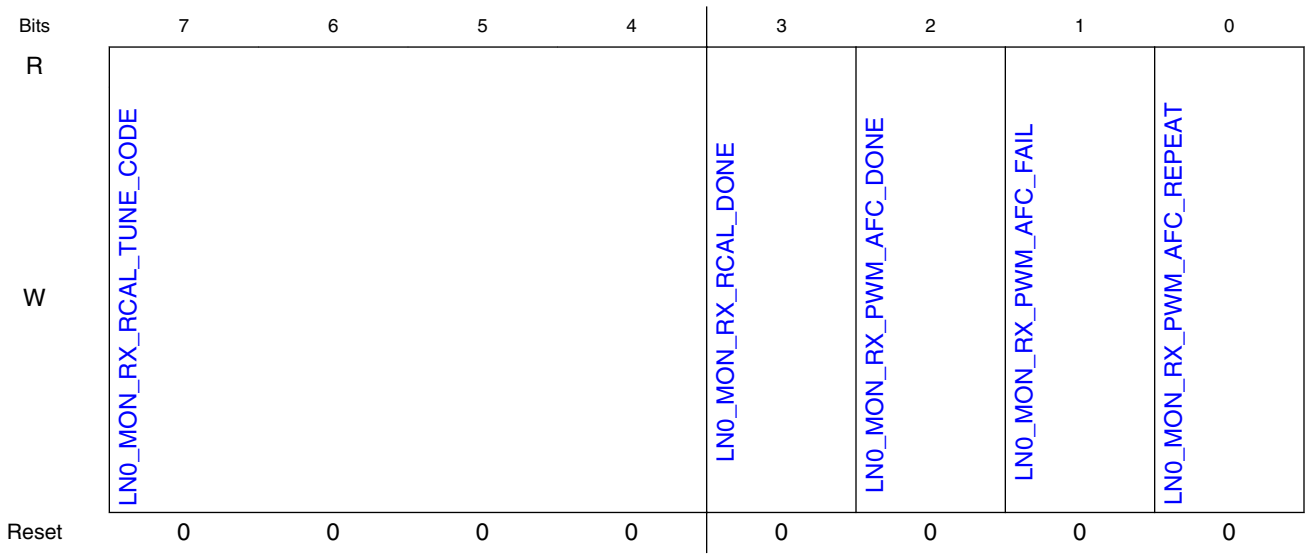
Field	Function
7-5 —	Reserved
4-1 LN0_MON_TX_RCAL_TUNE_CODE	
0 LN0_MON_TX_RCAL_DONE	

11.5.3.1.375 (TRSV_REG0F8)

11.5.3.1.375.1 Offset

Register	Offset
TRSV_REG0F8	7E0h

11.5.3.1.375.2 Diagram



11.5.3.1.375.3 Fields

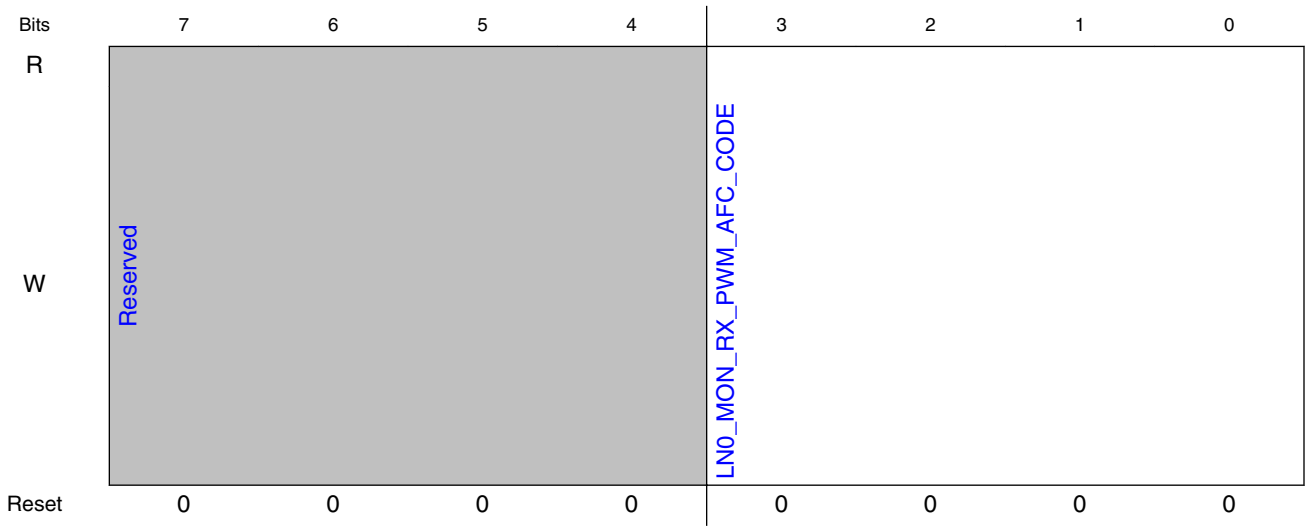
Field	Function
7-4 LN0_MON_RX_RCAL_TUNE_CODE	
3 LN0_MON_RX_RCAL_DONE	
2 LN0_MON_RX_PWM_AFC_DONE	
1 LN0_MON_RX_PWM_AFC_FAIL	
0 LN0_MON_RX_PWM_AFC_REPEAT	

11.5.3.1.376 (TRSV_REG0F9)

11.5.3.1.376.1 Offset

Register	Offset
TRSV_REG0F9	7E4h

11.5.3.1.376.2 Diagram



11.5.3.1.376.3 Fields

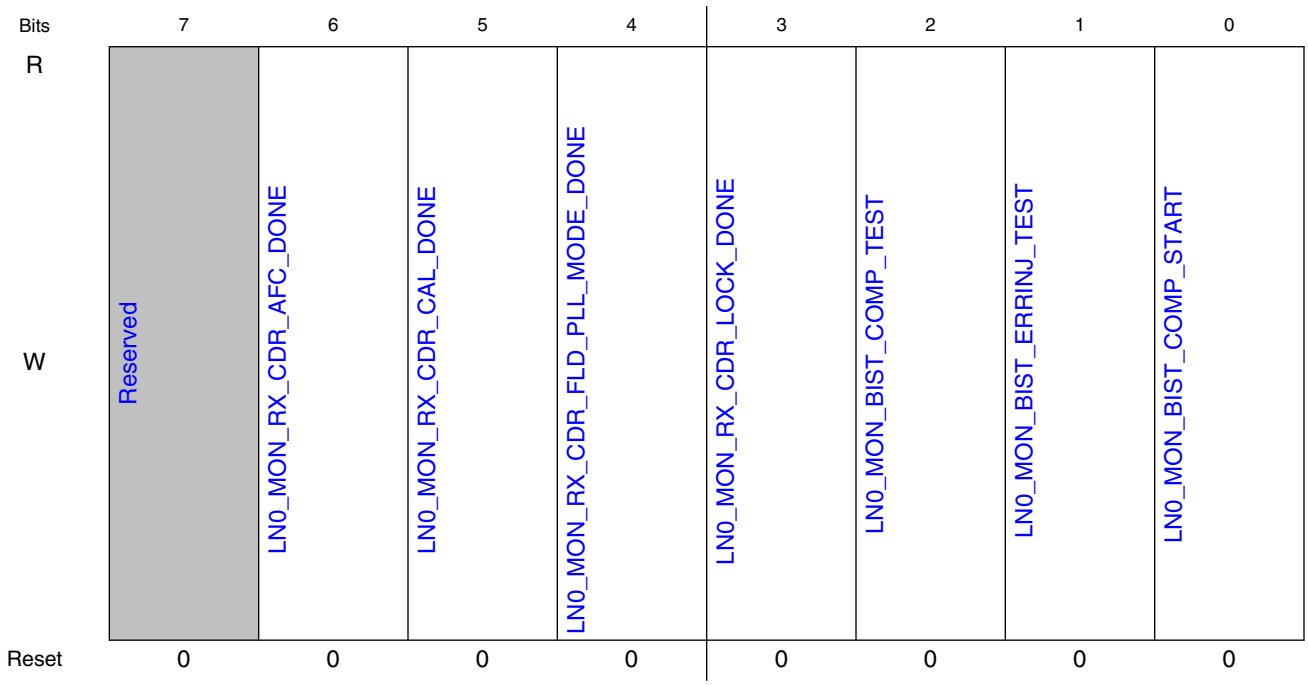
Field	Function
7-4 —	Reserved
3-0 LN0_MON_RX_PWM_AFC_CODE	

11.5.3.1.377 (TRSV_REG0FA)

11.5.3.1.377.1 Offset

Register	Offset
TRSV_REG0FA	7E8h

11.5.3.1.377.2 Diagram



11.5.3.1.377.3 Fields

Field	Function
7 —	Reserved
6 LN0_MON_RX_CDR_AFC_DONE	
5 LN0_MON_RX_CDR_CAL_DONE	
4 LN0_MON_RX_CDR_FLD_PLL_MODE_DONE	
3 LN0_MON_RX_CDR_LOCK_DONE	
2	

Table continues on the next page...

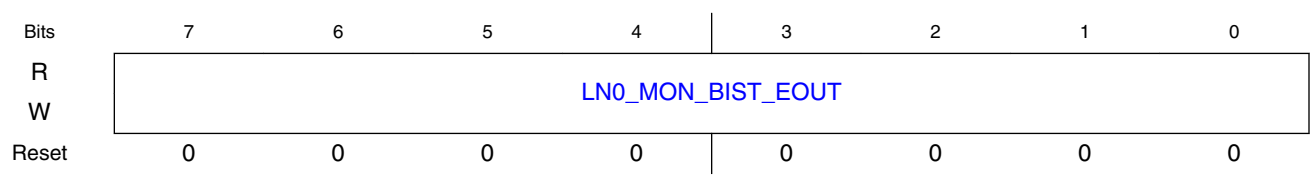
Field	Function
LN0_MON_BIST_COMP_TEST	
1 LN0_MON_BIST_ERRINJ_TEST	
0 LN0_MON_BIST_COMP_START	

11.5.3.1.378 (TRSV_REG0FB)

11.5.3.1.378.1 Offset

Register	Offset
TRSV_REG0FB	7ECh

11.5.3.1.378.2 Diagram



11.5.3.1.378.3 Fields

Field	Function
7-0 LN0_MON_BIST_EOUT	

11.5.3.1.379 (TRSV_REG0FC)

11.5.3.1.379.1 Offset

Register	Offset
TRSV_REG0FC	7F0h

11.5.3.1.379.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	LN0_ANA_TX_DRV_ACCDRV_POL_SEL_G1	LN0_ANA_TX_DRV_ACCDRV_CTRL_G1			LN0_ANA_TX_DRV_ACCDRV_POL_SEL_G2	LN0_ANA_TX_DRV_ACCDRV_CTRL_G2		
W								
Reset	0	0	0	0	0	0	0	0

11.5.3.1.379.3 Fields

Field	Function
7 LN0_ANA_TX_DRV_ACCDRV_POL_SEL_G1	
6-4 LN0_ANA_TX_DRV_ACCDRV_CTRL_G1	
3 LN0_ANA_TX_DRV_ACCDRV_POL_SEL_G2	
2-0	

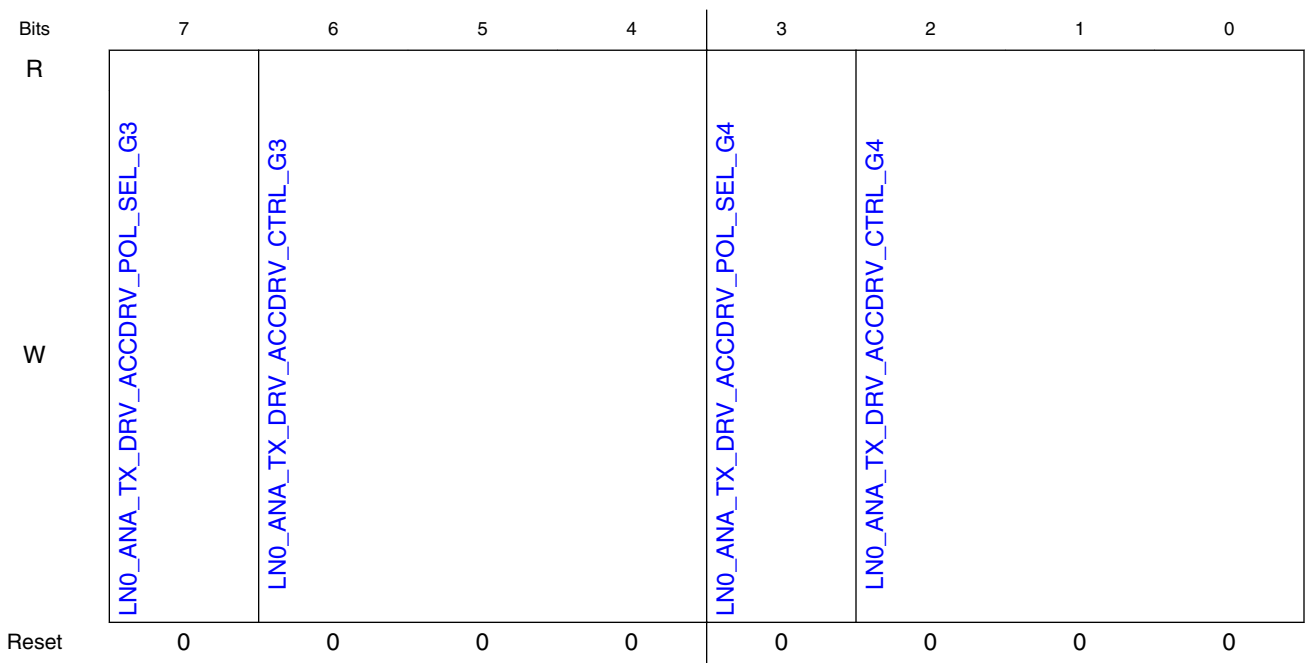
Field	Function
LN0_ANA_TX_DRV_ACCDRV_CTRL_G2	

11.5.3.1.380 (TRSV_REG0FD)

11.5.3.1.380.1 Offset

Register	Offset
TRSV_REG0FD	7F4h

11.5.3.1.380.2 Diagram



11.5.3.1.380.3 Fields

Field	Function
7 LN0_ANA_TX_DRV_ACCDRV_POL_SEL_G3	

Table continues on the next page...

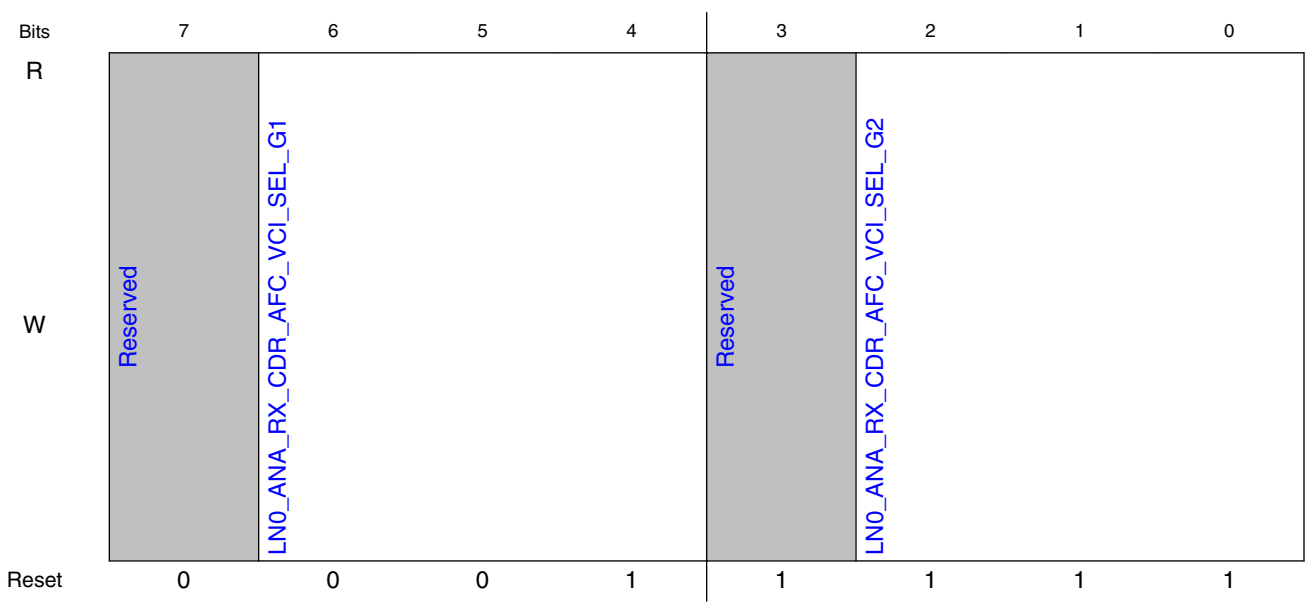
Field	Function
6-4 LN0_ANA_TX_DRV_ACCDRV_CTRL_G3	
3 LN0_ANA_TX_DRV_ACCDRV_POL_SEL_G4	
2-0 LN0_ANA_TX_DRV_ACCDRV_CTRL_G4	

11.5.3.1.381 (TRSV_REG0FE)

11.5.3.1.381.1 Offset

Register	Offset
TRSV_REG0FE	7F8h

11.5.3.1.381.2 Diagram



11.5.3.1.381.3 Fields

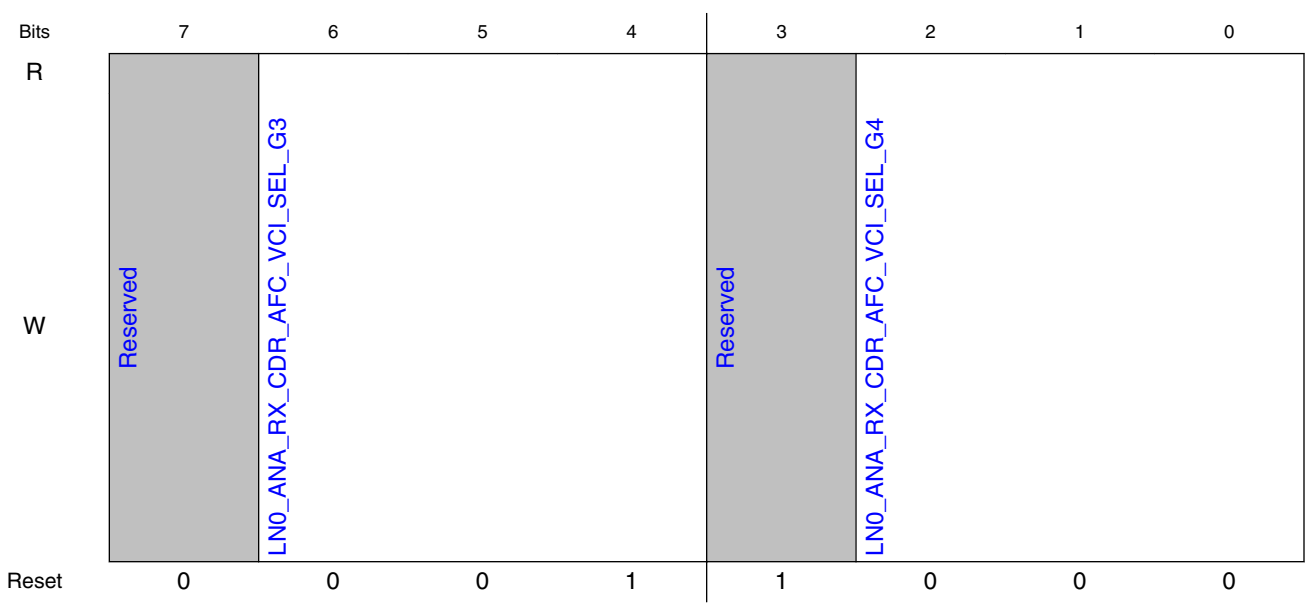
Field	Function
7 — 6-4 LN0_ANA_RX_CDR_AFC_VCI_SEL_G1	Reserved
3 — 2-0 LN0_ANA_RX_CDR_AFC_VCI_SEL_G2	Reserved

11.5.3.1.382 (TRSV_REG0FF)

11.5.3.1.382.1 Offset

Register	Offset
TRSV_REG0FF	7FCh

11.5.3.1.382.2 Diagram



11.5.3.1.382.3 Fields

Field	Function
7 —	Reserved
6-4 LN0_ANA_RX_CDR_AFC_VCI_SEL_G3	
3 —	Reserved
2-0 LN0_ANA_RX_CDR_AFC_VCI_SEL_G4	

11.6 Ethernet MAC (ENET)

11.6.1 Overview

The core implements a triple-speed 10/100/1000-Mbit/s Ethernet MAC compliant with the IEEE802.3-2002 standard. The MAC layer provides compatibility with half- or full-duplex 10/100-Mbit/s and full-duplex gigabit Ethernet LANs.

The MAC operation is fully programmable and can be used in Network Interface Card (NIC), bridging, or switching applications. The core implements the remote network monitoring (RMON) counters according to IETF RFC 2819.

The core also implements a hardware acceleration block to optimize the performance of network controllers providing TCP/IP, UDP, and ICMP protocol services. The acceleration block performs critical functions in hardware, which are typically implemented with large software overhead.

The core implements programmable embedded FIFOs that can provide buffering on the receive path for lossless flow control.

Advanced power management features are available with magic packet detection and programmable power-down modes.

A unified DMA (uDMA), internal to the ENET module, optimizes data transfer between the ENET core and the SoC, and supports an enhanced buffer descriptor programming model to support IEEE 1588 functionality.

The programmable Ethernet MAC with IEEE 1588 integrates a standard IEEE 802.3 Ethernet MAC with a time-stamping module. The IEEE 1588 standard provides accurate clock synchronization for distributed control nodes for industrial automation applications.

11.6.1.1 Block diagram

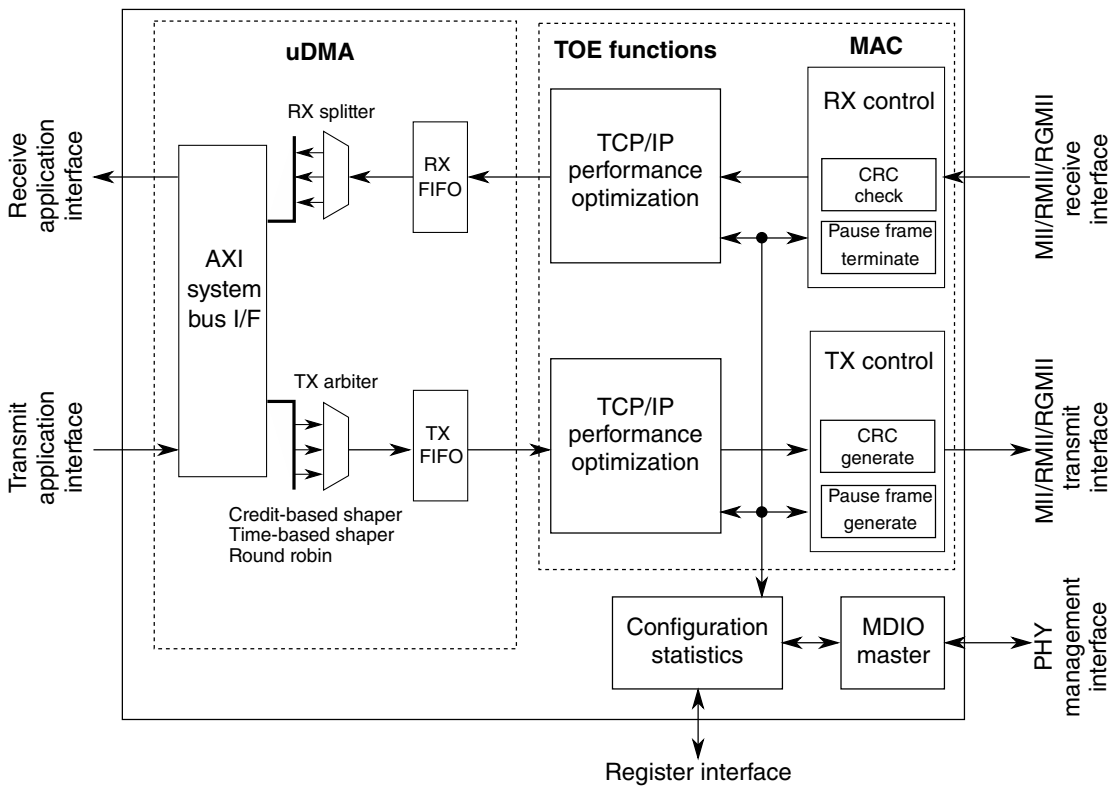


Figure 11-58. Ethernet MAC-NET core block diagram

11.6.1.2 Features

The MAC-NET core includes the following features.

11.6.1.2.1 Ethernet MAC features

- Implements the full 802.3 specification with preamble/SFD generation, frame padding generation, CRC generation and checking
- Supports zero-length preamble
- Dynamically configurable to support 10/100-Mbit/s and gigabit operation

- Supports 10/100 Mbit/s full-duplex and configurable half-duplex operation
- Supports gigabit full-duplex operation
- Compliant with the AMD magic packet detection with interrupt for node remote power management
- Seamless interface to commercial ethernet PHY devices via one of the following:
  - a 2-bit Reduced MII (RMII) operating at 50 MHz.
  - a (double data rate) 4-bit Reduced GMII (RGMII) operating at 125 MHz.
- CRC-32 checking at full speed with optional forwarding of the frame check sequence (FCS) field to the client
- CRC-32 generation and append on transmit or forwarding of user application provided FCS selectable on a per-frame basis
- In full-duplex mode:
  - Implements automated pause frame (802.3 x31A) generation and termination, providing flow control without user application intervention
  - Pause quanta used to form pause frames — dynamically programmable
  - Pause frame generation additionally controllable by user application offering flexible traffic flow control
  - Optional forwarding of received pause frames to the user application
  - Implements standard flow-control mechanism
- In half-duplex mode: provides full collision support, including jamming, backoff, and automatic retransmission
- Supports VLAN-tagged frames according to IEEE 802.1Q
- Programmable MAC address: Insertion on transmit; discards frames with mismatching destination address on receive (except broadcast and pause frames)
- Programmable promiscuous mode support to omit MAC destination address checking on receive
- Multicast and unicast address filtering on receive based on 64-entry hash table, reducing higher layer processing load
- Programmable frame maximum length providing support for any standard or proprietary frame length
- Statistics indicators for frame traffic and errors (alignment, CRC, length) and pause frames providing for IEEE 802.3 basic and mandatory management information database (MIB) package and remote network monitoring (RFC 2819)
- Simple handshake user application FIFO interface with fully programmable depth and threshold levels
- Provides separate status word for each received frame on the user interface providing information such as frame length, frame type, VLAN tag, and error information
- MII internal loopback
- MDIO master interface for PHY device configuration and management supports two programmable MDIO base addresses, and standard (IEEE 802.3 Clause 22) and extended (Clause 45) MDIO frame formats

- Supports legacy FEC buffer descriptors
- Interrupt coalescing reduces the number of interrupts generated by the MAC, reducing CPU loading
- Traffic-shaping bandwidth distribution supports credit-based and round-robin-based policies. Either policy can be combined with time-based shaping.
- AVB (Audio Video Bridging, IEEE 802.1Qav) features:
  - Credit-based bandwidth distribution policy can be combined with time-based shaping
  - AVB endpoint talker and listener support
  - Support for arbitration between different priority traffic (for example, AVB class A, AVB class B, and non-AVB)

11.6.1.2.2 IP protocol performance optimization features

- Operates on TCP/IP and UDP/IP and ICMP/IP protocol data or IP header only
- Enables wire-speed processing
- Supports IPv4 and IPv6
- Transparent passing of frames of other types and protocols
- Supports VLAN tagged frames according to IEEE 802.1q with transparent forwarding of VLAN tag and control field
- Automatic IP-header and payload (protocol specific) checksum calculation and verification on receive
- Automatic IP-header and payload (protocol specific) checksum generation and automatic insertion on transmit configurable on a per-frame basis
- Supports IP and TCP, UDP, ICMP data for checksum generation and checking
- Supports full header options for IPv4 and TCP protocol headers
- Provides IPv6 support to datagrams with base header only — datagrams with extension headers are passed transparently unmodifed/unchecked
- Provides statistics information for received IP and protocol errors
- Configurable automatic discard of erroneous frames
- Configurable automatic host-to-network (RX) and network-to-host (TX) byte order conversion for IP and TCP/UDP/ICMP headers within the frame
- Configurable padding remove for short IP datagrams on receive

- Configurable Ethernet payload alignment to allow for 32-bit word-aligned header and payload processing
- Programmable store-and-forward operation with clock and rate decoupling FIFOs

11.6.1.2.3 IEEE 1588 features

- Supports all IEEE 1588 frames.
- Allows reference clock to be chosen independently of network speed.
- Software-programmable precise time-stamping of ingress and egress frames
- Timer monitoring capabilities for system calibration and timing accuracy management
- Precise time-stamping of external events with programmable interrupt generation
- Programmable event and interrupt generation for external system control
- Supports hardware- and software-controllable timer synchronization.
- Provides a 4-channel IEEE 1588 timer. Each channel supports input capture and output compare using the 1588 counter.

11.6.2 Functional description

This section provides a complete functional description of the MAC-NET core.

11.6.2.1 Ethernet MAC frame formats

- Minimum length of 64 bytes
- Maximum length of 1518 bytes excluding the preamble and the start frame delimiter (SFD) bytes

An Ethernet frame consists of the following fields:

- Seven bytes preamble
- Start frame delimiter (SFD)
- Two address fields
- Length or type field

- Data field
- Frame check sequence (CRC value)

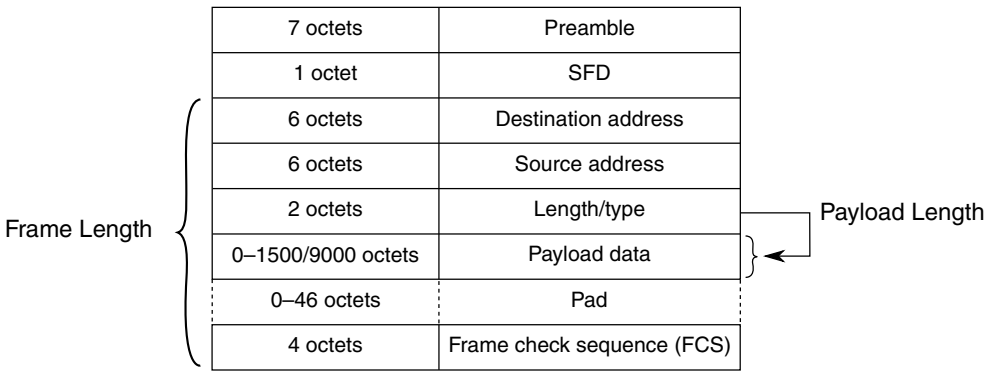


Figure 11-59. MAC frame format overview

Optionally, MAC frames can be VLAN-tagged with an additional four-byte field inserted between the MAC source address and the type/length field. VLAN tagging is defined by the IEEE P802.1q specification. VLAN-tagged frames have a maximum length of 1522 bytes, excluding the preamble and the SFD bytes.

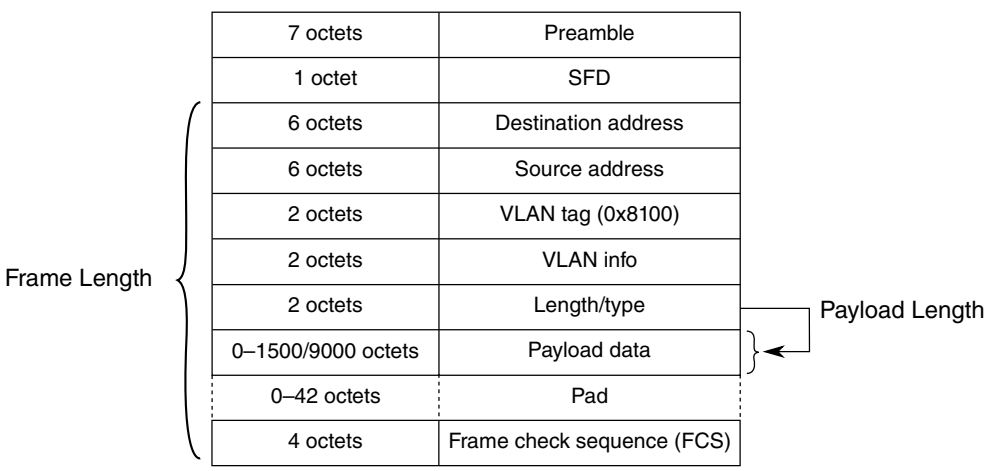


Figure 11-60. VLAN-tagged MAC frame format overview

Table 11-48. MAC frame definition

Term	Description
Frame length	Defines the length, in octets, of the complete frame without preamble and SFD. A frame has a valid length if it contains at least 64 octets and does not exceed the programmed maximum length.
Payload length	The length/type field indicates the length of the frame's payload section. The most significant byte is sent/received first.

Table continues on the next page...

Table 11-48. MAC frame definition (continued)

Term	Description
	• If the length/type field is set to a value less than 46, the payload is padded so that the minimum frame length requirement (64 bytes) is met. For VLAN-tagged frames, a value less than 42 indicates a padded frame. • If the length/type field is set to a value larger than the programmed frame maximum length (e.g. 1518) it is interpreted as a type field.
Destination and source address	48-bit MAC addresses. The least significant byte is sent/received first and the first two least significant bits of the MAC address distinguish MAC frames, as detailed in MAC address check.

Note

Although the IEEE specification defines a maximum frame length, the MAC core provides the flexibility to program any value for the frame maximum length.

11.6.2.1.1 Pause Frames

The receiving device generates a pause frame to indicate a congestion to the emitting device, which should stop sending data.

Pause frames are indicated by the length/type set to 0x8808. The two first bytes of a pause frame following the type, defines a 16-bit opcode field set to 0x0001 always. A 16-bit pause quanta is defined in the frame payload bytes 2 (P1) and 3 (P2) as defined in the following table. The P1 pause quanta byte is the most significant.

Table 11-49. Pause Frame Format (Values in Hex)

1	2	3	4	5	6	7	8	9	10	11	12	13	14
55	55	55	55	55	55	55	D5	01	80	C2	00	00	01
Preamble							SFD	Multicast Destination Address					
15	16	17	18	19	20	21	22	23	24	25	26	27 –68	
00	00	00	00	00	00	88	08	00	01	hi	lo	00	
Source Address						Type		Opcode		P1	P2	pad (42)	
69	70	71	72										
26	6B	AE	0A										
CRC-32													

There is no payload length field found within a pause frame and a pause frame is always padded with 42 bytes (0x00).

If a pause frame with a pause value greater than zero (XOFF condition) is received, the MAC stops transmitting data as soon the current frame transfer is completed. The MAC stops transmitting data for the value defined in pause quanta. One pause quanta fraction refers to 512 bit times.

If a pause frame with a pause value of zero (XON condition) is received, the transmitter is allowed to send data immediately (see Full-duplex flow control operation for details).

11.6.2.1.2 Magic packets

A magic packet is a unicast, multicast, or broadcast packet, which carries a defined sequence in the payload section.

Magic packets are received and inspected only under specific conditions as described in Magic packet detection.

The defined sequence to decode a magic packet is formed with a synchronization stream which consists of six consecutive 0xFF bytes, and is followed by sequence of sixteen consecutive unicast MAC addresses of the node to be awakened.

This sequence can be located anywhere in the magic packet payload. The magic packet is formed with a standard Ethernet header, optional padding, and CRC.

11.6.2.2 IP and higher layers frame format

The following sections use the term datagram to describe the protocol specific data unit that is found within the payload section of its container entity.

For example, an IP datagram specifies the payload section of an Ethernet frame. A TCP datagram specifies the payload section within an IP datagram.

11.6.2.2.1 Ethernet types

IP datagrams are carried in the payload section of an Ethernet frame. The Ethernet frame type/length field discriminates several datagram types.

The following table lists the types of interest:

Table 11-50. Ethernet type value examples

Type	Description
0x8100	VLAN-tagged frame. The actual type is found 4 octets later in the frame.
0x0800	IPv4
0x0806	ARP
0x86DD	IPv6

11.6.2.2.2 IPv4 datagram format

The following figure shows the IP Version 4 (IPv4) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words. The first byte sent/received is the leftmost byte of the first word (in other words, version/IHL field).

The IP header can contain further options, which are always padded if necessary to guarantee the payload following the header is aligned to a 32-bit boundary.

The IP header is immediately followed by the payload, which can contain further protocol headers (for example, TCP or UDP, as indicated by the protocol field value). The complete IP datagram is transported in the payload section of an Ethernet frame.

Table 11-51. IPv4 header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Version	IHL	TOS		Length			
Fragment ID				Flags	Fragment offset		
TTL		Protocol		Header checksum			
Source address							
Destination address							
Options							

Table 11-52. IPv4 header fields

Field name	Description
Version	4-bit IP version information. 0x4 for IPv4 frames.
IHL	4-bit Internet header length information. Determines number of 32-bit words found within the IP header. If no options are present, the default value is 0x5.
TOS	Type of service/DiffServ field.
Length	Total length of the datagram in bytes, including all octets of header and payload.
Fragment ID, flags, fragment offset	Fields used for IP fragmentation.
TTL	Time-to-live. In effect, is decremented at each router arrival. If zero, datagram must be discarded.
Protocol	Identifier of protocol that follows in the datagram.
Header checksum	Checksum of IP header. For computational purposes, this field's value is zero.
Source address	Source IP address.
Destination address	Destination IP address.

11.6.2.2.3 IPv6 datagram format

The following figure shows the IP version 6 (IPv6) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words and has a fixed length of ten words (40 bytes). The next header field identifies the type of the header that follows the IPv6 header. It is defined similar to the protocol identifier within IPv4, with new definitions for identifying extension headers. These headers can be inserted between the IPv6 header and the protocol header, which will shift the protocol header accordingly.The accelerator currently only supports IPv6 without extension headers (in other words, the next header specifies TCP, UDP, or IMCP).

The first byte sent/received is the leftmost byte of the first word (in other words, version/ traffic class fields).

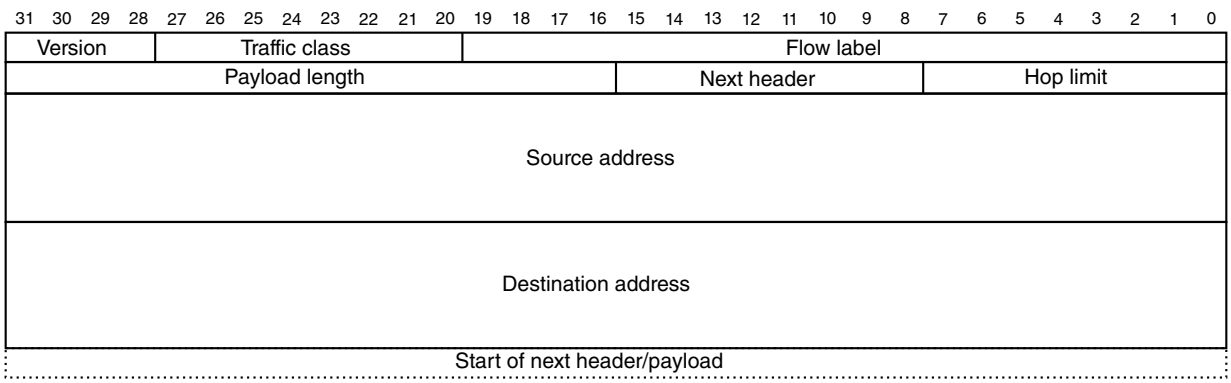


Figure 11-61. IPv6 header format

Table 11-53. IPv6 header fields

Field name	Description
Version	4-bit IP version information. 0x6 for all IPv6 frames.
Traffic class	8-bit field defining the traffic class.
Flow label	20-bit flow label identifying frames of the same flow.
Payload length	16-bit length of the datagram payload in bytes. It includes all octets following the IPv6 header.
Next header	Identifies the header that follows the IPv6 header. This can be the protocol header or any IPv6 defined extension header.
Hop limit	Hop counter, decremented by one by each station that forwards the frame. If hop limit is 0 the frame must be discarded.
Source address	128-bit IPv6 source address.
Destination address	128-bit IPv6 destination address.

11.6.2.2.4 Internet Control Message Protocol (ICMP) datagram format

An internet control message protocol (ICMP) is found following the IP header, if the protocol identifier is 1. The ICMP datagram has a four-octet header followed by additional message data.

Table 11-54. ICMP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Type		Code		Checksum			
ICMP message data							

Table 11-55. IP header fields

Field name	Description
Type	8-bit type information
Code	8-bit code that is related to the message type
Checksum	16-bit one's complement checksum over the complete ICMP datagram

11.6.2.2.5 User Datagram Protocol (UDP) datagram format

A user datagram protocol header is found after the IP header, when the protocol identifier is 17.

The payload of the datagram is after the UDP header. The header byte order follows the conventions given for the IP header above.

Table 11-56. UDP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source port				Destination port			
Length				Checksum			

Table 11-57. UDP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Length	Length of user data which immediately follows the header, including the UDP header (that is, minimum value is 8)
Checksum	Checksum over the complete datagram and some IP header information

11.6.2.2.6 TCP datagram format

A TCP header is found following the IP header, when the protocol identifier has a value of 6.

The TCP payload immediately follows the TCP header.

Table 11-58. TCP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source port				Destination port			
Sequence number							
Acknowledgement number							
Offset	Reserved		Flags	Window			
Checksum				Urgent pointer			
Options							

Table 11-59. TCP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Sequence number	Transmit sequence number
Ack. number	Receive sequence number
Offset	Data offset, which is number of 32-bit words within TCP header — if no options selected, defaults to value of 5
Flags	URG, ACK, PSH, RST, SYN, FIN flags
Window	TCP receive window size information
Checksum	Checksum over the complete datagram (TCP header and data) and IP header information
Options	Additional 32-bit words for protocol options

11.6.2.3 IEEE 1588 message formats

The following sections describe the IEEE 1588 message formats.

11.6.2.3.1 Transport encapsulation

The precision time protocol (PTP) datagrams are encapsulated in Ethernet frames using the UDP/IP transport mechanism, or optionally, with the newer 1588v2 directly in Ethernet frames (layer 2).

Typically, multicast addresses are used to allow efficient distribution of the synchronization messages.

11.6.2.3.1.1 UDP/IP

The 1588 messages (v1 and v2) can be transported using UDP/IP multicast messages.

The table below shows IP multicast groups defined for PTP. The table also shows their respective MAC layer multicast address mapping according to RFC 1112 (last three octets of IP follow the fixed value of 01-00-5E).

Table 11-60. UDP/IP multicast domains

Name	IP Address	MAC Address mapping
DefaultPTPdomain	224.0.1.129	01-00-5E-00-01-81
AlternatePTPdomain1	224.0.1.130	01-00-5E-00-01-82
AlternatePTPdomain2	224.0.1.131	01-00-5E-00-01-83
AlternatePTPdomain3	224.0.1.132	01-00-5E-00-01-84

Table 11-61. UDP port numbers

Message type	UDP port	Note
Event	319	Used for SYNC and DELAY_REQUEST messages
General	320	All other messages (for example, follow-up, delay-response)

11.6.2.3.1.2 Native Ethernet (PTPv2)

In addition to using UDP/IP frames, IEEE 1588v2 defines a native Ethernet frame format that uses ethertype = 0x88F7. The payload of the Ethernet frame immediately contains the PTP datagram, starting with the PTPv2 header.

Besides others, version 2 adds a peer delay mechanism to allow delay measurements between individual point-to-point links along a path over multiple nodes. The following multicast domains are also defined in PTPv2.

Table 11-62. PTPv2 multicast domains

Name	MAC address
Normal messages	01-1B-19-00-00-00
Peer delay messages	01-80-C2-00-00-0E

11.6.2.3.2 PTP header

All PTP frames contain a common header that determines the protocol version and the type of message, which defines the remaining content of the message.

All multi-octet fields are transmitted in big-endian order (the most significant byte is transmitted/received first).

The last four bits of versionPTP are at the same position (second byte) for PTPv1 and PTPv2 headers. This allows accurate identification by inspecting the first two bytes of the message.

11.6.2.3.2.1 PTPv1 header

Table 11-63. Common PTPv1 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	2	versionPTP = 0x0001							
2	2	versionNetwork							
4	16	subdomain							
20	1	messageType							
21	1	sourceCommunicationTechnology							
22	6	sourceUuid							
28	2	sourcePortId							
30	2	sequenceId							
32	1	control							
33	1	0x00							
34	2	flags							
36	4	reserved							

The type of message is encoded in the messageType and control fields as shown in Table 11-64 :

Table 11-64. PTPv1 message type identification

messageType	control	Message Name	Message
0x01	0x0	SYNC	Event message
0x01	0x1	DELAY_REQ	Event message
0x02	0x2	FOLLOW_UP	General message
0x02	0x3	DELAY_RESP	General message
0x02	0x4	MANAGEMENT	General message
other	other	—	Reserved

The field sequenceId is used to non-ambiguously identify a message.

11.6.2.3.2.2 PTPv2 header

Table 11-65. Common PTPv2 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	1	transportSpecific				messageId			
1	1	reserved				versionPTP = 0x2			
2	2	messageLength							
4	1	domainNumber							
5	1	reserved							
6	2	flags							
8	8	correctionField							
16	4	reserved							
20	10	sourcePortIdentity							
30	2	sequenceId							
32	1	control							
33	1	logMeanMessageInterval							

The type of message is encoded in the field messageId as follows:

Table 11-66. PTPv2 message type identification

messageId	Message name	Message
0x0	SYNC	Event message
0x1	DELAY_REQ	Event message
0x2	PATH_DELAY_REQ	Event message
0x3	PATH_DELAY_RESP	Event message
0x4–0x7	—	Reserved
0x8	FOLLOW_UP	General message
0x9	DELAY_RESP	General message
0xa	PATH_DELAY_FOLLOW_UP	General message
0xb	ANNOUNCE	General message
0xc	SIGNALING	General message
0xd	MANAGEMENT	General message

The PTPv2 flags field contains further details on the type of message, especially if one-step or two-step implementations are used. The one- or two-step implementation is controlled by the TWO_STEP bit in the first octet of the flags field as shown below. Reserved bits are cleared.

Table 11-67. PTPv2 message flags field definitions

Bit	Name	Description
0	ALTERNATE_MASTER	See IEEE 1588 Clause 17.4
1	TWO_STEP	1 Two-step clock 0 One-step clock
2	UNICAST	1 Transport layer address uses a unicast destination address 0 Multicast is used
3	—	Reserved
4	—	Reserved
5	Profile specific	
6	Profile specific	
7	—	Reserved

11.6.2.4 MAC receive

The MAC receive engine performs the following tasks:

- Check frame framing
- Remove frame preamble and frame SFD field
- Discard frame based on frame destination address field
- Terminate pause frames
- Check frame length
- Remove payload padding if it exists
- Calculate and verify CRC-32
- Write received frames in the core receive FIFO

If the MAC is programmed to operate in half-duplex mode, it will also check if the frame is received with a collision.

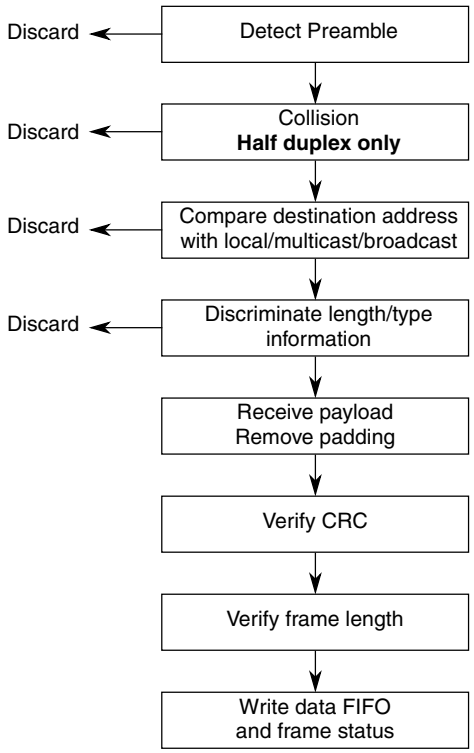


Figure 11-62. MAC receive flow

11.6.2.4.1 Collision detection in half-duplex mode

If the packet is received with a collision detected during reception of the first 64 bytes, the packet is discarded (if frame size was less than ~14 octets) or transmitted to the user application with an error and RxBD[CE] set.

11.6.2.4.2 Preamble processing

The IEEE 802.3 standard allows a maximum size of 56 bits (seven bytes) for the preamble, while the MAC core allows any preamble length, including zero length preamble.

The MAC core checks for the start frame delimiter (SFD) byte. If the next byte of the preamble, which is different from 0x55, is not 0xD5, the frame is discarded.

Although the IEEE specification dictates that the inner-packet gap should be at least 96 bits, the MAC core is designed to accept frames separated by only 64 10/100-Mbit/s operation (MII) bits.

The MAC core removes the preamble and SFD bytes.

11.6.2.4.3 MAC address check

The destination address bit 0 differentiates between multicast and unicast addresses.

- If bit 0 is 0, the MAC address is an individual (unicast) address.
- If bit 0 is 1, the MAC address defines a group (multicast) address.
- If all 48 bits of the MAC address are set, it indicates a broadcast address.

11.6.2.4.3.1 Unicast address check

If a unicast address is received, the destination MAC address is compared to the node MAC address programmed by the host in the PADDR1/2 registers.

If the destination address matches any of the programmed MAC addresses, the frame is accepted.

If Promiscuous mode is enabled (RCR[PROM] = 1) no address checking is performed and all unicast frames are accepted.

11.6.2.4.3.2 Multicast and unicast address resolution

The hash table algorithm used in the group and individual hash filtering operates as follows.

- The 48-bit destination address is mapped into one of 64 bits, represented by 64 bits in ENET*n*_GAUR/GALR (group address hash match) or ENET*n*_IAUR/IALR (individual address hash match).
- This mapping is performed by passing the 48-bit address through the on-chip 32-bit CRC generator and selecting the six most significant bits of the CRC-encoded result to generate a number between 0 and 63.
- The msb of the CRC result selects ENET*n*_GAUR (msb = 1) or ENET*n*_GALR (msb = 0).
- The five lsbs of the hash result select the bit within the selected register.
- If the CRC generator selects a bit set in the hash table, the frame is accepted; else, it is rejected.

For example, if eight group addresses are stored in the hash table and random group addresses are received, the hash table prevents roughly 56/64 (or 87.5%) of the group address frames from reaching memory. Those that do reach memory must be further filtered by the processor to determine if they truly contain one of the eight desired addresses.

The effectiveness of the hash table declines as the number of addresses increases.

The user must initialize the hash table registers. Use this CRC32 polynomial to compute the hash:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

If Promiscuous mode is enabled (ENET*n*_RCR[PROM] = 1) all unicast and multicast frames are accepted regardless of ENET*n*_GAUR/GALR and ENET*n*_IAUR/IALR settings.

11.6.2.4.3.3 Broadcast address reject

All broadcast frames are accepted if BC_REJ is cleared or ENET*n*_RCR[PROM] is set. If PROM is cleared when ENET*n*_RCR[BC_REJ] is set, all broadcast frames are rejected.

Table 11-68. Broadcast address reject programming

PROM	BC_REJ	Broadcast frames
0	0	Accepted
0	1	Rejected
1	0	Accepted
1	1	Accepted

11.6.2.4.3.4 Miss-bit implementation

For higher layer filtering purposes, RxBD[M] indicates an address miss when the MAC operates in promiscuous mode and accepts a frame that would otherwise be rejected.

If a group/individual hash or exact match does not occur and Promiscuous mode is enabled (RCR[PROM] = 1), the frame is accepted and the M bit is set in the buffer descriptor; otherwise, the frame is rejected.

This means the status bit is set in any of the following conditions during Promiscuous mode:

- A broadcast frame is received when BC_REJ is set
- A unicast is received that does not match either:

- Node address (PALR[PADDR1] and PAUR[PADDR2])
- Hash table for unicast (IAUR[IADDR1] and IALR[IADDR2])

- A multicast is received that does not match the GAUR[GADDR1] and GALR[GADDR2] hash table entries

11.6.2.4.4 Frame length/type verification: payload length check

If the length/type is less than 0x600 and NLC is set, the MAC checks the payload length and reports any error in the frame status word RCR[NLC] and interrupt bit EIR[PLR].

If the length/type is greater than or equal to 0x600, the MAC interprets the field as a type and no payload length check is performed.

The length check is performed on VLAN and stacked VLAN frames. If a padded frame is received, no length check can be performed due to the extended frame payload because padded frames can never have a payload length error.

11.6.2.4.5 Frame length/type verification: frame length check

When the receive frame length exceeds MAX_FL bytes, the BABR interrupt is generated and the RxBD[LG] bit is set.

The frame is not truncated unless the frame length exceeds the value programmed in ENET*n*_FTRL[TRUNC_FL]. If the frame is truncated, RxBD[TR] is set. In addition, a truncated frame always has the CRC error indication set (RxBD[CR]).

11.6.2.4.6 VLAN frames processing

VLAN frames have a length/type field set to 0x8100 immediately followed by a 16-Bit VLAN control information field.

VLAN-tagged frames are received as normal frames because the VLAN tag is not interpreted by the MAC function, and are pushed complete with the VLAN tag to the user application. If the length/type field of the VLAN-tagged frame, which is found four octets later in the frame, is less than 42, the padding is removed. In addition, the frame status word (RxBD[VLAN]) indicates that the current frame is VLAN tagged.

11.6.2.4.7 Pause frame termination

The receive engine terminates pause frames and does not transfer them to the receive FIFO. The quanta is extracted and sent to the MAC transmit path via a small internal clock rate decoupling asynchronous FIFO.

The quanta is written only if a correct CRC and frame length are detected by the control state machine. If not, the quanta is discarded and the MAC transmit path is not paused.

Valid pause frames are ignored if ENET*n*_RCR[FCE] is cleared and are forwarded to the client interface when ENET*n*_RCR[PAUFWD] is set.

11.6.2.4.8 CRC check

The CRC-32 field is checked and forwarded to the core FIFO interface if ENET*n*_RCR[CRCFWD] is cleared and ENET*n*_RCR[PADEN] is cleared. When CRCFWD is set (regardless of PADEN), the CRC-32 field is checked and terminated (not transmitted to the FIFO).

The CRC polynomial, as specified in the 802.3 standard, is:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

The 32 bits of the CRC value are placed in the frame check sequence (FCS) field with the x^{31} term as right-most bit of the first octet. The CRC bits are thus received in the following order: x^{31}, x^{30},..., x^{1}, x^{0}.

If a CRC error is detected, the frame is marked invalid and RxBD[CR] is set.

11.6.2.4.9 Frame padding removal

When a frame is received with a payload length field set to less than 46 (42 for VLAN-tagged frames and 38 for frames with stacked VLANs), the zero padding can be removed before the frame is written into the data FIFO depending on the setting of ENET*n*_RCR[PADEN].

Note

If a frame is received with excess padding (in other words, the length field is set as mentioned above, but the frame has more than 64 octets) and padding removal is enabled, then the padding is removed as normal and no error is reported if the frame is otherwise correct (for example: good CRC, less than maximum length, and no other error).

11.6.2.4.10 Frame classification (AVB)

To support protocols such as Audio Video Bridging (AVB, IEEE 802.1Qav), normal traffic is separated from time-sensitive traffic immediately at the application interface to allow storing them in different queues for further processing. For every frame received, a

classification based on VLAN priority can be performed. When a frame is received, and it contains a VLAN tag, the priority is compared against the values set in the classification match registers (see RCMR register and the following figure.

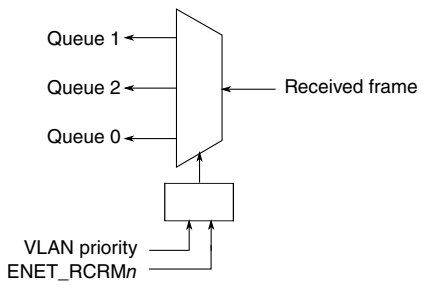


Figure 11-63. AVB frame classification

11.6.2.4.11 Receive flushing

RX flushing prevents frames in the RX FIFO from being blocked. Blocking can occur if the frame at the head of the RX FIFO cannot be forwarded because the ring it is associated with cannot accept it. This situation occurs when either the ring's RxBD[EMPTY] is not set or ENET_RDAR*n* is not set. When RX flushing is enabled, via QOS[RX_FLUSHn], the blocking frame will be flushed (discarded).

11.6.2.5 MAC transmit

Frame transmission starts when the transmit FIFO holds enough data.

After a transfer starts, the MAC transmit function performs the following tasks:

- Generates preamble and SFD field before frame transmission
- Generates XOFF pause frames if the receive FIFO reports a congestion or if ENET*n*_TCR[TFC_PAUSE] is set with ENET*n*_OPD[PAUSE_DUR] set to a non-zero value
- Generates XON pause frames if the receive FIFO congestion condition is cleared or if TFC_PAUSE is set with PAUSE_DUR cleared
- Suspends Ethernet frame transfer (XOFF) if a non-zero pause quanta is received from the MAC receive path
- Adds padding to the frame if required
- Calculates and appends CRC-32 to the transmitted frame
- Sends the frame with correct inter-packet gap (IPG) (deferring)

When the MAC is configured to operate in half-duplex mode, the following additional tasks are performed:

- Collision detection
- Frame retransmit after back-off timer expires

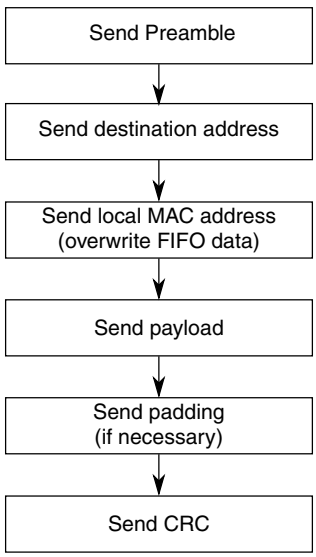


Figure 11-64. Frame transmit overview

11.6.2.5.1 Frame payload padding

The IEEE specification defines a minimum frame length of 64 bytes.

If the frame sent to the MAC from the user application has a size smaller than 60 bytes, the MAC automatically adds padding bytes (0x00) to comply with the Ethernet minimum frame length specification. Transmit padding is always performed and cannot be disabled.

If the MAC is not allowed to append a CRC (TxBD[TC] = 1), the user application is responsible for providing frames with a minimum length of 64 octets.

11.6.2.5.2 MAC address insertion

On each frame received from the core transmit FIFO interface, the source MAC address is either:

- Replaced by the address programmed in the PADDR1/2 fields (ENET*n*_TCR[ADDINS] = 1)
- Transparently forwarded to the Ethernet line (ENET*n*_TCR[ADDINS] = 0)

11.6.2.5.3 CRC-32 generation

The CRC-32 field is optionally generated and appended at the end of a frame.

The CRC polynomial, as specified in the 802.3 standard, is:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

The 32 bits of the CRC value are placed in the FCS field so that the x^{31} term is the right-most bit of the first octet. The CRC bits are thus transmitted in the following order: x^{31}, x^{30},..., x^{1}, x^{0}.

11.6.2.5.4 Inter-packet gap (IPG)

In full-duplex mode, after frame transmission and before transmission of a new frame, an IPG (programmed in ENET*n*_TIPG) is maintained. The minimum IPG can be programmed between 8 and 26 byte-times (64 and 208 bit-times).

In half-duplex mode, the core constantly monitors the line. Actual transmission of the data onto the network occurs only if it has been idle for a 96-bit time period, and any back-off time requirements have been satisfied. In accordance with the standard, the core begins to measure the IPG from CRS/GMII_CRS de-assertion.

11.6.2.5.5 Collision detection and handling — half-duplex operation only

A collision occurs on a half-duplex network when concurrent transmissions from two or more nodes take place. During transmission, the core monitors the line condition and detects a collision when the PHY device asserts COL.

When the core detects a collision while transmitting, it stops transmission of the data and transmits a 32-bit jam pattern. If the collision is detected during the preamble or the SFD transmission, the jam pattern is transmitted after completing the SFD, which results in a minimum 96-bit fragment. The jam pattern is a fixed pattern that is not compared to the actual frame CRC, and has a very low probability (0.532) of having a jam pattern identical to the CRC.

If a collision occurs before transmission of 64 bytes (including preamble and SFD), the MAC core waits for the backoff period and retransmits the packet data (stored in a 64-byte re-transmit buffer) that has already been sent on the line. The backoff period is generated from a pseudo-random process (truncated binary exponential backoff).

If a collision occurs after transmission of 64 bytes (including preamble and SFD), the MAC discards the remainder of the frame, optionally sets the LC interrupt bit, and sets TxBD[LCE].

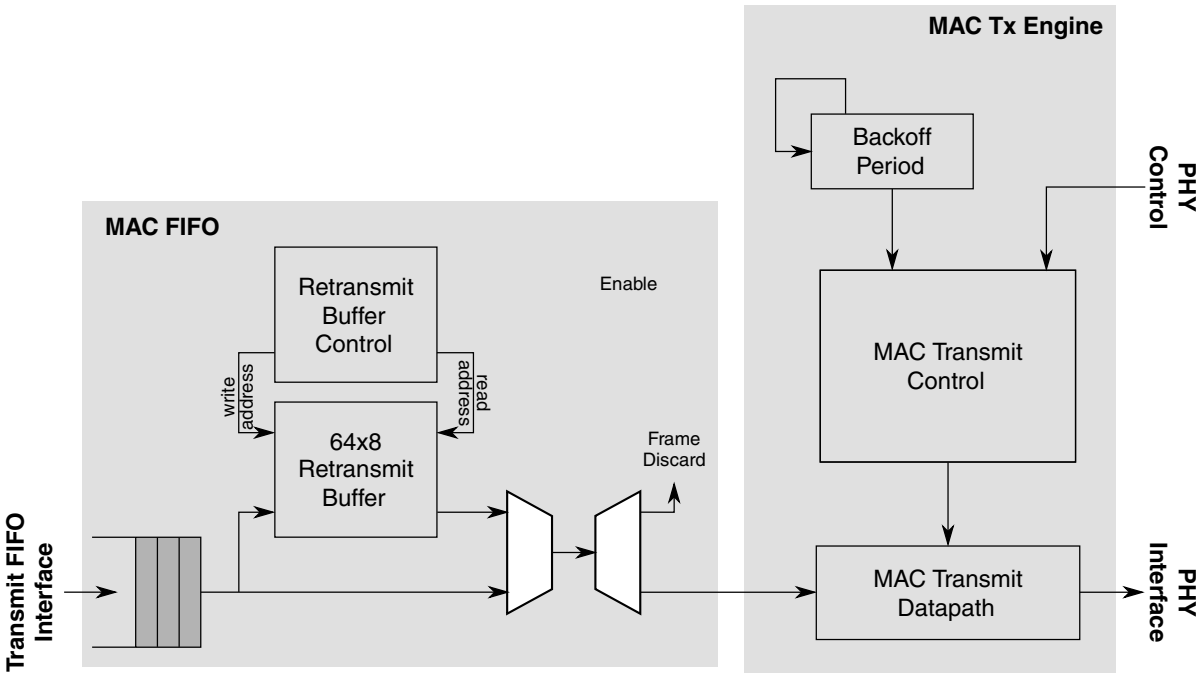


Figure 11-65. Packet re-transmit overview

The backoff time is represented by an integer multiple of slot times. One slot is equal to a 512-bit time period. The number of the delay slot times, before the n^{th} re-transmission attempt, is chosen as a uniformly-distributed random integer in the range:

- $0 < r < 2^k$
- $k = \min(n, N)$; where n is the number of retransmissions and $N = 10$

For example, after the first collision, the backoff period is 0 or 1 slot time. If a collision occurs on the first retransmission, the backoff period is 0, 1, 2, or 3, and so on.

The maximum backoff time (in 512-bit time slots) is limited by N = 10 as specified in the IEEE 802.3 standard.

If a collision occurs after 16 consecutive retransmissions, the core reports an excessive collision condition (ENET*n*_EIR[RL] interrupt field and TxBD[EE]) and discards the current packet from the FIFO.

In networks violating the standard requirements, a collision may occur after transmission of the first 64 bytes. In this case, the core stops the current packet transmission and discards the rest of the packet from the transmit FIFO. The core resumes transmission with the next packet available in the core transmit FIFO.

warning

Ethernet PHYs that support the SQE Test, or "heartbeat,"
feature must disable this feature. When this feature is enabled,

the PHY asserts the collision signal after a frame is transmitted to indicate to the ENET that the PHY's collision logic is working. This may cause data corruption in the next frame from the ENET. This corrupted frame contains up to 21 zero bytes which start somewhere within the MAC destination address field. The ENET, however, will still generate a good FCS (CRC-32) but with corrupted data.

11.6.2.5.6 Rate limiting / traffic shaping support

The MAC-NET supports two methods to optimize frame traffic for either time-sensitive AVB frames (Class A and Class B) or best-effort non-AVB frames:

- Round-robin scheme
- Credit-based traffic shaping

To ensure that sufficient bandwidth is allocated for the AVB frames, the credit-based shaper must be used. Either method can be combined with a time-based shaper to ensure that a frame is always transmitted in its correct time slot.

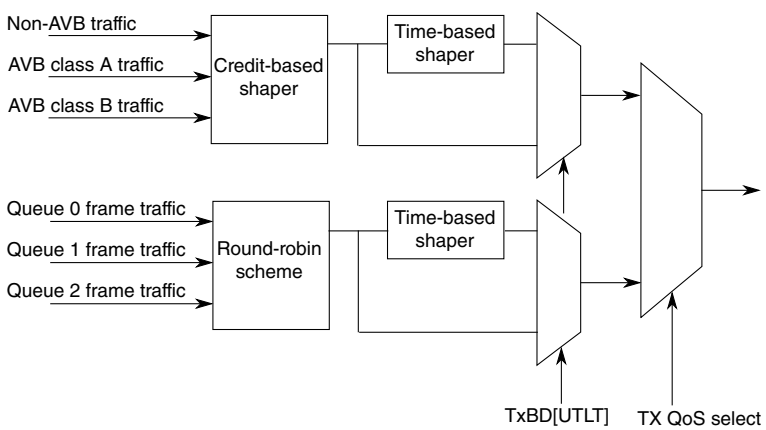


Figure 11-66. Transmit arbitration schemes

11.6.2.5.6.1 Round-robin policy

In the round-robin QoS scheme, each queue is given an equal opportunity to transmit one frame. For example, if queue *n* has a frame to transmit, the queue transmits its frame. After queue *n* has transmitted its frame, or if queue *n* does not have a frame to transmit, queue *n*+1 is then allowed to transmit its frame, and so on.

11.6.2.5.6.2 Credit-based shaper

The AVB credit based shaper acts independently, per class, to control the bandwidth distribution between normal traffic and time-sensitive traffic with respect to the total link bandwidth available. As per the IEEE 802.1Q, the maximum bandwidth distribution that can be allocated for the time-sensitive frames is 75%. The following example uses 70% for AVB frames and 30% for non-AVB frames:

Table 11-69. Bandwidth allocation example

AVB Class A	50%
AVB Class B	20%
non-AVB frames	30%
Total	100%

See DMACFG register for information on how to allocate bandwidth. The following figure shows how frame traffic is controlled within the MAC-NET core.

11.6.2.5.6.3 Time-based shaper

The time-based shaper enables the user to specify when a frame can be transmitted. It is always used in combination with either the round-robin scheme or the credit-based shaper. Use of the time-based shaper can ensure that frames are always transmitted in the correct time slot.

11.6.2.5.6.3.1 Time-based shaper example

Time-based shaping involves the following procedure:

1. Read the current value of the timer. See ATVR register and ATCR register.
2. Calculate the frame launch time and write this value to the TLT field of the frame's TxBD. See Enhanced transmit buffer descriptor.
3. Instruct the ENET to use the TLT by setting TxBD[UTLT].

The frame will be fetched and transmitted only if TxBD[TLT] is less than the current value of the timer. In other words, the timer must be past, that is, be greater than, the transmit launch time before the frame will be fetched and transmitted.

The transmit launch time must be not be greater than the current value of ENET_ATVR + (0.5 × ENET_ATPER). This means the application can not prepare frames with launch times beyond ENET_ATVR + (0.5 × ENET_ATPER). Because ENET_ATVR wraps, calculate TLT using the equation

$$(ENET_ATVR + (0.5 \times ENET_ATPER)) \bmod ENET_ATPER$$

For example, if ENET_ATPER is set to the recommended value of 1,000,000,000 ns (one second), then the user must not prepare future frames with launch times greater than (500,000,000 + ENET_ATVR).

As a specific example, if the current value of ENET_ATVR is 100,000, then you can prepare the following frames:

1. Frame 1 TxBD[TLT] = 100,000ns + 125,000ns
2. Frame 2 TxBD[TLT] = 100,000ns + 250,000ns
3. And so on

11.6.2.6 Full-duplex flow control operation

Three conditions are handled by the core's flow control engine:

- Remote device congestion — The remote device connected to the same Ethernet segment as the core reports congestion and requests that the core stop sending data.
- Core FIFO congestion — When the core's receive FIFO reaches a user-programmable threshold (RX section empty), the core sends a pause frame back to the remote device requesting the data transfer to stop.
- Local device congestion — Any device connected to the core can request (typically, via the host processor) the remote device to stop transmitting data.

11.6.2.6.1 Remote device congestion

When the MAC transmit control gets a valid pause quanta from the receive path and if ENET*n*_RCR[FCE] is set, the MAC transmit logic:

- Completes the transfer of the current frame.
- Stops sending data for the amount of time specified by the pause quanta in 512 bit time increments.
- Sets ENET*n*_TCR[RFC_PAUSE].

Frame transfer resumes when the time specified by the quanta expires and if no new quanta value is received, or if a new pause frame with a quanta value set to 0x0000 is received. The MAC also resets RFC_PAUSE to zero.

If ENET*n*_RCR[FCE] cleared, the MAC ignores received pause frames.

Optionally and independent of ENET*n*_RCR[FCE], pause frames are forwarded to the client interface if PAUFWD is set.

11.6.2.6.2 Local device/FIFO congestion

The MAC transmit engine generates pause frames when the local receive FIFO is not able to receive more than a pre-defined number of words (FIFO programmable threshold) or when pause frame generation is requested by the local host processor.

- To generate a pause frame, the host processor sets ENET*n*_TCR[TFC_PAUSE]. A single pause frame is generated when the current frame transfer is completed and TFC_PAUSE is automatically cleared. Optionally, an interrupt is generated.
- An XOFF pause frame is generated when the receive FIFO asserts its section empty flag (internal). An XOFF pause frame is generated automatically, when the current frame transfer completes.
- An XON pause frame is generated when the receive FIFO deasserts its section empty flag (internal). An XON pause frame is generated automatically, when the current frame transfer completes.

When an XOFF pause frame is generated, the pause quanta (payload byte P1 and P2) is filled with the value programmed in ENET*n*_OPD[PAUSE_DUR].

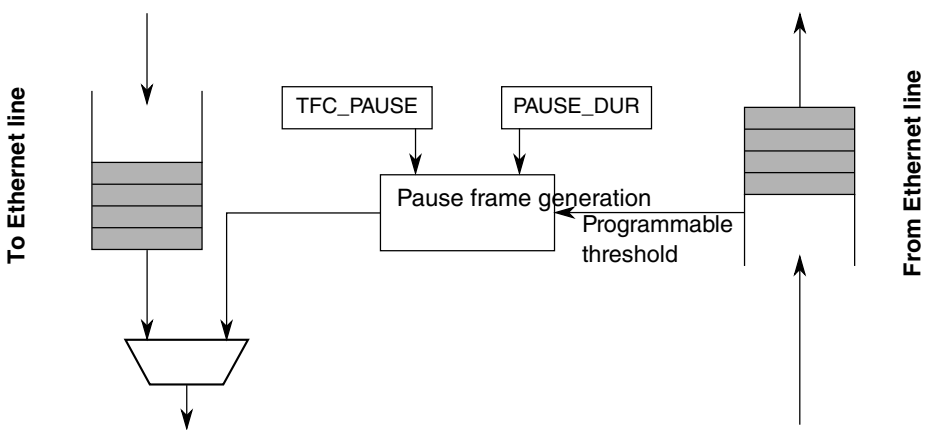


Figure 11-67. Pause frame generation overview

Note

Although the flow control mechanism should prevent any FIFO overflow on the MAC core receive path, the core receive FIFO is protected. When an overflow is detected on the receive FIFO, the current frame is truncated with an error indication set in the frame status word. The frame should subsequently be discarded by the user application.

11.6.2.7 Magic packet detection

Magic packet detection wakes a node that is put in power-down mode by the node management agent. Magic packet detection is supported only if the MAC is configured in sleep mode.

11.6.2.7.1 Sleep mode

To put the MAC in Sleep mode, set ENET*n*_ECR[SLEEP]. At the same time ENET*n*_ECR[MAGICEN] should also be set to enable magic packet detection.

In addition, if ENET is enabled, write 1 to ENET*n*_ECR[SLEEP] before entering into low power mode.

When the MAC is in Sleep mode:

- The transmit logic is disabled.
- The FIFO receive/transmit functions are disabled.
- The receive logic is kept in Normal mode, but it ignores all traffic from the line except magic packets. They are detected so that a remote agent can wake the node.

11.6.2.7.2 Magic packet detection

The core is designed to detect magic packets (See "Magic Packets" topics) with the destination address set to:

- Any multicast address
- The broadcast address
- The unicast address programmed in PADDR1/2

When a magic packet is detected, EIR[WAKEUP] is set and none of the statistic registers are incremented.

11.6.2.7.3 Wakeup

When a magic packet is detected, indicated by ENET*n*_EIR[WAKEUP], ENET*n*_ECR[SLEEP] should be cleared to resume normal operation of the MAC. Clearing the SLEEP bit automatically masks ENET*n*_ECR[MAGICEN], disabling magic packet detection.

11.6.2.8 IP accelerator functions

The following sections describe the IP accelerator functions.

11.6.2.8.1 Checksum calculation

The IP and ICMP, TCP, UDP checksums are calculated with one's complement arithmetic summing up 16-bit values.

- For ICMP, the checksum is calculated over the complete ICMP datagram, in other words without IP header.
- For TCP and UDP, the checksums contain the header and data sections and values from the IP header, which can be seen as a pseudo-header that is not actually present in the data stream.

Table 11-70. IPv4 pseudo-header for checksum calculation

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source address							
Destination address							
Zero			Protocol		TCP/UDP length		

Table 11-71. IPv6 pseudo-header for checksum calculation

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source address							
Destination address							
TCP/UDP length							
Zero						Next header	

The TCP/UDP length value is the length of the TCP or UDP datagram, which is equal to the payload of an IP datagram. It is derived by subtracting the IP header length from the complete IP datagram length that is given in the IP header (IPv4), or directly taken from the IP header (IPv6). The protocol field is the corresponding value from the IP header. The Zero fields are all zeroes.

For IPv6, the complete 128-bit addresses are considered. The next header value identifies the upper layer protocol as either TCP or UDP. It may differ from the next header value of the IPv6 header if extension headers are inserted before the protocol header.

The checksum calculation uses 16-bit words in network byte order: The first byte sent/received is the MSB, and the second byte sent/received is the LSB of the 16-bit value to add to the checksum. If the frame ends on an odd number of bytes, a zero byte is appended for checksum calculation only, and is not actually transmitted.

11.6.2.8.2 Additional padding processing

According to IEEE 802.3, any Ethernet frame must have a minimum length of 64 octets.

The MAC usually removes padding on receive when a frame with length information is received. Because IP frames have a type value instead of length, the MAC does not remove padding for short IP frames, as it is not aware of the frame contents.

The IP accelerator function can be configured to remove the Ethernet padding bytes that might follow the IP datagram.

On transmit, the MAC automatically adds padding as necessary to fill any frame to a 64-byte length.

11.6.2.8.3 32-bit Ethernet payload alignment

The data FIFOs allow inserting two additional arbitrary bytes in front of a frame. This extends the 14-byte Ethernet header to a 16-byte header, which leads to alignment of the Ethernet payload, following the Ethernet header, on a 32-bit boundary.

This function can be enabled for transmit and receive independently with the corresponding SHIFT16 bits in the ENET*n*_TACC and ENET*n*_RACC registers.

When enabled, the valid frame data is arranged as shown in Table 11-72.

Table 11-72. 64-bit interface data structure with SHIFT16 enabled

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0	Any value	Any value
Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8	Byte 7	Byte 6
...							

11.6.2.8.3.1 Receive processing

When ENET*n*_RACC[SHIFT16] is set, each frame is received with two additional bytes in front of the frame.

The user application must ignore these first two bytes and find the first byte of the frame in bits 23–16 of the first word from the RX FIFO.

Note

SHIFT16 must be set during initialization and kept set during the complete operation, because it influences the FIFO write behavior.

11.6.2.8.3.2 Transmit processing

When ENET*n*_TACC[SHIFT16] is set, the first two bytes of the first word written (bits 15–0) are discarded immediately by the FIFO write logic.

The SHIFT16 bit can be enabled/disabled for each frame individually if required, but can be changed only between frames.

11.6.2.8.4 Received frame discard

Because the receive FIFO must be operated in store and forward mode (ENET*n*_RSFL cleared), received frames can be discarded based on the following errors:

- The MAC function receives the frame with an error:
  - The frame has an invalid payload length
  - Frame length is greater than MAX_FL
  - Frame received with a CRC-32 error
  - Frame truncated due to receive FIFO overflow
  - Frame is corrupted as PHY signaled an error (RX_ERR asserted during reception)
- An IP frame is detected and the IP header checksum is wrong
- An IP frame with a valid IP header and a valid IP header checksum is detected, the protocol is known but the protocol-specific checksum is wrong

If one of the errors occurs and the IP accelerator function is configured to discard frames (ENET*n*_RACC), the frame is automatically discarded. Statistics are maintained normally and are not affected by this discard function.

11.6.2.8.5 IPv4 fragments

When an IPv4 IP fragment frame is received, only the IP header is inspected and its checksum verified. 32-bit alignment operates the same way on fragments as it does on normal IP frames, as specified above.

The IP fragment frame payload is not inspected for any protocol headers. As such, a protocol header would only exist in the very first fragment. To assist in protocol-specific checksum verification, the one's-complement sum is calculated on the IP payload (all bytes following the IP header) and provided with the frame status word.

The frame fragment status field (RxBD[FRAG]) is set to indicate a fragment reception, and the one's-complement sum of the IP payload is available in RxBD[Payload checksum].

Note

After all fragments have been received and reassembled, the application software can take advantage of the payload checksum delivered with the frame's status word to calculate the protocol-specific checksum of the datagram.

For example, if a TCP payload is delivered by multiple IP fragments, the application software can calculate the pseudo-header checksum value from the first fragment, and add the payload checksums delivered with the status for all fragments to verify the TCP datagram checksum.

11.6.2.8.6 IPv6 support

The following sections describe the IPv6 support.

11.6.2.8.6.1 Receive processing

An Ethernet frame of type 0x86DD identifies an IP Version 6 frame (IPv6) frame. If an IPv6 frame is received, the first IP header is inspected (first ten words), which is available in every IPv6 frame.

If the receive SHIFT16 function is enabled, the IP header is aligned on a 32-bit boundary allowing more efficient processing (see 32-bit Ethernet payload alignment).

For TCP and UDP datagrams, the pseudo-header checksum calculation is performed and verified.

To assist in protocol-specific checksum verification, the one's-complement sum is always calculated on the IP payload (all bytes following the IP header) and provided with the frame status word. For example, if extension headers were present, their sums can be subtracted in software from the checksum to isolate the TCP/UDP datagram checksum, if required.

11.6.2.8.6.2 Transmit processing

For IPv6 transmission, the SHIFT16 function is supported to process 32-bit aligned datagrams.

IPv6 has no IP header checksum; therefore, the IP checksum insertion configuration is ignored.

The protocol checksum is inserted only if the next header of the IP header is a known protocol (TCP, UDP, or ICMP). If a known protocol is detected, the checksum over all bytes following the IP header is calculated and inserted in the correct position.

The pseudo-header checksum calculation is performed for TCP and UDP datagrams accordingly.

11.6.2.9 Resets and stop controls

The following sections describe the resets and stop controls.

11.6.2.9.1 Hardware reset

To reset the Ethernet module, set ENET*n*_ECR[RESET].

11.6.2.9.2 Soft reset

When ENET*n*_ECR[ETHER_EN] is cleared during operation, the following occurs:

- uDMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers.
- A currently ongoing transmit is terminated by asserting GMII/TXER to the PHY.
- A currently ongoing transmit FIFO write from the application is terminated by stopping the write to the FIFO, and all further data from the application is ignored. All subsequent writes are ignored until re-enabled.
- A currently ongoing receive FIFO read is terminated. The RxBD has arbitrary values in this case.

11.6.2.9.3 Hardware freeze

When the processor enters debug mode and ECR[DBGEN] is set, the MAC enters a freeze state where it stops all transmit and receive activities gracefully.

The following happens when the MAC enters hardware freeze:

- A currently ongoing receive transaction on the receive application interface is completed as normal. No further frames are read from the FIFO.
- A currently ongoing transmit transaction on the transmit application interface is completed as normal (in other words, until writing end-of-packet (EOP)).
- A currently ongoing frame receive is completed normally, after which no further frames are accepted from the RMII/GMII.
- A currently ongoing frame transmit is completed normally, after which no further frames are transmitted.

11.6.2.9.4 Graceful stop

During a graceful stop, any currently ongoing transactions are completed normally and no further frames are accepted. The MAC can resume from a graceful stop without the need for a reset (for example, clearing ETHER_EN is not required).

The following conditions lead to a graceful stop of the MAC transmit or receive datapaths.

11.6.2.9.4.1 Graceful transmit stop (GTS)

When gracefully stopped, the MAC is no longer reading frame data from the transmit FIFO and has completed any ongoing transmission.

In any of the following conditions, the transmit datapath stops after an ongoing frame transmission has been completed normally.

- ENET*n*_TCR[GTS] is set by software.
- ENET*n*_TCR[TFC_PAUSE] is set by software requesting a pause frame transmission. The status (and register bit) is cleared after the pause frame has been sent.
- A pause frame was received stopping the transmitter. The stopped situation is terminated when the pause timer expires or a pause frame with zero quanta is received.

- MAC is placed in Sleep mode by software or the processor entering Stop mode (see Sleep mode).
- The MAC is in Hardware Freeze mode.

When the transmitter has reached its stopped state, the following events occur:

- The GRA interrupt is asserted, when transitioned into stopped state.
- In Hardware Freeze mode, the GRA interrupt does not wait for the application write completion and asserts when the transmit state machine (in other words, line side of TX FIFO) reaches its stopped state.

11.6.2.9.4.2 Graceful receive stop (GRS)

When gracefully stopped, the MAC is no longer writing frames into the receive FIFO.

The receive datapath stops after any ongoing frame reception has been completed normally, if any of the following conditions occur:

- MAC is placed in Sleep mode either by the software or the processor is in Stop mode). The MAC continues to receive frames and search for magic packets if enabled (see Magic packet detection). However, no frames are written into the receive FIFO, and therefore are not forwarded to the application.
- The MAC is in Hardware Freeze mode. The MAC does not accept any frames from the RMII/GMII.

When the receive datapath is stopped, the following events occur:

- If the RX is in the stopped state, RCR[GRS] is set
- The GRA interrupt is asserted when the transmitter and receiver are stopped
- Any ongoing receive transaction to the application (RX FIFO read) continues normally until the frame end of package (EOP) is reached. After this, the following occurs:
  - When Sleep mode is active, all further frames are discarded, flushing the RX FIFO
  - In Hardware Freeze mode, no further frames are delivered to the application and they stay in the receive FIFO.

Note

The assertion of GRS does not wait for an ongoing FIFO read transaction on the application side of the FIFO (FIFO read).

11.6.2.9.4.3 Graceful stop interrupt (GRA)

The graceful stopped interrupt (GRA) is asserted for the following conditions:

- In Sleep mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- In Hardware Freeze mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- The MAC transmit datapath is stopped for any other condition (GTS, TFC_PAUSE, pause received).

The GRA interrupt is triggered only once when the stopped state is entered. If the interrupt is cleared while the stop condition persists, no further interrupt is triggered.

11.6.2.10 IEEE 1588 functions

To allow for IEEE 1588 or similar time synchronization protocol implementations, the MAC is combined with a time-stamping module to support precise time-stamping of incoming and outgoing frames. Set ENET*n*_ECR[EN1588] to enable 1588 support.

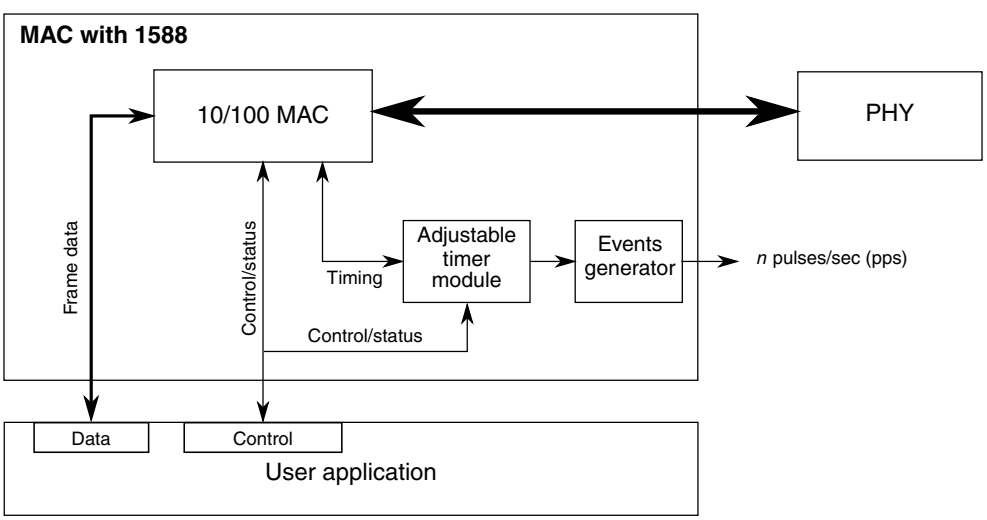


Figure 11-68. IEEE 1588 functions overview

11.6.2.10.1 Adjustable timer module

The adjustable timer module (TSM) implements the free-running counter (FRC), which generates the timestamps. The FRC operates with the time-stamping clock, which can be set to any value depending on your system requirements.

Through dedicated correction logic, the timer can be adjusted to allow synchronization to a remote master and provide a synchronized timing reference to the local system. The timer can be configured to cause an interrupt after a fixed time period, to allow synchronization of software timers or perform other synchronized system functions.

The timer is typically used to implement a period of one second; hence, its value ranges from 0 to (1×10^9)-1. The period event can trigger an interrupt, and software can maintain the seconds and hours time values as necessary.

11.6.2.10.1.1 Adjustable timer implementation

The adjustable timer consists of a programmable counter/accumulator and a correction counter. The periods of both counters and their increment rates are freely configurable, allowing very fine tuning of the timer.

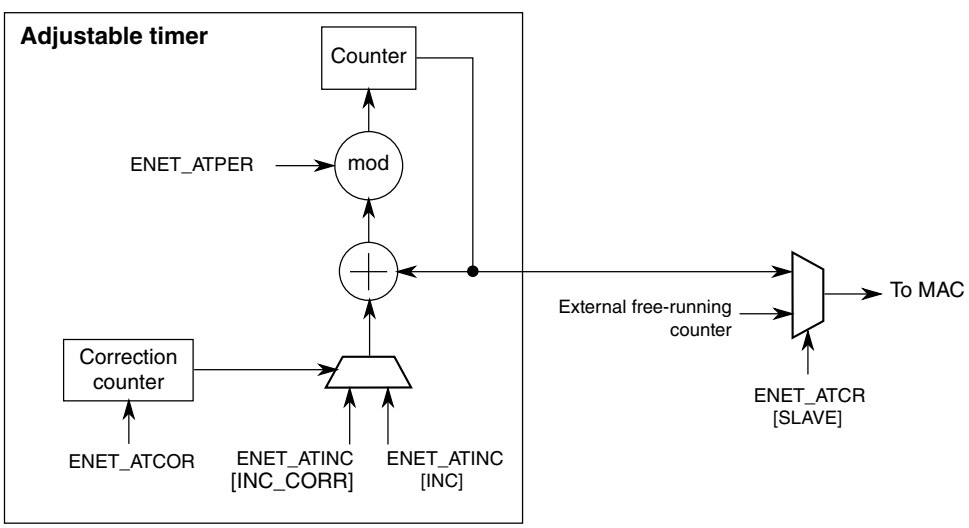


Figure 11-69. Adjustable timer implementation detail

The counter produces the current time. During each time-stamping clock cycle, a constant value is added to the current time as programmed in ENET*n*_ATINC. The value depends on the chosen time-stamping clock frequency. For example, if it operates at 125 MHz, setting the increment to eight represents 8 ns.

The period, configured in ENET*n*_ATPER, defines the modulo when the counter wraps. In a typical implementation, the period is set to 1×10^9 so that the counter wraps every second, and hence all timestamps represent the absolute nanoseconds within the one

second period. When the period is reached, the counter wraps to start again respecting the period modulo. This means it does not necessarily start from zero, but instead the counter is loaded with the value (Current + Inc –(1×10^9)), assuming the period is set to 1×10^9.

The correction counter operates fully independently, and increments by one with each time-stamping clock cycle. When it reaches the value configured in ENET*n*_ATCOR, it restarts and instructs the timer once to increment by the correction value, instead of the normal value.

The normal and correction increments are configured in ENET*n*_ATINC. To speed up the timer, set the correction increment more than the normal increment value. To slow down the timer, set the correction increment less than the normal increment value.

The correction counter only defines the distance of the corrective actions, not the amount. This allows very fine corrections and low jitter (in the range of 1 ns) independent of the chosen clock frequency.

By enabling slave mode (ENET*n*_ATCR[SLAVE] = 1), the timer is ignored and the current time is externally provided from one of the external modules. See the Chip Configuration details for which clock source is used. This is useful if multiple modules within the system must operate from a single timer. When slave mode is enabled, you still must set ENET*n*_ATINC[INC] to the value of the master, since it is used for internal comparisons.

11.6.2.10.2 Transmit timestamping

Only 1588 event frames need to be time-stamped on transmit. The client application (for example, the MAC driver) should detect 1588 event frames and set TxBD[TS] together with the frame.

If TxBD[TS] is set, the MAC records the timestamp for the frame in ENET*n*_ATSTMP. ENET*n*_EIR[TS_AVAIL] is set to indicate that a new timestamp is available.

Software implements a handshaking procedure by setting TxBD[TS] when it transmits the frame for which a timestamp is needed, and then waits for ENET*n*_EIR[TS_AVAIL] to determine when the timestamp is available. The timestamp is then read from ENET*n*_ATSTMP. This is done for all event frames. Other frames do not use TxBD[TS] and, therefore, do not interfere with the timestamp capture.

11.6.2.10.3 Receive timestamping

When a frame is received, the MAC latches the value of the timer when the frame's start of frame delimiter (SFD) field is detected, and provides the captured timestamp on RxBD[1588 timestamp]. This is done for all received frames.

11.6.2.10.4 Time synchronization

The adjustable timer module is available to synchronize the local clock of a node to a remote master. It implements a free running 32-bit counter, and also contains an additional correction counter.

The correction counter increases or decreases the rate of the free running counter, enabling very fine granular changes of the timer for synchronization, yet adding only very low jitter when performing corrections.

The application software implements, in a slave scenario, the required control algorithm, setting the correction to compensate for local oscillator drifts and locking the timer to the remote master clock on the network.

The timer and all timestamp-related information should be configured to show the true nanoseconds value of a second (in other words, the timer is configured to have a period of one second). Hence, the values range from 0 to (1×10^9)–1. In this application, the seconds counter is implemented in software using an interrupt function that is executed when the nanoseconds counter wraps at 1×10^9.

11.6.2.10.5 Input Capture and Output Compare

The Input Capture Output Compare block can be used to provide precise hardware timing for input and output events.

11.6.2.10.5.1 Input capture

The TCCR*n* capture registers latch the time value when the corresponding external event occurs. An event can be a rising-, falling-, or either-edge of one of the 1588_TMR*n* signals. An event will cause the corresponding TCSR*n*[TF] timer flag to be set, indicating that an input capture has occurred. If the corresponding interrupt is enabled with the TCSR*n*[TIE] field, an interrupt can be generated.

11.6.2.10.5.2 Output compare

The TCCR*n* compare registers are loaded with the time at which the corresponding event should occur. When the ENET free-running counter value matches the output compare reference value in the TCCR*n* register, the corresponding flag, TCSR*n*[TF], is set, indicating that an output compare has occurred. The corresponding interrupt, if enabled by TCSR*n*[TIE], will be generated.The corresponding 1588_TMR*n* output signal will be asserted according to TCSR*n*[TMODE].

NOTE

If TSCR*n*[TMODE] is set to 10X1b or 1010b then the timer output pin toggle-on-overflow will occur only when PINPER, PEREN, and EN bits of the ATCR register are one.

11.6.2.10.5.3 DMA requests

A DMA request can be enabled by setting TCSR*n*[TDRE]. The corresponding DMA request is generated when the TCSR*n*[TF] timer flag is set. When the DMA has completed, the corresponding TCSRn[TF] flag is cleared.

11.6.2.11 FIFO thresholds

The core FIFO thresholds are fully programmable to dynamically change the FIFO operation.

For example, store and forward transfer can be enabled by a simple change in the FIFO threshold registers.

The thresholds are defined in 64-bit words.

The receive and transmit FIFOs both have a depth of 1024 words.

11.6.2.11.1 Receive FIFO

Four programmable thresholds are available, which can be set to any value to control the core operation as follows.

Table 11-73. Receive FIFO thresholds definition

Register	Description
ENET*n*_RSFL [RX_SECTION_FULL]	When the FIFO level reaches the ENET*n*_RSFL value, the MAC status signal is asserted to indicate that data is available in the receive FIFO (cut-through operation). Once asserted, if the FIFO empties below the threshold set with ENET*n*_RAEM and if the end-of-frame is not yet stored in the FIFO, the status signal is deasserted again. If a frame has a size smaller than the threshold (in other words, an end-of-frame is available for the frame), the status is also asserted. To enable store and forward on the receive path, clear ENET*n*_RSFL. The MAC status signal is asserted only when a complete frame is stored in the receive FIFO. When programming a non-zero value to ENET*n*_RSFL (cut-through operation) it should be greater than ENET*n*_RAEM.
ENET*n*_RAEM [RX_ALMOST_EMPTY]	When the FIFO level reaches the ENET*n*_RAEM value, and the end-of-frame has not been received, the core receive read control stops the FIFO read (and subsequently stops transferring data to the MAC client application).

Table continues on the next page...

Table 11-73. Receive FIFO thresholds definition (continued)

Register	Description
	It continues to deliver the frame, if again more data than the threshold or the end-of-frame is available in the FIFO. Set ENET*n*_RAEM to a minimum of six.
ENET*n*_RAFL [RX_ALMOST_FULL]	When the FIFO level approaches the maximum and there is no more space remaining for at least ENETn_RAFL number of words, the MAC control logic stops writing data in the FIFO and truncates the receive frame to avoid FIFO overflow. The corresponding error status is set when the frame is delivered to the application. Set ENET*n*_RAFL to a minimum of 4.
ENET*n*_RSEM [RX_SECTION_EMPTY]	When the FIFO level reaches the ENET*n*_RSEM value, an indication is sent to the MAC transmit logic, which generates an XOFF pause frame. This indicates FIFO congestion to the remote Ethernet client. When the FIFO level goes below the value programmed in ENET*n*_RSEM, an indication is sent to the MAC transmit logic, which generates an XON pause frame. This indicates the FIFO congestion is cleared to the remote Ethernet client. Clearing ENET*n*_RSEM disables any pause frame generation.

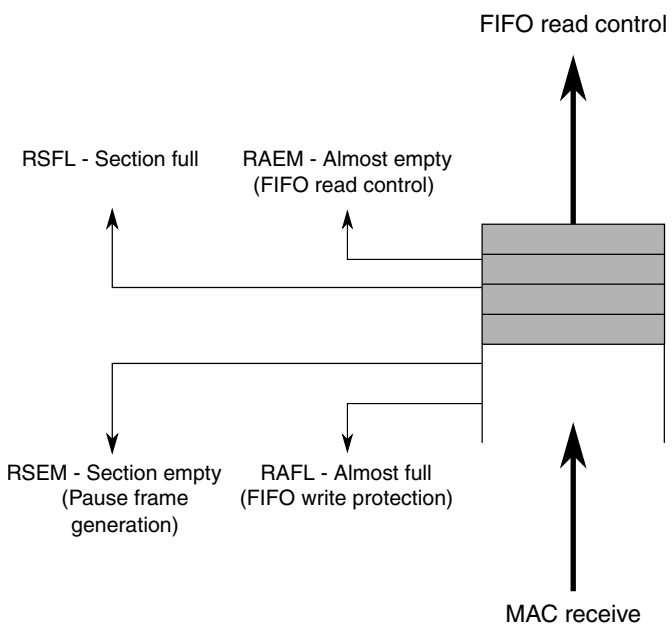


Figure 11-70. Receive FIFO overview

11.6.2.11.2 Transmit FIFO

Four programmable thresholds are available which control the core operation as described below.

Table 11-74. Transmit FIFO thresholds definition

Register	Description
ENET*n*_TAEM [TX_ALMOST _EMPTY]	When the FIFO level reaches the ENET*n*_TAEM value and no end-of-frame is available for the frame, the MAC transmit logic avoids a FIFO underflow by stopping FIFO reads and transmitting the Ethernet frame with an MII error indication. Set ENET*n*_TAEM to a minimum of 4.
ENET*n*_TAFL [TX_ALMOST _FULL]	When the FIFO level approaches the maximum, so that there is no more space for at least ENET*n*_TAFL number of words, the MAC deasserts its control signal to the application. If the application does not react on this signal, the FIFO write control logic avoids FIFO overflow by truncating the current frame and setting the error status. As a result, the frame is transmitted with an GMII/MII error indication. Set ENET*n*_TAFL to a minimum of 4. Larger values allow more latency for the application to react on the MAC control signal deassertion, before the frame is truncated. A typical setting is 8, which offers 3–4 clock cycles of latency to the application to react on the MAC control signal deassertion.
ENET*n*_TSEM [TX_SECTION _EMPTY]	When the FIFO level reaches the ENET*n*_TSEM value, a MAC status signal is deasserted to indicate that the transmit FIFO is getting full. This gives the ENET module an indication to slow or stop its write transaction to avoid a buffer overflow. This is a pure indication function to the application. It has no effect within the MAC. When ENET*n*_TSEM is 0, the signal is never deasserted.
ENET*n*_TFWR	When the FIFO level reaches the ENET*n*_TFWR value and when STRFWD is cleared, the MAC transmit control logic starts frame transmission before the end-of-frame is available in the FIFO (cut-through operation). If a complete frame has a size smaller than the ENET*n*_TFWR threshold, the MAC also transmits the frame to the line. To enable store and forward on the transmit path, set STRFWD. In this case, the MAC starts to transmit data only when a complete frame is stored in the transmit FIFO.

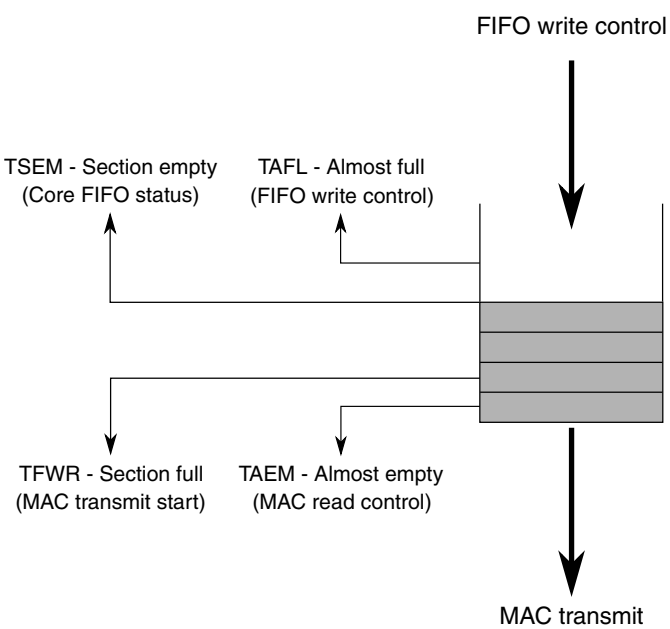


Figure 11-71. Transmit FIFO overview

11.6.2.12 Loopback options

The core implements internal loopback options, which are controlled by the following ENET*n*_RCR register fields:

Table 11-75. Loopback options

Register field	Description
LOOP	Internal MII loopback. The MAC transmit is returned to the MAC receive. No data is transmitted to the external interfaces.

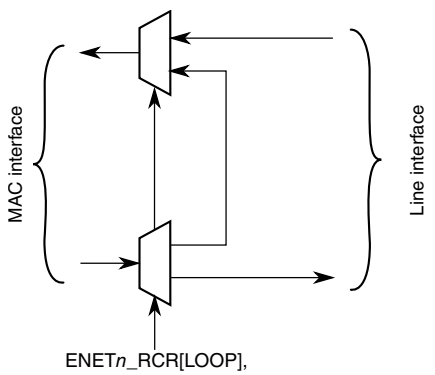


Figure 11-72. Loopback options

11.6.2.13 Legacy buffer descriptors

To support the Ethernet controller on previous chips, legacy FEC buffer descriptors are available. To enable legacy support, write 0 to ENET*n*_ECR[1588EN].

NOTE

- Legacy buffer descriptors are used in single-ring mode, that is, when DMA*n*CFG[DMA_CLASS_EN] are zero. In multi-ring mode, legacy TxBDs are used only with the round-robin scheme.
- The legacy buffer descriptor tables show the byte order for big-endian chips. ECR[DBSWP] must be set to 0 after reset to enable big-endian mode.

11.6.2.13.1 Legacy receive buffer descriptor

The following table shows the legacy FEC receive buffer descriptor. Table 11-79 contains the descriptions for each field.

Table 11-76. Legacy FEC receive buffer descriptor (RxBD)

	Byte 0								Byte 1							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 2	Data length															
Offset + 4	Rx data buffer pointer — high halfword															
Offset + 6	Rx data buffer pointer — low halfword															

11.6.2.13.2 Legacy transmit buffer descriptor

The following table shows the legacy FEC transmit buffer descriptor. Table 11-81 contains the descriptions for each field.

Table 11-77. Legacy FEC transmit buffer descriptor (TxBD)

	Byte 0								Byte 1							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	R	TO1	W	TO2	L	TC	ABC[1]	—	—	—	—	—	—	—	—	—
Offset + 2	Data Length															
Offset + 4	Tx Data Buffer Pointer — high halfword															
Offset + 6	Tx Data Buffer Pointer — low halfword															

1. This field is not supported by the uDMA.

11.6.2.14 Enhanced buffer descriptors

This section provides a description of the enhanced operation of the driver/uDMA via the buffer descriptors.

It is followed by a detailed description of the receive and transmit descriptor fields. To enable the enhanced features, set ENET*n*_ECR[1588EN].

NOTE

The enhanced buffer descriptor tables show the byte order for big-endian chips. ECR[DBSWP] must be set to 0 after reset to enable big-endian mode.

11.6.2.14.1 Enhanced receive buffer descriptor

The following table shows the enhanced uDMA receive buffer descriptor. Table 11-79 contains the descriptions for each field.

Table 11-78. Enhanced uDMA receive buffer descriptor (RxBD)

	Byte 0								Byte 1							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 2	Data length															
Offset + 4	Rx data buffer pointer – high halfword															
Offset + 6	Rx data buffer pointer – low halfword															
Offset + 8	ME	—	—	—	—	PE	CE	UC	INT	—	—	—	—	—	—	—
Offset + A	VPCP			—	—	—	—	—	—	—	ICE	PCR	—	VLAN	IPV6	FRAG
Offset + C	Header length					—	—	—	Protocol type							
Offset + E	Payload checksum															
Offset + 10	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – high halfword															
Offset + 16	1588 timestamp – low halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 11-79. Receive buffer descriptor field definitions

Word	Field	Description
Offset + 0	15 E	Empty. Written by the MAC (= 0) and user (= 1). 0 The data buffer associated with this BD is filled with received data, or data reception has aborted due to an error condition. The status and length fields have been updated as required. 1 The data buffer associated with this BD is empty, or reception is currently in progress.
Offset + 0	14 RO1	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.
Offset + 0	13 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location. 1 The next buffer descriptor is found at the location defined in ENET*n*_RDSR.
Offset + 0	12 RO2	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.
Offset + 0	11 L	Last in frame. Written by the uDMA. 0 The buffer is not the last in a frame.

Table continues on the next page...

Table 11-79. Receive buffer descriptor field definitions (continued)

Word	Field	Description
		1 The buffer is the last in a frame.
Offset + 0	10–9	Reserved, must be cleared.
Offset + 0	8 M	Miss. Written by the MAC. This field is set by the MAC for frames accepted in promiscuous mode, but flagged as a miss by the internal address recognition. Therefore, while in promiscuous mode, you can use this field to quickly determine whether the frame was destined to this station. This field is valid only if the L and PROM fields are set to 1. 0 The frame was received because of an address recognition hit. 1 The frame was received because of promiscuous mode. The information needed for this field comes from the promiscuous_miss(ff_rx_err_stat[26]) sideband signal.
Offset + 0	7 BC	Set if the DA is broadcast (FFFF_FFFF_FFFF).
Offset + 0	6 MC	Set if the DA is multicast and not BC.
Offset + 0	5 LG	Receive frame length violation. Written by the MAC. A frame length greater than RCR[MAX_FL] was recognized. This field is valid only if the L field is set. The receive data is not altered in any way unless the length exceeds TRUNC_FL bytes.
Offset + 0	4 NO	Receive non-octet aligned frame. Written by the MAC. A frame that contained a number of bits not divisible by 8 was received, and the CRC check that occurred at the preceding byte boundary generated an error or a PHY error occurred. This field is valid only if the L field is set. If this field is set, the CR field is not set.
Offset + 0	3	Reserved, must be cleared.
Offset + 0	2 CR	Receive CRC or frame error. Written by the MAC. This frame contains a PHY or CRC error and is an integral number of octets in length. This field is valid only if the L field is set.
Offset + 0	1 OV	Overrun. Written by the MAC. A receive FIFO overrun occurred during frame reception. If this field is set, the other status fields, M, LG, NO, and CR, lose their normal meaning and are zero. This field is valid only if the L field is set.
Offset + 0	0 TR	Set if the receive frame is truncated (frame length >TRUNC_FL). If the TR field is set, the frame must be discarded and the other error fields must be ignored because they may be incorrect.
Offset + 2	15–0 Data Length	Data length. Written by the MAC. Data length is the number of octets written by the MAC into this BD's data buffer if L is cleared (the value is equal to EMRBR), or the length of the frame including CRC if L is set. It is written by the MAC once as the BD is closed.
0ffset + 4	15–0 Data buffer pointer high	Receive data buffer pointer, high halfword[1]
Offset + 6	15–0 Data buffer pointer low	Receive data buffer pointer, low halfword
Offset + 8	15 ME	MAC error. This field is written by the uDMA. This field means that the frame stored in the system memory was received with an error (typically, a receive FIFO overflow). This field is only valid when the L field is set.
Offset + 8	14–11	Reserved, must be cleared.

Table continues on the next page...

Table 11-79. Receive buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 8	10 PE	PHY Error. This field is written by the uDMA. Set to "1"when the frame was received with an Error character on the PHY interface. The frame is invalid. This field is valid only when the L field is set.
Offset + 8	9 CE	Collision. This field is written by the uDMA. Set when the frame was received with a collision detected during reception. The frame is invalid and sent to the user application. This field is valid only when the L field is set.
Offset + 8	8 UC	Unicast. This field is written by the uDMA, and means that the frame is unicast. This field is valid regardless of whether the L field is set.
Offset + 8	7 INT	Generate RXB/RXF interrupt. This field is set by the user to indicate that the uDMA is to generate an interrupt on the *dma_int_rxb* / *dma_int_rxfevent*.
Offset + 8	6–0	Reserved, must be cleared.
Offset + A	15–13 VPCP	VLAN priority code point. This field is written by the uDMA to indicate the frame priority level. Valid values are from 0 (best effort) to 7 (highest). This value can be used to prioritize different classes of traffic (e.g., voice, video, data). This field is only valid if the L field is set.
Offset + A	12–6	Reserved, must be cleared.
Offset + A	5 ICE	IP header checksum error. This is an accelerator option. This field is written by the uDMA. Set when either a non-IP frame is received or the IP header checksum was invalid. An IP frame with less than 3 bytes of payload is considered to be an invalid IP frame. This field is only valid if the L field is set.
Offset + A	4 PCR	Protocol checksum error. This is an accelerator option. This field is written by the uDMA. Set when the checksum of the protocol is invalid or an unknown protocol is found and checksumming could not be performed. This field is only valid if the L field is set.
Offset + A	3	Reserved, must be cleared.
Offset + A	2 VLAN	VLAN. This is an accelerator option. This field is written by the uDMA. It means that the frame has a VLAN tag. This field is valid only if the L field is set.
Offset + A	1 IPV6	IPV6 Frame. This field is written by the uDMA. This field indicates that the frame has an IPv6 frame type. If this field is not set it means that an IPv4 or other protocol frame was received. This field is valid only if the L field is set.
Offset + A	0 FRAG	IPv4 Fragment.This is an accelerator option.This field is written by the uDMA. It indicates that the frame is an IPv4 fragment frame. This field is only valid when the L field is set.
Offset + C	15–11 Header length	Header length. This is an accelerator option. This field is written by the uDMA. This field is the sum of 32-bit words found within the IP and its following protocol headers. If an IP datagram with an unknown protocol is found, then the value is the length of the IP header. If no IP frame or an erroneous IP header is found, the value is 0. The following values are minimum values if no header options exist in the respective headers: • ICMP/IP: 6 (5 IP header, 1 ICMP header) • UDP/IP: 7 (5 IP header, 2 UDP header) • TCP/IP: 10 (5 IP header, 5 TCP header) This field is only valid if the L field is set.
Offset + C	10–8	Reserved, must be cleared.
Offset + C	7–0 Protocol type	Protocol type. This is an accelerator option. The 8-bit protocol field found within the IP header of the frame. It is valid only when ICE is cleared. This field is valid only when the L field is set.

Table continues on the next page...

Table 11-79. Receive buffer descriptor field definitions (continued)

Word	Field	Description
Offset + E	15–0 Payload checksum	Internet payload checksum. This is an accelerator option. It is the one's complement sum of the payload section of the IP frame. The sum is calculated over all data following the IP header until the end of the IP payload. This field is valid only when the L field is set.
Offset + 10	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 10	14–0	Reserved, must be cleared.
Offset + 12	15–0	Reserved, must be cleared.
Offset + 14	15–0 1588 timestamp	This value is written by the uDMA. It is only valid if the L field is set.
Offset + 16		
Offset + 18 – Offset + 1E	15–0	Reserved, must be cleared.

1. The receive buffer pointer, containing the address of the associated data buffer, must always be evenly divisible by 64. The buffer must reside in memory external to the MAC. The Ethernet controller never modifies this value.

11.6.2.14.2 Enhanced transmit buffer descriptor

Table 11-80. Enhanced transmit buffer descriptor (TxBD)

	Byte 0								Byte 1							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	R	TO1	W	TO2	L	TC	—	—	—	—	—	—	—	—	—	—
Offset + 2	Data length															
Offset + 4	Tx Data Buffer Pointer – high halfword															
Offset + 6	Tx Data Buffer Pointer – low halfword															
Offset + 8	—	INT	TS	PINS	IINS	—	—	UTLT	FTYPE				—	—	—	—
Offset + A	TXE	—	UE	EE	FE	LCE	OE	TSE	—	—	—	—	—	—	—	—
Offset + C	TLT – high halfword															
Offset + E	TLT – low halfword															
Offset + 10	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – high halfword															
Offset + 16	1588 timestamp – low halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 11-81. Enhanced transmit buffer descriptor field definitions

Word	Field	Description
Offset + 0	15 R	Ready. Written by the MAC and user. 0 The data buffer associated with this BD is not ready for transmission. You are free to manipulate this BD or its associated data buffer. The MAC clears this field after the buffer has been transmitted or after an error condition is encountered. 1 The data buffer, prepared for transmission by you, has not been transmitted or currently transmits. You may write no fields of this BD after this field is set.
Offset + 0	14 TO1	Transmit software ownership. This field is reserved for software use. This read/write field is not modified by hardware and its value does not affect hardware.
Offset + 0	13 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location 1 The next buffer descriptor is found at the location defined in ETDSR.
Offset + 0	12 TO2	Transmit software ownership. This field is reserved for use by software. This read/write field is not modified by hardware and its value does not affect hardware.
Offset + 0	11 L	Last in frame. Written by user. 0 The buffer is not the last in the transmit frame 1 The buffer is the last in the transmit frame
Offset + 0	10 TC	Transmit CRC. Written by user, and valid only when L is set. 0 End transmission immediately after the last data byte 1 Transmit the CRC sequence after the last data byte This field is valid only when the L field is set.
Offset + 0	9 ABC	Append bad CRC. **Note:** This field is not supported by the uDMA and is ignored.
Offset + 0	8–0	Reserved, must be cleared.
Offset + 2	15–0 Data Length	Data length, written by user. Data length is the number of octets the MAC should transmit from this BD's data buffer. It is never modified by the MAC.
Offset + 4	15–0 Data buffer pointer high	Tx data buffer pointer, high halfword. The buffer must reside in memory external to the MAC. This value is never modified by the Ethernet controller. **NOTE:** For optimal performance, make the transmit buffer pointer evenly divisible by 64.
Offset + 6	15–0 Data buffer pointer low	Tx data buffer pointer, low halfword
Offset + 8	15	Reserved, must be cleared.
Offset + 8	14 INT	Generate interrupt flags. This field is written by the user. This field is valid regardless of the L field and must be the same for all EBD for a given frame. The uDMA does not update this value.
Offset + 8	13 TS	Timestamp. This field is written by the user. This indicates that the uDMA is to generate a timestamp frame to the MAC. This field is valid regardless of the L field and must be the same for all EBD for the given frame. The uDMA does not update this value.

Table continues on the next page...

Table 11-81. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 8	12 PINS	Insert protocol specific checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the protocol checksum and overwrites the corresponding checksum field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.
Offset + 8	11 IINS	Insert IP header checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the IP header checksum and overwrites the corresponding header field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.
Offset + 8	10-9	Reserved, must be cleared.
Offset + 8	8 UTLT	Use transmit launch time. If set to 1, TLT high and low values are used to determine if the frame can be transmitted. This field must only be set in the *first* BD of a frame. TLT is always used in combination with either the round-robin scheme or the credit-based shaper. TLT can be used with single a BD queue. However, at least one DMA class, DMA*n*CFG[DMA_CLASS_ EN] must be enabled. For example, to use TLT with only queue 0, enable one of the DMA*n*CFG[DMA_CLASS_EN] fields and clear both TDAR1 and TDAR2. Although this enables multi-queue, only the single queue 0 is used.
Offset + 8	7–4 FTYPE	Type of frame to be transmitted. If a credit-based scheme is used, this field must match the BD ring queue. 0x0 – Non-AVB. Corresponds to ENET_TDSR. 0x1 – AVB Class A. Corresponds to ENET_TDSR1. 0x2 – AVB Class B. Corresponds to ENET_TDSR2. All other values are reserved.
Offset + 8	3–0	Reserved, must be cleared.
Offset + A	15 TXE	Transmit error occurred. This field is written by the uDMA. This field indicates that there was a transmit error of some sort reported with the frame. Effectively this field is an OR of the other error fields including UE, EE, FE, LCE, OE, and TSE. This field is valid only when the L field is set.
Offset + A	14	Reserved, must be cleared.
Offset + A	13 UE	Underflow error. This field is written by the uDMA. This field indicates that the MAC reported an underflow error on transmit. This field is valid only when the L field is set.
Offset + A	12 EE	Excess Collision error. This field is written by the uDMA. This field indicates that the MAC reported an excess collision error on transmit. This field is valid only when the L field is set.
Offset + A	11 FE	Frame with error. This field is written by the uDMA. This field indicates that the MAC reported that the uDMA reported an error when providing the packet. This field is valid only when the L field is set.
Offset + A	10 LCE	Late collision error. This field is written by the uDMA. This field indicates that the MAC reported that there was a Late Collision on transmit. This field is valid only when the L field is set.

Table continues on the next page...

Table 11-81. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
Offset + A	9 OE	Overflow error. This field is written by the uDMA. This field indicates that the MAC reported that there was a FIFO overflow condition on transmit. This field is only valid when the L field is set.
Offset + A	8 TSE	Timestamp error. This field is written by the uDMA. This field indicates that the MAC reported a different frame type then a timestamp frame. This field is valid only when the L field is set.
Offset + A	7–0	Reserved, must be cleared.
Offset + C	15–0 TLT high	Transmit launch time high. Specifies when frame can by transmitted.
Offset + E	15–0 TLT low	Transmit launch time low. Specifies when frame can by transmitted.
Offset + 10	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 10	14–0	Reserved, must be cleared.
Offset + 12	15–0	Reserved, must be cleared.
Offset + 14	15–0 1588 timestamp	This value is written by the uDMA . It is valid only when the L field is set.
Offset + 16		
Offset + 18–Offset + 1E	15–0	Reserved, must be cleared.

11.6.2.15 Client FIFO application interface

The FIFO interface is completely asynchronous from the Ethernet line, and the transmit and receive interface can operate at a different clock rate.

All transfers to/from the user application are handled independently of the core operation, and the core provides a simple interface to user applications based on a two-signal handshake.

11.6.2.15.1 Data structure description

The data structure defined in the following tables for the FIFO interface must be respected to ensure proper data transmission on the Ethernet line. Byte 0 is sent to and received from the line first.

Table 11-82. FIFO interface data structure

	63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Word 0	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Word 1	Byte 15	Byte 14	Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8
...	...							

The size of a frame on the FIFO interface may not be a modulo of 64-bit.

The user application may not care about the Ethernet frame formats in full detail. It needs to provide and receive an Ethernet frame with the following structure:

- Ethernet MAC destination address
- Ethernet MAC source address
- Optional 802.1q VLAN tag (VLAN type and info field)
- Ethernet length/type field
- Payload

Frames on the FIFO interface do not contain preamble and SFD fields, which are inserted and discarded by the MAC on transmit and receive, respectively.

- On receive, CRC and frame padding can be stripped or passed through transparently.
- On transmit, padding and CRC can be provided by the user application, or appended automatically by the MAC independently for each frame. No size restrictions apply.

Note

On transmit, if ENET*n*_TCR[ADDINS] is set, bytes 6–11 of each frame can be set to any value, since the MAC overwrites the bytes with the MAC address programmed in the ENET*n*_PAUR and ENET*n*_PALR registers.

Table 11-83. FIFO interface frame format

Byte number	Field
0–5	Destination MAC address
6–11	Source MAC address
12–13	Length/type field
14–N	Payload data

VLAN-tagged frames are supported on both transmit and receive, and implement additional information (VLAN type and info).

Table 11-84. FIFO interface VLAN frame format

Byte number	Field
0–5	Destination MAC address
6–11	Source MAC address

Table continues on the next page...

Table 11-84. FIFO interface VLAN frame format (continued)

Byte number	Field
12–15	VLAN tag and info
16–17	Length/type field
18–N	Payload data

Note

The standard defines that the LSB of the MAC address is sent/received first, while for all the other header fields — in other words, length/type, VLAN tag, VLAN info, and pause quanta — the MSB is sent/received first.

11.6.2.15.2 Data structure examples

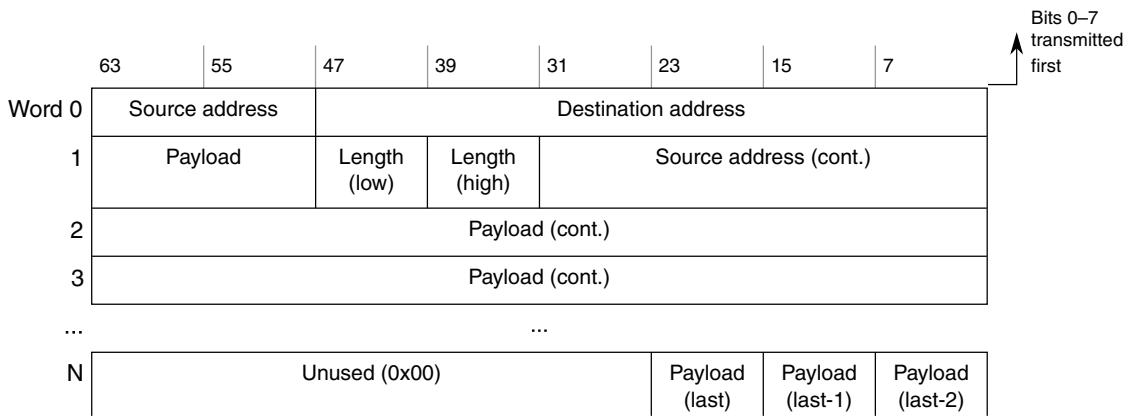


Figure 11-73. Normal Ethernet frame 64-bit mapping example

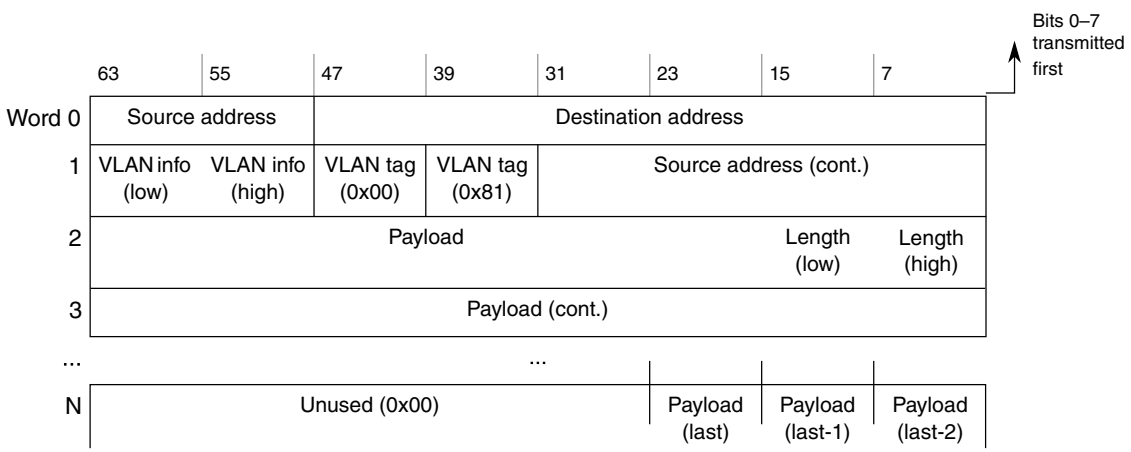


Figure 11-74. VLAN-tagged frame 64-bit mapping example

If CRC forwarding is enabled (CRCFWD = 0), the last four valid octets of the frame contain the FCS field. The non-significant bytes of the last word can have any value.

11.6.2.15.3 Frame status

A MAC layer status word and an accelerator status word is available in the receive buffer descriptor.

See Enhanced buffer descriptors for details.

The status is available with each frame with the last data of the frame.

If the frame status contains a MAC layer error (for example, CRC or length error), RxBD[ME] is also set with the last data of the frame.

11.6.2.16 FIFO protection

The following sections describe the FIFO protection mechanisms.

11.6.2.16.1 Transmit FIFO underflow

During a frame transfer, when the transmit FIFO reaches the almost empty threshold with no end-of-frame indication stored in the FIFO, the MAC logic:

- Stops reading data from the FIFO
- Asserts the MII error signal (MII_TXER) (1 in Figure 11-75) to indicate that the fragment already transferred is not valid
- Deasserts the MII transmit enable signal (MII_TXEN) to terminate the frame transfer (2)

After an underflow, when the application completes the frame transfer (3), the MAC transmit logic discards any new data available in the FIFO until the end of packet is reached (4) and sets the enhanced TxBD[UE] field.

The MAC starts to transfer data on the MII interface when the application sends a new frame with a start of frame indication (5).

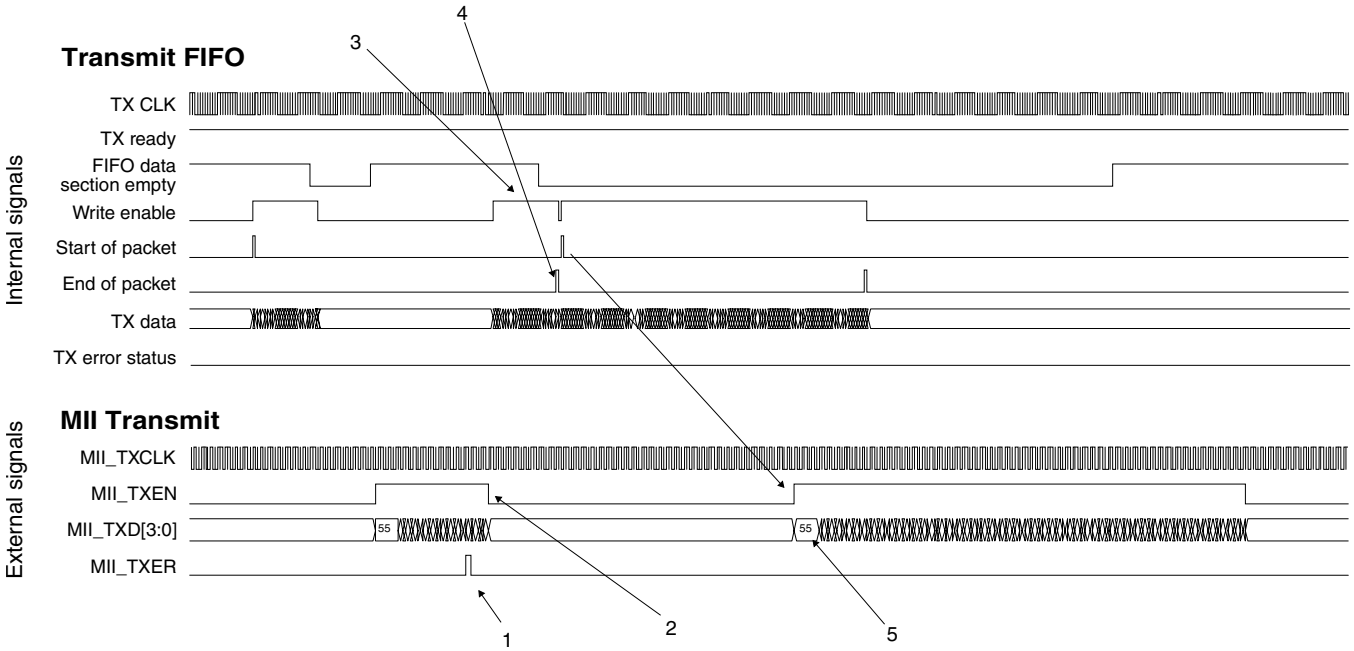


Figure 11-75. Transmit FIFO underflow protection

11.6.2.16.2 Transmit FIFO overflow

On the transmit path, when the FIFO reaches the programmable almost full threshold, the internal MAC ready signal is deasserted. The application should stop sending new data.

However, if the application keeps sending data, the transmit FIFO overflows, corrupting contents that were previously stored. The core logic sets the enhanced TxBD[OE] field for the next frame transmitted to indicate this overflow occurence.

Note

Overflow is a fatal error and must be addressed by resetting the core or clearing ENET*n*_ECR[ETHER_EN], to clear the FIFOs and prepare for normal operation again.

11.6.2.16.3 Receive FIFO overflow

During a frame reception, if the client application is not able to receive data (1), the MAC receive control truncates the incoming frame when the FIFO reaches the programmable almost-full threshold to avoid an overflow.

The frame is subsequently received on the FIFO interface with an error indication (enhanced RxBD[ME] field set together with receive end-of-packet) (2) with the truncation error status field set (3).

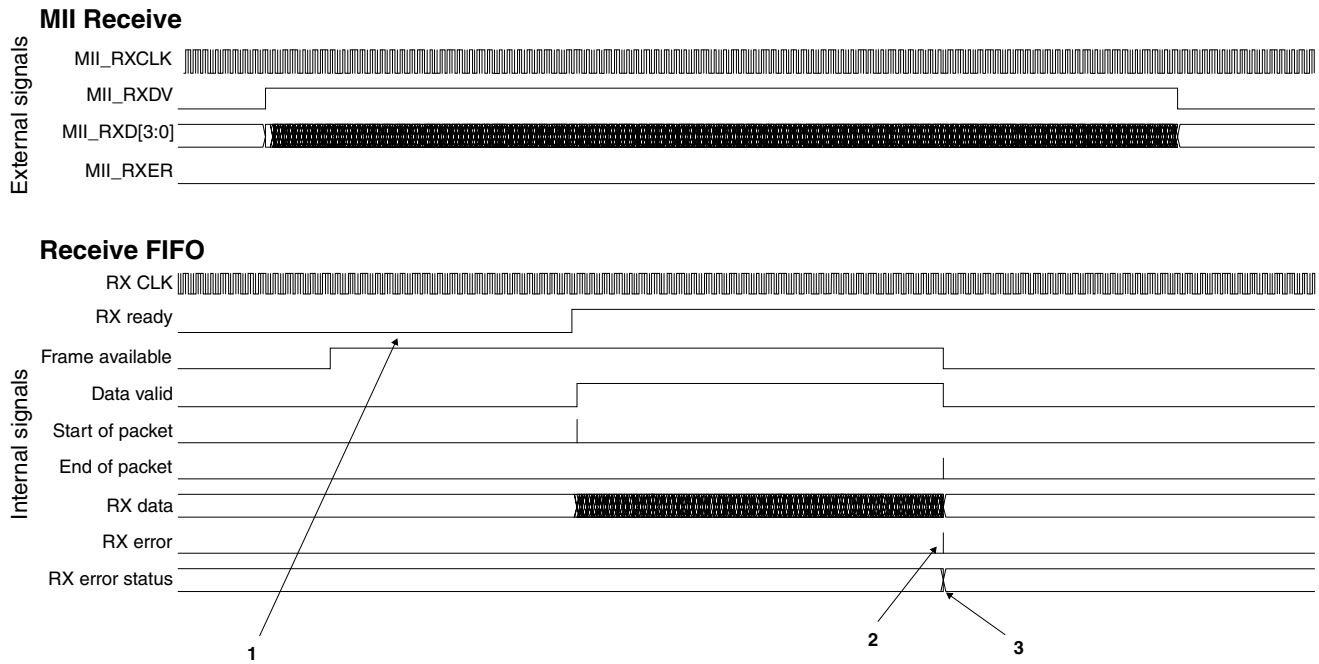


Figure 11-76. Receive FIFO overflow protection

11.6.2.17 PHY management interface

The MDIO interface is a two-wire management interface. The MDIO management interface implements a standardized method to access the PHY device management registers.

The core implements a master MDIO interface, which can be connected to up to 32 PHY devices.

11.6.2.17.1 MDIO clause 22 frame format

The core MDIO master controller communicates with the slave (PHY device) using frames that are defined in the following table.

A complete frame has a length of 64 bits made up of an optional 32-bit preamble, 14-bit command, 2-bit bus direction change, and 16-bit data. Each bit is transferred on the rising edge of the MDIO clock (MDC signal). The MDIO data signal is tri-stated between frames.

The core PHY management interface supports the standard MDIO specification (IEEE 802.3 Clause 22).

Table 11-85. MDIO clause 22 frame structure

ST	OP	PHYADR	REGADR	TA	DATA

Table 11-86. MDIO frame field descriptions

Field	Description
ST (2 bits)	Start indication field, programmed with ENET*n*_MMFR[ST] and equal to 01 for Standard MDIO (Clause 22).
OP (2 bits)	Opcode defines type of operation. Programmed with ENET*n*_MMFR[OP]. 01 Write operation 10 Read operation
PHYADR (5 bits)	Five-bit PHY device address, programmed with ENET*n*_MMFR[PA]. Up to 32 devices can be addressed.
REGADR (5 bits)	Five-bit register address, programmed with ENET*n*_MMFR[RA]. Each PHY can implement up to 32 registers.
TA (2 bits)	Turnaround time, programmed with ENET*n*_MMFR[TA]. Two bit-times are reserved for read operations to switch the data bus from write to read. The PHY device presents its register contents in the data phase and drives the bus from the second bit of the turnaround phase.
Data (16 bits)	Data, set by ENET*n*_MMFR[DATA]. Written to or read from the PHY

11.6.2.17.2 MDIO clause 45 frame format

The extended MDIO frame structure defined in IEEE 802.3 Clause 45 introduces indirect addressing. First, a write transaction to an address register is done, followed by a write or read transaction which will put the 16-bit data in the register or retrieve the register contents respectively. A preamble of 32 bits of logical ones is sent prior to every transaction. The MDIO data signal is tri-stated between frames.

The extended MDIO defines four transactions, which are determined by the two-bit opcode field.

Table 11-87. MDIO clause 45 frame structure

ST	OP	PRTAD	DEVAD	TA	ADDR/DATA

All bits are transmitted from left to right (Preamble bits first) and all fields have their Most-Significant bit sent first (leftmost in above table). The complete frame has a length of 64 bits (32-bit preamble, 14-bit command, 2-bit bus direction change, 16-bit data). Each bit is transferred with the rising edge of the MDIO clock (MDC).

The fields and transactions are summarized in the following tables.

Table 11-88. MDIO clause 45 frame field descriptions

Field	Description
ST	Start indication. Indicates the end of the preamble and start of the frame. This value is 00 for extended MDIO (Clause 45) frames.
OP	Opcode defines if a read or write operation is performed and is programmed with ENET*n*_MMFR[OP]. See Table 11-89 for more information. 00 Address write 01 Write operation 10 Read inc. operation 11 Read operation
PRTAD	The port address specifies a MDIO port. Each Port can have up to 32 devices which each can have a separate set of registers.
DEVAD	Device address. Up to 32 devices can be addressed (within a port).
TA	Turnaround time, programmed with ENET*n*_MMFR[TA]. Two bit-times are reserved for read operations to switch the data bus from write to read. The PHY device presents its register contents in the data phase and drives the bus from the second bit of the turnaround phase.
ADDR/DATA	16-bit address (for address write) or data, set by ENET*n*_MMFR[DATA], written to or read from the PHY.

Table 11-89. MDIO Clause 45 Transactions

Transaction Type	Description
Address	A write transaction to the internal address register of the device/port. The data section of the frame contains the value to be stored in the device's internal address "pointer" register for further transactions.
Write	Data write to a register. The 16 bit data will be written to the register identified by the device-internal address.
Read	Data is read from the register identified by the device-internal address.
Read inc.	Read with address postincrement. The register identified by the device-internal address is read. After this, the device-internal address is incremented. If the address register is all '1' (0xFFFF) no increment is done (i.e. increment does not wrap around).

11.6.2.17.3 MDIO clock generation

The MDC clock is generated from the internal bus clock (i.e., IPS bus clock) divided by the value programmed in ENET*n*_MSCR[MII_SPEED].

11.6.2.17.4 MDIO operation

To perform an MDIO access, set the MDIO command register (ENET*n*_MMFR) according to the description provided in MII Management Frame Register (ENETn_MMFR).

To check when the programmed access completes, read the ENET*n*_EIR[MII] field.

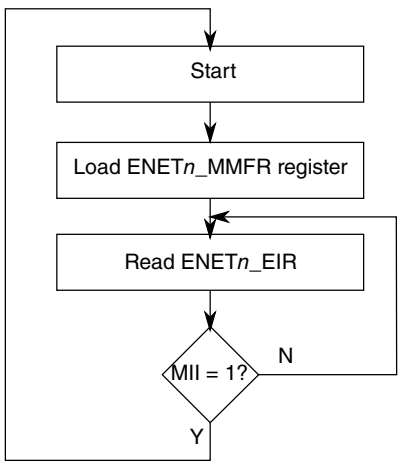


Figure 11-77. MDIO access overview

11.6.2.18 Ethernet interfaces

RMII 10/100 is implemented using interface converters/gaskets.

The following table shows how to configure ENET registers to select each interface.

Mode	ECR[SPEED]	RCR[RMII_10T]	RCR[RMII_MODE]	RCR[RGMII_EN]
RMII - 10 Mbit/s	0	1	1	0
RMII - 100 Mbit/s	0	0	1	0
RGMII - 10 Mbit/s	0	1	0	1
RGMII - 100 Mbit/s	0	0	0	1
RGMII - 1000 Mbit/s	1	—	0	1

11.6.2.18.1 RMII interface

In RMII receive mode, for normal reception following assertion of CRS_DV, RXD[1:0] is 00 until the receiver determines that the receive event has a proper start-of-stream delimiter (SSD).

The preamble appears (RXD[1:0]=01) and the MACs begin capturing data following detection of SFD.

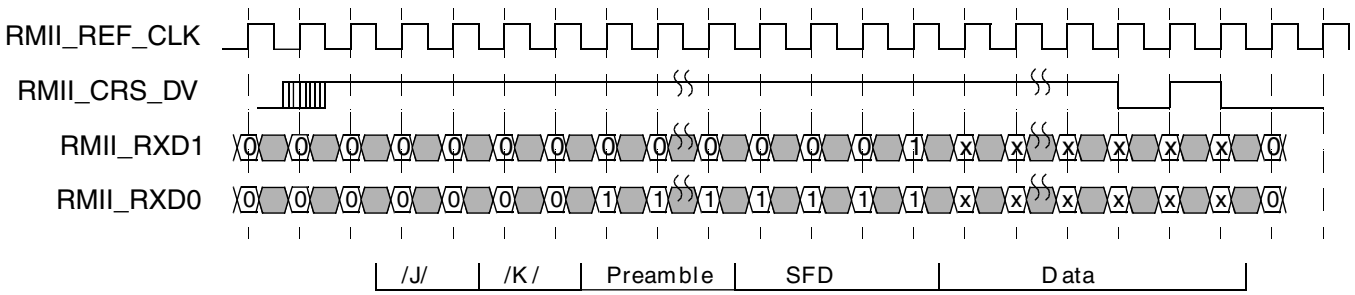


Figure 11-78. RMII receive operation

If a false carrier is detected (bad SSD), then RXD[1:0] is 10 until the end of the receive event. This is a unique pattern since a false carrier can only occur at the beginning of a packet where the preamble is decoded (RXD[1:0] = 01).

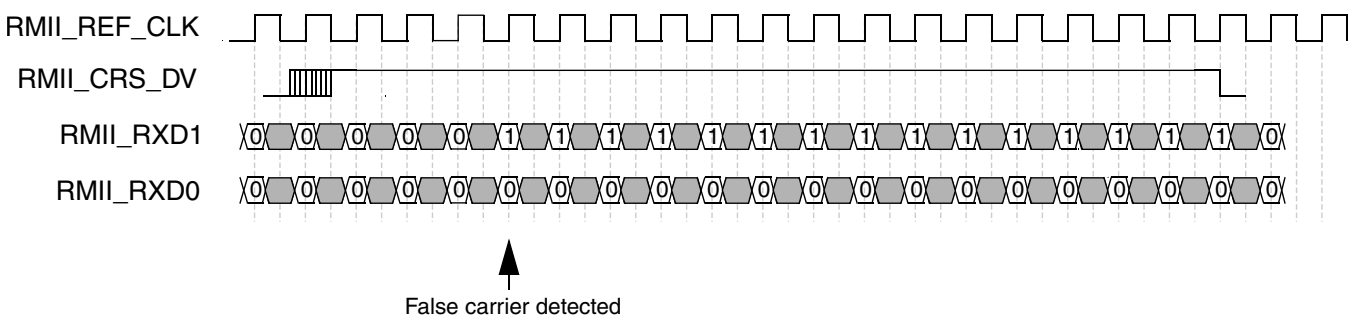


Figure 11-79. RMII receive operation with false carrier

In RMII transmit mode, TXD[1:0] provides valid data for each REF_CLK period while TXEN is asserted.

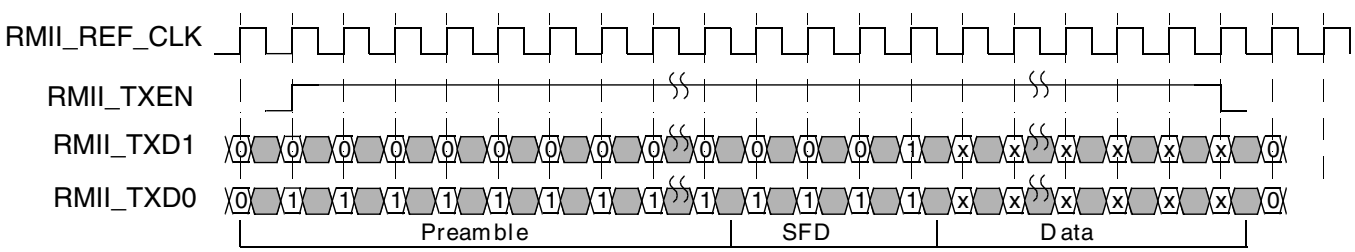


Figure 11-80. RMII transmit operation

11.6.2.18.2 RGMII interface

In RGMII modes, the data and control information is multiplexed by taking advantage of both edges of the reference clocks.

The data signals contain the lower four data bits on the rising edge and the upper four bits on the falling edge. The control signals are multiplexed into a single clock cycle using the same technique.

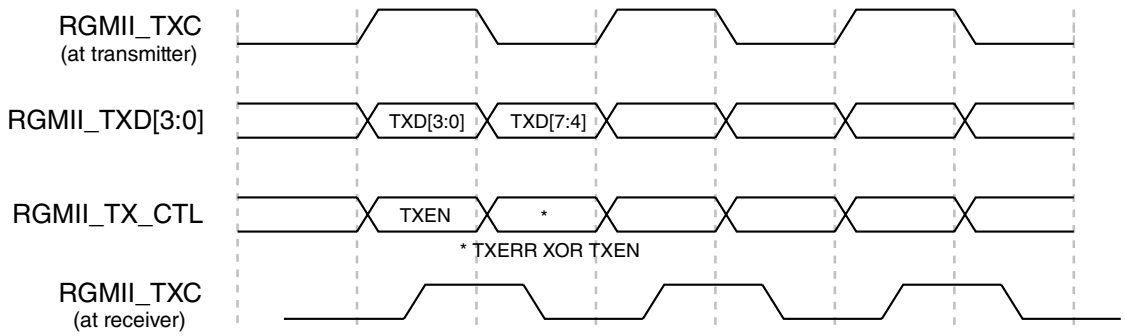


Figure 11-81. RGMII transmit operation

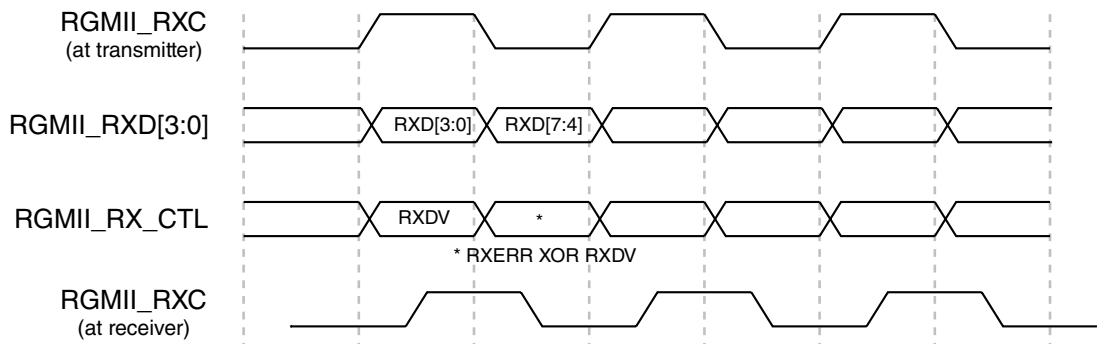


Figure 11-82. RGMII receive operation

11.6.2.19 AVB configuration

The following steps give an example of how to initialize the ENET module for AVB.

1. Set up ENET_QOS:
   - Set TX_SCHEME to 000b, credit-based scheme.
   - Set RX_FLUSH0 to 1, enable RX flush for ring 0.
2. Set up TX BD ring and RX BD ring for each queue, 0-2.
   - Program ENET_MRBR, ENET_TDSR, and ENET_RDSR.
   - Program ENET_MRBR1, ENET_TDSR1, and ENET_RDSR1.
   - Program ENET_MRBR2, ENET_TDSR2, and ENET_RDSR2.
   - Program each TX and RX BD ring queue for all classes used in memory.

NOTE

If using credit-based scheme, ensure that enhanced transmit buffer descriptor FTYPE field matches BD ring queue, for example:

- FTYPE = 0h corresponds to ENET_TDSR

- FTYPE = 1h corresponds to ENET_TDSR1
- FTYPE = 2h corresponds to ENET_TDSR2

3. Program ENET_DMA1CFG and ENET_DMA2CFG for class 1 and class 2 corresponding to BD ring queue 1 and 2. DMA_CLASS_EN must be set to one for that class to be enabled. See DMACFG register for information on how to program IDLE_SLOPE.
4. Program ENET_RCMR1 and ENET_RCMR2 for class 1 and class 2 matching for receive.

NOTE

Even if a match occurs, if ENET_DMA*n*CFG[DMA_CLASS_EN] is zero for the corresponding class, the RX frame will be automatically forwarded to BD ring queue 0.

5. Program the other ENET registers according to application requirements.
6. Program ENET_RDAR, ENET_RDAR1, ENET_RDAR2, ENET_TDAR, ENET_TDAR1, and ENET_TDAR2 to 0100_0000h according to the classes used.
7. Set ENET_ECR[ETHEREN] to one.

11.6.2.20 Interrupt coalescence

The purpose of the interrupt coalescing is to reduce the number of interrupts generated by the MAC so as to reduce the CPU loading.

To facilitate this interrupt coalescing for each queue, these registers are available with the same control and configuration fields.

- Transmit Interrupt Coalescing Register (TXICn) where n=0,1,2 for queue/class 0,1,2.
- Receive Interrupt Coalescing Register (RXICn) where n=0,1,2 for queue/class 0,1,2.

When coalescing is enabled by asserting the corresponding ICEN field and such interrupt is also enabled by the corresponding interrupt mask of the EIMR register, the MAC generates an interrupt when the threshold number of frames is reached (defined by ICFT) or when the threshold timer expires (defined by ICTT).

When coalescing is disabled by de-asserting ICEN, but interrupt is enabled by the corresponding interrupt mask of the EIMR register, the MAC generates an interrupt as they are received without using coalescing. Interrupt coalescing is done for each transmit and receive queue/class independently.

11.6.2.20.1 Interrupt coalescence setup

Interrupt coalescence supports both legacy and enhanced BDs. The following guidelines are recommended when setting up interrupt coalescence.

- When the MAC is configured for enhanced (IEEE 1588) mode, that is, enhanced BDs:
  - Set the INT bit in the enhanced received buffer descriptor to one.
  - Set the INT bit in the enhanced transmit buffer descriptor(s) to one.
- Clear the RXB, RXB1, and RXB2 fields in the EIMR register.
- Clear the TXB, TXB1, and TXB2 fields in the EIMR register.

11.6.2.20.2 Updating the frame count threshold on-the-fly

To update the ICFT field in the RXIC*n* and TXIC*n* registers:

1. Disable interrupt coalescence by clearing the appropriate ICEN field. This will allow the internal interrupt coalescence counter to reset to zero.

NOTE

When disabling interrupt coalescence, if an interrupt event is pending, that is, the interrupt counter is not zero, then an interrupt will occur.

2. Write the new threshold value to the ICFT field.
3. Set ICEN to one.

NOTE

The ICFT field can be updated on-the-fly without disabling the ICEN field. The hardware interrupt will continue and there is a possibility that an interrupt will occur depending on the state of the hardware counter and the previous ICFT value.

11.6.2.20.3 Updating the timer threshold on-the-fly

To update the ICTT field in the RXIC*n* and TXIC*n* registers:

1. Disable interrupt coalescence by clearing the appropriate ICEN field. This will allow the internal interrupt coalescence counter to reset to zero.

NOTE

When disabling interrupt coalescence, if an interrupt event is pending, that is, the interrupt counter is not zero, then an interrupt will occur.

2. Write the new timer value to the ICTT field.
3. Set ICEN to one.

11.6.2.21 Clocks

The table found here describes the clock sources for ENET. Please see the chip-specific clocking section for clock setting, configuration and gating information.

Table 11-90. Clocks

Clock name	Description
ipg_clk	Module clock
ipg_clk_rmii	Module clock for RMII
ipg_clk_s	Peripheral access clock
ipg_clk_time (ts_clk)	Peripheral clock
txc_sampling_clk	Transmit sampling clock

11.6.3 External Signals

The table found here describes the external signals of ENET.

Table 11-91. ENET External Signals

Signal	Description	Mode	Direction
1588_EVENT_IN	Capture/compare block input/ output event bus signal. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.	RMII	I
1588_EVENT_OUT	Capture/compare block input/ output event bus signal. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn	RMII	O

Table continues on the next page...

Table 11-91. ENET External Signals (continued)

Signal	Description	Mode	Direction
	register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set. NOTE: ENET_1588_EVENT0_OUT has a programmable output width, see IOMUXC_GPR[CLK_STRETCH], delayed one clock cycle in relation to all other EVENTx_OUT signals.		
MDC	Output clock provides a timing reference to the PHY for data transfers on the MDIO signal.	RMII	O
MDIO	Transfers control information between the external PHY and the media access controller. Data is synchronous to MDC. This signal is an input after reset.	RMII	IO
RX_ER	When asserted with RXDV, indicates the PHY detects an error in the current frame.	RMII	I
RX_EN	Asserting this input indicates the PHY has valid nibbles present on the MII. RX_EN must remain asserted from the first recovered nibble of the frame through to the last nibble. Asserting RX_EN must start no later than the SFD and exclude any EOF. In RMII mode, this pin also generates the CRS signal.	RMII	I
TX_ER	When asserted for one or more clock cycles while TXEN is also asserted, PHY sends one or more illegal symbols.	RMII	O
TX_EN	Indicates when valid nibbles are present on the MII. This signal is asserted with the first nibble of a preamble and is deasserted before the first	RMII	I

Table continues on the next page...

Table 11-91. ENET External Signals (continued)

Signal	Description	Mode	Direction
	TX_CLK following the final nibble of the frame.		
RDATA	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	RMII	I
TDATA	Serial output Ethernet data. Only valid during TX_EN assertion.	RMII	O
RGMII_RD0	Receive data bit 0 (received first)	RGMII	I
RGMII_RD1	Receive data bit 1	RGMII	I
RGMII_RD2	Receive data bit 2	RGMII	I
RGMII_RD3	Receive data bit 3	RGMII	I
RGMII_RX_CTL	Asserting this input indicates the PHY has valid nibbles present on the MII. RXDV must remain asserted from the first recovered nibble of the frame through to the last nibble. Asserting RXDV must start no later than the SFD and exclude any EOF.In RGMII mode, contains RXDV on the rising edge of RGMII_RXC, and a logical derivative of RXDV and RXERR (RXDV XOR RXERR) on the falling edge of RGMII_RXC.	RGMII	I
RGMII_RXC	Timing reference for RGMII_RXD[3:0] and RGMII_RX_CTL	RGMII	I
RGMII_TD0	Transmit data bit 0 (transmitted first)	RGMII	O
RGMII_TD1	Transmit data bit 1	RGMII	O
RGMII_TD2	Transmit data bit 2	RGMII	O
RGMII_TD3	Transmit data bit 3	RGMII	O
RGMII_TX_CTL	Indicates when valid nibbles are present on the MII. This signal is asserted with the first nibble of a preamble and is deasserted before the first TXCLK following the final nibble of the frame. In RGMII mode, contains TXEN on the rising edge of RGMII_TXC, and a logical derivative of TXEN and TXERR (TXEN	RGMII	O

Table continues on the next page...

Table 11-91. ENET External Signals (continued)

Signal	Description	Mode	Direction
	XOR TXERR) on the falling edge of RGMII_TXC.		
RGMII_TXC	Timing reference for RGMII_TXD[3:0] and RGMII_TX_CTL.	RGMII	O
REF_CLK1	In RMII mode, this signal is the reference clock for receive, transmit, and the control interface.	RMII	IO

11.6.4 Memory map/register definition

ENET registers must be read or written with 32-bit accesses. Non-32 bit accesses will terminate with an error.

Reserved bits should be written with 0 and ignored on read. Unused registers read zero and a write has no effect.

This table shows Ethernet registers organization.

Table 11-92. Register map summary

Offset Address	Section	Description
0x0000 – 0x01FF	Configuration	Core control and status registers
0x0200 – 0x03FF	Statistics counters	MIB and Remote Network Monitoring (RFC 2819) registers
0x0400 – 0x0430	1588 control	1588 adjustable timer (TSM) and 1588 frame control
0x0600 – 0x07FC	Capture/Compare block	Registers for the Capture/Compare block

11.6.4.1 ENET register descriptions

11.6.4.1.1 ENET memory map

ENET base address: 30BE_0000h

ENET1G base address: 4042_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	Interrupt Event Register (EIR)	32	WOZ	0000_0000
8	Interrupt Mask Register (EIMR)	32	RW	0000_0000
10	Receive Descriptor Active Register - Ring 0 (RDAR)	32	RW	0000_0000
14	Transmit Descriptor Active Register - Ring 0 (TDAR)	32	RW	0000_0000
24	Ethernet Control Register (ECR)	32	RW	7000_0000
40	MII Management Frame Register (MMFR)	32	RW	0000_0000
44	MII Speed Control Register (MSCR)	32	RW	0000_0000
64	MIB Control Register (MIBC)	32	RW	C000_0000
84	Receive Control Register (RCR)	32	RW	05EE_0001
C4	Transmit Control Register (TCR)	32	RW	0000_0000
E4	Physical Address Lower Register (PALR)	32	RW	0000_0000
E8	Physical Address Upper Register (PAUR)	32	RW	0000_8808
EC	Opcode/Pause Duration Register (OPD)	32	RW	0001_0000
F0 - F8	Transmit Interrupt Coalescing Register (TXIC0 - TXIC2)	32	RW	0000_0000
100 - 108	Receive Interrupt Coalescing Register (RXIC0 - RXIC2)	32	RW	0000_0000
118	Descriptor Individual Upper Address Register (IAUR)	32	RW	0000_0000
11C	Descriptor Individual Lower Address Register (IALR)	32	RW	0000_0000
120	Descriptor Group Upper Address Register (GAUR)	32	RW	0000_0000
124	Descriptor Group Lower Address Register (GALR)	32	RW	0000_0000
144	Transmit FIFO Watermark Register (TFWR)	32	RW	0000_0000
160	Receive Descriptor Ring 1 Start Register (RDSR1)	32	RW	0000_0000
164	Transmit Buffer Descriptor Ring 1 Start Register (TDSR1)	32	RW	0000_0000
168	Maximum Receive Buffer Size Register - Ring 1 (MRBR1)	32	RW	0000_0000
16C	Receive Descriptor Ring 2 Start Register (RDSR2)	32	RW	0000_0000
170	Transmit Buffer Descriptor Ring 2 Start Register (TDSR2)	32	RW	0000_0000
174	Maximum Receive Buffer Size Register - Ring 2 (MRBR2)	32	RW	0000_0000
180	Receive Descriptor Ring 0 Start Register (RDSR)	32	RW	0000_0000
184	Transmit Buffer Descriptor Ring 0 Start Register (TDSR)	32	RW	0000_0000
188	Maximum Receive Buffer Size Register - Ring 0 (MRBR)	32	RW	0000_0000
190	Receive FIFO Section Full Threshold (RSFL)	32	RW	0000_0000
194	Receive FIFO Section Empty Threshold (RSEM)	32	RW	0000_0000
198	Receive FIFO Almost Empty Threshold (RAEM)	32	RW	0000_0004
19C	Receive FIFO Almost Full Threshold (RAFL)	32	RW	0000_0004
1A0	Transmit FIFO Section Empty Threshold (TSEM)	32	RW	0000_0000
1A4	Transmit FIFO Almost Empty Threshold (TAEM)	32	RW	0000_0004
1A8	Transmit FIFO Almost Full Threshold (TAFL)	32	RW	0000_0008
1AC	Transmit Inter-Packet Gap (TIPG)	32	RW	0000_000C
1B0	Frame Truncation Length (FTRL)	32	RW	0000_07FF
1C0	Transmit Accelerator Function Configuration (TACC)	32	RW	0000_0000
1C4	Receive Accelerator Function Configuration (RACC)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1C8 - 1CC	Receive Classification Match Register for Class n (RCMR1 - RCMR2)	32	RW	0000_0000
1D8 - 1DC	DMA Class Based Configuration (DMA1CFG - DMA2CFG)	32	RW	0000_0000
1E0	Receive Descriptor Active Register - Ring 1 (RDAR1)	32	RW	0000_0000
1E4	Transmit Descriptor Active Register - Ring 1 (TDAR1)	32	RW	0000_0000
1E8	Receive Descriptor Active Register - Ring 2 (RDAR2)	32	RW	0000_0000
1EC	Transmit Descriptor Active Register - Ring 2 (TDAR2)	32	RW	0000_0000
1F0	QOS Scheme (QOS)	32	RW	0000_0000
204	Tx Packet Count Statistic Register (RMON_T_PACKETS)	32	RO	0000_0000
208	Tx Broadcast Packets Statistic Register (RMON_T_BC_PKT)	32	RO	0000_0000
20C	Tx Multicast Packets Statistic Register (RMON_T_MC_PKT)	32	RO	0000_0000
210	Tx Packets with CRC/Align Error Statistic Register (RMON_T_CRC_ALIGN)	32	RO	0000_0000
214	Tx Packets Less Than Bytes and Good CRC Statistic Register (RMON_T_UNDERSIZE)	32	RO	0000_0000
218	Tx Packets GT MAX_FL bytes and Good CRC Statistic Register (RMON_T_OVERSIZE)	32	RO	0000_0000
21C	Tx Packets Less Than 64 Bytes and Bad CRC Statistic Register (RMON_T_FRAG)	32	RO	0000_0000
220	Tx Packets Greater Than MAX_FL bytes and Bad CRC Statistic Register (RMON_T_JAB)	32	RO	0000_0000
224	Tx Collision Count Statistic Register (RMON_T_COL)	32	RO	0000_0000
228	Tx 64-Byte Packets Statistic Register (RMON_T_P64)	32	RO	0000_0000
22C	Tx 65- to 127-byte Packets Statistic Register (RMON_T_P65TO127)	32	RO	0000_0000
230	Tx 128- to 255-byte Packets Statistic Register (RMON_T_P128TO255)	32	RO	0000_0000
234	Tx 256- to 511-byte Packets Statistic Register (RMON_T_P256TO511)	32	RO	0000_0000
238	Tx 512- to 1023-byte Packets Statistic Register (RMON_T_P512TO1023)	32	RO	0000_0000
23C	Tx 1024- to 2047-byte Packets Statistic Register (RMON_T_P1024TO2047)	32	RO	0000_0000
240	Tx Packets Greater Than 2048 Bytes Statistic Register (RMON_T_P_GTE2048)	32	RO	0000_0000
244	Tx Octets Statistic Register (RMON_T_OCTETS)	32	RO	0000_0000
24C	Frames Transmitted OK Statistic Register (IEEE_T_FRAME_OK)	32	RO	0000_0000
250	Frames Transmitted with Single Collision Statistic Register (IEEE_T_1COL)	32	RO	0000_0000
254	Frames Transmitted with Multiple Collisions Statistic Register (IEEE_T_MCOL)	32	RO	0000_0000
258	Frames Transmitted after Deferral Delay Statistic Register (IEEE_T_DEF)	32	RO	0000_0000
25C	Frames Transmitted with Late Collision Statistic Register (IEEE_T_LCOL)	32	RO	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
260	Frames Transmitted with Excessive Collisions Statistic Register (IEEE_T_EXCOL)	32	RO	0000_0000
264	Frames Transmitted with Tx FIFO Underrun Statistic Register (IEEE_T_MACERR)	32	RO	0000_0000
268	Frames Transmitted with Carrier Sense Error Statistic Register (IEEE_T_CSERR)	32	RO	0000_0000
26C	Reserved Statistic Register (IEEE_T_SQE)	32	RO	0000_0000
270	Flow Control Pause Frames Transmitted Statistic Register (IEEE_T_FDXFC)	32	RO	0000_0000
274	Octet Count for Frames Transmitted w/o Error Statistic Register (IEEE_T_OCTETS_OK)	32	RO	0000_0000
284	Rx Packet Count Statistic Register (RMON_R_PACKETS)	32	RO	0000_0000
288	Rx Broadcast Packets Statistic Register (RMON_R_BC_PKT)	32	RO	0000_0000
28C	Rx Multicast Packets Statistic Register (RMON_R_MC_PKT)	32	RO	0000_0000
290	Rx Packets with CRC/Align Error Statistic Register (RMON_R_CRC_ALIGN)	32	RO	0000_0000
294	Rx Packets with Less Than 64 Bytes and Good CRC Statistic Register (RMON_R_UNDERSIZE)	32	RO	0000_0000
298	Rx Packets Greater Than MAX_FL and Good CRC Statistic Register (RMON_R_OVERSIZE)	32	RO	0000_0000
29C	Rx Packets Less Than 64 Bytes and Bad CRC Statistic Register (RMON_R_FRAG)	32	RO	0000_0000
2A0	Rx Packets Greater Than MAX_FL Bytes and Bad CRC Statistic Register (RMON_R_JAB)	32	RO	0000_0000
2A8	Rx 64-Byte Packets Statistic Register (RMON_R_P64)	32	RO	0000_0000
2AC	Rx 65- to 127-Byte Packets Statistic Register (RMON_R_P65TO127)	32	RO	0000_0000
2B0	Rx 128- to 255-Byte Packets Statistic Register (RMON_R_P128TO255)	32	RO	0000_0000
2B4	Rx 256- to 511-Byte Packets Statistic Register (RMON_R_P256TO511)	32	RO	0000_0000
2B8	Rx 512- to 1023-Byte Packets Statistic Register (RMON_R_P512TO1023)	32	RO	0000_0000
2BC	Rx 1024- to 2047-Byte Packets Statistic Register (RMON_R_P1024TO2047)	32	RO	0000_0000
2C0	Rx Packets Greater than 2048 Bytes Statistic Register (RMON_R_P_GTE2048)	32	RO	0000_0000
2C4	Rx Octets Statistic Register (RMON_R_OCTETS)	32	RO	0000_0000
2C8	Frames not Counted Correctly Statistic Register (IEEE_R_DROP)	32	RO	0000_0000
2CC	Frames Received OK Statistic Register (IEEE_R_FRAME_OK)	32	RO	0000_0000
2D0	Frames Received with CRC Error Statistic Register (IEEE_R_CRC)	32	RO	0000_0000
2D4	Frames Received with Alignment Error Statistic Register (IEEE_R_ALIGN)	32	RO	0000_0000
2D8	Receive FIFO Overflow Count Statistic Register (IEEE_R_MACERR)	32	RO	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2DC	Flow Control Pause Frames Received Statistic Register (IEEE_R_FDXFC)	32	RO	0000_0000
2E0	Octet Count for Frames Received without Error Statistic Register (IEEE_R_OCTETS_OK)	32	RO	0000_0000
400	Adjustable Timer Control Register (ATCR)	32	RW	0000_0000
404	Timer Value Register (ATVR)	32	RW	0000_0000
408	Timer Offset Register (ATOFF)	32	RW	0000_0000
40C	Timer Period Register (ATPER)	32	RW	3B9A_CA00
410	Timer Correction Register (ATCOR)	32	RW	0000_0000
414	Time-Stamping Clock Period Register (ATINC)	32	RW	0000_0000
418	Timestamp of Last Transmitted Frame (ATSTMP)	32	RO	0000_0000
604	Timer Global Status Register (TGSR)	32	W1C	0000_0000
608	Timer Control Status Register (TCSR0)	32	RW	0000_0000
60C	Timer Compare Capture Register (TCCR0)	32	RW	0000_0000
610	Timer Control Status Register (TCSR1)	32	RW	0000_0000
614	Timer Compare Capture Register (TCCR1)	32	RW	0000_0000
618	Timer Control Status Register (TCSR2)	32	RW	0000_0000
61C	Timer Compare Capture Register (TCCR2)	32	RW	0000_0000
620	Timer Control Status Register (TCSR3)	32	RW	0000_0000
624	Timer Compare Capture Register (TCCR3)	32	RW	0000_0000

11.6.4.1.2 Interrupt Event Register (EIR)

When an event occurs that sets a bit in EIR, an interrupt occurs if the corresponding bit in the interrupt mask register (EIMR) is also set. Writing a 1 to an EIR bit clears it; writing 0 has no effect. This register is cleared upon hardware reset.

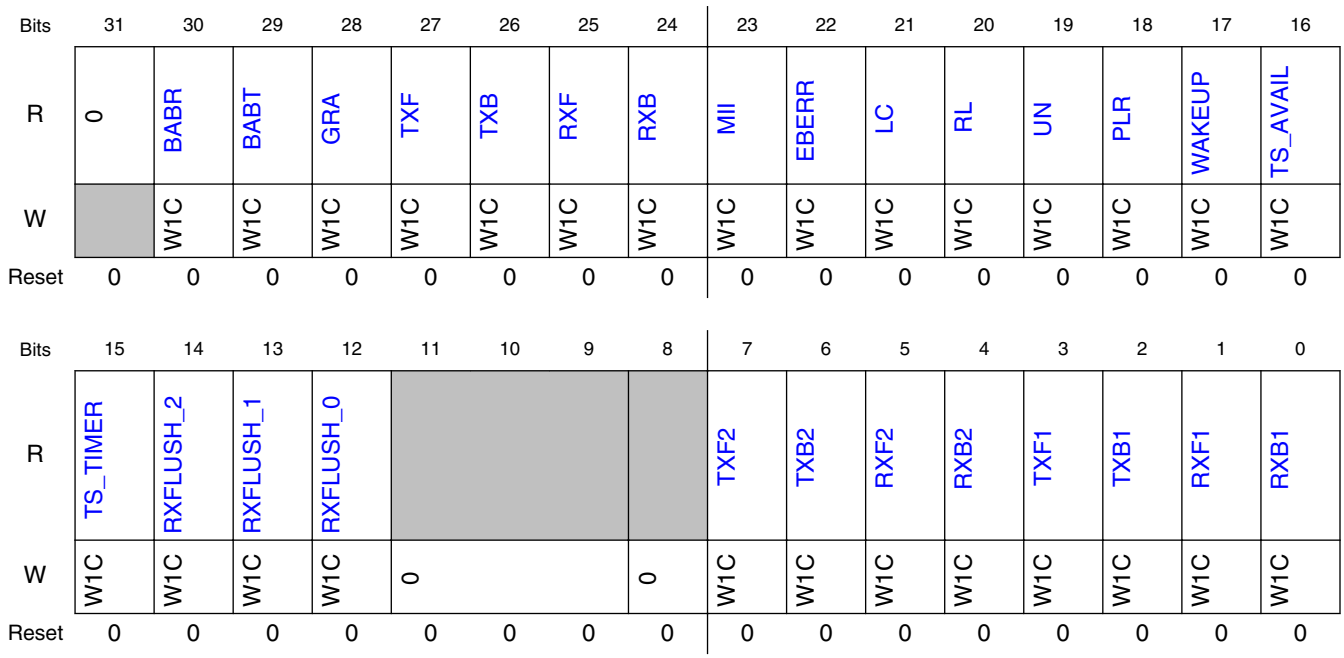
NOTE

TxBD[INT] and RxBD[INT] must be set to 1 to allow setting the corresponding EIR register flags in enhanced mode, ENET_ECR[EN1588] = 1. Legacy mode does not require these flags to be enabled.

11.6.4.1.2.1 Offset

Register	Offset
EIR	4h

11.6.4.1.2.2 Diagram



11.6.4.1.2.3 Fields

Field	Description
31 —	Reserved
30 BABR	Babbling Receive Error Indicates a frame was received with length in excess of RCR[MAX_FL] bytes.
29 BABT	Babbling Transmit Error Indicates the transmitted frame length exceeds RCR[MAX_FL] bytes. Usually this condition is caused when a frame that is too long is placed into the transmit data buffer(s). Truncation does not occur.
28 GRA	Graceful Stop Complete This interrupt is asserted after the transmitter is put into a pause state after completion of the frame currently being transmitted. See Graceful Transmit Stop (GTS) for conditions that lead to graceful stop. **NOTE:** The GRA interrupt is asserted only when the TX transitions into the stopped state. If this bit is cleared by writing 1 and the TX is still stopped, the bit is not set again.
27 TXF	Transmit Frame Interrupt Indicates a frame has been transmitted and the last corresponding buffer descriptor has been updated.
26 TXB	Transmit Buffer Interrupt Indicates a transmit buffer descriptor has been updated.
25 RXF	Receive Frame Interrupt Indicates a frame has been received and the last corresponding buffer descriptor has been updated.

Table continues on the next page...

Field	Description
24 RXB	Receive Buffer Interrupt Indicates a receive buffer descriptor is not the last in the frame has been updated.
23 MII	MII Interrupt. Indicates that the MII has completed the data transfer requested.
22 EBERR	Ethernet Bus Error Indicates a system bus error occurred when a uDMA transaction is underway. When this bit is set, ECR[ETHEREN] is cleared, halting frame processing by the MAC. When this occurs, software must ensure proper actions, possibly resetting the system, to resume normal operation.
21 LC	Late Collision Indicates a collision occurred beyond the collision window (slot time) in half-duplex mode. The frame truncates with a bad CRC and the remainder of the frame is discarded.
20 RL	Collision Retry Limit Indicates a collision occurred on each of 16 successive attempts to transmit the frame. The frame is discarded without being transmitted and transmission of the next frame commences. This error can only occur in half-duplex mode.
19 UN	Transmit FIFO Underrun Indicates the transmit FIFO became empty before the complete frame was transmitted. **NOTE:** In situations where the device has various masters generating high traffic, a FIFO underrun can occur on the transmit FIFO. To avoid transmit FIFO underrun, store and forward can be enabled in ENET_TFWR[STRFWD]. See Transmit FIFO Watermark Register (TFWR). Also, a higher priority can be set for ENET traffic using available means on the central bus fabric connecting the ENET module.
18 PLR	Payload Receive Error Indicates a frame was received with a payload length error. See Frame Length/Type Verification: Payload Length Check for more information.
17 WAKEUP	Node Wakeup Request Indication Read-only status bit to indicate that a magic packet has been detected. Will act only if ECR[MAGICEN] is set.
16 TS_AVAIL	Transmit Timestamp Available Indicates that the timestamp of the last transmitted timing frame is available in the ATSTMP register.
15 TS_TIMER	Timestamp Timer The adjustable timer reached the period event. A period event interrupt can be generated if ATCR[PEREN] is set and the timer wraps according to the periodic setting in the ATPER register. Set the timer period value before setting ATCR[PEREN].
14 RXFLUSH_2	RX DMA Ring 2 flush indication. This ring's RX frame has been flushed due to either RDAR2[RDAR] or RxBD[E] being clear and only if QOS[RXFLUSH_2] is enabled. **NOTE:** This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET / — / EIR ENET1G / EIR / —

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>13
RXFLUSH_1</td><td>RX DMA Ring 1 flush indication. This ring's RX frame has been flushed due to either RDAR1[RDAR] or RxBD[E] being clear and only if QOS[RXFLUSH_1] is enabled.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>ENET</td><td>—</td><td>EIR</td></tr><tr><td>ENET1G</td><td>EIR</td><td>—</td></tr></table></td></tr>
<tr><td>12
RXFLUSH_0</td><td>RX DMA Ring 0 flush indication. . This ring's RX frame has been flushed due to either RDAR[RDAR] or RxBD[E] being clear and only if QOS[RXFLUSH_0] is enabled.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>ENET</td><td>—</td><td>EIR</td></tr><tr><td>ENET1G</td><td>EIR</td><td>—</td></tr></table></td></tr>
<tr><td>11-9
—</td><td>This write-only field is reserved. It must always be written with the value 0.</td></tr>
<tr><td>8
—</td><td>This write-only field is reserved. It must always be written with the value 0.</td></tr>
<tr><td>7
TXF2</td><td>Transmit frame interrupt, class 2
This bit indicates that a frame has been transmitted and the last corresponding buffer descriptor has been updated (ring/class 2).
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>ENET</td><td>—</td><td>EIR</td></tr><tr><td>ENET1G</td><td>EIR</td><td>—</td></tr></table></td></tr>
<tr><td>6
TXB2</td><td>Transmit buffer interrupt, class 2
This field indicates that a transmit buffer descriptor has been updated (ring/class 2).
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>ENET</td><td>—</td><td>EIR</td></tr><tr><td>ENET1G</td><td>EIR</td><td>—</td></tr></table></td></tr>
<tr><td>5</td><td>Receive frame interrupt, class 2</td></tr>
</table>

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>RXF2</td><td>This field indicates that a frame has been received and the last corresponding buffer descriptor has been updated (ring/class 2).

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIR</td></tr>
<tr><td>ENET1G</td><td>EIR</td><td>—</td></tr>
</table>
</td></tr>
<tr><td>4

RXB2</td><td>Receive buffer interrupt, class 2

This field indicates that a receive buffer descriptor, that not the last in the frame, has been updated (ring/class 2).

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIR</td></tr>
<tr><td>ENET1G</td><td>EIR</td><td>—</td></tr>
</table>
</td></tr>
<tr><td>3

TXF1</td><td>Transmit frame interrupt, class 1

This bit indicates that a frame has been transmitted and the last corresponding buffer descriptor has been updated (ring/class 1).

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIR</td></tr>
<tr><td>ENET1G</td><td>EIR</td><td>—</td></tr>
</table>
</td></tr>
<tr><td>2

TXB1</td><td>Transmit buffer interrupt, class 1

This field indicates that a transmit buffer descriptor has been updated (ring/class 1).

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIR</td></tr>
<tr><td>ENET1G</td><td>EIR</td><td>—</td></tr>
</table>
</td></tr>
<tr><td>1

RXF1</td><td>Receive frame interrupt, class 1

This field indicates that a frame has been received and the last corresponding buffer descriptor has been updated (ring/class 1).</td></tr>
</table>

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIR</td></tr>
<tr><td>ENET1G</td><td>EIR</td><td>—</td></tr>
</table>
</td></tr>
<tr><td>0

RXB1</td><td>Receive buffer interrupt, class 1

This field indicates that a receive buffer descriptor, that not the last in the frame, has been updated (ring/class 1).

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIR</td></tr>
<tr><td>ENET1G</td><td>EIR</td><td>—</td></tr>
</table>
</td></tr>
</table>

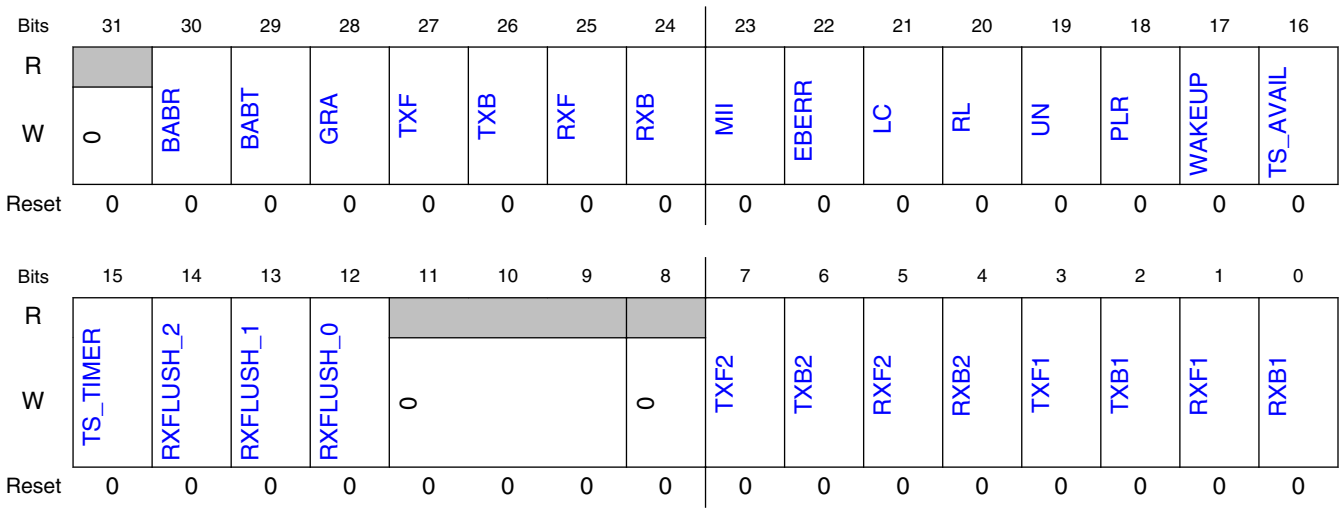
11.6.4.1.3 Interrupt Mask Register (EIMR)

EIMR controls which interrupt events are allowed to generate actual interrupts. A hardware reset clears this register. If the corresponding bits in the EIR and EIMR registers are set, an interrupt is generated. The interrupt signal remains asserted until a 1 is written to the EIR field (write 1 to clear) or a 0 is written to the EIMR field.

11.6.4.1.3.1 Offset

Register	Offset
EIMR	8h

11.6.4.1.3.2 Diagram



11.6.4.1.3.3 Fields

Field	Description
31 —	This write-only field is reserved. It must always be written with the value 0.
30 BABR	BABR Interrupt Mask Corresponds to interrupt source EIR[BABR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
29 BABT	BABT Interrupt Mask Corresponds to interrupt source EIR[BABT] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABT field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
28 GRA	GRA Interrupt Mask Corresponds to interrupt source EIR[GRA] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR GRA field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
27 TXF	TXF Interrupt Mask

Table continues on the next page...

Field	Description
	Corresponds to interrupt source EIR[TXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
26 TXB	TXB Interrupt Mask Corresponds to interrupt source EIR[TXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
25 RXF	RXF Interrupt Mask Corresponds to interrupt source EIR[RXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
24 RXB	RXB Interrupt Mask Corresponds to interrupt source EIR[RXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RXB field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
23 MII	MII Interrupt Mask Corresponds to interrupt source EIR[MII] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR MII field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
22 EBERR	EBERR Interrupt Mask Corresponds to interrupt source EIR[EBERR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR EBERR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
21 LC	LC Interrupt Mask Corresponds to interrupt source EIR[LC] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR LC field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

Table continues on the next page...

Field	Description
	0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
20 RL	RL Interrupt Mask Corresponds to interrupt source EIR[RL] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
19 UN	UN Interrupt Mask Corresponds to interrupt source EIR[UN] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR UN field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
18 PLR	PLR Interrupt Mask Corresponds to interrupt source EIR[PLR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR PLR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
17 WAKEUP	WAKEUP Interrupt Mask Corresponds to interrupt source EIR[WAKEUP] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR WAKEUP field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
16 TS_AVAIL	TS_AVAIL Interrupt Mask Corresponds to interrupt source EIR[TS_AVAIL] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TS_AVAIL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.
15 TS_TIMER	TS_TIMER Interrupt Mask Corresponds to interrupt source EIR[TS_TIMER] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TS_TIMER field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 - The corresponding interrupt source is masked. 1 - The corresponding interrupt source is not masked.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>14
RXFLUSH_2</td><td>Corresponds to interrupt source EIR[RXFLUSH_2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXFLUSH_2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET1G</td><td>EIMR</td><td>—</td></tr>
</table>

0 - The corresponding interrupt source is masked.

1 - The corresponding interrupt source is not masked.</td></tr>
<tr><td>13
RXFLUSH_1</td><td>Corresponds to interrupt source EIR[RXFLUSH_1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXFLUSH_1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET1G</td><td>EIMR</td><td>—</td></tr>
</table>

0 - The corresponding interrupt source is masked.

1 - The corresponding interrupt source is not masked.</td></tr>
<tr><td>12
RXFLUSH_0</td><td>Corresponds to interrupt source EIR[RXFLUSH_0] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXFLUSH_0] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET1G</td><td>EIMR</td><td>—</td></tr>
</table>

0 - The corresponding interrupt source is masked.

1 - The corresponding interrupt source is not masked.</td></tr>
<tr><td>11-9
—</td><td>This write-only field is reserved. It must always be written with the value 0.</td></tr>
<tr><td>8</td><td>This write-only field is reserved. It must always be written with the value 0.</td></tr>
</table>

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>—</td><td></td></tr>
<tr><td>7
TXF2</td><td>Transmit frame interrupt, class 2
Corresponds to interrupt source EIR[TXF2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXF2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET1G</td><td>EIMR</td><td>—</td></tr>
</table>
0 - The corresponding interrupt source is masked.
1 - The corresponding interrupt source is not masked.</td></tr>
<tr><td>6
TXB2</td><td>Transmit buffer interrupt, class 2
Corresponds to interrupt source EIR[TXB2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXB2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET1G</td><td>EIMR</td><td>—</td></tr>
</table>
0 - The corresponding interrupt source is masked.
1 - The corresponding interrupt source is not masked.</td></tr>
<tr><td>5
RXF2</td><td>Receive frame interrupt, class 2
Corresponds to interrupt source EIR[RXF2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXF2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET1G</td><td>EIMR</td><td>—</td></tr>
</table>
0 - The corresponding interrupt source is masked.
1 - The corresponding interrupt source is not masked.</td></tr>
<tr><td>4</td><td>Receive buffer interrupt, class 2</td></tr>
</table>

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>RXB2</td><td>Corresponds to interrupt source EIR[RXB2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXB2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET1G</td><td>EIMR</td><td>—</td></tr>
</table>

0 - The corresponding interrupt source is masked.

1 - The corresponding interrupt source is not masked.</td></tr>
<tr><td>3

TXF1</td><td>Transmit frame interrupt, class 1

Corresponds to interrupt source EIR[TXF1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXF1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET1G</td><td>EIMR</td><td>—</td></tr>
</table>

0 - The corresponding interrupt source is masked.

1 - The corresponding interrupt source is not masked.</td></tr>
<tr><td>2

TXB1</td><td>Transmit buffer interrupt, class 1

Corresponds to interrupt source EIR[TXB1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXB1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET1G</td><td>EIMR</td><td>—</td></tr>
</table>

0 - The corresponding interrupt source is masked.

1 - The corresponding interrupt source is not masked.</td></tr>
<tr><td>1

RXF1</td><td>Receive frame interrupt, class 1

Corresponds to interrupt source EIR[RXF1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXF1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.</td></tr>
</table>

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET1G</td><td>EIMR</td><td>—</td></tr>
</table>
0 - The corresponding interrupt source is masked.
1 - The corresponding interrupt source is not masked.
</td></tr>
<tr><td>0
RXB1</td><td>
Receive buffer interrupt, class 1
Corresponds to interrupt source EIR[RXB1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXB1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET1G</td><td>EIMR</td><td>—</td></tr>
</table>
0 - The corresponding interrupt source is masked.
1 - The corresponding interrupt source is not masked.
</td></tr>
</table>

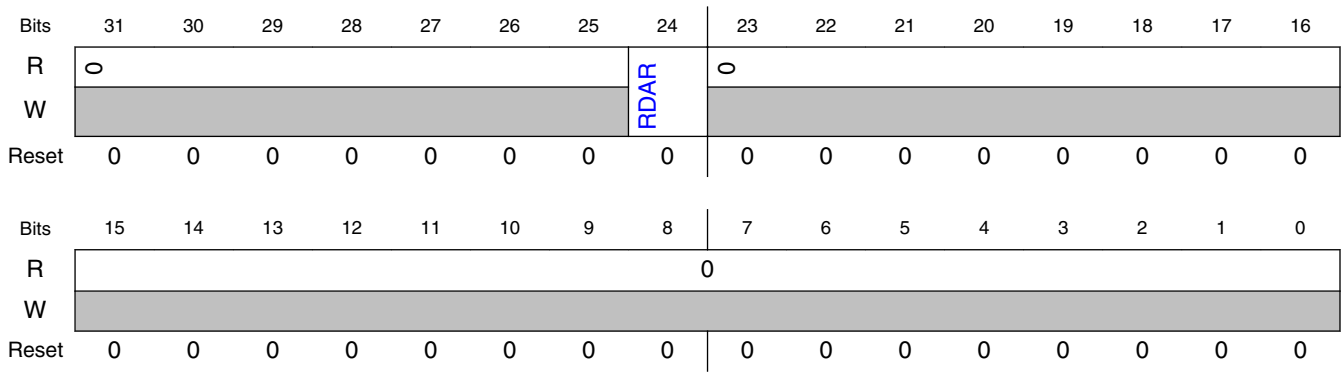
11.6.4.1.4 Receive Descriptor Active Register - Ring 0 (RDAR)

RDAR is a command register, written by the user, to indicate that the receive descriptor ring 0 has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

11.6.4.1.4.1 Offset

Register	Offset
RDAR	10h

11.6.4.1.4.2 Diagram



11.6.4.1.4.3 Fields

Field	Description
31-25 —	Reserved
24 RDAR	Receive Descriptor Active Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

11.6.4.1.5 Transmit Descriptor Active Register - Ring 0 (TDAR)

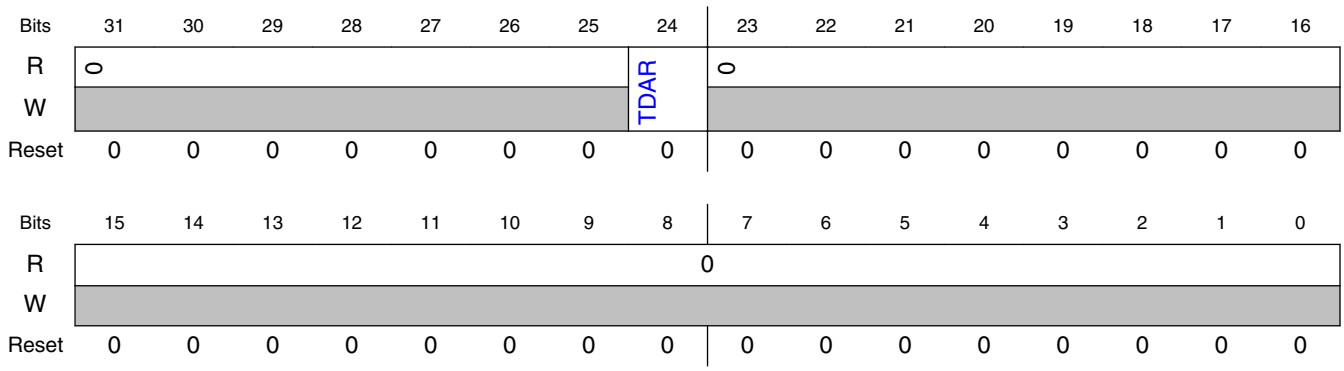
The TDAR is a command register that the user writes to indicate that the transmit descriptor ring 0 has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

The TDAR register is cleared at reset, when ECR[ETHEREN] transitions from set to cleared, or when ECR[RESET] is set.

11.6.4.1.5.1 Offset

Register	Offset
TDAR	14h

11.6.4.1.5.2 Diagram



11.6.4.1.5.3 Fields

Field	Description
31-25 —	Reserved
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

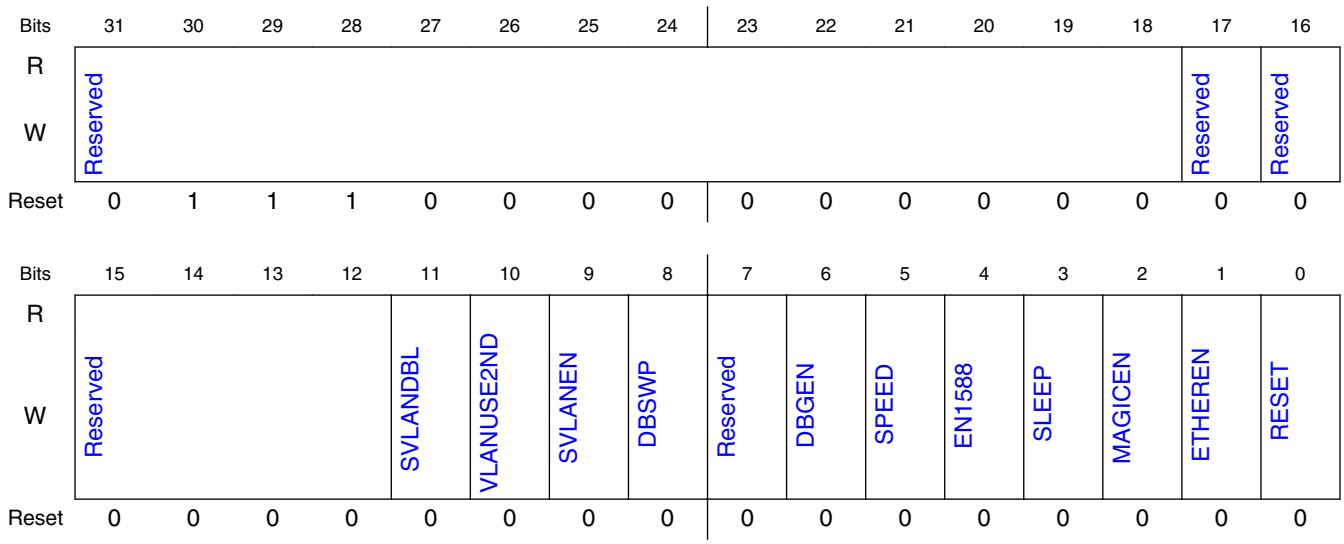
11.6.4.1.6 Ethernet Control Register (ECR)

ECR is a read/write user register, though hardware may also alter fields in this register. It controls many of the high level features of the Ethernet MAC, including legacy FEC support through the EN1588 field.

11.6.4.1.6.1 Offset

Register	Offset
ECR	24h

11.6.4.1.6.2 Diagram



11.6.4.1.6.3 Fields

Field	Description
31-18 —	Always write 01110000000000b to this field.
17 —	Always write 0 to this field.
16 —	Always write 0 to this field.
15-12 —	Always write 0 to this field.
11 SVLANDBL	S-VLAN double tag If enabled, S-VLAN detection requires a double-tagged frame to define a frame as being a VLAN frame. The following rules apply: • If the first tag is the S-VLAN type, it must be followed by a second tag with the C-VLAN type to declare the frame as VLAN frame. • If the first tag is the S-VLAN type but no 2nd tag follows, it is not considered a VLAN frame but instead treated as an untagged frame. • If the first tag is the C-VLAN type, it is considered a VLAN frame as normal. **NOTE:** VLANUSE2ND can be used to determine from which tag the data should be extracted. This applies only if SVLAN_EN = 1, ignored otherwise. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
ENET	—	ECR
ENET1G	ECR	—

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>0 - Disable S-VLAN double tag
1 - Enable S-VLAN double tag</td></tr>
<tr><td>10
VLANUSE2ND</td><td>VLAN use second tag
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>ECR</td></tr>
<tr><td>ENET1G</td><td>ECR</td><td>—</td></tr>
</table>
0 - Always extract data from the first VLAN tag if it exists.
1 - When a double-tagged frame is detected, the data of the second tag is extracted for further processing. A double-tagged frame is defined as: The first tag can be a C-VLAN or a S-VLAN (if SVLAN_ENA = 1) The second tag must be a C-VLAN</td></tr>
<tr><td>9
SVLANEN</td><td>S-VLAN enable
Enable additional detection of S-VLAN tag according to IEEE802.1Q.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>ECR</td></tr>
<tr><td>ENET1G</td><td>ECR</td><td>—</td></tr>
</table>
0 - Only the EtherType 0x8100 will be considered for VLAN detection.
1 - The EtherType 0x88a8 will be considered in addition to 0x8100 (C-VLAN) to identify a VLAN frame in receive. When a VLAN frame is identified, the two bytes following the VLAN type are extracted and used by the classification match comparators, RCMRn.</td></tr>
<tr><td>8
DBSWP</td><td>Descriptor Byte Swapping Enable
Swaps the byte locations of the buffer descriptors.
NOTE: This field resets to 0 and must not be changed.
0 - The buffer descriptor bytes are not swapped to support big-endian devices.
1 - The buffer descriptor bytes are swapped to support little-endian devices.</td></tr>
<tr><td>7
—</td><td>Always write 0 to this field.</td></tr>
<tr><td>6
DBGEN</td><td>Debug Enable
Enables the MAC to enter hardware freeze mode when the device enters debug mode.
0 - MAC continues operation in debug mode.
1 - MAC enters hardware freeze mode when the processor is in debug mode.</td></tr>
<tr><td>5
SPEED</td><td>Selects between 10/100-Mbit/s and 1000-Mbit/s modes of operation.</td></tr>
</table>

Table continues on the next page...

Field	Description
	NOTE: This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET / — / ECR ENET1G / ECR / — 0 - 10/100-Mbit/s mode 1 - 1000-Mbit/s mode
4 EN1588	EN1588 Enable Enables enhanced functionality of the MAC. 0 - Legacy FEC buffer descriptors and functions enabled. 1 - Enhanced frame time-stamping functions enabled. Has no effect within the MAC besides controlling the DMA control bit ena_1588.
3 SLEEP	Sleep Mode Enable 0 - Normal operating mode. 1 - Sleep mode.
2 MAGICEN	Magic Packet Detection Enable Enables/disables magic packet detection. **NOTE:** MAGICEN is relevant only if the SLEEP field is set. If MAGICEN is set, changing the SLEEP field enables/disables sleep mode and magic packet detection. **NOTE:** EIMR[WAKEUP] must be written to one if Magic packet wakeup is programed to wake up the chip from low power mode. 0 - Magic detection logic disabled. 1 - The MAC core detects magic packets and asserts EIR[WAKEUP] when a frame is detected.
1 ETHEREN	Ethernet Enable Enables/disables the Ethernet MAC. When the MAC is disabled, the buffer descriptors for an aborted transmit frame are not updated. The uDMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers. Hardware clears this field under the following conditions: • RESET is set by software • An error condition causes the EBERR field to set. **NOTE:** • ETHEREN must be set at the very last step during ENET configuration/setup/initialization, only *after* all other ENET-related registers have been configured. • If ETHEREN is cleared to 0 by software then next time ETHEREN is set, the EIR interrupts must cleared to 0 due to previous pending interrupts. 0 - Reception immediately stops and transmission stops after a bad CRC is appended to any currently transmitted frame. 1 - MAC is enabled, and reception and transmission are possible.
0 RESET	Ethernet MAC Reset When this field is set, it clears the ETHEREN field.

11.6.4.1.7 MII Management Frame Register (MMFR)

Writing to MMFR triggers a management frame transaction to the PHY device unless MSCR is programmed to zero.

If MSCR is changed from zero to non-zero during a write to MMFR, an MII frame is generated with the data previously written to the MMFR. This allows MMFR and MSCR to be programmed in either order if MSCR is currently zero.

If the MMFR register is written while frame generation is in progress, the frame contents are altered. Software must use the EIR[MII] interrupt indication to avoid writing to the MMFR register while frame generation is in progress.

11.6.4.1.7.1 Offset

Register	Offset
MMFR	40h

11.6.4.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ST		OP		PA					RA					TA	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.7.3 Fields

Field	Description
31-30 ST	Start Of Frame Delimiter See Table 11-85 (Clause 22) or Table 11-87 (Clause 45) for correct value.
29-28 OP	Operation Code See Table 11-85 (Clause 22) or Table 11-87 (Clause 45) for correct value.
27-23	PHY Address

Table continues on the next page...

Field	Description
PA	See Table 11-85 (Clause 22) or Table 11-87 (Clause 45) for correct value.
22-18 RA	Register Address See Table 11-85 (Clause 22) or Table 11-87 (Clause 45) for correct value.
17-16 TA	Turn Around This field must be programmed to 10 to generate a valid MII management frame.
15-0 DATA	Management Frame Data This is the field for data to be written to or read from the PHY register.

11.6.4.1.8 MII Speed Control Register (MSCR)

MSCR provides control of the MII clock (MDC pin) frequency and allows a preamble drop on the MII management frame.

The MII_SPEED field must be programmed with a value to provide an MDC frequency of less than or equal to 2.5 MHz to be compliant with the IEEE 802.3 MII specification. The MII_SPEED must be set to a non-zero value to source a read or write management frame. After the management frame is complete, the MSCR register may optionally be cleared to turn off MDC. The MDC signal generated has a 50% duty cycle except when MII_SPEED changes during operation. This change takes effect following a rising or falling edge of MDC.

For example, if the internal module clock (that is, peripheral bus clock) is 25 MHz, programming MII_SPEED to 0x4 results in an MDC as given in the following equation:

MII clock frequency = 25 MHz / ((4 + 1) x 2) = 2.5 MHz

The following table shows the optimum values for MII_SPEED as a function of IPS bus clock frequency.

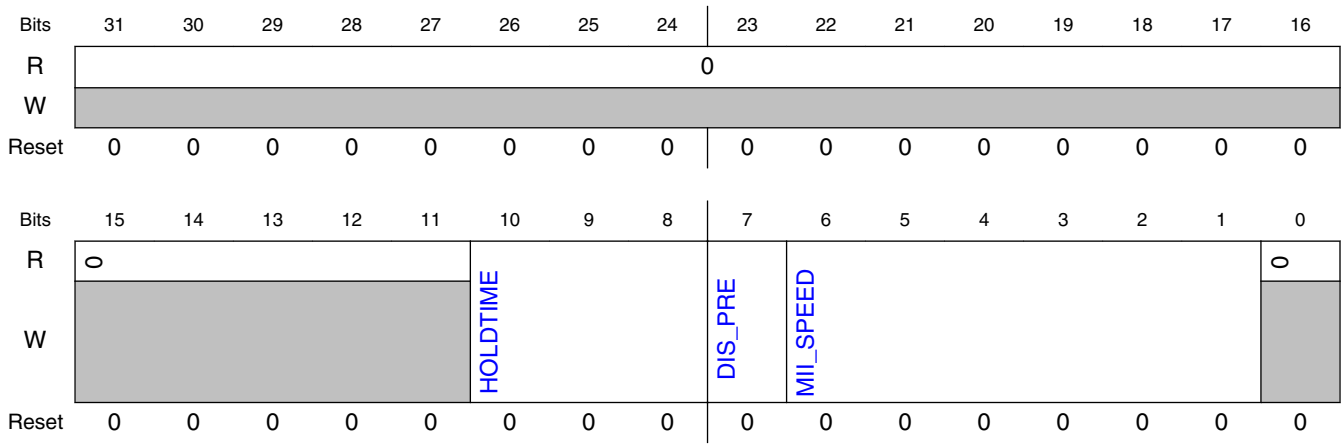
Table 11-93. Programming Examples for MSCR

Internal module clock frequency	MSCR [MII_SPEED]	MDC frequency
25 MHz	0x4	2.50 MHz
33 MHz	0x6	2.36 MHz
40 MHz	0x7	2.50 MHz
50 MHz	0x9	2.50 MHz
66 MHz	0xD	2.36 MHz

11.6.4.1.8.1 Offset

Register	Offset
MSCR	44h

11.6.4.1.8.2 Diagram



11.6.4.1.8.3 Fields

Field	Description
31-11 —	Reserved
10-8 HOLDTIME	Hold time On MDIO Output IEEE802.3 clause 22 defines a minimum of 10 ns for the hold time on the MDIO output. Depending on the host bus frequency, the setting may need to be increased. 000 - 1 internal module clock cycle 001 - 2 internal module clock cycles 010 - 3 internal module clock cycles 111 - 8 internal module clock cycles
7 DIS_PRE	Disable Preamble Enables/disables prepending a preamble to the MII management frame. The MII standard allows the preamble to be dropped if the attached PHY devices do not require it. 0 - Preamble enabled. 1 - Preamble (32 ones) is not prepended to the MII management frame.
6-1 MII_SPEED	MII Speed Controls the frequency of the MII management interface clock (MDC) relative to the internal module clock. A value of 0 in this field turns off MDC and leaves it in low voltage state. Any non-zero value results in the MDC frequency of:

Table continues on the next page...

Field	Description
	1/((MII_SPEED + 1) x 2) of the internal module clock frequency
0 —	Reserved

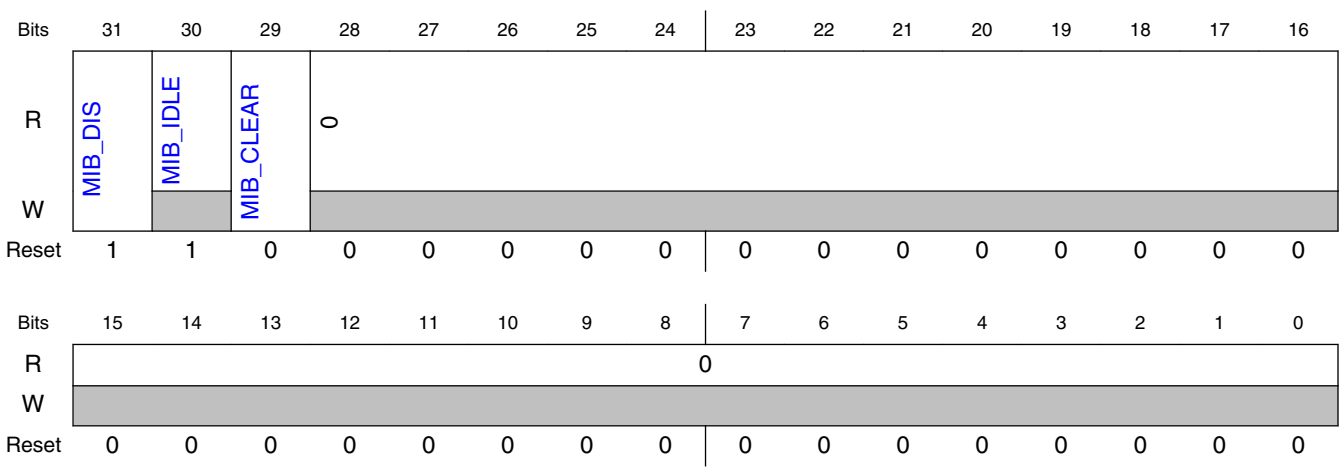
11.6.4.1.9 MIB Control Register (MIBC)

MIBC is a read/write register controlling and observing the state of the MIB block. Access this register to disable the MIB block operation or clear the MIB counters. The MIB_DIS field resets to 1.

11.6.4.1.9.1 Offset

Register	Offset
MIBC	64h

11.6.4.1.9.2 Diagram



11.6.4.1.9.3 Fields

Field	Description
31 MIB_DIS	Disable MIB Logic If this control field is set, 0 - MIB logic is enabled. 1 - MIB logic is disabled. The MIB logic halts and does not update any MIB counters.

Table continues on the next page...

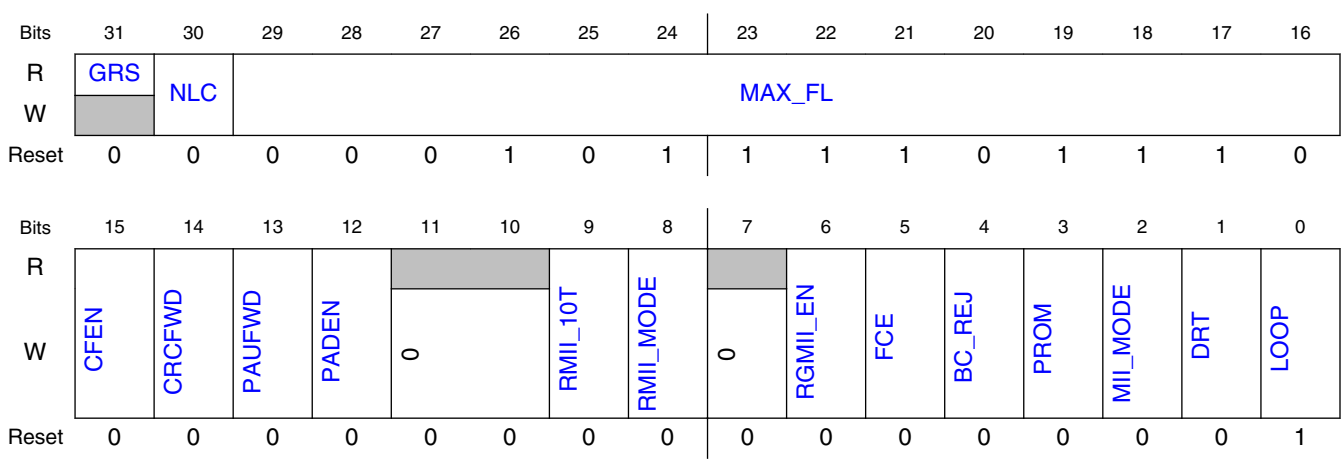
Field	Description
30 MIB_IDLE	MIB Idle 0 - The MIB block is updating MIB counters. 1 - The MIB block is not currently updating any MIB counters.
29 MIB_CLEAR	MIB Clear **NOTE:** This field is not self-clearing. To clear the MIB counters set and then clear this field. 0 - See note above. 1 - All statistics counters are reset to 0.
28-0 —	Reserved

11.6.4.1.10 Receive Control Register (RCR)

11.6.4.1.10.1 Offset

Register	Offset
RCR	84h

11.6.4.1.10.2 Diagram



11.6.4.1.10.3 Fields

Field	Description
31	Graceful Receive Stopped

Table continues on the next page...

Field	Description
GRS	Read-only status indicating that the MAC receive datapath is stopped. 0 - Receive not stopped 1 - Receive stopped
30 NLC	Payload Length Check Disable Enables/disables a payload length check. 0 - The payload length check is disabled. 1 - The core checks the frame's payload length with the frame length/type field. Errors are indicated in the EIR[PLR] field.
29-16 MAX_FL	Maximum Frame Length Resets to decimal 1518. Length is measured starting at DA and includes the CRC at the end of the frame. Transmit frames longer than MAX_FL cause the BABT interrupt to occur. Receive frames longer than MAX_FL cause the BABR interrupt to occur and set the LG field in the end of frame receive buffer descriptor. The recommended default value to be programmed is 1518 or 1522 if VLAN tags are supported.
15 CFEN	MAC Control Frame Enable Enables/disables the MAC control frame. 0 - MAC control frames with any opcode other than 0x0001 (pause frame) are accepted and forwarded to the client interface. 1 - MAC control frames with any opcode other than 0x0001 (pause frame) are silently discarded.
14 CRCFWD	Terminate/Forward Received CRC Specifies whether the CRC field of received frames is transmitted or stripped. **NOTE:** If padding function is enabled (PADEN = 1), CRCFWD is ignored and the CRC field is checked and always terminated and removed. 0 - The CRC field of received frames is transmitted to the user application. 1 - The CRC field is stripped from the frame.
13 PAUFWD	Terminate/Forward Pause Frames Specifies whether pause frames are terminated or forwarded. 0 - Pause frames are terminated and discarded in the MAC. 1 - Pause frames are forwarded to the user application.
12 PADEN	Enable Frame Padding Remove On Receive Specifies whether the MAC removes padding from received frames. 0 - No padding is removed on receive by the MAC. 1 - Padding is removed from received frames.
11-10 —	This write-only field is reserved. It must always be written with the value 0.
9 RMII_10T	Enables 10-Mbit/s mode of the RMII or RGMII . **NOTE:** The descriptions of the field settings vary by module instance. **Instance** / **Field value and description** ENET / 0 - 100-Mbit/s operation. 1 - 10-Mbit/s operation.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Instance</th><th>Field value and description</th></tr>
<tr><td>ENET1G</td><td>0 - 100-Mbit/s or 1-Gbit/s operation.
1 - 10-Mbit/s operation.</td></tr>
</table>
</td></tr>
<tr><td>8
RMII_MODE</td><td>RMII Mode Enable
Specifies whether the MAC is configured for MII mode or RMII operation , when ECR[SPEED] is cleared .
NOTE: Do not set both RCR[RGMII_EN] and RCR[RMII_MODE].
0 - MAC configured for MII mode.
1 - MAC configured for RMII operation.</td></tr>
<tr><td>7
—</td><td>This write-only field is reserved. It must always be written with the value 0.</td></tr>
<tr><td>6
RGMII_EN</td><td>RGMII Mode Enable
NOTE: Do not set both RCR[RGMII_EN] and RCR[RMII_MODE].
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>RCR</td></tr>
<tr><td>ENET1G</td><td>RCR</td><td>—</td></tr>
</table>
0 - MAC configured for non-RGMII operation
1 - MAC configured for RGMII operation. If ECR[SPEED] is set, the MAC is in RGMII 1000-Mbit/s mode. If ECR[SPEED] is cleared, the MAC is in RGMII 10/100-Mbit/s mode.</td></tr>
<tr><td>5
FCE</td><td>Flow Control Enable
If set, the receiver detects PAUSE frames. Upon PAUSE frame detection, the transmitter stops transmitting data frames for a given duration.
0 - Disable flow control
1 - Enable flow control</td></tr>
<tr><td>4
BC_REJ</td><td>Broadcast Frame Reject
If set, frames with destination address (DA) equal to 0xFFFF_FFFF_FFFF are rejected unless the PROM field is set. If BC_REJ and PROM are set, frames with broadcast DA are accepted and the MISS (M) is set in the receive buffer descriptor.
0 - Will not reject frames as described above
1 - Will reject frames as described above</td></tr>
<tr><td>3
PROM</td><td>Promiscuous Mode
All frames are accepted regardless of address matching.
0 - Disabled.
1 - Enabled.</td></tr>
<tr><td>2
MII_MODE</td><td>Media Independent Interface Mode
This field must always be set.
0 - Reserved.</td></tr>
</table>

Table continues on the next page...

Field	Description
	1 - MII or RMII mode, as indicated by the RMII_MODE field.
1 DRT	Disable Receive On Transmit 0 - Receive path operates independently of transmit (i.e., full-duplex mode). Can also be used to monitor transmit activity in half-duplex mode. 1 - Disable reception of frames while transmitting. (Normally used for half-duplex mode.)
0 LOOP	Internal Loopback This is an MII internal loopback, therefore MII_MODE must be written to 1 and RMII_MODE must be written to 0. 0 - Loopback disabled. 1 - Transmitted frames are looped back internal to the device and transmit MII output signals are not asserted. DRT must be cleared.

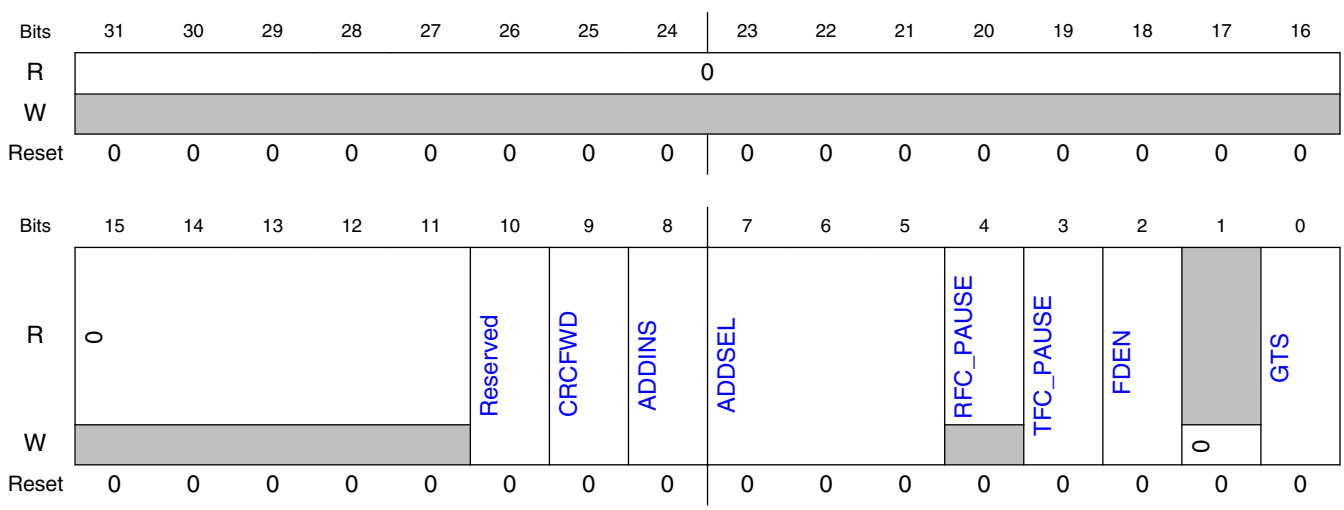
11.6.4.1.11 Transmit Control Register (TCR)

TCR is read/write and configures the transmit block. This register is cleared at system reset. FDEN can only be modified when ECR[ETHEREN] is cleared.

11.6.4.1.11.1 Offset

Register	Offset
TCR	C4h

11.6.4.1.11.2 Diagram



11.6.4.1.11.3 Fields

Field	Description
31-11 —	Reserved
10 —	This field is read/write and must be set to 0.
9 CRCFWD	Forward Frame From Application With CRC 0 - TxBD[TC] controls whether the frame has a CRC from the application. 1 - The transmitter does not append any CRC to transmitted frames, as it is expecting a frame with CRC from the application.
8 ADDINS	Set MAC Address On Transmit 0 - The source MAC address is not modified by the MAC. 1 - The MAC overwrites the source MAC address with the programmed MAC address according to ADDSEL.
7-5 ADDSEL	Source MAC Address Select On Transmit If ADDINS is set, indicates the MAC address that overwrites the source MAC address. 000 - Node MAC address programmed on PADDR1/2 registers. 100 - Reserved. 101 - Reserved. 110 - Reserved.
4 RFC_PAUSE	Receive Frame Control Pause This status field is set when a full-duplex flow control pause frame is received and the transmitter pauses for the duration defined in this pause frame. This field automatically clears when the pause duration is complete.
3 TFC_PAUSE	Transmit Frame Control Pause Pauses frame transmission. When this field is set, EIR[GRA] is set. With transmission of data frames stopped, the MAC transmits a MAC control PAUSE frame. Next, the MAC clears TFC_PAUSE and resumes transmitting data frames. If the transmitter pauses due to user assertion of GTS or reception of a PAUSE frame, the MAC may continue transmitting a MAC control PAUSE frame. 0 - No PAUSE frame transmitted. 1 - The MAC stops transmission of data frames after the current transmission is complete.
2 FDEN	Full-Duplex Enable If this field is set, frames transmit independent of carrier sense and collision inputs. Only modify this bit when ECR[ETHEREN] is cleared. 0 - Disable full-duplex 1 - Enable full-duplex
1 —	This write-only field is reserved. It must always be written with the value 0.
0 GTS	Graceful Transmit Stop When this field is set, MAC stops transmission after any frame currently transmitted is complete and EIR[GRA] is set. If frame transmission is not currently underway, the GRA interrupt is asserted immediately. After transmission finishes, clear GTS to restart. The next frame in the transmit FIFO is then transmitted. If an early collision occurs during transmission when GTS is set, transmission stops after the

Field	Description
	collision. The frame is transmitted again after GTS is cleared. There may be old frames in the transmit FIFO that transmit when GTS is reasserted. To avoid this, clear ECR[ETHEREN] following the GRA interrupt. 0 - Disable graceful transmit stop 1 - Enable graceful transmit stop

11.6.4.1.12 Physical Address Lower Register (PALR)

PALR contains the lower 32 bits (bytes 0, 1, 2, 3) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 0 through 3 of the six-byte source address field when transmitting PAUSE frames.

11.6.4.1.12.1 Offset

Register	Offset
PALR	E4h

11.6.4.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.12.3 Fields

Field	Description
31-0 PADDR1	Pause Address Bytes 0 (bits 31:24), 1 (bits 23:16), 2 (bits 15:8), and 3 (bits 7:0) of the 6-byte individual address are used for exact match and the source address field in PAUSE frames.

11.6.4.1.13 Physical Address Upper Register (PAUR)

PAUR contains the upper 16 bits (bytes 4 and 5) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 4 and 5 of the six-byte source address field when transmitting PAUSE frames. Bits 15:0 of PAUR contain a constant type field (0x8808) for transmission of PAUSE frames.

11.6.4.1.13.1 Offset

Register	Offset
PAUR	E8h

11.6.4.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TYPE															
W																
Reset	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

11.6.4.1.13.3 Fields

Field	Description
31-16 PADDR2	Bytes 4 (bits 31:24) and 5 (bits 23:16) of the 6-byte individual address used for exact match, and the source address field in PAUSE frames.
15-0 TYPE	Type Field In PAUSE Frames These fields have a constant value of 0x8808.

11.6.4.1.14 Opcode/Pause Duration Register (OPD)

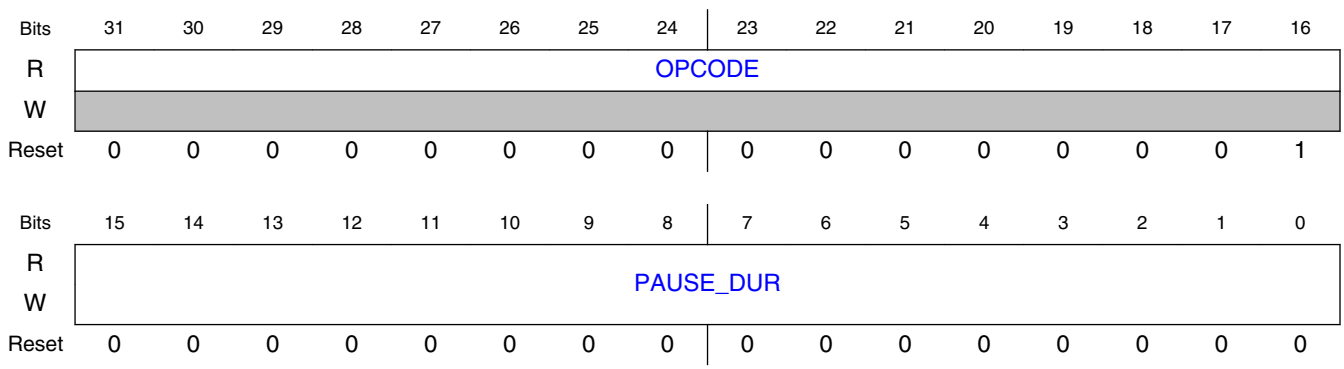
OPD is read/write accessible. This register contains the 16-bit opcode and 16-bit pause duration fields used in transmission of a PAUSE frame. The opcode field is a constant value, 0x0001. When another node detects a PAUSE frame, that node pauses

transmission for the duration specified in the pause duration field. The lower 16 bits of this register are not reset and you must initialize it.

11.6.4.1.14.1 Offset

Register	Offset
OPD	ECh

11.6.4.1.14.2 Diagram



11.6.4.1.14.3 Fields

Field	Description
31-16 OPCODE	Opcode Field In PAUSE Frames These fields have a constant value of 0x0001.
15-0 PAUSE_DUR	Pause Duration Pause duration field used in PAUSE frames.

11.6.4.1.15 Transmit Interrupt Coalescing Register (TXIC0 - TXIC2)

See Interrupt coalescence for more information.

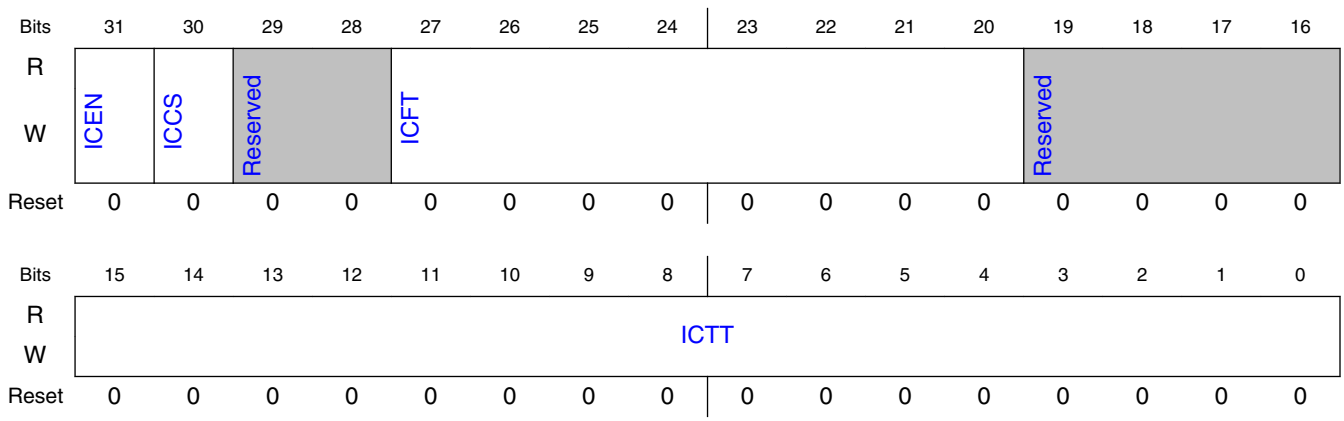
11.6.4.1.15.1 Offset

Register	Offset
TXIC0	F0h

Table continues on the next page...

Register	Offset
TXIC1	F4h
TXIC2	F8h

11.6.4.1.15.2 Diagram



11.6.4.1.15.3 Fields

Field	Description
31 ICEN	Interrupt Coalescing Enable 0 - Disable Interrupt coalescing. 1 - Enable Interrupt coalescing.
30 ICCS	Interrupt Coalescing Timer Clock Source Select 0 - Use MII/GMII TX clocks. 1 - Use ENET system clock.
29-28 —	This field must be set to 0.
27-20 ICFT	Interrupt coalescing frame count threshold This value determines the number of frames needed to be transmitted for raising an interrupt. Frame counter restarts after reaching this threshold value or after the expiring of the coalescing timer. Must be greater than zero to avoid unpredictable behavior.
19-16 —	This field must be set to 0.
15-0 ICTT	Interrupt coalescing timer threshold Interrupt coalescing timer threshold in units of 64 clock periods. This value determines the maximum amount of time after transmitting a frame before raising an interrupt. The threshold timer is disabled after expiring or number of frame transmission defined by ICFT and starts again upon transmission of the next first frame. Must be greater than zero to avoid unpredictable behavior.

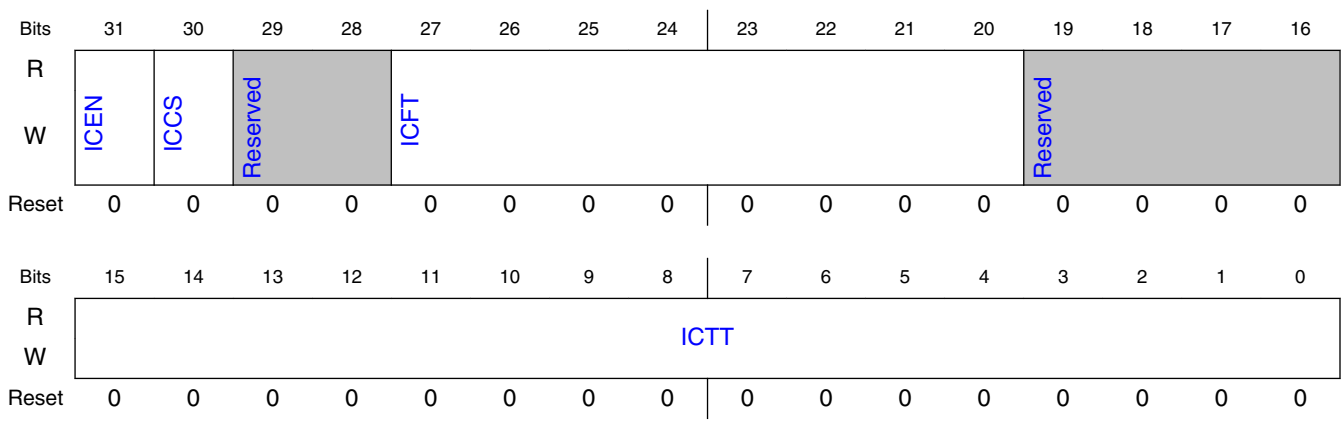
11.6.4.1.16 Receive Interrupt Coalescing Register (RXIC0 - RXIC2)

See Interrupt coalescence for more information.

11.6.4.1.16.1 Offset

Register	Offset
RXIC0	100h
RXIC1	104h
RXIC2	108h

11.6.4.1.16.2 Diagram



11.6.4.1.16.3 Fields

Field	Description
31 ICEN	Interrupt Coalescing Enable 0 - Disable Interrupt coalescing. 1 - Enable Interrupt coalescing.
30 ICCS	Interrupt Coalescing Timer Clock Source Select 0 - Use MII/GMII TX clocks. 1 - Use ENET system clock.
29-28 —	This field must be set to 0.
27-20	Interrupt coalescing frame count threshold

Table continues on the next page...

Field	Description
ICFT	This value determines the number of frames needed to be received for raising an interrupt. Frame counter restarts after reaching this threshold value or after the expiring of the coalescing timer. Must be greater than zero to avoid unpredictable behavior.
19-16 —	This field must be set to 0.
15-0 ICTT	Interrupt coalescing timer threshold Interrupt coalescing timer threshold in units of 64 clock periods. This value determines the maximum amount of time after receiving a frame before raising an interrupt. The threshold timer is disabled after expiring or number of frame reception defined by ICFT and starts again upon reception of the next first frame. Must be greater than zero to avoid unpredictable behavior.

11.6.4.1.17 Descriptor Individual Upper Address Register (IAUR)

IAUR contains the upper 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the destination address (DA) field of receive frames with an individual DA. This register is not reset and you must initialize it.

11.6.4.1.17.1 Offset

Register	Offset
IAUR	118h

11.6.4.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.17.3 Fields

Field	Description
31-0 IADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR1 contains hash index bit 63. Bit 0 of IADDR1 contains hash index bit 32.

11.6.4.1.18 Descriptor Individual Lower Address Register (IALR)

IALR contains the lower 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the DA field of receive frames with an individual DA. This register is not reset and you must initialize it.

11.6.4.1.18.1 Offset

Register	Offset
IALR	11Ch

11.6.4.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.18.3 Fields

Field	Description
31-0 IADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR2 contains hash index bit 31. Bit 0 of IADDR2 contains hash index bit 0.

11.6.4.1.19 Descriptor Group Upper Address Register (GAUR)

GAUR contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

11.6.4.1.19.1 Offset

Register	Offset
GAUR	120h

11.6.4.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.19.3 Fields

Field	Description
31-0 GADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR1 contains hash index bit 63. Bit 0 of GADDR1 contains hash index bit 32.

11.6.4.1.20 Descriptor Group Lower Address Register (GALR)

GALR contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

11.6.4.1.20.1 Offset

Register	Offset
GALR	124h

11.6.4.1.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.20.3 Fields

Field	Description
31-0 GADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR2 contains hash index bit 31. Bit 0 of GADDR2 contains hash index bit 0.

11.6.4.1.21 Transmit FIFO Watermark Register (TFWR)

If TFWR[STRFWD] is cleared, TFWR[TFWR] controls the amount of data required in the transmit FIFO before transmission of a frame can begin. This allows you to minimize transmit latency (TFWR = 00 or 01) or allow for larger bus access latency (TFWR = 11) due to contention for the system bus. Setting the watermark to a high value minimizes the risk of transmit FIFO underrun due to contention for the system bus. The byte counts associated with the TFWR field may need to be modified to match a given system requirement, for example, worst-case bus access latency by the transmit data uDMA channel.

When the FIFO level reaches the value the TFWR field and when the STR_FWD is set to ‘0’, the MAC transmit control logic starts frame transmission even before the end-of-frame is available in the FIFO (cut-through operation).

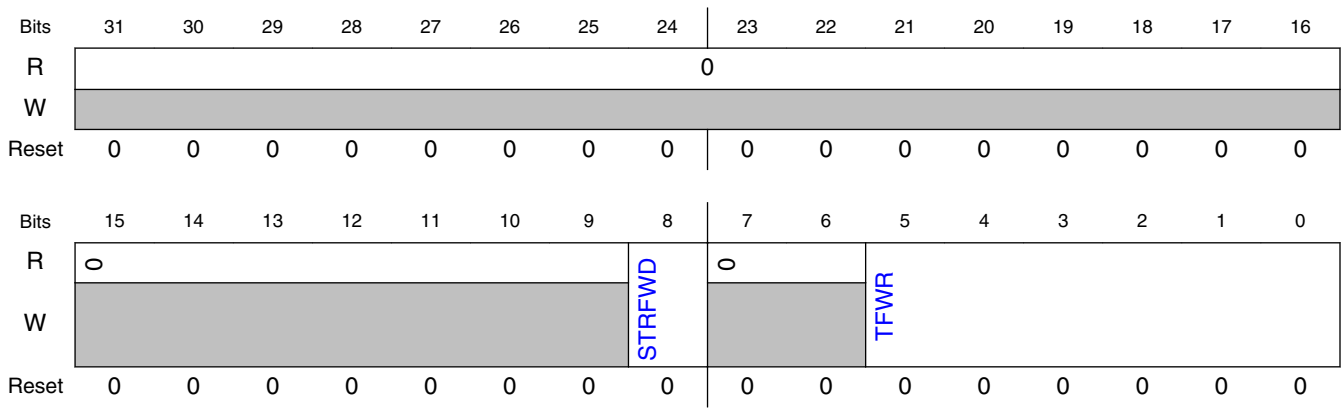
If a complete frame has a size smaller than the threshold programmed with TFWR, the MAC also transmits the Frame to the line.

To enable store and forward on the Transmit path, set STR_FWD to ‘1’. In this case, the MAC starts to transmit data only when a complete frame is stored in the Transmit FIFO.

11.6.4.1.21.1 Offset

Register	Offset
TFWR	144h

11.6.4.1.21.2 Diagram



11.6.4.1.21.3 Fields

Field	Description
31-9 —	Reserved
8 STRFWD	Store And Forward Enable 0 - Reset. The transmission start threshold is programmed in TFWR[TFWR]. 1 - Enabled.
7-6 —	Reserved
5-0 TFWR	Transmit FIFO Write If TFWR[STRFWD] is cleared, this field indicates the number of bytes, in steps of 64 bytes, written to the transmit FIFO before transmission of a frame begins. **NOTE:** If a frame with less than the threshold is written, it is still sent independently of this threshold setting. The threshold is relevant only if the frame is larger than the threshold given. 000000 - 64 bytes written. 000001 - 64 bytes written. 000010 - 128 bytes written. 000011 - 192 bytes written. 011111 - 1984 bytes written. 111111 - 4032 bytes written.

11.6.4.1.22 Receive Descriptor Ring 1 Start Register (RDSR1)

RDSR1 points to the beginning of circular receive buffer descriptor queue 1 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation.

11.6.4.1.22.1 Offset

Register	Offset
RDSR1	160h

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	RDSR1
ENET1G	RDSR1	—

11.6.4.1.22.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	R_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.22.3 Fields

Field	Description
31-3 R_DES_START	Pointer to the beginning of the receive buffer descriptor queue 1.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

11.6.4.1.23 Transmit Buffer Descriptor Ring 1 Start Register (TDSR1)

TDSR1 provides a pointer to the beginning of the circular transmit buffer descriptor queue 1 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation.

11.6.4.1.23.1 Offset

Register	Offset
TDSR1	164h

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	TDSR1
ENET1G	TDSR1	—

11.6.4.1.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	X_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	X_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.23.3 Fields

Field	Description
31-3 X_DES_START	Pointer to the beginning of transmit buffer descriptor queue 1.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

11.6.4.1.24 Maximum Receive Buffer Size Register - Ring 1 (MRBR1)

MRBR1 is a user-programmable register that dictates the maximum size of all ring-1 receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- To allow one maximum size frame per buffer, MRBR1 must be set to RCR[MAX_FL] or larger.
- R_BUF_SIZE is concatentated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To properly align the buffer, MRBR1 must be evenly divisible by 64. To ensure this, set the lower two bits of R_BUF_ZIZE to zero. The lower four bits are already set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR1 greater than or equal to 256 bytes.

NOTE

This register must be initialized before operation.

11.6.4.1.24.1 Offset

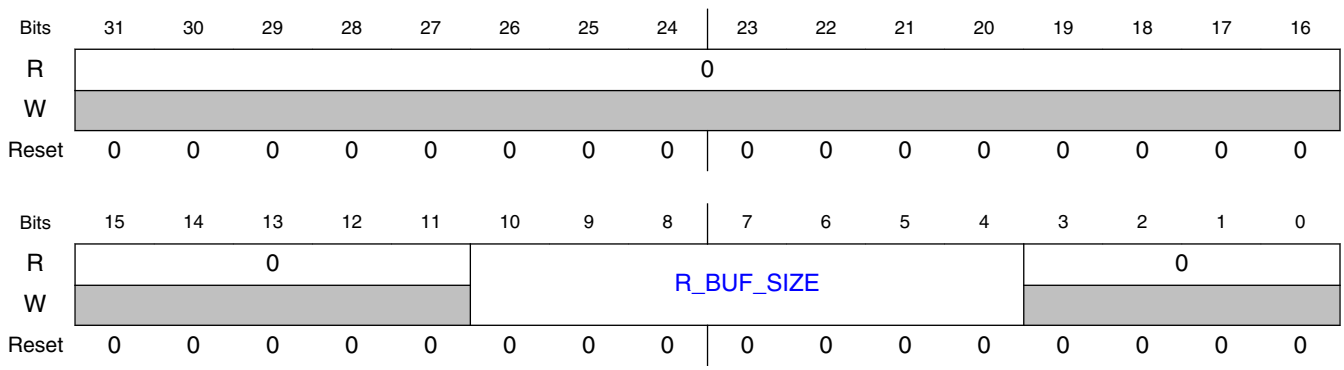
Register	Offset
MRBR1	168h

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	MRBR1
ENET1G	MRBR1	—

11.6.4.1.24.2 Diagram



11.6.4.1.24.3 Fields

Field	Description
31-11 —	Reserved
10-4 R_BUF_SIZE	Receive buffer size (in bytes)
3-0 —	Reserved

11.6.4.1.25 Receive Descriptor Ring 2 Start Register (RDSR2)

RDSR points to the beginning of circular receive buffer descriptor queue 2 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation

11.6.4.1.25.1 Offset

Register	Offset
RDSR2	16Ch

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	RDSR2
ENET1G	RDSR2	—

11.6.4.1.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	R_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.25.3 Fields

Field	Description
31-3	Pointer to the beginning of receive buffer descriptor queue 2.

Table continues on the next page...

Field	Description
R_DES_START	
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

11.6.4.1.26 Transmit Buffer Descriptor Ring 2 Start Register (TDSR2)

TDSR2 provides a pointer to the beginning of circular transmit buffer descriptor queue 2 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation

11.6.4.1.26.1 Offset

Register	Offset
TDSR2	170h

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	TDSR2
ENET1G	TDSR2	—

11.6.4.1.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	X_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	X_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.26.3 Fields

Field	Description
31-3 X_DES_START	Pointer to the beginning of transmit buffer descriptor queue 2.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

11.6.4.1.27 Maximum Receive Buffer Size Register - Ring 2 (MRBR2)

MRBR2 is a user-programmable register that dictates the maximum size of all ring-2 receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- To allow one maximum size frame per buffer, MRBR2 must be set to RCR[MAX_FL] or larger.
- R_BUF_SIZE is concatentated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To properly align the buffer, MRBR2 must be evenly divisible by 64. To ensure this, set the lower two bits of R_BUF_SIZE to zero. The lower four bits are already set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR2 greater than or equal to 256 bytes.

NOTE

This register must be initialized prior to operation

11.6.4.1.27.1 Offset

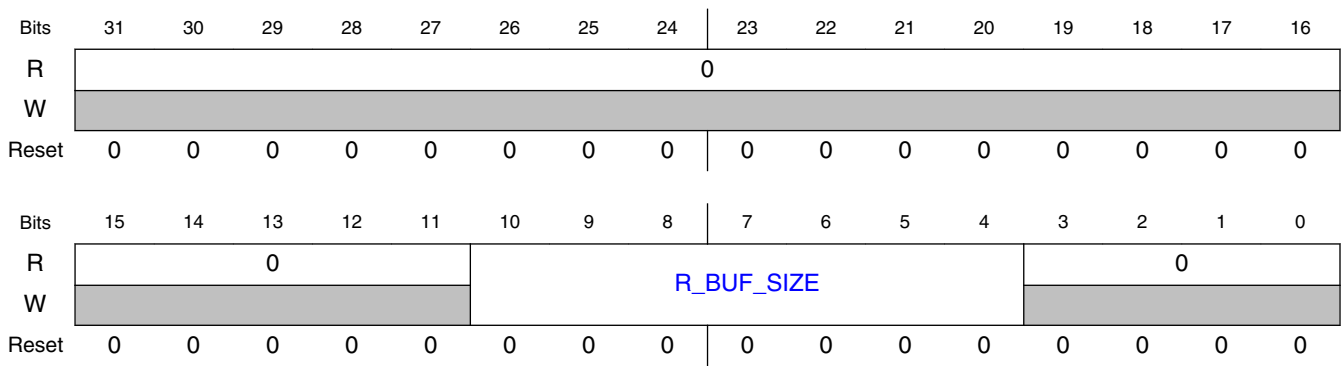
Register	Offset
MRBR2	174h

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	MRBR2
ENET1G	MRBR2	—

11.6.4.1.27.2 Diagram



11.6.4.1.27.3 Fields

Field	Description
31-11 —	Reserved
10-4 R_BUF_SIZE	Receive buffer size (in bytes)
3-0 —	Reserved

11.6.4.1.28 Receive Descriptor Ring 0 Start Register (RDSR)

RDSR points to the beginning of the circular receive buffer descriptor queue in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

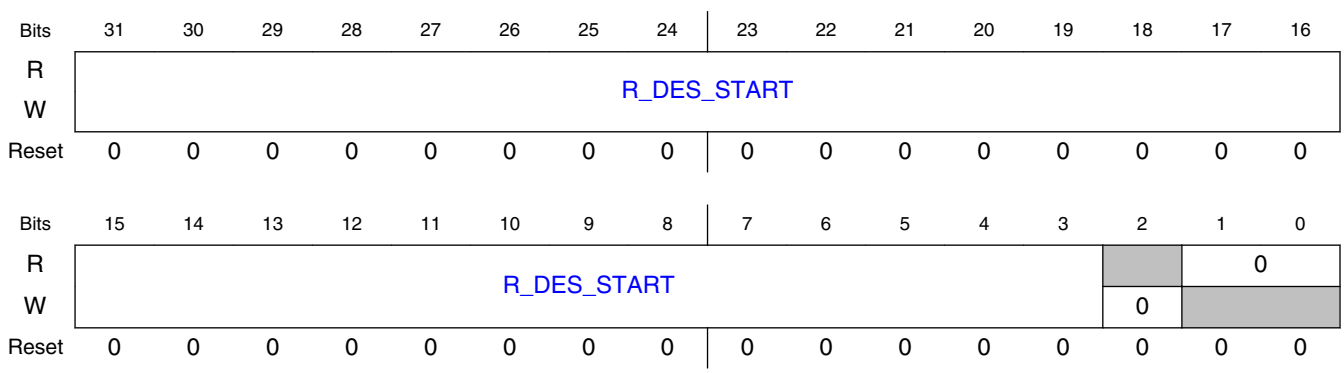
NOTE

This register must be initialized prior to operation

11.6.4.1.28.1 Offset

Register	Offset
RDSR	180h

11.6.4.1.28.2 Diagram



11.6.4.1.28.3 Fields

Field	Description
31-3 R_DES_START	Pointer to the beginning of the receive buffer descriptor queue. 0
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

11.6.4.1.29 Transmit Buffer Descriptor Ring 0 Start Register (TDSR)

TDSR provides a pointer to the beginning of the circular transmit buffer descriptor queue 0 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation.

11.6.4.1.29.1 Offset

Register	Offset
TDSR	184h

11.6.4.1.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	X_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	X_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.29.3 Fields

Field	Description
31-3 X_DES_START	Pointer to the beginning of the transmit buffer descriptor queue.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

11.6.4.1.30 Maximum Receive Buffer Size Register - Ring 0 (MRBR)

The MRBR is a user-programmable register that dictates the maximum size of all ring-0 receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- R_BUF_SIZE is concatentated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To allow one maximum size frame per buffer, MRBR must be set to RCR[MAX_FL] or larger.
- To properly align the buffer, MRBR must be evenly divisible by 64. To ensure this, set the lower two bits of R_BUF_SIZE to zero. The lower four bits of this register are already set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR greater than or equal to 256 bytes.

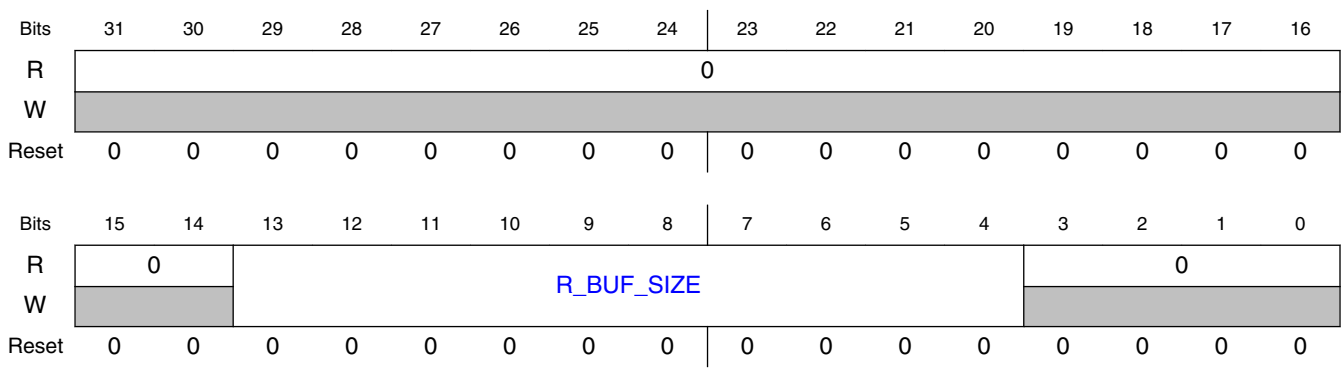
NOTE

This register must be initialized before operation.

11.6.4.1.30.1 Offset

Register	Offset
MRBR	188h

11.6.4.1.30.2 Diagram



11.6.4.1.30.3 Fields

Field	Description
31-14 —	Reserved
13-4 R_BUF_SIZE	Receive buffer size in bytes. This value, concatenated with the four least-significant bits of this register (which are always zero), is the effective maximum receive buffer size. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.
3-0 —	This field, which is always zero, is the four least-significant bits of the maximum receive buffer size.

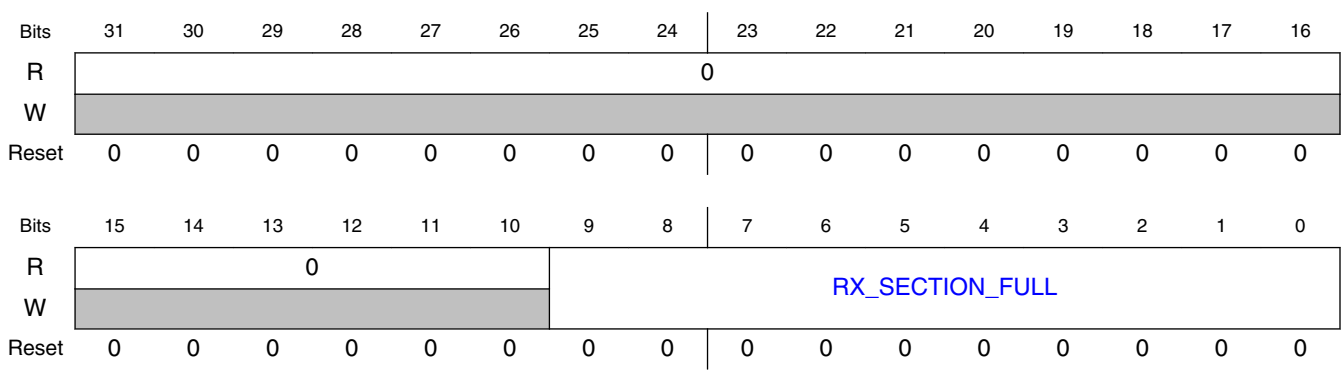
Instance	Field supported in	Field not supported in
ENET	MRBR	—
ENET1G	MRBR[10–4]	MRBR[13–11]

11.6.4.1.31 Receive FIFO Section Full Threshold (RSFL)

11.6.4.1.31.1 Offset

Register	Offset
RSFL	190h

11.6.4.1.31.2 Diagram



11.6.4.1.31.3 Fields

Field	Description
31-10 —	Reserved
9-0 RX_SECTION_FULL	Value Of Receive FIFO Section Full Threshold Value, in 64-bit words, of the receive FIFO section full threshold. Clear this field to enable store and forward on the RX FIFO. When programming a value greater than 0 (cut-through operation), it must be greater than RAEM[RX_ALMOST_EMPTY]. When the FIFO level reaches the value in this field, data is available in the Receive FIFO (cut-through operation). **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

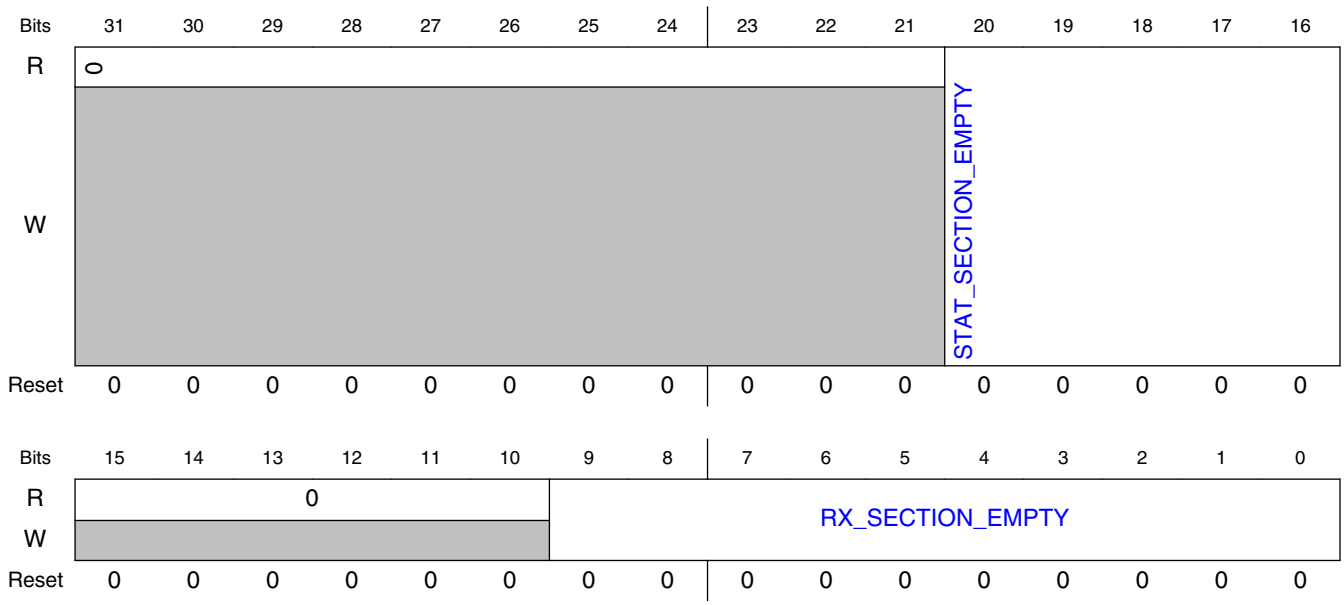
Instance	Field supported in	Field not supported in
ENET	RSFL[7–0]	RSFL[9–8]
ENET1G	RSFL	—

11.6.4.1.32 Receive FIFO Section Empty Threshold (RSEM)

11.6.4.1.32.1 Offset

Register	Offset
RSEM	194h

11.6.4.1.32.2 Diagram



11.6.4.1.32.3 Fields

Field	Description
31-21 —	Reserved
20-16 STAT_SECTION_EMPTY	RX Status FIFO Section Empty Threshold Defines number of frames in the receive FIFO, independent of its size, that can be accepted. If the limit is reached, reception will continue normally, however a pause frame will be triggered to indicate a possible congestion to the remote device to avoid FIFO overflow. A value of 0 disables automatic pause frame generation
15-10 —	Reserved
9-0 RX_SECTION_EMPTY	Value Of The Receive FIFO Section Empty Threshold Value, in 64-bit words, of the receive FIFO section empty threshold. When the FIFO has reached this level, a pause frame will be issued. A value of 0 disables automatic pause frame generation. When the FIFO level goes below the value programmed in this field, an XON pause frame is issued to indicate the FIFO congestion is cleared to the remote Ethernet client. **NOTE:** The section-empty threshold indications from both FIFOs are OR'ed to cause XOFF pause frame generation. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
ENET	RSEM[7–0]	RSEM[9–8]

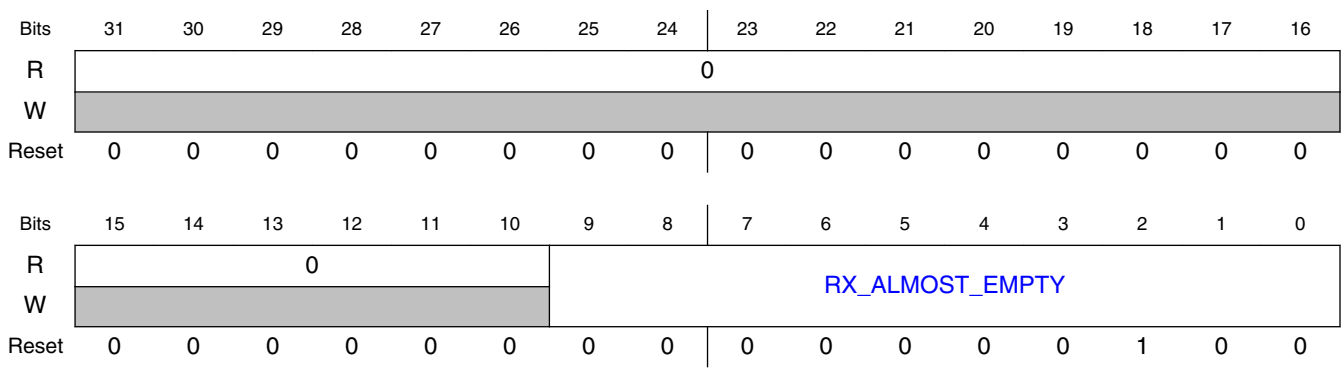
Field	Description		
	Instance	**Field supported in**	**Field not supported in**
	ENET1G	RSEM	—

11.6.4.1.33 Receive FIFO Almost Empty Threshold (RAEM)

11.6.4.1.33.1 Offset

Register	Offset
RAEM	198h

11.6.4.1.33.2 Diagram



11.6.4.1.33.3 Fields

Field	Description
31-10 —	Reserved
9-0 RX_ALMOST_EMPTY	Value Of The Receive FIFO Almost Empty Threshold Value, in 64-bit words, of the receive FIFO almost empty threshold. When the FIFO level reaches the value programmed in this field and the end-of-frame has not been received for the frame yet, the core receive read control stops FIFO read (and subsequently stops transferring data to the MAC client application). It continues to deliver the frame, if again more data than the threshold or the end-of-frame is available in the FIFO. A minimum value of 4 should be set. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

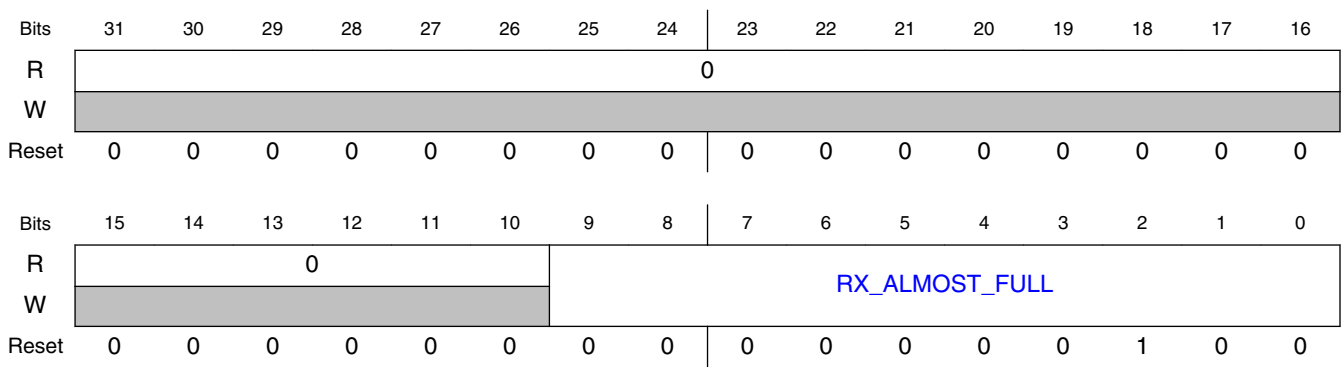
Field	Description
	<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>ENET</td><td>RAEM[7–0]</td><td>RAEM[9–8]</td></tr><tr><td>ENET1G</td><td>RAEM</td><td>—</td></tr></table>

11.6.4.1.34 Receive FIFO Almost Full Threshold (RAFL)

11.6.4.1.34.1 Offset

Register	Offset
RAFL	19Ch

11.6.4.1.34.2 Diagram



11.6.4.1.34.3 Fields

Field	Description
31-10 —	Reserved
9-0 RX_ALMOST_FULL	Value Of The Receive FIFO Almost Full Threshold Value, in 64-bit words, of the receive FIFO almost full threshold. When the FIFO level comes close to the maximum, so that there is no more space for at least RX_ALMOST_FULL number of words, the MAC stops writing data in the FIFO and truncates the received frame to avoid FIFO overflow. The corresponding error status will be set when the frame is delivered to the application. A minimum value of 4 should be set.

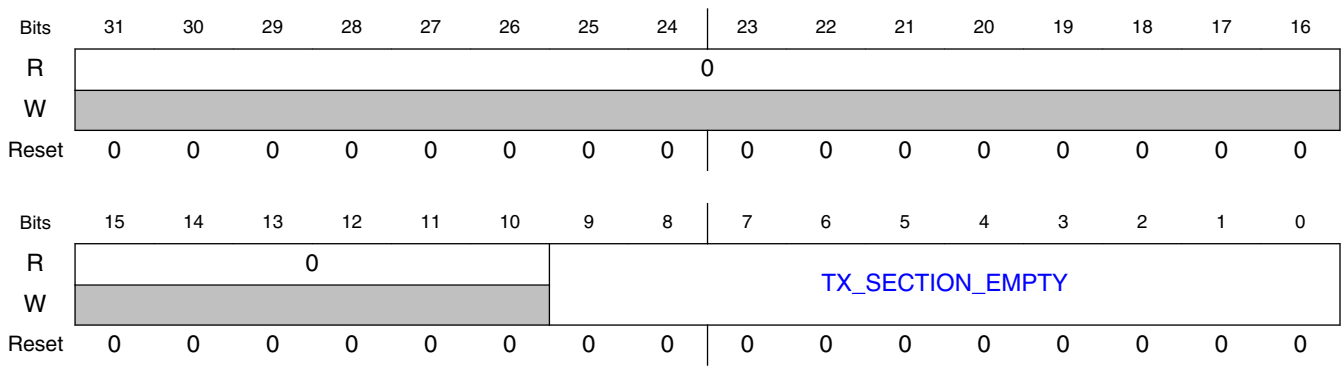
Field	Description
	NOTE: This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET / RAFL[7–0] / RAFL[9–8] ENET1G / RAFL / —

11.6.4.1.35 Transmit FIFO Section Empty Threshold (TSEM)

11.6.4.1.35.1 Offset

Register	Offset
TSEM	1A0h

11.6.4.1.35.2 Diagram



11.6.4.1.35.3 Fields

Field	Description
31-10 —	Reserved
9-0 TX_SECTION_EMPTY	Value Of The Transmit FIFO Section Empty Threshold Value, in 64-bit words, of the transmit FIFO section empty threshold. See Transmit FIFO for more information.

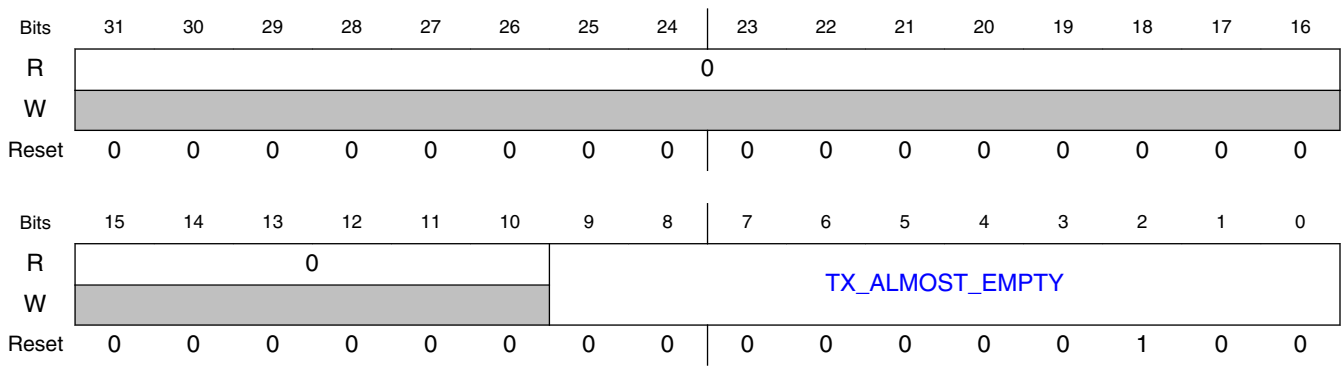
Field	Description
	NOTE: This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET / TSEM[7–0] / TSEM[9–8] ENET1G / TSEM / —

11.6.4.1.36 Transmit FIFO Almost Empty Threshold (TAEM)

11.6.4.1.36.1 Offset

Register	Offset
TAEM	1A4h

11.6.4.1.36.2 Diagram



11.6.4.1.36.3 Fields

Field	Description
31-10 —	Reserved
9-0 TX_ALMOST_EMPTY	Value of Transmit FIFO Almost Empty Threshold Value, in 64-bit words, of the transmit FIFO almost empty threshold.

Field	Description
	When the FIFO level reaches the value programmed in this field, and no end-of-frame is available for the frame, the MAC transmit logic, to avoid FIFO underflow, stops reading the FIFO and transmits a frame with an MII error indication. See Transmit FIFO for more information. A minimum value of 4 should be set. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

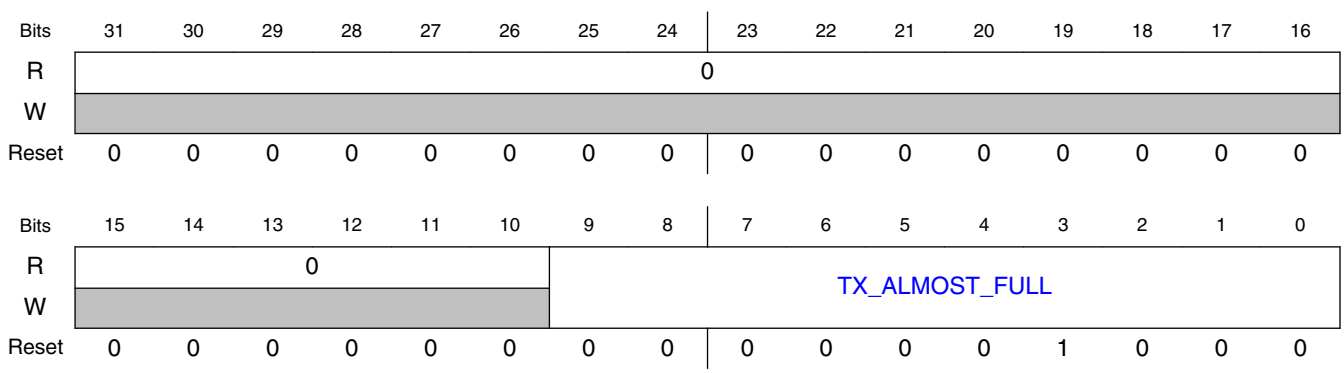
Instance	Field supported in	Field not supported in
ENET	TAEM[7–0]	TAEM[9–8]
ENET1G	TAEM	—

11.6.4.1.37 Transmit FIFO Almost Full Threshold (TAFL)

11.6.4.1.37.1 Offset

Register	Offset
TAFL	1A8h

11.6.4.1.37.2 Diagram



11.6.4.1.37.3 Fields

Field	Description
31-10 —	Reserved

Table continues on the next page...

Field	Description
9-0 TX_ALMOST_FULL	Value Of The Transmit FIFO Almost Full Threshold Value, in 64-bit words, of the transmit FIFO almost full threshold. A minimum value of six is required . A recommended value of at least 8 should be set allowing a latency of two clock cycles to the application. If more latency is required the value can be increased as necessary (latency = TAFL - 5). When the FIFO level comes close to the maximum, so that there is no more space for at least TX_ALMOST_FULL number of words, the pin ff_tx_rdy is deasserted. If the application does not react on this signal, the FIFO write control logic, to avoid FIFO overflow, truncates the current frame and sets the error status. As a result, the frame will be transmitted with an GMII/MII error indication. See Transmit FIFO for more information. **NOTE:** A FIFO overflow is a fatal error and requires a global reset on the transmit datapath or at least deassertion of ETHEREN. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

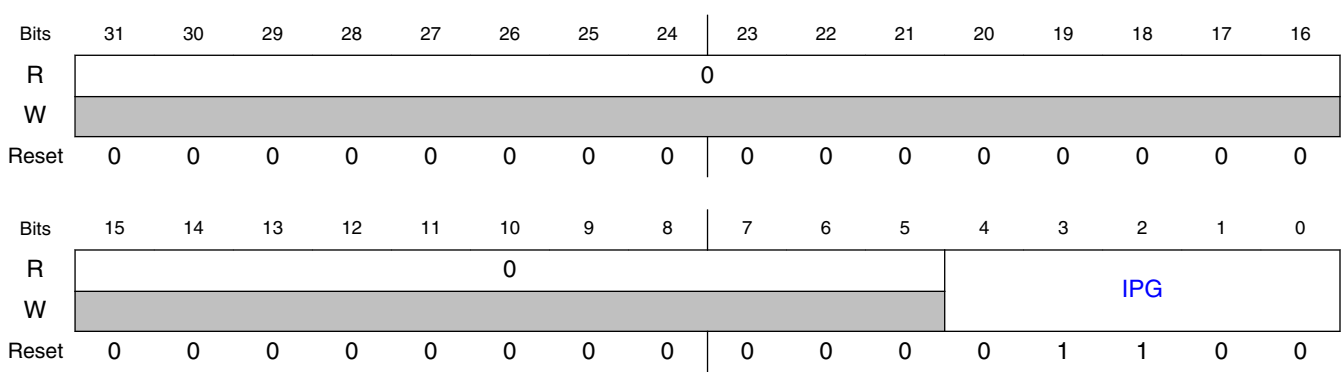
Instance	Field supported in	Field not supported in
ENET	TAFL[7–0]	TAFL[9–8]
ENET1G	TAFL	—

11.6.4.1.38 Transmit Inter-Packet Gap (TIPG)

11.6.4.1.38.1 Offset

Register	Offset
TIPG	1ACh

11.6.4.1.38.2 Diagram



11.6.4.1.38.3 Fields

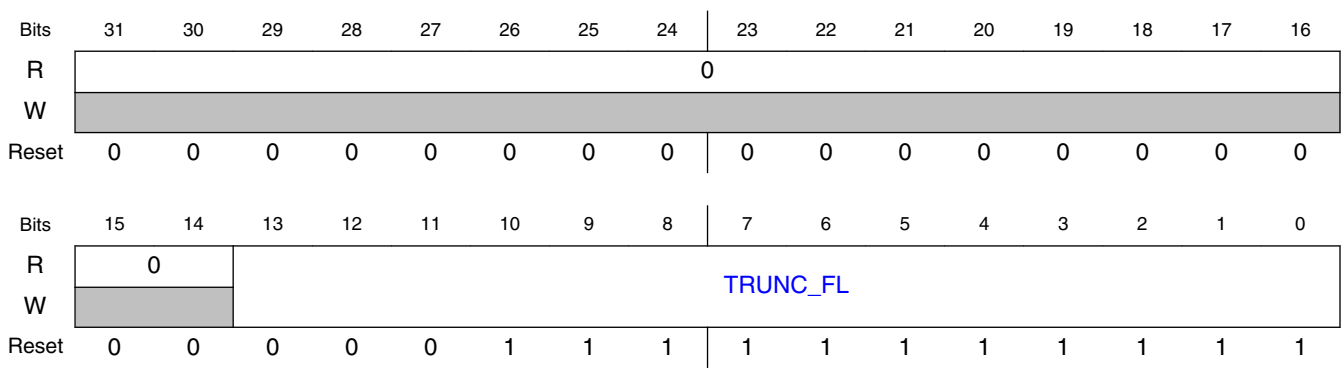
Field	Description
31-5 —	Reserved
4-0 IPG	Transmit Inter-Packet Gap Indicates the IPG, in bytes, between transmitted frames. Valid values range from 8 to 26. If the written value is less than 8 or greater than 26, the internal (effective) IPG is 12. **NOTE:** The IPG value read will be the value that was written, even if it is out of range.

11.6.4.1.39 Frame Truncation Length (FTRL)

11.6.4.1.39.1 Offset

Register	Offset
FTRL	1B0h

11.6.4.1.39.2 Diagram



11.6.4.1.39.3 Fields

Field	Description
31-14 —	Reserved
13-0 TRUNC_FL	Frame Truncation Length

Field	Description
	Indicates the value a receive frame is truncated, if it is greater than this value. Must be greater than or equal to RCR[MAX_FL]. **NOTE:** Truncation happens at TRUNC_FL. However, when truncation occurs, the application (FIFO) may receive less data, guaranteeing that it never receives more than the set limit.

11.6.4.1.40 Transmit Accelerator Function Configuration (TACC)

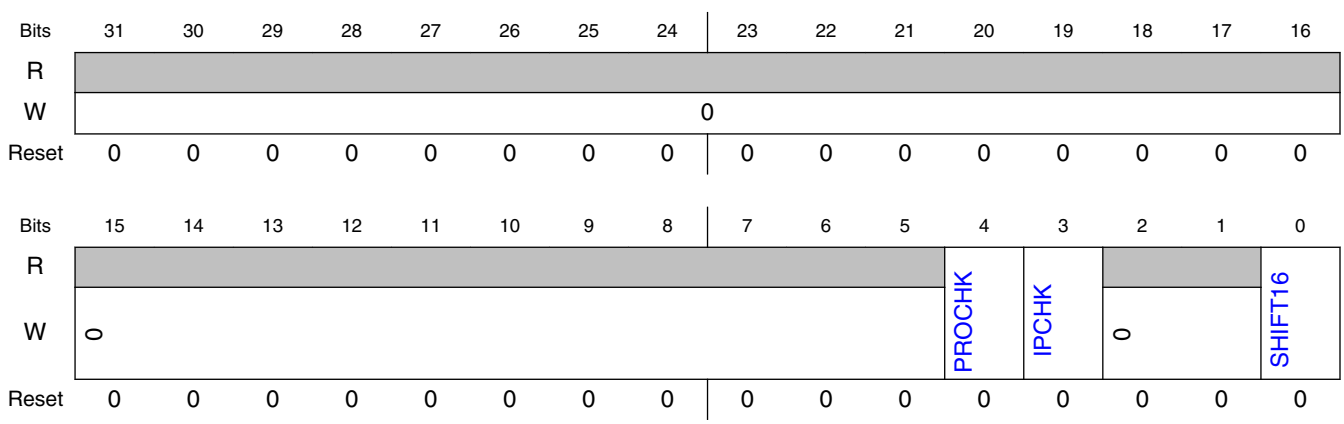
TACC controls accelerator actions when sending frames. The register can be changed before or after each frame, but it must remain unmodified during frame writes into the transmit FIFO.

The TFWR[STRFWD] field must be set to use the checksum feature.

11.6.4.1.40.1 Offset

Register	Offset
TACC	1C0h

11.6.4.1.40.2 Diagram



11.6.4.1.40.3 Fields

Field	Description
31-5 —	This write-only field is reserved. It must always be written with the value 0.

Table continues on the next page...

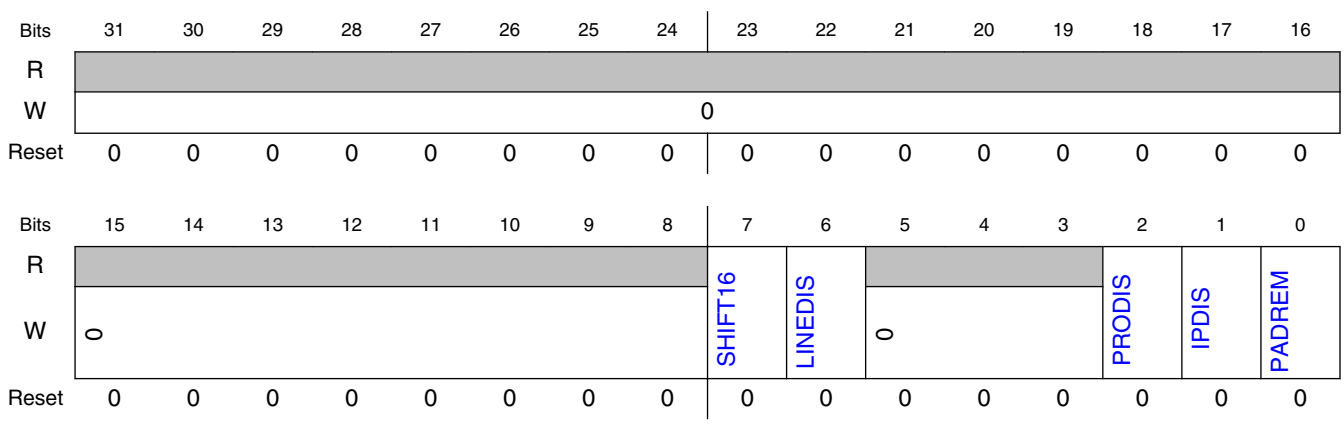
Field	Description
4 PROCHK	Enables insertion of protocol checksum. 0 - Checksum not inserted. 1 - If an IP frame with a known protocol is transmitted, the checksum is inserted automatically into the frame. The checksum field must be cleared. The other frames are not modified.
3 IPCHK	Enables insertion of IP header checksum. 0 - Checksum is not inserted. 1 - If an IP frame is transmitted, the checksum is inserted automatically. The IP header checksum field must be cleared. If a non-IP frame is transmitted the frame is not modified.
2-1 —	This write-only field is reserved. It must always be written with the value 0.
0 SHIFT16	TX FIFO Shift-16 0 - Disabled. 1 - Indicates to the transmit data FIFO that the written frames contain two additional octets before the frame data. This means the actual frame begins at bit 16 of the first word written into the FIFO. This function allows putting the frame payload on a 32-bit boundary in memory, as the 14-byte Ethernet header is extended to a 16-byte header.

11.6.4.1.41 Receive Accelerator Function Configuration (RACC)

11.6.4.1.41.1 Offset

Register	Offset
RACC	1C4h

11.6.4.1.41.2 Diagram



11.6.4.1.41.3 Fields

Field	Description
31-8 —	This write-only field is reserved. It must always be written with the value 0.
7 SHIFT16	RX FIFO Shift-16 When this field is set, the actual frame data starts at bit 16 of the first word read from the RX FIFO aligning the Ethernet payload on a 32-bit boundary. **NOTE:** This function only affects the FIFO storage and has no influence on the statistics, which use the actual length of the frame received. 0 - Disabled. 1 - Instructs the MAC to write two additional bytes in front of each frame received into the RX FIFO.
6 LINEDIS	Enable Discard Of Frames With MAC Layer Errors 0 - Frames with errors are not discarded. 1 - Any frame received with a CRC, length, or PHY error is automatically discarded and not forwarded to the user application interface.
5-3 —	This write-only field is reserved. It must always be written with the value 0.
2 PRODIS	Enable Discard Of Frames With Wrong Protocol Checksum 0 - Frames with wrong checksum are not discarded. 1 - If a TCP/IP, UDP/IP, or ICMP/IP frame is received that has a wrong TCP, UDP, or ICMP checksum, the frame is discarded. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
1 IPDIS	Enable Discard Of Frames With Wrong IPv4 Header Checksum 0 - Frames with wrong IPv4 header checksum are not discarded. 1 - If an IPv4 frame is received with a mismatching header checksum, the frame is discarded. IPv6 has no header checksum and is not affected by this setting. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
0 PADREM	Enable Padding Removal For Short IP Frames 0 - Padding not removed. 1 - Any bytes following the IP payload section of the frame are removed from the frame.

11.6.4.1.42 Receive Classification Match Register for Class n (RCMR1 - RCMR2)

This match register allows specifying up to four priorities, which are tested (OR'ed) simultaneously. The match detection uses the extracted VLAN field according to the rules for VLAN detection configured through the ECR register. If both match registers, RCMR1 and RCMR2, report a match at the same time, only the class 1 match is indicated as the final result.

11.6.4.1.42.1 Offset

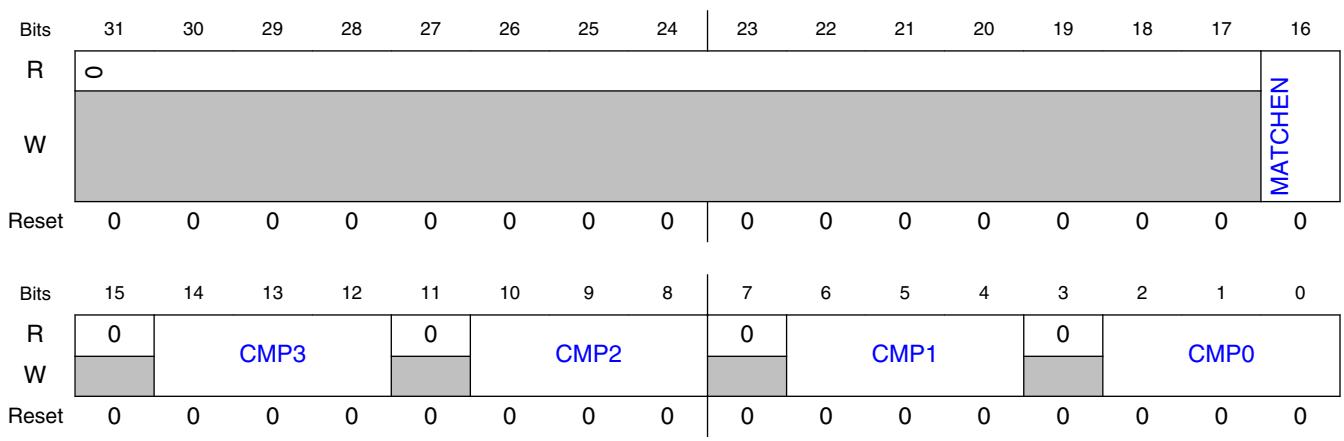
Register	Offset
RCMR1	1C8h
RCMR2	1CCh

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	RCMR1–RCMR2
ENET1G	RCMR1–RCMR2	—

11.6.4.1.42.2 Diagram



11.6.4.1.42.3 Fields

Field	Description
31-17 —	This write-only field is reserved. It must always be written with the value 0.
16 MATCHEN	Match Enable **NOTE:** A comparison is done only on incoming VLAN frames. If no VLAN frame is received no match will occur. If both match registers have overlapping compare values and hence can match both on the same frame, only class 1 will be indicated and the class 2 match is ignored.

Table continues on the next page...

Field	Description
	0 - Disabled (default): no compares will occur and the classification indicator for this class will never assert. 1 - The register contents are valid and a comparison with all compare values is done when a VLAN frame is received.
15 —	This write-only field is reserved. It must always be written with the value 0.
14-12 CMP3	Compare 3 Fourth value to compare against. If unused it must be set to the same value as CMP0.
11 —	This write-only field is reserved. It must always be written with the value 0.
10-8 CMP2	Compare 2 Third value to compare against. If unused it must be set to the same value as CMP0.
7 —	This write-only field is reserved. It must always be written with the value 0.
6-4 CMP1	Compare 1 Second value to compare against. If unused it must be set to the same value as CMP0.
3 —	This write-only field is reserved. It must always be written with the value 0.
2-0 CMP0	Compare 0 A three-bit value that will be compared with the frame's VLAN priority field (if a VLAN frame is received). All four compare values, CMP0..3, will be used in parallel. If any of the values match, a match for the class is reported (if MATCHEN is 1). **NOTE:** To implement a single priority match, all four compare values must be set to the same value.

11.6.4.1.43 DMA Class Based Configuration (DMA1CFG - DMA2CFG)

The DMA class based configuration registers are used to configure the DMA controller interface to support the additional class 1 (buffer descriptor ring 1) and class 2 (buffer descriptor ring 2) traffic and define configuration options such as bandwidth allocation as needed.

NOTE

The registers are cleared when ECR[ETHEREN] becomes 0.

11.6.4.1.43.1 Offset

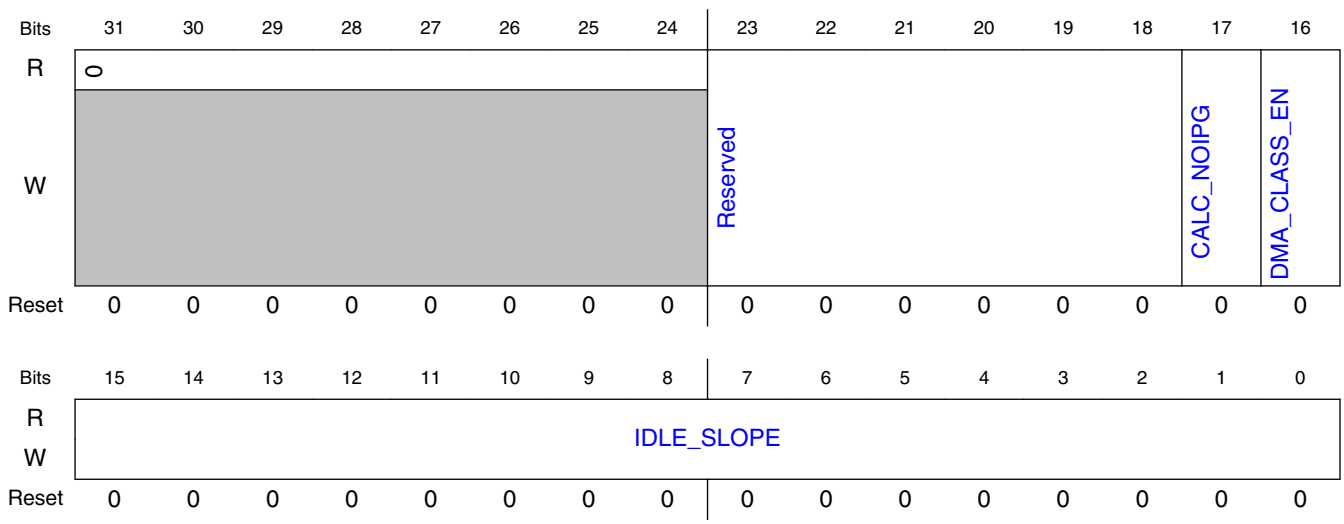
Register	Offset
DMA1CFG	1D8h
DMA2CFG	1DCh

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	DMA1CFG–DMA2CFG
ENET1G	DMA1CFG–DMA2CFG	—

11.6.4.1.43.2 Diagram



11.6.4.1.43.3 Fields

Field	Description
31-24 —	Reserved
23-18 —	Write only zeroes to this field **NOTE:** Write only zeroes to this field.
17 CALC_NOIPG	Calculate no IPG Disable inclusion of IPG bytes for bandwidth calculations. 0 - The traffic shaper function should consider 12 octets of IPG in addition to the frame data transferred for a frame when doing bandwidth calculations. This is the default. 1 - Addition of 12 bytes for the IPG should be omitted when calculating the bandwidth (for traffic shaping, when writing a frame into the transmit FIFO, the shaper will usually consider 12 bytes of IPG for every frame as part of the bandwidth allocated by the frame. This addition can be suppressed, meaning short frames will become more bandwidth than large frames due to the relation of data to IPG overhead).
16	DMA class enable

Table continues on the next page...

Field	Description
DMA_CLASS_EN	0 - The DMA controller's channel for the class is not used. Disabling the DMA controller of a class also requires disabling the class match comparator for the class (see registers RCMRn). When class 1 and class 2 queues are disabled then their frames will be placed in queue 0. 1 - Enable the DMA controller to support the corresponding descriptor ring for this class of traffic.
15-0 IDLE_SLOPE	Idle slope 16-bit value to define the per class idle slope setting used by the credit based shaper defining allocated bandwidth for the class. This value is used to calculate the BW (bandwidth) fraction, given by the equation, BW fraction = 1/(1+512/IDLE_SLOPE). Idle slope is restricted to certain values. For values less than 128, idle slope = 2^n, where n = 0, 1, 2, ...6. For values equal to or greater than 128, idle slope = 128×m, where m = 1, 2, 3, ...12. Example 1. BW fraction = 0.20 = 1/(1+(512/128)); therefore idleslope = 128. Example 2. BW fraction = 0.33 = 1/(1+(512/256)); therefore idleslope = 256. Example 3. BW fraction = 0.75 = 1/(1+(512/1536)); therefore idleslope = 1536. **NOTE:** For AVB applications, the BW fraction of class 1 and class 2 combined must not exceed .75.

11.6.4.1.44 Receive Descriptor Active Register - Ring 1 (RDAR1)

RDAR1 is a command register, written by the user, to indicate that the receive descriptor ring 1 has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

11.6.4.1.44.1 Offset

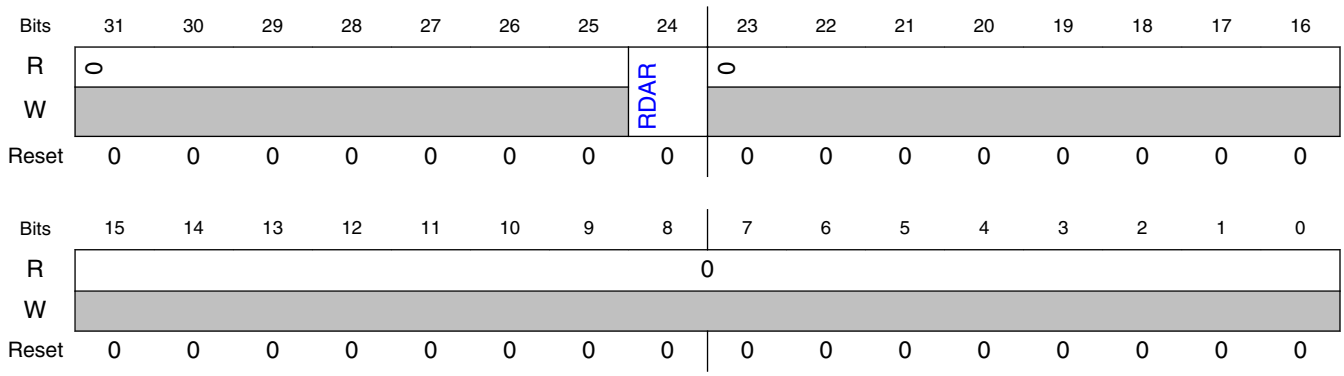
Register	Offset
RDAR1	1E0h

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	RDAR1
ENET1G	RDAR1	—

11.6.4.1.44.2 Diagram



11.6.4.1.44.3 Fields

Field	Description
31-25 —	Reserved
24 RDAR	Receive Descriptor Active Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

11.6.4.1.45 Transmit Descriptor Active Register - Ring 1 (TDAR1)

TDAR1 is a command register that the user writes to indicate that transmit descriptor ring 1 has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

The TDAR register is cleared at reset, when ECR[ETHEREN] transitions from set to cleared, or when ECR[RESET] is set.

11.6.4.1.45.1 Offset

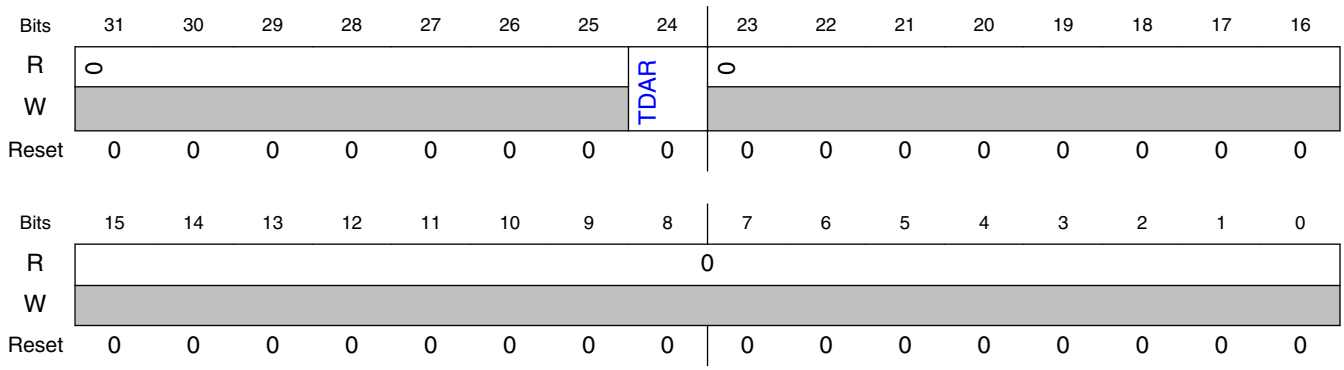
Register	Offset
TDAR1	1E4h

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	TDAR1
ENET1G	TDAR1	—

11.6.4.1.45.2 Diagram



11.6.4.1.45.3 Fields

Field	Description
31-25 —	Reserved
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

11.6.4.1.46 Receive Descriptor Active Register - Ring 2 (RDAR2)

RDAR2 is a command register, written by the user, to indicate that the receive descriptor ring 2 has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

11.6.4.1.46.1 Offset

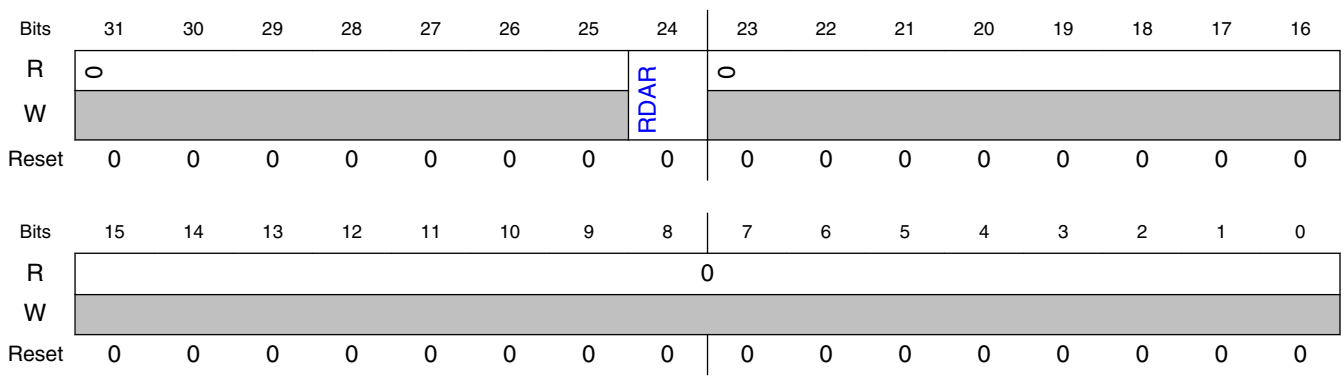
Register	Offset
RDAR2	1E8h

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	RDAR2
ENET1G	RDAR2	—

11.6.4.1.46.2 Diagram



11.6.4.1.46.3 Fields

Field	Description
31-25 —	Reserved
24 RDAR	Receive Descriptor Active Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

11.6.4.1.47 Transmit Descriptor Active Register - Ring 2 (TDAR2)

TDAR2 is a command register that the user writes to indicate that transmit descriptor ring 2 has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

11.6.4.1.47.1 Offset

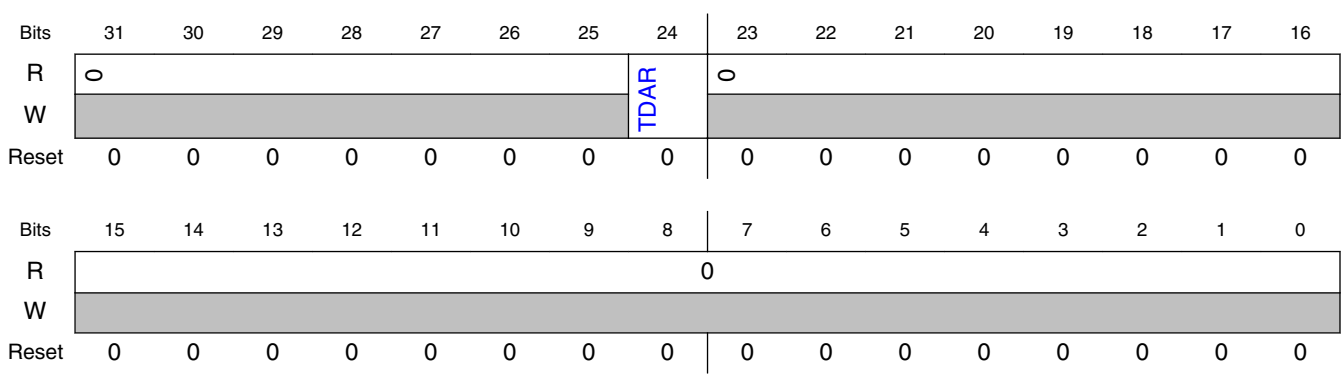
Register	Offset
TDAR2	1ECh

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	TDAR2
ENET1G	TDAR2	—

11.6.4.1.47.2 Diagram



11.6.4.1.47.3 Fields

Field	Description
31-25 —	Reserved
24	Transmit Descriptor Active

Table continues on the next page...

Field	Description
TDAR	Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

11.6.4.1.48 QOS Scheme (QOS)

This register sets the QOS scheme.

NOTE

When both class 1 and class 2 are disabled, RX flushing for these rings must also be disabled.

11.6.4.1.48.1 Offset

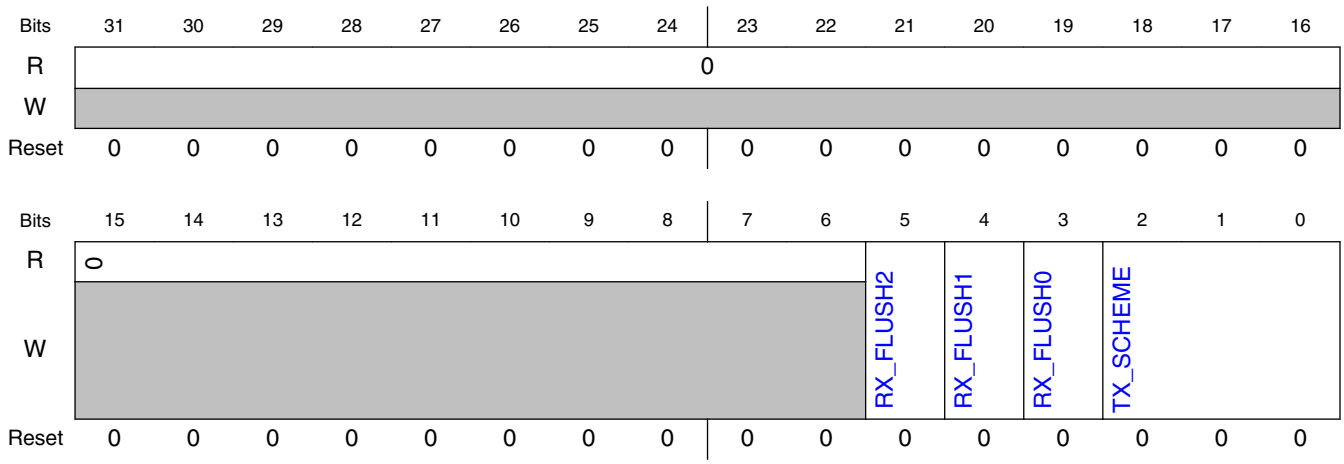
Register	Offset
QOS	1F0h

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	QOS
ENET1G	QOS	—

11.6.4.1.48.2 Diagram



11.6.4.1.48.3 Fields

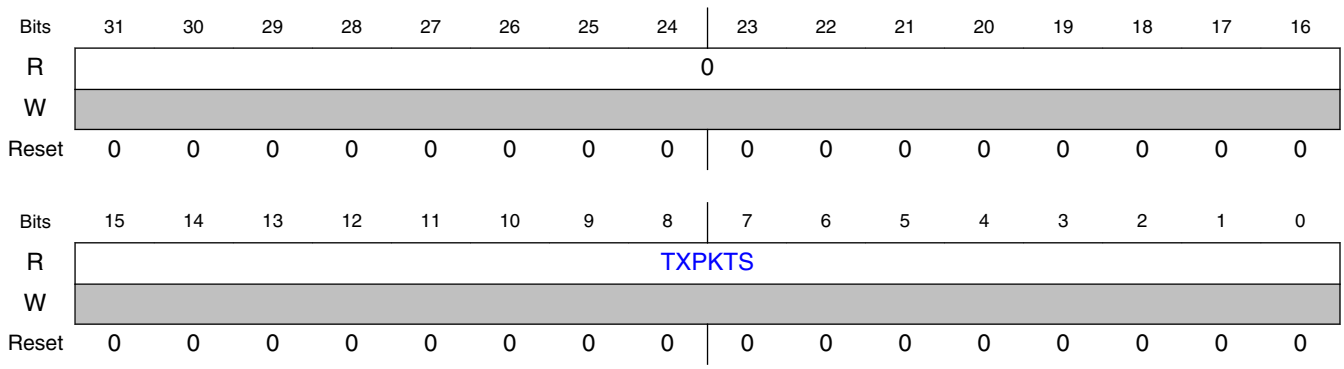
Field	Description
31-6 —	Reserved
5 RX_FLUSH2	RX Flush Ring 2 Enable or disable RX Flush for ring 2. See Receive flushing. 0 - Disable 1 - Enable
4 RX_FLUSH1	RX Flush Ring 1 Enable or disable RX Flush for ring 1. See Receive flushing. 0 - Disable 1 - Enable
3 RX_FLUSH0	RX Flush Ring 0 Enable or disable RX Flush for ring 0. See Receive flushing. 0 - Disable 1 - Enable
2-0 TX_SCHEME	TX scheme configuration Configuration information for DMA to select transmitter queue selection/arbitration scheme. 000 - Credit-based scheme 001 - Round-robin scheme 010-111 - Reserved

11.6.4.1.49 Tx Packet Count Statistic Register (RMON_T_PACKETS)

11.6.4.1.49.1 Offset

Register	Offset
RMON_T_PACKETS	204h

11.6.4.1.49.2 Diagram



11.6.4.1.49.3 Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Packet count Transmit packet count

11.6.4.1.50 Tx Broadcast Packets Statistic Register (RMON_T_BC_PKT)

RMON Tx Broadcast Packets

11.6.4.1.50.1 Offset

Register	Offset
RMON_T_BC_PKT	208h

11.6.4.1.50.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.50.3 Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of broadcast packets

11.6.4.1.51 Tx Multicast Packets Statistic Register (RMON_T_MC_PKT)

11.6.4.1.51.1 Offset

Register	Offset
RMON_T_MC_PKT	20Ch

11.6.4.1.51.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.51.3 Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of multicast packets

11.6.4.1.52 Tx Packets with CRC/Align Error Statistic Register (RMON_T_CRC_ALIGN)

11.6.4.1.52.1 Offset

Register	Offset
RMON_T_CRC_ALIGN	210h

11.6.4.1.52.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.52.3 Fields

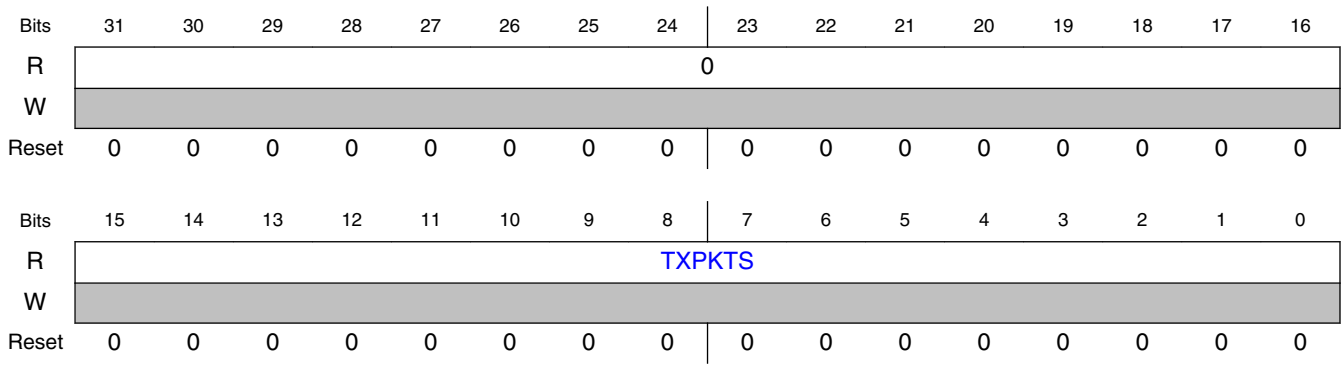
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of packets with CRC/align error

11.6.4.1.53 Tx Packets Less Than Bytes and Good CRC Statistic Register (RMON_T_UNDERSIZE)

11.6.4.1.53.1 Offset

Register	Offset
RMON_T_UNDERSIZE	214h

11.6.4.1.53.2 Diagram



11.6.4.1.53.3 Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets less than 64 bytes with good CRC

11.6.4.1.54 Tx Packets GT MAX_FL bytes and Good CRC Statistic Register (RMON_T_OVERSIZE)

11.6.4.1.54.1 Offset

Register	Offset
RMON_T_OVERSIZE	218h

11.6.4.1.54.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.54.3 Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets greater than MAX_FL bytes with good CRC

11.6.4.1.55 Tx Packets Less Than 64 Bytes and Bad CRC Statistic Register (RMON_T_FRAG)

.

11.6.4.1.55.1 Offset

Register	Offset
RMON_T_FRAG	21Ch

11.6.4.1.55.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.55.3 Fields

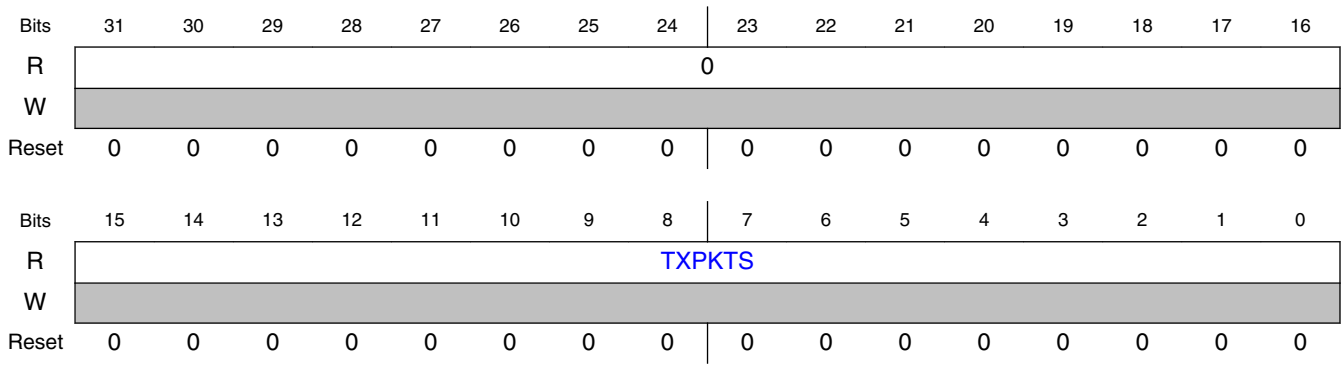
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of packets less than 64 bytes with bad CRC

11.6.4.1.56 Tx Packets Greater Than MAX_FL bytes and Bad CRC Statistic Register (RMON_T_JAB)

11.6.4.1.56.1 Offset

Register	Offset
RMON_T_JAB	220h

11.6.4.1.56.2 Diagram



11.6.4.1.56.3 Fields

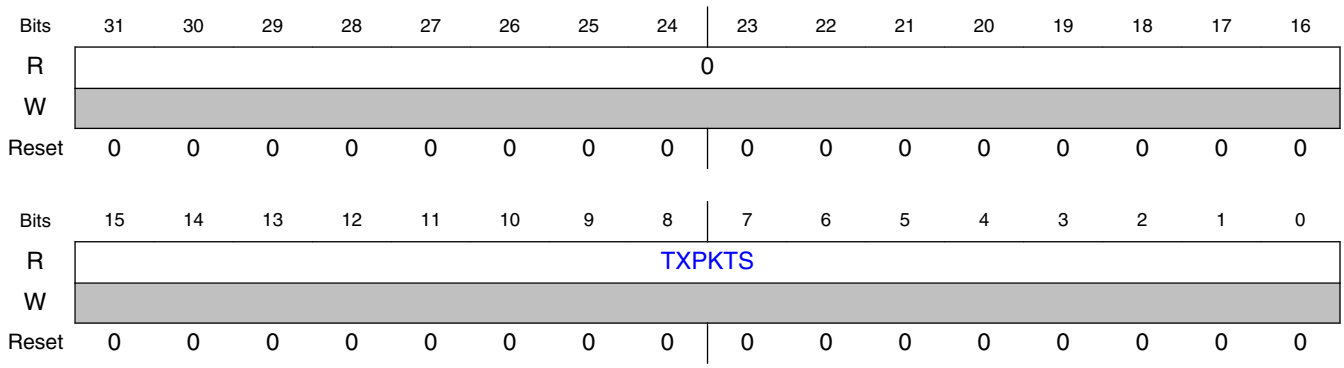
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets greater than MAX_FL bytes and bad CRC

11.6.4.1.57 Tx Collision Count Statistic Register (RMON_T_COL)

11.6.4.1.57.1 Offset

Register	Offset
RMON_T_COL	224h

11.6.4.1.57.2 Diagram



11.6.4.1.57.3 Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of transmit collisions

11.6.4.1.58 Tx 64-Byte Packets Statistic Register (RMON_T_P64)

.

11.6.4.1.58.1 Offset

Register	Offset
RMON_T_P64	228h

11.6.4.1.58.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.58.3 Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of 64-byte transmit packets

11.6.4.1.59 Tx 65- to 127-byte Packets Statistic Register (RMON_T_P65TO127)

11.6.4.1.59.1 Offset

Register	Offset
RMON_T_P65TO127	22Ch

11.6.4.1.59.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.59.3 Fields

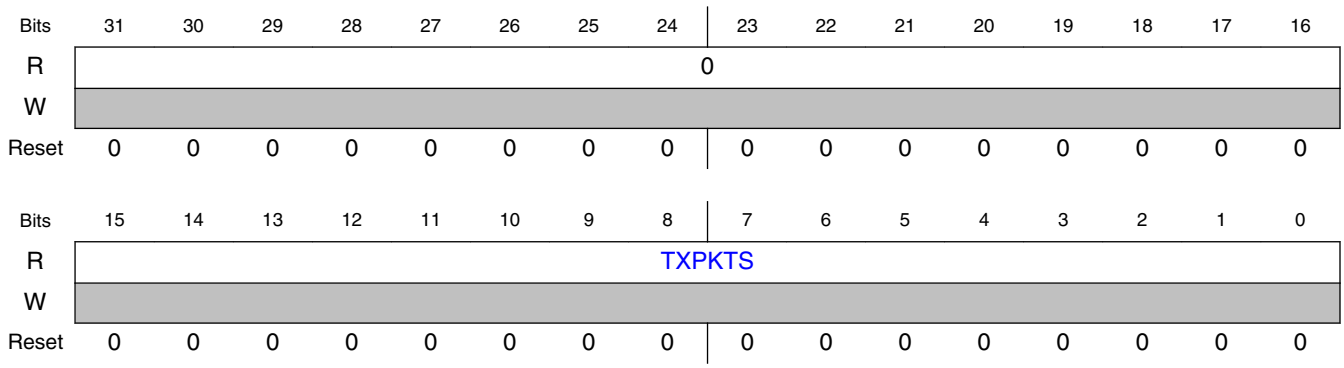
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of 65- to 127-byte transmit packets

11.6.4.1.60 Tx 128- to 255-byte Packets Statistic Register (RMON_T_P128TO255)

11.6.4.1.60.1 Offset

Register	Offset
RMON_T_P128TO255	230h

11.6.4.1.60.2 Diagram



11.6.4.1.60.3 Fields

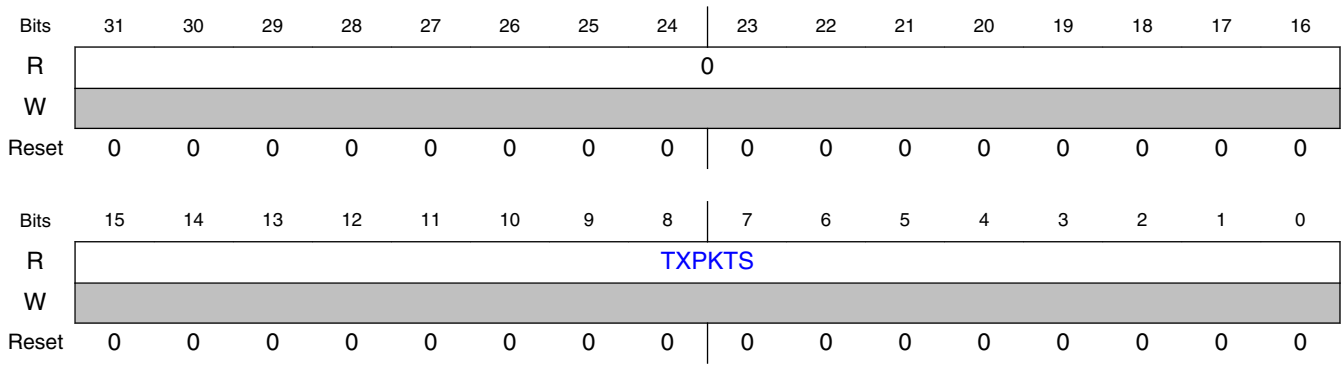
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of 128- to 255-byte transmit packets

11.6.4.1.61 Tx 256- to 511-byte Packets Statistic Register (RMON_T_P256TO511)

11.6.4.1.61.1 Offset

Register	Offset
RMON_T_P256TO511	234h

11.6.4.1.61.2 Diagram



11.6.4.1.61.3 Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of 256- to 511-byte transmit packets

11.6.4.1.62 Tx 512- to 1023-byte Packets Statistic Register (RMON_T_P512TO1023)

.

11.6.4.1.62.1 Offset

Register	Offset
RMON_T_P512TO1023	238h

11.6.4.1.62.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.62.3 Fields

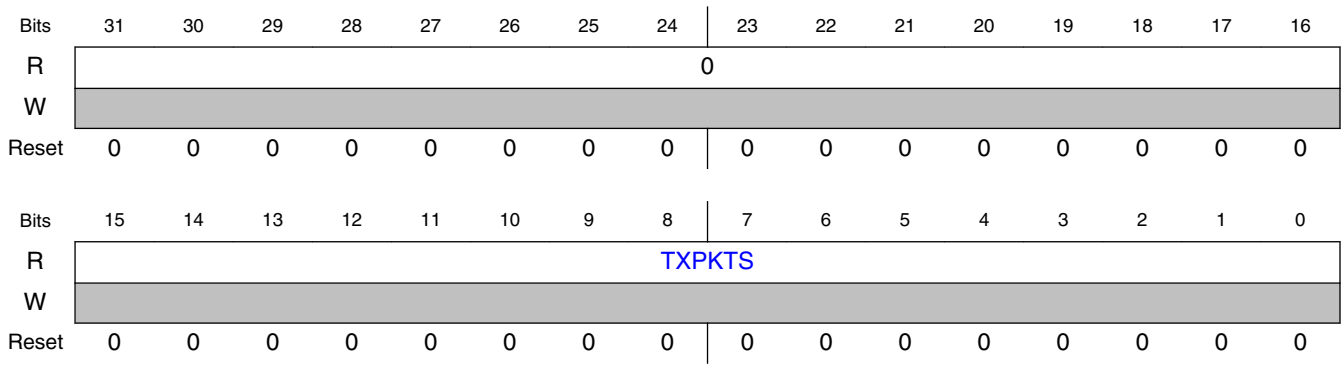
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of 512- to 1023-byte transmit packets

11.6.4.1.63 Tx 1024- to 2047-byte Packets Statistic Register (RMON_T_P1024TO2047)

11.6.4.1.63.1 Offset

Register	Offset
RMON_T_P1024TO2047	23Ch

11.6.4.1.63.2 Diagram



11.6.4.1.63.3 Fields

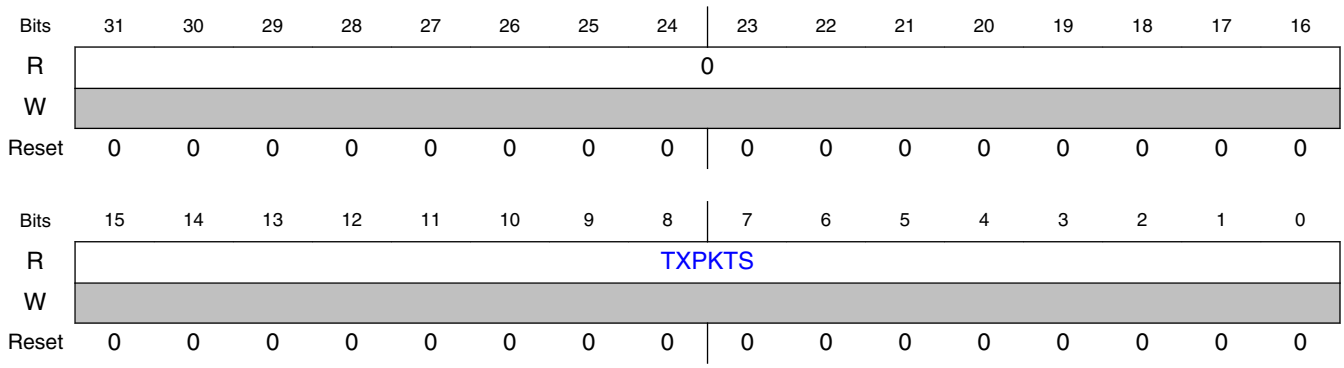
Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of 1024- to 2047-byte transmit packets

11.6.4.1.64 Tx Packets Greater Than 2048 Bytes Statistic Register (RMON_T_P_GTE2048)

11.6.4.1.64.1 Offset

Register	Offset
RMON_T_P_GTE2048	240h

11.6.4.1.64.2 Diagram



11.6.4.1.64.3 Fields

Field	Description
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets greater than 2048 bytes

11.6.4.1.65 Tx Octets Statistic Register (RMON_T_OCTETS)

11.6.4.1.65.1 Offset

Register	Offset
RMON_T_OCTETS	244h

11.6.4.1.65.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXOCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXOCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.65.3 Fields

Field	Description
31-0 TXOCTS	Number of transmit octets

11.6.4.1.66 Frames Transmitted OK Statistic Register (IEEE_T_FRAME_OK)

11.6.4.1.66.1 Offset

Register	Offset
IEEE_T_FRAME_OK	24Ch

11.6.4.1.66.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.66.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted OK **NOTE:** Does not increment for the broadcast frames when broadcast reject is enabled and promiscuous mode is disabled within the receive control register (RCR).

11.6.4.1.67 Frames Transmitted with Single Collision Statistic Register (IEEE_T_1COL)

11.6.4.1.67.1 Offset

Register	Offset
IEEE_T_1COL	250h

11.6.4.1.67.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.67.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with one collision

11.6.4.1.68 Frames Transmitted with Multiple Collisions Statistic Register (IEEE_T_MCOL)

11.6.4.1.68.1 Offset

Register	Offset
IEEE_T_MCOL	254h

11.6.4.1.68.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.68.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with multiple collisions

11.6.4.1.69 Frames Transmitted after Deferral Delay Statistic Register (IEEE_T_DEF)

11.6.4.1.69.1 Offset

Register	Offset
IEEE_T_DEF	258h

11.6.4.1.69.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.69.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with deferral delay

11.6.4.1.70 Frames Transmitted with Late Collision Statistic Register (IEEE_T_LCOL)

11.6.4.1.70.1 Offset

Register	Offset
IEEE_T_LCOL	25Ch

11.6.4.1.70.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.70.3 Fields

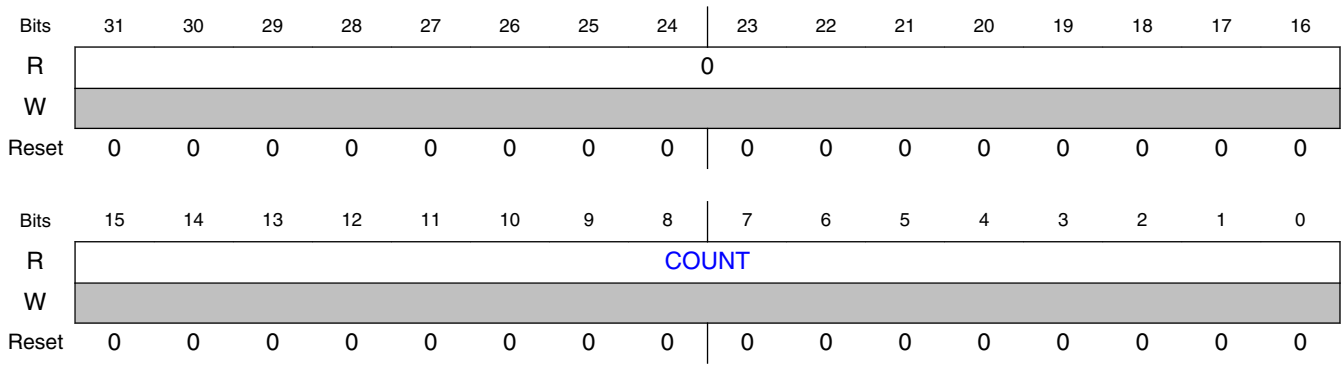
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with late collision

11.6.4.1.71 Frames Transmitted with Excessive Collisions Statistic Register (IEEE_T_EXCOL)

11.6.4.1.71.1 Offset

Register	Offset
IEEE_T_EXCOL	260h

11.6.4.1.71.2 Diagram



11.6.4.1.71.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with excessive collisions

11.6.4.1.72 Frames Transmitted with Tx FIFO Underrun Statistic Register (IEEE_T_MACERR)

11.6.4.1.72.1 Offset

Register	Offset
IEEE_T_MACERR	264h

11.6.4.1.72.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.72.3 Fields

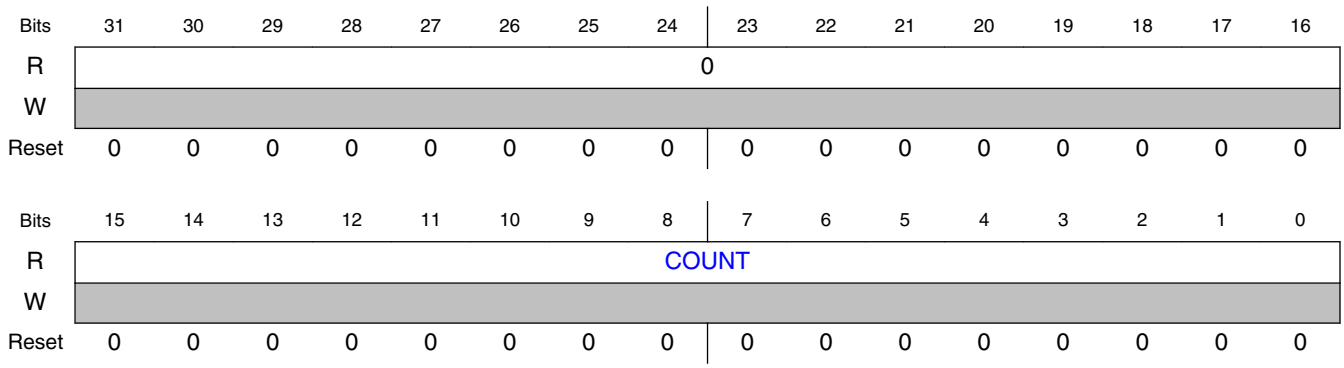
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with transmit FIFO underrun

11.6.4.1.73 Frames Transmitted with Carrier Sense Error Statistic Register (IEEE_T_CSERR)

11.6.4.1.73.1 Offset

Register	Offset
IEEE_T_CSERR	268h

11.6.4.1.73.2 Diagram



11.6.4.1.73.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with carrier sense error

11.6.4.1.74 Reserved Statistic Register (IEEE_T_SQE)

11.6.4.1.74.1 Offset

Register	Offset
IEEE_T_SQE	26Ch

11.6.4.1.74.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.74.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	This read-only field is reserved and always has the value 0 This read-only field is reserved and always has the value 0. **NOTE:** Counter not implemented as no SQE information is available.

11.6.4.1.75 Flow Control Pause Frames Transmitted Statistic Register (IEEE_T_FDXFC)

11.6.4.1.75.1 Offset

Register	Offset
IEEE_T_FDXFC	270h

11.6.4.1.75.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.75.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of flow-control pause frames transmitted

11.6.4.1.76 Octet Count for Frames Transmitted w/o Error Statistic Register (IEEE_T_OCTETS_OK)

11.6.4.1.76.1 Offset

Register	Offset
IEEE_T_OCTETS_OK	274h

11.6.4.1.76.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.76.3 Fields

Field	Description
31-0 COUNT	Octet count for frames transmitted without error Counts total octets (includes header and FCS fields).

11.6.4.1.77 Rx Packet Count Statistic Register (RMON_R_PACKETS)

11.6.4.1.77.1 Offset

Register	Offset
RMON_R_PACKETS	284h

11.6.4.1.77.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.77.3 Fields

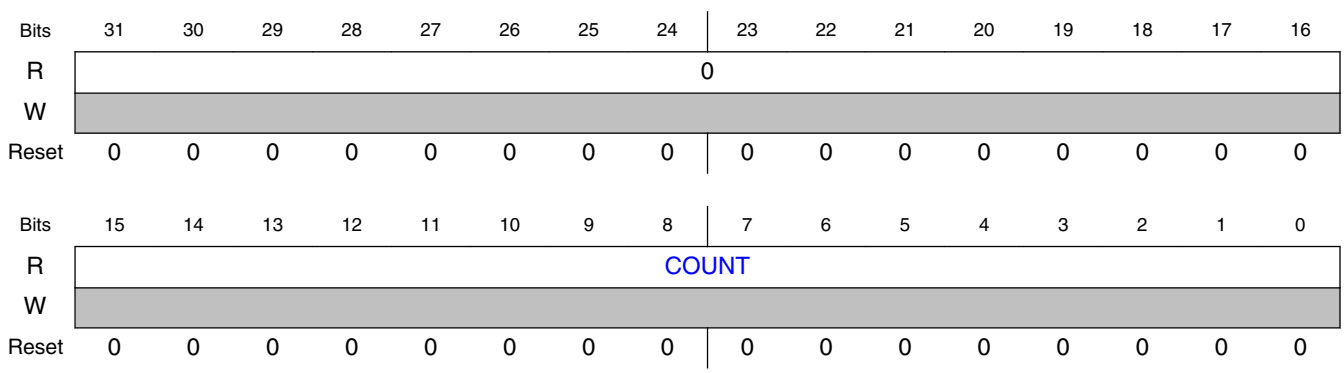
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of packets received

11.6.4.1.78 Rx Broadcast Packets Statistic Register (RMON_R_BC_PKT)

11.6.4.1.78.1 Offset

Register	Offset
RMON_R_BC_PKT	288h

11.6.4.1.78.2 Diagram



11.6.4.1.78.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive broadcast packets

11.6.4.1.79 Rx Multicast Packets Statistic Register (RMON_R_MC_PKT)

11.6.4.1.79.1 Offset

Register	Offset
RMON_R_MC_PKT	28Ch

11.6.4.1.79.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.79.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive multicast packets

11.6.4.1.80 Rx Packets with CRC/Align Error Statistic Register (RMON_R_CRC_ALIGN)

11.6.4.1.80.1 Offset

Register	Offset
RMON_R_CRC_ALIGN	290h

11.6.4.1.80.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.80.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive packets with CRC or align error

11.6.4.1.81 Rx Packets with Less Than 64 Bytes and Good CRC Statistic Register (RMON_R_UNDERSIZE)

11.6.4.1.81.1 Offset

Register	Offset
RMON_R_UNDERSIZE	294h

11.6.4.1.81.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.81.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive packets with less than 64 bytes and good CRC

11.6.4.1.82 Rx Packets Greater Than MAX_FL and Good CRC Statistic Register (RMON_R_OVERSIZE)

11.6.4.1.82.1 Offset

Register	Offset
RMON_R_OVERSIZE	298h

11.6.4.1.82.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.82.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive packets greater than MAX_FL and good CRC

11.6.4.1.83 Rx Packets Less Than 64 Bytes and Bad CRC Statistic Register (RMON_R_FRAG)

11.6.4.1.83.1 Offset

Register	Offset
RMON_R_FRAG	29Ch

11.6.4.1.83.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.83.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive packets with less than 64 bytes and bad CRC

11.6.4.1.84 Rx Packets Greater Than MAX_FL Bytes and Bad CRC Statistic Register (RMON_R_JAB)

11.6.4.1.84.1 Offset

Register	Offset
RMON_R_JAB	2A0h

11.6.4.1.84.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.84.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of receive packets greater than MAX_FL and bad CRC

11.6.4.1.85 Rx 64-Byte Packets Statistic Register (RMON_R_P64)

11.6.4.1.85.1 Offset

Register	Offset
RMON_R_P64	2A8h

11.6.4.1.85.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.85.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of 64-byte receive packets

11.6.4.1.86 Rx 65- to 127-Byte Packets Statistic Register (RMON_R_P65TO127)

11.6.4.1.86.1 Offset

Register	Offset
RMON_R_P65TO127	2ACh

11.6.4.1.86.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.86.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of 65- to 127-byte recieve packets

11.6.4.1.87 Rx 128- to 255-Byte Packets Statistic Register (RMON_R_P128TO255)

11.6.4.1.87.1 Offset

Register	Offset
RMON_R_P128TO255	2B0h

11.6.4.1.87.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.87.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of 128- to 255-byte recieve packets

11.6.4.1.88 Rx 256- to 511-Byte Packets Statistic Register (RMON_R_P256TO511)

11.6.4.1.88.1 Offset

Register	Offset
RMON_R_P256TO511	2B4h

11.6.4.1.88.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.88.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of 256- to 511-byte recieve packets

11.6.4.1.89 Rx 512- to 1023-Byte Packets Statistic Register (RMON_R_P512TO1023)

11.6.4.1.89.1 Offset

Register	Offset
RMON_R_P512TO1023	2B8h

11.6.4.1.89.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.89.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of 512- to 1023-byte recieve packets

11.6.4.1.90 Rx 1024- to 2047-Byte Packets Statistic Register (RMON_R_P1024TO2047)

11.6.4.1.90.1 Offset

Register	Offset
RMON_R_P1024TO2047	2BCh

11.6.4.1.90.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.90.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of 1024- to 2047-byte recieve packets

11.6.4.1.91 Rx Packets Greater than 2048 Bytes Statistic Register (RMON_R_P_GTE2048)

11.6.4.1.91.1 Offset

Register	Offset
RMON_R_P_GTE2048	2C0h

11.6.4.1.91.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.91.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of greater-than-2048-byte recieve packets

11.6.4.1.92 Rx Octets Statistic Register (RMON_R_OCTETS)

11.6.4.1.92.1 Offset

Register	Offset
RMON_R_OCTETS	2C4h

11.6.4.1.92.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.92.3 Fields

Field	Description
31-0 COUNT	Number of receive octets

11.6.4.1.93 Frames not Counted Correctly Statistic Register (IEEE_R_DROP)

Counter increments if a frame with invalid or missing SFD character is detected and has been dropped. None of the other counters increments if this counter increments.

11.6.4.1.93.1 Offset

Register	Offset
IEEE_R_DROP	2C8h

11.6.4.1.93.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.93.3 Fields

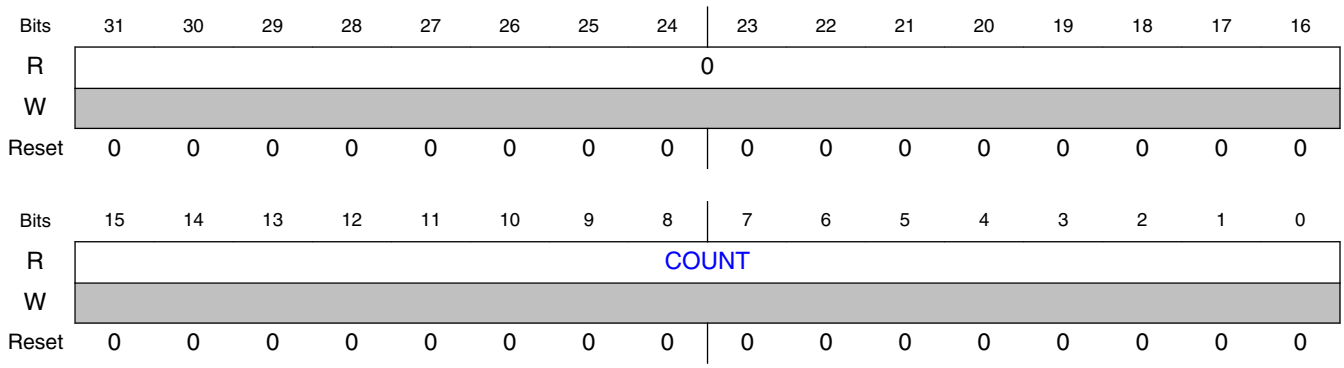
Field	Description
31-16 —	Reserved
15-0 COUNT	Frame count

11.6.4.1.94 Frames Received OK Statistic Register (IEEE_R_FRAME_OK)

11.6.4.1.94.1 Offset

Register	Offset
IEEE_R_FRAME_OK	2CCh

11.6.4.1.94.2 Diagram



11.6.4.1.94.3 Fields

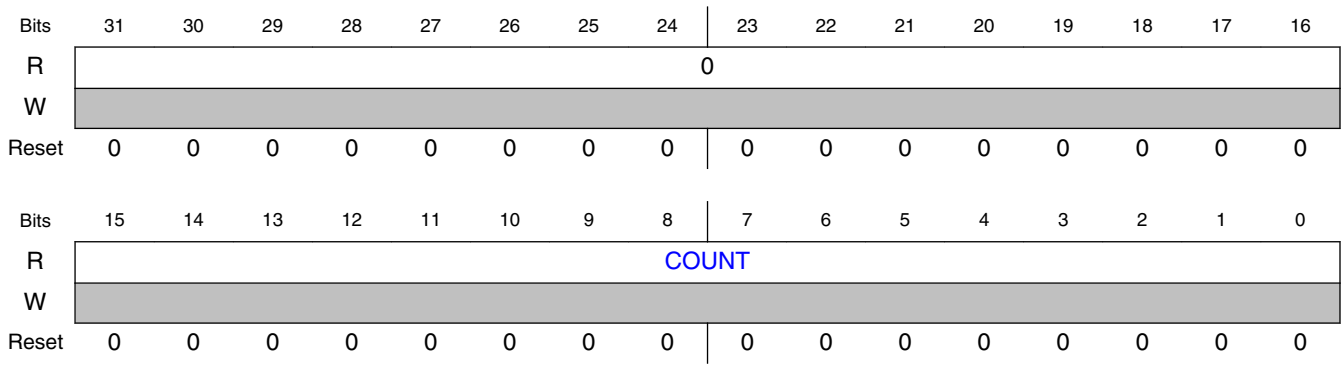
Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames received OK

11.6.4.1.95 Frames Received with CRC Error Statistic Register (IEEE_R_CRC)

11.6.4.1.95.1 Offset

Register	Offset
IEEE_R_CRC	2D0h

11.6.4.1.95.2 Diagram



11.6.4.1.95.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames received with CRC error

11.6.4.1.96 Frames Received with Alignment Error Statistic Register (IEEE_R_ALIGN)

11.6.4.1.96.1 Offset

Register	Offset
IEEE_R_ALIGN	2D4h

11.6.4.1.96.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.96.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of frames received with alignment error

11.6.4.1.97 Receive FIFO Overflow Count Statistic Register (IEEE_R_MACERR)

11.6.4.1.97.1 Offset

Register	Offset
IEEE_R_MACERR	2D8h

11.6.4.1.97.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.97.3 Fields

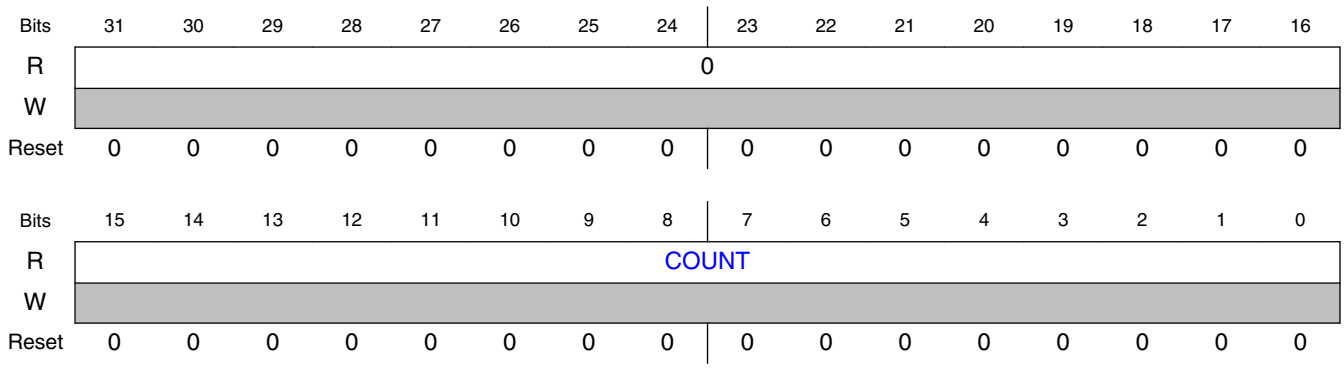
Field	Description
31-16 —	Reserved
15-0 COUNT	Receive FIFO overflow count

11.6.4.1.98 Flow Control Pause Frames Received Statistic Register (IEEE_R_FDXFC)

11.6.4.1.98.1 Offset

Register	Offset
IEEE_R_FDXFC	2DCh

11.6.4.1.98.2 Diagram



11.6.4.1.98.3 Fields

Field	Description
31-16 —	Reserved
15-0 COUNT	Number of flow-control pause frames received

11.6.4.1.99 Octet Count for Frames Received without Error Statistic Register (IEEE_R_OCTETS_OK)

11.6.4.1.99.1 Offset

Register	Offset
IEEE_R_OCTETS_OK	2E0h

11.6.4.1.99.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.99.3 Fields

Field	Description
31-0 COUNT	Number of octets for frames received without error **NOTE:** Counts total octets (includes header and FCS fields). Does not increment for the broadcast frames when broadcast reject is enabled and promiscuous mode is disabled within the receive control register (RCR).

11.6.4.1.100 Adjustable Timer Control Register (ATCR)

ATCR command fields can trigger the corresponding events directly. It is not necessary to preserve any of the configuration fields when a command field is set in the register, that is, no read-modify-write is required.

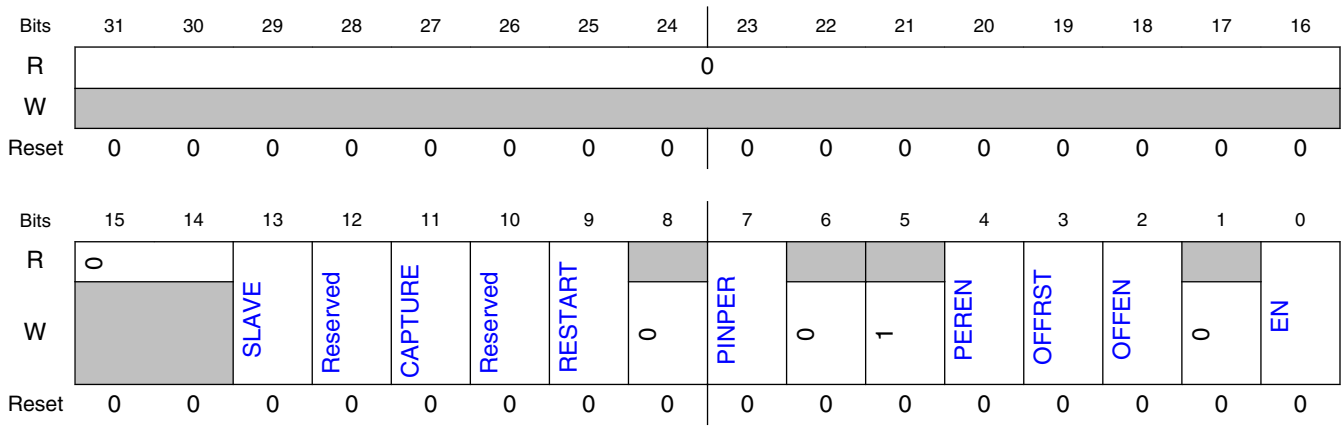
NOTE

The CAPTURE and RESTART fields and bits 12 and 10 must be 0 in order to write to the other fields in this register.

11.6.4.1.100.1 Offset

Register	Offset
ATCR	400h

11.6.4.1.100.2 Diagram



11.6.4.1.100.3 Fields

Field	Description
31-14 —	Reserved
13 SLAVE	Enable Timer Slave Mode 0 - The timer is active and all configuration fields in this register are relevant. 1 - The internal timer is disabled and the externally provided timer value is used. All other fields, except CAPTURE, in this register have no effect. CAPTURE can still be used to capture the current timer value.
12 —	Always write 0 to this field Always write 0 to this field.
11 CAPTURE	Capture Timer Value When this field is set, all other fields are ignored during a write. This field automatically clears to 0 after the command completes. **NOTE:** **WARNING:** To ensure that the correct time value is read from the ATVR register, a minimum amount of time must elapse from issuing this command to reading the ATVR register. This minimum time is defined by the greater of either six register clock cycles or six 1588/timestamp clock cycles. 0 - No effect. 1 - The current time is captured and can be read from the ATVR register.
10 —	Always write 0 to this field Always write 0 to this field.
9 RESTART	Reset Timer Resets the timer to zero. This has no effect on the counter enable. If the counter is enabled when this field is set, the timer is reset to zero and starts counting from there. When set, all other fields are ignored during a write. This field automatically clears to 0 after the command completes. RESTART should be used when the timer is enabled. **NOTE:** **WARNING:** The Reset Timer command requires at least 6 clock cycles of either the register clock or the 1588/timestamp clock, whichever is greater, to complete.

Table continues on the next page...

Field	Description
8 —	Reserved
7 PINPER	Enables event signal output external pin frc_evt_period assertion on period event Enables event signal output assertion on period event. **NOTE:** Not all devices contain the event signal output. See the chip configuration details. 0 - Disable. 1 - Enable.
6 —	Reserved
5 —	This field must be written always with one **NOTE:** This field must be written always with one.
4 PEREN	Enable Periodical Event 0 - Disable. 1 - A period event interrupt can be generated (EIR[TS_TIMER]) and the event signal output is asserted when the timer wraps around according to the periodic setting ATPER. The timer period value must be set before setting this bit. Not all devices contain the event signal output. See the chip configuration details.
3 OFFRST	Reset Timer On Offset Event 0 - The timer is not affected and no action occurs, besides clearing OFFEN, when the offset is reached. 1 - If OFFEN is set, the timer resets to zero when the offset setting is reached. The offset event does not cause a timer interrupt.
2 OFFEN	Enable One-Shot Offset Event 0 - Disable. 1 - The timer can be reset to zero when the given offset time is reached (offset event). The field is cleared when the offset event is reached, so no further event occurs until the field is set again. The timer offset value must be set before setting this field.
1 —	Reserved
0 EN	Enable Timer 0 - The timer stops at the current value. 1 - The timer starts incrementing.

11.6.4.1.101 Timer Value Register (ATVR)

11.6.4.1.101.1 Offset

Register	Offset
ATVR	404h

11.6.4.1.101.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ATIME															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATIME															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.101.3 Fields

Field	Description
31-0 ATIME	A write sets the timer. A read returns the last captured value. To read the current value, issue a capture command (i.e., set ATCR[CAPTURE]) prior to reading this register.

11.6.4.1.102 Timer Offset Register (ATOFF)

11.6.4.1.102.1 Offset

Register	Offset
ATOFF	408h

11.6.4.1.102.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.102.3 Fields

Field	Description
31-0 OFFSET	Offset value for one-shot event generation. When the timer reaches the value, an event can be generated to reset the counter. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds.

11.6.4.1.103 Timer Period Register (ATPER)

11.6.4.1.103.1 Offset

Register	Offset
ATPER	40Ch

11.6.4.1.103.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PERIOD															
Reset	0	0	1	1	1	0	1	1	1	0	0	1	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERIOD															
Reset	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

11.6.4.1.103.3 Fields

Field	Description
31-0 PERIOD	Value for generating periodic events Value for generating periodic events. Each instance the timer reaches this value, the period event occurs and the timer restarts. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds. The value should be initialized to 1,000,000,000 (1×10^9) to represent a timer wrap around of one second. The increment value set in ATINC should be set to the true nanoseconds of the period of clock ts_clk, hence implementing a true 1 second counter. **NOTE:** The value of PERIOD has the following constraint: 2^{32} – ENET_ATINC[INC_COR] – 3×ENET_ATINC[INC] ≥ PERIOD > 0.

11.6.4.1.104 Timer Correction Register (ATCOR)

11.6.4.1.104.1 Offset

Register	Offset
ATCOR	410h

11.6.4.1.104.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	COR														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.104.3 Fields

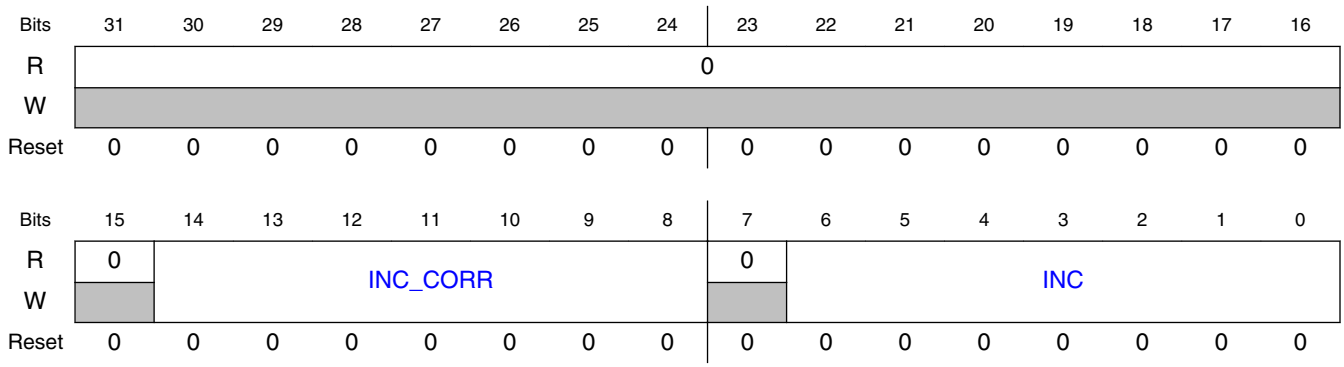
Field	Description
31 —	Reserved
30-0 COR	Correction Counter Wrap-Around Value Defines after how many timer clock cycles (ts_clk) the correction counter should be reset and trigger a correction increment on the timer. The amount of correction is defined in ATINC[INC_CORR]. A value of 0 disables the correction counter and no corrections occur. **NOTE:** This value is given in clock cycles, not in nanoseconds as all other values.

11.6.4.1.105 Time-Stamping Clock Period Register (ATINC)

11.6.4.1.105.1 Offset

Register	Offset
ATINC	414h

11.6.4.1.105.2 Diagram



11.6.4.1.105.3 Fields

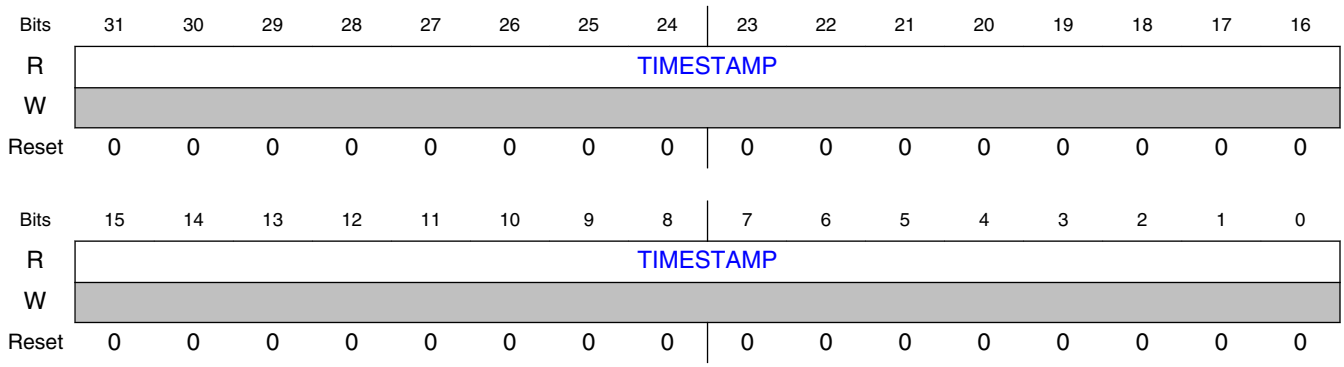
Field	Description
31-15 —	Reserved
14-8 INC_CORR	Correction Increment Value This value is added every time the correction timer expires (every clock cycle given in ATCOR). A value less than INC slows down the timer. A value greater than INC speeds up the timer.
7 —	Reserved
6-0 INC	Clock Period Of The Timestamping Clock (ts_clk) In Nanoseconds The timer increments by this amount each clock cycle. For example, set to 10 for 100 MHz, 8 for 125 MHz, 5 for 200 MHz. **NOTE:** For highest precision, use a value that is an integer fraction of the period set in ATPER.

11.6.4.1.106 Timestamp of Last Transmitted Frame (ATSTMP)

11.6.4.1.106.1 Offset

Register	Offset
ATSTMP	418h

11.6.4.1.106.2 Diagram



11.6.4.1.106.3 Fields

Field	Description
31-0 TIMESTAMP	Timestamp of the last frame transmitted by the core that had TxBD[TS] set the ff_tx_ts_frm signal asserted from the user application Timestamp of the last frame transmitted by the core that had TxBD[TS] set . This register is only valid when EIR[TS_AVAIL] is set.

11.6.4.1.107 Timer Global Status Register (TGSR)

11.6.4.1.107.1 Offset

Register	Offset
TGSR	604h

11.6.4.1.107.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												TF3	TF2	TF1	TF0
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.107.3 Fields

Field	Description
31-4 —	Reserved
3 TF3	Copy Of Timer Flag For Channel 3 0 - Timer Flag for Channel 3 is clear 1 - Timer Flag for Channel 3 is set
2 TF2	Copy Of Timer Flag For Channel 2 0 - Timer Flag for Channel 2 is clear 1 - Timer Flag for Channel 2 is set
1 TF1	Copy Of Timer Flag For Channel 1 0 - Timer Flag for Channel 1 is clear 1 - Timer Flag for Channel 1 is set
0 TF0	Copy Of Timer Flag For Channel 0 0 - Timer Flag for Channel 0 is clear 1 - Timer Flag for Channel 0 is set

11.6.4.1.108 Timer Control Status Register (TCSR0 - TCSR3)

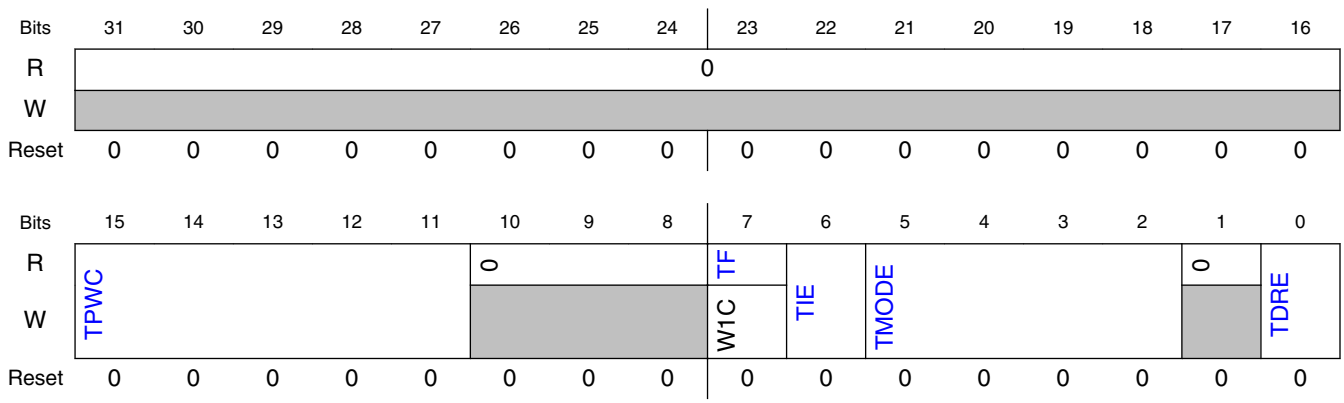
11.6.4.1.108.1 Offset

Register	Offset
TCSR0	608h
TCSR1	610h

Table continues on the next page...

Register	Offset
TCSR2	618h
TCSR3	620h

11.6.4.1.108.2 Diagram



11.6.4.1.108.3 Fields

Field	Description
31-16 —	Reserved
15-11 TPWC	Timer PulseWidth Control Specifies the pulse width associated with TMODE values of 1110 or 11X1. Updating this field takes a few cycles to register because it is synchronized to the 1588 clock. When changing this field: 1. Always disable the channel and read the TMODE field to verify that the channel is disabled. 2. Set TPWC to the desired value. 3. Reenable the channel. **NOTE:** This field is not supported in every instance. The following table includes only supported registers. 00000 - Pulse width is one 1588-clock cycle. 00001 - Pulse width is two 1588-clock cycles. 00010 - Pulse width is three 1588-clock cycles. 00011 - Pulse width is four 1588-clock cycles. 11111 - Pulse width is 32 1588-clock cycles.

Instance	Field supported in	Field not supported in
ENET	TCSR0–TCSR3	—
ENET1G	—	TCSR0–TCSR3

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>10-8
—</td><td>Reserved</td></tr>
<tr><td>7
TF</td><td>Timer Flag
Sets when input capture or output compare occurs. This flag is double buffered between the module clock and 1588 clock domains. When this field is 1, it can be cleared to 0 by writing 1 to it.
0 - Input Capture or Output Compare has not occurred.
1 - Input Capture or Output Compare has occurred.</td></tr>
<tr><td>6
TIE</td><td>Timer Interrupt Enable
0 - Interrupt is disabled
1 - Interrupt is enabled</td></tr>
<tr><td>5-2
TMODE</td><td>Timer Mode
Updating the Timer Mode field takes a few cycles to register because it is synchronized to the 1588 clock. The version of Timer Mode returned on a read is from the 1588 clock domain. When changing Timer Mode, always disable the channel and read this register to verify the channel is disabled first.
NOTE: The descriptions of the field settings vary by module instance.
<table>
<tr><th>Instance</th><th>Field value and description</th></tr>
<tr><td>ENET</td><td>0000 - Timer Channel is disabled.
0001 - Timer Channel is configured for Input Capture on rising edge.
0010 - Timer Channel is configured for Input Capture on falling edge.
0011 - Timer Channel is configured for Input Capture on both edges.
0100 - Timer Channel is configured for Output Compare - software only.
0101 - Timer Channel is configured for Output Compare - toggle output on compare.
0110 - Timer Channel is configured for Output Compare - clear output on compare.
0111 - Timer Channel is configured for Output Compare - set output on compare.
1000 - Reserved
1010 - Timer Channel is configured for Output Compare - clear output on compare, set output on overflow.
10x1 - Timer Channel is configured for Output Compare - set output on compare, clear output on overflow.
110x - Reserved
1110 - Timer Channel is configured for Output Compare - pulse output low on compare for 1 to 32 1588-clock cycles as specified by TPWC.
1111 - Timer Channel is configured for Output Compare - pulse output high on compare for 1 to 32 1588-clock cycles as specified by TPWC.</td></tr>
<tr><td>ENET1G</td><td>0000 - Timer Channel is disabled.
0001 - Timer Channel is configured for Input Capture on rising edge.
0010 - Timer Channel is configured for Input Capture on falling edge.
0011 - Timer Channel is configured for Input Capture on both edges.</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

Field	Description
	<table><tr><th>Instance</th><th>Field value and description</th></tr><tr><td></td><td>0100 - Timer Channel is configured for Output Compare - software only. 0101 - Timer Channel is configured for Output Compare - toggle output on compare. 0110 - Timer Channel is configured for Output Compare - clear output on compare. 0111 - Timer Channel is configured for Output Compare - set output on compare. 1000 - Reserved 1010 - Timer Channel is configured for Output Compare - clear output on compare, set output on overflow. 10x1 - Timer Channel is configured for Output Compare - set output on compare, clear output on overflow. 110x - Reserved 1110 - Timer Channel is configured for Output Compare - pulse output low on compare for one 1588-clock cycle. 1111 - Timer Channel is configured for Output Compare - pulse output high on compare for one 1588-clock cycle.</td></tr></table>
1 —	Reserved
0 TDRE	Timer DMA Request Enable 0 - DMA request is disabled 1 - DMA request is enabled

11.6.4.1.109 Timer Compare Capture Register (TCCR0 - TCCR3)

11.6.4.1.109.1 Offset

Register	Offset
TCCR0	60Ch
TCCR1	614h
TCCR2	61Ch
TCCR3	624h

11.6.4.1.109.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TCC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TCC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.6.4.1.109.3 Fields

Field	Description
31-0 TCC	Timer Capture Compare This register is double buffered between the module clock and 1588 clock domains. When configured for compare, the 1588 clock domain updates with the value in the module clock domain whenever the Timer Channel is first enabled and on each subsequent compare. Write to this register with the first compare value before enabling the Timer Channel. When the Timer Channel is enabled, write the second compare value either immediately, or at least before the first compare occurs. After each compare, write the next compare value before the previous compare occurs and before clearing the Timer Flag. The compare occurs one 1588 clock cycle after the IEEE 1588 Counter increments past the compare value in the 1588 clock domain. If the compare value is less than the value of the 1588 Counter when the Timer Channel is first enabled, then the compare does not occur until following the next overflow of the 1588 Counter. If the compare value is greater than the IEEE 1588 Counter when the 1588 Counter overflows, or the compare value is less than the value of the IEEE 1588 Counter after the overflow, then the compare occurs one 1588 clock cycle following the overflow. When configured for capture, the value of the IEEE 1588 Counter is captured into the 1588 clock domain and then updated into the module clock domain, provided the Timer Flag is clear. Always read the capture value before clearing the Timer Flag.

11.7 Ethernet Quality Of Service (ENET_QOS)

11.7.1 Overview

NOTE

This section and all its sub-sections are Synopsys Proprietary. Used with permission.

The Ethernet Quality Of Service module supports the 10/100/1000Mbps applications in compliance with IEEE802.3-2015 specifications. It can support advanced applications such as Time-Sensitive Networking (TSN) and Audio Video Bridge (AVB). Other prominent features of the MAC core include a flexible receive parser, media clock recovery and generation, and safety.

11.7.1.1 Block diagram

The following figure shows the block diagram of this module. The module has four main blocks to perform all functions. The AHB interface is connected to all DMA channels. The DMA arbiter helps in arbitration of all paths (transmit and receive) in channels. Each channel has a separate set of control and status registers (CSR) for managing the transmit and receive functions, descriptor handling, and interrupt handling.

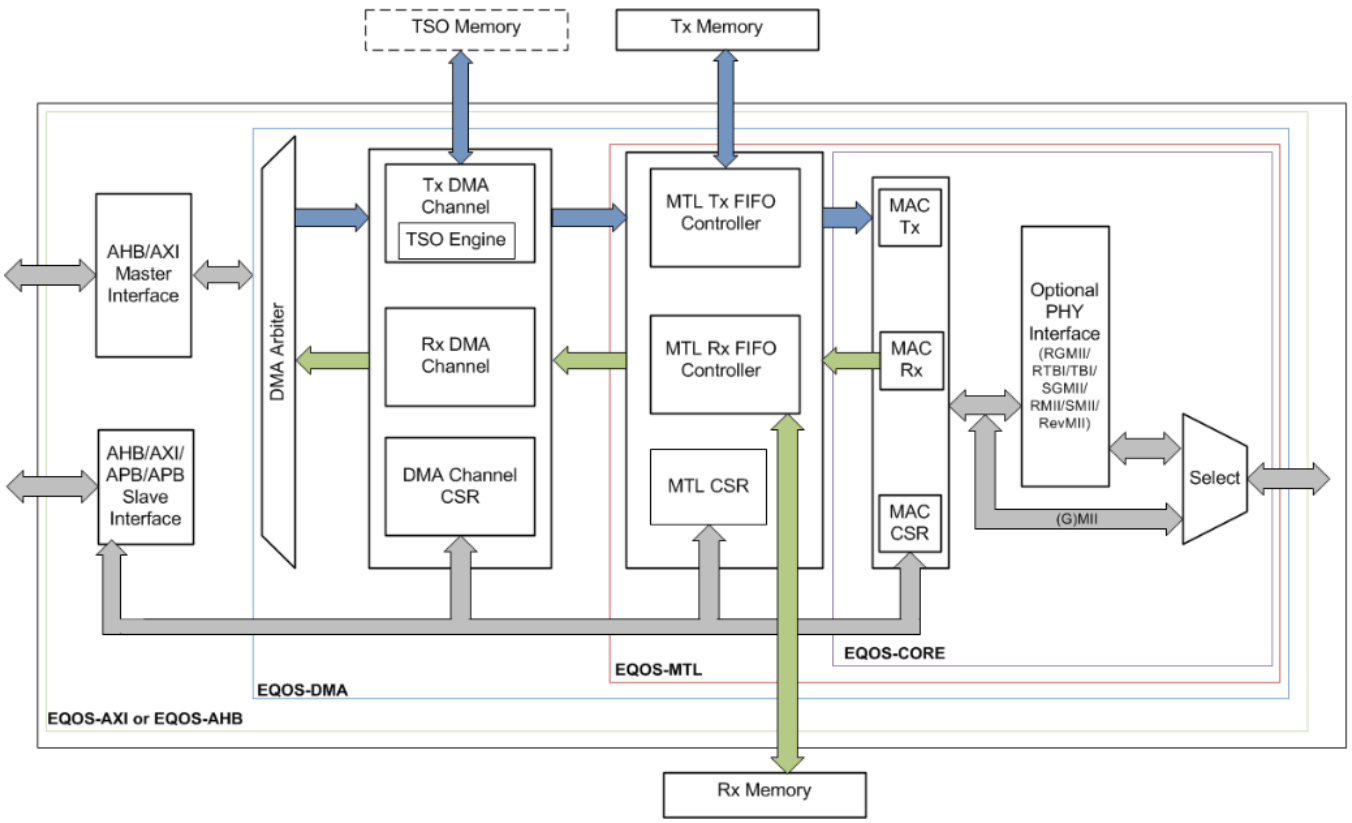


Figure 11-83. System-Level Block Diagram

11.7.1.2 Features

This module supports the following features in addition to the default features defined in IEEE802.3 specification

- RMII (10/100Mbps), RGMII (10/100/1000Mbps)
- Time aware shaper (IEEE802.1Qbv), Time synchronization (IEEE1588-2008), and Frame pre-emption (IEEE802.1Qbu) for Time-Sensitive Networking (TSN)
- Media clock recovery and generation for AVB
- RMII specification version 1.2 from RMII consortium
- Separate interface for Transmit, Receive and Control path
- Broadcast/Multicast Packet duplication
- Full duplex flow control operations (IEEE 802.3x pause packet and priority flow control)
- Network statistics with MAC management (RMON) counters
- Flexibility to control the Pulse per second (PPS) output signal
- MDIO (clause 22 and Clause 45) interface for configuration and management of PHY device
- Preamble and start of packet data (SFD) insertion/deletion
- Automatic CRC and pad generation/stripping option, option to disable CRC checking
- Programmable inter-packet gap
- Source address insertion/replacement
- VLAN insertion, replacement and deletion in transmitted packets with per-packet or static-global control
- Insertion, replacement and deletion of up to two VLAN tags
- Insertion, replacement and deletion of queue/channel based VLAN tags
- IEEE802.1Q VLAN tag detection and option to delete from received packets
- Programmable safety watchdog timeout limits
- Flexible address filtering modes:
    - Up to 2 additional 48-bit perfect (DA) address filters with masks for each byte
    - Up to 2 48-bit SA address comparison check with masks for each byte
    - 64-bit Hash filter for multicast and unicast (DA) addresses
    - Option to pass all multi-cast addressed packets
    - Promiscuous mode to pass all packets without any filtering for network monitoring
    - Pass all incoming packets (as per filter) with a status report
- Additional packet filtering:
    - VLAN tag-based: Perfect match and Hash-based filtering. Filtering based on either outer or inner VLAN tag is possible
    - Layer 3 and Layer 4-based: TCP or UDP over IPv4 or IPv6
    - Extended VLAN tag based filtering 4 filter selections

11.7.2 Functional Description

11.7.2.1 Architecture

This section describes the interfaces, protocols, and functionality. Following are the main blocks of the ENET QOS block.

- DMA Controller
- MAC Transaction Layer (MTL)
- MAC
- Interrupts

11.7.2.1.1 DMA controller

The DMA has independent Transmit (Tx) and Receive (Rx) engines, and a control and status register (CSR) space. The Tx engine transfers data from the system memory to the MTL interface, whereas the Rx engine transfers data from the MTL interface to the system memory. The DMA engine uses descriptors to efficiently move data from source to destination with minimal Host intervention. The DMA is designed for packet-oriented data transfers such as packets in Ethernet. The DMA controller can be programmed to interrupt the Host for situations such as Packet Transmit and Receive Transfer completion, and other normal or error conditions.

The DMA and Host communicate through the following two data structures:

- Control and Status registers (CSR)
- Descriptor lists and data buffers

The DMA supports up to 5 Tx and 5 Rx Descriptor lists (or DMA channels). The base address of each list is written to the respective Tx Descriptor List Address register and Rx Descriptor List Address register. The descriptor list is forward linked and the next descriptor is always considered at a fixed offset to the current one. The offset is controlled by the DSL field of DMA_Ch[n]_Control register. The number of descriptors in the list is programmed in the respective Tx (or Rx) Descriptor Ring Length register. Once the DMA processes the last descriptor in the list, it automatically jumps back to the descriptor in the List Address register to create a descriptor ring.

The descriptor lists reside in the physical memory address space of the application. Each descriptor can point to a maximum of two buffers in the system memory. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.

A data buffer resides in the application physical memory space and consists of an entire packet or part of a packet but cannot exceed a single packet. Buffers contain only data. Buffer status is maintained in the descriptor. Data chaining refers to packets that span multiple data buffers. However, a single descriptor cannot span multiple packets. The DMA skips to the data buffer of next packet when end of packet (EOP) is detected.

This module supports the ring structure for the DMA descriptor. For more information on Descriptor, see the Descriptors.

Following sections describes the operation of DMA Controller.

11.7.2.1.1.1 DMA Controller Bus Burst Access

The DMA engines attempt to transfer data in a burst of maximum size as programmed in the PBL bit fields of Transmit Control and Receive Control registers of the respective DMA. The Rx and Tx descriptors are always accessed in the maximum possible (limited by PBL or 16 * 8/bus width) burst length for 16 bytes to be read. The burst transfers initiated by the DMA can be split into multiple burst transfers as per the AHB requirements and the settings of DMA_SysBus_Mode register.

The Tx DMA initiates a data transfer only when sufficient space is available in the MTL Tx Queue to accommodate either of the following:

- Bytes corresponding to the configured burst (PBL * bus_width/8)
- Remaining bytes in Tx Buffer without EOP
- Number of bytes till EOP

The Rx DMA initiates a data transfer in the following conditions:

- Sufficient data is available in MTL Rx Queue to accommodate the configured burst
- EOP (when it is less than the configured burst length) is detected in the Rx Queue

11.7.2.1.1.2 DMA application data buffer alignment

The Tx and Rx data buffers do not have any restrictions on start address alignment. For example, in systems with 32-bit memory, the start address for buffers can be aligned to any of the four bytes. However, the DMA always initiates write transfers with address aligned to the bus width and dummy data (old data) in the invalid byte lanes. This typically happens during the transfer of the beginning or end of an Ethernet packet. For the receiver, the software driver should discard the dummy bytes based on the start address of the buffer and size of the packet.

Table 11-94. Data buffer alignment examples

Examples	
Buffer Read	If the Tx buffer address is 32'h00000FF2 (for 32-bit data bus), and 15 bytes need to be transferred, the DMA reads five full words from address 32'h00000FF0, but when transferring data to the MTL Tx queue, the extra bytes (the first two bytes) are dropped or ignored. Similarly, the last 3 bytes of the last

Table continues on the next page...

Table 11-94. Data buffer alignment examples (continued)

	transfer are also ignored. The DMA always ensures that it transfers a full 32-bit data to the MTL Tx queue, unless it is the end of packet.
Buffer Write	If the Rx buffer address is 32'h0000FF2 (for 64-bit data bus) and 16 bytes of a received packet need to be transferred, the DMA writes 3 full words from address 32'h00000FF0. However, the first 2 bytes of the first transfer and the last 6 bytes of the third transfer have dummy data. The DMA considers the offset address only if it is the first Rx buffer of the packet. The DMA ignores the offset address and performs full word writes for the middle and the last Rx buffer of the packet.

11.7.2.1.1.3 DMA buffer size calculations

The DMA does not update the size fields in the Tx and Rx descriptors. The DMA updates only the status fields (RDES and TDES) of the descriptors. The driver has to perform the size calculations.

The Tx DMA transfers the exact number of bytes (indicated by buffer size field of TDES2) towards the MAC. If a descriptor is marked as first (FD bit of TDES3 is set), the DMA marks the first transfer from the buffer as SOP. If a descriptor is marked as last (LD bit of TDES3), the DMA marks the last transfer from that data buffer as EOP to the MTL.

The Rx DMA transfers data to a buffer until the buffer is full or the end of packet is received from the MTL. When the FD bit of a descriptor is set, the amount of valid data in a buffer is accurately indicated by the buffer size field (programmed in DMA Channel Receive Control register) minus the data buffer pointer offset. The offset is zero when the data buffer pointer is aligned to the data bus width. If a descriptor is marked as last, the buffer may not be full (as indicated by the buffer size in Bits [14:1] of Receive Control register). To compute the amount of valid data in this final buffer, the driver must read the packet length (PL bits of RDES3[14:0]) and subtract the sum of the buffer sizes of the preceding buffers in this packet. The Rx DMA always transfers the start of next packet with a new descriptor.

11.7.2.1.2 MAC transaction layer

The MAC Transaction Layer (MTL) provides the FIFO memory Interface to buffer and regulate the packets between system memory and the MAC. It also enables the data to be transferred between the system clock and MAC clock domains. The MTL layer has two

data paths: Transmit path and Receive Path. The size of the FIFO is 8 Kb for Tx and Rx. The MTL communicates with the host through Application Transmit Interface (ATI) on the Transmit path and Application Receive Interface (ARI) on the receive path.

11.7.2.1.2.1 Transmit path

Internal DMA handles all transactions for the transmit path through ATI. Internal DMA pushes the Ethernet packets read from system memory into the corresponding queue. The packet is then popped out and transferred to the MAC when the queue threshold is reached (threshold mode) or complete packet is in the queue (store-and-forward mode). When EOP is transferred, the status of the transmission is taken from the MAC and transferred back to the internal DMA.

11.7.2.1.2.1.1 Transmit control word

The following control information related to packet transmission is provided as a part of the Control Word through ATI interface:

- Packet Length (if DCB is enabled with WFQ scheduling algorithm)
- CRC Pad Control
- Source Address Insertion Control
- VLAN Insertion and Replacement Control and VLAN Tag for Outer and Inner VLAN Tags
- TCP/IP Checksum Insertion Control
- One-step Timestamping Control Correction
- Transmit Timestamp Enable

11.7.2.1.2.1.2 Transmit operation

The following two modes of operation triggers reading of the data towards the MAC:

- Threshold mode: In Threshold (or cut-through) mode, as soon as the number of bytes in the Queue cross the configured threshold level (or when the end of packet is written before the threshold is crossed), the data is ready to be popped out and forwarded to the MAC. The threshold level is configured by using the TTC bits of MTL_TxQ0_Operation_Mode register corresponding to an MTL queue.
- Store-and-forward mode:

In store-and-forward mode, the MTL pops out the packet towards the MAC only when one or more of the following conditions are true:

- A complete packet is stored in the Queue
- The Tx FIFO becomes almost full
- The ATI watermark becomes low

The watermark becomes low when the requested queue does not have space to accommodate the requested burst length on the ATI. Therefore, the MTL when operating in the store-and-forward mode allows packet transmission even if the packet length is bigger than the Tx queue size.

The application can flush complete content of the Tx queue by setting Bit 0 (FTQ) of Transmit Operation Mode register corresponding to an MTL queue. This bit is self-clearing and initializes the Queue pointers to the default state. If the FTQ bit is set during a packet transfer from the MTL to the MAC, the MTL stops further transfer because the queue is considered to be empty. Therefore, an underflow event occurs at the MAC transmitter.

11.7.2.1.2.1.3 Initialization flow

Upon reset, the MTL is ready to manage the flow of data to and from the DMA, and MAC. For configurations with a single Transmit Queue, there are no initialization requirements for enabling the MTL. For configurations with multiple Transmit Queues, initialize the queue size for each of the queues by programming the TQS bits of MTL_TxQ0_Operation_Mode register corresponding to a Transmit queue. Also, initialize the MAC block. Internal DMA controllers must be individually enabled through their respective CSRs.

11.7.2.1.2.2 Receive path

The MTL Rx module receives the packets from the MAC and pushes them into the Rx Queue. The status (fill level) of the queue is indicated to DMA when it crosses the configured Receive threshold (RTC bits[1:0] of MTL_RxQ0_Operation_Mode register of corresponding MTL queue), or the complete packet is received. The MTL also indicates the fill level of the queue so that the DMA can initiate preconfigured burst transfers towards the AHB interface.

For more information on queues, see Multiple Channels and Queues Support.

11.7.2.1.2.2.1 Receive Operation

During a Receive operation, the MTL is a slave for the MAC. The general sequence of events is as follows:

1. When the MAC receives a packet, it indicates the availability of receive data.
2. The MAC indicates the SOP and EOP delimiters.
3. The MTL accepts the data and pushes it into corresponding Rx queue.
4. After the EOP is transferred, the MAC drives the status word which is also pushed into the corresponding Rx queue by the MTL.

5. If IEEE 1588 timestamp feature is enabled and the 64-bit timestamp is available along with the packet status, it is pushed into the Rx queue as a part of the status word. Therefore, two additional locations are taken per packet to store the timestamp in the Rx queue.
6. The MTL takes the data out of the queue and sends it to the DMA depending on the following modes:

11.7.2.1.2.2.1.1 Threshold mode

In the (default) Threshold mode, the MTL reads the data and indicates its availability to the application or DMA when one of the following occurs:

- Data bytes equal to the threshold amount are written to the Rx queue (RTC bits[1:0] of MTL_RxQ0_Op-eration_Moderegister of corresponding MTL queue)
- A full packet of data is received into the queue

11.7.2.1.2.2.1.2 Store and forward mode

In the store-and-forward mode (when Bit 5 of MTL_RxQ0_Operation_Mode register of a queue is set to 1), the initial Rx queue locations are reserved for the status words before writing the SOP. A packet is read out only after it is completely written into the Rx queue. In this mode, all error packets are dropped (if configured through Bit 4 of MTL_RxQ0_Operation_Mode register of a queue) such that only valid packets are read and forwarded to the application.

11.7.2.1.2.2.2 Multi packet receive operation

In Threshold mode, the packet status is available immediately after the packet data. In store-and-forward mode, the packet data is available after the packet status. The MTL is capable of storing any number of packets into the queue as long as it is not full.

If the MAC receives a packet when the corresponding Rx queue is full, the MTL ignores that packet and overflow pulse is generated on the corresponding Rx queue. In addition, the MTL increments the overflow counter in the MTL_RxQ0_Missed_Packet_Overflow_Cnt of corresponding queue.

11.7.2.1.2.2.3 Error handling in receive operation

Following things would happen, If the MTL Rx queue is full before it receives the EOP data from the MAC.

1. An overflow is declared
2. The whole packet (including the status word) is dropped

3. The overflow counter in the DMA (Overflow Counter register of corresponding MTL queue) is incremented.

This is true even if Bit 4 (FEP) of MTL_RxQ0_Operation_Mode register of corresponding MTL queue is set.

If the start address of such a packet has already been transferred to the Read Controller, the rest of the packet is dropped and a dummy EOP is written to the queue along with the status word with overflow status. The status indicates a partial packet because of overflow. In such packets, the Packet Length field is invalid. If the MTL Receive Queue is configured to operate in the store-and-forward mode and the length of the received packet is more than the queue size, overflow occurs and all such packets are dropped.

The MTL Rx Control logic can filter error and undersized packets, if enabled by using the FEP and FUP bits of MTL_RxQ0_Operation_Mode register of corresponding MTL queue. If the start address of such a packet has already been transferred to the Rx Queue Read Controller, that packet is not filtered. The start address of the packet is transferred to the Read Controller after the packet crosses the receive threshold set by Bits[1:0] of MTL_RxQ0_Operation_Mode register of corresponding MTL queue.

If the MTL Receive Queue is configured to operate in the store-and-forward mode, all error packets can be filtered and dropped. For the application or DMA to flush the error packet being read from the queue, it must assert the flush signal. The MTL then stops transferring data to the application (DMA). It internally reads the rest of the packet and drops it. The MTL then starts the transfer of next packet (if available).

11.7.2.1.3 MAC

MAC can support RMII or RGMII Phy interface. MAC consists of Mac Transmit interface (MTI), MAC Receive Interface (MRI) and MAC Control Interface (MCI).

11.7.2.1.3.1 MAC transmission

The MAC transmission process is as follows:

1. Transmission is initiated when the MAC Transaction Layer (MTL) pushes in data with the SOP signal asserted.
2. When the SOP signal is detected, the MAC accepts the data and begins transmitting to the RMII or RGMII.
3. After the EOP is transferred to the MAC, the MAC does one of the following:
    - The MAC completes the normal transmission and gives the transmission status to the MTL.

- If a normal collision (in half-duplex mode) occurs during transmission, the MAC gives the Transmit Status with retry bit set to the MTL. The MAC gives the Retry request till one of the following is true:
  - Packet is successfully transmitted
  - Maximum retry requests expire
  - When maximum retry requests expire, the MAC aborts the packet transmission with Excessive Collision Transmit Status. The MAC accepts and drops all further data until the next SOP is received. The MTL block should retransmit the same packet from SOP on observing a Retry request (in the Status) from the MAC.
- If any one of the following happens, the MAC aborts the packet transmission:
  - No carrier (half-duplex mode)
  - Loss of carrier (half-duplex mode)
  - Excessive deferral (half-duplex mode)
  - Late collisions (half-duplex mode)
  - Jabber
  - The MAC accepts and drops all further data until the next SOP is received

4. The MAC issues an underflow status if the MTL is not able to provide the data continuously during the transmission. The MAC accepts and drops all further data until the next SOP is received.
5. During the normal transfer of a packet from MTL, if the MAC receives a SOP without getting an EOP for the previous packet, it ignores the SOP and considers the new packet as continuation of the previous packet.

11.7.2.1.3.2 MAC reception

A receive operation is initiated when the MAC detects an SFD on RMII, RGMII, or MII. The MAC strips the preamble and SFD before proceeding to process the packet. The header fields are checked for filtering and the FCS field used to verify the CRC for the packet. The received packet is stored in a shallow buffer until the address filtering is performed. The packet is dropped in the MAC if it fails the address filter.

11.7.2.1.4 Interrupts

This module supports interrupt coalescing which reduces the number of interrupts generated by IP and reduces the CPU load. Various interrupt event gets captured in status register. If interrupt enable bit of an interrupt is set then corresponding interrupt will be generated based on event status in status registers. INTM (interrupt mode) bit decides that interrupt signal is a level signal or pulse signal.

Interrupt output signals are shown below which are synchronous to control and Status register (CSR) clock.

- Sbd_intr_o – common interrupt
- Sbd_perch_tx_intr_o[max_dma_ch] – Interrupt per Tx channel
- Sbd_perch_rx_intr_o [max_dma_ch] – Interrupt per Rx channel

NOTE

max_dma_ch is equal to number of Tx/Rx queues which is 2.

The sbd_intr_o common interrupt is level signal. When sbd_intr_o interrupt is asserted, then corresponding interrupt event source can be found in DMA_Interrupt status register. DMA_Interrupt status register is read only register which contains the interrupt event source bits corresponding to each DMA channel (Tx and Rx queue pair), MAC transaction layer and MAC blocks. Software must then read the following registers corresponding to the bits that are set.

- MAC_Interrupt_Status register
- MTL_Interrupt_Status register
- DMA_CH(n)_Status register

sbd_intr_o interrupt gets de-asserted only when all the enabled interrupt events are cleared in their respective status registers and correspondingly all the bits in the DMA_Interrupt_Status register are cleared.

The DMA_CH[n]_Status register captures all the interrupt events of that Tx DMA and Rx DMA channel pair. The DMA_CH[n]_Interrupt_Enable register contains the corresponding enable bits for each of the interrupt event. There are two groups of interrupts in the DMA channel namely Normal and Abnormal interrupts. They are indicated by bits[15:14] of DMA_CH[n]_Status register respectively. The normal group is for events that happen during the normal transfer of packets (TI, RI, TBU) while the abnormal interrupt events are for error events

Interrupts are not queued. If the same interrupt event occurs again before the driver responds to the previous one, no additional interrupts are generated.

The common sbd_intr_o output signal is asserted for the transfer complete interrupts only when the corresponding interrupts are enabled in DMA_CH[n]_Interrupt_Enable register.

This module also supports the following per Channel Transfer Complete interrupt signals.

- sbd_perch_tx_intr_o[max_dma_ch] (Transmit Per Channel Interrupts)
- sbd_perch_rx_intr_o[max_dma_ch] (Receive Per Channel Interrupts)

The behavior of the RI/TI/ sbd_perch_tx_intr_o[]/sbd_perch_rx_intr_o[] changes depending on the settings of INTM field bit[17:16] in DMA_Mode. Following table explains the Transfer Complete Interrupt behavior.

Table 11-95. Transfer Complete Interrupt Behavior

Interrupt Mode	Behavior of the sbd_perch_tx_intr_o[] and sbd_perch_rx_intr_o[]	Behavior of TI/RI and sbd_intr_o
INTM=1	These signals reflect the value of corresponding TI/RI bits in DMA_CH[n]_Status register when the corresponding interrupt enable is set. Hence they are level signals and are cleared by the application by writing 1'b1 to the RI/TI status bits. This signal will not be asserted when the corresponding interrupt enable bit is not set.	The TI/RI is set as explained above. However, the sbd_intr_o signal and NIS status bit are not asserted for any RI/TI events.
INTM=2	In this mode, RI/TI interrupts are queued. These signals reflect the value of corresponding TI/RI bits inDMA_CH[n]_Status register when the corresponding interrupt enable is set. They are level signals and cleared by software by writing 1'b1 to the RI/TI status bits. However, it is set again if another TI/RI event(s) is detected before the TI/RI bits are cleared for the previous event.	The RI/TI status bits are set whenever the Transfer Complete event is detected and gets reset whenever software driver clears those bits by writing 1. However, if another Transfer Complete event is detected before it is cleared (serviced) by the software, then the module automatically set these status bits again. However, the sbd_intr_o signal is not generated based on TI/RI

11.7.2.2 Using PHY interfaces

This module support the following modes:

- RMII 10/100 Mbps interface
- RGMII 10/100/1000 Mbps interface

The internal Phy_intf_sel input signal decides which mode has been selected. Phy_intf_sel signal gets sampled at reset. PS and FES bit of MAC configuration register selects the speed of mode.

The module supports the IEEE802.3-2015 specification for RMII and RGMII specification version 1.2 from RMII consortium.

NOTE

RMII reference clock (50 MHz) can be fed to IP from an external source or internally from PLL on SoC.

The module supports can access the phy registers through Station Management Agent (SMA) which is a two wire Station management interface (MIM). One wire is management data clock (MDC) and other wire is management data input/output (MDIO). As per IEEE 802.3 specification, maximum operating frequency of MDC is 2.5 Mhz which can be derived from system clock using a divider. The CR bit of MAC_MDIO_Address register can be programmed to generate different MDC clock frequency.

SMA supports both MDIO clause 45 and clause 22 frame structure as per IEEE802.3 specification. The C45E bit of MAC_MDIO_Address register can be set to enable the clause 45 frame structure.

11.7.2.3 VLAN and double VLAN insertion, deletion, replacement and tagging

These sections describes the VLAN and Double VLAN features.

11.7.2.3.1 Double VLAN processing

The module can support processing of two VLAN tags in double VLAN tagging processing. These two VLAN tags can also be named as inner (C-VLAN) and outer (S-VLAN) VLAN tag. If packet is having only one tag then it would be considered as outer tag.

It supports the following function in Double VLAN processing feature using the MAC_VLAN_Incl, MAC_Inner_VLAN_Incl and MAC_VLAN_Tag registers.

- Insertion, replacement, or deletion of up to two VLAN tags in the Transmit path.
- Packet filtering and stripping based on any one of the two VLAN Tags in the Receive path. Stripping and providing up to two VLAN Tags in the Receive path as a part of the Receive status.

11.7.2.3.1.1 Transmit path

Following table describes the features supported by the MAC on the Transmit side.

Table 11-96. Double VLAN Processing Features in Transmit Path

Feature	Description
Support for C-VLAN and S-VLAN Tag types	The inner or outer VLAN tag can be of C-VLAN and S-VLAN type. The VLAN type is specified through the CSVL bit of MAC_VLAN_INCL and MAC_INNER_VLAN_INCL registers.

Table continues on the next page...

Table 11-96. Double VLAN Processing Features in Transmit Path (continued)

	The module supports processing of any sequence of outer and inner VLAN tags. **NOTE:** The module does not support the C-VLAN S-VLAN sequence. The MAC does not check whether the packet provided by the application has a valid sequence of the VLAN Tag types or the insertion or replacement operation results in invalid sequence of VLAN Tag type. Therefore, the application must provide correct sequence of VLAN Tag types and program the MAC in such a way that it results in correct sequence of VLAN Tag types in the transmitted packet. The application must ensure the following: • The inner tag should not be S-VLAN when outer C-VLAN Tag insertion is enabled. • The outer tag should not be C-VLAN when inner S-VLAN Tag insertion is enabled. • The inner tag should not be S-VLAN when outer tag should be replaced with C-VLAN. • The outer tag should not be C-VLAN when inner tag should be replaced with S-VLAN.
VLAN Tag deletion	VLAN tag deletion for outer or inner tag can be enabled through VLC field in the MAC_VLAN_Incl or MAC_Inner_VLAN_Incl register, respectively. When VLAN deletion is enabled, the MAC deletes the tag present at the corresponding position. When a packet has only one tag, it is considered as the outer tag. If inner tag deletion is enabled and the packet has only one tag, the MAC does not delete the inner tag.
VLAN Tag Insertion or Replacement	VLAN tag insertion or replacement for outer or inner tag can be enabled through VLC field in the MAC_VLAN_Incl or MAC_Inner_VLAN_Incl register, respectively. When VLAN tag insertion or replacement is enabled, the VLTI bit in the MAC_VLAN_Incl or MAC_Inner_VLAN_Incl register is used to determine whether the VLAN tag should be taken from the register or the Control Word.

11.7.2.3.1.2 Receive path

Following table describes the features supported by the MAC on the receive side and the corresponding bits in the MAC_VLAN_Tag register.

Table 11-97. Double VLAN Processing in Receive Path

Feature	Description
Outer or inner VLAN tag-based filtering	The MAC can filter packets based on the outer or inner VLAN tag through the ERIVLT bit.

Table continues on the next page...

Table 11-97. Double VLAN Processing in Receive Path (continued)

C-VLAN or S-VLAN tag-based filtering	The MAC can filter packets based on the C-VLAN or S-VLAN type based on the ERSVLM bit.
Outer and Inner VLAN Tag stripping	The MAC can strip the outer and inner VLAN Tags from received frame based on the EVLS and EIVLS bits.
16-bit outer and inner VLAN Tag and Type in Rx status	The MAC can provide the 16-bit outer and inner VLAN Tag and Type in the Rx status based on the EVLRXS and EIVLRXS bits, respectively.
Disabling or skipping checking of outer VLAN Tag type	The MAC can disable or skip checking of outer VLAN Tag type to match C-VLAN or SVLAN based on the DOVLTC bit.

11.7.2.3.1.3 Source address and VLAN insertion, replacement, or deletion

The module supports the SA and VLAN insertion, replacement and deletion. The source address (SA) and VLAN fields are Tx packet-related control information that are provided as part of the control word through interfaces. IP supports the feature to insert or replace the source address based on the information in the MAC Address Registers, and also the feature to insert, replace or delete the VLAN fields (VLAN Type and VLAN Tag) based on the setting of the VLTI bit in the MAC_VLAN_Incl register. SA insertion or replacement feature can be enabled for all Transmit packets or selective packets. Similarly, VLAN insertion, replacement, or deletion feature can be enabled for all Tx packets or selective packets.

11.7.2.3.1.4 Programming source address insertion or replacement

The software can use the SA insertion or replacement feature to instruct the MAC to do the following for Tx packets:

- Insert the content of the MAC Address Registers in the SA field
- Replace the content of the SA field with the content of the MAC Address Registers

When SA insertion is enabled, the application must ensure that the packets sent to the MAC do not have the SA field. The MAC does not check whether the SA field is present in the Transmit packet and it inserts the content of MAC Address Registers in the SA field. Similarly, when SA replacement is enabled, the application must ensure that the SA field is present in the packets sent to the MAC.

The MAC replaces the six bytes following the Destination Address field in the Transmit packet with the content of the MAC Address Registers.

- SARC bit of MAC_Configuration register can be program to enable the SA insertion or replacement for all Tx packets.
- SA insertion for selective packet can be enabled by programming the SA Insertion Control field (bits [25:23] of TDES3) in the first Transmit descriptor of the packet. When Bit 25 of TDES3 is set, the SA Insertion Control field indicates insertion or replacement by MAC Address1 registers. When Bit 25 of TDES3 is reset, it indicates insertion or replacement by MAC Address 0 registers.

If MAC Address1 Registers are not enabled, the MAC Address0 registers are used for insertion or replacement irrespective of the value of the most-significant bit of the SA Insertion Control field.

11.7.2.3.1.5 Programming VLAN insertion, replacement, or deletion

The software can use the VLAN insertion, replacement, or deletion feature to instruct the MAC to do the following for Tx packets:

- Delete the VLAN Type and VLAN Tag fields
- Insert or replace the VLAN Type and VLAN Tag fields

Insertion or replacement is done based on the setting of VLTI bit in the MAC_VLAN_Incl register as described in the following table:

Table 11-98. VLAN Insertion or Replacement Based on VLTI Bit

Condition	Description
VLTI bit is set	The MAC inserts or replaces the following: • VLAN Type field (C-VLAN or S-VLAN as indicated by the CSVL bit of MAC_VLAN_Incl register) • VLAN Tag field depending on the Content of the VT field of Transmit context descriptor of the packet
VLTI bit is reset	The MAC inserts or replaces the following: • VLAN Type field (C-VLAN or S-VLAN as indicated by the CSVL bit of MAC_VLAN_Incl register) • VLAN Tag field with the VLT field of MAC_VLAN_Incl register

When VLAN replacement or deletion is enabled, the MAC checks if the VLAN Type field (0x8100 or 0x88a8) is present after the DA and SA fields in the Transmit packet. The replace or delete operation does not occur if the VLAN Type field is not detected in two bytes following the DA and SA fields. However, when VLAN insertion is enabled, the MAC does not check the presence of VLAN Type field in the Transmit packet and just inserts the VLAN Type and VLAN Tag fields.

VLAN insertion, replacement, or deletion feature can be enabled for all Tx packets by programming the VLC and VLP fields of MAC_VLAN_Incl register.

VLAN insertion, replacement, or deletion for selective packets can be enabled by programming the VTIR field of TDES2 Normal Descriptor.

In addition, the VLP (VLAN Priority control) bit must be reset in MAC_VLAN_Incl register (for outer VLAN) and MAC_Inner_VLAN_Incl register (in inner VLAN) for the MAC to take the control inputs from the host, depending on the configuration.

11.7.2.4 Packet Filtering

There are various packets filtering scheme available in MAC receiver. Filtering sequence in their precedence order of received packet is shown in the following figure.

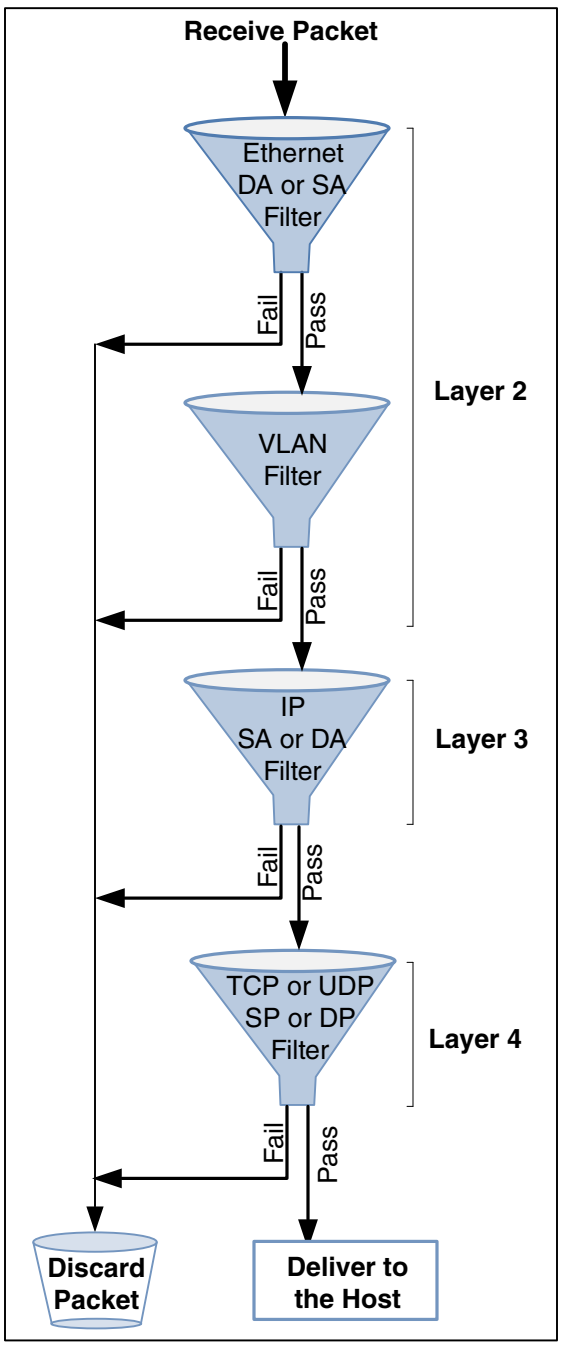


Figure 11-84. Packet Filtering Sequence

If any of the Layer filters are not enabled, that filter is bypassed and the subsequent filter is applied. A packet that fails any of the filters, is discarded. However, the discarded packet can be forwarded to the host based on the register control.

11.7.2.4.1 Source address or destination address filtering

The Address Filtering Module of the MAC checks the source address (SA) and destination address (DA) fields of each incoming packet. Programming of different type of address filtering is described in the following sections.

11.7.2.4.1.1 Unicast destination address filtering

MAC supports up to 3 MAC addresses for unicast perfect filtering. If perfect filtering is selected (HUC bit of MAC_Packet_Filter register is reset), the MAC compares all 48 bits of received unicast address with the programmed MAC address for any match. Default MacAddr0 is always enabled. The MacAddr1 to MacAddr2 addresses are selected with an individual enable bit. Each byte of MacAddr1 to MacAddr2 can be masked during comparison with corresponding received DA byte by setting the corresponding Mask Byte Control bit in the register. This enables group address filtering for the DA.

In hash filtering mode (when HUC bit is set), the MAC performs imperfect filtering for unicast addresses using a 64-bit Hash table. For hash filtering, the MAC uses the upper 6 bits CRC of the received destination address to index the content of the Hash table. A value of 000000 selects Bit 0 of selected register, and a value of 111111 selects Bit 63 of Hash Table register. If the corresponding bit (indicated by the 6-bit CRC) is set to 1, the unicast packet is considered to have passed the Hash filter; otherwise, the packet is considered to have failed the Hash filter.

11.7.2.4.1.2 Multicast destination address filtering

PM bit of MAC_Packet_Filter register can be enabled to pass all multicast packets or disabled to do multicast addresses filtering based on HMC bit of MAC_Packet_Filter register. The multicast address is compared with the programmed MAC Destination Address registers (1–2). Also, Group address filtering is supported.

In Hash filtering mode, the MAC performs imperfect filtering using a 64-bit Hash table. The MAC uses the upper 6-bits CRC of received multicast address to index the content of the Hash table. A value of 000000 selects Bit 0 of selected register and a value of 111111 selects Bit 63 of the Hash Table register. If the corresponding bit is set to 1, the multicast packet is considered to have passed the Hash filter. Otherwise, the packet is considered to have failed the Hash filter.

11.7.2.4.1.3 Hash or perfect address filtering

HPF bit of MAC_Packet_Filter register must be configured for enabling or disabling the destination address filtering by either Hash filter or Perfect filter. HUC and HMC bit of MAC_Packet_Filter register must be configured to select the Hash filtering or Perfect filtering for Unicast/Multicast packet.

11.7.2.4.1.4 Broadcast address filtering

The MAC does not filter any broadcast packets by default. DBF bit of MAC_Packet_filter register must be set to reject the broadcast packets.

11.7.2.4.1.5 Unicast source address filtering

By default, MAC compares the source address field with the value programmed in the source address registers. Bit 30 of MAC address register [1-2] can be configure to use it for source address instead of destination address comparison. The MAC also supports group filtering with source address(SA) . You can filter a group of addresses by masking one or more bytes of the address. The MAC drops the packets that fail the SA filter if the SAF bit is set in MAC_Packet_Filter register. Otherwise, the result of the SA filter is given as a status bit in the Receive Status word. When the SAF bit is set, the SA filter and DA filter result is AND'ed to decide whether the packet needs to be forwarded. This means that the packet is dropped if either filter fails. The packet is forwarded to the application only if the packet passes both filters in-order.

11.7.2.4.1.6 Inverse filtering

For DA and SA filtering, you can invert the filter-match result at the final output by setting the DAIF and SAIF bits of MAC_Packet_Filter register. The DAIF bit is applicable for both Unicast and Multicast DA packets. The result of the unicast or multicast destination address filter is inverted in this mode. Similarly, when the SAIF bit is set, the result of unicast SA filter is reversed.

Following two tables summarize the DA and SA filtering based on the type of packets received.

Table 11-99. Destination Address Filtering

Packet Type	PR	HPF	HUC	DAIF	HMC	PM	DBF	DA Filter Operation
Broadcast	1	X	X	X	X	X	X	Pass

Table continues on the next page...

Table 11-99. Destination Address Filtering (continued)

	0	X	X	X	X	X	0	Pass
	0	X	X	X	X	X	1	Fail
Unicast	1	X	X	X	X	X	X	Pass all packets
	0	X	0	0	X	X	X	Pass on Perfect/ Group filter match
	0	X	0	1	X	X	X	Fail on Perfect/ Group filter match
	0	0	1	0	X	X	X	Pass on Hash filter match
	0	0	1	1	X	X	X	Fail on Hash filter match
	0	1	1	0	X	X	X	Pass on Hash or Perfect/ Group filter match
	0	1	1	1	X	X	X	Fail on Hash or Perfect/ Group filter match
Multicast	1	X	X	X	X	X	X	Pass all packets
	X	X	X	X	X	1	X	Pass all packets
	0	X	X	0	0	0	X	Pass on Perfect/ Group filter match and drop Pause packets if PCF = 0x
	0	0	X	0	1	0	X	Pass on Hash filter match and drop Pause packets if PCF = 0x
	0	1	X	0	1	0	X	Pass on Hash or Perfect/ Group filter match and drop Pause packets if PCF = 0x

Table continues on the next page...

Table 11-99. Destination Address Filtering (continued)

	0	X	X	1	0	0	X	Fail on Perfect/ Group filter match and drop Pause packets if PCF = 0x
	0	0	X	1	1	0	X	Fail on Hash filter match and drop Pause packets if PCF = 0x
	0	1	X	1	1	0	X	Fail on Hash or Perfect/ Group filter match and drop Pause packets if PCF = 0x

Table 11-100. Source Address Filtering

Packet Type	PR	SAIF	SAF	SA Filter Operation
Unicast	1	X	X	Pass all packets.
	0	0	0	Pass status on Perfect or Group filter match but do not drop packets that fail
	0	1	0	Fail status on Perfect or Group filter match but do not drop packet
	0	0	1	Pass on Perfect or Group filter match and drop packets that fail
	0	1	1	Fail on Perfect or Group filter match and drop packets that fail

NOTE

When the RA bit of MAC_Packet_Filter register is set, all packets are forwarded to the system along with the correct result of the address filtering in the Rx Status.

11.7.2.4.2 VLAN Filtering

VLAN tag filter can be done by either perfect match or hash table. Depending upon the configuration, MAC can compare either lower 12 bits or all 16 bits of received VLAN tag for perfect match. MAC forwarded the VLAN tagged packet with match status while it drops all the packets which don't matches.

MAC uses 16-bit VLAN hash table for group address filtering based on the VLAN tag. If hash filtering is enabled then most significant 4-bit of CRC-32 of VLAN tag are used to index the content of the VLAN Hash Table Register. A value of 1 in the VLAN Hash Table register, corresponding to the index, indicates that the VLAN tag of the packet matched and the packet should be forwarded. A value of 0 indicates that VLAN-tagged packet should be dropped.

The MAC also supports the inverse matching for VLAN packets. In the inverse matching mode, when the VLAN tag of a packet matches the perfect or hash filter, the packet should be dropped. If the VLAN perfect and VLAN hash match are enabled, a packet is considered as matched if either the VLAN hash or the VLAN perfect filter matches. When inverse match is set, a packet is forwarded only when both perfect and hash filters indicate mismatch.

Table below shows the different possibilities for VLAN matching and the final VLAN match status. When the RA bit of MAC_Packet_Filter register is set, all packets are received and the VLAN match status is indicated in the VF bit of RDES2 Normal Descriptor (Write-Back Format). When the RA bit is not set and the VTFE bit is set in MAC_Packet_Filter register, the packet is dropped if the final VLAN match status is Fail.

When VLAN VID is programmed to 0 in the VL field of MAC_VLAN_Tag register, all VLAN-tagged packets are considered as perfect matched but the status of the VLAN hash match depends on the VTHM and VTIM bits in MAC_VLAN_Tag register.

Table 11-101. VLAN Match Status

VID	VLAN Perfect Filter Match Result	VTHM Bit	VLAN Hash Filter Match Result	VTIM Bit	Final VLAN Match Status
VID = 0	Pass	0	X	X	Pass
	Pass	1	X	0	Pass
	Pass	1	Fail	1	Pass
	Pass	1	Pass	1	Fail
VID!= 0	Pass	X	X	0	Pass
	Fail	0	X	0	Fail
	Fail	1	Fail	0	Fail
	Fail	1	Pass	0	Pass

Table continues on the next page...

Table 11-101. VLAN Match Status (continued)

VID	VLAN Perfect Filter Match Result	VTHM Bit	VLAN Hash Filter Match Result	VTIM Bit	Final VLAN Match Status
	Fail	0	X	1	Pass
	Pass	X	X	1	Fail
	Fail	1	Pass	1	Fail
	Fail	1	Fail	1	Pass

In this table, X represents any value.

11.7.2.4.3 VLAN Filter Fail Packets Queue

When VLAN filtering is enabled, the VLAN Filter Fail Packets can be routed to a programmable Queue (VFFQ) when Bit RA = 1 or VTFE = 0 of MAC_Packet_Filter register and the enable bit (VFFQE of MAC_RXQ_CTRL4) for the queue is set.

The packets that passes the VLAN filtering, are routed to Rx queues based on the VLAN TAG priority field by programming the PSRQ in the corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers. The packets that fail the VLAN filter are discarded if RA=0 or VTFE=1. However, when RA=1 or VTFE=0, the VLAN filter fail packets are still forwarded to the application. In such a scenario, when the VLAN Filter Fail Queue Enable (VFFQE) bit is set, the VLAN filter fail packets are forwarded to the Rx queue number programmed in the VFFQ bit. If VFFQE=0, the Rx queue number is determined by the VLAN priority mapping as per the PSRQ fields.

Table below shows the Rx Queue routing table for Unicast tagged packets, with DA/SA Filter enabled.

Table 11-102. Rx Queue Routing table for Unicast Tagged Packets

RA	VFTE	SA/DA Filter Result	VLAN Filter Result	VFFQE	Queue Routing
X	X	PASS	PASS	X	PSRQ
0	0	PASS	FAIL	0	PSRQ
0	0	PASS	FAIL	1	VFFQ
0	X	FAIL	X	X	DROPPED
0	1	PASS	FAIL	X	DROPPED
1	X	FAIL	X	0	UFFQ*/PSRQ
1	X	FAIL	X	1	UFFQ*/VFFQ
1	X	PASS	FAIL	0	PSRQ
1	X	PASS	FAIL	1	VFFQ

X : Don't care condition * : When UFFQE is enabled else PSRQ

11.7.2.4.4 Extended Receive VLAN Filtering and Routing

When Extended Rx VLAN filtering and routing is enabled then both perfect filtering and Hash filtering can be enabled. The overall VLAN filter result is based on the Perfect filter result and Hash filter result (if enabled). Filter result is passed to application/software as part of the status bit. Extended Routing will happen only if VLAN filter has passed. Routing is only based on Perfect filter result. Each Perfect filter has a DMA channel enable and DMA channel number filed which should be programmed for routing.

For more details about routing, see details on 'Extended VLAN Based DMA Selection' in Dynamic (Per Packet) Mapping.

11.7.2.4.4.1 Comparison mode

Software/Application has following comparison option for each VLAN tag filter

- It can program the MAC to compare the inner or outer VLAN tag with either 12 bits or 16 bits programmed VID.
- It can choose that VID comparison is for SVLAN or CVLAN type frames, if type check is enabled for a filter.

11.7.2.4.4.2 Filtering

Perfect Filtering is done based on the MAC_VLAN_Tag_Filter registers. For each VLAN Tag Filter, the MAC will compare the relevant VLAN Tag ID and gives a result. If any one of the VLAN Tag Filters gives a match then the frame is considered to have passed the VLAN Tag Filters. If the frame mismatches all the filters, then the frame is considered to have failed the VLAN filter. This behavior is applicable only when the Inverse Filtering is not enabled in MAC_VLAN_Tag_Ctrl Register.

If inverse Filtering is enabled and the frame has mis-matched all the relevant filters then it is considered to have passed the VLAN filter. If the frame matches any one of the relevant filters then it is considered as a fail. If none of the enabled filters can perform a comparison or if none of the filters are enabled, then the frame is bypassed to the application.

The overall filter result and the programming on the VTFE and RA bits of the MAC Filter Register determine if the frame will be dropped or forwarded to the application. If RA = 1 or VTFE = 0, it does not matter if the filter result is a pass or fail. The frame is

always forwarded. If RA = 0 and VTFE = 1, only then, if the VLAN Tag Filter result is a pass does the MAC forward the frame. If the frame is forwarded to the application, then the relevant filter result is indicated through the Status bits.

11.7.2.4.4.3 Filter Status

The Extended Receive VLAN Filtering and Routing feature provides two status bits to indicate the comparison result of the VLAN tags.

By default, the MAC indicates the VLAN Filter Status through one bit in the status – VF field in RDES2. When Extended RX VLAN Filtering and Routing is enabled, two status bits are used to indicate the comparison result of VLAN tags. The Outer VLAN Tag Filter Pass and Inner VLAN Tag Filter Pass bits are defined in the following positions. The status indicated through these bits is highly dependent on the programming as explained below.

In RDES2:

- Bit 15 – Outer VLAN Tag Filter Status
- Bit 14 – Inner VLAN Tag Filter Status

In ARI status: MAC Filter Status:

- Bit 15 – Outer VLAN Tag Filter Status
- Bit 14 – Inner VLAN Tag Filter Status

In MRI Status:

- Bit 47 – Outer VLAN Tag Filter Status
- Bit 46 – Inner VLAN Tag Filter Status

Outer VLAN Tag Filter Status (OTS)

- In perfect Filtering, without inverse filtering enabled, if this bit is set, it indicates that the frame's Outer VLAN Tag has matched one of the VLAN Tag Filters.
- If this bit is reset, it indicates that the frame's Outer VLAN Tag has either failed the relevant Outer VLAN Tag Filters or bypassed them.
- If none of the filters are enabled for Outer VLAN Tag Comparison, then this bit is reset.

- If Inverse Filtering is enabled and this bit is set, then the frame's VLAN Tag has passed all the relevant VLAN Tag Filters. If it is reset, then it has failed at least one of or bypassed all the Filters programmed for Outer VLAN Tag Comparison.
- This bit is valid for both Single and Double VLAN Tagged frames.

Inner VLAN Tag Filter Status (ITS)

- In perfect Filtering, without inverse filtering enabled, if this bit is set, it indicates that the frame's Inner VLAN Tag has matched one of the VLAN Tag Filters.
- If this bit is reset, it indicates that the frame's Inner VLAN Tag has either failed the relevant Inner VLAN Tag Filters or bypassed them.
- If none of the filters are enabled for Inner VLAN Tag Comparison, then this bit is reset.
- If Inverse Filtering is enabled and this bit is set, then the frame's VLAN Tag has passed all the relevant VLAN Tag Filters. If it is reset, then it has failed at least one of or bypassed all the Filters programmed for Inner VLAN Tag Comparison.
- This bit is valid for only Double VLAN Tagged frames, when Double VLAN Processing is enabled.

The application must look at the status bits and the programming to determine if the Frame has passed or failed the VLAN Filter.

Following two tables 'OTS and ITS Bit Values with At Least 1 Perfect Filter Enabled' and 'OTS and ITS Bit Values with Only VLAN Hash Filter Enabled' show the possible Filter combinations and the corresponding filter results. These tables explain the scenarios when Double VLAN Processing and Hash VLAN filter are enabled in the design.

Legend for the table 'OTS and ITS Bit Values with At Least 1 Perfect Filter Enabled'

- VTIM: VLAN Tag Inverse Match Enable – bit 17 in VLAN_Tag_Ctrl Register.
- HFO: Hash Filter enabled for Outer VLAN Tag Comparison - bit 25 and bit 27 in VLAN_Tag_Ctrl Register.
- HFI: Hash Filter enabled for Inner VLAN Tag Comparison – bit 25 and bit 27 in VLAN_Tag_Ctrl Register.
- PFO – Perfect Filter comparison enabled for Outer VLAN Tag - Any of the MAC_VLAN_Tag_Filter Registers is enabled (bit 16 is set) and programmed for Outer VLAN Tag comparison (bit 20 is set to 0).

- PFI – Perfect Filter comparison enabled for Inner VLAN Tag - Any of the MAC_VLAN_Tag_Filter Registers is enabled (bit 16 is set) and programmed for Inner VLAN Tag comparison (bit 20 is set to 1).
- OTS – Outer VLAN Tag Filter Status
- ITS – Inner VLAN Tag Filter Status

Following table shows the possible values of status bits (OTS and ITS) when at least one Perfect filter is enabled.

Table 11-103. OTS and ITS Bit Values with At Least 1 Perfect Filter Enabled

VTIM	HFO	HFI	PFO	PFI	OTS	ITS
0	0	0	0	1	0	1/0
0	0	0	1	0	1/0	0
0	0	0	1	1	1/0	1/0
0	1	0	1	1	1/0	1/0
0	1	0	1	0	1/0	0
0	1	0	0	1	1/0	1/0
0	0	1	1	1	1/0	1/0
0	0	1	1	0	1/0	1/0
0	0	1	0	1	0	1/0
1	0	0	0	1	0	1/0
1	0	0	1	0	1/0	0
1	0	0	1	1	1/0	1/0
1	1	0	1	1	1/0	1/0
1	1	0	1	0	1/0	0
1	1	0	0	1	1/0	1/0
1	0	1	1	1	1/0	1/0
1	0	1	1	0	1/0	1/0
1	0	1	0	1	0	1/0

Following table shows the possible values of status bits (OTS and ITS) when none of the perfect filters are enabled and only the VLAN Hash Filter is enabled.

Table 11-104. OTS and ITS Bit Values with Only VLAN Hash Filter Enabled

VTIM	HFO	HFI	OTS	ITS
0	0	0	0	0
0	1	0	1/0	0
0	0	1	1/0	1/0
1	0	0	1/0	0
1	1	0	1/0	0

Table continues on the next page...

Table 11-104. OTS and ITS Bit Values with Only VLAN Hash Filter Enabled (continued)

VTIM	HFO	HFI	OTS	ITS
1	0	1	1/0	1/0

With no perfect filters enabled, any VLAN packet is considered to have bypassed the perfect filter. If Hash Filter is enabled for one of the Tags, then the respective Status bit depends on the Filter's result. The Status bits are set to 0 if VLAN Hash Filter is also not enabled.

If the value of ITS/OTS is shown as 1/0; then it indicates that the final result is dependent on the enabled relevant filter's result.

Example 1: The second row of table 'OTS and ITS Bit Values with Only VLAN Hash Filter Enabled' indicates that at least one Perfect Filter is enabled for Outer VLAN tag comparison and none of the filters are enabled for Inner VLAN tag comparison. Inverse VLAN Filtering is not enabled. The bit OTS is given as 1/0. If the received frame passes at least one of the enabled Outer VLAN Tag filters then the bit is set to 1. If the frame doesn't pass any of the enabled Outer VLAN Tag filters, then the bit is set to 0.

Example 2: Last Row of table 'OTS and ITS Bit Values with Only VLAN Hash Filter Enabled' indicates that Inverse Filtering is enabled, Hash Filter and at least one perfect filter is enabled for Inner VLAN Tag comparison, then if the received frame's Inner VLAN tag mismatches with both the Hash Filter and all the enabled Perfect filters, then the frame will have the ITS bit set to 1 else it is set to 0. OTS will be set to 0 as no comparison is performed.

11.7.2.4.4.4 Stripping

Each of the VLAN Tags has individual control over stripping. The programming options of always strip, never strip, strip on pass and strip on fail are available. Inner or Outer VLAN Tag Stripping is based on the pass or fail results of the individual tag. If a tag is bypassed by all the relevant filters, stripping is not applicable for the tag.

- If strip on Pass is enabled for the outer VLAN Tag, then the stripping occurs only if the Outer VLAN tag has passed the relevant filters. The Outer VLAN Tag Filter Result bit will be set.
- If strip on Fail is enabled for the outer VLAN Tag, then stripping occurs only if the Outer VLAN Tag has failed relevant filters. The Outer VLAN Tag Filter Result Bit will be reset.

- If the Outer VLAN tag of the received frame is bypassed by the entire filter (no comparison has been made), then the tag is not stripped, though the Status Bit is still 0.
- As multiple filters are enabled, it is possible that the received VLAN frame could have matched any one or more of the filters. The VLAN Tag's value is not always deterministic from the filter status bits.

If the software/application strips the VLAN Tag based on the filter result, it might lose the VID. So the suggested use is, if stripping is enabled for any of the tags, the tag can be put in the status. For this the software/application will have to enable the respective "VLAN Tag in Status" bit - 24 or 31 in the MAC VLAN Tag Control Register.

See section Programming Guidelines for Extended VLAN Filtering and Routing on Receive

11.7.2.4.5 Layer 3 and Layer 4 Filtering

IP supports the Layer 3 based packet filtering which is IP Source or Destination Address filtering in the IPv4 or IPv6 packets, and Layer 4 based packet filtering which is Source or Destination Port number filtering in TCP or UDP. When layer 3 and layer 4 packet filtering is enabled then packets are filtered in the following manner.

- Matched Packets: The MAC forwards the packets that match all enabled fields to the application along with the status. The MAC gives the matched field status only if the IPC bit of MAC_Configuration register is set and one of the following conditions is true:
  - All enabled Layer 3 and Layer 4 fields match
  - At least one of the enabled field matches and other fields are bypassed or disabled

When multiple Layer 3 and Layer 4 filters are enabled, any filter match is considered as a match. If more than one filter matches, the MAC provides the status of the lowest filter where Filter 0 is the lowest filter and Filter 3 is the highest filter.

- Unmatched Packets: The MAC drops the packets that do not match any of the enabled fields. You can use the inverse match feature to block or drop a packet with specific TCP or UDP over IP fields and forward all other packets.

- Non-TCP or UDP IP Packets: By default, all non-TCP or UDP IP packets are bypassed from the Layer 3 and Layer 4 filters. MAC can optionally be programmed to drop all non-TCP or UDP over IP packets.
- IP has a control register “MAC_L3_L4_Control” to control the Layer L3 and L4 packet filtering along with address registers to program the Layer L3 and L4 fields to be matched.

11.7.2.4.6 Flexible Receive Parser

When this function is enabled, all incoming packets are parsed as per the programmable instruction stored in memory. It does the following function

- Packet Filter (Accept or Reject)
- DMA Channel Selection

Flexible receive parser block operates over the first 128 bytes data of receive packet based on 64 bytes of instruction on each packet. Following figure shows the functional block diagram of Flexible Receive Parser.

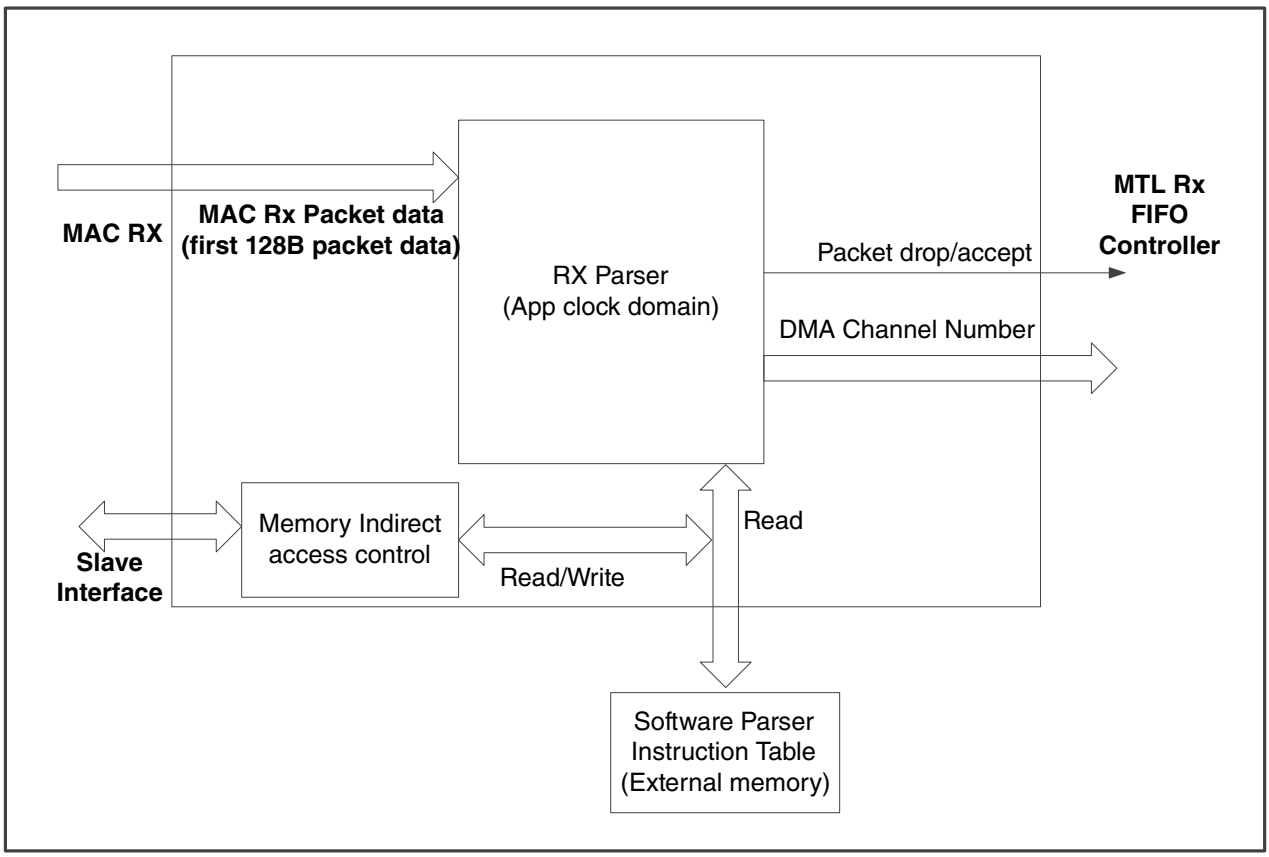


Figure 11-85. Functional block diagram of flexible recieve parser

11.7.2.4.6.1 Instruction Table Format of the Flexible Receive Parser

Software must build the 96-bit wide instruction before enabling the receive parser feature. Each instruction is programmed as an entry in the instruction memory.

The format of each entry in the Instruction table is shown in the following table.

Table 11-105. Rx Parser Instruction Entry Format

Field	Field Name	Description
31:0	MATCH_DATA	The 4-byte data. This data is used for comparing with incoming packet data starts at the frame offset as defined in [79:72] of the entry. The comparison is done only on those bits whose corresponding mask bits are set to 1 (MATCH_EN bits of [63:32]). See the Notes (below this table) on the byte

Table continues on the next page...

Table 11-105. Rx Parser Instruction Entry Format (continued)

Field	Field Name	Description
		order relative to Ethernet arrival data and endianness supported by QoS IP on the slave interface. The programming of this instruction table should adhere to the same.
62:32	MATCH_EN	When the MATCH_EN is set to 1, the corresponding Packet data bit is used for comparing. Otherwise, corresponding data bit is don't care.
64	AF	Accept frame
65	RF	Reject frame
66	IM	Inverse match
67	NIC	Next instruction control
71:68	Reserved	-
79:72	Frame Offset	[77:72] indicates the frame offset in terms of 4 bytes. (Here [79:78] always 0)Frame offset in the packet data to be compared for Match.This is in terms of 4 bytes. Value Actual Frame offset in bytes 0 01 42 8… ..63 252 The max value programmed in this Instruction Table is 128 B. The 'Actual Frame Offset' should not cross this limit.
87:80	OK Index	If (NIC==0) Memory Index to be used next when ENTRY_MATCH==1 and neither AF or RF is set to 1. If (NIC==1) Memory Index to be used next when ENTRY_MATCH==0 Note: ENTRY_MATCH is a calculated value based on MATCH_DATA, MATCH EN, and IM. It is not a field.
92:88	DMA CH NO	Indicates the DMA Channel Number (1-bit for each). This will be used when ENTRY_MATCH==1 and AF==1 (Accept frame). The encoding is as follows: • bit[0]- DMA Channel Number 0 • bit[1]- DMA Channel Number 1 • bit[2]- DMA Channel Number 2 • bit[3]- DMA Channel Number 3 • bit[4]- DMA Channel Number 4. Note: The DMA Channel number is encoded using bit wise as QoS IP support multicasting/broadcast across DMA Channels. See section 3.3.12 for more details and proper usage.
128:93	Reserved	0 (This field is only for software view and reserved for future enhancements. The memory width remains as 93-bits for now)

The parser begins parsing from the 0th entry, for each received packet. The subsequent parsing entry location (next entry or OK_INDEX[]) depends on the parser result of the current entry. Following figure shows the detailed flow.

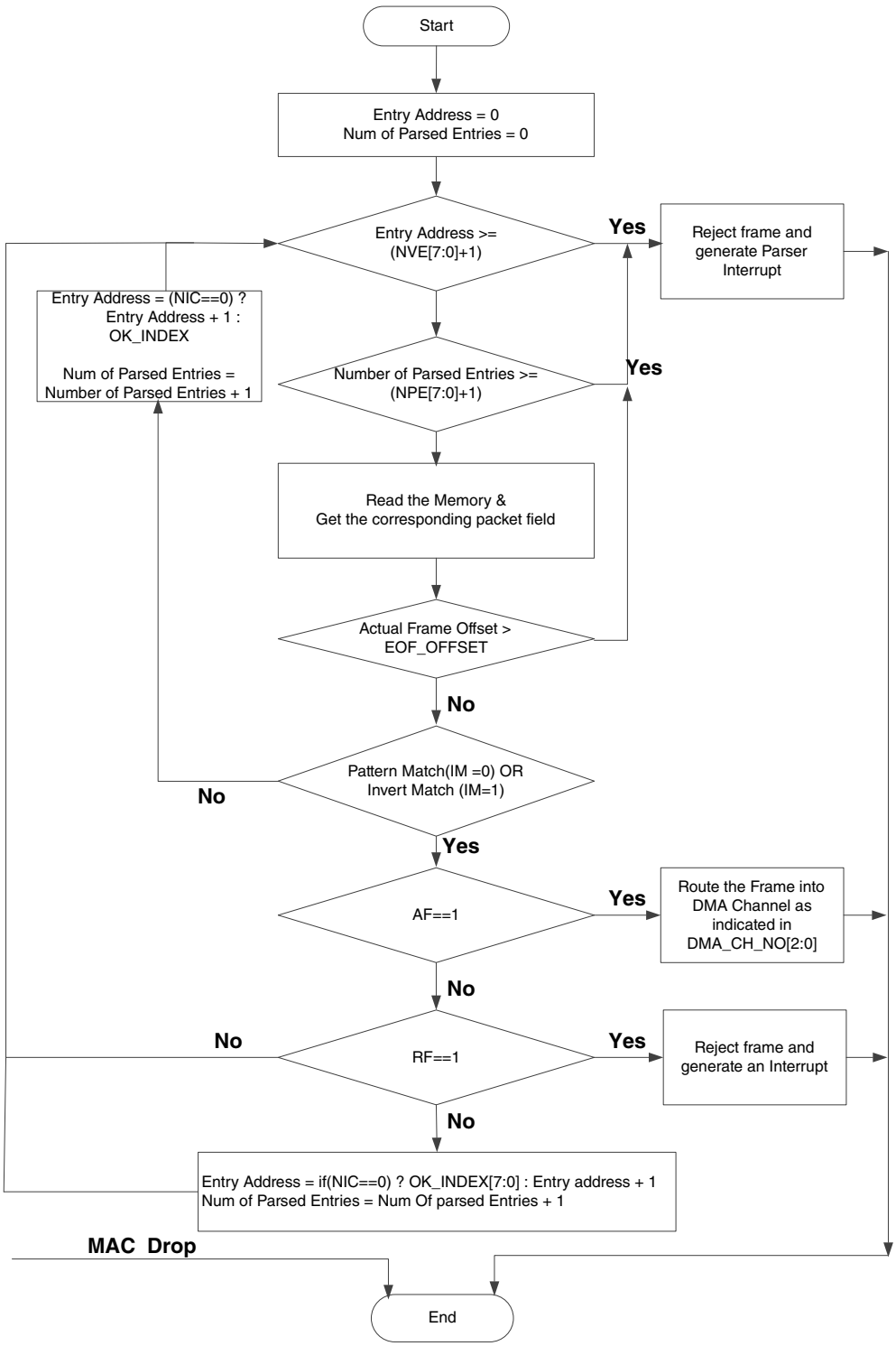


Figure 11-86. Flexible recieve parser flow chart

In the above figure, MAC Drop means the packet is dropped by the MAC due to another filter condition (for example, DA filter) and will not reach the Flexible Receive Parser.

The EOF_OFFSET shown in the figure is equal to the least among the following:

- Actual Packet size
- 128 Byte when frame pre-emption feature is not enabled/configured
- 128 Byte when frame pre-emption feature is enabled and packet is pre-emptible.
- 64 Byte when frame pre-emption feature is enabled and packet is Express Packet

11.7.2.4.6.2 Number of Valid Entries (NVE)

NVE indicates the number of valid entries in the instruction table. The default value is 64. Software can change this value by writing into MTL_RXP_Control_Status registers field NVE[7:0].

While parsing, if the Entry address is found to be is greater than NVE[7:0]+1, an error is flagged in the MTL_RXP_Interrupt_Control_Status register bit[0], NVEOVIS (Number of Valid Entries overflow status). IP generates an interrupt status (MTL_Interrupt_Status, bit[23], MTLPIS (MTL Parser Interrupt Status)).

11.7.2.4.6.3 Number of Parsable Entries (NPE)

NPE indicates the number of entries that can be parsed on an incoming packet. The default value is 64. Software can change this value by writing into MTL_RXP_Control_Status registers field NPE[7:0] (Number of Parsable Entries) bits [21:16].

While parsing, if the number of parsed entries is more than NPE[7:0]+1, an error is flagged in the MTL_RXP_Interrupt_Control_Status register bit[1], NPEOVIS (Number of Parsable Entries overflow status). IP generates an interrupt status (MTL_Interrupt_Status, bit[23], MTLPIS (MTL Parser Interrupt Status).

While NVE[] indicates is the number of valid entries in the Rx parser memory built by software, NPE[] indicates the worstcase parsable entries considering the various possible paths that the parser can take. (NVE[] >= NPE[]).

11.7.2.4.6.4 Frame Offset

It indicates the frame offset, in terms of 4 bytes which is used while comparing against the current table entry.

In an entry, an error is flagged in the TL_RXP_Interrupt_Control_Status register bit[2], FOOVIS (Frame Offset Overflow Interrupt Status) in any of the following cases:

When Frame Offset is more than

- Packet size
- 128B if pre-emption is disabled
- 128B if pre-emption is enabled and packet is pre-empitble
- 64B if pre-emption is enabled and packet is an Express packet

IP generates an interrupt status (MTL_Interrupt_Status bit[23]) and MTLPIS (MTL Parser Interrupt Status).

NOTE

If the frame ends after the (frame_offset *4) bytes but before ((frame_offset *4)+4) bytes, the Receive parser declares Frame Offset Error if comparison is enabled (via MASK bits) for the non-received bytes; from (frame_offset *4) to ((frame_offset *4)+4).

11.7.2.4.6.5 Frame Reject

In cases where an Entry matches (considering the Inverse match, IM bit in the instruction table) and Frame Accept bit is not set, and Frame Reject bit (RF) is set, the frame is dropped.

When a frame is dropped due to RF = 1, IP generates an interrupt status (MTL_Interrupt_Status, bit[23], MTLPIS (MTL Parser Interrupt Status).

11.7.2.4.6.6 Out of Order Processing

The Rx parser flow can be out of order. The decision tree need not be based on the order in which packet arrives. That is, the decision tree can be out of order; the decision tree need not be based on the same order as the packet data arrival (data within one packet). This implies that the next entry and next frame offset (OK_INDEX[]/ FRAME_OFFSET[]) can be less than the current entry address and current Frame Offset.

11.7.2.4.6.7 DMA Channel Selection

When the Rx parser is enabled (FRPE bit[16] is set to 1 in MTL_Operation_Mode register) the Rx parser overrides the DMA selection criteria of the MAC.

When the Rx parser is disabled (FRPE bit[16] is set to 0 in MTL_Operation_Mode register) the MAC/MTL determines the DMA channel number.

11.7.2.4.6.8 Receive Queue Selection

The Rx Queue is determined by the existing MAC functionality, irrespective of the enabling/disabling of the Rx Parser. This is because the Rx parser is mainly designed for selecting the DMA channel and Rx queuing is decided by the MAC to enable the proper functioning of the Pause/PFC feature.

NOTE

When there are multiple queues and multiple DMA Channels are enabled, a particular DMA channel might receive out of order packets. This is because queue selection is based on MAC criteria (VLAN Priority and other criteria) and Rx DMA channel selection is based on Rx parser.

11.7.2.4.6.9 MAC Packet Filtering/Drop/Error Handling

The MAC does the packet filtering based on the received packet fields. Disable the Packet Filtering features inside the MAC when the Rx parser is enabled. To disable the Packet Filtering features inside the MAC,

1. Set Promiscuous Mode (PR, bit[0] of MAC_Packet_Filter register)
2. Set all other bits in the MAC_Packet_Filter register to its default values

When MAC decides to drop the packet (due to Rx MAC dependent Error such as GMII Error, Receive Watch Dog Error, CRC Error and Giant Frame Error) the packet is dropped irrespective of the Rx parser decision.

11.7.2.4.6.10 PAD Strip/CRC Strip Handling

The pad strip and CRC strip are controlled by the software and applicable to all the received packets. When these features are enabled, software must build the Rx parser instructions accordingly.

NOTE

Enable PAD and CRC stripping in real use case.

11.7.2.4.6.11 VLAN Strip Handling

The MAC supports stripping the VLANs, outer as well as inner VLAN. When MAC is programmed to strip the VLAN, the Rx parser considers those VLAN tags to be part of the incoming packet.

11.7.2.4.6.12 Multicast and Broadcast Support

IP supports multicast and broadcast packet and the packet is routed to the highest queue number.

When Rx Parser is enabled, the packet routing is decided by the DMA Channel Numbers (DMA CH NO, Bits [95:88]) in the Instruction Table.

When Rx Parser is disabled, the packet routing is determined by DCS (DMA Channel Select) field of the MAC_AddressX_High register. For more details, see Broadcast/ Multicast Packet Duplication.

NOTE

DCS/DMA CH NO field is per DMA Channel control and therefore, multiple DMA channels can be selected for routing the packet.

11.7.2.4.6.13 Pre-emption Support

The MAC supports the frame pre-emption feature when the Flexible Rx parser feature is enabled. The minimum packet size for Frame Preemption is 64 Bytes. This means that:

- The Rx parser can operate on 64 bytes of Express traffic.
- The Rx parser can operate on 128 bytes of pre-emptible traffic, as a 64-bit packet of pre-emptible traffic might be interrupted by Express traffic before that packet is fully received, so the parser needs to have 64 bits on either side of the express traffic in order to capture the full 64-bits of a pre-empted packet.

11.7.2.4.6.14 Software Access to the Flexible Receive Parser Memory

Software can read/write into the Rx parser memory, using the following registers via indirect addressing.

- MTL_RXP_Indirect_Acc_Control_Status
- MTL_RXP_Indirect_Acc_data

For more details, see ENET_QOS register descriptions.

11.7.2.4.6.15 Statistical Counters

IP supports the following statistics counters for flexible receive parser.

1. MTL_RXP_Drop_Cnt
2. DMA_RXP_Error_Cnt

DMA_CH(#i)_Rxp_Accept_cnt (i=0; i<2)

For more details, see ENET_QOS register descriptions.

11.7.2.4.6.16 Changing the Instruction Table by Software

Software can update the instruction table in the memory in the following ways

- Disable the MAC Receiver and Receive Parser by setting the RE field of the MAC_Configuration register to 0 and FRPE bit of MTL_Operation_mode register to 0. Wait for Rx Parser to become inactive (RXPIA, MTL_RXP_Control_Status register bit[31]).
- (Optional) Program in the Entry address 0 to unconditionally (all MATCH_EN bit 0) skip to OK_INDEX[] (to certain location in the memory) where it is programmed to reject or accept all packets.

11.7.2.4.6.17 Receive Cut-Through Functionality

In the case of cut-through mode, the Rx controller attempts to transfer the received packets to DMA just after the cut-through threshold amount of packet data is received (RTC field in the MTL_RxQx_Operation_Mode register).

However due to Rx parser's result worst case arrival time can be more than the programmed cut-through threshold. So, the Rx controller delays the packet transfer to DMA accordingly. So, the cut-through functionality is re-defined as follows

The Rx controller attempts to transfer packet to DMA after at least RTC number of bytes have been received and Rx parser results are available.

11.7.2.4.6.18 Receive Packet Drop Indication

In most cases, the packet drops internally in the Rx Queue. However, back-to-back packets can arrive for the same queue and first packet's Rx parser result is not available when the second packet arrives. In such cases, the packet cannot be flushed internally and forwarded to DMA or the Application Interface and status is indicated accordingly. For more details, see RDES2, bit 16(RXPD) in Receive Normal Descriptor (Write-Back Format).

11.7.2.4.6.19 Runt Packet Handling

The Rx parser can perform the operation with 64B minimum size packet. If the runt packet is received (should be indicated by MAC), the Rx parser assumes that it gets dropped in the MAC. This is because, back-to-back runt packets (< 64B) cannot be handled in the Rx parser.

11.7.2.4.6.20 Uncorrectable ECC Error Handling

When the parser detects an uncorrectable ECC error while parsing the parser memory, the packet is not dropped. Instead, the packet is sent to the application with an Error Indication. The RXPI (RX Parser Incomplete) is set in the packet status along with Error Summary (ES) bit.

For more details, see Receive Normal Descriptor (Write-Back Format).

11.7.2.5 IEEE 1588 Timestamp Support

The MAC supports the IEEE 1588 Precision Time Protocol (PTP). The MAC is able to do precise time stamping and capturing of incoming and outgoing frame to enable support of PTP with its associated software. The MAC supports the all clock types defined in IEEE 1588-2008. Typical frequency of PTP timestamp clock is 125 MHz. The MAC supports:

- Taking a timestamp snapshot of all packets or only PTP type packets.
- Taking a timestamp snapshot of only event messages.
- Identifying the PTP message type, version, and PTP payload in packets sent directly over Ethernet and sending the PTP message type status given by the MAC. The PTP status is finally written back to the Receive Descriptor: RDES1 Normal Descriptor (Write-Back Format).
- Measuring sub-second time in decimal or binary format.

11.7.2.5.1 Delay Request-Response Mechanism

The system or network is classified into the master and slave nodes for distributing the timing and clock information. Following figure shows the process that PTP uses for synchronizing a slave node to a master node by exchanging PTP messages.

Figure 11-87. Networked Time Synchronization

As shown in the figure, PTP uses the following process:

1. The master broadcasts the PTP Sync messages to all its nodes. The Sync message contains the reference time information of the master. This message leaves the system of the master at t1. This time must be captured for Ethernet ports at the internal GMII interface.
2. The slave receives the Sync message and also captures the exact time, t2, using its timing reference.
3. The master sends a Follow_up message to the slave, which contains t1 information for later use.
4. The slave sends a Delay_Req message to the master and notes the exact time, t3, at which this packet leaves the internal GMII interface.
5. The master receives the message, capturing the exact time t4, at which the message enters its system.
6. The master sends the t4 information to the slave in the Delay_Resp message.
7. The slave uses the four values of t1, t2, t3, and t4 to synchronize its local timing reference to the timing reference of the master.

Most of the PTP implementation is done in the software above the Ethernet layer. However, the hardware support is required to capture the exact time when specific PTP packets enter or leave the Ethernet port at the internal GMII interface. This timing information must be captured and returned to the software for proper implementation of PTP with high accuracy.

11.7.2.5.2 Peer-to-Peer PTP Transparent Clock (P2P TC) Message Support

The IEEE 1588-2008 supports peer-to-peer PTP (Pdelay) message in addition to the SYNC, Delay Request, Follow-up, and Delay Response messages. Following figure shows the method to calculate the propagation delay in clocks supporting peer-to-peer path correction.

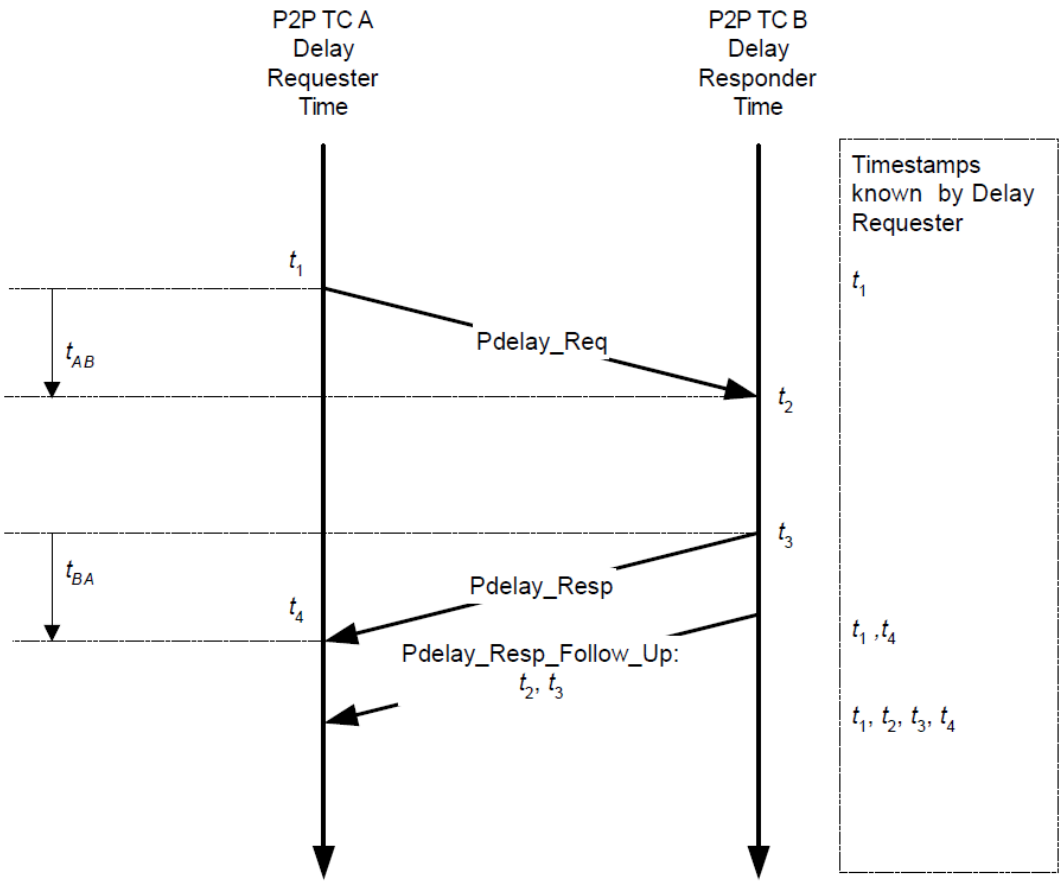


Figure 11-88. Propagation Delay Calculation in Clocks Supporting Peer-to-Peer Path Correction

As shown in the above figure, the propagation delay is calculated in the following way:

1. Port 1 issues a Pdelay_Req message and generates a timestamp (t1) for the Pdelay_Req message.
2. Port 2 receives the Pdelay_Req message and generates a timestamp (t2) for this message.
3. Port 2 returns a Pdelay_Resp message and generates a timestamp (t3) for this message. To minimize errors because of any frequency offset between the two ports, Port 2 returns the Pdelay_Resp message as quickly as possible after the receipt of the Pdelay_Req message. Port 2 returns any one of the following:

- Difference between the timestamps t2 and t3 in the Pdelay_Resp message
- Difference between the timestamps t2 and t3 in the Pdelay_Resp_Follow_Up message
- Timestamps t2 and t3 in the Pdelay_Resp and Pdelay_Resp_Follow_Up messages, respectively

4. Port 1 generates a timestamp (t4) on receiving the Pdelay_Resp message.
5. Port 1 uses all four timestamps to compute the mean link delay.

11.7.2.5.3 Timestamp Correction

According to the IEEE 1588 specification, a timestamp must be captured when the PTP message timestamp point (leading edge of the first bit of the octet immediately following the Start Frame Delimiter octet) crosses the boundary between the node and the network. As the MAC takes the timestamp at an internal point far from the actual boundary of the node and network, this captured timestamp is corrected/updated for the ingress/egress path latency (including the delay in the PHY layers). Further correction is done for the inaccuracies/errors introduced due to the clock (internal GMII Tx, Rx clock) being different at the capture point as compared to the PTP clock (clk_ptp_ref_i) that is used to generate the time. The resultant CDC (Clock Domain Crossing) circuits add error depending on the clock period of the internal GMII or MII and PTP clocks.

11.7.2.5.3.1 Ingress Correction

In the Receive side the timestamp captured at the internal snapshot point is delayed (later in time) as compared to the time at which that packet's SFD bit is received at the port's boundary. Therefore, the captured timestamp must be reduced by the ingress latency and the errors in CDC sampling. This correction value must be determined/calculated by the software and written into the MAC_Timestamp_Ingress_Corr_* registers.

The correction value consists of the following 3 components:

1. External latency in the PHY layer between boundary point and the input of the core.

   If the PHY is compliant to the IEEE 802.3 Clause 45 MMD registers, it has a register indicating the maximum and minimum ingress latency. The software can read these registers and determine the average ingress latency in the PHY. Alternatively (if the PHY does not support these registers), the ingress latency must be determined from its datasheet or timing characteristics.

2. Internal latency from the input of the core to the internal capture point.

The internal ingress latency can be read from the MAC_Ingress_Timestamp_Latency register. This is a read-only register and gives the latency in scaled Nanoseconds format defined in IEEE 1588 Clause 5.3.2. The latency differs based on the active PHY interface (for example, RMII) and the operating speed. Therefore, the software must read this register after any speed change in the MAC to determine the current internal latency.

3. CDC Synchronization

   The CDC synchronization error is almost equal to 2 times the clock-period of the PTP clock (clk_ptp_ref_i).

The values determined from these 3 components should be added by the software and must be written into the TSIC and TSICSNS fields of the MAC_Timestamp_Ingress_Corr_* registers.

NOTE

The value written to the register must be negative (two's complement as explained below), because it has to be subtracted from the captured timestamp. The MAC Receiver adds the value in this register to the captured timestamp and then gives the resultant value as the timestamp of the received packet.

When TSCTRLSSR bit in MAC_Timestamp_Control register is set, the nanoseconds field of the captured timestamp is in decimal format with a granularity of 1ns. So the Bit31 of TSIC must be set to 1 (for negative value) and bits[30:0] must be written with "10^9 - total ingress_correction_value[nanosecond part]" represented in binary. For example, if the required correction value is -5 ns, then the value is 0xBB9A_C9FB.

When TSCTRLSSR bit in MAC_Timestamp_Control register is reset, the nanoseconds field of the captured timestamp is in binary format with a granularity of ~0.466ns. Therefore, bits[30:0] must be written with "2^31 - total ingress_correction_value" represented in binary with bit[31] = 1.

11.7.2.5.3.2 Egress Correction

In the Transmit side the timestamp captured at the internal snapshot point is earlier (advanced in time) as compared to the time at which that packet's SFD bit is output at the port's boundary. Therefore, the captured timestamp must be compensated by the egress latency and the errors in CDC sampling. This correction value must be determined/ calculated by the software and written into the MAC_Timestamp_Egress_Corr_* registers.

The correction value consists of the following 3 components:

1. External latency in the PHY layer between the output of the core and the boundary of the port and the network

   If the PHY is compliant to the IEEE 802.3 Clause 45 MMD registers, it has a register indicating the maximum and minimum egress latency. The software can read these registers and determine the average egress latency in the PHY. Alternatively (if the PHY does not support these registers), the egress latency must be determined from its datasheet or timing characteristics.

2. Internal latency from the internal capture point and the output of the core

   This internal egress latency can be read from the MAC_Egress_Timestamp_Latency register. This is a read-only register and gives the latency in scaledNanoseconds format defined in IEEE 1588 Clause 5.3.2. The latency will differ based on the active PHY interface (for example, RMII) and the operating speed. Hence the software must read this register after any speed change in the MAC to determine the current internal latency.

3. CDC synchronization error

   The CDC synchronization error value for one-step timestamping would be = (1 * period of clk_ptp_ref_i + 4 * period of clk_tx_i). Otherwise (Two-step timestamping mode), the value = -(2 * period of clk_ptp_ref_i).

11.7.2.5.3.3 Frequency Range of Reference Timing Clock

The timestamp information is transferred across asynchronous clock domains, from the MAC clock domain to the application/system clock domain. Therefore, a minimum delay is required between two consecutive timestamp captures. This delay is 4 clock cycles of RMII and 3 clock cycles of PTP clocks. If the delay between two timestamp captures is less than this delay, the MAC does not take a timestamp snapshot for the second packet.

11.7.2.5.3.4 PTP Processing and Control

Following table shows the common message header for the PTP messages. This format is taken from the IEEE 1588-2008.

Table 11-106. Message Format Defined in IEEE 1588-2008

Bits								Octet	Offset
7	6	5	4	3	2	1	0		
transportSpecific				messageType				1	0
Reserved				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
Reserved								1	5
flagField								2	6
correctionField								8	8
Reserved								4	16
sourcePortIdentity								10	20
sequenceId								2	30
controlField (*)								1	32
logMessageInterval								1	33

(*) – control Field is used in version 1. In version 2, message Type field is used for detecting different message types.

There are some fields in the Ethernet payload that are being use to detect the PTP packet type and control the snapshot to be taken. These fields are given below in table for PTP packets over Ethernet.

Following table provides information about the fields that are matched to control the snapshots for the PTP packets sent over Ethernet for IEEE 1588 version 1 and 2. The octet positions for the tagged packets are offset by 4. This is based on the IEEE 1588-2008, Annex D and the message format.

Table 11-107. Ethernet PTP Packet Fields Required for Control And Status

Field Matched	Octet Position	Matched Value	Description
MAC Destination Multicast Address[1]	0–5	01-1B-19-00-00-00 01-80-C2-00-00-0E	All PTP messages can use any of the following multicast addresses[2]: • 01-1B-19-00-00-00 • 01-80-C2-00-00-0E[3]
MAC Packet Type	12, 13	0x88F7	PTP Ethernet packet
PTP Control Field (IEEE 1588 version 1)	46	0x00, 0x01, 0x02, 0x03, or 0x04	• 0x00: SYNC • 0x01: Delay_Req • 0x02: Follow_Up • 0x03: Delay_Resp • 0x04: Management

Table continues on the next page...

Table 11-107. Ethernet PTP Packet Fields Required for Control And Status (continued)

Field Matched	Octet Position	Matched Value	Description
PTP Message Type Field (IEEE 1588 version 2)	14 (nibble)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, or 0xD	• 0x0: SYNC • 0x1: Delay_Req • 0x2: Pdelay_Req • 0x3: Pdelay_Resp • 0x8: Follow_Up • 0x9: Delay_Resp • 0xA: Pdelay_Resp_Follow_Up • 0xB: Announce • 0xC: Signaling • 0xD: Management
PTP Version	15 (nibble)	0x1 or 0x2	• 0x1: Supports PTP version 1 • 0x2: Supports PTP version 2

1. The unicast address match of destination addresses (DA), programmed in MAC address 0 to 31, is used if the TSENMACADDR bit of MAC_Timestamp_Control register is set.
2. IEEE 1588-2008, Annex F
3. The MAC does not consider the PTP version 1 messages with Peer delay multicast address (01-80-C2-00-00-0E) as valid PTP messages.

11.7.2.5.4 Transmit Path Functions

The MAC captures a timestamp when the Start Packet Delimiter (SFD) of a packet is sent on the internal GMII interface. The packets, for which the timestamps has to be captured can be controlled on per-packet basis. Each Transmit packet can be marked to indicate whether a timestamp should be captured for it. The MAC does not process the transmitted packets to identify the PTP packets. You can specify the packets by using the control bits in the Transmit descriptor. The MAC returns the timestamp to the software inside the corresponding Transmit descriptor, thus connecting the time-stamp automatically to the specific PTP packet. The 64-bit timestamp information is written to the TDES0 and TDES1 fields. The TDES0 field holds the 32 least significant bits of the timestamp.

11.7.2.5.5 Receive Path Functions

The MAC can be programmed to capture the timestamp of all packets received on the internal GMII interface or to process packets to identify the valid PTP messages. Use the following options of the MAC_Timestamp_Control register to control the snapshot of the time to be sent to the application:

- Enable snapshot for all packets
- Enable snapshot for IEEE 1588 version 1 or version 2 timestamp
- Enable snapshot for PTP packets transmitted directly over Ethernet
- Enable timestamp snapshot for the received packet for IPv4 or IPv6
- Enable timestamp snapshot only for EVENT messages (SYNC, DELAY_REQ, PDELAY_REQ, or PDELAY_RESP)
- Enable the node to be a master or slave and select the snapshot type

Table 11-108. Timestamp Snapshot Dependency on Register Bits

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	PTP Messages
00	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp
00	0	1	SYNC
00	1	1	Delay_Req
01	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp,Pdelay_Req, Pdelay_Resp,Pdelay_Resp_Follow_Up
01	0	1	SYNC, Pdelay_Req, Pdelay_Resp
01	1	1	Delay_Req, Pdelay_Req, Pdelay_Resp
10	x	x	SYNC, Delay_Req
11	x	x	Pdelay_Req, Pdelay_Resp

The DMA returns the timestamp to the software inside the corresponding Receive Descriptor. The extended status, containing the timestamp message status and the IPC status, is written in normal descriptor RDES1 and the snapshot of the timestamp is written in RDES0 and RDES1 fields of context descriptor. The RDES0 field holds the 32 least significant bits of the timestamp.

NOTE

For Programming Guidelines for IEEE 1588 Timestamping (System Time Correction), see System Time Correction

11.7.2.5.6 IEEE 1588 System Time Source

The ENET QOS block supports both external and internal time source for reference timestamping.

- External Time source uses the 64-bit external time reference and clock as input in IP. The clock input is used to synchronize the external timing reference into MAC clock domain. Upper 32-bit indicates the time in seconds while lower 32-bit indicates the time in nanoseconds.
- Internal time source uses the only clock input to generate timing reference internally for snapshot and capture timestamps. Internal time reference has two field.
  - UInteger48 seconds Field: The seconds field is the integer portion of the timestamp in units of seconds. It is 48-bits wide.

    For example, 2.000000001 seconds are represented as seconds Field = 0x0000_0000_0002.

  - UInteger32 nano secondsField The nanoseconds field is the fractional portion of the timestamp in units of nanoseconds.

    For example, 2.000000001 seconds are represented as nanoSeconds = 0x0000_0001. The nanoseconds field supports the following two modes:

    - Digital rollover mode: In this mode, the maximum value in the nanoseconds field is 0x3B9A_C9FF, that is, (10e9-1) nanoseconds.
    - Binary rollover mode: In this mode, the nanoseconds field rolls over and increments the seconds field after value 0x7FFF_FFFF. Accuracy is ~0.466 ns per bit.

There is a system time register module which is being used when internal time reference is being used. The 80-bit time is maintained in this module and updated using the input reference clock (clk_ptp_ref_i). This time is the source for taking snapshots (timestamps) of Ethernet packets being transmitted or received at the internal GMII interface.

The system time counter can be initialized or corrected using the coarse correction method. In this method, the initial value or the offset value is written to the Timestamp Update register. For initialization, the system time counter is written with the value in the Timestamp Update register. For system time correction, the offset value is added to or subtracted from the system time.

In the fine correction method, the frequency offset and/or frequency drift of a slave clock (clk_ptp_ref_i) with respect to the master clock (as defined in IEEE 1588-2002) is corrected over a period of time instead of in one clock, as in coarse correction. This helps maintain linear time and does not introduce drastic changes (or a large jitter) in the reference time between PTP Sync message intervals.

NOTE

For Programming Guidelines for IEEE 1588 Timestamping (For Internal Time-stamp Source Configuration). see Initialization Guidelines for System Time Generation

11.7.2.5.7 IEEE 1588 Higher Word Register

If system time source is internal, then higher word register can be invoked. MAC timestamp is 64-bit wide. The values of the upper 16-bits of the seconds field can be read from the CSR register.

11.7.2.5.8 Flexible Pulse-Per-Second Output

The ENET QOS block supports the flexibility of programming the start or stop time, width and interval of pulse generated on pulse per second (PPS) output. IP supports 4 such output signals. By default, the block is in fixed pulse per second output mode with an interval of 1 second.

Initially. the start time should be programmed in target time registers "MAC_PPSx_Target_Time_Seconds and MAC_PPSx_Target_Time_Nanoseconds" for all desired/enabled PPS output. Reprogramming the start or stop time must be done after the synchronization of the earlier programmed value. Bit 31 of the MAC_PPS#_Target_Time_Nanoseconds register indicates that the synchronization is complete. If the application programs a start or stop time that has already elapsed, the MAC sets an error status bit indicating the programming error. If enabled, the MAC also sets the Target Time Reached interrupt event. The application can cancel the start or stop request only if the corresponding start or stop time has not elapsed. If the time has elapsed, the cancel command has no effect.

The PPS width and interval are programmed in terms of granularity of system time, that is, number of the units of sub-second increment value.

NOTE

See Programming Guidelines for Flexible Pulse-Per-Second Output

11.7.2.5.9 Media Clock Generation and Recovery

Media clock recovery and generation needs a dedicated hardware and software to complete the operation. IP generates a reference clock with the help of dedicated hardware which is then multiplied to get the desired media clock. Software extracts the presentation time from the incoming PTP 1722 frames and programs these value into CSR registers so IP can read them for media clock recovery.

As shown in the following figure, the timestamp array and the DMA module are external to IP. IP supports a 32-bit presentation time counter in nanoseconds that wraps at the full32 bit value to match up with 1722 presentation time format and an handshake mechanism with external application(such as DMA) to program the presentation time into CSR register.

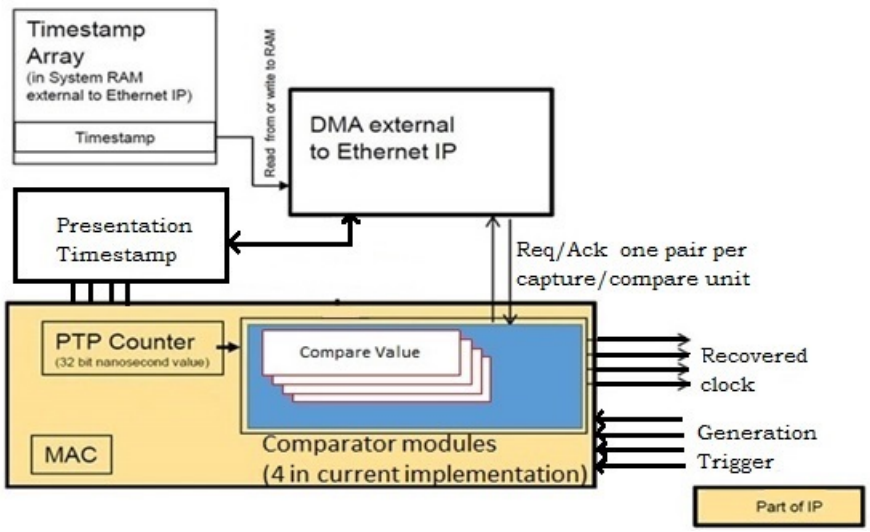


Figure 11-89. Block Diagram of Media Clock Generation and Recovery

11.7.2.5.9.1 Presentation Time Counter

Presentation time counter is another view of the PTP System time (1588 timer). For a given PTP System time (PTP[63:32] represents time in seconds and PTP[31:0] represents time in nanoseconds), there exists a corresponding Presentation Time (32-bit value in nanosecond). The Presentation time computed is referred to as Current Presentation Time (CPT).

The Current Presentation Time (CPT) can be viewed as derived from 64bit PTP System Time. PTPNS[63:0] = (PTP[63:32] * 32'd1,000,000,000) + PTP[31:0] where, PTPNS is the PTP system time converted into a 64bit nanosecond value.

Current Presentation Time[31:0] = PTPNS[31:0]

Media Clock Recovery is possible when the System Time is internally generated in IP and CPT is computed in the same way as the Internal PTP System Time generation. Because CPT and System Time values are in nanoseconds, and must be synchronous, the increment cycle of both the timers are same. The timer is updated at the same instance, but the updated value could be different. The software must compute a separate 32-bit

update value in nanoseconds for the CPT update. CPT sampled at different edges of a triggering input results in the generation of Media Clock timestamps which are inserted in 1722 based AVBTP packets, to recover the clock at the destination.

11.7.2.5.9.2 Comparator Modules

As shown in the figure 'Block Diagram of Media Clock Generation and Recovery' the module supports four comparator modules to handle multiple media clocks that might be required. The Comparator modules hand shake with the application to

Program the timestamps into the MAC_PPS(#i)_Target_Time_Seconds register when PTGE field of MAC_Timestamp_Control register is set.

Set the Presentation Time Control bits of the particular instance for recovery mode.

The timestamps received from the application are referred as Target Presentation Time (TPT).

When MCGREN#i field of MAC_PPS_Control register is set to 1, the IP operates in MCGR mode. The comparator module sends up to two requests to the application for TPT write, when PTGE field of MAC_Timestamp_Control register is set and the Presentation time control bits of the particular instance is set for MCGR mode. Subsequent requests are generated each time a Presentation Time match occurs. (CPT transitions from a value less than TPT, to a value greater than or equal to CPT). The first request is asserted when a specific comparator instance is set to MCGR mode with a non-zero Presentation Time Control and an additional request is made when the first data is received into the Comparator. This allows the application to write the next TPT value while IP processes the previous TPT value for a match.

The TPT read from the MAC_PPS(#i)_Target_Time_Second is considered as a future time. A toggle/pulse is generated in the next cycle when a match is detected. When a match is detected, the mcgr_dma_req_o#i request for that comparator is asserted (to obtain next TPT) until the corresponding application acknowledgment is set.

The presentation control bits (PPSCMD#i field of MAC_PPS_Control register) determine the shape of the generated waveform. Also, these bits can be programmed to either toggle or generate a high/low pulse for one PTP clock cycle, upon match.

Media Clock Recovery: A match is when the free-running CPT value matches the received TPT value. A output signal EMAC_TMR signal as shown in the IO mux sheet is asserted (toggle, low pulse, or high pulse) based on the presentation control value programmed.

Media Clock Generation: Based on the presentation control value programmed in MAC_PPS_Control register, the particular comparator captures the presentation time and programs it into MAC_PPS(#i)_Target_Time_Second register. A request is raised to the application to read the captured timestamp. Until the acknowledgment is received from the application, acknowledging that the read operation is complete, no new timestamps are captured.

11.7.2.5.9.3 Media Clock Generation and Recovery Flow

Following figure shows the media clock generation and recovery flow. For both media clock generation and recovery, set the PTGE field of MAC_Timestamp register for CPT generation and set the MCGREN#i field of MAC_PPS_Control Register for the corresponding instance to be enabled in the MCGR mode.

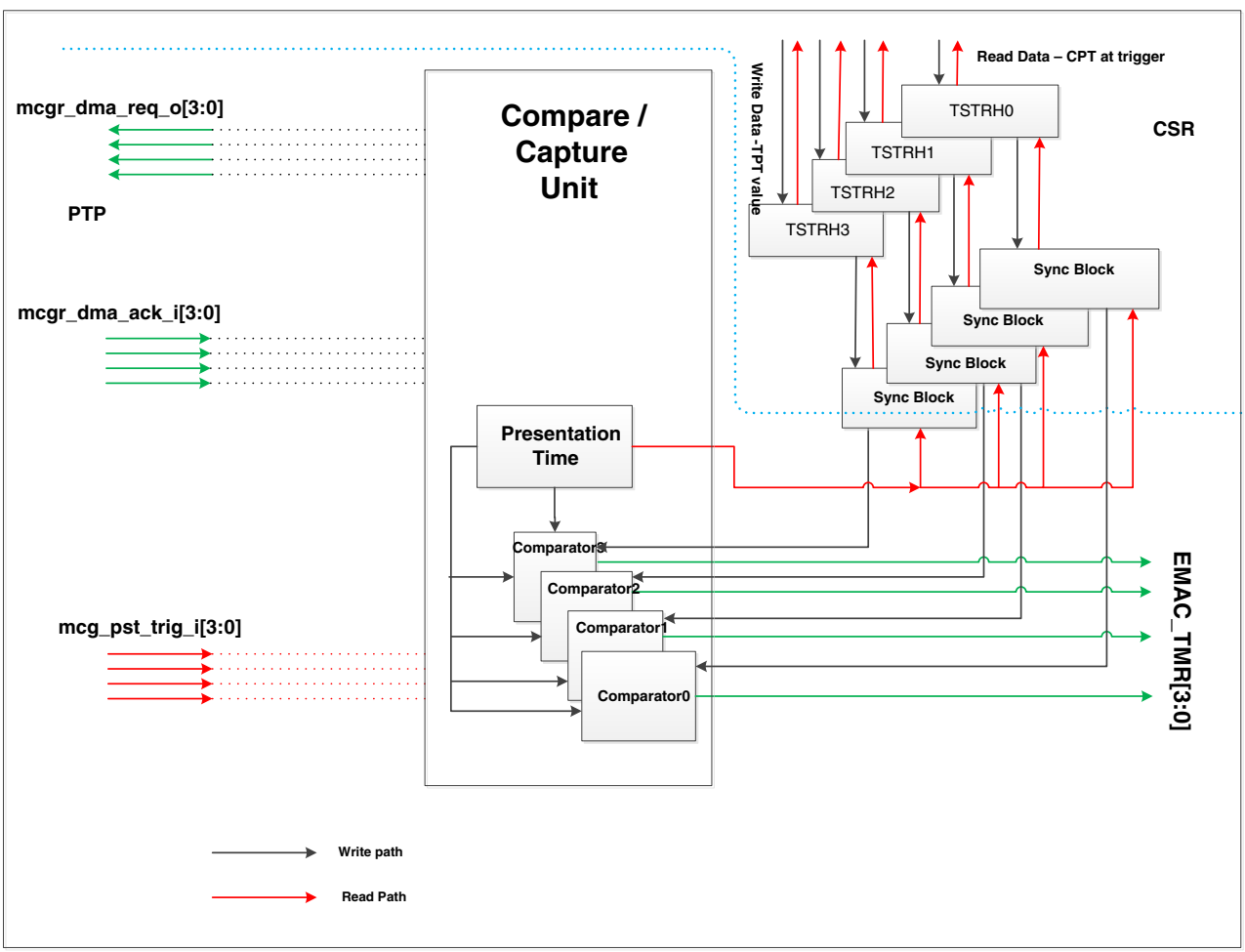


Figure 11-90. Media Clock Generation and Recovery Flow

NOTE

1. The consecutive triggers to sample the presentation time should always be asserted after a few PTP clock cycles to allow for synchronization delays. (This is not an issue when the input trigger maximum frequency is 8KHz and the PTP clock is running at least at 1MHz)
2. DMA acknowledgment can be both posted as well as non-posted, in the Media Clock Recovery Mode.
3. See "Programming Guidelines for Media Clock Generation and Recovery"

11.7.2.5.10 One step timestamp

IP supports the one step time stamp feature. It can be enabled for a packet by setting the Bit 20 (OSTC) in the control word.

NOTE

See Programming Guidelines for IEEE 1588 Timestamping.

11.7.2.5.11 IEEE 1588 Sub Nanoseconds Timestamp

The ENET QOS block supports the ingress and egress correction accuracy in sub-nanoseconds. It is programmed in the TSICSNS and TSECSNS fields of the MAC Timestamp Ingress Correction Subnanosecond and MAC Timestamp Egress Correction Subnanosecond registers, respectively. In this case, the correction has an unit of nanosecond, multiplied by 2^8. The least significant 8 bits of the value should be programmed in the sub-nanoseconds register.

For example, if the required egress correction is 3.6 nS and TSCTRLSSR bit is set, then TSEC must be 0x3 and TSECSNS must be 0x99 (0.6 * 256). Similarly, if the required ingress correction is -3.6 ns and TSCTRLSSR bit is set, then TSIC must be 0xBB9A_C9FC and TSICSNS must be 0x66 (fractional part of (10^9 - 3.6) * 256).

11.7.2.6 Multiple Channels and Queues Support

IP supports the five queues and channel on Tx and Rx paths.

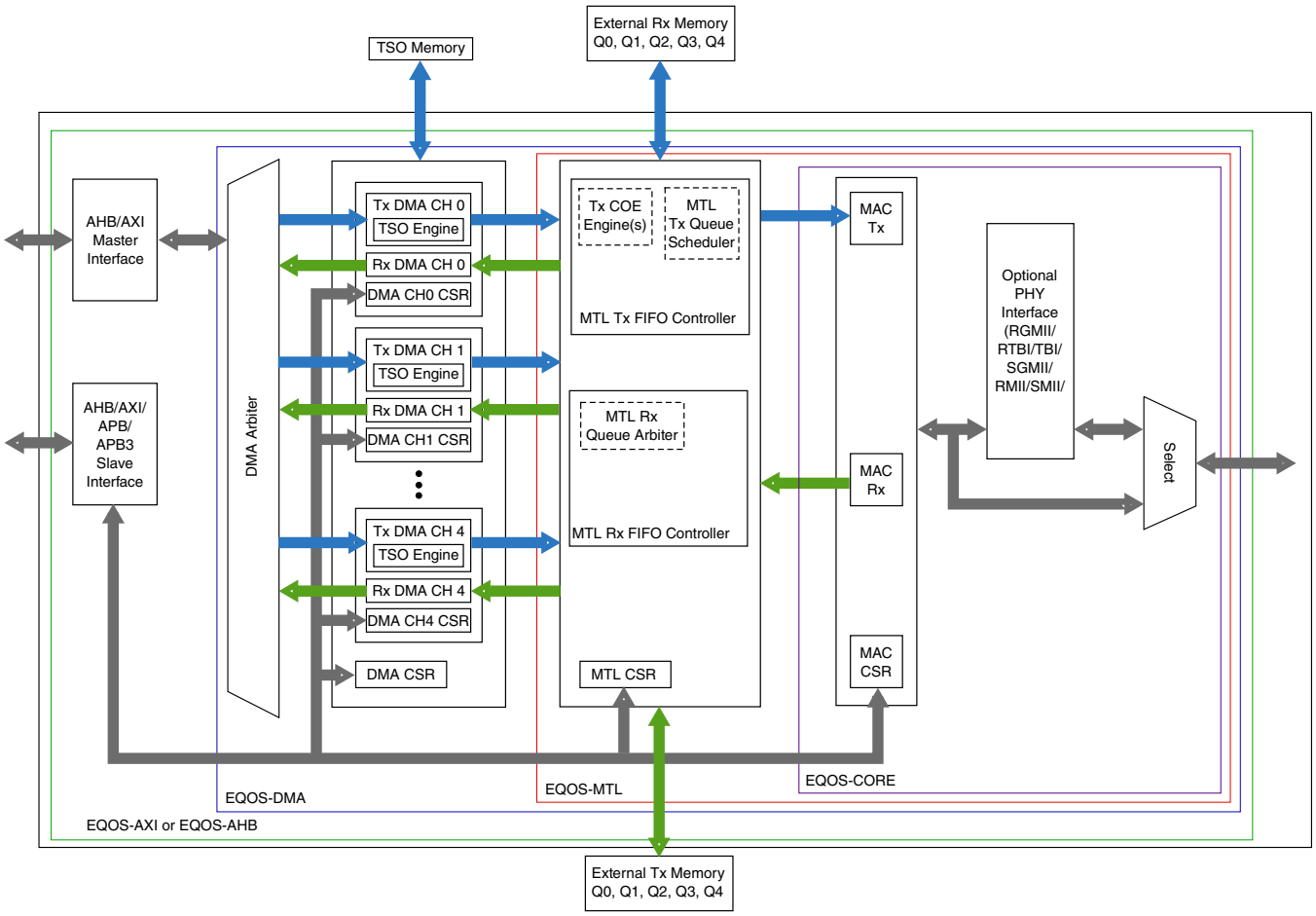


Figure 11-91. Module with Multiple Channels and Queues Block Diagram

DMA arbiter of IP supports two types of arbitration scheme, fixed priority and weighted round-robin. The DMA arbiter performs the arbitration between the Tx and Rx paths of DMA channels for accessing descriptors and data buffers. The DA bit of the DMA_Mode register specifies the arbitration scheme (fixed or weighted round-robin) between the Tx and Rx DMA of a channel.

Priority between the corresponding Tx DMA and Rx DMA can be set by using TXPR field of the DMA_Mode register. PR field of the DMA_Mode register can be set to specify the weighted priority between the Tx DMA and Rx DMA in round robin arbitration scheme.

Following table provides information about the priority scheme between Tx DMA and Rx DMA.

Table 11-109. Priority Scheme for Tx DMA and Rx DMA

Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Priority Schemes
x	x	x	0	1	Rx always has priority over Tx

Table continues on the next page...

Table 11-109. Priority Scheme for Tx DMA and Rx DMA (continued)

Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Priority Schemes
0	0	0	0	0	Tx and Rx have equal priority. Rx gets the access first on simultaneous requests
0	0	1	0	0	Rx has priority over Tx in ratio 2:1
0	1	0	0	0	Rx has priority over Tx in ratio 3:1
0	1	1	0	0	Rx has priority over Tx in ratio 4:1
1	0	0	0	0	Rx has priority over Tx in ratio 5:1
1	0	1	0	0	Rx has priority over Tx in ratio 6:1
1	1	0	0	0	Rx has priority over Tx in ratio 7:1
1	1	1	0	0	Rx has priority over Tx in ratio 8:1
x	x	x	1	1	Tx always has priority over Rx
0	0	0	1	0	Tx and Rx have equal priority. Tx gets the access first on simultaneous requests
0	0	1	1	0	Tx has priority over Rx in ratio 2:1
0	1	0	1	0	Tx has priority over Rx in ratio 3:1
0	1	1	1	0	Tx has priority over Rx in ratio 4:1
1	0	0	1	0	Tx has priority over Rx in ratio 5:1
1	0	1	1	0	Tx has priority over Rx in ratio 6:1
1	1	0	1	0	Tx has priority over Rx in ratio 7:1
1	1	1	1	0	Tx has priority over Rx in ratio 8:1

11.7.2.6.1 Multiple Queues and Channels Support in the Transmit Path

IP supports the five transmit queues and five channels.

NOTE

See Programming Guidelines for Multi-Channel Multi-Queuing

11.7.2.6.2 Multiple Queues and Channels Support in the Receive Path

IP supports five receive channels and five queues.

In the Receive direction, the MTL Rx Controller selects the Rx DMA for which it is transferring or reading the data from the Rx FIFO memory. This scheduling is based on the programming done in the respective MTL_RxQ[x]_Control register.

Each Rx DMA indicates when it is ready to transfer data and the size of the burst-length (number of beats) that it has to transfer. The scheduler checks whether sufficient data (of requested burst length) is available to be transferred to these DMAs and then selects the Rx DMA that gets serviced using the programmed priorities.

NOTE

See Programming Guidelines for Multi-Channel Multi-Queuing.

11.7.2.6.3 Rx Queue to DMA Mapping

The packets in the MTL Rx Queues can be routed to any one of the multiple DMA channels by programming the register MTL_RxQ_DMA_Map0 Register for queues 0 through 3, and MTL_RxQ_DMA_Map1 for queue 4.

The following types of Rx Queue to DMA mapping are possible through programming.

11.7.2.6.3.1 Static Mapping

In this mode, all of the packets of an Rx Queue are connected to a specific DMA channel. For example, all the packets from Rx Queue 0 can be routed to a DMA channel by programming Q0MDMACH (bit[3:0]) and Q0DDMACH (bit[7] = 0) of the MTL_RxQ_DMA_Map0 Register.

Similarly, packets from other Rx Queues can be routed to any DMA channel by programming register fields corresponding to each Queue.

11.7.2.6.3.2 Dynamic (Per Packet) Mapping

In this mode, the destination DMA channel of a packet being read from a Rx Queue is not constant but decided independently for each packet. For example, if you set the Q1DDMACH bit of the MTL_RxQ_DMA_Map0 register, the static mapping is disabled

for Rx Queue 1 and the value in Q1MDMACH is ignored. The destination DMA channel is decided by the MAC receiver for each packet, depending on the following in decreasing order of priority:

1. L3-L4 filter Based DMA Selection: The TCP/UDP and IP header fields of the received packet are matched against the corresponding values programmed and enabled for comparison in the MAC_L3_L4_Address_Control register. If the match is successful, the DMA channel number programmed in the DMCHN field of the MAC_L3_L4_Ad-dress_Control register is selected as the destination DMA channel number provided DMCHEN bit of the same register is set. If none of the L3-L4 Registers give a comparison match, then the module proceeds to the next step below.
2. Extended VLAN Based DMA Selection: Extended routing is applicable only if the VLAN Filter has passed. Routing is only based on Perfect Filter Result. Each perfect filter has a DMA Channel Enable and a DMA Channel Number field which can be programmed. Routing is applicable for a filter, only if the DMA Channel Enable bit is set. The frame is routed to the smallest Matching Filter's DMA Channel provided it is enabled. If that filter's DMA Channel number is not enabled, the frame gets routed to Channel 0. For example, if a frame's VLAN tag matches filters 7, 3, and 1. Then the MAC checks if Filter 1's DMA Channel Number is enabled through programming. If yes, the frame gets routed to the programmed value; otherwise it gets routed to DMA. When the inverse filter is enabled; is routed to the least mismatched filter's DMA channel number provided it is enabled. If the DMA Channel enable bit is not set, then the frame is routed based on DA based addressing or to Channel 0.

   If Hash Filter is also enabled, it is used to determine the Filter result only. Routing will still depend on the Perfect Filters enabled. If none of the perfect filters are enabled or if all of them are bypassed, then the VLAN Filter based routing will not take place. The frame will be routed via DA Based addressing or to Channel 0. If all the perfect filters give a fail result and the Hash Filter has passed, then the VLAN Filter result is a pass but routing will be based on DA Based Addressing or to Channel 0. Similar behavior is seen when inverse Filtering is enabled as well.
3. Ethernet DA-Based DMA Selection: The DA address of the received packet is compared against the programmed DA values in MAC Address Registers. If the address matches any of the programmed values, the corresponding DCS field (when enabled) determines the destination DMA channel number.

If none of the previous operations is able to make a successful match/decision, then the packet is routed to DMA Channel 0 by default.

11.7.2.6.3.2.1 Broadcast/Multicast Packet Duplication

Broadcast/Multicast Packet Duplication feature provides a mechanism to send the received Broadcast/Multicast packets to multiple DMA channels.

The DCS field of the of MAC_Address0_High and other optional MAC_Address(#i)_High (i=1 to 2) registers determines the DMA channel number to which the received packet (that matches the MAC Address present in that register) must be routed.

Packet duplication is not supported in other DMA channel routing mechanisms such as Extended VLAN, or L3-L4 based routing.

The Packet Duplication feature is supported only on the highest MTL Queue configured. Therefore, packet duplication for Broadcast/Multicast packets requires that the MCBCQ field of MAC_RxQ_Ctrl1 register is programmed to the highest RxQ present in the configuration.

11.7.2.6.4 Selection of Tag Priorities Assigned to Rx Queues

The VLAN Tag priorities can be assigned to Rx Queues by programming the PSRQ field in the corresponding MAC_RxQ_Ctrl2 registers. The bit corresponding to the VLAN Tag priority can be set in the PSRQ field for assigning that priority to the Rx Queue. For example, if you want to assign VLAN Tag priority of 3 to Rx Queue 0, set bit [3] in PSRQ field of MAC_RxQ_Ctrl2 register. The VLAN Tag priority assigned to particular Rx Queue must be unique, that is, more than one Rx Queue cannot be assigned the same VLAN Tag priority. However, more than one VLAN Tag priorities can be assigned to same Rx Queue.

The settings in the PSRQ field is used for VLAN Tagged Rx packet routing to Rx Queues as well as for PFC based Tx flow control. The received VLAN Tagged Rx packet is routed to Rx Queue that has the VLAN Tag priority match. In PFC based Tx flow control, PSRQ field corresponding to a particular Rx Queue is used for enabling VLAN Tag priorities in the PFC packet transmitted when corresponding Rx Queue threshold levels are reached.

11.7.2.6.5 Rx Side Routing from MAC to Queues

IP supports Rx Side Routing from the MAC to Queues. The MAC routes the Rx packets to the Rx Queues based on following packet types in that order

- Unicast/Multicast destination address packets that fail the destination address filter
- Multicast/Broadcast destination address packets that pass the destination address filter
- VLAN Tag Priority field in VLAN Tagged AV data packets

- AV Control packets
- VLAN Tagged IEEE 1588 PTP over Ethernet packets
- Untagged IEEE1588 PTP over Ethernet packets
- VLAN Tag Priority field in VLAN Tagged packets
- Untagged packets

The outer C-VLAN Tagged AV data Rx packets can be routed based on the priorities assigned to Rx Queues through PSRQ field in corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers and the corresponding Rx Queue is enabled for AV through RXQ#EN field in MAC_RxQ_Ctrl0 register. These packets may be single VLAN Tagged with C-VLAN type or Double VLAN Tagged with outer VLAN Tag of C-VLAN type when Double VLAN feature is enabled (EDVLP bit in MAC_VLAN_Tag register is set) with inner C-VLAN Tagged or inner S-VLAN Tagged when SVLAN processing is enabled (ESVL bit in MAC_VLAN_Tag register is set). This type of Rx packet routing is available when AV feature and Multiple Rx Queues are selected in the configuration.

The AV Control Rx packets can be routed based on the Rx Queue number specified in the AVCPQ field in MAC_RxQ_Ctrl1 register and corresponding Rx Queue is enabled for AV through RXQ#EN field in MAC_RxQ_Ctrl0 register. These packets may be single VLAN Tagged with C-VLAN type or Double VLAN Tagged with outer VLAN Tag of C-VLAN type when Double VLAN feature is enabled (EDVLP bit in MAC_VLAN_Tag register is set) with inner C-VLAN Tagged or inner S-VLAN Tagged when SVLAN processing is enabled (ESVL bit in MAC_VLAN_Tag register is set). This type of Rx packet routing is available when AV feature and Multiple Rx Queues are selected in the configuration.

The VLAN Tagged Rx packets can be routed based on the priorities assigned to Rx Queues through PSRQ field in corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers and the corresponding Rx Queue is enabled for DCB/generic through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

The untagged IEEE 1588 PTP over Ethernet Rx packets can be routed based on the Rx Queue number specified in the PTPQ field in MAC_RxQ_Ctrl1 register and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. The VLAN tagged IEEE 1588 PTP over Ethernet Rx packets can be routed based on the priorities assigned to Rx Queues through PSRQ field in corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers or the Rx Queue number specified in the PTPQ field in MAC_RxQ_Ctrl1 register and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This is determined by programming in TPQC field of MAC_RxQ_Ctrl1 register. This type of Rx packet routing is available when IEEE 1588 Timestamp feature support and Multiple Rx Queues are selected in the configuration. The VLAN Tagged IEEE 1588 PTP over Ethernet Rx packets are detected only when

802.1AS mode is disabled (AV8021ASMEN bit in MAC_Timestamp_Control register is set to 0), otherwise this type of Rx packets are routed as generic VLAN Tagged Rx packets.

The multicast/broadcast destination address Rx packets that pass the destination address filter can be routed based on the Rx Queue number specified in the MCBCQ field of MAC_RxQ_Ctrl1 register when enabled through MCBCQEN bit of MAC_RxQ_Ctrl1 register and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

The untagged Rx packets can be routed based on the Rx Queue number specified in the UPQ field of MAC_RxQ_Ctrl1 register and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

The unicast destination address Rx packets that fail the destination address filter can be routed based on the Rx Queue number specified in the UFFQ field of MAC_RxQ_Routing_Ctrl register when enabled through UFFQE bit of MAC_RxQ_Routing_Ctrl register, RA bit of MAC_Packet_Filter register is set to 1 and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

The multicast destination address Rx packets that fail the destination address filter can be routed based on the Rx Queue number specified in the MFFQ field of MAC_RxQ_Routing_Ctrl register when enabled through MFFQE bit of MAC_RxQ_Routing_Ctrl register, RA bit of MAC_Packet_Filter register is set to 1 and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

If the Rx packet cannot be classified in any of the defined packet type for routing, it will be routed through Rx Queue 0.

11.7.2.6.6 Rx Side Arbitration Between DMA and MTL

IP controller supports Rx side arbitration between the DMA and the MTL.

After completing the current packet processing, the DMA Channel Controller fetches the next descriptor. After the descriptor fetching is complete, the DMA Channel Controller evaluates the amount of data to be to be transferred to the Rx Buffer based on the programmed PBL and Rx Buffer Length. Accordingly, it requests the MTL to transfer the data.

After servicing the current request, the MTL Rx Queue arbitration scheme selects Rx Queue based on the arbitration scheme (RAA field of the MTL_Operation_Mode register) and the weights programmed in the Queue <n> Receive Control Register. The arbitration is done among Queues for which DMA is ready to service. After the Rx Queue is selected, PBL (Programmable Burst Length) amount of data is read out from that Queue and is routed to the Rx DMA Channel based on the Rx Channel selection criteria.

The arbitration takes place when every PBL data transfer is completed and descriptors are ready for the processing from at least one DMA Channel.

11.7.2.6.7 Tx Side Arbitration between DMA and MTL

IP support Tx side arbitration between the DMA and the MTL. Since the number of Tx DMA channels is always equal to the number of Tx Queues in the MTL, they are always mapped directly. For instance, each Tx DMA pushes data into its respective Tx Queue assigned to it.

The data inside each Tx Queue is stored in packets. Hence, if two DMAs are allowed to transfer data into the same queue, when a Tx DMA starts a packet transfer, the other DMA cannot transfer data unless the previous packet is completely pushed-in. This means that the second DMA remains idle until the first packet is transferred. Hence, each DMA is always connected directly to its corresponding Tx Queue.

11.7.2.6.8 Time-Sensitive Networking (TSN)

The ENET QOS block supports TSN in 100 Mbps or 1000 Mbps mode, which enables the transfer of time-sensitive traffic over a network. The block implements the IEEE 802.1Qbv-2015 (Enhancements to Scheduling Traffic), IEEE 802.3br (Interspersing Traffic) and IEEE 802.1Qbu (Frame Preemption) TSN protocol.

The Transmit path of Queue 0 supports the strict priority algorithm and is used for best-effort traffic while transmit paths of additional queues support traffic management by using the credit-based shaper algorithm. For a queue, the credit-based shaper algorithm determines that a queue is available for transmission if the following conditions are true:

- The queue contains one or more packets.
- The credit for the queue is positive as per the algorithm.
- AV traffic can be received on all queues. The block can also receive untagged PTP packets in addition to AV traffic on any queue.

11.7.2.6.9 Queue Modes

Both Tx and Rx queues can handle AV traffic.

Queueing is based on WRR (Weighted Round Robin) or WSP (Weighted Strict Priority) algorithm for generic traffic while it is based on CBS (Credit Base Shaper) and SP (Strict Priority) algorithm for AV traffic.

Whether an Rx Queue is configured for generic or AV-based routing depends on the RXQ#EN field of corresponding queue. Queueing is done based on VLAN tag priority. The VLAN tag priority should match the PSRQ field of the MAC_RXQ_CTRL2 registers. The Rx packets can also be routed to a particular DMA channel based on the DCS field of perfectly-matched MAC Address register.

By default, untagged packets in generic traffic are routed to Receive queue specified in UPQ field of MAC_RXQ_CTRL1 register. Queue 0 is the default value of UPQ field. You can override the default value with any other value, for the UPQ field.

AV control packets (tagged or untagged) are routed based on the AVCPQ field of the MAC_RXQ_CTRL1 register. The PTP over Ethernet packets are routed based on the AVPTPQ field of MAC_RXQ_CTRL1 register.

11.7.2.6.10 Queue Priorities

Priority of an Rx queue can be programmed in corresponding field of MAC_RxQ_Ctrl2 registers. AV queue (high priority) should be programmed as higher priority than the Best-effort queue (low priority).

11.7.2.6.11 Enhancements to Scheduled Traffic (EST)

The IEEE 802.1Qbv-2015 defines the schedule for each of the queues on every egress port which makes the implementation aware of traffic arrival schedule. This information can be used to block the lower priority traffic from transmission in this time window/slot. This ensures that scheduled traffic is forwarded from sender to receiver through all the network nodes with a deterministic delay.

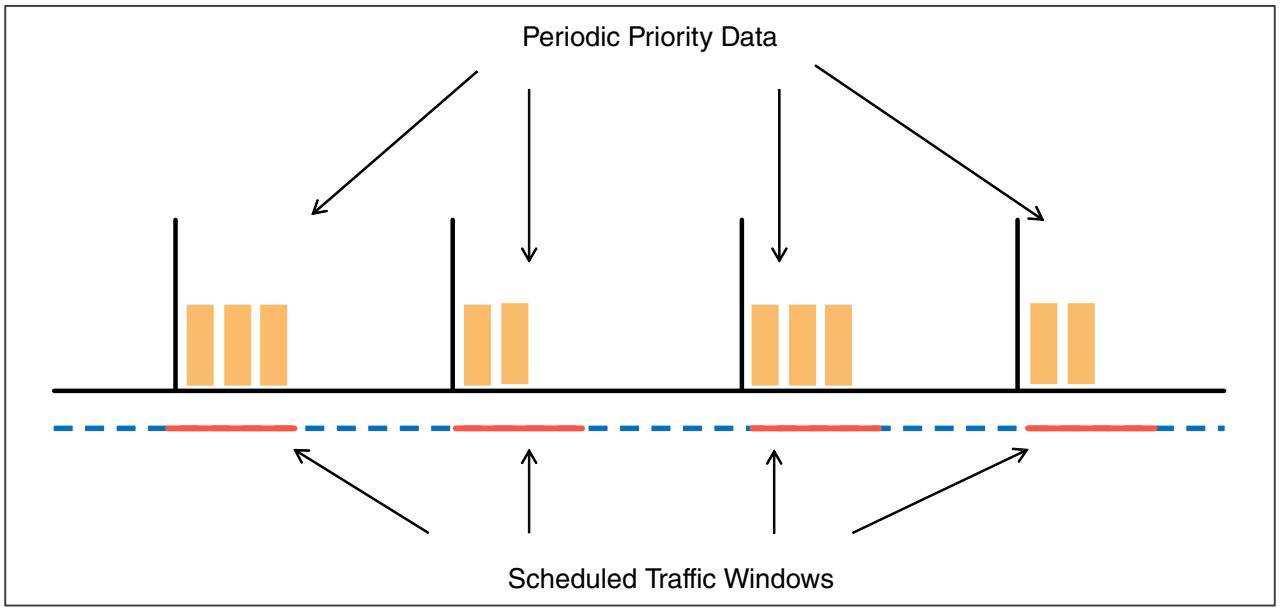


Figure 11-92. Time Aware Shaper Implementation for Traffic Scheduling

One of the important requirements to achieve a low latency is to ensure there are no interfering frames during the scheduled windows that are reserved for high priority traffic. The use of scheduled traffic imposes limitations while starting a transmission.

As shown in the following figure, if an interfering frame begins transmission just before the start of a reserved time period, it can extend transmission into the reserved window, and potentially interfere with higher priority traffic. Therefore, a guard band whose size is equal to the largest possible interfering frame is required before the window starts.

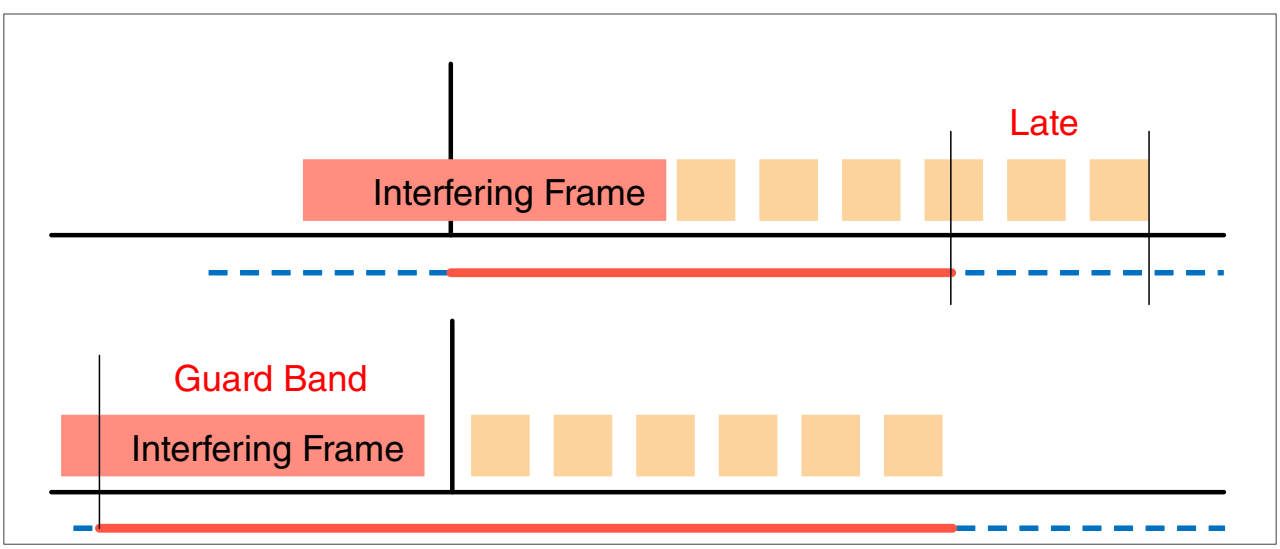


Figure 11-93. Implementing a Guard Band to Avoid Delays Due to Interfering Frames

A larger guard band equates to a less efficient use of network bandwidth. However, with the implementation of IEEE802.1Qbu (frame preemption), this issue is addressed. Frame preemption breaks the interfering frame into smaller fragments. Therefore, the guard band needs to be only as large as the largest possible interfering fragment instead of the largest possible interfering frame.

During the guard band, only the frames that can complete the transmission of the entire frame before the next gate close event are permitted. This ensures that the high priority traffic can always start at the beginning of the window reserved for it.

11.7.2.6.11.1 Frequently Used Terms in EST

- Gate Control List: The list in the hardware memory that holds the gate controls and the associated time intervals.
- Gate Controls: For a given schedule (row in gate control list) there is a Gate Open (O) and Gate Close (C) state associated with each traffic class. The set of O or C values, whose width is same as configured TCs is called the Gate Controls. Example: CCOCO means TC4=C, TC3=C, TC2=O and so on.
- Time Interval: Time interval (in nanoseconds) is a 16, 20, or 24-bit configurable field in the Gate Control list that indicates the time for which the associated gate controls are valid.
- Base Time Register: Each Gate Control List is associated with a 64-bit Base Time Register that holds the start time (in PTP format) for the list.

11.7.2.6.11.2 Updates to the Transmit Scheduling to Support EST

To support EST, following updates are required to the transmit scheduling:

- Implementation of Gate Control List (GCL)
- Enforcing gate controls in the scheduler
- Accounting for Gate Open (O) duty cycle in the computation of idleSlope (CBS)

11.7.2.6.11.2.1 Implementation of Gate Control List (GCL)

Following figure shows block diagram from the IEEE 802.1Qbv spec that illustrates how the gate control list governs the gate close(C) and open(O) events based on the schedule provided for each event. GCL has the following two parts:

- Time Interval: Defines the time in nano seconds for which the gate controls are valid and should be applied before reading the next gate controls from the list. Configurable width of 16, 20 or 24 to represent a maximum of 64us, 1ms, or 16ms schedule interval respectively. Supports left shift of the time interval upto 7 bits to be

able to apply a multiplication factor from 1 to 128ns (in steps of 2(power)n). The maximum value (post shifting) of this field must be 999,999,999 ns.

- Gate Control: Defines the Open (O) represented by logic 1 or Close (C) represented by logic 0 state for the gate of each TC.

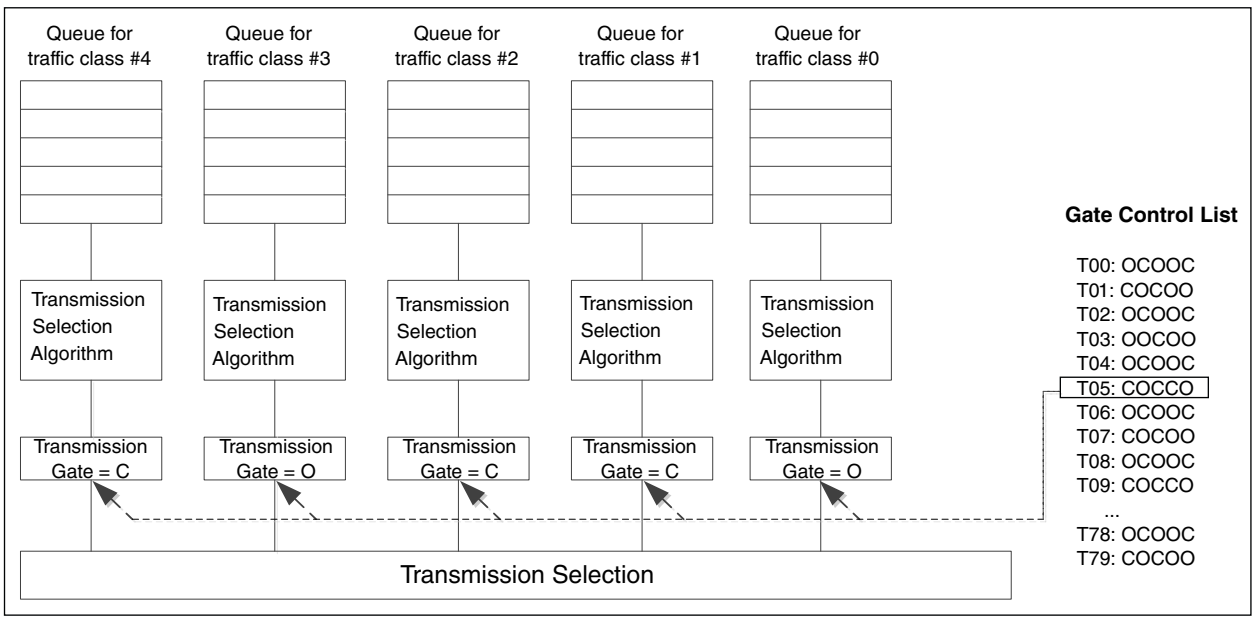


Figure 11-94. Block Diagram from IEEE 802.1 Qbv Specification - GCL Governing Gate Close and Open Events

The implementation of GCL consists of the following two Gate Control Lists:

- HWOL - Hardware Owned List which is a list for hardware access.
- SWOL - Software Owned List which is a list for software access.

The access to these lists is mutually exclusive. The ownership to the list is set by hardware in the SWOL field of MTL_EST_Status register. Following table provides the implementation details of GCL.

Table 11-110. External Memory Used for Holding the Two Gate-Control Lists

Gate Control (up to 5 bits)	Time Interval (ns) 16, 20, or 24 bits	
OOCCC	T0	HWOL or SWOL
OOOOC	T1	
CCCCO	T2	
CCCCC	T3	
I	I	
I	I	

Table continues on the next page...

Table 11-110. External Memory Used for Holding the Two Gate-Control Lists (continued)

Gate Control (up to 5 bits)	Time Interval (ns) 16, 20, or 24 bits	
OCOCO		
OOOOC	T last	
OOOOC	T0	SWOL or HWOL
OOOOC	T1	
OOCCC	T2	
OOOOC	T3	
I	I	
I	I	
OCOCC		
OOCCC	T last	

11.7.2.6.11.2.2 Registers Related to Gate Control List

Following are the set of 4 registers (one for each GCL) related to GCL. These registers are implemented through Indirect Addressing using the GCL_Control and GCL_Data registers.

1. 64-bit Base Time Register (BTR)

2. 40-bit Cycle Time (CTR)

3. m-bit Time Extension (TER)

4. n-bit [Gate Control] List Length (LLR) (n depends of the configured GCL depth)

11.7.2.6.11.2.2.1 Base Time Register (BTR)

Base Time Registers is a 64-bit register that holds the start time to execute the GCL. The format of the BTR is same as the PTP format (upper 32-bits holds time in seconds and lower 32 bits hold time in nano seconds). Once the execution of a given list begins, the implementation can update the value in BTR to indicate the next list execution begin time.

11.7.2.6.11.2.2.2 Cycle Time Register (CTR)

Cycle Time Register is a 40-bit register that holds the time at which the execution of the GCL should be repeated. The Cycle Time Register consists of an 8-bit value in seconds, and a 32-bit value in nano seconds (similar to the PTP time format with truncated seconds register). For a given Gate Control List the start time is "Base Time" + N *

"Cycle Time" where N is an integer value representing the iteration number starting with 0 for first iteration. If the Gate Control List execution takes longer than the Cycle time, then the list is truncated at the Cycle Time and the subsequent loop begins at Cycle Time.

11.7.2.6.11.2.2.3 Time Extension Register (TER)

Time Extension Register (TER) is a m-bit (where m = Configured Time Interval width + 7) register that holds the amount of time (in nano seconds) the current Gate Control List can be extended before switching to the new Gate Control List. This will be useful in avoiding small fragments of the current list being executed before switching to a new list.

11.7.2.6.11.2.2.4 List Length Register (LLR)

List Length Register (LLR) is an n-bit register (when n is 7, 8, 9, 10, or 11 for a GCL configured depth of 64, 128, 256, 512, or 1024 respectively) that holds the integer value of the length of the GCL (that is the number of valid rows in each GCL). The processing of the GCL stops after the number of rows read equals to the LLR value.

11.7.2.6.11.2.3 Transmission Gating Implementation

A Bridge or an end station can be enhanced to allow transmission from each TC that is yet to be scheduled relative to a known timescale. To achieve this, a transmission gate is associated with each TC; the state of the transmission gate determines if the queued frames can be selected for transmission.

For a TC, the transmission gate can be in one of the following two states:

1. OPEN:Queued frames are selected for transmission, in accordance with the definition of the transmission selection algorithm associated with the TC.
2. CLOSED: Queued frames are not selected for transmission.

A frame on a traffic class queue is not available for transmission if the transmission gate is in the closed state or if there is insufficient time available to transmit the entirety of that frame before the next gate-close event associated with that queue.

The implementation has visibility to the current schedule of gate controls and the immediate next schedule of the gate controls. So the maximum Gate Open period does not exceed the sum of the two Time Intervals. This is because, a frame is selected for transmission only if the gate is currently Open and the duration of gate open interval is greater than the time taken to transfer the entire frame.

The implementation must know the frame size before the transmission, so that the transmission overruns can be avoided and only the frames that can complete are scheduled at all times.

The implementation adequately compensates for the implementation delays in the data transfer from the buffer to the line by offsetting the current time with all the relevant delays (provided by the CTOV field of MTL_EST_Control register). This ensures that the schedule provided is always accurately implemented at the line.

11.7.2.6.11.3 Idle Slope Computation Updates

When EST is enabled, credit is accumulated only when the gate is open; therefore, the effective data rate of the idleSlope must be increased to reflect the duty cycle for the transmission gate associated with the queue.

The idleSlope is computed based on the GateOpenTime and OperCycleTime values. Program the idleSlope registers (implemented one per CBS enabled TC) based on the following equation. The existing MTL register has sufficient bit width to accommodate the new values for idleSlope.

idleSlope = (operIdleSlope(N) * OperCycle/GateOpenTime)

11.7.2.6.11.3.1 Operational Details of GCL

Set the switch to software List, the SSWL field of the MTL_EST_Control Register, so that hardware can access the programmed gate control list. The first set of gate controls are applied when the current time is equal to the value in the Base Time Register (BTR) and is held until the programmed "Time Interval" value.

To avoid transmission overruns, one additional gate control event is always read ahead from the list. This enables the GCL to determine the next gate close events (if any) for the TCs that are open.

The scheduling based on the Gate Open state and Time Interval of only the current and subsequent schedule. An internal accumulator is used to add the time intervals when gate controls are applied. BTR + Accumulator holds the time at which the next set of gate controls are to be applied.

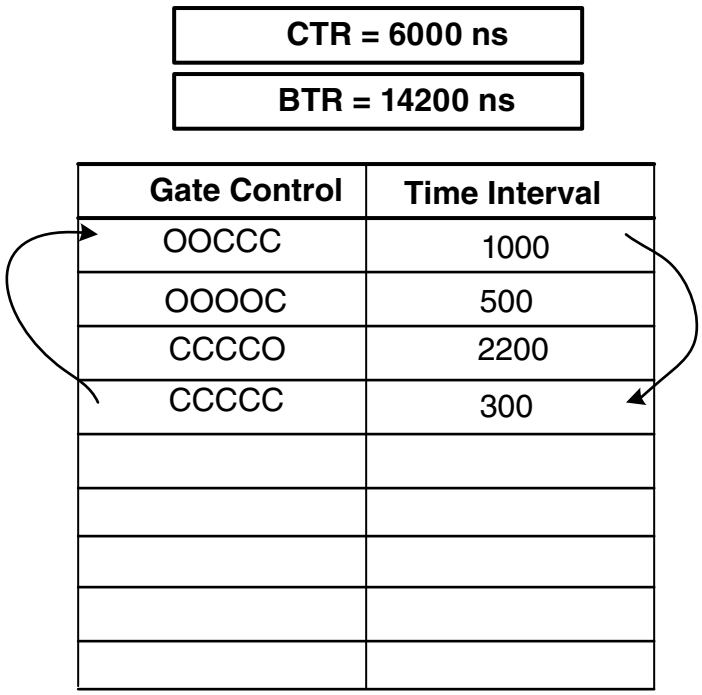


Figure 11-95. GCL and Associated Registers - BaseTime and CycleTime

The GCL is read progressively from the first row adhering to the schedule. The read operations continue until the List Length (from LLR Register) is reached and the execution of the list restarts at BTR + CTR time. At this point the BTR is incremented by value in CTR to mark the beginning of a new cycle. In the absence of any gate controls, all the gates are deemed to be in Open state, during the execution of the list.

In cases where the execution time of the list is greater than the CycleTime, the list is truncated and the next iteration starts when the current time equals BTR + CTR.

Table 11-111. GCL and Associated Registers - BTR and CTR

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
14200	OOCCC	1000	14200
15200	OOOOC	1500	14200
15700	CCCCO	3700	14200
17900	CCCCC	4000	14200
18200	OOOOO	0	20200
20200	OOCCC	1000	20200
21200	OOOOC	1500	20200

In the example in table, the execution starts at 14200 and the first set of Gate Controls "OOCCC" are applied immediately. As the time interval is 1000ns the next set of gate controls are applied at 14200 (BTR) + 1000 (Accumulator) = 15200 ns . As there are no gate controls available after the execution of the last gate control and before the next iteration of the loop, the gates are deemed to be in open state during that time period as depicted at time 18200 in the table.

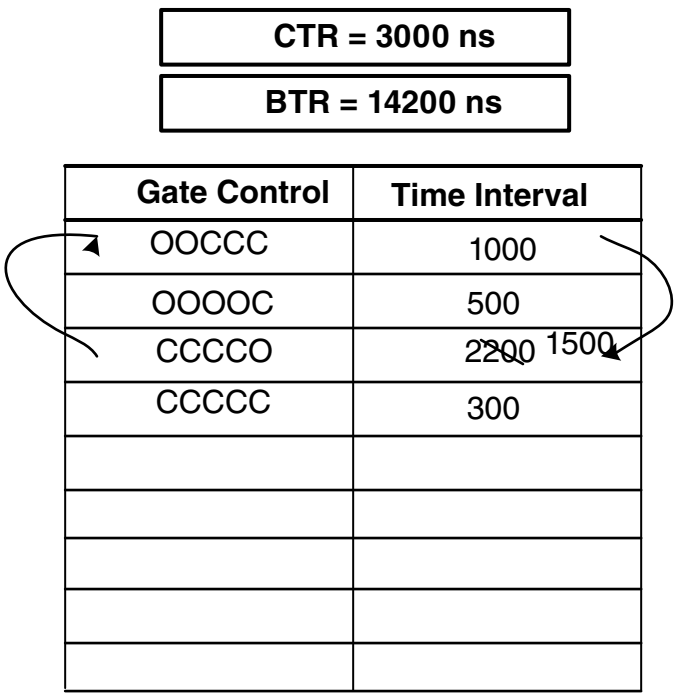


Figure 11-96. GCL and Associated Registers - BaseTime and CycleTime List Execution Time > CycleTime

As the list execution takes longer than the allocated CycleTime, the list is truncated and the list is started from the BTR+CTR as shown in the following table.

Table 11-112. GCL and Associated Registers - BTR and CTR, Execution Time > Cycle Time

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
14200	OOCCC	1000	14200
15200	OOOOC	1500	14200
15700	CCCCO	3700	14200
17200	OOCCC	1000	17200
18200	OOOOC	1500	17200
18700	CCCCO	3700	17200

While applying the third set of gate controls, the BTR + CycleTime (17200) < BTR + Accumulator (17900), so the list is truncated and execution switches to a new iteration at 17200.

11.7.2.6.11.3.2 Installing a New GCL

When a new software programmed GCL is available and must be executed at the new BTR value, the switch to the new GCL can happen in one of the following ways.

- New Base Time aligned with current schedule
- New Base Time unaligned with current schedule

11.7.2.6.11.3.2.1 New Base Time Aligned with Current Schedule

If the choice of cycle time for the new gating cycle is unchanged from the cycle time for the current gating cycle, and if the BTR chosen for the new gating cycle (new BTR) is an integer multiple of the current cycle time (+ current BTR), then the new gating cycle exactly lines up with the old gating cycle, that is, the cycle start times for the new gating cycle is same as they would have been for the old configuration. This could be considered to be the ideal case and allows the new gating cycle to be installed and executed with no timing issues. The implementation completes the execution of an iteration of the current list and switches to the new list at the beginning of the BaseTime listed in the new list.

If (New BaseTime > = Current Time)

ConfigChangeTime = New BaseTime

Else If (New BaseTime < Current Time)

1. Set the BTRError

2. ConfigChangeTime = (New BaseTime + N* New CycleTime)

where N is the smallest integer for which the relation ConfigChangeTime >= CurrentTime and (N =< 8) is TRUE.

When N > 8 the hardware cannot auto recover and the loop count value in BTRError reporting is set to 1111 requiring the software to reprogram the New Base Time.

Following figure illustrates the installation of the new GCL along with the timelines.

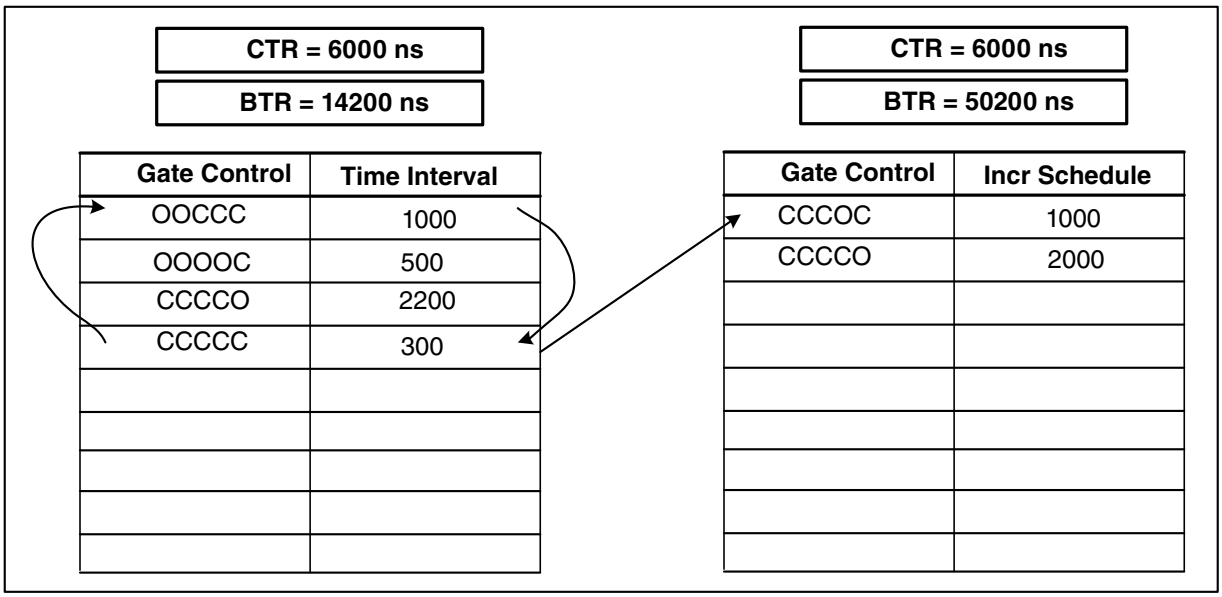


Figure 11-97. Switching to a New Configuration that is Aligned with the Existing Configuration

In the above example, after the sixth iteration of the first GCL, the BTR values of the old and new GCL are equal. At that point the new GCL is processed as a natural extension to the existing GCL.

Table 11-113. GCL and Associated Registers - BTR and CTR

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
44200	OOCCC	1000	44200
45200	OOOOC	1500	44200
45700	CCCCO	3700	44200
47900	CCCCC	4000	44200
48200	OOOOO	0	50200
50200	CCCOC	1000	50200
51200	CCCCO	3000	50200
53200	OOOOO	0	56200

11.7.2.6.11.3.2.2 New Base Time Unaligned with Current Schedule

If new CycleTime differs from current CycleTime or New BaseTime is in the future and is not an integer multiple of current CycleTime, then the old and new cycles do not line up; when new BaseTime is reached (when the new configuration is installed and starts to execute), the last old cycle is normally truncated to start the first new cycle. This could be undesirable if it results in a very short last old cycle; arguably it would be better to simply extend the penultimate old cycle by that small amount, rather than starting a very short

cycle. The Cycle Time Extension Register (related to the current config list) allows this extension of the last old cycle to be done in a defined way; if the last complete old cycle ends normally in less than current Cycle Time Extension (TER) ns before the new base time, then the last complete cycle before new BaseTime is reached is extended so that it ends at new BaseTime.

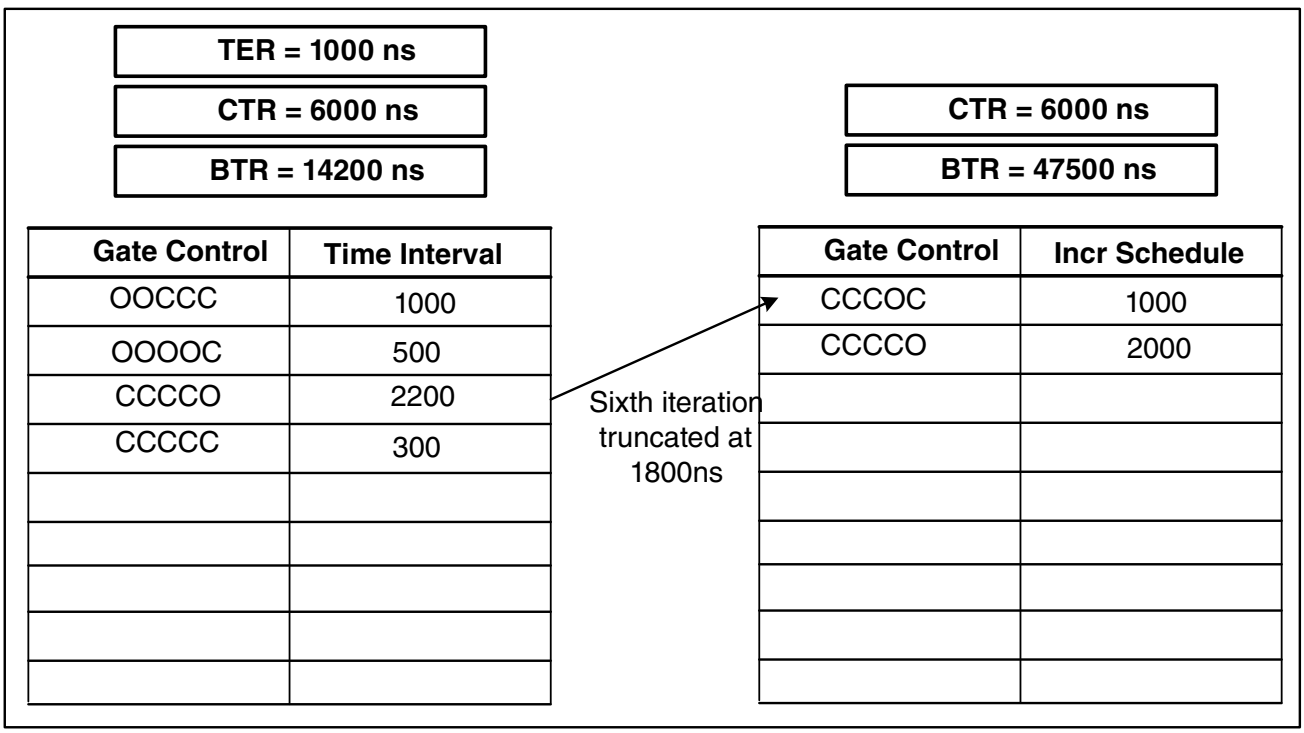


Figure 11-98. Switching to New List Truncating the Current List

At the end of the fifth iteration the Current time + Cycle time extension (TER) < New BTR so the sixth iteration of current configuration is started. During the sixth iteration of the current list when the new BTR value is smaller than the next schedule in the current list, it switches to the new list.

Table 11-114. Extending to New List By Truncating the Current List

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
44200	OOCCC	1000	44200
45200	OOOOC	1500	44200
45700	CCCCO	3700	44200
47500	CCCOC	4000	44200
48500	CCCCO	0	50200
50500	OOOOO	1000	50200

Below is an example where the current config list is extended instead of starting a new iteration as the extension time of 800ns is less than the allowed cycle extension time (TER) of 1000ns.

Figure 11-99. Switching to New List By Extending the Current List

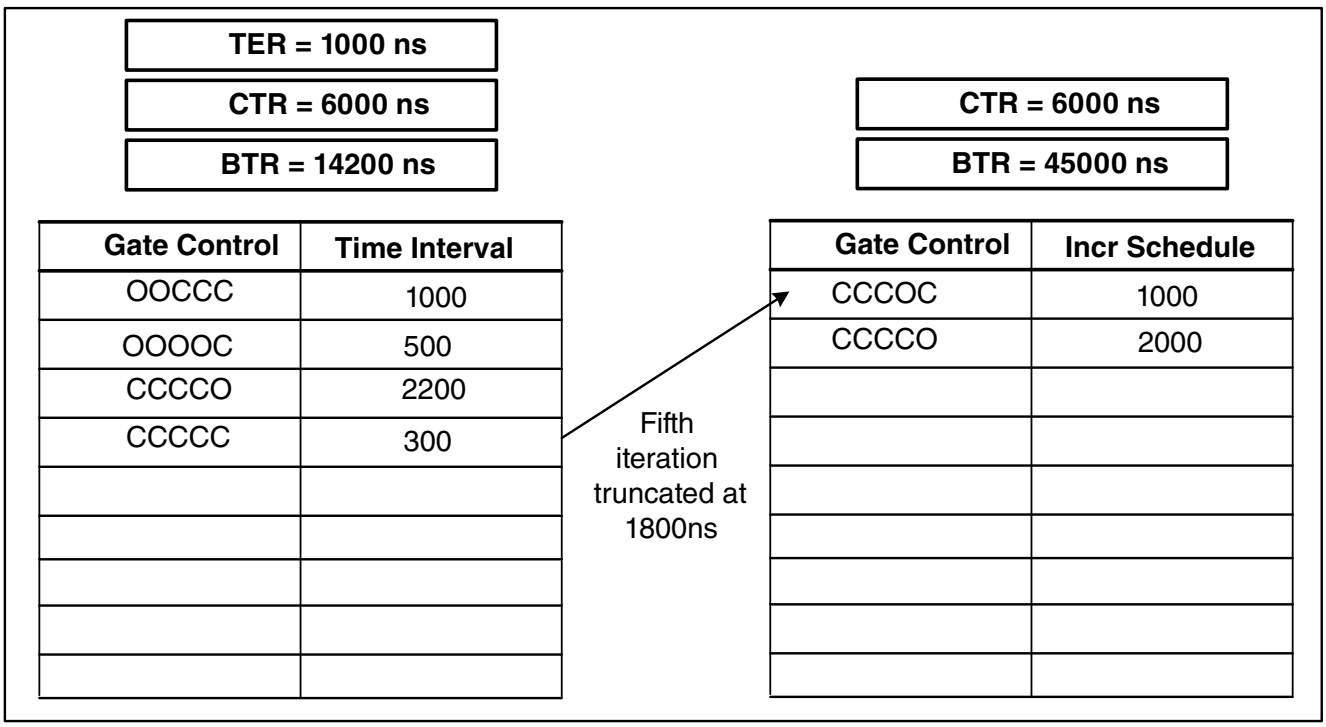


Table 11-115. Switching to New List By Extending the Current List

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
38200	OOCCC	1000	38200
39200	OOOOC	1500	38200
39700	CCCCO	3700	38200
41900	CCCCC	4000	38200
42200	OOOOO	0	44200
45000	CCCOC	1000	45000
46000	CCCCO	3000	45000
48000	OOOOO	0	51000

11.7.2.6.11.4 Impact of Transmit Scheduling Algorithms on EST

When EST is used in isolation, the Gate Control List manages the final open/close state of the Queues along with the checks carried out by the Transmission Selection Algorithm in MTL. As the Gate Controls operate on a predefined repetitive schedule, it is recommended to use Strict Priority or Credit Based Shaper (CBS) scheduling algorithms.

Other algorithms such as the Weighted Round Robin (WRR), Deficit Weighted Round Robin (DWRR) and Weighted Fair Queuing (WFQ) implement masking of the queues based on the current winning queue. The algorithm is applied only among the group of queues that open simultaneously. To ensure Queues whose gates are "Open" get priority, these algorithms are modified to treat "gate open" queues and "gate closed" queues as separate groups giving priority to gate open queues.

For example, consider 4 queues (Q3, Q2, Q1, Q0) with weights 4:3:2:1; Q3 and Q2 are in Open state in slot one slot, while Q1 and Q0 are in Open state in another slot. In this case, the scheduler works as follows:

1. In the first slot, the Q3 and Q2 are serviced in the ratio of 4:3 for the duration the slot is open.
2. In the second slot, the queues Q1 and Q0 are serviced in the ratio of 2:1.
3. Fresh arbitration is started every time a slot is opened.

In other words, the traffic does not get distributed in the intended ratio of 4:3:2:1; but as two groups with different ratios and only for the duration of the slot when the gates are open continuously

NOTE

See Programming Guidelines for EST

11.7.2.6.12 Frame Preemption (FPE)

Frame preemption breaks the interfering frame into smaller fragments. Therefore, the guard band needs to be only as large as the largest possible interfering fragment instead of the largest possible interfering frame. During the guard band, only the frames that can complete the transmission of the entire frame before the next gate close event are permitted. This ensures that the high priority traffic can always start at the beginning of the window reserved for it.

Preemption allows one or more higher priority (express) frames to interrupt the transmission of a lower priority (preemptable) frame; the preemptable frame transmission is resumed and completed after the express frame transmission is complete. To support frame preemption, the following two abstractions of the MAC are used:

- A preemptable MAC, called pMAC, which carries the preemptable traffic.
- An express MAC, called eMAC, which carries the express traffic.

In the implementation, only the parts of the MAC that hold the state during preemption are replicated and represented as pMAC and eMAC. The MAC uses the following two ways to put the transmission of the preemtable traffic on hold, in the presence of express traffic:

- The MTL scheduler interrupts the preemptable traffic that is currently being transmitted. When the preemption capability is active, the MAC interrupts the transmission and reception of preemptable frames. A preempted fragment can be followed by zero or more express frames, before the continuation fragments. The preemptable frame can be fragmented any number of times, however, the minimum final and non-final fragment length criterion must be met. However, interleaving of more than 1 preemptable packet is not permitted. This implies that if a preemptable packet is fragmented by an express packet, another preemptable packet cannot be transferred until all the remaining fragments of the first preempted packet are transferred.
- The MTL scheduler prevents starting the transmission of preemptable traffic. When the preemption capability is inactive, the pMAC entity is disabled and only express traffic is transmitted or received. Frame preemption feature can be enabled by setting EFPE field of the MAC_FPE_CTRL_STS register.

11.7.2.6.12.1 GCL Modification to Support FPE

In the EST-only configuration, the GCL entry has up to 24 bits of Time Interval and up to 8 high order bits representing the Gate Open/Close state requirements as shown in the following figure.

Gate Control (up to 5 bits)	Time Interval(ns) 16, 20 , or 24 bits
OOCCC	T0
OOOOC	T1
CCCCO	T2
CCCCC	T3
OCOCO	T126
OOOOC	T127

Figure 11-100. Gate Control List When FPE is Disabled

EST only supports SetGate operation, which implies that the gates are set to either open or close at a given time interval. However, when both Frame Preemption (FPE) and EST are enabled, the GCL also supports Set-and-Hold-MAC and Set-and-Release-MAC operations, in addition to the SetGate operation. To enable the support of hold and release operations, the format of the GCL is slightly changed while configuring and enabling the FPE. The first Queue (Q0) is always programmed to carry preemption traffic and therefore it is always Open. The bit corresponding to Q0 is renamed as Release/Hold MAC control. The TX Queues whose packets are preemptable are indicated by setting the PEC field of the MTL_FPE_CTRL_STS register. The GCL bit of the corresponding Queue are ignored and considered as always "Open". So, even if the software writes a "0" ("C"), it is ignored for such queues.

Gate Control (up to 4 bits)	Release/Hold Indication	Time Interval(ns) 16, 20, or 24 bits
CCCO	0	T0
CCCC	0	T1
OCCC	1	T2
COCC	1	T3
CCOC	1	T4
CCCO	0	T5
CCCC	0	T6

Figure 11-101. Gate Control List When FPE is Enabled

When the Release/Hold bit transitions from a '0' to '1', a Set-and-Hold-MAC operation is performed. This marks the cease of the preemptable traffic. This is achieved by sending a Hold request to MTL "ha" ns in advance (where ha is the time interval mentioned in the Hold Advance (HADV) field of the MTL_FPE_Advance register). When the Release/ Hold bit transitions from a '1' to '0' a Set-and-Release-MAC operation is performed. This marks the resuming of the preemptable traffic. This is achieved by sending a Release request to MTL "ra" ns in advance (where ra is the time interval mentioned in the Release Advance (RADV) field of the MTL_FPE_Advance register). The preemptable traffic is blocked for the time duration the Release/Hold bit is set to a '1' in the Gate Control List.

11.7.2.6.12.2 Impact of Preemption on CBS

In Credit Based Shaper (CBS), the definition of "Transmit" is:

- TRUE for the duration of frame transmission from the queue.
- FALSE when frame transmission from the queue is complete.
- When CBS algorithm is used with frame preemption, the value of "Transmit" is TRUE only while the frame is being transmitted by the MAC. If the frame transmission is delayed or interrupted (for instance, a preemptable frame transmission is interrupted to allow the transmission of an express frame from a different queue, or the start of an express frame is delayed because a preemptable frame is being transmitted) the value of "Transmit" is FALSE until transmission of the frame commences or is resumed.

Also, the value of "Transmit" is FALSE during the transmission of any overhead that is a consequence of frame preemption, (for example, when additional frame overhead, such as mCRC or Fragment Count, is added to the preemptable frame.

At any given time, if there are no frames in the queue, and the value of Transmit is FALSE, and credit is positive value, the credit value is set to zero if there is no preemptable frame from the queue for which transmission is in progress but has been interrupted.

11.7.2.6.12.3 mPacket Format

When the preemption capability is active, MAC sends mPackets to the PHY. An mPacket can be one of the following:

1. A express packet
2. A preemptable packet
3. An initial fragment of a preemptable packet
4. A continuation fragment of a preemptable packet

Start mPacket Delimiter (SMD)

The value of the SMD indicates whether the mPacket contains an express packet, the start of a preemptable packet (initial fragment or complete packet), or any of continuation fragments of a preemptable packet. Following table shows the valid SMD values.

Table 11-116. Possible SMD Values of mPacket

mPacket Type	Notation	Frame Count	Value
verify packet	SMD-V	-	0x07
respond packet	SMD-R	-	0x19
express packet	SMD-E	-	0xD5
preemptable packet start	SMD-S0	0	0xE6
	SMD-S1	1	0x4C
	SMD-S2	2	0x7F
	SMD-S3	3	0xB3
continuation fragment	SMD-C0	0	0x61
	SMD-C1	1	0x52
	SMD-C2	2	0x9E
	SMD-C3	3	0x2A

frag_count

A frag_count is a modulo-4 counter that increments for each continuation fragment of the preemptable packet. The frag_count protects against mPacket reassembly errors by enabling detection of the loss of up to 3 packet fragments.

The frag_count field is present only in mPackets with SMD-C notation (continuation fragment). The frag_count is zero in the first continuation fragment of each preemptable packet.

Table 11-117. Possible frag_count Values

frag_count	Value
0	0xE6
1	0x4C
2	0x7F
3	0xB3

mData

The contents of the packet from the MAC, starting with the first byte after the SFD to the last byte before the FCS are sent in the mData fields of one or more mPackets for that frame. The minimum size of the mData field is 60 bytes.

CRC The CRC field contains a cyclic redundancy check (CRC) and has an indication of the final mPacket of a frame. In the final mPacket of a frame, the CRC field contains the last 4 octets of the MAC frame (the FCS field).

For other mPackets, the CRC field contains an mCRC value. The mCRC is calculated on the octets of the packet from the first octet of the frame (the octet following the SFD of preemption frames) to the last octet of the packet transmitted in that mPacket by performing an XOR of the calculated 32 bit CRC value of the fragment and a value of 0x0000FFFF.

Summary of Packet Formats

- Express Packet: 7bytes of PREAMBLE, SMD-E, Data, and CRC
- Complete Preemptable Packet: 7 bytes of PREAMBLE, <Current Preemptable packet SMD>, Data, CRC
- Initial Fragment (non-final) of Preemptable Packet: 7 bytes of PREAMBLE, <Current Preemptable packet SMD>, Data, mCRC
- Continuation fragments (non-final) of Preemptable packet: 6 bytes of PREAMBLE, <Current Preemptable continuation fragment SMD>, <Current Preemptable continuation fragment FC>, Data, mCRC
- Final fragment of Preemptable packet: 6 bytes of PREAMBLE, <Current Preemptable continuation fragment SMD>, <Current Preemptable continuation fragment FC>, Data, CRC

Table 11-118. Current and Previous SMD Values

Previous Preemptable packet SMD	Current Preemptable packet SMD
SMD-S0	SMD-S1
SMD-S1	SMD-S2
SMD-S2	SMD-S3
SMD-S3	SMD-S0

Table 11-119. Current and Previous SMD Values

Previous Preemptable fragment SMD	Previous Preemptable fragment FC	Current Preemptable continuation fragment SMD	Current Preemptable continuation fragment FC
SMD-S0	NA	SMD-C0	FC0
SMD-S1	NA	SMD-C1	FC0
SMD-S2	NA	SMD-C2	FC0
SMD-S3	NA	SMD-C3	FC0
SMD-C0	FC0	SMD-C0	FC1
SMD-C0	FC1	SMD-C0	FC2
SMD-C0	FC2	SMD-C0	FC3
SMD-C0	FC3	SMD-C0	FC0
SMD-C1	FC0	SMD-C1	FC1
SMD-C1	FC1	SMD-C1	FC2
SMD-C1	FC2	SMD-C1	FC3
SMD-C1	FC3	SMD-C1	FC0
SMD-C2	FC0	SMD-C2	FC1
SMD-C2	FC1	SMD-C2	FC2
SMD-C2	FC2	SMD-C2	FC3
SMD-C2	FC3	SMD-C2	FC0
SMD-C3	FC0	SMD-C3	FC1
SMD-C3	FC1	SMD-C3	FC2
SMD-C3	FC2	SMD-C3	FC3
SMD-C3	FC3	SMD-C3	FC0

11.7.2.6.12.4 Transmit Preemption

When FPE is enabled (setting EFPE=1 in MAC_FPE_CTRL_STS register), the MTL preempts a preemptable frame, when a "hold" request is asserted (EST/Qbv configured and enabled) express frames are available for transmission, that is, frame is present in MTL FIFO and is qualified for arbitration, after ensuring that the minimum mPacket mData field size is met. Therefore, preemption occurs only if at least 60 bytes of the preemptable frame have been transmitted and at least 64 bytes (including the frame CRC) remain to be transmitted.

The earliest starting position of preemption is controlled by the AFSZ field of the MTL_FPE_CTRL_STS Register. Preemption does not occur until at least 64 * (1+ AFSZ) - 4 bytes of the preemptable frame have been sent.

When preemption occurs, all the preemptable queues are blocked and only the express queues are allowed to arbitrate (if more than one express queue has traffic) and transmit.

Continuation fragment of the preempted frame is the first frame to be transmitted after "release" request is asserted (EST/Qbv configured and enabled) and all the express traffic transmission completes.

NOTE

PTP packets should be transmitted as express packets.

MTL communicates the following frame-type information to the MAC using a 2-bit preemption control signal on the MTI interface qualified by SoF and EoF.

- Express Frame
- Preemption Frame (Full or Fragment)
- Continuation Fragment (non-Final or Final)

Table 11-120. Preemption Control Values on MTI Interface for Various Frame Types

Qualifier	Preemption Control Value	Frame Type
SoF	00	Start Express
SoF	01	Start Preemption
SoF	10	Continuation Fragment
SoF	11	Reserved
EoF	00	End of Frame
EoF	01	End of Fragment

MTL should wait for the previous fragment status to be received before resuming the continuation fragments of a preempted frame. Fragment status is described in detail in the Tx Fragment Status section.

11.7.2.6.12.4.1 MAC Tx Preemption

MAC supports preemption by implementing the functionality needed to generate the mPackets as described in “mPacket Format”.

Based on the Preemption Control value received on the MTI interface (qualified with SoF and EoF), MAC determines the frame type (shown in the above table 'Preemption Control Values on MTI Interface for Various Frame Types') and generates mPackets accordingly.

When the preemption capability is active, MAC replaces the SFD of a preemption packet with an SMD-S value. A 2-bit rolling frame count is encoded in the SMD-S value.

The SMD-E value is the same as the SFD value so the SFD of an express packet does not need to be replaced.

11.7.2.6.12.4.1.1 Tx Fragment Status

MAC sends Tx fragment status to indicate successful transmission of fragmented mPackets.

In case of a transmission error (underflow, jabber, and so on) the frame status is sent (and not a fragment status) with an error indication along with all other relevant status fields. In case of receiving an error status for a transmitted fragment, the MTL drops the remaining fragments and does not send any more continuation fragments.

11.7.2.6.12.5 Receive Preemption

11.7.2.6.12.5.1 MAC Receive Preemption

When FPE is enabled, the MAC Receiver pases the incoming packets and differentiates between Express packets and preemptable packets. An SMD containing an SMD-E indicates express packet, and SMD containing an SMD-S indicates the first mPacket of a preemptable packet.

If an mPacket containing an SMD-S is received when MAC has not completed receiving the previous preempted packet, MAC sets a CRC Error status for the previously received partial packet.

When an SMD-S is detected, MAC records the frame count indicated by the SMD and then begins sending data on the MRI interface.

The MAC checks the last four bytes of the mPacket . If the last four bytes of the mPacket do not match CRC, that indicates the end of the packet with or without a CRC error as per the CRC check result. If the last four octets of the mPacket match, that indicates that the packet was preempted.

An SMD containing an SMD-C indicates an mPacket that continues the data for a preempted packet. Upon receiving an SMD value of SMD-C, MAC checks the following:

1. A preempted packet is in progress
2. The frame count indicated by the SMD matches the frame count of the packet in progress
3. The frag_count value indicates the next fragment count.

If any of the above check fails, the mPacket is discarded and MAC sets a CRC Error status for the partially received packet.

If all the checks pass, the next fragment count is incremented modulo 4.

When a packet is preempted, the MAC saves the state of the partially received packet (filter check status, timestamp, length fields etc.) and will be able to process any received Express packets before the continuation fragment is received.

The MAC Receiver sends a "dummy status" for all the mPacket fragments successfully received and sends the Rx status with the final fragment. If an error is detected during any of the fragments the Rx status is sent and the fragment is marked as final fragment. All subsequent continuation fragments received for this packet are dropped in the MAC.

The MAC communicates the following frame type information to the MTL using a 2-bit preemption control signal.

1. Express Frame
2. Preemption Frame (Full or Fragment)
3. Continuation Fragment (non-Final or Final)

Table 11-121. Preemption Control Values on MRI Interface for Various Frame Types

Qualifier	Preemption Control Value	Frame Type
SoF	00	Start Express
SoF	01	Start Preemption
SoF	10	Continuation Fragment
SoF	11	Reserved
EoF	00	End of Frame
EoF	01	End of Fragment

11.7.2.6.12.5.1.1 Data Alignment

When a received frame cannot be fragmented on any byte boundary, MAC retains the unaligned bytes of data in the previous fragment and resends them with the next fragment as shown in the following figure.

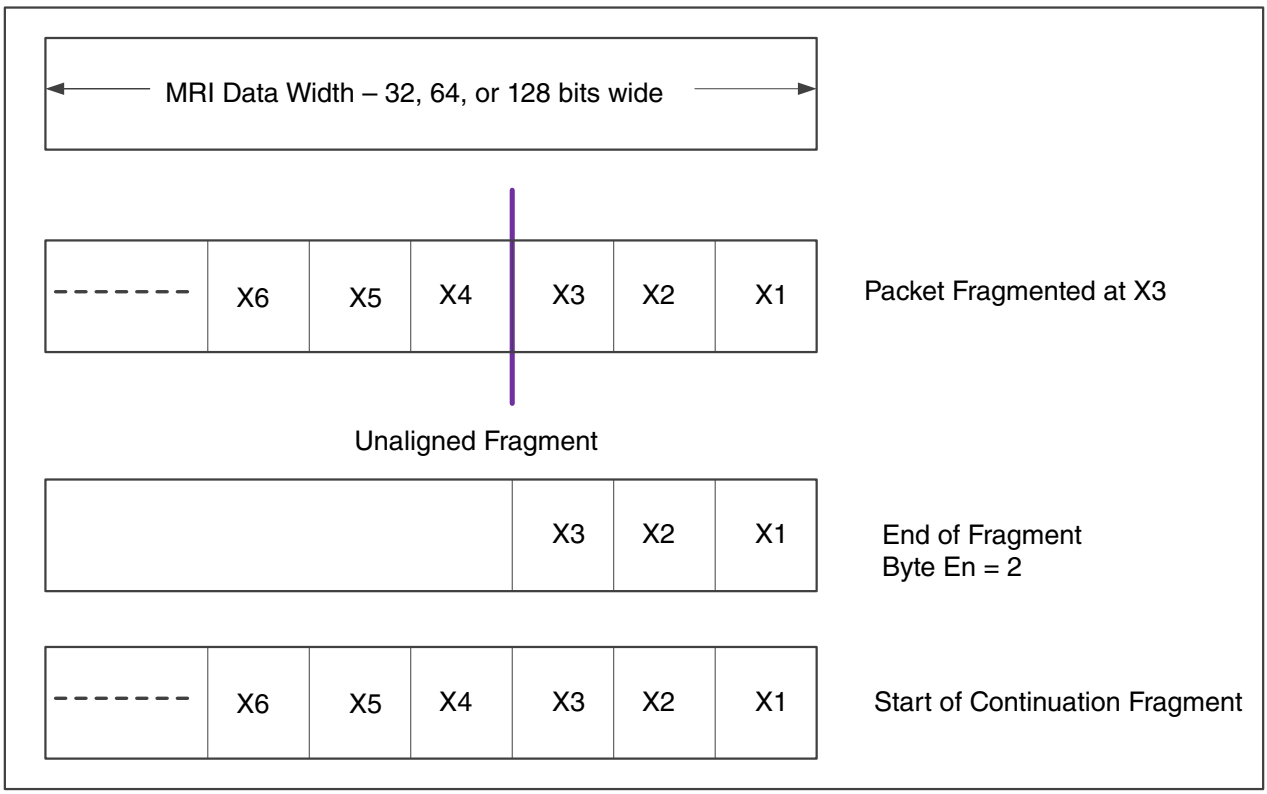


Figure 11-102. Data Alignment Feature of MAC

11.7.2.6.12.5.1.2 MTL Receive Preemption

The MTL Rx must have at least 2 Rx queues to support FPE function as the preemptable packets and the express packets must be routed to separate Rx queues. The destination Rx queue of a received packet is controlled by MAC_RxQ_Ctrl[n] registers. Program these registers such that Preemptable traffic and express traffic are not routed to the same Rx queue. As the queue mapping for tagged packets is based on VLAN user-priority field, this implies that priority of preemptable and express packets are mutually exclusive. In other words, packets of a certain priority (traffic class) are either express or preemptable but cannot be both.

For the preemptable frames, in addition to the PSRQ (Priority Selected in Rx queue) based queue routing, a programmable "Frame Preemption Residue Queue" (FPRQ) is supported to route all other Preemption Packets received (Untagged, SA/ DA or VLAN Filter Fails forwarded due to RA being set or VTFE being reset).

Table 11-122. Preemption Control Values on MRI Interface for Various Frame Types

Preemption Packet Type	Queue Routing
AV Tagged Packets passing the SA/DA and VLAN filters	PSRQ*
DCB/Generic Tagged Packets passing the SA/DA and VLAN filters	PSRQ**
Tagged Packets failing the SA/DA and VLAN filters without setting the Receive All (RA = 0) or setting VTFE = 1	Dropped
All other packets OR when RA = 1	FPRQ

11.7.2.6.12.5.1.3 MTL Receive Arbitration

On the ARI Interface, data is fetched from the MTL Rx FIFO based on the arbitration selected in the MTL_Operation_Mode register. Frame based arbitration can be used only when all the MTL Preemption Queues are operating in Store and Forward mode, else there will be loss of bandwidth on the ARI interface because all the fragments of a preemptable packet is not available. Therefore, any express packets received between the fragments are blocked until all the fragments are received and transferred; this defeats the purpose of express traffic.

When operating in either Threshold (cut through) or Store and forward modes of operation, PBL based arbitration is recommended over Frame based arbitration. In case of PBL based arbitration, the watermark check is always performed and the arbitration/ transfer of data in terms of chunks of "PBL" size of data which is almost similar to the concepts of "fragments". Therefore the express queue packets get blocked for less time as well as the ARI interface transfers the data without loss of efficiency.

11.7.2.6.12.6 Verify and Respond mPackets

When FPE function is present, the MAC can receive and detect the Verify and Respond mPackets, even when FPE is not enabled by software. When MAC detects valid Verify/ Respond mPackets, it notifies the software by setting the RVER and RRSP fields of MAC_FPE_CTRL_STS register respectively. Optionally an interrupt can be generated. As such packets have empty (all-zero) data payload, they are dropped inside the MAC and not forwarded to the MRI.

Software can set the SVER and SRSP fields of MAC_FPE_CTRL_STS register to request MAC to transmit Verify and Respond mPackets respectively. Upon successful transmission of these frames, MAC clears the SVER/SRSP bits and sets the TVER & TRSP fields of MAC_FPE_CTRL_STS register. Optionally an interrupt can be generated when these events occur.

11.7.2.6.12.7 Frame Preemption and MMC Counter and Interrupt Registers

The following MMC counters and associated Interrupt registers are instantiated/present in the MAC when Frame Preemption feature is enabled.

Table 11-123. MMC Counters and Associated Interrupt Registers

Frame Assembly Error Counter	Description	Associated MMC Counter
Frame Assembly Error Counter	A 32-bit counter that provides the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value.	MMC_Rx_Packet_Assembly_Err_Cntr
Frame SMD Error Counter	A 32-bit counter that provides the number of received MAC frames rejected due to arriving with an SMD-C when there was no preceding preempted frame.	MMC_Rx_Packet_SMD_Err_Cntr
Frame Assembly OK Counter	A 32-bit counter that provides the number of MAC frames that were successfully reassembled.	MMC_Rx_Packet_Assembly_Ok_Cntr
MAC Rx Fragment Counter	A 32-bit counter that provides the number of additional mPackets received due to preemption.	MMC_Rx_FPE_Fragment_Cntr
MAC Tx Fragment Counter	A 32-bit counter that provides the number of additional mPackets transmitted due to preemption.	MMC_Tx_FPE_Fragment_Cntr
Hold Request Counter	A 32-bit counter that maintains the count of number of times a hold request is given to MAC.	MMC_Tx_Hold_Req_Cntr

11.7.2.6.12.7.1 Additional Registers Associated With MMC Interrupts

Following are the additional registers associated with MMC interrupts for the MMC error counters:

- MMC_FPE_Tx_Interrupt
- MMC_FPE_Tx_Interrupt_Mask
- MMC_Rx_Interrupt
- MMC_FPE_Rx_Interrupt_Mask

11.7.2.6.13 Time-Based Scheduling

Time-based scheduling feature is suitable for traffic whose periodicity and rate are predictable. To improve the quality-of-service of such traffic,

- The transmit DMA fetches the packet from the host memory for transmission at designated time. This helps the software to setup the Transmit descriptors in advance

even before packet is ready/available. It reduces the overhead on the software and avoids constant monitoring of the time and preparing descriptors just in time when the packet is targeted to be transmitted.
- The MAC transmits the packet only at the designated/pre-determined time even if the packets are fetched in advance. This helps in maintaining a constant transmission rate that can be consumed by the receiver station; therefore avoiding congestion and excessive buffering in the network.

The time-based scheduling feature supports fetching and launching an Ethernet packet at (or after) a pre-determined time. The time-based scheduling is supported only in the following modes/configurations:

- Full duplex mode
- Link speed is 100Mbps or higher

The following figure shows the time-based scheduling flow.

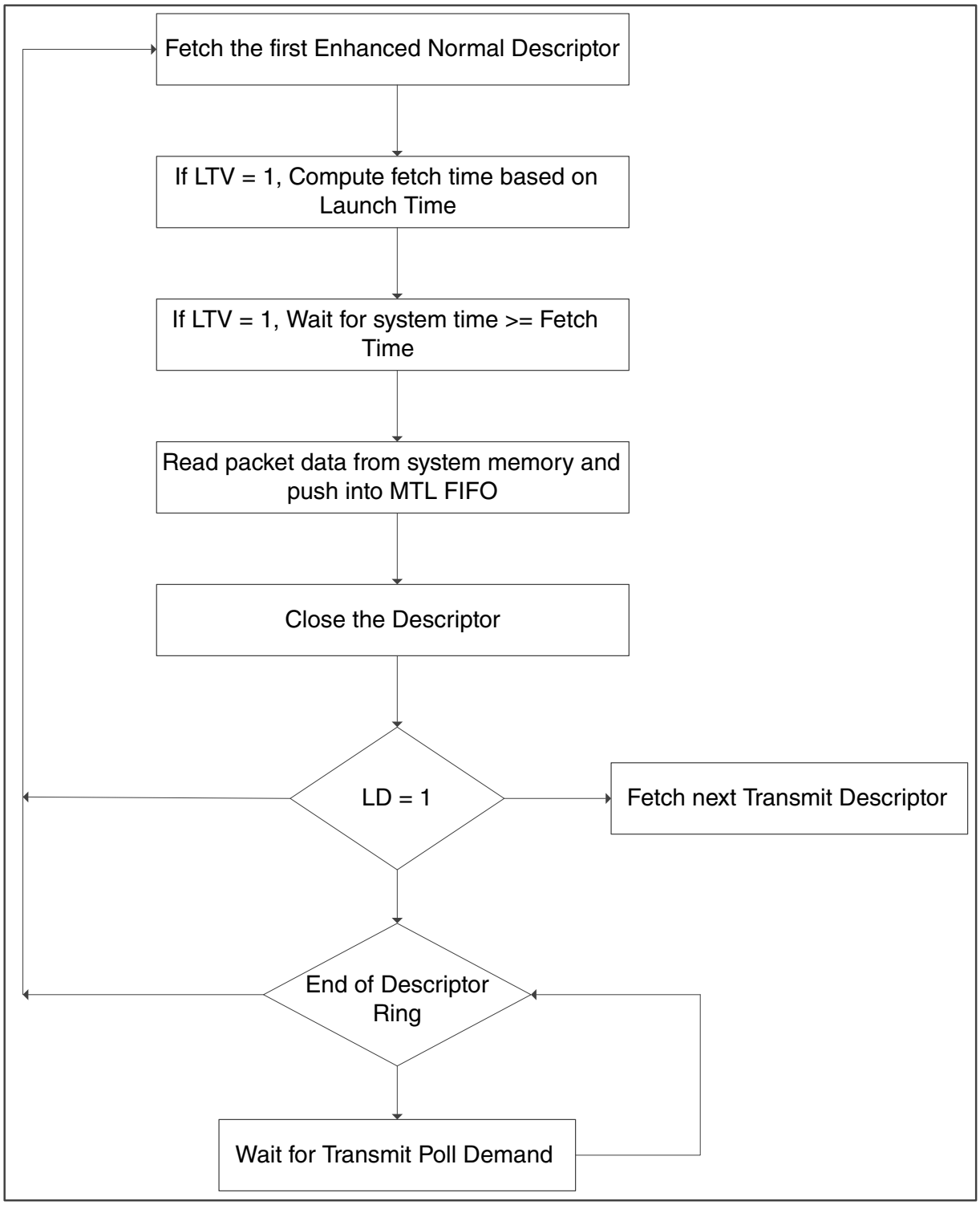


Figure 11-103. Time-Based Scheduling Flow

Following are the definitions specific to the time-based scheduling feature:

11.7.2.6.13.1 Launch Time

Launch time is the time beyond which the MTL can schedule the packet for transmission. The launch time is reset after the transmission of one frame. Following are the two formats of the launch time.

Normal/Absolute Format:

In this format, the launch time is interpreted in the Normal/Absolute format if the ESTM bit of the MTL_TBS_CTRL register is set to 0.

The launch time is a 32-bit value, where most-significant 8-bits represent the time in seconds and the rest 24 bits represent the time in 256 ns. The launch time is compared against the IEEE 1588 based System/PTP Time (bits[39:8]) and rolls over after 256 seconds.

The maximum value of the lower 32-bits of System Time is 999,999,999 decimal (0x3B9AC9FF) and it wraps to 0 when reaching this value (representing a full second). Therefore, the maximum value of the lower 24 bits of the Launch Time (after multiplying by 256) must be 0x3B9AC9.

As the maximum value of Launch time is 256 seconds,

- Launch Time is greater than current System Time when its value is between System Time[39:8] and System Time[39:8] + 128 sec
- Launch Time is less than current System Time when its value is between "System Time[39:8] + 128sec" and "System Time[39:8] + 256 sec", because this is a modulo 256 computation.

EST/Offset Format

In this format, the Launch time is an offset value relative to the time indicated by the Base Time Register (BTR) of the GCL list provided in the GCL Slot Number (GSN). The value in the BTR is always updated to the start time of the current loop of the GCL.

For each packet, the GSN value and the Launch Time value are specified in the Enhanced Transmit Descriptor in the DMA configurations, or as Control Word in the MTL configurations.

The launch time offset is a 32-bit value; the upper 8-bits represent the time in seconds and the rest 24 bits represent the time in 256 ns, which is added to the BTR value corresponding to GCL Slot Number. The value of the Launch Time Offset must be smaller than the value of the Cycle Time (specified in the CTR register that is

implemented when EST is enabled). If the CTR is greater than or equal to 1s, the maximum value of the lower-24 bits of Launch Time offset should be 999,999,999 decimal (0x3B9AC9FF).

In this format, the launch time is a 64-bit value, which is interpreted as, Launch Time = Launch GSN BTR[63:0] + (Launch Time Offset[31:0] << 8), which is compared with the System Time [63:0].

GSN BTR is the Base Time value at which the Launch GSN loop started execution.

GCL Slot Number (GSN) A modulo 16 count of the GCL loop count is implemented and known as GSN. The count is incremented for every new GCL loop; installation of new GCL list does not impact the count. The current GCL slot number can be obtained by reading the CGSN field of the MTL_EST_Status register.

The maximum value of GSN is 15. Therefore GSN values between Current GSN (CGSN field of MTL_EST_Status register) and CGSN+8 represent current or future slots; all other GSN values are interpreted as elapsed slots or past slots. So, for the correct interpretation of time, GSN value must be between CGSN and CGSN+8, for correct interpretation of time.

11.7.2.6.13.2 Launch Expiry Time

Normal/Absolute Mode

In the normal mode, when the LEOV (Launch Expiry Offset Valid bit of MTL_TBS_CTRL register) is set, the Launch Expiry Offset (LEOS field of MTL_TBS_CTRL register) determines the maximum amount of time a frame is eligible for launch, starting from the time the frame becomes eligible for launch.

The Launch Expiry Offset is a 24-bit value defined in 256ns units, with a maximum possible value of 999,999,999 ns (0x3B9AC9FF).

Launch Expiry Time = (Launch Time[39:8] + LEOS[32:8]) * 256ns

The packet with a specific Launch time is considered eligible for transmission when the Launch time is less than the System Time and (if LEOV is set) the System Time is less than the Launch Expiry time.

When the System time is greater than the Launch Expiry time the frame is categorized as expired and will be dropped from the MTL FIFO.

NOTE

- For correct interpretation and meaningful operation, the Fetch, Launch, and Launch Expiry time should never be set

to a value larger than Current System Time + 128 sec; such a value is interpreted as time that has already elapsed.
- In full duplex mode, the frames dropped from the MTL FIFO have Error Summary (Bit 15) and Excessive Deferral (bit 3) of TxStatus set.

EST/Offset Mode

In the EST mode of operation, when the LEOV field of the MTL_TBS_CTRL register is set, the launch expiry GSN offset (LEGOS field of the MTL_TBS_CTRL register) and Launch Expiry Offset (LEOS field of MTL_TBS_CTRL register) determine the maximum amount of time a frame remains eligible for launch, starting from the time the frame becomes eligible for launch.

Launch Expiry Offset = (LEGOS: LEOS)

LEGOS holds the GSN offset (multiples of CTR time) and LEOS holds maximum value of CTR (sub CTR values) in ns.

Launch Expiry offset is a 24-bit value defined in the units of 256 ns, with a maximum possible value of the smaller of 999,999,999ns or CTR-1 ns.

The Launch Expiry GSN is computed as follows:

Launch Expiry GSN = (Launch GSN + LEGOS + CCMA) Where, CCMA is the CTR Carry due to Modulo addition. This value is 1 if ((Launch Time Offset + LEOS) << 8) is equal to or greater than CTR.

When CCMA = 1, Launch Expiry Offset = (Launch Time Offset+ LEOS) - CTR

When CCMA = 0,

Launch Expiry Offset = (Launch Time Offset + LEOS).

Launch Expiry Time = Launch Expiry GSN BTR[63:0] + Launch Expiry Time Offset.

When LEOV is not set, Launch Expiry Time is not checked.

When LEOV is set, and

- the system time is greater than the Launch Expiry time, the frame is dropped from MTL FIFO. The frame is considered as expired.
- the launch time is smaller than the System Time and Launch Expiry Time is greater than system time, the frame is considered eligible for launch.

NOTE

- Max value of LEGOS is 7. This implies that when LEOV is set, the frame has a maximum life time of <8 GCL loop iterations after it becomes eligible for Launch.
- The slot number of the First GCL List executed each time after EST is enabled, is zero.

11.7.2.6.13.3 Fetch Time

The fetch time accounts for all possible delays in the DMA fetch operation and ensures that the frame is present in the MTL FIFO before the launch time.

If FTOV field of DMA_TBS_CTRL register is set, then fetch time can be indicated for each packet. If the FTOV filed is not set, the Fetch Time Offset is not valid and the DMA fetches packets without any time constraints.

11.7.2.6.13.3.1 Normal/Absolute Mode

In the Normal mode fetch time derived/calculated by reducing the time specified in the Fetch Time Offset (FTOS) field of DMA_TBS_CTRL register from the given Launch time.

Normal mode of operation, the Fetch Launch Time is computed as:

Fetch Time[39:8] = (Launch Time[39:8] - FTOS[31:8])

The Fetch time is 32-bits and is compared against System Time[39:8] to determine the Eligible for fetching the frame:

- The Fetch time is defined as "greater" than System Time if the Fetch Time is in the range of "System Time[39:8]" and "System Time[39:8] + 128 sec". The frame is considered as not-eligible for fetch.
- The Fetch time is defined as "smaller" than the System Time if the Fetch time is in the range of "System Time[39:8] + 128 sec" and "System Time[39:8] + 256 sec". The frame is considered as eligible for fetch.

This is a modulo 256 computation.

11.7.2.6.13.3.2 EST/Offset Mode

In the EST/Offset mode, the Fetch GSN Offset (FGOS field of MTL_TBS_CTRL register) provides the Slot Number offset to be deducted from the Launch GSN. In this case the FTOS value should be the smaller of 999,999,999ns or CTR-1 ns.

If (Launch Time Offset >= FTOS):

Fetch Time Offset = ((Launch Time Offset - FTOS) * 256ns)

CBFS(CTR Borrow for Fetch Subtraction) = 0

If (Launch Time Offset < FTOS) Fetch Time Offset = CTR + ((Launch Time Offset - FTOS) * 256ns)

CBFS (CTR Borrow for Fetch Subtraction) = 1

The Fetch GSN is computed as follows:

Fetch GSN = Launch GSN - FGOS - CBFS

Fetch Time = Fetch GSN BTR[63:0] + Fetch Time Offset

The frame is marked eligible for DMA fetch when the fetch time is smaller than the System Time.

11.7.2.6.13.3.3 DMA Operations Sequence

Following is the sequence of operation when FTOV = 1:

1. Fetch the first Enhanced Normal Descriptor (FD is set)
2. If LTV is set and Fetch enabled, compute the Fetch Time based on the Launch Time Wait for the System Time to be greater than Fetch Time
3. Read the Frame (Data) from the host memory and transfer to MTL FIFO
4. Close the Normal Descriptor
5. Fetch the next Normal Descriptor (if the previous Descriptor was not the Last)
6. Repeat steps 4 to 6, until the last descriptor of the frame (LD is set)

After the Last Descriptor of a frame, the next enhanced normal descriptor should be programmed with a new Launch Time and with LTV bit set. Otherwise, the subsequent frames will be processed without any time restrictions.

NOTE

See Programming the Launch Time in Time-Based Scheduling

11.7.2.7 TCP/IP Offloading Features

11.7.2.7.1 Transmit Checksum Offload Engine

The MAC has a Checksum Offload Engine (COE) to support checksum calculation and insertion in the Transmit path, using which, the software can offload the task of checksum insertion to the hardware. In the transmit path MAC calculates the checksum

and inserts it in the Tx packet. This feature helps in reducing the load on the software and can improve the overall throughput of the system. This feature is supported for only Q0 queue.

The checksum offload engine module supports two types of checksum calculation and insertion. The checksum engine can be controlled for each packet by setting the CIC bits (TDES3 Bits[17:16]).

Note: The checksum for TCP, UDP, or ICMP is calculated over a complete packet, and then inserted into its corresponding header field. Because of this requirement, when this function is enabled, the Tx FIFO automatically operates in the store-and-forward mode even if IP is configured for Threshold (cut-through) mode.

11.7.2.7.1.1 IP Header Checksum Engine

In IPv4 datagrams, the integrity of the header fields is indicated by the 16-bit Header Checksum field (the eleventh and twelfth bytes of the IPv4 datagram). The COE detects an IPv4 datagram when the Type field of Ethernet packet has the value 0x0800 and the Version field of IP datagram has the value 0x4. The checksum field of the input packet is ignored during calculation and replaced with the calculated value.

NOTE

IPv6 headers do not have a checksum field. Therefore, the COE does not modify the IPv6 header fields.

The result of this IP header checksum calculation is indicated by the IP Header Error status bit in the Transmit status (TDES3[0]). This status bit is set whenever the values of the Ethernet Type field and the Version field of IP header are not consistent, or when the Ethernet packet does not have enough data, as indicated by the IP header Length field. In other words, this bit is set when an IP header error is asserted under the following circumstances:

- For IPv4 datagrams:
  - The received Ethernet type is 0x0800, but the Version field of IP header is not equal to 0x4.
  - The IPv4 Header Length field indicates a value less than 0x5 (20 bytes).
  - The total packet length is less than the value given in the IPv4 Header Length field.
- For IPv6 datagrams:
  - The Ethernet type is 0x86dd but the IP header Version field is not equal to 0x6.
  - The packet ends before the IPv6 header (40 bytes) or extension header (as given in the corresponding Header Length field in an extension header) is completely received.

11.7.2.7.1.2 TCP/UDP/ICMP Checksum Engine

The TCP/UDP/ICMP Checksum Engine processes the IPv4 or IPv6 header (including extension headers) and determines whether the encapsulated payload is TCP, UDP, or ICMP. The checksum is calculated for the TCP, UDP, or ICMP payload and inserted into its corresponding field in the header. The Tx COE can work in the following two modes:

- The TCP, UDP, or ICMPv6 pseudo-header is not included in the checksum calculation and is assumed to be present in the Checksum field of the input packet. This engine includes the Checksum field in the checksum calculation, and then replaces the Checksum field with the final calculated checksum.
- The engine ignores the Checksum field, includes the TCP, UDP, or ICMPv6 pseudo-header data into the checksum calculation, and overwrites the checksum field with the final calculated value.

NOTE

For ICMP-over-IPv4 packets, the Checksum field in the ICMP packet must always be 16'h0000 in both modes, because pseudo-headers are not defined for such packets. If it does not equal 16’h0000, an incorrect checksum may be inserted into the packet.

The result of this operation is indicated by the Payload Checksum Error status bit in the Transmit Status vector (TDES3[12]). This engine sets the Payload Checksum Error status bit when:

- It detects that the packet has been forwarded to the MAC Transmitter engine in the store-and-forward mode without the end of packet (EOP) being written to the FIFO.
- when the packet ends before the number of bytes indicated by the Payload Length field in the IP Header is received.

When the packet is longer than the indicated payload length, the COE ignores them as stuff bytes and no error is reported. When this engine detects the first type of error, it does not modify the TCP, UDP, or ICMP header. For the second error type, it still inserts the calculated checksum into the corresponding header field. The following table describes the functions supported by Transmit Checksum Offload engine based on the packet type. When the MAC does not insert the checksum, it is indicated as "No" in the table.

NOTE

You should not enable checksum insertion for IPv4 or IPv6 packets that are greater than the frame size constraint specified in Description of Transmit Checksum Offload Engine because

it may result in incorrect checksum insertion or unexpected behavior.

Table 11-124. Transmit Checksum Offload Engine Functions for Different Packet Types

Packet Type	Hardware IP headerchecksum insertion	Hardware TCP/UDPchecksum insertion
Non-IPv4 or IPv6 packet	No	No
IPv4 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in MAC) but less than or equal to the frame size constraint specified in Transmit Checksum Offload Engine.	Yes	Yes
IPv6 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in MAC) but less than or equal to the frame size constraint specified in Transmit Checksum Offload Engine.	Not Applicable	Yes
IPv4 with TCP, UDP, or ICMP	Yes	Yes
IPv4 packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet is an IPv4 fragment	Yes	No
IPv6 packet with the following next header fields in main or extension headers:		
Hop-by-hop options (in IPv6 main header)	Not Applicable	Yes
Hop-by-hop options (in IPv6 extension header)	Not Applicable	No
Destinations options	Not Applicable	Yes
Routing (with segment left 0)	Not Applicable	No
Routing (with segment left > 0)	Not Applicable	No
TCP, UDP, or ICMP	Not Applicable	Yes
Authentication	Not Applicable	No
Any other next header field in main or extension headers	Not Applicable	No
IPv4 packet has TCP header with Options fields	Yes	Yes
IPv4 Tunnels:	#160;	#160;
IPv4 packet in an IPv4 tunnel	Yes (IPv4 tunnel header)	No
IPv6 packet in an IPv4 tunnel	Yes (IPv4 tunnel header)	No
IPv6 Tunnels:	#160;	#160;
IPv4 packet in an IPv6 tunnel	Not applicable	No
IPv6 packet in an IPv6 tunnel	Not applicable	No
IPv4 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Yes	Yes

Table continues on the next page...

Table 11-124. Transmit Checksum Offload Engine Functions for Different Packet Types (continued)

Packet Type	Hardware IP headerchecksum insertion	Hardware TCP/UDPchecksum insertion
IPv6 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Not applicable	Yes
IPv4 frames with security features (such as encapsulated security payload)	Yes	No
IPv6 frames with security features (such as encapsulated security payload)	Not applicable	No

11.7.2.7.2 Receive Checksum Offload Engine

IP provides the Checksum Offload Engine that is used to detect any error in an IPv4 or IPv6 packet in the receive path. The MAC verifies the checksum field of the received packet with the internally calculated checksum and provides the status.

The Receive Checksum Offload engine is used to detect errors in IP packets by calculating the header checksum and further matching it with the received header checksum. This engine also identifies a TCP, UDP, or ICMP payload in received IP packets and calculates the checksum of such payloads appropriately.

Here, the IPv4 or IPv6 packets in the received Ethernet packets are detected and processed for data integrity. The MAC receiver identifies IPv4 or IPv6 packets by checking for value 0x0800 or 0x86DD, respectively, in the Type field of the received Ethernet packet. This identification is applicable to single VLAN-tagged packets. It is also applicable to double VLAN-tagged packets when the Enable Double VLAN Processing option is selected and the EDVLP bit of the MAC_VLAN_Tag register is set.

The Rx COE calculates the IPv4 header checksums and checks that they match the received IPv4 header checksums. The result of this operation (pass or fail) is given to the RFC module for insertion into the receive status word. The IP Header Error bit is set for any mismatch between the indicated payload type (Ethernet Type field) and the IP header version, or when the received packet does not have enough bytes, as indicated by the Length field of the IPv4 header (or when fewer than 20 bytes are available in an IPv4 or IPv6 header).

This engine also identifies a TCP, UDP, or ICMP payload in the received IP datagrams (IPv4 or IPv6) and calculates the checksum of such payloads properly, as defined in the TCP, UDP, or ICMP specifications. This engine includes the TCP, UDP, or ICMPv6 pseudo-header bytes for checksum calculation and checks whether the received checksum field matches the calculated value. The result of this operation is given as a Payload

Checksum Error bit in the receive status word. This status bit is also set if the length of the TCP, UDP, or ICMP payload does not match the expected payload length given in the IP header.

Following table describes the functions supported by the Rx COE based on the packet type. When the payload of an IP packet is not processed (indicated as "No" in the table), the information (whether the checksum engine is bypassed or not) is given in the receive status.

NOTE

The MAC does not append any payload checksum bytes to the received Ethernet packets.

Table 11-125. Receive Checksum Offload Engine Functions for Different Packet Types

Packet type	Hardware IP header checksum checking	Hardware TCP/UDP/ICMP checksum checking
Non-IPv4 or IPv6	No	No
IPv4 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in the MAC)	Yes	Yes
IPv6 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in the MAC)	Not Applicable	Yes
IPv4 with TCP, UDP, or ICMP	Yes	Yes
IPv4 header's protocol field contains a protocol other than TCP, UDP, or ICMP	Yes	No
IPv4 packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet is an IPv4 fragment	Yes	No
IPv6 packet with the following next header fields in main or extension headers: • Hop-by-hop options (in IPv6 main header) • Hop-by-hop options (in IPv6 extension header) • Destinations options • Routing (with segment left 0) • Routing (with segment left > 0) • TCP, UDP, or ICMP • Any other next header field in main or extension headers	• Not Applicable • Not Applicable • Not Applicable • Not Applicable • Not Applicable • Not Applicable • Not Applicable	• Yes • No • Yes • Yes • No • Yes • No
IPv4 packet has TCP header with Options fields	Yes	Yes

Table continues on the next page...

Table 11-125. Receive Checksum Offload Engine Functions for Different Packet Types (continued)

Packet type	Hardware IP header checksum checking	Hardware TCP/UDP/ICMP checksum checking
IPv4 Tunnels: • IPv4 packet in an IPv4 tunnel • IPv6 packet in an IPv4 tunnel	• Yes (IPv4 tunnel header) • Yes (IPv4 tunnel header)	• No • No
IPv4 Tunnels: • IPv4 packet in an IPv6 tunnel • IPv6 packet in an IPv6 tunnel	• Not Applicable • Not Applicable	• No • No
IPv4 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled)	Yes	Yes
IPv6 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled)	Not Applicable	Yes
IPv4 frames with security features (such as encapsulated security payload)	Yes	No
IPv6 frames with security features (such as encapsulated security payload)	Not Applicable	No

Set the IPC bit of the MAC_Configuration register for enabling Receive Checksum offload.

11.7.2.8 MAC Management Counters

IP supports storing the statistics about the received and transmitted packets in registers that are accessible through the application/software. The MMC module maintains a set of registers for gathering statistics on the received and transmitted packets. The MMC counters are free running.

For MMC register details, see registers "MMC_Control" to "MMC_Rx_FPE_Fragment_Cntr" in ENET_QOS register descriptions.

11.7.2.9 Flow Control

11.7.2.9.1 Transmit Flow Control

The Transmit Flow Control involves transmitting Pause packets in full-duplex mode and backpressure in half-duplex mode to control the flow of packets from the remote end.

11.7.2.9.1.1 Flow control in Full duplex mode

IP supports the IEEE802.3x Pause Packets.

Pause packet structure is shown in the following table.

Table 11-126. Pause Packet Fields

Field	Description
DA	Contains the special multicast address
SA	Contains the MAC address 0
Type	Contains 8808
MAC Control opcode	Contains 0001 for IEEE 802.3x Pause Control packets; 0101 for PFC packets
PT	Contains Pause time specified in the PT field of the MAC_Q#_Tx_Flow_Ctrl register

11.7.2.9.1.2 Pause Packet Control

When the FCB bit is set, the MAC generates and transmits a single Pause packet. If the FCB bit is set again after the Pause packet transmission is complete, the MAC sends another Pause packet irrespective of whether the pause time is complete or not. To extend the pause or terminate the pause prior to the time specified in the previously-transmitted Pause packet, the application should program the Pause Time register with appropriate value and then again set the FCB bit.

11.7.2.9.1.3 Flow Control in Half-Duplex Mode

In half-duplex mode, the MAC uses the deferral mechanism for the flow control (backpressure). When the application requests to stop receiving packets, the MAC sends a JAM pattern of 32 bytes when it senses a packet reception, provided the transmit flow control is enabled. This results in a collision and the remote station backs off. If the application requests a packet to be transmitted, it is scheduled and transmitted even when the backpressure is activated. If the backpressure is kept activated for a long time (and more than 16 consecutive collision events occur), the remote stations abort the transmission because of excessive collisions.

Following table describes the flow control in the Tx path for Queue 0 based on the setting of the following bits:

- EHFC bit of MTL_RxQ0_Operation_Mode register

- TFE bit of MAC_Q0_Tx_Flow_Ctrl register
- DM bit of MAC_Configuration register

Flow control is similar for all queues.

Table 11-127. Tx MAC Flow Control

EHFC	TFE	DM	Description
x	0	x	The MAC transmitter does not perform the flow control or backpressure operation.
0	1	0	The MAC transmitter performs back-pressure when Bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set or the sideband signal sbd_flowctrl_i is 1.
1	1	0	The MAC transmitter performs back-pressure when Bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set or the sideband signal sbd_flowctrl_i is 1. In addition, the MAC Tx performs back-pressure when Rx Queue level crosses the threshold set by Bits[10:8] of MTL_RxQ0_Operation_Mode register.
0	1	1	The MAC transmitter sends the Pause packet when Bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set or the sideband signal sbd_flowctrl_i is 1.
1	1	1	The MAC transmitter sends the Pause packet when Bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set or the sideband signal sbd_flowctrl_i is 1. In addition, the MAC Tx sends a Pause packet when Rx Queue level crosses the threshold set by Bits[10:8] of MTL_RxQ0_Operation_Mode register.

11.7.2.9.1.4 Enabling Transmit Flow Control

To independently enable Transmit Flow Control for each Tx queue, set the TFE bit in the MAC_Q#_Tx_Flow_Ctrl register.

11.7.2.9.2 Receive Flow Control

In the Receive path, the Flow Control is functional only in the full-duplex mode. If any Pause packet is received in the half-duplex mode, the packet is considered as a normal control packet.

NOTE

Receive pause packets should have a frame size of 64 bytes.

The Receive Flow Control is implemented by the MAC based on the bit value of the respective register, and the destination address and different fields of the received packet.

Following table describes the flow control in the Rx path based on the setting of the following bits:

- RFE bit of MAC_RX_FLOW_CTRL register
- DM bit of MAC_CONFIGURATION register

Table 11-128. Rx MAC Flow Control

RFE	DM	Description
0	x	The MAC receiver does not detect the received Pause packets.
1	0	The MAC receiver does not detect the received Pause packets but recognizes such packets as Control packets.
1	1	The MAC receiver detects or processes the Pause packets and responds to such packets by stopping the MAC transmitter.

11.7.2.9.2.1 Enabling Receive Flow Control

To enable Pause Flow Control, set the RFE bit in the MAC_Rx_Flow_Ctrl register.

11.7.2.10 Loopback Mode

The MAC supports Loopback of transmitted packets to its receiver.

The following are some guidelines for using the loopback mode:

- Enable loopback only with the full-duplex mode. In half-duplex mode, the carrier sense signal (crs) or collision (col) signal inputs get sampled which may result into issues such as packet dropping.

- If the loopback mode is enabled without connecting a PHY chip (for example, in FPGA setup), you should externally generate the Tx and Rx clocks and provide these clocks to the MAC.
- Do not loop back big packets. Big packets (>1522 bytes) may get corrupted in the loopback FIFO.

To enable this feature, program the LM bit of the MAC_Configuration register.

You can enable loopback for all PHY interfaces. The data is always looped back through internal asynchronous FIFO on to the internal Receive GMII interface, irrespective of which PHY interface is selected. The loopback data is also passed through the corresponding transmit PHY interface block, on to the Ethernet line.

11.7.2.11 Descriptors

11.7.2.11.1 Overview

The DMA in the Ethernet subsystem transfers data based on a linked list of descriptors. The application creates the descriptors in the system memory. The module supports the following two types of descriptors:

- Normal Descriptor: Normal descriptors are used for packet data and to provide control information applicable to the packets to be transmitted.
- Context Descriptor: Context descriptors are used to provide control information applicable to the packet to be transmitted.

Each normal descriptor contains two buffers and two address pointers. These buffers enable the adapter port to be compatible with various types of memory management schemes.

NOTE

There is no limit for the number of descriptors that can be used for a single packet.

11.7.2.11.2 Descriptor Structure

The module supports the ring structure for DMA descriptor as shown in the following figure.

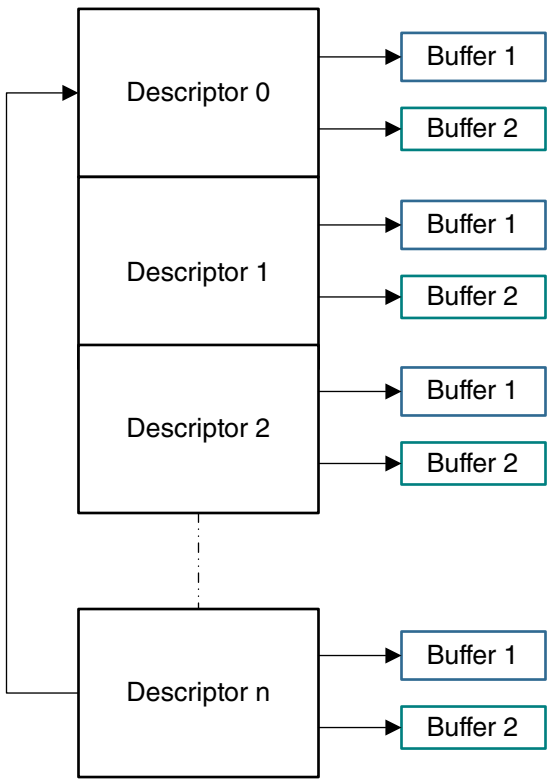


Figure 11-104. Descriptor Ring Structure

In Ring structure, descriptors are separated by the Word, DWord, or LWord number programmed in the DSL field of the DMA_CH#_Control register. The application needs to program the total ring length, that is, the total number of descriptors in ring span in the following registers of a DMA channel:

- Transmit Descriptor Ring Length Register (DMA_CH#_TxDesc_Ring_Length)
- Receive Descriptor Ring Length Register (DMA_CH#_RxDesc_Ring_Length)

The Descriptor Tail Pointer Register contains the pointer to the descriptor address (N). The base address and the current descriptor pointer decide the address of the current descriptor that the DMA can process. The descriptors up to one location less than the one indicated by the descriptor tail pointer (N – 1) are owned by the DMA. The DMA continues to process the descriptors until the following condition occurs:

Current Descriptor Pointer == Descriptor Tail Pointer;

The DMA goes into the Suspend mode when this condition occurs. The application must perform a write to the Descriptor Tail pointer register and update the tail pointer so that the following condition is true:

Current Descriptor Pointer < Descriptor Tail Pointer;

The DMA automatically wraps around the base address when the end of ring is reached, as shown in the following figure.

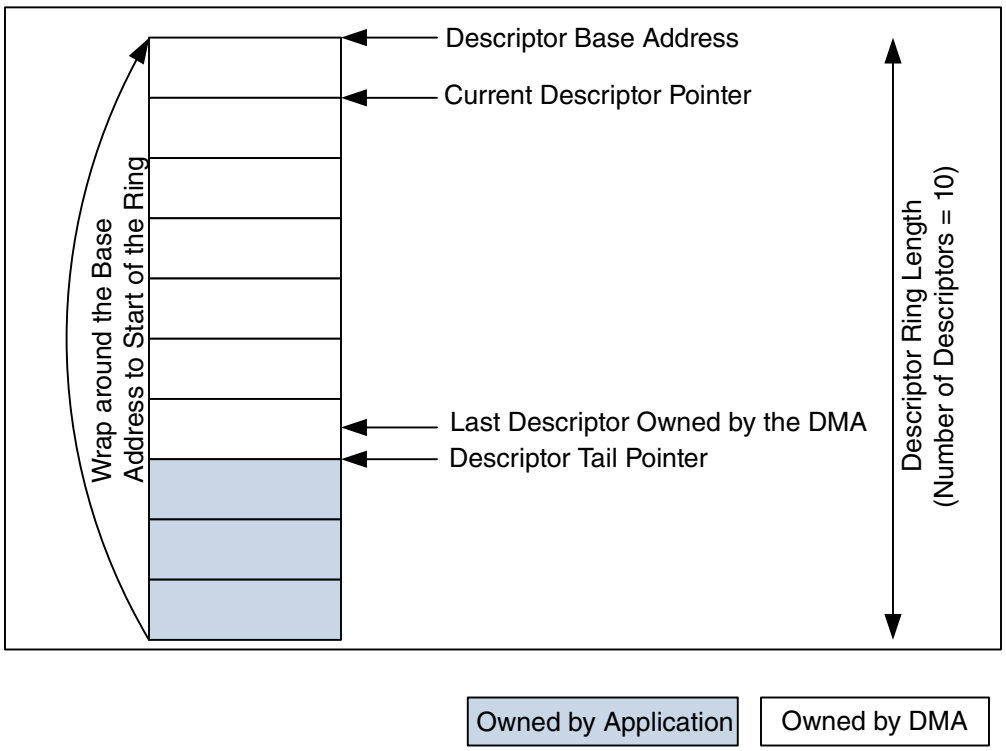


Figure 11-105. DMA Descriptor Ring

For descriptors owned by the application, the OWN bit of DES3 is reset to 0. For descriptors owned by the DMA, the OWN bit is set to 1. If the application has only one descriptor in the beginning, the application sets the last descriptor address (tail pointer) to Descriptor Base Address + 1. The DMA processes the first descriptor and then waits for the application to advance the tail pointer.

11.7.2.11.3 Transmit Descriptor

The DMA in the module requires at least one descriptor for a transmit packet. In addition to two buffers, two byte-count buffers, and two address pointers, the transmit descriptor has control fields which can be used to control the MAC operation on per-transmit packet basis. The Transmit Normal descriptor has two formats: Read format and Write-Back format

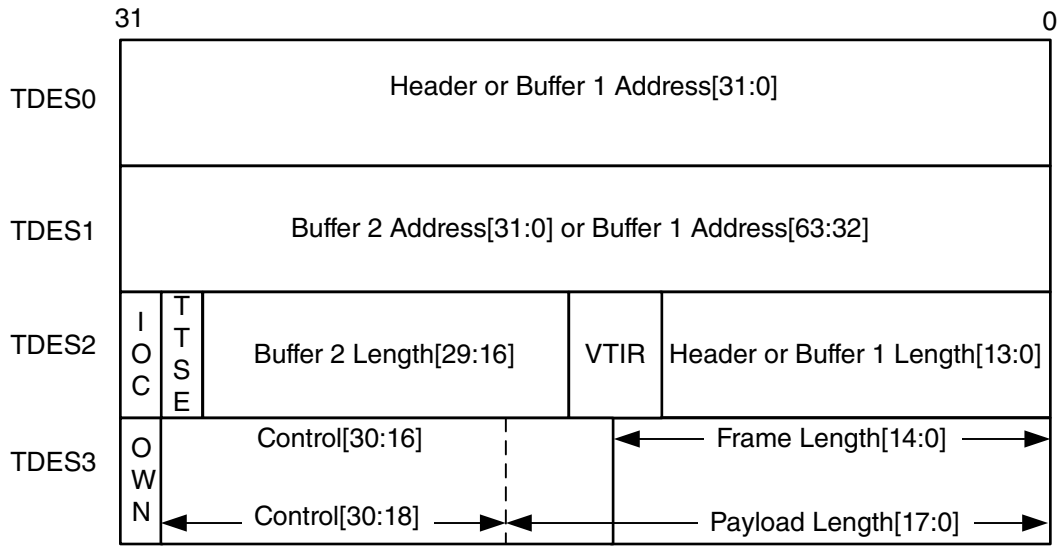
NOTE

TSO Split header is not supported.

11.7.2.11.3.1 Transmit Normal Descriptor (Read Format)

Following figure shows the Read Format for a Transmit normal descriptor. Table: 'TDES0 Normal Descriptor (Read Format)' through Table: 'TDES3 Normal Descriptor (Read Format)' describe the read format for the Transmit Normal Descriptors: TDES0, TDES1, TDES2, and TDES3.

Figure 11-106. Transmit Descriptor Read Format



11.7.2.11.3.1.1 TDES0 Normal Descriptor (Read Format)

Table 11-129. TDES0 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF1AP	Buffer 1 Address Pointer or TSO Header Address Pointer These bits indicate the physical address of Buffer 1. These bits indicate the TSO Header Address pointer when the following bits are set: • TSE bit of TDES3 • FD bit of TDES3

11.7.2.11.3.1.2 TDES1 Normal Descriptor (Read Format)

Table 11-130. TDES1 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF2AP	Buffer 2 or Buffer 1 Address Pointer

Table 11-130. TDES1 Normal Descriptor (Read Format)

Bit	Name	Description
		This bit indicates the physical address of Buffer 2 when a descriptor ring structure is used. There is no limitation for the buffer address alignment. In 40- or 48-bit addressing mode, these bits indicate the most-significant 8- or 16- bits of the Buffer 1 Address Pointer.

11.7.2.11.3.1.3 TDES2 Normal Descriptor (Read Format)

Table 11-131. TDES2 Normal Descriptor (Read Format)

31	30	29-16	15:14	13:0
IOC	TTSE/ TMWD	B2L	VTIR	HL or B1L

Table 11-132. TDES2 Normal Descriptor (Read Format)

Bits	Name	Description
31	IOC	Interrupt on Completion This bit controls the setting of TI and ETI status bits in the DMA_CH#_Status register. When ETIC = 1 and TDES2[LD] = 0, this bit sets the ETI bit. When TDES3[LD] = 1, this bit sets the TI status bit.
30	TTSE/TMWD	Transmit Timestamp Enable or External TSO Memory Write Enable This bit enables the IEEE1588 time stamping for Transmit packet referenced by the descriptor, if TSE bit is not set. If TSE bit is set and external TSO memory is enabled, setting this bit disables external TSO memory writing for this packet.
29:16	B2L	Buffer 2 Length The driver sets this field. When set, this field indicates Buffer 2 length.
15:14	VTIR	VLAN Tag Insertion or Replacement These bits request the MAC to perform VLAN tagging or untagging before transmitting the packets. The application must set the CRC Pad Control bits appropriately when VLAN Tag Insertion, Replacement, or Deletion is enabled for the packet. The following list describes the values of these bits:

Table continues on the next page...

Table 11-132. TDES2 Normal Descriptor (Read Format) (continued)

Bits	Name	Description
		• 2'b00: Do not add a VLAN tag. • 2'b01: Remove the VLAN tag from the packets before transmission. This option should be used only with the VLAN packets. • 2'b10: Insert a VLAN tag with the tag value programmed in the MAC_VLAN_Incl register or context descriptor. • 2'b11: Replace the VLAN tag in packets with the tag value programmed in the MAC_VLAN_Incl register or context descriptor. This option should be used only with the VLAN packets. These bits are valid when the Enable SA and VLAN Insertion on Tx option is selected while configuring the core.
13:0	HL or B1L	Header Length or Buffer 1 Length For Header length only bits [9:0] are taken. The size 13:0 is applicable only when interpreting buffer 1 length. If the TCP Segmentation Offload feature is enabled through the TSE bit of TDES3, this field is equal to the header length. When the TSE bit is set in TDES3, the header length includes the length in bytes from Ethernet Source address till the end of the TCP header. The maximum header length supported for TSO feature is 1023 bytes. The maximum header length supported for TSO feature is 1023 bytes. If the TCP Segmentation Offload feature is not enabled, this field is equal to Buffer 1 length.

11.7.2.11.3.1.4 TDES3 Normal Descriptor (Read Format)

Table 11-133. TDES2 Normal Descriptor (Read Format)

31	30	29	28	27:26	25:23	22:19	18	17:16	15	14:0
OWN	CTXT	FD	LD	CPC	SAIC	SLOTNUM or THL	TSE	CIC/TPL	TPL	FL/TPL

Table 11-134. TDES3 Normal Descriptor (Read Format)

Bits	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit after it completes the transfer of data given in the associated buffer(s).
30	CTXT	Context Type This bit should be set to 1'b0 for normal descriptor.
29	FD	First Descriptor When this bit is set, it indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor When this bit is set, it indicates that the buffer contains the last segment of the packet. When this bit is set, the B1L or B2L field should have a non-zero value.
27:26	CPC	CRC Pad Control This field controls the CRC and Pad Insertion for Tx packet. This field is valid only when the first descriptor bit (TDES3[29]) is set. The following list describes the values of Bits[27:26]: • 2'b00: CRC and Pad Insertion The MAC appends the cyclic redundancy check (CRC) at the end of the transmitted packet of length greater than or equal to 60 bytes. The MAC automatically appends padding and CRC to a packet with length less than 60 bytes. • 2'b01: CRC Insertion (Disable Pad Insertion) The MAC appends the CRC at the end of the transmitted packet but it does not append padding. The application should ensure that the padding bytes are present in the packet being transferred from the Transmit Buffer, that is, the packet being transferred from the Transmit Buffer is of length greater than or equal to 60 bytes. • 2'b10: Disable CRC Insertion

Table continues on the next page...

Table 11-134. TDES3 Normal Descriptor (Read Format) (continued)

Bits	Name	Description
		The MAC does not append the CRC at the end of the transmitted packet. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer. • 2'b11: CRC Replacement The MAC replaces the last four bytes of the transmitted packet with recalculated CRC bytes. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer. This field is valid only for the first descriptor. Note: When the TSE bit is set, the MAC ignores this field because the CRC and pad insertion is always done for segmentation.
25:23	SAIC	SA Insertion Control These bits request the MAC to add or replace the Source Address field in the Ethernet packet with the value given in the MAC Address 0 or 1 register. The application must set the CRC Pad Control bits appropriately when SA Insertion Control is enabled for the packet. Bit 25 specifies the MAC Address Register (1 or 0) value that is used for Source Address insertion or replacement. The following list describes the values of Bits[24:23]: • 2'b00: Do not include the source address • 2'b01: Include the source address. For reliable transmission, the application must provide frames without source addresses. • 2'b10: Replace the source address. For reliable transmission, the application must provide frames with source addresses. • 2'b11: Reserved

Table continues on the next page...

Table 11-134. TDES3 Normal Descriptor (Read Format) (continued)

Bits	Name	Description
		These bits are valid in the EQOS-DMA, EQOS-AXI, and EQOS-AHB configurations when the Enable SA and VLAN Insertion on Tx option is selected while configuring the core and when the First Segment control bit (TDES3 [29]) is set. This field is valid only for the first descriptor.
22:19	SLOTNUMor THL	SLOTNUM: Slot Number Control Bits in AV Mode These bits indicate the slot interval in which the data should be fetched from the corresponding buffers addressed by TDES0 or TDES1. When the Transmit descriptor is fetched, the DMA compares the slot number value in this field with the slot interval maintained in the RSN field DMA_CH#_Slot_Function_Control_Status. It fetches the data from the buffers only if a value matches. These bits are valid only for the AV channels. THL: TCP/UDP Header Length If the TSE bit is set, this field contains the length of the TCP/UDP header. The minimum value of this field must be 5 for TCP header. The value must be equal to 2 for UDP header. This field is valid only for the first descriptor.
18	TSE	TSO Split header is not supported. The value of this bit is always zero.
17:16	CIC/TPL	Checksum Insertion Control or TCP Payload Length These bits control the checksum calculation and insertion. The following list describes the bit encoding: • 2'b00: Checksum Insertion Disabled. • 2'b01: Only IP header checksum calculation and insertion are enabled.

Table continues on the next page...

Table 11-134. TDES3 Normal Descriptor (Read Format) (continued)

Bits	Name	Description
		• 2'b10: IP header checksum and payload checksum calculation and insertion are enabled, but pseudo-header checksum is not calculated in hardware. • 2'b11: IP Header checksum and payload checksum calculation and insertion are enabled, and pseudo-header checksum is calculated in hardware. This field is valid when the Enable Transmit TCP/IP Checksum Offload option is selected and the TSE bit is reset. When the TSE bit is set, this field contains the upper bits [17:16] of the TCP Payload (or IP Payload for UDP fragmentation). This allows the TCP/UDP packet length field to be spanned across TDES3[17:0] to provide 256 KB packet length support. This field is valid only for the first descriptor.
15	TPL	Reserved or TCP Payload Length When the TSE bit is reset, this bit is reserved. When the TSE bit is set, this is Bit 15 of the TCP payload length [17:0]. This field is valid only when the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected while configuring the core.
14:0	FL/TPL	Frame Length or TCP Payload Length This field is equal to the length of the packet to be transmitted in bytes. When the TSE bit is not set, this field is equal to the total length of the packet to be transmitted: Ethernet Header Length + TCP /IP Header Length – Preamble Length – SFD Length + Ethernet Payload Length When the TSE bit is set, this field is equal to the lower 15 bits of the TCP payload length in case of segmentation and IP payload in case of UDP fragmentation.

Table 11-134. TDES3 Normal Descriptor (Read Format)

Bits	Name	Description
		In case of segmentation, this length does not include Ethernet header or TCP/UDP/IP header length. In case of fragmentation, this length does not include Ethernet header and IP header. When DWRR/WFQ algorithm is NOT enabled, value written into this field is not used when TSE = 0.

11.7.2.11.3.2 Transmit Normal Descriptor (Write-Back Format)

The write-back format of the Transmit Descriptor includes timestamp low, timestamp high, OWN, and Status bits.

The write-back format is applicable only for the last descriptor of the corresponding packet. The LD bit (TDES3[28]) is set in the descriptor where the DMA writes back the status and timestamp information for the corresponding Transmit packet.

Following figure illustrates the write-back format of the Transmit Descriptor.

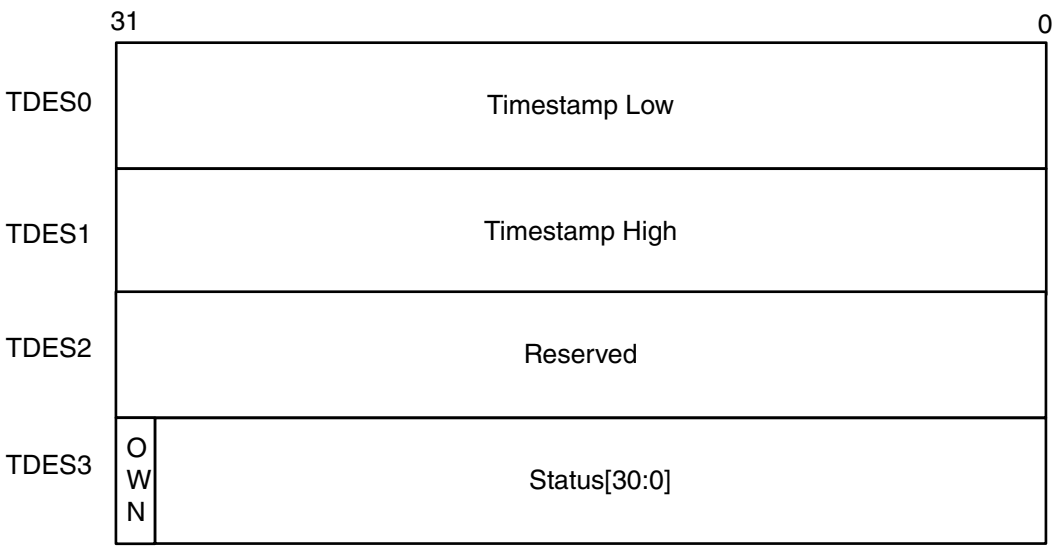


Figure 11-107. Transmit Descriptor Write-Back Format

TDES0 Normal Descriptor (Write-Back Format)

As described in following table, this format is only applicable to the last descriptor of a packet.

Table 11-135. TDES0 Normal Descriptor (Write-Back Format)TDES0 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low The DMA updates this field with least significant 32 bits of the timestamp captured for the corresponding Transmit packet. The DMA writes the timestamp only if TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Descriptor bit (LD) in the descriptor is set and the Timestamp status (TTSS) bit is set.

TDES1 Normal Descriptor (Write-Back Format)

As mentioned in the following table, this format is only applicable to the last descriptor of a packet.

Table 11-136. TDES1 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:0	TTSH	Transmit Packet Timestamp High The DMA updates this field with the most significant 32 bits of the timestamp captured for corresponding transmit packet. The DMA writes the timestamp only if the TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Descriptor bit (LD) in the descriptor is set and Timestamp status (TTSS) bit is set.

TDES2 Normal Descriptor (Write-Back Format)

This format is applicable only to the last descriptor of a packet.

Table 11-137. TDES2 Normal Descriptor (Write-Back Format)

Bit	Description
31:0	Reserved

TDES3 Normal Descriptor (Write-Back Format)

This format is applicable only to the last descriptor of a packet.

Table 11-138. TDES3 Normal Descriptor (Write-Back Format)

31	30	29	28	27:24	23	22:18	17	16	15	14	13	12	11	10	9	8	7:4	3	2	1	0
OWN	CTXT	FD	LD	Rsvd	DE	Rsvd	TTSS	EUE	ES	JT	FF	PCE	LoC	NC	LC	EC	CC	ED	UF	DB	IHE

Table 11-139. TDES3 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the the module DMA owns the descriptor. The DMA clears this bit when it completes the packet transmission. After the write-back is complete, this bit is set to 'b0.
30	CTXT	Context Type This bit should be set to 'b0 for Normal descriptor.
29	FD	First Descriptor This bit indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor This bit is set 'b1 for last descriptor of a packet. The DMA writes the status fields only in the last descriptor of the packet.
27:24	Rsvd	Reserved
23	DE	Descriptor Error When this bit is set, it indicates that the descriptor content is incorrect. The DMA sets this bit during write-back while closing the descriptor. Descriptor Errors can be: • Incorrect sequence from the context descriptor. For example, a location after the first descriptor for a packet. • All 1s • CTXT, LD, and FD bits set to 1. Note 1: When Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA will set the TI bit in the DMA_CH#_Status register

Table continues on the next page...

Table 11-139. TDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be considered as the First Descriptor (even if FD bit is not set) and partial packet is sent.
22:18	Rsvd	Reserved
17	TTSS	Tx Timestamp Status This status bit indicates that a timestamp has been captured for the corresponding transmit packet. When this bit is set, TDES0 and TDES1 have timestamp values that were captured for the Transmit packet. This field is valid only when the Last Segment control bit (TDES3 [28]) in a descriptor is set. This bit is valid only when IEEE1588 timestamping feature is enabled; otherwise, it is reserved.
16	EUE	ECC Uncorrectable Error Status Indicates the ECC uncorrectable error in the TSO memory. Note: Uncorrectable error in Transmit FIFO memory is reported with (Bit 13) FF = 1. This is because, all such packets are flushed by the module.
15	ES	Error Summary This bit indicates the logical OR of the following bits: • TDES3[0]: IP Header Error • TDES3[14]: Jabber Timeout • TDES3[13]: Packet Flush • TDES3[12]: Payload Checksum Error • TDES3[11]: Loss of Carrier • TDES3[10]: No Carrier • TDES3[9]: Late Collision • TDES3[8]: Excessive Collision • TDES3[3]: Excessive Deferral • TDES3[2]: Underflow Error This bit is also set when EUE (bit 16) is set.
14	JT	Jabber Timeout

Table continues on the next page...

Table 11-139. TDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		This bit indicates that the MAC transmitter has experienced a jabber time-out. This bit is set only when the JD bit of the MAC_Configuration register is not set.
13	FF	Packet Flushed This bit indicates that the DMA or MTL flushed the packet because of a software flush command given by the CPU.
12	PCE	Payload Checksum Error This bit indicates that the Checksum Offload engine had a failure and did not insert any checksum into the encapsulated TCP, UDP, or ICMP payload. This failure can be either because of insufficient bytes, as indicated by the Payload Length field of the IP Header or the MTL starting to forward the packet to the MAC transmitter in Store-and-Forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet packet being transmitted to avoid deadlock, the MTL starts forwarding the packet when the FIFO is full, even in the store-and-forward mode. This error can also occur when Bus Error is detected during packet transfer. When the Full Checksum Offload engine is not enabled, this bit is reserved.
11	LoC	Loss of Carrier This bit indicates that Loss of Carrier occurred during packet transmission (that is, the gmii_crs_i signal was inactive for one or more transmit clock periods during packet transmission). This is valid only for the packets transmitted without collision and when the MAC operates in the half-duplex mode.
10	NC	No Carrier This bit indicates that the carrier sense signal form the PHY was not asserted during transmission.
9	LC	Late Collision

Table continues on the next page...

Table 11-139. TDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		This bit indicates that packet transmission was aborted because a collision occurred after the collision window (512 byte times including Preamble and Carrier Extension in GMII mode). This bit is not valid if Underflow Error is set.
8	EC	Excessive Collision This bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current packet. If the DR bit is set in the MAC_Configuration register, this bit is set after first collision and the transmission of the packet is aborted.
7:4	CC	Collision Count This 4-bit counter value indicates the number of collisions occurred before the packet was transmitted. The count is not valid when the EC bit is set.
3	ED	Excessive Deferral This bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1000 Mbps mode or Jumbo Packet enabled mode) if DC bit is set in the MAC_Configuration register. When TBS is enabled in full duplex mode and this bit is set, it indicates that the frame has been dropped after the expiry time has reached.
2	UF	Underflow Error This bit indicates that the MAC aborted the packet because the data arrived late from the system memory. The underflow error can occur because of either of the following conditions: • The DMA encountered an empty Transmit Buffer while transmitting the packet • The application filled the MTL Tx FIFO slower than the MAC transmit rate The transmission process enters the suspended state and sets the underflow bit corresponding to a queue in the MTL_Interrupt_Status register.
1	DB	Deferred Bit

Table continues on the next page...

Table 11-139. TDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		This bit indicates that the MAC deferred before transmitting because of presence of carrier. This bit is valid only in the half-duplex mode.
0	IHE	IP Header Error When IP Header Error is set, this bit indicates that the Checksum Offload engine detected an IP header error. This bit is valid only when Tx Checksum Offload is enabled. Otherwise, it is reserved. If COE detects an IP header error, it still inserts an IPv4 header checksum if the Ethernet Type field indicates an IPv4 payload. In full duplex mode, when EST/Qbv is enabled and this bit is set, it indicates the frame drop status due to Frame Size error or Schedule Error.

11.7.2.11.3.3 Transmit Context Descriptor

The context descriptor is used to provide the timestamps for one-step timestamp correction, VLAN Tag ID for VLAN insertion feature.

The Transmit Context descriptor can be provided any time before a packet descriptor. The context is valid for the current packet and subsequent packets. The context descriptor is used to provide the timestamps for one-step timestamp correction and VLAN Tag ID for VLAN insertion feature. Write back is done on a context descriptor only to reset the OWN bit.

NOTE

The VLAN Tag IDs and MSS values, provided by the application in a context descriptor with their corresponding Valid bits set, are stored internally by the DMA. When the outer or inner VLAN tag is provided with the Valid bit set, the DMA always passes the last valid VLAN tag to the MTL. The application cannot invalidate the valid VLAN tag stored by the DMA. The VLAN tag is inserted or replaced based on the control inputs provided for the packet.

The Inner VLAN Tag Control input is used only for the next packet that immediately follows the context descriptor. The application must provide a context descriptor before the normal descriptor of each packet for which the DMA should use the inner VLAN Tag control input.

The following figure shows the format of the Transmit Context descriptor.

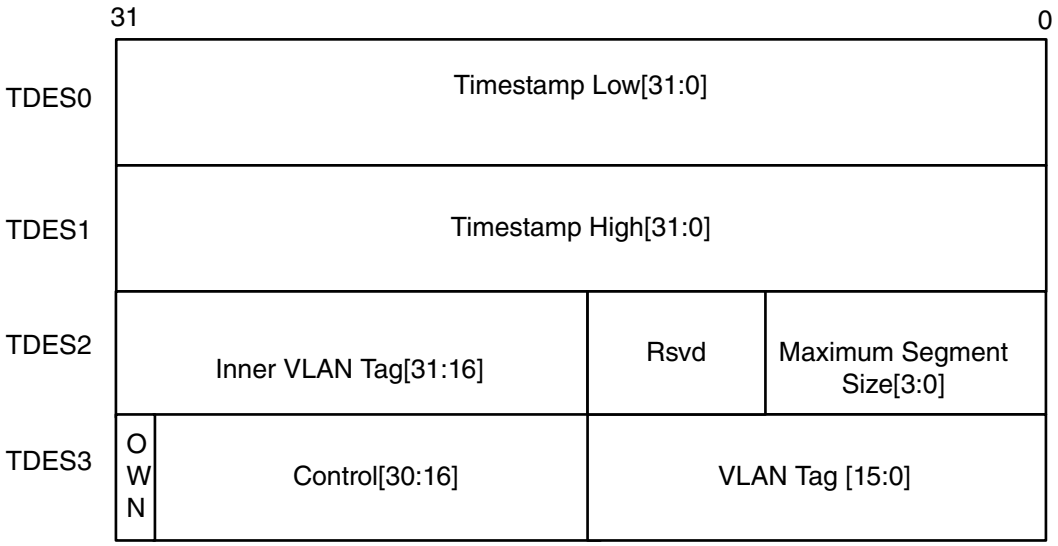


Figure 11-108. Transmit Context Descriptor Format

11.7.2.11.3.3.1 TDES0 Context Descriptor

Following table describes the format of the TDES0 Context descriptor.

Table 11-140. TDES0 Context Descriptor

Bit	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low For one-step correction, the driver can provide the lower 32 bits of timestamp in this descriptor word. The DMA uses this value as the low word for doing one-step timestamp correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

11.7.2.11.3.3.2 TDES1 Context Descriptor

Following table describes the format of the TDES1 Context descriptor.

Table 11-141. TDES1 Context Descriptor

Bit	Name	Description
31:0	TTSH	Transmit Packet Timestamp High For one-step correction, the driver can provide the upper 32 bits of timestamp in this descriptor. The DMA uses this value as the high word for doing one-step timestamp correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

11.7.2.11.3.3.3 TDES2 Context Descriptor

Following table explains the format of the TDES2 Context descriptor.

Table 11-142. TDES2 Context Descriptor

Bit	Name	Description
31:16	IVT	Inner VLAN Tag When the IVLTV bit of TDES3 context descriptor is set and the TCMSSV and OSTC bits of TDES3 context descriptor are reset, TDES2[31:16] contains the inner VLAN Tag to be inserted in the subsequent Transmit packets.
15:14	Rsvd	Reserved
13:0	MSS	TSO Split header is not supported. The value of this bit is always zero.

11.7.2.11.3.3.4 TDES3 Context Descriptor

Table 11-143. TDES3 Context Descriptor

31	30	29:28	27	26	25:24	23	22:18	17	16	15:0
OWN	CTXT	Rsvd	OSTC	TCMSSV	Rsvd	CDE	Rsvd	IVLTV	VLTV	VT

Follwoing table details the format of the TDES3 Context descriptor.

Table 11-144. TDES3 Context Descriptor

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the module DMA owns the descriptor. When this bit is reset, it indicates that the

Table continues on the next page...

Table 11-144. TDES3 Context Descriptor (continued)

Bit	Name	Description
		application owns the descriptor. The DMA clears this bit immediately after the read.
30	CTXT	Context Type This bit should be set to 1'b1 for Context descriptor.
29:28	Rsvd	Reserved
27	OSTC	One-Step Timestamp Correction Enable When this bit is set, the DMA performs a one-step timestamp correction with reference to the timestamp values provided in TDES0 and TDES1.
26	TCMSSV	One-Step Timestamp Correction Input or MSS Valid When this bit and the OSTC bit are set, it indicates that the Timestamp Correction input provided in TDES0 and TDES1 is valid. When the OSTC bit is reset and this bit and the TSE bit of TDES3 are set in subsequent normal descriptor, it indicates that the MSS input in TDES2 is valid.
25:24	Rsvd	Reserved
23	CDE	Context Descriptor Error When this bit is set, it indicates that the descriptor content is incorrect.The DMA sets this bit during write-back while closing the context descriptor. Descriptor Errors can be: • Incorrect sequence from the context descriptor. For example, a location before the first descriptor for a packet. • All 1s. • CD, LD, and FD bits set to 1. Note 1: When Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA will set the TI bit in the DMA_CH#_Status register Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be

Table continues on the next page...

Table 11-144. TDES3 Context Descriptor (continued)

Bit	Name	Description
		considered as the First Descriptor (even if FD bit is not set) and partial packet is sent.
22:20	Rsvd	Reserved
19:18	IVTIR	Inner VLAN Tag Insert or Replace When this bit is set, these bits request the MAC to perform Inner VLAN tagging or un-tagging before transmitting the packets. If the packet is modified for VLAN tags, the MAC automatically recalculates and replaces the CRC bytes. The following list describes the values of these bits: • 2'b00: Do not add the inner VLAN tag. • 2'b01: Remove the inner VLAN tag from the packets before transmission. This option should be used only with the VLAN frames. • 2'b10: Insert an inner VLAN tag with the tag value programmed in the MAC_In-ner_VLAN_Incl register or context descriptor. • 2'b11: Replace the inner VLAN tag in packets with the tag value programmed in the MAC_Inner_VLAN_Incl register or context descriptor. This option should be used only with the VLAN frames. These bits are valid when the Enable SA and VLAN Insertion on Tx and Enable Double VLAN Processing options are selected.
17	IVLTV	Inner VLAN Tag Valid When this bit is set, it indicates that the IVT field of TDES2 is valid.
16	VLTV	VLAN Tag Valid When this bit is set, it indicates that the VT field of TDES3 is valid.
15:0	VT	VLAN Tag This field contains the VLAN Tag to be inserted or replaced in the packet. This field is used as VLAN Tag only when the VLTI bit of the MAC_VLAN_Incl register is reset.

11.7.2.11.4 Receive Descriptor

The DMA in the module attempts to read a descriptor only if the Tail Pointer is different from the Base Pointer or current pointer. It is recommended to have a descriptor ring with a length that can accommodate at least two complete packets received by the MAC. Otherwise, the performance of the DMA is impacted greatly because of the unavailability of the descriptors. In such situations, the RxFIFO in MTL becomes full and starts dropping packets.

The following Receive Descriptors are present:

- Normal descriptors
- Context descriptors

All RX descriptors are prepared by the software and given to the DMA as "Normal" Descriptors with the content as shown in Receive Normal Descriptor (Read Format). The DMA reads this descriptor and after transferring a received packet (or part of) to the buffers indicated by the descriptor, the Rx DMA will close the descriptor with the corresponding packet status. The format of this status is given in the "Receive Normal Descriptor (Write-Back Format)".

For some packets, the normal descriptor bits are not enough to write the complete status. For such packets, the RX DMA writes the extended status to the next descriptor (without processing or using the Buffers/ Pointers embedded in that descriptor). The format and content of the descriptor write back is described in "Receive Context Descriptor".

11.7.2.11.4.1 Receive Normal Descriptor (Read Format)

The read format for a Receive Normal descriptor is made up of a header or Buffer 1 address, reserved field, payload or Buffer 2 or Next Descriptor address, a 30-bit reserved filed, OWN bit, and an interrupt bit.

Following figure shows the Read Format for a Receive Normal Descriptor.

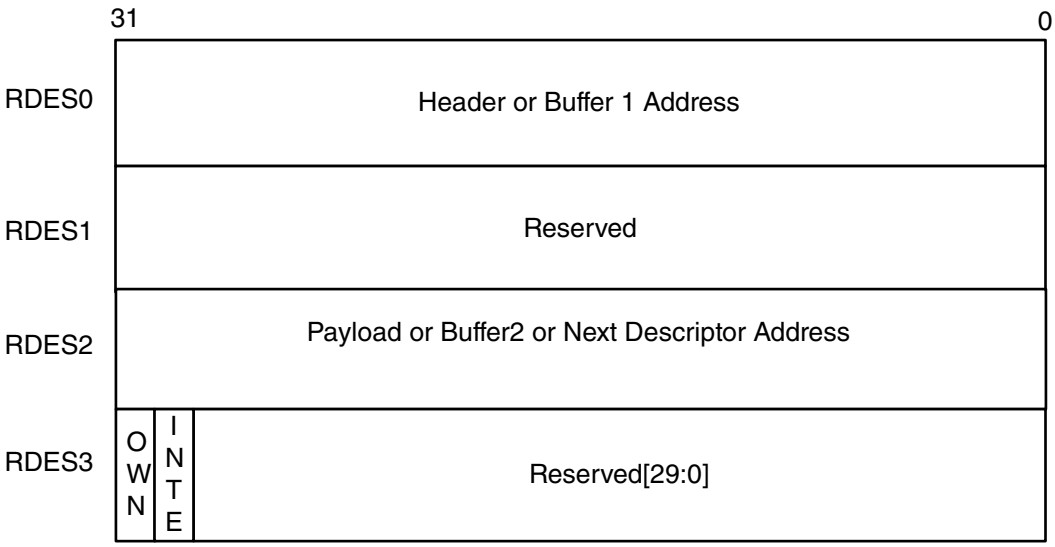


Figure 11-109. Receive Normal Descriptor Read Format

NOTE

In the Receive Descriptor (Read Format), if the Buffer Address field is all 0s, the module does not transfer data to that buffer and skips to the next buffer or next descriptor.

11.7.2.11.4.1.1 RDES0 Normal Descriptor (Read Format)

Following table explains the read format of the RDES0 Normal Descriptor.

Table 11-145. RDES0 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF1AP	Header or Buffer 1 Address Pointer When the SPH bit of Control register of a channel is reset, these bits indicate the physical address of Buffer 1. When the SPH bit is set, these bits indicate the physical address of Header Buffer where the Rx DMA writes the L2/L3/L4 header bytes of the received packet. The application can program a byte-aligned address for this buffer which means that the LS bits of this field can be non-zero. However, while transferring the start of packet, the DMA performs a Write operation with RDES0[1:0] (or RDES0[2:0]/[3:0] in case of 64-/128-bit configuration) as zero. However, the packet data is shifted as per actual offset as given by buffer address pointer.

Table 11-145. RDES0 Normal Descriptor (Read Format)

Bit	Name	Description
		If the address pointer points to a buffer where the middle or last part of the packet is stored, the DMA ignores the offset address and writes to the full location as indicated by the data-width.

11.7.2.11.4.1.2 RDES1 Normal Descriptor (Read Format)

Following table describes the read format of the RDES1 Normal Descriptor.

Table 11-146. RDES1 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	Reserved or BUF1AP	In 64-bit addressing mode, this field contains the most-significant 32 bits of the Buffer 1 Address Pointer. Otherwise, this field is reserved.

11.7.2.11.4.1.3 RDES2 Normal Descriptor (Read Format)

Following table provides the read format of the RDES2 Normal Descriptor.

Table 11-147. RDES2 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF2AP	Buffer 2 Address Pointer These bits indicate the physical address of Buffer 2. When the SPH bit of the DMA_CH#_Control register is set, the buffer address pointer must be bus width-aligned, that is, RDES2[3:0, 2:0, or 1:0] = 0 corresponding to 128, 64, or 32 bus width. LSBs are ignored internally. When the SPH bit of the DMA_CH#_Control register is reset, there is no limitations on the RDES2 value. However, the RxDMA uses the LS Bits of the pointer address only while transferring the start bytes of a packet. If the BUF2AP is giving the address of a buffer in which the middle or last part of a packet is stored, the DMA ignores BUF2AP[3:0 or 2:0 or 1:0] (corresponding to 128- or 64- or 32-bit data-bus) and writes to the complete location.

11.7.2.11.4.1.4 RDES3 Normal Descriptor (Read Format)

Following table describes the read format of the RDES3 Normal Descriptor.

31	30	19-26	25	24	23:0
OWN	IOC	Rsvd	BUF2V	BUF1V	Rsvd

Table 11-148. RDES3 Normal Descriptor (Read Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the module DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: • The DMA completes the packet reception • The buffers associated with the descriptor are full
30	IOC	Interrupt Enabled on Completion When this bit is set, an interrupt is issued to the application when the DMA closes this descriptor.
29:26	Rsvd	Reserved
25	BUF2V	Buffer 2 Address Valid When this bit is set, it indicates to the DMA that the buffer 2 address specified in RDES2 is valid. The application must set this bit so that the DMA can use the address, to which the Buffer 2 address in RDES2 is pointing, to write received packet data.
24	BUF1V	Buffer 1 Address Valid When set, this indicates to the DMA that the buffer 1 address specified in RDES1 is valid. The application must set this value if the address pointed to by Buffer 1 address in RDES1 can be used by the DMA to write received packet data.
23:0	Rsvd	Reserved

11.7.2.11.4.2 Receive Normal Descriptor (Write-Back Format)

The following figure illustrates the write-back format for a Receive Normal descriptor.

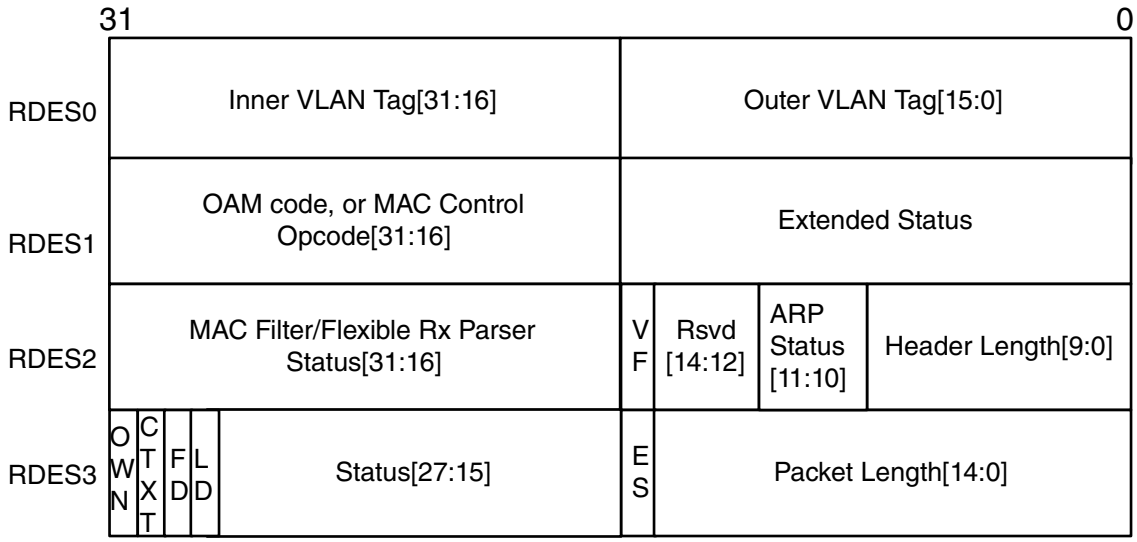


Figure 11-110. Receive Normal Descriptor (Write-Back Format)

NOTE

When Flexible RX Parser is enabled, RDES2[31:16] indicates Parser status, and not the MAC filter status. MAC filter status is not available when Flexible RX parser is enabled.

11.7.2.11.4.2.1 RDES0 Normal Descriptor (Write-Back Format)

Following table describes the write-back format for the RDES0 Normal Descriptor.

Table 11-149. RDES0 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:16	IVT	Inner VLAN Tag This field contains the Inner VLAN tag of the received packet if the RS0V bit of RDES3 is set. This is valid only when Double VLAN tag processing and VLAN tag stripping are enabled.
15:0	OVT	Outer VLAN Tag This field contains the Outer VLAN tag of the received packet if the RS0V bit of RDES3 is set.

11.7.2.11.4.2.2 RDES1 Normal Descriptor (Write-Back Format)

The Status fields in write-back format are valid only for the last descriptor (RDES3[28] is set). Following table provides the details of the write-back format for RDES1 Normal Descriptor.

31:16	15	14	13	12	11:8	7	6	5	4	3	2:0
OPC	TD	TSA	PV	PFT	PMT	IPCE	IPCB	IPV6	IPV4	IPHE	PT

Table 11-150. RDES1 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:16	OPC	OAM Sub-Type Code, or MAC Control Packet opcode OAM Sub-Type Code If Bits[18:16] of RDES3 are set to 3'b111, this field contains the OAM sub-type and code fields. MAC Control Packet opcode The bits[15:8] of RDES3 contains the subtype and bits [7:0] contains the code.
15	TD	Timestamp Dropped This bit indicates that the timestamp was captured for this packet but it got dropped in the MTL Rx FIFO because of overflow. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.
14	TSA	Timestamp Available When Timestamp is present, this bit indicates that the timestamp value is available in a context descriptor word 2 (RDES2) and word 1(RDES1). This is valid only when the Last Descriptor bit (RDES3 [28]) is set. The context descriptor is written in the next descriptor just after the last normal descriptor for a packet.
13	PV	PTP Version This bit indicates that the received PTP message has the IEEE 1588 version 2 format. When this bit is reset, it indicates the IEEE 1588 version 1 format. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.

Table continues on the next page...

Table 11-150. RDES1 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
12	PFT	PTP Packet Type This bit indicates that the PTP message is sent directly over Ethernet. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.
11:8	PMT	PTP Message Type These bits are encoded to give the type of the message received: • 0000: No PTP message received • 0001: SYNC (all clock types) • 0010: Follow_Up (all clock types) • 0011: Delay_Req (all clock types) • 0100: Delay_Resp (all clock types) • 0101: Pdelay_Req (in peer-to-peer transparent clock) • 0110: Pdelay_Resp (in peer-to-peer transparent clock) • 0111: Pdelay_Resp_Follow_Up (in peer-to-peer transparent clock) • 1000: Announce • 1001: Management • 1010: Signaling • 1011–1110: Reserved • 1111: PTP packet with Reserved message type These bits are available only when you select the Timestamp feature.
7	IPCE	IP Payload Error When this bit is set, it indicates either of the following: • The 16-bit IP payload checksum (that is, the TCP, UDP, or ICMP checksum) calculated by the MAC does not match the corresponding checksum field in the received segment.

Table continues on the next page...

Table 11-150. RDES1 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		• The TCP, UDP, or ICMP segment length does not match the payload length value in the IP Header field. • The TCP, UDP, or ICMP segment length is less than minimum allowed segment length for TCP, UDP, or ICMP. Bit 15 (ES) of RDES3 is not set when this bit is set.
6	IPCB	IP Checksum Bypassed This bit indicates that the checksum offload engine is bypassed. This bit is available when you select the Enable Receive TCP/IP Checksum Check feature.
5	IPV6	IPv6 header Present This bit indicates that an IPV6 header is detected. When the Enable Split Header Feature option is selected and the SPH bit of Control Register of a channel is set, the IPV6 header is available in the header buffer area to which RDES0 is pointing.
4	IPV4	IPV4 Header Present This bit indicates that an IPV4 header is detected. When the SPH bit of RDES3 is set, the IPV4 header is available in the header buffer area to which RDES0 is pointing.
3	IPHE	IP Header Error When this bit is set, it indicates either of the following: • The 16-bit IPv4 header checksum calculated by the MAC does not match the received checksum bytes. • The IP datagram version is not consistent with the Ethernet Type value. • Ethernet packet does not have the expected number of IP header bytes. This bit is valid when either Bit 5 or Bit 4 is set. This bit is available when you select the Enable Receive TCP/IP Checksum Check feature.
2:0	PT	Payload Type

Table 11-150. RDES1 Normal Descriptor (Write-Back Format)

Bit	Name	Description
		These bits indicate the type of payload encapsulated in the IP datagram processed by the Receive Checksum Offload Engine (COE): • 3'b000: Unknown type or IP/AV payload not processed • 3'b001: UDP • 3'b010: TCP • 3'b011: ICMP • 3'b110: AV Tagged Data Packet • 3'b111: AV Tagged Control Packet • 3'b101: AV Untagged Control Packet • 3'b100: IGMP if IPV4 Header Present bit is set else DCB (LLDP) Control Packet If the COE does not process the payload of an IP datagram because there is an IP header error or fragmented IP, it sets these bits to 3'b000.

11.7.2.11.4.2.3 RDES2 Normal Descriptor (Write-Back Format)

31:29	28	27	26:19	18	17	16	15	14	13:11	10	9:0
L3L4FM	L4FM	L3FM	MADRM	HF	DAF	SAF	OTS	ITS	Rsvd	ARPNR	HL

Table 11-151. RDES2 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:29	L3L4FM	Layer 3 and Layer 4 Filter Number Matched These bits indicate the number of the Layer 3 and Layer 4 Filter that matched the received packet: • 000: Filter 0 • 001: Filter 1 • 010: Filter 2 • 011: Filter 3 • 100: Filter 4 • 101: Filter 5

Table continues on the next page...

Table 11-151. RDES2 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		• 110: Filter 6 • 111: Filter 7 This field is valid only when Bit 28 or Bit 27 is set high. When more than one filter matches, these bits give the number of lowest filter. Note: This status is not available when Flexible RX Parser is enabled.
28	L4FM	Layer 4 Filter Match When this bit is set, it indicates that the received packet matches one of the enabled Layer 4 Port Number fields. This status is given only when one of the following conditions is true: • Layer 3 fields are not enabled and all enabled Layer 4 fields match • All enabled Layer 3 and Layer 4 filter fields match When more than one filter matches, this bit gives the layer 4 filter status of filter indicated by Bits[31:29]. Note: This status is not available when Flexible RX Parser is enabled.
27	L3FM	Layer 3 Filter Match When this bit is set, it indicates that the received packet matches one of the enabled Layer 3 IP Address fields. This status is given only when one of the following conditions is true: • All enabled Layer 3 fields match and all enabled Layer 4 fields are bypassed • All enabled filter fields match When more than one filter matches, this bit gives the layer 3 filter status of filter indicated by Bits[31:29]. Note: This status is not available when Flexible RX Parser is enabled.
26:19	MADRM	MAC Address Match or Hash Value When the HF bit is reset, this field contains the MAC address register number that matched the Destination address of the received packet. This field is valid only if the DAF bit is reset.

Table continues on the next page...

Table 11-151. RDES2 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		When the HF bit is set, this field contains the hash value computed by the MAC. A packet passes the hash filter when the bit corresponding to the hash value is set in the hash filter register. Note: This status is not available when Flexible RX Parser is enabled.
18	HF	Hash Filter Status When this bit is set, it indicates that the packet passed the MAC address hash filter. Bits[26:19] indicate the hash value. Note: This status is not available when Flexible RX Parser is enabled.
17	DAF/RXPI	Destination Address Filter Fail When Flexible RX Parser is disabled, and this bit is set, it indicates that the packet failed the DA Filter in the MAC. When Flexible RX Parser is enabled, this bit is set to indicate that the packet parsing is incomplete (RXPI) due to ECC error. Note: When this bit is set, ES bit of RDES3 is also set.
16	SAF/RXPD	SA Address Filter Fail When Flexible RX Parser is disabled, and this bit is set, it indicates that the packet failed the SA Filter in the MAC. When Flexible RX Parser is enabled, this bit is set to indicate that the packet is dropped (RXPD) by the parser. Note: When this bit is set, ES bit of RDES3 is also set.
15	OTS	VLAN Filter Status When set, this bit indicates that the VLAN Tag of the received packed passed the VLAN filter. This bit is redefined as Outer VLAN Tag Filter Status (OTS). This bit is valid for both Single and Double VLAN Tagged frames
14	ITS	Inner VLAN Tag Filter Status (ITS) This bit is valid only for Double VLAN Tagged frames, when Double VLAN Processing is enabled. For more information, see Filter Status.
13:11	Rsvd	Reserved

Table continues on the next page...

Table 11-151. RDES2 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
10	ARPNR	ARP Reply Not Generated When this bit is set, it indicates that the MAC did not generate the ARP Reply for received ARP Request packet. This bit is set when the MAC is busy transmitting ARP reply to earlier ARP request (only one ARP request is processed at a time). This bit is reserved when the Enable IPv4 ARP Offload option is not selected.
9:0	HL	L3/L4 Header Length This field contains the length of the header of the packet split by the MAC at L3 or L4 header boundary as identified by the MAC receiver. This field is valid only when the first descriptor bit is set (FD = 1). The header data is written to the Buffer 1 address of corresponding descriptor. If header length is zero, this field is not valid. It implies that the MAC did not identify and split the header. This field is valid when the Enable Split Header Feature option is selected.

11.7.2.11.4.2.4 RDES3 Normal Descriptor (Write-Back Format)

Following table describes the write-back format for the RDES3 Normal Descriptor.

31	30	29	28	27	26	25	24	23	22	21	20	19	18:16	15	14:0
OWN	CTXT	FD	LD	RS2V	RS1V	RS0V	CE	GP	RWT	OE	RE	DE	LT	ES	PL

Table 11-152. RDES3 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the module DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true:

Table continues on the next page...

Table 11-152. RDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		• The DMA completes the packet reception • The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor When this bit is set, it indicates that the current descriptor is a context type descriptor. The DMA writes 1'b0 to this bit for normal receive descriptor. When CTXT and FD bits are used together, {CTXT, FD} • 2'b00: Intermediate Descriptor • 2'b01: First Descriptor • 2'b10: Reserved • 2'b11: Descriptor Error (due to all 1s) Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data Receive DMA will set the CDE bit in DMA_CH#_Status register but not the RI bit even when IOC is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.
29	FD	First Descriptor When this bit is set, it indicates that this descriptor contains the first buffer of the packet. If the size of the first buffer is 0, the second buffer contains the beginning of the packet. If the size of the second buffer is also 0, the next descriptor contains the beginning of the packet. See the CTXT bit description for details of using the CTXT bit and FD bit together.
28	LD	Last Descriptor When this bit is set, it indicates that the buffers to which this descriptor is pointing are the last buffers of the packet.
27	RS2V	Receive Status RDES2 Valid

Table continues on the next page...

Table 11-152. RDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		When this bit is set, it indicates that the status in RDES2 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
26	RS1V	Receive Status RDES1 Valid When this bit is set, it indicates that the status in RDES1 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
25	RS0V	Receive Status RDES0 Valid When this bit is set, it indicates that the status in RDES0 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
24	CE	CRC Error When this bit is set, it indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received packet. This field is valid only when the LD bit of RDES3 is set.
23	GP	Giant Packet When this bit is set, it indicates that the packet length exceeds the specified maximum Ethernet size of 1518, 1522, or 2000 bytes (9018 or 9022 bytes if jumbo packet enable is set). Note: Giant packet indicates only the packet length. It does not cause any packet truncation.
22	RWT	Receive Watchdog Timeout When this bit is set, it indicates that the Receive Watchdog Timer has expired while receiving the current packet. The current packet is truncated after watchdog timeout.
21	OE	Overflow Error When this bit is set, it indicates that the received packet is damaged because of buffer overflow in Rx FIFO. Note: This bit is set only when the DMA transfers a partial packet to the application. This happens only when the Rx FIFO is operating in the threshold mode. In the store-and-forward mode, all partial packets are dropped completely in Rx FIFO.
20	RE	Receive Error

Table continues on the next page...

Table 11-152. RDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		When this bit is set, it indicates that the gmii_rxer_i signal is asserted while the gmii_rxdv_i signal is asserted during packet reception. This error also includes carrier extension error in the GMII and half-duplex mode. Error can be of less or no extension, or error (rxd!= 0f) during extension.
19	DE	Dribble Bit Error When this bit is set, it indicates that the received packet has a non-integer multiple of bytes (odd nibbles). This bit is valid only in the MII Mode.
18:16	LT	Length/Type Field This field indicates if the packet received is a length packet or a type packet. The encoding of the 3 bits is as follows: • 3'b000: The packet is a length packet • 3'b001: The packet is a type packet. • 3'b011: The packet is a ARP Request packet type • 3'b100: The packet is a type packet with VLAN Tag • 3'b101: The packet is a type packet with Double VLAN Tag • 3'b110: The packet is a MAC Control packet type • 3'b111: The packet is a OAM packet type • 3'b010: Reserved
15	ES	Error Summary When this bit is set, it indicates the logical OR of the following bits: • RDES3[24]: CRC Error • RDES3[19]: Dribble Error • RDES3[20]: Receive Error • RDES3[22]: Watchdog Timeout • RDES3[21]: Overflow Error • RDES3[23]: Giant Packet

Table continues on the next page...

Table 11-152. RDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		• RDES2[17]: Destination Address Filter Fail, when Flexible RX Parser is enabled • RDES2[16]: SA Address Filter Fail, when Flexible RX Parser is enabled This field is valid only when the LD bit of RDES3 is set.
14:0	PL	Packet Length These bits indicate the byte length of the received packet that was transferred to system memory (including CRC). This field is valid when the LD bit of RDES3 is set and Overflow Error bits are reset. The packet length also includes the two bytes appended to the Ethernet packet when IP checksum calculation is enabled and the received packet is not a MAC control packet. This field is valid when the LD bit of RDES3 is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current packet.

11.7.2.11.4.3 Receive Context Descriptor

This descriptor is read-only for the application. Only the DMA can write to this descriptor. The context descriptor provides information about the extended status related to the last received packet. The Bit 30 of RDES3 indicates the context type descriptor.

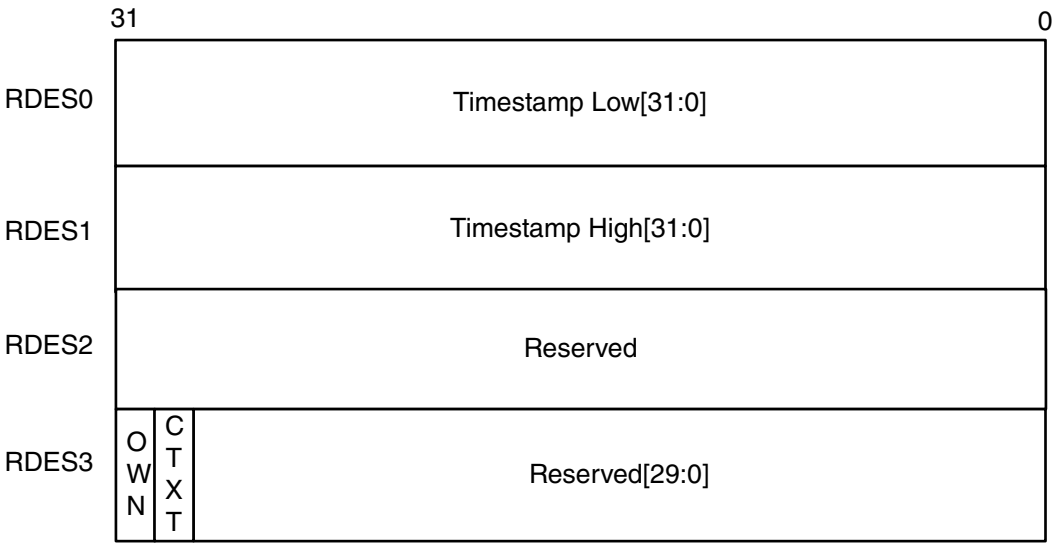


Figure 11-111. Receive Context Descriptor Format

11.7.2.11.4.3.1 RDES0 Context Descriptor

Table 11-153. RDES0 Context Descriptor

Bit	Name	Description
31:0	RTSL	Receive Packet Timestamp Low The DMA updates this field with least significant 32 bits of the timestamp captured for corresponding Receive packet. When this field and the RTSH field of RDES1 show all-ones value, the timestamp must be considered as corrupt.

11.7.2.11.4.3.2 RDES1 Context Descriptor

Table 11-154. RDES1 Context Descriptor

Bit	Field	Description
31:0	RTSH	Receive Packet Timestamp High The DMA updates this field with most significant 32 bits of the timestamp captured for corresponding receive packet. When this field and the RTSL field of RDES0 show all-ones value, the timestamp must be considered as corrupt.

11.7.2.11.4.3.3 RDES2 Context Descriptor

Table 11-155. RDES2 Context Descriptor

Bit	Description
31:0	Reserved

11.7.2.11.4.3.4 RDES3 Context Descriptor

31	30	29	28:0
OWN	CTXT	DE	Rsvd

Table 11-156. RDES3 Context Descriptor

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: • The DMA completes the packet reception • The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor When this bit is set, it indicates that the current descriptor is a context descriptor. The DMA writes 1'b1 to this bit for context descriptor. DMA writes 2’b11 to indicate a descriptor error due to all 1s. When CTXT and DE bits are used together, {CTXT, DE} • 2’b00: Reserved • 2’b01: Reserved • 2’b10: Context Descriptor • 2’b11: Descriptor Error Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data Receive DMA will set the CDE bit in DMA_CH#_Status

Table continues on the next page...

Table 11-156. RDES3 Context Descriptor (continued)

Bit	Name	Description
		register but not the RI bit even when IOC is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.
29	DE	Descriptor Error See the CTXT bit description for details of using theDE bit along with CTXT bit.
28:0	Rsvd	Reserved

11.7.2.11.5 Enhanced Descriptor for Time-Based Scheduling

The time-based scheduling feature needs 32 bytes Enhanced Descriptors to be enabled on all the DMA channels that intend to use the feature (by setting the EDSE bit of DMA_CH(#i)_TX_Control register).

The structure of 32-byte Descriptor for the Context and the Normal Descriptor in Read and Write formats are described in the following sections.

11.7.2.11.5.1 Enhanced Normal Descriptor - Read (32-Bit Mode)

The fields present in the first 16 bytes of the Enhanced Descriptor format of normal descriptor are as follows:

- LTV

  Launch Time Valid indicates the Launch Time (LT) and GSN fields present in the descriptor are valid. The LTV must be set only if the FD bit of the descriptor is not.
- GSN

  This field indicates the GCL slot number associated with the packet.
- Launch Time (LT)

  This field indicates the Launch Time associated with the packet.

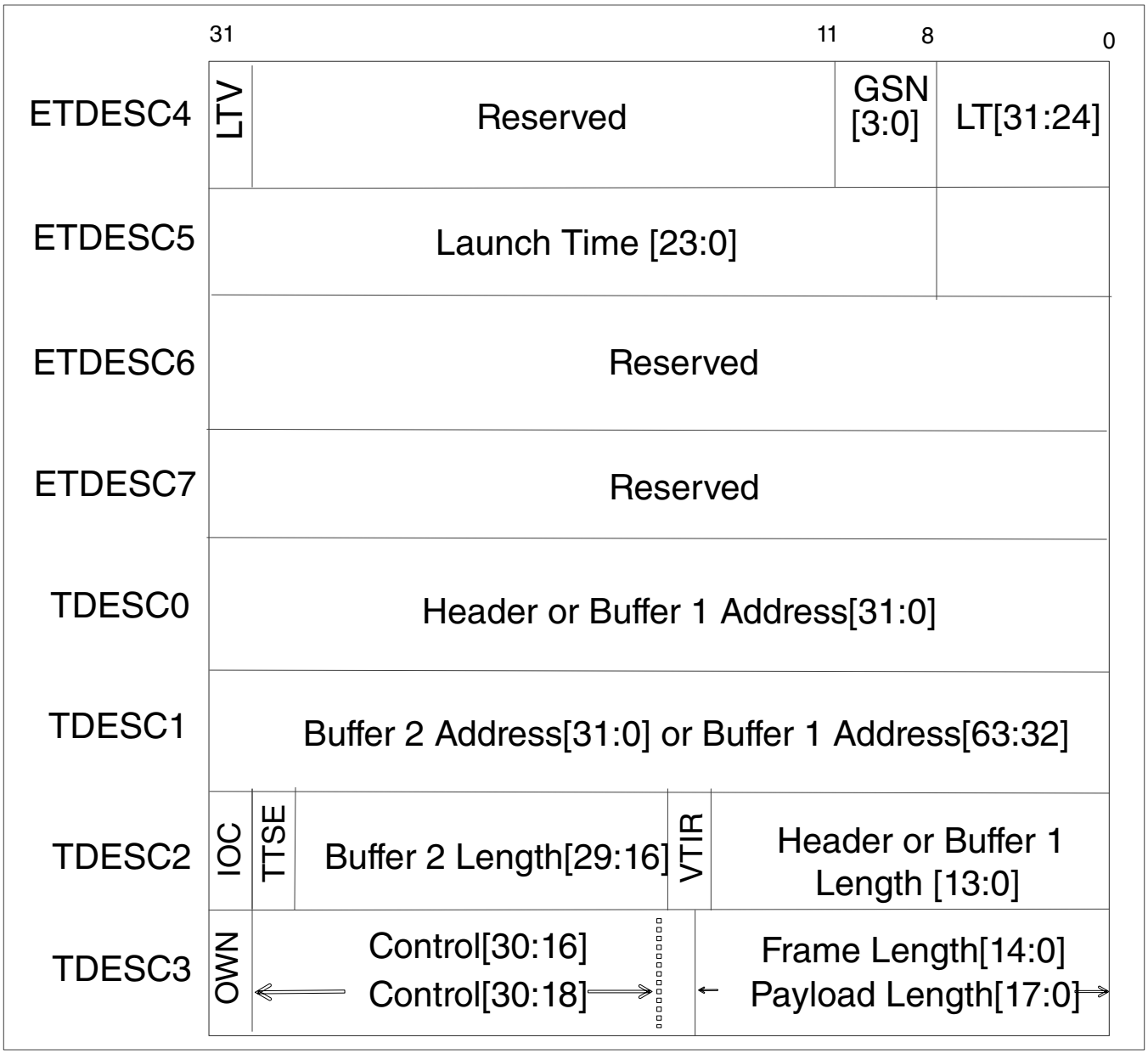


Figure 11-112. Enhanced Normal Descriptor - Read (32-Bit Mode)

11.7.2.11.5.2 Enhanced Normal Descriptor (Write, 32-Bit Mode)

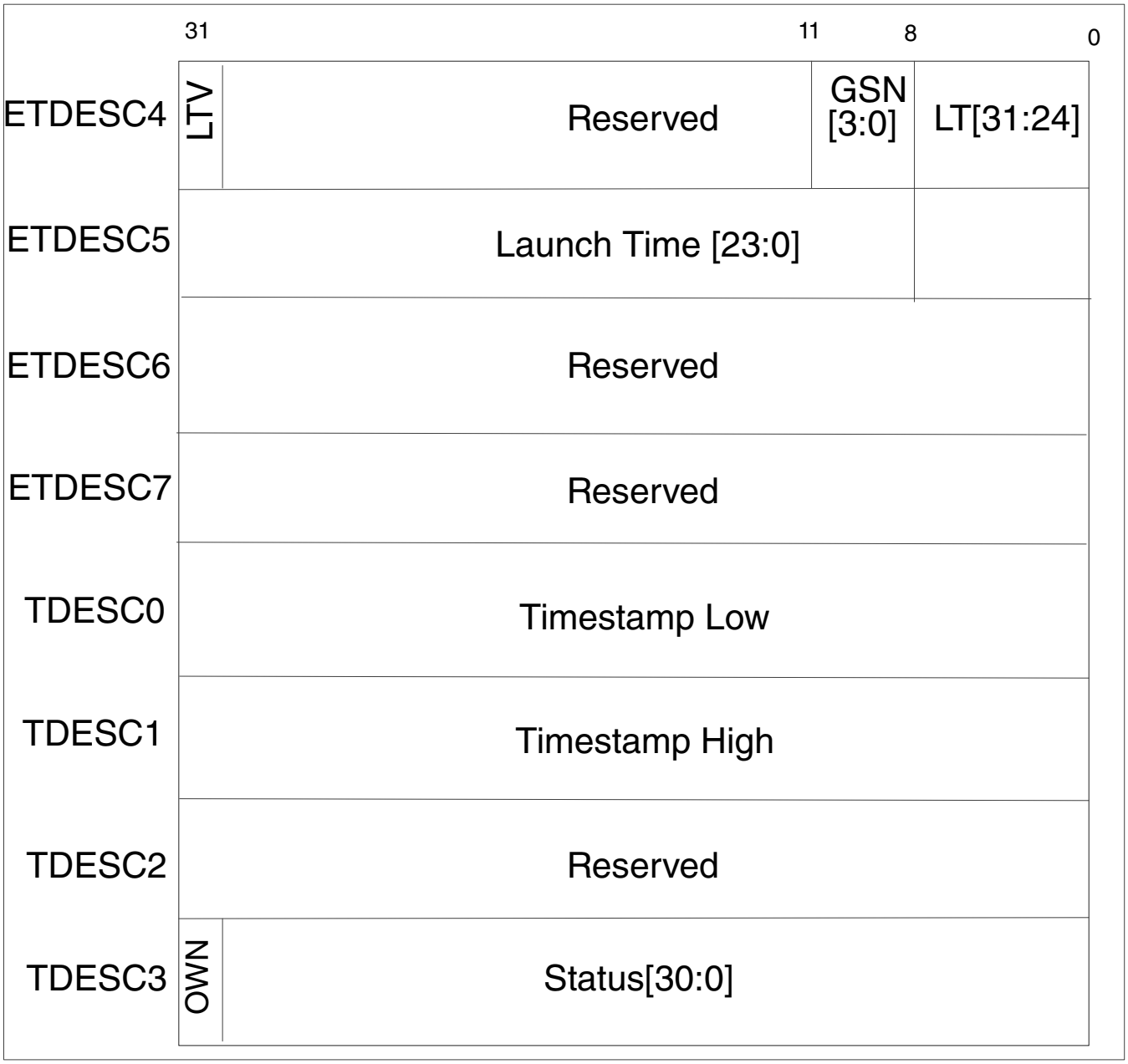


Figure 11-113. Enhanced Normal Descriptor Write (32-Bit Mode)

NOTE

- All the enhanced descriptor examples, assumes a 32-bit data width configurations. However, the same definitions are valid in 64 and 128 bit configurations.

- In both the write back formats, no modifications can be done to the extended 16 bytes. The rest of the 16 bytes (TDESC0 to TDESC3) are written back as per the previous 16 bytes Descriptor format.
- Fetch time when enabled overrides the AV slot function
- When an unaligned new GCL list (with a different CTR) is installed it is recommended not to have traffic during the installation of the new list. Any traffic during the switching of the lists might have unpredictable behavior regarding Fetch, Launch, and Launch expiry as the CTR and BTR values get updated while the frame is being processed

11.7.2.11.5.3 Enhanced Context Descriptor (Read, 32-bit Mode)

The first 16 bytes of the Enhanced context descriptor (ETDESC4 to ETDESC7) are reserved and must be zeros. The fields present in last bytes of the Descriptor (TDESC0 to TDESC3) are same as the context descriptor (TDESC0 to TDESC3) in 16 byte format.

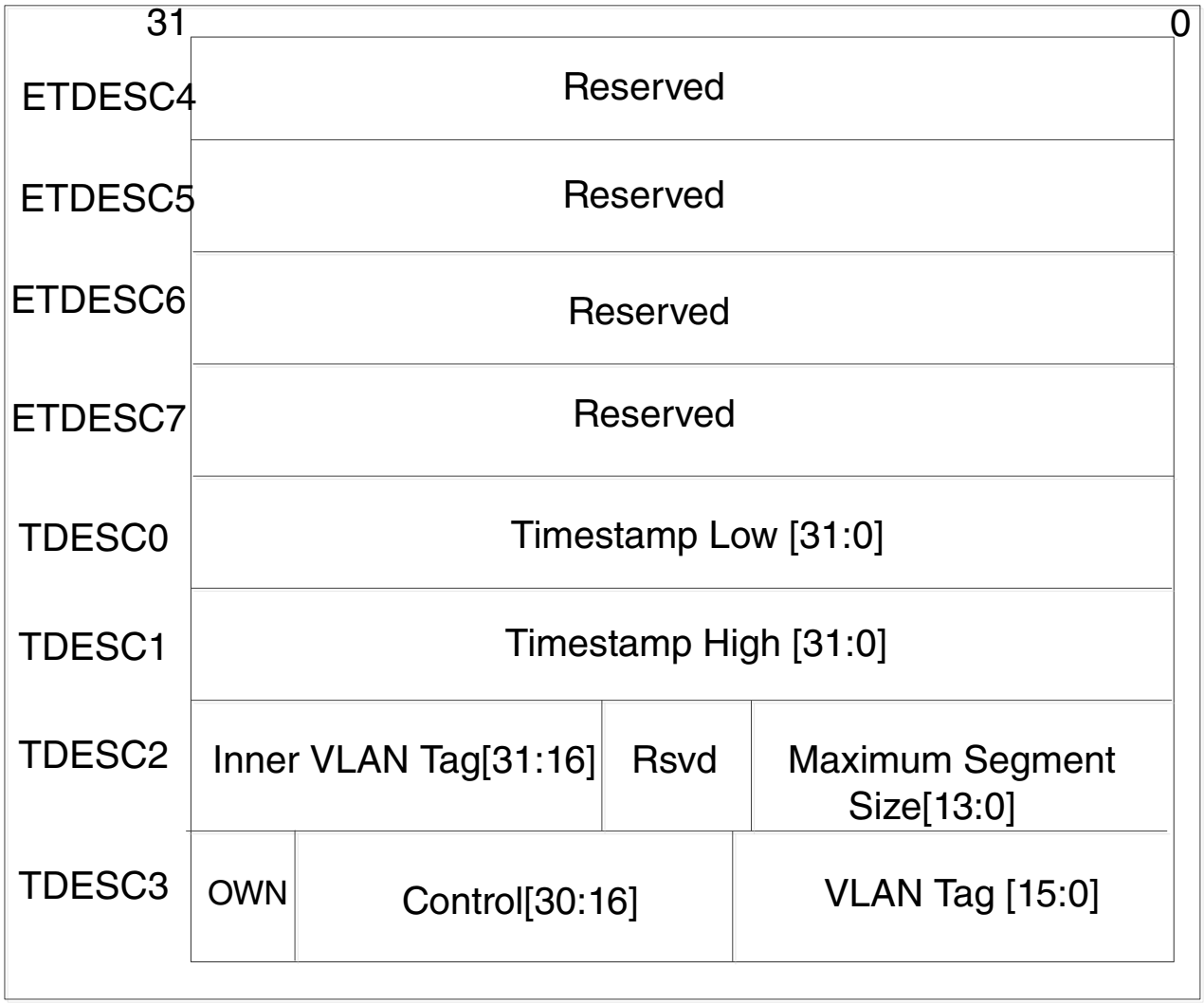
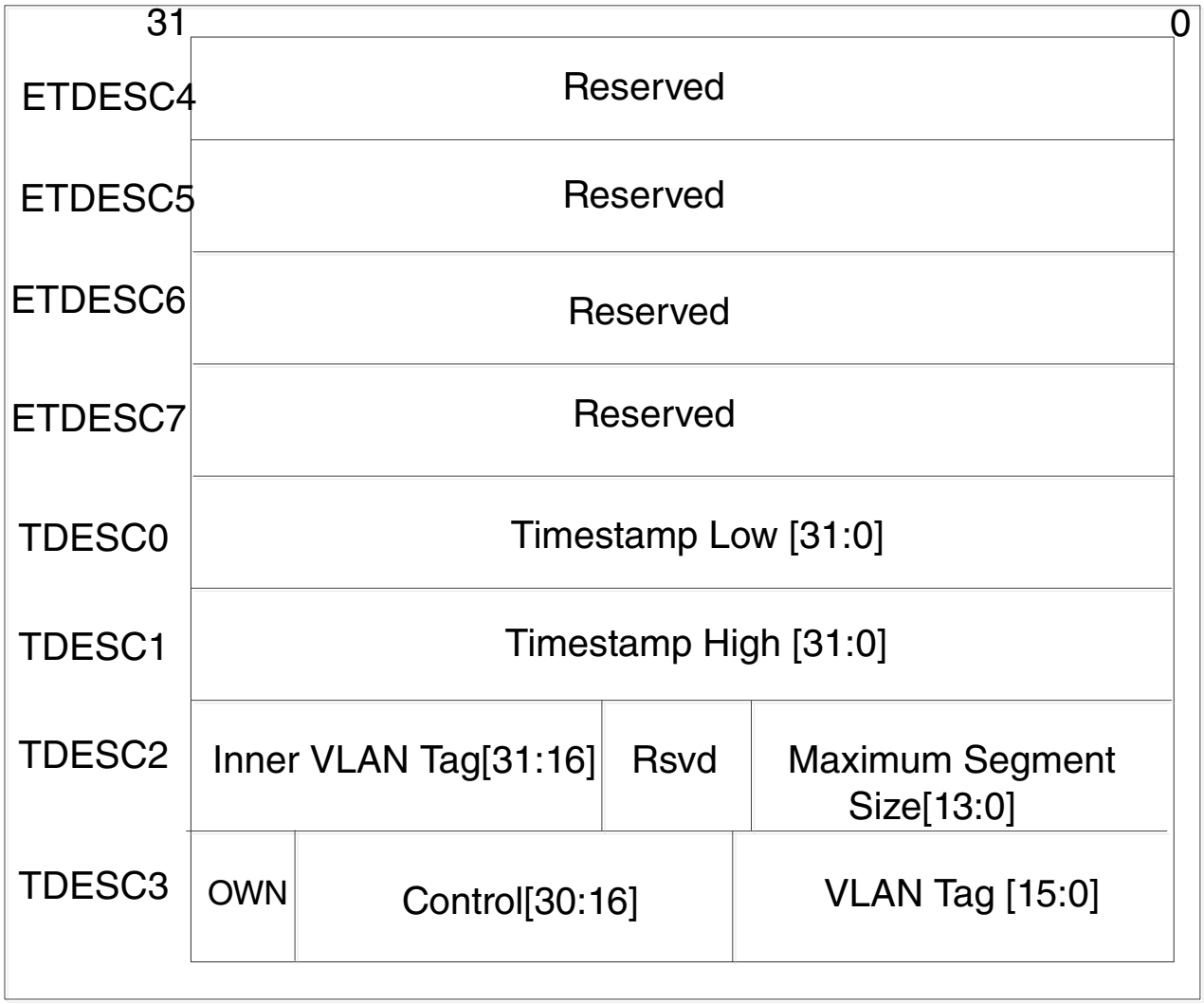


Figure 11-114. Enhanced Context Descriptor - Read


31
0
ETDESC4
Reserved
ETDESC5
Reserved
ETDESC6
Reserved
ETDESC7
Reserved
TDESC0
Timestamp Low [31:0]
TDESC1
Timestamp High [31:0]
TDESC2
Inner VLAN Tag[31:16]
Rsvd
Maximum Segment Size[13:0]
TDESC3
OWN
Control[30:16]
VLAN Tag [15:0]

11.7.2.11.5.4 Enhanced Context Descriptor (Write, 32 Bit Mode)

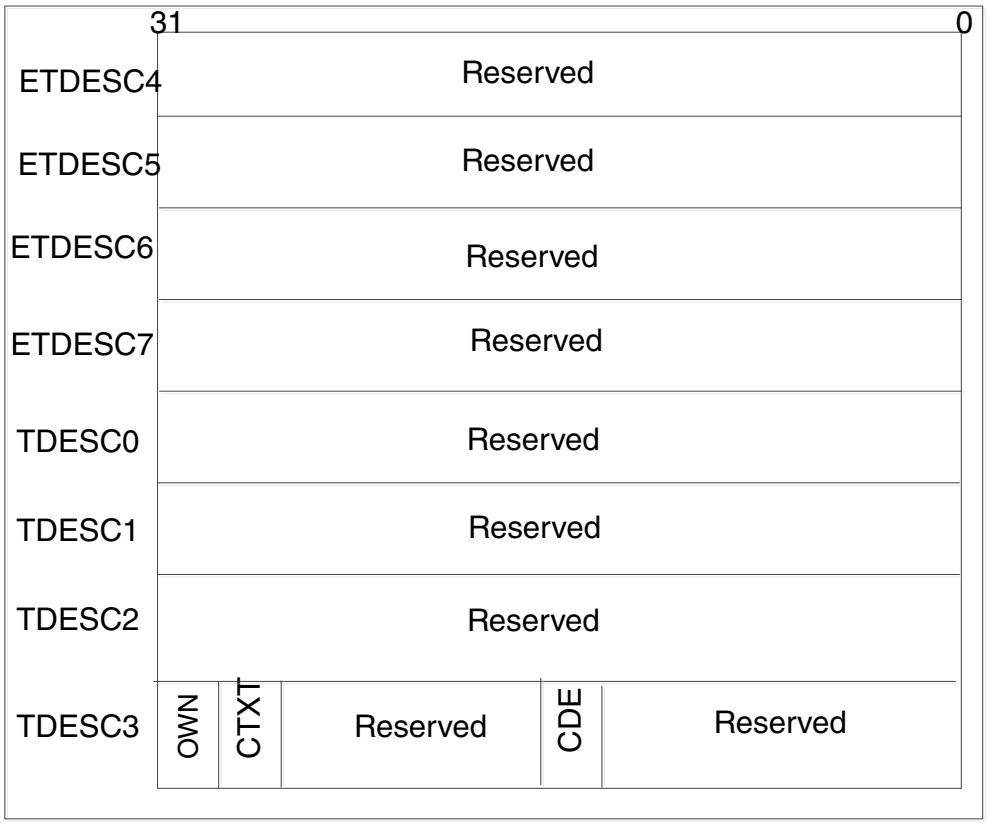


Figure 11-115. Enhanced Context Descriptor - Write

11.7.2.12 Clocks

The table found here describes the clock sources for ENET_QOS. Please see the chip-specific clocking section for clock setting, configuration and gating information.

Table 11-157. ENET_QOS Clocks

Clock name	Description
aclk_i	Bus clock
clk_csr_i	CSR clock, same source as aclk_i
ptp_ref_i	PTP clock

11.7.3 External signals

Table 11-158. External signals

Port name	Interface	I/O	Description
1588_EVENT_AUX_IN	RMII	I	
1588_EVENT_IN	RMII	I	
1588_EVENT_OUT	RMII	O	
TX_CLK	RGMII	O for (RGMII)	RGMII: The transmit clock signal is always provided by the MAC on the TX_CLK line. Source-synchronous clocking is used: the clock signal that is output (by either the PHY or the MAC) is synchronous with the data signals.
RX_CLK	RGMII	I	RGMII: Received clock signal (recovered from incoming received data)
REF_CLK1	RMII	I/O	This 50Mhz clock used by the RMII interface. When RMII mode is selected, RX_CLK (25Mhz or 2.5Mhz) should be derived from this clock.
TX_EN	RMII, RGMII	O	The MAC drives this signal. This signal has multiple functions depending on the selected PHY interface. • RMII: When high, indicates that valid data is being transmitted on the phy_txd_o bus. Synchronous to REF_CLK. • RGMII: Contains TX_EN on the rising edge of TX_CTL, and a logical derivative of TX_EN and TX_ER (TX_EN XOR TX_ER) on the falling edge of TX_CTL.
TX_DATA	RMII, RGMII	O	This is a group of transmit data signals driven by the MAC. It has multiple functions depending on the selected PHY interface. • RMII: Bits[1:0] provide the RMII transmit data. The data is valid only

Table continues on the next page...

Table 11-158. External signals (continued)

Port name	Interface	I/O	Description
			when the TX_EN signal is high. Synchronous to REF_CLK. • RGMII: Bits[3:0] provide serial output Ethernet data
RX_DATA	RMII, RGMII	I	This is a group of data signals received from the PHY. It has multiple functions depending on the selected PHY interface. • RMII: Bits [1:0] provide the RMII receive data. The data is valid only when the RX_DV signal is high. Synchronous to REF_CLK. • RGMII: Bits [3:0] contains the Ethernet input data transferred from the PHY to the media-access controller.
MDC	RMII, RGMII	O	Management Data Clock. The MAC provides timing reference for the MDIO or MII through this periodic clock. This clock is generated from the application clock through a clock divider controlled by MAC_MDIO_ADDRESS[CR].
MDIO	RMII, RGMII	I/O	Transfers control information between the external PHY and the MAC. Data is synchronous to MDC. This signal is an input after reset.
RX_DV	MII, RMII	I	The PHY drives this signal. It has multiple functions depending on the selected PHY interface. • RMII: Contains the crs and data valid information of the Receive interface. Synchronous to REF_CLK.
RX_EN	RGMII	I	In RGMII mode, contains RX_DV on the rising edge of RX_CLK, and a logical derivative of RX_EV and RX_ER (RX_EV XOR RX_ER) on the falling edge of RX_CLK.

11.7.4 Initialization

This section provides the instructions for initializing the DMA or MAC registers in the proper sequence.

11.7.4.1 Initializing DMA

Complete the following steps to initialize the DMA:

1. Provide a software reset. This resets all of the MAC internal registers and logic (bit-0 of DMA_Mode).
2. Wait for the completion of the reset process (poll bit 0 of the DMA_MODE, which is only cleared after the reset operation is completed).
3. Program the following fields to initialize the DMA_SYSBUS_MODE register:
   a. AAL
   b. Fixed burst or undefined burst
   c. Burst mode values in case of AHB bus interface, OSR_LMT in case of AXI bus interface.
   d. If fixed length value is enabled, select the maximum burst length possible on the AXI Bus (bits [7:1])
4. Create a descriptor list for transmit and receive. In addition, ensure that the descriptors are owned by DMA (set bit 31 of descriptor TDES3/RDES3). For more information about descriptors, see Descriptors
5. Program the Transmit and Receive Ring length registers (DMA_CH(#i)_TXDESC_RING_LENGTH (for i = 0; i <= 1) and DMA_CH(#i)_RXDESC_RING_LENGTH (for i = 0; i <= 1)). The ring length programmed must be at least 4.

   NOTE

   The descriptor address from the start to the end of the ring must not cross the 4GB boundary.

6. Initialize receive and transmit descriptor list address with the base address of the transmit and receive descriptor (DMA_CH(#i)_TXDESC_LIST_ADDRESS (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1), DMA_CH(#i)_RXDESC_LIST_ADDRESS (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)). Also, program transmit and receive tail pointer registers indicating to the DMA about the available descriptors (DMA_CH(#i)_TXDESC_TAIL_POINTER (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) and

DMA_CH(#i)_RXDESC_TAIL_POINTER (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1))

NOTE

For 40-bit or 48-bit addressing mode, program the higher address List registers (DMA_CH[n]_TXDESC_LIST_HADDRESS, DMA_CH[n]_RXDESC_LIST_HADDRESS).

The tailpointer registers must be advanced to the location immediately after the descriptors that are set, for the DMA to know that additional descriptors are available.

7. Program the settings of the following registers for the parameters like maximum burst-length (PBL) initiated by DMA, descriptor skip lengths, OSP in case of TxDMA, RBSZ in case of RxDMA, and so on:
   - DMA_CH(#i)_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1)
   - DMA_CH(#i)_TX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1)
   - DMA_CH(#i)_RX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)
8. Enable the interrupts by programming the DMA_CH(#i)_Interrupt_Enable (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register.
9. Start the Receive and Transmit DMAs by setting SR (bit 0) of the DMA_CH(#i)_RX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1) and ST (bit 0) of the DMA_CH(#i)_TX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register.
10. Repeat steps 4 to 9 for all the Tx DMA and Rx DMA channels selected in the hardware.

11.7.4.2 Initializing MTL Registers

The Transaction Layer (MTL) registers must be initialized to establish the transmit and receive operating modes and commands.

Complete the following steps to initialize the MTL registers:

1. Program the Tx Scheduling (SCHALG) and Receive Arbitration Algorithm (RAA) fields in MTL_Operation_Mode to initialize the MTL operation in case of multiple Tx and Rx queues.
2. Program the Receive Queue to DMA mapping in MTL_RxQ_DMA_Map0 and MTL_RxQ_DMA_Map1 registers.
3. Program the following fields to initialize the mode of operation in the MTL_TxQ0_Operation_Mode register.
   a. Transmit Store And Forward (TSF) or Transmit Threshold Control (TTC) in case of threshold mode
   b. Transmit Queue Enable (TXQEN) to value 2'b10 to enable Transmit Queue0
   c. Transmit Queue Size (TQS)
4. Program the following fields to initialize the mode of operation in the MTL_RxQ0_Operation_Mode register:
   a. Receive Store and Forward (RSF) or RTC in case of Threshold mode
   b. Flow Control Activation and De-activation thresholds for MTL Receive FIFO (RFA and RFD)
   c. Error Packet and undersized good Packet forwarding enable (FEP and FUP)
   d. Receive Queue Size (RQS)
5. Repeat previous two steps for all MTL Tx and Rx queues selected in the configuration.

11.7.4.3 Initializing MAC

The MAC configuration registers establish the operating mode of the MAC. These registers must be initialized before initializing the DMA.

The following MAC Initialization operations can be performed after DMA initialization. If the MAC initialization is completed before the DMA is configured, enable the MAC receiver (last step in the following sequence) only after the DMA is active. Otherwise, received frames fill the Rx FIFO and overflow.

1. Provide the MAC address registers: MAC_ADDRESS0_HIGH and MAC_ADDRESS0_LOW. If more than one MAC address is enabled in your configuration, program the MAC addresses appropriately.
2. Program the following fields to set the appropriate filters for the incoming frames in the MAC_PACKET_FILTER register:
   a. Receive All
   b. Promiscuous mode
   c. Hash or Perfect Filter
   d. Unicast, multicast, broadcast, and control frames filter settings

3. Program the following fields for proper flow control in the MAC_Q0_TX_FLOW_CTRL register:
   a. Pause time and other Pause frame control bits
   b. Transmit Flow control bits
   c. Flow Control Busy
4. Program the MAC_INTERRUPT_ENABLE register, as required, and if applicable, for your configuration.
5. Program the appropriate fields in the MAC_CONFIGURATION register. For ex: Inter-packet gap while transmission and jabber disable.
6. Set bit 0 and 1 in MAC_CONFIGURATION register to start the MAC transmitter and receiver.

11.7.5 Application Information

11.7.5.1 Performing Normal Receive and Transmit Operation

During normal operation of the module, normal and transmit interrupts are read, descriptors polled, the DMA is suspended (if it does not own descriptors), and values of current host transmitter or receiver descriptor pointers are read for debugging.

For normal operation, complete the following steps:

1. For normal transmit and receive interrupts, read the interrupt status. Then, poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).
2. Set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
3. If the descriptors are not owned by the DMA (or no descriptor is available), the DMA goes into SUSPEND state. The transmission or reception can be resumed by freeing the descriptors and writing the descriptor tail pointer to Tx/Rx tail pointer register (DMA_CH[n]_TXDESC_TAIL_POINTER and DMA_CH[n]_RXDESC_TAIL_POINTER).
4. The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (DMA_CH[n]_CURRENT_APP_TXDESC and DMA_CH[n]_CURRENT_APP_RXDESC).
5. The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (Register DMA_CH[n]_CURRENT_APP_TXBUFFER and DMA_CH[n]_CURRENT_APP_RXBUFFER).

11.7.5.2 Stopping and Starting Transmission

Complete the following steps to pause the transmission.

NOTE

The steps are provided for Channel 0.

1. Disable the Transmit DMA (if applicable) by clearing Bit 0 (ST) of DMA_CH(#i)_TX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) Register.
2. Wait for any previous frame transmissions to complete. You can check this by reading the appropriate bits of MTL_TXQ0_DEBUG Register (TRCSTS is not 01 and TXQSTS=0).
3. Disable the MAC transmitter and MAC receiver by clearing Bit (RE) and Bit 1(TE) of the MAC_Config-uration Register.
4. Disable the Receive DMA (if applicable), after making sure that the data in the Rx FIFO is transferred to the system memory (by reading the appropriate bits of MTL_TXQ0_DEBUG Register, PRXQ=0 and RXQSTS=00).
5. Make sure that both Tx Queue and Rx Queue are empty (TXQSTS is 0 in MTL_TXQ0_DEBUG Register and RXQSTS is 0 in MTL_RXQ0_DEBUG Register).
6. To restart the operation, first start the DMAs, and then enable the MAC Transmitter and Receiver.

NOTE

- Do not change the configuration (such as duplex mode, speed, port, or loop back) when the MAC is actively transmitting or receiving. These parameters are changed by software only when the MAC transmitter and receiver are not active.
- Similarly, do not change the DMA-related configuration when Transmit and Receive DMA are active.

11.7.5.3 Programming Guidelines for Switching to New Descriptor List in RxDMA

Switching to a new descriptor list is different in the Rx DMA compared to the Tx DMA. Switching to a new descriptor list is permitted when the RxDMA is in SUSPEND state, as clarified by the following points:

- Generally, RxDMA prepares the descriptors in advance.
- If the RxDMA goes to SUSPEND due to descriptors not being available, a major failure occurs (software is not able to free the filled-up descriptors/buffers). If this issue is not rectified immediately, frames will be lost because of an RxFIFO Overflow. Therefore, the software is allowed to create a new descriptor list and program the RxDMA to start using it immediately, without going into STOP state.

11.7.5.4 Programming Guidelines for Multi-Channel Multi-Queuing

11.7.5.4.1 Transmit

1. Program the Transmit queue size in the TQS field of MTL_TXQ[n]_OPERATION_MODE register. Based on the value programmed in the TQS field, the size of the queue is determined.

   In the Transmit operation, the number of channels is equal to the number of the queues. Due to this reason, the Channel-to-Queue mapping is fixed.

2. For a queue to be used, the queue must be enabled in TXQEN in the corresponding MTL_TXQ[n]_OPERATION_MODE Register. In DMA configurations, the ST bit of DMA_CH[n]_TX_CONTROL Register and corresponding TXQEN in MTL_TXQ[n]_OPERATION_MODE Register needs to be enabled.
3. The scheduling method needs to be programmed in SCHALG of MTL_OPERATION_MODE register.
4. Program the MTL_TXQ[n]_Quantum_Weight for DCB queue as per the selected algorithm. In case of CBS algorithm in AVB queues, the MTL_TXQ[n]_ETS_CONTROL, MTL_TXQ[n]_SENDSLOPECREDIT, MTL_TXQ[n]_HICREDIT and MTL_TXQ[n]_LOCREDIT registers also need to be programmed as required.
5. If DCB is enabled and PFC function is required, program MAC_TXQ_PRTY_MAP0 Register to assign a fixed priority to the queue. This priority assigned is used for determining if the corresponding queue should stop transmitting packet based on the received PFC packet.

11.7.5.4.2 Receive

1. Program the Receive queue size in the RQS field of MTL_RXQ[n]_OPERATION_MODE Register. Based on the value programmed in RQS field, the size of the queue is determined.
2. Enable the Receive Queues 0 to 4 in the fields RXQ0EN to RXQ4EN in MAC_RXQ_CTRL0 Register for AV or DCB. In DMA configurations, SR bit of

statically or dynamically mapped DMA_CH[n]_RX_CONTROL Register and corresponding RXQ[n]_EN in MAC_RXQ_CTRL0 Register needs to be enabled.

3. The MAC routes the Rx packets to the Rx Queues based on following packet types:
   a. AV PTP Packets: Based on the programming of AVPTPQ in MAC_RXQ_CTRL1 Register.
   b. AV Untagged Control packets: Based on the programming of AVCPQ in MAC_RXQ_CTRL1 Register.
   c. Data Center Bridging (DCB) related Link Layer Discovery Protocol (LLDP) packets. Program DCBCPQ in MAC_RXQ_CTRL1 Register to indicate to MAC which queue should get the DCB packets.
   d. VLAN Tag Priority field in VLAN Tagged packets: Program PSRQ4-0 of the MAC_RXQ_CTRL2 and MAC_RXQ_CTRL3 Register for the routing of tagged packets based on the USP (user Priority) field of the received packets to the Rx Queues 0 to 4.
   e. The AV tagged control and data packets are also routed based on PSRQ field of the MAC_RXQ_CTRL2 and MAC_RXQ_CTRL3 registers.

NOTE

The priorities set in PSRQ4-0 should be unique.

4. If multiple RX DMA channels are enabled, the following programming should be done for proper arbitration and mapping:
   a. Program the RAA field of MTL_OPERATION_MODE register to select the arbitration algorithm to decide which RxQ is read out from the RxFIFO memory.
   b. Program the MTL_RXQ[n]_CONTROL to decide the weights and the packet arbitration for each RxQ.
   c. If static mapping is programmed in MTL_RXQ_DMA_MAP[n] register (RXQ[n]DADMACH is reset to 0), bits RXQx2DMA and others need to be programmed to select the channel for which each queue is mapped.
   d. Set RXQ[n]DADMACH bit in MTL_RXQ_DMA_MAP0 Register to select dynamic mapping of packets in each RxQueue.
   e. In dynamic channel mapping, the routing of a packet to a specific RxDMA channel is decided by the value of DCS field in the lowest MAC Address Register.

11.7.5.4.2.1 Programming Guidelines for Recovering from DMA Channel Failure

When the DMA channel issues a bus error, follow these steps to recover from the failure.

Recovering from the Receive DMA Channel Failure

1. Set the RPF bit to 1. This flushes all the packets one after the other.

This step is optional. However, setting this bit prevents HOL (head-of-line) blocking in the Rx queues when packets sent to the RXDMA are stopped due to bus error.

2. Reprogram the specific Registers of the DMA channel.
3. Start the DMA channel.

Recovering from the Transmit DMA Channel Failure

1. Stop the specific DMA channel, even if it is in active state.
2. Flush the corresponding MTL queue.
3. Reprogram the specific Registers of the DMA channel.
4. Start the DMA channel.

NOTE

Due to the known limitations in the design, reprogramming the DMA channel registers might not always be successful in recovering from a Bus Error. If the module is not fully functional after reprogramming the DMA, as a workaround, issue a Soft Reset to recover from the Bus Error.

11.7.5.5 Programming Guidelines for IEEE 1588 Timestamping

11.7.5.5.1 Initialization Guidelines for System Time Generation

You can enable the timestamp feature by setting Bit 0 of the MAC_TIMESTAMP_CONTROL Register. However, it is essential that the timestamp counter be initialized after this bit is set. Complete the following steps during the module initialization:

1. Mask the Timestamp Trigger interrupt by clearing the bit 16 of MAC_INTERRUPT_ENABLE Register.
2. Set Bit 0 of MAC_TIMESTAMP_CONTROL Register to enable timestamping.
3. Program MAC_SUB_SECOND_INCREMENT Register based on the PTP clock frequency.
4. If you are using the Fine Correction approach, program MAC_Timestamp_Addend and set Bit 5 of MAC_TIMESTAMP_CONTROL Register.
5. Poll the MAC_TIMESTAMP_CONTROL Register until Bit 5 is cleared.
6. Program Bit 1 of MAC_TIMESTAMP_CONTROL Register to select the Fine Update method (if required).
7. Program MAC_SYSTEM_TIME_SECONDS_UPDATE Register and MAC_SYSTEM_TIME_NANOSECONDS_UPDATE Register with the appropriate time value.
8. Set Bit 2 in MAC_TIMESTAMP_CONTROL Register.

9. The timestamp counter starts operation as soon as it is initialized with the value written in the Timestamp Update registers. If one-step timestamping is enabled
   a. To enable one-step timestamping, program Bit 27 of the TDES3 Context Descriptor.
   b. Program registers MAC_TIMESTAMP_INGRESS_ASYM_CORR and MAC_TIMESTAMP_EGRESS_ASYM_CORR to update the correction field in PDelay_Req PTP messages.
10. Enable the MAC receiver and transmitter for proper timestamping.

NOTE

If timestamp operation is disabled by clearing Bit 0 of MAC_TIMESTAMP_CONTROL Register, repeat all these steps to restart the timestamp operation.

11.7.5.5.2 System Time Correction

11.7.5.5.2.1 Coarse Correction Method

To synchronize or update the system time in one process (coarse correction method):

1. Set the offset (positive or negative) in the Timestamp Update registers(MAC_SYSTEM_TIME_SECONDS_UPDATE and MAC_SYSTEM_TIME_NANOSECONDS_UPDATE).
2. Set Bit 3 (TSUPDT) of MAC_TIMESTAMP_CONTROL REGISTER.

The value in the Timestamp Update registers is added to or subtracted from the system time when the TSUPDT bit is cleared.

11.7.5.5.2.2 Fine Correction Method

To synchronize or update the system time to reduce system-time jitter (fine correction method):

1. With the help of the algorithm, calculate the rate by which you want to make the system time increments slower or faster.
2. Update the MAC_TIMESTAMP_ADDEND with the new value and set Bit 5 of the MAC_TIMESTAMP_CONTROL Register.
3. Wait for the time for which you want the new value of the Addend register to be active. You can do this by enabling the Timestamp Trigger interrupt after the system time reaches the target value.
4. Program the required target time in MAC_PPS[n]_TARGET_TIME_SECONDS Register and MAC_PPS[n]_TARGET_TIME_NANOSECONDS Register.

5. Enable the Timestamp interrupt in bit 12 of MAC_INTERRUPT_ENABLE register.
6. Set bit 4 in Register MAC_Timestamp_Control.
7. When this trigger causes an interrupt, read MAC_INTERRUPT_STATUS Register.
8. Reprogram MAC_TIMESTAMP_ADDEND Register with the old value and set bit 5 again.

11.7.5.6 Programming Guidelines for AV Feature

After you enable the AV feature in the module controller, there are several programming tasks to be followed:

- Initializing the DMA for QOS-AHB configurations only
- Enabling slot number checking
- Enabling average bits per slot reporting
- Disabling flow control for AV-enabled queues (transmit and receive flow control)

11.7.5.6.1 Initializing the DMA in Audio Video Feature

The first step in programming the AV feature in a QOS-AHB configuration is to initialize the DMA.

Use this initialization sequence for QOS-AHB/QOS-AXI configurations with AV feature.

1. Provide a software reset to reset all QOS internal registers and logic (bit 0 in DMA_MODE register).
2. Wait for the completion of the reset process. Poll bit 0 of the DMA_MODE register), which is cleared only after the reset operation is completed.
3. Program the fields to initialize the DMA register by setting the values in DMA_MODE register.
4. Create a proper descriptor list for transmit and receive. In addition, ensure that the DMA owns the Transmit and Receive descriptors. When OSF mode is used, at least two TX descriptors are required.

   For more information about descriptors, see Descriptors.

5. Make sure that your software creates three or more different transmit or receive descriptors in the list before reusing any of the descriptors.
6. Program the Transmit and Receive Ring length registers (DMA_CH(#i)_TXDESC_RING_LENGTH (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) and DMA_CH(#i)_RXDESC_RING_LENGTH (for i = 0; i <=

DWC_EQOS_NUM_DMA_RX_CH-1)). The ring length programmed must be at least 4.

7. Initialize receive and transmit descriptor list address with the base address of the transmit and receive descriptor (DMA_CH(#i)_TXDESC_LIST_ADDRESS (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1), DMA_CH(#i)_TXDESC_LIST_ADDRESS (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1)). In addition, you must program the Transmit and Receive tail pointer registers indicating to the DMA about the available descriptors (DMA_CH(#i)_TXDESC_TAIL_POINTER (for i = 0; i <= DWC_EQOS_NUM_D-MA_TX_CH-1) and DMA_CH(#i)_RXDESC_TAIL_POINTER (for i = 0; i <= DWC_EQOS_NUM_D-MA_RX_CH-1)).
8. Program the following fields to initialize the mode of operation in the MTL_TXQ0_OPERATION_MODE register:
   a. Transmit Store And Forward (TSF)
   b. Transmit Threshold Control (TTC)
   c. Transmit Queue Enable (TXQEN) to value 2‘b10 to enable Transmit Queue0
   d. Transmit Queue Size (TQS)
9. Enable the interrupts by programming the DMA_CH(#i)_INTERRUPT_ENABLE (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register.
10. Repeat steps 4 through 9 for all additional channels of AV feature.
11. Program the CBS control register, idleSlope, sendSlope, hiCredit, and loCredit registers of the AV Queues.
12. Start the Receive and Transmit DMA by setting bit 0 of the DMA_CH(#i)_TX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register and bit 0 of the DMA_CH(#i)_RX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1) register.

11.7.5.6.2 Enabling Slot Number Checking

You can enable slot number checking to specify the intervals at which the DMA channels mapped to AV Queues fetch the frames from the AHB/AXI system bus.

These steps should be completed after step 11 and before step 12 of "Initializing DMA".

You can use the slot number check feature to specify the intervals at which the DMA Channels mapped to AV Queues fetch the frames from the AHB/AXI system bus. This feature is useful for a uniform and periodic transfer of the AV traffic from the host memory. The feature is available only when you enable timestamping and program the MAC_SUB_SECOND_INCREMENT register. To enable the slot number checking:

1. Enable timestamping by following the steps described in "Initialization Guidelines for System Time Generation".
2. Make sure that the SLOTNUM field (bits 22:19) of TDES3 Normal Descriptor (Read Format) contains a valid slot number. You can read the current reference slot number from the DMA_CH(#i)_Slot_Function_Control_Status (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1).
3. Set Bit 0 (ESC) of the Slot Function Control and Status register of a channel to enable the slot number checking.

11.7.5.6.3 Enabling Average Bits Per Slot Reporting

You can enable the reporting of the average bits that are transmitted in a slot.

The CBS Status register of the additional AV channels provides information about the average bits that are transmitted in a slot. The software can asynchronously read this register to retrieve information about the average bits transmitted per slot. To enable average bits per slot reporting:

1. Enable timestamping by following the steps described in the Initialization Guidelines for System Time Generation.
2. Program Bits [6:4], SLC, of the MTL_TXQ[n]_ETS_CONTROL register of a channel with the number of slots over which the average transmitted bits per slot need to be computed.
3. Enable Bit 9(ABPSSIE) of the MTL_Q[n]_INTERRUPT_CONTROL_STATUS register of a channel to generate the average bits per slot interrupt.

NOTE

- The frequency of this interrupt depends on the value programmed in step 2. For example, when you program value 0 in the SLC field, the interrupt is generated at every 125 microsecond.
- When not required, you can disable this interrupt to stop the interrupt flooding.

4. Read Bits [16:0], ABS, from the MTL_TXQ[n]_ETS_STATUS register of a channel on each interrupt.

NOTE

The software can read the ABS bits in polling mode even if the ABPSIE bit is not enabled. When high, bit 1 (ABPSIS) of the MTL_TXQ[n]_ETS_STATUS register indicates that a new value is updated in the ABS field.

11.7.5.6.4 Disabling Flow Control for AV Enabled Queues

11.7.5.6.4.1 Transmit Flow

Program the EHFC (Enable Hardware Flow Control) bit of corresponding Rx Queue's MTL_RXQ[n]_OPERATION_MODE register to 0.

11.7.5.6.4.2 Receive Flow Control

Program the PSTQ[n] field corresponding to AV enabled Tx Queue in MAC_TXQ_PRTY_MAP0/1 register to 0.

11.7.5.6.5 Programming Guidelines for Flexible Pulse-Per-Second Output

After you enable the Flexible Pulse-Per-Second Output feature in the module controller, you can perform the following tasks:

- Generating Single Pulse on PPS
- Generating Next Pulse on PPS
- Generating a Pulse Train on PPS
- Generating an Interrupt without Affecting the PPS

11.7.5.6.5.1 Generating Single Pulse on PPS

To generate single pulse on PPS:

1. Program 11 or 10 (for interrupt) in Bits [6:5], TRGTMODSEL, of MAC_PPS_CONTROL Register. This instructs the MAC to use the Target Time registers (MAC_PPS0_TARGET_TIME_SECONDS and MAC_PPS0_TARGET_TIME_NANOSECONDS) for start time of PPS signal output.
2. Program the start time value in the Target Time registers (MAC_PPS0_TARGET_TIME_SECONDS and MAC_PPS0_TARGET_TIME_NANOSECONDS).
3. Program the width of the PPS signal output in MAC_PPS(#i)_WIDTH (for i = 0; i <= 3) Register.
4. Program Bits [3:0], PPSCMD, of MAC_PPS_CONTROL to 0001. This instructs the MAC to generate single pulse on the PPS signal output at the time programmed in the Target Time registers.

11.7.5.6.5.2 Generating Next Pulse on PPS

When the PPSCMD is executed (PPSCMD bits = 0), you can cancel the pulse generation by giving the Cancel Start Command (PPSCMD=0011) before the programmed start time elapses. You can also program the behavior of the next pulse in advance.

To program the next pulse:

1. Program the start time for the next pulse in the Target Time registers. This time should be more than the time at which the falling edge occurs for the previous pulse.
2. Program the width of the next PPS signal output in MAC_PPS(#i)_WIDTH (for i = 0; i <= DWC_EQOS_PPS_OUT_NUM-1).
3. Program Bits [3:0], PPSCMD, of MAC_PPS_CONTROL to generate a single pulse after the time at which the previous pulse is de-asserted. This instructs the MAC to generate single pulse on the PPS signal output, at the time programmed in Target Time registers.

   If you give this command before the previous pulse becomes low, then the new command overwrites the previous command and the QOS may generate only 1 extended pulse.

11.7.5.6.5.3 Generating a Pulse Train on PPS

To generate a pulse train on PPS:

1. Program 11 or 10 (for interrupt) in Bits [6:5], TRGTMODSEL, of MAC_PPS_Control Register. This instructs the MAC to use the Target Time registers for start time of the PPS signal output.
2. Program the start time value in the Target Time registers.
3. Program the interval value between the train of pulses on the PPS signal output in MAC_PPS(#i)_INTERVAL (for i = 0; i <= DWC_EQOS_PPS_OUT_NUM-1) Register.
4. Program the width of the PPS signal output in MAC_PPS(#i)_WIDTH (for i = 0; i <= DWC_EQOS_PPS_OUT_NUM-1) Register.
5. Program Bits[3:0], PPSCMD, of MAC_PPS_CONTROL Register to 0010. This instructs the MAC to generate train of pulses on the PPS signal output with start time programmed in Target Time registers.

   By default, the PPS pulse train is free-running unless stopped by ‘STOP Pulse train at time’ or ‘STOP Pulse Train immediately’ commands.

6. Program the stop value in the Target Time registers. Ensure that Bit 31 (TSTRBUSY) of MAC_PPS(#i)_TARGET_TIME_NANOSECONDS (for i = 0; i <= DWC_EQOS_PPS_OUT_NUM-1) Register is reset before programming the Target Time registers again.
7. Program the PPSCMD field (bit 3:0) of MAC_PPS_CONTROL to 0100. This stops the train of pulses on PPS signal output after the programmed stop time specified in Step 6 elapses.

   You can stop the pulse train at any time by programming 0101 in the PPSCMD field. Similarly, you can cancel the Stop Pulse train command (given in Step 7) by programming 0110 in the PPSCMD field before the time (programmed in Step 6) elapses. You can cancel the pulse train generation by programming 0011 in the PPSCMD field before the programmed start time (in Step 2) elapses.

11.7.5.6.5.4 Generating an Interrupt without Affecting the PPS

You can use bits[6:5], TRGTMODSEL, of the MAC_PPS_CONTROL Register to program the Target Time registers to do any one of the following:

- Generate only interrupts.
- Generate interrupts and the PPS start and stop time.
- Generate only PPS start and stop time.

To program the Target Time registers to generate only interrupt events:

1. Program 00 (for interrupt) in Bits [6:5], TRGTMODSEL, of MAC_PPS_CONTROL Register. This instructs the MAC to use the Target Time registers for target time interrupt.
2. Program a target time value in the Target Time registers. This instructs the MAC to generate an interrupt when the target time elapses.

   If Bits [6:5], TRGTMODSEL, are changed (for example, to control the PPS), then the interrupt generation is overwritten with the new mode and new programmed Target Time register value.

11.7.5.6.6 Programming Guidelines for VLAN Filtering on Receive

To program VLAN filtering on receive:

1. Program MAC_VLAN_TAG register for the following bit to select the filtering method:

a. ETV: Enable 12-Bit VLAN Tag Comparison or 16-bit VLAN Tag comparison.
b. VTHM: VLAN Tag Hash Table Match Enable.
c. ERIVLT: Enable inner VLAN Tag or outer VLAN Tag (to enable the inner or outer VLAN Tag filtering, Double VLAN Processing should enabled by setting EDVLP)
d. ERSVLM: Enable Receive S-VLAN Match or C-VLAN match (for S-VLAN processing to be enabled, set ESVL)
e. DOVLTC: Ignores VLAN Type for Tag Match
f. VTIM: to enable VLAN Tag Inverse Match instead of the normal VLAN Tag matching

2. Program VL of MAC_VLAN_Tag Register for the 12-bit or 16-bit VLAN tag.
3. If Hash filtering of VLAN tag is enabled, program MAC_VLAN_HASH_TABLE Register. When ETV bit is reset, upper 4 bits of the calculated CRC-32 of VLAN Tag are inverted and used to index the content of MAC_VLAN_Hash_Table register. When ETV bit is set, upper 4 bits of calculated CRC-32 of VLAN Tag are used to index the content of MAC_VLAN_HASH_TABLE register. For example, when ETV bit is set, a hash value of 4b'1000 selects Bit 8 of the VLAN Hash table. When ETV bit is reset a hash value of 4'b1000 selects Bit 7 of the VLAN Hash table.

11.7.5.6.7 Programming Guidelines for Extended VLAN Filtering and Routing on Receive

For the indirect access of the per VLAN Tag registers, follow these steps.

- Write

  a. Write the required data into the MAC VLAN Tag Data register.

  b. Program the VLAN Tag Control Register's OFS field with the required Filter Register's offset and command type to the CT field. For a write command, set this bit to 0.

  c. Write 1 to the OB field and wait till the OB bit is reset to do the next write. This will guarantee that the appropriate VLAN Tag Filter Register has been programmed.

- Read

  a. Program the VLAN Tag Control Register's OFS Field with the required register's offset and command type to the CT field. For a read command, set this bit to 1.

  b. Write 1 to the OB field and wait till the OB bit is reset. The appropriate VLAN Tag Filter register's value is available in the MAC VLAN Tag Data Register.

11.7.5.6.8 Programming Sequence for Queue/Channel Based VLAN Inclusion Register

To program the queue/channel-based VLAN inclusion register:

NOTE

Indirect VLAN include registers are not accessed while setting the CBTI bit.

1. Set the CBTI bit of MAC_VLAN_INCL register to 1, to enable queue/channel based VLAN Tag insertion on all transmitted packets. This bit must be set before any indirect access to the queue/channel specific MAC_VLAN_INCL(#i) register.
2. Program the VLAN Tag and VLAN Type to be inserted in packets from a particular queue/channel in VLT and CSVL fields of MAC_VLAN_INCL register; corresponding offset address in ADDR field (0 for queue/channel 0, 1 for queue/channel 1, and so on) must be set. Set the RDWR bit to 0 to indicate write access. The write to byte 0 (byte 3 in Big Endian mode) of MAC_VLAN_INCL register initiates access to indirect access MAC_VLAN_INCL(#i) register.
3. The BUSY bit of MAC_VLAN_INCL register is set by the module to indicate the progress of access to indirect access MAC_VLAN_INCL(#i) register. On completion of the access, the BUSY bit is cleared. The application must not attempt subsequent access to MAC_VLAN_INCL(#i) register when the BUSY bit is 1.
4. Repeat step 2 and step 3 to program VLAN Tag and VLAN Type to be inserted in packets from the remaining queues/channels. The application must ensure that the required VLAN Tag and VLAN Type for all the queues/channels are programmed; otherwise unintended VLAN Tag and VLAN Type might be inserted.

11.7.5.7 Programming Guidelines for EST

Program the gate control values and time intervals in the Software Owned Gate Control List (SWOL) along with the other EST related registers that are described in ENET_QOS register descriptions to appropriate values. The following sub-sections provide step by step details for programming the GCL and the other EST related registers.

11.7.5.7.1 Programming the GCL and GCL Linked Registers

To program the Gate Control List (GCL) and the four other registers implemented per GCL:

1. The GCL and the four other GCL-linked registers should be accessed via indirect addressing using the MTL_EST_GCL_CONTROL and MTL_EST_GCL_DATA registers. The SWOL bit of the MTL_EST_STATUS register indicates if GCL0 or GCL1 is owned by software.

2. To program the GCL, write the 32-bit write data to MTL_EST_GCL_DATA register. Then program TL_EST_GCL_CONTROL register to write the write address and other control information.
3. In the MTL_EST_GCL_DATA register, Write Data consists of up to 8 bits (5 configurable, 3 reserved) of Gate Controls and up to 24 bits (configurable) of Time Interval. Gate Close is indicated by programming a '0' and Gate Open is indicated by programming a '1'. For a 4-TC and 20-bit Time Interval Configuration, the data width is 24-bits and the remaining 8-bits are reserved/read-only. The data should be written in the following format.

   {8'h0, TC3, TC2, TC1, TC0, 20-bit Time Interval} where TCx = 0 or 1.

4. MTL_EST_GCL_CONTROL register: Program the SRWO bit to 1 (to start a Write Op) and program the address and R/W fields appropriately.
5. Poll and check for the SRWO bit to be cleared by hardware to indicate the completion of the previous operation before initiating a new R/W operation via the same indirect addressing mode.
6. Repeat steps 3, 4, 5 until the programming of the GCL is completed.
7. Using the same indirect addressing method described above, program the BTR, CTR, TER and LLR registers. Set the GCRR bit in the MTL_EST_GCL_CONTROL register appropriately. The GCRR bit interprets the address field as belonging to these registers (instead of the GCL).

After programming of the GCL and the related registers, program the MTL_EST_CONTROL register to allow hardware to own and process the GCL.When the List Length (as indicated in LLR) is 1, the associated Time Interval should be smaller than the value of the Cycle Time Register. Otherwise, an error is reported (as detailed in the Error handling section) as a single set of gate controls add no value in the TSN context.

NOTE

The time unit in all the GCL related registers is seconds and nanoseconds. In cases where internally generated PTP System Time is used, the nanoseconds field must be programmed to use the Digital Rollover mode. (TSCTRLSSR field of MAC_TIMESTAMP_CONTROL must be set to value '1')

11.7.5.7.2 Programming the EST Registers

After completing the steps mentioned in "Programming the GCL and GCL Linked Registers" , program the MTL_EST_CONTROL register.

1. Set the Current Time Offset Value and Time Internal Left Shift fields of the MTL_EST_CONTROL register appropriately. Also, set the Enable EST (EEST) and Switch to SWOL (SWOL) bits.

NOTE

The CTOV recommendations for GMII for SPRAM configurations are:

- 96 * Tx clock period, for 32 and 64 bit data width configurations.
- 128 * Tx clock period for 128 bit data width configurations.

The CTOV recommendations for GMII for non-SPRAM configurations are:

- 30 * Tx clock period, when SA/VLAN insertion is enabled.
- 22 * Tx clock period, when SA/VLAN insertion is not enabled.

2. This will enable the hardware to own and process the new GCL and make a switch to the new GCL at the BTR value. The hardware provides an interrupt (if enabled) when the switch to the new list happens.
3. Appropriate action has to be taken by software to address any other interrupts (explained in the Inter-rupts section) received during the hardware execution of the GCL.

Set the SSWL (Switch to Software Owned List) bit in the MTL_EST_CONTROL register to handoff to hardware. Software is not allowed to write to the GCL and GCL linked registers when the SSWL bit is set, because the new GCL might be in use by the hardware.

The SSWL is reset/cleared by the hardware when it successfully switches to the new list. The hardware also flips the SWOL bit to indicate the new GCL that the software owns.

To install a new GCL, program the GCL it owns (indicated by SWOL bit) as described in "Programming the GCL and GCL Linked Registers" and then program the MTL_EST_CONTROL register as described above. Ensure that the new BTR is set to an appropriate value to avoid BTRError that might need software intervention in some cases.

To avoid transmission overruns, the packet length (frame size) information should be available at all times. Therefore, in the DMA configurations, program the packet length in the first descriptor of every Tx frame. Similarly, in the MTL configuration, provide the packet length in the control word.

NOTE

To avoid transmission overruns, the Packet Length provided in the Transmit Descriptor should account for the SA and VLAN insertion, if applicable, that is, Packet Length should represent the actual packet length on the Ethernet line.

11.7.5.8 Programming the Launch Time in Time-Based Scheduling

The Launch Time is programmed in the Enhanced Normal Transmit Descriptors in DMA configurations and is driven as a control word in MTL configurations as follows:

The OSTC and Launch Time features are mutually exclusive and should not be used together; in case of a conflict (if OSTC = LTV = 1 in MTL configuration), priority is given to OSTC and the Launch Time is ignored.

In DMA configuration, if a context descriptor is received with a valid OSTC values immediately before receiving a first normal descriptor with LTV bit set, then the LTV is ignored.

11.7.5.9 Programming Guidelines for Media Clock Generation and Recovery

11.7.5.9.1 Programming Guidelines for Media Clock Generation

1. To set the PPS instance in Media Clock generation mode, program the appropriate Presentation Time Control (supported generation modes "1001-1011") to PPSCTRL_PPSCMD (for 0th instance)/PPSCMD#i (for 1,2,3 instances) of MAC_PPS_CONTROL register.
2. Based on the selected PPS instance, application must drive the appropriate trigger signal to the corresponding mcg_pst_trig_i[#i].
3. Based on the programmed mode, DUT captures the timestamp and programs it in the MAC_PPS(#i)_TARGET_TIME_SECONDS register. Then, mcgr_dma_req_o[#i] is set.
4. For every request generated by the DUT, the module reads the corresponding MAC_PPS(#i)_TARGET_TIME_SECONDS register and acknowledge the read request by asserting the corresponding mcgr_dma_ack_i[#i].

11.7.5.9.2 Programming Guidelines for Media Clock Recovery

1. Enable the Current Presentation Time (CPT) Counter by setting the PTGE field of MAC_TIMESTAMP_CONTROL register. In addition to programming the initialization values for the System Time, update the MAC_PRESN_TIME_UPDT register with the equivalent Presentation Time init value. Only then the TSINIT field of MAC_TIMESTAMP_CONTROL register is set.
2. The Increment values used for the System time is also used for the Current Presentation time. This is because, the increment value is in Sub-seconds and Sub-nanoseconds.
3. When an update is needed, along with programming the update values for the System Time, the MAC_PRESN_TIME_UPDT register should be updated with the equivalent Presentation Time update value (32-bit ns). Finally, TSUPDT field of MAC_TIMESTAMP_CONTROL register must be set for the update to happen.
4. Program the MCGREN#i of MAC_PPS_CONTROL register to enable the PPS instance to operate in MCGR mode.
5. To set the PPS instance in Media Clock recovery mode, program the appropriate Presentation Time Control (supported recovery modes "0001 - 0011") to PPSCTRL_PPSCMD (for 0th instance)/PPSCMD#i (for 1,2,3 instances) of MAC_PPS_Control register.
6. After PPS instance is set in recovery mode, corresponding mcgr_dma_req_o[#i] is set by the DUT. For every request generated by the DUT, program the target presentation time into the MAC_PPS(#i)_TARGET_TIME_SECONDS register and acknowledge the request by asserting the corresponding mcgr_d-ma_ack_i[#i]. Acknowledgment can be given before/after programming the Target presentation time.
7. Observe the recovered media clock on the corresponding ptp_pps_o[#i].

11.7.5.10 Programming Guidelines for ECC Protection for Memories

1. Enable the ECC feature for the respective memory by setting the appropriate bit in MTL_ECC_CONTROL register.
2. A correctable interrupt (sbd_sfty_ce_intr_o) or an uncorrectable interrupt (sbd_sfty_ue_intr_o) are generated to indicate to the application if any correctable/ uncorrectable/address errors are detected. Appropriate status is indicated in the DMA/MTL_ECC_INTERRUPT_STATUS register.

3. Debug mode is provided for each memory to specify errors.

NOTE

Enable the ECC feature before the traffic is online (or after the reset), otherwise false interrupts might be triggered.

11.7.5.10.1 Programming Guidelines for ECC Hardware Error Injection (Debug Mode)

1. Enable the ECC error injection feature for MTL TX/RX and DMA TSO, by setting the appropriate bit in the MTL_DBG_CTL register.
2. Enable the ECC error injection feature for the EST memory by setting the appropriate bit in the EST_GCL_CONTROL register.
3. Enable the ECC error injection for RX Parser memory by setting the appropriate bit in the MTL_RXP_INDIRECT_ACC_CONTROL_STATUS register. To access Rx Parser memory in debug mode, disable the Rx Parser feature by setting the FRPE bit of MTL_OPERATION_MODE register to 0.
4. Where multiple CSR writes are required, for writing single data word into the memory, the application should ensure that all the CSR writes corresponding to one memory write should have the same value for the error injection control word.

11.7.5.11 Programming Guidelines for On-Chip Datapath Parity Protection

Follow the below steps for on-chip datapath parity protection:

1. Set the EDPP bit of MTL_DPP_CONTROL register to enable the generation of parity and detection of parity Mismatch on datapath.
2. An uncorrectable interrupt (sbd_sfty_ue_intr_o) is generated to indicate to the application whether any parity Mismatches are detected and the appropriate status will be indicated in DMA/MTL_SAFETY_INTERRUPT_STATUS and MAC_DPP_FSM_INTERRUPT_STATUS register.
3. Debug mode is supported for each parity generator, to insert error.

NOTE

To avoid false safety interrupts, enable the datapath parity protection before the start of Rx or Tx traffic.

11.7.5.12 Programming Guidelines for FSM Parity and Timeout

To program the FSM parity and timeout:

1. Set the PRTYEN and TMOUTEN fields of MAC_FSM_CONTROL register to enable the FSM parity and timeout feature respectively.
2. For parity error detection, force the FSMs to enter into a state with even number of 1s and wait for the safety uncorrectable interrupt to be set with FSMPES of MAC_DPP_FSM_INTERRUPT_STATUS.
3. For Error Injection mode in FSM parity, select the appropriate clock domain for which parity error injection should be set using the [23:16] bits of MAC_FSM_CONTROL register. Wait for the safety uncorrectable interrupt to be set with FSMPES of MAC_DPP_FSM_INTERRUPT_STATUS.
4. For FSM Timeouts, program the large and normal mode values indicated by LTMRMD and NTMRMD of MAC_FSM_ACT_TIMER register.
5. Select large or normal mode tic generation per clock domain by using the [31:24] bits of MAC_FSM_CONTROL Register.
6. Set the TMR field of the MAC_ACT_TIMER field with the appropriate number of CSR clock cycles used to generate 1-microsecond tic.
7. Force the FSM in a particular clock domain to be in an active state for Timeout duration (for example, T-2T). Check for the safety interrupt and the status bit for the corresponding clock domain getting set in [15:8] bits of MAC_DPP_FSM_INTERRUPT_STATUS register.
8. For Error Injection mode in FSM timeout, select the appropriate clock domain for which timeout error injection must be set using the [15:8] bits of MAC_FSM_CONTROL register. Wait for the safety uncorrectable interrupt along with the corresponding error status set in MAC_DPP_FSM_INTERRUPT_STATUS.
9. For application/CSR interface timeout, only normal mode tic generation is used. Based on the timeouts of the configured interface, appropriate fields (MSTTES, SLVTES) of MAC_DPP_FSM_INTERRUPT_STATUS are set along with the safety interrupt.

11.7.6 Memory Map and register definition

This section includes the ENET_QOS module memory map and detailed descriptions of all registers.

11.7.6.1 ENET_QOS register descriptions

11.7.6.1.1 ENET_QOS memory map

ENET_QOS base address: 30BF_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	MAC Configuration Register (MAC_CONFIGURATION)	32	RW	0000_0000
4	MAC Extended Configuration Register (MAC_EXT_CONFIGURATION)	32	RW	0000_0000
8	MAC Packet Filter (MAC_PACKET_FILTER)	32	RW	0000_0000
C	Watchdog Timeout (MAC_WATCHDOG_TIMEOUT)	32	RW	0000_0000
10	MAC Hash Table Register 0 (MAC_HASH_TABLE_REG0)	32	RW	0000_0000
14	MAC Hash Table Register 1 (MAC_HASH_TABLE_REG1)	32	RW	0000_0000
50	MAC VLAN Tag Control (MAC_VLAN_TAG_CTRL)	32	RW	0000_0000
54	MAC VLAN Tag Data (MAC_VLAN_TAG_DATA)	32	RW	0000_0000
58	MAC VLAN Hash Table (MAC_VLAN_HASH_TABLE)	32	RW	0000_0000
60	VLAN Tag Inclusion or Replacement (MAC_VLAN_INCL)	32	RW	0000_0000
64	MAC Inner VLAN Tag Inclusion or Replacement (MAC_INNER_VLAN_INCL)	32	RW	0000_0000
70	MAC Q0 Tx Flow Control (MAC_Q0_TX_FLOW_CTRL)	32	RW	0000_0000
74	MAC Q1 Tx Flow Control (MAC_Q1_TX_FLOW_CTRL)	32	RW	0000_0000
78	MAC Q2 Tx Flow Control (MAC_Q2_TX_FLOW_CTRL)	32	RW	0000_0000
7C	MAC Q3 Tx Flow Control (MAC_Q3_TX_FLOW_CTRL)	32	RW	0000_0000
80	MAC Q4 Tx Flow Control (MAC_Q4_TX_FLOW_CTRL)	32	RW	0000_0000
90	MAC Rx Flow Control (MAC_RX_FLOW_CTRL)	32	RW	0000_0000
94	Receive Queue Control 4 (MAC_RXQ_CTRL4)	32	RW	0000_0000
98	Transmit Queue Priority Mapping 0 (MAC_TXQ_PRTY_MAP0)	32	RW	0000_0000
9C	Transmit Queue Priority Mapping 1 (MAC_TXQ_PRTY_MAP1)	32	RW	0000_0000
A0	Receive Queue Control 0 (MAC_RXQ_CTRL0)	32	RW	0000_0000
A4	Receive Queue Control 1 (MAC_RXQ_CTRL1)	32	RW	0000_0000
A8	Receive Queue Control 2 (MAC_RXQ_CTRL2)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
AC	Receive Queue Control 3 (MAC_RXQ_CTRL3)	32	RW	0000_0000
B0	Interrupt Status (MAC_INTERRUPT_STATUS)	32	RO	0000_0000
B4	Interrupt Enable (MAC_INTERRUPT_ENABLE)	32	RW	0000_0000
B8	Receive Transmit Status (MAC_RX_TX_STATUS)	32	RO	0000_0000
C0	PMT Control and Status (MAC_PMT_CONTROL_STATUS)	32	RW	0000_0000
C4	Remote Wakeup Filter (MAC_RWK_PACKET_FILTER)	32	RW	0000_0000
D0	LPI Control and Status (MAC_LPI_CONTROL_STATUS)	32	RW	0000_0000
D4	LPI Timers Control (MAC_LPI_TIMERS_CONTROL)	32	RW	03E8_0000
D8	Tx LPI Entry Timer Control (MAC_LPI_ENTRY_TIMER)	32	RW	0000_0000
DC	One-microsecond Reference Timer (MAC_ONEUS_TIC_COUNTER)	32	RW	0000_0063
F8	PHY Interface Control and Status (MAC_PHYIF_CONTROL_STATUS)	32	RW	0000_0000
110	MAC Version (MAC_VERSION)	32	RO	0000_1051
114	MAC Debug (MAC_DEBUG)	32	RO	0000_0000
11C	Optional Features or Functions 0 (MAC_HW_FEATURE0)	32	RO	0EFD_71F7
120	Optional Features or Functions 1 (MAC_HW_FEATURE1)	32	RO	4199_79A6
124	Optional Features or Functions 2 (MAC_HW_FEATURE2)	32	RO	4410_4104
128	Optional Features or Functions 3 (MAC_HW_FEATURE3)	32	RO	0C39_5632
200	MDIO Address (MAC_MDIO_ADDRESS)	32	RW	0000_0000
204	MAC MDIO Data (MAC_MDIO_DATA)	32	RW	0000_0000
230	CSR Software Control (MAC_CSR_SW_CTRL)	32	RW	0000_0000
234	Frame Preemption Control (MAC_FPE_CTRL_STS)	32	RW	0000_0000
240	32-bit Binary Rollover Equivalent Time (MAC_PRESN_TIME_NS)	32	RO	0000_0000
244	MAC 1722 Presentation Time (MAC_PRESN_TIME_UPDT)	32	RW	0000_0000
300	MAC Address0 High (MAC_ADDRESS0_HIGH)	32	RW	8000_FFFF
304	MAC Address0 Low (MAC_ADDRESS0_LOW)	32	RW	FFFF_FFFF
308	MAC Address1 High (MAC_ADDRESS1_HIGH)	32	RW	0000_FFFF
30C	MAC Address1 Low (MAC_ADDRESS1_LOW)	32	RW	FFFF_FFFF
310	MAC Address2 High (MAC_ADDRESS2_HIGH)	32	RW	0000_FFFF
314	MAC Address2 Low (MAC_ADDRESS2_LOW)	32	RW	FFFF_FFFF
318	MAC Address3 High (MAC_ADDRESS3_HIGH)	32	RW	0000_FFFF
31C	MAC Address3 Low (MAC_ADDRESS3_LOW)	32	RW	FFFF_FFFF
320	MAC Address4 High (MAC_ADDRESS4_HIGH)	32	RW	0000_FFFF
324	MAC Address4 Low (MAC_ADDRESS4_LOW)	32	RW	FFFF_FFFF
328	MAC Address5 High (MAC_ADDRESS5_HIGH)	32	RW	0000_FFFF
32C	MAC Address5 Low (MAC_ADDRESS5_LOW)	32	RW	FFFF_FFFF
330	MAC Address6 High (MAC_ADDRESS6_HIGH)	32	RW	0000_FFFF
334	MAC Address6 Low (MAC_ADDRESS6_LOW)	32	RW	FFFF_FFFF
338	MAC Address7 High (MAC_ADDRESS7_HIGH)	32	RW	0000_FFFF
33C	MAC Address7 Low (MAC_ADDRESS7_LOW)	32	RW	FFFF_FFFF

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
340	MAC Address8 High (MAC_ADDRESS8_HIGH)	32	RW	0000_FFFF
344	MAC Address8 Low (MAC_ADDRESS8_LOW)	32	RW	FFFF_FFFF
348	MAC Address9 High (MAC_ADDRESS9_HIGH)	32	RW	0000_FFFF
34C	MAC Address9 Low (MAC_ADDRESS9_LOW)	32	RW	FFFF_FFFF
350	MAC Address10 High (MAC_ADDRESS10_HIGH)	32	RW	0000_FFFF
354	MAC Address10 Low (MAC_ADDRESS10_LOW)	32	RW	FFFF_FFFF
358	MAC Address11 High (MAC_ADDRESS11_HIGH)	32	RW	0000_FFFF
35C	MAC Address11 Low (MAC_ADDRESS11_LOW)	32	RW	FFFF_FFFF
360	MAC Address12 High (MAC_ADDRESS12_HIGH)	32	RW	0000_FFFF
364	MAC Address12 Low (MAC_ADDRESS12_LOW)	32	RW	FFFF_FFFF
368	MAC Address13 High (MAC_ADDRESS13_HIGH)	32	RW	0000_FFFF
36C	MAC Address13 Low (MAC_ADDRESS13_LOW)	32	RW	FFFF_FFFF
370	MAC Address14 High (MAC_ADDRESS14_HIGH)	32	RW	0000_FFFF
374	MAC Address14 Low (MAC_ADDRESS14_LOW)	32	RW	FFFF_FFFF
378	MAC Address15 High (MAC_ADDRESS15_HIGH)	32	RW	0000_FFFF
37C	MAC Address15 Low (MAC_ADDRESS15_LOW)	32	RW	FFFF_FFFF
380	MAC Address16 High (MAC_ADDRESS16_HIGH)	32	RW	0000_FFFF
384	MAC Address16 Low (MAC_ADDRESS16_LOW)	32	RW	FFFF_FFFF
388	MAC Address17 High (MAC_ADDRESS17_HIGH)	32	RW	0000_FFFF
38C	MAC Address17 Low (MAC_ADDRESS17_LOW)	32	RW	FFFF_FFFF
390	MAC Address18 High (MAC_ADDRESS18_HIGH)	32	RW	0000_FFFF
394	MAC Address18 Low (MAC_ADDRESS18_LOW)	32	RW	FFFF_FFFF
398	MAC Address19 High (MAC_ADDRESS19_HIGH)	32	RW	0000_FFFF
39C	MAC Address19 Low (MAC_ADDRESS19_LOW)	32	RW	FFFF_FFFF
3A0	MAC Address20 High (MAC_ADDRESS20_HIGH)	32	RW	0000_FFFF
3A4	MAC Address20 Low (MAC_ADDRESS20_LOW)	32	RW	FFFF_FFFF
3A8	MAC Address21 High (MAC_ADDRESS21_HIGH)	32	RW	0000_FFFF
3AC	MAC Address21 Low (MAC_ADDRESS21_LOW)	32	RW	FFFF_FFFF
3B0	MAC Address22 High (MAC_ADDRESS22_HIGH)	32	RW	0000_FFFF
3B4	MAC Address22 Low (MAC_ADDRESS22_LOW)	32	RW	FFFF_FFFF
3B8	MAC Address23 High (MAC_ADDRESS23_HIGH)	32	RW	0000_FFFF
3BC	MAC Address23 Low (MAC_ADDRESS23_LOW)	32	RW	FFFF_FFFF
3C0	MAC Address24 High (MAC_ADDRESS24_HIGH)	32	RW	0000_FFFF
3C4	MAC Address24 Low (MAC_ADDRESS24_LOW)	32	RW	FFFF_FFFF
3C8	MAC Address25 High (MAC_ADDRESS25_HIGH)	32	RW	0000_FFFF
3CC	MAC Address25 Low (MAC_ADDRESS25_LOW)	32	RW	FFFF_FFFF
3D0	MAC Address26 High (MAC_ADDRESS26_HIGH)	32	RW	0000_FFFF
3D4	MAC Address26 Low (MAC_ADDRESS26_LOW)	32	RW	FFFF_FFFF
3D8	MAC Address27 High (MAC_ADDRESS27_HIGH)	32	RW	0000_FFFF
3DC	MAC Address27 Low (MAC_ADDRESS27_LOW)	32	RW	FFFF_FFFF

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
3E0	MAC Address28 High (MAC_ADDRESS28_HIGH)	32	RW	0000_FFFF
3E4	MAC Address28 Low (MAC_ADDRESS28_LOW)	32	RW	FFFF_FFFF
3E8	MAC Address29 High (MAC_ADDRESS29_HIGH)	32	RW	0000_FFFF
3EC	MAC Address29 Low (MAC_ADDRESS29_LOW)	32	RW	FFFF_FFFF
3F0	MAC Address30 High (MAC_ADDRESS30_HIGH)	32	RW	0000_FFFF
3F4	MAC Address30 Low (MAC_ADDRESS30_LOW)	32	RW	FFFF_FFFF
3F8	MAC Address31 High (MAC_ADDRESS31_HIGH)	32	RW	0000_FFFF
3FC	MAC Address31 Low (MAC_ADDRESS31_LOW)	32	RW	FFFF_FFFF
400	MAC Address32 High (MAC_ADDRESS32_HIGH)	32	RW	0000_FFFF
404	MAC Address32 Low (MAC_ADDRESS32_LOW)	32	RW	FFFF_FFFF
408	MAC Address33 High (MAC_ADDRESS33_HIGH)	32	RW	0000_FFFF
40C	MAC Address33 Low (MAC_ADDRESS33_LOW)	32	RW	FFFF_FFFF
410	MAC Address34 High (MAC_ADDRESS34_HIGH)	32	RW	0000_FFFF
414	MAC Address34 Low (MAC_ADDRESS34_LOW)	32	RW	FFFF_FFFF
418	MAC Address35 High (MAC_ADDRESS35_HIGH)	32	RW	0000_FFFF
41C	MAC Address35 Low (MAC_ADDRESS35_LOW)	32	RW	FFFF_FFFF
420	MAC Address36 High (MAC_ADDRESS36_HIGH)	32	RW	0000_FFFF
424	MAC Address36 Low (MAC_ADDRESS36_LOW)	32	RW	FFFF_FFFF
428	MAC Address37 High (MAC_ADDRESS37_HIGH)	32	RW	0000_FFFF
42C	MAC Address37 Low (MAC_ADDRESS37_LOW)	32	RW	FFFF_FFFF
430	MAC Address38 High (MAC_ADDRESS38_HIGH)	32	RW	0000_FFFF
434	MAC Address38 Low (MAC_ADDRESS38_LOW)	32	RW	FFFF_FFFF
438	MAC Address39 High (MAC_ADDRESS39_HIGH)	32	RW	0000_FFFF
43C	MAC Address39 Low (MAC_ADDRESS39_LOW)	32	RW	FFFF_FFFF
440	MAC Address40 High (MAC_ADDRESS40_HIGH)	32	RW	0000_FFFF
444	MAC Address40 Low (MAC_ADDRESS40_LOW)	32	RW	FFFF_FFFF
448	MAC Address41 High (MAC_ADDRESS41_HIGH)	32	RW	0000_FFFF
44C	MAC Address41 Low (MAC_ADDRESS41_LOW)	32	RW	FFFF_FFFF
450	MAC Address42 High (MAC_ADDRESS42_HIGH)	32	RW	0000_FFFF
454	MAC Address42 Low (MAC_ADDRESS42_LOW)	32	RW	FFFF_FFFF
458	MAC Address43 High (MAC_ADDRESS43_HIGH)	32	RW	0000_FFFF
45C	MAC Address43 Low (MAC_ADDRESS43_LOW)	32	RW	FFFF_FFFF
460	MAC Address44 High (MAC_ADDRESS44_HIGH)	32	RW	0000_FFFF
464	MAC Address44 Low (MAC_ADDRESS44_LOW)	32	RW	FFFF_FFFF
468	MAC Address45 High (MAC_ADDRESS45_HIGH)	32	RW	0000_FFFF
46C	MAC Address45 Low (MAC_ADDRESS45_LOW)	32	RW	FFFF_FFFF
470	MAC Address46 High (MAC_ADDRESS46_HIGH)	32	RW	0000_FFFF
474	MAC Address46 Low (MAC_ADDRESS46_LOW)	32	RW	FFFF_FFFF
478	MAC Address47 High (MAC_ADDRESS47_HIGH)	32	RW	0000_FFFF
47C	MAC Address47 Low (MAC_ADDRESS47_LOW)	32	RW	FFFF_FFFF

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
480	MAC Address48 High (MAC_ADDRESS48_HIGH)	32	RW	0000_FFFF
484	MAC Address48 Low (MAC_ADDRESS48_LOW)	32	RW	FFFF_FFFF
488	MAC Address49 High (MAC_ADDRESS49_HIGH)	32	RW	0000_FFFF
48C	MAC Address49 Low (MAC_ADDRESS49_LOW)	32	RW	FFFF_FFFF
490	MAC Address50 High (MAC_ADDRESS50_HIGH)	32	RW	0000_FFFF
494	MAC Address50 Low (MAC_ADDRESS50_LOW)	32	RW	FFFF_FFFF
498	MAC Address51 High (MAC_ADDRESS51_HIGH)	32	RW	0000_FFFF
49C	MAC Address51 Low (MAC_ADDRESS51_LOW)	32	RW	FFFF_FFFF
4A0	MAC Address52 High (MAC_ADDRESS52_HIGH)	32	RW	0000_FFFF
4A4	MAC Address52 Low (MAC_ADDRESS52_LOW)	32	RW	FFFF_FFFF
4A8	MAC Address53 High (MAC_ADDRESS53_HIGH)	32	RW	0000_FFFF
4AC	MAC Address53 Low (MAC_ADDRESS53_LOW)	32	RW	FFFF_FFFF
4B0	MAC Address54 High (MAC_ADDRESS54_HIGH)	32	RW	0000_FFFF
4B4	MAC Address54 Low (MAC_ADDRESS54_LOW)	32	RW	FFFF_FFFF
4B8	MAC Address55 High (MAC_ADDRESS55_HIGH)	32	RW	0000_FFFF
4BC	MAC Address55 Low (MAC_ADDRESS55_LOW)	32	RW	FFFF_FFFF
4C0	MAC Address56 High (MAC_ADDRESS56_HIGH)	32	RW	0000_FFFF
4C4	MAC Address56 Low (MAC_ADDRESS56_LOW)	32	RW	FFFF_FFFF
4C8	MAC Address57 High (MAC_ADDRESS57_HIGH)	32	RW	0000_FFFF
4CC	MAC Address57 Low (MAC_ADDRESS57_LOW)	32	RW	FFFF_FFFF
4D0	MAC Address58 High (MAC_ADDRESS58_HIGH)	32	RW	0000_FFFF
4D4	MAC Address58 Low (MAC_ADDRESS58_LOW)	32	RW	FFFF_FFFF
4D8	MAC Address59 High (MAC_ADDRESS59_HIGH)	32	RW	0000_FFFF
4DC	MAC Address59 Low (MAC_ADDRESS59_LOW)	32	RW	FFFF_FFFF
4E0	MAC Address60 High (MAC_ADDRESS60_HIGH)	32	RW	0000_FFFF
4E4	MAC Address60 Low (MAC_ADDRESS60_LOW)	32	RW	FFFF_FFFF
4E8	MAC Address61 High (MAC_ADDRESS61_HIGH)	32	RW	0000_FFFF
4EC	MAC Address61 Low (MAC_ADDRESS61_LOW)	32	RW	FFFF_FFFF
4F0	MAC Address62 High (MAC_ADDRESS62_HIGH)	32	RW	0000_FFFF
4F4	MAC Address62 Low (MAC_ADDRESS62_LOW)	32	RW	FFFF_FFFF
4F8	MAC Address63 High (MAC_ADDRESS63_HIGH)	32	RW	0000_FFFF
4FC	MAC Address63 Low (MAC_ADDRESS63_LOW)	32	RW	FFFF_FFFF
700	MMC Control (MAC_MMC_CONTROL)	32	RW	0000_0000
704	MMC Rx Interrupt (MAC_MMC_RX_INTERRUPT)	32	RO	0000_0000
708	MMC Tx Interrupt (MAC_MMC_TX_INTERRUPT)	32	RO	0000_0000
70C	MMC Rx Interrupt Mask (MAC_MMC_RX_INTERRUPT_MASK)	32	RW	0000_0000
710	MMC Tx Interrupt Mask (MAC_MMC_TX_INTERRUPT_MASK)	32	RW	0000_0000
714	Tx Octet Count Good and Bad (MAC_TX_OCTET_COUNT_GOOD_BAD)	32	RO	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
718	Tx Packet Count Good and Bad (MAC_TX_PACKET_COUNT_GOOD_BAD)	32	RO	0000_0000
71C	Tx Broadcast Packets Good (MAC_TX_BROADCAST_PACKETS_GOOD)	32	RO	0000_0000
720	Tx Multicast Packets Good (MAC_TX_MULTICAST_PACKETS_GOOD)	32	RO	0000_0000
724	Tx Good and Bad 64-Byte Packets (MAC_TX_64OCTETS_PACKETS_GOOD_BAD)	32	RO	0000_0000
728	Tx Good and Bad 65 to 127-Byte Packets (MAC_TX_65TO127OCTETS_PACKETS_GOOD_BAD)	32	RO	0000_0000
72C	Tx Good and Bad 128 to 255-Byte Packets (MAC_TX_128TO255OCTETS_PACKETS_GOOD_BAD)	32	RO	0000_0000
730	Tx Good and Bad 256 to 511-Byte Packets (MAC_TX_256TO511OCTETS_PACKETS_GOOD_BAD)	32	RO	0000_0000
734	Tx Good and Bad 512 to 1023-Byte Packets (MAC_TX_512TO1023OCTETS_PACKETS_GOOD_BAD)	32	RO	0000_0000
738	Tx Good and Bad 1024 to Max-Byte Packets (MAC_TX_1024TOMAXOCTETS_PACKETS_GOOD_BAD)	32	RO	0000_0000
73C	Good and Bad Unicast Packets Transmitted (MAC_TX_UNICAST_PACKETS_GOOD_BAD)	32	RO	0000_0000
740	Good and Bad Multicast Packets Transmitted (MAC_TX_MULTICAST_PACKETS_GOOD_BAD)	32	RO	0000_0000
744	Good and Bad Broadcast Packets Transmitted (MAC_TX_BROADCAST_PACKETS_GOOD_BAD)	32	RO	0000_0000
748	Tx Packets Aborted By Underflow Error (MAC_TX_UNDERFLOW_ERROR_PACKETS)	32	RO	0000_0000
74C	Single Collision Good Packets Transmitted (MAC_TX_SINGLE_COLLISION_GOOD_PACKETS)	32	RO	0000_0000
750	Multiple Collision Good Packets Transmitted (MAC_TX_MULTIPLE_COLLISION_GOOD_PACKETS)	32	RO	0000_0000
754	Deferred Packets Transmitted (MAC_TX_DEFERRED_PACKETS)	32	RO	0000_0000
758	Late Collision Packets Transmitted (MAC_TX_LATE_COLLISION_PACKETS)	32	RO	0000_0000
75C	Excessive Collision Packets Transmitted (MAC_TX_EXCESSIVE_COLLISION_PACKETS)	32	RO	0000_0000
760	Carrier Error Packets Transmitted (MAC_TX_CARRIER_ERROR_PACKETS)	32	RO	0000_0000
764	Bytes Transmitted in Good Packets (MAC_TX_OCTET_COUNT_GOOD)	32	RO	0000_0000
768	Good Packets Transmitted (MAC_TX_PACKET_COUNT_GOOD)	32	RO	0000_0000
76C	Packets Aborted By Excessive Deferral Error (MAC_TX_EXCESSIVE_DEFERRAL_ERROR)	32	RO	0000_0000
770	Pause Packets Transmitted (MAC_TX_PAUSE_PACKETS)	32	RO	0000_0000
774	Good VLAN Packets Transmitted (MAC_TX_VLAN_PACKETS_GOOD)	32	RO	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
778	Good Oversize Packets Transmitted (MAC_TX_OSIZE_PACKETS_GOOD)	32	RO	0000_0000
780	Good and Bad Packets Received (MAC_RX_PACKETS_COUNT_GOOD_BAD)	32	RO	0000_0000
784	Bytes in Good and Bad Packets Received (MAC_RX_OCTET_COUNT_GOOD_BAD)	32	RO	0000_0000
788	Bytes in Good Packets Received (MAC_RX_OCTET_COUNT_GOOD)	32	RO	0000_0000
78C	Good Broadcast Packets Received (MAC_RX_BROADCAST_PACKETS_GOOD)	32	RO	0000_0000
790	Good Multicast Packets Received (MAC_RX_MULTICAST_PACKETS_GOOD)	32	RO	0000_0000
794	CRC Error Packets Received (MAC_RX_CRC_ERROR_PACKETS)	32	RO	0000_0000
798	Alignment Error Packets Received (MAC_RX_ALIGNMENT_ERROR_PACKETS)	32	RO	0000_0000
79C	Runt Error Packets Received (MAC_RX_RUNT_ERROR_PACKETS)	32	RO	0000_0000
7A0	Jabber Error Packets Received (MAC_RX_JABBER_ERROR_PACKETS)	32	RO	0000_0000
7A4	Good Undersize Packets Received (MAC_RX_UNDERSIZE_PACKETS_GOOD)	32	RO	0000_0000
7A8	Good Oversize Packets Received (MAC_RX_OVERSIZE_PACKETS_GOOD)	32	RO	0000_0000
7AC	Good and Bad 64-Byte Packets Received (MAC_RX_64OCTETS_PACKETS_GOOD_BAD)	32	RO	0000_0000
7B0	Good and Bad 64-to-127 Byte Packets Received (MAC_RX_65TO127OCTETS_PACKETS_GOOD_BAD)	32	RO	0000_0000
7B4	Good and Bad 128-to-255 Byte Packets Received (MAC_RX_128TO255OCTETS_PACKETS_GOOD_BAD)	32	RO	0000_0000
7B8	Good and Bad 256-to-511 Byte Packets Received (MAC_RX_256TO511OCTETS_PACKETS_GOOD_BAD)	32	RO	0000_0000
7BC	Good and Bad 512-to-1023 Byte Packets Received (MAC_RX_512TO1023OCTETS_PACKETS_GOOD_BAD)	32	RO	0000_0000
7C0	Good and Bad 1024-to-Max Byte Packets Received (MAC_RX_1024TOMAXOCTETS_PACKETS_GOOD_BAD)	32	RO	0000_0000
7C4	Good Unicast Packets Received (MAC_RX_UNICAST_PACKETS_GOOD)	32	RO	0000_0000
7C8	Length Error Packets Received (MAC_RX_LENGTH_ERROR_PACKETS)	32	RO	0000_0000
7CC	Out-of-range Type Packets Received (MAC_RX_OUT_OF_RANGE_TYPE_PACKETS)	32	RO	0000_0000
7D0	Pause Packets Received (MAC_RX_PAUSE_PACKETS)	32	RO	0000_0000
7D4	Missed Packets Due to FIFO Overflow (MAC_RX_FIFO_OVERFLOW_PACKETS)	32	RO	0000_0000
7D8	Good and Bad VLAN Packets Received (MAC_RX_VLAN_PACKETS_GOOD_BAD)	32	RO	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
7DC	Watchdog Error Packets Received (MAC_RX_WATCHDOG_ERROR_PACKETS)	32	RO	0000_0000
7E0	Receive Error Packets Received (MAC_RX_RECEIVE_ERROR_PACKETS)	32	RO	0000_0000
7E4	Good Control Packets Received (MAC_RX_CONTROL_PACKETS_GOOD)	32	RO	0000_0000
7EC	Microseconds Tx LPI Asserted (MAC_TX_LPI_USEC_CNTR)	32	RO	0000_0000
7F0	Number of Times Tx LPI Asserted (MAC_TX_LPI_TRAN_CNTR)	32	RO	0000_0000
7F4	Microseconds Rx LPI Sampled (MAC_RX_LPI_USEC_CNTR)	32	RO	0000_0000
7F8	Number of Times Rx LPI Entered (MAC_RX_LPI_TRAN_CNTR)	32	RO	0000_0000
800	MMC IPC Receive Interrupt Mask (MAC_MMC_IPC_RX_INTERRUPT_MASK)	32	RW	0000_0000
808	MMC IPC Receive Interrupt (MAC_MMC_IPC_RX_INTERRUPT)	32	RO	0000_0000
810	Good IPv4 Datagrams Received (MAC_RXIPV4_GOOD_PACKETS)	32	RO	0000_0000
814	IPv4 Datagrams Received with Header Errors (MAC_RXIPV4_HEADER_ERROR_PACKETS)	32	RO	0000_0000
818	IPv4 Datagrams Received with No Payload (MAC_RXIPV4_NO_PAYLOAD_PACKETS)	32	RO	0000_0000
81C	IPv4 Datagrams Received with Fragmentation (MAC_RXIPV4_FRAGMENTED_PACKETS)	32	RO	0000_0000
820	IPv4 Datagrams Received with UDP Checksum Disabled (MAC_RXIPV4_UDP_CHECKSUM_DISABLED_PACKETS)	32	RO	0000_0000
824	Good IPv6 Datagrams Received (MAC_RXIPV6_GOOD_PACKETS)	32	RO	0000_0000
828	IPv6 Datagrams Received with Header Errors (MAC_RXIPV6_HEADER_ERROR_PACKETS)	32	RO	0000_0000
82C	IPv6 Datagrams Received with No Payload (MAC_RXIPV6_NO_PAYLOAD_PACKETS)	32	RO	0000_0000
830	IPv6 Datagrams Received with Good UDP (MAC_RXUDP_GOOD_PACKETS)	32	RO	0000_0000
834	IPv6 Datagrams Received with UDP Checksum Error (MAC_RXUDP_ERROR_PACKETS)	32	RO	0000_0000
838	IPv6 Datagrams Received with Good TCP Payload (MAC_RXTCP_GOOD_PACKETS)	32	RO	0000_0000
83C	IPv6 Datagrams Received with TCP Checksum Error (MAC_RXTCP_ERROR_PACKETS)	32	RO	0000_0000
840	IPv6 Datagrams Received with Good ICMP Payload (MAC_RXICMP_GOOD_PACKETS)	32	RO	0000_0000
844	IPv6 Datagrams Received with ICMP Checksum Error (MAC_RXICMP_ERROR_PACKETS)	32	RO	0000_0000
850	Good Bytes Received in IPv4 Datagrams (MAC_RXIPV4_GOOD_OCTETS)	32	RO	0000_0000
854	Bytes Received in IPv4 Datagrams with Header Errors (MAC_RXIPV4_HEADER_ERROR_OCTETS)	32	RO	0000_0000
858	Bytes Received in IPv4 Datagrams with No Payload (MAC_RXIPV4_NO_PAYLOAD_OCTETS)	32	RO	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
85C	Bytes Received in Fragmented IPv4 Datagrams (MAC_RXIPV4_FRAGMENTED_OCTETS)	32	RO	0000_0000
860	Bytes Received with UDP Checksum Disabled (MAC_RXIPV4_UDP_CHECKSUM_DISABLE_OCTETS)	32	RO	0000_0000
864	Bytes Received in Good IPv6 Datagrams (MAC_RXIPV6_GOOD_OCTETS)	32	RO	0000_0000
868	Bytes Received in IPv6 Datagrams with Data Errors (MAC_RXIPV6_HEADER_ERROR_OCTETS)	32	RO	0000_0000
86C	Bytes Received in IPv6 Datagrams with No Payload (MAC_RXIPV6_NO_PAYLOAD_OCTETS)	32	RO	0000_0000
870	Bytes Received in Good UDP Segment (MAC_RXUDP_GOOD_OCTETS)	32	RO	0000_0000
874	Bytes Received in UDP Segment with Checksum Errors (MAC_RXUDP_ERROR_OCTETS)	32	RO	0000_0000
878	Bytes Received in Good TCP Segment (MAC_RXTCP_GOOD_OCTETS)	32	RO	0000_0000
87C	Bytes Received in TCP Segment with Checksum Errors (MAC_RXTCP_ERROR_OCTETS)	32	RO	0000_0000
880	Bytes Received in Good ICMP Segment (MAC_RXICMP_GOOD_OCTETS)	32	RO	0000_0000
884	Bytes Received in ICMP Segment with Checksum Errors (MAC_RXICMP_ERROR_OCTETS)	32	RO	0000_0000
8A0	MMC FPE Transmit Interrupt (MAC_MMC_FPE_TX_INTERRUPT)	32	RO	0000_0000
8A4	MMC FPE Transmit Mask Interrupt (MAC_MMC_FPE_TX_INTERRUPT_MASK)	32	RW	0000_0000
8A8	MMC FPE Transmitted Fragment Counter (MAC_MMC_TX_FPE_FRAGMENT_CNTR)	32	RO	0000_0000
8AC	MMC FPE Transmitted Hold Request Counter (MAC_MMC_TX_HOLD_REQ_CNTR)	32	RO	0000_0000
8C0	MMC FPE Receive Interrupt (MAC_MMC_FPE_RX_INTERRUPT)	32	RO	0000_0000
8C4	MMC FPE Receive Interrupt Mask (MAC_MMC_FPE_RX_INTERRUPT_MASK)	32	RW	0000_0000
8C8	MMC Receive Packet Reassembly Error Counter (MAC_MMC_RX_PACKET_ASSEMBLY_ERR_CNTR)	32	RO	0000_0000
8CC	MMC Receive Packet SMD Error Counter (MAC_MMC_RX_PACKET_SMD_ERR_CNTR)	32	RO	0000_0000
8D0	MMC Receive Packet Successful Reassembly Counter (MAC_MMC_RX_PACKET_ASSEMBLY_OK_CNTR)	32	RO	0000_0000
8D4	MMC FPE Received Fragment Counter (MAC_MMC_RX_FPE_FRAGMENT_CNTR)	32	RO	0000_0000
900	Layer 3 and Layer 4 Control of Filter 0 (MAC_L3_L4_CONTROL0)	32	RW	0000_0000
904	Layer 4 Address 0 (MAC_LAYER4_ADDRESS0)	32	RW	0000_0000
910	Layer 3 Address 0 Register 0 (MAC_LAYER3_ADDR0_REG0)	32	RW	0000_0000
914	Layer 3 Address 1 Register 0 (MAC_LAYER3_ADDR1_REG0)	32	RW	0000_0000
918	Layer 3 Address 2 Register 0 (MAC_LAYER3_ADDR2_REG0)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
91C	Layer 3 Address 3 Register 0 (MAC_LAYER3_ADDR3_REG0)	32	RW	0000_0000
930	Layer 3 and Layer 4 Control of Filter 1 (MAC_L3_L4_CONTROL1)	32	RW	0000_0000
934	Layer 4 Address 0 (MAC_LAYER4_ADDRESS1)	32	RW	0000_0000
940	Layer 3 Address 0 Register 1 (MAC_LAYER3_ADDR0_REG1)	32	RW	0000_0000
944	Layer 3 Address 1 Register 1 (MAC_LAYER3_ADDR1_REG1)	32	RW	0000_0000
948	Layer 3 Address 2 Register 1 (MAC_LAYER3_ADDR2_REG1)	32	RW	0000_0000
94C	Layer 3 Address 3 Register 1 (MAC_LAYER3_ADDR3_REG1)	32	RW	0000_0000
960	Layer 3 and Layer 4 Control of Filter 2 (MAC_L3_L4_CONTROL2)	32	RW	0000_0000
964	Layer 4 Address 2 (MAC_LAYER4_ADDRESS2)	32	RW	0000_0000
970	Layer 3 Address 0 Register 2 (MAC_LAYER3_ADDR0_REG2)	32	RW	0000_0000
974	Layer 3 Address 0 Register 2 (MAC_LAYER3_ADDR1_REG2)	32	RW	0000_0000
978	Layer 3 Address 2 Register 2 (MAC_LAYER3_ADDR2_REG2)	32	RW	0000_0000
97C	Layer 3 Address 3 Register 2 (MAC_LAYER3_ADDR3_REG2)	32	RW	0000_0000
990	Layer 3 and Layer 4 Control of Filter 3 (MAC_L3_L4_CONTROL3)	32	RW	0000_0000
994	Layer 4 Address 3 (MAC_LAYER4_ADDRESS3)	32	RW	0000_0000
9A0	Layer 3 Address 0 Register 3 (MAC_LAYER3_ADDR0_REG3)	32	RW	0000_0000
9A4	Layer 3 Address 1 Register 3 (MAC_LAYER3_ADDR1_REG3)	32	RW	0000_0000
9A8	Layer 3 Address 2 Register 3 (MAC_LAYER3_ADDR2_REG3)	32	RW	0000_0000
9AC	Layer 3 Address 3 Register 3 (MAC_LAYER3_ADDR3_REG3)	32	RW	0000_0000
9C0	Layer 3 and Layer 4 Control of Filter 4 (MAC_L3_L4_CONTROL4)	32	RW	0000_0000
9C4	Layer 4 Address 4 (MAC_LAYER4_ADDRESS4)	32	RW	0000_0000
9D0	Layer 3 Address 0 Register 4 (MAC_LAYER3_ADDR0_REG4)	32	RW	0000_0000
9D4	Layer 3 Address 1 Register 4 (MAC_LAYER3_ADDR1_REG4)	32	RW	0000_0000
9D8	Layer 3 Address 2 Register 4 (MAC_LAYER3_ADDR2_REG4)	32	RW	0000_0000
9DC	Layer 3 Address 3 Register 4 (MAC_LAYER3_ADDR3_REG4)	32	RW	0000_0000
9F0	Layer 3 and Layer 4 Control of Filter 5 (MAC_L3_L4_CONTROL5)	32	RW	0000_0000
9F4	Layer 4 Address 5 (MAC_LAYER4_ADDRESS5)	32	RW	0000_0000
A00	Layer 3 Address 0 Register 5 (MAC_LAYER3_ADDR0_REG5)	32	RW	0000_0000
A04	Layer 3 Address 1 Register 5 (MAC_LAYER3_ADDR1_REG5)	32	RW	0000_0000
A08	Layer 3 Address 2 Register 5 (MAC_LAYER3_ADDR2_REG5)	32	RW	0000_0000
A0C	Layer 3 Address 3 Register 5 (MAC_LAYER3_ADDR3_REG5)	32	RW	0000_0000
A20	Layer 3 and Layer 4 Control of Filter 6 (MAC_L3_L4_CONTROL6)	32	RW	0000_0000
A24	Layer 4 Address 6 (MAC_LAYER4_ADDRESS6)	32	RW	0000_0000
A30	Layer 3 Address 0 Register 6 (MAC_LAYER3_ADDR0_REG6)	32	RW	0000_0000
A34	Layer 3 Address 1 Register 6 (MAC_LAYER3_ADDR1_REG6)	32	RW	0000_0000
A38	Layer 3 Address 2 Register 6 (MAC_LAYER3_ADDR2_REG6)	32	RW	0000_0000
A3C	Layer 3 Address 3 Register 6 (MAC_LAYER3_ADDR3_REG6)	32	RW	0000_0000
A50	Layer 3 and Layer 4 Control of Filter 0 (MAC_L3_L4_CONTROL7)	32	RW	0000_0000
A54	Layer 4 Address 7 (MAC_LAYER4_ADDRESS7)	32	RW	0000_0000
A60	Layer 3 Address 0 Register 7 (MAC_LAYER3_ADDR0_REG7)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
A64	Layer 3 Address 1 Register 7 (MAC_LAYER3_ADDR1_REG7)	32	RW	0000_0000
A68	Layer 3 Address 2 Register 7 (MAC_LAYER3_ADDR2_REG7)	32	RW	0000_0000
A6C	Layer 3 Address 3 Register 7 (MAC_LAYER3_ADDR3_REG7)	32	RW	0000_0000
B00	Timestamp Control (MAC_TIMESTAMP_CONTROL)	32	RW	0000_2000
B04	Subsecond Increment (MAC_SUB_SECOND_INCREMENT)	32	RW	0000_0000
B08	System Time Seconds (MAC_SYSTEM_TIME_SECONDS)	32	RO	0000_0000
B0C	System Time Nanoseconds (MAC_SYSTEM_TIME_NANOSECONDS)	32	RO	0000_0000
B10	System Time Seconds Update (MAC_SYSTEM_TIME_SECONDS_UPDATE)	32	RW	0000_0000
B14	System Time Nanoseconds Update (MAC_SYSTEM_TIME_NANOSECONDS_UPDATE)	32	RW	0000_0000
B18	Timestamp Addend (MAC_TIMESTAMP_ADDEND)	32	RW	0000_0000
B1C	System Time - Higher Word Seconds (MAC_SYSTEM_TIME_HIGHER_WORD_SECONDS)	32	RW	0000_0000
B20	Timestamp Status (MAC_TIMESTAMP_STATUS)	32	RO	0000_0000
B30	Transmit Timestamp Status Nanoseconds (MAC_TX_TIMESTAMP_STATUS_NANOSECONDS)	32	RO	0000_0000
B34	Transmit Timestamp Status Seconds (MAC_TX_TIMESTAMP_STATUS_SECONDS)	32	RO	0000_0000
B40	Auxiliary Timestamp Control (MAC_AUXILIARY_CONTROL)	32	RW	0000_0000
B48	Auxiliary Timestamp Nanoseconds (MAC_AUXILIARY_TIMESTAMP_NANOSECONDS)	32	RO	0000_0000
B4C	Auxiliary Timestamp Seconds (MAC_AUXILIARY_TIMESTAMP_SECONDS)	32	RO	0000_0000
B50	Timestamp Ingress Asymmetry Correction (MAC_TIMESTAMP_INGRESS_ASYM_CORR)	32	RW	0000_0000
B54	imestamp Egress Asymmetry Correction (MAC_TIMESTAMP_EGRESS_ASYM_CORR)	32	RW	0000_0000
B58	Timestamp Ingress Correction Nanosecond (MAC_TIMESTAMP_INGRESS_CORR_NANOSECOND)	32	RW	0000_0000
B5C	Timestamp Egress Correction Nanosecond (MAC_TIMESTAMP_EGRESS_CORR_NANOSECOND)	32	RW	0000_0000
B60	Timestamp Ingress Correction Subnanosecond (MAC_TIMESTAMP_INGRESS_CORR_SUBNANOSEC)	32	RW	0000_0000
B64	Timestamp Egress Correction Subnanosecond (MAC_TIMESTAMP_EGRESS_CORR_SUBNANOSEC)	32	RW	0000_0000
B68	Timestamp Ingress Latency (MAC_TIMESTAMP_INGRESS_LATENCY)	32	RO	0000_0000
B6C	Timestamp Egress Latency (MAC_TIMESTAMP_EGRESS_LATENCY)	32	RO	0000_0000
B70	PPS Control (MAC_PPS_CONTROL)	32	RW	0000_0000
B80	PPS0 Target Time Seconds (MAC_PPS0_TARGET_TIME_SECONDS)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
B84	PPS0 Target Time Nanoseconds (MAC_PPS0_TARGET_TIME_NANOSECONDS)	32	RW	0000_0000
B88	PPS0 Interval (MAC_PPS0_INTERVAL)	32	RW	0000_0000
B8C	PPS0 Width (MAC_PPS0_WIDTH)	32	RW	0000_0000
B90	PPS1 Target Time Seconds (MAC_PPS1_TARGET_TIME_SECONDS)	32	RW	0000_0000
B94	PPS1 Target Time Nanoseconds (MAC_PPS1_TARGET_TIME_NANOSECONDS)	32	RW	0000_0000
B98	PPS1 Interval (MAC_PPS1_INTERVAL)	32	RW	0000_0000
B9C	PPS1 Width (MAC_PPS1_WIDTH)	32	RW	0000_0000
BA0	PPS2 Target Time Seconds (MAC_PPS2_TARGET_TIME_SECONDS)	32	RW	0000_0000
BA4	PPS2 Target Time Nanoseconds (MAC_PPS2_TARGET_TIME_NANOSECONDS)	32	RW	0000_0000
BA8	PPS2 Interval (MAC_PPS2_INTERVAL)	32	RW	0000_0000
BAC	PPS2 Width (MAC_PPS2_WIDTH)	32	RW	0000_0000
BB0	PPS3 Target Time Seconds (MAC_PPS3_TARGET_TIME_SECONDS)	32	RW	0000_0000
BB4	PPS3 Target Time Nanoseconds (MAC_PPS3_TARGET_TIME_NANOSECONDS)	32	RW	0000_0000
BB8	PPS3 Interval (MAC_PPS3_INTERVAL)	32	RW	0000_0000
BBC	PPS3 Width (MAC_PPS3_WIDTH)	32	RW	0000_0000
BC0	PTP Offload Engine Control (MAC_PTO_CONTROL)	32	RW	0000_0000
BC4	Source Port Identity 0 (MAC_SOURCE_PORT_IDENTITY0)	32	RW	0000_0000
BC8	Source Port Identity 1 (MAC_SOURCE_PORT_IDENTITY1)	32	RW	0000_0000
BCC	Source Port Identity 2 (MAC_SOURCE_PORT_IDENTITY2)	32	RW	0000_0000
BD0	Log Message Interval (MAC_LOG_MESSAGE_INTERVAL)	32	RW	0000_0000
C00	MTL Operation Mode (MTL_OPERATION_MODE)	32	RW	0000_0000
C08	FIFO Debug Access Control and Status (MTL_DBG_CTL)	32	RW	0000_0000
C0C	FIFO Debug Status (MTL_DBG_STS)	32	RW	0000_0018
C10	FIFO Debug Data (MTL_FIFO_DEBUG_DATA)	32	RW	0000_0000
C20	MTL Interrupt Status (MTL_INTERRUPT_STATUS)	32	RO	0000_0000
C30	Receive Queue and DMA Channel Mapping 0 (MTL_RXQ_DMA_MAP0)	32	RW	0000_0000
C34	Receive Queue and DMA Channel Mapping 1 (MTL_RXQ_DMA_MAP1)	32	RW	0000_0000
C40	Time Based Scheduling Control (MTL_TBS_CTRL)	32	RW	0000_0000
C50	Enhancements to Scheduled Transmission Control (MTL_EST_CONTROL)	32	RW	0000_0000
C58	Enhancements to Scheduled Transmission Status (MTL_EST_STATUS)	32	RW	0000_0000
C60	EST Scheduling Error (MTL_EST_SCH_ERROR)	32	RW	0000_0000
C64	EST Frame Size Error (MTL_EST_FRM_SIZE_ERROR)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
C68	EST Frame Size Capture (MTL_EST_FRM_SIZE_CAPTURE)	32	RO	0000_0000
C70	EST Interrupt Enable (MTL_EST_INTR_ENABLE)	32	RW	0000_0000
C80	EST GCL Control (MTL_EST_GCL_CONTROL)	32	RW	0000_0000
C84	EST GCL Data (MTL_EST_GCL_DATA)	32	RW	0000_0000
C90	Frame Preemption Control and Status (MTL_FPE_CTRL_STS)	32	RW	0000_0000
C94	Frame Preemption Hold and Release Advance (MTL_FPE_ADVANCE)	32	RW	0000_0000
CA0	RXP Control Status (MTL_RXP_CONTROL_STATUS)	32	RW	80FF_00FF
CA4	RXP Interrupt Control Status (MTL_RXP_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000
CA8	RXP Drop Count (MTL_RXP_DROP_CNT)	32	RO	0000_0000
CAC	RXP Error Count (MTL_RXP_ERROR_CNT)	32	RO	0000_0000
CB0	RXP Indirect Access Control and Status (MTL_RXP_INDIRECT_ACC_CONTROL_STATUS)	32	RW	0000_0000
CB4	RXP Indirect Access Data (MTL_RXP_INDIRECT_ACC_DATA)	32	RW	0000_0000
D00	Queue 0 Transmit Operation Mode (MTL_TXQ0_OPERATION_MODE)	32	RW	0000_0000
D04	Queue 0 Underflow Counter (MTL_TXQ0_UNDERFLOW)	32	RO	0000_0000
D08	Queue 0 Transmit Debug (MTL_TXQ0_DEBUG)	32	RO	0000_0000
D14	Queue 0 ETS Status (MTL_TXQ0_ETS_STATUS)	32	RO	0000_0000
D18	Queue 0 Quantum or Weights (MTL_TXQ0_QUANTUM_WEIGHT)	32	RW	0000_0000
D2C	Queue 0 Interrupt Control Status (MTL_Q0_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000
D30	Queue 0 Receive Operation Mode (MTL_RXQ0_OPERATION_MODE)	32	RW	0000_0000
D34	Queue 0 Missed Packet and Overflow Counter (MTL_RXQ0_MISSED_PACKET_OVERFLOW_CNT)	32	RO	0000_0000
D38	Queue 0 Receive Debug (MTL_RXQ0_DEBUG)	32	RO	0000_0000
D3C	Queue 0 Receive Control (MTL_RXQ0_CONTROL)	32	RW	0000_0000
D40	Queue 1 Transmit Operation Mode (MTL_TXQ1_OPERATION_MODE)	32	RW	0000_0000
D44	Queue 1 Underflow Counter (MTL_TXQ1_UNDERFLOW)	32	RO	0000_0000
D48	Queue 1 Transmit Debug (MTL_TXQ1_DEBUG)	32	RO	0000_0000
D50	Queue 1 ETS Control (MTL_TXQ1_ETS_CONTROL)	32	RW	0000_0000
D54	Queue 1 ETS Status (MTL_TXQ1_ETS_STATUS)	32	RO	0000_0000
D58	Queue 1 idleSlopeCredit, Quantum or Weights (MTL_TXQ1_QUANTUM_WEIGHT)	32	RW	0000_0000
D5C	Queue 1 sendSlopeCredit (MTL_TXQ1_SENDSLOPECREDIT)	32	RW	0000_0000
D60	Queue 1 hiCredit (MTL_TXQ1_HICREDIT)	32	RW	0000_0000
D64	Queue 1 loCredit (MTL_TXQ1_LOCREDIT)	32	RW	0000_0000
D6C	Queue 1 Interrupt Control Status (MTL_Q1_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
D70	Queue 1 Receive Operation Mode (MTL_RXQ1_OPERATION_MODE)	32	RW	0000_0000
D74	Queue 1 Missed Packet and Overflow Counter (MTL_RXQ1_MISSED_PACKET_OVERFLOW_CNT)	32	RO	0000_0000
D78	Queue 1 Receive Debug (MTL_RXQ1_DEBUG)	32	RO	0000_0000
D7C	Queue 1 Receive Control (MTL_RXQ1_CONTROL)	32	RW	0000_0000
D80	Queue 2 Transmit Operation Mode (MTL_TXQ2_OPERATION_MODE)	32	RW	0000_0000
D84	Queue 2 Underflow Counter (MTL_TXQ2_UNDERFLOW)	32	RO	0000_0000
D88	Queue 2 Transmit Debug (MTL_TXQ2_DEBUG)	32	RO	0000_0000
D90	Queue 2 ETS Control (MTL_TXQ2_ETS_CONTROL)	32	RW	0000_0000
D94	Queue 2 ETS Status (MTL_TXQ2_ETS_STATUS)	32	RO	0000_0000
D98	Queue 2 idleSlopeCredit, Quantum or Weights (MTL_TXQ2_QUANTUM_WEIGHT)	32	RW	0000_0000
D9C	Queue 2 sendSlopeCredit (MTL_TXQ2_SENDSLOPECREDIT)	32	RW	0000_0000
DA0	Queue 2 hiCredit (MTL_TXQ2_HICREDIT)	32	RW	0000_0000
DA4	Queue 2 loCredit (MTL_TXQ2_LOCREDIT)	32	RW	0000_0000
DAC	Queue 2 Interrupt Control Status (MTL_Q2_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000
DB0	Queue 2 Receive Operation Mode (MTL_RXQ2_OPERATION_MODE)	32	RW	0000_0000
DB4	Queue 2 Missed Packet and Overflow Counter (MTL_RXQ2_MISSED_PACKET_OVERFLOW_CNT)	32	RO	0000_0000
DB8	Queue 2 Receive Debug (MTL_RXQ2_DEBUG)	32	RO	0000_0000
DBC	Queue 2 Receive Control (MTL_RXQ2_CONTROL)	32	RW	0000_0000
DC0	Queue 3 Transmit Operation Mode (MTL_TXQ3_OPERATION_MODE)	32	RW	0000_0000
DC4	Queue 3 Underflow Counter (MTL_TXQ3_UNDERFLOW)	32	RO	0000_0000
DC8	Queue 3 Transmit Debug (MTL_TXQ3_DEBUG)	32	RO	0000_0000
DD0	Queue 3 ETS Control (MTL_TXQ3_ETS_CONTROL)	32	RW	0000_0000
DD4	Queue 3 ETS Status (MTL_TXQ3_ETS_STATUS)	32	RO	0000_0000
DD8	Queue 3 idleSlopeCredit, Quantum or Weights (MTL_TXQ3_QUANTUM_WEIGHT)	32	RW	0000_0000
DDC	Queue 3 sendSlopeCredit (MTL_TXQ3_SENDSLOPECREDIT)	32	RW	0000_0000
DE0	Queue 3 hiCredit (MTL_TXQ3_HICREDIT)	32	RW	0000_0000
DE4	Queue 3 loCredit (MTL_TXQ3_LOCREDIT)	32	RW	0000_0000
DEC	Queue 3 Interrupt Control Status (MTL_Q3_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000
DF0	Queue 3 Receive Operation Mode (MTL_RXQ3_OPERATION_MODE)	32	RW	0000_0000
DF4	Queue 3 Missed Packet and Overflow Counter (MTL_RXQ3_MISSED_PACKET_OVERFLOW_CNT)	32	RO	0000_0000
DF8	Queue 3 Receive Debug (MTL_RXQ3_DEBUG)	32	RO	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
DFC	Queue 3 Receive Control (MTL_RXQ3_CONTROL)	32	RW	0000_0000
E00	Queue 4 Transmit Operation Mode (MTL_TXQ4_OPERATION_MODE)	32	RW	0000_0000
E04	Queue 4 Underflow Counter (MTL_TXQ4_UNDERFLOW)	32	RO	0000_0000
E08	Queue 4 Transmit Debug (MTL_TXQ4_DEBUG)	32	RO	0000_0000
E10	Queue 4 ETS Control (MTL_TXQ4_ETS_CONTROL)	32	RW	0000_0000
E14	Queue 4 ETS Status (MTL_TXQ4_ETS_STATUS)	32	RO	0000_0000
E18	Queue 4 idleSlopeCredit, Quantum or Weights (MTL_TXQ4_QUANTUM_WEIGHT)	32	RW	0000_0000
E1C	Queue 4 sendSlopeCredit (MTL_TXQ4_SENDSLOPECREDIT)	32	RW	0000_0000
E20	Queue 4 hiCredit (MTL_TXQ4_HICREDIT)	32	RW	0000_0000
E24	Queue 4 loCredit (MTL_TXQ4_LOCREDIT)	32	RW	0000_0000
E2C	Queue 4 Interrupt Control Status (MTL_Q4_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000
E30	Queue 4 Receive Operation Mode (MTL_RXQ4_OPERATION_MODE)	32	RW	0000_0000
E34	Queue 4 Missed Packet and Overflow Counter (MTL_RXQ4_MISSED_PACKET_OVERFLOW_CNT)	32	RO	0000_0000
E38	Queue 4 Receive Debug (MTL_RXQ4_DEBUG)	32	RO	0000_0000
E3C	Queue 4 Receive Control (MTL_RXQ4_CONTROL)	32	RW	0000_0000
1000	DMA Bus Mode (DMA_MODE)	32	RW	0000_0000
1004	DMA System Bus Mode (DMA_SYSBUS_MODE)	32	RW	0101_0000
1008	DMA Interrupt Status (DMA_INTERRUPT_STATUS)	32	RO	0000_0000
100C	DMA Debug Status 0 (DMA_DEBUG_STATUS0)	32	RO	0000_0000
1010	DMA Debug Status 1 (DMA_DEBUG_STATUS1)	32	RO	0000_0000
1040	AXI LPI Entry Interval Control (DMA_AXI_LPI_ENTRY_INTERVAL)	32	RW	0000_0000
1050	TBS Control (DMA_TBS_CTRL)	32	RW	0000_0000
1100	DMA Channel 0 Control (DMA_CH0_CONTROL)	32	RW	0000_0000
1104	DMA Channel 0 Transmit Control (DMA_CH0_TX_CONTROL)	32	RW	0000_0000
1108	DMA Channel 0 Receive Control (DMA_CH0_RX_CONTROL)	32	RW	0000_0000
1114	Channel 0 Tx Descriptor List Address register (DMA_CH0_TXDESC_LIST_ADDRESS)	32	RW	0000_0000
111C	Channel 0 Rx Descriptor List Address register (DMA_CH0_RXDESC_LIST_ADDRESS)	32	RW	0000_0000
1120	Channel 0 Tx Descriptor Tail Pointer (DMA_CH0_TXDESC_TAIL_POINTER)	32	RW	0000_0000
1128	Channel 0 Rx Descriptor Tail Pointer (DMA_CH0_RXDESC_TAIL_POINTER)	32	RW	0000_0000
112C	Channel 0 Tx Descriptor Ring Length (DMA_CH0_TXDESC_RING_LENGTH)	32	RW	0000_0000
1130	Channel 0 Rx Descriptor Ring Length (DMA_CH0_RXDESC_RING_LENGTH)	32	RW	0000_0000
1134	Channel 0 Interrupt Enable (DMA_CH0_INTERRUPT_ENABLE)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1138	Channel 0 Receive Interrupt Watchdog Timer (DMA_CH0_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000
113C	Channel 0 Slot Function Control and Status (DMA_CH0_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0
1144	Channel 0 Current Application Transmit Descriptor (DMA_CH0_CURRENT_APP_TXDESC)	32	RO	0000_0000
114C	Channel 0 Current Application Receive Descriptor (DMA_CH0_CURRENT_APP_RXDESC)	32	RO	0000_0000
1154	Channel 0 Current Application Transmit Buffer Address (DMA_CH0_CURRENT_APP_TXBUFFER)	32	RO	0000_0000
115C	Channel 0 Current Application Receive Buffer Address (DMA_CH0_CURRENT_APP_RXBUFFER)	32	RO	0000_0000
1160	DMA Channel 0 Status (DMA_CH0_STATUS)	32	RW	0000_0000
1164	Channel 0 Missed Frame Counter (DMA_CH0_MISS_FRAME_CNT)	32	RO	0000_0000
1168	Channel 0 RXP Frames Accepted Counter (DMA_CH0_RXP_ACCEPT_CNT)	32	RO	0000_0000
116C	Channel 0 Receive ERI Counter (DMA_CH0_RX_ERI_CNT)	32	RO	0000_0000
1180	DMA Channel 1 Control (DMA_CH1_CONTROL)	32	RW	0000_0000
1184	DMA Channel 1 Transmit Control (DMA_CH1_TX_CONTROL)	32	RW	0000_0000
1188	DMA Channel 1 Receive Control (DMA_CH1_RX_CONTROL)	32	RW	0000_0000
1194	Channel 1 Tx Descriptor List Address (DMA_CH1_TXDESC_LIST_ADDRESS)	32	RW	0000_0000
119C	Channel 1 Rx Descriptor List Address (DMA_CH1_RXDESC_LIST_ADDRESS)	32	RW	0000_0000
11A0	Channel 1 Tx Descriptor Tail Pointer (DMA_CH1_TXDESC_TAIL_POINTER)	32	RW	0000_0000
11A8	Channel 1 Rx Descriptor Tail Pointer (DMA_CH1_RXDESC_TAIL_POINTER)	32	RW	0000_0000
11AC	Channel 1 Tx Descriptor Ring Length (DMA_CH1_TXDESC_RING_LENGTH)	32	RW	0000_0000
11B0	Channel 1 Rx Descriptor Ring Length (DMA_CH1_RXDESC_RING_LENGTH)	32	RW	0000_0000
11B4	Channel 1 Interrupt Enable (DMA_CH1_INTERRUPT_ENABLE)	32	RW	0000_0000
11B8	Channel 1 Receive Interrupt Watchdog Timer (DMA_CH1_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000
11BC	Channel 1 Slot Function Control and Status (DMA_CH1_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0
11C4	Channel 1 Current Application Transmit Descriptor (DMA_CH1_CURRENT_APP_TXDESC)	32	RO	0000_0000
11CC	Channel 1 Current Application Receive Descriptor (DMA_CH1_CURRENT_APP_RXDESC)	32	RO	0000_0000
11D4	Channel 1 Current Application Transmit Buffer Address (DMA_CH1_CURRENT_APP_TXBUFFER)	32	RO	0000_0000
11DC	Channel 1 Current Application Receive Buffer Address (DMA_CH1_CURRENT_APP_RXBUFFER)	32	RO	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
11E0	DMA Channel 1 Status (DMA_CH1_STATUS)	32	RW	0000_0000
11E4	Channel 1 Missed Frame Counter (DMA_CH1_MISS_FRAME_CNT)	32	RO	0000_0000
11E8	Channel 1 RXP Frames Accepted Counter (DMA_CH1_RXP_ACCEPT_CNT)	32	RO	0000_0000
11EC	Channel 1 Receive ERI Counter (DMA_CH1_RX_ERI_CNT)	32	RO	0000_0000
1200	DMA Channel 2 Control (DMA_CH2_CONTROL)	32	RW	0000_0000
1204	DMA Channel 2 Transmit Control (DMA_CH2_TX_CONTROL)	32	RW	0000_0000
1208	DMA Channel 2 Receive Control (DMA_CH2_RX_CONTROL)	32	RW	0000_0000
1214	Channel 2 Tx Descriptor List Address (DMA_CH2_TXDESC_LIST_ADDRESS)	32	RW	0000_0000
121C	Channel 2 Rx Descriptor List Address (DMA_CH2_RXDESC_LIST_ADDRESS)	32	RW	0000_0000
1220	Channel 2 Tx Descriptor Tail Pointer (DMA_CH2_TXDESC_TAIL_POINTER)	32	RW	0000_0000
1228	Channel 2 Rx Descriptor Tail Pointer (DMA_CH2_RXDESC_TAIL_POINTER)	32	RW	0000_0000
122C	Channel 2 Tx Descriptor Ring Length (DMA_CH2_TXDESC_RING_LENGTH)	32	RW	0000_0000
1230	Channel 2 Rx Descriptor Ring Length (DMA_CH2_RXDESC_RING_LENGTH)	32	RW	0000_0000
1234	Channel 2 Interrupt Enable (DMA_CH2_INTERRUPT_ENABLE)	32	RW	0000_0000
1238	Channel 2 Receive Interrupt Watchdog Timer (DMA_CH2_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000
123C	Channel 2 Slot Function Control and Status (DMA_CH2_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0
1244	Channel 2 Current Application Transmit Descriptor (DMA_CH2_CURRENT_APP_TXDESC)	32	RO	0000_0000
124C	Channel 2 Current Application Receive Descriptor (DMA_CH2_CURRENT_APP_RXDESC)	32	RO	0000_0000
1254	Channel 2 Current Application Transmit Buffer Address (DMA_CH2_CURRENT_APP_TXBUFFER)	32	RO	0000_0000
125C	Channel 2 Current Application Receive Buffer Address (DMA_CH2_CURRENT_APP_RXBUFFER)	32	RO	0000_0000
1260	DMA Channel 2 Status (DMA_CH2_STATUS)	32	RW	0000_0000
1264	Channel 2 Missed Frame Counter (DMA_CH2_MISS_FRAME_CNT)	32	RO	0000_0000
1268	Channel 2 RXP Frames Accepted Counter (DMA_CH2_RXP_ACCEPT_CNT)	32	RO	0000_0000
126C	Channel 2 Receive ERI Counter (DMA_CH2_RX_ERI_CNT)	32	RO	0000_0000
1280	DMA Channel 3 Control (DMA_CH3_CONTROL)	32	RW	0000_0000
1284	DMA Channel 3 Transmit Control (DMA_CH3_TX_CONTROL)	32	RW	0000_0000
1288	DMA Channel 3 Receive Control (DMA_CH3_RX_CONTROL)	32	RW	0000_0000
1294	Channel 3 Tx Descriptor List Address (DMA_CH3_TXDESC_LIST_ADDRESS)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
129C	Channel 3 Rx Descriptor List Address (DMA_CH3_RXDESC_LIST_ADDRESS)	32	RW	0000_0000
12A0	Channel 3 Tx Descriptor Tail Pointer (DMA_CH3_TXDESC_TAIL_POINTER)	32	RW	0000_0000
12A8	Channel 3 Rx Descriptor Tail Pointer (DMA_CH3_RXDESC_TAIL_POINTER)	32	RW	0000_0000
12AC	Channel 3 Tx Descriptor Ring Length (DMA_CH3_TXDESC_RING_LENGTH)	32	RW	0000_0000
12B0	Channel 3 Rx Descriptor Ring Length (DMA_CH3_RXDESC_RING_LENGTH)	32	RW	0000_0000
12B4	Channel 3 Interrupt Enable (DMA_CH3_INTERRUPT_ENABLE)	32	RW	0000_0000
12B8	Channel 3 Receive Interrupt Watchdog Time (DMA_CH3_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000
12BC	Channel 3 Slot Function Control and Status (DMA_CH3_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0
12C4	Channel 3 Current Application Transmit Descriptor (DMA_CH3_CURRENT_APP_TXDESC)	32	RO	0000_0000
12CC	Channel 3 Current Application Receive Descriptor (DMA_CH3_CURRENT_APP_RXDESC)	32	RO	0000_0000
12D4	Channel 3 Current Application Transmit Buffer Address (DMA_CH3_CURRENT_APP_TXBUFFER)	32	RO	0000_0000
12DC	Channel 3 Current Application Receive Buffer Address (DMA_CH3_CURRENT_APP_RXBUFFER)	32	RO	0000_0000
12E0	DMA Channel 3 Status (DMA_CH3_STATUS)	32	RW	0000_0000
12E4	Channel 3 Missed Frame Counter (DMA_CH3_MISS_FRAME_CNT)	32	RO	0000_0000
12E8	Channel 3 RXP Frames Accepted Counter (DMA_CH3_RXP_ACCEPT_CNT)	32	RO	0000_0000
12EC	Channel 3 Receive ERI Counter (DMA_CH3_RX_ERI_CNT)	32	RO	0000_0000
1300	DMA Channel 4 Control (DMA_CH4_CONTROL)	32	RW	0000_0000
1304	DMA Channel 4 Transmit Control (DMA_CH4_TX_CONTROL)	32	RW	0000_0000
1308	DMA Channel 4 Receive Control (DMA_CH4_RX_CONTROL)	32	RW	0000_0000
1314	Channel 4 Tx Descriptor List Address (DMA_CH4_TXDESC_LIST_ADDRESS)	32	RW	0000_0000
131C	Channel 4 Rx Descriptor List Address (DMA_CH4_RXDESC_LIST_ADDRESS)	32	RW	0000_0000
1320	Channel 4 Tx Descriptor Tail Pointer (DMA_CH4_TXDESC_TAIL_POINTER)	32	RW	0000_0000
1328	Channel 4 Rx Descriptor Tail Pointer (DMA_CH4_RXDESC_TAIL_POINTER)	32	RW	0000_0000
132C	Channel 4 Tx Descriptor Ring Length (DMA_CH4_TXDESC_RING_LENGTH)	32	RW	0000_0000
1330	Channel 4 Rx Descriptor Ring Length (DMA_CH4_RXDESC_RING_LENGTH)	32	RW	0000_0000
1334	Channel 4 Interrupt Enable (DMA_CH4_INTERRUPT_ENABLE)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1338	Channel 4 Receive Interrupt Watchdog Timer (DMA_CH4_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000
133C	Channel 4 Slot Function Control and Status (DMA_CH4_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0
1344	Channel 4 Current Application Transmit Descriptor (DMA_CH4_CURRENT_APP_TXDESC)	32	RO	0000_0000
134C	Channel 4 Current Application Receive Descriptor (DMA_CH4_CURRENT_APP_RXDESC)	32	RO	0000_0000
1354	Channel 4 Current Application Transmit Buffer Address (DMA_CH4_CURRENT_APP_TXBUFFER)	32	RO	0000_0000
135C	Channel 4 Current Application Receive Buffer Address (DMA_CH4_CURRENT_APP_RXBUFFER)	32	RO	0000_0000
1360	DMA Channel 4 Status (DMA_CH4_STATUS)	32	RW	0000_0000
1364	Channel 4 Missed Frame Counter (DMA_CH4_MISS_FRAME_CNT)	32	RO	0000_0000
1368	Channel 4 RXP Frames Accepted Counter (DMA_CH4_RXP_ACCEPT_CNT)	32	RO	0000_0000
136C	Channel 4 Receive ERI Counter (DMA_CH4_RX_ERI_CNT)	32	RO	0000_0000

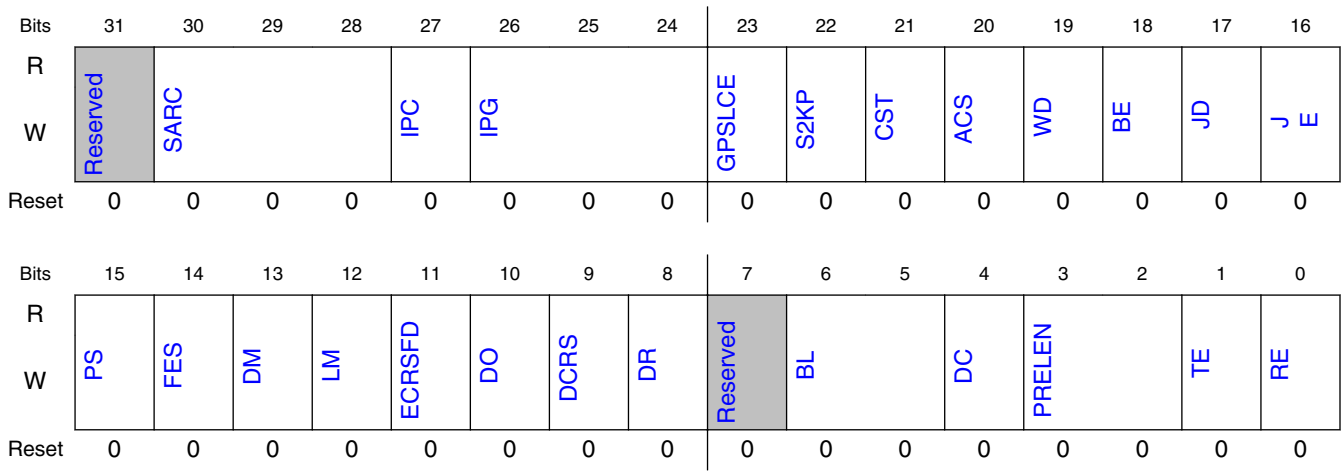
11.7.6.1.2 MAC Configuration Register (MAC_CONFIGURATION)

The MAC Configuration Register establishes the operating mode of the MAC.

11.7.6.1.2.1 Offset

Register	Offset
MAC_CONFIGURATION	0h

11.7.6.1.2.2 Diagram



11.7.6.1.2.3 Fields

Field	Description
31 —	Reserved
30-28 SARC	Source Address Insertion or Replacement Control This field controls the source address insertion or replacement for all transmitted packets. Bit 30 specifies which MAC Address register (0 or 1) is used for source address insertion or replacement based on the values of Bits[29:28]: 2'b0x: - The mti_sa_ctrl_i and ati_sa_ctrl_i input signals control the SA field generation. 2'b10: - If Bit 30 is set to 0, the MAC inserts the content of the MAC Address 0 registers in the SA field of all transmitted packets. - If Bit 30 is set to 1 and the Enable MAC Address Register 1 option is selected while configuring the core, the MAC inserts the content of the MAC Address 1 registers in the SA field of all transmitted packets. 2'b11: - If Bit 30 is set to 0, the MAC replaces the content of the MAC Address 0 registers in the SA field of all transmitted packets. - If Bit 30 is set to 1 and the MAC Address Register 1 is enabled, the MAC replaces the content of the MAC Address 1 registers in the SA field of all transmitted packets. Note: - Changes to this field take effect only on the start of a packet. If you write to this register field when a packet is being transmitted, only the subsequent packet can use the updated value, that is, the current packet does not use the updated value. 000 - mti_sa_ctrl_i and ati_sa_ctrl_i input signals control the SA field generation 010 - Contents of MAC Addr-0 inserted in SA field

Table continues on the next page...

Field	Description
	011 - Contents of MAC Addr-0 replaces SA field 110 - Contents of MAC Addr-1 inserted in SA field 111 - Contents of MAC Addr-1 replaces SA field
27 IPC	Checksum Offload When set, this bit enables the IPv4 header checksum checking and IPv4 or IPv6 TCP, UDP, or ICMP payload checksum checking. When this bit is reset, the COE function in the receiver is disabled. The Layer 3 and Layer 4 Packet Filter and Enable Split Header features automatically selects the IPC Full Checksum Offload Engine on the Receive side. When any of these features are enabled, you must set the IPC bit. 0 - IP header/payload checksum checking is disabled 1 - IP header/payload checksum checking is enabled
26-24 IPG	Inter-Packet Gap These bits control the minimum IPG between packets during transmission. Inter-Packet Gap These bits control the minimum IPG between packets during transmission. This range of minimum IPG is valid in full-duplex mode. In the half-duplex mode, the minimum IPG can be configured only for 64-bit times (IPG = 100). Lower values are not considered. When a JAM pattern is being transmitted because of backpressure activation, the MAC does not consider the minimum IPG. The above function (IPG less than 96 bit times) is valid only when EIPGEN bit in MAC_EXT_CONFIGURATION register is reset. When EIPGEN is set, then the minimum IPG (greater than 96 bit times) is controlled as per the description given in EIPG field in MAC_EXT_CONFIGURATION register. 000 - 96 bit times IPG 001 - 88 bit times IPG 010 - 80 bit times IPG 011 - 72 bit times IPG 100 - 64 bit times IPG 101 - 56 bit times IPG 110 - 48 bit times IPG 111 - 40 bit times IPG
23 GPSLCE	Giant Packet Size Limit Control Enable When this bit is set, the MAC considers the value in GPSL field in MAC_EXT_CONFIGURATION register to declare a received packet as Giant packet. This field must be programmed to more than 1,518 bytes. Otherwise, the MAC considers 1,518 bytes as giant packet limit. When this bit is reset, the MAC considers a received packet as Giant packet when its size is greater than 1,518 bytes (1522 bytes for tagged packet). The watchdog timeout limit, Jumbo Packet Enable and 2K Packet Enable have higher precedence over this bit, that is the MAC considers a received packet as Giant packet when its size is greater than 9,018 bytes (9,022 bytes for tagged packet) with Jumbo Packet Enabled and greater than 2,000 bytes with 2K Packet Enabled. The watchdog timeout, if enabled, terminates the received packet when watchdog limit is reached. Therefore, the programmed giant packet limit should be less than the watchdog limit to get the giant packet status. 0 - Giant Packet Size Limit Control is disabled

Table continues on the next page...

Field	Description
	1 - Giant Packet Size Limit Control is enabled
22 S2KP	IEEE 802. IEEE 802.3as Support for 2K Packets When this bit is set, the MAC considers all packets with up to 2,000 bytes length as normal packets. When the JE bit is not set, the MAC considers all received packets of size more than 2K bytes as Giant packets. When this bit is reset and the JE bit is not set, the MAC considers all received packets of size more than 1,518 bytes (1,522 bytes for tagged) as giant packets. For more information about how the setting of this bit and the JE bit impact the Giant packet status, see the Table, Gaint Packet Status based on S2KP and JE Bits. Note: When the JE bit is set, setting this bit has no effect on the giant packet status. 0 - Support upto 2K packet is disabled 1 - Support upto 2K packet is Enabled
21 CST	CRC stripping for Type packets When this bit is set, the last four bytes (FCS) of all packets of Ether type (type field greater than 1,536) are stripped and dropped before forwarding the packet to the application. CRC stripping for Type packets When this bit is set, the last four bytes (FCS) of all packets of Ether type (type field greater than 1,536) are stripped and dropped before forwarding the packet to the application. Note: For information about how the settings of the ACS bit and this bit impact the packet length, see the Table, Packet Length based on the CST and ACS Bits. 0 - CRC stripping for Type packets is disabled 1 - CRC stripping for Type packets is enabled
20 ACS	Automatic Pad or CRC Stripping When this bit is set, the MAC strips the Pad or FCS field on the incoming packets only if the value of the length field is less than 1,536 bytes. Automatic Pad or CRC Stripping When this bit is set, the MAC strips the Pad or FCS field on the incoming packets only if the value of the length field is less than 1,536 bytes. All received packets with length field greater than or equal to 1,536 bytes are passed to the application without stripping the Pad or FCS field. When this bit is reset, the MAC passes all incoming packets to the application, without any modification. Note: For information about how the settings of CST bit and this bit impact the packet length, see the Table, Packet Length based on the CST and ACS Bit . 0 - Automatic Pad or CRC Stripping is disabled 1 - Automatic Pad or CRC Stripping is enabled
19 WD	Watchdog Disable When this bit is set, the MAC disables the watchdog timer on the receiver. The MAC can receive packets of up to 16,383 bytes. When this bit is reset, the MAC does not allow more than 2,048 bytes (10,240 if JE is set high) of the packet being received. The MAC cuts off any bytes received after 2,048 bytes. 0 - Watchdog is enabled 1 - Watchdog is disabled
18 BE	Packet Burst Enable When this bit is set, the MAC allows packet bursting during transmission in the GMII half-duplex mode. 0 - Packet Burst is disabled 1 - Packet Burst is enabled

Table continues on the next page...

Field	Description
17 JD	Jabber Disable When this bit is set, the MAC disables the jabber timer on the transmitter. The MAC can transfer packets of up to 16,383 bytes. When this bit is reset, if the application sends more than 2,048 bytes of data (10,240 if JE is set high) during transmission, the MAC does not send rest of the bytes in that packet. 0 - Jabber is enabled 1 - Jabber is disabled
16 JE	Jumbo Packet Enable When this bit is set, the MAC allows jumbo packets of 9,018 bytes (9,022 bytes for VLAN tagged packets) without reporting a giant packet error in the Rx packet status. 0 - Jumbo packet is disabled 1 - Jumbo packet is enabled
15 PS	Port Select This bit selects the Ethernet line speed. This bit, along with Bit 14, selects the exact line speed. In the 10/100 Mbps-only (always 1) or 1000 Mbps-only (always 0) configurations, this bit is read-only (RO) with appropriate value. In default 10/100/1000 Mbps configurations, this bit is read-write (R/W). The mac_speed_o[1] signal reflects the value of this bit. 0 - For 1000 or 2500 Mbps operations 1 - For 10 or 100 Mbps operations
14 FES	Speed This bit selects the speed mode. The mac_speed_o[0] signal reflects the value of this bit. 0 - 10 Mbps when PS bit is 1 and 1 Gbps when PS bit is 0 1 - 100 Mbps when PS bit is 1 and 2.5 Gbps when PS bit is 0
13 DM	Duplex Mode When this bit is set, the MAC operates in the full-duplex mode in which it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in the full-duplex-only configurations. 0 - Half-duplex mode 1 - Full-duplex mode
12 LM	Loopback Mode When this bit is set, the MAC operates in the loopback mode at GMII. The (G)MII Rx clock input (clk_rx_i) is required for the loopback to work properly. This is because the Tx clock is not internally looped back. 0 - Loopback is disabled 1 - Loopback is enabled
11 ECRSFD	Enable Carrier Sense Before Transmission in Full-Duplex Mode When this bit is set, the MAC transmitter checks the CRS signal before packet transmission in the full-duplex mode. The MAC starts the transmission only when the CRS signal is low. When this bit is reset, the MAC transmitter ignores the status of the CRS signal. 0 - ECRSFD is disabled 1 - ECRSFD is enabled
10 DO	Disable Receive Own

Table continues on the next page...

Field	Description
	When this bit is set, the MAC disables the reception of packets when the gmii_txen_o is asserted in the half-duplex mode. When this bit is reset, the MAC receives all packets given by the PHY. This bit is not applicable in the full-duplex mode. 0 - Enable Receive Own 1 - Disable Receive Own
9 DCRS	Disable Carrier Sense During Transmission When this bit is set, the MAC transmitter ignores the (G)MII CRS signal during packet transmission in the half-duplex mode. As a result, no errors are generated because of Loss of Carrier or No Carrier during transmission. When this bit is reset, the MAC transmitter generates errors because of Carrier Sense. The MAC can even abort the transmission. 0 - Enable Carrier Sense During Transmission 1 - Disable Carrier Sense During Transmission
8 DR	Disable Retry When this bit is set, the MAC attempts only one transmission. When a collision occurs on the GMII interface, the MAC ignores the current packet transmission and reports a Packet Abort with excessive collision error in the Tx packet status. When this bit is reset, the MAC retries based on the settings of the BL field. This bit is applicable only in the half-duplex mode. 0 - Enable Retry 1 - Disable Retry
7 —	Reserved
6-5 BL	Back-Off Limit The back-off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000/2500 Mbps; 512 bit times for 10/100 Mbps) for which the MAC waits before rescheduling a transmission attempt during retries after a collision. n = retransmission attempt. The random integer r takes the value in the range 0 <= r < 2^k This bit is applicable only in the half-duplex mode. 00 - k = min(n,10) 01 - k = min(n,8) 10 - k = min(n,4) 11 - k = min(n,1)
4 DC	Deferral Check When this bit is set, the deferral check function is enabled in the MAC. The MAC issues a Packet Abort status, along with the excessive deferral error bit set in the Tx packet status, when the Tx state machine is deferred for more than 24,288 bit times in 10 or 100 Mbps mode. If the MAC is configured for 1000/2500 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but it is prevented because of an active carrier sense signal (CRS) on GMII.

Table continues on the next page...

Field	Description
	The defer time is not cumulative. For example, if the transmitter defers for 10,000 bit times because the CRS signal is active and the CRS signal becomes inactive, the transmitter transmits and collision happens. Because of collision, the transmitter needs to back off and then defer again after back off completion. In such a scenario, the deferral timer is reset to 0, and it is restarted. When this bit is reset, the deferral check function is disabled and the MAC defers until the CRS signal goes inactive. This bit is applicable only in the half-duplex mode. 0 - Deferral check function is disabled 1 - Deferral check function is enabled
3-2 PRELEN	Preamble Length for Transmit packets These bits control the number of preamble bytes that are added to the beginning of every Tx packet. The preamble reduction occurs only when the MAC is operating in the full-duplex mode. 00 - 7 bytes of preamble 01 - 5 bytes of preamble 10 - 3 bytes of preamble 11 - Reserved
1 TE	Transmitter Enable When this bit is set, the Tx state machine of the MAC is enabled for transmission on the GMII interface. When this bit is reset, the MAC Tx state machine is disabled after it completes the transmission of the current packet. The Tx state machine does not transmit any more packets. 0 - Transmitter is disabled 1 - Transmitter is enabled
0 RE	Receiver Enable When this bit is set, the Rx state machine of the MAC is enabled for receiving packets from the GMII interface. When this bit is reset, the MAC Rx state machine is disabled after it completes the reception of the current packet. The Rx state machine does not receive any more packets from the GMII interface. 0 - Receiver is disabled 1 - Receiver is enabled

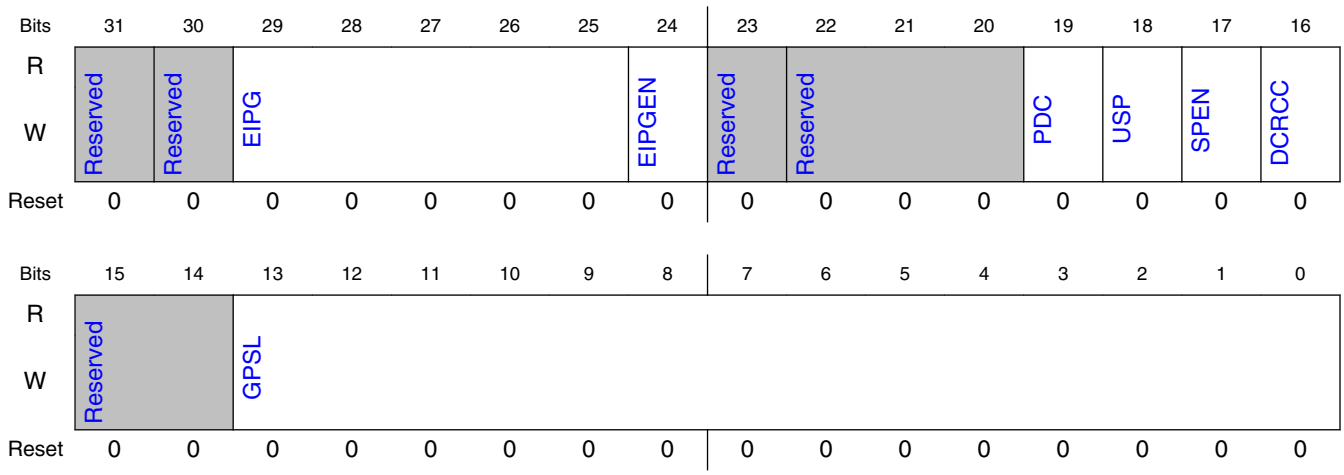
11.7.6.1.3 MAC Extended Configuration Register (MAC_EXT_CONFIGURATION)

The MAC Extended Configuration Register establishes the operating mode of the MAC.

11.7.6.1.3.1 Offset

Register	Offset
MAC_EXT_CONFIGURATION	4h

11.7.6.1.3.2 Diagram



11.7.6.1.3.3 Fields

Field	Description
31 —	Reserved
30 —	Reserved
29-25 EIPG	Extended Inter-Packet Gap The value in this field is applicable when the EIPGEN bit is set. This field (as Most Significant bits), along with IPG field in CONFIGURATION register, gives the minimum IPG greater than 96 bit times in steps of 8 bit times: {EIPG, IPG} 8'h00 - 104 bit times 8'h01 - 112 bit times 8'h02 - 120 bit times ---------- 8'hFF - 2144 bit times
24 EIPGEN	Extended Inter-Packet Gap Enable When this bit is set, the MAC interprets EIPG field and IPG field in CONFIGURATION register together as minimum IPG greater than 96 bit times in steps of 8 bit times. When this bit is reset, the MAC ignores EIPG field and interprets IPG field in CONFIGURATION register as minimum IPG less than or equal to 96 bit times in steps of 8 bit times. Note: The extended Inter-Packet Gap feature must be enabled when operating in Full-Duplex mode only. There may be undesirable effects on back-pressure function and frame transmission if it is enabled in Half-Duplex mode. 0 - Extended Inter-Packet Gap is disabled 1 - Extended Inter-Packet Gap is enabled
23	Reserved

Table continues on the next page...

Field	Description
—	
22-20 —	Reserved
19 PDC	Packet Duplication Control When this bit is set, the received packet with Multicast/Broadcast Destination address is routed to multiple Receive DMA Channels. The Receive DMA Channels is identified by the DCS field of MAC_ADDRESS(#i)_HIGH register corresponding to the MAC Address register that matches the Multicast/Broadcast Destination address in the received packet. The DCS field is interpreted to be a one-hot value, each bit corresponding to the Receive DMA Channel. When this bit is reset, the received packet is routed to single Receive DMA Channel. The Receive DMA Channel is identified by the DCS field of ADDRESS(#i)_High register corresponding to the MAC Address register that matches the Destination address in the received packet. The DCS field is interpreted as a binary value. 0 - Packet Duplication Control is disabled 1 - Packet Duplication Control is enabled
18 USP	Unicast Slow Protocol Packet Detect When this bit is set, the MAC detects the Slow Protocol packets with unicast address of the station specified in the MAC_ADDRESS0_HIGH and MAC_ADDRESS0_LOW registers. The MAC also detects the Slow Protocol packets with the Slow Protocols multicast address (01-80-C2-00-00-02). When this bit is reset, the MAC detects only Slow Protocol packets with the Slow Protocol multicast address specified in the IEEE 802.3-2015, Section 5. 0 - Unicast Slow Protocol Packet Detection is disabled 1 - Unicast Slow Protocol Packet Detection is enabled
17 SPEN	Slow Protocol Detection Enable When this bit is set, MAC processes the Slow Protocol packets (Ether Type 0x8809) and provides the Rx status. The MAC discards the Slow Protocol packets with invalid sub-types. When this bit is reset, the MAC forwards all error-free Slow Protocol packets to the application. The MAC considers such packets as normal Type packets. 0 - Slow Protocol Detection is disabled 1 - Slow Protocol Detection is enabled
16 DCRCC	Disable CRC Checking for Received Packets When this bit is set, the MAC receiver does not check the CRC field in the received packets. When this bit is reset, the MAC receiver always checks the CRC field in the received packets. 0 - CRC Checking is enabled 1 - CRC Checking is disabled
15-14 —	Reserved
13-0 GPSL	Giant Packet Size Limit If the received packet size is greater than the value programmed in this field in units of bytes, the MAC declares the received packet as Giant packet. The value programmed in this field must be greater than or equal to 1,518 bytes. Any other programmed value is considered as 1,518 bytes. For VLAN tagged packets, the MAC adds 4 bytes to the programmed value. When the Enable Double VLAN Processing option is selected, the MAC adds 8 bytes to the programmed value for double VLAN tagged packets. The value in this field is applicable when the GPSLCE bit is set in MAC_CONFIGURATION register.

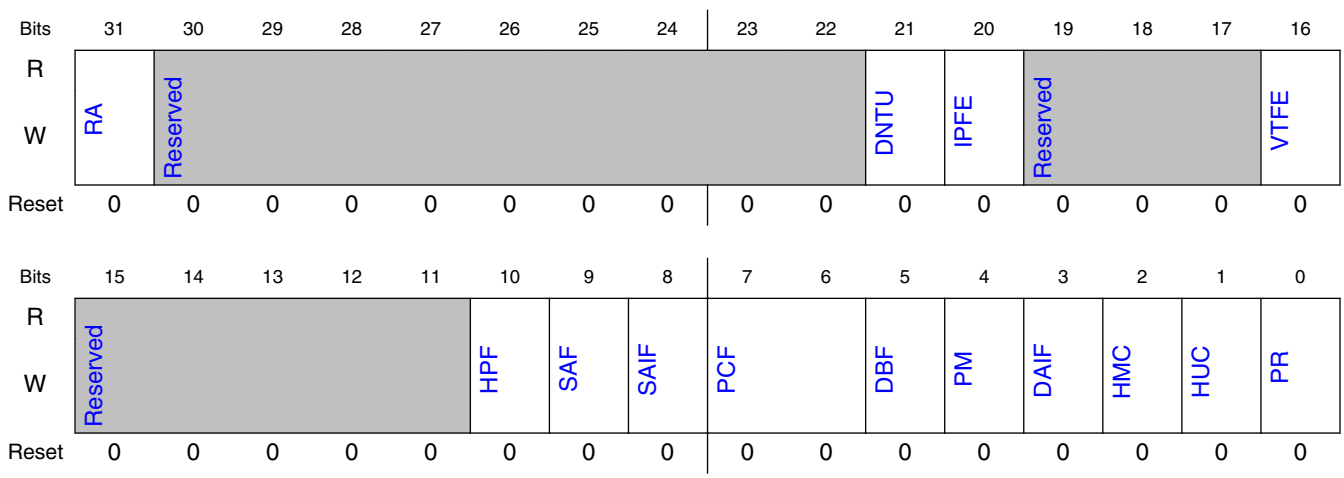
11.7.6.1.4 MAC Packet Filter (MAC_PACKET_FILTER)

The MAC Packet Filter register contains the filter controls for receiving packets. Some of the controls from this register go to the address check block of the MAC which performs the first level of address filtering. The second level of filtering is performed on the incoming packet based on other controls such as Pass Bad Packets and Pass Control Packets.

11.7.6.1.4.1 Offset

Register	Offset
MAC_PACKET_FILTER	8h

11.7.6.1.4.2 Diagram



11.7.6.1.4.3 Fields

Field	Description
31 RA	Receive All When this bit is set, the MAC Receiver module passes all received packets to the application, irrespective of whether they pass the address filter or not. The result of the SA or DA filtering is updated (pass or fail) in the corresponding bit in the Rx Status Word. When this bit is reset, the Receiver module passes only those packets to the application that pass the SA or DA address filter. 0 - Receive All is disabled 1 - Receive All is enabled

Table continues on the next page...

Field	Description
30-22 —	Reserved
21 DNTU	Drop Non-TCP/UDP over IP Packets When this bit is set, the MAC drops the non-TCP or UDP over IP packets. The MAC forward only those packets that are processed by the Layer 4 filter. When this bit is reset, the MAC forwards all non-TCP or UDP over IP packets. 0 - Forward Non-TCP/UDP over IP Packets 1 - Drop Non-TCP/UDP over IP Packets
20 IPFE	Layer 3 and Layer 4 Filter Enable When this bit is set, the MAC drops packets that do not match the enabled Layer 3 and Layer 4 filters. If Layer 3 or Layer 4 filters are not enabled for matching, this bit does not have any effect. When this bit is reset, the MAC forwards all packets irrespective of the match status of the Layer 3 and Layer 4 fields. 0 - Layer 3 and Layer 4 Filters are disabled 1 - Layer 3 and Layer 4 Filters are enabled
19-17 —	Reserved
16 VTFE	VLAN Tag Filter Enable When this bit is set, the MAC drops the VLAN tagged packets that do not match the VLAN Tag. When this bit is reset, the MAC forwards all packets irrespective of the match status of the VLAN Tag. 0 - VLAN Tag Filter is disabled 1 - VLAN Tag Filter is enabled
15-11 —	Reserved
10 HPF	Hash or Perfect Filter When this bit is set, the address filter passes a packet if it matches either the perfect filtering or hash filtering as set by the HMC or HUC bit. When this bit is reset and the HUC or HMC bit is set, the packet is passed only if it matches the Hash filter. 0 - Hash or Perfect Filter is disabled 1 - Hash or Perfect Filter is enabled
9 SAF	Source Address Filter Enable When this bit is set, the MAC compares the SA field of the received packets with the values programmed in the enabled SA registers. If the comparison fails, the MAC drops the packet. When this bit is reset, the MAC forwards the received packet to the application with updated SAF bit of the Rx Status depending on the SA address comparison. Note: According to the IEEE specification, Bit 47 of the SA is reserved. However, in DWC_ether_qos, the MAC compares all 48 bits. The software driver should take this into consideration while programming the MAC address registers for SA. 0 - SA Filtering is disabled 1 - SA Filtering is enabled
8	SA Inverse Filtering

Table continues on the next page...

Field	Description
SAIF	When this bit is set, the Address Check block operates in the inverse filtering mode for SA address comparison. If the SA of a packet matches the values programmed in the SA registers, it is marked as failing the SA Address filter. When this bit is reset, if the SA of a packet does not match the values programmed in the SA registers, it is marked as failing the SA Address filter. 0 - SA Inverse Filtering is disabled 1 - SA Inverse Filtering is enabled
7-6 PCF	Pass Control Packets These bits control the forwarding of all control packets (including unicast and multicast Pause packets). 00 - MAC filters all control packets from reaching the application 01 - MAC forwards all control packets except Pause packets to the application even if they fail the Address filter 10 - MAC forwards all control packets to the application even if they fail the Address filter 11 - MAC forwards the control packets that pass the Address filter
5 DBF	Disable Broadcast Packets When this bit is set, the AFM module blocks all incoming broadcast packets. In addition, it overrides all other filter settings. When this bit is reset, the AFM module passes all received broadcast packets. 0 - Enable Broadcast Packets 1 - Disable Broadcast Packets
4 PM	Pass All Multicast When this bit is set, it indicates that all received packets with a multicast destination address (first bit in the destination address field is '1') are passed. When this bit is reset, filtering of multicast packet depends on HMC bit. 0 - Pass All Multicast is disabled 1 - Pass All Multicast is enabled
3 DAIF	DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast packets. When this bit is reset, normal filtering of packets is performed. 0 - DA Inverse Filtering is disabled 1 - DA Inverse Filtering is enabled
2 HMC	Hash Multicast When this bit is set, the MAC performs the destination address filtering of received multicast packets according to the hash table. When this bit is reset, the MAC performs the perfect destination address filtering for multicast packets, that is, it compares the DA field with the values programmed in DA registers. 0 - Hash Multicast is disabled 1 - Hash Multicast is enabled
1 HUC	Hash Unicast When this bit is set, the MAC performs the destination address filtering of unicast packets according to the hash table.

Table continues on the next page...

Field	Description
	When this bit is reset, the MAC performs a perfect destination address filtering for unicast packets, that is, it compares the DA field with the values programmed in DA registers. 0 - Hash Unicast is disabled 1 - Hash Unicast is enabled
0 PR	Promiscuous Mode When this bit is set, the Address Filtering module passes all incoming packets irrespective of the destination or source address. The SA or DA Filter Fails status bits of the Rx Status Word are always cleared when PR is set. 0 - Promiscuous Mode is disabled 1 - Promiscuous Mode is enabled

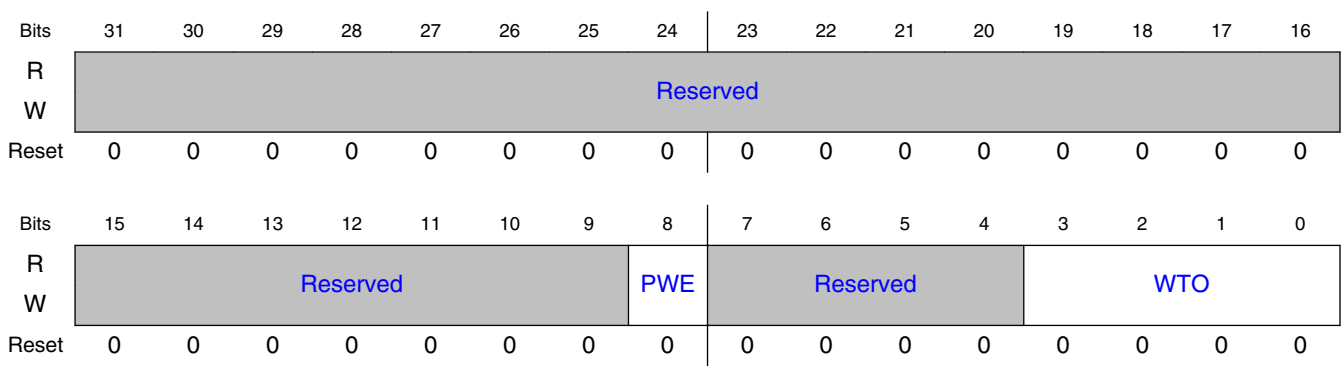
11.7.6.1.5 Watchdog Timeout (MAC_WATCHDOG_TIMEOUT)

The Watchdog Timeout register controls the watchdog timeout for received packets.

11.7.6.1.5.1 Offset

Register	Offset
MAC_WATCHDOG_TIMEOUT	Ch

11.7.6.1.5.2 Diagram



11.7.6.1.5.3 Fields

Field	Description
31-9 —	Reserved

Table continues on the next page...

Field	Description
8 PWE	Programmable Watchdog Enable When this bit is set and the WD bit of the MAC_CONFIGURATION register is reset, the WTO field is used as watchdog timeout for a received packet. When this bit is cleared, the watchdog timeout for a received packet is controlled by setting of WD and JE bits in MAC_CONFIGURATION register. 0 - Programmable Watchdog is disabled 1 - Programmable Watchdog is enabled
7-4 —	Reserved
3-0 WTO	Watchdog Timeout When the PWE bit is set and the WD bit of the MAC_CONFIGURATION register is reset, this field is used as watchdog timeout for a received packet. If the length of a received packet exceeds the value of this field, such packet is terminated and declared as an error packet. Note: When the PWE bit is set, the value in this field should be more than 1,522 (0x05F2). Otherwise, the IEEE 802.3-specified valid tagged packets are declared as error packets and then dropped. 0000 - 2 KB 0001 - 3 KB 0010 - 4 KB 0011 - 5 KB 0100 - 6 KB 0101 - 7 KB 0110 - 8 KB 0111 - 9 KB 1000 - 10 KB 1001 - 11 KB 1010 - 12 KB 1011 - 13 KB 1100 - 14 KB 1101 - 15 KB 1110 - 16383 Bytes 1111 - Reserved

11.7.6.1.6 MAC Hash Table Register 0 (MAC_HASH_TABLE_REG0)

The Hash Table Register 0 contains the first 32 bits of the hash table, when the width of the hash table is 128 or 256 bits. You can specify the width of the hash table by using the Hash Table Size option in coreConsultant.

The Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming packet is passed through the CRC logic and the upper six (seven in 128-bit Hash or eight in 256-bit Hash) bits of the CRC register are used to index the content of the Hash table. The most significant bits determines the register to be used (Hash Table Register X), and the least significant five bits determine the bit within the register. For example, a hash value of 6'b100000 (in 64-bit Hash) selects Bit 0 of the Hash Table Register 1, a value of 7b'1110000 (in 128-bit Hash) selects Bit 16 of the Hash Table Register 3 and a value of 8b'10111111 (in 256-bit Hash) selects Bit 31 of the Hash Table Register 5.

The hash value of the destination address is calculated in the following way:

- Calculate the 32-bit CRC for the DA (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).

- Perform bitwise reversal for the value obtained in Step 1.

- Take the upper 6 (or 7 or 8) bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1'b1, the packet is accepted. Otherwise, it is rejected. If the PM bit is set in MAC_Packet_Filter, all multicast packets are accepted regardless of the multicast hash values.

If the Hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Hash Table Register X registers are written.

If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.6.1 Offset

Register	Offset
MAC_HASH_TABLE_REG0	10h

11.7.6.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HT31T0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HT31T0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.6.3 Fields

Field	Description
31-0 HT31T0	MAC Hash Table First 32 Bits This field contains the first 32 Bits [31:0] of the Hash table.

11.7.6.1.7 MAC Hash Table Register 1 (MAC_HASH_TABLE_REG1)

The Hash Table Register 1 contains the second 32 bits of the hash table. You can specify the width of the hash table by using the Hash Table Size option in coreConsultant.

The Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming packet is passed through the CRC logic and the upper six (seven in 128-bit Hash or eight in 256-bit Hash) bits of the CRC register are used to index the content of the Hash table. The most significant bits determines the register to be used (Hash Table Register X), and the least significant five bits determine the bit within the register. For example, a hash value of 6'b100000 (in 64-bit Hash) selects Bit 0 of the Hash Table Register 1, a value of 7b'1110000 (in 128-bit Hash) selects Bit 16 of the Hash Table Register 3 and a value of 8b'10111111 (in 256-bit Hash) selects Bit 31 of the Hash Table Register 5.

The hash value of the destination address is calculated in the following way:

- Calculate the 32-bit CRC for the DA (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).

- Perform bitwise reversal for the value obtained in Step 1.

- Take the upper 6 (or 7 or 8) bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1'b1, the packet is accepted. Otherwise, it is rejected. If the PM bit is set in MAC_Packet_Filter, all multicast packets are accepted regardless of the multicast hash values.

If the Hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Hash Table Register X registers are written.

If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.7.1 Offset

Register	Offset
MAC_HASH_TABLE_REG1	14h

11.7.6.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HT63T32															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HT63T32															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.7.3 Fields

Field	Description
31-0 HT63T32	MAC Hash Table Second 32 Bits This field contains the second 32 Bits [63:32] of the Hash table.

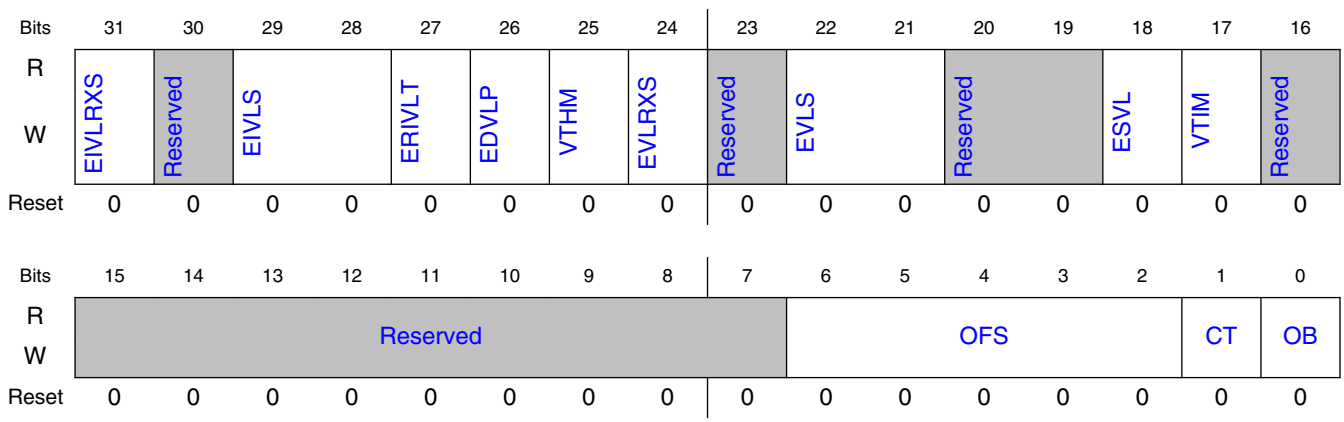
11.7.6.1.8 MAC VLAN Tag Control (MAC_VLAN_TAG_CTRL)

This register is the redefined format of the MAC VLAN Tag Register. It is used for indirect addressing. It contains the address offset, command type and Busy Bit for CSR access of the Per VLAN Tag registers.

11.7.6.1.8.1 Offset

Register	Offset
MAC_VLAN_TAG_CTRL	50h

11.7.6.1.8.2 Diagram



11.7.6.1.8.3 Fields

Field	Description
31 EIVLRXS	Enable Inner VLAN Tag in Rx Status When this bit is set, the MAC provides the inner VLAN Tag in the Rx status. When this bit is reset, the MAC does not provide the inner VLAN Tag in Rx status. 0 - Inner VLAN Tag in Rx status is disabled 1 - Inner VLAN Tag in Rx status is enabled
30 —	Reserved
29-28 EIVLS	Enable Inner VLAN Tag Stripping on Receive This field indicates the stripping operation on inner VLAN Tag in received packet. 00 - Do not strip 01 - Strip if VLAN filter passes 10 - Strip if VLAN filter fails 11 - Always strip
27	ERIVLT

Table continues on the next page...

Field	Description
ERIVLT	0 - Inner VLAN tag is disabled 1 - Inner VLAN tag is enabled
26 EDVLP	Enable Double VLAN Processing When this bit is set, the MAC enables processing of up to two VLAN Tags on Tx and Rx (if present). When this bit is reset, the MAC enables processing of up to one VLAN Tag on Tx and Rx (if present). 0 - Double VLAN Processing is disabled 1 - Double VLAN Processing is enabled
25 VTHM	VLAN Tag Hash Table Match Enable When this bit is set, the most significant four bits of CRC of VLAN Tag (ones-complement of most significant four bits of CRC of VLAN Tag when ETV bit is reset) are used to index the content of the MAC_VLAN_Hash_Table register. A value of 1 in the VLAN Hash Table register, corresponding to the index, indicates that the packet matched the VLAN hash table. When the ETV bit is set, the CRC of the 12-bit VLAN Identifier (VID) is used for comparison. When the ETV bit is reset, the ones-complement of the CRC of the 16-bit VLAN tag is used for comparison. When this bit is reset, the VLAN Hash Match operation is not performed. 0 - VLAN Tag Hash Table Match is disabled 1 - VLAN Tag Hash Table Match is enabled
24 EVLRXS	Enable VLAN Tag in Rx status When this bit is set, MAC provides the outer VLAN Tag in the Rx status. When this bit is reset, the MAC does not provide the outer VLAN Tag in Rx status. 0 - VLAN Tag in Rx status is disabled 1 - VLAN Tag in Rx status is enabled
23 —	Reserved
22-21 EVLS	Enable VLAN Tag Stripping on Receive This field indicates the stripping operation on the outer VLAN Tag in received packet. 00 - Do not strip 01 - Strip if VLAN filter passes 10 - Strip if VLAN filter fails 11 - Always strip
20-19 —	Reserved
18 ESVL	Enable S-VLAN When this bit is set, the MAC transmitter and receiver consider the S-VLAN packets (Type = 0x88A8) as valid VLAN tagged packets. 0 - S-VLAN is disabled 1 - S-VLAN is enabled
17 VTIM	VLAN Tag Inverse Match Enable When this bit is set, this bit enables the VLAN Tag inverse matching. The packets without matching VLAN Tag are marked as matched. When reset, this bit enables the VLAN Tag perfect matching. The packets with matched VLAN Tag are marked as matched. 0 - VLAN Tag Inverse Match is disabled 1 - VLAN Tag Inverse Match is enabled

Table continues on the next page...

Field	Description
16-7 —	Reserved
6-2 OFS	Offset This field holds the address offset of the MAC VLAN Tag Filter Register which the application is trying to access. The width of the field depends on the number of MAC VLAN Tag Registers enabled.
1 CT	Command Type This bit indicates if the current register access is a read or a write. When set, it indicate a read operation. When reset, it indicates a write operation. 0 - Write operation 1 - Read operation
0 OB	Operation Busy This bit is set along with a read or write command for initiating the indirect access to per VLAN Tag Filter register. This bit is reset when the read or write command to per VLAN Tag Filter indirect access register is complete. The next indirect register access can be initiated only after this bit is reset. During a write operation, the bit is reset only after the data has been written into the Per VLAN Tag register. During a read operation, the data should be read from the VLAN_TAG_DATA register only after this bit is reset. 0 - Operation Busy is disabled 1 - Operation Busy is enabled

11.7.6.1.9 MAC VLAN Tag Data (MAC_VLAN_TAG_DATA)

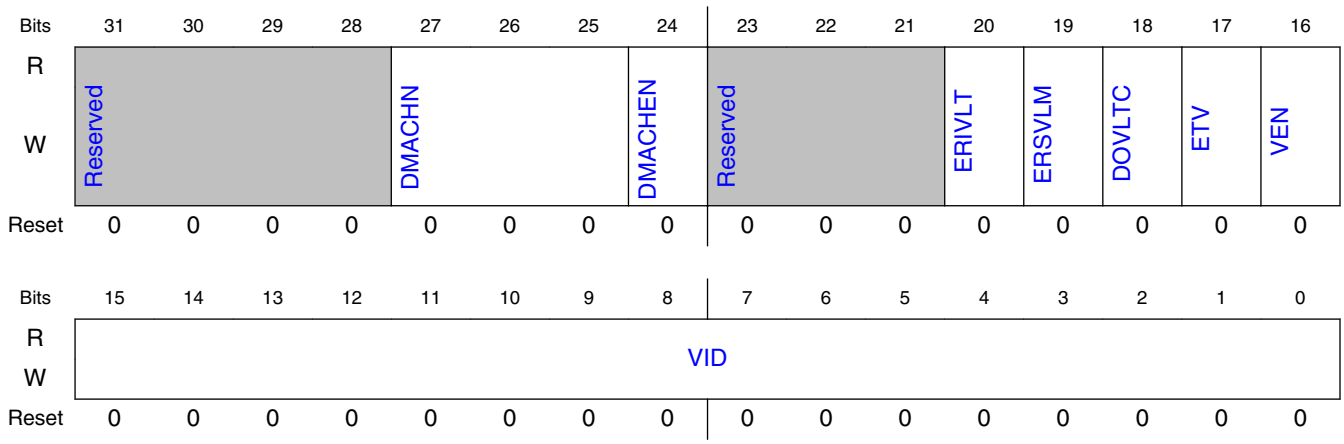
This register holds the read/write data for Indirect Access of the Per VLAN Tag registers. During the read access, this field contains valid read data only after the OB bit is reset.

During the write access, this field should be valid prior to setting the OB bit in the MAC_VLAN_Tag_Ctrl Register.

11.7.6.1.9.1 Offset

Register	Offset
MAC_VLAN_TAG_DATA	54h

11.7.6.1.9.2 Diagram



11.7.6.1.9.3 Fields

Field	Description
31-28 —	Reserved
27-25 DMACHN	DMA Channel Number The DMA Channel number to which the VLAN Tagged Frame is to be routed if it passes this VLAN Tag Filter is programmed in this field. If the Routing based on VLAN Tag Filter is not necessary, this field need not be programmed.
24 DMACHEN	DMA Channel Number Enable This bit is the Enable for the DMA Channel Number value programmed in the field DMACH. When this bit is reset, the Routing does not occur based on VLAN Filter result. The frame is routed based on DA Based DMA Channel Routing. 0 - DMA Channel Number is disabled 1 - DMA Channel Number is enabled
23-21 —	Reserved
20 ERIVLT	Enable Inner VLAN Tag Comparison This bit is valid only when VLAN Tag Enable of the Filter is set. When this bit and the EDVLP field are set, the MAC receiver enables operation on the inner VLAN Tag (if present). When this bit is reset, the MAC receiver enables operation on the outer VLAN Tag (if present). 0 - Inner VLAN tag comparison is disabled 1 - Inner VLAN tag comparison is enabled
19 ERSVLM	Enable S-VLAN Match for received Frames This bit is valid only when VLAN Tag Enable of the Filter is set. When this bit is set, the MAC receiver enables filtering or matching for S-VLAN (Type = 0x88A8) packets.

Table continues on the next page...

Field	Description
	When this bit is reset, the MAC receiver enables filtering or matching for C-VLAN (Type = 0x8100) packets. 0 - Receive S-VLAN Match is disabled 1 - Receive S-VLAN Match is enabled
18 DOVLTC	Disable VLAN Type Comparison This bit is valid only when VLAN Tag Enable of the Filter is set. When this bit is set, the MAC does not check whether the VLAN Tag specified by the Enable Inner VLAN Tag Comparison bit is of type S-VLAN or C-VLAN. When this bit is reset, the MAC filters or matches the VLAN Tag specified by the Enable Inner VLAN Tag Comparison bit only when VLAN Tag type is similar to the one specified by the Enable S-VLAN Match for received Frames bit. 0 - VLAN type comparison is enabled 1 - VLAN type comparison is disabled
17 ETV	12bits or 16bits VLAN comparison This bit is valid only when VEN of the Filter is set. When this bit is set, a 12-bit VLAN identifier is used for comparing and filtering instead of the complete 16-bit VLAN tag. Bits [11:0] of VLAN tag are compared with the corresponding field in the received VLAN-tagged packet. 0 - 16 bit VLAN comparison 1 - 12 bit VLAN comparison
16 VEN	VLAN Tag Enable This bit is used to enable or disable the VLAN Tag. When this bit is set, the MAC compares the VLAN Tag of received packet with the VLAN Tag ID. When this bit is reset, no comparison is performed irrespective of the programming of the other fields. 0 - VLAN Tag is disabled 1 - VLAN Tag is enabled
15-0 VID	VLAN Tag ID This field holds the VLAN Tag value which is used by the MAC for perfect comparison. It is valid when VLAN Tag Enable is set.

11.7.6.1.10 MAC VLAN Hash Table (MAC_VLAN_HASH_TABLE)

When VTHM bit of the MAC_VLAN_TAG register is set, the 16-bit VLAN Hash Table register is used for group address filtering based on the VLAN tag. For hash filtering, the content of the 16-bit VLAN tag or 12-bit VLAN ID (based on the ETV bit of MAC_VLAN_TAG Register) in the incoming packet is passed through the CRC logic. When ETV bit of MAC_VLAN_TAG register is set, the upper four bits of the calculated CRC are used to index the contents of the VLAN Hash table. When ETV bit of MAC_VLAN_TAG register is reset, the ones-complement of upper four bits of the calculated CRC are used to index the contents of the VLAN Hash table. For example,

when ETV bit is set a hash value of 4b'1000 selects Bit 8 of the VLAN Hash table, whereas when ETV bit is reset a hash value of 4b'1000 selects Bit 7 of the VLAN Hash table.

The hash value of the destination address is calculated in the following way:

- Calculate the 32-bit CRC for the VLAN tag or ID (For steps to calculate CRC32, see Section 3.2.8 of IEEE 802.3).

- Perform bitwise reversal for the value obtained in step 1.

- Take the upper four bits from the value obtained in step 2.

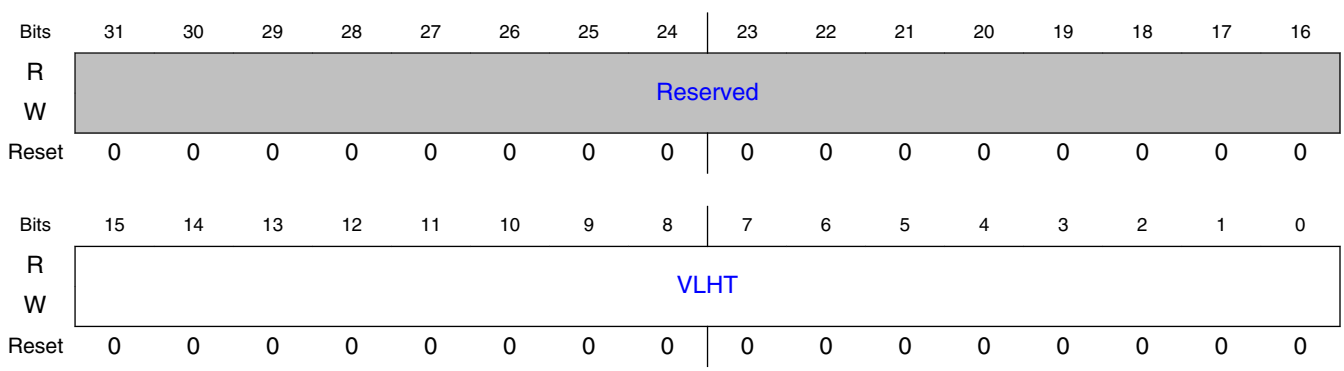
If the VLAN hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[15:8] (in little-endian mode) or Bits[7:0] (in big-endian mode) of this register are written.

- If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.10.1 Offset

Register	Offset
MAC_VLAN_HASH_TABLE	58h

11.7.6.1.10.2 Diagram



11.7.6.1.10.3 Fields

Field	Description
31-16	Reserved

Table continues on the next page...

Field	Description
—	
15-0 VLHT	VLAN Hash Table This field contains the 16-bit VLAN Hash Table.

11.7.6.1.11 VLAN Tag Inclusion or Replacement (MAC_VLAN_INCL)

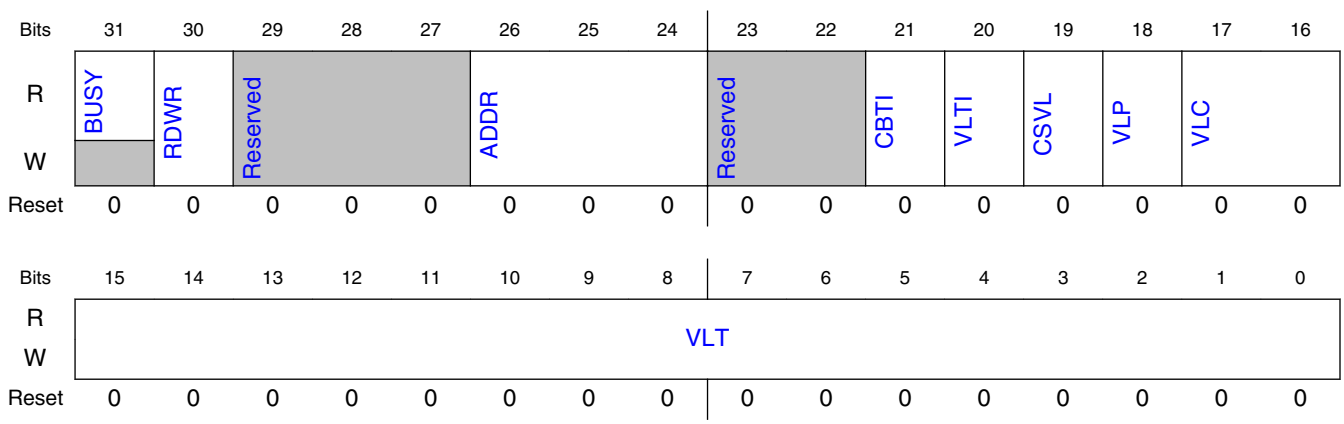
The VLAN Tag Inclusion or Replacement register contains the VLAN tag for insertion or replacement in the Transmit packets.

The VLAN Tag Inclusion or Replacement register contains the VLAN tag for insertion or replacement in the Transmit packets. It also contains the VLAN tag insertion controls.

11.7.6.1.11.1 Offset

Register	Offset
MAC_VLAN_INCL	60h

11.7.6.1.11.2 Diagram



11.7.6.1.11.3 Fields

Field	Description
31 BUSY	Busy This bit indicates the status of the read/write operation of indirect access to the queue/channel specific VLAN inclusion register. For write operation write to a register is complete when this bit is reset. For read operation the read data is valid when the bit is reset.

Table continues on the next page...

Field	Description
	The application must make sure that this bit is reset before attempting subsequent access to this register. 0 - Busy status not detected 1 - Busy status detected
30 RDWR	Read write control This bit controls the read or write operation for indirectly accessing the queue/channel specific VLAN Inclusion register. When set indicates write operation and when reset indicates read operation. This does not have any effect when CBTI is reset. 0 - Read operation of indirect access 1 - Write operation of indirect access
29-27 —	Reserved
26-24 ADDR	Address This field selects one of the queue/channel specific VLAN Inclusion register for read/write access. This does not have any effect when CBTI is reset.
23-22 —	Reserved
21 CBTI	Channel based tag insertion When this bit is set, outer VLAN tag is inserted for every packets transmitted by the MAC. The tag value is taken from the queue/channel specific VLAN tag register. The VLTI, VLP, VLC, and VLT fields of this register are ignored when this bit is set. When this bit is set, a write operation to byte 3 of this register initiates the read/write access to the indirect register. When reset, outer VLAN operation is based on the setting of VLTI, VLP, VLC and VLT fields of this register. 0 - Channel based tag insertion is disabled 1 - Channel based tag insertion is enabled
20 VLTI	VLAN Tag Input When this bit is set, it indicates that the VLAN tag to be inserted or replaced in Tx packet should be taken from: - The Tx descriptor 0 - VLAN Tag Input is disabled 1 - VLAN Tag Input is enabled
19 CSVL	C-VLAN or S-VLAN When this bit is set, S-VLAN type (0x88A8) is inserted or replaced in the 13th and 14th bytes of transmitted packets. When this bit is reset, C-VLAN type (0x8100) is inserted or replaced in the 13th and 14th bytes of transmitted packets. 0 - C-VLAN type (0x8100) is inserted or replaced 1 - S-VLAN type (0x88A8) is inserted or replaced
18 VLP	VLAN Priority Control When this bit is set, the control bits[17:16] are used for VLAN deletion, insertion, or replacement. When this bit is reset, the mti_vlan_ctrl_i control input is used and bits[17:16] are ignored. 0 - VLAN Priority Control is disabled 1 - VLAN Priority Control is enabled

Table continues on the next page...

Field	Description
17-16 VLC	VLAN Tag Control in Transmit Packets - 2'b00: No VLAN tag deletion, insertion, or replacement - 2'b01: VLAN tag deletion The MAC removes the VLAN type (bytes 13 and 14) and VLAN tag (bytes 15 and 16) of all transmitted packets with VLAN tags. VLAN Tag Control in Transmit Packets - 2'b00: No VLAN tag deletion, insertion, or replacement - 2'b01: VLAN tag deletion The MAC removes the VLAN type (bytes 13 and 14) and VLAN tag (bytes 15 and 16) of all transmitted packets with VLAN tags. - 2'b10: VLAN tag insertion The MAC inserts VLT in bytes 15 and 16 of the packet after inserting the Type value (0x8100 or 0x88a8) in bytes 13 and 14. This operation is performed on all transmitted packets, irrespective of whether they already have a VLAN tag. - 2'b11: VLAN tag replacement The MAC replaces VLT in bytes 15 and 16 of all VLAN-type transmitted packets (Bytes 13 and 14 are 0x8100 or 0x88a8). Note: Changes to this field take effect only on the start of a packet. If you write this register field when a packet is being transmitted, only the subsequent packet can use the updated value, that is, the current packet does not use the updated value. 00 - No VLAN tag deletion, insertion, or replacement 01 - VLAN tag deletion 10 - VLAN tag insertion 11 - VLAN tag replacement
15-0 VLT	VLAN Tag for Transmit Packets This field contains the value of the VLAN tag to be inserted or replaced. The value must only be changed when the transmit lines are inactive or during the initialization phase. Bits[15:13] are the User Priority field, Bit 12 is the CFI/DEI field, and Bits[11:0] are the VID field in the VLAN tag. The following list describes the bits of this field: - Bits[15:13]: User Priority - Bit 12: Canonical Format Indicator (CFI) or Drop Eligible Indicator (DEI) - Bits[11:0]: VLAN Identifier (VID) field of VLAN tag

11.7.6.1.12 MAC Inner VLAN Tag Inclusion or Replacement (MAC_INNER_VLAN_INCL)

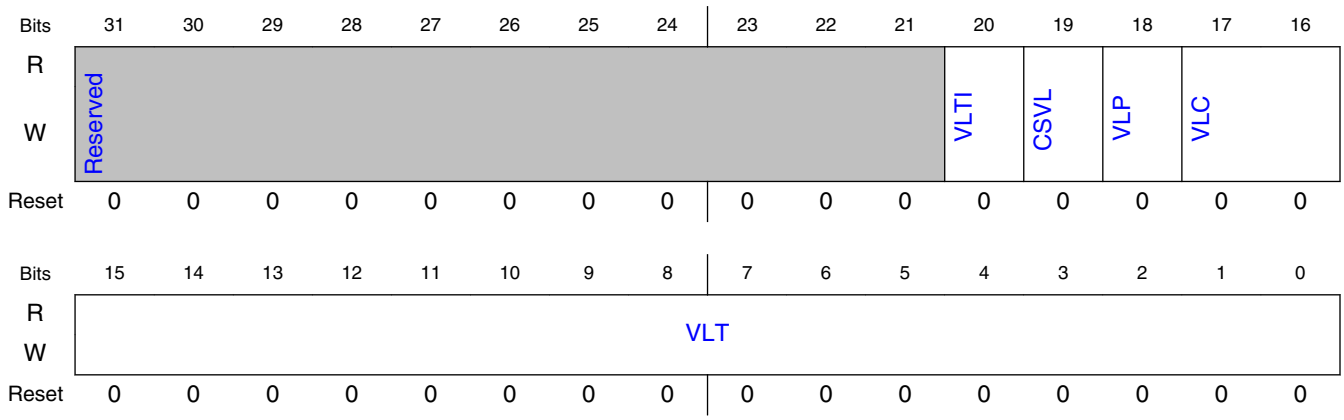
The Inner VLAN Tag Inclusion or Replacement register contains the inner VLAN tag to be inserted or replaced in the Transmit packet.

The Inner VLAN Tag Inclusion or Replacement register contains the inner VLAN tag to be inserted or replaced in the Transmit packet. It also contains the inner VLAN tag insertion controls.

11.7.6.1.12.1 Offset

Register	Offset
MAC_INNER_VLAN_INCL	64h

11.7.6.1.12.2 Diagram



11.7.6.1.12.3 Fields

Field	Description
31-21 —	Reserved
20 VLTI	VLAN Tag Input When this bit is set, it indicates that the VLAN tag to be inserted or replaced in Tx packet should be taken from: - The Tx descriptor 0 - VLAN Tag Input is disabled 1 - VLAN Tag Input is enabled
19 CSVL	C-VLAN or S-VLAN When this bit is set, S-VLAN type (0x88A8) is inserted or replaced in the 13th and 14th bytes of transmitted packets. When this bit is reset, C-VLAN type (0x8100) is inserted or replaced in the 13th and 14th bytes of transmitted packets. 0 - C-VLAN type (0x8100) is inserted 1 - S-VLAN type (0x88A8) is inserted
18 VLP	VLAN Priority Control When this bit is set, the VLC field is used for VLAN deletion, insertion, or replacement. When this bit is reset, the mti_vlan_ctrl_i control input is used and the VLC field is ignored. 0 - VLAN Priority Control is disabled 1 - VLAN Priority Control is enabled
17-16	VLAN Tag Control in Transmit Packets

Table continues on the next page...

Field	Description
VLC	- 2'b00: No VLAN tag deletion, insertion, or replacement - 2'b01: VLAN tag deletion The MAC removes the VLAN type (bytes 17 and 18) and VLAN tag (bytes 19 and 20) of all transmitted packets with VLAN tags. - 2'b10: VLAN tag insertion The MAC inserts VLT in bytes 19 and 20 of the packet after inserting the Type value (0x8100 or 0x88a8) in bytes 17 and 18. This operation is performed on all transmitted packets, irrespective of whether they already have a VLAN tag. - 2'b11: VLAN tag replacement The MAC replaces VLT in bytes 19 and 20 of all VLAN-type transmitted packets (Bytes 17 and 18 are 0x8100 or 0x88a8). Note: Changes to this field take effect only on the start of a packet. If you write this register field when a packet is being transmitted, only the subsequent packet can use the updated value, that is, the current packet does not use the updated value. 00 - No VLAN tag deletion, insertion, or replacement 01 - VLAN tag deletion 10 - VLAN tag insertion 11 - VLAN tag replacement
15-0 VLT	VLAN Tag for Transmit Packets This field contains the value of the VLAN tag to be inserted or replaced. The value must only be changed when the transmit lines are inactive or during the initialization phase. Bits[15:13] are the User Priority field, Bit 12 is the CFI/DEI field, and Bits[11:0] are the VID field in the VLAN tag. The following list describes the bits of this field: - Bits[15:13]: User Priority - Bit 12: Canonical Format Indicator (CFI) or Drop Eligible Indicator (DEI) - Bits[11:0]: VLAN Identifier (VID) field of VLAN tag

11.7.6.1.13 MAC Q0 Tx Flow Control (MAC_Q0_TX_FLOW_CTRL)

The Flow Control register controls the generation and reception of the Control (Pause Command) packets by the Flow control module of the MAC.

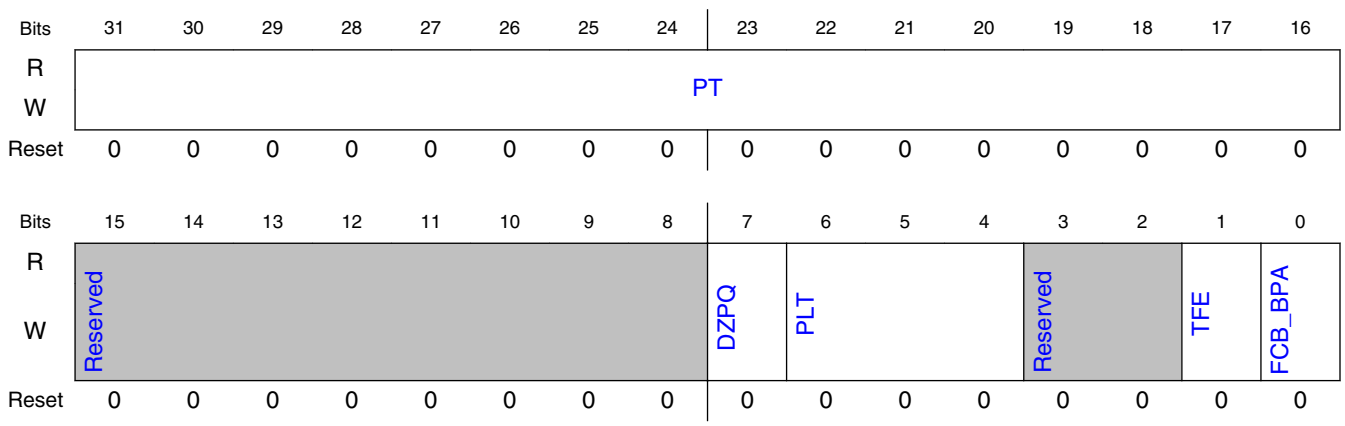
The Flow Control register controls the generation and reception of the Control (Pause Command) packets by the Flow control module of the MAC. A Write to a register with the Busy bit set to 1 triggers the Flow Control block to generate a Pause packet. The fields of the control packet are selected as specified in the 802.3x specification, and the Pause Time value from this register is used in the Pause Time field of the control packet. The Busy bit remains set until the control packet is transferred onto the cable. The application must make sure that the Busy bit is cleared before writing to the register.

When the PFCE bit in the MAC_Q0_RX_FLOW_CTRL register is enabled, this register controls the generation of Priority Flow Control (PFC) frames with priorities mapped according to PSRQ0 in the MAC_Q0_RXQ_CTRL2 register.

11.7.6.1.13.1 Offset

Register	Offset
MAC_Q0_TX_FLOW_CTRL	70h

11.7.6.1.13.2 Diagram



11.7.6.1.13.3 Fields

Field	Description
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0 - Zero-Quanta Pause packet generation is enabled 1 - Zero-Quanta Pause packet generation is disabled
6-4	Pause Low Threshold

Table continues on the next page...

Field	Description
PLT	This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000 - Pause Time minus 4 Slot Times (PT -4 slot times) 001 - Pause Time minus 28 Slot Times (PT -28 slot times) 010 - Pause Time minus 36 Slot Times (PT -36 slot times) 011 - Pause Time minus 144 Slot Times (PT -144 slot times) 100 - Pause Time minus 256 Slot Times (PT -256 slot times) 101 - Pause Time minus 512 Slot Times (PT -512 slot times) 110 - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable Full-Duplex Mode: In the full-duplex mode, when this bit is set, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. Half-Duplex Mode: In the half-duplex mode, when this bit is set, the MAC enables the backpressure operation. When this bit is reset, the backpressure feature is disabled. 0 - Transmit Flow Control is disabled 1 - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy or Backpressure Activate This bit initiates a Pause packet in the full-duplex mode and activates the backpressure function in the half-duplex mode if the TFE bit is set. Full-Duplex Mode: In the full-duplex mode, this bit should be read as 1'b0 before writing to this register. To initiate a Pause packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When Pause packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Half-Duplex Mode: When this bit is set (and TFE bit is set) in the half-duplex mode, the MAC asserts the backpressure. During backpressure, when the MAC receives a new packet, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically ORed with the mti_flowctrl_i input signal for the backpressure function. When the MAC is configured for the full-duplex mode, the BPA is automatically disabled. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flow Control Busy or Backpressure Activate is disabled

Field	Description
	1 - Flow Control Busy or Backpressure Activate is enabled

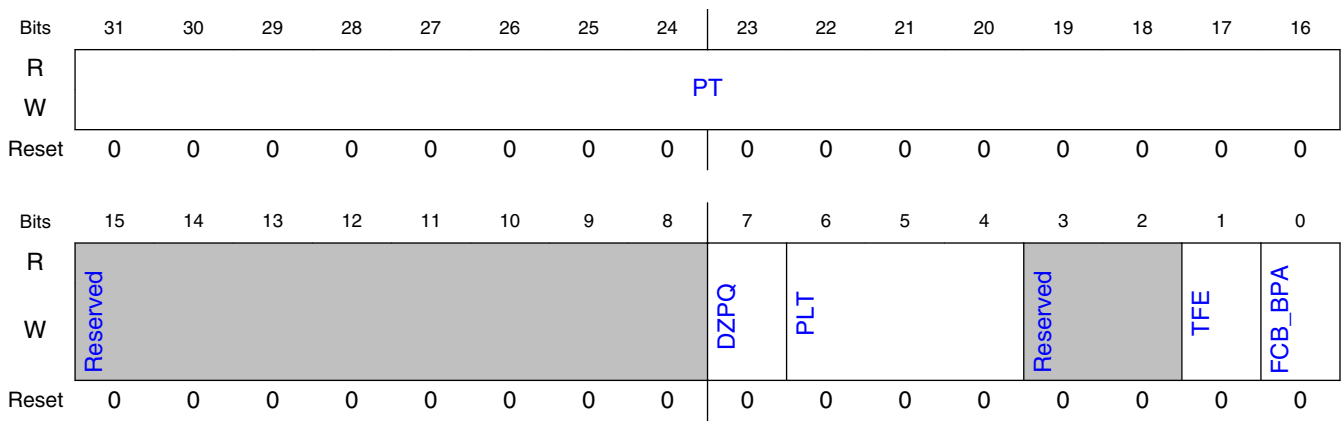
11.7.6.1.14 MAC Q1 Tx Flow Control (MAC_Q1_TX_FLOW_CTRL)

This register controls the generation of PFC Control packets of priorities mapped as per the PSRQi field in the MAC_Q1_RXQ_CTRL2/MAC_Q1_RXQ_CTRL3 registers.

11.7.6.1.14.1 Offset

Register	Offset
MAC_Q1_TX_FLOW_CTRL	74h

11.7.6.1.14.2 Diagram



11.7.6.1.14.3 Fields

Field	Description
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause

Table continues on the next page...

Field	Description
	When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0 - Zero-Quanta Pause packet generation is enabled 1 - Zero-Quanta Pause packet generation is disabled
6-4 PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000 - Pause Time minus 4 Slot Times (PT -4 slot times) 001 - Pause Time minus 28 Slot Times (PT -28 slot times) 010 - Pause Time minus 36 Slot Times (PT -36 slot times) 011 - Pause Time minus 144 Slot Times (PT -144 slot times) 100 - Pause Time minus 256 Slot Times (PT -256 slot times) 101 - Pause Time minus 512 Slot Times (PT -512 slot times) 110 - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable When this bit is set in full-duplex mode, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. 0 - Transmit Flow Control is disabled 1 - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy This bit initiates a PFC packet if the TFE bit is set. To initiate a PFC packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When PFC packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flow Control Busy or Backpressure Activate is disabled 1 - Flow Control Busy or Backpressure Activate is enabled

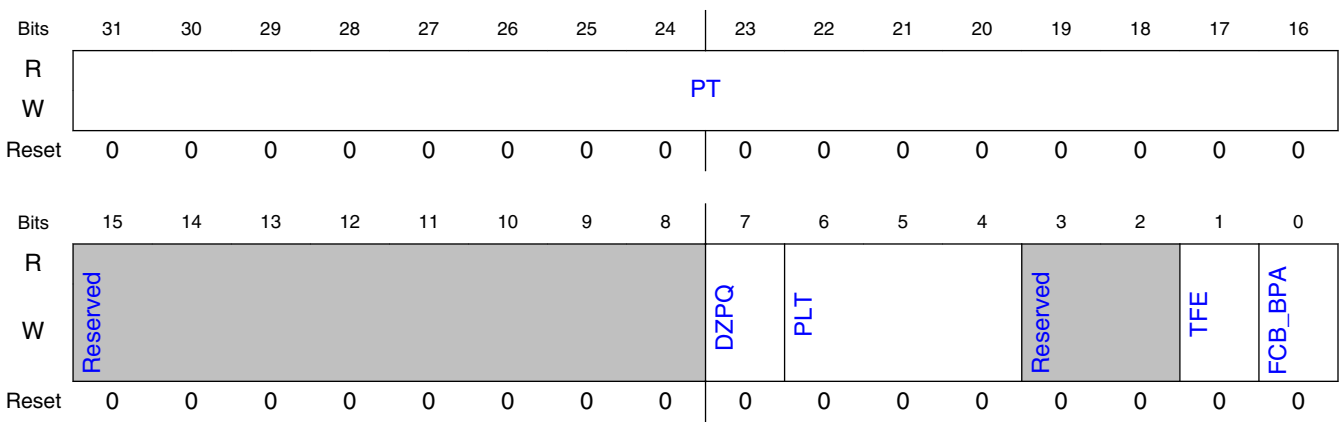
11.7.6.1.15 MAC Q2 Tx Flow Control (MAC_Q2_TX_FLOW_CTRL)

This register controls the generation of PFC Control packets of priorities mapped as per the PSRQi field in the MAC_Q2_RXQ_CTRL2/MAC_Q2_RXQ_CTRL3 registers.

11.7.6.1.15.1 Offset

Register	Offset
MAC_Q2_TX_FLOW_CTRL	78h

11.7.6.1.15.2 Diagram



11.7.6.1.15.3 Fields

Field	Description
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0 - Zero-Quanta Pause packet generation is enabled 1 - Zero-Quanta Pause packet generation is disabled

Table continues on the next page...

Field	Description
6-4 PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000 - Pause Time minus 4 Slot Times (PT -4 slot times) 001 - Pause Time minus 28 Slot Times (PT -28 slot times) 010 - Pause Time minus 36 Slot Times (PT -36 slot times) 011 - Pause Time minus 144 Slot Times (PT -144 slot times) 100 - Pause Time minus 256 Slot Times (PT -256 slot times) 101 - Pause Time minus 512 Slot Times (PT -512 slot times) 110 - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable When this bit is set in full-duplex mode, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. 0 - Transmit Flow Control is disabled 1 - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy This bit initiates a PFC packet if the TFE bit is set. To initiate a PFC packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When PFC packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flow Control Busy or Backpressure Activate is disabled 1 - Flow Control Busy or Backpressure Activate is enabled

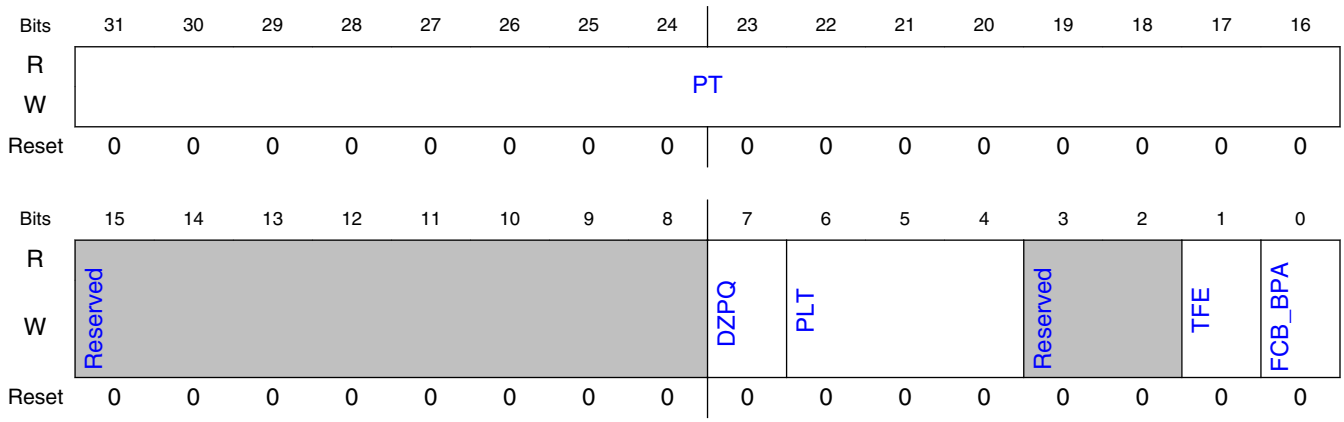
11.7.6.1.16 MAC Q3 Tx Flow Control (MAC_Q3_TX_FLOW_CTRL)

This register controls the generation of PFC Control packets of priorities mapped as per the PSRQi field in the MAC_Q3_RXQ_CTRL2/MAC_Q3_RXQ_CTRL3 registers.

11.7.6.1.16.1 Offset

Register	Offset
MAC_Q3_TX_FLOW_CTRL	7Ch

11.7.6.1.16.2 Diagram



11.7.6.1.16.3 Fields

Field	Description
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0 - Zero-Quanta Pause packet generation is enabled 1 - Zero-Quanta Pause packet generation is disabled
6-4 PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted.

Table continues on the next page...

Field	Description
	The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000 - Pause Time minus 4 Slot Times (PT -4 slot times) 001 - Pause Time minus 28 Slot Times (PT -28 slot times) 010 - Pause Time minus 36 Slot Times (PT -36 slot times) 011 - Pause Time minus 144 Slot Times (PT -144 slot times) 100 - Pause Time minus 256 Slot Times (PT -256 slot times) 101 - Pause Time minus 512 Slot Times (PT -512 slot times) 110 - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable When this bit is set in full-duplex mode, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. 0 - Transmit Flow Control is disabled 1 - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy This bit initiates a PFC packet if the TFE bit is set. To initiate a PFC packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When PFC packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flow Control Busy or Backpressure Activate is disabled 1 - Flow Control Busy or Backpressure Activate is enabled

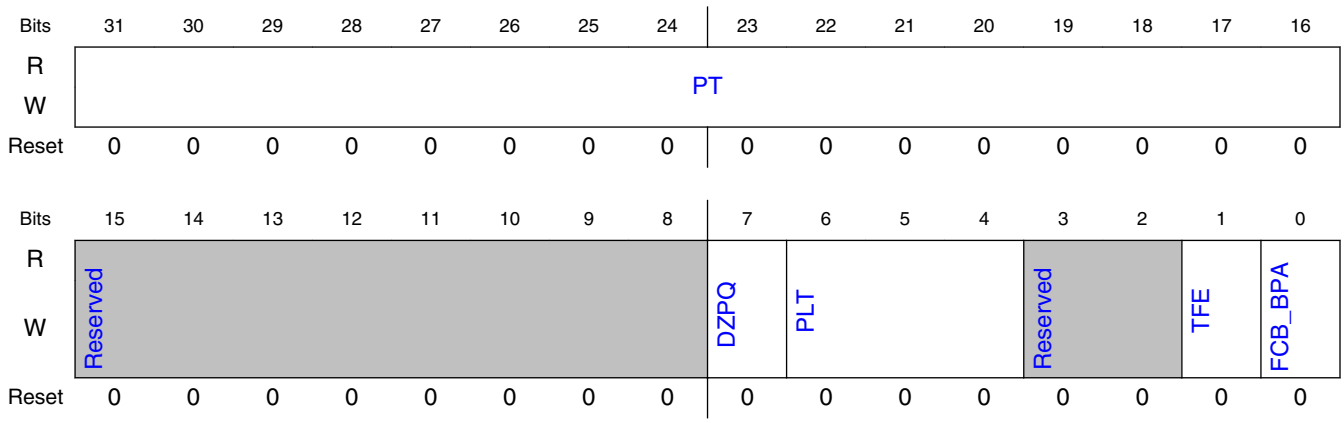
11.7.6.1.17 MAC Q4 Tx Flow Control (MAC_Q4_TX_FLOW_CTRL)

This register controls the generation of PFC Control packets of priorities mapped as per the PSRQi field in the MAC_Q4_RXQ_CTRL2/MAC_Q4_RXQ_CTRL3 registers.

11.7.6.1.17.1 Offset

Register	Offset
MAC_Q4_TX_FLOW_CTRL	80h

11.7.6.1.17.2 Diagram



11.7.6.1.17.3 Fields

Field	Description
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0 - Zero-Quanta Pause packet generation is enabled 1 - Zero-Quanta Pause packet generation is disabled
6-4 PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000 - Pause Time minus 4 Slot Times (PT -4 slot times) 001 - Pause Time minus 28 Slot Times (PT -28 slot times) 010 - Pause Time minus 36 Slot Times (PT -36 slot times)

Table continues on the next page...

Field	Description
	011 - Pause Time minus 144 Slot Times (PT -144 slot times) 100 - Pause Time minus 256 Slot Times (PT -256 slot times) 101 - Pause Time minus 512 Slot Times (PT -512 slot times) 110 - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable When this bit is set in full-duplex mode, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. 0 - Transmit Flow Control is disabled 1 - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy This bit initiates a PFC packet if the TFE bit is set. To initiate a PFC packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When PFC packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flow Control Busy or Backpressure Activate is disabled 1 - Flow Control Busy or Backpressure Activate is enabled

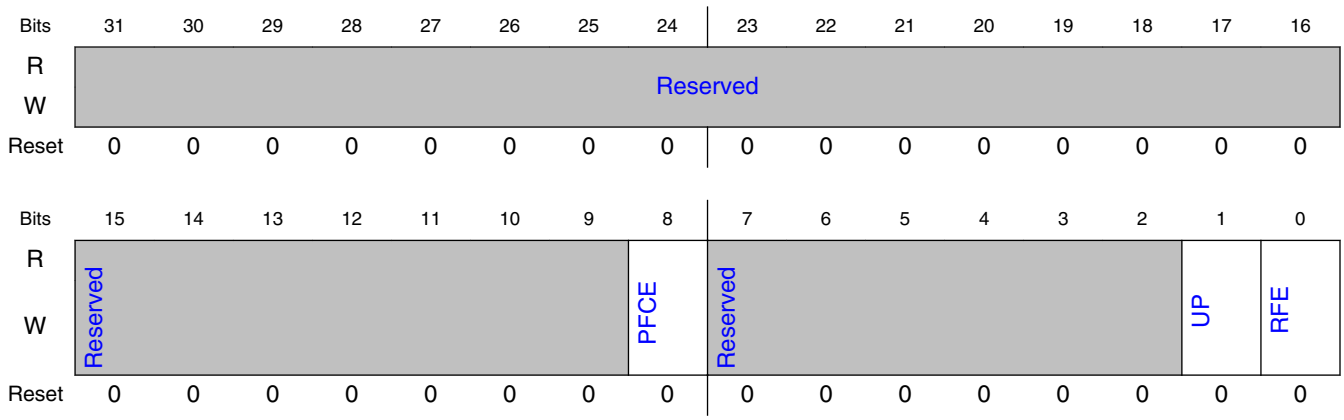
11.7.6.1.18 MAC Rx Flow Control (MAC_RX_FLOW_CTRL)

The Receive Flow Control register controls the pausing of MAC Transmit based on the received Pause packet.

11.7.6.1.18.1 Offset

Register	Offset
MAC_RX_FLOW_CTRL	90h

11.7.6.1.18.2 Diagram



11.7.6.1.18.3 Fields

Field	Description
31-9 —	Reserved
8 PFCE	Priority Based Flow Control Enable When this bit is set, it enables generation and reception of priority-based flow control (PFC) packets. When this bit is reset, it enables generation and reception of 802.3x Pause control packets. 0 - Priority Based Flow Control is disabled 1 - Priority Based Flow Control is enabled
7-2 —	Reserved
1 UP	Unicast Pause Packet Detect A pause packet is processed when it has the unique multicast address specified in the IEEE 802.3. When this bit is set, the MAC can also detect Pause packets with unicast address of the station. This unicast address should be as specified in MAC_ADDRESS0_HIGH and MAC_ADDRESS0_LOW. When this bit is reset, the MAC only detects Pause packets with unique multicast address. Note: The MAC does not process a Pause packet if the multicast address is different from the unique multicast address. This is also applicable to the received PFC packet when the Priority Flow Control (PFC) is enabled. The unique multicast address (0x01_80_C2_00_00_01) is as specified in IEEE 802.1 Qbb-2011. 0 - Unicast Pause Packet Detect disabled 1 - Unicast Pause Packet Detect enabled
0 RFE	Receive Flow Control Enable When this bit is set and the MAC is operating in full-duplex mode, the MAC decodes the received Pause packet and disables its transmitter for a specified (Pause) time. When this bit is reset or the MAC is operating in half-duplex mode, the decode function of the Pause packet is disabled. When PFC is enabled, flow control is enabled for PFC packets. The MAC decodes the received PFC packet and disables the Transmit queue, with matching priorities, for a duration of received Pause time. 0 - Receive Flow Control is disabled

Field	Description
	1 - Receive Flow Control is enabled

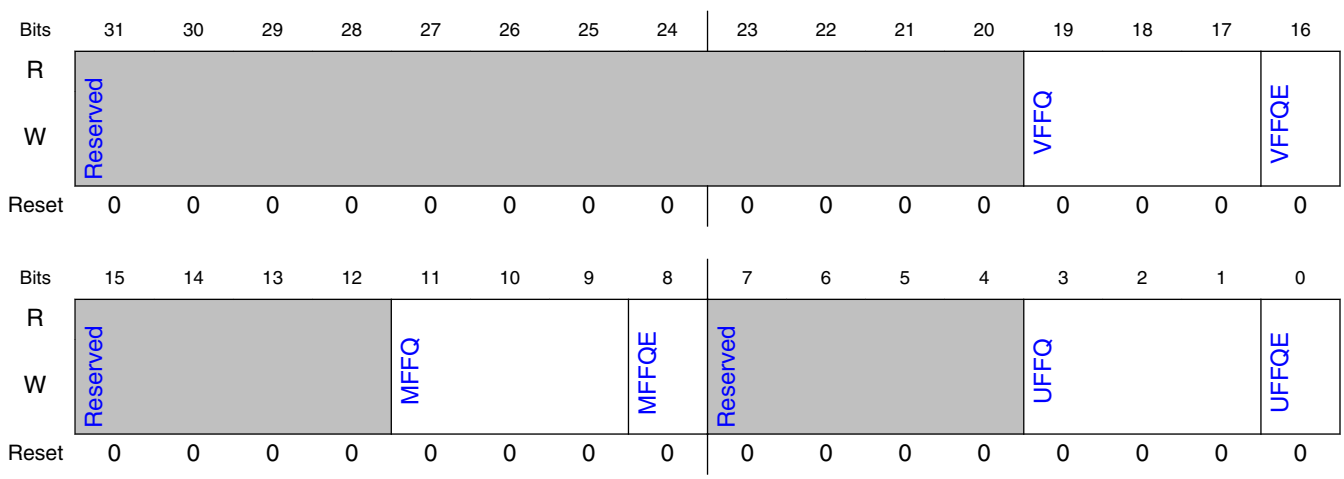
11.7.6.1.19 Receive Queue Control 4 (MAC_RXQ_CTRL4)

The Receive Queue Control 4 register controls the routing of unicast and multicast packets that fail the Destination or Source address filter to the Rx queues.

11.7.6.1.19.1 Offset

Register	Offset
MAC_RXQ_CTRL4	94h

11.7.6.1.19.2 Diagram



11.7.6.1.19.3 Fields

Field	Description
31-20 —	Reserved
19-17 VFFQ	VLAN Tag Filter Fail Packets Queue This field holds the Rx queue number to which the tagged packets failing the Destination or Source Address filter (and UFFQE/MFFQE not enabled) or failing the VLAN tag filter must be routed to. This field is valid only when the VFFQE bit is set.
16 VFFQE	VLAN Tag Filter Fail Packets Queuing Enable

Table continues on the next page...

Field	Description
	When this bit is set, the tagged packets which fail the Destination or Source address filter or fail the VLAN tag filter, are routed to the Rx Queue Number programmed in the VFFQ. When this bit is reset, the tagged packets which fail the Destination or Source address filter or fail the VLAN tag filter are routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set. 0 - VLAN tag Filter Fail Packets Queuing is disabled 1 - VLAN tag Filter Fail Packets Queuing is enabled
15-12 —	Reserved
11-9 MFFQ	Multicast Address Filter Fail Packets Queue. This field holds the Rx queue number to which the Multicast packets failing the Destination or Source Address filter are routed to. This field is valid only when the MFFQE bit is set.
8 MFFQE	Multicast Address Filter Fail Packets Queuing Enable. When this bit is set, the Multicast packets which fail the Destination or Source address filter is routed to the Rx Queue Number programmed in the MFFQ. When this bit is reset, the Multicast packets which fail the Destination or Source address filter is routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set. 0 - Multicast Address Filter Fail Packets Queuing is disabled 1 - Multicast Address Filter Fail Packets Queuing is enabled
7-4 —	Reserved
3-1 UFFQ	Unicast Address Filter Fail Packets Queue. Unicast Address Filter Fail Packets Queue. This field holds the Rx queue number to which the Unicast packets failing the Destination or Source Address filter are routed to. This field is valid only when the UFFQE bit is set.
0 UFFQE	Unicast Address Filter Fail Packets Queuing Enable. When this bit is set, the Unicast packets which fail the Destination or Source address filter is routed to the Rx Queue Number programmed in the UFFQ. When this bit is reset, the Unicast packets which fail the Destination or Source address filter is routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set. 0 - Unicast Address Filter Fail Packets Queuing is disabled 1 - Unicast Address Filter Fail Packets Queuing is enabled

11.7.6.1.20 Transmit Queue Priority Mapping 0 (MAC_TXQ_PRTY_MAP0)

The Transmit Queue Priority Mapping 0 register contains the priority values assigned to Tx Queue 0 through Tx Queue 3.

11.7.6.1.20.1 Offset

Register	Offset
MAC_TXQ_PRTY_MAP0	98h

11.7.6.1.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PSTQ3								PSTQ2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PSTQ1								PSTQ0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.20.3 Fields

Field	Description
31-24 PSTQ3	Priorities Selected in Transmit Queue 3 This bit is similar to the PSTQ0 bit.
23-16 PSTQ2	Priorities Selected in Transmit Queue 2 This bit is similar to the PSTQ0 bit.
15-8 PSTQ1	Priorities Selected in Transmit Queue 1 This bit is similar to the PSTQ0 bit.
7-0 PSTQ0	Priorities Selected in Transmit Queue 0 This field holds the priorities assigned to Tx Queue 0 by the software. This field determines if Tx Queue 0 should be blocked from transmitting specified pause time when a PFC packet is received with priorities matching the priorities programmed in this field. If the content of this field is not mutually exclusive to corresponding fields of other Transmit queues, that is, same priority is mapped to multiple Tx queues, the MAC blocks all queues with matching priority for specified time.

11.7.6.1.21 Transmit Queue Priority Mapping 1 (MAC_TXQ_PRTY_MAP1)

The Transmit Queue Priority Mapping 1 register contains the priority values assigned to Tx Queue 4.

11.7.6.1.21.1 Offset

Register	Offset
MAC_TXQ_PRTY_MAP1	9Ch

11.7.6.1.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PSTQ4							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.21.3 Fields

Field	Description
31-24 —	Reserved
23-16 —	Reserved
15-8 —	Reserved
7-0 PSTQ4	Priorities Selected in Transmit Queue 4 This field holds the priorities assigned to Tx Queue 4 by the software. This field determines if Tx Queue 4 should be blocked from transmitting specified pause time when a PFC packet is received with priorities matching the priorities programmed in this field. If the content of this field is not mutually exclusive to corresponding fields of other Transmit queues, that is, same priority is mapped to multiple Tx queues, the MAC blocks all queues with matching priority for specified time.

11.7.6.1.22 Receive Queue Control 0 (MAC_RXQ_CTRL0)

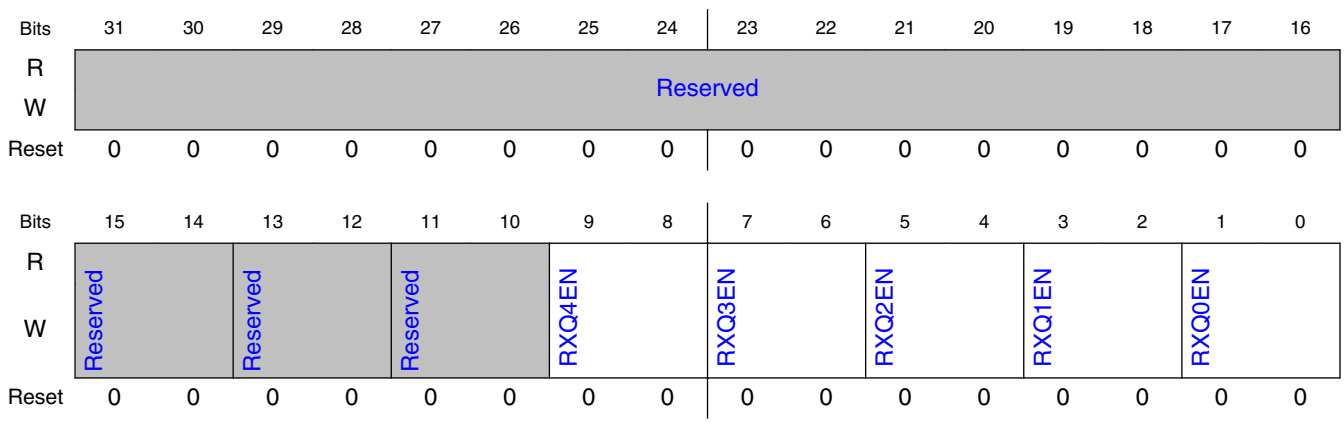
The Receive Queue Control 0 register controls the queue management in the MAC Receiver.

Note: In multiple Rx queues configuration, all the queues are disabled by default. Enable the Rx queue by programming the corresponding field in this register.

11.7.6.1.22.1 Offset

Register	Offset
MAC_RXQ_CTRL0	A0h

11.7.6.1.22.2 Diagram



11.7.6.1.22.3 Fields

Field	Description
31-16 —	Reserved
15-14 —	Reserved
13-12 —	Reserved
11-10 —	Reserved
9-8 RXQ4EN	Receive Queue 4 Enable This field is similar to the RXQ0EN field. 00 - Queue not enabled 01 - Queue enabled for AV 10 - Queue enabled for DCB/Generic 11 - Reserved
7-6	Receive Queue 3 Enable This field is similar to the RXQ0EN field.

Table continues on the next page...

Field	Description
RXQ3EN	00 - Queue not enabled 01 - Queue enabled for AV 10 - Queue enabled for DCB/Generic 11 - Reserved
5-4 RXQ2EN	Receive Queue 2 Enable This field is similar to the RXQ0EN field. 00 - Queue not enabled 01 - Queue enabled for AV 10 - Queue enabled for DCB/Generic 11 - Reserved
3-2 RXQ1EN	Receive Queue 1 Enable This field is similar to the RXQ0EN field. 00 - Queue not enabled 01 - Queue enabled for AV 10 - Queue enabled for DCB/Generic 11 - Reserved
1-0 RXQ0EN	Receive Queue 0 Enable This field indicates whether Rx Queue 0 is enabled for AV or DCB. 00 - Queue not enabled 01 - Queue enabled for AV 10 - Queue enabled for DCB/Generic 11 - Reserved

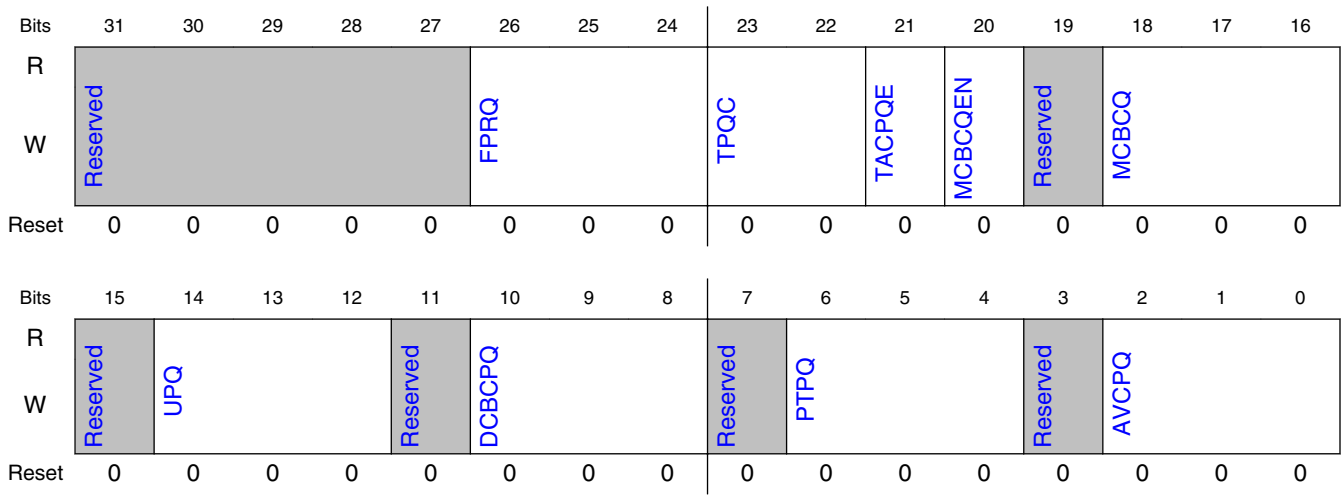
11.7.6.1.23 Receive Queue Control 1 (MAC_RXQ_CTRL1)

The Receive Queue Control 1 register controls the routing of multicast, broadcast, AV, DCB, and untagged packets to the Rx queues.

11.7.6.1.23.1 Offset

Register	Offset
MAC_RXQ_CTRL1	A4h

11.7.6.1.23.2 Diagram



11.7.6.1.23.3 Fields

Field	Description
31-27 —	Reserved
26-24 FPRQ	Frame Preemption Residue Queue This field holds the Rx queue number to which the residual preemption frames must be forwarded. Preemption frames that are tagged and pass the SA/DA/VLAN filtering are routed based on PSRQ and all other frames are treated as residual frames and is routed to the queue number mentioned in this field. The Queue-0 is used as a default queue for express frames, so this field cannot be programmed to a value 0.
23-22 TPQC	Tagged PTP over Ethernet Packets Queuing Control. This field controls the routing of the VLAN Tagged PTPoE packets. If DWC_EQOS_AV_ENABLE is selected in the configuration, the following programmable options are allowed. - 2'b00: VLAN Tagged PTPoE packets are routed as generic VLAN Tagged packet (based on PSRQ for only non-AV enabled Rx Queues). - 2'b01: VLAN Tagged PTPoE packets are routed to Rx Queue specified by PTPQ field (That Rx Queue can be enabled for AV or non-AV traffic). - 2'b10: VLAN Tagged PTPoE packets are routed to only AV enabled Rx Queues based on PSRQ. - 2'b11: Reserved If DWC_EQOS_AV_ENABLE is not selected in the configuration, the following programmable options are allowed. - 1'b0: VLAN Tagged PTPoE packets are routed as generic VLAN Tagged packet (based on PSRQ for DCB/Generic enabled Rx Queues). - 1'b1: VLAN Tagged PTPoE packets are routed to Rx Queues specified by PTPQ field.
21 TACPQE	Tagged AV Control Packets Queuing Enable.

Table continues on the next page...

Field	Description
	When set, the MAC routes the received Tagged AV Control packets to the Rx queue specified by AVCPQ field. When reset, the MAC routes the received Tagged AV Control packets based on the tag priority matching the PSRQ fields in MAC_RXQ_CTRL2 and MAC_RXQ_CTRL3 registers. 0 - Tagged AV Control Packets Queuing is disabled 1 - Tagged AV Control Packets Queuing is enabled
20 MCBCQEN	Multicast and Broadcast Queue Enable This bit specifies that Multicast or Broadcast packets routing to the Rx Queue is enabled and the Multicast or Broadcast packets must be routed to Rx Queue specified in MCBCQ field. 0 - Multicast and Broadcast Queue is disabled 1 - Multicast and Broadcast Queue is enabled
19 —	Reserved
18-16 MCBCQ	Multicast and Broadcast Queue This field specifies the Rx Queue onto which Multicast or Broadcast Packets are routed. Any Rx Queue enabled for Generic/DCB/AV traffic can be used to route the Multicast or Broadcast Packets. 000 - Receive Queue 0 001 - Receive Queue 1 010 - Receive Queue 2 011 - Receive Queue 3 100 - Receive Queue 4 101 - Reserved 110 - Reserved 111 - Reserved
15 —	Reserved
14-12 UPQ	Untagged Packet Queue This field indicates the Rx Queue to which Untagged Packets are to be routed. Any Rx Queue enabled for Generic/DCB/AV traffic can be used to route the Untagged Packets. 000 - Receive Queue 0 001 - Receive Queue 1 010 - Receive Queue 2 011 - Receive Queue 3 100 - Receive Queue 4 101 - Reserved 110 - Reserved 111 - Reserved
11 —	Reserved
10-8 DCBCPQ	DCB Control Packets Queue This field specifies the Rx queue on which the received DCB control packets are routed.

Table continues on the next page...

Field	Description
	The DCB data packets are routed based on the PSRQ field of the Transmit Flow Control Register of corresponding queue. 000 - Receive Queue 0 001 - Receive Queue 1 010 - Receive Queue 2 011 - Receive Queue 3 100 - Receive Queue 4 101 - Reserved 110 - Reserved 111 - Reserved
7 —	Reserved
6-4 PTPQ	PTP Packets Queue This field specifies the Rx queue on which the PTP packets sent over the Ethernet payload (not over IPv4 or IPv6) are routed. When the AV8021ASMEN bit of MAC_TIMESTAMP_CONTROL register is set, only untagged PTP over Ethernet packets are routed on an Rx Queue. If the bit is not set, then based on programming of TPQC field, both tagged and untagged PTPoE packets can be routed to this Rx Queue. 000 - Receive Queue 0 001 - Receive Queue 1 010 - Receive Queue 2 011 - Receive Queue 3 100 - Receive Queue 4 101 - Reserved 110 - Reserved 111 - Reserved
3 —	Reserved
2-0 AVCPQ	AV Untagged Control Packets Queue This field specifies the Receive queue on which the received AV tagged and untagged control packets are routed. The AV tagged (when TACPQE bit is set) and untagged control packets are routed to Receive queue specified by this field. 000 - Receive Queue 0 001 - Receive Queue 1 010 - Receive Queue 2 011 - Receive Queue 3 100 - Receive Queue 4 101 - Reserved 110 - Reserved 111 - Reserved

11.7.6.1.24 Receive Queue Control 2 (MAC_RXQ_CTRL2)

This register controls the routing of tagged packets based on the USP (user Priority) field of the received packets to the RxQueues 0 to 3.

11.7.6.1.24.1 Offset

Register	Offset
MAC_RXQ_CTRL2	A8h

11.7.6.1.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PSRQ3								PSRQ2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PSRQ1								PSRQ0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.24.3 Fields

Field	Description
31-24 PSRQ3	Priorities Selected in the Receive Queue 3 This field decides the priorities assigned to Rx Queue 3. All packets with priorities that match the values set in this field are routed to Rx Queue 3. For example, if PSRQ3[6, 3] are set, packets with USP field equal to 3 or 6 are routed to Rx Queue 3. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues. This field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 3 crosses the flow control threshold settings.
23-16 PSRQ2	Priorities Selected in the Receive Queue 2 This field decides the priorities assigned to Rx Queue 2. All packets with priorities that match the values set in this field are routed to Rx Queue 2. For example, if PSRQ2[1, 0] are set, packets with USP field equal to 1 or 0 are routed to Rx Queue 2. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues.

Table continues on the next page...

Field	Description
	this field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 2 crosses the flow control threshold settings.
15-8 PSRQ1	Priorities Selected in the Receive Queue 1 This field decides the priorities assigned to Rx Queue 1. All packets with priorities that match the values set in this field are routed to Rx Queue 1. For example, if PSRQ1[4] is set, packets with USP field equal to 4 are routed to Rx Queue 1. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues. this field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 1 crosses the flow control threshold settings.
7-0 PSRQ0	Priorities Selected in the Receive Queue 0 This field decides the priorities assigned to Rx Queue 0. All packets with priorities that match the values set in this field are routed to Rx Queue 0. For example, if PSRQ0[5] is set, packets with USP field equal to 5 are routed to Rx Queue 0. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues. this field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 0 crosses the flow control threshold settings.

11.7.6.1.25 Receive Queue Control 3 (MAC_RXQ_CTRL3)

This register controls the routing of tagged packets based on the USP (user Priority) field of the received packets to Rx Queue 4.

11.7.6.1.25.1 Offset

Register	Offset
MAC_RXQ_CTRL3	ACh

11.7.6.1.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PSRQ4							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.25.3 Fields

Field	Description
31-24 —	Reserved
23-16 —	Reserved
15-8 —	Reserved
7-0 PSRQ4	Priorities Selected in the Receive Queue 4 This field decides the priorities assigned to Rx Queue 4. All packets with priorities that match the values set in this field are routed to Rx Queue 4. For example, if PSRQ4[7:4] is set, packets with USP field equal to 7, 6, 5, or 4 are routed to Rx Queue 4. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues. This field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 4 crosses the flow control threshold settings.

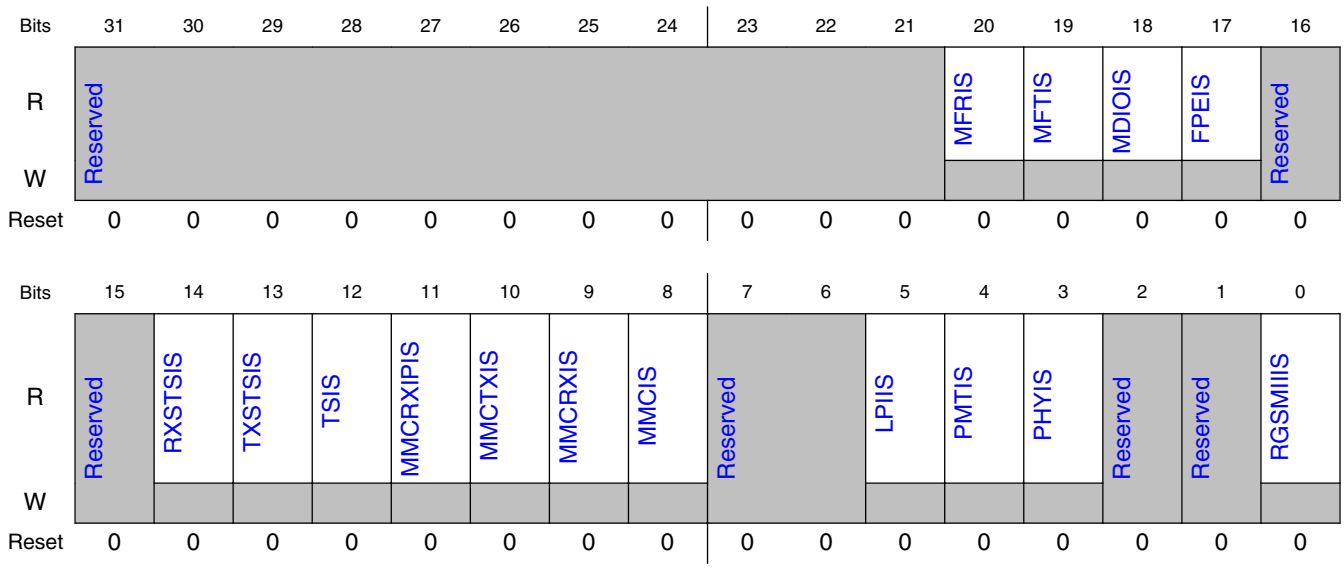
11.7.6.1.26 Interrupt Status (MAC_INTERRUPT_STATUS)

The Interrupt Status register contains the status of interrupts.

11.7.6.1.26.1 Offset

Register	Offset
MAC_INTERRUPT_STATUS	B0h

11.7.6.1.26.2 Diagram



11.7.6.1.26.3 Fields

Field	Description
31-21 —	Reserved
20 MFRIS	MMC FPE Receive Interrupt Status This bit is set high when an interrupt is generated in the MMC FPE Receive Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option along with FPE support. 0 - MMC FPE Receive Interrupt status not active 1 - MMC FPE Receive Interrupt status active
19 MFTIS	MMC FPE Transmit Interrupt Status This bit is set high when an interrupt is generated in the MMC FPE Transmit Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option along with FPE support. 0 - MMC FPE Transmit Interrupt status not active 1 - MMC FPE Transmit Interrupt status active
18 MDIOIS	MDIO Interrupt Status This bit indicates an interrupt event after the completion of MDIO operation. To reset this bit, the application has to read this bit/Write 1 to this bit when RCWE bit of CSR_SW_CTRL register is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - MDIO Interrupt status not active 1 - MDIO Interrupt status active
17	Frame Preemption Interrupt Status

Table continues on the next page...

Field	Description
FPEIS	This bit indicates an interrupt event during the operation of Frame Preemption (Bits[19:16] of MAC_FPE_CTRL_STS register is set). To reset this bit, the application must clear the event in MAC_FPE_CTRL_STS that has caused the Interrupt. 0 - Frame Preemption Interrupt status not active 1 - Frame Preemption Interrupt status active
16 —	Reserved
15 —	Reserved
14 RXSTSIS	Receive Status Interrupt This bit indicates the status of received packets. This bit is set when the RWT bit is set in the MAC_RX_TX_STATUS register. This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of CSR_SW_CTRL register is set) in the MAC_RX_TX_STATUS register. 0 - Receive Interrupt status not active 1 - Receive Interrupt status active
13 TXSTSIS	Transmit Status Interrupt This bit indicates the status of transmitted packets. This bit is set when any of the following bits is set in the MAC_RX_TX_STATUS register: - Excessive Collision (EXCOL) - Late Collision (LCOL) - Excessive Deferral (EXDEF) - Loss of Carrier (LCARR) - No Carrier (NCARR) - Jabber Timeout (TJT) This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of CSR_SW_CTRL register is set) in the MAC_RX_TX_STATUS register. 0 - Transmit Interrupt status not active 1 - Transmit Interrupt status active
12 TSIS	Timestamp Interrupt Status If the Timestamp feature is enabled, this bit is set when any of the following conditions is true: - The system time value is equal to or exceeds the value specified in the Target Time High and Low registers. - There is an overflow in the Seconds register. - The Target Time Error occurred, that is, programmed target time already elapsed. If the Auxiliary Snapshot feature is enabled, this bit is set when the auxiliary snapshot trigger is asserted. In configurations other than EQOS_CORE, when drop transmit status is enabled in MTL, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and MAC_TXTIMESTAMP_STATUS_SECONDS registers.

Table continues on the next page...

Field	Description
	When PTP offload feature is enabled, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and MAC_TX_TIMESTAMP_STATUS_SECONDS registers, for PTO generated Delay Request and Pdelay request packets. This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of CSR_SW_CTRL register is set) in the MAC_TIMESTAMP_STATUS register. 0 - Timestamp Interrupt status not active 1 - Timestamp Interrupt status active
11 MMCRXIPIS	MMC Receive Checksum Offload Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) and Enable Receive TCP/IP Checksum Check options. 0 - MMC Receive Checksum Offload Interrupt status not active 1 - MMC Receive Checksum Offload Interrupt status active
10 MMCTXIS	MMC Transmit Interrupt Status This bit is set high when an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. 0 - MMC Transmit Interrupt status not active 1 - MMC Transmit Interrupt status active
9 MMCRXIS	MMC Receive Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. 0 - MMC Receive Interrupt status not active 1 - MMC Receive Interrupt status active
8 MMCIS	MMC Interrupt Status This bit is set high when Bit 11, Bit 10, or Bit 9 is set high. This bit is cleared only when all these bits are low. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. 0 - MMC Interrupt status not active 1 - MMC Interrupt status active
7-6 —	Reserved
5 LPIIS	LPI Interrupt Status When the Energy Efficient Ethernet feature is enabled, this bit is set for any LPI state entry or exit in the MAC Transmitter or Receiver. This bit is cleared when the corresponding interrupt source bit of MAC_LPI_CONTROL_STATUS register is read (or corresponding interrupt source bit of MAC_LPI_CONTROL_STATUS register is written to 1 when RCWE bit of MAC_CSR_SW_CTRL register is set). 0 - LPI Interrupt status not active 1 - LPI Interrupt status active
4	PMT Interrupt Status

Table continues on the next page...

Field	Description
PMTIS	This bit is set when a Magic packet or Wake-on-LAN packet is received in the power-down mode (RWKPRCVD and MGKPRCVD bits in MAC_PMT_CONTROL_STATUS register). This bit is cleared when corresponding interrupt source bit are cleared because of a Read operation to the MAC_PMT_CONTROL_STATUS register (or corresponding interrupt source bit of MAC_PMT_CONTROL_STATUS register is written to 1 when RCWE bit of CSR_SW_CTRL register is set). This bit is valid only when you select the Enable Power Management option. 0 - PMT Interrupt status not active 1 - PMT Interrupt status active
3 PHYIS	PHY Interrupt This bit is set when rising edge is detected on the phy_intr_i input. This bit is cleared when this register is read (or this bit is written to 1 when RCWE bit of CSR_SW_CTRL register is set). 0 - PHY Interrupt not detected 1 - PHY Interrupt detected
2 —	Reserved
1 —	Reserved
0 RGSMIIIS	RGMII or SMII Interrupt Status This bit is set because of any change in value of the Link Status of RGMII or SMII interface (LNKSTS bit in MAC_PHYIF_CONTROL_STATUS register). This bit is cleared when the PHYIF_CONTROL_STATUS register is read (or LNKSTS bit of MAC_PHYIF_CONTROL_STATUS register is written to 1 when RCWE bit of MAC_CSR_SW_CTRL register is set). This bit is valid only when you select the optional RGMII or SMII PHY interface. 0 - RGMII or SMII Interrupt Status is not active 1 - RGMII or SMII Interrupt Status is active

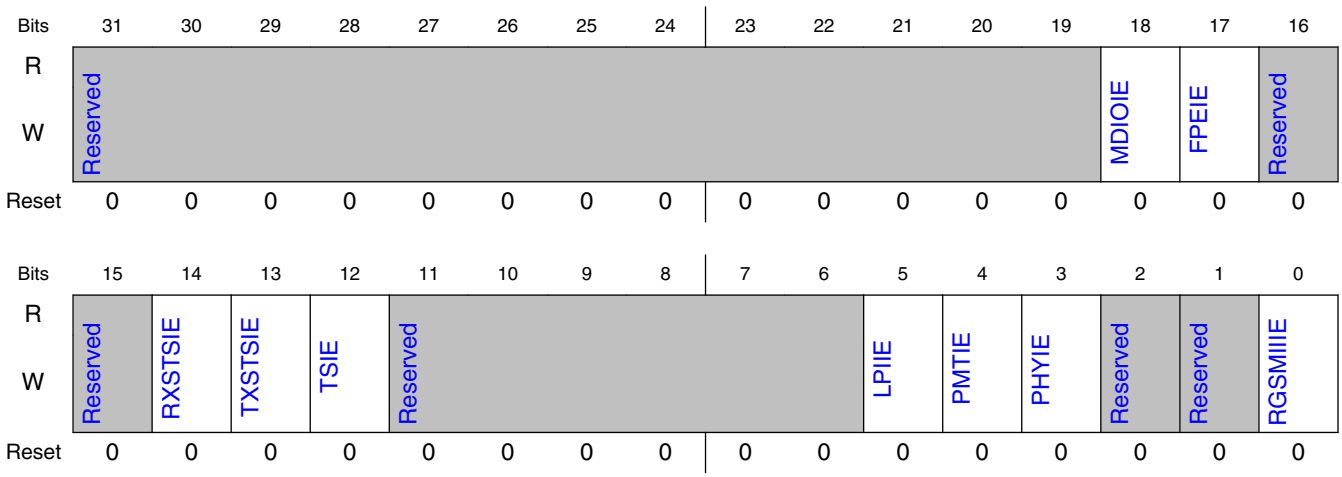
11.7.6.1.27 Interrupt Enable (MAC_INTERRUPT_ENABLE)

The Interrupt Enable register contains the masks for generating the interrupts.

11.7.6.1.27.1 Offset

Register	Offset
MAC_INTERRUPT_ENABLE	B4h

11.7.6.1.27.2 Diagram



11.7.6.1.27.3 Fields

Field	Description
31-19 —	Reserved
18 MDIOIE	MDIO Interrupt Enable When this bit is set, it enables the assertion of the interrupt when MDIOIS field is set in the MAC_INTERRUPT_STATUS register. 0 - MDIO Interrupt is disabled 1 - MDIO Interrupt is enabled
17 FPEIE	Frame Preemption Interrupt Enable When this bit is set, it enables the assertion of the interrupt when FPEIS field is set in the MAC_INTERRUPT_STATUS. 0 - Frame Preemption Interrupt is disabled 1 - Frame Preemption Interrupt is enabled
16 —	Reserved
15 —	Reserved
14 RXSTSIE	Receive Status Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[RXSTSIS]. 0 - Receive Status Interrupt is disabled 1 - Receive Status Interrupt is enabled
13 TXSTSIE	Transmit Status Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[TXSTSIS]. 0 - Timestamp Status Interrupt is disabled 1 - Timestamp Status Interrupt is enabled
12 TSIE	Timestamp Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[TSIS].

Table continues on the next page...

Field	Description
	0 - Timestamp Interrupt is disabled 1 - Timestamp Interrupt is enabled
11-6 —	Reserved
5 LPIIE	LPI Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[LPIIS]. 0 - LPI Interrupt is disabled 1 - LPI Interrupt is enabled
4 PMTIE	PMT Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[PMTIS]. 0 - PMT Interrupt is disabled 1 - PMT Interrupt is enabled
3 PHYIE	PHY Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[PHYIS]. 0 - PHY Interrupt is disabled 1 - PHY Interrupt is enabled
2 —	Reserved
1 —	Reserved
0 RGSMIIIE	RGMII or SMII Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of RGSMIIIS bit in MAC_INTERRUPT_STATUS register. 0 - RGMII or SMII Interrupt is disabled 1 - RGMII or SMII Interrupt is enabled

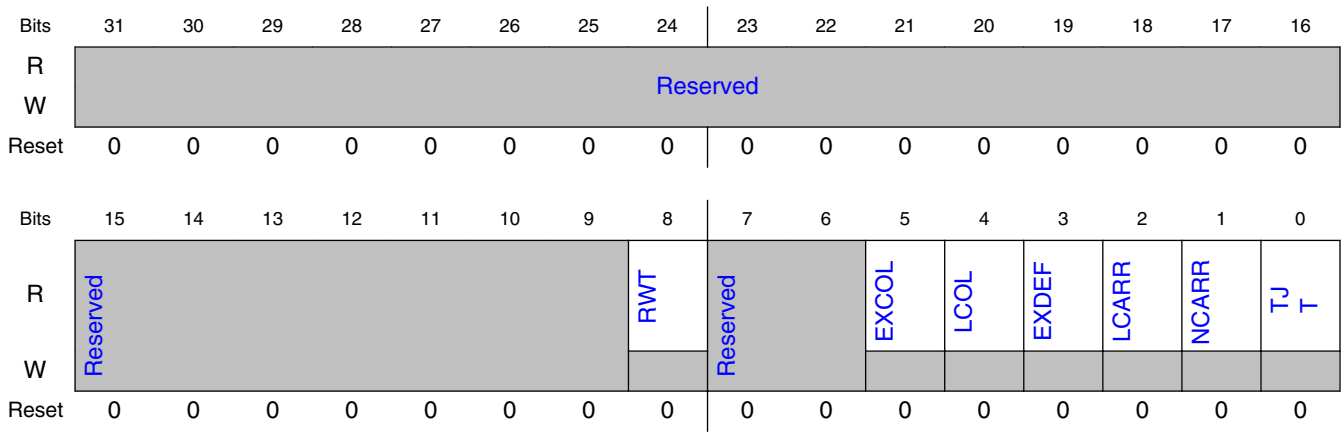
11.7.6.1.28 Receive Transmit Status (MAC_RX_TX_STATUS)

The Receive Transmit Status register contains the Receive and Transmit Error status.

11.7.6.1.28.1 Offset

Register	Offset
MAC_RX_TX_STATUS	B8h

11.7.6.1.28.2 Diagram



11.7.6.1.28.3 Fields

Field	Description
31-9 —	Reserved
8 RWT	Receive Watchdog Timeout This bit is set when a packet with length greater than 2,048 bytes is received (10, 240 bytes when Jumbo Packet mode is enabled) and the WD bit is reset in the MAC_CONFIGURATION register. Receive Watchdog Timeout This bit is set when a packet with length greater than 2,048 bytes is received (10, 240 bytes when Jumbo Packet mode is enabled) and the WD bit is reset in the MAC_CONFIGURATION register. This bit is set when a packet with length greater than 16,383 bytes is received and the WD bit is set in the MAC_CONFIGURATION register. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - No receive watchdog timeout 1 - Receive watchdog timed out
7-6 —	Reserved
5 EXCOL	Excessive Collisions When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the transmission aborted after 16 successive collisions while attempting to transmit the current packet. Excessive Collisions When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the transmission aborted after 16 successive collisions while attempting to transmit the current packet. If the DR bit is set in the CONFIGURATION register, this bit is set after the first collision and the packet transmission is aborted. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - No collision

Table continues on the next page...

Field	Description
	1 - Excessive collision is sensed
4 LCOL	Late Collision When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the packet transmission aborted because a collision occurred after the collision window (512 bytes including Preamble and Carrier Extension in GMII mode). Late Collision When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the packet transmission aborted because a collision occurred after the collision window (512 bytes including Preamble and Carrier Extension in GMII mode). This bit is not valid if the Underflow error occurs. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - No collision 1 - Late collision is sensed
3 EXDEF	Excessive Deferral When the DTXSTS bit is set in the MAC_OPERATION_MODE register and the DC bit is set in the MAC_CONFIGURATION register, this bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 in 1000/2500 Mbps mode or when Jumbo packet is enabled). Excessive Deferral When the DTXSTS bit is set in the MAC_OPERATION_MODE register and the DC bit is set in the MAC_CONFIGURATION register, this bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 in 1000/2500 Mbps mode or when Jumbo packet is enabled). Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - No Excessive deferral 1 - Excessive deferral
2 LCARR	Loss of Carrier When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the loss of carrier occurred during packet transmission, that is, the phy_crs_i signal was inactive for one or more transmission clock periods during packet transmission. Loss of Carrier When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the loss of carrier occurred during packet transmission, that is, the phy_crs_i signal was inactive for one or more transmission clock periods during packet transmission. This bit is valid only for packets transmitted without collision. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Carrier is present 1 - Loss of carrier
1 NCARR	No Carrier When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the carrier signal from the PHY is not present at the end of preamble transmission. No Carrier When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the carrier signal from the PHY is not present at the end of preamble transmission. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Carrier is present

Table continues on the next page...

Field	Description
	1 - No carrier
0 TJT	Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired which happens when the packet size exceeds 2,048 bytes (10,240 bytes when the Jumbo packet is enabled) and JD bit is reset in the MAC_CONFIGURATION register. Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired which happens when the packet size exceeds 2,048 bytes (10,240 bytes when the Jumbo packet is enabled) and JD bit is reset in the MAC_CONFIGURATION register. This bit is set when the packet size exceeds 16,383 bytes and the JD bit is set in the MAC_CONFIGURATION register. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - No Transmit Jabber Timeout 1 - Transmit Jabber Timeout occurred

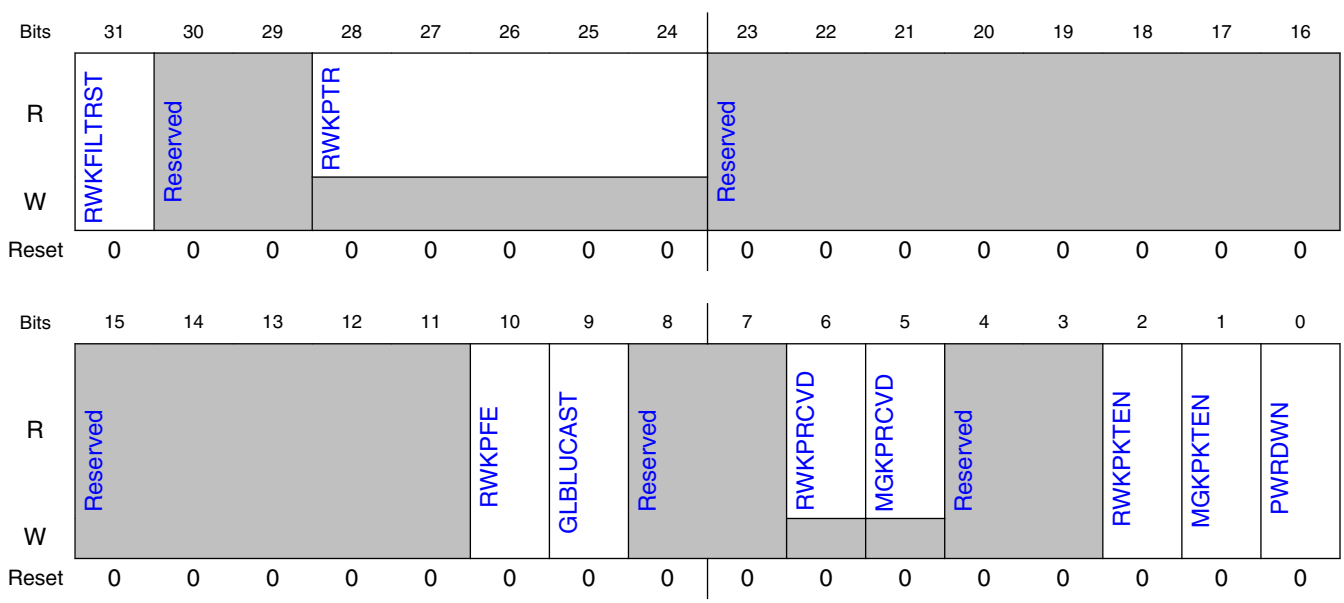
11.7.6.1.29 PMT Control and Status (MAC_PMT_CONTROL_STATUS)

The PMT Control and Status Register.

11.7.6.1.29.1 Offset

Register	Offset
MAC_PMT_CONTROL_STATUS	C0h

11.7.6.1.29.2 Diagram



11.7.6.1.29.3 Fields

Field	Description
31 RWKFILTRST	Remote Wake-Up Packet Filter Register Pointer Reset When this bit is set, the remote wake-up packet filter register pointer is reset to 3'b000. Remote Wake-Up Packet Filter Register Pointer Reset When this bit is set, the remote wake-up packet filter register pointer is reset to 3'b000. It is automatically cleared after 1 clock cycle. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Remote Wake-Up Packet Filter Register Pointer is not Reset 1 - Remote Wake-Up Packet Filter Register Pointer is Reset
30-29 —	Reserved
28-24 RWKPTR	Remote Wake-up FIFO Pointer This field gives the current value (0 to 7, 15, or 31 when 4, 8, or 16 Remote Wake-up Packet Filters are selected) of the Remote Wake-up Packet Filter register pointer. Remote Wake-up FIFO Pointer This field gives the current value (0 to 7, 15, or 31 when 4, 8, or 16 Remote Wake-up Packet Filters are selected) of the Remote Wake-up Packet Filter register pointer. When the value of this pointer is equal to maximum for the selected number of Remote Wake-up Packet Filters, the contents of the Remote Wake-up Packet Filter Register are transferred to the clk_rx_i domain when a Write occurs to that register.
23-11 —	Reserved
10 RWKPFE	Remote Wake-up Packet Forwarding Enable When this bit is set along with RWKPKTEN, the MAC receiver drops all received frames until it receives the expected Wake-up frame. Remote Wake-up Packet Forwarding Enable When this bit is set along with RWKPKTEN, the MAC receiver drops all received frames until it receives the expected Wake-up frame. All frames after that event including the received wake-up frame are forwarded to application. This bit is then self-cleared on receiving the wake-up packet. The application can also clear this bit before the expected wake-up frame is received. In such cases, the MAC reverts to the default behavior where packets received are forwarded to the application. This bit must only be set when RWKPKTEN is set high and PWRDWN is set low. The setting of this bit has no effect when PWRDWN is set high. Note: If Magic Packet Enable and Wake-Up Frame Enable are both set along with setting of this bit and Magic Packet is received prior to wake-up frame, this bit is self-cleared on receiving Magic Packet, the received Magic packet is dropped, and all frames after received Magic Packet are forwarded to application. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Remote Wake-up Packet Forwarding is disabled 1 - Remote Wake-up Packet Forwarding is enabled
9 GLBLUCAST	Global Unicast When this bit set, any unicast packet filtered by the MAC (DAF) address recognition is detected as a remote wake-up packet. 0 - Global unicast is disabled 1 - Global unicast is enabled

Table continues on the next page...

Field	Description
8-7 —	Reserved
6 RWKPRCVD	Remote Wake-Up Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a remote wake-up packet. Remote Wake-Up Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a remote wake-up packet. This bit is cleared when this register is read. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Remote wake-up packet is received 1 - Remote wake-up packet is received
5 MGKPRCVD	Magic Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a magic packet. Magic Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a magic packet. This bit is cleared when this register is read. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - No Magic packet is received 1 - Magic packet is received
4-3 —	Reserved
2 RWKPKTEN	Remote Wake-Up Packet Enable When this bit is set, a power management event is generated when the MAC receives a remote wake-up packet. 0 - Remote wake-up packet is disabled 1 - Remote wake-up packet is enabled
1 MGKPKTEN	Magic Packet Enable When this bit is set, a power management event is generated when the MAC receives a magic packet. 0 - Magic Packet is disabled 1 - Magic Packet is enabled
0 PWRDWN	Power Down When this bit is set, the MAC receiver drops all received packets until it receives the expected magic packet or remote wake-up packet. Power Down When this bit is set, the MAC receiver drops all received packets until it receives the expected magic packet or remote wake-up packet. This bit is then self-cleared and the power-down mode is disabled. The software can clear this bit before the expected magic packet or remote wake-up packet is received. The packets received by the MAC after this bit is cleared are forwarded to the application. This bit must only be set when the Magic Packet Enable, Global Unicast, or Remote Wake-Up Packet Enable bit is set high. Note: You can gate-off the CSR clock during the power-down mode. However, when the CSR clock is gated-off, you cannot perform any read or write operations on this register. Therefore, the Software cannot clear this bit. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Power down is disabled 1 - Power down is enabled

11.7.6.1.30 Remote Wakeup Filter (MAC_RWK_PACKET_FILTER)

The Remote Wakeup Filter registers are implemented as 8, 16, or 32 indirect access registers (wkuppktfilter_reg#i) based on whether 4, 8, or 16 Remote Wakeup Filters are selected in the configuration and accessed by application through MAC_RWK_PACKET_FILTER register.

The Remote Wakeup Filter registers are implemented as 8, 16, or 32 indirect access registers (wkuppktfilter_reg#i) based on whether 4, 8, or 16 Remote Wakeup Filters are selected in the configuration and accessed by application through MAC_RWK_PACKET_FILTER register. When the Remote Wakeup Filters are to be programmed, the entire set of wkuppktfilter_reg registers must be written. The wkuppktfilter_reg register is programmed by sequentially writing the eight, sixteen or thirty-two register values in MAC_RWK_PACKET_FILTER register for wkuppktfilter_REG0, wkuppktfilter_REG1, ..., wkuppktfilter_REG31 respectively. The wkuppktfilter_reg register is read in a similar way. The MAC updates the wkuppktfilter_reg register current pointer value in RWKPTR field of MAC_PMT_CONTROL_STATUS register.

The Remote Wakeup Filters are arranged in blocks of 4 filters each and each such block have eight 32-bit wide registers, viz. wkuppktfilter_REG0-7, wkuppktfilter_reg8-15, wkuppktfilter_REG16-23 and wkuppktfilter_REG24-31. The fields of Remote Wakeup Filter are described as follows:

Filter i Byte Mask: The filter i byte mask register defines the bytes of the packet that are examined by filter i (0, 1, 2, 3, .., 15) to determine whether or not a packet is a wake-up packet.

- The MSB (31st bit) must be zero.

- Bit j[30:0] is the byte mask.

- If Bit j (byte number) of the byte mask is set, the CRC block processes the Filter i Offset + j of the incoming packet; otherwise Filter i Offset + j is ignored.

Filter i Command: The 4-bit filter i command controls the filter i operation.

- Bit 3 specifies the address type, defining the destination address type of the pattern. When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet.

- Bit 2 (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC-16 value. Bit 2, along with Bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as "Pattern 1 AND NOT Pattern 2".

- Bit 1 (And_Previous) implements the Boolean logic. When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set. The details are provided below:

-- The And_Previous bit setting is applicable within a set of 4 filters.

-- Setting of And_Previous bit of filter that is not enabled has no effect, that is setting And_Previous bit of lowest number filter in the set of 4 filters has no effect. For example, setting of And_Previous bit of Filter 0 has no effect.

-- If And_Previous bit is set for filter to form AND chained filter, the AND chain breaks at the point any filter is not enabled. For example: If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set) but Filter 1 is not enabled (bit 0 in Filter 1 command is reset), then only Filter 2 result is considered. If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 3 And_Previous bit is set (bit 1 in Filter 3 command is set), but Filter 1 is not enabled (bit 0 in Filter 1 command is reset), then only Filter 2 result ANDed with Filter 3 result is considered. If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 3 And_Previous bit is set (bit 1 in Filter 3 command is set), but Filter 2 is not enabled (bit 0 in Filter 2 command is reset), then since setting of Filter 2 And_Previous bit has no effect only Filter 1 result ORed with Filter 3 result is considered.

-- If filters chained by And_Previous bit setting have complementary programming, then a frame may never pass the AND chained filter. For example, if Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 1 Address_Type bit is set (bit 3 in Filter 1 command is set) indicating multicast detection and Filter 2 Address_Type bit is reset (bit 3 in Filter 2 command is reset) indicating unicast detection or vice versa, a remote wakeup frame does not pass the AND chained filter as a remote wakeup frame cannot be of both unicast and multicast address type.

- Bit 0 is the enable for filter i. If Bit 0 is not set, filter i is disabled.

Filter i Offset: The filter i offset register defines the offset (within the packet) from which the filter i examines the packets.

- This 8-bit pattern-offset is the offset for the filter i first byte to be examined.

- The minimum allowed offset is 12, which refers to the 13th byte of the packet.

- The offset value 0 refers to the first byte of the packet.

Filter i CRC-16: The filter i CRC-16 register contains the CRC-16 value calculated from the pattern and the byte mask programmed in the Remote Wakeup filter register.

- The 16-bit CRC calculation uses the following polynomial: G(x) = x^16 + x^15 + x^2 + 1

- Each mask, used in the hash function calculation, is compared with a 16-bit value associated with that mask. Each filter has the following:

-- 32-bit Mask: Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1, the corresponding byte is taken into the CRC-16 calculation.

-- 8-bit Offset Pointer: Specifies the byte to start the CRC-16 computation. The pointer and the mask are used together to locate the bytes to be used in the CRC-16 calculations.

The structure of the RWK Filter registers is shown in "Figure 10-11 Remote Wake-Up Packet Filter Register".

11.7.6.1.30.1 Offset

Register	Offset
MAC_RWK_PACKET_FILTER	C4h

11.7.6.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WKUPFRMFTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WKUPFRMFTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.30.3 Fields

Field	Description
31-0 WKUPFRMFTR	RWK Packet Filter This field contains the various controls of RWK Packet filter.

11.7.6.1.31 LPI Control and Status (MAC_LPI_CONTROL_STATUS)

The LPI Control and Status Register controls the LPI functions and provides the LPI interrupt status.

The LPI Control and Status Register controls the LPI functions and provides the LPI interrupt status. The status bits are cleared when this register is read.

11.7.6.1.31.1 Offset

Register	Offset
MAC_LPI_CONTROL_STATUS	D0h

11.7.6.1.31.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										LPITCSE	LPIATE	LPITXA	PLSEN	PLS	LPIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						RLPIST	TLPIST	Reserved				RLPIEX	RLPIEN	TLPIEX	TLPIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.31.3 Fields

Field	Description
31-22 —	Reserved
21 LPITCSE	LPI Tx Clock Stop Enable When this bit is set, the MAC asserts sbd_tx_clk_gating_ctrl_o signal high after it enters Tx LPI mode to indicate that the Tx clock to MAC can be stopped. LPI Tx Clock Stop Enable When this bit is set, the MAC asserts sbd_tx_clk_gating_ctrl_o signal high after it enters Tx LPI mode to indicate that the Tx clock to MAC can be stopped. When this bit is reset, the MAC does not assert sbd_tx_clk_gating_ctrl_o signal high after it enters Tx LPI mode.

Table continues on the next page...

Field	Description
	If RGMII Interface is selected, the Tx clock is required for transmitting the LPI pattern. The Tx Clock cannot be gated and so the LPITCSE bit cannot be programmed. 0 - LPI Tx Clock Stop is disabled 1 - LPI Tx Clock Stop is enabled
20 LPIATE	LPI Timer Enable This bit controls the automatic entry of the MAC Transmitter into and exit out of the LPI state. LPI Timer Enable This bit controls the automatic entry of the MAC Transmitter into and exit out of the LPI state. When LPIATE, LPITXA and LPIEN bits are set, the MAC Transmitter enters LPI state only when the complete MAC TX data path is IDLE for a period indicated by the MAC_LPI_ENTRY_TIMER register. After entering LPI state, if the data path becomes non-IDLE (due to a new packet being accepted for transmission), the Transmitter exits LPI state but does not clear LPIEN bit. This enables the re-entry into LPI state when it is IDLE again. When LPIATE is 0, the LPI Auto timer is disabled and MAC Transmitter enters LPI state based on the settings of LPITXA and LPIEN bit descriptions. 0 - LPI Timer is disabled 1 - LPI Timer is enabled
19 LPITXA	LPI Tx Automate This bit controls the behavior of the MAC when it is entering or coming out of the LPI mode on the Transmit side. LPI Tx Automate This bit controls the behavior of the MAC when it is entering or coming out of the LPI mode on the Transmit side. This bit is not functional in the EQOS-CORE configurations in which the Tx clock gating is done during the LPI mode. If the LPITXA and LPIEN bits are set to 1, the MAC enters the LPI mode only after all outstanding packets (in the core) and pending packets (in the application interface) have been transmitted. The MAC comes out of the LPI mode when the application sends any packet for transmission or the application issues a Tx FIFO Flush command. In addition, the MAC automatically clears the LPIEN bit when it exits the LPI state. If Tx FIFO Flush is set in the FTQ bit of MAC_TXQ0_OPERATION_MODE register, when the MAC is in the LPI mode, it exits the LPI mode. When this bit is 0, the LPIEN bit directly controls behavior of the MAC when it is entering or coming out of the LPI mode. 0 - LPI Tx Automate is disabled 1 - LPI Tx Automate is enabled
18 PLSEN	PHY Link Status Enable This bit enables the link status received on the RGMII, SGMII, or SMII Receive paths to be used for activating the LPI LS TIMER. PHY Link Status Enable This bit enables the link status received on the RGMII, SGMII, or SMII Receive paths to be used for activating the LPI LS TIMER. When this bit is set, the MAC uses the link-status bits of the MAC_PHYIF_CONTROL_STATUS register and the PLS bit for the LPI LS Timer trigger. When this bit is reset, the MAC ignores the link-status bits of the MAC_PHYIF_CONTROL_STATUS register and takes only the PLS bit. 0 - PHY Link Status is disabled 1 - PHY Link Status is enabled
17	PHY Link Status This bit indicates the link status of the PHY.

Table continues on the next page...

Field	Description
PLS	PHY Link Status This bit indicates the link status of the PHY. The MAC Transmitter asserts the LPI pattern only when the link status is up (OKAY) at least for the time indicated by the LPI LS TIMER. When this bit is set, the link is considered to be okay (UP) and when this bit is reset, the link is considered to be down. 0 - link is down 1 - link is okay (UP)
16 LPIEN	LPI Enable When this bit is set, it instructs the MAC Transmitter to enter the LPI state. LPI Enable When this bit is set, it instructs the MAC Transmitter to enter the LPI state. When this bit is reset, it instructs the MAC to exit the LPI state and resume normal transmission. This bit is cleared when the LPITXA bit is set and the MAC exits the LPI state because of the arrival of a new packet for transmission. 0 - LPI state is disabled 1 - LPI state is enabled
15-10 —	Reserved
9 RLPIST	Receive LPI State When this bit is set, it indicates that the MAC is receiving the LPI pattern on the GMII interface. 0 - Receive LPI state not detected 1 - Receive LPI state detected
8 TLPIST	Transmit LPI State When this bit is set, it indicates that the MAC is transmitting the LPI pattern on the GMII interface. 0 - Transmit LPI state not detected 1 - Transmit LPI state detected
7-4 —	Reserved
3 RLPIEX	Receive LPI Exit When this bit is set, it indicates that the MAC Receiver has stopped receiving the LPI pattern on the GMII interface, exited the LPI state, and resumed the normal reception. Receive LPI Exit When this bit is set, it indicates that the MAC Receiver has stopped receiving the LPI pattern on the GMII interface, exited the LPI state, and resumed the normal reception. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Note: This bit may not be set if the MAC stops receiving the LPI pattern for a very short duration, such as, less than three clock cycles of CSR clock. 0 - Receive LPI exit not detected 1 - Receive LPI exit detected
2 RLPIEN	Receive LPI Entry When this bit is set, it indicates that the MAC Receiver has received an LPI pattern and entered the LPI state. Receive LPI Entry When this bit is set, it indicates that the MAC Receiver has received an LPI pattern and entered the LPI state. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set).

Table continues on the next page...

Field	Description
	Note: This bit may not be set if the MAC stops receiving the LPI pattern for a very short duration, such as, less than three clock cycles of CSR clock. 0 - Receive LPI entry not detected 1 - Receive LPI entry detected
1 TLPIEX	Transmit LPI Exit When this bit is set, it indicates that the MAC transmitter exited the LPI state after the application cleared the LPIEN bit and the LPI TW Timer has expired. Transmit LPI Exit When this bit is set, it indicates that the MAC transmitter exited the LPI state after the application cleared the LPIEN bit and the LPI TW Timer has expired. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). 0 - Transmit LPI exit not detected 1 - Transmit LPI exit detected
0 TLPIEN	Transmit LPI Entry When this bit is set, it indicates that the MAC Transmitter has entered the LPI state because of the setting of the LPIEN bit. Transmit LPI Entry When this bit is set, it indicates that the MAC Transmitter has entered the LPI state because of the setting of the LPIEN bit. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). 0 - Transmit LPI entry not detected 1 - Transmit LPI entry detected

11.7.6.1.32 LPI Timers Control (MAC_LPI_TIMERS_CONTROL)

The LPI Timers Control register controls the timeout values in the LPI states.

The LPI Timers Control register controls the timeout values in the LPI states. It specifies the time for which the MAC transmits the LPI pattern and also the time for which the MAC waits before resuming the normal transmission.

11.7.6.1.32.1 Offset

Register	Offset
MAC_LPI_TIMERS_CONTROL	D4h

11.7.6.1.32.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved						LST									
Reset	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TWT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.32.3 Fields

Field	Description
31-26 —	Reserved
25-16 LST	LPI LS Timer This field specifies the minimum time (in milliseconds) for which the link status from the PHY should be up (OKAY) before the LPI pattern can be transmitted to the PHY. LPI LS Timer This field specifies the minimum time (in milliseconds) for which the link status from the PHY should be up (OKAY) before the LPI pattern can be transmitted to the PHY. The MAC does not transmit the LPI pattern even when the LPIEN bit is set unless the LPI LS Timer reaches the programmed terminal count. The default value of the LPI LS Timer is 1000 (1 sec) as defined in the IEEE standard.
15-0 TWT	LPI TW Timer This field specifies the minimum time (in microseconds) for which the MAC waits after it stops transmitting the LPI pattern to the PHY and before it resumes the normal transmission. LPI TW Timer This field specifies the minimum time (in microseconds) for which the MAC waits after it stops transmitting the LPI pattern to the PHY and before it resumes the normal transmission. The TLPIEX status bit is set after the expiry of this timer.

11.7.6.1.33 Tx LPI Entry Timer Control (MAC_LPI_ENTRY_TIMER)

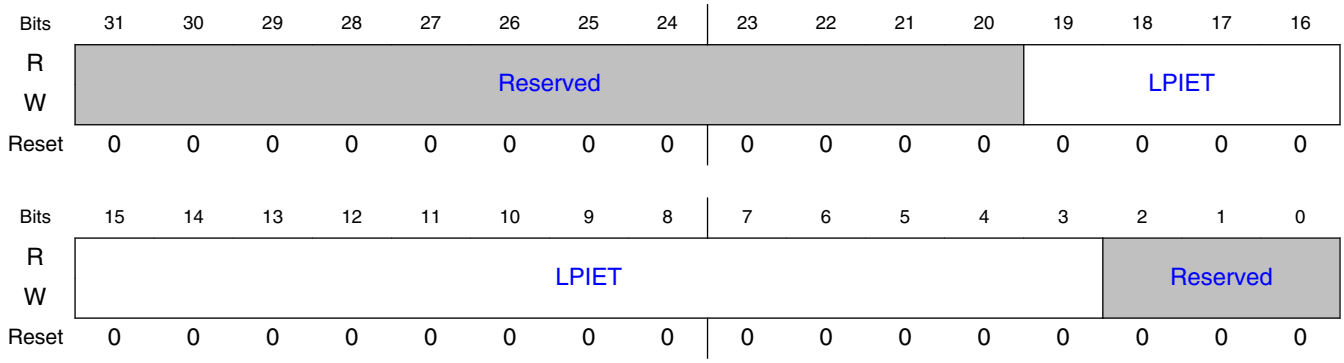
This register controls the Tx LPI entry timer.

This register controls the Tx LPI entry timer. This counter is enabled only when bit[20] (LPITE) bit of MAC_LPI_CONTROL_STATUS is set to 1.

11.7.6.1.33.1 Offset

Register	Offset
MAC_LPI_ENTRY_TIMER	D8h

11.7.6.1.33.2 Diagram



11.7.6.1.33.3 Fields

Field	Description
31-20 —	Reserved
19-3 LPIET	LPI Entry Timer This field specifies the time in microseconds the MAC waits to enter LPI mode, after it has transmitted all the frames. LPI Entry Timer This field specifies the time in microseconds the MAC waits to enter LPI mode, after it has transmitted all the frames. This field is valid and used only when LPITE and LPITXA are set to 1. Bits [2:0] are read-only so that the granularity of this timer is in steps of 8 micro-seconds.
2-0 —	Reserved

11.7.6.1.34 One-microsecond Reference Timer (MAC_ONEUS_TIC_COUNTER)

This register controls the generation of the Reference time (1 microsecond tic) for all the LPI timers. This timer has to be programmed by the software initially.

11.7.6.1.34.1 Offset

Register	Offset
MAC_ONEUS_TIC_COUNTER	DCh

11.7.6.1.34.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				TIC_1US_CNTR											
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1

11.7.6.1.34.3 Fields

Field	Description
31-12 —	Reserved
11-0 TIC_1US_CNTR	1US TIC Counter The application must program this counter so that the number of clock cycles of CSR clock is 1us. 1US TIC Counter The application must program this counter so that the number of clock cycles of CSR clock is 1us. (Subtract 1 from the value before programming). For example if the CSR clock is 100MHz then this field needs to be programmed to value 100 - 1 = 99 (which is 0x63). This is required to generate the 1US events that are used to update some of the EEE related counters.

11.7.6.1.35 PHY Interface Control and Status (MAC_PHYIF_CONTROL_STATUS)

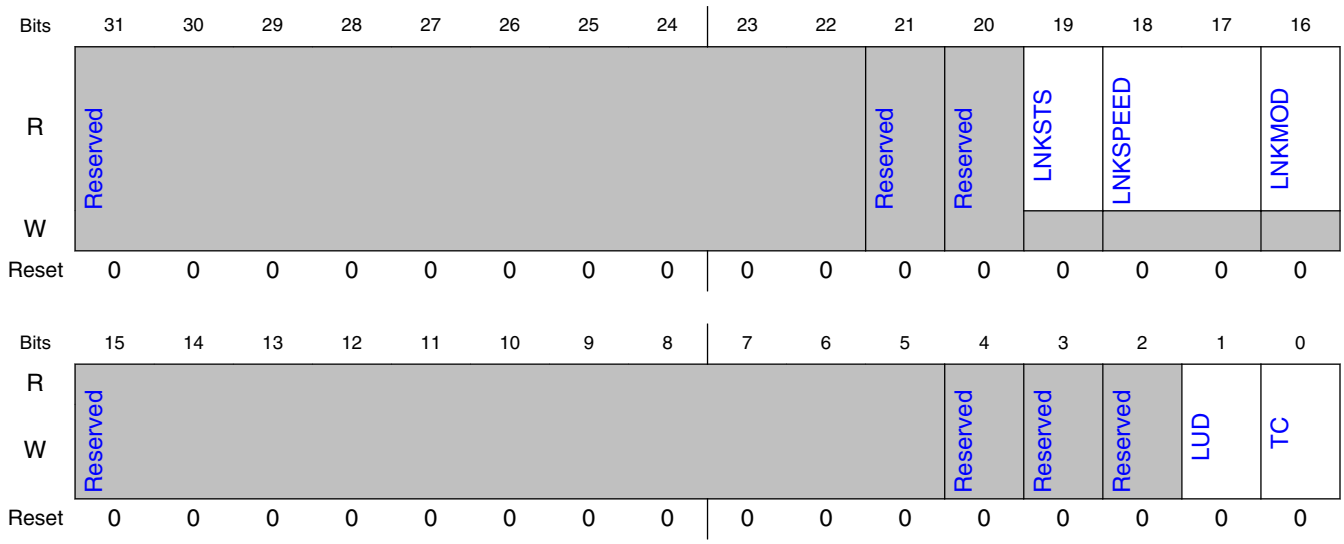
The PHY Interface Control and Status register indicates the status signals received by the SGMII, RGMII, or SMII interface (selected at reset) from the PHY.

The PHY Interface Control and Status register indicates the status signals received by the SGMII, RGMII, or SMII interface (selected at reset) from the PHY. This register is optional.

11.7.6.1.35.1 Offset

Register	Offset
MAC_PHYIF_CONTROL_STATUS	F8h

11.7.6.1.35.2 Diagram



11.7.6.1.35.3 Fields

Field	Description
31-22 —	Reserved
21 —	Reserved
20 —	Reserved
19 LNKSTS	Link Status This bit indicates whether the link is up (1'b1) or down (1'b0). 0 - Link down 1 - Link up
18-17 LNKSPEED	Link Speed This bit indicates the current speed of the link. 00 - 2.5 MHz 01 - 25 MHz 10 - 125 MHz 11 - Reserved

Table continues on the next page...

Field	Description
16 LNKMOD	Link Mode This bit indicates the current mode of operation of the link. 0 - Half-duplex mode 1 - Full-duplex mode
15-5 —	Reserved
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 LUD	Link Up or Down This bit indicates whether the link is up or down during transmission of configuration in the RGMII, SGMII, or SMII interface. 0 - Link down 1 - Link up
0 TC	Transmit Configuration in RGMII, SGMII, or SMII When set, this bit enables the transmission of duplex mode, link speed, and link up or down information to the PHY in the RGMII, SMII, or SGMII port. Transmit Configuration in RGMII, SGMII, or SMII When set, this bit enables the transmission of duplex mode, link speed, and link up or down information to the PHY in the RGMII, SMII, or SGMII port. When this bit is reset, no such information is driven to the PHY. The details of this feature are provided in the following sections: - "Reduced Gigabit Media Independent Interface" - "Serial Media Independent Interface" - "Serial Gigabit Media Independent Interface" 0 - Disable Transmit Configuration in RGMII, SGMII, or SMII 1 - Enable Transmit Configuration in RGMII, SGMII, or SMII

11.7.6.1.36 MAC Version (MAC_VERSION)

The version register identifies the version of the DWC_ether_qos.

The version register identifies the version of the DWC_ether_qos. This register contains two bytes: one that Synopsys uses to identify the core release number, and the other that you set while configuring the core.

11.7.6.1.36.1 Offset

Register	Offset
MAC_VERSION	110h

11.7.6.1.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	USERVER								SNPSVER							
W																
Reset	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	1

11.7.6.1.36.3 Fields

Field	Description
31-16 —	Reserved
15-8 USERVER	User-defined Version (8'h10)
7-0 SNPSVER	Synopsys-defined Version

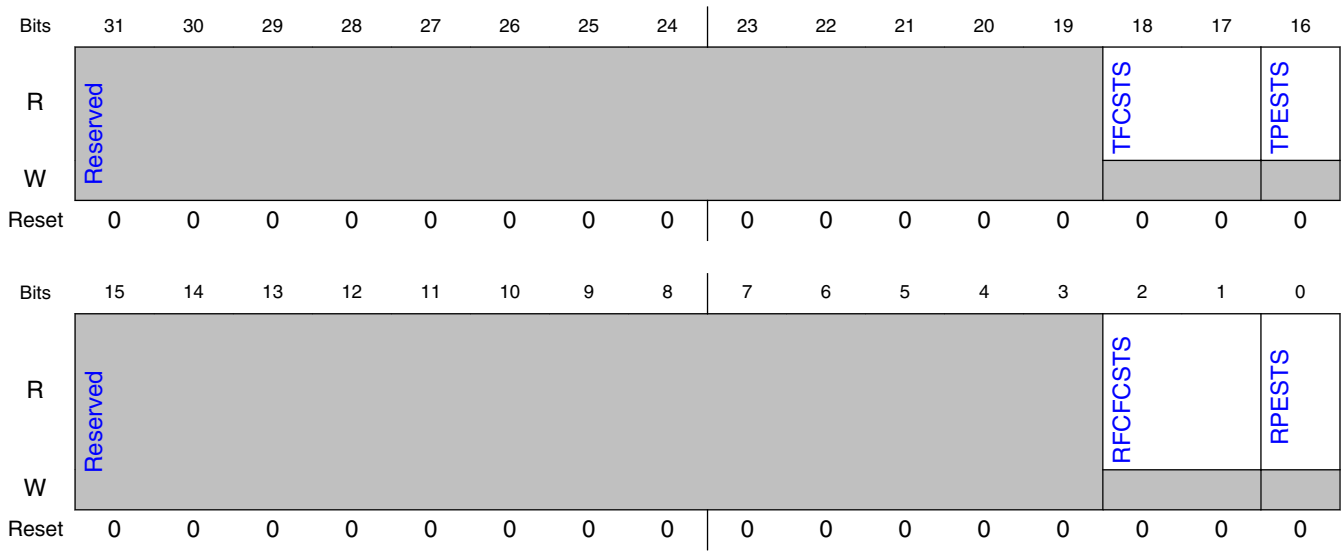
11.7.6.1.37 MAC Debug (MAC_DEBUG)

The Debug register provides the debug status of various MAC blocks.

11.7.6.1.37.1 Offset

Register	Offset
MAC_DEBUG	114h

11.7.6.1.37.2 Diagram



11.7.6.1.37.3 Fields

Field	Description
31-19 —	Reserved
18-17 TFCSTS	MAC Transmit Packet Controller Status This field indicates the state of the MAC Transmit Packet Controller module. 00 - Idle state 01 - Waiting for one of the following: Status of the previous packet OR IPG or back off period to be over 10 - Generating and transmitting a Pause control packet (in full-duplex mode) 11 - Transferring input packet for transmission
16 TPESTS	MAC GMII or MII Transmit Protocol Engine Status When this bit is set, it indicates that the MAC GMII or MII transmit protocol engine is actively transmitting data, and it is not in the Idle state. 0 - MAC GMII Transmit Protocol Engine Status not detected 1 - MAC GMII Transmit Protocol Engine Status detected
15-3 —	Reserved
2-1 RFCFCSTS	MAC Receive Packet Controller FIFO Status When this bit is set, this field indicates the active state of the small FIFO Read and Write controllers of the MAC Receive Packet Controller module.
0 RPESTS	MAC GMII Receive Protocol Engine Status When this bit is set, it indicates that the MAC GMII receive protocol engine is actively receiving data, and it is not in the Idle state. 0 - MAC GMII Receive Protocol Engine Status not detected 1 - MAC GMII Receive Protocol Engine Status detected

11.7.6.1.38 Optional Features or Functions 0 (MAC_HW_FEATURE0)

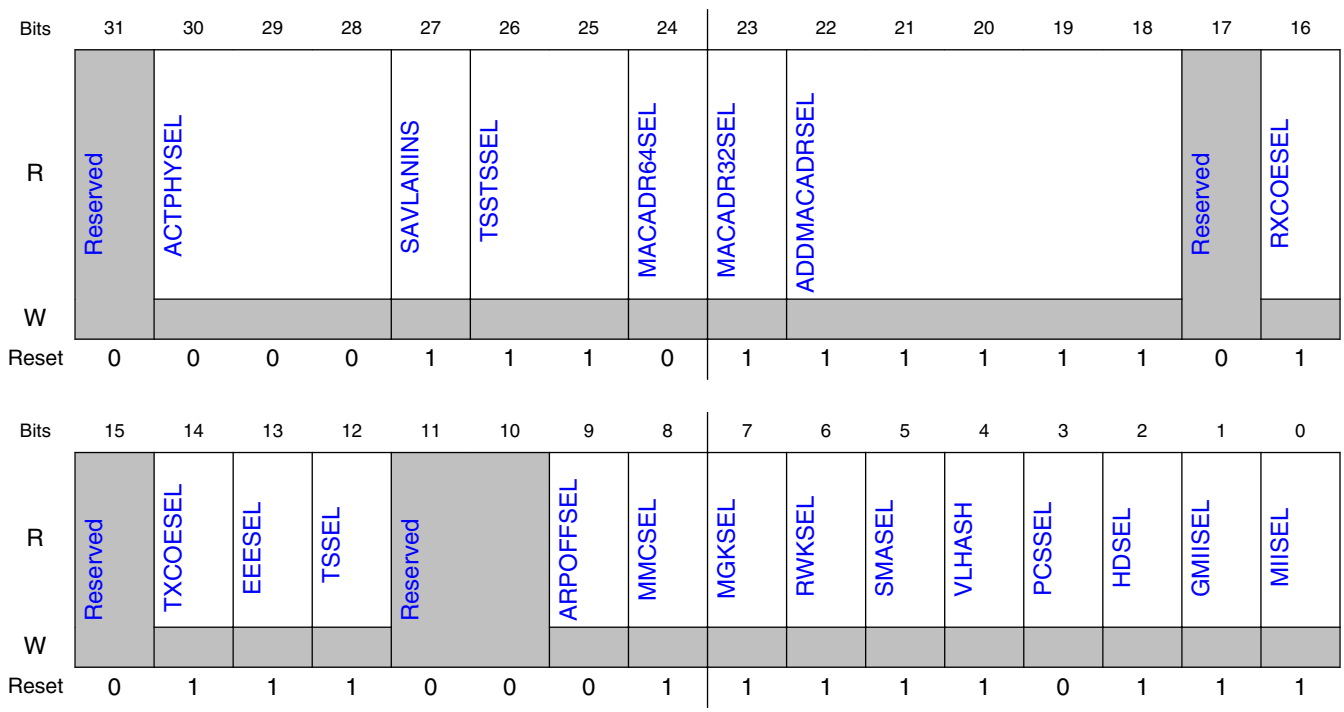
This register indicates the presence of first set of the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

Note: All bits are set or reset according to the features selected while configuring the core in coreConsultant.

11.7.6.1.38.1 Offset

Register	Offset
MAC_HW_FEATURE0	11Ch

11.7.6.1.38.2 Diagram



11.7.6.1.38.3 Fields

Field	Description
31	Reserved

Table continues on the next page...

Field	Description
—	
30-28 ACTPHYSEL	Active PHY Selected When you have multiple PHY interfaces in your configuration, this field indicates the sampled value of phy_intf_sel_i during reset de-assertion. 000 - GMII 001 - RGMII 010 - SGMII 011 - TBI 100 - RMII 101 - RTBI 110 - SMII 111 - RevMII
27 SAVLANINS	Source Address or VLAN Insertion Enable This bit is set to 1 when the Enable SA and VLAN Insertion on Tx option is selected 0 - Source Address or VLAN Insertion Enable option is not selected 1 - Source Address or VLAN Insertion Enable option is selected
26-25 TSSTSSEL	Timestamp System Time Source This bit indicates the source of the Timestamp system time: This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected 00 - Reserved 01 - Internal 10 - External 11 - Both
24 MACADR64SEL	MAC Addresses 64-127 Selected This bit is set to 1 when the Enable Additional 64 MAC Address Registers (64-127) option is selected 0 - MAC Addresses 64-127 Select option is not selected 1 - MAC Addresses 64-127 Select option is selected
23 MACADR32SEL	MAC Addresses 32-63 Selected This bit is set to 1 when the Enable Additional 32 MAC Address Registers (32-63) option is selected 0 - MAC Addresses 32-63 Select option is not selected 1 - MAC Addresses 32-63 Select option is selected
22-18 ADDMACADRSEL	MAC Addresses 1-31 Selected This bit is set to 1 when the non-zero value is selected for Enable Additional 1-31 MAC Address Registers option
17 —	Reserved
16 RXCOESEL	Receive Checksum Offload Enabled This bit is set to 1 when the Enable Receive TCP/IP Checksum Check option is selected 0 - Receive Checksum Offload Enable option is not selected 1 - Receive Checksum Offload Enable option is selected
15 —	Reserved
14	Transmit Checksum Offload Enabled This bit is set to 1 when the Enable Transmit TCP/IP Checksum Insertion option is selected

Table continues on the next page...

Field	Description
TXCOESEL	0 - Transmit Checksum Offload Enable option is not selected 1 - Transmit Checksum Offload Enable option is selected
13 EEESEL	Energy Efficient Ethernet Enabled This bit is set to 1 when the Enable Energy Efficient Ethernet (EEE) option is selected 0 - Energy Efficient Ethernet Enable option is not selected 1 - Energy Efficient Ethernet Enable option is selected
12 TSSEL	IEEE 1588-2008 Timestamp Enabled This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected 0 - IEEE 1588-2008 Timestamp Enable option is not selected 1 - IEEE 1588-2008 Timestamp Enable option is selected
11-10 —	Reserved
9 ARPOFFSEL	ARP Offload Enabled This bit is set to 1 when the Enable IPv4 ARP Offload option is selected 0 - ARP Offload Enable option is not selected 1 - ARP Offload Enable option is selected
8 MMCSEL	RMON Module Enable This bit is set to 1 when the Enable MAC Management Counters (MMC) option is selected 0 - RMON Module Enable option is not selected 1 - RMON Module Enable option is selected
7 MGKSEL	PMT Magic Packet Enable This bit is set to 1 when the Enable Magic Packet Detection option is selected 0 - PMT Magic Packet Enable option is not selected 1 - PMT Magic Packet Enable option is selected
6 RWKSEL	PMT Remote Wake-up Packet Enable This bit is set to 1 when the Enable Remote Wake-Up Packet Detection option is selected 0 - PMT Remote Wake-up Packet Enable option is not selected 1 - PMT Remote Wake-up Packet Enable option is selected
5 SMASEL	SMA (MDIO) Interface This bit is set to 1 when the Enable Station Management (MDIO Interface) option is selected 0 - SMA (MDIO) Interface not selected 1 - SMA (MDIO) Interface selected
4 VLHASH	VLAN Hash Filter Selected This bit is set to 1 when the Enable VLAN Hash Table Based Filtering option is selected 0 - VLAN Hash Filter not selected 1 - VLAN Hash Filter selected
3 PCSSEL	PCS Registers (TBI, SGMII, or RTBI PHY interface) This bit is set to 1 when the TBI, SGMII, or RTBI PHY interface option is selected 0 - No PCS Registers (TBI, SGMII, or RTBI PHY interface) 1 - PCS Registers (TBI, SGMII, or RTBI PHY interface)
2 HDSEL	Half-duplex Support This bit is set to 1 when the half-duplex mode is selected 0 - No Half-duplex support 1 - Half-duplex support
1	1000 Mbps Support This bit is set to 1 when 1000 Mbps is selected as the Mode of Operation

Table continues on the next page...

Field	Description
GMIISEL	0 - No 1000 Mbps support 1 - 1000 Mbps support
0 MIISEL	10 or 100 Mbps Support This bit is set to 1 when 10/100 Mbps is selected as the Mode of Operation 0 - No 10 or 100 Mbps support 1 - 10 or 100 Mbps support

11.7.6.1.39 Optional Features or Functions 1 (MAC_HW_FEATURE1)

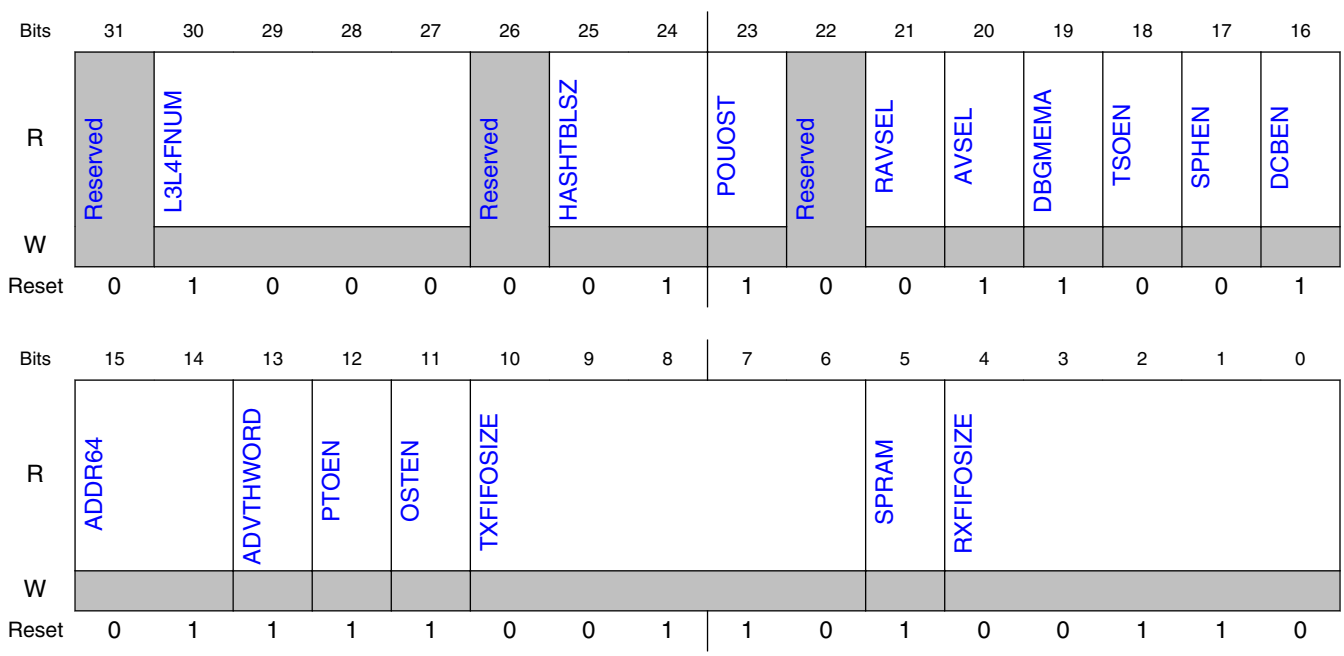
This register indicates the presence of second set of the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

Note: All bits are set or reset according to the features selected while configuring the core in coreConsultant.

11.7.6.1.39.1 Offset

Register	Offset
MAC_HW_FEATURE1	120h

11.7.6.1.39.2 Diagram



11.7.6.1.39.3 Fields

Field	Description
31 —	Reserved
30-27 L3L4FNUM	Total number of L3 or L4 Filters This field indicates the total number of L3 or L4 filters: 0000 - No L3 or L4 Filter 0001 - 1 L3 or L4 Filter 0010 - 2 L3 or L4 Filters 0011 - 3 L3 or L4 Filters 0100 - 4 L3 or L4 Filters 0101 - 5 L3 or L4 Filters 0110 - 6 L3 or L4 Filters 0111 - 7 L3 or L4 Filters 1000 - 8 L3 or L4 Filters
26 —	Reserved
25-24 HASHTBLSZ	Hash Table Size This field indicates the size of the hash table: 00 - No hash table 01 - 64 10 - 128 11 - 256
23 POUOST	One Step for PTP over UDP/IP Feature Enable This bit is set to 1 when the Enable One step timestamp for PTP over UDP/IP feature is selected. 0 - One Step for PTP over UDP/IP Feature is not selected 1 - One Step for PTP over UDP/IP Feature is selected
22 —	Reserved
21 RAVSEL	Rx Side Only AV Feature Enable This bit is set to 1 when the Enable Audio Video Bridging option on Rx Side Only is selected. 0 - Rx Side Only AV Feature is not selected 1 - Rx Side Only AV Feature is selected
20 AVSEL	AV Feature Enable This bit is set to 1 when the Enable Audio Video Bridging option is selected. 0 - AV Feature is not selected 1 - AV Feature is selected
19 DBGMEMA	DMA Debug Registers Enable This bit is set to 1 when the Debug Mode Enable option is selected 0 - DMA Debug Registers option is not selected 1 - DMA Debug Registers option is selected
18	TCP Segmentation Offload Enable This bit is set to 1 when the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected

Table continues on the next page...

Field	Description
TSOEN	0 - TCP Segmentation Offload Feature is not selected 1 - TCP Segmentation Offload Feature is selected
17 SPHEN	Split Header Feature Enable This bit is set to 1 when the Enable Split Header Structure option is selected 0 - Split Header Feature is not selected 1 - Split Header Feature is selected
16 DCBEN	DCB Feature Enable This bit is set to 1 when the Enable Data Center Bridging option is selected 0 - DCB Feature is not selected 1 - DCB Feature is selected
15-14 ADDR64	Address Width. Address Width. This field indicates the configured address width: 00 - 32 01 - 40 10 - 48 11 - Reserved
13 ADVTHWORD	IEEE 1588 High Word Register Enable This bit is set to 1 when the Add IEEE 1588 Higher Word Register option is selected 0 - IEEE 1588 High Word Register option is not selected 1 - IEEE 1588 High Word Register option is selected
12 PTOEN	PTP Offload Enable This bit is set to 1 when the Enable PTP Timestamp Offload Feature is selected. 0 - PTP Offload feature is not selected 1 - PTP Offload feature is selected
11 OSTEN	One-Step Timestamping Enable This bit is set to 1 when the Enable One-Step Timestamp Feature is selected. 0 - One-Step Timestamping feature is not selected 1 - One-Step Timestamping feature is selected
10-6 TXFIFOSIZE	MTL Transmit FIFO Size This field contains the configured value of MTL Tx FIFO in bytes expressed as Log to base 2 minus 7, that is, Log2(TXFIFO_SIZE) -7: 00000 - 128 bytes 00001 - 256 bytes 00010 - 512 bytes 00011 - 1024 bytes 00100 - 2048 bytes 00101 - 4096 bytes 00110 - 8192 bytes 00111 - 16384 bytes 01000 - 32 KB 01001 - 64 KB 01010 - 128 KB 01011 - Reserved

Table continues on the next page...

Field	Description
5 SPRAM	Single Port RAM Enable This bit is set to 1 when the Use single port RAM Feature is selected. 0 - Single Port RAM feature is not selected 1 - Single Port RAM feature is selected
4-0 RXFIFOSIZE	MTL Receive FIFO Size This field contains the configured value of MTL Rx FIFO in bytes expressed as Log to base 2 minus 7, that is, Log2(RXFIFO_SIZE) -7: 00000 - 128 bytes 00001 - 256 bytes 00010 - 512 bytes 00011 - 1024 bytes 00100 - 2048 bytes 00101 - 4096 bytes 00110 - 8192 bytes 00111 - 16384 bytes 01000 - 32 KB 01001 - 64 KB 01010 - 128 KB 01011 - 256 KB 01100 - Reserved

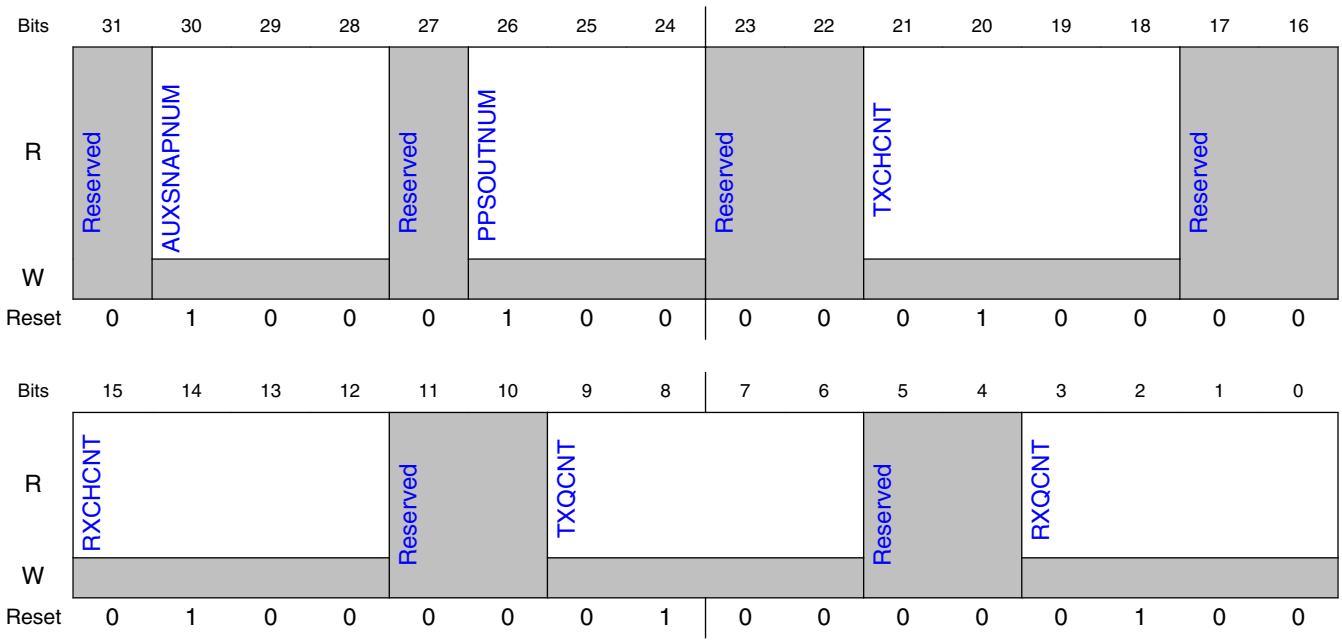
11.7.6.1.40 Optional Features or Functions 2 (MAC_HW_FEATURE2)

This register indicates the presence of third set of the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

11.7.6.1.40.1 Offset

Register	Offset
MAC_HW_FEATURE2	124h

11.7.6.1.40.2 Diagram



11.7.6.1.40.3 Fields

Field	Description
31 —	Reserved
30-28 AUXSNAPNUM	Number of Auxiliary Snapshot Inputs This field indicates the number of auxiliary snapshot inputs: 000 - No auxiliary input 001 - 1 auxiliary input 010 - 2 auxiliary input 011 - 3 auxiliary input 100 - 4 auxiliary input 101 - Reserved
27 —	Reserved
26-24 PPSOUTNUM	Number of PPS Outputs This field indicates the number of PPS outputs: 000 - No PPS output 001 - 1 PPS output 010 - 2 PPS output 011 - 3 PPS output 100 - 4 PPS output 101 - Reserved
23-22	Reserved

Table continues on the next page...

Field	Description
—	
21-18 TXCHCNT	Number of DMA Transmit Channels This field indicates the number of DMA Transmit channels: 0000 - 1 MTL Tx Channel 0001 - 2 MTL Tx Channels 0010 - 3 MTL Tx Channels 0011 - 4 MTL Tx Channels 0100 - 5 MTL Tx Channels 0101 - Reserved 0110 - Reserved 0111 - Reserved
17-16 —	Reserved
15-12 RXCHCNT	Number of DMA Receive Channels This field indicates the number of DMA Receive channels: 0000 - 1 MTL Rx Channel 0001 - 2 MTL Rx Channels 0010 - 3 MTL Rx Channels 0011 - 4 MTL Rx Channels 0100 - 5 MTL Rx Channels 0101 - Reserved 0110 - Reserved 0111 - Reserved
11-10 —	Reserved
9-6 TXQCNT	Number of MTL Transmit Queues This field indicates the number of MTL Transmit queues: 0000 - 1 MTL Tx Queue 0001 - 2 MTL Tx Queues 0010 - 3 MTL Tx Queues 0011 - 4 MTL Tx Queues 0100 - 5 MTL Tx Queues 0101 - Reserved 0110 - Reserved 0111 - Reserved
5-4 —	Reserved
3-0 RXQCNT	Number of MTL Receive Queues This field indicates the number of MTL Receive queues: 0000 - 1 MTL Rx Queue 0001 - 2 MTL Rx Queues 0010 - 3 MTL Rx Queues 0011 - 4 MTL Rx Queues

Field	Description
	0100 - 5 MTL Rx Queues 0101 - Reserved 0110 - Reserved 0111 - Reserved

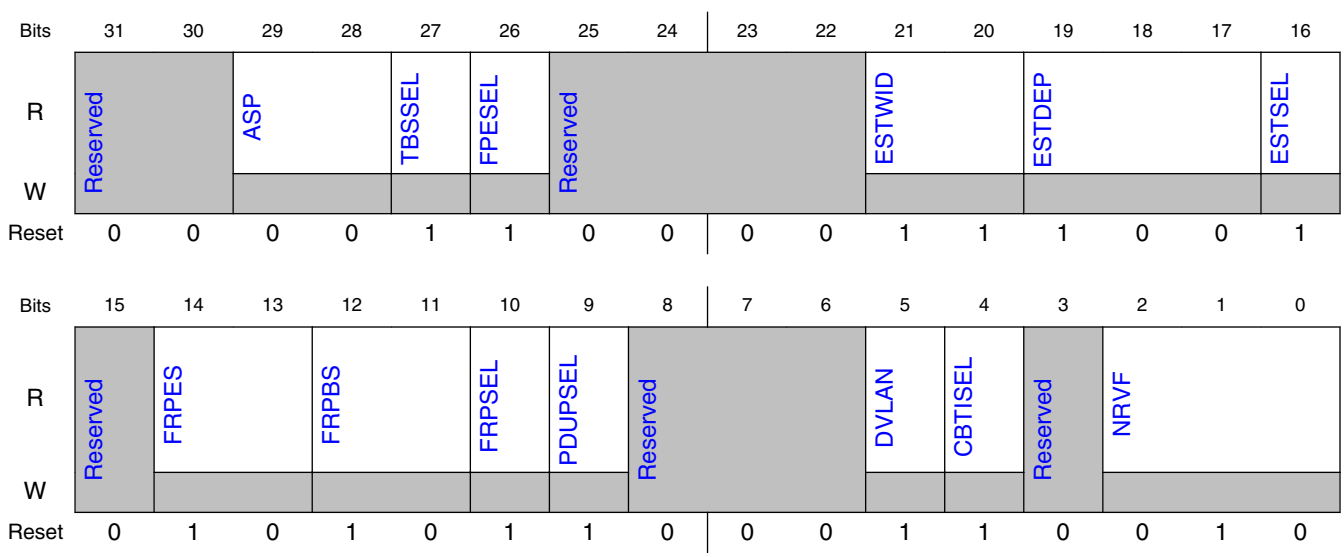
11.7.6.1.41 Optional Features or Functions 3 (MAC_HW_FEATURE3)

This register indicates the presence of fourth set the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

11.7.6.1.41.1 Offset

Register	Offset
MAC_HW_FEATURE3	128h

11.7.6.1.41.2 Diagram



11.7.6.1.41.3 Fields

Field	Description
31-30 —	Reserved
29-28 ASP	Automotive Safety Package Following are the encoding for the different Safety features 00 - No Safety features selected 01 - Only "ECC protection for external memory" feature is selected 10 - All the Automotive Safety features are selected without the "Parity Port Enable for external interface" feature 11 - All the Automotive Safety features are selected with the "Parity Port Enable for external interface" feature
27 TBSSEL	Time Based Scheduling Enable This bit is set to 1 when the Time Based Scheduling feature is selected. 0 - Time Based Scheduling Enable feature is not selected 1 - Time Based Scheduling Enable feature is selected
26 FPESEL	Frame Preemption Enable This bit is set to 1 when the Enable Frame preemption feature is selected. 0 - Frame Preemption Enable feature is not selected 1 - Frame Preemption Enable feature is selected
25-22 —	Reserved
21-20 ESTWID	Width of the Time Interval field in the Gate Control List This field indicates the width of the Configured Time Interval Field 00 - Width not configured 01 - 16 10 - 20 11 - 24
19-17 ESTDEP	Depth of the Gate Control List This field indicates the depth of Gate Control list expressed as Log2(DWC_EQOS_EST_DEP)-5 000 - No Depth configured 001 - 64 010 - 128 011 - 256 100 - 512 101 - 1024 110 - Reserved
16 ESTSEL	Enhancements to Scheduling Traffic Enable This bit is set to 1 when the Enable Enhancements to Scheduling Traffic feature is selected. 0 - Enable Enhancements to Scheduling Traffic feature is not selected 1 - Enable Enhancements to Scheduling Traffic feature is selected
15 —	Reserved

Table continues on the next page...

Field	Description
14-13 FRPES	Flexible Receive Parser Table Entries size This field indicates the Max Number of Parser Entries supported by Flexible Receive Parser. 00 - 64 Entries 01 - 128 Entries 10 - 256 Entries 11 - Reserved
12-11 FRPBS	Flexible Receive Parser Buffer size This field indicates the supported Max Number of bytes of the packet data to be Parsed by Flexible Receive Parser. 00 - 64 Bytes 01 - 128 Bytes 10 - 256 Bytes 11 - Reserved
10 FRPSEL	Flexible Receive Parser Selected This bit is set to 1 when the Enable Flexible Programmable Receive Parser option is selected. 0 - Flexible Receive Parser feature is not selected 1 - Flexible Receive Parser feature is selected
9 PDUPSEL	Broadcast/Multicast Packet Duplication This bit is set to 1 when the Broadcast/Multicast Packet Duplication feature is selected. 0 - Broadcast/Multicast Packet Duplication feature is not selected 1 - Broadcast/Multicast Packet Duplication feature is selected
8-6 —	Reserved
5 DVLAN	Double VLAN Tag Processing Selected This bit is set to 1 when the Enable Double VLAN Processing Feature is selected. 0 - Double VLAN option is not selected 1 - Double VLAN option is selected
4 CBTISEL	Queue/Channel based VLAN tag insertion on Tx Enable This bit is set to 1 when the Enable Queue/Channel based VLAN tag insertion on Tx Feature is selected. 0 - Enable Queue/Channel based VLAN tag insertion on Tx feature is not selected 1 - Enable Queue/Channel based VLAN tag insertion on Tx feature is selected
3 —	Reserved
2-0 NRVF	Number of Extended VLAN Tag Filters Enabled This field indicates the Number of Extended VLAN Tag Filters selected: 000 - No Extended Rx VLAN Filters 001 - 4 Extended Rx VLAN Filters 010 - 8 Extended Rx VLAN Filters 011 - 16 Extended Rx VLAN Filters 100 - 24 Extended Rx VLAN Filters 101 - 32 Extended Rx VLAN Filters 110 - Reserved

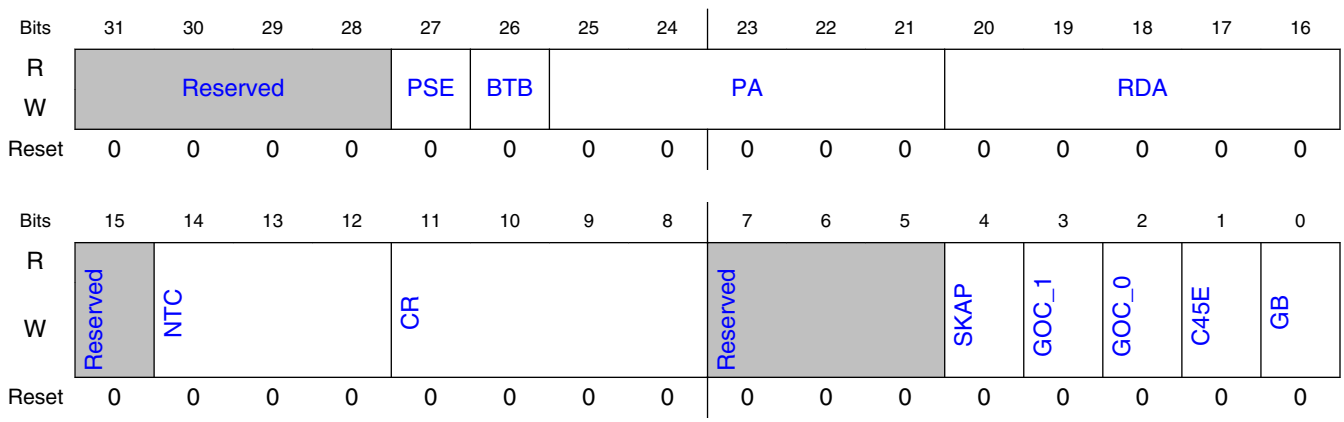
11.7.6.1.42 MDIO Address (MAC_MDIO_ADDRESS)

The MDIO Address register controls the management cycles to external PHY through a management interface.

11.7.6.1.42.1 Offset

Register	Offset
MAC_MDIO_ADDRESS	200h

11.7.6.1.42.2 Diagram



11.7.6.1.42.3 Fields

Field	Description
31-28 —	Reserved
27 PSE	Preamble Suppression Enable When this bit is set, the SMA suppresses the 32-bit preamble and transmits MDIO frames with only 1 preamble bit. Preamble Suppression Enable When this bit is set, the SMA suppresses the 32-bit preamble and transmits MDIO frames with only 1 preamble bit. When this bit is 0, the MDIO frame always has 32 bits of preamble as defined in the IEEE specifications. 0 - Preamble Suppression disabled 1 - Preamble Suppression enabled

Table continues on the next page...

Field	Description
26 BTB	Back to Back transactions When this bit is set and the NTC has value greater than 0, then the MAC informs the completion of a read or write command at the end of frame transfer (before the trailing clocks are transmitted). Back to Back transactions When this bit is set and the NTC has value greater than 0, then the MAC informs the completion of a read or write command at the end of frame transfer (before the trailing clocks are transmitted). The software can thus initiate the next command which is executed immediately irrespective of the number trailing clocks generated for the previous frame. When this bit is reset, then the read/write command completion (GB is cleared) only after the trailing clocks are generated. In this mode, it is ensured that the NTC is always generated after each frame. This bit must not be set when NTC=0. 0 - Back to Back transactions disabled 1 - Back to Back transactions enabled
25-21 PA	Physical Layer Address This field indicates which Clause 22 PHY devices (out of 32 devices) the MAC is accessing. Physical Layer Address This field indicates which Clause 22 PHY devices (out of 32 devices) the MAC is accessing. This field indicates which Clause 45 capable PHYs (out of 32 PHYs) the MAC is accessing.
20-16 RDA	Register/Device Address These bits select the PHY register in selected Clause 22 PHY device. Register/Device Address These bits select the PHY register in selected Clause 22 PHY device. These bits select the Device (MMD) in selected Clause 45 capable PHY.
15 —	Reserved
14-12 NTC	Number of Trailing Clocks This field controls the number of trailing clock cycles generated on gmii_mdc_o (MDC) after the end of transmission of MDIO frame. Number of Trailing Clocks This field controls the number of trailing clock cycles generated on gmii_mdc_o (MDC) after the end of transmission of MDIO frame. The valid values can be from 0 to 7. Programming the value to 3'h3 indicates that there are additional three clock cycles on the MDC line after the end of MDIO frame transfer.
11-8 CR	CSR Clock Range The CSR Clock Range selection determines the frequency of the MDC clock according to the CSR clock frequency used in your design: - 0000: CSR clock = 60-100 MHz; MDC clock = CSR clock/42 - 0001: CSR clock = 100-150 MHz; MDC clock = CSR clock/62 - 0010: CSR clock = 20-35 MHz; MDC clock = CSR clock/16 - 0011: CSR clock = 35-60 MHz; MDC clock = CSR clock/26 - 0100: CSR clock = 150-250 MHz; MDC clock = CSR clock/102 - 0101: CSR clock = 250-300 MHz; MDC clock = CSR clock/124 - 0110: CSR clock = 300-500 MHz; MDC clock = CSR clock/204 - 0111: CSR clock = 500-800 MHz; MDC clock = CSR clock/324 The suggested range of CSR clock frequency applicable for each value (when Bit 11 = 0) ensures that the MDC clock is approximately between 1. CSR Clock Range The CSR Clock Range selection determines the frequency of the MDC clock according to the CSR clock frequency used in your design:

Table continues on the next page...

Field	Description
	- 0000: CSR clock = 60-100 MHz; MDC clock = CSR clock/42 - 0001: CSR clock = 100-150 MHz; MDC clock = CSR clock/62 - 0010: CSR clock = 20-35 MHz; MDC clock = CSR clock/16 - 0011: CSR clock = 35-60 MHz; MDC clock = CSR clock/26 - 0100: CSR clock = 150-250 MHz; MDC clock = CSR clock/102 - 0101: CSR clock = 250-300 MHz; MDC clock = CSR clock/124 - 0110: CSR clock = 300-500 MHz; MDC clock = CSR clock/204 - 0111: CSR clock = 500-800 MHz; MDC clock = CSR clock/324 The suggested range of CSR clock frequency applicable for each value (when Bit 11 = 0) ensures that the MDC clock is approximately between 1.0 MHz to 2.5 MHz freqency range. When Bit 11 is set, you can achieve a higher frequency of the MDC clock than the frequency limit of 2.5 MHz (specified in the IEEE 802.3) and program a clock divider of lower value. For example, when CSR clock is of 100 MHz frequency and you program these bits as 1010, the resultant MDC clock is of 12.5 MHz which is above the range specified in IEEE 802.3. Program the following values only if the interfacing chips support faster MDC clocks: - 1000: CSR clock/4 - 1001: CSR clock/6 - 1010: CSR clock/8 - 1011: CSR clock/10 - 1100: CSR clock/12 - 1101: CSR clock/14 - 1110: CSR clock/16 - 1111: CSR clock/18 These bits are not used for accessing RevMII. These bits are read-only if the RevMII interface is selected as single PHY interface.
7-5 —	Reserved
4 SKAP	Skip Address Packet When this bit is set, the SMA does not send the address packets before read, write, or post-read increment address packets. Skip Address Packet When this bit is set, the SMA does not send the address packets before read, write, or post-read increment address packets. This bit is valid only when C45E is set. 0 - Skip Address Packet is disabled 1 - Skip Address Packet is enabled
3 GOC_1	GMII Operation Command 1 This bit is higher bit of the operation command to the PHY or RevMII, GOC_1 and GOC_O is encoded as follows: - 00: Reserved - 01: Write - 10: Post Read Increment Address for Clause 45 PHY - 11: Read When Clause 22 PHY or RevMII is enabled, only Write and Read commands are valid. 0 - GMII Operation Command 1 is disabled 1 - GMII Operation Command 1 is enabled
2 GOC_0	GMII Operation Command 0 This is the lower bit of the operation command to the PHY or RevMII. GMII Operation Command 0

Table continues on the next page...

Field	Description
	This is the lower bit of the operation command to the PHY or RevMII. When in SMA mode (MDIO master) this bit along with GOC_1 determines the operation to be performed to the PHY. When only RevMII is selected in configuration this bit is read-only and tied to 1. 0 - GMII Operation Command 0 is disabled 1 - GMII Operation Command 0 is enabled
1 C45E	Clause 45 PHY Enable When this bit is set, Clause 45 capable PHY is connected to MDIO. Clause 45 PHY Enable When this bit is set, Clause 45 capable PHY is connected to MDIO. When this bit is reset, Clause 22 capable PHY is connected to MDIO. 0 - Clause 45 PHY is disabled 1 - Clause 45 PHY is enabled
0 GB	GMII Busy The application sets this bit to instruct the SMA to initiate a Read or Write access to the MDIO slave. GMII Busy The application sets this bit to instruct the SMA to initiate a Read or Write access to the MDIO slave. The MAC clears this bit after the MDIO frame transfer is completed. Hence the software must not write or change any of the fields in MAC_MDIO_ADDRESS and MAC_MDIO_DATA registers as long as this bit is set. For write transfers, the application must first write 16-bit data in the GDI field (and also RA field when C45E is set) in MAC_MDIO_DATA register before setting this bit. When C45E is set, it should also write into the RA field of MAC_MDIO_DATA register before initiating a read transfer. When a read transfer is completed (GB=0), the data read from the PHY register is valid in the GD field of the MAC_MDIO_DATA register. Note: Even if the addressed PHY is not present, there is no change in the functionality of this bit. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - GMII Busy is disabled 1 - GMII Busy is enabled

11.7.6.1.43 MAC MDIO Data (MAC_MDIO_DATA)

The MAC_MDIO_DATA register stores the Write data to be written to the PHY register located at the address specified in MAC_MDIO_ADDRESS. This register also stores the Read data from the PHY register located at the address specified by MAC_MDIO_ADDRESS register.

11.7.6.1.43.1 Offset

Register	Offset
MAC_MDIO_DATA	204h

11.7.6.1.43.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.43.3 Fields

Field	Description
31-16 RA	Register Address This field is valid only when C45E is set. Register Address This field is valid only when C45E is set. It contains the Register Address in the PHY to which the MDIO frame is intended for.
15-0 GD	GMII Data This field contains the 16-bit data value read from the PHY or RevMII after a Management Read operation or the 16-bit data value to be written to the PHY or RevMII before a Management Write operation.

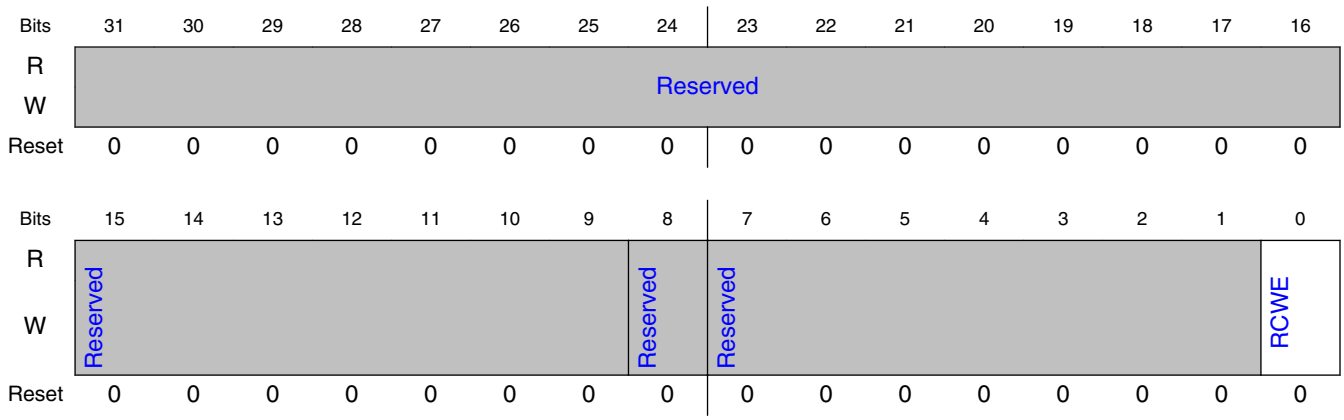
11.7.6.1.44 CSR Software Control (MAC_CSR_SW_CTRL)

This register contains SW programmable controls for changing the CSR access response and status bits clearing.

11.7.6.1.44.1 Offset

Register	Offset
MAC_CSR_SW_CTRL	230h

11.7.6.1.44.2 Diagram



11.7.6.1.44.3 Fields

Field	Description
31-9 —	Reserved
8 —	Reserved
7-1 —	Reserved
0 RCWE	Register Clear on Write 1 Enable When this bit is set, the access mode of some register fields changes to Clear on Write 1, the application needs to set that respective bit to 1 to clear it. Register Clear on Write 1 Enable When this bit is set, the access mode of some register fields changes to Clear on Write 1, the application needs to set that respective bit to 1 to clear it. When this bit is reset, the access mode of these register fields remain as Clear on Read. 0 - Register Clear on Write 1 is disabled 1 - Register Clear on Write 1 is enabled

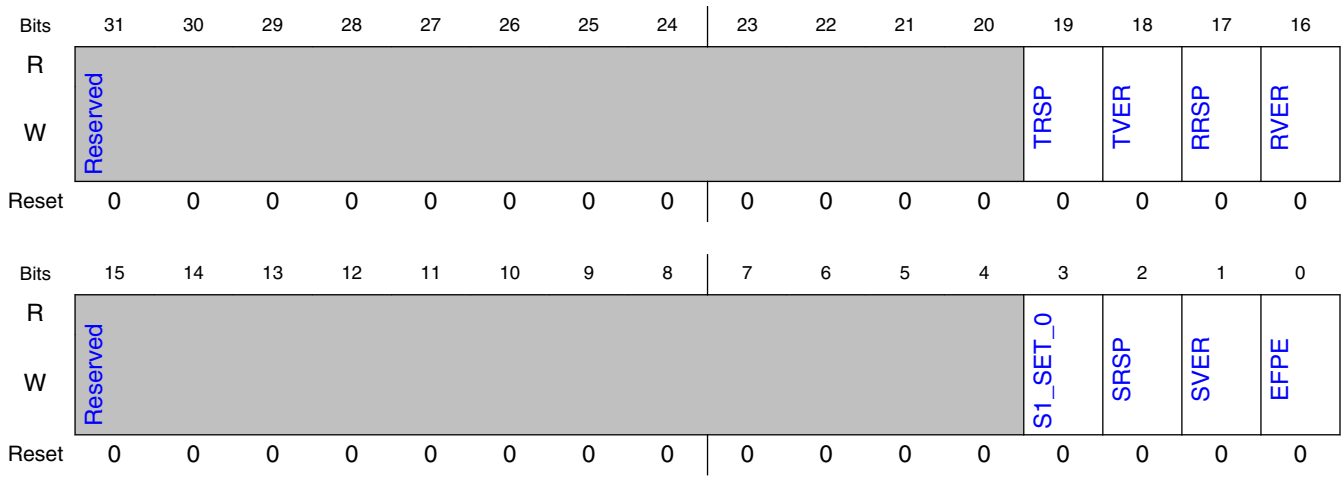
11.7.6.1.45 Frame Preemption Control (MAC_FPE_CTRL_STS)

This register controls the operation of Frame Preemption.

11.7.6.1.45.1 Offset

Register	Offset
MAC_FPE_CTRL_STS	234h

11.7.6.1.45.2 Diagram



11.7.6.1.45.3 Fields

Field	Description
31-20 —	Reserved
19 TRSP	Transmitted Respond Frame Set when a Respond mPacket is transmitted (triggered by setting SRSP field). Transmitted Respond Frame Set when a Respond mPacket is transmitted (triggered by setting SRSP field). An interrupt can be generated for this event if FPEIE bit of MAC_INTERRUPT_ENABLE is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Not transmitted Respond Frame 1 - transmitted Respond Frame
18 TVER	Transmitted Verify Frame Set when a Verify mPacket is transmitted (triggered by setting SVER field). Transmitted Verify Frame Set when a Verify mPacket is transmitted (triggered by setting SVER field). An interrupt can be generated for this event if FPEIE bit of MAC_INTERRUPT_ENABLE is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Not transmitted Verify Frame 1 - transmitted Verify Frame
17 RRSP	Received Respond Frame Set when a Respond mPacket is received. Received Respond Frame Set when a Respond mPacket is received. An interrupt can be generated for this event if FPEIE bit of MAC_INTERRUPT_ENABLE is set.

Table continues on the next page...

Field	Description
	Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Not received Respond Frame 1 - Received Respond Frame
16 RVER	Received Verify Frame Set when a Verify mPacket is received. Received Verify Frame Set when a Verify mPacket is received. An interrupt can be generated for this event if FPEIE bit of MAC_INTERRUPT_ENABLE is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Not received Verify Frame 1 - Received Verify Frame
15-4 —	Reserved
3 S1_SET_0	Synopsys Reserved, Must be set to "0". Synopsys Reserved, Must be set to "0". This field is reserved for Synopsys Internal use, and must always be set to "0" unless instructed by Synopsys. Setting to "1" might cause unexpected behavior in the IP.
2 SRSP	Send Respond mPacket When set indicates hardware to send a Respond mPacket. Send Respond mPacket When set indicates hardware to send a Respond mPacket. Reset by hardware after sending the Respond mPacket. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Send Respond mPacket is disabled 1 - Send Respond mPacket is enabled
1 SVER	Send Verify mPacket When set indicates hardware to send a verify mPacket. Send Verify mPacket When set indicates hardware to send a verify mPacket. Reset by hardware after sending the Verify mPacket. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Send Verify mPacket is disabled 1 - Send Verify mPacket is enabled
0 EFPE	Enable Tx Frame Preemption When set Frame Preemption Tx functionality is enabled. 0 - Tx Frame Preemption is disabled 1 - Tx Frame Preemption is enabled

11.7.6.1.46 32-bit Binary Rollover Equivalent Time (MAC_PRESN_TIME_NS)

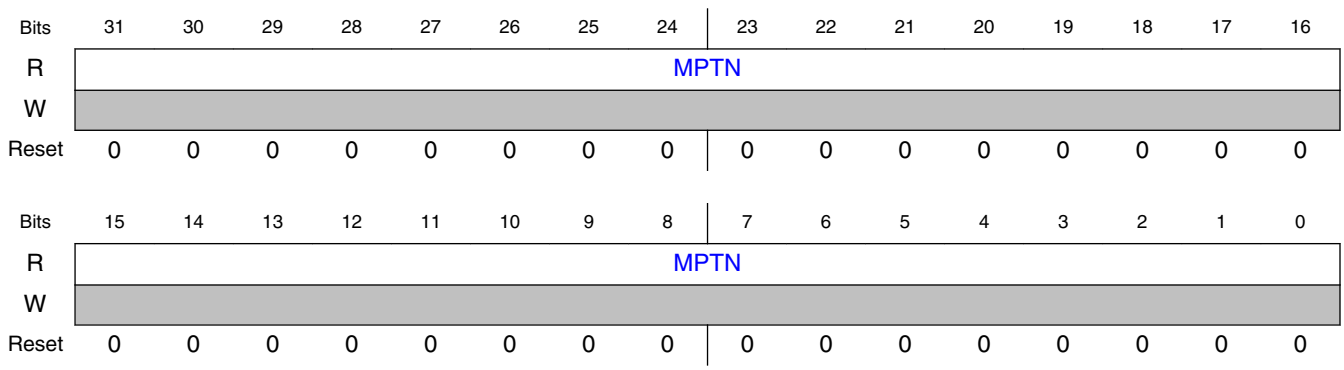
This register contains the 32-bit binary rollover equivalent time of the PTP System Time in ns

Exists when DWC_EQOS_FLEXI_PPS_OUT_EN is configured

11.7.6.1.46.1 Offset

Register	Offset
MAC_PRESN_TIME_NS	240h

11.7.6.1.46.2 Diagram



11.7.6.1.46.3 Fields

Field	Description
31-0 MPTN	MAC 1722 Presentation Time in ns These bits indicate the value of the 32-bit binary rollover equivalent time of the PTP System Time in ns

11.7.6.1.47 MAC 1722 Presentation Time (MAC_PRESN_TIME_UPDT)

This field holds the 32-bit value of MAC 1722 Presentation Time in ns, that should be added to the Current Presentation Time Counter value.

This field holds the 32-bit value of MAC 1722 Presentation Time in ns, that should be added to the Current Presentation Time Counter value. Init happens when TSINIT is set, and update happens when the TSUPDT bit is set (TSINIT and TSINIT defined in TIMESTAMP_CONTROL register).

Exists when DWC_EQOS_FLEXI_PPS_OUT_EN is configured

11.7.6.1.47.1 Offset

Register	Offset
MAC_PRESN_TIME_UPDT	244h

11.7.6.1.47.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MPTU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MPTU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.47.3 Fields

Field	Description
31-0 MPTU	MAC 1722 Presentation Time Update This field holds the init value or the update value for the presentation time. MAC 1722 Presentation Time Update This field holds the init value or the update value for the presentation time. When used for update, this field holds the 32-bit value in ns, that should be added to the Current Presentation Time Counter value. Init happens when TSINIT is set, and update happens when the TSUPDT bit is set (TSINIT and TSINIT defined in MAC_TIMESTAMP_CONTROL register). When ADDSUB field of MAC_SYSTEM_TIME_NANOSECONDS_UPDATE is set, this value is directly used for subtraction

11.7.6.1.48 MAC Address0 High (MAC_ADDRESS0_HIGH)

The MAC Address0 High register holds the upper 16 bits of the first 6-byte MAC address of the station.

The MAC Address0 High register holds the upper 16 bits of the first 6-byte MAC address of the station. The first DA byte that is received on the (G)MII interface corresponds to the LS byte (Bits [7:0]) of the MAC Address Low register. For example, if

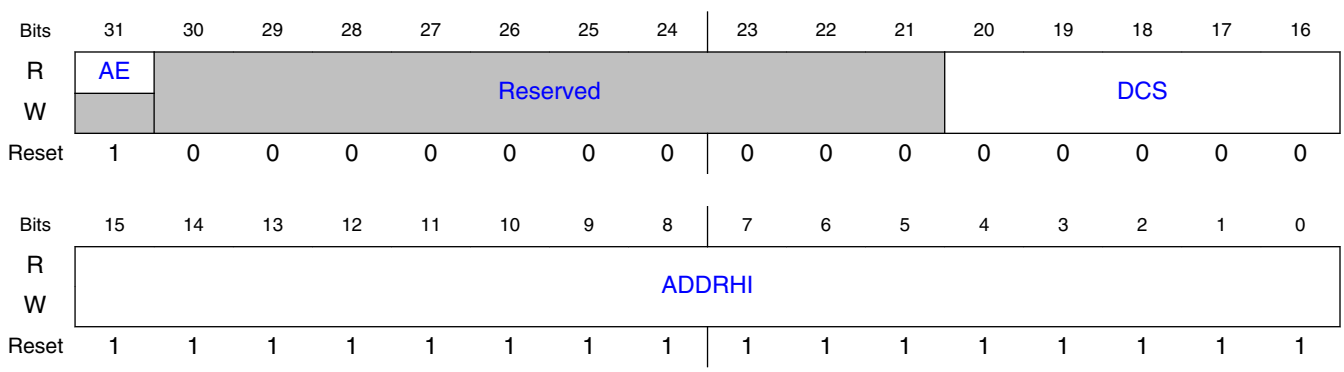
0x112233445566 is received (0x11 in lane 0 of the first column) on the (G)MII as the destination address, then the MacAddress0 Register [47:0] is compared with 0x665544332211.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, then the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address0 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.48.1 Offset

Register	Offset
MAC_ADDRESS0_HIGH	300h

11.7.6.1.48.2 Diagram



11.7.6.1.48.3 Fields

Field	Description
31 AE	Address Enable This bit is always set to 1. 0 - INVALID : This bit must be always set to 1 1 - This bit is always set to 1
30-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address0 content is routed. DMA Channel Select

Table continues on the next page...

Field	Description
	If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address0 content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address0 content is routed.
15-0 ADDRHI	MAC Address0[47:32] This field contains the upper 16 bits [47:32] of the first 6-byte MAC address. MAC Address0[47:32] This field contains the upper 16 bits [47:32] of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.

11.7.6.1.49 MAC Address0 Low (MAC_ADDRESS0_LOW)

The MAC Address0 Low register holds the lower 32 bits of the 6-byte first MAC address of the station.

11.7.6.1.49.1 Offset

Register	Offset
MAC_ADDRESS0_LOW	304h

11.7.6.1.49.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.49.3 Fields

Field	Description
31-0	MAC Address0[31:0] This field contains the lower 32 bits of the first 6-byte MAC address.

Field	Description
ADDRLO	MAC Address0[31:0] This field contains the lower 32 bits of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.

11.7.6.1.50 MAC Address1 High (MAC_ADDRESS1_HIGH)

The MAC Address1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.50.1 Offset

Register	Offset
MAC_ADDRESS1_HIGH	308h

11.7.6.1.50.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.50.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.51 MAC Address1 Low (MAC_ADDRESS1_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.51.1 Offset

Register	Offset
MAC_ADDRESS1_LOW	30Ch

11.7.6.1.51.2 Diagram

11.7.6.1.51.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.52 MAC Address2 High (MAC_ADDRESS2_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.52.1 Offset

Register	Offset
MAC_ADDRESS2_HIGH	310h

11.7.6.1.52.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.52.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet.

Table continues on the next page...

Field	Description
	0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.53 MAC Address2 Low (MAC_ADDRESS2_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.53.1 Offset

Register	Offset
MAC_ADDRESS2_LOW	314h

11.7.6.1.53.2 Diagram

11.7.6.1.53.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.54 MAC Address3 High (MAC_ADDRESS3_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.54.1 Offset

Register	Offset
MAC_ADDRESS3_HIGH	318h

11.7.6.1.54.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.54.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21	Reserved

Table continues on the next page...

Field	Description
—	
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.55 MAC Address3 Low (MAC_ADDRESS3_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.55.1 Offset

Register	Offset
MAC_ADDRESS3_LOW	31Ch

11.7.6.1.55.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.55.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.56 MAC Address4 High (MAC_ADDRESS4_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.56.1 Offset

Register	Offset
MAC_ADDRESS4_HIGH	320h

11.7.6.1.56.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.56.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.57 MAC Address4 Low (MAC_ADDRESS4_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.57.1 Offset

Register	Offset
MAC_ADDRESS4_LOW	324h

11.7.6.1.57.2 Diagram

11.7.6.1.57.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.58 MAC Address5 High (MAC_ADDRESS5_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.58.1 Offset

Register	Offset
MAC_ADDRESS5_HIGH	328h

11.7.6.1.58.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.58.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet.

Table continues on the next page...

Field	Description
	0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.59 MAC Address5 Low (MAC_ADDRESS5_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.59.1 Offset

Register	Offset
MAC_ADDRESS5_LOW	32Ch

11.7.6.1.59.2 Diagram

11.7.6.1.59.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.60 MAC Address6 High (MAC_ADDRESS6_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

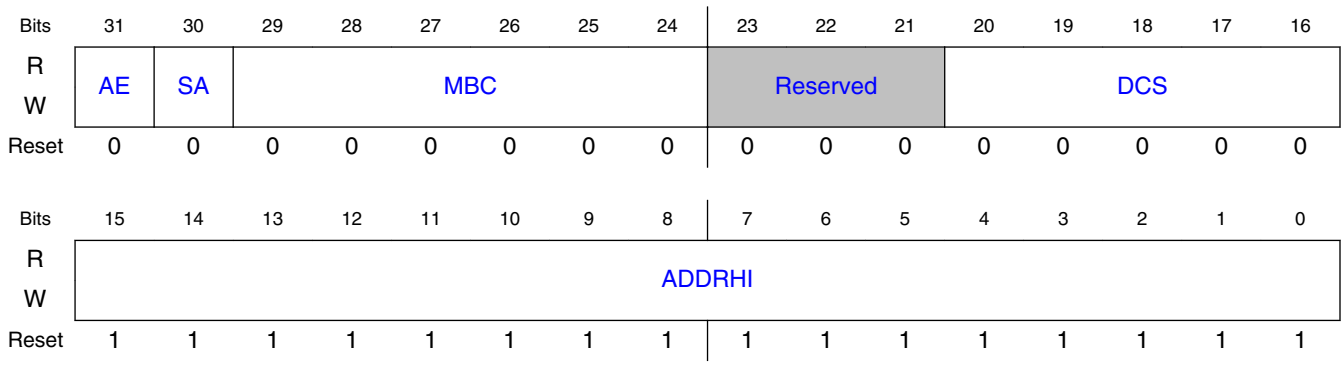
The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.60.1 Offset

Register	Offset
MAC_ADDRESS6_HIGH	330h

11.7.6.1.60.2 Diagram



11.7.6.1.60.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21	Reserved

Table continues on the next page...

Field	Description
—	
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.61 MAC Address6 Low (MAC_ADDRESS6_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.61.1 Offset

Register	Offset
MAC_ADDRESS6_LOW	334h

11.7.6.1.61.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.61.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.62 MAC Address7 High (MAC_ADDRESS7_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.62.1 Offset

Register	Offset
MAC_ADDRESS7_HIGH	338h

11.7.6.1.62.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.62.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.63 MAC Address7 Low (MAC_ADDRESS7_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.63.1 Offset

Register	Offset
MAC_ADDRESS7_LOW	33Ch

11.7.6.1.63.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.63.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.64 MAC Address8 High (MAC_ADDRESS8_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.64.1 Offset

Register	Offset
MAC_ADDRESS8_HIGH	340h

11.7.6.1.64.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.64.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet.

Table continues on the next page...

Field	Description
	0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.65 MAC Address8 Low (MAC_ADDRESS8_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.65.1 Offset

Register	Offset
MAC_ADDRESS8_LOW	344h

11.7.6.1.65.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.65.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.66 MAC Address9 High (MAC_ADDRESS9_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

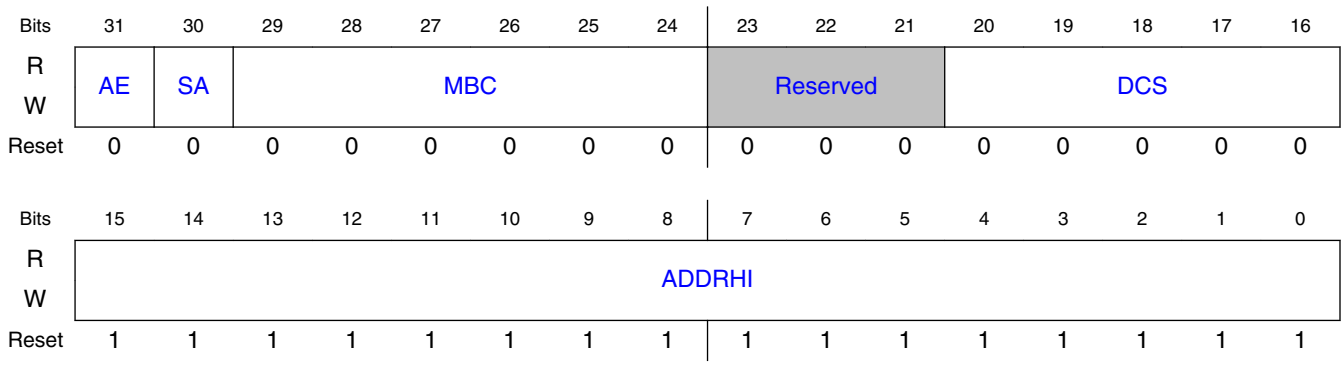
The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.66.1 Offset

Register	Offset
MAC_ADDRESS9_HIGH	348h

11.7.6.1.66.2 Diagram



11.7.6.1.66.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21	Reserved

Table continues on the next page...

Field	Description
—	
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.67 MAC Address9 Low (MAC_ADDRESS9_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.67.1 Offset

Register	Offset
MAC_ADDRESS9_LOW	34Ch

11.7.6.1.67.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.67.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.68 MAC Address10 High (MAC_ADDRESS10_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.68.1 Offset

Register	Offset
MAC_ADDRESS10_HIGH	350h

11.7.6.1.68.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.68.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.69 MAC Address10 Low (MAC_ADDRESS10_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.69.1 Offset

Register	Offset
MAC_ADDRESS10_LOW	354h

11.7.6.1.69.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.69.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.70 MAC Address11 High (MAC_ADDRESS11_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.70.1 Offset

Register	Offset
MAC_ADDRESS11_HIGH	358h

11.7.6.1.70.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.70.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address

Table continues on the next page...

Field	Description
	When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.71 MAC Address11 Low (MAC_ADDRESS11_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.71.1 Offset

Register	Offset
MAC_ADDRESS11_LOW	35Ch

11.7.6.1.71.2 Diagram

11.7.6.1.71.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.72 MAC Address12 High (MAC_ADDRESS12_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.72.1 Offset

Register	Offset
MAC_ADDRESS12_HIGH	360h

11.7.6.1.72.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.72.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0]

Table continues on the next page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.73 MAC Address12 Low (MAC_ADDRESS12_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.73.1 Offset

Register	Offset
MAC_ADDRESS12_LOW	364h

11.7.6.1.73.2 Diagram

11.7.6.1.73.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.74 MAC Address13 High (MAC_ADDRESS13_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

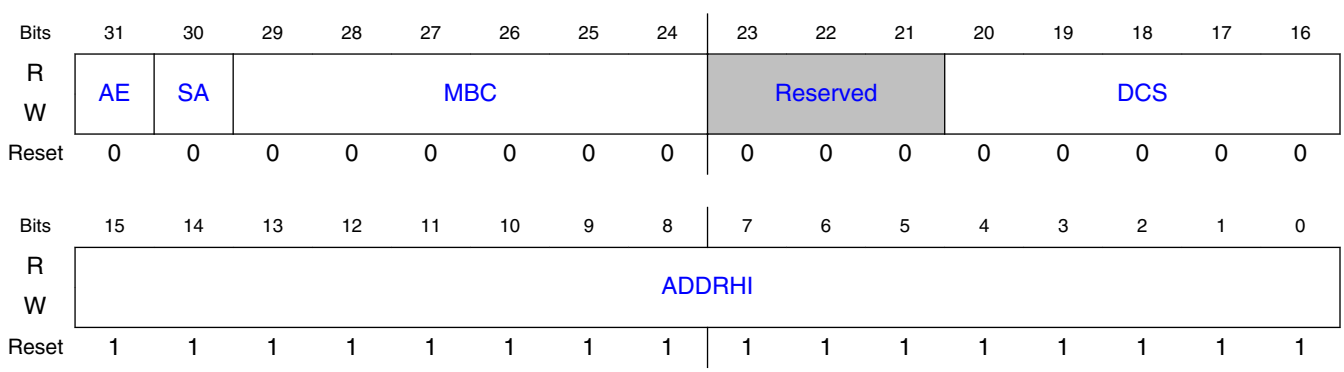
The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.74.1 Offset

Register	Offset
MAC_ADDRESS13_HIGH	368h

11.7.6.1.74.2 Diagram



11.7.6.1.74.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.75 MAC Address13 Low (MAC_ADDRESS13_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.75.1 Offset

Register	Offset
MAC_ADDRESS13_LOW	36Ch

11.7.6.1.75.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.75.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.76 MAC Address14 High (MAC_ADDRESS14_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

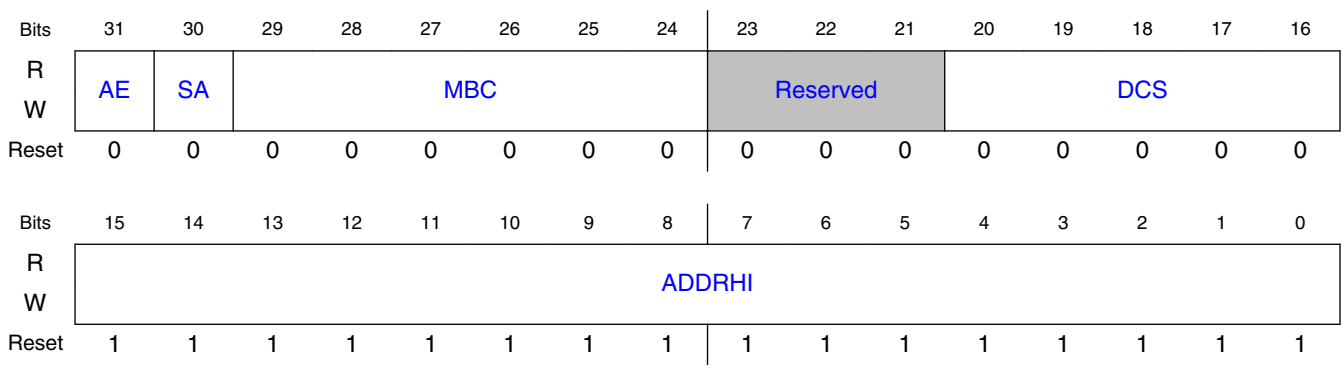
The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.76.1 Offset

Register	Offset
MAC_ADDRESS14_HIGH	370h

11.7.6.1.76.2 Diagram



11.7.6.1.76.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address

Table continues on the next page...

Field	Description
	When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.77 MAC Address14 Low (MAC_ADDRESS14_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.77.1 Offset

Register	Offset
MAC_ADDRESS14_LOW	374h

11.7.6.1.77.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.77.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.78 MAC Address15 High (MAC_ADDRESS15_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.78.1 Offset

Register	Offset
MAC_ADDRESS15_HIGH	378h

11.7.6.1.78.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.78.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0]

Table continues on the next page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.79 MAC Address15 Low (MAC_ADDRESS15_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.79.1 Offset

Register	Offset
MAC_ADDRESS15_LOW	37Ch

11.7.6.1.79.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.79.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.80 MAC Address16 High (MAC_ADDRESS16_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.80.1 Offset

Register	Offset
MAC_ADDRESS16_HIGH	380h

11.7.6.1.80.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.80.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.81 MAC Address16 Low (MAC_ADDRESS16_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.81.1 Offset

Register	Offset
MAC_ADDRESS16_LOW	384h

11.7.6.1.81.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.81.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.82 MAC Address17 High (MAC_ADDRESS17_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

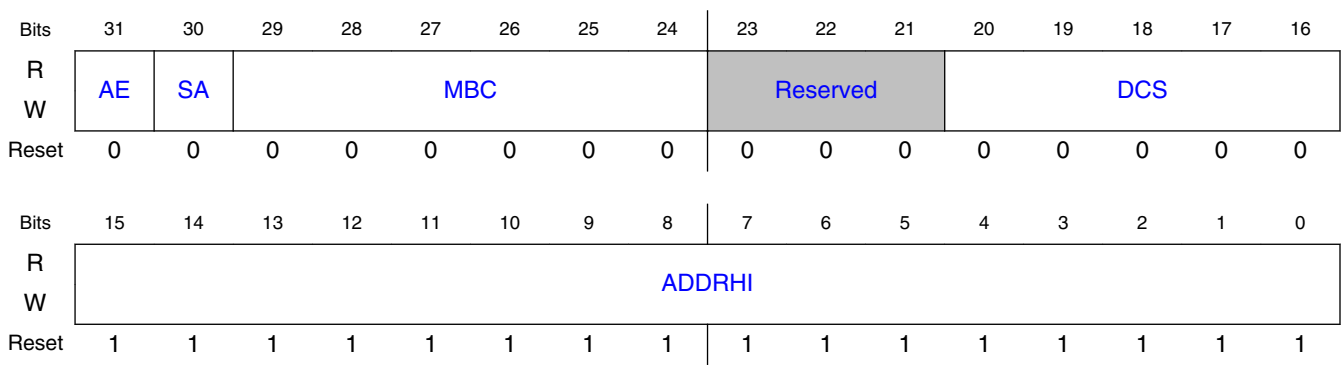
The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.82.1 Offset

Register	Offset
MAC_ADDRESS17_HIGH	388h

11.7.6.1.82.2 Diagram



11.7.6.1.82.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address

Table continues on the next page...

Field	Description
	When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.83 MAC Address17 Low (MAC_ADDRESS17_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.83.1 Offset

Register	Offset
MAC_ADDRESS17_LOW	38Ch

11.7.6.1.83.2 Diagram

11.7.6.1.83.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.84 MAC Address18 High (MAC_ADDRESS18_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.84.1 Offset

Register	Offset
MAC_ADDRESS18_HIGH	390h

11.7.6.1.84.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.84.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0]

Table continues on the next page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.85 MAC Address18 Low (MAC_ADDRESS18_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.85.1 Offset

Register	Offset
MAC_ADDRESS18_LOW	394h

11.7.6.1.85.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.85.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.86 MAC Address19 High (MAC_ADDRESS19_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.86.1 Offset

Register	Offset
MAC_ADDRESS19_HIGH	398h

11.7.6.1.86.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.86.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.87 MAC Address19 Low (MAC_ADDRESS19_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.87.1 Offset

Register	Offset
MAC_ADDRESS19_LOW	39Ch

11.7.6.1.87.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.87.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.88 MAC Address20 High (MAC_ADDRESS20_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.88.1 Offset

Register	Offset
MAC_ADDRESS20_HIGH	3A0h

11.7.6.1.88.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.88.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address

Table continues on the next page...

Field	Description
	When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.89 MAC Address20 Low (MAC_ADDRESS20_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.89.1 Offset

Register	Offset
MAC_ADDRESS20_LOW	3A4h

11.7.6.1.89.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.89.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.90 MAC Address21 High (MAC_ADDRESS21_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

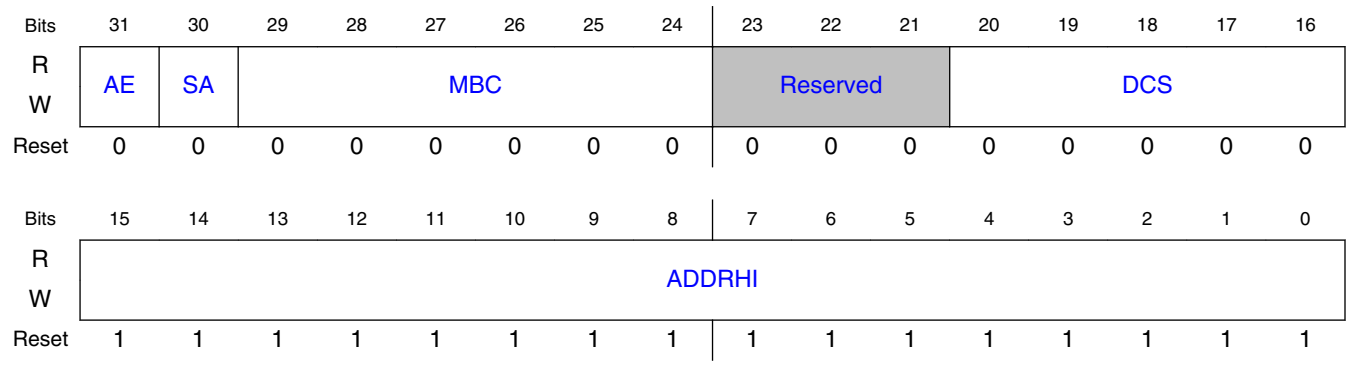
The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.90.1 Offset

Register	Offset
MAC_ADDRESS21_HIGH	3A8h

11.7.6.1.90.2 Diagram



11.7.6.1.90.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0]

Table continues on the next page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.91 MAC Address21 Low (MAC_ADDRESS21_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.91.1 Offset

Register	Offset
MAC_ADDRESS21_LOW	3ACh

11.7.6.1.91.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.91.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.92 MAC Address22 High (MAC_ADDRESS22_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.92.1 Offset

Register	Offset
MAC_ADDRESS22_HIGH	3B0h

11.7.6.1.92.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.92.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.93 MAC Address22 Low (MAC_ADDRESS22_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.93.1 Offset

Register	Offset
MAC_ADDRESS22_LOW	3B4h

11.7.6.1.93.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.93.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.94 MAC Address23 High (MAC_ADDRESS23_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.94.1 Offset

Register	Offset
MAC_ADDRESS23_HIGH	3B8h

11.7.6.1.94.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.94.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address

Table continues on the next page...

Field	Description
	When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.95 MAC Address23 Low (MAC_ADDRESS23_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.95.1 Offset

Register	Offset
MAC_ADDRESS23_LOW	3BCh

11.7.6.1.95.2 Diagram

11.7.6.1.95.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.96 MAC Address24 High (MAC_ADDRESS24_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.96.1 Offset

Register	Offset
MAC_ADDRESS24_HIGH	3C0h

11.7.6.1.96.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.96.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0]

Table continues on the next page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.97 MAC Address24 Low (MAC_ADDRESS24_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.97.1 Offset

Register	Offset
MAC_ADDRESS24_LOW	3C4h

11.7.6.1.97.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.97.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.98 MAC Address25 High (MAC_ADDRESS25_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

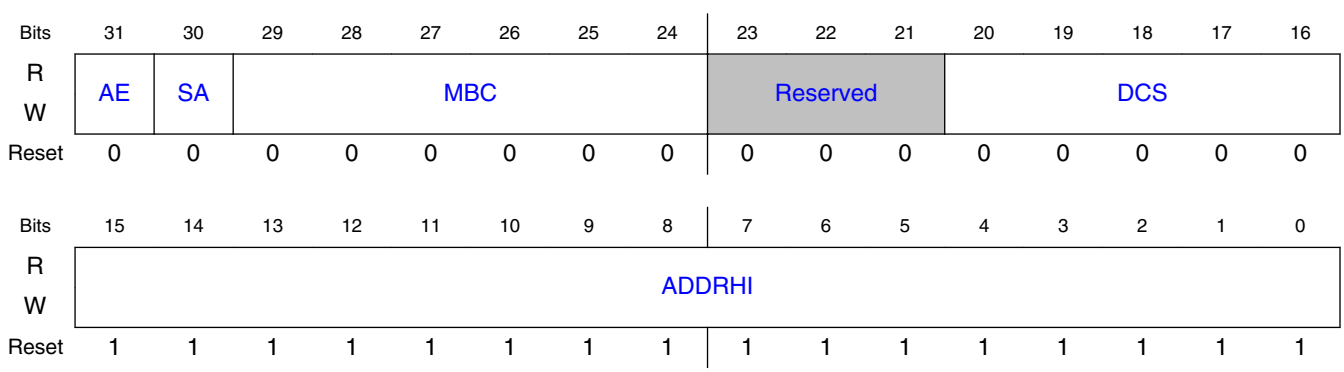
The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.98.1 Offset

Register	Offset
MAC_ADDRESS25_HIGH	3C8h

11.7.6.1.98.2 Diagram



11.7.6.1.98.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.99 MAC Address25 Low (MAC_ADDRESS25_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.99.1 Offset

Register	Offset
MAC_ADDRESS25_LOW	3CCh

11.7.6.1.99.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.99.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.100 MAC Address26 High (MAC_ADDRESS26_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.100.1 Offset

Register	Offset
MAC_ADDRESS26_HIGH	3D0h

11.7.6.1.100.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.100.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address

Table continues on the next page...

Field	Description
	When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.101 MAC Address26 Low (MAC_ADDRESS26_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.101.1 Offset

Register	Offset
MAC_ADDRESS26_LOW	3D4h

11.7.6.1.101.2 Diagram

11.7.6.1.101.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.102 MAC Address27 High (MAC_ADDRESS27_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.102.1 Offset

Register	Offset
MAC_ADDRESS27_HIGH	3D8h

11.7.6.1.102.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.102.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0]

Table continues on the next page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.103 MAC Address27 Low (MAC_ADDRESS27_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.103.1 Offset

Register	Offset
MAC_ADDRESS27_LOW	3DCh

11.7.6.1.103.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.103.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.104 MAC Address28 High (MAC_ADDRESS28_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

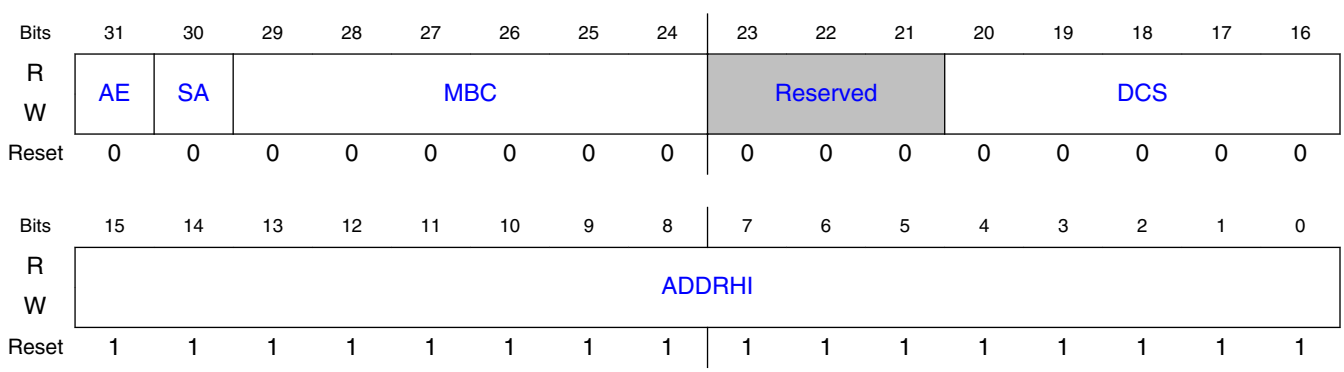
The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.104.1 Offset

Register	Offset
MAC_ADDRESS28_HIGH	3E0h

11.7.6.1.104.2 Diagram



11.7.6.1.104.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.105 MAC Address28 Low (MAC_ADDRESS28_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.105.1 Offset

Register	Offset
MAC_ADDRESS28_LOW	3E4h

11.7.6.1.105.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.105.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.106 MAC Address29 High (MAC_ADDRESS29_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.106.1 Offset

Register	Offset
MAC_ADDRESS29_HIGH	3E8h

11.7.6.1.106.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.106.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address

Table continues on the next page...

Field	Description
	When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.107 MAC Address29 Low (MAC_ADDRESS29_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.107.1 Offset

Register	Offset
MAC_ADDRESS29_LOW	3ECh

11.7.6.1.107.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.107.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.108 MAC Address30 High (MAC_ADDRESS30_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

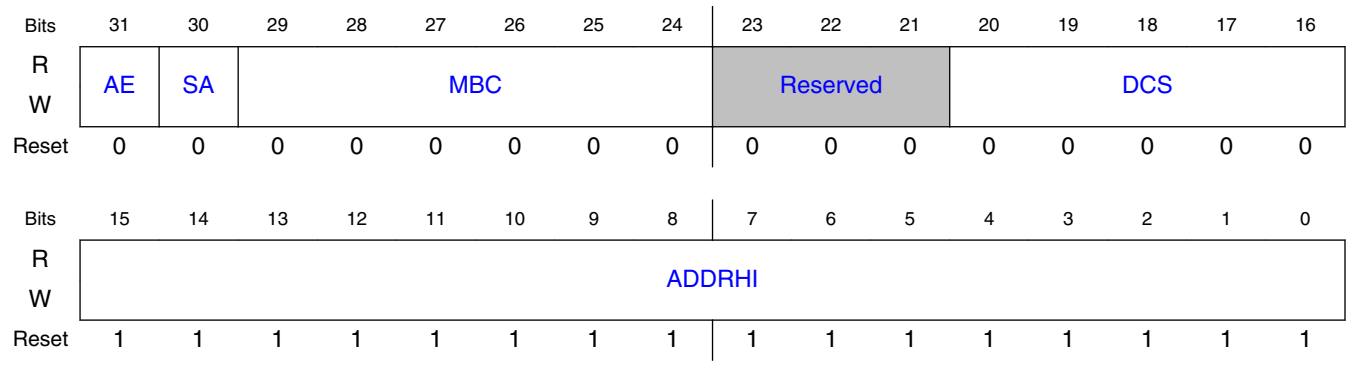
The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.108.1 Offset

Register	Offset
MAC_ADDRESS30_HIGH	3F0h

11.7.6.1.108.2 Diagram



11.7.6.1.108.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0]

Table continues on the next page...

Field	Description
	You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.109 MAC Address30 Low (MAC_ADDRESS30_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.109.1 Offset

Register	Offset
MAC_ADDRESS30_LOW	3F4h

11.7.6.1.109.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.109.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.110 MAC Address31 High (MAC_ADDRESS31_HIGH)

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

11.7.6.1.110.1 Offset

Register	Offset
MAC_ADDRESS31_HIGH	3F8h

11.7.6.1.110.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.110.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0 - Compare with Destination Address 1 - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS${i}_High[15:8] - Bit 28: ADDRESS${i}_High[7:0] - Bit 27: ADDRESS${i}_Low[31:24] - .. - Bit 24: ADDRESS${i}_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

11.7.6.1.111 MAC Address31 Low (MAC_ADDRESS31_LOW)

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

11.7.6.1.111.1 Offset

Register	Offset
MAC_ADDRESS31_LOW	3FCh

11.7.6.1.111.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.111.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

11.7.6.1.112 MAC Address32 High (MAC_ADDRESS32_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.112.1 Offset

Register	Offset
MAC_ADDRESS32_HIGH	400h

11.7.6.1.112.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.112.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.113 MAC Address32 Low (MAC_ADDRESS32_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.113.1 Offset

Register	Offset
MAC_ADDRESS32_LOW	404h

11.7.6.1.113.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.113.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.114 MAC Address33 High (MAC_ADDRESS33_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.114.1 Offset

Register	Offset
MAC_ADDRESS33_HIGH	408h

11.7.6.1.114.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.114.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.115 MAC Address33 Low (MAC_ADDRESS33_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.115.1 Offset

Register	Offset
MAC_ADDRESS33_LOW	40Ch

11.7.6.1.115.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.115.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.116 MAC Address34 High (MAC_ADDRESS34_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.116.1 Offset

Register	Offset
MAC_ADDRESS34_HIGH	410h

11.7.6.1.116.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.116.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.117 MAC Address34 Low (MAC_ADDRESS34_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.117.1 Offset

Register	Offset
MAC_ADDRESS34_LOW	414h

11.7.6.1.117.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.117.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.118 MAC Address35 High (MAC_ADDRESS35_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.118.1 Offset

Register	Offset
MAC_ADDRESS35_HIGH	418h

11.7.6.1.118.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.118.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.119 MAC Address35 Low (MAC_ADDRESS35_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.119.1 Offset

Register	Offset
MAC_ADDRESS35_LOW	41Ch

11.7.6.1.119.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.119.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.120 MAC Address36 High (MAC_ADDRESS36_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.120.1 Offset

Register	Offset
MAC_ADDRESS36_HIGH	420h

11.7.6.1.120.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.120.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.121 MAC Address36 Low (MAC_ADDRESS36_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.121.1 Offset

Register	Offset
MAC_ADDRESS36_LOW	424h

11.7.6.1.121.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.121.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.122 MAC Address37 High (MAC_ADDRESS37_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.122.1 Offset

Register	Offset
MAC_ADDRESS37_HIGH	428h

11.7.6.1.122.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.122.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.123 MAC Address37 Low (MAC_ADDRESS37_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.123.1 Offset

Register	Offset
MAC_ADDRESS37_LOW	42Ch

11.7.6.1.123.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.123.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.124 MAC Address38 High (MAC_ADDRESS38_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.124.1 Offset

Register	Offset
MAC_ADDRESS38_HIGH	430h

11.7.6.1.124.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.124.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.125 MAC Address38 Low (MAC_ADDRESS38_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.125.1 Offset

Register	Offset
MAC_ADDRESS38_LOW	434h

11.7.6.1.125.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.125.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.126 MAC Address39 High (MAC_ADDRESS39_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.126.1 Offset

Register	Offset
MAC_ADDRESS39_HIGH	438h

11.7.6.1.126.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.126.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.127 MAC Address39 Low (MAC_ADDRESS39_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.127.1 Offset

Register	Offset
MAC_ADDRESS39_LOW	43Ch

11.7.6.1.127.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.127.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.128 MAC Address40 High (MAC_ADDRESS40_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.128.1 Offset

Register	Offset
MAC_ADDRESS40_HIGH	440h

11.7.6.1.128.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.128.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.129 MAC Address40 Low (MAC_ADDRESS40_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.129.1 Offset

Register	Offset
MAC_ADDRESS40_LOW	444h

11.7.6.1.129.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.129.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.130 MAC Address41 High (MAC_ADDRESS41_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.130.1 Offset

Register	Offset
MAC_ADDRESS41_HIGH	448h

11.7.6.1.130.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.130.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.131 MAC Address41 Low (MAC_ADDRESS41_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.131.1 Offset

Register	Offset
MAC_ADDRESS41_LOW	44Ch

11.7.6.1.131.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.131.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.132 MAC Address42 High (MAC_ADDRESS42_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.132.1 Offset

Register	Offset
MAC_ADDRESS42_HIGH	450h

11.7.6.1.132.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.132.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.133 MAC Address42 Low (MAC_ADDRESS42_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.133.1 Offset

Register	Offset
MAC_ADDRESS42_LOW	454h

11.7.6.1.133.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.133.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.134 MAC Address43 High (MAC_ADDRESS43_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.134.1 Offset

Register	Offset
MAC_ADDRESS43_HIGH	458h

11.7.6.1.134.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.134.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.135 MAC Address43 Low (MAC_ADDRESS43_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.135.1 Offset

Register	Offset
MAC_ADDRESS43_LOW	45Ch

11.7.6.1.135.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.135.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.136 MAC Address44 High (MAC_ADDRESS44_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.136.1 Offset

Register	Offset
MAC_ADDRESS44_HIGH	460h

11.7.6.1.136.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.136.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.137 MAC Address44 Low (MAC_ADDRESS44_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.137.1 Offset

Register	Offset
MAC_ADDRESS44_LOW	464h

11.7.6.1.137.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.137.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.138 MAC Address45 High (MAC_ADDRESS45_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.138.1 Offset

Register	Offset
MAC_ADDRESS45_HIGH	468h

11.7.6.1.138.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.138.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.139 MAC Address45 Low (MAC_ADDRESS45_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.139.1 Offset

Register	Offset
MAC_ADDRESS45_LOW	46Ch

11.7.6.1.139.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.139.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.140 MAC Address46 High (MAC_ADDRESS46_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.140.1 Offset

Register	Offset
MAC_ADDRESS46_HIGH	470h

11.7.6.1.140.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.140.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.141 MAC Address46 Low (MAC_ADDRESS46_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.141.1 Offset

Register	Offset
MAC_ADDRESS46_LOW	474h

11.7.6.1.141.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.141.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.142 MAC Address47 High (MAC_ADDRESS47_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.142.1 Offset

Register	Offset
MAC_ADDRESS47_HIGH	478h

11.7.6.1.142.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.142.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.143 MAC Address47 Low (MAC_ADDRESS47_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.143.1 Offset

Register	Offset
MAC_ADDRESS47_LOW	47Ch

11.7.6.1.143.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.143.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.144 MAC Address48 High (MAC_ADDRESS48_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.144.1 Offset

Register	Offset
MAC_ADDRESS48_HIGH	480h

11.7.6.1.144.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.144.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.145 MAC Address48 Low (MAC_ADDRESS48_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.145.1 Offset

Register	Offset
MAC_ADDRESS48_LOW	484h

11.7.6.1.145.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.145.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.146 MAC Address49 High (MAC_ADDRESS49_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.146.1 Offset

Register	Offset
MAC_ADDRESS49_HIGH	488h

11.7.6.1.146.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.146.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.147 MAC Address49 Low (MAC_ADDRESS49_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.147.1 Offset

Register	Offset
MAC_ADDRESS49_LOW	48Ch

11.7.6.1.147.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.147.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.148 MAC Address50 High (MAC_ADDRESS50_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.148.1 Offset

Register	Offset
MAC_ADDRESS50_HIGH	490h

11.7.6.1.148.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.148.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.149 MAC Address50 Low (MAC_ADDRESS50_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.149.1 Offset

Register	Offset
MAC_ADDRESS50_LOW	494h

11.7.6.1.149.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.149.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.150 MAC Address51 High (MAC_ADDRESS51_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.150.1 Offset

Register	Offset
MAC_ADDRESS51_HIGH	498h

11.7.6.1.150.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.150.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.151 MAC Address51 Low (MAC_ADDRESS51_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.151.1 Offset

Register	Offset
MAC_ADDRESS51_LOW	49Ch

11.7.6.1.151.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.151.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.152 MAC Address52 High (MAC_ADDRESS52_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.152.1 Offset

Register	Offset
MAC_ADDRESS52_HIGH	4A0h

11.7.6.1.152.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.152.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.153 MAC Address52 Low (MAC_ADDRESS52_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.153.1 Offset

Register	Offset
MAC_ADDRESS52_LOW	4A4h

11.7.6.1.153.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.153.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.154 MAC Address53 High (MAC_ADDRESS53_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.154.1 Offset

Register	Offset
MAC_ADDRESS53_HIGH	4A8h

11.7.6.1.154.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.154.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.155 MAC Address53 Low (MAC_ADDRESS53_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.155.1 Offset

Register	Offset
MAC_ADDRESS53_LOW	4ACh

11.7.6.1.155.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.155.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.156 MAC Address54 High (MAC_ADDRESS54_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.156.1 Offset

Register	Offset
MAC_ADDRESS54_HIGH	4B0h

11.7.6.1.156.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.156.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.157 MAC Address54 Low (MAC_ADDRESS54_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.157.1 Offset

Register	Offset
MAC_ADDRESS54_LOW	4B4h

11.7.6.1.157.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.157.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.158 MAC Address55 High (MAC_ADDRESS55_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.158.1 Offset

Register	Offset
MAC_ADDRESS55_HIGH	4B8h

11.7.6.1.158.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.158.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.159 MAC Address55 Low (MAC_ADDRESS55_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.159.1 Offset

Register	Offset
MAC_ADDRESS55_LOW	4BCh

11.7.6.1.159.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.159.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.160 MAC Address56 High (MAC_ADDRESS56_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.160.1 Offset

Register	Offset
MAC_ADDRESS56_HIGH	4C0h

11.7.6.1.160.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.160.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.161 MAC Address56 Low (MAC_ADDRESS56_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.161.1 Offset

Register	Offset
MAC_ADDRESS56_LOW	4C4h

11.7.6.1.161.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.161.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.162 MAC Address57 High (MAC_ADDRESS57_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.162.1 Offset

Register	Offset
MAC_ADDRESS57_HIGH	4C8h

11.7.6.1.162.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.162.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.163 MAC Address57 Low (MAC_ADDRESS57_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.163.1 Offset

Register	Offset
MAC_ADDRESS57_LOW	4CCh

11.7.6.1.163.2 Diagram

11.7.6.1.163.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.164 MAC Address58 High (MAC_ADDRESS58_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.164.1 Offset

Register	Offset
MAC_ADDRESS58_HIGH	4D0h

11.7.6.1.164.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.164.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.165 MAC Address58 Low (MAC_ADDRESS58_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.165.1 Offset

Register	Offset
MAC_ADDRESS58_LOW	4D4h

11.7.6.1.165.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.165.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.166 MAC Address59 High (MAC_ADDRESS59_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.166.1 Offset

Register	Offset
MAC_ADDRESS59_HIGH	4D8h

11.7.6.1.166.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.166.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.167 MAC Address59 Low (MAC_ADDRESS59_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.167.1 Offset

Register	Offset
MAC_ADDRESS59_LOW	4DCh

11.7.6.1.167.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.167.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.168 MAC Address60 High (MAC_ADDRESS60_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.168.1 Offset

Register	Offset
MAC_ADDRESS60_HIGH	4E0h

11.7.6.1.168.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.168.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.169 MAC Address60 Low (MAC_ADDRESS60_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.169.1 Offset

Register	Offset
MAC_ADDRESS60_LOW	4E4h

11.7.6.1.169.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.169.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.170 MAC Address61 High (MAC_ADDRESS61_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.170.1 Offset

Register	Offset
MAC_ADDRESS61_HIGH	4E8h

11.7.6.1.170.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.170.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.171 MAC Address61 Low (MAC_ADDRESS61_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.171.1 Offset

Register	Offset
MAC_ADDRESS61_LOW	4ECh

11.7.6.1.171.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.171.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.172 MAC Address62 High (MAC_ADDRESS62_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.172.1 Offset

Register	Offset
MAC_ADDRESS62_HIGH	4F0h

11.7.6.1.172.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.172.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.173 MAC Address62 Low (MAC_ADDRESS62_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.173.1 Offset

Register	Offset
MAC_ADDRESS62_LOW	4F4h

11.7.6.1.173.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.173.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

11.7.6.1.174 MAC Address63 High (MAC_ADDRESS63_HIGH)

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

11.7.6.1.174.1 Offset

Register	Offset
MAC_ADDRESS63_HIGH	4F8h

11.7.6.1.174.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.174.3 Fields

Field	Description
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0 - Address is ignored 1 - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

11.7.6.1.175 MAC Address63 Low (MAC_ADDRESS63_LOW)

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

11.7.6.1.175.1 Offset

Register	Offset
MAC_ADDRESS63_LOW	4FCh

11.7.6.1.175.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.7.6.1.175.3 Fields

Field	Description
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

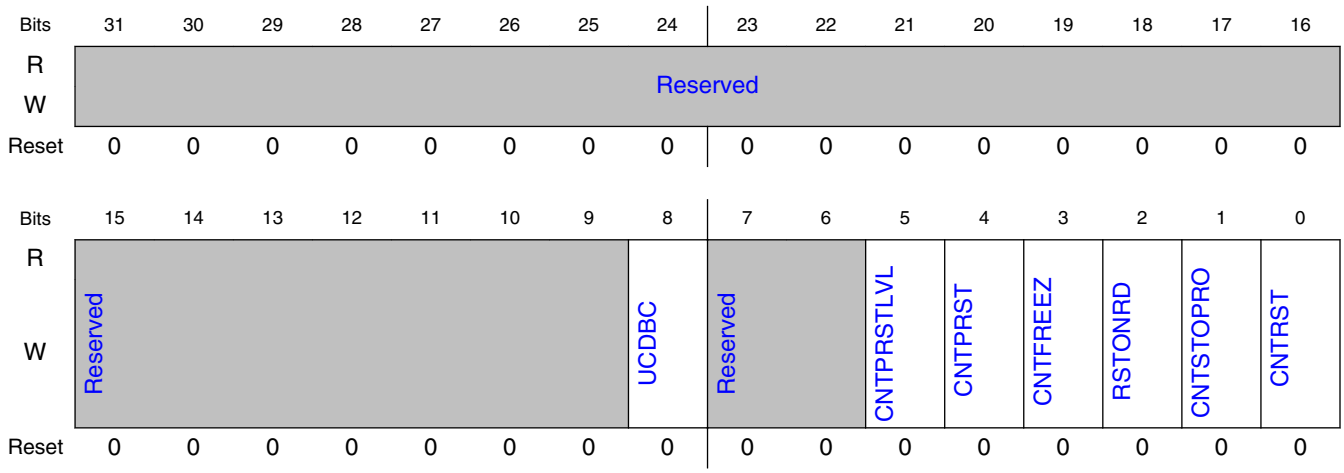
11.7.6.1.176 MMC Control (MAC_MMC_CONTROL)

This register establishes the operating mode of MMC.

11.7.6.1.176.1 Offset

Register	Offset
MAC_MMC_CONTROL	700h

11.7.6.1.176.2 Diagram



11.7.6.1.176.3 Fields

Field	Description
31-9 —	Reserved
8 UCDBC	Update MMC Counters for Dropped Broadcast Packets Note: The CNTRST bit has a higher priority than the CNTPRST bit. Update MMC Counters for Dropped Broadcast Packets Note: The CNTRST bit has a higher priority than the CNTPRST bit. Therefore, when the software tries to set both bits in the same write cycle, all counters are cleared and the CNTPRST bit is not set. When set, the MAC updates all related MMC Counters for Broadcast packets that are dropped because of the setting of the DBF bit of MAC_Packet_Filter register. When reset, the MMC Counters are not updated for dropped Broadcast packets. 0 - Update MMC Counters for Dropped Broadcast Packets is disabled 1 - Update MMC Counters for Dropped Broadcast Packets is enabled
7-6 —	Reserved
5 CNTPRSTLVL	Full-Half Preset When this bit is low and the CNTPRST bit is set, all MMC counters get preset to almost-half value. Full-Half Preset

Table continues on the next page...

Field	Description
	When this bit is low and the CNTPRST bit is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (Half 2KBytes) and all packet-counters gets preset to 0x7FFF_FFF0 (Half 16). When this bit is high and the CNTPRST bit is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (Full 2KBytes) and all packet-counters gets preset to 0xFFFF_FFF0 (Full 16). For 16-bit counters, the almost-half preset values are 0x7800 and 0x7FF0 for the respective octet and packet counters. Similarly, the almost-full preset values for the 16-bit counters are 0xF800 and 0xFFF0. 0 - Full-Half Preset is disabled 1 - Full-Half Preset is enabled
4 CNTPRST	Counters Preset When this bit is set, all counters are initialized or preset to almost full or almost half according to the CNTPRSTLVL bit. Counters Preset When this bit is set, all counters are initialized or preset to almost full or almost half according to the CNTPRSTLVL bit. This bit is cleared automatically after 1 clock cycle. This bit, along with the CNTPRSTLVL bit, is useful for debugging and testing the assertion of interrupts because of MMC counter becoming half-full or full. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0 - Counters Preset is disabled 1 - Counters Preset is enabled
3 CNTFREEZ	MMC Counter Freeze When this bit is set, it freezes all MMC counters to their current value. MMC Counter Freeze When this bit is set, it freezes all MMC counters to their current value. Until this bit is reset to 0, no MMC counter is updated because of any transmitted or received packet. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode. 0 - MMC Counter Freeze is disabled 1 - MMC Counter Freeze is enabled
2 RSTONRD	Reset on Read When this bit is set, the MMC counters are reset to zero after Read (self-clearing after reset). Reset on Read When this bit is set, the MMC counters are reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (Bits[7:0]) is read. 0 - Reset on Read is disabled 1 - Reset on Read is enabled
1 CNTSTOPRO	Counter Stop Rollover When this bit is set, the counter does not roll over to zero after reaching the maximum value. 0 - Counter Stop Rollover is disabled 1 - Counter Stop Rollover is enabled
0 CNTRST	Counters Reset When this bit is set, all counters are reset. Counters Reset When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0 - Counters are not reset

Field	Description
	1 - All counters are reset

11.7.6.1.177 MMC Rx Interrupt (MAC_MMC_RX_INTERRUPT)

This register maintains the interrupts generated from all Receive statistics counters.

This register maintains the interrupts generated from all Receive statistics counters.

The MMC Receive Interrupt register maintains the interrupts that are generated when the following occur:

- Receive statistic counters reach half of their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter).

- Receive statistic counters cross their maximum values (0xFFFF_FFFF for 32 bit counter and 0xFFFF for 16 bit counter).

When the Counter Stop Rollover is set, interrupts are set but the counter remains at all-ones. The MMC Receive Interrupt register

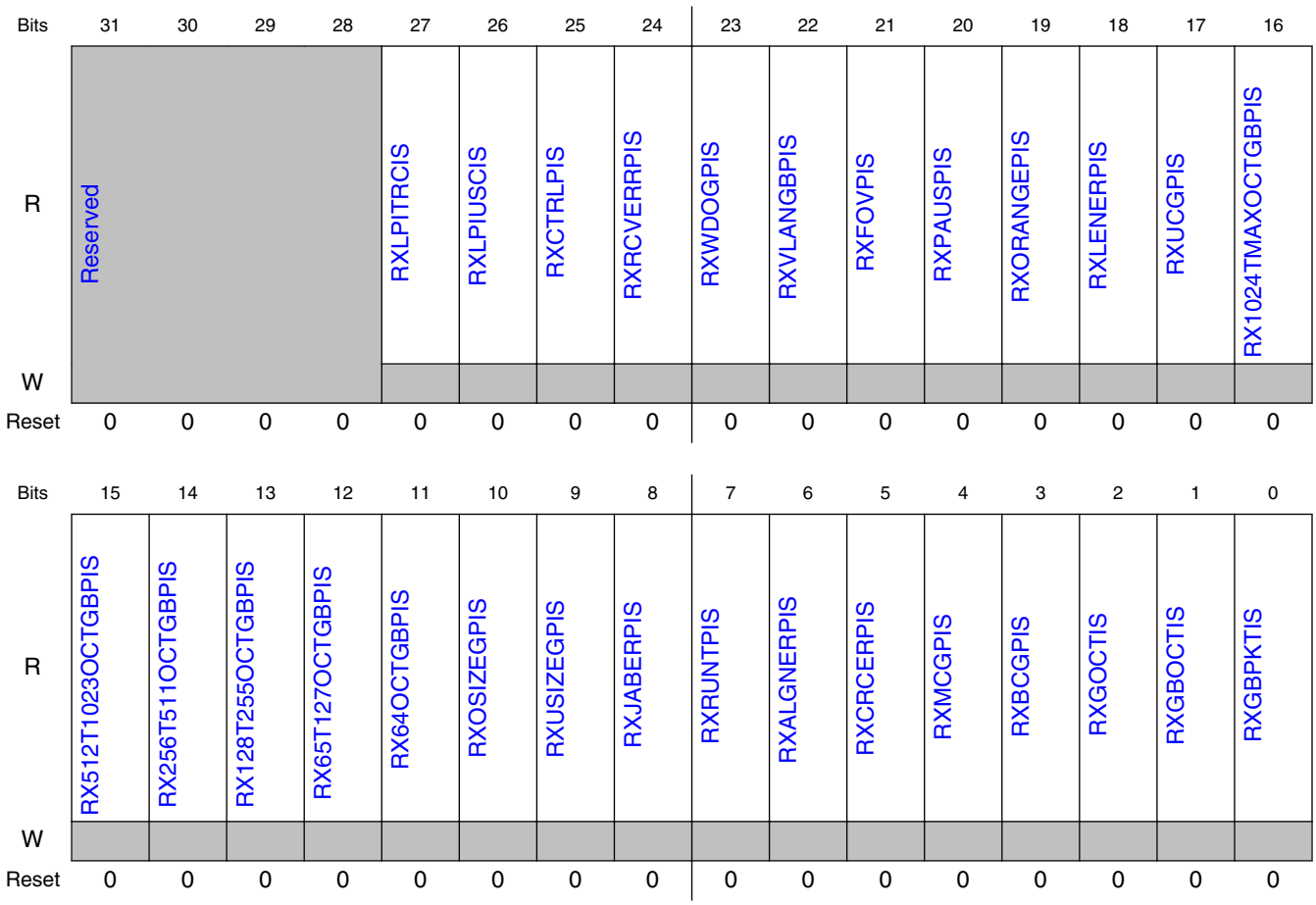
is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (Bits[7:0]) of the respective counter must be read to clear the interrupt bit.

Note: R_SS_RC means that this register bit is set internally, and it is cleared when the Counter register is read.

11.7.6.1.177.1 Offset

Register	Offset
MAC_MMC_RX_INTERRUPT	704h

11.7.6.1.177.2 Diagram



11.7.6.1.177.3 Fields

Field	Description
31-28 —	Reserved
27 RXLPITRCIS	MMC Receive LPI transition counter interrupt status This bit is set when the Rx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. MMC Receive LPI transition counter interrupt status This bit is set when the Rx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive LPI transition Counter Interrupt Status not detected 1 - MMC Receive LPI transition Counter Interrupt Status detected
26 RXLPIUSCIS	MMC Receive LPI microsecond counter interrupt status This bit is set when the Rx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. MMC Receive LPI microsecond counter interrupt status

Table continues on the next page...

Field	Description
	This bit is set when the Rx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive LPI microsecond Counter Interrupt Status not detected 1 - MMC Receive LPI microsecond Counter Interrupt Status detected
25 RXCTRLPIS	MMC Receive Control Packet Counter Interrupt Status This bit is set when the rxctrlpackets_g counter reaches half of the maximum value or the maximum value. MMC Receive Control Packet Counter Interrupt Status This bit is set when the rxctrlpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Control Packet Counter Interrupt Status not detected 1 - MMC Receive Control Packet Counter Interrupt Status detected
24 RXRCVERRPIS	MMC Receive Error Packet Counter Interrupt Status This bit is set when the rxrcverror counter reaches half of the maximum value or the maximum value. MMC Receive Error Packet Counter Interrupt Status This bit is set when the rxrcverror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Error Packet Counter Interrupt Status not detected 1 - MMC Receive Error Packet Counter Interrupt Status detected
23 RXWDOGPIS	MMC Receive Watchdog Error Packet Counter Interrupt Status This bit is set when the rxwatchdog error counter reaches half of the maximum value or the maximum value. MMC Receive Watchdog Error Packet Counter Interrupt Status This bit is set when the rxwatchdog error counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Watchdog Error Packet Counter Interrupt Status not detected 1 - MMC Receive Watchdog Error Packet Counter Interrupt Status detected
22 RXVLANGBPIS	MMC Receive VLAN Good Bad Packet Counter Interrupt Status This bit is set when the rxvlanpackets_gb counter reaches half of the maximum value or the maximum value. MMC Receive VLAN Good Bad Packet Counter Interrupt Status This bit is set when the rxvlanpackets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive VLAN Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive VLAN Good Bad Packet Counter Interrupt Status detected
21 RXFOVPIS	MMC Receive FIFO Overflow Packet Counter Interrupt Status This bit is set when the rxfifooverflow counter reaches half of the maximum value or the maximum value. MMC Receive FIFO Overflow Packet Counter Interrupt Status This bit is set when the rxfifooverflow counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive FIFO Overflow Packet Counter Interrupt Status not detected

Table continues on the next page...

Field	Description
	1 - MMC Receive FIFO Overflow Packet Counter Interrupt Status detected
20 RXPAUSPIS	MMC Receive Pause Packet Counter Interrupt Status This bit is set when the rxpausepackets counter reaches half of the maximum value or the maximum value. MMC Receive Pause Packet Counter Interrupt Status This bit is set when the rxpausepackets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Pause Packet Counter Interrupt Status not detected 1 - MMC Receive Pause Packet Counter Interrupt Status detected
19 RXORANGEPIS	MMC Receive Out Of Range Error Packet Counter Interrupt Status. MMC Receive Out Of Range Error Packet Counter Interrupt Status. This bit is set when the rxoutofrangetype counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Out Of Range Error Packet Counter Interrupt Status not detected 1 - MMC Receive Out Of Range Error Packet Counter Interrupt Status detected
18 RXLENERPIS	MMC Receive Length Error Packet Counter Interrupt Status This bit is set when the rxlengtherror counter reaches half of the maximum value or the maximum value. MMC Receive Length Error Packet Counter Interrupt Status This bit is set when the rxlengtherror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Length Error Packet Counter Interrupt Status not detected 1 - MMC Receive Length Error Packet Counter Interrupt Status detected
17 RXUCGPIS	MMC Receive Unicast Good Packet Counter Interrupt Status This bit is set when the rxunicastpackets_g counter reaches half of the maximum value or the maximum value. MMC Receive Unicast Good Packet Counter Interrupt Status This bit is set when the rxunicastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Unicast Good Packet Counter Interrupt Status not detected 1 - MMC Receive Unicast Good Packet Counter Interrupt Status detected
16 RX1024TMAXOCTGBPIS	MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status detected
15 RX512T1023OCTGBPIS	MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx512to1023octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status

Table continues on the next page...

Field	Description
	This bit is set when the rx512to1023octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status detected
14 RX256T511OCTGBPIS	MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx256to511octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx256to511octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status detected
13 RX128T255OCTGBPIS	MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx128to255octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx128to255octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status detected
12 RX65T127OCTGBPIS	MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx65to127octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx65to127octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status detected
11 RX64OCTGBPIS	MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx64octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx64octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status detected
10 RXOSIZEGPIS	MMC Receive Oversize Good Packet Counter Interrupt Status This bit is set when the rxoversize_g counter reaches half of the maximum value or the maximum value. MMC Receive Oversize Good Packet Counter Interrupt Status This bit is set when the rxoversize_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Oversize Good Packet Counter Interrupt Status not detected

Table continues on the next page...

Field	Description
	1 - MMC Receive Oversize Good Packet Counter Interrupt Status detected
9 RXUSIZEGPIS	MMC Receive Undersize Good Packet Counter Interrupt Status This bit is set when the rxundersize_g counter reaches half of the maximum value or the maximum value. MMC Receive Undersize Good Packet Counter Interrupt Status This bit is set when the rxundersize_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Undersize Good Packet Counter Interrupt Status not detected 1 - MMC Receive Undersize Good Packet Counter Interrupt Status detected
8 RXJABERPIS	MMC Receive Jabber Error Packet Counter Interrupt Status This bit is set when the rxjabbererror counter reaches half of the maximum value or the maximum value. MMC Receive Jabber Error Packet Counter Interrupt Status This bit is set when the rxjabbererror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Jabber Error Packet Counter Interrupt Status not detected 1 - MMC Receive Jabber Error Packet Counter Interrupt Status detected
7 RXRUNTPIS	MMC Receive Runt Packet Counter Interrupt Status This bit is set when the rxrunterror counter reaches half of the maximum value or the maximum value. MMC Receive Runt Packet Counter Interrupt Status This bit is set when the rxrunterror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Runt Packet Counter Interrupt Status not detected 1 - MMC Receive Runt Packet Counter Interrupt Status detected
6 RXALGNERPIS	MMC Receive Alignment Error Packet Counter Interrupt Status This bit is set when the rxalignmenterror counter reaches half of the maximum value or the maximum value. MMC Receive Alignment Error Packet Counter Interrupt Status This bit is set when the rxalignmenterror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Alignment Error Packet Counter Interrupt Status not detected 1 - MMC Receive Alignment Error Packet Counter Interrupt Status detected
5 RXCRCERPIS	MMC Receive CRC Error Packet Counter Interrupt Status This bit is set when the rxcrcerror counter reaches half of the maximum value or the maximum value. MMC Receive CRC Error Packet Counter Interrupt Status This bit is set when the rxcrcerror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive CRC Error Packet Counter Interrupt Status not detected 1 - MMC Receive CRC Error Packet Counter Interrupt Status detected
4 RXMCGPIS	MMC Receive Multicast Good Packet Counter Interrupt Status This bit is set when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. MMC Receive Multicast Good Packet Counter Interrupt Status

Table continues on the next page...

Field	Description
	This bit is set when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Multicast Good Packet Counter Interrupt Status not detected 1 - MMC Receive Multicast Good Packet Counter Interrupt Status detected
3 RXBCGPIS	MMC Receive Broadcast Good Packet Counter Interrupt Status This bit is set when the rxbroadcastpackets_g counter reaches half of the maximum value or the maximum value. MMC Receive Broadcast Good Packet Counter Interrupt Status This bit is set when the rxbroadcastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Broadcast Good Packet Counter Interrupt Status not detected 1 - MMC Receive Broadcast Good Packet Counter Interrupt Status detected
2 RXGOCTIS	MMC Receive Good Octet Counter Interrupt Status This bit is set when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. MMC Receive Good Octet Counter Interrupt Status This bit is set when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Good Octet Counter Interrupt Status not detected 1 - MMC Receive Good Octet Counter Interrupt Status detected
1 RXGBOCTIS	MMC Receive Good Bad Octet Counter Interrupt Status This bit is set when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. MMC Receive Good Bad Octet Counter Interrupt Status This bit is set when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Good Bad Octet Counter Interrupt Status not detected 1 - MMC Receive Good Bad Octet Counter Interrupt Status detected
0 RXGBPKTIS	MMC Receive Good Bad Packet Counter Interrupt Status This bit is set when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. MMC Receive Good Bad Packet Counter Interrupt Status This bit is set when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive Good Bad Packet Counter Interrupt Status not detected 1 - MMC Receive Good Bad Packet Counter Interrupt Status detected

11.7.6.1.178 MMC Tx Interrupt (MAC_MMC_TX_INTERRUPT)

This register maintains the interrupts generated from all Transmit statistics counters.

This register maintains the interrupts generated from all Transmit statistics counters.

The MMC Transmit Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values

(0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter).

When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones.

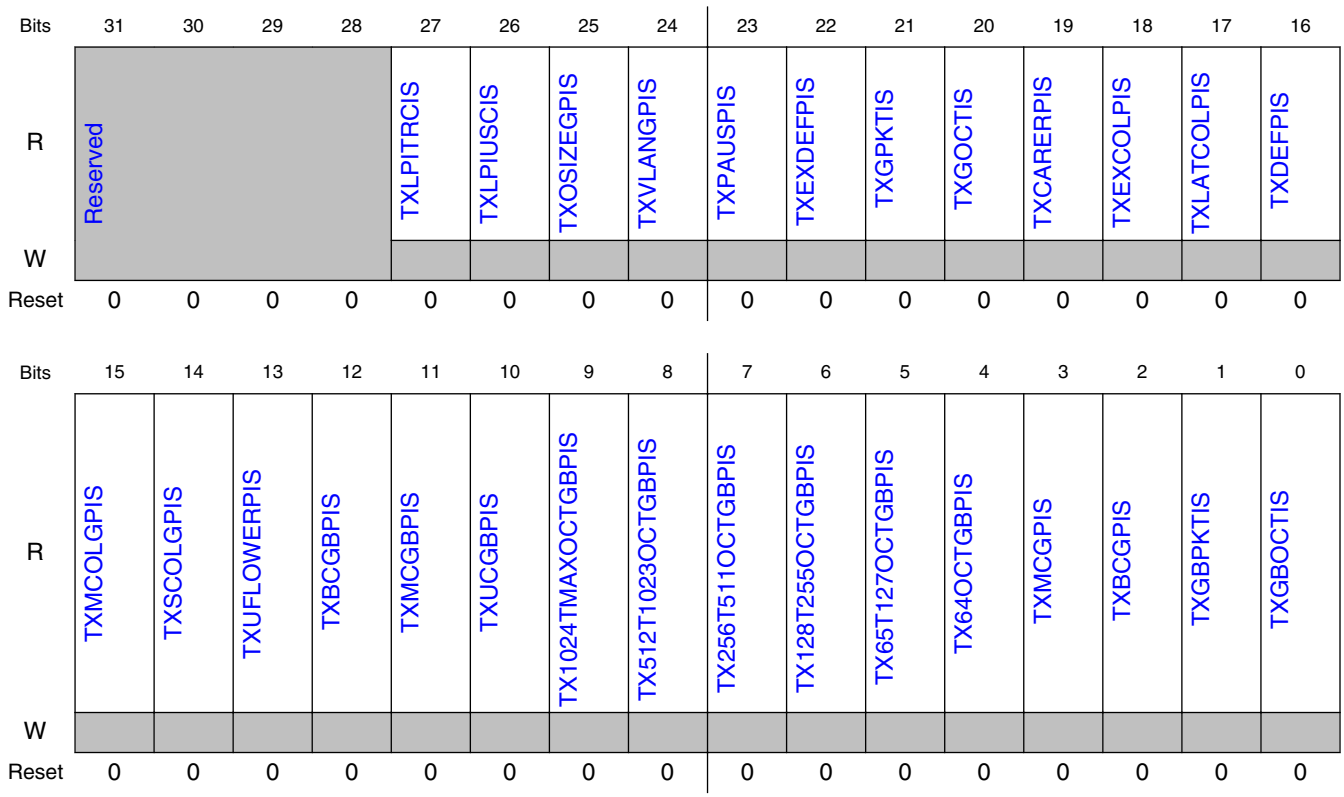
The MMC Transmit Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read.

The least significant byte lane (Bits[7:0]) of the respective counter must be read to clear the interrupt bit.

11.7.6.1.178.1 Offset

Register	Offset
MAC_MMC_TX_INTERRUPT	708h

11.7.6.1.178.2 Diagram



11.7.6.1.178.3 Fields

Field	Description
31-28 —	Reserved
27 TXLPITRCIS	MMC Transmit LPI transition counter interrupt status This bit is set when the Tx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. MMC Transmit LPI transition counter interrupt status This bit is set when the Tx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit LPI transition Counter Interrupt Status not detected 1 - MMC Transmit LPI transition Counter Interrupt Status detected
26 TXLPIUSCIS	MMC Transmit LPI microsecond counter interrupt status This bit is set when the Tx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. MMC Transmit LPI microsecond counter interrupt status This bit is set when the Tx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event.

Table continues on the next page...

Field	Description
	0 - MMC Transmit LPI microsecond Counter Interrupt Status not detected
	1 - MMC Transmit LPI microsecond Counter Interrupt Status detected
25 TXOSIZEGPIS	MMC Transmit Oversize Good Packet Counter Interrupt Status This bit is set when the txoversize_g counter reaches half of the maximum value or the maximum value. MMC Transmit Oversize Good Packet Counter Interrupt Status This bit is set when the txoversize_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Oversize Good Packet Counter Interrupt Status not detected 1 - MMC Transmit Oversize Good Packet Counter Interrupt Status detected
24 TXVLANGPIS	MMC Transmit VLAN Good Packet Counter Interrupt Status This bit is set when the txvlanpackets_g counter reaches half of the maximum value or the maximum value. MMC Transmit VLAN Good Packet Counter Interrupt Status This bit is set when the txvlanpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit VLAN Good Packet Counter Interrupt Status not detected 1 - MMC Transmit VLAN Good Packet Counter Interrupt Status detected
23 TXPAUSPIS	MMC Transmit Pause Packet Counter Interrupt Status This bit is set when the txpausepacketserror counter reaches half of the maximum value or the maximum value. MMC Transmit Pause Packet Counter Interrupt Status This bit is set when the txpausepacketserror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Pause Packet Counter Interrupt Status not detected 1 - MMC Transmit Pause Packet Counter Interrupt Status detected
22 TXEXDEFPIS	MMC Transmit Excessive Deferral Packet Counter Interrupt Status This bit is set when the txexcessdef counter reaches half of the maximum value or the maximum value. MMC Transmit Excessive Deferral Packet Counter Interrupt Status This bit is set when the txexcessdef counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Excessive Deferral Packet Counter Interrupt Status not detected 1 - MMC Transmit Excessive Deferral Packet Counter Interrupt Status detected
21 TXGPKTIS	MMC Transmit Good Packet Counter Interrupt Status This bit is set when the txpacketcount_g counter reaches half of the maximum value or the maximum value. MMC Transmit Good Packet Counter Interrupt Status This bit is set when the txpacketcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Good Packet Counter Interrupt Status not detected 1 - MMC Transmit Good Packet Counter Interrupt Status detected
20	MMC Transmit Good Octet Counter Interrupt Status This bit is set when the txoctetcount_g counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Field	Description
TXGOCTIS	MMC Transmit Good Octet Counter Interrupt Status This bit is set when the txoctetcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Good Octet Counter Interrupt Status not detected 1 - MMC Transmit Good Octet Counter Interrupt Status detected
19 TXCARERPIS	MMC Transmit Carrier Error Packet Counter Interrupt Status This bit is set when the txcarriererror counter reaches half of the maximum value or the maximum value. MMC Transmit Carrier Error Packet Counter Interrupt Status This bit is set when the txcarriererror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Carrier Error Packet Counter Interrupt Status not detected 1 - MMC Transmit Carrier Error Packet Counter Interrupt Status detected
18 TXEXCOLPIS	MMC Transmit Excessive Collision Packet Counter Interrupt Status This bit is set when the txexesscol counter reaches half of the maximum value or the maximum value. MMC Transmit Excessive Collision Packet Counter Interrupt Status This bit is set when the txexesscol counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Excessive Collision Packet Counter Interrupt Status not detected 1 - MMC Transmit Excessive Collision Packet Counter Interrupt Status detected
17 TXLATCOLPIS	MMC Transmit Late Collision Packet Counter Interrupt Status This bit is set when the txlatecol counter reaches half of the maximum value or the maximum value. MMC Transmit Late Collision Packet Counter Interrupt Status This bit is set when the txlatecol counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Late Collision Packet Counter Interrupt Status not detected 1 - MMC Transmit Late Collision Packet Counter Interrupt Status detected
16 TXDEFPIS	MMC Transmit Deferred Packet Counter Interrupt Status This bit is set when the txdeferred counter reaches half of the maximum value or the maximum value. MMC Transmit Deferred Packet Counter Interrupt Status This bit is set when the txdeferred counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Deferred Packet Counter Interrupt Status not detected 1 - MMC Transmit Deferred Packet Counter Interrupt Status detected
15 TXMCOLGPIS	MMC Transmit Multiple Collision Good Packet Counter Interrupt Status This bit is set when the txmulticol_g counter reaches half of the maximum value or the maximum value. MMC Transmit Multiple Collision Good Packet Counter Interrupt Status This bit is set when the txmulticol_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Multiple Collision Good Packet Counter Interrupt Status not detected 1 - MMC Transmit Multiple Collision Good Packet Counter Interrupt Status detected

Table continues on the next page...

Field	Description
14 TXSCOLGPIS	MMC Transmit Single Collision Good Packet Counter Interrupt Status This bit is set when the txsinglecol_g counter reaches half of the maximum value or the maximum value. MMC Transmit Single Collision Good Packet Counter Interrupt Status This bit is set when the txsinglecol_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Single Collision Good Packet Counter Interrupt Status not detected 1 - MMC Transmit Single Collision Good Packet Counter Interrupt Status detected
13 TXUFLOWERPIS	MMC Transmit Underflow Error Packet Counter Interrupt Status This bit is set when the txunderflowerror counter reaches half of the maximum value or the maximum value. MMC Transmit Underflow Error Packet Counter Interrupt Status This bit is set when the txunderflowerror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Underflow Error Packet Counter Interrupt Status not detected 1 - MMC Transmit Underflow Error Packet Counter Interrupt Status detected
12 TXBCGBPIS	MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status This bit is set when the txbroadcastpackets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status This bit is set when the txbroadcastpackets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status detected
11 TXMCGBPIS	MMC Transmit Multicast Good Bad Packet Counter Interrupt Status The bit is set when the txmulticastpackets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Multicast Good Bad Packet Counter Interrupt Status The bit is set when the txmulticastpackets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Multicast Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit Multicast Good Bad Packet Counter Interrupt Status detected
10 TXUCGBPIS	MMC Transmit Unicast Good Bad Packet Counter Interrupt Status This bit is set when the txunicastpackets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Unicast Good Bad Packet Counter Interrupt Status This bit is set when the txunicastpackets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Unicast Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit Unicast Good Bad Packet Counter Interrupt Status detected
9 TX1024TMAXOCTGBPIS	MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status

Table continues on the next page...

Field	Description
	This bit is set when the tx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status detected
8 TX512T1023OCTGBPIS	MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx512to1023octets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx512to1023octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status detected
7 TX256T511OCTGBPIS	MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx256to511octets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx256to511octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status detected
6 TX128T255OCTGBPIS	MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx128to255octets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx128to255octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status detected
5 TX65T127OCTGBPIS	MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx65to127octets_gb counter reaches half the maximum value, and also when it reaches the maximum value. MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx65to127octets_gb counter reaches half the maximum value, and also when it reaches the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status detected
4 TX64OCTGBPIS	MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx64octets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx64octets_gb counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Field	Description
	Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status detected
3 TXMCGPIS	MMC Transmit Multicast Good Packet Counter Interrupt Status This bit is set when the txmulticastpackets_g counter reaches half of the maximum value or the maximum value. MMC Transmit Multicast Good Packet Counter Interrupt Status This bit is set when the txmulticastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Multicast Good Packet Counter Interrupt Status not detected 1 - MMC Transmit Multicast Good Packet Counter Interrupt Status detected
2 TXBCGPIS	MMC Transmit Broadcast Good Packet Counter Interrupt Status This bit is set when the txbroadcastpackets_g counter reaches half of the maximum value or the maximum value. MMC Transmit Broadcast Good Packet Counter Interrupt Status This bit is set when the txbroadcastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Broadcast Good Packet Counter Interrupt Status not detected 1 - MMC Transmit Broadcast Good Packet Counter Interrupt Status detected
1 TXGBPKTIS	MMC Transmit Good Bad Packet Counter Interrupt Status This bit is set when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Good Bad Packet Counter Interrupt Status This bit is set when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Good Bad Packet Counter Interrupt Status not detected 1 - MMC Transmit Good Bad Packet Counter Interrupt Status detected
0 TXGBOCTIS	MMC Transmit Good Bad Octet Counter Interrupt Status This bit is set when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Good Bad Octet Counter Interrupt Status This bit is set when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Transmit Good Bad Octet Counter Interrupt Status not detected 1 - MMC Transmit Good Bad Octet Counter Interrupt Status detected

11.7.6.1.179 MMC Rx Interrupt Mask (MAC_MMC_RX_INTERRUPT_MASK)

This register maintains the masks for interrupts generated from all Receive statistics counters.

This register maintains the masks for interrupts generated from all Receive statistics counters.

The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when receive statistic counters reach half of their maximum value or the maximum values.

This register is 32 bit wide.

11.7.6.1.179.1 Offset

Register	Offset
MAC_MMC_RX_INTERRUPT_MASK	70Ch

11.7.6.1.179.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				RXLPITRCIM	RXLPIUSCIM	RXCTRLPIM	RXRCVERRPIM	RXWDOGPIM	RXVLANGBPIM	RXFOVPIM	RXPAUSPIM	RXORANGEPIM	RXLENERPIM	RXUCGPIM	RX1024TMAXOCTGBPIM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX512T1023OCTGBPIM	RX256T511OCTGBPIM	RX128T255OCTGBPIM	RX65T127OCTGBPIM	RX64OCTGBPIM	RXOSIZEGPIM	RXUSIZEGPIM	RXJABERPIM	RXRUNTPIM	RXALGNERPIM	RXCRCERPIM	RXMCGPIM	RXBCGPIM	RXGOCTIM	RXGBOCTIM	RXGBPKTIM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.179.3 Fields

Field	Description
31-28 —	Reserved
27 RXLPITRCIM	MMC Receive LPI transition counter interrupt Mask Setting this bit masks the interrupt when the Rx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. 0 - MMC Receive LPI transition counter interrupt Mask is disabled 1 - MMC Receive LPI transition counter interrupt Mask is enabled
26 RXLPIUSCIM	MMC Receive LPI microsecond counter interrupt Mask Setting this bit masks the interrupt when the Rx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. 0 - MMC Receive LPI microsecond counter interrupt Mask is disabled 1 - MMC Receive LPI microsecond counter interrupt Mask is enabled
25 RXCTRLPIM	MMC Receive Control Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxctrlpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Control Packet Counter Interrupt Mask is disabled 1 - MMC Receive Control Packet Counter Interrupt Mask is enabled
24 RXRCVERRPIM	MMC Receive Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxrcverror counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive Error Packet Counter Interrupt Mask is enabled
23 RXWDOGPIM	MMC Receive Watchdog Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxwatchdog counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Watchdog Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive Watchdog Error Packet Counter Interrupt Mask is enabled
22 RXVLANGBPIM	MMC Receive VLAN Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxvlanpackets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive VLAN Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive VLAN Good Bad Packet Counter Interrupt Mask is enabled
21 RXFOVPIM	MMC Receive FIFO Overflow Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxfifooverflow counter reaches half of the maximum value or the maximum value. 0 - MMC Receive FIFO Overflow Packet Counter Interrupt Mask is disabled 1 - MMC Receive FIFO Overflow Packet Counter Interrupt Mask is enabled
20 RXPAUSPIM	MMC Receive Pause Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxpausepackets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Pause Packet Counter Interrupt Mask is disabled 1 - MMC Receive Pause Packet Counter Interrupt Mask is enabled
19 RXORANGEPIM	MMC Receive Out Of Range Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoutofrangetype counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Out Of Range Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive Out Of Range Error Packet Counter Interrupt Mask is enabled
18 RXLENERPIM	MMC Receive Length Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxlengtherror counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Length Error Packet Counter Interrupt Mask is disabled

Table continues on the next page...

Field	Description
	1 - MMC Receive Length Error Packet Counter Interrupt Mask is enabled
17 RXUCGPIM	MMC Receive Unicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxunicastpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Unicast Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive Unicast Good Packet Counter Interrupt Mask is enabled
16 RX1024TMAXOCTGBPIM	MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask. MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask. Setting this bit masks the interrupt when the rx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is enabled
15 RX512T1023OCTGBPIM	MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx512to1023octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is enabled
14 RX256T511OCTGBPIM	MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx256to511octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is enabled
13 RX128T255OCTGBPIM	MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx128to255octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is enabled
12 RX65T127OCTGBPIM	MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx65to127octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is enabled
11 RX64OCTGBPIM	MMC Receive 64 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx64octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive 64 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive 64 Octet Good Bad Packet Counter Interrupt Mask is enabled
10 RXOSIZEGPIM	MMC Receive Oversize Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoversize_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Oversize Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive Oversize Good Packet Counter Interrupt Mask is enabled
9 RXUSIZEGPIM	MMC Receive Undersize Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxundersize_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Undersize Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive Undersize Good Packet Counter Interrupt Mask is enabled

Table continues on the next page...

Field	Description
8 RXJABERPIM	MMC Receive Jabber Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxjabbererror counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Jabber Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive Jabber Error Packet Counter Interrupt Mask is enabled
7 RXRUNTPIM	MMC Receive Runt Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxrunterror counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Runt Packet Counter Interrupt Mask is disabled 1 - MMC Receive Runt Packet Counter Interrupt Mask is enabled
6 RXALGNERPIM	MMC Receive Alignment Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxalignmenterror counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Alignment Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive Alignment Error Packet Counter Interrupt Mask is enabled
5 RXCRCERPIM	MMC Receive CRC Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxcrcerror counter reaches half of the maximum value or the maximum value. 0 - MMC Receive CRC Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive CRC Error Packet Counter Interrupt Mask is enabled
4 RXMCGPIM	MMC Receive Multicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Multicast Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive Multicast Good Packet Counter Interrupt Mask is enabled
3 RXBCGPIM	MMC Receive Broadcast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxbroadcastpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Broadcast Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive Broadcast Good Packet Counter Interrupt Mask is enabled
2 RXGOCTIM	MMC Receive Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive Good Octet Counter Interrupt Mask is enabled
1 RXGBOCTIM	MMC Receive Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Good Bad Octet Counter Interrupt Mask is disabled 1 - MMC Receive Good Bad Octet Counter Interrupt Mask is enabled
0 RXGBPKTIM	MMC Receive Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Receive Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Receive Good Bad Packet Counter Interrupt Mask is enabled

11.7.6.1.180 MMC Tx Interrupt Mask (MAC_MMC_TX_INTERRUPT_MASK)

This register maintains the masks for interrupts generated from all Transmit statistics counters.

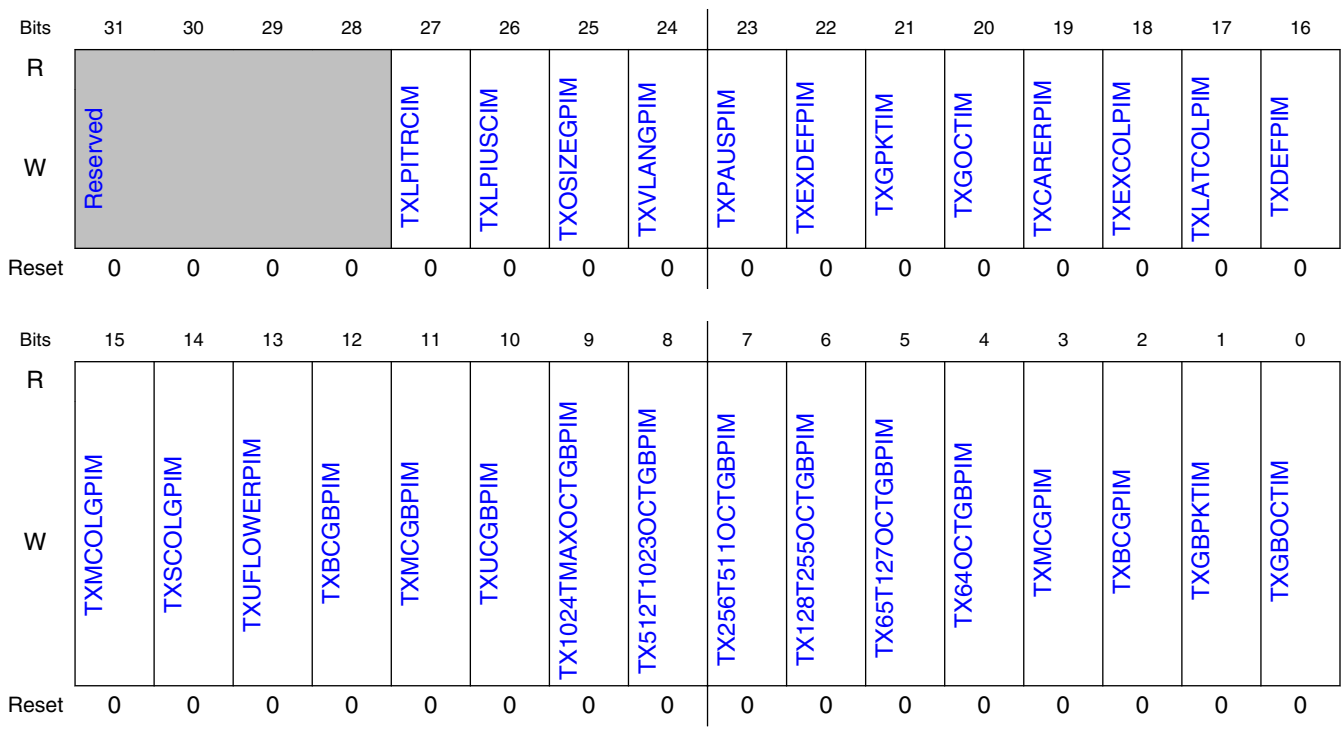
This register maintains the masks for interrupts generated from all Transmit statistics counters.

The MMC Transmit Interrupt Mask register maintains the masks for the interrupts generated when the transmit statistic counters reach half of their maximum value or the maximum values. This register is 32 bit wide.

11.7.6.1.180.1 Offset

Register	Offset
MAC_MMC_TX_INTERRUPT_MASK	710h

11.7.6.1.180.2 Diagram



11.7.6.1.180.3 Fields

Field	Description
31-28 —	Reserved

Table continues on the next page...

Field	Description
27 TXLPITRCIM	MMC Transmit LPI transition counter interrupt Mask Setting this bit masks the interrupt when the Tx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit LPI transition counter interrupt Mask is disabled 1 - MMC Transmit LPI transition counter interrupt Mask is enabled
26 TXLPIUSCIM	MMC Transmit LPI microsecond counter interrupt Mask Setting this bit masks the interrupt when the Tx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit LPI microsecond counter interrupt Mask is disabled 1 - MMC Transmit LPI microsecond counter interrupt Mask is enabled
25 TXOSIZEGPIM	MMC Transmit Oversize Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txoversize_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Oversize Good Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Oversize Good Packet Counter Interrupt Mask is enabled
24 TXVLANGPIM	MMC Transmit VLAN Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txvlanpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit VLAN Good Packet Counter Interrupt Mask is disabled 1 - MMC Transmit VLAN Good Packet Counter Interrupt Mask is enabled
23 TXPAUSPIM	MMC Transmit Pause Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpausepackets counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Pause Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Pause Packet Counter Interrupt Mask is enabled
22 TXEXDEFPIM	MMC Transmit Excessive Deferral Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txexcessdef counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Excessive Deferral Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Excessive Deferral Packet Counter Interrupt Mask is enabled
21 TXGPKTIM	MMC Transmit Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpacketcount_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Good Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Good Packet Counter Interrupt Mask is enabled
20 TXGOCTIM	MMC Transmit Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Good Octet Counter Interrupt Mask is disabled 1 - MMC Transmit Good Octet Counter Interrupt Mask is enabled
19 TXCARERPIM	MMC Transmit Carrier Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txcarriererror counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Carrier Error Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Carrier Error Packet Counter Interrupt Mask is enabled
18 TXEXCOLPIM	MMC Transmit Excessive Collision Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txexcesscol counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Excessive Collision Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Excessive Collision Packet Counter Interrupt Mask is enabled
17 TXLATCOLPIM	MMC Transmit Late Collision Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txlatecol counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Late Collision Packet Counter Interrupt Mask is disabled

Table continues on the next page...

Field	Description
	1 - MMC Transmit Late Collision Packet Counter Interrupt Mask is enabled
16 TXDEFPIM	MMC Transmit Deferred Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txdeferred counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Deferred Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Deferred Packet Counter Interrupt Mask is enabled
15 TXMCOLGPIM	MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticol_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask is enabled
14 TXSCOLGPIM	MMC Transmit Single Collision Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txsinglecol_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Single Collision Good Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Single Collision Good Packet Counter Interrupt Mask is enabled
13 TXUFLOWERPIM	MMC Transmit Underflow Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txunderflowerror counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Underflow Error Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Underflow Error Packet Counter Interrupt Mask is enabled
12 TXBCGBPIM	MMC Transmit Broadcast Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txbroadcastpackets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Broadcast Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Broadcast Good Bad Packet Counter Interrupt Mask is enabled
11 TXMCGBPIM	MMC Transmit Multicast Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticastpackets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Multicast Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Multicast Good Bad Packet Counter Interrupt Mask is enabled
10 TXUCGBPIM	MMC Transmit Unicast Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txunicastpackets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Unicast Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Unicast Good Bad Packet Counter Interrupt Mask is enabled
9 TX1024TMAXOCTGBPIM	MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is enabled
8 TX512T1023OCTGBPIM	MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx512to1023octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is enabled
7 TX256T511OCTGBPIM	MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx256to511octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is disabled

Table continues on the next page...

Field	Description
	1 - MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is enabled
6 TX128T255OCTGBPIM	MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx128to255octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is enabled
5 TX65T127OCTGBPIM	MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx65to127octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is enabled
4 TX64OCTGBPIM	MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx64octets_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Mask is enabled
3 TXMCGPIM	MMC Transmit Multicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticastpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Multicast Good Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Multicast Good Packet Counter Interrupt Mask is enabled
2 TXBCGPIM	MMC Transmit Broadcast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txbroadcastpackets_g counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Broadcast Good Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Broadcast Good Packet Counter Interrupt Mask is enabled
1 TXGBPKTIM	MMC Transmit Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Good Bad Packet Counter Interrupt Mask is disabled 1 - MMC Transmit Good Bad Packet Counter Interrupt Mask is enabled
0 TXGBOCTIM	MMC Transmit Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. 0 - MMC Transmit Good Bad Octet Counter Interrupt Mask is disabled 1 - MMC Transmit Good Bad Octet Counter Interrupt Mask is enabled

11.7.6.1.181 Tx Octet Count Good and Bad (MAC_TX_OCTET_COUNT_GOOD_BAD)

This register provides the number of bytes transmitted by the DWC_ether_qos, exclusive of preamble and retried bytes, in good and bad packets.

11.7.6.1.181.1 Offset

Register	Offset
MAC_TX_OCTET_COUNT_GOOD_BAD	714h

11.7.6.1.181.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.181.3 Fields

Field	Description
31-0 TXOCTGB	Tx Octet Count Good Bad This field indicates the number of bytes transmitted, exclusive of preamble and retried bytes, in good and bad packets.

11.7.6.1.182 Tx Packet Count Good and Bad (MAC_TX_PACKET_COUNT_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos, exclusive of retried packets.

11.7.6.1.182.1 Offset

Register	Offset
MAC_TX_PACKET_COUNT_GOOD_BAD	718h

11.7.6.1.182.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.182.3 Fields

Field	Description
31-0 TXPKTGB	Tx Packet Count Good Bad This field indicates the number of good and bad packets transmitted, exclusive of retried packets.

11.7.6.1.183 Tx Broadcast Packets Good (MAC_TX_BROADCAST_PACKETS_GOOD)

This register provides the number of good broadcast packets transmitted by DWC_ether_qos.

11.7.6.1.183.1 Offset

Register	Offset
MAC_TX_BROADCAST_PACKETS_GOOD	71Ch

11.7.6.1.183.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXBCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXBCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.183.3 Fields

Field	Description
31-0 TXBCASTG	Tx Broadcast Packets Good This field indicates the number of good broadcast packets transmitted.

11.7.6.1.184 Tx Multicast Packets Good (MAC_TX_MULTICAST_PACKETS_GOOD)

This register provides the number of good multicast packets transmitted by DWC_ether_qos.

11.7.6.1.184.1 Offset

Register	Offset
MAC_TX_MULTICAST_ PACKETS_GOOD	720h

11.7.6.1.184.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXMCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXMCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.184.3 Fields

Field	Description
31-0 TXMCASTG	Tx Multicast Packets Good This field indicates the number of good multicast packets transmitted.

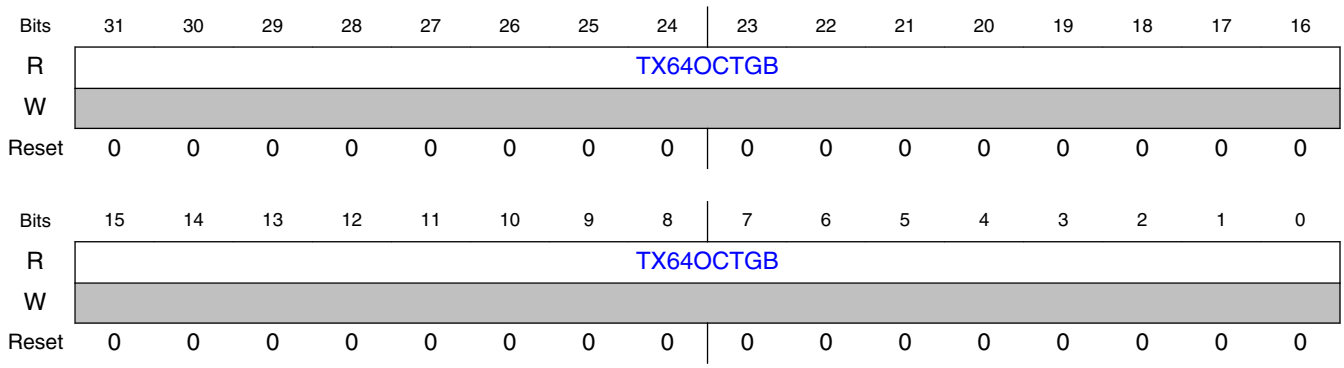
11.7.6.1.185 Tx Good and Bad 64-Byte Packets (MAC_TX_64OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length 64 bytes, exclusive of preamble and retried packets.

11.7.6.1.185.1 Offset

Register	Offset
MAC_TX_64OCTETS_PACKETS_GOOD_BAD	724h

11.7.6.1.185.2 Diagram



11.7.6.1.185.3 Fields

Field	Description
31-0 TX64OCTGB	Tx 64Octets Packets Good_Bad This field indicates the number of good and bad packets transmitted with length 64 bytes, exclusive of preamble and retried packets.

11.7.6.1.186 Tx Good and Bad 65 to 127-Byte Packets (MAC_TX_65TO127OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length between 65 and 127 (inclusive) bytes, exclusive of preamble and retried packets.

11.7.6.1.186.1 Offset

Register	Offset
MAC_TX_65TO127OCTETS_PACKETS_GOOD_BAD	728h

11.7.6.1.186.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX65_127OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX65_127OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.186.3 Fields

Field	Description
31-0 TX65_127OCTGB	Tx 65To127Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 65 and 127 (inclusive) bytes, exclusive of preamble and retried packets.

11.7.6.1.187 Tx Good and Bad 128 to 255-Byte Packets (MAC_TX_128TO255OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length between 128 to 255 (inclusive) bytes, exclusive of preamble and retried packets.

11.7.6.1.187.1 Offset

Register	Offset
MAC_TX_128TO255OCTETS_PACKETS_GOOD_BAD	72Ch

11.7.6.1.187.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX128_255OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX128_255OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.187.3 Fields

Field	Description
31-0 TX128_255OCTGB	Tx 128To255Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 128 and 255 (inclusive) bytes, exclusive of preamble and retried packets.

11.7.6.1.188 Tx Good and Bad 256 to 511-Byte Packets (MAC_TX_256TO511OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length between 256 to 511 (inclusive) bytes, exclusive of preamble and retried packets.

11.7.6.1.188.1 Offset

Register	Offset
MAC_TX_256TO511OCTETS_PACKETS_GOOD_BAD	730h

11.7.6.1.188.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX256_511OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX256_511OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.188.3 Fields

Field	Description
31-0 TX256_511OCTGB	Tx 256To511Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 256 and 511 (inclusive) bytes, exclusive of preamble and retried packets.

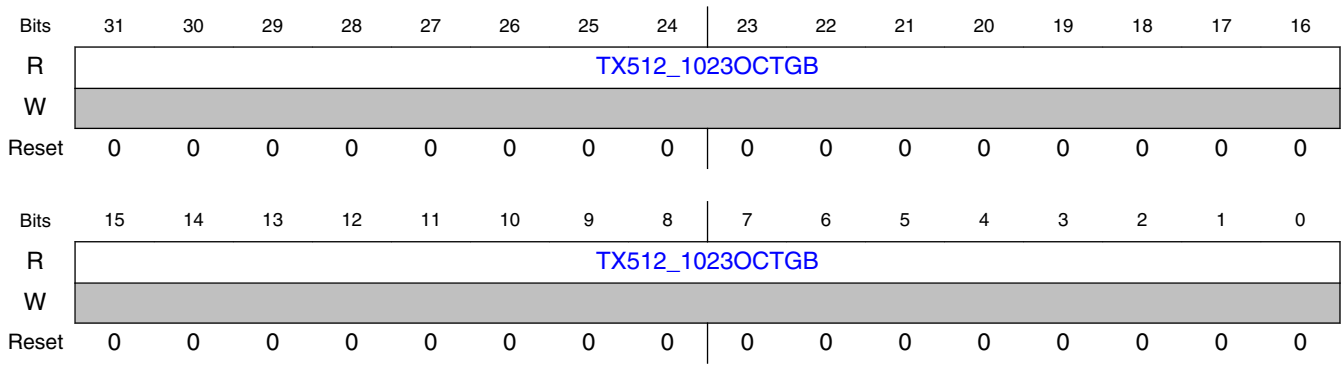
11.7.6.1.189 Tx Good and Bad 512 to 1023-Byte Packets (MAC_TX_512TO1023OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length 512 to 1023 (inclusive) bytes, exclusive of preamble and retried packets.

11.7.6.1.189.1 Offset

Register	Offset
MAC_TX_512TO1023OCTETS_PACKETS_GOOD_BAD	734h

11.7.6.1.189.2 Diagram



11.7.6.1.189.3 Fields

Field	Description
31-0 TX512_1023OCTGB	Tx 512To1023Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 512 and 1023 (inclusive) bytes, exclusive of preamble and retried packets.

11.7.6.1.190 Tx Good and Bad 1024 to Max-Byte Packets (MAC_TX_1024TOMAXOCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length 1024 to maxsize (inclusive) bytes, exclusive of preamble and retried packets.

11.7.6.1.190.1 Offset

Register	Offset
MAC_TX_1024TOMAXOCTETS_PACKETS_GOOD_BAD	738h

11.7.6.1.190.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX1024_MAXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX1024_MAXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.190.3 Fields

Field	Description
31-0 TX1024_MAXOCTGB	Tx 1024ToMaxOctets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 1024 and maxsize (inclusive) bytes, exclusive of preamble and retried packets.

11.7.6.1.191 Good and Bad Unicast Packets Transmitted (MAC_TX_UNICAST_PACKETS_GOOD_BAD)

This register provides the number of good and bad unicast packets transmitted by DWC_ether_qos.

11.7.6.1.191.1 Offset

Register	Offset
MAC_TX_UNICAST_PACKETS_GOOD_BAD	73Ch

11.7.6.1.191.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXUCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXUCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.191.3 Fields

Field	Description
31-0 TXUCASTGB	Tx Unicast Packets Good Bad This field indicates the number of good and bad unicast packets transmitted.

11.7.6.1.192 Good and Bad Multicast Packets Transmitted (MAC_TX_MULTICAST_PACKETS_GOOD_BAD)

This register provides the number of good and bad multicast packets transmitted by DWC_ether_qos.

11.7.6.1.192.1 Offset

Register	Offset
MAC_TX_MULTICAST_ PACKETS_GOOD_BAD	740h

11.7.6.1.192.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXMCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXMCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.192.3 Fields

Field	Description
31-0 TXMCASTGB	Tx Multicast Packets Good Bad This field indicates the number of good and bad multicast packets transmitted.

11.7.6.1.193 Good and Bad Broadcast Packets Transmitted (MAC_TX_BROADCAST_PACKETS_GOOD_BAD)

This register provides the number of good and bad broadcast packets transmitted by DWC_ether_qos.

11.7.6.1.193.1 Offset

Register	Offset
MAC_TX_BROADCAST_PACKETS_GOOD_BAD	744h

11.7.6.1.193.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXBCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXBCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.193.3 Fields

Field	Description
31-0 TXBCASTGB	Tx Broadcast Packets Good Bad This field indicates the number of good and bad broadcast packets transmitted.

11.7.6.1.194 Tx Packets Aborted By Underflow Error (MAC_TX_UNDERFLOW_ERROR_PACKETS)

This register provides the number of packets aborted by DWC_ether_qos because of packets underflow error.

11.7.6.1.194.1 Offset

Register	Offset
MAC_TX_UNDERFLOW_ERROR_PACKETS	748h

11.7.6.1.194.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXUNDRFLW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXUNDRFLW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.194.3 Fields

Field	Description
31-0 TXUNDRFLW	Tx Underflow Error Packets This field indicates the number of packets aborted because of packets underflow error.

11.7.6.1.195 Single Collision Good Packets Transmitted (MAC_TX_SINGLE_COLLISION_GOOD_PACKETS)

This register provides the number of successfully transmitted packets by DWC_ether_qos after a single collision in the half-duplex mode.

11.7.6.1.195.1 Offset

Register	Offset
MAC_TX_SINGLE_COLLISION_GOOD_PACKETS	74Ch

11.7.6.1.195.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXSNGLCOLG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXSNGLCOLG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.195.3 Fields

Field	Description
31-0 TXSNGLCOLG	Tx Single Collision Good Packets This field indicates the number of successfully transmitted packets after a single collision in the half-duplex mode.

11.7.6.1.196 Multiple Collision Good Packets Transmitted (MAC_TX_MULTIPLE_COLLISION_GOOD_PACKETS)

This register provides the number of successfully transmitted packets by DWC_ether_qos after multiple collisions in the half-duplex mode.

11.7.6.1.196.1 Offset

Register	Offset
MAC_TX_MULTIPLE_COLLISION_GOOD_PACKETS	750h

11.7.6.1.196.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXMULTCOLG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXMULTCOLG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.196.3 Fields

Field	Description
31-0 TXMULTCOLG	Tx Multiple Collision Good Packets This field indicates the number of successfully transmitted packets after multiple collisions in the half-duplex mode.

11.7.6.1.197 Deferred Packets Transmitted (MAC_TX_DEFERRED_PACKETS)

This register provides the number of successfully transmitted by DWC_ether_qos after a deferral in the half-duplex mode.

11.7.6.1.197.1 Offset

Register	Offset
MAC_TX_DEFERRED_PACKETS	754h

11.7.6.1.197.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXDEFRD															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXDEFRD															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.197.3 Fields

Field	Description
31-0 TXDEFRD	Tx Deferred Packets This field indicates the number of successfully transmitted after a deferral in the half-duplex mode.

11.7.6.1.198 Late Collision Packets Transmitted (MAC_TX_LATE_COLLISION_PACKETS)

This register provides the number of packets aborted by DWC_ether_qos because of late collision error.

11.7.6.1.198.1 Offset

Register	Offset
MAC_TX_LATE_COLLISION_PACKETS	758h

11.7.6.1.198.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXLATECOL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXLATECOL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.198.3 Fields

Field	Description
31-0 TXLATECOL	Tx Late Collision Packets This field indicates the number of packets aborted because of late collision error.

11.7.6.1.199 Excessive Collision Packets Transmitted (MAC_TX_EXCESSIVE_COLLISION_PACKETS)

This register provides the number of packets aborted by DWC_ether_qos because of excessive (16) collision errors.

11.7.6.1.199.1 Offset

Register	Offset
MAC_TX_EXCESSIVE_ COLLISION_PACKETS	75Ch

11.7.6.1.199.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXEXSCOL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXEXSCOL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.199.3 Fields

Field	Description
31-0 TXEXSCOL	Tx Excessive Collision Packets This field indicates the number of packets aborted because of excessive (16) collision errors.

11.7.6.1.200 Carrier Error Packets Transmitted (MAC_TX_CARRIER_ERROR_PACKETS)

This register provides the number of packets aborted by DWC_ether_qos because of carrier sense error (no carrier or loss of carrier).

11.7.6.1.200.1 Offset

Register	Offset
MAC_TX_CARRIER_ERROR_PACKETS	760h

11.7.6.1.200.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXCARR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXCARR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.200.3 Fields

Field	Description
31-0 TXCARR	Tx Carrier Error Packets This field indicates the number of packets aborted because of carrier sense error (no carrier or loss of carrier).

11.7.6.1.201 Bytes Transmitted in Good Packets (MAC_TX_OCTET_COUNT_GOOD)

This register provides the number of bytes transmitted by DWC_ether_qos, exclusive of preamble, only in good packets.

11.7.6.1.201.1 Offset

Register	Offset
MAC_TX_OCTET_COUNT_GOOD	764h

11.7.6.1.201.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXOCTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXOCTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.201.3 Fields

Field	Description
31-0 TXOCTG	Tx Octet Count Good This field indicates the number of bytes transmitted, exclusive of preamble, only in good packets.

11.7.6.1.202 Good Packets Transmitted (MAC_TX_PACKET_COUNT_GOOD)

This register provides the number of good packets transmitted by DWC_ether_qos.

11.7.6.1.202.1 Offset

Register	Offset
MAC_TX_PACKET_COUNT_GOOD	768h

11.7.6.1.202.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXPKTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.202.3 Fields

Field	Description
31-0 TXPKTG	Tx Packet Count Good This field indicates the number of good packets transmitted.

11.7.6.1.203 Packets Aborted By Excessive Deferral Error (MAC_TX_EXCESSIVE_DEFERRAL_ERROR)

This register provides the number of packets aborted by DWC_ether_qos because of excessive deferral error (deferred for more than two max-sized packet times).

11.7.6.1.203.1 Offset

Register	Offset
MAC_TX_EXCESSIVE_ DEFERRAL_ERROR	76Ch

11.7.6.1.203.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXEXSDEF															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXEXSDEF															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.203.3 Fields

Field	Description
31-0 TXEXSDEF	Tx Excessive Deferral Error This field indicates the number of packets aborted because of excessive deferral error (deferred for more than two max-sized packet times).

11.7.6.1.204 Pause Packets Transmitted (MAC_TX_PAUSE_PACKETS)

This register provides the number of good Pause packets transmitted by DWC_ether_qos.

11.7.6.1.204.1 Offset

Register	Offset
MAC_TX_PAUSE_PACKETS	770h

11.7.6.1.204.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXPAUSE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPAUSE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.204.3 Fields

Field	Description
31-0 TXPAUSE	Tx Pause Packets This field indicates the number of good Pause packets transmitted.

11.7.6.1.205 Good VLAN Packets Transmitted (MAC_TX_VLAN_PACKETS_GOOD)

This register provides the number of good VLAN packets transmitted by DWC_ether_qos.

11.7.6.1.205.1 Offset

Register	Offset
MAC_TX_VLAN_PACKETS_GOOD	774h

11.7.6.1.205.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXVLANG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXVLANG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.205.3 Fields

Field	Description
31-0 TXVLANG	Tx VLAN Packets Good This field provides the number of good VLAN packets transmitted.

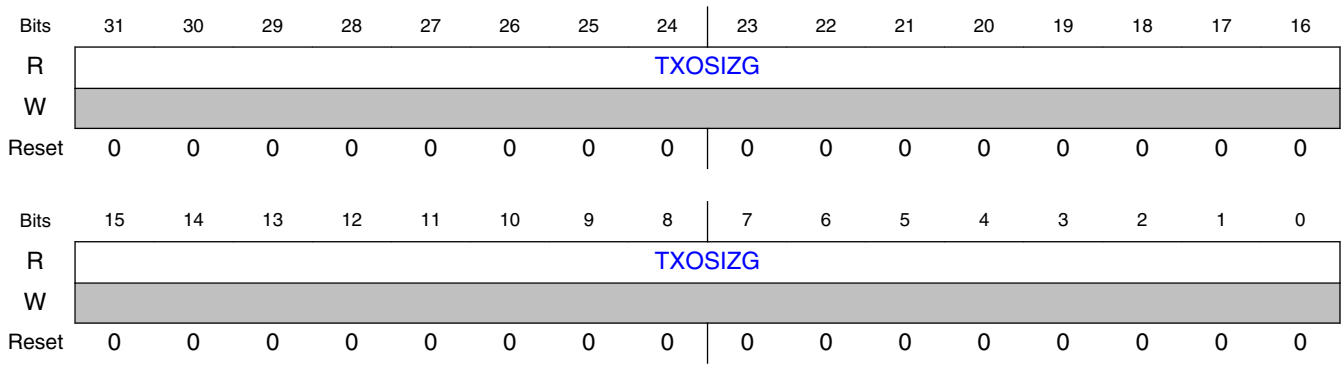
11.7.6.1.206 Good Oversize Packets Transmitted (MAC_TX_OSIZE_PACKETS_GOOD)

This register provides the number of packets transmitted by DWC_ether_qos without errors and with length greater than the maxsize (1,518 or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in S2KP bit of the CONFIGURATION register).

11.7.6.1.206.1 Offset

Register	Offset
MAC_TX_OSIZE_PACKETS_GOOD	778h

11.7.6.1.206.2 Diagram



11.7.6.1.206.3 Fields

Field	Description
31-0 TXOSIZG	Tx OSize Packets Good This field indicates the number of packets transmitted without errors and with length greater than the maxsize (1,518 or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in S2KP bit of the CONFIGURATION register).

11.7.6.1.207 Good and Bad Packets Received (MAC_RX_PACKETS_COUNT_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos.

11.7.6.1.207.1 Offset

Register	Offset
MAC_RX_PACKETS_COUNT_GOOD_BAD	780h

11.7.6.1.207.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.207.3 Fields

Field	Description
31-0 RXPKTGB	Rx Packets Count Good Bad This field indicates the number of good and bad packets received.

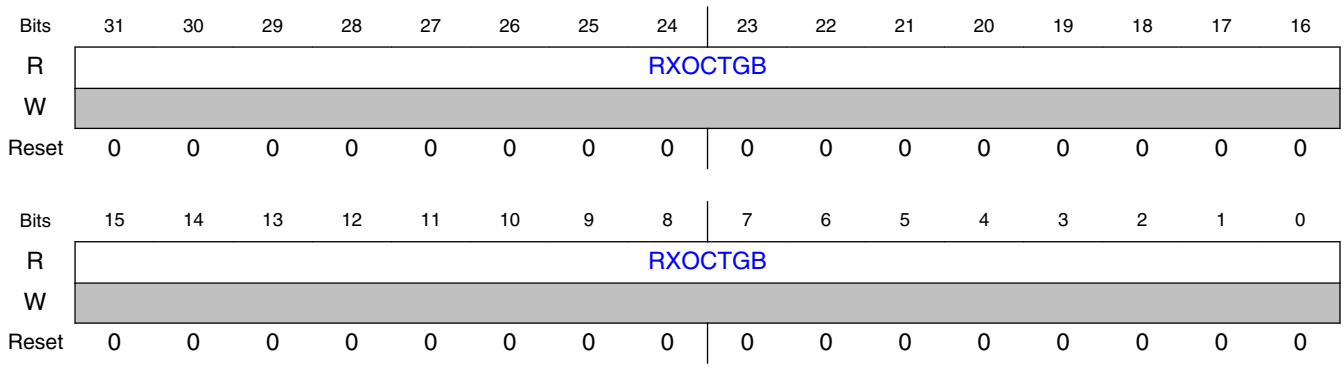
11.7.6.1.208 Bytes in Good and Bad Packets Received (MAC_RX_OCTET_COUNT_GOOD_BAD)

This register provides the number of bytes received by DWC_ther_qos, exclusive of preamble, in good and bad packets.

11.7.6.1.208.1 Offset

Register	Offset
MAC_RX_OCTET_COUNT_GOOD_BAD	784h

11.7.6.1.208.2 Diagram



11.7.6.1.208.3 Fields

Field	Description
31-0 RXOCTGB	Rx Octet Count Good Bad This field indicates the number of bytes received, exclusive of preamble, in good and bad packets.

11.7.6.1.209 Bytes in Good Packets Received (MAC_RX_OCTET_COUNT_GOOD)

This register provides the number of bytes received by DWC_ether_qos, exclusive of preamble, only in good packets.

11.7.6.1.209.1 Offset

Register	Offset
MAC_RX_OCTET_COUNT_GOOD	788h

11.7.6.1.209.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXOCTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXOCTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.209.3 Fields

Field	Description
31-0 RXOCTG	Rx Octet Count Good This field indicates the number of bytes received, exclusive of preamble, only in good packets.

11.7.6.1.210 Good Broadcast Packets Received (MAC_RX_BROADCAST_PACKETS_GOOD)

This register provides the number of good broadcast packets received by DWC_ether_qos.

11.7.6.1.210.1 Offset

Register	Offset
MAC_RX_BROADCAST_PACKETS_GOOD	78Ch

11.7.6.1.210.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXBCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXBCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.210.3 Fields

Field	Description
31-0 RXBCASTG	Rx Broadcast Packets Good This field indicates the number of good broadcast packets received.

11.7.6.1.211 Good Multicast Packets Received (MAC_RX_MULTICAST_PACKETS_GOOD)

This register provides the number of good multicast packets received by DWC_ether_qos.

11.7.6.1.211.1 Offset

Register	Offset
MAC_RX_MULTICAST_ PACKETS_GOOD	790h

11.7.6.1.211.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXMCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXMCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.211.3 Fields

Field	Description
31-0 RXMCASTG	Rx Multicast Packets Good This field indicates the number of good multicast packets received.

11.7.6.1.212 CRC Error Packets Received (MAC_RX_CRC_ERROR_PACKETS)

This register provides the number of packets received by DWC_ether_qos with CRC error.

11.7.6.1.212.1 Offset

Register	Offset
MAC_RX_CRC_ERROR_PACKETS	794h

11.7.6.1.212.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXCRCERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXCRCERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.212.3 Fields

Field	Description
31-0 RXCRCERR	Rx CRC Error Packets This field indicates the number of packets received with CRC error.

11.7.6.1.213 Alignment Error Packets Received (MAC_RX_ALIGNMENT_ERROR_PACKETS)

This register provides the number of packets received by DWC_ether_qos with alignment (dribble) error.

This register provides the number of packets received by DWC_ether_qos with alignment (dribble) error. It is valid only in 10/100 mode.

11.7.6.1.213.1 Offset

Register	Offset
MAC_RX_ALIGNMENT_ERROR_PACKETS	798h

11.7.6.1.213.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXALGNERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXALGNERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.213.3 Fields

Field	Description
31-0 RXALGNERR	Rx Alignment Error Packets This field indicates the number of packets received with alignment (dribble) error. Rx Alignment Error Packets This field indicates the number of packets received with alignment (dribble) error. It is valid only in 10/100 mode.

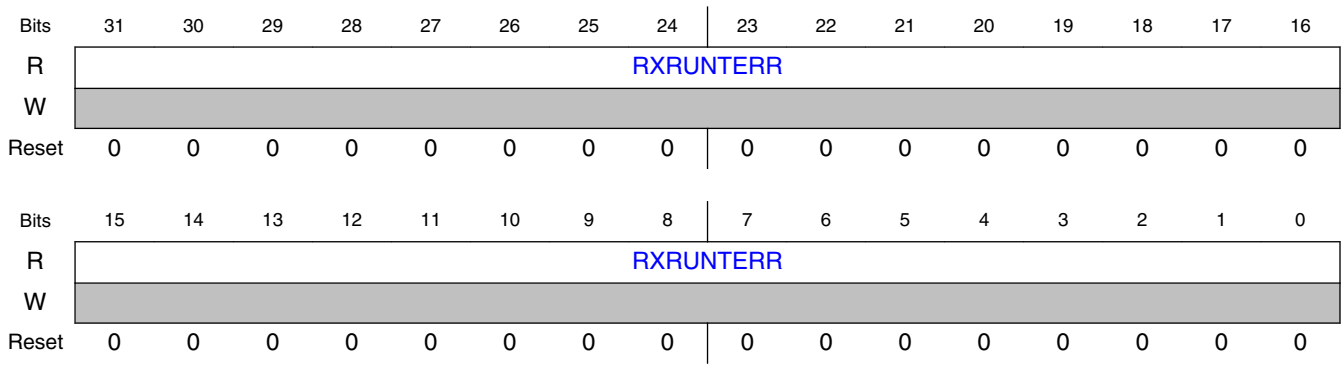
11.7.6.1.214 Runt Error Packets Received (MAC_RX_RUNT_ERROR_PACKETS)

This register provides the number of packets received by DWC_ether_qos with runt (length less than 64 bytes and CRC error) error.

11.7.6.1.214.1 Offset

Register	Offset
MAC_RX_RUNT_ERROR_PACKETS	79Ch

11.7.6.1.214.2 Diagram



11.7.6.1.214.3 Fields

Field	Description
31-0 RXRUNTERR	Rx Runt Error Packets This field indicates the number of packets received with runt (length less than 64 bytes and CRC error) error.

11.7.6.1.215 Jabber Error Packets Received (MAC_RX_JABBER_ERROR_PACKETS)

This register provides the number of giant packets received by DWC_ether_qos with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error.

This register provides the number of giant packets received by DWC_ether_qos with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. If Jumbo Packet mode is enabled, packets of length greater than 9,018 bytes (9,022 bytes for VLAN tagged) are considered as giant packets.

11.7.6.1.215.1 Offset

Register	Offset
MAC_RX_JABBER_ERROR_PACKETS	7A0h

11.7.6.1.215.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXJABERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXJABERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.215.3 Fields

Field	Description
31-0 RXJABERR	Rx Jabber Error Packets This field indicates the number of giant packets received with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. Rx Jabber Error Packets This field indicates the number of giant packets received with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. If Jumbo Packet mode is enabled, packets of length greater than 9,018 bytes (9,022 bytes for VLAN tagged) are considered as giant packets.

11.7.6.1.216 Good Undersize Packets Received (MAC_RX_UNDERSIZE_PACKETS_GOOD)

This register provides the number of packets received by DWC_ether_qos with length less than 64 bytes, without any errors.

11.7.6.1.216.1 Offset

Register	Offset
MAC_RX_UNDERSIZE_ PACKETS_GOOD	7A4h

11.7.6.1.216.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXUNDERSZG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXUNDERSZG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.216.3 Fields

Field	Description
31-0 RXUNDERSZG	Rx Undersize Packets Good This field indicates the number of packets received with length less than 64 bytes, without any errors.

11.7.6.1.217 Good Oversize Packets Received (MAC_RX_OVERSIZE_PACKETS_GOOD)

This register provides the number of packets received by DWC_ether_qos without errors, with length greater than the maxsize (1,518 bytes or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in the S2KP bit of the MAC_CONFIGURATION register).

11.7.6.1.217.1 Offset

Register	Offset
MAC_RX_OVERSIZE_PACKETS_GOOD	7A8h

11.7.6.1.217.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXOVERSZG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXOVERSZG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.217.3 Fields

Field	Description
31-0 RXOVERSZG	Rx Oversize Packets Good This field indicates the number of packets received without errors, with length greater than the maxsize (1,518 bytes or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in the S2KP bit of the MAC_CONFIGURATION register).

11.7.6.1.218 Good and Bad 64-Byte Packets Received (MAC_RX_64OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos with length 64 bytes, exclusive of the preamble.

11.7.6.1.218.1 Offset

Register	Offset
MAC_RX_64OCTETS_PACKETS_GOOD_BAD	7ACh

11.7.6.1.218.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX64OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX64OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.218.3 Fields

Field	Description
31-0 RX64OCTGB	Rx 64 Octets Packets Good Bad This field indicates the number of good and bad packets received with length 64 bytes, exclusive of the preamble.

11.7.6.1.219 Good and Bad 64-to-127 Byte Packets Received (MAC_RX_65TO127OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos with length between 65 and 127 (inclusive) bytes, exclusive of the preamble.

11.7.6.1.219.1 Offset

Register	Offset
MAC_RX_65TO127OCTETS_PACKETS_GOOD_BAD	7B0h

11.7.6.1.219.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX65_127OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX65_127OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.219.3 Fields

Field	Description
31-0 RX65_127OCTGB	Rx 65-127 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 65 and 127 (inclusive) bytes, exclusive of the preamble.

11.7.6.1.220 Good and Bad 128-to-255 Byte Packets Received (MAC_RX_128TO255OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos with length between 128 and 255 (inclusive) bytes, exclusive of the preamble.

11.7.6.1.220.1 Offset

Register	Offset
MAC_RX_128TO255OCTETS_PACKETS_GOOD_BAD	7B4h

11.7.6.1.220.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX128_255OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX128_255OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.220.3 Fields

Field	Description
31-0 RX128_255OCTGB	Rx 128-255 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 128 and 255 (inclusive) bytes, exclusive of the preamble.

11.7.6.1.221 Good and Bad 256-to-511 Byte Packets Received (MAC_RX_256TO511OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos with length between 256 and 511 (inclusive) bytes, exclusive of the preamble.

11.7.6.1.221.1 Offset

Register	Offset
MAC_RX_256TO511OCTETS_PACKETS_GOOD_BAD	7B8h

11.7.6.1.221.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX256_511OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX256_511OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.221.3 Fields

Field	Description
31-0 RX256_511OCTGB	Rx 256-511 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 256 and 511 (inclusive) bytes, exclusive of the preamble.

11.7.6.1.222 Good and Bad 512-to-1023 Byte Packets Received (MAC_RX_512TO1023OCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos with length between 512 and 1023 (inclusive) bytes, exclusive of the preamble.

11.7.6.1.222.1 Offset

Register	Offset
MAC_RX_512TO1023OCTETS_PACKETS_GOOD_BAD	7BCh

11.7.6.1.222.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX512_1023OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX512_1023OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.222.3 Fields

Field	Description
31-0 RX512_1023OCTGB	RX 512-1023 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 512 and 1023 (inclusive) bytes, exclusive of the preamble.

11.7.6.1.223 Good and Bad 1024-to-Max Byte Packets Received (MAC_RX_1024TOMAXOCTETS_PACKETS_GOOD_BAD)

This register provides the number of good and bad packets received by DWC_ether_qos with length between 1024 and maxsize (inclusive) bytes, exclusive of the preamble.

11.7.6.1.223.1 Offset

Register	Offset
MAC_RX_1024TOMAXOCTETS_PACKETS_GOOD_BAD	7C0h

11.7.6.1.223.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX1024_MAXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX1024_MAXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.223.3 Fields

Field	Description
31-0 RX1024_MAXOCTGB	Rx 1024-Max Octets Good Bad This field indicates the number of good and bad packets received with length between 1024 and maxsize (inclusive) bytes, exclusive of the preamble.

11.7.6.1.224 Good Unicast Packets Received (MAC_RX_UNICAST_PACKETS_GOOD)

This register provides the number of good unicast packets received by DWC_ether_qos.

11.7.6.1.224.1 Offset

Register	Offset
MAC_RX_UNICAST_PACKETS_GOOD	7C4h

11.7.6.1.224.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXUCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXUCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.224.3 Fields

Field	Description
31-0 RXUCASTG	Rx Unicast Packets Good This field indicates the number of good unicast packets received.

11.7.6.1.225 Length Error Packets Received (MAC_RX_LENGTH_ERROR_PACKETS)

This register provides the number of packets received by DWC_ether_qos with length error (Length Type field not equal to packet size), for all packets with valid length field.

11.7.6.1.225.1 Offset

Register	Offset
MAC_RX_LENGTH_ERROR_PACKETS	7C8h

11.7.6.1.225.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXLENERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXLENERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.225.3 Fields

Field	Description
31-0 RXLENERR	Rx Length Error Packets This field indicates the number of packets received with length error (Length Type field not equal to packet size), for all packets with valid length field.

11.7.6.1.226 Out-of-range Type Packets Received (MAC_RX_OUT_OF_RANGE_TYPE_PACKETS)

This register provides the number of packets received by DWC_ether_qos with length field not equal to the valid packet size (greater than 1,500 but less than 1,536).

11.7.6.1.226.1 Offset

Register	Offset
MAC_RX_OUT_OF_RANGE_TYPE_PACKETS	7CCh

11.7.6.1.226.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXOUTOFRNG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXOUTOFRNG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.226.3 Fields

Field	Description
31-0 RXOUTOFRNG	Rx Out of Range Type Packet This field indicates the number of packets received with length field not equal to the valid packet size (greater than 1,500 but less than 1,536).

11.7.6.1.227 Pause Packets Received (MAC_RX_PAUSE_PACKETS)

This register provides the number of good and valid Pause packets received by DWC_ether_qos.

11.7.6.1.227.1 Offset

Register	Offset
MAC_RX_PAUSE_PACKETS	7D0h

11.7.6.1.227.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXPAUSEPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXPAUSEPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.227.3 Fields

Field	Description
31-0 RXPAUSEPKT	Rx Pause Packets This field indicates the number of good and valid Pause packets received.

11.7.6.1.228 Missed Packets Due to FIFO Overflow (MAC_RX_FIFO_OVERFLOW_PACKETS)

This register provides the number of missed received packets because of FIFO overflow in DWC_ether_qos.

11.7.6.1.228.1 Offset

Register	Offset
MAC_RX_FIFO_OVERFLOW_PACKETS	7D4h

11.7.6.1.228.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXFIFOOVFL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXFIFOOVFL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.228.3 Fields

Field	Description
31-0 RXFIFOOVFL	Rx FIFO Overflow Packets This field indicates the number of missed received packets because of FIFO overflow.

11.7.6.1.229 Good and Bad VLAN Packets Received (MAC_RX_VLAN_PACKETS_GOOD_BAD)

This register provides the number of good and bad VLAN packets received by DWC_ether_qos.

11.7.6.1.229.1 Offset

Register	Offset
MAC_RX_VLAN_PACKETS_GOOD_BAD	7D8h

11.7.6.1.229.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXVLANPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXVLANPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.229.3 Fields

Field	Description
31-0 RXVLANPKTGB	Rx VLAN Packets Good Bad This field indicates the number of good and bad VLAN packets received.

11.7.6.1.230 Watchdog Error Packets Received (MAC_RX_WATCHDOG_ERROR_PACKETS)

This register provides the number of packets received by DWC_ether_qos with error because of watchdog timeout error (packets with a data load larger than 2,048 bytes (when JE and WD bits are reset in MAC_CONFIGURATION register), 10,240 bytes (when JE bit is set and WD bit is reset in MAC_CONFIGURATION register), 16,384 bytes (when WD bit is set in MAC_CONFIGURATION register) or the value programmed in the MAC_WATCHDOG_TIMEOUT register).

11.7.6.1.230.1 Offset

Register	Offset
MAC_RX_WATCHDOG_ERROR_PACKETS	7DCh

11.7.6.1.230.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXWDGERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXWDGERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.230.3 Fields

Field	Description
31-0 RXWDGERR	Rx Watchdog Error Packets This field indicates the number of packets received with error because of watchdog timeout error (packets with a data load larger than 2,048 bytes (when JE and WD bits are reset in MAC_CONFIGURATION register), 10,240 bytes (when JE bit is set and WD bit is reset in MAC_CONFIGURATION register), 16,384 bytes (when WD bit is set in MAC_CONFIGURATION register) or the value programmed in the MAC_WATCHDOG_TIMEOUT register).

11.7.6.1.231 Receive Error Packets Received (MAC_RX_RECEIVE_ERROR_PACKETS)

This register provides the number of packets received by DWC_ether_qos with Receive error or Packet Extension error on the GMII interface.

11.7.6.1.231.1 Offset

Register	Offset
MAC_RX_RECEIVE_ERROR_PACKETS	7E0h

11.7.6.1.231.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXRCVERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXRCVERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.231.3 Fields

Field	Description
31-0 RXRCVERR	Rx Receive Error Packets This field indicates the number of packets received with Receive error or Packet Extension error on the GMII interface.

11.7.6.1.232 Good Control Packets Received (MAC_RX_CONTROL_PACKETS_GOOD)

This register provides the number of good control packets received by DWC_ether_qos.

11.7.6.1.232.1 Offset

Register	Offset
MAC_RX_CONTROL_PACKETS_GOOD	7E4h

11.7.6.1.232.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXCTRLG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXCTRLG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.232.3 Fields

Field	Description
31-0 RXCTRLG	Rx Control Packets Good This field indicates the number of good control packets received.

11.7.6.1.233 Microseconds Tx LPI Asserted (MAC_TX_LPI_USEC_CNTR)

This register provides the number of microseconds Tx LPI is asserted by DWC_ether_qos.

11.7.6.1.233.1 Offset

Register	Offset
MAC_TX_LPI_USEC_CNTR	7ECh

11.7.6.1.233.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXLPIUSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXLPIUSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.233.3 Fields

Field	Description
31-0 TXLPIUSC	Tx LPI Microseconds Counter This field indicates the number of microseconds Tx LPI is asserted. Tx LPI Microseconds Counter This field indicates the number of microseconds Tx LPI is asserted. For every Tx LPI Entry and Exit, the Timer value can have an error of +/- 1 microsecond.

11.7.6.1.234 Number of Times Tx LPI Asserted (MAC_TX_LPI_TRAN_CNTR)

This register provides the number of times DWC_ether_qos has entered Tx LPI.

11.7.6.1.234.1 Offset

Register	Offset
MAC_TX_LPI_TRAN_CNTR	7F0h

11.7.6.1.234.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXLPITRC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXLPITRC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.234.3 Fields

Field	Description
31-0 TXLPITRC	Tx LPI Transition counter This field indicates the number of times Tx LPI Entry has occurred. Tx LPI Transition counter

Field	Description
	This field indicates the number of times Tx LPI Entry has occurred. Even if Tx LPI Entry occurs in Automate Mode (because of LPITXA bit set in the LPI Control and Status register), the counter will increment.

11.7.6.1.235 Microseconds Rx LPI Sampled (MAC_RX_LPI_USEC_CNTR)

This register provides the number of microseconds Rx LPI is sampled by DWC_ether_qos.

11.7.6.1.235.1 Offset

Register	Offset
MAC_RX_LPI_USEC_CNTR	7F4h

11.7.6.1.235.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXLPIUSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXLPIUSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.235.3 Fields

Field	Description
31-0 RXLPIUSC	Rx LPI Microseconds Counter This field indicates the number of microseconds Rx LPI is asserted. Rx LPI Microseconds Counter This field indicates the number of microseconds Rx LPI is asserted. For every Rx LPI Entry and Exit, the Timer value can have an error of +/- 1 microsecond.

11.7.6.1.236 Number of Times Rx LPI Entered (MAC_RX_LPI_TRAN_CNTR)

This register provides the number of times DWC_ether_qos has entered Rx LPI.

11.7.6.1.236.1 Offset

Register	Offset
MAC_RX_LPI_TRAN_CNTR	7F8h

11.7.6.1.236.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXLPITRC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXLPITRC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.236.3 Fields

Field	Description
31-0 RXLPITRC	Rx LPI Transition counter This field indicates the number of times Rx LPI Entry has occurred.

11.7.6.1.237 MMC IPC Receive Interrupt Mask (MAC_MMC_IPC_RX_INTERRUPT_MASK)

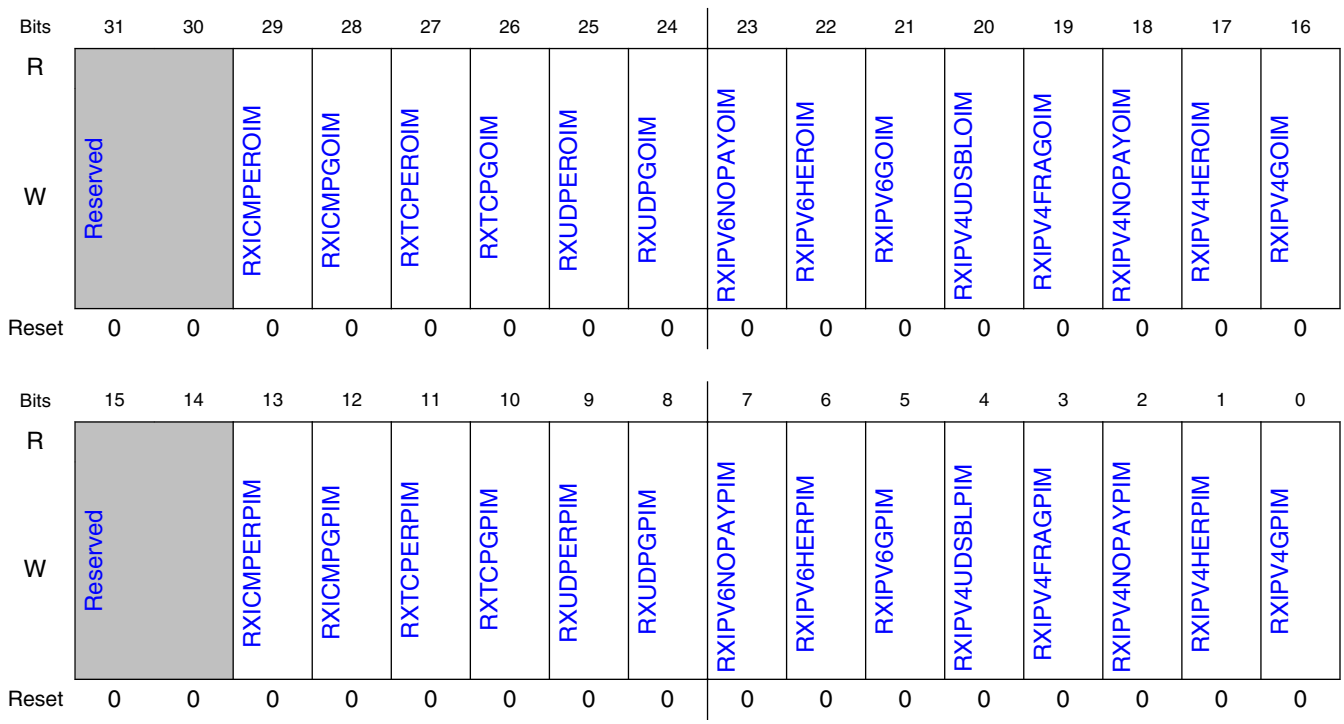
This register maintains the mask for the interrupt generated from the receive IPC statistic counters.

The MMC Receive Checksum Off load Interrupt Mask register maintains the masks for the interrupts generated when the receive IPC (Checksum Off load) statistic counters reach half their maximum value, and when they reach their maximum values. This register is 32 bits wide.

11.7.6.1.237.1 Offset

Register	Offset
MAC_MMC_IPC_RX_INTERRUPT_MASK	800h

11.7.6.1.237.2 Diagram



11.7.6.1.237.3 Fields

Field	Description
31-30 —	Reserved
29 RXICMPEROIM	MMC Receive ICMP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive ICMP Error Octet Counter Interrupt Mask is disabled

Table continues on the next page...

Field	Description
	1 - MMC Receive ICMP Error Octet Counter Interrupt Mask is enabled
28 RXICMPGOIM	MMC Receive ICMP Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_gd_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive ICMP Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive ICMP Good Octet Counter Interrupt Mask is enabled
27 RXTCPEROIM	MMC Receive TCP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive TCP Error Octet Counter Interrupt Mask is disabled 1 - MMC Receive TCP Error Octet Counter Interrupt Mask is enabled
26 RXTCPGOIM	MMC Receive TCP Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_gd_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive TCP Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive TCP Good Octet Counter Interrupt Mask is enabled
25 RXUDPEROIM	MMC Receive UDP Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive UDP Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive UDP Good Octet Counter Interrupt Mask is enabled
24 RXUDPGOIM	MMC Receive IPV6 No Payload Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_gd_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 No Payload Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 No Payload Octet Counter Interrupt Mask is enabled
23 RXIPV6NOPAYOIM	MMC Receive IPV6 Header Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_nopay_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 Header Error Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 Header Error Octet Counter Interrupt Mask is enabled
22 RXIPV6HEROIM	MMC Receive IPV6 Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_hdrerr_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 Good Octet Counter Interrupt Mask is enabled
21 RXIPV6GOIM	MMC Receive IPV6 Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_gd_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 Good Octet Counter Interrupt Mask is enabled
20 RXIPV4UDSBLOIM	MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_udsbl_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Mask is enabled
19 RXIPV4FRAGOIM	MMC Receive IPV4 Fragmented Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_frag_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 Fragmented Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 Fragmented Octet Counter Interrupt Mask is enabled

Table continues on the next page...

Field	Description
18 RXIPV4NOPAYOIM	MMC Receive IPV4 No Payload Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_nopay_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 No Payload Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 No Payload Octet Counter Interrupt Mask is enabled
17 RXIPV4HEROIM	MMC Receive IPV4 Header Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 Header Error Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 Header Error Octet Counter Interrupt Mask is enabled
16 RXIPV4GOIM	MMC Receive IPV4 Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_gd_octets counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 Good Octet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 Good Octet Counter Interrupt Mask is enabled
15-14 —	Reserved
13 RXICMPERPIM	MMC Receive ICMP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive ICMP Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive ICMP Error Packet Counter Interrupt Mask is enabled
12 RXICMPGPIM	MMC Receive ICMP Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_gd_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive ICMP Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive ICMP Good Packet Counter Interrupt Mask is enabled
11 RXTCPERPIM	MMC Receive TCP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive TCP Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive TCP Error Packet Counter Interrupt Mask is enabled
10 RXTCPGPIM	MMC Receive TCP Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_gd_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive TCP Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive TCP Good Packet Counter Interrupt Mask is enabled
9 RXUDPERPIM	MMC Receive UDP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive UDP Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive UDP Error Packet Counter Interrupt Mask is enabled
8 RXUDPGPIM	MMC Receive UDP Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_gd_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive UDP Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive UDP Good Packet Counter Interrupt Mask is enabled
7 RXIPV6NOPAYPIM	MMC Receive IPV6 No Payload Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_nopay_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 No Payload Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 No Payload Packet Counter Interrupt Mask is enabled

Table continues on the next page...

Field	Description
6 RXIPV6HERPIM	MMC Receive IPV6 Header Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 Header Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 Header Error Packet Counter Interrupt Mask is enabled
5 RXIPV6GPIM	MMC Receive IPV6 Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV6 Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV6 Good Packet Counter Interrupt Mask is enabled
4 RXIPV4UDSBLPIM	MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_udsbl_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Mask is enabled
3 RXIPV4FRAGPIM	MMC Receive IPV4 Fragmented Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_frag_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 Fragmented Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 Fragmented Packet Counter Interrupt Mask is enabled
2 RXIPV4NOPAYPIM	MMC Receive IPV4 No Payload Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_nopay_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 No Payload Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 No Payload Packet Counter Interrupt Mask is enabled
1 RXIPV4HERPIM	MMC Receive IPV4 Header Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 Header Error Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 Header Error Packet Counter Interrupt Mask is enabled
0 RXIPV4GPIM	MMC Receive IPV4 Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 Good Packet Counter Interrupt Mask is disabled 1 - MMC Receive IPV4 Good Packet Counter Interrupt Mask is enabled

11.7.6.1.238 MMC IPC Receive Interrupt (MAC_MMC_IPC_RX_INTERRUPT)

This register maintains the interrupt that the receive IPC statistic counters generate.

The MMC Receive Checksum Offload Interrupt register maintains the interrupts generated when receive IPC statistic counters reach half their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32 bit counter and 0xFFFF for 16 bit counter). When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones.

The MMC Receive Checksum Offload Interrupt register is 32 bit wide. When the MMC IPC counter that caused the interrupt is read, its corresponding interrupt bit is cleared. The counter's least-significant byte lane (Bits[7:0]) must be read to clear the interrupt bit.

11.7.6.1.238.1 Offset

Register	Offset
MAC_MMC_IPC_RX_INTERRUPT	808h

11.7.6.1.238.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		RXICMPEROIS	RXICMPGOIS	RXTCPEROIS	RXTCPGOIS	RXUDPEROIS	RXUDPGOIS	RXIPV6NOPAYOIS	RXIPV6HEROIS	RXIPV6GOIS	RXIPV4UDSBLOIS	RXIPV4FRAGOIS	RXIPV4NOPAYOIS	RXIPV4HEROIS	RXIPV4GOIS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		RXICMPERPIS	RXICMPGPIS	RXTCPERPIS	RXTCPGPIS	RXUDPERPIS	RXUDPGPIS	RXIPV6NOPAYPIS	RXIPV6HERPIS	RXIPV6GPIS	RXIPV4UDSBLPIS	RXIPV4FRAGPIS	RXIPV4NOPAYPIS	RXIPV4HERPIS	RXIPV4GPIS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.238.3 Fields

Field	Description
31-30 —	Reserved
29 RXICMPEROIS	MMC Receive ICMP Error Octet Counter Interrupt Status This bit is set when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Field	Description
	MMC Receive ICMP Error Octet Counter Interrupt Status This bit is set when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive ICMP Error Octet Counter Interrupt Status not detected 1 - MMC Receive ICMP Error Octet Counter Interrupt Status detected
28 RXICMPGOIS	MMC Receive ICMP Good Octet Counter Interrupt Status This bit is set when the rxicmp_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive ICMP Good Octet Counter Interrupt Status This bit is set when the rxicmp_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive ICMP Good Octet Counter Interrupt Status not detected 1 - MMC Receive ICMP Good Octet Counter Interrupt Status detected
27 RXTCPEROIS	MMC Receive TCP Error Octet Counter Interrupt Status This bit is set when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. MMC Receive TCP Error Octet Counter Interrupt Status This bit is set when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive TCP Error Octet Counter Interrupt Status not detected 1 - MMC Receive TCP Error Octet Counter Interrupt Status detected
26 RXTCPGOIS	MMC Receive TCP Good Octet Counter Interrupt Status This bit is set when the rxtcp_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive TCP Good Octet Counter Interrupt Status This bit is set when the rxtcp_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive TCP Good Octet Counter Interrupt Status not detected 1 - MMC Receive TCP Good Octet Counter Interrupt Status detected
25 RXUDPEROIS	MMC Receive UDP Error Octet Counter Interrupt Status This bit is set when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. MMC Receive UDP Error Octet Counter Interrupt Status This bit is set when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive UDP Error Octet Counter Interrupt Status not detected 1 - MMC Receive UDP Error Octet Counter Interrupt Status detected
24 RXUDPGOIS	MMC Receive UDP Good Octet Counter Interrupt Status This bit is set when the rxudp_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive UDP Good Octet Counter Interrupt Status

Table continues on the next page...

Field	Description
	This bit is set when the rxudp_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive UDP Good Octet Counter Interrupt Status not detected 1 - MMC Receive UDP Good Octet Counter Interrupt Status detected
23 RXIPV6NOPAYOIS	MMC Receive IPV6 No Payload Octet Counter Interrupt Status This bit is set when the rxipv6_nopay_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 No Payload Octet Counter Interrupt Status This bit is set when the rxipv6_nopay_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV6 No Payload Octet Counter Interrupt Status not detected 1 - MMC Receive IPV6 No Payload Octet Counter Interrupt Status detected
22 RXIPV6HEROIS	MMC Receive IPV6 Header Error Octet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 Header Error Octet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV6 Header Error Octet Counter Interrupt Status not detected 1 - MMC Receive IPV6 Header Error Octet Counter Interrupt Status detected
21 RXIPV6GOIS	MMC Receive IPV6 Good Octet Counter Interrupt Status This bit is set when the rxipv6_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 Good Octet Counter Interrupt Status This bit is set when the rxipv6_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV6 Good Octet Counter Interrupt Status not detected 1 - MMC Receive IPV6 Good Octet Counter Interrupt Status detected
20 RXIPV4UDSBLOIS	MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status This bit is set when the rxipv4_udsbl_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status This bit is set when the rxipv4_udsbl_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status not detected 1 - MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status detected
19 RXIPV4FRAGOIS	MMC Receive IPV4 Fragmented Octet Counter Interrupt Status This bit is set when the rxipv4_frag_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Fragmented Octet Counter Interrupt Status

Table continues on the next page...

Field	Description
	This bit is set when the rxipv4_frag_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 Fragmented Octet Counter Interrupt Status not detected 1 - MMC Receive IPV4 Fragmented Octet Counter Interrupt Status detected
18 RXIPV4NOPAYOIS	MMC Receive IPV4 No Payload Octet Counter Interrupt Status This bit is set when the rxipv4_nopay_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 No Payload Octet Counter Interrupt Status This bit is set when the rxipv4_nopay_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 No Payload Octet Counter Interrupt Status not detected 1 - MMC Receive IPV4 No Payload Octet Counter Interrupt Status detected
17 RXIPV4HEROIS	MMC Receive IPV4 Header Error Octet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Header Error Octet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 Header Error Octet Counter Interrupt Status not detected 1 - MMC Receive IPV4 Header Error Octet Counter Interrupt Status detected
16 RXIPV4GOIS	MMC Receive IPV4 Good Octet Counter Interrupt Status This bit is set when the rxipv4_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Good Octet Counter Interrupt Status This bit is set when the rxipv4_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 Good Octet Counter Interrupt Status not detected 1 - MMC Receive IPV4 Good Octet Counter Interrupt Status detected
15-14 —	Reserved
13 RXICMPERPIS	MMC Receive ICMP Error Packet Counter Interrupt Status This bit is set when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. MMC Receive ICMP Error Packet Counter Interrupt Status This bit is set when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive ICMP Error Packet Counter Interrupt Status not detected 1 - MMC Receive ICMP Error Packet Counter Interrupt Status detected
12 RXICMPGPIS	MMC Receive ICMP Good Packet Counter Interrupt Status This bit is set when the rxicmp_gd_pkts counter reaches half of the maximum value or the maximum value. MMC Receive ICMP Good Packet Counter Interrupt Status

Table continues on the next page...

Field	Description
	This bit is set when the rxicmp_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive ICMP Good Packet Counter Interrupt Status not detected 1 - MMC Receive ICMP Good Packet Counter Interrupt Status detected
11 RXTCPERPIS	MMC Receive TCP Error Packet Counter Interrupt Status This bit is set when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. MMC Receive TCP Error Packet Counter Interrupt Status This bit is set when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive TCP Error Packet Counter Interrupt Status not detected 1 - MMC Receive TCP Error Packet Counter Interrupt Status detected
10 RXTCPGPIS	MMC Receive TCP Good Packet Counter Interrupt Status This bit is set when the rxtcp_gd_pkts counter reaches half of the maximum value or the maximum value. MMC Receive TCP Good Packet Counter Interrupt Status This bit is set when the rxtcp_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive TCP Good Packet Counter Interrupt Status not detected 1 - MMC Receive TCP Good Packet Counter Interrupt Status detected
9 RXUDPERPIS	MMC Receive UDP Error Packet Counter Interrupt Status This bit is set when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. MMC Receive UDP Error Packet Counter Interrupt Status This bit is set when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive UDP Error Packet Counter Interrupt Status not detected 1 - MMC Receive UDP Error Packet Counter Interrupt Status detected
8 RXUDPGPIS	MC Receive UDP Good Packet Counter Interrupt Status This bit is set when the rxudp_gd_pkts counter reaches half of the maximum value or the maximum value. MC Receive UDP Good Packet Counter Interrupt Status This bit is set when the rxudp_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive UDP Good Packet Counter Interrupt Status not detected 1 - MMC Receive UDP Good Packet Counter Interrupt Status detected
7 RXIPV6NOPAYPIS	MMC Receive IPV6 No Payload Packet Counter Interrupt Status This bit is set when the rxipv6_nopay_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 No Payload Packet Counter Interrupt Status This bit is set when the rxipv6_nopay_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV6 No Payload Packet Counter Interrupt Status not detected

Table continues on the next page...

Field	Description
	1 - MMC Receive IPV6 No Payload Packet Counter Interrupt Status detected
6 RXIPV6HERPIS	MMC Receive IPV6 Header Error Packet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 Header Error Packet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV6 Header Error Packet Counter Interrupt Status not detected 1 - MMC Receive IPV6 Header Error Packet Counter Interrupt Status detected
5 RXIPV6GPIS	MMC Receive IPV6 Good Packet Counter Interrupt Status This bit is set when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 Good Packet Counter Interrupt Status This bit is set when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV6 Good Packet Counter Interrupt Status not detected 1 - MMC Receive IPV6 Good Packet Counter Interrupt Status detected
4 RXIPV4UDSBLPIS	MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Status This bit is set when the rxipv4_udsbl_pkts counter reaches half of the maximum value or the maximum value. 0 - MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Status not detected 1 - MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Status detected
3 RXIPV4FRAGPIS	MMC Receive IPV4 Fragmented Packet Counter Interrupt Status This bit is set when the rxipv4_frag_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Fragmented Packet Counter Interrupt Status This bit is set when the rxipv4_frag_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 Fragmented Packet Counter Interrupt Status not detected 1 - MMC Receive IPV4 Fragmented Packet Counter Interrupt Status detected
2 RXIPV4NOPAYPIS	MMC Receive IPV4 No Payload Packet Counter Interrupt Status This bit is set when the rxipv4_nopay_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 No Payload Packet Counter Interrupt Status This bit is set when the rxipv4_nopay_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 No Payload Packet Counter Interrupt Status not detected 1 - MMC Receive IPV4 No Payload Packet Counter Interrupt Status detected
1 RXIPV4HERPIS	MMC Receive IPV4 Header Error Packet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Header Error Packet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Field	Description
	Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 Header Error Packet Counter Interrupt Status not detected 1 - MMC Receive IPV4 Header Error Packet Counter Interrupt Status detected
0 RXIPV4GPIS	MMC Receive IPV4 Good Packet Counter Interrupt Status This bit is set when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Good Packet Counter Interrupt Status This bit is set when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - MMC Receive IPV4 Good Packet Counter Interrupt Status not detected 1 - MMC Receive IPV4 Good Packet Counter Interrupt Status detected

11.7.6.1.239 Good IPv4 Datagrams Received (MAC_RXIPV4_GOOD_PACKETS)

This register provides the number of good IPv4 datagrams received by DWC_ether_qos with the TCP, UDP, or ICMP payload.

11.7.6.1.239.1 Offset

Register	Offset
MAC_RXIPV4_GOOD_PACKETS	810h

11.7.6.1.239.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4GDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4GDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.239.3 Fields

Field	Description
31-0 RXIPV4GDPKT	RxIPv4 Good Packets This field indicates the number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload.

11.7.6.1.240 IPv4 Datagrams Received with Header Errors (MAC_RXIPV4_HEADER_ERROR_PACKETS)

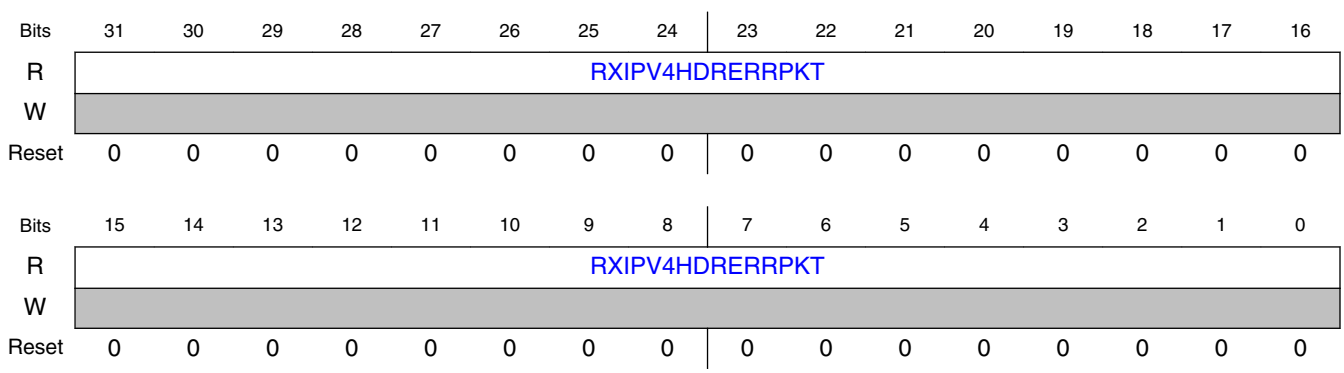
RxIPv4 Header Error Packets

This register provides the number of IPv4 datagrams received by DWC_ether_qos with header (checksum, length, or version mismatch) errors.

11.7.6.1.240.1 Offset

Register	Offset
MAC_RXIPV4_HEADER_ERROR_PACKETS	814h

11.7.6.1.240.2 Diagram



11.7.6.1.240.3 Fields

Field	Description
31-0 RXIPV4HDRERRPKT	RxIPv4 Header Error Packets This field indicates the number of IPv4 datagrams received with header (checksum, length, or version mismatch) errors.

11.7.6.1.241 IPv4 Datagrams Received with No Payload (MAC_RXIPV4_NO_PAYLOAD_PACKETS)

This register provides the number of IPv4 datagram packets received by DWC_ether_qos that did not have a TCP, UDP, or ICMP payload.

11.7.6.1.241.1 Offset

Register	Offset
MAC_RXIPV4_NO_PAYLOAD_PACKETS	818h

11.7.6.1.241.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4NOPAYPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4NOPAYPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.241.3 Fields

Field	Description
31-0 RXIPV4NOPAYPKT	RxIPv4 Payload Packets This field indicates the number of IPv4 datagram packets received that did not have a TCP, UDP, or ICMP payload.

11.7.6.1.242 IPv4 Datagrams Received with Fragmentation (MAC_RXIPV4_FRAGMENTED_PACKETS)

This register provides the number of good IPv4 datagrams received by DWC_ether_qos with fragmentation.

11.7.6.1.242.1 Offset

Register	Offset
MAC_RXIPV4_FRAGMENTED_PACKETS	81Ch

11.7.6.1.242.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4FRAGPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4FRAGPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.242.3 Fields

Field	Description
31-0 RXIPV4FRAGPKT	RxIPv4 Fragmented Packets This field indicates the number of good IPv4 datagrams received with fragmentation.

11.7.6.1.243 IPv4 Datagrams Received with UDP Checksum Disabled (MAC_RXIPV4_UDP_CHECKSUM_DISABLED_PACKETS)

This register provides the number of good IPv4 datagrams received by DWC_ether_qos that had a UDP payload with checksum disabled.

11.7.6.1.243.1 Offset

Register	Offset
MAC_RXIPV4_UDP_CHECKSUM_DISABLED_PACKETS	820h

11.7.6.1.243.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4UDSBLPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4UDSBLPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.243.3 Fields

Field	Description
31-0 RXIPV4UDSBLPKT	RxIPv4 UDP Checksum Disabled Packets This field indicates the number of good IPv4 datagrams received that had a UDP payload with checksum disabled.

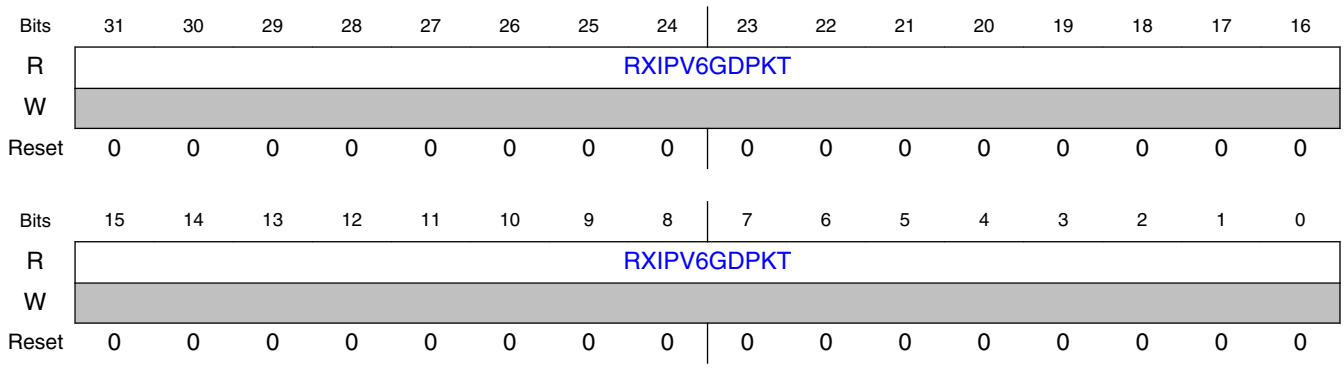
11.7.6.1.244 Good IPv6 Datagrams Received (MAC_RXIPV6_GOOD_PACKETS)

This register provides the number of good IPv6 datagrams received by DWC_ether_qos with the TCP, UDP, or ICMP payload.

11.7.6.1.244.1 Offset

Register	Offset
MAC_RXIPV6_GOOD_PACKETS	824h

11.7.6.1.244.2 Diagram



11.7.6.1.244.3 Fields

Field	Description
31-0 RXIPV6GDPKT	RxIPv6 Good Packets This field indicates the number of good IPv6 datagrams received with the TCP, UDP, or ICMP payload.

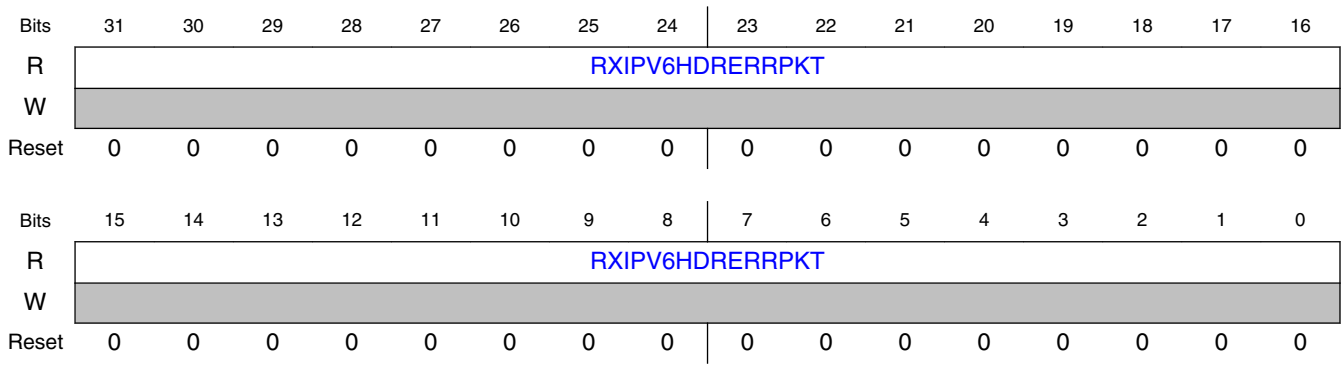
11.7.6.1.245 IPv6 Datagrams Received with Header Errors (MAC_RXIPV6_HEADER_ERROR_PACKETS)

This register provides the number of IPv6 datagrams received by DWC_ether_qos with header (length or version mismatch) errors.

11.7.6.1.245.1 Offset

Register	Offset
MAC_RXIPV6_HEADER_ERROR_PACKETS	828h

11.7.6.1.245.2 Diagram



11.7.6.1.245.3 Fields

Field	Description
31-0 RXIPV6HDRERRPKT	RxIPv6 Header Error Packets This field indicates the number of IPv6 datagrams received with header (length or version mismatch) errors.

11.7.6.1.246 IPv6 Datagrams Received with No Payload (MAC_RXIPV6_NO_PAYLOAD_PACKETS)

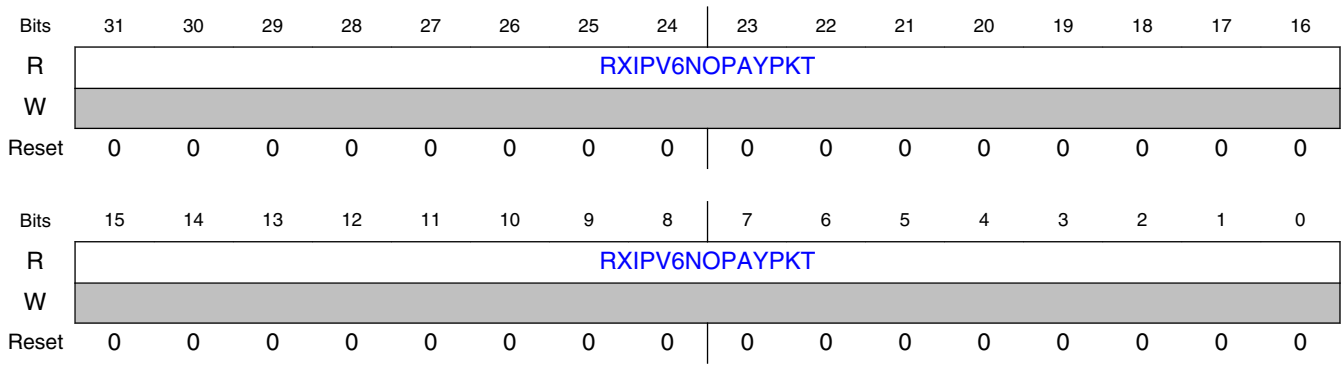
This register provides the number of IPv6 datagram packets received by DWC_ether_qos that did not have a TCP, UDP, or ICMP payload.

This register provides the number of IPv6 datagram packets received by DWC_ether_qos that did not have a TCP, UDP, or ICMP payload. This includes all IPv6 datagrams with fragmentation or security extension headers.

11.7.6.1.246.1 Offset

Register	Offset
MAC_RXIPV6_NO_PAYLOAD_PACKETS	82Ch

11.7.6.1.246.2 Diagram



11.7.6.1.246.3 Fields

Field	Description
31-0 RXIPV6NOPAYPKT	RxIPv6 Payload Packets This field indicates the number of IPv6 datagram packets received that did not have a TCP, UDP, or ICMP payload. RxIPv6 Payload Packets This field indicates the number of IPv6 datagram packets received that did not have a TCP, UDP, or ICMP payload. This includes all IPv6 datagrams with fragmentation or security extension headers.

11.7.6.1.247 IPv6 Datagrams Received with Good UDP (MAC_RXUDP_GOOD_PACKETS)

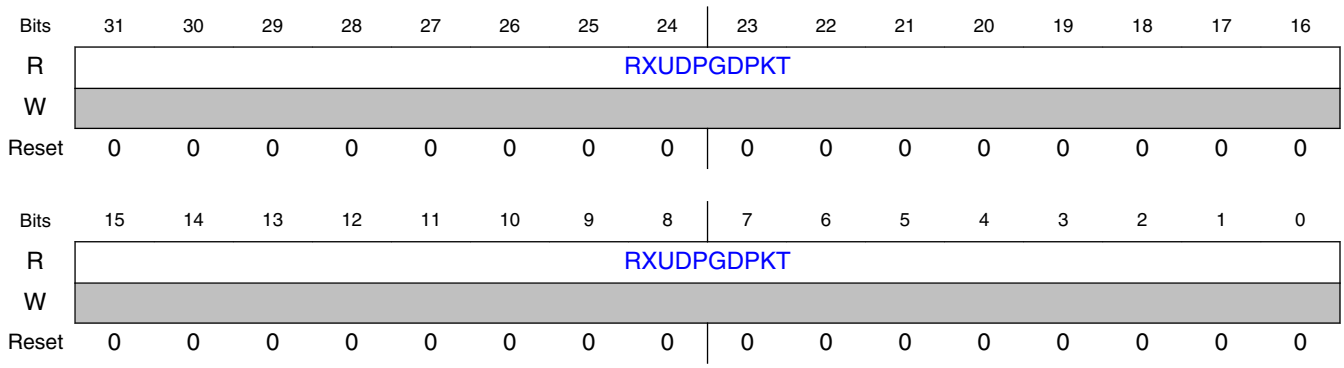
This register provides the number of good IP datagrams received by DWC_ether_qos with a good UDP payload.

This register provides the number of good IP datagrams received by DWC_ether_qos with a good UDP payload. This counter is not updated when the RxIPv4_UDP_Checksum_Disabled_Packets counter is incremented.

11.7.6.1.247.1 Offset

Register	Offset
MAC_RXUDP_GOOD_PACKETS	830h

11.7.6.1.247.2 Diagram



11.7.6.1.247.3 Fields

Field	Description
31-0 RXUDPGDPKT	RxUDP Good Packets This field indicates the number of good IP datagrams received with a good UDP payload. RxUDP Good Packets This field indicates the number of good IP datagrams received with a good UDP payload. This counter is not updated when the RxIPv4_UDP_Checksum_Disabled_Packets counter is incremented.

11.7.6.1.248 IPv6 Datagrams Received with UDP Checksum Error (MAC_RXUDP_ERROR_PACKETS)

This register provides the number of good IP datagrams received by DWC_ether_qos whose UDP payload has a checksum error.

11.7.6.1.248.1 Offset

Register	Offset
MAC_RXUDP_ERROR_PACKETS	834h

11.7.6.1.248.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXUDPERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXUDPERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.248.3 Fields

Field	Description
31-0 RXUDPERRPKT	RxUDP Error Packets This field indicates the number of good IP datagrams received whose UDP payload has a checksum error.

11.7.6.1.249 IPv6 Datagrams Received with Good TCP Payload (MAC_RXTCP_GOOD_PACKETS)

This register provides the number of good IP datagrams received by DWC_ether_qos with a good TCP payload.

11.7.6.1.249.1 Offset

Register	Offset
MAC_RXTCP_GOOD_PACKETS	838h

11.7.6.1.249.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXTCPGDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXTCPGDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.249.3 Fields

Field	Description
31-0 RXTCPGDPKT	RxTCP Good Packets This field indicates the number of good IP datagrams received with a good TCP payload.

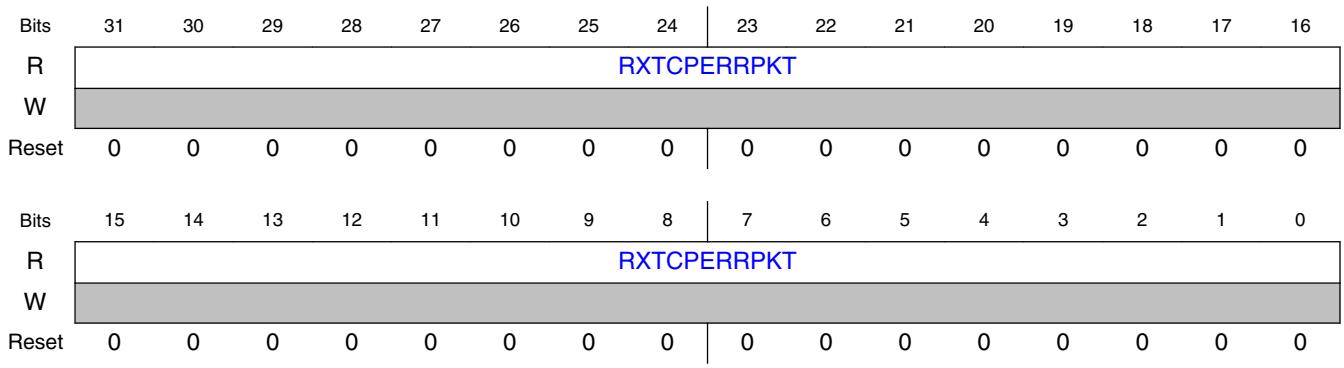
11.7.6.1.250 IPv6 Datagrams Received with TCP Checksum Error (MAC_RXTCP_ERROR_PACKETS)

This register provides the number of good IP datagrams received by DWC_ether_qos whose TCP payload has a checksum error.

11.7.6.1.250.1 Offset

Register	Offset
MAC_RXTCP_ERROR_PACKETS	83Ch

11.7.6.1.250.2 Diagram



11.7.6.1.250.3 Fields

Field	Description
31-0 RXTCPERRPKT	RxTCP Error Packets This field indicates the number of good IP datagrams received whose TCP payload has a checksum error.

11.7.6.1.251 IPv6 Datagrams Received with Good ICMP Payload (MAC_RXICMP_GOOD_PACKETS)

This register provides the number of good IP datagrams received by DWC_ether_qos with a good ICMP payload.

11.7.6.1.251.1 Offset

Register	Offset
MAC_RXICMP_GOOD_PACKETS	840h

11.7.6.1.251.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXICMPGDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXICMPGDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.251.3 Fields

Field	Description
31-0 RXICMPGDPKT	RxICMP Good Packets This field indicates the number of good IP datagrams received with a good ICMP payload.

11.7.6.1.252 IPv6 Datagrams Received with ICMP Checksum Error (MAC_RXICMP_ERROR_PACKETS)

This register provides the number of good IP datagrams received by DWC_ether_qos whose ICMP payload has a checksum error.

11.7.6.1.252.1 Offset

Register	Offset
MAC_RXICMP_ERROR_PACKETS	844h

11.7.6.1.252.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXICMPERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXICMPERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.252.3 Fields

Field	Description
31-0 RXICMPERRPKT	RxICMP Error Packets This field indicates the number of good IP datagrams received whose ICMP payload has a checksum error.

11.7.6.1.253 Good Bytes Received in IPv4 Datagrams (MAC_RXIPV4_GOOD_OCTETS)

This register provides the number of bytes received by DWC_ether_qos in good IPv4 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.253.1 Offset

Register	Offset
MAC_RXIPV4_GOOD_OCTETS	850h

11.7.6.1.253.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4GDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4GDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.253.3 Fields

Field	Description
31-0 RXIPV4GDOCT	RxIPv4 Good Octets This field indicates the number of bytes received in good IPv4 datagrams encapsulating TCP, UDP, or ICMP data. RxIPv4 Good Octets This field indicates the number of bytes received in good IPv4 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.254 Bytes Received in IPv4 Datagrams with Header Errors (MAC_RXIPV4_HEADER_ERROR_OCTETS)

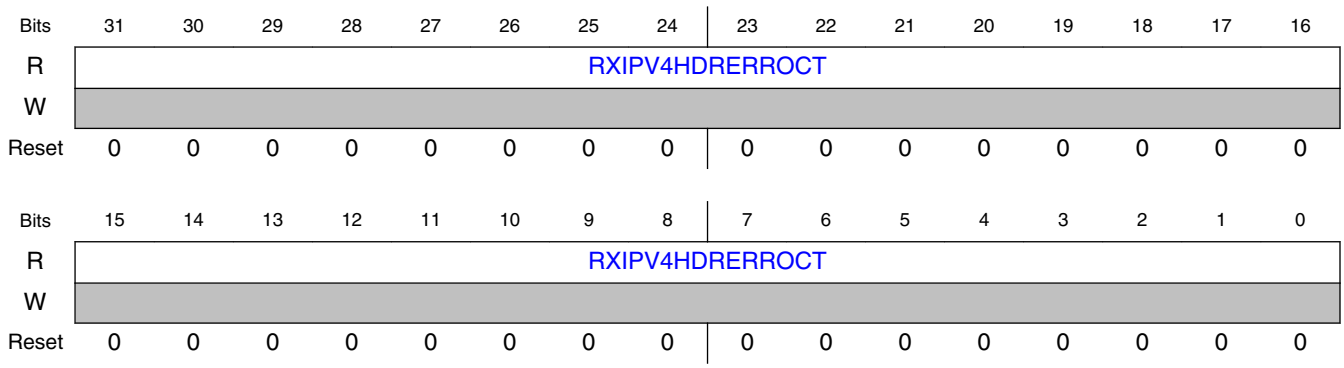
This register provides the number of bytes received by DWC_ether_qos in IPv4 datagrams with header errors (checksum, length, version mismatch).

This register provides the number of bytes received by DWC_ether_qos in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.254.1 Offset

Register	Offset
MAC_RXIPV4_HEADER_ERROR_OCTETS	854h

11.7.6.1.254.2 Diagram



11.7.6.1.254.3 Fields

Field	Description
31-0 RXIPV4HDRERROCT	RxIPv4 Header Error Octets This field indicates the number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). RxIPv4 Header Error Octets This field indicates the number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.255 Bytes Received in IPv4 Datagrams with No Payload (MAC_RXIPV4_NO_PAYLOAD_OCTETS)

This register provides the number of bytes received by DWC_ether_qos in IPv4 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.255.1 Offset

Register	Offset
MAC_RXIPV4_NO_PAYLOAD_OCTETS	858h

11.7.6.1.255.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4NOPAYOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4NOPAYOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.255.3 Fields

Field	Description
31-0 RXIPV4NOPAYOCT	RxIPv4 Payload Octets This field indicates the number of bytes received in IPv4 datagrams that did not have a TCP, UDP, or ICMP payload. RxIPv4 Payload Octets This field indicates the number of bytes received in IPv4 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.256 Bytes Received in Fragmented IPv4 Datagrams (MAC_RXIPV4_FRAGMENTED_OCTETS)

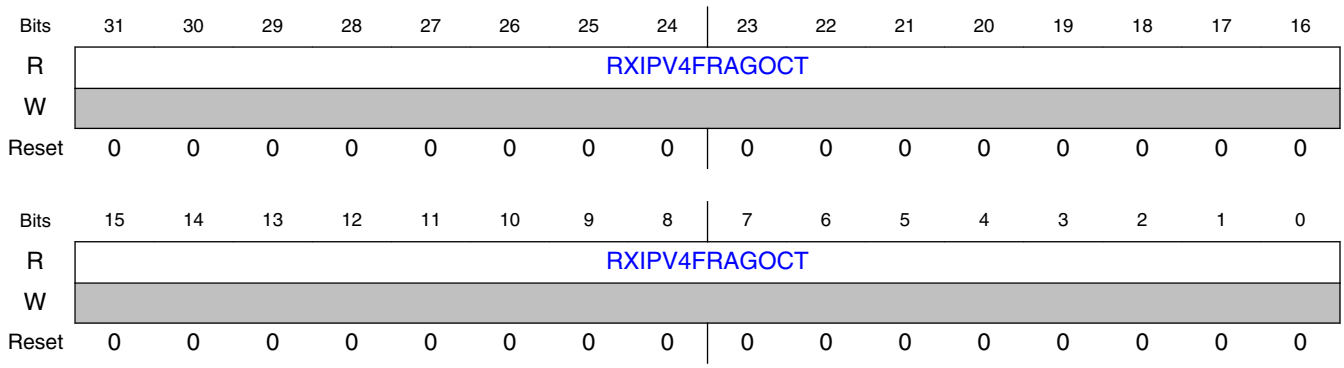
This register provides the number of bytes received by DWC_ether_qos in fragmented IPv4 datagrams.

This register provides the number of bytes received by DWC_ether_qos in fragmented IPv4 datagrams. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.256.1 Offset

Register	Offset
MAC_RXIPV4_FRAGMENTED_OCTETS	85Ch

11.7.6.1.256.2 Diagram



11.7.6.1.256.3 Fields

Field	Description
31-0 RXIPV4FRAGOCT	RxIPv4 Fragmented Octets This field indicates the number of bytes received in fragmented IPv4 datagrams. RxIPv4 Fragmented Octets This field indicates the number of bytes received in fragmented IPv4 datagrams. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.257 Bytes Received with UDP Checksum Disabled (MAC_RXIPV4_UDP_CHECKSUM_DISABLE_OCTETS)

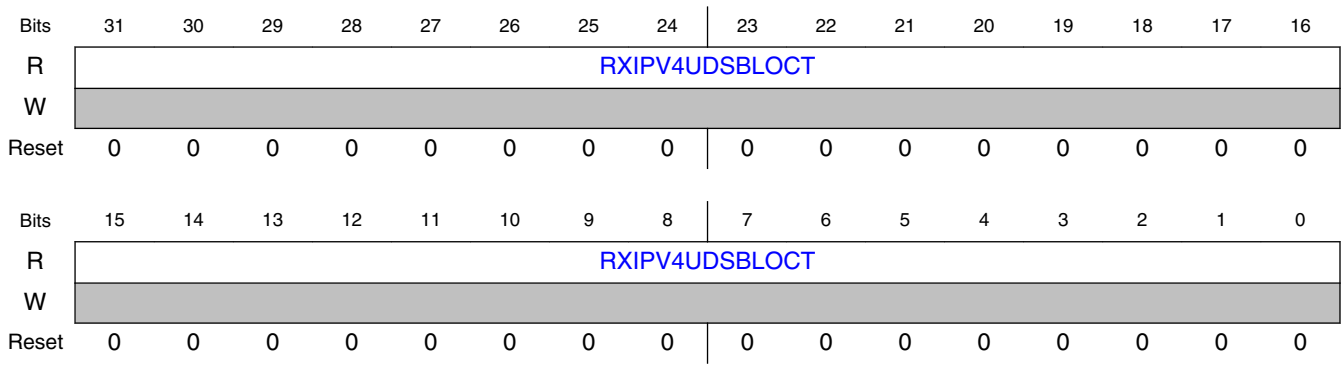
This register provides the number of bytes received by DWC_ether_qos in a UDP segment that had the UDP checksum disabled.

This register provides the number of bytes received by DWC_ether_qos in a UDP segment that had the UDP checksum disabled. This counter does not count IP Header bytes. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.257.1 Offset

Register	Offset
MAC_RXIPV4_UDP_CHECKSUM_DISABLE_OCTETS	860h

11.7.6.1.257.2 Diagram



11.7.6.1.257.3 Fields

Field	Description
31-0 RXIPV4UDSBLOCT	RxIPv4 UDP Checksum Disable Octets This field indicates the number of bytes received in a UDP segment that had the UDP checksum disabled. RxIPv4 UDP Checksum Disable Octets This field indicates the number of bytes received in a UDP segment that had the UDP checksum disabled. This counter does not count IP Header bytes. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.258 Bytes Received in Good IPv6 Datagrams (MAC_RXIPV6_GOOD_OCTETS)

This register provides the number of bytes received by DWC_ether_qos in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data.

This register provides the number of bytes received by DWC_ether_qos in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.258.1 Offset

Register	Offset
MAC_RXIPV6_GOOD_OCTETS	864h

11.7.6.1.258.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV6GDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV6GDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.258.3 Fields

Field	Description
31-0 RXIPV6GDOCT	RxIPv6 Good Octets This field indicates the number of bytes received in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data. RxIPv6 Good Octets This field indicates the number of bytes received in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.259 Bytes Received in IPv6 Datagrams with Data Errors (MAC_RXIPV6_HEADER_ERROR_OCTETS)

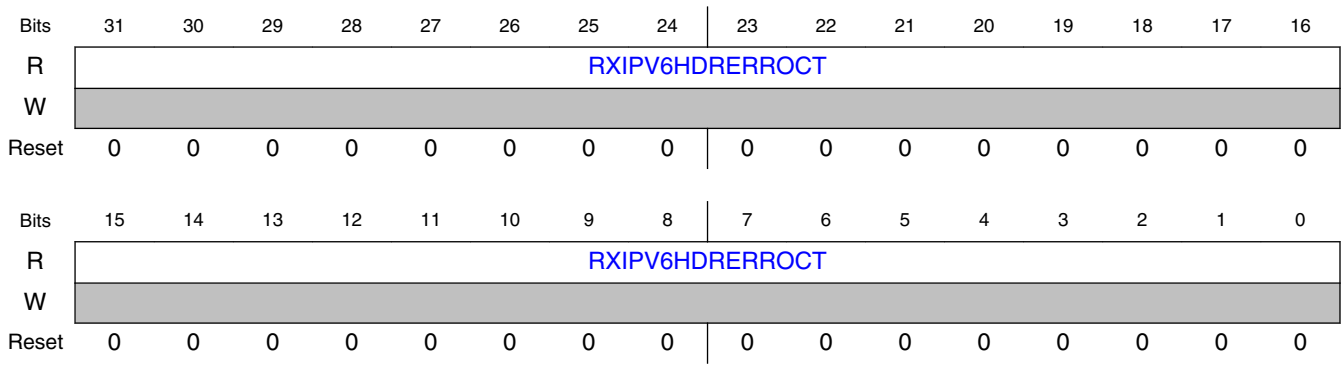
This register provides the number of bytes received by DWC_ether_qos in IPv6 datagrams with header errors (length, version mismatch).

This register provides the number of bytes received by DWC_ether_qos in IPv6 datagrams with header errors (length, version mismatch). The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.259.1 Offset

Register	Offset
MAC_RXIPV6_HEADER_ERROR_OCTETS	868h

11.7.6.1.259.2 Diagram



11.7.6.1.259.3 Fields

Field	Description
31-0 RXIPV6HDRERROCT	RxIPv6 Header Error Octets This field indicates the number of bytes received in IPv6 datagrams with header errors (length, version mismatch). RxIPv6 Header Error Octets This field indicates the number of bytes received in IPv6 datagrams with header errors (length, version mismatch). The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.260 Bytes Received in IPv6 Datagrams with No Payload (MAC_RXIPV6_NO_PAYLOAD_OCTETS)

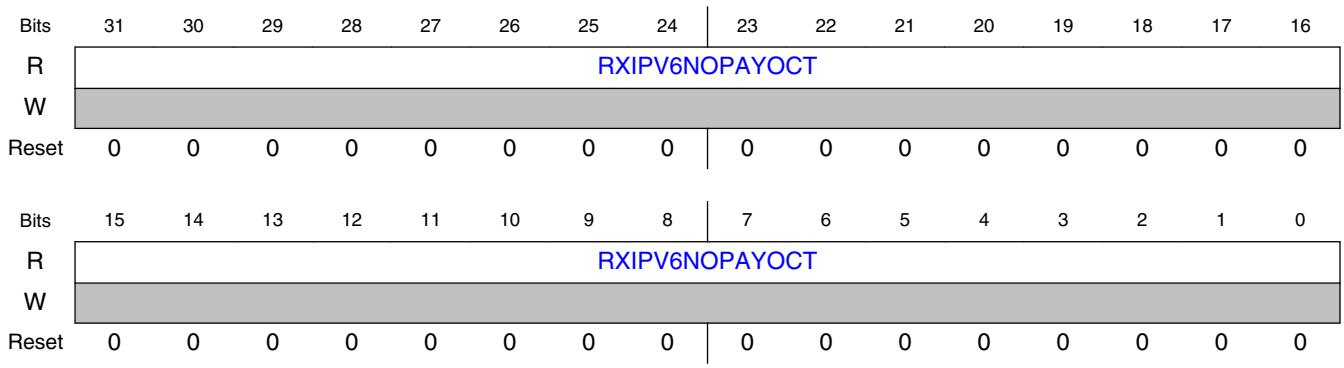
This register provides the number of bytes received by DWC_ether_qos in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload.

This register provides the number of bytes received by DWC_ether_qos in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.260.1 Offset

Register	Offset
MAC_RXIPV6_NO_PAYLOAD_OCTETS	86Ch

11.7.6.1.260.2 Diagram



11.7.6.1.260.3 Fields

Field	Description
31-0 RXIPV6NOPAYOCT	RxIPv6 Payload Octets This field indicates the number of bytes received in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload. RxIPv6 Payload Octets This field indicates the number of bytes received in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

11.7.6.1.261 Bytes Received in Good UDP Segment (MAC_RXUDP_GOOD_OCTETS)

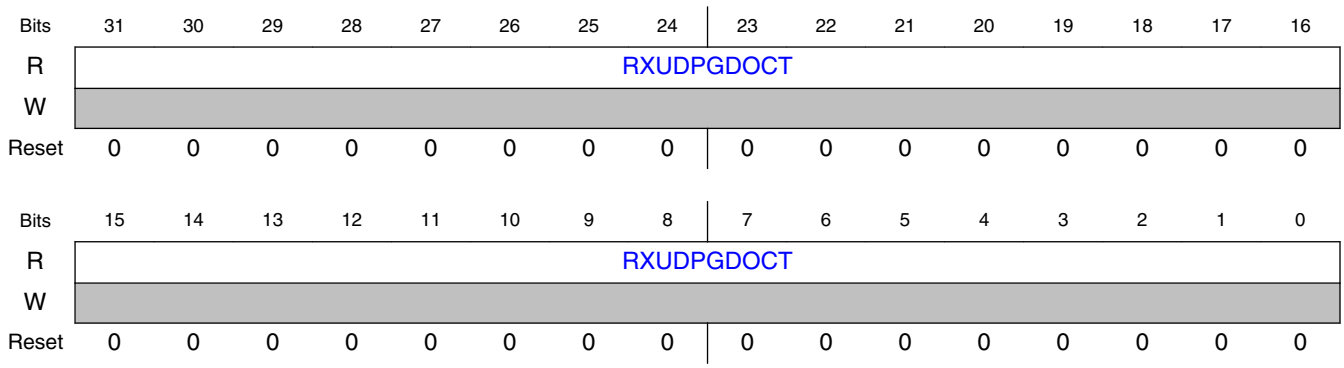
This register provides the number of bytes received by DWC_ether_qos in a good UDP segment.

This register provides the number of bytes received by DWC_ether_qos in a good UDP segment. This counter does not count IP header bytes.

11.7.6.1.261.1 Offset

Register	Offset
MAC_RXUDP_GOOD_OCTETS	870h

11.7.6.1.261.2 Diagram



11.7.6.1.261.3 Fields

Field	Description
31-0 RXUDPGDOCT	RxUDP Good Octets This field indicates the number of bytes received in a good UDP segment. RxUDP Good Octets This field indicates the number of bytes received in a good UDP segment. This counter does not count IP header bytes.

11.7.6.1.262 Bytes Received in UDP Segment with Checksum Errors (MAC_RXUDP_ERROR_OCTETS)

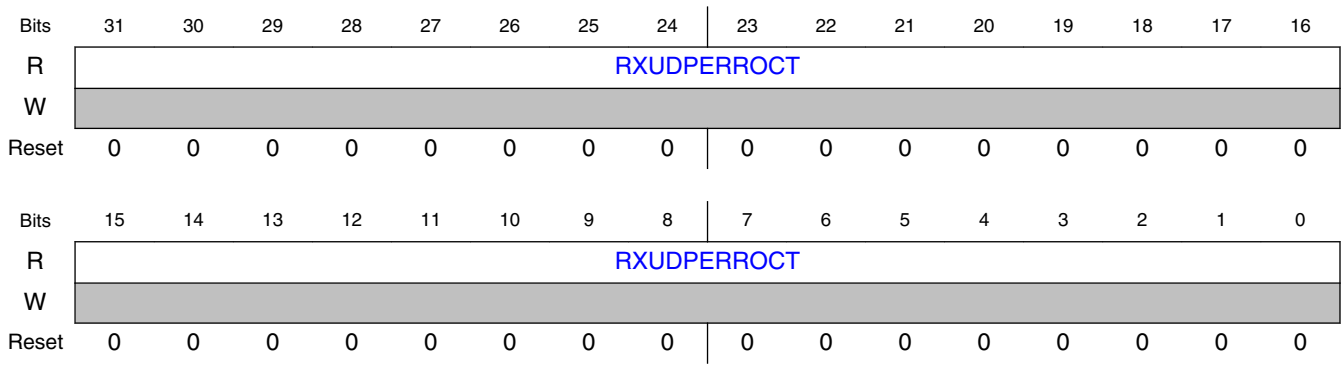
This register provides the number of bytes received by DWC_ether_qos in a UDP segment that had checksum errors.

This register provides the number of bytes received by DWC_ether_qos in a UDP segment that had checksum errors. This counter does not count IP header bytes.

11.7.6.1.262.1 Offset

Register	Offset
MAC_RXUDP_ERROR_OCTETS	874h

11.7.6.1.262.2 Diagram



11.7.6.1.262.3 Fields

Field	Description
31-0 RXUDPERROCT	RxUDP Error Octets This field indicates the number of bytes received in a UDP segment that had checksum errors. RxUDP Error Octets This field indicates the number of bytes received in a UDP segment that had checksum errors. This counter does not count IP header bytes.

11.7.6.1.263 Bytes Received in Good TCP Segment (MAC_RXTCP_GOOD_OCTETS)

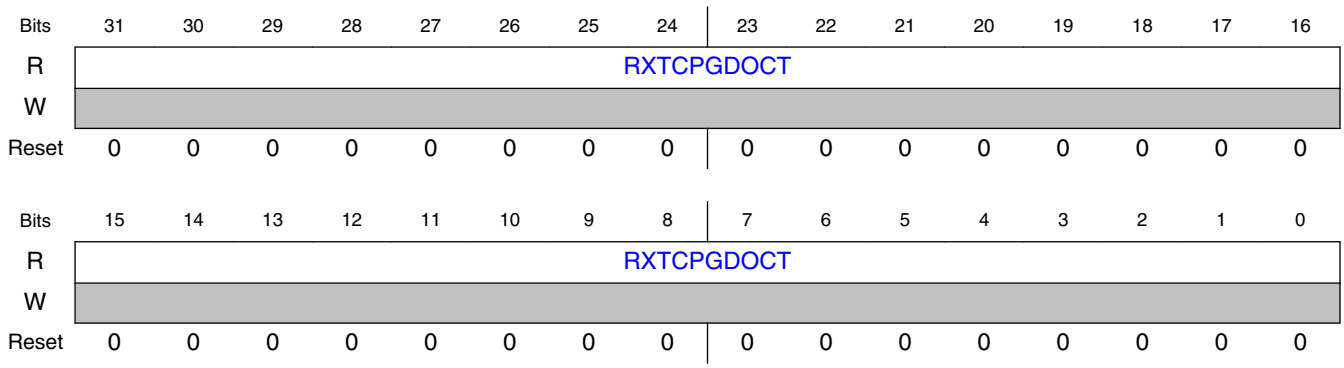
This register provides the number of bytes received by DWC_ether_qos in a good TCP segment.

This register provides the number of bytes received by DWC_ether_qos in a good TCP segment. This counter does not count IP header bytes.

11.7.6.1.263.1 Offset

Register	Offset
MAC_RXTCP_GOOD_OCTETS	878h

11.7.6.1.263.2 Diagram



11.7.6.1.263.3 Fields

Field	Description
31-0 RXTCPGDOCT	RxTCP Good Octets This field indicates the number of bytes received in a good TCP segment. RxTCP Good Octets This field indicates the number of bytes received in a good TCP segment. This counter does not count IP header bytes.

11.7.6.1.264 Bytes Received in TCP Segment with Checksum Errors (MAC_RXTCP_ERROR_OCTETS)

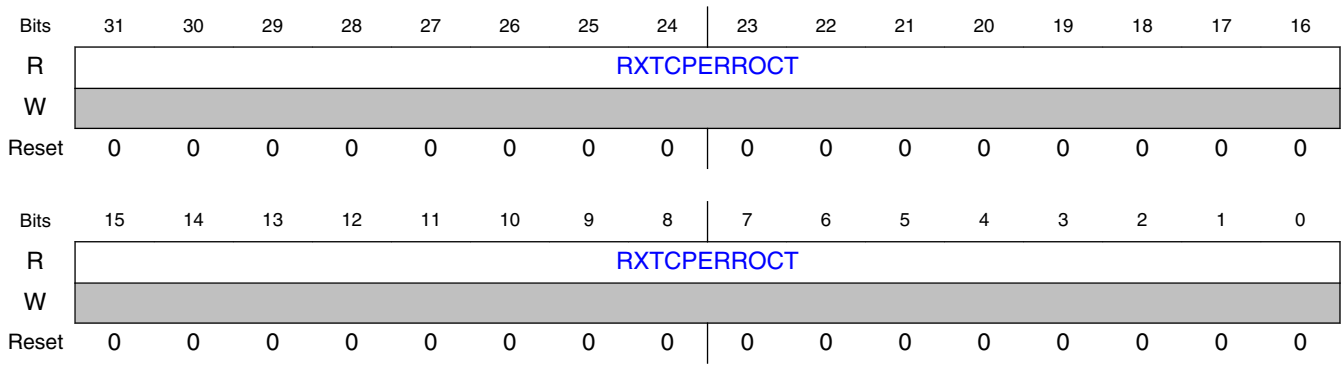
This register provides the number of bytes received by DWC_ether_qos in a TCP segment that had checksum errors.

This register provides the number of bytes received by DWC_ether_qos in a TCP segment that had checksum errors. This counter does not count IP header bytes.

11.7.6.1.264.1 Offset

Register	Offset
MAC_RXTCP_ERROR_OCTETS	87Ch

11.7.6.1.264.2 Diagram



11.7.6.1.264.3 Fields

Field	Description
31-0 RXTCPERROCT	RxTCP Error Octets This field indicates the number of bytes received in a TCP segment that had checksum errors. RxTCP Error Octets This field indicates the number of bytes received in a TCP segment that had checksum errors. This counter does not count IP header bytes.

11.7.6.1.265 Bytes Received in Good ICMP Segment (MAC_RXICMP_GOOD_OCTETS)

This register provides the number of bytes received by DWC_ether_qos in a good ICMP segment.

This register provides the number of bytes received by DWC_ether_qos in a good ICMP segment. This counter does not count IP header bytes.

11.7.6.1.265.1 Offset

Register	Offset
MAC_RXICMP_GOOD_OCTETS	880h

11.7.6.1.265.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXICMPGDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXICMPGDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.265.3 Fields

Field	Description
31-0 RXICMPGDOCT	RxICMP Good Octets This field indicates the number of bytes received in a good ICMP segment. RxICMP Good Octets This field indicates the number of bytes received in a good ICMP segment. This counter does not count IP header bytes.

11.7.6.1.266 Bytes Received in ICMP Segment with Checksum Errors (MAC_RXICMP_ERROR_OCTETS)

This register provides the number of bytes received by DWC_ether_qos in a ICMP segment that had checksum errors.

This register provides the number of bytes received by DWC_ether_qos in a ICMP segment that had checksum errors. This counter does not count IP header bytes.

11.7.6.1.266.1 Offset

Register	Offset
MAC_RXICMP_ERROR_OCTETS	884h

11.7.6.1.266.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXICMPERROCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXICMPERROCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.266.3 Fields

Field	Description
31-0 RXICMPERROCT	RxICMP Error Octets This field indicates the number of bytes received in a ICMP segment that had checksum errors. RxICMP Error Octets This field indicates the number of bytes received in a ICMP segment that had checksum errors. This counter does not count IP header bytes.

11.7.6.1.267 MMC FPE Transmit Interrupt (MAC_MMC_FPE_TX_INTERRUPT)

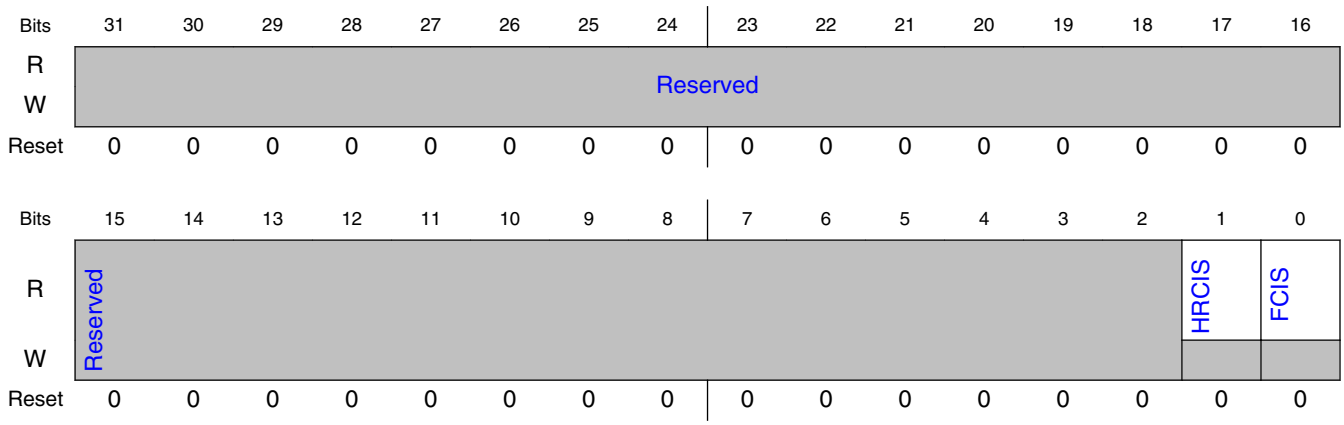
This register maintains the interrupts generated from all FPE related Transmit statistics counters.

This register maintains the interrupts generated from all FPE related Transmit statistics counters. The MMC FPE Transmit Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter). When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones. The MMC FPE Transmit Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (Bits[7:0]) of the respective counter must be read to clear the interrupt bit.

11.7.6.1.267.1 Offset

Register	Offset
MAC_MMC_FPE_TX_INTERRUPT	8A0h

11.7.6.1.267.2 Diagram



11.7.6.1.267.3 Fields

Field	Description
31-2 —	Reserved
1 HRCIS	MMC Tx Hold Request Counter Interrupt Status This bit is set when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. MMC Tx Hold Request Counter Interrupt Status This bit is set when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE with AV_EST Enabled configuration. 0 - MMC Tx Hold Request Counter Interrupt Status not detected 1 - MMC Tx Hold Request Counter Interrupt Status detected
0 FCIS	MMC Tx FPE Fragment Counter Interrupt status This bit is set when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. MMC Tx FPE Fragment Counter Interrupt status This bit is set when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event.

Field	Description
	Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Tx FPE Fragment Counter Interrupt status not detected 1 - MMC Tx FPE Fragment Counter Interrupt status detected

11.7.6.1.268 MMC FPE Transmit Mask Interrupt (MAC_MMC_FPE_TX_INTERRUPT_MASK)

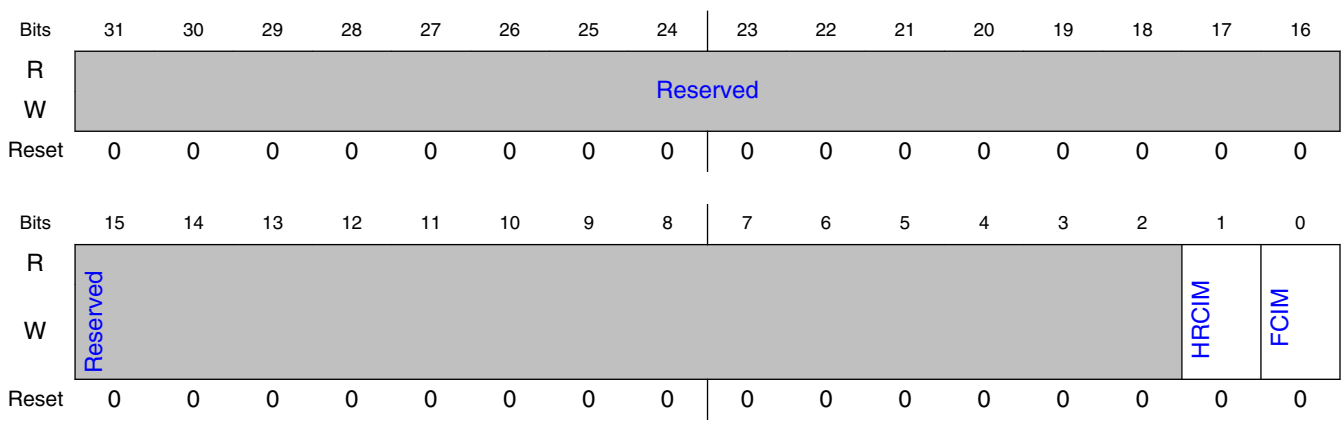
This register maintains the masks for interrupts generated from all FPE related Transmit statistics counters.

This register maintains the masks for interrupts generated from all FPE related Transmit statistics counters. The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when FPE related receive statistic counters reach half of their maximum value or the maximum values. This register is 32 bit wide.

11.7.6.1.268.1 Offset

Register	Offset
MAC_MMC_FPE_TX_INTERRUPT_MASK	8A4h

11.7.6.1.268.2 Diagram



11.7.6.1.268.3 Fields

Field	Description
31-2	Reserved

Table continues on the next page...

Field	Description
—	
1 HRCIM	MMC Transmit Hold Request Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. MMC Transmit Hold Request Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE with AV_EST Enabled configuration. 0 - MMC Transmit Hold Request Counter Interrupt Mask is disabled 1 - MMC Transmit Hold Request Counter Interrupt Mask is enabled
0 FCIM	MMC Transmit Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. MMC Transmit Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Transmit Fragment Counter Interrupt Mask is disabled 1 - MMC Transmit Fragment Counter Interrupt Mask is enabled

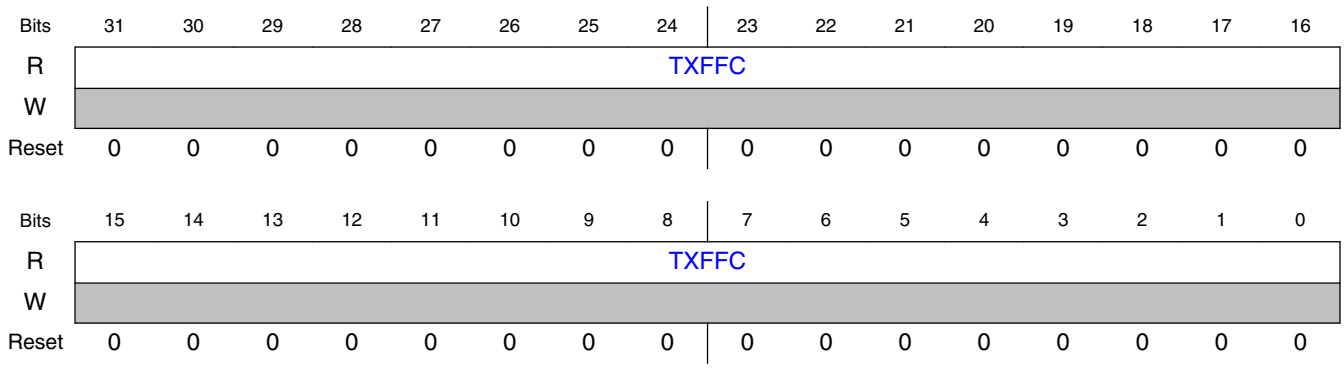
11.7.6.1.269 MMC FPE Transmitted Fragment Counter (MAC_MMC_TX_FPE_FRAGMENT_CNTR)

This register provides the number of additional mPackets transmitted due to preemption.

11.7.6.1.269.1 Offset

Register	Offset
MAC_MMC_TX_FPE_FRAGMENT_CNTR	8A8h

11.7.6.1.269.2 Diagram



11.7.6.1.269.3 Fields

Field	Description
31-0 TXFFC	Tx FPE Fragment counter This field indicates the number of additional mPackets that has been transmitted due to preemption Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

11.7.6.1.270 MMC FPE Transmitted Hold Request Counter (MAC_MMC_TX_HOLD_REQ_CNTR)

This register provides the count of number of times a hold request is given to MAC

11.7.6.1.270.1 Offset

Register	Offset
MAC_MMC_TX_HOLD_REQ_CNTR	8ACh

11.7.6.1.270.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXHRC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXHRC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.270.3 Fields

Field	Description
31-0 TXHRC	Tx Hold Request Counter This field indicates count of number of a hold request is given to MAC. Tx Hold Request Counter This field indicates count of number of a hold request is given to MAC. Exists when any one of the RX/TX MMC counters are enabled during FPE with AV_EST Enabled configuration.

11.7.6.1.271 MMC FPE Receive Interrupt (MAC_MMC_FPE_RX_INTERRUPT)

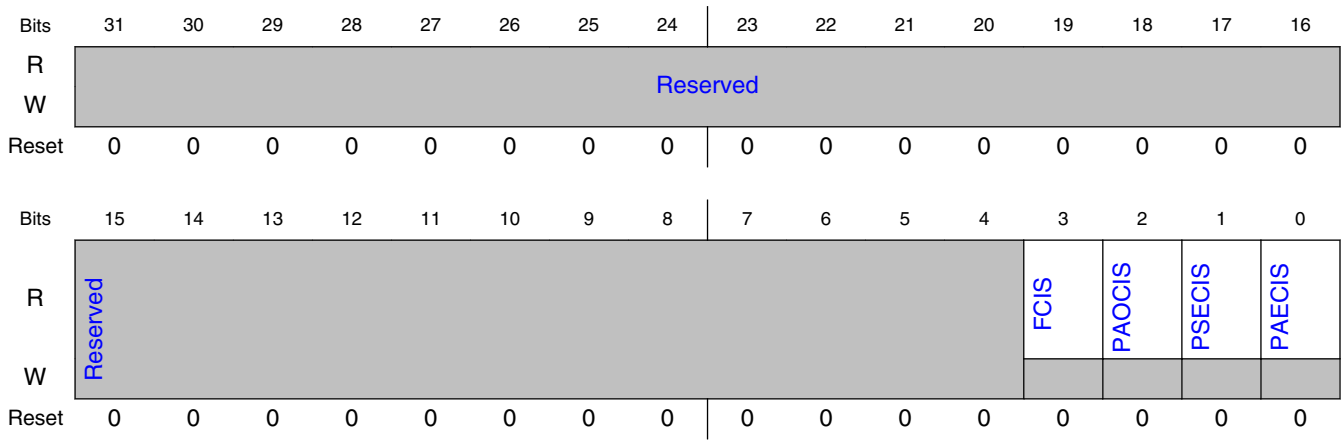
This register maintains the interrupts generated from all FPE related Receive statistics counters.

This register maintains the interrupts generated from all FPE related Receive statistics counters. The MMC FPE Receive Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter). When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones. The MMC FPE Receive Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (Bits[7:0]) of the respective counter must be read to clear the interrupt bit.

11.7.6.1.271.1 Offset

Register	Offset
MAC_MMC_FPE_RX_INTERRUPT	8C0h

11.7.6.1.271.2 Diagram



11.7.6.1.271.3 Fields

Field	Description
31-4 —	Reserved
3 FCIS	MMC Rx FPE Fragment Counter Interrupt Status This bit is set when the Rx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx FPE Fragment Counter Interrupt Status This bit is set when the Rx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx FPE Fragment Counter Interrupt Status not detected 1 - MMC Rx FPE Fragment Counter Interrupt Status detected
2 PAOCIS	MMC Rx Packet Assembly OK Counter Interrupt Status This bit is set when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet Assembly OK Counter Interrupt Status This bit is set when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

Table continues on the next page...

Field	Description
	0 - MMC Rx Packet Assembly OK Counter Interrupt Status not detected 1 - MMC Rx Packet Assembly OK Counter Interrupt Status detected
1 PSECIS	MMC Rx Packet SMD Error Counter Interrupt Status This bit is set when the Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet SMD Error Counter Interrupt Status This bit is set when the Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx Packet SMD Error Counter Interrupt Status not detected 1 - MMC Rx Packet SMD Error Counter Interrupt Status detected
0 PAECIS	MMC Rx Packet Assembly Error Counter Interrupt Status This bit is set when the Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet Assembly Error Counter Interrupt Status This bit is set when the Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx Packet Assembly Error Counter Interrupt Status not detected 1 - MMC Rx Packet Assembly Error Counter Interrupt Status detected

11.7.6.1.272 MMC FPE Receive Interrupt Mask (MAC_MMC_FPE_RX_INTERRUPT_MASK)

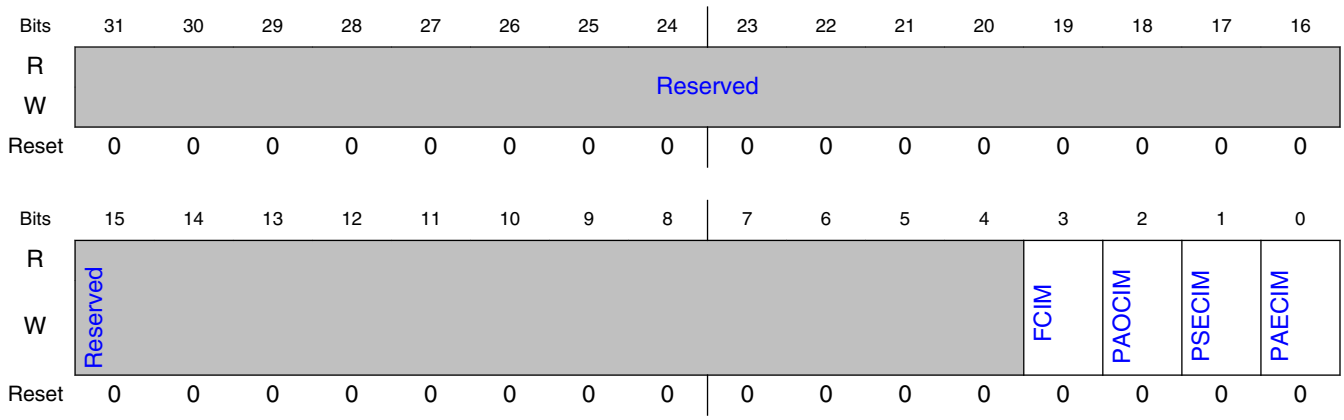
This register maintains the masks for interrupts generated from all FPE related Receive statistics counters.

This register maintains the masks for interrupts generated from all FPE related Receive statistics counters. The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when FPE related receive statistic counters reach half of their maximum value or the maximum values. This register is 32 bit wide.

11.7.6.1.272.1 Offset

Register	Offset
MAC_MMC_FPE_RX_INTERRUPT_MASK	8C4h

11.7.6.1.272.2 Diagram



11.7.6.1.272.3 Fields

Field	Description
31-4 —	Reserved
3 FCIM	MMC Rx FPE Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx FPE Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx FPE Fragment Counter Interrupt Mask is disabled 1 - MMC Rx FPE Fragment Counter Interrupt Mask is enabled
2 PAOCIM	MMC Rx Packet Assembly OK Counter Interrupt Mask Setting this bit masks the interrupt when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet Assembly OK Counter Interrupt Mask Setting this bit masks the interrupt when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx Packet Assembly OK Counter Interrupt Mask is disabled 1 - MMC Rx Packet Assembly OK Counter Interrupt Mask is enabled
1 PSECIM	MMC Rx Packet SMD Error Counter Interrupt Mask Setting this bit masks the interrupt when the R Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet SMD Error Counter Interrupt Mask Setting this bit masks the interrupt when the R Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx Packet SMD Error Counter Interrupt Mask is disabled 1 - MMC Rx Packet SMD Error Counter Interrupt Mask is enabled

Table continues on the next page...

Field	Description
0 PAECIM	MMC Rx Packet Assembly Error Counter Interrupt Mask Setting this bit masks the interrupt when the R Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet Assembly Error Counter Interrupt Mask Setting this bit masks the interrupt when the R Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0 - MMC Rx Packet Assembly Error Counter Interrupt Mask is disabled 1 - MMC Rx Packet Assembly Error Counter Interrupt Mask is enabled

11.7.6.1.273 MMC Receive Packet Reassembly Error Counter (MAC_MMC_RX_PACKET_ASSEMBLY_ERR_CNTR)

This register provides the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value.

11.7.6.1.273.1 Offset

Register	Offset
MAC_MMC_RX_PACKET_ASSEMBLY_ERR_CNTR	8C8h

11.7.6.1.273.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PAEC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PAEC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.273.3 Fields

Field	Description
31-0	Rx Packet Assembly Error Counter This field indicates the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value.

Field	Description
PAEC	Rx Packet Assembly Error Counter This field indicates the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

11.7.6.1.274 MMC Receive Packet SMD Error Counter (MAC_MMC_RX_PACKET_SMD_ERR_CNTR)

This register provides the number of received MAC frames rejected due to unknown SMD value and MAC frame fragments rejected due to arriving with an SMD-C when there was no preceding preempted frame.

11.7.6.1.274.1 Offset

Register	Offset
MAC_MMC_RX_PACKET_SMD_ERR_CNTR	8CCh

11.7.6.1.274.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PSEC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PSEC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.274.3 Fields

Field	Description
31-0 PSEC	Rx Packet SMD Error Counter This field indicates the number of MAC frames rejected due to unknown SMD value and MAC frame fragments rejected due to arriving with an SMD-C when there was no preceding preempted frame. Rx Packet SMD Error Counter This field indicates the number of MAC frames rejected due to unknown SMD value and MAC frame fragments rejected due to arriving with an SMD-C when there was no preceding preempted frame.

Field	Description
	Exists when at least one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

11.7.6.1.275 MMC Receive Packet Successful Reassembly Counter (MAC_MMC_RX_PACKET_ASSEMBLY_OK_CNTR)

This register provides the number of MAC frames that were successfully reassembled and delivered to MAC.

11.7.6.1.275.1 Offset

Register	Offset
MAC_MMC_RX_PACKET_ASSEMBLY_OK_CNTR	8D0h

11.7.6.1.275.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PAOC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PAOC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.275.3 Fields

Field	Description
31-0 PAOC	Rx Packet Assembly OK Counter This field indicates the number of MAC frames that were successfully reassembled and delivered to MAC. Rx Packet Assembly OK Counter This field indicates the number of MAC frames that were successfully reassembled and delivered to MAC. Exists when at least one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

11.7.6.1.276 MMC FPE Received Fragment Counter (MAC_MMC_RX_FPE_FRAGMENT_CNTR)

This register provides the number of additional mPackets received due to preemption.

11.7.6.1.276.1 Offset

Register	Offset
MAC_MMC_RX_FPE_FRAGMENT_CNTR	8D4h

11.7.6.1.276.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FFC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FFC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.276.3 Fields

Field	Description
31-0 FFC	Rx FPE Fragment Counter This field indicates the number of additional mPackets received due to preemption Exists when at least one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

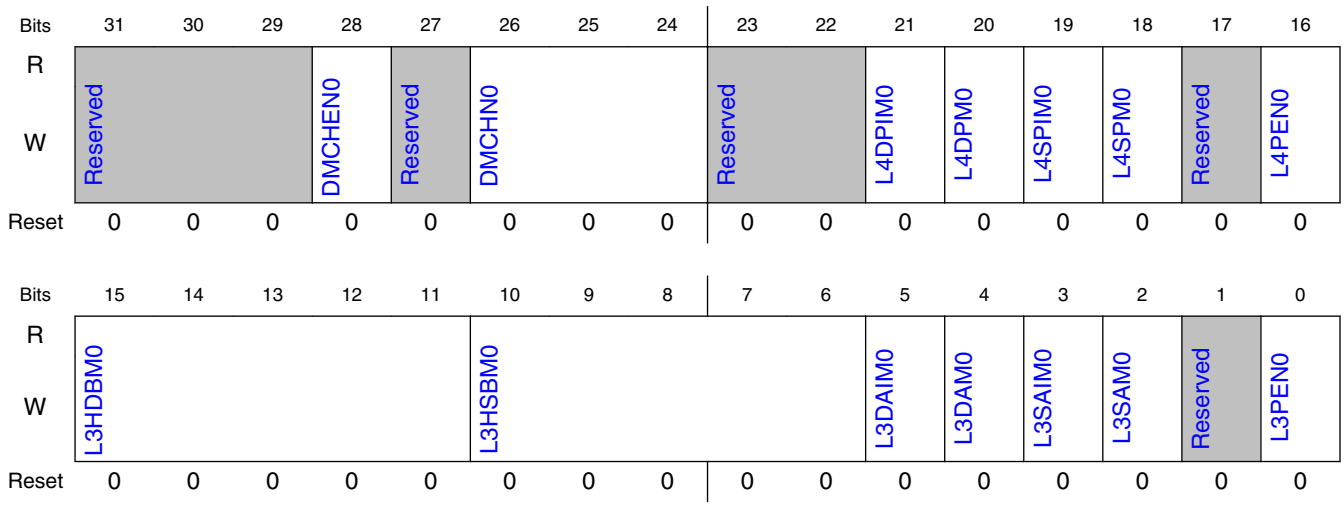
11.7.6.1.277 Layer 3 and Layer 4 Control of Filter 0 (MAC_L3_L4_CONTROL0)

The Layer 3 and Layer 4 Control register controls the operations of filter 0 of Layer 3 and Layer 4.

11.7.6.1.277.1 Offset

Register	Offset
MAC_L3_L4_CONTROL0	900h

11.7.6.1.277.2 Diagram



11.7.6.1.277.3 Fields

Field	Description
31-29 —	Reserved
28 DMCHEN0	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN0	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number

Table continues on the next page...

Field	Description
	When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM0	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM0 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM0	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM0	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM0	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN0	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching.

Table continues on the next page...

Field	Description
	The Layer 4 matching is done only when the L4SPM0 or L4DPM0 bit is set. 0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM0	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM0	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM0	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable

Table continues on the next page...

Field	Description
	When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM0	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM0	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2 L3SAM0	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN0	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

11.7.6.1.278 Layer 4 Address 0 (MAC_LAYER4_ADDRESS0)

The MAC_LAYER4_ADDRESS(#i), MAC_L3_L4_CONTROL(#i), MAC_LAYER3_ADDR0_Reg(#i), MAC_LAYER3_ADDR1_Reg(#i), MAC_LAYER3_ADDR2_Reg(#i) and MAC_LAYER3_ADDR3_Reg(#i) registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

11.7.6.1.278.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS0	904h

11.7.6.1.278.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.278.3 Fields

Field	Description
31-16 L4DP0	Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4DPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP0	Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

11.7.6.1.279 Layer 3 Address 0 Register 0 (MAC_LAYER3_ADDR0_REG0)

For IPv4 packets, the Layer 3 Address 0 Register 0 register contains the 32-bit IP Source Address field.

For IPv4 packets, the Layer 3 Address 0 Register 0 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.279.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG0	910h

11.7.6.1.279.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A00															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A00															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.279.3 Fields

Field	Description
31-0 L3A00	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

11.7.6.1.280 Layer 3 Address 1 Register 0 (MAC_LAYER3_ADDR1_REG0)

For IPv4 packets, the Layer 3 Address 1 Register 0 register contains the 32-bit IP Destination Address field.

For IPv4 packets, the Layer 3 Address 1 Register 0 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.280.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG0	914h

11.7.6.1.280.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A10															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A10															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.280.3 Fields

Field	Description
31-0 L3A10	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

11.7.6.1.281 Layer 3 Address 2 Register 0 (MAC_LAYER3_ADDR2_REG0)

The Layer 3 Address 2 Register 0 register is reserved for IPv4 packets.

The Layer 3 Address 2 Register 0 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.281.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG0	918h

11.7.6.1.281.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A20															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A20															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.281.3 Fields

Field	Description
31-0 L3A20	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

11.7.6.1.282 Layer 3 Address 3 Register 0 (MAC_LAYER3_ADDR3_REG0)

The Layer 3 Address 3 Register 0 register is reserved for IPv4 packets.

The Layer 3 Address 3 Register 0 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.282.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG0	91Ch

11.7.6.1.282.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A30															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A30															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.282.3 Fields

Field	Description
31-0 L3A30	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

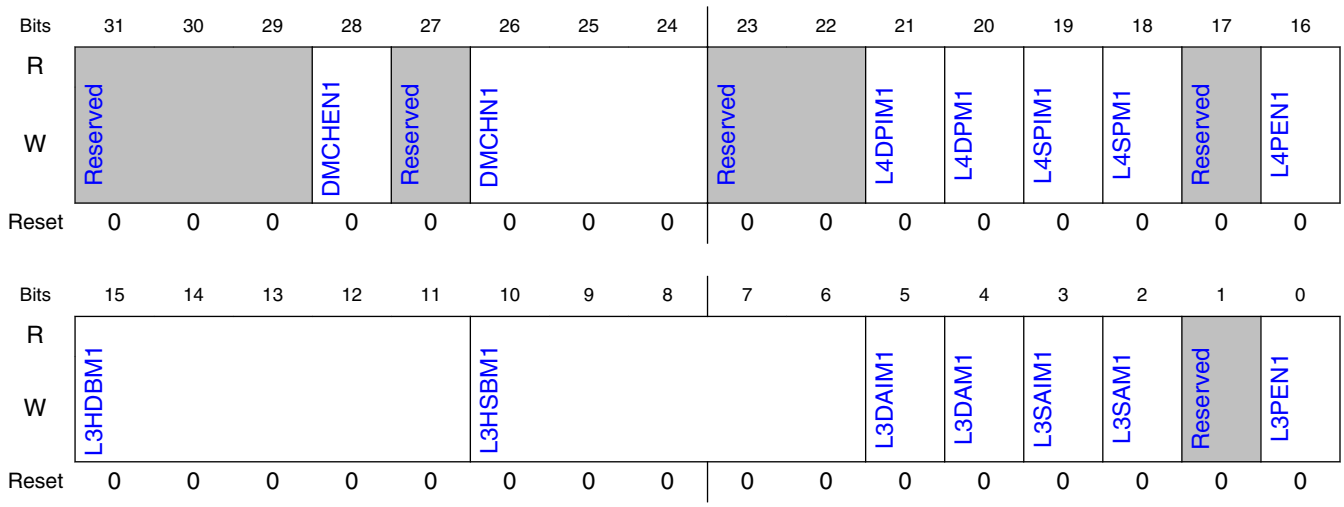
11.7.6.1.283 Layer 3 and Layer 4 Control of Filter 1 (MAC_L3_L4_CONTROL1)

The Layer 3 and Layer 4 Control register controls the operations of filter 0 of Layer 3 and Layer 4.

11.7.6.1.283.1 Offset

Register	Offset
MAC_L3_L4_CONTROL 1	930h

11.7.6.1.283.2 Diagram



11.7.6.1.283.3 Fields

Field	Description
31-29 —	Reserved
28 DMCHEN1	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN1	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM1	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable

Table continues on the next page...

Field	Description
	When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM0 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM1	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM1	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM1	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN1	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM0 bit is set. 0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM1	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets:

Table continues on the next page...

Field	Description
	This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM1	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM1	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM1	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching.

Table continues on the next page...

Field	Description
	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM1	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2 L3SAM1	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN1	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

11.7.6.1.284 Layer 4 Address 0 (MAC_LAYER4_ADDRESS1)

The MAC_LAYER4_ADDRESS(#i), MAC_L3_L4_CONTROL(#i), MAC_LAYER3_ADDR0_Reg(#i), MAC_LAYER3_ADDR1_Reg(#i), MAC_LAYER3_ADDR2_Reg(#i) and MAC_LAYER3_ADDR3_Reg(#i) registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

11.7.6.1.284.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS1	934h

11.7.6.1.284.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.284.3 Fields

Field	Description
31-16 L4DP1	Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field

Table continues on the next page...

Field	Description
	When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4DPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP1	Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

11.7.6.1.285 Layer 3 Address 0 Register 1 (MAC_LAYER3_ADDR0_REG1)

For IPv4 packets, the Layer 3 Address 0 Register 1 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.285.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG1	940h

11.7.6.1.285.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A01															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A01															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.285.3 Fields

Field	Description
31-0 L3A01	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

11.7.6.1.286 Layer 3 Address 1 Register 1 (MAC_LAYER3_ADDR1_REG1)

For IPv4 packets, the Layer 3 Address 1 Register 0 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.286.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG1	944h

11.7.6.1.286.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A11															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A11															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.286.3 Fields

Field	Description
31-0 L3A11	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

11.7.6.1.287 Layer 3 Address 2 Register 1 (MAC_LAYER3_ADDR2_REG1)

The Layer 3 Address 2 Register 1 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.287.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG1	948h

11.7.6.1.287.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A21															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A21															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.287.3 Fields

Field	Description
31-0 L3A21	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

11.7.6.1.288 Layer 3 Address 3 Register 1 (MAC_LAYER3_ADDR3_REG1)

The Layer 3 Address 3 Register 1 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.288.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG1	94Ch

11.7.6.1.288.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A31															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A31															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.288.3 Fields

Field	Description
31-0 L3A31	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

11.7.6.1.289 Layer 3 and Layer 4 Control of Filter 2 (MAC_L3_L4_CONTROL2)

The Layer 3 and Layer 4 Control register controls the operations of filter 0 of Layer 3 and Layer 4.

11.7.6.1.289.1 Offset

Register	Offset
MAC_L3_L4_CONTROL2	960h

11.7.6.1.289.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN2	Reserved	DMCHN2			Reserved		L4DPIM2	L4DPM2	L4SPIM2	L4SPM2	Reserved	L4PEN2
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM2					L3HSBM2					L3DAIM2	L3DAM2	L3SAIM2	L3SAM2	Reserved	L3PEN2
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.289.3 Fields

Field	Description
31-29 —	Reserved
28 DMCHEN2	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN2	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM2	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM0 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM2	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM2	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high.

Table continues on the next page...

Field	Description
	0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM2	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN2	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM0 bit is set. 0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM2	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.

Table continues on the next page...

Field	Description
10-6 L3HSBM2	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM2	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM2	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM2	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2 L3SAM2	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable

Table continues on the next page...

Field	Description
	When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN2	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

11.7.6.1.290 Layer 4 Address 2 (MAC_LAYER4_ADDRESS2)

The MAC_LAYER4_ADDRESS(#i), MAC_L3_L4_CONTROL(#i), MAC_LAYER3_ADDR0_Reg(#i), MAC_LAYER3_ADDR1_Reg(#i), MAC_LAYER3_ADDR2_Reg(#i) and MAC_LAYER3_ADDR3_Reg(#i) registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

11.7.6.1.290.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS2	964h

11.7.6.1.290.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.290.3 Fields

Field	Description
31-16 L4DP2	Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4DPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP2	Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

11.7.6.1.291 Layer 3 Address 0 Register 2 (MAC_LAYER3_ADDR0_REG2)

For IPv4 packets, the Layer 3 Address 0 Register 2 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.291.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG2	970h

11.7.6.1.291.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A02															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A02															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.291.3 Fields

Field	Description
31-0 L3A02	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

11.7.6.1.292 Layer 3 Address 0 Register 2 (MAC_LAYER3_ADDR1_REG2)

For IPv4 packets, the Layer 3 Address 1 Register 2 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.292.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG2	974h

11.7.6.1.292.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A12															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A12															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.292.3 Fields

Field	Description
31-0 L3A12	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

11.7.6.1.293 Layer 3 Address 2 Register 2 (MAC_LAYER3_ADDR2_REG2)

The Layer 3 Address 2 Register 2 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.293.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG2	978h

11.7.6.1.293.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A22															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A22															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.293.3 Fields

Field	Description
31-0 L3A22	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

11.7.6.1.294 Layer 3 Address 3 Register 2 (MAC_LAYER3_ADDR3_REG2)

The Layer 3 Address 3 Register 2 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.294.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG2	97Ch

11.7.6.1.294.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A32															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A32															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.294.3 Fields

Field	Description
31-0 L3A32	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

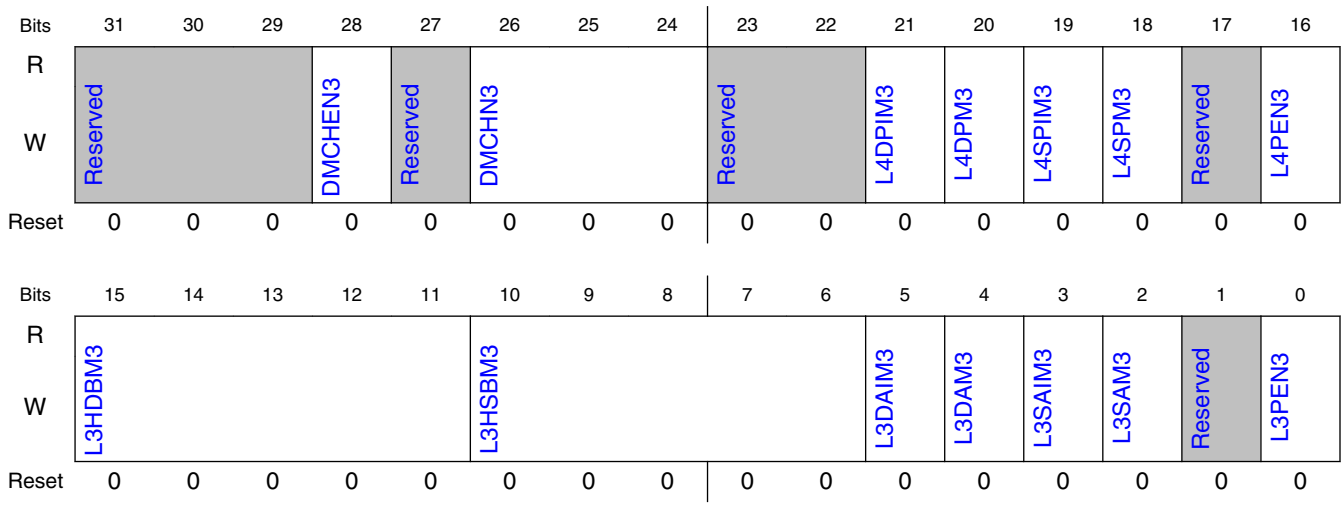
11.7.6.1.295 Layer 3 and Layer 4 Control of Filter 3 (MAC_L3_L4_CONTROL3)

The Layer 3 and Layer 4 Control register controls the operations of filter 0 of Layer 3 and Layer 4.

11.7.6.1.295.1 Offset

Register	Offset
MAC_L3_L4_CONTROL 3	990h

11.7.6.1.295.2 Diagram



11.7.6.1.295.3 Fields

Field	Description
31-29 —	Reserved
28 DMCHEN3	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN3	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number

Table continues on the next page...

Field	Description
	When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM3	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM0 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM3	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM3	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM3	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN3	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching.

Table continues on the next page...

Field	Description
	The Layer 4 matching is done only when the L4SPM0 or L4DPM0 bit is set. 0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM3	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM3	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM3	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable

Table continues on the next page...

Field	Description
	When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM3	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM3	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2 L3SAM3	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN3	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

11.7.6.1.296 Layer 4 Address 3 (MAC_LAYER4_ADDRESS3)

The MAC_LAYER4_ADDRESS(#i), MAC_L3_L4_CONTROL(#i), MAC_LAYER3_ADDR0_Reg(#i), MAC_LAYER3_ADDR1_Reg(#i), MAC_LAYER3_ADDR2_Reg(#i) and MAC_LAYER3_ADDR3_Reg(#i) registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

11.7.6.1.296.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS3	994h

11.7.6.1.296.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.296.3 Fields

Field	Description
31-16 L4DP3	Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4DPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP3	Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

11.7.6.1.297 Layer 3 Address 0 Register 3 (MAC_LAYER3_ADDR0_REG3)

For IPv4 packets, the Layer 3 Address 0 Register 3 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.297.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG3	9A0h

11.7.6.1.297.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A03															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A03															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.297.3 Fields

Field	Description
31-0 L3A03	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

11.7.6.1.298 Layer 3 Address 1 Register 3 (MAC_LAYER3_ADDR1_REG3)

For IPv4 packets, the Layer 3 Address 1 Register 3 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.298.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG3	9A4h

11.7.6.1.298.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A13															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A13															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.298.3 Fields

Field	Description
31-0 L3A13	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

11.7.6.1.299 Layer 3 Address 2 Register 3 (MAC_LAYER3_ADDR2_REG3)

The Layer 3 Address 2 Register 3 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.299.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG3	9A8h

11.7.6.1.299.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A23															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A23															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.299.3 Fields

Field	Description
31-0 L3A23	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

11.7.6.1.300 Layer 3 Address 3 Register 3 (MAC_LAYER3_ADDR3_REG3)

The Layer 3 Address 3 Register 3 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.300.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG3	9ACh

11.7.6.1.300.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A33															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A33															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.300.3 Fields

Field	Description
31-0 L3A33	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

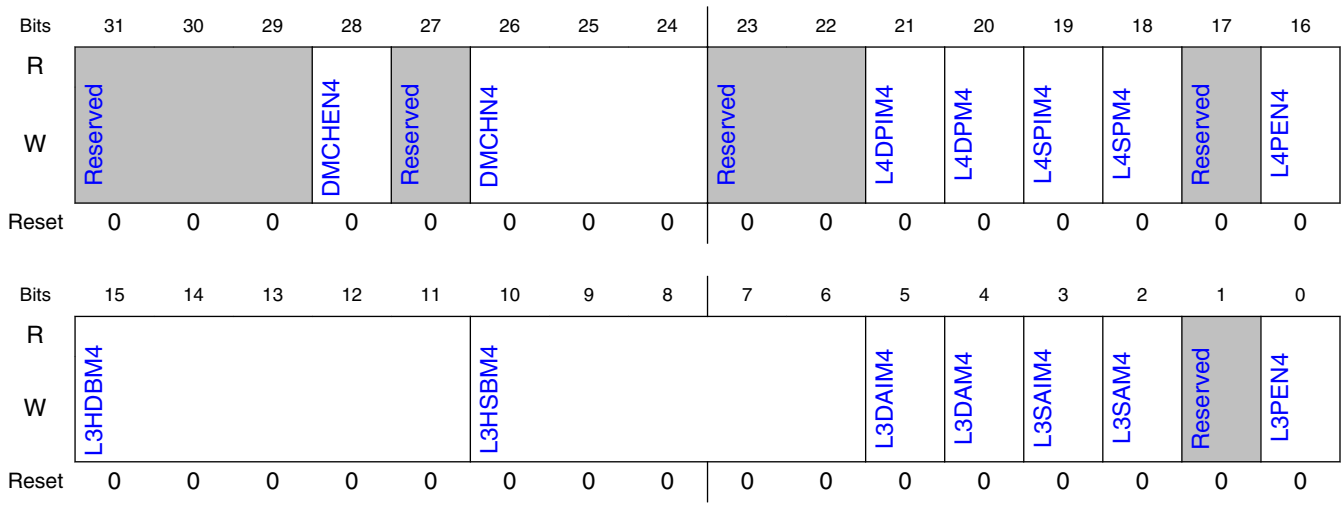
11.7.6.1.301 Layer 3 and Layer 4 Control of Filter 4 (MAC_L3_L4_CONTROL4)

The Layer 3 and Layer 4 Control register controls the operations of filter 0 of Layer 3 and Layer 4.

11.7.6.1.301.1 Offset

Register	Offset
MAC_L3_L4_CONTROL 4	9C0h

11.7.6.1.301.2 Diagram



11.7.6.1.301.3 Fields

Field	Description
31-29 —	Reserved
28 DMCHEN4	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN4	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM4	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable

Table continues on the next page...

Field	Description
	When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM0 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM4	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM4	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM4	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN4	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM0 bit is set. 0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM4	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets:

Table continues on the next page...

Field	Description
	This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM4	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM4	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM4	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching.

Table continues on the next page...

Field	Description
	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM4	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2 L3SAM4	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN4	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

11.7.6.1.302 Layer 4 Address 4 (MAC_LAYER4_ADDRESS4)

The MAC_LAYER4_ADDRESS(#i), MAC_L3_L4_CONTROL(#i), MAC_LAYER3_ADDR0_Reg(#i), MAC_LAYER3_ADDR1_Reg(#i), MAC_LAYER3_ADDR2_Reg(#i) and MAC_LAYER3_ADDR3_Reg(#i) registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

11.7.6.1.302.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS4	9C4h

11.7.6.1.302.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.302.3 Fields

Field	Description
31-16 L4DP4	Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field

Table continues on the next page...

Field	Description
	When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4DPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP4	Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

11.7.6.1.303 Layer 3 Address 0 Register 4 (MAC_LAYER3_ADDR0_REG4)

For IPv4 packets, the Layer 3 Address 0 Register 4 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.303.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG4	9D0h

11.7.6.1.303.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A04															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A04															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.303.3 Fields

Field	Description
31-0 L3A04	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

11.7.6.1.304 Layer 3 Address 1 Register 4 (MAC_LAYER3_ADDR1_REG4)

For IPv4 packets, the Layer 3 Address 1 Register 4 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.304.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG4	9D4h

11.7.6.1.304.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A14															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A14															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.304.3 Fields

Field	Description
31-0 L3A14	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

11.7.6.1.305 Layer 3 Address 2 Register 4 (MAC_LAYER3_ADDR2_REG4)

The Layer 3 Address 2 Register 4 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.305.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG4	9D8h

11.7.6.1.305.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A24															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A24															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.305.3 Fields

Field	Description
31-0 L3A24	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

11.7.6.1.306 Layer 3 Address 3 Register 4 (MAC_LAYER3_ADDR3_REG4)

The Layer 3 Address 3 Register 4 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.306.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG4	9DCh

11.7.6.1.306.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A34															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A34															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.306.3 Fields

Field	Description
31-0 L3A34	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

11.7.6.1.307 Layer 3 and Layer 4 Control of Filter 5 (MAC_L3_L4_CONTROL5)

The Layer 3 and Layer 4 Control register controls the operations of filter 0 of Layer 3 and Layer 4.

11.7.6.1.307.1 Offset

Register	Offset
MAC_L3_L4_CONTROL5	9F0h

11.7.6.1.307.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN5	Reserved	DMCHN5			Reserved		L4DPIM5	L4DPM5	L4SPIM5	L4SPM5	Reserved	L4PEN5
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM5					L3HSBM5					L3DAIM5	L3DAM5	L3SAIM5	L3SAM5	Reserved	L3PEN5
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.307.3 Fields

Field	Description
31-29 —	Reserved
28 DMCHEN5	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN5	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM5	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM0 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM5	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM5	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high.

Table continues on the next page...

Field	Description
	0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM5	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN5	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM0 bit is set. 0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM5	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.

Table continues on the next page...

Field	Description
10-6 L3HSBM5	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM5	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM5	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM5	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2 L3SAM5	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable

Table continues on the next page...

Field	Description
	When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN5	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

11.7.6.1.308 Layer 4 Address 5 (MAC_LAYER4_ADDRESS5)

The MAC_LAYER4_ADDRESS(#i), MAC_L3_L4_CONTROL(#i), MAC_LAYER3_ADDR0_Reg(#i), MAC_LAYER3_ADDR1_Reg(#i), MAC_LAYER3_ADDR2_Reg(#i) and MAC_LAYER3_ADDR3_Reg(#i) registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

11.7.6.1.308.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS5	9F4h

11.7.6.1.308.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.308.3 Fields

Field	Description
31-16 L4DP5	Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4DPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP5	Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

11.7.6.1.309 Layer 3 Address 0 Register 5 (MAC_LAYER3_ADDR0_REG5)

For IPv4 packets, the Layer 3 Address 0 Register 5 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.309.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG5	A00h

11.7.6.1.309.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A05															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A05															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.309.3 Fields

Field	Description
31-0 L3A05	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

11.7.6.1.310 Layer 3 Address 1 Register 5 (MAC_LAYER3_ADDR1_REG5)

For IPv4 packets, the Layer 3 Address 1 Register 5 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.310.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG5	A04h

11.7.6.1.310.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A15															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A15															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.310.3 Fields

Field	Description
31-0 L3A15	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

11.7.6.1.311 Layer 3 Address 2 Register 5 (MAC_LAYER3_ADDR2_REG5)

The Layer 3 Address 2 Register 5 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.311.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG5	A08h

11.7.6.1.311.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A25															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A25															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.311.3 Fields

Field	Description
31-0 L3A25	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

11.7.6.1.312 Layer 3 Address 3 Register 5 (MAC_LAYER3_ADDR3_REG5)

The Layer 3 Address 3 Register 5 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.312.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG5	A0Ch

11.7.6.1.312.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A35															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A35															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.312.3 Fields

Field	Description
31-0 L3A35	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

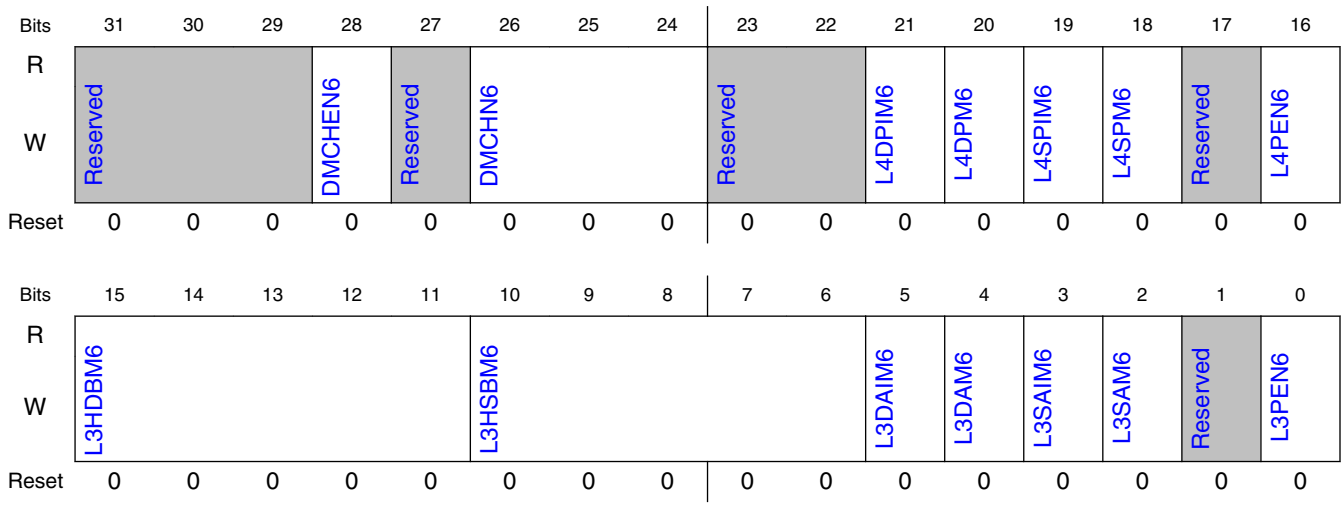
11.7.6.1.313 Layer 3 and Layer 4 Control of Filter 6 (MAC_L3_L4_CONTROL6)

The Layer 3 and Layer 4 Control register controls the operations of filter 0 of Layer 3 and Layer 4.

11.7.6.1.313.1 Offset

Register	Offset
MAC_L3_L4_CONTROL6	A20h

11.7.6.1.313.2 Diagram



11.7.6.1.313.3 Fields

Field	Description
31-29 —	Reserved
28 DMCHEN6	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN6	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number

Table continues on the next page...

Field	Description
	When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM6	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM0 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM6	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM6	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM6	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN6	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching.

Table continues on the next page...

Field	Description
	The Layer 4 matching is done only when the L4SPM0 or L4DPM0 bit is set. 0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM6	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM6	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM6	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable

Table continues on the next page...

Field	Description
	When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM6	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM6	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2 L3SAM6	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN6	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

11.7.6.1.314 Layer 4 Address 6 (MAC_LAYER4_ADDRESS6)

The MAC_LAYER4_ADDRESS(#i), MAC_L3_L4_CONTROL(#i), MAC_LAYER3_ADDR0_Reg(#i), MAC_LAYER3_ADDR1_Reg(#i), MAC_LAYER3_ADDR2_Reg(#i) and MAC_LAYER3_ADDR3_Reg(#i) registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

11.7.6.1.314.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS6	A24h

11.7.6.1.314.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP6															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP6															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.314.3 Fields

Field	Description
31-16 L4DP6	Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4DPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP6	Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

11.7.6.1.315 Layer 3 Address 0 Register 6 (MAC_LAYER3_ADDR0_REG6)

For IPv4 packets, the Layer 3 Address 0 Register 6 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.315.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG6	A30h

11.7.6.1.315.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A06															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A06															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.315.3 Fields

Field	Description
31-0 L3A06	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

11.7.6.1.316 Layer 3 Address 1 Register 6 (MAC_LAYER3_ADDR1_REG6)

For IPv4 packets, the Layer 3 Address 1 Register 6 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.316.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG6	A34h

11.7.6.1.316.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A16															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A16															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.316.3 Fields

Field	Description
31-0 L3A16	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

11.7.6.1.317 Layer 3 Address 2 Register 6 (MAC_LAYER3_ADDR2_REG6)

The Layer 3 Address 2 Register 6 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.317.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG6	A38h

11.7.6.1.317.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A26															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A26															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.317.3 Fields

Field	Description
31-0 L3A26	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

11.7.6.1.318 Layer 3 Address 3 Register 6 (MAC_LAYER3_ADDR3_REG6)

The Layer 3 Address 3 Register 6 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.318.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG6	A3Ch

11.7.6.1.318.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A36															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A36															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.318.3 Fields

Field	Description
31-0 L3A36	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

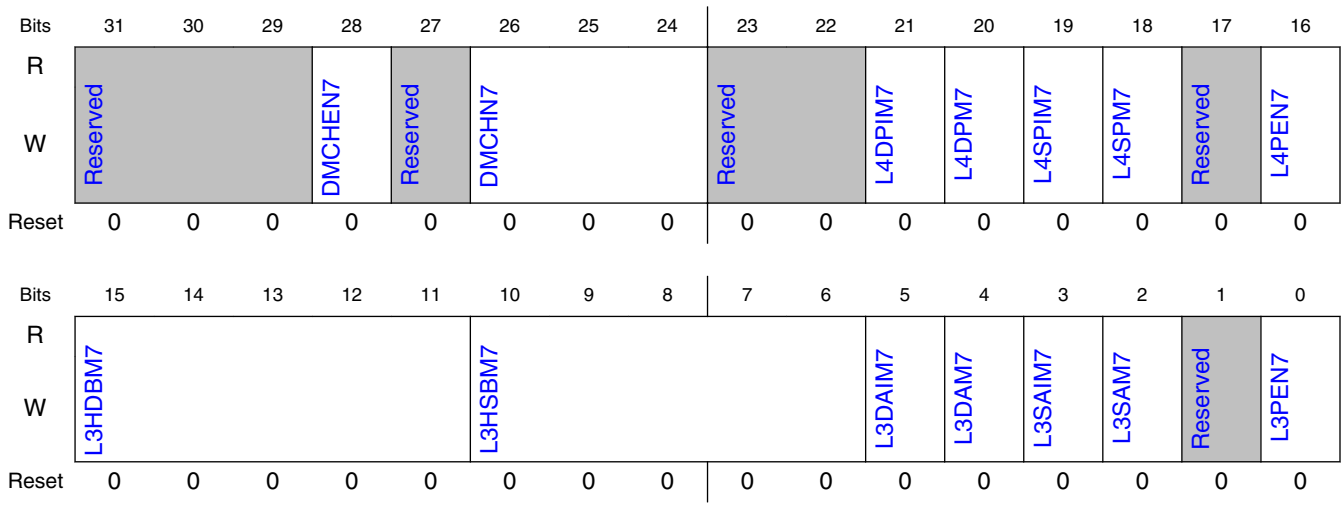
11.7.6.1.319 Layer 3 and Layer 4 Control of Filter 0 (MAC_L3_L4_CONTROL7)

The Layer 3 and Layer 4 Control register controls the operations of filter 0 of Layer 3 and Layer 4.

11.7.6.1.319.1 Offset

Register	Offset
MAC_L3_L4_CONTROL 7	A50h

11.7.6.1.319.2 Diagram



11.7.6.1.319.3 Fields

Field	Description
31-29 —	Reserved
28 DMCHEN7	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0 - DMA Channel Select is disabled 1 - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN7	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM7	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable

Table continues on the next page...

Field	Description
	When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM0 bit is set high. 0 - Layer 4 Destination Port Inverse Match is disabled 1 - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM7	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0 - Layer 4 Destination Port Match is disabled 1 - Layer 4 Destination Port Match is enabled
19 L4SPIM7	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0 - Layer 4 Source Port Inverse Match is disabled 1 - Layer 4 Source Port Inverse Match is enabled
18 L4SPM7	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0 - Layer 4 Source Port Match is disabled 1 - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN7	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM0 bit is set. 0 - Layer 4 Protocol is disabled 1 - Layer 4 Protocol is enabled
15-11 L3HDBM7	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets:

Table continues on the next page...

Field	Description
	This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM7	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM7	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0 - Layer 3 IP DA Inverse Match is disabled 1 - Layer 3 IP DA Inverse Match is enabled
4 L3DAM7	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching.

Table continues on the next page...

Field	Description
	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0 - Layer 3 IP DA Match is disabled 1 - Layer 3 IP DA Match is enabled
3 L3SAIM7	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0 - Layer 3 IP SA Inverse Match is disabled 1 - Layer 3 IP SA Inverse Match is enabled
2 L3SAM7	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0 - Layer 3 IP SA Match is disabled 1 - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN7	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0 - Layer 3 Protocol is disabled 1 - Layer 3 Protocol is enabled

11.7.6.1.320 Layer 4 Address 7 (MAC_LAYER4_ADDRESS7)

The MAC_LAYER4_ADDRESS(#i), MAC_L3_L4_CONTROL(#i), MAC_LAYER3_ADDR0_Reg(#i), MAC_LAYER3_ADDR1_Reg(#i), MAC_LAYER3_ADDR2_Reg(#i) and MAC_LAYER3_ADDR3_Reg(#i) registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

11.7.6.1.320.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS7	A54h

11.7.6.1.320.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP7															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP7															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.320.3 Fields

Field	Description
31-16 L4DP7	Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field

Table continues on the next page...

Field	Description
	When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4DPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP7	Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

11.7.6.1.321 Layer 3 Address 0 Register 7 (MAC_LAYER3_ADDR0_REG7)

For IPv4 packets, the Layer 3 Address 0 Register 7 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.321.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG7	A60h

11.7.6.1.321.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A07															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A07															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.321.3 Fields

Field	Description
31-0 L3A07	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

11.7.6.1.322 Layer 3 Address 1 Register 7 (MAC_LAYER3_ADDR1_REG7)

For IPv4 packets, the Layer 3 Address 1 Register 7 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

11.7.6.1.322.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG7	A64h

11.7.6.1.322.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A17															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A17															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.322.3 Fields

Field	Description
31-0 L3A17	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

11.7.6.1.323 Layer 3 Address 2 Register 7 (MAC_LAYER3_ADDR2_REG7)

The Layer 3 Address 2 Register 7 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.323.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG7	A68h

11.7.6.1.323.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A27															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A27															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.323.3 Fields

Field	Description
31-0 L3A27	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

11.7.6.1.324 Layer 3 Address 3 Register 7 (MAC_LAYER3_ADDR3_REG7)

The Layer 3 Address 3 Register 7 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

11.7.6.1.324.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG7	A6Ch

11.7.6.1.324.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A37															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A37															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.324.3 Fields

Field	Description
31-0 L3A37	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

11.7.6.1.325 Timestamp Control (MAC_TIMESTAMP_CONTROL)

This register controls the operation of the System Time generator and processing of PTP packets for timestamping in the Receiver.

11.7.6.1.325.1 Offset

Register	Offset
MAC_TIMESTAMP_CONTROL	B00h

11.7.6.1.325.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			AV8021ASMEN	Reserved			TXTSSTSM	Reserved			ESTI	CSC	TSENMACADDR	SNAPTYPSEL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSMSTRENA	TSEVNTENA	TSIPV4ENA	TSIPV6ENA	TSIPENA	TSVER2ENA	TSCTRLSSR	TSENALL	Reserved	PTGE	TSADDREG	Reserved	TSUPDT	TSINIT	TSCFUPDT	TSENA
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.325.3 Fields

Field	Description
31-29 —	Reserved
28 AV8021ASMEN	AV 802. AV 802.1AS Mode Enable When this bit is set, the MAC processes only untagged PTP over Ethernet packets for providing PTP status and capturing timestamp snapshots, that is, IEEE 802.1AS mode of operation. When PTP offload feature is enabled, for the purpose of PTP offload, the transport specific field in the PTP header is generated and checked based on the value of this bit. 0 - AV 802.1AS Mode is disabled 1 - AV 802.1AS Mode is enabled
27-25 —	Reserved
24 TXTSSTSM	Transmit Timestamp Status Mode When this bit is set, the MAC overwrites the earlier transmit timestamp status even if it is not read by the software. Transmit Timestamp Status Mode When this bit is set, the MAC overwrites the earlier transmit timestamp status even if it is not read by the software. The MAC indicates this by setting the TXTSSMIS bit of the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS register. When this bit is reset, the MAC ignores the timestamp status of current packet if the timestamp status of previous packet is not read by the software. The MAC indicates this by setting the TXTSSMIS bit of the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS register. 0 - Transmit Timestamp Status Mode is disabled 1 - Transmit Timestamp Status Mode is enabled
23-21 —	Reserved
20 ESTI	External System Time Input When this bit is set, the MAC uses the external 64-bit reference System Time input for the following: - To take the timestamp provided as status - To insert the timestamp in transmit PTP packets when One-step Timestamp or Timestamp Offload feature is enabled. External System Time Input When this bit is set, the MAC uses the external 64-bit reference System Time input for the following: - To take the timestamp provided as status - To insert the timestamp in transmit PTP packets when One-step Timestamp or Timestamp Offload feature is enabled. When this bit is reset, the MAC uses the internal reference System Time. 0 - External System Time Input is disabled 1 - External System Time Input is enabled
19 CSC	Enable checksum correction during OST for PTP over UDP/IPv4 packets When this bit is set, the last two bytes of PTP message sent over UDP/IPv4 is updated to keep the UDP checksum correct, for changes made to origin timestamp and/or correction field as part of one step timestamp operation. Enable checksum correction during OST for PTP over UDP/IPv4 packets

Table continues on the next page...

Field	Description
	When this bit is set, the last two bytes of PTP message sent over UDP/IPv4 is updated to keep the UDP checksum correct, for changes made to origin timestamp and/or correction field as part of one step timestamp operation. The application shall form the packet with these two dummy bytes. When reset, no updates are done to keep the UDP checksum correct. The application shall form the packet with UDP checksum set to 0. 0 - checksum correction during OST for PTP over UDP/IPv4 packets is disabled 1 - checksum correction during OST for PTP over UDP/IPv4 packets is enabled
18 TSENMACADDR	Enable MAC Address for PTP Packet Filtering When this bit is set, the DA MAC address (that matches any MAC Address register) is used to filter the PTP packets when PTP is directly sent over Ethernet. Enable MAC Address for PTP Packet Filtering When this bit is set, the DA MAC address (that matches any MAC Address register) is used to filter the PTP packets when PTP is directly sent over Ethernet. When this bit is set, received PTP packets with DA containing a special multicast or unicast address that matches the one programmed in MAC address registers are considered for processing as indicated below, when PTP is directly sent over Ethernet. For normal time stamping operation, MAC address registers 0 to 31 is considered for unicast destination address matching. For PTP offload, only MAC address register 0 is considered for unicast destination address matching. 0 - MAC Address for PTP Packet Filtering is disabled 1 - MAC Address for PTP Packet Filtering is enabled
17-16 SNAPTYPSEL	Select PTP packets for Taking Snapshots These bits, along with Bits 15 and 14, decide the set of PTP packet types for which snapshot needs to be taken. Select PTP packets for Taking Snapshots These bits, along with Bits 15 and 14, decide the set of PTP packet types for which snapshot needs to be taken. The encoding is given in Timestamp Snapshot Dependency on Register Bits Table.
15 TSMSTRENA	Enable Snapshot for Messages Relevant to Master When this bit is set, the snapshot is taken only for the messages that are relevant to the master node. Enable Snapshot for Messages Relevant to Master When this bit is set, the snapshot is taken only for the messages that are relevant to the master node. Otherwise, the snapshot is taken for the messages relevant to the slave node. 0 - Snapshot for Messages Relevant to Master is disabled 1 - Snapshot for Messages Relevant to Master is enabled
14 TSEVNTENA	Enable Timestamp Snapshot for Event Messages When this bit is set, the timestamp snapshot is taken only for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). Enable Timestamp Snapshot for Event Messages When this bit is set, the timestamp snapshot is taken only for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). When this bit is reset, the snapshot is taken for all messages except Announce, Management, and Signaling. For more information about the timestamp snapshots, see Timestamp Snapshot Dependency on Register Bits Table. 0 - Timestamp Snapshot for Event Messages is disabled 1 - Timestamp Snapshot for Event Messages is enabled
13 TSIPV4ENA	Enable Processing of PTP Packets Sent over IPv4-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv4-UDP packets. Enable Processing of PTP Packets Sent over IPv4-UDP

Table continues on the next page...

Field	Description
	When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv4-UDP packets. When this bit is reset, the MAC ignores the PTP transported over IPv4-UDP packets. This bit is set by default. 0 - Processing of PTP Packets Sent over IPv4-UDP is disabled 1 - Processing of PTP Packets Sent over IPv4-UDP is enabled
12 TSIPV6ENA	Enable Processing of PTP Packets Sent over IPv6-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv6-UDP packets. Enable Processing of PTP Packets Sent over IPv6-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv6-UDP packets. When this bit is clear, the MAC ignores the PTP transported over IPv6-UDP packets. 0 - Processing of PTP Packets Sent over IPv6-UDP is disabled 1 - Processing of PTP Packets Sent over IPv6-UDP is enabled
11 TSIPENA	Enable Processing of PTP over Ethernet Packets When this bit is set, the MAC receiver processes the PTP packets encapsulated directly in the Ethernet packets. Enable Processing of PTP over Ethernet Packets When this bit is set, the MAC receiver processes the PTP packets encapsulated directly in the Ethernet packets. When this bit is reset, the MAC ignores the PTP over Ethernet packets. 0 - Processing of PTP over Ethernet Packets is disabled 1 - Processing of PTP over Ethernet Packets is enabled
10 TSVER2ENA	Enable PTP Packet Processing for Version 2 Format When this bit is set, the IEEE 1588 version 2 format is used to process the PTP packets. Enable PTP Packet Processing for Version 2 Format When this bit is set, the IEEE 1588 version 2 format is used to process the PTP packets. When this bit is reset, the IEEE 1588 version 1 format is used to process the PTP packets. The IEEE 1588 formats are described in 'PTP Processing and Control'. 0 - PTP Packet Processing for Version 2 Format is disabled 1 - PTP Packet Processing for Version 2 Format is enabled
9 TSCTRLSSR	Timestamp Digital or Binary Rollover Control When this bit is set, the Timestamp Low register rolls over after 0x3B9A_C9FF value (that is, 1 nanosecond accuracy) and increments the timestamp (High) seconds. Timestamp Digital or Binary Rollover Control When this bit is set, the Timestamp Low register rolls over after 0x3B9A_C9FF value (that is, 1 nanosecond accuracy) and increments the timestamp (High) seconds. When this bit is reset, the rollover value of sub-second register is 0x7FFF_FFFF. The sub-second increment must be programmed correctly depending on the PTP reference clock frequency and the value of this bit. 0 - Timestamp Digital or Binary Rollover Control is disabled 1 - Timestamp Digital or Binary Rollover Control is enabled
8 TSENALL	Enable Timestamp for All Packets When this bit is set, the timestamp snapshot is enabled for all packets received by the MAC. 0 - Timestamp for All Packets disabled 1 - Timestamp for All Packets enabled
7 —	Reserved

Table continues on the next page...

Field	Description
6 PTGE	Presentation Time Generation Enable When this bit is set the Presentation Time generation will be enabled. 0 - Presentation Time Generation is disabled 1 - Presentation Time Generation is enabled
5 TSADDREG	Update Addend Register When this bit is set, the content of the Timestamp Addend register is updated in the PTP block for fine correction. Update Addend Register When this bit is set, the content of the Timestamp Addend register is updated in the PTP block for fine correction. This bit is cleared when the update is complete. This bit should be zero before it is set. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Addend Register is not updated 1 - Addend Register is updated
4 —	Reserved
3 TSUPDT	Update Timestamp When this bit is set, the system time is updated (added or subtracted) with the value specified in MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers. Update Timestamp When this bit is set, the system time is updated (added or subtracted) with the value specified in MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers. This bit should be zero before updating it. This bit is reset when the update is complete in hardware. The Timestamp Higher Word register (if enabled during core configuration) is not updated. When Media Clock Generation and Recovery is configured (DWC_EQOS_FLEXI_PPS_OUT_EN) and enabled MAC_Presn_Time_Updt should also be updated before setting this field. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Timestamp is not updated 1 - Timestamp is updated
2 TSINIT	Initialize Timestamp When this bit is set, the system time is initialized (overwritten) with the value specified in the MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers. Initialize Timestamp When this bit is set, the system time is initialized (overwritten) with the value specified in the MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers. This bit should be zero before it is updated. This bit is reset when the initialization is complete. The Timestamp Higher Word register (if enabled during core configuration) can only be initialized. When Media Clock Generation and Recovery is configured (DWC_EQOS_FLEXI_PPS_OUT_EN) and enabled MAC_Presn_Time_Updt should also be updated before setting this field. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Timestamp is not initialized 1 - Timestamp is initialized
1 TSCFUPDT	Fine or Coarse Timestamp Update When this bit is set, the Fine method is used to update system timestamp. Fine or Coarse Timestamp Update

Table continues on the next page...

Field	Description
	When this bit is set, the Fine method is used to update system timestamp. When this bit is reset, Coarse method is used to update the system timestamp. 0 - Coarse method is used to update system timestamp 1 - Fine method is used to update system timestamp
0 TSENA	Enable Timestamp When this bit is set, the timestamp is added for Transmit and Receive packets. Enable Timestamp When this bit is set, the timestamp is added for Transmit and Receive packets. When disabled, timestamp is not added for transmit and receive packets and the Timestamp Generator is also suspended. You need to initialize the Timestamp (system time) after enabling this mode. On the Receive side, the MAC processes the 1588 packets only if this bit is set. 0 - Timestamp is disabled 1 - Timestamp is enabled

11.7.6.1.326 Subsecond Increment (MAC_SUB_SECOND_INCREMENT)

This register specifies the value to be added to the internal system time register every cycle of clk_ptp_ref_i clock.

11.7.6.1.326.1 Offset

Register	Offset
MAC_SUB_SECOND_INCREMENT	B04h

11.7.6.1.326.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								SSINC							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SNSINC								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.326.3 Fields

Field	Description
31-24 —	Reserved
23-16 SSINC	Sub-second Increment Value The value programmed in this field is accumulated every clock cycle (of clk_ptp_i) with the contents of the sub-second register. Sub-second Increment Value The value programmed in this field is accumulated every clock cycle (of clk_ptp_i) with the contents of the sub-second register. For example, when the PTP clock is 50 MHz (period is 20 ns), you should program 20 (0x14) when the System Time Nanoseconds register has an accuracy of 1 ns [Bit 9 (TSCTRLSSR) is set in TIMESTAMP_CONTROL]. When TSCTRLSSR is clear, the Nanoseconds register has a resolution of ~0.465 ns. In this case, you should program a value of 43 (0x2B) which is derived by 20 ns/0.465.
15-8 SNSINC	Sub-nanosecond Increment Value This field contains the sub-nanosecond increment value, represented in nanoseconds multiplied by 2^8. Sub-nanosecond Increment Value This field contains the sub-nanosecond increment value, represented in nanoseconds multiplied by 2^8. This value is accumulated with the sub-nanoseconds field of the subsecond register. For example, when TSCTRLSSR field in the TIMESTAMP_CONTROL register is set. and if the required increment is 5.3ns, then SSINC should be 0x05 and SNSINC should be 0x4C.
7-0 —	Reserved

11.7.6.1.327 System Time Seconds (MAC_SYSTEM_TIME_SECONDS)

The System Time Seconds register, along with System Time Nanoseconds register, indicates the current value of the system time maintained by the MAC. Though it is updated on a continuous basis, there is some delay from the actual time because of clock domain transfer latencies (from clk_ptp_ref_i to CSR clock).

11.7.6.1.327.1 Offset

Register	Offset
MAC_SYSTEM_TIME_SECONDS	B08h

11.7.6.1.327.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TSS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.327.3 Fields

Field	Description
31-0 TSS	Timestamp Second The value in this field indicates the current value in seconds of the System Time maintained by the MAC.

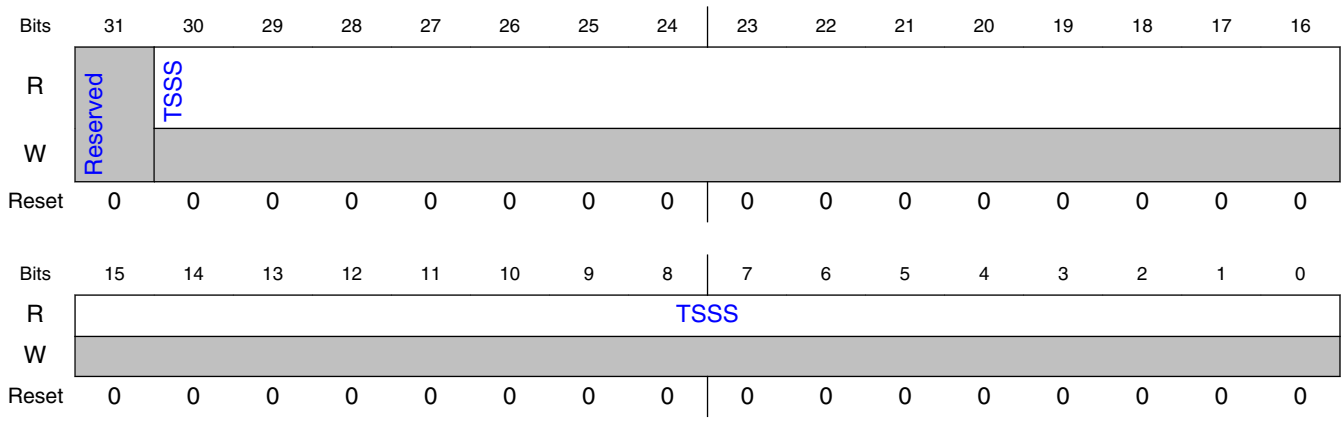
11.7.6.1.328 System Time Nanoseconds (MAC_SYSTEM_TIME_NANOSECONDS)

The System Time Nanoseconds register, along with System Time Seconds register, indicates the current value of the system time maintained by the MAC.

11.7.6.1.328.1 Offset

Register	Offset
MAC_SYSTEM_TIME_NANOSECONDS	B0Ch

11.7.6.1.328.2 Diagram



11.7.6.1.328.3 Fields

Field	Description
31 —	Reserved
30-0 TSSS	Timestamp Sub Seconds The value in this field has the sub-second representation of time, with an accuracy of 0. Timestamp Sub Seconds The value in this field has the sub-second representation of time, with an accuracy of 0.46 ns. When Bit 9 is set in TIMESTAMP_CONTROL, each bit represents 1 ns. The maximum value is 0x3B9A_C9FF after which it rolls-over to zero.

11.7.6.1.329 System Time Seconds Update (MAC_SYSTEM_TIME_SECONDS_UPDATE)

The System Time Seconds Update register, along with the System Time Nanoseconds Update register, initializes or updates the system time maintained by the MAC. You must write both registers before setting the TSINIT or TSUPDT bits in DWC_eqos_top_map/EQOS_MAC/MAC_TIMESTAMP_CONTROL.

11.7.6.1.329.1 Offset

Register	Offset
MAC_SYSTEM_TIME_SECONDS_UPDATE	B10h

11.7.6.1.329.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.329.3 Fields

Field	Description
31-0 TSS	Timestamp Seconds The value in this field is the seconds part of the update. Timestamp Seconds The value in this field is the seconds part of the update. When ADDSUB is reset, this field must be programmed with the seconds part of the update value. When ADDSUB is set, this field must be programmed with the complement of the seconds part of the update value. For example, if 2.000000001 seconds need to be subtracted from the system time, the TSS field in the MAC_Timestamp_Seconds_Update register must be 0xFFFF_FFFE (that is, 2^32 - 2).

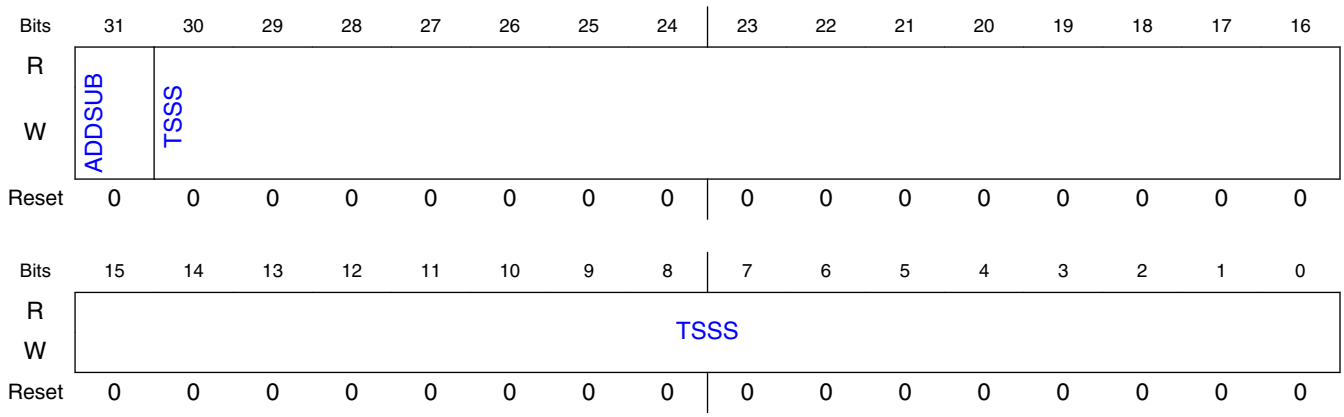
11.7.6.1.330 System Time Nanoseconds Update (MAC_SYSTEM_TIME_NANOSECONDS_UPDATE)

MAC System Time Nanoseconds Update register.

11.7.6.1.330.1 Offset

Register	Offset
MAC_SYSTEM_TIME_NANOSECONDS_UPDATE	B14h

11.7.6.1.330.2 Diagram



11.7.6.1.330.3 Fields

Field	Description
31 ADDSUB	Add or Subtract Time When this bit is set, the time value is subtracted with the contents of the update register. Add or Subtract Time When this bit is set, the time value is subtracted with the contents of the update register. When this bit is reset, the time value is added with the contents of the update register. 0 - Add time 1 - Subtract time
30-0 TSSS	Timestamp Sub Seconds The value in this field is the sub-seconds part of the update. Timestamp Sub Seconds The value in this field is the sub-seconds part of the update. When ADDSUB is reset, this field must be programmed with the sub-seconds part of the update value, with an accuracy based on the TSCTRLSSR bit of the TIMESTAMP_CONTROL register. When ADDSUB is set, this field must be programmed with the complement of the sub-seconds part of the update value as described below. When TSCTRLSSR bit in MAC_TIMESTAMP_CONTROL is set, the programmed value must be 10^9 - <sub-second value>. When TSCTRLSSR bit in MAC_TIMESTAMP_CONTROL is reset, the programmed value must be 2^31 - <sub-second_value>. When the TSCTRLSSR bit is reset in the TMAC_IMESTAMP_CONTROL register, each bit represents an accuracy of 0.46 ns. When the TSCTRLSSR bit is set in the MAC_TIMESTAMP_CONTROL register, each bit represents 1 ns and the programmed value should not exceed 0x3B9A_C9FF. For example, if 2.000000001 seconds need to be subtracted from the system time, then the TSSS field in the MAC_Timestamp_Nanoseconds_Update register must be 0x7FFF_FFFF (that is, 2^31 - 1), when TSCTRLSSR bit in MAC_TIMESTAMP_CONTROL is reset and 0x3B9A_C9FF (that is, 10^9 - 1), when TSCTRLSSR bit in MAC_TIMESTAMP_CONTROL is set.

11.7.6.1.331 Timestamp Addend (MAC_TIMESTAMP_ADDEND)

Timestamp Addend register. This register value is used only when the system time is configured for Fine Update mode (TSCFUPDT bit in the MAC_TIMESTAMP_CONTROL register). The content of this register is added to a 32-bit accumulator in every clock cycle (of clk_ptp_ref_i) and the system time is updated whenever the accumulator overflows.

11.7.6.1.331.1 Offset

Register	Offset
MAC_TIMESTAMP_ADDEND	B18h

11.7.6.1.331.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.331.3 Fields

Field	Description
31-0 TSAR	Timestamp Addend Register This field indicates the 32-bit time value to be added to the Accumulator register to achieve time synchronization.

11.7.6.1.332 System Time - Higher Word Seconds (MAC_SYSTEM_TIME_HIGHER_WORD_SECONDS)

System Time - Higher Word Seconds register.

11.7.6.1.332.1 Offset

Register	Offset
MAC_SYSTEM_TIME_HIGHER_WORD_SECONDS	B1Ch

11.7.6.1.332.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSHWR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.332.3 Fields

Field	Description
31-16 —	Reserved
15-0 TSHWR	Timestamp Higher Word Register This field contains the most-significant 16-bits of timestamp seconds value. Timestamp Higher Word Register This field contains the most-significant 16-bits of timestamp seconds value. This register is optional. You can add this register by selecting the Add IEEE 1588 Higher Word Register option. This register is directly written to initialize the value and it is incremented when there is an overflow from 32-bits of the System Time Seconds register. Access restriction applies. Updated based on the event. Setting 1 sets. Setting 0 clears.

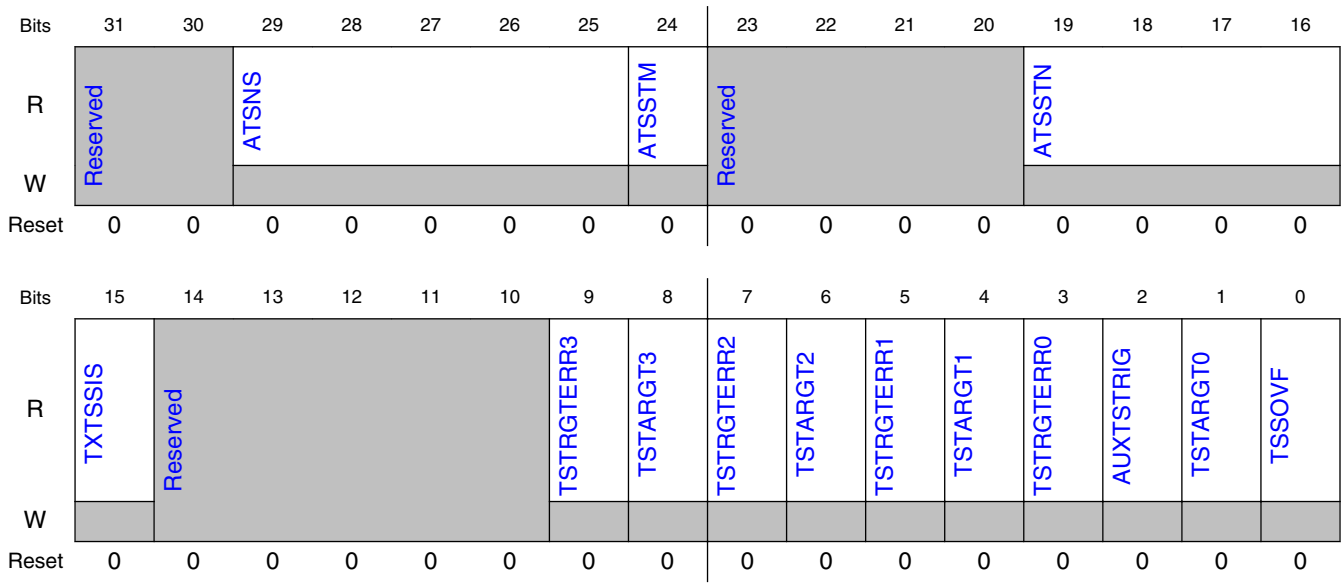
11.7.6.1.333 Timestamp Status (MAC_TIMESTAMP_STATUS)

Timestamp Status register. All bits except Bits[27:25] gets cleared when the application reads this register.

11.7.6.1.333.1 Offset

Register	Offset
MAC_TIMESTAMP_STATUS	B20h

11.7.6.1.333.2 Diagram



11.7.6.1.333.3 Fields

Field	Description
31-30 —	Reserved
29-25 ATSNS	Number of Auxiliary Timestamp Snapshots This field indicates the number of Snapshots available in the FIFO. Number of Auxiliary Timestamp Snapshots This field indicates the number of Snapshots available in the FIFO. A value equal to the selected depth of FIFO (4, 8, or 16) indicates that the Auxiliary Snapshot FIFO is full. These bits are cleared (to 00000) when the Auxiliary snapshot FIFO clear bit is set. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected.
24 ATSSTM	Auxiliary Timestamp Snapshot Trigger Missed This bit is set when the Auxiliary timestamp snapshot FIFO is full and external trigger was set. Auxiliary Timestamp Snapshot Trigger Missed This bit is set when the Auxiliary timestamp snapshot FIFO is full and external trigger was set. This indicates that the latest snapshot is not stored in the FIFO. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected.

Table continues on the next page...

Field	Description
	0 - Auxiliary Timestamp Snapshot Trigger Missed status not detected 1 - Auxiliary Timestamp Snapshot Trigger Missed status detected
23-20 —	Reserved
19-16 ATSSTN	Auxiliary Timestamp Snapshot Trigger Identifier These bits identify the Auxiliary trigger inputs for which the timestamp available in the Auxiliary Snapshot Register is applicable. Auxiliary Timestamp Snapshot Trigger Identifier These bits identify the Auxiliary trigger inputs for which the timestamp available in the Auxiliary Snapshot Register is applicable. When more than one bit is set at the same time, it means that corresponding auxiliary triggers were sampled at the same clock. These bits are applicable only if the number of Auxiliary snapshots is more than one. One bit is assigned for each trigger as shown in the following list: - Bit 16: Auxiliary trigger 0 - Bit 17: Auxiliary trigger 1 - Bit 18: Auxiliary trigger 2 - Bit 19: Auxiliary trigger 3 The software can read this register to find the triggers that are set when the timestamp is taken. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15 TXTSSIS	Tx Timestamp Status Interrupt Status In non-EQOS_CORE configurations when drop transmit status is enabled in MTL, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and MAC_TX_TIMESTAMP_STATUS_SECONDS registers. Tx Timestamp Status Interrupt Status In non-EQOS_CORE configurations when drop transmit status is enabled in MTL, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and TMAC_X_TIMESTAMP_STATUS_SECONDS registers. When PTP offload feature is enabled, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and MAC_TX_TIMESTAMP_STATUS_SECONDS registers, for PTO generated Delay Request and Pdelay request packets. This bit is cleared when the MAC_TX_TIMESTAMP_STATUS_SECONDS register is read (or write to MAC_TX_TIMESTAMP_STATUS_SECONDS register when RCWE bit of CSR_SW_CTRL register is set). 0 - Tx Timestamp Status Interrupt status not detected 1 - Tx Timestamp Status Interrupt status detected
14-10 —	Reserved
9 TSTRGTERR3	Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS registers elapses. Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS registers elapses. This bit is cleared when the application reads this bit. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_CTRL register is set). Self-set to 1 on internal event.

Table continues on the next page...

Field	Description
	0 - Timestamp Target Time Error status not detected 1 - Timestamp Target Time Error status detected
8 TSTARGT3	Timestamp Target Time Reached for Target Time PPS3 When this bit is set, it indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS registers. Timestamp Target Time Reached for Target Time PPS3 When this bit is set, it indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS registers. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Reached for Target Time PPS3 status not detected 1 - Timestamp Target Time Reached for Target Time PPS3 status detected
7 TSTRGTERR2	Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS registers elapses. Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS registers elapses. This bit is cleared when the application reads this bit. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Error status not detected 1 - Timestamp Target Time Error status detected
6 TSTARGT2	Timestamp Target Time Reached for Target Time PPS2 When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS registers. Timestamp Target Time Reached for Target Time PPS2 When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS registers. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Reached for Target Time PPS2 status not detected 1 - Timestamp Target Time Reached for Target Time PPS2 status detected
5 TSTRGTERR1	Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS registers elapses. Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS registers elapses. This bit is cleared when the application reads this bit. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Error status not detected

Table continues on the next page...

Field	Description
	1 - Timestamp Target Time Error status detected
4 TSTARGT1	Timestamp Target Time Reached for Target Time PPS1 When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS registers. Timestamp Target Time Reached for Target Time PPS1 When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS registers. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Reached for Target Time PPS1 status not detected 1 - Timestamp Target Time Reached for Target Time PPS1 status detected
3 TSTRGTERR0	Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS0_Target_Time_Seconds and MAC_PPS0_Target_Time_Nanoseconds registers elapses. Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS0_Target_Time_Seconds and MAC_PPS0_TARGET_TIME_NANOSECONDS registers elapses. This bit is cleared when the application reads this bit. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Error status not detected 1 - Timestamp Target Time Error status detected
2 AUXTSTRIG	Auxiliary Timestamp Trigger Snapshot This bit is set high when the auxiliary snapshot is written to the FIFO. Auxiliary Timestamp Trigger Snapshot This bit is set high when the auxiliary snapshot is written to the FIFO. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Auxiliary Timestamp Trigger Snapshot status not detected 1 - Auxiliary Timestamp Trigger Snapshot status detected
1 TSTARGT0	Timestamp Target Time Reached When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS0_Target_Time_Seconds and MAC_PPS0_Target_Time_Nanoseconds registers. Timestamp Target Time Reached When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS0_TARGET_TIME_SECONDS and MAC_PPS0_TARGET_TIME_NANOSECONDS registers. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Target Time Reached status not detected 1 - Timestamp Target Time Reached status detected
0 TSSOVF	Timestamp Seconds Overflow When this bit is set, it indicates that the seconds value of the timestamp (when supporting version 2 format) has overflowed beyond 32'hFFFF_FFFF.

Field	Description
	Timestamp Seconds Overflow When this bit is set, it indicates that the seconds value of the timestamp (when supporting version 2 format) has overflowed beyond 32'hFFFF_FFFF. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0 - Timestamp Seconds Overflow status not detected 1 - Timestamp Seconds Overflow status detected

11.7.6.1.334 Transmit Timestamp Status Nanoseconds (MAC_TX_TIMESTAMP_STATUS_NANOSECONDS)

This register contains the nanosecond part of timestamp captured for Transmit packets when Tx status is disabled.

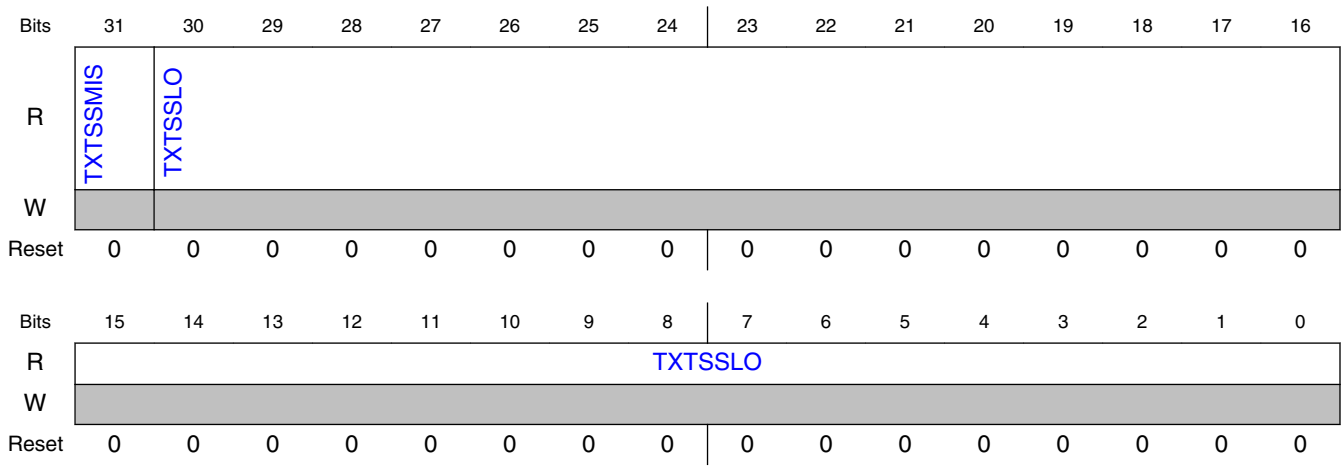
The MAC_TX_TIMESTAMP_STATUS_NANOSECONDS register, along with MAC_TX_TIMESTAMP_STATUS_SECONDS, gives the 64-bit timestamp captured for the PTP packet successfully transmitted by the MAC. This value is considered to be read by the application when the last byte of MAC_TX_TIMESTAMP_STATUS_NANOSECONDS is read. In the little-endian mode, this means when bits[31:24] are read; in big-endian mode, bits[7:0] are read.

If the application does not read these registers and timestamp of another packet is captured, then either the current timestamp is lost (overwritten) or the new timestamp is lost (dropped), depending on the setting of the TXTSSTSM bit of the MAC_TIMESTAMP_CONTROL register. The status bit TXTSC bit [15] in MAC_TIMESTAMP_STATUS register is set whenever the MAC transmitter captures the timestamp.

11.7.6.1.334.1 Offset

Register	Offset
MAC_TX_TIMESTAMP_STATUS_NANOSECONDS	B30h

11.7.6.1.334.2 Diagram



11.7.6.1.334.3 Fields

Field	Description
31 TXTSSMIS	Transmit Timestamp Status Missed When this bit is set, it indicates one of the following: - The timestamp of the current packet is ignored if TXTSSTSM bit of the TIMESTAMP_CONTROL register is reset - The timestamp of the previous packet is overwritten with timestamp of the current packet if TXTSSTSM bit of the MAC_TIMESTAMP_CONTROL register is set. Transmit Timestamp Status Missed When this bit is set, it indicates one of the following: - The timestamp of the current packet is ignored if TXTSSTSM bit of the TIMESTAMP_CONTROL register is reset - The timestamp of the previous packet is overwritten with timestamp of the current packet if TXTSSTSM bit of the MAC_TIMESTAMP_CONTROL register is set. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Transmit Timestamp Status Missed status not detected 1 - Transmit Timestamp Status Missed status detected
30-0 TXTSSLO	Transmit Timestamp Status Low This field contains the 31 bits of the Nanoseconds field of the Transmit packet's captured timestamp.

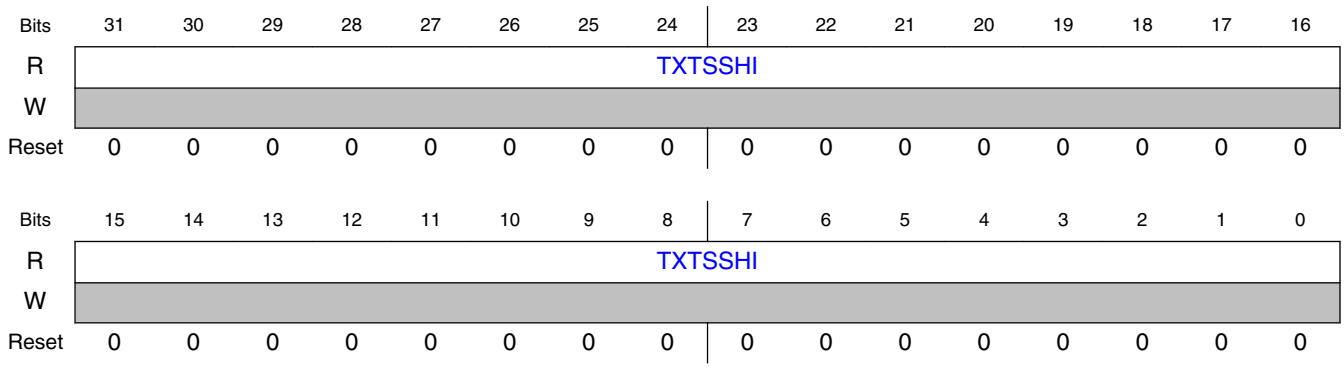
11.7.6.1.335 Transmit Timestamp Status Seconds (MAC_TX_TIMESTAMP_STATUS_SECONDS)

The register contains the higher 32 bits of the timestamp (in seconds) captured when a PTP packet is transmitted.

11.7.6.1.335.1 Offset

Register	Offset
MAC_TX_TIMESTAMP_STATUS_SECONDS	B34h

11.7.6.1.335.2 Diagram



11.7.6.1.335.3 Fields

Field	Description
31-0 TXTSSHI	Transmit Timestamp Status High This field contains the lower 32 bits of the Seconds field of Transmit packet's captured timestamp.

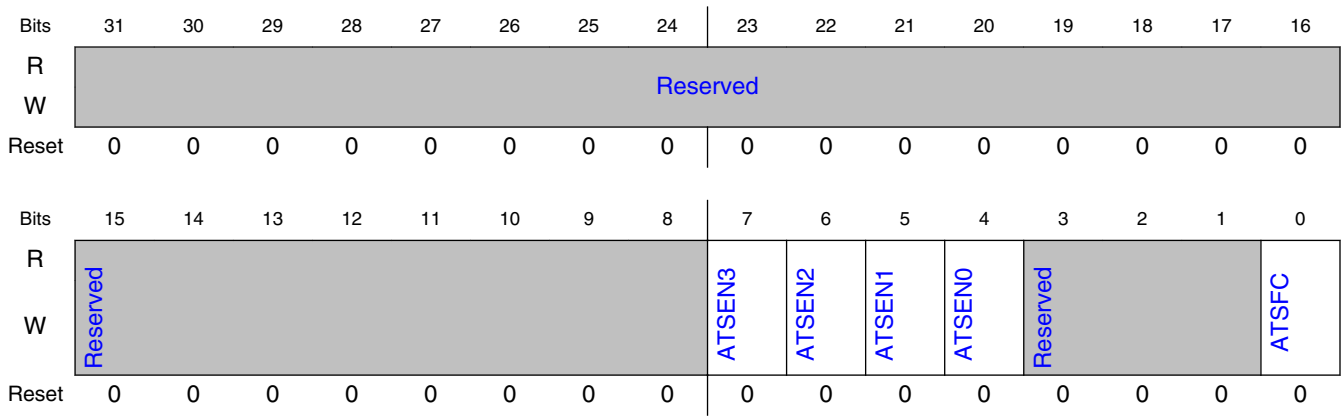
11.7.6.1.336 Auxiliary Timestamp Control (MAC_AUXILIARY_CONTROL)

The Auxiliary Timestamp Control register controls the Auxiliary Timestamp snapshot.

11.7.6.1.336.1 Offset

Register	Offset
MAC_AUXILIARY_CONTROL	B40h

11.7.6.1.336.2 Diagram



11.7.6.1.336.3 Fields

Field	Description
31-8 —	Reserved
7 ATSEN3	Auxiliary Snapshot 3 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 3. Auxiliary Snapshot 3 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 3. When this bit is set, the auxiliary snapshot of the event on ptp_aux_trig_i[3] input is enabled. When this bit is reset, the events on this input are ignored. 0 - Auxiliary Snapshot $i is disabled 1 - Auxiliary Snapshot $i is enabled
6 ATSEN2	Auxiliary Snapshot 2 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 2. Auxiliary Snapshot 2 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 2. When this bit is set, the auxiliary snapshot of the event on ptp_aux_trig_i[2] input is enabled. When this bit is reset, the events on this input are ignored. 0 - Auxiliary Snapshot $i is disabled 1 - Auxiliary Snapshot $i is enabled
5 ATSEN1	Auxiliary Snapshot 1 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 1. Auxiliary Snapshot 1 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 1. When this bit is set, the auxiliary snapshot of the event on ptp_aux_trig_i[1] input is enabled. When this bit is reset, the events on this input are ignored. 0 - Auxiliary Snapshot $i is disabled 1 - Auxiliary Snapshot $i is enabled
4 ATSEN0	Auxiliary Snapshot 0 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 0. Auxiliary Snapshot 0 Enable

Table continues on the next page...

Field	Description
	This bit controls the capturing of Auxiliary Snapshot Trigger 0. When this bit is set, the auxiliary snapshot of the event on ptp_aux_trig_i[0] input is enabled. When this bit is reset, the events on this input are ignored. 0 - Auxiliary Snapshot $i is disabled 1 - Auxiliary Snapshot $i is enabled
3-1 —	Reserved
0 ATSFC	Auxiliary Snapshot FIFO Clear When set, this bit resets the pointers of the Auxiliary Snapshot FIFO. Auxiliary Snapshot FIFO Clear When set, this bit resets the pointers of the Auxiliary Snapshot FIFO. This bit is cleared when the pointers are reset and the FIFO is empty. When this bit is high, the auxiliary snapshots are stored in the FIFO. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Auxiliary Snapshot FIFO Clear is disabled 1 - Auxiliary Snapshot FIFO Clear is enabled

11.7.6.1.337 Auxiliary Timestamp Nanoseconds (MAC_AUXILIARY_TIMESTAMP_NANOSECONDS)

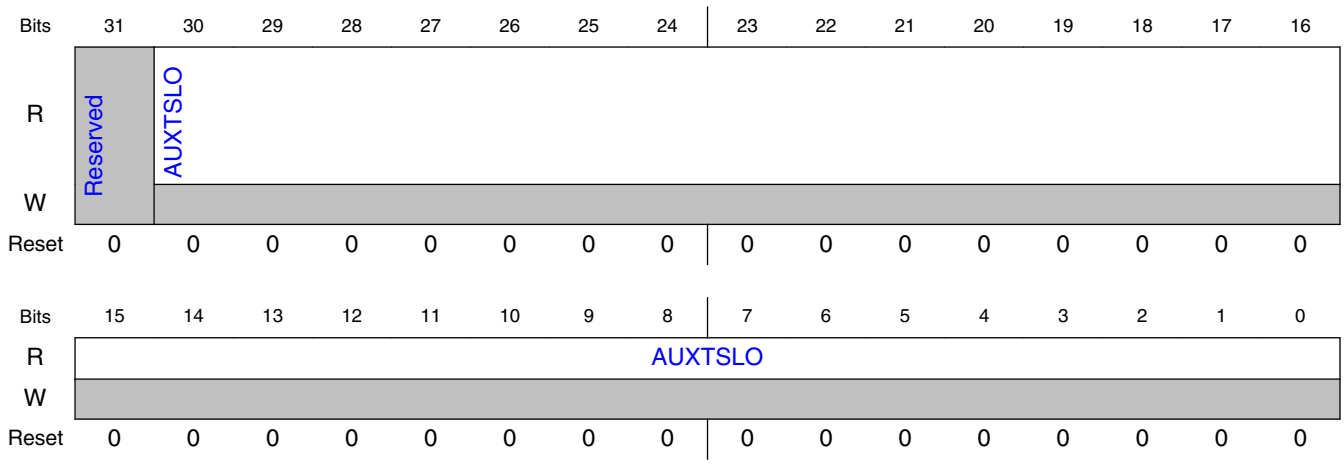
The Auxiliary Timestamp Nanoseconds register, along with MAC_Auxiliary_Timestamp_Seconds, gives the 64-bit timestamp stored as auxiliary snapshot. These two registers form the read port of a 64-bit wide FIFO with a depth of 4, 8, or 16 as selected while configuring the core.

You can store multiple snapshots in this FIFO. Bits[29:25] in MAC_TIMESTAMP_STATUS indicate the fill-level of the FIFO. The top of the FIFO is removed only when the last byte of MAC_AUXILIARY_TIMESTAMP_SECONDS register is read. In the little-endian mode, this means when Bits[31:24] are read and in big-endian mode, Bits[7:0] are read.

11.7.6.1.337.1 Offset

Register	Offset
MAC_AUXILIARY_TIMESTAMP_NANOSECONDS	B48h

11.7.6.1.337.2 Diagram



11.7.6.1.337.3 Fields

Field	Description
31 —	Reserved
30-0 AUXTSLO	Auxiliary Timestamp Contains the lower 31 bits (nanoseconds field) of the auxiliary timestamp.

11.7.6.1.338 Auxiliary Timestamp Seconds (MAC_AUXILIARY_TIMESTAMP_SECONDS)

The Auxiliary Timestamp - Seconds register contains the lower 32 bits of the Seconds field of the auxiliary timestamp register.

11.7.6.1.338.1 Offset

Register	Offset
MAC_AUXILIARY_TIMESTAMP_SECONDS	B4Ch

11.7.6.1.338.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	AUXTSHI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AUXTSHI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.338.3 Fields

Field	Description
31-0 AUXTSHI	Auxiliary Timestamp Contains the lower 32 bits of the Seconds field of the auxiliary timestamp.

11.7.6.1.339 Timestamp Ingress Asymmetry Correction (MAC_TIMESTAMP_INGRESS_ASYM_CORR)

The MAC Timestamp Ingress Asymmetry Correction register contains the Ingress Asymmetry Correction value to be used while updating correction field in PDelay_Resp PTP messages.

11.7.6.1.339.1 Offset

Register	Offset
MAC_TIMESTAMP_INGRESS_ASYM_CORR	B50h

11.7.6.1.339.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OSTIAC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OSTIAC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.339.3 Fields

Field	Description
31-0 OSTIAC	One-Step Timestamp Ingress Asymmetry Correction This field contains the ingress path asymmetry value to be added to correctionField of Pdelay_Resp PTP packet. One-Step Timestamp Ingress Asymmetry Correction This field contains the ingress path asymmetry value to be added to correctionField of Pdelay_Resp PTP packet. The programmed value should be in units of nanoseconds and multiplied by 2^16. For example, 2.5 ns is represented as 0x00028000. The value can also be negative, which is represented in 2's complement form with bit 31 representing the sign bit.

11.7.6.1.340 imestamp Egress Asymmetry Correction (MAC_TIMESTAMP_EGRESS_ASYM_CORR)

The MAC Timestamp Egress Asymmetry Correction register contains the Egress Asymmetry Correction value to be used while updating the correction field in PDelay_Req PTP messages.

11.7.6.1.340.1 Offset

Register	Offset
MAC_TIMESTAMP_EGRESS_ASYM_CORR	B54h

11.7.6.1.340.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OSTEAC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OSTEAC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.340.3 Fields

Field	Description
31-0 OSTEAC	One-Step Timestamp Egress Asymmetry Correction This field contains the egress path asymmetry value to be subtracted from correctionField of Pdelay_Resp PTP packet. One-Step Timestamp Egress Asymmetry Correction This field contains the egress path asymmetry value to be subtracted from correctionField of Pdelay_Resp PTP packet. The programmed value must be the negated value in units of nanoseconds multiplied by 2^16. For example, if the required correction is +2.5 ns, the programmed value must be 0xFFFD_8000, which is the 2's complement of 0x0002_8000(2.5 * 216). Similarly, if the required correction is -3.3 ns, the programmed value is 0x0003_4CCC (3.3 * 216).

11.7.6.1.341 Timestamp Ingress Correction Nanosecond (MAC_TIMESTAMP_INGRESS_CORR_NANOSECOND)

This register contains the correction value in nanoseconds to be used with the captured timestamp value in the ingress path.

11.7.6.1.341.1 Offset

Register	Offset
MAC_TIMESTAMP_INGRESS_CORR_NANOSECOND	B58h

11.7.6.1.341.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSIC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSIC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.341.3 Fields

Field	Description
31-0 TSIC	Timestamp Ingress Correction This field contains the ingress path correction value as defined by the Ingress Correction expression.

11.7.6.1.342 Timestamp Egress Correction Nanosecond (MAC_TIMESTAMP_EGRESS_CORR_NANOSECOND)

This register contains the correction value in nanoseconds to be used with the captured timestamp value in the egress path.

11.7.6.1.342.1 Offset

Register	Offset
MAC_TIMESTAMP_EGRESS_CORR_NANOSECOND	B5Ch

11.7.6.1.342.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.342.3 Fields

Field	Description
31-0 TSEC	Timestamp Egress Correction This field contains the nanoseconds part of the egress path correction value as defined by the Egress Correction expression.

11.7.6.1.343 Timestamp Ingress Correction Subnanosecond (MAC_TIMESTAMP_INGRESS_CORR_SUBNANOSEC)

This register contains the sub-nanosecond part of the correction value to be used with the captured timestamp value, for ingress direction.

11.7.6.1.343.1 Offset

Register	Offset
MAC_TIMESTAMP_INGRESS_CORR_SUBNANOSEC	B60h

11.7.6.1.343.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSICSNS								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.343.3 Fields

Field	Description
31-16 —	Reserved
15-8 TSICSNS	Timestamp Ingress Correction, sub-nanoseconds This field contains the sub-nanoseconds part of the ingress path correction value as defined by the "Ingress Correction" expression.
7-0 —	Reserved

11.7.6.1.344 Timestamp Egress Correction Subnanosecond (MAC_TIMESTAMP_EGRESS_CORR_SUBNANOSEC)

This register contains the sub-nanosecond part of the correction value to be used with the captured timestamp value, for egress direction.

11.7.6.1.344.1 Offset

Register	Offset
MAC_TIMESTAMP_EGRESS_CORR_SUBNANOSEC	B64h

11.7.6.1.344.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSECSNS								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.344.3 Fields

Field	Description
31-16 —	Reserved
15-8 TSECSNS	Timestamp Egress Correction, sub-nanoseconds This field contains the sub-nanoseconds part of the egress path correction value as defined by the "Egress Correction" expression.
7-0 —	Reserved

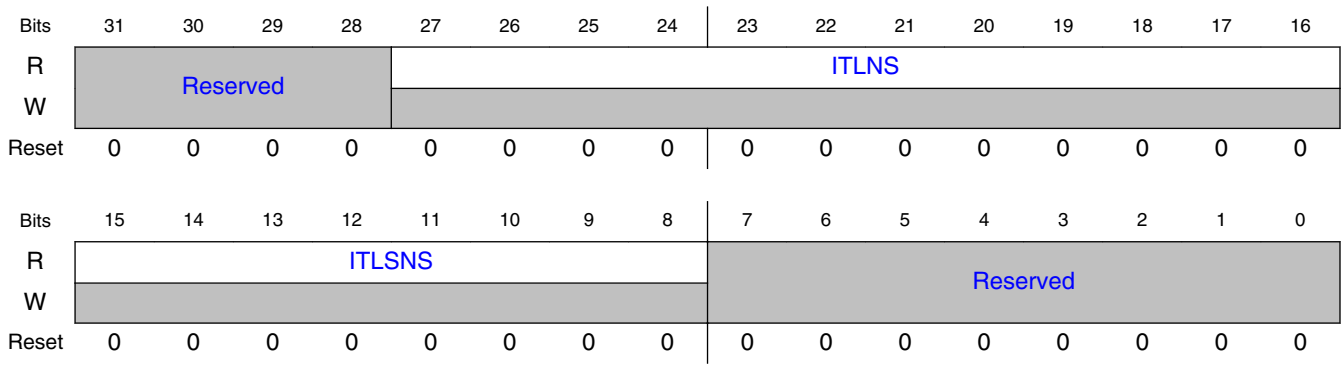
11.7.6.1.345 Timestamp Ingress Latency (MAC_TIMESTAMP_INGRESS_LATENCY)

This register holds the Ingress MAC latency.

11.7.6.1.345.1 Offset

Register	Offset
MAC_TIMESTAMP_INGRESS_LATENCY	B68h

11.7.6.1.345.2 Diagram



11.7.6.1.345.3 Fields

Field	Description
31-28 —	Reserved
27-16 ITLNS	Ingress Timestamp Latency, in sub-nanoseconds This register holds the average latency in sub-nanoseconds between the input ports (phy_rxd_i) of MAC and the actual point (GMII) where the ingress timestamp is taken. Ingress Timestamp Latency, in sub-nanoseconds This register holds the average latency in sub-nanoseconds between the input ports (phy_rxd_i) of MAC and the actual point (GMII) where the ingress timestamp is taken. Ingress correction value is computed as described in the section 7.1.2.4.1 of QoS Databook.
15-8 ITLSNS	Ingress Timestamp Latency, in nanoseconds This register holds the average latency in nanoseconds between the input ports (phy_rxd_i) of MAC and the actual point (GMII) where the ingress timestamp is taken. Ingress Timestamp Latency, in nanoseconds This register holds the average latency in nanoseconds between the input ports (phy_rxd_i) of MAC and the actual point (GMII) where the ingress timestamp is taken. Ingress correction value is computed as described in the section 7.1.2.4.1 of QoS Databook.
7-0 —	Reserved

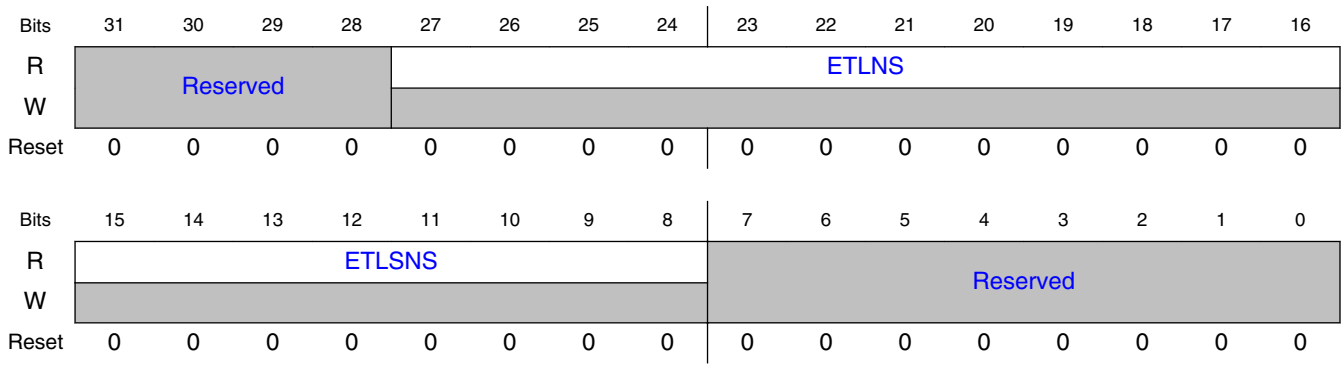
11.7.6.1.346 Timestamp Egress Latency (MAC_TIMESTAMP_EGRESS_LATENCY)

This register holds the Egress MAC latency.

11.7.6.1.346.1 Offset

Register	Offset
MAC_TIMESTAMP_EGRESS_LATENCY	B6Ch

11.7.6.1.346.2 Diagram



11.7.6.1.346.3 Fields

Field	Description
31-28 —	Reserved
27-16 ETLNS	Egress Timestamp Latency, in nanoseconds This register holds the average latency in nanoseconds between the actual point (GMII) where the egress timestamp is taken and the output ports (phy_txd_o) of the MAC. Egress Timestamp Latency, in nanoseconds This register holds the average latency in nanoseconds between the actual point (GMII) where the egress timestamp is taken and the output ports (phy_txd_o) of the MAC. Ingress correction value is computed as described in the section 7.1.2.4.2 of QoS Databook.
15-8 ETLSNS	Egress Timestamp Latency, in sub-nanoseconds This register holds the average latency in sub-nanoseconds between the actual point (GMII) where the egress timestamp is taken and the output ports (phy_txd_o) of the MAC. Egress Timestamp Latency, in sub-nanoseconds This register holds the average latency in sub-nanoseconds between the actual point (GMII) where the egress timestamp is taken and the output ports (phy_txd_o) of the MAC. Ingress correction value is computed as described in the section 7.1.2.4.2 of QoS Databook.
7-0 —	Reserved

11.7.6.1.347 PPS Control (MAC_PPS_CONTROL)

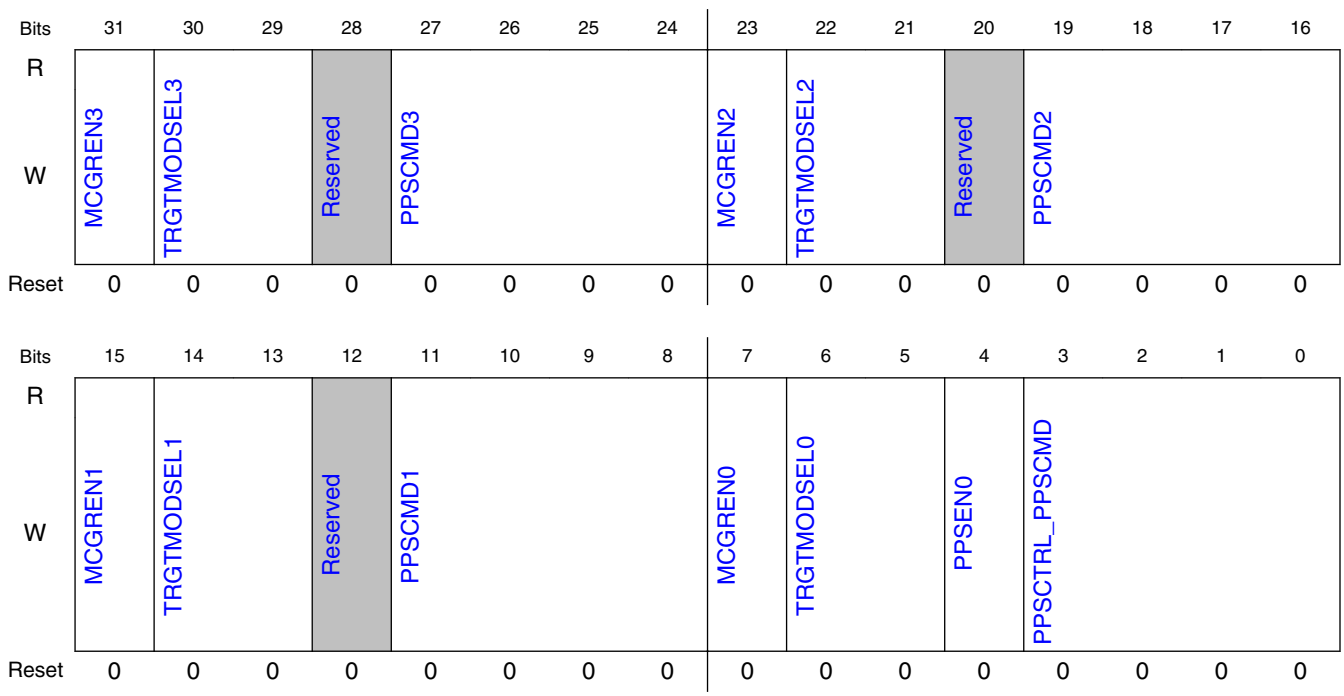
PPS Control register.

Bits[30:24] of this register are valid only when four Flexible PPS outputs are selected. Bits[22:16] are valid only when three or more Flexible PPS outputs are selected. Bits[14:8] are valid only when two or more Flexible PPS outputs are selected. Bits[6:4] are valid only when Flexible PPS feature is selected.

11.7.6.1.347.1 Offset

Register	Offset
MAC_PPS_CONTROL	B70h

11.7.6.1.347.2 Diagram



11.7.6.1.347.3 Fields

Field	Description
31 MCGREN3	MCGR Mode Enable for PPS3 Output This field enables the 3rd PPS instance to operate in PPS or MCGR mode.

Table continues on the next page...

Field	Description
	MCGR Mode Enable for PPS3 Output This field enables the 3rd PPS instance to operate in PPS or MCGR mode. When set it operates in MCGR mode and on reset it operates in PPS mode.
30-29 TRGTMODSEL3	Target Time Register Mode for PPS3 Output This field indicates the Target Time registers (MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS) mode for PPS3 output signal. 00 - Target Time registers are programmed only for generating the interrupt event. The Flexible PPS function must not be enabled in this mode, otherwise spurious transitions may be observed on the corresponding ptp_pps_o output port 01 - Reserved 10 - Target Time registers are programmed for generating the interrupt event and starting or stopping the PPS0 output signal generation 11 - Target Time registers are programmed only for starting or stopping the PPS0 output signal generation. No interrupt is asserted
28 —	Reserved
27-24 PPSCMD3	Flexible PPS3 Output Control This field controls the flexible PPS3 output (ptp_pps_o[3]) signal. Flexible PPS3 Output Control This field controls the flexible PPS3 output (ptp_pps_o[3]) signal. This field is similar to the PPSCMD0[2:0] field. If MCGREN3 is set, then PPSCMD3 indicated by these 4 bits [27:24] are taken as Presentation Time Control bits for media clock generation and recovery for comparator instance 3. This field is similar to the PPSCMD0 Presentation Time Control bits. If MCGREN3 is not set then only 3 bits from [26:24] is used as PPSCMD3 and the 4th bit is to be set as 0. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.
23 MCGREN2	MCGR Mode Enable for PPS2 Output This field enables the 2nd PPS instance to operate in PPS or MCGR mode. MCGR Mode Enable for PPS2 Output This field enables the 2nd PPS instance to operate in PPS or MCGR mode. When set it operates in MCGR mode and on reset it operates in PPS mode. 0 - 2nd PPS instance is disabled to operate in PPS or MCGR mode 1 - 2nd PPS instance is enabled to operate in PPS or MCGR mode
22-21 TRGTMODSEL2	Target Time Register Mode for PPS2 Output This field indicates the Target Time registers (MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS) mode for PPS2 output signal. 00 - Target Time registers are programmed only for generating the interrupt event. The Flexible PPS function must not be enabled in this mode, otherwise spurious transitions may be observed on the corresponding ptp_pps_o output port 01 - Reserved 10 - Target Time registers are programmed for generating the interrupt event and starting or stopping the PPS0 output signal generation 11 - Target Time registers are programmed only for starting or stopping the PPS0 output signal generation. No interrupt is asserted
20 —	Reserved

Table continues on the next page...

Field	Description
19-16 PPSCMD2	Flexible PPS2 Output Control This field controls the flexible PPS2 output (ptp_pps_o[2]) signal. Flexible PPS2 Output Control This field controls the flexible PPS2 output (ptp_pps_o[2]) signal. This field is similar to the PPSCMD0 field. If MCGREN2 is set, then PPSCMD2 indicated by these 4 bits [19:16] are taken as Presentation Time Control bits for media clock generation and recovery for comparator instance 2. This field is similar to the PPSCMD0 Presentation Time Control bits. If MCGREN2 is not set then only 3 bits from [18:16] is used as PPSCMD2 and the 4th bit is to be set as 0. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.
15 MCGREN1	MCGR Mode Enable for PPS1 Output This field enables the 1st PPS instance to operate in PPS or MCGR mode. MCGR Mode Enable for PPS1 Output This field enables the 1st PPS instance to operate in PPS or MCGR mode. When set it operates in MCGR mode and on reset it operates in PPS mode. 0 - 1st PPS instance is disabled to operate in PPS or MCGR mode 1 - 1st PPS instance is enabled to operate in PPS or MCGR mode
14-13 TRGTMODSEL1	Target Time Register Mode for PPS1 Output This field indicates the Target Time registers (MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS) mode for PPS1 output signal. 00 - Target Time registers are programmed only for generating the interrupt event. The Flexible PPS function must not be enabled in this mode, otherwise spurious transitions may be observed on the corresponding ptp_pps_o output port 01 - Reserved 10 - Target Time registers are programmed for generating the interrupt event and starting or stopping the PPS0 output signal generation 11 - Target Time registers are programmed only for starting or stopping the PPS0 output signal generation. No interrupt is asserted
12 —	Reserved
11-8 PPSCMD1	Flexible PPS1 Output Control This field controls the flexible PPS1 output (ptp_pps_o[1]) signal. Flexible PPS1 Output Control This field controls the flexible PPS1 output (ptp_pps_o[1]) signal. This field is similar to the PPSCMD0 field. If MCGREN1 is set, then PPSCMD1 indicated by these 4 bits [11:8] are taken as Presentation Time Control bits for media clock generation and recovery for comparator instance 1. This field is similar to the PPSCMD0 Presentation Time Control bits. If MCGREN1 is not set then only 3 bits from [10:8] is used as PPSCMD1 and the 4th bit is to be set as 0. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.
7 MCGREN0	MCGR Mode Enable for PPS0 Output This field enables the 0th PPS instance to operate in PPS or MCGR mode. MCGR Mode Enable for PPS0 Output This field enables the 0th PPS instance to operate in PPS or MCGR mode. When set it operates in MCGR mode and on reset it operates in PPS mode. 0 - 0th PPS instance is enabled to operate in PPS mode 1 - 0th PPS instance is enabled to operate in MCGR mode

Table continues on the next page...

Field	Description
6-5 TRGTMODSEL0	Target Time Register Mode for PPS0 Output This field indicates the Target Time registers (MAC_PPS0_TARGET_TIME_SECONDS and MAC_PPS0_TARGET_TIME_NANOSECONDS) mode for PPS0 output signal: 00 - Target Time registers are programmed only for generating the interrupt event. The Flexible PPS function must not be enabled in this mode, otherwise spurious transitions may be observed on the corresponding ptp_pps_o output port 01 - Reserved 10 - Target Time registers are programmed for generating the interrupt event and starting or stopping the PPS0 output signal generation 11 - Target Time registers are programmed only for starting or stopping the PPS0 output signal generation. No interrupt is asserted
4 PPSEN0	Flexible PPS Output Mode Enable When this bit is set, Bits[3:0] function as PPSCMD. Flexible PPS Output Mode Enable When this bit is set, Bits[3:0] function as PPSCMD. When this bit is reset, Bits[3:0] function as PPSCTRL (Fixed PPS mode). 0 - Flexible PPS Output Mode is disabled 1 - Flexible PPS Output Mode is enabled
3-0 PPSCTRL_PPSCMD	PPS Output Frequency Control This field controls the frequency of the PPS0 output (ptp_pps_o[0]) signal. PPS Output Frequency Control This field controls the frequency of the PPS0 output (ptp_pps_o[0]) signal. The default value of PPSCTRL is 0000, and the PPS output is 1 pulse (of width clk_ptp_i) every second. For other values of PPSCTRL, the PPS output becomes a generated clock of following frequencies: - 0001: The binary rollover is 2 Hz, and the digital rollover is 1 Hz. - 0010: The binary rollover is 4 Hz, and the digital rollover is 2 Hz. - 0011: The binary rollover is 8 Hz, and the digital rollover is 4 Hz. - 0100: The binary rollover is 16 Hz, and the digital rollover is 8 Hz. - .. - 1111: The binary rollover is 32.768 KHz and the digital rollover is 16.384 KHz. Note: In the binary rollover mode, the PPS output (ptp_pps_o) has a duty cycle of 50 percent with these frequencies. In the digital rollover mode, the PPS output frequency is an average number. The actual clock is of different frequency that gets synchronized every second. For example: - When PPSCTRL = 0001, the PPS (1 Hz) has a low period of 537 ms and a high period of 463 ms - When PPSCTRL = 0010, the PPS (2 Hz) is a sequence of One clock of 50 percent duty cycle and 537 ms period Second clock of 463 ms period (268 ms low and 195 ms high) - When PPSCTRL = 0011, the PPS (4 Hz) is a sequence of Three clocks of 50 percent duty cycle and 268 ms period Fourth clock of 195 ms period (134 ms low and 61 ms high) This behavior is because of the non-linear toggling of bits in the digital rollover mode in the MAC_System_Time_Nanoseconds register.

Field	Description
	or Flexible PPS Output (ptp_pps_o[0]) Control Programming these bits with a non-zero value instructs the MAC to initiate an event. When the command is transferred or synchronized to the PTP clock domain, these bits get cleared automatically. The software should ensure that these bits are programmed only when they are 'all-zero'. The following list describes the values of PPSCMD0: - 0000: No Command - 0001: START Single Pulse This command generates single pulse rising at the start point defined in MAC_PPS0_Target_Time_Seconds and MAC_PPS0_Target_Time_Nanoseconds register and of a duration defined in the PPS0 Width Register. - 0010: START Pulse Train This command generates the train of pulses rising at the start point defined in the Target Time Registers and of a duration defined in the PPS0 Width Register and repeated at interval defined in the PPS Interval Register. By default, the PPS pulse train is free-running unless stopped by the 'Stop Pulse train at time' or 'Stop Pulse Train immediately' commands. - 0011: Cancel START This command cancels the START Single Pulse and START Pulse Train commands if the system time has not crossed the programmed start time. - 0100: STOP Pulse train at time This command stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 0010) after the time programmed in the Target Time registers elapses. - 0101: STOP Pulse Train immediately This command immediately stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 0010). - 110: Cancel STOP Pulse train This command cancels the STOP pulse train at time command if the programmed stop time has not elapsed. The PPS pulse train becomes free-running on the successful execution of this command. - 0111-1111: Reserved or Presentation Time Control If MCGREN0 is set then these bits are treated as Presentation time control bits. The following list describes the values of PPSCMD0: - 0000: No Operation related to MCGR will be carried out. If set to this value in the mid of clock recovery or generation all the processing inputs will be flushed. - 0001: Capture the Presentation time at rising edge of mcg_pst_trig_i[0] into the MAC_PPS0_Target_Time_Seconds register - 0010: Capture the Presentation time at falling edge of mcg_pst_trig_i[0] into the MAC_PPS0_Target_Time_Seconds register - 0011: Capture the Presentation time at both edges of mcg_pst_trig_i[0] into the MAC_PPS0_Target_Time_Seconds register - 0100-1000: Reserved - 1001: Toggle output on compare - 1010: Pulse output low on compare for one PTP-clock cycle - 1011: Pulse output high on compare for one PTP-clock cycle

Field	Description
	- 1100-1111: Reserved

11.7.6.1.348 PPS0 Target Time Seconds (MAC_PPS0_TARGET_TIME_SECONDS)

The PPS0 Target Time Seconds register, along with PPS0 Target Time Nanoseconds register, is used to schedule an interrupt event [Bit 1 of MAC_TIMESTAMP_STATUS] when the system time exceeds the value programmed in these registers.

11.7.6.1.348.1 Offset

Register	Offset
MAC_PPS0_TARGET_TIME_SECONDS	B80h

11.7.6.1.348.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSTRH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSTRH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.348.3 Fields

Field	Description
31-0 TSTRH0	PPS Target Time Seconds Register This field stores the time in seconds. PPS Target Time Seconds Register This field stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on Target Time mode selected for the corresponding PPS output in the MAC_PPS_CONTROL register.

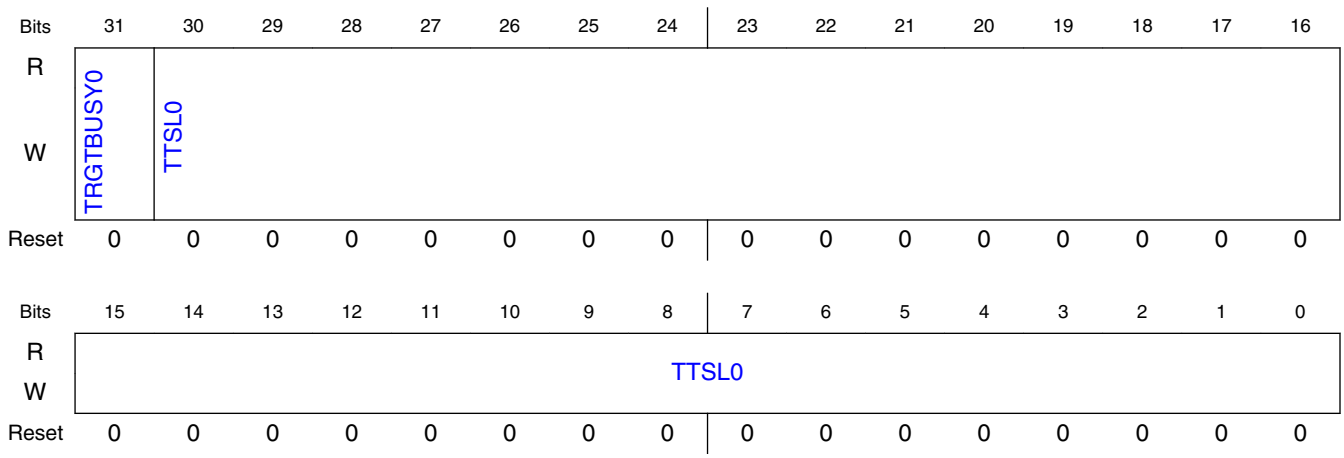
11.7.6.1.349 PPS0 Target Time Nanoseconds (MAC_PPS0_TARGET_TIME_NANOSECONDS)

PPS0 Target Time Nanoseconds register.

11.7.6.1.349.1 Offset

Register	Offset
MAC_PPS0_TARGET_TIME_NANOSECONDS	B84h

11.7.6.1.349.2 Diagram



11.7.6.1.349.3 Fields

Field	Description
31 TRGTBUSY0	PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the MAC_PPS_CONTROL register is programmed to 010 or 011. Programming the PPSCMD0 field to 010 or 011 instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted. 0 - PPS Target Time Register Busy status is not detected 1 - PPS Target Time Register Busy is detected
30-0 TTSL0	Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. Target Time Low for PPS Register

Field	Description
	This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the value in both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on the TRGTMODSEL0 field (Bits [6:5]) in PPS_CONTROL. When the TSCTRLSSR bit is reset in the MAC_TIMESTAMP_CONTROL register, this value should be (time in ns / 0.465). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. When the TSCTRLSSR bit is set in the MAC_TIMESTAMP_CONTROL register, this value should not exceed 0x3B9A_C9FF. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

11.7.6.1.350 PPS0 Interval (MAC_PPS0_INTERVAL)

The PPS0 Interval register contains the number of units of sub-second increment value between the rising edges of PPS0 signal output (ptp_pps_o[0]).

11.7.6.1.350.1 Offset

Register	Offset
MAC_PPS0_INTERVAL	B88h

11.7.6.1.350.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSINT0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSINT0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.350.3 Fields

Field	Description
31-0 PPSINT0	PPS Output Signal Interval These bits store the interval between the rising edges of PPS0 signal output. PPS Output Signal Interval

Field	Description
	These bits store the interval between the rising edges of PPS0 signal output. The interval is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between the rising edges of PPS0 signal output is 100 ns (that is, 5 units of sub-second increment value), you should program value 4 (5-1) in this register.

11.7.6.1.351 PPS0 Width (MAC_PPS0_WIDTH)

The PPS0 Width register contains the number of units of sub-second increment value between the rising and corresponding falling edges of PPS0 signal output (ptp_pps_o[0]).

11.7.6.1.351.1 Offset

Register	Offset
MAC_PPS0_WIDTH	B8Ch

11.7.6.1.351.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSWIDTH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSWIDTH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.351.3 Fields

Field	Description
31-0 PPSWIDTH0	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. The width is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and width between the rising and corresponding falling edges of PPS0 signal output is 80 ns (that is, four units of sub-second increment value), you should program value 3 (4-1) in this register.

Field	Description
	Note: The value programmed in this register must be lesser than the value programmed in MAC_PPS0_Interval.

11.7.6.1.352 PPS1 Target Time Seconds (MAC_PPS1_TARGET_TIME_SECONDS)

The PPS1 Target Time Seconds register, along with PPS1 Target Time Nanoseconds register, is used to schedule an interrupt event [Bit 1 of TIMESTAMP_STATUS] when the system time exceeds the value programmed in these registers.

11.7.6.1.352.1 Offset

Register	Offset
MAC_PPS1_TARGET_TIME_SECONDS	B90h

11.7.6.1.352.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSTRH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSTRH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.352.3 Fields

Field	Description
31-0 TSTRH1	PPS Target Time Seconds Register This field stores the time in seconds. PPS Target Time Seconds Register This field stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on Target Time mode selected for the corresponding PPS output in the PPS_CONTROL register.

Field	Description
	If DWC_EQOS_FLEXI_PPS_OUT_EN is enabled in the configuration and PTGE field of TIMESTAMP_CONTROL Register is set with Presentation time control set in recovery mode, then these bits indicate the TPT being programmed by the application and in generation mode it indicates the CPT generated at the sampled trigger.

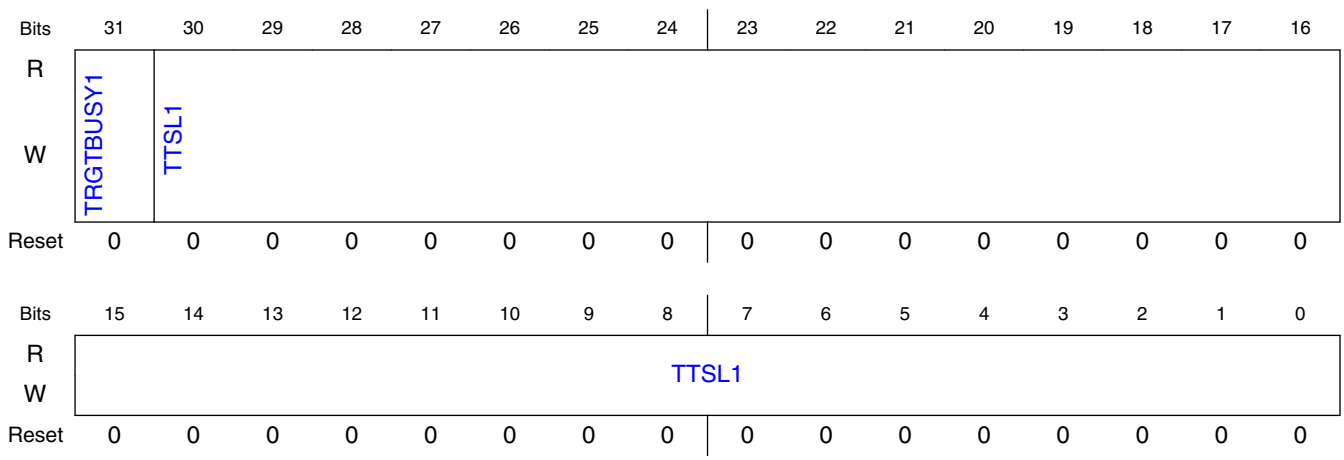
11.7.6.1.353 PPS1 Target Time Nanoseconds (MAC_PPS1_TARGET_TIME_NANOSECONDS)

PPS1 Target Time Nanoseconds register.

11.7.6.1.353.1 Offset

Register	Offset
MAC_PPS1_TARGET_TIME_NANOSECONDS	B94h

11.7.6.1.353.2 Diagram



11.7.6.1.353.3 Fields

Field	Description
31 TRGTBUSY1	PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. Programming the PPSCMD0 field to 010 or 011 instructs the MAC to synchronize the Target Time Registers to the PTP clock domain.

Table continues on the next page...

Field	Description
	The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted. 0 - PPS Target Time Register Busy status is not detected 1 - PPS Target Time Register Busy is detected
30-0 TTSL1	Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the value in both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on the TRGTMODSEL0 field (Bits [6:5]) in PPS_CONTROL. When the TSCTRLSSR bit is reset in the TIMESTAMP_CONTROL register, this value should be (time in ns / 0.465). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. When the TSCTRLSSR bit is set in the TIMESTAMP_CONTROL register, this value should not exceed 0x3B9A_C9FF. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

11.7.6.1.354 PPS1 Interval (MAC_PPS1_INTERVAL)

The PPS1 Interval register contains the number of units of sub-second increment value between the rising edges of PPS0 signal output (ptp_pps_o[0]).

11.7.6.1.354.1 Offset

Register	Offset
MAC_PPS1_INTERVAL	B98h

11.7.6.1.354.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSINT1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSINT1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.354.3 Fields

Field	Description
31-0 PPSINT1	PPS Output Signal Interval These bits store the interval between the rising edges of PPS0 signal output. PPS Output Signal Interval These bits store the interval between the rising edges of PPS0 signal output. The interval is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between the rising edges of PPS0 signal output is 100 ns (that is, 5 units of sub-second increment value), you should program value 4 (5-1) in this register.

11.7.6.1.355 PPS1 Width (MAC_PPS1_WIDTH)

The PPS1 Width register contains the number of units of sub-second increment value between the rising and corresponding falling edges of PPS0 signal output (ptp_pps_o[0]).

11.7.6.1.355.1 Offset

Register	Offset
MAC_PPS1_WIDTH	B9Ch

11.7.6.1.355.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSWIDTH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSWIDTH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.355.3 Fields

Field	Description
31-0 PPSWIDTH1	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. PPS Output Signal Width

Field	Description
	These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. The width is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and width between the rising and corresponding falling edges of PPS0 signal output is 80 ns (that is, four units of sub-second increment value), you should program value 3 (4-1) in this register. Note: The value programmed in this register must be lesser than the value programmed in MAC_PPS0_Interval.

11.7.6.1.356 PPS2 Target Time Seconds (MAC_PPS2_TARGET_TIME_SECONDS)

The PPS2 Target Time Seconds register, along with PPS2 Target Time Nanoseconds register, is used to schedule an interrupt event [Bit 1 of TIMESTAMP_STATUS] when the system time exceeds the value programmed in these registers.

11.7.6.1.356.1 Offset

Register	Offset
MAC_PPS2_TARGET_TIME_SECONDS	BA0h

11.7.6.1.356.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSTRH2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSTRH2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.356.3 Fields

Field	Description
31-0 TSTRH2	PPS Target Time Seconds Register This field stores the time in seconds. PPS Target Time Seconds Register

Field	Description
	This field stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on Target Time mode selected for the corresponding PPS output in the PPS_CONTROL register. If DWC_EQOS_FLEXI_PPS_OUT_EN is enabled in the configuration and PTGE field of TIMESTAMP_CONTROL Register is set with Presentation time control set in recovery mode, then these bits indicate the TPT being programmed by the application and in generation mode it indicates the CPT generated at the sampled trigger.

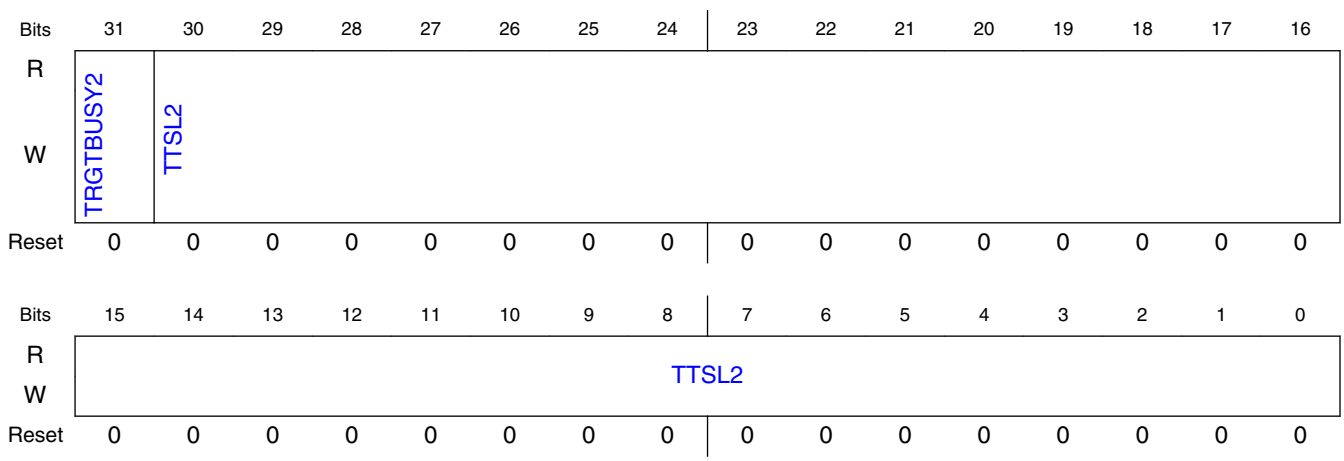
11.7.6.1.357 PPS2 Target Time Nanoseconds (MAC_PPS2_TARGET_TIME_NANOSECONDS)

PPS2 Target Time Nanoseconds register.

11.7.6.1.357.1 Offset

Register	Offset
MAC_PPS2_TARGET_TIME_NANOSECONDS	BA4h

11.7.6.1.357.2 Diagram



11.7.6.1.357.3 Fields

Field	Description
31 TRGTBUSY2	PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. PPS Target Time Register Busy

Table continues on the next page...

Field	Description
	The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. Programming the PPSCMD0 field to 010 or 011 instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted. 0 - PPS Target Time Register Busy status is not detected 1 - PPS Target Time Register Busy is detected
30-0 TTSL2	Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the value in both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on the TRGTMODSEL0 field (Bits [6:5]) in PPS_CONTROL. When the TSCTRLSSR bit is reset in the TIMESTAMP_CONTROL register, this value should be (time in ns / 0.465). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. When the TSCTRLSSR bit is set in the TIMESTAMP_CONTROL register, this value should not exceed 0x3B9A_C9FF. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

11.7.6.1.358 PPS2 Interval (MAC_PPS2_INTERVAL)

The PPS2 Interval register contains the number of units of sub-second increment value between the rising edges of PPS0 signal output (ptp_pps_o[0]).

11.7.6.1.358.1 Offset

Register	Offset
MAC_PPS2_INTERVAL	BA8h

11.7.6.1.358.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSINT2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSINT2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.358.3 Fields

Field	Description
31-0 PPSINT2	PPS Output Signal Interval These bits store the interval between the rising edges of PPS0 signal output. PPS Output Signal Interval These bits store the interval between the rising edges of PPS0 signal output. The interval is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between the rising edges of PPS0 signal output is 100 ns (that is, 5 units of sub-second increment value), you should program value 4 (5-1) in this register.

11.7.6.1.359 PPS2 Width (MAC_PPS2_WIDTH)

The PPS2 Width register contains the number of units of sub-second increment value between the rising and corresponding falling edges of PPS0 signal output (ptp_pps_o[0]).

11.7.6.1.359.1 Offset

Register	Offset
MAC_PPS2_WIDTH	BACh

11.7.6.1.359.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSWIDTH2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSWIDTH2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.359.3 Fields

Field	Description
31-0 PPSWIDTH2	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output.

Field	Description
	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. The width is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and width between the rising and corresponding falling edges of PPS0 signal output is 80 ns (that is, four units of sub-second increment value), you should program value 3 (4-1) in this register. Note: The value programmed in this register must be lesser than the value programmed in MAC_PPS0_Interval.

11.7.6.1.360 PPS3 Target Time Seconds (MAC_PPS3_TARGET_TIME_SECONDS)

The PPS3 Target Time Seconds register, along with PPS3 Target Time Nanoseconds register, is used to schedule an interrupt event [Bit 1 of TIMESTAMP_STATUS] when the system time exceeds the value programmed in these registers.

11.7.6.1.360.1 Offset

Register	Offset
MAC_PPS3_TARGET_TIME_SECONDS	BB0h

11.7.6.1.360.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSTRH3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSTRH3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.360.3 Fields

Field	Description
31-0	PPS Target Time Seconds Register This field stores the time in seconds.

Field	Description
TSTRH3	PPS Target Time Seconds Register This field stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on Target Time mode selected for the corresponding PPS output in the PPS_CONTROL register.

11.7.6.1.361 PPS3 Target Time Nanoseconds (MAC_PPS3_TARGET_TIME_NANOSECONDS)

PPS3 Target Time Nanoseconds register.

11.7.6.1.361.1 Offset

Register	Offset
MAC_PPS3_TARGET_TIME_NANOSECONDS	BB4h

11.7.6.1.361.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TRGTBUSY3	TTSL3														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TTSL3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.361.3 Fields

Field	Description
31 TRGTBUSY3	PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. PPS Target Time Register Busy

Table continues on the next page...

Field	Description
	The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. Programming the PPSCMD0 field to 010 or 011 instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted. 0 - PPS Target Time Register Busy status is not detected 1 - PPS Target Time Register Busy is detected
30-0 TTSL3	Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the value in both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on the TRGTMODSEL0 field (Bits [6:5]) in PPS_CONTROL. When the TSCTRLSSR bit is reset in the TIMESTAMP_CONTROL register, this value should be (time in ns / 0.465). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. When the TSCTRLSSR bit is set in the TIMESTAMP_CONTROL register, this value should not exceed 0x3B9A_C9FF. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

11.7.6.1.362 PPS3 Interval (MAC_PPS3_INTERVAL)

The PPS3 Interval register contains the number of units of sub-second increment value between the rising edges of PPS0 signal output (ptp_pps_o[0]).

11.7.6.1.362.1 Offset

Register	Offset
MAC_PPS3_INTERVAL	BB8h

11.7.6.1.362.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSINT3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSINT3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.362.3 Fields

Field	Description
31-0 PPSINT3	PPS Output Signal Interval These bits store the interval between the rising edges of PPS0 signal output. PPS Output Signal Interval These bits store the interval between the rising edges of PPS0 signal output. The interval is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between the rising edges of PPS0 signal output is 100 ns (that is, 5 units of sub-second increment value), you should program value 4 (5-1) in this register.

11.7.6.1.363 PPS3 Width (MAC_PPS3_WIDTH)

The PPS3 Width register contains the number of units of sub-second increment value between the rising and corresponding falling edges of PPS0 signal output (ptp_pps_o[0]).

11.7.6.1.363.1 Offset

Register	Offset
MAC_PPS3_WIDTH	BBCh

11.7.6.1.363.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSWIDTH3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSWIDTH3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.363.3 Fields

Field	Description
31-0 PPSWIDTH3	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output.

Field	Description
	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. The width is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and width between the rising and corresponding falling edges of PPS0 signal output is 80 ns (that is, four units of sub-second increment value), you should program value 3 (4-1) in this register. Note: The value programmed in this register must be lesser than the value programmed in MAC_PPS0_Interval.

11.7.6.1.364 PTP Offload Engine Control (MAC_PTO_CONTROL)

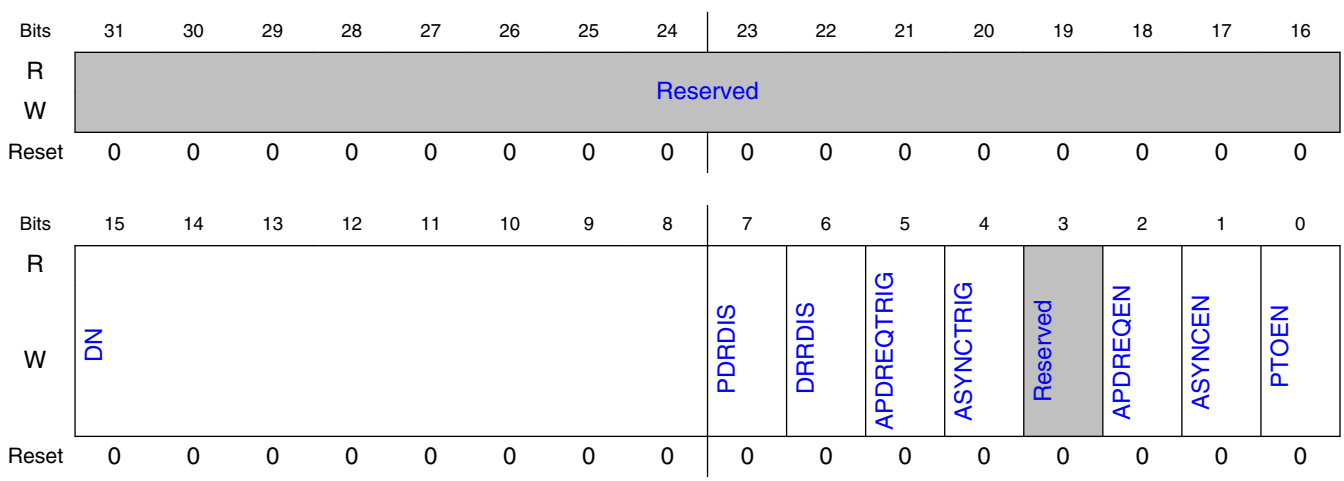
This register controls the PTP Offload Engine operation.

This register controls the PTP Offload Engine operation. This register is available only when the Enable PTP Timestamp Offload feature is selected.

11.7.6.1.364.1 Offset

Register	Offset
MAC_PTO_CONTROL	BC0h

11.7.6.1.364.2 Diagram



11.7.6.1.364.3 Fields

Field	Description
31-16 —	Reserved
15-8 DN	Domain Number This field indicates the domain Number in which the PTP node is operating. Domain Number This field indicates the domain Number in which the PTP node is operating.
7 PDRDIS	Disable Peer Delay Response response generation When this bit is set, the Peer Delay Response (Pdelay_Resp) response is not be generated for received Peer Delay Request (Pdelay_Req) request packet, as required by the programmed mode. Disable Peer Delay Response response generation When this bit is set, the Peer Delay Response (Pdelay_Resp) response is not be generated for received Peer Delay Request (Pdelay_Req) request packet, as required by the programmed mode. Note: Setting this bit to 1 affects the normal PTP Offload operation and the time synchronization. So, this bit must be set only if there is problem with Pdelay_Resp generation in Hardware and/or Pdelay_Resp generation is handled by Software. 0 - Peer Delay Response response generation is enabled 1 - Peer Delay Response response generation is disabled
6 DRRDIS	Disable PTO Delay Request/Response response generation When this bit is set, the Delay Request and Delay response is not generated for received SYNC and Delay request packet respectively, as required by the programmed mode. Disable PTO Delay Request/Response response generation When this bit is set, the Delay Request and Delay response is not generated for received SYNC and Delay request packet respectively, as required by the programmed mode. 0 - PTO Delay Request/Response response generation is enabled 1 - PTO Delay Request/Response response generation is disabled
5 APDREQTRIG	Automatic PTP Pdelay_Req message Trigger When this bit is set, one PTP Pdelay_Req message is transmitted. Automatic PTP Pdelay_Req message Trigger When this bit is set, one PTP Pdelay_Req message is transmitted. This bit is automatically cleared after the PTP Pdelay_Req message is transmitted. The application should set the APDREQEN bit for this operation. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Automatic PTP Pdelay_Req message Trigger is disabled 1 - Automatic PTP Pdelay_Req message Trigger is enabled
4 ASYNCTRIG	Automatic PTP SYNC message Trigger When this bit is set, one PTP SYNC message is transmitted. Automatic PTP SYNC message Trigger When this bit is set, one PTP SYNC message is transmitted. This bit is automatically cleared after the PTP SYNC message is transmitted. The application should set the ASYNCEN bit for this operation. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Automatic PTP SYNC message Trigger is disabled 1 - Automatic PTP SYNC message Trigger is enabled
3	Reserved

Table continues on the next page...

Field	Description
—	
2 APDREQEN	Automatic PTP Pdelay_Req message Enable When this bit is set, PTP Pdelay_Req message is generated periodically based on interval programmed or trigger from application, when the MAC is programmed to be in Peer-to-Peer Transparent mode. 0 - Automatic PTP Pdelay_Req message is disabled 1 - Automatic PTP Pdelay_Req message is enabled
1 ASYNCEN	Automatic PTP SYNC message Enable When this bit is set, PTP SYNC message is generated periodically based on interval programmed or trigger from application, when the MAC is programmed to be in Clock Master mode. 0 - Automatic PTP SYNC message is disabled 1 - Automatic PTP SYNC message is enabled
0 PTOEN	PTP Offload Enable When this bit is set, the PTP Offload feature is enabled. 0 - PTP Offload feature is disabled 1 - PTP Offload feature is enabled

11.7.6.1.365 Source Port Identity 0 (MAC_SOURCE_PORT_IDENTITY0)

This register contains Bits[31:0] of the 80-bit Source Port Identity of the PTP node. This register is available only when the Enable PTP Timestamp Offload feature is selected.

11.7.6.1.365.1 Offset

Register	Offset
MAC_SOURCE_PORT_IDENTITY0	BC4h

11.7.6.1.365.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SPI0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPI0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.365.3 Fields

Field	Description
31-0 SPI0	Source Port Identity 0 This field indicates bits [31:0] of sourcePortIdentity of PTP node.

11.7.6.1.366 Source Port Identity 1 (MAC_SOURCE_PORT_IDENTITY1)

This register contains Bits[63:32] of the 80-bit Source Port Identity of the PTP node. This register is available only when the Enable PTP Timestamp Offload feature is selected.

11.7.6.1.366.1 Offset

Register	Offset
MAC_SOURCE_PORT_IDENTITY1	BC8h

11.7.6.1.366.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SPI1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPI1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.366.3 Fields

Field	Description
31-0 SPI1	Source Port Identity 1 This field indicates bits [63:32] of sourcePortIdentity of PTP node.

11.7.6.1.367 Source Port Identity 2 (MAC_SOURCE_PORT_IDENTITY2)

This register contains Bits[79:64] of the 80-bit Source Port Identity of the PTP node. This register is available only when the Enable PTP Timestamp Offload feature is selected.

11.7.6.1.367.1 Offset

Register	Offset
MAC_SOURCE_PORT_IDENTITY2	BCCh

11.7.6.1.367.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPI2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.367.3 Fields

Field	Description
31-16 —	Reserved
15-0 SPI2	Source Port Identity 2 This field indicates bits [79:64] of sourcePortIdentity of PTP node.

11.7.6.1.368 Log Message Interval (MAC_LOG_MESSAGE_INTERVAL)

This register contains the periodic intervals for automatic PTP packet generation. This register is available only when the Enable PTP Timestamp Offload feature is selected.

11.7.6.1.368.1 Offset

Register	Offset
MAC_LOG_MESSAGE_INTERVAL	BD0h

11.7.6.1.368.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LMPDRI								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					DRSYNCR			LSI							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.368.3 Fields

Field	Description
31-24 LMPDRI	Log Min Pdelay_Req Interval This field indicates logMinPdelayReqInterval of PTP node. Log Min Pdelay_Req Interval This field indicates logMinPdelayReqInterval of PTP node. This is used to schedule the periodic Pdelay request packet transmission. Allowed values are -15 to 15.Negative value must be represented in 2's-complement form. For example, if the required value is -1, the value programmed must be 0xFF.
23-11 —	Reserved
10-8 DRSYNCR	Delay_Req to SYNC Ratio In Slave mode, it is used for controlling frequency of Delay_Req messages transmitted. Delay_Req to SYNC Ratio In Slave mode, it is used for controlling frequency of Delay_Req messages transmitted. - 0: DelayReq generated for every received SYNC - 1: DelayReq generated every alternate reception of SYNC - 2: for every 4 SYNC messages - 3: for every 8 SYNC messages - 4: for every 16 SYNC messages - 5: for every 32 SYNC messages - 6-7: Reserved

Table continues on the next page...

Field	Description
	The master sends this information (logMinDelayReqInterval) in the DelayResp PTP messages to the slave. The DWC_ether_qos Receiver processes this value from the received DelayResp messages and updates this field accordingly. In the Slave mode, the host must not write/update this register unless it has to override the received value. In Master mode, the sum of this field and logSyncInterval (LSI) field is provided in the logMinDelayReqInterval field of the generated multicast Delay_Resp PTP message. Access restriction applies. Updated based on the event. Setting 1 sets. Setting 0 clears. 000 - DelayReq generated for every received SYNC 001 - DelayReq generated every alternate reception of SYNC 010 - for every 4 SYNC messages 011 - for every 8 SYNC messages 100 - for every 16 SYNC messages 101 - for every 32 SYNC messages 110 - Reserved
7-0 LSI	Log Sync Interval This field indicates the periodicity of the automatically generated SYNC message when the PTP node is Master. Log Sync Interval This field indicates the periodicity of the automatically generated SYNC message when the PTP node is Master. Allowed values are -15 to 15. Negative value must be represented in 2's-complement form. For example, if the required value is -1, the value programmed must be 0xFF.

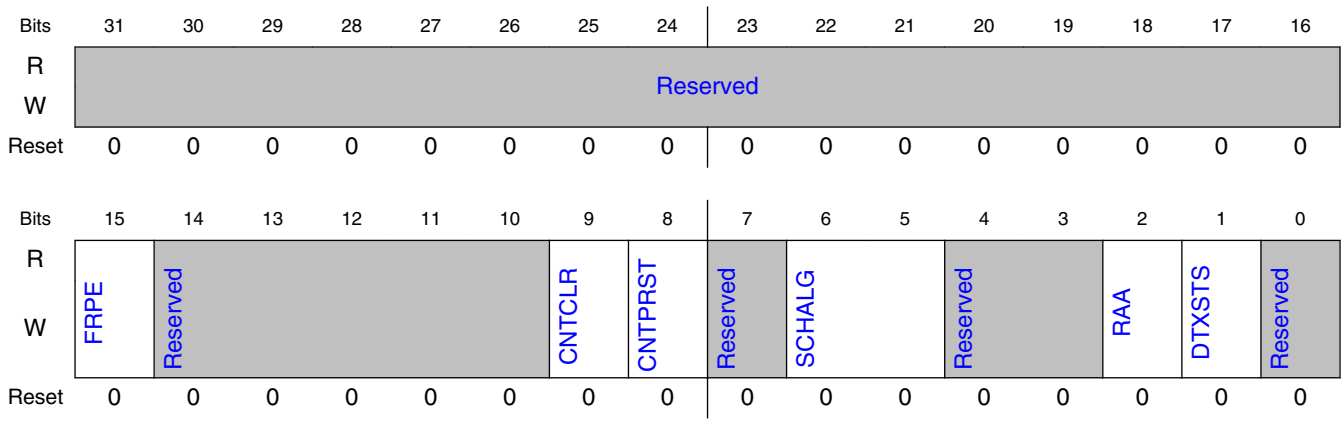
11.7.6.1.369 MTL Operation Mode (MTL_OPERATION_MODE)

The Operation Mode register establishes the Transmit and Receive operating modes and commands.

11.7.6.1.369.1 Offset

Register	Offset
MTL_OPERATION_MODE	C00h

11.7.6.1.369.2 Diagram



11.7.6.1.369.3 Fields

Field	Description
31-16 —	Reserved
15 FRPE	Flexible Rx parser Enable When this bit is set to 1, the Programmable Rx Parser functionality is enabled. Flexible Rx parser Enable When this bit is set to 1, the Programmable Rx Parser functionality is enabled. When the Rx parser is disabled and if the Rx parser is in the middle of the parsing then it gets disabled only after completing the current packet parsing. When the Rx parser is enabled from disabled state then the Rx parser gets activated for the next immediate packet. 0 - Flexible Rx parser is disabled 1 - Flexible Rx parser is enabled
14-10 —	Reserved
9 CNTCLR	Counters Reset When this bit is set, all counters are reset. Counters Reset When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle. If this bit is set along with CNT_PRESET bit, CNT_PRESET has precedence. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Counters are not reset 1 - All counters are reset
8 CNTPRST	Counters Preset When this bit is set, - MTL_TxQ[0-7]_Underflow register is initialized/preset to 12'h7F0. Counters Preset When this bit is set, - MTL_TxQ[0-7]_Underflow register is initialized/preset to 12'h7F0.

Table continues on the next page...

Field	Description
	- Missed Packet and Overflow Packet counters in MTL_RxQ[0-7]_Missed_Packet_Overflow_Cnt register is initialized/preset to 12'h7F0. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Counters Preset is disabled 1 - Counters Preset is enabled
7 —	Reserved
6-5 SCHALG	Tx Scheduling Algorithm This field indicates the algorithm for Tx scheduling: 00 - WRR algorithm 01 - WFQ algorithm when DCB feature is selected.Otherwise, Reserved 10 - DWRR algorithm when DCB feature is selected.Otherwise, Reserved 11 - Strict priority algorithm
4-3 —	Reserved
2 RAA	Receive Arbitration Algorithm This field is used to select the arbitration algorithm for the Rx side. Receive Arbitration Algorithm This field is used to select the arbitration algorithm for the Rx side. - 0: Strict priority (SP) Queue 0 has the lowest priority and the last queue has the highest priority. - 1: Weighted Strict Priority (WSP) 0 - Strict priority (SP) 1 - Weighted Strict Priority (WSP)
1 DTXSTS	Drop Transmit Status When this bit is set, the Tx packet status received from the MAC is dropped in the MTL. Drop Transmit Status When this bit is set, the Tx packet status received from the MAC is dropped in the MTL. When this bit is reset, the Tx packet status received from the MAC is forwarded to the application. 0 - Drop Transmit Status is disabled 1 - Drop Transmit Status is enabled
0 —	Reserved

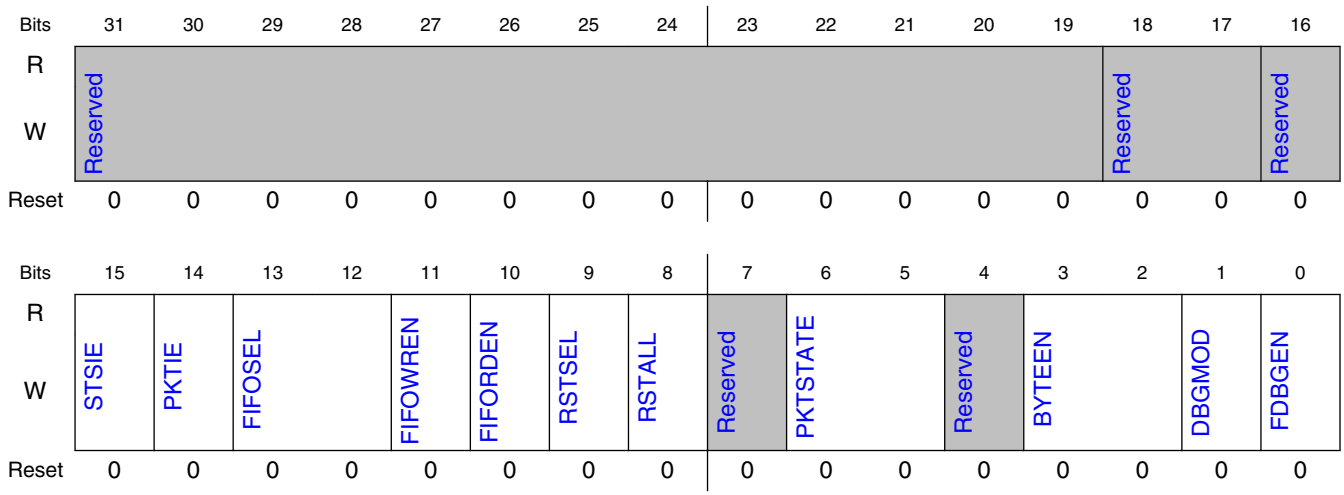
11.7.6.1.370 FIFO Debug Access Control and Status (MTL_DBG_CTL)

The FIFO Debug Access Control and Status register controls the operation mode of FIFO debug access.

11.7.6.1.370.1 Offset

Register	Offset
MTL_DBG_CTL	C08h

11.7.6.1.370.2 Diagram



11.7.6.1.370.3 Fields

Field	Description
31-19 —	Reserved
18-17 —	Reserved
16 —	Reserved
15 STSIE	Transmit Status Available Interrupt Status Enable When this bit is set, an interrupt is generated when Transmit status is available in slave mode. 0 - Transmit Packet Available Interrupt Status is disabled 1 - Transmit Packet Available Interrupt Status is enabled
14 PKTIE	Receive Packet Available Interrupt Status Enable When this bit is set, an interrupt is generated when EOP of received packet is written to the Rx FIFO. 0 - Receive Packet Available Interrupt Status is disabled 1 - Receive Packet Available Interrupt Status is enabled
13-12 FIFOSEL	FIFO Selected for Access This field indicates the FIFO selected for debug access: 00 - Tx FIFO 01 - Tx Status FIFO (only read access when SLVMOD is set)

Table continues on the next page...

Field	Description
	10 - TSO FIFO (cannot be accessed when SLVMOD is set) 11 - Rx FIFO
11 FIFOWREN	FIFO Write Enable When this bit is set, it enables the Write operation on selected FIFO when FIFO Debug Access is enabled. FIFO Write Enable When this bit is set, it enables the Write operation on selected FIFO when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0 - FIFO Write is disabled 1 - FIFO Write is enabled
10 FIFORDEN	FIFO Read Enable When this bit is set, it enables the Read operation on selected FIFO when FIFO Debug Access is enabled. FIFO Read Enable When this bit is set, it enables the Read operation on selected FIFO when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0 - FIFO Read is disabled 1 - FIFO Read is enabled
9 RSTSEL	Reset Pointers of Selected FIFO When this bit is set, the pointers of the currently-selected FIFO are reset when FIFO Debug Access is enabled. Reset Pointers of Selected FIFO When this bit is set, the pointers of the currently-selected FIFO are reset when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0 - Reset Pointers of Selected FIFO is disabled 1 - Reset Pointers of Selected FIFO is enabled
8 RSTALL	Reset All Pointers When this bit is set, the pointers of all FIFOs are reset when FIFO Debug Access is enabled. Reset All Pointers When this bit is set, the pointers of all FIFOs are reset when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0 - Reset All Pointers is disabled 1 - Reset All Pointers is enabled
7 —	Reserved
6-5 PKTSTATE	Encoded Packet State This field is used to write the control information to the Tx FIFO or Rx FIFO. Encoded Packet State

Table continues on the next page...

Field	Description
	This field is used to write the control information to the Tx FIFO or Rx FIFO. Tx FIFO: - 00: Packet Data - 01: Control Word - 10: SOP Data - 11: EOP Data Rx FIFO: - 00: Packet Data - 01: Normal Status - 10: Last Status - 11: EOP 00 - Packet Data 01 - Control Word/Normal Status 10 - SOP Data/Last Status 11 - EOP Data/EOP
4 —	Reserved
3-2 BYTEEN	Byte Enables This field indicates the number of data bytes valid in the data register during Write operation. Byte Enables This field indicates the number of data bytes valid in the data register during Write operation. This is valid only when PKTSTATE is 2'b10 (EOP) and Tx FIFO or Rx FIFO is selected. 00 - Byte 0 valid 01 - Byte 0 and Byte 1 are valid 10 - Byte 0, Byte 1, and Byte 2 are valid 11 - All four bytes are valid
1 DBGMOD	Debug Mode Access to FIFO When this bit is set, it indicates that the current access to the FIFO is read, write, and debug access. Debug Mode Access to FIFO When this bit is set, it indicates that the current access to the FIFO is read, write, and debug access. In this mode, the following access types are allowed: - Read and Write access to Tx FIFO, TSO FIFO, and Rx FIFO - Read access is allowed to Tx Status FIFO. When this bit is reset, it indicates that the current access to the FIFO is slave access bypassing the DMA. In this mode, the following access are allowed: - Write access to the Tx FIFO - Read access to the Rx FIFO and Tx Status FIFO 0 - Debug Mode Access to FIFO is disabled 1 - Debug Mode Access to FIFO is enabled
0	FIFO Debug Access Enable When this bit is set, it indicates that the debug mode access to the FIFO is enabled.

Field	Description
FDBGEN	FIFO Debug Access Enable When this bit is set, it indicates that the debug mode access to the FIFO is enabled. When this bit is reset, it indicates that the FIFO can be accessed only through a master interface. 0 - FIFO Debug Access is disabled 1 - FIFO Debug Access is enabled

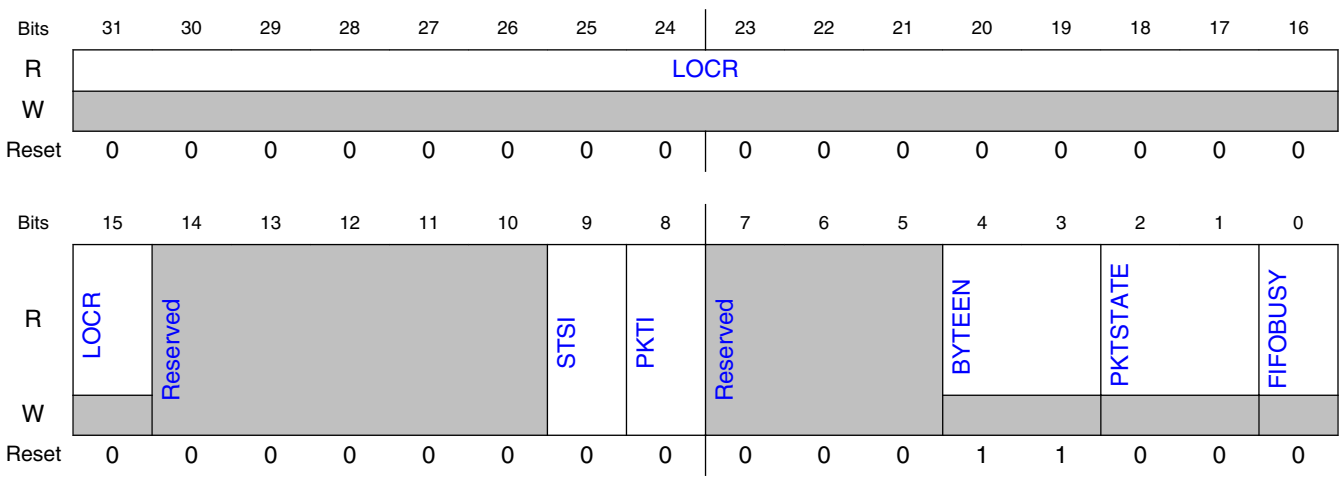
11.7.6.1.371 FIFO Debug Status (MTL_DBG_STS)

The FIFO Debug Status register contains the status of FIFO debug access.

11.7.6.1.371.1 Offset

Register	Offset
MTL_DBG_STS	C0Ch

11.7.6.1.371.2 Diagram



11.7.6.1.371.3 Fields

Field	Description
31-15 LOCR	Remaining Locations in the FIFO Slave Access Mode: This field indicates the space available in selected FIFO. Remaining Locations in the FIFO Slave Access Mode: This field indicates the space available in selected FIFO.

Table continues on the next page...

Field	Description
	Debug Access Mode: This field contains the Write or Read pointer value of the selected FIFO during Write or Read operation, respectively. Reset: In single Tx Queue configurations, (DWC_EQOS_TXFIFO_SIZE/(DWC_EQOS_DATAWIDTH/8)), Otherwise 0000H
14-10 —	Reserved
9 STSI	Transmit Status Available Interrupt Status When set, this bit indicates that the Slave mode Tx packet is transmitted, and the status is available in Tx Status FIFO. Transmit Status Available Interrupt Status When set, this bit indicates that the Slave mode Tx packet is transmitted, and the status is available in Tx Status FIFO. This bit is reset when 1 is written to this bit. 0 - Transmit Status Available Interrupt Status not detected 1 - Transmit Status Available Interrupt Status detected
8 PKTI	Receive Packet Available Interrupt Status When set, this bit indicates that MAC layer has written the EOP of received packet to the Rx FIFO. Receive Packet Available Interrupt Status When set, this bit indicates that MAC layer has written the EOP of received packet to the Rx FIFO. This bit is reset when 1 is written to this bit. 0 - Receive Packet Available Interrupt Status not detected 1 - Receive Packet Available Interrupt Status detected
7-5 —	Reserved
4-3 BYTEEN	Byte Enables This field indicates the number of data bytes valid in the data register during Read operation. Byte Enables This field indicates the number of data bytes valid in the data register during Read operation. This is valid only when PKTSTATE is 2'b10 (EOP) and Tx FIFO or Rx FIFO is selected. 00 - Byte 0 valid 01 - Byte 0 and Byte 1 are valid 10 - Byte 0, Byte 1, and Byte 2 are valid 11 - All four bytes are valid
2-1 PKTSTATE	Encoded Packet State This field is used to get the control or status information of the selected FIFO. Encoded Packet State This field is used to get the control or status information of the selected FIFO. Tx FIFO: - 00: Packet Data - 01: Control Word - 10: SOP Data - 11: EOP Data Rx FIFO:

Table continues on the next page...

Field	Description
	- 00: Packet Data - 01: Normal Status - 10: Last Status - 11: EOP This field is applicable only for Tx FIFO and Rx FIFO during Read operation. 00 - Packet Data 01 - Control Word/Normal Status 10 - SOP Data/Last Status 11 - EOP Data/EOP
0 FIFOBUSY	FIFO Busy When set, this bit indicates that a FIFO operation is in progress in the MAC and content of the following fields is not valid: - All other fields of this register - All fields of the MTL_FIFO_DEBUG_DATA register 0 - FIFO Busy not detected 1 - FIFO Busy detected

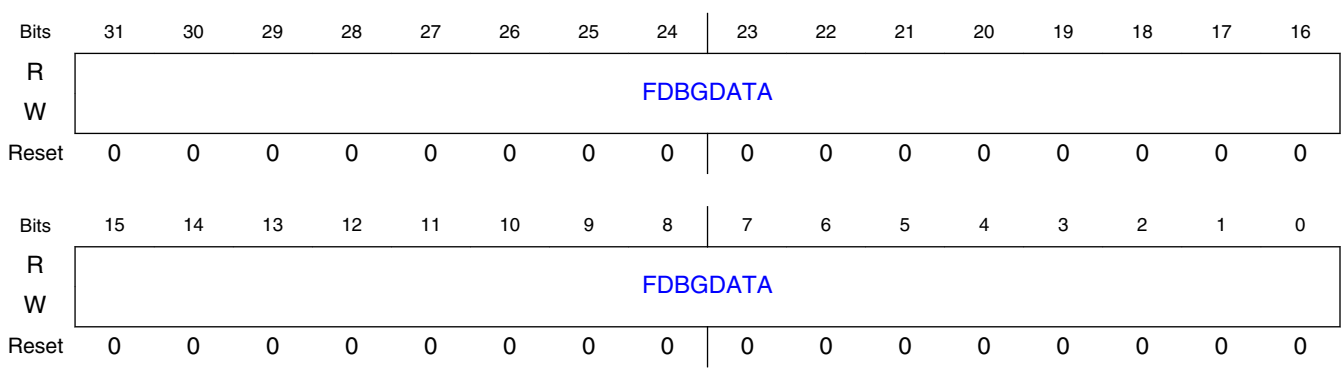
11.7.6.1.372 FIFO Debug Data (MTL_FIFO_DEBUG_DATA)

The FIFO Debug Data register contains the data to be written to or read from the FIFOs.

11.7.6.1.372.1 Offset

Register	Offset
MTL_FIFO_DEBUG_DATA	C10h

11.7.6.1.372.2 Diagram



11.7.6.1.372.3 Fields

Field	Description
31-0 FDBGDATA	FIFO Debug Data During debug or slave access write operation, this field contains the data to be written to the Tx FIFO, Rx FIFO, or TSO FIFO. FIFO Debug Data During debug or slave access write operation, this field contains the data to be written to the Tx FIFO, Rx FIFO, or TSO FIFO. During debug or slave access read operation, this field contains the data read from the Tx FIFO, Rx FIFO, TSO FIFO, or Tx Status FIFO.

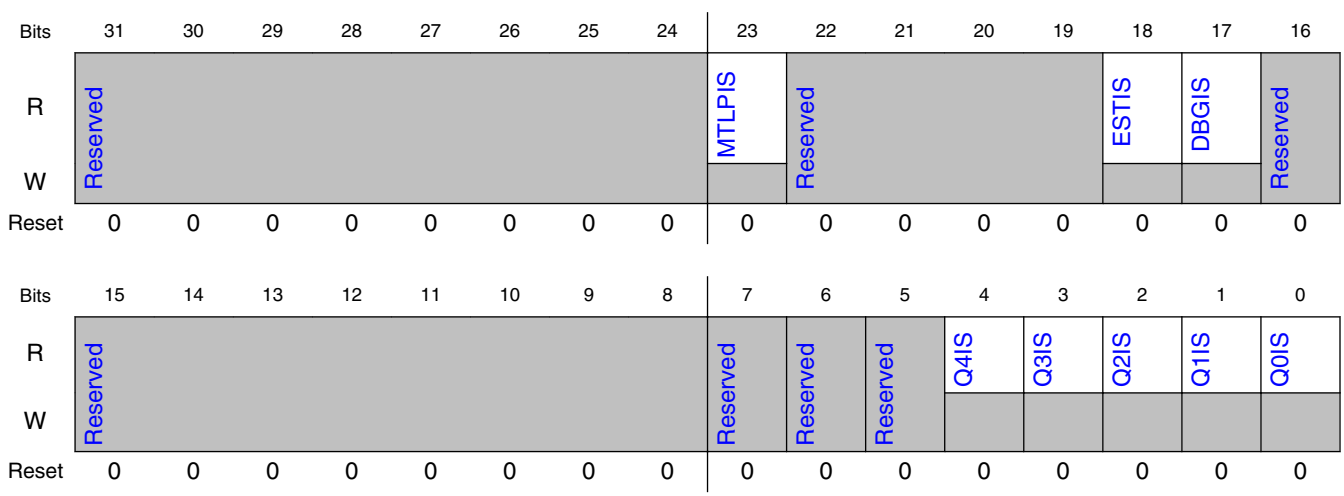
11.7.6.1.373 MTL Interrupt Status (MTL_INTERRUPT_STATUS)

The software driver (application) reads this register during interrupt service routine or polling to determine the interrupt status of MTL queues and the MAC.

11.7.6.1.373.1 Offset

Register	Offset
MTL_INTERRUPT_STATUS	C20h

11.7.6.1.373.2 Diagram



11.7.6.1.373.3 Fields

Field	Description
31-24	Reserved

Table continues on the next page...

Field	Description
—	
23 MTLPIS	MTL Rx Parser Interrupt Status This bit indicates that there is an interrupt from Rx Parser Block. MTL Rx Parser Interrupt Status This bit indicates that there is an interrupt from Rx Parser Block. To reset this bit, the application must read the MTL_RXP_INTERRUPT_STATUS register to get the exact cause of the interrupt and clear its source. 0 - MTL Rx Parser Interrupt status not detected 1 - MTL Rx Parser Interrupt status detected
22-19 —	Reserved
18 ESTIS	EST (TAS- 802. EST (TAS- 802.1Qbv) Interrupt Status This bit indicates an interrupt event during the operation of 802.1Qbv. To reset this bit, the application must clear the error/event that has caused the Interrupt. 0 - EST (TAS- 802.1Qbv) Interrupt status not detected 1 - EST (TAS- 802.1Qbv) Interrupt status detected
17 DBGIS	Debug Interrupt status This bit indicates an interrupt event during the slave access. Debug Interrupt status This bit indicates an interrupt event during the slave access. To reset this bit, the application must read the FIFO Debug Access Status register to get the exact cause of the interrupt and clear its source. 0 - Debug Interrupt status not detected 1 - Debug Interrupt status detected
16 —	Reserved
15-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 Q4IS	Queue 4 Interrupt status This bit indicates that there is an interrupt from Queue 4. Queue 4 Interrupt status This bit indicates that there is an interrupt from Queue 4. To reset this bit, the application must read the MTL_Q4_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0 - Queue 4 Interrupt status not detected 1 - Queue 4 Interrupt status detected
3 Q3IS	Queue 3 Interrupt status This bit indicates that there is an interrupt from Queue 3. Queue 3 Interrupt status

Table continues on the next page...

Field	Description
	This bit indicates that there is an interrupt from Queue 3. To reset this bit, the application must read the MTL_Q3_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0 - Queue 3 Interrupt status not detected 1 - Queue 3 Interrupt status detected
2 Q2IS	Queue 2 Interrupt status This bit indicates that there is an interrupt from Queue 2. Queue 2 Interrupt status This bit indicates that there is an interrupt from Queue 2. To reset this bit, the application must read the MTL_Q2_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0 - Queue 2 Interrupt status not detected 1 - Queue 2 Interrupt status detected
1 Q1IS	Queue 1 Interrupt status This bit indicates that there is an interrupt from Queue 1. Queue 1 Interrupt status This bit indicates that there is an interrupt from Queue 1. To reset this bit, the application must read the MTL_Q1_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0 - Queue 1 Interrupt status not detected 1 - Queue 1 Interrupt status detected
0 Q0IS	Queue 0 Interrupt status This bit indicates that there is an interrupt from Queue 0. Queue 0 Interrupt status This bit indicates that there is an interrupt from Queue 0. To reset this bit, the application must read Queue 0 Interrupt Control and Status register to get the exact cause of the interrupt and clear its source. 0 - Queue 0 Interrupt status not detected 1 - Queue 0 Interrupt status detected

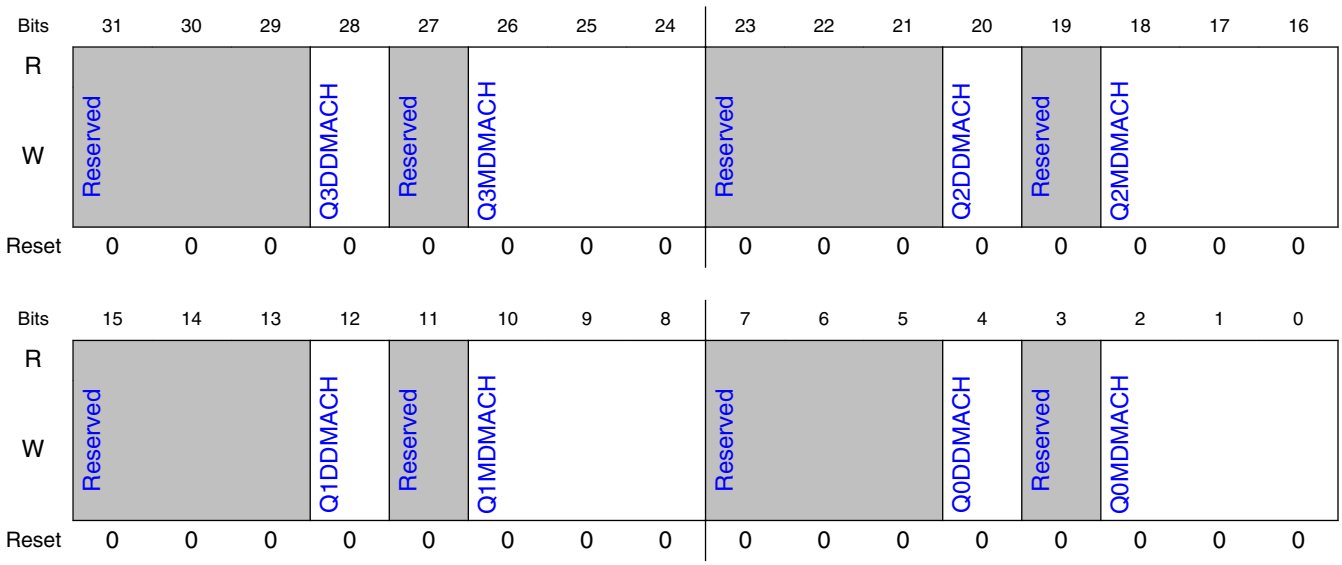
11.7.6.1.374 Receive Queue and DMA Channel Mapping 0 (MTL_RXQ_DMA_MAP0)

The Receive Queue and DMA Channel Mapping 0 register is reserved in EQOS-CORE and EQOS-MTL configurations.

11.7.6.1.374.1 Offset

Register	Offset
MTL_RXQ_DMA_MAP0	C30h

11.7.6.1.374.2 Diagram



11.7.6.1.374.3 Fields

Field	Description
31-29 —	Reserved
28 Q3DDMACH	Queue 3 Enabled for Dynamic (per packet) DMA Channel Selection When set, this bit indicates that the packets received in Queue 3 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 3 Enabled for Dynamic (per packet) DMA Channel Selection When set, this bit indicates that the packets received in Queue 3 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 3 are routed to the DMA Channel programmed in the Q3MDMACH field (Bits[26:24]). 0 - Queue 3 disabled for DA-based DMA Channel Selection 1 - Queue 3 enabled for DA-based DMA Channel Selection
27 —	Reserved
26-24 Q3MDMACH	Queue 3 Mapped to DMA Channel This field controls the routing of the received packet in Queue 3 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q3DDMACH field is reset. Queue 3 Mapped to DMA Channel This field controls the routing of the received packet in Queue 3 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1

Table continues on the next page...

Field	Description
	- 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q3DDMACH field is reset. Note: The width of this field depends on the number of RX DMA channels and not all the values may be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the others are reserved
23-21 —	Reserved
20 Q2DDMACH	Queue 2 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 2 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 2 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 2 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 2 are routed to the DMA Channel programmed in the Q2MDMACH field (Bits[18:16]). 0 - Queue 2 disabled for DA-based DMA Channel Selection 1 - Queue 2 enabled for DA-based DMA Channel Selection
19 —	Reserved
18-16 Q2MDMACH	Queue 2 Mapped to DMA Channel This field controls the routing of the received packet in Queue 2 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q2DDMACH field is reset.
15-13 —	Reserved
12 Q1DDMACH	Queue 1 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 1 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 1 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 1 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 1 are routed to the DMA Channel programmed in the Q1MDMACH field (Bits[10:8]). 0 - Queue 1 disabled for DA-based DMA Channel Selection 1 - Queue 1 enabled for DA-based DMA Channel Selection
11 —	Reserved

Table continues on the next page...

Field	Description
10-8 Q1MDMACH	Queue 1 Mapped to DMA Channel This field controls the routing of the received packet in Queue 1 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q1DDMACH field is reset. Queue 1 Mapped to DMA Channel This field controls the routing of the received packet in Queue 1 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q1DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values may be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the other bits are reserved.
7-5 —	Reserved
4 Q0DDMACH	Queue 0 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 0 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 0 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 0 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 0 are routed to the DMA Channel programmed in the Q0MDMACH field. 0 - Queue 0 disabled for DA-based DMA Channel Selection 1 - Queue 0 enabled for DA-based DMA Channel Selection
3 —	Reserved
2-0 Q0MDMACH	Queue 0 Mapped to DMA Channel This field controls the routing of the packet received in Queue 0 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q0DDMACH field is reset. Queue 0 Mapped to DMA Channel This field controls the routing of the packet received in Queue 0 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3

Field	Description
	- 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q0DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values may be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the other bits are reserved.

11.7.6.1.375 Receive Queue and DMA Channel Mapping 1 (MTL_RXQ_DMA_MAP1)

The Receive Queue and DMA Channel Mapping 1 register is reserved in EQOS-CORE and EQOS-MTL configurations.

11.7.6.1.375.1 Offset

Register	Offset
MTL_RXQ_DMA_MAP1	C34h

11.7.6.1.375.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			Reserved	Reserved	Reserved			Reserved			Reserved	Reserved	Reserved		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			Reserved	Reserved	Reserved			Reserved			Q4DDMACH	Reserved	Q4MDMACH		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.375.3 Fields

Field	Description
31-29 —	Reserved
28 —	Reserved
27 —	Reserved
26-24 —	Reserved
23-21 —	Reserved
20 —	Reserved
19 —	Reserved
18-16 —	Reserved
15-13 —	Reserved
12 —	Reserved
11 —	Reserved
10-8 —	Reserved
7-5 —	Reserved
4 Q4DDMACH	Queue 4 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 4 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 4 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 4 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 4 are routed to the DMA Channel programmed in the Q4MDMACH field. 0 - Queue 4 disabled for DA-based DMA Channel Selection 1 - Queue 4 enabled for DA-based DMA Channel Selection
3 —	Reserved

Table continues on the next page...

Field	Description
2-0 Q4MDMACH	Queue 4 Mapped to DMA Channel This field controls the routing of the packet received in Queue 4 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q4DDMACH field is reset. Queue 4 Mapped to DMA Channel This field controls the routing of the packet received in Queue 4 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q4DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values may be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the other bits are reserved.

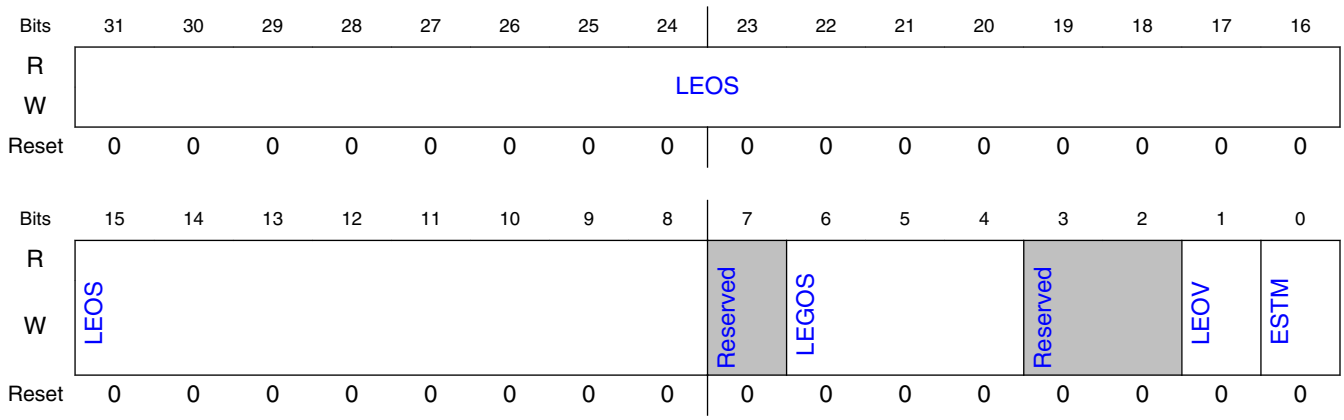
11.7.6.1.376 Time Based Scheduling Control (MTL_TBS_CTRL)

This register controls the operation of Time Based Scheduling.

11.7.6.1.376.1 Offset

Register	Offset
MTL_TBS_CTRL	C40h

11.7.6.1.376.2 Diagram



11.7.6.1.376.3 Fields

Field	Description
31-8 LEOS	Launch Expiry Offset The value in units of 256 nanoseconds that has to be added to the Launch time to compute the Launch Expiry time. Launch Expiry Offset The value in units of 256 nanoseconds that has to be added to the Launch time to compute the Launch Expiry time. Value valid only when LEOV is set. Max value: 999,999,999 ns, additionally should be smaller than CTR-1 value when ESTM mode is set since this value is a modulo CTR value.
7 —	Reserved
6-4 LEGOS	Launch Expiry GSN Offset The number GSN slots that has to be added to the Launch GSN to compute the Launch Expiry time. Launch Expiry GSN Offset The number GSN slots that has to be added to the Launch GSN to compute the Launch Expiry time. Value valid only when LEOV is set.
3-2 —	Reserved
1 LEOV	Launch Expiry Offset Valid When set indicates the LEOS field is valid. Launch Expiry Offset Valid When set indicates the LEOS field is valid. When not set, indicates the Launch Expiry Offset is not valid and the MTL must not check for Launch expiry time. 0 - LEOS field is invalid 1 - LEOS field is valid
0 ESTM	EST offset Mode When this bit is set, the Launch Time value used in Time Based Scheduling is interpreted as an EST offset value and is added to the Base Time Register (BTR) of the current list. EST offset Mode

Field	Description
	When this bit is set, the Launch Time value used in Time Based Scheduling is interpreted as an EST offset value and is added to the Base Time Register (BTR) of the current list. When reset, the Launch Time value is used as an absolute value that should be compared with the System time [39:8]. 0 - EST offset Mode is disabled 1 - EST offset Mode is enabled

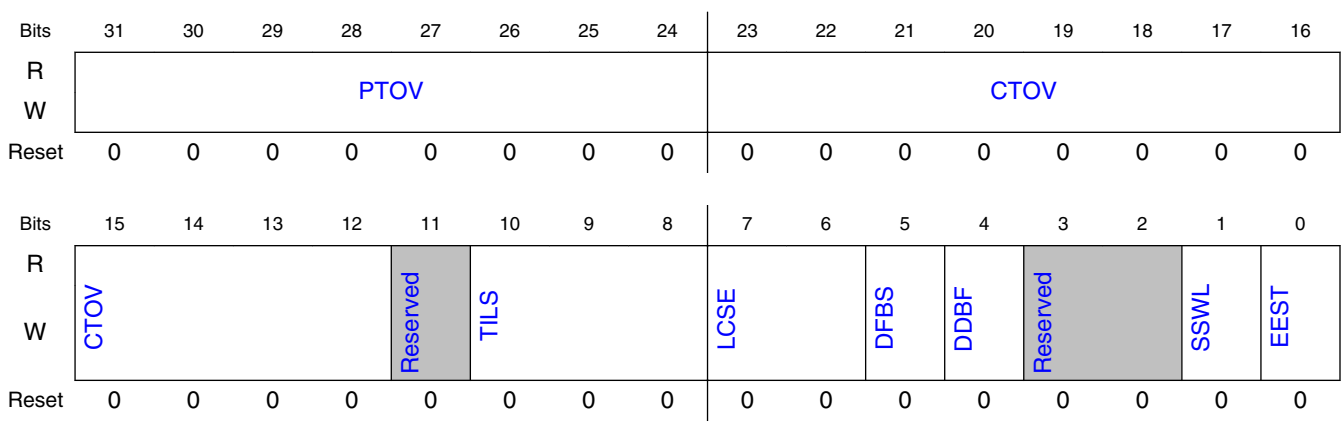
11.7.6.1.377 Enhancements to Scheduled Transmission Control (MTL_EST_CONTROL)

This register controls the operation of Enhancements to Scheduled Transmission (IEEE802.1Qbv).

11.7.6.1.377.1 Offset

Register	Offset
MTL_EST_CONTROL	C50h

11.7.6.1.377.2 Diagram



11.7.6.1.377.3 Fields

Field	Description
31-24	PTP Time Offset Value The value of PTP Clock period multiplied by 6 in nanoseconds.
PTOV	PTP Time Offset Value

Table continues on the next page...

Field	Description
	The value of PTP Clock period multiplied by 6 in nanoseconds. This value is needed to avoid transmission overruns at the beginning of the installation of a new GCL.
23-12 CTOV	Current Time Offset Value Provides a 12 bit time offset value in nano second that is added to the current time to compensate for all the implementation pipeline delays such as the CDC sync delay, buffering delays, data path delays etc. Current Time Offset Value Provides a 12 bit time offset value in nano second that is added to the current time to compensate for all the implementation pipeline delays such as the CDC sync delay, buffering delays, data path delays etc. This offset helps to ensure that the impact of gate controls is visible on the line exactly at the pre-determined schedule (or as close to the schedule as possible).
11 —	Reserved
10-8 TILS	Time Interval Left Shift Amount This field provides the left shift amount for the programmed Time Interval values used in the Gate Control Lists. Time Interval Left Shift Amount This field provides the left shift amount for the programmed Time Interval values used in the Gate Control Lists. - 000: No left shift needed (equal to x1ns) - 001: Left shift TI by 1 bit (equal to x2ns) - 010: Left shift TI by 2 bits (equal to x4ns) - . - . - 100: Left shift TI by 7 bits (equal to x128ns) Based on the configuration one or more bits of this field should be treated as Reserved/Read-Only.
7-6 LCSE	Loop Count to report Scheduling Error Programmable number of GCL list iterations before reporting an HLBS error defined in EST_STATUS register. 00 - 4 iterations 01 - 8 iterations 10 - 16 iterations 11 - 32 iterations
5 DFBS	Drop Frames causing Scheduling Error When set frames reported to cause HOL Blocking due to not getting scheduled (HLBS field of EST_STATUS register) after 4,8,16,32 (based on LCSE field of this register) GCL iterations are dropped. 0 - Do not Drop Frames causing Scheduling Error 1 - Drop Frames causing Scheduling Error
4 DDBF	Do not Drop frames during Frame Size Error When set, frames are not be dropped during Head-of-Line blocking due to Frame Size Error (HLBF field of MTL_EST_STATUS register). 0 - Drop frames during Frame Size Error 1 - Do not Drop frames during Frame Size Error
3-2 —	Reserved
1 SSWL	Switch to S/W owned list When set indicates that the software has programmed that list that it currently owns (SWOL) and the hardware should switch to the new list based on the new BTR.

Table continues on the next page...

Field	Description
	Switch to S/W owned list When set indicates that the software has programmed that list that it currently owns (SWOL) and the hardware should switch to the new list based on the new BTR. Hardware clears this bit when the switch to the SWOL happens to indicate the completion of the switch or when an BTR error (BTRE in Status register) is set. When BTRE is set this bit is cleared but SWOL is not updated as the switch was not successful. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Switch to S/W owned list is disabled 1 - Switch to S/W owned list is enabled
0 EEST	Enable EST When reset, the gate control list processing is halted and all gates are assumed to be in Open state. Enable EST When reset, the gate control list processing is halted and all gates are assumed to be in Open state. Should be set for the hardware to start processing the gate control lists. During the toggle from 0 to 1, the gate control list processing starts only after the SSWL bit it set. When DWC_EQOS_ASP_ECC is selected during the configuration, if any uncorrectable error is detected in the EST memory the hardware will reset this bit and disable the EST function. 0 - EST is disabled 1 - EST is enabled

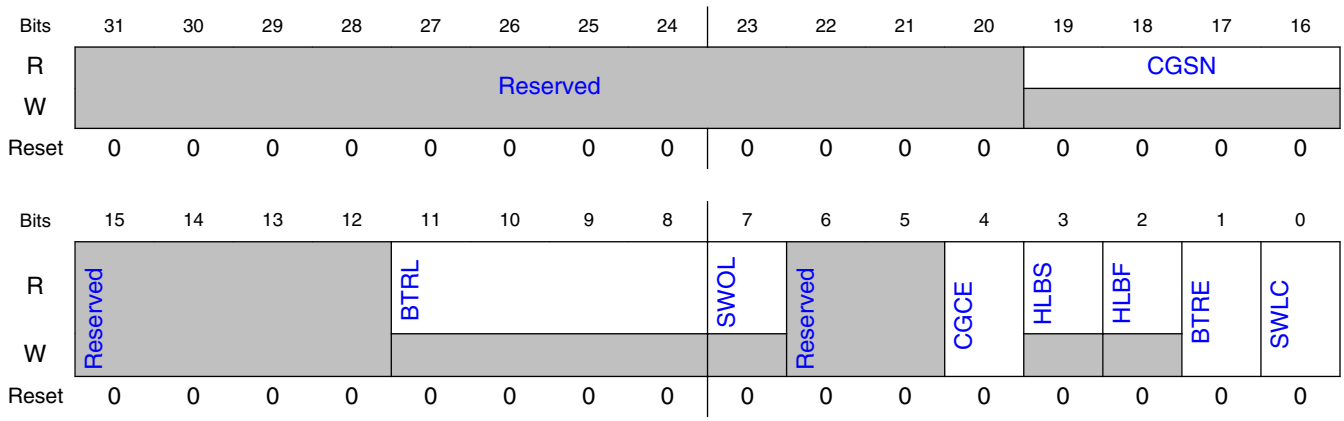
11.7.6.1.378 Enhancements to Scheduled Transmission Status (MTL_EST_STATUS)

This register provides Status related to Enhancements to Scheduled Transmission (IEEE802.1Qbv).

11.7.6.1.378.1 Offset

Register	Offset
MTL_EST_STATUS	C58h

11.7.6.1.378.2 Diagram



11.7.6.1.378.3 Fields

Field	Description
31-20 —	Reserved
19-16 CGSN	Current GCL Slot Number Indicates the slot number of the GCL list. Current GCL Slot Number Indicates the slot number of the GCL list. Slot number is a modulo 16 count of the GCL List loops executed so far. Even if a new GCL list is installed, the count is incremental.
15-12 —	Reserved
11-8 BTRL	BTR Error Loop Count Provides the minimum count (N) for which the equation Current Time =< New BTR + (N * New Cycle Time) becomes true. BTR Error Loop Count Provides the minimum count (N) for which the equation Current Time =< New BTR + (N * New Cycle Time) becomes true. N = "1111" indicates the iterations exceeded the value of 8 and the hardware was not able to update New BTR to be equal to or greater than Current Time. Software intervention is needed to update the New BTR. Value cleared when BTRE field of this register is cleared.
7 SWOL	S/W owned list When '0' indicates Gate control list number "0" is owned by software and when "1" indicates the Gate Control list "1" is owned by the software. S/W owned list When '0' indicates Gate control list number "0" is owned by software and when "1" indicates the Gate Control list "1" is owned by the software. Any reads/writes by the software (using indirect access via GCL_Control) is directed to the list indicated by this value by default. The inverse of this value is treated as HWOL. R/W operations performed by hardware are directed to the list pointed by HWOL by default. 0 - Gate control list number "0" is owned by software 1 - Gate control list number "1" is owned by software
6-5	Reserved

Table continues on the next page...

Field	Description
—	
4 CGCE	Constant Gate Control Error This error occurs when the list length (LLR) is 1 and the programmed Time Interval (TI) value after the optional Left Shifting is less than or equal to the Cycle Time (CTR). Constant Gate Control Error This error occurs when the list length (LLR) is 1 and the programmed Time Interval (TI) value after the optional Left Shifting is less than or equal to the Cycle Time (CTR). The above programming implies Gates are either always Closed or always Open based on the Gate Control values; the same effect can be achieved by other simpler (non TSN) programming mechanisms. Since the implementation does not support such a programming an error is reported. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Constant Gate Control Error not detected 1 - Constant Gate Control Error detected
3 HLBS	Head-Of-Line Blocking due to Scheduling Set when the frame is not able to win arbitration and get scheduled even after 4 iterations of the GCL. Head-Of-Line Blocking due to Scheduling Set when the frame is not able to win arbitration and get scheduled even after 4 iterations of the GCL. Indicates to software a potential programming error. The one hot encoded values of the Queue Numbers that are not able to make progress are indicated in the MTL_EST_SCH_ERROR register. Bit cleared when MTL_EST_SCH_ERROR register is all zeroes. 0 - Head-Of-Line Blocking due to Scheduling not detected 1 - Head-Of-Line Blocking due to Scheduling detected
2 HLBF	Head-Of-Line Blocking due to Frame Size Set when HOL Blocking is noticed on one or more Queues as a result of none of the Time Intervals of gate open in the GCL being greater than or equal to the duration needed for frame size (or frame fragment size when preemption is enabled) transmission. Head-Of-Line Blocking due to Frame Size Set when HOL Blocking is noticed on one or more Queues as a result of none of the Time Intervals of gate open in the GCL being greater than or equal to the duration needed for frame size (or frame fragment size when preemption is enabled) transmission. The one hot encoded Queue numbers that are experiencing HLBF are indicated in the MTL_EST_FRM_SIZE_ERROR register. Additionally, the first Queue number that experienced HLBF along with the frame size is captured in MTL_EST_FRM_SIZE_CAPTURE register. Bit cleared when MTL_EST_FRAME_SIZE_ ERROR register is all zeros. 0 - Head-Of-Line Blocking due to Frame Size not detected 1 - Head-Of-Line Blocking due to Frame Size detected
1 BTRE	BTR Error When "1" indicates a programming error in the BTR of SWOL where the programmed value is less than current time. BTR Error When "1" indicates a programming error in the BTR of SWOL where the programmed value is less than current time. If the BTRL = "1111", SWOL is not updated and Software should reprogram the BTR to a value greater than current time and then set SSWL to reinitiate the switch to SWOL. Else if the value of BTRL < "1111", SWOL is updated and this field indicates the number of iterations (of + CycleTime) taken by hardware to update the BTR to a value greater than Current Time. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - BTR Error not detected 1 - BTR Error detected

Table continues on the next page...

Field	Description
0 SWLC	Switch to S/W owned list Complete When "1" indicates the hardware has successfully switched to the SWOL, and the SWOL bit has been updated to that effect. Switch to S/W owned list Complete When "1" indicates the hardware has successfully switched to the SWOL, and the SWOL bit has been updated to that effect. Cleared when the SSWL of MTL_EST_CONTROL register transitions from 0 to 1, or on a software write. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Switch to S/W owned list Complete not detected 1 - Switch to S/W owned list Complete detected

11.7.6.1.379 EST Scheduling Error (MTL_EST_SCH_ERROR)

This register provides the One Hot encoded Queue Numbers that are having the Scheduling related error (timeout).

11.7.6.1.379.1 Offset

Register	Offset
MTL_EST_SCH_ERROR	C60h

11.7.6.1.379.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SEQN				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.379.3 Fields

Field	Description
31-5 —	Reserved

Table continues on the next page...

Field	Description
4-0 SEQN	Schedule Error Queue Number The One Hot Encoded Queue Numbers that have experienced error/ timeout described in HLBS field of status register. Schedule Error Queue Number The One Hot Encoded Queue Numbers that have experienced error/timeout described in HLBS field of status register. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

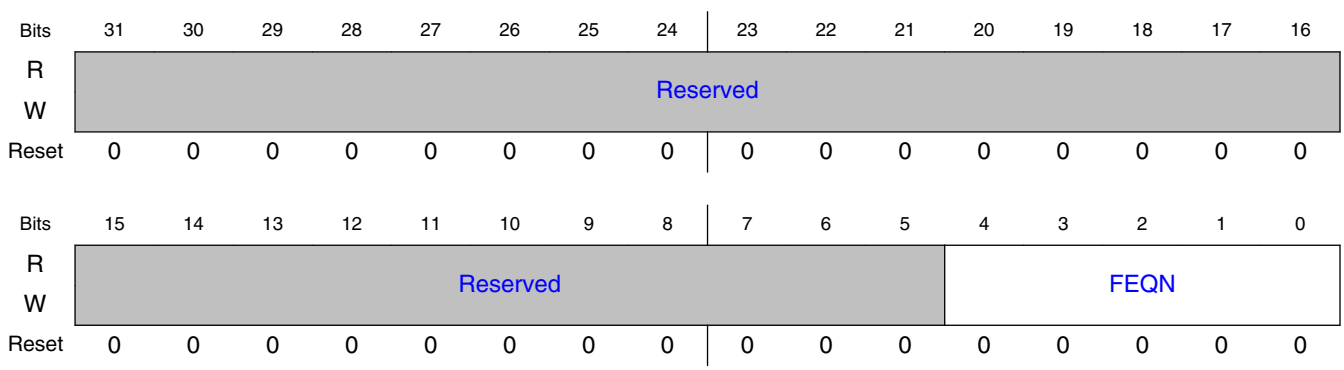
11.7.6.1.380 EST Frame Size Error (MTL_EST_FRM_SIZE_ERROR)

This register provides the One Hot encoded Queue Numbers that are having the Frame Size related error.

11.7.6.1.380.1 Offset

Register	Offset
MTL_EST_FRM_SIZE_ERROR	C64h

11.7.6.1.380.2 Diagram



11.7.6.1.380.3 Fields

Field	Description
31-5 —	Reserved
4-0 FEQN	Frame Size Error Queue Number The One Hot Encoded Queue Numbers that have experienced error described in HLBF field of status register. Frame Size Error Queue Number

Field	Description
	The One Hot Encoded Queue Numbers that have experienced error described in HLBF field of status register. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

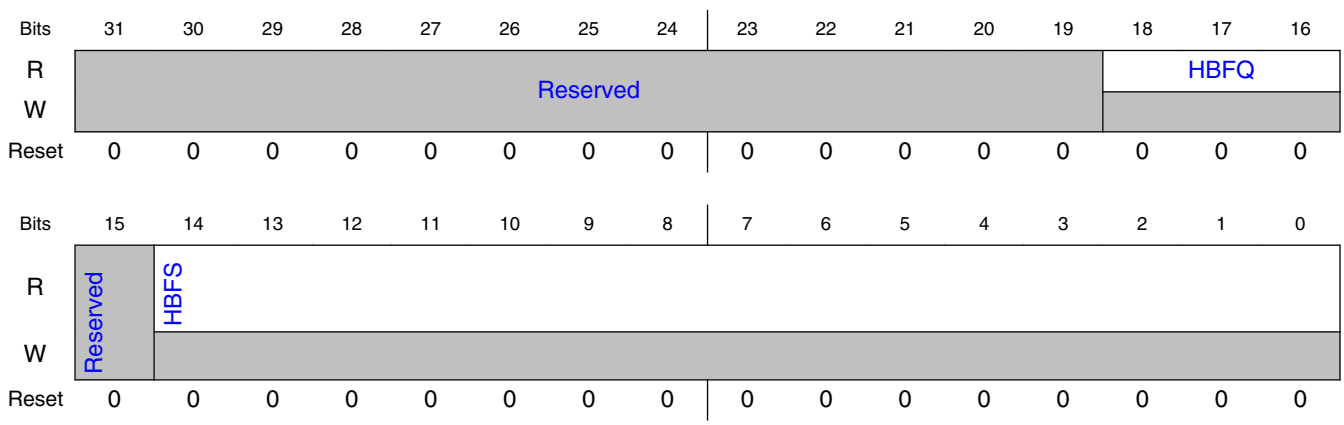
11.7.6.1.381 EST Frame Size Capture (MTL_EST_FRM_SIZE_CAPTURE)

This register captures the Frame Size and Queue Number of the first occurrence of the Frame Size related error. Up on clearing it captures the data of immediate next occurrence of a similar error.

11.7.6.1.381.1 Offset

Register	Offset
MTL_EST_FRM_SIZE_CAPTURE	C68h

11.7.6.1.381.2 Diagram



11.7.6.1.381.3 Fields

Field	Description
31-19 —	Reserved
18-16	Queue Number of HLBF Captures the binary value of the of the first Queue (number) experiencing HLBF error (see HLBF field of status register).

Table continues on the next page...

Field	Description
HBFQ	Queue Number of HLBF Captures the binary value of the of the first Queue (number) experiencing HLBF error (see HLBF field of status register). Value once written is not altered by any subsequent queue errors of similar nature. Once cleared the queue number of the next occurring HLBF error is captured. Width is based on the number of Tx Queues configured; remaining bits are Read-Only. Cleared when MTL_EST_FRM_SIZE_ERROR register is all zeros.
15 —	Reserved
14-0 HBFS	Frame Size of HLBF Captures the Frame Size of the dropped frame related to queue number indicated in HBFQ field of this register. Frame Size of HLBF Captures the Frame Size of the dropped frame related to queue number indicated in HBFQ field of this register. Contents of this register should be considered invalid, if this field is zero. Cleared when MTL_EST_FRM_SIZE_ERROR register is all zeros.

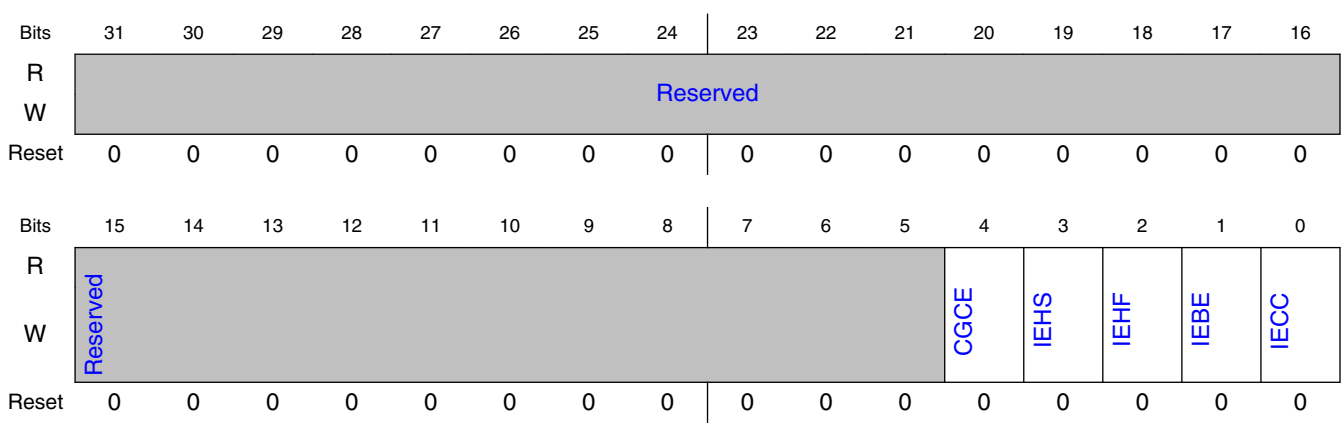
11.7.6.1.382 EST Interrupt Enable (MTL_EST_INTR_ENABLE)

This register implements the Interrupt Enable bits for the various events that generate an interrupt. Bit positions have a 1 to 1 correlation with the status bit positions in MTL_ETS_STATUS register.

11.7.6.1.382.1 Offset

Register	Offset
MTL_EST_INTR_ENABLE	C70h

11.7.6.1.382.2 Diagram



11.7.6.1.382.3 Fields

Field	Description
31-5 —	Reserved
4 CGCE	Interrupt Enable for CGCE When set, generates interrupt when the Constant Gate Control Error occurs and is indicated in the status. Interrupt Enable for CGCE When set, generates interrupt when the Constant Gate Control Error occurs and is indicated in the status. When reset this event does not generate an interrupt 0 - Interrupt for CGCE is disabled 1 - Interrupt for CGCE is enabled
3 IEHS	Interrupt Enable for HLBS When set, generates interrupt when the Head-of-Line Blocking due to Scheduling issue and is indicated in the status. Interrupt Enable for HLBS When set, generates interrupt when the Head-of-Line Blocking due to Scheduling issue and is indicated in the status. When reset this event does not generate an interrupt. 0 - Interrupt for HLBS is disabled 1 - Interrupt for HLBS is enabled
2 IEHF	Interrupt Enable for HLBF When set, generates interrupt when the Head-of-Line Blocking due to Frame Size error occurs and is indicated in the status. Interrupt Enable for HLBF When set, generates interrupt when the Head-of-Line Blocking due to Frame Size error occurs and is indicated in the status. When reset this event does not generate an interrupt. 0 - Interrupt for HLBF is disabled 1 - Interrupt for HLBF is enabled
1 IEBE	Interrupt Enable for BTR Error When set, generates interrupt when the BTR Error occurs and is indicated in the status. Interrupt Enable for BTR Error When set, generates interrupt when the BTR Error occurs and is indicated in the status. When reset this event does not generate an interrupt. 0 - Interrupt for BTR Error is disabled 1 - Interrupt for BTR Error is enabled
0 IECC	Interrupt Enable for Switch List When set, generates interrupt when the configuration change is successful and the hardware has switched to the new list. Interrupt Enable for Switch List When set, generates interrupt when the configuration change is successful and the hardware has switched to the new list. When reset this event does not generate an interrupt. 0 - Interrupt for Switch List is disabled 1 - Interrupt for Switch List is enabled

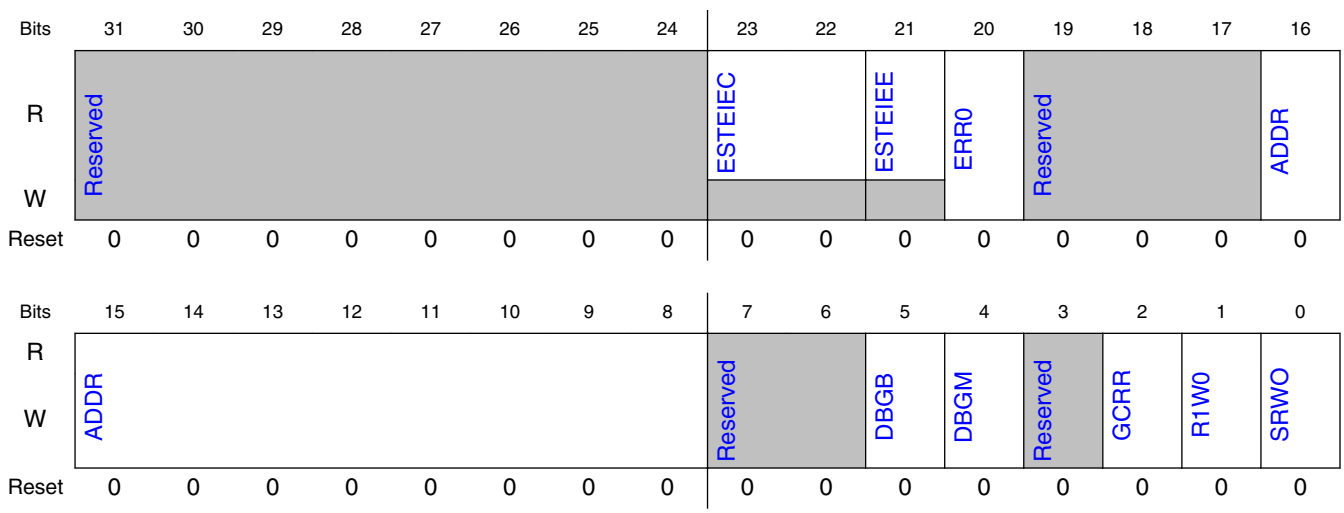
11.7.6.1.383 EST GCL Control (MTL_EST_GCL_CONTROL)

This register provides the control information for reading/writing to the Gate Control lists.

11.7.6.1.383.1 Offset

Register	Offset
MTL_EST_GCL_CONTROL	C80h

11.7.6.1.383.2 Diagram



11.7.6.1.383.3 Fields

Field	Description
31-24 —	Reserved
23-22 ESTEIEC	ECC Inject Error Control for EST Memory When EIEE bit of this register is set, following are the errors inserted based on the value encoded in this field. ECC Inject Error Control for EST Memory When EIEE bit of this register is set, following are the errors inserted based on the value encoded in this field. This filed will be valid only if DWC_EQOS_ASP_ECC feature is selected during the configuration, else it will be reserved. 00 - Insert 1 bit error 01 - Insert 2 bit errors

Table continues on the next page...

Field	Description
	10 - Insert 3 bit errors 11 - Insert 1 bit error in address field
21 ESTEIEE	EST ECC Inject Error Enable When set along with EEST bit of MTL_EST_CONTROL register, enables the ECC error injection feature. EST ECC Inject Error Enable When set along with EEST bit of MTL_EST_CONTROL register, enables the ECC error injection feature. When reset, disables the ECC error injection feature. 0 - EST ECC Inject Error is disabled 1 - EST ECC Inject Error is enabled
20 ERR0	When set indicates the last write operation was aborted as software writes to GCL and GCL registers is prohibited when SSWL bit of MTL_EST_CONTROL Register is set. When set indicates the last write operation was aborted as software writes to GCL and GCL registers is prohibited when SSWL bit of MTL_EST_CONTROL Register is set. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - ERR0 is disabled 1 - ERR1 is enabled
19-17 —	Reserved
16-8 ADDR	Gate Control List Address: (GCLA when GCRR is "0"). Gate Control List Address: (GCLA when GCRR is "0"). Provides the address (row number) of the Gate Control List at which the R/W operation has to be performed. By default the Gate Control List pointed by SWOL of MTL_EST_STATUS is selected for R/W, however if the DBGM bit of this register is set, a debug mode access is given to R/W from DBGB. The width of this field is dependent on DWC_EQOS_EST_DEP; unused bits should be treated as read only. Gate Control list Related Registers Address: (GCRA when GCRR is "1"). By default the GCL related register set pointed by SWOL of MTL_EST_STATUS is selected for R/W, however if the DBGM bit of this register is set, a debug mode access is given to R/W from DBGB. Lower 3 bits are only used in this mode, higher order bits are treated as dont cares. - 000: BTR Low (31:0) - 001: BTR High (63:31) - 010: CTR Low (31:0) - 011: CTR High (39:32) - 100: TER (31:0) - 101: LLR (n:0) (where n is (log{DWC_EQOS_EST_DEP} / log2)) - Others: Reserved
7-6 —	Reserved
5 DBGB	Debug Mode Bank Select When set to "0" indicates R/W in debug mode should be directed to Bank 0 (GCL0 and corresponding Time related registers). Debug Mode Bank Select

Table continues on the next page...

Field	Description
	When set to "0" indicates R/W in debug mode should be directed to Bank 0 (GCL0 and corresponding Time related registers). When set to "1" indicates R/W in debug mode should be directed to Bank 1 (GCL1 and corresponding Time related registers). This value is used when DBGM is set and overrides by value of SWOL which is normally used. 0 - R/W in debug mode should be directed to Bank 0 1 - R/W in debug mode should be directed to Bank 1
4 DBGM	Debug Mode When set to "1" indicates R/W in debug mode where the memory bank (for GCL and Time related registers) is explicitly provided by DBGB value, when set to "0" SWOL bit is used to determine which bank to use. 0 - Debug Mode is disabled 1 - Debug Mode is enabled
3 —	Reserved
2 GCRR	Gate Control Related Registers When set to "1" indicates the R/W access is for the GCL related registers (BTR, CTR, TER, LLR) whose address is provided by GCRA. Gate Control Related Registers When set to "1" indicates the R/W access is for the GCL related registers (BTR, CTR, TER, LLR) whose address is provided by GCRA. When "0" indicates R/W should be directed to GCL from the address provided by GCLA. 0 - Gate Control Related Registers are disabled 1 - Gate Control Related Registers are enabled
1 R1W0	Read '1', Write '0': When set to '1': Read Operation When set to '0': Write Operation. 0 - Write Operation 1 - Read Operation
0 SRWO	Start Read/Write Op When set indicates a Read/Write Op has started and is in progress. Start Read/Write Op When set indicates a Read/Write Op has started and is in progress. When reset by hardware indicates the R/W Op has completed or an error has occurred (when bit 20 is set) Reads: Data can be read from MTL_EST_GCL_DATA register after this bit is reset Writes: MTL_EST_GCL_DATA should be programmed with write data before setting SRWO. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Start Read/Write Op disabled 1 - Start Read/Write Op enabled

11.7.6.1.384 EST GCL Data (MTL_EST_GCL_DATA)

This register holds the read data or write data in case of reads and writes respectively.

11.7.6.1.384.1 Offset

Register	Offset
MTL_EST_GCL_DATA	C84h

11.7.6.1.384.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GCD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GCD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.384.3 Fields

Field	Description
31-0 GCD	Gate Control Data The data corresponding to the address selected in the MTL_GCL_CONTROL register. Gate Control Data The data corresponding to the address selected in the MTL_GCL_CONTROL register. Used for both Read and Write operations.

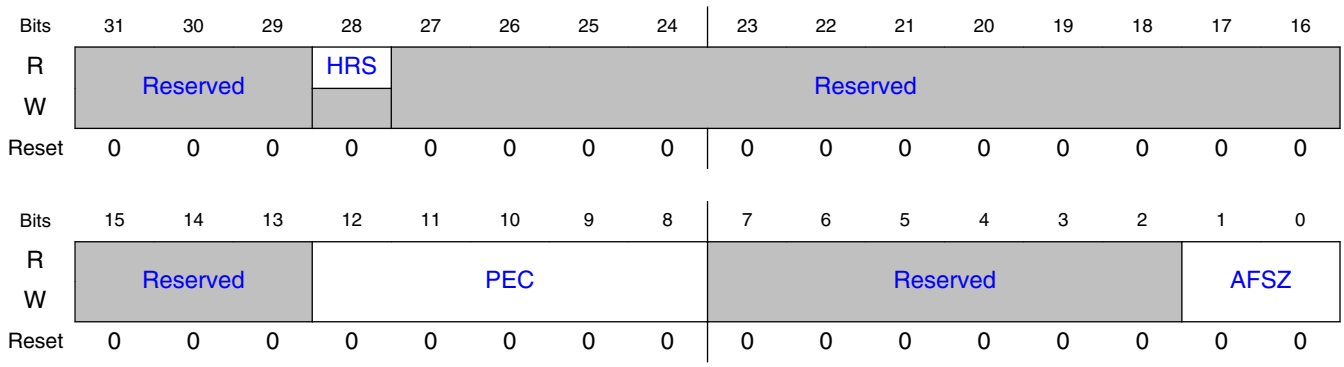
11.7.6.1.385 Frame Preemption Control and Status (MTL_FPE_CTRL_STS)

This register controls the operation of, and provides status for Frame Preemption (IEEE802.1Qbu/802.3br).

11.7.6.1.385.1 Offset

Register	Offset
MTL_FPE_CTRL_STS	C90h

11.7.6.1.385.2 Diagram



11.7.6.1.385.3 Fields

Field	Description
31-29 —	Reserved
28 HRS	Hold/Release Status - 1: Indicates a Set-and-Hold-MAC operation was last executed and the pMAC is in Hold State. Hold/Release Status - 1: Indicates a Set-and-Hold-MAC operation was last executed and the pMAC is in Hold State. - 0: Indicates a Set-and-Release-MAC operation was last executed and the pMAC is in Release State. 0 - Indicates a Set-and-Release-MAC operation was last executed and the pMAC is in Release State 1 - Indicates a Set-and-Hold-MAC operation was last executed and the pMAC is in Hold State
27-16 —	Reserved
15-13 —	Reserved
12-8 PEC	Preemption Classification When set indicates the corresponding Queue must be classified as preemptable, when '0' Queue is classified as express. Preemption Classification When set indicates the corresponding Queue must be classified as preemptable, when '0' Queue is classified as express. When both EST (Qbv) and Preemption are enabled, Queue-0 is always assumed to be preemptable. When EST (Qbv) is enabled Queues categorized as preemptable here are always assumed to be in "Open" state in the Gate Control List.
7-2 —	Reserved
1-0 AFSZ	Additional Fragment Size used to indicate, in units of 64 bytes, the minimum number of bytes over 64 bytes required in non-final fragments of preempted frames. Additional Fragment Size used to indicate, in units of 64 bytes, the minimum number of bytes over 64 bytes required in non-final fragments of preempted frames. The minimum non-final fragment size is (AFSZ +1) * 64 bytes

11.7.6.1.386 Frame Preemption Hold and Release Advance (MTL_FPE_ADVANCE)

This register holds the Hold and Release Advance time.

11.7.6.1.386.1 Offset

Register	Offset
MTL_FPE_ADVANCE	C94h

11.7.6.1.386.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RADV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HADV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.386.3 Fields

Field	Description
31-16 RADV	Release Advance The maximum time in nanoseconds that can elapse between issuing a RELEASE to the MAC and the MAC being ready to resume transmission of preemptable frames, in the absence of there being any express frames available for transmission.
15-0 HADV	Hold Advance The maximum time in nanoseconds that can elapse between issuing a HOLD to the MAC and the MAC ceasing to transmit any preemptable frame that is in the process of transmission or any preemptable frames that are queued for transmission.

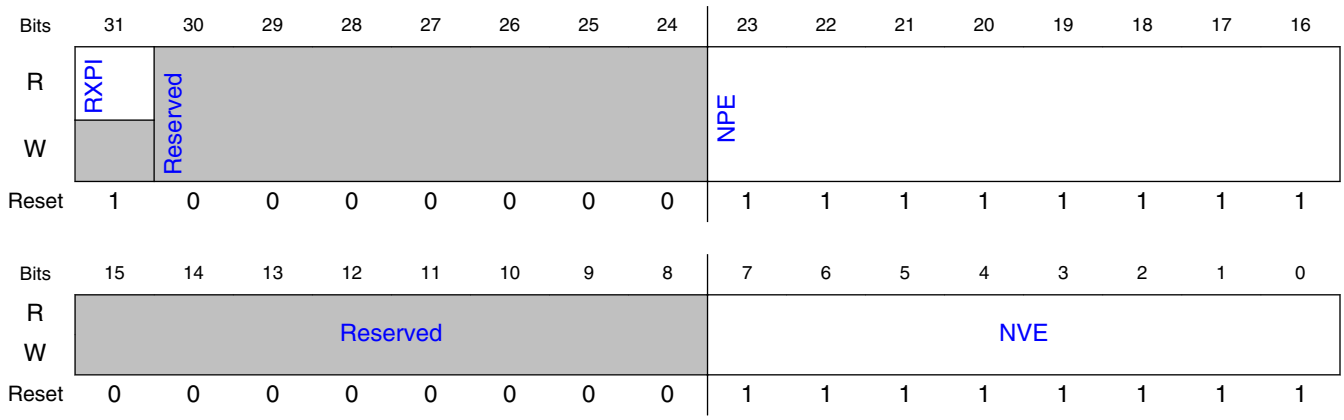
11.7.6.1.387 RXP Control Status (MTL_RXP_CONTROL_STATUS)

The RXP Control Status register establishes the operating mode of Rx Parser and provides some status.

11.7.6.1.387.1 Offset

Register	Offset
MTL_RXP_CONTROL_STATUS	CA0h

11.7.6.1.387.2 Diagram



11.7.6.1.387.3 Fields

Field	Description
31 RXPI	RX Parser in Idle state This status bit is set to 1 when the Rx parser is in Idle State and waiting for a new packet for processing. RX Parser in Idle state This status bit is set to 1 when the Rx parser is in Idle State and waiting for a new packet for processing. This bit is used as a handshake with software when parser gets disables. After disabling, when bit is set then software can update the Rx parser instruction table. 0 - RX Parser not in Idle state 1 - RX Parser in Idle state
30-24 —	Reserved
23-16 NPE	Number of parsable entries in the Instruction table This control indicates the number of parsable entries in the Instruction Memory. Number of parsable entries in the Instruction table This control indicates the number of parsable entries in the Instruction Memory. This is used in Rx parser logic to detect programming Error. In case number of parsed entries for a packet is more than this entry then NPEOVIS bit in the MTL_RXP_INTERRUPT_CONTROL_STATUS register is set.
15-8 —	Reserved

Table continues on the next page...

Field	Description
7-0 NVE	Number of valid entries in the Instruction table This control indicates the number of valid entries in the Instruction Memory. Number of valid entries in the Instruction table This control indicates the number of valid entries in the Instruction Memory. This is used in Rx parser logic to detect any programming Error. In case while parsing Table address (memory address) found to be more than this entry then NVEOVIS bit in the MTL_RXP_INTERRUPT_CONTROL_STATUS register is set. Note: The minimum value of this should be 2.

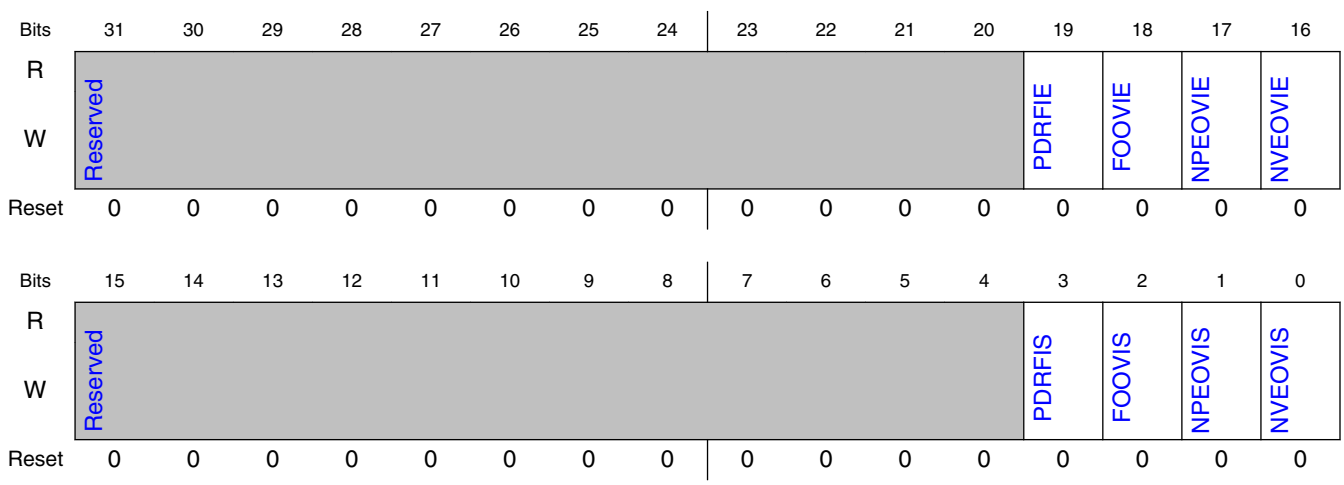
11.7.6.1.388 RXP Interrupt Control Status (MTL_RXP_INTERRUPT_CONTROL_STATUS)

The RXP Interrupt Control Status register provides enable control for the interrupts and provides interrupt status.

11.7.6.1.388.1 Offset

Register	Offset
MTL_RXP_INTERRUPT_CONTROL_STATUS	CA4h

11.7.6.1.388.2 Diagram



11.7.6.1.388.3 Fields

Field	Description
31-20 —	Reserved
19 PDRFIE	Packet Drop due to RF Interrupt Enable When this bit is set, the PDRFIS interrupt is enabled. Packet Drop due to RF Interrupt Enable When this bit is set, the PDRFIS interrupt is enabled. When this bit is reset, the PDRFIS interrupt is disabled. 0 - Packet Drop due to RF Interrupt is disabled 1 - Packet Drop due to RF Interrupt is enabled
18 FOOVIE	Frame Offset Overflow Interrupt Enable When this bit is set, the FOOVIS interrupt is enabled. Frame Offset Overflow Interrupt Enable When this bit is set, the FOOVIS interrupt is enabled. When this bit is reset, the FOOVIS interrupt is disabled. 0 - Frame Offset Overflow Interrupt is disabled 1 - Frame Offset Overflow Interrupt is enabled
17 NPEOVIE	Number of Parsable Entries Overflow Interrupt Enable When this bit is set, the NPEOVIS interrupt is enabled. Number of Parsable Entries Overflow Interrupt Enable When this bit is set, the NPEOVIS interrupt is enabled. When this bit is reset, the NPEOVIS interrupt is disabled. 0 - Number of Parsable Entries Overflow Interrupt is disabled 1 - Number of Parsable Entries Overflow Interrupt is enabled
16 NVEOVIE	Number of Valid Entries Overflow Interrupt Enable When this bit is set, the NVEOVIS interrupt is enabled. Number of Valid Entries Overflow Interrupt Enable When this bit is set, the NVEOVIS interrupt is enabled. When this bit is reset, the NVEOVIS interrupt is disabled. 0 - Number of Valid Entries Overflow Interrupt is disabled 1 - Number of Valid Entries Overflow Interrupt is enabled
15-4 —	Reserved
3 PDRFIS	Packet Dropped due to RF Interrupt Status If the Rx Parser result says to drop the packet by setting RF=1 in the instruction memory, then this bit is set to 1. Packet Dropped due to RF Interrupt Status If the Rx Parser result says to drop the packet by setting RF=1 in the instruction memory, then this bit is set to 1. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Packet Dropped due to RF Interrupt Status not detected 1 - Packet Dropped due to RF Interrupt Status detected
2 FOOVIS	Frame Offset Overflow Interrupt Status While parsing if the Instruction table entry's 'Frame Offset' found to be more than EOF offset, then then this bit is set.

Table continues on the next page...

Field	Description
	Frame Offset Overflow Interrupt Status While parsing if the Instruction table entry's 'Frame Offset' found to be more than EOF offset, then then this bit is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Frame Offset Overflow Interrupt Status not detected 1 - Frame Offset Overflow Interrupt Status detected
1 NPEOVIS	Number of Parsable Entries Overflow Interrupt Status While parsing a packet if the number of parsed entries found to be more than NPE[] (Number of Parseable Entries in MTL_RXP_CONTROL register),then this bit is set to 1. Number of Parsable Entries Overflow Interrupt Status While parsing a packet if the number of parsed entries found to be more than NPE[] (Number of Parseable Entries in MTL_RXP_CONTROL register),then this bit is set to 1. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Number of Parsable Entries Overflow Interrupt Status not detected 1 - Number of Parsable Entries Overflow Interrupt Status detected
0 NVEOVIS	Number of Valid Entries Overflow Interrupt Status While parsing if the Instruction address found to be more than NVE (Number of Valid Entries in MTL_RXP_CONTROL register), then this bit is set to 1. Number of Valid Entries Overflow Interrupt Status While parsing if the Instruction address found to be more than NVE (Number of Valid Entries in MTL_RXP_CONTROL register), then this bit is set to 1. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Number of Valid Entries Overflow Interrupt Status not detected 1 - Number of Valid Entries Overflow Interrupt Status detected

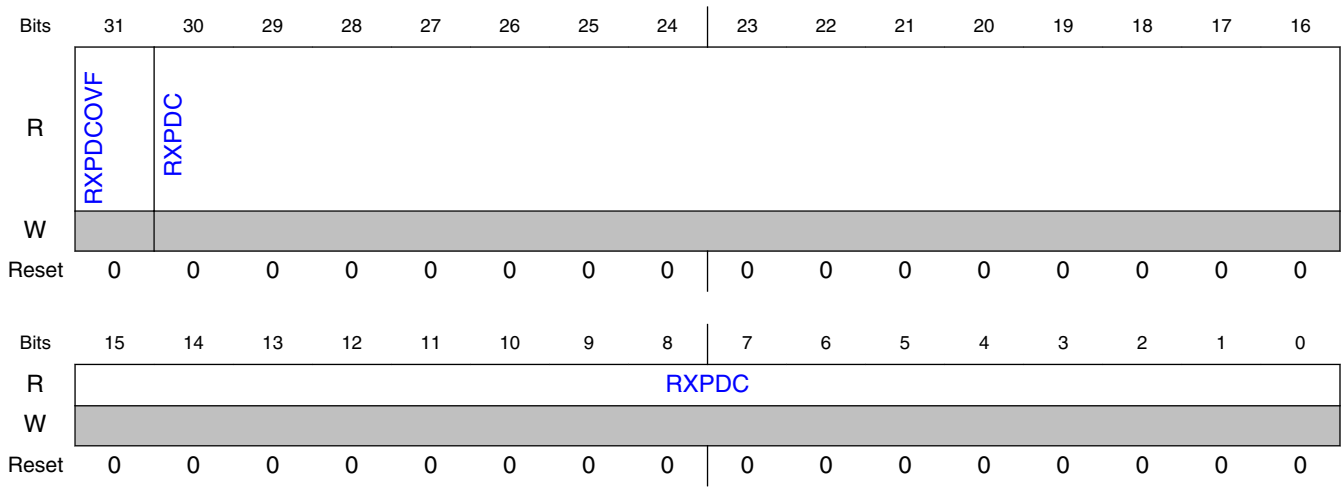
11.7.6.1.389 RXP Drop Count (MTL_RXP_DROP_CNT)

The MTL_RXP_DROP_CNT register provides the drop count of Rx Parser initiated drops.

11.7.6.1.389.1 Offset

Register	Offset
MTL_RXP_DROP_CNT	CA8h

11.7.6.1.389.2 Diagram



11.7.6.1.389.3 Fields

Field	Description
31 RXPDCOVF	Rx Parser Drop Counter Overflow Bit When set, this bit indicates that the MTL_RXP_DROP_CNT (RXPDC) Counter field crossed the maximum limit. Rx Parser Drop Counter Overflow Bit When set, this bit indicates that the MTL_RXP_DROP_CNT (RXPDC) Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Drop count overflow not occurred 1 - Rx Parser Drop count overflow occurred
30-0 RXPDC	Rx Parser Drop count This 31-bit counter is implemented whenever a Rx Parser Drops a packet due to RF =1. Rx Parser Drop count This 31-bit counter is implemented whenever a Rx Parser Drops a packet due to RF =1. The counter is cleared when the register is read.

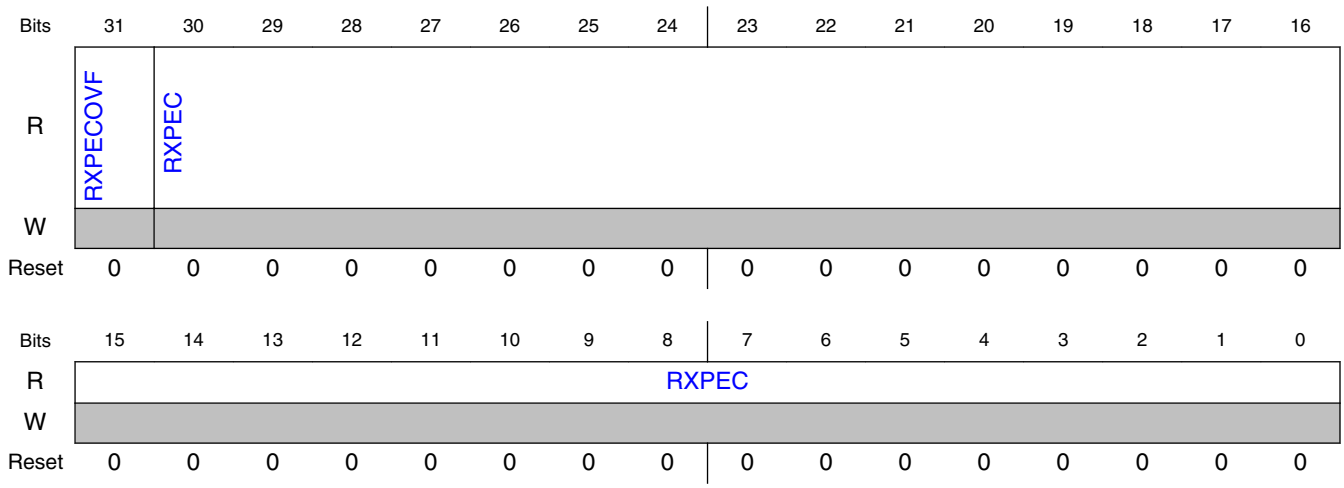
11.7.6.1.390 RXP Error Count (MTL_RXP_ERROR_CNT)

The RXP_ERROR_CNT register provides the Rx Parser related error occurrence count.

11.7.6.1.390.1 Offset

Register	Offset
MTL_RXP_ERROR_CNT	CACh

11.7.6.1.390.2 Diagram



11.7.6.1.390.3 Fields

Field	Description
31 RXPECOVF	Rx Parser Error Counter Overflow Bit When set, this bit indicates that the MTL_RXP_ERROR_CNT (RXPEC) Counter field crossed the maximum limit. Rx Parser Error Counter Overflow Bit When set, this bit indicates that the MTL_RXP_ERROR_CNT (RXPEC) Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Error count overflow not occurred 1 - Rx Parser Error count overflow occurred
30-0 RXPEC	Rx Parser Error count This 31-bit counter is implemented whenever a Rx Parser encounters following Error scenarios - Entry address >= NVE[] - Number Parsed Entries >= NPE[] - Entry address > EOF data entry address The counter is cleared when the register is read.

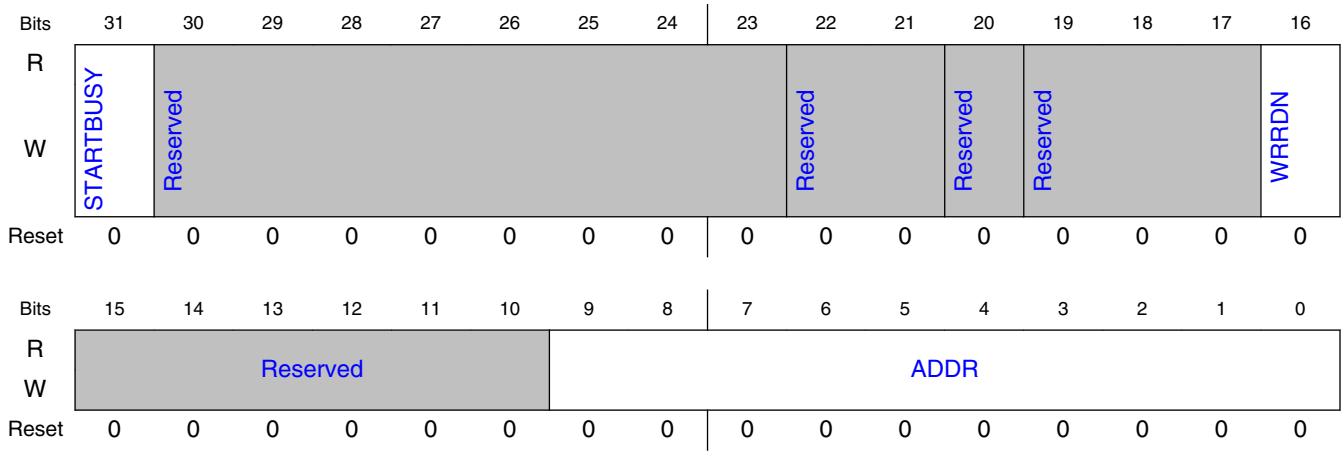
11.7.6.1.391 RXP Indirect Access Control and Status (MTL_RXP_INDIRECT_ACC_CONTROL_STATUS)

The RXP_INDIRECT_ACC_CONTROL_STATUS register provides the Indirect Access control and status for Rx Parser memory.

11.7.6.1.391.1 Offset

Register	Offset
MTL_RXP_INDIRECT_ACC_CONTROL_STATUS	CB0h

11.7.6.1.391.2 Diagram



11.7.6.1.391.3 Fields

Field	Description
31 STARTBUSY	FRP Instruction Table Access Busy When this bit is set to 1 by the software then it indicates to start the Read/Write operation from/to the Rx Parser Memory. FRP Instruction Table Access Busy When this bit is set to 1 by the software then it indicates to start the Read/Write operation from/to the Rx Parser Memory. Software should read this bit as 0 before issuing read or write request to access the Parser Memory Instructions. This bit when set to 1 indicates that hardware is busy until its gets cleared by hardware and software should not issue any read or write operation. 0 - hardware not busy 1 - hardware is busy (Read/Write operation from/to the Rx Parser Memory)
30-23 —	Reserved
22-21 —	Reserved
20 —	Reserved
19-17 —	Reserved
16 WRRDN	Read Write Control When this bit is set to 1 indicates the write operation to the Rx Parser Memory. Read Write Control When this bit is set to 1 indicates the write operation to the Rx Parser Memory. When this bit is set to 0 indicates the read operation to the Rx Parser Memory.

Table continues on the next page...

Field	Description
	0 - Read operation to the Rx Parser Memory 1 - Write operation to the Rx Parser Memory
15-10 —	Reserved
9-0 ADDR	FRP Instruction Table Offset Address This field indicates the ADDR of the 32-bit entry in Rx parser instruction table. FRP Instruction Table Offset Address This field indicates the ADDR of the 32-bit entry in Rx parser instruction table. Each entry has 128-bit (4x32-bit words). The X depends on the configurable DWC_EQOS_FRP_ENTRIES If DWC_EQOS_FRP_ENTRIES == 256 , then X = 9 If DWC_EQOS_FRP_ENTRIES = 128 , then X = 8 IF DWC_EQOS_FRP_ENTRIES = 64, then X = 7 This is can be written by the software before issuing any Read/Write command. The hardware auto-increments this field after the read-write operation gets completed.

11.7.6.1.392 RXP Indirect Access Data (MTL_RXP_INDIRECT_ACC_DATA)

The RXP_INDIRECT_ACC_DATA registers holds the data associated to Indirect Access to Rx Parser memory.

11.7.6.1.392.1 Offset

Register	Offset
MTL_RXP_INDIRECT_ACC_DATA	CB4h

11.7.6.1.392.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.392.3 Fields

Field	Description
31-0 DATA	FRP Instruction Table Write/Read Data Software should write this register before issuing any write command. FRP Instruction Table Write/Read Data Software should write this register before issuing any write command. The hardware provides the read data from the Rx Parser Memory for read operation when STARTBUSY =0 after read command.

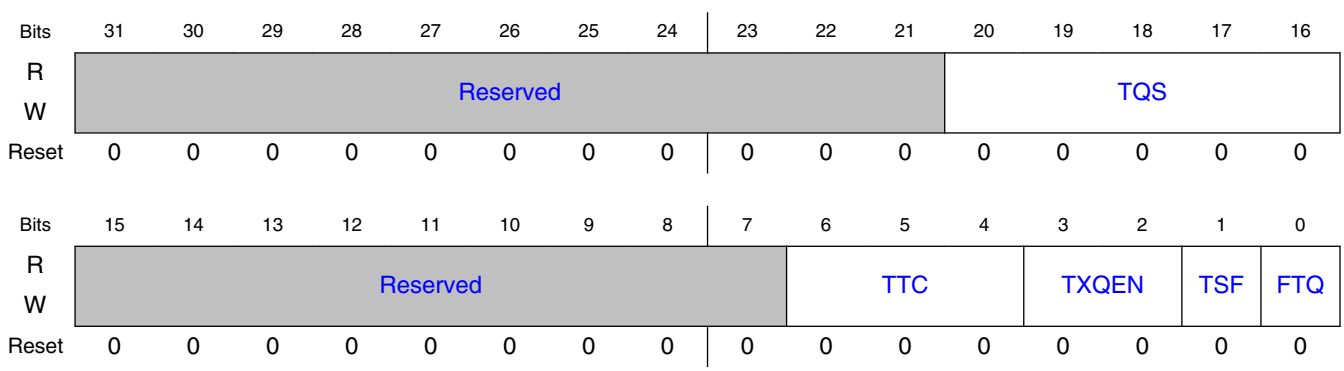
11.7.6.1.393 Queue 0 Transmit Operation Mode (MTL_TXQ0_OPERATION_MODE)

The Queue 0 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

11.7.6.1.393.1 Offset

Register	Offset
MTL_TXQ0_OPERATION_MODE	D00h

11.7.6.1.393.2 Diagram



11.7.6.1.393.3 Fields

Field	Description
31-21	Reserved

Table continues on the next page...

Field	Description
—	
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000 - 32 001 - 64 010 - 96 011 - 128 100 - 192 101 - 256 110 - 384 111 - 512
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Reserved - 2'b10: Enabled - 2'b11: Reserved This field is Read Only in Single Queue configurations and Read Write in Multiple Queue configurations. Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00 - Not enabled 01 - Enable in AV mode (Reserved in non-AV) 10 - Enabled 11 - Reserved

Table continues on the next page...

Field	Description
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0 - Transmit Store and Forward is disabled 1 - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ1_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flush Transmit Queue is disabled 1 - Flush Transmit Queue is enabled

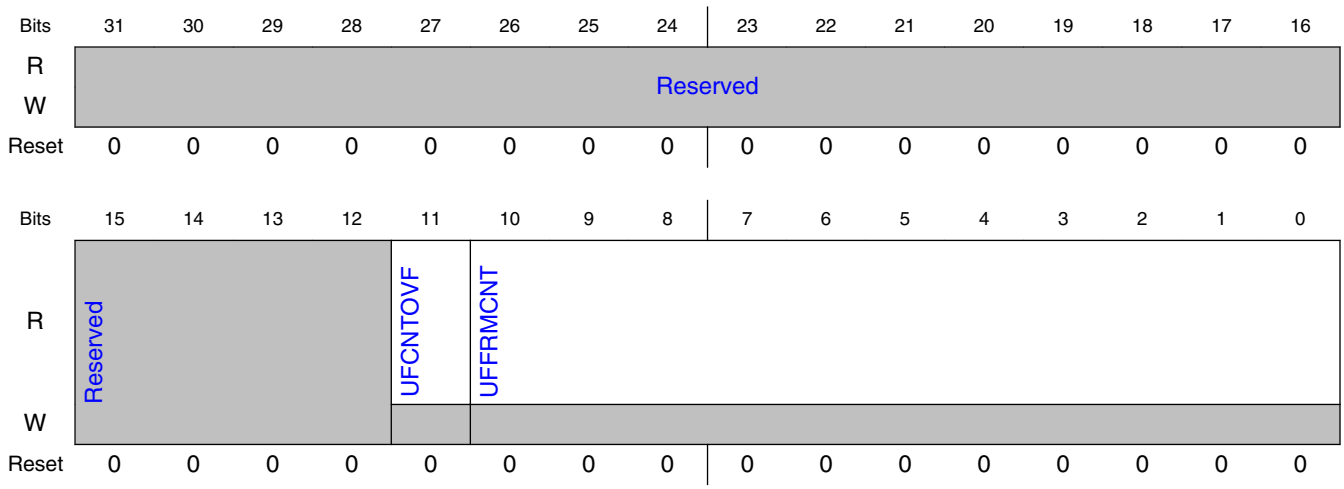
11.7.6.1.394 Queue 0 Underflow Counter (MTL_TXQ0_UNDERFLOW)

The Queue 0 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

11.7.6.1.394.1 Offset

Register	Offset
MTL_TXQ0_UNDERFLOW	D04h

11.7.6.1.394.2 Diagram



11.7.6.1.394.3 Fields

Field	Description
31-12 —	Reserved
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow not detected for Underflow Packet Counter 1 - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

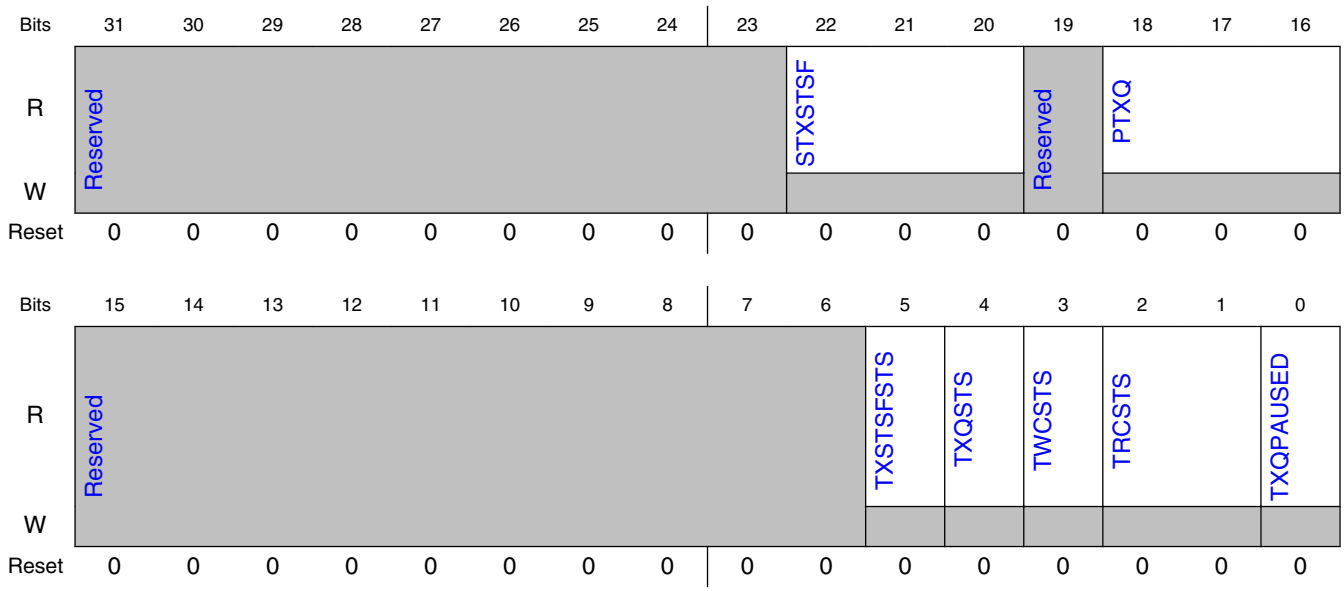
11.7.6.1.395 Queue 0 Transmit Debug (MTL_TXQ0_DEBUG)

The Queue 0 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

11.7.6.1.395.1 Offset

Register	Offset
MTL_TXQ0_DEBUG	D08h

11.7.6.1.395.2 Diagram



11.7.6.1.395.3 Fields

Field	Description
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue.

Table continues on the next page...

Field	Description
	When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0 - MTL Tx Status FIFO Full status is not detected 1 - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0 - MTL Tx Queue Not Empty status is not detected 1 - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0 - MTL Tx Queue Write Controller status is not detected 1 - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00 - Idle state 01 - Read state (transferring data to the MAC transmitter) 10 - Waiting for pending Tx Status from the MAC transmitter 11 - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0 - Transmit Queue in Pause status is not detected 1 - Transmit Queue in Pause status is detected

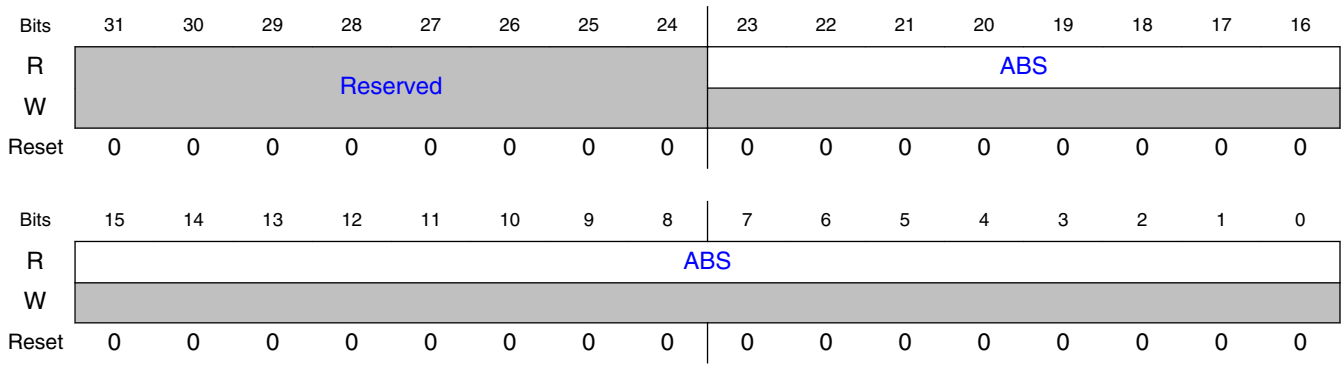
11.7.6.1.396 Queue 0 ETS Status (MTL_TXQ0_ETS_STATUS)

The Queue 0 ETS Status register provides the average traffic transmitted in Queue 0.

11.7.6.1.396.1 Offset

Register	Offset
MTL_TXQ0_ETS_STATUS	D14h

11.7.6.1.396.2 Diagram



11.7.6.1.396.3 Fields

Field	Description
31-24 —	Reserved
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. When the DCB operation is enabled for Queue 0, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

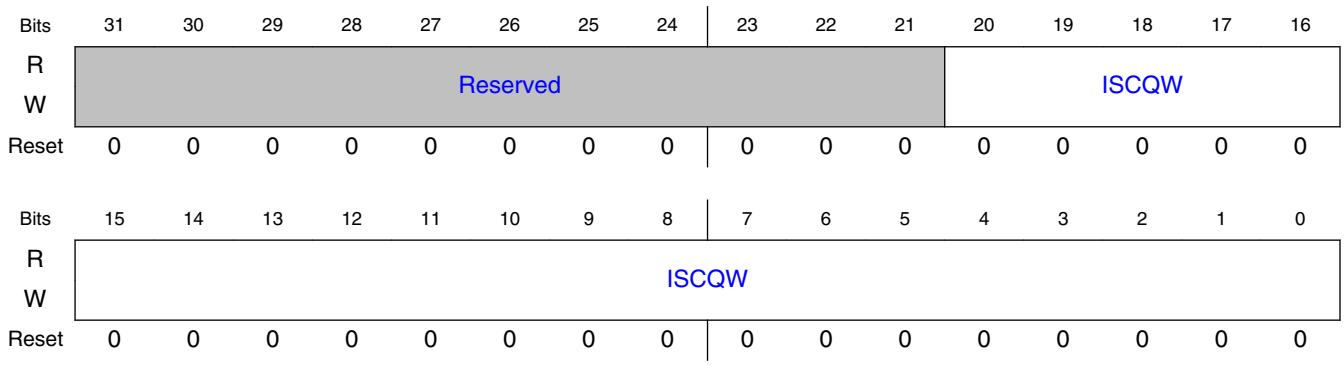
11.7.6.1.397 Queue 0 Quantum or Weights (MTL_TXQ0_QUANTUM_WEIGHT)

The Queue 0 Quantum or Weights register contains the quantum value for Deficit Weighted Round Robin (DWRR), weights for the Weighted Round Robin (WRR), and Weighted Fair Queuing (WFQ) for Queue 0.

11.7.6.1.397.1 Offset

Register	Offset
MTL_TXQ0_QUANTUM_WEIGHT	D18h

11.7.6.1.397.2 Diagram



11.7.6.1.397.3 Fields

Field	Description
31-21 —	Reserved
20-0 ISCQW	Quantum or Weights When the DCB operation is enabled with DWRR algorithm for Queue 0 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. Quantum or Weights When the DCB operation is enabled with DWRR algorithm for Queue 0 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. When DCB operation is enabled with WFQ algorithm for Queue 0 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. The higher the programmed weights lesser the bandwidth allocated for the particular Transmit Queue. This is because the weights are used to compute the packet finish time (weights*packet_size). Lesser the finish time, higher the probability of the packet getting scheduled first and using more bandwidth. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 0 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero.

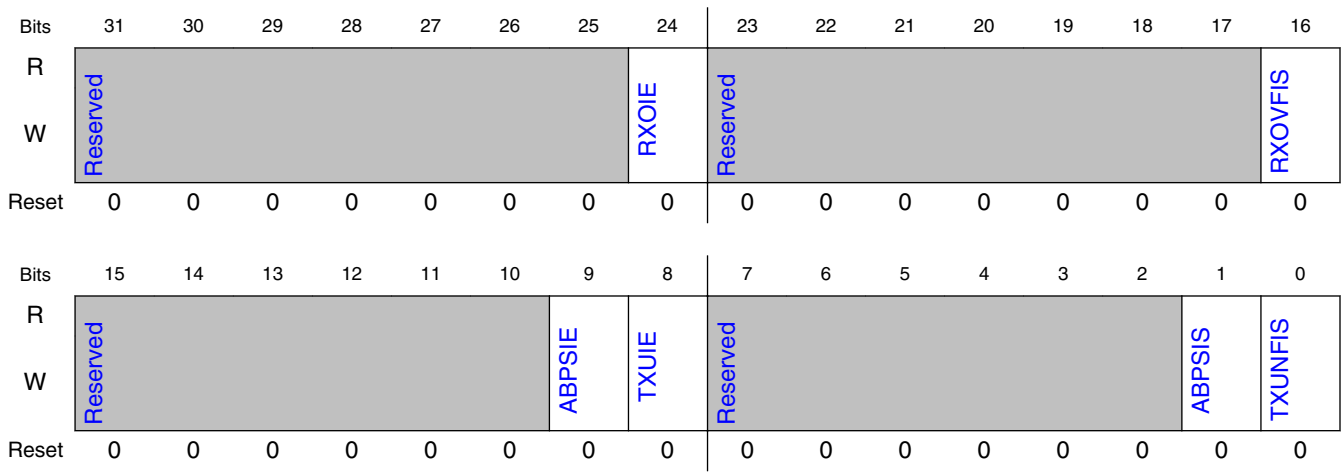
11.7.6.1.398 Queue 0 Interrupt Control Status (MTL_Q0_INTERRUPT_CONTROL_STATUS)

This register contains the interrupt enable and status bits for the queue 0 interrupts.

11.7.6.1.398.1 Offset

Register	Offset
MTL_Q0_INTERRUPT_CONTROL_STATUS	D2Ch

11.7.6.1.398.2 Diagram



11.7.6.1.398.3 Fields

Field	Description
31-25 —	Reserved
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0 - Receive Queue Overflow Interrupt is disabled 1 - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved

Table continues on the next page...

Field	Description
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Queue Overflow Interrupt Status not detected 1 - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9 ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0 - Average Bits Per Slot Interrupt is disabled 1 - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0 - Transmit Queue Underflow Interrupt Status is disabled 1 - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Average Bits Per Slot Interrupt Status not detected 1 - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

Field	Description
	0 - Transmit Queue Underflow Interrupt Status not detected 1 - Transmit Queue Underflow Interrupt Status detected

11.7.6.1.399 Queue 0 Receive Operation Mode (MTL_RXQ0_OPERATION_MODE)

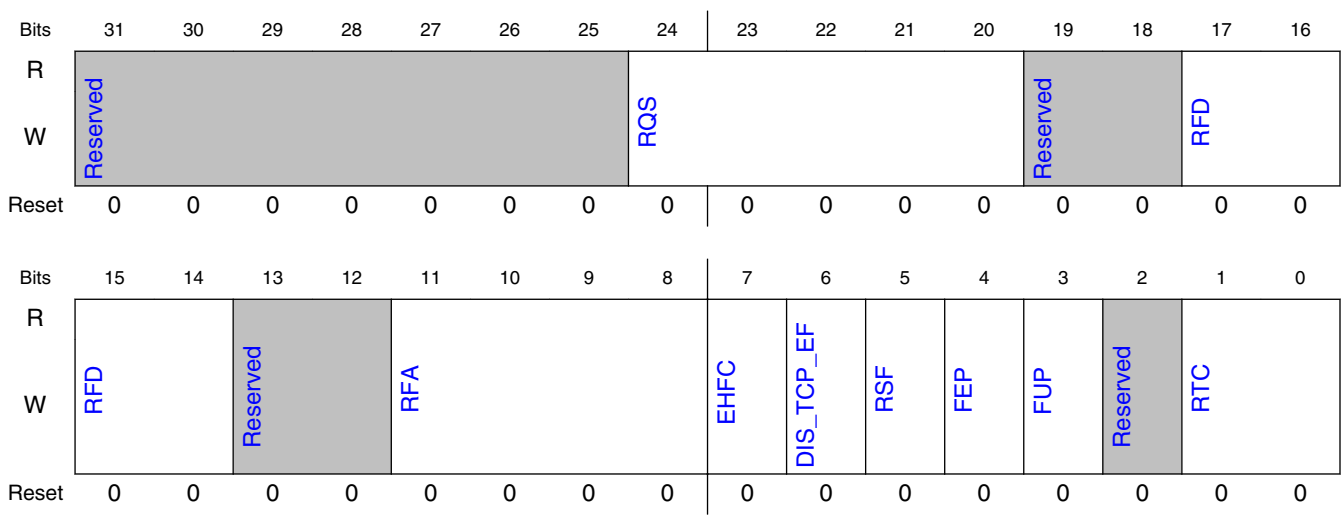
The Queue 0 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

11.7.6.1.399.1 Offset

Register	Offset
MTL_RXQ0_OPERATION_MODE	D30h

11.7.6.1.399.2 Diagram



11.7.6.1.399.3 Fields

Field	Description
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
19-18 —	Reserved
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.

Table continues on the next page...

Field	Description
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0 - Hardware Flow Control is disabled 1 - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0 - Dropping of TCP/IP Checksum Error Packets is enabled 1 - Dropping of TCP/IP Checksum Error Packets is disabled
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0 - Receive Queue Store and Forward is disabled 1 - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0 - Forward Error Packets is disabled 1 - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0 - Forward Undersized Good Packets is disabled

Table continues on the next page...

Field	Description
	1 - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00 - 64 01 - 32 10 - 96 11 - 128

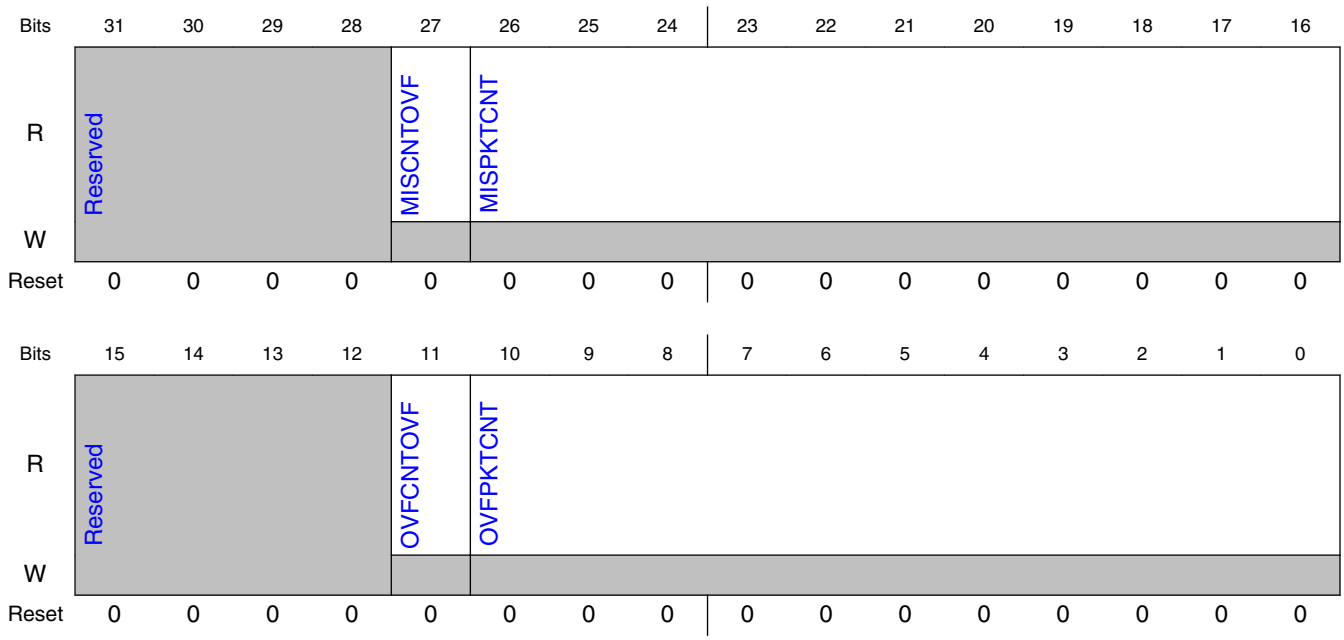
11.7.6.1.400 Queue 0 Missed Packet and Overflow Counter (MTL_RXQ0_MISSED_PACKET_OVERFLOW_CNT)

The Queue 0 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

11.7.6.1.400.1 Offset

Register	Offset
MTL_RXQ0_MISSED_PACKET_OVERFLOW_CNT	D34h

11.7.6.1.400.2 Diagram



11.7.6.1.400.3 Fields

Field	Description
31-28 —	Reserved
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Missed Packet Counter overflow not detected 1 - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved

Table continues on the next page...

Field	Description
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow Counter overflow not detected 1 - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

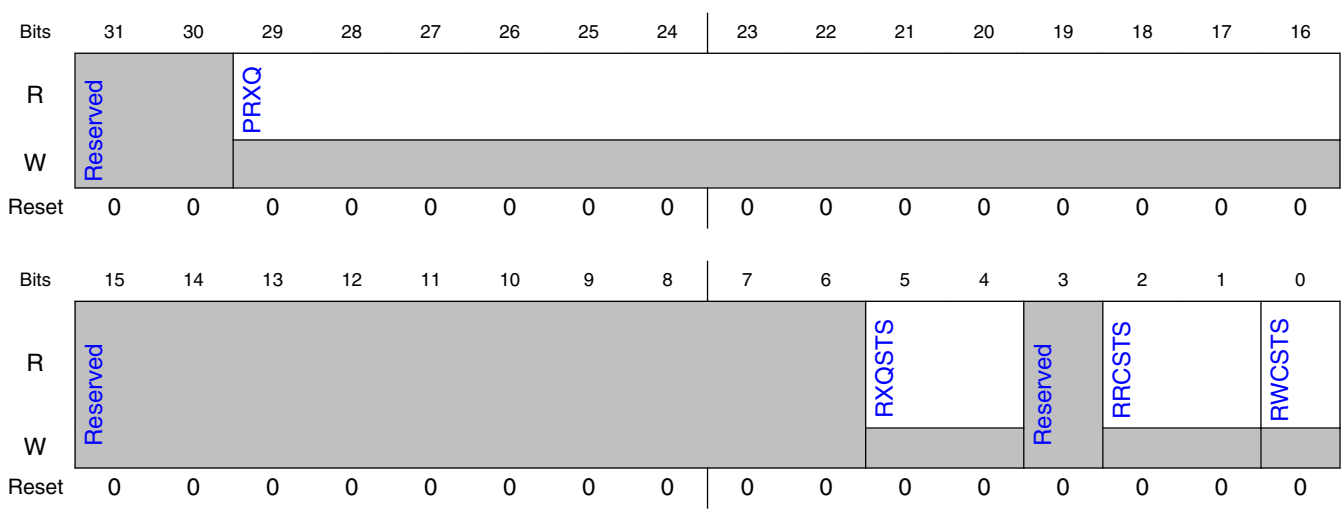
11.7.6.1.401 Queue 0 Receive Debug (MTL_RXQ0_DEBUG)

The Queue 0 Receive Debug register gives the debug status of various blocks related to the Receive queue.

11.7.6.1.401.1 Offset

Register	Offset
MTL_RXQ0_DEBUG	D38h

11.7.6.1.401.2 Diagram



11.7.6.1.401.3 Fields

Field	Description
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00 - Rx Queue empty 01 - Rx Queue fill-level below flow-control deactivate threshold 10 - Rx Queue fill-level above flow-control activate threshold 11 - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00 - Idle state 01 - Reading packet data 10 - Reading packet status (or timestamp) 11 - Flushing the packet data and status
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0 - MTL Rx Queue Write Controller Active Status not detected 1 - MTL Rx Queue Write Controller Active Status detected

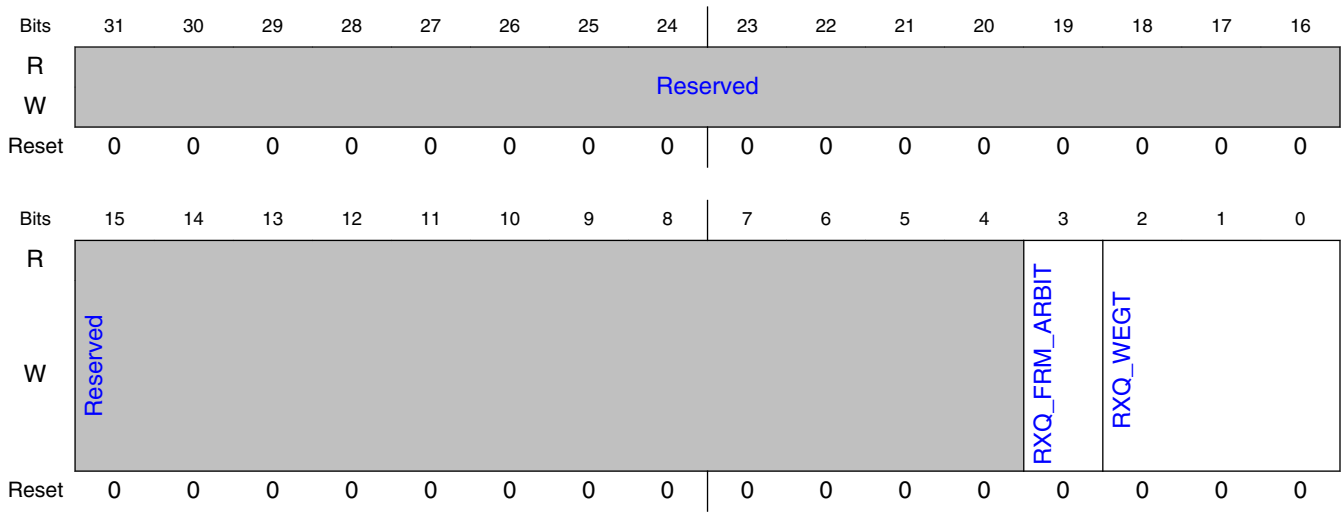
11.7.6.1.402 Queue 0 Receive Control (MTL_RXQ0_CONTROL)

The Queue 0 Receive Control register controls the receive arbitration and passing of received packets to the application.

11.7.6.1.402.1 Offset

Register	Offset
MTL_RXQ0_CONTROL	D3Ch

11.7.6.1.402.2 Diagram



11.7.6.1.402.3 Fields

Field	Description
31-4 —	Reserved
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0 - Receive Queue Packet Arbitration is disabled 1 - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

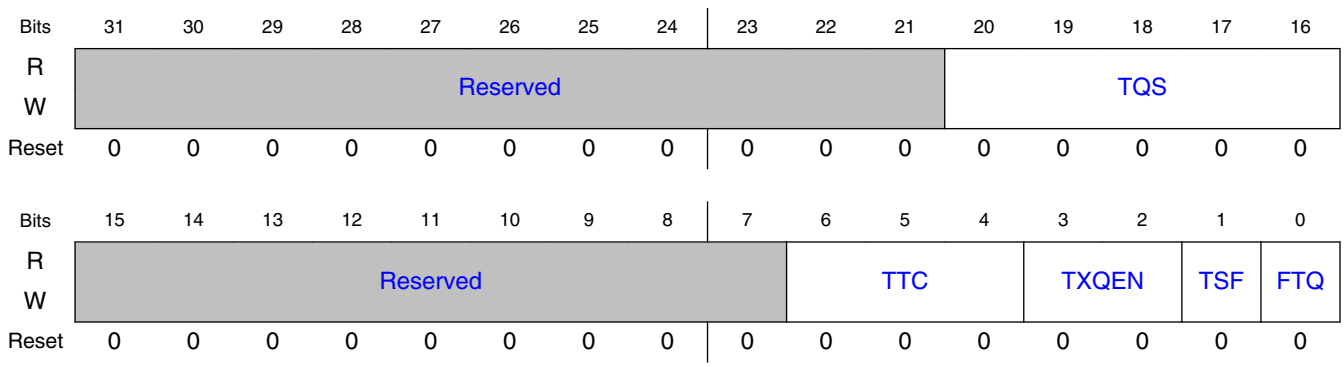
11.7.6.1.403 Queue 1 Transmit Operation Mode (MTL_TXQ1_OPERATION_MODE)

The Queue 1 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

11.7.6.1.403.1 Offset

Register	Offset
MTL_TXQ1_OPERATION_MODE	D40h

11.7.6.1.403.2 Diagram



11.7.6.1.403.3 Fields

Field	Description
31-21 —	Reserved
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved

Table continues on the next page...

Field	Description
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000 - 32 001 - 64 010 - 96 011 - 128 100 - 192 101 - 256 110 - 384 111 - 512
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Enable in AV mode - 2'b10: Enabled - 2'b11: Reserved Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00 - Not enabled 01 - Enable in AV mode (Reserved in non-AV) 10 - Enabled 11 - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0 - Transmit Store and Forward is disabled 1 - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ1_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission.

Field	Description
	Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flush Transmit Queue is disabled 1 - Flush Transmit Queue is enabled

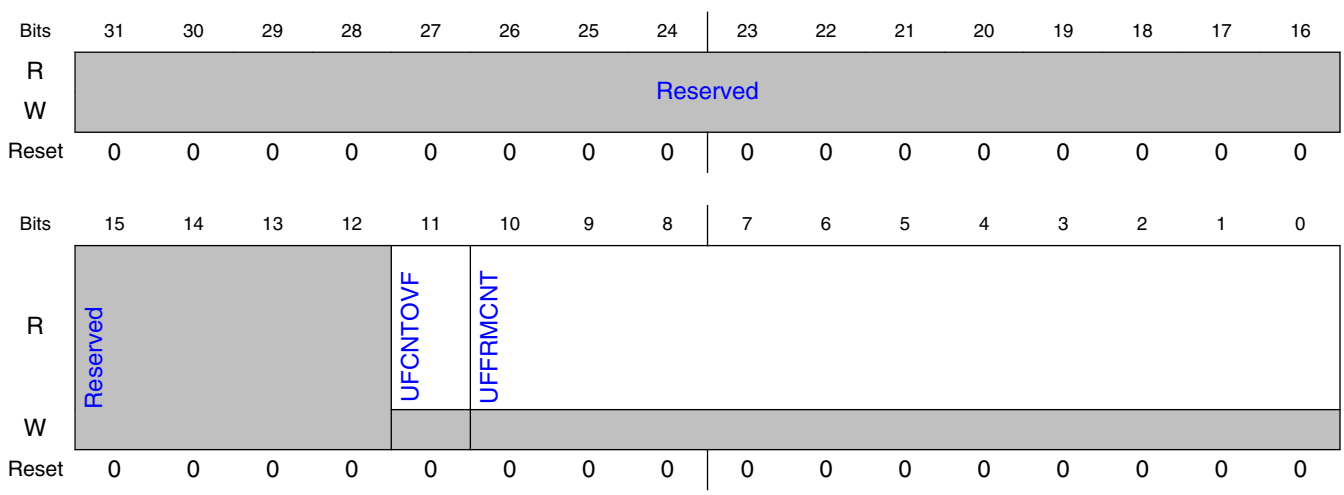
11.7.6.1.404 Queue 1 Underflow Counter (MTL_TXQ1_UNDERFLOW)

The Queue 1 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

11.7.6.1.404.1 Offset

Register	Offset
MTL_TXQ1_UNDERFLOW	D44h

11.7.6.1.404.2 Diagram



11.7.6.1.404.3 Fields

Field	Description
31-12	Reserved

Table continues on the next page...

Field	Description
—	
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow not detected for Underflow Packet Counter 1 - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

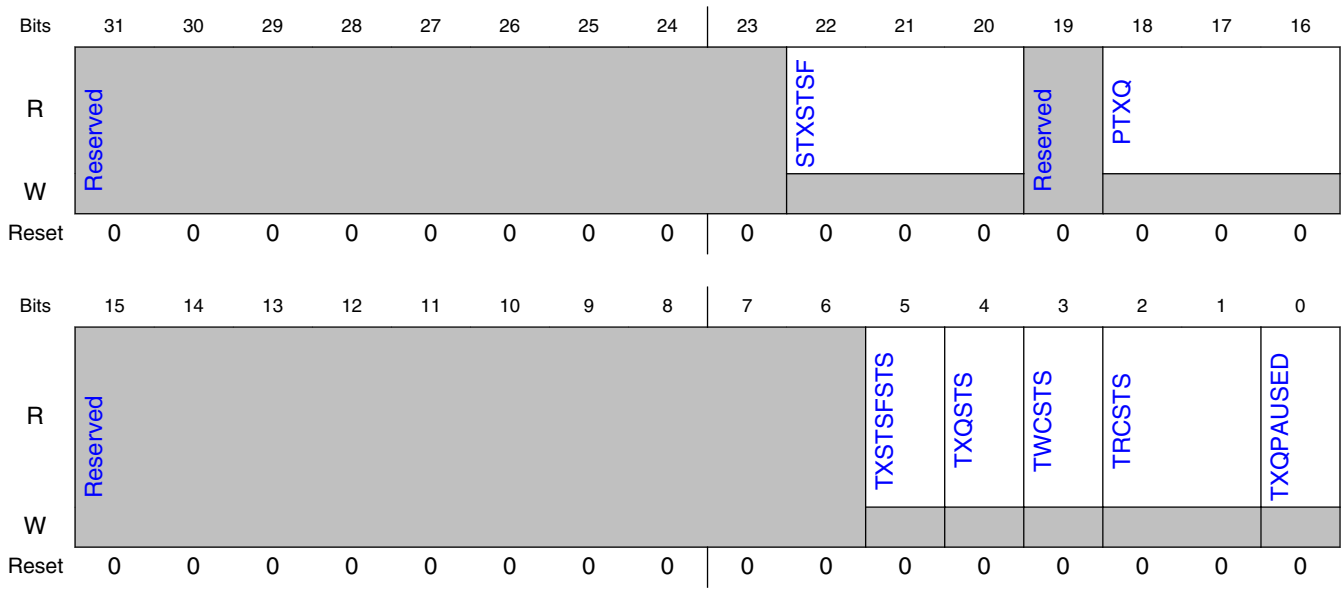
11.7.6.1.405 Queue 1 Transmit Debug (MTL_TXQ1_DEBUG)

The Queue 1 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

11.7.6.1.405.1 Offset

Register	Offset
MTL_TXQ1_DEBUG	D48h

11.7.6.1.405.2 Diagram



11.7.6.1.405.3 Fields

Field	Description
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission.

Table continues on the next page...

Field	Description
	0 - MTL Tx Status FIFO Full status is not detected 1 - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0 - MTL Tx Queue Not Empty status is not detected 1 - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0 - MTL Tx Queue Write Controller status is not detected 1 - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00 - Idle state 01 - Read state (transferring data to the MAC transmitter) 10 - Waiting for pending Tx Status from the MAC transmitter 11 - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0 - Transmit Queue in Pause status is not detected 1 - Transmit Queue in Pause status is detected

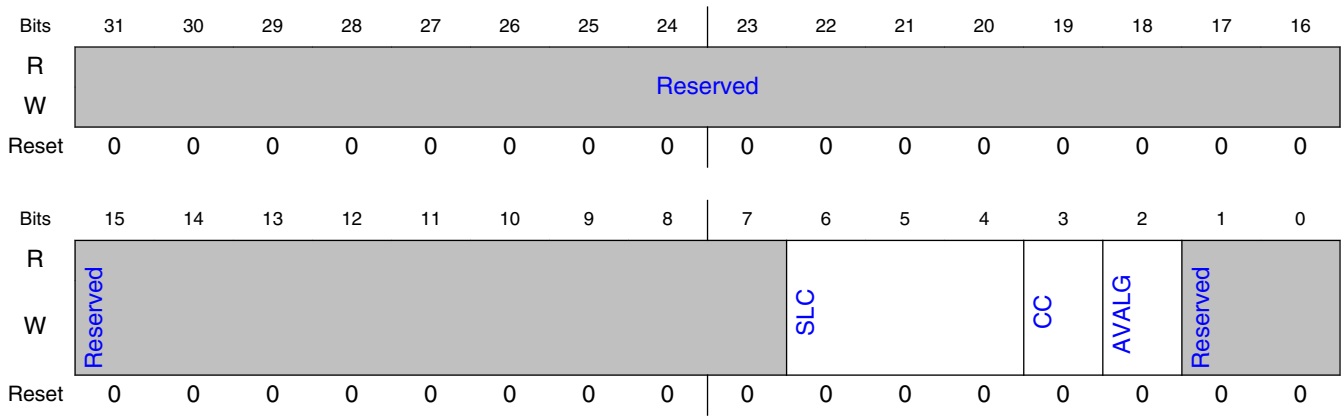
11.7.6.1.406 Queue 1 ETS Control (MTL_TXQ1_ETS_CONTROL)

The Queue 1 ETS Control register controls the enhanced transmission selection operation.

11.7.6.1.406.1 Offset

Register	Offset
MTL_TXQ1_ETS_CONTROL	D50h

11.7.6.1.406.2 Diagram



11.7.6.1.406.3 Fields

Field	Description
31-7 —	Reserved
6-4 SLC	Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH[n]_Slot_Interval register) over which the average transmitted bits per slot, provided in the MTL_TXQ[N]_ETS_STATUS register, need to be computed for Queue. Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH[n]_Slot_Interval register) over which the average transmitted bits per slot, provided in the MTL_TXQ[N]_ETS_STATUS register, need to be computed for Queue. The encoding is as follows: 000 - 1 slot 001 - 2 slots 010 - 4 slots 011 - 8 slots 100 - 16 slots 101 - Reserved
3 CC	Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 1. Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 1. The credit accumulates even when there is no packet waiting in Channel 1 and another channel is transmitting. When this bit is reset, the accumulated credit parameter in the credit-based shaper algorithm logic is set to zero when there is positive credit and no packet to transmit in Channel 1. When there is no packet waiting in Channel 1 and other channel is transmitting, no credit is accumulated. 0 - Credit Control is disabled 1 - Credit Control is enabled

Table continues on the next page...

Field	Description
2 AVALG	AV Algorithm When Queue 1 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 1 traffic. AV Algorithm When Queue 1 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 1 traffic. When reset, strict priority is selected. 0 - CBS Algorithm is disabled 1 - CBS Algorithm is enabled
1-0 —	Reserved

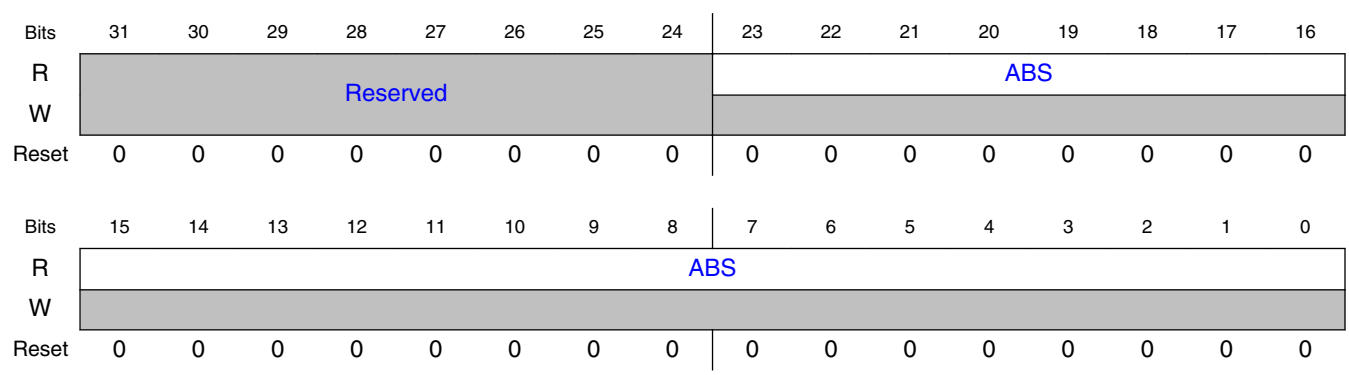
11.7.6.1.407 Queue 1 ETS Status (MTL_TXQ1_ETS_STATUS)

The Queue 1 ETS Status register provides the average traffic transmitted in Queue 1.

11.7.6.1.407.1 Offset

Register	Offset
MTL_TXQ1_ETS_STATUS	D54h

11.7.6.1.407.2 Diagram



11.7.6.1.407.3 Fields

Field	Description
31-24	Reserved

Table continues on the next page...

Field	Description
—	
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TXQ[N]_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps and 9C_4000 in 2500 Mbps mode respectively. When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

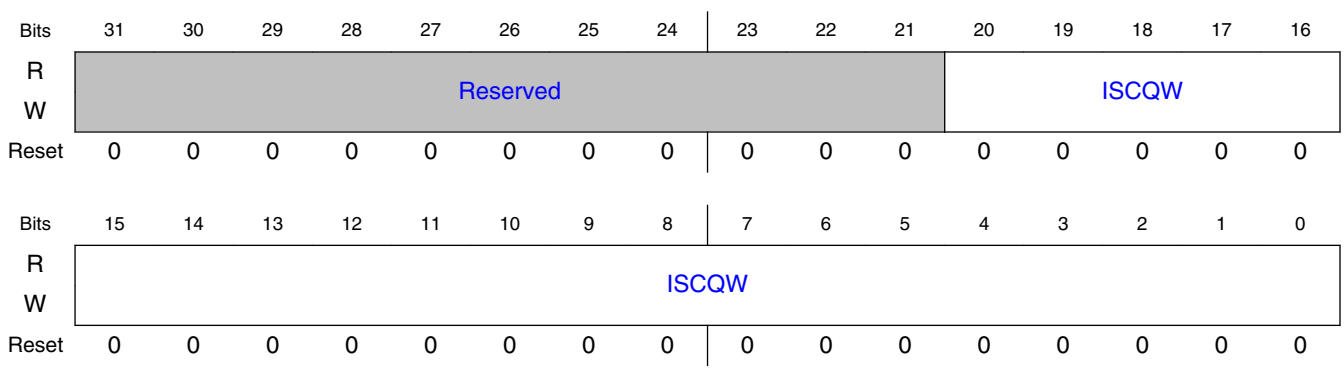
11.7.6.1.408 Queue 1 idleSlopeCredit, Quantum or Weights (MTL_TXQ1_QUANTUM_WEIGHT)

The Queue 1 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted in Queue 1.

11.7.6.1.408.1 Offset

Register	Offset
MTL_TXQ1_QUANTUM_WEIGHT	D58h

11.7.6.1.408.2 Diagram



11.7.6.1.408.3 Fields

Field	Description
31-21	Reserved

Table continues on the next page...

Field	Description
—	
20-0 ISCQW	idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps; 3.2 ns for 2500 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. Bits[20:14] must be written to zero. - Quantum When the DCB operation is enabled with DWRR algorithm for Queue 1 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. - Weights When DCB operation is enabled with WFQ algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero. - Note 1: In multiple Queue configuration this field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is MTL_OPERATION_MODE register. - Note 2: For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula: (previous_finish_time of particular Transmit Queue + (weights*packet_size)) - Note 3: The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.

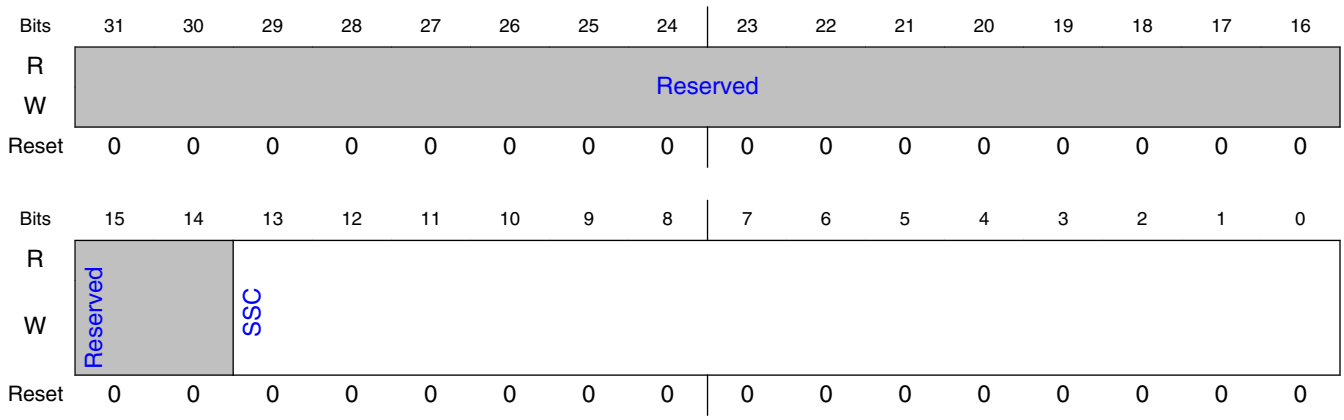
11.7.6.1.409 Queue 1 sendSlopeCredit (MTL_TXQ1_SENDSLOPECREDIT)

The sendSlopeCredit register contains the sendSlope credit value required for the credit-based shaper algorithm for the Queue.

11.7.6.1.409.1 Offset

Register	Offset
MTL_TXQ1_SENDSLOPECREDIT	D5Ch

11.7.6.1.409.2 Diagram



11.7.6.1.409.3 Fields

Field	Description
31-14 —	Reserved
13-0 SSC	sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 1. sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns, 8 ns and 3.2 ns for 100 Mbps, 1000 Mbps and 2500 Mbps respectively) when the credit is decreasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. This field should be programmed with absolute sendSlopeCredit value. The credit-based shaper logic subtracts it from the accumulated credit when Channel 1 is selected for transmission.

11.7.6.1.410 Queue 1 hiCredit (MTL_TXQ1_HICREDIT)

The hiCredit register contains the hiCredit value required for the credit-based shaper algorithm for the Queue.

11.7.6.1.410.1 Offset

Register	Offset
MTL_TXQ1_HICREDIT	D60h

11.7.6.1.410.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			HC												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.410.3 Fields

Field	Description
31-29 —	Reserved
28-0 HC	hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. This is the maximum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value is maxInterferenceSize, that is, best-effort maximum packet size (16,384 bytes or 131,072 bits). The value to be specified is 131,072 * 1,024 = 134,217,728 or 0x0800_0000.

11.7.6.1.411 Queue 1 loCredit (MTL_TXQ1_LOCREDIT)

The loCredit register contains the loCredit value required for the credit-based shaper algorithm for the Queue.

11.7.6.1.411.1 Offset

Register	Offset
MTL_TXQ1_LOCREDIT	D64h

11.7.6.1.411.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			LC												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.411.3 Fields

Field	Description
31-29 —	Reserved
28-0 LC	loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. This is the minimum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value to be programmed is corresponds to twice the maxFrameSize transmitted from this queue. If the maxFrameSize is 8192 bytes, then (8192*2) * 8 * 1024 = 134,217,728 or 0x0800_0000. Because it is a negative value, the programmed value is 2's complement of the value, that is, 0x1800_0000.

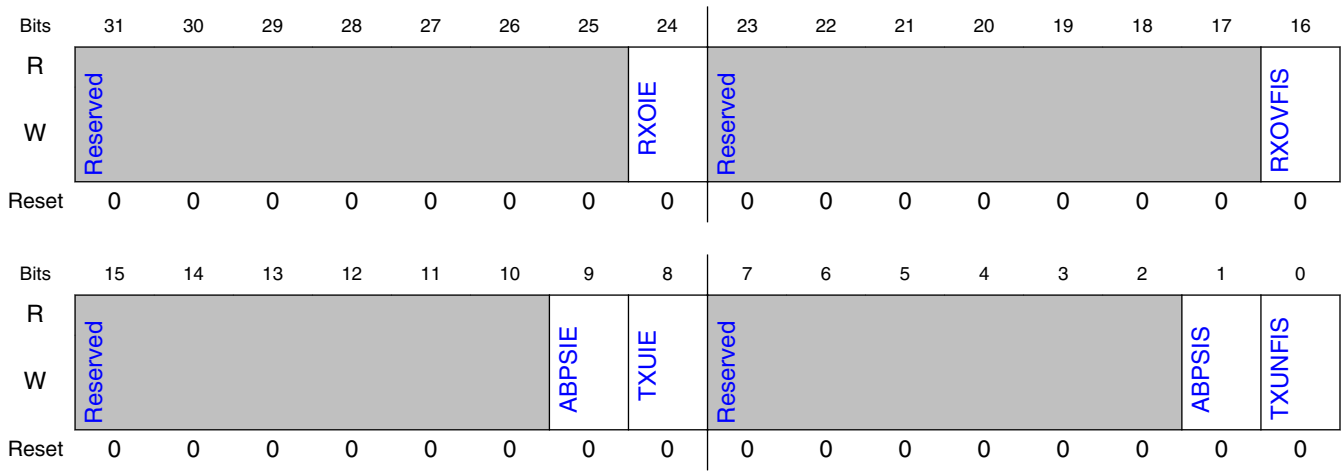
11.7.6.1.412 Queue 1 Interrupt Control Status (MTL_Q1_INTERRUPT_CONTROL_STATUS)

This register contains the interrupt enable and status bits for the queue 1 interrupts.

11.7.6.1.412.1 Offset

Register	Offset
MTL_Q1_INTERRUPT_CONTROL_STATUS	D6Ch

11.7.6.1.412.2 Diagram



11.7.6.1.412.3 Fields

Field	Description
31-25 —	Reserved
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0 - Receive Queue Overflow Interrupt is disabled 1 - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Queue Overflow Interrupt Status not detected 1 - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9 ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. Average Bits Per Slot Interrupt Enable

Table continues on the next page...

Field	Description
	When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0 - Average Bits Per Slot Interrupt is disabled 1 - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0 - Transmit Queue Underflow Interrupt Status is disabled 1 - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Average Bits Per Slot Interrupt Status not detected 1 - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Queue Underflow Interrupt Status not detected 1 - Transmit Queue Underflow Interrupt Status detected

11.7.6.1.413 Queue 1 Receive Operation Mode (MTL_RXQ1_OPERATION_MODE)

The Queue 1 Receive Operation Mode register establishes the Receive queue operating modes and command.

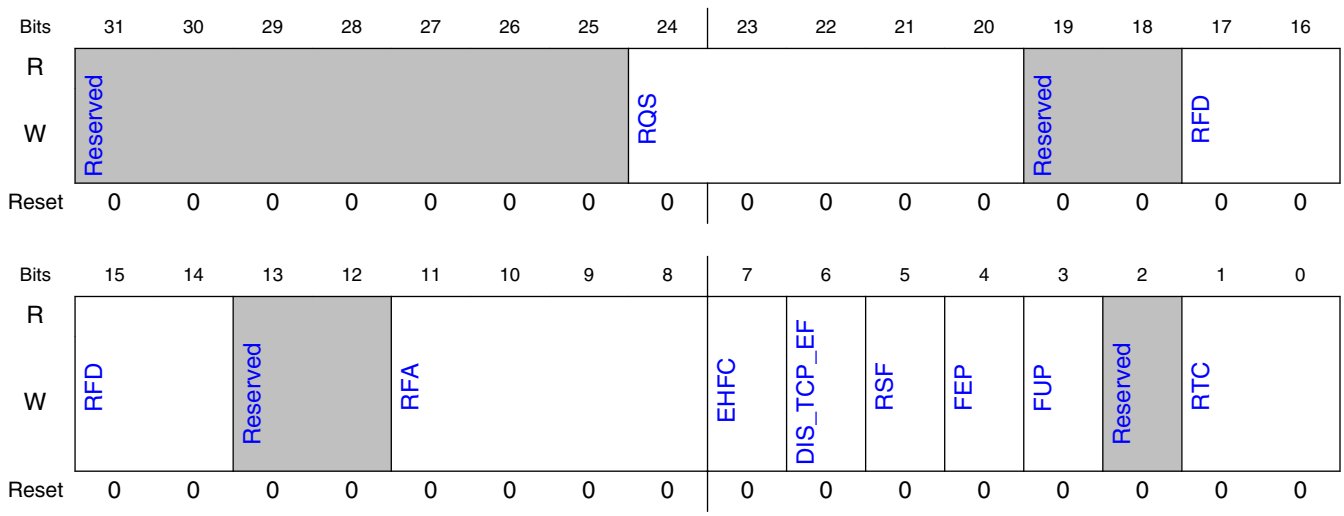
The Queue 1 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

11.7.6.1.413.1 Offset

Register	Offset
MTL_RXQ1_OPERATION_MODE	D70h

11.7.6.1.413.2 Diagram



11.7.6.1.413.3 Fields

Field	Description
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
19-18 —	Reserved

Table continues on the next page...

Field	Description
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0 - Hardware Flow Control is disabled 1 - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0 - Dropping of TCP/IP Checksum Error Packets is enabled 1 - Dropping of TCP/IP Checksum Error Packets is disabled

Table continues on the next page...

Field	Description
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0 - Receive Queue Store and Forward is disabled 1 - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0 - Forward Error Packets is disabled 1 - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0 - Forward Undersized Good Packets is disabled 1 - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00 - 64 01 - 32 10 - 96 11 - 128

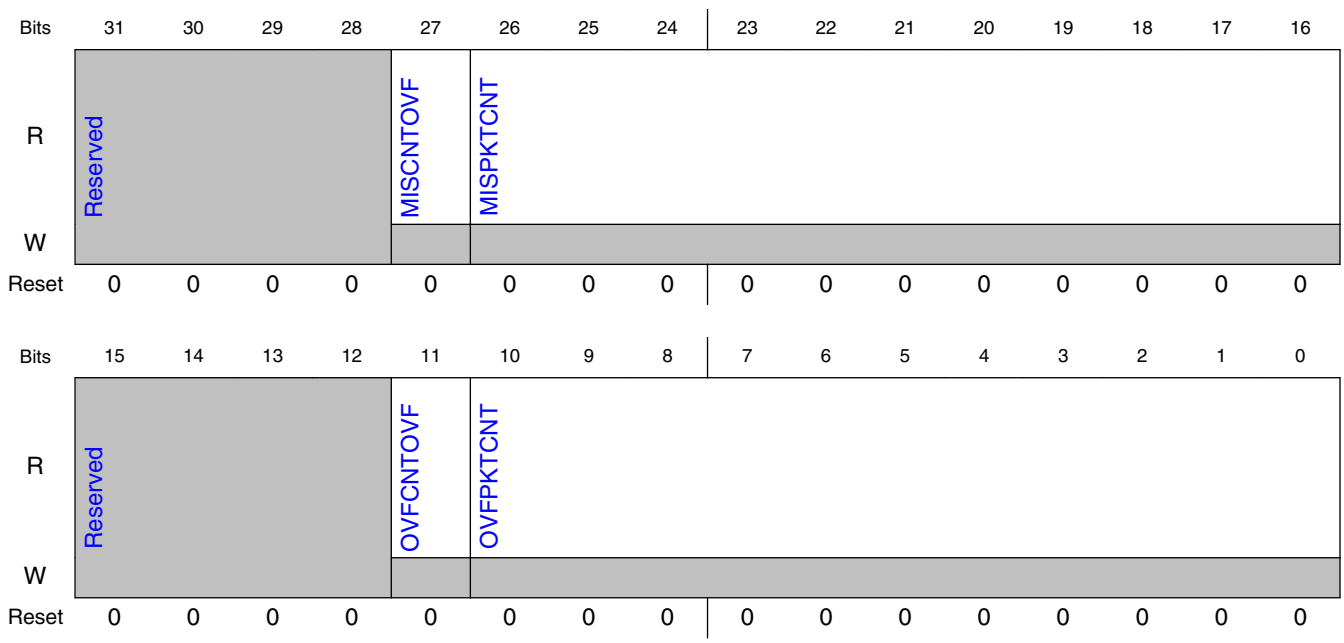
11.7.6.1.414 Queue 1 Missed Packet and Overflow Counter (MTL_RXQ1_MISSED_PACKET_OVERFLOW_CNT)

The Queue 1 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

11.7.6.1.414.1 Offset

Register	Offset
MTL_RXQ1_MISSED_PACKET_OVERFLOW_CNT	D74h

11.7.6.1.414.2 Diagram



11.7.6.1.414.3 Fields

Field	Description
31-28 —	Reserved

Table continues on the next page...

Field	Description
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Missed Packet Counter overflow not detected 1 - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow Counter overflow not detected 1 - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

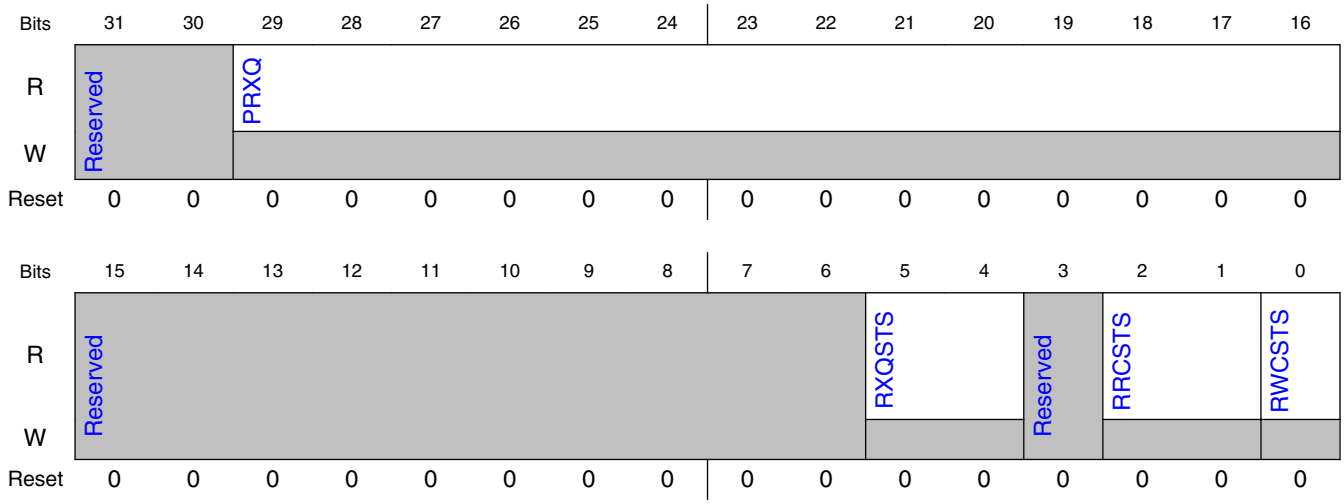
11.7.6.1.415 Queue 1 Receive Debug (MTL_RXQ1_DEBUG)

The Queue 1 Receive Debug register gives the debug status of various blocks related to the Receive queue.

11.7.6.1.415.1 Offset

Register	Offset
MTL_RXQ1_DEBUG	D78h

11.7.6.1.415.2 Diagram



11.7.6.1.415.3 Fields

Field	Description
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00 - Rx Queue empty 01 - Rx Queue fill-level below flow-control deactivate threshold 10 - Rx Queue fill-level above flow-control activate threshold 11 - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00 - Idle state 01 - Reading packet data 10 - Reading packet status (or timestamp) 11 - Flushing the packet data and status

Table continues on the next page...

Field	Description
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0 - MTL Rx Queue Write Controller Active Status not detected 1 - MTL Rx Queue Write Controller Active Status detected

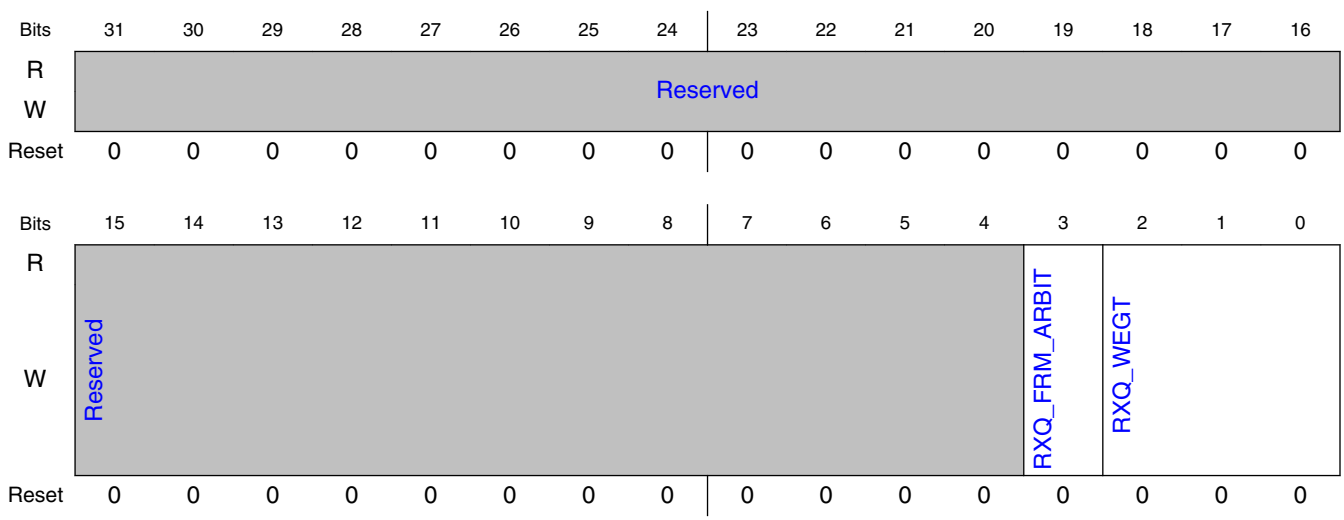
11.7.6.1.416 Queue 1 Receive Control (MTL_RXQ1_CONTROL)

The Queue 1 Receive Control register controls the receive arbitration and passing of received packets to the application.

11.7.6.1.416.1 Offset

Register	Offset
MTL_RXQ1_CONTROL	D7Ch

11.7.6.1.416.2 Diagram



11.7.6.1.416.3 Fields

Field	Description
31-4 —	Reserved

Table continues on the next page...

Field	Description
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0 - Receive Queue Packet Arbitration is disabled 1 - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

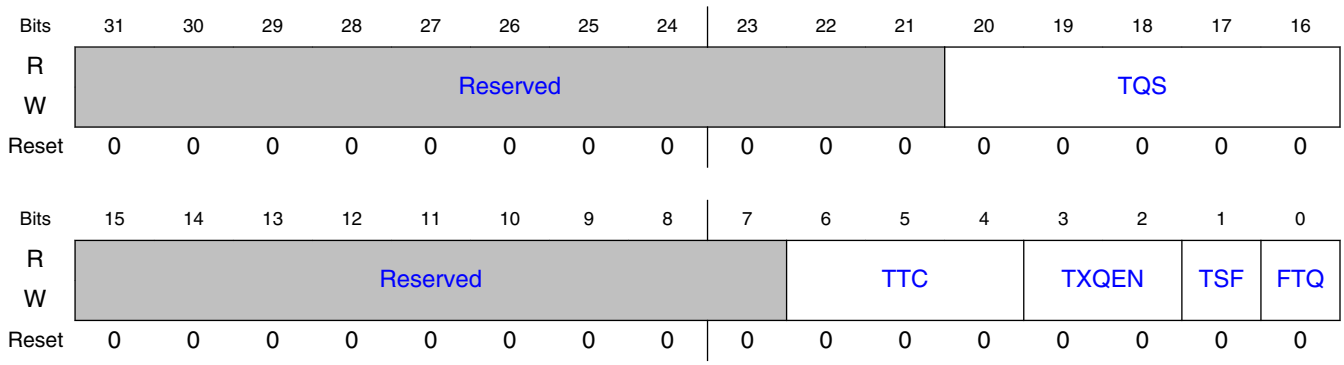
11.7.6.1.417 Queue 2 Transmit Operation Mode (MTL_TXQ2_OPERATION_MODE)

The Queue 2 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

11.7.6.1.417.1 Offset

Register	Offset
MTL_TXQ2_OPERATION_MODE	D80h

11.7.6.1.417.2 Diagram



11.7.6.1.417.3 Fields

Field	Description
31-21 —	Reserved
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000 - 32 001 - 64 010 - 96 011 - 128 100 - 192 101 - 256 110 - 384 111 - 512

Table continues on the next page...

Field	Description
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Enable in AV mode - 2'b10: Enabled - 2'b11: Reserved Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00 - Not enabled 01 - Enable in AV mode (Reserved in non-AV) 10 - Enabled 11 - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0 - Transmit Store and Forward is disabled 1 - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ1_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flush Transmit Queue is disabled 1 - Flush Transmit Queue is enabled

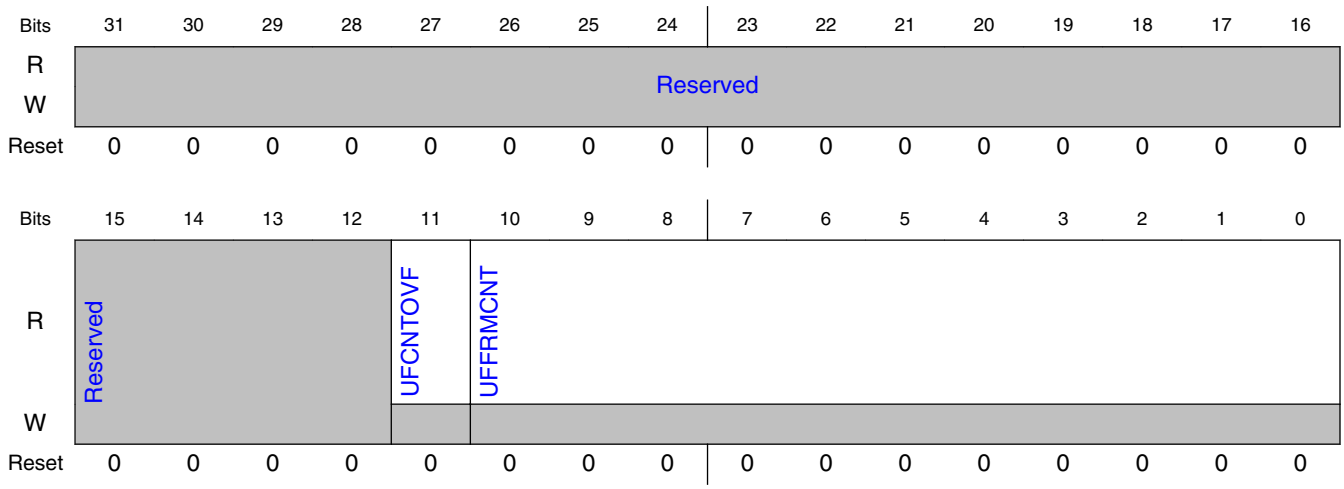
11.7.6.1.418 Queue 2 Underflow Counter (MTL_TXQ2_UNDERFLOW)

The Queue 2 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

11.7.6.1.418.1 Offset

Register	Offset
MTL_TXQ2_UNDERFLOW	D84h

11.7.6.1.418.2 Diagram



11.7.6.1.418.3 Fields

Field	Description
31-12 —	Reserved
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow not detected for Underflow Packet Counter 1 - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

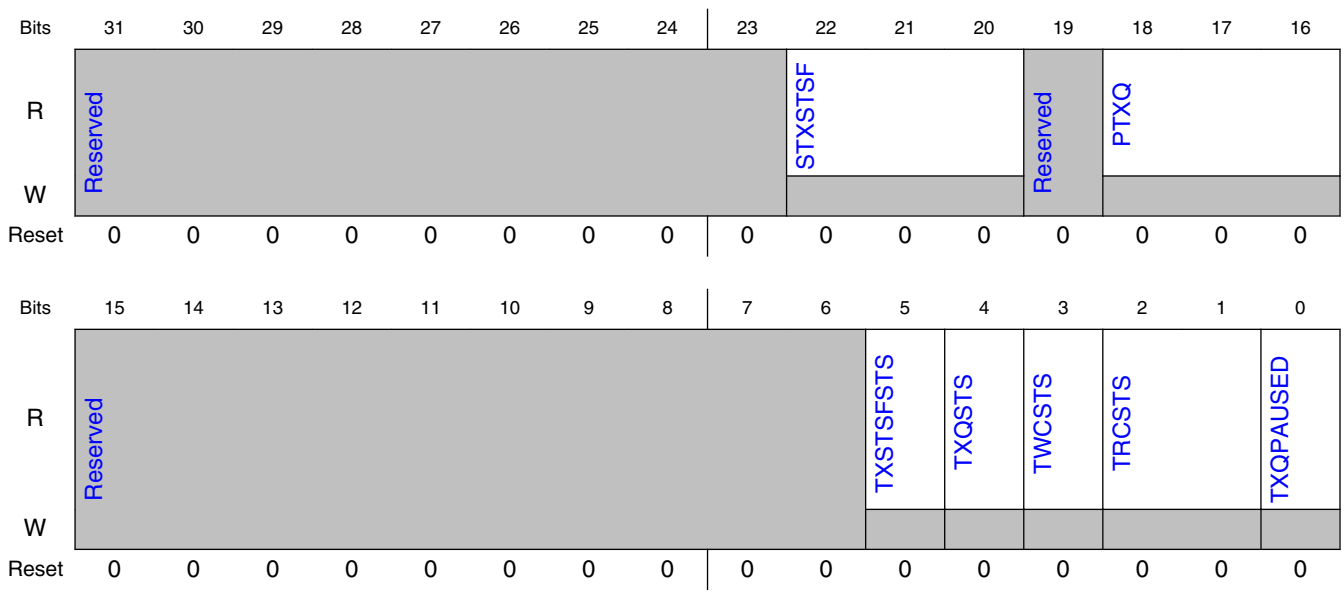
11.7.6.1.419 Queue 2 Transmit Debug (MTL_TXQ2_DEBUG)

The Queue 2 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

11.7.6.1.419.1 Offset

Register	Offset
MTL_TXQ2_DEBUG	D88h

11.7.6.1.419.2 Diagram



11.7.6.1.419.3 Fields

Field	Description
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.

Table continues on the next page...

Field	Description
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0 - MTL Tx Status FIFO Full status is not detected 1 - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0 - MTL Tx Queue Not Empty status is not detected 1 - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0 - MTL Tx Queue Write Controller status is not detected 1 - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00 - Idle state 01 - Read state (transferring data to the MAC transmitter) 10 - Waiting for pending Tx Status from the MAC transmitter 11 - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0 - Transmit Queue in Pause status is not detected 1 - Transmit Queue in Pause status is detected

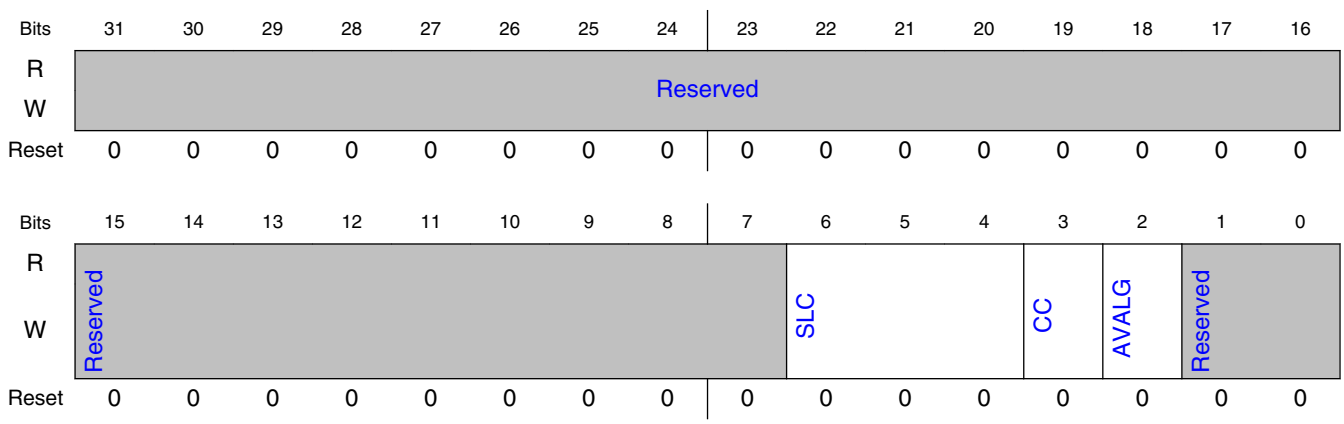
11.7.6.1.420 Queue 2 ETS Control (MTL_TXQ2_ETS_CONTROL)

The Queue 2 ETS Control register controls the enhanced transmission selection operation.

11.7.6.1.420.1 Offset

Register	Offset
MTL_TXQ2_ETS_CONTROL	D90h

11.7.6.1.420.2 Diagram



11.7.6.1.420.3 Fields

Field	Description
31-7 —	Reserved
6-4 SLC	Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH[N]_SLOT_INTERVAL register) over which the average transmitted bits per slot, provided in the MTL_TXQ[N]_ETS_STATUS register, need to be computed for Queue. Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH[N]_SLOT_INTERVAL register) over which the average transmitted bits per slot, provided in the MTL_TXQ[N]_ETS_STATUS register, need to be computed for Queue. The encoding is as follows: 000 - 1 slot 001 - 2 slots 010 - 4 slots 011 - 8 slots

Table continues on the next page...

Field	Description
	100 - 16 slots 101 - Reserved
3 CC	Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 1. Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 1. The credit accumulates even when there is no packet waiting in Channel 1 and another channel is transmitting. When this bit is reset, the accumulated credit parameter in the credit-based shaper algorithm logic is set to zero when there is positive credit and no packet to transmit in Channel 1. When there is no packet waiting in Channel 1 and other channel is transmitting, no credit is accumulated. 0 - Credit Control is disabled 1 - Credit Control is enabled
2 AVALG	AV Algorithm When Queue 1 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 1 traffic. AV Algorithm When Queue 1 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 1 traffic. When reset, strict priority is selected. 0 - CBS Algorithm is disabled 1 - CBS Algorithm is enabled
1-0 —	Reserved

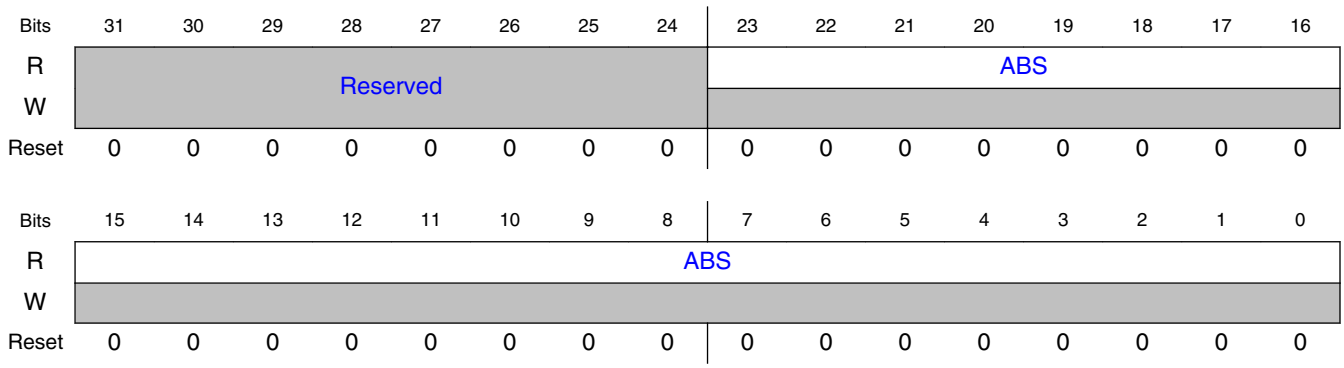
11.7.6.1.421 Queue 2 ETS Status (MTL_TXQ2_ETS_STATUS)

The Queue 2 ETS Status register provides the average traffic transmitted in Queue 2.

11.7.6.1.421.1 Offset

Register	Offset
MTL_TXQ2_ETS_STATUS	D94h

11.7.6.1.421.2 Diagram



11.7.6.1.421.3 Fields

Field	Description
31-24 —	Reserved
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TXQ[n]_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps and 9C_4000 in 2500 Mbps mode respectively. When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

11.7.6.1.422 Queue 2 idleSlopeCredit, Quantum or Weights (MTL_TXQ2_QUANTUM_WEIGHT)

The Queue 2 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted in Queue 2.

11.7.6.1.422.1 Offset

Register	Offset
MTL_TXQ2_QUANTUM_WEIGHT	D98h

11.7.6.1.422.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											ISCQW				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ISCQW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.422.3 Fields

Field	Description
31-21 —	Reserved
20-0 ISCQW	idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps; 3.2 ns for 2500 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. Bits[20:14] must be written to zero. - Quantum When the DCB operation is enabled with DWRR algorithm for Queue 1 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. - Weights When DCB operation is enabled with WFQ algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero. - Note 1: In multiple Queue configuration this field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is MTL_OPERATION_MODE register. - Note 2: For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula: (previous_finish_time of particular Transmit Queue + (weights*packet_size))

Field	Description
	- Note 3: The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.

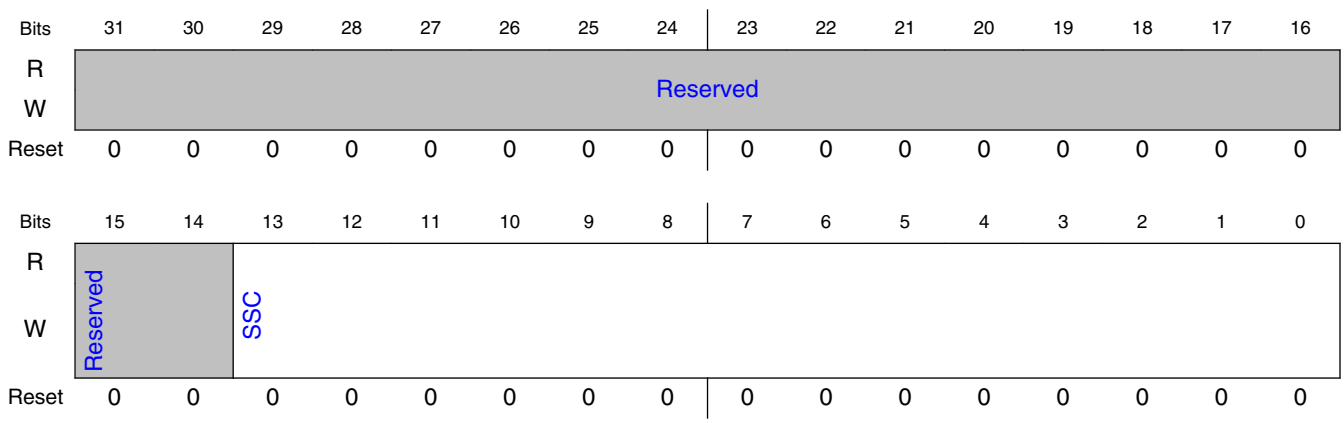
11.7.6.1.423 Queue 2 sendSlopeCredit (MTL_TXQ2_SENDSLOPECREDIT)

The sendSlopeCredit register contains the sendSlope credit value required for the credit-based shaper algorithm for the Queue.

11.7.6.1.423.1 Offset

Register	Offset
MTL_TXQ2_SENDSLOPECREDIT	D9Ch

11.7.6.1.423.2 Diagram



11.7.6.1.423.3 Fields

Field	Description
31-14 —	Reserved
13-0 SSC	sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 1. sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns, 8 ns and 3.2 ns for 100 Mbps, 1000 Mbps and 2500 Mbps respectively) when the credit is decreasing. The software

Field	Description
	should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. This field should be programmed with absolute sendSlopeCredit value. The credit-based shaper logic subtracts it from the accumulated credit when Channel 1 is selected for transmission.

11.7.6.1.424 Queue 2 hiCredit (MTL_TXQ2_HICREDIT)

The hiCredit register contains the hiCredit value required for the credit-based shaper algorithm for the Queue.

11.7.6.1.424.1 Offset

Register	Offset
MTL_TXQ2_HICREDIT	DA0h

11.7.6.1.424.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			HC												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.424.3 Fields

Field	Description
31-29 —	Reserved
28-0 HC	hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. This is the maximum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024.

Field	Description
	The maximum value is maxInterferenceSize, that is, best-effort maximum packet size (16,384 bytes or 131,072 bits). The value to be specified is 131,072 * 1,024 = 134,217,728 or 0x0800_0000.

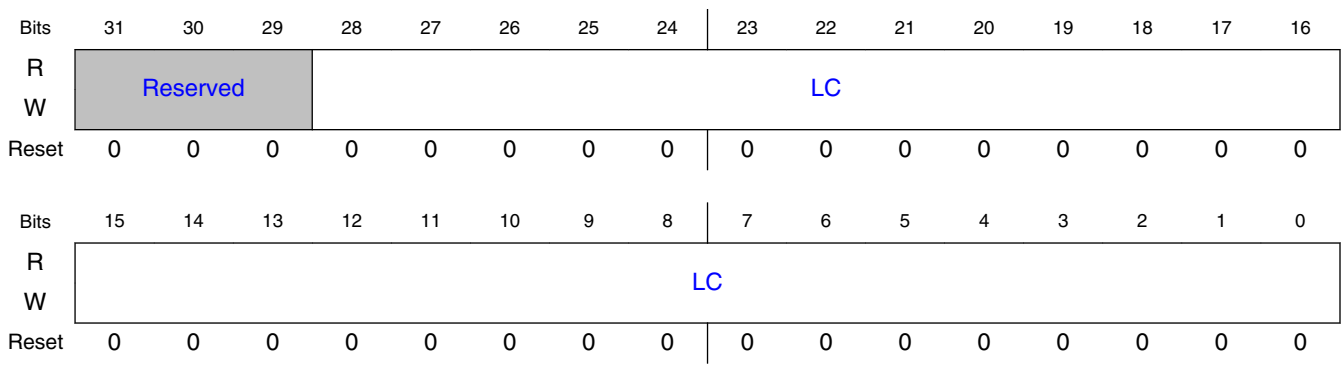
11.7.6.1.425 Queue 2 loCredit (MTL_TXQ2_LOCREDIT)

The loCredit register contains the loCredit value required for the credit-based shaper algorithm for the Queue.

11.7.6.1.425.1 Offset

Register	Offset
MTL_TXQ2_LOCREDIT	DA4h

11.7.6.1.425.2 Diagram



11.7.6.1.425.3 Fields

Field	Description
31-29 —	Reserved
28-0 LC	loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. This is the minimum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value to be programmed is corresponds to twice the maxFrameSize transmitted from this queue. If the maxFrameSize is 8192 bytes, then (8192*2) * 8 * 1024 = 134,217,728 or 0x0800_0000. Because it is a negative value, the programmed value is 2's complement of the value, that is, 0x1800_0000.

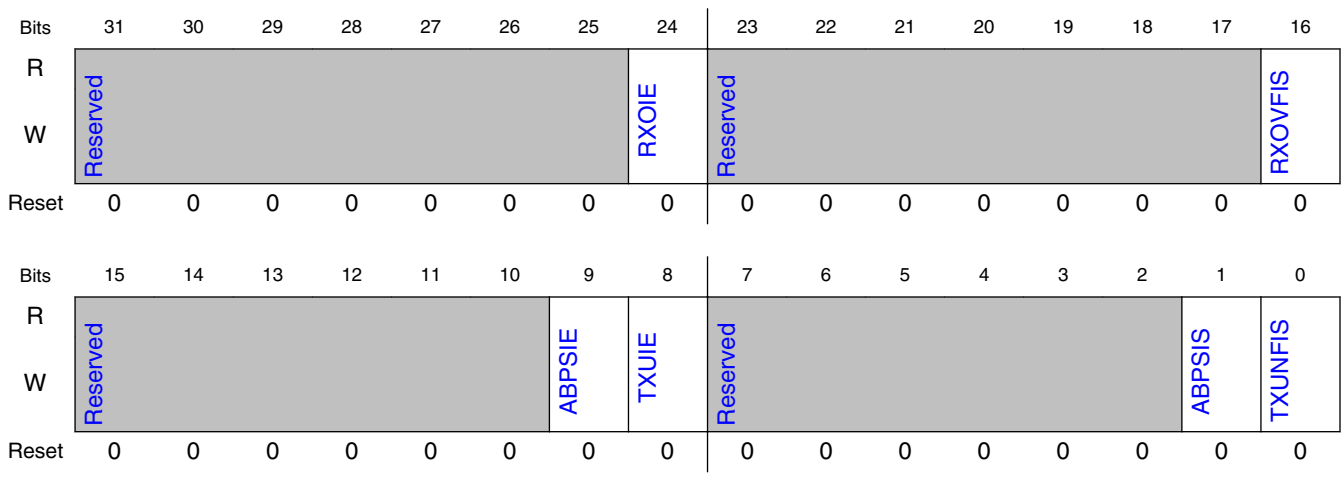
11.7.6.1.426 Queue 2 Interrupt Control Status (MTL_Q2_INTERRUPT_CONTROL_STATUS)

This register contains the interrupt enable and status bits for the queue 2 interrupts.

11.7.6.1.426.1 Offset

Register	Offset
MTL_Q2_INTERRUPT_CONTROL_STATUS	DACh

11.7.6.1.426.2 Diagram



11.7.6.1.426.3 Fields

Field	Description
31-25 —	Reserved
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0 - Receive Queue Overflow Interrupt is disabled 1 - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved

Table continues on the next page...

Field	Description
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Queue Overflow Interrupt Status not detected 1 - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9 ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0 - Average Bits Per Slot Interrupt is disabled 1 - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0 - Transmit Queue Underflow Interrupt Status is disabled 1 - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Average Bits Per Slot Interrupt Status not detected 1 - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

Field	Description
	0 - Transmit Queue Underflow Interrupt Status not detected 1 - Transmit Queue Underflow Interrupt Status detected

11.7.6.1.427 Queue 2 Receive Operation Mode (MTL_RXQ2_OPERATION_MODE)

The Queue 2 Receive Operation Mode register establishes the Receive queue operating modes and command.

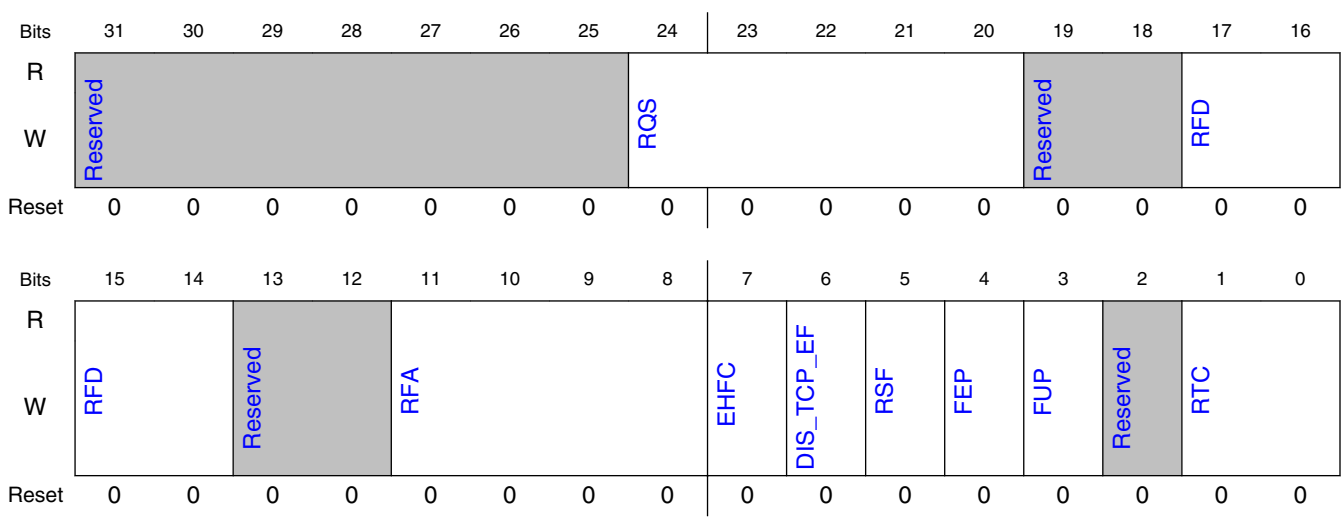
The Queue 2 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

11.7.6.1.427.1 Offset

Register	Offset
MTL_RXQ2_OPERATION_MODE	DB0h

11.7.6.1.427.2 Diagram



11.7.6.1.427.3 Fields

Field	Description
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
19-18 —	Reserved
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.

Table continues on the next page...

Field	Description
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0 - Hardware Flow Control is disabled 1 - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0 - Dropping of TCP/IP Checksum Error Packets is enabled 1 - Dropping of TCP/IP Checksum Error Packets is disabled
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0 - Receive Queue Store and Forward is disabled 1 - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0 - Forward Error Packets is disabled 1 - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0 - Forward Undersized Good Packets is disabled

Table continues on the next page...

Field	Description
	1 - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00 - 64 01 - 32 10 - 96 11 - 128

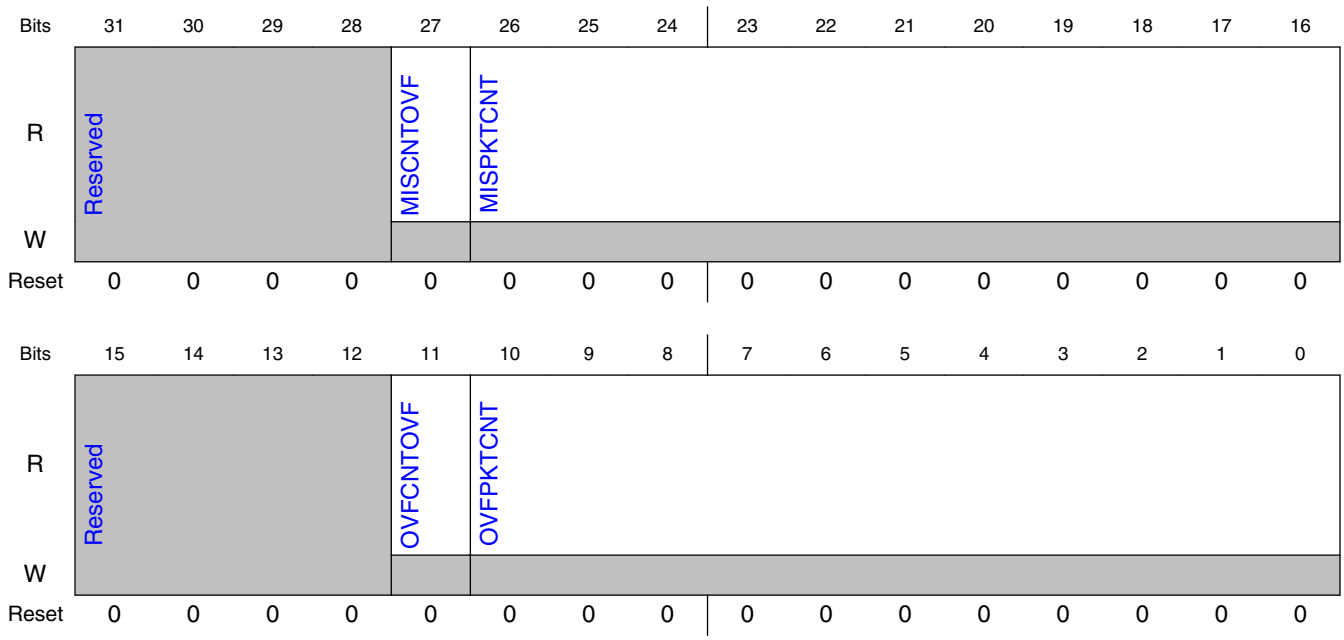
11.7.6.1.428 Queue 2 Missed Packet and Overflow Counter (MTL_RXQ2_MISSED_PACKET_OVERFLOW_CNT)

The Queue 2 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

11.7.6.1.428.1 Offset

Register	Offset
MTL_RXQ2_MISSED_PACKET_OVERFLOW_CNT	DB4h

11.7.6.1.428.2 Diagram



11.7.6.1.428.3 Fields

Field	Description
31-28 —	Reserved
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Missed Packet Counter overflow not detected 1 - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved

Table continues on the next page...

Field	Description
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow Counter overflow not detected 1 - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

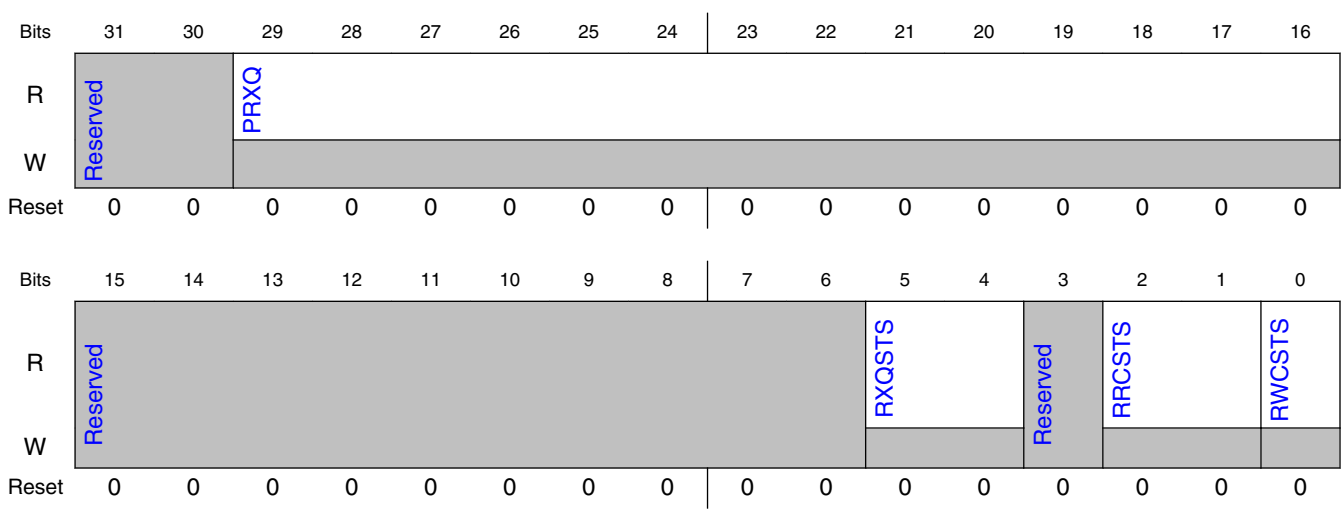
11.7.6.1.429 Queue 2 Receive Debug (MTL_RXQ2_DEBUG)

The Queue 2 Receive Debug register gives the debug status of various blocks related to the Receive queue.

11.7.6.1.429.1 Offset

Register	Offset
MTL_RXQ2_DEBUG	DB8h

11.7.6.1.429.2 Diagram



11.7.6.1.429.3 Fields

Field	Description
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00 - Rx Queue empty 01 - Rx Queue fill-level below flow-control deactivate threshold 10 - Rx Queue fill-level above flow-control activate threshold 11 - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00 - Idle state 01 - Reading packet data 10 - Reading packet status (or timestamp) 11 - Flushing the packet data and status
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0 - MTL Rx Queue Write Controller Active Status not detected 1 - MTL Rx Queue Write Controller Active Status detected

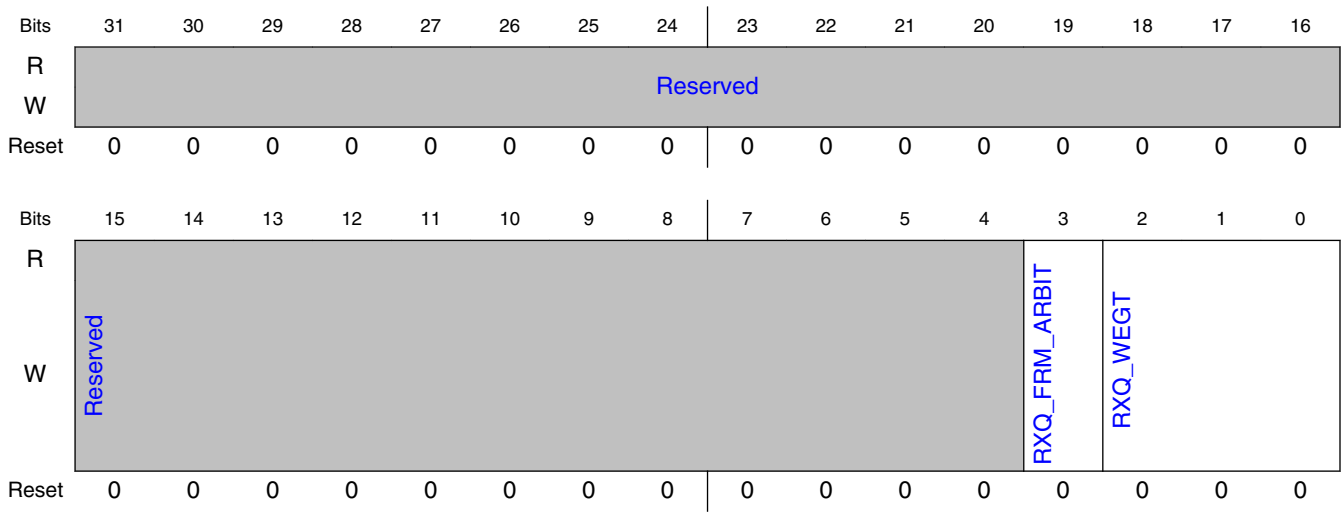
11.7.6.1.430 Queue 2 Receive Control (MTL_RXQ2_CONTROL)

The Queue 2 Receive Control register controls the receive arbitration and passing of received packets to the application.

11.7.6.1.430.1 Offset

Register	Offset
MTL_RXQ2_CONTROL	DBCh

11.7.6.1.430.2 Diagram



11.7.6.1.430.3 Fields

Field	Description
31-4 —	Reserved
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0 - Receive Queue Packet Arbitration is disabled 1 - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

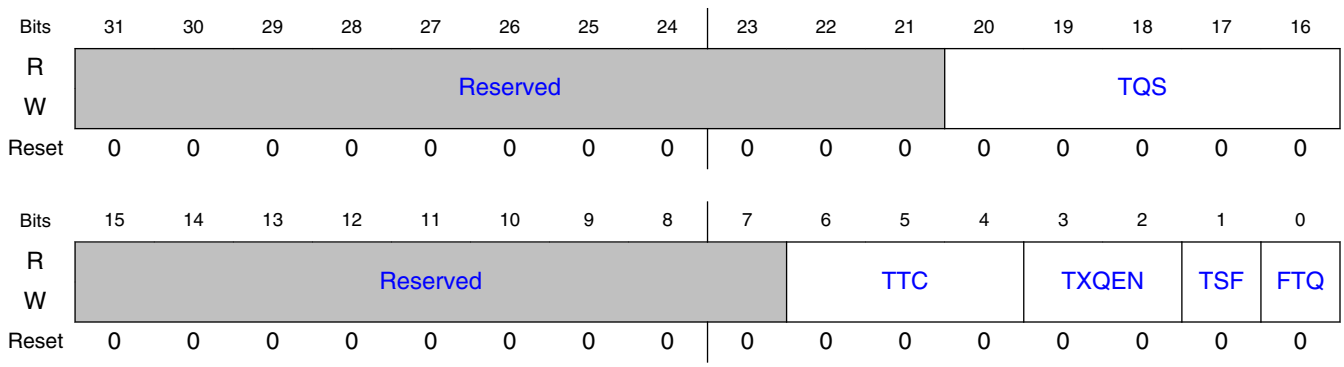
11.7.6.1.431 Queue 3 Transmit Operation Mode (MTL_TXQ3_OPERATION_MODE)

The Queue 3 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

11.7.6.1.431.1 Offset

Register	Offset
MTL_TXQ3_OPERATION_MODE	DC0h

11.7.6.1.431.2 Diagram



11.7.6.1.431.3 Fields

Field	Description
31-21 —	Reserved
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved

Table continues on the next page...

Field	Description
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000 - 32 001 - 64 010 - 96 011 - 128 100 - 192 101 - 256 110 - 384 111 - 512
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Enable in AV mode - 2'b10: Enabled - 2'b11: Reserved Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00 - Not enabled 01 - Enable in AV mode (Reserved in non-AV) 10 - Enabled 11 - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0 - Transmit Store and Forward is disabled 1 - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ1_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission.

Field	Description
	Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flush Transmit Queue is disabled 1 - Flush Transmit Queue is enabled

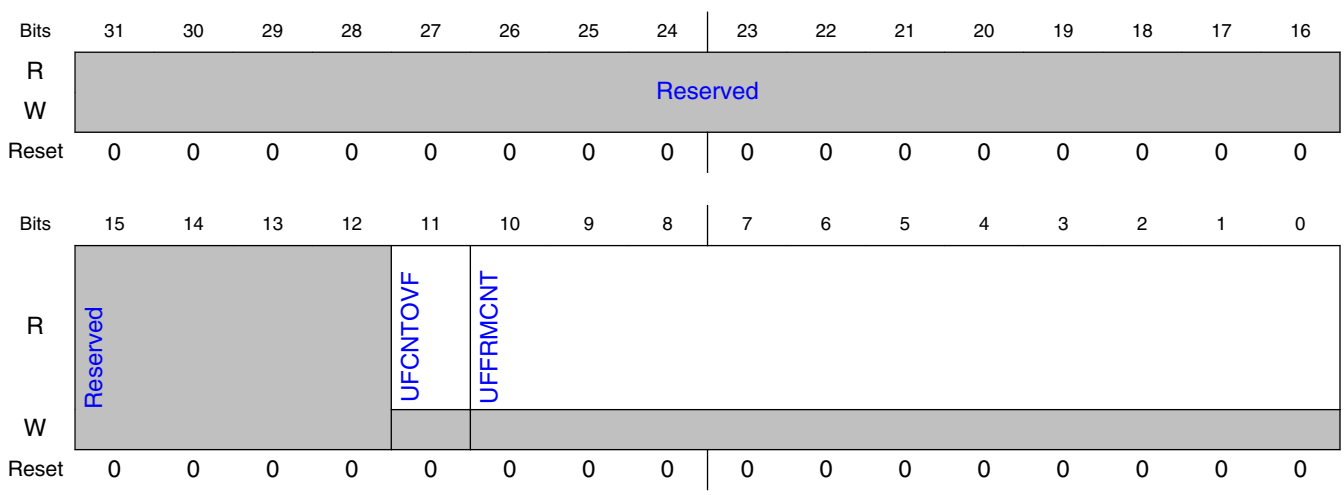
11.7.6.1.432 Queue 3 Underflow Counter (MTL_TXQ3_UNDERFLOW)

The Queue 3 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

11.7.6.1.432.1 Offset

Register	Offset
MTL_TXQ3_UNDERFLOW	DC4h

11.7.6.1.432.2 Diagram



11.7.6.1.432.3 Fields

Field	Description
31-12	Reserved

Table continues on the next page...

Field	Description
—	
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow not detected for Underflow Packet Counter 1 - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

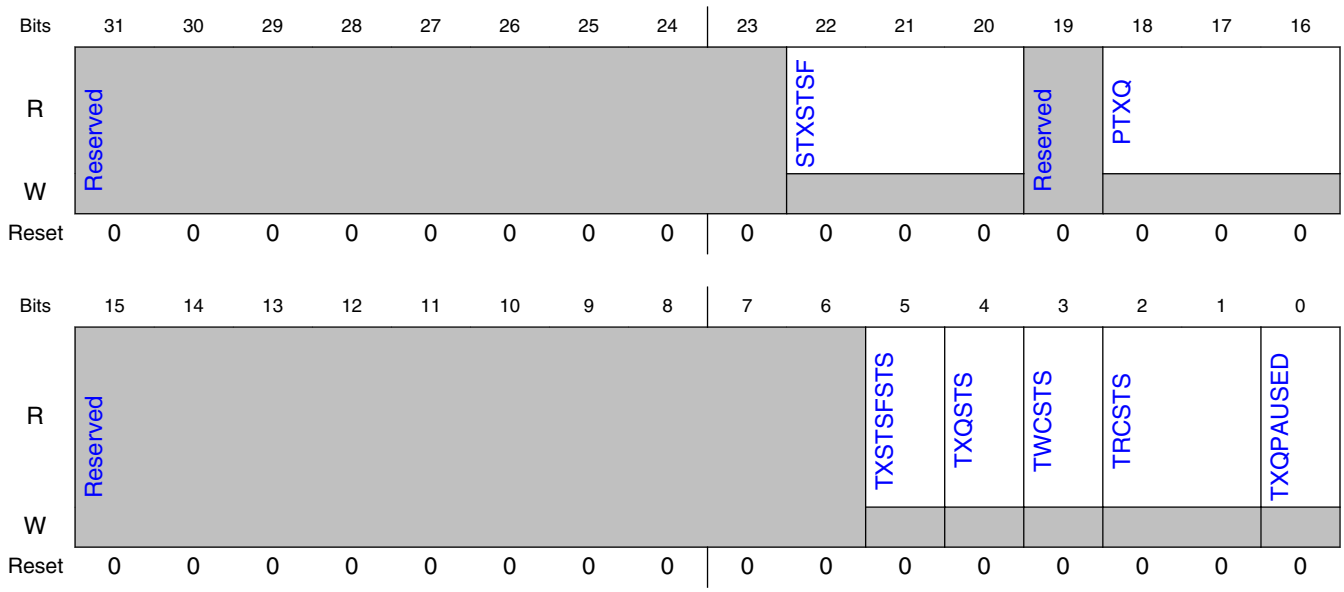
11.7.6.1.433 Queue 3 Transmit Debug (MTL_TXQ3_DEBUG)

The Queue 3 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

11.7.6.1.433.1 Offset

Register	Offset
MTL_TXQ3_DEBUG	DC8h

11.7.6.1.433.2 Diagram



11.7.6.1.433.3 Fields

Field	Description
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission.

Table continues on the next page...

Field	Description
	0 - MTL Tx Status FIFO Full status is not detected 1 - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0 - MTL Tx Queue Not Empty status is not detected 1 - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0 - MTL Tx Queue Write Controller status is not detected 1 - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00 - Idle state 01 - Read state (transferring data to the MAC transmitter) 10 - Waiting for pending Tx Status from the MAC transmitter 11 - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0 - Transmit Queue in Pause status is not detected 1 - Transmit Queue in Pause status is detected

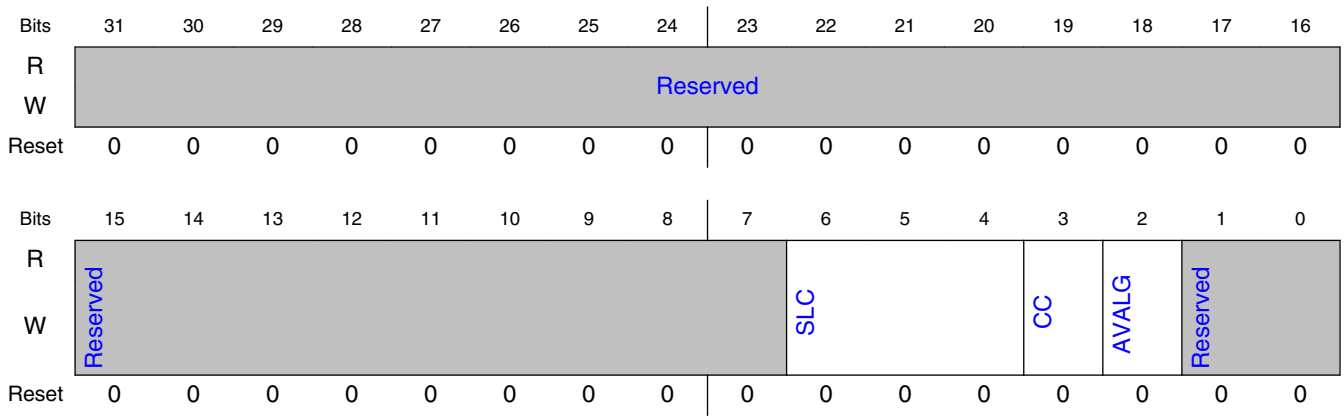
11.7.6.1.434 Queue 3 ETS Control (MTL_TXQ3_ETS_CONTROL)

The Queue 3 ETS Control register controls the enhanced transmission selection operation.

11.7.6.1.434.1 Offset

Register	Offset
MTL_TXQ3_ETS_CONTROL	DD0h

11.7.6.1.434.2 Diagram



11.7.6.1.434.3 Fields

Field	Description
31-7 —	Reserved
6-4 SLC	Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH[N]_SLOT_INTERVAL register) over which the average transmitted bits per slot, provided in the MTL_TXQ[N]_ETS_STATUS register, need to be computed for Queue. Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH[N]_SLOT_INTERVAL register) over which the average transmitted bits per slot, provided in the MTL_TXQ[N]_ETS_STATUS register, need to be computed for Queue. The encoding is as follows: 000 - 1 slot 001 - 2 slots 010 - 4 slots 011 - 8 slots 100 - 16 slots 101 - Reserved
3 CC	Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 1. Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 1. The credit accumulates even when there is no packet waiting in Channel 1 and another channel is transmitting. When this bit is reset, the accumulated credit parameter in the credit-based shaper algorithm logic is set to zero when there is positive credit and no packet to transmit in Channel 1. When there is no packet waiting in Channel 1 and other channel is transmitting, no credit is accumulated. 0 - Credit Control is disabled 1 - Credit Control is enabled

Table continues on the next page...

Field	Description
2 AVALG	AV Algorithm When Queue 1 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 1 traffic. AV Algorithm When Queue 1 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 1 traffic. When reset, strict priority is selected. 0 - CBS Algorithm is disabled 1 - CBS Algorithm is enabled
1-0 —	Reserved

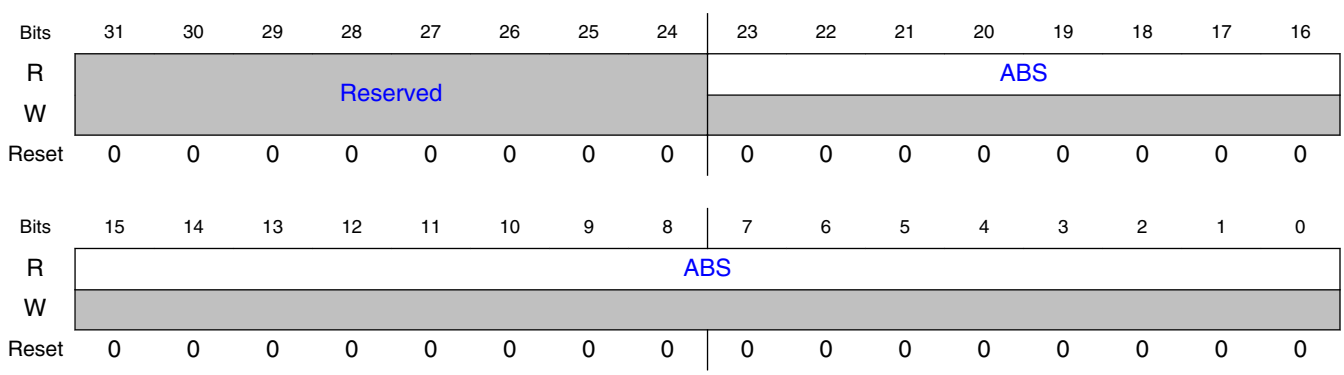
11.7.6.1.435 Queue 3 ETS Status (MTL_TXQ3_ETS_STATUS)

The Queue 3 ETS Status register provides the average traffic transmitted in Queue 3.

11.7.6.1.435.1 Offset

Register	Offset
MTL_TXQ3_ETS_STATUS	DD4h

11.7.6.1.435.2 Diagram



11.7.6.1.435.3 Fields

Field	Description
31-24	Reserved

Table continues on the next page...

Field	Description
—	
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TXQ[n]_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps and 9C_4000 in 2500 Mbps mode respectively. When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

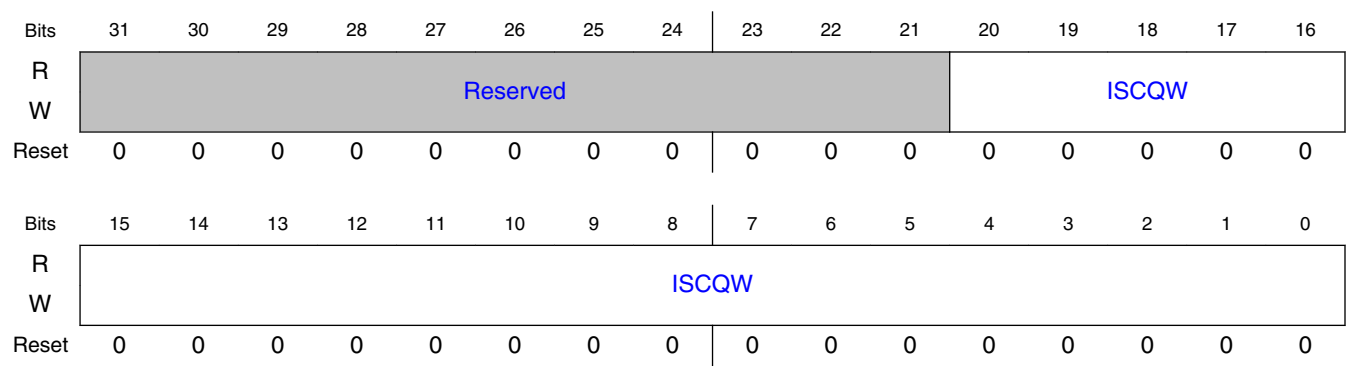
11.7.6.1.436 Queue 3 idleSlopeCredit, Quantum or Weights (MTL_TXQ3_QUANTUM_WEIGHT)

The Queue 3 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted in Queue 3.

11.7.6.1.436.1 Offset

Register	Offset
MTL_TXQ3_QUANTUM_WEIGHT	DD8h

11.7.6.1.436.2 Diagram



11.7.6.1.436.3 Fields

Field	Description
31-21	Reserved

Table continues on the next page...

Field	Description
—	
20-0 ISCQW	idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps; 3.2 ns for 2500 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. Bits[20:14] must be written to zero. - Quantum When the DCB operation is enabled with DWRR algorithm for Queue 1 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. - Weights When DCB operation is enabled with WFQ algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero. - Note 1: In multiple Queue configuration this field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is OPERATION_MODE register. - Note 2: For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula: (previous_finish_time of particular Transmit Queue + (weights*packet_size)) - Note 3: The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.

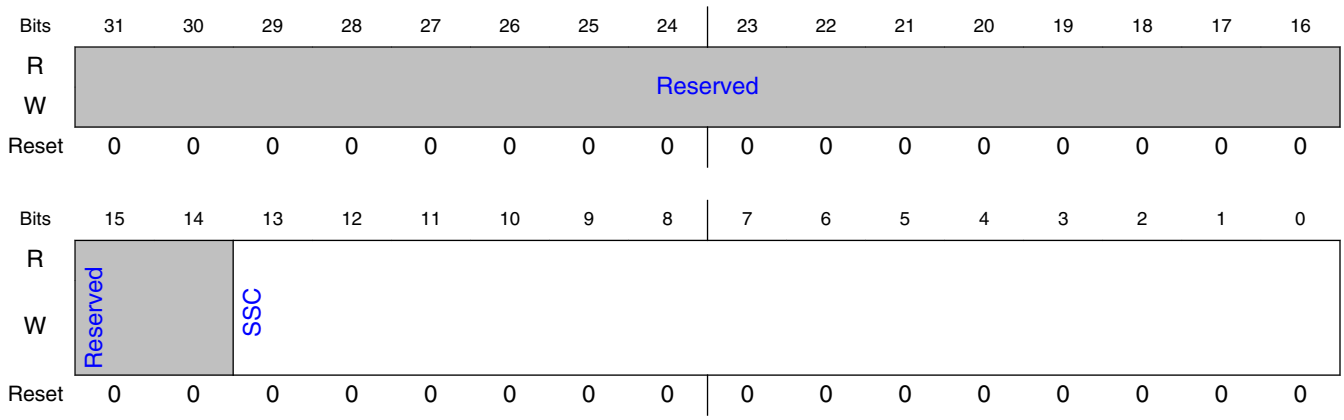
11.7.6.1.437 Queue 3 sendSlopeCredit (MTL_TXQ3_SENDSLOPECREDIT)

The sendSlopeCredit register contains the sendSlope credit value required for the credit-based shaper algorithm for the Queue.

11.7.6.1.437.1 Offset

Register	Offset
MTL_TXQ3_SENDSLOPECREDIT	DDCh

11.7.6.1.437.2 Diagram



11.7.6.1.437.3 Fields

Field	Description
31-14 —	Reserved
13-0 SSC	sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 1. sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns, 8 ns and 3.2 ns for 100 Mbps, 1000 Mbps and 2500 Mbps respectively) when the credit is decreasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. This field should be programmed with absolute sendSlopeCredit value. The credit-based shaper logic subtracts it from the accumulated credit when Channel 1 is selected for transmission.

11.7.6.1.438 Queue 3 hiCredit (MTL_TXQ3_HICREDIT)

The hiCredit register contains the hiCredit value required for the credit-based shaper algorithm for the Queue.

11.7.6.1.438.1 Offset

Register	Offset
MTL_TXQ3_HICREDIT	DE0h

11.7.6.1.438.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			HC												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.438.3 Fields

Field	Description
31-29 —	Reserved
28-0 HC	hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. This is the maximum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value is maxInterferenceSize, that is, best-effort maximum packet size (16,384 bytes or 131,072 bits). The value to be specified is 131,072 * 1,024 = 134,217,728 or 0x0800_0000.

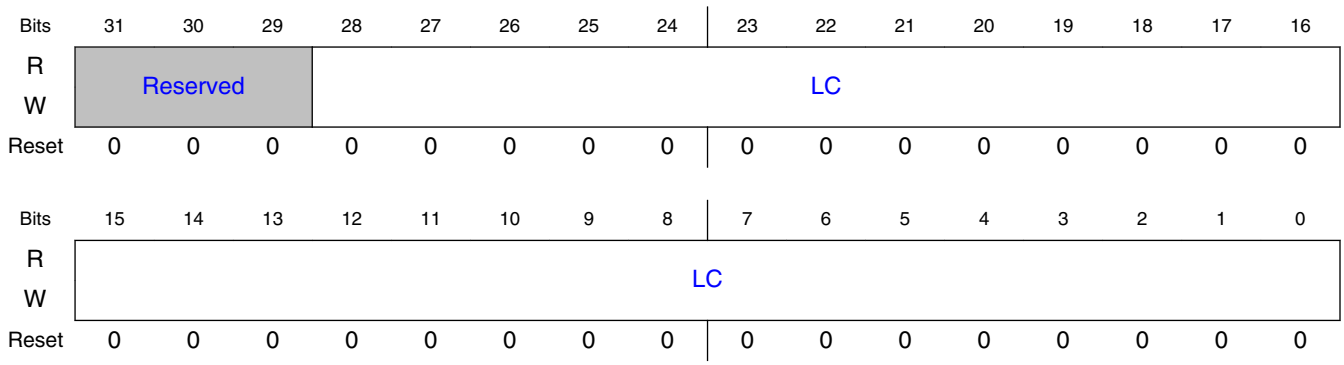
11.7.6.1.439 Queue 3 loCredit (MTL_TXQ3_LOCREDIT)

The loCredit register contains the loCredit value required for the credit-based shaper algorithm for the Queue.

11.7.6.1.439.1 Offset

Register	Offset
MTL_TXQ3_LOCREDIT	DE4h

11.7.6.1.439.2 Diagram



11.7.6.1.439.3 Fields

Field	Description
31-29 —	Reserved
28-0 LC	loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. This is the minimum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value to be programmed is corresponds to twice the maxFrameSize transmitted from this queue. If the maxFrameSize is 8192 bytes, then (8192*2) * 8 * 1024 = 134,217,728 or 0x0800_0000. Because it is a negative value, the programmed value is 2's complement of the value, that is, 0x1800_0000.

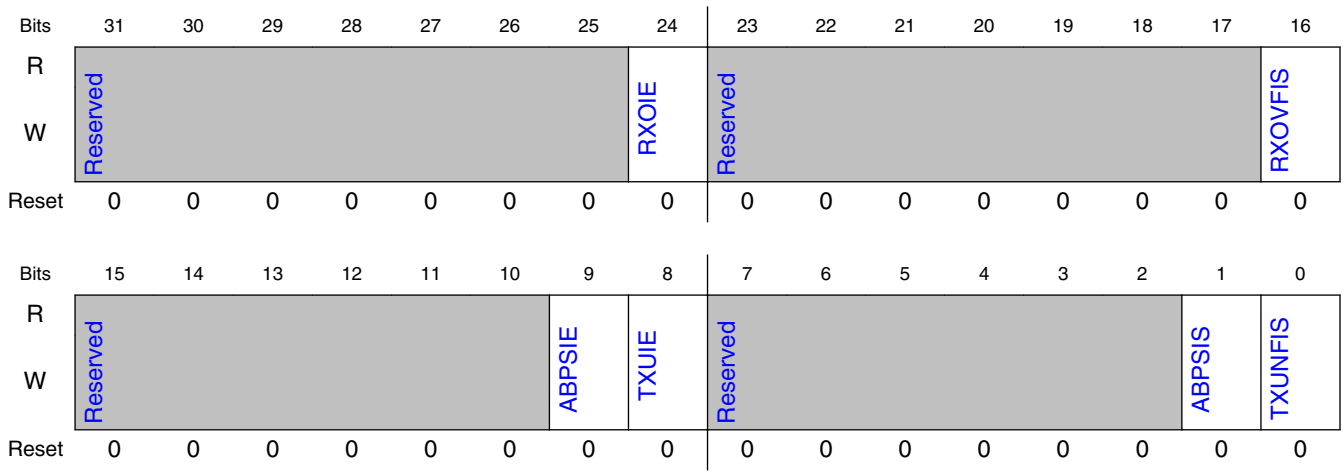
11.7.6.1.440 Queue 3 Interrupt Control Status (MTL_Q3_INTERRUPT_CONTROL_STATUS)

This register contains the interrupt enable and status bits for the queue 3 interrupts.

11.7.6.1.440.1 Offset

Register	Offset
MTL_Q3_INTERRUPT_CONTROL_STATUS	DECh

11.7.6.1.440.2 Diagram



11.7.6.1.440.3 Fields

Field	Description
31-25 —	Reserved
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0 - Receive Queue Overflow Interrupt is disabled 1 - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Queue Overflow Interrupt Status not detected 1 - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9 ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. Average Bits Per Slot Interrupt Enable

Table continues on the next page...

Field	Description
	When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0 - Average Bits Per Slot Interrupt is disabled 1 - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0 - Transmit Queue Underflow Interrupt Status is disabled 1 - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Average Bits Per Slot Interrupt Status not detected 1 - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Queue Underflow Interrupt Status not detected 1 - Transmit Queue Underflow Interrupt Status detected

11.7.6.1.441 Queue 3 Receive Operation Mode (MTL_RXQ3_OPERATION_MODE)

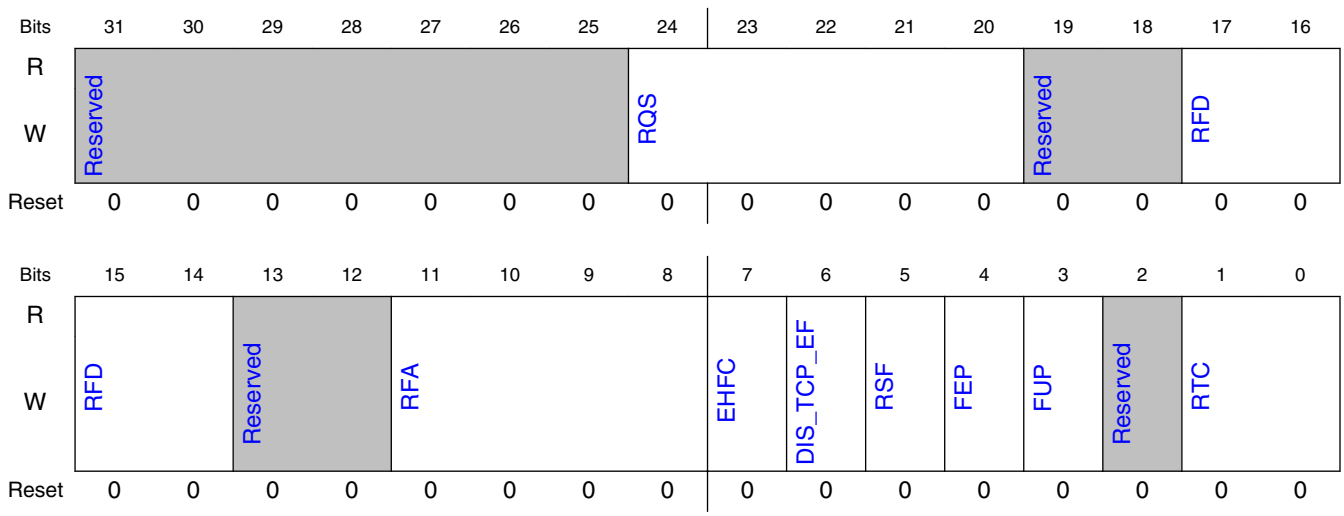
The Queue 3 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

11.7.6.1.441.1 Offset

Register	Offset
MTL_RXQ3_OPERATION_MODE	DF0h

11.7.6.1.441.2 Diagram



11.7.6.1.441.3 Fields

Field	Description
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
19-18 —	Reserved

Table continues on the next page...

Field	Description
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0 - Hardware Flow Control is disabled 1 - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0 - Dropping of TCP/IP Checksum Error Packets is enabled 1 - Dropping of TCP/IP Checksum Error Packets is disabled

Table continues on the next page...

Field	Description
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0 - Receive Queue Store and Forward is disabled 1 - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0 - Forward Error Packets is disabled 1 - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0 - Forward Undersized Good Packets is disabled 1 - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00 - 64 01 - 32 10 - 96 11 - 128

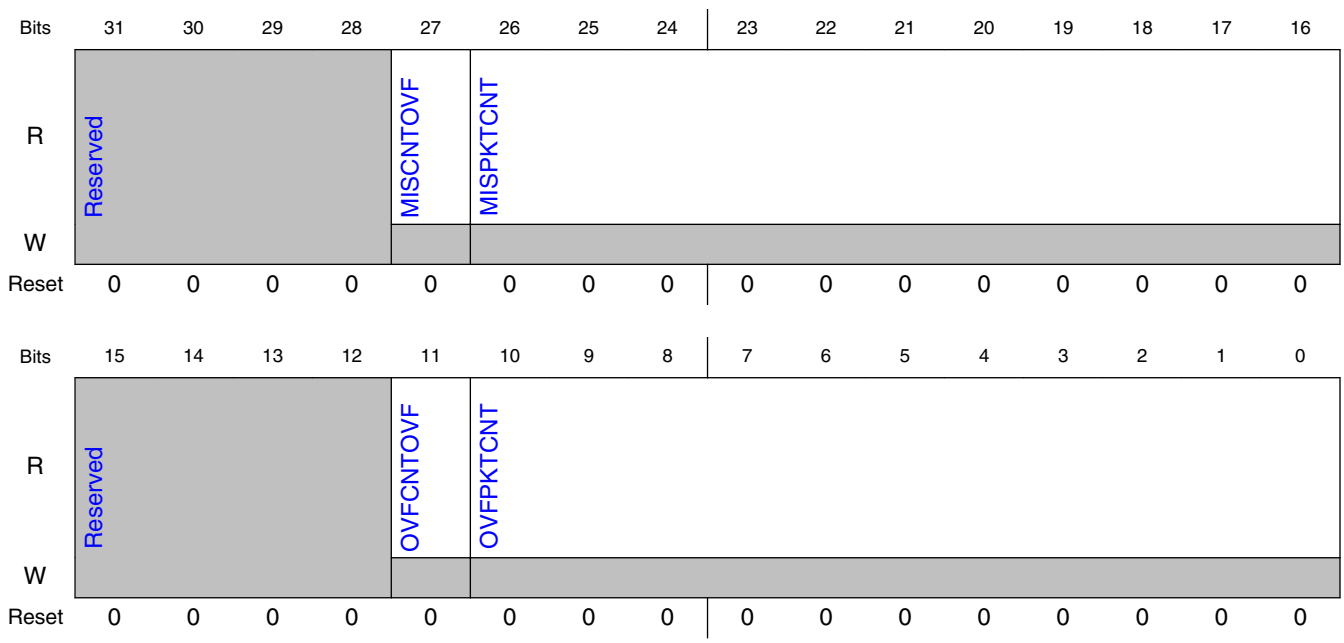
11.7.6.1.442 Queue 3 Missed Packet and Overflow Counter (MTL_RXQ3_MISSED_PACKET_OVERFLOW_CNT)

The Queue 3 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

11.7.6.1.442.1 Offset

Register	Offset
MTL_RXQ3_MISSED_PACKET_OVERFLOW_CNT	DF4h

11.7.6.1.442.2 Diagram



11.7.6.1.442.3 Fields

Field	Description
31-28 —	Reserved

Table continues on the next page...

Field	Description
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Missed Packet Counter overflow not detected 1 - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow Counter overflow not detected 1 - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

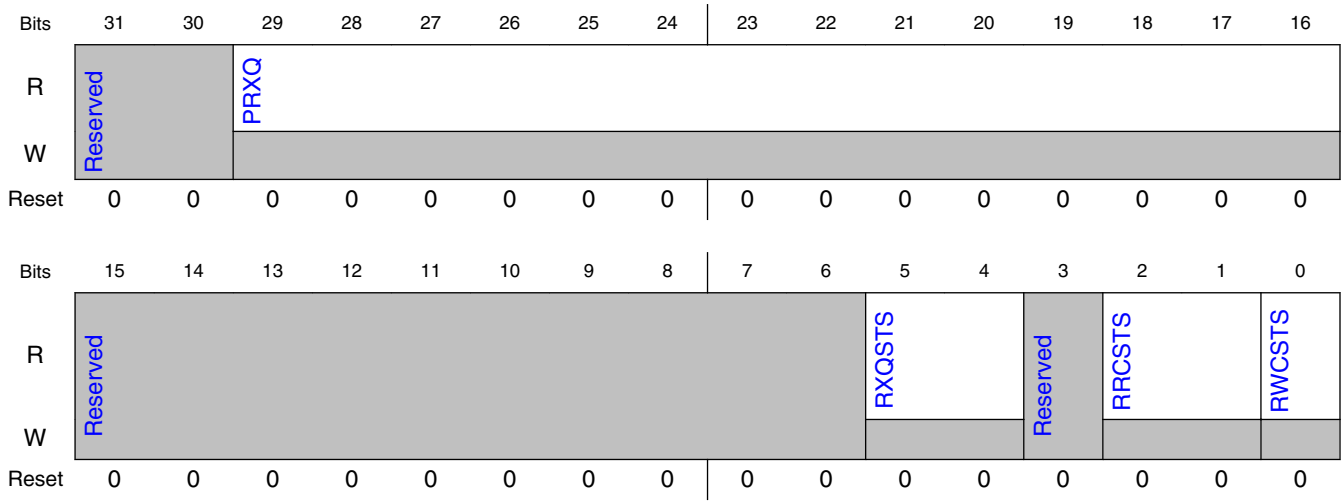
11.7.6.1.443 Queue 3 Receive Debug (MTL_RXQ3_DEBUG)

The Queue 3 Receive Debug register gives the debug status of various blocks related to the Receive queue.

11.7.6.1.443.1 Offset

Register	Offset
MTL_RXQ3_DEBUG	DF8h

11.7.6.1.443.2 Diagram



11.7.6.1.443.3 Fields

Field	Description
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00 - Rx Queue empty 01 - Rx Queue fill-level below flow-control deactivate threshold 10 - Rx Queue fill-level above flow-control activate threshold 11 - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00 - Idle state 01 - Reading packet data 10 - Reading packet status (or timestamp) 11 - Flushing the packet data and status

Table continues on the next page...

Field	Description
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0 - MTL Rx Queue Write Controller Active Status not detected 1 - MTL Rx Queue Write Controller Active Status detected

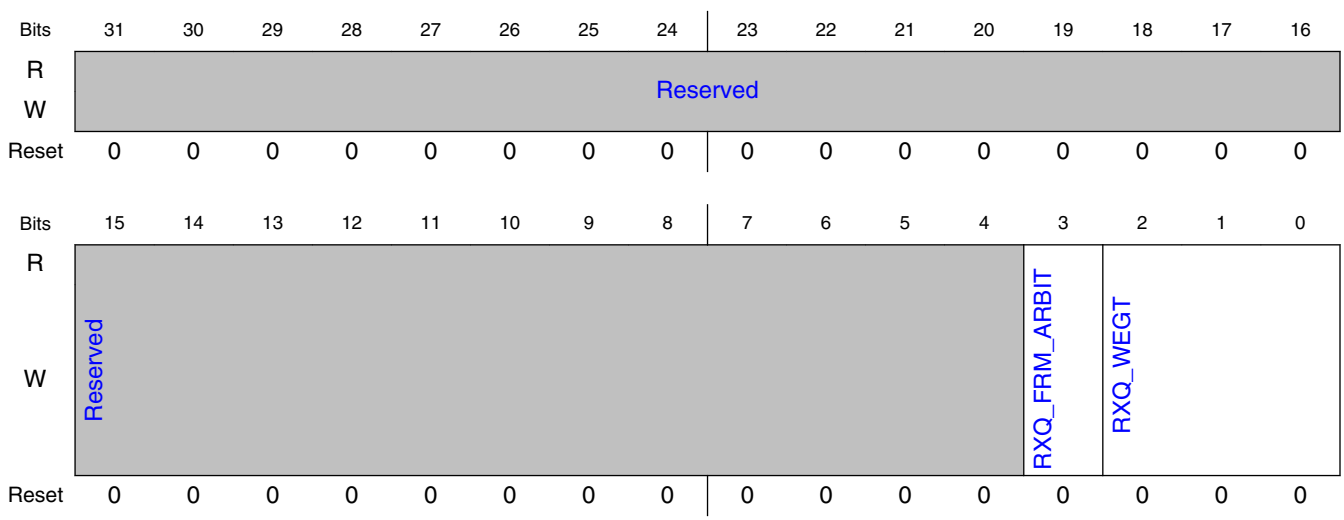
11.7.6.1.444 Queue 3 Receive Control (MTL_RXQ3_CONTROL)

The Queue 3 Receive Control register controls the receive arbitration and passing of received packets to the application.

11.7.6.1.444.1 Offset

Register	Offset
MTL_RXQ3_CONTROL	DFCh

11.7.6.1.444.2 Diagram



11.7.6.1.444.3 Fields

Field	Description
31-4 —	Reserved

Table continues on the next page...

Field	Description
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0 - Receive Queue Packet Arbitration is disabled 1 - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

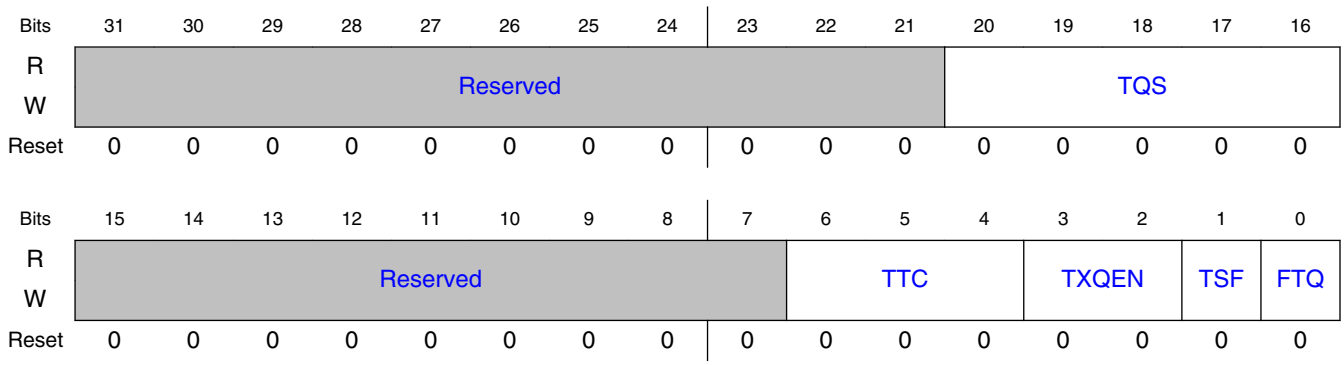
11.7.6.1.445 Queue 4 Transmit Operation Mode (MTL_TXQ4_OPERATION_MODE)

The Queue 4 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

11.7.6.1.445.1 Offset

Register	Offset
MTL_TXQ4_OPERATION_MODE	E00h

11.7.6.1.445.2 Diagram



11.7.6.1.445.3 Fields

Field	Description
31-21 —	Reserved
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000 - 32 001 - 64 010 - 96 011 - 128 100 - 192 101 - 256 110 - 384 111 - 512

Table continues on the next page...

Field	Description
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Enable in AV mode - 2'b10: Enabled - 2'b11: Reserved Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00 - Not enabled 01 - Enable in AV mode (Reserved in non-AV) 10 - Enabled 11 - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0 - Transmit Store and Forward is disabled 1 - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ1_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Flush Transmit Queue is disabled 1 - Flush Transmit Queue is enabled

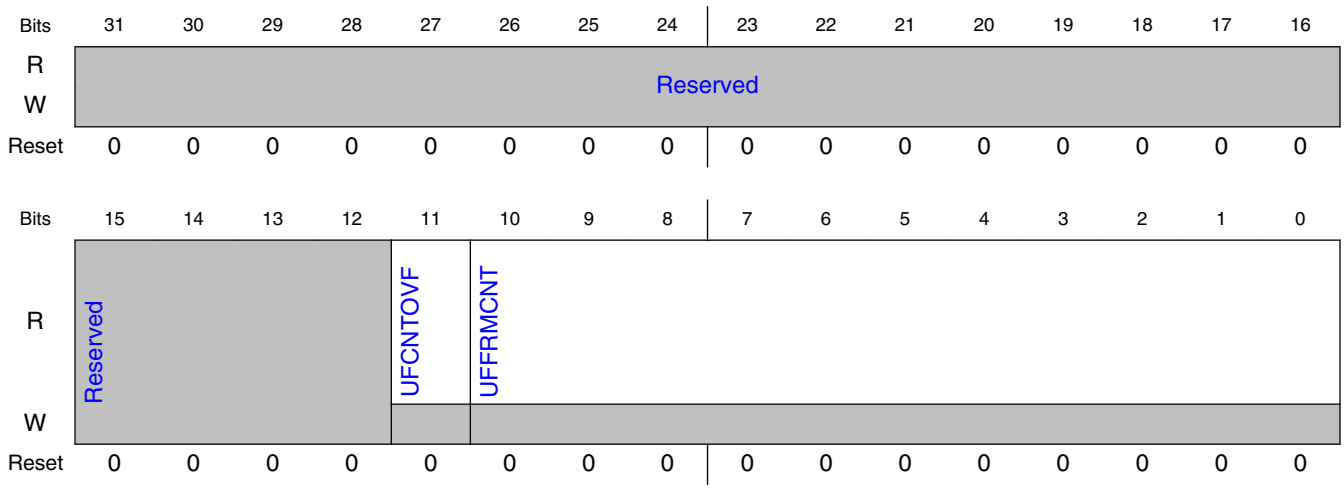
11.7.6.1.446 Queue 4 Underflow Counter (MTL_TXQ4_UNDERFLOW)

The Queue 4 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

11.7.6.1.446.1 Offset

Register	Offset
MTL_TXQ4_UNDERFLOW	E04h

11.7.6.1.446.2 Diagram



11.7.6.1.446.3 Fields

Field	Description
31-12 —	Reserved
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow not detected for Underflow Packet Counter 1 - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

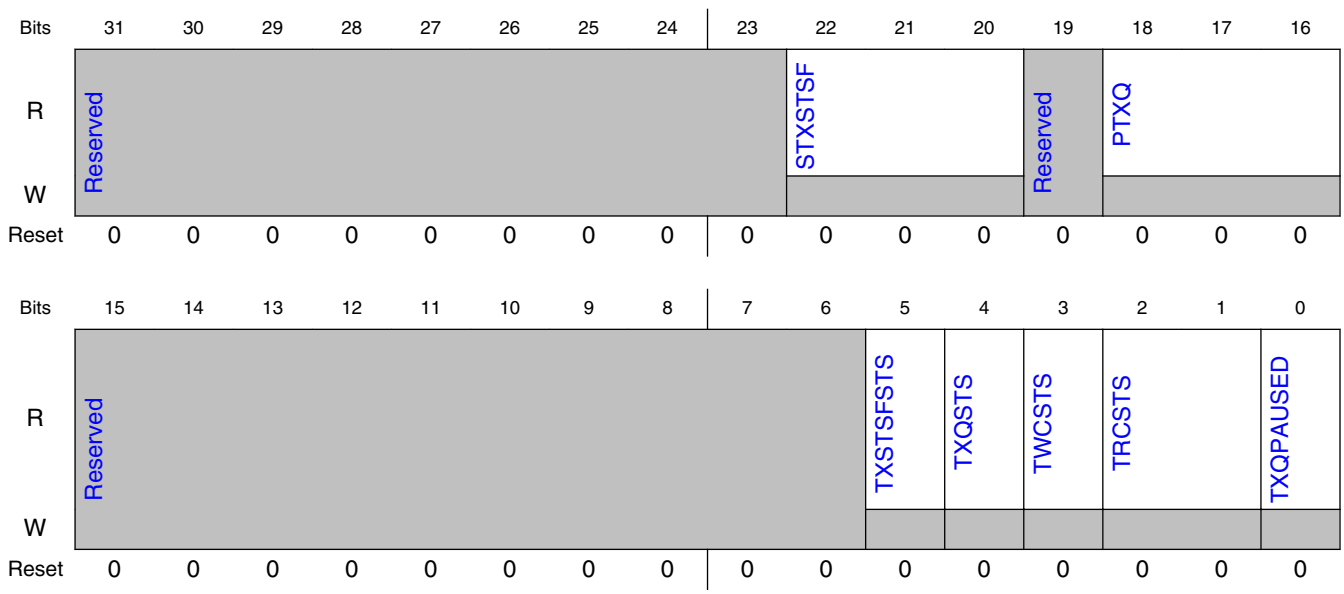
11.7.6.1.447 Queue 4 Transmit Debug (MTL_TXQ4_DEBUG)

The Queue 4 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

11.7.6.1.447.1 Offset

Register	Offset
MTL_TXQ4_DEBUG	E08h

11.7.6.1.447.2 Diagram



11.7.6.1.447.3 Fields

Field	Description
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.

Table continues on the next page...

Field	Description
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0 - MTL Tx Status FIFO Full status is not detected 1 - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0 - MTL Tx Queue Not Empty status is not detected 1 - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0 - MTL Tx Queue Write Controller status is not detected 1 - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00 - Idle state 01 - Read state (transferring data to the MAC transmitter) 10 - Waiting for pending Tx Status from the MAC transmitter 11 - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0 - Transmit Queue in Pause status is not detected 1 - Transmit Queue in Pause status is detected

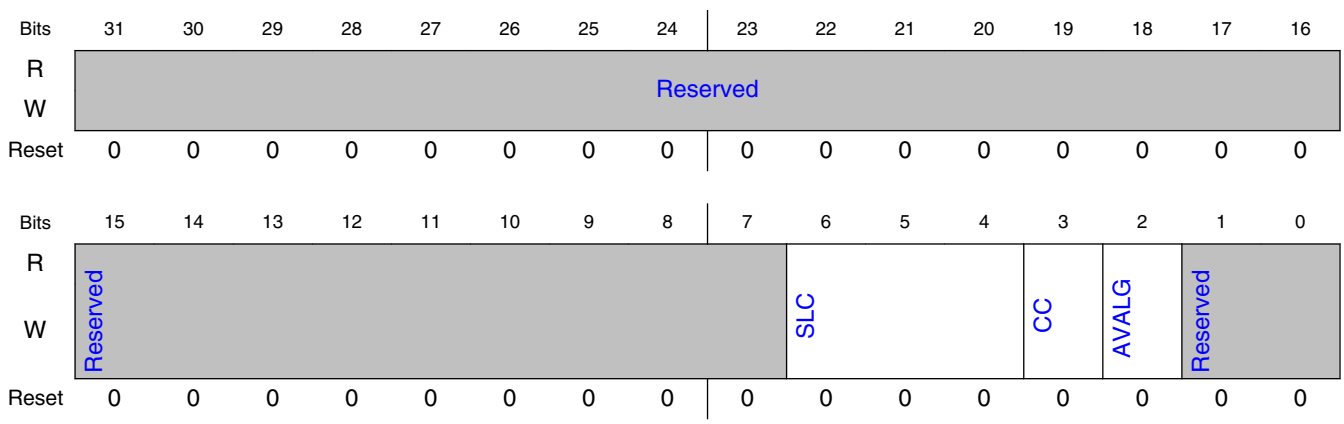
11.7.6.1.448 Queue 4 ETS Control (MTL_TXQ4_ETS_CONTROL)

The Queue 4 ETS Control register controls the enhanced transmission selection operation.

11.7.6.1.448.1 Offset

Register	Offset
MTL_TXQ4_ETS_CONTROL	E10h

11.7.6.1.448.2 Diagram



11.7.6.1.448.3 Fields

Field	Description
31-7 —	Reserved
6-4 SLC	Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH[N]_SLOT_INTERVAL register) over which the average transmitted bits per slot, provided in the MTL_TXQ[N]_ETS_STATUS register, need to be computed for Queue. Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH[N]_SLOT_INTERVAL register) over which the average transmitted bits per slot, provided in the MTL_TXQ[N]_ETS_STATUS register, need to be computed for Queue. The encoding is as follows: 000 - 1 slot 001 - 2 slots 010 - 4 slots 011 - 8 slots

Table continues on the next page...

Field	Description
	100 - 16 slots 101 - Reserved
3 CC	Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 1. Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 1. The credit accumulates even when there is no packet waiting in Channel 1 and another channel is transmitting. When this bit is reset, the accumulated credit parameter in the credit-based shaper algorithm logic is set to zero when there is positive credit and no packet to transmit in Channel 1. When there is no packet waiting in Channel 1 and other channel is transmitting, no credit is accumulated. 0 - Credit Control is disabled 1 - Credit Control is enabled
2 AVALG	AV Algorithm When Queue 1 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 1 traffic. AV Algorithm When Queue 1 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 1 traffic. When reset, strict priority is selected. 0 - CBS Algorithm is disabled 1 - CBS Algorithm is enabled
1-0 —	Reserved

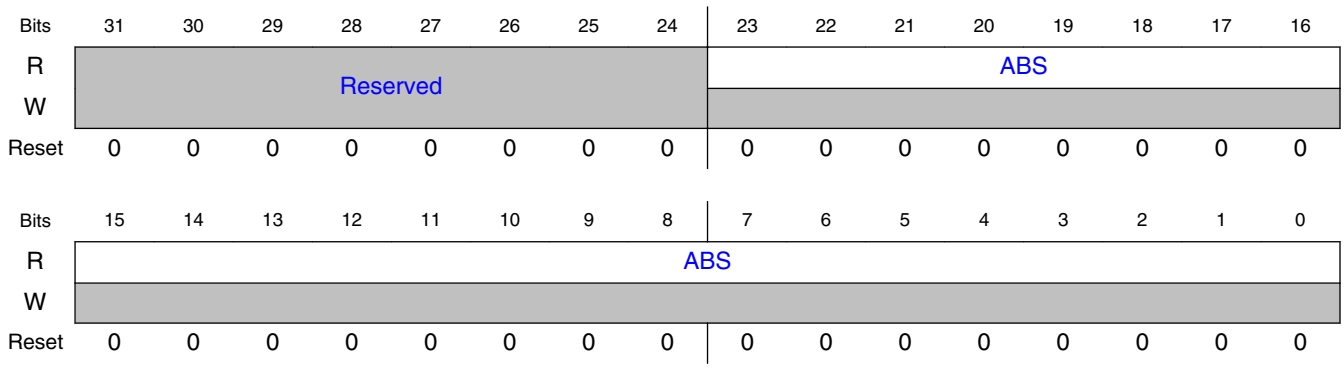
11.7.6.1.449 Queue 4 ETS Status (MTL_TXQ4_ETS_STATUS)

The Queue 4 ETS Status register provides the average traffic transmitted in Queue 4.

11.7.6.1.449.1 Offset

Register	Offset
MTL_TXQ4_ETS_STATUS	E14h

11.7.6.1.449.2 Diagram



11.7.6.1.449.3 Fields

Field	Description
31-24 —	Reserved
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TXQ[n]_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps and 9C_4000 in 2500 Mbps mode respectively. When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

11.7.6.1.450 Queue 4 idleSlopeCredit, Quantum or Weights (MTL_TXQ4_QUANTUM_WEIGHT)

The Queue 4 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted in Queue 4.

11.7.6.1.450.1 Offset

Register	Offset
MTL_TXQ4_QUANTUM_WEIGHT	E18h

11.7.6.1.450.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											ISCQW				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ISCQW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.450.3 Fields

Field	Description
31-21 —	Reserved
20-0 ISCQW	idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps; 3.2 ns for 2500 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. Bits[20:14] must be written to zero. - Quantum When the DCB operation is enabled with DWRR algorithm for Queue 1 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. - Weights When DCB operation is enabled with WFQ algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero. - Note 1: In multiple Queue configuration this field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is OPERATION_MODE register. - Note 2: For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula: (previous_finish_time of particular Transmit Queue + (weights*packet_size))

Field	Description
	- Note 3: The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.

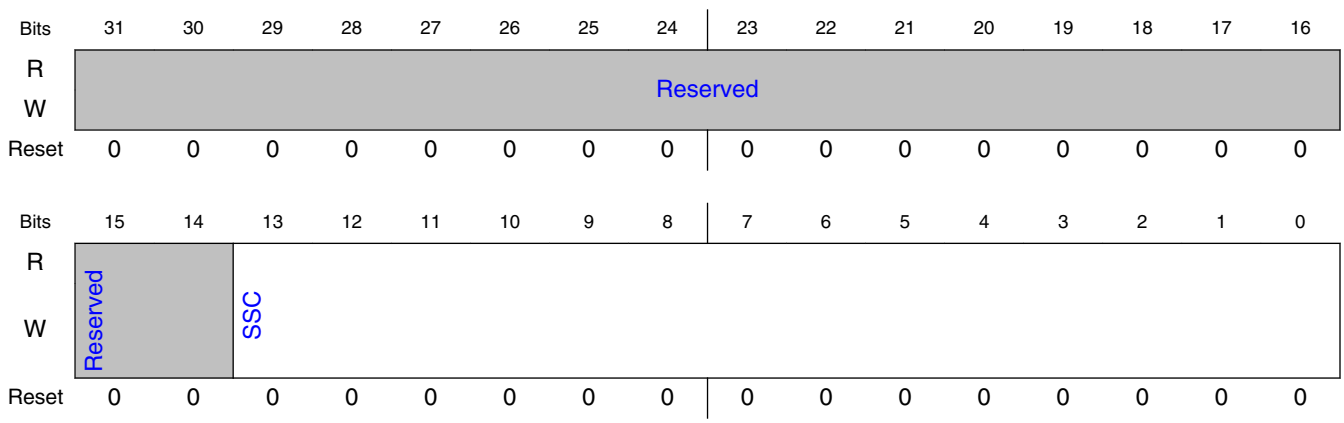
11.7.6.1.451 Queue 4 sendSlopeCredit (MTL_TXQ4_SENDSLOPECREDIT)

The sendSlopeCredit register contains the sendSlope credit value required for the credit-based shaper algorithm for the Queue.

11.7.6.1.451.1 Offset

Register	Offset
MTL_TXQ4_SENDSLOPECREDIT	E1Ch

11.7.6.1.451.2 Diagram



11.7.6.1.451.3 Fields

Field	Description
31-14 —	Reserved
13-0 SSC	sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 1. sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns, 8 ns and 3.2 ns for 100 Mbps, 1000 Mbps and 2500 Mbps respectively) when the credit is decreasing. The software

Field	Description
	should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. This field should be programmed with absolute sendSlopeCredit value. The credit-based shaper logic subtracts it from the accumulated credit when Channel 1 is selected for transmission.

11.7.6.1.452 Queue 4 hiCredit (MTL_TXQ4_HICREDIT)

The hiCredit register contains the hiCredit value required for the credit-based shaper algorithm for the Queue.

11.7.6.1.452.1 Offset

Register	Offset
MTL_TXQ4_HICREDIT	E20h

11.7.6.1.452.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			HC												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.452.3 Fields

Field	Description
31-29 —	Reserved
28-0 HC	hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. This is the maximum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024.

Field	Description
	The maximum value is maxInterferenceSize, that is, best-effort maximum packet size (16,384 bytes or 131,072 bits). The value to be specified is 131,072 * 1,024 = 134,217,728 or 0x0800_0000.

11.7.6.1.453 Queue 4 loCredit (MTL_TXQ4_LOCREDIT)

The loCredit register contains the loCredit value required for the credit-based shaper algorithm for the Queue.

11.7.6.1.453.1 Offset

Register	Offset
MTL_TXQ4_LOCREDIT	E24h

11.7.6.1.453.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			LC												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.453.3 Fields

Field	Description
31-29 —	Reserved
28-0 LC	loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. This is the minimum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value to be programmed is corresponds to twice the maxFrameSize transmitted from this queue. If the maxFrameSize is 8192 bytes, then (8192*2) * 8 * 1024 = 134,217,728 or 0x0800_0000. Because it is a negative value, the programmed value is 2's complement of the value, that is, 0x1800_0000.

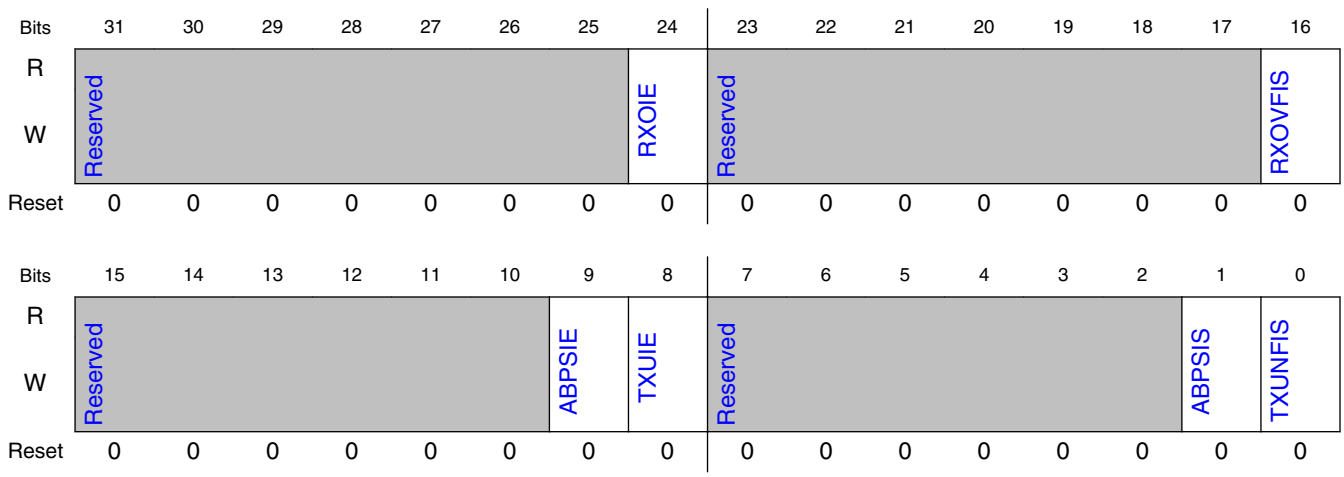
11.7.6.1.454 Queue 4 Interrupt Control Status (MTL_Q4_INTERRUPT_CONTROL_STATUS)

This register contains the interrupt enable and status bits for the queue 4 interrupts.

11.7.6.1.454.1 Offset

Register	Offset
MTL_Q4_INTERRUPT_CONTROL_STATUS	E2Ch

11.7.6.1.454.2 Diagram



11.7.6.1.454.3 Fields

Field	Description
31-25 —	Reserved
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0 - Receive Queue Overflow Interrupt is disabled 1 - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved

Table continues on the next page...

Field	Description
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Queue Overflow Interrupt Status not detected 1 - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9 ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0 - Average Bits Per Slot Interrupt is disabled 1 - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0 - Transmit Queue Underflow Interrupt Status is disabled 1 - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Average Bits Per Slot Interrupt Status not detected 1 - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

Field	Description
	0 - Transmit Queue Underflow Interrupt Status not detected 1 - Transmit Queue Underflow Interrupt Status detected

11.7.6.1.455 Queue 4 Receive Operation Mode (MTL_RXQ4_OPERATION_MODE)

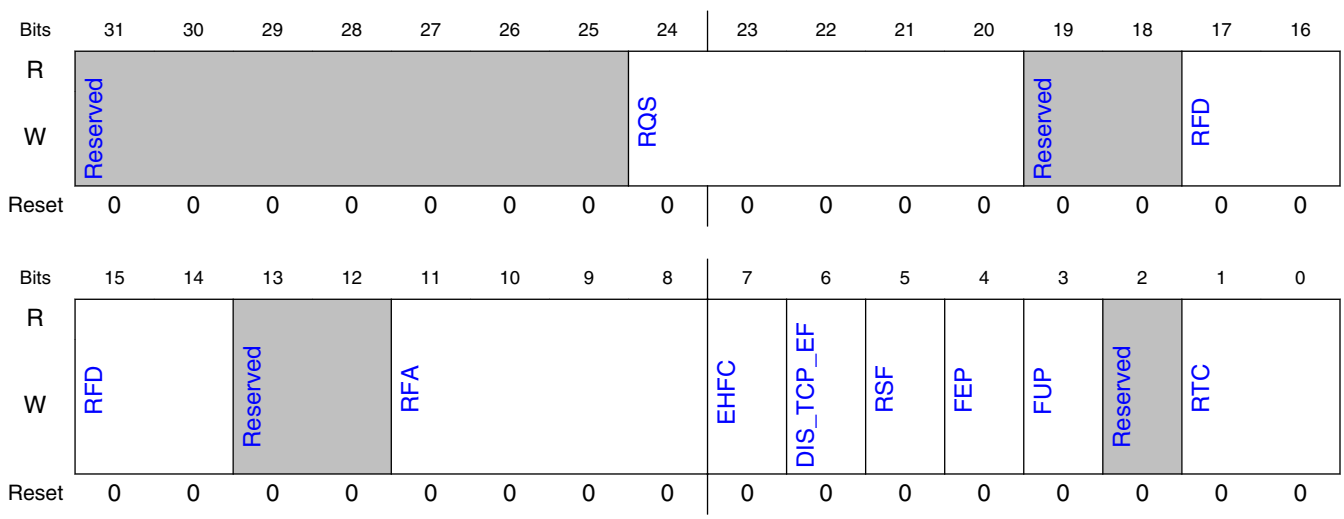
The Queue 4 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

11.7.6.1.455.1 Offset

Register	Offset
MTL_RXQ4_OPERATION_MODE	E30h

11.7.6.1.455.2 Diagram



11.7.6.1.455.3 Fields

Field	Description
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
19-18 —	Reserved
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.

Table continues on the next page...

Field	Description
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0 - Hardware Flow Control is disabled 1 - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0 - Dropping of TCP/IP Checksum Error Packets is enabled 1 - Dropping of TCP/IP Checksum Error Packets is disabled
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0 - Receive Queue Store and Forward is disabled 1 - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0 - Forward Error Packets is disabled 1 - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0 - Forward Undersized Good Packets is disabled

Table continues on the next page...

Field	Description
	1 - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00 - 64 01 - 32 10 - 96 11 - 128

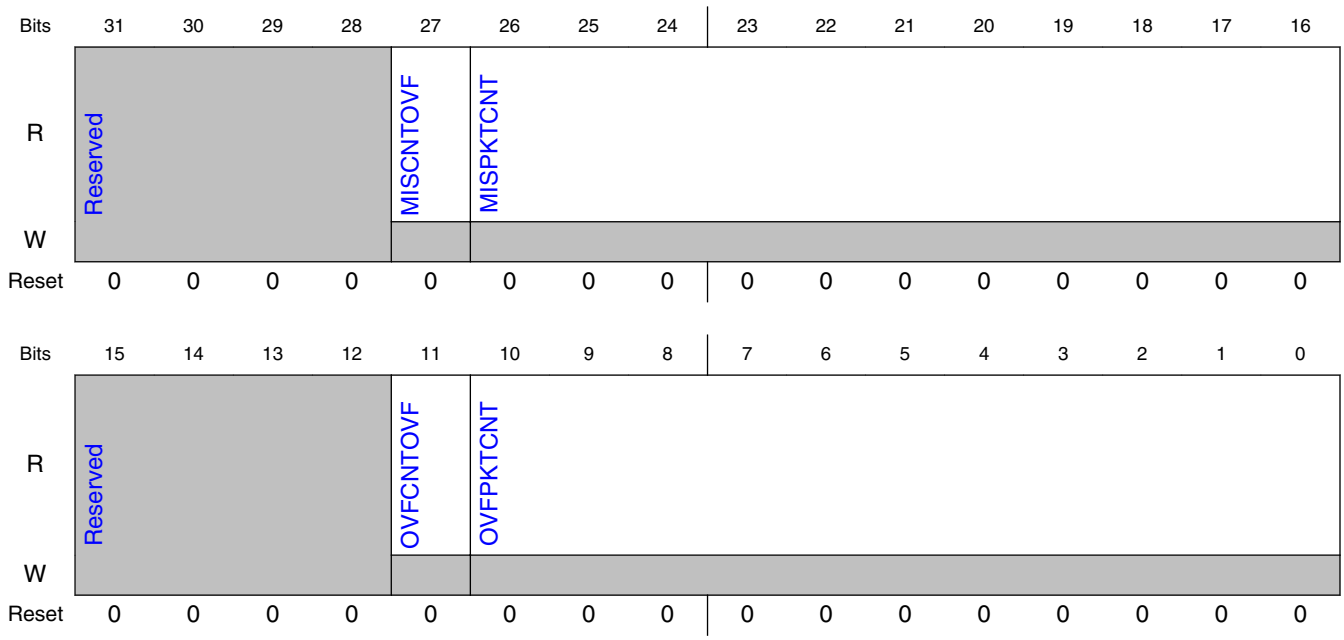
11.7.6.1.456 Queue 4 Missed Packet and Overflow Counter (MTL_RXQ4_MISSED_PACKET_OVERFLOW_CNT)

The Queue 4 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

11.7.6.1.456.1 Offset

Register	Offset
MTL_RXQ4_MISSED_PACKET_OVERFLOW_CNT	E34h

11.7.6.1.456.2 Diagram



11.7.6.1.456.3 Fields

Field	Description
31-28 —	Reserved
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Missed Packet Counter overflow not detected 1 - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved

Table continues on the next page...

Field	Description
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Overflow Counter overflow not detected 1 - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

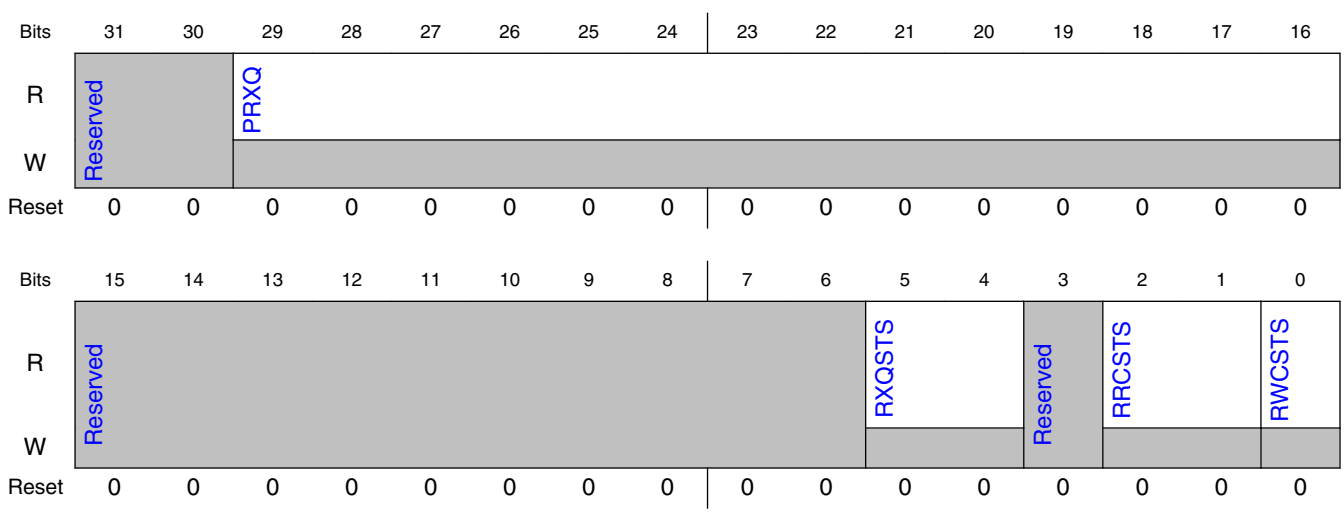
11.7.6.1.457 Queue 4 Receive Debug (MTL_RXQ4_DEBUG)

The Queue 4 Receive Debug register gives the debug status of various blocks related to the Receive queue.

11.7.6.1.457.1 Offset

Register	Offset
MTL_RXQ4_DEBUG	E38h

11.7.6.1.457.2 Diagram



11.7.6.1.457.3 Fields

Field	Description
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00 - Rx Queue empty 01 - Rx Queue fill-level below flow-control deactivate threshold 10 - Rx Queue fill-level above flow-control activate threshold 11 - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00 - Idle state 01 - Reading packet data 10 - Reading packet status (or timestamp) 11 - Flushing the packet data and status
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0 - MTL Rx Queue Write Controller Active Status not detected 1 - MTL Rx Queue Write Controller Active Status detected

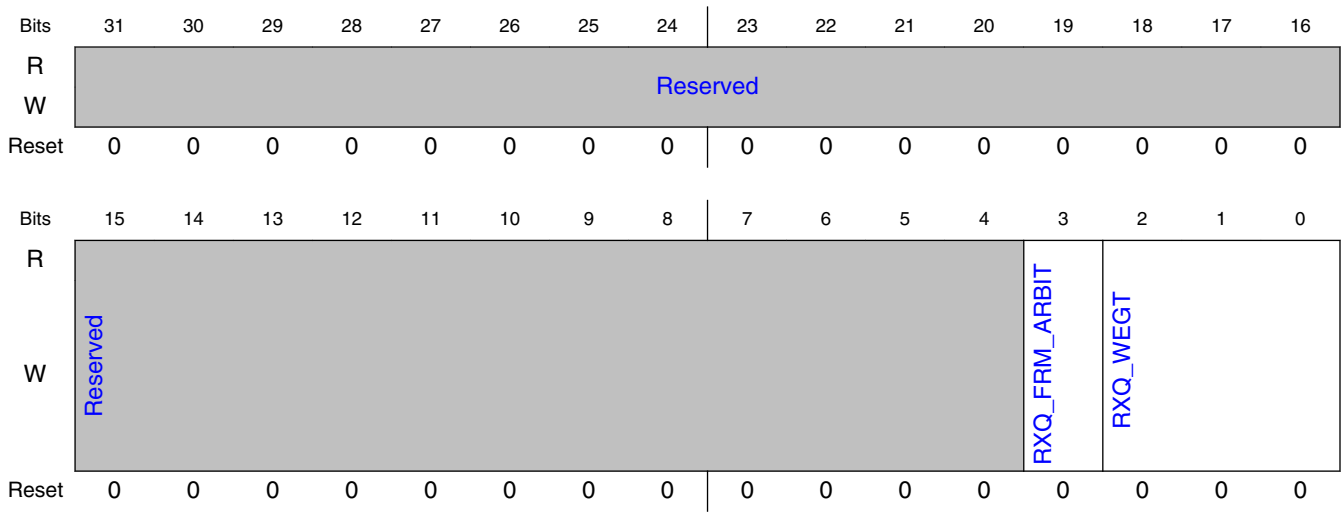
11.7.6.1.458 Queue 4 Receive Control (MTL_RXQ4_CONTROL)

The Queue 4 Receive Control register controls the receive arbitration and passing of received packets to the application.

11.7.6.1.458.1 Offset

Register	Offset
MTL_RXQ4_CONTROL	E3Ch

11.7.6.1.458.2 Diagram



11.7.6.1.458.3 Fields

Field	Description
31-4 —	Reserved
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0 - Receive Queue Packet Arbitration is disabled 1 - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

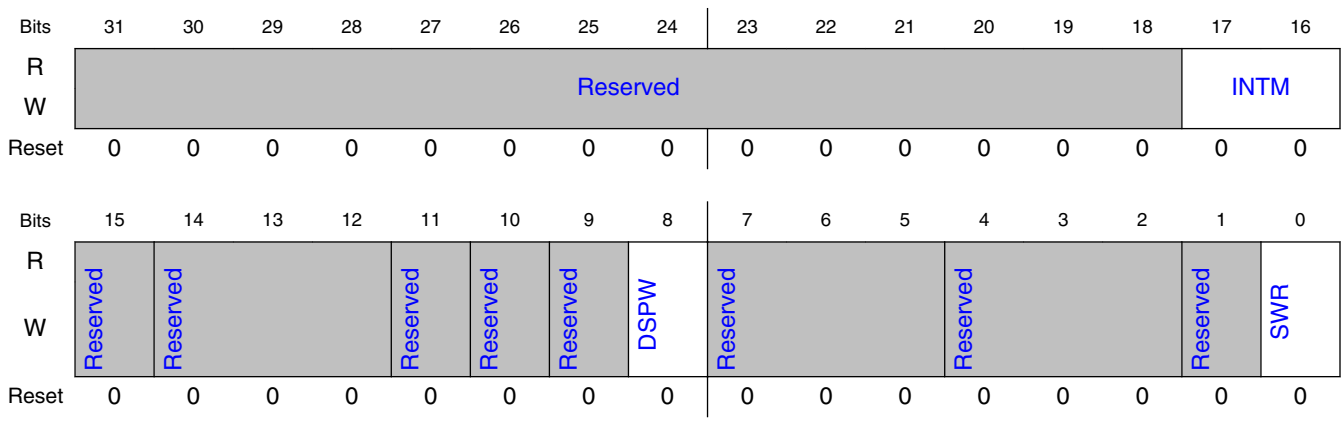
11.7.6.1.459 DMA Bus Mode (DMA_MODE)

The Bus Mode register establishes the bus operating modes for the DMA.

11.7.6.1.459.1 Offset

Register	Offset
DMA_MODE	1000h

11.7.6.1.459.2 Diagram



11.7.6.1.459.3 Fields

Field	Description
31-18 —	Reserved
17-16 INTM	Interrupt Mode This field defines the interrupt mode of DWC_ether_qos. Interrupt Mode This field defines the interrupt mode of DWC_ether_qos. The behavior of the following outputs changes depending on the following settings: - sbd_perch_tx_intr_o[] (Transmit Per Channel Interrupt) - sbd_perch_rx_intr_o[] (Receive Per Channel Interrupt) - sbd_intr_o (Common Interrupt) It also changes the behavior of the RI/TI bits in the STATUS. - 00: sbd_perch_* are pulse signals for each TX/RX packet transfer completion events (irrespective of whether corresponding interrupts are enabled) for which IOC bits are enabled in descriptor. sbd_intr_o is also asserted when corresponding interrupts are enabled and cleared only when software clears the corresponding RI/TI status bits.

Table continues on the next page...

Field	Description
	- 01: sbd_perch_* are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. The sbd_intr_o is not asserted for these TX/RX packet transfer completion events. - 10: sbd_perch_* are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. However, the signal is asserted again if the same event occurred again before it was cleared. The sbd_intr_o is not asserted for these TX/RX packet transfer completion events. - 11: Reserved For more details please refer Table "DWC_ether_qos Transfer Complete Interrupt Behavior". 00 - See above description 01 - See above description 10 - See above description 11 - Reserved
15 —	Reserved
14-12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 DSPW	Descriptor Posted Write When this bit is set to 0, the descriptor writes are always non-posted. Descriptor Posted Write When this bit is set to 0, the descriptor writes are always non-posted. When this bit is set to 1, the descriptor writes are non-posted only when IOC (Interrupt on completion) is set in last descriptor, otherwise the descriptor writes are always posted. 0 - Descriptor Posted Write is disabled 1 - Descriptor Posted Write is enabled
7-5 —	Reserved
4-2 —	Reserved
1 —	Reserved
0 SWR	Software Reset When this bit is set, the MAC and the DMA controller reset the logic and all internal registers of the DMA, MTL, and MAC. Software Reset

Field	Description
	When this bit is set, the MAC and the DMA controller reset the logic and all internal registers of the DMA, MTL, and MAC. This bit is automatically cleared after the reset operation is complete in all DWC_ether_qos clock domains. Before reprogramming any DWC_ether_qos register, a value of zero should be read in this bit. This bit must be read at least 4 CSR clock cycles after it is written to 1. Note: The reset operation is complete only when all resets in all active clock domains are de-asserted. Therefore, it is essential that all PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion. The time to complete the software reset operation depends on the frequency of the slowest active clock. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0 - Software Reset is disabled 1 - Software Reset is enabled

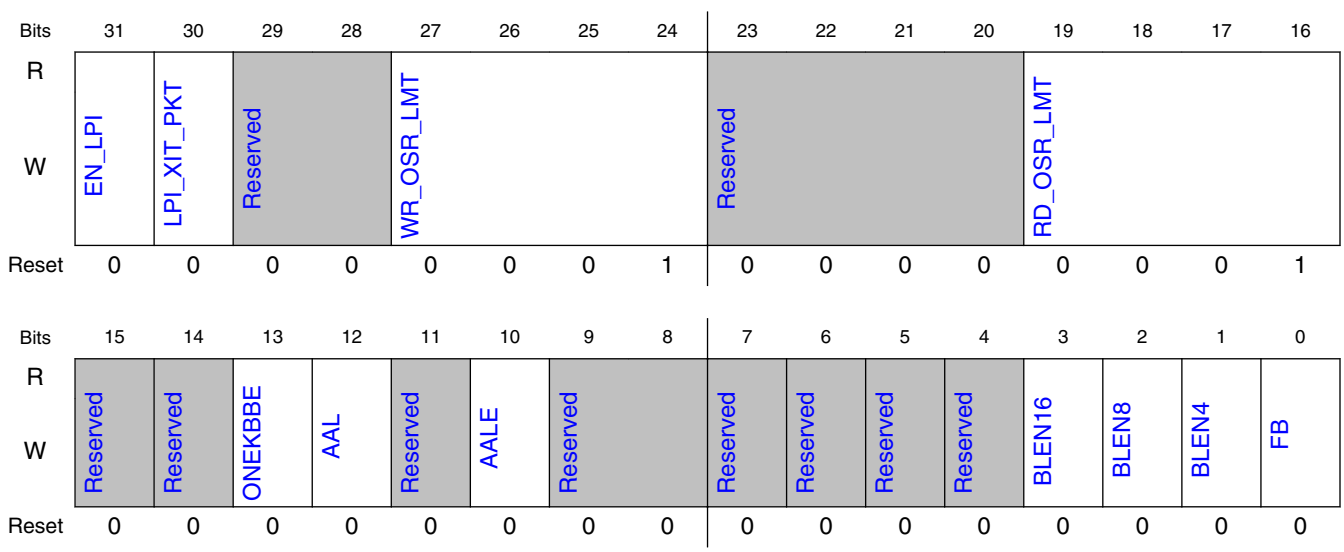
11.7.6.1.460 DMA System Bus Mode (DMA_SYSBUS_MODE)

The System Bus mode register controls the behavior of the AHB or AXI master. It mainly controls burst splitting and number of outstanding requests.

11.7.6.1.460.1 Offset

Register	Offset
DMA_SYSBUS_MODE	1004h

11.7.6.1.460.2 Diagram



11.7.6.1.460.3 Fields

Field	Description
31 EN_LPI	Enable Low Power Interface (LPI) When set to 1, this bit enables the LPI mode supported by the EQOS-AXI configuration and accepts the LPI request from the AXI System Clock controller. Enable Low Power Interface (LPI) When set to 1, this bit enables the LPI mode supported by the EQOS-AXI configuration and accepts the LPI request from the AXI System Clock controller. When set to 0, this bit disables the LPI mode and always denies the LPI request from the AXI System Clock controller. 0 - Low Power Interface (LPI) is disabled 1 - Low Power Interface (LPI) is enabled
30 LPI_XIT_PKT	Unlock on Magic Packet or Remote Wake-Up Packet When set to 1, this bit enables the AXI master to come out of the LPI mode only when the magic packet or remote wake-up packet is received. Unlock on Magic Packet or Remote Wake-Up Packet When set to 1, this bit enables the AXI master to come out of the LPI mode only when the magic packet or remote wake-up packet is received. When set to 0, this bit enables the AXI master to come out of the LPI mode when any packet is received. 0 - Unlock on Magic Packet or Remote Wake-Up Packet is disabled 1 - Unlock on Magic Packet or Remote Wake-Up Packet is enabled
29-28 —	Reserved
27-24 WR_OSR_LMT	AXI Maximum Write Outstanding Request Limit This value limits the maximum outstanding request on the AXI write interface. AXI Maximum Write Outstanding Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT + 1 Note: - Bit 26 is reserved if DWC_ETHER_QOS_AXI_MAX_WR_REQ = 4 - Bit 27 is reserved if DWC_ETHER_QOS_AXI_MAX_WR_REQ!= 16
23-20 —	Reserved
19-16 RD_OSR_LMT	AXI Maximum Read Outstanding Request Limit This value limits the maximum outstanding request on the AXI read interface. AXI Maximum Read Outstanding Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT + 1 Note: - Bit 18 is reserved if parameter DWC_ETHER_QOS_AXI_MAX_RD_REQ = 4 - Bit 19 is reserved if parameter DWC_ETHER_QOS_AXI_MAX_RD_REQ!= 16
15 —	Reserved
14	Reserved

Table continues on the next page...

Field	Description
—	
13 ONEKBBE	1 KB Boundary Crossing Enable for the EQOS-AXI Master When set, the burst transfers performed by the EQOS-AXI master do not cross 1 KB boundary. 1 KB Boundary Crossing Enable for the EQOS-AXI Master When set, the burst transfers performed by the EQOS-AXI master do not cross 1 KB boundary. When reset, the burst transfers performed by the EQOS-AXI master do not cross 4 KB boundary. 0 - 1 KB Boundary Crossing for the EQOS-AXI Master Beats is disabled 1 - 1 KB Boundary Crossing for the EQOS-AXI Master Beats is enabled
12 AAL	Address-Aligned Beats When this bit is set to 1, the EQOS-AXI or EQOS-AHB master performs address-aligned burst transfers on Read and Write channels. 0 - Address-Aligned Beats is disabled 1 - Address-Aligned Beats is enabled
11 —	Reserved
10 AALE	Automatic AXI LPI enable When set to 1, enables the AXI master to enter into LPI state when there is no activity in the DWC_ether_qos for number of system clock cycles programmed in the LPIEI field of DMA_AXI_LPI_ENTRY_INTERVAL register. 0 - Automatic AXI LPI is disabled 1 - Automatic AXI LPI is enabled
9-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 BLEN16	AXI Burst Length 16 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 16 on the AXI interface. AXI Burst Length 16 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 16 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. 0 - No effect 1 - AXI Burst Length 16
2 BLEN8	AXI Burst Length 8 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 8 on the AXI interface. AXI Burst Length 8 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 8 on the AXI interface.

Table continues on the next page...

Field	Description
	When the FB bit is set to 0, setting this bit has no effect. 0 - No effect 1 - AXI Burst Length 8
1 BLEN4	AXI Burst Length 4 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 4 on the AXI interface. AXI Burst Length 4 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 4 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. 0 - No effect 1 - AXI Burst Length 4
0 FB	Fixed Burst Length When this bit is set to 1, the EQOS-AXI master initiates burst transfers of specified lengths as given below. Fixed Burst Length When this bit is set to 1, the EQOS-AXI master initiates burst transfers of specified lengths as given below. - Burst transfers of fixed burst lengths as indicated by the BLEN256, BLEN128, BLEN64, BLEN32, BLEN16, BLEN8, or BLEN4 field - Burst transfers of length 1 When this bit is set to 0, the EQOS-AXI master initiates burst transfers that are equal to or less than the maximum allowed burst length programmed in Bits[7:1]. 0 - Fixed Burst Length is disabled 1 - Fixed Burst Length is enabled

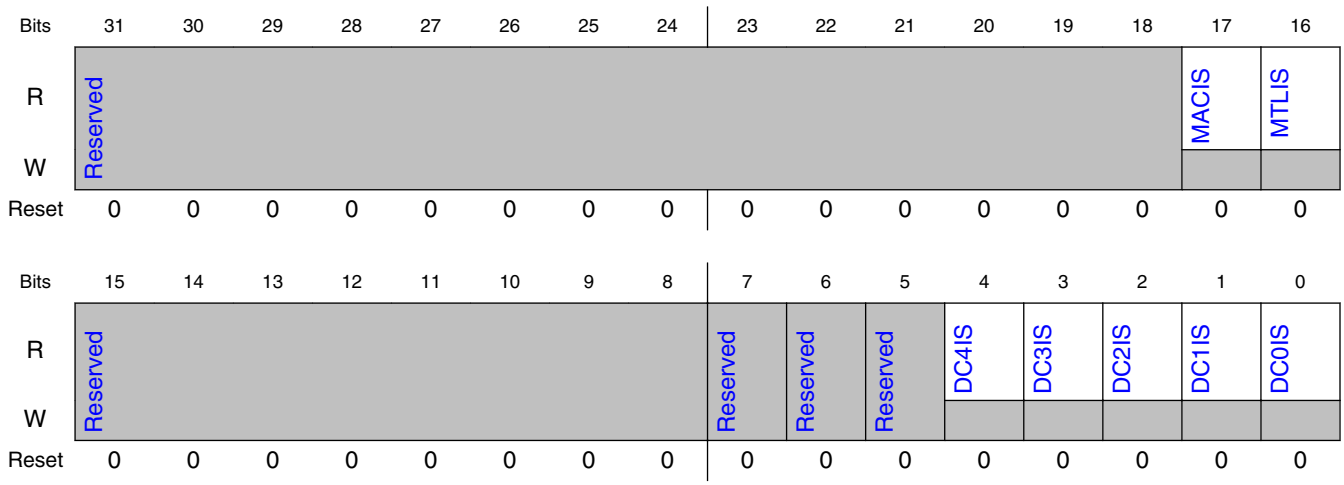
11.7.6.1.461 DMA Interrupt Status (DMA_INTERRUPT_STATUS)

The application reads this Interrupt Status register during interrupt service routine or polling to determine the interrupt status of DMA channels, MTL queues, and the MAC.

11.7.6.1.461.1 Offset

Register	Offset
DMA_INTERRUPT_STATUS	1008h

11.7.6.1.461.2 Diagram



11.7.6.1.461.3 Fields

Field	Description
31-18 —	Reserved
17 MACIS	MAC Interrupt Status This bit indicates an interrupt event in the MAC. MAC Interrupt Status This bit indicates an interrupt event in the MAC. To reset this bit to 1'b0, the software must read the corresponding register in the MAC to get the exact cause of the interrupt and clear its source. 0 - MAC Interrupt Status not detected 1 - MAC Interrupt Status detected
16 MTLIS	MTL Interrupt Status This bit indicates an interrupt event in the MTL. MTL Interrupt Status This bit indicates an interrupt event in the MTL. To reset this bit to 1'b0, the software must read the corresponding register in the MTL to get the exact cause of the interrupt and clear its source. 0 - MTL Interrupt Status not detected 1 - MTL Interrupt Status detected
15-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4	DMA Channel 4 Interrupt Status This bit indicates an interrupt event in DMA Channel 4.

Table continues on the next page...

Field	Description
DC4IS	DMA Channel 4 Interrupt Status This bit indicates an interrupt event in DMA Channel 4. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 4 to get the exact cause of the interrupt and clear its source. 0 - DMA Channel 4 Interrupt Status not detected 1 - DMA Channel 4 Interrupt Status detected
3 DC3IS	DMA Channel 3 Interrupt Status This bit indicates an interrupt event in DMA Channel 3. DMA Channel 3 Interrupt Status This bit indicates an interrupt event in DMA Channel 3. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 3 to get the exact cause of the interrupt and clear its source. 0 - DMA Channel 3 Interrupt Status not detected 1 - DMA Channel 3 Interrupt Status detected
2 DC2IS	DMA Channel 2 Interrupt Status This bit indicates an interrupt event in DMA Channel 2. DMA Channel 2 Interrupt Status This bit indicates an interrupt event in DMA Channel 2. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 2 to get the exact cause of the interrupt and clear its source. 0 - DMA Channel 2 Interrupt Status not detected 1 - DMA Channel 2 Interrupt Status detected
1 DC1IS	DMA Channel 1 Interrupt Status This bit indicates an interrupt event in DMA Channel 1. DMA Channel 1 Interrupt Status This bit indicates an interrupt event in DMA Channel 1. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 1 to get the exact cause of the interrupt and clear its source. 0 - DMA Channel 1 Interrupt Status not detected 1 - DMA Channel 1 Interrupt Status detected
0 DC0IS	DMA Channel 0 Interrupt Status This bit indicates an interrupt event in DMA Channel 0. DMA Channel 0 Interrupt Status This bit indicates an interrupt event in DMA Channel 0. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 0 to get the exact cause of the interrupt and clear its source. 0 - DMA Channel 0 Interrupt Status not detected 1 - DMA Channel 0 Interrupt Status detected

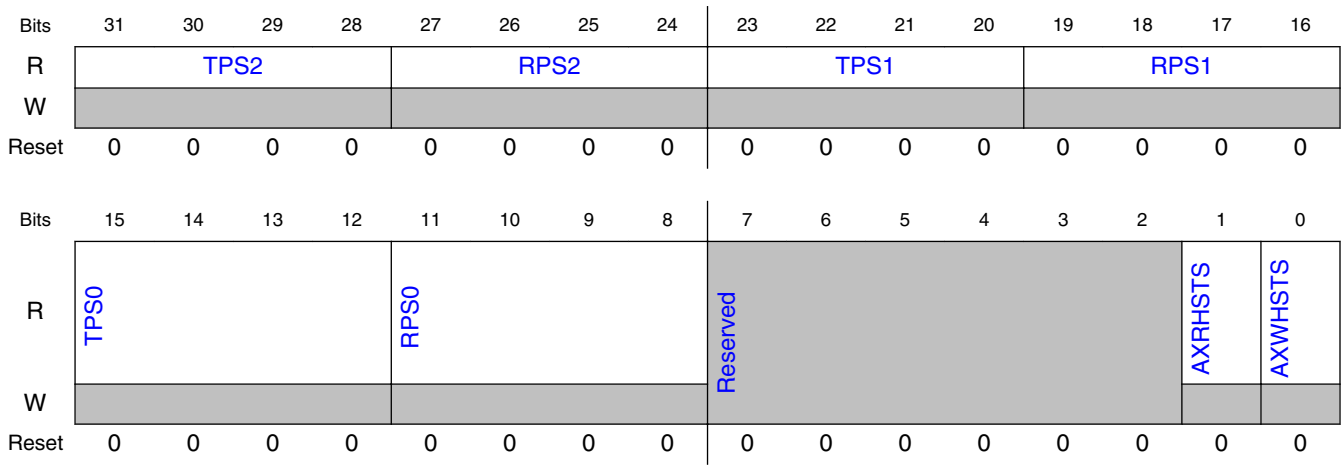
11.7.6.1.462 DMA Debug Status 0 (DMA_DEBUG_STATUS0)

The Debug Status 0 register gives the Receive and Transmit process status for DMA Channel 0-Channel 2 for debugging purpose.

11.7.6.1.462.1 Offset

Register	Offset
DMA_DEBUG_STATUS0	100Ch

11.7.6.1.462.2 Diagram



11.7.6.1.462.3 Fields

Field	Description
31-28 TPS2	DMA Channel 2 Transmit Process State This field indicates the Tx DMA FSM state for Channel 2. DMA Channel 2 Transmit Process State This field indicates the Tx DMA FSM state for Channel 2. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Transmit Command issued) 0001 - Running (Fetching Tx Transfer Descriptor) 0010 - Running (Waiting for status) 0011 - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0100 - Timestamp write state 0101 - Reserved for future use 0110 - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0111 - Running (Closing Tx Descriptor)
27-24 RPS2	DMA Channel 2 Receive Process State This field indicates the Rx DMA FSM state for Channel 2. DMA Channel 2 Receive Process State This field indicates the Rx DMA FSM state for Channel 2. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Receive Command issued) 0001 - Running (Fetching Rx Transfer Descriptor) 0010 - Reserved for future use 0011 - Running (Waiting for Rx packet) 0100 - Suspended (Rx Descriptor Unavailable) 0101 - Running (Closing the Rx Descriptor)

Table continues on the next page...

Field	Description
	0110 - Timestamp write state
	0111 - Running (Transferring the received packet data from the Rx buffer to the system memory)
23-20	DMA Channel 1 Transmit Process State This field indicates the Tx DMA FSM state for Channel 1.
TPS1	DMA Channel 1 Transmit Process State
	This field indicates the Tx DMA FSM state for Channel 1.
	The MSB of this field always returns 0. This field does not generate an interrupt.
	0000 - Stopped (Reset or Stop Transmit Command issued)
	0001 - Running (Fetching Tx Transfer Descriptor)
	0010 - Running (Waiting for status)
	0011 - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO))
	0100 - Timestamp write state
	0101 - Reserved for future use
	0110 - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow)
	0111 - Running (Closing Tx Descriptor)
19-16	DMA Channel 1 Receive Process State This field indicates the Rx DMA FSM state for Channel 1.
RPS1	DMA Channel 1 Receive Process State
	This field indicates the Rx DMA FSM state for Channel 1.
	The MSB of this field always returns 0. This field does not generate an interrupt.
	0000 - Stopped (Reset or Stop Receive Command issued)
	0001 - Running (Fetching Rx Transfer Descriptor)
	0010 - Reserved for future use
	0011 - Running (Waiting for Rx packet)
	0100 - Suspended (Rx Descriptor Unavailable)
	0101 - Running (Closing the Rx Descriptor)
	0110 - Timestamp write state
	0111 - Running (Transferring the received packet data from the Rx buffer to the system memory)
15-12	DMA Channel 0 Transmit Process State This field indicates the Tx DMA FSM state for Channel 0.
TPS0	DMA Channel 0 Transmit Process State
	This field indicates the Tx DMA FSM state for Channel 0.
	The MSB of this field always returns 0. This field does not generate an interrupt.
	0000 - Stopped (Reset or Stop Transmit Command issued)
	0001 - Running (Fetching Tx Transfer Descriptor)
	0010 - Running (Waiting for status)
	0011 - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO))
	0100 - Timestamp write state
	0101 - Reserved for future use
	0110 - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow)
	0111 - Running (Closing Tx Descriptor)
11-8	DMA Channel 0 Receive Process State This field indicates the Rx DMA FSM state for Channel 0.

Table continues on the next page...

Field	Description
RPS0	DMA Channel 0 Receive Process State This field indicates the Rx DMA FSM state for Channel 0. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Receive Command issued) 0001 - Running (Fetching Rx Transfer Descriptor) 0010 - Reserved for future use 0011 - Running (Waiting for Rx packet) 0100 - Suspended (Rx Descriptor Unavailable) 0101 - Running (Closing the Rx Descriptor) 0110 - Timestamp write state 0111 - Running (Transferring the received packet data from the Rx buffer to the system memory)
7-2 —	Reserved
1 AXRHSTS	AXI Master Read Channel Status When high, this bit indicates that the read channel of the AXI master is active, and it is transferring the data. 0 - AXI Master Read Channel Status not detected 1 - AXI Master Read Channel Status detected
0 AXWHSTS	AXI Master Write Channel When high, this bit indicates that the write channel of the AXI master is active, and it is transferring data. 0 - AXI Master Write Channel or AHB Master Status not detected 1 - AXI Master Write Channel or AHB Master Status detected

11.7.6.1.463 DMA Debug Status 1 (DMA_DEBUG_STATUS1)

The Debug Status1 register gives the Receive and Transmit process status for DMA Channel 3-Channel 6.

11.7.6.1.463.1 Offset

Register	Offset
DMA_DEBUG_STATUS1	1010h

11.7.6.1.463.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				Reserved				Reserved				Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TPS4				RPS4				TPS3				RPS3			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.463.3 Fields

Field	Description
31-28 —	Reserved
27-24 —	Reserved
23-20 —	Reserved
19-16 —	Reserved
15-12 TPS4	DMA Channel 4 Transmit Process State This field indicates the Tx DMA FSM state for Channel 4. DMA Channel 4 Transmit Process State This field indicates the Tx DMA FSM state for Channel 4. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Transmit Command issued) 0001 - Running (Fetching Tx Transfer Descriptor) 0010 - Running (Waiting for status) 0011 - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0100 - Timestamp write state 0101 - Reserved for future use 0110 - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0111 - Running (Closing Tx Descriptor)
11-8 RPS4	DMA Channel 4 Receive Process State This field indicates the Rx DMA FSM state for Channel 4. DMA Channel 4 Receive Process State This field indicates the Rx DMA FSM state for Channel 4. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Receive Command issued)

Table continues on the next page...

Field	Description
	0001 - Running (Fetching Rx Transfer Descriptor) 0010 - Reserved for future use 0011 - Running (Waiting for Rx packet) 0100 - Suspended (Rx Descriptor Unavailable) 0101 - Running (Closing the Rx Descriptor) 0110 - Timestamp write state 0111 - Running (Transferring the received packet data from the Rx buffer to the system memory)
7-4 TPS3	DMA Channel 3 Transmit Process State This field indicates the Tx DMA FSM state for Channel 3. DMA Channel 3 Transmit Process State This field indicates the Tx DMA FSM state for Channel 3. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Transmit Command issued) 0001 - Running (Fetching Tx Transfer Descriptor) 0010 - Running (Waiting for status) 0011 - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0100 - Timestamp write state 0101 - Reserved for future use 0110 - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0111 - Running (Closing Tx Descriptor)
3-0 RPS3	DMA Channel 3 Receive Process State This field indicates the Rx DMA FSM state for Channel 3. DMA Channel 3 Receive Process State This field indicates the Rx DMA FSM state for Channel 3. The MSB of this field always returns 0. This field does not generate an interrupt. 0000 - Stopped (Reset or Stop Receive Command issued) 0001 - Running (Fetching Rx Transfer Descriptor) 0010 - Reserved for future use 0011 - Running (Waiting for Rx packet) 0100 - Suspended (Rx Descriptor Unavailable) 0101 - Running (Closing the Rx Descriptor) 0110 - Timestamp write state 0111 - Running (Transferring the received packet data from the Rx buffer to the system memory)

11.7.6.1.464 AXI LPI Entry Interval Control (DMA_AXI_LPI_ENTRY_INTERVAL)

This register is used to control the AXI LPI entry interval.

11.7.6.1.464.1 Offset

Register	Offset
DMA_AXI_LPI_ENTRY_INTERVAL	1040h

11.7.6.1.464.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												LPIEI			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.464.3 Fields

Field	Description
31-4 —	Reserved
3-0 LPIEI	LPI Entry Interval Contains the number of system clock cycles, multiplied by 64, to wait for an activity in the DWC_ether_qos to enter into the AXI low power state 0 indicates 64 clock cycles

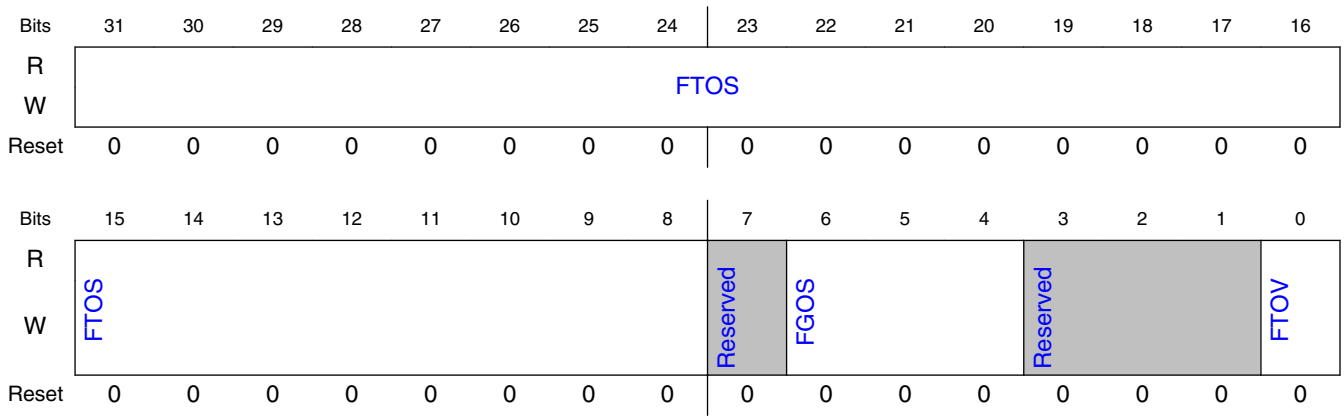
11.7.6.1.465 TBS Control (DMA_TBS_CTRL)

This register is used to control the TBS attributes.

11.7.6.1.465.1 Offset

Register	Offset
DMA_TBS_CTRL	1050h

11.7.6.1.465.2 Diagram



11.7.6.1.465.3 Fields

Field	Description
31-8 FTOS	Fetch Time Offset The value in units of 256 nanoseconds, that has to be deducted from the Launch time to compute the Fetch Time. Fetch Time Offset The value in units of 256 nanoseconds, that has to be deducted from the Launch time to compute the Fetch Time. Max value: 999,999,999 ns, additionally should be smaller than CTR-1 value when ESTM mode is set since this value is a modulo CTR value.
7 —	Reserved
6-4 FGOS	Fetch GSN Offset The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Fetch GSN Offset The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Value valid only when FTOV is set.
3-1 —	Reserved
0 FTOV	Fetch Time Offset Valid When set indicates the FTOS field is valid. Fetch Time Offset Valid When set indicates the FTOS field is valid. When not set, indicates the Fetch Offset is not valid and the DMA engine can fetch the frames from host memory without any time restrictions. 0 - Fetch Time Offset is invalid 1 - Fetch Time Offset is valid

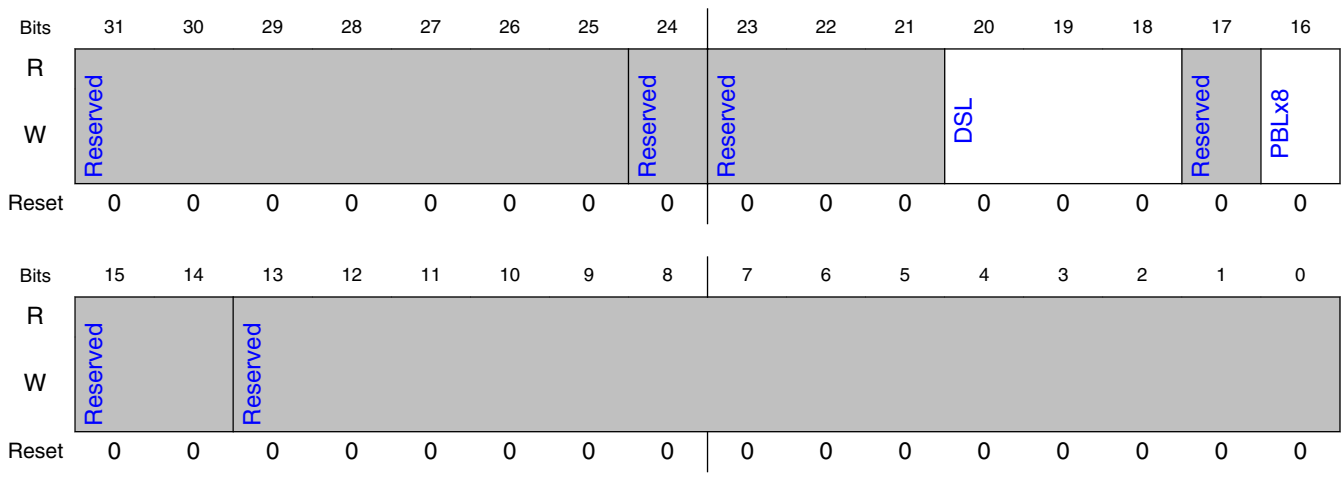
11.7.6.1.466 DMA Channel 0 Control (DMA_CH0_CONTROL)

The DMA Channel 0 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

11.7.6.1.466.1 Offset

Register	Offset
DMA_CH0_CONTROL	1100h

11.7.6.1.466.2 Diagram



11.7.6.1.466.3 Fields

Field	Description
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length

Table continues on the next page...

Field	Description
	This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16 PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH0_TX_CONTROL and Bits[21:16] in DMA_CH0_RX_CONTROL is multiplied by eight times. 8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH0_TX_CONTROL and Bits[21:16] in DMA_CH0_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0 - 8xPBL mode is disabled 1 - 8xPBL mode is enabled
15-14 —	Reserved
13-0 —	Reserved

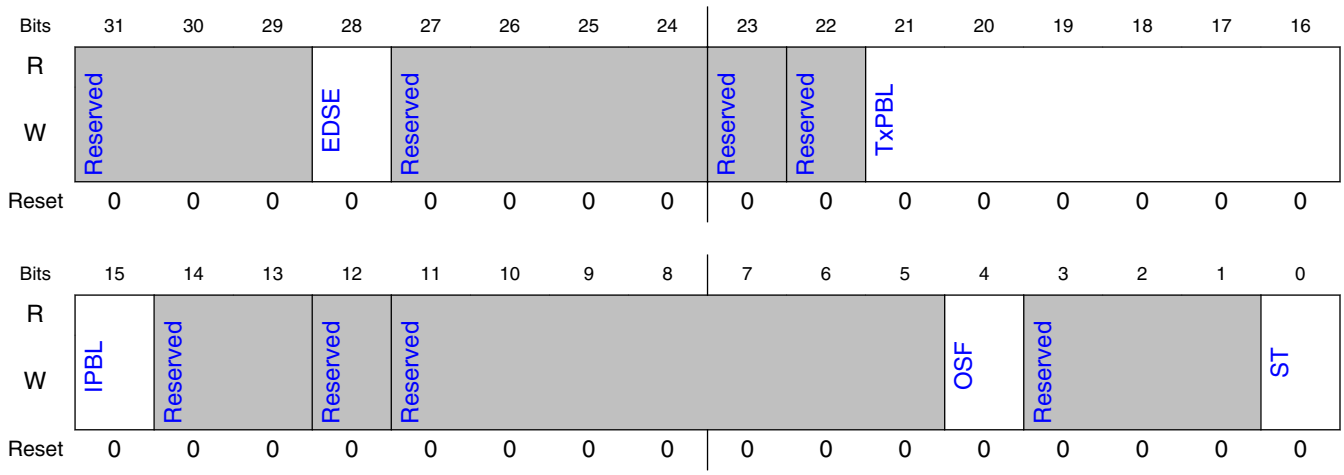
11.7.6.1.467 DMA Channel 0 Transmit Control (DMA_CH0_TX_CONTROL)

The DMA Channel 0 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

11.7.6.1.467.1 Offset

Register	Offset
DMA_CH0_TX_CONTROL	1104h

11.7.6.1.467.2 Diagram



11.7.6.1.467.3 Fields

Field	Description
31-29 —	Reserved
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0 - Enhanced Descriptor is disabled 1 - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in CONTROL register.

Table continues on the next page...

Field	Description
	2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ[n]_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0 - Ignore PBL Requirement is disabled 1 - Ignore PBL Requirement is enabled
14-13 —	Reserved
12 —	Reserved
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0 - Operate on Second Packet disabled 1 - Operate on Second Packet enabled
3-1 —	Reserved
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH0_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH0_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to

Field	Description
	program TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0 - Stop Transmission Command 1 - Start Transmission Command

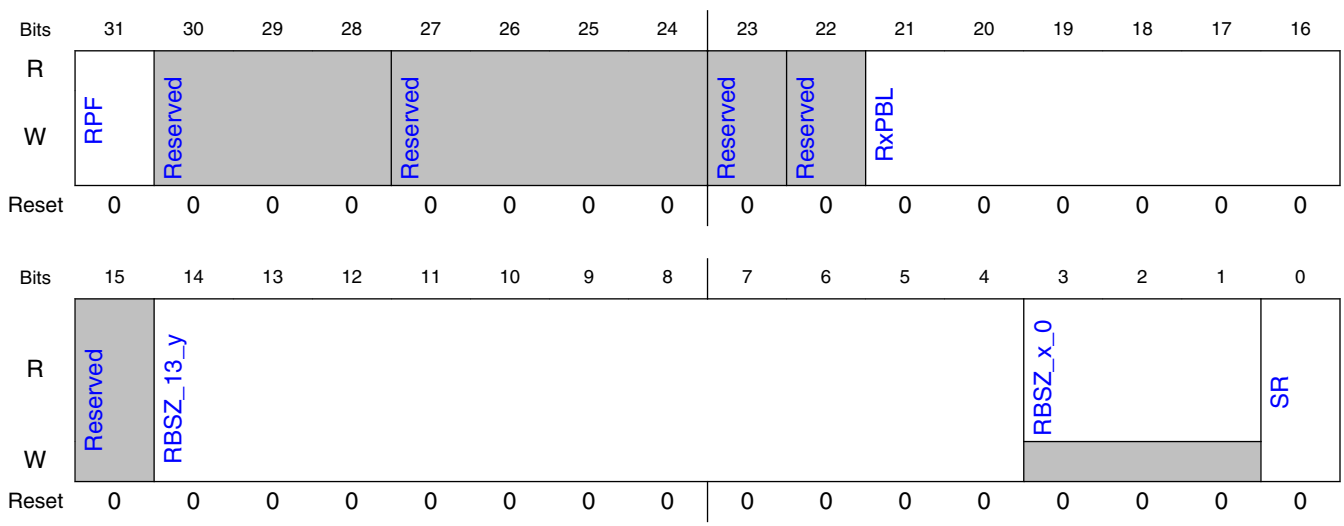
11.7.6.1.468 DMA Channel 0 Receive Control (DMA_CH0_RX_CONTROL)

The DMA Channel 0 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

11.7.6.1.468.1 Offset

Register	Offset
DMA_CH0_RX_CONTROL	1108h

11.7.6.1.468.2 Diagram



11.7.6.1.468.3 Fields

Field	Description
31 RPF	Rx Packet Flush. Rx Packet Flush.

Table continues on the next page...

Field	Description
	When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0 - Rx Packet Flush is disabled 1 - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the DMA_CH0_CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ[n]_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0.

Table continues on the next page...

Field	Description
RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH0_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH0_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH0_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0 - Stop Receive 1 - Start Receive

11.7.6.1.469 Channel 0 Tx Descriptor List Address register (DMA_CH0_TXDESC_LIST_ADDRESS)

The Channel 0 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

11.7.6.1.469.1 Offset

Register	Offset
DMA_CH0_TXDESC_LIST_ADDRESS	1114h

11.7.6.1.469.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDESLA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDESLA													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.469.3 Fields

Field	Description
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

11.7.6.1.470 Channel 0 Rx Descriptor List Address register (DMA_CH0_RXDESC_LIST_ADDRESS)

The Channel 0 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

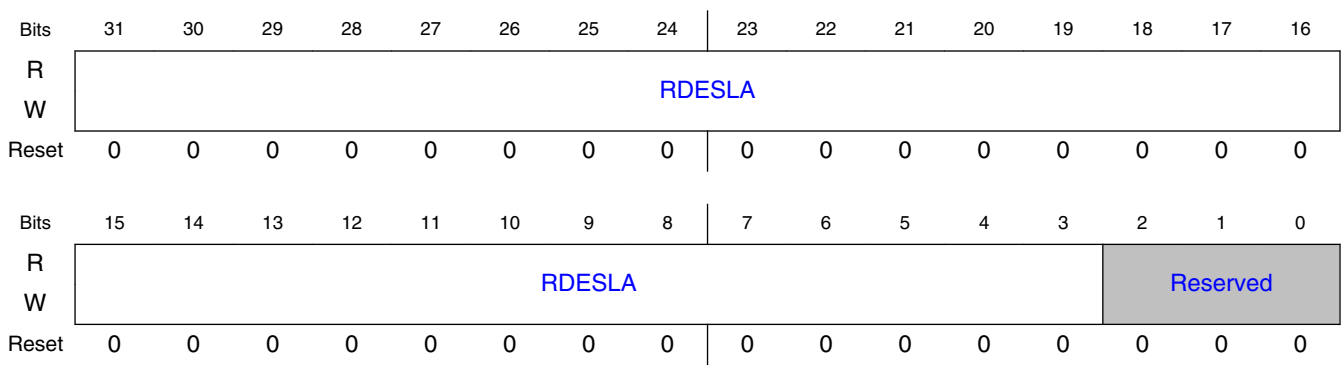
This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

11.7.6.1.470.1 Offset

Register	Offset
DMA_CH0_RXDESC_LIST_ADDRESS	111Ch

11.7.6.1.470.2 Diagram



11.7.6.1.470.3 Fields

Field	Description
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration

Table continues on the next page...

Field	Description
	- 31:4 for 128-bit configuration
2-0 —	Reserved

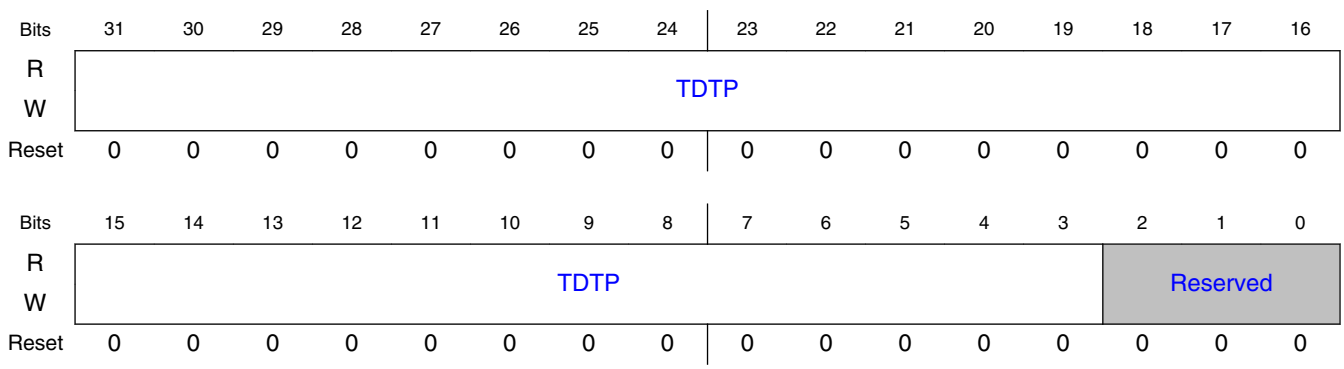
11.7.6.1.471 Channel 0 Tx Descriptor Tail Pointer (DMA_CH0_TXDESC_TAIL_POINTER)

The Channel 0 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

11.7.6.1.471.1 Offset

Register	Offset
DMA_CH0_TXDESC_TAIL_POINTER	1120h

11.7.6.1.471.2 Diagram



11.7.6.1.471.3 Fields

Field	Description
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration

Table continues on the next page...

Field	Description
	- 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

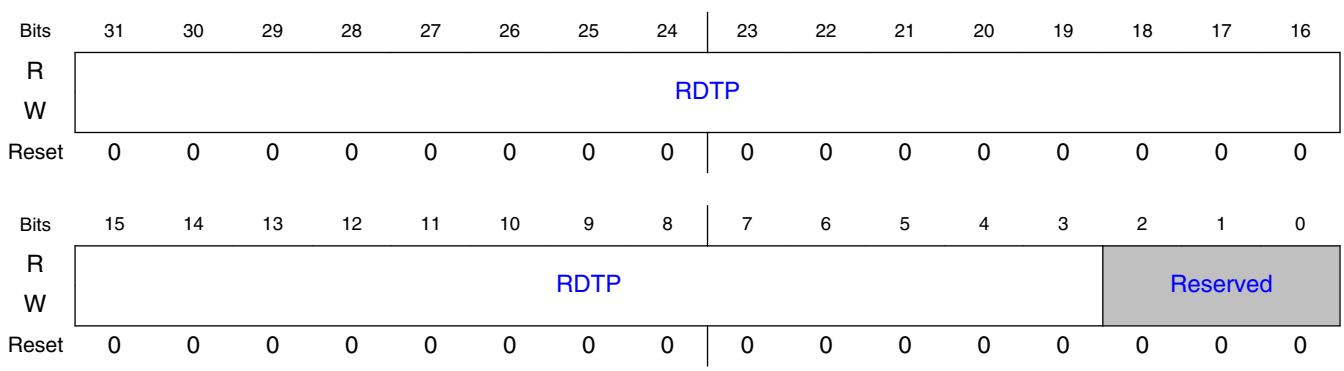
11.7.6.1.472 Channel 0 Rx Descriptor Tail Pointer (DMA_CH0_RXDESC_TAIL_POINTER)

The Channel 0 Rx Descriptor Tail Pointer points to an offset from the base and indicates the location of the last valid descriptor.

11.7.6.1.472.1 Offset

Register	Offset
DMA_CH0_RXDESC_TAIL_POINTER	1128h

11.7.6.1.472.2 Diagram



11.7.6.1.472.3 Fields

Field	Description
31-3 RDTP	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration:

Table continues on the next page...

Field	Description
	- 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

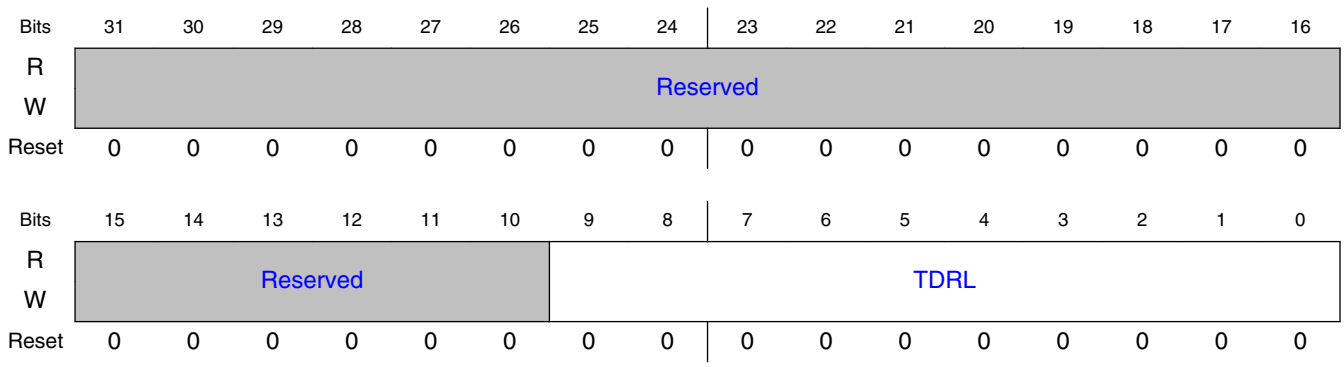
11.7.6.1.473 Channel 0 Tx Descriptor Ring Length (DMA_CH0_TXDESC_RING_LENGTH)

The Channel 0 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

11.7.6.1.473.1 Offset

Register	Offset
DMA_CH0_TXDESC_RING_LENGTH	112Ch

11.7.6.1.473.2 Diagram



11.7.6.1.473.3 Fields

Field	Description
31-10 —	Reserved
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring.

Field	Description
	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. Synopsys recommends a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

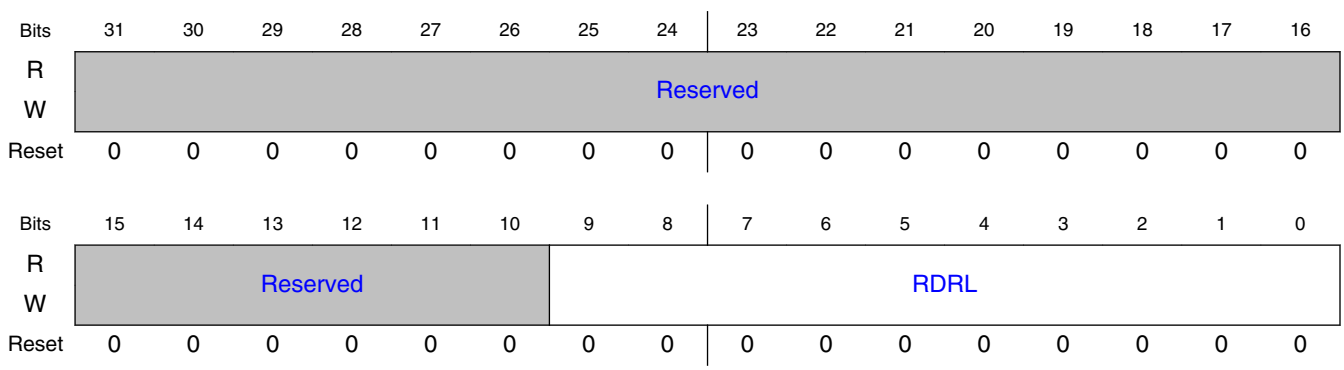
11.7.6.1.474 Channel 0 Rx Descriptor Ring Length (DMA_CH0_RXDESC_RING_LENGTH)

The Channel 0 Rx Descriptor Ring Length register contains the length of the Receive descriptor circular ring.

11.7.6.1.474.1 Offset

Register	Offset
DMA_CH0_RXDESC_RING_LENGTH	1130h

11.7.6.1.474.2 Diagram



11.7.6.1.474.3 Fields

Field	Description
31-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length

Field	Description
	This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

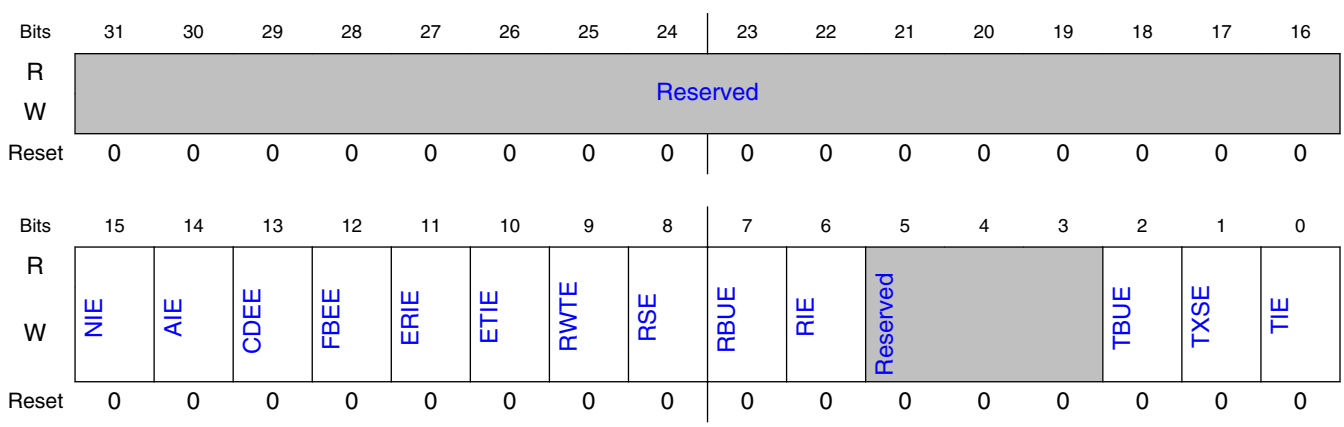
11.7.6.1.475 Channel 0 Interrupt Enable (DMA_CH0_INTERRUPT_ENABLE)

The Channel 0 Interrupt Enable register enables the interrupts reported by the Status register.

11.7.6.1.475.1 Offset

Register	Offset
DMA_CH0_INTERRUPT_ENABLE	1134h

11.7.6.1.475.2 Diagram



11.7.6.1.475.3 Fields

Field	Description
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable

Table continues on the next page...

Field	Description
	When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0 - Normal Interrupt Summary is disabled 1 - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled. 0 - Abnormal Interrupt Summary is disabled 1 - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0 - Context Descriptor Error is disabled 1 - Context Descriptor Error is enabled
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0 - Fatal Bus Error is disabled 1 - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled.

Table continues on the next page...

Field	Description
	0 - Early Receive Interrupt is disabled 1 - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0 - Early Transmit Interrupt is disabled 1 - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0 - Receive Watchdog Timeout is disabled 1 - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0 - Receive Stopped is disabled 1 - Receive Stopped is enabled
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0 - Receive Buffer Unavailable is disabled 1 - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0 - Receive Interrupt is disabled 1 - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled.

Table continues on the next page...

Field	Description
	0 - Transmit Buffer Unavailable is disabled 1 - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0 - Transmit Stopped is disabled 1 - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0 - Transmit Interrupt is disabled 1 - Transmit Interrupt is enabled

11.7.6.1.476 Channel 0 Receive Interrupt Watchdog Timer (DMA_CH0_RX_INTERRUPT_WATCHDOG_TIMER)

The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA.

The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the STATUS register.

11.7.6.1.476.1 Offset

Register	Offset
DMA_CH0_RX_INTERRUPT_WATCHDOG_TIMER	1138h

11.7.6.1.476.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														RWTU	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RWT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.476.3 Fields

Field	Description
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024 - 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15-8 —	Reserved
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

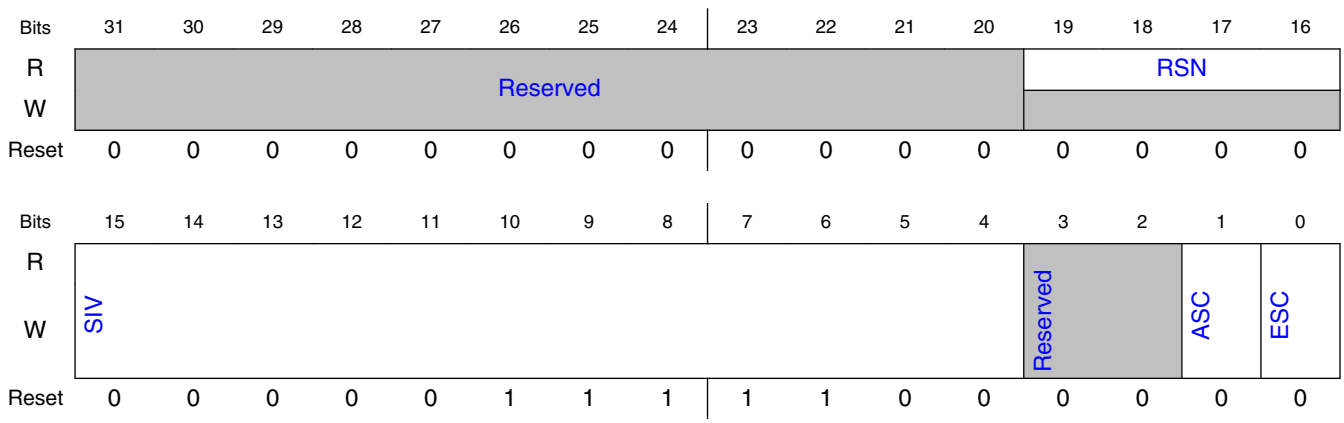
11.7.6.1.477 Channel 0 Slot Function Control and Status (DMA_CH0_SLOT_FUNCTION_CONTROL_STATUS)

The Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

11.7.6.1.477.1 Offset

Register	Offset
DMA_CH0_SLOT_FUNCTION_CONTROL_STATUS	113Ch

11.7.6.1.477.2 Diagram



11.7.6.1.477.3 Fields

Field	Description
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us

Table continues on the next page...

Field	Description
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0 - Advance Slot Check is disabled 1 - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or - ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0 - Slot Comparison is disabled 1 - Slot Comparison is enabled

11.7.6.1.478 Channel 0 Current Application Transmit Descriptor (DMA_CH0_CURRENT_APP_TXDESC)

The Channel 0 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

11.7.6.1.478.1 Offset

Register	Offset
DMA_CH0_CURRENT_APP_TXDESC	1144h

11.7.6.1.478.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.478.3 Fields

Field	Description
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

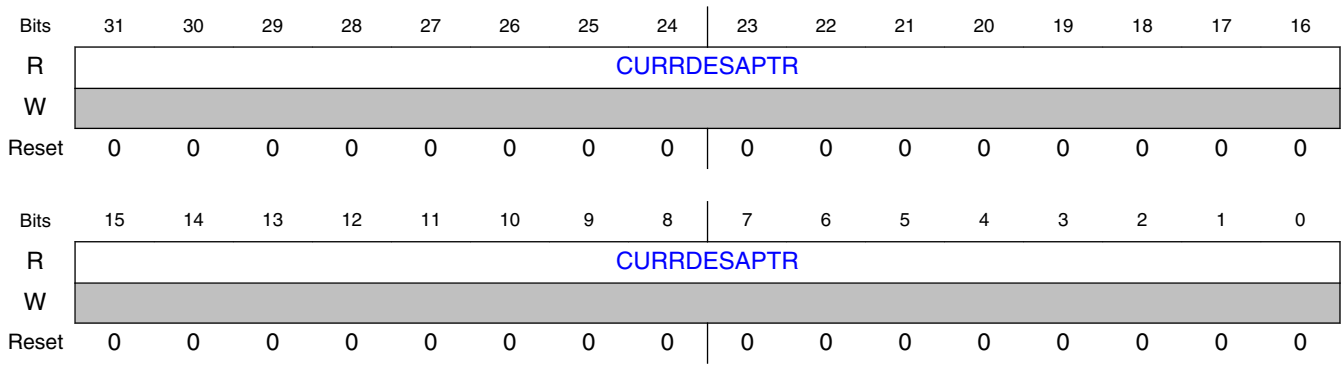
11.7.6.1.479 Channel 0 Current Application Receive Descriptor (DMA_CH0_CURRENT_APP_RXDESC)

The Channel 0 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

11.7.6.1.479.1 Offset

Register	Offset
DMA_CH0_CURRENT_APP_RXDESC	114Ch

11.7.6.1.479.2 Diagram



11.7.6.1.479.3 Fields

Field	Description
31-0 CURRDESAPTR	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

11.7.6.1.480 Channel 0 Current Application Transmit Buffer Address (DMA_CH0_CURRENT_APP_TXBUFFER)

The Channel 0 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

11.7.6.1.480.1 Offset

Register	Offset
DMA_CH0_CURRENT_APP_TXBUFFER	1154h

11.7.6.1.480.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.480.3 Fields

Field	Description
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

11.7.6.1.481 Channel 0 Current Application Receive Buffer Address (DMA_CH0_CURRENT_APP_RXBUFFER)

The Channel 0 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

11.7.6.1.481.1 Offset

Register	Offset
DMA_CH0_CURRENT_APP_RXBUFFER	115Ch

11.7.6.1.481.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.481.3 Fields

Field	Description
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

11.7.6.1.482 DMA Channel 0 Status (DMA_CH0_STATUS)

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

11.7.6.1.482.1 Offset

Register	Offset
DMA_CH0_STATUS	1160h

11.7.6.1.482.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										REB			TEB		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.482.3 Fields

Field	Description
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access

Table continues on the next page...

Field	Description
	- Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH0_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH0_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Normal Interrupt Summary status not detected 1 - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH0_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Abnormal Interrupt Summary status not detected 1 - Abnormal Interrupt Summary status detected

Table continues on the next page...

Field	Description
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all ones descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all ones descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Context Descriptor Error status not detected 1 - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Fatal Bus Error status not detected 1 - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Receive Interrupt status not detected 1 - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Transmit Interrupt status not detected 1 - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0 - Receive Watchdog Timeout status not detected

Table continues on the next page...

Field	Description
	1 - Receive Watchdog Timeout status detected
8 RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Process Stopped status not detected 1 - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode, the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Buffer Unavailable status not detected 1 - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Interrupt status not detected 1 - Receive Interrupt status detected
5-3 —	Reserved
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Buffer Unavailable status not detected

Table continues on the next page...

Field	Description
	1 - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Process Stopped status not detected 1 - Transmit Process Stopped status detected
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Interrupt status not detected 1 - Transmit Interrupt status detected

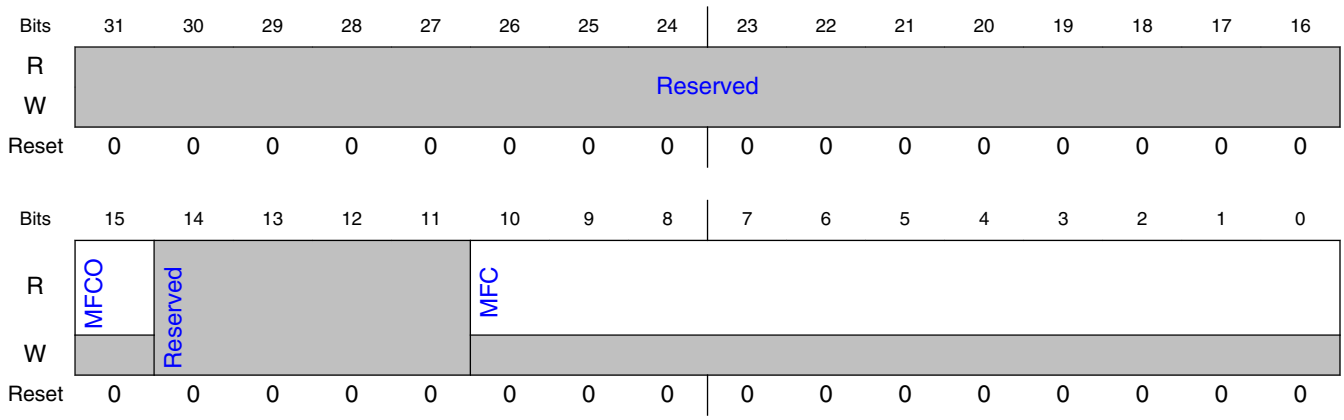
11.7.6.1.483 Channel 0 Missed Frame Counter (DMA_CH0_MISS_FRAME_CNT)

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in RX_CONTROL register.

11.7.6.1.483.1 Offset

Register	Offset
DMA_CH0_MISS_FRAME_CNT	1164h

11.7.6.1.483.2 Diagram



11.7.6.1.483.3 Fields

Field	Description
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Miss Frame Counter overflow not occurred 1 - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH0_RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH0_RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

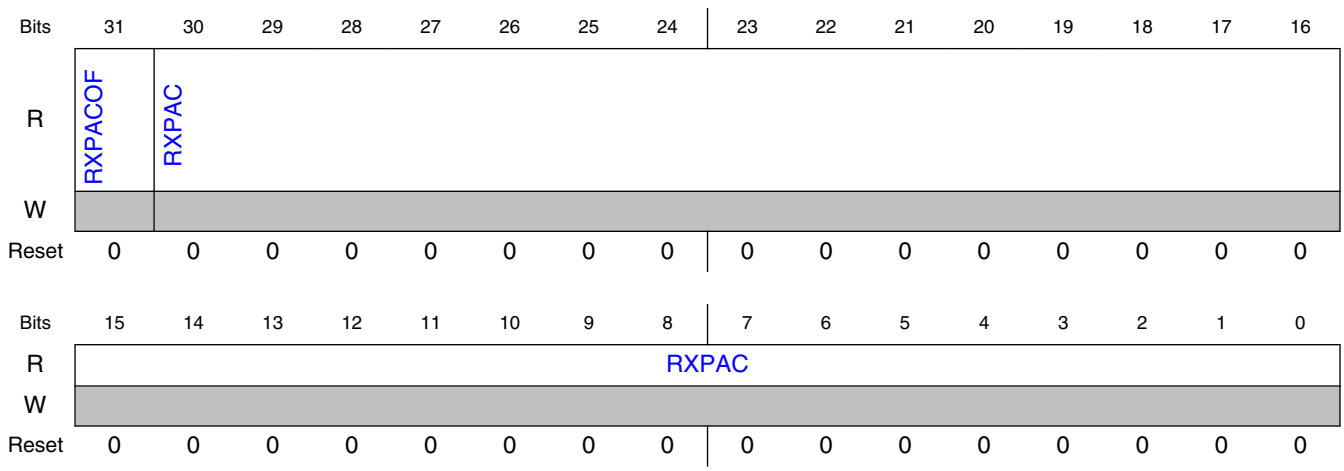
11.7.6.1.484 Channel 0 RXP Frames Accepted Counter (DMA_CH0_RXP_ACCEPT_CNT)

The RXP_ACCEPT_CNT register provides the count of the number of frames accepted by Rx Parser.

11.7.6.1.484.1 Offset

Register	Offset
DMA_CH0_RXP_ACCEPT_CNT	1168h

11.7.6.1.484.2 Diagram



11.7.6.1.484.3 Fields

Field	Description
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Accept Counter overflow not occurred 1 - Rx Parser Accept Counter overflow occurred
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

11.7.6.1.485 Channel 0 Receive ERI Counter (DMA_CH0_RX_ERI_CNT)

The DMA_CH0_RX_ERI_CNT register provides the count of the number of times ERI was asserted.

11.7.6.1.485.1 Offset

Register	Offset
DMA_CH0_RX_ERI_CNT	116Ch

11.7.6.1.485.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				ECNT											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.485.3 Fields

Field	Description
31-12 —	Reserved
11-0 ECNT	ERI Counter When ERIC bit of RX_CONTROL register is set, this counter increments for burst transfer completed by the Rx DMA from the start of packet transfer. ERI Counter When ERIC bit of RX_CONTROL register is set, this counter increments for burst transfer completed by the Rx DMA from the start of packet transfer. This counter will get reset at the start of new packet.

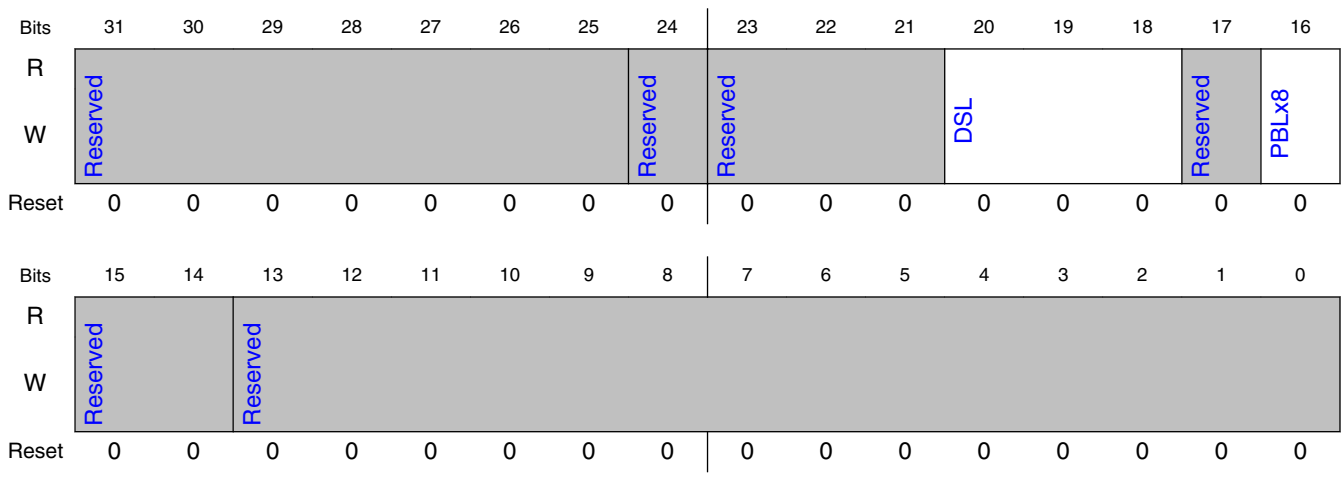
11.7.6.1.486 DMA Channel 1 Control (DMA_CH1_CONTROL)

The DMA Channel 1 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

11.7.6.1.486.1 Offset

Register	Offset
DMA_CH1_CONTROL	1180h

11.7.6.1.486.2 Diagram



11.7.6.1.486.3 Fields

Field	Description
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length

Table continues on the next page...

Field	Description
	This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16 PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in TX_CONTROL and Bits[21:16] in DMA_CH1_RX_CONTROL is multiplied by eight times. 8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in TX_CONTROL and Bits[21:16] in DMA_CH1_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0 - 8xPBL mode is disabled 1 - 8xPBL mode is enabled
15-14 —	Reserved
13-0 —	Reserved

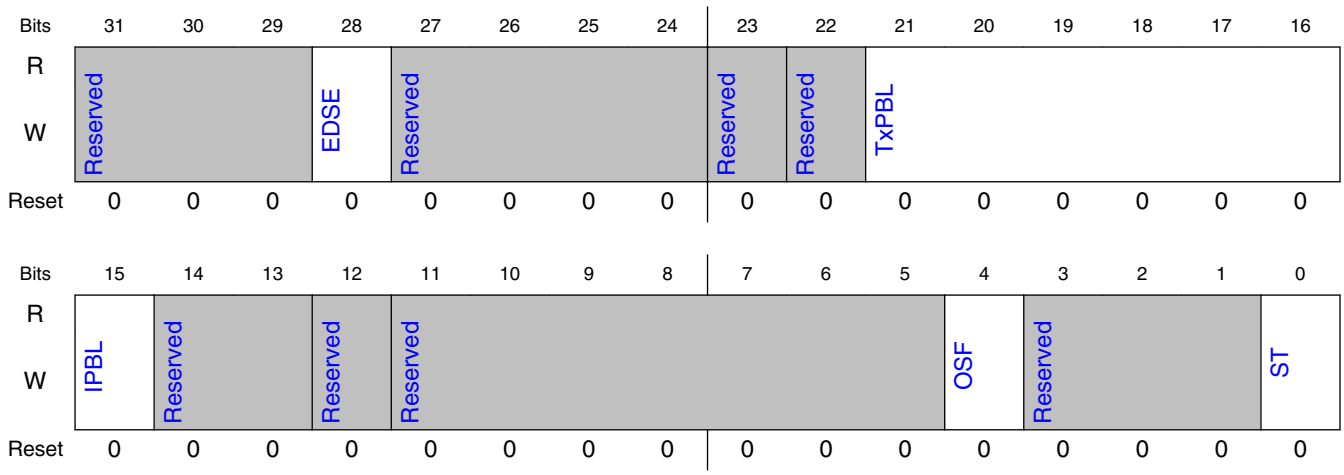
11.7.6.1.487 DMA Channel 1 Transmit Control (DMA_CH1_TX_CONTROL)

The DMA Channel 1 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

11.7.6.1.487.1 Offset

Register	Offset
DMA_CH1_TX_CONTROL	1184h

11.7.6.1.487.2 Diagram



11.7.6.1.487.3 Fields

Field	Description
31-29 —	Reserved
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0 - Enhanced Descriptor is disabled 1 - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in CONTROL register.

Table continues on the next page...

Field	Description
	2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ[n]_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0 - Ignore PBL Requirement is disabled 1 - Ignore PBL Requirement is enabled
14-13 —	Reserved
12 —	Reserved
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0 - Operate on Second Packet disabled 1 - Operate on Second Packet enabled
3-1 —	Reserved
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH1_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH1_STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH1_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to

Field	Description
	program TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0 - Stop Transmission Command 1 - Start Transmission Command

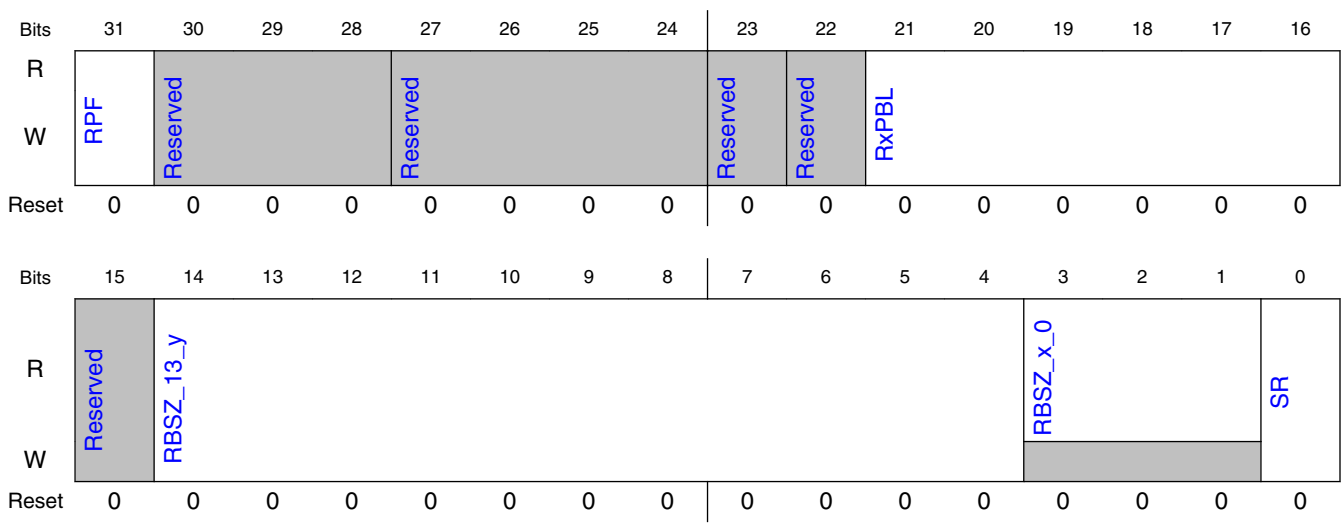
11.7.6.1.488 DMA Channel 1 Receive Control (DMA_CH1_RX_CONTROL)

The DMA Channel 1 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

11.7.6.1.488.1 Offset

Register	Offset
DMA_CH1_RX_CONTROL	1188h

11.7.6.1.488.2 Diagram



11.7.6.1.488.3 Fields

Field	Description
31 RPF	Rx Packet Flush. Rx Packet Flush.

Table continues on the next page...

Field	Description
	When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0 - Rx Packet Flush is disabled 1 - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ[n]_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0.

Table continues on the next page...

Field	Description
RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH1_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH1_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH1_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0 - Stop Receive 1 - Start Receive

11.7.6.1.489 Channel 1 Tx Descriptor List Address (DMA_CH1_TXDESC_LIST_ADDRESS)

The Channel 1 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in DMA_CH1_TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

11.7.6.1.489.1 Offset

Register	Offset
DMA_CH1_TXDESC_LIST_ADDRESS	1194h

11.7.6.1.489.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDESLA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDESLA													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.489.3 Fields

Field	Description
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

11.7.6.1.490 Channel 1 Rx Descriptor List Address (DMA_CH1_RXDESC_LIST_ADDRESS)

The Channel 1 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

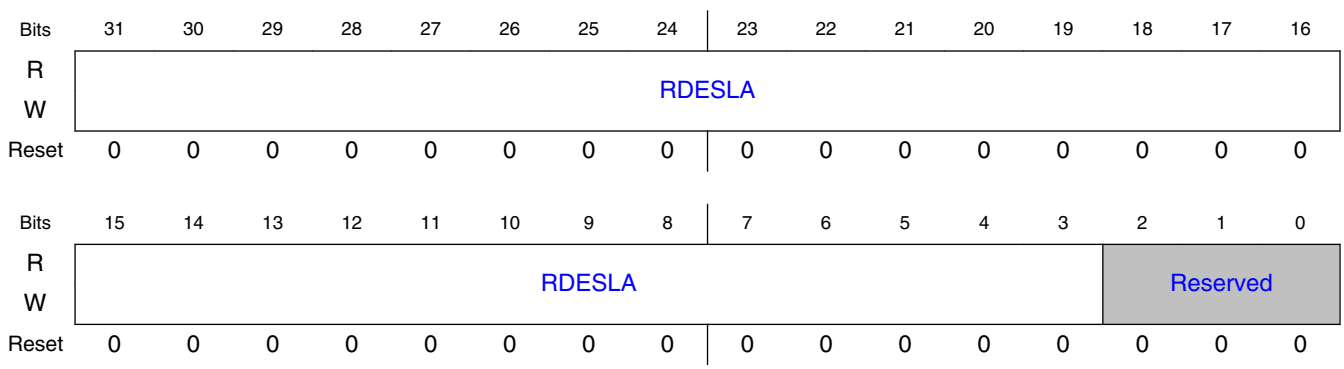
This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in DMA_CH1_RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

11.7.6.1.490.1 Offset

Register	Offset
DMA_CH1_RXDESC_LIST_ADDRESS	119Ch

11.7.6.1.490.2 Diagram



11.7.6.1.490.3 Fields

Field	Description
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration

Table continues on the next page...

Field	Description
	- 31:4 for 128-bit configuration
2-0 —	Reserved

11.7.6.1.491 Channel 1 Tx Descriptor Tail Pointer (DMA_CH1_TXDESC_TAIL_POINTER)

The Channel 1 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

11.7.6.1.491.1 Offset

Register	Offset
DMA_CH1_TXDESC_TAIL_POINTER	11A0h

11.7.6.1.491.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDTP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDTP													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.491.3 Fields

Field	Description
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration

Table continues on the next page...

Field	Description
	- 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

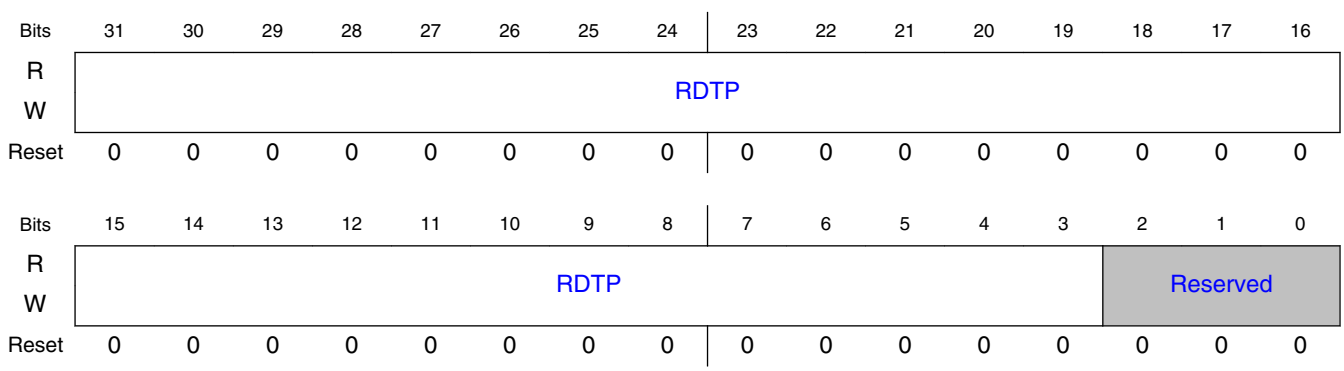
11.7.6.1.492 Channel 1 Rx Descriptor Tail Pointer (DMA_CH1_RXDESC_TAIL_POINTER)

The Channel 1 Rx Descriptor Tail Pointer points to an offset from the base and indicates the location of the last valid descriptor.

11.7.6.1.492.1 Offset

Register	Offset
DMA_CH1_RXDESC_TAIL_POINTER	11A8h

11.7.6.1.492.2 Diagram



11.7.6.1.492.3 Fields

Field	Description
31-3 RDTP	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration:

Table continues on the next page...

Field	Description
	- 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

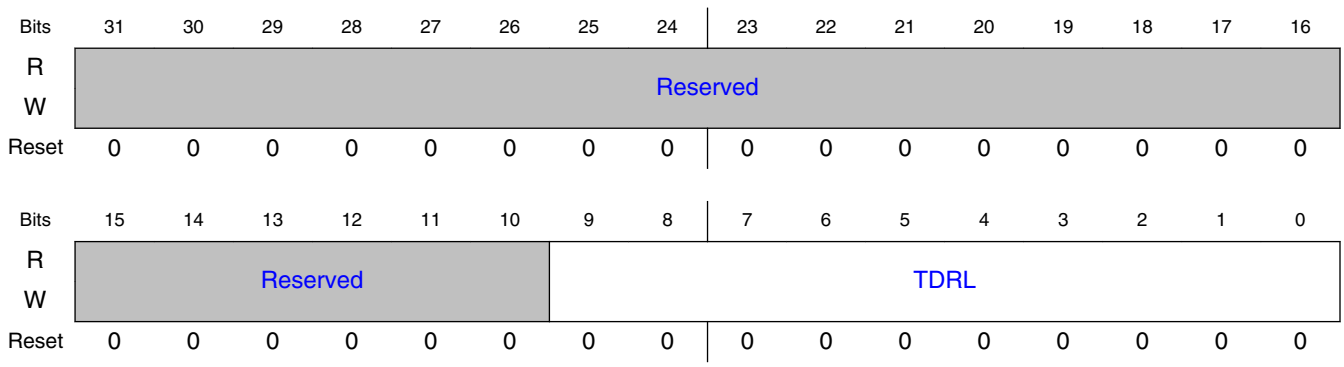
11.7.6.1.493 Channel 1 Tx Descriptor Ring Length (DMA_CH1_TXDESC_RING_LENGTH)

The Channel 1 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

11.7.6.1.493.1 Offset

Register	Offset
DMA_CH1_TXDESC_RING_LENGTH	11ACh

11.7.6.1.493.2 Diagram



11.7.6.1.493.3 Fields

Field	Description
31-10 —	Reserved
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring.

Field	Description
	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. Synopsys recommends a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

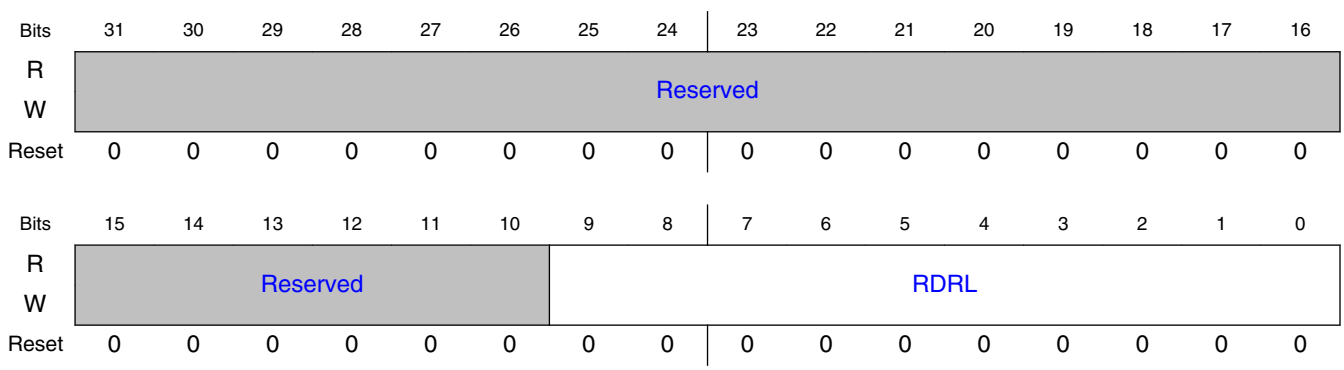
11.7.6.1.494 Channel 1 Rx Descriptor Ring Length (DMA_CH1_RXDESC_RING_LENGTH)

The Channel 1 Rx Descriptor Ring Length register contains the length of the Receive descriptor circular ring.

11.7.6.1.494.1 Offset

Register	Offset
DMA_CH1_RXDESC_RING_LENGTH	11B0h

11.7.6.1.494.2 Diagram



11.7.6.1.494.3 Fields

Field	Description
31-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length

Field	Description
	This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

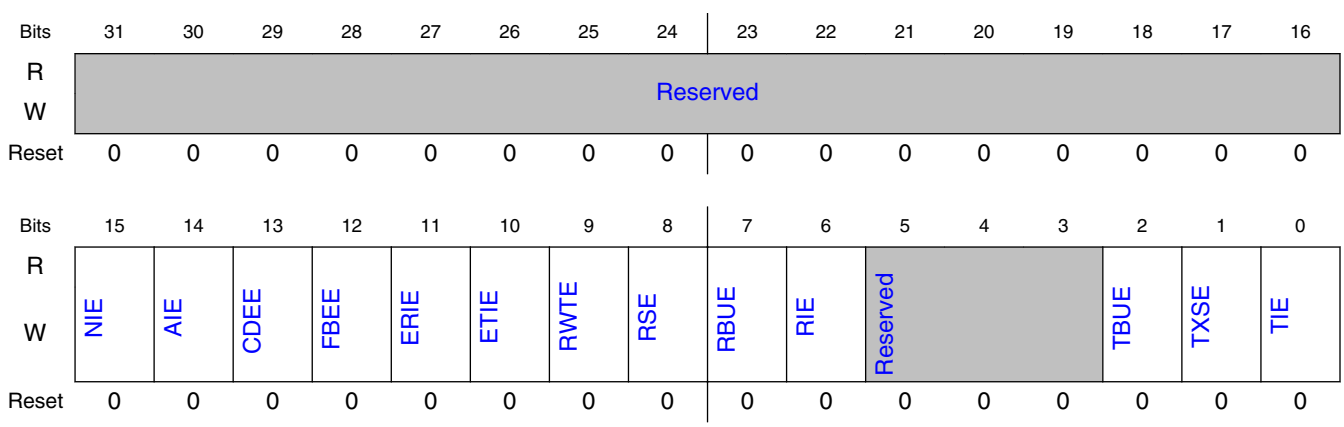
11.7.6.1.495 Channel 1 Interrupt Enable (DMA_CH1_INTERRUPT_ENABLE)

The Channel 1 Interrupt Enable register enables the interrupts reported by the Status register.

11.7.6.1.495.1 Offset

Register	Offset
DMA_CH1_INTERRUPT_ENABLE	11B4h

11.7.6.1.495.2 Diagram



11.7.6.1.495.3 Fields

Field	Description
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable

Table continues on the next page...

Field	Description
	When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0 - Normal Interrupt Summary is disabled 1 - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled. 0 - Abnormal Interrupt Summary is disabled 1 - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0 - Context Descriptor Error is disabled 1 - Context Descriptor Error is enabled
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0 - Fatal Bus Error is disabled 1 - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled.

Table continues on the next page...

Field	Description
	0 - Early Receive Interrupt is disabled 1 - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0 - Early Transmit Interrupt is disabled 1 - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0 - Receive Watchdog Timeout is disabled 1 - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0 - Receive Stopped is disabled 1 - Receive Stopped is enabled
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0 - Receive Buffer Unavailable is disabled 1 - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0 - Receive Interrupt is disabled 1 - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled.

Table continues on the next page...

Field	Description
	0 - Transmit Buffer Unavailable is disabled 1 - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0 - Transmit Stopped is disabled 1 - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0 - Transmit Interrupt is disabled 1 - Transmit Interrupt is enabled

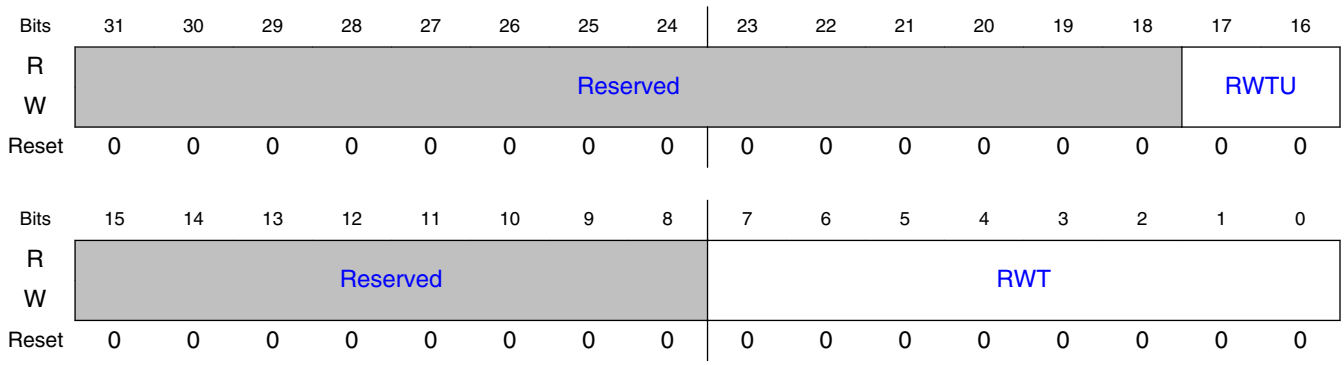
11.7.6.1.496 Channel 1 Receive Interrupt Watchdog Timer (DMA_CH1_RX_INTERRUPT_WATCHDOG_TIMER)

The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the STATUS register.

11.7.6.1.496.1 Offset

Register	Offset
DMA_CH1_RX_INTERRUPT_WATCHDOG_TIMER	11B8h

11.7.6.1.496.2 Diagram



11.7.6.1.496.3 Fields

Field	Description
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024 - 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15-8 —	Reserved
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

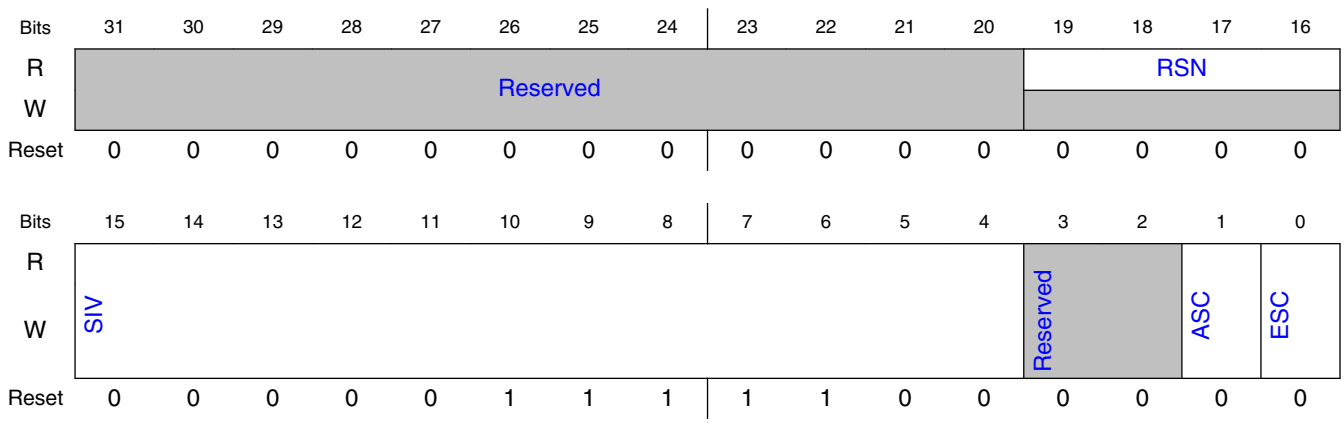
11.7.6.1.497 Channel 1 Slot Function Control and Status (DMA_CH1_SLOT_FUNCTION_CONTROL_STATUS)

The Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

11.7.6.1.497.1 Offset

Register	Offset
DMA_CH1_SLOT_FUNCTION_CONTROL_STATUS	11BCh

11.7.6.1.497.2 Diagram



11.7.6.1.497.3 Fields

Field	Description
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us

Table continues on the next page...

Field	Description
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0 - Advance Slot Check is disabled 1 - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or - ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0 - Slot Comparison is disabled 1 - Slot Comparison is enabled

11.7.6.1.498 Channel 1 Current Application Transmit Descriptor (DMA_CH1_CURRENT_APP_TXDESC)

The Channel 1 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

11.7.6.1.498.1 Offset

Register	Offset
DMA_CH1_CURRENT_APP_TXDESC	11C4h

11.7.6.1.498.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.498.3 Fields

Field	Description
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

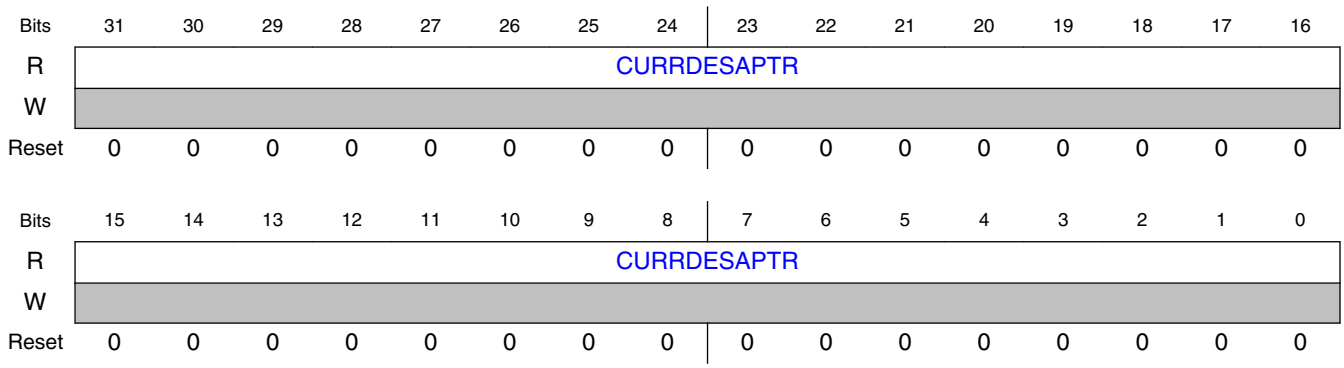
11.7.6.1.499 Channel 1 Current Application Receive Descriptor (DMA_CH1_CURRENT_APP_RXDESC)

The Channel 1 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

11.7.6.1.499.1 Offset

Register	Offset
DMA_CH1_CURRENT_APP_RXDESC	11CCh

11.7.6.1.499.2 Diagram



11.7.6.1.499.3 Fields

Field	Description
31-0 CURRDESAPTR	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

11.7.6.1.500 Channel 1 Current Application Transmit Buffer Address (DMA_CH1_CURRENT_APP_TXBUFFER)

The Channel 1 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

11.7.6.1.500.1 Offset

Register	Offset
DMA_CH1_CURRENT_APP_TXBUFFER	11D4h

11.7.6.1.500.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.500.3 Fields

Field	Description
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

11.7.6.1.501 Channel 1 Current Application Receive Buffer Address (DMA_CH1_CURRENT_APP_RXBUFFER)

The Channel 1 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

11.7.6.1.501.1 Offset

Register	Offset
DMA_CH1_CURRENT_APP_RXBUFFER	11DCh

11.7.6.1.501.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.501.3 Fields

Field	Description
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

11.7.6.1.502 DMA Channel 1 Status (DMA_CH1_STATUS)

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

11.7.6.1.502.1 Offset

Register	Offset
DMA_CH1_STATUS	11E0h

11.7.6.1.502.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										REB			TEB		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.502.3 Fields

Field	Description
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access

Table continues on the next page...

Field	Description
	- Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH1_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH1_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Normal Interrupt Summary status not detected 1 - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Abnormal Interrupt Summary status not detected 1 - Abnormal Interrupt Summary status detected

Table continues on the next page...

Field	Description
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Context Descriptor Error status not detected 1 - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Fatal Bus Error status not detected 1 - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Receive Interrupt status not detected 1 - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Transmit Interrupt status not detected 1 - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0 - Receive Watchdog Timeout status not detected

Table continues on the next page...

Field	Description
	1 - Receive Watchdog Timeout status detected
8 RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Process Stopped status not detected 1 - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode, the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Buffer Unavailable status not detected 1 - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Interrupt status not detected 1 - Receive Interrupt status detected
5-3 —	Reserved
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Buffer Unavailable status not detected

Table continues on the next page...

Field	Description
	1 - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Process Stopped status not detected 1 - Transmit Process Stopped status detected
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Interrupt status not detected 1 - Transmit Interrupt status detected

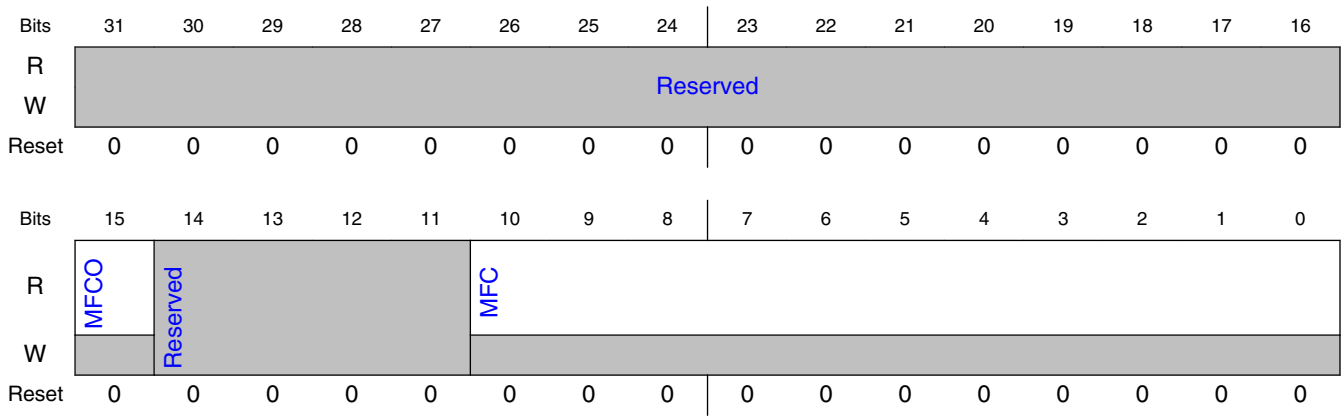
11.7.6.1.503 Channel 1 Missed Frame Counter (DMA_CH1_MISS_FRAME_CNT)

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in RX_CONTROL register.

11.7.6.1.503.1 Offset

Register	Offset
DMA_CH1_MISS_FRAME_CNT	11E4h

11.7.6.1.503.2 Diagram



11.7.6.1.503.3 Fields

Field	Description
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Miss Frame Counter overflow not occurred 1 - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH1_RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

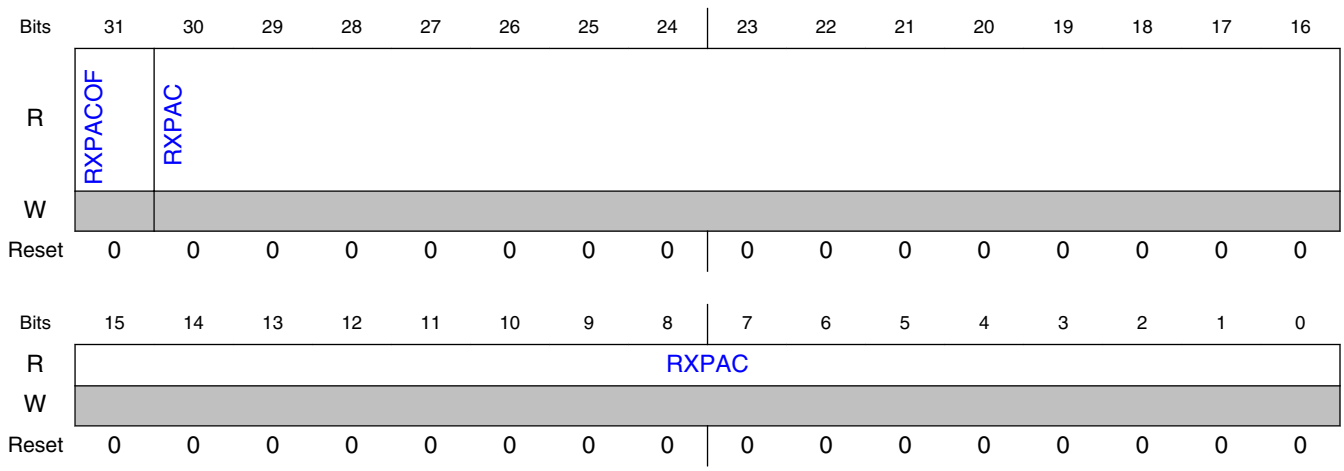
11.7.6.1.504 Channel 1 RXP Frames Accepted Counter (DMA_CH1_RXP_ACCEPT_CNT)

The Channel 1 RXP Frames Accepted register provides the count of the number of frames accepted by Rx Parser.

11.7.6.1.504.1 Offset

Register	Offset
DMA_CH1_RXP_ACCEPT_CNT	11E8h

11.7.6.1.504.2 Diagram



11.7.6.1.504.3 Fields

Field	Description
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Accept Counter overflow not occurred 1 - Rx Parser Accept Counter overflow occurred
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

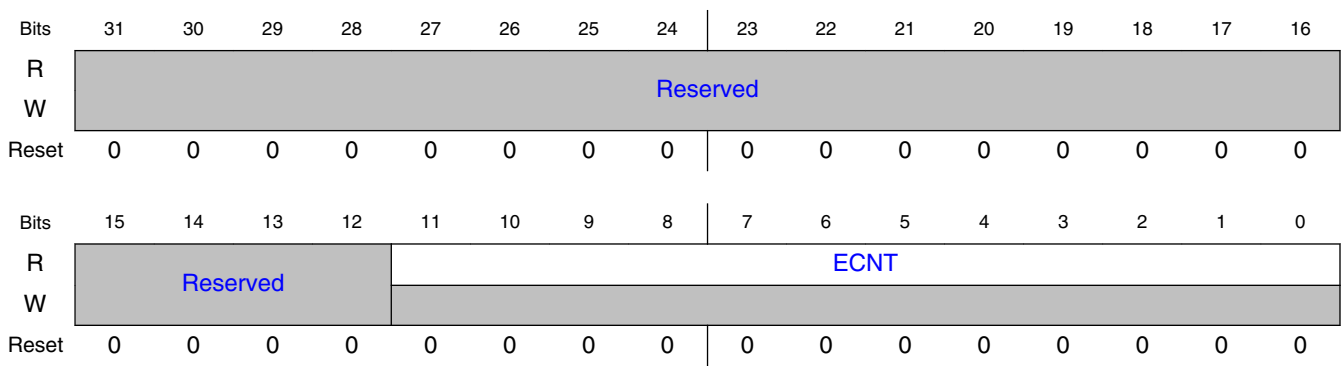
11.7.6.1.505 Channel 1 Receive ERI Counter (DMA_CH1_RX_ERI_CNT)

The Channel 1 Receive ERI Counter register provides the count of the number of times ERI was asserted.

11.7.6.1.505.1 Offset

Register	Offset
DMA_CH1_RX_ERI_CNT	11ECh

11.7.6.1.505.2 Diagram



11.7.6.1.505.3 Fields

Field	Description
31-12 —	Reserved
11-0 ECNT	ERI Counter When ERIC bit of RX_CONTROL register is set, this counter increments for burst transfer completed by the Rx DMA from the start of packet transfer. ERI Counter When ERIC bit of RX_CONTROL register is set, this counter increments for burst transfer completed by the Rx DMA from the start of packet transfer. This counter will get reset at the start of new packet.

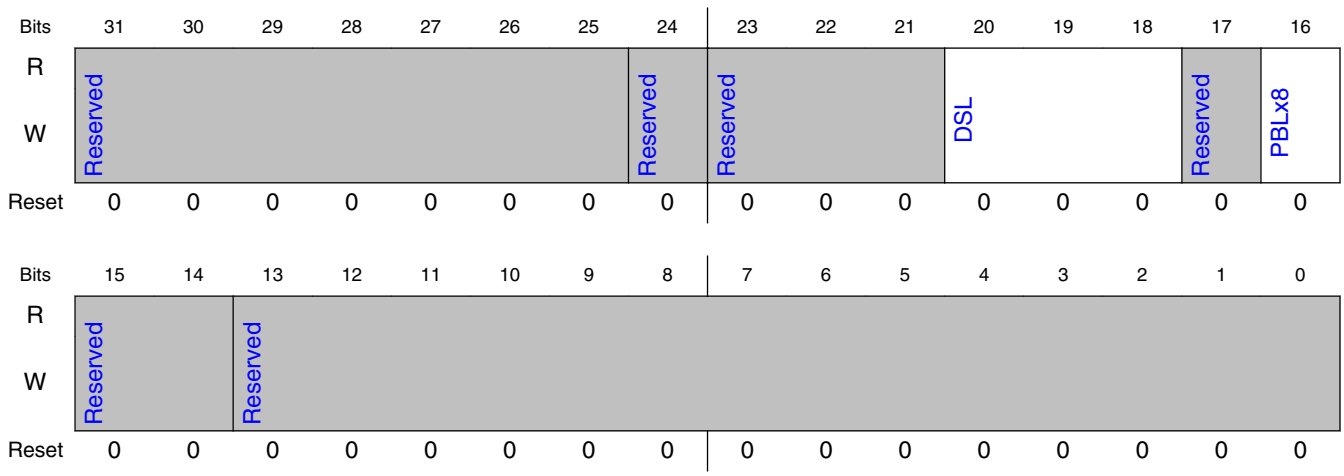
11.7.6.1.506 DMA Channel 2 Control (DMA_CH2_CONTROL)

The DMA Channel 2 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

11.7.6.1.506.1 Offset

Register	Offset
DMA_CH2_CONTROL	1200h

11.7.6.1.506.2 Diagram



11.7.6.1.506.3 Fields

Field	Description
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH2_TX_CONTROL and Bits[21:16] in DMA_CH2_RX_CONTROL is multiplied by eight times.

Table continues on the next page...

Field	Description
PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH2_TX_CONTROL and Bits[21:16] in DMA_CH2_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0 - 8xPBL mode is disabled 1 - 8xPBL mode is enabled
15-14 —	Reserved
13-0 —	Reserved

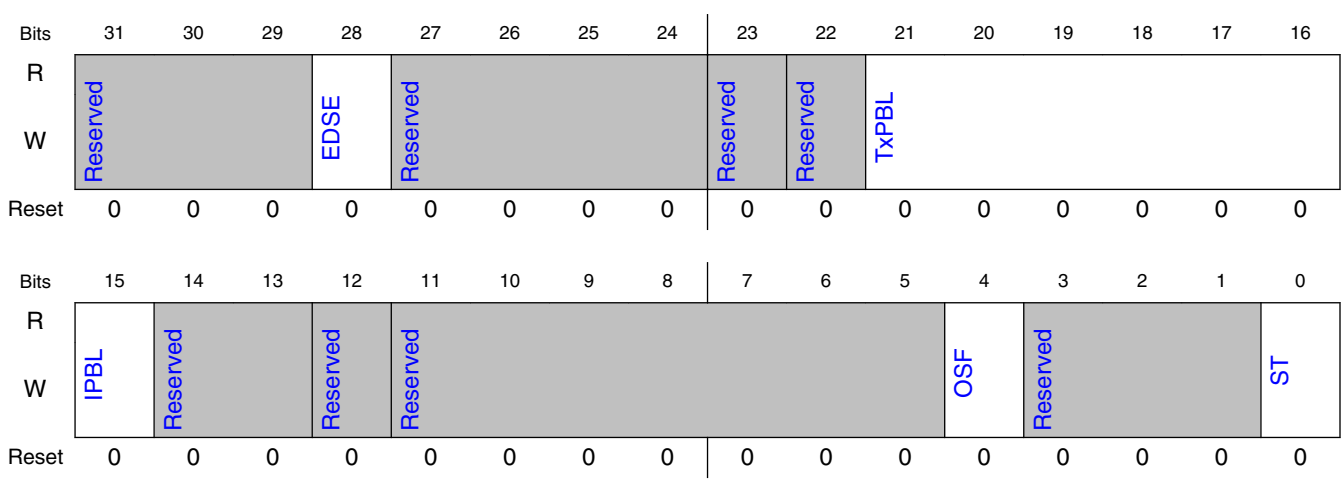
11.7.6.1.507 DMA Channel 2 Transmit Control (DMA_CH2_TX_CONTROL)

The DMA Channel 2 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

11.7.6.1.507.1 Offset

Register	Offset
DMA_CH2_TX_CONTROL	1204h

11.7.6.1.507.2 Diagram



11.7.6.1.507.3 Fields

Field	Description
31-29 —	Reserved
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0 - Enhanced Descriptor is disabled 1 - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in DMA_CH2_CONTROL register. 2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ[n]_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0 - Ignore PBL Requirement is disabled 1 - Ignore PBL Requirement is enabled
14-13	Reserved

Table continues on the next page...

Field	Description
—	
12 —	Reserved
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0 - Operate on Second Packet disabled 1 - Operate on Second Packet enabled
3-1 —	Reserved
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH2_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH2_STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH2_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program DMA_CH2_TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0 - Stop Transmission Command 1 - Start Transmission Command

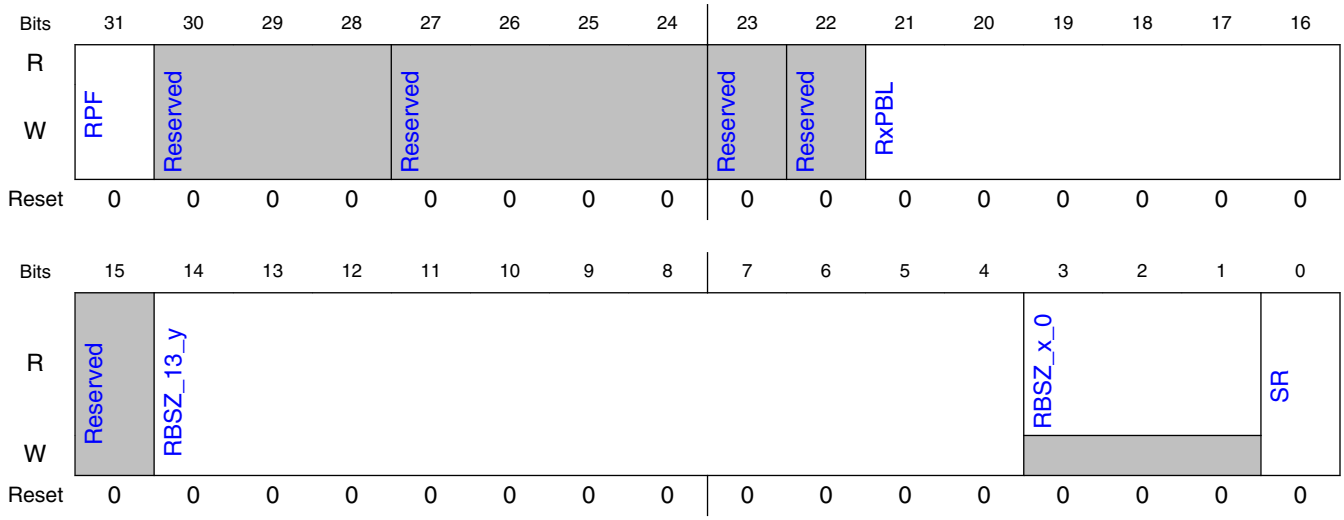
11.7.6.1.508 DMA Channel 2 Receive Control (DMA_CH2_RX_CONTROL)

The DMA Channel 2 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

11.7.6.1.508.1 Offset

Register	Offset
DMA_CH2_RX_CONTROL	1208h

11.7.6.1.508.2 Diagram



11.7.6.1.508.3 Fields

Field	Description
31 RPF	Rx Packet Flush. Rx Packet Flush. When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0 - Rx Packet Flush is disabled 1 - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer.

Table continues on the next page...

Field	Description
	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the DMA_CH2_CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ[n]_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1 RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH2_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH2_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH2_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable.

Field	Description
	When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0 - Stop Receive 1 - Start Receive

11.7.6.1.509 Channel 2 Tx Descriptor List Address (DMA_CH2_TXDESC_LIST_ADDRESS)

The Channel 2 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

11.7.6.1.509.1 Offset

Register	Offset
DMA_CH2_TXDESC_LIST_ADDRESS	1214h

11.7.6.1.509.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDESLA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDESLA													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.509.3 Fields

Field	Description
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

11.7.6.1.510 Channel 2 Rx Descriptor List Address (DMA_CH2_RXDESC_LIST_ADDRESS)

The Channel 2 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

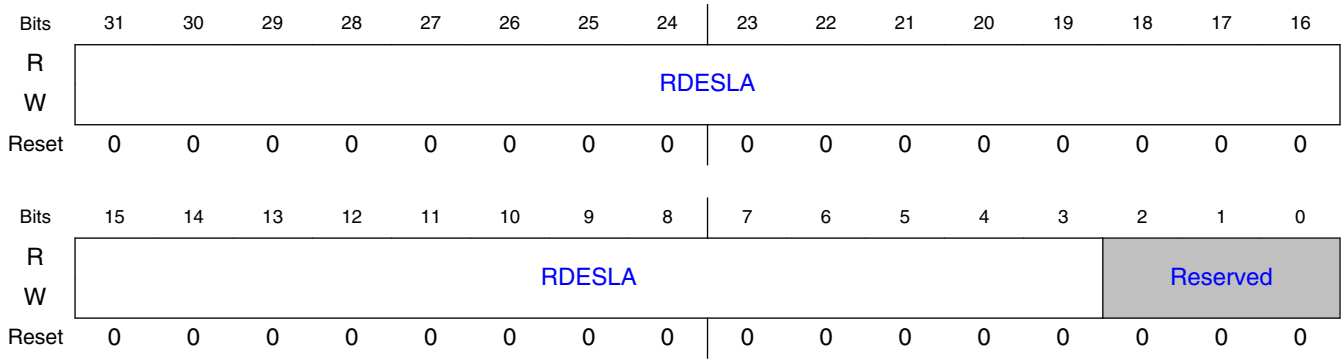
This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

11.7.6.1.510.1 Offset

Register	Offset
DMA_CH2_RXDESC_LIST_ADDRESS	121Ch

11.7.6.1.510.2 Diagram



11.7.6.1.510.3 Fields

Field	Description
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

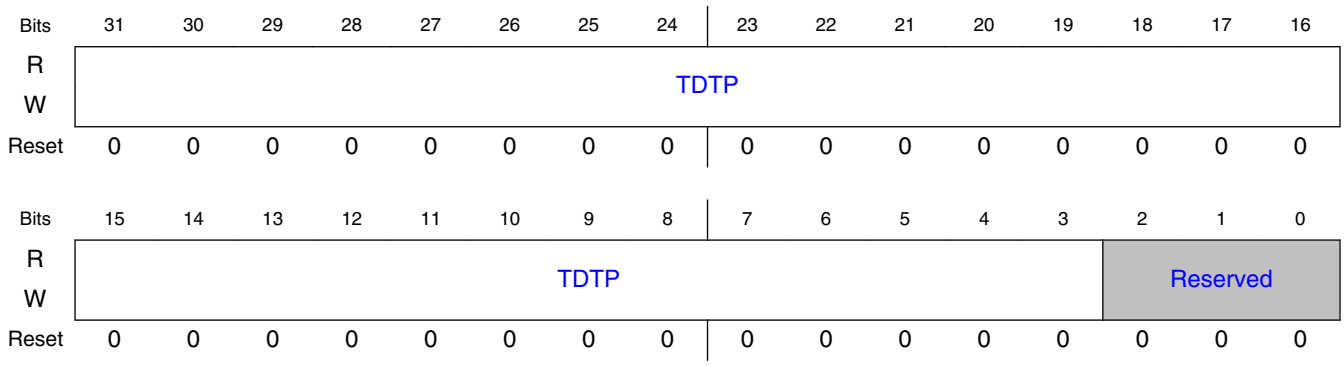
11.7.6.1.511 Channel 2 Tx Descriptor Tail Pointer (DMA_CH2_TXDESC_TAIL_POINTER)

The Channel 2 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

11.7.6.1.511.1 Offset

Register	Offset
DMA_CH2_TXDESC_TAIL_POINTER	1220h

11.7.6.1.511.2 Diagram



11.7.6.1.511.3 Fields

Field	Description
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

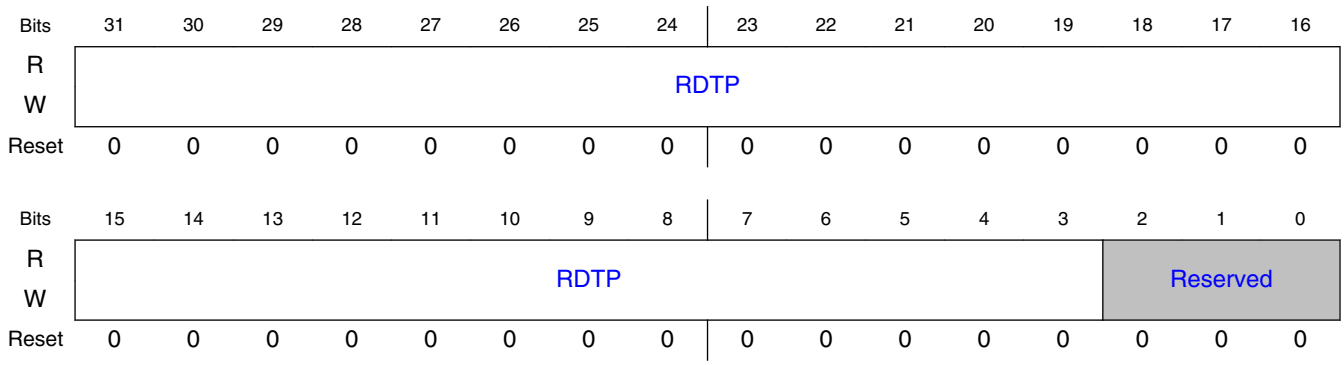
11.7.6.1.512 Channel 2 Rx Descriptor Tail Pointer (DMA_CH2_RXDESC_TAIL_POINTER)

The Channel 2 Rx Descriptor Tail Pointer Points to an offset from the base and indicates the location of the last valid descriptor.

11.7.6.1.512.1 Offset

Register	Offset
DMA_CH2_RXDESC_TAIL_POINTER	1228h

11.7.6.1.512.2 Diagram



11.7.6.1.512.3 Fields

Field	Description
31-3 RDTP	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

11.7.6.1.513 Channel 2 Tx Descriptor Ring Length (DMA_CH2_TXDESC_RING_LENGTH)

The Channel 2 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

11.7.6.1.513.1 Offset

Register	Offset
DMA_CH2_TXDESC_RING_LENGTH	122Ch

11.7.6.1.513.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						TDRL									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.513.3 Fields

Field	Description
31-10 —	Reserved
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. Synopsys recommends a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

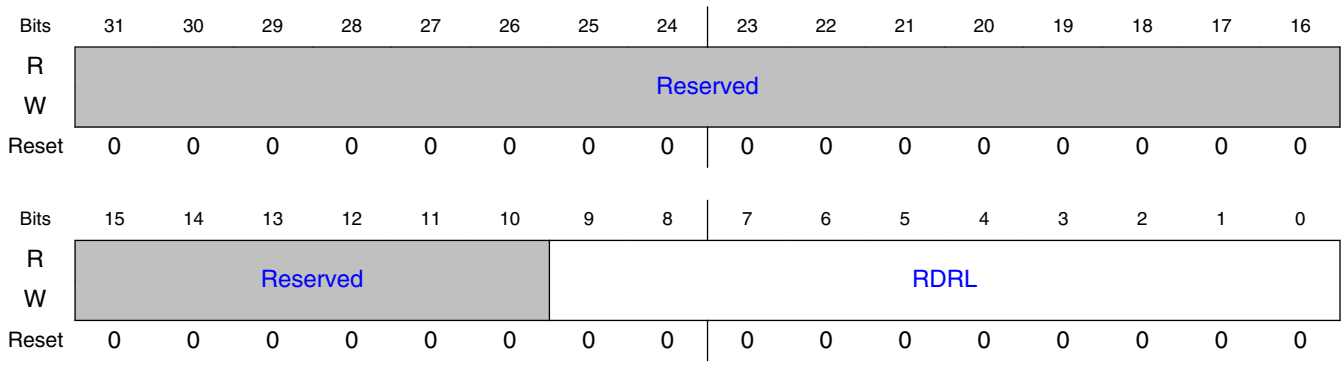
11.7.6.1.514 Channel 2 Rx Descriptor Ring Length (DMA_CH2_RXDESC_RING_LENGTH)

The Channel 2 Rx Descriptor Ring Length register contains the length of the Receive descriptor circular ring.

11.7.6.1.514.1 Offset

Register	Offset
DMA_CH2_RXDESC_RING_LENGTH	1230h

11.7.6.1.514.2 Diagram



11.7.6.1.514.3 Fields

Field	Description
31-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

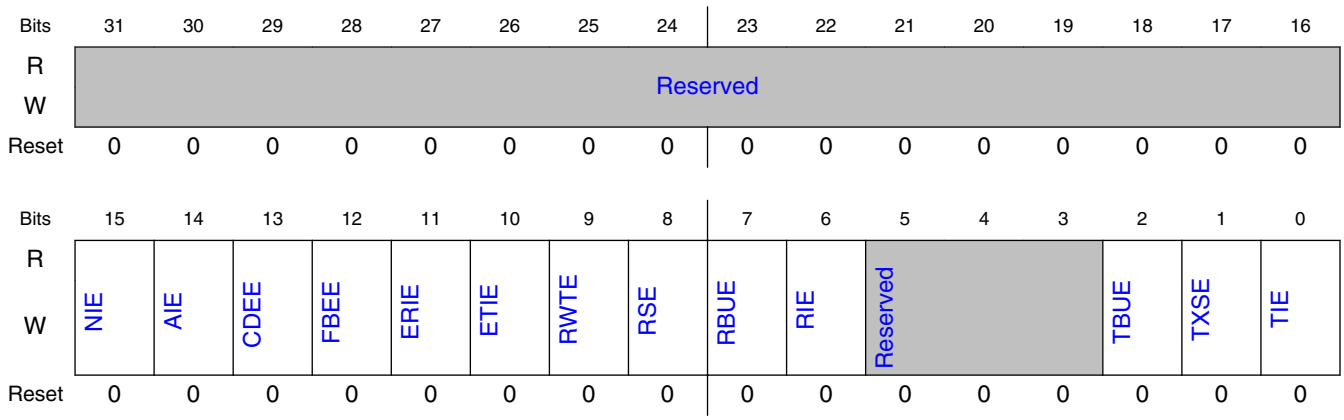
11.7.6.1.515 Channel 2 Interrupt Enable (DMA_CH2_INTERRUPT_ENABLE)

The Channel 2 Interrupt Enable register enables the interrupts reported by the Status register.

11.7.6.1.515.1 Offset

Register	Offset
DMA_CH2_INTERRUPT_ENABLE	1234h

11.7.6.1.515.2 Diagram



11.7.6.1.515.3 Fields

Field	Description
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0 - Normal Interrupt Summary is disabled 1 - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error

Table continues on the next page...

Field	Description
	When this bit is reset, the abnormal interrupt summary is disabled. 0 - Abnormal Interrupt Summary is disabled 1 - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0 - Context Descriptor Error is disabled 1 - Context Descriptor Error is enabled
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0 - Fatal Bus Error is disabled 1 - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0 - Early Receive Interrupt is disabled 1 - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0 - Early Transmit Interrupt is disabled 1 - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0 - Receive Watchdog Timeout is disabled 1 - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled.

Table continues on the next page...

Field	Description
	0 - Receive Stopped is disabled 1 - Receive Stopped is enabled
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0 - Receive Buffer Unavailable is disabled 1 - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0 - Receive Interrupt is disabled 1 - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0 - Transmit Buffer Unavailable is disabled 1 - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0 - Transmit Stopped is disabled 1 - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0 - Transmit Interrupt is disabled 1 - Transmit Interrupt is enabled

11.7.6.1.516 Channel 2 Receive Interrupt Watchdog Timer (DMA_CH2_RX_INTERRUPT_WATCHDOG_TIMER)

The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the STATUS register.

11.7.6.1.516.1 Offset

Register	Offset
DMA_CH2_RX_INTERRUPT_WATCHDOG_TIMER	1238h

11.7.6.1.516.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														RWTU	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RWT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.516.3 Fields

Field	Description
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024 - 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15-8 —	Reserved

Table continues on the next page...

Field	Description
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

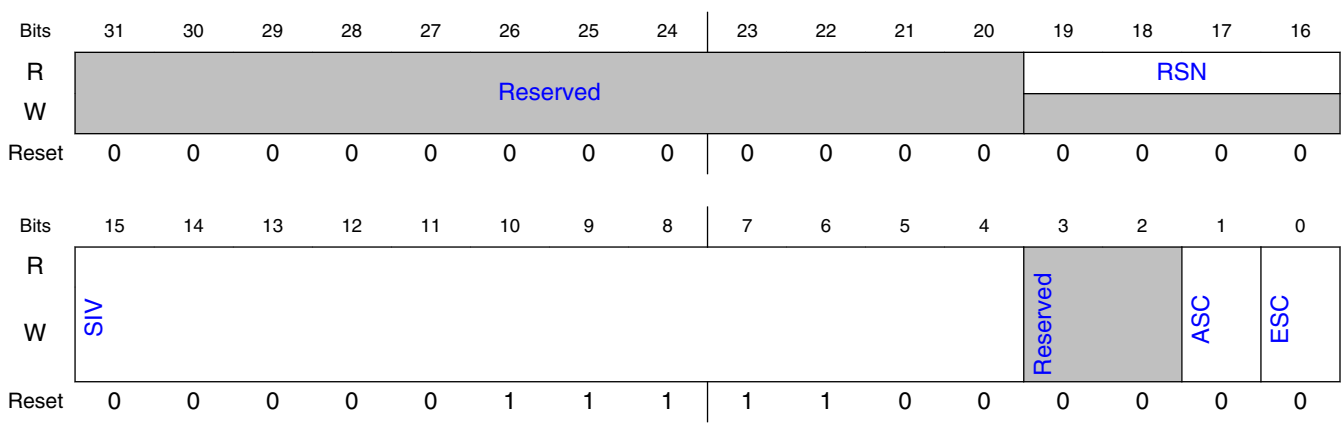
11.7.6.1.517 Channel 2 Slot Function Control and Status (DMA_CH2_SLOT_FUNCTION_CONTROL_STATUS)

The Channel 2 Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

11.7.6.1.517.1 Offset

Register	Offset
DMA_CH2_SLOT_FUNCTION_CONTROL_STATUS	123Ch

11.7.6.1.517.2 Diagram



11.7.6.1.517.3 Fields

Field	Description
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0 - Advance Slot Check is disabled 1 - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or - ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0 - Slot Comparison is disabled 1 - Slot Comparison is enabled

11.7.6.1.518 Channel 2 Current Application Transmit Descriptor (DMA_CH2_CURRENT_APP_TXDESC)

The Channel 2 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

11.7.6.1.518.1 Offset

Register	Offset
DMA_CH2_CURRENT_APP_TXDESC	1244h

11.7.6.1.518.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.518.3 Fields

Field	Description
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

11.7.6.1.519 Channel 2 Current Application Receive Descriptor (DMA_CH2_CURRENT_APP_RXDESC)

The Channel 2 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

11.7.6.1.519.1 Offset

Register	Offset
DMA_CH2_CURRENT_APP_RXDESC	124Ch

11.7.6.1.519.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.519.3 Fields

Field	Description
31-0 CURRDESAPTR	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

11.7.6.1.520 Channel 2 Current Application Transmit Buffer Address (DMA_CH2_CURRENT_APP_TXBUFFER)

The Channel 2 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

11.7.6.1.520.1 Offset

Register	Offset
DMA_CH2_CURRENT_APP_TXBUFFER	1254h

11.7.6.1.520.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.520.3 Fields

Field	Description
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

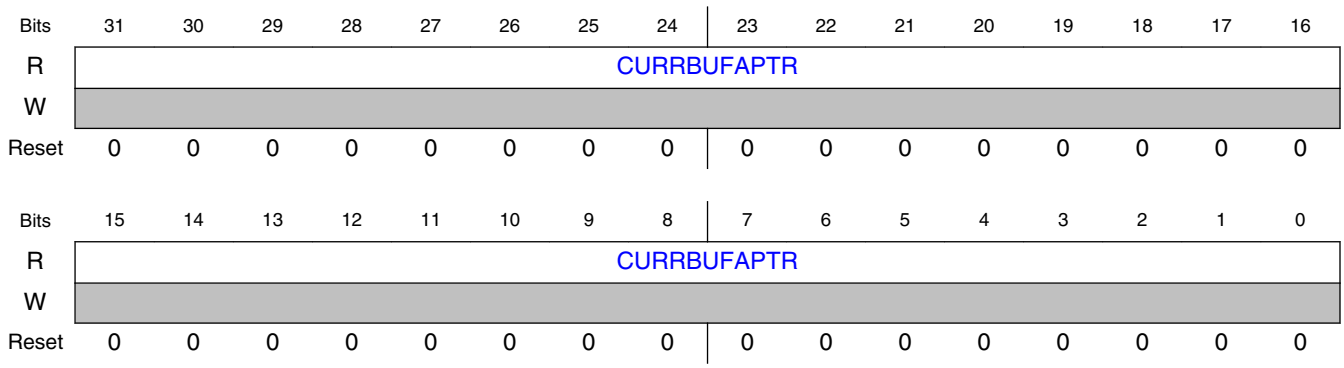
11.7.6.1.521 Channel 2 Current Application Receive Buffer Address (DMA_CH2_CURRENT_APP_RXBUFFER)

The Channel 2 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

11.7.6.1.521.1 Offset

Register	Offset
DMA_CH2_CURRENT_APP_RXBUFFER	125Ch

11.7.6.1.521.2 Diagram



11.7.6.1.521.3 Fields

Field	Description
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

11.7.6.1.522 DMA Channel 2 Status (DMA_CH2_STATUS)

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

11.7.6.1.522.1 Offset

Register	Offset
DMA_CH2_STATUS	1260h

11.7.6.1.522.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										REB			TEB		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.522.3 Fields

Field	Description
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access

Table continues on the next page...

Field	Description
	- Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH2_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH2_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Normal Interrupt Summary status not detected 1 - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH2_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH2_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Abnormal Interrupt Summary status not detected 1 - Abnormal Interrupt Summary status detected

Table continues on the next page...

Field	Description
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Context Descriptor Error status not detected 1 - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Fatal Bus Error status not detected 1 - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Receive Interrupt status not detected 1 - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Transmit Interrupt status not detected 1 - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0 - Receive Watchdog Timeout status not detected

Table continues on the next page...

Field	Description
	1 - Receive Watchdog Timeout status detected
8 RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Process Stopped status not detected 1 - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode, the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Buffer Unavailable status not detected 1 - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Interrupt status not detected 1 - Receive Interrupt status detected
5-3 —	Reserved
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Buffer Unavailable status not detected

Table continues on the next page...

Field	Description
	1 - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Process Stopped status not detected 1 - Transmit Process Stopped status detected
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Interrupt status not detected 1 - Transmit Interrupt status detected

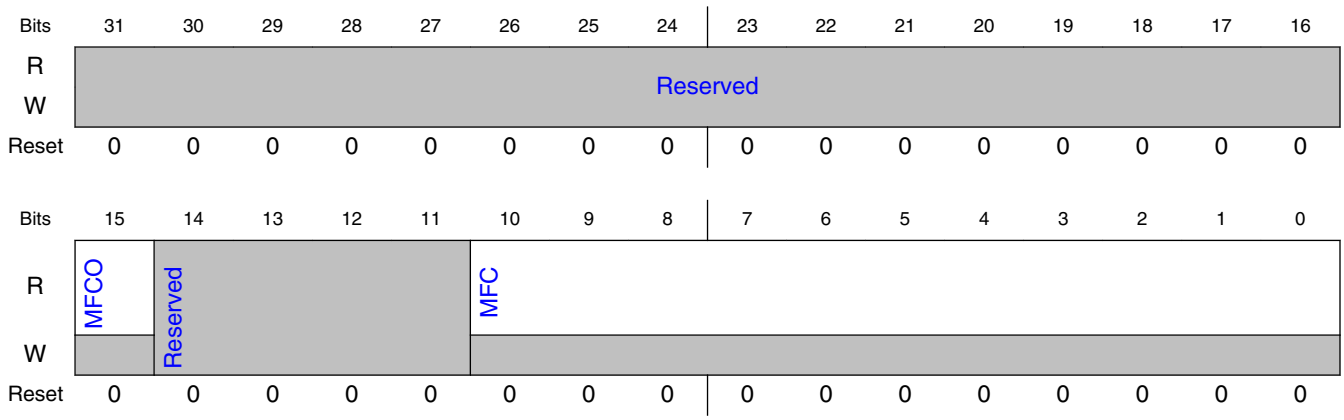
11.7.6.1.523 Channel 2 Missed Frame Counter (DMA_CH2_MISS_FRAME_CNT)

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in DMA_CH2_RX_CONTROL register.

11.7.6.1.523.1 Offset

Register	Offset
DMA_CH2_MISS_FRAME_CNT	1264h

11.7.6.1.523.2 Diagram



11.7.6.1.523.3 Fields

Field	Description
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Miss Frame Counter overflow not occurred 1 - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH2_RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH2_RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

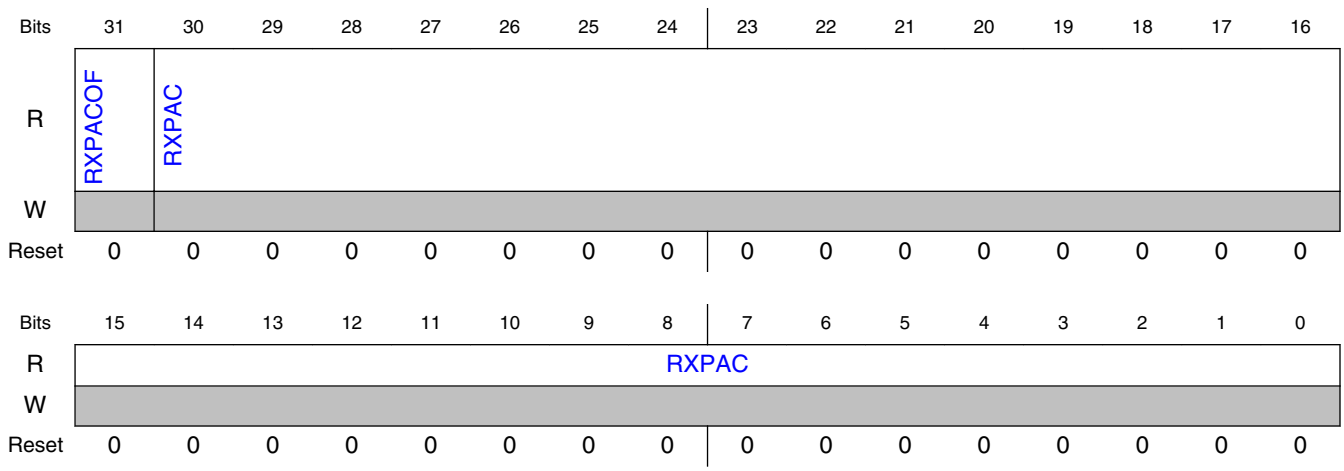
11.7.6.1.524 Channel 2 RXP Frames Accepted Counter (DMA_CH2_RXP_ACCEPT_CNT)

The Channel 2 RXP Frames Accepted Counter register provides the count of the number of frames accepted by Rx Parser.

11.7.6.1.524.1 Offset

Register	Offset
DMA_CH2_RXP_ACCEPT_CNT	1268h

11.7.6.1.524.2 Diagram



11.7.6.1.524.3 Fields

Field	Description
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Accept Counter overflow not occurred 1 - Rx Parser Accept Counter overflow occurred
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

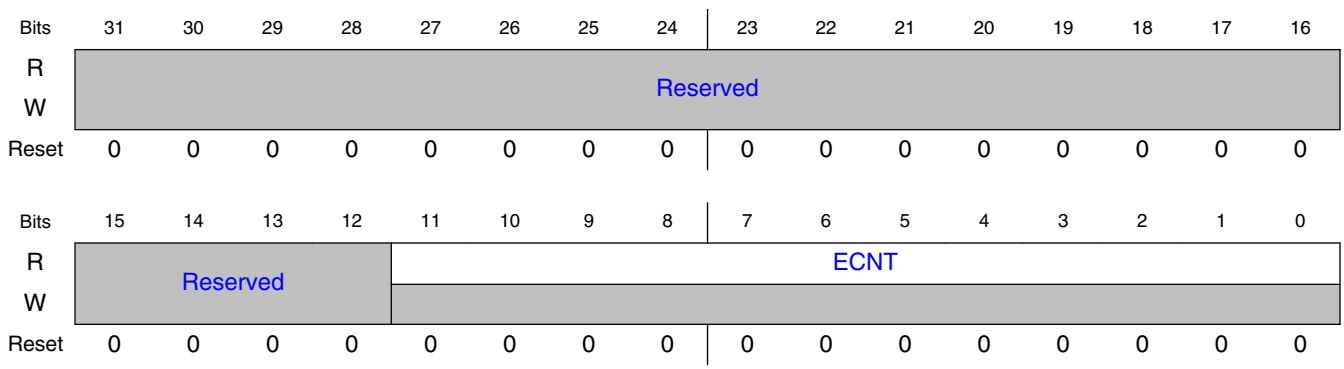
11.7.6.1.525 Channel 2 Receive ERI Counter (DMA_CH2_RX_ERI_CNT)

The Channel 2 Receive ERI Counter register provides the count of the number of times ERI was asserted.

11.7.6.1.525.1 Offset

Register	Offset
DMA_CH2_RX_ERI_CNT	126Ch

11.7.6.1.525.2 Diagram



11.7.6.1.525.3 Fields

Field	Description
31-12 —	Reserved
11-0 ECNT	ERI Counter When ERIC bit of DMA_CH2_RX_CONTROL register is set, this counter increments for burst transfer completed by the Rx DMA from the start of packet transfer. ERI Counter When ERIC bit of DMA_CH2_RX_CONTROL register is set, this counter increments for burst transfer completed by the Rx DMA from the start of packet transfer. This counter will get reset at the start of new packet.

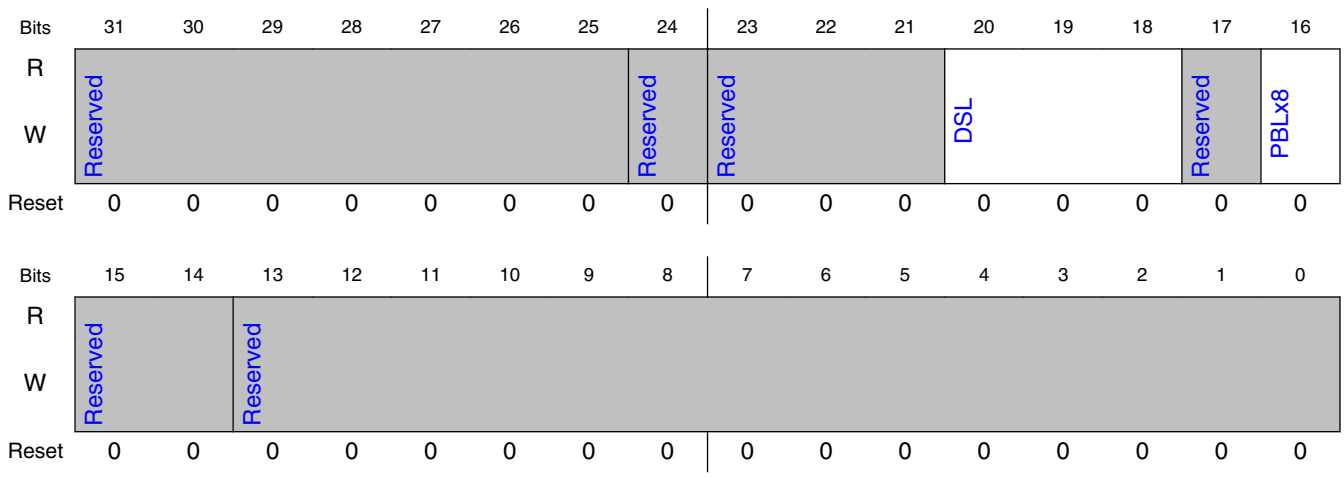
11.7.6.1.526 DMA Channel 3 Control (DMA_CH3_CONTROL)

The DMA Channel 3 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

11.7.6.1.526.1 Offset

Register	Offset
DMA_CH3_CONTROL	1280h

11.7.6.1.526.2 Diagram



11.7.6.1.526.3 Fields

Field	Description
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length

Table continues on the next page...

Field	Description
	This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16 PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH3_TX_CONTROL and Bits[21:16] in DMA_CH3_RX_CONTROL is multiplied by eight times. 8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH3_TX_CONTROL and Bits[21:16] in DMA_CH3_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0 - 8xPBL mode is disabled 1 - 8xPBL mode is enabled
15-14 —	Reserved
13-0 —	Reserved

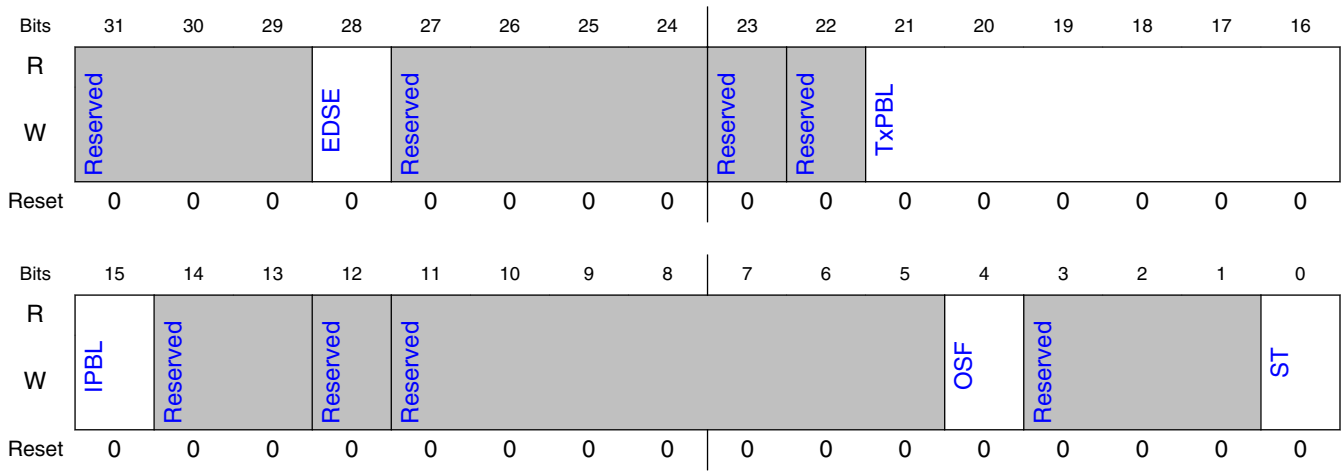
11.7.6.1.527 DMA Channel 3 Transmit Control (DMA_CH3_TX_CONTROL)

The DMA Channel 3 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

11.7.6.1.527.1 Offset

Register	Offset
DMA_CH3_TX_CONTROL	1284h

11.7.6.1.527.2 Diagram



11.7.6.1.527.3 Fields

Field	Description
31-29 —	Reserved
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0 - Enhanced Descriptor is disabled 1 - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in DMA_CH3_CONTROL register.

Table continues on the next page...

Field	Description
	2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ[n]_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0 - Ignore PBL Requirement is disabled 1 - Ignore PBL Requirement is enabled
14-13 —	Reserved
12 —	Reserved
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0 - Operate on Second Packet disabled 1 - Operate on Second Packet enabled
3-1 —	Reserved
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH3_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH3_STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH3_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to

Field	Description
	program DMA_CH3_TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0 - Stop Transmission Command 1 - Start Transmission Command

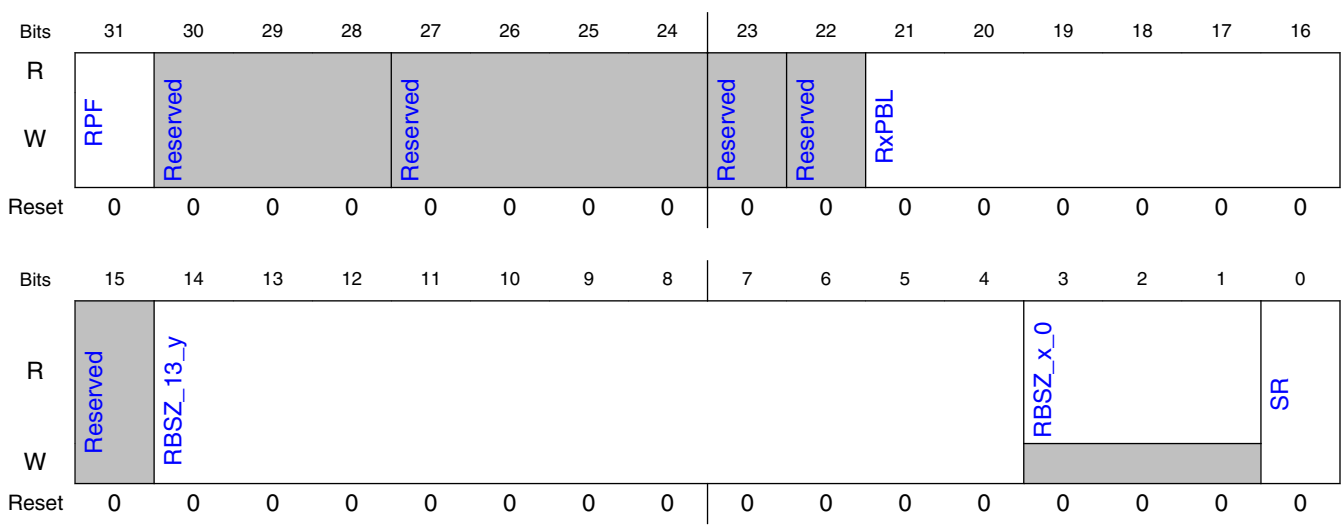
11.7.6.1.528 DMA Channel 3 Receive Control (DMA_CH3_RX_CONTROL)

The DMA Channel 3 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

11.7.6.1.528.1 Offset

Register	Offset
DMA_CH3_RX_CONTROL	1288h

11.7.6.1.528.2 Diagram



11.7.6.1.528.3 Fields

Field	Description
31 RPF	Rx Packet Flush. Rx Packet Flush.

Table continues on the next page...

Field	Description
	When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0 - Rx Packet Flush is disabled 1 - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ[n]_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0.

Table continues on the next page...

Field	Description
RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH3_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH3_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH3_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0 - Stop Receive 1 - Start Receive

11.7.6.1.529 Channel 3 Tx Descriptor List Address (DMA_CH3_TXDESC_LIST_ADDRESS)

The Channel 3 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

11.7.6.1.529.1 Offset

Register	Offset
DMA_CH3_TXDESC_LIST_ADDRESS	1294h

11.7.6.1.529.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDESLA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDESLA													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.529.3 Fields

Field	Description
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

11.7.6.1.530 Channel 3 Rx Descriptor List Address (DMA_CH3_RXDESC_LIST_ADDRESS)

The Channel 3 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

11.7.6.1.530.1 Offset

Register	Offset
DMA_CH3_RXDESC_LIST_ADDRESS	129Ch

11.7.6.1.530.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RDESLA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RDESLA													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.530.3 Fields

Field	Description
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration

Table continues on the next page...

Field	Description
	- 31:4 for 128-bit configuration
2-0 —	Reserved

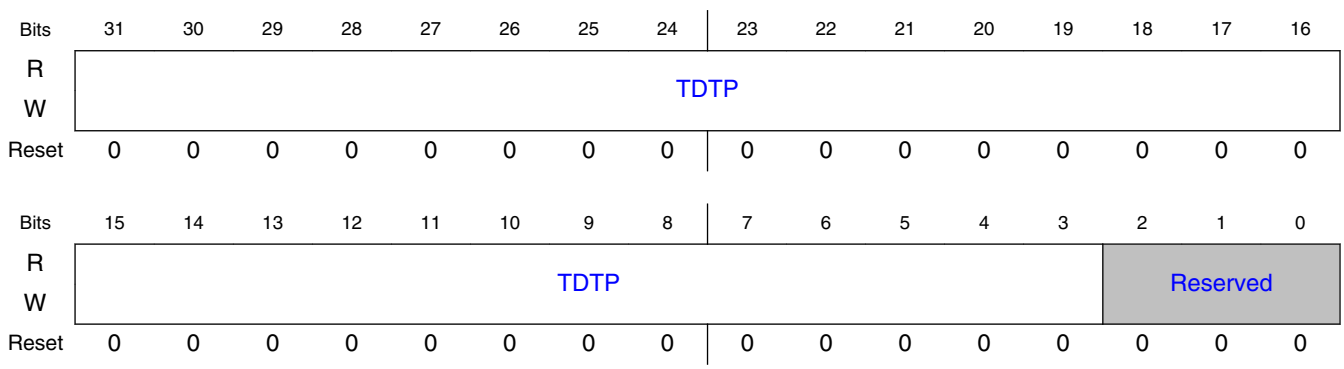
11.7.6.1.531 Channel 3 Tx Descriptor Tail Pointer (DMA_CH3_TXDESC_TAIL_POINTER)

The Channel 3 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

11.7.6.1.531.1 Offset

Register	Offset
DMA_CH3_TXDESC_TAIL_POINTER	12A0h

11.7.6.1.531.2 Diagram



11.7.6.1.531.3 Fields

Field	Description
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration

Table continues on the next page...

Field	Description
	- 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

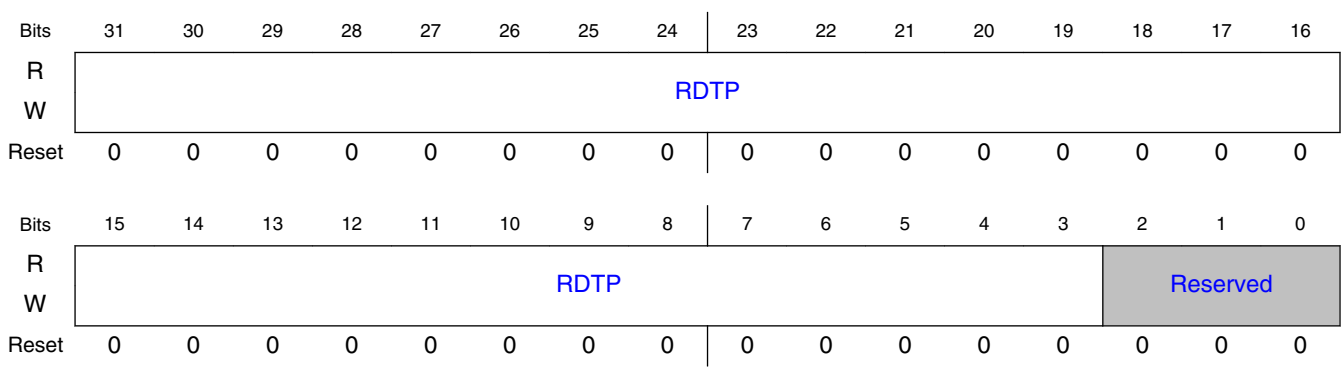
11.7.6.1.532 Channel 3 Rx Descriptor Tail Pointer (DMA_CH3_RXDESC_TAIL_POINTER)

The Channel 3 Rx Descriptor Tail Pointer points to an offset from the base and indicates the location of the last valid descriptor.

11.7.6.1.532.1 Offset

Register	Offset
DMA_CH3_RXDESC_TAIL_POINTER	12A8h

11.7.6.1.532.2 Diagram



11.7.6.1.532.3 Fields

Field	Description
31-3 RDTP	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration:

Table continues on the next page...

Field	Description
	- 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

11.7.6.1.533 Channel 3 Tx Descriptor Ring Length (DMA_CH3_TXDESC_RING_LENGTH)

The Channel 3 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

11.7.6.1.533.1 Offset

Register	Offset
DMA_CH3_TXDESC_RING_LENGTH	12ACh

11.7.6.1.533.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						TDRL									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.533.3 Fields

Field	Description
31-10 —	Reserved
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring.

Field	Description
	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. Synopsys recommends a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

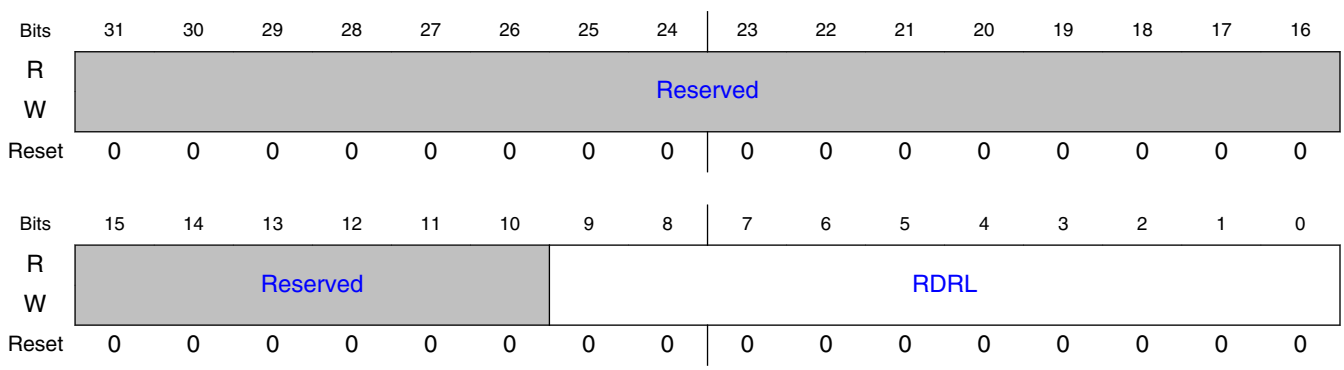
11.7.6.1.534 Channel 3 Rx Descriptor Ring Length (DMA_CH3_RXDESC_RING_LENGTH)

The Channel 3 Rx Descriptor Ring Length register contains the length of the Receive descriptor circular ring.

11.7.6.1.534.1 Offset

Register	Offset
DMA_CH3_RXDESC_RING_LENGTH	12B0h

11.7.6.1.534.2 Diagram



11.7.6.1.534.3 Fields

Field	Description
31-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length

Field	Description
	This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

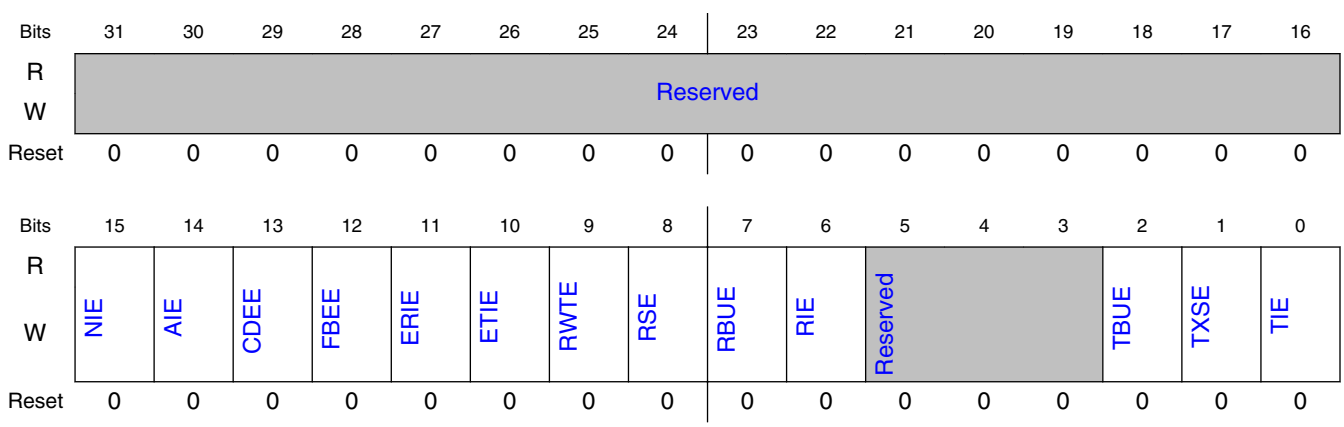
11.7.6.1.535 Channel 3 Interrupt Enable (DMA_CH3_INTERRUPT_ENABLE)

The Channel 3 Interrupt Enable register enables the interrupts reported by the Status register.

11.7.6.1.535.1 Offset

Register	Offset
DMA_CH3_INTERRUPT_ENABLE	12B4h

11.7.6.1.535.2 Diagram



11.7.6.1.535.3 Fields

Field	Description
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable

Table continues on the next page...

Field	Description
	When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the DMA_CH3_STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0 - Normal Interrupt Summary is disabled 1 - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled. 0 - Abnormal Interrupt Summary is disabled 1 - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0 - Context Descriptor Error is disabled 1 - Context Descriptor Error is enabled
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0 - Fatal Bus Error is disabled 1 - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled.

Table continues on the next page...

Field	Description
	0 - Early Receive Interrupt is disabled 1 - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0 - Early Transmit Interrupt is disabled 1 - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0 - Receive Watchdog Timeout is disabled 1 - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0 - Receive Stopped is disabled 1 - Receive Stopped is enabled
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0 - Receive Buffer Unavailable is disabled 1 - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0 - Receive Interrupt is disabled 1 - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled.

Table continues on the next page...

Field	Description
	0 - Transmit Buffer Unavailable is disabled 1 - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0 - Transmit Stopped is disabled 1 - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0 - Transmit Interrupt is disabled 1 - Transmit Interrupt is enabled

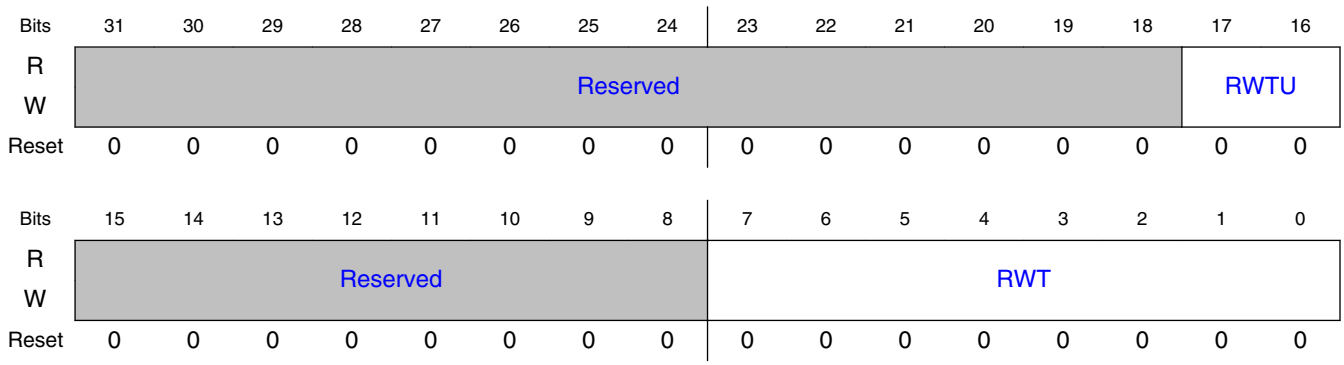
11.7.6.1.536 Channel 3 Receive Interrupt Watchdog Time (DMA_CH3_RX_INTERRUPT_WATCHDOG_TIMER)

The Channel 3 Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the STATUS register.

11.7.6.1.536.1 Offset

Register	Offset
DMA_CH3_RX_INTERRUPT_WATCHDOG_TIMER	12B8h

11.7.6.1.536.2 Diagram



11.7.6.1.536.3 Fields

Field	Description
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024 - 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15-8 —	Reserved
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

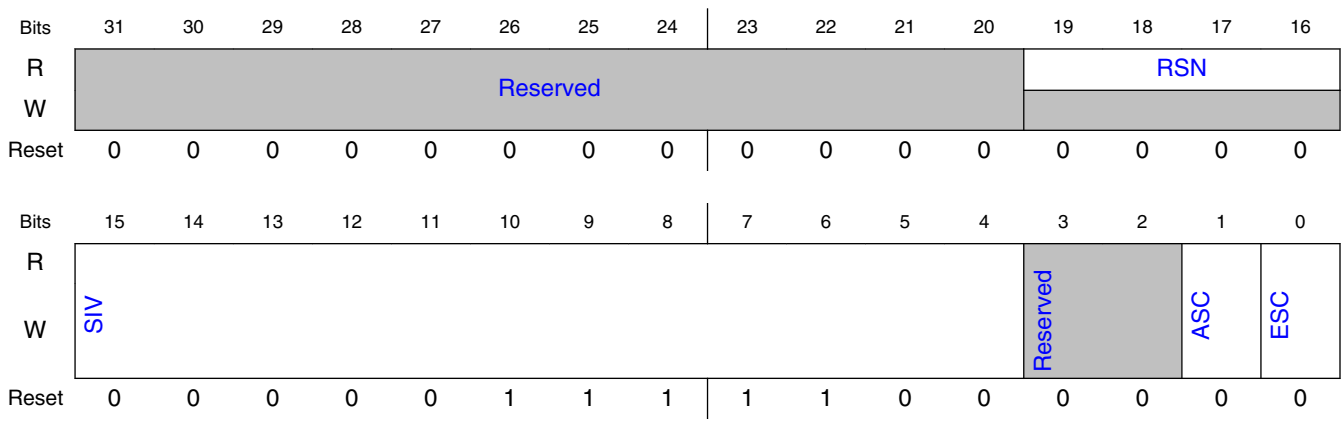
11.7.6.1.537 Channel 3 Slot Function Control and Status (DMA_CH3_SLOT_FUNCTION_CONTROL_STATUS)

The Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

11.7.6.1.537.1 Offset

Register	Offset
DMA_CH3_SLOT_FUNCTION_CONTROL_STATUS	12BCh

11.7.6.1.537.2 Diagram



11.7.6.1.537.3 Fields

Field	Description
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us

Table continues on the next page...

Field	Description
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0 - Advance Slot Check is disabled 1 - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or - ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0 - Slot Comparison is disabled 1 - Slot Comparison is enabled

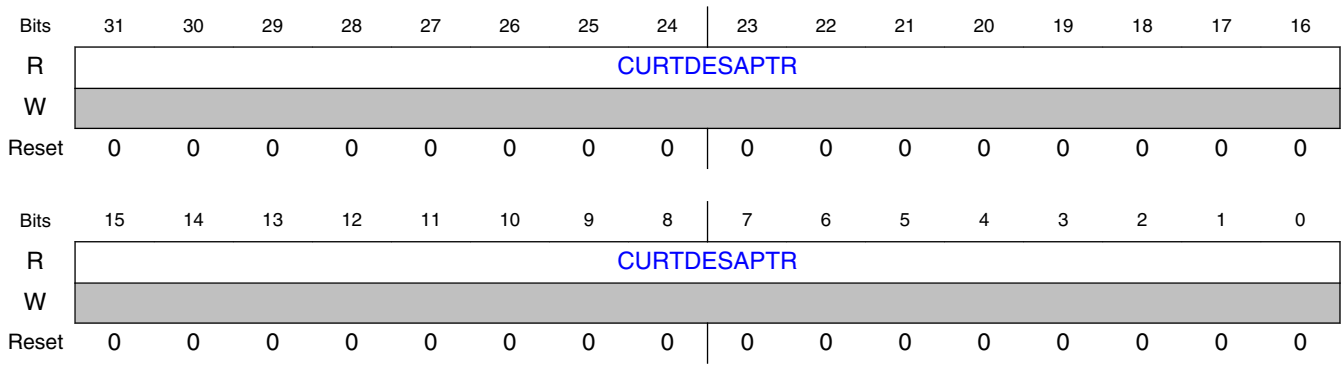
11.7.6.1.538 Channel 3 Current Application Transmit Descriptor (DMA_CH3_CURRENT_APP_TXDESC)

The Channel 3 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

11.7.6.1.538.1 Offset

Register	Offset
DMA_CH3_CURRENT_APP_TXDESC	12C4h

11.7.6.1.538.2 Diagram



11.7.6.1.538.3 Fields

Field	Description
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

11.7.6.1.539 Channel 3 Current Application Receive Descriptor (DMA_CH3_CURRENT_APP_RXDESC)

The Channel 3 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

11.7.6.1.539.1 Offset

Register	Offset
DMA_CH3_CURRENT_APP_RXDESC	12CCh

11.7.6.1.539.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.539.3 Fields

Field	Description
31-0 CURRDESAPTR	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

11.7.6.1.540 Channel 3 Current Application Transmit Buffer Address (DMA_CH3_CURRENT_APP_TXBUFFER)

The Channel 3 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

11.7.6.1.540.1 Offset

Register	Offset
DMA_CH3_CURRENT_APP_TXBUFFER	12D4h

11.7.6.1.540.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.540.3 Fields

Field	Description
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

11.7.6.1.541 Channel 3 Current Application Receive Buffer Address (DMA_CH3_CURRENT_APP_RXBUFFER)

The Channel 3 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

11.7.6.1.541.1 Offset

Register	Offset
DMA_CH3_CURRENT_APP_RXBUFFER	12DCh

11.7.6.1.541.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.541.3 Fields

Field	Description
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

11.7.6.1.542 DMA Channel 3 Status (DMA_CH3_STATUS)

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

11.7.6.1.542.1 Offset

Register	Offset
DMA_CH3_STATUS	12E0h

11.7.6.1.542.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										REB			TEB		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.542.3 Fields

Field	Description
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access

Table continues on the next page...

Field	Description
	- Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH3_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH3_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH3_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH3_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Normal Interrupt Summary status not detected 1 - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH3_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH3_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Abnormal Interrupt Summary status not detected 1 - Abnormal Interrupt Summary status detected

Table continues on the next page...

Field	Description
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Context Descriptor Error status not detected 1 - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Fatal Bus Error status not detected 1 - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Receive Interrupt status not detected 1 - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Transmit Interrupt status not detected 1 - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0 - Receive Watchdog Timeout status not detected

Table continues on the next page...

Field	Description
	1 - Receive Watchdog Timeout status detected
8 RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Process Stopped status not detected 1 - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode, the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Buffer Unavailable status not detected 1 - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Interrupt status not detected 1 - Receive Interrupt status detected
5-3 —	Reserved
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Buffer Unavailable status not detected

Table continues on the next page...

Field	Description
	1 - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Process Stopped status not detected 1 - Transmit Process Stopped status detected
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Interrupt status not detected 1 - Transmit Interrupt status detected

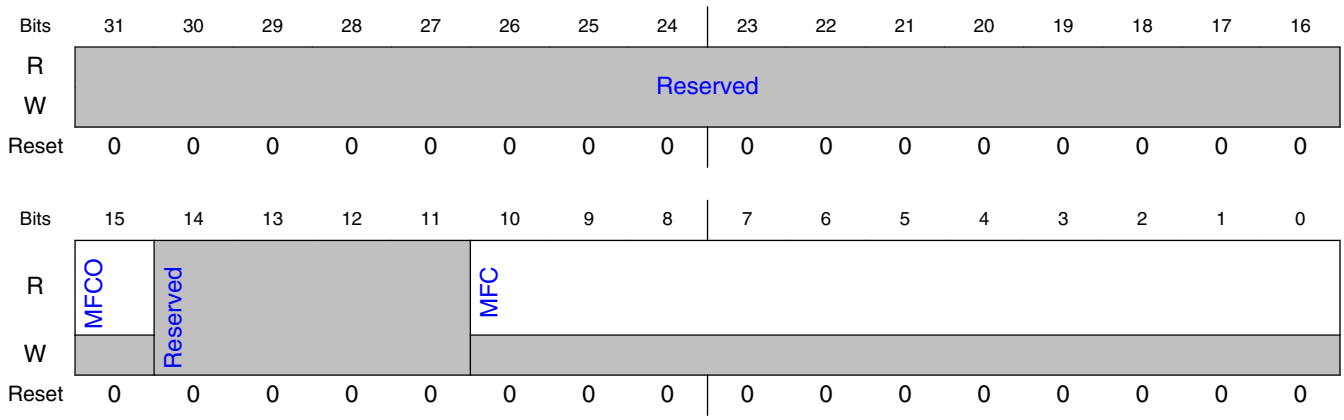
11.7.6.1.543 Channel 3 Missed Frame Counter (DMA_CH3_MISS_FRAME_CNT)

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in DMA_CH3_RX_CONTROL register.

11.7.6.1.543.1 Offset

Register	Offset
DMA_CH3_MISS_FRAME_CNT	12E4h

11.7.6.1.543.2 Diagram



11.7.6.1.543.3 Fields

Field	Description
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Miss Frame Counter overflow not occurred 1 - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH3_RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

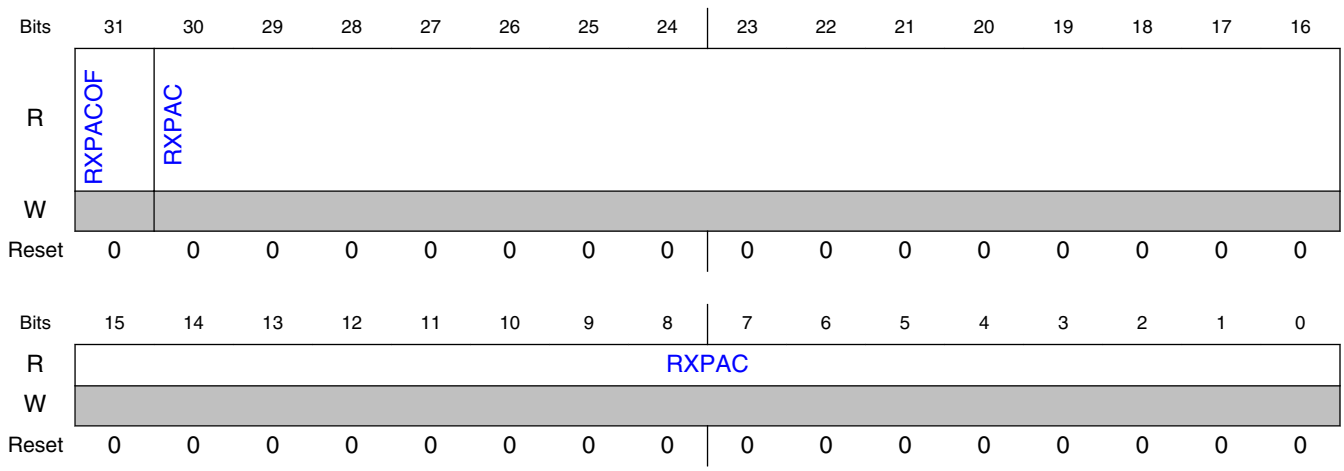
11.7.6.1.544 Channel 3 RXP Frames Accepted Counter (DMA_CH3_RXP_ACCEPT_CNT)

The Channel 3 RXP Frames Accepted Counter register provides the count of the number of frames accepted by Rx Parser.

11.7.6.1.544.1 Offset

Register	Offset
DMA_CH3_RXP_ACCEPT_CNT	12E8h

11.7.6.1.544.2 Diagram



11.7.6.1.544.3 Fields

Field	Description
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Accept Counter overflow not occurred 1 - Rx Parser Accept Counter overflow occurred
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

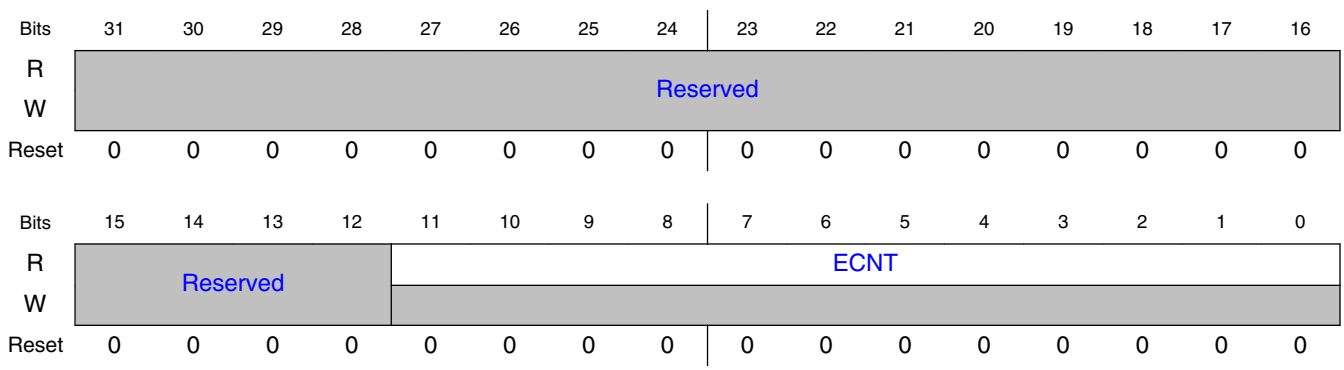
11.7.6.1.545 Channel 3 Receive ERI Counter (DMA_CH3_RX_ERI_CNT)

The Channel 3 Receive ERI Counter register provides the count of the number of times ERI was asserted.

11.7.6.1.545.1 Offset

Register	Offset
DMA_CH3_RX_ERI_CNT	12ECh

11.7.6.1.545.2 Diagram



11.7.6.1.545.3 Fields

Field	Description
31-12 —	Reserved
11-0 ECNT	ERI Counter When ERIC bit of DMA_CH3_RX_CONTROL register is set, this counter increments for burst transfer completed by the Rx DMA from the start of packet transfer. ERI Counter When ERIC bit of DMA_CH3_RX_CONTROL register is set, this counter increments for burst transfer completed by the Rx DMA from the start of packet transfer. This counter will get reset at the start of new packet.

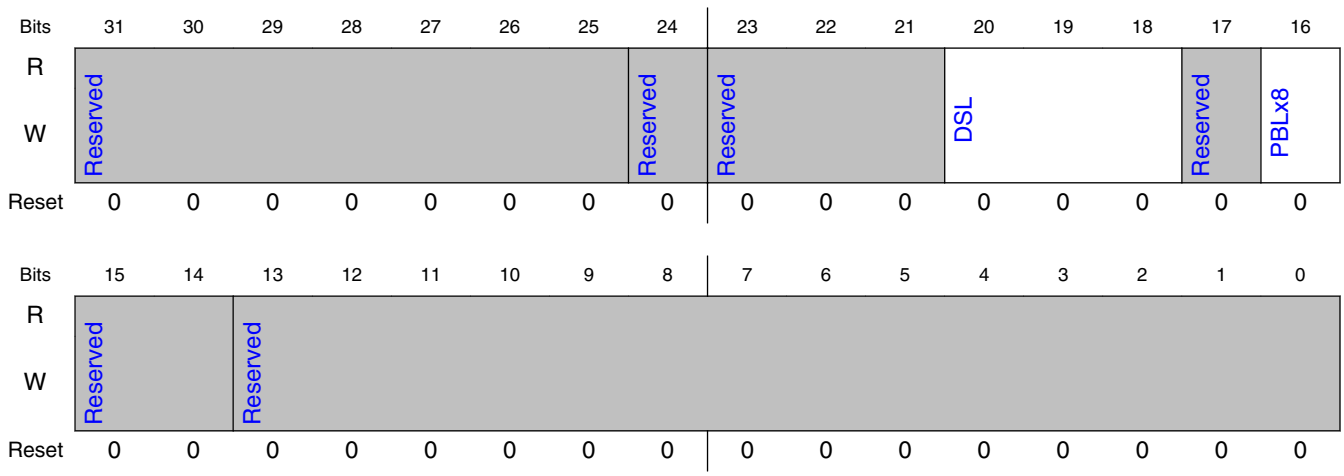
11.7.6.1.546 DMA Channel 4 Control (DMA_CH4_CONTROL)

The DMA Channel 4 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

11.7.6.1.546.1 Offset

Register	Offset
DMA_CH4_CONTROL	1300h

11.7.6.1.546.2 Diagram



11.7.6.1.546.3 Fields

Field	Description
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH4_TX_CONTROL and Bits[21:16] in DMA_CH4_RX_CONTROL is multiplied by eight times.

Table continues on the next page...

Field	Description
PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH4_TX_CONTROL and Bits[21:16] in DMA_CH4_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0 - 8xPBL mode is disabled 1 - 8xPBL mode is enabled
15-14 —	Reserved
13-0 —	Reserved

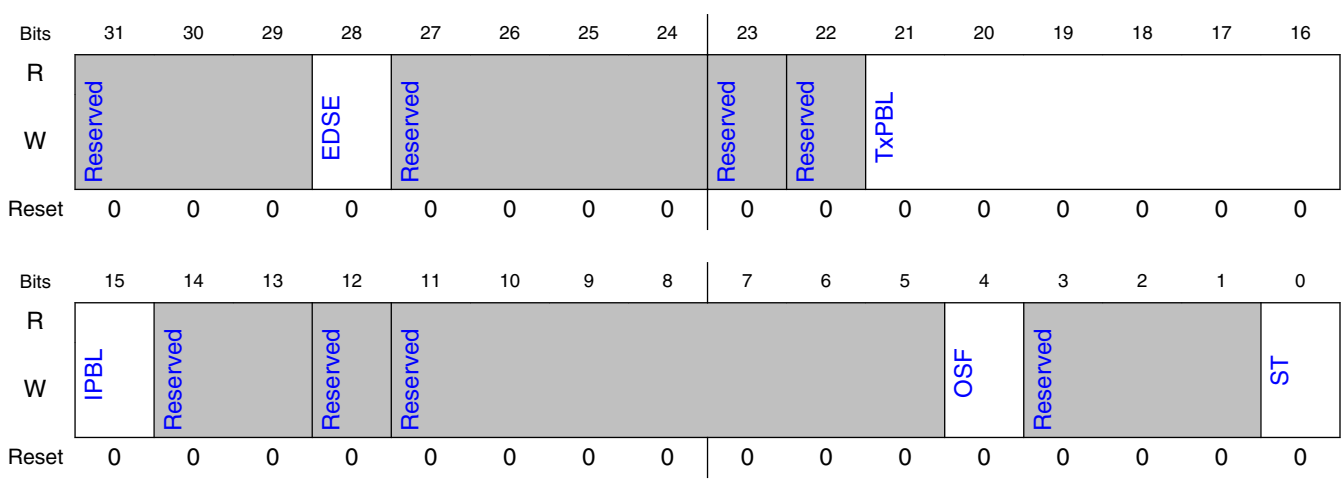
11.7.6.1.547 DMA Channel 4 Transmit Control (DMA_CH4_TX_CONTROL)

The DMA Channel 4 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

11.7.6.1.547.1 Offset

Register	Offset
DMA_CH4_TX_CONTROL	1304h

11.7.6.1.547.2 Diagram



11.7.6.1.547.3 Fields

Field	Description
31-29 —	Reserved
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0 - Enhanced Descriptor is disabled 1 - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in CONTROL register. 2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ[n]_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0 - Ignore PBL Requirement is disabled 1 - Ignore PBL Requirement is enabled
14-13	Reserved

Table continues on the next page...

Field	Description
—	
12 —	Reserved
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0 - Operate on Second Packet disabled 1 - Operate on Second Packet enabled
3-1 —	Reserved
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH4_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH4_STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH4_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program DMA_CH4_TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0 - Stop Transmission Command 1 - Start Transmission Command

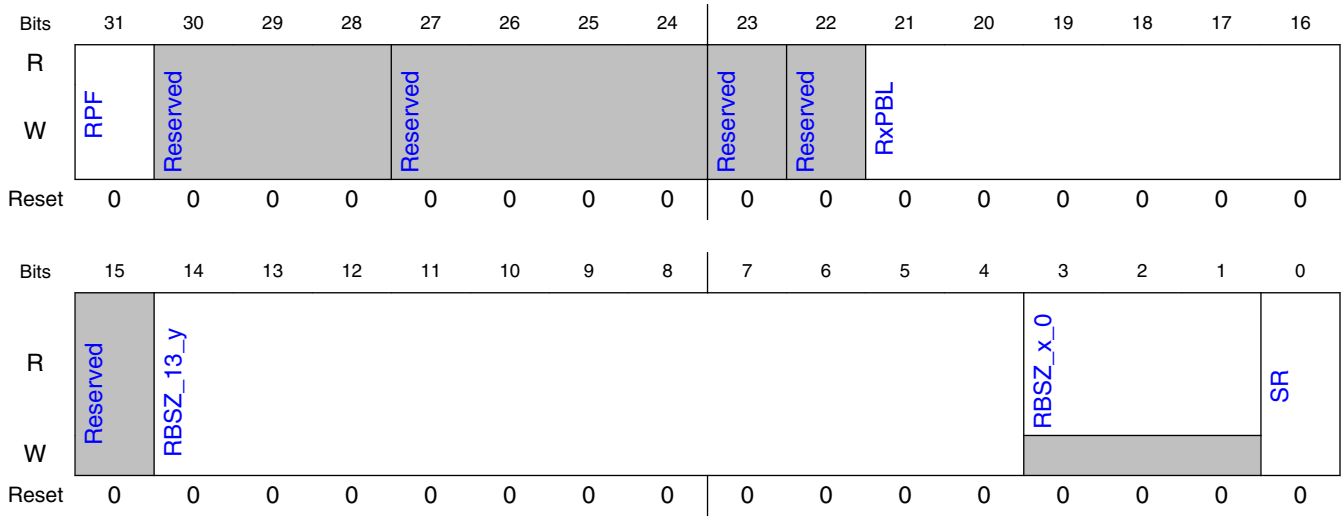
11.7.6.1.548 DMA Channel 4 Receive Control (DMA_CH4_RX_CONTROL)

The DMA Channel 4 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

11.7.6.1.548.1 Offset

Register	Offset
DMA_CH4_RX_CONTROL	1308h

11.7.6.1.548.2 Diagram



11.7.6.1.548.3 Fields

Field	Description
31 RPF	Rx Packet Flush. Rx Packet Flush. When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0 - Rx Packet Flush is disabled 1 - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer.

Table continues on the next page...

Field	Description
	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the DMA_CH4_CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ[n]_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1 RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH4_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH4_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH4_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable.

Field	Description
	When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0 - Stop Receive 1 - Start Receive

11.7.6.1.549 Channel 4 Tx Descriptor List Address (DMA_CH4_TXDESC_LIST_ADDRESS)

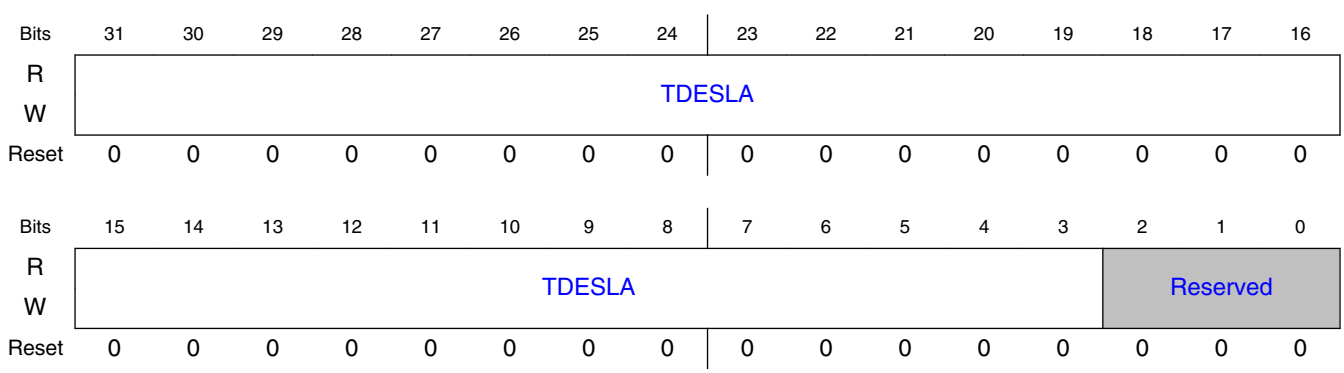
The Channel 4 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

11.7.6.1.549.1 Offset

Register	Offset
DMA_CH4_TXDESC_LIST_ADDRESS	1314h

11.7.6.1.549.2 Diagram



11.7.6.1.549.3 Fields

Field	Description
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

11.7.6.1.550 Channel 4 Rx Descriptor List Address (DMA_CH4_RXDESC_LIST_ADDRESS)

The Channel 4 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

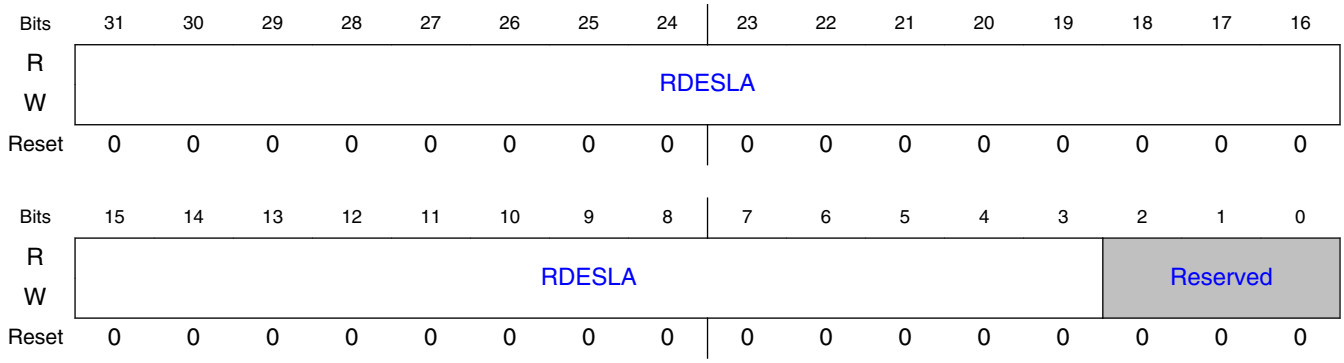
This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

11.7.6.1.550.1 Offset

Register	Offset
DMA_CH4_RXDESC_LIST_ADDRESS	131Ch

11.7.6.1.550.2 Diagram



11.7.6.1.550.3 Fields

Field	Description
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

11.7.6.1.551 Channel 4 Tx Descriptor Tail Pointer (DMA_CH4_TXDESC_TAIL_POINTER)

The Channel 4 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

11.7.6.1.551.1 Offset

Register	Offset
DMA_CH4_TXDESC_TAIL_POINTER	1320h

11.7.6.1.551.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDTP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDTP													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.551.3 Fields

Field	Description
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

11.7.6.1.552 Channel 4 Rx Descriptor Tail Pointer (DMA_CH4_RXDESC_TAIL_POINTER)

The Channel 4 Rx Descriptor Tail Pointer points to an offset from the base and indicates the location of the last valid descriptor.

11.7.6.1.552.1 Offset

Register	Offset
DMA_CH4_RXDESC_TAIL_POINTER	1328h

11.7.6.1.552.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RDTP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RDTP													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.552.3 Fields

Field	Description
31-3 RDTP	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

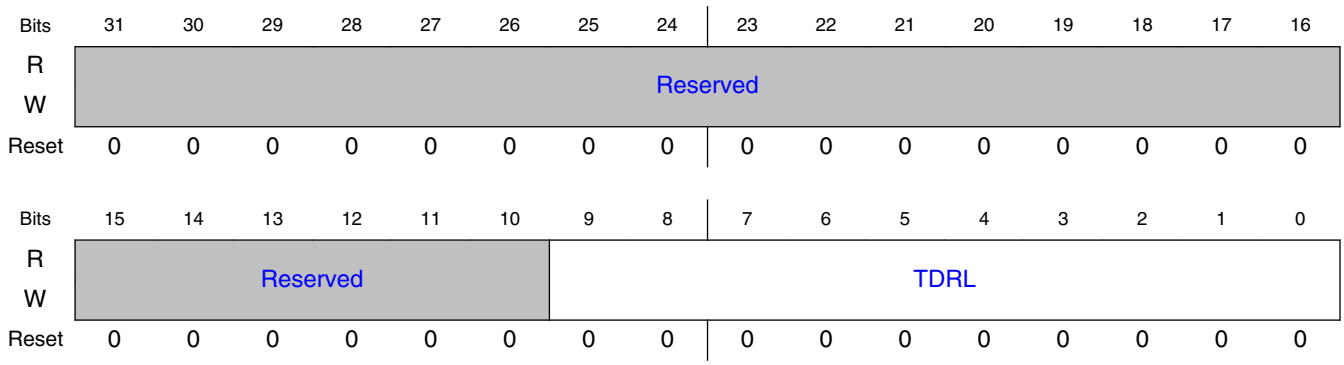
11.7.6.1.553 Channel 4 Tx Descriptor Ring Length (DMA_CH4_TXDESC_RING_LENGTH)

The Channel 4 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

11.7.6.1.553.1 Offset

Register	Offset
DMA_CH4_TXDESC_RING_LENGTH	132Ch

11.7.6.1.553.2 Diagram



11.7.6.1.553.3 Fields

Field	Description
31-10 —	Reserved
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. Synopsys recommends a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

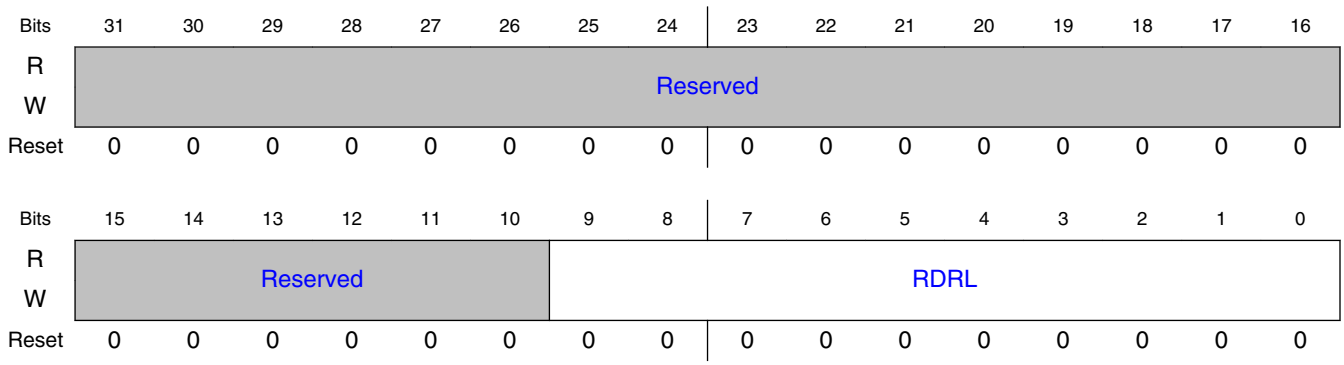
11.7.6.1.554 Channel 4 Rx Descriptor Ring Length (DMA_CH4_RXDESC_RING_LENGTH)

The Channel 4 Rx Descriptor Ring Length register contains the length of the Receive descriptor circular ring.

11.7.6.1.554.1 Offset

Register	Offset
DMA_CH4_RXDESC_RING_LENGTH	1330h

11.7.6.1.554.2 Diagram



11.7.6.1.554.3 Fields

Field	Description
31-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

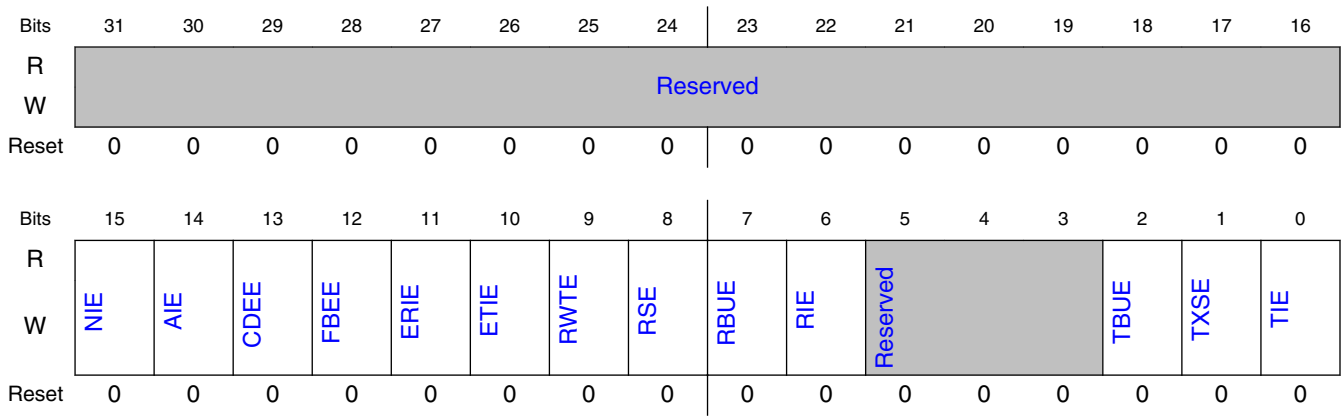
11.7.6.1.555 Channel 4 Interrupt Enable (DMA_CH4_INTERRUPT_ENABLE)

The Channel 4 Interrupt Enable register enables the interrupts reported by the Status register.

11.7.6.1.555.1 Offset

Register	Offset
DMA_CH4_INTERRUPT_ENABLE	1334h

11.7.6.1.555.2 Diagram



11.7.6.1.555.3 Fields

Field	Description
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0 - Normal Interrupt Summary is disabled 1 - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error

Table continues on the next page...

Field	Description
	When this bit is reset, the abnormal interrupt summary is disabled. 0 - Abnormal Interrupt Summary is disabled 1 - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0 - Context Descriptor Error is disabled 1 - Context Descriptor Error is enabled
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0 - Fatal Bus Error is disabled 1 - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0 - Early Receive Interrupt is disabled 1 - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0 - Early Transmit Interrupt is disabled 1 - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0 - Receive Watchdog Timeout is disabled 1 - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled.

Table continues on the next page...

Field	Description
	0 - Receive Stopped is disabled 1 - Receive Stopped is enabled
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0 - Receive Buffer Unavailable is disabled 1 - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0 - Receive Interrupt is disabled 1 - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0 - Transmit Buffer Unavailable is disabled 1 - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0 - Transmit Stopped is disabled 1 - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0 - Transmit Interrupt is disabled 1 - Transmit Interrupt is enabled

11.7.6.1.556 Channel 4 Receive Interrupt Watchdog Timer (DMA_CH4_RX_INTERRUPT_WATCHDOG_TIMER)

The Channel 4 Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the STATUS register.

11.7.6.1.556.1 Offset

Register	Offset
DMA_CH4_RX_INTERRUPT_WATCHDOG_TIMER	1338h

11.7.6.1.556.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														RWTU	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RWT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.556.3 Fields

Field	Description
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024 - 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15-8 —	Reserved

Table continues on the next page...

Field	Description
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the DMA_CH4_STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

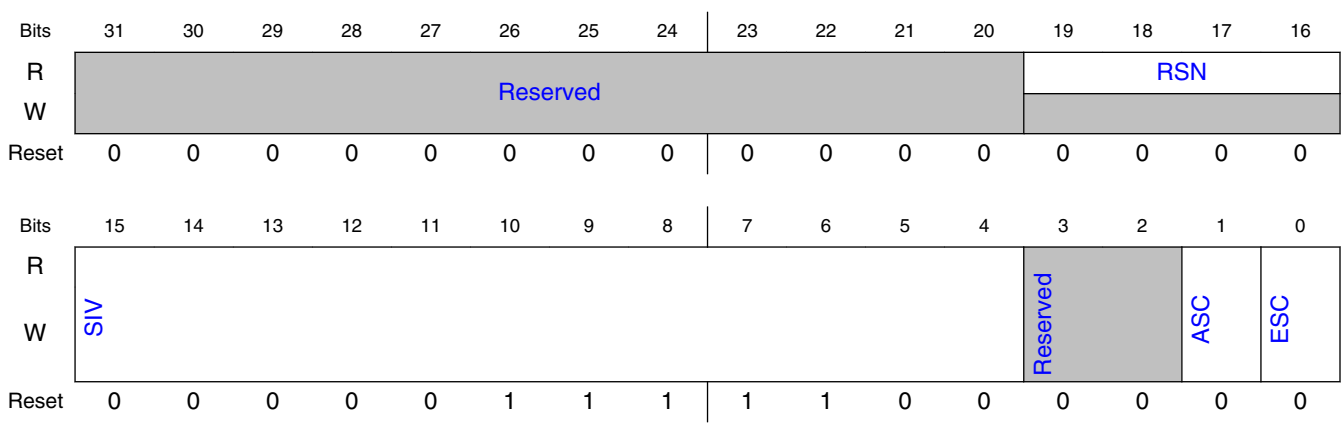
11.7.6.1.557 Channel 4 Slot Function Control and Status (DMA_CH4_SLOT_FUNCTION_CONTROL_STATUS)

The Channel 4 Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

11.7.6.1.557.1 Offset

Register	Offset
DMA_CH4_SLOT_FUNCTION_CONTROL_STATUS	133Ch

11.7.6.1.557.2 Diagram



11.7.6.1.557.3 Fields

Field	Description
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0 - Advance Slot Check is disabled 1 - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or - ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0 - Slot Comparison is disabled 1 - Slot Comparison is enabled

11.7.6.1.558 Channel 4 Current Application Transmit Descriptor (DMA_CH4_CURRENT_APP_TXDESC)

The Channel 4 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

11.7.6.1.558.1 Offset

Register	Offset
DMA_CH4_CURRENT_APP_TXDESC	1344h

11.7.6.1.558.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.558.3 Fields

Field	Description
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

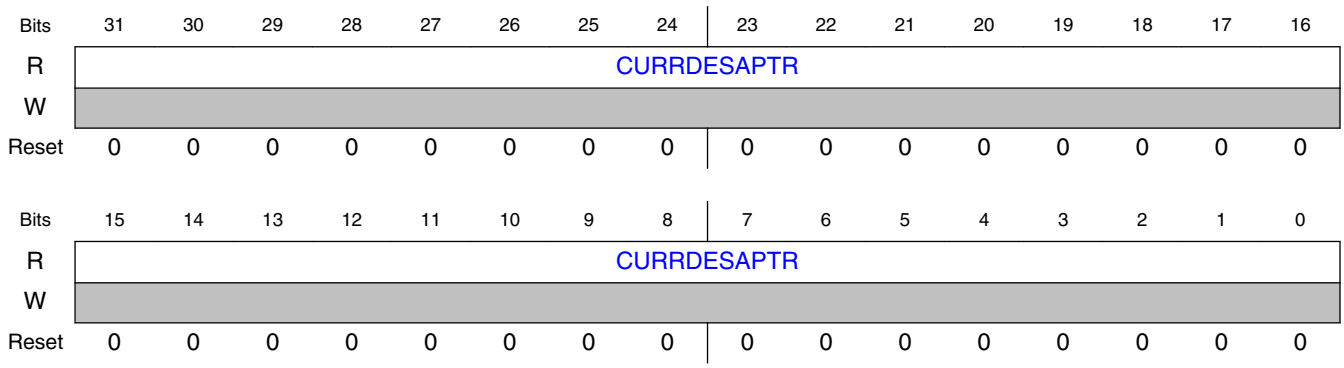
11.7.6.1.559 Channel 4 Current Application Receive Descriptor (DMA_CH4_CURRENT_APP_RXDESC)

The Channel 4 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

11.7.6.1.559.1 Offset

Register	Offset
DMA_CH4_CURRENT_APP_RXDESC	134Ch

11.7.6.1.559.2 Diagram



11.7.6.1.559.3 Fields

Field	Description
31-0 CURRDESAPTR	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

11.7.6.1.560 Channel 4 Current Application Transmit Buffer Address (DMA_CH4_CURRENT_APP_TXBUFFER)

The Channel 4 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

11.7.6.1.560.1 Offset

Register	Offset
DMA_CH4_CURRENT_APP_TXBUFFER	1354h

11.7.6.1.560.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.560.3 Fields

Field	Description
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

11.7.6.1.561 Channel 4 Current Application Receive Buffer Address (DMA_CH4_CURRENT_APP_RXBUFFER)

The Channel 4 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

11.7.6.1.561.1 Offset

Register	Offset
DMA_CH4_CURRENT_APP_RXBUFFER	135Ch

11.7.6.1.561.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.561.3 Fields

Field	Description
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

11.7.6.1.562 DMA Channel 4 Status (DMA_CH4_STATUS)

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

11.7.6.1.562.1 Offset

Register	Offset
DMA_CH4_STATUS	1360h

11.7.6.1.562.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										REB			TEB		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.562.3 Fields

Field	Description
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access

Table continues on the next page...

Field	Description
	- Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Normal Interrupt Summary status not detected 1 - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Abnormal Interrupt Summary status not detected 1 - Abnormal Interrupt Summary status detected

Table continues on the next page...

Field	Description
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Context Descriptor Error status not detected 1 - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Fatal Bus Error status not detected 1 - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Receive Interrupt status not detected 1 - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Early Transmit Interrupt status not detected 1 - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0 - Receive Watchdog Timeout status not detected

Table continues on the next page...

Field	Description
	1 - Receive Watchdog Timeout status detected
8 RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Process Stopped status not detected 1 - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode, the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Buffer Unavailable status not detected 1 - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Receive Interrupt status not detected 1 - Receive Interrupt status detected
5-3 —	Reserved
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DMA_CH4_DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Buffer Unavailable status not detected

Table continues on the next page...

Field	Description
	1 - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Process Stopped status not detected 1 - Transmit Process Stopped status detected
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0 - Transmit Interrupt status not detected 1 - Transmit Interrupt status detected

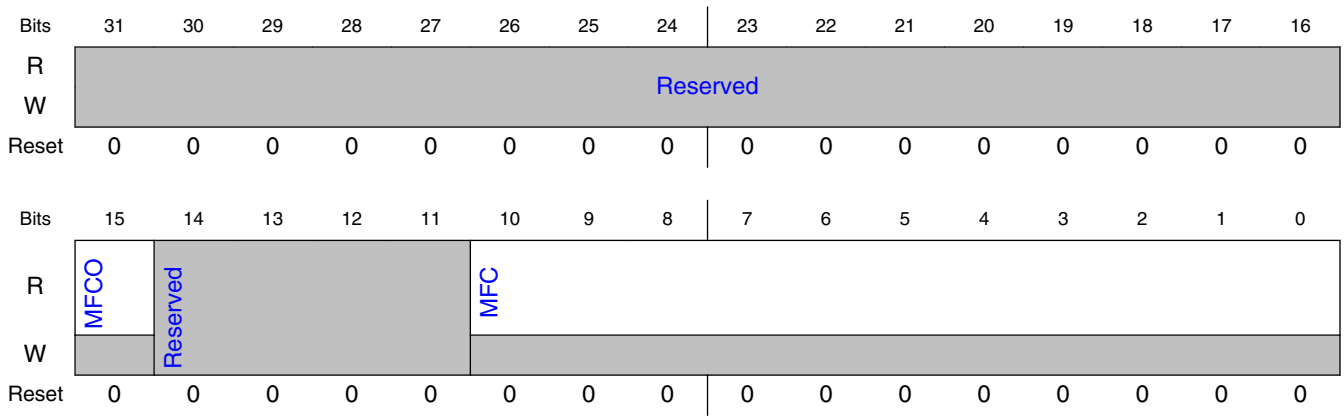
11.7.6.1.563 Channel 4 Missed Frame Counter (DMA_CH4_MISS_FRAME_CNT)

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in DMA_CH4_RX_CONTROL register.

11.7.6.1.563.1 Offset

Register	Offset
DMA_CH4_MISS_FRAME_CNT	1364h

11.7.6.1.563.2 Diagram



11.7.6.1.563.3 Fields

Field	Description
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Miss Frame Counter overflow not occurred 1 - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

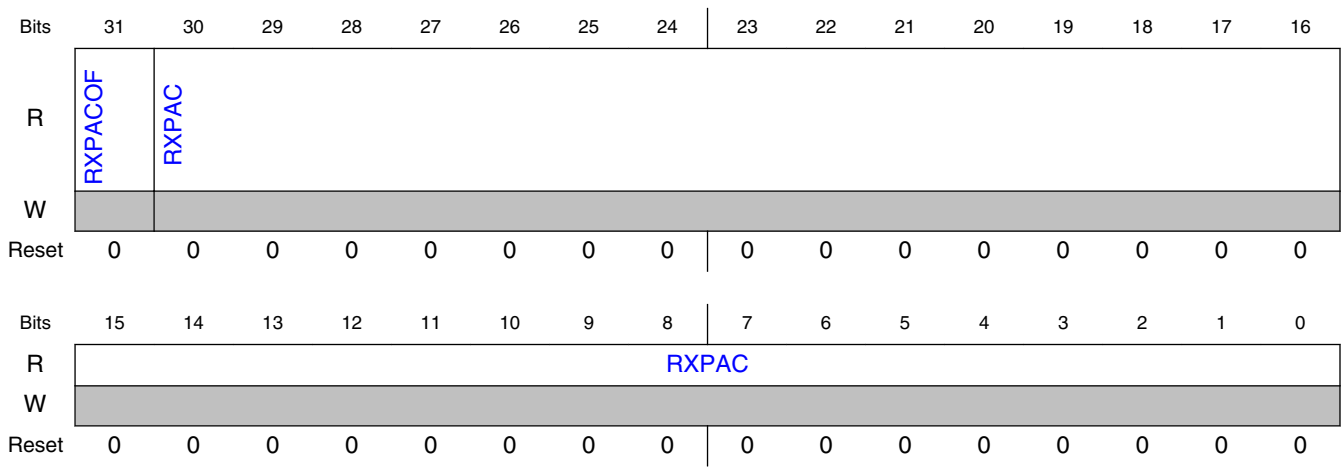
11.7.6.1.564 Channel 4 RXP Frames Accepted Counter (DMA_CH4_RXP_ACCEPT_CNT)

The Channel 4 RXP Frames Accepted Counter register provides the count of the number of frames accepted by Rx Parser.

11.7.6.1.564.1 Offset

Register	Offset
DMA_CH4_RXP_ACCEPT_CNT	1368h

11.7.6.1.564.2 Diagram



11.7.6.1.564.3 Fields

Field	Description
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0 - Rx Parser Accept Counter overflow not occurred 1 - Rx Parser Accept Counter overflow occurred
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

11.7.6.1.565 Channel 4 Receive ERI Counter (DMA_CH4_RX_ERI_CNT)

The Channel 4 Receive ERI Counter register provides the count of the number of times ERI was asserted.

11.7.6.1.565.1 Offset

Register	Offset
DMA_CH4_RX_ERI_CNT	136Ch

11.7.6.1.565.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				ECNT											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.7.6.1.565.3 Fields

Field	Description
31-12 —	Reserved
11-0 ECNT	ERI Counter When ERIC bit of DMA_CH4_RX_CONTROL register is set, this counter increments for burst transfer completed by the Rx DMA from the start of packet transfer. ERI Counter When ERIC bit of DMA_CH4_RX_CONTROL register is set, this counter increments for burst transfer completed by the Rx DMA from the start of packet transfer. This counter will get reset at the start of new packet.

11.8 FlexCAN

11.8.1 Overview

The FlexCAN module is a communication controller implementing the CAN protocol according to the ISO 11898-1 standard and CAN 2.0 B protocol specifications.

The CAN protocol was primarily designed to be used as a vehicle serial data bus, meeting the specific real-time processing and reliable operation requirements in the EMI environment of a vehicle. The FlexCAN module is a full implementation of the CAN protocol specification, the CAN with Flexible Data rate (CAN FD) protocol, and the CAN 2.0 version B protocol, which supports both standard and extended message frames and long payloads.

NOTE

Rx FIFO cannot be used in FD mode.

NOTE

FD mode is limited to Message Buffer (MB) mode only, the enhanced Rx FIFO with DMA support is not supported.

11.8.1.1 Block diagram

A general block diagram is shown in Figure 11-116, which describes the main subblocks implemented in the FlexCAN module.

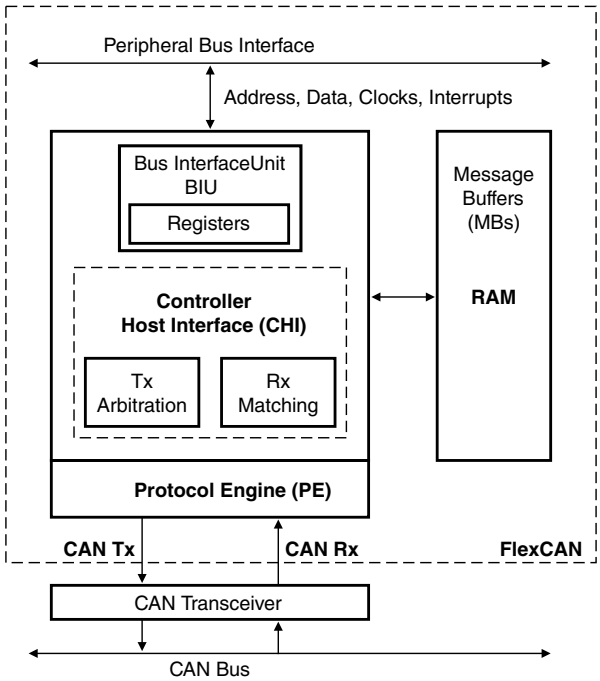


Figure 11-116. FlexCAN block diagram

The Protocol Engine (PE) submodule manages the serial communication on the CAN bus:

- Requesting RAM access for receiving and transmitting message frames
- Validating received messages
- Performing error handling
- Detecting CAN FD messages

The Controller Host Interface (CHI) submodule manages message buffer selection for reception and transmission, taking care of arbitration and ID matching algorithms for both CAN FD and non-CAN FD message formats.

The Bus Interface Unit (BIU) submodule controls access to and from the internal interface bus, in order to establish connection to the CPU and to other blocks. Clocks, address and data buses, interrupt outputs, DMA and test signals are accessed through the BIU.

11.8.1.2 Features

The FlexCAN module includes these distinctive features:

- Full implementation of the CAN with Flexible Data Rate (CAN FD) protocol specification and CAN protocol specification, Version 2.0 B
  - Standard data frames

  - Extended data frames
  - Zero to sixty four bytes data length
  - Content-related addressing
- Compliant with the ISO 11898-1 standard
- Flexible mailboxes configurable to store 0 to 8, 16, 32, or 64 bytes data length
- Each mailbox configurable as receive or transmit, all supporting standard and extended messages
- Individual Rx Mask registers per mailbox
- Full-featured Rx FIFO with storage capacity for up to six frames and automatic internal pointer handling with DMA support
- Transmission abort capability
- Flexible message buffers (MBs), totaling 64 message buffers of 8 bytes data length each, configurable as Rx or Tx
- Programmable clock source to the CAN Protocol Interface, either peripheral clock or oscillator clock
- RAM not used by reception or transmission structures can be used as general purpose RAM space
- Listen-Only mode capability
- Programmable Loop-Back mode supporting self-test operation
- Programmable transmission priority scheme: lowest ID, lowest buffer number, or highest priority
- Time stamp based on 16-bit free running timer
- Global network time, synchronized by a specific message
- Maskable interrupts
- Independence from the transmission medium (an external transceiver is assumed)
- Short latency time due to an arbitration scheme for high-priority messages
- Low power modes, with programmable wakeup on bus activity
- Transceiver delay compensation feature when transmitting CAN FD messages at faster data rates
- Remote request frames may be managed automatically or by software
- CAN bit time settings and configuration bits can only be written in Freeze mode
- Tx mailbox status (Lowest priority buffer or empty buffer)
- Identifier Acceptance Filter Hit Indicator (IDHIT) register for received frames
- SYNCH bit available in Error in Status 1 register to indicate that the module is synchronous with CAN bus
- CRC status for transmitted message
- Rx FIFO Global Mask register
- Selectable priority between mailboxes and Rx FIFO during matching process
- Powerful Rx FIFO ID filtering, capable of matching incoming IDs against either 128 extended, 256 standard, or 512 partial (8 bit) IDs, with up to 32 individual masking capability

- 100% backward compatibility with previous FlexCAN version
- Supports detection and correction of errors in memory read accesses

  Each byte of FlexCAN memory is associated to 5 parity bits, and the error correction mechanism ensures that in this 13-bit word, errors in one bit can be corrected (correctable errors) and errors in 2 bits can be detected but not corrected (non-correctable errors).

11.8.2 Functional description

The FlexCAN module is a CAN protocol engine with a very flexible mailbox system for transmitting and receiving CAN frames. The mailbox system is composed of a set of message buffers (MB) that store configuration and control data, time stamp, message ID, and data (see Message buffer structure).

For Classical CAN frames, simultaneous reception through FIFO and mailbox is supported. For CAN FD frames, reception is supported through mailboxes only. For mailbox reception, a matching algorithm makes it possible to store received frames only into MBs that have the same ID programmed in the ID field. A masking scheme makes it possible to match the ID programmed on the MB with a range of IDs on received CAN frames. For transmission, an arbitration algorithm decides the prioritization of MBs to be transmitted based on the message ID (optionally augmented by 3 local priority bits) or the MB ordering.

Before proceeding with the functional description, an important concept must be explained. A message buffer is said to be active at a given time if it can participate in both the matching and arbitration processes. An Rx MB with a 0b code is inactive (see Table 11-186). Similarly, a Tx MB with a 1000b or 1001b code is also inactive (see Table 11-187).

The FlexCAN module is also able to receive and transmit messages in CAN FD format. The message buffers are sized to adequately store the quantity of data bytes selected by the FDCTRL[MBDSRn] fields. The quantity of FD MBs available for a given quantity of data bytes is described in the FDCTRL register. See also FlexCAN memory partition for CAN FD.

11.8.2.1 Modes of operation

The FlexCAN module has these functional modes:

- Normal mode (User or Supervisor):

In Normal mode, the module operates receiving and/or transmitting message frames, errors are managed normally, and all CAN Protocol functions are enabled. User and Supervisor modes differ in the access to some restricted control registers.

- Freeze mode:

  Freeze mode is enabled when MCR[FRZ] is asserted. If enabled, Freeze mode is entered when MCR[HALT] is set or when is requested at chip level and MCR[FRZ_ACK] is asserted by the FlexCAN. In this mode, no transmission or reception of frames is done and synchronicity to the CAN bus is lost. See Freeze mode for more information.

- Loop-Back mode:

  The module enters this mode when CTRL1[LPB] is asserted. In this mode, FlexCAN performs an internal loop back that can be used for self-test operation. The bit stream output of the transmitter is internally fed back to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic '1'). FlexCAN behaves as it normally does when transmitting and treats its own transmitted message as a message received from a remote node. In this mode, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated.

- Listen-Only mode:

  The module enters this mode when CTRL1[LOM] is asserted. In this mode, transmission is disabled, all error counters are frozen, and the module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station will be received. If FlexCAN detects a message that has not been acknowledged, it will flag a BIT0 error (without changing the REC), as if it was trying to acknowledge the message.

- CAN FD Active mode:

  In this mode, FlexCAN is capable of transmitting and receiving all messages formatted according to the CAN FD Standard (2.0) and 2.0B Protocol in an interleaved fashion. The CPU can set the FlexCAN into CAN FD Active mode by configuring MCR[FDEN] in Freeze mode.

  It is important to know which features are available in CAN FD active mode. This table lists the differences between FD and classical modes.

Table 11-159. Differences between CAN classical and CAN FD

Feature	CAN classical	CAN FD
Rx FIFO DMA	Yes	No
Rx FIFO	Yes	No

For low-power operation, the FlexCAN module has:

- Module Disable mode:

  This low-power mode is entered when MCR[MDIS] is asserted by the CPU and MCR[LPM_ACK] is asserted by FlexCAN. When disabled, the module issues a request to disable the clocks to the CAN Protocol Engine and Controller Host Interface submodules. Exit from this mode is done by negating MCR[MDIS]. See Module Disable mode for more information.

- Doze mode:

  This low power mode is entered when MCR[DOZE] is asserted and Doze mode is requested at chip level, and MCR[LPM_ACK] is asserted by FlexCAN. When in Doze mode, the module issues a request to disable the clocks to the CAN Protocol Engine and the CAN Controller-Host Interface submodules. Exit from this mode happens when MCR[DOZE] is negated, when the chip is removed from Doze mode, or when activity is detected on the CAN bus and the Self Wake Up mechanism is enabled. See Doze mode for more information.

- Stop mode:

  This low power mode is entered when Stop mode is requested at chip level and MCR[LPM_ACK] is asserted by the FlexCAN. When in Stop Mode, the module puts itself in an inactive state and then informs the CPU that the clocks can be shut down globally. Exit from this mode happens when the Stop mode request is removed, or when activity is detected on the CAN bus and the Self Wake Up mechanism is enabled. See Stop mode for more information.

11.8.2.1.1 Modes of operation details

The FlexCAN module has functional modes and low-power modes. See Modes of operation for an introductory description of all the modes of operation. The following sub-sections contain functional details on Freeze mode and the low-power modes.

CAUTION

"Permanent Dominant" failure on CAN Bus line is not supported by FlexCAN. If a Low-Power request or Freeze

mode request is done during a "Permanent Dominant", the corresponding acknowledge can never be asserted.

11.8.2.1.1.1 Freeze mode

This mode is requested either by the CPU through the assertion of MCR[HALT] or when the chip is put into Debug mode. In both cases it is also necessary that MCR[FRZ] be asserted and the module not be in a low-power mode. This mode is also requested by FlexCAN through the automatic assertion of both MCR[HALT] and MCR[FRZ] when MECR[NCEFAFRZ] is set and a non-correctable error is detected in a memory read access performed by FlexCAN internal processes (see Response to errors).

The acknowledgement is obtained through the assertion by the FlexCAN of MECR[FRZ_ACK]. The CPU must only consider the FlexCAN in Freeze mode when both request and acknowledgement conditions are satisfied.

When Freeze mode is requested, FlexCAN does the following:

- Waits to be in either Intermission, Passive Error, Bus Off, or Idle state.
- Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in does not prevent entering Freeze mode.
- Ignores the Rx input pin and drives the Tx pin as recessive.
- Stops the prescaler, thus halting all CAN protocol activities.
- Grants write access to the Error Counters register, which is read-only in other modes.
- Sets MCR[NOTRDY] and MCR[FRZACK].

After requesting Freeze mode, the user must wait for MCR[FRZ_ACK] to be asserted before executing any other action; otherwise FlexCAN may operate in an unpredictable way. In Freeze mode, all memory-mapped registers are accessible, except for CTRL1[CLKSRC], which can be read but cannot be written.

Exiting Freeze mode is done in one of the following ways:

- CPU negates MCR[FRZ].
- The chip is removed from Debug mode and/or the HALT bit is negated.

MCR[FRZ_ACK] is negated after the protocol engine recognizes the negation of the freeze request. When out of Freeze mode, FlexCAN tries to resynchronize to the CAN bus by waiting for 11 consecutive recessive bits.

11.8.2.1.1.2 Module Disable mode

This low-power mode is normally used to temporarily disable a complete FlexCAN block, with no power consumption. It is requested by the CPU through the assertion of MCR[MDIS], and the acknowledgement is obtained through the assertion by the FlexCAN of MCR[LPMACK]. The CPU must only consider the FlexCAN in Disable mode when both request and acknowledgement conditions are satisfied.

If the module is disabled during Freeze mode, it requests to disable the clocks to the PE and CHI submodules, sets MCR[LPMACK] and negates MCR[FRZACK].

If the module is disabled during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and then checks it to be recessive.
- Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive.
- Shuts down the clocks to the PE and CHI submodules
- Sets MCR[NOTRDY] and MCR[LPMACK].

The Bus Interface Unit continues to operate, enabling the CPU to access memory-mapped registers, except the Rx Mailboxes Global Mask registers, the Rx Buffer 14 Mask register, the Rx Buffer 15 Mask register,. The message buffers, the Rx Individual Mask registers, and the reserved words within RAM may not be accessed when the module is in Disable mode. Exiting from this mode is done by negating the MDIS bit by the CPU, which causes the FlexCAN to request to resume the clocks and negate the LPMACK bit after the CAN protocol engine recognizes the negation of disable mode requested by the CPU.

11.8.2.1.1.3 Doze mode

This is a system low power mode in which the CPU bus is kept alive and a global Doze mode request is sent to all peripherals asking them to enter low-power mode. When Doze mode is globally requested, MCR[DOZE] needs to have been asserted previously for Doze mode to be triggered. The acknowledgement is obtained through the assertion by the FlexCAN of MCR[LPMACK]. The CPU must only consider the FlexCAN to be in Doze mode when both request and acknowledgement conditions are satisfied.

If Doze mode is triggered during Freeze mode, FlexCAN requests to shut down the clocks to the PE and CHI submodules, sets MCR[LPMACK] and negates MCR[FRZACK]. If Doze mode is triggered during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive.
- Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive.
- Shuts down the clocks to the PE and CHI submodules.
- Sets MCR[NOTRDY] and MCR[LPMACK].

The Bus Interface Unit continues to operate, enabling the CPU to access memory-mapped registers, except the Rx Mailboxes Global Mask registers, the Rx Buffer 14 Mask register, the Rx Buffer 15 Mask register. The the message buffers, the Rx Individual Mask registers, and the reserved words within RAM may not be accessed when the module is in Doze mode.

Exiting Doze mode is done in one of the following ways:

- CPU removing the Doze mode request
- CPU negating MCR [DOZE]
- Self Wake mechanism

In the Self Wake mechanism, if MCR[SLFWAK] was set at the time FlexCAN entered Doze mode, then upon detection of a recessive to dominant transition on the CAN bus, FlexCAN negates the DOZE bit, requests to resume its clocks and negates the LPMACK after the CAN protocol engine recognizes the negation of the Doze mode request. It also sets ESR [WAKINT] and, if enabled by MCR[WAKMSK], generates a Wake-Up interrupt to the CPU. FlexCAN will then wait for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it will not receive the frame that woke it up. The following table details the effect of SLFWAK and WAKMSK upon wakeup from Doze mode.

Table 11-160. Wakeup from Doze mode

SLFWAK	WAKINT	WAKMSK	FlexCAN clocks enabled	Wakeup interrupt generated
0	—	—	No	No

Table continues on the next page...

Table 11-160. Wakeup from Doze mode (continued)

SLFWAK	WAKINT	WAKMSK	FlexCAN clocks enabled	Wakeup interrupt generated
0	—	—	No	No
1	0	0	No	No
1	0	1	No	No
1	1	0	Yes	No
1	1	1	Yes	Yes

The sensitivity to CAN bus activity can be modified by applying a low-pass filter function to the Rx CAN input line when in Doze mode. (See MCR[WAKSRC].) This feature can be used to protect FlexCAN from waking up due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic interference within noisy environments.

11.8.2.1.1.4 Stop mode

This is a system low-power mode in which all chip clocks can be stopped for maximum power savings. The Stop mode is globally requested by the CPU and acknowledgement is obtained through the assertion by the FlexCAN of a Stop Acknowledgement signal. The CPU must only consider the FlexCAN in Stop mode when both request and acknowledgement conditions are satisfied.

If FlexCAN receives the global Stop mode request during Freeze mode, it sets MCR[LPMACK], negates MCR[FRZACK], and then sends the Stop Acknowledge signal to the CPU, in order to shut down the clocks globally.

If Stop mode is requested during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive
- Sets MCR[NOTRDY] and MCR[LPMACK]
- Sends a Stop Acknowledge signal to the CPU, so that it can shut down the clocks globally

Stop mode is exited when the CPU resumes the clocks and removes the Stop mode request. This can be as a result of the Self Wake mechanism.

In the Self Wake mechanism, if MCR[SLFWAK] was set at the time FlexCAN entered Stop mode, then upon detection of a recessive to dominant transition on the CAN bus, FlexCAN sets ESR[WAKINT] and, if enabled by MCR[WAKMSK], generates a Wake Up interrupt to the CPU. Upon receiving the interrupt, the CPU should resume the clocks and remove the Stop mode request. FlexCAN will then wait for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it will not receive the frame that woke it up. The following table details the effect of MCR[SLFWAK] and MCR[WAKMSK] upon wakeup from Stop mode. Note that wakeup from Stop mode only works when both bits are asserted.

After the CAN protocol engine recognizes the negation of the Stop mode request, the FlexCAN negates MCR[LPMACK]. FlexCAN will then wait for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it will not receive the frame that woke it up.

Table 11-161. Wakeup from Stop mode

SLFWAK	WAKINT	WAKMSK	Chip clocks enabled	Wakeup interrupt generated
0	—	—	No	No
0	—	—	No	No
1	0	0	No	No
1	0	1	No	No
1	1	0	No	No
1	1	1	Yes	Yes

The sensitivity to CAN bus activity can be modified by applying a low-pass filter function to the Rx CAN input line when in Stop mode. (See the description in MCR[WAKSRC].) This feature can be used to protect FlexCAN from waking up due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic interference within noisy environments.

11.8.2.2 Transmit process

To transmit a CAN frame, the CPU must prepare a message buffer for transmission by executing the following procedure:

1. Check whether the respective interrupt bit is set and clear it.
2. If the MB is active (transmission pending), write the ABORT code (1001b) to the CODE field of the Control and Status word to request an abort of the transmission.
3. Wait for the corresponding IFLAG bit to be asserted by polling the IFLAG register, or by the interrupt request if enabled by the respective IMASK bit.

4. Read back the CODE field to check if the transmission was aborted or transmitted (see Transmission abort mechanism).
5. Clear the corresponding interrupt flag.
6. Write the ID register (containing the local priority if enabled via MCR[LPRIO_EN]).
7. Write payload data bytes.
8. Configure the Control and Status word with the desired configuration.
   a. Set ID type via MB_CS[IDE].
   b. Set Remote Transmission Request (if needed) via MB_CS[RTR].
   c. If MCR[FDEN] is enabled, also configure the MB_CS[EDL] and MB_CS[BRS] fields. For details about the relationship between the written value and transmitted value of the MB_CS[ESI] field, see Table 11-165.[2]
   d. Set Data Length Code in bytes via MB_CS[DLC]. See Table 11-188 for detailed information.
   e. Activate the message buffer to transmit the CAN frame by setting MB_CS[CODE] to Ch.

NOTE

It is strongly recommended that all the fields in MB_CS word be configured in only one 32-bit write operation to maximize software performance. If the fields are configured in separate writes, the MB_CS[CODE] must be the last write in the C/S word.

When the MB is activated, it participates in the arbitration process and is eventually transmitted according to its priority. When the DLC value stored in the MB selected for transmission is larger than the respective MB payload size, FlexCAN adds the necessary number of bytes with constant CCh pattern to complete the expected DLC.

At the end of the successful transmission:

- The value of the free running timer is written into the Time Stamp field.
- The CODE field in the Control and Status word is updated.
- Both CRC and FDCRC registers are updated.
- A status flag is set in the Interrupt Flag register.
- An interrupt is generated if allowed by the corresponding Interrupt Mask Register bit.

The new CODE field after transmission depends on the code that was used to activate the MB (see Table 11-186 and Table 11-187 in Message buffer structure).

2. The ESI field does not need to be written, and will automatically be transmitted dominant by error active nodes and recessive by error passive nodes. Note that there is an exception if the FlexCAN is operating as a network gateway: In that case, the CPU writes the MB_CS[ESI] bit according to the error status of the node which sent the message.

When the Abort feature is enabled (MCR[AEN] is asserted), after the Interrupt flag is asserted for a mailbox configured as transmit buffer, the mailbox is blocked. Therefore the CPU is not able to update it until the Interrupt Flag is negated by the CPU. This means that the CPU must clear the corresponding IFLAG bit before starting to prepare this MB for a new transmission or reception.

NOTE

If backwards compatibility is desired (MCR[AEN] is negated), write the INACTIVE code (1000b) to the CODE field to inactivate the MB. However, in this case the pending frame may be transmitted without notification (see Mailbox inactivation).

11.8.2.3 Arbitration process

The arbitration process scans the mailboxes, searching for the Tx mailbox that holds the message to be sent in the next opportunity. This mailbox is called the arbitration winner.

The scan starts from the lowest number mailbox and runs toward the higher ones.

The arbitration process is triggered in the following events:

- From the CRC field of the CAN frame. The start point depends on the value of CTRL2[TASD].
- During the Error Delimiter field of a CAN frame.
- During the Overload Delimiter field of a CAN frame.
- When the winner is inactivated and the CAN bus has still not reached the first bit of the Intermission field.
- When there is a CPU write to the C/S word of a winner MB and the CAN bus has still not reached the first bit of the Intermission field.
- When CHI is in Idle state and the CPU writes to the C/S word of any MB.
- When FlexCAN exits Bus Off state.
- Upon leaving Freeze mode or Low Power mode.

If the arbitration process does not manage to evaluate all mailboxes before the CAN bus has reached the first bit of the Intermission field, the temporary arbitration winner is invalidated and the FlexCAN will not compete for the CAN bus in the next opportunity.

The arbitration process selects the winner among the active Tx mailboxes at the end of the scan according to the settings of both CTRL1[LBUF] and MCR[LPRIOEN].

11.8.2.3.1 Lowest-number mailbox first

If CTRL1[LBUF] is asserted, the first (lowest number) active Tx mailbox found is the arbitration winner. MCR[LPRIOEN] has no effect when CTRL1[LBUF] is asserted.

11.8.2.3.2 Highest-priority mailbox first

If CTRL1[LBUF] is negated, then the arbitration process searches the active Tx mailbox with the highest priority, which means that this mailbox's frame would have a higher probability to win the arbitration on CAN bus when multiple external nodes compete for the bus at the same time.

The sequence of bits considered for this arbitration is called the *arbitration value* of the mailbox. The highest-priority Tx mailbox is the one that has the lowest arbitration value among all Tx mailboxes.

If two or more mailboxes have equivalent arbitration values, the mailbox with the lowest number is the arbitration winner.

The composition of the arbitration value depends on the MCR[LPRIOEN] setting.

11.8.2.3.2.1 Local priority disabled

If MCR[LPRIOEN] is negated the arbitration value is built in the exact sequence of bits as they would be transmitted in a CAN frame (see the following table) in such a way that the local priority is disabled.

Table 11-162. Composition of the arbitration value when local priority is disabled

Format	Mailbox arbitration value (32 bits)				
Standard (IDE = 0)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	- (18 bits)	- (1 bit)
Extended (IDE = 1)	Extended ID[28:18] (11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

11.8.2.3.2.2 Local priority enabled

To enable local priority, MCR[LPRIOEN] must be asserted. In this case the mailbox PRIO field is included at the very left of the arbitration value (see the following table).

Table 11-163. Composition of the arbitration value when local priority is enabled

Format	Mailbox arbitration value (35 bits)					
Standard (IDE = 0)	PRIO (3 bits)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	- (18 bits)	- (1 bit)
Extended (IDE = 1)	PRIO (3 bits)	Extended ID[28:18] (11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

Because the PRIO field is the most significant part of the arbitration value, mailboxes with low PRIO values have higher priority than mailboxes with high PRIO values regardless of the rest of their arbitration values.

Note that the PRIO field is not part of the frame on the CAN bus. Its purpose is only to affect the internal arbitration process.

11.8.2.3.3 Arbitration process (continued)

After the arbitration winner is found, its content is copied to a hidden auxiliary MB called Tx Serial Message Buffer (Tx SMB), which has the same structure as a normal MB but is not user accessible. This operation is called move-out and after it is done, write access to the C/S word of the corresponding MB is blocked (if MCR[AEN] is asserted). Write access is restored in the following events:

- After the MB is transmitted and the corresponding IFLAG bit is cleared by the CPU.
- FlexCAN enters Freeze mode or Bus Off.
- FlexCAN loses the bus arbitration or there is an error during the transmission.

At the first opportunity window on the CAN bus, the message on the Tx SMB is transmitted according to the CAN protocol rules.

Arbitration process can be triggered in the following situations:

- During Rx and Tx frames from CAN CRC field to end of frame. CTRL2[TASD] value may be changed to optimize the arbitration start point.
- During CAN Bus Off state from TX_ERR_CNT=124 to 128. CTRL2[TASD] value may be changed to optimize the arbitration start point.
- During C/S write by CPU in Bus Idle. First C/S write starts arbitration process, and a second C/S write during this same arbitration restarts the process. If other C/S writes are performed, Tx arbitration process is pending. If there is no arbitration winner after the arbitration process has finished, then the TX arbitration machine begins a new arbitration process. If there is a pending arbitration and Bus Idle state starts, then an arbitration process is triggered. In this case the first and second C/S write in Bus Idle will not restart the arbitration process. It is possible that there is not enough time to finish arbitration in Wait For Bus Idle state and the next state is Idle. In this case

the scan is not interrupted, and it is completed during Bus Idle state. During this arbitration C/S write does not cause arbitration restart.
- Arbitration winner deactivation during a valid arbitration window.
- Upon exiting Freeze mode (first bit of the Wait For Bus Idle state). If there is a re-synchronization during Wait For Bus Idle, the arbitration process is restarted.

Arbitration process stops in the following situations:

- All mailboxes were scanned.
- A Tx active mailbox is found if lowest buffer feature is enabled.
- Arbitration winner inactivation or abort during any arbitration process.
- There was not enough time to finish Tx arbitration process (for instance, when a deactivation was performed near the end of frame). In this case arbitration process is pending.
- Error or overload flag in the bus.
- Low Power or Freeze mode request in Idle state.

Arbitration is considered pending as described below:

- It was not possible to finish arbitration process in time.
- C/S write during arbitration if write is performed in a MB whose number is lower than the Tx arbitration pointer.
- Any C/S write if there is no Tx arbitration process in progress.
- Rx Match has just updated a Rx code to Tx code.
- Entering Bus Off state.

C/S write during arbitration has the following effect:

- If C/S write is performed in the arbitration winner, a new process is restarted immediately.
- If C/S write is performed in a MB whose number is higher than the Tx arbitration pointer, the ongoing arbitration process will scan this MB as normal.

11.8.2.4 Receive process

To be able to receive CAN frames into a mailbox, the CPU must prepare it for reception by executing the following steps:

1. If the mailbox is active (either Tx or Rx) inactivate the mailbox (see Mailbox inactivation), preferably with a safe inactivation (see Transmission abort mechanism).
2. Write the ID word into the mailbox.

3. Write the EMPTY code (0100b) to the CODE field of the Control and Status word to activate the mailbox. No setup is required for EDL, BRS, and ESI bits—they are overwritten by the respective bit fields in the received message.

After the MB is activated, it will be able to receive frames that match the programmed filter. At the end of a successful reception, the mailbox is updated by the move-in process (see Move-in) as follows:

1. The received data field (8 bytes at most for Classical CAN message format and up to 64 bytes for CAN FD message format) is stored.
2. The received Identifier field is stored.
3. The value of the Free Running Timer at the time of the second bit of frame's Identifier field is written into the mailbox's Time Stamp field.
4. The received SRR, IDE, RTR, EDL, BRS, ESI, and DLC fields are stored.
5. The CODE field in the Control and Status word is updated (see Table 11-186 and Table 11-187 in Section Message buffer structure).
6. A status flag is set in the Interrupt Flag Register and an interrupt is generated if allowed by the corresponding Interrupt Mask Register bit.

The recommended way for the CPU to service (read) the frame received in a mailbox is by the following procedure:

1. Read the Control and Status word of that mailbox.
2. Check if the BUSY bit is deasserted, indicating that the mailbox is locked. Repeat step 1) while it is asserted. See Mailbox lock mechanism.
3. Read the contents of the mailbox. After the mailbox is locked, its contents won't be modified by FlexCAN move-in processes. See Move-in.
4. Acknowledge the proper flag at IFLAG registers.
5. Read the free running timer to unlock the mailbox.

The CPU should poll for frame reception by the status flag bit for the specific mailbox in one of the IFLAG registers and not by the CODE field of that mailbox. Polling the CODE field does not work because after a frame is received and the CPU services the mailbox (by reading the C/S word followed by unlocking the mailbox), the CODE field will not return to EMPTY. It will remain FULL, as explained in Table 11-186. If the CPU tries to work around this behavior by writing to the C/S word to force an EMPTY code after reading the mailbox without a prior safe inactivation, a newly received frame matching the filter of that mailbox may be lost.

CAUTION

In summary: never do polling by reading directly the C/S word of the mailboxes. Instead, read the IFLAG registers.

Note that the received frame's Identifier field is always stored in the matching mailbox, thus the contents of the ID field in a mailbox may change if the match was due to masking. When MCR[SRXDIS] is asserted, FlexCAN will not store frames transmitted by itself in any MB, even if it contains a matching Rx mailbox, and no interrupt flag or interrupt signal will be generated. Otherwise, when MCR[SRXDIS] is deasserted, FlexCAN can receive frames transmitted by itself if a matching Rx mailbox exists.

When MCR[DMA] is asserted, upon receiving a frame in FIFO, IFLAG1[BUF5I] generates a DMA request and does not generate a CPU interrupt (see Rx FIFO under DMA operation). The IMASK1 bits in Rx FIFO region are not used.

The DMA controller must service the received frame using the following procedure:

1. Read the Control and Status word (read 80h address, optional).
2. Read the ID field (read 84h address, optional).
3. Read all data bytes (start read at 88h address, optional).

11.8.2.5 Matching process

The matching process scans the MB memory looking for Rx MBs programmed with the same ID as the one received from the CAN bus. . The matching starts from the lowest number message buffer toward the higher ones.

For Rx FIFO see Rx FIFO.

As the frame is being received, it is stored in a hidden auxiliary MB called Rx Serial Message Buffer (Rx SMB).

The matching process start point depends on the following conditions:

- If the received frame is a remote frame, the start point is the CRC field of the frame.
- If the received frame is a data frame with DLC field equal to zero, the start point is the CRC field of the frame.
- If the received frame is a data frame with DLC field different than zero, the start point is the DATA field of the frame.

If a matching ID is found in one of the mailboxes, the contents of the Rx SMB are transferred to to the matched mailbox by the move-in process. If any CAN protocol error is detected then no match results are transferred to the matched mailbox at the end of reception.

The matching process scans all matching elements of the active Rx mailboxes (CODE is EMPTY, FULL, OVERRUN, or RANSWER) in search of a successful comparison with the matching elements of the Rx SMB that is receiving the frame on the CAN bus. The Rx SMB has the same structure as a mailbox. The reception structures (mailboxes) associated with the matching elements that had a successful comparison are the *matched structures*. The *matching winner* is selected at the end of the scan among those matched structures and depends on conditions described ahead. See the following table.

Table 11-164. Matching architecture

Structure	SMB[RTR]	CTRL2[RRS]	CTRL2[EACEN]	MB[IDE]	MB[RTR]	MB[ID[1]]	MB[CODE]
Mailbox	0	—	0	cmp[2]	no_cmp[3]	cmp_msk[4]	EMPTY or FULL or OVERRUN
Mailbox	0	—	1	cmp_msk	cmp_msk	cmp_msk	EMPTY or FULL or OVERRUN
Mailbox	1	0	—	cmp	no_cmp	cmp	RANSWER
Mailbox	1	1	0	cmp	no_cmp	cmp_msk	EMPTY or FULL or OVERRUN
Mailbox	1	1	1	cmp_msk	cmp_msk	cmp_msk	EMPTY or FULL or OVERRUN

1. For mailbox structure, If SMB[IDE] is asserted, the ID is 29 bits (ID Standard + ID Extended). If SMB[IDE] is negated, the ID is only 11 bits (ID Standard). For FIFO structure, the ID depends on IDAM.
2. cmp: Compares the Rx SMB contents with the MB contents regardless the masks.
3. no_cmp: The Rx SMB contents are not compared with the MB contents.
4. cmp_msk: Compares the Rx SMB contents with MB contents taking into account the masks.

A reception structure is free-to-receive when any of the following conditions is satisfied:

- The CODE field of the mailbox is EMPTY.
- The CODE field of the mailbox is either FULL or OVERRUN and it has already been serviced (the C/S word was read by the CPU and unlocked as described in Mailbox lock mechanism).
- The CODE field of the mailbox is either FULL or OVERRUN and an inactivation (see Mailbox inactivation) is performed.

The scan order for mailboxes is from the matching element with lowest number to the higher ones.

The matching winner search for mailboxes is affected by MCR[IRMQ]. If it is negated, the matching winner is the first matched mailbox regardless if it is free-to-receive or not. If it is asserted, the matching winner is selected according to the priority below:

1. the first free-to-receive matched mailbox;

2. the last non free-to-receive matched mailbox.

If a non-safe mailbox inactivation (see Mailbox inactivation) occurs during matching process and the mailbox inactivated is the temporary matching winner, then the temporary matching winner is invalidated. The matching elements scan is not stopped nor restarted—it continues normally. The consequence is that the current matching process works as if the matching elements compared before the inactivation did not exist, therefore a message may be lost.

Suppose, for example, that IRMQ is enabled, there are two MBs with the same ID, and FlexCAN starts receiving messages with that ID. Let us say that these MBs are the second and the fifth in the array. When the first message arrives, the matching algorithm finds the first match in MB number 2. The code of this MB is EMPTY, so the message is stored there. When the second message arrives, the matching algorithm finds MB number 2 again, but it is not "free-to-receive", so it keeps looking, finds MB number 5 and stores the message there. If yet another message with the same ID arrives, the matching algorithm finds out that there are no matching MBs that are "free-to-receive", so it decides to overwrite the last matched MB, which is number 5. In doing so, it sets the CODE field of the MB to indicate OVERRUN.

The ability to match the same ID in more than one MB can be exploited to implement a reception queue to allow more time for the CPU to service the MBs. By programming more than one MB with the same ID, received messages are queued into the MBs. The CPU can examine the Time Stamp field of the MBs to determine the order in which the messages arrived.

Matching to a range of IDs is possible by using ID acceptance masks. FlexCAN supports individual masking per MB (see the description of Rx Individual Mask Registers (RXIMR0 - RXIMR63)). During the matching algorithm, if a mask bit is asserted, then the corresponding ID bit is compared. If the mask bit is negated, the corresponding ID bit is a "don't care". Note that the Individual Mask Registers are implemented in RAM, so they are not initialized out of reset. Also, they can only be programmed when the module is in Freeze mode; otherwise, they are blocked by hardware.

FlexCAN also supports an alternate masking scheme with only four mask registers (RXMGMASK, RX14MASK, and RX15MASK) for backwards compatibility with legacy applications. This alternate masking scheme is enabled when the IRMQ bit in the MCR register is negated.

11.8.2.6 Move process

There are two types of move process: move-in and move-out.

11.8.2.6.1 Move-in

The move-in process is the copy of a message received by an Rx SMB to an Rx mailbox that has matched it. Each Rx SMB has its own move-in process, but only one is performed at a given time as described ahead. The move-in starts only when the message held by the Rx SMB has a corresponding matching winner (see Matching process) and all of the following conditions are true:

- The CAN bus has either reached or already gone past:
  - The second bit of Intermission field next to the frame that carried the message that is in the Rx SMB.
  - The first bit of an overload frame next to the frame that carried the message that is in the Rx SMB.
- There is no ongoing matching process.
- The destination mailbox is not locked by the CPU.
- There is no ongoing move-in process from another Rx SMB. If more than one move-in process is to be started at the same time, both are performed and the newest substitutes for the oldest.

The term pending move-in is used throughout the documentation and stands for a move-to-be that still does not satisfy all of the aforementioned conditions.

The move-in is cancelled and the Rx SMB is able to receive another message if any of the following conditions is satisfied:

- The destination mailbox is inactivated after the CAN bus has reached the first bit of Intermission field next to the frame that carried the message and its matching process has finished.
- There is a previous pending move-in to the same destination mailbox.
- The Rx SMB is receiving a frame transmitted by the FlexCAN itself and self-reception is disabled (MCR[SRXDIS] is asserted).
- Any CAN protocol error is detected.

Note that the pending move-in is not cancelled if the module enters Freeze or Low-Power mode. It stays on hold, only waiting for Freeze and Low-Power mode to be exited and the module to be unlocked. If an MB is unlocked during Freeze mode, the move-in happens immediately.

The move-in process is the execution by the FlexCAN of the following steps:

1. Read all data words from the Rx SMB in accordance with the selected payload size for the Rx storage element.
2. Write all data words to the Rx mailbox in accordance with the selected payload size for the Rx storage element. If the data size of the storage element is smaller than the

original payload size described in the message's DLC field, the payload is truncated and the high order bytes that do not fit the destination size are lost.
3. Read the Control/Status and ID words from the Rx SMB.
4. Write Control/Status and ID words to the Rx mailbox, and update the CODE field.

The move-in process is not atomic, in such a way that it is immediately cancelled by the inactivation of the destination mailbox (see Mailbox inactivation). In this case the mailbox may remain partially updated, thus incoherent.

The BUSY Bit (least significant bit of the CODE field) of the destination message buffer is asserted while the move-in is being performed to alert the CPU that the message buffer content is temporarily incoherent.

11.8.2.6.2 Move-out

The move-out process is the copy of the content from a Tx Mailbox to the Tx SMB when a message for transmission is available (see Arbitration process). The move-out occurs in the following conditions:

- The first bit of Intermission field
- During Bus Off state when TX Error Counter is in the 124 to 128 range
- During Bus Idle state
- During Wait For Bus Idle state

The move-out process is not atomic. Only the CPU has priority to access the memory concurrently out of Bus Idle state. In Bus Idle, the move-out has the lowest priority to the concurrent memory accesses.

11.8.2.7 Data coherence

In order to maintain data coherency and FlexCAN proper operation, the CPU must obey the rules described in Transmit process and Receive process.

11.8.2.7.1 Transmission abort mechanism

The abort mechanism provides a safe way to request the abort of a pending transmission. A feedback mechanism is provided to inform the CPU if the transmission was aborted or if the frame could not be aborted and was transmitted instead.

Two primary conditions must be fulfilled in order to abort a transmission:

- MCR[AEN] must be asserted.
- The first CPU action must be the writing of abort code (1001b) into the CODE field of the Control and Status word.

Active MBs configured for transmission must be aborted first before they can be updated. If the abort code is written to a Mailbox that is currently being transmitted or to a Mailbox that was already loaded into the Tx SMB for transmission, the write operation is blocked and the transmission is not disturbed. However, the abort request is captured and kept pending until one of the following conditions is satisfied:

- The module loses the bus arbitration.
- There is an error during the transmission.
- The module is put into Freeze mode.
- The module enters Bus Off state.
- There is an overload frame.

If none of the conditions above are reached:

- The MB is transmitted correctly.
- The interrupt flag is set in the IFLAG register.
- An interrupt to the CPU is generated (if enabled).

The abort request is automatically cleared when the interrupt flag is set. On the other hand, if only one of the above conditions is reached, the frame is not transmitted; therefore:

- The abort code is written into the CODE field.
- The interrupt flag is set in the IFLAG.
- An interrupt is (optionally) generated to the CPU.

If the CPU writes the abort code before the transmission begins internally, then the write operation is not blocked; therefore, the MB is updated and the interrupt flag is set. In this way the CPU only needs to read the abort code to make sure the active MB was safely inactivated. Although the AEN bit is asserted and the CPU wrote the abort code, in this case the MB is inactivated and not aborted, because the transmission did not start yet. One Mailbox is aborted only when the abort request is captured and kept pending until one of the previous conditions is satisfied.

11.8.2.7.2 Mailbox inactivation

Inactivation is a mechanism provided to protect the mailbox against updates by the FlexCAN internal processes, thus allowing the CPU to rely on mailbox data coherence after having updated it, even in Normal mode.

Inactivation of transmission mailboxes must be performed just when MCR[AEN] is deasserted.

If a mailbox is inactivated, it participates in neither the arbitration process nor the matching process until it is reactivated. See Transmit process and Receive process for more detailed instructions on how to inactivate and reactivate a mailbox.

To inactivate a mailbox, the CPU must update its CODE field to INACTIVE (either 0b or 1000b).

Because you will not be able to synchronize the CODE field update with the FlexCAN internal processes, an inactivation can have the following consequences:

- A frame in the bus that matches the filtering of the inactivated Rx mailbox may be lost without notice, even if there are other mailboxes with the same filter.
- A frame containing the message within the inactivated Tx mailbox may be transmitted without setting the respective IFLAG.

In order to perform a safe inactivation and avoid the above consequences for Tx mailboxes, the CPU must use the transmission abort mechanism (see Transmission abort mechanism).

The inactivation automatically unlocks the mailbox (see Mailbox lock mechanism).

11.8.2.7.3 Mailbox lock mechanism

Other than mailbox inactivation, FlexCAN has another data coherence mechanism for the receive process. When the CPU reads the Control and Status word of an Rx MB with codes FULL or OVERRUN, FlexCAN assumes that the CPU wants to read the whole MB in an atomic operation, and therefore it sets an internal lock flag for that MB. The lock is released when the CPU reads the free running timer (global unlock operation), or when it reads the Control and Status word of another MB regardless of its code. A CPU write into the C/S word also unlocks the MB, but this procedure is not recommended for normal unlock use because it cancels a pending move and potentially may lose a received message. The MB locking prevents a new frame from being written into the MB while the CPU is reading it.

NOTE

The locking mechanism applies only to Rx MBs that have a code different than INACTIVE (0b) or EMPTY[1] (0100b). Also, Tx MBs cannot be locked.

Suppose, for example, that the second and the fifth MBs of the array are programmed with the same ID, and FlexCAN has already received and stored messages into these two MBs. Suppose now that the CPU decides to read MB number 5 and at the same time another message with the same ID is arriving. When the CPU reads the Control and

1. In previous FlexCAN versions, reading the C/S word locked the MB even if it was EMPTY. This behavior is maintained when the IRMQ bit is negated.

Status word of MB number 5, this MB is locked. The new message arrives and the matching algorithm finds out that there are no free-to-receive MBs, so it decides to override MB number 5. However, this MB is locked, so the new message cannot be written there. It will remain in the Rx SMB waiting for the MB to be unlocked, and only then will be written to the MB.

If the MB is not unlocked in time and yet another new message with the same ID arrives, then the new message overwrites the one on the Rx SMB and there will be no indication of lost messages either in the CODE field of the MB or in the Error and Status Register.

While the message is being moved in from the Rx SMB to the MB, the BUSY bit on the CODE field is asserted. If the CPU reads the Control and Status word and finds out that the BUSY bit is set, it should defer accessing the MB until the BUSY bit is negated.

Note

If the BUSY bit is asserted or if the MB is empty, then reading the Control and Status word does not lock the MB.

Inactivation takes precedence over locking. If the CPU inactivates a locked Rx MB, then its lock status is negated and the MB is marked as invalid for the current matching round. Any pending message on the Rx SMB will not be transferred anymore to the MB. An MB is unlocked when the CPU reads the Free Running Timer Register (see Free Running Timer (TIMER)), or the C/S word of another MB.

Lock and unlock mechanisms have the same functionality in both Normal and Freeze modes.

An unlock during Normal or Freeze mode results in the move-in of the pending message. However, the move-in is postponed if an unlock occurs during a low power mode (see Modes of operation), and it takes place only when the module returns to Normal or Freeze modes.

11.8.2.8 Rx FIFO

The Rx FIFO is receive-only and is enabled by asserting MCR[RFEN]. The reset value of this bit is zero to maintain software backward compatibility with previous versions of the module that did not have the FIFO feature.

CAUTION

Rx FIFO must not be enabled when CAN FD feature is enabled.

The FIFO is 6-message deep. The memory region occupied by the FIFO structure (both message buffers and FIFO engine) is described in Rx FIFO structure. The CPU can read the received messages sequentially, in the order they were received, by repeatedly reading a message buffer structure at the output of the FIFO.

IFLAG1[BUF5I] (Frames available in Rx FIFO) is asserted when there is at least one frame available to be read from the FIFO. An interrupt is generated if it is enabled by the corresponding mask bit. Upon receiving the interrupt, the CPU can read the message (accessing the output of the FIFO as a message buffer) and the RXFIR register, and then clear the interrupt. If there are more messages in the FIFO the act of clearing the interrupt updates the output of the FIFO with the next message and updates RXFIR with the attributes of that message, reissuing the interrupt to the CPU. Otherwise, the flag remains negated. The output of the FIFO is valid only while IFLAG1[BUF5I] is asserted.

IFLAG1[BUF6I] (Rx FIFO Warning) is asserted when the number of unread messages within the Rx FIFO is increased to five from four due to the reception of a new one, meaning that the Rx FIFO is almost full. The flag remains asserted until the CPU clears it.

IFLAG1[BUF7I] (Rx FIFO Overflow) is asserted when an incoming message was lost because the Rx FIFO is full. Note that the flag will not be asserted when the Rx FIFO is full and the message was captured by a Mailbox. The flag remains asserted until the CPU clears it.

Clearing one of those three flags does not affect the state of the other two.

An interrupt is generated if an IFLAG bit is asserted and the corresponding mask bit is asserted too.

A powerful filtering scheme is provided to accept only frames intended for the target application, reducing the interrupt servicing workload. The filtering criteria is specified by programming a table of up to 128 32-bit registers, according to CTRL2[RFFN] setting, that can be configured to one of the following formats (see also Rx FIFO structure):

- Format A: 128 IDAFs (extended or standard IDs including IDE and RTR)
- Format B: 256 IDAFs (standard IDs or extended 14-bit ID slices including IDE and RTR)
- Format C: 512 IDAFs (standard or extended 8-bit ID slices)

Note

A chosen format is applied to all entries of the filter table. It is not possible to mix formats within the table.

Every frame available in the FIFO has a corresponding IDHIT (Identifier Acceptance Filter Hit Indicator) that can be read in the IDHIT field from C/S word, as shown in the Rx FIFO Structure description. Another way the CPU can obtain this information is by accessing the RXFIR register. RXFIR[IDHIT] refers to the message at the output of the FIFO and is valid while the IFLAG1[BUF5I] flag is asserted. The RXFIR register must be read only before clearing the flag, which guarantees that the information refers to the correct frame within the FIFO.

Up to 32 elements of the filter table are individually affected by the Individual Mask Registers (RXIMRx), according to the setting of CTRL2[RFFN], allowing very powerful filtering criteria to be defined. If MCR[IRMQ] is negated, then the FIFO filter table is affected by RXFGMASK.

NOTE

For more information about the difference between FD and non-FD regarding this feature, see Table 11-159.

11.8.2.8.1 Rx FIFO under DMA operation

The receive-only FIFO can support DMA. This feature is enabled by asserting both MCR[RFEN] and MCR[DMA]. The reset value of MCR[DMA] is zero to maintain backward compatibility with previous versions of the module that did not have the DMA feature.

The DMA controller can read the received message by reading a message buffer structure at the FIFO output port at the 80h–8Ch address range.

NOTE

FlexCAN supports 32-bit access only for DMA transfers.

When MCR[DMA] is asserted the CPU must not access the FIFO output port address range. Before enabling MCR[DMA], the CPU must service the IFLAGs asserted in the Rx FIFO region. Otherwise, these IFLAGs may show that the FIFO has data to be serviced, and mistakenly generate a DMA request. Before disabling MCR[DMA], the CPU must perform a clear FIFO operation.

IFLAG1[BUF5I] (Frames available in Rx FIFO) is asserted when there is at least one frame available to be read from the FIFO. Consequently a DMA request is generated simultaneously. Upon receiving the request, the DMA controller can read the message (accessing the output of the FIFO as a message buffer). The DMA reading process must end by reading address 8Ch, which clears IFLAG1[BUF5I] and updates both the FIFO output with the next message (if FIFO is not empty) and the RXFIR register with the

attributes of the new message. If there are more messages stored in the FIFO, IFLAG1[BUF5I] will be re-asserted and another DMA request is issued. Otherwise, the flag remains negated.

IFLAG1[BUF6I] and IFLAG1[BUF7I] are not used when the DMA feature is enabled.

When FlexCAN is working with DMA, the CPU does not receive any Rx FIFO interruption and must not clear the related IFLAGs. In addition, the related IMASKs are not used to mask the generation of DMA requests.

NOTE

For more information about the difference between FD and non-FD regarding this feature, see Table 11-159.

11.8.2.8.2 Clear FIFO operation

When MCR[RFEN] is asserted, the clear FIFO operation is a feature used to empty FIFO contents. With MCR[RFEN] asserted the Clear FIFO occurs when the CPU writes one in IFLAG1[BUF0I]. This operation can only be performed in Freeze mode and is blocked by hardware in other modes. This operation does not clear the FIFO IFLAGs; consequently the CPU must service all FIFO IFLAGs before executing the clear FIFO task.

When Rx FIFO is working with DMA, the clear FIFO operation clears IFLAG1[BUF5I] and the DMA request is canceled.

CAUTION

Clear FIFO operation does not clear IFLAGs, except when MCR[DMA] is asserted; in this case only IFLAG1[BUF5I] is cleared.

11.8.2.9 CAN protocol related features

This section describes the CAN protocol related features.

11.8.2.9.1 CAN FD ISO compliance

The CAN FD protocol has been improved to increase the failure detection capability that was in the original CAN FD protocol, which is also called non-ISO CAN FD, by CAN in Automation (CiA). A three-bit stuff counter and a parity bit have been introduced in the improved CAN FD protocol, now called ISO CAN FD. The CRC calculation has also been modified. All these improvements make the ISO CAN FD protocol incompatible

with the non-FD CAN FD protocol. The non-ISO CAN FD is still supported by FlexCAN so that it can be used mainly during an intermediate phase, for evaluation and development purposes.

Therefore, it is strongly recommended to configure FlexCAN to the ISO CAN FD protocol by setting the ISOCANFDEN field in the CTRL2 register.

11.8.2.9.2 CAN FD frames

The ISO 11898-1 standard specifies the Classical Frame format compliant to ISO 11898-1 (2003) and introduces the CAN Flexible Data Rate Frame format. The Classical Frame format allows bit rates up to 1 Mbit/s and payloads up to 8 bytes per frame. The Flexible Data Rate Frame format allows bit rates higher than 1 Mbit/s and payloads longer than 8 bytes per frame. FlexCAN can receive and transmit CAN FD messages interleaved with Classical CAN messages.

There are three additional control bits in the CAN FD frame. The Extended Data Length (EDL) bit enables a longer data payload with different data length coding. The Bit Rate Switch (BRS) bit decides whether the bit rate is switched inside a CAN FD format frame. The Error State Indicator (ESI) flag is transmitted dominant by error active nodes, and recessive by error passive nodes. There are no Remote Frames (see Remote frames) in the CAN FD format. A message configured to transmit a Remote Frame is always sent out in the Classical CAN format. When an FD frame is received and matches a mailbox, the RTR bit in the receiving message buffer is negated. The RTR bit must be considered in classical frames only.

CAN FD messages may be formatted as long frames where the data field exceeds 8 bytes, and may range from 12 up to 64 bytes. They can also be configured to support bit rate switching, where the control field, the data field, and the CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame. Messages in Classical CAN format are limited to transport a maximum payload of 8 bytes at nominal rate. The following figure illustrates the message formats for Classical and FD frames with either standard or extended ID.

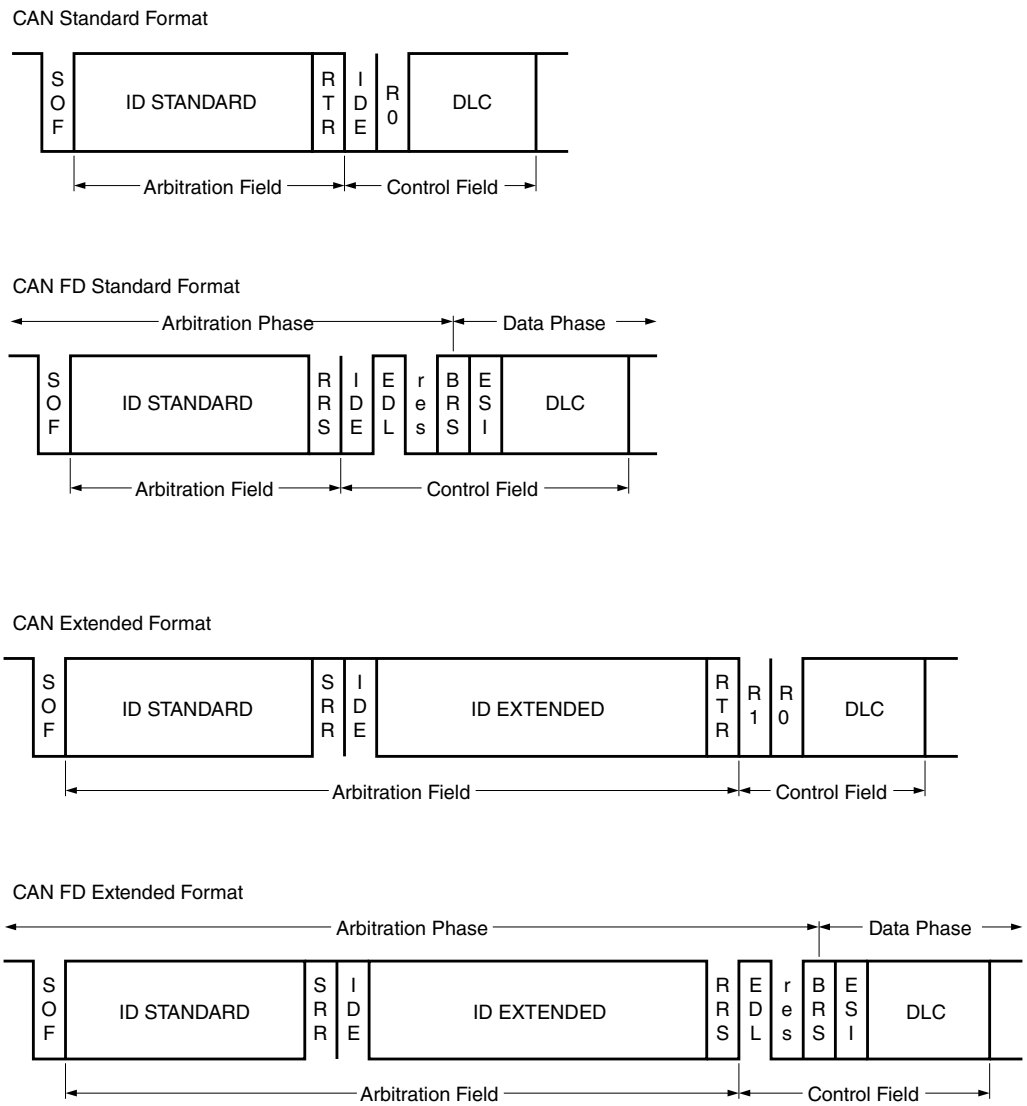


Figure 11-117. CAN message formats

The ability to receive and transmit CAN FD messages is enabled by MCR[FDEN]. Either a recessive R0 bit in CAN frames with 11-bit identifiers or a recessive R1 bit in CAN frames with 29-bit identifiers are decoded as an EDL bit (not a reserved one). A CAN FD frame is recognized by a recessive EDL bit, and a Classical CAN frame is recognized by a dominant EDL bit. The BRS bit specifies whether this frame switches the bit rate in its data phase. A long frame is decoded in accordance with the DLC field value (see DLC definition in Message buffer structure).

CAN FD messages can be transmitted with two different bit rates. The first part of a CAN FD frame, from the Start Of Frame (SOF) bit until the Bit Rate Switch (BRS) bit, also called the arbitration phase, is transmitted with the nominal bit rate based on a set of nominal CAN bit timing configuration values. The second part, from the BRS bit until the CRC Delimiter bit, also named the data phase, is transmitted with the data bit rate

defined by a second set of CAN data bit timing configuration values. Finally, from the CRC Delimiter until the Intermission bits, the transmission returns to nominal bit rate. In CAN FD frames with bit rate switching, the bit timing is changed inside the frame at the sample point of the BRS bit if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the CBT register (also by CTRL1 register for backward compatibility). Upon detecting a recessive BRS bit, the CAN data bit timing is used as defined by the FDCBT register.

NOTE

If the length of the time quantum in the nominal bit timing and the length of the time quantum in the data bit timing are not identical, a quantization error of up to one time quantum of the arbitration phase may be present as a phase error. This situation can occur after the switch from arbitration to data phase and will last until the next synchronization event. Thus, the length of the time quantum should be the same in nominal and data bit timing in order to minimize the chance of error frames on the CAN bus, and to optimize the clock tolerance in networks that use FD frames.

FDCTRL[FDRATE] enables the transmission of all frames with bit rate switching if the BRS bit in the selected Tx MB is set. If FDRATE is negated, the transmission is performed at nominal rate regardless of the BRS bit value. FDCTRL[FDRATE] can be written any time but takes effect only for the next message transmitted or received.

The nominal bit timing is resumed at either the sample point of the CRC Delimiter bit or when an error is detected, whichever occurs first. The following figure describes the mechanism for entering and leaving the data phase when BRS bit is recessive.

CAN FD Frame

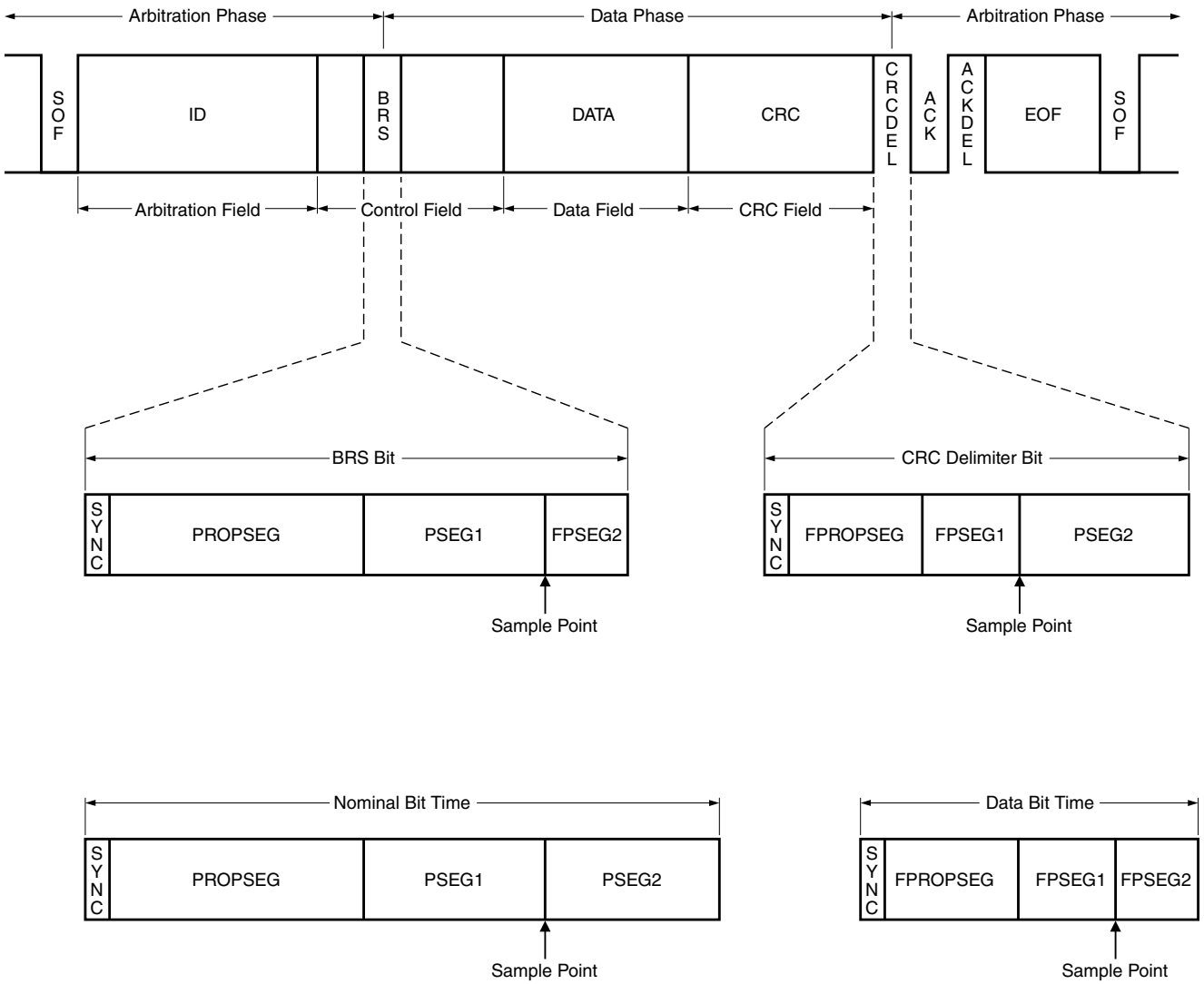


Figure 11-118. Bit rate switching mechanism for CAN FD messages

NOTE

In Classical CAN frames, the CRC delimiter is one single recessive bit. In CAN FD frames, the CRC delimiter may consist of one or two recessive bits. FlexCAN sends only one recessive bit as the CRC delimiter, but it accepts two recessive bits before the edge from recessive to dominant that starts the acknowledge slot. As a receiver, FlexCAN sends its acknowledge bit after the first CRC delimiter bit. In CAN FD frames, FlexCAN accepts a two-bit dominant ACK slot as a valid ACK to compensate for phase shifts between the receivers.

The maximum configurable bit rate in the CAN FD data phase depends on the clock frequency of CAN_PE subblock. For example, with a CAN_PE clock frequency of 40 MHz and the shortest configurable bit time of 5 time quanta, the bit rate in the data phase is 8 Mbit/s.[4]

The value of the ESI bit is determined either by the transmitter's error state at the start of the transmission, if the frame is originated in the FlexCAN node, or by the original transmitting node in case FlexCAN is acting as a gateway for the message. If the transmitter is error passive, ESI is transmitted recessive; otherwise, it is transmitted dominant. The permutations of the relationship between the written value and the transmitted value of the ESI are shown in this table.

Table 11-165. Written versus transmitted values of ESI field

FlexCAN fault confinement status at start of frame	ESI bit Of Tx MB	Transmitted ESI
Error active	0	0 (Error Active)
Error passive	0	1 (Error Passive)
Error active	1	1 (Error Passive)
Error passive	1	1 (Error Passive)

There are different CRC polynomials for different CAN frame formats. The first polynomial, CRC_15, is used for all frames in Classical CAN format. The second, CRC_17, is used for frames in CAN FD format with a data field up to sixteen bytes long. The third, CRC_21, is used for frames in CAN FD format with a data field longer than sixteen bytes. Each polynomial results in a Hamming distance of 6. At the start of the frame, all three CRC polynomials are calculated concurrently. The CRC sequence to be transmitted is selected by the values of the EDL bit and the DLC bit field. When receiving a message, FlexCAN decodes EDL and DLC to select the adequate CRC polynomial to check for a CRC error.

In CAN FD format frames, stuff bits are included in the bit stream for CRC calculation. In Classical CAN format frames, stuff bits are not included. After the transmission of the last bit relevant to the CRC calculation, the FDCRC register stores the calculated CRC for the transmitted message, with the adequate length in accordance to the type of message, for both CAN FD and non-FD messages. The CRCR register reports a valid CRC for Classical CAN messages only.

In CAN FD format frames, the CAN bit stuffing method is changed for the CRC sequence so that the stuff bits are inserted at fixed positions. When FlexCAN is transmitting a CAN FD frame, a fixed stuff bit is inserted just before the first bit of the CRC sequence, even if the last bits of the preceding field do not fulfill the CAN stuff

4. The frequency used in this example may not be supported on this chip; it is shown only to demonstrate how the maximum configurable bit rate is calculated.

condition. Additional stuff bits are inserted after each fourth bit of the CRC sequence. The value of any fixed stuff bit is the inverse value of its preceding bit. When FlexCAN is receiving a CAN FD frame, it discards the fixed stuff bits from the bit stream for the CRC check. A stuff error is detected if the fixed stuff bit has the same value as its preceding bit.

FlexCAN detects errors in CAN FD frames the same way as in Classical CAN frames. The error counters RXERRCNT and TXERRCNT in the ECR register accumulate the counts of Rx and Tx errors, respectively, for both FD and non-FD frames indiscriminately. There are two extra error counters (RXERRCNT_FAST and TXERRCNT_FAST) that accumulate Rx and Tx errors occurring in the data phase of CAN FD frames with the BRS bit set only. The rules for updating the error counters are the same for both CAN FD and non-FD frames (see ECR register).

Error Flags BITERR1, BITERR0, ACKERR, CRCERR, FRMERR, and STFERR in the ESR1 register report errors in both CAN FD and non-FD frames. They also generate the ERRINT interrupt if CTRL1[ERRMSK] is asserted. The ESR1 register has additional error flags (BITERR1_FAST, BITERR0_FAST, CRCERR_FAST, FRMERR_FAST, and STFERR_FAST) to individually indicate the occurrence of errors in the data phase of CAN FD frames with the BRS bit set. There is no ACKERR detected in the data phase of a CAN FD frame. Fault confinement status reported in ESR1[FLTCONF] is the same for both CAN FD and Classical CAN frames, and is based on RXERRCNT and TXERRCNT error counters only. Information contained in RXERRCNT_FAST and TXERRCNT_FAST counters may be considered as status to help detect the error nature related to the bit rate value.

When FlexCAN is in the data phase, either transmitting or receiving a CAN FD message, and detects an error, it immediately switches back to the arbitration phase and to the nominal rate to start an error flag.

Resynchronization and hard synchronization occur in CAN FD frames in the same way as in Classical CAN ones. Additionally, a hard synchronization is also performed at the recessive to dominant edge from EDL to R0 in CAN FD format frames. FlexCAN does not resynchronize while transmitting in the CAN FD data phase.

11.8.2.9.3 Transceiver delay compensation

The CAN FD protocol allows the transmission and reception of data at a higher bit rate than the nominal rate used in the arbitration phase when the message's BRS bit is set. This feature enables the use of rates up to 8 Mbps.

During the data phase of a CAN FD frame, the transmitter detects a bit error if it cannot receive its own latest transmitted bit at the sample point of that bit. When bit rate switching is enabled (BRS bit is asserted), the length of the CAN bit time in the data

phase can become shorter than the transceiver's loop delay, thus impeding the correct comparison between the transmitted bit and the received bit within the current CAN bit time interval.

Note that the TDC process defines a secondary sample point where the transmitted bit is correctly compared with the received bit in order to check for bit errors.

The TDC mechanism can be enabled by FDCTRL[TDCEN] and is effective only during the data phase of FD frames having the BRS bit set. It has no effect either on non-FD frames, or on FD frames transmitted at normal bit rate. The TDC is active from the sample point of the BRS bit until the sample point of the CRC Delimiter bit, provided the respective message under transmission has the BRS bit set. When it is active, a comparison is done between the real received bit and the delayed transmitted bit, where the delay is calculated based on the measured transceiver loop delay.

NOTE

The actual value of the CRC Delimiter bit is disregarded by transmitters using the transceiver delay compensation mechanism. A global error at the end of the CRC field will cause the receivers to send error frames that the transmitter will detect during Acknowledge or End of Frame.

For every transmitted FD frame having the BRS bit set, the delay measurement is triggered by the transition from the recessive EDL bit to the dominant R0 bit (as shown in the next figure). The loop delay is measured in Protocol Engine (PE) clock periods (CANCLK, see Protocol timing), from the transmitted EDL-R0 edge to the received EDL-R0 edge. The position of the secondary sample point is defined by the measured loop delay time added to an offset value specified in FDCTRL[TDCOFF]. FDCTRL[TDCVAL] stores the result of this calculation. The TDCVAL value saturates at its maximum value of 63 CANCLK when the delay measurement is too long.

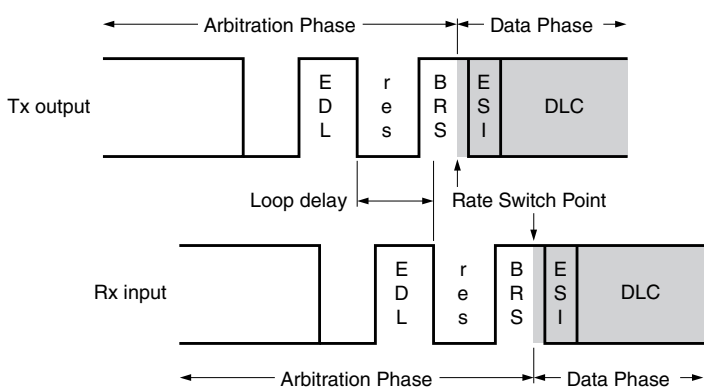


Figure 11-119. Transceiver loop delay measurement

The measured loop delay is not enough to be used to define the secondary sample point because it relates to the CAN bit edges. The transceiver delay compensation offset TDCOFF is used to shift the secondary sample point from the edge to an intermediate point inside the bit time, far away from its edges. Therefore, the TDCOFF value cannot be larger than the CAN bit duration in the data phase.

If the secondary sample point is set very near the CAN bit edge (SYNC field), then problems may occur during the bit sampling in the data phase. For the TDC to work reliably, the offset has to use optimal settings. To be sure the bit sampling is performed in the best region, the TDC offset should be configured as shown in this equation:

$$Offset = (FPSEG1 + FPROPSEG + 2) \times (FPRESDIV + 1)$$

The following figure shows the SSP position when these settings are used.

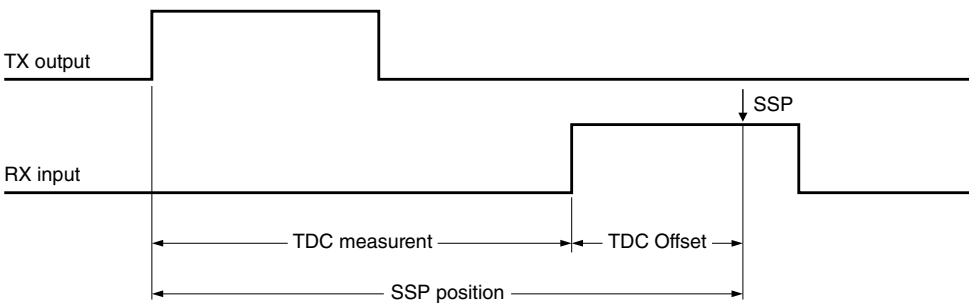


Figure 11-120. SSP position with optimal values

During the data phase of CAN FD frames with bit rate switching enabled, at the onset of every Tx CAN bit, the transmitted Tx bit value is temporarily stored in a buffer and a time countdown based on FDCTRL[TDCVAL] is started which ends with the comparison of the received Rx bit (delayed by the external loop delay plus the specified offset) with the stored Tx bit. If a bit error is detected at the secondary sample point, the FlexCAN issues an error flag to the CAN bus at the next sample point.

During the arbitration phase the delay compensation is always disabled. The maximum delay which can be compensated by the FlexCAN's transceiver delay compensation during the data phase is 3 CAN bit times – 2 Tq. Beyond this limit, the FDCTRL[TDCFAIL] flag is set to indicate when the transceiver delay compensation mechanism is out of range, unable to compensate the transceiver loop delay.

11.8.2.9.4 Remote frames

A remote frame is a special kind of frame. You can program a mailbox to be a remote request frame by configuring the mailbox as Transmit with the RTR bit set to one. After the remote request frame is transmitted successfully, the mailbox becomes a receive message buffer, with the same ID as before.

When a remote request frame is received by FlexCAN, it can be treated in three ways, depending on remote request storing (CTRL2[RRS]) and Rx FIFO Enable (MCR[RFEN]):

- If RRS is negated the frame's ID is compared to the IDs of the transmit message buffers with the CODE field 1010b. If there is a matching ID, then this mailbox frame will be transmitted. Note that if the matching mailbox has the RTR bit set, then FlexCAN will transmit a remote frame as a response. The received remote request frame is not stored in a receive buffer. It is only used to trigger a transmission of a frame in response. The mask registers are not used in remote frame matching, and all ID bits (except RTR) of the incoming received frame should match. In the case that a remote request frame is received and matches a mailbox, this message buffer immediately enters the internal arbitration process, but is considered as a normal Tx mailbox, with no higher priority. The data length of this frame is independent of the DLC field in the remote frame that initiated its transmission.
- If RRS is asserted the frame's ID is compared to the IDs of the receive mailboxes with the CODE field 0100b, 0010b, or 0110b. If there is a matching ID, then this mailbox will store the remote frame in the same fashion of a data frame. No automatic remote response frame will be generated. The mask registers are used in the matching process.
- If RFEN is asserted FlexCAN will not generate an automatic response for remote request frames that match the FIFO filtering criteria. If the remote frame matches one of the target IDs, it will be stored in the FIFO and presented to the CPU. Note that for filtering formats A and B, it is possible to select whether remote frames are accepted or not. For format C, remote frames are always accepted (if they match the ID). Remote request frames are considered as normal frames, and generate a FIFO overflow when a successful reception occurs and the FIFO is already full.

NOTE

There is no remote frame in the CAN FD format. The RTR bit is replaced by a fixed dominant RRS bit. FlexCAN receives and transmits remote frames in the Classical CAN format.

11.8.2.9.5 Overload frames

FlexCAN does transmit overload frames when the following conditions are detected on the CAN bus:

- Detection of a dominant bit in the first/second bit of Intermission
- Detection of a dominant bit at the 7th bit (last) of End of Frame field (Rx frames)
- Detection of a dominant bit at the 8th bit (last) of Error Frame Delimiter or Overload Frame Delimiter

11.8.2.9.6 Time stamp

The value of the free running timer is sampled at the beginning of the Identifier field on the CAN bus, and is stored at the end of move-in in the TIME STAMP field, providing network behavior with respect to time.

The free running timer is clocked by the FlexCAN bit-clock, which defines the baud rate on the CAN bus. During a message transmission/reception, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it counts using the previously programmed baud rate.

The free running timer is not incremented during Disable, Doze, Stop, and Freeze modes. It can be reset upon a specific frame reception, enabling network time synchronization. See the TSYN description in Control 1 register (CTRL1).

11.8.2.9.7 Protocol timing

The following figure shows the structure of the clock generation circuitry that feeds the CAN Protocol Engine (PE) submodule. The clock source field CTRL1[CLKSRC] defines whether the internal clock is connected to the output of a crystal oscillator (oscillator clock) or to the peripheral clock. In order to guarantee reliable operation, the clock source should be selected when the module is in Disable mode (MCR[MDIS] is set).

NOTE

Please see the clock distribution chapter (module clocks table) to identify the proper clock source.

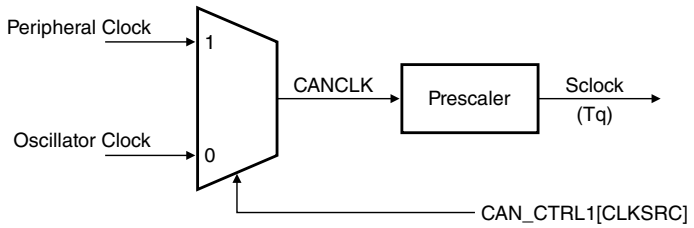


Figure 11-121. CAN engine clocking scheme

The oscillator clock should be selected whenever a tight tolerance (up to 0.1%) is required in the CAN bus timing. The crystal oscillator clock has better jitter performance than the peripheral clock.

The FlexCAN module supports a variety of means to set up bit timing parameters that are required by the CAN protocol. The Control 1 register (CTRL1) has various fields used to control bit timing parameters: PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW.

The CAN Bit Timing register (CBT) extends the range of the CAN bit timing variables in CTRL1. The CAN FD Bit Timing register (FDCBT) provides a second set of CAN bit timing variables to be applied at the data phase of CAN FD frames with the Bit Rate Switch (BRS) set.

NOTE

When the CAN FD feature is enabled, always set CBT[BTF] and configure the CAN bit timing variables in CBT. See CAN Bit Timing Register (CBT).

The PRESDIV field (as well as its extended range EPRESDIV and FDPRESDIV for the data phase bits of CAN FD messages) defines the prescaler value (see the equation below) that generates the serial clock (Sclock), whose period defines the time quantum used to compose the CAN waveform. A time quantum (Tq) is the atomic unit of time managed by the CAN engine.

$$Tq = \frac{(PRESDIV + 1)}{f_{CANCLK}}$$

Figure 11-122. Time quantum

The bit rate, which defines the rate the CAN message is either received or transmitted, is given by the formula:

$$\text{CAN Bit Time} = (\text{Number of Time Quanta in 1 bit time}) * Tq$$

$$\text{Bit Rate} = \frac{1}{\text{CAN Bit Time}}$$

Figure 11-123. CAN bit time and baud rate

A bit time is subdivided into three segments[1] (see Figure 11-124, Figure 11-125 and Table 11-166):

- SYNC_SEG: This segment has a fixed length of one time quantum. Signal edges are expected to happen within this section.
- Time Segment 1: This segment includes the propagation segment and the phase segment 1 of the CAN standard. It can be programmed by setting CTRL1[PROPSEG] and CTRL1[PSEG1] so that their sum (plus 2) is in the range of 2 to 16 time quanta. When CBT[BTF] is asserted, FlexCAN uses EPROPSEG and EPSEG1 fields from CBT register so that their sum (plus 2) is in the range of 2 to 96 time quanta. For messages in CAN FD format with the BRS bit set, FlexCAN uses FDPROPSEG and FDPSEG1 from FDCBT instead, so that their sum (plus 1) is in the range of 2 to 39 time quanta.
- Time Segment 2: This segment represents the phase segment 2 of the CAN standard. It can be programmed by setting CTRL1[PSEG2] (plus 1) to be 2 to 8 time quanta long. When CBT[BTF] is asserted, FlexCAN uses EPSEG2 fields of CBT register so that its value (plus 1) is in the range of 2 to 32 time quanta. For messages in CAN FD format with the BRS bit set, FlexCAN uses FDPSEG2 from FDCBT instead, so that its value (plus 1) is in the range of 2 to 8 time quanta. The time segment 2 cannot be smaller than the Information Processing Time (IPT), which value is 2 time quanta in FlexCAN.

NOTE

The bit time defined by the above time segments must not be smaller than five time quanta. For bit time calculations, use an Information Processing Time (IPT) of two, which is the value implemented in the FlexCAN module.

1. For further explanation of the underlying concepts, see ISO 11898-1. See also the CAN 2.0A/B protocol specification for bit timing.

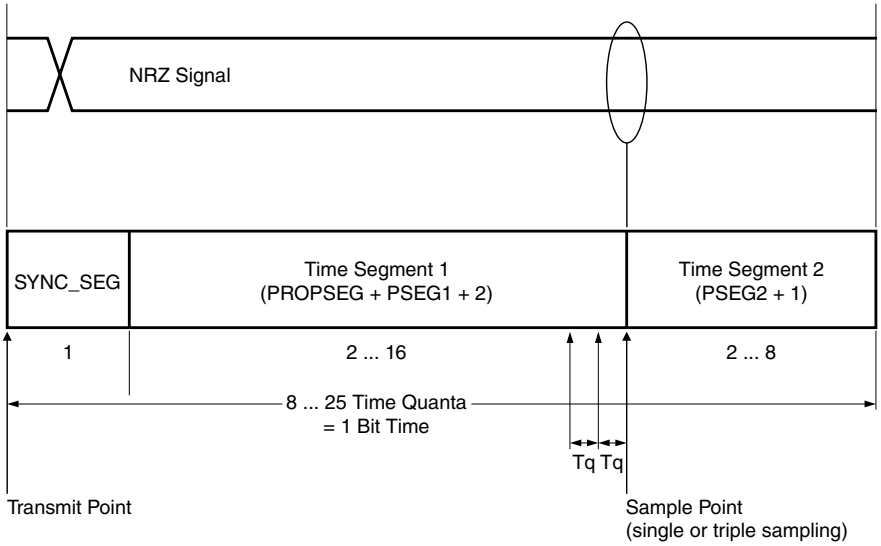


Figure 11-124. Segments within the bit time (example using CTRL1 bit timing variables for Classical CAN format)

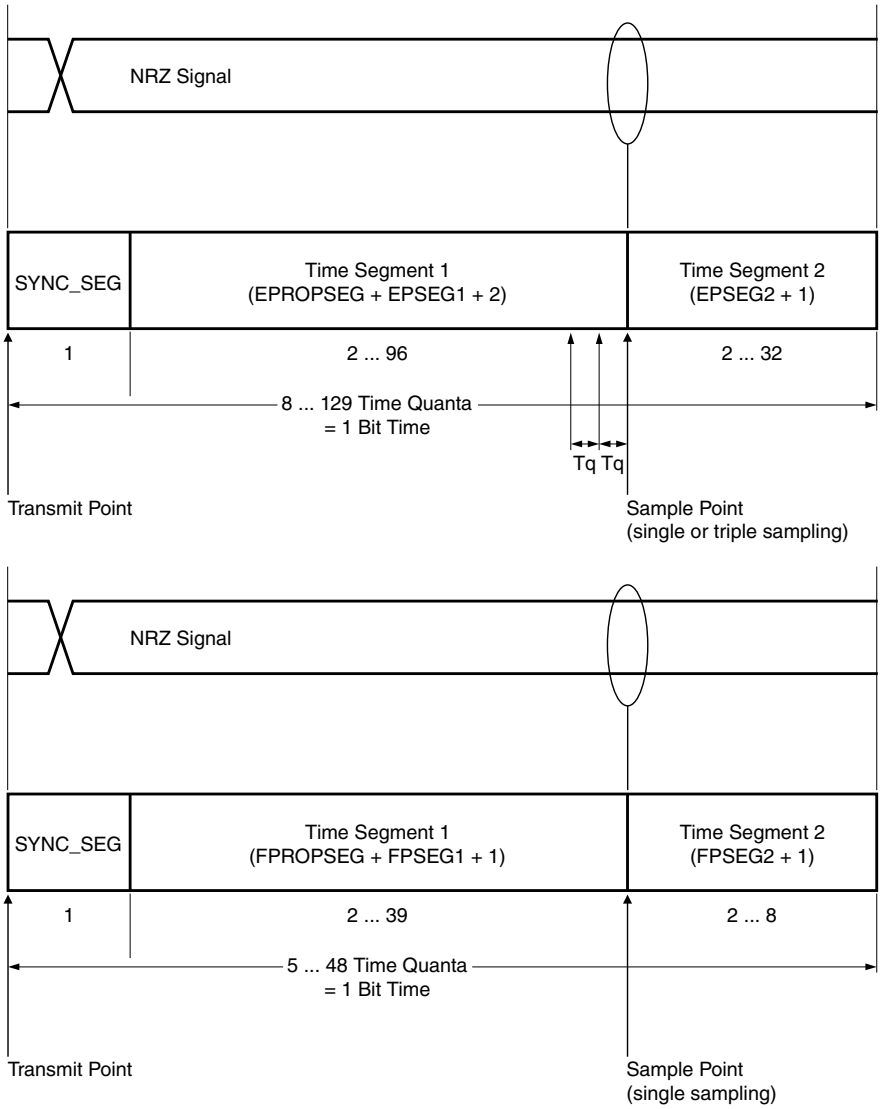


Figure 11-125. Segments within the bit time (example using CBT and FDCBT bit timing variables for CAN FD format)

Table 11-166. Time segment syntax

Syntax	Description
SYNC_SEG	System expects transitions to occur on the bus during this period.
TSEG1	Corresponds to the sum of PROPSEG and PSEG1.
TSEG2	Corresponds to the PSEG2 value.
Transmit point	A node in transmit mode transfers a new value to the CAN bus at this point.
Sample point	A node samples the bus at this point. If the three samples per bit option is selected, then this point marks the position of the third sample.

The following table gives some examples of the CAN compliant segment settings for Classical CAN format (Bosch CAN 2.0B) (non-FD) messages.

Table 11-167. Bosch CAN 2.0B standard compliant bit time segment settings

Time segment 1	Time segment 2	Re-synchronization jump width
5 .. 10	2	1 .. 2
4 .. 11	3	1 .. 3
5 .. 12	4	1 .. 4
6 .. 13	5	1 .. 4
7 .. 14	6	1 .. 4
8 .. 15	7	1 .. 4
9 .. 16	8	1 .. 4

Note

The user must ensure the bit time settings are in compliance with the CAN Protocol standard (ISO 11898-1).

Whenever CAN bit is used as a measure of time duration (for example, estimating the occurrence of a CAN bit event in a message), the number of peripheral clocks in one CAN bit (NumClkBit) can be calculated as:

$$\text{NumClkBit} = \frac{f_{SYS}}{f_{CANCLK}} \times (\text{PRESDIV} + 1) \times (\text{PROPSEG} + \text{PSEG1} + \text{PSEG2} + 4)$$

Figure 11-126. Number of peripheral clocks per CAN bit when CTRL2[BTE] = 0

where:

- NumClkBit is the number of peripheral clocks in one CAN bit.
- f_{CANCLK} is the Protocol Engine (PE) Clock (see Figure 11-121), in Hz.
- f_{SYS} is the frequency of operation of the system (CHI) clock, in Hz.
- PSEG1 is the value in CTRL1[PSEG1] field.
- PSEG2 is the value in CTRL1[PSEG2] field.
- PROPSEG is the value in CTRL1[PROPSEG] field.
- PRESDIV is the value in CTRL1[PRESDIV] field.

The formula above is also applicable to the alternative CAN bit timing variables described in the CAN Bit Timing register (CBT) and also to the CAN FD Bit Timing register (FDCBT).

For example, 180 CAN bits = (180 x NumClkBit) peripheral clock periods.

11.8.2.9.8 Arbitration and matching timing

During normal reception and transmission, the matching, arbitration, move-in, and move-out processes are executed during certain time windows inside the CAN frame, as shown in the following figures.

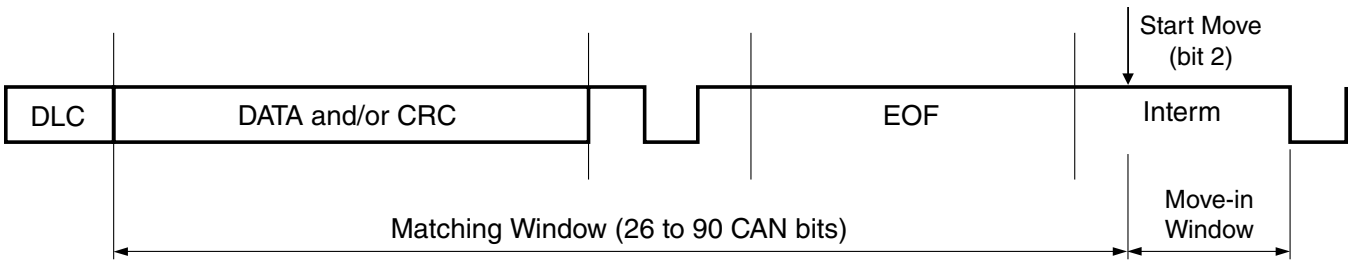


Figure 11-127. Matching and move-in time windows

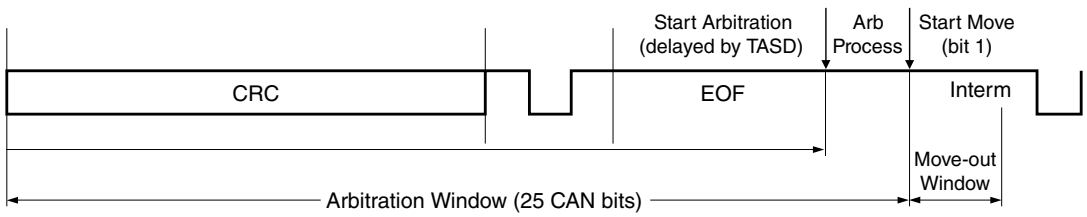


Figure 11-128. Arbitration and move-out time windows

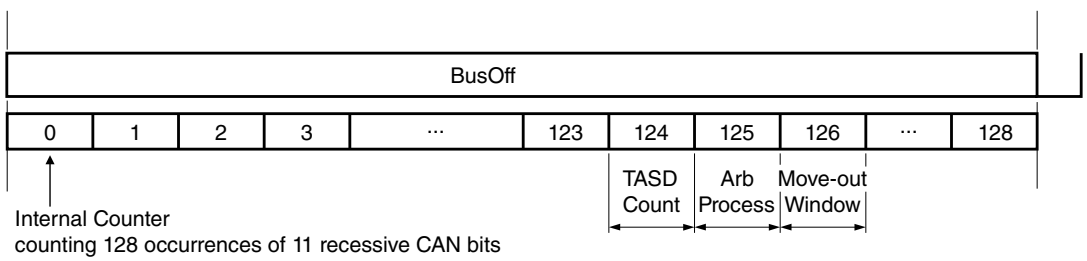


Figure 11-129. Arbitration at the end of bus off and move-out time windows

NOTE

In the preceding figures, the matching and arbitration timing does not take into account the delay caused by the concurrent memory access due to the CPU or other internal FlexCAN subblocks.

11.8.2.9.9 Tx arbitration start delay

CTRL2[TASD] (Tx Arbitration Start Delay) is a variable that indicates the number of CAN bits used by FlexCAN to delay the Tx arbitration process start point from the first bit of CRC field of the current frame. This variable can be written only in Freeze mode because it is blocked by hardware in other modes.

The transmission performance is impacted by the ability of the CPU to reconfigure message buffers (MBs) for transmission after the end of the internal arbitration process, where FlexCAN finds the winner MB for transmission (see Arbitration process). If the arbitration ends too early before the first bit of Intermission field, then there is a chance that the CPU reconfigures some Tx MBs and the winner MB is no longer the best candidate to be transmitted.

TASD is useful to optimize the transmission performance by defining the arbitration start point, as shown in the next figure, based on factors such as:

- Peripheral-to-oscillator clock ratio
- CAN bit timing variables that determine the CAN bit rate
- Number of message buffers (MBs) in use by the matching and arbitration processes

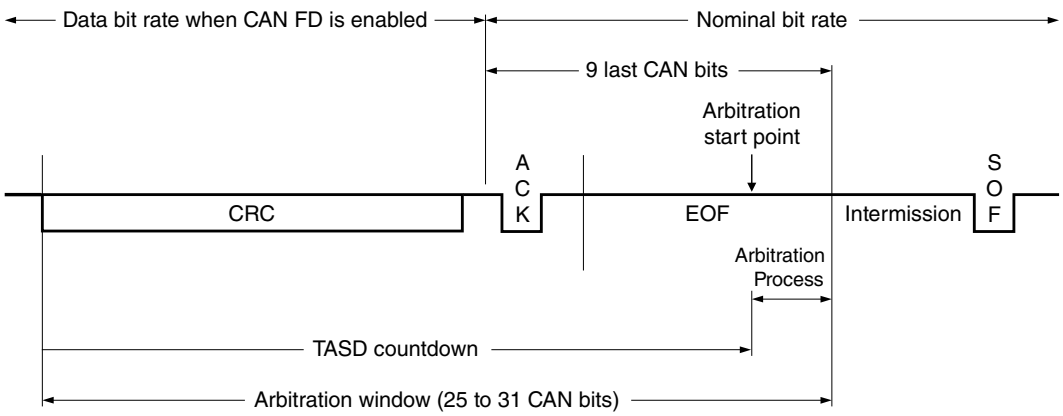


Figure 11-130. Optimal Tx Arbitration start point

The duration of an arbitration process, in terms of CAN bits, is directly proportional to the number of available MBs and to the CAN bit rate, and inversely proportional to the peripheral clock frequency.

The optimal arbitration timing is that in which the last MB is scanned just before the first bit of the Intermission field of a CAN frame. For instance, if there are few MBs and the peripheral/oscillator clock ratio is high and the CAN baud rate is low, then the arbitration can be placed closer to the frame's end, adding more delay to its start point, and vice-versa.

If TASD is set to 0 then the arbitration start is not delayed and more time is reserved for arbitration. On the other hand, if TASD is close to 24 then the CPU can configure a Tx MB later and less time is reserved for arbitration. If too little time is reserved for arbitration the FlexCAN may be not be able to find a winner MB in time to be transmitted with the best chance to win the bus arbitration against external nodes on the CAN bus.

The optimal TASD value can be calculated as follows:

For CAN FD frames and (MAXMB + 1) ≤ NMB_{END}

$$TASD = 31 - \frac{2 * (MAXMB + 1) + 4}{CPCB_N}$$

For CAN FD frames and (MAXMB + 1) > NMB_{END}

$$TASD = 22 - \frac{2 * (MAXMB + 1) - NMB_{END}}{CPCB_F}$$

For non-FD frames

$$TASD = 25 - \frac{2 * (MAXMB + 1) + 4}{CPCB}$$

Figure 11-131. Optimal value for TASD

where:

$$NMB_{END} = \frac{(9 * CPCB_N) - 4}{2}$$

$$BITRATE_N = \left(\frac{f_{CANCLK}}{[1 + (EPSEG1 + 1) + (EPSEG2 + 1) + (EPROPSEG + 1)] \times (EPRESDIV + 1)} \right)$$

$$BITRATE_F = \left(\frac{f_{CANCLK}}{[1 + (FPSEG1 + 1) + (FPSEG2 + 1) + FPROPSEG] \times (FPRESDIV + 1)} \right)$$

$$CPCB_N = \frac{f_{SYS}}{BITRATE_N}$$

$$CPCB_F = \frac{f_{SYS}}{BITRATE_F}$$

$$CPCB = CPCB_N$$

Figure 11-132. Variables used in TASD calculation

- MAXMB is the value in CTRL1[MAXMB].
- NMB_{END} is the number of message buffers that can be scanned by the arbitration process during the 9 last CAN bits at the end of a frame (see the figure above).
- $BITRATE_N$ is the CAN bit rate in bits per second calculated by the nominal CAN bit time variables.
- $BITRATE_F$ is the CAN bit rate in bits per second calculated by the data CAN bit time variables.
- $CPCB_N$ is the number of peripheral clocks per CAN bit in nominal bit rate for CAN FD frames.
- $CPCB_F$ is the number of peripheral clocks per CAN bit in data bit rate for CAN FD frames.
- CPCB is the number of peripheral clocks per CAN bit for non-FD frames.
- f_{CANCLK} is the oscillator clock, in Hz.
- f_{SYS} is the peripheral clock, in Hz.
- EPSEG1 is the value in CBT[EPSEG1] (CTRL1[PSEG1] can also be used).

- EPSEG2 is the value in CBT[EPSEG2] (CTRL1[PSEG2] can also be used).
- EPROPSEG is the value in CBT[EPROPSEG] (CTRL1[PROPSEG] can also be used).
- EPRESDIV is the value in CBT[EPRESDIV] (CTRL1[PRESDIV] can also be used).
- FPSEG1 is the value in FDCBT[FPSEG1].
- FPSEG2 is the value in FDCBT[FPSEG2].
- FPROPSEG is the value in FDCBT[FPROPSEG].
- FPRESDIV is the value in FDCBT[FPRESDIV].

See also Protocol timing for more details.

The following tables give the TASD value calculated for some configuration cases.

Case 1:

- Clock ratio = 2:1 (example: peripheral clock 80 MHz and oscillator clock 40 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 11-168. TASD values:

Number of message buffers	TASD value	Maximum bit rate in data phase (MBd)
16	24	Invalid
32	24	8.0
64	23	8.0

Case 2:

- Clock ratio = 1:1 (example: peripheral clock 40 MHz and oscillator clock 40 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 11-169. TASD values:

Number of message buffers	TASD value	Maximum bit rate in data phase (MBd)
16	24	Invalid
32	23	6.67
54	22	5.0
64	21	3.33

Case 3:

- Clock ratio = 2:1 (example: peripheral clock 40 MHz and oscillator clock 20 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 11-170. TASD values:

Number of message buffers	TASD value	Maximum bit rate in data phase (MBd)
16	24	Invalid
32	23	4.0
54	22	4.0
64	21	3.33

11.8.2.10 Clocks

The following table describes the clock sources for FlexCAN. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 11-171. FlexCAN clocks

Clock name	Description
ipg_clk	Peripheral clock
ipg_clk_chi	Control Host Interface (CHI) clock
ipg_clk_pe	Protocol Engine (PE) clock
ipg_clk_pe_nogate	Protocol Engine clock (no gating)
ipg_clk_s	Peripheral access clock

11.8.2.10.1 Clock domains and restrictions

The FlexCAN module has two clock domains asynchronous to each other:

- The bus domain feeds the Control Host Interface (CHI) submodule.
- The oscillator domain feeds the CAN Protocol Engine (PE) submodule.

When CTRL1[CLKSRC] is set, synchronous operation occurs because both domains are connected to the peripheral clock (creating a 1:1 ratio between the peripheral and oscillator clocks).

When the two domains are connected to clocks with different frequencies and/or phases, there are restrictions on the frequency relationship between the two clock domains. In the case of asynchronous operation, the bus domain clock frequency must always be greater than the oscillator domain clock frequency.

NOTE

Asynchronous operation with a 1:1 ratio between peripheral and oscillator clocks is not allowed.

When doing matching and arbitration, FlexCAN needs to scan the whole message buffer memory during the time slot of one CAN frame, comprised of a number of CAN bits. In order to have sufficient time to do that, the following requirements must be observed:

- The peripheral clock frequency cannot be smaller than the oscillator clock frequency.
- There must be a minimum number of peripheral clocks per CAN bit, as specified in the table shown below.

Table 11-172. Minimum number of peripheral clocks per CAN bit for Classical CAN format

Number of mailboxes	Value of MCR[RFEN]	Minimum number of peripheral clocks per CAN bit
16	0	16
32	0	16
64	0	25
16	1	16
32	1	17
64	1	30

For classical frame format, the minimum number of peripheral clocks per CAN bit specified in the preceding table determines the minimum peripheral clock frequency for a given number of mailboxes and for an expected CAN bit rate. The CAN bit rate depends on the number of time quanta in a CAN bit, which can be defined by adjusting one or more of the bit timing values contained in either the Control 1 register (CTRL1) or CAN Bit Time register (CBT). The time quantum (Tq) is defined in Protocol timing. The minimum number of time quanta per CAN bit must be 8; therefore, the oscillator clock frequency should be at least 8 times the CAN bit rate.

For CAN FD frame format, there are some constraints that need to be satisfied. The number of peripheral clocks per CAN bit in nominal bit rate (NumClkNomBit) can be calculated by the equation below.

$$\text{NumClkNomBit} = \frac{f_{SYS}}{f_{CANCLK}} \times (\text{PRESDIV} + 1) \times (\text{PROPSEG} + \text{PSEG1} + \text{PSEG2} + 4)$$

$$= \frac{f_{SYS}}{\text{NomBitRate}}$$

Figure 11-133. Number of peripheral clocks per nominal CAN bit

where PRESDIV, PSEG1, and PSEG2 are CAN bit time values in CTRL1 register. Alternatively, EPRESDIV, EPSEG1, and EPSEG2 values in CBT register can be used instead. NumClkNomBit can also be calculated as a function of the expected nominal bit rate used in the arbitration phase (NomBitRate), as shown in the equation above.

The number of CAN bits in the data phase of an FD frame with the BRS bit set (fast CAN bits, in short) depends on the number of data bytes in the payload. The number of fast CAN bits (NumOfFastBits) can be determined in the table below. The fewer the number of data bytes, the fewer the number of fast CAN bits, and less time is available for FlexCAN to scan the whole message buffer memory during the internal matching and arbitration processes.

Table 11-173. Number of fast CAN bits in a CAN FD frame

Minimum number of data bytes	DLC field	NumOfFastBits
0	0h	21
1	1h	29
2	2h	37
3	3h	45
4	4h	53
5	5h	61
6	6h	69
7	7h	77
8	8h	85
12	9h	117
16	Ah	149
20	Bh	186
24	Ch	218
32	Dh	282
48	Eh	410
64	Fh	538

The critical part of a CAN FD frame is during the data phase, where the CAN bit rate is faster than in the arbitration phase. The minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) can be calculated to guarantee that enough time is available for FlexCAN to scan the message buffer memory during reception and transmission. The equation below calculates this constraint.

$$\text{MinNumClkFastBit}_A = \frac{(\ 8.5 \times \text{MaxNumOfMb}\ ) + 64 - (\ 9 \times \text{NumClkNomBit}\ )}{\text{NumOfFastBits}}$$

Figure 11-134. Minimum number of peripheral clocks per fast CAN bit for FlexCAN scan process

where MaxNumOfMb is the maximum number of available mailboxes defined in MCR[MAXMB].

The clock domain crossing circuit between the CHI and PE subblocks also imposes a minimum number of peripheral clocks per fast CAN bit for the handshake mechanism to work properly without losing status information through the interface, as shown in the equation below.

$$MinNumClkFastBit_B = 3 \times \left(1 + \frac{f_{SYS}}{f_{CANCLK}}\right)$$

Figure 11-135. Minimum number of peripheral clocks per fast CAN bit for FlexCAN clock domain interface

Therefore, the minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) is determined by the larger of the two values calculated above.

$$MinNumClkFastBit = Maximum\ (\ MinNumClkFastBit_A, MinNumClkFastBit_B\ )$$

Figure 11-136. Minimum number of peripheral clocks per fast CAN bit

Then, the maximum CAN bit rate in the data phase of CAN FD frames (DataBitRateMAX) can be calculated as below.

$$DataBitrate_{MAX} = \frac{f_{CANCLK}}{ROUNDUP\left(\frac{MinNumClkFastBit \times f_{CANCLK}}{f_{SYS}}\right)}$$

Figure 11-137. Maximum achievable baud rate for data phase

The peripheral and oscillator clock frequencies, the maximum number of mailboxes, and the expected nominal bit rate affect the maximum data bit rate attainable by FlexCAN in CAN FD mode. Also, the data bit rate depends on the minimum payload size of FD frames used in a given application.

To illustrate how the CAN FD bit rate is affected by the configuration of FlexCAN variables, an application example with the peripheral and oscillator clock frequencies set to 50 MHz and 40 MHz, respectively, is considered.

1. Considering the nominal bit rate as 1 Mbps, the number of peripheral clocks per CAN bit in nominal bit rate is calculated as below.

$$NumClkNomBit = \frac{50 \times 10^6}{1 \times 10^6} = 50$$

Figure 11-138. Calculation example for number of peripheral clocks per nominal CAN bit

2. The number of fast CAN bits (NumOfFastBits) is determined in the table presented above. For example, if the minimum payload in FD frames is 8 bytes, then there are 85 CAN bits in the data phase.
3. Assuming the maximum number of mailboxes is 96, the minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) can be calculated.

$$\text{MinNumClkFastBit}_A = \frac{(8.5 \times 96) + 64 - (9 \times 50)}{85} = 5.06$$

Figure 11-139. Calculation example for number of peripheral clocks per fast CAN bit for FlexCAN scan process

$$\text{MinNumClkFastBit}_B = 3 \times \left(1 + \frac{50}{40}\right) = 6.75$$

Figure 11-140. Calculation example for number of peripheral clocks per fast CAN bit for FlexCAN clock domain interface

$$\text{MinNumClkFastBit} = \text{Maximum}\ (5.06, 6.75) = 6.75$$

Figure 11-141. Calculation example for number of peripheral clocks per fast CAN bit

4. The maximum CAN bit rate in the data phase can finally be found.

$$\text{DataBitRate}_{MAX} = \frac{40 \times 10^6}{\text{ROUNDUP}\left(\frac{6.75 \times 40 \times 10^6}{50 \times 10^6}\right)} = 6.667\ \text{Mbps}$$

Figure 11-142. Calculation example for maximum achievable baud rate

As demonstrated in this example, even though the oscillator clock frequency (40 MHz) is adequate to generate a data rate of 8 Mbps in CAN FD mode, the specific FlexCAN configuration limits this rate to 6.667 Mbps. This limitation is mainly due to the low peripheral clock frequency that imposes the MinNumClkFastBitB bound.

The table below shows the maximum data rate for CAN FD according to clock frequencies, payload size, and number of available mailboxes. See in this table that, for some cases, if the number of available mailboxes is reduced, the FlexCAN can then achieve a data rate up to 8 Mbps.

Table 11-174. Maximum CAN bit rate in data phase on CAN FD frames

Peripheral clock frequency (MHz)	Payload size	Number of available mailboxes	Maximum data rate (Mbps)
40	8	94	6.667
40	8	114	5.0
40	12	117	6.667
40	12	128	5.714
50	12 to 64	128	6.667
60	8	126	8.0
60	12	128	8.0
67	6	128	8.0
80	3	128	8.0
100	0	128	8.0

11.8.2.11 Reset

You can reset FlexCAN in the following ways:

1. Chip level hard reset, which resets all memory-mapped registers asynchronously.
2. MCR[SOFTRST], which resets some of the memory-mapped registers synchronously. See Table 11-177 to see what registers are affected by soft reset.

Soft reset is synchronous and has to follow an internal request/acknowledge procedure across clock domains. Therefore, it may take some time to fully propagate its effects. MCR[SOFTRST] remains asserted while soft reset is pending, so software can poll this bit to know when the reset has completed. Also, soft reset cannot be applied while clocks are shut down in a low-power mode. The low-power mode should be exited and the clocks resumed before applying soft reset.

The clock source should be selected when the module is in Disable mode (see CTRL1[CLKSRC] description in Control 1 Register (CTRL1)). After the clock source is selected and the module is enabled (MCR[MDIS] bit negated), FlexCAN automatically enters Freeze mode. In Freeze mode, FlexCAN is un-synchronized to the CAN bus, MCR[HALT] and MCR[FRZ] are set, the internal state machines are disabled, then MCR[FRZACK] and MCR[NOTRDY] are set. The Tx pin is in recessive state and FlexCAN does not initiate any transmission or reception of CAN frames. Note that the message buffers and the Rx Individual Mask registers are not affected by reset, so they are not automatically initialized.

11.8.2.12 Interrupts

The module has many interrupt sources: interrupts due to message buffers and interrupts due to the ORed interrupts from MBs, Bus Off, Bus Off Done, Error, Error Fast (errors detected in the data phase of CAN FD format messages with the BRS bit set), Wake Up, Tx Warning, and Rx Warning.

Each one of the message buffers can be an interrupt source, if its corresponding IMASK bit is set. There is no distinction between Tx and Rx interrupts for a particular buffer, under the assumption that the buffer is initialized for either transmission or reception. Each of the buffers has an assigned flag bit in the IFLAG registers. The bit is set when the corresponding buffer completes a successful transfer and is cleared when the CPU writes one to it (unless another interrupt is generated at the same time).

Note

It must be guaranteed that the CPU clears only the bit causing the current interrupt. For this reason, bit manipulation

instructions (BSET) must not be used to clear interrupt flags. These instructions may cause accidental clearing of interrupt flags which are set after entering the current interrupt service routine.

If the Rx FIFO is enabled (MCR[RFEN] = 1) and DMA is disabled (MCR[DMA] = 0), the interrupts corresponding to MBs 0 to 7 have different meanings. Bit 7 of the IFLAG1 register becomes the "FIFO Overflow" flag; bit 6 becomes the "FIFO Warning" flag, bit 5 becomes the "Frames Available in FIFO" flag and bits 4-0 are unused. See the description of Interrupt Flags 1 Register (IFLAG1) for more information.

If both Rx FIFO and DMA are enabled (MCR[RFEN] and MCR[DMA] = 1) the FlexCAN does not generate any FIFO interrupt. Bit 5 of the IFLAG1 register still indicates "Frames Available in FIFO" and generates a DMA request. Bits 7, 6, 4-0 are unused.

CAUTION

FIFO cannot be enabled when CAN FD feature is enabled.

For a combined interrupt where multiple MB interrupt sources are OR'd together, the interrupt is generated when any of the associated MBs (or FIFO, if applicable) generates an interrupt. In this case, the CPU must read the IFLAG registers to determine which MB or FIFO source caused the interrupt.

The following interrupt sources generate interrupts like the MB interrupt sources, and can be read from ESR1 register

- Bus Off
- Bus Off Done
- Error
- Error Fast
- Wake Up
- Tx Warning
- Rx Warning

The Bus Off, Error, Tx Warning, and Rx Warning interrupt mask bits are located in the CTRL1 register; the Wake-Up interrupt mask bit is located in MCR.

11.8.2.13 Bus interface

CPU access to FlexCAN registers is subject to the following rules:

- Unrestricted read and write access to supervisor registers (registers identified with S/U in Table 11-177 in Supervisor Mode or with S only) results in an access error.

- Read and write access to implemented reserved address space results in an access error.
- Write access to positions whose bits are all currently read-only results in an access error. If at least one of the bits is not read-only then no access error is issued. Write permission to positions or some of their bits can change depending on the mode of operation or transitory state. See register and field descriptions for details.
- Read and write access to unimplemented address space results in access error.
- Read and write access to RAM located positions during Low Power mode results in an access error.
- It is possible for the RXIMR memory region to be considered as general purpose memory and available for access. There are two ways of doing this:
  a. If MCR[IRMQ] is cleared, the individual masks (RXIMR) are disabled. In this case the RXIMR memory region is considered as general purpose memory.
  b. If MCR[MAXMB] is programmed with a value smaller than the available number of MBs, then the unused memory space can be used as general purpose RAM space. Note that reserved words within RAM cannot be used. As an example, suppose FlexCAN's RAM can support up to 16 MBs, CTRL2[RFFN] is 0h, and MCR[MAXMB] is programmed with zero. The maximum number of MBs in this case becomes one. The RAM starts at 0080h, and the space 0080h–008Fh is used by the one MB. The memory space 0090h–017Fh is available. The space 0180h–087Fh is reserved. The space 0880h–0883h is used by the one individual mask and the available memory in the mask registers space would be 0884h–08BFh. From 08C0h–09DFh there are reserved words for internal use which cannot be used as general purpose RAM. As a general rule, free memory space for general purpose depends only on MAXMB.
  c. If MCR[FDEN] is enabled, general purpose memory can be used out of Freeze mode only.

Table 11-175. Access permissions

Supervisor access	0				1		
Supervisor mode - MCR[SUPV] = 1	0			1	any		
Modes of operation	Normal	Freeze	Low power	*	Normal	Freeze	Low power
MCR	bus error	bus error	bus error	bus error	read/write	read/write	read/write
CTRL1	read/write	read write	read/write	bus error	read/write	read/write	read/write
TIMER	read/write	read/write	read/write	bus error	read/write	read/write	read/write
TCR	bus error	bus error	bus error	bus error	bus error	bus error	bus error
RXMGMASK[1]	bus error for write operation	read/write	bus error for write operation	bus error	bus error for write operation	read/write	bus error for write operation

Table continues on the next page...

Table 11-175. Access permissions (continued)

Supervisor access	0				1		
Supervisor mode - MCR[SUPV] = 1	0			1	any		
Modes of operation	Normal	Freeze	Low power	*	Normal	Freeze	Low power
RX14MASK[1]	bus error for write operation	read/write	bus error for write operation	bus error	bus error for write operation	read/write	bus error for write operation
RX15MASK[1]	bus error for write operation	read/write	bus error for write operation	bus error	bus error for write operation	read/write	bus error for write operation
ECR	bus error for write operation	read/write	bus error for write operation	bus error	bus error for write operation	read/write	bus error for write operation
ESR1	read/write	read/write	read/write	bus error	read/write	read/write	read/write
IMASK2	read/write	read/write	read/write	bus error	read/write	read/write	read/write
IMASK1	read/write	read/write	read/write	bus error	read/write	read/write	read/write
IFLAG2	read/write	read/write	read/write	bus error	read/write	read/write	read/write
IFLAG1	read/write	read/write	read/write	bus error	read/write	read/write	read/write
CTRL2	read/write	read/write	read/write	bus error	read/write	read/write	read/write
ESR2	read/write	read/write	read/write	bus error	read/write	read/write	read/write
CRCR	bus error for write operation	bus error for write operation	bus error for write operation	bus error	bus error for write operation	bus error for write operation	bus error for write operation
RXFGMASK[1]	bus error for write operation	read/write	bus error for write operation	bus error	bus error for write operation	read/write	bus error for write operation
RXFIR[1]	bus error for write operation	bus error for write operation	bus error for write operation	bus error	bus error for write operation	bus error for write operation	bus error for write operation
CBT	bus error for write operation	read/write	bus error for write operation	bus error	bus error for write operation	read/write	bus error for write operation
DBG1	bus error for write operation	bus error for write operation	bus error for write operation	bus error for write operation	bus error for write operation	bus error for write operation	bus error for write operation
DBG2	bus error for write operation	bus error for write operation	bus error for write operation	bus error for write operation	bus error for write operation	bus error for write operation	bus error for write operation
MB[1, 2]	read/write	read/write	read/write	bus error	read/write	read/write	read/write
FIFO header[1]	read/write	read/write	read/write	bus error	read/write	read/write	read/write
FIFO reserved space[1]	bus error	bus error	bus error	bus error	bus error	bus error	bus error
FIFO filters[1]	read/write	read/write	read/write	bus error	read/write	read/write	read/write
RXIMR[1]	bus error	read/write	bus error	bus error	bus error	read/write	bus error
MECR	read/write	read/write	read/write	bus error	read/write	read/write	read/write
ERRIAR	read/write	read/write	read/write	bus error	read/write	read/write	read/write

Table continues on the next page...

Table 11-175. Access permissions (continued)

Supervisor access	0				1		
Supervisor mode - MCR[SUPV] = 1	0			1	any		
Modes of operation	Normal	Freeze	Low power	*	Normal	Freeze	Low power
ERRIDPR	read/write	read/write	read/write	bus error	read/write	read/write	read/write
ERRIPPR	read/write	read/write	read/write	bus error	read/write	read/write	read/write
RERRAR[3]	read/write	read/write	read/write	bus error	read/write	read/write	read/write
RERRDR[3]	read/write	read/write	read/write	bus error	read/write	read/write	read/write
RERRSYNR[3]	read/write	read/write	read/write	bus error	read/write	read/write	read/write
ERRSR	read/write	read/write	read/write	bus error	read/write	read/write	read/write
FDCTRL	read/write	read/write	read/write	bus error	read/write	read/write	read/write
FDCBT	read/write[3]	read/write	read/write[3]	bus error	read/write[3]	read/write	read/write[3]
FDCRC	bus error for write operation	bus error for write operation	bus error for write operation	bus error	bus error for write operation	bus error for write operation	bus error for write operation
General purpose RAM[1]	read/write	read/write	read/write	bus error	read/write	read/write	read/write
Reserved space (used)[1]	bus error	bus error	bus error	bus error	bus error	bus error	bus error
Reserved space (empty)[1]	bus error	bus error	bus error	bus error	bus error	bus error	bus error

1. Access in low power is only possible if RAM_clk and ipg_clk are enabled.
2. If MCR[RFEN] = 1, see FIFO access rules below.
3. Write operation has no effect.

11.8.2.14 Detection and correction of memory errors

NOTE

All FlexCAN memory must be initialized before starting its operation in order to have the parity bits in memory properly updated. CTRL2[WRMFRZ] grants write access to all memory positions that require initialization, ranging from 080h–ADFh and from F28h–FFFh when the CAN FD feature is enabled. The RXMGMASK, RX14MASK, RX15MASK, and RXFGMASK registers need to be initialized as well. MCR[RFEN] must not be set during memory initialization.

FlexCAN supports detection and correction of errors in memory read accesses. Each byte of FlexCAN memory is associated to five parity bits that ensure a Hamming distance of four. The error correction mechanism ensures that in this 13-bit word, errors in one bit can be corrected (correctable errors) and errors in two bits can be detected but not corrected (non-correctable errors). Errors in more than two bits may not be detected. In

case of non-correctable errors, the corrupted data is not changed by the error correction logic. When a read access is performed, the parity bits are used to calculate a syndrome, which indicates the error in each byte.

FlexCAN detects a noncorrectable error in the event either an all-zeros or an all-ones read occurs. See description of Error Report Syndrome Register (RERRSYNR).

Memory errors are indicated to the host through status register (Error Status Register (ERRSR)) and bus transfer errors, and reported through report registers (Error Report Address register (RERRAR), Error Report Data register (RERRDR) and Error Report Syndrome register (RERRSYNR)).

The error detection and correction mechanism can be activated or not, controlled by MECR[ECCDIS]. When disabled, updates on indications and reporting registers are stopped, but the parity bits are still calculated and written along with data in memory write operations to ensure that memory has consistent parity bits associated with the data.

To avoid accidentally changing the critical error correction configuration, this protocol must be followed to enable the update of the Memory Error Control register (MECR):

1. By default, CTRL2[ECRWRE] is zero and MECR[ECRWRDIS] is one.
2. Set CTRL2[ECRWRE].
3. Clear MECR[ECRWRDIS].
4. All writes to Memory Error Control register (MECR) must keep MECR[ECRWRDIS] cleared.
5. After configuration is done, lock the Memory Error Control register (MECR) by either setting MECR[ECRWRDIS] or clearing CTRL2[ECRWRE].

11.8.2.14.1 Sources of the memory access

The FlexCAN memory can be accessed by two major sources (or requestors):

- Host (CPU): the largest word accessed is 32-bit.
- FlexCAN internal processes (Rx matching, Tx arbitration, move-in on reception, move-out on transmission): the largest word accessed is 64-bit.

The way that noncorrectable errors are indicated and reported depends on the source of access.

11.8.2.14.2 Error indication

Memory errors are indicated by flags HANCEIF, FANCEIF, and CEIF in the Error Status Register (ERRSR). Noncorrectable errors detected in memory reads requested by the host are indicated separately than the ones detected in requests by FlexCAN internal processes. FlexCAN makes no distinction of the source of the access when correctable

errors are detected. There are three independent flags for these three cases, and each flag will raise an interrupt unless it is masked by mask bits in Memory Error Control Register (MECR). If both noncorrectable and correctable errors are found in different bytes in the same read operation, both flags are set.

A noncorrectable error detected in host access is also indicated as a bus transfer error. A bus wait request may be asserted to extend the memory transaction to the moment the report registers are updated. This indication cannot be masked. If the flag bit ERRSR[HANCEIF] is not masked, the same noncorrectable error will raise a bus transfer error and an interrupt request.

Each indication flag has one overrun flag in Error Status Register (ERRSR). The overrun flags do not request interrupts. Overrun flags for noncorrectable errors indicate that other errors of the same nature were detected after current error being treated; overrun flags for correctable errors indicate that other errors of the same nature were detected before the current error being treated.

This is the recommended handling sequence for error indication:

1. Get error report information from report registers.
2. Use this information to take proper measures in the application.
3. Clear the HANCEIF, FANCEIF, and CEIF flags.
4. If the overrun flag is active:
    a. Alert application that at least one error could not be managed.
    b. Clear the overrun flag.

The FlexCAN internal processes can access memory in transactions larger than 32 bits. For the indication, this kind of access is considered a consecutive sequence of 32-bit accesses. If errors are found in two or more 32-bit words the interrupt and overrun flags are set simultaneously.

11.8.2.14.3 Error reporting

The report registers Error Report Address register (RERRAR), Error Report Data register (RERRDR), and Error Report Syndrome register (RERRSYNR) provide detailed information about the address read, raw data, and syndrome read with error, and are indicated by the flags described in Error indication. The address, data, and syndrome registers are updated simultaneously along with the error flags, according to these rules:

1. If either of the two non-correctable error flags is currently set, the report registers are not updated (the previous non-correctable error reporting is preserved).
2. Otherwise (either no error flag is currently set or only the correctable error flag is currently set), the report registers are updated according to the new error; or

according to the most severe of new errors if non-correctable and correctable errors are simultaneously detected.

Reporting of errors detected in accesses larger than 32-bit follows the rules described in Error indication and in the description of Error Report Address Register (RERRAR).

The address reported in RERRAR and defined in ERRIAR are not the same listed in the module memory map. The relation between the reported addresses and the respective ones in the module memory map is shown in the description of Error Injection Address Register (ERRIAR).

Addresses reported when reading memory portions organized as FIFOs, such as the Rx FIFO Structure and the Rx FIFO Information Register (RXFIR), refer to the address of the specific entry accessed in the FIFO, not to the FIFO base address.

To assure coherence of the error report registers it is necessary to turn off the report update by setting MECR[RERRDIS] before reading the report registers.

11.8.2.14.4 Response to errors

Correctable errors have no consequence on FlexCAN operation because affected data is corrected before its use by the host or FlexCAN internal processes.

For host-initiated reads, a noncorrectable error may affect the host, but does not affect FlexCAN operation.

Non-correctable errors detected on memory reads requested by the FlexCAN internal processes may result in incorrect operation depending on the state of MECR[NCEFAFRZ], as follows:

- During reception (either matching or move-in processes), when a noncorrectable error occurs, an incorrect destination may be selected to store the incoming frame, a corrupted frame may be stored in the correct destination, or both. If MECR[NCEFAFRZ] is set, FlexCAN stops operation automatically and enters in Freeze mode to prevent corrupted data from being treated as valid by FlexCAN internal processes. When MECR[NCEFAFRZ] is negated, FlexCAN continues working and a corrupted frame is received.
- During arbitration process, when a noncorrectable error occurs, either a non-highest priority Tx message buffer may be mistakenly selected for transmission or its data may be corrupted. If MECR[NCEFAFRZ] is set, FlexCAN stops operation automatically and enters Freeze mode before starting the move-out. When MECR[NCEFAFRZ] is negated, FlexCAN proceeds to move-out with a corrupted frame that will be transmitted on the CAN bus.
- During move-out process, when a noncorrectable error occurs, a corrupted frame is copied from the selected Tx MB that won the arbitration to the Tx SMB for

transmission. If MECR[NCEFAFRZ] is set, FlexCAN stops operation automatically and enters Freeze mode before starting the transmission. When MECR[NCEFAFRZ] is negated, the corrupted frame is transferred from the Tx SMB to the Protocol Engine (PE) sub-block and is transmitted on the CAN bus.

- A noncorrectable error can also be detected beyond the move-out process, when Tx data is read from Tx SMB (buffer located in RAM) to be transferred to the PE subblock for transmission. In this case, a frame with corrupted ID and/or data is transmitted on the CAN bus.

  To prevent the frame from being successfully received by the external nodes, FlexCAN inverts all bits in the CRC field (CRC sequence plus CRC delimiter), and transmits an error flag just after CRC delimiter as a result of self-detecting a Bit1 error and a form error due to the CRC field inversion. When MECR[NCEFAFRZ] is set, FlexCAN stops operation automatically and enters Freeze mode just after the error frame. When MECR[NCEFAFRZ] is negated, FlexCAN may attempt to re-transmit the same frame, as long as no other higher priority Tx MB is subsequently configured for transmission.

  In the event the noncorrectable error persists, FlexCAN eventually reaches the Bus Off state because of consecutive error detections. ECR[TXERRCNT] is updated every time FlexCAN inverts the CRC field, causing errors as described above.

When MECR[NCEFAFRZ] is set and FlexCAN enters Freeze mode, only the CPU can cause FlexCAN to exit Freeze mode and resume Normal mode. The assertion of MECR[NCEFAFRZ] is the only way to prevent corrupted frames from being transmitted on the CAN bus up to the move-out internal process.

The error report registers can provide information to the application for customized management of these situations.

11.8.2.14.5 Error injection

The error injection registers ERRIAR, ERRIDPR, and ERRIPPR are used to inject errors in memory reads in order to force errors and consequently update the indication and reporting registers. The relation between the error injection addresses and the respective ones in the module memory map is shown in the description of Error Injection Address Register (ERRIAR).

The injection is done by flipping the data and parity bits correspondent to the bits in 1 in ERRIDPR and ERRIPPR. Injection can be selected specifically for memory accesses requested by the host or by FlexCAN internal processes.

In case of accesses larger than 32-bits, the EXTERRIE bit in Memory Error Control register (MECR) extends the injection pattern, replicating it in 32-bit words to fill the width of the access.

NOTE

It is very unlikely, but error injection may correct a bit with error. This will not raise the error flags and reports as expected.

To ensure coherence among error injection registers and avoid spurious error injections, MECR[HAERRIE] and MECR[FAERRIE] must be cleared when configuring the memory injection registers.

11.8.3 FlexCAN signal descriptions

The FlexCAN module has two I/O signals connected to the external chip pins. These signals are summarized in the following table and described in more detail in the next subsections.

Table 11-176. FlexCAN signal descriptions

Signal	Description	I/O
CAN Rx	CAN receive pin	Input
CAN Tx	CAN transmit pin	Output

11.8.3.1 CAN Rx

This pin is the receive pin from the CAN bus transceiver. Dominant state is represented by logic level 0. Recessive state is represented by logic level 1.

11.8.3.2 CAN Tx

This pin is the transmit pin to the CAN bus transceiver. Dominant state is represented by logic level 0. Recessive state is represented by logic level 1.

11.8.4 Initialization/application information

This section provides instructions for initializing the FlexCAN module.

11.8.4.1 FlexCAN initialization sequence

For any configuration change/initialization it is required that FlexCAN be put into Freeze mode (see Freeze mode). Note that the module needs to be initialized after every reset.

The following is a generic initialization sequence applicable to the FlexCAN module:

1. Initialize the Module Configuration register (MCR).
   a. Enable the individual filtering per MB and reception queue features by setting IRMQ.
   b. Enable the warning interrupts by setting WRNEN.
   c. If required, disable frame self reception by setting SRXDIS.
   d. Enable the Rx FIFO by setting RFEN.
   e. If Rx FIFO is enabled and DMA is required, set DMA.
   f. Enable the abort mechanism by setting AEN.
   g. Enable the local priority feature by setting LPRIOEN.
2. Initialize the Control 1 register (CTRL1) and optionally the CAN Bit Timing register (CBT). Initialize also the CAN FD CAN Bit Timing register (FDCBT).
   a. Determine the bit timing parameters: PROPSEG, PSEG1, PSEG2, and RJW.
   b. Optionally determine the bit timing parameters: EPROPSEG, EPSEG1, EPSEG2, and ERJW.
   c. Determine the CAN FD bit timing parameters: FPROPSEG, FPSEG1, FPSEG2, and FRJW.
   d. Determine the bit rate by programming the PRESDIV field and optionally the EPRESDIV field.
   e. Determine the CAN FD bit rate by programming the FPRESDIV field.
   f. Determine the internal arbitration mode (LBUF).
3. All FlexCAN memory must be initialized if Error Code Correction (ECC) is enabled. See Detection and correction of memory errors.
4. Initialize the message buffers.
   a. The control and status word of all message buffers must be initialized.
   b. If Rx FIFO was enabled, the ID filter table must be initialized.
   c. Other entries in each message buffer should be initialized as required.
5. Initialize the Rx Individual Mask registers (RXIMRn).
6. Set required interrupt mask bits in
   - IMASK registers (for all MB interrupts)
   - MCR register (for Wake-Up interrupt)
   - CTRL1 / CTRL2 registers (for Bus Off and Error interrupts)
7. Negate MCR[HALT].

After the last step listed above, FlexCAN attempts to synchronize to the CAN bus.

11.8.5 Memory map/register definition

This section describes the registers and data structures in the FlexCAN module. The base address of the module depends on the particular memory map of the chip.

11.8.5.1 FlexCAN memory mapping

The address space occupied by FlexCAN has 128 bytes for registers starting at the module base address, followed by embedded RAM starting at address offset 0080h.

Each individual register is identified by its complete name and the corresponding mnemonic . The access type can be Supervisor (S) or Unrestricted (U). Most of the registers can be configured to have either Supervisor or Unrestricted access by programming MCR[SUPV]. These registers are identified as S/U in the Access column of Table 11-177.

NOTE

An invalid register access will result in a bus error. This includes reading a write-only register, writing a read-only register, or accessing an invalid address.

Table 11-177. Register access and reset information

Register	Access type	Affected by hard reset	Affected by soft reset
Module Configuration Register (MCR)	S	Yes	Yes
Control 1 Register (CTRL1)	S/U	Yes	No
Free Running Timer Register (TIMER)	S/U	Yes	Yes
Rx Mailboxes Global Mask Register (RXMGMASK)	S/U	No	No
Rx Buffer 14 Mask Register (RX14MASK)	S/U	No	No
Rx Buffer 15 Mask Register (RX15MASK)	S/U	No	No
Error Counter Register (ECR)	S/U	Yes	Yes
Error and Status 1 Register (ESR1)	S/U	Yes	Yes
Interrupt Masks 2 Register (IMASK2)	S/U	Yes	Yes
Interrupt Masks 1 Register (IMASK1)	S/U	Yes	Yes
Interrupt Flags 2 Register (IFLAG2)	S/U	Yes	Yes
Interrupt Flags 1 Register (IFLAG1)	S/U	Yes	Yes
Control 2 Register (CTRL2)	S/U	Yes	No
Error and Status 2 Register (ESR2)	S/U	Yes	Yes
CRC Register (CRCR)	S/U	Yes	Yes
Rx FIFO Global Mask Register (RXFGMASK)	S/U	No	No

Table continues on the next page...

Table 11-177. Register access and reset information (continued)

Register	Access type	Affected by hard reset	Affected by soft reset
Rx FIFO Information Register (RXFIR)	S/U	No	No
CAN Bit Timing Register (CBT)	S/U	Yes	No
Message buffers	S/U	No	No
Rx Individual Mask Registers	S/U	No	No
Memory Error Control Register (MECR)	S/U	Yes	Yes
Error Injection Address Register (ERRIAR)	S/U	Yes	Yes
Error Injection Data Pattern Register (ERRIDPR)	S/U	Yes	Yes
Error Injection Parity Pattern Register (ERRIPPR)	S/U	Yes	Yes
Error Report Address Register (RERRAR)	S/U	Yes	Yes
Error Report Data Register (RERRDR)	S/U	Yes	Yes
Error Report Syndrome Register (RERRSYNR)	S/U	Yes	Yes
Error Status Register (ERRSR)	S/U	Yes	Yes
CAN FD Control Register (FDCTRL)	S/U	Yes	No
CAN FD Bit Timing Register (FDCBT)	S/U	Yes	No
CAN FD CRC Register (FDCRC)	S/U	Yes	Yes

The FlexCAN module can store CAN messages for transmission and reception using mailboxes and Rx FIFO structures.

11.8.5.2 CAN register descriptions

The table below shows the FlexCAN memory map.

The address range from offset 80h–47Fh allocates the sixty-four 128-bit message buffers (MBs).

The memory maps for the message buffers are in FlexCAN message buffer memory map.

11.8.5.2.1 CAN memory map

CAN1 base address: 308C_0000h

CAN2 base address: 308D_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Module Configuration Register (MCR)	32	RW	5980_000Fh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4h	Control 1 Register (CTRL1)	32	RW	0000_0000h
8h	Free Running Timer (TIMER)	32	RW	0000_0000h
10h	Rx Mailboxes Global Mask Register (RXMGMASK)	32	RW	Table 11-177
14h	Rx 14 Mask Register (RX14MASK)	32	RW	Table 11-177
18h	Rx 15 Mask Register (RX15MASK)	32	RW	Table 11-177
1Ch	Error Counter (ECR)	32	RW	0000_0000h
20h	Error and Status 1 Register (ESR1)	32	R2C	0000_0000h
24h	Interrupt Masks 2 Register (IMASK2)	32	RW	0000_0000h
28h	Interrupt Masks 1 Register (IMASK1)	32	RW	0000_0000h
2Ch	Interrupt Flags 2 Register (IFLAG2)	32	W1C	0000_0000h
30h	Interrupt Flags 1 Register (IFLAG1)	32	W1C	0000_0000h
34h	Control 2 Register (CTRL2)	32	RW	0080_0000h
38h	Error and Status 2 Register (ESR2)	32	RO	0000_0000h
44h	CRC Register (CRCR)	32	RO	0000_0000h
48h	Rx FIFO Global Mask Register (RXFGMASK)	32	RW	Table 11-180
4Ch	Rx FIFO Information Register (RXFIR)	32	RO	Table 11-180
50h	CAN Bit Timing Register (CBT)	32	RW	0000_0000h
880h - 97Ch	Rx Individual Mask Registers (RXIMR0 - RXIMR63)	32	RW	Table 11-180
AE0h	Memory Error Control Register (MECR)	32	RW	800C_0080h
AE4h	Error Injection Address Register (ERRIAR)	32	RW	0000_0000h
AE8h	Error Injection Data Pattern Register (ERRIDPR)	32	RW	0000_0000h
AECh	Error Injection Parity Pattern Register (ERRIPPR)	32	RW	0000_0000h
AF0h	Error Report Address Register (RERRAR)	32	RO	0000_0000h
AF4h	Error Report Data Register (RERRDR)	32	RO	0000_0000h
AF8h	Error Report Syndrome Register (RERRSYNR)	32	RO	0000_0000h
AFCh	Error Status Register (ERRSR)	32	W1C	0000_0000h
C00h	CAN FD Control Register (FDCTRL)	32	RW	8000_0100h
C04h	CAN FD Bit Timing Register (FDCBT)	32	RW	0000_0000h
C08h	CAN FD CRC Register (FDCRC)	32	RO	0000_0000h

11.8.5.2.2 Module Configuration Register (MCR)

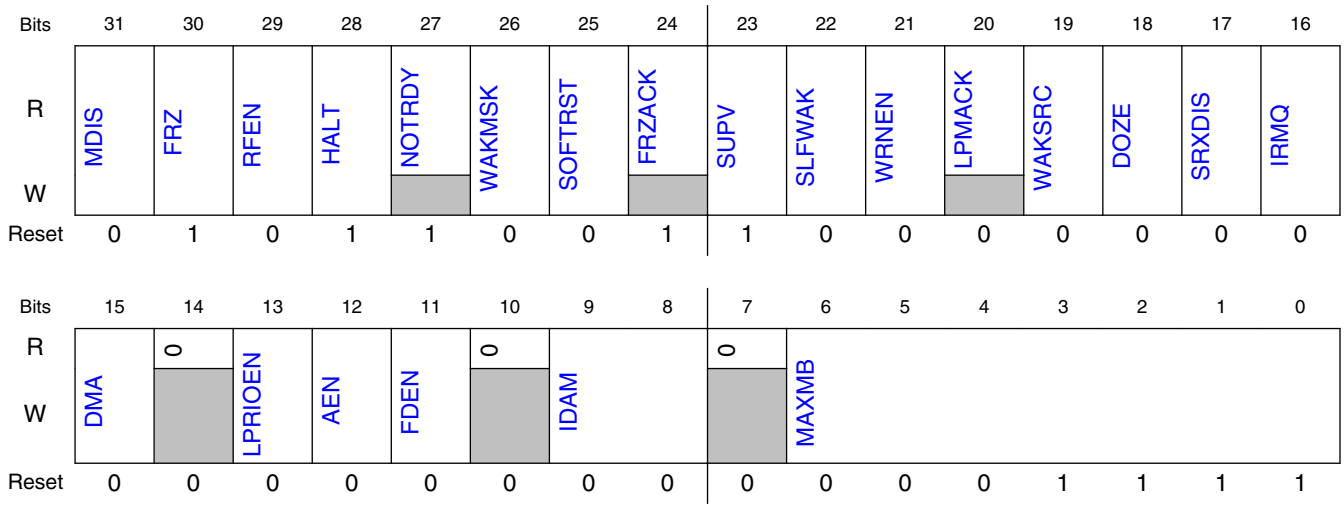
11.8.5.2.2.1 Offset

Register	Offset
MCR	0h

11.8.5.2.2.2 Function

This register defines global system configurations, such as the module operation modes and the maximum message buffer configuration.

11.8.5.2.2.3 Diagram



11.8.5.2.2.4 Fields

Field	Function
31 MDIS	Module Disable This bit controls whether FlexCAN is enabled or not. When disabled, FlexCAN disables the clocks to the CAN Protocol Engine and Controller Host Interface submodules. This bit is not affected by soft reset. 0b - Enable the FlexCAN module. 1b - Disable the FlexCAN module.
30 FRZ	Freeze Enable The FRZ bit specifies the FlexCAN behavior when MCR[HALT] is set or when Debug mode is requested at chip level. When FRZ is asserted, FlexCAN is enabled to enter Freeze mode. Negation of this bit field causes FlexCAN to exit from Freeze mode. This bit is set by hardware when a non-correctable error is detected (MECR[NCEFAFRZ] is asserted). 0b - Not enabled to enter Freeze mode. 1b - Enabled to enter Freeze mode.
29 RFEN	Rx FIFO Enable This bit controls whether the Rx FIFO feature is enabled or not. When RFEN is set, MBs 0 to 5 cannot be used for normal reception and transmission because the corresponding memory region (80h–DCh) is used by the FIFO engine as well as additional MBs (up to 32, depending on CTRL2[RFFN] setting) which are used as Rx FIFO ID filter table elements. RFEN also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in Table 11-172. This bit can be written in Freeze mode only, because it is blocked by hardware in other modes. **NOTE:** This bit cannot be set when CAN FD operation is enabled (see FDEN bit). 0b - Rx FIFO not enabled. 1b - Rx FIFO enabled.

Table continues on the next page...

Field	Function
28 HALT	Halt FlexCAN Assertion of this bit puts the FlexCAN module into Freeze mode. The CPU should clear it after initializing the message buffers and the Control registers CTRL1 and CTRL2. No reception or transmission is performed by FlexCAN before this bit is cleared. Freeze mode cannot be entered when FlexCAN is in a low power mode. The HALT bit is set by hardware when a non-correctable error is detected and MECR[NCEFAFRZ] is asserted. 0b - No Freeze mode request. 1b - Enters Freeze mode if the FRZ bit is asserted.
27 NOTRDY	FlexCAN Not Ready This read-only bit indicates that FlexCAN is either in Disable mode, Doze mode, Stop mode or Freeze mode. It is negated when FlexCAN has exited these modes. This bit is not affected by soft reset. 0b - FlexCAN module is either in Normal mode, Listen-Only mode, or Loop-Back mode. 1b - FlexCAN module is either in Disable mode, Doze mode, Stop mode, or Freeze mode.
26 WAKMSK	Wake Up Interrupt Mask This bit enables the Wake Up interrupt generation under Self Wake Up mechanism. 0b - Wake Up interrupt is disabled. 1b - Wake Up interrupt is enabled.
25 SOFTRST	Soft Reset When SOFTRST is asserted, FlexCAN resets its internal state machines and some of the memory-mapped registers. SOFTRST can be asserted directly by the CPU when it writes to the MCR register. Because soft reset is synchronous and has to follow a request/acknowledge procedure across clock domains, it may take some time to fully propagate its effect. The SOFTRST bit remains asserted when reset is pending, and is automatically negated when reset completes. Therefore, software can poll this bit to know when the soft reset has completed. Soft reset cannot be applied when clocks are shut down in a low power mode. The module should be first removed from low power mode, and then soft reset can be applied. This bit is not affected by soft reset. 0b - No reset request. 1b - Resets the registers affected by soft reset.
24 FRZACK	Freeze Mode Acknowledge This read-only bit indicates that FlexCAN is in Freeze mode and its prescaler is stopped. The Freeze mode request cannot be granted until current transmission or reception processes have finished. Therefore the software can poll the FRZACK bit to know when FlexCAN has actually entered Freeze mode. If Freeze mode request is negated, then this bit is negated after the FlexCAN prescaler is running again. If Freeze mode is requested when FlexCAN is in a low power mode, then the FRZACK bit will be set only when the low-power mode is exited. See Freeze mode. This bit is not affected by soft reset. **NOTE:** FRZACK will be asserted within 178 CAN bits from the Freeze mode request by the CPU, and negated within 2 CAN bits after the Freeze mode request removal (see Protocol timing). 0b - FlexCAN not in Freeze mode, prescaler running. 1b - FlexCAN in Freeze mode, prescaler stopped.
23 SUPV	Supervisor Mode This bit configures the FlexCAN to be either in Supervisor or User mode. The registers affected by this bit are marked as S/U in the "Access type" column of Table 11-177. Reset value of this bit is 1, so the affected registers start with Supervisor access allowance only. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - FlexCAN is in User mode. Affected registers allow both Supervisor and Unrestricted accesses. 1b - FlexCAN is in Supervisor mode. Affected registers allow only Supervisor access. Unrestricted access behaves as though the access was done to an unimplemented register location.
22	Self Wake Up

Table continues on the next page...

Field	Function
SLFWAK	This bit enables the Self Wake Up feature when FlexCAN is in a low-power mode other than Disable mode. When this feature is enabled, the FlexCAN module monitors the bus for wakeup event, that is, a recessive-to-dominant transition. If a wakeup event is detected during Doze mode, FlexCAN requests to resume its clocks and, if enabled to do so, generates a Wake Up interrupt to the CPU. If a wakeup event is detected during Stop mode, then FlexCAN generates, if enabled to do so, a Wake Up interrupt to the CPU so that it can exit Stop mode globally and FlexCAN can request to resume the clocks. When FlexCAN is in a low-power mode other than Disable mode, this bit cannot be written as it is blocked by hardware. 0b - FlexCAN Self Wake Up feature is disabled. 1b - FlexCAN Self Wake Up feature is enabled.
21 WRNEN	Warning Interrupt Enable When asserted, this bit enables the generation of the TWRNINT and RWRNINT flags in the Error and Status Register 1 (ESR1). If WRNEN is negated, the TWRNINT and RWRNINT flags will always be zero, independent of the values of the error counters, and no warning interrupt will ever be generated. This bit can be written in Freeze mode only, because it is blocked by hardware in other modes. 0b - TWRNINT and RWRNINT bits are zero, independent of the values in the error counters. 1b - TWRNINT and RWRNINT bits are set when the respective error counter transitions from less than 96 to greater than or equal to 96.
20 LPMACK	Low-Power Mode Acknowledge This read-only bit indicates that FlexCAN is in a low-power mode (Disable mode, Doze mode, Stop mode). A low-power mode cannot be entered until all current transmission or reception processes have finished, so the CPU can poll the LPMACK bit to know when FlexCAN has actually entered low power mode. This bit is not affected by soft reset. **NOTE:** LPMACK will be asserted within 180 CAN bits from the low-power mode request by the CPU, and negated within 2 CAN bits after the low-power mode request removal (see Protocol timing). 0b - FlexCAN is not in a low-power mode. 1b - FlexCAN is in a low-power mode.
19 WAKSRC	Wake Up Source This bit defines whether the integrated low-pass filter is applied to protect the Rx CAN input from spurious wakeup. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - FlexCAN uses the unfiltered Rx input to detect recessive to dominant edges on the CAN bus. 1b - FlexCAN uses the filtered Rx input to detect recessive to dominant edges on the CAN bus.
18 DOZE	Doze Mode Enable This bit defines whether FlexCAN is allowed to enter low-power mode when Doze mode is requested at chip level. This bit is automatically reset when FlexCAN wakes up from Doze mode upon detecting activity on the CAN bus (self wakeup enabled). 0b - FlexCAN is not enabled to enter low-power mode when Doze mode is requested. 1b - FlexCAN is enabled to enter low-power mode when Doze mode is requested.
17 SRXDIS	Self Reception Disable This bit defines whether FlexCAN is allowed to receive frames transmitted by itself. If this bit is asserted, frames transmitted by the module will not be stored in any MB, regardless if the MB is programmed with an ID that matches the transmitted frame, and no interrupt flag or interrupt signal will be generated due to the frame reception. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Self-reception enabled. 1b - Self-reception disabled.
16	Individual Rx Masking And Queue Enable

Table continues on the next page...

Field	Function
IRMQ	This bit indicates whether Rx matching process will be based either on individual masking and queue or on masking scheme with RXMGMASK, RX14MASK, RX15MASK, and RXFGMASK. This bit can be written in Freeze mode only because it is blocked by hardware in other modes. 0b - Individual Rx masking and queue feature are disabled. For backward compatibility with legacy applications, the reading of C/S word locks the MB even if it is EMPTY. 1b - Individual Rx masking and queue feature are enabled.
15 DMA	DMA Enable DMA controls whether the DMA feature is enabled or not. The DMA feature can only be used in Rx FIFO, so consequently MCR[RFEN] must be asserted. When DMA and RFEN are set, IFLAG1[BUF5I] generates the DMA request and no RX FIFO interrupt is generated. This bit can be written in Freeze mode only as it is blocked by hardware in other modes. 0b - DMA feature for RX FIFO disabled. 1b - DMA feature for RX FIFO enabled.
14 —	Reserved
13 LPRIOEN	Local Priority Enable This bit is provided for backwards compatibility with legacy applications. It controls whether the local priority feature is enabled or not. It is used to expand the ID used during the arbitration process. With this expanded ID concept, the arbitration process is done based on the full 32-bit word, but the actual transmitted ID still has 11-bit for standard frames and 29-bit for extended frames. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Local Priority disabled. 1b - Local Priority enabled.
12 AEN	Abort Enable When asserted, this bit enables the Tx abort mechanism. This mechanism guarantees a safe procedure for aborting a pending transmission, so that no frame is sent in the CAN bus without notification. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. **NOTE:** When MCR[AEN] is asserted, only the abort mechanism (see Transmission abort mechanism) must be used for updating mailboxes configured for transmission. **CAUTION:** Writing the Abort code into Rx mailboxes can cause unpredictable results when MCR[AEN] is asserted. 0b - Abort disabled. 1b - Abort enabled.
11 FDEN	CAN FD operation enable This bit enables the CAN with Flexible Data rate (CAN FD) operation. This bit can be written in Freeze mode only. **NOTE:** FlexCAN is able to transmit FD frame format according to ISO 11898-1. **NOTE:** The Rx FIFO Enable (RFEN) bit cannot be set if FDEN is asserted. 0b - CAN FD is disabled. FlexCAN is able to receive and transmit messages in CAN 2.0 format. 1b - CAN FD is enabled. FlexCAN is able to receive and transmit messages in both CAN FD and CAN 2.0 formats.
10 —	Reserved
9-8 IDAM	ID Acceptance Mode This 2-bit field identifies the format of the Rx FIFO ID filter table elements. Note that all elements of the table are configured at the same time by this field (they are all the same format). See Rx FIFO structure. This field can be written only in Freeze mode because it is blocked by hardware in other modes. 00b - Format A: One full ID (standard and extended) per ID filter table element.

Table continues on the next page...

Field	Function
	01b - Format B: Two full standard IDs or two partial 14-bit (standard and extended) IDs per ID filter table element. 10b - Format C: Four partial 8-bit standard IDs per ID filter table element. 11b - Format D: All frames rejected.
7 —	Reserved
6-0 MAXMB	Number Of The Last Message Buffer This 7-bit field defines the number of the last message buffers that will take part in the matching and arbitration processes. The reset value (0Fh) is equivalent to a 16 MB configuration. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Number of the last MB = MAXMB **NOTE:** MAXMB must be programmed with a value smaller than or equal to the number of available message buffers, as described in FlexCAN memory partition for CAN FD. Additionally, the definition of MAXMB value must take into account the region of MBs occupied by Rx FIFO and its ID filters table space defined by CTRL2[RFFN]. MAXMB also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in Table 11-172.

11.8.5.2.3 Control 1 Register (CTRL1)

11.8.5.2.3.1 Offset

Register	Offset
CTRL1	4h

11.8.5.2.3.2 Function

This Register is defined for specific FlexCAN control features related to the CAN bus, such as bit rate, programmable sampling point within an Rx bit, Loop Back mode, Listen-Only mode, Bus Off recovery behavior, and interrupt enabling (Bus-Off, Error, Warning). It also determines the division factor for the clock prescaler.

The CAN bit timing variables (PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW) can also be configured in the CBT register, which extends the range of all these variables. If CBT[BTF] is set, PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW fields of CTRL1 become read only.

NOTE

When the CAN FD feature is enabled, do not use the PRESDIV, RJW, PSEG1, PSEG2, and PROPSEG fields of the CTRL1 register for CAN bit timing. Instead use the CBT

register's EPRESDIV, ERJW, EPSEG1, EPSEG2, and EPROPSEG fields.

NOTE

The CAN bit variables in CTRL1 and in CBT are stored in the same register.

The contents of this register are not affected by soft reset.

11.8.5.2.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PRESDIV								RJW		PSEG1			PSEG2		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BOFFMSK	ERRMSK	CLKSRC	LPB	TWRNMSK	RWRNMSK	Reserved	Reserved	SMP	BOFFREC	TSYN	LBUF	LOM	PROPSEG		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.3.4 Fields

Field	Function
31-24 PRESDIV	Prescaler Division Factor This 8-bit field defines the ratio between the PE clock frequency and the serial clock (Sclock) frequency. The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the PE clock frequency. The maximum value of this field is FFh, which gives a minimum Sclock frequency equal to the PE clock frequency divided by 256. See Protocol timing for more information. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Sclock frequency = PE clock frequency / (PRESDIV + 1).
23-22 RJW	Resync Jump Width This 2-bit field defines the maximum number of time quanta that a bit time can be changed by one resynchronization. One time quantum is equal to the Sclock period. The valid programmable values are 0–3. See Protocol timing for more information. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Resync Jump Width = RJW + 1.
21-19 PSEG1	Phase Segment 1 This 3-bit field defines the length of phase segment 1 in the bit time. The valid programmable values are 0–7. See Protocol timing for more information. This field can be written only in Freeze mode because it is blocked by hardware in other modes.

Table continues on the next page...

Field	Function
	Phase Buffer Segment 1 = (PSEG1 + 1) × Time-Quanta.
18-16 PSEG2	Phase Segment 2 This 3-bit field defines the length of phase segment 2 in the bit time. The valid programmable values are 1–7. See Protocol timing for more information. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 2 = (PSEG2 + 1) × Time-Quanta.
15 BOFFMSK	Bus Off Interrupt Mask This bit provides a mask for the Bus Off interrupt ESR1[BOFFINT]. 0b - Bus Off interrupt disabled. 1b - Bus Off interrupt enabled.
14 ERRMSK	Error Interrupt Mask This bit provides a mask for the Error interrupt ESR1[ERRINT]. 0b - Error interrupt disabled. 1b - Error interrupt enabled.
13 CLKSRC	CAN Engine Clock Source This bit selects the clock source to the CAN Protocol Engine (PE) to be either the peripheral clock or the oscillator clock. The selected clock is the one fed to the prescaler to generate the serial clock (Sclock). In order to guarantee reliable operation, this bit can be written only in Disable mode because it is blocked by hardware in other modes. See Protocol timing. **NOTE:** The user must ensure the protocol engine clock tolerance according to the CAN Protocol standard (ISO 11898-1). **NOTE:** See the clock distribution chapter (module clocks table) to identify the proper clock source. 0b - The CAN engine clock source is the oscillator clock. Under this condition, the oscillator clock frequency must be lower than the bus clock. 1b - The CAN engine clock source is the peripheral clock.
12 LPB	Loop Back Mode This bit configures FlexCAN to operate in Loop-Back mode. In this mode, FlexCAN performs an internal loop back that can be used for self-test operation. The bit stream output of the transmitter is fed back internally to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic 1). FlexCAN behaves as it normally does when transmitting, and treats its own transmitted message as a message received from a remote node. In this mode, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field, generating an internal acknowledge bit to ensure proper reception of its own message. Both transmit and receive interrupts are generated. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. **NOTE:** In this mode, MCR[SRXDIS] cannot be asserted because this will impede the self-reception of a transmitted message. **NOTE:** FDCTRL[TDCEN] must be disabled when LPB is asserted. 0b - Loop Back disabled. 1b - Loop Back enabled.
11 TWRNMSK	Tx Warning Interrupt Mask This bit provides a mask for the Tx Warning interrupt associated with the TWRNINT flag in the Error and Status Register 1 (ESR1). This bit is read as zero when MCR[WRNEN] is negated. This bit can be written only if MCR[WRNEN] is asserted. 0b - Tx Warning interrupt disabled. 1b - Tx Warning interrupt enabled.
10 RWRNMSK	Rx Warning Interrupt Mask

Table continues on the next page...

Field	Function
	This bit provides a mask for the Rx Warning interrupt associated with the RWRNINT flag in the Error and Status Register 1 (ESR1). This bit is read as zero when MCR[WRNEN] bit is negated. This bit can be written only if MCR[WRNEN] bit is asserted. 0b - Rx Warning interrupt disabled. 1b - Rx Warning interrupt enabled.
9 —	Reserved
8 —	Reserved
7 SMP	CAN Bit Sampling This bit defines the sampling mode of CAN bits at the Rx input. It can be written in Freeze mode only, because it is blocked by hardware in other modes. **NOTE:** For proper operation, to assert SMP it is necessary to guarantee a minimum value of two TQs in CTRL1[PSEG1] (or CBT[EPSEG1]).This bit cannot be asserted when CAN FD is enabled (MCR[FDEN] = 1). 0b - Just one sample is used to determine the bit value. 1b - Three samples are used to determine the value of the received bit: the regular one (sample point) and two preceding samples; a majority rule is used.
6 BOFFREC	Bus Off Recovery This bit defines how FlexCAN recovers from Bus Off state. If this bit is negated, automatic recovering from Bus Off state occurs according to the CAN Specification 2.0B. If the bit is asserted, automatic recovering from Bus Off is disabled and the module remains in Bus Off state until the bit is negated by the user. If the negation occurs before 128 sequences of 11 recessive bits are detected on the CAN bus, then Bus Off recovery happens as if the BOFFREC bit had never been asserted. If the negation occurs after 128 sequences of 11 recessive bits occurred, then FlexCAN will resynchronize to the bus by waiting for 11 recessive bits before joining the bus. After negation, the BOFFREC bit can be reasserted again during Bus Off, but it will be effective only the next time the module enters Bus Off. If BOFFREC was negated when the module entered Bus Off, asserting it during Bus Off will not be effective for the current Bus Off recovery. **NOTE:** See Bus off in the CAN Protocol standard (ISO 11898-1) for details. 0b - Automatic recovering from Bus Off state enabled. 1b - Automatic recovering from Bus Off state disabled.
5 TSYN	Timer Sync This bit enables a mechanism that resets the free running timer each time a message is received in message buffer 0. This feature provides means to synchronize multiple FlexCAN stations with a special “SYNC” message, that is, global network time. If MCR[RFEN] is set (Rx FIFO enabled), the first available mailbox, according to CTRL2[RFFN] setting, is used for timer synchronization instead of MB0. This bit can be written in Freeze mode only because it is blocked by hardware in other modes. 0b - Timer sync feature disabled 1b - Timer sync feature enabled
4 LBUF	Lowest Buffer Transmitted First This bit defines the ordering mechanism for message buffer transmission. When asserted, MCR[LPRIOEN] does not affect the priority arbitration. This bit can be written in Freeze mode only, because it is blocked by hardware in other modes. 0b - Buffer with highest priority is transmitted first. 1b - Lowest number buffer is transmitted first.
3 LOM	Listen-Only Mode

Table continues on the next page...

Field	Function
	This bit configures FlexCAN to operate in Listen-Only mode. In this mode, transmission is disabled, all error counters described in the ECR register are frozen, and the module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station will be received. If FlexCAN detects a message that has not been acknowledged, it will flag a BIT0 error without changing the receive error counter (ECR[RXERRCNT]), as if it was trying to acknowledge the message. Listen-Only mode is acknowledged by the state of ESR1[FLTCONF] indicating Passive Error. There can be some delay between the Listen-Only mode request and acknowledge. This bit can be written in Freeze mode only because it is blocked by hardware in other modes. 0b - Listen-Only mode is deactivated. 1b - FlexCAN module operates in Listen-Only mode.
2-0 PROPSEG	Propagation Segment This 3-bit field defines the length of the propagation segment in the bit time. The valid programmable values are 0–7. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Propagation segment time = (PROPSEG + 1) × time-quanta. Time-quantum = one Sclock period.

11.8.5.2.4 Free Running Timer (TIMER)

11.8.5.2.4.1 Offset

Register	Offset
TIMER	8h

11.8.5.2.4.2 Function

This register represents a 16-bit free-running counter that can be read and written by the CPU. The timer starts from 0h after Reset, counts linearly to FFFFh, and wraps around.

The timer is incremented by the CAN bit clock, which defines the baud rate on the CAN bus. During a message transmission/reception, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it counts using the previously programmed baud rate. The timer is not incremented during Disable, Doze, Stop and Freeze modes.

The timer value is captured when the second bit of the identifier field of any frame is on the CAN bus. This captured value is written into the time stamp entry in a message buffer after a successful reception or transmission of a message.

If CTRL1[TSYN] is asserted, the timer is reset whenever a message is received in the first available mailbox, according to CTRL2[RFFN] setting.

The CPU can write to this register anytime. However, if the write occurs at the same time that the timer is being reset by a reception in the first mailbox, then the write value is discarded.

Reading this register affects the mailbox unlocking procedure (see Mailbox lock mechanism).

11.8.5.2.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TIMER															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.4.4 Fields

Field	Function
31-16 —	Reserved
15-0 TIMER	Timer Value Contains the free-running counter value.

11.8.5.2.5 Rx Mailboxes Global Mask Register (RXMGMASK)

11.8.5.2.5.1 Offset

Register	Offset
RXMGMASK	10h

11.8.5.2.5.2 Function

This register is located in RAM.

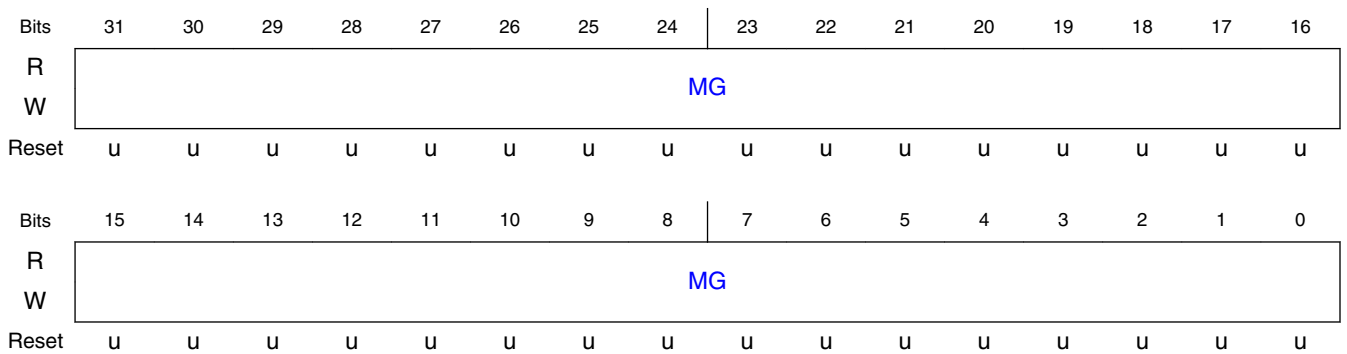
RXMGMASK is provided for legacy application support.

- When MCR[IRMQ] is negated, RXMGMASK is always in effect (the bits in the MG field will mask the mailbox filter bits).
- When MCR[IRMQ] is asserted, RXMGMASK has no effect (the bits in the MG field will not mask the mailbox filter bits).

RXMGMASK is used to mask the filter fields of all Rx MBs, excluding MBs 14-15, which have individual mask registers.

This register can only be written in Freeze mode as it is blocked by hardware in other modes.

11.8.5.2.5.3 Diagram



11.8.5.2.5.4 Fields

Field	Function
31-0 MG	Rx Mailboxes Global Mask Bits These bits mask the mailbox filter bits. Note that the alignment with the ID word of the mailbox is not perfect as the two most significant MG bits affect the fields RTR and IDE, which are located in the Control and Status word of the mailbox. The following table shows in detail which MG bits mask each mailbox filter field.

SMB[RTR] #d31442e2467a1310	CTRL2[RRS]	CTRL2[EACEN]	Mailbox filter fields			
			MB[RTR]	MB[IDE]	MB[ID]	Reserved
0	—	0	note #d31442e2495a1310	note #d31442e2500a1310	MG[28:0]	MG[31:29]
0	—	1	MG[31]	MG[30]	MG[28:0]	MG[29]
1	0	—	—	—	—	MG[31:0]
1	1	0	—	—	MG[28:0]	MG[31:29]
1	1	1	MG[31]	MG[30]	MG[28:0]	MG[29]

0b - The corresponding bit in the filter is "don't care."

Field	Function
	1b - The corresponding bit in the filter is checked.

11.8.5.2.6 Rx 14 Mask Register (RX14MASK)

11.8.5.2.6.1 Offset

Register	Offset
RX14MASK	14h

11.8.5.2.6.2 Function

This register is located in RAM.

RX14MASK is provided for legacy application support. When MCR[IRMQ] is asserted, RX14MASK has no effect.

RX14MASK is used to mask the filter fields of message buffer 14.

This register can only be programmed when the module is in Freeze mode as it is blocked by hardware in other modes.

11.8.5.2.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RX14M															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX14M															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

11.8.5.2.6.4 Fields

Field	Function
31-0 RX14M	Rx Buffer 14 Mask Bits Each mask bit masks the corresponding mailbox 14 filter field in the same way that RXMGMASK masks other mailboxes' filters. See the description of the RXMGMASK register.

Field	Function
	0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

11.8.5.2.7 Rx 15 Mask Register (RX15MASK)

11.8.5.2.7.1 Offset

Register	Offset
RX15MASK	18h

11.8.5.2.7.2 Function

This register is located in RAM.

RX15MASK is provided for legacy application support. When MCR[IRMQ] is asserted, RX15MASK has no effect.

RX15MASK is used to mask the filter fields of message buffer 15.

This register can be programmed only when the module is in Freeze mode because it is blocked by hardware in other modes.

11.8.5.2.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RX15M															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX15M															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

11.8.5.2.7.4 Fields

Field	Function
31-0 RX15M	Rx Buffer 15 Mask Bits

Field	Function
	Each mask bit masks the corresponding mailbox 15 filter field in the same way that RXMGMASK masks other mailboxes' filters. See the description of the RXMGMASK register. 0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

11.8.5.2.8 Error Counter (ECR)

11.8.5.2.8.1 Offset

Register	Offset
ECR	1Ch

11.8.5.2.8.2 Function

This register has four 8-bit fields reflecting the value of the FlexCAN error counters:

- Transmit Error Counter (TXERRCNT field)
- Receive Error Counter (RXERRCNT field)
- Transmit Error Counter for errors detected in the data phase of CAN FD messages with the BRS bit set (TXERRCNT_FAST field)
- Receive Error Counter for errors detected in the data phase of CAN FD messages with the BRS bit set (RXERRCNT_FAST field)

The TXERRCNT and RXERRCNT counters take into account all errors in both CAN FD and non-FD message formats. TXERRCNT_FAST and RXERRCNT_FAST are dedicated to count only the errors that occur in the data phase of CAN FD frames with the BRS bit set.

The Fault Confinement state (ESR1[FLTCONF]) is updated based on TXERRCNT and RXERRCNT counters only. TXERRCNT and RXERRCNT counters can be written in Freeze mode only. TXERRCNT_FAST and RXERRCNT_FAST counters are read-only except in Freeze mode when the CPU can write value zero. The rules for increasing and decreasing these counters are described in the CAN protocol and are completely implemented in the FlexCAN module.

The following are the basic rules for FlexCAN bus state transitions:

- If the value of TXERRCNT or RXERRCNT becomes greater than or equal to 128, ESR1[FLTCONF] is updated to reflect Error Passive state.

- If the FlexCAN state is Error Passive, and either TXERRCNT or RXERRCNT decrements to a value less than or equal to 127 when the other already satisfies this condition, ESR1[FLTCONF] is updated to reflect Error Active state.
- If the value of TXERRCNT becomes greater than 255, ESR1[FLTCONF] is updated to reflect Bus Off state, and an interrupt may be issued. The value of TXERRCNT is then reset to zero.
- If FlexCAN is in Bus Off state, then TXERRCNT is cascaded together with another internal counter to count the 128th occurrences of 11 consecutive recessive bits on the bus. Hence, TXERRCNT is reset to zero and counts in a manner where the internal counter counts 11 such bits and then wraps around when incrementing the TXERRCNT. When TXERRCNT reaches the value of 128, ESR1[FLTCONF] is updated to be Error Active and both error counters are reset to zero. At any instance of dominant bit following a stream of less than 11 consecutive recessive bits, the internal counter resets itself to zero without affecting the TXERRCNT value. The TXERRCNT_FAST counter is frozen during Bus Off.
- If during system startup only one node is operating, then its TXERRCNT increases in each message it is trying to transmit, as a result of acknowledge errors (indicated by the ACKERR bit in the Error and Status Register). After the transition to Error Passive state, TXERRCNT does not increment anymore by acknowledge errors. Therefore the device never goes to the Bus Off state.
- If RXERRCNT increases to a value greater than 127, it is not incremented further, even if more errors are detected when being a receiver. At the next successful message reception, the counter is set to a value between 119 and 127 to return to Error Active state.
- TXERRCNT_FAST and RXERRCNT_FAST error counter values increment and decrement based on errors detected only in the data phase of CAN FD frames with the BRS bit set, following the same increment and decrement rules as TXERRCNT and RXERRCNT counters. These counters do not wrap around and get stuck at their maximum value (255). They stop counting and keep their values frozen when FlexCAN is in Bus Off state. They are reset when FlexCAN leaves Bus Off state and restart counting after FlexCAN returns to Error Active state.

NOTE

See Fault confinement in the CAN Protocol standard (ISO 11898-1) for details.

11.8.5.2.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXERRCNT_FAST								TXERRCNT_FAST							
W	0								0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXERRCNT								TXERRCNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.8.4 Fields

Field	Function
31-24 RXERRCNT_FAST	Receive Error Counter for fast bits Receive error counter for errors detected in the data phase of received CAN FD messages with the BRS bit set. The RXERRCNT_FAST counter is read-only except in Freeze mode, where the CPU can write an 8-bit zero value only.
23-16 TXERRCNT_FAST	Transmit Error Counter for fast bits Transmit error counter for errors detected in the data phase of transmitted CAN FD messages with the BRS bit set. The TXERRCNT_FAST counter is read-only except in Freeze mode, where the CPU can write an 8-bit zero value only.
15-8 RXERRCNT	Receive Error Counter Receive error counter for all errors detected in received messages. The RXERRCNT counter is read-only except in Freeze mode, where it can be written by the CPU.
7-0 TXERRCNT	Transmit Error Counter Transmit error counter for all errors detected in transmitted messages. The TXERRCNT counter is read-only except in Freeze mode, where it can be written by the CPU.

11.8.5.2.9 Error and Status 1 Register (ESR1)

11.8.5.2.9.1 Offset

Register	Offset
ESR1	20h

11.8.5.2.9.2 Function

This register reports various error conditions detected in the reception and transmission of a CAN frame, some general status of the device, and is the source of some interrupts to the CPU. The reported error conditions are:

NOTE

Reading Host can clear these field.

- Errors detected in CAN frames of any format:
  - BIT1ERR
  - BIT0ERR
  - ACKKERR
  - CRCERR
  - FRMERR
  - STFERR
- Errors detected in the data phase of CAN FD frames with the BRS bit set only:
  - BIT1ERR_FAST
  - BIT0ERR_FAST
  - CRCERR_FAST
  - FRMERR_FAST
  - STFERR_FAST

An error detected in a single CAN frame may be reported by one or more error flags. Also, error reporting is cumulative, in case more error events happen in the next frames when the CPU does not attempt to read this register.

Status bits:
- TXWRN
- RXWRN
- IDLE
- TX
- FLTCONF
- RX
- SYNCH

Interrupt bits:
- BOFFINT
- BOFFDONEINT
- ERRINT
- ERRINT_FAST
- WAKINT
- TWRNINT
- RWRNINT

It is recommended that the CPU use the following procedure when servicing interrupt requests generated by these bits:

1. Read this register to capture all error condition and status bits. This action clears the respective bits that were set since the last read access.
2. Write 1 to clear the interrupt bit that has triggered the interrupt request.
3. Write 1 to clear the ERR_OVR bit, if it is set.

Starting from all error flags cleared, a first error event sets either the ERRINT or the ERRINT_FAST (provided the corresponding mask bit is asserted). If other error events in subsequent frames happen before the CPU serves the interrupt request, the ERR_OVR bit is set to indicate that errors from different frames have accumulated.

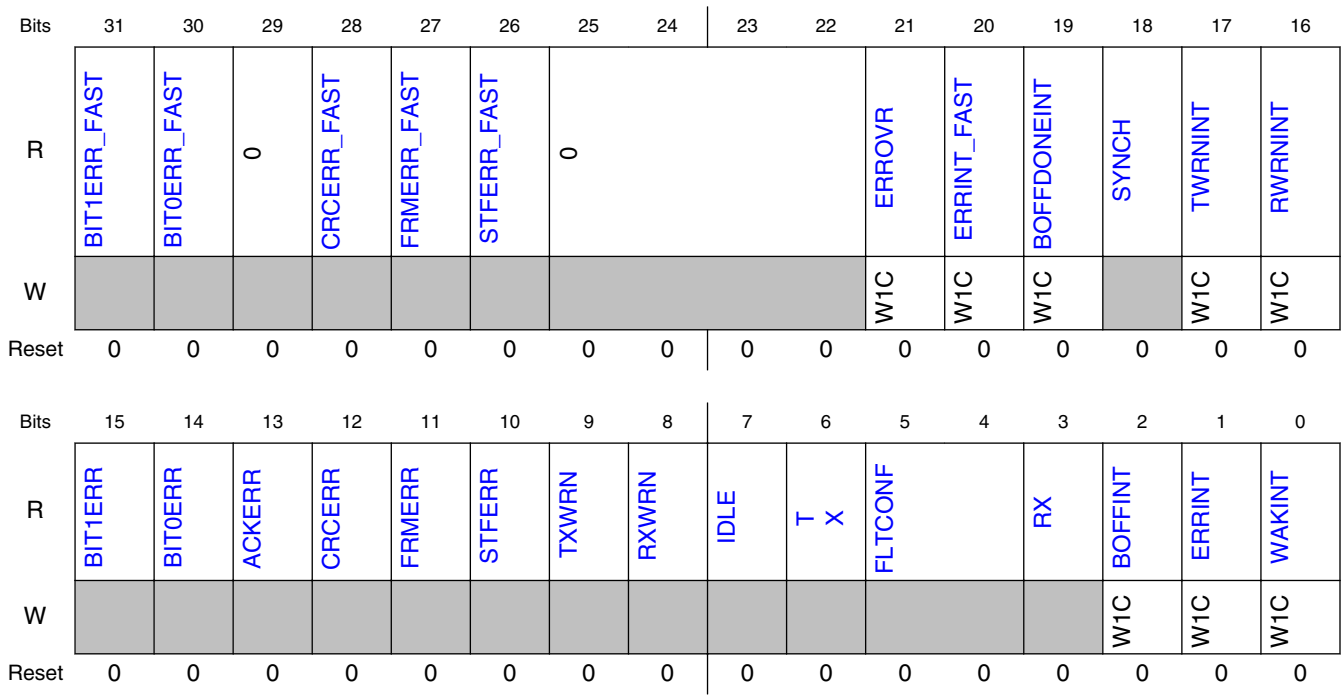
Table 11-178. Can bus status

SYNCH	IDLE	TX	RX	FlexCAN state
0	0	0	0	Not synchronized to CAN bus
1	1	x	x	Idle
1	0	1	0	Transmitting
1	0	0	1	Receiving

NOTE

See Fault confinement in the CAN Protocol standard (ISO 11898-1) for details.

11.8.5.2.9.3 Diagram



11.8.5.2.9.4 Fields

Field	Function
31 BIT1ERR_FAST	Bit1 Error in the Data Phase of CAN FD frames with the BRS bit set This bit indicates when an inconsistency occurs between the transmitted and the received bit in the data phase of CAN FD frames with the BRS bit set. After a read operation, the field's value clears to 0. 0b - No such occurrence. 1b - At least one bit sent as recessive is received as dominant.
30 BIT0ERR_FAST	Bit0 Error in the Data Phase of CAN FD frames with the BRS bit set This bit indicates when an inconsistency occurs between the transmitted and the received bit in the data phase of CAN FD frames with the BRS bit set. After a read operation, the field's value clears to 0. 0b - No such occurrence. 1b - At least one bit sent as dominant is received as recessive.
29 —	Reserved
28 CRCERR_FAST	Cyclic Redundancy Check Error in the CRC field of CAN FD frames with the BRS bit set This bit indicates that a CRC error has been detected by the receiver node in the CRC field of CAN FD frames with the BRS bit set, that is, the calculated CRC is different from the received. After a read operation, the field's value clears to 0. 0b - No such occurrence.

Table continues on the next page...

Field	Function
	1b - A CRC error occurred since last read of this register.
27 FRMERR_FAST	Form Error in the Data Phase of CAN FD frames with the BRS bit set This bit indicates that a form error has been detected by the receiver node in the data phase of CAN FD frames with the BRS bit set, that is, a fixed-form bit field contains at least one illegal bit. After a read operation, the field's value clears to 0. 0b - No such occurrence. 1b - A form error occurred since last read of this register.
26 STFERR_FAST	Stuffing Error in the Data Phase of CAN FD frames with the BRS bit set This bit indicates that a stuffing error has been detected in the data phase of CAN FD frames with the BRS bit set. After a read operation, the field's value clears to 0. 0b - No such occurrence. 1b - A stuffing error occurred since last read of this register.
25-22 —	Reserved
21 ERROVR	Error Overrun This bit indicates that an error condition occurred when any error flag is already set. This bit is cleared by writing it to 1. 0b - Overrun has not occurred. 1b - Overrun has occurred.
20 ERRINT_FAST	Error interrupt for errors detected in Data Phase of CAN FD frames with BRS bit set This bit indicates that at least one of the error bits detected in the data phase of CAN FD frames with the BRS bit set (BIT1ERR_FAST, BIT0ERR_FAST, CRCERR_FAST, FRMERR_FAST, or STFERR_FAST) is set. If the corresponding mask bit CTRL2[ERRMSK_FAST] is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect. 0b - No such occurrence. 1b - Indicates setting of any error bit detected in the data phase of CAN FD frames with the BRS bit set.
19 BOFFDONEINT	Bus Off Done Interrupt This bit is set when the Tx Error Counter (TXERRCNT) has finished counting 128 occurrences of 11 consecutive recessive bits on the CAN bus and is ready to leave Bus Off. If the corresponding mask bit in the Control 2 Register (BOFFDONEMSK) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect. 0b - No such occurrence. 1b - FlexCAN module has completed Bus Off process.
18 SYNCH	CAN Synchronization Status This read-only flag indicates whether the FlexCAN is synchronized to the CAN bus and able to participate in the communication process. It is set and cleared by the FlexCAN. See the table in the overall ESR1 register description. 0b - FlexCAN is not synchronized to the CAN bus. 1b - FlexCAN is synchronized to the CAN bus.
17 TWRNINT	Tx Warning Interrupt Flag If MCR[WRNEN] is asserted, TWRNINT is set when the TXWRN flag transitions from 0 to 1, meaning that the Tx error counter reached 96. If the corresponding mask bit in the Control 1 Register (CTRL1[TWRNMSK]) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When MCR[WRNEN] is negated, this flag is masked. CPU must clear this flag before disabling MCR[WRNEN]. Otherwise it will be set when MCR[WRNEN] is set again. Writing 0 has no effect. This flag is not generated during Bus Off state. This bit is not updated during Freeze mode.

Table continues on the next page...

Field	Function
	0b - No such occurrence. 1b - The Tx error counter transitioned from less than 96 to greater than or equal to 96.
16 RWRNINT	Rx Warning Interrupt Flag If MCR[WRNEN] is asserted, RWRNINT is set when the RXWRN flag transitions from 0 to 1, meaning that the Rx error counters reached 96. If the corresponding mask bit in the Control 1 Register (CTRL1[RWRNMSK]) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When WRNEN is negated, this flag is masked. CPU must clear this flag before disabling the bit. Otherwise it will be set when WRNEN is set again. Writing 0 has no effect. This bit is not updated during Freeze mode. 0b - No such occurrence. 1b - The Rx error counter transitioned from less than 96 to greater than or equal to 96.
15 BIT1ERR	Bit1 Error This bit indicates when an inconsistency occurs between the transmitted and the received bit in a non-CAN FD message or in the arbitration or data phase of a CAN FD message. **NOTE:** This bit is not set by a transmitter in case of arbitration field or ACK slot, or in case of a node sending a passive error flag that detects dominant bits. After a read operation, the field's value clears to 0. 0b - No such occurrence. 1b - At least one bit sent as recessive is received as dominant.
14 BIT0ERR	Bit0 Error This bit indicates when an inconsistency occurs between the transmitted and the received bit in a non-CAN FD message or in the arbitration or data phase of a CAN FD message. After a read operation, the field's value clears to 0. 0b - No such occurrence. 1b - At least one bit sent as dominant is received as recessive.
13 ACKERR	Acknowledge Error This bit indicates that an Acknowledge Error has been detected by the transmitter node, that is, a dominant bit has not been detected during the ACK SLOT. After a read operation, the field's value clears to 0. 0b - No such occurrence. 1b - An ACK error occurred since last read of this register.
12 CRCERR	Cyclic Redundancy Check Error This bit indicates that a Cyclic Redundancy Check (CRC) error has been detected by the receiver node either in a non-FD message or in the arbitration or data phase of a frame in CAN FD format, that is, the calculated CRC is different from the received. After a read operation, the field's value clears to 0. 0b - No such occurrence. 1b - A CRC error occurred since last read of this register.
11 FRMERR	Form Error This bit indicates that a form error has been detected in a non-FD message or else in an FD message's arbitration or data phase by the receiver node, that is, a fixed-form bit field contains at least one illegal bit. After a read operation, the field's value clears to 0. 0b - No such occurrence. 1b - A Form Error occurred since last read of this register.
10	Stuffing Error

Table continues on the next page...

Field	Function
STFERR	This bit indicates that a stuffing error has been detected in a non-FD message or else in an FD message's arbitration or data phase by the receiver node. After a read operation, the field's value clears to 0. 0b - No such occurrence. 1b - A stuffing error occurred since last read of this register.
9 TXWRN	TX Error Warning This bit indicates when repetitive errors are occurring during message transmission and is affected by the value of TXERRCNT in ECR register only. This bit is not updated during Freeze mode. After a read operation, the field's value clears to 0. 0b - No such occurrence. 1b - TXERRCNT is greater than or equal to 96.
8 RXWRN	Rx Error Warning This bit indicates when repetitive errors are occurring during message reception and is affected by the value of RXERRCNT in ECR register only. This bit is not updated during Freeze mode. After a read operation, the field's value clears to 0. 0b - No such occurrence. 1b - RXERRCNT is greater than or equal to 96.
7 IDLE	IDLE This bit indicates when CAN bus is in IDLE state. See the table in the overall ESR1 register description. 0b - No such occurrence. 1b - CAN bus is now IDLE.
6 TX	FlexCAN In Transmission This bit indicates if FlexCAN is transmitting a message. See the table in the overall ESR1 register description. 0b - FlexCAN is not transmitting a message. 1b - FlexCAN is transmitting a message.
5-4 FLTCONF	Fault Confinement State This 2-bit field indicates the confinement state of the FlexCAN module. If CTRL1[LOM] is asserted, after a delay that depends on the CAN bit timing, ESR1[FLTCONF] will indicate Error Passive. The very same delay affects the way that ESR1[FLTCONF] reflects an update to ECR register by the CPU. It may be necessary to wait up to one CAN bit time to get them coherent again. This bit field is affected by soft reset, but if the LOM bit is asserted, its reset value lasts just one CAN bit. After this time, ESR1[FLTCONF] reports Error Passive. 00b - Error Active 01b - Error Passive 1xb - Bus Off
3 RX	FlexCAN In Reception This bit indicates if FlexCAN is receiving a message. See the table in the overall ESR1 register description. 0b - FlexCAN is not receiving a message. 1b - FlexCAN is receiving a message.
2 BOFFINT	Bus Off Interrupt This bit is set when FlexCAN enters Bus Off state. If the corresponding mask bit in the Control Register 1 (CTRL1[BOFFMSK]) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect.

Table continues on the next page...

Field	Function
	0b - No such occurrence. 1b - FlexCAN module entered Bus Off state.
1 ERRINT	Error Interrupt This bit indicates that at least one of the error bits (BIT1ERR, BIT0ERR, ACKERR, CRCERR, FRMERR, or STFERR) is set. If the corresponding mask bit CTRL1[ERRMSK] is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect. 0b - No such occurrence. 1b - Indicates setting of any error bit in the Error and Status register.
0 WAKINT	Wake-Up Interrupt This field applies when FlexCAN is in low-power mode under Self Wake Up mechanism: • Doze mode • Stop mode When a recessive-to-dominant transition is detected on the CAN bus and if MCR[WAKMSK] is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When MCR[SLFWAK] is negated, this flag is masked. The CPU must clear this flag before disabling the bit. Otherwise it will be set when SLFWAK is set again. Writing 0 has no effect. 0b - No such occurrence. 1b - Indicates a recessive to dominant transition was received on the CAN bus.

11.8.5.2.10 Interrupt Masks 2 Register (IMASK2)

11.8.5.2.10.1 Offset

Register	Offset
IMASK2	24h

11.8.5.2.10.2 Function

This register allows any number of a range of the 32 message buffer interrupts to be enabled or disabled for MB63 to MB32. It contains one interrupt mask bit per buffer, enabling the CPU to determine which buffer generates an interrupt after a successful transmission or reception, that is, when the corresponding IFLAG2 bit is set.

11.8.5.2.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BUF63TO32M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF63TO32M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.10.4 Fields

Field	Function
31-0 BUF63TO32M	Buffer MBi Mask Each bit enables or disables the corresponding FlexCAN message buffer interrupt for MB63 to MB32. **NOTE:** Setting or clearing a bit in the IMASK2 Register can assert or negate an interrupt request, if the corresponding IFLAG2 bit is set. 0b - The corresponding buffer interrupt is disabled. 1b - The corresponding buffer interrupt is enabled.

11.8.5.2.11 Interrupt Masks 1 Register (IMASK1)

11.8.5.2.11.1 Offset

Register	Offset
IMASK1	28h

11.8.5.2.11.2 Function

This register allows any number of a range of the 32 message buffer interrupts to be enabled or disabled for MB31 to MB0. It contains one interrupt mask bit per buffer, enabling the CPU to determine which buffer generates an interrupt after a successful transmission or reception, that is, when the corresponding IFLAG1 bit is set.

11.8.5.2.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BUF31TO0M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF31TO0M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.11.4 Fields

Field	Function
31-0 BUF31TO0M	Buffer MBi Mask Each bit enables or disables the corresponding FlexCAN message buffer interrupt for MB31 to MB0. **NOTE:** Setting or clearing a bit in the IMASK1 Register can assert or negate an interrupt request, if the corresponding IFLAG1 bit is set. 0b - The corresponding buffer interrupt is disabled. 1b - The corresponding buffer interrupt is enabled.

11.8.5.2.12 Interrupt Flags 2 Register (IFLAG2)

11.8.5.2.12.1 Offset

Register	Offset
IFLAG2	2Ch

11.8.5.2.12.2 Function

This register defines the flags for the 32 message buffer interrupts for MB63 to MB32. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the respective IFLAG2 bit. If the corresponding IMASK2 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect.

Before updating MCR[MAXMB], CPU must service the IFLAG2 bits whose MB value is greater than the MAXMB to be updated; otherwise, they will remain set and be inconsistent with the number of MBs available.

11.8.5.2.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUF63TO32I															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF63TO32I															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.12.4 Fields

Field	Function
31-0 BUF63TO32I	Buffer MBi Interrupt Each bit flags the corresponding FlexCAN message buffer interrupt for MB63 to MB32. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception. 1b - The corresponding buffer has successfully completed transmission or reception.

11.8.5.2.13 Interrupt Flags 1 Register (IFLAG1)

11.8.5.2.13.1 Offset

Register	Offset
IFLAG1	30h

11.8.5.2.13.2 Function

This register defines the flags for the 32 message buffer interrupts for MB31 to MB0. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG1 bit. If the corresponding IMASK1 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect. There is an exception when DMA for Rx FIFO is enabled, as described below.

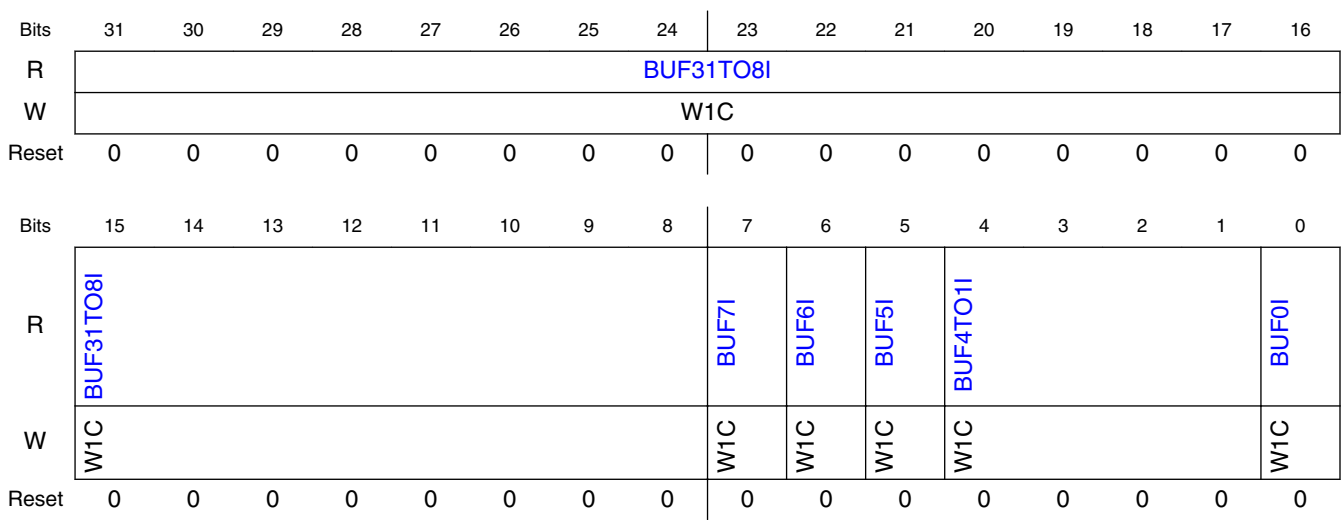
The BUF7I to BUF5I flags are also used to represent FIFO interrupts when the Rx FIFO is enabled. When MCR[RFEN] is set and MCR[DMA] is negated, the function of the eight least significant interrupt flags changes: BUF7I, BUF6I, and BUF5I indicate operating conditions of the FIFO , BUF0I is used to empty FIFO, and BUF4I to BUF1I bits are reserved.

Before enabling MCR[RFEN], the CPU must service the IFLAG bits asserted in the Rx FIFO region; see Rx FIFO. Otherwise, these IFLAG bits will mistakenly show the related MBs now belonging to FIFO as having contents to be serviced. When MCR[RFEN] is negated, the FIFO flags must be cleared. The same care must be taken when a CTRL2[RFFN] value is selected, extending Rx FIFO filters beyond MB7. For example, when RFFN is 8h, the MB0–23 range is occupied by Rx FIFO filters and related IFLAG bits must be cleared.

When both the MCR[RFEN] and MCR[DMA] bits are asserted (DMA feature for Rx FIFO enabled), the function of the eight least significant interrupt flags (BUF7I–BUF0I) are changed to support the DMA operation. BUF7I and BUF6I are not used, as well as BUF4I–BUF1I. BUF5I indicates operating condition of FIFO, and BUF0I is used to empty FIFO. Moreover, BUF5I does not generate a CPU interrupt, but generates a DMA request. IMASK1 bits in Rx FIFO region are not considered when bit MCR[DMA] is enabled. In addition the CPU must not clear the flag BUF5I when DMA is enabled. Before enabling MCR[DMA], the CPU must service the IFLAGs asserted in the Rx FIFO region. When MCR[DMA] is negated, the FIFO must be empty. FIFO must be disabled when MCR[FDEN] is enabled.

Before updating MCR[MAXMB], CPU must service the IFLAG1 bits whose MB value is greater than the MCR[MAXMB] to be updated; otherwise, they will remain set and be inconsistent with the number of MBs available.

11.8.5.2.13.3 Diagram



11.8.5.2.13.4 Fields

Field	Function
31-8 BUF31TO8I	Buffer MBi Interrupt Each bit flags the corresponding FlexCAN message buffer interrupt for MB31 to MB8. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception. 1b - The corresponding buffer has successfully completed transmission or reception.
7 BUF7I	Buffer MB7 Interrupt Or Rx FIFO Overflow When MCR[RFEN] is cleared (Rx FIFO disabled), this bit flags the interrupt for MB7. **NOTE:** This flag is cleared by the FlexCAN whenever MCR[RFEN] is changed by CPU writes. The BUF7I flag represents Rx FIFO overflow when MCR[RFEN] is set. In this case, the flag indicates that a message was lost because the Rx FIFO is full. Note that the flag will not be asserted when the Rx FIFO is full and the message was captured by a mailbox. 0b - No occurrence of MB7 completing transmission/reception when MCR[RFEN]=0, or of Rx FIFO overflow when MCR[RFEN]=1 1b - MB7 completed transmission/reception when MCR[RFEN]=0, or Rx FIFO overflow when MCR[RFEN]=1
6 BUF6I	Buffer MB6 Interrupt Or Rx FIFO Warning When MCR[RFEN] is cleared (Rx FIFO disabled), this bit flags the interrupt for MB6. **NOTE:** This flag is cleared by the FlexCAN whenever MCR[RFEN] is changed by CPU writes. The BUF6I flag represents Rx FIFO warning when MCR[RFEN] is set. In this case, the flag indicates when the number of unread messages within the Rx FIFO is increased to five from four due to the reception of a new one, meaning that the Rx FIFO is almost full. Note that if the flag is cleared when the number of unread messages is greater than four, it does not assert again until the number of unread messages within the Rx FIFO is decreased to be equal to or less than four. 0b - No occurrence of MB6 completing transmission/reception when MCR[RFEN]=0, or of Rx FIFO almost full when MCR[RFEN]=1 1b - MB6 completed transmission/reception when MCR[RFEN]=0, or Rx FIFO almost full when MCR[RFEN]=1
5 BUF5I	Buffer MB5 Interrupt Or Frames available in Rx FIFO When MCR[RFEN] is cleared (Rx FIFO disabled), this field flags the interrupt for MB5. **NOTE:** This flag is cleared by the FlexCAN whenever MCR[RFEN] is changed by CPU writes. When MCR[RFEN] is set (Rx FIFO enabled), the BUF5I flag represents "Frames available in Rx FIFO" and indicates that at least one frame is available to be read from the Rx FIFO. When the MCR[DMA] bit is enabled, this flag generates a DMA request and the CPU must not clear this bit by writing 1 in BUF5I. 0b - No occurrence of MB5 completing transmission/reception when MCR[RFEN]=0, or of frame(s) available in the FIFO, when MCR[RFEN]=1 1b - MB5 completed transmission/reception when MCR[RFEN]=0, or frame(s) available in the Rx FIFO when MCR[RFEN]=1. It generates a DMA request in case of MCR[RFEN] and MCR[DMA] are enabled.
4-1 BUF4TO1I	Buffer MBi Interrupt Or Reserved When MCR[RFEN] is cleared (Rx FIFO disabled), these bits flag the interrupts for MB4 to MB1. **NOTE:** These flags are cleared by the FlexCAN whenever MCR[RFEN] is changed by CPU writes. The BUF4TO1I flags are reserved when MCR[RFEN] is set.

Table continues on the next page...

Field	Function
	0b - The corresponding buffer has no occurrence of successfully completed transmission or reception when MCR[RFEN]=0. 1b - The corresponding buffer has successfully completed transmission or reception when MCR[RFEN]=0.
0 BUF0I	Buffer MB0 Interrupt Or Clear FIFO bit When MCR[RFEN] is cleared (Rx FIFO disabled), this field flags the interrupt for MB0. If the Rx FIFO is enabled, this bit is used to trigger the clear FIFO operation. This operation empties FIFO contents. Before performing this operation the CPU must service all FIFO related IFLAGs. When MCR[DMA] is enabled this operation also clears the BUF5I flag and consequently aborts the DMA request. The clear FIFO operation occurs when the CPU writes 1 in BUF0I. It is only allowed in Freeze mode and is blocked by hardware in other conditions. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception when MCR[RFEN]=0. 1b - The corresponding buffer has successfully completed transmission or reception when MCR[RFEN]=0.

11.8.5.2.14 Control 2 Register (CTRL2)

11.8.5.2.14.1 Offset

Register	Offset
CTRL2	34h

11.8.5.2.14.2 Function

This register complements Control1 register, providing control bits for memory write access in Freeze mode, for extending FIFO filter quantity, and to adjust the operation of internal FlexCAN processes like matching and arbitration.

The contents of this register are not affected by soft reset.

This table shows how the Rx FIFO filter structure is determined by the value of CTRL2[RFFN]. See the CTRL2[RFFN] field description for more information.

Table 11-179. Rx FIFO filter: possible structures

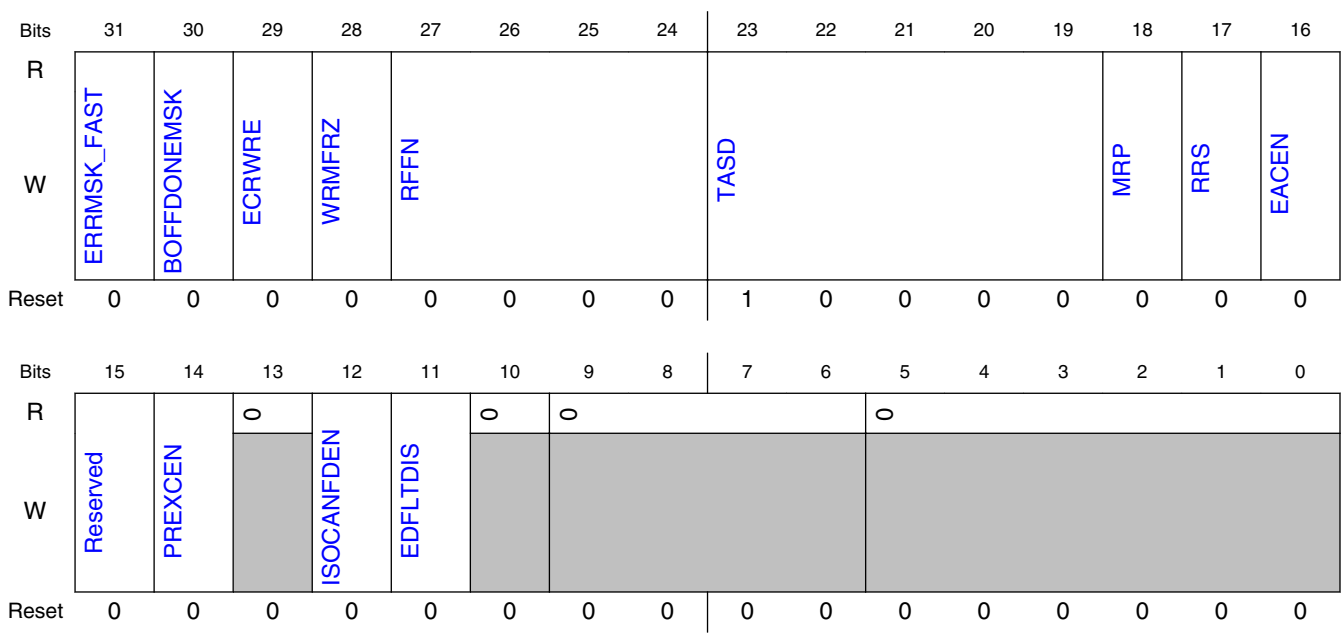
RFFN[3:0]	Number of Rx FIFO filter elements	Message buffers occupied by Rx FIFO and ID filter table	Remaining available mailboxes	Rx FIFO ID filter table elements affected by Rx individual masks	Rx FIFO ID filter table elements affected by Rx FIFO global mask
0h	8	MB 0-7	MB 8-63	Elements 0-7	none
1h	16	MB 0-9	MB 10-63	Elements 0-9	Elements 10-15
2h	24	MB 0-11	MB 12-63	Elements 0-11	Elements 12-23

Table continues on the next page...

Table 11-179. Rx FIFO filter: possible structures (continued)

RFFN[3:0]	Number of Rx FIFO filter elements	Message buffers occupied by Rx FIFO and ID filter table	Remaining available mailboxes	Rx FIFO ID filter table elements affected by Rx individual masks	Rx FIFO ID filter table elements affected by Rx FIFO global mask
3h	32	MB 0-13	MB 14-63	Elements 0-13	Elements 14-31
4h	40	MB 0-15	MB 16-63	Elements 0-15	Elements 16-39
5h	48	MB 0-17	MB 18-63	Elements 0-17	Elements 18-47
6h	56	MB 0-19	MB 20-63	Elements 0-19	Elements 20-55
7h	64	MB 0-21	MB 22-63	Elements 0-21	Elements 22-63
8h	72	MB 0-23	MB 24-63	Elements 0-23	Elements 24-71
9h	80	MB 0-25	MB 26-63	Elements 0-25	Elements 26-79
Ah	88	MB 0-27	MB 28-63	Elements 0-27	Elements 28-87
Bh	96	MB 0-29	MB 30-63	Elements 0-29	Elements 30-95
Ch	104	MB 0-31	MB 32-63	Elements 0-31	Elements 32-103
Dh	112	MB 0-33	MB 34-63	Elements 0-31	Elements 32-111
Eh	120	MB 0-35	MB 36-63	Elements 0-31	Elements 32-119
Fh	128	MB 0-37	MB 38-63	Elements 0-31	Elements 32-127

11.8.5.2.14.3 Diagram



11.8.5.2.14.4 Fields

Field	Function
31 ERRMSK_FAST	Error Interrupt Mask for errors detected in the data phase of fast CAN FD frames This bit provides a mask for the ERRINT_FAST interrupt in ESR1 register. 0b - ERRINT_FAST error interrupt disabled. 1b - ERRINT_FAST error interrupt enabled.
30 BOFFDONEMSK	Bus Off Done Interrupt Mask This bit provides a mask for the bus off done interrupt in ESR1 register. 0b - Bus off done interrupt disabled. 1b - Bus off done interrupt enabled.
29 ECRWRE	Error-correction Configuration Register Write Enable This bit enables the MECR register to be updated. This bit is automatically set to 0 if the protocol described in Detection and correction of memory errors is not followed. 0b - Disable update. 1b - Enable update.
28 WRMFRZ	Write-Access To Memory In Freeze Mode Enable unrestricted write access to FlexCAN memory in Freeze mode. This bit can only be written in Freeze mode and has no effect out of Freeze mode. MCR[RFEN] must not be set during FlexCAN memory initialization. 0b - Maintain the write access restrictions. 1b - Enable unrestricted write access to FlexCAN memory.
27-24 RFFN	Number Of Rx FIFO Filters This 4-bit field defines the number of Rx FIFO filters, as shown in Table 11-179. The maximum selectable number of filters is determined by the chip. This field can only be written in Freeze mode as it is blocked by hardware in other modes. This field must not be programmed with values that cause the number of message buffers occupied by Rx FIFO and ID Filter to exceed the number of mailboxes present, as defined by MCR[MAXMB]. **NOTE:** Each group of eight filters occupies a memory space equivalent to two message buffers, which means that the more filters are implemented the fewer mailboxes will be available. Considering that the Rx FIFO occupies the memory space originally reserved for MB0-5, RFFN should be programmed with a value corresponding to a number of filters not greater than the number of available memory words, which can be calculated as follows: (SETUP_MB - 6) × 4 where SETUP_MB is the smaller of the parameter NUMBER_OF_MB and MCR[MAXMB]. The number of remaining mailboxes available will be: (SETUP_MB - 8) - (RFFN × 2) If the number of Rx FIFO filters programmed through RFFN exceeds the SETUP_MB value (memory space available), then the exceeding ones will not be functional. **NOTE:** • The number of the last remaining available mailboxes is defined by the smaller of NUMBER_OF_MB minus 1 and MCR[MAXMB]. • If Rx Individual Mask registers are not enabled then all Rx FIFO filters are affected by the Rx FIFO Global Mask.
23-19 TASD	Tx Arbitration Start Delay This 5-bit field indicates how many CAN bits the Tx arbitration process start point can be delayed from the first bit of CRC field on CAN bus. See Tx arbitration start delay for more details. This field can be written only in Freeze mode because it is blocked by hardware in other modes.

Table continues on the next page...

Field	Function
18 MRP	Mailboxes Reception Priority If this bit is set the matching process starts from the mailboxes and if no match occurs the matching continues on the Rx FIFO. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Matching starts from Rx FIFO and continues on mailboxes. 1b - Matching starts from mailboxes and continues on Rx FIFO.
17 RRS	Remote Request Storing If this bit is asserted a remote request frame is submitted to a matching process and stored in the corresponding message buffer in the same fashion as a data frame. No automatic remote response frame will be generated. If this bit is negated the remote request frame is submitted to a matching process and an automatic remote response frame is generated if a message buffer with CODE = 1010b is found with the same ID. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Remote response frame is generated. 1b - Remote request frame is stored.
16 EACEN	Entire Frame Arbitration Field Comparison Enable For Rx Mailboxes This bit controls the comparison of IDE and RTR bits within Rx mailbox filters with their corresponding bits in the incoming frame by the matching process. This bit does not affect matching for Rx FIFO. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Rx mailbox filter's IDE bit is always compared and RTR is never compared despite mask bits. 1b - Enables the comparison of both Rx mailbox filter's IDE and RTR bit with their corresponding bits within the incoming frame. Mask bits do apply.
15 —	Reserved When writing to this field, always write the reset value.
14 PREXCEN	Protocol Exception Enable This bit enables the protocol exception feature. This field is writable only in Freeze mode. **NOTE:** See Protocol exception event in the CAN Protocol standard (ISO 11898-1) for details. 0b - Protocol exception is disabled. 1b - Protocol exception is enabled.
13 —	Reserved
12 ISOCANFDEN	ISO CAN FD Enable This field enables the CAN FD protocol according to ISO specification (ISO 11898-1) (see CAN FD ISO compliance). This field is writable only in Freeze mode. **NOTE:** FlexCAN is able to transmit FD frame format according to CAN Protocol standard (ISO 11898-1). 0b - FlexCAN operates using the non-ISO CAN FD protocol. 1b - FlexCAN operates using the ISO CAN FD protocol (ISO 11898-1).
11 EDFLTDIS	Edge Filter Disable This bit disables the edge filter used during the bus integration state.

Table continues on the next page...

Field	Function
	When the Edge Filter is enabled, two consecutive nominal time quanta with dominant bus state are required to detect an edge that causes synchronization. When synchronization occurs, the counting of the sequence of eleven consecutive recessive bits is restarted. The edge filter prevents the dominant pulses that are shorter than a nominal bit time (present during the data phase of an FD frame) from being mistaken for an idle condition. This field is writable only in Freeze mode. **NOTE:** See Bus Integration state in the CAN Protocol standard (ISO 11898-1) for details. 0b - Edge filter is enabled 1b - Edge filter is disabled
10 —	Reserved
9-6 —	Reserved
5-0 —	Reserved

11.8.5.2.15 Error and Status 2 Register (ESR2)

11.8.5.2.15.1 Offset

Register	Offset
ESR2	38h

11.8.5.2.15.2 Function

This register reports some general status information.

11.8.5.2.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0									LPTM						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	VPS	IMB	0												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.15.4 Fields

Field	Function
31-23 —	Reserved
22-16 LPTM	Lowest Priority Tx Mailbox If ESR2[VPS] is asserted, this field indicates the lowest number inactive mailbox (see the ESR2[IMB] bit description). If there is no inactive mailbox then the mailbox indicated depends on the value of CTRL1[LBUF]. If CTRL1[LBUF] is negated, then the mailbox indicated is the one that has the greatest arbitration value (see Highest-priority mailbox first). If CTRL1[LBUF] is asserted then the mailbox indicated is the highest number active Tx mailbox. If a Tx mailbox is being transmitted it is not considered in LPTM calculation. If ESR2[IMB] is not asserted and a frame is transmitted successfully, LPTM is updated with its mailbox number.
15 —	Reserved
14 VPS	Valid Priority Status This bit indicates whether ESR2[IMB] and ESR2[LPTM] contents are currently valid or not. It is asserted upon every complete Tx arbitration process unless the CPU writes to Control and Status word of a mailbox that has already been scanned, that is, it is behind Tx Arbitration Pointer, during the Tx arbitration process. If there is no inactive mailbox and only one Tx mailbox that is being transmitted then VPS is not asserted. This bit is negated upon the start of every Tx arbitration process or upon a write to Control and Status word of any mailbox. **NOTE:** ESR2[VPS] is not affected by any CPU write into Control Status (C/S) of an MB that is blocked by abort mechanism. When MCR[AEN] is asserted, the abort code write in C/S of an MB that is being transmitted (pending abort), or any write attempt into a Tx MB with IFLAG set is blocked. 0b - Contents of IMB and LPTM are invalid. 1b - Contents of IMB and LPTM are valid.
13 IMB	Inactive Mailbox If ESR2[VPS] is asserted, this bit indicates whether there is any inactive mailbox (CODE field is either 1000b or 0b). This bit is asserted in the following cases: • During arbitration, if an ESR2[LPTM] is found and it is inactive. • If ESR2[IMB] is not asserted and a frame is transmitted successfully. This bit is always cleared in start of arbitration (see Arbitration process). **NOTE:** ESR2[LPTM] mechanism has the following behavior: if an MB is successfully transmitted and ESR2[IMB]=0 (no inactive mailbox), then ESR2[VPS] and ESR2[IMB] are asserted and the index related to the MB just transmitted is loaded into ESR2[LPTM]. 0b - If ESR2[VPS] is asserted, the ESR2[LPTM] is not an inactive mailbox. 1b - If ESR2[VPS] is asserted, there is at least one inactive mailbox. LPTM content is the number of the first one.
12-0 —	Reserved

11.8.5.2.16 CRC Register (CRCR)

11.8.5.2.16.1 Offset

Register	Offset
CRCR	44h

11.8.5.2.16.2 Function

This register provides information about the CRC of transmitted messages for non-FD messages. This register only reports the 15 low order bits of CRC calculations for messages in CAN FD format that require either 17 or 21 bits. For CAN FD format frames, the FDCRC register must be used. This register is updated at the same time the Tx interrupt flag is asserted.

NOTE

See CRC sequence calculation in the CAN Protocol standard (ISO 11898-1) for details.

11.8.5.2.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0									MBCRC						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	TXCRC														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.16.4 Fields

Field	Function
31-23 —	Reserved
22-16 MBCRC	CRC Mailbox This field indicates the number of the mailbox corresponding to the value in CRCR[TXCRC] field.
15 —	Reserved
14-0 TXCRC	Transmitted CRC value This field indicates the CRC value of the last transmitted message for non-FD frames. For FD frames, CRC value is reported in FDCRC register.

11.8.5.2.17 Rx FIFO Global Mask Register (RXFGMASK)

11.8.5.2.17.1 Offset

Register	Offset
RXFGMASK	48h

11.8.5.2.17.2 Function

This register is located in RAM.

If Rx FIFO is enabled, RXFGMASK is used to mask the Rx FIFO ID filter table elements that do not have a corresponding RXIMR according to CTRL2[RFFN] field setting.

This register can only be written in Freeze mode as it is blocked by hardware in other modes.

The following table shows how the FGM bits correspond to each IDAF field.

Table 11-180. Correspondence of Rx FIFO global mask bits to IDF fields

Rx FIFO ID filter table elements format (MCR[IDAM])	Identifier acceptance filter fields					
	RTR	IDE	RXIDA	RXIDB [1]	RXIDC [2]	Reserved
A	FGM[31]	FGM[30]	FGM[29:1]	—	—	FGM[0]
B	FGM[31], FGM[15]	FGM[30], FGM[14]	—	FGM[29:16], FGM[13:0]		—
C	—	—		—	FGM[31:24], FGM[23:16], FGM[15:8], FGM[7:0]	

1. If MCR[IDAM] is equivalent to format B, only the fourteen most significant bits of the identifier of the incoming frame are compared with the Rx FIFO filter.
2. If MCR[IDAM] is equivalent to format C, only the eight most significant bits of the identifier of the incoming frame are compared with the Rx FIFO filter.

11.8.5.2.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FGM															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FGM															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

11.8.5.2.17.4 Fields

Field	Function
31-0 FGM	Rx FIFO Global Mask Bits These bits mask the ID filter table elements bits in a perfect alignment. 0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

11.8.5.2.18 Rx FIFO Information Register (RXFIR)

11.8.5.2.18.1 Offset

Register	Offset
RXFIR	4Ch

11.8.5.2.18.2 Function

RXFIR provides information on Rx FIFO.

This register is the port through which the CPU accesses the output of the RXFIR FIFO located in RAM. The RXFIR FIFO is written by the FlexCAN whenever a new message is moved into the Rx FIFO. Also, its output is updated whenever the output of the Rx FIFO is updated with the next message. See Rx FIFO for instructions on reading this register.

NOTE

RXFIR can be written only during memory initialization, due to the error code correction (ECC) feature. In every other case the register is read-only.

11.8.5.2.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							IDHIT								
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

11.8.5.2.18.4 Fields

Field	Function
31-9 —	Reserved
8-0 IDHIT	Identifier Acceptance Filter Hit Indicator This field indicates which Identifier Acceptance filter was hit by the received message that is in the output of the Rx FIFO. If multiple filters match the incoming message ID then the first matching IDAF found (lowest number) by the matching process is indicated. This field is valid only when IFLAG1[BUF5I] is asserted.

11.8.5.2.19 CAN Bit Timing Register (CBT)

11.8.5.2.19.1 Offset

Register	Offset
CBT	50h

11.8.5.2.19.2 Function

This register is an alternative way to store the CAN bit timing variables described in CTRL1 register. EPRESDIV, EPROPSEG, EPSEG1, EPSEG2, and ERJW are extended versions of PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW fields respectively.

The BTF bit selects the use of the timing variables defined in this register.

The contents of this register are not affected by soft reset.

NOTE

The CAN bit variables in CTRL1 and in CBT are stored in the same register.

NOTE

When the CAN FD feature is enabled (MCR[FDEN] is set), always set CBT[BTF].

NOTE

The user must ensure bit time settings and protocol engine tolerance are in compliance with the CAN Protocol standard (ISO 11898-1).

11.8.5.2.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BTF	EPRESDIV										ERJW				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EPROPSEG						EPSEG1					EPSEG2				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.19.4 Fields

Field	Function
31 BTF	Bit Timing Format Enable Enables the use of extended CAN bit timing fields EPRESDIV, EPROPSEG, EPSEG1, EPSEG2. and ERJW replacing the CAN bit timing variables defined in CTRL1 register. This field can be written in Freeze mode only. 0b - Extended bit time definitions disabled. 1b - Extended bit time definitions enabled.

Table continues on the next page...

Field	Function
30-21 EPRESDIV	Extended Prescaler Division Factor This 10-bit field defines the ratio between the PE clock frequency and the serial clock (Sclock) frequency when CBT[BTF] is asserted, otherwise it has no effect. It extends the CTRL1[PRESDIV] value range. The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the PE clock frequency (see Protocol timing). This field can be written only in Freeze mode because it is blocked by hardware in other modes. Sclock frequency = PE clock frequency / (EPRESDIV + 1)
20-16 ERJW	Extended Resync Jump Width This 5-bit field defines the maximum number of time quanta that a bit time can be changed by one re-synchronization when CBT[BTF] bit is asserted, otherwise it has no effect. It extends the CTRL1[RJW] value range. One time quantum is equal to the Sclock period. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Resync Jump Width = ERJW + 1.
15-10 EPROPSEG	Extended Propagation Segment This 6-bit field defines the length of the propagation segment in the bit time when CBT[BTF] bit is asserted, otherwise it has no effect. It extends the CTRL1[PROPSEG] value range. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Propagation Segment Time = (EPROPSEG + 1) × Time-Quanta. Time-Quantum = one Sclock period.
9-5 EPSEG1	Extended Phase Segment 1 This 5-bit field defines the length of phase segment 1 in the bit time when CBT[BTF] bit is asserted, otherwise it has no effect. It extends the CTRL1[PSEG1] value range. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 1 = (EPSEG1 + 1) × Time-Quanta. Time-Quantum = one Sclock period.
4-0 EPSEG2	Extended Phase Segment 2 This 5-bit field defines the length of phase segment 2 in the bit time when CBT[BTF] bit is asserted, otherwise it has no effect. It extends the CTRL1[PSEG2] value range. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 1 = (EPSEG2 + 1) × Time-Quanta. Time-Quantum = one Sclock period.

11.8.5.2.20 Rx Individual Mask Registers (RXIMR0 - RXIMR63)

11.8.5.2.20.1 Offset

For n = 0 to 63:

Register	Offset
RXIMRn	880h + (n × 4h)

11.8.5.2.20.2 Function

The RX Individual Mask registers are used to store the acceptance masks for ID filtering in Rx MBs and the Rx FIFO.

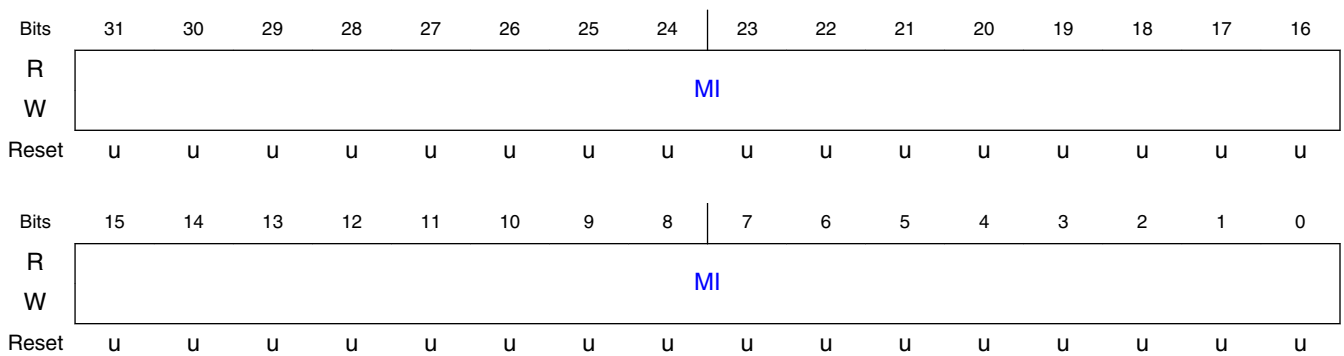
When the Rx FIFO is disabled (MCR[RFEN] bit is negated), an individual mask is provided for each available Rx mailbox on a one-to-one correspondence. When the Rx FIFO is enabled (MCR[RFEN] bit is asserted), an individual mask is provided for each Rx FIFO ID filter table element on a one-to-one correspondence depending on the setting of CTRL2[RFFN] (see Rx FIFO).

RXIMR0 stores the individual mask associated with either MB0 or ID filter table element 0, RXIMR1 stores the individual mask associated with either MB1 or ID filter table element 1, and so on.

RXIMR registers can only be accessed by the CPU when the module is in Freeze mode; otherwise, they are blocked by hardware. These registers are not affected by reset. They are located in RAM and must be explicitly initialized prior to any reception.

It is possible for the RXIMR memory region to be accessed as general purpose memory. See Bus interface for more information.

11.8.5.2.20.3 Diagram



11.8.5.2.20.4 Fields

Field	Function
31-0 MI	Individual Mask Bits Each individual mask bit masks the corresponding bit in both the mailbox filter and Rx FIFO ID filter table element in distinct ways. For mailbox filters, see the RXMGMASK register description. For Rx FIFO ID filter table elements, see the RXFGMASK register description.

Field	Function
	0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

11.8.5.2.21 Memory Error Control Register (MECR)

11.8.5.2.21.1 Offset

Register	Offset
MECR	AE0h

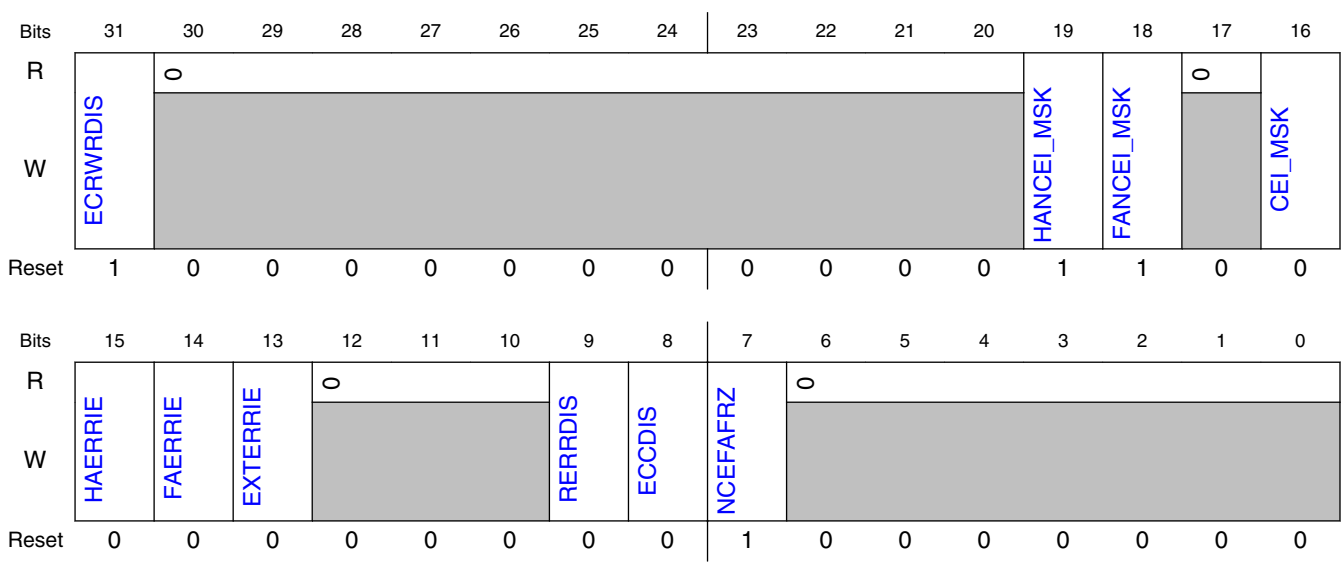
11.8.5.2.21.2 Function

This register contains control bits for memory error detection and correction (ECC).

NOTE

When bit CTRL2[ECRWRE] is 0, writes in this register are blocked, except in the ECRWRDIS bit.

11.8.5.2.21.3 Diagram



11.8.5.2.21.4 Fields

Field	Function
31 ECRWRDIS	Error Configuration Register Write Disable Disables writes on this register. This bit is automatically set to 1 (disabled) when CTRL2[ECRWRE] is enabled. The protocol described in section Detection and correction of memory errors must be followed. 0b - Write is enabled. 1b - Write is disabled.
30-20 —	Reserved
19 HANCEI_MSK	Host Access With Non-Correctable Errors Interrupt Mask Enables the interrupt in case of non-correctable errors detected in memory reads issued by the host (CPU). 0b - Interrupt is disabled. 1b - Interrupt is enabled.
18 FANCEI_MSK	FlexCAN Access With Non-Correctable Errors Interrupt Mask Enables the interrupt in case of non-correctable errors detected in memory reads issued by the FlexCAN internal processes. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
17 —	Reserved
16 CEI_MSK	Correctable Errors Interrupt Mask Enables the interrupt in case of correctable errors detected in memory reads issued by the host or FlexCAN internal processes. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
15 HAERRIE	Host Access Error Injection Enable Enables the injection of errors only in memory reads issued by the host (CPU). 0b - Injection is disabled. 1b - Injection is enabled.
14 FAERRIE	FlexCAN Access Error Injection Enable Enables the injection of errors only in memory reads issued by the FlexCAN internal processes. 0b - Injection is disabled. 1b - Injection is enabled.
13 EXTERRIE	Extended Error Injection Enable Memory accesses performed by internal FlexCAN processes are 64-bit. This bit extends the error injection on 32-bit memory accesses to the complementary 32-bit word using the same 32-bit error injection data and parity words. See Error Injection Data Pattern Register (ERRIDPR) and Error Injection Parity Pattern Register (ERRIPPR). 0b - Error injection is applied only to the 32-bit word. 1b - Error injection is applied to the 64-bit word.
12-10 —	Reserved

Table continues on the next page...

Field	Function
9 RERRDIS	Error Report Disable Disables the update of the error report registers. The update of error-related flags and the generation of bus transfer errors are still active. **NOTE:** When reading the report registers, this bit must be set to assure coherence on the consecutive register reads. 0b - Enable updates of the error report registers. 1b - Disable updates of the error report registers.
8 ECCDIS	Error Correction Disable Disables completely the memory detection and correction mechanism. Besides disabling the error report mechanism, it also stops the update of the error-related flags and generation of bus transfer errors. The parity bits continue being calculated and written into memory on write transactions. 0b - Enable memory error correction. 1b - Disable memory error correction.
7 NCEFAFRZ	Non-Correctable Errors In FlexCAN Access Put Device In Freeze Mode Determines the response when a non-correctable error is detected in a memory read performed by FlexCAN internal processes. In this case, entering Freeze mode prevents corrupted data from being treated as valid by FlexCAN internal processes. 0b - Keep normal operation. 1b - Put FlexCAN in Freeze mode (see section "Freeze mode").
6-0 —	Reserved

11.8.5.2.22 Error Injection Address Register (ERRIAR)

11.8.5.2.22.1 Offset

Register	Offset
ERRIAR	AE4h

11.8.5.2.22.2 Function

This register holds the address where error is to be injected.

Use the following table to convert from the memory map address to the location in the physical FlexCAN RAM (where pairs of values are provided, the first is the address for MCR[FDEN] negated, the second is for MCR[FDEN] asserted):

Table 11-181. Error injection address for Classical CAN format

RAM contents	Injection address	Memory map
FlexCAN registers	Not mapped	—

Table continues on the next page...

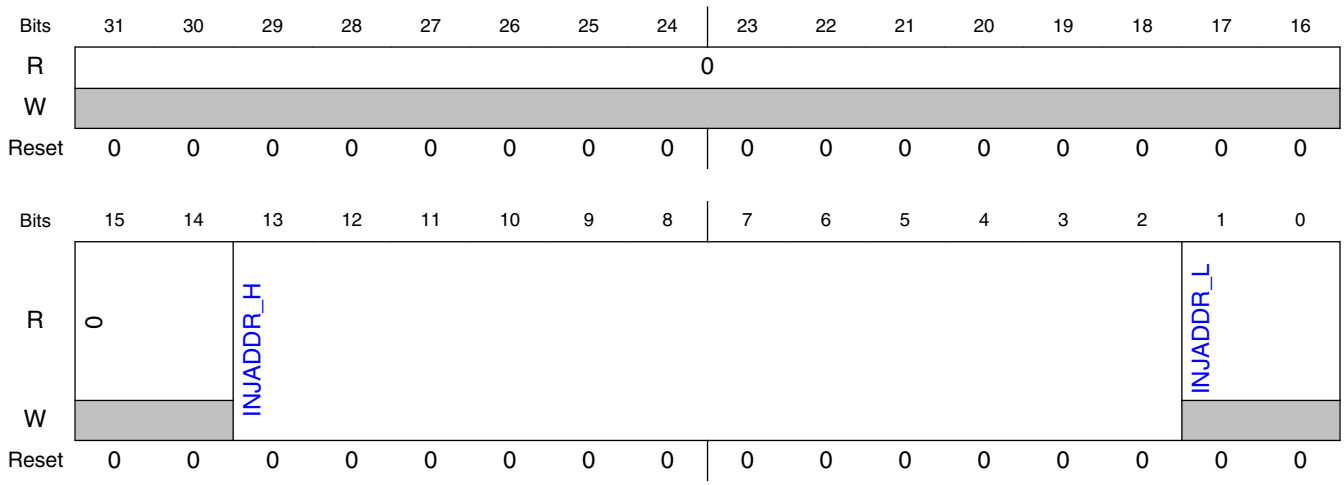
Table 11-181. Error injection address for Classical CAN format (continued)

RAM contents	Injection address	Memory map
MBs	0000h	0080h
Reserved	—	0480h
RXIMRs	0400h	0880h
Reserved	—	0980h
RXFIR_0	0500h	0A80h
RXFIR_1	0504h	0A84h
RXFIR_2	0508h	0A88h
RXFIR_3	050Ch	0A8Ch
RXFIR_4	0510h	0A90h
RXFIR_5	0514h	0A94h
Reserved	—	0A98h
RXMGMASK	0520h	0AA0h
RXFGMASK	0524h	0AA4h
RX14MASK	0528h	0AA8h
RX15MASK	052Ch	0AACh
Tx_SMB	0530h	0AB0h / 0F28h
Rx_SMB0	0540h / 0578h	0AC0h / 0F70h
Rx_SMB1	0550h / 05C0h	0AD0h / 0FB8h
ECC registers	Not mapped	0AE0h
Reserved	—	0B00h
CAN FD registers	Not mapped	0C00h
Reserved	—	0C0Ch

NOTE

For RXFIR addresses, you must inject ECC errors in host access only.

11.8.5.2.22.3 Diagram



11.8.5.2.22.4 Fields

Field	Function
31-14 —	Reserved
13-2 INJADDR_H	Error Injection Address High This read-write field defines the twelve most significant bits of the physical RAM address where error is to be injected (see table above).
1-0 INJADDR_L	Error Injection Address Low This read-only field defines the two least significant bits of the physical RAM address where error is to be injected. They ensure that the address is on a thirty-two-bit boundary.

11.8.5.2.23 Error Injection Data Pattern Register (ERRIDPR)

11.8.5.2.23.1 Offset

Register	Offset
ERRIDPR	AE8h

11.8.5.2.23.2 Function

Holds the error pattern to be injected in the data word read from memory.

11.8.5.2.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DFLIP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DFLIP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.23.4 Fields

Field	Function
31-0 DFLIP	Data flip pattern Bits set to 1 in the flip pattern cause the corresponding data bit in the word read from memory to invert.

11.8.5.2.24 Error Injection Parity Pattern Register (ERRIPPR)

11.8.5.2.24.1 Offset

Register	Offset
ERRIPPR	AECh

11.8.5.2.24.2 Function

Holds the error pattern to be injected in parity bits read from memory along with data word.

Bits set to 1 in the flip pattern cause the corresponding parity bit in the word read from memory to invert.

11.8.5.2.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			PFLIP3					0			PFLIP2				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			PFLIP1					0			PFLIP0				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.24.4 Fields

Field	Function
31-29 —	Reserved
28-24 PFLIP3	Parity Flip Pattern For Byte 3 (most significant)
23-21 —	Reserved
20-16 PFLIP2	Parity Flip Pattern For Byte 2
15-13 —	Reserved
12-8 PFLIP1	Parity Flip Pattern For Byte 1
7-5 —	Reserved
4-0 PFLIP0	Parity Flip Pattern For Byte 0 (Least Significant)

11.8.5.2.25 Error Report Address Register (RERRAR)

11.8.5.2.25.1 Offset

Register	Offset
RERRAR	AF0h

11.8.5.2.25.2 Function

Reports the address used for an access in which an error (correctable or non-correctable) was detected, and also reports the identification of the source of that access.

This address is always reported using a 32-bit alignment. Non-aligned accesses (ERRADDR[1:0] non-0) are reported with the address aligned and data is reported in RERRDR accordingly shifted. In case of errors detected in accesses larger than 32-bit (as performed by FlexCAN internal processes), the address of the 32-bit word which the error was detected is reported. In case of errors detected in more than one 32-bit word, only the least significant address is reported.

Table 11-182. Source of memory access

SAID[2:0]	Error during...
0	Move-out FlexCAN access
1	Move-in
2	Tx Arbitration
3	Rx Matching
4	Move-out Host access
5-7	Reserved

11.8.5.2.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							NCE	0					SAID		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		ERRADDR													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.25.4 Fields

Field	Function
31-25 —	Reserved
24	Non-Correctable Error

Table continues on the next page...

Field	Function
NCE	Indicates that the report is due to a non-correctable error. 0b - Reporting a correctable error 1b - Reporting a non-correctable error
23-19 —	Reserved
18-16 SAID	SAID SAID[2] — Identification of the requester of the memory read request: • 0 = Requested by FlexCAN internal processes • 1 = Requested by host (CPU) SAID[1] — Details of FlexCAN operation: • 0 = Move • 1 = Scanning SAID[0] — Operation that requested the memory read: • 0 = Transmission • 1 = Reception For more information see Table 11-182.
15-14 —	Reserved
13-0 ERRADDR	Address Where Error Detected See the description of the Error Injection Address Register (ERRIAR).

11.8.5.2.26 Error Report Data Register (RERRDR)

11.8.5.2.26.1 Offset

Register	Offset
RERRDR	AF4h

11.8.5.2.26.2 Function

Reports the raw data (unmodified by the correction performed by ECC logic) read from memory with error. The value reported does not represent the transient values of the BUSY bit (see Table 11-186) when reading a message buffer.

11.8.5.2.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.26.4 Fields

Field	Function
31-0 RDATA	Raw data word read from memory with error

11.8.5.2.27 Error Report Syndrome Register (RERRSYNR)

11.8.5.2.27.1 Offset

Register	Offset
RERRSYNR	AF8h

11.8.5.2.27.2 Function

Holds the syndrome detected in a memory read with error. It also reports the bytes which were read in this 32-bit read transaction.

Each SYNDn field indicates the type of error and which bit in byte (n) is affected by the error. (SYND3 corresponds to the most significant byte in the data word read from memory; SYND0 corresponds to the least significant.)

Table 11-183. Syndrome definition

SYNDn (hex)	Type	Bit affected
00	—	none (no error)
01	Code	0

Table continues on the next page...

Table 11-183. Syndrome definition (continued)

SYNDn (hex)	Type	Bit affected
02	Code	1
04	Code	2
07	Data	5
08	Code	3
0E	Data	7
10	Code	4
13	Data	2
15	Data	6
16	Data	1
19	Data	3
1A	Data	4
1C	Data	0
06	—	All-zeros non-correctable error
1F	—	All-ones non-correctable error
All others	—	Non-correctable error

Each BEn field indicates which byte in the 32-bit word reported was effectively read. The syndrome bits are calculated for all bytes, even for the non-read ones. Errors detected in non-read bytes are indicated (see Error indication) and reported (see Error reporting).

11.8.5.2.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BE3	0		SYND3					BE2	0		SYND2				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BE1	0		SYND1					BE0	0		SYND0				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.8.5.2.27.4 Fields

Field	Function
31 BE3	Byte Enabled For Byte 3 (most significant) 0b - The byte was not read. 1b - The byte was read.

Table continues on the next page...

Field	Function
30-29 —	Reserved
28-24 SYND3	Error Syndrome For Byte 3 (most significant) See Table 11-183.
23 BE2	Byte Enabled For Byte 2 0b - The byte was not read. 1b - The byte was read.
22-21 —	Reserved
20-16 SYND2	Error Syndrome For Byte 2 See Table 11-183.
15 BE1	Byte Enabled For Byte 1 0b - The byte was not read. 1b - The byte was read.
14-13 —	Reserved
12-8 SYND1	Error Syndrome for Byte 1 See Table 11-183.
7 BE0	Byte Enabled For Byte 0 (least significant) 0b - The byte was not read. 1b - The byte was read.
6-5 —	Reserved
4-0 SYND0	Error Syndrome For Byte 0 (least significant) See Table 11-183.

11.8.5.2.28 Error Status Register (ERRSR)

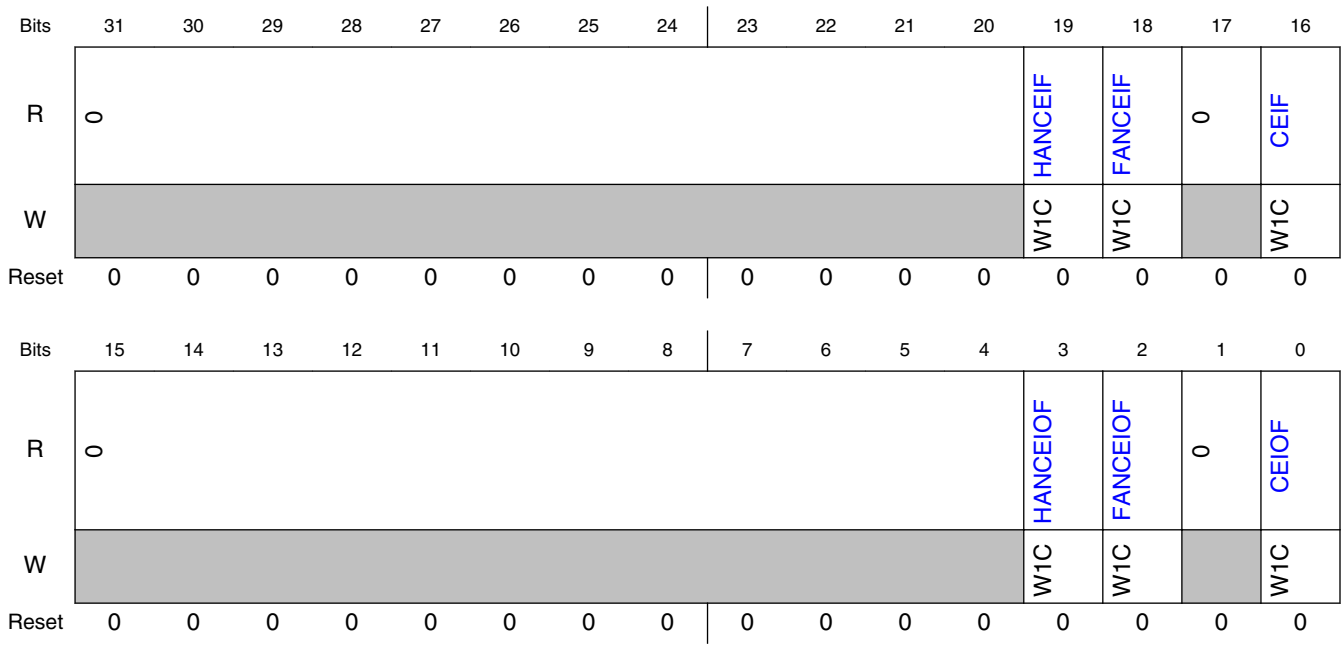
11.8.5.2.28.1 Offset

Register	Offset
ERRSR	AFCh

11.8.5.2.28.2 Function

Holds the status bits of the error correction and detection operations. These flags can be cleared by writing 1 to them. Writing 0 has no effect.

11.8.5.2.28.3 Diagram



11.8.5.2.28.4 Fields

Field	Function
31-20 —	Reserved
19 HANCEIF	Host Access With Non-Correctable Error Interrupt Flag Indicates that a non-correctable error was detected in a memory read initiated by host. A bus transfer error is asserted for that access. If MECR[HANCEI_MSK] is set, the interrupt is asserted. 0b - No non-correctable errors were detected in host accesses so far. 1b - A non-correctable error was detected in a host access.
18 FANCEIF	FlexCAN Access With Non-Correctable Error Interrupt Flag Indicates that a non-correctable error was detected in a memory read initiated by FlexCAN internal processes. If MECR[FANCEI_MSK] is set, the interrupt is asserted. 0b - No non-correctable errors were detected in FlexCAN accesses so far. 1b - A non-correctable error was detected in a FlexCAN access.
17 —	Reserved
16 CEIF	Correctable Error Interrupt Flag Indicates that a correctable error was detected in a memory read. If MECR[CEI_MSK] is set, the interrupt is asserted. 0b - No correctable errors were detected so far. 1b - A correctable error was detected.
15-4 —	Reserved

Table continues on the next page...

Field	Function
3 HANCEIOF	Host Access With Non-Correctable Error Interrupt Overrun Flag Indicates that a non-correctable error was detected in a memory read initiated by host when HANCEIF was set. No interrupt is associated with this flag. See Error indication. 0b - No overrun on non-correctable errors in host access 1b - Overrun on non-correctable errors in host access
2 FANCEIOF	FlexCAN Access With Non-Correctable Error Interrupt Overrun Flag Indicates that a non-correctable error was detected in a memory read initiated by FlexCAN internal processes when FANCEIF was set. No interrupt is associated to this flag. See Error indication. 0b - No overrun on non-correctable errors in FlexCAN access 1b - Overrun on non-correctable errors in FlexCAN access
1 —	Reserved
0 CEIOF	Correctable Error Interrupt Overrun Flag Indicates that a correctable error was detected in a memory read when CEIF was set. No interrupt is associated to this flag. See Error indication. 0b - No overrun on correctable errors 1b - Overrun on correctable errors

11.8.5.2.29 CAN FD Control Register (FDCTRL)

11.8.5.2.29.1 Offset

Register	Offset
FDCTRL	C00h

11.8.5.2.29.2 Function

This register contains control bits for the CAN FD operation. It also defines the data size of message buffers allocated in different partitions of RAM (memory blocks) as described in the table below.

When an 8-byte payload is selected:

- Block R0 allocates MB0 to MB31.
- Block R1 allocates MB32 to MB63.

When a payload larger than 8 bytes is selected, the maximum number of MBs in a block is limited as described below:

Table 11-184. Number of message buffers

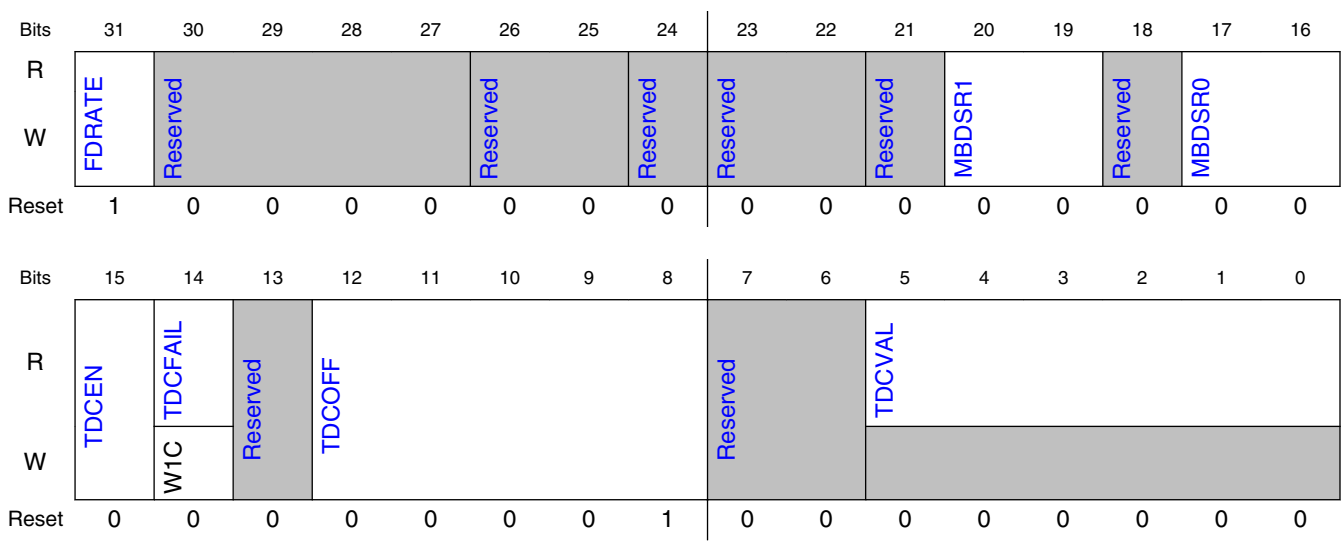
Payload size	Maximum number of message buffers per RAM block
8 bytes	32
16 bytes	21
32 bytes	12
64 bytes	7

NOTE

One memory block fits exactly 32 MBs with an 8-byte payload. For other possible payload sizes, empty memory may exist between the last MB in a block and the beginning of the next block. This empty memory corresponds to less than one MB, and must not be used.

The contents of this register are not affected by soft reset.

11.8.5.2.29.3 Diagram



11.8.5.2.29.4 Fields

Field	Function
31 FDRATE	Bit Rate Switch Enable This bit enables the effect of the Bit Rate Switch (BRS bit) during the data phase of Tx messages. The CPU can write this bit any time. However, its effect turns active only when the CAN bus is in Wait for Bus Idle, Bus Idle, or Bus Off state, or when the current frame under reception or transmission reaches the interframe space.

Table continues on the next page...

Field	Function
	By negating FDCTRL[FDRATE], the CPU can force all bits in CAN FD messages to be transmitted in nominal bit rate, no matter what the value is in the BRS bit of the Tx MBs. 0b - Transmit a frame in nominal rate. The BRS bit in the Tx MB has no effect. 1b - Transmit a frame with bit rate switching if the BRS bit in the Tx MB is recessive.
30-27 —	Reserved
26-25 —	Reserved
24 —	Reserved
23-22 —	Reserved
21 —	Reserved
20-19 MBDSR1	Message Buffer Data Size for Region 1 This two-bit field selects the data size (8, 16, 32, or 64 bytes) for region R1 of message buffers allocated in RAM. It can be written in Freeze mode only. 00b - Selects 8 bytes per message buffer. 01b - Selects 16 bytes per message buffer. 10b - Selects 32 bytes per message buffer. 11b - Selects 64 bytes per message buffer.
18 —	Reserved
17-16 MBDSR0	Message Buffer Data Size for Region 0 This two-bit field selects the data size (8, 16, 32, or 64 bytes) for region R0 of message buffers allocated in RAM. It can be written in Freeze mode only. 00b - Selects 8 bytes per message buffer. 01b - Selects 16 bytes per message buffer. 10b - Selects 32 bytes per message buffer. 11b - Selects 64 bytes per message buffer.
15 TDCEN	Transceiver Delay Compensation Enable This bit can be used to enable and disable the TDC feature. It can be written in Freeze mode only. **NOTE:** See Transmitter delay compensation in the CAN Protocol standard (ISO 11898-1) for details. **NOTE:** TDC must be disabled when Loop Back mode is enabled (see CTRL1[LPB]). 0b - TDC is disabled 1b - TDC is enabled
14 TDCFAIL	Transceiver Delay Compensation Fail This bit indicates when the Transceiver Delay Compensation (TDC) mechanism is out of range, unable to compensate the transceiver's loop delay and successfully compare the delayed received bits to the transmitted ones (see Transceiver delay compensation). TDCFAIL sets in the first time FlexCAN detects the out of range condition. The CPU needs to write one to clear it. 0b - Measured loop delay is in range. 1b - Measured loop delay is out of range.

Table continues on the next page...

Field	Function
13 —	Reserved
12-8 TDCOFF	Transceiver Delay Compensation Offset This bit field contains the offset value to be added to the measured transceiver's loop delay in order to define the position of the delayed comparison point when bit rate switching is active. See Transceiver delay compensation for more details on how the loop delay measurement is performed. TDCOFF can be written in Freeze mode only. Its value can be defined in Protocol Engine (PE) Clock periods (CANCLK, see Protocol timing for more details), and must be selected to be smaller than the CAN bit duration in the data bit rate for proper operation. **NOTE:** It is not recommended to have TDCOFF equal to zero.
7-6 —	Reserved
5-0 TDCVAL	Transceiver Delay Compensation Value This register contains the value of the transceiver loop delay measured from the transmitted EDL to R0 transition edge to the respective received one added to the TDCOFF value specified in the FDCTRL register. This value is an integer multiple of the Protocol Engine (PE) Clock period (CANCLK). See Protocol timing for more details on how the loop delay measurement is performed.

11.8.5.2.30 CAN FD Bit Timing Register (FDCBT)

11.8.5.2.30.1 Offset

Register	Offset
FDCBT	C04h

11.8.5.2.30.2 Function

This register stores the CAN bit timing variables used in the data phase of CAN FD messages when the FDCTRL[FDRATE] is set, compatible with CAN FD specification. FPRESDIV, FPROPSEG, FPSEG1, FPSEG2, and FRJW are used to define the time quantum duration, the number of time quanta per CAN bit, and the sample point position for the data bit rate portion of a CAN FD message with the BRS bit set.

The contents of this register are not affected by soft reset.

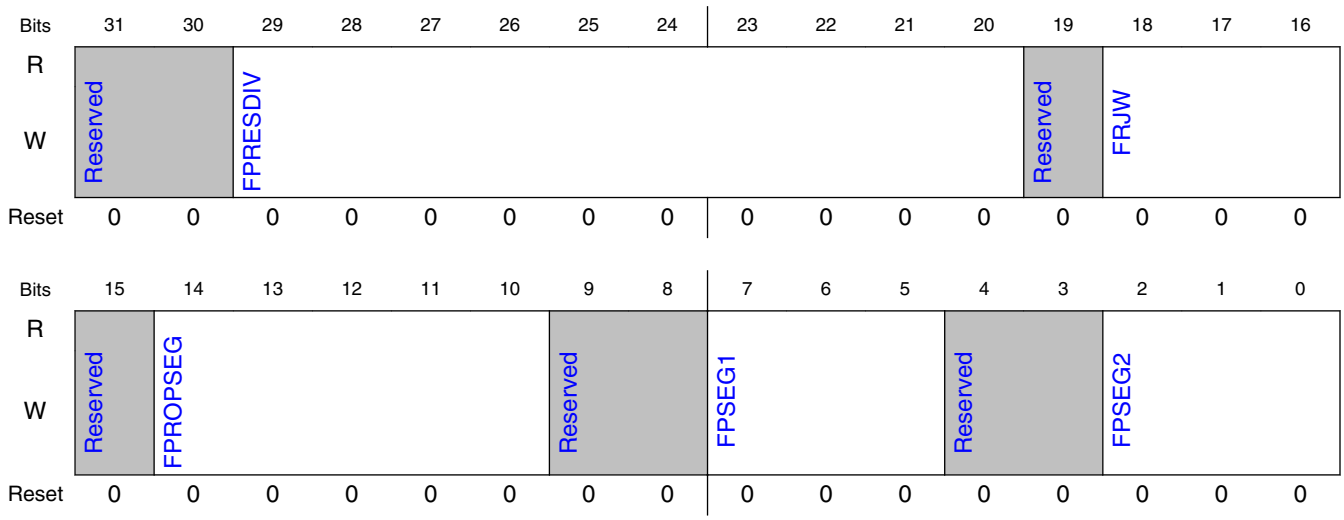
NOTE

The sum of the Fast Propagation Segment (FPROPSEG) and Fast Phase Segment 1 (FPSEG1) must be at least two time quanta.

NOTE

The user must ensure bit time settings and protocol engine tolerance are in compliance with the CAN Protocol standard (ISO 11898-1).

11.8.5.2.30.3 Diagram



11.8.5.2.30.4 Fields

Field	Function
31-30 —	Reserved
29-20 FPRESDIV	Fast Prescaler Division Factor This 10-bit field defines the ratio between the PE clock frequency and the serial clock (Sclock) frequency in the data bit rate portion of a CAN FD message with the BRS bit set. The Sclock period defines the time quantum of the CAN FD protocol for the data bit rate. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Sclock frequency = PE clock frequency / (FPRESDIV + 1). **NOTE:** To minimize errors when processing FD frames, use the same value for FPRESDIV and PRESDIV (in CBT or CTRL1). For more details see the first NOTE in section CAN FD frames.
19 —	Reserved
18-16 FRJW	Fast Resync Jump Width This 3-bit field defines the maximum number of time quanta that a bit time can be changed by one resynchronization in the data bit rate portion of a CAN FD message with the BRS bit set. One time quantum is equal to the Sclock period. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Resync Jump Width = FSJW + 1.

Table continues on the next page...

Field	Function
15 —	Reserved
14-10 FPROPSEG	Fast Propagation Segment This 5-bit field defines the length of the propagation segment in the bit time in the data bit rate portion of a CAN FD message with the BRS bit set. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Propagation Segment Time = FPROPSEG × Time-Quanta. Time-Quantum = one Sclock period.
9-8 —	Reserved
7-5 FPSEG1	Fast Phase Segment 1 This 3-bit field defines the length of phase segment 1 in the bit time in the data bit rate portion of a CAN FD message with the BRS bit set. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Segment 1 = (FPSEG1 + 1) × Time-Quanta. Time-Quantum = one Sclock period.
4-3 —	Reserved
2-0 FPSEG2	Fast Phase Segment 2 This 3-bit field defines the length of phase segment 2 in the data bit rate portion of a CAN FD message with the BRS bit set. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Segment 2 = (FPSEG2 + 1) × Time-Quanta. Time-Quantum = one Sclock period.

11.8.5.2.31 CAN FD CRC Register (FDCRC)

11.8.5.2.31.1 Offset

Register	Offset
FDCRC	C08h

11.8.5.2.31.2 Function

This register provides information about the CRC of transmitted messages.

FlexCAN uses different CRC polynomials for different frame formats, as shown below.

The CRC_15 polynomial is used for all frames in CAN format. The CRC_17 polynomial is used for frames in CAN FD format with a DATA FIELD up to sixteen bytes. The CRC_21 polynomial is used for frames in CAN FD format with a DATA FIELD longer than sixteen bytes. Each polynomial shown below results in a Hamming distance of 6. This register is updated at the same time the Tx Interrupt flag is asserted.

$$CRC_15 = C599h: \ (x^{15} + x^{14} + x^{10} + x^{8} + x^{7} + x^{4} + x^{3} + 1)$$

$$CRC_17 = 3685Bh: \ (x^{17} + x^{16} + x^{14} + x^{13} + x^{11} + x^{6} + x^{4} + x^{3} + x^{1} + 1)$$

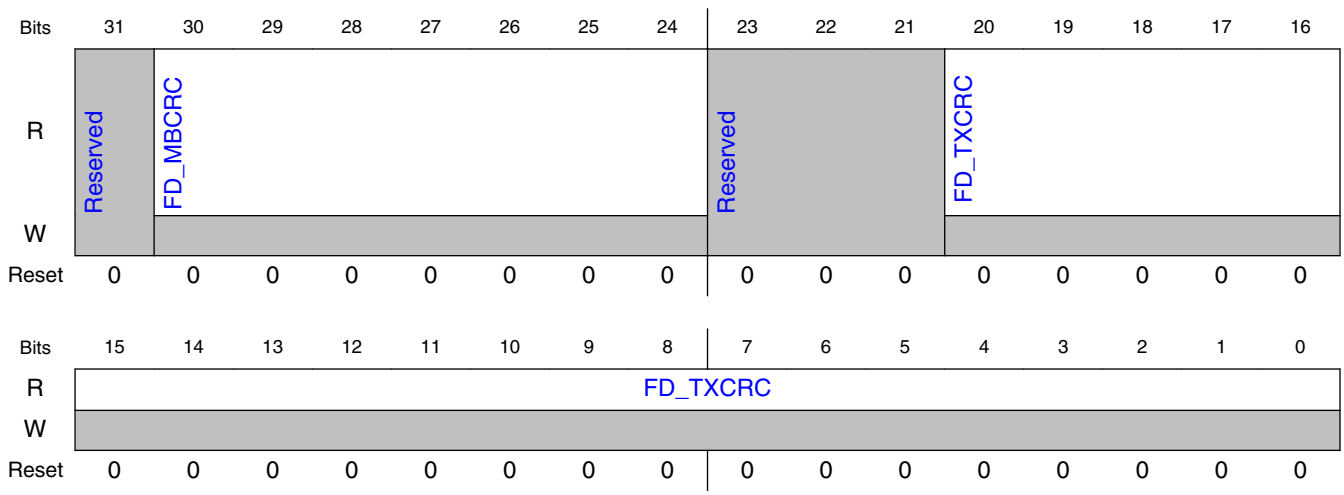
$$CRC_21 = 302899h: \ (x^{21} + x^{20} + x^{13} + x^{11} + x^{7} + x^{4} + x^{3} + 1)$$

Figure 11-143. CRC polynomial used on CAN frame

NOTE

See CRC sequence calculation in the CAN Protocol standard (ISO 11898-1) for details.

11.8.5.2.31.3 Diagram



11.8.5.2.31.4 Fields

Field	Function
31 —	Reserved
30-24 FD_MBCRC	CRC Mailbox Number for FD_TXCRC This field indicates the number of the mailbox corresponding to the value in the FD_TXCRC field, for both FD and non-FD frames. It reports the same information as in CRCR[MBCRC].
23-21 —	Reserved

Table continues on the next page...

Field	Function
20-0 FD_TXCRC	Extended Transmitted CRC value This 21-bit field contains the CRC value calculated over the most recent transmitted message. Different CRC polynomials are used for different frame formats. A 15-bit polynomial, CRC_15, is used for all frames in CAN format. The second 17-bit polynomial, CRC_17, is used for frames in CAN FD format with a data field up to sixteen bytes long. The third 21-bit polynomial, CRC_21, is used for frames in CAN FD format with a data field longer than sixteen bytes. For CRC_15 and CRC_17, the 6 most significant bits and the 4 most significant bits are reported as zeros, respectively. For CRC_15, this register has the same content as CRC register.

11.8.5.3 Message buffer structure

The message buffer structure used by the FlexCAN module is represented in the following figure. Both extended (29-bit identifier) and standard (11-bit identifier) frames used in the CAN specification (Version 2.0 Part B) are represented. Each individual MB is formed by 16, 24, 40, or 72 bytes, depending on the quantity of data bytes allocated for the message payload: 8, 16, 32, or 64 data bytes, respectively.

The memory area 80h–47Fh is used by the mailboxes. When CAN FD is enabled, the exact address for each MB depends on the size of its payload. See FlexCAN memory partition for CAN FD for more detailed information.

Table 11-185. Message buffer structure — example with 64-byte payload

	31	30	29	28	27	24	23	22	21	20	19	18	17	16	15	8	7	0
0h	EDL	BRS	ESI		CODE			SRR	IDE	RTR	DLC				TIME STAMP			
4h	PRIO			ID (standard/extended)											ID (extended)			
8h	Data byte 0							Data byte 1							Data byte 2		Data byte 3	
Ch	Data byte 4							Data byte 5							Data byte 6		Data byte 7	
10h	Data byte 8							Data byte 9							Data byte 10		Data byte 11	
14h	Data byte 12							Data byte 13							Data byte 14		Data byte 15	
18h	Data byte 16							Data byte 17							Data byte 18		Data byte 19	
1Ch	Data byte 20							Data byte 21							Data byte 22		Data byte 23	
20h	Data byte 24							Data byte 25							Data byte 26		Data byte 27	
24h	Data byte 28							Data byte 29							Data byte 30		Data byte 31	
28h	Data byte 32							Data byte 33							Data byte 34		Data byte 35	
2Ch	Data byte 36							Data byte 37							Data byte 38		Data byte 39	
30h	Data byte 40							Data byte 41							Data byte 42		Data byte 43	
34h	Data byte 44							Data byte 45							Data byte 46		Data byte 47	
38h	Data byte 48							Data byte 49							Data byte 50		Data byte 51	
3Ch	Data byte 52							Data byte 53							Data byte 54		Data byte 55	

Table continues on the next page...

Table 11-185. Message buffer structure — example with 64-byte payload (continued)

40h	Data byte 56	Data byte 57	Data byte 58	Data byte 59
44h	Data byte 60	Data byte 61	Data byte 62	Data byte 63
		= Unimplemented or reserved		

EDL — Extended Data Length

This bit distinguishes between CAN format and CAN FD format frames. The EDL bit must not be set for message buffers configured to RANSWER with code field 1010b (see table below).

BRS — Bit Rate Switch

This bit defines whether the bit rate is switched inside a CAN FD format frame.

ESI — Error State Indicator

This bit indicates if the transmitting node is error active or error passive.

CODE — Message Buffer Code

This 4-bit field can be accessed (read or write) by the CPU and by the FlexCAN module itself, as part of the message buffer matching and arbitration process. The encoding is shown in Table 11-186 and Table 11-187. See Functional description for additional information.

Table 11-186. Message buffer code for Rx buffers

CODE description	Rx code BEFORE receive new frame	SRV[1]	Rx code AFTER successful reception[2]	RRS[3]	Comment
0000b: INACTIVE — MB is not active.	INACTIVE	—	—	—	MB does not participate in the matching process.
0100b: EMPTY — MB is active and empty.	EMPTY	—	FULL	—	When a frame is received successfully (after the Move-in process), the CODE field is automatically updated to FULL.
0010b: FULL — MB is full.	FULL	Yes	FULL	—	The act of reading the C/S word followed by unlocking the MB (SRV) does not make the code return to EMPTY. It remains FULL. If a

Table continues on the next page...

Table 11-186. Message buffer code for Rx buffers (continued)

CODE description	Rx code BEFORE receive new frame	SRV[1]	Rx code AFTER successful reception[2]	RRS[3]	Comment
					new frame is moved to the MB after the MB was serviced, the code still remains FULL. See Matching process for matching details related to FULL code.
		No	OVERRUN	—	If the MB is FULL and a new frame is moved to this MB before the CPU services it, the CODE field is automatically updated to OVERRUN. See Matching process for details about overrun behavior.
0110b: OVERRUN — MB is being overwritten into a full buffer.	OVERRUN	Yes	FULL	—	If the CODE field indicates OVERRUN and CPU has serviced the MB, when a new frame is moved to the MB then the code returns to FULL.
		No	OVERRUN	—	If the CODE field already indicates OVERRUN, and another new frame must be moved, the MB will be overwritten again, and the code will remain OVERRUN. See Matching process for details about overrun behavior.
1010b: RANSWER[4] — A frame was configured to recognize a Remote Request frame and transmit	RANSWER	—	TANSWER(1110b)	0	A Remote Answer was configured to recognize a remote request frame received. After that an MB is set to transmit a response frame. The code is

Table continues on the next page...

Table 11-186. Message buffer code for Rx buffers (continued)

CODE description	Rx code BEFORE receive new frame	SRV[1]	Rx code AFTER successful reception[2]	RRS[3]	Comment
a Response frame in return.[5]					automatically changed to TANSWER (1110b). See Matching process for details. If CTRL2[RRS] is negated, transmit a response frame whenever a remote request frame with the same ID is received.
		—	—	1	This code is ignored during matching and arbitration process. See Matching process for details.
CODE[0]=1: BUSY — FlexCAN is updating the contents of the MB. The CPU must not access the MB.	BUSY[6]	—	FULL	—	Indicates that the MB is being updated. It will be negated automatically and does not interfere with the next CODE.
		—	OVERRUN	—	

1. SRV: Serviced MB. MB was read and unlocked by reading TIMER or other MB.
2. A frame is considered a successful reception after the frame to be moved to MB (move-in process). See Move-in for details.
3. Remote Request Stored bit, see "Control 2 register (CTRL2)" for details.
4. Code 1010b is not considered Tx and an MB with this code should not be aborted.
5. Code 1010b must be used in message buffers configured in CAN FD format, having the EDL bit set.
6. Note that for Tx MBs, the BUSY bit should be ignored upon read, except when AEN bit is set in the MCR register. If this bit is asserted, the corresponding MB does not participate in the matching process.

Table 11-187. Message buffer code for Tx buffers

CODE Description	Tx Code BEFORE tx frame	MB RTR	Tx Code AFTER successful transmission	Comment
1000b: INACTIVE — MB is not active	INACTIVE	—	—	MB does not participate in arbitration process.
1001b: ABORT — MB is aborted	ABORT	—	—	MB does not participate in arbitration process.
1100b: DATA — MB is a Tx data frame (MB RTR must be 0)	DATA	0	INACTIVE	Transmit data frame unconditionally once. After transmission, the MB automatically

Table continues on the next page...

Table 11-187. Message buffer code for Tx buffers (continued)

CODE Description	Tx Code BEFORE tx frame	MB RTR	Tx Code AFTER successful transmission	Comment
				returns to the INACTIVE state.
1100b: REMOTE — MB is a Tx Remote Request frame (MB RTR must be 1)	REMOTE	1	EMPTY	Transmit remote request frame unconditionally once. After transmission, the MB automatically becomes an Rx Empty MB with the same ID.
1110b: TANSWER — MB is a Tx Response frame from an incoming Remote Request frame	TANSWER	—	RANSWER	This is an intermediate code that is automatically written to the MB by the CHI as a result of a match to a remote request frame. The remote response frame will be transmitted unconditionally once, and then the code will automatically return to RANSWER (1010b). The CPU can also write this code with the same effect. The remote response frame can be either a data frame or another remote request frame depending on the RTR bit value. See Matching process and Arbitration process for details.

SRR — Substitute Remote Request

Fixed recessive bit, used only in extended format. It must be set to one by the user for transmission (Tx Buffers) and will be stored with the value received on the CAN bus for Rx receiving buffers. It can be received as either recessive or dominant. If FlexCAN receives this bit as dominant, then it is interpreted as an arbitration loss.

1 = Recessive value is compulsory for transmission in extended format frames

0 = Dominant is not a valid value for transmission in extended format frames

IDE — ID Extended Bit

This field identifies whether the frame format is standard or extended.

1 = Frame format is extended

0 = Frame format is standard

RTR — Remote Transmission Request

This bit affects the behavior of remote frames and is part of the reception filter. See Table 11-186, Table 11-187, and the description of the RRS field in Control 2 register (CTRL2) for additional details.

If FlexCAN transmits this bit as '1' (recessive) and receives it as '0' (dominant), it is interpreted as an arbitration loss. If this bit is transmitted as '0' (dominant), then if it is received as '1' (recessive), the FlexCAN module treats it as a bit error. If the value received matches the value transmitted, it is considered a successful bit transmission.

1 = Indicates the current MB may have a remote request frame to be transmitted if MB is Tx. If the MB is Rx then incoming remote request frames may be stored.

0 = Indicates the current MB has a data frame to be transmitted. In Rx MB it may be considered in matching processes.

NOTE

When configuring CAN FD frames, the RTR bit must be negated.

DLC — Length of Data in Bytes

This 4-bit field is the length (in bytes) of the Rx or Tx data, which is located in offset 8h–Fh of the MB space (see Table 11-185). In reception, this field is written by the FlexCAN module, copied from the DLC (Data Length Code) field of the received frame. In transmission, this field is written by the CPU and corresponds to the DLC field value of the frame to be transmitted. When RTR = 1, the frame to be transmitted is a remote frame and does not include the data field, regardless of the DLC field (see Table 11-188).

TIME STAMP — Free-Running Counter Time Stamp

This 16-bit field is a copy of the Free-Running Timer, captured for Tx and Rx frames at the time when the beginning of the Identifier field appears on the CAN bus.

PRIO — Local priority

This 3-bit field is used only when MCR[LPRIO_EN] is set, and it only makes sense for Tx mailboxes. These bits are not transmitted. They are appended to the regular ID to define the transmission priority. See Arbitration process.

ID — Frame Identifier

In standard frame format, only the 11 most significant bits (28 to 18) are used for frame identification in both receive and transmit cases. The 18 least significant bits are ignored. In extended frame format, all bits are used for frame identification in both receive and transmit cases.

DATA BYTE 0 to 63 — Data Field

Up to sixty four bytes can be used for a data frame, depending on the size of payload selected for the message buffers.

For Rx frames, the data is stored as it is received from the CAN bus. DATA BYTE (*n*) is valid only if *n* is less than DLC as shown in the table below.

Table 11-188. DATA BYTEs validity

DLC	Valid DATA BYTEs
0	none
1	DATA BYTE 0
2	DATA BYTE 0 to 1
3	DATA BYTE 0 to 2
4	DATA BYTE 0 to 3
5	DATA BYTE 0 to 4
6	DATA BYTE 0 to 5
7	DATA BYTE 0 to 6
8	DATA BYTE 0 to 7
9	DATA BYTE 0 to 11
10	DATA BYTE 0 to 15
11	DATA BYTE 0 to 19
12	DATA BYTE 0 to 23
13	DATA BYTE 0 to 31
14	DATA BYTE 0 to 47
15	DATA BYTE 0 to 63

11.8.5.4 FlexCAN memory partition for CAN FD

When CAN FD is enabled, the FlexCAN RAM can be partitioned in blocks of 512 bytes. Each block can accommodate a number of message buffers which depends on the configuration provided by FDCTRL[MBDSRn] bit fields as shown in table below.

Table 11-189. RAM partition

RAM block	Number of MBs with 8 bytes (default range)	Size control bit field in FDCTRL register	Number of MBs of different sizes, per block
0	0 to 31	MBDSR0	MBDSR0=00, 32 MBs with 8 bytes payload MBDSR0=01, 21 MBs with 16 bytes payload MBDSR0=10, 12 MBs with 32 bytes payload MBDSR0=11, 7 MBs with 64 bytes payload
1	32 to 63	MBDSR1	MBDSR1=00, 32 MBs with 8 bytes payload MBDSR1=01, 21 MBs with 16 bytes payload MBDSR1=10, 12 MBs with 32 bytes payload MBDSR1=11, 7 MBs with 64 bytes payload

When payload sizes of 16, 32, or 64 bytes are configured in some or all RAM blocks, the total number of MBs and its respective number order may differ from the default configuration of 8 bytes. For example, suppose Block0 is configured to 8 bytes payload, Block1 to 16 bytes, Then the following table indicates how the message buffers will be arranged in RAM.

Table 11-190. RAM partition example

RAM block	Payload size	Number of MBs in the RAM block	Message buffer range
0	FDCTRL[MBDSR0]=00, 8 bytes payload	32	0 to 31
1	FDCTRL[MBDSR1]=01, 16 bytes payload	21	32 to 52

11.8.5.5 FlexCAN message buffer memory map

The FlexCAN memory buffers are allocated in memory according to the tables below.

Table 11-191. 8-byte message buffers

Address offset (hex)	MBDSR=b00 8-byte payload
0080	MB0
0090	MB1
00A0	MB2
00B0	MB3
00C0	MB4
00D0	MB5
00E0	MB6
00F0	MB7
0100	MB8
0110	MB9
0120	MB10
0130	MB11
0140	MB12
0150	MB13
0160	MB14
0170	MB15
0180	MB16
0190	MB17
01A0	MB18
01B0	MB19
01C0	MB20
01D0	MB21
01E0	MB22
01F0	MB23
0200	MB24
0210	MB25
0220	MB26
0230	MB27
0240	MB28
0250	MB29
0260	MB30
0270	MB31
0280	MB32
0290	MB33
02A0	MB34
02B0	MB35
02C0	MB36
02D0	MB37

Table continues on the next page...

Table 11-191. 8-byte message buffers (continued)

Address offset (hex)	MBDSR=b00 8-byte payload
02E0	MB38
02F0	MB39
0300	MB40
0310	MB41
0320	MB42
0330	MB43
0340	MB44
0350	MB45
0360	MB46
0370	MB47
0380	MB48
0390	MB49
03A0	MB50
03B0	MB51
03C0	MB52
03D0	MB53
03E0	MB54
03F0	MB55
0400	MB56
0410	MB57
0420	MB58
0430	MB59
0440	MB60
0450	MB61
0460	MB62
0470	MB63

Table 11-192. 16-byte message buffers

Address offset (hex)	MBDSR=b01 16-byte payload
0080	MB0
0098	MB1
00B0	MB2
00C8	MB3
00E0	MB4
00F8	MB5
0110	MB6

Table continues on the next page...

Table 11-192. 16-byte message buffers (continued)

Address offset (hex)	MBDSR=b01 16-byte payload
0128	MB7
0140	MB8
0158	MB9
0170	MB10
0188	MB11
01A0	MB12
01B8	MB13
01D0	MB14
01E8	MB15
0200	MB16
0218	MB17
0230	MB18
0248	MB19
0260	MB20
0280	MB21
0298	MB22
02B0	MB23
02C8	MB24
02E0	MB25
02F8	MB26
0310	MB27
0328	MB28
0340	MB29
0358	MB30
0370	MB31
0388	MB32
03A0	MB33
03B8	MB34
03D0	MB35
03E8	MB36
0400	MB37
0418	MB38
0430	MB39
0448	MB40
0460	MB41

Table 11-193. 32-byte message buffers

Address offset (hex)	MBDSR=b10 32-byte payload
0080	MB0
00A8	MB1
00D0	MB2
00F8	MB3
0120	MB4
0148	MB5
0170	MB6
0198	MB7
01C0	MB8
01E8	MB9
0210	MB10
0238	MB11
0280	MB12
02A8	MB13
02D0	MB14
02F8	MB15
0320	MB16
0348	MB17
0370	MB18
0398	MB19
03C0	MB20
03E8	MB21
0410	MB22
0438	MB23

Table 11-194. 64-byte message buffers

Address offset (hex)	MBDSR=b11 64-byte payload
0080	MB0
00C8	MB1
0110	MB2
0158	MB3
01A0	MB4
01E8	MB5
0230	MB6
0280	MB7
02C8	MB8

Table continues on the next page...

Table 11-194. 64-byte message buffers (continued)

Address offset (hex)	MBDSR=b11 64-byte payload
0310	MB9
0358	MB10
03A0	MB11
03E8	MB12
0430	MB13

11.8.5.6 Rx FIFO structure

When MCR[RFEN] is set, the memory area 80h–DCh (which is normally occupied by MBs 0–5) is used by the reception FIFO engine.

The region 80h–8Ch contains the output of the FIFO which must be read by the CPU as a message buffer. This output contains the oldest message that has been received but not yet read. The region 90h–DCh is reserved for internal use of the FIFO engine.

An additional memory area, which starts at E0h and may extend up to 2DCh (normally occupied by MBs 6–37) depending on the CTRL2[RFFN] field setting, contains the ID filter table (configurable from 8 to 128 table elements) that specifies filtering criteria for accepting frames into the FIFO.

Out of reset, the ID filter table flexible memory area defaults to E0h and extends only to FCh, which corresponds to MBs 6 to 7 for RFFN = 0, for backward compatibility with previous versions of FlexCAN.

The following shows the Rx FIFO data structure.

Table 11-195. Rx FIFO structure

	31 28 24	23 22	21	20	19 18 17 16	15 8	7 0
80h	IDHIT	SRR	IDE	RTR	DLC	TIME STAMP	
84h		ID standard			ID extended		
88h	Data byte 0	Data byte 1				Data byte 2	Data byte 3
8Ch	Data byte 4	Data byte 5				Data byte 6	Data byte 7
90h–DCh	Reserved						
E0h	ID filter table element 0						
E4h	ID filter table element 1						
E8h–2D4h	ID filter table elements 2 to 125						
2D8h	ID filter table element 126						

Table continues on the next page...

Table 11-195. Rx FIFO structure (continued)

2DCh	ID filter table element 127
	= Unimplemented or reserved

Each ID filter table element occupies an entire 32-bit word and can be compounded by one, two, or four Identifier Acceptance Filters (IDAF) depending on the MCR[IDAM] field setting. The following figures show the IDAF indexation.

The following table shows the three different formats of the ID table elements. Note that all elements of the table must have the same format. See Rx FIFO for more information.

Table 11-196. ID table structure

Format	31	30	29 24	23 16	15	14	13 8	7 1	0
A	RTR	IDE	RXIDA (standard = 29–19, extended = 29–1)						
B	RTR	IDE	RXIDB_0 (standard = 29–19, extended = 29–16)		RTR	IDE	RXIDB_1 (standard = 13–3, extended = 13–0)		
C	RXIDC_0 (std/ext = 31–24)		RXIDC_1 (std/ext = 23–16)		RXIDC_2 (std/ext = 15–8)		RXIDC_3 (std/ext = 7–0)		
	= Unimplemented or Reserved								

RTR — Remote frame

This bit specifies if remote frames are accepted into the FIFO if they match the target ID.

1 = Remote frames can be accepted and data frames are rejected

0 = Remote frames are rejected and data frames can be accepted

IDE — Extended frame

Specifies whether extended or standard frames are accepted into the FIFO if they match the target ID.

1 = Extended frames can be accepted and standard frames are rejected

0 = Extended frames are rejected and standard frames can be accepted

RXIDA — Rx frame identifier (Format A)

Specifies an ID to be used as acceptance criteria for the FIFO. In the standard frame format, only the 11 most significant bits (29 to 19) are used for frame identification. In the extended frame format, all bits are used.

RXIDB_0, RXIDB_1 — Rx frame identifier (Format B)

Specifies an ID to be used as acceptance criteria for the FIFO. In the standard frame format, the 11 most significant bits (a full standard ID) (29 to 19 and 13 to 3) are used for frame identification. In the extended frame format, all 14 bits of the field are compared to the 14 most significant bits of the received ID.

RXIDC_0, RXIDC_1, RXIDC_2, RXIDC_3 — Rx frame identifier (Format C)

Specifies an ID to be used as acceptance criteria for the FIFO. In both standard and extended frame formats, all 8 bits of the field are compared to the 8 most significant bits of the received ID.

IDHIT — Identifier acceptance filter hit indicator

This 9-bit field indicates which identifier acceptance filter was hit by the received message that is in the output of the Rx FIFO. See Rx FIFO for more information.

11.8.6 Glossary

Active message buffer

A message buffer is *active* if it can participate in the current matching or arbitration process.

Bus interface unit

FlexCAN submodule responsible for the interface to the CPU.

Bus off

See CAN Specification.[6]

CAN

Controller Area Network, a serial communication protocol defined in CAN Specification[6] and ISO International Standard.[7]

CHI

Controller-host interface, a FlexCAN submodule responsible for message buffer matching and arbitration algorithms.

CRC

Cyclic redundancy check

Dominant bit

A dominant bit wins the arbitration on the CAN bus. It is transmitted as 0.

Doze mode

A system low-power mode where the CPU bus remains active and a global Doze mode request is sent to all peripherals asking them to enter low power mode.

6. Controller Area Network – CAN Specification Version 2.0 Part A, Part B, Robert Bosch GmbH, 1991.
7. ISO International Standard – ISO 11898 First Edition 1993 Road Vehicles – Interchange of Digital Information – Controller Area Network (CAN) for high-speed Communication.

Error active

See CAN Specification.[7]

Error frame

See CAN Specification.[7]

Error passive

See CAN Specification.[7]

Form error

See CAN Specification.[7]

Hard reset

A reset coming from an external pin and/or following power-on. It resets everything.

Idle

See CAN Specification.[7]

Information processing time

See CAN Specification.[7]

Intermission

See CAN Specification.[7]

Matching elements

Data used in the matching process, such as ID, filter, mask, etc.

MB

See Message buffer.

Message buffer

Internal FlexCAN data structure containing bytes received from, or to be transmitted to, the CAN line, as well as information about this data.

Overload frame

See CAN Specification.[7]

Phase buffer segment

See CAN Specification.[7]

Recessive bit

A recessive bit loses the arbitration on the CAN bus. It is transmitted as 1.

Sclock

Serial clock. Obtained by dividing the clock feeding the CAN engine (oscillator or bus clock) by a prescaler factor. The Sclock period defines the time quantum for CAN protocol timing.

Soft reset

Global reset typically used by peripherals to reinitialize some of its registers, but not all of them.

Stop mode

A system low-power mode in which all chip clocks are stopped for maximum power savings.

Stuffing error

See CAN Specification.[7]

Time quantum

This is equal to the Sclock period. It is the minimum time period used to compose the CAN protocol bit timing.

Chapter 12
Timers

12.1 General Purpose Timer (GPT)

12.1.1 Overview

The General Purpose Timer (GPT) is a 32-bit up-counter with clock source and frequency scaling options to provide a wide range of timing rates for various system applications. External hardware can send the GPT a "capture event" signal on an input pin to load the current timer value into a register for software to read. This capture signal can be configured to trigger the GPT on a rising or/and falling edge. Also, when the timer matches a preset value, the GPT can generate an interrupt for software and a "compare event" signal on output pins for external hardware.

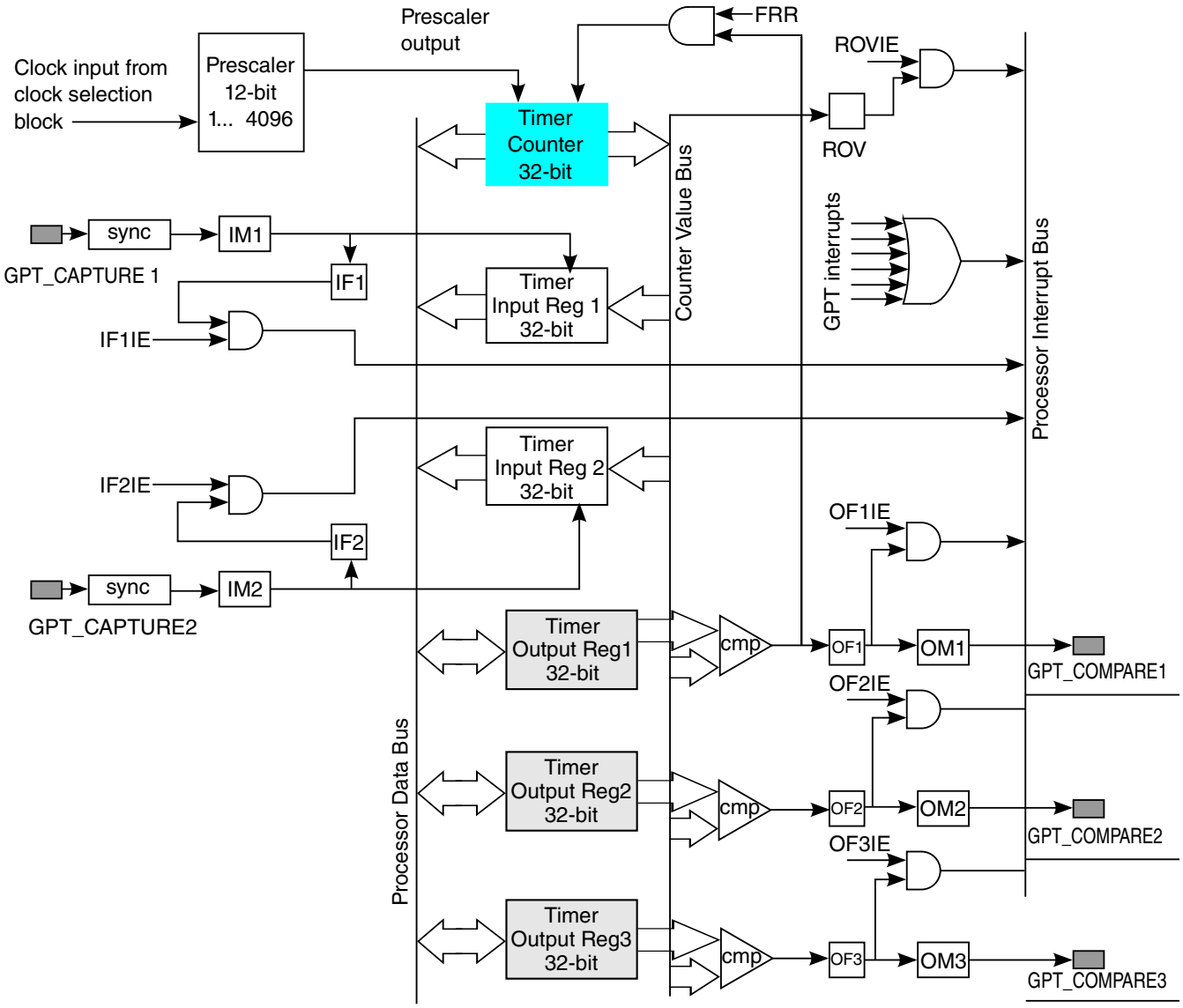


Figure 12-1. GPT Block Diagram

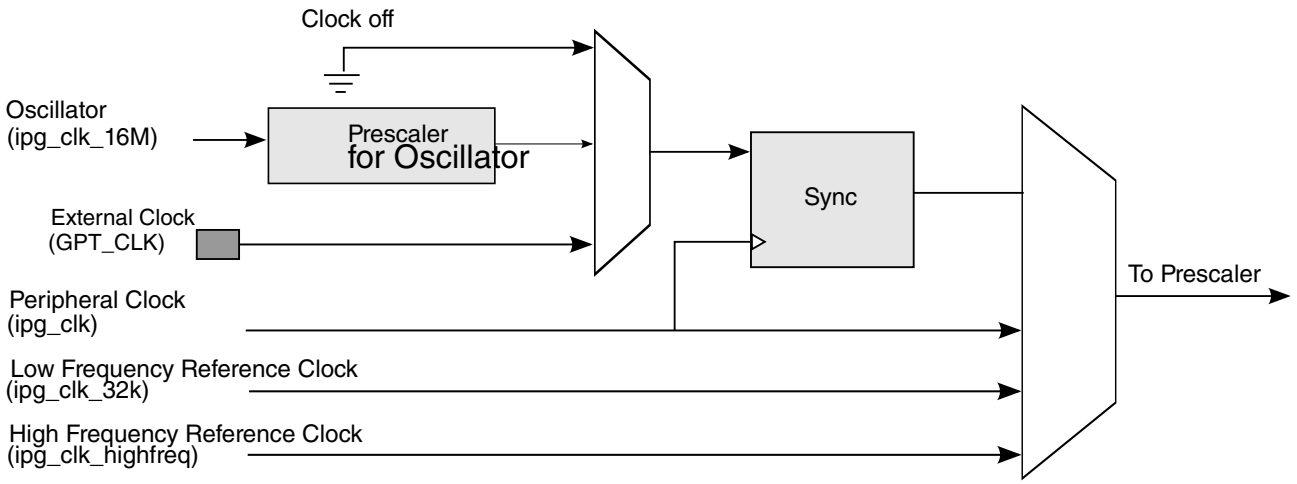


Figure 12-2. GPT Counter Clocks Diagram

12.1.1.1 Features

- One 32-bit up-counter with clock source selection, including external clock.
- Two input capture channels with a programmable trigger edge.
- Three output compare channels with a programmable output mode. A "forced compare" feature is also available.
- Can be programmed to be *active* in low power and debug modes.
- Interrupt generation at capture, compare, and rollover events.
- Restart or free-run modes for counter operations.

12.1.2 Functional Description

This section provides a complete functional description of the GPT.

12.1.2.1 Operating Modes

The GPT counter can be programmed to work in either of two modes: Restart mode or Free-Run mode.

12.1.2.1.1 Restart Mode

In Restart mode (selectable through the GPT Control Register GPT_CR), when the counter reaches the compared value, the counter resets and starts again from 0x00000000. The Restart feature is associated only with Compare Channel 1.

Any write access to the Compare register of Channel 1 will reset the GPT counter. This is done to avoid possibly missing a compare event when compare value is changed from a higher value to lower value while counting is proceeding.

For the other two compare channels, when the compare event occurs the counter is *not reset*.

12.1.2.1.2 Free-Run Mode

In Free-Run mode, when compare events occur for all 3 channels, the counter is *not reset*; instead the counter continues to count until 0xffffffff, and then rolls over (to 0x00000000).

12.1.2.2 Operation

The General Purpose Timer (GPT) has a single counter (GPT_CNT) that is a 32-bit free-running *up-counter,* which starts counting *after it is enabled by software* (EN=1). The counter's clock source is the output of the prescaler labelled "Prescaler output" in Figure 12-1.

- If the GPT timer is disabled (EN=0), then the Main Counter *and* Prescaler Counter freeze their current count values. The ENMOD bit determines the value of the GPT counter when the EN bit is set and the Counter is enabled again.
  - If the ENMOD bit is set (=1), then the Main Counter and Prescaler Counter values are reset to 0, when GPT is enabled (EN=1).
  - If ENMOD bit is programmed to 0, then the Main Counter and Prescaler Counter restart counting from their frozen values, when GPT is enabled again (EN=1).
- If GPT is programmed to be disabled in a low power mode (STOP/WAIT), then the Main Counter and Prescaler Counter freeze at their current count values *when* GPT enters low power mode. When GPT exits a low power mode, the Main Counter and Prescaler Counter start counting from their frozen values *regardless* of the ENMOD bit value. Note that the GPT_CNT can be read *at any time* by the processor, and that *both* Input Capture Channels use the *same* counter (GPT_CNT).
- A hardware reset resets all the GPT registers to their respective reset values. All registers except the Output Compare Registers (OCR1, OCR2, OCR3) obtain a value of 0x0. The Compare registers are reset to 0xFFFF_FFFF.
- The software reset (SWR bit in the GPT_CR control register) resets *all* of the register bits *except* the EN, ENMOD, STOPEN, WAITEN, and DBGEN bits. The state of these bits is not affected by a software reset. Note that a software reset can be given *while the GPT is disabled.*

12.1.2.2.1 Input Capture

There are two Input Capture Channels, and each Input Capture Channel has a dedicated capture pin, capture register and input edge detection/selection logic. Each input capture function has an associated status flag, and can cause the processor to make an interrupt service request.

When a selected edge transition occurs on an Input Capture pin, the contents of the GPT_CNT is captured on the corresponding capture register and the appropriate interrupt status flag is set. An interrupt request can be generated when the transition is detected *if* its corresponding enable bit is set (in the Interrupt Register). The capture can be programmed to occur on the input pin's rising edge, falling edge, on both rising and falling edges, or the capture can be disabled. The events are synchronized with the clock that was selected to run the counter. Only those transitions that occur at least one clock cycle (clock selected to run the counter) *after* the previous recorded transition will be guaranteed to trigger a capture event. There can be up to one clock cycle of uncertainty in the latching of the input transition. The Input Capture registers can be read *at any time* without affecting their values.

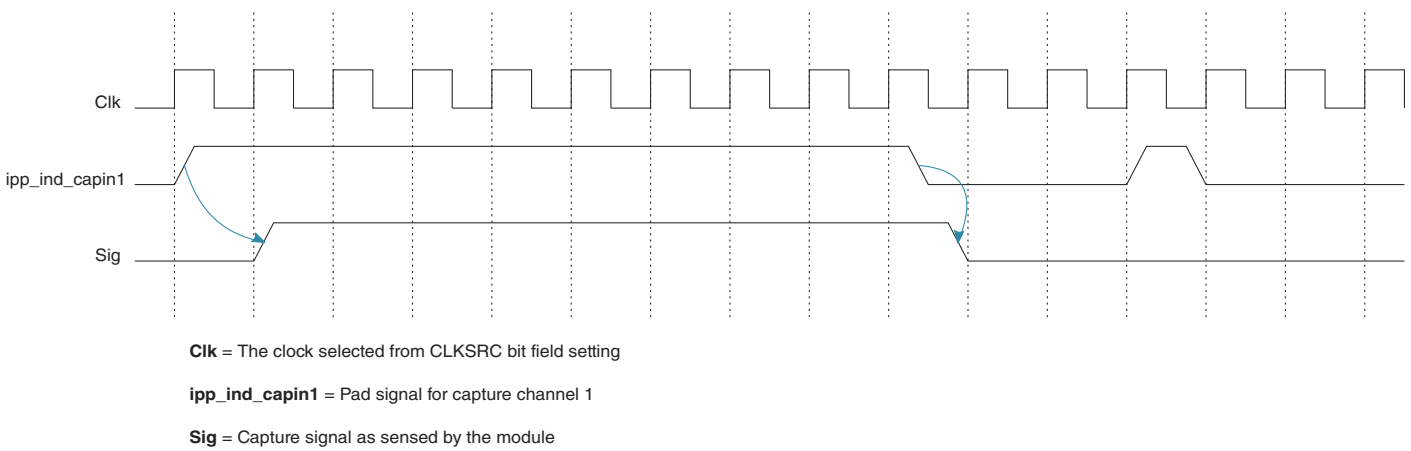


Figure 12-3. Input Capture Event Timing

12.1.2.2.2 Output Compare

The three Output Compare Channels *use the same counter* (GPT_CNT) as the Input Capture Channels. When the programmed content of an Output Compare register matches the value in GPT_CNT, an output compare status flag is set and an interrupt is generated (if the corresponding bit is set in the interrupt register). Consequently, the Output Compare timer pin is set, cleared, toggled, not affected at all or provide an active-low pulse for one input clock period according to the Output Compare Operating Mode bits in the Control Register (CR[OM*n*]).

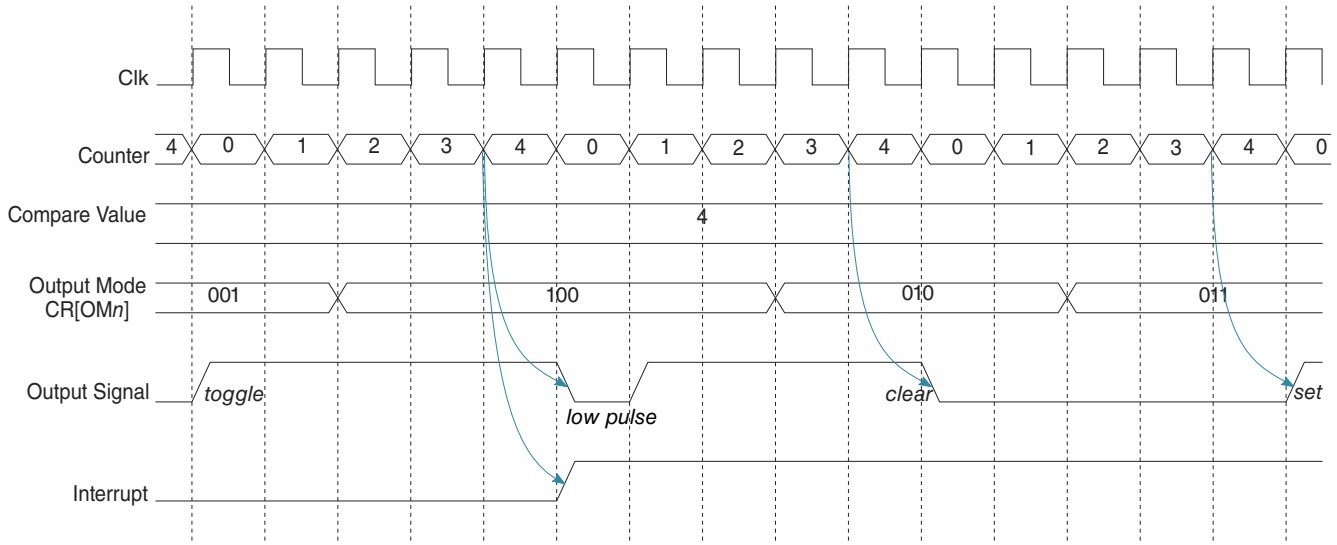


Figure 12-4. Output Compare and Interrupt Timing

There is also a "forced-compare" feature that allows the software to generate a compare event when required, *without the condition of the counter value that is equal to the compare value*. The action taken as a result of a forced compare is the same as when an output compare match occurs, *except that the status flags are not set and no interrupt can be generated*. Forced channels take programmed action immediately after the write to the force-compare bits. These bits are self-negating and always read as zeros.

12.1.2.2.3 Low Power Mode Behavior

In Low Power modes, if the clock from the selected clock source is available (except for the External Clock (GPT_CLK), which can be used *only if* the Peripheral Clock (ipg_clk) is available), the counter will continue to run depending on whether the control bit for that mode is set. If the clock is not present or if the corresponding low power bit in the GPT_CR control register is 0, the Main Counter and the Prescaler Counter freeze at their current values and resume counting (from their frozen values) when the Low Power mode is exited.

12.1.2.2.4 Debug Mode Behavior

In Debug mode, the modules in the device have the option of continuing to run or be halted.

- If the DBGEN bit is set, then the GPT timer will continue to run in Debug mode.
- If the DBGEN bit is not set (in the GPT_CR control register), then the GPT timer is halted.

12.1.2.3 Clocks

The clock that is input to the prescaler can be selected from multiple clock sources. The following table describes the clock sources for GPT. See chip clock control block for clock source connection, configuration and gating information.

Table 12-1. GPT Clocks

Clock name	Description
Peripheral Clock (ipg_clk)	This is the internal clock used to access the module registers. If the Peripheral Clock or the External Clock (GPT_CLK) is selected as the Clock Source (CLKSRC=001 or 011), then the Peripheral Clock will be ON in normal GPT operations. In Low Power modes, if the GPT is programmed to be disabled (STOPEN or WAITEN or DOZEN=0), then the Peripheral Clock (ipg_clk) can be switched OFF.
Low Reference Clock (ipg_clk_32k)	This low-frequency 32-kHz reference clock (provided by the CCM) is intended to be ON in Low Power mode when the Peripheral Clock (ipg_clk) is turned OFF, thereby enabling the GPT to be operated using the Low Reference Clock in Low Power mode.
High-Frequency Clock (ipg_clk_highfreq)	Provided by the Clock Controller Module (CCM), this high-frequency reference clock is intended to be ON in Normal Power mode when the Peripheral Clock (ipg_clk) is turned OFF, thereby enabling the GPT to be operated using the High Frequency Clock *in Normal Power mode*.
Oscillator Clock (ipg_clk_16M)	This 16-MHz oscillator reference clock is intended to be used against frequency changes of the Peripheral Clock (ipg_clk) to provide a more accurate timer clock for system operation. The CCM provides the 16 MHz Oscillator Clock *without* synchronizing it to the System Bus Clock (ahb_clk) in Normal functional mode. Synchronization is done in GPT module. Before synchronization, the 16 MHz Oscillator Clock is divided by a dedicated prescaler to make sure the clock frequency less than half of System Bus Clock (ahb_clk).
External Clock (GPT_CLK)	The External Clock comes from *outside* the device and can be selected to run the GPT counter. The External Clock frequency is limited to < 1/4 frequency of the Peripheral Clock (ipg_clk) for proper GPT operations. Note that in Low Power modes, *if* the Peripheral Clock (ipg_clk) is not available, then the External Clock *cannot be used* to run the counter.

The clock input source is configured using the clock source field (CLKSRC, in the GPT_CR control register). The clock input to the prescaler can be disabled by programming the CLKSRC bits (of the GPT_CR control register) to 000. **The CLKSRC field value should be changed** *only after disabling the GPT* (by setting the EN bit in the GPT_CR to 0).

The PRESCALER field selects the divide ratio of the input clock that drives the main counter. The prescaler can divide the input clock by a value (from 1 to 4096) and can be changed *at any time*. A change in the value of the PRESCALER field *immediately affects* the output clock frequency.

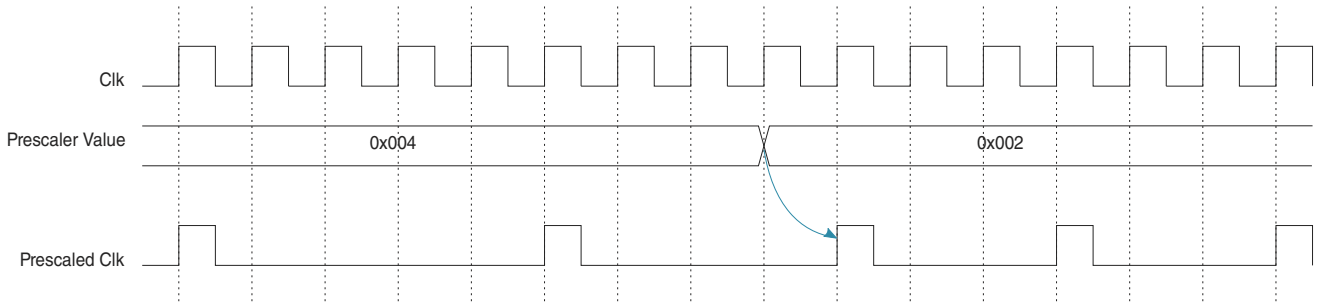


Figure 12-5. Prescaler Value Change Timing Diagram

12.1.2.4 Interrupts

There are 6 different interrupts that are generated by the GPT. If the selected clock for running the counter is available, then *all interrupts can be generated in Low Power and Debug modes*.

- Rollover Interrupt

  The Rollover Interrupt is generated when the GPT counter reaches 0xffffffff, then resets to 0x00000000 and continues counting. The Rollover Interrupt is enabled by the ROVIE bit in the GPT_IR register; the associated status bit is the ROV bit in the GPT_SR register.

- Input Capture Interrupt 1, 2

  After a capture event occurs, the associated Input Capture Channel generates an interrupt. The "capture event" interrupts are enabled by the IF2IE and IF1IE bits (in the GPT_IR register); the associated status bits are IF2 and IF1 (in the GPT_SR register). The capture of the counter value because of a capture event is *not affected by a pending capture interrupt*. The Capture register is updated with a new counter value when a capture event occurs, regardless of whether that Capture Channels' interrupt has been serviced or not.

- Output Compare Interrupt 1, 2, 3

  After a compare event occurs, the associated Output Compare Channel generates an interrupt. The "compare event" interrupts are enabled by the OF3IE, OF2IE, and OF1IE bits (in the GPT_IR register); the associated status bits are OF3, OF2, and OF1 (in the GPT_SR register). A "forced compare" does not generate an interrupt.

A *cumulative* interrupt line is also present, which is asserted whenever any of the above interrupts are posted. The cumulative interrupt line has *no* associated enables or status bits.

12.1.3 External Signals

The GPT follows the IP Bus protocol for interfacing with the processor core. The GPT does not have *any interface signals with any other module inside the chip,* except for the clock and reset inputs (from the clock and reset controller module) and for the interrupt signals *to* the processor interrupt handler. There are functional and clock inputs, and functional output signals going outside the chip boundary.

The following table describes all block signals that connect off-chip. Please see IOMUX Controller for GPT pin mux assignments.

Table 12-2. Off-Chip Module Signals

Name	Direction	Function	Reset State	Pull-Up
GPT_CLK	I	Input pin for an external clock that the counter can be operated at.	-	Passive Hysteresis
GPT_CAPTURE1	I	Input pin for a capture event for Input Capture Channel 1.	-	Passive
GPT_CAPTURE2	I	Input pin for a capture event for Input Capture Channel 2.	-	Passive
GPT_COMPARE1	O	Output pin that indicates a "compare event" occurrence in Output Compare Channel 1.	0	Passive
GPT_COMPARE2	O	Output pin that indicates a "compare event" occurrence in Output Compare Channel 2.	0	Passive
GPT_COMPARE3	O	Output pin that indicates a "compare event" occurrence in Output Compare Channel 3.	0	Passive

12.1.3.1 External Clock Input

The GPT counter can be operated using an external clock from outside the device, and this is the input pin used for that purpose. The external clock input (GPT_CLK) is treated as asynchronous to the peripheral clock (ipg_clk). To ensure proper operations of GPT, the external clock input frequency should be less than 1/4 of frequency of the peripheral clock (ipg_clk). Hysteresis characteristics on this pad will be required because this is a clock input.

12.1.3.2 Input Capture Trigger Signals

The GPT counter value can be stored in a register, triggered by an event from *outside the device*. A positive / negative edge on these GPT_CAPTURE*n* signals can trigger this capture event. These signals are treated as asynchronous to the peripheral clock (ipg_clk). Only those transitions which occur *at least a single clock cycle* (the clock selected to run the counter) *after the previous recorded transition* are guaranteed to trigger a capture event.

12.1.3.3 Output Compare Signals

The GPT_COMPARE*n* signals indicate that an output compare event has occurred on the corresponding Output Compare Channel *n*; see Output Compare.

12.1.4 Initialization/ Application Information

12.1.4.1 Selecting the Clock Source

The Clock Source field in the Control Register (CR[CLKSRC]) selects the clock source. CLKSRC should be changed only after disabling the GPT (CR[EN]=0). To change the clock source, follow this sequence:

1. Disable the GPT: CR[EN]=0
2. Disable GPT interrupts: Clear the Interrupt Register (IR)
3. Configure the Output Mode to unconnected/ disconnected for all channels: CR[OM*n*]=0
4. Disable the Input Capture Modes: CR[IM*n*]=0
5. Change the clock source CR[CLKSRC] to the desired value.
6. Assert a GPT software reset: CR[SWR]=1
7. Clear the flags in the GPT Status Register (SR) by writing 1s to them.
8. Reset the GPT counter to zero: CR[ENMOD]=1
9. Enable the GPT: CR[EN]=1
10. Enable GPT interrupts: For the desired interrupts, write 1s to their enable bits in IR.

12.1.5 Memory map and register definitions

12.1.5.1 GPT register descriptions

The GPT has 10 user-accessible 32-bit registers used to configure, operate and monitor the GPT.

A write access to the GPT Status Registers (Read-only registers ICR1, ICR2, CNT) generates a bus exception.

12.1.5.1.1 GPT memory map

DEFAULT_NICKNAME base address: 0h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	GPT Control Register (CR)	32	RW	0000_0000
4	GPT Prescaler Register (PR)	32	RW	0000_0000
8	GPT Status Register (SR)	32	W1C	0000_0000
C	GPT Interrupt Register (IR)	32	RW	0000_0000
10 - 18	GPT Output Compare Register (OCR1 - OCR3)	32	RW	FFFF_FFFF
1C - 20	GPT Input Capture Register (ICR1 - ICR2)	32	RO	0000_0000
24	GPT Counter Register (CNT)	32	RO	0000_0000

12.1.5.1.2 GPT Control Register (CR)

Contains configuration options for GPT operations, such as selecting the clock source for the counter and selecting a channel's output pin response when a compare event occurs. The CR register also includes the control bit (SWR) to trigger a software reset.

NOTE

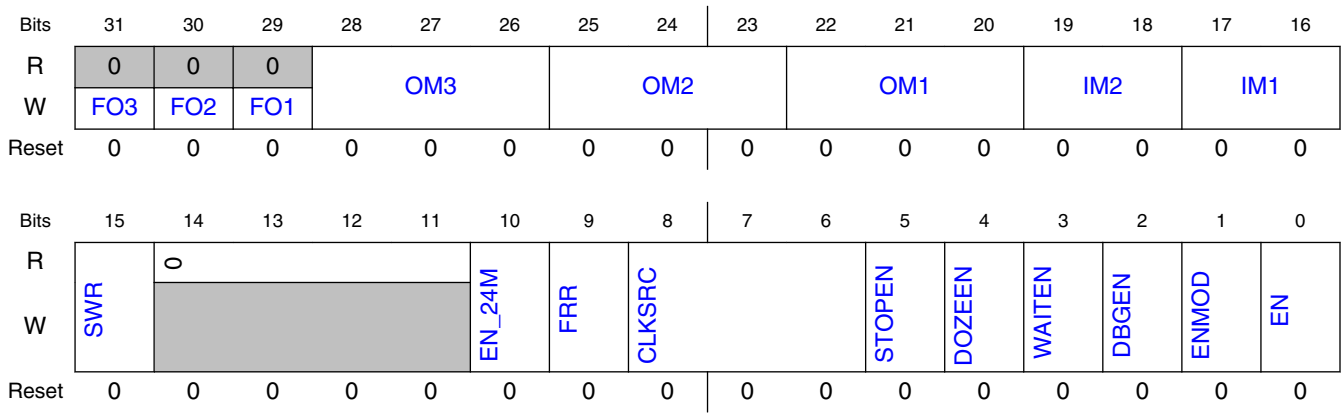
Setting the SWR bit resets **all of the GPT registers** to their default reset values except for the EN, ENMOD, STOPEN, DOZEEN, WAITEN, and DBGEN bits in this register.

For applications where precise timing is critical, be aware that writes to the CR register take effect after one cycle of the module's register bus clock (1 wait state of latency). Reads to the register occur immediately (0 wait states).

12.1.5.1.2.1 Offset

Register	Offset
CR	0h

12.1.5.1.2.2 Diagram



12.1.5.1.2.3 Fields

Field	Description
31-29 FOn	Force Output Compare for Channel n By setting the FO*n* bit, software can force the timer Output Compare *n* pin to respond according to the pin action programmed in the OM*n* bits (in this register). This bit is self-clearing and always reads as zero. **NOTE:** The OF*n* flag in the Status register (SR) is not set. 0 - No effect 1 - Trigger the programmed response on the pin
28-26: OM3 25-23: OM2 22-20: OM1	Output Compare Operating Mode for Channel n Selects the response on the output pin when a compare event occurs. 000 - Output disabled. No response on pin. 001 - Toggle output pin 010 - Clear output pin 011 - Set output pin 1xx - Generate a low pulse that is one input clock cycle wide on the output pin. When OMn is first programmed as 1xx, the output pin is set to one immediately on the next input clock (if it was not one already). "Input clock" here refers to the clock selected by the CLKSRC field of this register.
19-18: IM2 17-16: IM1	Input Capture Operating Mode for Channel n Selects what transition on the input pin triggers a capture event. 00 - Capture disabled 01 - Capture on rising edge only

Table continues on the next page...

Field	Description
	10 - Capture on falling edge only 11 - Capture on both edges
15 SWR	Software Reset Setting SWR resets **all of the registers** to their default reset values except for the DOZEEN, EN, ENMOD, STOPEN, WAITEN, and DBGEN bits in this register. The SWR bit remains set while the module is resetting. The SWR bit automatically clears when the reset procedure finishes. 0 - GPT is not in software reset state 1 - GPT is in software reset state
14-11 —	Reserved Software should write 0s to this reserved field. This field always reads as zero.
10 EN_24M	Enable Oscillator Clock Input Enables the 16-MHz clock input from an oscillator. 0 - Disable 1 - Enable
9 FRR	Free-Run or Restart Mode Selects the behavior of the GPT when a compare event in channel 1 occurs. 0 - Restart mode. After a compare event, the counter resets to 0x0000_0000 and resumes counting. 1 - Free-Run mode. After a compare event, the counter continues counting until 0xFFFF_FFFF and then rolls over to 0.
8-6 CLKSRC	Clock Source Select Selects which clock goes to the prescaler and subsequently runs the GPT counter. **NOTE:** The CLKSRC field value should be changed only after disabling the GPT (EN=0). For other programming requirements when changing the clock source, see section Selecting the Clock Source. 000 - No clock 001 - Peripheral Clock (ipg_clk) 010 - High Frequency Reference Clock (ipg_clk_highfreq) 011 - External Clock 100 - Low Frequency Reference Clock (ipg_clk_32k) 101 - Oscillator as Reference Clock (ipg_clk_16M)
5 STOPEN	GPT Stop Mode Enable Enables GPT operation *during Stop mode*. 0 - Disable in Stop mode 1 - Enable in Stop mode
4 DOZEEN	GPT Doze Mode Enable Enables GPT operation *during Doze mode*. 0 - Disable in Doze mode 1 - Enable in Doze mode
3 WAITEN	GPT Wait Mode Enable Enables GPT operation *during Wait mode*.

Table continues on the next page...

Field	Description
	0 - Disable in Wait mode 1 - Enable in Wait mode
2 DBGEN	GPT Debug Mode Enable Enables GPT operation *during Debug mode*. 0 - Disable in Debug mode 1 - Enable in Debug mode
1 ENMOD	GPT Enable Mode When the GPT is disabled (EN=0), both the Main Counter and Prescaler Counter *freeze their current count values*. The ENMOD bit selects the value of the Main Counter and Prescaler Counter when GPT is enabled again (EN=1). **NOTE:** If GPT is programmed to be disabled in a low power mode (STOP/WAIT), the Main Counter and Prescaler Counter *freeze at their current count values* when the GPT enters low power mode. When GPT exits low power mode, the Main Counter and Prescaler Counter start counting from their frozen values, regardless of the ENMOD bit value. **NOTE:** A software reset (SWR=1) clears the Main Counter and Prescaler Counter values, regardless of the value of EN or ENMOD bits. 0 - Restart counting from their frozen values after GPT is enabled (EN=1). 1 - Reset counting from 0 after GPT is enabled (EN=1).
0 EN	GPT Enable Enables the GPT module. **NOTE:** Before setting the EN bit, *all registers should be properly programmed*. 0 - Disable 1 - Enable

12.1.5.1.3 GPT Prescaler Register (PR)

Contains the *divide value* of the clock that runs the counter.

12.1.5.1.3.1 Offset

Register	Offset
PR	4h

12.1.5.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PRESCALER24M				PRESCALER											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.1.5.1.3.3 Fields

Field	Description
31-16 —	Reserved Software should write 0s to these reserved bits. These bits always read as zero.
15-12 PRESCALER24M	Prescaler divide value for the oscillator clock The 16-MHz oscillator clock is divided by [PRESCALER24M + 1] before selected by the CLKSRC field. If the 16-MHz oscillator clock is not selected, this field takes no effect. 0000 - Divide by 1 0001 - Divide by 2 1111 - Divide by 16
11-0 PRESCALER	Prescaler divide value The clock selected by the CLKSRC field is divided by [PRESCALER + 1], and then used to run the counter. **NOTE:** A change in the value of PRESCALER causes the Prescaler counter to reset and a new count period to start immediately. See Clocks for an example timing diagram. 000000000000 - Divide by 1 000000000001 - Divide by 2 111111111111 - Divide by 4096

12.1.5.1.4 GPT Status Register (SR)

Contains flags to indicate that a counter has rolled over, and if any event has occurred on the Input Capture and Output Compare channels. The flags are cleared by writing a 1 to them.

12.1.5.1.4.1 Offset

Register	Offset
SR	8h

12.1.5.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										ROV	IF2	IF1	OF3	OF2	OF1
W											W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.1.5.1.4.3 Fields

Field	Description
31-6 —	Reserved Software should write 0s to these reserved bits. These bits always read as zero.
5 ROV	Rollover Flag Indicates that the counter has reached its *maximum possible value* and *rolled over* to 0 (from which the counter continues counting). ROV is set only when the counter has reached 0xFFFFFFFF in both Restart and Free-Run modes. 0 - Rollover has not occurred. 1 - Rollover has occurred.
4-3 IFn	Input Capture Flag for Channel n Indicates that a capture event has occurred on Input Capture channel *n*. 0 - Capture event has not occurred. 1 - Capture event has occurred.
2-0 OFn	Output Compare Flag for Channel n Indicates that a compare event has occurred on Output Compare channel *n*. 0 - Compare event has not occurred. 1 - Compare event has occurred.

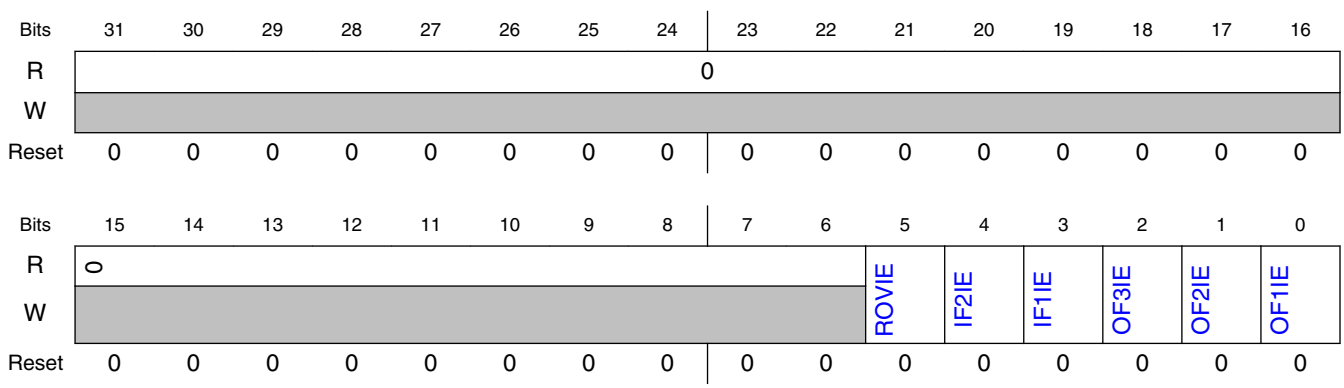
12.1.5.1.5 GPT Interrupt Register (IR)

Contains enable bits that control whether interrupts are generated after rollover, input capture and output compare events.

12.1.5.1.5.1 Offset

Register	Offset
IR	Ch

12.1.5.1.5.2 Diagram



12.1.5.1.5.3 Fields

Field	Description
31-6 —	Reserved • Writing a value to these reserved bits does not affect GPT operations. • These reserved bits are always read as zero.
5 ROVIE	Rollover Interrupt Enable Enables the Rollover interrupt. 0 - Disable 1 - Enable
4-3 IFnIE	Input Capture Flag for Channel n Interrupt Enable Enables the Input Capture Flag for Channel *n* interrupt. 0 - Disable 1 - Enable
2-0	Output Compare Flag for Channel n Interrupt Enable

Field	Description
OFnIE	Enables the Output Compare Flag for Channel *n* interrupt. 0 - Disable 1 - Enable

12.1.5.1.6 GPT Output Compare Register (OCR1 - OCR3)

Holds the value that determines when a compare event is generated on the corresponding Output Compare Channel *n*.

NOTE

The behavior of OCR1 is different compared with OCR2 and OCR3:

- A write access to OCR1 while in Restart mode (CR[FRR]=0) resets the GPT counter.
- Writes to OCR1 take effect after one cycle of the module's register bus clock (1 wait state of latency). Reads to the register occur immediately (0 wait states).

12.1.5.1.6.1 Offset

Register	Offset
OCR1	10h
OCR2	14h
OCR3	18h

12.1.5.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	COMP															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMP															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

12.1.5.1.6.3 Fields

Field	Description
31-0 COMP	Compare Value When the counter value equals COMP, a compare event is generated.

12.1.5.1.7 GPT Input Capture Register (ICR1 - ICR2)

Holds the value that was in the counter during the last capture event on the corresponding Input Capture Channel *n*.

12.1.5.1.7.1 Offset

Register	Offset
ICR1	1Ch
ICR2	20h

12.1.5.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CAPT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.1.5.1.7.3 Fields

Field	Description
31-0 CAPT	Capture Value After a capture event on Input Capture Channel *n* occurs, the current value of the counter is loaded into the corresponding ICR*n*[CAPT].

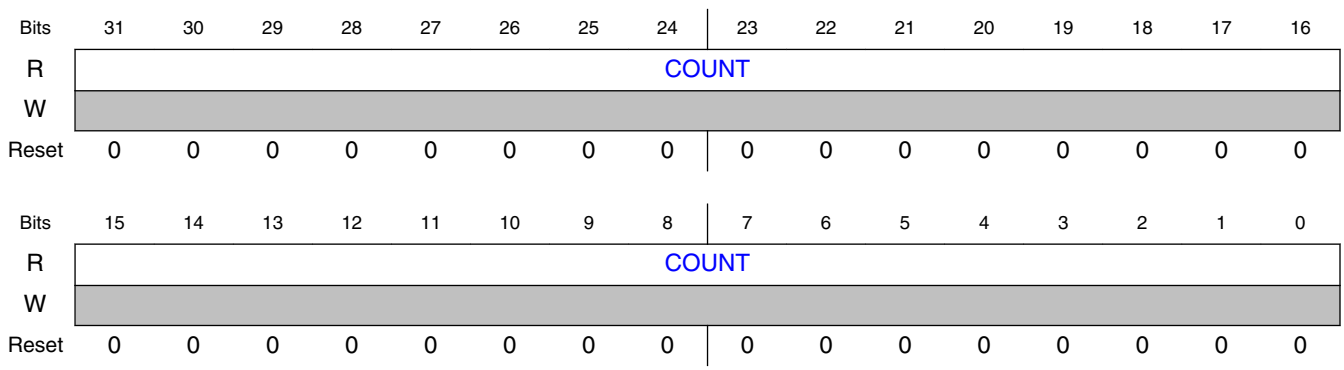
12.1.5.1.8 GPT Counter Register (CNT)

Contains the current value of the main counter. CNT can be read at any time *without affecting the counting process* of the GPT.

12.1.5.1.8.1 Offset

Register	Offset
CNT	24h

12.1.5.1.8.2 Diagram



12.1.5.1.8.3 Fields

Field	Description
31-0 COUNT	Counter Value Provides the current value of the GPT counter.

12.2 Pulse Width Modulation (PWM)

12.2.1 Overview

The Pulse Width Modulation (PWM) has a 16-bit counter, and is optimized to generate sound from stored sample audio images and it can also generate tones. It uses 16-bit resolution and a 4 x 16 data FIFO.

This section presents an overview of the PWM. A block diagram of the PWM module is shown in the figure below.

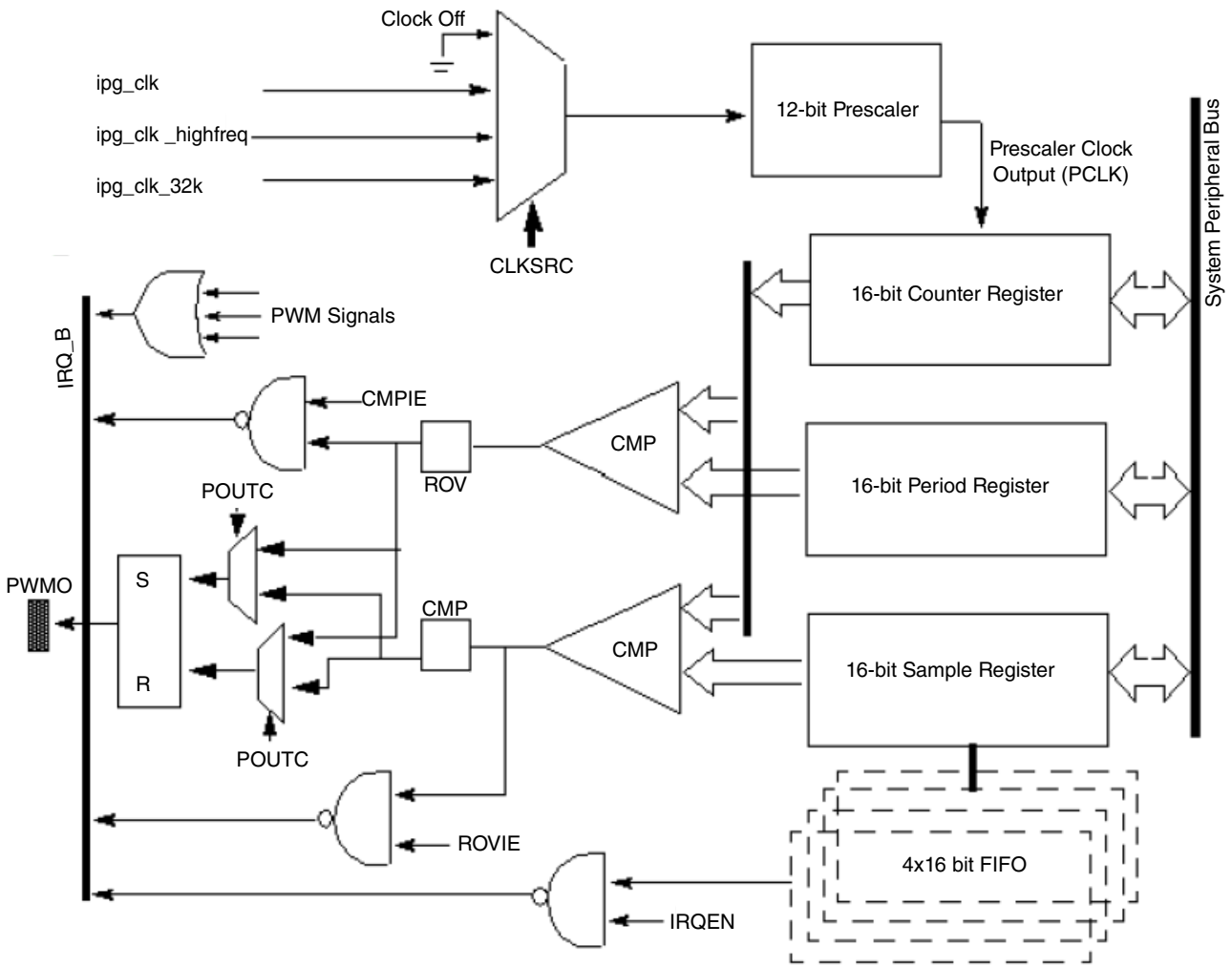


Figure 12-6. Pulse-Width Modulator Block Diagram

The following features characterize the PWM:

- 16-bit up-counter with clock source selection
- 4 x 16 FIFO to minimize interrupt overhead
- 12-bit prescaler for division of clock
- Sound and melody generation
- Active high or active low configured output
- Can be programmed to be active in low-power mode
- Can be programmed to be active in debug mode
- Interrupts at compare and rollover

12.2.2 External Signals

The PWM follows IP Bus protocol when interfacing with the processor core. PWM does not have any interface signals with any other block inside the chip except for clock and reset inputs from the Clock Control Module (CCM), System Reset Controller (SRC), and interrupt signals to the processor interrupt handler. There is a single output signal.

The following table outlines the external signals.

12.2.3 Functional Description

The following sections detail the PWM operation and function.

12.2.3.1 Operation

The output of the PWM is a toggling signal whose frequency and duty cycle can be modulated by programming the appropriate registers. It has a 16-bit up counter which counts from 0x0000 until the counter value equals the PWM_PR + 1. After this match occurs the counter is reset to 0x0000.

At the beginning of a count period cycle, the PWMO pin is set to one (default) and the counter begins counting up from 0x0000. The sample value in the sample FIFO is compared on each count of prescaler clock. When the sample and count values match, the PWMO signal is cleared to zero (default). The counter continues counting until the period match occurs and subsequently another period cycle begins.

When the PWM is enabled, the counter starts running and generates an output with the reset values in the period and sample registers. It is recommended that the programming of these registers be done before PWM is enabled.

A hardware reset results in all the PWM count and sample registers being cleared and the FIFO being flushed. The control register shows that FIFO is empty and it can be written into, and the PWM is disabled. A software reset has the same results, however the state of the DBGEN, STOPEN, DOZEN, and WAITEN bits in the control register are not affected. Software reset can be asserted even when the PWM is in disabled state.

12.2.3.1.1 FIFO

Digital sample values can be loaded into the pulse-width modulator as 16-bit words. The endianess can be changed using the BCTR and HCTR bits of the control register. A 4-word (16-bit) FIFO minimizes interrupt overhead. A maskable interrupt is generated when the number of data words fall below the water level set by the FWM field in the control register.

A write to the PWM_SAR sample register results in the value being stored into the FIFO if it is not full. A write when the FIFO is full sets FWE (FIFO write error) bit in the status register and the FIFO contents remain unchanged. The FIFO can be written at any time, but can be read only when the PWM is enabled. The PWM_SR[FIFOAV] field shows how many data words are currently contained in the FIFO and whether or not it can be written into.

A read on the sample register yields the current FIFO value that is being used, or will be used, by the PWM for generation on the output signal. Therefore, a write and a subsequent read on the sample register may result in different values being obtained.

12.2.3.1.2 Rollover and Compare Event

The counter is reset to 0x0000 after its value equals the PWM_PR[PERIOD] + 1 and resumes counting thereafter. This event is referred to as a rollover. For example, if PWM_PR[PERIOD] = 0x0000, the counter is reset when it equals 0x0001. When PWM_PR[PERIOD] = 0xFFFF or 0xFFFE, the counter is reset when it equals 0xFFFF. For more information, see the PWM Period Register (PWM_PR) description.

During a rollover event the output is either set (default), reset or has no effect according to the programming of the POUTC field in the control register. This event can also generate an interrupt if the respective interrupt enable bit is set in the control register.

When the counter value reaches the sample value, the output of the PWM is reset (default), set or has no effect according to the programming of the POUTC field of control register. This event is referred to as a compare event. This event can also generate an interrupt if the respective interrupt enable bit is set in the control register.

If the rollover event sets the PWM output signal, the compare event will reset it and vice versa for a particular programming configuration of POUTC field.

12.2.3.1.3 Low Power Mode Behavior

In low power mode, if the clock from the selected clock source is available, the PWM counter continues to run and an output is produced, depending on whether the control bit for that mode is set or not. In the absence of the clock itself, or if the corresponding low power bit in the control register is 0, the counter is reset and resumes counting when it exits the low power mode.

12.2.3.1.4 Debug Mode Behavior

In debug mode, PWM has the option of continuing to run or be halted. If the DBGEN bit is not set in the PWM_PWMCR, the PWM is halted. If the DBGEN bit is set, then the PWM will continue to run in the debug mode.

12.2.4 Enable Sequence for the PWM

The sequence found here should be used to enable the PWM.

1. Configure the desired settings for the PWM Control Register (PWMx_PWMCR) while keeping the PWM disabled (PWMx_PWMCR[0]=0).
2. Enable the desired interrupts in the PWM Interrupt Register (PWMx_PWMIR).
3. One to three initial samples may be written to the PWM Sample Register (PWMx_PWMSAR). The initial sample values will be loaded into the PWM FIFO even if the PWM is not yet enabled. Do not write a 4th sample because the FIFO will become full and trigger a FIFO Write Error (FWE). This error will prevent the PWM from starting once it is enabled.
4. Check the FIFO Write Error status bit (FWE), the Compare status bit (CMP) and the Roll-over status bit (ROV) in the PWM Status Register (PWMx_PWMSR) to make sure they are all zero. Any non-zero status bits should be cleared by writing a 1 to them.
5. Write the desired period to the PWM Period Register (PWMx_PWMPR).
6. Enable the PWM by writing a 1 to the PWM Enable bit, PWMx_PWMCR[0], while maintaining the other register bits in their previously configured state.

12.2.5 Disable Sequence for the PWM

The PWM can be disabled at any time by clearing the PWM enable bit, PWMx_PWMCR[0] to 0.

Any data remaining in the FIFO will not be produced at the PWM output after the PWM has been disabled and will remain in the FIFO until the PWM is enabled again. A software reset (setting PWMx_PWMCR[3] to 1) or a hardware reset will clear the FIFO and any remaining data will be lost.

12.2.6 PWM Memory Map/Register Definition

The PWM includes six user-accessible 32-bit registers.

PWM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3066_0000	PWM Control Register (PWM1_PWMCR)	32	R/W	0000_0000h	12.2.6.1/ 5188
3066_0004	PWM Status Register (PWM1_PWMSR)	32	w1c	0000_0008h	12.2.6.2/ 5191
3066_0008	PWM Interrupt Register (PWM1_PWMIR)	32	R/W	0000_0000h	12.2.6.3/ 5192
3066_000C	PWM Sample Register (PWM1_PWMSAR)	32	R/W	0000_0000h	12.2.6.4/ 5193
3066_0010	PWM Period Register (PWM1_PWMPR)	32	R/W	0000_FFFEh	12.2.6.5/ 5193
3066_0014	PWM Counter Register (PWM1_PWMCNR)	32	R	0000_0000h	12.2.6.6/ 5194
3067_0000	PWM Control Register (PWM2_PWMCR)	32	R/W	0000_0000h	12.2.6.1/ 5188
3067_0004	PWM Status Register (PWM2_PWMSR)	32	w1c	0000_0008h	12.2.6.2/ 5191
3067_0008	PWM Interrupt Register (PWM2_PWMIR)	32	R/W	0000_0000h	12.2.6.3/ 5192
3067_000C	PWM Sample Register (PWM2_PWMSAR)	32	R/W	0000_0000h	12.2.6.4/ 5193
3067_0010	PWM Period Register (PWM2_PWMPR)	32	R/W	0000_FFFEh	12.2.6.5/ 5193
3067_0014	PWM Counter Register (PWM2_PWMCNR)	32	R	0000_0000h	12.2.6.6/ 5194
3068_0000	PWM Control Register (PWM3_PWMCR)	32	R/W	0000_0000h	12.2.6.1/ 5188
3068_0004	PWM Status Register (PWM3_PWMSR)	32	w1c	0000_0008h	12.2.6.2/ 5191
3068_0008	PWM Interrupt Register (PWM3_PWMIR)	32	R/W	0000_0000h	12.2.6.3/ 5192

Table continues on the next page...

PWM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3068_000C	PWM Sample Register (PWM3_PWMSAR)	32	R/W	0000_0000h	12.2.6.4/ 5193
3068_0010	PWM Period Register (PWM3_PWMPR)	32	R/W	0000_FFFEh	12.2.6.5/ 5193
3068_0014	PWM Counter Register (PWM3_PWMCNR)	32	R	0000_0000h	12.2.6.6/ 5194
3069_0000	PWM Control Register (PWM4_PWMCR)	32	R/W	0000_0000h	12.2.6.1/ 5188
3069_0004	PWM Status Register (PWM4_PWMSR)	32	w1c	0000_0008h	12.2.6.2/ 5191
3069_0008	PWM Interrupt Register (PWM4_PWMIR)	32	R/W	0000_0000h	12.2.6.3/ 5192
3069_000C	PWM Sample Register (PWM4_PWMSAR)	32	R/W	0000_0000h	12.2.6.4/ 5193
3069_0010	PWM Period Register (PWM4_PWMPR)	32	R/W	0000_FFFEh	12.2.6.5/ 5193
3069_0014	PWM Counter Register (PWM4_PWMCNR)	32	R	0000_0000h	12.2.6.6/ 5194

12.2.6.1 PWM Control Register (PWMx_PWMCR)

The PWM control register (PWM_PWMCR) is used to configure the operating settings of the PWM. It contains the prescaler for the clock division.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				FWM		STOPEN	DOZEN	WAITEN	DBGEN	BCTR	HCTR	POUTC		CLKSRC	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PRESCALER												SWR	REPEAT		EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWMx_PWMCR field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–26 FWM	FIFO Water Mark. These bits are used to set the data level at which the FIFO empty flag will be set and the corresponding interrupt generated 00 FIFO empty flag is set when there are more than or equal to 1 empty slots in FIFO 01 FIFO empty flag is set when there are more than or equal to 2 empty slots in FIFO 10 FIFO empty flag is set when there are more than or equal to 3 empty slots in FIFO 11 FIFO empty flag is set when there are more than or equal to 4 empty slots in FIFO
25 STOPEN	Stop Mode Enable. This bit keeps the PWM functional while in stop mode. When this bit is cleared, the input clock is gated off in stop mode. This bit is not affected by software reset. It is cleared by hardware reset. 0 Inactive in stop mode 1 Active in stop mode
24 DOZEN	Doze Mode Enable. This bit keeps the PWM functional in doze mode. When this bit is cleared, the input clock is gated off in doze mode. This bit is not affected by software reset. It is cleared by hardware reset. 0 Inactive in doze mode 1 Active in doze mode
23 WAITEN	Wait Mode Enable. This bit keeps the PWM functional in wait mode. When this bit is cleared, the input clock is gated off in wait mode. This bit is not affected by software reset. It is cleared by hardware reset. 0 Inactive in wait mode 1 Active in wait mode
22 DBGEN	Debug Mode Enable. This bit keeps the PWM functional in debug mode. When this bit is cleared, the input clock is gated off in debug mode. This bit is not affected by software reset. It is cleared by hardware reset. 0 Inactive in debug mode 1 Active in debug mode
21 BCTR	Byte Data Swap Control. This bit determines the byte ordering of the 16-bit data when it goes into the FIFO from the sample register. 0 byte ordering remains the same 1 byte ordering is reversed
20 HCTR	Half-word Data Swap Control. This bit determines which half word data from the 32-bit IP Bus interface is written into the lower 16 bits of the sample register. 0 Half word swapping does not take place 1 Half words from write data bus are swapped
19–18 POUTC	PWM Output Configuration. This bit field determines the mode of PWM output on the output pin. 00 Output pin is set at rollover and cleared at comparison 01 Output pin is cleared at rollover and set at comparison 10 PWM output is disconnected 11 PWM output is disconnected
17–16 CLKSRC	Select Clock Source. These bits determine which clock input will be selected for running the counter. After reset the system functional clock is selected. The input clock can also be turned off if these bits are set to 00. This field value should only be changed when the PWM is disabled

Table continues on the next page...

PWMx_PWMCR field descriptions (continued)

Field	Description
	00 Clock is off 01 ipg_clk 10 ipg_clk_highfreq 11 ipg_clk_32k
15–4 PRESCALER	Counter Clock Prescaler Value. This bit field determines the value by which the clock will be divided before it goes to the counter. 0x000 Divide by 1 0x001 Divide by 2 0xfff Divide by 4096
3 SWR	Software Reset. PWM is reset when this bit is set to 1. It is a self clearing bit. A write 1 to this bit is a single wait state write cycle. When the block is in reset state this bit is set and is cleared when the reset procedure is over. Setting this bit resets all the registers to their reset values except for the DBGEN, STOPEN, DOZEN, and WAITEN bits in this control register. 0 PWM is out of reset 1 PWM is undergoing reset
2–1 REPEAT	Sample Repeat. This bit field determines the number of times each sample from the FIFO is to be used. 00 Use each sample once 01 Use each sample twice 10 Use each sample four times 11 Use each sample eight times
0 EN	PWM Enable. This bit enables the PWM. If this bit is not enabled, the clock prescaler and the counter is reset. When the PWM is enabled, it begins a new period, the output pin is set to start a new period while the prescaler and counter are released and counting begins. To make the PWM work with softreset and disable/enable, users can do software reset by seting the SWR bit, wait software reset done, configure the registers, and then enable the PWM by setting this bit to "1" Users can also disable/enable the PWM if PWM would like to be stopped and resumed with same registers configurations . 0 PWM disabled 1 PWM enabled

12.2.6.2 PWM Status Register (PWMx_PWMSR)

The PWM status register (PWM_PWMSR) contains seven bits which display the state of the FIFO and the occurrence of rollover and compare events. The FIFOAV bit is read-only but the other four bits can be cleared by writing 1 to them. The FE, ROV, and CMP bits are associated with FIFO-Empty, Roll-over, and Compare interrupts, respectively.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									FWE	CMP	ROV	FE	FIFOAV		
W										w1c	w1c	w1c	w1c			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

PWMx_PWMSR field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 FWE	FIFO Write Error Status. This bit shows that an attempt has been made to write FIFO when it is full. 0 FIFO write error not occurred 1 FIFO write error occurred
5 CMP	Compare Status. This bit shows that a compare event has occurred. 0 Compare event not occurred 1 Compare event occurred
4 ROV	Roll-over Status. This bit shows that a roll-over event has occurred. 0 Roll-over event not occurred 1 Roll-over event occurred
3 FE	FIFO Empty Status Bit. This bit indicates the FIFO data level in comparison to the water level set by FWM field in the control register. 0 Data level is above water mark 1 When the data level falls below the mark set by FWM field
FIFOAV	FIFO Available. These read-only bits indicate the data level remaining in the FIFO. An attempted write to these bits will not affect their value and no transfer error is generated. 000 No data available 001 1 word of data in FIFO

Table continues on the next page...

PWMx_PWMSR field descriptions (continued)

Field	Description
	010 2 words of data in FIFO 011 3 words of data in FIFO 100 4 words of data in FIFO 101 unused 110 unused 111 unused

12.2.6.3 PWM Interrupt Register (PWMx_PWMIR)

The PWM Interrupt register (PWM_PWMIR) contains three bits which control the generation of the compare, rollover and FIFO empty interrupts.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													CIE	RIE	FIE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWMx_PWMIR field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CIE	Compare Interrupt Enable. This bit controls the generation of the Compare interrupt. 0 Compare Interrupt not enabled 1 Compare Interrupt enabled
1 RIE	Roll-over Interrupt Enable. This bit controls the generation of the Rollover interrupt. 0 Roll-over interrupt not enabled 1 Roll-over Interrupt enabled
0 FIE	FIFO Empty Interrupt Enable. This bit controls the generation of the FIFO Empty interrupt. 0 FIFO Empty interrupt disabled 1 FIFO Empty interrupt enabled

12.2.6.4 PWM Sample Register (PWMx_PWMSAR)

The PWM sample register (PWM_PWMSAR) is the input to the FIFO. 16-bit words are loaded into the FIFO. The FIFO can be written at any time, but can be read only when the PWM is enabled. The PWM will run at the last set duty-cycle setting if all the values of the FIFO has been utilized, until the FIFO is reloaded or the PWM is disabled. When a new value is written, the duty cycle changes after the current period is over.

A value of zero in the sample register will result in the PWMO output signal always being low/high (POUTC = 00 it will be low and POUTC = 01 it will be high), and no output waveform will be produced. If the value in this register is higher than the PERIOD + 1, the output will never be set/reset depending on POUTC value.

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																SAMPLE															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWMx_PWMSAR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SAMPLE	Sample Value. This is the input to the 4x16 FIFO. The value in this register denotes the value of the sample being currently used.

12.2.6.5 PWM Period Register (PWMx_PWMPR)

The PWM period register (PWM_PWMPR) determines the period of the PWM output signal. After the counter value matches PERIOD + 1, the counter is reset to start another period.

PWMO (Hz) = PCLK(Hz) / (period +2)

A value of zero in the PWM_PWMPR will result in a period of two clock cycles for the output signal. Writing 0xFFFF to this register will achieve the same result as writing 0xFFFE.

A change in the period value due to a write in PWM_PWMPR results in the counter being reset to zero and the start of a new count period.

NOTE

Settings PWM_PWMPR to 0xFFFF when PWMx_PWMCR REPEAT bits are set to non-zero values is not allowed.

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																PERIOD															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

PWMx_PWMPR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PERIOD	Period Value. These bits determine the Period of the count cycle. The counter counts up to [Period Value] +1 and is then reset to 0x0000.

12.2.6.6 PWM Counter Register (PWMx_PWMCNR)

The read-only pulse-width modulator counter register (PWM_PWMCNR) contains the current count value and can be read at any time without disturbing the counter.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWMx_PWMCNR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Counter Value. These bits are the counter register value and denotes the current count state the counter register is in.

Chapter 13
Display, Imaging, and Camera

13.1 Display, Imaging, and Camera Overview

13.1.1 Overview

This section describes the high-level details of the peripherals that comprise the display, imaging, and camera interfaces on the chip. These peripherals are interconnected through the chip to support several application specific solutions that include the use of ML / AI accelerator and vision.

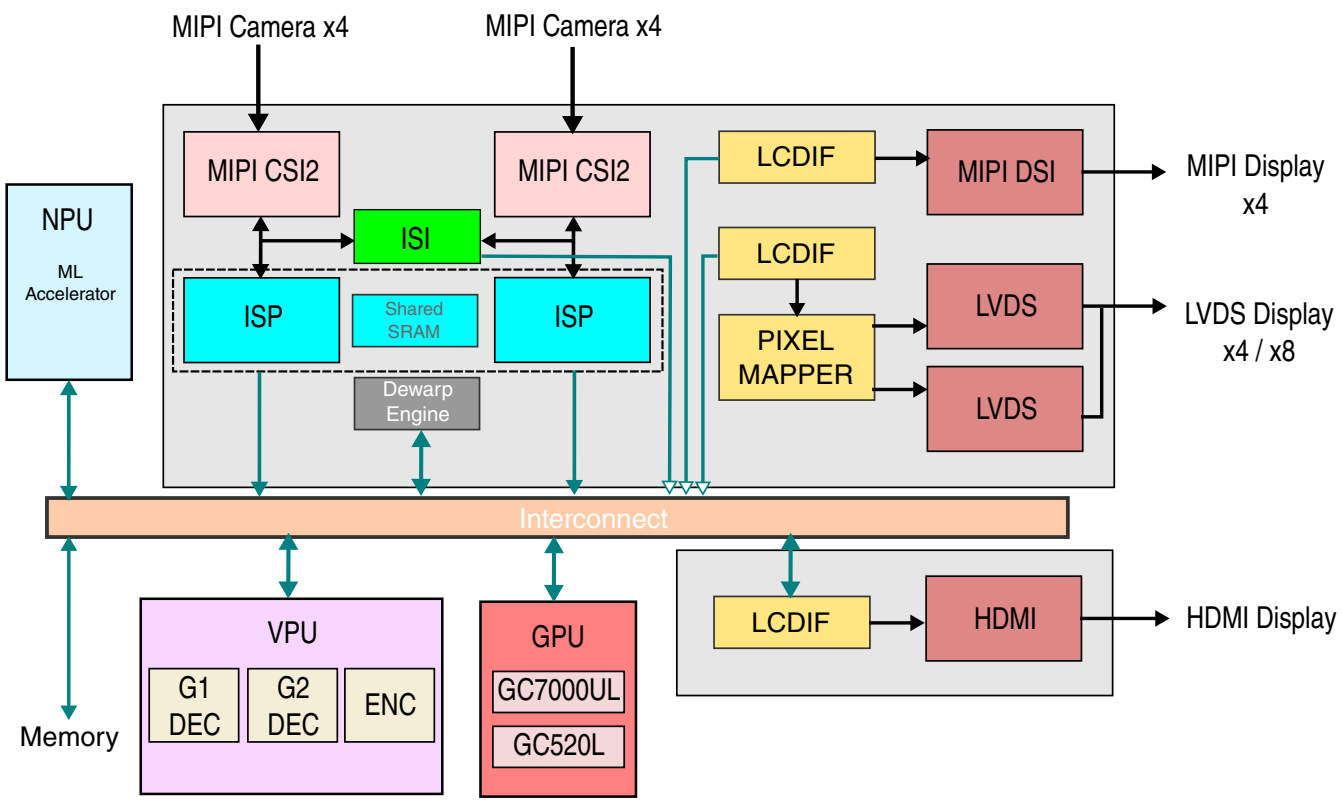


Figure 13-1. Display, Imaging, Camera I/F Diagram

13.1.2 Display Interface

The display and camera subsystem consists of several major blocks: LCDIF, LVDS Display Bridge and LVDS PHY, ISI, ISP, HDMI Tx Controller and PHY, MIPI CSI and MIPI DSI.

- The chip supports three LCDIF controllers. One LCDIF drives MIPI DSI, one LCDIF drives LVDS Tx, and one LCDIF drives HDMI Tx. The LCDIF is a simple display controller which has been widely used on i.MX chips. The key features of the display controller include:
  - Supports up to 1920x1200p60 display per LCDIF if no more than 2 instances used simultaneously, or 2x 1080p60 + 1x 4kp30 on HDMI if all 3 instances used simultaneously.
  - Supports 8-bit / 16-bit / 18-bit / 24-bit / 32-bit pixel depth
  - Supports DOTCLK mode for MIPI-DPI interface
  - Support one layer
- This chip supports one LVDS Tx display and Pixel Mapper. The LVDS port may be used as follows:
  - Single channel (4 lanes) supports resolutions up to 720p60.
  - Dual asynchronous channels (8 data, 2 clocks)
    - Supports resolutions up to 1920x1200p60.
    - Intended for a single panel with two interfaces, when transferring across two channels (even pixel/odd pixel).
    - The Pixel Mapper splits and reorders the pixels from the single LCDIF display output into an odd and even pixel stream.
- This chip supports one 4-lane MIPI DSI display with pixels from the LCDIF. The key features of the MIPI DSI (controller and PHY) include:
  - Compliant to MIPI-DSI standard v1.2
  - Support up to 4 data lanes
  - Maximum resolution limited to resolutions achievable with a 250MHz pixel clock and active pixel rate of 200Mpixel/s with 24-bit RGB. This includes resolutions such as:
    - 1080 p60
    - WUXGA (1920x1200) at 60 Hz
    - 1920x1440 at 60 Hz
    - UWHD (2560x1080) at 60 Hz
    - WQHD (2560x1440) can be supported by reduced blanking mode
  - Support 80Mbps - 1.5Gbps data rate in high speed operation
  - Support 10Mbps data rate in low power operation
- The ISI is a simple camera interface. It supports image processing and transfer via a bus master interface for up to 2 cameras. The key features of the ISI include:

- Many standard pixel formats commonly used in many camera input protocols
- Programmable resolutions up to 4K
- Image processing for
  - 2 processed camera stream at 1080p30
  - 2 unprocessed camera stream (i.e. no scaling) at 4Kp30, depending on system load and use case
- Image down scaling via decimation and bi-phase filtering
- Color space conversion
- Interlaced to progressive conversions

- The chip supports two instances of an ISP. Key features of each ISP include:
  - Bayer de-mosaicing and filtering (including denoising, sharpening and blurring)
  - Defect pixel cluster correction
  - Color processor
  - Chromatic aberration correction
  - Denoise
  - Histogram
  - Lens shading correction
  - Wide Dynamic Range (local tone mapping)
  - Color noise removal
  - Automatic white balance measurements (AWB)
  - Exposure measurement for AE (AEC/AGC) Auto focus measurement (AF)
  - High Dynamic Range (HDR), supported with:
    - Native HDR sensor with compand output
    - 2-exposure and 3-exposure Line-interleaved (staggered) HDR:
      - Full resolution for multiple exposures is output by the image sensor, and the sensor outputs are interleaved on a per-line basis. No impact to vertical resolution, but frame rate is reduced by a factor of 2 (for 2-exposure HDR) or 3 (for 3-exposure HDR).
  - High Dynamic Range (HDR) is not supported with:
    - 2-exposure Line-interlaced HDR
      - Even and odd rows are from two different exposures, reducing vertical resolution of each exposure by a factor of 2, but without impact to frame rate.
  - When one ISP is used, it supports:
    - maximum resolution up to 12MP (4096x3072)
    - maximum average pixel rate (averaged over blanking time) up to 300Mpixel/s at nominal voltage and 375Mpixel/s at overdrive voltage
    - maximum pixel clock (considering blanking time) up to 400MHz at nominal voltage and 500 MHz at overdrive voltage
  - When two ISPs are used simultaneously, each supports:
    - maximum resolution up to 1080p (1936x1188)

        - maximum average pixel rate (averaged over blanking time) up to 187.5Mpixel/s at nominal and overdrive voltage
        - maximum pixel clock (considering blanking time) up to 266MHz at nominal and overdrive voltage
    - A separate Dewarp engine is also supported, capable of reading from and writing to DRAM to adjust an image such as from a fisheye lens at resolution up to the maximum resolution supported by the ISP. One fisheye lens is supported at up to 4kp30 or 12MP (4096x3072) at 30fps.
- The chip supports two 4-lane MIPI CSI2 camera input. The key features of the MIPI CSI2 (controller and PHY) include:
    - Compliant to MIPI-CSI2 standard
    - Support up to 4 data lanes
    - When one ISP is used, MIPI CSI interface 1 supports:
        - Pixel clock up to 400MHz at nominal voltage and 500MHz at overdrive voltage
        - 80Mbps - 1.5Gbps per lane data rate in high speed operation
    - When two ISPs are used, both MIPI CSI interfaces supports:
        - Pixel clock up to 266MHz at nominal and overdrive voltage
        - 80Mbps - 1.5Gbps per lane data rate in high speed operation (maximum speed of 1.5Gbps per lane is only supported when 2 data lanes are used)
    - When used in conjunction with output to the ISP, the RAW8, RAW10, RAW12, and RAW14 image formats are supported
    - When used in conjunction with output to the ISI, the YUV420 (8-bit and 10-bit), YUV422 (8-bit and 10-bit), RGB565, RGB666, RGB888 image formats are supported
    - Support 10Mbps data rate in low power operation
- The chip supports a HDMI 2.0a Tx interface (controller and PHY). It supports following features:
    - HDMI 2.0a complaint
    - Supports a maximum display resolution of 4Kp30
    - HDMI 2.1 eARC is also supported

13.1.3 Graphics Processing Unit (GPU)

The GPU consists of a 3D graphics core and a 2D graphics core.

3D graphics core features include:
- Two Shader Execution Units
- Supports OpenGL ES 1.1, 2.0, 3.0, 3.1
- Supports OpenCL 3.0
- Supports OpenVG 1.1

- Supports OpenGL 4.0
- Supports EGL 1.5
- Supports Vulkan 1.1
- Supports tile format

2D graphics core features include:

- Supports multi-source composition
- Supports one-pass filter
- Supports tile format from 3D graphics core

13.1.4 Video Processing Unit (VPU)

The chip supports video decoding for various video format using the Hantro G2 + G1. Features include:

- 1080p60 VP9 Profile 0, 2 (10 bit) decoder (Hantro G2), input video stream can be 10-bit, the output decoded video is always 8-bit after post-processing in G2 core
- 1080p60 HEVC/H.265 decoder (Hantro G2)
- 1080p60 AVC/H.264 Baseline, Main, High decoder (Hantro G1)
- 1080p60 VP8 decoder (Hantro G1)

The video encoding features include:

- 1080p60 AVC/H.264 Encoder
- 1080p60 HEVC/H.265 Encoder

13.1.5 Image Signal Processor (ISP)

The Image Signal Processors (ISP) receive an image from the camera sensor and converts it from raw Bayer to YUV so it can be processed by the chip. The ISP also provides additional processing to improve the image quality. Supported image quality processes include:

- HDR to retain image details in high contrast scenes
- Dewarp to correct the image geometry caused by lens distortion (e.g. fisheye lens)
- Image enhancements (e.g. AWB, Denoise, AE, etc)

There are two instances of ISP on the chip and each is connected to separate instances of MIPI CSI. Both ISP instances support YCbCr420, YCbCR422, RAW8, RAW10, and RAW12 output pixel formats. The supported resolutions and formats of the two ISPs are shown in the table below:

Table 13-1. Supported Resolution and Formats - 1 ISP

Resolution	Without HDR	2-exposure line-interleaved HDR	3-exposure line-interleaved HDR
12MP (4096x3072)	30 fps	15 fps	10 fps
4k (3848x2168)	45 fps	22.5 fps	15 fps
1080p (1936x1198)	120 fps	60 fps	40 fps

Table 13-2. Supported Resolution and Formats - 2 ISPs used simultaneously (capabilities for each ISP)

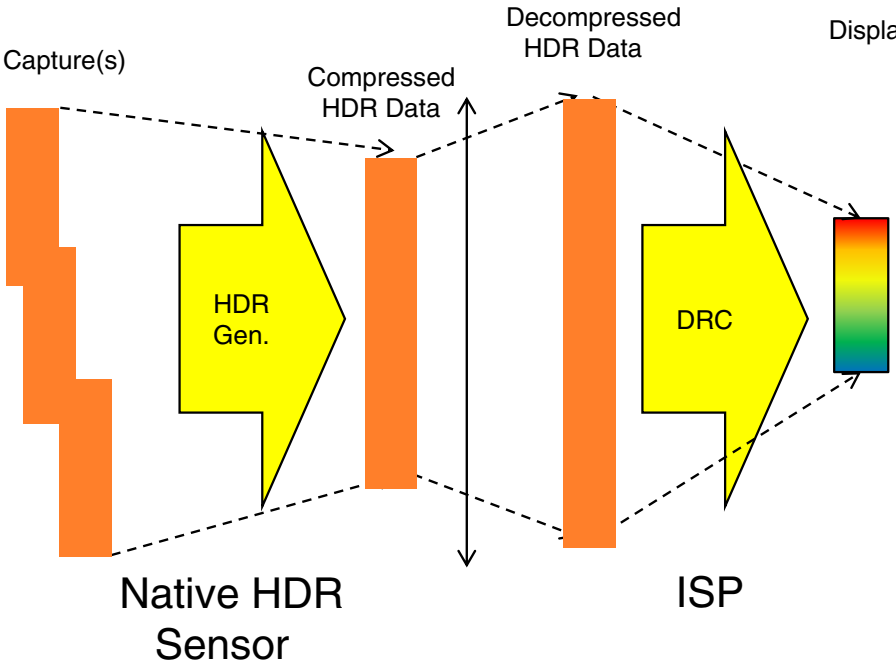
Resolution	Without HDR	2-exposure line-interleaved HDR	3-exposure line-interleaved HDR
12MP (4096x3072)	-	-	-
4k (3848x2168)	-	-	-
1080p (1936x1198)	80 fps	40 fps	27 fps

13.1.5.1 High-Dynamic Range (HDR)

The ISP supports the following HDR generation for high-quality on-the-fly dynamic range compression (DRC):

- Native HDR Sensor with compand output (max 14 bit compressed input)
- Digital Overlap High-Dynamic Range Sensor (DOL-HDR), staggered HDR

For Native HDR sensors, the multi-exposure stitching is done internally and has line buffers.



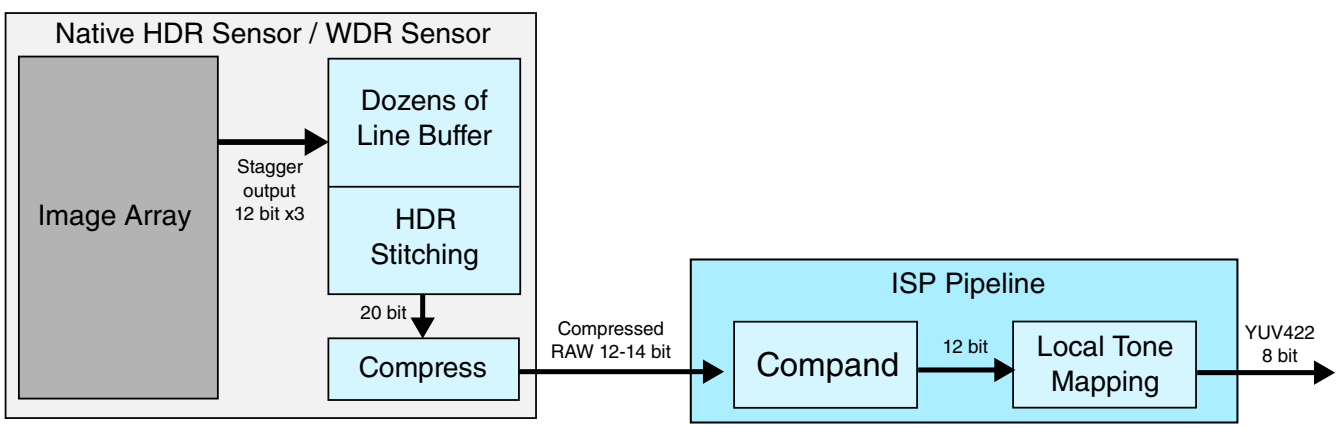


Figure 13-2. Native HDR

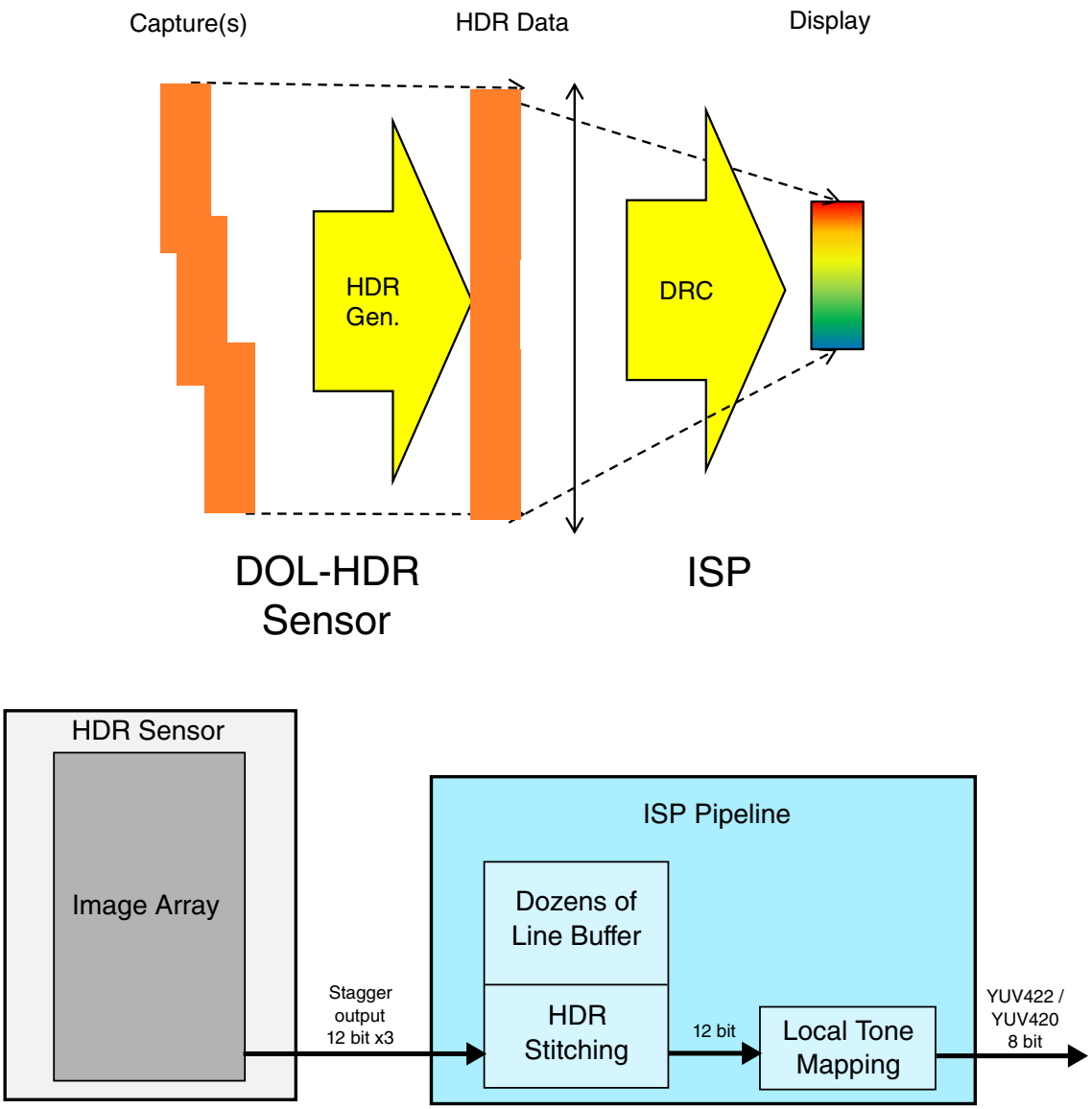


Figure 13-3. DOL-HDR

For 2 and 3-exposure DOL / Stagger HDR Sensors, the sensor outputs multi-exposure staggered pixels. The ISP allocates line buffers to buffer the exposure data and then does the stitching.

13.1.6 Dewarp Engine

A Dewarp Engine is supported on the chip that has a configurable index map to implement many different transforms to include:

- Wide FOV correction
- Lens distortion from low-cost lenses
- Keystone correction

- Fisheye correction
- Scaling and rotation

Dewarp supports one camera up to 4k or 12 MP (4096x3072) at 30 fps. Dewarp also supports two cameras up to 1080p60. YUV422 and YUV420 input and output formats are also supported

13.1.7 Neural Processing Unit (NPU)

The Neural Processing Unit (NPU) is a Machine Learning (ML) accelerator. The NPU accelerates vision image processing functions and provides enhanced performance for real-time use cases. The primary APIs supported are OpenVX 1.2.

13.2 MEDIA BLK_CTRL

13.2.1 Overview

The MEDIAMIX domain contains control and status registers known as Media Block Control (MEDIA BLK_CTRL). These registers have miscellaneous top-level controls for peripherals within the MEDIAMIX domain. Some of these controls includes clocking, interrupt, and reset sources. Please see the clock control chapter and interrupt chapter for more information regarding the clock input sources and IRQ event mapping to MEDIAMIX.

13.2.1.1 Block Diagram

The HDMI BLK_CTRL interface connects to the MEDIAMIX domain peripherals. A block diagram illustrating the MEDIAMIX domain is shown below.

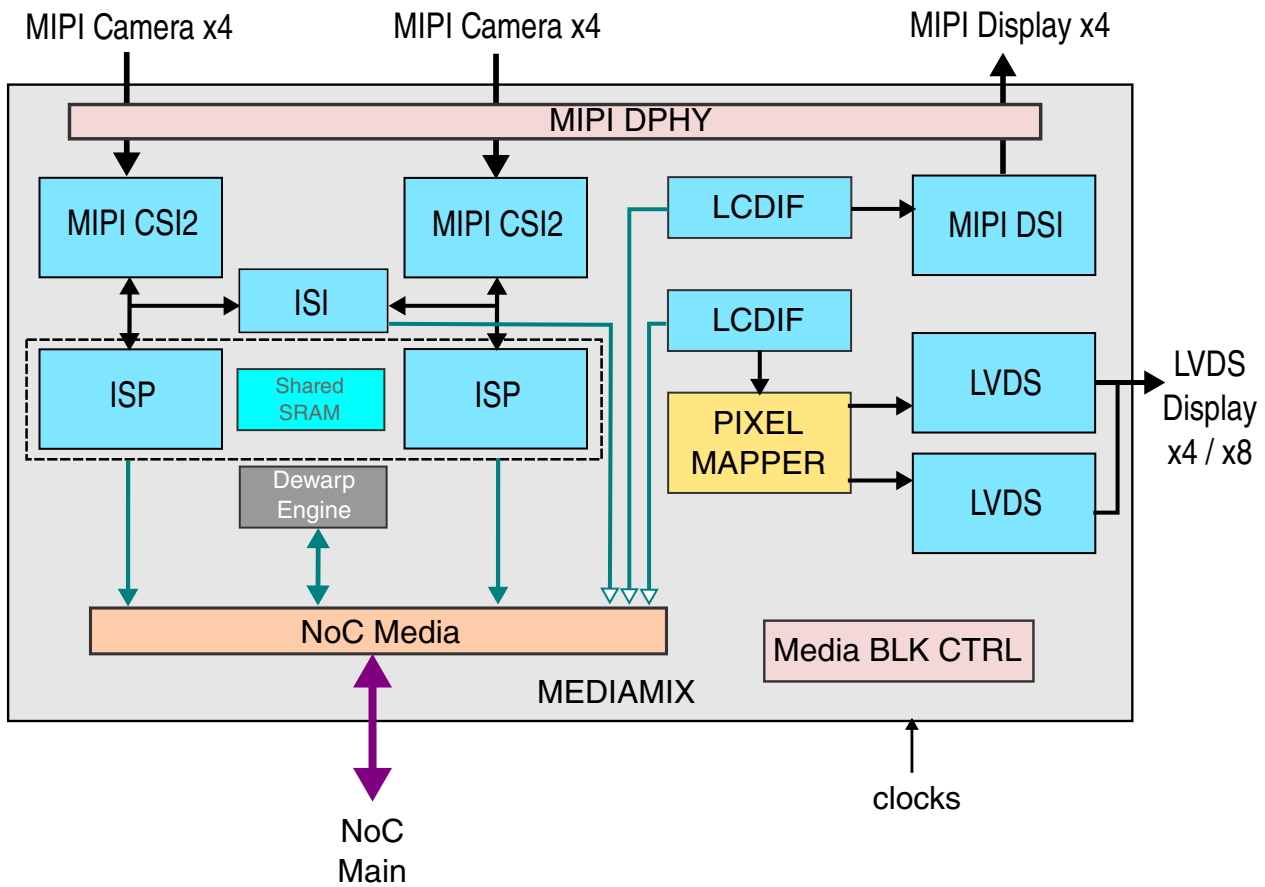


Figure 13-4. MEDIAMIX Block Diagram

13.2.2 Clocks and Resets

The Media BLK_CTRL contains clock distribution and gating controls, as well as reset handling to several of the MEDIAMIX peripherals. Please see related clock and reset controls in the BLK_CTRL register description. For more information about the input clocks, system-level resets, and IRQ to the domain, please see the related block guide and system-level assignments in the control blocks for clocks, reset, and IRQ assignments.

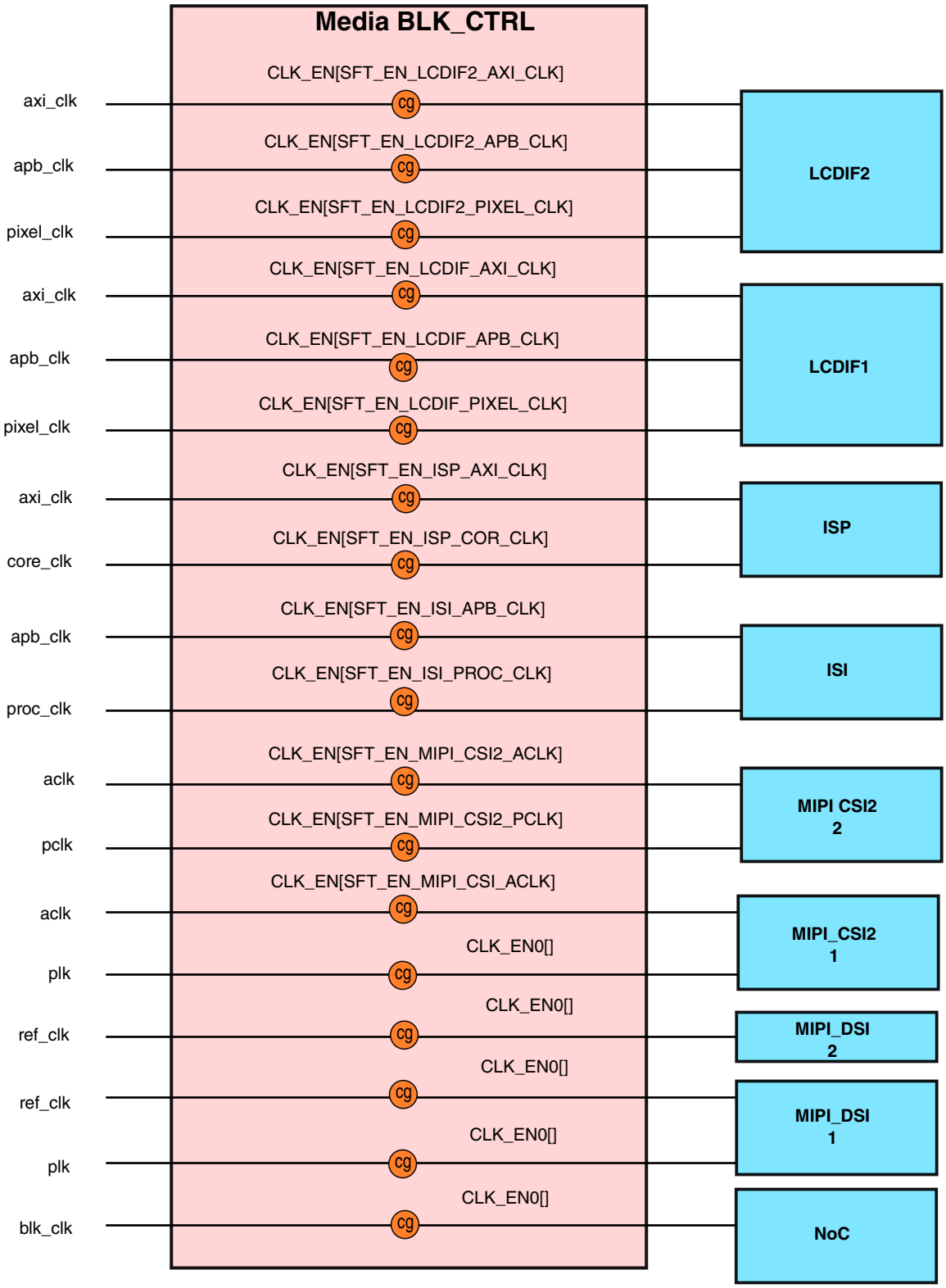


Figure 13-5. Media BLK_CTRL Clocks

13.2.3 Memory Map and register definition

This section includes the MEDIA BLK_CTRL module memory map and descriptions of all registers.

13.2.3.1 Register descriptions

13.2.3.1.1 Memory map

MEDIA_BLK_CTRL base address: 32EC_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Media Mix Software Reset Register (SFT_RSTN)	32	RW	0000_0000h
4h	Media Mix Clock Enable Register (CLK_EN)	32	RW	0000_0000h
8h	MIPI PHY Control Register (MIPI_RESET_DIV)	32	RW	0000_0000h
Ch	Master PLL PMS Value setting Register (MIPI_M_PLLPMS)	32	RW	0000_0000h
10h	Master PLL Control Low Register (MIPI_M_PLLCTL_LOW)	32	RW	0000_0000h
14h	Master PLL Control High Register (MIPI_M_PLLCTL_HIGH)	32	RW	0000_0000h
18h	Master and Slave DPHY Control Low Register (MIPI_B_DPHYCTL_LOW)	32	RW	0000_0000h
1Ch	Master and Slave DPHY Control High Register (MIPI_B_DPHYCTL_HIGH)	32	RW	0000_0000h
20h	Master and Slave DPHY Control Low Register (MIPI_M_DPHYCTL_LOW)	32	RW	0000_0000h
24h	Master and Slave DPHY Control High Register (MIPI_M_DPHYCTL_HIGH)	32	RW	0007_7400h
28h	Master and Slave DPHY Control Low Register (MIPI_S_DPHYCTL_LOW)	32	RW	0000_0000h
2Ch	Master and Slave DPHY Control High Register (MIPI_S_DPHYCTL_HIGH)	32	RW	0000_0000h
4Ch	LCDIF ARCACHE Control Register (LCDIF_ARCACHE_CTRL)	32	RW	0000_0000h
50h	ISI CACHE Control Register (ISI_CACHE_CTRL)	32	RW	0000_0000h
54h	LDO Control Register (LDO_CTRL)	32	RW	0000_0000h
58h	LDO Trim Register (LDO_TRIM)	32	RW	0000_0000h
5Ch	LDB Control Register (LDB_CTRL)	32	RW	0000_0000h
60h	Gasket 0 Control Register (GASKET_0_CTRL)	32	RW	0000_0000h
64h	Gasket 0 Video Horizontal Size Register (GASKET_0_HSIZE)	32	RW	0000_0000h
68h	Gasket 0 Video Vertical Size Register (GASKET_0_VSIZE)	32	RW	0000_0000h
6Ch	Gasket 0 Video Horizontal Front Porch Register (GASKET_0_HFP)	32	RW	0000_0000h

Table continues on the next page...

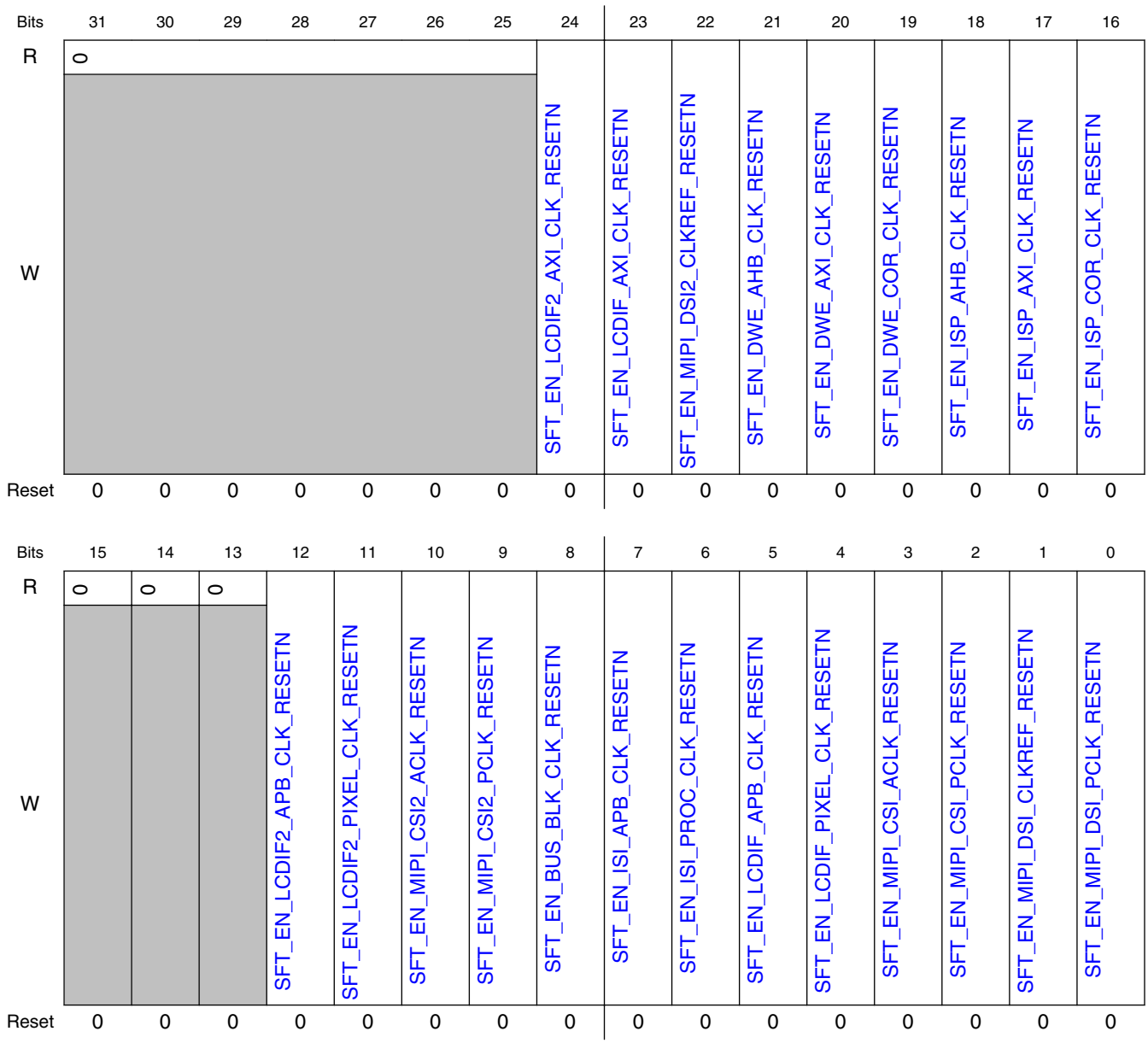
Offset	Register	Width (In bits)	Access	Reset value
70h	Gasket 0 Video Horizontal Back Porch Register (GASKET_0_HBP)	32	RW	0000_0000h
74h	Gasket 0 Video Vertical Front Porch Register (GASKET_0_VFP)	32	RW	0000_0000h
78h	Gasket 0 Video Vertical Back Porch Register (GASKET_0_VBP)	32	RW	0000_0000h
7Ch	Gasket 0 ISI Pixel Count Register (GASKET_0_ISI_PIXEL_CNT)	32	RW	0000_0000h
80h	Gasket 0 ISI Line Count Register (GASKET_0_ISI_LINE_CNT)	32	RW	0000_0000h
84h	Gasket 0 ISI Pixel Control Information Register (GASKET_0_ISI_PIXEL_CTRL)	32	RW	0000_0000h
90h	Gasket 1 Control Register (GASKET_1_CTRL)	32	RW	0000_0000h
94h	Gasket 1 Video Horizontal Size Register (GASKET_1_HSIZE)	32	RW	0000_0000h
98h	Gasket 1 Video Vertical Size Register (GASKET_1_VSIZE)	32	RW	0000_0000h
9Ch	Gasket 1 Video Horizontal Front Porch Register (GASKET_1_HFP)	32	RW	0000_0000h
A0h	Gasket 1 Video Horizontal Back Porch Register (GASKET_1_HBP)	32	RW	0000_0000h
A4h	Gasket 1 Video Vertical Front Porch Register (GASKET_1_VFP)	32	RW	0000_0000h
A8h	Gasket 1 Video Vertical Back Porch Register (GASKET_1_VBP)	32	RW	0000_0000h
ACh	Gasket 1 ISI Pixel Count Register (GASKET_1_ISI_PIXEL_CNT)	32	RW	0000_0000h
B0h	Gasket 1 ISI Line Count Register (GASKET_1_ISI_LINE_CNT)	32	RW	0000_0000h
B4h	Gasket 1 ISI Pixel Control Information Register (GASKET_1_ISI_PIXEL_CTRL)	32	RW	0000_0000h
120h	Master and Slave DPHY Control Low Register (MIPI_B2_DPHYCTL_LOW)	32	RW	0000_0000h
124h	Master and Slave DPHY Control High Register (MIPI_B2_DPHYCTL_HIGH)	32	RW	0000_0000h
128h	LVDS Control Register (LVDS_CTRL)	32	RW	0000_0000h
12Ch	AXI Limit Control Register (AXI_LIMIT_CONTROL)	32	RW	0000_0000h
130h	AXI Limit Threshold Register 0 (AXI_LIMIT_THRESH0)	32	RW	0000_0000h
134h	AXI Limit Threshold Register 1 (AXI_LIMIT_THRESH1)	32	RW	0000_0000h
138h	ISP Dewarp Control Register (ISP_DEWARP_CONTROL)	32	RW	0000_0000h

13.2.3.1.2 Media Mix Software Reset Register (SFT_RSTN)

13.2.3.1.2.1 Offset

Register	Offset
SFT_RSTN	0h

13.2.3.1.2.2 Diagram



13.2.3.1.2.3 Fields

Field	Function
31-25 —	Reserved
24 SFT_EN_LCDIF2_AXI_CLK_RESETN	sft_en_lcdif2_axi_clk_resetn lcdif2_axi_clk software reset 0b - software reset enable for lcdif2_axi_clk 1b - software reset disable for lcdif2_axi_clk

Table continues on the next page...

Field	Function
23 SFT_EN_LCDIF_AXI_CLK_RESETN	sft_en_lcdif_axi_clk_resetn lcdif_axi_clk software reset 0b - software reset enable for lcdif_axi_clk 1b - software reset disable for lcdif_axi_clk
22 SFT_EN_MIPI_DSI2_CLKREF_RESETN	sft_en_mipi_dsi2_CLKREF_resetn mipi_dsi2_CLKREF software reset 0b - software reset enable for mipi_dsi2_CLKREF 1b - software reset disable for mipi_dsi2_CLKREF
21 SFT_EN_DWE_AHB_CLK_RESETN	sft_en_dwe_ahb_clk_resetn; dwe_ahb_clk software reset 0b - software reset enable for dwe_ahb_clk 1b - software reset disable for dwe_ahb_clk
20 SFT_EN_DWE_AXI_CLK_RESETN	sft_en_dwe_axi_clk_resetn; dwe_axi_clk software reset 0b - software reset enable for dwe_axi_clk 1b - software reset disable for dwe_axi_clk
19 SFT_EN_DWE_COR_CLK_RESETN	sft_en_dwe_cor_clk_resetn; dwe_cor_clk software reset 0b - software reset enable for dwe_cor_clk 1b - software reset disable for dwe_cor_clk
18 SFT_EN_ISP_AHB_CLK_RESETN	sft_en_isp_ahb_clk_resetn; isp_ahb_clk software reset 0b - software reset enable for isp_ahb_clk 1b - software reset disable for isp_ahb_clk
17 SFT_EN_ISP_AXI_CLK_RESETN	sft_en_isp_axi_clk_resetn; isp_axi_clk software reset 0b - software reset enable for isp_axi_clk 1b - software reset disable for isp_axi_clk
16 SFT_EN_ISP_COR_CLK_RESETN	sft_en_isp_cor_clk_resetn; isp_cor_clk software reset 0b - software reset enable for isp_cor_clk 1b - software reset disable for isp_cor_clk
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 SFT_EN_LCDIF2_APB_CLK_RESETN	sft_en_lcdif2_apb_clk_resetn lcdif2_apb_clk software reset 0b - software reset enable for lcdif2_apb_clk 1b - software reset disable for lcdif2_apb_clk
11 SFT_EN_LCDIF2_PIXEL_CLK_RESETN	sft_en_lcdif2_pixel_clk_resetn lcdif2_pixel_clk software reset 0b - software reset enable for lcdif2_pixel_clk 1b - software reset disable for lcdif2_pixel_clk

Table continues on the next page...

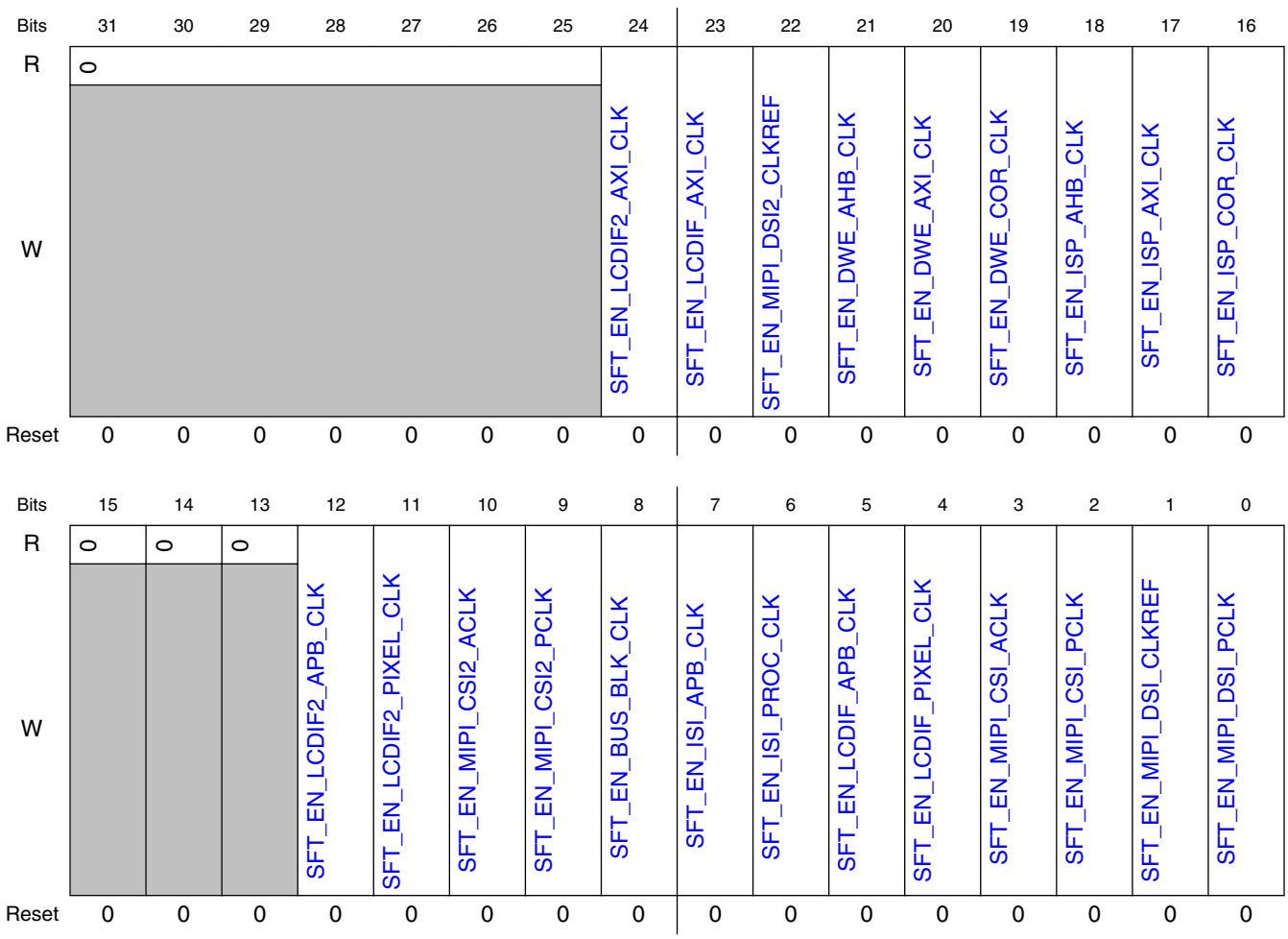
Field	Function
10 SFT_EN_MIPI_CSI2_ACLK_RESETN	sft_en_mipi_csi2_aclk_resetn mipi_csi2_aclk software reset 0b - software reset enable for mipi_csi2_aclk 1b - software reset disable for mipi_csi2_aclk
9 SFT_EN_MIPI_CSI2_PCLK_RESETN	sft_en_mipi_csi2_pclk_resetn mipi_csi2_pclk software reset 0b - software reset enable for mipi_csi2_pclk 1b - software reset disable for mipi_csi2_pclk
8 SFT_EN_BUS_BLK_CLK_RESETN	sft_en_bus_blk_clk_resetn bus_blk_clk software reset 0b - software reset enable for bus_blk_clk 1b - software reset disable for bus_blk_clk
7 SFT_EN_ISI_APB_CLK_RESETN	sft_en_isi_apb_clk_resetn isi_apb_clk software reset 0b - software reset enable for isi_apb_clk 1b - software reset disable for isi_apb_clk
6 SFT_EN_ISI_PROC_CLK_RESETN	sft_en_isi_proc_clk_resetn isi_proc_clk software reset 0b - software reset enable for isi_proc_clk 1b - software reset disable for isi_proc_clk
5 SFT_EN_LCDIF_APB_CLK_RESETN	sft_en_lcdif_apb_clk_resetn lcdif_apb_clk software reset 0b - software reset enable for lcdif_apb_clk 1b - software reset disable for lcdif_apb_clk
4 SFT_EN_LCDIF_PIXEL_CLK_RESETN	sft_en_lcdif_pixel_clk_resetn lcdif_pixel_clk software reset 0b - software reset enable for lcdif_pixel_clk 1b - software reset disable for lcdif_pixel_clk
3 SFT_EN_MIPI_CSI_ACLK_RESETN	sft_en_mipi_csi_aclk_resetn mipi_csi_aclk software reset 0b - software reset enable for mipi_csi_aclk 1b - software reset disable for mipi_csi_aclk
2 SFT_EN_MIPI_CSI_PCLK_RESETN	sft_en_mipi_csi_pclk_resetn mipi_csi_pclk software reset 0b - software reset enable for mipi_csi_pclk 1b - software reset disable for mipi_csi_pclk
1 SFT_EN_MIPI_DSI_CLKREF_RESETN	sft_en_mipi_dsi_CLKREF_resetn mipi_dsi_CLKREF software reset 0b - software reset enable for mipi_dsi_CLKREF 1b - software reset disable for mipi_dsi_CLKREF
0 SFT_EN_MIPI_DSI_PCLK_RESETN	sft_en_mipi_dsi_pclk_resetn mipi_dsi_pclk software reset 0b - software reset enable for mipi_dsi_pclk 1b - software reset disable for mipi_dsi_pclk

13.2.3.1.3 Media Mix Clock Enable Register (CLK_EN)

13.2.3.1.3.1 Offset

Register	Offset
CLK_EN	4h

13.2.3.1.3.2 Diagram



13.2.3.1.3.3 Fields

Field	Function
31-25 —	Reserved

Table continues on the next page...

Field	Function
24 SFT_EN_LCDIF2_AXI_CLK	sft_en_lcdif2_axi_clk Clock enable 0b - clock disable (gated) for lcdif2_axi_clk 1b - clock enable for lcdif2_axi_clk
23 SFT_EN_LCDIF_AXI_CLK	sft_en_lcdif_axi_clk Clock enable 0b - clock disable (gated) for lcdif_axi_clk 1b - clock enable for lcdif_axi_clk
22 SFT_EN_MIPI_DSI2_CLKREF	sft_en_mipi_dsi2_CLKREF Clock enable 0b - clock disable (gated) for mipi_dsi2_CLKREF 1b - clock enable for mipi_dsi2_CLKREF
21 SFT_EN_DWE_AHB_CLK	sft_en_dwe_ahb_clk Clock enable 0b - clock disable (gated) for dwe_ahb_clk 1b - clock enable for dwe_ahb_clk
20 SFT_EN_DWE_AXI_CLK	sft_en_dwe_axi_clk Clock enable 0b - clock disable (gated) for dwe_axi_clk 1b - clock enable for dwe_axi_clk
19 SFT_EN_DWE_COR_CLK	sft_en_dwe_cor_clk Clock enable 0b - clock disable (gated) for dwe_cor_clk 1b - clock enable for dwe_cor_clk
18 SFT_EN_ISP_AHB_CLK	sft_en_isp_ahb_clk Clock enable 0b - clock disable (gated) for isp_ahb_clk 1b - clock enable for isp_ahb_clk
17 SFT_EN_ISP_AXI_CLK	sft_en_isp_axi_clk Clock enable 0b - clock disable (gated) for isp_axi_clk 1b - clock enable for isp_axi_clk
16 SFT_EN_ISP_COR_CLK	sft_en_isp_cor_clk Clock enable 0b - clock disable (gated) for isp_cor_clk 1b - clock enable for isp_cor_clk
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 SFT_EN_LCDIF2_APB_CLK	sft_en_lcdif2_apb_clk Clock enable 0b - clock disable (gated) for lcdif2_apb_clk 1b - clock enable for lcdif2_apb_clk

Table continues on the next page...

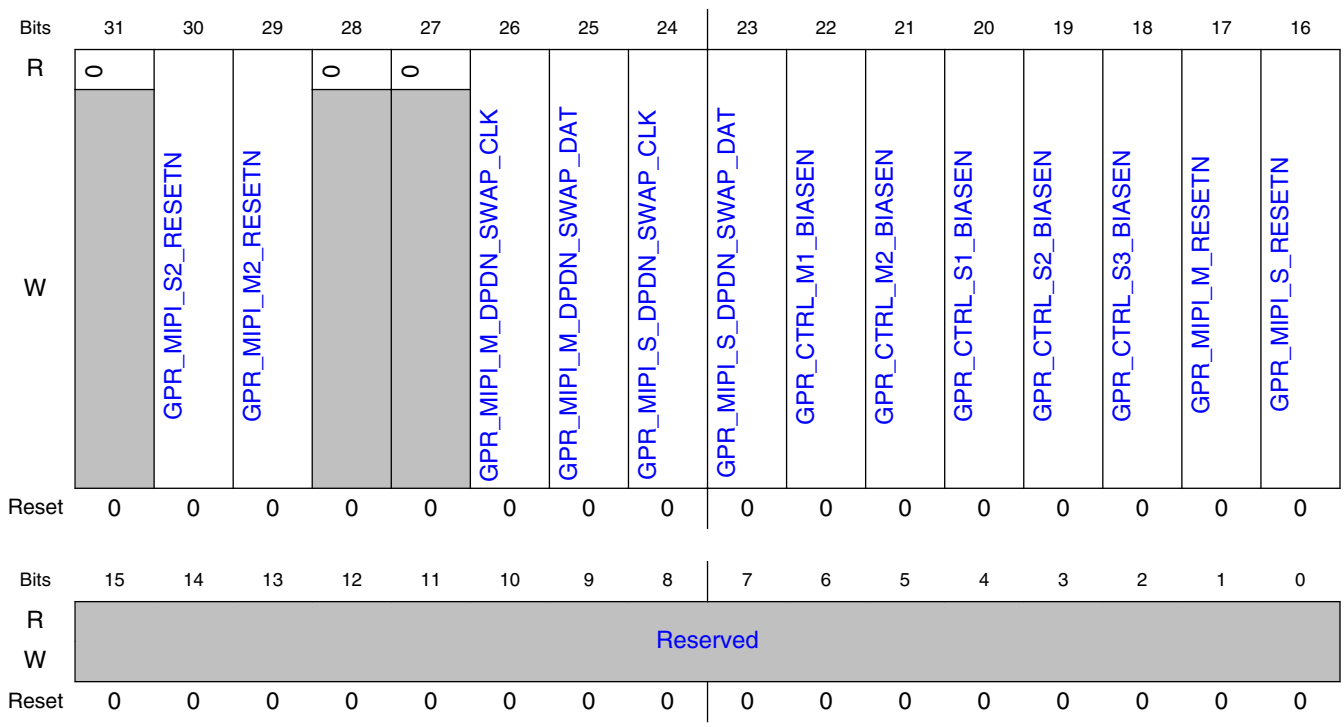
Field	Function
11 SFT_EN_LCDIF2_PIXEL_CLK	sft_en_lcdif2_pixel_clk Clock enable 0b - clock disable (gated) for lcdif2_pixel_clk 1b - clock enable for lcdif2_pixel_clk
10 SFT_EN_MIPI_CSI2_ACLK	sft_en_mipi_csi2_aclk Clock enable 0b - clock disable (gated) for mipi_csi2_aclk 1b - clock enable for mipi_csi2_aclk
9 SFT_EN_MIPI_CSI2_PCLK	sft_en_mipi_csi2_pclk Clock enable 0b - clock disable (gated) for mipi_csi2_pclk 1b - clock enable for mipi_csi2_pclk
8 SFT_EN_BUS_BLK_CLK	sft_en_bus_blk_clk Clock enable 0b - clock disable (gated) for bus_blk_clk 1b - clock enable for bus_blk_clk
7 SFT_EN_ISI_APB_CLK	sft_en_isi_apb_clk Clock enable 0b - clock disable (gated) for isi_apb_clk 1b - clock enable for isi_apb_clk
6 SFT_EN_ISI_PROC_CLK	sft_en_isi_proc_clk Clock enable 0b - clock disable (gated) for isi_proc_clk 1b - clock enable for isi_proc_clk
5 SFT_EN_LCDIF_APB_CLK	sft_en_lcdif_apb_clk Clock enable 0b - clock disable (gated) for lcdif_apb_clk 1b - clock enable for lcdif_apb_clk
4 SFT_EN_LCDIF_PIXEL_CLK	sft_en_lcdif_pixel_clk Clock enable 0b - clock disable (gated) for lcdif_pixel_clk 1b - clock enable for lcdif_pixel_clk
3 SFT_EN_MIPI_CSI_ACLK	sft_en_mipi_csi_aclk Clock enable 0b - clock disable (gated) for mipi_csi_aclk 1b - clock enable for mipi_csi_aclk
2 SFT_EN_MIPI_CSI_PCLK	sft_en_mipi_csi_pclk Clock enable 0b - clock disable (gated) for mipi_csi_pclk 1b - clock enable for mipi_csi_pclk
1 SFT_EN_MIPI_DSI_CLKREF	sft_en_mipi_dsi_CLKREF Clock enable 0b - clock disable (gated) for mipi_dsi_CLKREF 1b - clock enable for mipi_dsi_CLKREF
0 SFT_EN_MIPI_DSI_PCLK	sft_en_mipi_dsi_pclk Clock enable 0b - clock disable (gated) for mipi_dsi_pclk 1b - clock enable for mipi_dsi_pclk

13.2.3.1.4 MIPI PHY Control Register (MIPI_RESET_DIV)

13.2.3.1.4.1 Offset

Register	Offset
MIPI_RESET_DIV	8h

13.2.3.1.4.2 Diagram



13.2.3.1.4.3 Fields

Field	Function
31 —	Reserved
30 GPR_MIPI_S2_RESETN	GPR_MIPI_S2_RESETN 0b - MIPI DPHY S2_RESETN reset enable 1b - MIPI DPHY S2_RESETN reset disable
29	GPR_MIPI_M2_RESETN 0b - MIPI DPHY M2_RESETN reset enable

Table continues on the next page...

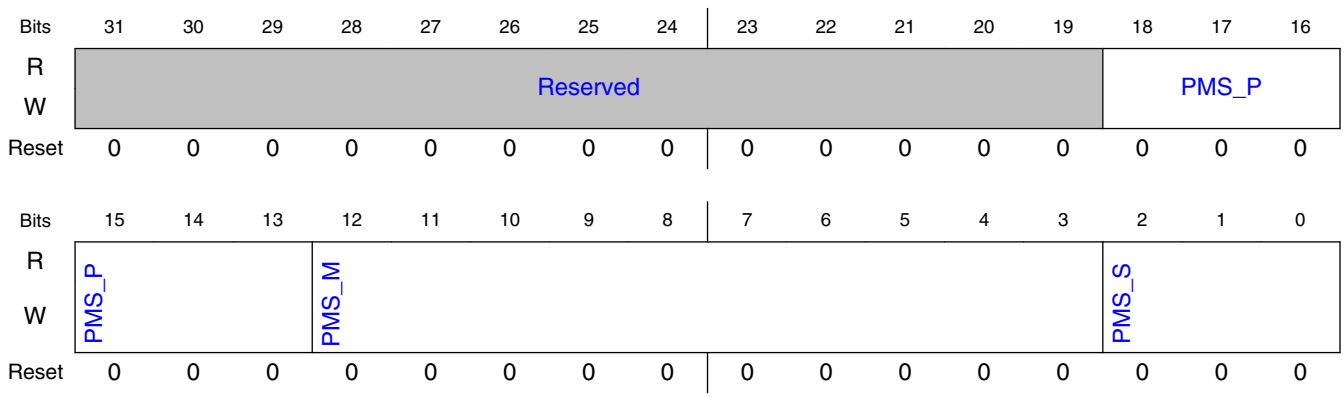
Field	Function
GPR_MIPI_M2_RESETN	1b - MIPI DPHY M2_RESETN reset disable
28 —	Reserved
27 —	Reserved
26 GPR_MIPI_M_DPDN_SWAP_CLK	GPR_MIPI_M_DPDN_SWAP_CLK 0b - Master DPHY clock lane DP and DN swap disable 1b - Master DPHY clock lane DP and DN swap enable
25 GPR_MIPI_M_DPDN_SWAP_DAT	GPR_MIPI_M_DPDN_SWAP_DAT 0b - Master DPHY data lane DP and DN swap disable 1b - Master DPHY data lane DP and DN swap enable
24 GPR_MIPI_S_DPDN_SWAP_CLK	GPR_MIPI_S_DPDN_SWAP_CLK 0b - Slave DPHY clock lane DP and DN swap disable 1b - Slave DPHY clock lane DP and DN swap enable
23 GPR_MIPI_S_DPDN_SWAP_DAT	GPR_MIPI_S_DPDN_SWAP_DAT 0b - Master DPHY data lane DP and DN swap disable 1b - Master DPHY data lane DP and DN swap enable
22 GPR_CTRL_M1_BIASEN	GPR_CTRL_M1_BIASEN. Used in MIPI PHY 0b - M1_BIASEN disable 1b - M1_BIASEN active
21 GPR_CTRL_M2_BIASEN	GPR_CTRL_M2_BIASEN. Used in MIPI PHY 0b - M2_BIASEN disable 1b - M2_BIASEN active
20 GPR_CTRL_S1_BIASEN	GPR_CTRL_S1_BIASEN. Used in MIPI PHY 0b - S1_BIASEN disable 1b - S1_BIASEN active
19 GPR_CTRL_S2_BIASEN	GPR_CTRL_S2_BIASEN. Used in MIPI PHY 0b - S2_BIASEN disable 1b - S2_BIASEN active
18 GPR_CTRL_S3_BIASEN	GPR_CTRL_S3_BIASEN. Used in MIPI PHY 0b - S3_BIASEN disable 1b - S3_BIASEN active
17 GPR_MIPI_M_RESETN	GPR_MIPI_M_RESETN 0b - MIPI DPHY M_RESETN reset enable 1b - MIPI DPHY M_RESETN reset disable
16 GPR_MIPI_S_RESETN	GPR_MIPI_S_RESETN 0b - MIPI DPHY S_RESETN reset enable 1b - MIPI DPHY S_RESETN reset disable
15-0 —	Reserved

13.2.3.1.5 Master PLL PMS Value setting Register (MIPI_M_PLLPMS)

13.2.3.1.5.1 Offset

Register	Offset
MIPI_M_PLLPMS	Ch

13.2.3.1.5.2 Diagram



13.2.3.1.5.3 Fields

Field	Function
31-19 —	-
18-13 PMS_P	PMS_P Specifies the PLL PMS value for the P divider **NOTE:** The programmable divider range should be within 1 to 63 to ensure PLL stability. The P divider value should also ensure the input frequency (Fin) is divided down between the range of 6 MHz to 300 MHz. **NOTE:** The M and P divider values should be considered together to ensure VCO ouput frequency (VCO_out) range is between 1050 MHz to 2100 MHz. Please refer to the topic DPHY PLL for more information. 000000b - Do not program, can cause malfunction 000001b - Divide by 1 000010b - Divide by 2 000011b - Divide by 3 000100b - Divide by 4 000101b - Divide by 5 011110b - Divide by 30 011111b - Divide by 31 100000b - Divide by 32

Table continues on the next page...

Field	Function
	100001b - Divide by 33
12-3 PMS_M	PMS_M Specifies the PLL PMS value for the M divider **NOTE:** The programmable divider range should be within 64 to 1023 to ensure PLL stability. **NOTE:** The M and P divider values should be considered together to ensure VCO ouput frequency (VCO_out) range is between 1050 MHz to 2100 MHz. Please refer to the topic DPHY PLL for more information. 000000000b - Do not program, can cause malfunction 000011001b - Divide by 25 000011010b - Divide by 26 000011011b - Divide by 27 000011100b - Divide by 28 000011101b - Divide by 29 001111010b - Divide by 122 001111011b - Divide by 123 001111100b - Divide by 124 001111101b - Divide by 125
2-0 PMS_S	PMS_S Specifies the PLL PMS value for the S divider Please refer to the topic DPHY PLL for more information. 00b - Divide by 1 01b - Divide by 2 10b - Divide by 4 11b - Divide by 8

13.2.3.1.6 Master PLL Control Low Register (MIPI_M_PLLCTL_LOW)

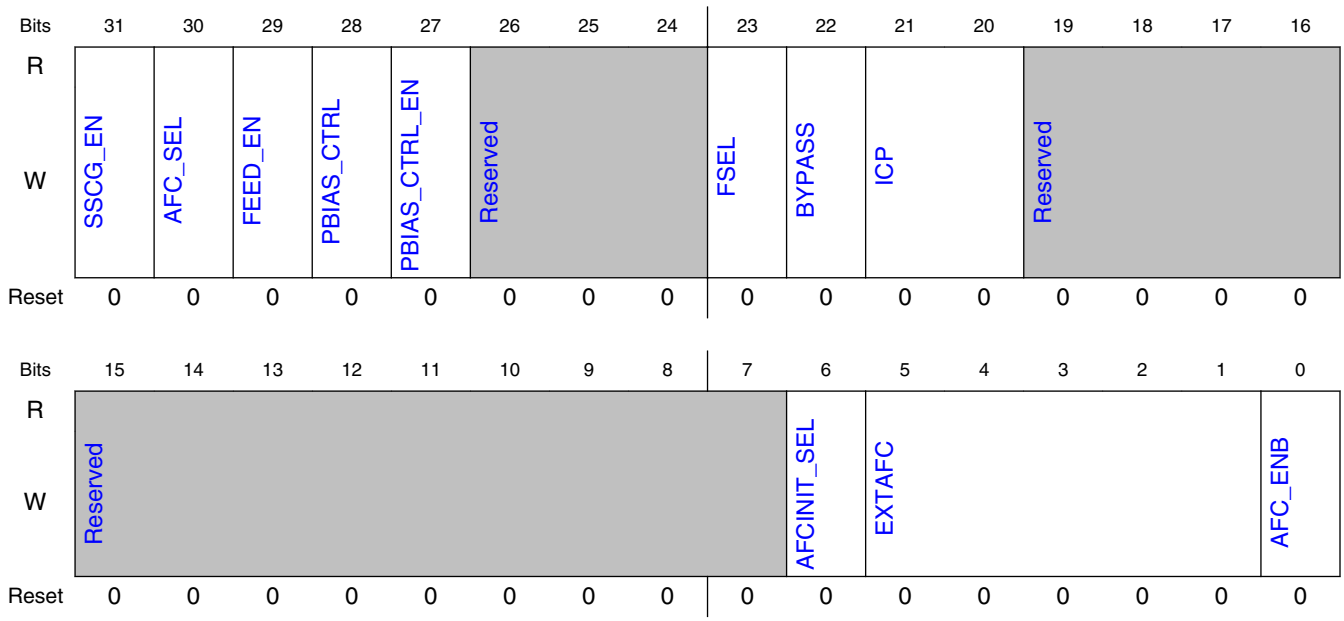
13.2.3.1.6.1 Offset

Register	Offset
MIPI_M_PLLCTL_LOW	10h

13.2.3.1.6.2 Function

This GPR is the lower 32-bits for the DPHY PLL Master PLL Control Register

13.2.3.1.6.3 Diagram



13.2.3.1.6.4 Fields

Field	Function
31 SSCG_EN	Enable pin for dithered mode. (Active-high)
30 AFC_SEL	AFC operation mode select pin Default value is 1
29 FEED_EN	FEED_OUT enable pin (active-high) Default value is 0
28 PBIAS_CTRL	PBIAS pull-down initial voltage control pin PBIAS_CTRL=1 : 0.50*VDD PBIAS_CTRL=0 : 0.67*VDD (default)
27 PBIAS_CTRL_EN	PBIAS voltage pull-down enable pin (active-high) Default value is 0
26-24 —	- -
23 FSEL	Monitoring frequency select pin FSEL = 0 for FEED_OUT = FREF FSEL = 1 for FEED_OUT = FEED
22 BYPASS	BYPASS BYPASS = 1, bypass mode is enabled If FOUT=FIN, BYPASS = 0, PLL operates normally

Table continues on the next page...

Field	Function
21-20 ICP	Controls the charge-pump current Default value is 0
19-7 —	- -
6 AFCINIT_SEL	AFC initial delay select pin AFCINIT_SEL = 0 : nominal delay time (default) AFCINIT_SEL = 1 : increase delay time
5-1 EXTAFC	EXTAFC AFC_ENB =1, AFC is disabled. FVCO is calibrated manually by EXTAFC[4:0]
0 AFC_ENB	Automatic Frequency Control Enable/Disable 0b - AFC is enabled 1b - AFC is disabled

13.2.3.1.7 Master PLL Control High Register (MIPI_M_PLLCTL_HIGH)

13.2.3.1.7.1 Offset

Register	Offset
MIPI_M_PLLCTL_HIGH	14h

13.2.3.1.7.2 Function

This GPR is for the higher 32-bits of the DPHY Master PLL Control Register

13.2.3.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	K															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SEL_PF		MRR						MFR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.7.4 Fields

Field	Function
31-16 K	Value of 16-bit Delta-Sigma Modulator (DSM)
15-14 SEL_PF	Value of 2-bit modulation method control 00: down spread 01: up spread 1x: center spread
13-8 MRR	Value of 6-bit Modulation Rate (MR) control
7-0 MFR	Value of 8-bit Modulation Frequency (MF) control

13.2.3.1.8 Master and Slave DPHY Control Low Register (MIPI_B_DPHYCTL_LOW)

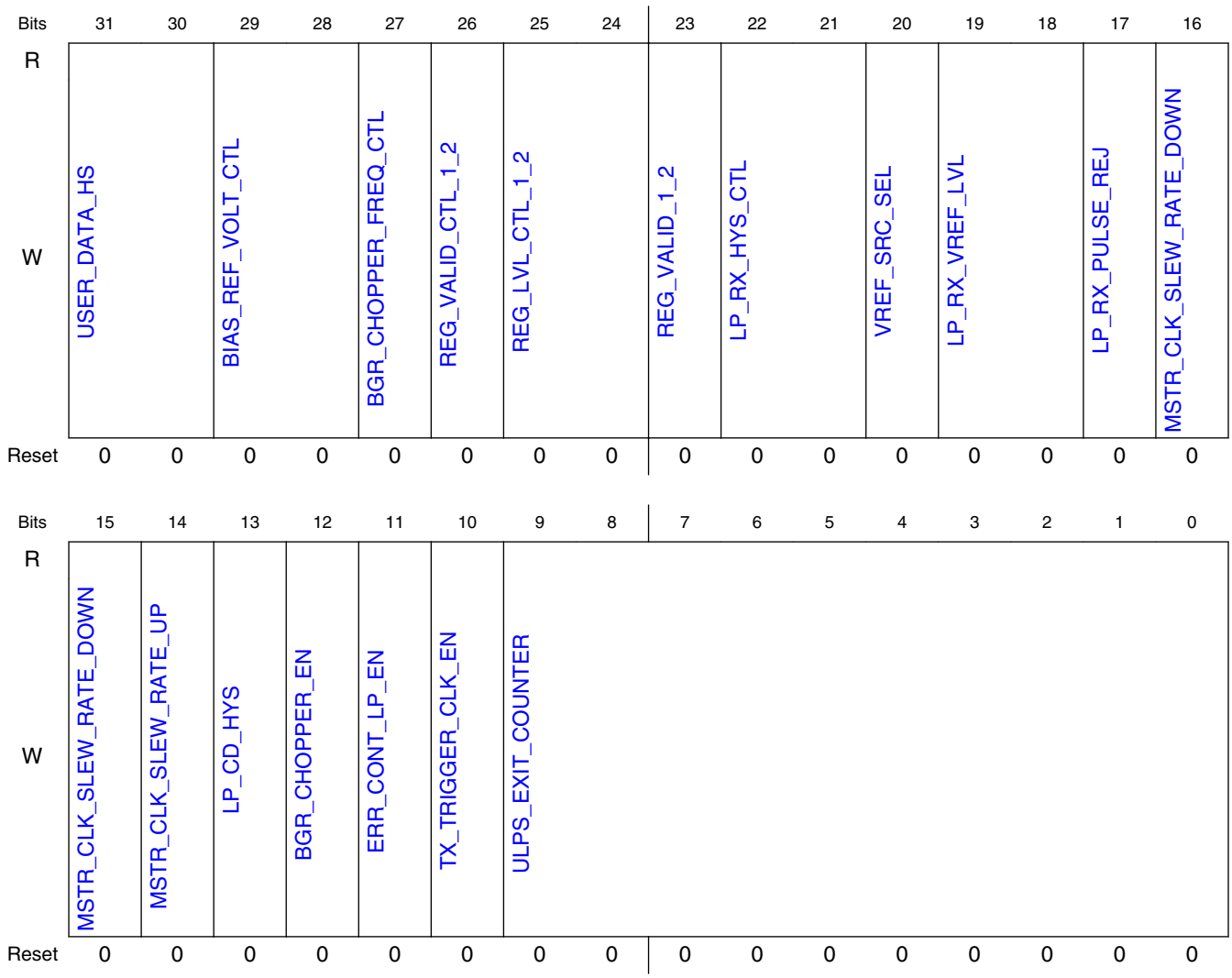
13.2.3.1.8.1 Offset

Register	Offset
MIPI_B_DPHYCTL_LOW	18h

13.2.3.1.8.2 Function

This GPR is the lower 32-bits for the B_DPHYCTL Register

13.2.3.1.8.3 Diagram



13.2.3.1.8.4 Fields

Field	Function
31-30 USER_DATA_HS	User Data Pattern for HS Loopback mode Lower 2-bit of USER_DATA_HS. Can be 2'b00 to 2'b11. Please also refer to MIPI_B_DPHYCTL_HIGH[USER_DATA_HS] for the higher 6-bits.
29-28 BIAS_REF_VOLT_CTL	BIAS_REF_VOLT_CTL Bias Reference Voltage 710m Control 00b - 712mV 01b - 724mV 10b - 733mV 11b - 706mV
27	BGR_CHOPPER_FREQ_CTL

Table continues on the next page...

Field	Function
BGR_CHOPPER_FREQ_CTL	BGR Chopper Frequency Control 0b - 3MHz 1b - 1.5MHz
26 REG_VALID_CTL_1_2	REG_VALID_CTL_1_2 VREG12_EXTPWR Enable Control 0b - Internal 1.2V regulator 1b - External 1.2V power
25-24 REG_LVL_CTL_1_2	REG_LVL_CTL_1_2 1.2 Regulator Level Control 00b - 1.2V 01b - 1.23V 10b - 1.27V 11b - 1.26V
23 REG_VALID_1_2	REG_VALID_1_2 1.2 Regulator Valid Level Section 0b - Use "ulps_en" signal 1b - Use valid signal from 1.2V regulator
22-21 LP_RX_HYS_CTL	LP_RX_HYS_CTL LP-RX Hysteresis Level Control 00b - 80mV 01b - 100mV 10b - 120mV 11b - 140mV
20 VREF_SRC_SEL	VREF_SRC_SEL VREF Source Selection 0b - Generated from the BGR 1b - Generated from the current mirror
19-18 LP_RX_VREF_LVL	LP_RX_VREF_LVL LP-RX VREF Level Control 00b - 715mV 01b - 743mV 10b - 650mV 11b - 682mV
17 LP_RX_PULSE_REJ	LP_RX_PULSE_REJ LP-RX Pulse Rejection Control 0b - Enable 1b - Disable
16-15 MSTR_CLK_SLEW_RATE_DOWN	MSTR_CLK_SLEW_RATE_DOWN Master Clock Lane's LP-TX Driver Slew Rate Down Control 00b - No change 01b - Decrease the slew rate by about 15% 10b - Decrease the slew rate by about 15% 11b - Decrease the slew rate by about 30%
14	MSTR_CLK_SLEW_RATE_UP Master Clock Lane's LP-TX Driver Slew Rate Up Control

Table continues on the next page...

Field	Function
MSTR_CLK_SLEW_RATE_UP	0b - No change 1b - Slew Rate UP
13 LP_CD_HYS	LP_CD_HYS LP-CD Hysteresis Level Control 0b - 60mV 1b - 70mV
12 BGR_CHOPPER_EN	BGR_CHOPPER_EN BGR Chopper Function Enable/Disable Control 0b - Enable 1b - Disable
11 ERR_CONT_LP_EN	ERR_CONT_LP_EN Error Contention LP Enable/Disable Control 0b - Enable 1b - Disable
10 TX_TRIGGER_CLK_EN	TX_TRIGGER_CLK_EN TX Trigger Clk Enable/Disable 0b - Enable 1b - Disable
9-0 ULPS_EXIT_COUNTER	ULPS_EXIT_COUNTER ULPS EXIT Counter value Control You should set B_DPHYCTL[9:0] during initial or power-up sequence. 0000000000b - 0.01 MHz 0000000011b - 0.10 MHz 0000011001b - 1.00 MHz 0000110010b - 2.00 MHz 0001001011b - 3.00 MHz 0001100100b - 4.00 MHz 0001111101b - 5.00 MHz 0010010110b - 6.00 MHz 0010101111b - 7.00 MHz 0011001000b - 8.00 MHz 0011100001b - 9.00 MHz 0011111010b - 10.00 MHz 0100010011b - 11.00 MHz 0100101100b - 12.00 MHz 0101000101b - 13.00 MHz 0101011110b - 14.00 MHz 0101110111b - 15.00 MHz 0110010000b - 16.00 MHz 0110101001b - 17.00 MHz 0111000010b - 18.00 MHz 0111011011b - 19.00 MHz 0111110100b - 20.00 MHz

13.2.3.1.9 Master and Slave DPHY Control High Register (MIPI_B_DPHYCTL_HIGH)

13.2.3.1.9.1 Offset

Register	Offset
MIPI_B_DPHYCTL_HIGH	1Ch

13.2.3.1.9.2 Function

This GPR is the higher 32-bits for the B_DPHYCTL Register

13.2.3.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSTR_DATA3_SLEW_RATE_DOWN		MSTR_DATA3_SLEW_RATE_UP	MSTR_DATA2_SLEW_RATE_DOWN		MSTR_DATA2_SLEW_RATE_UP	MSTR_DATA1_SLEW_RATE_DOWN		MSTR_DATA1_SLEW_RATE_UP	MSTR_DATA0_SLEW_RATE_DOWN		MSTR_DATA0_SLEW_RATE_UP	Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	LP_VREF_REG_SRC_SEL	RX_SKEW_CALIB_FIX_CODE_EN	Reserved	DCC_DONE_CTL	Reserved	BGR_VOLT_TUNING_CTL		HS_MODE_CTL		USER_DATA_HS					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.9.4 Fields

Field	Function
31-30 MSTR_DATA3_SLEW_RATE_DOWN	MSTR_DATA3_SLEW_RATE_DOWN Master Data3 Lane's LP-TX Driver Slew Rate Down Control 00b - No change 01b - Decrease the slew rate by about 15% 10b - Decrease the slew rate by about 15% 11b - Decrease the slew rate by about 30%
29 MSTR_DATA3_SLEW_RATE_UP	MSTR_DATA3_SLEW_RATE_UP Master Data3 Lane's LP-TX Driver Slew Rate Up Control 0b - No change 1b - Slew Rate UP
28-27 MSTR_DATA2_SLEW_RATE_DOWN	MSTR_DATA2_SLEW_RATE_DOWN Master Data2 Lane's LP-TX Driver Slew Rate Down Control 00b - No change 01b - Decrease the slew rate by about 15% 10b - Decrease the slew rate by about 15% 11b - Decrease the slew rate by about 30%
26 MSTR_DATA2_SLEW_RATE_UP	MSTR_DATA2_SLEW_RATE_UP Master Data2 Lane's LP-TX Driver Slew Rate Up Control 0b - No change 1b - Slew Rate UP
25-24 MSTR_DATA1_SLEW_RATE_DOWN	MSTR_DATA1_SLEW_RATE_DOWN Master Data1 Lane's LP-TX Driver Slew Rate Down Control 00b - No change 01b - Decrease the slew rate by about 15% 10b - Decrease the slew rate by about 15% 11b - Decrease the slew rate by about 30%
23 MSTR_DATA1_SLEW_RATE_UP	MSTR_DATA1_SLEW_RATE_UP Master Data1 Lane's LP-TX Driver Slew Rate Up Control 0b - No change 1b - Slew Rate UP
22-21 MSTR_DATA0_SLEW_RATE_DOWN	MSTR_DATA0_SLEW_RATE_DOWN Master Data0 Lane's LP-TX Driver Slew Rate Down Control 00b - No change 01b - Decrease the slew rate by about 15% 10b - Decrease the slew rate by about 15% 11b - Decrease the slew rate by about 30%
20 MSTR_DATA0_SLEW_RATE_UP	MSTR_DATA0_SLEW_RATE_UP Master Data0 Lane's LP-TX Driver Slew Rate Up Control 0b - No change 1b - Slew Rate UP
19-15	-

Table continues on the next page...

Field	Function
—	-
14 LP_VREF_REG_SRC_SEL	LP_VREF_REG_SRC_SEL LP Regulator Vref Source Selection 0b - Generated from BGR 1b - Generated from Current Mirror
13 RX_SKEW_CALIB_FIX_CODE_EN	RX_SKEW_CALIB_FIX_CODE_EN RX Skew Calibration Fixing Code Enable/Disable Control 0b - Disable 1b - Enable
12 —	- -
11 DCC_DONE_CTL	DCC_DONE_CTL DCC "DONE" Signal Control 0b - "DONE" from DCC block 1b - U"DONE" is always 1
10 —	- -
9-8 BGR_VOLT_TUNING_CTL	BGR_VOLT_TUNING_CTL BGR Voltage Tuning Control 00b - 820mV 01b - 760mV 10b - 800mV 11b - 840mV
7-6 HS_MODE_CTL	HS_MODE_CTL HS Loopback Mode Control 00b - Designated Pattern 01b - PRBS7 10b - All zero 11b - User Data Pattern
5-0 USER_DATA_HS	User Data Pattern for HS Loopback mode Higher 6-bit of USER_DATA_HS. Can be 6'b000000 to 6'b111111. Please also refer to MIPI_B_DPHYCTL_HIGH[USER_DATA_HS] for the lower 2-bits.

13.2.3.1.10 Master and Slave DPHY Control Low Register (MIPI_M_DPHYCTL_LOW)

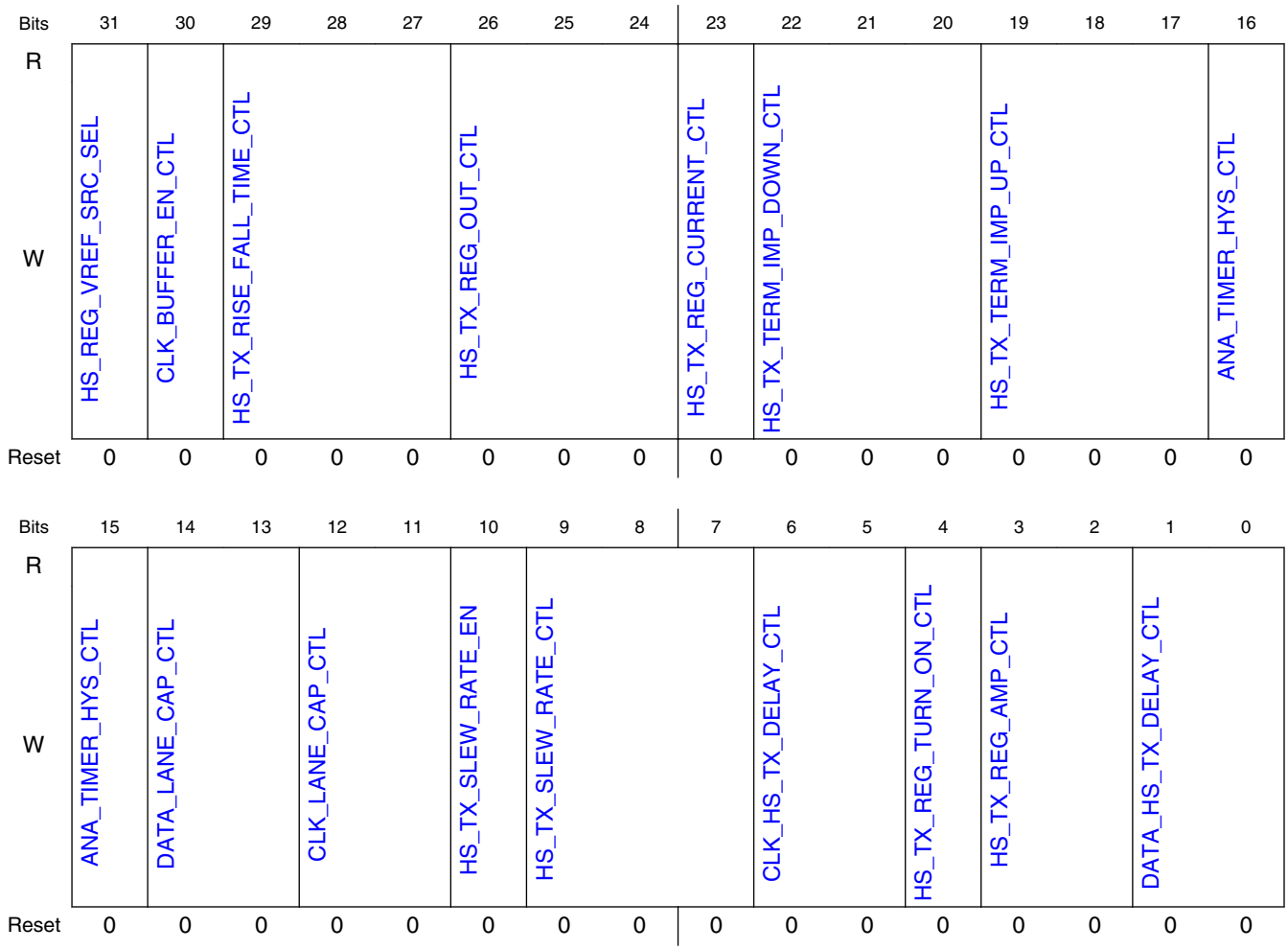
13.2.3.1.10.1 Offset

Register	Offset
MIPI_M_DPHYCTL_LOW	20h

13.2.3.1.10.2 Function

This GPR is the lower 32-bits for the M_DPHYCTL Register

13.2.3.1.10.3 Diagram



13.2.3.1.10.4 Fields

Field	Function
31 HS_REG_VREF_SRC_SEL	HS_REG_VREF_SRC_SEL HS regulator Vref Source Selection 0b - Generated from BGR 1b - Generated from Current Mirror
30 CLK_BUFFER_EN_CTL	CLK_BUFFER_EN_CTL Clock Buffer Enable Select Control 0b - HS_TX enable

Table continues on the next page...

Field	Function
	1b - PLL lock
29-27 HS_TX_RISE_FALL_TIME_CTL	HS_TX_RISE_FALL_TIME_CTL HS-TX Rise and Fall Time Control 000b - 135mV 001b - 130mV 010b - 125mV 011b - 120mV 100b - 230mV 101b - 225mV 110b - 220mV 111b - 215mV
26-24 HS_TX_REG_OUT_CTL	HS_TX_REG_OUT_CTL HS-TX Regulator Output Level Control 000b - 400mV 001b - 410mV 010b - 420mV 011b - 440mV 100b - 200mV 101b - 360mV 110b - 380mV 111b - 390mV
23 HS_TX_REG_CURRENT_CTL	HS_TX_REG_CURRENT_CTL HS-TX Regulator Reference Current Control 0b - No change 1b - 2.5 uA
22-20 HS_TX_TERM_IMP_DOWN_CTL	HS_TX_TERM_IMP_DOWN_CTL HS-TX Driver Termination Impedance Control (Down) 000b - 50 ohm 001b - 52 ohm 010b - 54 ohm 011b - 56 ohm 100b - 44 ohm 101b - 46 ohm 110b - 47 ohm 111b - 48 ohm
19-17 HS_TX_TERM_IMP_UP_CTL	HS_TX_TERM_IMP_UP_CTL HS-TX Driver Termination Impedance Control (Up) 000b - 50 ohm 001b - 52 ohm 010b - 54 ohm 011b - 56 ohm 100b - 44 ohm 101b - 46 ohm 110b - 47 ohm 111b - 48 ohm
16-15 ANA_TIMER_HYS_CTL	ANA_TIMER_HYS_CTL Analog Timer Hysteresis Control 00b - 70mV 01b - 95mV

Table continues on the next page...

Field	Function
	10b - 95mV 11b - 110mV
14-13 DATA_LANE_CAP_CTL	DATA_LANE_CAP_CTL Data Lane Cap. Value Control for TCLK_PREPARE 00b - No change 01b - -6.6% 10b - 13.3% 11b - 6.6%
12-11 CLK_LANE_CAP_CTL	CLK_LANE_CAP_CTL Clock Lane Cap. Value Control for TCLK_PREPARE 00b - No change 01b - -6.6% 10b - 13.3% 11b - 6.6%
10 HS_TX_SLEW_RATE_EN	HS_TX_SLEW_RATE_EN HS-TX Driver Slew Rate Enable 0b - SRC Disable 1b - SRC Enable
9-7 HS_TX_SLEW_RATE_CTL	HS_TX_SLEW_RATE_CTL HS-TX Driver Slew Rate Control(SRC) Value
6-5 CLK_HS_TX_DELAY_CTL	CLK_HS_TX_DELAY_CTL Clock Lane HS-TX Driver Delay Control 00b - No change 01b - 25 ps 10b - 55 ps 11b - 90 ps
4 HS_TX_REG_TURN_ON_CTL	HS_TX_REG_TURN_ON_CTL HS-TX Regulator Turn-on Control. 0b - No Change 1b - Always Turn-On the HS Regulator
3-2 HS_TX_REG_AMP_CTL	HS_TX_REG_AMP_CTL HS-TX Regulator Amp current Control 00b - No Change 01b - 12.5 uA 10b - 50 uA 11b - 16.6 uA
1-0 DATA_HS_TX_DELAY_CTL	DATA_HS_TX_DELAY_CTL Data Lane HS-TX Driver Delay Control 00b - No change 01b - 25 ps 10b - 55 ps 11b - 90 ps

13.2.3.1.11 Master and Slave DPHY Control High Register (MIPI_M_DPHYCTL_HIGH)

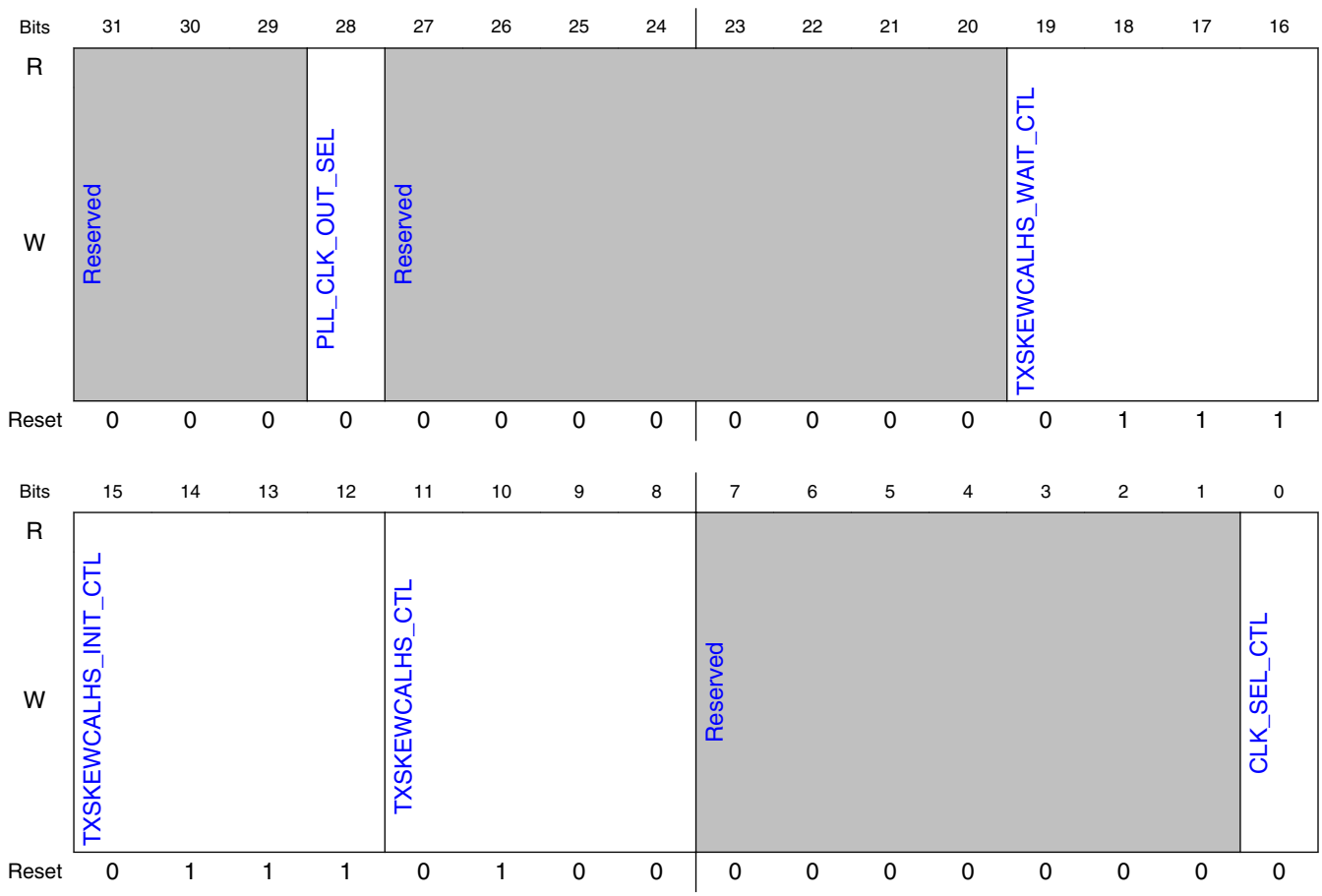
13.2.3.1.11.1 Offset

Register	Offset
MIPI_M_DPHYCTL_HIGH	24h

13.2.3.1.11.2 Function

This GPR is the higher 32-bits for the M_DPHYCTL Register

13.2.3.1.11.3 Diagram



13.2.3.1.11.4 Fields

Field	Function
31-29 —	- Reserved
28 PLL_CLK_OUT_SEL	PLL_CLK_OUT_SEL PLL Clock Selection 0b - Disable OUT to Other lane 1b - Enable OUT to Other lane
27-20 —	- Reserved
19-16 TXSKEWCALHS_WAIT_CTL	TXSKEWCALHS_WAIT_CTL TxSkewCalHS Initial Wait Time Control Timing count for the waiting initial skew calibration function after power on. If M_DPHYCTL[51:48] = 0, the initial skew calibration function does not work.
15-12 TXSKEWCALHS_INIT_CTL	TXSKEWCALHS_INIT_CTL TxSkewCalHS Initial Run Time Control Timing count for the initial skew calibration function's running time. If M_DPHYCTL[47:44] = 0, timing is same as count 7.
11-8 TXSKEWCALHS_CTL	TXSKEWCALHS_CTL TxSkewCalHS Run Time Control Timing count for the skew calibration function's running time when M_TXSKEWCALHS is enabled. If M_DPHYCTL[43:40] = 0, timing is same as count 4.
7-1 —	- Reserved
0 CLK_SEL_CTL	CLK_SEL_CTL Clock Signal Control Selection (Use Only in case of M4S0, M1S0) 0b - Generated from Internal PLL 1b - Generated from External PLL

13.2.3.1.12 Master and Slave DPHY Control Low Register (MIPI_S_DPHYCTL_LOW)

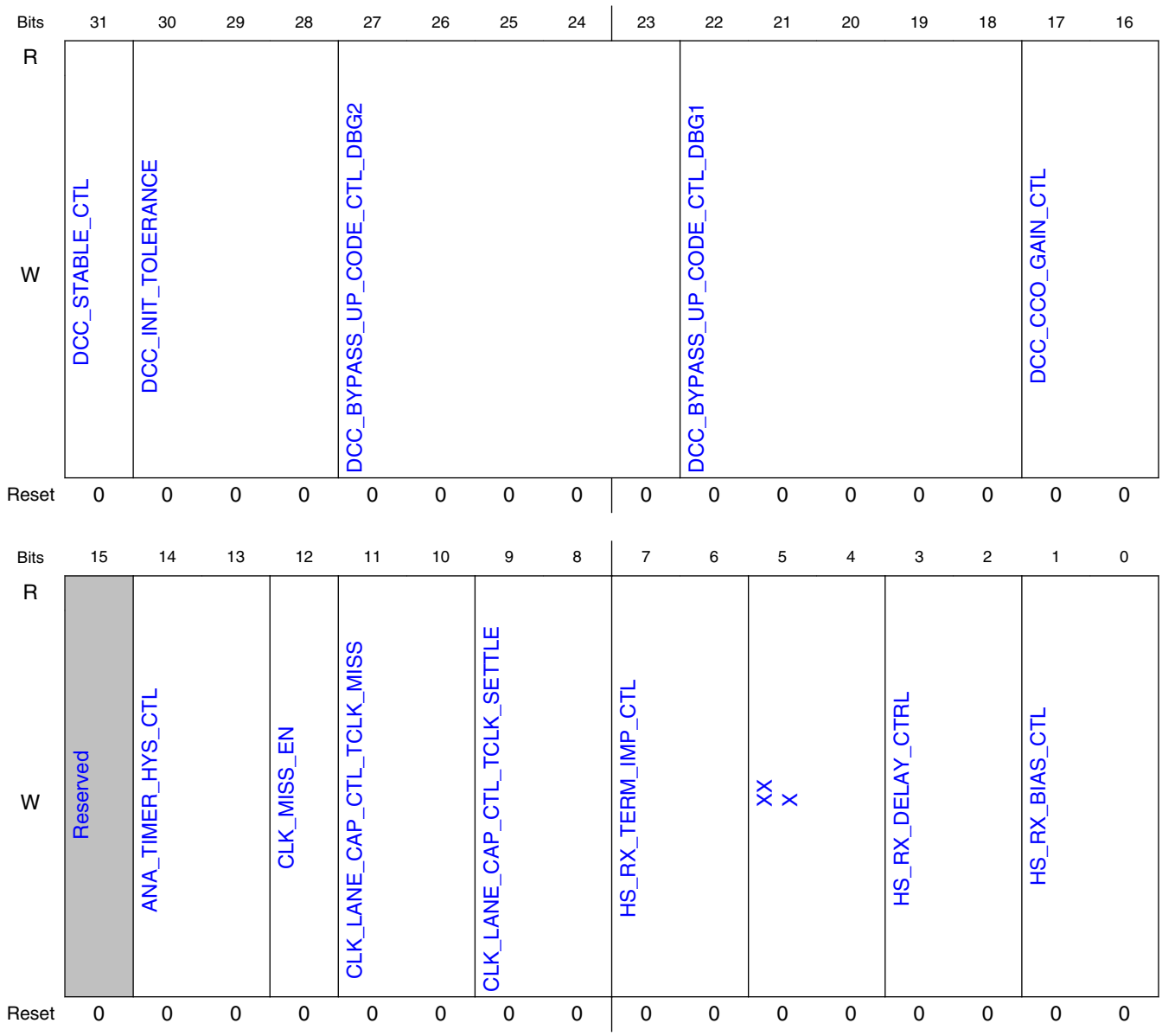
13.2.3.1.12.1 Offset

Register	Offset
MIPI_S_DPHYCTL_LOW	28h

13.2.3.1.12.2 Function

This GPR is the lower 32-bits for the S_DPHYCTL Register

13.2.3.1.12.3 Diagram



13.2.3.1.12.4 Fields

Field	Function
31 DCC_STABLE_CTL	DCC_STABLE_CTL DCC Stable Control 0b - 1 number counter running

Table continues on the next page...

Field	Function
	1b - 2 number counter running
30-28 DCC_INIT_TOLERANCE	DCC_INIT_TOLERANCE DCC Initial Tolerance 000b - 4 001b - 5 010b - 6 011b - 7 100b - 0 101b - 1 110b - 2 111b - 3
27-23 DCC_BYPASS_UP_CODE_CTL_DBG2	DCC_BYPASS_UP_CODE_CTL_DBG2 DCC Bypass Up Code Control for debugging
22-18 DCC_BYPASS_UP_CODE_CTL_DBG1	DCC_BYPASS_UP_CODE_CTL_DBG1 DCC Bypass Up Code Control for debugging
17-16 DCC_CCO_GAIN_CTL	DCC_CCO_GAIN_CTL DCC CCO Gain Control 00b - 1/1 01b - 1/4 10b - 1/0.66 11b - 1/1.33
15 —	- Reserved
14-13 ANA_TIMER_HYS_CTL	ANA_TIMER_HYS_CTL Analog Timer Hysteresis Control 00b - 70mV 01b - 95mV 10b - 95mV 11b - 110mV
12 CLK_MISS_EN	CLK_MISS_EN Clock Miss Function Enable/Disable Control 0b - Enable 1b - Disable
11-10 CLK_LANE_CAP_CTL_TCLK_MISS	CLK_LANE_CAP_CTL_TCLK_MISS Clock Lane Cap. Value Control for TCLK-MISS 00b - No change 01b - 6.6% 10b - 13.2% 11b - 19.8%
9-8 CLK_LANE_CAP_CTL_TCLK_SETTLE	CLK_LANE_CAP_CTL_TCLK_SETTLE Clock Lane Cap. Value Control for TCLK-SETTLE 00b - No change

Table continues on the next page...

Field	Function
	01b - 3.45% 10b - 6.9% 11b - 10.35%
7-6 HS_RX_TERM_IMP_CTL	HS_RX_TERM_IMP_CTL S-RX Termination Impedance Control 00b - 98 ohm 01b - 106 ohm 10b - 85 ohm 11b - 91 ohm
5-4 XXX	xxx 00b - - 01b - 30 ps 10b - 60 ps 11b - 90 ps
3-2 HS_RX_DELAY_CTRL	HS_RX_DELAY_CTRL Clock Lane HS RX Delay Control 00b - 0 ps 01b - 30 ps 10b - 60 ps 11b - 90 ps
1-0 HS_RX_BIAS_CTL	HS_RX_BIAS_CTL HS RX Bias Control 00b - 25 uA 01b - 30 uA 10b - 37.5 uA 11b - 50 uA

13.2.3.1.13 Master and Slave DPHY Control High Register (MIPI_S_DPHYCTL_HIGH)

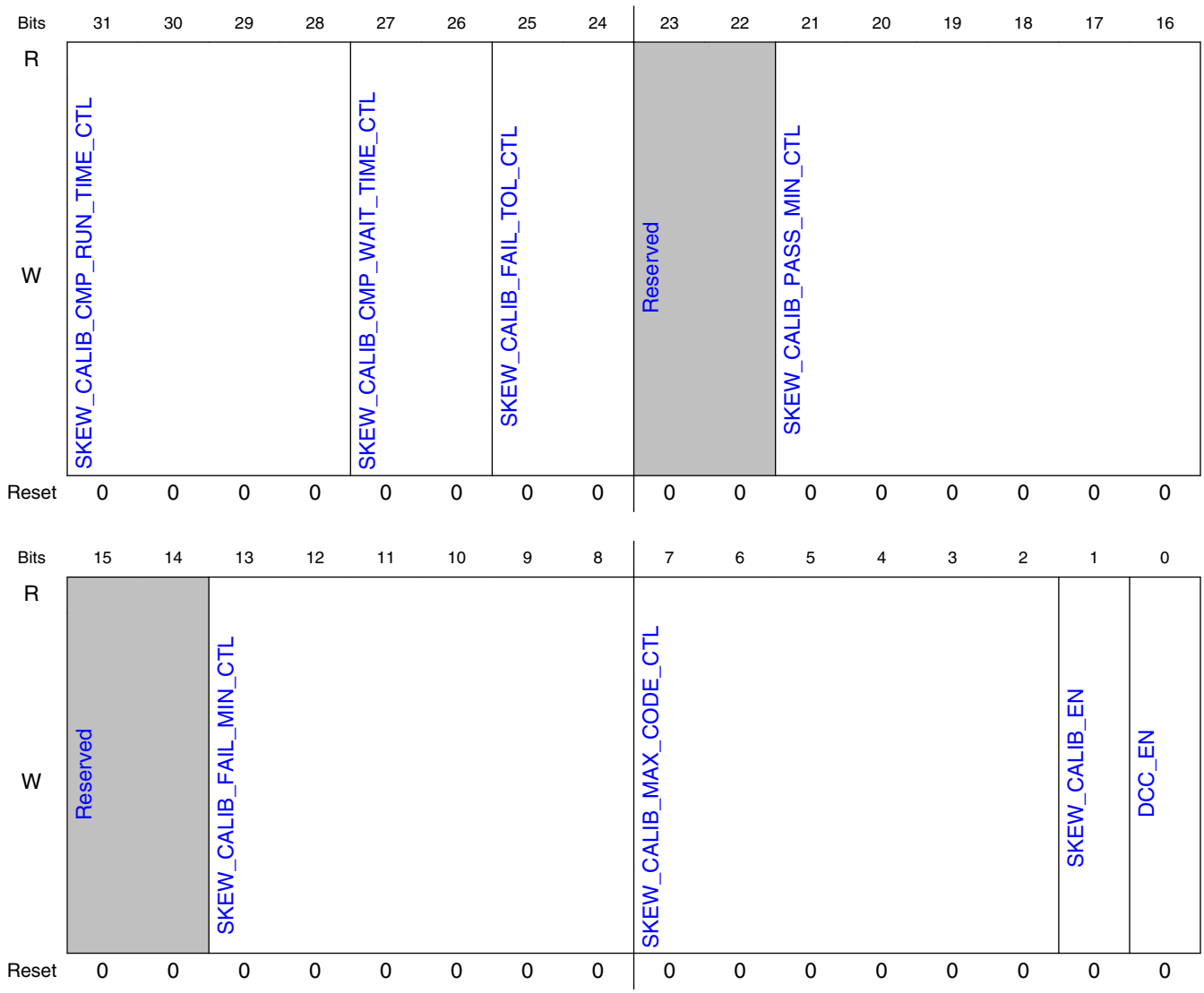
13.2.3.1.13.1 Offset

Register	Offset
MIPI_S_DPHYCTL_HIGH	2Ch

13.2.3.1.13.2 Function

This GPR is the higher 32-bits for the S_DPHYCTL Register

13.2.3.1.13.3 Diagram



13.2.3.1.13.4 Fields

Field	Function
31-28 SKEW_CALIB_CMP_RUN_TIME_CTL	SKEW_CALIB_CMP_RUN_TIME_CTL RX Skew Calibration Compare-run time Control
27-26 SKEW_CALIB_CMP_WAIT_TIME_CTL	SKEW_CALIB_CMP_WAIT_TIME_CTL RX Skew Calibration Compare-wait time Control Control waiting time at changing delay code 00b - Min (Fast) 01b - Reserved

Table continues on the next page...

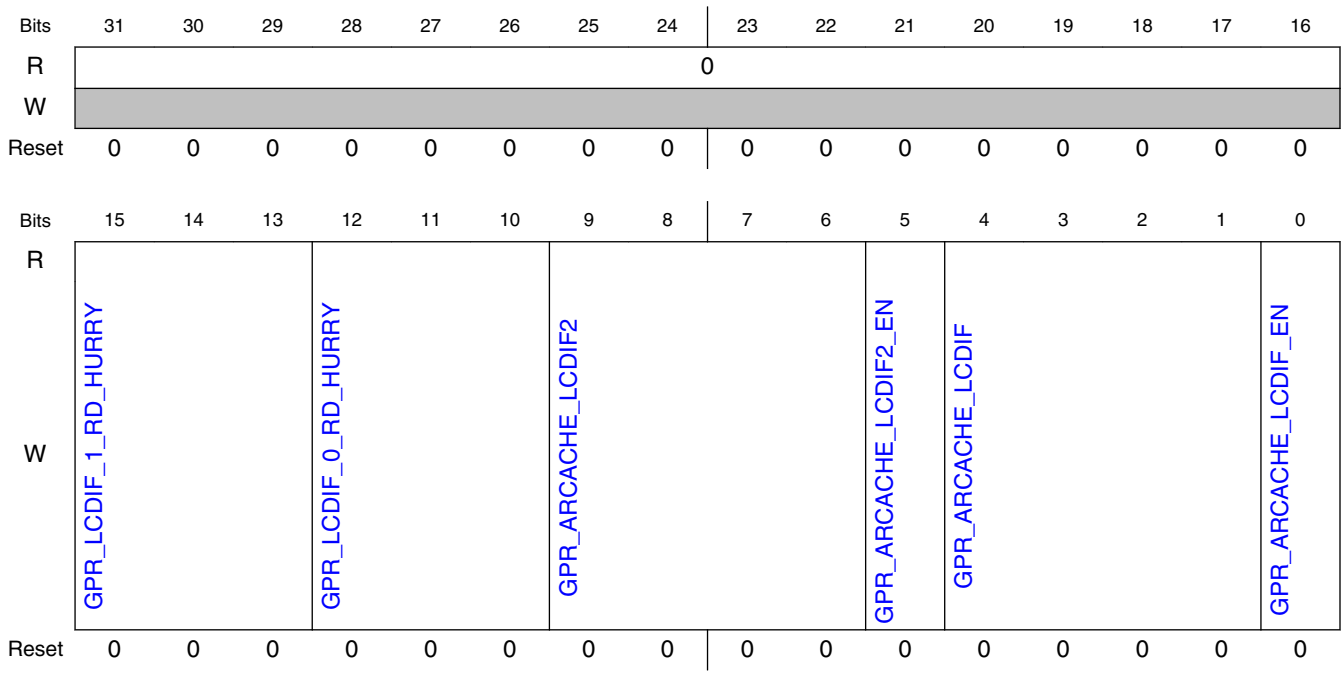
Field	Function
	10b - Reserved 11b - Mx (Slow)
25-24 SKEW_CALIB_FAIL_TOL_CTL	SKEW_CALIB_FAIL_TOL_CTL RX Skew Calibration Fail Tolerance Control 00b - Recognizes the pass although fail appears 3 times 01b - Recognizes the pass although fail appears 2 times 10b - Recognizes the pass although fail appears 1 time 11b - Reserved
23-22 —	- Reserved
21-16 SKEW_CALIB_PASS_MIN_CTL	SKEW_CALIB_PASS_MIN_CTL RX Skew Calibration Pass-min Control
15-14 —	- Reserved
13-8 SKEW_CALIB_FAIL_MIN_CTL	SKEW_CALIB_FAIL_MIN_CTL RX Skew Calibration Fail-min Control
7-2 SKEW_CALIB_MAX_CODE_CTL	SKEW_CALIB_MAX_CODE_CTL RX Skew Calibration Max Code Control
1 SKEW_CALIB_EN	SKEW_CALIB_EN Skew Calibration Function Enable/Disable Control 0b - Skew Calibration Disable 1b - Skew Calibration Enable
0 DCC_EN	DCC_EN DCC Function Enable/Disable 0b - DCC Disable 1b - DCC Enable

13.2.3.1.14 LCDIF ARCACHE Control Register (LCDIF_ARCACHE_CTRL)

13.2.3.1.14.1 Offset

Register	Offset
LCDIF_ARCACHE_CTRL	4Ch

13.2.3.1.14.2 Diagram



13.2.3.1.14.3 Fields

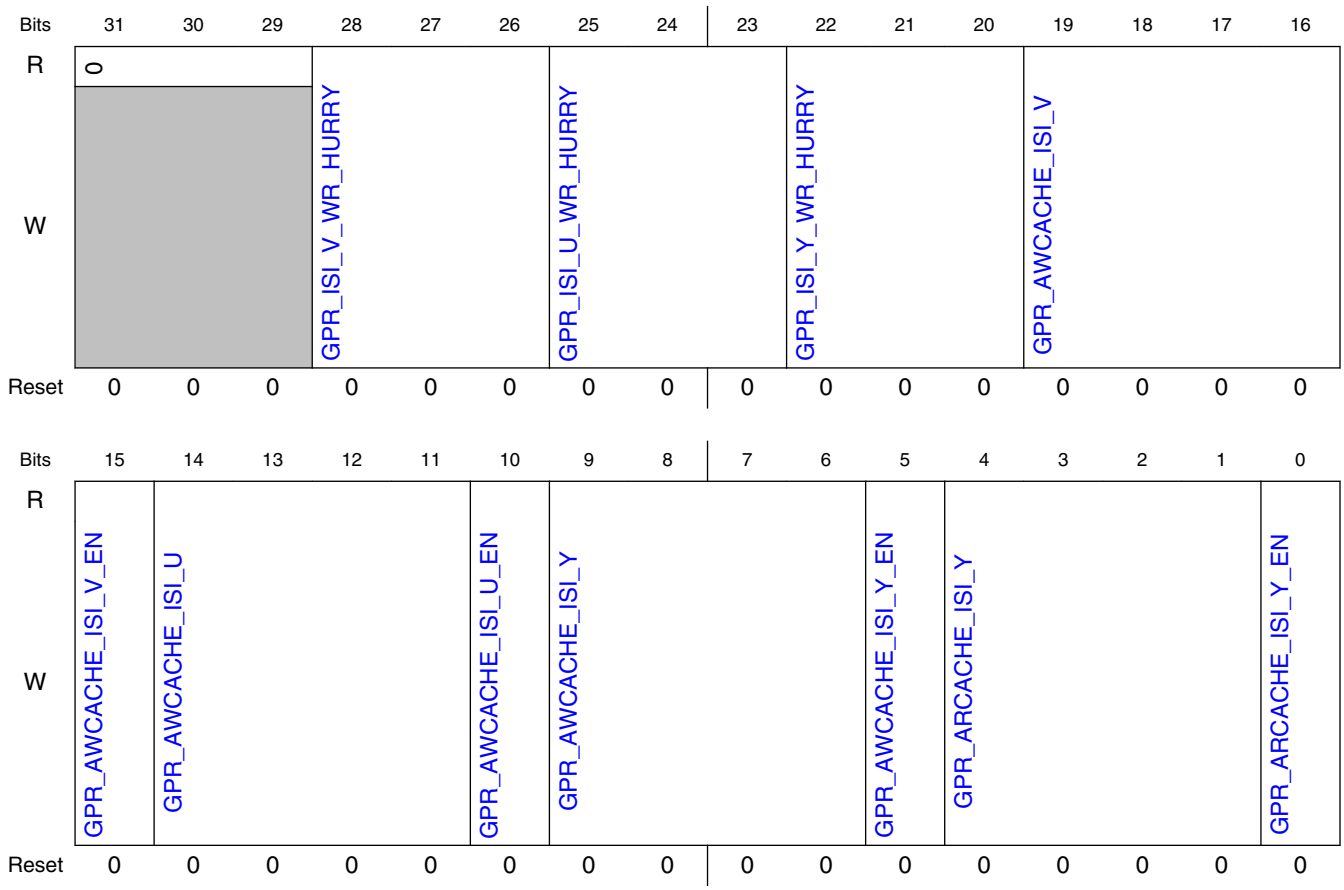
Field	Function
31-16 —	Reserved
15-13 GPR_LCDIF_1_RD_HURRY	GPR_lcdif_1_rd_hurry
12-10 GPR_LCDIF_0_RD_HURRY	GPR_lcdif_0_rd_hurry
9-6 GPR_ARCACHE_LCDIF2	Control LCDIF2 AXI master ARCACHE type
5 GPR_ARCACHE_LCDIF2_EN	AXI master ARCACHE control enable 0b - disable LCDIF2 AXI master ARCACHE control 1b - enable LCDIF2 AXI master ARCACHE control
4-1 GPR_ARCACHE_LCDIF	Control LCDIF AXI master ARCACHE type
0 GPR_ARCACHE_LCDIF_EN	AXI master ARCACHE control enable 0b - disable LCDIF AXI master ARCACHE control 1b - enable LCDIF AXI master ARCACHE control

13.2.3.1.15 ISI CACHE Control Register (ISI_CACHE_CTRL)

13.2.3.1.15.1 Offset

Register	Offset
ISI_CACHE_CTRL	50h

13.2.3.1.15.2 Diagram



13.2.3.1.15.3 Fields

Field	Function
31-29 —	Reserved
28-26	GPR_ISI_v_wr_hurry

Table continues on the next page...

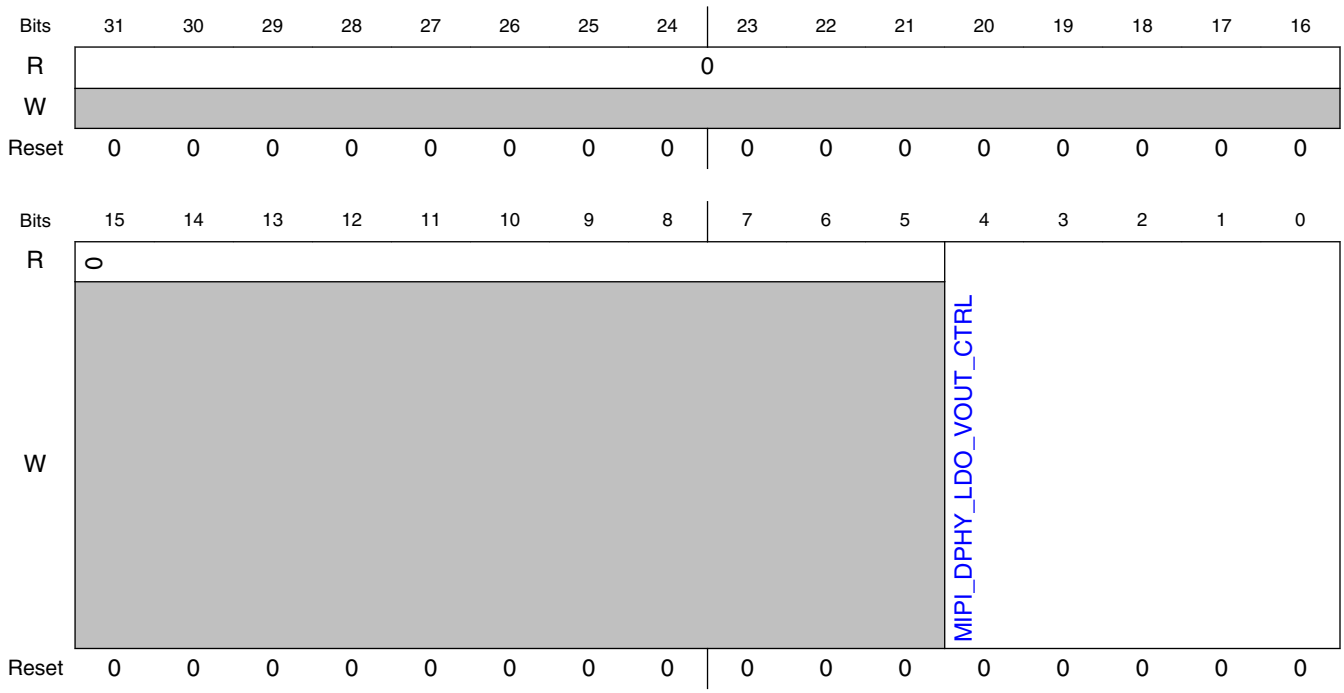
Field	Function
GPR_ISI_V_WR_HURRY	
25-23 GPR_ISI_U_WR_HURRY	GPR_ISI_u_wr_hurry
22-20 GPR_ISI_Y_WR_HURRY	GPR_ISI_y_wr_hurry
19-16 GPR_AWCACHE_ISI_V	Control ISI V channel AXI master AWCACHE type
15 GPR_AWCACHE_ISI_V_EN	ISI V channel AXI master AWCACHE control enable 0b - disable 1b - enable
14-11 GPR_AWCACHE_ISI_U	Control ISI U channel AXI master AWCACHE type
10 GPR_AWCACHE_ISI_U_EN	ISI U channel AXI master AWCACHE control enable 0b - disable 1b - enable
9-6 GPR_AWCACHE_ISI_Y	Control ISI Y channel AXI master AWCACHE type
5 GPR_AWCACHE_ISI_Y_EN	ISI Y channel AXI master AWCACHE control enable 0b - disable 1b - enable
4-1 GPR_ARCACHE_ISI_Y	Control ISI Y channel AXI master ARCACHE type
0 GPR_ARCACHE_ISI_Y_EN	ISI Y channel AXI master ARCACHE control enable 0b - disable 1b - enable

13.2.3.1.16 LDO Control Register (LDO_CTRL)

13.2.3.1.16.1 Offset

Register	Offset
LDO_CTRL	54h

13.2.3.1.16.2 Diagram



13.2.3.1.16.3 Fields

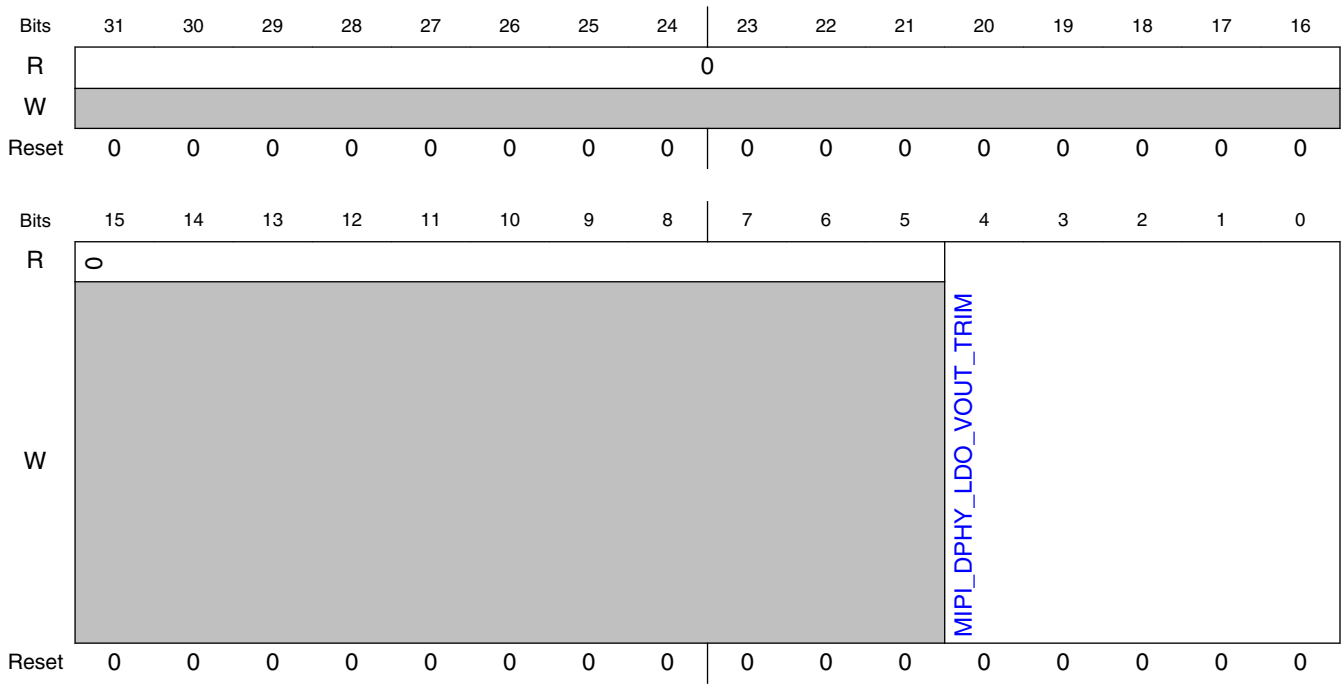
Field	Function
31-5 —	Reserved
4-0 MIPI_DPHY_LDO_VOUT_CTRL	LDO output control port(high level 0.8v,low level 0v)

13.2.3.1.17 LDO Trim Register (LDO_TRIM)

13.2.3.1.17.1 Offset

Register	Offset
LDO_TRIM	58h

13.2.3.1.17.2 Diagram



13.2.3.1.17.3 Fields

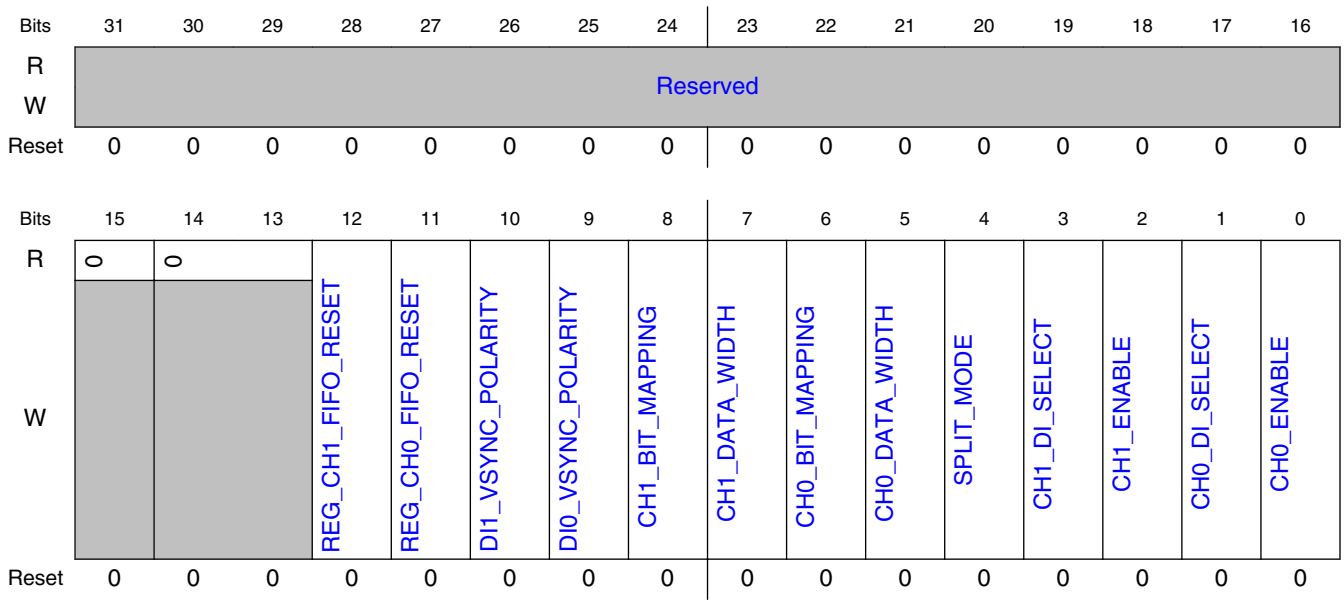
Field	Function
31-5 —	Reserved
4-0 MIPI_DPHY_LDO_VOUT_TRIM	LDO output trimming port(high level 0.8v,low level 0v)

13.2.3.1.18 LDB Control Register (LDB_CTRL)

13.2.3.1.18.1 Offset

Register	Offset
LDB_CTRL	5Ch

13.2.3.1.18.2 Diagram



13.2.3.1.18.3 Fields

Field	Function
31-16 —	Reserved
15 —	Reserved
14-13 —	Reserved
12 REG_CH1_FIFO_RESET	reg_ch1_fifo_reset SW reset bit for the ch1 async FIFO
11 REG_CH0_FIFO_RESET	reg_ch0_fifo_reset SW reset bit for the ch0 async FIFO
10 DI1_VSYNC_POLARITY	di1 VSYNC polarity select 0b - negative polarity 1b - positive polarity
9 DI0_VSYNC_POLARITY	di0 VSYNC polarity select 0b - negative polarity 1b - positive polarity
8 CH1_BIT_MAPPING	ch1_bit_mapping 0b - SPWG mapping 1b - JEIDA mapping

Table continues on the next page...

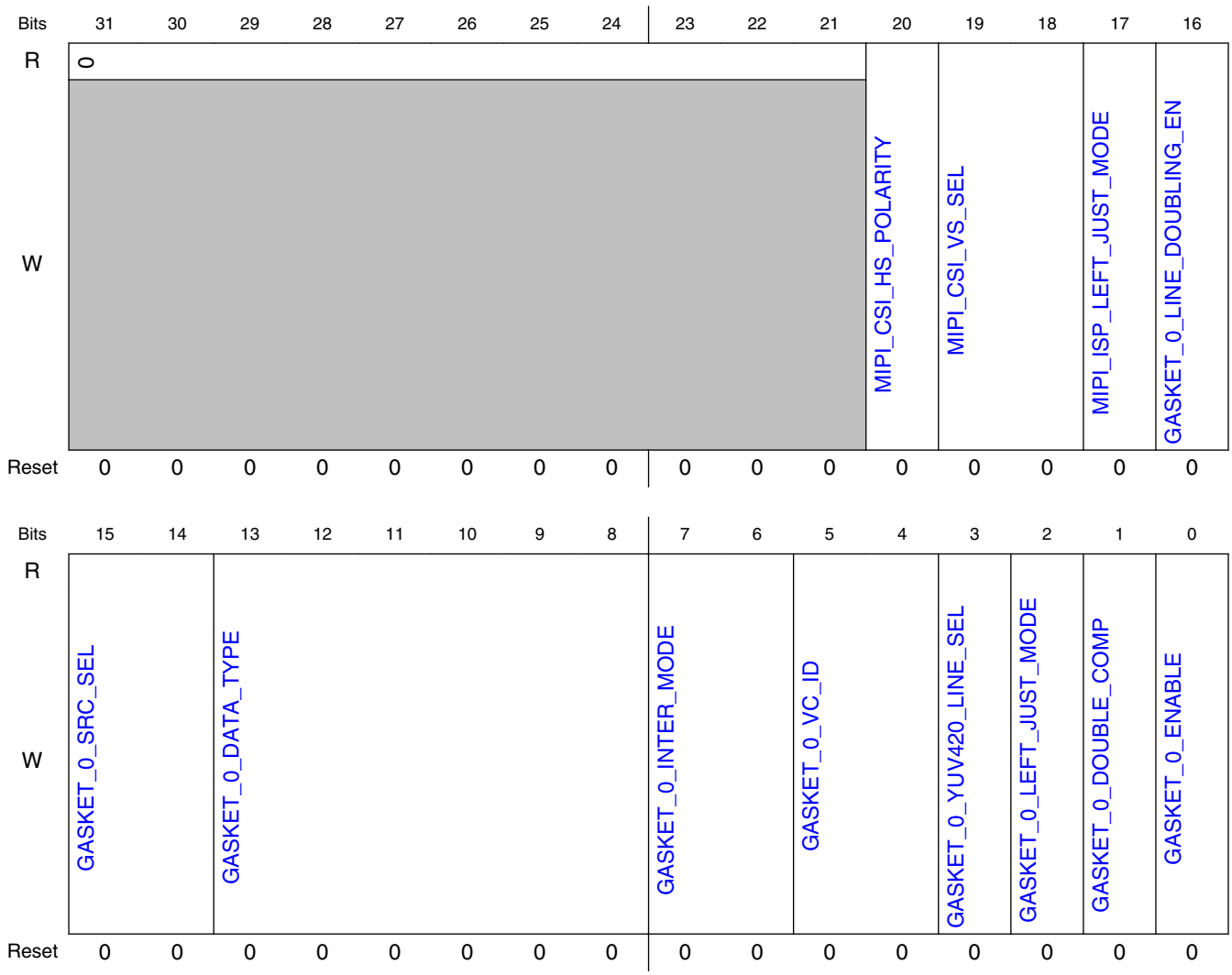
Field	Function
7 CH1_DATA_WIDTH	ch1_data_width
6 CH0_BIT_MAPPING	ch0_bit_mapping 0b - SPWG mapping 1b - JEIDA mapping
5 CH0_DATA_WIDTH	ch0_data_width 0b - 18bits 1b - 24bits
4 SPLIT_MODE	split_mode
3 CH1_DI_SELECT	ch1_di_select ch1 source select 0b - LDB data from source 0 1b - LDB data from source 1
2 CH1_ENABLE	ch1_enable
1 CH0_DI_SELECT	ch0_di_select 0b - LDB data from source 0 1b - LDB data from source 1
0 CH0_ENABLE	ch0_enable

13.2.3.1.19 Gasket 0 Control Register (GASKET_0_CTRL)

13.2.3.1.19.1 Offset

Register	Offset
GASKET_0_CTRL	60h

13.2.3.1.19.2 Diagram



13.2.3.1.19.3 Fields

Field	Function
31-21 —	Reserved
20 MIPI_CSI_HS_POLARITY	mipi_csi_hs_polarity
19-18 MIPI_CSI_VS_SEL	mipi_csi_vs_sel
17	mipi_isp_left_just_mode

Table continues on the next page...

Field	Function
MIPI_ISP_LEFT_JUST_MODE	
16 GASKET_0_LINE_DOUBLING_EN	Gasket 0 output for ISI de-interlace line_doubling mode enable 0b - Gasket 0 not output for ISI de-interlace line_doubling mode 1b - Gasket 0 output for ISI de-interlace line_doubling mode
15-14 GASKET_0_SRC_SEL	Gasket 0 source when support ISI de-interlace line_doubling mode 00b - source from mipi_csi channel 0 01b - source from mipi_csi channel 1 10b - source from mipi_csi channel 2 11b - source from mipi_csi channel 3
13-8 GASKET_0_DATA_TYPE	Gasket 0 data type Note: Need support MIPI_CSI user-defined data type. 011000b - YUV420 8-bit 011001b - YUV420 10-bit 011010b - Legacy YUV420 8-bit 011100b - YUV420 8-bit(Chroma Shifted Pixel Sampling) 011101b - YUV420 10-bit(Chroma Shifted Pixel Sampling) 011110b - YUV422 8-bit 011111b - YUV422 10-bit 100010b - RGB565 100011b - RGB666 100100b - RGB888 101000b - RAW6 101001b - RAW7 101010b - RAW8 101011b - RAW10 101100b - RAW12 101101b - RAW14
7-6 GASKET_0_INTER_MODE	Gasket 0 interlace mode 00b - not interlaced 01b - interlaced left 10b - interlaced right 11b - reserved
5-4 GASKET_0_VC_ID	Gasket 0 Virtual channel identifier, In non-interlaced mode, each ISI process channel has fixed input from MIPI_CSI. In interlaced mode (line doubling deinterlaced mode), maybe different ISI process channel has same input from MIPI_CSI.
3 GASKET_0_YUV420_LINE_SEL	Gasket 0 YUV420 ODD/EVEN line first select 0b - Gasket 0 ODD line first for YUV420 data type 1b - Gasket 0 EVEN line first for YUV420 data type
2 GASKET_0_LEFT_JUST_MODE	Gasket 0 Left justified mode 0b - unused LSB equal lease significant bit of valid data 1b - unused LSB equal most significant bit of valid data
1 GASKET_0_DOUBLE_COMP	Gasket 0 double component enable 0b - Gasket 0 input single component per pixel clock for YUV422 1b - Gasket 0 input double component per pixel clock for YUV422
0	Gasket 0 enable 0b - Gasket output disable 1b - Gasket 0 output enable

Field	Function
GASKET_0_ENABLE	

13.2.3.1.20 Gasket 0 Video Horizontal Size Register (GASKET_0_HSIZE)

13.2.3.1.20.1 Offset

Register	Offset
GASKET_0_HSIZE	64h

13.2.3.1.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_0_HSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_0_HSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.20.3 Fields

Field	Function
31-0 GASKET_0_HSIZE	Gasket 0 video Horizontal size(count in pixel)

13.2.3.1.21 Gasket 0 Video Vertical Size Register (GASKET_0_VSIZE)

13.2.3.1.21.1 Offset

Register	Offset
GASKET_0_VSIZE	68h

13.2.3.1.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_0_VSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_0_VSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.21.3 Fields

Field	Function
31-0 GASKET_0_VSIZE	Gasket 0 video Vertical size(count in line)

13.2.3.1.22 Gasket 0 Video Horizontal Front Porch Register (GASKET_0_HFP)

13.2.3.1.22.1 Offset

Register	Offset
GASKET_0_HFP	6Ch

13.2.3.1.22.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_0_HFP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_0_HFP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.22.3 Fields

Field	Function
31-0 GASKET_0_HFP	Gasket 0 video Horizontal front porch(count in pixel)

13.2.3.1.23 Gasket 0 Video Horizontal Back Porch Register (GASKET_0_HBP)

13.2.3.1.23.1 Offset

Register	Offset
GASKET_0_HBP	70h

13.2.3.1.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_0_HBP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_0_HBP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.23.3 Fields

Field	Function
31-0 GASKET_0_HBP	Gasket 0 video Horizontal back porch(count in pixel)

13.2.3.1.24 Gasket 0 Video Vertical Front Porch Register (GASKET_0_VFP)

13.2.3.1.24.1 Offset

Register	Offset
GASKET_0_VFP	74h

13.2.3.1.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_0_VFP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_0_VFP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.24.3 Fields

Field	Function
31-0 GASKET_0_VFP	Gasket 0 video Vertical front porch(count in line)

13.2.3.1.25 Gasket 0 Video Vertical Back Porch Register (GASKET_0_VBP)

13.2.3.1.25.1 Offset

Register	Offset
GASKET_0_VBP	78h

13.2.3.1.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_0_VBP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_0_VBP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.25.3 Fields

Field	Function
31-0 GASKET_0_VBP	Gasket 0 video Vertical back porch(count in line)

13.2.3.1.26 Gasket 0 ISI Pixel Count Register (GASKET_0_ISI_PIXEL_CNT)

13.2.3.1.26.1 Offset

Register	Offset
GASKET_0_ISI_PIXEL_CNT	7Ch

13.2.3.1.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_0_ISI_PIXEL_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_0_ISI_PIXEL_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.26.3 Fields

Field	Function
31-0 GASKET_0_ISI_PIXEL_CNT	Gasket 0 output to ISI pixel count status

13.2.3.1.27 Gasket 0 ISI Line Count Register (GASKET_0_ISI_LINE_CNT)

13.2.3.1.27.1 Offset

Register	Offset
GASKET_0_ISI_LINE_CNT	80h

13.2.3.1.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_0_ISI_LINE_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_0_ISI_LINE_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.27.3 Fields

Field	Function
31-0 GASKET_0_ISI_LINE_CNT	Gasket 0 output to ISI line count status

13.2.3.1.28 Gasket 0 ISI Pixel Control Information Register (GASKET_0_ISI_PIXEL_CTRL)

13.2.3.1.28.1 Offset

Register	Offset
GASKET_0_ISI_PIXEL_CTRL	84h

13.2.3.1.28.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				GASKET_0_ISI_PIXEL_CTRL											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.28.3 Fields

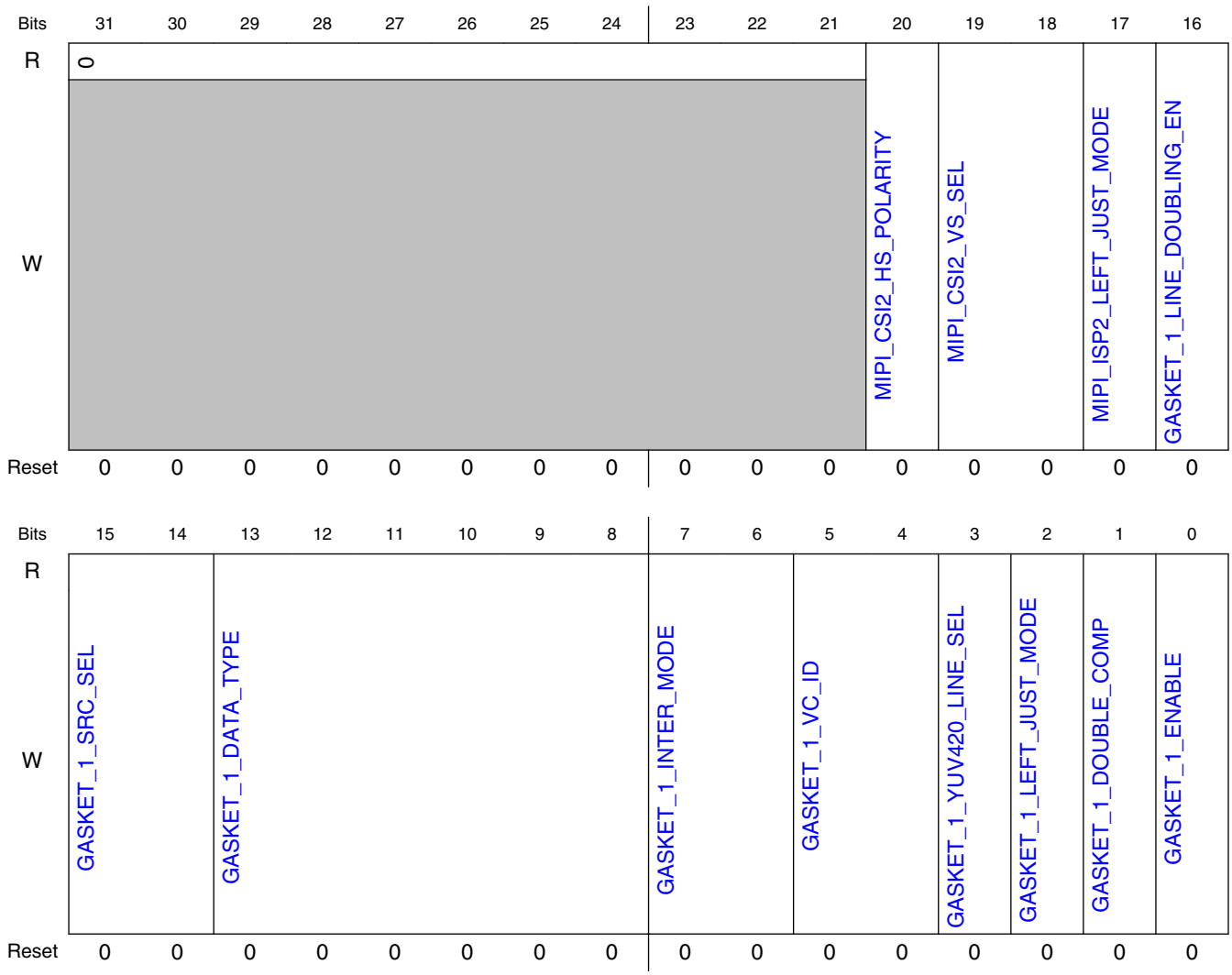
Field	Function
31-12 —	Reserved
11-0 GASKET_0_ISI_PIXEL_CTRL	Gasket 0 output to ISI pixel control information status

13.2.3.1.29 Gasket 1 Control Register (GASKET_1_CTRL)

13.2.3.1.29.1 Offset

Register	Offset
GASKET_1_CTRL	90h

13.2.3.1.29.2 Diagram



13.2.3.1.29.3 Fields

Field	Function
31-21 —	Reserved
20 MIPI_CSI2_HS_POLARITY	mipi_csi2_hs_polarity
19-18 MIPI_CSI2_VS_SEL	mipi_csi2_vs_sel
17	mipi_isp2_left_just_mode

Table continues on the next page...

Field	Function
MIPI_ISP2_LEFT_JUST_MODE	
16 GASKET_1_LINE_DOUBLING_EN	Gasket 1 output for ISI de-interlace line_doubling mode enable 0b - Gasket 0 not output for ISI de-interlace line_doubling mode 1b - Gasket 1 output for ISI de-interlace line_doubling mode
15-14 GASKET_1_SRC_SEL	Gasket 1 source when support ISI de-interlace line_doubling mode 00b - source from mipi_csi channel 0 01b - source from mipi_csi channel 1 10b - source from mipi_csi channel 2 11b - source from mipi_csi channel 3
13-8 GASKET_1_DATA_TYPE	Gasket 1 data type Note: need support MIPI_CSI user-defined data type. 011000b - YUV420 8-bit 011001b - YUV420 10-bit 011010b - Legacy YUV420 8-bit 011100b - YUV420 8-bit(Chroma Shifted Pixel Sampling) 011101b - YUV420 10-bit(Chroma Shifted Pixel Sampling) 011110b - YUV422 8-bit 011111b - YUV422 10-bit 100010b - RGB565 100011b - RGB666 100100b - RGB888 101000b - RAW6 101001b - RAW7 101010b - RAW8 101011b - RAW10 101100b - RAW12 101101b - RAW14
7-6 GASKET_1_INTER_MODE	Gasket 1 interlace mode 00b - not interlaced 01b - interlaced left 10b - interlaced right 11b - reserved
5-4 GASKET_1_VC_ID	Gasket 1 Virtual channel identifier, In non-interlaced mode, each ISI process channel have fixed input from MIPI_CSI. In interlaced mode (line doubling deinterlaced mode), maybe different ISI process channel have same input from. MIPI_CSI.
3 GASKET_1_YUV420_LINE_SEL	Gasket 1 YUV420 ODD/EVEN line first select 0b - Gasket 1 ODD line first for YUV420 data type 1b - Gasket 1 EVEN line first for YUV420 data type
2 GASKET_1_LEFT_JUST_MODE	Gasket 1 Left justified mode 0b - unused LSB equal lease significant bit of valid data 1b - unused LSB equal most significant bit of valid data
1 GASKET_1_DOUBLE_COMP	Gasket 1 double component enable 0b - Gasket 1 input single component per pixel clock for YUV422 1b - Gasket 1 input double component per pixel clock for YUV422
0	Gasket 1 enable 0b - Gasket output disable 1b - Gasket 1 output enable;

Field	Function
GASKET_1_ENABLE	

13.2.3.1.30 Gasket 1 Video Horizontal Size Register (GASKET_1_HSIZE)

13.2.3.1.30.1 Offset

Register	Offset
GASKET_1_HSIZE	94h

13.2.3.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.30.3 Fields

Field	Function
31-0 GP0	Gasket 1 video Horizontal size(count in pixel)

13.2.3.1.31 Gasket 1 Video Vertical Size Register (GASKET_1_VSIZE)

13.2.3.1.31.1 Offset

Register	Offset
GASKET_1_VSIZE	98h

13.2.3.1.31.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_1_VSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_1_VSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.31.3 Fields

Field	Function
31-0 GASKET_1_VSIZE	Gasket 1 video Vertical size(count in line)

13.2.3.1.32 Gasket 1 Video Horizontal Front Porch Register (GASKET_1_HFP)

13.2.3.1.32.1 Offset

Register	Offset
GASKET_1_HFP	9Ch

13.2.3.1.32.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_1_HFP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_1_HFP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.32.3 Fields

Field	Function
31-0 GASKET_1_HFP	Gasket 1 video Horizontal front porch(count in pixel)

13.2.3.1.33 Gasket 1 Video Horizontal Back Porch Register (GASKET_1_HBP)

13.2.3.1.33.1 Offset

Register	Offset
GASKET_1_HBP	A0h

13.2.3.1.33.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_1_HBP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_1_HBP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.33.3 Fields

Field	Function
31-0 GASKET_1_HBP	Gasket 1 video Horizontal back porch(count in pixel)

13.2.3.1.34 Gasket 1 Video Vertical Front Porch Register (GASKET_1_VFP)

13.2.3.1.34.1 Offset

Register	Offset
GASKET_1_VFP	A4h

13.2.3.1.34.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_1_VFP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_1_VFP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.34.3 Fields

Field	Function
31-0 GASKET_1_VFP	Gasket 1 video Vertical front porch(count in line)

13.2.3.1.35 Gasket 1 Video Vertical Back Porch Register (GASKET_1_VBP)

13.2.3.1.35.1 Offset

Register	Offset
GASKET_1_VBP	A8h

13.2.3.1.35.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_1_VBP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_1_VBP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.35.3 Fields

Field	Function
31-0 GASKET_1_VBP	Gasket 1 video Vertical back porch(count in line)

13.2.3.1.36 Gasket 1 ISI Pixel Count Register (GASKET_1_ISI_PIXEL_CNT)

13.2.3.1.36.1 Offset

Register	Offset
GASKET_1_ISI_PIXEL_CNT	ACh

13.2.3.1.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_1_ISI_PIXEL_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_1_ISI_PIXEL_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.36.3 Fields

Field	Function
31-0 GASKET_1_ISI_PIXEL_CNT	Gasket 1 output to ISI pixel count status

13.2.3.1.37 Gasket 1 ISI Line Count Register (GASKET_1_ISI_LINE_CNT)

13.2.3.1.37.1 Offset

Register	Offset
GASKET_1_ISI_LINE_CNT	B0h

13.2.3.1.37.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GASKET_1_ISI_LINE_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GASKET_1_ISI_LINE_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.37.3 Fields

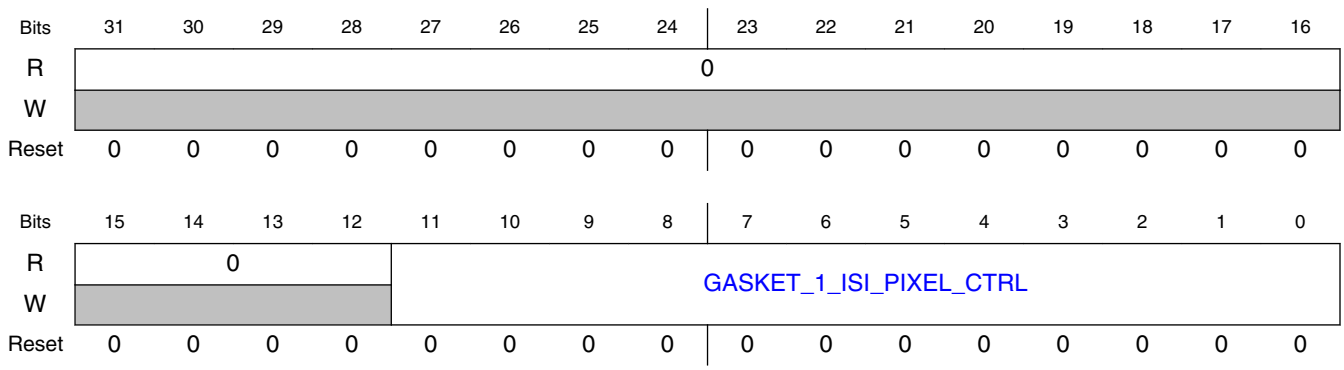
Field	Function
31-0 GASKET_1_ISI_LINE_CNT	Gasket 1 output to ISI line count status

13.2.3.1.38 Gasket 1 ISI Pixel Control Information Register (GASKET_1_ISI_PIXEL_CTRL)

13.2.3.1.38.1 Offset

Register	Offset
GASKET_1_ISI_PIXEL_CTRL	B4h

13.2.3.1.38.2 Diagram



13.2.3.1.38.3 Fields

Field	Function
31-12 —	Reserved
11-0 GASKET_1_ISI_PIXEL_CTRL	Gasket 1 output to ISI pixel control information status

13.2.3.1.39 Master and Slave DPHY Control Low Register (MIPI_B2_DPHYCTL_LOW)

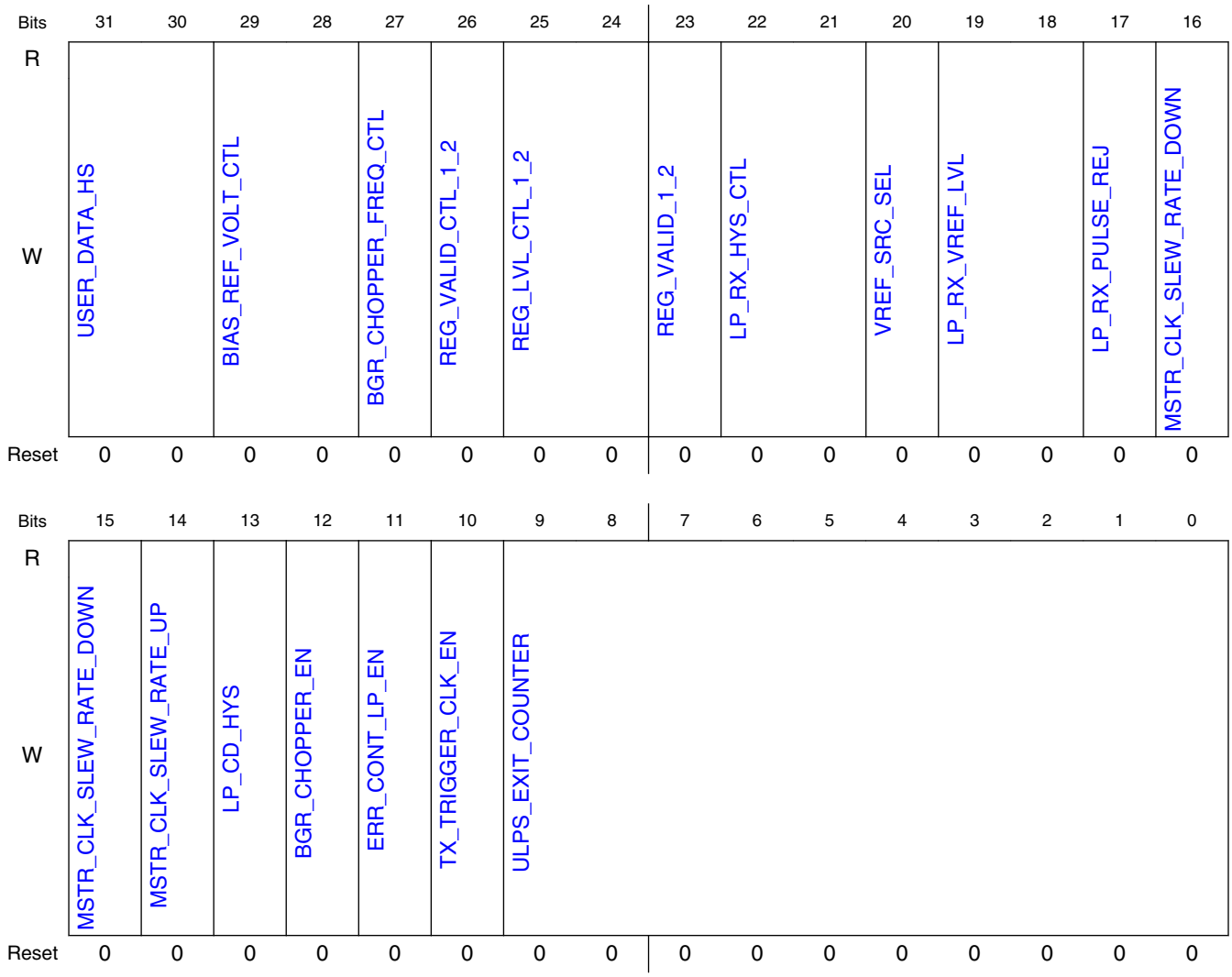
13.2.3.1.39.1 Offset

Register	Offset
MIPI_B2_DPHYCTL_LOW	120h

13.2.3.1.39.2 Function

This GPR is the lower 32-bits for the B2_DPHYCTL Register

13.2.3.1.39.3 Diagram



13.2.3.1.39.4 Fields

Field	Function
31-30 USER_DATA_HS	User Data Pattern for HS Loopback mode Lower 2-bit of USER_DATA_HS. Can be 2'b00 to 2'b11. Please also refer to MIPI_B_DPHYCTL_HIGH[USER_DATA_HS] for the higher 6-bits.
29-28	BIAS_REF_VOLT_CTL Bias Reference Voltage 710m Control

Table continues on the next page...

Field	Function
BIAS_REF_VOLT_CTL	00b - 712mV 01b - 724mV 10b - 733mV 11b - 706mV
27 BGR_CHOPPER_FREQ_CTL	BGR_CHOPPER_FREQ_CTL BGR Chopper Frequency Control 0b - 3MHz 1b - 1.5MHz
26 REG_VALID_CTL_1_2	REG_VALID_CTL_1_2 VREG12_EXTPWR Enable Control 0b - Internal 1.2V regulator 1b - External 1.2V power
25-24 REG_LVL_CTL_1_2	REG_LVL_CTL_1_2 1.2 Regulator Level Control 00b - 1.2V 01b - 1.23V 10b - 1.27V 11b - 1.26V
23 REG_VALID_1_2	REG_VALID_1_2 1.2 Regulator Valid Level Section 0b - Use "ulps_en" signal 1b - Use valid signal from 1.2V regulator
22-21 LP_RX_HYS_CTL	LP_RX_HYS_CTL LP-RX Hysteresis Level Control 00b - 80mV 01b - 100mV 10b - 120mV 11b - 140mV
20 VREF_SRC_SEL	VREF_SRC_SEL VREF Source Selection 0b - Generated from the BGR 1b - Generated from the current mirror
19-18 LP_RX_VREF_LVL	LP_RX_VREF_LVL LP-RX VREF Level Control 00b - 715mV 01b - 743mV 10b - 650mV 11b - 682mV
17 LP_RX_PULSE_REJ	LP_RX_PULSE_REJ LP-RX Pulse Rejection Control 0b - Enable 1b - Disable
16-15	MSTR_CLK_SLEW_RATE_DOWN Master Clock Lane's LP-TX Driver Slew Rate Down Control 00b - No change

Table continues on the next page...

Field	Function
MSTR_CLK_SLEW_RATE_DOWN	01b - Decrease the slew rate by about 15% 10b - Decrease the slew rate by about 15% 11b - Decrease the slew rate by about 30%
14 MSTR_CLK_SLEW_RATE_UP	MSTR_CLK_SLEW_RATE_UP Master Clock Lane's LP-TX Driver Slew Rate Up Control 0b - No change 1b - Slew Rate UP
13 LP_CD_HYS	LP_CD_HYS LP-CD Hysteresis Level Control 0b - 60mV 1b - 70mV
12 BGR_CHOPPER_EN	BGR_CHOPPER_EN BGR Chopper Function Enable/Disable Control 0b - Enable 1b - Disable
11 ERR_CONT_LP_EN	ERR_CONT_LP_EN Error Contention LP Enable/Disable Control 0b - Enable 1b - Disable
10 TX_TRIGGER_CLK_EN	TX_TRIGGER_CLK_EN TX Trigger Clk Enable/Disable 0b - Enable 1b - Disable
9-0 ULPS_EXIT_COUNTER	ULPS_EXIT_COUNTER ULPS EXIT Counter value Control You should set B_DPHYCTL[9:0] during initial or power-up sequence. 0000000000b - 0.01 MHz 0000000011b - 0.10 MHz 0000011001b - 1.00 MHz 0000110010b - 2.00 MHz 0001001011b - 3.00 MHz 0001100100b - 4.00 MHz 0001111101b - 5.00 MHz 0010010110b - 6.00 MHz 0010101111b - 7.00 MHz 0011001000b - 8.00 MHz 0011100001b - 9.00 MHz 0011111010b - 10.00 MHz 0100010011b - 11.00 MHz 0100101100b - 12.00 MHz 0101000101b - 13.00 MHz 0101011110b - 14.00 MHz 0101110111b - 15.00 MHz 0110010000b - 16.00 MHz 0110101001b - 17.00 MHz 0111000010b - 18.00 MHz 0111011011b - 19.00 MHz 0111110100b - 20.00 MHz

13.2.3.1.40 Master and Slave DPHY Control High Register (MIPI_B2_DPHYCTL_HIGH)

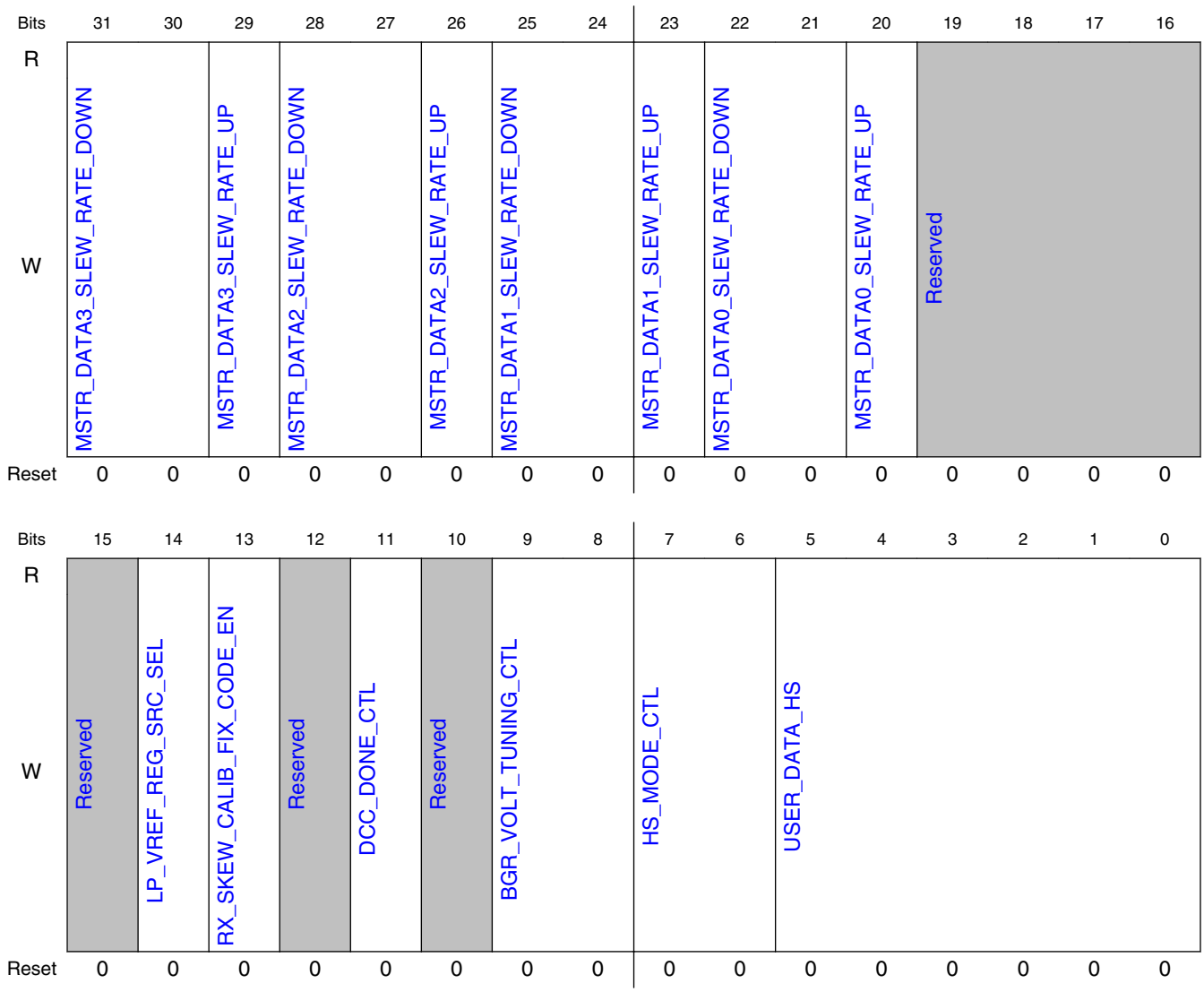
13.2.3.1.40.1 Offset

Register	Offset
MIPI_B2_DPHYCTL_HIGH	124h

13.2.3.1.40.2 Function

This GPR is the higher 32-bits for the B2_DPHYCTL Register

13.2.3.1.40.3 Diagram



13.2.3.1.40.4 Fields

Field	Function
31-30 MSTR_DATA3_SLEW_RATE_DOWN	MSTR_DATA3_SLEW_RATE_DOWN Master Data3 Lane's LP-TX Driver Slew Rate Down Control 00b - No change 01b - Decrease the slew rate by about 15% 10b - Decrease the slew rate by about 15% 11b - Decrease the slew rate by about 30%
29	MSTR_DATA3_SLEW_RATE_UP Master Data3 Lane's LP-TX Driver Slew Rate Up Control 0b - No change

Table continues on the next page...

Field	Function
MSTR_DATA3_SLEW_RATE_UP	1b - Slew Rate UP
28-27 MSTR_DATA2_SLEW_RATE_DOWN	MSTR_DATA2_SLEW_RATE_DOWN Master Data2 Lane's LP-TX Driver Slew Rate Down Control 00b - No change 01b - Decrease the slew rate by about 15% 10b - Decrease the slew rate by about 15% 11b - Decrease the slew rate by about 30%
26 MSTR_DATA2_SLEW_RATE_UP	MSTR_DATA2_SLEW_RATE_UP Master Data2 Lane's LP-TX Driver Slew Rate Up Control 0b - No change 1b - Slew Rate UP
25-24 MSTR_DATA1_SLEW_RATE_DOWN	MSTR_DATA1_SLEW_RATE_DOWN Master Data1 Lane's LP-TX Driver Slew Rate Down Control 00b - No change 01b - Decrease the slew rate by about 15% 10b - Decrease the slew rate by about 15% 11b - Decrease the slew rate by about 30%
23 MSTR_DATA1_SLEW_RATE_UP	MSTR_DATA1_SLEW_RATE_UP Master Data1 Lane's LP-TX Driver Slew Rate Up Control 0b - No change 1b - Slew Rate UP
22-21 MSTR_DATA0_SLEW_RATE_DOWN	MSTR_DATA0_SLEW_RATE_DOWN Master Data0 Lane's LP-TX Driver Slew Rate Down Control 00b - No change 01b - Decrease the slew rate by about 15% 10b - Decrease the slew rate by about 15% 11b - Decrease the slew rate by about 30%
20 MSTR_DATA0_SLEW_RATE_UP	MSTR_DATA0_SLEW_RATE_UP Master Data0 Lane's LP-TX Driver Slew Rate Up Control 0b - No change 1b - Slew Rate UP
19-15 —	- -
14 LP_VREF_REG_SRC_SEL	LP_VREF_REG_SRC_SEL LP Regulator Vref Source Selection 0b - Generated from BGR 1b - Generated from Current Mirror
13 RX_SKEW_CALIB_FIX_CODE_EN	RX_SKEW_CALIB_FIX_CODE_EN RX Skew Calibration Fixing Code Enable/Disable Control 0b - Disable 1b - Enable
12	-

Table continues on the next page...

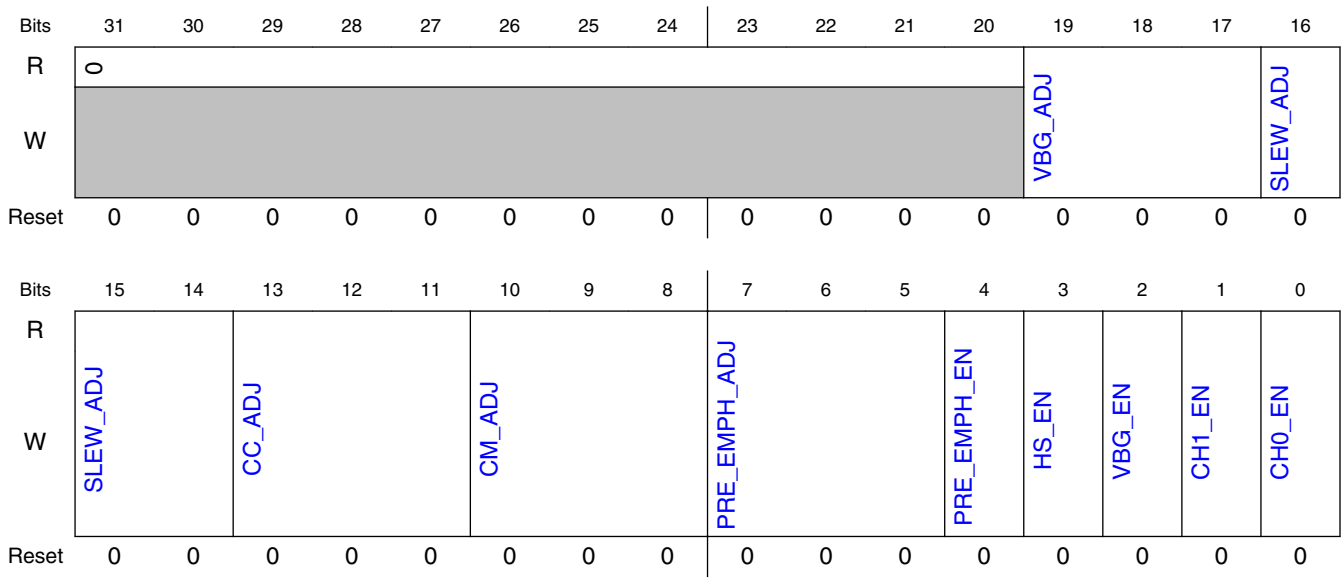
Field	Function
—	-
11 DCC_DONE_CTL	DCC_DONE_CTL DCC "DONE" Signal Control 0b - "DONE" from DCC block 1b - U"DONE" is always 1
10 —	- -
9-8 BGR_VOLT_TUNING_CTL	BGR_VOLT_TUNING_CTL BGR Voltage Tuning Control 00b - 820mV 01b - 760mV 10b - 800mV 11b - 840mV
7-6 HS_MODE_CTL	HS_MODE_CTL HS Loopback Mode Control 00b - Designated Pattern 01b - PRBS7 10b - All zero 11b - User Data Pattern
5-0 USER_DATA_HS	User Data Pattern for HS Loopback mode Higher 6-bit of USER_DATA_HS. Can be 6'b000000 to 6'b111111. Please also refer to MIPI_B_DPHYCTL_HIGH[USER_DATA_HS] for the lower 2-bits.

13.2.3.1.41 LVDS Control Register (LVDS_CTRL)

13.2.3.1.41.1 Offset

Register	Offset
LVDS_CTRL	128h

13.2.3.1.41.2 Diagram



13.2.3.1.41.3 Fields

Field	Function
31-20 —	Reserved
19-17 VBG_ADJ	Bandgap adjustment.
16-14 SLEW_ADJ	Output transition time adjustment.
13-11 CC_ADJ	Output current adjustment.
10-8 CM_ADJ	Output common mode(Vos) adjustment.
7-5 PRE_EMPH_ADJ	Pre-emphasis adjustment.
4 PRE_EMPH_EN	Enable pre-emphasis
3 HS_EN	hs_en 1b - enable the 100 Ohm terminated resistor in the chip, at the same time, the power dissipation will also be double.
2 VBG_EN	Bandgap enable.
1	channel1 enable

Table continues on the next page...

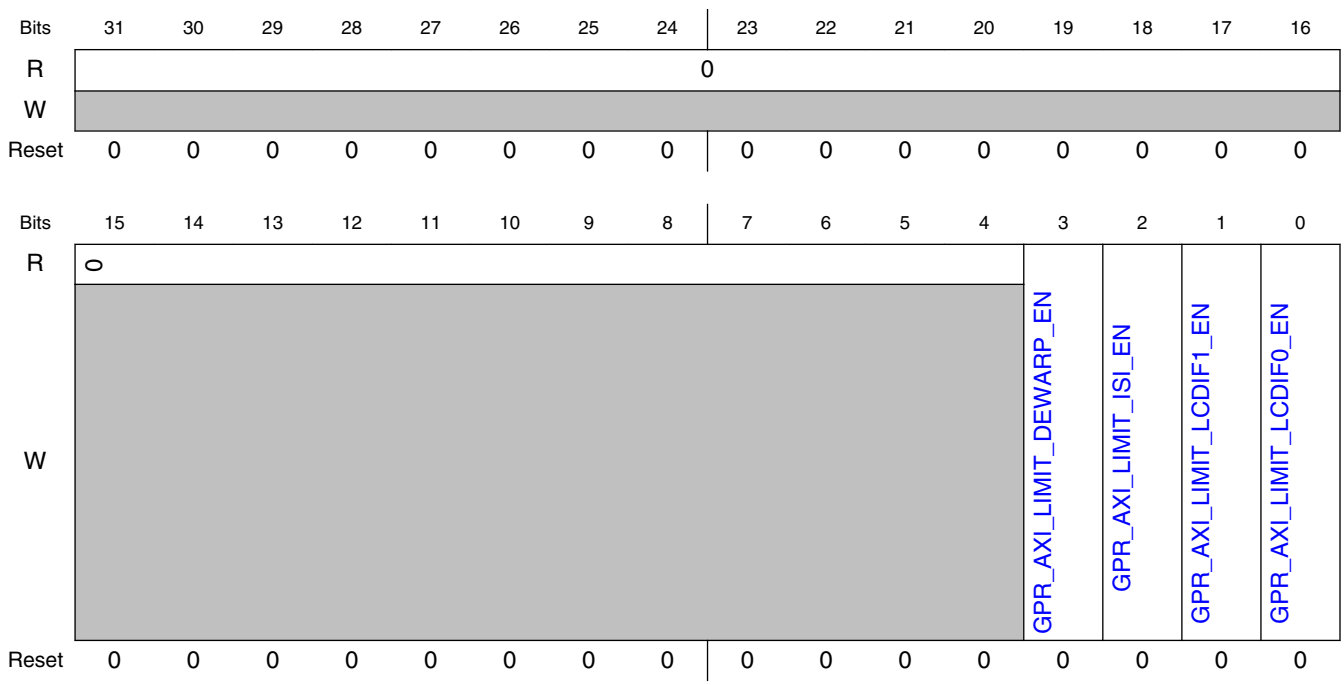
Field	Function
CH1_EN	
0 CH0_EN	channel0 enable

13.2.3.1.42 AXI Limit Control Register (AXI_LIMIT_CONTROL)

13.2.3.1.42.1 Offset

Register	Offset
AXI_LIMIT_CONTROL	12Ch

13.2.3.1.42.2 Diagram



13.2.3.1.42.3 Fields

Field	Function
31-4 —	Reserved
3	gpr_axi_limit_dewarp_en

Table continues on the next page...

Field	Function
GPR_AXI_LIMIT_DEWARP_EN	
2 GPR_AXI_LIMIT_ISI_EN	gpr_axi_limit_isi_en
1 GPR_AXI_LIMIT_LCDIF1_EN	gpr_axi_limit_lcdif1_en
0 GPR_AXI_LIMIT_LCDIF0_EN	gpr_axi_limit_lcdif0_en

13.2.3.1.43 AXI Limit Threshold Register 0 (AXI_LIMIT_THRESH0)

13.2.3.1.43.1 Offset

Register	Offset
AXI_LIMIT_THRESH0	130h

13.2.3.1.43.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR_AXI_LIMIT_LCDIF1_THRESH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR_AXI_LIMIT_LCDIF0_THRESH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.43.3 Fields

Field	Function
31-16 GPR_AXI_LIMIT_LCDIF1_THRESH	gpr_axi_limit_lcdif1_thresh

Table continues on the next page...

Field	Function
15-0 GPR_AXI_LIMIT_LCDIF0_THRESH	gpr_axi_limit_lcdif0_thresh

13.2.3.1.44 AXI Limit Threshold Register 1 (AXI_LIMIT_THRESH1)

13.2.3.1.44.1 Offset

Register	Offset
AXI_LIMIT_THRESH1	134h

13.2.3.1.44.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR_AXI_LIMIT_DEWARP_THRESH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR_AXI_LIMIT_ISI_THRESH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.2.3.1.44.3 Fields

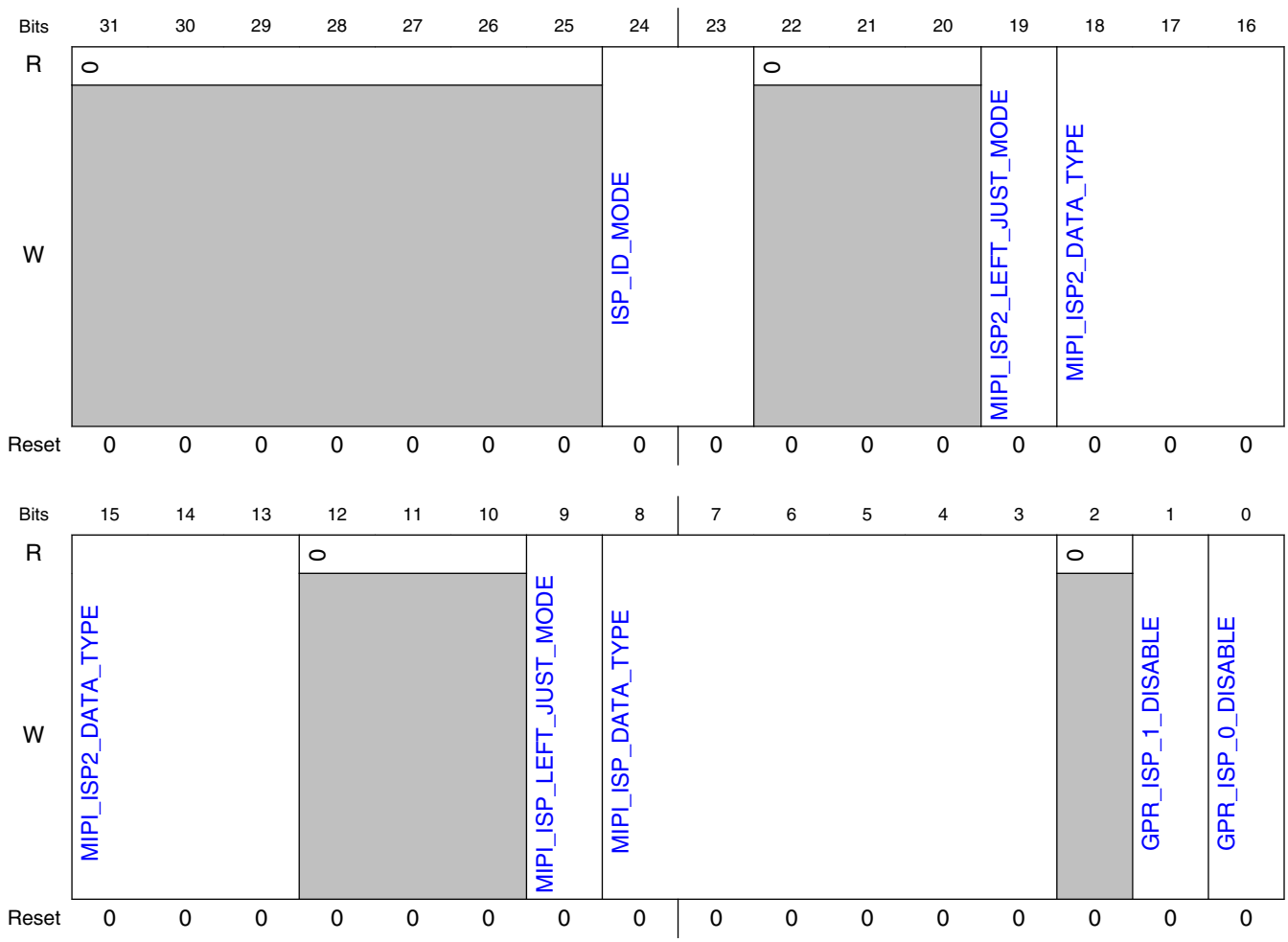
Field	Function
31-16 GPR_AXI_LIMIT_DEWARP_THRESH	gpr_axi_limit_dewarp_thresh
15-0 GPR_AXI_LIMIT_ISI_THRESH	gpr_axi_limit_isi_thresh

13.2.3.1.45 ISP Dewarp Control Register (ISP_DEWARP_CONTROL)

13.2.3.1.45.1 Offset

Register	Offset
ISP_DEWARP_CONTROL	138h

13.2.3.1.45.2 Diagram



13.2.3.1.45.3 Fields

Field	Function
31-25	Reserved

Table continues on the next page...

Field	Function
—	
24-23 ISP_ID_MODE	isp_id_mode 00b - vc_id_disable; ID will not toggle during no data transmit; 01b - vc_id_012 toggle 0,2 during no data transmit; 10b - vc_id_01 toggle 0,1 during no data transmit; 11b - vc_id_02 toggle 0,1,2 during no data transmit;
22-20 —	Reserved
19 MIPI_ISP2_LEFT_JUST_MODE	mipi_isp2_left_just_mode
18-13 MIPI_ISP2_DATA_TYPE	mipi_isp2_data_type 101000b - RAW6 101001b - RAW7 101010b - RAW8 101011b - RAW10 101100b - RAW12 101101b - RAW14
12-10 —	Reserved
9 MIPI_ISP_LEFT_JUST_MODE	mipi_isp_left_just_mode
8-3 MIPI_ISP_DATA_TYPE	mipi_isp_data_type 101000b - RAW6 101001b - RAW7 101010b - RAW8 101011b - RAW10 101100b - RAW12 101101b - RAW14
2 —	Reserved
1 GPR_ISP_1_DISABLE	gpr_isp_1_disable
0 GPR_ISP_0_DISABLE	gpr_isp_0_disable

13.3 LCD Interface (LCDIF)

13.3.1 Overview

The LCD Interface (LCDIF) is a system master that fetches graphics stored in memory and display them on a TFT LCD panel. A wide range of panel sizes is supported and the timing of the interface signals is highly configurable. Graphics are read directly from memory. Graphics may be encoded in a variety of formats to optimize memory usage.

13.3.1.1 Block diagram

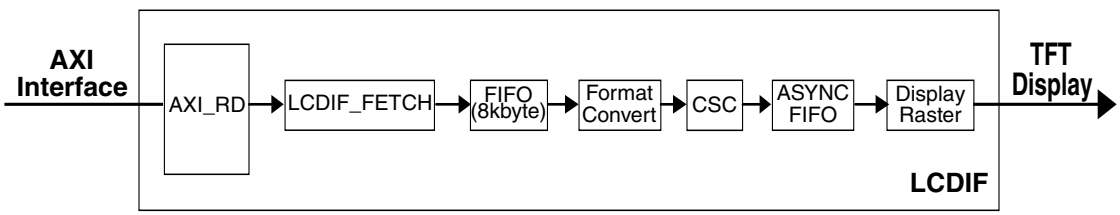


Figure 13-6. LCDIF block diagram

The above block diagram shows the LCDIF architecuture. The input source can be configured with:

- Horizontal and vertical size of graphic
- Position of graphic on the panel
- Address of graphic in memory
- Color formats or Color palettes

The values in the registers can be changed at any time, the panel content will be updated automatically in the panel refresh cycle when SHADOW_EN bit is set.

13.3.1.2 Features

The LCDIF includes the following features:

- The display mode supports DOTCLK mode only.
- One layer can support:
  - Programmable plane size (Width/Height/Pitch) and X/Y offset on the panel.
  - Index color support (1/2/4/8 bpp).

13.3.2 Functional description

The LCDIF is a master on the bus fabric; it fetches graphic source information directly from memory and sends data to a TFT LCD pannel.

As the LCDIF is a master on the NIC, it can access directly any memory or device connected to the NIC as a slave. Therefore, any compatiable graphic stored anywhere on-chip can be displayed on the connected TFT LCD panel with no further intervention from CPU.

13.3.2.1 Graphics and data format

The memory location of the graphic that is displayed on the layer is defined by register in the control descriptor for the layer . This 32-bit value can contain the address of any memory location in the memory map of the system.

The format of the data that describes the graphic is defined by the BPP bit field in the control descriptor for the layer . By choosing an appropriate format, it is possible to optimize the memory required by the graphics in use.

The CSC module receives pixels in any color space and can convert the pixels into any of RGB, YUV, or YCbCr color spaces.

All coefficients are programmable and in the two's compliment notation. The output pixels are passed onto the LUT and rotation engine for further processing.

The following equations indicate the CSC modules ALU architecture.

Selecting RGB output in CSC0_CTRL[CSC_MODE] configures the ALU in the following manner:

R = A1(Y-D1) + A2(U-D2) + A3(V-D3)

G = B1(Y-D1) + B2(U-D2) + B3(V-D3)

B = C1(Y-D1) + C2(U-D2) + C3(V-D3)

Selecting YUV output configures the ALU in the alternate manner:

Y = A1*R + A2*G + A3*B + D1

U = B1*R + B2*G + B3*B + D2

V = C1*R + C2*G + C3*B + D3

Saturation of each color channel is checked and corrected for excursions outside the nominal color space. Overflow for the three channels are saturated at 0x255 and underflow is saturated at 0x00.

13.3.2.2 Clocks

The table below shows the clocks described in this module:

Table 13-3. Clocks

Clocks	Description
pix_clk	Pixel Clock This clock should be configured according to TFT panel specification for supported display resolution
apb_clk	System Bus access clock
b_clk	LCDIF peripheral access bus clock connected to apb_clk

13.3.2.3 Reset

There is one asynchronous chip reset input to LCDIF, all the internal registers should be forced into their reset state when reset_n is active low.

There is one software reset bit (CTRL[SW_RESET]), which can reset all the internal registers except user configured registers.

13.3.2.4 Interrupts

There are two output interrupt signals (lcdif_d0_irq and lcdif_d1_irq) in this module. They have similar functions and can be used for different CPU domain in SoC.

The VSYNC interrupt flag indicates that the LCDIF has begun the next panel refresh period.

The UNDERRUN interrupt flag indicates the TXFIFO for display in underrun condition.

The VS_BLANK interrupt flag gives indication of the start of the vertical blanking period.

There are FIFO_EMPTY, DMA_DONE and DMA_ERR interrupt flags for each layer which indicate the status of pixel FIFO, frame buffer fetch and AXI interface respectively.

13.3.2.5 External Signals

There are no external signals pinned out for LCDIF.

13.3.2.6 Application information

The figure below shows the details of software flow for the LCDIF module.

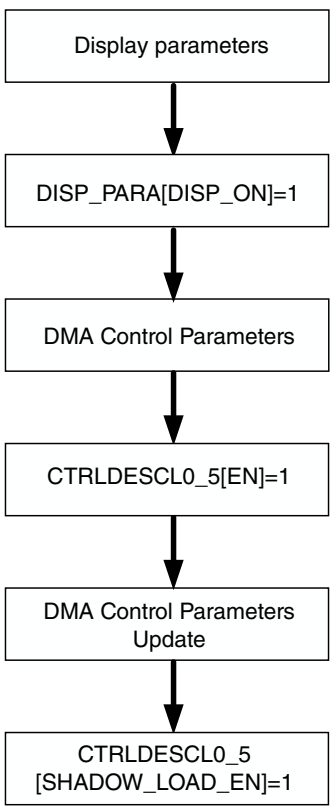


Figure 13-7. Software flow diagram

13.3.3 Memory Map and register definition

This section includes the LCDIF module memory map and detailed descriptions of all registers.

13.3.3.1 LCDIF register descriptions

13.3.3.1.1 LCDIF memory map

LCDIF1 base address: 32E8_0000h

LCDIF2 base address: 32E9_0000h

LCDIF3 base address: 32FC_6000h

Offset	Register	Width (In bits)	Access	Reset value
0h	LCDIF display control Register (CTRL)	32	RW	8000_0000h
4h	LCDIF display control Register (CTRL_SET)	32	RW	8000_0000h
8h	LCDIF display control Register (CTRL_CLR)	32	RW	8000_0000h
Ch	LCDIF display control Register (CTRL_TOG)	32	RW	8000_0000h
10h	Display Parameter Register (DISP_PARA)	32	RW	0000_0000h
14h	Display Size Register (DISP_SIZE)	32	RW	0000_0000h
18h	Horizontal Sync Parameter Register (HSYN_PARA)	32	RW	0003_0003h
1Ch	Vertical Sync Parameter Register (VSYN_PARA)	32	RW	0003_0003h
20h	Vertical and Horizontal Pulse Width Parameter Register (VSYN_HSYN_WIDTH)	32	RW	0003_0003h
24h	Interrupt Status Register for domain 0 (INT_STATUS_D0)	32	W1C	0000_0000h
28h	Interrupt Enable Register for domain 0 (INT_ENABLE_D0)	32	RW	0000_0000h
30h	Interrupt Status Register for domain 0 (INT_STATUS_D1)	32	W1C	0000_0000h
34h	Interrupt Enable Register for domain 0 (INT_ENABLE_D1)	32	RW	0000_0000h
200h	Control Descriptor Layer Register 1 (CTRLDESCL0_1)	32	RW	0000_0000h
208h	Control Descriptor Layer Register 3 (CTRLDESCL0_3)	32	RW	0000_0000h
20Ch	Control Descriptor Layer Register 4 (CTRLDESCL_LOW0_4)	32	RW	0000_0000h
210h	Control Descriptor Layer Register 4 (CTRLDESCL_HIGH0_4)	32	RW	0000_0000h
214h	Control Descriptor Layer Register 5 (CTRLDESCL0_5)	32	RW	0000_0000h
21Ch	Color Space Conversion Ctrl Register (CSC0_CTRL)	32	RW	0000_0001h
220h	Color Space Conversion Coefficient Register 0 (CSC0_COEF0)	32	RW	0000_0000h
224h	Color Space Conversion Coefficient Register 1 (CSC0_COEF1)	32	RW	0000_0000h
228h	Color Space Conversion Coefficient Register 2 (CSC0_COEF2)	32	RW	0000_0000h
22Ch	Color Space Conversion Coefficient Register 3 (CSC0_COEF3)	32	RW	0000_0000h
230h	Color Space Conversion Coefficient Register 4 (CSC0_COEF4)	32	RW	0000_0000h
234h	Color Space Conversion Coefficient Register 0 (CSC0_COEF5)	32	RW	0000_0000h
238h	Memory request priority threshold register (PANIC0_THRES)	32	RW	0100_0100h

13.3.3.1.2 LCDIF display control Register (CTRL)

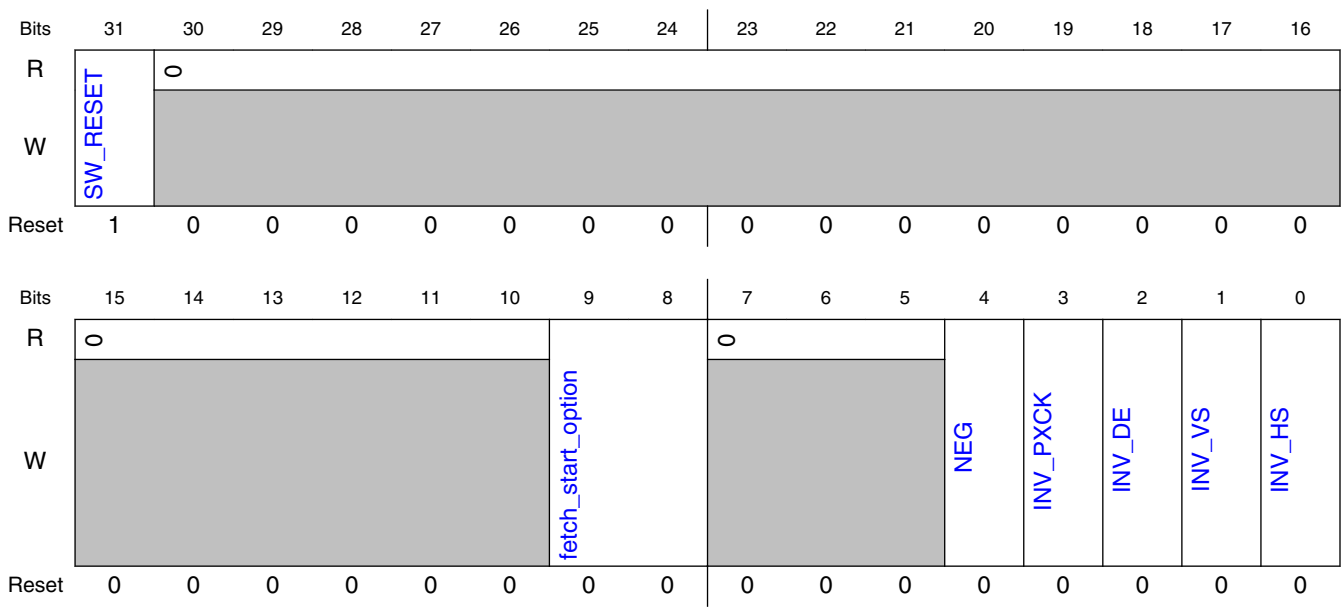
13.3.3.1.2.1 Offset

Register	Offset	Description
CTRL	0h	LCDIF display control Register
CTRL_SET	4h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL
CTRL_CLR	8h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL
CTRL_TOG	Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL

13.3.3.1.2.2 Function

This register control the polarity for HSYNC/VSYNC/DE of display.

13.3.3.1.2.3 Diagram



13.3.3.1.2.4 Fields

Field	Function
31 SW_RESET	SW_RESET Software reset, high active. When write 1 ,all logical will be reset. 0b - No action

Table continues on the next page...

Field	Function
	1b - All LCDIF internal registers are forced into their reset state. User registers are not affected.
30-10 —	Reserved
9-8 fetch_start_option	Indicates when to start fetching for new frame. This signals also decide the shadow load, fifo clear time 00b - fetch start as soon as FPV begins(as the end of the data_enable) 01b - fetch start as soon as PWV begins 10b - fetch start as soon as BPV begins 11b - fetch start as soon as RESV begins(still have hsync blanking for margin)
7-5 —	Reserved
4 NEG	Indicates if value at the output (pixel data output) needs to be negated. 0b - Output is to remain same 1b - Output to be negated
3 INV_PXCK	Polarity change of Pixel Clock. 0b - Display samples data on the falling edge 1b - Display samples data on the rising edge
2 INV_DE	Invert Data Enable polarity 0b - Data enable is active high 1b - Data enable is active low
1 INV_VS	Invert Vertical synchronization signal. 0b - VSYNC signal not inverted (active HIGH). 1b - Invert VSYNC signal (active LOW).
0 INV_HS	Invert Horizontal synchronization signal. 0b - HSYNC signal not inverted (active HIGH). 1b - Invert HSYNC signal (active LOW).

13.3.3.1.3 Display Parameter Register (DISP_PARA)

13.3.3.1.3.1 Offset

Register	Offset
DISP_PARA	10h

13.3.3.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DISP_ON	SWAP_EN	LINE_PATTERN				DISP_MODE		BGND_R							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BGND_G								BGND_B							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.3.3.1.3.3 Fields

Field	Function
31 DISP_ON	Display panel On/Off mode. 0b - Display Off. 1b - Display On.
30 SWAP_EN	output data swap enable. 0b - swap disable 1b - swap enbale, output data will swap the high 16bits with the low 16bits.
29-26 LINE_PATTERN	LCDIF line output order. 0000b - RGB/YUV. 0001b - RBG. 0010b - GBR. 0011b - GRB/UYV. 0100b - BRG. 0101b - BGR. 0110b - RGB555. 0111b - RGB565. 1000b - YUYV at [16:0]. 1001b - UYVY at [16:0]. 1010b - YVYU at [16:0]. 1011b - YUYV at [16:0]. 1100b - YUYV at [23:8]. 1101b - UYVY at [23:8]. 1110b - YVYU at [23:8]. 1111b - YUYV at [23:8].
25-24 DISP_MODE	LCDIF operating mode. 00b - Normal mode. Panel content controlled by layer configuration. 01b - Test Mode1.(BGND Color Display) 10b - Test Mode2.(Column Color Bar) 11b - Test Mode3.(Row Color Bar)
23-16 BGND_R	Red component of the default color displayed in the sectors where no layer is active.

Table continues on the next page...

Field	Function
15-8 BGND_G	Green component of the default color displayed in the sectors where no layer is active.
7-0 BGND_B	Blue component of the default color displayed in the sectors where no layer is active.

13.3.3.1.4 Display Size Register (DISP_SIZE)

13.3.3.1.4.1 Offset

Register	Offset
DISP_SIZE	14h

13.3.3.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DELTA_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DELTA_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.3.3.1.4.3 Fields

Field	Function
31-16 DELTA_Y	Sets the display size vertical resolution in pixels.
15-0 DELTA_X	Sets the display size horizontal resolution in pixels.

13.3.3.1.5 Horizontal Sync Parameter Register (HSYN_PARA)

13.3.3.1.5.1 Offset

Register	Offset
HSYN_PARA	18h

13.3.3.1.5.2 Function

HSYN_PARA register sets timing parameters related to the horizontal synchronization signal generation for the TFT. The fields FP_H, BP_H, and PW_H stand for HSYNC signal front-porch, back-porch, and active pulse width, respectively.

13.3.3.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BP_H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP_H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

13.3.3.1.5.4 Fields

Field	Function
31-16 BP_H	HSYNC back-porch pulse width (in pixel clock cycles). Pulse width has a minimum value of 1.
15-0 FP_H	HSYNC front-porch pulse width (in pixel clock cycles). Pulse width has a minimum value of 1.

13.3.3.1.6 Vertical Sync Parameter Register (VSYN_PARA)

13.3.3.1.6.1 Offset

Register	Offset
VSYN_PARA	1Ch

13.3.3.1.6.2 Function

VSYN_PARA register sets timing parameters of the TFT related to the vertical synchronization signal generation. The fields FP_V, BP_V, and PW_V stand for VSYNC signal front-porch, back-porch, and active pulse width, respectively.

13.3.3.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BP_V															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP_V															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

13.3.3.1.6.4 Fields

Field	Function
31-16 BP_V	VSYNC back-porch pulse width (in horizontal line cycles). Pulse width has a minimum value of 1.
15-0 FP_V	VSYNC front-porch pulse width (in horizontal line cycles). Pulse width has a minimum value of 1.

13.3.3.1.7 Vertical and Horizontal Pulse Width Parameter Register (VSYN_HSYN_WIDTH)

13.3.3.1.7.1 Offset

Register	Offset
VSYN_HSYN_WIDTH	20h

13.3.3.1.7.2 Function

VSYN_PARA register sets timing parameters of the TFT related to the vertical synchronization signal generation. The fields FP_V, BP_V, and PW_V stand for VSYNC signal front-porch, back-porch, and active pulse width, respectively.

13.3.3.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PW_V															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PW_H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

13.3.3.1.7.4 Fields

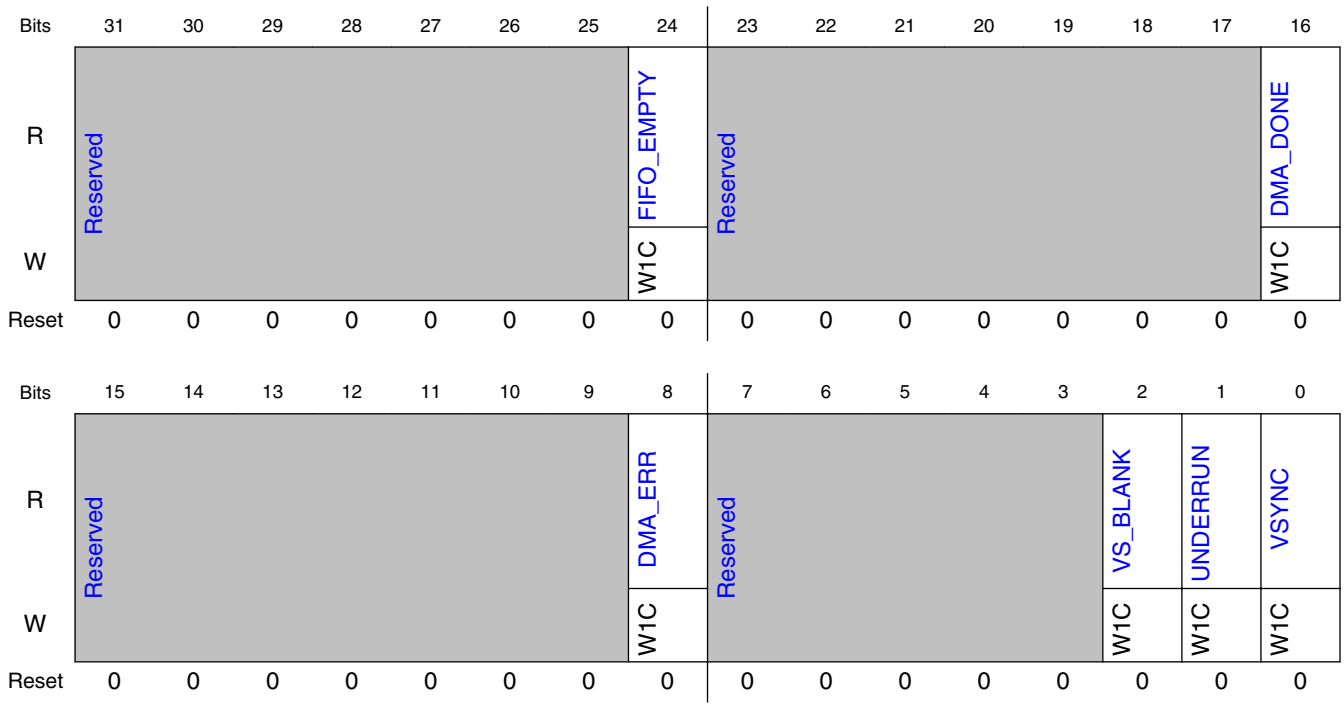
Field	Function
31-16 PW_V	VSYNC active pulse width (in horizontal line cycles). Pulse width has a minimum value of 1.
15-0 PW_H	HSYNC active pulse width (in pixel clock cycles). Pulse width has a minimum value of 1.

13.3.3.1.8 Interrupt Status Register for domain 0 (INT_STATUS_D0)

13.3.3.1.8.1 Offset

Register	Offset
INT_STATUS_D0	24h

13.3.3.1.8.2 Diagram



13.3.3.1.8.3 Fields

Field	Function
31-25 —	Reserved
24 FIFO_EMPTY	Interrupt flag to indicate that which FIFO in the pixel blending underflowed. • bit 0 -> FIFO has underflowed
23-17 —	Reserved
16 DMA_DONE	Interrupt flag to indicate that which PLANE has fetched the last pixel from memory. • bit 0 -> PLANE1 has fetched done
15-9 —	Reserved
8 DMA_ERR	Interrupt flag to indicate that which PLANE has Read Error on the AXI interface. • bit 0 -> PLANE1 has Read Error
7-3 —	Reserved
2 VS_BLANK	Interrupt flag to indicate vertical blanking period.
1	Interrupt flag to indicate the output buffer underrun condition.

Table continues on the next page...

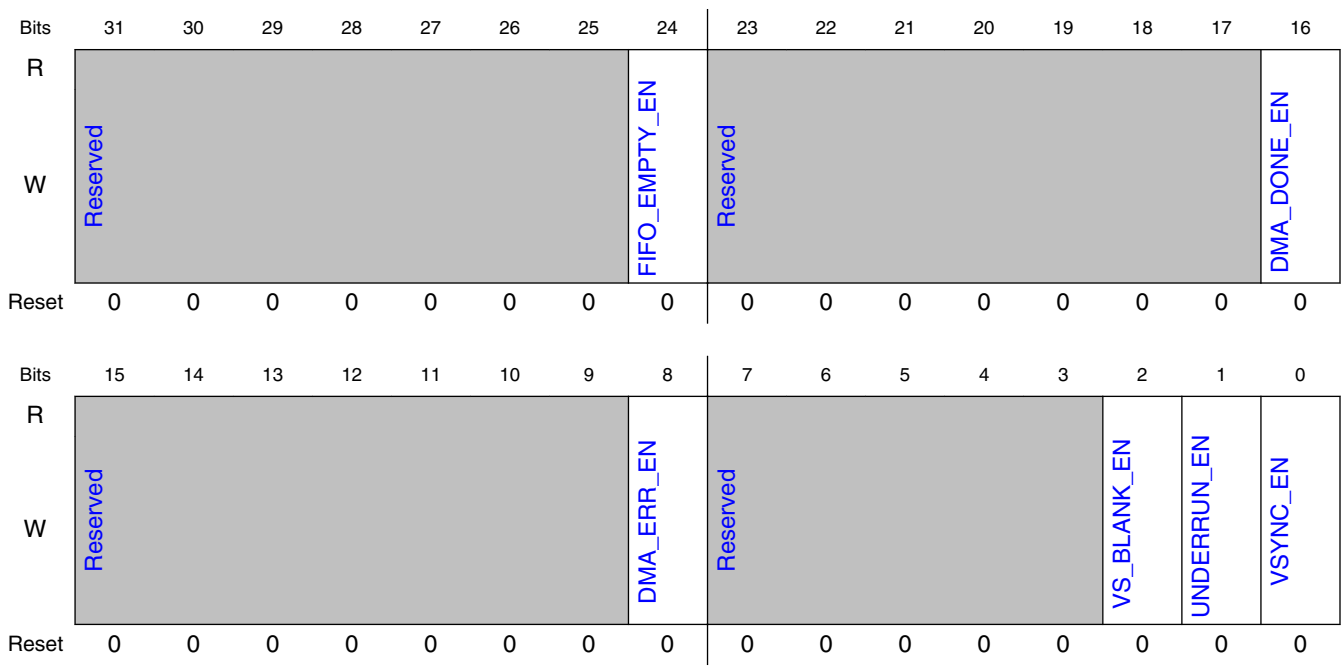
Field	Function
UNDERRUN	
0 VSYNC	Interrupt flag to indicate that the vertical synchronization phase(The beginning of a frame).

13.3.3.1.9 Interrupt Enable Register for domain 0 (INT_ENABLE_D0)

13.3.3.1.9.1 Offset

Register	Offset
INT_ENABLE_D0	28h

13.3.3.1.9.2 Diagram



13.3.3.1.9.3 Fields

Field	Function
31-25 —	Reserved
24	Enable Interrupt flag to indicate that which FIFO in the pixel blending underflowed.

Table continues on the next page...

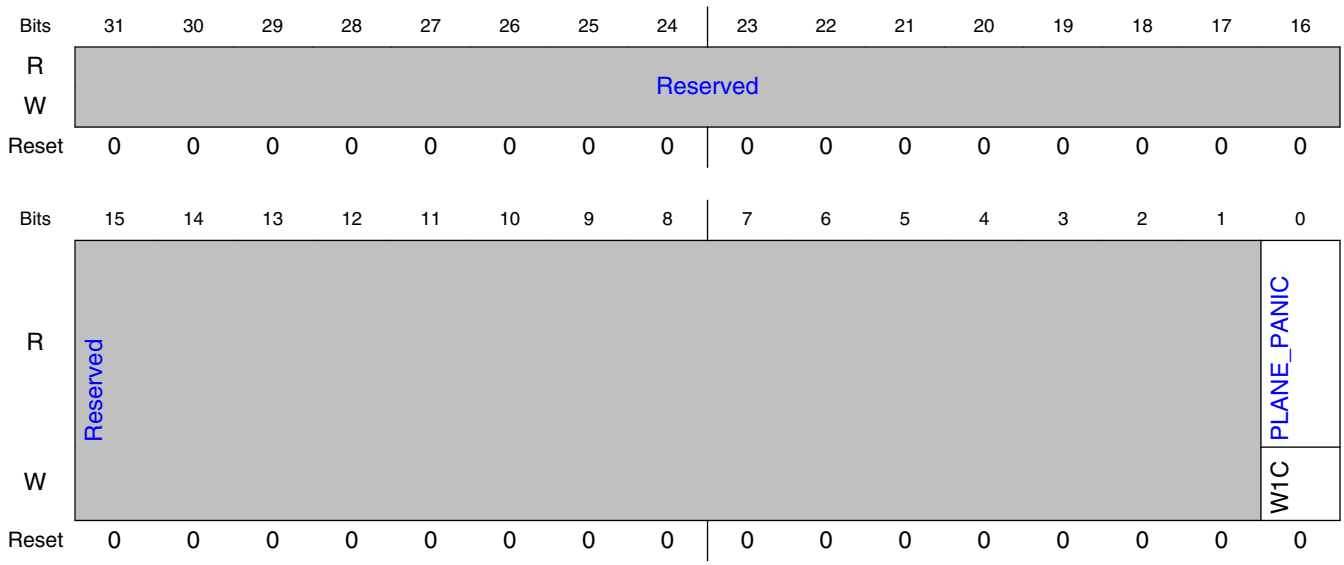
Field	Function
FIFO_EMPTY_EN	• bit 0 -> FIFO1 underflowed enable
23-17 —	Reserved
16 DMA_DONE_EN	Enable Interrupt flag to indicate that which PLANE has fetched the last pixel from memory. • bit 0 -> PLANE1 fetched done enable
15-9 —	Reserved
8 DMA_ERR_EN	Enable Interrupt flag to indicate that which PLANE has Read Error on the AXI interface. • bit 0 -> PLANE1 Read Error enable
7-3 —	Reserved
2 VS_BLANK_EN	Enable Interrupt flag to indicate vertical blanking period.
1 UNDERRUN_EN	Enable Interrupt flag to indicate the output buffer underrun condition.
0 VSYNC_EN	Enable Interrupt flag to indicate that the vertical synchronization phase(The beginning of a frame).

13.3.3.1.10 Interrupt Status Register for domain 0 (INT_STATUS_D1)

13.3.3.1.10.1 Offset

Register	Offset
INT_STATUS_D1	30h

13.3.3.1.10.2 Diagram



13.3.3.1.10.3 Fields

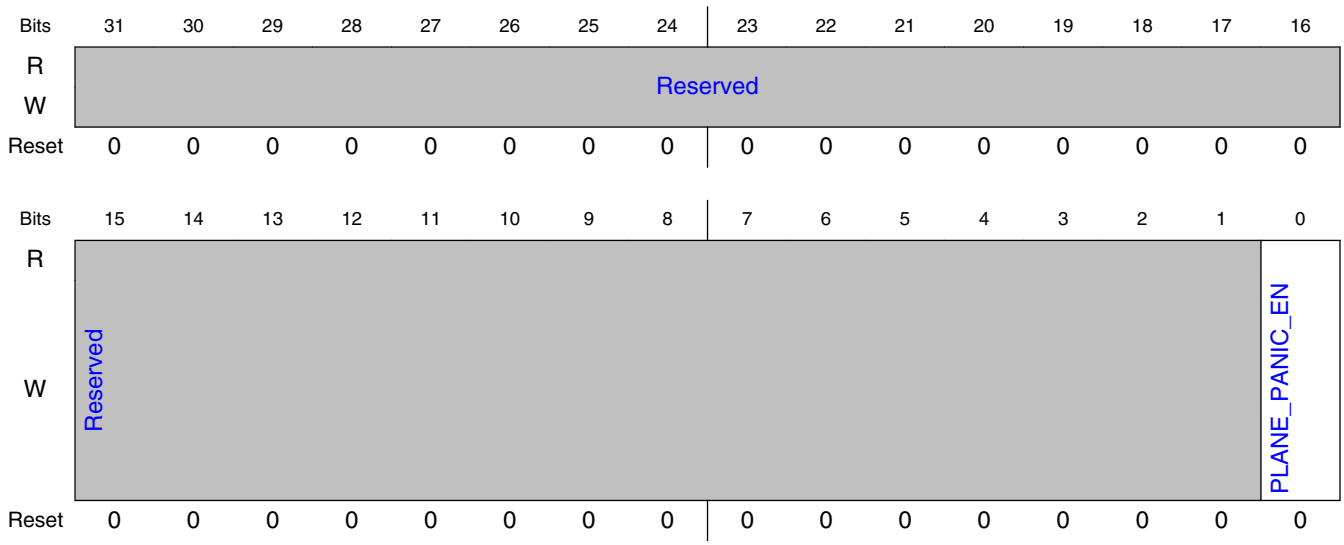
Field	Function
31-1 —	Reserved
0 PLANE_PANIC	plane panic to indicate that which FIFO reaches the panic threshold. • bit 0 -> FIFO1 panic

13.3.3.1.11 Interrupt Enable Register for domain 0 (INT_ENABLE_D1)

13.3.3.1.11.1 Offset

Register	Offset
INT_ENABLE_D1	34h

13.3.3.1.11.2 Diagram



13.3.3.1.11.3 Fields

Field	Function
31-1 —	Reserved
0 PLANE_PANIC_EN	Enable Interrupt flag to indicate that which FIFO in the pixel blending underflowed. • bit 0 -> FIFO1 panic enable

13.3.3.1.12 Control Descriptor Layer Register 1 (CTRLDESCL0_1)

13.3.3.1.12.1 Offset

Register	Offset
CTRLDESCL0_1	200h

13.3.3.1.12.2 Function

This register sets the height and width of the layer associated with the register.

13.3.3.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HEIGHT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WIDTH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.3.3.1.12.4 Fields

Field	Function
31-16 HEIGHT	Height of the layer in pixels.
15-0 WIDTH	Width of the layer in pixels. The layer width must be in multiples of the number of pixels that can be stored in 32 bits, and therefore differs depending on color encoding. For example, if 2 bits per pixel format is used, then the layer width must be configured in multiples of 16.

13.3.3.1.13 Control Descriptor Layer Register 3 (CTRLDESCL0_3)

13.3.3.1.13.1 Offset

Register	Offset
CTRLDESCL0_3	208h

13.3.3.1.13.2 Function

This register sets the buffer line pitch of layer data.

13.3.3.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PITCH															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.3.3.1.13.4 Fields

Field	Function
31-16 —	Reserved
15-0 PITCH	Number of bytes between 2 vertically adjacent pixels in system memory. Byte granularity is supported, but SW should align to 64B boundry.

13.3.3.1.14 Control Descriptor Layer Register 4 (CTRLDESCL_LOW0_4)

13.3.3.1.14.1 Offset

Register	Offset
CTRLDESCL_LOW0_4	20Ch

13.3.3.1.14.2 Function

This register sets the low 32bits of the beginning address of layer data.

13.3.3.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.3.3.1.14.4 Fields

Field	Function
31-0 ADDR_LOW	Address of layer data in the memory. The address programmed should be 64-bit aligned.

13.3.3.1.15 Control Descriptor Layer Register 4 (CTRLDESCL_HIGH0_4)

13.3.3.1.15.1 Offset

Register	Offset
CTRLDESCL_HIGH0_4	210h

13.3.3.1.15.2 Function

This register sets the high 4bits of the beginning address of layer data.

13.3.3.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												ADDR_HIGH			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.3.3.1.15.4 Fields

Field	Function
31-4 —	Reserved
3-0 ADDR_HIGH	Address of layer data in the memory. The address programmed should be 64-bit aligned.

13.3.3.1.16 Control Descriptor Layer Register 5 (CTRLDESCL0_5)

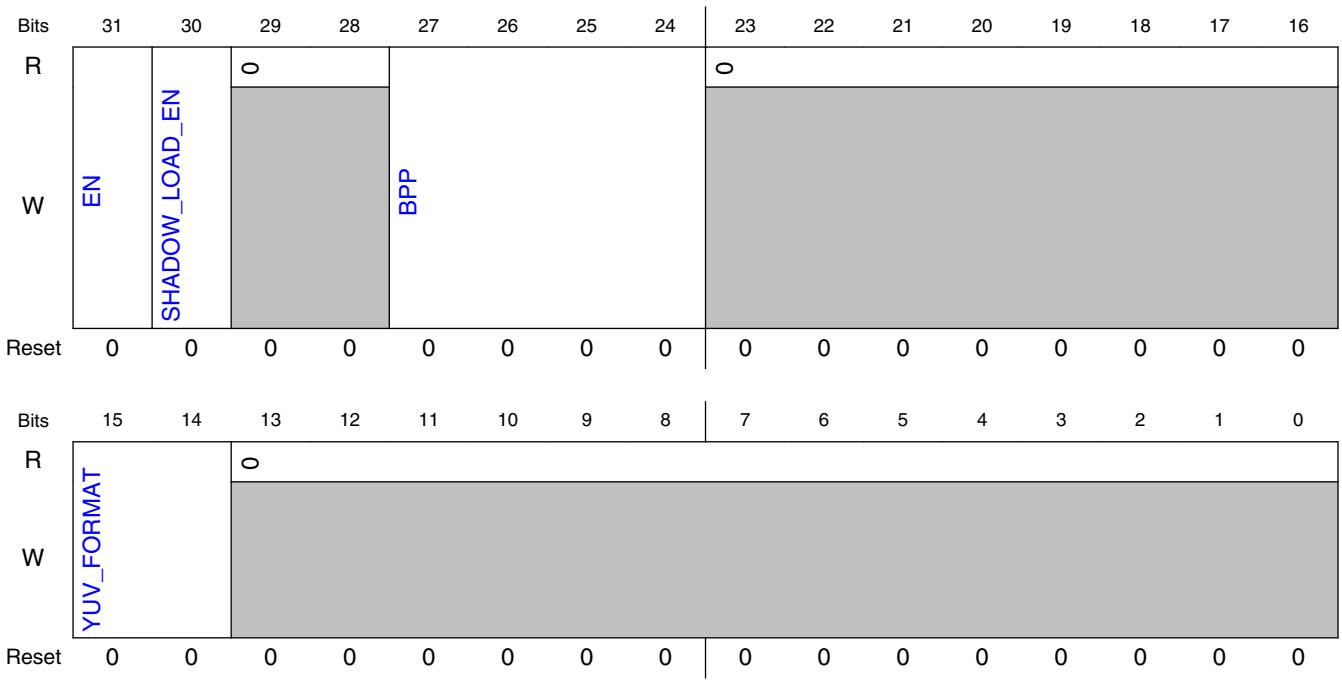
13.3.3.1.16.1 Offset

Register	Offset
CTRLDESCL0_5	214h

13.3.3.1.16.2 Function

This register controls various graphics options and whether the layer is enabled.

13.3.3.1.16.3 Diagram



13.3.3.1.16.4 Fields

Field	Function
31 EN	Enable the layer for DMA. 0b - OFF 1b - ON
30 SHADOW_LOAD_EN	Shadow Load Enable The SHADOW_LOAD_EN bit is written to 1 by software after all DMA control registers are written. If set to 1, shadowed control registers are updated to the active control registers on VSYNC of next frame. If set to 0, shadowed control registers are not loaded into the active control registers. The previous active control register settings will be used to process the next frame. Hardware will automatically clear this bit, when the shadow registers are loaded to the active control regsisters.
29-28 —	Reserved
27-24 BPP	Layer encoding format (bit per pixel) 0100b - 16 bpp (RGB565) 0101b - 16 bpp (ARGB1555) 0110b - 16 bpp (ARGB4444) 0111b - YCbCr422 1000b - 24 bpp (RGB888) 1001b - 32 bpp (ARGB8888) 1010b - 32 bpp (ABGR8888)
23-16 —	Reserved
15-14	The YUV422 input format selection.

Table continues on the next page...

Field	Function
YUV_FORMAT	00b - The YUV422 32bit memory is {Y2,V1,Y1,U1} 01b - The YUV422 32bit memory is {Y2,U1,Y1,V1} 10b - The YUV422 32bit memory is {V1,Y2,U1,Y1} 11b - The YUV422 32bit memory is {U1,Y2,V1,Y1}
13-0 —	Reserved

13.3.3.1.17 Color Space Conversion Ctrl Register (CSC0_CTRL)

13.3.3.1.17.1 Offset

Register	Offset
CSC0_CTRL	21Ch

13.3.3.1.17.2 Function

The CSC control register will configure the CSC module to perform color space converison between the RGB/YUV/YCbCr color spaces

// Converting from YUV/YCbCr color spaces to the RGB color space uses the

// R = A1(Y-D1) + A2(U-D2) + A3(V-D3)

// G = B1(Y-D1) + B2(U-D2) + B3(V-D3)

// B = C1(Y-D1) + C2(U-D2) + C3(V-D3)

//Converting from the RGB color space to YUV/YCbCr color spaces uses the

//following equation structure:

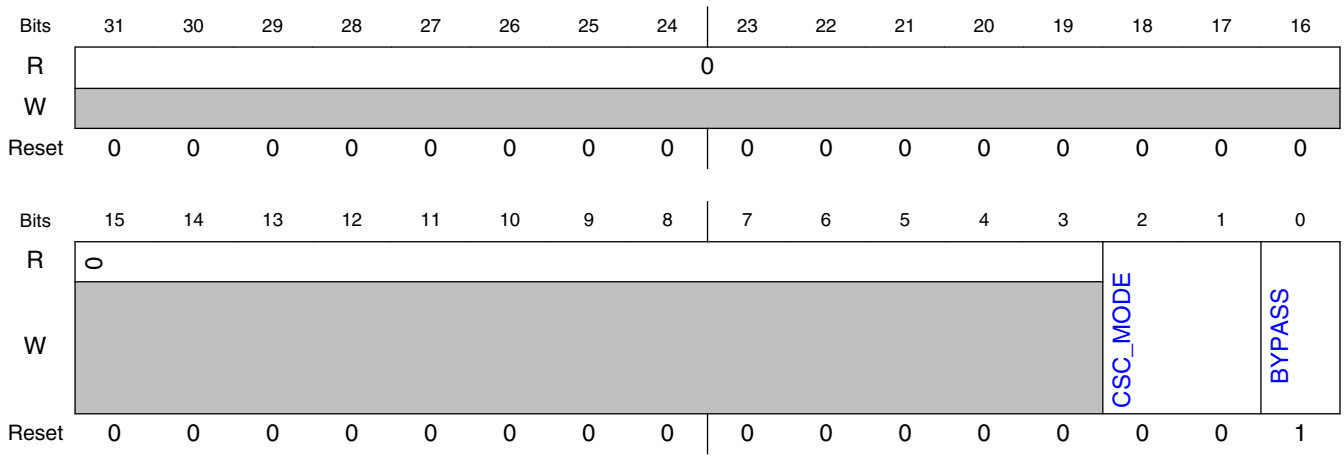
// Y = A1*R + A2*G + A3*B + D1

// U = B1*R + B2*G + B3*B + D2

// V = C1*R + C2*G + C3*B + D3

//All math is signed, so all coefficients come in as two's comp numbers

13.3.3.1.17.3 Diagram



13.3.3.1.17.4 Fields

Field	Function
31-3 —	Reserved
2-1 CSC_MODE	This field controls how the CSC unit operates on pixels when the CSC is not bypassed. 0x0 YUV2RGB-- Convert from YUV to RGB. 0x1 YCbCr2RGB-- Convert from YCbCr to RGB. 0x2 RGB2YUV -- Convert from RGB to YUV. 0x3 RGB2YCbCr -- Convert from RGB to YCbCr.
0 BYPASS	This bit controls whether the pixels entering the CSC2 unit get converted or not. When BYPASS is set, no operations occur on the pixels. When BYPASS is cleared, the selected CSC operation takes place.

13.3.3.1.18 Color Space Conversion Coefficient Register 0 (CSC0_COEF0)

13.3.3.1.18.1 Offset

Register	Offset
CSC0_COEF0	220h

13.3.3.1.18.2 Function

This register contains color space conversion coefficients in two's complement notation.

13.3.3.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					A2										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					A1										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.3.3.1.18.4 Fields

Field	Function
31-27 —	Reserved
26-16 A2	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 A1	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

13.3.3.1.19 Color Space Conversion Coefficient Register 1 (CSC0_COEF1)

13.3.3.1.19.1 Offset

Register	Offset
CSC0_COEF1	224h

13.3.3.1.19.2 Function

This register contains color space conversion coefficients in two's complement notation.

13.3.3.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					B1										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					A3										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.3.3.1.19.4 Fields

Field	Function
31-27 —	Reserved
26-16 B1	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 A3	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

13.3.3.1.20 Color Space Conversion Coefficient Register 2 (CSC0_COEF2)

13.3.3.1.20.1 Offset

Register	Offset
CSC0_COEF2	228h

13.3.3.1.20.2 Function

This register contains color space conversion coefficients in two's complement notation.

13.3.3.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					B3										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					B2										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.3.3.1.20.4 Fields

Field	Function
31-27 —	Reserved
26-16 B3	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 B2	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

13.3.3.1.21 Color Space Conversion Coefficient Register 3 (CSC0_COEF3)

13.3.3.1.21.1 Offset

Register	Offset
CSC0_COEF3	22Ch

13.3.3.1.21.2 Function

This register contains color space conversion coefficients in two's complement notation.

13.3.3.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					C2										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					C1										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.3.3.1.21.4 Fields

Field	Function
31-27 —	Reserved
26-16 C2	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 —	Reserved
10-0 C1	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

13.3.3.1.22 Color Space Conversion Coefficient Register 4 (CSC0_COEF4)

13.3.3.1.22.1 Offset

Register	Offset
CSC0_COEF4	230h

13.3.3.1.22.2 Function

This register contains color space conversion coefficients in two's complement notation.

13.3.3.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								D1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					C3										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.3.3.1.22.4 Fields

Field	Function
31-25 —	Reserved
24-16 D1	Two's complement D1 coefficient integer offset to be added.
15-11 —	Reserved
10-0 C3	Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

13.3.3.1.23 Color Space Conversion Coefficient Register 0 (CSC0_COEF5)

13.3.3.1.23.1 Offset

Register	Offset
CSC0_COEF5	234h

13.3.3.1.23.2 Function

This register contains color space conversion coefficients in two's complement notation.

13.3.3.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								D3							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								D2							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.3.3.1.23.4 Fields

Field	Function
31-25 —	Reserved
24-16 D3	Two's complement D3 coefficient integer offset to be added.
15-9 —	Reserved
8-0 D2	Two's complement D2 coefficient integer offset to be added.

13.3.3.1.24 Memory request priority threshold register (PANIC0_THRES)

13.3.3.1.24.1 Offset

Register	Offset
PANIC0_THRES	238h

13.3.3.1.24.2 Function

This register is used to activate panic signal when the number of pixels reaches the programmed threshold. It can be used to manipulate access priority to meet the required pixel throughput

13.3.3.1.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								PANIC_THRES_LOW							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PANIC_THRES_HIGH							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

13.3.3.1.24.4 Fields

Field	Function
31-25 —	Reserved
24-16 PANIC_THRES_LOW	This value should be set to a value of pixels, from (0+1)*128bits to (511+1)*128bits. When the number of pixels in the input pixel FIFO is less than this value, the panic control output will be raised.This signal can be used to raise the access LCDIF's access priority. • 0: threshold is 1*128bits • 1: threshold is 2*128bits • . • . • . • 511:threshold is 512*128bits
15-9 —	Reserved
8-0 PANIC_THRES_HIGH	This value should be set to a value of pixels, from (0+1)*128bits to (511+1)*128bits. When the number of pixels in the input pixel FIFO is more than this value, the panic control output will be removed.This signal can be used to raise the access LCDIF's access priority. • 0: threshold is 1*128bits • 1: threshold is 2*128bits • . • . • . • 511:threshold is 512*128bits

13.4 Image Sensing Interface (ISI)

13.4.1 Overview

Image Sensor Interface (ISI) module interfaces up to 2 pixel link sources to obtain the image data for processing in its pipeline channels. Each pipeline processes the image line from a configured source and performs one or more functions that are configured by software, such as down scaling, color space conversion, de-interlacing, alpha insertion, cropping and rotation (horizontal and vertical). The processed image is stored into programmable memory locations.

This module can process up to 2 image sources at the same time and can concatenate the internal line buffers such that higher resolution images can be processed at the expense of keeping some channels unused. This module implements limited flow control mechanism to control output from its internal buffer flushing or sourcing an image from memory. Depending on the format type, this module is capable of processing and storing one line of pixels from the incoming frame.

13.4.1.1 Block diagram

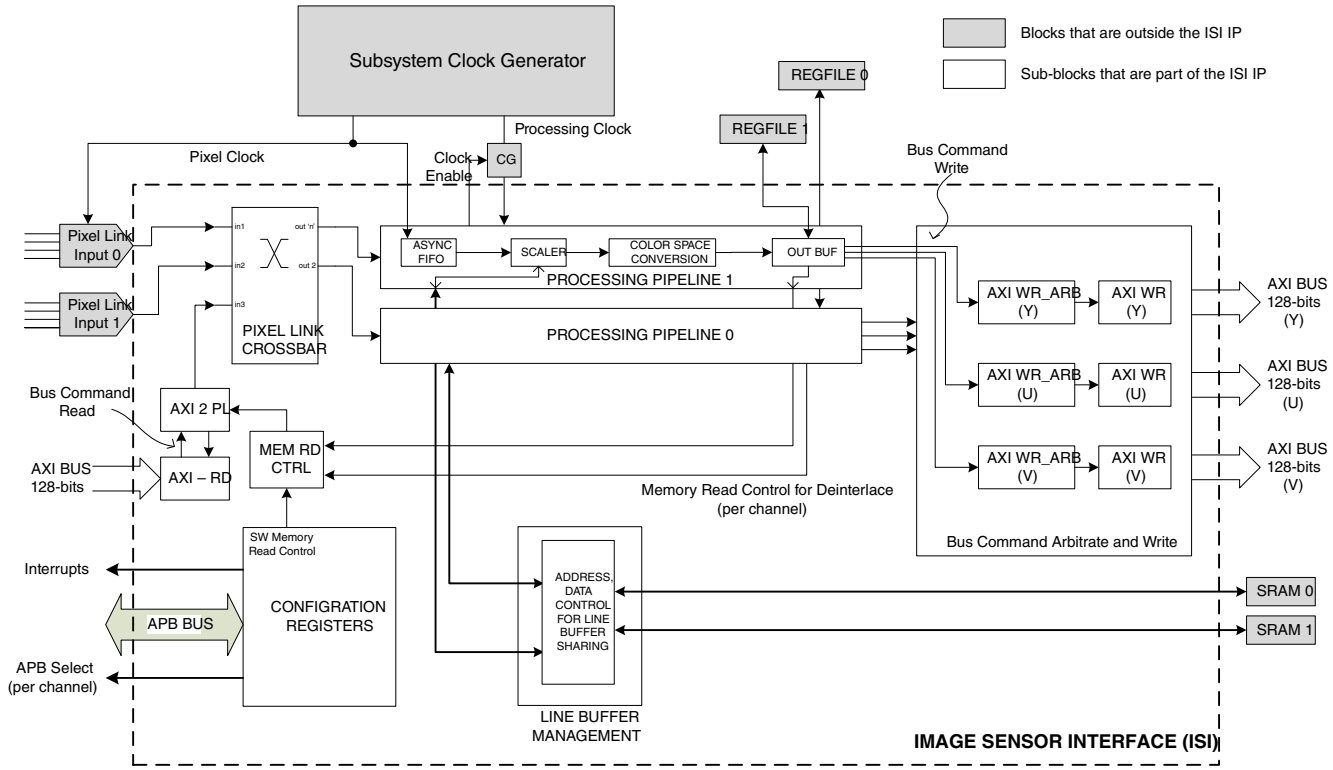


Figure 13-8. ISI block diagram

13.4.1.2 Features

The ISI includes the following features:

- Supports one source of 4K resolution at 30 fps (24bpp).
- Supports up to 2K resolution at 30 or 60 fps (24bpp) on each channel.
- Input sources supported:
  - 2 pixel link interfaces running in parallel that can interface to 2 camera sensors.
  - System Memory (AXI master; internally converted to Pixel Link Interface).
- Each processing pipeline or channel can be assigned to the same or different pixel input source.
- Stream Multiplexing
  - Duplicates input stream into multiple outputs.
  - Simple De-Interlacing methods supported for interlaced input sources:
    - Weaving
    - Line Doubling
    - Linear Interpolation or Averaging
  - One input source can be processed in more than one processing channels.
- Stream Manipulation Features
  - Supported Pixel Formats when storing image into memory
    - RAW8, RAW10, RAW12, RAW16

        - RGB888, BGR888, RGB565, RGB 10-bit, BGR 10-bit
        - YUV444, YUV422, YUV420 (8-bit, 10-bit, 12-bit) in planar or semi-planar formats
        - Plus more formats listed in the description of FORMAT field in the channel's IMG_CTRL register
    - Downscaling of input image via Decimation and Bilinear filtering
        - Decimation by 2, 4, or 8 supported
        - Bilinear filter further downscales by 1.0 to 2.0 (fractional downscaling)
    - Color Space Conversion (CSC)
        - RGB, YUV, YCbCr
        - User defined color space matrix based conversion
    - Alpha channel insertion for RGB formats
        - Global alpha value.
        - Separate rectangular region of interest (ROI) alpha value. ROI alpha values have higher priority than global alpha value.
            - Up to 4 ROI non-overlapping rectangles supported
    - Mirroring (Image Flip): Horizontal and Vertical flips supported.
    - Frame Awareness, Frame Skipping
        - Clean frame start, shutdown based on HSYNC and VSYNC.
        - Buffer overrun protection.
        - Buffer under run deterministic behavior.
- Stream Output Options
    - Output to Memory
        - Input source will be converted to and processed by the processing pipeline as YUV444 or RGB.
        - Processed images will be output from pipeline and stored into memory location specified by software.
        - Full line storage available at processing channel output before outputting data to AXI. This storage can automatically split or combine depending on the output format being used.
        - Dual buffered addresses used in ping pong fashion with active buffer status indication.
        - Line and Frame stored interrupt status to software to track progress of frame.
- Flow Control
    - Panic indication to software and device to increase priority of its write transactions to avoid potential overflow in output buffers. Software can configure thresholds for panic indication.

- When pixels are sourced from memory or input line buffers are flushed, software has the option to program the rate at which pixels will be sent out. By default, one pixel per clock is sent out.
- Back pressure mechanism to stall the channel pipeline during line buffer flushing and sourcing image from memory, when AXI bus is unable accept data and the output buffers are low on storage.

13.4.2 Design Overview

The design architecture and data flow can be understood by going one block at a time.

Pixel Link Crossbar: The ISI module comprises of 2 parallel processing pipelines which can be assigned to any of the input image source via the Pixel Link Crossbar. The crossbar is a 3 input, 2 output multiplexer where each output port can be configured to connect to any of the 3 inputs. The 3 inputs are numbered from 0 to 2; of which input 3 is always a memory location. One input can be assigned to multiple outputs on the crossbar. There is no latency in the crossbar. The image input to the module is through the Pixel Line Slave port which provides the clock, data & control buses and HSYNC and VSYNC signals. The crossbar connects these slave port signals to any of the 2 processing channels which pull the pixel data out from the asynchronous FIFO in the Pixel Line Slave port. The only exception is when the image is read from memory. In this case, the ISI module will convert the lines read from memory into Pixel Link Slave interface signals so that the signals seen at the crossbar and processing channels are same as other sources.

Processing Pipeline: Each processing channel comprises of a Down Scaling Engine, Color Space Conversion (CSC) block and Output Buffer Control block. The scaler performs decimation and bilinear filtering of the input image. The output buffer control perform various functions like alpha insertion, cropping, horizontal or vertical flipping and generating AXI Write transactions for storing the processed image into the memory. It supports storing the image as 1/2/3 planar at different memory locations. Each processing channel has one line buffer associated with it. Each line buffer can store up to 2048 pixels with 8-bits per color component. For processing higher resolution images (like 4K), line buffer from other channels can be combined to provide more storage. However, in this case, the channels from which line buffers are borrowed will remain disabled and cannot function. There is a restriction on software that buffers from adjacent channels can only be concatenated. This means channel 1 line buffer can be connected to channel 2 line buffer and no other channel. Additionally, when performing scaling on YUV420 input data, an additional line buffer is needed which can be configured independently of line buffer concatenation. Software must ensure correct channel's line buffers are used.

Line Buffer Management: Processing channels are capable of handling up to 4K image resolution. However, the line buffer storage in one channel is enough for 2K image resolution. In order to be able to process and store a 4K image, line buffer from adjacent channel needs to be chained with the current channel's line buffer. The very last channel's line buffer cannot be concatenated with other channels, hence 4K resolution cannot be supported on that channel. The line buffer management block handles the muxing and concatenation of line buffers across channels. Software will configure chaining of line buffers and how many buffers to use in parallel. Thus, for example, if channel 2 is processing a 4K image then the address bus of this channel will be able to access line buffers from channels 2 and 3. Channel 1 will be able to address line buffer 1 only since it can process a lower resolution image only. When line buffers get chained, the output buffers or ping pong buffers also get chained automatically.

NOTE

For Line Buffer Management, each channel has a corresponding clock which is used. The CHNLn_CTRL[CLK_EN] bit must also be set for each channel in order to enable chain buffering.

This sub-block also handles sharing of line buffer as second line buffer for a primary channel that is performing YUV420 scaling, which requires two line buffers. The software configures the channel number whose line buffer acts as the second line buffer for the primary channel performing the scaling. This block routes the read/write data between the primary or second line buffers as addressed by the primary channel. Note that output buffers are not shared between primary and secondary channels and hence the secondary channel can be used in bypass mode to process images as required by software or can remain disabled.

Output Buffer Control and AXI Output: The output buffer control block of each processing channel will perform image cropping, flipping (horizontal and vertical) and alpha value insertion, in addition to collecting the processed image pixels in a ping pong buffer and send out bursts of 128 bytes or 256 bytes (configurable) of data over the AXI interface to memory. This block also performs the function of de-interlacing the frames while storing them into memory. The final processed image pixels can be formatted to be stored in any of the multiple image formats mentioned in the description of the FORMAT field of the IMG_CTRL register. The output buffer control also manages the concatenation of ping pong buffers per component based on output image format. The combined storage of the 3 ping pong buffers is up to a full line of the image frame depending on the output image format. Flow control and panic mechanisms are implemented to ensure the data is sent out at a rate acceptable to the system and output buffers do not overlow.

AXI Memory Read: The ISI module is capable of reading an image from memory in different formats as mentioned in the MEM_RD_CTRL register. The module reads the image via AXI read and internally drives the image through a pixel to the processing channel. This operation is also flow controlled and software can configure the rate at which pixels are processed in the channel and sent out. By default, one pixel per clock is processed when the read FIFO is not empty.

Configuration Registers: Each channel has the same set of configuration registers which are aligned to a 4 KB boundary and will be programmable via APB interface. The module does not check for correctness of values being programmed. It is up to the software to ensure this.

13.4.3 Functional description

The following sections describe functional details of the ISI module.

13.4.3.1 Initialization and Channel Configuration

This section describes the suggested steps to configure the individual channels in ISI and handle interrupts for proper and efficient image processing. Every channel that needs to be configured must follow these suggested initialization steps for correct operation of the module.

First the general channel configuration is done as follows:

- Set the CNHL_CTRL[CLK_EN] bit to 1 to enable the clock to the channel. This is required to access other registers in the channel and also for image processing by channel.
- If the channel is supposed to be bypassed i.e. No Scaling and No Color Conversion is needed; set the CHNL_CTRL[CHNL_BYPASS] bit to 1, else set to 0. This will cause the image coming into the channel to be directly stored into memory without any kind of processing. However, image flipping (horizontal and/or vertical), image cropping, alpha insertion (RGB format only) and format conversion (YUV444 to YUV422 or YUV420) can still be done while bypass is enabled.
- Select the source of input image. The input can be any of the 2 Pixel Links (0 to 1). An additional pixel link, (Pixel Link 2), can be selected as a source. Pixel Link 2 will always read an image from a programmed location in memory. Configure the source in CHNL_CTRL[SRC] field.

NOTE

- Please refer to SOC Configuration for actual source types connected to Pixel Links 0 to 1.

- Select the type of source for the channel between non-memory Pixel Links (image sourced from camera sensor, SRC_TYPE = 0) and Memory Pixel Link (image sourced from memory, SRC_TYPE = 1). A memory Pixel Link refers to a Pixel Link 2 which sources the image from a programmed memory location only. This is useful for de-interlacing and processing image directly from memory. For channels that do not support sourcing image from memory, the source type is always a Non-memory Pixel Link. Configure the source type for channel in CHNL_CTRL[SRC_TYPE] field.
- Configure the image width and height in channel's CHNL_IMG_CFG register. For example, a 1920x1080 image will have CHNL_IMG_CFG[WIDTH] = 0x780 and CHNL_IMG_CFG[HEIGHT] = 0x438.
- Configure the memory address where the processed or bypassed data from the channel has to be stored. The image can be stored as 1-planar (all color components stored together), 2-planar (Y component stored in one memory address and UV stored into a different memory address), and 3-planar (each component stored in different memory address). For 1-planar addressing, configure the 32-bit address in CHNL_OUT_BUF1_ADDR_Y and CHNL_OUT_BUF2_ADDR_Y registers. For 2-planar and 3-planar addressing, the CHNL_OUT_BUF1_ADDR_U and CHNL_OUT_BUF2_ADDR_U are additionally used. For 3-planar addressing, the CHNL_OUT_BUF1_ADDR_V and CHNL_OUT_BUF2_ADDR_V are additionally used. Refer to section on Output Buffer Addressing for more details on how the two buffer addresses are used.

NOTE

RGB and RAW images are always stored as 1-planar while other YUV formats can be stored as 1/2/3-planar.

- Configure the output buffer line pitch in the CHNL_OUT_BUF_PITCH[LINE_PITCH] register. The line pitch is the number of bytes between the start location of two lines. It is mainly used to have line addresses start at an aligned boundary.

NOTE

- The line pitch is same for all image components irrespective of they being stored in 1-planar, 2-planar, or 3-planar. In 2 and 3-planar output, the U and V component will have the same line pitch as the Y component.
- Ensure that line pitch is programmed to be greater than number of bytes in one line.

- Toggle the address load bits i.e. CHNL_OUT_BUF_CTRL[LOAD_BUF1_ADDR] and CHNL_OUT_BUF_CTRL[LOAD_BUF2_ADDR] to allow new configured addresses to be picked and used in the channel, when it is enabled later.

Follow the steps detailed in appropriate sections based on which functionality is to be used for the application.

- Refer to the Scaling Operation section for details about scaler and configuring the scaler for use.
- Refer to the Color Space Conversion (CSC) section for details about CSC and configuring the block for use.
- Refer to the Image De-interlacing section for details about different de-interlacing methods and configuration for enabling each de-interlacing method for the channels.
- Refer to the High-Res Image Handling section for enabling controls to allow higher than 2K image resolution to be handled by the channel.
- Refer to the Sourcing Image from Memory section for details on how to configure channel 0 to read and process an image that is sourced from memory.
- Refer to the Output Buffer Management section for details on how to configure controls for image cropping, flipping, alpha insertion, handle buffer overflows and request higher AXI channel priority for emptying the buffers.
- Refer to the Flow Control section for details on how to configure the rate of pixel outout from this module on to the AXI bus.

Once the above steps are completed, the channel can now be enabled and monitored using the following steps:

- Enable the necessary interrupt by writing '1' to appropriate interrupt enable bits in the CHNL_IER register.
- Enable the channel by writing '1' to the CHNL_CTRL[CHNL_EN] bit. This is enable to channel to start accepting images from the selected source.

- Monitor the status of channel through the CHNL_STS register. Each bit provides a status or error indication for the software to take appropriate action. Interrupt status bits can be cleared by writing 1 to respective bits.
- When required, the complete channel can be reset to its default reset state by writing '1' to the CHNL_CTRL[SW_RST] bit. Software must keep this bit '1' for at least 5 to 6 NOP cycles to allow reset to propagate and then set this bit to '0' to complete the reset cycle. After the software reset, the channel will return to the same default state as it would have been after a hardware reset. No configuration register is reset during the software reset. Software may additionally clear the CHNL_CTRL[CHNL_EN] bit while software reset is asserted to control the restarting of channel after software reset completes.

When configuring multiple channels, the above suggested steps must be carried out for all channels.

13.4.3.2 Scaling Operation

The scaling engine works in two parts. It performs pre-decimation of the image followed by bilinear filtering to achieve the desired down scaling factor.

The decimation filter provide a maximum down scaling factor of 8, and the subsequent bilinear filter provides a maximum down scaling factor of 2. Combined, the maximum scaling factor can be up to 16. The decimation and bilinear scaling engines are independently programmable. There is also an initial offset that is programmable to allow more source data to be considered in the bilinear scaling engine. The scaled output image is presented to the CSC module as YUV444 or RGB888 pixels with a single byte for each color channel.

13.4.3.2.1 Configuring the scaler

Channel initialization is a prerequisite to configuring the scaler. Steps to configure scaler are as follows:

- Configure the scale offset values for X and Y dimensions in CHNL_SCALE_OFSET register. This is the initial offset from (0,0) coordinate, where the first output scaled pixel will be computed.
- Configure the scale factor values for X and Y dimensions in CHNL_SCALE_FACTOR register. This is step factor through which the image will be sampled for down scaling.
- Configure the decimation factor values for X and Y dimensions in CHNL_IMG_CTRL[DEC_X] and CHNL_IMG_CTRL[DEC_Y] fields. The final scaling factor that will be used by scaler is: = (Decimation Factor) x (Scale Factor)

- Set the CHNL_IMG_CTRL[YCBCR_MODE] bit to 1 if the incoming image or the converted image is in $YC_b C_r$ format.
- Set the CHNL_SCL_IMG_CFG register to the dimension of image after scaling. This register must be configured whenever scaler is enabled, even with 1:1 scaling.
- **Special Consideration for YUV420 image input:** When a channel has to handle YUV420 image type at the scaler input, it must follow these additional steps:
  - If scaler is bypassed, no additional action is needed. The scaler will convert the input YUV420 image to YUV444 for processing in the channel.
  - If scaler is enabled, an additional line buffer is needed by the channel to perform scaling correctly. Set the CHNL_CTRL[SEC_LB_SRC] to the channel number whose line buffer is to be used. The channel whose line buffer is being shared cannot be used and will remain disabled.
  - For higher than 2K resolution, it is important to set the CHNL_CTRL[CHAIN_BUF] in both channels i.e. channel performing the scaling and the channel whose line buffer is being shared.

NOTE

For both types of YUV420 data format with scaler enabled, it must be ensured that the minimum VSYNC width should be 2 lines. Additionally for YUV420 Type 2 format, the minimum HYSNC front porch width should be 1 clock. For all other cases, the minimum VSYNC width should be 1 line and minimum HSYNC porch (front and back) width should be 0 clock.

- Return to initialization steps to complete the initialization of other blocks and enabling the channel.

Example: If the image needs to be down scaled by a factor of 4.5 in the X and Y dimension, then the software can program the scaler as:

- CHNL_IMG_CTRL[DEC_X] and CHNL_IMG_CTRL[DEC_Y] = 10 or decimate by 4
- CHNL_SCALE_FACTOR[X_SCALE] and CHNL_SCALE_FACTOR[Y_SCALE] = 0x2200 indicating a scaling factor of 1.125
- This will give the overall down scaling by 4 x 1.125 = 4.5

13.4.3.3 Color Space Conversion (CSC)

The Color Space Conversion or CSC is a full duplex color space converter that can convert into either RGB or YUV (or YC_bC_r) color spaces depending on the desired output pixel format. All coefficients are programmed as two's compliment numbers. The

CSC unit can be individually bypassed on a channel if it is not desired in the application. The output of the CSC is sent to the output buffer control block for further processing and storing into memory.

The following equations indicate the CSC module's ALU architecture.

(a) RGB to YUV (or YC_bC_r) conversion

- Y = (A1 x R) + (A2 x G) + (A3 x B) + D1
- U = (B1 x R) + (B2 x G) + (B3 x B) + D2
- V = (C1 x R) + (C2 x G) + (C3 x B) + D3

(b) YUV (or YC_bC_r) to RGB conversion

- R = (A1 x (Y - D1)) + (A2 x (U - D2)) + (A3 x (V - D3))
- G = (B1 x (Y - D1)) + (B2 x (U - D2)) + (B3 x (V - D3))
- B = (C1 x (Y - D1)) + (C2 x (U - D2)) + (C3 x (V - D3))

where A1 to A3, B1 to B3, C1 to C3 and D1 to D3 are the programmable coefficients and YUV and RGB refer to the pixel values.

Saturation of each color channel is checked and corrected for excursions outside the nominal color space. Overflow for the three channels are saturated at 0x255 and underflow is saturated at 0x00.

13.4.3.3.1 Configuring the CSC

Channel initialization is a prerequisite to configuring the CSC. Steps to configure CSC are as follows:

- If CSC is not required to be run and the channel is not bypassed, then CSC block can be bypassed by setting the CHNL_IMG_CTRL[CSC_BYP].
- If CSC is required then configure the coefficients in the CHNL_CSC_COEFF0 to CHNL_CSC_COEFF5 registers.
- Set the CHNL_IMG_CTRL[YCBCR_MODE] bit to 1 if the incoming image or the image after color space conversion is in $YC_b C_r$ mode.
- Set the CHNL_IMG_CTRL[CSC_MODE] field to configure the required conversion method.
- Return to initialization steps to complete the initialization of other blocks and enabling the channel.

13.4.3.4 Image De-interlacing

ISI supports image de-interlacing for images that are interlaced across frames. One of the following de-interlace methods can be used to form the de-interlaced image.

- ***Weaving -*** Weaving is done by combining consecutive fields together to form the new frame. Weaving will result in artifacts known as "combing," when the pixels in one frame do not line up with the pixels in the other, forming a jagged edge.
  - To perform weaving in an ISI channel, one field's lines are stored in memory with double line pitch; thereby leaving one line space in memory for the second field's lines to be stored. The two interlaced frames get woven while being stored to memory. Software has the choice to have odd frame on top or even frame on top.

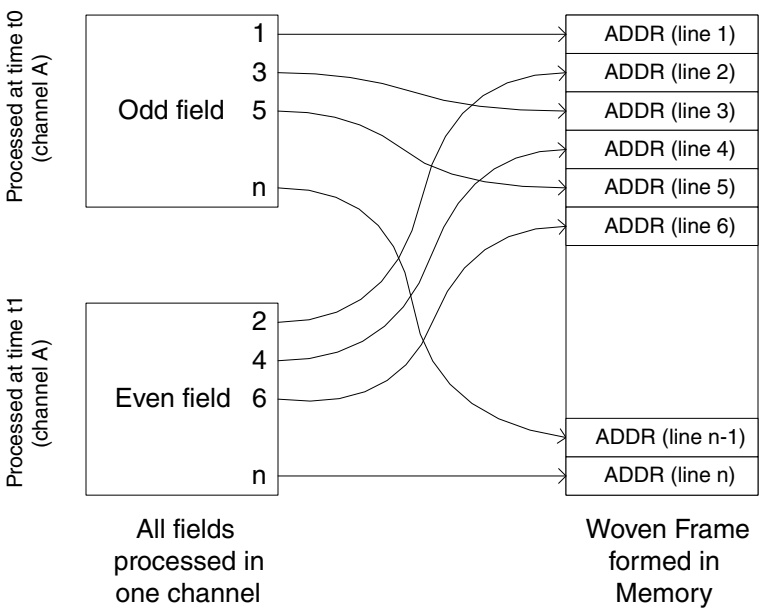


Figure 13-9. Weave De-interlacing

- ***Line Doubling -***Line doubling takes the lines of each interlaced field (consisting of only even or odd lines) and doubles them, filling the entire frame. This results in the video having a frame rate identical to the field rate, but each frame having half the vertical resolution, or resolution equal to that of each field that the frame was made from. Line doubling prevents combing artifacts but causes a noticeable reduction in picture quality since each frame displayed is doubled and really only at the original half field resolution.
  - To perform line doubling in an ISI channel, two channels are enabled and configured the same way except for the output memory address. One channel will store the image at location ADDR, while the other channel will store the

image at location ADDR + Line Pitch. Both channels will process the same image lines and store them at a line pitch distance in memory thereby causing the line to be doubled.

- Any two channels can be used to perform line doubling.

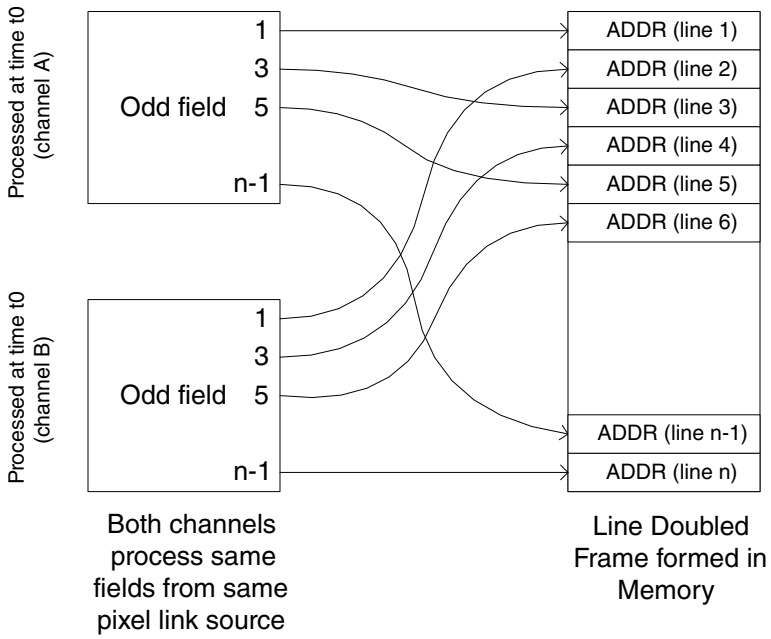


Figure 13-10. Line Doubling De-interlace

- ***Blending (linear interpolation)* -** is done by averaging consecutive fields to be displayed as one frame. Combing is avoided because the images are on top of each other. This instead leaves an artifact known as ghosting.
  - To perform blending in an ISI channel, two channels are required because blending is carried out in two steps:
    1. Perform weaving of the two fields and store them to an intermediate (or temporary) location.
    2. Read back from temporary location using another channel, perform averaging using the scaler and store the final frame into the desired memory location.

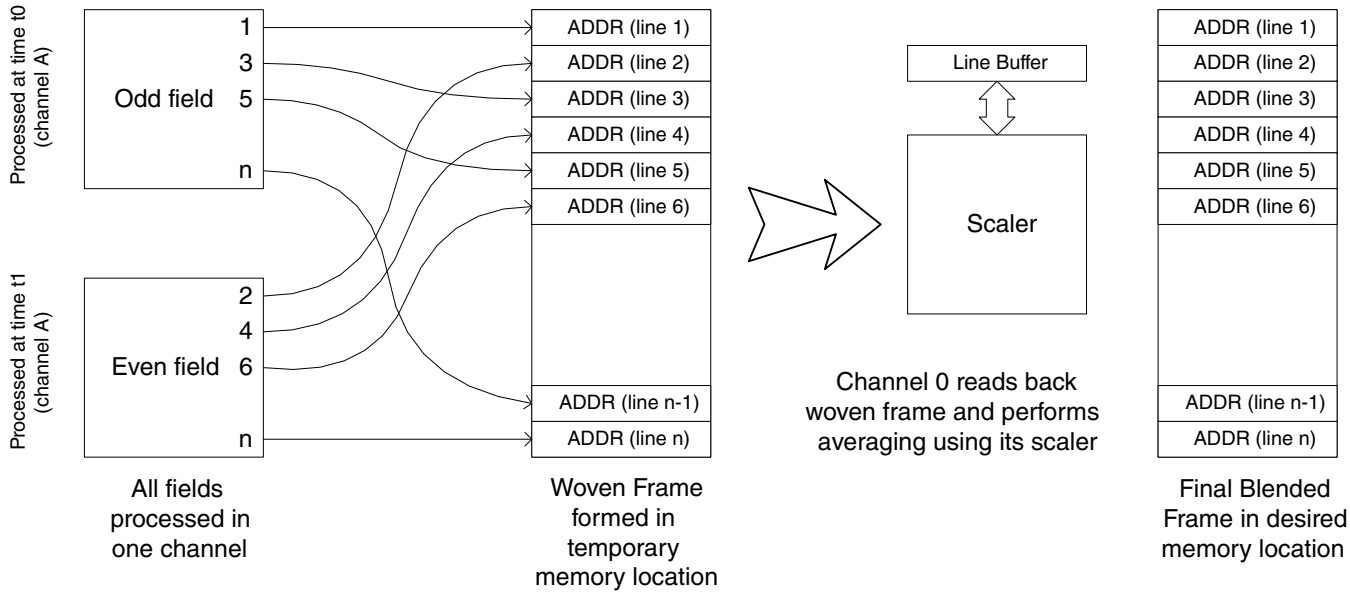


Figure 13-11. Blend De-interlace

13.4.3.4.1 Configuring de-interlacing

Channel initialization is a prerequisite to configuring the de-interlace block. Steps to configure the de-interlace block are as follows:

- ***Weaving -*** Configure the weaving de-interlace method by writing "010" or "011" to the CHNL_IMG_CTRL[DEINT] field to enable weave de-interlace with Odd frame first or Even frame first, respectively.
- ***Line Doubling -*** Two channels need to be configured to perform line double de-interlacing. Configure line doubling de-interlace method by writing "11x" to the CHNL_IMG[DEINT] field to enable line doubling in each field to form a complete de-interlaced frame. Both channels should be configured with the same configuration except for the final output memory address which should differ by one line pitch between the channels. If the address in one channel is configured as "A", then the other channel should have the address configured as "A + line_pitch". The line pitch is configured in the CHNL_OUT_BUF_PITCH[LINE_PITCH] field.
- ***Blending (linear interpolation) -*** Two channels need to be configured when performing blend de-interlacing. One channel performs weaving and the other channel (now referred to as read channel) performs the averaging. Configure blend de-interlace method by writing "100" or "101" to the CHNL_IMG_CTRL[DEINT] field of that channel's register that performs the weaving operation (this is also the channel that aources the image from the interlaced camera sensor). The read channel which performs the averaging should have this field set to "00x", i.e. no de-interlacing (this channel sources the image from memory).
    - Configure the temporary memory location as output address, line pitch and frame pitch in CHNL_IN_BUF_ADDR[ADDR], CHNL_IN_BUF_PITCH[FRM_PITCH] and

CHNL_IN_BUF_PITCH[LINE_PITCH] respectively, for the channel performing the weave operation. Software must allocate enough memory to store two frames with a frame pitch before the previous frames get overwritten. CHNL_OUT_BUF_PITCH, CHNL_OUT_BUF1_ADDR_U/V and CHNL_OUT_BUF2_ADDR_Y/U/V are not used.

- Set the output format in CHNL_IMG_CTRL[FORMAT] field of the channel performing the weave operation to one of the formats mentioned in CHNL_MEM_RD_CTRL[IMG_TYPE] field. This will ensure that output of weave channel is same as that can be read by the read channel.
- Configure the read channel CHNL_IN_BUF_ADDR[ADDR] and CHNL_IN_BUF_PITCH to point to the temporary memory location and pitch values (which are CHNL_OUT_BUF1_ADDR_Y[ADDR], CHNL_IN_BUF_PITCH of the weave channel).
- Configure the read channel CHNL_SCALE_OFFSET[Y_OFFSET] to 0.5 to perform averaging of the lines.
- Configure the read channel CHNL_OUT_BUF1_ADDR_*[ADDR], CHNL_OUT_BUF2_ADDR_*[ADDR] and CHNL_OUT_BUF_PITCH[LINE_PITCH] registers to configure the final output memory location of the blended frame.
- Configure the read channel CHNL_MEM_RD_CTRL[IMG_TYPE] same as the channel output format of the channel performing the weaving operation. There is no need to configure the CHNL_MEM_RD_CTRL[READ_MEM] bit in channel 0 for blend de-interlacing since the weaving channel automatically initiates the reads from the read channel.

- Return to initialization steps to complete the initialization of other blocks and enabling the channel.

NOTE

For all channels which have the CHNL_IMG_CTRL[DEINT] set to values other than "00x; the scaler & CSC will be automatically bypassed. However, the read channel scaler and CSC will work for blend de-interlacing since we do not set the CHNL_IMG_CTRL[DEINT] field for the read channel.

NOTE

Other image operations (alpha insertion, cropping, horizontal and vertical flipping) can still be carried out along with de-interlacing with the only exception that these operations *should not* be done in the weaving channel during blending de-interlace. They can be carried out on the final blended image frame in the read channel.

13.4.3.5 High-Res Image Handling

Each channel in ISI is capable of handling up to 4K image resolution. However, the line buffer and output buffer storage in one channel is enough for 2K image resolution. In order to be able to process and store a 4K image, line buffer and output buffers from adjacent channel need to be chained with the current channel's line buffer and output buffers respectively. Chaining of line buffers and output buffers has a limitation that only the subsquent channels can be chained. For example, if channel 1 has to handle a 4K image, its line buffer and output buffers can be chained to line buffer and output buffers from channel 2, respectively and no other channel. In generic terms, line buffer and output buffers of channel 'n' can be chained with line buffer and output buffers, respectively, of channel 'n+1' only.

NOTE

If a channel's line buffer and output buffers are chained with another channel, then that channel should be kept disabled and cannot be used for any image processing.

Channel initialization is a prerequisite to configuring chain control in the channel. Steps to configure chaining of line buffers are as follows:

- Configure the channel's CHNL_CTRL[CHAIN_BUF] field to "01" to support a line resolution up to 4096 or 4K pixels.
- Ensure that the CHNL_CTRL[CHNL_EN] = 0 for the channel whose line buffer has been chained with a previous channel's line buffer.
- Return to initialization steps to complete the initialization of other blocks and enabling the channel.

13.4.3.6 Output Buffer Management

The output buffer block in ISI provides some further image processing options, like:

- ***Alpha Insertion in RGB fomrats***
  - ISI supports global alpha value insertion to RGB images in which one alpha value is inserted to all pixels in the image.
  - ISI supports up to 4 non-overlapping regions of interests or ROIs in which different alpha values can be inserted.
  - If both ROI Alpha and Global Alpha values are enabled, then ROI Alpha values takes priority over Global Alpha value.

  - Pixels that do not lie in any of the ROI can have a global alpha value (if enabled) or not have any alpha value inserted (if global alpha insertion is disabled).
  - The Upper Left Coordinate (ULC) and the Lower Right Coordinate (LRC) can be configured individually for each ROI. The ULC and LRC should lie within the dimensions specified by the scaled image configuration register (CHNL_SCL_IMG_CFG).
- ***Image Flipping***
  - ISI supports horizontal and vertical image flipping.
  - Either one or both flipping can be enabled.
- ***Image Cropping***
  - ISI support cropping in both X and Y dimensions based on the programmed ULC and LRC. The ULC and LRC should lie within the dimensions specified by the scaled image configuration register (CHNL_SCL_IMG_CFG).

NOTE

The order in which the above operations are carried out are:
Alpha Insertion, Cropping and Flipping.

The above image manipulation operations are carried out before the image is stored into the output buffers. The image input to the output buffer block (from scaler/CSC) is in either RGB or YUV444. The output buffer logic then formats each pixel from these two formats to the desired output format as specified in the CHNL_IMG_CTRL[FORMAT] field. The output buffers consolidate the formatted image pixel bytes and send out 128-bytes or 256-bytes AXI Write Bursts to store the image into memory. ISI has in-built pre-overflow warning mechanism through which it can warn the subsystem and request higher AXI write channel priority to avoid any overflows in its buffers. In any case, if an overflow does occur, ISI will stop storing pixels for that line and starts storing pixels from the next line. This will cause the frame to be corrupted and software should take appropriate action for such frames. For images sourced from memory, ISI can stall its internal pipeline or control the rate at which pixels are processed in the pipeline. More details in Flow Control section.

There are two destination addresses (CHNL_OUT_BUF1_ADDR_* and CHNL_OUT_BUF2_ADDR_*) that need to be programmed. The BUF1 address is used when the channel is enabled for the first time. This address is internally incremented by a line pitch (CHNL_OUT_BUF_PITCH) at end of each line, except last line, to compute the address for each line storage. ISI generates a line_stored interrupt when a full line is completely stored and a frame_stored after the last line is completely stored in memory. The next frame is stored at BUF2 address in the same way. Software may update CHNL_OUT_BUF1_ADDR_* at this point, if desired. On every frame_stored interrupt, the ISI toggles between the two buffer addresses. Status bits provide indication to

software on which buffer is currently active. Software can update the buffer address regiters at any time but they will take effect only on when corresponding load address bits are toggled. The new address gets loaded when the current frame completes.

The ISI automatically manages the output buffer size depending on the output image format. For 2 and 3-planar formats, the output buffer is divided accordingly to each component based on amount of bytes used by the component. For 1-planar, RGB and RAW, the output buffer function as a unified buffer. This allows efficient utilization of output buffers even when they are concatenated.

13.4.3.6.1 Configuring additional image processing options

Please follow the channel initialization steps prior to configuring the additional image processing controls. To configure the image processing options:

- ***Alpha Insertion***
  - Configure CHNL_IMG_CTRL[GBL_ALPHA_VAL] to set the global alpha value and set CHNL_IMG_CTRL[GBL_ALPHA_EN] = '1' if a global alpha value is to be inserted in the image. This should be set when both global and regional alpha values are to be inserted.
  - Configure CHNL_ROI_n_ALPHA[ALPHA] and CHNL_ROI_n_ALPHA[ALPHA_EN] = 1 (where 'n' = 0, 1, 2, or 3) to have region specific alpha values.
  - Configure the region of interest coordinates in CHNL_ROI_n_ULC and CHNL_ROI_n_LRC registers, where 'n' = 0, 1, 2, 3. Regions should not overlap nor should they lie outside the scaled image dimensions.
- ***Image Flipping***
  - Configure horizontal flipping of output image by setting CHNL_IMG_CTRL[HFLIP_EN] = 1.
  - Configure vertical flipping of output image by setting CHNL_IMG_CTRL[VFLIP_EN] = 1.
- ***Image Cropping***
  - Configure the cropped region of image in CHNL_CROP_ULC and CHNL_CROP_LRC registers. This is region that will be stored to memory.
  - Enable cropping by setting CHNL_IMG_CTRL[CROP_EN] = 1.

NOTE

The ULC and LRC (both for alpha insertion and cropping) should be configured based on the image resolution output from scaler and not the original image resolution. The output buffer block works on the output from CSC (when no bypass) or Scaler (if CSC is bypassed). Only during a channel bypass or 1:1 scaling should the original resolution be used for determining the ULC and LRC.

13.4.3.6.2 Configuring output buffer controls

Please follow the channel initialization steps prior to configuring the output buffer controls. To configure the output buffer controls:

- Output Memory Addressing
  - Configure the destination memory address in the CHNL_OUT_BUF1_ADDR_n and CHNL_OUT_BUF2_ADDR_n (where n = Y, U, V).
  - Configure the output image line pitch in CHNL_OUT_BUF_PITCH[LINE_PITCH]. The line pitch is the spacing between two lines in terms of number of bytes.
  - Toggle the address load bits i.e. CHNL_OUT_BUF_CTRL[LOAD_BUF1_ADDR] and CHNL_OUT_BUF_CTRL[LOAD_BUF2_ADDR] to allow new configured addresses to be picked and used in the channel.
  - On every CHNL_STS[FRM_STRD] interrupt assertion, software must update the appropriate buffer 1 or buffer 2 address. If this is not done, the same address will be used and frame will be overwritten.
  - On every update to the CHNL_OUT_BUF1_ADDR_n or CHNL_OUT_BUF2_ADDR_n registers, the appropriate address load bits should be set to '1'. Without setting the load bit, the new address will not be picked.
  - The active buffer to which the current frame is being stored to is indicated via status bits - CHNL_STS[BUF1_ACTIVE], CHNL_STS[BUF2_ACTIVE].

NOTE

When performing de-interlace, the output memory addressing changes for that channel. Please refer to Image De-interlacing section for configuring the destination memory addresses for that channel.

NOTE

Ensure that line pitch is programmed to be greater than number of bytes in one line.

- Return to initialization steps to complete the initialization of other blocks and enabling the channel.

13.4.3.7 Sourcing Image from Memory

A channel can be configured to read image from memory in two cases:

- During Blending De-interlace, hardware controlled memory reads occur. Refer to Image Deinterlacing section for more details.
- Software controlled memory read.

This section details the configuration required for software controlled memory reads. Please follow the channel initialization steps prior to configuring the memory image read on a channel. The steps to configure channel are:

- Configure the memory address from where the image is to be sourced in the CHNL_IN_BUF_ADDR[ADDR].
- Configure the line and frame pitch in the CHNL_IN_BUF_PITCH register.
- Configure the read image format in the CHNL_MEM_RD_CTRL[IMG_TYPE] field.
- Complete the initialization of other blocks and interrupts for channel 0 and set CHNL_CTRL[CHNL_EN] = 1.
- Set the CHNL_MEM_RD_CTRL[READ_MEM] = 1 to enable reading of frame from memory.
- Wait for CHNL_STS[MEM_RD_DONE] interrupt to set before initiating a new image read. When this interrupt occurs, set CHNL_MEM_RD_CTRL[READ_MEM] = 0 followed by CHNL_MEM_RD_CTRL[READ_MEM] = 1 to start the next read.

NOTE

- When hardware controlled reading is enabled, then software controlled reads will not be available, and vice-versa.
- Only one channel can perform read at a given time.

13.4.3.8 Flow Control

The ISI implements the following flow control mechanisms to compensate for low bandwidth on the device's AXI bus:

- **Panic and overflow**
  - ISI provides a "panic" signal to the subsystem to request for higher priority on the AXI write channel to empty the ping pong buffers before they overflow. The assertion of panic signal can be set to assert when the output ping pong buffers get filled to a certain configurable threshold. Depending on the reaction time from subsystem, the appropriate threshold can be set in the CHNL_OUT_BUF_CTRL[OFLW_PANIC_SET_THD_n] bits, where n = Y, U, V. The panic signal de-asserts when the amount of data inside the ping pong buffers goes below the configurable threshold.
  - The panic status is also indicated via interrupt bits - CHNL_STS[PANIC_n_BUF], where n = Y, U, V. The status bits do not clear if the amount of data inside the ping pong buffers goes below the configurable threshold.

    - In case an overflow does occur, ISI aborts storing pixels from the current line and resyncs to the next line and starts storing pixels thereon. The status of overflow in buffers is indicated via interrupt bits - CHNL_STS[OFLW_n_BUF], where n = Y, U, V.
    - Enable overflow and panic interrupts for the channel in the CHNL_IER register.
- **Back pressure to stall internal pipeline**
    - ISI implements an automatic back pressure to stall the channel's pipeline in case the AXI bus is not ready to accept data and the ping pong buffers have 128-bytes of space left.
    - The back pressure to the channel pipeline is asserted when scaler line buffers are flushed between frames or image is sourced from memory.
    - However, data can still be lost during scaler line buffer flushing if the back pressure stalls the pipeline for too long. This could happen when a new frame starts and flush is not completed due to back pressure. In this case, ISI moves to process the next frame and indicates an overflow error.
- **Adjustable flow rate**
    - ISI provides registers to configure the rate at which pixels are processed in the pipeline when image is sourced from memory and/or when scaler line buffers are flushed. By default the rate is one pixel per clock, assuming read buffers are full with data.
    - Normally, the camera rate would be less than 1 pixel per clock and during flushing the high rate can peak the bandwidth demand on AXI bus. Using the CHNL_FLOW_CTRL register, software can program the rate to match the camera rate as desired.
    - The rate is programmed as a ratio in the CHNL_FLOW_CTRL[FC_NUMER] and CHNL_FLOW_CTRL[FC_DENOM]. ISI will send pixels for FC_NUMER number of clocks out of FC_DENOM clocks.

Return to initialization steps to complete the initialization of other blocks and enabling the channel.

13.4.3.9 Error Resiliency

ISI implements resiliency against possible error frames and recovery from them. An error frame can have one of the following:

- HSYNC at the wrong time, thereby causing less/more pixels in frame. ISI recovers silently and there is no indication of this error event.

- VSYNC at the wrong time, thereby causing less/more lines in frame. ISI asserts CHNL_STS[EARLY_VSYNC] or CHNL_STS[LATE_VSYNC] interrupts bits to indicate to the software about this condition and ISI resyncs to the next frame. Enable the interrupt by writing '1' to the corresponding interrupt enable bit in the CHNL_IER regsiter.
- Unequal lines between interlaced fields. The ISI expects both frames to have equal number of lines and will assert CHNL_STS[EARLY_VSYNC] for the interlaced frame that has less lines that the other frame.
- Missing or extra interlaced field (odd or even). ISI will mask out the unexpected field and not flag any error in this event.
- Frame with no pixels. ISI recovers silently and there is no indication of this error event.
- Frame line with no pixels. ISI recovers silently and there is no indication of this error event.

All the error resiliency mechanisms ensure that the frame is not stored outside its given memory range in the event of an error frame being received by ISI.

13.4.3.10 Clocks

The table below shows the clocks described in this module:

Table 13-4. Clocks

Clocks	Description
Pixel Clock	The pixel link clock comes from the Camera via the pixel link interface.
APB Clock	APB clock is for register accesses.
AXI Clock	AXI clock is the clock on AXI bus at the output.

13.4.3.10.1 Clocking constraint

The clocking requirement for ISI module is that the processing clock must be greater than or equal to the incoming pixel link clock. The reason for this is that ISI can process one pixel per processing clock while the incoming pixel link can input one pixel per pixel link clock. If the incoming pixel link is faster than the processing clock, then the ISI can lose data as data is coming in faster than it can process it.

13.4.4 External Signals

There are no device I/O signals associated with this module.

13.4.5 Memory Map and register definition

This section includes the ISI module memory map and detailed descriptions of all registers.

NOTE

Software must ensure correctness of values being programmed to ensure smooth operation. The module does not check for correctness.

13.4.5.1 ISI Memory Map register descriptions

The memory map comprises of 32-bit aligned registers which can be accessed via 32-bit accesses only. Write access to reserved locations will generate transfer error. Read access to reserved locations will also generate transfer error and the read data bus will show all 0s.

The module will not check for correctness of programmed values in the registers and software must ensure that correct values are being written.

13.4.5.1.1 ISI memory map

ISI base address: 32E0_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Channel Control Register (CHNL_CTRL)	32	RW	0000_0000h
4h	Channel Image Control Register (CHNL_IMG_CTRL)	32	RW	0000_0000h
8h	Channel Output Buffer Control Register (CHNL_OUT_BUF_CTRL)	32	RW	0000_0000h
Ch	Channel Image Configuration (CHNL_IMG_CFG)	32	RW	0438_0780h
10h	Channel Interrupt Enable Register (CHNL_IER)	32	RW	0000_0000h
14h	Channel Status Register (CHNL_STS)	32	W1C	0000_0000h
18h	Channel Scale Factor Register (CHNL_SCALE_FACTOR)	32	RW	1000_1000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1Ch	Channel Scale Offset Register (CHNL_SCALE_OFFSET)	32	RW	0000_0000h
20h	Channel Crop Upper Left Corner Coordinate Register (CHNL_CROP_ULC)	32	RW	0000_0000h
24h	Channel Crop Lower Right Corner Coordinate Register (CHNL_CROP_LRC)	32	RW	0000_0000h
28h	Channel Color Space Conversion Coefficient Register 0 (CHNL_CSC_COEFF0)	32	RW	0000_0000h
2Ch	Channel Color Space Conversion Coefficient Register 1 (CHNL_CSC_COEFF1)	32	RW	0000_0000h
30h	Channel Color Space Conversion Coefficient Register 2 (CHNL_CSC_COEFF2)	32	RW	0000_0000h
34h	Channel Color Space Conversion Coefficient Register 3 (CHNL_CSC_COEFF3)	32	RW	0000_0000h
38h	Channel Color Space Conversion Coefficient Register 4 (CHNL_CSC_COEFF4)	32	RW	0000_0000h
3Ch	Channel Color Space Conversion Coefficient Register 5 (CHNL_CSC_COEFF5)	32	RW	0000_0000h
40h	Channel Alpha Value Register for Region of Interest 0 (CHNL_ROI_0_ALPHA)	32	RW	0000_0000h
44h	Channel Upper Left Coordinate Register for Region of Interest 0 (CHNL_ROI_0_ULC)	32	RW	0000_0000h
48h	Channel Lower Right Coordinate Register for Region of Interest 0 (CHNL_ROI_0_LRC)	32	RW	0000_0000h
4Ch	Channel Alpha Value Register for Region of Interest 1 (CHNL_ROI_1_ALPHA)	32	RW	0000_0000h
50h	Channel Upper Left Coordinate Register for Region of Interest 1 (CHNL_ROI_1_ULC)	32	RW	0000_0000h
54h	Channel Lower Right Coordinate Register for Region of Interest 1 (CHNL_ROI_1_LRC)	32	RW	0000_0000h
58h	Channel Alpha Value Register for Region of Interest 2 (CHNL_ROI_2_ALPHA)	32	RW	0000_0000h
5Ch	Channel Upper Left Coordinate Register for Region of Interest 2 (CHNL_ROI_2_ULC)	32	RW	0000_0000h
60h	Channel Lower Right Coordinate Register for Region of Interest 2 (CHNL_ROI_2_LRC)	32	RW	0000_0000h
64h	Channel Alpha Value Register for Region of Interest 3 (CHNL_ROI_3_ALPHA)	32	RW	0000_0000h
68h	Channel Upper Left Coordinate Register for Region of Interest 3 (CHNL_ROI_3_ULC)	32	RW	0000_0000h
6Ch	Channel Lower Right Coordinate Register for Region of Interest 3 (CHNL_ROI_3_LRC)	32	RW	0000_0000h
70h	Channel RGB or Luma (Y) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_Y)	32	RW	0000_0000h
74h	Channel Chroma (U/Cb/UV/CbCr) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_U)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
78h	Channel Chroma (V/Cr) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_V)	32	RW	0000_0000h
7Ch	Channel Output Buffer Pitch (CHNL_OUT_BUF_PITCH)	32	RW	0000_0000h
80h	Channel Input Buffer Address (CHNL_IN_BUF_ADDR)	32	RW	0000_0000h
84h	Channel Input Buffer Pitch (CHNL_IN_BUF_PITCH)	32	RW	0000_0000h
88h	Channel Memory Read Control (CHNL_MEM_RD_CTRL)	32	RW	0000_0000h
8Ch	Channel RGB or Luma (Y) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_Y)	32	RW	0000_0000h
90h	Channel Chroma (U/Cb/UV/CbCr) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_U)	32	RW	0000_0000h
94h	Channel Chroma (V/Cr) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_V)	32	RW	0000_0000h
98h	Channel Scaled Image Configuration (CHNL_SCL_IMG_CFG)	32	RW	0438_0780h
9Ch	Channel Flow Control Register (CHNL_FLOW_CTRL)	32	RW	0000_0000h
A0h	Channel Output Y-Buffer 1 Extended Address Bits Register (CHNL_Y_BUF1_XTND_ADDR)	32	RW	0000_0000h
A4h	Channel Output U-Buffer 1 Extended Address Bits Register (CHNL_U_BUF1_XTND_ADDR)	32	RW	0000_0000h
A8h	Channel Output V-Buffer 1 Extended Address Bits Register (CHNL_V_BUF1_XTND_ADDR)	32	RW	0000_0000h
ACh	Channel Output Y-Buffer 2 Extended Address Bits Register (CHNL_Y_BUF2_XTND_ADDR)	32	RW	0000_0000h
B0h	Channel Output U-Buffer 2 Extended Address Bits Register (CHNL_U_BUF2_XTND_ADDR)	32	RW	0000_0000h
B4h	Channel Output V-Buffer 2 Extended Address Bits Register (CHNL_V_BUF2_XTND_ADDR)	32	RW	0000_0000h
B8h	Channel Input Buffer Extended Address Bits Register (CHNL_IN_BUF_XTND_ADDR)	32	RW	0000_0000h

13.4.5.1.2 Channel Control Register (CHNL_CTRL)

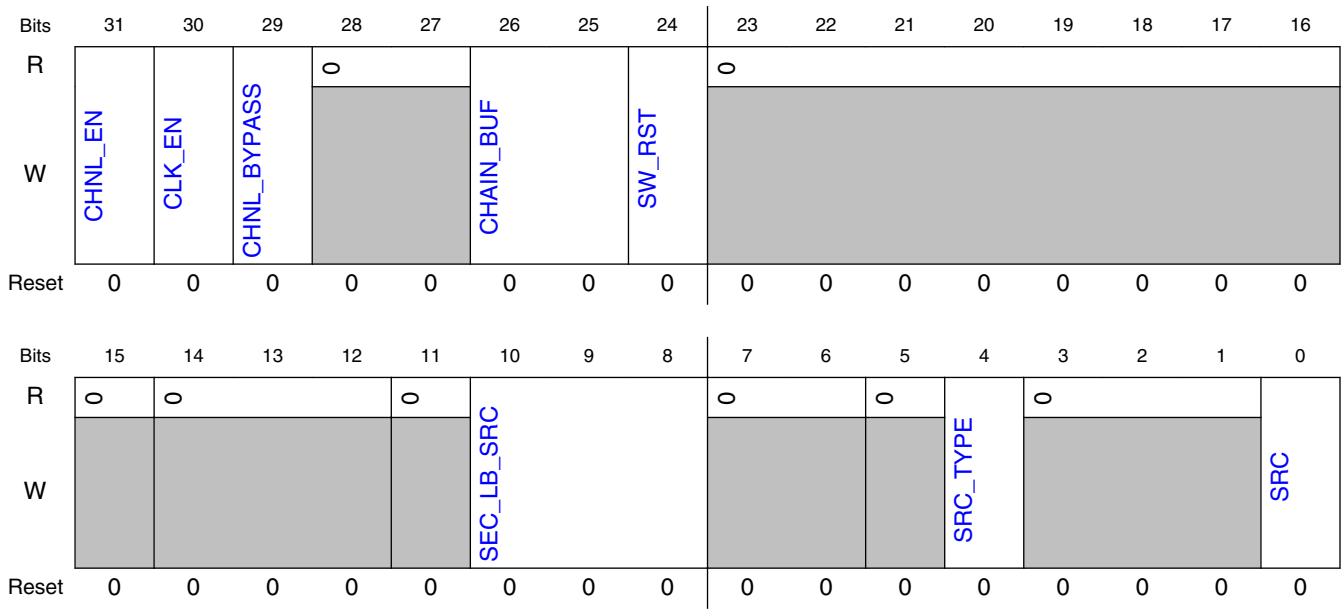
13.4.5.1.2.1 Offset

Register	Offset
CHNL_CTRL	0h

13.4.5.1.2.2 Function

This register provides the global settings for the channel pipeline and controls how the internal blocks and line buffer perform the image processing.

13.4.5.1.2.3 Diagram



13.4.5.1.2.4 Fields

Field	Function
31 CHNL_EN	Enable channel processing Enables the image processing in the channel. This bit must be set last. All other channel configuration must happen prior to setting this bit. 0b - Processing channel is disabled 1b - Processing channel is enabled
30 CLK_EN	Channel clock enable This bit controls the clock enable for the channel processing clock. When set to '1' the clock gating (outside the ISI block) is disabled and clock to the channel is enabled. 0b - Channel processing clock is disabled 1b - Channel processing clock is enabled
29 CHNL_BYPASS	Channel bypass enable For certain applications, the ISI module might not be required to do any scaling nor color space conversion on the image. For such cases, this bit should be set to bypass the channel pipeline and store the image into memory. The output image format CHNLn_IMG_CTRL[FORMAT] field should be set to same as the incoming image format. Image flipping, cropping and alpha insertion can still be carried out. **NOTE:** For de-interlacing purposes, the channel is automatically bypassed for one frame; but this bit should not be set for de-interlacing **NOTE:** If this bit is set, CSC will be bypassed even if CHNL_IMG_CTRL[CSC_BYP] is not set. 0b - Channel is not bypassed 1b - Channel is bypassed
28-27 —	Reserved field. Reads only zeros

Table continues on the next page...

Field	Function
26-25 CHAIN_BUF	Chain line buffer control For images having higher than 2048 horizontal resolution (pixels), the line buffer RAM (for channel 'n') will require to be concatenated with channel 'n+1' line buffer RAM to provide a larger storage. This field controls how the adjacent channel line buffer RAM are chained together. For example when this field is set to '01', line buffers (for channel 'n' and 'n+1') are chained; the channel 'n+1' cannot be enabled and are blocked for use. 00b - No line buffers chained (supports 2048 or less horizontal resolution) 01b - 2 line buffers chained (supports 4096 horizontal resolution). Line buffers of channels 'n' and 'n +1' are chained. 10b - Reserved for future use 11b - Reserved for future use
24 SW_RST	Software reset bit Writng a '1' to this bit will reset the processing pipeline similar to a hardware reset. This bit does not self clear and hence when writing a '1', software must allow suffucient delay (4 to 6 NOP cycles) for reset propagation before writing a '0' to disable the software reset. **NOTE:** Asserting the software reset does not clear the registers. It will affect the processing pipeline and the Memory Read Control. Any on-going AXI transactions will complete and will not be interrupted by the software reset. 0b - No Reset 1b - Channel pipeline is under software reset
23-15 —	Reserved field. Reads only zeros
14-12 —	Reserved field. Reads only zeros
11 —	Reserved field. Reads only zeros
10-8 SEC_LB_SRC	Secondary line buffer source Specifies the channel number whose line buffer will be used as the secondary line buffer in the primary channel performing YUV420 scaling. This field will be ignored for other cases. When this field is set, the primary channel takes over control of the secondary channel's line buffer. The secondary channel cannot be used for any processing. **NOTE:** When performing YUV420 scaling, care should be given that an existing functional channel's line buffer should not be used including the same channel itself. Doing so could result in unexpected results in both channels. Note that the line buffers are the internal storage of the channel and not the output memory buffers.
7-6 —	Reserved bits. Reads only zeros.
5 —	Reserved field. Reads only zeros
4 SRC_TYPE	Type of selected input image source Defines the type of image input source. Should be set to '1' only when source of image is from memory. For all other sources, this bit is '0'. 0b - Image input source is Pixel Link 1b - Image input source is Memory
3-1 —	Reserved field. Reads only zeros

Table continues on the next page...

Field	Function
0 SRC	Input image source port selection Source selects the desired input port of the Pixel Link Crossbar from which the image will be sourced. This field should not be changed when the channel pipeline is working and must be configured prior to enabling the channel processing. 0b - Image will be sourced from input port 0 of the Pixel Link Crossbar 1b - Image will be sourced from input port 1 of the Pixel Link Crossbar When SRC_TYPE is set to select Memory as the image input source, this bit is ignored.

13.4.5.1.3 Channel Image Control Register (CHNL_IMG_CTRL)

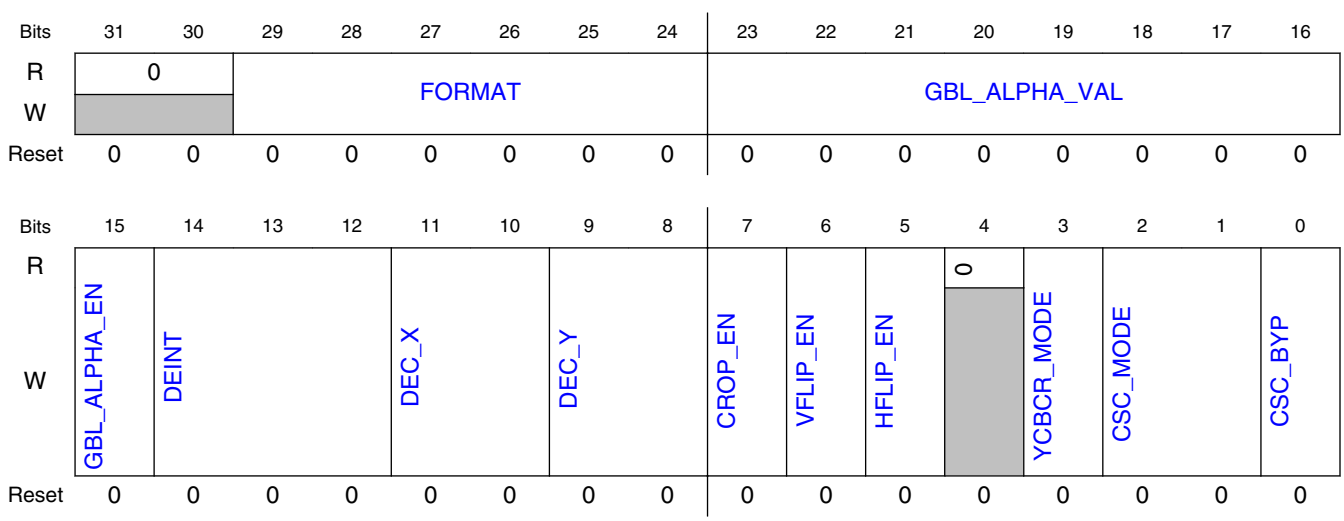
13.4.5.1.3.1 Offset

Register	Offset
CHNL_IMG_CTRL	4h

13.4.5.1.3.2 Function

This register prvoides bits that control how the input image is processed in the processing pipeline/channel.

13.4.5.1.3.3 Diagram



13.4.5.1.3.4 Fields

Field	Function
31-30 RSVD0	Reserved field. Reads only zeros
29-24 FORMAT	Output image format This field specifies the image format to be used when storing the image output from channel pipeline to memory. The UV byte lanes are synonymous with CbCr byte lanes for YUV output pixel formats. For example, the YUV420_2P* formats should be selected when the output is YCbCr 2-plane 420 output format. **NOTE:** For RAW pixel formats, if the output pixel format has few bits per pixel than the input pixel format, the most significant bits of each pixel are truncated by the ISP. 000000b - RGBA8888 - RGB format with alpha in LSB; 8-bits per component. 'A' indicates alpha value. 000001b - ABGR8888 - BGR format with alpha in MSB; 8-bits per component. 'A' indicates alpha value. 000010b - ARGB8888 - RGB format with alpha in MSB; 8-bits per component. 'A' indicates alpha value. 000011b - RGBX888 - RGB format with 8-bits per color component (unpacked and MSB-alinged in 32-bit DWORD). 'X' indicates the waste bits. 000100b - XBGR888 - BGR format with 8-bits per color component (unpacked and LSB aligned in 32-bit DWORD). 'X' indicates the waste bits. 000101b - XRGB888 - RGB format with 8-bits per color component (unpacked and LSB aligned in 32-bit DWORD). 'X' indicates the waste bits. 000110b - RGB888P - RGB format with 8-bits per color component (packed into 24-bits). No waste bits. 000111b - BGR888P - BGR format with 8-bits per color component (packed into 24-bits). No waste bits. 001000b - A2BGR10 - BGR format with 2-bits alpha in MSB; 10-bits per color component. 'A' indicates alpha value. 001001b - A2RGB10 - RGB format with 2-bits alpha in MSB; 10-bits per color component. 'A' indicates alpha value. 001010b - RGB565 - RGB format with 5-bits of R, B; 6-bits of G (packed into 16-bits WORD). No waste bits. 001011b - RAW8 - 8-bit RAW data packed into 32-bit DWORD 001100b - RAW10 - 10-bit RAW data packed into 16-bit WORD with 6 LSBs waste bits 001101b - RAW10P - 10-bit RAW data packed into 32-bit DWORD 001110b - RAW12 - 12-bit RAW data packed into 16-bit DWORD with 4 LSBs waste bits 001111b - RAW16 - 16-bit RAW data packed into 32-bit DWORD 010000b - YUV444_1P8P with 8-bits per color component; 1-plane, YUV interleaved packed bytes 010001b - YUV444_2P8P with 8-bits per color component; 2-plane, UV interleaved packed bytes 010010b - YUV444_3P8P with 8-bits per color component; 3-plane, non-interleaved packed bytes 010011b - YUV444_1P8 with 8-bits per color component; 1-plane YUV interleaved unpacked bytes (8 MSBs waste bits in 32-bit DWORD) 010100b - YUV444_1P10 with 10-bits per color component; 1-plane, YUV interleaved unpacked bytes (6 LSBs waste bits in 16-bit WORD) 010101b - YUV444_2P10 with 10-bits per color component; 2-plane, UV interleaved unpacked bytes (6 LSBs waste bits in 16-bit WORD) 010110b - YUV444_3P10 with 10-bits per color component; 3-plane, non-interleaved unpacked bytes (6 LSBs waste bits in 16-bit WORD) 010111b - Reserved for future use 011000b - YUV444_1P10P with 10-bits per color component; 1-plane, YUV interleaved packed bytes (2 MSBs waste bits in 32-bit DWORD) 011001b - YUV444_2P10P with 10-bits per color component; 2-plane, UV interleaved packed bytes (2 MSBs waste bits in 32-bit DWORD)

Table continues on the next page...

Field	Function
	011010b - YUV444_3P10P with 10-bits per color component; 3-plane, non-interleaved packed bytes (2 MSBs waste bits in 32-bit DWORD) 011011b - Reserved for future use 011100b - YUV444_1P12 with 12-bits per color component; 1-plane, YUV interleaved unpacked bytes (4 LSBs waste bits in 16-bit WORD) 011101b - YUV444_2P12 with 12-bits per color component; 2-plane, UV interleaved unpacked bytes (4 LSBs waste bits in 16-bit WORD) 011110b - YUV444_3P12 with 12-bits per color component; 3-plane, non-interleaved unpacked bytes (4 LSBs waste bits in 16-bit WORD) 011111b - Reserved for future use 100000b - YUV422_1P8P with 8-bits per color component; 1-plane, YUV interleaved packed bytes 100001b - YUV422_2P8P with 8-bits per color component; 2-plane, UV interleaved packed bytes 100010b - YUV422_3P8P with 8-bits per color component; 3-plane, non-interleaved packed bytes 100011b - Reserved for future use 100100b - YUV422_1P10 with 10-bits per color component; 1-plane, YUV interleaved unpacked bytes (6 LSBs waste bits in 16-bit WORD) 100101b - YUV422_2P10 with 10-bits per color component; 2-plane, UV interleaved unpacked bytes (6 LSBs waste bits in 16-bit WORD) 100110b - YUV422_3P10 with 10-bits per color component; 3-plane, non-interleaved unpacked bytes (6 LSBs waste bits in 16-bit WORD) 100111b - Reserved for future use 101000b - YUV422_1P10P with 10-bits per color component; 1-plane, YUV interleaved packed bytes (2 MSBs waste bits in 32-bit DWORD) 101001b - YUV422_2P10P with 10-bits per color component; 2-plane, UV interleaved packed bytes (2 MSBs waste bits in 32-bit DWORD) 101010b - YUV422_3P10P with 10-bits per color component; 3-plane, non-interleaved packed bytes (2 MSBs waste bits in 32-bit DWORD) 101011b - Reserved for future use 101100b - YUV422_1P12 with 12-bits per color component; 1-plane, YUV interleaved unpacked bytes (4 LSBs waste bits in 16-bit WORD) 101101b - YUV422_2P12 with 12-bits per color component; 2-plane, UV interleaved unpacked bytes (4 LSBs waste bits in 16-bit WORD) 101110b - YUV422_3P12 with 12-bits per color component; 3-plane, non-interleaved unpacked bytes (4 LSBs waste bits in 16-bit WORD) 101111b - Reserved for future use 110000b - Reserved for future use 110001b - YUV420_2P8P with 8-bits per color component; 2-plane, UV interleaved packed bytes 110010b - YUV420_3P8P with 8-bits per color component; 3-plane, non-interleaved packed bytes 110011b - Reserved for future use 110100b - Reserved for future use 110101b - YUV420_2P10 with 10-bits per color component; 2-plane, UV interleaved unpacked bytes (6 LSBs waste bits in 16-bit WORD) 110110b - YUV420_3P10 with 10-bits per color component; 3-plane, non-interleaved unpacked bytes (6 LSBs waste bits in 16-bit WORD) 110111b - Reserved for future use 111000b - Reserved for future use 111001b - YUV420_2P10P with 10-bits per color component; 2-plane, UV interleaved packed bytes (2 MSBs waste bits in 32-bit DWORD) 111010b - YUV420_3P10P with 10-bits per color component; 3-plane, non-interleaved packed bytes (2 MSBs waste bits in 32-bit DWORD) 111011b - Reserved for future use 111100b - Reserved for future use 111101b - YUV420_2P12 with 12-bits per color component; 2-plane, UV interleaved unpacked bytes (4 LSBs waste bits in 16-bit WORD) 111110b - YUV420_3P12 with 12-bits per color component; 3-plane, non-interleaved unpacked bytes (4 LSBs waste bits in 16-bit WORD) 111111b - Reserved for future use

Table continues on the next page...

Field	Function
23-16 GBL_ALPHA_VAL	Global alpha value This is the alpha value that will be inserted with all RGB pixels in the image when being stored into the memory. For RGB 10-bit formats, only 2 MSbits will be used as alpha value for insertion. When different alpha values are required to be inserted in upto 4 regions of interest then the ROI_x_ALPHA, ROI_x_ULC and ROI_x_LRC regsiters should be used. 00000000-11111111b - Alpha value to be inserted with all RGB pixels
15 GBL_ALPHA_EN	Global alpha value insertion enable This bit controls if the global alpha value will be inserted in the RGB image or not. When different alpha values are required to be inserted in upto 4 regions of interest then the ROI_x_ALPHA, ROI_x_ULC and ROI_x_LRC regsiters should be used. 0b - Global Alpha value insertion is disabled 1b - Global Alpha value insertion is enabled
14-12 DEINT	De-interlace control Frames that are interlaced when sent over the pixel link require de-interlacing. This field defines the basic de-interlacing method that will be used to combine the interlaced frames. Refer to Functional Description for more details on deinterlacing techniques and how to configure the channels to perform de-interlacing. 000, 001b - No de-interlacing done 010b - Weave de-interlacing (Odd, Even) method used 011b - Weave de-interlacing (Even, Odd) method used 100b - Blending or linear interpolation (Odd + Even) de-interlacing method used 101b - Blending or linear interpolation (Even + Odd) de-interlacing method used 110, 111b - Line doubling de-interlacing method used. Both Odd and Even fields are doubled.
11-10 DEC_X	Horizontal pre-decimation control This field controls the pre-decimation filter for down scaling of the input image in the horizontal direction. 00b - Pre-decimation filter is disabled. Bilinear scaling filter is still operational. 01b - Decimate by 2 10b - Decimate by 4 11b - Decimate by 8
9-8 DEC_Y	Vertical pre-decimation control This field controls the pre-decimation filter for down scaling of the input image in the vertical direction. 00b - Pre-decimation filter is disabled. Bilinear scaling filter is still operational. 01b - Decimate by 2 10b - Decimate by 4 11b - Decimate by 8
7 CROP_EN	Output image cropping enable When set to '1', this bit enables cropping of the image. Cropping of image is carried out after down scaling. Hence, the cropping coordinates should be programmed with respect to scaled image. The cropping coordinates are configured in the CROP_ULC and CROP_LRC registers. 0b - Image cropping is disabled 1b - Image cropping is enabled
6 VFLIP_EN	Veritical flip control When set to '1', this bit enables vertical image flipping. 0b - Vertical image flip disabled 1b - Vertical image flip enabled
5 HFLIP_EN	Horizontal flip control When set to '1', this bit enables horizontal image flipping. 0b - Horizantal image flip disabled 1b - Horizontal image flip enabled
4	Reserved field. Reads only zeros

Table continues on the next page...

Field	Function
RSVD2	
3 YCBCR_MODE	YCbCr Mode Set to 1 when performing YCbCr conversion to RGB. Set to 0 when converting RGB to YUV data. This bit changes the behavior of the scaler when performing U/V scaling. 0b - YCbCr mode is disabled 1b - YCbCr mode is enabled
2-1 CSC_MODE	Color Space Conversion operating mode This field controls how the CSC unit operates on pixels when the CSC is not bypassed. 00b - Convert from YUV to RGB 01b - Convert from YCbCr to RGB 10b - Convert from RGB to YUV 11b - Convert from RGB to YCbCr
0 CSC_BYP	Color Space Conversion bypass control When set to '1', the color space convertor block is bypassed and no operation occurs on the pixels. When set to '0', color conversion selected by the CSC_MODE bits takes place. 0b - CSC is operational 1b - CSC is bypassed

13.4.5.1.4 Channel Output Buffer Control Register (CHNL_OUT_BUF_CTRL)

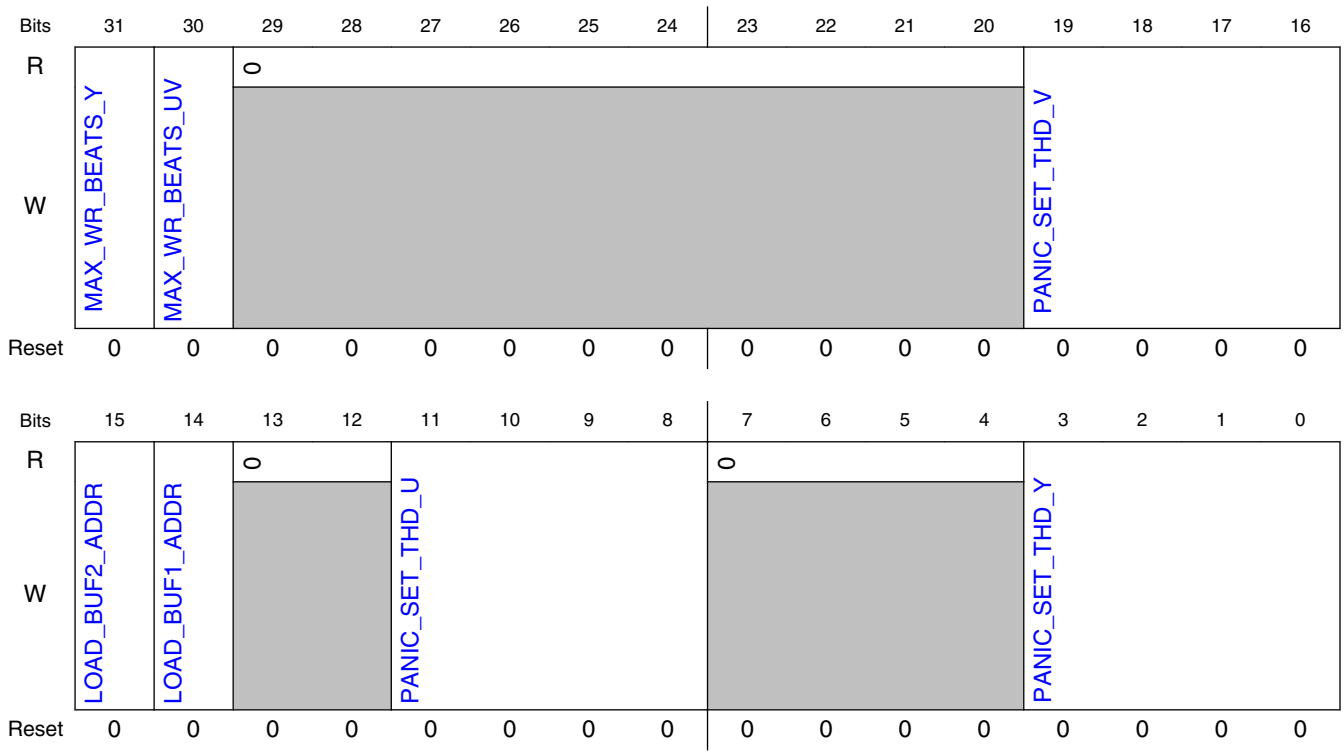
13.4.5.1.4.1 Offset

Register	Offset
CHNL_OUT_BUF_CTRL	8h

13.4.5.1.4.2 Function

This register controls how the output buffer loads the destination memory address and configures the overflow panic controls.

13.4.5.1.4.3 Diagram



13.4.5.1.4.4 Fields

Field	Function
31 MAX_WR_BEATS_Y	Maximum AXI write beats for Y-buffer When Y-buffer is concatenated with other buffers (i.e. Y+U+V or Y+V), software has the option to increase throughput by increasing the maxmium write beats to 16 (i.e. 256 bytes) per write request. When buffer is not combined, this bit has no effect and maximum of 8-beats per write request are supported. 0b - Maximum write beats per write request are 8 (i.e. 128 bytes) 1b - Maximum write beats per write request are 16 (i.e. 256 bytes)
30 MAX_WR_BEATS_UV	Maximum AXI write beats for U and V-buffers When U-buffer and V-buffer are concatenated, software has the option to increase throughput by increasing the maxmium write beats to 16 (i.e. 256 bytes) per write request. When buffers are not combined, this bit has no effect and maximum of 8-beats per write request are supported. 0b - Maximum write beats per write request are 8 (i.e. 128 bytes) 1b - Maximum write beats per write request are 16 (i.e. 256 bytes)
29-20 —	Reserved field. Reads only zeros
19-16 PANIC_SET_THD_V	Overflow panic set threshold value for V output buffer This field provides the threshold after which the Potential Overflow Panic Alert Interrupt (STS[OFLW_PANIC_V_BUF]) and Panic signal to subsystem will be asserted to request higher priority on AXI to drain the output buffers of the channel. Panic signal will de-assert when the amount of filled rows in ping pong buffer fall below the programmed threshold. Panic interrupt bit will clear when a '1' is written to it.

Table continues on the next page...

Field	Function
	0000b - No panic alert will be asserted 0001-1111b - Panic will assert when buffer is n * 6.25% full, where n = 1 to 15
15 LOAD_BUF2_ADDR	Load Buffer 2 Address from CHNLOUT_BUF2_ADDR_* registers When a new value is required to be used, software must toggle this bit to indicate a new buffer address has been programmed for use. On every toggle of this bit, the new address values will be shadowed in design.
14 LOAD_BUF1_ADDR	Load Buffer 1 Address from CHNLOUT_BUF1_ADDR_* registers When a new value is required to be used, software must toggle this bit to indicate a new buffer address has been programmed for use. On every toggle of this bit, the new address values will be shadowed in design.
13-12 —	Reserved field. Reads only zeros
11-8 PANIC_SET_THD_U	Overflow panic set threshold value for U output buffer This field provides the threshold after which the Potential Overflow Panic Alert Interrupt (STS[OFLW_PANIC_U_BUF]) and Panic signal to subsystem will be asserted to request higher priority on AXI to drain the output buffers of the channel. Panic signal will de-assert when the amount of filled rows in ping pong buffer fall below the programmed threshold. Panic interrupt bit will clear when a '1' is written to it. 0000b - No panic alert will be asserted 0001-1111b - Panic will assert when buffer is n * 6.25% full, where n = 1 to 15
7-4 —	Reserved field. Reads only zeros
3-0 PANIC_SET_THD_Y	Overflow panic set threshold value for Y/RGB output buffer This field provides the threshold after which the Potential Overflow Panic Alert Interrupt (STS[OFLW_PANIC_Y_BUF]) and Panic signal to subsystem will be asserted to request higher priority on AXI to drain the output buffers of the channel. Panic signal will de-assert when the amount of filled rows in ping pong buffer fall below the programmed threshold. Panic interrupt bit will clear when a '1' is written to it. 0000b - No panic alert will be asserted 0001-1111b - Panic will assert when buffer is n * 6.25% full, where n = 1 to 15

13.4.5.1.5 Channel Image Configuration (CHNL_IMG_CFG)

13.4.5.1.5.1 Offset

Register	Offset
CHNL_IMG_CFG	Ch

13.4.5.1.5.2 Function

This register provides the details about the image width (pixels) and height (lines). This register can be used to specify the image resolution for any type of source for the processing channel.

13.4.5.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			HEIGHT												
W																
Reset	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WIDTH												
W																
Reset	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0

13.4.5.1.5.4 Fields

Field	Function
31-29 RSVD1	Reserved field. Reads only zeros.
28-16 HEIGHT	Input image height (lines) This field provides the height of input image in terms of number of lines for both cases when the image is input from pixel link or from memory. Default height is 1080 lines.
15-13 RSVD0	Reserved field. Reads only zeros.
12-0 WIDTH	Input image width (pixels) This field provides the width of input image in terms of number of pixels for both cases when the image is input from pixel link or from memory. Default width is 1920 pixels.

13.4.5.1.6 Channel Interrupt Enable Register (CHNL_IER)

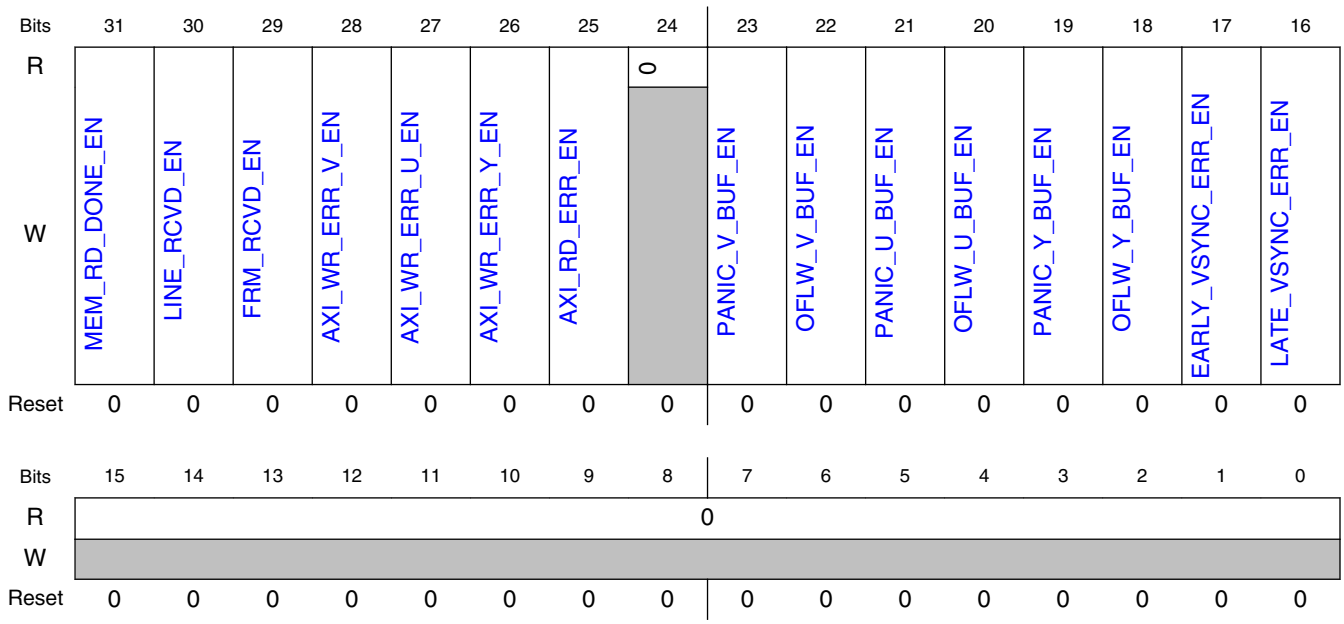
13.4.5.1.6.1 Offset

Register	Offset
CHNL_IER	10h

13.4.5.1.6.2 Function

This register provides bits to enable the generation of interrupt to the processor.

13.4.5.1.6.3 Diagram



13.4.5.1.6.4 Fields

Field	Function
31 MEM_RD_DONE_EN	Memory read complete interrupt enable bit Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
30 LINE_RCVD_EN	Line received interrupt enable bit Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
29 FRM_RCVD_EN	Frame received interrupt enable bit Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
28 AXI_WR_ERR_V_EN	AXI bus read error interrupt enable bit for V data buffer Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
27	AXI bus read error interrupt enable bit for U data buffer Enables assertion of interrupt from the channel when corresponding status bit is asserted.

Table continues on the next page...

Field	Function
AXI_WR_ERR_U_EN	0b - Interrupt is disabled 1b - Interrupt is enabled
26 AXI_WR_ERR_Y_EN	AXI bus read error interrupt enable bit for Y/RGB data buffer Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
25 AXI_RD_ERR_EN	AXI bus read error interrupt enable bit Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
24 —	Reserved
23 PANIC_V_BUF_EN	V output buffer potential overflow panic interrupt enable bit Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
22 OFLW_V_BUF_EN	V output buffer overflow interrupt enable bit Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
21 PANIC_U_BUF_EN	U output buffer potential overflow panic interrupt enable bit Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
20 OFLW_U_BUF_EN	U output buffer overflow interrupt enable bit Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
19 PANIC_Y_BUF_EN	Y output buffer potential overflow panic interrupt enable bit Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
18 OFLW_Y_BUF_EN	Y output buffer overflow interrupt enable bit Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
17 EARLY_VSYNC_ERR_EN	VSYNC timing (Early) error interrupt enable bit Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
16 LATE_VSYNC_ERR_EN	VSYNC timing (Late) error interrupt enable bit Enables assertion of interrupt from the channel when corresponding status bit is asserted. 0b - Interrupt is disabled 1b - Interrupt is enabled
15-0 RSVD0	Reserved field. Reads only zeros.

13.4.5.1.7 Channel Status Register (CHNL_STS)

13.4.5.1.7.1 Offset

Register	Offset
CHNL_STS	14h

13.4.5.1.7.2 Function

This register provides the interrupt bits and read only bits that give the status of the module.

13.4.5.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MEM_RD_DONE	LINE_STRD	FRM_STRD	AXI_WR_ERR_V	AXI_WR_ERR_U	AXI_WR_ERR_Y	AXI_RD_ERR	0	PANIC_V_BUF	OFLW_V_BUF	PANIC_U_BUF	OFLW_U_BUF	PANIC_Y_BUF	OFLW_Y_BUF	EARLY_VSYNC_ERR	LATE_VSYNC_ERR
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C		W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					MEM_RD_OFLOW	BUF2_ACTIVE	BUF1_ACTIVE	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.7.4 Fields

Field	Function
31	Memory read complete interrupt flag

Table continues on the next page...

Field	Function
MEM_RD_DONE	Indicates that the channel has read the full image from the input memory location (IN_BUF_ADDR) and software should configure the system for next image to be read from memory. 0b - Image read from memory not complete or not started 1b - Image read from memory completed
30 LINE_STRD	Line received and stored interrupt flag Indicates that a line has been received and stored into memory. This interrupt flag asserts for every line received and stored into memory. 0b - No new line received 1b - New line received and stored into memory
29 FRM_STRD	Frame stored successfully interrupt flag Indicates that a frame has been received and stored into memory. This interrupt flag will assert for every frame received and stored into memory. 0b - No frame being received or in progress 1b - One full frame has been received and stored in memory
28 AXI_WR_ERR_V	AXI Bus write error interrupt flag for V data buffer Indicates an error occured on the AXI Bus when storing V data to memory. 0b - No error 1b - Error occured during write
27 AXI_WR_ERR_U	AXI Bus write error interrupt flag for U data buffer Indicates an error occured on the AXI Bus when storing U data to memory. 0b - No error 1b - Error occured during write
26 AXI_WR_ERR_Y	AXI Bus write error interrupt flag for Y/RGB data buffer Indicates an error occured on the AXI Bus when storing Y/RGB data to memory. 0b - No error 1b - Error occured during write
25 AXI_RD_ERR	AXI Bus read error interrupt flag Indicates an error occured on the AXI Bus when reading image data to memory. 0b - No error 1b - Error occured during read
24 —	Reserved
23 PANIC_V_BUF	V output buffer potential overflow panic alert interrupt flag Indicates that the channel requires the software to increase the AXI bus priority for the V channel so that it can empty its output buffers otherwise the buffer might overflow and loose valid pixel data. Software must take action on this interrupt such that buffers are emptied out before any overflow occurs. The threshold for this panic interrupt can be set in the OUT_BUF_CTRL register. 0b - Buffer has not crossed the panic threshold limit 1b - Panic threshold limit crossed. Software must take action.
22 OFLW_V_BUF	Overflow in U output buffer interrupt flag Indicates overflow due to internal buffers not being read via AXI and are now full and not accepting new valid pixels in the output buffer control stage. The number of bytes lost during the overflow event is also reported in the status register. 0b - No overflow 1b - Overflow has occured in the channel
21 PANIC_U_BUF	U output buffer potential overflow panic alert interrupt flag

Table continues on the next page...

Field	Function
	Indicates that the channel requires the software to increase the AXI bus priority for the U channel so that it can empty its output buffers otherwise the buffer might overflow and loose valid pixel data. Software must take action on this interrupt such that buffers are emptied out before any overflow occurs. The threshold for this panic interrupt can be set in the OUT_BUF_CTRL register. 0b - Buffer has not crossed the panic threshold limit 1b - Panic threshold limit crossed. Software must take action.
20 OFLW_U_BUF	Overflow in U output buffer interrupt flag Indicates overflow due to internal buffers not being read via AXI and are now full and not accepting new valid pixels in the output buffer control stage. The number of bytes lost during the overflow event is also reported in the status register. 0b - No overflow 1b - Overflow has occured in the channel
19 PANIC_Y_BUF	Y/RGB output buffer potential overflow panic alert interrupt flag Indicates that the channel requires the software to increase the AXI bus priority for the Y/RGB channel so that it can empty its output buffers otherwise the buffer might overflow and loose valid pixel data. Software must take action on this interrupt such that buffers are emptied out before any overflow occurs. The threshold for this panic interrupt can be set in the OUT_BUF_CTRL register. 0b - Buffer has not crossed the panic threshold limit 1b - Panic threshold limit crossed. Software must take action.
18 OFLW_Y_BUF	Overflow in Y/RGB output buffer interrupt flag Indicates overflow due to internal buffers not being read via AXI and are now full and not accepting new valid pixels in the output buffer control stage. The number of bytes lost during the overflow event is also reported in the status register. 0b - No overflow 1b - Overflow has occured in the channel
17 EARLY_VSYNC_ERR	VSYNC timing (Early) error interrupt flag Indicates that the VSYNC for the frame was detected earlier than expected. The resultant frame will have corrupt data. 0b - No error 1b - VSYNC detected earlier than expected
16 LATE_VSYNC_ERR	VSYNC timing (Late) error interrupt flag Indicates that the VSYNC for the frame was detected later than expected. The resultant frame will have correct data. The extra lines in frame will be ignored. 0b - No error 1b - VSYNC detected later than expected
15-11 RSVD1	Reserved field. Reads only zeros.
10 MEM_RD_OFLOW	Memory read FIFO overflow error status Read only bit that indicates that read from memory has caused an internal FIFO to overflow. Data loss has occurred during memory read. Status is valid only when a memory read has been initiated. 0b - No overflow occurred during memory read 1b - FIFO overflow occurred during memory read
9 BUF2_ACTIVE	Current frame being stored in Buffer 2 Address Read only bit that indicates Buffer 2 Address is being used to store current frame. 0b - Buffer 2 Address inactive 1b - Buffer 2 Address in use
8 BUF1_ACTIVE	Current frame being stored in Buffer 1 Address Read only bit that indicates Buffer 1 Address is being used to store current frame. 0b - Buffer 1 Address inactive

Table continues on the next page...

Field	Function
	1b - Buffer 1 Address in use
7-0 —	Reserved

13.4.5.1.8 Channel Scale Factor Register (CHNL_SCALE_FACTOR)

13.4.5.1.8.1 Offset

Register	Offset
CHNL_SCALE_FACTOR	18h

13.4.5.1.8.2 Function

This register provides the scale factor for the Bilinear Filter in the Scaling Engine. The maximum down scaling factor is 1/2 such that the output image in either axis is 1/2 the size of the source. The scale down factor can range from 1.0 to 2.0. Decimation and the bilinear filter should be used together to achieve scaling by more than a factor of 2.

To reduce the image by a factor of two in the output frame buffer, a value of 10.0000_0000_0000 should be loaded into this register.

13.4.5.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		Y_SCALE													
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		X_SCALE													
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.8.4 Fields

Field	Function
31-30	Reserved field. Reads only zeros

Table continues on the next page...

Field	Function
RSVD0	
29-16 Y_SCALE	Vertical scaling factor This is a two bit integer and 12 bit fractional representation (##.####_####_####) of the Y-dimension scaling factor for the bilinear filter. The maximum value programmed should be 2 since scaling down by a factor greater than 2 is not supported with the bilinear filter.
15-14 RSVD1	Reserved field. Reads only zeros
13-0 X_SCALE	Horizontal scaling factor This is a two bit integer and 12 bit fractional representation (##.####_####_####) of the X-dimension scaling factor for the bilinear filter. The maximum value programmed should be 2 since scaling down by a factor greater than 2 is not supported with the bilinear filter.

13.4.5.1.9 Channel Scale Offset Register (CHNL_SCALE_OFFSET)

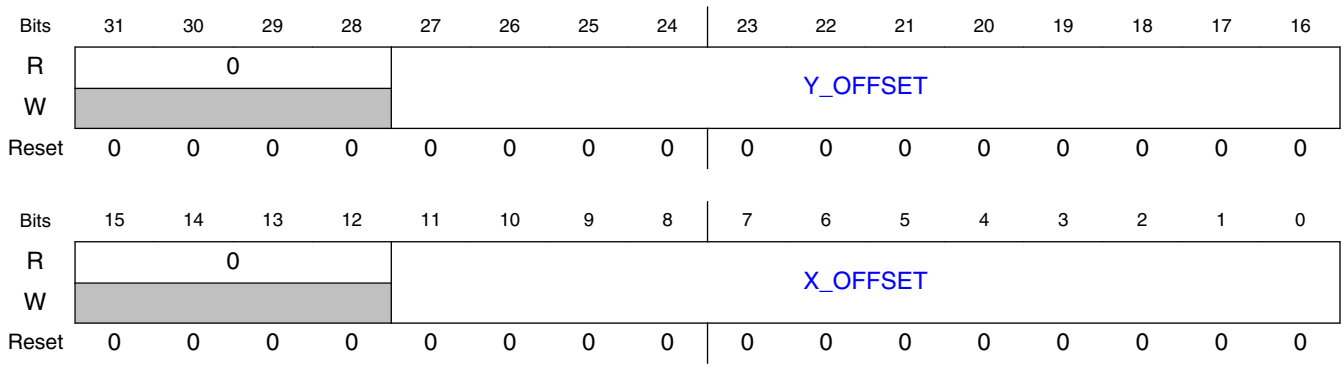
13.4.5.1.9.1 Offset

Register	Offset
CHNL_SCALE_OFFSET	1Ch

13.4.5.1.9.2 Function

This register provides the initial scale offset for the Bilinear Filter. The X and Y offset provides the ability to access the source image with a per sub-pixel granularity. This provides the capability to use all source pixels to effect the output image. The fixed offset values can be used for sub-pixel adjustments in the bilinear scaling filter. For example, when scaling an image down by a factor of 2, an initial offset of 0x0 would result in sub-sampling every other pixel. If a fixed offset of 0x800 (1/2), all pixels are used in scaling the final output pixel value. In this case, the first output pixel would be the sum of (1/2*P0) + (1/2*P1). This fixed offset is applied after the decimation filter stage, and before the bilinear filter stage.

13.4.5.1.9.3 Diagram



13.4.5.1.9.4 Fields

Field	Function
31-28 RSVD0	Reserved field. Reads only zeros
27-16 Y_OFFSET	Vertical scaling offset This is a 12 bit fractional representation (0.####_####_####) of the Y-dimension scaling offset. This represents a fixed pixel offset which gets added to the scaled address to determine source data for the scaling engine.
15-12 RSVD1	Reserved field. Reads only zeros
11-0 X_OFFSET	Horizontal scaling offset This is a 12 bit fractional representation (0.####_####_####) of the X-dimension scaling offset. This represents a fixed pixel offset which gets added to the scaled address to determine source data for the scaling engine.

13.4.5.1.10 Channel Crop Upper Left Corner Coordinate Register (CHNL_CROP_ULC)

13.4.5.1.10.1 Offset

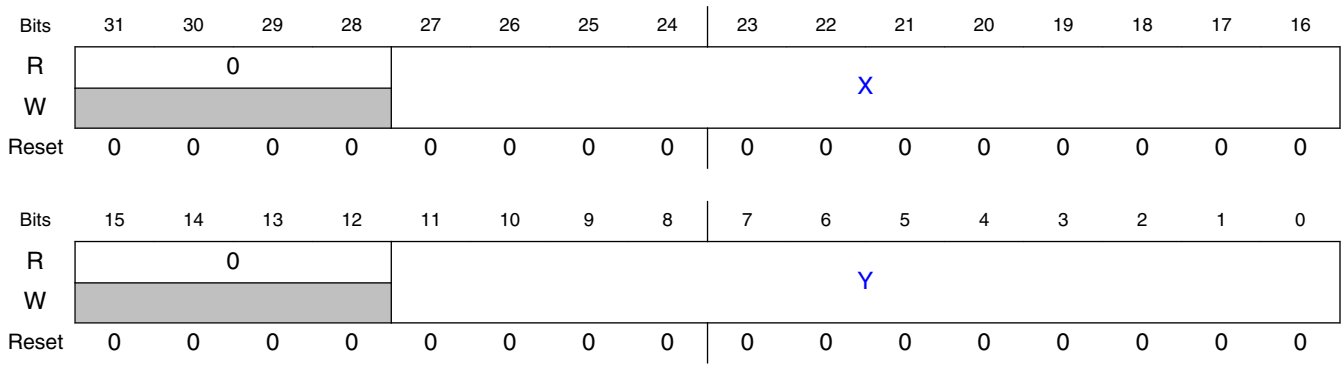
Register	Offset
CHNL_CROP_ULC	20h

13.4.5.1.10.2 Function

This register provides the upper left coordinate of the source image which has to be cropped. The ULC should be set as per the scaled image resolution (if scaler is enabled)

and as per the input image resolution (if scaler is bypassed). The output image after cropping will comprise of pixels of the source image that are within the region specified by the upper left coordinate (ULC) and lower right coordinate (LRC) registers. Cropping operation will be done when CROP_EN bit is set to '1' in the IMG_CTRL register.

13.4.5.1.10.3 Diagram



13.4.5.1.10.4 Fields

Field	Function
31-28 RSVD0	Reserved field. Reads only zeros
27-16 X	Upper Left X-coordinate This field indicates the upper left X-coordinate (in pixels) of the scaled image. The range of valid values includes 0 to WIDTH - 1.
15-12 RSVD1	Reserved field. Reads only zeros
11-0 Y	Upper Left Y-coordinate This field indicates the upper left Y-coordinate (in pixels) of the scaled image. The range of valid values includes 0 to HEIGHT - 1.

13.4.5.1.11 Channel Crop Lower Right Corner Coordinate Register (CHNL_CROP_LRC)

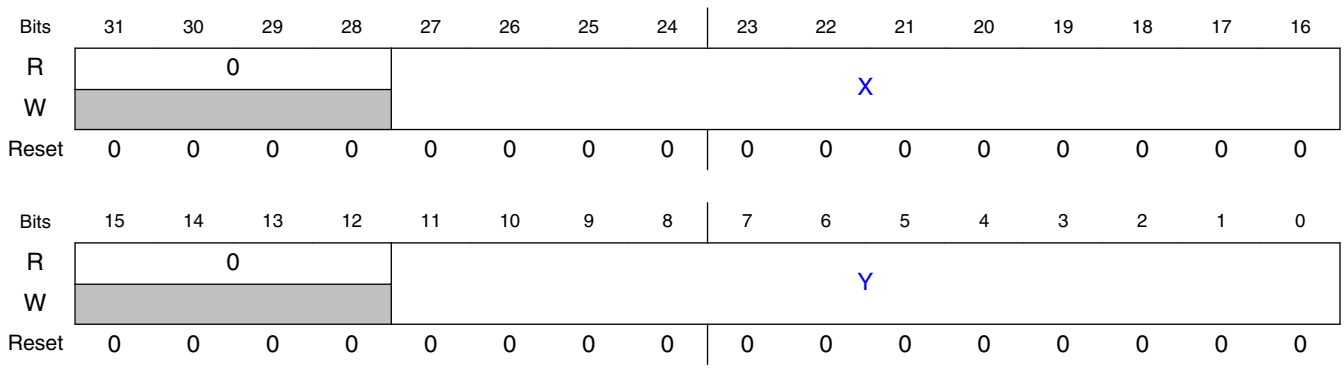
13.4.5.1.11.1 Offset

Register	Offset
CHNL_CROP_LRC	24h

13.4.5.1.11.2 Function

This register provides the lower right coordinate of the source image which has to be cropped. The LRC should be set as per the scaled image resolution (if scaler is enabled) and as per the input image resolution (if scaler is bypassed). The output image will comprise of pixels of the source image that are within the region specified by the upper left coordinate (ULC) and lower right coordinate (LRC) registers. Cropping operation will be done when CROP_EN bit is set to '1' in the IMG_CTRL register.

13.4.5.1.11.3 Diagram



13.4.5.1.11.4 Fields

Field	Function
31-28 RSVD0	Reserved field. Reads only zeros
27-16 X	Lower Right X-coordinate This field indicates the lower right X-coordinate (in pixels) of the source image. The range of valid values includes 0 to WIDTH - 1.
15-12 RSVD1	Reserved field. Reads only zeros
11-0 Y	Lower Right Y-coordinate This field indicates the lower right Y-coordinate (in pixels) of the srouce image. The range of valid values includes 0 to HEIGHT - 1.

13.4.5.1.12 Channel Color Space Conversion Coefficient Register 0 (CHNL_CSC_COEFF0)

13.4.5.1.12.1 Offset

Register	Offset
CHNL_CSC_COEFF0	28h

13.4.5.1.12.2 Function

This register contains color space conversion coefficients in two's complement notation.

13.4.5.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					A2										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					A1										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.12.4 Fields

Field	Function
31-27 RSVD0	Reserved Field. Reads only zeros
26-16 A2	CSC Coefficient A2 value Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 RSVD1	Reserved Field. Reads only zeros
10-0 A1	CSC Coefficient A1 value Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

13.4.5.1.13 Channel Color Space Conversion Coefficient Register 1 (CHNL_CSC_COEFF1)

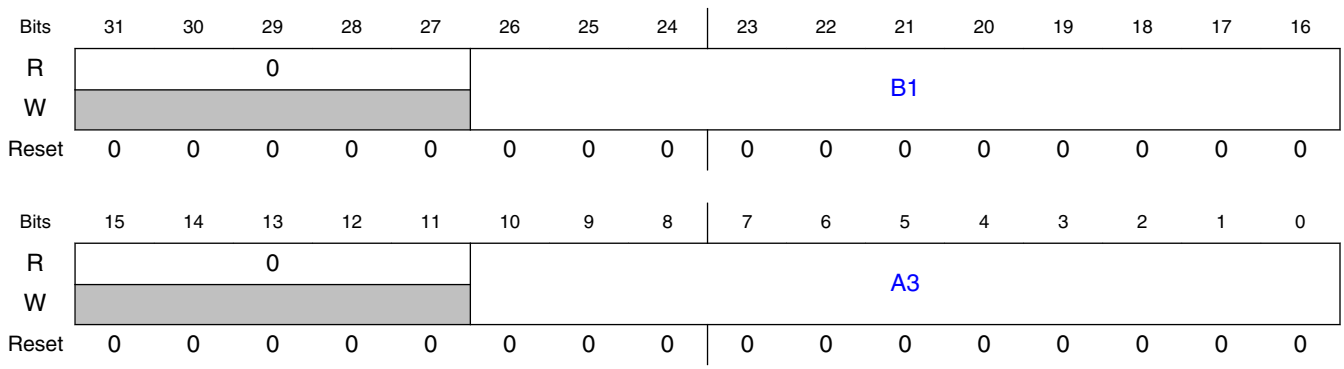
13.4.5.1.13.1 Offset

Register	Offset
CHNL_CSC_COEFF1	2Ch

13.4.5.1.13.2 Function

This register contains color space conversion coefficients in two's complement notation.

13.4.5.1.13.3 Diagram



13.4.5.1.13.4 Fields

Field	Function
31-27 RSVD0	Reserved Field. Reads only zeros
26-16 B1	CSC Coefficient B1 value Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 RSVD1	Reserved Field. Reads only zeros
10-0 A3	CSC Coefficient A3 value Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

13.4.5.1.14 Channel Color Space Conversion Coefficient Register 2 (CHNL_CSC_COEFF2)

13.4.5.1.14.1 Offset

Register	Offset
CHNL_CSC_COEFF2	30h

13.4.5.1.14.2 Function

This register contains color space conversion coefficients in two's complement notation.

13.4.5.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					B3										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					B2										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.14.4 Fields

Field	Function
31-27 RSVD0	Reserved Field. Reads only zeros
26-16 B3	CSC Coefficient B3 value Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 RSVD1	Reserved Field. Reads only zeros
10-0 B2	CSC Coefficient B2 value Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

13.4.5.1.15 Channel Color Space Conversion Coefficient Register 3 (CHNL_CSC_COEFF3)

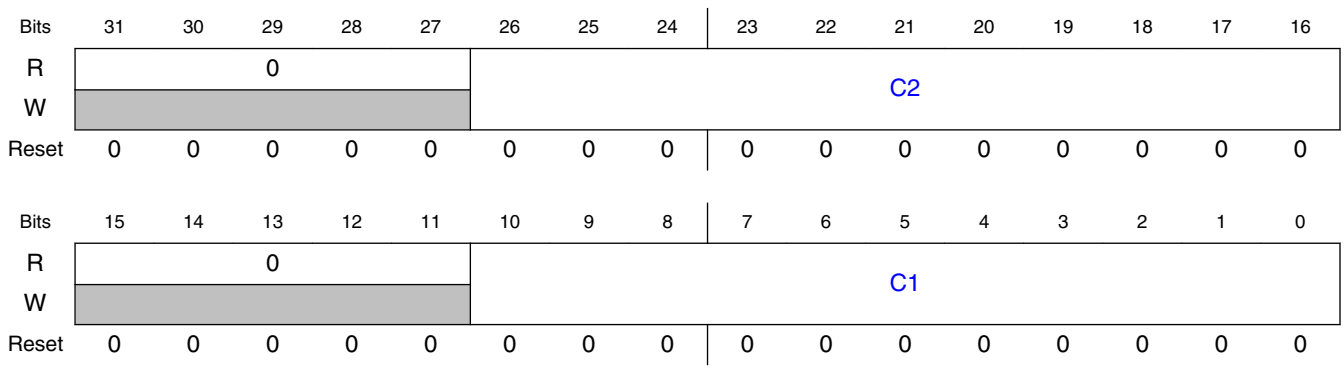
13.4.5.1.15.1 Offset

Register	Offset
CHNL_CSC_COEFF3	34h

13.4.5.1.15.2 Function

This register contains color space conversion coefficients in two's complement notation.

13.4.5.1.15.3 Diagram



13.4.5.1.15.4 Fields

Field	Function
31-27 RSVD0	Reserved Field. Reads only zeros
26-16 C2	CSC Coefficient C2 value Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.
15-11 RSVD1	Reserved Field. Reads only zeros
10-0 C1	CSC Coefficient C1 value Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

13.4.5.1.16 Channel Color Space Conversion Coefficient Register 4 (CHNL_CSC_COEFF4)

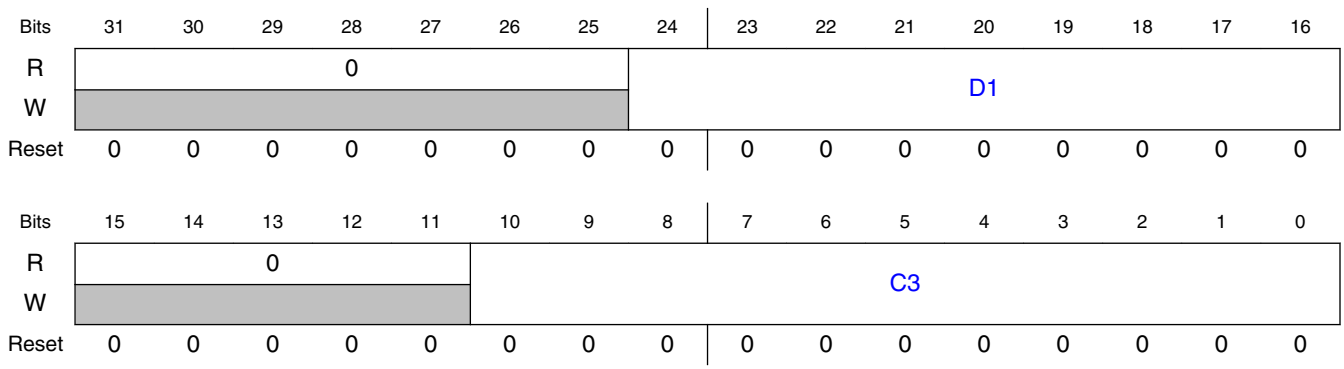
13.4.5.1.16.1 Offset

Register	Offset
CHNL_CSC_COEFF4	38h

13.4.5.1.16.2 Function

This register contains color space conversion coefficients in two's complement notation.

13.4.5.1.16.3 Diagram



13.4.5.1.16.4 Fields

Field	Function
31-25 RSVD0	Reserved Field. Reads only zeros
24-16 D1	CSC Coefficient D1 value Two's complement coefficient integer offset to be added.
15-11 RSVD1	Reserved Field. Reads only zeros
10-0 C3	CSC Coefficient C3 value Two's complement coefficient offset. This coefficient has a sign bit, 2 bits integer, and 8 bits of fraction as ###.####_####.

13.4.5.1.17 Channel Color Space Conversion Coefficient Register 5 (CHNL_CSC_COEFF5)

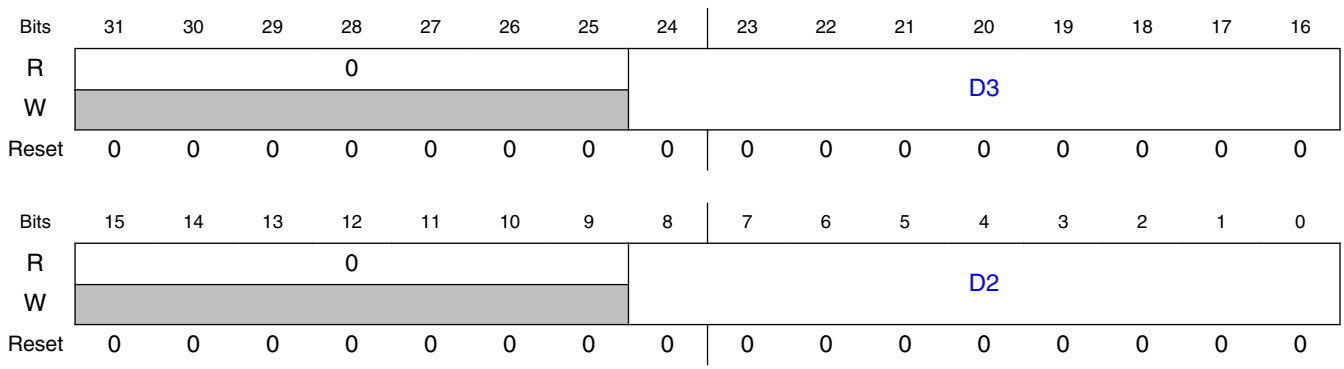
13.4.5.1.17.1 Offset

Register	Offset
CHNL_CSC_COEFF5	3Ch

13.4.5.1.17.2 Function

This register contains color space conversion coefficients in two's complement notation.

13.4.5.1.17.3 Diagram



13.4.5.1.17.4 Fields

Field	Function
31-25 RSVD0	Reserved Field. Reads only zeros
24-16 D3	CSC Coefficient D3 value Two's complement coefficient integer offset to be added.
15-9 RSVD1	Reserved Field. Reads only zeros
8-0 D2	CSC Coefficient D2 value Two's complement coefficient integer offset to be added.

13.4.5.1.18 Channel Alpha Value Register for Region of Interest 0 (CHNL_ROI_0_ALPHA)

13.4.5.1.18.1 Offset

Register	Offset
CHNL_ROI_0_ALPHA	40h

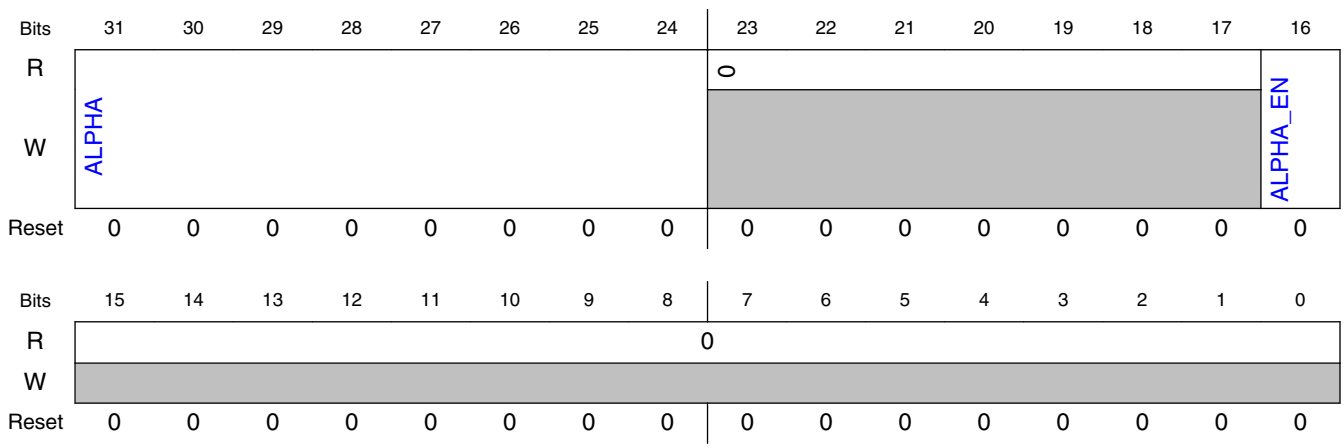
13.4.5.1.18.2 Function

This register provides the alpha value and control to enable alpha value insertion into region of interest 0 of an image. The region is defined by the ROI_0_ULC and ROI_0_LRC registers. The alpha insertion happens after scaling is done, so the ULC and LRC should correspond to the scaled image dimensions.

NOTE

If one value of alpha is to be inserted globally for the whole image then use the IMG_CTRL register to provide the global alpha value and enable control. The ROI_* register will not be used.

13.4.5.1.18.3 Diagram



13.4.5.1.18.4 Fields

Field	Function
31-24 ALPHA	Alpha Value to be inserted with image Alpha value to be inserted into a region of interest of an image if ALPHA_EN is set to '1'. The region of interest is defined by the ROI_0_ULC and ROI_0_LRC regsiters.
23-17 RSVD0	Reserved field. Reads only zeros

Table continues on the next page...

Field	Function
16 ALPHA_EN	Alpha value insertion enable Controls whether alpha value is inserted or not into a region of interest of an image. The region of interest is defined by the ROI_0_ULC and ROI_0_LRC regsiters. 0b - Alpha value insertion is disabled 1b - Alpha value insertion is enabled
15-0 RSVD1	Reserved field. Reads only zeros

13.4.5.1.19 Channel Upper Left Coordinate Register for Region of Interest 0 (CHNL_ROI_0_ULC)

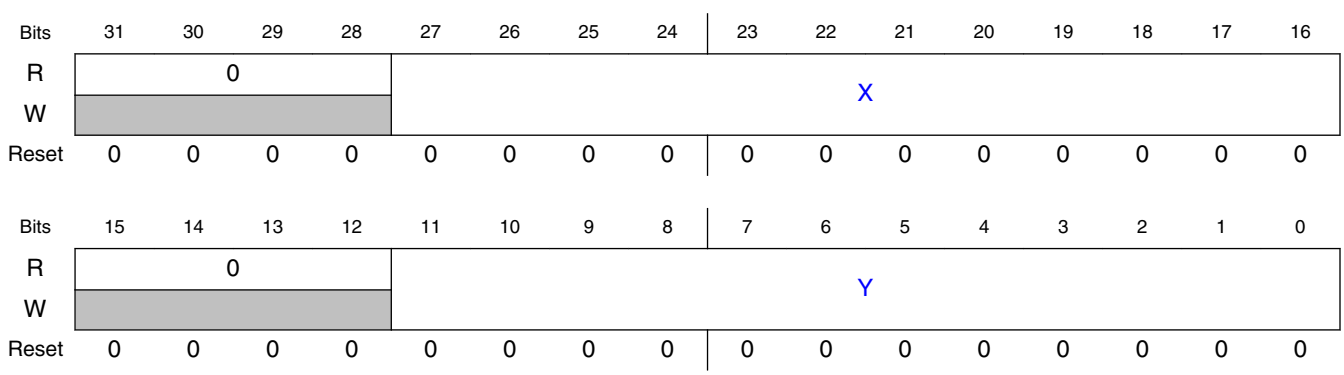
13.4.5.1.19.1 Offset

Register	Offset
CHNL_ROI_0_ULC	44h

13.4.5.1.19.2 Function

Defines the upper left coordinate for the region of interest 0. The alpha insertion happens after scaling is done, so the ULC and LRC should correspond to the scaled image dimensions. Pixel locations that are greater than or equal to the upper left coordinates and less than or equal to the lower right coordinates define the region of interest.

13.4.5.1.19.3 Diagram



13.4.5.1.19.4 Fields

Field	Function
31-28 RSVD0	Reserved field. Reads only zeros
27-16 X	Upper Left X-coordinate This field indicates the upper left X-coordinate (in pixels) of the region of interest 0.
15-12 RSVD1	Reserved field. Reads only zeros
11-0 Y	Upper Left Y-coordinate This field indicates the upper left Y-coordinate (in pixels) of the region of interest 0.

13.4.5.1.20 Channel Lower Right Coordinate Register for Region of Interest 0 (CHNL_ROI_0_LRC)

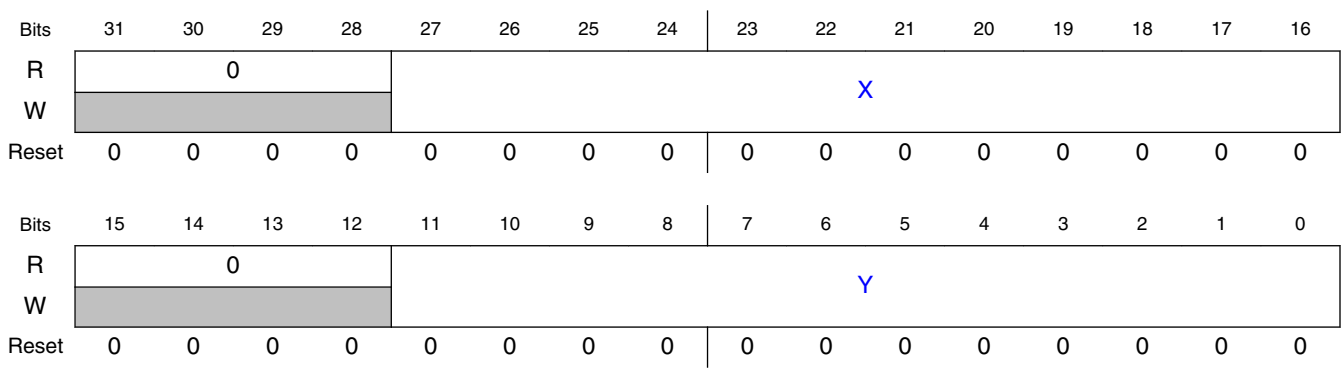
13.4.5.1.20.1 Offset

Register	Offset
CHNL_ROI_0_LRC	48h

13.4.5.1.20.2 Function

Defines the lower right coordinate for the region of interest 0. The alpha insertion happens after scaling is done, so the ULC and LRC should correspond to the scaled image dimensions. Pixel locations that are greater than or equal to the upper left coordinates and less than or equal to the lower right coordinates define the region of interest.

13.4.5.1.20.3 Diagram



13.4.5.1.20.4 Fields

Field	Function
31-28 RSVD0	Reserved field. Reads only zeros
27-16 X	Lower Right X-coordinate This field indicates the lower right X-coordinate (in pixels) of the region of interest 0.
15-12 RSVD1	Reserved field. Reads only zeros
11-0 Y	Lower Right Y-coordinate This field indicates the lower right Y-coordinate (in pixels) of the region of interest 0.

13.4.5.1.21 Channel Alpha Value Register for Region of Interest 1 (CHNL_ROI_1_ALPHA)

13.4.5.1.21.1 Offset

Register	Offset
CHNL_ROI_1_ALPHA	4Ch

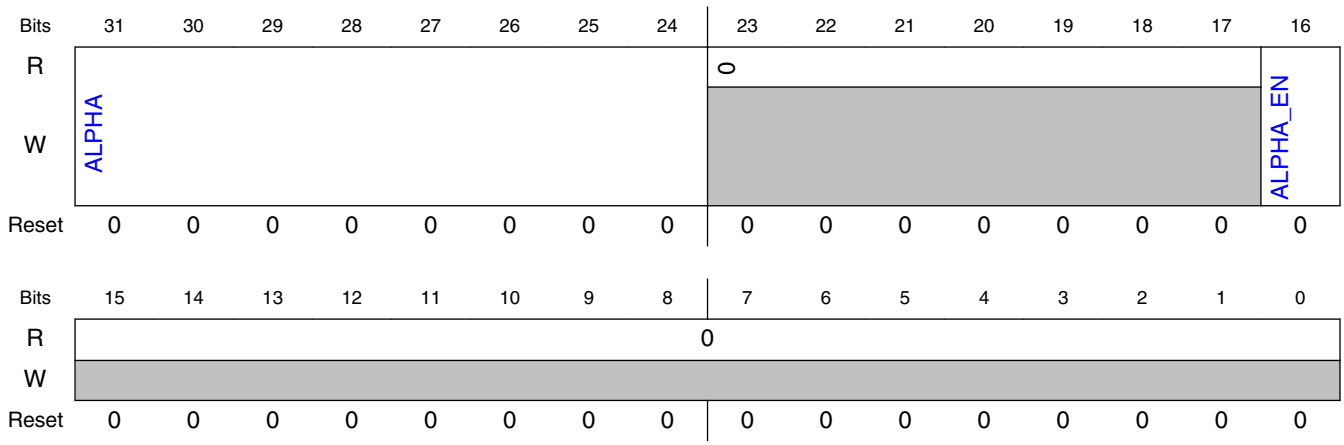
13.4.5.1.21.2 Function

This register provides the alpha value and control to enable alpha value insertion into region of interest 1 of an image. The region is defined by the ROI_1_ULC and ROI_1_LRC registers. The alpha insertion happens after scaling is done, so the ULC and LRC should correspond to the scaled image dimensions.

NOTE

If one value of alpha is to be inserted globally for the whole image then use the IMG_CTRL register to provide the global alpha value and enable control. The ROI_* register will not be used.

13.4.5.1.21.3 Diagram



13.4.5.1.21.4 Fields

Field	Function
31-24 ALPHA	Alpha Value to be inserted with image Alpha value to be inserted into a region of interest of an image if ALPHA_EN is set to '1'. The region of interest is defined by the ROI_1_ULC and ROI_1_LRC regsiters.
23-17 RSVD0	Reserved field. Reads only zeros
16 ALPHA_EN	Alpha value insertion enable Controls whether alpha value is inserted or not into a region of interest of an image. The region of interest is defined by the ROI_1_ULC and ROI_1_LRC regsiters. 0b - Alpha value insertion is disabled 1b - Alpha value insertion is enabled
15-0 RSVD1	Reserved field. Reads only zeros

13.4.5.1.22 Channel Upper Left Coordinate Register for Region of Interest 1 (CHNL_ROI_1_ULC)

13.4.5.1.22.1 Offset

Register	Offset
CHNL_ROI_1_ULC	50h

13.4.5.1.22.2 Function

Defines the upper left coordinate for the region of interest 1. The alpha insertion happens after scaling is done, so the ULC and LRC should correspond to the scaled image dimensions. Pixel locations that are greater than or equal to the upper left coordinates and less than or equal to the lower right coordinates define the region of interest.

13.4.5.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				X											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				Y											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.22.4 Fields

Field	Function
31-28 RSVD0	Reserved field. Reads only zeros
27-16 X	Upper Left X-coordinate This field indicates the upper left X-coordinate (in pixels) of the region of interest 1.
15-12 RSVD1	Reserved field. Reads only zeros
11-0 Y	Upper Left Y-coordinate This field indicates the upper left Y-coordinate (in pixels) of the region of interest 1.

13.4.5.1.23 Channel Lower Right Coordinate Register for Region of Interest 1 (CHNL_ROI_1_LRC)

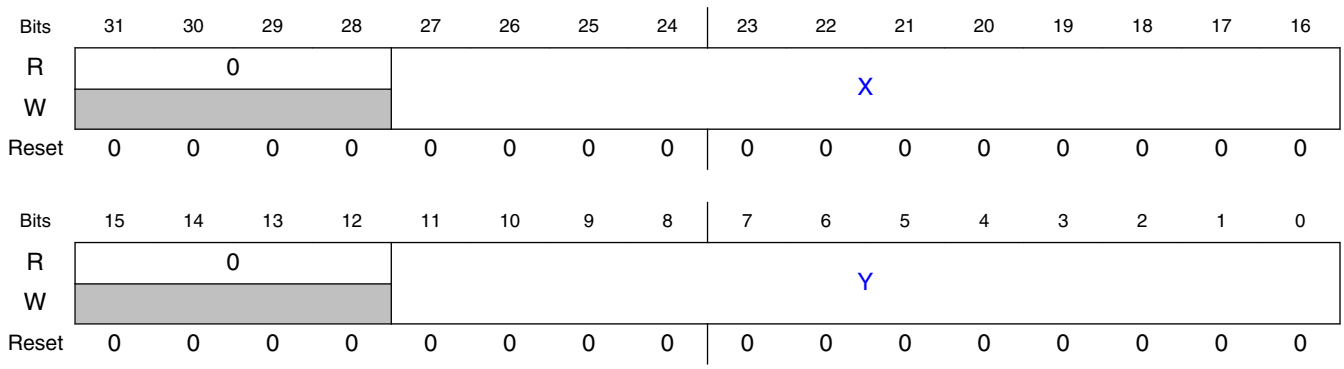
13.4.5.1.23.1 Offset

Register	Offset
CHNL_ROI_1_LRC	54h

13.4.5.1.23.2 Function

Defines the lower right coordinate for the region of interest 1. The alpha insertion happens after scaling is done, so the ULC and LRC should correspond to the scaled image dimensions. Pixel locations that are greater than or equal to the upper left coordinates and less than or equal to the lower right coordinates define the region of interest.

13.4.5.1.23.3 Diagram



13.4.5.1.23.4 Fields

Field	Function
31-28 RSVD0	Reserved field. Reads only zeros
27-16 X	Lower Right X-coordinate This field indicates the lower right X-coordinate (in pixels) of the region of interest 1.
15-12 RSVD1	Reserved field. Reads only zeros
11-0 Y	Lower Right Y-coordinate This field indicates the lower right Y-coordinate (in pixels) of the region of interest 1.

13.4.5.1.24 Channel Alpha Value Register for Region of Interest 2 (CHNL_ROI_2_ALPHA)

13.4.5.1.24.1 Offset

Register	Offset
CHNL_ROI_2_ALPHA	58h

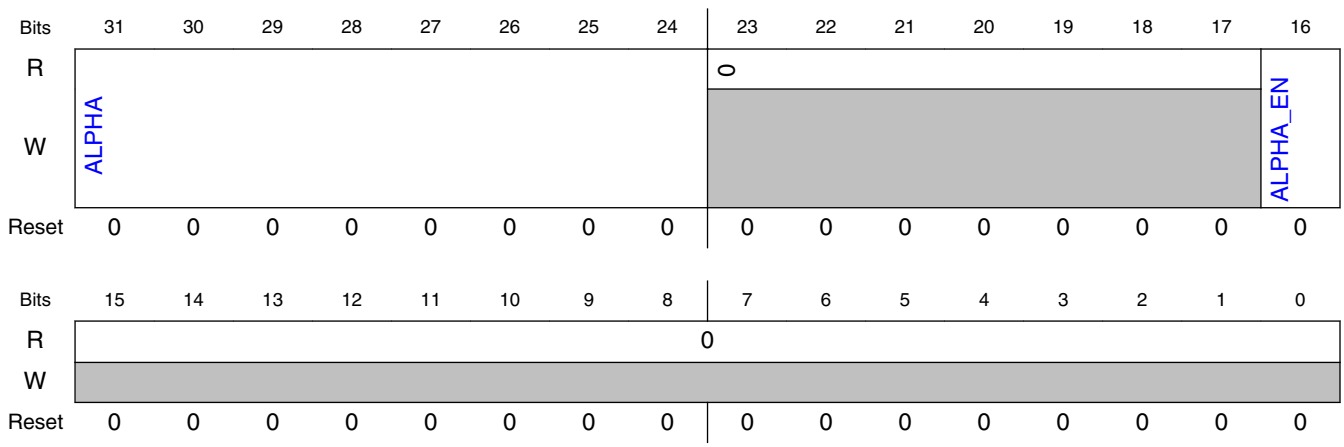
13.4.5.1.24.2 Function

This register provides the alpha value and control to enable alpha value insertion into region of interest 2 of an image. The region is defined by the ROI_2_ULC and ROI_2_LRC registers. The alpha insertion happens after scaling is done, so the ULC and LRC should correspond to the scaled image dimensions.

NOTE

If one value of alpha is to be inserted globally for the whole image then use the IMG_CTRL register to provide the global alpha value and enable control. The ROI_* register will not be used.

13.4.5.1.24.3 Diagram



13.4.5.1.24.4 Fields

Field	Function
31-24 ALPHA	Alpha Value to be inserted with image Alpha value to be inserted into a region of interest of an image if ALPHA_EN is set to '1'. The region of interest is defined by the ROI_2_ULC and ROI_2_LRC regsiters.
23-17 RSVD0	Reserved field. Reads only zeros
16 ALPHA_EN	Alpha value insertion enable Controls whether alpha value is inserted or not into a region of interest of an image. The region of interest is defined by the ROI_2_ULC and ROI_2_LRC regsiters. 0b - Alpha value insertion is disabled 1b - Alpha value insertion is enabled
15-0 RSVD1	Reserved field. Reads only zeros

13.4.5.1.25 Channel Upper Left Coordinate Register for Region of Interest 2 (CHNL_ROI_2_ULC)

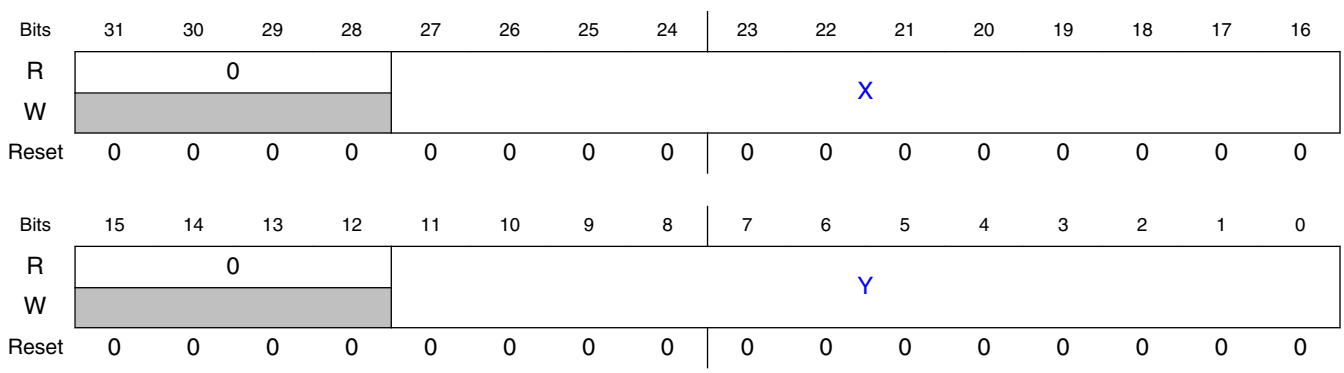
13.4.5.1.25.1 Offset

Register	Offset
CHNL_ROI_2_ULC	5Ch

13.4.5.1.25.2 Function

Defines the upper left coordinate for the region of interest 2. The alpha insertion happens after scaling is done, so the ULC and LRC should correspond to the scaled image dimensions. Pixel locations that are greater than or equal to the upper left coordinates and less than or equal to the lower right coordinates define the region of interest.

13.4.5.1.25.3 Diagram



13.4.5.1.25.4 Fields

Field	Function
31-28 RSVD0	Reserved field. Reads only zeros
27-16 X	Upper Left X-coordinate This field indicates the upper left X-coordinate (in pixels) of the region of interest 2.
15-12 RSVD1	Reserved field. Reads only zeros
11-0 Y	Upper Left Y-coordinate This field indicates the upper left Y-coordinate (in pixels) of the region of interest 2.

13.4.5.1.26 Channel Lower Right Coordinate Register for Region of Interest 2 (CHNL_ROI_2_LRC)

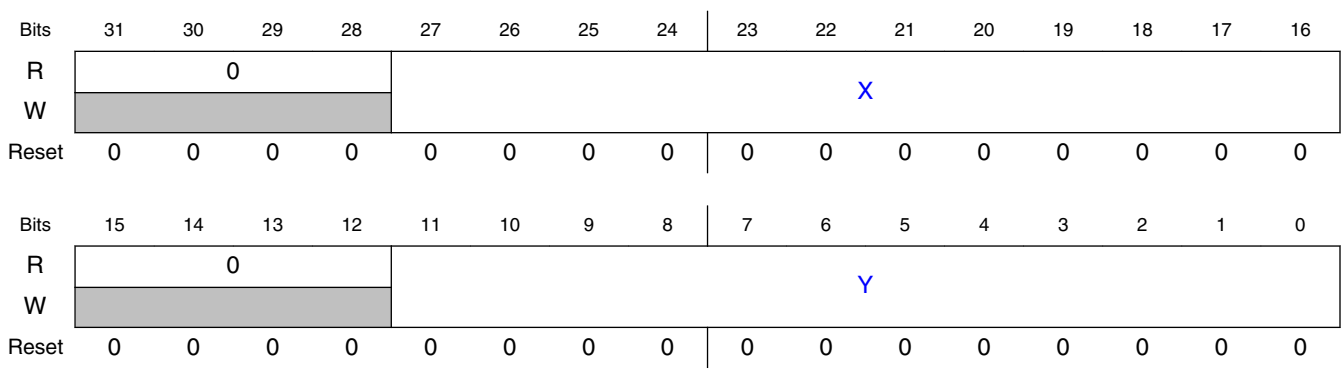
13.4.5.1.26.1 Offset

Register	Offset
CHNL_ROI_2_LRC	60h

13.4.5.1.26.2 Function

Defines the lower right coordinate for the region of interest 2. The alpha insertion happens after scaling is done, so the ULC and LRC should correspond to the scaled image dimensions. Pixel locations that are greater than or equal to the upper left coordinates and less than or equal to the lower right coordinates define the region of interest.

13.4.5.1.26.3 Diagram



13.4.5.1.26.4 Fields

Field	Function
31-28 RSVD0	Reserved field. Reads only zeros
27-16 X	Lower Right X-coordinate This field indicates the lower right X-coordinate (in pixels) of the region of interest 2.
15-12 RSVD1	Reserved field. Reads only zeros
11-0	Lower Right Y-coordinate

Field	Function
Y	This field indicates the lower right Y-coordinate (in pixels) of the region of interest 2.

13.4.5.1.27 Channel Alpha Value Register for Region of Interest 3 (CHNL_ROI_3_ALPHA)

13.4.5.1.27.1 Offset

Register	Offset
CHNL_ROI_3_ALPHA	64h

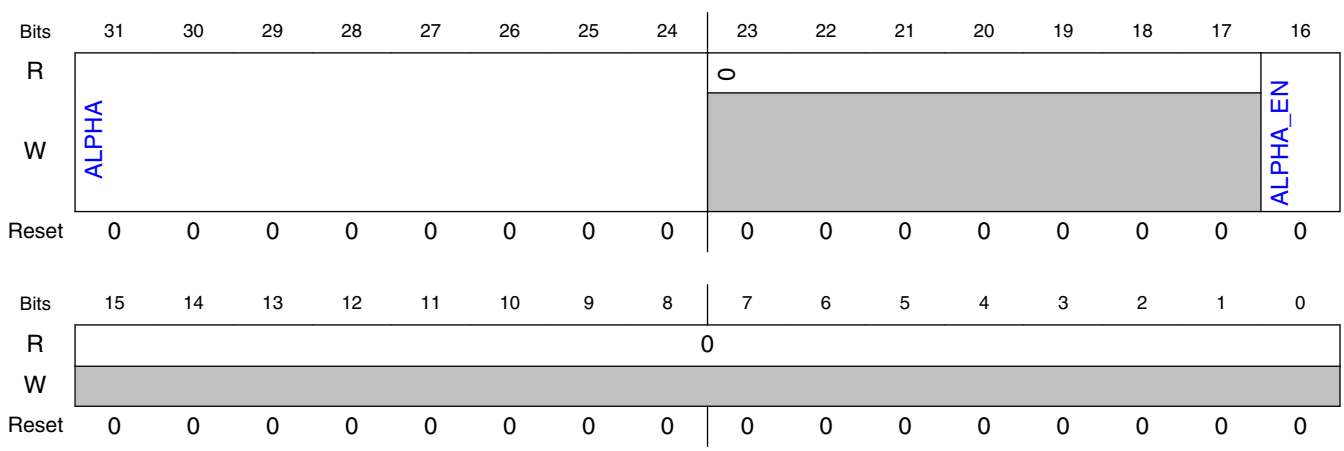
13.4.5.1.27.2 Function

This register provides the alpha value and control to enable alpha value insertion into region of interest 3 of an image. The region is defined by the ROI_3_ULC and ROI_3_LRC registers. The alpha insertion happens after scaling is done, so the ULC and LRC should correspond to the scaled image dimensions.

NOTE

If one value of alpha is to be inserted globally for the whole image then use the IMG_CTRL register to provide the global alpha value and enable control. The ROI_* register will not be used.

13.4.5.1.27.3 Diagram



13.4.5.1.27.4 Fields

Field	Function
31-24 ALPHA	Alpha Value to be inserted with image Alpha value to be inserted into a region of interest of an image if ALPHA_EN is set to '1'. The region of interest is defined by the ROI_3_ULC and ROI_3_LRC regsiters.
23-17 RSVD0	Reserved field. Reads only zeros
16 ALPHA_EN	Alpha value insertion enable Controls whether alpha value is inserted or not into a region of interest of an image. The region of interest is defined by the ROI_3_ULC and ROI_3_LRC regsiters. 0b - Alpha value insertion is disabled 1b - Alpha value insertion is enabled
15-0 RSVD1	Reserved field. Reads only zeros

13.4.5.1.28 Channel Upper Left Coordinate Register for Region of Interest 3 (CHNL_ROI_3_ULC)

13.4.5.1.28.1 Offset

Register	Offset
CHNL_ROI_3_ULC	68h

13.4.5.1.28.2 Function

Defines the upper left coordinate for the region of interest 3. The alpha insertion happens after scaling is done, so the ULC and LRC should correspond to the scaled image dimensions. Pixel locations that are greater than or equal to the upper left coordinates and less than or equal to the lower right coordinates define the region of interest.

13.4.5.1.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				X											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				Y											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.28.4 Fields

Field	Function
31-28 RSVD0	Reserved field. Reads only zeros
27-16 X	Upper Left X-coordinate This field indicates the upper left X-coordinate (in pixels) of the region of interest 3.
15-12 RSVD1	Reserved field. Reads only zeros
11-0 Y	Upper Left Y-coordinate This field indicates the upper left Y-coordinate (in pixels) of the region of interest 3.

13.4.5.1.29 Channel Lower Right Coordinate Register for Region of Interest 3 (CHNL_ROI_3_LRC)

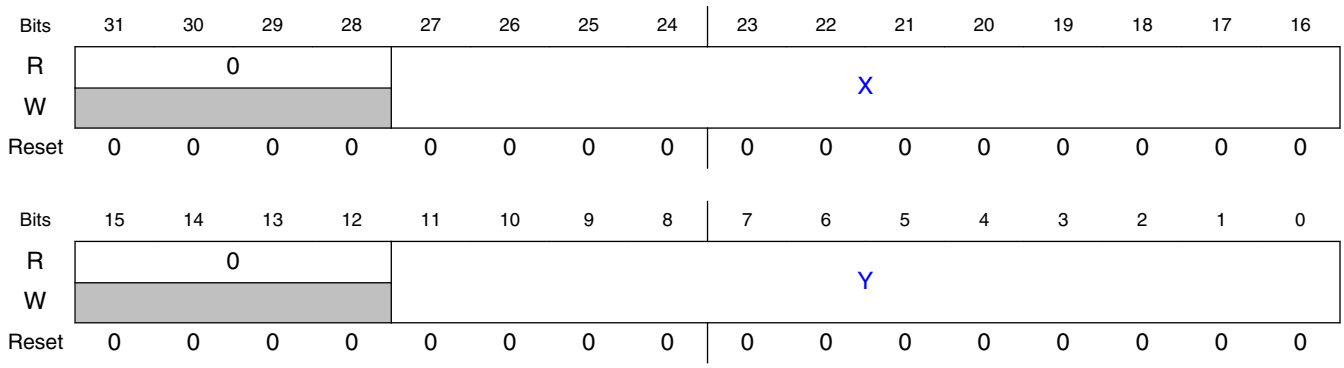
13.4.5.1.29.1 Offset

Register	Offset
CHNL_ROI_3_LRC	6Ch

13.4.5.1.29.2 Function

Defines the lower right coordinate for the region of interest 3. The alpha insertion happens after scaling is done, so the ULC and LRC should correspond to the scaled image dimensions. Pixel locations that are greater than or equal to the upper left coordinates and less than or equal to the lower right coordinates define the region of interest.

13.4.5.1.29.3 Diagram



13.4.5.1.29.4 Fields

Field	Function
31-28 RSVD0	Reserved field. Reads only zeros
27-16 X	Lower Right X-coordinate This field indicates the lower right X-coordinate (in pixels) of the region of interest 3.
15-12 RSVD1	Reserved field. Reads only zeros
11-0 Y	Lower Right Y-coordinate This field indicates the lower right Y-coordinate (in pixels) of the region of interest 3.

13.4.5.1.30 Channel RGB or Luma (Y) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_Y)

13.4.5.1.30.1 Offset

Register	Offset
CHNL_OUT_BUF1_ADDR_Y	70h

13.4.5.1.30.2 Function

Starting memory address of Buffer 1 where the RGB or Y (Luma) data of the processed image is stored.

13.4.5.1.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.30.4 Fields

Field	Function
31-0 ADDR	Starting address for the RGB or Y (luma) memory location

13.4.5.1.31 Channel Chroma (U/Cb/UV/CbCr) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_U)

13.4.5.1.31.1 Offset

Register	Offset
CHNL_OUT_BUF1_ADDR_U	74h

13.4.5.1.31.2 Function

Starting memory address of Buffer 1 where the U/Cb data of the processed image is stored. In two plane operation, this register points to the starting memory address where the UV/CbCr chroma of the processed image is stored. Not used in one plane operation.

13.4.5.1.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.31.4 Fields

Field	Function
31-0 ADDR	Starting address for the U/Cb or 2 plane UV/CbCr Chroma memory location

13.4.5.1.32 Channel Chroma (V/Cr) Output Buffer 1 Address (CHNL_OUT_BUF1_ADDR_V)

13.4.5.1.32.1 Offset

Register	Offset
CHNL_OUT_BUF1_ADDR_V	78h

13.4.5.1.32.2 Function

Starting memory address of Buffer 1 where the V/Cr data of the processed image is stored. In one or two plane operation, this register is not used.

13.4.5.1.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.32.4 Fields

Field	Function
31-0 ADDR	Starting address for the V/Cr memory location

13.4.5.1.33 Channel Output Buffer Pitch (CHNL_OUT_BUF_PITCH)

13.4.5.1.33.1 Offset

Register	Offset
CHNL_OUT_BUF_PITCH	7Ch

13.4.5.1.33.2 Function

This register contains the output buffer pitch in bytes. Any byte value will indicate the vertical pitch. This value will be used in output pixel address calculations.

13.4.5.1.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LINE_PITCH															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.33.4 Fields

Field	Function
31-16 —	Reserved field. Reads only zeros.
15-0 LINE_PITCH	Output Buffer Line Pitch Indicates the number of bytes in memory between two vertically adjacent pixels.

13.4.5.1.34 Channel Input Buffer Address (CHNL_IN_BUF_ADDR)

13.4.5.1.34.1 Offset

Register	Offset
CHNL_IN_BUF_ADDR	80h

13.4.5.1.34.2 Function

This reigster provides the start address for reading in stand-alone or deinterlace image frames from memory. Blending (or Averaging) deinterlacing stores the odd frame image into memory address specified by this register. The channel with SRC set as Memory reads back the image and sends it to the deinterlacing logic.

The storage location allows for two frames to be stored. When reading in image frames from memory, the module will switch between the two locations automatically. The read operation starts automatically when blend deinterlacing is to be done. When processing stand-alone images from memory, software must trigger the processing of images. The

format of image that can be stored in this location can be RGB, YUV444 or YUV422 only. The image format is set automatically when one channel is configured to perform Blend deintrlacing.

For deinterlacing, the software just needs to configure the deinterlace method and source type in one channel and configure only the source type as memory in another channel. The module can pair the deinterlace and memory channels automatically to perform Blend Deinteralce. For other deinterlace methods, read back of frames from memory is not required. More details on Deinterlacing in Functional Description chapter.

13.4.5.1.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.34.4 Fields

Field	Function
31-0 ADDR	Starting Address from where the input image is read in from. Software can update the IN_BUF_ADDR during direct memory reads. For de-interlacing, it is mandatory for the software to reserve two frame size + pitch range of addresses for the intermediate frames. The hardware logic will determine the addresses for the intermediate frames during deinterlacing as the intermediate frames are of no concern to software.

13.4.5.1.35 Channel Input Buffer Pitch (CHNL_IN_BUF_PITCH)

13.4.5.1.35.1 Offset

Register	Offset
CHNL_IN_BUF_PITCH	84h

13.4.5.1.35.2 Function

This register provides the line and frame pitch for the image to be read out from memory location set in the IN_BUF_ADDR register.

13.4.5.1.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FRM_PITCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LINE_PITCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.35.4 Fields

Field	Function
31-16 FRM_PITCH	Frame Pitch Indicates the number of bytes in memory between two frames being read in from memory. Software can alternatively keep the frame pitch as zero and update the IN_BUF_ADDR during direct memory reads. For de-interlacing, the frame pitch is required to store the intermediate frames. Since the intermediate frames are of no concern to software, it is mandatory for the software to reserve two frame size + pitch range of addresses for the intermediate frames.
15-0 LINE_PITCH	Line Pitch Indicates the number of bytes in memory between two vertically adjacent pixels.

13.4.5.1.36 Channel Memory Read Control (CHNL_MEM_RD_CTRL)

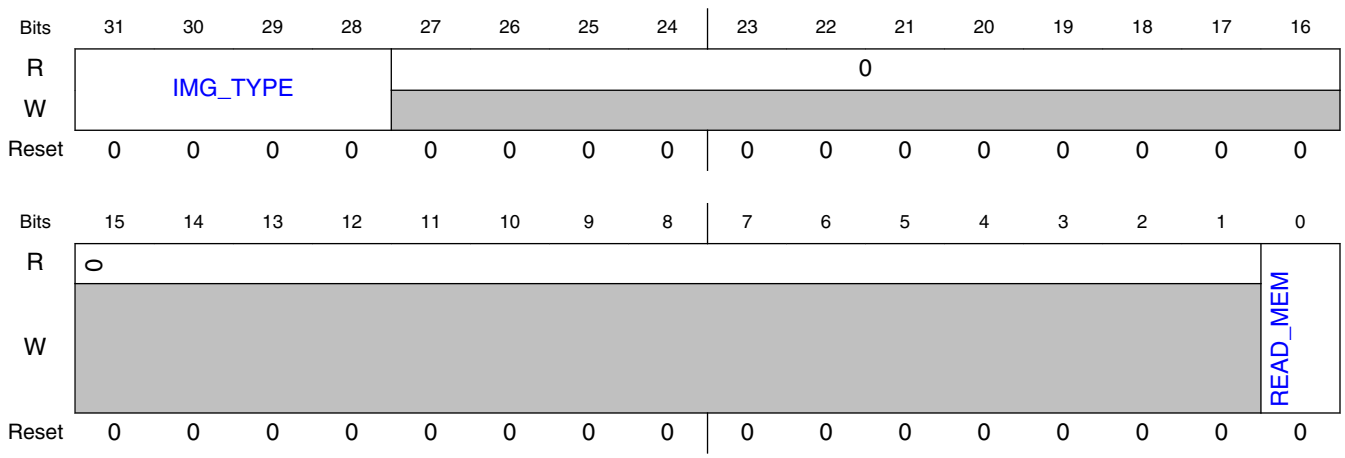
13.4.5.1.36.1 Offset

Register	Offset
CHNL_MEM_RD_CTRL	88h

13.4.5.1.36.2 Function

When the image source type (CTRL[SRC_TYPE])is memory and stand alone image is being read from memory (no deitnerlacing being done), this register also provides the image format information and control to start reading image from memory. This register is not used during deinterlacing where image read back from memory is controlled by deinterlace logic. Both deinterlace image read and standalone image read cannot happen together, even on different channels.

13.4.5.1.36.3 Diagram



13.4.5.1.36.4 Fields

Field	Function
31-28 IMG_TYPE	Input image format When an image is being read from memory (even during deitnerlacing), this field specifies the format of the image. 0000b - BGR8P - BGR format with 8-bits per color component (packed into 32-bit DWORD) 0001b - RGB8P - RGB format with 8-bits per color component (packed into 32-bit DWORD) 0010b - XRGB8 - RGB format with 8-bits per color component (unpacked and LSB aligned in 32-bit DWORD) 0011b - RGBX8 - RGB format with 8-bits per color component (unpacked and MSBalinged in 32-bit DWORD) 0100b - XBGR8 - BGR format with 8-bits per color component (unpacked and LSB aligned in 32-bit DWORD) 0101b - RGB565 - RGB format with 5-bits of R, B; 6-bits of G (packed into 32-bit DWORD) 0110b - A2BGR10 - BGR format with 2-bits alpha in MSB; 10-bits per color component 0111b - A2RGB10 - RGB format with 2-bits alpha in MSB; 10-bits per color component 1000b - YUV444_1P8P with 8-bits per color component; 1-plane, YUV interleaved packed bytes 1001b - YUV444_1P10 with 10-bits per color component; 1-plane, YUV interleaved unpacked bytes (6 LSBs waste bits in 16-bit WORD) 1010b - YUV444_1P10P with 10-bits per color component; 1-plane, YUV interleaved packed bytes (2 MSBs waste bits in 32-bit WORD) 1011b - YUV444_1P12 with 12-bits per color component; 1-plane, YUV interleaved unpacked bytes (4 LSBs waste bits in 16-bit WORD)

Table continues on the next page...

Field	Function
	1100b - YUV444_1P8 with 8-bits per color component; 1-plane YUV interleaved unpacked bytes (8 MSBs waste bits in 32-bit DWORD) 1101b - YUV422_1P8P with 8-bits per color component; 1-plane YUV interleaved packed bytes 1110b - YUV422_1P10 with 10-bits per color component; 1-plane, YUV interleaved unpacked bytes (6 LSBs waste bits in 16-bit WORD) 1111b - YUV422_1P12 with 12-bits per color component; 1-plane, YUV interleaved packed bytes (4 MSBs waste bits in 16-bit WORD)
27-1 RSVD0	Reserved field. Reads only zeros
0 READ_MEM	Initiate read from memory Setting this bit to 1 will initiate direct read of image from memory. One one full image frame will be read. To read another image frame from memory, this bit must be set to '0' and then to '1'. Bit can be set to '0' after setting to '1' while read from memory is in progress; this will not stop the read operation. Read will stop when one full frame is read. This bit is not used during deinterlacing. 0b - No reads from memory done 1b - Reads from memory initiated

13.4.5.1.37 Channel RGB or Luma (Y) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_Y)

13.4.5.1.37.1 Offset

Register	Offset
CHNL_OUT_BUF2_ADDR_Y	8Ch

13.4.5.1.37.2 Function

Starting memory address of Buffer 2 where the RGB or Y (Luma) data of the processed image is stored.

13.4.5.1.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.37.4 Fields

Field	Function
31-0 ADDR	Starting address for the RGB or Y (luma) memory location

13.4.5.1.38 Channel Chroma (U/Cb/UV/CbCr) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_U)

13.4.5.1.38.1 Offset

Register	Offset
CHNL_OUT_BUF2_ADDR_U	90h

13.4.5.1.38.2 Function

Starting memory address of Buffer 2 where the U/Cb data of the processed image is stored. In two plane operation, this register points to the starting memory address where the UV/CbCr chroma of the processed image is stored. Not used in one plane operation.

13.4.5.1.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.38.4 Fields

Field	Function
31-0 ADDR	Starting address for the U/Cb or 2 plane UV/CbCr Chroma memory location

13.4.5.1.39 Channel Chroma (V/Cr) Output Buffer 2 Address (CHNL_OUT_BUF2_ADDR_V)

13.4.5.1.39.1 Offset

Register	Offset
CHNL_OUT_BUF2_ADDR_V	94h

13.4.5.1.39.2 Function

Starting memory address of Buffer 2 where the V/Cr data of the processed image is stored. In one or two plane operation, this register is not used.

13.4.5.1.39.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.39.4 Fields

Field	Function
31-0 ADDR	Starting address for the V/Cr memory location

13.4.5.1.40 Channel Scaled Image Configuration (CHNL_SCL_IMG_CFG)

13.4.5.1.40.1 Offset

Register	Offset
CHNL_SCL_IMG_CFG	98h

13.4.5.1.40.2 Function

This register provides the details about the image width (pixels) and height (lines) at the output of scaler. Since the image after scaling can be flipped and/or cropped, the dimension of image after scaling becomes an important parameter for the output buffer control logic.

The scaled image dimension is computed as Dimension in CHNL_IMG_CFG register divided by the scale factor in CHNL_SCALE_FACTOR and rounded to the closest integer.

NOTE

Even is scaler is bypassed, this register must be programmed with the correct image dimensions.

13.4.5.1.40.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			HEIGHT												
W																
Reset	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WIDTH												
W																
Reset	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0

13.4.5.1.40.4 Fields

Field	Function
31-29 RSVD1	Reserved field. Reads only zeros.
28-16 HEIGHT	Scaled image height (lines) This field provides the height of scaled image in terms of number of lines. Default height is 1080 lines for 1:1 scaling.

Table continues on the next page...

Field	Function
15-13 RSVD0	Reserved field. Reads only zeros.
12-0 WIDTH	Scaled image width (pixels) This field provides the width of the scaled image in terms of number of pixels. Default width is 1920 pixels for 1:1 scaling.

13.4.5.1.41 Channel Flow Control Register (CHNL_FLOW_CTRL)

13.4.5.1.41.1 Offset

Register	Offset
CHNL_FLOW_CTRL	9Ch

13.4.5.1.41.2 Function

This register provides the amount of usable AXI bus bandwidth for writing out the pixel data from the channel's output buffers (Y or U or V). The usable bandwidth is specified as a fraction (i.e. numerator/denominator) where denominator >= numerator. This flow control kicks in for the whole frame when the source of frame is memory or for only the last line when source is a pixel link input and line buffer inside the scaler is being used (i.e. when scaler is enabled or YUV420 data is being input).

13.4.5.1.41.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								FC_NUMER							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								FC_DENOM							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.5.1.41.4 Fields

Field	Function
31-24	Reserved

Table continues on the next page...

Field	Function
—	
23-16 FC_NUMER	Numertor value of fraction of usable bandwidth This value determines the clock cycles out of FC_DENOM number of clock cycles during which pixels can be written out. FC_NUMER cannot be be greater than FC_DENOM. Setting this field to 0 will disable flow control. Setting FC_NUMER = FC_DENOM will allow the AXI write request to consume full bandwidth available. 00000000b - Flow control is disabled.
15-8 —	Reserved
7-0 FC_DENOM	Denominator value of fraction of usable bandwidth This value determines the total cycles within which FC_NUMER clock cycles will be used to write out the data on the AXI write bus. FC_DENOM cannot be less than FC_NUMER. Setting this field to 0 will disable flow control. Setting FC_NUMER = FC_DENOM will allow the AXI write request to consume full bandwidth available. 00000000b - Invalid value. Flow control will be disabled.

13.4.5.1.42 Channel Output Y-Buffer 1 Extended Address Bits Register (CHNL_Y_BUF1_XTND_ADDR)

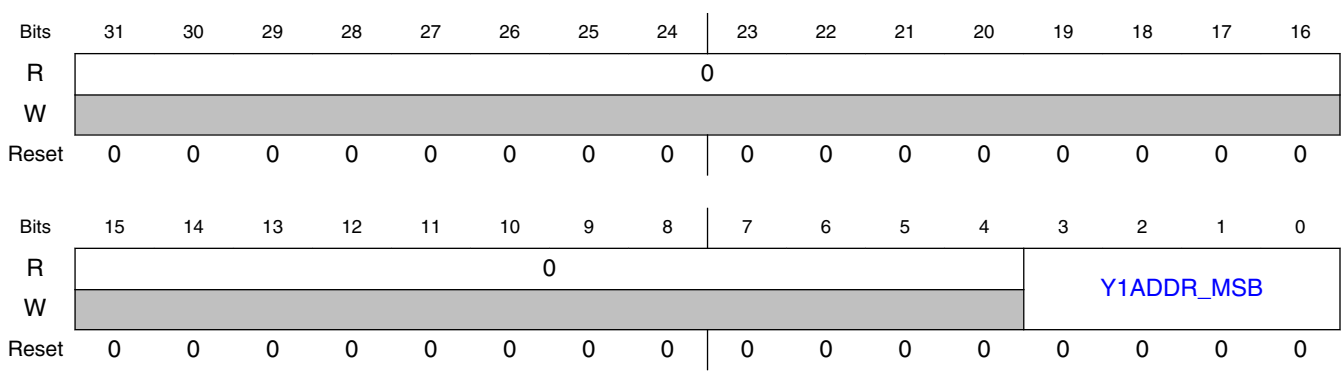
13.4.5.1.42.1 Offset

Register	Offset
CHNL_Y_BUF1_XTND_ADDR	A0h

13.4.5.1.42.2 Function

This register provides additional MSB bits to address more than 32-bits of address space on the AXI bus.

13.4.5.1.42.3 Diagram



13.4.5.1.42.4 Fields

Field	Function
31-4 —	Reserved bits. Reads only zeros.
3-0 Y1ADDR_MSB	Extended Address Most Significant Bits Extended MSB bits that will be combined with CHNL_OUT_BUF1_ADDR_Y[ADDR] to form an address that is more than 32-bits wide on the AXI bus.

13.4.5.1.43 Channel Output U-Buffer 1 Extended Address Bits Register (CHNL_U_BUF1_XTND_ADDR)

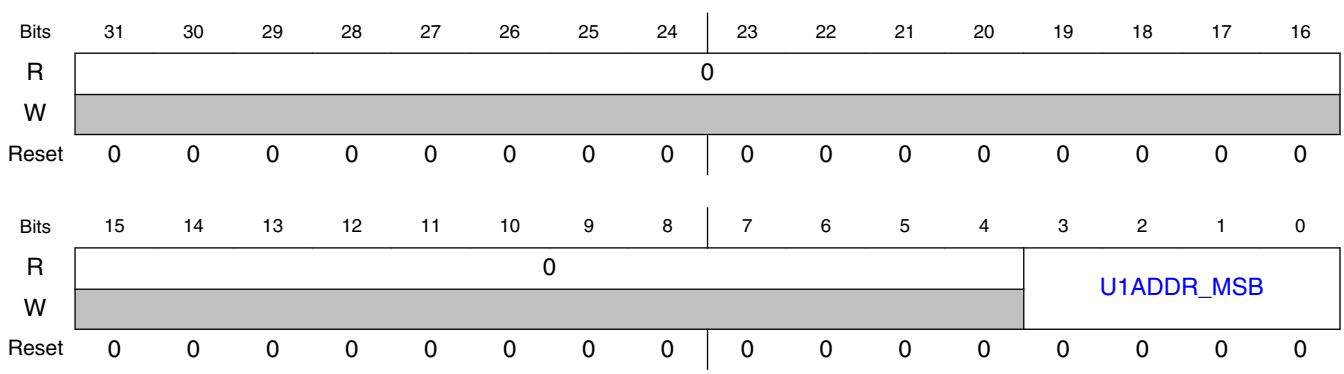
13.4.5.1.43.1 Offset

Register	Offset
CHNL_U_BUF1_XTND_ADDR	A4h

13.4.5.1.43.2 Function

This register provides additional MSB bits to address more than 32-bits of address space on the AXI bus.

13.4.5.1.43.3 Diagram



13.4.5.1.43.4 Fields

Field	Function
31-4 —	Reserved bits. Reads only zeros.
3-0 U1ADDR_MSB	Extended Address Most Significant Bits Extended MSB bits that will be combined with CHNL_OUT_BUF1_ADDR_U[ADDR] to form an address that is more than 32-bits wide on the AXI bus.

13.4.5.1.44 Channel Output V-Buffer 1 Extended Address Bits Register (CHNL_V_BUF1_XTND_ADDR)

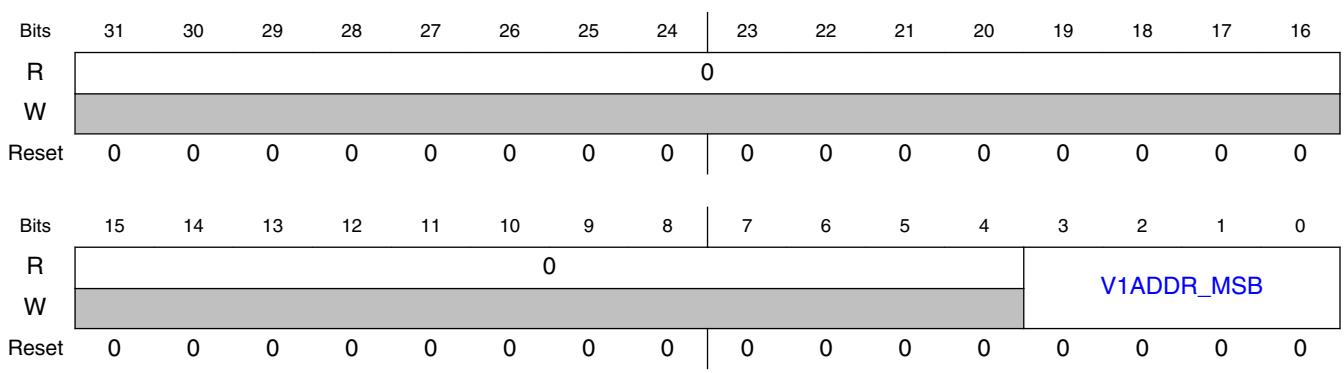
13.4.5.1.44.1 Offset

Register	Offset
CHNL_V_BUF1_XTND_ADDR	A8h

13.4.5.1.44.2 Function

This register provides additional MSB bits to address more than 32-bits of address space on the AXI bus.

13.4.5.1.44.3 Diagram



13.4.5.1.44.4 Fields

Field	Function
31-4	Reserved bits. Reads only zeros.

Table continues on the next page...

Field	Function
—	
3-0 V1ADDR_MSB	Extended Address Most Significant Bits Extended MSB bits that will be combined with CHNL_OUT_BUF1_ADDR_V[ADDR] to form an address that is more than 32-bits wide on the AXI bus.

13.4.5.1.45 Channel Output Y-Buffer 2 Extended Address Bits Register (CHNL_Y_BUF2_XTND_ADDR)

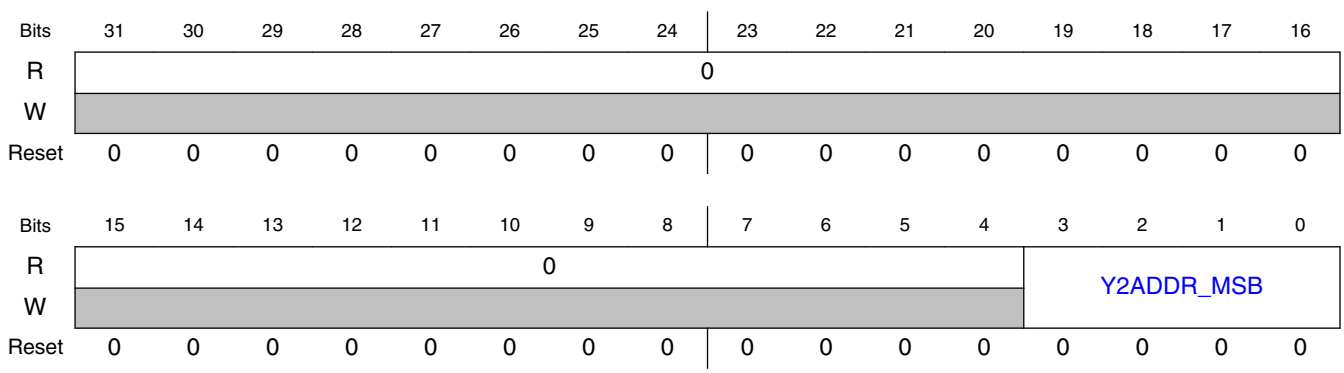
13.4.5.1.45.1 Offset

Register	Offset
CHNL_Y_BUF2_XTND_ADDR	ACh

13.4.5.1.45.2 Function

This register provides additional MSB bits to address more than 32-bits of address space on the AXI bus.

13.4.5.1.45.3 Diagram



13.4.5.1.45.4 Fields

Field	Function
31-4 —	Reserved bits. Reads only zeros.
3-0	Extended Address Most Significant Bits

Field	Function
Y2ADDR_MSB	Extended MSB bits that will be combined with CHNL_OUT_BUF2_ADDR_Y[ADDR] to form an address that is more than 32-bits wide on the AXI bus.

13.4.5.1.46 Channel Output U-Buffer 2 Extended Address Bits Register (CHNL_U_BUF2_XTND_ADDR)

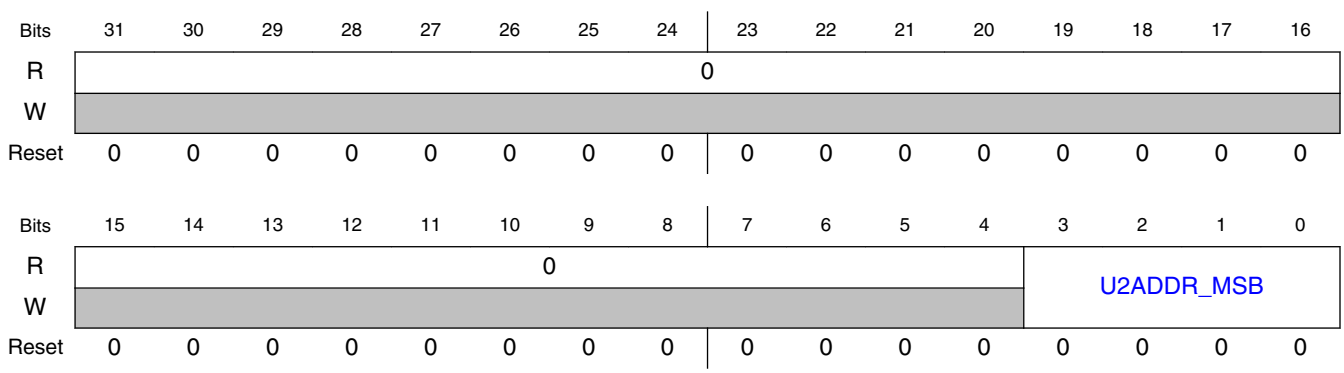
13.4.5.1.46.1 Offset

Register	Offset
CHNL_U_BUF2_XTND_ADDR	B0h

13.4.5.1.46.2 Function

This register provides additional MSB bits to address more than 32-bits of address space on the AXI bus.

13.4.5.1.46.3 Diagram



13.4.5.1.46.4 Fields

Field	Function
31-4 —	Reserved bits. Reads only zeros.
3-0 U2ADDR_MSB	Extended Address Most Significant Bits Extended MSB bits that will be combined with CHNL_OUT_BUF2_ADDR_U[ADDR] to form an address that is more than 32-bits wide on the AXI bus.

13.4.5.1.47 Channel Output V-Buffer 2 Extended Address Bits Register (CHNL_V_BUF2_XTND_ADDR)

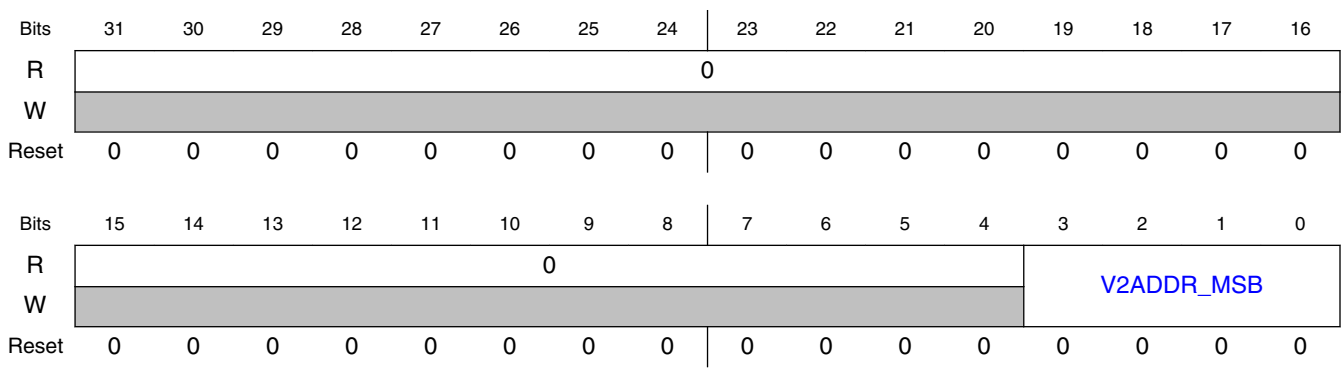
13.4.5.1.47.1 Offset

Register	Offset
CHNL_V_BUF2_XTND_ADDR	B4h

13.4.5.1.47.2 Function

This register provides additional MSB bits to address more than 32-bits of address space on the AXI bus.

13.4.5.1.47.3 Diagram



13.4.5.1.47.4 Fields

Field	Function
31-4 —	Reserved bits. Reads only zeros.
3-0 V2ADDR_MSB	Extended Address Most Significant Bits Extended MSB bits that will be combined with CHNL_OUT_BUF2_ADDR_V[ADDR] to form an address that is more than 32-bits wide on the AXI bus.

13.4.5.1.48 Channel Input Buffer Extended Address Bits Register (CHNL_IN_BUF_XTND_ADDR)

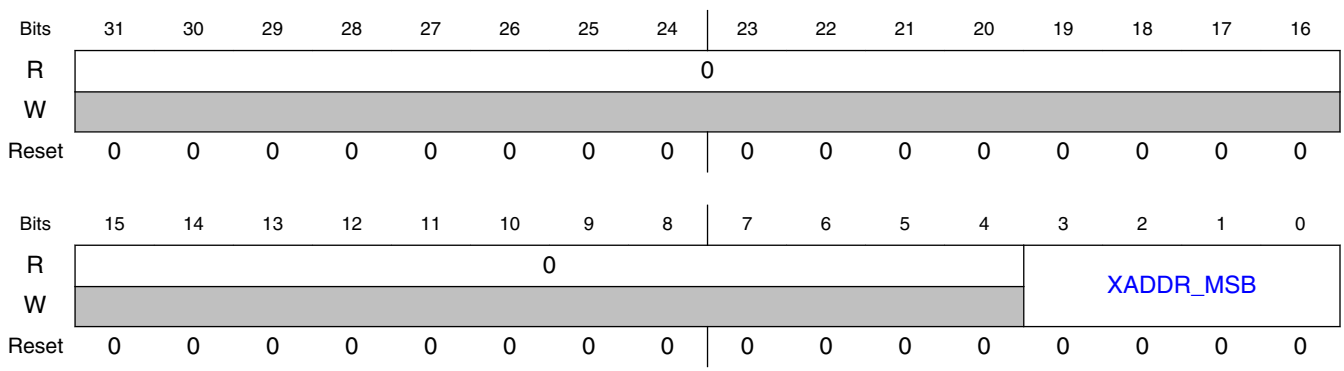
13.4.5.1.48.1 Offset

Register	Offset
CHNL_IN_BUF_XTND_ADDR	B8h

13.4.5.1.48.2 Function

This register provides additional MSB bits to address more than 32-bits of address space on the AXI bus for the read channel.

13.4.5.1.48.3 Diagram



13.4.5.1.48.4 Fields

Field	Function
31-4 —	Reserved bits. Reads only zeros.
3-0 XADDR_MSB	Extended Address Most Significant Bits Extended MSB bits that will be combined with CHNL_IN_BUF_ADDR[ADDR] to form an address that is more than 32-bits wide on the AXI bus.

13.5 MIPI CSI Host Controller (MIPI_CSI)

13.5.1 Overview

MIPI Camera Serial Interface (MIPI_CSI2) is the camera interface for this chip. It works with MIPI DPHY module to connect to the host processor.

MIPI_CSI2 supports RAW, YUV and RGB image formats when used in conjunction with output to ISI (Image Sensing Interface) or ISP (Image Signal Processing). '

13.5.1.1 Block Diagram

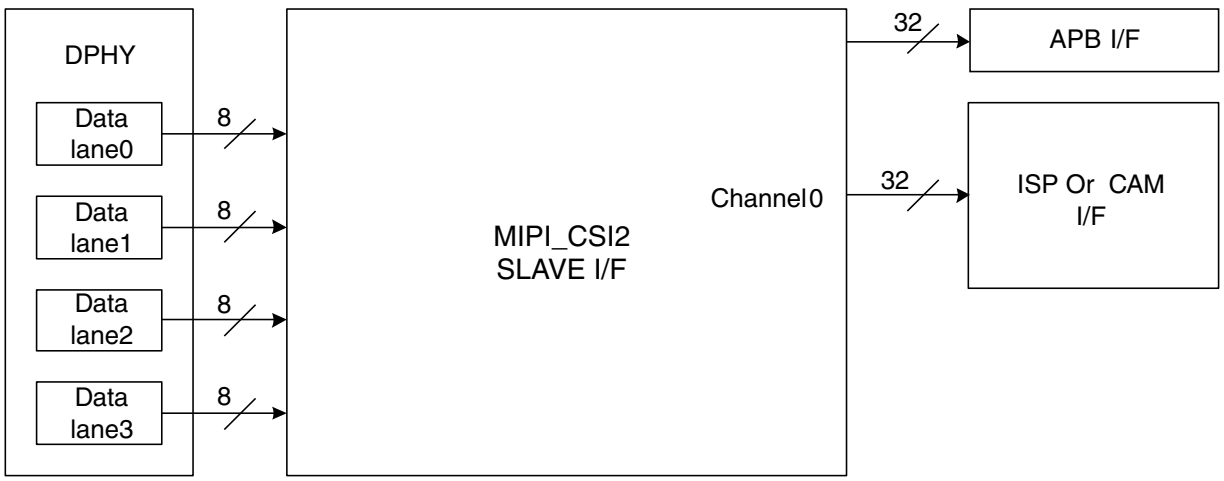


Figure 13-12. Block diagram of camera system with MIPI CSI Slave Interface

13.5.1.2 Features

- Compliant to MIPI D-PHY specification V1.2
- Compliant to MIPI CSI2 Specification V1.3 except for C-PHY feature and Embedded 8-bit non Image Data features.
- Support primary and secondary Image format
  - YUV420, YUV420 (Legacy), YUV420 (CSPS), YUV422 of 8-bits and 10-bits
  - RGB565, RGB666, RGB888
  - RAW6, RAW7, RAW8, RAW10, RAW12, RAW14
- Support up to 4 lanes of D-PHY
- Interfaces
  - Compatible to PPI(Protocol-to-PHY Interface) in MIPI D-PHY Specification
  - AMBA3.0 APB Slave for Register configuration.
  - Image output data buswidth : 32 bits
  - Supports data interleave
- Image memory
  - Size of SRAM is 4KB
- Pixel clock can be gated when no ppi data is coming.

13.5.2 Functional Description

13.5.2.1 MIPI CSI Slave and D-PHY I/F block diagram

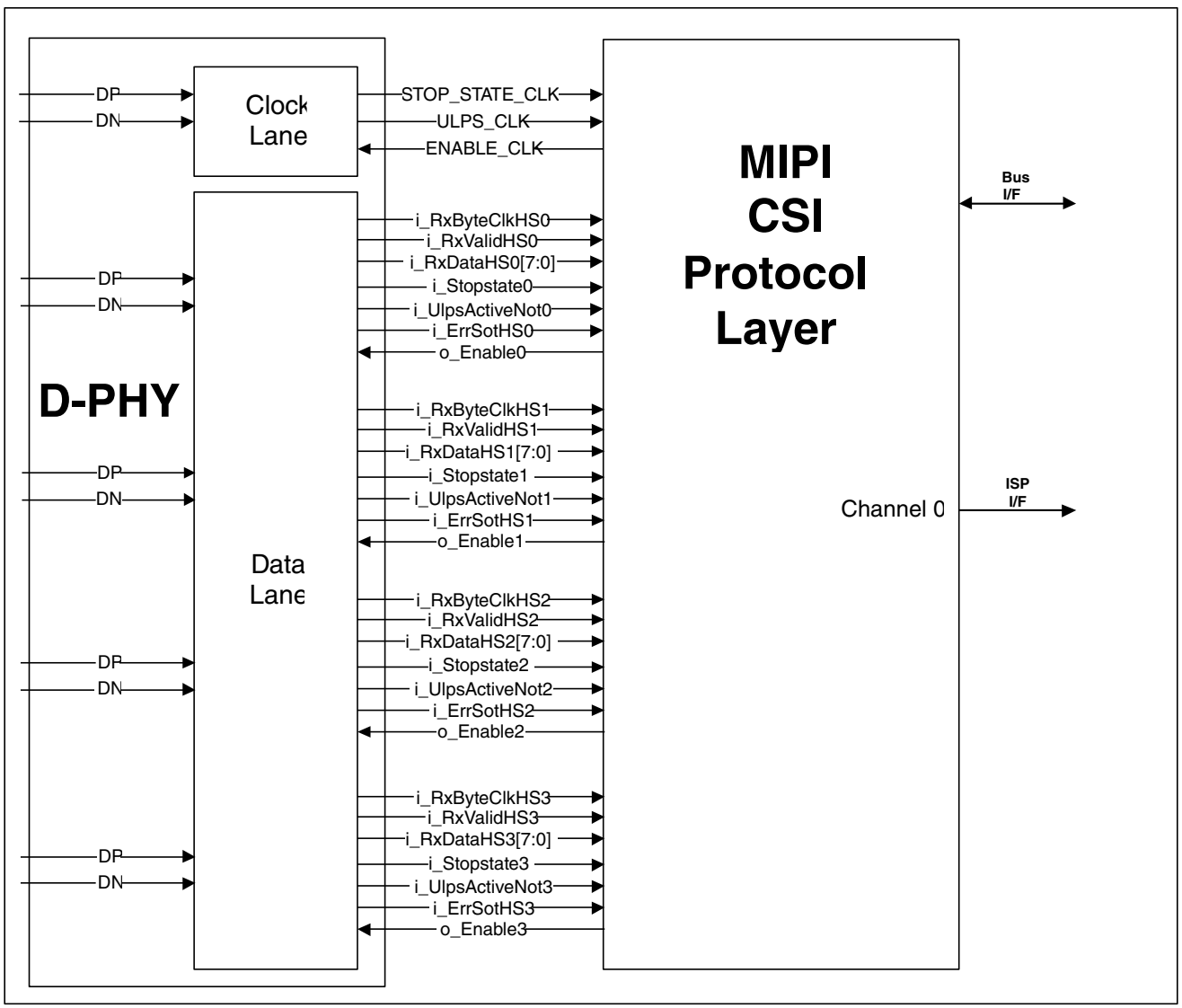


Figure 13-13. PPI I/F block diagram

13.5.2.2 Memory Bridge

The following table describes the configurable-sized memories.

Table 13-5. Memory bridge list

Usage	Memory Type	Size	Description
Image Data	2 port SRAM	4KB (1024 x 32bits)	Store the image data

There is the timing diagram for read memory in the figure below. If the mux for the split memories is needed, register for storing the address is necessary. Because the RDATA of memory is prepared at 1 clock later when the REN and RADDR are accepted.

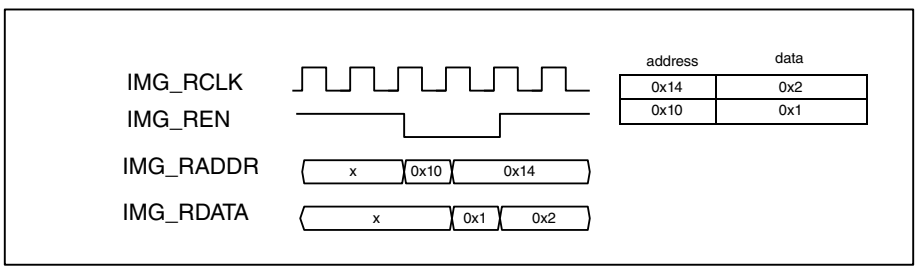


Figure 13-14. Timing diagram of Read memory

13.5.2.3 Protocol

13.5.2.3.1 D-PHY layer FSM

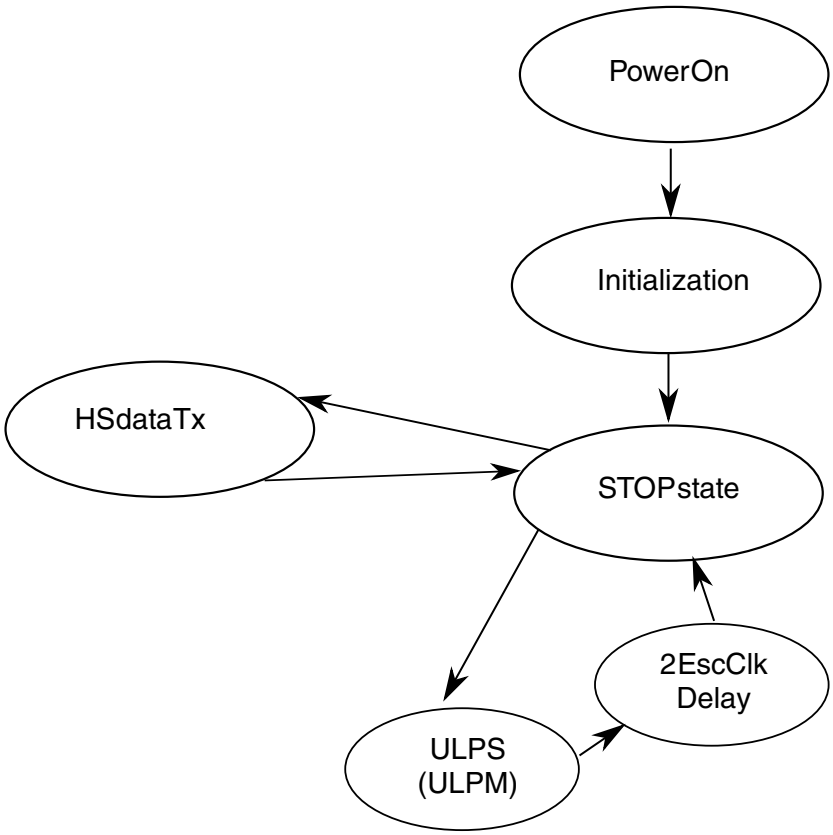


Figure 13-15. D-PHY Finite State Machine (FSM)

CSIS V3.6.3 system supports HSDT(High Speed Data transfer) and ULPS(ULPM – Utra-Low Power State or Mode) only. There is no trigger function, Low Power Data Transmission (LPDT) and Bus Turn Around (BTA).

13.5.2.3.2 High Speed Data Transfer

In the following figure, the upper 5 signals are related with clock lane and the lower 6 signals are related with data lane. Dp and Dn of upper signals are D-PHY channel of clock lane. BYTE_CLK is generated clock in the D-PHY. Dp and Dn of lower signals are Data lane. Another signals of lower are Phy Status signals. STOPstate (and STOPstateClk) indicates that differential channel state of D-PHY is LP11 (the voltage level of Dp and Dn is 1.2V)

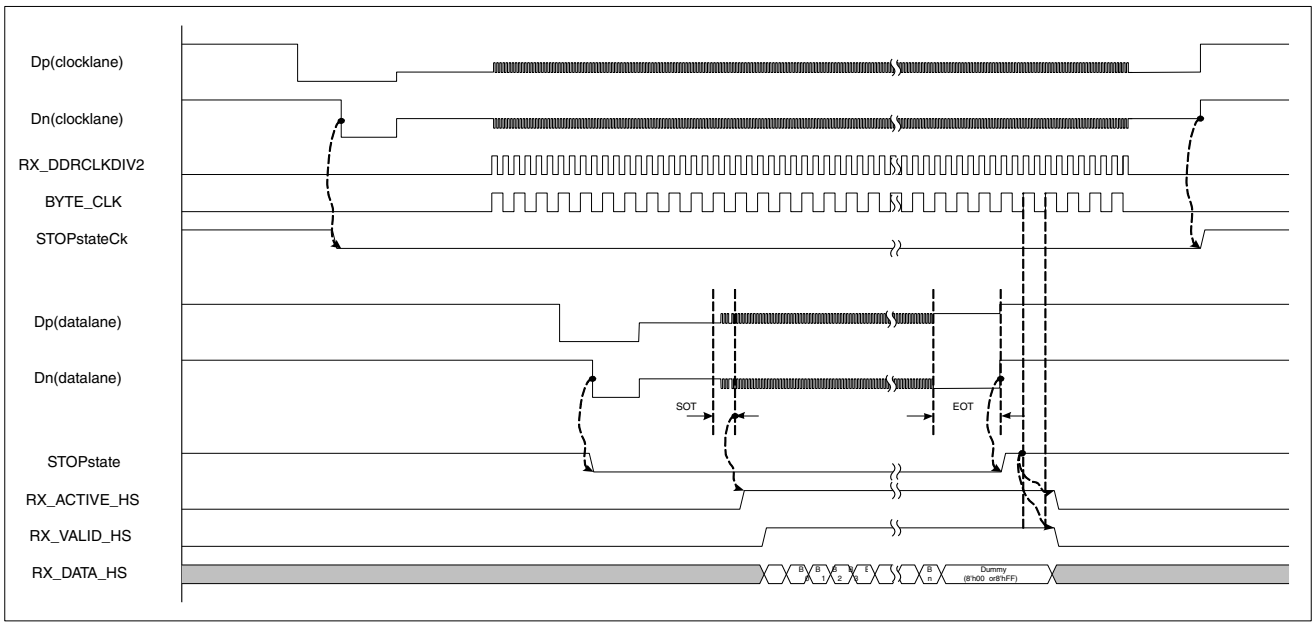


Figure 13-16. Timing Diagram of High Speed Data Transfer

13.5.2.3.3 Ultra-Low Power Mode

In the following figure, UlpsActiveNot* and STOPstate* is in PPI. ULPS command generated by D-PHY Tx is only for D-PHY Rx and it is decoded by D-PHY Rx and transferred to Link with Ulps data.

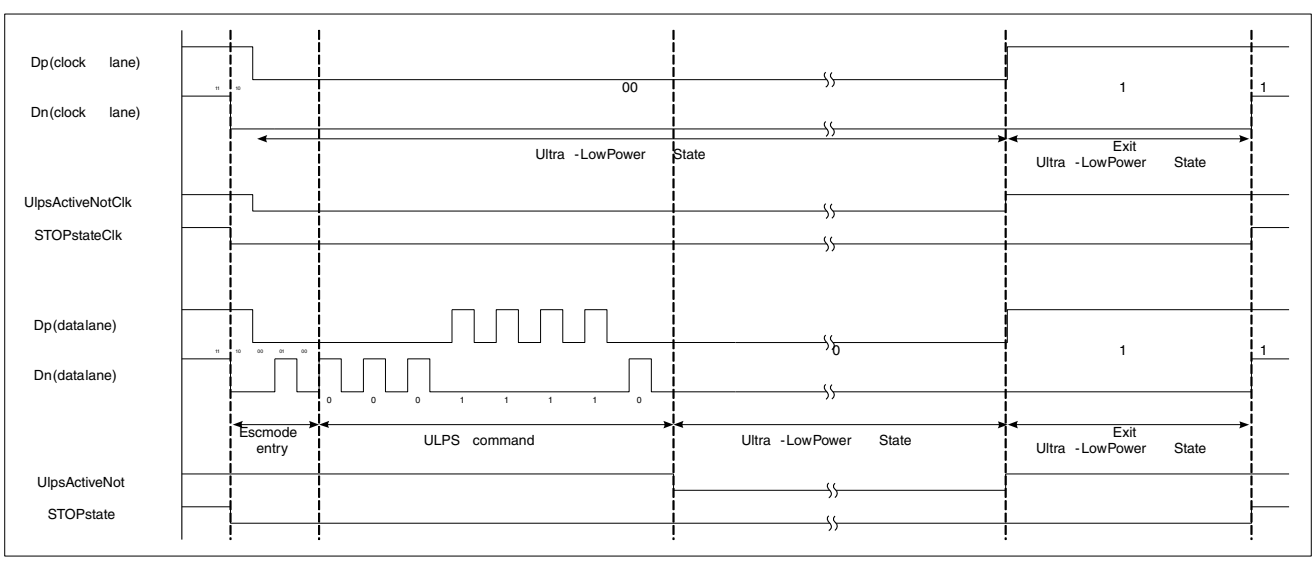


Figure 13-17. Timing Diagram of Ultra Low Power Mode

13.5.2.4 Interfaces

13.5.2.4.1 PPI Interface

Table 13-6. PPI I/F pin description

Port Name	I/O	Sync	Description
ENABLE_CLK	O	Async level	Enable clock lane module
STOP_STATE_CLK	I	Async level	Lanes in stop_state
ULPS_CLK	I	Async level	Clock lane is in Ultra Low-Power State (ULPS)
i_RxByteClkHS%	I	-	Rx byte clock in HS mode
i_RxValidHS%	I	Sync RX_BYTE_CLK_HS%	High-Speed receive data valid
i_RxDataHS%	I	Sync RX_BYTE_CLK_HS%	High-Speed receive data
i_ErrSotHS%	I	Sync RX_BYTE_CLK_HS%	Start of Transmission error (Pulse high)
o_Enable%	O	Async level	Enable Data lane module
i_Stopstate%	I	Async level	Lanes in stop_state
i_UlpsActiveNot%	I	Async level	Data Lane is in Ultra Low-Power State (ULPS)

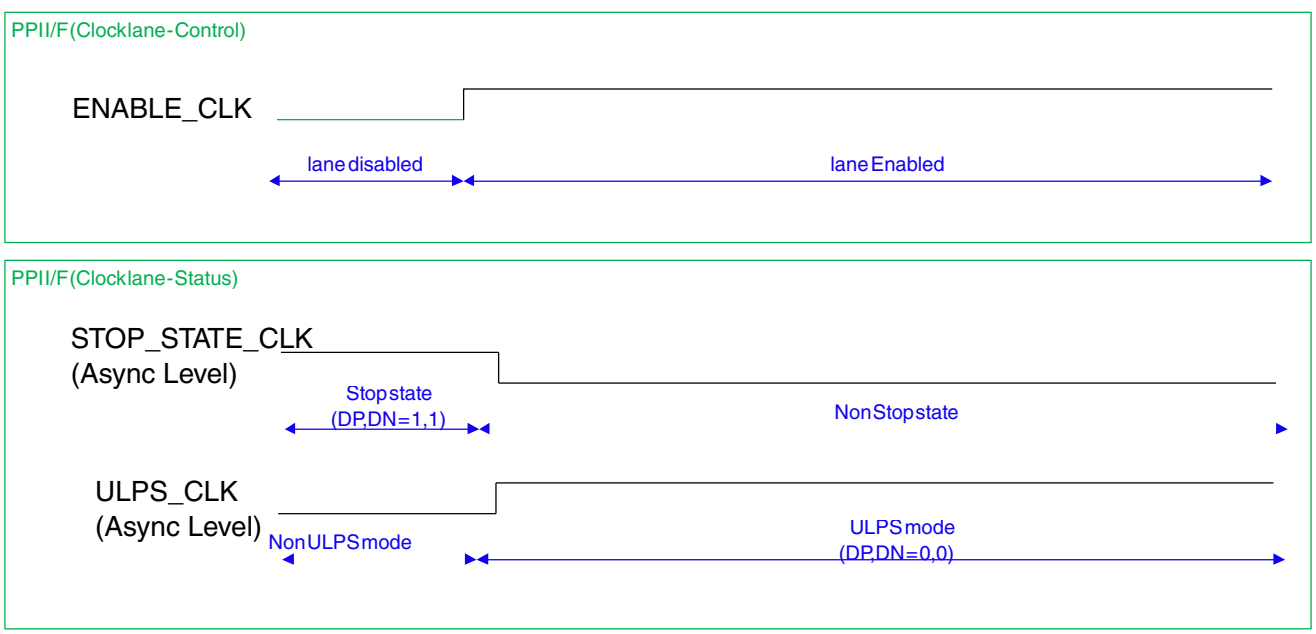


Figure 13-18. PPI I/F (Clock lane) timing diagram

NOTE

DPHY should supply i_RxByteClkHS% at least 15 clock period after falling i_RxValidHS%

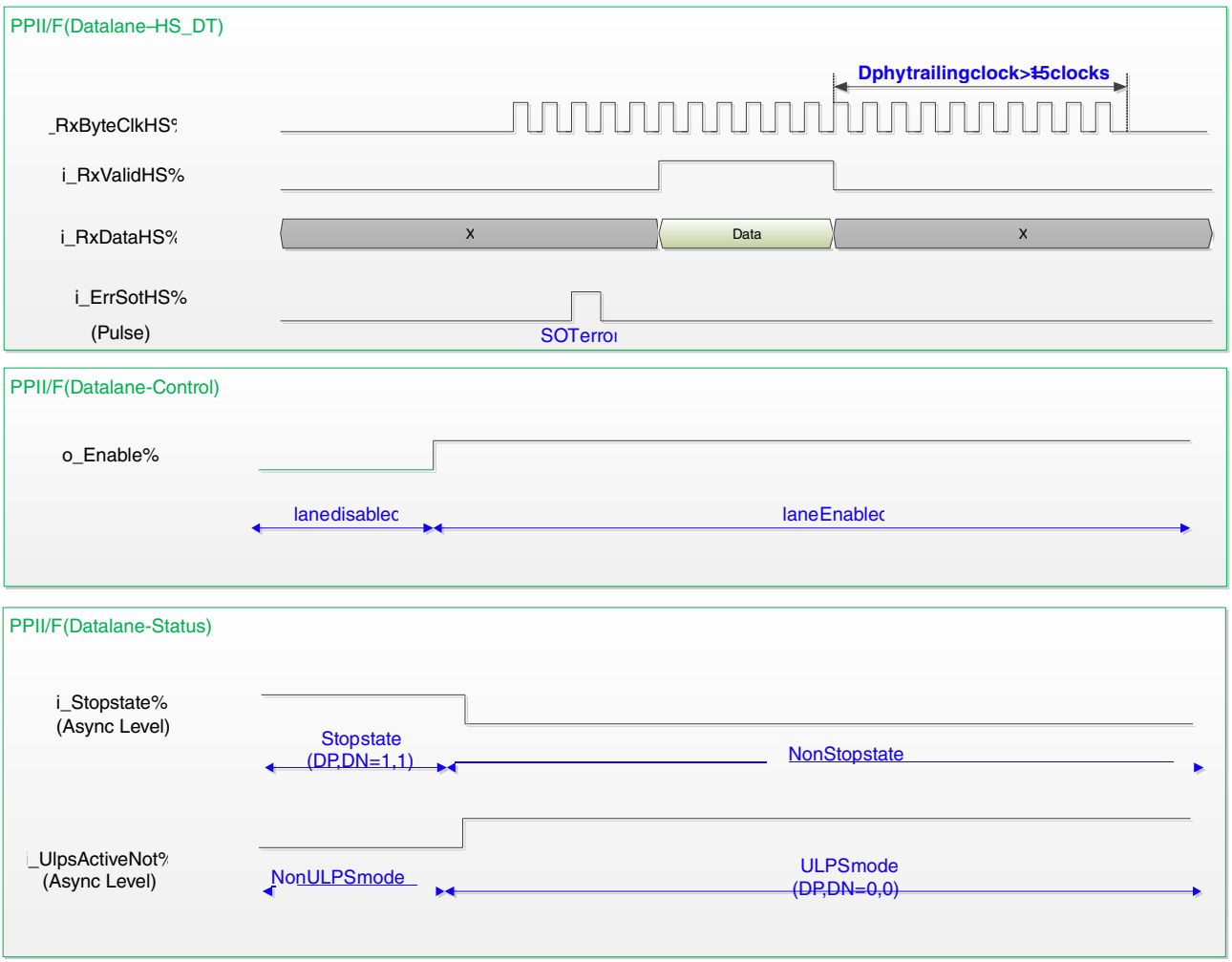


Figure 13-19. PPI I/F (Data lane) timing diagram

13.5.2.4.2 ISP Interface

CSIS V3.6.3 has output signals which are PIX_CLK, VVALID, HVALID, BVALID, and DATA.

Table 13-7. ISP I/F pin description

Port Name	I/O	Sync	Description
O_PIXCLK	O	-	Pixel clock. This signal is generated from I_ACLK.
O_VVALID	O	O_PIX_CLK	Vertical sync signal. This signal is set during whole frame.
O_HVALID	O	O_PIX_CLK	Horizontal sync signal.

Table continues on the next page...

Table 13-7. ISP I/F pin description (continued)

Port Name	I/O	Sync	Description
			This signal is set during each line.
O_BVALID	O	O_PIX_CLK	Byte valid signal.
O_DATA	O	O_PIX_CLK	Pixel data bus. The pixel is aligned based on its data type.

The following figure describes the timing diagram of ISP I/F. All signals are synchronized with the rising edge of PIXCLK.

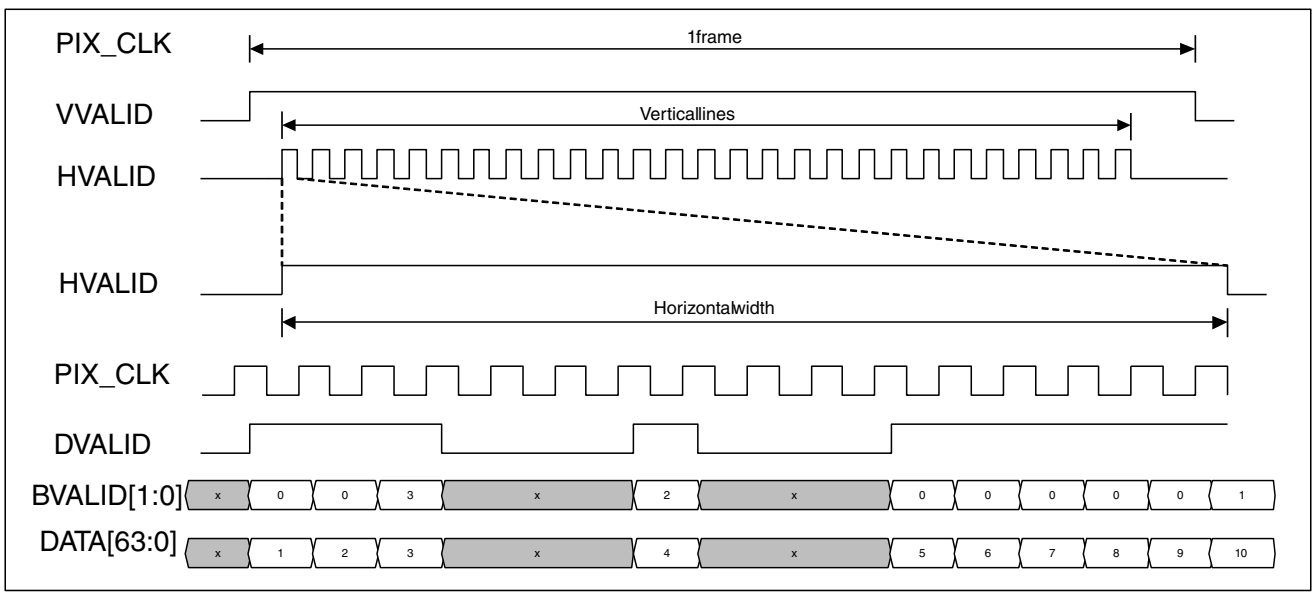


Figure 13-20. Output protocol of ISP Wrapper of CSIS C3.6.3

The following figure describes the sync timing of frame.

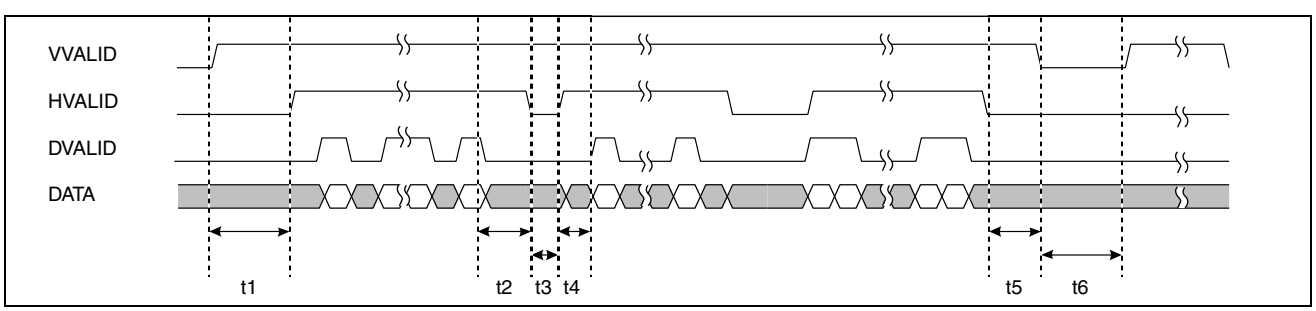


Figure 13-21. Frame sync timing diagram

The elements in the following table control the interval in Figure 13-21. Those elements are configurable by register in ISP SYNC register of CH0.

Table 13-8. Frame sync timing description

	Description	Minimum cycle (# PIX_CLK)	Maximum cycle (# PIX_CLK)
t1	Interval between rising of VVALID and first rising HVALID	1	-
t2	Interval between last falling of DVALID and falling HVALID	Hsync_Lintv + 1	-
t3	Interval between falling of HVALID and rising of next HVALID	1	-
t4	Interval between rising of HVALID and first rising of DVALID	0	-
t5	Interval between last falling of HVALID and falling of VVALID	1	-
t6	Interval between falling of VVALID and rising of next VVALID	1	-

13.5.2.4.3 SRAM Interface

MIPI CSI2 has output signals which are PIX_CLK, VVALID, HVALID, BVALID, and DATA.

Table 13-9. SRAM interface description

Port Name	I/O	Sync	Description
O_IMG_MEM_WCLK	O	-	Write clock for the SRAM of IMG_MEM
O_IMG_MEM_WEN	O	O_IMG_MEM_WCLK	Write enable for the SRAM of IMG_MEM (active low)
O_IMG_MEM_WADDR	O	O_IMG_MEM_WCLK	Write address for the SRAM of IMG_MEM
O_IMG_MEM_WDATA	O	O_IMG_MEM_WCLK	Write data for the SRAM of IMG_MEM
O_IMG_MEM_RCLK	O	O_IMG_MEM_WCLK	Read clock for the SRAM of IMG_MEM
O_IMG_MEM_REN	O	O_IMG_MEM_RCLK	Read enable for the SRAM of IMG_MEM (active low)
O_IMG_MEM_RADDR	O	O_IMG_MEM_RCLK	Read address for the SRAM of IMG_MEM
O_IMG_MEM_RDATA	I	O_IMG_MEM_RCLK	Read data from the SRAM of IMG_MEM

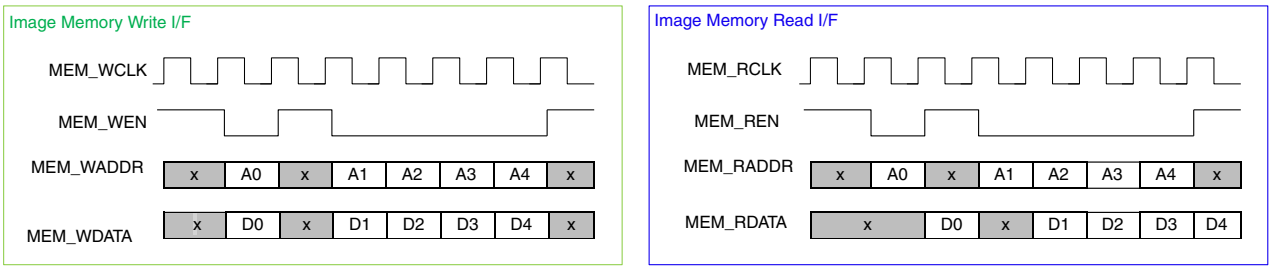


Figure 13-22. SRAM interface timing diagram

13.5.2.5 Image Data alignment

13.5.2.5.1 Normal Data alignment

The following figure describes the present of each bit in the data bus of ISP I/F.

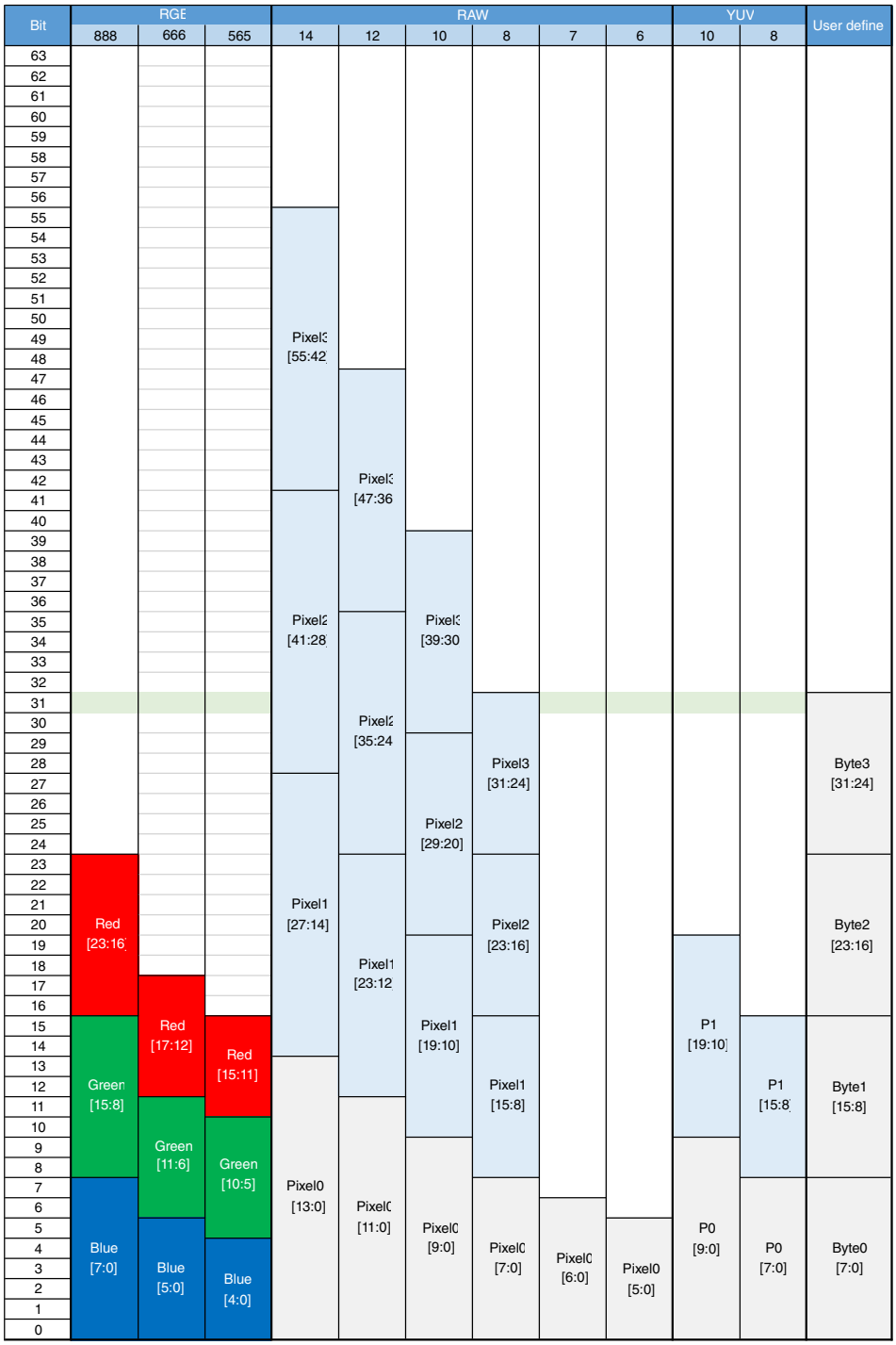


Figure 13-23. Pixel alignment

Pixel IF is adopted the little endian. If the current pixel mode is single, Pixel0 is a valid pixel. Higher pixels are the invalid signals. If the current pixel mode is dual, pixel0 and pixel1 is the valid pixels. In the case of Quad pixel mode, pixel0, pixel1, pixel2 and pixel3 is the valid pixels.

13.5.2.5.2 YUV data stream

Table 13-10. Output Image Stream of YUV formats

YUV format	Line	Image stream of content
YUV420 legacy	Odd	U1 Y1 Y2 U3 Y3 Y4 U5 Y5 ...
	Even	V1 Y1 Y2 V3 Y3 Y4 V5 Y5 ...
YUV420	Odd	Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 ...
	Even	U1 Y1 V1 Y2 U3 Y3 V3 Y4 ...
YUV422	Odd/Even	U1 Y1 V1 Y2 U3 Y3 V3 Y4 ...

13.5.2.5.3 RGB swapped data alignment

When the “RGB_SWAP” bit of ISP_CONFIG_CH% register is set, the RGB data alignment of data bus is swapped. This is shown in the following figure.

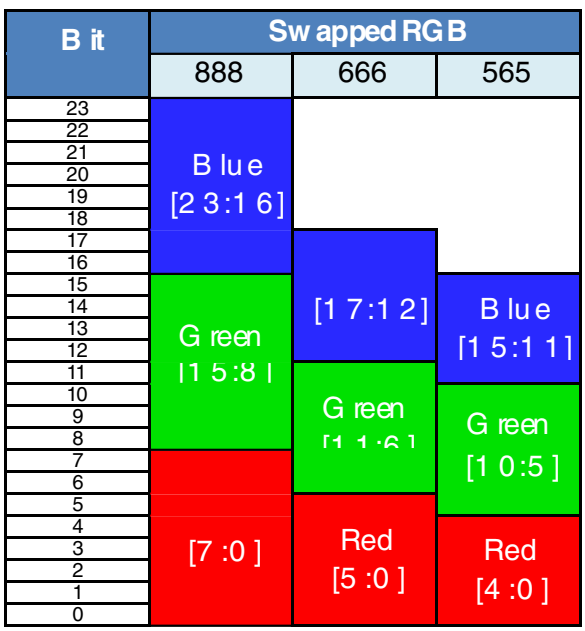


Figure 13-24. RGB swapped data alignment

13.5.2.5.4 Parallel-memory-storing data alignment

When ISP_CONFIG_CH[PARALLEL] bit is set, the outer bus width is 32 bits. If the byte count of the payload is not a multiple of 4, by performing the 0 padding makes the last data to 32bit.

13.5.2.6 Configuration

13.5.2.6.1 Image resolution

MIPI_CSI2 Slave block needs the configuration of Image resolution to measure one line length exactly and detect frame end. Vertical resolution register has 16bits (16'h0001 ~ 16'hFFFF) Horizontal resolution register has 16bits (16'h0004 ~ 16'hFFFC).

13.5.2.6.2 Image data format

In MIPI_CSI2, Image data format should be configured for transferring image data to ISP or CAM I/F.

13.5.2.6.3 YUV format

MIPI_CSI2 supports YUV formats: YUV420 8bit, YUV420 10bit, YUV420 Legacy, YUV420 8bit Chroma Shifted Pixel Sampling, YUV420 10bit Chroma Shifted Pixel Sampling, YUV422 8bit, and YUV422 10bit The following table describes Data type of YUV packets.

Table 13-11. Data type of YUV packets

Data type	Description
0x18	YUV420 8-bit
0x19	YUV420 10-bit
0x1A	Legacy YUV420 8-bit
0x1B	Reserved
0x1C	YUV420 8-bit CSPS (Chroma Shifted Pixel Sampling)
0x1D	YUV420 10-bit CSPS (Chroma Shifted Pixel Sampling)
0x1E	YUV422 8-bit
0x1F	YUV422 10-bit

13.5.2.6.4 RGB format

MIPI_CSI2 supports RGB formats: RGB444, RGB555, RGB565, RGB666, and RGB888. The following table describes Data type of RGB packets.

Table 13-12. Data type of RGB packets

Data type	Description
0x20	RGB444 (not supported)
0x21	RGB555 (not supported)
0x22	RGB565
0x23	RGB666
0x24	RGB888
0x25	Reserved

Table continues on the next page...

Table 13-12. Data type of RGB packets (continued)

Data type	Description
0x26	
0x27	

13.5.2.6.5 RAW format (Bayer RGB)

MIPI_CSI2 supports RAW formats: RAW6, RAW7, RAW8, RAW10, RAW12, and RAW14. The following table describes Data type of RAW packets.

Table 13-13. Data type of RAW packets

Data type	Description
0x28	RAW6
0x29	RAW7
0x2A	RAW8
0x2B	RAW10
0x2C	RAW12
0x2D	RAW14
0x2E	Reserved
0x2F	

13.5.2.6.6 Null and Blanking Data

For both the null and blanking data types CSIS V3.6.3 ignore the content of the packet payload data.

Table 13-14. Null and Blanking Data type

Data type	Description
0x10	Null
0x11	Blanking Data

13.5.2.7 Synchronization Short Packet Data type

Image Frame begins with a Frame Start and finishes with a Frame End packet. With these Frame synch packet, CSIS V3.6.3 generate Vertical Sync signal (VVALID). The Line synchronization packet (Line Start and Line End) generate Horizontal Sync signal (HVALID). But it can be generated by image data packet so that CSIS V3.6.3 doesn't support the Line Synchronization packets (0x02~0x03) and ignore it if it is received.

Table 13-15. Synchronization short packet data type

Data Type	Description
0x00	Frame Start Code
0x01	Frame End Code
0x02	Line Start Code (Not supported)
0x03	Line End Code (Not supported)
0x04 to 0x07	Reserved

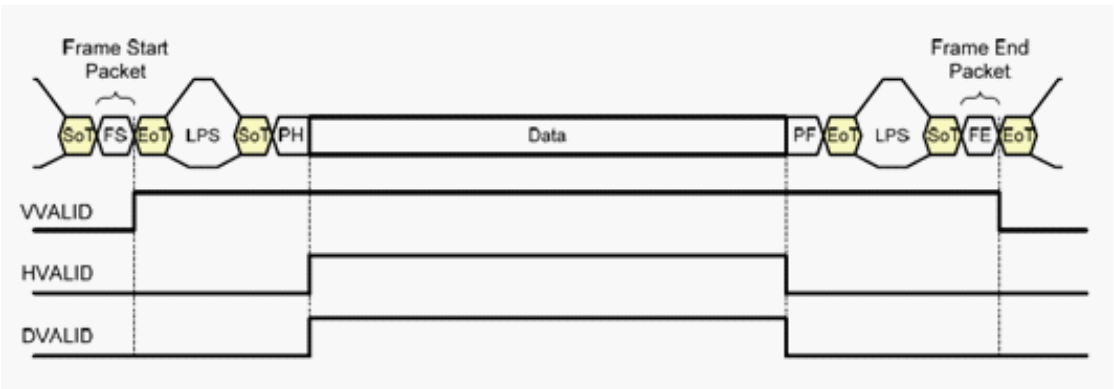


Figure 13-25. Synchronization example

13.5.2.8 Shadow Update

The configurations related to ISP registers (which are dataformat, vci, resolution and so on.) should not be varied during the run-time. So, they are updated by programming 'update_shadow' (in MIPI_CSI_CSIS_COMMON_CTRL register) in idle state. The scheme is described in the following figure.

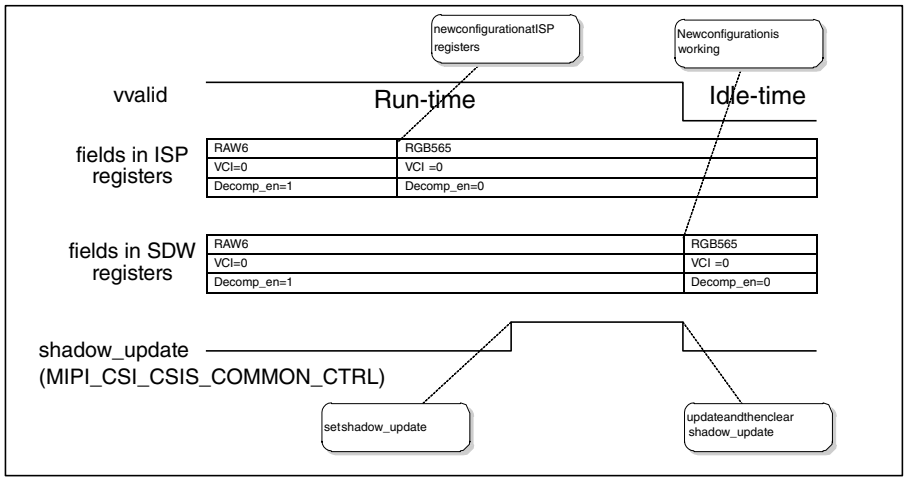


Figure 13-26. Operation of shadow_update

13.5.2.9 Clock Specification

MIPI_CSI2 have 3 clock sources: RX_BYTE_CLK_HS%, I_PCLK, and I_WRAP_CLK.

Table 13-16. Clock sources

Clock source	Description
I_PCLK	PCLK is system clock generated by general processor PLL.
i_RxByteClkHS%	This signal comes from data lane 0,1,2,3 of D-PHY.
I_ACLK	I_ACLK is external clock for pixel clock which is for using ISP or CAM I/F

13.5.2.9.1 Clock Domain

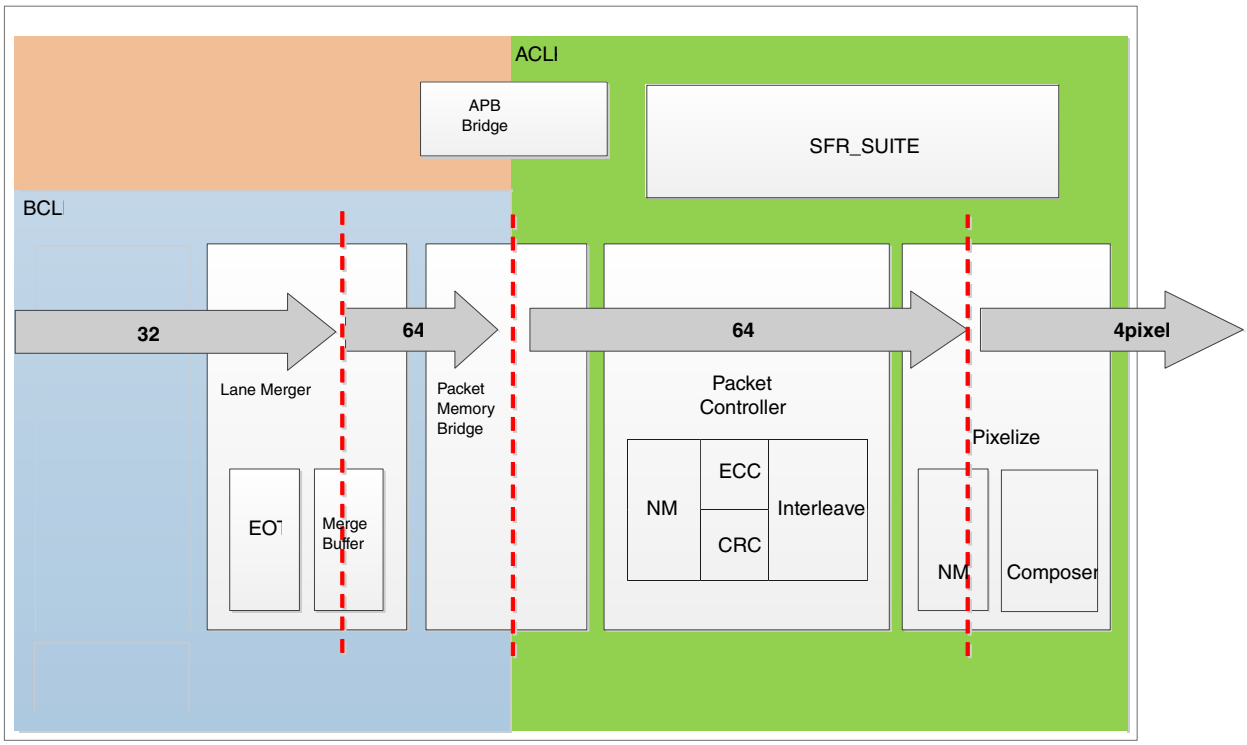


Figure 13-27. Block diagram of clock domain

The number of clock domain of CSIS V3.6.3 is 3:

- BYTE clock (RxByteCLKSH0)
- Pixel clock (I_ACLK)
- APB clock (I_PCLK)

The relationship between input and output bandwidth is that the output bandwidth should be faster than input bandwidth. There is an equation of previous relationship:

(FREQ. OF RX_BYTE_CLK_HS) X (NUMBER OF DATA LANE) X 8BITS ≤ (FREQ. OF PIXEL CLOCK) X (BITWIDTH OF IMAGE FORMAT) X (NUMBER OF PIXEL PER CLOCK)

When parallel mode is on, **(BITWIDTH OF IMAGE FORMAT) X (NUMBER OF PIXEL PER CLOCK) = 32**

13.5.2.10 Reset

Table 13-17. Reset

Reset	Usage
I_PRESETn	Global Asynchronous reset

13.5.2.11 Interrupts

Table 13-18. Interrupts

Signal	Description
O_INT_CSIS	Interrupt request signal

13.5.3 External Signals

Table 13-19. External Signals

Signal	Description	Direction
CLK_N	Negative MIPI Rx Differential Clock	I
CLK_P	Positive MIPI Rx Differential Clock	I
Dn_N	Negative MIPI Rx Differential Data	I
Dn_P	Positive MIPI Rx Differential Data	I

13.5.4 Initial Sequence

CSI Rx should be reset with D-PHY. Below Signal diagram is referred from Samsung D-PHY datasheet. This document describes the SFR setting and timing information.

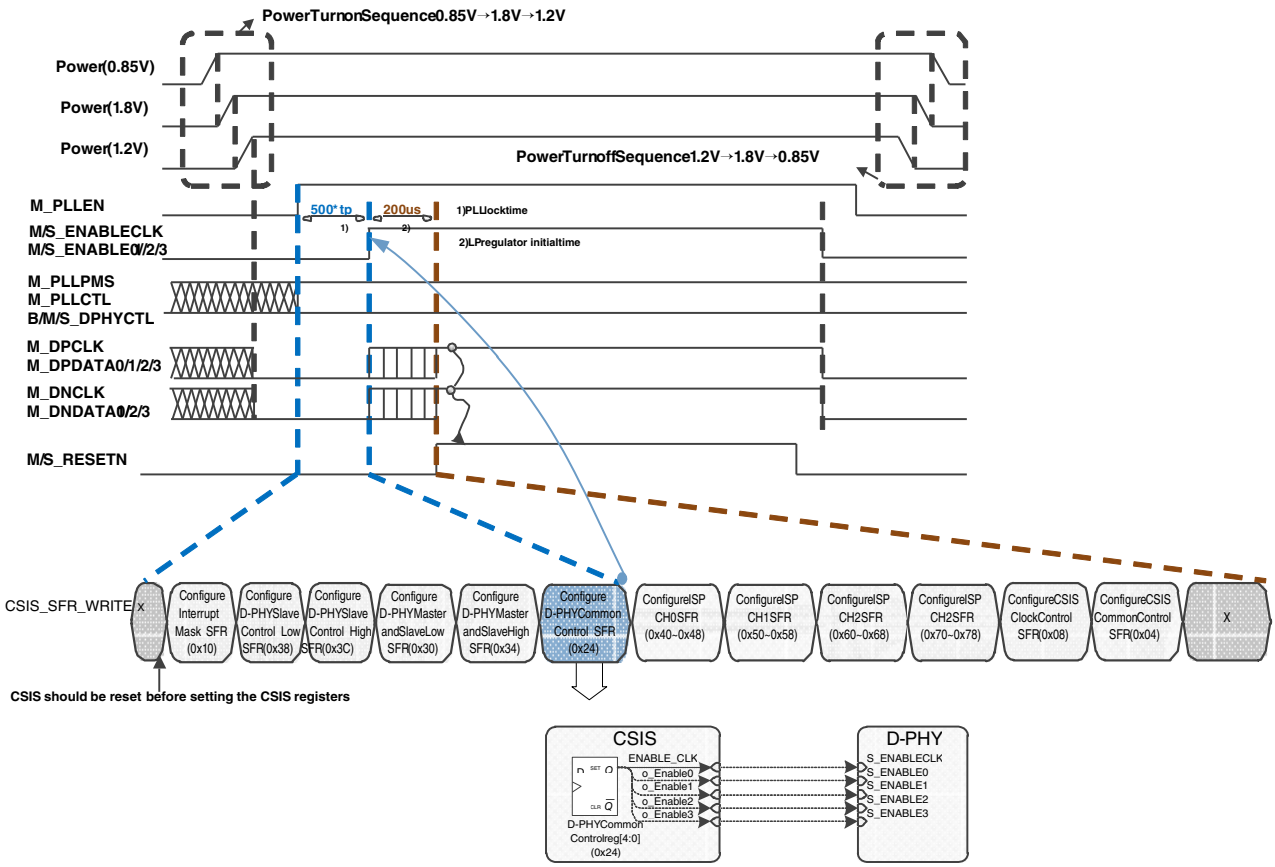


Figure 13-28. Initialize sequence

13.5.5 Application Information

13.5.5.1 Programming Model

Table 13-20. Behavior programming each case

No.	Pseudo Code	Description
1	APB_WRITE(MIPI_CSI_INTERRUPT_MASK, -)	set Interrupt Mask
2	APB_WRITE(MIPI_CSI_DPHY_MASTER_SLAVE_CTRL_LOW, -)	set DPHYBCTRL LSB
3	APB_WRITE(MIPI_CSI_DPHY_MASTER_SLAVE_CTRL_HIGH, -)	set DPHYBCTRL MSB
4	APB_WRITE(MIPI_CSI_DPHY_COMMON_CTRL, 0x11)	set DPHY HSSETTLE
5	APB_WRITE(MIPI_CSI_ISP_CONFIG_CH0, -)	set ISP_CH0 configuration

Table continues on the next page...

Table 13-20. Behavior programming each case (continued)

No.	Pseudo Code	Description
6	APB_WRITE(MIPI_CSI_ISP_RESOLUTION_CH0, -)	set ISP_CH0 resolution
7	APB_WRITE(MIPI_CSI_ISP_SYNC_CH0, -)	set ISP_CH0 sync
8	APB_WRITE(MIPI_CSI_CSIS_CLOCK_CTRL, -)	set CSIS clock control
9	APB_WRITE(MIPI_CSI_CSIS_COMMON_CTRL, -)	set CSIS common control(enable CSI)

13.5.6 MIPI CSI Memory Map/Register Definition

MIPI CSI1 Base address: 32E4_0000h

MIPI CSI2 Base address: 32E5_0000h

MIPI_CSI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
32E4_0004	CSIS Common Control Register (MIPI_CSI1_CSIS_COMMON_CTRL)	32	R/W	0000_4000h	13.5.6.1/ 5411
32E4_0008	CSIS Clock Control Register (MIPI_CSI1_CSIS_CLOCK_CTRL)	32	R/W	0000_00F0h	13.5.6.2/ 5413
32E4_0010	Interrupt mask register 0 (MIPI_CSI1_INTERRUPT_MASK_0)	32	R/W	0000_0000h	13.5.6.3/ 5414
32E4_0014	Interrupt source register 0 (MIPI_CSI1_INTERRUPT_SOURCE_0)	32	R/W	0000_0000h	13.5.6.4/ 5416
32E4_0018	Interrupt mask register 1 (MIPI_CSI1_INTERRUPT_MASK_1)	32	R/W	0000_0000h	13.5.6.5/ 5418
32E4_001C	Interrupt source register 1 (MIPI_CSI1_INTERRUPT_SOURCE_1)	32	R/W	0000_0000h	13.5.6.6/ 5419
32E4_0020	D-PHY status register (MIPI_CSI1_DPHY_STATUS)	32	R/W	0000_00F1h	13.5.6.7/ 5420
32E4_0024	D-PHY common control register (MIPI_CSI1_DPHY_COMMON_CTRL)	32	R/W	0000_0000h	13.5.6.8/ 5422
32E4_0030	D-PHY Master and Slave Control register Low (MIPI_CSI1_DPHY_MASTER_SLAVE_CTRL_LOW)	32	R/W	0000_0000h	13.5.6.9/ 5423
32E4_0034	D-PHY Master and Slave Control register HIGH (MIPI_CSI1_DPHY_MASTER_SLAVE_CTRL_HIGH)	32	R/W	0000_0000h	13.5.6.10/ 5426
32E4_0038	D-PHY Slave Control register Low (MIPI_CSI1_DPHY_SLAVE_CTRL_LOW)	32	R/W	0000_0000h	13.5.6.11/ 5429
32E4_003C	D-PHY Slave Control register HIGH (MIPI_CSI1_DPHY_SLAVE_CTRL_HIGH)	32	R/W	0000_0000h	13.5.6.12/ 5431

Table continues on the next page...

MIPI_CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
32E4_0040	ISP Configuration Register (MIPI_CSI1_ISP_CONFIG_CH0)	32	R/W	0000_08FCh	13.5.6.13/ 5433
32E4_0044	ISP Resolution Register (MIPI_CSI1_ISP_RESOLUTION_CH0)	32	R/W	8000_8000h	13.5.6.14/ 5434
32E4_0048	ISP SYNC Register (MIPI_CSI1_ISP_SYNC_CH0)	32	R/W	0000_0000h	13.5.6.15/ 5435
32E4_0080	Shadow Configuration Register (MIPI_CSI1_SHADOW_CONFIG_CH0)	32	R	0000_08FCh	13.5.6.16/ 5436
32E4_0084	Shadow Resolution Register (MIPI_CSI1_SHADOW_RESOLUTION_CH0)	32	R	8000_8000h	13.5.6.17/ 5437
32E4_0088	Shadow SYNC Register (MIPI_CSI1_SHADOW_SYNC_CH0)	32	R	0000_0000h	13.5.6.18/ 5437
32E4_0100	Frame Counter (MIPI_CSI1_FRAME_COUNTER_CH0)	32	R/W	0000_0000h	13.5.6.19/ 5438
32E4_0110	Line Interrupt Ratio (MIPI_CSI1_LINE_INTERRUPT_RATIO_CH0)	32	R/W	0000_0000h	13.5.6.20/ 5438
32E5_0004	CSIS Common Control Register (MIPI_CSI2_CSIS_COMMON_CTRL)	32	R/W	0000_4000h	13.5.6.1/ 5411
32E5_0008	CSIS Clock Control Register (MIPI_CSI2_CSIS_CLOCK_CTRL)	32	R/W	0000_00F0h	13.5.6.2/ 5413
32E5_0010	Interrupt mask register 0 (MIPI_CSI2_INTERRUPT_MASK_0)	32	R/W	0000_0000h	13.5.6.3/ 5414
32E5_0014	Interrupt source register 0 (MIPI_CSI2_INTERRUPT_SOURCE_0)	32	R/W	0000_0000h	13.5.6.4/ 5416
32E5_0018	Interrupt mask register 1 (MIPI_CSI2_INTERRUPT_MASK_1)	32	R/W	0000_0000h	13.5.6.5/ 5418
32E5_001C	Interrupt source register 1 (MIPI_CSI2_INTERRUPT_SOURCE_1)	32	R/W	0000_0000h	13.5.6.6/ 5419
32E5_0020	D-PHY status register (MIPI_CSI2_DPHY_STATUS)	32	R/W	0000_00F1h	13.5.6.7/ 5420
32E5_0024	D-PHY common control register (MIPI_CSI2_DPHY_COMMON_CTRL)	32	R/W	0000_0000h	13.5.6.8/ 5422
32E5_0030	D-PHY Master and Slave Control register Low (MIPI_CSI2_DPHY_MASTER_SLAVE_CTRL_LOW)	32	R/W	0000_0000h	13.5.6.9/ 5423
32E5_0034	D-PHY Master and Slave Control register HIGH (MIPI_CSI2_DPHY_MASTER_SLAVE_CTRL_HIGH)	32	R/W	0000_0000h	13.5.6.10/ 5426
32E5_0038	D-PHY Slave Control register Low (MIPI_CSI2_DPHY_SLAVE_CTRL_LOW)	32	R/W	0000_0000h	13.5.6.11/ 5429
32E5_003C	D-PHY Slave Control register HIGH (MIPI_CSI2_DPHY_SLAVE_CTRL_HIGH)	32	R/W	0000_0000h	13.5.6.12/ 5431
32E5_0040	ISP Configuration Register (MIPI_CSI2_ISP_CONFIG_CH0)	32	R/W	0000_08FCh	13.5.6.13/ 5433
32E5_0044	ISP Resolution Register (MIPI_CSI2_ISP_RESOLUTION_CH0)	32	R/W	8000_8000h	13.5.6.14/ 5434

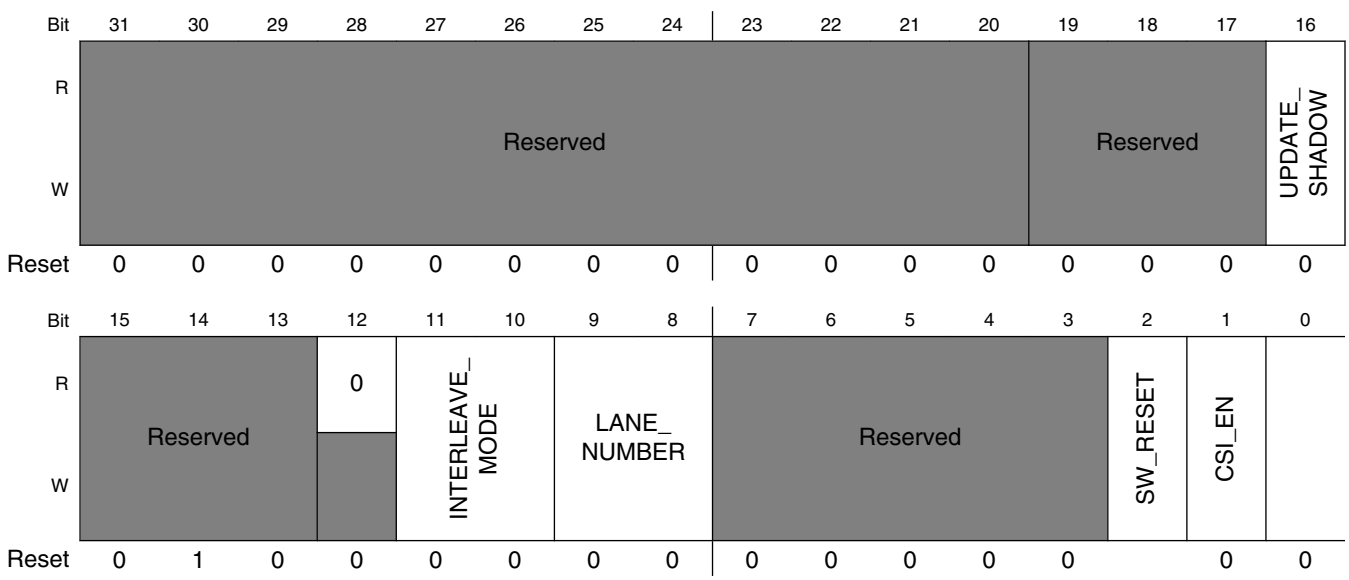
Table continues on the next page...

MIPI_CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
32E5_0048	ISP SYNC Register (MIPI_CSI2_ISP_SYNC_CH0)	32	R/W	0000_0000h	13.5.6.15/ 5435
32E5_0080	Shadow Configuration Register (MIPI_CSI2_SHADOW_CONFIG_CH0)	32	R	0000_08FCh	13.5.6.16/ 5436
32E5_0084	Shadow Resolution Register (MIPI_CSI2_SHADOW_RESOLUTION_CH0)	32	R	8000_8000h	13.5.6.17/ 5437
32E5_0088	Shadow SYNC Register (MIPI_CSI2_SHADOW_SYNC_CH0)	32	R	0000_0000h	13.5.6.18/ 5437
32E5_0100	Frame Counter (MIPI_CSI2_FRAME_COUNTER_CH0)	32	R/W	0000_0000h	13.5.6.19/ 5438
32E5_0110	Line Interrupt Ratio (MIPI_CSI2_LINE_INTERRUPT_RATIO_CH0)	32	R/W	0000_0000h	13.5.6.20/ 5438

13.5.6.1 CSIS Common Control Register (MIPI_CSIx_CSIS_COMMON_CTRL)

Address: Base address + 4h offset



MIPI_CSIx_CSIS_COMMON_CTRL field descriptions

Field	Description
31–20 -	This field is reserved.
19–17 -	This field is reserved.

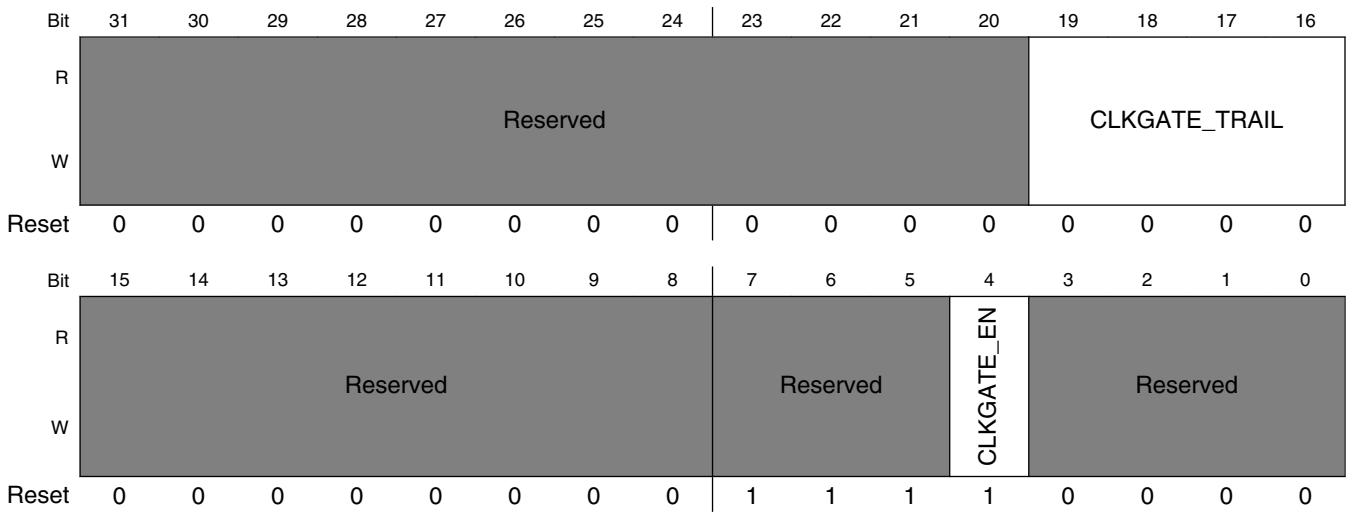
Table continues on the next page...

MIPI_CSIx_CSIS_COMMON_CTRL field descriptions (continued)

Field	Description
16 UPDATE_ SHADOW	Strobe of updating shadow registers After configuration, User has to set this bit for updating shadow registers. This bit is cleared automatically after updating shadow registers. 1 update the shadow registers 0 default
15–13 -	This field is reserved.
12 Reserved	This read-only field is reserved and always has the value 0.
11–10 INTERLEAVE_ MODE	Select Interleave mode ISP Configuration register of CH# is used for data interleave. 00 CH0 only, no data interleave 01 DT (Data type) only 10 Reserved 11 Reserved
9–8 LANE_NUMBER	Number of data lane 00 1 data lane 01 2 data lane 10 3 data lane 11 4 data lane
7–3 -	This field is reserved. -
1 SW_RESET	Software reset All writable registers in CSI2 go back to initial state. This bit is de-asserted automatically after the software reset is done. **NOTE:** Notice : Almost MIPI CSI2 block uses “ByteClk” from D- PHY. This “ByteClk” is not continuous clock. User has to assert software reset when Camera module is turned off. 0 Ready 1 Reset
0 CSI_EN	MIPI CSI2 system enable 0 Disable 1 Enable

13.5.6.2 CSIS Clock Control Register (MIPI_CSIx_CSIS_CLOCK_CTRL)

Address: Base address + 8h offset

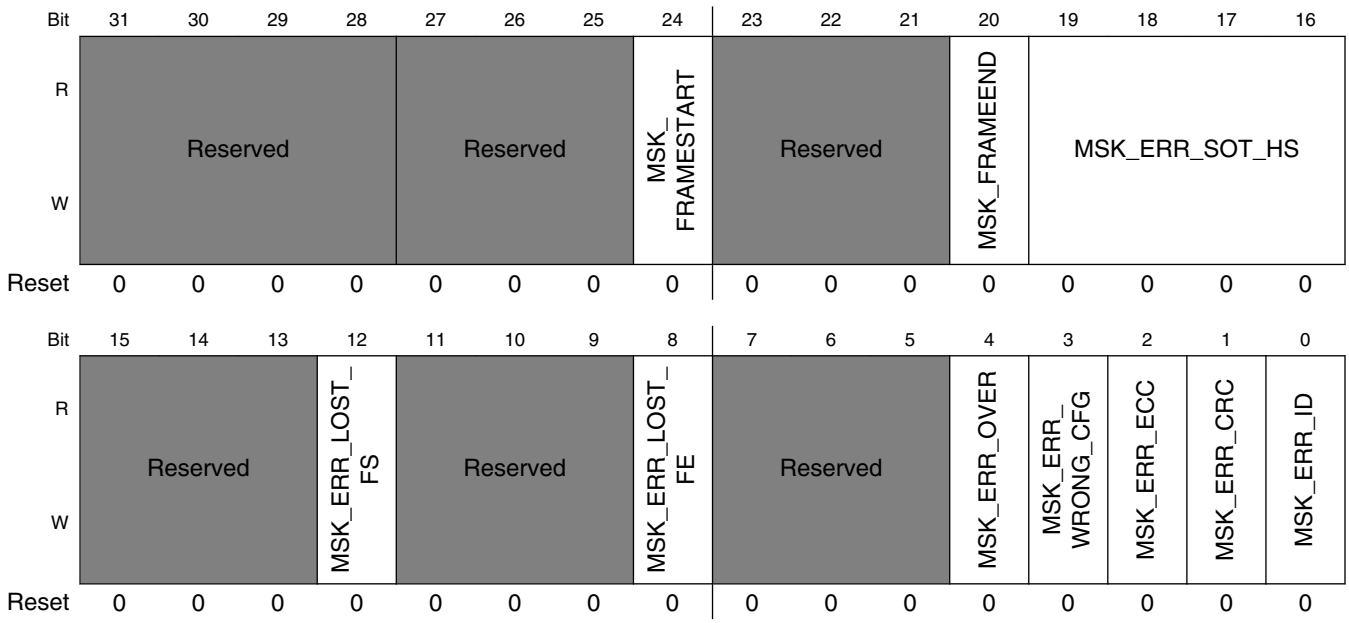


MIPI_CSIx_CSIS_CLOCK_CTRL field descriptions

Field	Description
31–20 -	This field is reserved.
19–16 CLKGATE_ TRAIL	0 ~ 3 (1~4 Trailing clocks) **NOTE:** Trailing clocks used for poping the F/F of ISP or CAM
15–8 -	This field is reserved.
7–5 -	This field is reserved.
4 CLKGATE_EN	0 Pixel clock is always alive 1 Pixel clock is alive during the interval of frame
-	This field is reserved.

13.5.6.3 Interrupt mask register 0 (MIPI_CSIx_INTERRUPT_MASK_0)

Address: Base address + 10h offset



MIPI_CSIx_INTERRUPT_MASK_0 field descriptions

Field	Description
31–28 -	This field is reserved.
27–25 -	This field is reserved.
24 MSK_ FRAMESTART	FS packet is received, CH0. 0 Disable (masked) 1 Enable (unmasked)
23–21 -	This field is reserved.
20 MSK_ FRAMEEND	FE packet is received, CH0. 0 Disable (masked) 1 Enable (unmasked)
19–16 MSK_ERR_ SOT_HS	Start of transmission error [Lane3, Lane2, Lane1, Lane0] 0 Disable (masked) 1 Enable (unmasked)
15–13 -	This field is reserved.
12 MSK_ERR_ LOST_FS	Lost of Frame Start packet, CH0. 0 Disable (masked) 1 Enable (unmasked)

Table continues on the next page...

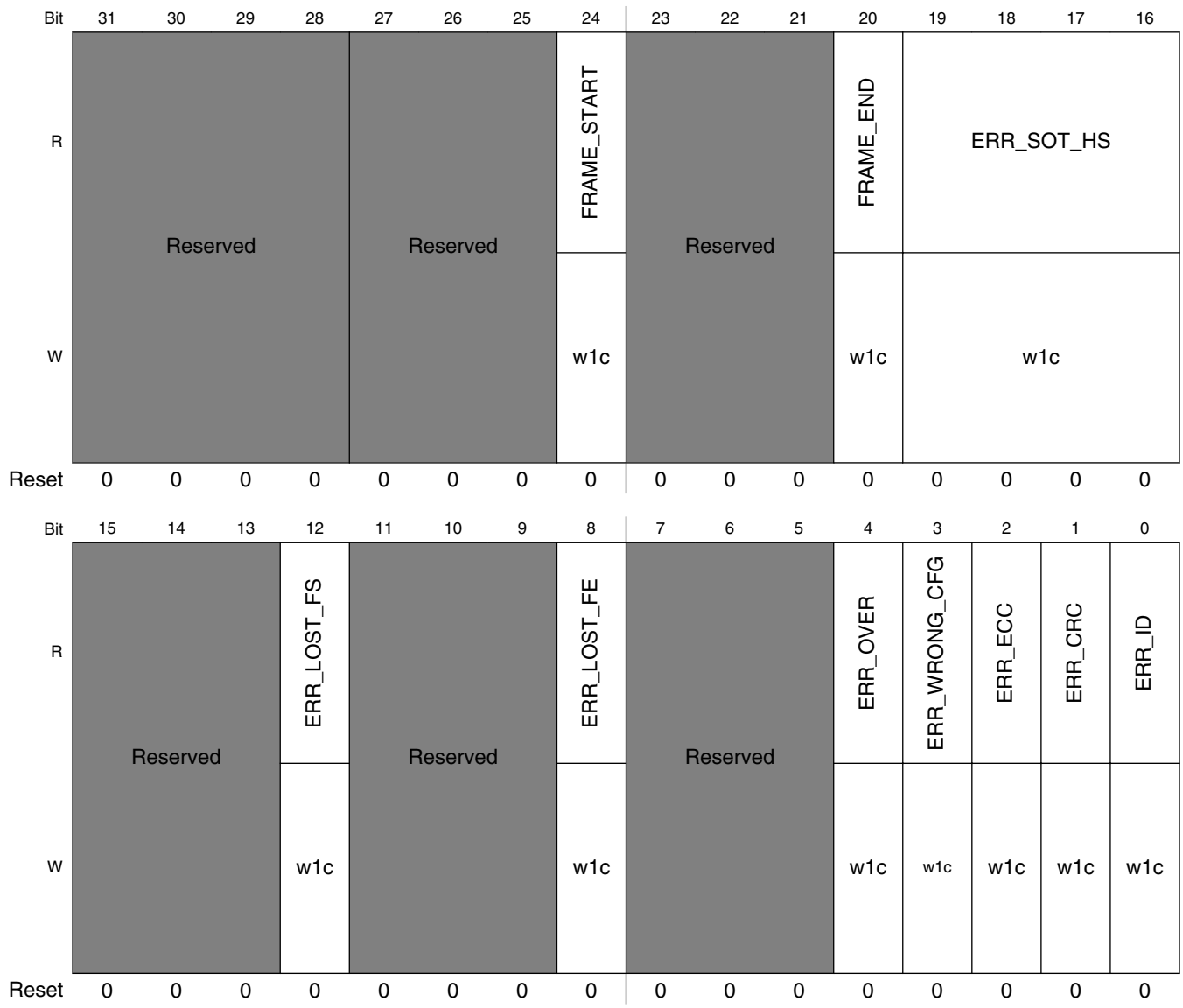
MIPI_CSIx_INTERRUPT_MASK_0 field descriptions (continued)

Field	Description
11–9 -	This field is reserved.
8 MSK_ERR_ LOST_FE	Lost of Frame End packet, CH0. 0 Disable (masked) 1 Enable (unmasked)
7–5 -	This field is reserved.
4 MSK_ERR_ OVER	Image FIFO overflow interrupt 0 Disable (masked) 1 Enable (unmasked)
3 MSK_ERR_ WRONG_CFG	Wrong configuration 0 Disable (masked) 1 Enable (unmasked)
2 MSK_ERR_ECC	ECC error 0 Disable (masked) 1 Enable (unmasked)
1 MSK_ERR_CRC	CRC error 0 Disable (masked) 1 Enable (unmasked)
0 MSK_ERR_ID	Unknown ID error 0 Disable (masked) 1 Enable (unmasked)

13.5.6.4 Interrupt source register 0 (MIPI_CSIx_INTERRUPT_SOURCE_0)

Write ‘1’ clears status bit and write ‘0’ has no effect. This register is valid only during csis is enabled. All of register fields should be cleared after csis is enabled.

Address: Base address + 14h offset



MIPI_CSIx_INTERRUPT_SOURCE_0 field descriptions

Field	Description
31–28 -	This field is reserved.

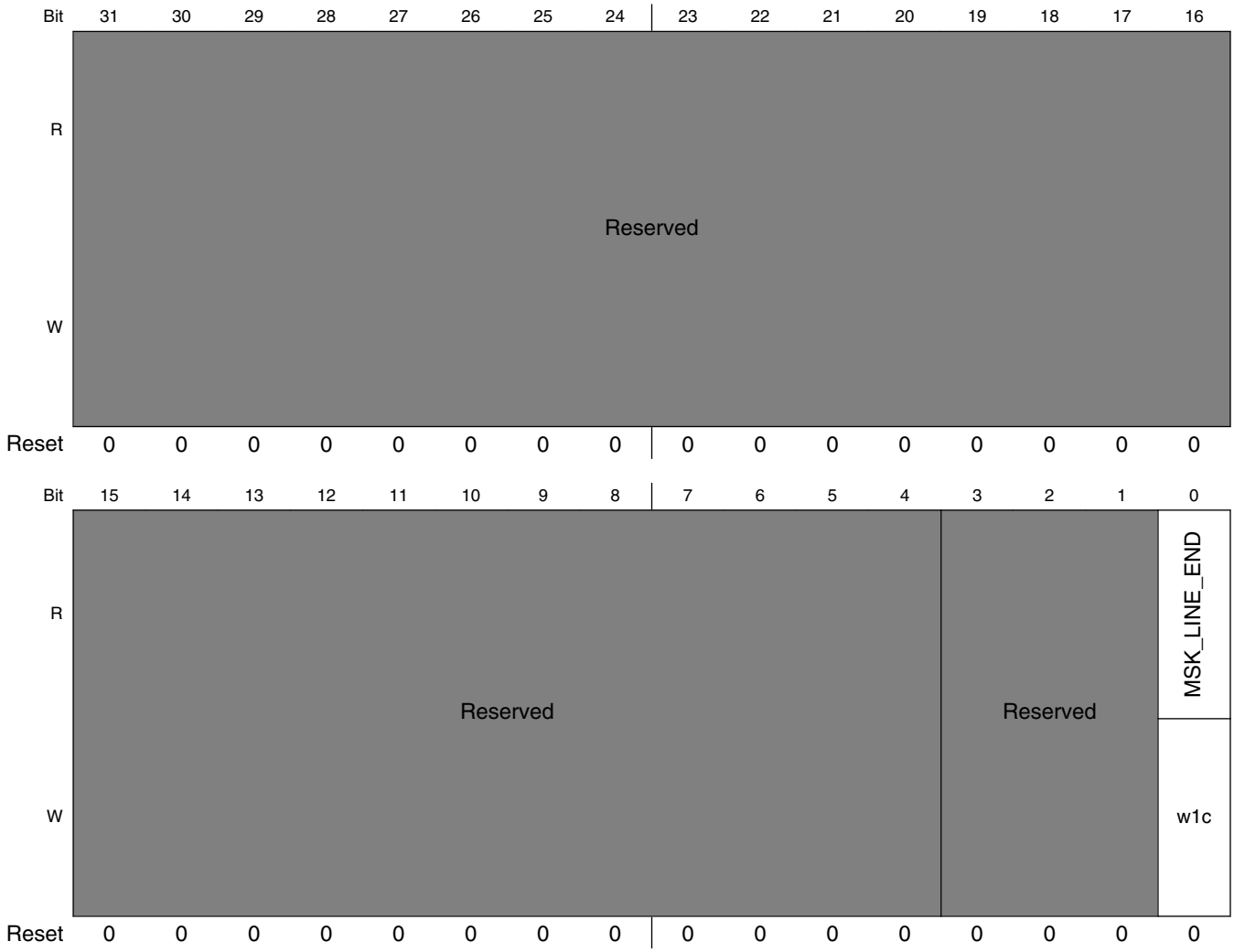
Table continues on the next page...

MIPI_CSIx_INTERRUPT_SOURCE_0 field descriptions (continued)

Field	Description
27–25 -	This field is reserved.
24 FRAME_START	FS packet is received, CH0.
23–21 -	This field is reserved.
20 FRAME_END	FE packet is received, CH0.
19–16 ERR_SOT_HS	Start of transmission error, [Lane3,Lane2,Lane1,Lane0]
15–13 -	This field is reserved.
12 ERR_LOST_FS	Indication of lost of Frame Start packet, CH0. This field is set to '1' when the image data is received without frame start packet.
11–9 -	This field is reserved.
8 ERR_LOST_FE	Indication of lost of Frame End packet, CH0.
7–5 -	This field is reserved.
4 ERR_OVER	Overflow is caused in image FIFO. Outer bandwidth has to be faster than imputer bandwidth. But image FIFO can be overflow because of user's fault. There are 2 ways for preventing overflow. • Tune output pixel clock faster than current. • Tune input byte clock slowr than current. First case : WCLK_Src in CSIS_CTRL register shoud be set 1, and then assign faster clock Second case : user can set register in camera module through I2C channel. When this interrupt is generated: • Turn camera off • Assert software reset, if you didn't assert software reset, MIPI CSIS could not receive any more data. • Tune the clock frequency and re-configure all related registers. MIPI CSIS module is ready for operating.
3 ERR_WRONG_CFG	Wrong configuration **NOTE:** The received packet is not allocated to the PHYsical channel at any PHYsical channels (due to wrong data type or virtual channel id).
2 ERR_ECC	ECC error
1 ERR_CRC	CRC error
0 ERR_ID	Unknown ID error

13.5.6.5 Interrupt mask register 1 (MIPI_CSIx_INTERRUPT_MASK_1)

Address: Base address + 18h offset



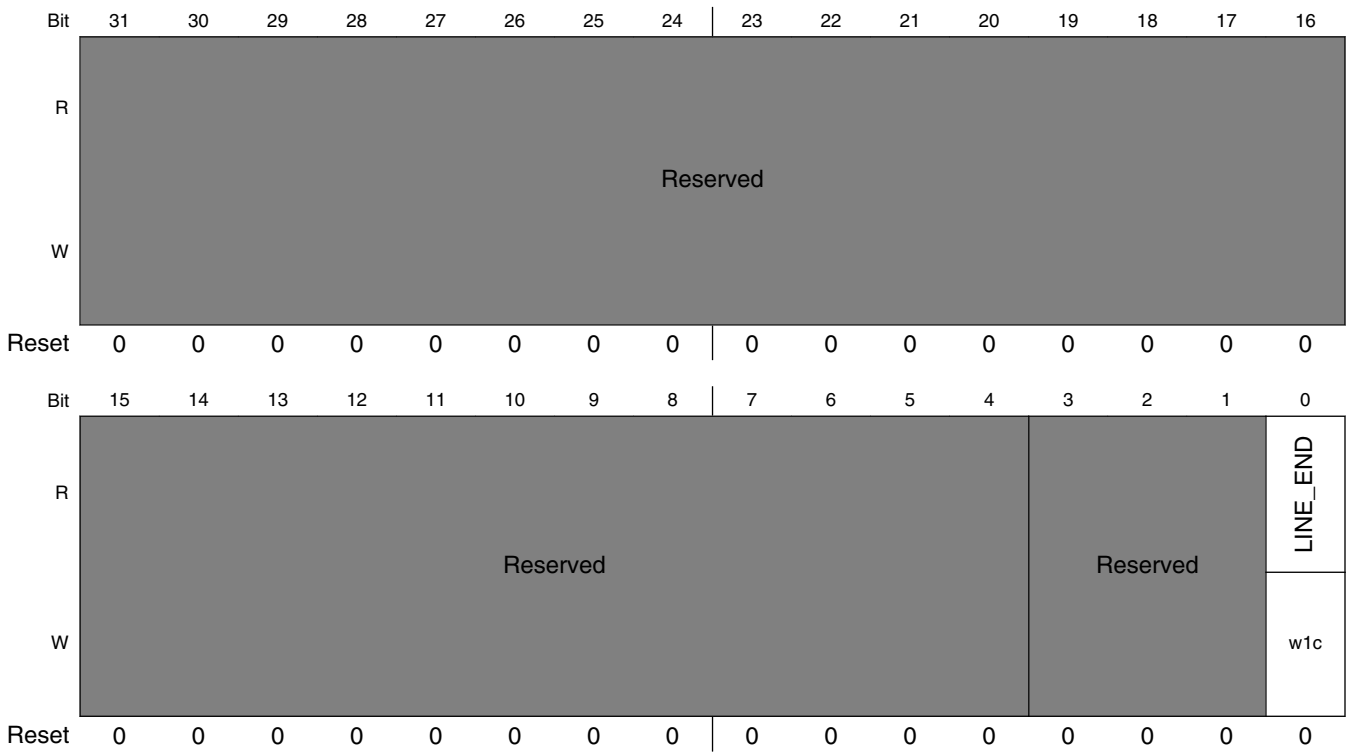
MIPI_CSIx_INTERRUPT_MASK_1 field descriptions

Field	Description
31–4 -	This field is reserved.
3–1 -	This field is reserved.
0 MSK_LINE_END	End of specific line, CH0.

13.5.6.6 Interrupt source register 1 (MIPI_CSIx_INTERRUPT_SOURCE_1)

Write ‘1’ clears status bit and write ‘0’ has no effect. This register is valid only during csis is enabled. All of register fields should be cleared after csis is enabled.

Address: Base address + 1Ch offset

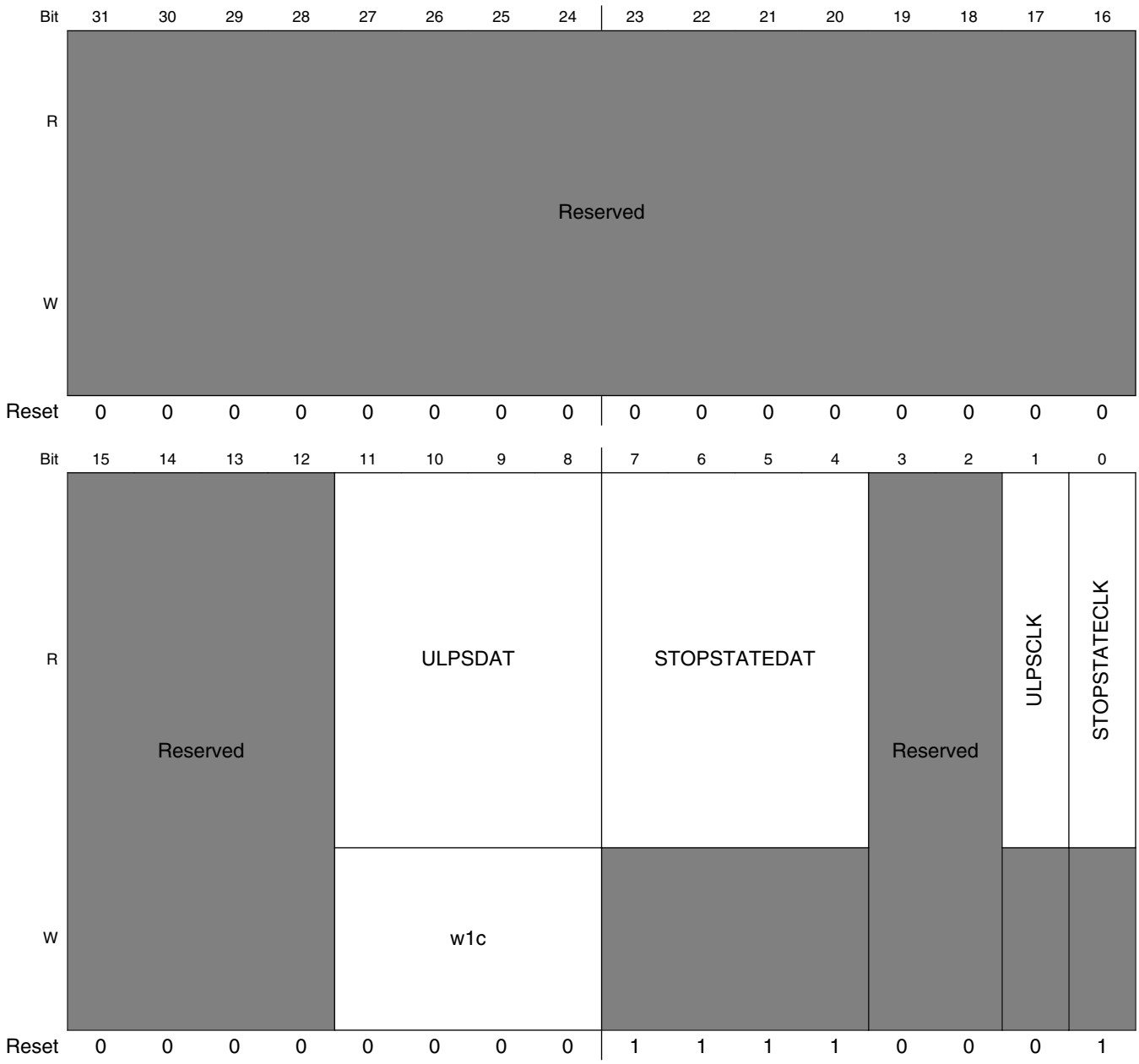


MIPI_CSIx_INTERRUPT_SOURCE_1 field descriptions

Field	Description
31–4 -	This field is reserved.
3–1 -	This field is reserved.
0 LINE_END	End of specific line, CH0.

13.5.6.7 D-PHY status register (MIPI_CSIx_DPHY_STATUS)

Address: Base address + 20h offset



MIPI_CSIx_DPHY_STATUS field descriptions

Field	Description
31–12 -	This field is reserved.
11–8 ULPSDAT	Data lane [3:0] is in ULPS [7] : data lane 3

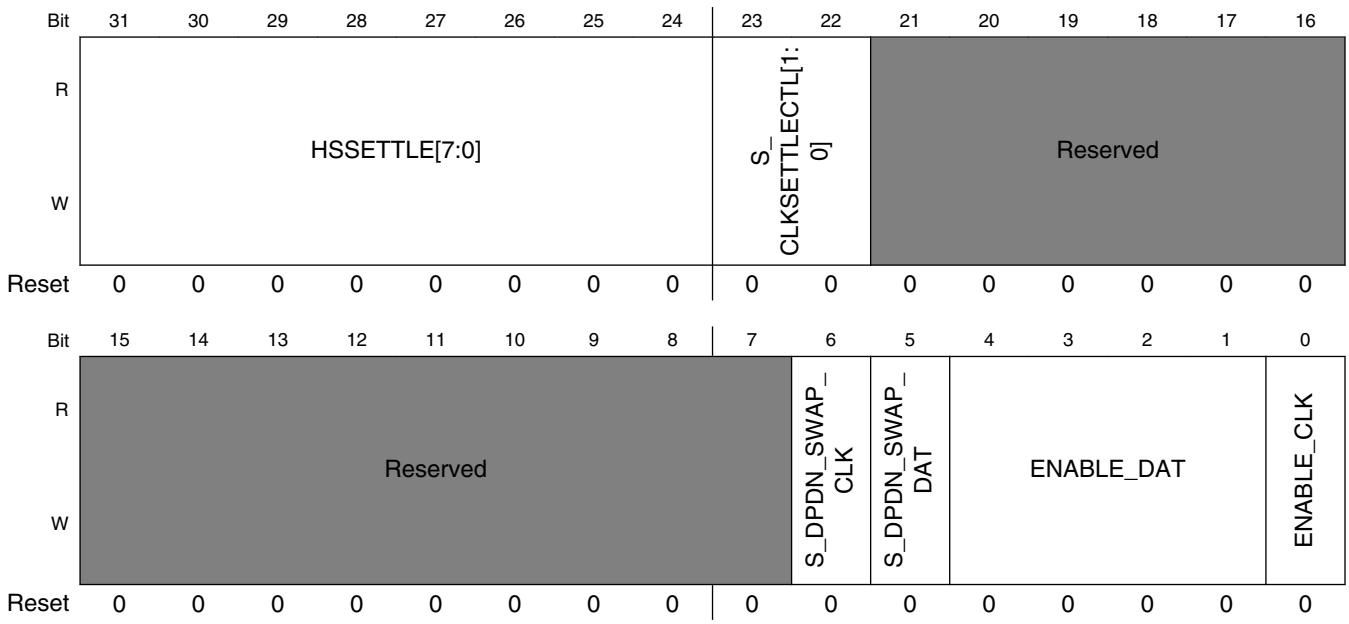
Table continues on the next page...

MIPI_CSIx_DPHY_STATUS field descriptions (continued)

Field	Description
	[6] : data lane 2 [5] : data lane 1 [4] : data lane 0 0 Not ULPS 1 ULPS
7–4 STOPSTATEDAT	Data lane [3:0] is in Stop State [7] : data lane 3 [6] : data lane 2 [5] : data lane 1 [4] : data lane 0 0 Not Stop state 1 Stop state
3–2 -	This field is reserved.
1 ULPSCLK	0 Not ULPS 1 ULPS
0 STOPSTATECLK	Clock lane is in Stop State 0 Not Stop state 1 Stop state

13.5.6.8 D-PHY common control register (MIPI_CSIx_DPHY_COMMON_CTRL)

Address: Base address + 24h offset



MIPI_CSIx_DPHY_COMMON_CTRL field descriptions

Field	Description
31–24 HSSETTLE[7:0]	HS-RX settle time control Slave Clock Lane Control for Ths-settle **NOTE:** Contact NXP support for more information about this field/module.
23–22 S_CLKSETTLECTL[1:0]	Slave clock lane control for Tclk-settle 2'b0x - 110ns to 280ns (v0.87 to v1.00) 2'b10 - 150 ns to 430ns (v0.83 to v0.86) 2'b11 - 60 ns to 140ns (v0.82)
21–7 -	This field is reserved.
6 S_DPDN_SWAP_CLK	Swapping Dp and Dn channel of clock lane. 0 Default 1 Swapped
5 S_DPDN_SWAP_DAT	Swapping Dp and Dn channel of data lanes. 0 Default 1 Swapped
4–1 ENABLE_DAT	D-PHY enable [3] : data lane 3

Table continues on the next page...

MIPI_CSIx_DPHY_COMMON_CTRL field descriptions (continued)

Field	Description
	[2] : data lane 2 [1] : data lane 1 [0] : data lane 0 0 Disable 1 Enable
0 ENABLE_CLK	D-PHY clock lane enable 0 Disable 1 Enable

13.5.6.9 D-PHY Master and Slave Control register Low (MIPI_CSIx_DPHY_MASTER_SLAVE_CTRL_LOW)

Depending on D-PHY, the bit-width of D-PHY control register is different. Since DPHY_1.5G_LN20LPE, it requires more than 32bits and DPHY_BCTRL_H (0x34) is usable.

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	USER_DATA_PATTERN_LOW		BIAS_REF_VOLT		BGR_CHOPPER_FREQ	VREG12_EXTPWR_EN_CTL	REG_1P2_LVL_CTL		REG_1P2_LVL_SEL	LP_RX_HYS_LVL		VREF_SRC_SEL	LP_RX_VREF_LVL		LP_RX_PULSE_REJECT	MSTRCLK_LP_SLEW_RATE_DOWN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSTRCLK_LP_SLEW_RATE_DOWN	MSTRCLK_LP_SLEW_RATE_UP	LP_CD_HYS	BGR_CHOPPER_EN	ERRCONTENTION_LP_EN	TXTRIGGER_CLK_EN	B_DPHYCTRL									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_CSIx_DPHY_MASTER_SLAVE_CTRL_LOW field descriptions

Field	Description
31–30 USER_DATA_ PATTERN_LOW	User Data Pattern for HS Loopback mode
29–28 BIAS_REF_VOLT	Bias Reference Voltage 710m Control 2'b00 = 715 mV 2'b01 = 724 mV 2'b10 = 733 mV 2'b11 = 706 mV
27 BGR_CHOPPER_ FREQ	BGR Chopper Frequency Control 1'b0 = 3 MHz 1'b1 = 1.5 MHz
26 VREG12_EXTPWR_ EN_CTL	VREG12_EXTPWR Enable Control (No regulator in JF) 1'b0 = Internal 1.2 V regulator 1'b1 = External 1.2 V power
25–24 REG_1P2_LVL_CTL	1.2V Regulator Level Control (No regulator in JF) 2'b00 = 1.2V (Default) 2'b01 = 1.23V 2'b10 = 1.17V 2'b11 = 1.26V
23 REG_1P2_LVL_SEL	1.2V Regulator Valid Level Selection 1'b0 = Use "ulps_en" signal 1'b1 = Use valid signal from 1.2 V regulator
22–21 LP_RX_HYS_LVL	LP-RX Hysteresis Level Control 2'b00 = 80 mV (Default) 2'b01 = 100 mV 2'b10 = 120 mV 2'b11 = 140 mV
20 VREF_SRC_SEL	Vref Source Selection 1'b0 = Generated from the BGR (Default) 1'b1 = Generated from the Current Mirror
19–18 LP_RX_VREF_LVL	LP-RX Vref Level Control 2'b00 = 715 mV (Default) 2'b01 = 743 mV 2'b10 = 650 mV 2'b11 = 682 mV
17 LP_RX_PULSE_ REJECT	LP-RX Pulse Rejection Control 1'b0 = Enable (Default)

Table continues on the next page...

MIPI_CSIx_DPHY_MASTER_SLAVE_CTRL_LOW field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>1'b1 = Disable</td></tr>
<tr><td>16–15
MSTRCLK_LP_
SLEW_RATE_DOWN</td><td>Master Clock Lane's LP-TX Driver Slew Rate Down Control
2'b00 = No Change (Default)
2'b01 = Decrease the slew rate about 15 %
2'b10 = Decrease the slew rate about 15 %
2'b11 = Decrease the slew rate about 30 %</td></tr>
<tr><td>14
MSTRCLK_LP_
SLEW_RATE_UP</td><td>Master Clock Lane's LP-TX Driver Slew Rate Up Control
1'b0 = No change (Default)
1'b1 = Slew Rate UP</td></tr>
<tr><td>13
LP_CD_HYS</td><td>LP-CD Hysteresis Level Control
1'b0 = 60mV (Default)
1'b1 = 70mV</td></tr>
<tr><td>12
BGR_CHOPPER_EN</td><td>BGR Chopper Function Enable
1'b0 = Enable (Default)
1'b1 = Disable</td></tr>
<tr><td>11
ERRCONTENTION_
LP_EN</td><td>ErrContention LP Enable
1'b0 = Enable (Default)
1'b1 = Disable</td></tr>
<tr><td>10
TXTRIGGER_CLK_
EN</td><td>TxTrigger_Clk Enable
1'b0 = Enable (Default)
1'b1 = Disable</td></tr>
<tr><td>B_DPHYCTRL</td><td>ULPS EXIT Counter Value Control. You should set B_DPHYCTL[9:0] during initial or power-up sequence

<table>
<tr><th>M_TXCLKESC Frequency</th><th>B_DPHYCTL[9:0] Value in HEX</th><th>B_DPHYCTL[9:0] Value in BIN</th></tr>
<tr><td>20.00 MHz</td><td>10'h 1F4</td><td>10'b 0111110100</td></tr>
<tr><td>19.00 MHz</td><td>10'h 1DB</td><td>10'b 0111011011</td></tr>
<tr><td>18.00 MHz</td><td>10'h 1C2</td><td>10'b 0111000010</td></tr>
<tr><td>17.00 MHz</td><td>10'h 1A9</td><td>10'b 0110101001</td></tr>
<tr><td>16.00 MHz</td><td>10'h 190</td><td>10'b 0110010000</td></tr>
<tr><td>15.00 MHz</td><td>10'h 177</td><td>10'b 0101110111</td></tr>
<tr><td>14.00 MHz</td><td>10'h 15E</td><td>10'b 0101011110</td></tr>
<tr><td>13.00 MHz</td><td>10'h 145</td><td>10'b 0101000101</td></tr>
<tr><td>12.00 MHz</td><td>10'h 12C</td><td>10'b 0100101100</td></tr>
<tr><td>11.00 MHz</td><td>10'h 113</td><td>10'b 0100010011</td></tr>
<tr><td>10.00 MHz</td><td>10'h 0FA</td><td>10'b 0011111010</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

MIPI_CSIx_DPHY_MASTER_SLAVE_CTRL_LOW field descriptions (continued)

Field	Description		
	9.00 MHz	10'h 0E1	10'b 0011100001
	8.00 MHz	10'h 0C8	10'b 0011001000
	7.00 MHz	10'h 0AF	10'b 0010101111
	6.00 MHz	10'h 096	10'b 0010010110
	5.00 MHz	10'h 07D	10'b 0001111101
	4.00 MHz	10'h 064	10'b 0001100100
	3.00 MHz	10'h 04B	10'b 0001001011
	2.00 MHz	10'h 032	10'b 0000110010
	1.00 MHz	10'h 019	10'b 0000011001
	0.10 MHz	10'h 003	10'b 0000000011
	0.01 MHz	10'h 000	10'b 0000000000

13.5.6.10 D-PHY Master and Slave Control register HIGH (MIPI_CSIx_DPHY_MASTER_SLAVE_CTRL_HIGH)

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MST_DATA3_TX_SLEW_DOWN		MST_DATA3_TX_SLEW_UP	MST_DATA2_TX_SLEW_DOWN		MST_DATA2_TX_SLEW_UP	MST_DATA1_TX_SLEW_DOWN		MST_DATA1_TX_SLEW_UP	MST_DATA0_TX_SLEW_DOWN		MST_DATA0_TX_SLEW_UP	Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	LP_REG_VREF_SRC_SEL	RX_SKEW_CALIB_FIX_EN	Reserved	DCC_DONE	-	BGR_VOLT_TUNE		HS_LOOPBACK_MODE_CTL	USER_DATA_PATTERN_HIGH						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_CSIx_DPHY_MASTER_SLAVE_CTRL_HIGH field descriptions

Field	Description
31–30 MST_DATA3_TX_SLEW_DOWN	Master Data3 Lane's LP-TX Driver Slew Rate Down Control 2'b00 = No Change (Default) 2'b01 = Decrease the slew rate about 15 % 2'b10 = Decrease the slew rate about 15 % 2'b11 = Decrease the slew rate about 30 %
29 MST_DATA3_TX_SLEW_UP	Master Data3 Lane's LP-TX Driver Slew Rate Up Control 1'b0 = No Change (Default) 1'b1 = Slew Rata UP
28–27 MST_DATA2_TX_SLEW_DOWN	Master Data2 Lane's LP-TX Driver Slew Rate Down Control 2'b00 = No Change (Default) 2'b01 = Decrease the slew rate about 15 % 2'b10 = Decrease the slew rate about 15 % 2'b11 = Decrease the slew rate about 30 %
26 MST_DATA2_TX_SLEW_UP	Master Data2 Lane's LP-TX Driver Slew Rate Up Control 1'b0 = No Change (Default) 1'b1 = Slew Rata UP
25–24 MST_DATA1_TX_SLEW_DOWN	Master Data1 Lane's LP-TX Driver Slew Rate Down Control 2'b00 = No Change (Default) 2'b01 = Decrease the slew rate about 15 % 2'b10 = Decrease the slew rate about 15 % 2'b11 = Decrease the slew rate about 30 %
23 MST_DATA1_TX_SLEW_UP	Master Data1 Lane's LP-TX Driver Slew Rate Up Control 1'b0 = No Change (Default) 1'b1 = Slew Rata UP
22–21 MST_DATA0_TX_SLEW_DOWN	Master Data0 Lane's LP-TX Driver Slew Rate Down Control 2'b00 = No Change (Default) 2'b01 = Decrease the slew rate about 15 % 2'b10 = Decrease the slew rate about 15 % 2'b11 = Decrease the slew rate about 30 %
20 MST_DATA0_TX_SLEW_UP	Master Data0 Lane's LP-TX Driver Slew Rate Up Control
19–15 -	Reserved This field is reserved.
14 LP_REG_VREF_SRC_SEL	LP Regulator Vref Source Selection (No regulator in JF) 1'b0 = Generated from the BGR (Default) 1'b1 = Generated from the Current Mirror

Table continues on the next page...

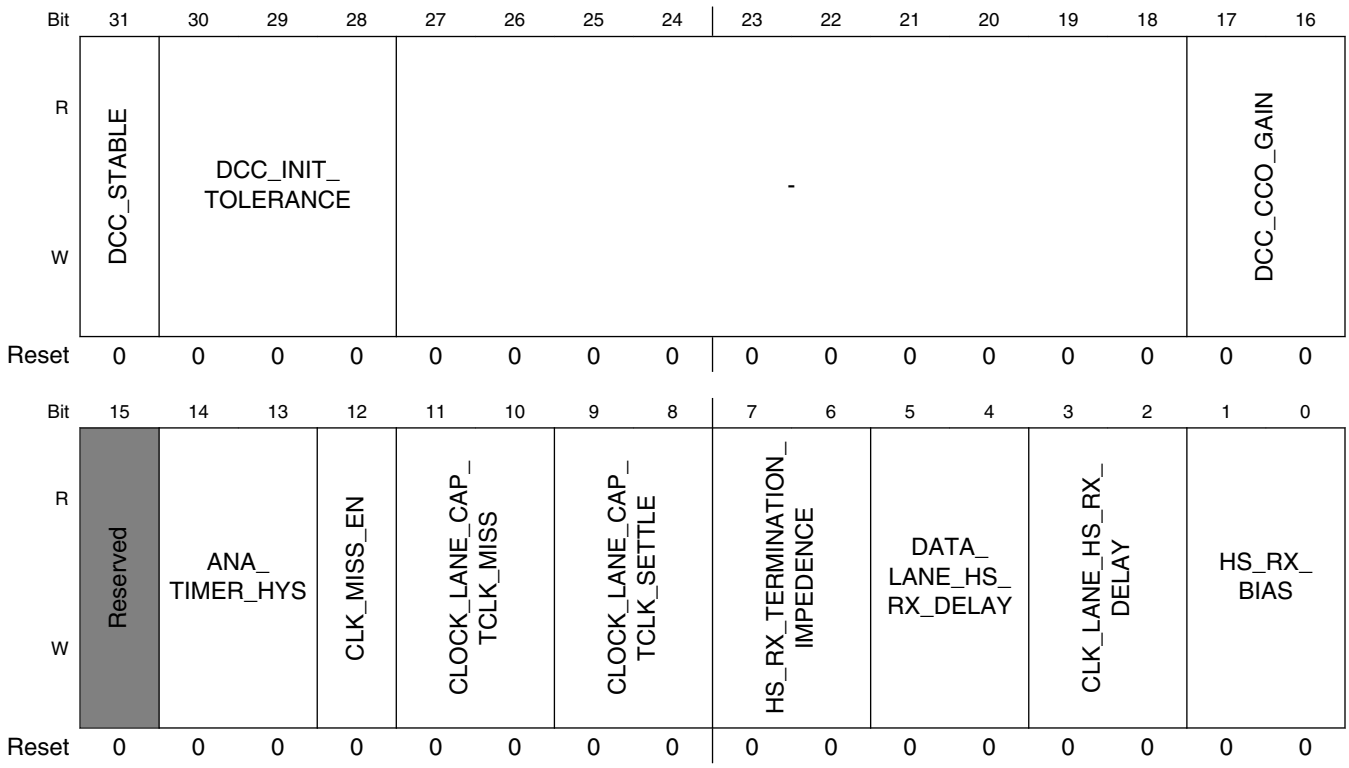
MIPI_CSIx_DPHY_MASTER_SLAVE_CTRL_HIGH field descriptions (continued)

Field	Description
13 RX_SKEW_ CALIB_FIX_EN	RX Skew Calibration Fixing Code Enable/Disable Control 1'b0 = Disable 1'b1 = Enable
12 -	Reserved This field is reserved.
11 DCC_DONE	DCC "DONE" Signal Control 1'b0 = "DONE" from DCC block 1'b1 = "DONE" is always 1.
10 -	Reserved
9–8 BGR_VOLT_ TUNE	BGR voltage tuning control 2'00 = 820mV(Default) 2'01 = 760mV 2'10 = 800mV 2'11 = 840mV
7–6 HS_ LOOPBACK_ MODE_CTL	HS Loopback Mode Control 2'b00 = Designated Pattern 2'b01 = PRBS7 2'b10 = All zero 2'b11 = User Data Pattern
USER_DATA_ PATTERN_HIGH	User Data Pattern for HS Loopback mode

13.5.6.11 D-PHY Slave Control register Low (MIPI_CSIx_DPHY_SLAVE_CTRL_LOW)

Depending on D-PHY, the bit-width of D-PHY control register is different. Since DPHY_1.5G_LN20LPE, it requires more than 32bits and DPHY_SCTRL0_H(0x3C) is usable

Address: Base address + 38h offset



MIPI_CSIx_DPHY_SLAVE_CTRL_LOW field descriptions

Field	Description
31 DCC_STABLE	DCC Stable Control 1'b0 = 1 number counter running (Default) 1'b1 = 2 number counter running
30–28 DCC_INIT_ TOLERANCE	DCC Initial Tolerance 3'b000 = 4 (Default) 3'b001 = 5 3'b010 = 6 3'b011 = 7 3'b100 = 0 3'b101 = 1

Table continues on the next page...

MIPI_CSIx_DPHY_SLAVE_CTRL_LOW field descriptions (continued)

Field	Description
	3'b110 = 2 3'b111 = 3
27–18 -	Reserved
17–16 DCC_CCO_GAIN	DCC CCO Gain Control 2'b00 = 1/1(Default) 2'b01 = 1/4 2'b10 = 1/0.66 2'b11 = 1/1.33
15 -	Reserved This field is reserved.
14–13 ANA_TIMER_HYS	Analog Timer Hysteresis Control 2'b00 = 70 mV (Default) 2'b01 = 95 mV 2'b10 = 95 mV 2'b11 = 110 mV
12 CLK_MISS_EN	Clock Miss Function Enable/Disable Control 1'b0 = Enable (Default) 1'b1 = Disable
11–10 CLOCK_LANE_CAP_TCLK_MISS	Clock Lane Cap. Value Control for Tclk_miss 2'b00 = No Change (Default) 2'b01 = 6.6 % 2'b10 = 13.2 % 2'b11 = 19.8 %
9–8 CLOCK_LANE_CAP_TCLK_SETTLE	Clock Lane Cap. Value Control for Tclk-settle 2'b00 = No Change (Default) 2'b01 = 3.45 % 2'b10 = 6.9 % 2'b11 = 10.35 %
7–6 HS_RX_TERMINATION_IMPEDENCE	HS-RX Termination Impedance Control 2'b00 = 98 Ohm (Default) 2'b01 = 106 Ohm 2'b10 = 85 Ohm 2'b11 = 91 Ohm
5–4 DATA_LANE_HS_RX_DELAY	Data Lane HS RX Delay Control 2'b00 = 0 ps (Default) 2'b01 = 30 ps

Table continues on the next page...

MIPI_CSIx_DPHY_SLAVE_CTRL_LOW field descriptions (continued)

Field	Description
	2'b10 = 60 ps 2'b11 = 90 ps
3–2 CLK_LANE_HS_RX_DELAY	Clock Lane HS RX Delay Control 2'b00 = 0 ps (Default) 2'b01 = 30 ps 2'b10 = 60 ps 2'b11 = 90 ps
HS_RX_BIAS	HS RX Bias Control 2'b00 = 25 μA (Default) 2'b01 = 30 μA 2'b10 = 37.5 μA 2'b11 = 50 uμA

13.5.6.12 D-PHY Slave Control register HIGH (MIPI_CSIx_DPHY_SLAVE_CTRL_HIGH)

Address: Base address + 3Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RX_SKEW_CALIB_COMPARE_RUN_TIME				RX_SKEW_CALIB_COMPARE_WAIT_TIME		RX_SKEW_CALIB_FAIL_TOLERANCE		Reserved		RX_SKEW_CALIB_PASS_MIN					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		RX_SKEW_CALIB_FAIL_MIN						RX_SKEW_CALIB_MAX						SKEW_CALIB_EN	DCC_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_CSIx_DPHY_SLAVE_CTRL_HIGH field descriptions

Field	Description
31–28 RX_SKEW_CALIB_COMPARE_RUN_TIME	RX Skew Calibration Compare-run time Control Control the number of successive pass condition

Table continues on the next page...

MIPI_CSIx_DPHY_SLAVE_CTRL_HIGH field descriptions (continued)

Field	Description
27–26 RX_SKEW_CALIB_COMPARE_WAIT_TIME	RX Skew Calibration Compare-wait time Control Control waiting time at changing delay code 2'b00 = min (fast) 2'b11 = max (slow)
25–24 RX_SKEW_CALIB_FAIL_TOLERANCE	RX Skew Calibration Fail-tolerance Control 2'b00 = Recognizes the pass although fail appears 3 times 2'b01 = Recognizes the pass although fail appears 2 times. 2'b10 = Recognizes the pass although fail appears 1 time
23–22 -	Reserved This field is reserved.
21–16 RX_SKEW_CALIB_PASS_MIN	RX Skew Calibration Pass-min Control
15–14 -	Reserved This field is reserved.
13–8 RX_SKEW_CALIB_FAIL_MIN	RX Skew Calibration Fail-min Control
7–2 RX_SKEW_CALIB_MAX	RX Skew Calibration Max Code Control
1 SKEW_CALIB_EN	Skew Calibration Function Enable/Disable Control 1'b0 = Skew Calibration Disable 1'b1 = Skew Calibration Enable
0 DCC_EN	DCC Function Enable/Disable Control 1'b0 = DCC Disable 1'b1 = DCC Enable

13.5.6.13 ISP Configuration Register (MIPI_CSIx_ISP_CONFIG*n*)

Address: Base address + 40h offset + (16d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		PIXEL_MODE		PARALLEL	RGB_SWAP	Reserved		DATAFORMAT[5:0]						Reserved	
Reset	0	0	0	0	1	0	0	0	1	1	1	1	1	1	0	0

MIPI_CSIx_ISP_CONFIG*n* field descriptions

Field	Description
31–14 -	This field is reserved.
13–12 PIXEL_MODE	Pixel mode selection, 'b11 : Invalid 'b10 : Quad pixel mode (RAW8/10/12) 'b01 : Dual pixel mode (RAW8/10/12, YUV422) 'b00 : Single pixel mode
11 PARALLEL	Output bus width of CH0 is 32 bits. When this bit is set, the outer bus width of CSIS V3.3 is 32. Refer to figures in chapter 12 of MIPI CSI-2 standard spec. 0 Normal output 1 32bit data alignment
10 RGB_SWAP	Swapping RGB sequence 0 MSB is R and LSB is B 1 MSB is B and LSB is R (swapped)
9–8 -	This field is reserved.
7–2 DATAFORMAT[5:0]	Image Data Format 0x18 - YUV420 (8bit) 0x19 - YUV420 (10bit) 0x1A - YUV420 (8bit legacy)

Table continues on the next page...

MIPI_CSIx_ISP_CONFIG*n* field descriptions (continued)

Field	Description
	0x1C - YUV420 (8bit CSPS) 0x1D - YUV420 (10bit CSPS) 0x1E - YUV422 (8bit) 0x1F - YUV422 (10bit) 0x20 - (NOT SUPPORTED) RGB444 0x21 - (NOT SUPPORTED) RGB555 0x22 - RGB565 0x23 - RGB666 0x24 - RGB888 0x28 - RAW6 0x29 - RAW7 0x2A - RAW8 0x2B - RAW10 0x2C - RAW12 0x2D - RAW14 **NOTE:** Not described types are ignored.
-	This field is reserved.

13.5.6.14 ISP Resolution Register (MIPI_CSIx_ISP_RESOLUTION*n*)

Address: Base address + 44h offset + (16d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VRESOL																HRESOL															
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_CSIx_ISP_RESOLUTION*n* field descriptions

Field	Description
31–16 VRESOL	Vertical Image resolution Input boundary : 0x0001 ~ 0xFFFF
HRESOL	Horizontal Image resolution Input boundary of each image format. Hresol value must comply with the terms of the standard spec and it must be a multiple of four. Input boundary : 0x0004 ~ 0xFFFC

13.5.6.15 ISP SYNC Register (MIPI_CSIx_ISP_SYNC*n*)

Address: Base address + 48h offset + (16d × i), where i=0d to 0d

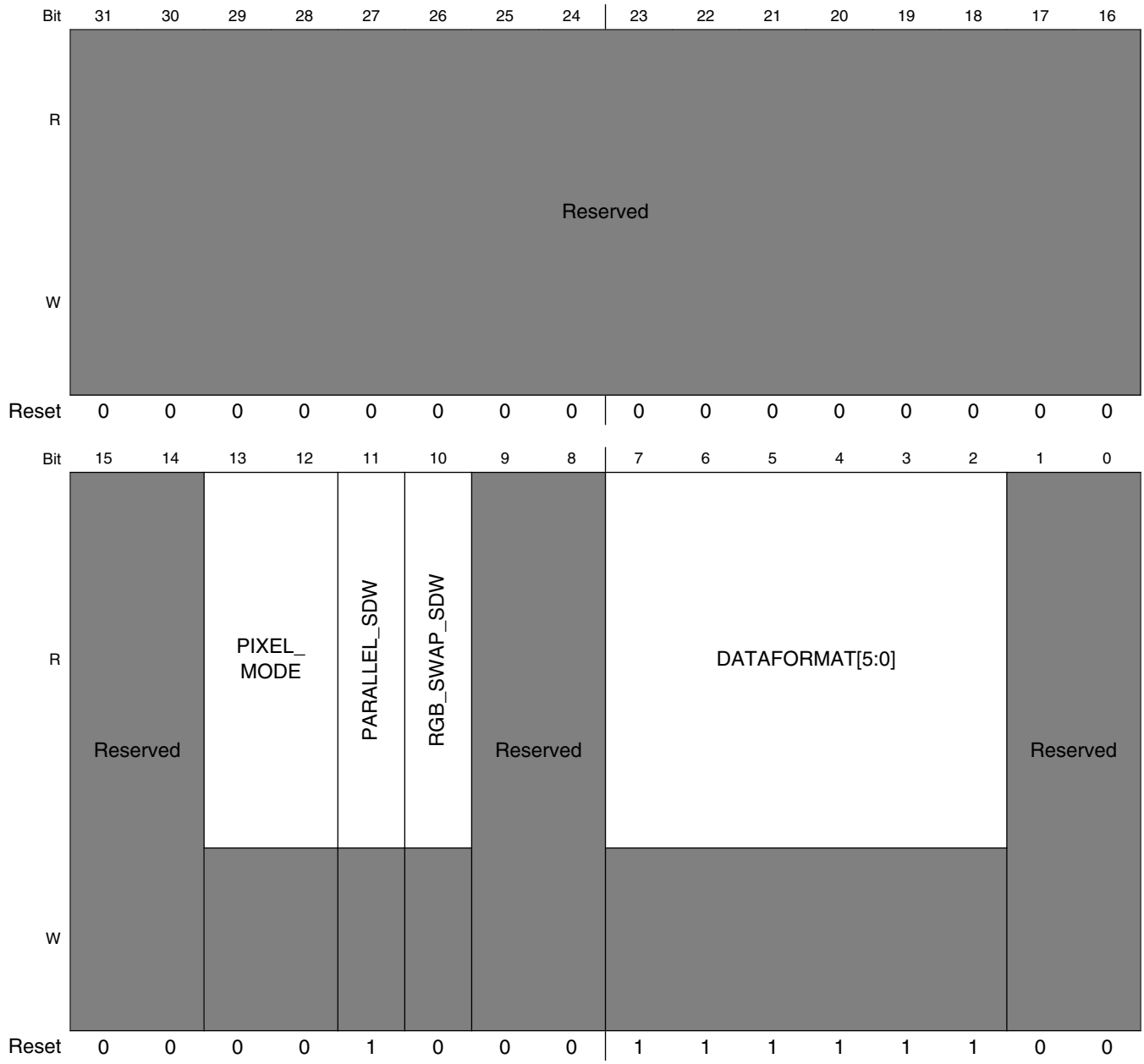
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								HSYNC_LINTV								Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_CSIx_ISP_SYNC*n* field descriptions

Field	Description
31–24 -	This field is reserved.
23–18 HSYNC_LINTV	Interval between last falling of DVALID and falling of HSync. In Figure 2-12, t2 is this. 6'h00 ~ 6'h3F cycle of Pixel clock
-	This field is reserved.

13.5.6.16 Shadow Configuration Register (MIPI_CSIx_SHADOW_CONFIG*n*)

Address: Base address + 80h offset + (16d × i), where i=0d to 0d



MIPI_CSIx_SHADOW_CONFIG*n* field descriptions

Field	Description
31–14 -	This field is reserved.

Table continues on the next page...

MIPI_CSIx_SHADOW_CONFIG*n* field descriptions (continued)

Field	Description
13–12 PIXEL_MODE	Current pixel_mode
11 PARALLEL_SDW	Current Parallel
10 RGB_SWAP_SDW	Current RGB_SWAP
9–8 -	This field is reserved.
7–2 DATAFORMAT[5:0]	Current Image Data Format
-	This field is reserved.

13.5.6.17 Shadow Resolution Register (MIPI_CSIx_SHADOW_RESOLUTION*n*)

Address: Base address + 84h offset + (16d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VRESOL_SDW																HRESOL_SDW															
W																																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_CSIx_SHADOW_RESOLUTION*n* field descriptions

Field	Description
31–16 VRESOL_SDW	Current Vertical Image resolution
HRESOL_SDW	Current Horizontal Image resolution

13.5.6.18 Shadow SYNC Register (MIPI_CSIx_SHADOW_SYNC*n*)

Address: Base address + 88h offset + (16d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								HSYNC_LINTV_SDW								Reserved															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_CSIx_SHADOW_SYNC*n* field descriptions

Field	Description
31–24 -	This field is reserved.
23–18 HSYNC_LINTV_SDW	Current interval between Hsync falling and Hsync rising (Line interval)
-	This field is reserved.

13.5.6.19 Frame Counter (MIPI_CSIx_FRAME_COUNTER*n*)

Address: Base address + 100h offset + (4d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FRM_CNT																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_CSIx_FRAME_COUNTER*n* field descriptions

Field	Description
FRM_CNT	Number of input frames

13.5.6.20 Line Interrupt Ratio (MIPI_CSIx_LINE_INTERRUPT_RATIO*n*)

Address: Base address + 110h offset + (4d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LINE_INTR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_CSIx_LINE_INTERRUPT_RATIO*n* field descriptions

Field	Description
LINE_INTR	The line number that user wants to set the line_end interrupt.

13.6 MIPI DSI Host Controller (MIPI_DSI)

13.6.1 Overview

MIPI Display Serial Interface (DSI) is a flexible, high-performance core that allows communication with MIPI DSI compliant peripherals.

13.6.1.1 Block Diagram

13.6.1.1.1 Total system block diagram

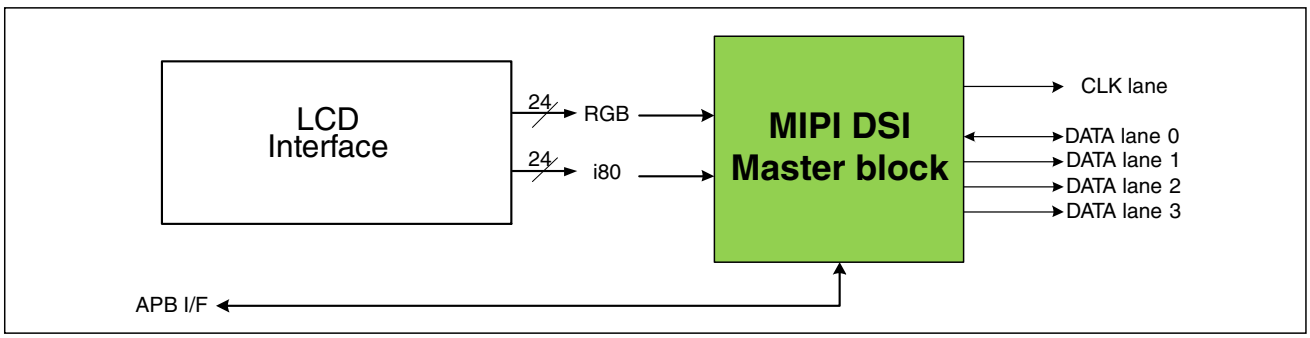


Figure 13-29. MIPI DSI master system block diagram

13.6.1.2 Features

The key features of MIPI DSI include:

- Complies to MIPI DSI Standard Specification V1.01r11
  - Maximum resolution ranges up to WQHD (2560x1440)
    - It should be decided on bandwidth between input clock (video clock) and output clock (D-PHY HS clock).
  - Supports 1, 2, 3, or 4 data lanes
  - Supports pixel format: 16bpp, 18bpp packed, 18bpp loosely packed (3 byte format), and 24bpp
- Interfaces
  - Complies with Protocol-to-PHY Interface (PPI) in 1.0Gbps / 1.5Gbps MIPI D-PHY
  - Supports RGB Interface for Video Image Input from general display controller
  - Supports S-i80(Synchronous i80) Interface for Command Mode Image input from display controller
  - Supports PMS control interface for PLL to configure byte clock frequency
  - Supports Prescaler to generate escape clock from byte clock

NOTE

Virtual Channel support is not implemented in chip level.

13.6.2 Functional description

The following sections describe functional details of the MIPI DSI module.

13.6.2.1 MIPI DSI master and D-PHY I/F block diagram

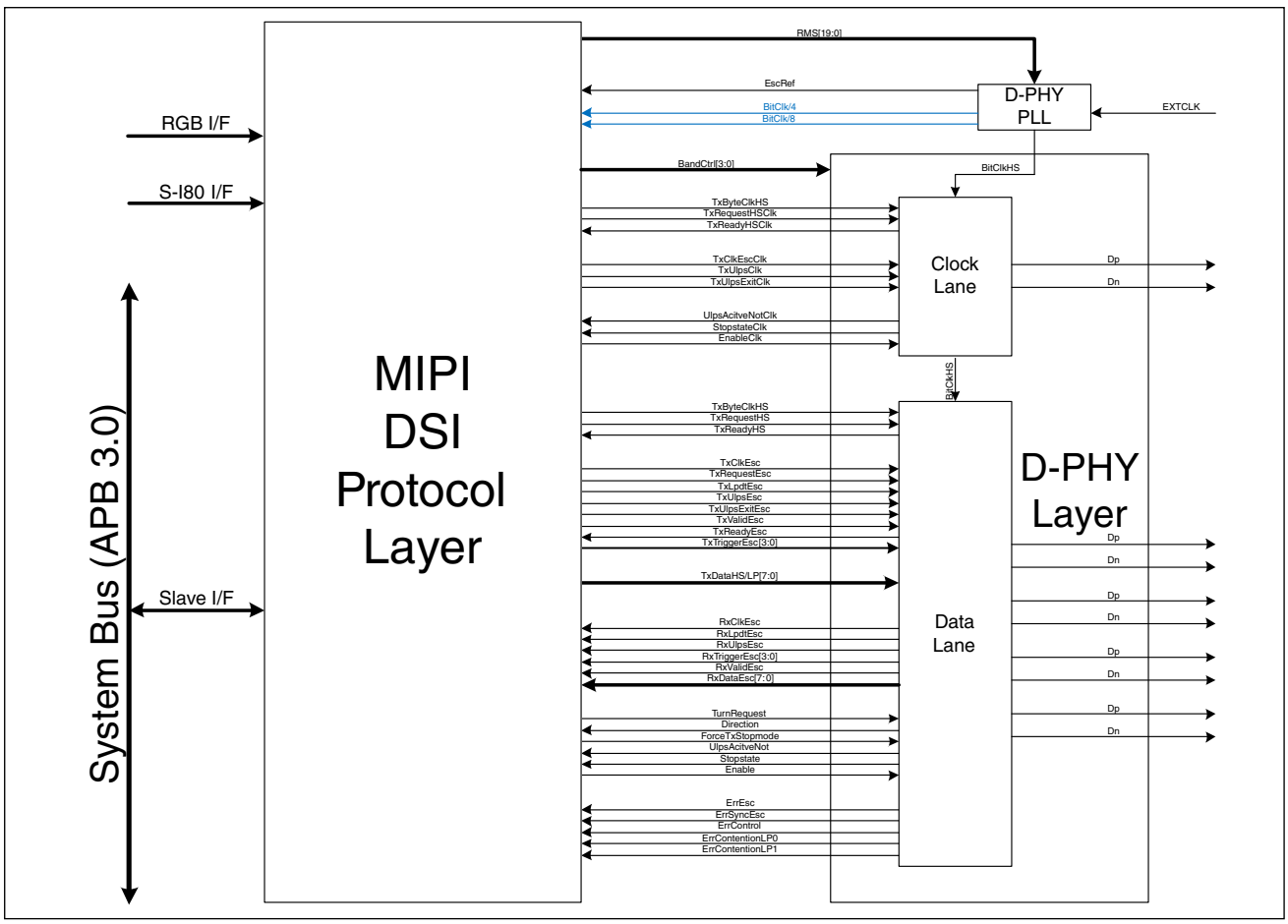


Figure 13-30. PPI I/F block diagram

13.6.2.2 MIPI DSI master block diagram

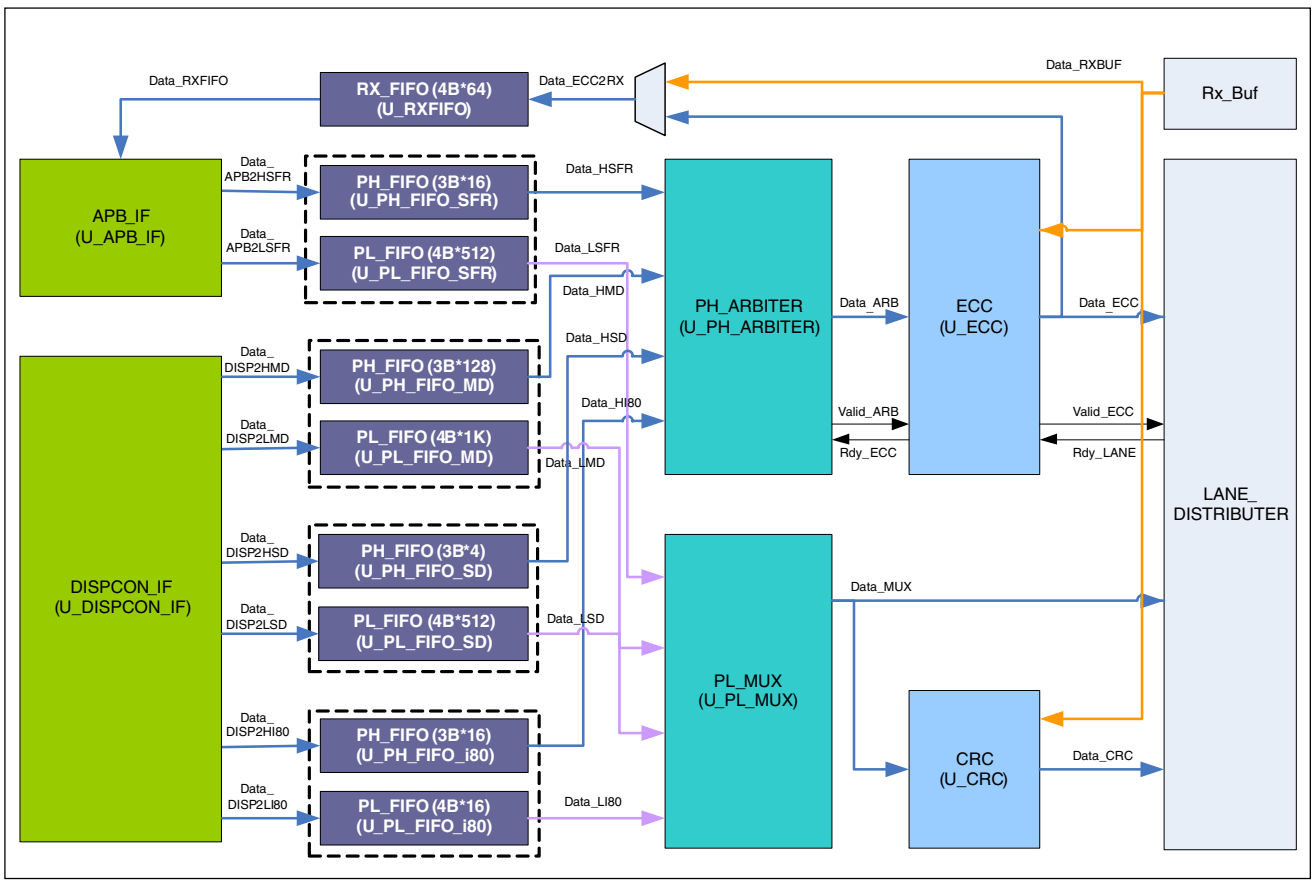


Figure 13-31. MIPI DSI master module block diagram

13.6.2.3 Internal primary FIFOs

Table 13-21 describes configurable-sized primary FIFOs.

Table 13-21. Internal primary FIFO list

Port	FIFO type	Size	Description
Main display for RGB I/F and S-i80 I/F	Packet Header FIFO	3Byte X 128 depth	Packet header FIFO for main display
	Payload FIFO	4Byte X 1024 depth	Payload FIFO for main display image (SRAM used)
Sub display for S-i80 I/F image data	Packet Header FIFO	3Byte X 4 depth	Packet header FIFO for S-i80 I/F sub display
	Payload FIFO	4Byte X 512 depth	Payload FIFO for S-i80 I/F sub display image (SRAM used)
Command for S-i80 I/F command	Packet Header FIFO	3Byte X 16 depth	Packet header FIFO for S-i80 I/F command packet

Table continues on the next page...

Table 13-21. Internal primary FIFO list (continued)

Port	FIFO type	Size	Description
	Payload FIFO	4Byte X 16 depth	Payload FIFO for S-i80 I/F command long packet payload
SFR for general packets	Packet Header FIFO	3Byte X 16 depth	Packet header FIFO for general packet
	Payload FIFO	4Byte X 512 depth	Payload FIFO for general long packet (SRAM used)
RX FIFO	Packet header and Payload FIFO	4Byte X 64 depth	Rx FIFO for LPDR. This FIFO is common for packet header and payload

13.6.2.4 Packet header arbitration

There are four-packet headers FIFOs for Tx, namely, main display, sub display, S-i80 INTERFACE command, and SFR FIFO. The main and sub display FIFO packet headers contain the image data, while the S-i80 INTERFACE command FIFO packet header contains the command packets. On the other hand, the SFR FIFO packet header contains command packets, sub display image data (in Video mode), and so on.

The packet header arbiter has a “Fixed priority” algorithm. Priority order is fixed as main display, sub display, S-i80 INTERFACE command, and SFR FIFO packet header.

In the Video mode, sub display, and S-i80 INTERFACE command FIFO are not used. The SFR FIFO packet header checks if the main display FIFO is empty (no request) in not-active image region and then sends its request.

13.6.2.5 RxFIFO Structure

To read the packets received via low power data receiving mode, RxFIFO acts like an SFR. RxFIFO is an asynchronous FIFO with ByteClk and PCLK domains as input clock and output clock domains respectively. The Rx data is synchronized to RxClk. RXBUF has four Rx Byte buffers for aligning byte to word.

The packet headers of all the packets stored in RXFIFO are word-aligned, that is, the first byte of a packet is always stored in LSB. For example, if a long packet has 7-byte payload, the last byte is filled with dummy byte and the next packet is stored in the next word, as shown in Figure 13-32.

NOTE

CRC data is not stored in RxFIFO.

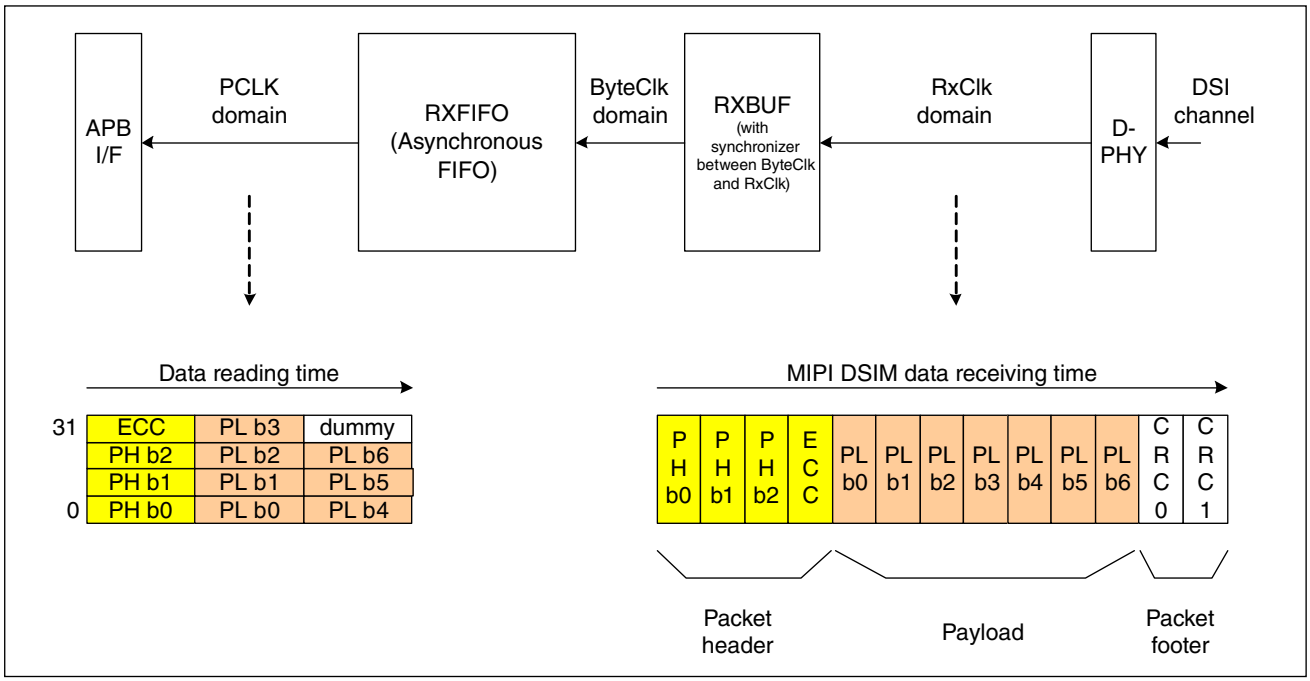


Figure 13-32. Rx data word alignment

13.6.2.6 Interfaces and protocol

Display Controller-to-DSI Conceptual Signal Converting Diagram in Video Mode.

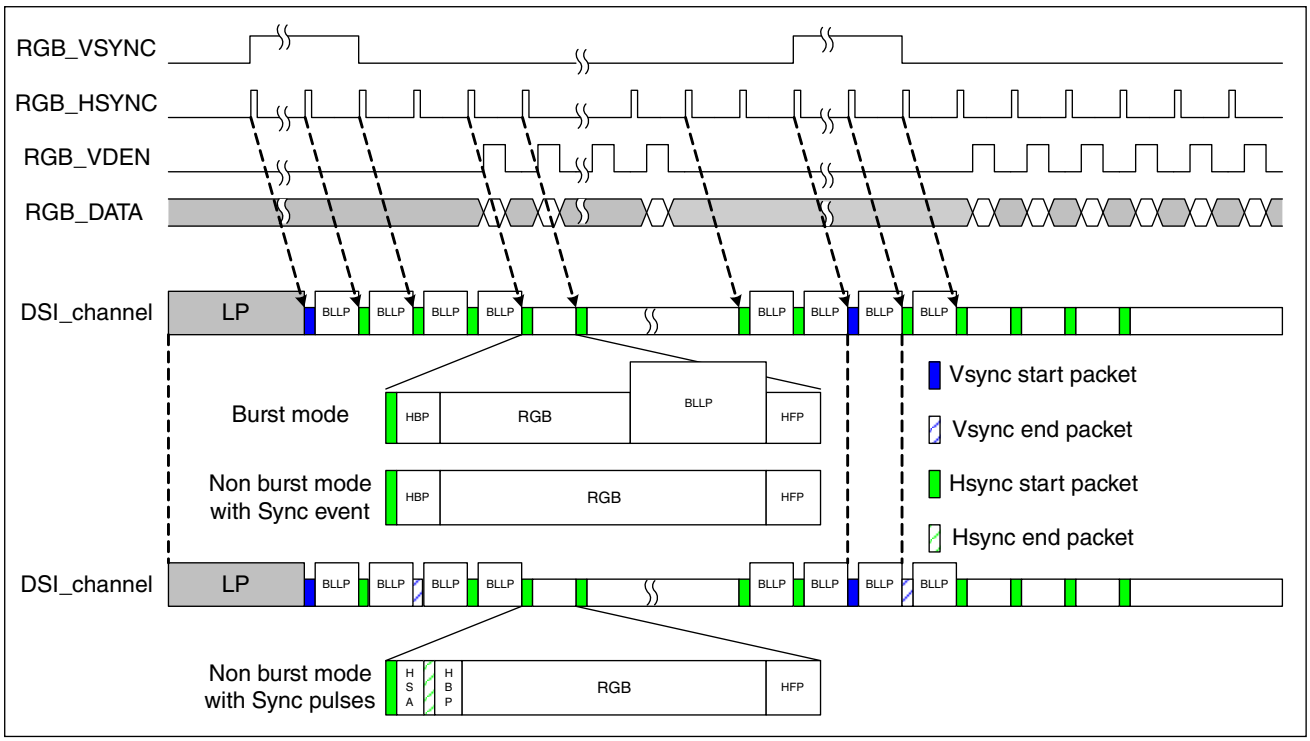


Figure 13-33. Signal converting diagram in video mode

13.6.2.7 Interface timing and protocol

13.6.2.7.1 Display controller interface

MIPI DSI Master (DSIM) has two-display controller interfaces, namely, RGB INTERFACE for main display and CPU INTERFACE (S-i80 INTERFACE) for main display/sub display. The Video mode uses RGB INTERFACE while the Command mode uses CPU INTERFACE.

The RGB image data is loaded on the data bus of RGB INTERFACE and S-i80 INTERFACE with the same order: RGB_VD[23:0] or SYS_VDOUT[23:0] is {R[7:0],G[7:0],B[7:0]}. Each byte aligns to the most significant bit. For instance, in the 12-bit mode, only three 4-bit values are valid as R, G, and B each, that is, data[23:20], data[15:12], and data[7:4]. The DSIM ignores rest of the bits.

13.6.2.7.2 RGB interface

Vsync, Hsync, and VDEN are active high signals. Among the three signals, Vsync and Hsync are pulse types that spend several video clocks. RGB_VD[23:0] is {R[7:0],G[7:0],B[7:0]}. All sync signals are synchronized to the rising edge of

RGB_VCLK. The display controller sends minimum one horizontal line length of Vsync pulse, V back porch (VBP), and V front porch (VFP). Hsync pulse width should be longer than 1-byte clock cycle.

13.6.2.7.2.1 HSA mode

HSA mode specifies the Horizontal Sync Area packet mode.

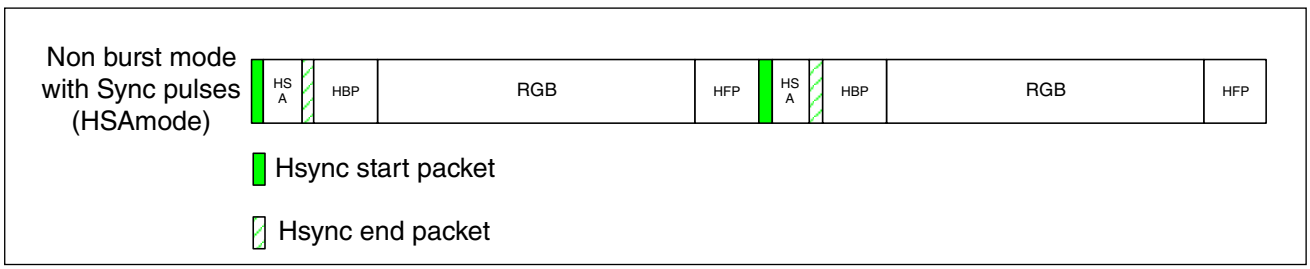


Figure 13-34. Block timing diagram of HSA enable mode (HSA mode reset : DSI_CONFIG[20] = 0)

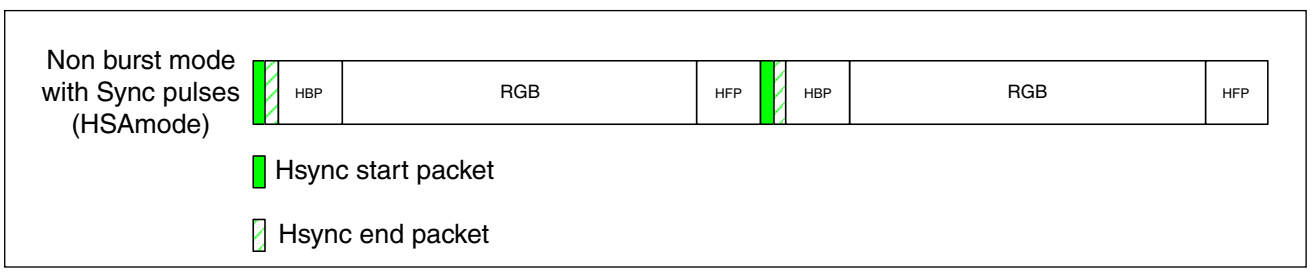


Figure 13-35. Block timing diagram of HSA disable mode (HSA mode set : DSI_CONFIG[20] = 1)

13.6.2.7.2.2 HBP mode

This option is valid only for burst mode. HBP mode specifies the Horizontal Back Porch packet mode.

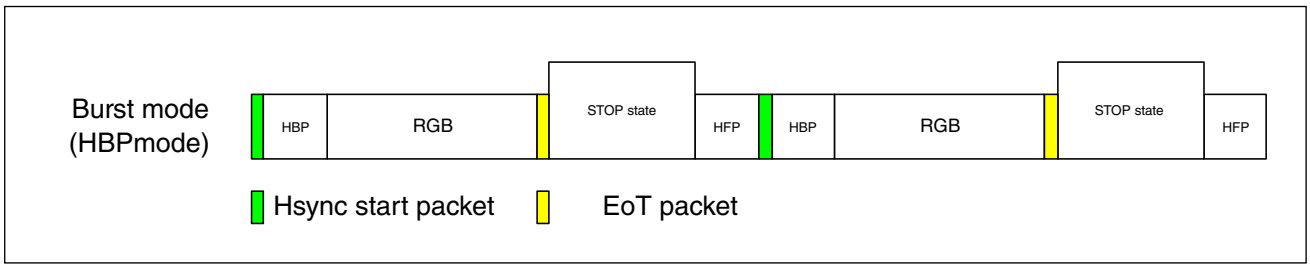


Figure 13-36. Block timing diagram of HBP enable mode (HBP mode reset : DSI_CONFIG[21] = 0)

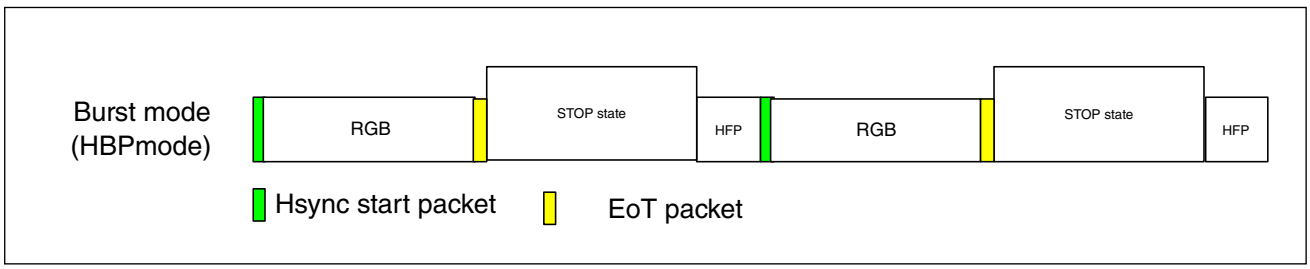


Figure 13-37. Block timing diagram of HBP disable mode (HBP mode set : DSI_CONFIG[21] = 1)

13.6.2.7.2.3 HFP mode

This option is valid only for burst mode. HFP mode specifies Horizontal Front Porch packet mode.

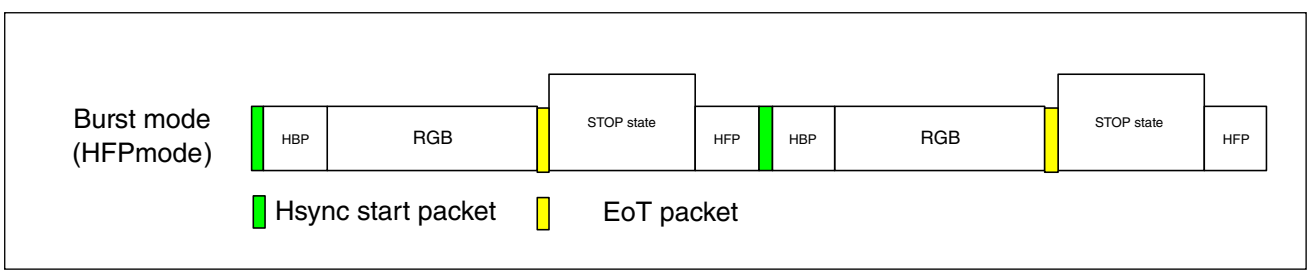


Figure 13-38. Block timing diagram of HFP enable mode (HFP mode set : DSI_CONFIG[22] = 0)

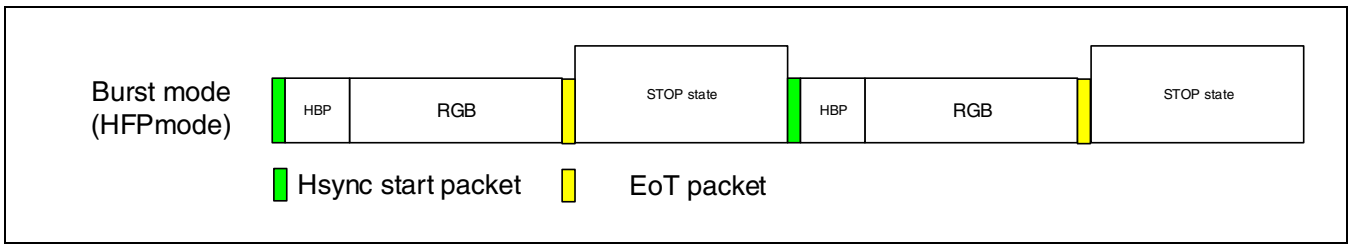


Figure 13-39. Block timing diagram of HFP disable mode (HFP mode set : DSI_CONFIG[22] = 1)

13.6.2.7.2.4 HSE mode

HSE mode is Horizontal Sync End packet mode.

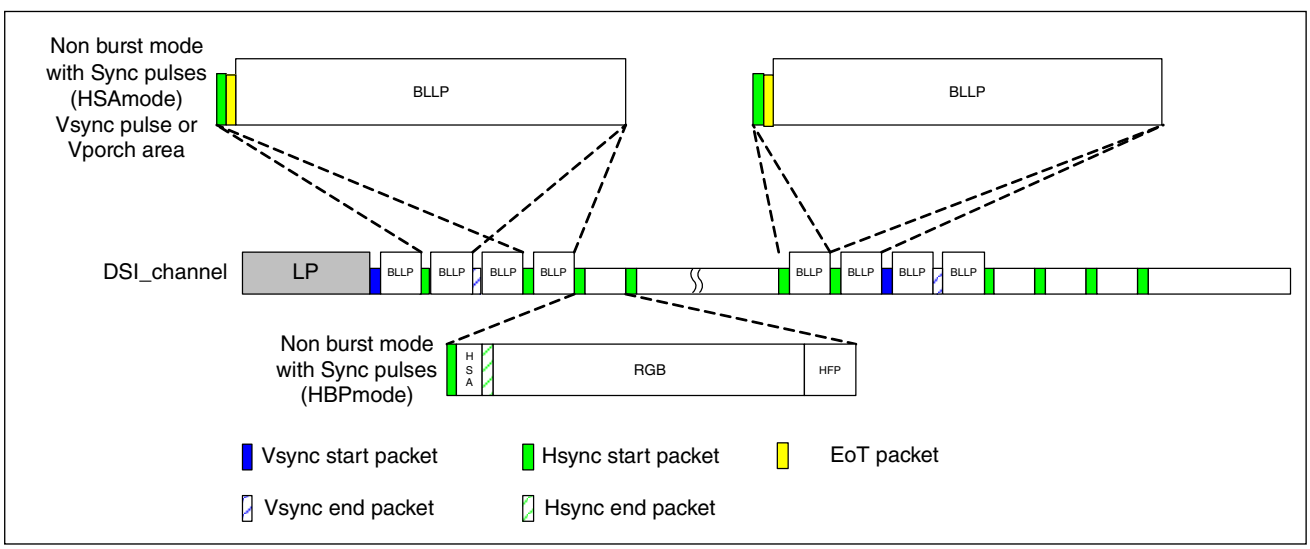


Figure 13-40. Block timing diagram of HSE enable mode (HSE mode reset : DSI_CONFIG[23] = 0)

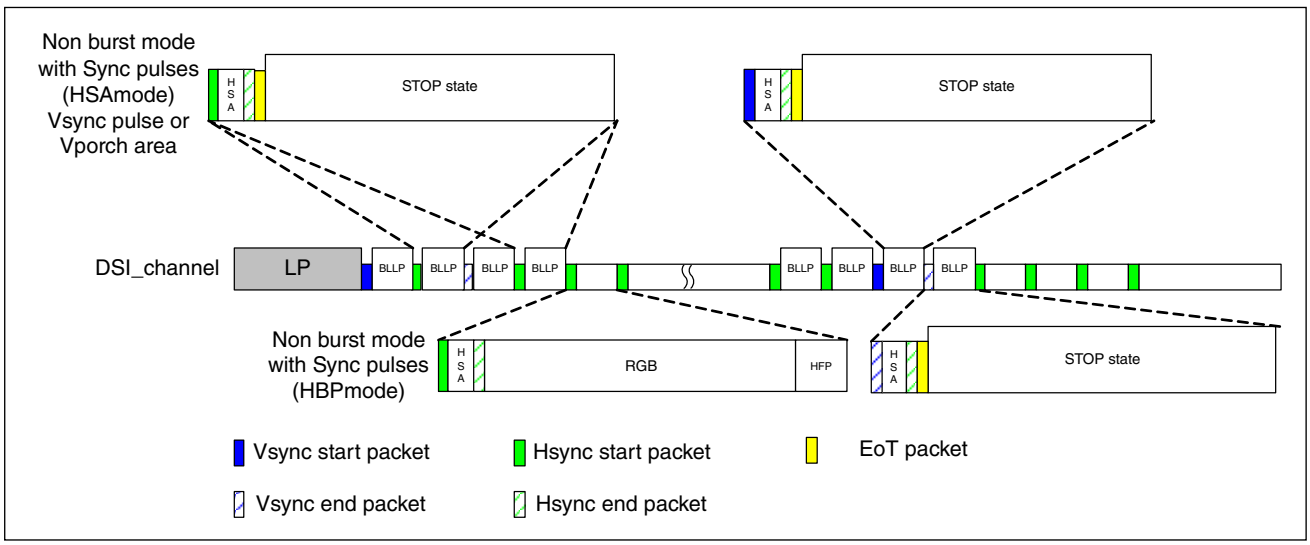


Figure 13-41. Block timing diagram of HSE disable mode (HSE mode set : DSI_CONFIG[23] = 1)

13.6.2.7.2.5 Transfer general data in video mode

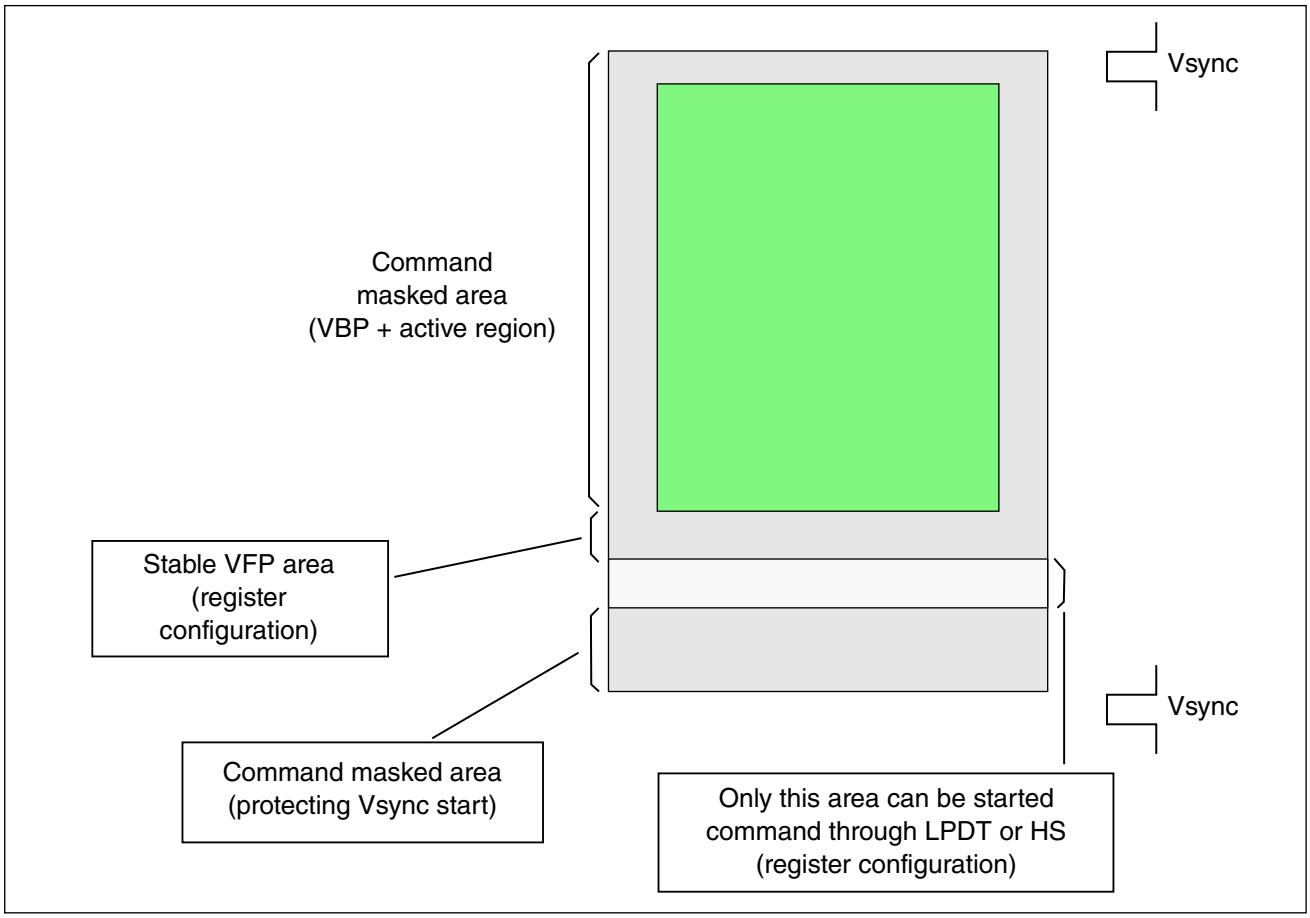


Figure 13-42. Stable VFP Area Before Command Transfer Allowing Area

MIPI DSIM Converts RGB Interface to Video Mode.

Vsync and Hsync packets are extremely important to protect image in Video mode. MIPI DSIM allows several lines in VFP area to transfer general data transfer. As shown in Figure 13-42, the vertical front porch is divided into three areas, namely, Stable VFP area, Command allowed area, and Command masked area.

The DSI_MVPORCH register configures stable VFP area. Configuration boundary is 11'h000 ~ 11'h7FFF.

The register also configures the command allowed area. Configuration boundary is 4'h0 ~ 4'hF in DSI_MVPORCH. Only this area is allowed to start "command transfer" through HS mode or Low Power Data Transmission (LPDT). In LPDT, data transferring takes a long time to complete (approximately hundreds of microseconds or more). In this time, Hsync packet does not arrive due to LPDT long packet. MIPI DSIM comprises of big size FIFO for lost Hsync packet. After LPDT, MIPI DSIM transfers these Hsync packets immediately through HS mode.

To protect Vsync, command masked area is masked area. This area is calculated using LPDT bandwidth. For example, if EscClk is 10MHz, the maximum long packet payload size is 1KB and LPDT, LPDT transferring time is 824us (packet size: 1030byte, LPDT maximum bandwidth: 10Mbps). If one line time is 20us, the line timing violation occurs in 42 lines. Therefore, command masked area is larger than 42 + "block". This "block" is transferring time of the violated Hsync packets.

Display controller should be configured in such a way that VFP lines are sum of stable VFP, command allowed area, and command masked area.

13.6.2.7.3 S-i80 interface

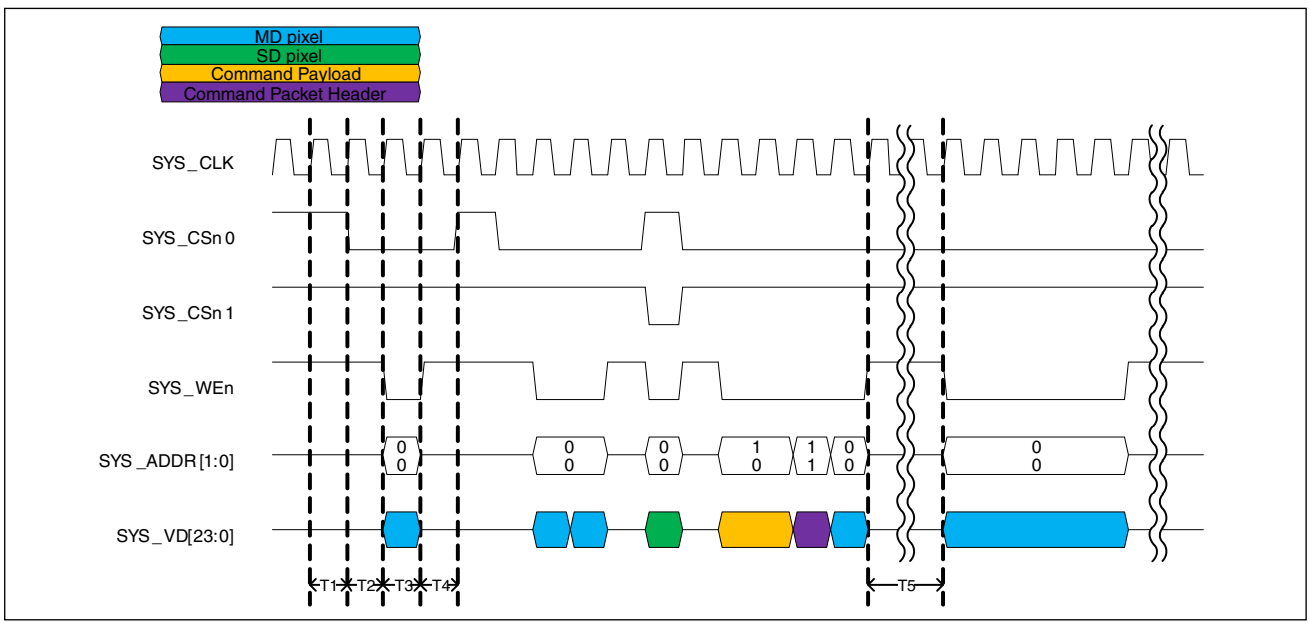


Figure 13-43. S-i80 I/F timing diagram

- T1 ≥ 1 clock cycle.
- T2 ≥ 0 clock cycle.
- T3 ≥ 1 clock cycle.
- T4 ≥ 0 clock cycle.
- T2+T3+T4 > 1 cycle of Byte clock.
- T5 ≥ 10 clock cycle (every horizontal blank period).

A display controller generates these signals with its internal clock: SYS_CS0/CS1, SYS_WE, and SYS_VD. MIPI DSI master decodes the SYS_ADDR. Table 13-22 describes the Command mode Interface address map.

Table 13-22. Command mode interface address map

SYS_ADDR[1:0]	Description
2'b00	Image data
2'b01	Reserved
2'b10	Payload data
2'b11	Packet Header

Figure 13-44 shows how MIPI DSI Master makes packet from the image data stream via S-i80 INTERFACE in Command mode. MIPI DSI master makes packet from the first line with Display Command Set (DCS) command "write_memory_start" and the other lines with DCS command "write_memory_continue".

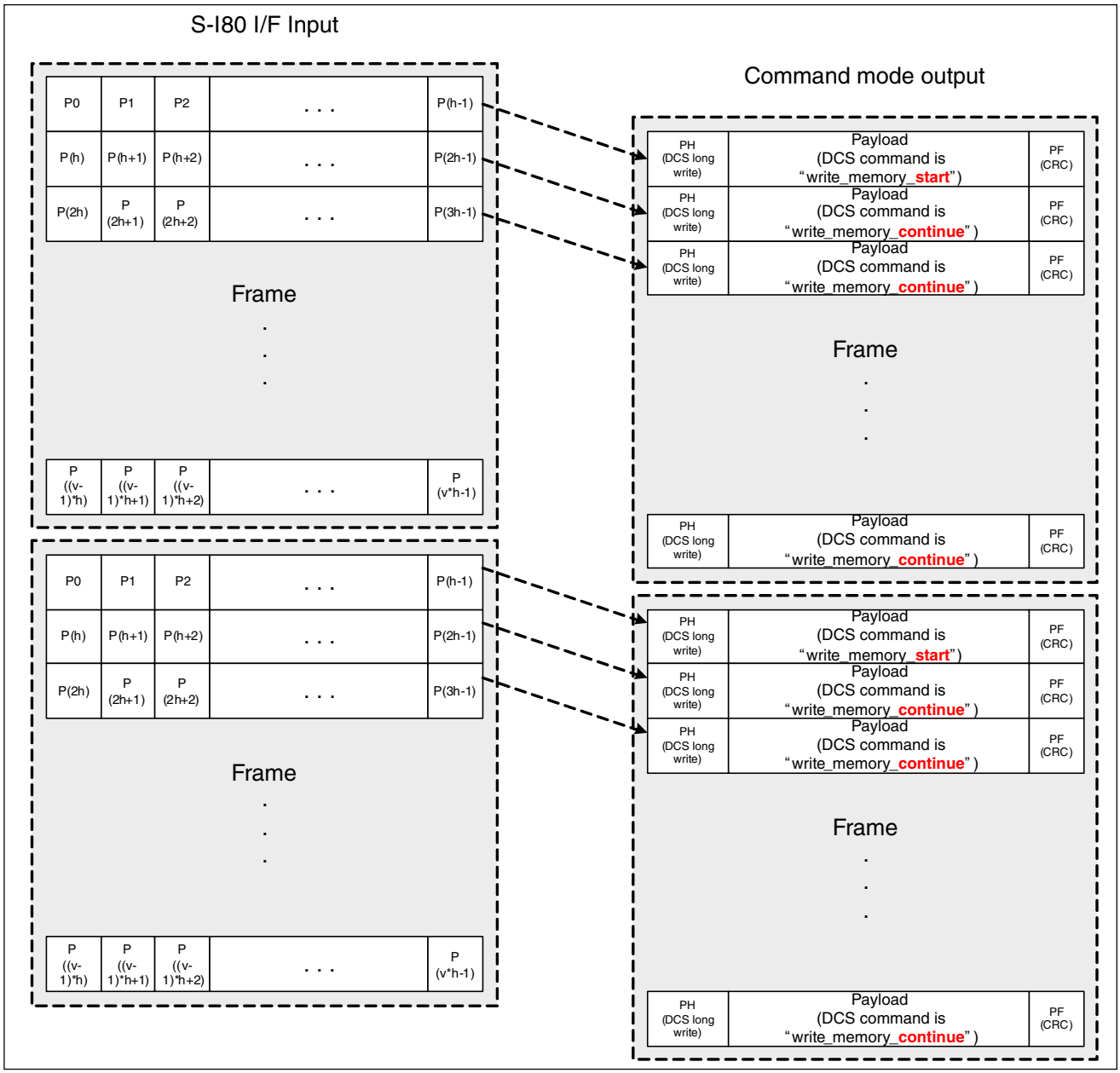


Figure 13-44. Packetizing for MIPI DSI command mode from S-i80 I/F

13.6.2.7.4 Relation between input transactions and DSI transactions

Table 13-23. Relation between input transactions and DSI transactions

Input Interface	Input transaction	DSI Transaction
RGB	RGB transaction	Specifies the RGB Packet. 888, 666, 666 (loosely packed), and 565 should be specified via register configuration.
RGB / S-i80	RGB / S-i80 Image Transaction	Specifies the Data type, that is, "DCS Long Write packet".

Table continues on the next page...

Table 13-23. Relation between input transactions and DSI transactions (continued)

Input Interface	Input transaction	DSI Transaction
		(DCS command is "memory write start/continue".)
S-i80	S-i80 Command Transaction	Specifies any DSI packet. Bytes in S-i80 transaction should be the same bytes in DSI packets.
SFR	Header and Payload FIFO access	Specifies any DSI Packets. Bytes in APB transaction should be the same bytes in DSI packets.

13.6.2.7.5 PPI interface timing and protocol

13.6.2.7.5.1 High speed data transfer

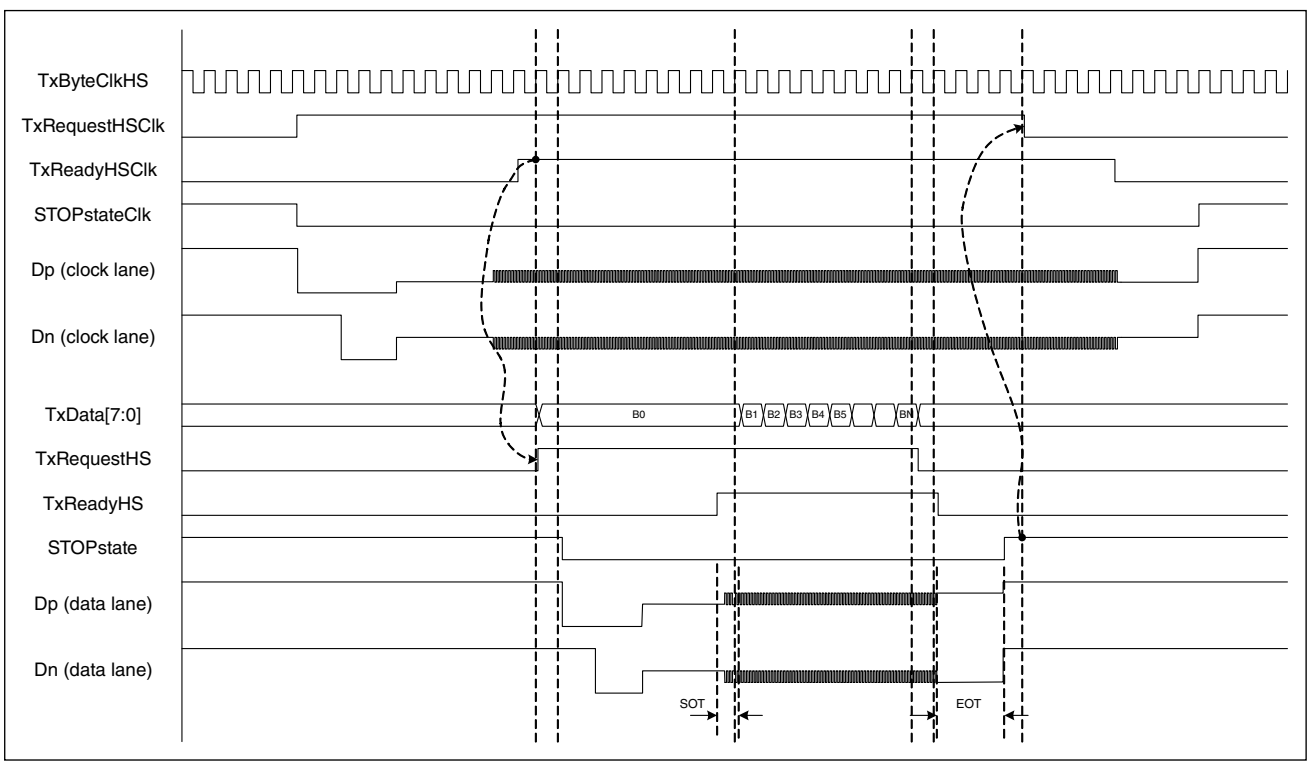


Figure 13-45. Timing diagram of high speed data transfer

13.6.2.7.5.2 Low power data transfer

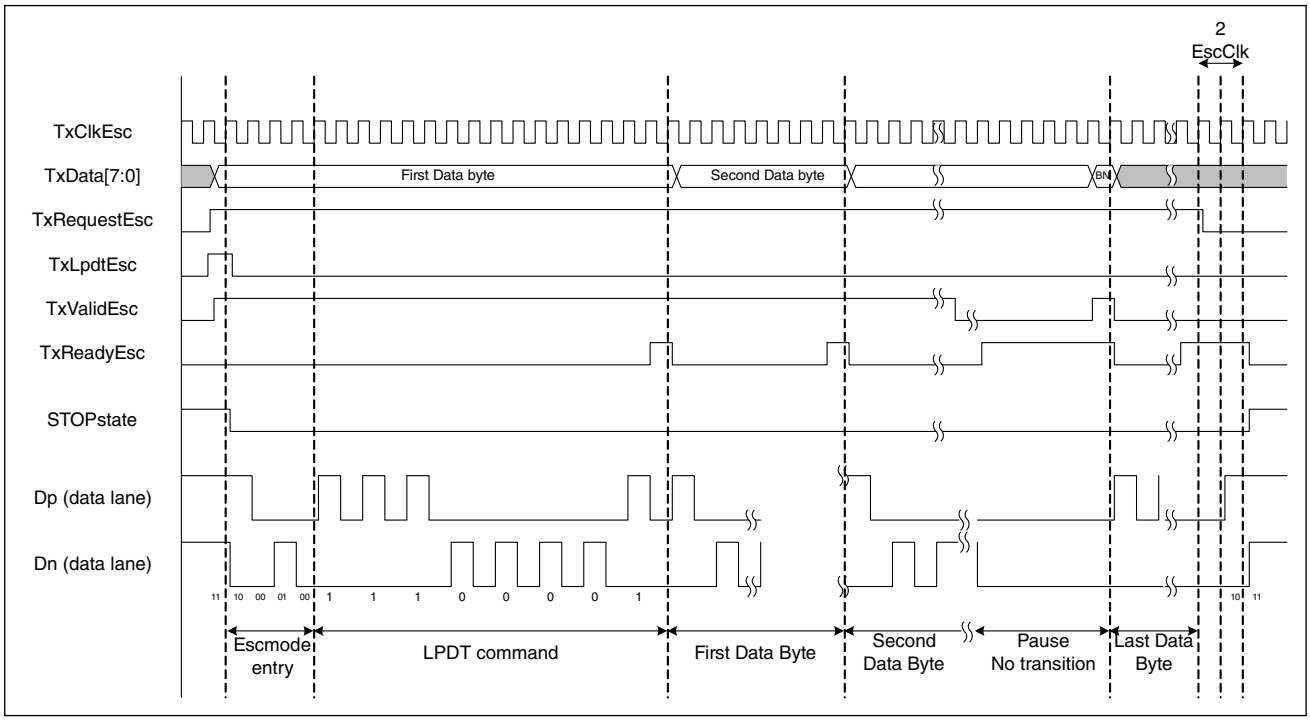


Figure 13-46. Timing diagram of low power data transfer

13.6.2.7.5.3 Ultra-Low power state

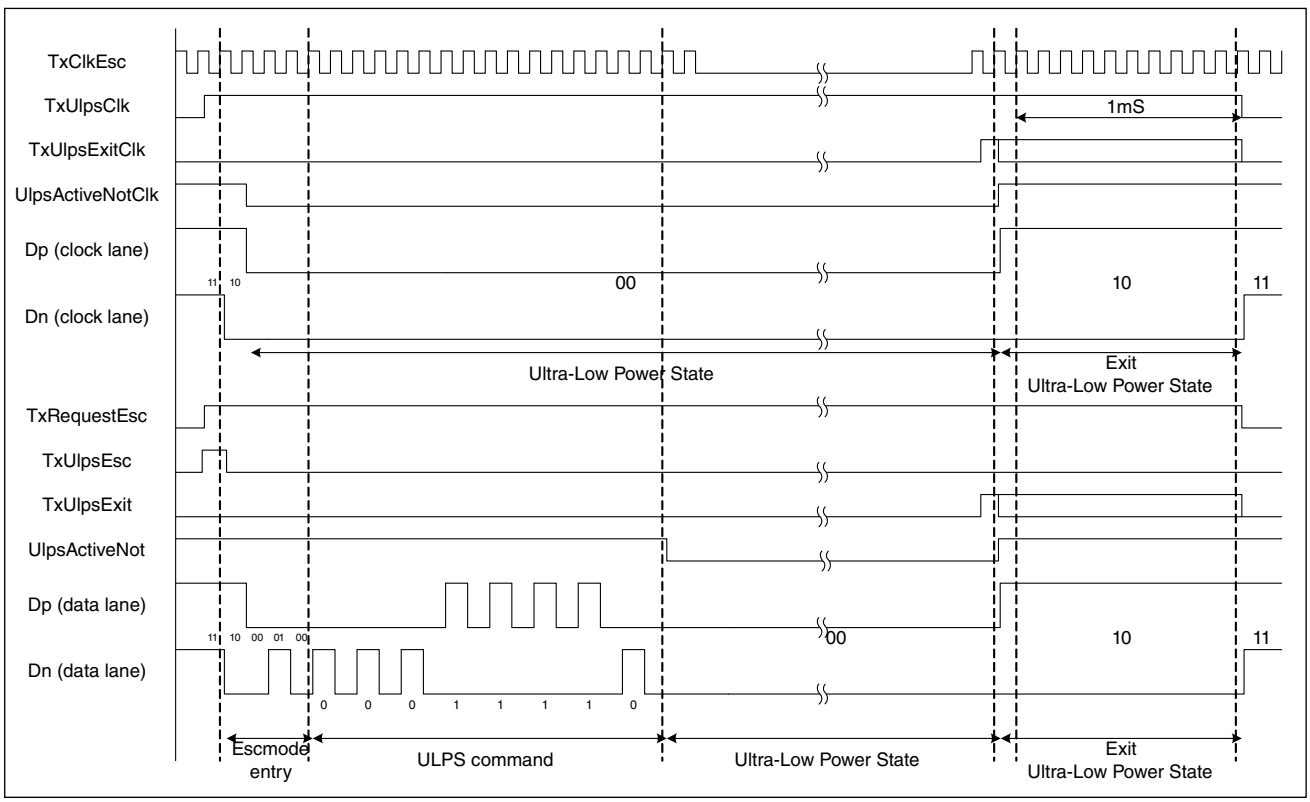


Figure 13-47. Timing diagram of ultra low power state

13.6.2.7.5.4 Trigger function

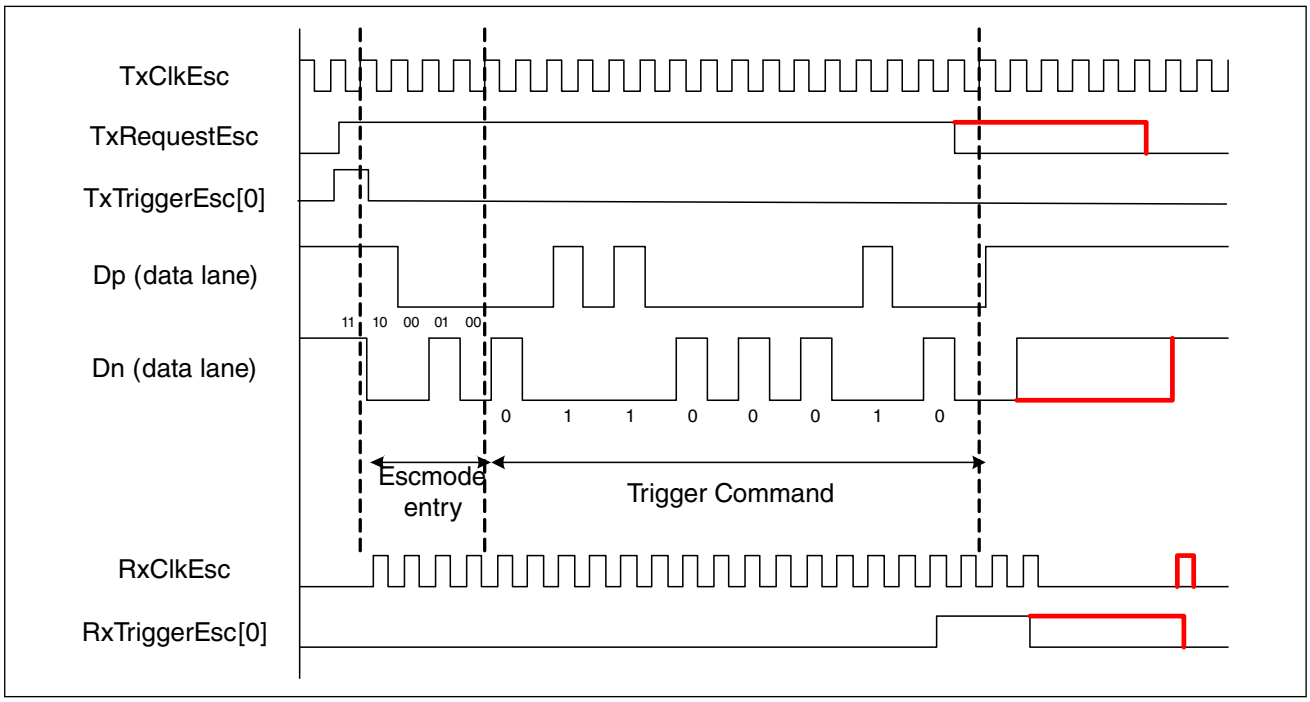


Figure 13-48. Diagram of trigger function "Remote Reset"

There are four types of trigger commands:

- 01100010 : remote reset – forward direction
- 01011101 : Tearing Effect – reverse direction
- 00100001 : Acknowledge – reverse direction
- 10100000 : Unknown

If DSI D-PHY received these signals, D-PHY indicates trigger function to protocol layer.

13.6.2.7.5.5 Turn-around and low power data receiving

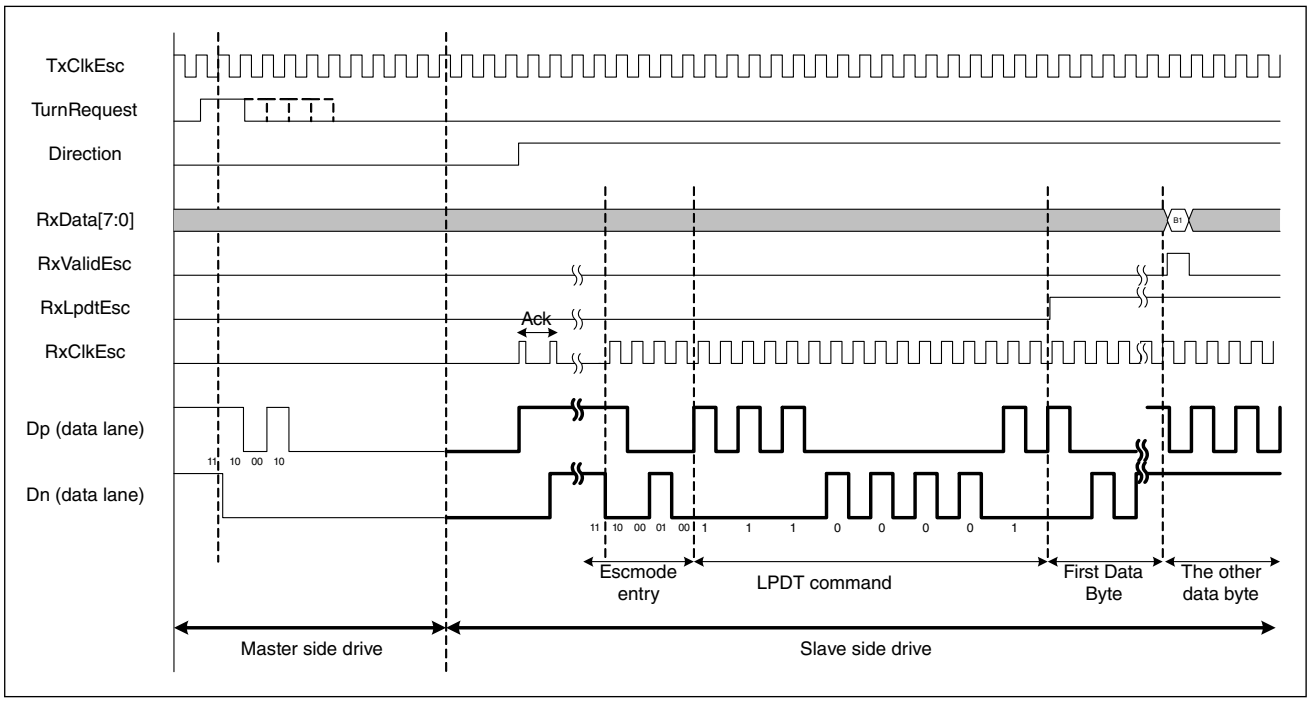


Figure 13-49. Timing diagram of Bus-Turn-Around

13.6.2.7.5.6 Tearing Effect (TE) signaling

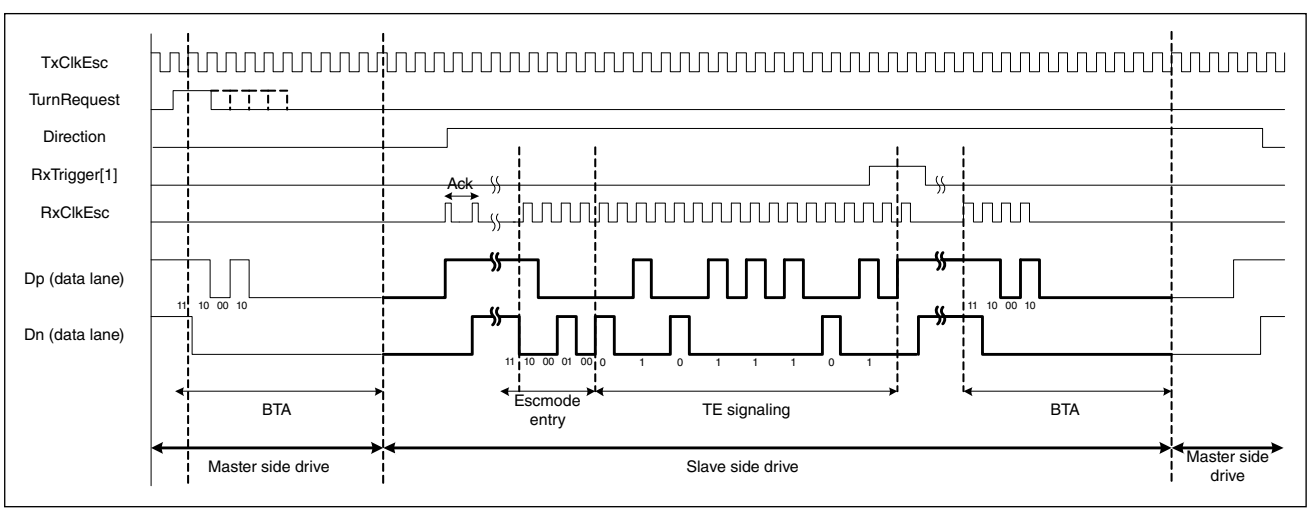


Figure 13-50. TE signaling

13.6.2.8 Configuration and Modes

13.6.2.8.1 Video mode VS Command mode

MIPI DSI Master Block supports two modes, namely, Video mode and Command mode.

Command mode uses RGB interface and S-i80 interface. Mode can change via SFRs.

13.6.2.8.2 Single display

For single display configuration, use video mode or command mode.

13.6.2.8.3 PLL

To transmit Image data, MIPI DSI Master Block needs high frequency clock (80MHz ~ 1GHz) generated by PLL.

To configure PLL, MIPI DSI Master comprises of SFRs and corresponding interface signals. PLL is embedded in PHY module.

13.6.2.8.4 Buffer

In MIPI DSI standard specification, DSI Master sends image stream in burst mode. The image stream transmits in high-speed and bit-clock frequency. This mode allows the device to stay in stop state longer to reduce power consumption. For this mode, MIPI DSI Master has a line buffer to store one complete line and send it faster at the next line time.

13.6.3 Clocks

The table below shows the clocks described in this module:

Table 13-24. Clocks

Clocks	Description
PCLK	APB Clock This signal represents the output clock domain in RX_FIFO module.

13.6.4 Reset

There is software reset register DSI_SWRST, with register bits: Software reset and Function reset that resets all FF in MIPI DSI. DSI_INTSRC[SwRstRelease] and DSI_INTMSK[MskSwRstRelease] are two register release bits that releases software reset.

There is a remote reset trigger function specified while configuring MIPI DSI master. It is set using DSI_ESCMODE[TxTriggerRst] register bit and is cleared automatically post the trigger operation.

MIPI DSI has a status bit, DSI_STATUS[SwRstRls] which specifies the software reset status. Please refer to the MIPI_DSI register descriptions for more information.

13.6.5 Interrupts

The interrupt source register DSI_INTSRC, identifies various interrupt sources. The clock stable bit in the interrupt source register, PllStable is set when the timer value in PLL timer register reads 0x00000000. DSI_INTMSK is an interrupt mask register that masks the interrupt sources. The register bits in the interrupt source register are set even if the corresponding bits in DS_INTMSK are masked off by the interrupt mask register.

13.6.6 External Signals

Table 13-25. External Signals

Signal	Description	Direction
CLK_N	Negative MIPI Rx Differential Clock	I
CLK_P	Positive MIPI Rx Differential Clock	I
Dn_N	Negative MIPI Rx Differential Data	I
Dn_P	Positive MIPI Rx Differential Data	I

13.6.7 Initialization

13.6.7.1 Initialization after power-on / reset

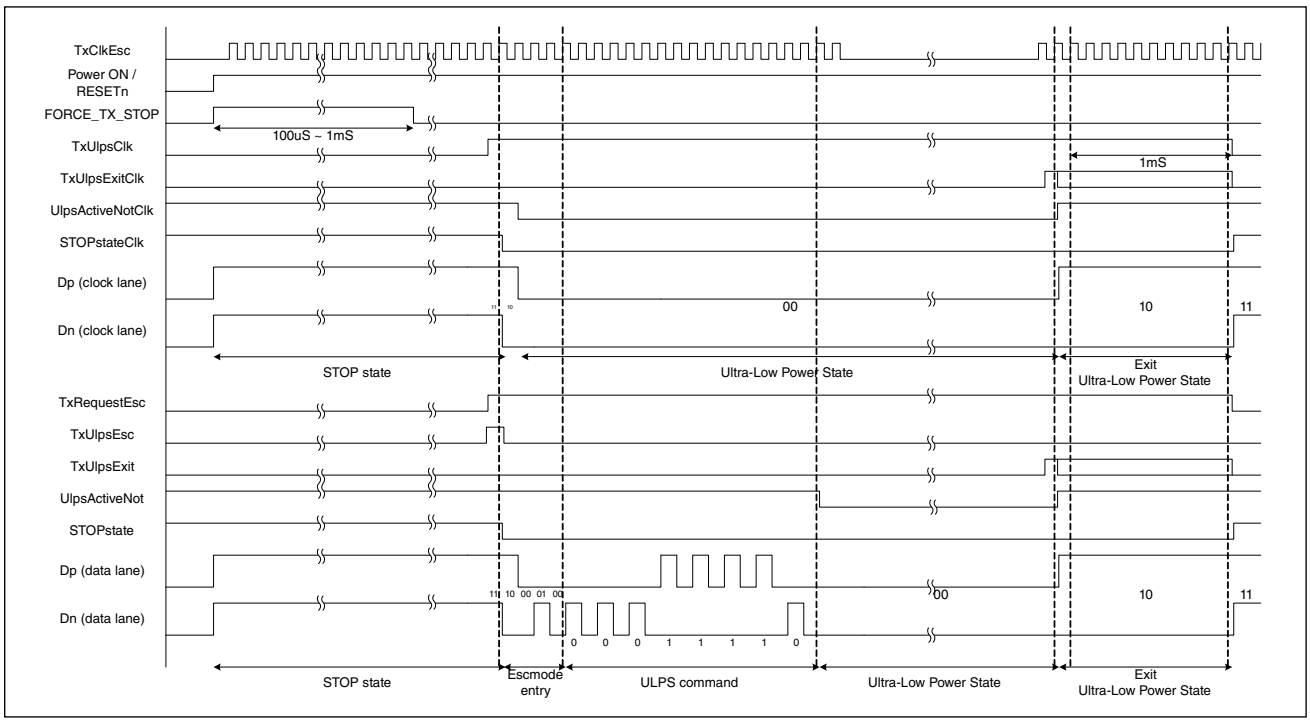


Figure 13-51. Timing diagram of initialization

13.6.8 Application Information

13.6.8.1 Display mode

13.6.8.1.1 Video mode

Video mode starting sequence:

1. Set RGB I/F in Display Controller(RGB888), also set Video mode at MIPI DSI Master.
2. Configure all parameters for displaying image (define resolution, back porch, front porch, data lane number, HS clock frequency(byte clock), etc).
3. Start image transmission.

13.6.8.1.2 Command mode

Command mode starting sequence:

1. Set S-i80 I/F in Display Controller (RGB888), also set Command mode at MIPI DSI Master.
2. Configure all parameters for displaying image.

3. Set the command at MIPI DSI registers using MIPI DCS.
4. S-i80 I/F is just used to transmit image data. The users have to read display peripheral status and write command through MIPI DSI registers.

13.6.8.2 Programming Model

Table 13-26. Behavior programming each case

No.	Mode	Case	Pseudo Code

1	Video	Initialize	1. reset or power on
			2. SFR_Write(Display Controller_registers, RGB I/F_888);
			3. SFR_Write(DSI_CLKCTRL,{16'h1000,Esc_prescaler}); //EscClk = ByteClk/Esc_prescaler;
			4. SFR_Write(DSI_MDRESOL,{M_vert,M_horizon});
			5. SFR_Write(DSI_MVPORCH,M_Vback);
			6. SFR_Write(DSI_MHPORCH,{M_Hfront,M_Hback});
			7. SFR_Write(DSI_MSYNC,{M_VSA,M_HSA}); // In SyncEvent mode, Ignore it
			8. SFR_Write(DSIM_CONFIG,{HFPdismode, MPix, Sync_inform,BLLP,Burst, Videomode(1),NumofDatlane,Lane_EN); // set the value of each bit
			9. SFR_Write(DSI_PLLCTRL, {4'h0, Freq_band,4'h8,PMS}); // PLL enable
			10. Wait for 100uS ~ 1mS using system timer after reset or power on.
			In this time, ForceSTOPstate at DSIM_CONFIG is assigned to D-PHY by default.
			100uS ~ 1mS is not mandatory. Please see the DSI slave spec.
			11. SFR_Write(DSI_CONFIG, ForceSTOPstate(0)); // disable
			12. while(~SFR_Read(DSI_STATUS, StopstateClk, StopstateDat));
			D-PHY & protocol layer(MIPI DSI Master block) in STOP state
			13. SFR_Write(DSI_ESCMODE, TxUlps(1),TxUlpsClk(1));
			D-PHY & protocol layer(MIPI DSI Master block) in ULPS
			In ULPS, User can turn off Byte/Esc clock for Low power estimation
			SFR_Write(DSI_CLKCTRL, ESCClk_EN(0), BytClk_En(0));
			Or
			SFR_Write(DSI_CLKCTRL,ByteClk_EN(0),Lane_EscClkEn[4 :0](0));
			If user wants to turn on display, Turn on EscClk and
			14. SFR_Write(DSI_ESCMODE, TxUlpsExit(1),TxUlpsClkExit(1));
			15. while(~SFR_Read(DSI_STATUS, UlpsClk, UlpsDat));
			16. SFR_Write(DSI_ESCMODE, TxUlpsExit(0), TxUlpsClkExit(0));
			17. Wait for about 1mS using system timer.
			1mS is not mandatory, just an example. Please see the DSI slave spec.
			18. SFR_Write(DSI_ESCMODE, TxUlps(0),TxUlpsClk(0));
			19. while(~SFR_Read(DSI_STATUS, StopstateClk, StopstateDat));
			D-PHY & protocol layer(MIPI DSI Master block) in STOP state

Table continues on the next page...

Table 13-26. Behavior programming each case (continued)

			20. MIPI DSI Master is ready for HS/LP operation.
2	Video	HS Data Transfer	In Video mode, there is no pseudo code. After initializing, SFR_Write(Display Controller_register, START); If line image is full in line buffer, MIPI DSI master transmit data through long data packet. If user wants to send general command to display from memory, user will have to set following sequence. 1. SFR_Write(DSI_PAYLOAD,PAYLOAD_DATA); 2. SFR_Write(DSI_PKTHDR,PKTHDR_DATA);
3	Video	LP Data Transfer	If user wants to send general command to display from memory via LPDT: 1. SFR_Write(DSI_ESCMODE, TxLpdt); 2. If DSIM transfer data in HS mode, DSIM will wait STOPstate and transfer mode change HS to LP. 3. same as the previous test scenario cases.
4	Video	ULPS	ULPS sequence is same the initialization process (test scenario #1), 13~20
5	Video	Trigger function	The process is as follows: 1. SFR_Write(DSI_ESCMODE, TxTriggerRst); 2. If DSI master state is not STOP state, DSIM will wait for LP stop mode and transmit trigger automatically.
6	Video	BTA	There are two cases of BTA. One case is after transferring general read data packet or DCS command write(set_tear_on) and the other one is BTA_req bit setting by software. Previous case is automatically setting BTA. Next case is as follows: If user reads the state or data of display panel, user will use BTA. 1. SFR_Write(DSI_ESCMODE, BTA); 2. If DSIM transfer data, DSIM will wait STOPstate. 3. After BTA assert, DSIM release DSI channel. In reverse direction (DSI slave has DSI channel grant), all DSIM FIFO is flushed and masked Video/SFR inputs. Only LPRX data input is unmasked. 4. If DSIS releases DSI channel to DSIM, Int_BusTurnOver is generated. After ISR, DSIM can transfer data.

7	Command	Initialize	1. reset or power on
			2. SFR_Write(Display Controller_registers, S-i80 I/F_888);
			3. SFR_Write(DSI_CLKCTRL,{16'h1000,Esc_prescaler});
			4. SFR_Write(DSI_MDRESOL,{M_vert,M_horizon});
			6. SFR_Write(DSI_CONFIG,{MPix, Sync_inform,NumofDatlane,Videomode(0), Lane_EN); // set the value of each bit
			7. SFR_Write(DSI_PLLCTRL, {4'h0, Freq_band,4'h8,PMS}); // PLL enable
			8. Wait for 100uS ~ 1mS using system timer after reset or power on. In this time, ForceSTOPstate at DSIM_CONFIG is assigned to D-PHY by default. 100uS ~ 1mS is not mandatory. Please see the DSI slave spec.

Table continues on the next page...

Table 13-26. Behavior programming each case (continued)

<table>
<tr><td></td><td></td><td></td><td>9. SFR_Write(DSI_CONFIG, ForceSTOPstate(0)); // disable
10. while(~SFR_Read(DSI_STATUS, StopstateClk, StopstateDat));
D-PHY & protocol layer(MIPI DSI Master block) in STOP state
11. SFR_Write(DSI_ESCMODE, TxUlps(1),TxUlpsClk(1));
D-PHY & protocol layer(MIPI DSI Master block) in ULPS
In ULPS, User can turn off Byte/Esc clock for Low power estimation
SFR_Write(DSI_CLKCTRL, ESCClk_EN(0), BytClk_En(0));
Or
SFR_Write(DSI_CLKCTRL,ByteClk_EN(0),Lane_EscClkEn[4 :0](0));
If user wants to turn on display, Turn on EscClk and
12. SFR_Write(DSI_ESCMODE, TxUlpsExit(1),TxUlpsClkExit(1));
13. while(~SFR_Read(DSIM_STATUS, UlpsClk, UlpsDat));
14. SFR_Write(DSI_ESCMODE, TxUlpsExit(0), TxUlpsClkExit(0));
15. Wait for about 1mS using system timer.
1mS is not mandatory, just an example. Please see the DSI slave spec.
16. SFR_Write(DSI_ESCMODE, TxUlps(0),TxUlpsClk(0));
17. while(~SFR_Read(DSI_STATUS, StopstateClk, StopstateDat));
D-PHY & protocol layer(MIPI DSI Master block) in STOP state
18. MIPI DSI Master is ready for HS/LP operation.</td></tr>
<tr><td>8</td><td>Command</td><td>HS Data Transfer</td><td>In Command mode, there is no pseudo code.
After initializing,
SFR_Write(Display Controller_register, START);
If line image is full in line buffer, MIPI DSI master transmit data through long data packet.
<table>
<tr><th>SYS_ ADD[1]</th><th>SYS_ ADD[0]</th><th>SYS_ CS0</th><th>SYS_ CS1</th><th>Contents</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Main display Image</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>Payload data</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>Packet Header data</td></tr>
<tr><td colspan="4">The other cases</td><td>Reserved</td></tr>
</table>
If user wants to send general command to display from memory, user will have to set following sequence.
SFR_Write(DSI_PAYLOAD,PAYLOAD_DATA);
SFR_Write(DSI_PKTHDR,PKTHDR_DATA);
If user wants to update partial image,
wait after Display Controller transfer last line.
wait(DSI_INTSRC, Frame_done); or check STOPstate.
SFR_Write(DSI_MDRESOL,{M_vert,M_horizon}); or</td></tr>
</table>

Table continues on the next page...

Table 13-26. Behavior programming each case (continued)

			Turn on Display Controller for Image transfer
9	Command	LP Data Transfer	If user wants to send general command to display from memory via LPDT. SFR_Write(DSI_ESCMODE, TxLpdt); If DSIM transfer data in HS mode, DSIM will wait STOPstate and transfer mode change HS to LP. same as the previous test scenario cases.
10	Command	ULPS	ULPS sequence is same the initialization process (test scenario #1), 13~20
11	Command	Trigger function	The process is as follows: SFR_Write(DSI_ESCMODE, TxTriggerRst); If DSI master state is not STOP state, DSIM will wait for LP stop mode and transmit trigger automatically.
12	Command	BTA	There are two cases of BTA. One case is after transferring general read data packet or DCS command write(set_tear_on) and the other one is BTA_req bit setting by software. Previous case is automatically setting BTA. Next case is as follows: If user read the state or data of display panel, user will use BTA. SFR_Write(DSI_ESCMODE, BTA); If DSIM transfer data, DSIM will wait STOPstate. After BTA assert, DSIM release DSI channel. In reverse direction (DSI slave has DSI channel grant), all DSIM FIFO is flushed and masked Video/SFR inputs. Only LPRX data input is unmasked. If DSIS releases DSI channel to DSIM, “Int_BusTurnOver” is generated. After ISR, DSIM can transfer data.

13	ISR
	(Error)
			SFR_Write(Global_INTC_MASK, INTMASK(DSIM));
			SFR_Read(Global_INTC_SRC);
			SFR_Write(DSI_INTMSK, all_int_mask);
			SFR_Read(DSI_INTSRC);
			Error reporting to application layer
			SFR_Write(DSI_INTSRC, intsrc); // clear interrupt source
			SFR_Write(DSI_INTMSK, all_int_unmask);
			SFR_Write(Global_INT_SRC, intsrc); // clear interrupt source
			SFR_Write(Global_INT_MASK, INTUNMASK(DSIM));
			SFR_Write(DSI_CONFIG, ForceSTOPstate(1));
			enough time later
			SFR_Write(DSI_CONFIG, ForceSTOPstate(0));

14	ISR (Rx Data)		After BTA, wait interrupt “Int_BusTurnOver”. The process is as follows: SFR_Write(Global_INTC_MASK, INTMASK(DSIM)); SFR_Read(Global_INTC_SRC); SFR_Write(DSI_INTMSK, all_int_mask); SFR_Read(DSI_INTSRC);

Table continues on the next page...

Table 13-26. Behavior programming each case (continued)

			SFR_Write(DSI_INTSRC, {BusTurnOver,RxDone}); // clear interrupt source
			SFR_Write(DSI_INTMSK, all_int_unmask);
			SFR_Write(Global_INT_SRC, intsrc); // clear interrupt source
			SFR_Write(Global_INT_MASK, INTUNMASK(DSIM));
			SFR_Read(DSI_RXFIFO);

15	ISR
	(trigger flag)
			After BTA, wait interrupt “Int_BusTurnOver”. The process is as follows:
			SFR_Write(Global_INTC_MASK, INTMASK(DSIM));
			SFR_Read(Global_INTC_SRC);
			SFR_Write(DSI_INTMSK, all_int_mask);
			SFR_Read(DSI_INTSRC);
			flag reporting to application layer
			SFR_Write(DSI_INTSRC, {BusTurnOver, trigflags}); // clear interrupt source
			SFR_Write(DSI_INTMSK, all_int_unmask);
			SFR_Write(Global_INT_SRC, intsrc); // clear interrupt source
			SFR_Write(Global_INT_MASK, INTUNMASK(DSIM));

16	Software reset	Function reset	If you want to assert Software reset, all clocks must be turned on because software reset is a synchronous reset. Turn on RGB_VCLK only from Display controller SFR_Write(DSI_PLLTMR, PllTimer); SFR_Write(DSI_CLKCTRL, {EscClkEn, EscPrescaler}); SFR_Write(DSI_PLLCTRL, {PllEn,PMS}); wait Pll stable (polling status register or ISR) SFR_Write(DSI_SWRST, FuncRst); wait for software reset release (polling status register or ISR) ready for operation
		Software reset	If you want to assert Software reset, all clocks must be turned on because software reset is synchronous reset. Turn on RGB_VCLK only from Display controller SFR_Write(DSI_PLLTMR, PllTimer); SFR_Write(DSI_CLKCTRL, {EscClkEn, EscPrescaler}); SFR_Write(DSI_PLLCTRL, {PllEn,PMS}); wait Pll stable (polling status register or ISR) SFR_Write(DSI_SWRST, SwRst); wait software reset release (polling status register or ISR) re-configure all registers ready for operation
17	Clock source changing from PLL to external		Turn on RGB_VCLK(only clock) from Display controller and STOP data inputting to MIPI DSIM SFR_Write(DSI_SWRST, FuncRst); or SFR_Write(DSIM_SWRST, SwRst); Turn off any inputs in MIPI DSIM.

Table continues on the next page...

Table 13-26. Behavior programming each case (continued)

	clock source		SFR_Write(DSI_PLLCTRL, PLL disable); SRF_Write(DSI_CLKCTRL, ByteClkSel); SFR_Write(DSI_SWRST, FuncRst); or SFR_Write(DSIM_SWRST, SwRst); reconfiguration Turn on Display controller or input data to MIPI DSIM ready for operation
18	Usage of EoT packet when user wants to transfer command packet through HS mode.		Video mode after turning on display controller FSM of Display controller I/F of MIPI DSIM generates EoT packet automatically Just write command or general packet in SFR FIFO Video mode after turning off/before turning on display controller Write command or general packet in SFR FIFO If user does not want to transfer general or command packet, user should write EoT packet in SFR FIFO. If general or command packets are transferred through LP mode, previous programming step is "don't care".

13.6.9 Register Map

This section includes the MIPI DSI module memory map and detailed descriptions of the registers.

13.6.9.1 MIPI_DSI register descriptions

13.6.9.1.1 MIPI_DSI memory map

MIPI_DSI base address: 32E6_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Specifies the DSI version register. (DSI_VERSION)	32	RO	0106_0100h
4h	Specifies the status register. (DSI_STATUS)	32	RO	0010_010Fh
8h	Specifies the RGB FSM status register. (DSI_RGB_STATUS)	32	RO	0000_0001h
Ch	Specifies the software reset register. (DSI_SWRST)	32	RW	0000_0000h
10h	Specifies the clock control register. (DSI_CLKCTRL)	32	RW	0000_FFFFh
14h	Specifies the time out register. (DSI_TIMEOUT)	32	RW	00FF_FFFFh
18h	Specifies the configuration register. (DSI_CONFIG)	32	RW	0200_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1Ch	Specifies the escape mode register. (DSI_ESCMODE)	32	RW	0000_0000h
20h	Specifies the main display image resolution register. (DSI_MDRESOL)	32	RW	0300_0400h
24h	Specifies the main display Vporch register. (DSI_MVPORCH)	32	RW	F000_0000h
28h	Specifies the main display Hporch register. (DSI_MHPORCH)	32	RW	0000_0000h
2Ch	Specifies the main display Sync Area register. (DSI_MSYNC)	32	RW	0000_0000h
30h	Specifies the sub display image resolution register. (DSI_SDRESOL)	32	RW	0300_0400h
34h	Specifies the interrupt source register. (DSI_INTSRC)	32	RW	0000_0000h
38h	Specifies the interrupt mask register. (DSI_INTMSK)	32	RW	BB37_FFFFh
3Ch	Specifies the packet header FIFO register. (DSI_PKTHDR)	32	WO	0000_0000h
40h	Specifies the payload FIFO register. (DSI_PAYLOAD)	32	WO	0000_0000h
44h	Specifies the read FIFO register. (DSI_RXFIFO)	32	RO	Table 13-113
48h	Specifies the FIFO threshold level register. (DSI_FIFOTHLD)	32	RW	0000_01FFh
4Ch	Specifies the FIFO status and control register. (DSI_FIFOCTRL)	32	RW	0155_551Fh
50h	Specifies the FIFO memory AC characteristic register. (DSI_MEMACCHR)	32	RW	0000_4040h
78h	Specifies the Multi Packet, Packet Go register. (DSI_MULTI_PKT)	32	RW	0000_0002h
90h	Specifies the 1Gbps D-PHY PLL control register. (DSI_PLLCTRL_1G)	32	RW	0000_0000h
94h	Specifies the PLL control register. (DSI_PLLCTRL)	32	RW	0000_0000h
98h	Specifies the PLL control register 1. (DSI_PLLCTRL1)	32	RW	0000_0000h
9Ch	Specifies the PLL control register 2. (DSI_PLLCTRL2)	32	RW	0000_0000h
A0h	Specifies the PLL timer register. (DSI_PLLTMR)	32	RW	FFFF_FFFFh
A4h	Specifies the D-PHY control register 1. (DSI_PHYCTRL_B1)	32	RW	0000_0000h
A8h	Specifies the D-PHY control register 2. (DSI_PHYCTRL_B2)	32	RW	0000_0000h
ACh	Specifies the D-PHY control register 1. (DSI_PHYCTRL_M1)	32	RW	0000_0000h
B0h	Specifies the D-PHY control register 2. (DSI_PHYCTRL_M2)	32	RW	0000_0000h
B4h	Specifies the D-PHY timing register. (DSI_PHYTIMING)	32	RW	0000_0000h
B8h	Specifies the D-PHY timing register 1. (DSI_PHYTIMING1)	32	RW	0000_0000h
BCh	Specifies the D-PHY timing register 2. (DSI_PHYTIMING2)	32	RW	0000_0000h

13.6.9.1.2 Specifies the DSI version register. (DSI_VERSION)

13.6.9.1.2.1 Offset

Register	Offset
DSI_VERSION	0h

13.6.9.1.2.2 Function

Specifies the DSI version register.

13.6.9.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Version															
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Version															
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

13.6.9.1.2.4 Fields

Field	Function
31-0 Version	Specifies the DSI version information

13.6.9.1.3 Specifies the status register. (DSI_STATUS)

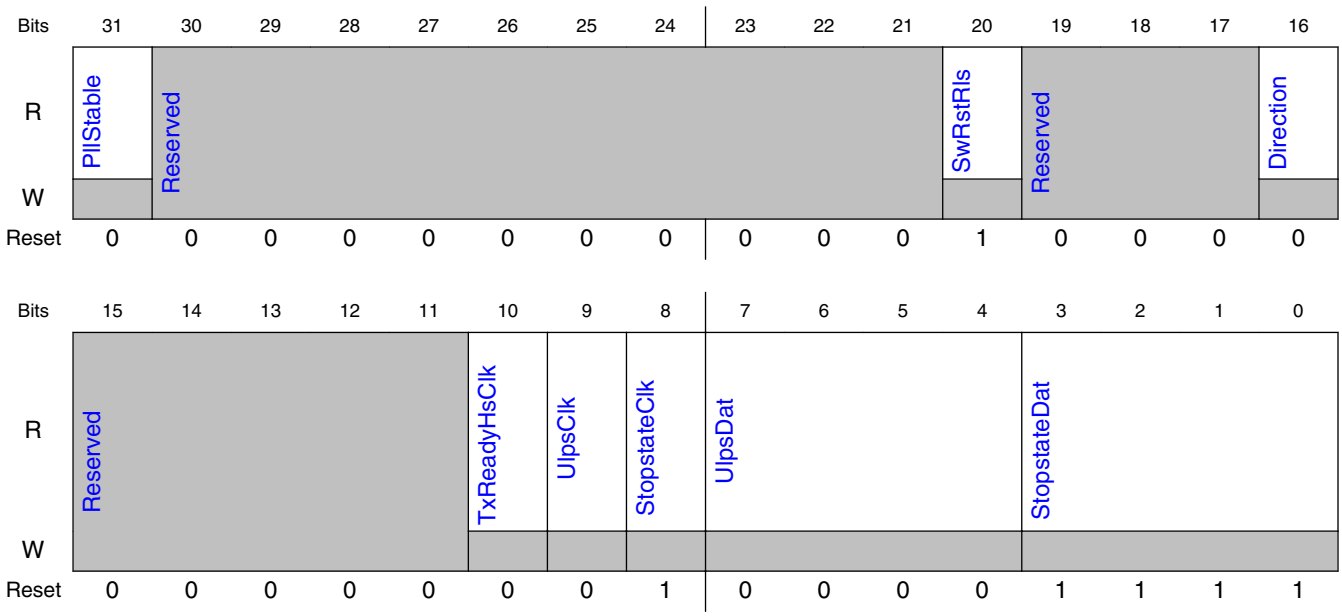
13.6.9.1.3.1 Offset

Register	Offset
DSI_STATUS	4h

13.6.9.1.3.2 Function

This register reads and checks internal and interface status. It also checks FSM status, Line buffer status, current image line number, and so on.

13.6.9.1.3.3 Diagram



13.6.9.1.3.4 Fields

Field	Function
31 PllStable	D-phy pll generates stable byteclk.
30-21 —	Reserved
20 SwRstRls	Specifies the software reset status. 0 = Reset state 1 = Release state
19-17 —	Reserved
16 Direction	Specifies the data direction indicator. 0 = Forward direction 1 = Backward direction
15-11 —	Reserved
10 TxReadyHsClk	Specifies the HS clock ready at clock lane. 0 = Not ready for transmitting HS data at clock lane 1 = Ready for transmitting HS data at clock lane
9 UlpsClk	Specifies the ULPS indicator at clock lane. 0 = No ULPS in clock lane

Table continues on the next page...

Field	Function
	1 = ULSP in clock lane
8 StopstateClk	Specifies the stop state indicator at clock lane. 0 = No stop state in clock lane 1 = Stop state in clock lane
7-4 UlpsDat	Specifies the ULPS indicator at data lanes. UlpsDat[0]: Data lane 0 UlpsDat[1]: Data lane 1 UlpsDat[2]: Data lane 2 UlpsDat[3]: Data lane 3 0 = No ULPS in each data lane 1 = ULPS in each data lane
3-0 StopstateDat	Specifies the stop state indicator at data lane. StopstateDat[0]: Data lane 0 StopstateDat[1]: Data lane 1 StopstateDat[2]: Data lane 2 StopstateDat[3]: Data lane 3 0 = No stop state in each data lane 1 = Stop state in each data lane

13.6.9.1.4 Specifies the RGB FSM status register. (DSI_RGB_STATUS)

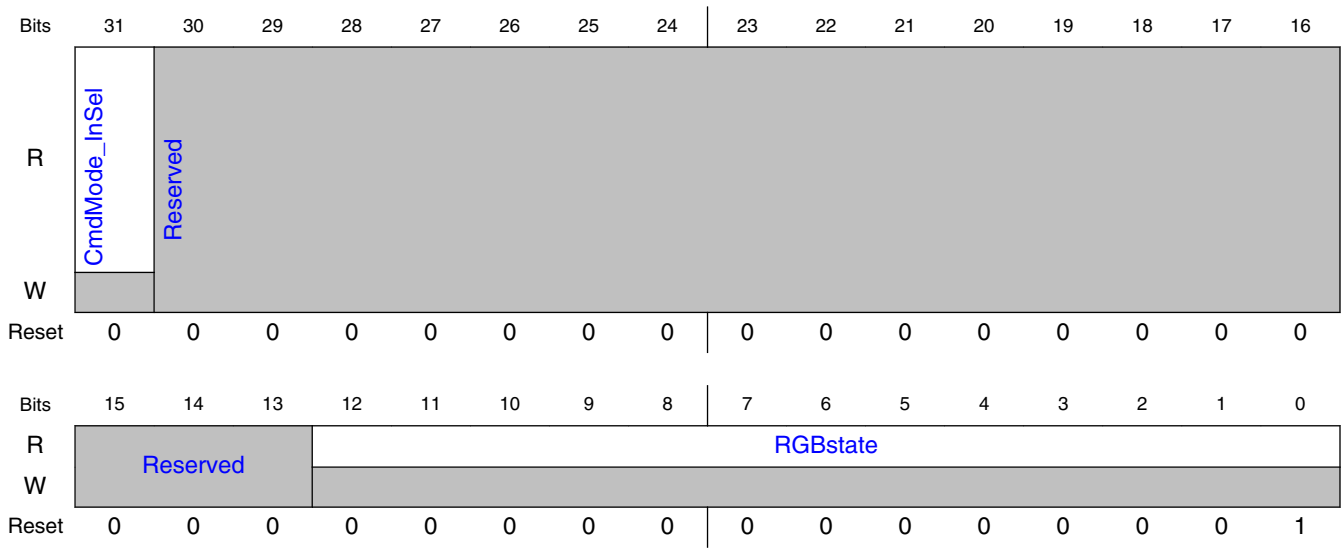
13.6.9.1.4.1 Offset

Register	Offset
DSI_RGB_STATUS	8h

13.6.9.1.4.2 Function

Specifies the RGB FSM status register.

13.6.9.1.4.3 Diagram



13.6.9.1.4.4 Fields

Field	Function
31 CmdMode_InSel	Specifies the command mode input selection 0 = using RGB video interface 1 = using S-i80 interface
30-13 —	Reserved
12-0 RGBstate	Specifies the RGB packetize FSM status. 0x0001 = IDLE 0x0002 = STOP 0x0004 = VSyS 0x0008 = VSE 0x0010 = HSS 0x0020 = HSA 0x0040 = HSE 0x0080 = HBP 0x0100 = RGB 0x0200 = NULL 0x0400 = HFP 0x0800 = EOT 0x1000 = NHOLD

13.6.9.1.5 Specifies the software reset register. (DSI_SWRST)

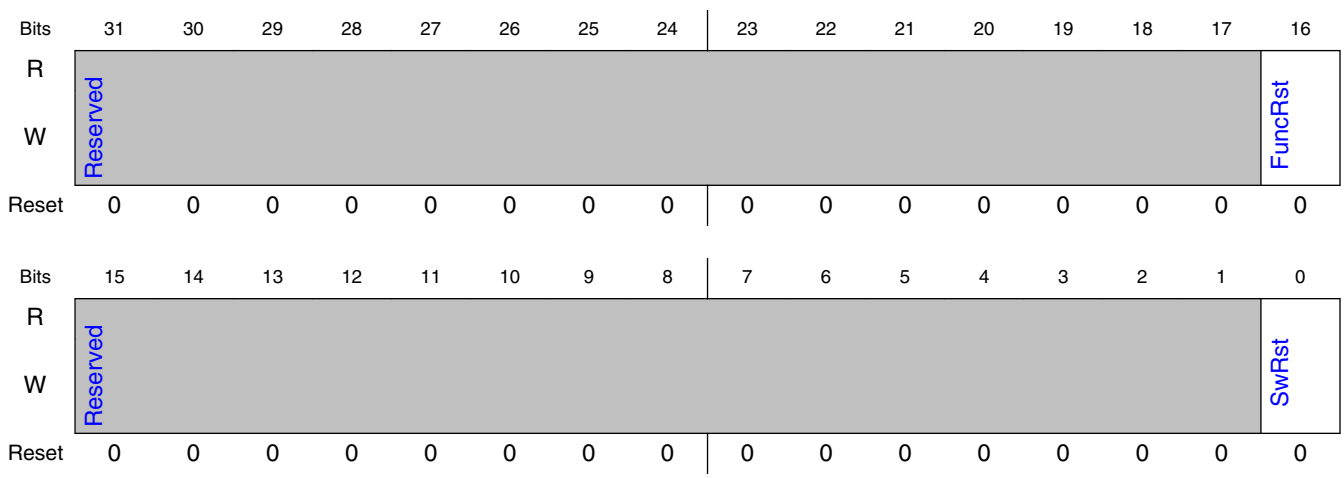
13.6.9.1.5.1 Offset

Register	Offset
DSI_SWRST	Ch

13.6.9.1.5.2 Function

Specifies the software reset register.

13.6.9.1.5.3 Diagram



13.6.9.1.5.4 Fields

Field	Function
31-17 —	Reserved
16 FuncRst	Specifies the software reset (High active). "Function reset" resets all FF in MIPI DSI (except SFRs: STATUS, RGB_STATUS, SWRST, CLKCTRL, TIMEOUT, CONFIG, ESCMODE[1], MDRESOL, MVPORCH, MHPORCH, MSYNC, INTMSK, SDRESOL, S3D_CTL, P3D_CTL, MIC_CTL, P3D_ON_MIC_OFF_HSA, P3D_OFF_MIC_ON_HSA, P3D_ON_MIC_ON_HSA, P3D_ON_MIC_OFF_HFP, P3D_OFF_MIC_ON_HFP, P3D_ON_MIC_ON_HFP, MULTI_PKT, FIFOTHLD, FIFOCTRL, [2], MEMACCHR, PLLCTRL, PLLTMR, PHYACCHR, and VERINFORM). 0 = Standby 1 = Reset
15-1	Reserved

Table continues on the next page...

Field	Function
—	
0 SwRst	Specifies the software reset (High active). “Software reset” resets all FF in MIPI DSI (except some SFRs: STATUS, SWRST, CLKCTRL, PLLCTRL, PLLTMR, PHYTUNE and VERINFORM). 0 = Standby 1 = Reset

1. ForceStopstate, CmdLpdt, TxLpdt
2. nInitRx, nInitSfr, nInitI80, nInitSub, nInitMD

13.6.9.1.6 Specifies the clock control register. (DSI_CLKCTRL)

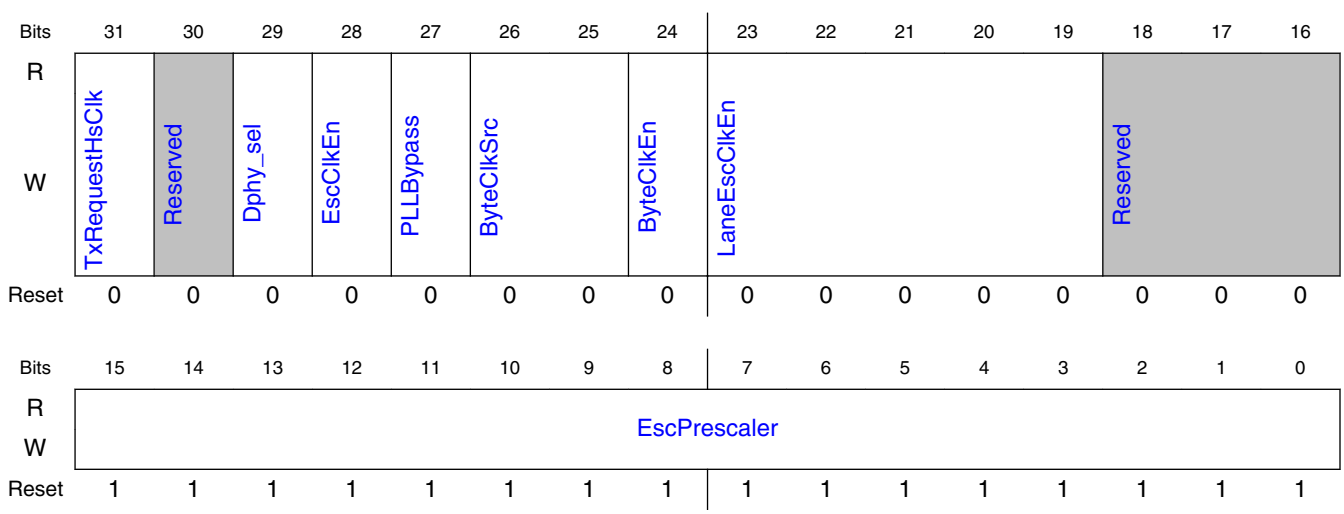
13.6.9.1.6.1 Offset

Register	Offset
DSI_CLKCTRL	10h

13.6.9.1.6.2 Function

Specifies the clock control register.

13.6.9.1.6.3 Diagram



13.6.9.1.6.4 Fields

Field	Function
31 TxRequestHsClk	Specifies the HS clock request for HS transfer at clock lane (Turn on HS clock)
30 —	Reserved
29 Dphy_sel	D-PHY select 0 = 1.5Gbps D-PHY (default) 1 = 1Gbps D-PHY
28 EscClkEn	Enables the escape clock generating prescaler. 0 = Disables 1 = Enables
27 PLLBypass	Sets the PLLBypass signal connected to D-PHY module input for selecting clock source bit. 0 = PLL output 1 = External Serial clock This bit must be set to 0.
26-25 ByteClkSrc	Selects byte clock source. (It must be 2'b00.) 00 = D-PHY PLL (default). PLL_out clock is used to generate ByteClk by dividing 8.
24 ByteClkEn	Enables byte clock. 0 = Disables 1 = Enables
23-19 LaneEscClkEn	Enables escape clock for D-phy lane. LaneEscClkEn[0] = Clock lane LaneEscClkEn[1] = Data lane 0 LaneEscClkEn[2] = Data lane 1 LaneEscClkEn[3] = Data lane 2 LaneEscClkEn[4] = Data lane 3 0 = Disables 1 = Enables
18-16 —	Reserved
15-0 EscPrescaler	Specifies the escape clock prescaler value. The escape clock frequency range varies up to 20MHz. **NOTE:** The requirement for BTA is that the Host Escclk frequency should range between 66.7 ~ 150% of the peripheral escape clock frequency. EscClk = ByteClk / (EscPrescaler)

13.6.9.1.7 Specifies the time out register. (DSI_TIMEOUT)

13.6.9.1.7.1 Offset

Register	Offset
DSI_TIMEOUT	14h

13.6.9.1.7.2 Function

Specifies the time out register.

13.6.9.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								BtaTout							
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LpdrTout															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

13.6.9.1.7.4 Fields

Field	Function
31-24 —	Reserved
23-16 BtaTout	Specifies the timer for BTA. This register specifies time out from BTA request to change the direction with respect to Tx escape clock.
15-0 LpdrTout	Specifies the timer for LP Rx mode timeout. This register specifies time out on how long RxValid deasserts, after RxLpdt asserts with respect to Tx escape clock. RxValid specifies Rx data valid indicator. RxLpdt specifies an indicator that D-phy is under RxLpdt mode. RxValid and RxLpdt specifies signal from D-phy.

13.6.9.1.8 Specifies the configuration register. (DSI_CONFIG)

13.6.9.1.8.1 Offset

Register	Offset
DSI_CONFIG	18h

13.6.9.1.8.2 Function

This register configures MIPI DSI master such as data lane number, input interface, porch area, frame rate, BTA, LPDT, ULPS, and so on.

The figure below shows the timing diagram of the clock lane:

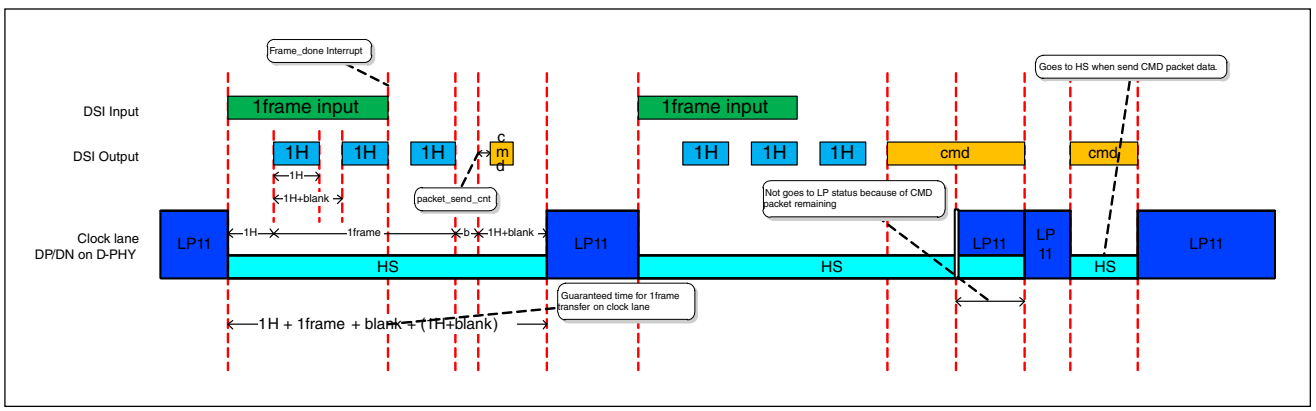
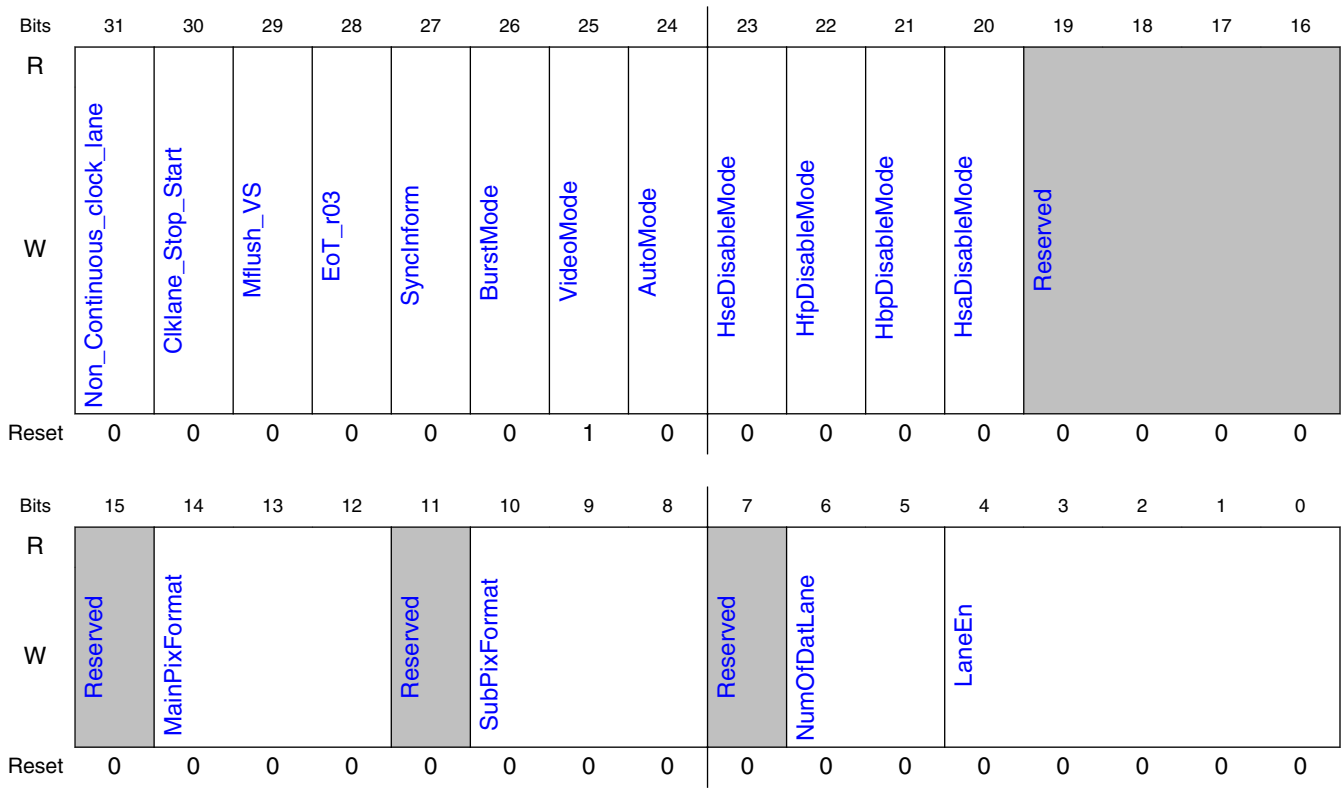


Figure 13-52. Non-continuous clock lane timing diagram

13.6.9.1.8.3 Diagram



13.6.9.1.8.4 Fields

Field	Function
31 Non_Continuous_clock_lane	Non-continuous clock mode. This feature is D-PHY clock lane control DP/DN Please refer to the diagram above to view the clock lane timing diagram. Please refer to the DSI_MULTI_PKT register, which shows the configuration of count values. 0 = Disable (default) 1 = Enable
30 Clklane_Stop_Start	PHY clock lane On/Off for Electrostatic Discharge (ESD). It is operated when first HBP to second HFP in every VSYNC. This feature is not guaranteed on Command mode. 0 = Disable (default) 1 = Enable
29 Mflush_VS	Auto flush of MD FIFO using Vsync pulse. It needs that Main display FIFO should be flushed for deleting garbage data. 0 = Enable (default) 1 = Disable

Table continues on the next page...

Field	Function
28 EoT_r03	Disables EoT packet in HS mode. 0 = Enables EoT packet generation for V1.01r11 1 = Disables EoT packet generation for V1.01r03
27 SyncInform	Selects Sync Pulse or Event mode in Video mode. 0 = Event mode (non burst, burst) 1 = Pulse mode (non burst only) In command mode, this bit is ignored.
26 BurstMode	Selects Burst mode in Video mode In Non-burst mode, RGB data area is filled with RGB data and Null packets, according to input bandwidth of RGB interface. In Burst mode, RGB data area is filled with RGB data only. 0 = Non-burst mode 1 = Burst mode In command mode, this bit is ignored.
25 VideoMode	Specifies display configuration. 0 = Command mode 1 = Video mode
24 AutoMode	Specifies auto vertical count mode. In Video mode, the vertical line transition uses line counter configured by Vertical Sync Active (VSA), VBP, and Vertical resolution. If this bit is set to '1', the line counter does not use VSA and VBP registers. 0 = Configuration mode 1 = Auto mode In command mode, this bit is ignored.
23 HseDisableMode	In Vsync pulse and Vporch area, MIPI DSI master transfers only Hsync start packet to MIPI DSI slave at MIPI DSI spec 1.1r02. This bit transfers Hsync end packet in Vsync pulse and Vporch area (optional). 0 = Disables transfer 1 = Enables transfer In command mode, this bit is ignored.
22 HfpDisableMode	Specifies HFP disable mode. If this bit set, DSI master ignores HFP area in Video mode. 0 = Enables 1 = Disables In command mode, this bit is ignored.
21 HbpDisableMode	Specifies HBP disable mode. If this bit set, DSI master ignores HBP area in Video mode. 0 = Enables 1 = Disables In command mode, this bit is ignored.
20 HsaDisableMode	Specifies HSA disable mode. If this bit set, DSI master ignores HSA area in Video mode. 0 = Enables 1 = Disables In command mode, this bit is ignored.

Table continues on the next page...

Field	Function
19-15 —	Reserved
14-12 MainPixFormat	Specifies pixel stream format for main display. 000 = 3bpp (for Command mode only) 001 = 8bpp (for Command mode only) 010 = 12bpp (for Command mode only) 011 = 16bpp (for Command mode only) 100 = 16-bit RGB (565) (for Video mode only) 101 = 18-bit RGB (666: packed pixel stream) (for Video mode only) 110 = 18-bit RGB (666: loosely packed pixel stream) for Common 111 = 24-bit RGB (888) for common
11 —	Reserved
10-8 SubPixFormat	Specifies pixel stream format for sub display. 000 = 3bpp (for Command mode only) 001 = 8bpp (for Command mode only) 010 = 12bpp (for Command mode only) 011 = 16bpp (for Command mode only) 100 = 16-bit RGB (565) (for Video mode only) 101 = 18-bit RGB (666: packed pixel stream) for Video mode only) 110 = 18-bit RGB (666: loosely packed pixel stream) for common) 111 = 24-bit RGB (888) (for Common)
7 —	Reserved
6-5 NumOfDatLane	Sets the data lane number. 00 = Data lane 0 (1 data lane) 01 = Data lane 0 ~ 1 (2 data lanes) 10 = Data lane 0 ~ 2 (3 data lanes) 11 = Data lane 0 ~ 3 (4 data lanes)
4-0 LaneEn	Enables the lane. If Lane_EN is disabled, the lane ignores input and drives initial value through output port. 0 = Lane is off. 1 = Lane is on. LaneEn[0] = Clock lane enabler LaneEn[1] = Data lane 0 enabler LaneEn[2] = Data lane 1 enabler LaneEn[3] = Data lane 2 enabler LaneEn[4] = Data lane 3 enabler

13.6.9.1.9 Specifies the escape mode register. (DSI_ESCMODE)

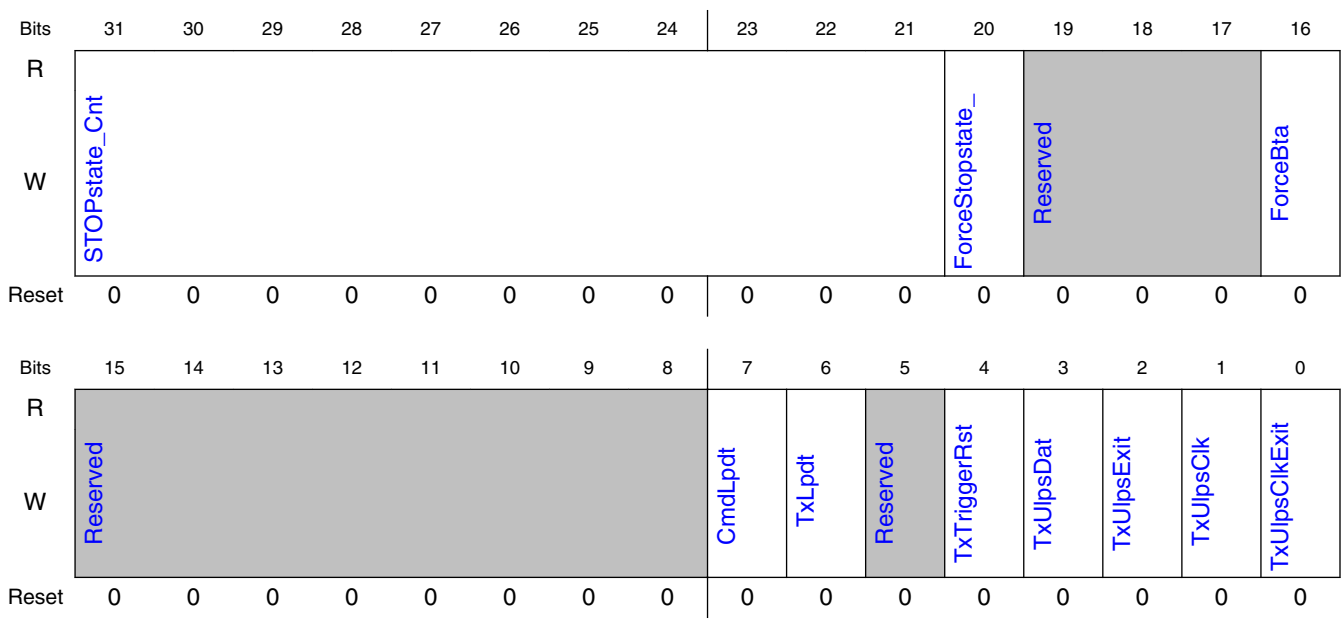
13.6.9.1.9.1 Offset

Register	Offset
DSI_ESCMODE	1Ch

13.6.9.1.9.2 Function

This register configures MIPI DSI master.

13.6.9.1.9.3 Diagram



13.6.9.1.9.4 Fields

Field	Function
31-21 STOPstate_Cnt	After transmitting read packet or write “set_tear_on” command, BTA requests to D-phy automatically. This counter value specifies the interval value between transmitting read packet (or write “set_tear_on” command) and BTA request. 11'h000 = 2 EscClk 11'h001 = 2 EscClk + 1 EscClk 11'h3FF = 2 EscClk + 1023 EscClk
20	Forces Stopstate for D-PHY.

Table continues on the next page...

Field	Function
ForceStopstate_	
19-17 —	Reserved
16 ForceBta	Forces Bus Turn Around. 1 = Sends the protocol layer request to D-PHY. MIPI DSI peripheral becomes master after BTA sequence. This bit clears automatically after receiving BTA acknowledge from MIPI DSI peripheral.
15-8 —	Reserved
7 CmdLpdt	Specifies LPDT transfers command in SFR FIFO. 0 = HS Mode 1 = LP Mode
6 TxLpdt	Specifies data transmission in LP mode (all data transfer in LPDT). 0 = HS Mode 1 = LP Mode
5 —	Reserved
4 TxTriggerRst	Specifies remote reset trigger function. After trigger operation, these bits will be cleared automatically.
3 TxUlpsDat	Specifies ULPS request for data lane. Manually clears after ULPS exit.
2 TxUlpsExit	Specifies ULPS exit request for data lane. Manually clears after ULPS exit.
1 TxUlpsClk	Specifies ULPS request for clock lane. Manually clears after ULPS exit.
0 TxUlpsClkExit	Specifies ULPS exit request for clock lane. Manually clears after ULPS exit.

13.6.9.1.10 Specifies the main display image resolution register. (DSI_MDRESOL)

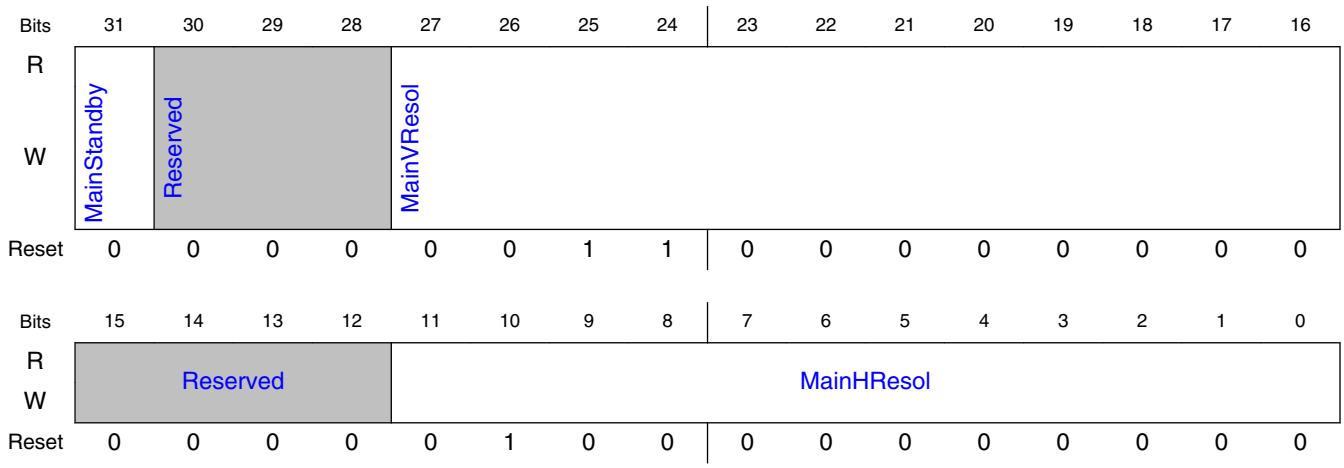
13.6.9.1.10.1 Offset

Register	Offset
DSI_MDRESOL	20h

13.6.9.1.10.2 Function

Specifies the main display image resolution register.

13.6.9.1.10.3 Diagram



13.6.9.1.10.4 Fields

Field	Function
31 MainStandby	Specifies standby for receiving DISPCON output in Command mode after setting all configuration. 0 = Not ready 1 = Stand by Standby should be set after configuration (resolution, reqtype, pixelform, and so on) is set for command mode. In Video mode, if this bit value is 0, data is not transferred.
30-28 —	Reserved
27-16 MainVResol	Specifies Vertical resolution (1 ~ 2047).
15-12 —	Reserved
11-0 MainHResol	Specifies Horizontal resolution (1 ~ 2047).

13.6.9.1.11 Specifies the main display Vporch register. (DSI_MVPORCH)

13.6.9.1.11.1 Offset

Register	Offset
DSI_MVPORCH	24h

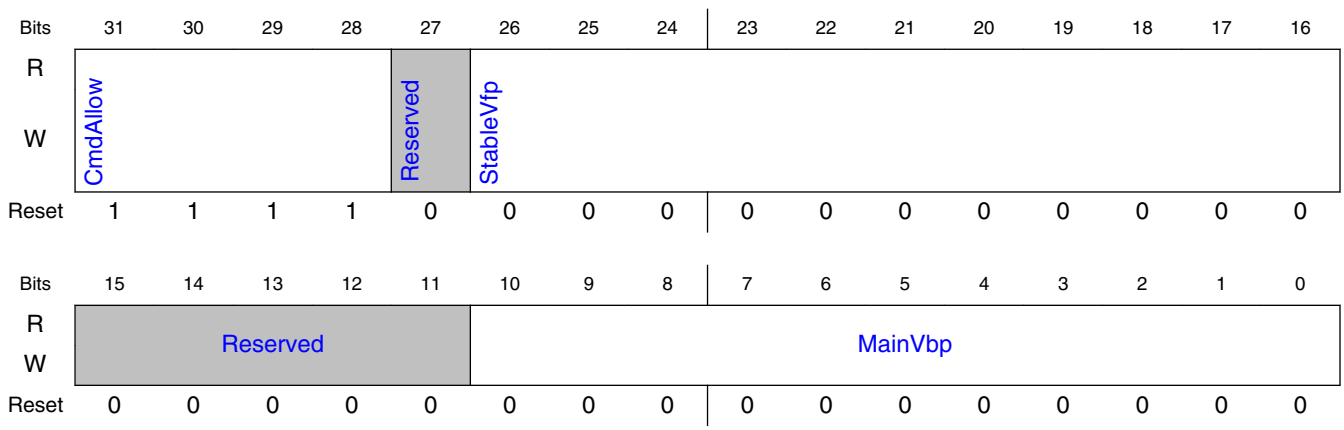
13.6.9.1.11.2 Function

Specifies the main display Vporch register.

NOTE

Transfers command packets after Stable VFP area. Display controller VFP lines should be set based on sum of these values: Stable VFP, command allowing area and command masked area. Please refer to Transfer general data in video mode for more information

13.6.9.1.11.3 Diagram



13.6.9.1.11.4 Fields

Field	Function
31-28 CmdAllow	Specifies the number of horizontal lines, where command packet transmission is allowed after Stable VFP period. For more information, please see Figure 13-42.
27 —	Reserved
26-16 StableVfp	Specifies the number of horizontal lines, where command packet transmission is not allowed after end of active region. For more information, please see Figure 13-42. **NOTE:** In Command mode, these bits are ignored.
15-11 —	Reserved
10-0 MainVbp	Specifies vertical back porch width for Video mode (line count). In Command mode, these bits are ignored.

13.6.9.1.12 Specifies the main display Hporch register. (DSI_MHPORCH)

13.6.9.1.12.1 Offset

Register	Offset
DSI_MHPORCH	28h

13.6.9.1.12.2 Function

Specifies the main display Hporch register.

13.6.9.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MainHfp															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MainHbp															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.9.1.12.4 Fields

Field	Function
31-16 MainHfp	Specifies the horizontal front porch width for Video mode. HFP is specified using blank packet. These bits specify the word counts for blank packet in HFP. In Command mode, these bits are ignored.
15-0 MainHbp	Specifies the horizontal back porch width for Video mode. HBP is specified using blank packet. These bits specify the word counts for blank packet in HBP. In Command mode, these bits are ignored.

13.6.9.1.13 Specifies the main display Sync Area register. (DSI_MSYNC)

13.6.9.1.13.1 Offset

Register	Offset
DSI_MSYNC	2Ch

13.6.9.1.13.2 Function

Specifies the main display Sync Area register.

13.6.9.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MainVsa										Reserved					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MainHsa															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.9.1.13.4 Fields

Field	Function
31-22 MainVsa	Specifies the vertical sync pulse width for Video mode (Line count). In command mode, these bits are ignored.
21-16 —	Reserved
15-0 MainHsa	Specifies the horizontal sync pulse width for Video mode. HSA is specified using blank packet. These bits specify word counts for blank packet in HSA. In command mode, these bits are ignored.

13.6.9.1.14 Specifies the sub display image resolution register. (DSI_SDRESOL)

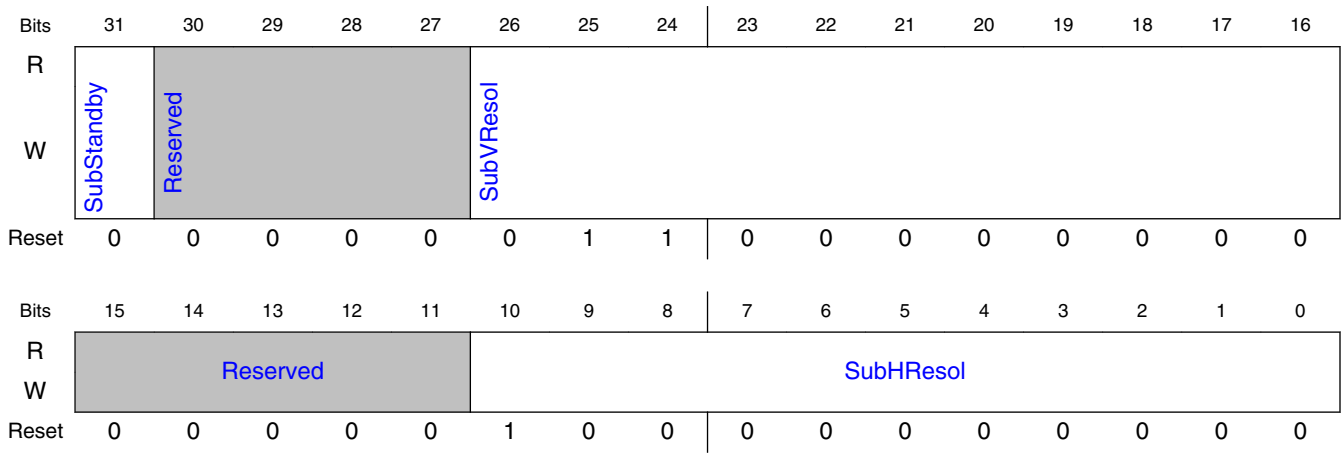
13.6.9.1.14.1 Offset

Register	Offset
DSI_SDRESOL	30h

13.6.9.1.14.2 Function

Specifies the sub display image resolution register.

13.6.9.1.14.3 Diagram



13.6.9.1.14.4 Fields

Field	Function
31 SubStandby	Specifies standby for receiving DISPCON output in Command mode after setting all configuration. 0 = Not ready 1 = Standby Standby should be set after configuration (resolution, reqtype, pixelform, and so on) is set for command mode. In Video mode, this bit is ignored.
30-27 —	Reserved
26-16 SubVResol	Specifies the Vertical resolution (1 ~ 1024).
15-11 —	Reserved
10-0 SubHResol	Specifies the Horizontal resolution (1 ~ 1024).

13.6.9.1.15 Specifies the interrupt source register. (DSI_INTSRC)

13.6.9.1.15.1 Offset

Register	Offset
DSI_INTSRC	34h

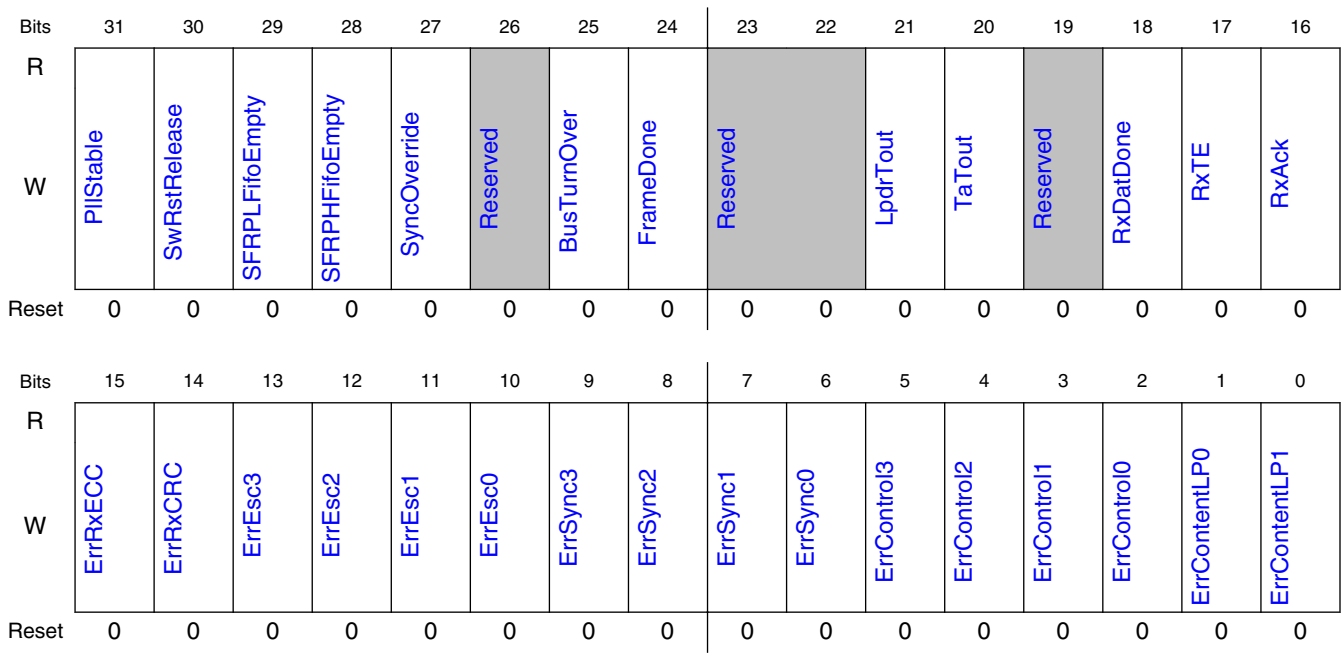
13.6.9.1.15.2 Function

This register identifies interrupt sources. Internal block error, data transmit interrupt, inter-layer (D_PHY) error, etc.

The bits are set even if they are masked off by DSI_INTMSK_REG.

Write '1' to clear the Interrupt.

13.6.9.1.15.3 Diagram



13.6.9.1.15.4 Fields

Field	Function
31 PllStable	Indicates that D-phy PLL is stable.
30 SwRstRelease	Releases the software reset.
29 SFRPLFifoEmpty	Specifies the SFR payload FIFO empty.
28 SFRPHFifoEmpty	Specifies the SFR Packet Header FIFO empty

Table continues on the next page...

Field	Function
27 SyncOverride	Indicates that other DSI command transfer have overridden sync timing.
26 —	Reserved
25 BusTurnOver	Indicates when bus grant turns over from DSI slave to DSI master.
24 FrameDone	Indicates when MIPI DSI transfers the whole image frame. **NOTE:** If Hsync is not received during two line times, internal timer is timed out and this bit is flagged.
23-22 —	Reserved
21 LpdrTout	Specifies the LP Rx timeout. Please refer to the DSI_TIMEOUT register.
20 TaTout	Turns around Acknowledge Timeout. Please refer to the DSI_TIMEOUT register.
19 —	Reserved
18 RxDatDone	Completes receiving data.
17 RxTE	Receives TE Rx trigger.
16 RxAck	Receives ACK Rx trigger.
15 ErrRxECC	Specifies the ECC multi bit error in LPDR.
14 ErrRxCRC	Specifies the CRC error in LPDR.
13 ErrEsc3	Specifies the escape mode entry error lane 3. For more information, please refer to the standard D-PHY specification.
12 ErrEsc2	Specifies the escape mode entry error lane 2. For more information, please refer to the standard D-PHY specification.
11 ErrEsc1	Specifies the escape mode entry error lane 1. For more information, please refer to the standard D-PHY specification.
10 ErrEsc0	Specifies the escape mode entry error lane 0. For more information, please refer to the standard D-PHY specification.
9 ErrSync3	Specifies the LPDT sync error lane 3. For more information, please refer to the standard D-PHY specification.
8 ErrSync2	Specifies the LPDT Sync Error lane2. For more information, refer to standard D-PHY specification.
7	Specifies the LPDT Sync Error lane1. For more information, refer to standard D-PHY specification.

Table continues on the next page...

Field	Function
ErrSync1	
6 ErrSync0	Specifies the LPDT Sync Error lane0. For more information, refer to standard D-PHY specification.
5 ErrControl3	Controls Error lane3. For more information, please refer to the standard D-PHY specification.
4 ErrControl2	Controls Error lane2. For more information, please refer to the standard D-PHY specification.
3 ErrControl1	Controls Error lane1. For more information, please refer to the standard D-PHY specification.
2 ErrControl0	Controls Error lane0. For more information, please refer to the standard D-PHY specification.
1 ErrContentLP0	Specifies the LP0 Contention Error (only lane0, because BTA occurs at lane0 only). For more information, please refer to the standard D-PHY specification.
0 ErrContentLP1	Specifies the LP1 Contention Error (only lane0, because BTA occurs at lane0 only). For more information, please refer to the standard D-PHY specification.

13.6.9.1.16 Specifies the interrupt mask register. (DSI_INTMSK)

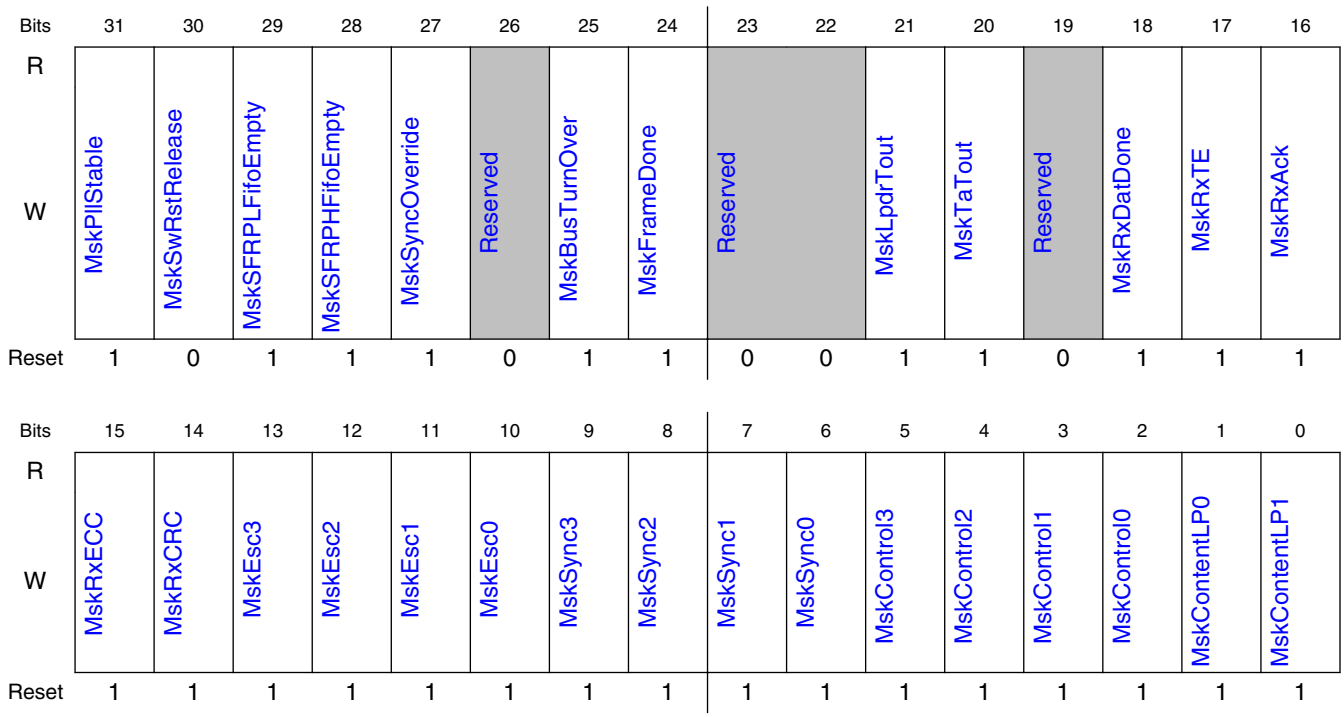
13.6.9.1.16.1 Offset

Register	Offset
DSI_INTMSK	38h

13.6.9.1.16.2 Function

This register masks interrupt sources.

13.6.9.1.16.3 Diagram



13.6.9.1.16.4 Fields

Field	Function
31 MskPllStable	Indicates that D-PHY PLL is stable.
30 MskSwRstRelease	Releases software reset.
29 MskSFRPLFifoEmpty	Empties SFR payload FIFO.
28 MskSFRPHFifoEmpty	Interrupt Mask for SFR packet header FIFO empty
27 MskSyncOverride	Indicates that other DSI command transfer have overridden sync timing.
26 —	Reserved
25 MskBusTurnOver	Indicates when bus grant turns over from DSI slave to DSI master.

Table continues on the next page...

Field	Function
24 MskFrameDone	Indicates when MIPI DSI transfers whole image frame.
23-22 —	Reserved
21 MskLpdrTout	Specifies LP Rx timeout. Please refer to the DSI_TIMEOUT register.
20 MskTaTout	Specifies turnaround acknowledge timeout. Please refer to the DSI_TIMEOUT register.
19 —	Reserved
18 MskRxDatDone	Specifies completion of data receiving.
17 MskRxTE	Specifies receipt of TE Rx trigger.
16 MskRxAck	Specifies receipt of ACK Rx trigger.
15 MskRxECC	Specifies ECC multibit error in LPDR.
14 MskRxCRC	Specifies CRC error in LPDR.
13 MskEsc3	Specifies escape mode entry error in lane3. For more information, please refer to the standard D-PHY specification.
12 MskEsc2	Specifies escape mode entry error in lane2. For more information, please refer to the standard D-PHY specification.
11 MskEsc1	Specifies escape mode entry error in lane1. For more information, please refer to the standard D-PHY specification.
10 MskEsc0	Specifies escape mode entry error in lane0. For more information, please refer to the standard D-PHY specification.
9 MskSync3	Specifies LPDT sync error in lane3. For more information, refer to standard D-PHY specification.
8 MskSync2	Specifies LPDT sync error in lane2. For more information, refer to standard D-PHY specification.
7 MskSync1	Specifies LPDT sync error in lane1. For more information, refer to standard D-PHY specification.
6 MskSync0	Specifies LPDT sync error in lane0. For more information, refer to standard D-PHY specification.
5 MskControl3	Controls error in lane3. For more information, please refer to the standard D-PHY specification.
4	Controls error in lane2. For more information, please refer to the standard D-PHY specification.

Table continues on the next page...

Field	Function
MskControl2	
3 MskControl1	Controls error in lane1. For more information, please refer to the standard D-PHY specification.
2 MskControl0	Controls error in lane0. For more information, please refer to the standard D-PHY specification.
1 MskContentLP0	Specifies LP0 contention error. For more information, refer to standard D-PHY specification.
0 MskContentLP1	Specifies LP1 contention error. For more information, refer to standard D-PHY specification.

13.6.9.1.17 Specifies the packet header FIFO register. (DSI_PKTHDR)

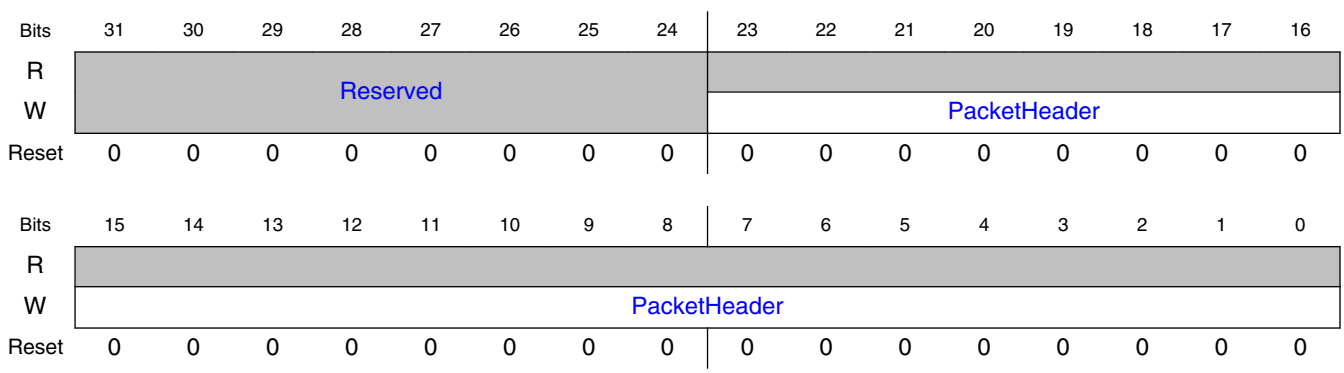
13.6.9.1.17.1 Offset

Register	Offset
DSI_PKTHDR	3Ch

13.6.9.1.17.2 Function

This register is the FIFO for packet header to send DSI packets.

13.6.9.1.17.3 Diagram



13.6.9.1.17.4 Fields

Field	Function
31-24	Reserved

Table continues on the next page...

Field	Function
—	
23-0 PacketHeader	Writes the packet header of Tx packet. [7:0] = DI [15:8] = Dat0 (Word Count lower byte for long packet) [23:16] = Dat1 (Word Count upper byte for long packet)

13.6.9.1.18 Specifies the payload FIFO register. (DSI_PAYLOAD)

13.6.9.1.18.1 Offset

Register	Offset
DSI_PAYLOAD	40h

13.6.9.1.18.2 Function

This register specifies the FIFO for payload to send DSI packets.

13.6.9.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	Payload															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	Payload															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.9.1.18.4 Fields

Field	Function
31-0 Payload	Writes the Payload of Tx packet.

13.6.9.1.19 Specifies the read FIFO register. (DSI_RXFIFO)

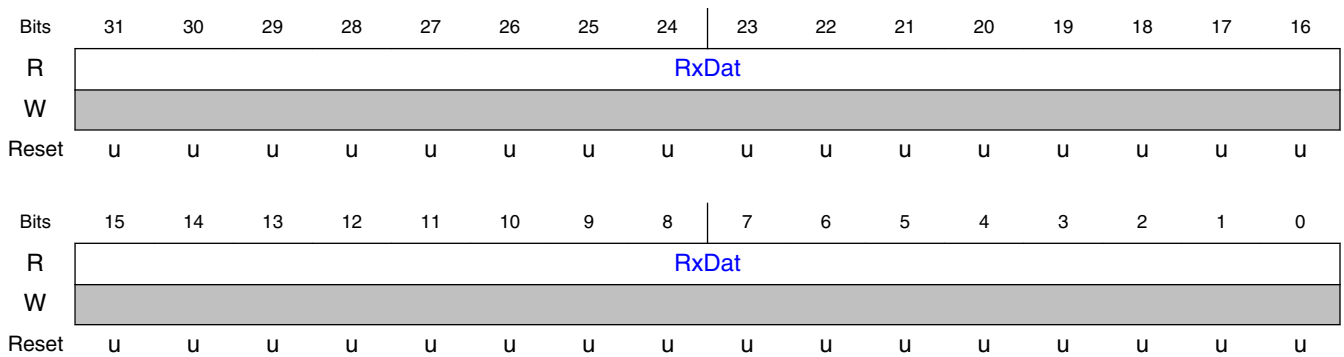
13.6.9.1.19.1 Offset

Register	Offset
DSI_RXFIFO	44h

13.6.9.1.19.2 Function

This register is the gate of FIFO read.

13.6.9.1.19.3 Diagram



13.6.9.1.19.4 Fields

Field	Function
31-0 RxDat	In the Rx mode, you can read Rx data through this register. **NOTE:** The CRC in packet is not stored in RxFIFO.

13.6.9.1.20 Specifies the FIFO threshold level register. (DSI_FIFOTHLD)

13.6.9.1.20.1 Offset

Register	Offset
DSI_FIFOTHLD	48h

13.6.9.1.20.2 Function

Specifies the FIFO threshold level register.

13.6.9.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							WfullLevelSfr								
Reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

13.6.9.1.20.4 Fields

Field	Function
31-9 —	Reserved
8-0 WfullLevelSfr	Almost full level of SFR payload FIFO

13.6.9.1.21 Specifies the FIFO status and control register. (DSI_FIFOCTRL)

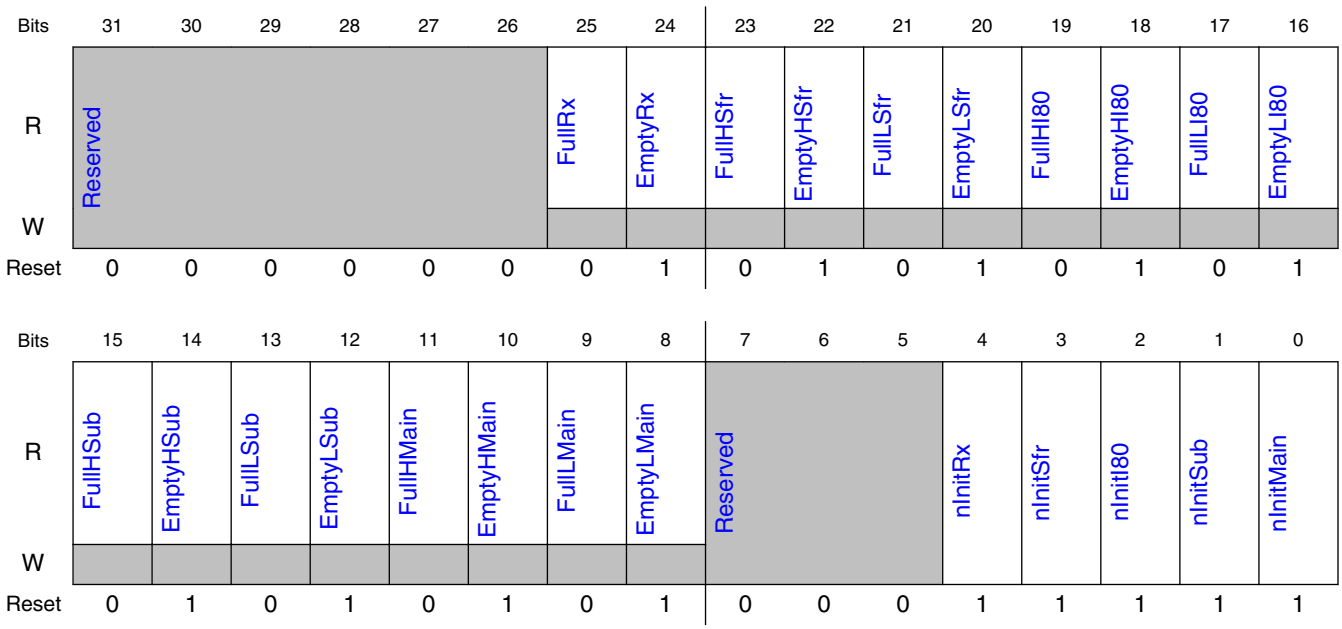
13.6.9.1.21.1 Offset

Register	Offset
DSI_FIFOCTRL	4Ch

13.6.9.1.21.2 Function

Specifies the FIFO status and control register.

13.6.9.1.21.3 Diagram



13.6.9.1.21.4 Fields

Field	Function
31-26 —	Reserved
25 FullRx	Rx FIFO full
24 EmptyRx	Rx FIFO empty
23 FullHSfr	SFR packet header FIFO full
22 EmptyHSfr	SFR packet header FIFO empty
21 FullLSfr	SFR payload FIFO full
20 EmptyLSfr	SFR payload FIFO empty
19 FullHI80	S-i80 packet header FIFO full
18 EmptyHI80	S-i80 packet header FIFO empty
17	S-i80 payload FIFO full

Table continues on the next page...

Field	Function
FullLI80	
16 EmptyLI80	S-i80 payload FIFO empty
15 FullHSub	Sub display packet header FIFO full
14 EmptyHSub	Sub display packet header FIFO empty
13 FullLSub	Sub display payload FIFO full
12 EmptyLSub	Sub display payload FIFO empty
11 FullHMain	Main display packet header FIFO full
10 EmptyHMain	Main display packet header FIFO empty
9 FullLMain	Main display payload FIFO full
8 EmptyLMain	Main display payload FIFO empty
7-5 —	Reserved
4 nInitRx	MD FIFO read point initialize
3 nInitSfr	SFR FIFO write point initialize
2 nInitI80	S-i80 FIFO write point initialize
1 nInitSub	SD FIFO write point initialize
0 nInitMain	MD FIFO write point initialize

13.6.9.1.22 Specifies the FIFO memory AC characteristic register. (DSI_MEMACCHR)

13.6.9.1.22.1 Offset

Register	Offset
DSI_MEMACCHR	50h

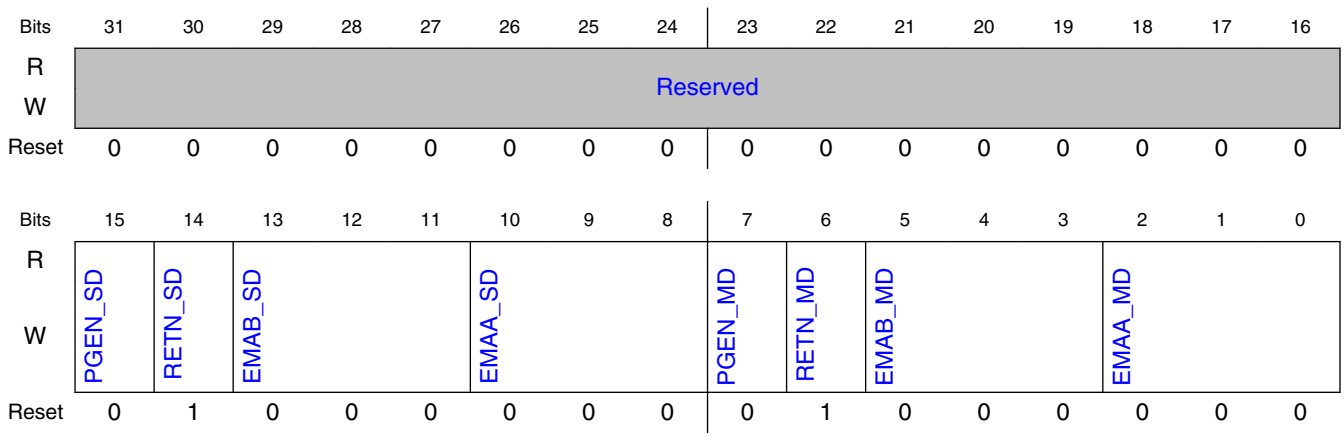
13.6.9.1.22.2 Function

Specifies the FIFO memory AC characteristic register.

NOTE

In the current design, these memory port controls were disabled. User can ignore the function of this register.

13.6.9.1.22.3 Diagram



13.6.9.1.22.4 Fields

Field	Function
31-16 —	Reserved
15 PGEN_SD	Sub display FIFO memory power gating
14 RETN_SD	Sub display FIFO memory Retention
13-11 EMAB_SD	Sub display FIFO memory B port margin adjustment
10-8 EMAA_SD	Sub display FIFO memory A port margin adjustment
7 PGEN_MD	Main display FIFO memory power gating
6 RETN_MD	Main display FIFO memory Retention
5-3	Main display FIFO memory B port margin adjustment

Table continues on the next page...

Field	Function
EMAB_MD	
2-0 EMAA_MD	Main display FIFO memory A port margin adjustment

13.6.9.1.23 Specifies the Multi Packet, Packet Go register. (DSI_MULTI_PKT)

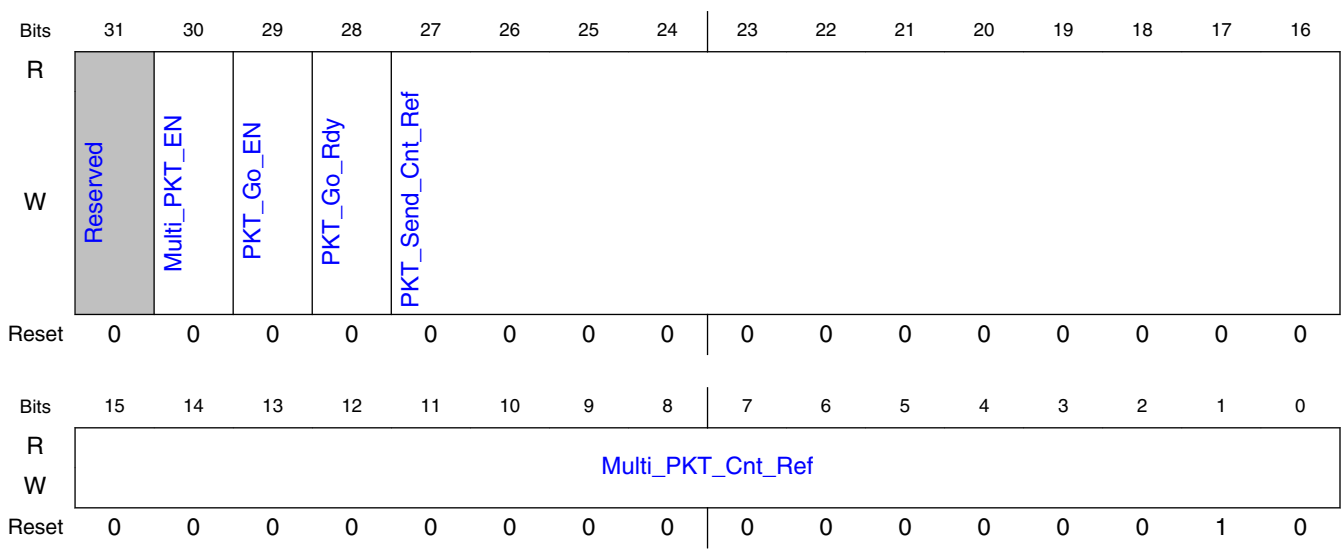
13.6.9.1.23.1 Offset

Register	Offset
DSI_MULTI_PKT	78h

13.6.9.1.23.2 Function

Specifies the Multi Packet, Packet Go register.

13.6.9.1.23.3 Diagram



13.6.9.1.23.4 Fields

Field	Function
31 —	Reserved

Table continues on the next page...

Field	Function
30 Multi_PKT_EN	Specifies the send multi command packets on single transmission. Multi command packet can transfer by only HS mode. ‘0’ = Send single command packet. ‘1’ = Send multi command packets.
29 PKT_Go_EN	Specifies the send command packet(s) per frame enable Packet go can transfer by only HS mode. ‘0’ = Send command packet(s) during every VFP. ‘1’ = Send command packet(s) during 1frame VFP.
28 PKT_Go_Rdy	Specifies the send command packet(s) on this frame VFP. If packet transferred or PKT_Go_EN is disabled, this bit goes to 1'b0.
27-16 PKT_Send_Cnt_Ref	Specifies the command packet(s) send point indicator. That value is 1H time + D-PHY LP-HS-LP latency by ByteCLK. Do not use '0'.
15-0 Multi_PKT_Cnt_Ref	Specifies the number of packets on single transmission. Do not use '0' and '1'. 0 and 1 are same as '2'

13.6.9.1.24 Specifies the 1Gbps D-PHY PLL control register. (DSI_PLLCTRL_1G)

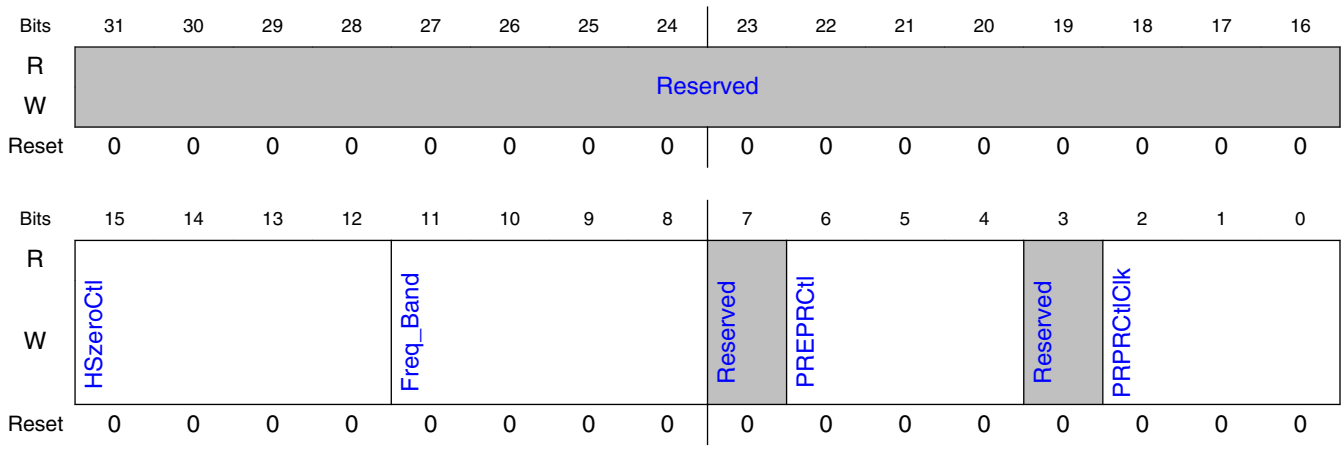
13.6.9.1.24.1 Offset

Register	Offset
DSI_PLLCTRL_1G	90h

13.6.9.1.24.2 Function

This register configures 1Gbps D-PHY PLL control, D-PHY, clock range indication, and so on.

13.6.9.1.24.3 Diagram



13.6.9.1.24.4 Fields

Field	Function
31-16 —	Reserved
15-12 HSzeroCtl	1Gbps D-PHY HS-Zero driving timing control.
11-8 Freq_Band	1Gbps D-PHY Timing control for D-PHY global operation timing.
7 —	Reserved
6-4 PREPRCtl	1Gbps D-PHY PLL Tclk-prepare and Ths-prepare driving control.
3 —	Reserved
2-0 PRPRCtlClk	1Gbps D-PHY PLL Ths-prepare driving time control.

13.6.9.1.25 Specifies the PLL control register. (DSI_PLLCTRL)

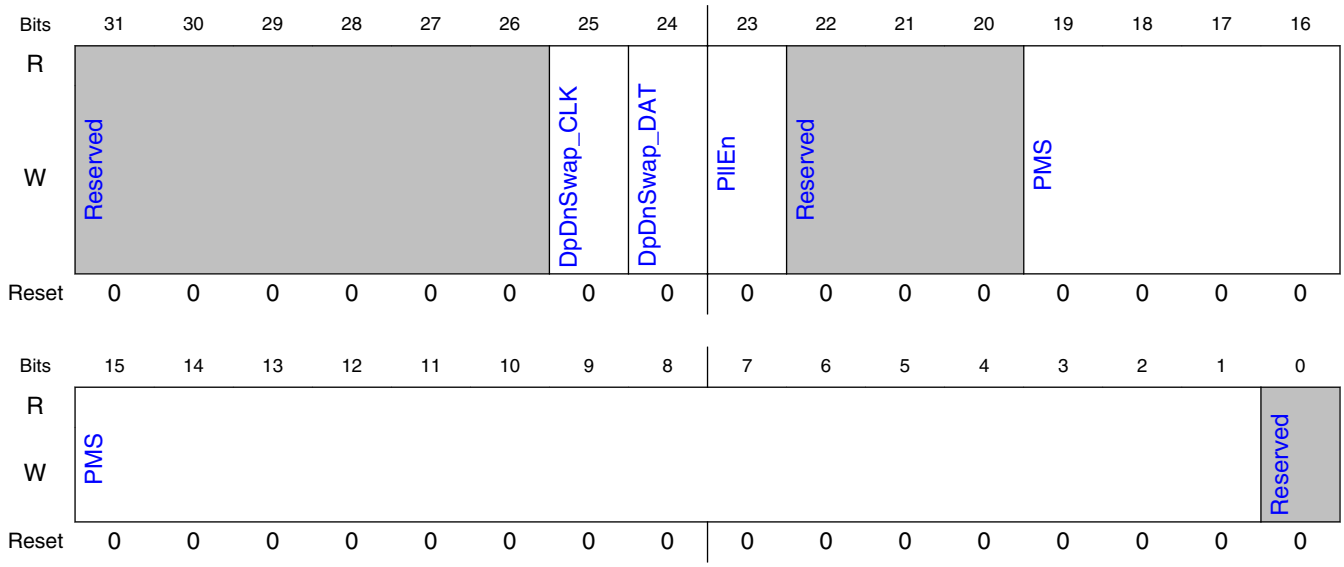
13.6.9.1.25.1 Offset

Register	Offset
DSI_PLLCTRL	94h

13.6.9.1.25.2 Function

This register configures PLL control, D-PHY, clock range indication, and so on.

13.6.9.1.25.3 Diagram



13.6.9.1.25.4 Fields

Field	Function
31-26 —	Reserved
25 DpDnSwap_CLK	Swaps Dp/Dn channel of clock lane. If this bit is set, Dp and Dn channel swap each other.
24 DpDnSwap_DAT	Swaps Dp/Dn channel of Data lanes. If this bit is set, Dp and Dn channel swap each other.
23 PllEn	Enables PLL.
22-20 —	Reserved
19-1 PMS	Specifies the PLL PMS value.
0 —	Reserved

13.6.9.1.26 Specifies the PLL control register 1. (DSI_PLLCTRL1)

13.6.9.1.26.1 Offset

Register	Offset
DSI_PLLCTRL1	98h

13.6.9.1.26.2 Function

This register configures D-PHY PLL SSCG control1.

13.6.9.1.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SSCG_EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.9.1.26.4 Fields

Field	Function
SSCG_EN	Enable pin for dithered mode. (Active-high)
30-0	Reserved

13.6.9.1.27 Specifies the PLL control register 2. (DSI_PLLCTRL2)

13.6.9.1.27.1 Offset

Register	Offset
DSI_PLLCTRL2	9Ch

13.6.9.1.27.2 Function

This register configures D-PHY PLL SSCG control2.

13.6.9.1.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SEL_PF		MRR						MFR							
Reset			0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.9.1.27.4 Fields

Field	Function
31-16 —	Reserved
14-15 SEL_PF	Value of 2-bit modulation method control 00: down spread 01: up spread 1x: center spread
13-8 MRR	Value of 6-bit Modulation Rate (MR) control
7-0 MFR	Value of 8-bit Modulation Frequency (MF) control

13.6.9.1.28 Specifies the PLL timer register. (DSI_PLLTMR)

13.6.9.1.28.1 Offset

Register	Offset
DSI_PLLTMR	A0h

13.6.9.1.28.2 Function

Specifies the PLL timer register.

13.6.9.1.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PllTimer															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PllTimer															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

13.6.9.1.28.4 Fields

Field	Function
31-0 PllTimer	Specifies the PLL Timer for stability of the generated clock (System clock cycle base). If the timer value goes to 0x00000000, the clock stable bit of status and interrupt register is set.

13.6.9.1.29 Specifies the D-PHY control register 1. (DSI_PHYCTRL_B1)

13.6.9.1.29.1 Offset

Register	Offset
DSI_PHYCTRL_B1	A4h

13.6.9.1.29.2 Function

D-PHY Master and Slave Analog block characteristics control registers (B_DPHYCTL).

13.6.9.1.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	B_DPHYCTL0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	B_DPHYCTL0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.9.1.29.4 Fields

Field	Function
31-0 B_DPHYCTL0	B_DPHYCTL[31:0] to D-PHY. For more details refer to B_DPHYCTL register in Media Block Control chapter

13.6.9.1.30 Specifies the D-PHY control register 2. (DSI_PHYCTRL_B2)

13.6.9.1.30.1 Offset

Register	Offset
DSI_PHYCTRL_B2	A8h

13.6.9.1.30.2 Function

D-PHY Master and Slave Analog block characteristics control registers (B_DPHYCTL).

13.6.9.1.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	B_DPHYCTL1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	B_DPHYCTL1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.9.1.30.4 Fields

Field	Function
31-0 B_DPHYCTL1	B_DPHYCTL[63:32] to D-PHY

13.6.9.1.31 Specifies the D-PHY control register 1. (DSI_PHYCTRL_M1)

13.6.9.1.31.1 Offset

Register	Offset
DSI_PHYCTRL_M1	ACh

13.6.9.1.31.2 Function

D-PHY Master Analog block characteristics control registers (M_DPHYCTL).

13.6.9.1.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	M_DPHYCTL0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	M_DPHYCTL0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.9.1.31.4 Fields

Field	Function
31-0 M_DPHYCTL0	M_DPHYCTL[31:0] to D-PHY

13.6.9.1.32 Specifies the D-PHY control register 2. (DSI_PHYCTRL_M2)

13.6.9.1.32.1 Offset

Register	Offset
DSI_PHYCTRL_M2	B0h

13.6.9.1.32.2 Function

D-PHY Master Analog block characteristics control registers (M_DPHYCTL).

13.6.9.1.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	M_DPHYCTL1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	M_DPHYCTL1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.9.1.32.4 Fields

Field	Function
31-0 M_DPHYCTL1	M_DPHYCTL[63:32] to D-PHY

13.6.9.1.33 Specifies the D-PHY timing register. (DSI_PHYTIMING)

13.6.9.1.33.1 Offset

Register	Offset
DSI_PHYTIMING	B4h

13.6.9.1.33.2 Function

D-PHY Master global operating timing registers.

13.6.9.1.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	M_TLPXCTL								M_THSEXITCTL							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.9.1.33.4 Fields

Field	Function
31-16 —	Reserved
15-8 M_TLPXCTL	M_TLPXCTL[7:0] to D-PHY
7-0 M_THSEXITCTL	M_THSEXITCTL[7:0] to D-PHY

13.6.9.1.34 Specifies the D-PHY timing register 1. (DSI_PHYTIMING1)

13.6.9.1.34.1 Offset

Register	Offset
DSI_PHYTIMING1	B8h

13.6.9.1.34.2 Function

D-PHY Master global operating timing registers.

13.6.9.1.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	M_TCLKPRPRCTL								M_TCLKZEROCTL							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	M_TCLKPOSTCTL								M_TCLKTRAILCTL							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.9.1.34.4 Fields

Field	Function
31-24 M_TCLKPRPRCTL	M_TCLKPRPRCTL[7:0] to D-PHY
23-16 M_TCLKZEROCTL	M_TCLKZEROCTL[7:0] to D-PHY
15-8 M_TCLKPOSTCTL	M_TCLKPOSTCTL[7:0] to D-PHY
7-0 M_TCLKTRAILCTL	M_TCLKTRAILCTL[7:0] to D-PHY

13.6.9.1.35 Specifies the D-PHY timing register 2. (DSI_PHYTIMING2)

13.6.9.1.35.1 Offset

Register	Offset
DSI_PHYTIMING2	BCh

13.6.9.1.35.2 Function

D-PHY Master global operating timing registers.

13.6.9.1.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								M_THSPRPRCTL							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	M_THSZEROCTL								M_THSTRAILCTL							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.9.1.35.4 Fields

Field	Function
31-24 —	Reserved
23-16 M_THSPRPRCTL	M_THSPRPRCTL[7:0] to D-PHY
15-8 M_THSZEROCTL	M_THSZEROCTL[7:0] to D-PHY
7-0 M_THSTRAILCTL	M_THSTRAILCTL[7:0] to D-PHY

13.7 MIPI D-PHY (MIPI_DPHY)

13.7.1 Overview

The MIPI D-PHY core IP is a flexible, low power and high-speed serial interface that connects a display driver or a camera sensor to a host processor. The D-PHY provides a synchronous connection between Master and Slave with clock signal originating at the Master and terminating at the Slave. The main operation modes are HS mode (High Speed Mode) and LP mode (Low Power Mode).

13.7.1.1 Block Diagram

The MIPI D-PHY core consist of 5 modules.

- Common block(BIAS/PLL)
- Master clock lane
- Master data 4-lanes
- Slave clock lane
- Slave data 4-lanes

The following figure illustrates the architectural block diagram of 4-lane D-PHY with common block. PLL offers the operating clock for Master block operation. Slave block uses clock from slave clock lane input PAD at the normal operation mode. BIAS block make the analog reference voltages and currents for analog function. HS mode 0.4V voltage regulator is embedded into IP and is just used for Master blocks.

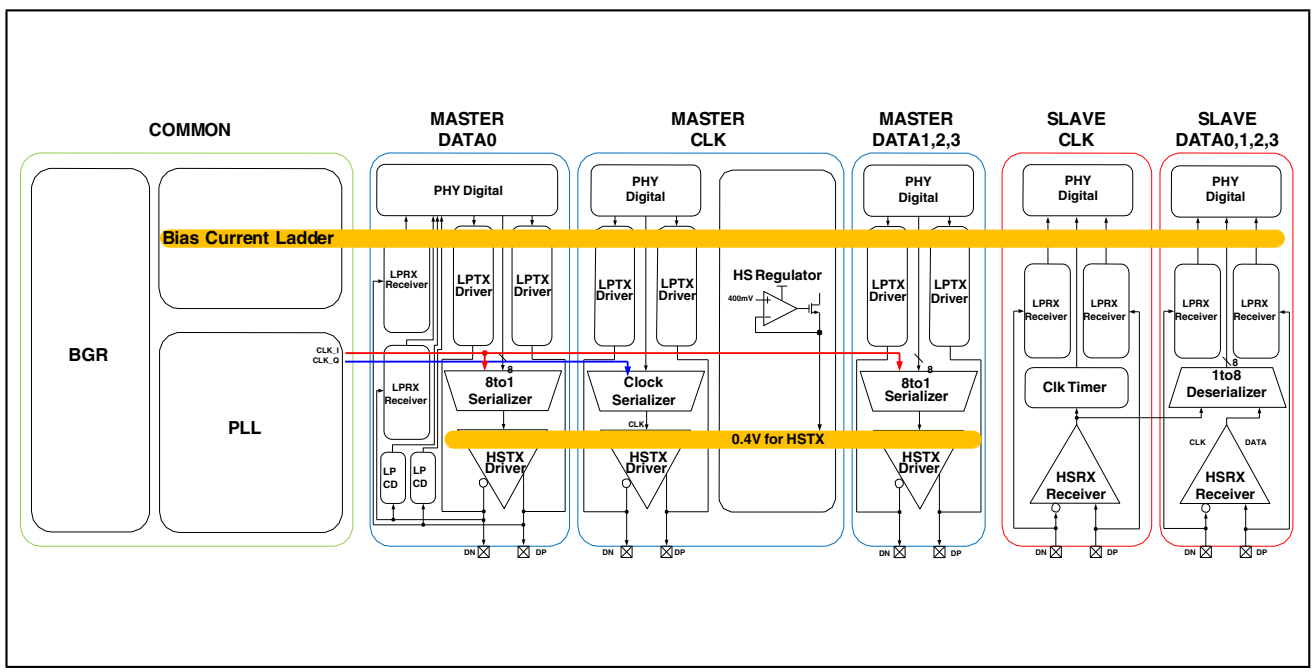


Figure 13-53. D-PHY Architecture

13.7.1.2 Features

The features of MIPI D-PHY are:

- D-PHY spec v1.2 compatible.
- Synchronous link between Master (data source) and Slave (data sink).
- All lanes support high-speed transmission in the forward direction.
- Bi-directional data transmission in Low-Power mode at the Master Data Lane 0 only.
- Uses token passing to control the communication direction of the link.

- High-Speed mode for fast data traffic and Low-Power mode for controls and low-speed data transmission.
- High-Speed mode: Differential and terminated, 200 mV swing: 80 – 1500 Mbps.
- Low-Power mode: Single-ended and non-terminated, 1.2 V swing: 10 Mbps maximum.

NOTE

Use this mode for low-speed asynchronous data communications or controls.

13.7.2 Functional Description

The main functional blocks of MIPI D-PHY are described below.

13.7.2.1 Common

The Common block consists of a band-gap reference (BGR), bias generator, and PLL. Detailed information on each block is described in the following sections.

13.7.2.2 BGR and bias generator

The BGR circuit generates high-accuracy reference voltage of 820mV across PVT variation. A bias generator makes internal-resistor (RMRES)-referred currents, and these currents are distributed to each block in Common and Lanes. The internal resistor-referred current is generated using internally integrated replica resistor to compensate resistance error across PVT variation.

13.7.2.3 Phase Locked Loop (PLL)

The PLL in the Common synthesizes high-speed clock, which is used for TX serializer from a reference clock. The nominal reference frequency is 26MHz.

In the PHY, the high frequency clock from PLL is used in the serialization of TX data. The following are the main features of the PLL:

- Supports output frequency from 80MHz to 2.1GHz clock
- Supports pre-defined and programmable divider setting for MIPI D-PHY specification
- Supports spread spectrum clocking (SSC) with sigma-delta modulated fractional divider.

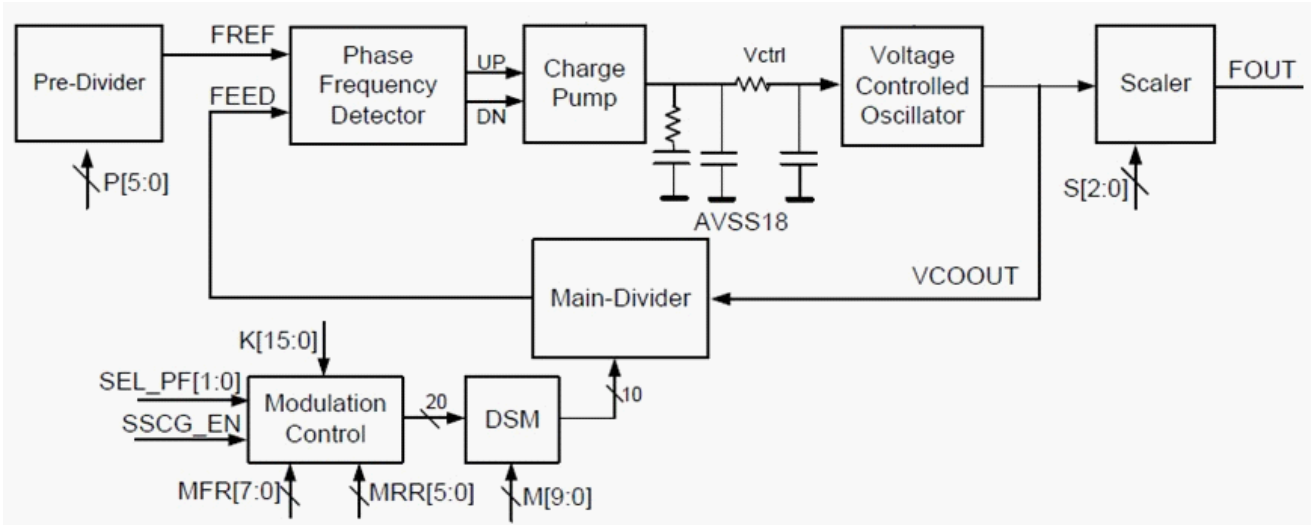


Figure 13-54. Phase Locked Loop (PLL)

13.7.2.4 DPHY PLL

The MIPI DSI DPHY PLL is a high performance PLL based frequency synthesizer that incorporates a independent output divider. The PLL block diagram is shown below.

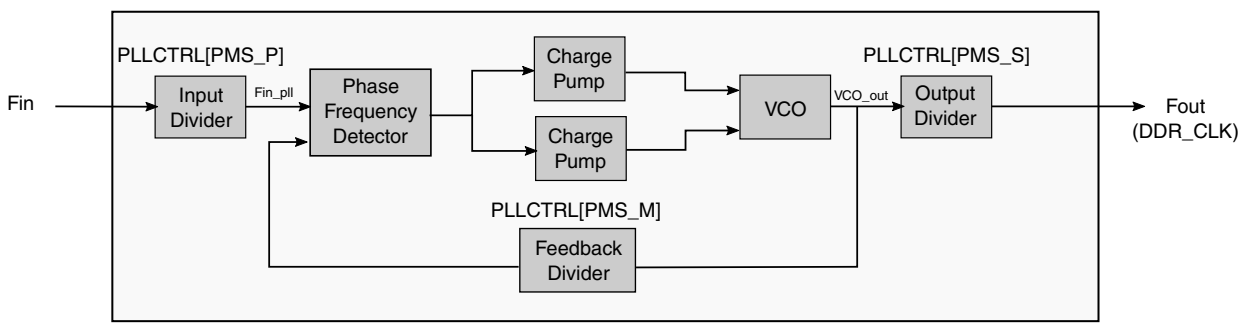


Figure 13-55. MIPI DSI DPHY PLL Block Diagram

The DPHY PLL input clock is Fin. The PLL output (Fout) multiplies the input frequency (Fin) by `(M / (P * 2^S))`.

NOTE

Setting P or M divider values as zero can cause malfunction of the PLL.

NOTE

The P and M divider values must be programmed to maintain VCO stability. The VCO output frequency (VCO_out) range of the MIPI PLL is from 1050 MHz to 2100 MHz.

Table 13-27. DPHY PLL Parameters

Parameter	Description	Range
Fin	Input Frequency	6MHz-300MHz
Fin_pll	P divider frequency	2MHz-30MHz
VCO_out	(M*Fin) / P	1050MHz-1500MHz
Fout	(M*Fin)/(P*2^S)	32.8125MHz-1500MHz
P	Input Divider	1-63
M	Feedback Divider	64-1023
S	Output Divider (2^S)	1,2,4,8

13.7.2.5 DPHY PLL SSCG

When SSCG_EN=1, the spread spectrum mode is enabled. The associated formulas and values are as follows:

- Modulation frequency (MF) = F_{FIN}/ p / mfr / (2^5) [Hz]
- Modulation Rate (MR) = mfr x mrr / m / (2^6) x 100 [%]
- Where, 0 ≤ mfr ≤ 255, 1 ≤ mrr ≤63, 0 ≤ mrr x mfr ≤ 512

13.7.2.6 Serializer

The serializer accepts 8-bit data from the PHY digital which is synchronized to TX byte clock. The parallel data is converted to the high-speed serial data. Finally, serializer transmits the serial data through the driver.

13.7.2.7 HS TX Driver

The driver generates baud-rate voltage waveform on TX output pads. The driver comprises of pre-driver and main-driver. The output data of serializer are delivered to the main-driver via the pre-driver. The main driver is based on voltage-mode driver to achieve a small power consumption and voltage swing.

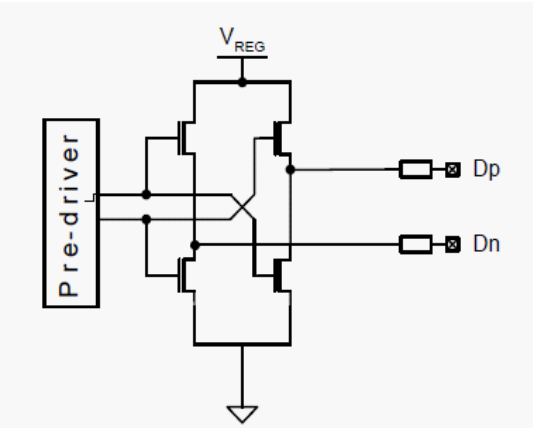


Figure 13-56. HS TX Driver

13.7.2.8 LP TX Driver

In the low power mode, LP TX driver sends the low frequency data under 10Mbps. Generally LP-11 stop state and HS enter sequence (LP-11 ◊ LP-01 ◊ LP-00) are used mostly.

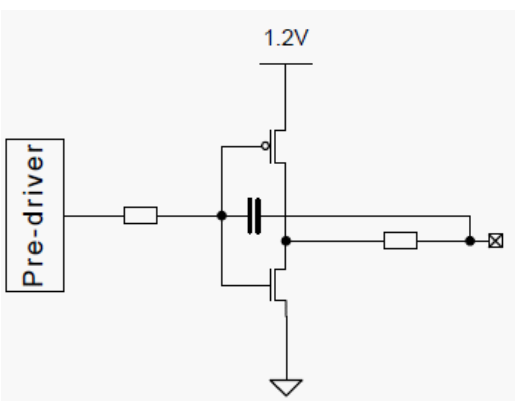


Figure 13-57. LP TX Driver

13.7.2.9 De-Serializer

In the BIST (Built-In Self-Test) mode, the serial data from the serializer is fed back to the de-serializer. And in the real mode, the real data from HS-RX receiver is sent to the de-serializer. De-serializer converted serial data to 8- bit parallel data and transferred them into PHY digital circuit for sending data to link or checking bit error at BIST mode.

13.7.2.10 HS RX Receiver

The high speed data from the MIPI D-PHY master has the MIPI D-PHY electrical characteristics. HS RX receiver convert it to the TTL CMOS level and send it to de-serializer for making the low parallel data.

13.7.2.11 LP RX Receiver

In the low power mode, LP RX receiver make esc clock from LP signal and converts the voltage level from 1.2V level to TTL CMOS logic level. And it sends them to PHY digital for command interpretation. And LP RX receiver always watches the DP/DN signal level. If DP/DN goes to LP-11, PHY recognizes Stop-state, and enters to low power mode.

13.7.2.12 PLL and Clock Lane Connection

MIPI D-PHY is a source synchronous system where the Master clock lane sends high speed differential clock to Slave clock lane. Master PLL generates the high speed clock from input clock. During LP (Low-Power) mode, LP signals are sent by the master block. LP-TX sends the LP signals requried for changing modes between LP (Low-Power) mode and HS (High-Power) mode. Slave block has a termination resistor between the differential signals that is enabled during HS mode.

The following figure illustrates the PLL and Clock Lane connection.

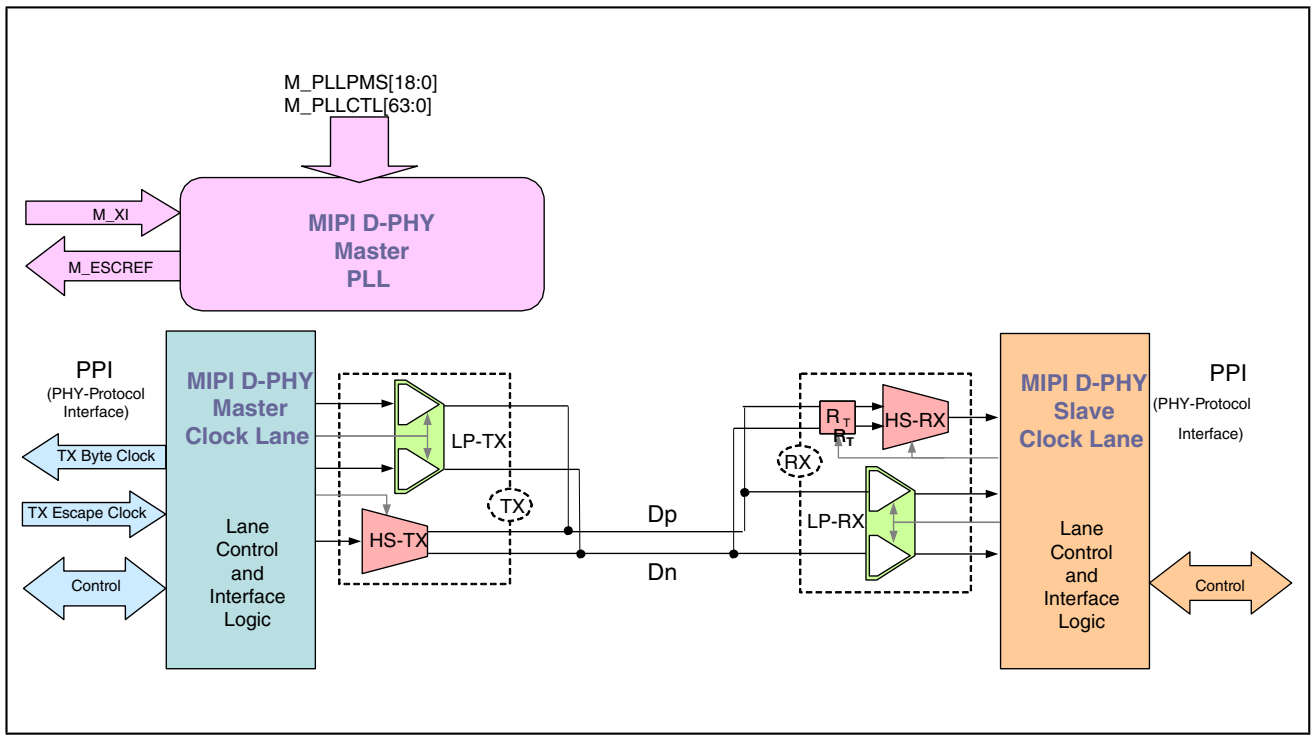


Figure 13-58. PLL and Clock Lane Connection

13.7.2.13 Master PLL: PHY-Protocol Interface (PPI) Signals

The following table describes the PPI signals of Master PLL.

Table 13-28. Master PLL: PPI Signals

Name	I/O	Active Level	Description
M_XI	I	N/A	PLL reference input clocks. You should shield and route the inputs shortly.
M_PLLPMS[18:0]	I	N/A	80 MHz to 1.5 GHz Frequency Control Signals from the register: PMS setting.
M_PLLCTL[63:0]	I	N/A	PLL control.
M_BITCLKHSDIV4	O	N/A	Used for Protocol Layer multiple Lanes FIFO controls (maximum 525 MHz).
M_PLLEN	I	High	Initializes PLL during LOW state and enable when it is HIGH state. NOTE: You must assert to Low state during power-up sequence for a stable initialization.

13.7.2.14 Data Lane Connection

The following figure illustrates the Data0 Lane connection between Master and the Slave. Master Data0 lane is important for DSI interface as it has the LP-CD and LP-RX. LP-RX is used for Turn-Around function. It used to be called as Bus Turn-Around (BTA) and it is needed for sending Low Power Data Transmission from Slave to Master. Low Power - Contention Detector (LP-CD) is used for detecting the abnormal state when both of Master LP-TX and Slave LP-TX at BTA mode are enabled and the output signals are conflicted.

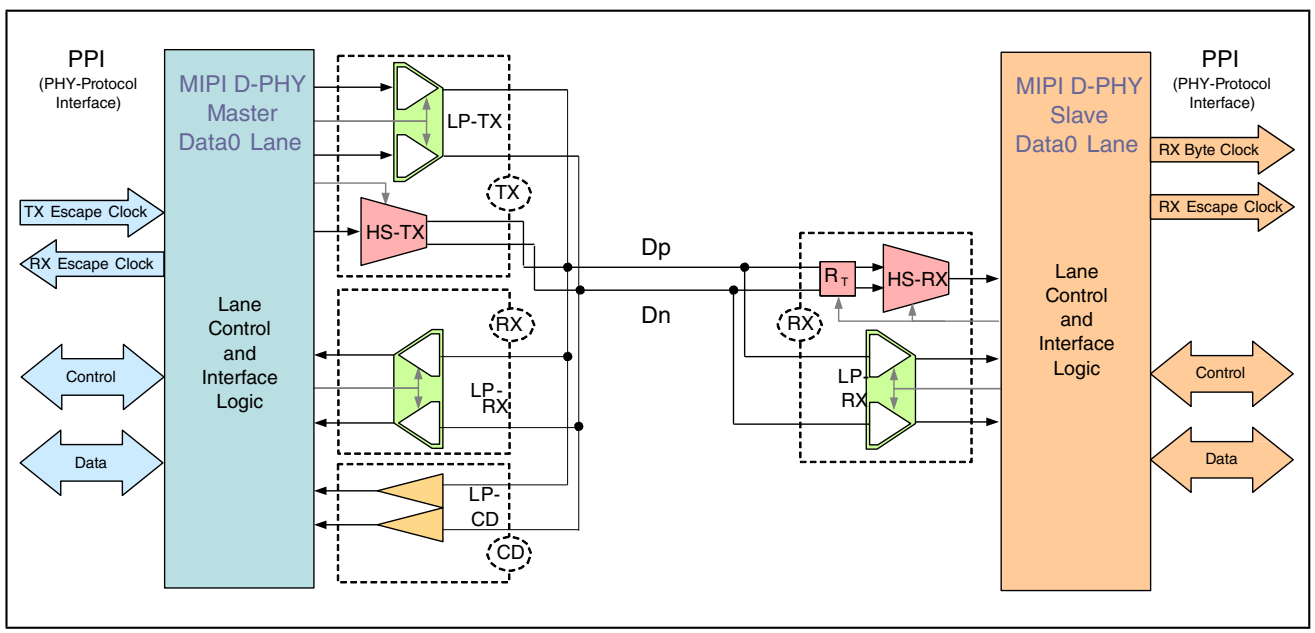


Figure 13-59. Data0 Lane Connection

The following figure illustrates the Data Lane connection between Master and the Slave for 1, 2 and 3 lanes. LP mode signals are used to switch between LP mode and HS mode. HS mode is used for interfacing high speed signals. Slave block has termination resistor between differential signals that is enabled during HS mode.

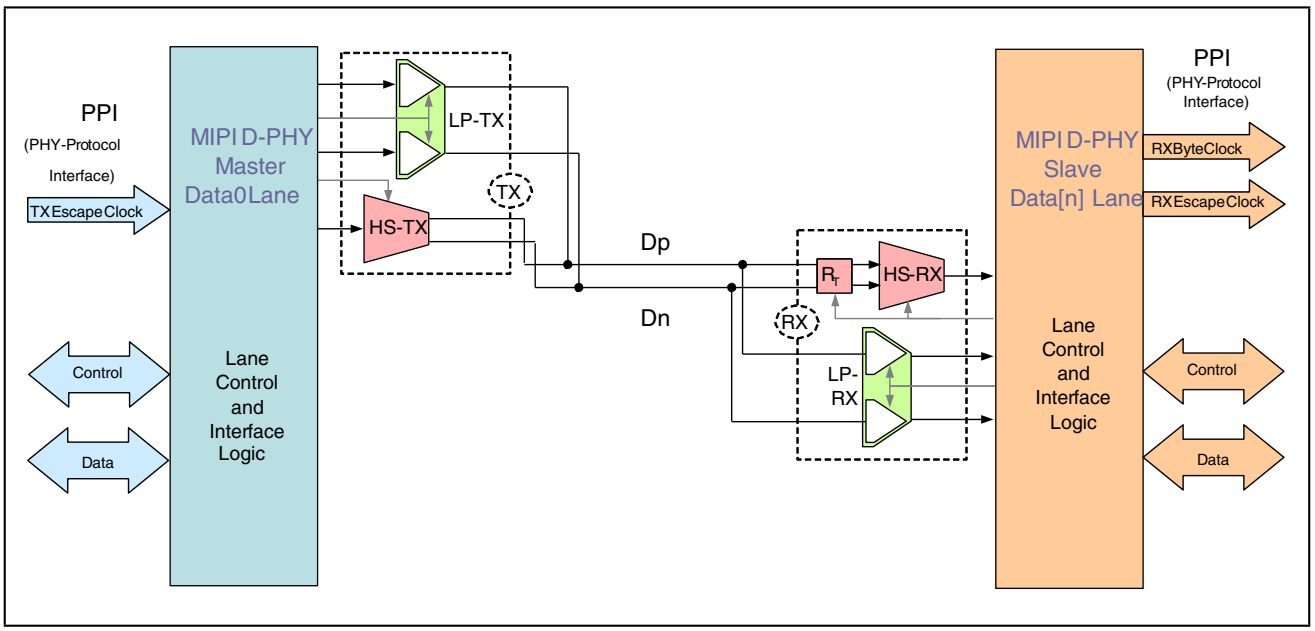


Figure 13-60. Data[n] Lane Connection

13.7.2.15 Protocol Support Summary

A summary of supported standards with corresponding data rates is listed in the below table.

Table 13-29. Supported Standards

Standard	Lane	Data Rate
MIPI D-PHY V1.2	4	80Mbps~1500Mbps

13.7.2.16 IP Interface

MIPI D-PHY configuration contains a Clock Lane Module and 1–4 Data Lane Modules. "0/1/2/3" in signals signify the data lane 0/1/2/3.

13.7.2.16.1 Pad Signals

The MIPI D-PHY core includes the I/O pads. I/O pads are dedicated to D-PHY and you cannot share them with any other signals. You can determine the number of power pads and signal pads from the core module configurations.

The following table describes the pad signals.

Table 13-30. Pad Signals

Mode	I/O	Active Level	Description
M_VREG_0P4V	I/O	N/A	Regulator capacitor connection
M_DPCLK	O	N/A	Master CLK Lane DP
M_DNCLK	O	N/A	Master CLK Lane DN
M_DPDATA0	O	N/A	Master DATA0 Lane DP
M_DNDATA0	O	N/A	Master DATA0 Lane DN
M_DPDATA1	O	N/A	Master DATA1 Lane DP
M_DNDATA1	O	N/A	Master DATA1 Lane DN
M_DPDATA2	O	N/A	Master DATA2 Lane DP
M_DNDATA2	O	N/A	Master DATA2 Lane DN
M_DPDATA3	O	N/A	Master DATA3 Lane DP
M_DNDATA3	O	N/A	Master DATA3 Lane DN
S_DPCLK	I	N/A	Slave CLK Lane DP
S_DNCLK	I	N/A	Slave CLK Lane DN
S_DPDATA0	I	N/A	Slave DATA0 Lane DP
S_DNDATA0	I	N/A	Slave DATA0 Lane DN
S_DPDATA1	I	N/A	Slave DATA1 Lane DP
S_DNDATA1	I	N/A	Slave DATA1 Lane DN
S_DPDATA2	I	N/A	Slave DATA2 Lane DP
S_DNDATA2	I	N/A	Slave DATA2 Lane DN
S_DPDATA3	I	N/A	Slave DATA3 Lane DP
S_DNDATA3	I	N/A	Slave DATA3 Lane DN
VDD18_PLL	Supply	N/A	1.8V Power for PLL
DVDD08D_PLL	Supply	N/A	0.9V Power for PLL
MS_VDD_0P8	Supply	N/A	0.9 V Power for Internal Logic
MS_VDD18	Supply	N/A	1.8 V Power for Analog
M_VDD12	Supply	N/A	1.2 V Power for Analog
MS_VSS	GND	N/A	Ground
RTO/SNS/VB0/VB1/V18/V25	I/O	N/A	These signals are I/O pad's ring signals. If you want to get more detail information about these signals, check the datasheet of GPIO.

13.7.2.16.2 ESD Protection Port

The anti-parallel diode is located between the D-PHY ground and the SoC logic ground for ESD protection.

Table 13-31. ESD Protection Port

Mode	I/O	Active Level	Description
VSS_CDM	Supply	N/A	This port is connected to the SoC logic ground with wide metal. See Figure 21.

13.7.2.16.3 Control Register: PPI Signals

13.7.2.16.3.1 Timing Control Register of High-Speed Data Transmission

To enable High-Speed Transmission, the following registers need to be set. For more information on the registers, refer to the register description section.

The following table describes the Timing Control Registers of High-Speed Data Transmission.

Table 13-32. Timing Control Registers of High-Speed Data Transmission

Name	I/O	Active Level	Description
M_TCLKPRPRCTL[7:0]	I	N/A	Master Clock Lane Control Register for TCLK-PREPARE.
M_TCLKZEROCTL[7:0]	I	N/A	Master Clock Lane Control Register for TCLK-ZERO.
M_TCLKPOSTCTL[7:0]	I	N/A	Master Clock Lane Control Register for TCLK-POST.
M_TCLKTRAILCTL[7:0]	I	N/A	Master Clock Lane Control Register for TCLK-TRAIL.
M_THSPRPRCTL[7:0]	I	N/A	Master Data Lanes Control Register for THS-PREPARE.
M_THSZEROCTL[7:0]	I	N/A	Master Data Lanes Control Register for THS-ZERO.
M_THSTRAILCTL[7:0]	I	N/A	Master Data Lanes Control Register for THS-TRAIL.
M_TLPXCTL[7:0]	I	N/A	Master Clock and Data Lanes Control Register for TLPX.
M_THSEXITCTL[7:0]	I	N/A	Master Clock and Data Lanes Control Register for THS-EXIT.
S_CLKSETTLECTL[1:0]	I	N/A	Slave Clock Lane Control Register for TCLK-SETTLE. 2'b0x = 110 ns to 280ns (v0.87 to v1.2) 2'b10 = 150 ns to 430ns (v0.83 to v0.86) 2'b11 = 60 ns to 140ns (v0.82)
S_HSSETTLECTL[7:0]	I	N/A	Slave Clock Lane Control Register for THS-SETTLE.

13.7.2.16.4 PPI Requests and Timing Conditions

13.7.2.16.4.1 PPI Request Conditions

Following are the request conditions invoked by the D-PHY:

- The Correct request timing window is in the Stop state.
- Accepts only one request from protocol.
- D-PHY selects only one request based on the service priority when you assert two or more requests.

13.7.2.16.4.2 PPI Requests and D-PHY Timing

HS-TX:

- **Latency** - The minimum required FIFO depth is 128, because the HS-TX requests the HS-TX ready latency.
- **Data Lane** - D-PHY ensures $T_{HS-EXIT}$ (minimum 100 ns) timing.
- **Clock Lane** - D-PHY must guarantee the $T_{CLK-PRE}$ and $T_{CLK-POST}$ timings.

NOTE

The protocol layer need not control these timings.

- **HS-2-HS Request Interval** - Not implied.

LP:

- **LP–2–LP Request** - The LP-2-LP request wait for the 2-ESC CLK.

LP and HS:

- **LP–2–HS Request Interval** - LP-2-HS request interval wait for the 2-ESC CLK in the Stop state.
- **HS-2-LP Request Interval** - Not implied.

13.7.2.16.4.3 Protocol Watchdog Timers

Following are the features of the Protocol Watchdog Timers:

- **HS RX Timeout** - Protocol specific.
  - The Protocol times-out when there is no EoT received within a certain period in the HS RX mode.
- **HS TX Timeout** - Protocol specific.
  - The maximum transmission length in HS TX is bounded.
- **Escape Mode Timeout** - Protocol specific.
  - The Escape mode timeout should be greater than the Escape mode Silence Limit of the other device.

- **Escape Mode Silence Timeout** - Protocol specific.
  - A device has a bound length for LP TX-00 during Escape mode, after which the other device may timeout. For example, a display module must have an Escape Mode Silence Limit, after which the host processor can timeout.
- **Turnaround Errors**
  - The Protocol times out initiates appropriate action when there is no acknowledgement observed within a certain period of time.

13.7.2.16.5 Core Interface Timing Diagrams

13.7.2.16.5.1 Clock Lane: HS-TX and HS-RX Function

The following figure illustrates the HS-TX and HS-RX function of the Clock Lane.

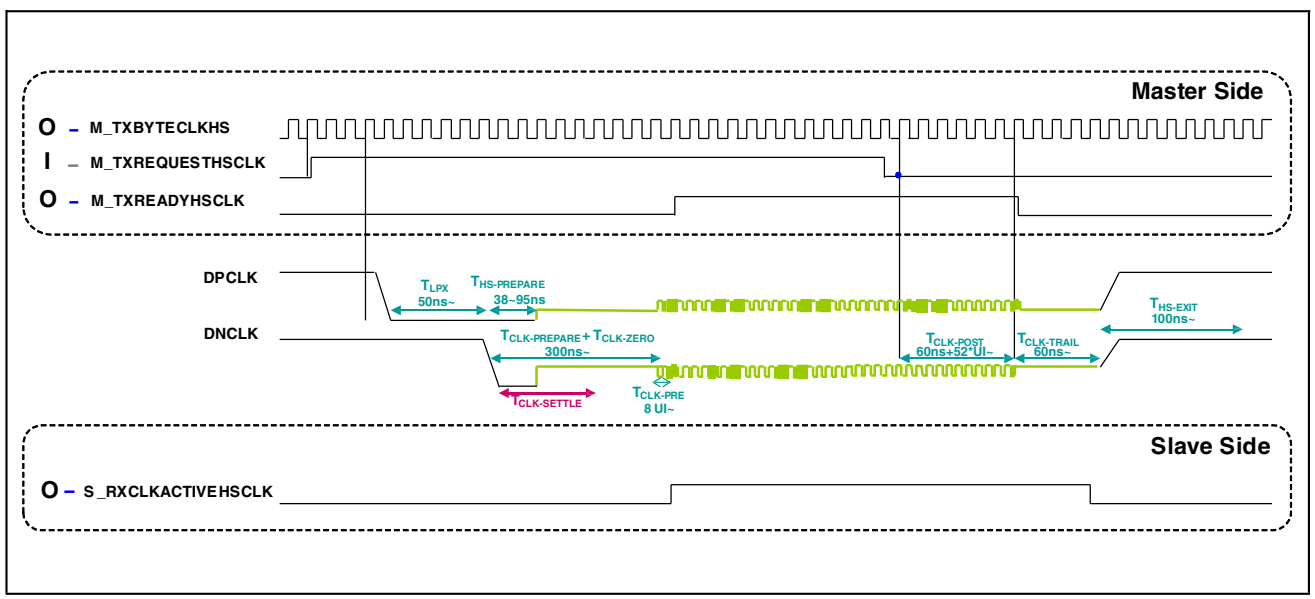


Figure 13-61. Clock Lane: HS-TX and HS-RX Function

13.7.2.16.5.2 Data Lane: HS-TX and HS-RX Function

The following figure illustrates the HS-TX and HS-RX function of the Data Lane.

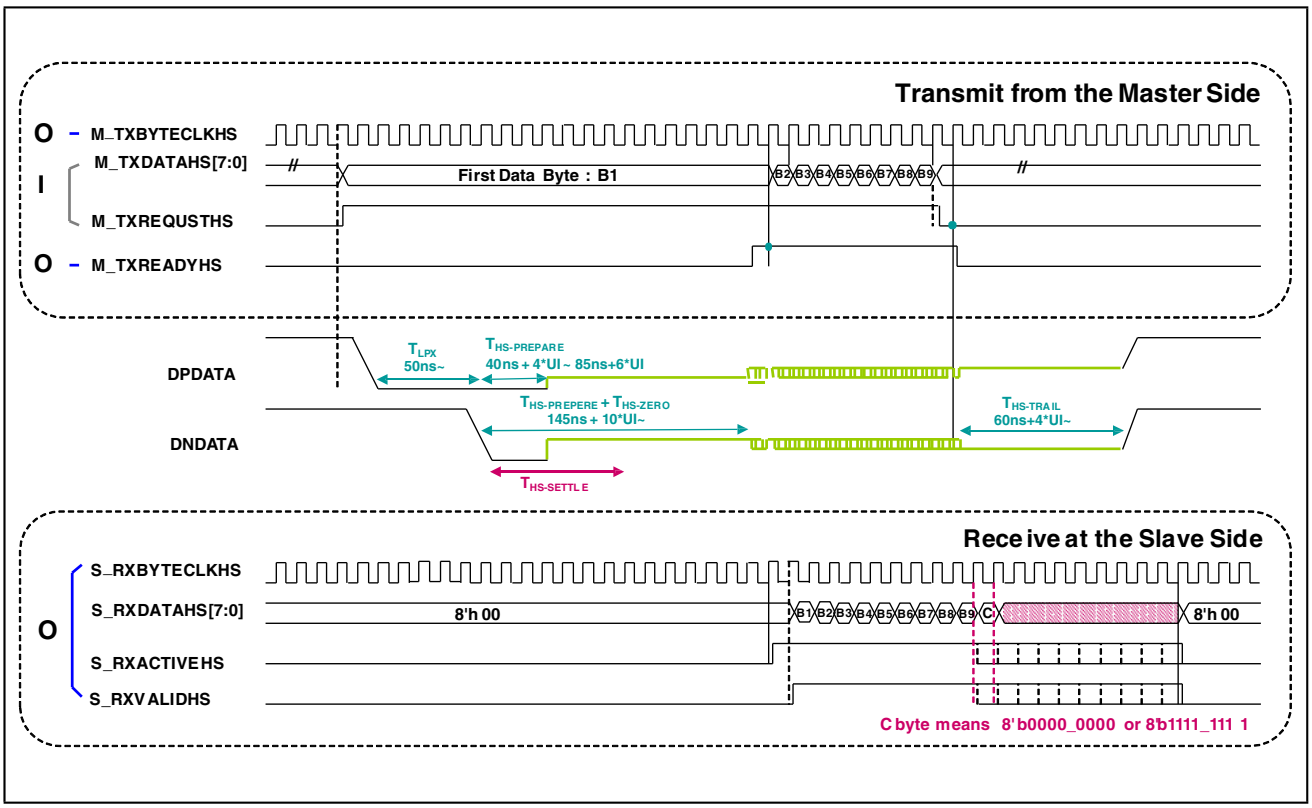


Figure 13-62. Data Lane: HS-TX and HS-RX Function

13.7.2.16.5.3 Clock Lane: ULPS Function

The following figure illustrates the ULPS function of the Clock Lane.

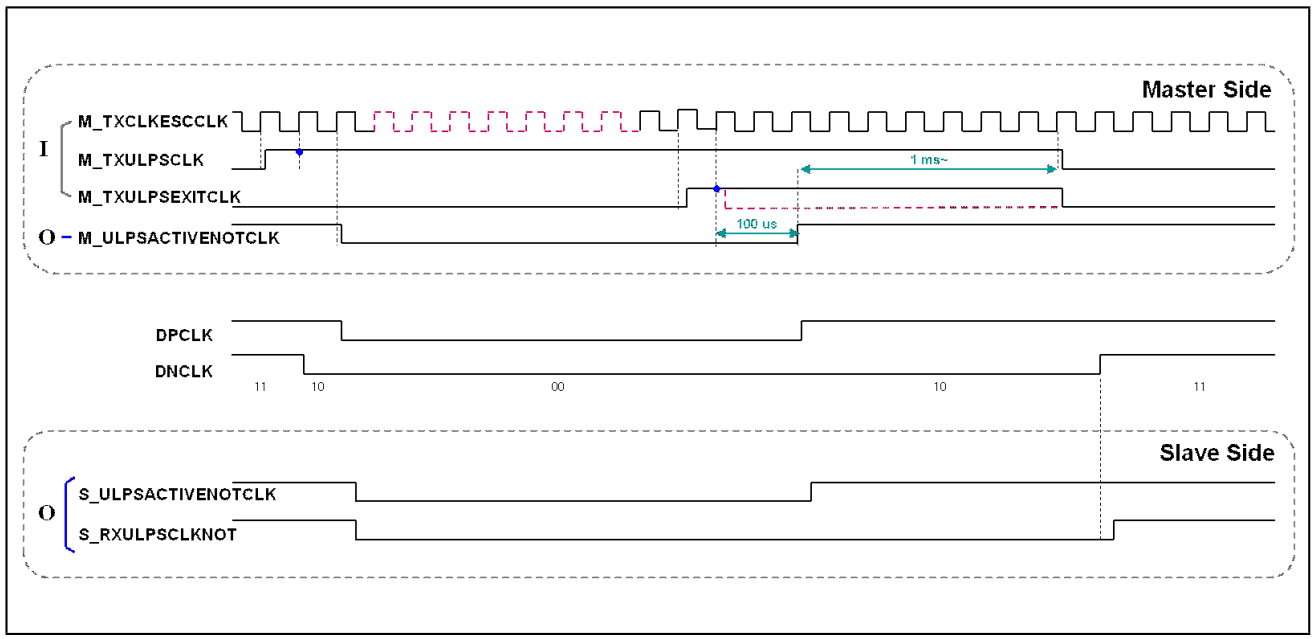


Figure 13-63. Clock Lane: ULPS Function

13.7.2.16.5.4 Data Lane: ULPS Function

The following figure illustrates the ULPS function of the Data Lane.

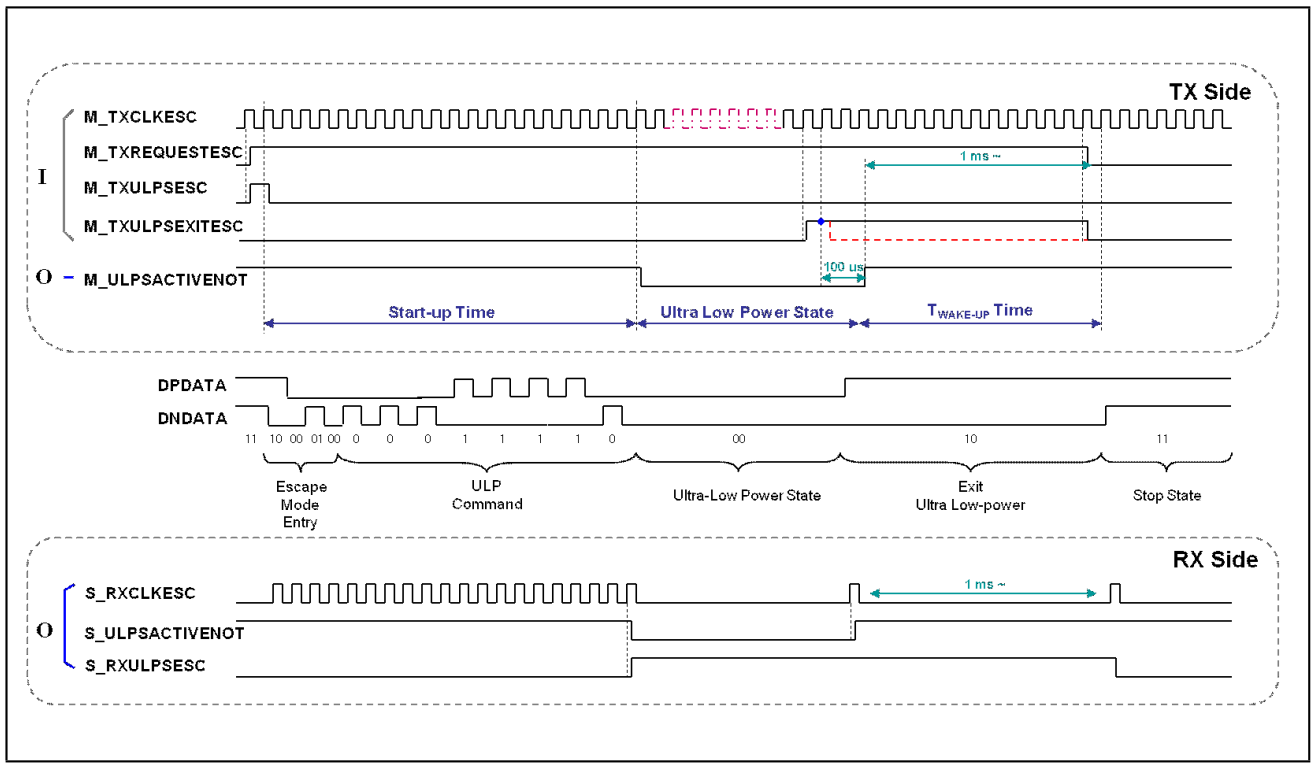


Figure 13-64. Data Lane: ULPS Function

13.7.2.16.5.5 Data Lane: LP-TX and LP-RX Function

The following figure illustrates the LP-TX and LP-RX function of the Data Lane.

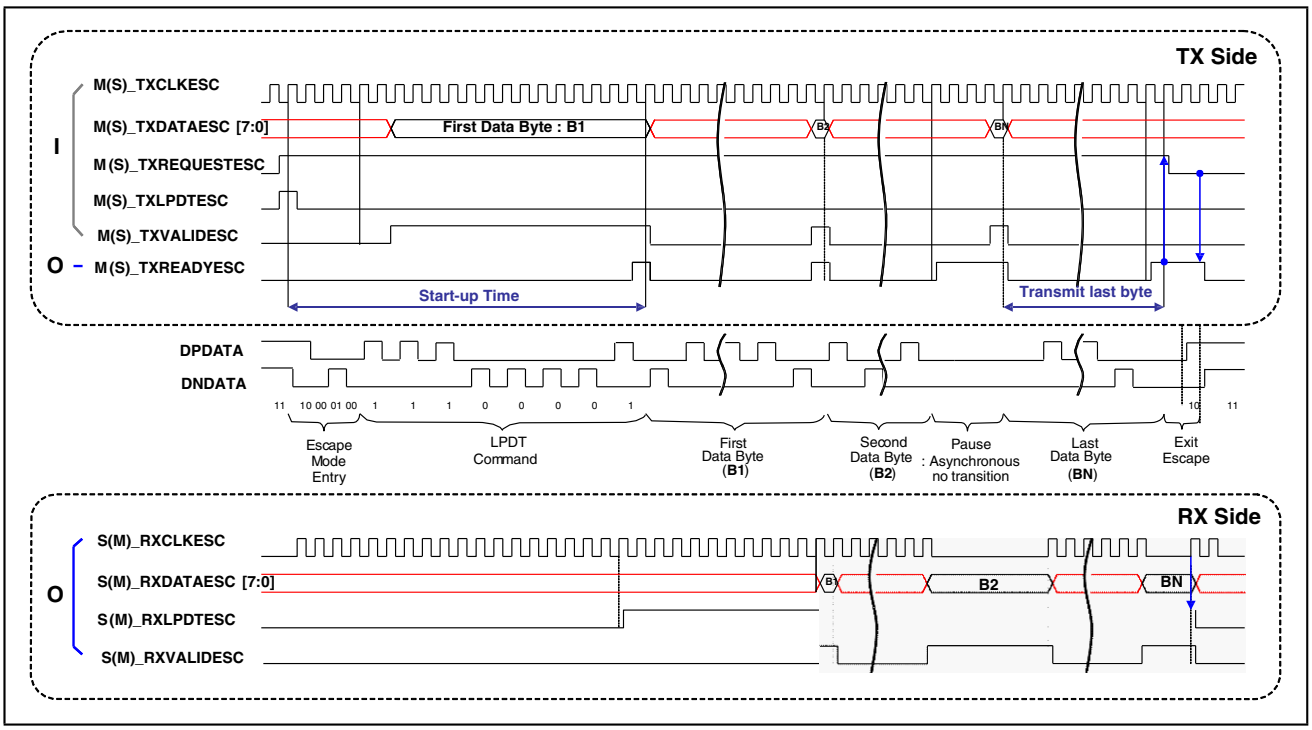


Figure 13-65. Data Lane: LP-TX and LP-RX Function

13.7.2.16.5.6 Data Lane: Remote Trigger Reset

The following figure illustrates the remote trigger reset of the Data Lane.

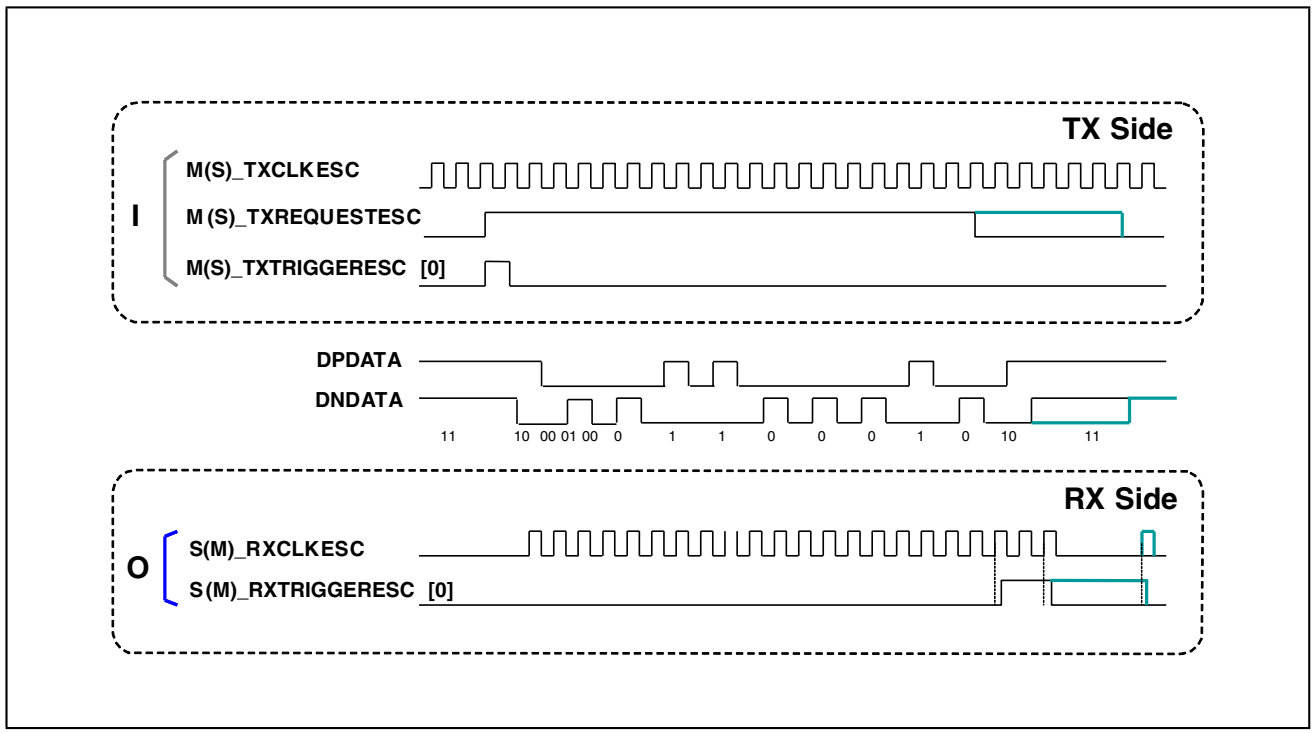


Figure 13-66. Data Lane: Remote Trigger Reset

13.7.2.16.5.7 Data Lane: Turn Around

The following figure illustrates the turnaround of the Data Lane.

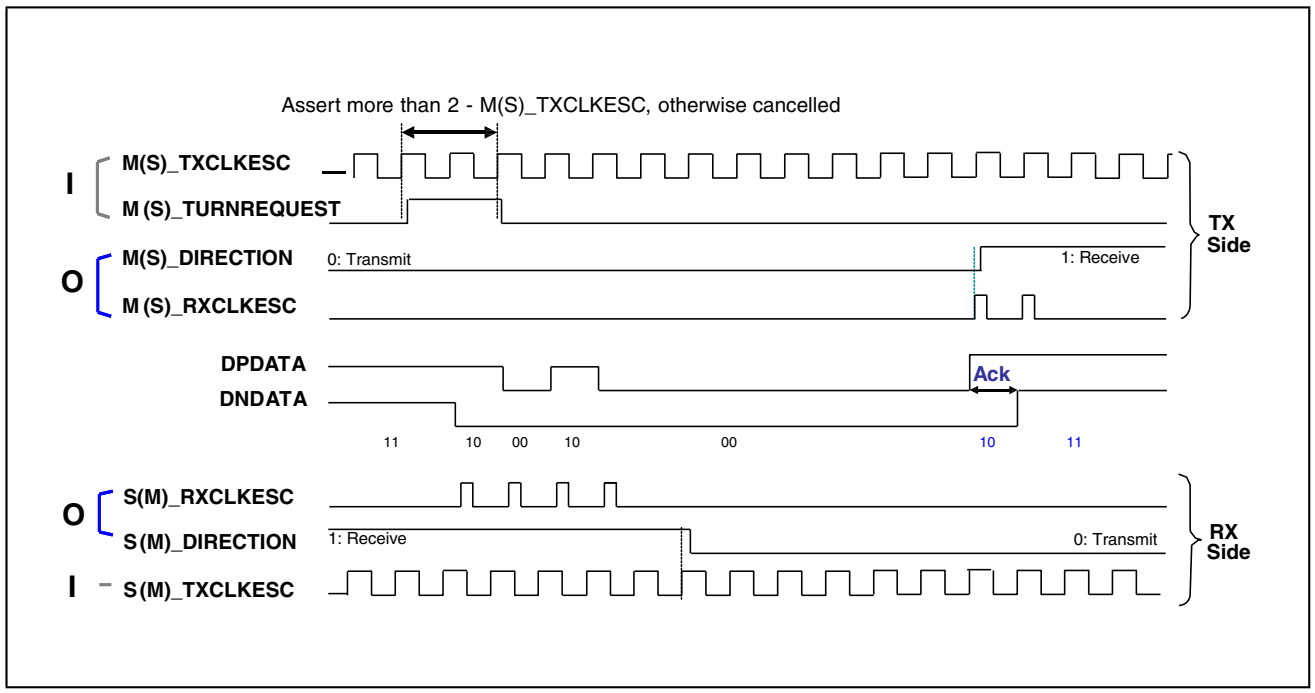


Figure 13-67. Data Lane: Turn Around

13.7.2.16.5.8 Skew Calibration

The following figure illustrates the Skew Calibration sequence.

The protocol should assert the M_TXSKEWCALHS synchronized M_TXBYTECLKHS for start Skew Calibration. PHY-Protocol Interface output signals of the Slave are 1'b0 in the Skew Calibration mode.

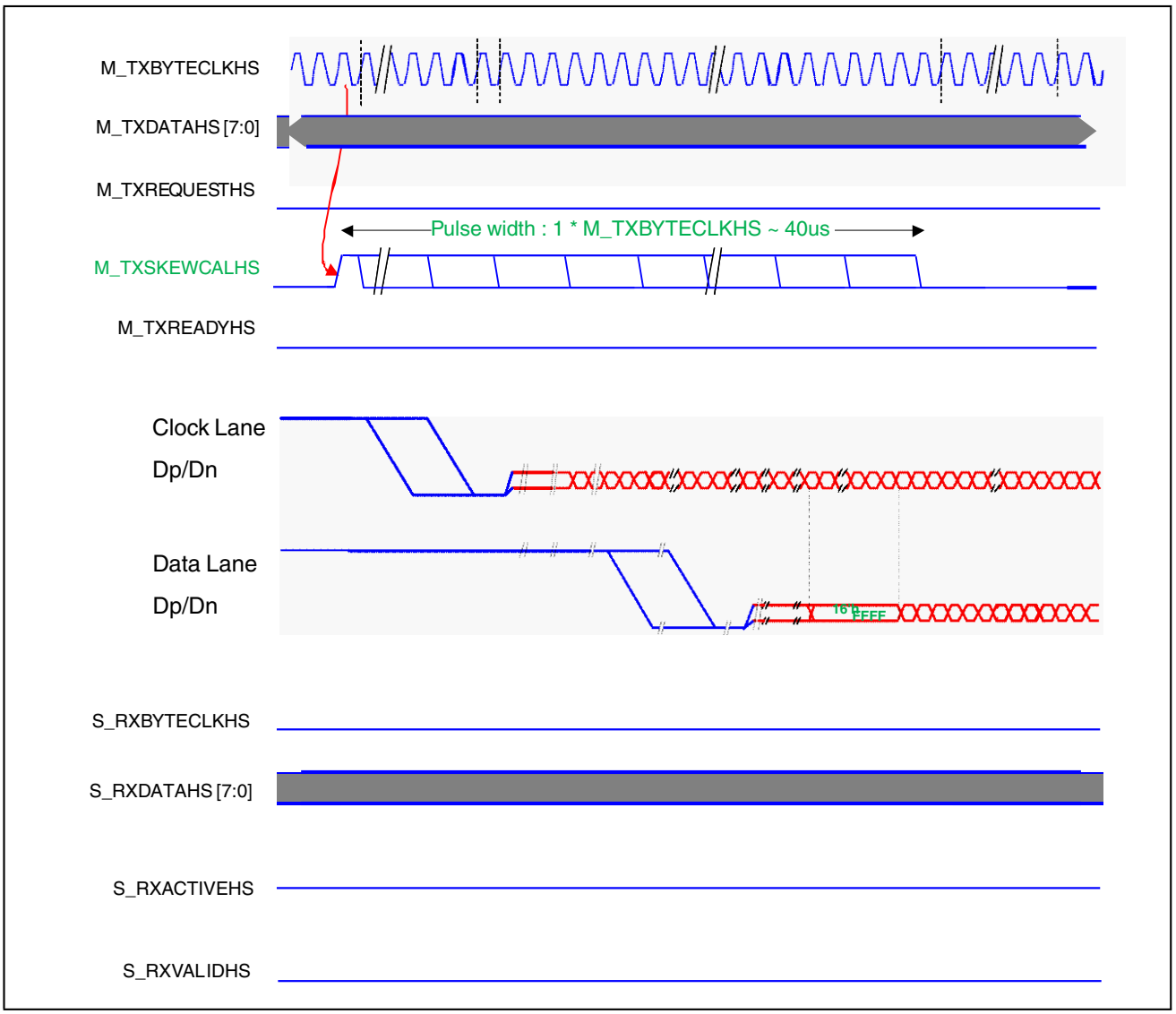


Figure 13-68. Skew Calibration

13.7.3 Clocks

The table below shows the clock signal in MIPI D-PHY:

Table 13-33. Clocks

Clocks	Description
M_XI	Master PLL reference input clock.
M_TXBYTECLKHS	HS Clock to Master Clock Lane and Master Data0/1/2/3 Lane.
S_RXBYTECLKHS0	HS Clock from Slave Data Lane0. Assign output PAD for monitoring.

13.7.4 Reset Signals

The following table describes the reset and test mode signals.

Table 13-34. Reset : PPI Signals

Name	I/O	Active Level	Description
M_RESETN	I	Low	Master Clock and Data Lane Reset (Active Low)
S_RESETN	I	Low	Slave Clock and Data Lane Reset (Active Low)
INTERNAL_LOOPBACK_EN	I	High	Internal or External Loopback selection signal • 1'b0 = External Loopback Mode (default) • 1'b1 = Internal Loopback Mode
SCAN_MODE	I	High	Scan Test mode selection
M_SCAN_CLK_0	I	N/A	Master Data0 Lane Scan Test Mode Clock
M_SCAN_CLK_1	I	N/A	Master Data1 Lane Scan Test Mode Clock
M_SCAN_CLK_2	I	N/A	Master Data2 Lane Scan Test Mode Clock
M_SCAN_CLK_3	I	N/A	Master Data3 Lane Scan Test Mode Clock
M_SCAN_IN_0	I	N/A	Master Data0 Lane Scan Test Mode Input
M_SCAN_IN_1	I	N/A	Master Data1 Lane Scan Test Mode Input
M_SCAN_IN_2	I	N/A	Master Data2 Lane Scan Test Mode Input
M_SCAN_IN_3	I	N/A	Master Data3 Lane Scan Test Mode Input
M_SCAN_OUT_0	O	N/A	Master Data0 Lane Scan Test Mode Output
M_SCAN_OUT_1	O	N/A	Master Data1 Lane Scan Test Mode Output
M_SCAN_OUT_2	O	N/A	Master Data2 Lane Scan Test Mode Output
M_SCAN_OUT_3	O	N/A	Master Data3 Lane Scan Test Mode Output
S_SCAN_CLK_0	I	N/A	Slave Data0 Lane Scan Test Mode Clock
S_SCAN_CLK_1	I	N/A	Slave Data1 Lane Scan Test Mode Clock

Table continues on the next page...

Table 13-34. Reset : PPI Signals (continued)

Name	I/O	Active Level	Description
S_SCAN_CLK_2	I	N/A	Slave Data2 Lane Scan Test Mode Clock
S_SCAN_CLK_3	I	N/A	Slave Data3 Lane Scan Test Mode Clock
S_SCAN_IN_0	I	N/A	Slave Data0 Lane Scan Test Mode Input
S_SCAN_IN_1	I	N/A	Slave Data1 Lane Scan Test Mode Input
S_SCAN_IN_2	I	N/A	Slave Data2 Lane Scan Test Mode Input
S_SCAN_IN_3	I	N/A	Slave Data3 Lane Scan Test Mode Input
S_SCAN_OUT_0	O	N/A	Slave Data0 Lane Scan Test Mode Output
S_SCAN_OUT_1	O	N/A	Slave Data1 Lane Scan Test Mode Output
S_SCAN_OUT_2	O	N/A	Slave Data2 Lane Scan Test Mode Output
S_SCAN_OUT_3	O	N/A	Slave Data3 Lane Scan Test Mode Output
M_PLL_LOCK_INT	O	N/A	Dummy (NC)
M_PLL_SER_CLK	O	N/A	Dummy (NC)
S_LOOPBACK_PASS_0	O	High	Slave Data0 Lane Loopback Test Mode Output
S_LOOPBACK_PASS_1	O	High	Slave Data1 Lane Loopback Test Mode Output
S_LOOPBACK_PASS_2	O	High	Slave Data2 Lane Loopback Test Mode Output
S_LOOPBACK_PASS_3	O	High	Slave Data3 Lane Loopback Test Mode Output
S_LOOPBACK_FAIL_0	O	High	Slave Data0 Lane Loopback Test Mode Output
S_LOOPBACK_FAIL_1	O	High	Slave Data1 Lane Loopback Test Mode Output
S_LOOPBACK_FAIL_2	O	High	Slave Data2 Lane Loopback Test Mode Output
S_LOOPBACK_FAIL_3	O	High	Slave Data3 Lane Loopback Test Mode Output
S_ATB_OUT_0/1/2/3[31:30]	O	N/A	Dummy
S_ATB_OUT_0/1/2/3[29:24]	O	N/A	RX Skew Calibration results value (Data Lane)
S_ATB_OUT_0/1/2/3[23:22]	O	N/A	Dummy
S_ATB_OUT_0/1/2/3[21:16]	O	N/A	RX Skew Calibration results value (Clock Lane)
S_ATB_OUT_0/1/2/3[15:2]	O	N/A	Dummy

Table continues on the next page...

Table 13-34. Reset : PPI Signals (continued)

Name	I/O	Active Level	Description
S_ATB_OUT_0/1/2/3[1]	O	N/A	RX Skew Calibration done signal
S_ATB_OUT_0/1/2/3[0]	O	N/A	RX Skew Calibration compare pass signal

13.7.5 Interrupts

There are no interrupt signals for MIPI D-PHY block.

13.7.6 External Signals

The external signal that are associated with MIPI blocks are listed in their respective chapters.

13.7.7 Initialization

The below figure illustrates the initialization sequence.

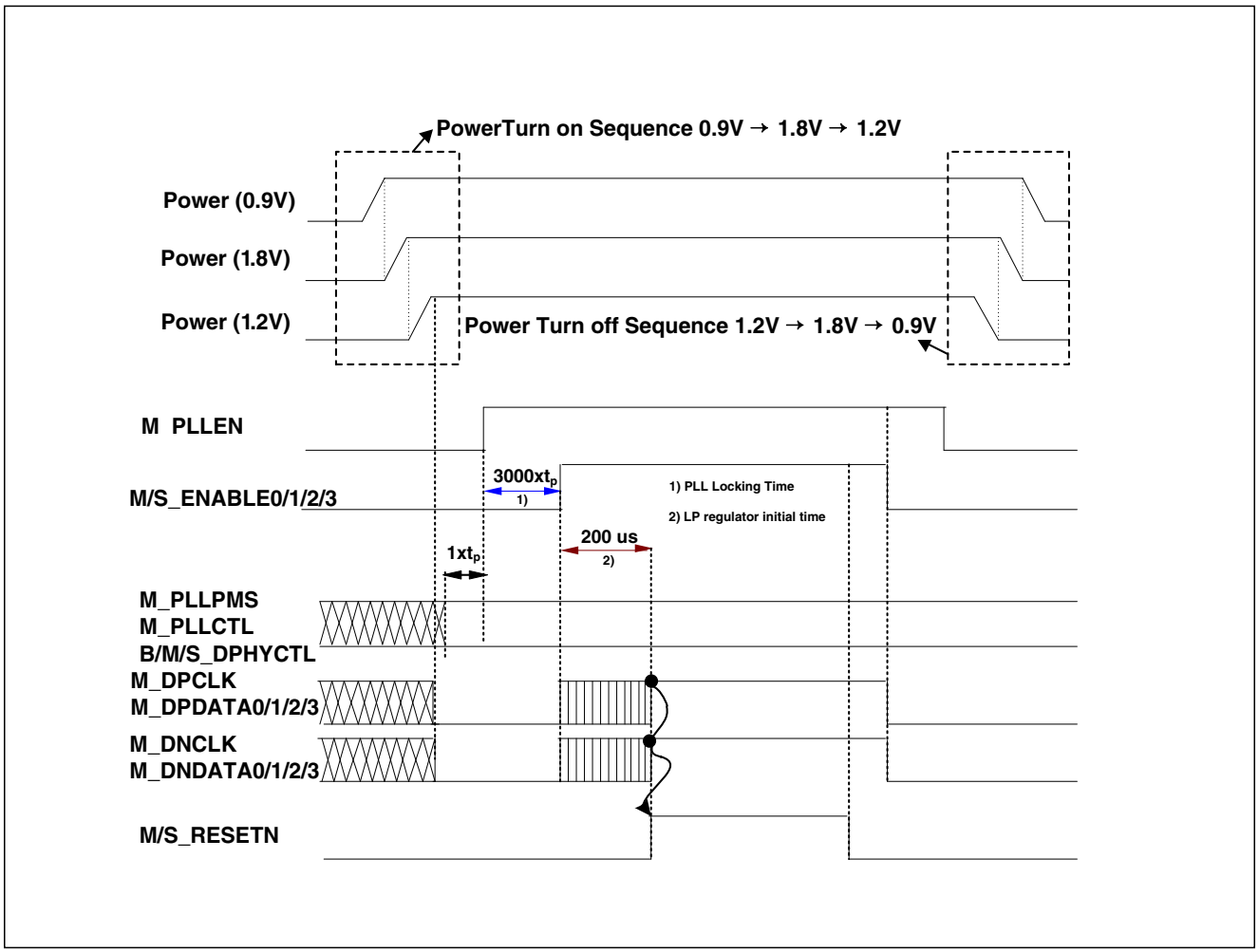


Figure 13-69. Initialization Sequence

13.7.8 MIPI PHY Memory Map/Register Definition

13.7.8.1 MIPI DPHY Register Definition

MIPI D-PHY Registers are accessed from Media Block Control module memory map registers.

13.8 LVDS Display Bridge (LDB)

13.8.1 Overview

The LVDS Display Bridge (LDB) connects to an External LVDS Display Interface.

The purpose of the LDB is to support flow of synchronous RGB data to external display devices through the LVDS interface.

13.8.1.1 Block Diagram

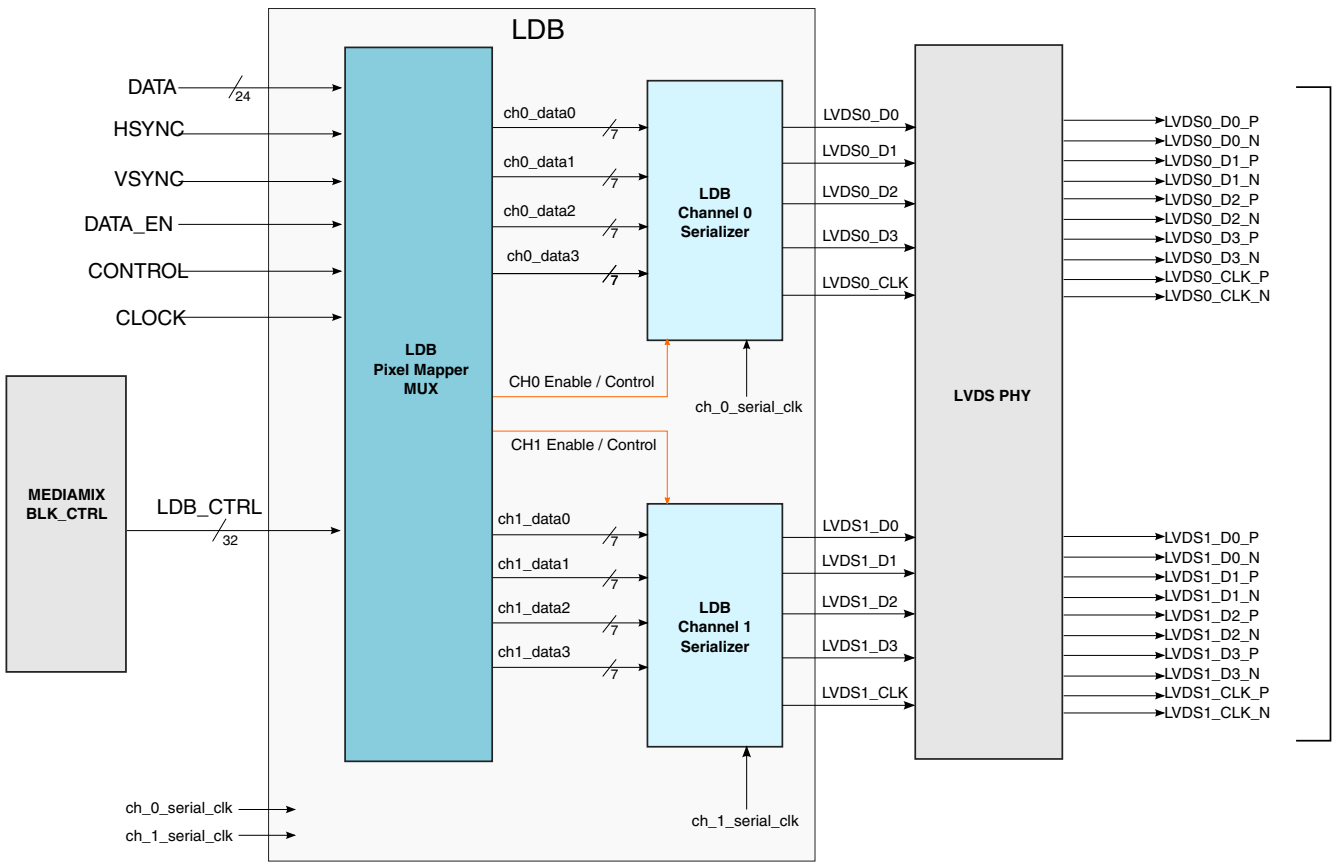


Figure 13-70. LDB Block Diagram

13.8.1.2 Features

LVDS Display Bridge block provides the following functionalities:

- Connectivity to relevant devices - Displays with LVDS receivers.
- Arranging the data as required by the external display receiver and by LVDS display standards.
- Synchronization and control capabilities.

13.8.2 Functional Description

The chip supports LVDS Tx display and Pixel Mapper. The LVDS port can be used as follows:

- Single channel (4 lanes) output at up to 80MHz pixel clock and LVDS clock. This supports resolutions up to 1366x768p60.
- For 4-lane LVDS, channel 0 or channel 1 can be used.
- Dual asynchronous channels (8 data, 2 clocks). This is intended for a single panel with two interfaces, transferring across two channels (even pixel/odd pixel). This is supported at up to 160MHz pixel clock, which is up to 80MHz LVDS clock (due to 2 pixels per LVDS clock). This supports resolutions above 1366x768p60, up to 1080p60.

The Pixel Mapper splits and reorders the pixels from the single LCDIF display output into an odd and even pixel stream. Splitting and reordering is required to match the LVDS Displays speed and channel requirements. Both VESA and JEIDA pixel mapping is supported.

The pixel mapper supports the following modes:

Table 13-35. Pixel Mapper Modes

Use Case	LVDS Channel 0	LVDS Channel 1
Single0	Display Interface(DI) of LCDIF	Disabled
Single1	Disabled	Display Interface(DI) of LCDIF
Dual	Display Interface(DI) of LCDIF	Display Interface(DI) of LCDIF
Split	Display Interface(DI) of LCDIF	Display Interface(DI) of LCDIF

13.8.2.1 Clock Signals

Table 13-36. External Signals

Signal	Description
Serial Clock	CHn Serial Clock

13.8.3 External Signals

The following table describes the external signals of LDB:

Table 13-37. LDB External Signals

Signal	Description
LVDS0_CLK_P	LVDS0 Positive Clock Signal
LVDS0_CLK_N	LVDS0 Negative Clock Signal
LVDS0_D[3:0]_P	LVDS0 Positive Data Signal
LVDS0_D[3:0]_N	LVDS0 Negative Data Signal
LVDS1_CLK_P	LVDS1 Positive Clock Signal
LVDS1_CLK_N	LVDS1 Negative Clock Signal
LVDS1_D[3:0]_P	LVDS1 Positive Data Signal
LVDS1_D[3:0]_N	LVDS1 Negative Data Signal

13.8.4 Register Map

Please refer to the MEDIAMIX BLK_CTL register (LDB_CTRL) for information to configure the Pixel Mapper modes, data widths, and data mappings.

Please refer to the MEDIAMIX BLK_CTL register (LVDS_CTRL) for information to configure band gap adjustment, output transition time and current adjustments.

13.9 HDMI TX Controller

13.9.1 Overview

The High Definition Multimedia Interface (HDMI) TX is a wired digital interconnect that replaces the analog TV out or VGA out. HDMI is capable of transferring uncompressed video, audio, and data using a single cable, compatible with the HDMI v2.0a spec. The video pixel rates are typically from 25 MHz up to 297MHz . HDMI can support a number of audio standards. It can support up to eight IEC60958 L-PCM audio channels, IEC61937 compressed non-linear PCM and HBR (high bit rate) audio interface. HDMI has the capability of automatically setting the display format configuration (intelligent link).

NOTE

The information within this block guide is Synopsys Proprietary. Used with permission.

13.9.1.1 Block diagram

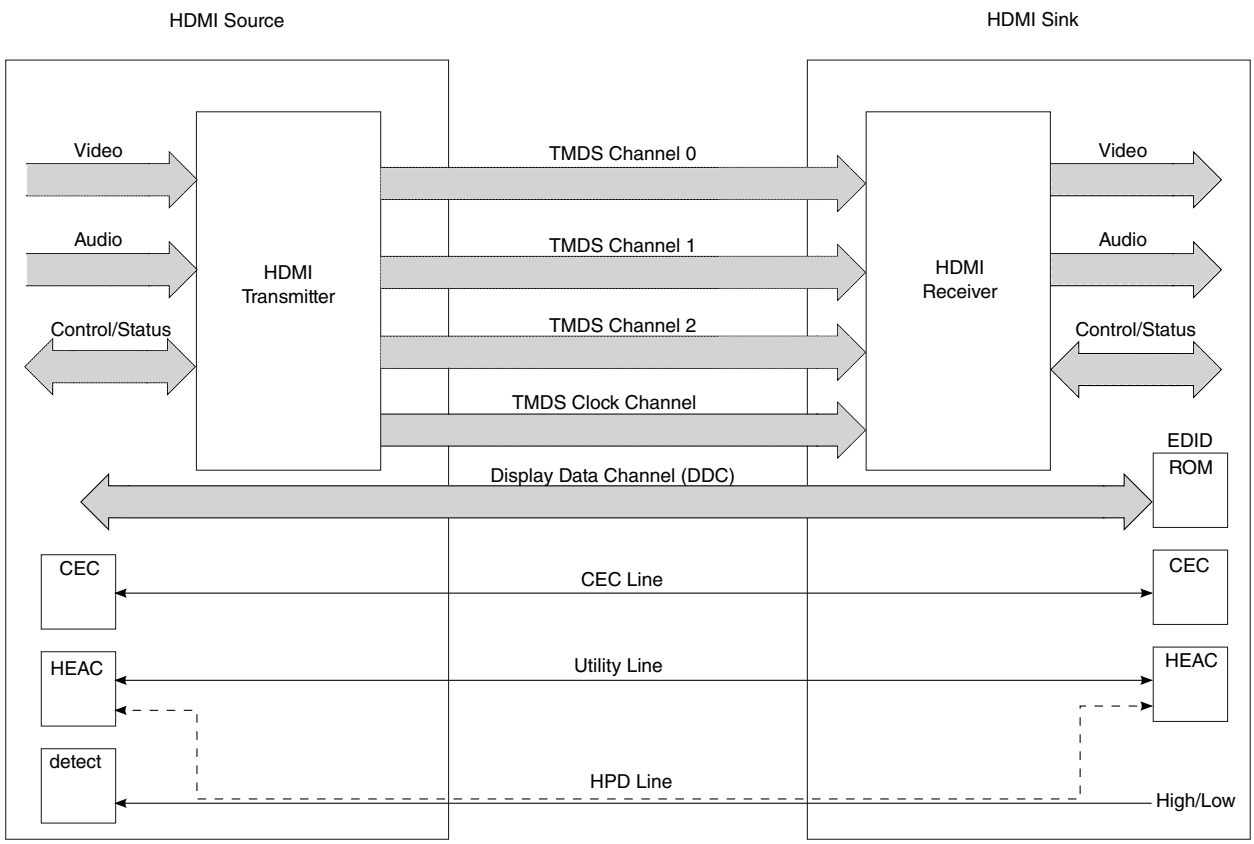


Figure 13-71. HDMI TX Controller Block Diagram

13.9.1.2 Features

HDMI TX includes the following features:

- Compatible with HDMI v2.0a spec
- Decode/encode the following video modes:
  - All 2D video formats described in the CEA-861-F specification up to 4Kp30.
- Decode/encode the following audio modes:
  - IEC 60958 or IEC 61937 Frames with an audio packet of type 0x02:

- Packets of type 0x02:
  - Layout 0 (stereo or compressed audio)
  - Layout 1 (multi-channel up to 8 channels)
- Packets of type 0x0E (Multi-stream Audio Samples)
  - Contains 2 or more streams

- High Bit Rate (HBR)
  - Packtes of type 0x09
    - DolbyTrueHD and DTS HD Master Audio formats

NOTE

HDCP is not supported.

13.9.2 Functional Description

The HDMI TX provides a variety of standard audio, video, and system interfaces.

Figure below illustrates the top level diagram of the HDMI TX solution.

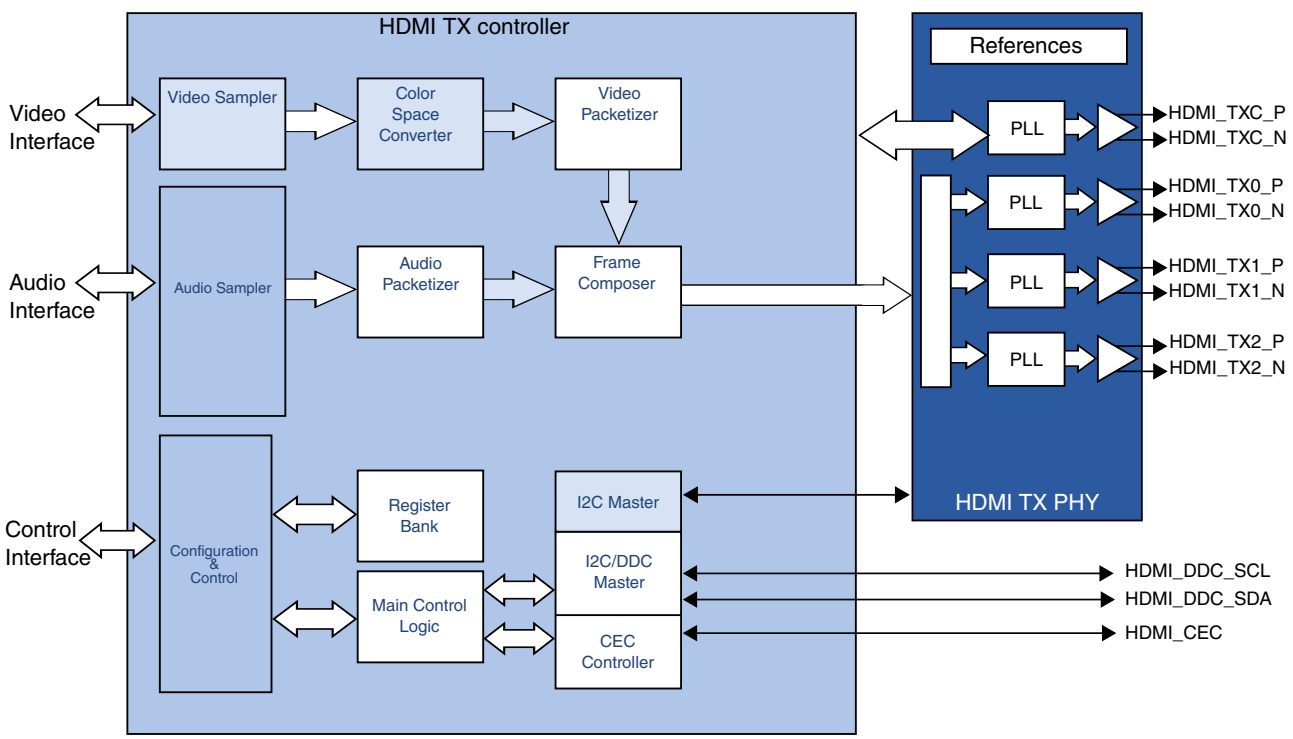


Figure 13-72. HDMI TX Top Level Block Diagram

The input video stream can be either RGB 4:4:4, YcbCr 4:2:2, or YcbCr 4:44 in single data rate (SDR) bus formats. The video mode's timing format must follow the CEA-861-E specification. An embedded color space conversion allows the pixel color format to be converted on the HDMI source side to match the best with the HDMI sink capabilities.

HDMI_TX can output video and inserts audio up to eight-channels over low resolution video formats by performing automatic pixel repetition over the input video stream.

HDMI system architecture consists of sources (transmitter) and sinks (receiver). The HDMI cable and connectors carry four differential pairs that make up the TMDS data and clock channels. These channels are used to carry video, audio, and auxiliary data. In addition, HDMI carries a VESA Data Display Channel (DDC). The DDC is used for configuration and status exchange between a single source and a single sink. The optional CEC protocol provides high-level control functions between all of the various audiovisual products in a user's environment.

Audio, video, and auxiliary data is transmitted across the three TMDS data channels. TMDS clock running, the video pixel rate is transmitted on the TMDS clock channel and used by the receiver as a frequency reference for data recovery on the three TMDS data channels. Video at the default 24-bit color depth is carried at a TMDS clock rate equal to the pixel clock rate. Higher color depths are carried using a correspondingly higher TMDS clock rate. Video formats with TMDS rates below 25MHz (such as, 13.5MHz for 480i/NTSC) can be transmitted using a pixel-repetition scheme. The video pixels can be encoded in either RGB, YCBCR 4:4:4, or YCBCR 4:2:2 formats.

HDMI uses a packet structure to transmit audio and auxiliary data across the TMDS channels. To attain the highest reliability required of audio and control data, this data is protected with a BCH error correction code and is encoded using a special error reduction code to produce the transmitted 10-bit word.

Basic audio functionality consists of a single IEC 60958 L-PCM audio stream (two audio channels) at sample rates of 32 KHz, 44.1 KHz, or 48 KHz, which can accommodate any normal stereo stream. Optionally, HDMI can carry L-PCM audio at sample rates up to 192KHz and with three to eight audio channels. HDMI can also carry an IEC 61937 compressed (such as, surround sound) audio stream at bit rates up to 49.152 Mbps. For bit rates above 6.144 Mbps, compressed audio streams conforming to IEC 61937 are carried using HBR Audio Stream Packets. Each packet carries four IEC 60958 frames, which corresponds to (4x2x16 =) 128 contiguous bits of an IEC 61937 stream.

The source uses the DDC to read the sink's Enhanced Extended Display Identification Data (E-EDID) to obtain the sink's configuration and/or capabilities.

The HDMI TX Controller schedules the three periods: Video Data Period, Data Island period, and Control period. During the Video Data Period, the active pixels of an active video line are transmitted. During the Data Island period, audio and auxiliary data are transmitted using a series of packets. The Control period is used when no video, audio, or auxiliary data needs to be transmitted. A Control Period is required between any two periods that are not Control Periods.

An example of each period placement is shown in the figure below.

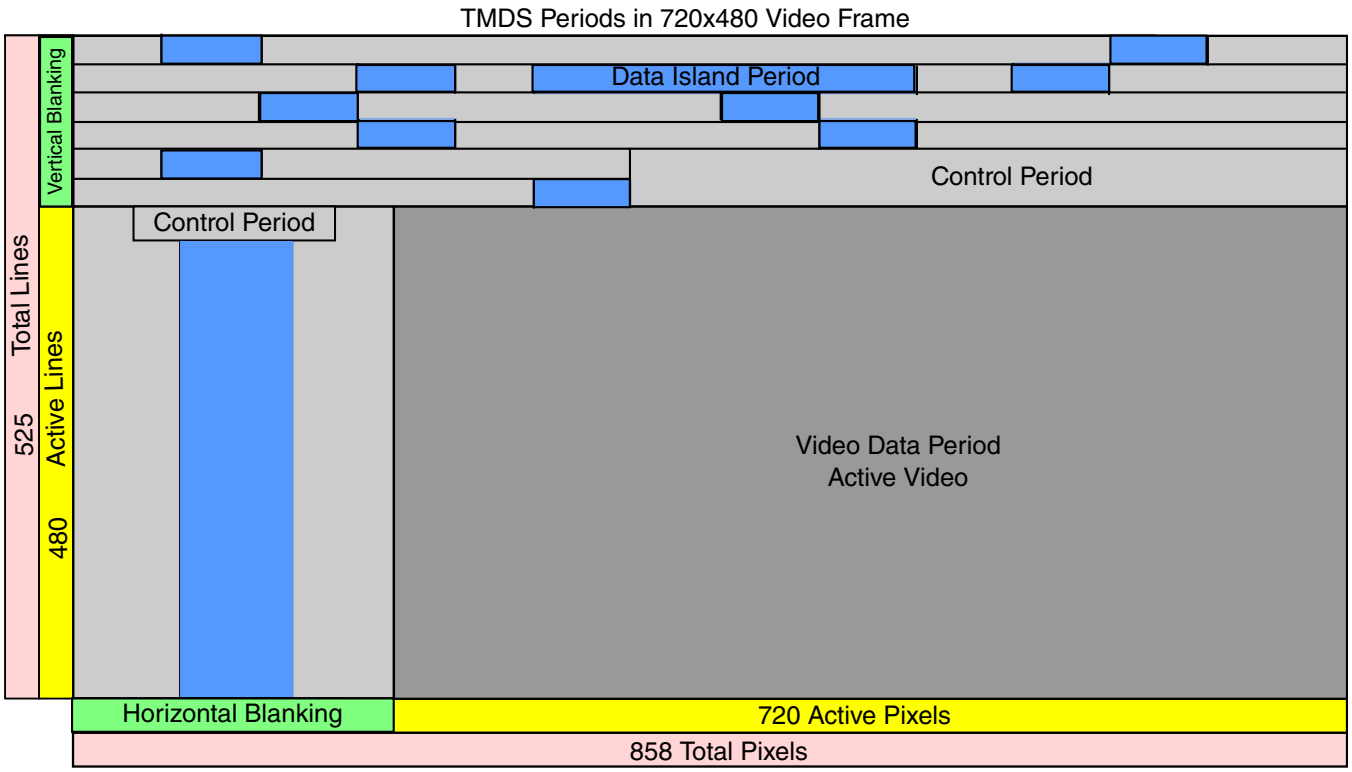


Figure 13-73. TMDS Periods in 720x480p Video Frame

13.9.2.1 Video Pixel Sampler

The Video pixel sampler block is responsible for the video data synchronization, according to the video data input mapping defined by the Color Depth (Deep Color) and format configuration.

The video pixel sampler registers base address is 0x0200 (relative to the HDMI_TX Controller base address).

For each video timing format, there is a specific timing parameters defined in the CEA-861-E specification.

The following timing diagram is an example for the video mode format 1 (640x480p @ 59.94/60 Hz): Data Enable = idataen, HSYNC = ihsync, VSYNC = ivsync.

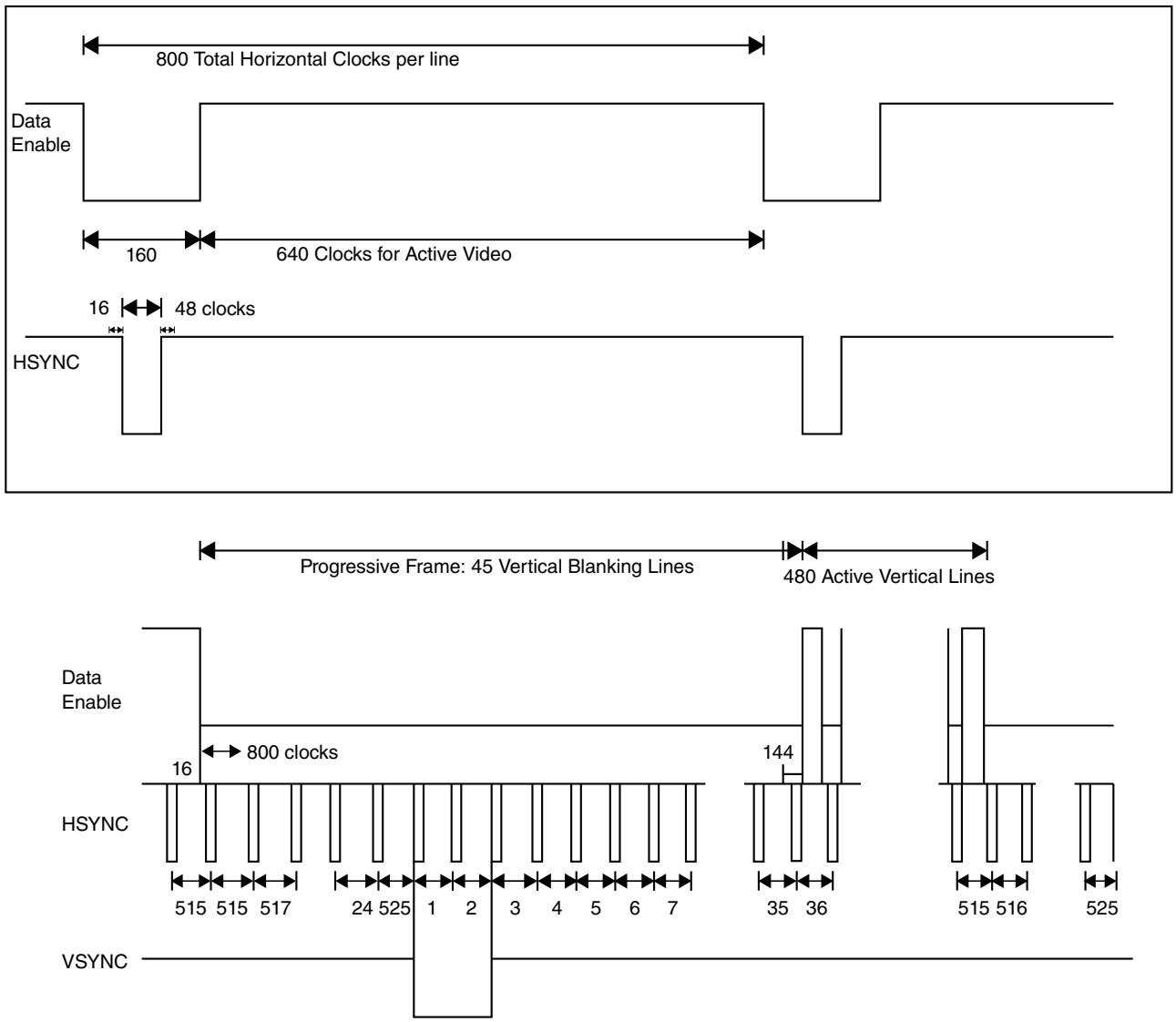


Figure 13-74. Timing Parameters for 640x480p @ 59.94/60 Hz

For a complete list of timing parameters and diagrams, refer to the CEA-861-E specification. The SDR video sample input format is illustrated in the figure below.

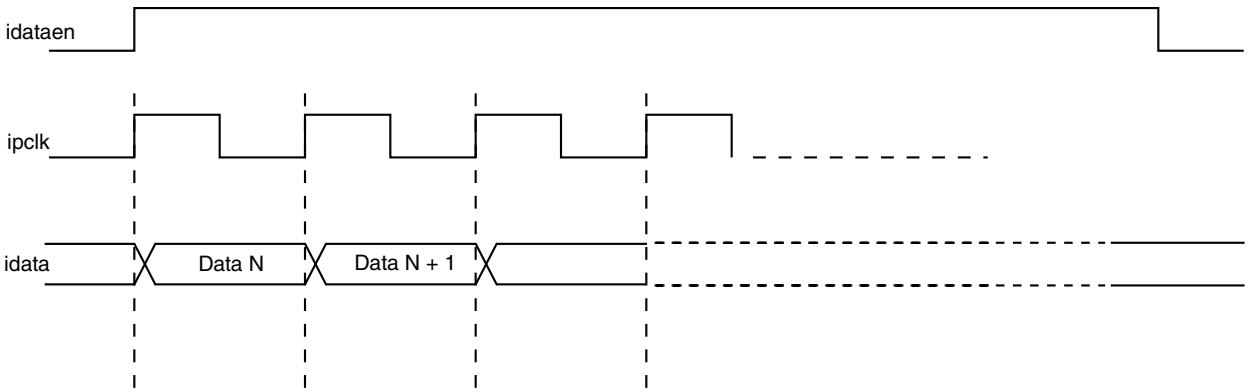


Figure 13-75. Video Sample Timing Interface for RGB and YCbCr SDR Format

13.9.2.2 Video Packetizer

This block is responsible for:

- Pixel repetition (if not already performed in the input video stream and needed by the user)
- 10-bit, 12-bit, and 16-bit packing when in deep color modes
- YCC 422 remapping according to the HDMI specification
- Clock rate transformation from pixel or repetition clock to the final TMDS clock domain (by means of FIFOs)

13.9.2.3 Color Space Conversion

This block is responsible for carrying out the following video color space conversion functions:

- RGB to/from YCbCr
- 4:2:2 to/from 4:4:4 up (pixel repetition or linear interpolation)/down-converter
- Limited to/from full quantization range conversion

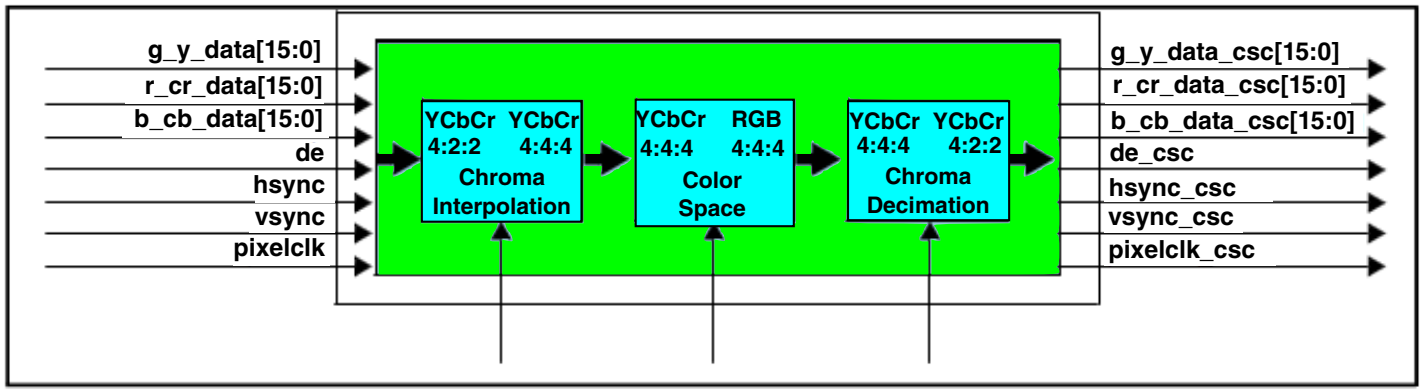


Figure 13-76. Color Space Converter Simplified Block Diagram

The color space conversion matrix is ruled by the following equations listed below. The color space conversion registers base address is 0x4100 (relative to the HDMI_TX Controller base address).

$$out_1 = (X_1 \times in_1 /4096 + X_2 \times in_2 /4096 + X_3 \times in_3 /4096 + X_4) \times 2^{(scale-2)}$$

$$out_2 = (Y_1 \times in_1 /4096 + Y_2 \times in_2 /4096 + Y_3 \times in_3 /4096 + Y_4) \times 2^{(scale-2)}$$

$$out_3 = (Z_1 \times in_1 /4096 + Z_2 \times in_2 /4096 + Z_3 \times in_3 /4096 + Z_4) \times 2^{(scale-2)}$$

13.9.2.4 Audio Interfaces

The supported audio input interface is:

- General Parallel Audio (GPA) Interface

The audio sampler registers base address is 0x3500 (relative to the HDMI_TX Controller base address).

13.9.2.4.1 Generic Parallel Audio Interface

The Generic Parallel Audio interface can be seen as an enhancement of the HBR interface.

It supports:

- Three types of Audio (L-PCM, NL-PCM, HBR).
- No Audio clock required.
- ACR information, N and CTS are configured by software, the samples are clocked-in using the system clock.

13.9.2.5 Frame Composer

This block is responsible for assembling video, audio, and data packets in a consistent frame that are streamed to the HDMI TX PHY.

The HDMI standard precisely describes the packet insertion timing and distribution that must be followed to correctly compose an HDMI TMDS (transition minimized differential signaling) stream. In this context, there are data island packets that-when available (ready for insertion in output stream)-have higher priority over others. Two packet descriptor queues are responsible for prioritizing packet insertion.

The higher priority packets are described in Table below. These packets are inserted in the output stream as soon as data to compose them is available.

Table 13-38. High Priority Data Island Packets

Packet	Description
Audio Clock Regeneration (ACR)	Indicates to sink device the N/CTS values that should be used in the ACR process
Audio Sample (AUDS)	Transports L-PCM and IEC 61937 compressed audio
General Control (GCP)	Indicates Color Depth, Pixel Packing phase, and AV mute information to sink device

The packets described in tables below can be considered as low priority packets-even though they have precise timing insertion-because their insertion timing is large (for example, one per frame or one per two frames without specific location for some of the packet types and on user request transmit for others).

Table 13-39. Low Priority Data Island Packets

Packet	Description
Audio Content Protection (ACP)	Used to convey content-related information about the active audio stream transmitted
Audio InfoFrame (AUDI)	Indicates characteristics of the active audio stream by using IEC 60958 channel status bits, IEC 61937 burst info, and/or stream data (if present).

Table continues on the next page...

Table 13-39. Low Priority Data Island Packets (continued)

Null (NULL)	Ignored by sink devices.
Vendor Specific (VSD) InfoFrame	According to CEA-861-E standard.
AVI infoFrame (AVI)	Video information from source to sink.
Source Data Product Descriptor (SPD) infoFrame	Name and product type of the source device.

The Frame Composer distributes and assembles the data island packets according to the module register bank configuration. The block allows extended control periods to appear with a certain programmed spacing. The Frame Composer uses two packet buffers that allow a packet to be composed while another is being sent to the output HDMI stream.

Packet requests are inserted into the packet queues by a data island flexible scheduler. The HDMI 1.3a specification requires that packet distribution and insertion timing correctly compose an output HDMI TMDS stream. In this context, there are data island packets that are sent on data availability, while others are sent once per frame or once per two frames. and finally others that are sent on user request. Classification of the packets according to this insertion timing is described in the table below.

Table 13-40. Packet Classification

Packet	Classification
Audio Clock Regeneration (ACR)	Sent on data availability.
Audio Sample (AUDS)	Sent on data availability (precede ACR if present).
Audio Content Protection (ACP)	On user request or automatic insertion.
Audio InfoFrame (AUDI)	Once per two frames.
Null (NULL)	On user request or automatic insertion to fill Data Island period.
General Control (GCP)	Once per frame.
International Standard Recording Code (ISRC1/ ISRC2)	On user request.
Vendor Specific (VSD) InfoFrame	On user request or automatic insertion.
AVI infoFrame (AVI)	Once per frame.
Source Data Product Descriptor (SPD) infoFrame	On user request or automatic insertion.

The Data Island Scheduler (DIS) handles packet distribution in the Frame Composer. The DIS is a round- robin (RDRB) state machine that is able to schedule packet insertion on an input video frame or line basis. The DIS is fully configurable and can schedule any packet type to be inserted at a given input video frame rate or input video line rate.

While determining packet distribution on an input video frame or line basis, the DIS schedules the packets to be inserted in the output HDMI stream by inserting the packet descriptor in the corresponding packet priority queue, according to packet priority classification.

After the packet descriptor has been inserted in the packet priority queues, the Data Island Packer (DIP) is responsible for assembling and sequencing the packets for output HDMI stream insertion.

Dedicated ECC generators and checksum byte-wide sum hardware generate the BCH ECC parity codes and infoFrames checksums for all the data islands packets.

The content of of supported packets are configured through the registers bank starting at address 0x1000 (relative to the HDMI_TX Controller base address).

13.9.2.6 EDID I²C E-DDC Interface

The Enhanced Display Data Channel (E-DDC) channel is a dedicated I2C master interface that allows the read of sink E-EDID based on system needs.

Data read from sink E-EDID can be then transferred to the I2C Master register bank, starting at address 0x7E00 (relative to the HDMI_TX Controller base address).

13.9.2.7 CEC Hardware Engine

Consumer Electronics Control (CEC) is a protocol that provides high-level control functions between all of the various audiovisual products in a user's environment.

It is an optional feature in the HDMI Specification. It uses only one bidirectional line for transmission and reception.

All transactions on the CEC line consist of an initiator and one or more followers. The initiator is responsible for sending the message structure and the data. The follower is the recipient of any data and is responsible for setting any acknowledgement bits.

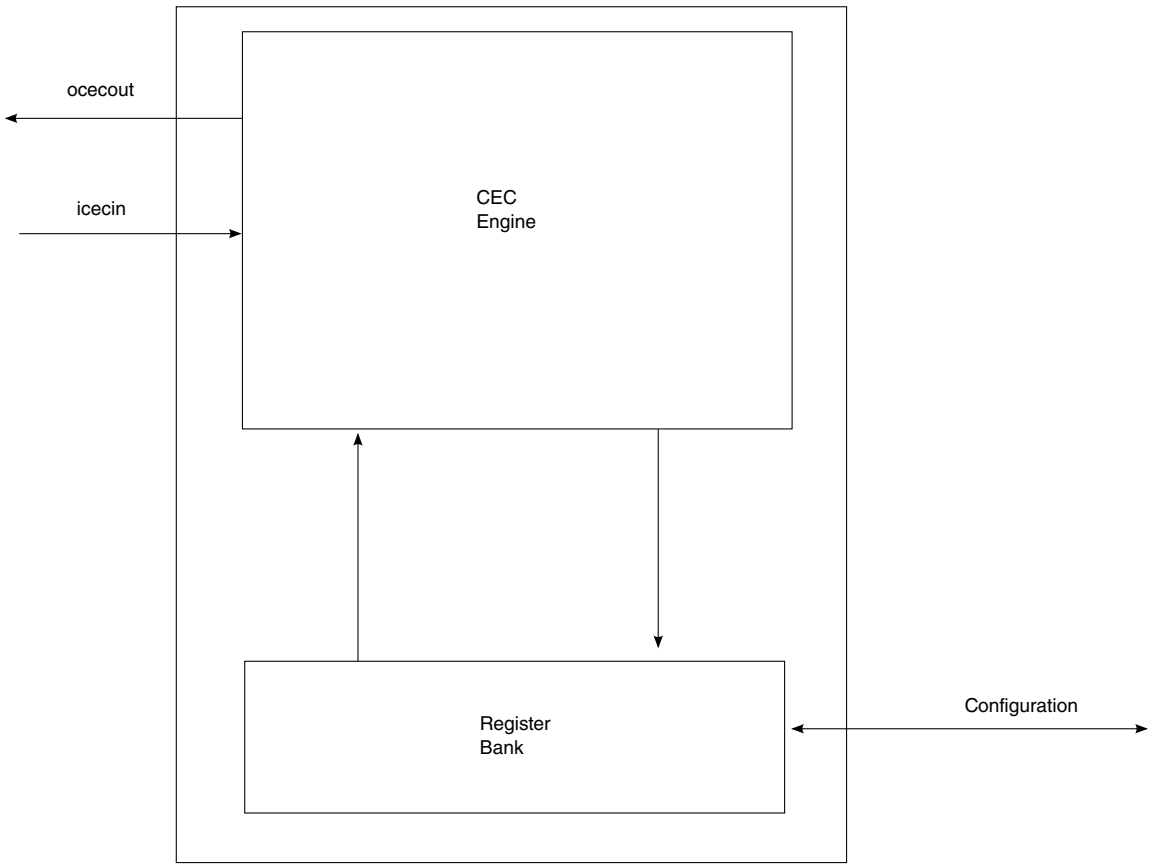


Figure 13-77. CEC Engine Simplified Block Diagram

There are two operation modes for a CEC controller.

- Initiator Mode
  - In this mode, the CEC controller sends messages out and waits for a follower to feedback. The CEC controller works in this mode when it starts to send a frame. After the transmission is done, it automatically returns to the follower mode (no software control involved).
- Follower Mode
  - In this mode, the CEC controller receives messages and feeds back the initiator with appropriate signals. The CEC controller always works in the follower mode whenever it is not transmitting any data.

The CEC engine registers base address is 0x7D00 (relative to the HDMI_TX Controller base address).

13.9.3 Memory Map and register definition

This section includes the HDMI_TX Controller module memory map and detailed descriptions of all registers.

Each set of registers in the following sections has its own base offset, which must be added to the HDMI_TX Controller base address shown below.

HDMI_TX Controller base address: 32FD_8000h

NOTE

The information within this block guide is Synopsys Proprietary. Used with permission.

13.9.3.1 Identification register descriptions

13.9.3.1.1 Memory map

Identification registers base address: 0h

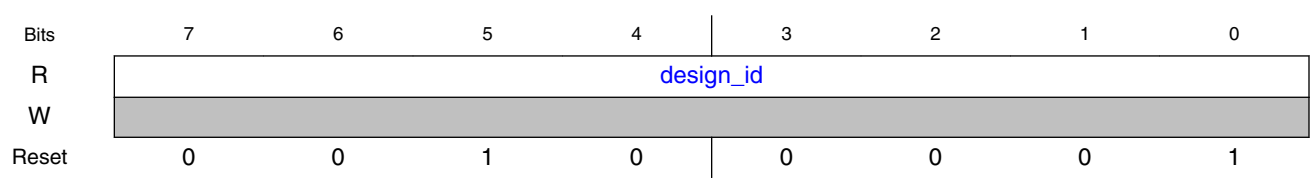
Offset	Register	Width (In bits)	Access	Reset value
0h	Design Identification Register (design_id)	8	RO	21h

13.9.3.1.2 Design Identification Register (design_id)

13.9.3.1.2.1 Offset

Register	Offset
design_id	0h

13.9.3.1.2.2 Diagram



13.9.3.1.2.3 Fields

Field	Function
7-0 design_id	Design ID code fixed by Synopsys that Identifies the instantiated DWC_hdmi_tx controller. Design ID code fixed by Synopsys that Identifies the instantiated DWC_hdmi_tx controller. For example, DWC_hdmi_tx 2.13a, DESIGN_ID = 21

13.9.3.2 Interrupt register descriptions

13.9.3.2.1 Memory map

Interrupt registers base address: 100h

Offset	Register	Width (In bits)	Access	Reset value
0h	Frame Composer Interrupt Status Register 0 (Packet Interrupts) (ih_fc_stat0)	8	W1C	00h
1h	Frame Composer Interrupt Status Register 1 (Packet Interrupts) (ih_fc_stat1)	8	W1C	00h
2h	Frame Composer Interrupt Status Register 2 (Packet Interrupts) (ih_fc_stat2)	8	W1C	00h
3h	Audio Sampler Interrupt Status Register (FIFO Threshold, Underflow and Overflow Interrupts) (ih_as_stat0)	8	W1C	00h
4h	PHY Interface Interrupt Status Register (RXSENSE, PLL Lock and HPD Interrupts) (ih_phy_stat0)	8	W1C	00h
5h	E-DDC I2C Master Interrupt Status Register (Done and Error Interrupts) (ih_i2cm_stat0)	8	W1C	00h
6h	CEC Interrupt Status Register (Functional Operation Interrupts) (ih_cec_stat0)	8	W1C	00h
7h	Video Packetizer Interrupt Status Register (FIFO Full and Empty Interrupts) (ih_vp_stat0)	8	W1C	00h
8h	PHY GEN2 I2C Master Interrupt Status Register (Done and Error Interrupts) (ih_i2cmphy_stat0)	8	W1C	00h
70h	Interruption Handler Decode Assist Register (ih_decode)	8	RO	00h
80h	Frame Composer Interrupt Mute Control Register 0 (ih_mute_fc_stat0)	8	RW	18h
81h	Frame Composer Interrupt Mute Control Register 1 (ih_mute_fc_stat1)	8	RW	04h
82h	Frame Composer Interrupt Mute Control Register 2 (ih_mute_fc_stat2)	8	RW	10h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
83h	Audio Sampler Interrupt Mute Control Register (ih_mute_as_stat0)	8	RW	08h
84h	PHY Interface Interrupt Mute Control Register (ih_mute_phy_stat0)	8	RW	00h
85h	E-DDC I2C Master Interrupt Mute Control Register (ih_mute_i2cm_stat0)	8	RW	04h
86h	CEC Interrupt Mute Control Register (ih_mute_cec_stat0)	8	RW	00h
87h	Video Packetizer Interrupt Mute Control Register (ih_mute_vp_stat0)	8	RW	00h
88h	PHY GEN2 I2C Master Interrupt Mute Control Register (ih_mute_i2cmphy_stat0)	8	RW	00h
FFh	Global Interrupt Mute Control Register (ih_mute)	8	RW	03h

13.9.3.2.2 Frame Composer Interrupt Status Register 0 (Packet Interrupts) (ih_fc_stat0)

13.9.3.2.2.1 Offset

Register	Offset
ih_fc_stat0	0h

13.9.3.2.2.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	AUDI	ACP	HBR	MAS	NVBI	AUDS	ACR	NULL
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0

13.9.3.2.2.3 Fields

Field	Function
7 AUDI	Active after successful transmission of an Audio InfoFrame packet.
6 ACP	Active after successful transmission of an Audio Content Protection packet.
5	Active after successful transmission of an Audio HBR packet.

Table continues on the next page...

Field	Function
HBR	
4 MAS	Active after successful transmission of an MultiStream Audio packet
3 NVBI	Active after successful transmission of an NTSC VBI packet
2 AUDS	Active after successful transmission of an Audio Sample packet. Active after successful transmission of an Audio Sample packet. Due to high number of audio sample packets transmitted, this interrupt is by default masked at frame composer.
1 ACR	Active after successful transmission of an Audio Clock Regeneration (N/CTS transmission) packet.
0 NULL	Active after successful transmission of an Null packet. Active after successful transmission of an Null packet. Due to high number of audio sample packets transmitted, this interrupt is by default masked at frame composer.

13.9.3.2.3 Frame Composer Interrupt Status Register 1 (Packet Interrupts) (ih_fc_stat1)

13.9.3.2.3.1 Offset

Register	Offset
ih_fc_stat1	1h

13.9.3.2.3.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	GMD	ISCR1	ISCR2	VSD	SPD	AMP	AVI	GCP
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0

13.9.3.2.3.3 Fields

Field	Function
7 GMD	Active after successful transmission of an Gamut metadata packet.

Table continues on the next page...

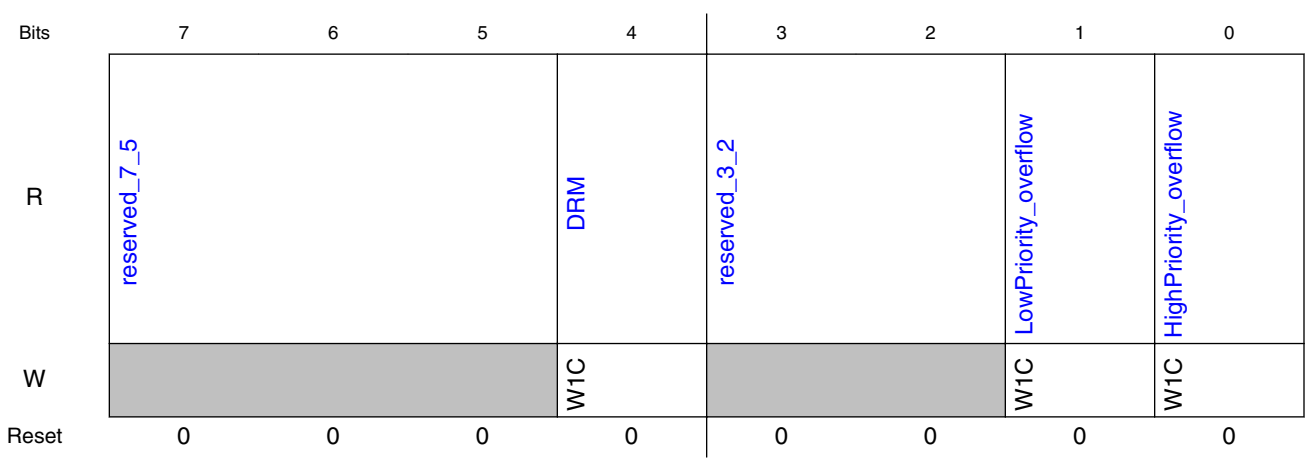
Field	Function
6 ISCR1	Active after successful transmission of an International Standard Recording Code 1 packet.
5 ISCR2	Active after successful transmission of an International Standard Recording Code 2 packet
4 VSD	Active after successful transmission of an Vendor Specific Data InfoFrame packet.
3 SPD	Active after successful transmission of an Source Product Descriptor InfoFrame packet.
2 AMP	Active after successful transmission of an Audio Metadata packet
1 AVI	Active after successful transmission of an AVI InfoFrame packet.
0 GCP	Active after successful transmission of an General Control Packet.

13.9.3.2.4 Frame Composer Interrupt Status Register 2 (Packet Interrupts) (ih_fc_stat2)

13.9.3.2.4.1 Offset

Register	Offset
ih_fc_stat2	2h

13.9.3.2.4.2 Diagram



13.9.3.2.4.3 Fields

Field	Function
7-5 reserved_7_5	Reserved and read as zero
4 DRM	Active after successful transmission of an DRM packet
3-2 reserved_3_2	Reserved and read as zero
1 LowPriority_overflow	Frame Composer low priority packet queue descriptor overflow indication
0 HighPriority_overflow	Frame Composer high priority packet queue descriptor overflow indication

13.9.3.2.5 Audio Sampler Interrupt Status Register (FIFO Threshold, Underflow and Overflow Interrupts) (ih_as_stat0)

13.9.3.2.5.1 Offset

Register	Offset
ih_as_stat0	3h

13.9.3.2.5.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	reserved_7_5			reserved_4	fifo_overrun	Aud_fifo_underflow_thr	Aud_fifo_underflow	Aud_fifo_overflow
W					W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0

13.9.3.2.5.3 Fields

Field	Function
7-5 reserved_7_5	Reserved and read as zero
4 reserved_4	Reserved and read as zero
3 fifo_overrun	Indicates an overrun on the audio FIFO.
2 Aud_fifo_underflow_thr	Audio Sampler audio FIFO empty threshold (four samples) indication for the legacy HBR audio interface. Audio Sampler audio FIFO empty threshold (four samples) indication for the legacy HBR audio interface. For AHB_DMA, this bit indicates that the number of samples in the FIFO is equal to (or less) than the number of active audio channels. This bit is not relevant for I2S, SPDIF, and GPA interfaces.
1 Aud_fifo_underflow	Audio Sampler audio FIFO empty indication.
0 Aud_fifo_overflow	Audio Sampler audio FIFO full indication.

13.9.3.2.6 PHY Interface Interrupt Status Register (RXSENSE, PLL Lock and HPD Interrupts) (ih_phy_stat0)

13.9.3.2.6.1 Offset

Register	Offset
ih_phy_stat0	4h

13.9.3.2.6.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	reserved_7_6		RX_SENSE_3	RX_SENSE_2	RX_SENSE_1	RX_SENSE_0	TX_PHY_LOCK	HPD
W			W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0

13.9.3.2.6.3 Fields

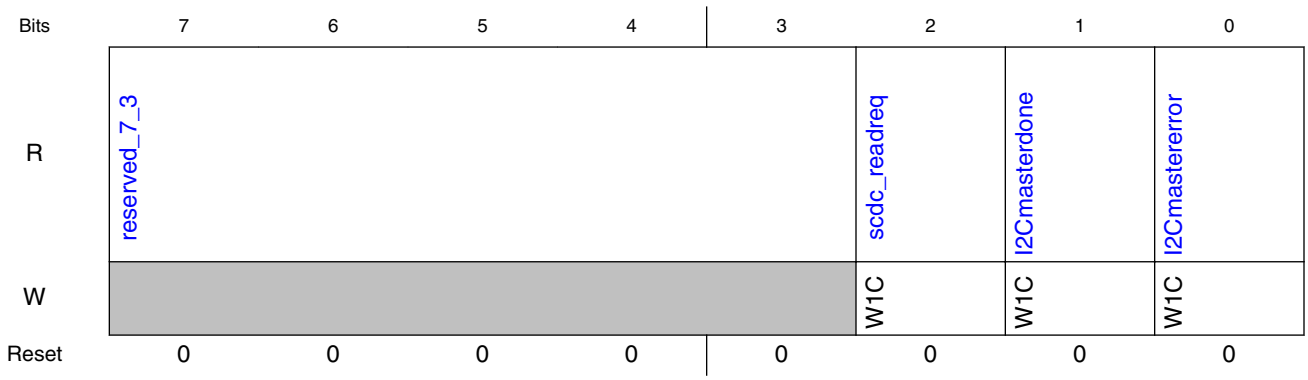
Field	Function
7-6 reserved_7_6	Reserved and read as zero
5 RX_SENSE_3	TX PHY RX_SENSE indication for driver 3. TX PHY RX_SENSE indication for driver 3. You may need to mask or change polarity of this interrupt after it has become active.
4 RX_SENSE_2	TX PHY RX_SENSE indication for driver 2. TX PHY RX_SENSE indication for driver 2. You may need to mask or change polarity of this interrupt after it has become active.
3 RX_SENSE_1	TX PHY RX_SENSE indication for driver 1. TX PHY RX_SENSE indication for driver 1. You may need to mask or change polarity of this interrupt after it has become active.
2 RX_SENSE_0	TX PHY RX_SENSE indication for driver 0. TX PHY RX_SENSE indication for driver 0. You may need to mask or change polarity of this interrupt after it has become active.
1 TX_PHY_LOCK	TX PHY PLL lock indication. TX PHY PLL lock indication. For more information, refer to the PHY databook. You may need to mask or change polarity of this interrupt after it has become active.
0 HPD	HDMI Hot Plug Detect indication. HDMI Hot Plug Detect indication. You may need to mask or change polarity of this interrupt after it has become active.

13.9.3.2.7 E-DDC I2C Master Interrupt Status Register (Done and Error Interrupts) (ih_i2cm_stat0)

13.9.3.2.7.1 Offset

Register	Offset
ih_i2cm_stat0	5h

13.9.3.2.7.2 Diagram



13.9.3.2.7.3 Fields

Field	Function
7-3 reserved_7_3	Reserved and read as zero
2 scdc_readreq	I2C Master SCDC read request indication.
1 I2Cmasterdone	I2C Master done indication
0 I2Cmastererror	I2C Master error indication

13.9.3.2.8 CEC Interrupt Status Register (Functional Operation Interrupts) (ih_cec_stat0)

13.9.3.2.8.1 Offset

Register	Offset
ih_cec_stat0	6h

13.9.3.2.8.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	reserved_7	WAKEUP	ERROR_FOLLOW	ERROR_INITIATOR	ARB_LOST	NACK	EOM	DONE
W		W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0

13.9.3.2.8.3 Fields

Field	Function
7 reserved_7	Reserved and read as zero
6 WAKEUP	CEC Wake-up indication
5 ERROR_FOLLOW	CEC Error Follow indication
4 ERROR_INITIATOR	CEC Error Initiator indication
3 ARB_LOST	CEC Arbitration Lost indication
2 NACK	CEC Not Acknowledge indication
1 EOM	CEC End of Message Indication
0 DONE	CEC Done Indication

13.9.3.2.9 Video Packetizer Interrupt Status Register (FIFO Full and Empty Interrupts) (ih_vp_stat0)

13.9.3.2.9.1 Offset

Register	Offset
ih_vp_stat0	7h

13.9.3.2.9.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	fifofullrepet	fifoemptyrepet	fifofullpp	fifoemptypp	fifofullremap	fifoemptyremap	reserved_1	reserved_0
W	W1C	W1C	W1C	W1C	W1C	W1C		
Reset	0	0	0	0	0	0	0	0

13.9.3.2.9.3 Fields

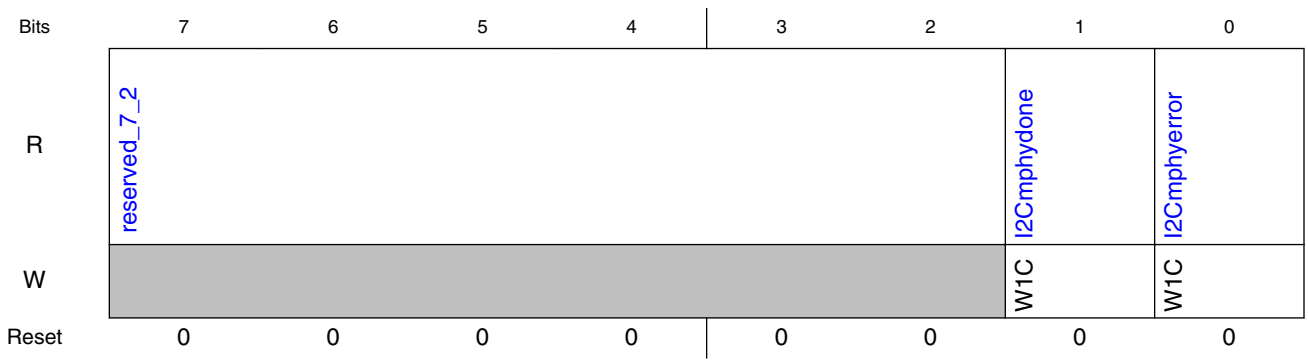
Field	Function
7 fifofullrepet	Video Packetizer pixel repeater FIFO full interrupt
6 fifoemptyrepet	Video Packetizer pixel repeater FIFO empty interrupt
5 fifofullpp	Video Packetizer pixel packing FIFO full interrupt
4 fifoemptypp	Video Packetizer pixel packing FIFO empty interrupt
3 fifofullremap	Video Packetizer pixel YCbCr 422 re-mapper FIFO full interrupt
2 fifoemptyremap	Video Packetizer pixel YCbCr 422 re-mapper FIFO empty interrupt
1 reserved_1	Reserved and read as zero
0 reserved_0	Reserved and read as zero

13.9.3.2.10 PHY GEN2 I2C Master Interrupt Status Register (Done and Error Interrupts) (ih_i2cmphy_stat0)

13.9.3.2.10.1 Offset

Register	Offset
ih_i2cmphy_stat0	8h

13.9.3.2.10.2 Diagram



13.9.3.2.10.3 Fields

Field	Function
7-2 reserved_7_2	Reserved and read as zero
1 I2Cmphydone	I2C Master PHY done indication
0 I2Cmphyerror	I2C Master PHY error indication

13.9.3.2.11 Interruption Handler Decode Assist Register (ih_decode)

13.9.3.2.11.1 Offset

Register	Offset
ih_decode	70h

13.9.3.2.11.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	ih_fc_stat0	ih_fc_stat1	ih_fc_stat2_vp	ih_as_stat0	ih_phy	ih_i2cm_stat0	ih_cec_stat0	ih_ahbdmaaud_stat0
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.2.11.3 Fields

Field	Function
7 ih_fc_stat0	Interruption active at the ih_fc_stat0 register
6 ih_fc_stat1	Interruption active at the ih_fc_stat1 register
5 ih_fc_stat2_vp	Interruption active at the ih_fc_stat2 or ih_vp_stat0 register
4 ih_as_stat0	Interruption active at the ih_as_stat0 register
3 ih_phy	Interruption active at the ih_phy_stat0 or ih_i2cmphy_stat0 register
2 ih_i2cm_stat0	Interruption active at the ih_i2cm_stat0 register
1 ih_cec_stat0	Interruption active at the ih_cec_stat0 register
0 ih_ahbdmaaud_stat0	Interruption active at the ih_ahbdmaaud_stat0 register

13.9.3.2.12 Frame Composer Interrupt Mute Control Register 0 (ih_mute_fc_stat0)

13.9.3.2.12.1 Offset

Register	Offset
ih_mute_fc_stat0	80h

13.9.3.2.12.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	AUDI	ACP	HBR	MAS	NVBI	AUDS	ACR	NULL
W								
Reset	0	0	0	1	1	0	0	0

13.9.3.2.12.3 Fields

Field	Function
7 AUDI	When set to 1, mutes ih_fc_stat0[7]
6 ACP	When set to 1, mutes ih_fc_stat0[6]
5 HBR	When set to 1, mutes ih_fc_stat0[5]
4 MAS	When set to 1, mutes ih_fc_stat0[4].
3 NVBI	When set to 1, mutes ih_fc_stat0[3].
2 AUDS	When set to 1, mutes ih_fc_stat0[2]
1 ACR	When set to 1, mutes ih_fc_stat0[1]
0 NULL	When set to 1, mutes ih_fc_stat0[0]

13.9.3.2.13 Frame Composer Interrupt Mute Control Register 1 (ih_mute_fc_stat1)

13.9.3.2.13.1 Offset

Register	Offset
ih_mute_fc_stat1	81h

13.9.3.2.13.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	GMD	ISCR1	ISCR2	VSD	SPD	AMP	AVI	GCP
Reset	0	0	0	0	0	1	0	0

13.9.3.2.13.3 Fields

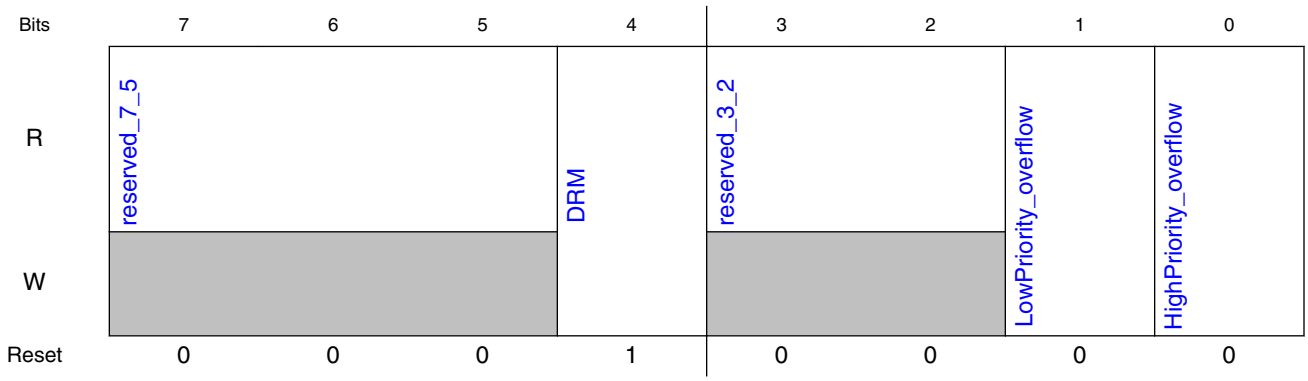
Field	Function
7 GMD	When set to 1, mutes ih_fc_stat1[7]
6 ISCR1	When set to 1, mutes ih_fc_stat1[6]
5 ISCR2	When set to 1, mutes ih_fc_stat1[5]
4 VSD	When set to 1, mutes ih_fc_stat1[4]
3 SPD	When set to 1, mutes ih_fc_stat1[3]
2 AMP	When set to 1, mutes ih_fc_stat1[2].
1 AVI	When set to 1, mutes ih_fc_stat1[1]
0 GCP	When set to 1, mutes ih_fc_stat1[0]

13.9.3.2.14 Frame Composer Interrupt Mute Control Register 2 (ih_mute_fc_stat2)

13.9.3.2.14.1 Offset

Register	Offset
ih_mute_fc_stat2	82h

13.9.3.2.14.2 Diagram



13.9.3.2.14.3 Fields

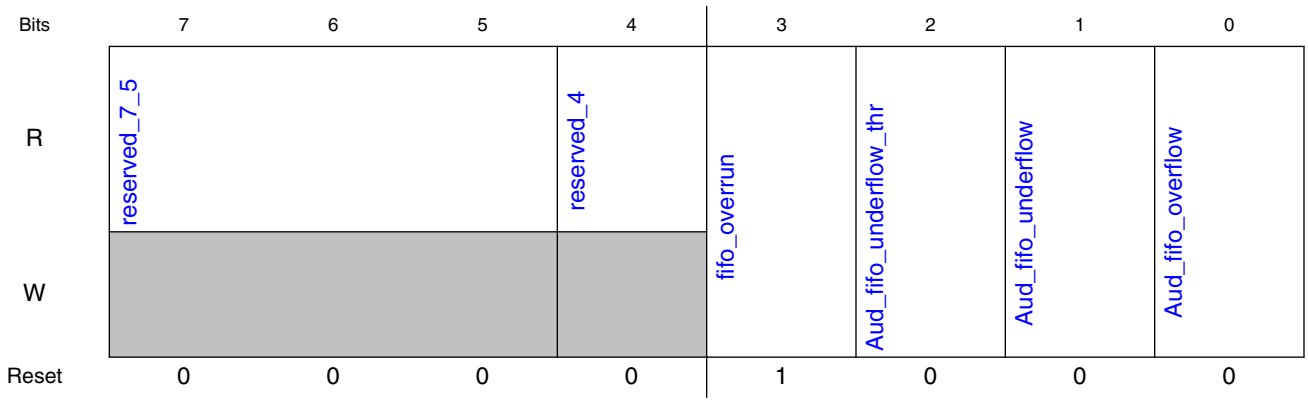
Field	Function
7-5 reserved_7_5	Reserved and read as zero
4 DRM	When set to 1, mutes ih_fc_stat2[4].
3-2 reserved_3_2	Reserved and read as zero
1 LowPriority_overflow	When set to 1, mutes ih_fc_stat2[1]
0 HighPriority_overflow	When set to 1, mutes ih_fc_stat2[0]

13.9.3.2.15 Audio Sampler Interrupt Mute Control Register (ih_mute_as_stat0)

13.9.3.2.15.1 Offset

Register	Offset
ih_mute_as_stat0	83h

13.9.3.2.15.2 Diagram



13.9.3.2.15.3 Fields

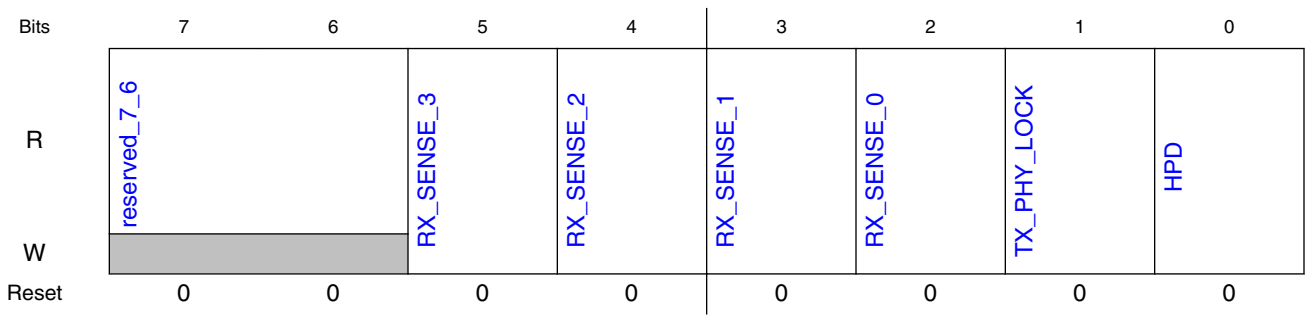
Field	Function
7-5 reserved_7_5	Reserved and read as zero
4 reserved_4	Reserved and read as zero
3 fifo_overrun	When set to 1, mutes ih_as_stat0[3]
2 Aud_fifo_underflow_thr	When set to 1, mutes ih_as_stat0[2]
1 Aud_fifo_underflow	When set to 1, mutes ih_as_stat0[1]
0 Aud_fifo_overflow	When set to 1, mutes ih_as_stat0[0]

13.9.3.2.16 PHY Interface Interrupt Mute Control Register (ih_mute_phy_stat0)

13.9.3.2.16.1 Offset

Register	Offset
ih_mute_phy_stat0	84h

13.9.3.2.16.2 Diagram



13.9.3.2.16.3 Fields

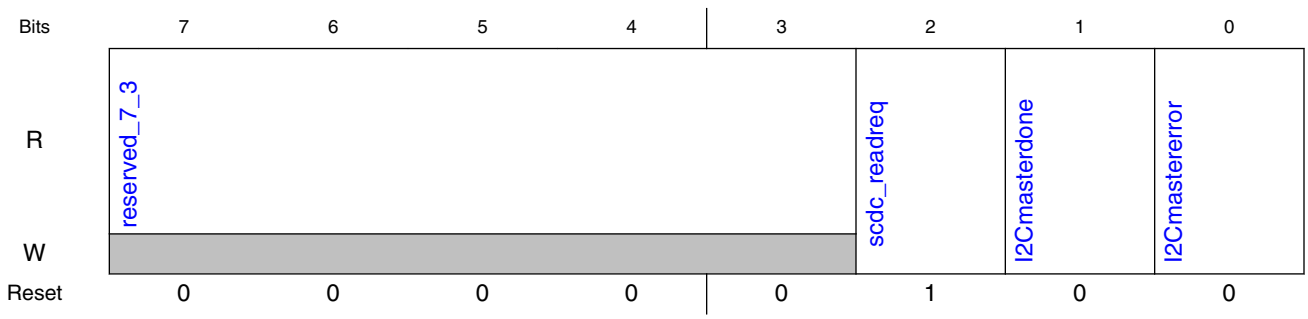
Field	Function
7-6 reserved_7_6	Reserved and read as zero
5 RX_SENSE_3	When set to 1, mutes ih_phy_stat0[5]
4 RX_SENSE_2	When set to 1, mutes ih_phy_stat0[4]
3 RX_SENSE_1	When set to 1, mutes ih_phy_stat0[3]
2 RX_SENSE_0	When set to 1, mutes ih_phy_stat0[2]
1 TX_PHY_LOCK	When set to 1, mutes ih_phy_stat0[1]
0 HPD	When set to 1, mutes ih_phy_stat0[0]

13.9.3.2.17 E-DDC I2C Master Interrupt Mute Control Register (ih_mute_i2cm_stat0)

13.9.3.2.17.1 Offset

Register	Offset
ih_mute_i2cm_stat0	85h

13.9.3.2.17.2 Diagram



13.9.3.2.17.3 Fields

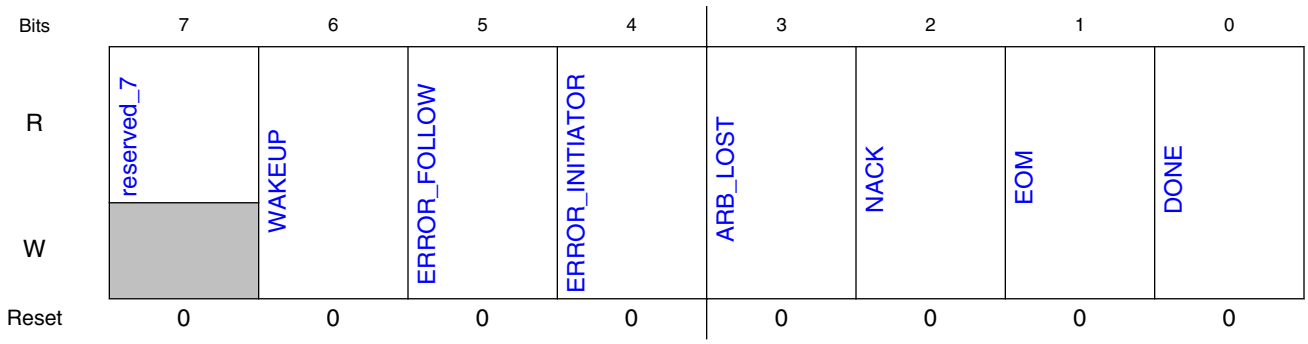
Field	Function
7-3 reserved_7_3	Reserved and read as zero
2 scdc_readreq	When set to 1, mutes ih_i2cm_stat0[2]
1 I2Cmasterdone	When set to 1, mutes ih_i2cm_stat0[1]
0 I2Cmastererror	When set to 1, mutes ih_i2cm_stat0[0]

13.9.3.2.18 CEC Interrupt Mute Control Register (ih_mute_cec_stat0)

13.9.3.2.18.1 Offset

Register	Offset
ih_mute_cec_stat0	86h

13.9.3.2.18.2 Diagram



13.9.3.2.18.3 Fields

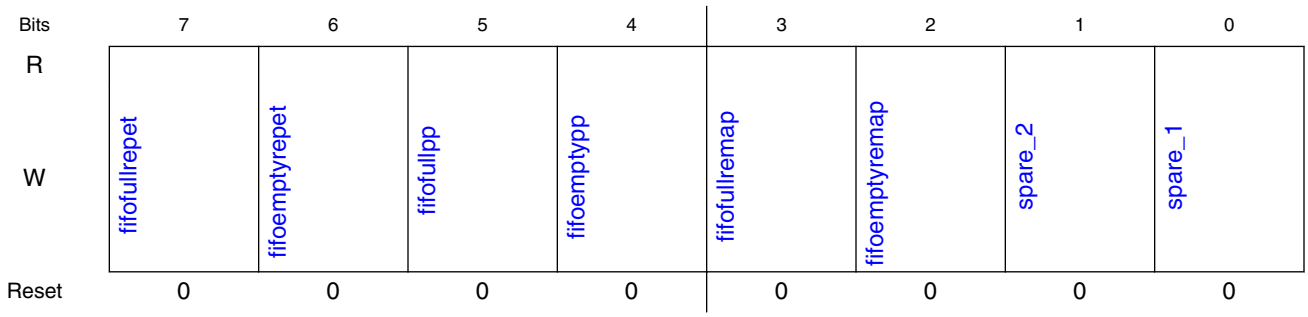
Field	Function
7 reserved_7	Reserved and read as zero
6 WAKEUP	When set to 1, mutes ih_cec_stat0[6]
5 ERROR_FOLLOW	When set to 1, mutes ih_cec_stat0[5]
4 ERROR_INITIATOR	When set to 1, mutes ih_cec_stat0[4]
3 ARB_LOST	When set to 1, mutes ih_cec_stat0[3]
2 NACK	When set to 1, mutes ih_cec_stat0[2]
1 EOM	When set to 1, mutes ih_cec_stat0[1]
0 DONE	When set to 1, mutes ih_cec_stat0[0]

13.9.3.2.19 Video Packetizer Interrupt Mute Control Register (ih_mute_vp_stat0)

13.9.3.2.19.1 Offset

Register	Offset
ih_mute_vp_stat0	87h

13.9.3.2.19.2 Diagram



13.9.3.2.19.3 Fields

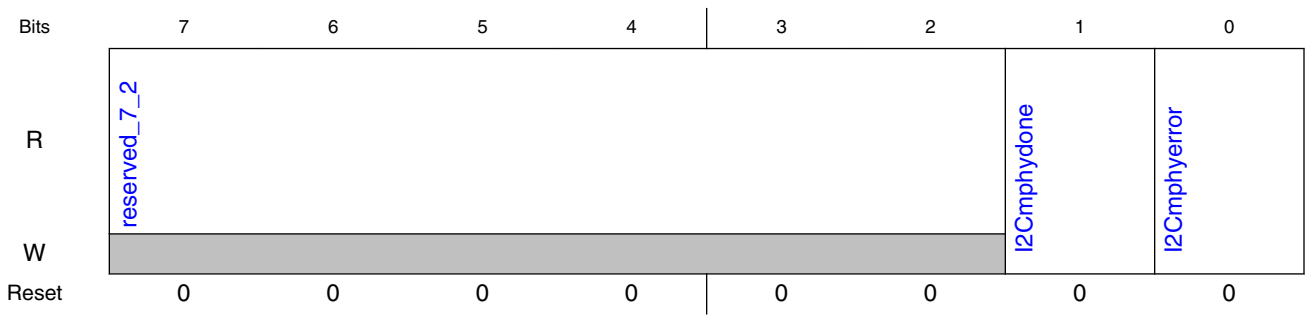
Field	Function
7 fifofullrepet	When set to 1, mutes ih_vp_stat0[7]
6 fifoemptyrepet	When set to 1, mutes ih_vp_stat0[6]
5 fifofullpp	When set to 1, mutes ih_vp_stat0[5]
4 fifoemptypp	When set to 1, mutes ih_vp_stat0[4]
3 fifofullremap	When set to 1, mutes ih_vp_stat0[3]
2 fifoemptyremap	When set to 1, mutes ih_vp_stat0[2]
1 spare_2	Reserved as "spare" bit with no associated functionality.
0 spare_1	Reserved as "spare" bit with no associated functionality.

13.9.3.2.20 PHY GEN2 I2C Master Interrupt Mute Control Register (ih_mute_i2cmphy_stat0)

13.9.3.2.20.1 Offset

Register	Offset
ih_mute_i2cmphy_stat0	88h

13.9.3.2.20.2 Diagram



13.9.3.2.20.3 Fields

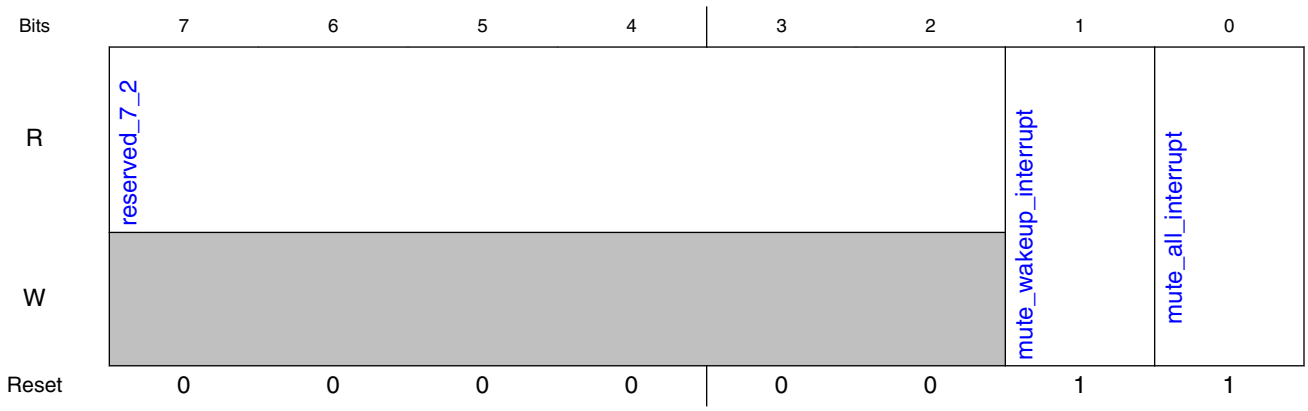
Field	Function
7-2 reserved_7_2	Reserved and read as zero
1 I2Cmphydone	When set to 1, mutes ih_i2cmphy_stat0[1]
0 I2Cmphyerror	When set to 1, mutes ih_i2cmphy_stat0[0]

13.9.3.2.21 Global Interrupt Mute Control Register (ih_mute)

13.9.3.2.21.1 Offset

Register	Offset
ih_mute	FFh

13.9.3.2.21.2 Diagram



13.9.3.2.21.3 Fields

Field	Function
7-2 reserved_7_2	Reserved and read as zero
1 mute_wakeup_interrupt	When set to 1, mutes the main interrupt output port. When set to 1, mutes the main interrupt output port. The sticky bit interrupts continue with their state accessible through the configuration bus, only the main interrupt line is muted.
0 mute_all_interrupt	When set to 1, mutes the main interrupt line (where all interrupts are ORed). When set to 1, mutes the main interrupt line (where all interrupts are ORed). The sticky bit interrupts continue with their state; only the main interrupt line is muted.

13.9.3.3 VideoSampler register descriptions

13.9.3.3.1 Memory map

VideoSampler registers base address: 200h

Offset	Register	Width (In bits)	Access	Reset value
0h	Video Input Mapping and Internal Data Enable Configuration Register (tx_invid0)	8	RW	01h
1h	Video Input Stuffing Enable Register (tx_instuffing)	8	RW	00h
2h	Video Input gy Data Channel Stuffing Register 0 (tx_gydata0)	8	RW	00h
3h	Video Input gy Data Channel Stuffing Register 1 (tx_gydata1)	8	RW	00h

Table continues on the next page...

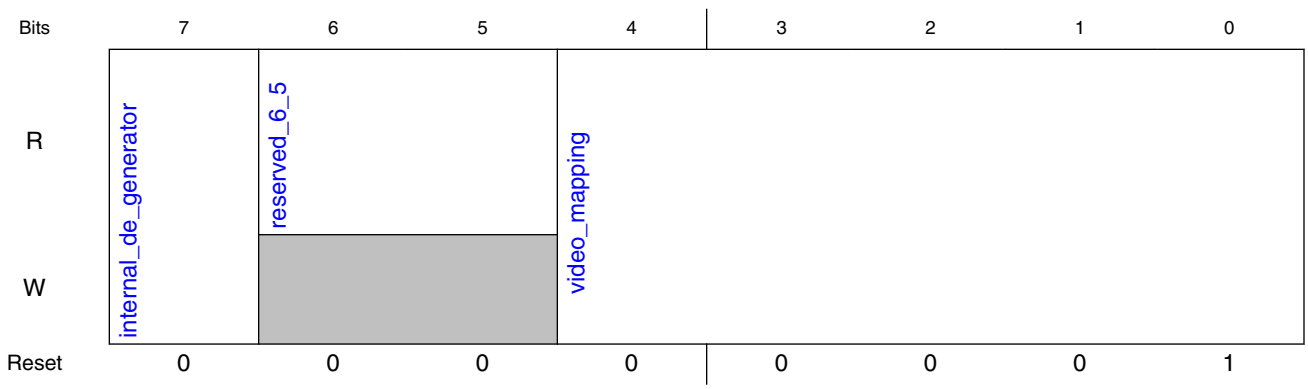
Offset	Register	Width (In bits)	Access	Reset value
4h	Video Input rcr Data Channel Stuffing Register 0 (tx_rcrdata0)	8	RW	00h
5h	Video Input rcr Data Channel Stuffing Register 1 (tx_rcrdata1)	8	RW	00h
6h	Video Input bcb Data Channel Stuffing Register 0 (tx_bcbdata0)	8	RW	00h
7h	Video Input bcb Data Channel Stuffing Register 1 (tx_bcbdata1)	8	RW	00h

13.9.3.3.2 Video Input Mapping and Internal Data Enable Configuration Register (tx_invid0)

13.9.3.3.2.1 Offset

Register	Offset
tx_invid0	0h

13.9.3.3.2.2 Diagram



13.9.3.3.2.3 Fields

Field	Function
7 internal_de_generator	Internal data enable (DE) generator enable. Internal data enable (DE) generator enable. If data enable is not available for the input video, set this bit to one to activate the internal data enable generator. Attention: This feature only works for input video modes that have native repetition (such as, all CEA videos). No desired pixel repetition can be used with this feature because these configurations only affect the Frame Composer and not this block. The DE Generator does not work for the following conditions: - Transmission of video with CEA VIC 39

Table continues on the next page...

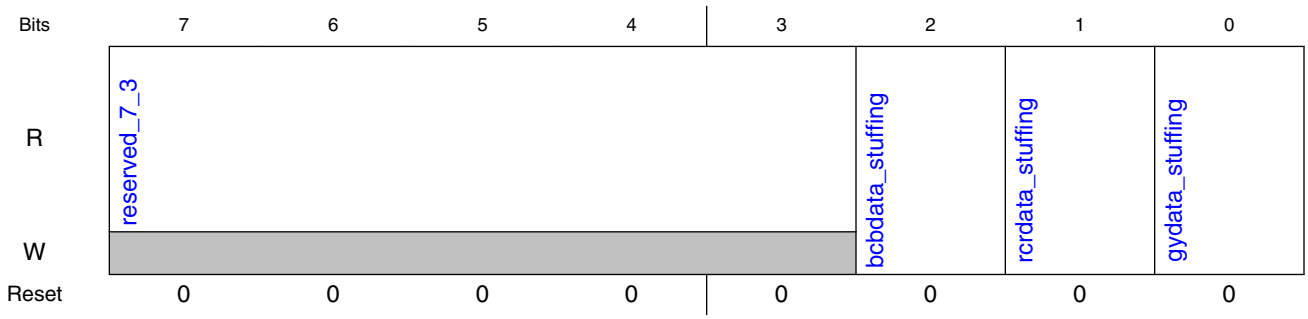
Field	Function
	- Transmission of 3D video using the field alternative structure
6-5 reserved_6_5	Reserved for future use
4-0 video_mapping	Video Input mapping (color space/color depth): 0x01: RGB 4:4:4/8 bits 0x03: RGB 4:4:4/10 bits 0x05: RGB 4:4:4/12 bits 0x07: RGB 4:4:4/16 bits 0x09: YCbCr 4:4:4 or 4:2:0/8 bits 0x0B: YCbCr 4:4:4 or 4:2:0/10 bits 0x0D: YCbCr 4:4:4 or 4:2:0/12 bits 0x0F: YCbCr 4:4:4 or 4:2:0/16 bits 0x16: YCbCr 4:2:2/8 bits 0x14: YCbCr 4:2:2/10 bits 0x12: YCbCr 4:2:2/12 bits 0x17: YCbCr 4:4:4 (IPI)/8 bits 0x18: YCbCr 4:4:4 (IPI)/10 bits 0x19: YCbCr 4:4:4 (IPI)/12 bits 0x1A: YCbCr 4:4:4 (IPI)/16 bits 0x1B: YCbCr 4:2:2 (IPI)/12 bits 0x1C: YCbCr 4:2:0 (IPI)/8 bits 0x1D: YCbCr 4:2:0 (IPI)/10 bits 0x1E: YCbCr 4:2:0 (IPI)/12 bits 0x1F: YCbCr 4:2:0 (IPI)/16 bits Note: IPI means Image Pixel Interface and it is a proprietary interface used on SNPS MIPI Controllers. Video Input mapping (color space/color depth): 0x01: RGB 4:4:4/8 bits 0x03: RGB 4:4:4/10 bits 0x05: RGB 4:4:4/12 bits 0x07: RGB 4:4:4/16 bits 0x09: YCbCr 4:4:4 or 4:2:0/8 bits 0x0B: YCbCr 4:4:4 or 4:2:0/10 bits 0x0D: YCbCr 4:4:4 or 4:2:0/12 bits 0x0F: YCbCr 4:4:4 or 4:2:0/16 bits 0x16: YCbCr 4:2:2/8 bits 0x14: YCbCr 4:2:2/10 bits 0x12: YCbCr 4:2:2/12 bits 0x17: YCbCr 4:4:4 (IPI)/8 bits 0x18: YCbCr 4:4:4 (IPI)/10 bits 0x19: YCbCr 4:4:4 (IPI)/12 bits 0x1A: YCbCr 4:4:4 (IPI)/16 bits 0x1B: YCbCr 4:2:2 (IPI)/12 bits 0x1C: YCbCr 4:2:0 (IPI)/8 bits 0x1D: YCbCr 4:2:0 (IPI)/10 bits 0x1E: YCbCr 4:2:0 (IPI)/12 bits 0x1F: YCbCr 4:2:0 (IPI)/16 bits Note: IPI means Image Pixel Interface and it is a proprietary interface used on SNPS MIPI Controllers. It allows easy video mapping between SNPS HDMI and MIPI Controllers

13.9.3.3.3 Video Input Stuffing Enable Register (tx_instuffing)

13.9.3.3.3.1 Offset

Register	Offset
tx_instuffing	1h

13.9.3.3.3.2 Diagram



13.9.3.3.3.3 Fields

Field	Function
7-3 reserved_7_3	Reserved and read as zero
2 bcbdata_stuffing	- 0b: When the dataen signal is low, the value in the bcbdata[15:0] output is the one sampled from the corresponding input data. - 0b: When the dataen signal is low, the value in the bcbdata[15:0] output is the one sampled from the corresponding input data. - 1b: When the dataen signal is low, the value in the bcbdata[15:0] output is given by the values in the TX_BCBDTA0 and TX_BCBDATA1 registers.
1 rcrdata_stuffing	- 0b: When the dataen signal is low, the value in the rcrdata[15:0] output is the one sampled from the corresponding input data. - 0b: When the dataen signal is low, the value in the rcrdata[15:0] output is the one sampled from the corresponding input data. - 1b: When the dataen signal is low, the value in the rcrdata[15:0] output is given by the values in TX_RCRDTA0 and TX_RCRDATA1 registers.
0 gydata_stuffing	- 0b: When the dataen signal is low, the value in the gydata[15:0] output is the one sampled from the corresponding input data. - 0b: When the dataen signal is low, the value in the gydata[15:0] output is the one sampled from the corresponding input data. - 1b: When the dataen signal is low, the value in the gydata[15:0] output is given by the values in TX_GYDTA0 and TX_GYDATA1 registers.

13.9.3.3.4 Video Input gy Data Channel Stuffing Register 0 (tx_gydata0)

13.9.3.3.4.1 Offset

Register	Offset
tx_gydata0	2h

13.9.3.3.4.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	gydata							
Reset	0	0	0	0	0	0	0	0

13.9.3.3.4.3 Fields

Field	Function
7-0 gydata	This register defines the value of gydata[7:0] when TX_INSTUFFING[0] (gydata_stuffing) is set to 1b.

13.9.3.3.5 Video Input gy Data Channel Stuffing Register 1 (tx_gydata1)

13.9.3.3.5.1 Offset

Register	Offset
tx_gydata1	3h

13.9.3.3.5.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	gydata							
Reset	0	0	0	0	0	0	0	0

13.9.3.3.5.3 Fields

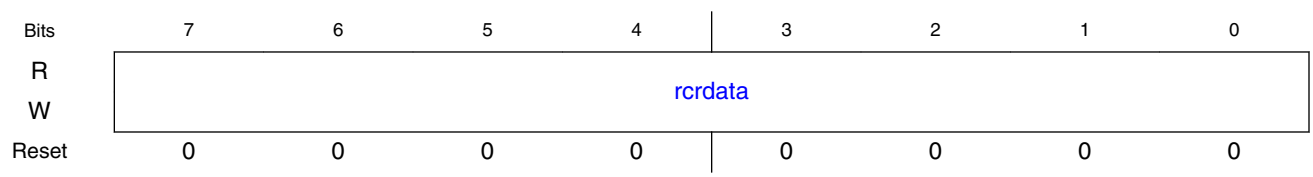
Field	Function
7-0 gydata	This register defines the value of gydata[15:8] when TX_INSTUFFING[0] (gydata_stuffing) is set to 1b.

13.9.3.3.6 Video Input rcr Data Channel Stuffing Register 0 (tx_rcrdata0)

13.9.3.3.6.1 Offset

Register	Offset
tx_rcrdata0	4h

13.9.3.3.6.2 Diagram



13.9.3.3.6.3 Fields

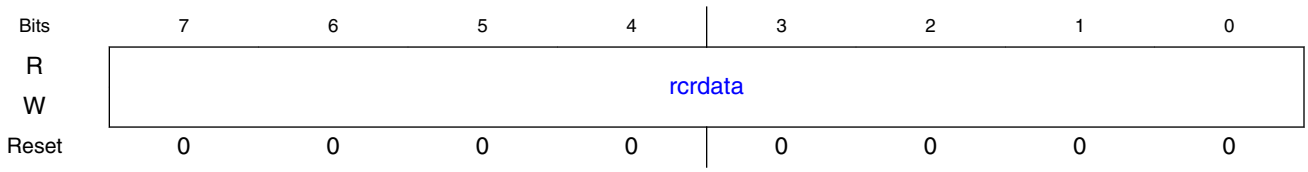
Field	Function
7-0 rcrdata	This register defines the value of rcrydata[7:0] when TX_INSTUFFING[1] (rcrdata_stuffing) is set to 1b.

13.9.3.3.7 Video Input rcr Data Channel Stuffing Register 1 (tx_rcrdata1)

13.9.3.3.7.1 Offset

Register	Offset
tx_rcrdata1	5h

13.9.3.3.7.2 Diagram



13.9.3.3.7.3 Fields

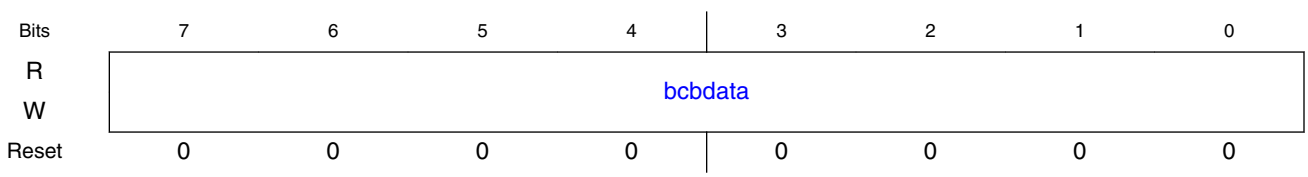
Field	Function
7-0 rcrdata	This register defines the value of rcrydata[15:8] when TX_INSTUFFING[1] (rcrdata_stuffing) is set to 1b.

13.9.3.3.8 Video Input bcb Data Channel Stuffing Register 0 (tx_bcbdata0)

13.9.3.3.8.1 Offset

Register	Offset
tx_bcbdata0	6h

13.9.3.3.8.2 Diagram



13.9.3.3.8.3 Fields

Field	Function
7-0 bcbdata	This register defines the value of bcbdata[7:0] when TX_INSTUFFING[2] (bcbdata_stuffing) is set to 1b.

13.9.3.3.9 Video Input bcb Data Channel Stuffing Register 1 (tx_bcbdata1)

13.9.3.3.9.1 Offset

Register	Offset
tx_bcbdata1	7h

13.9.3.3.9.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	bcbdata							
Reset	0	0	0	0	0	0	0	0

13.9.3.3.9.3 Fields

Field	Function
7-0 bcbdata	This register defines the value of bcbdata[15:8] when TX_INSTUFFING[2] (bcbdata_stuffing) is set to 1b.

13.9.3.4 VideoPacketizer register descriptions

13.9.3.4.1 Memory map

VideoPacketizer registers base address: 800h

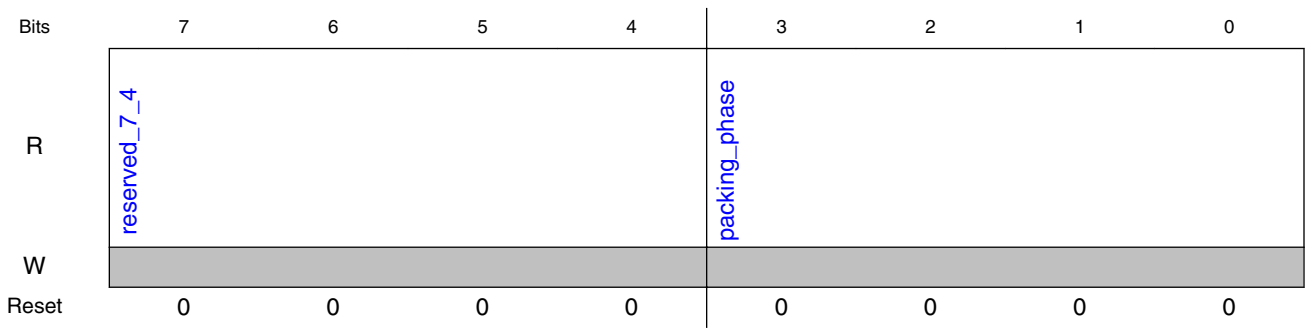
Offset	Register	Width (In bits)	Access	Reset value
0h	Video Packetizer Packing Phase Status Register (vp_status)	8	RO	00h
1h	Video Packetizer Pixel Repetition and Color Depth Register (vp_pr_cd)	8	RW	00h
2h	Video Packetizer Stuffing and Default Packing Phase Register (vp_stuff)	8	RW	00h
3h	Video Packetizer YCbCr 422 Remapping Register (vp_remap)	8	RW	00h
4h	Video Packetizer Output and Enable Configuration Register (vp_conf)	8	RW	24h
7h	Video Packetizer Interrupt Mask Register (vp_mask)	8	RW	00h

13.9.3.4.2 Video Packetizer Packing Phase Status Register (vp_status)

13.9.3.4.2.1 Offset

Register	Offset
vp_status	0h

13.9.3.4.2.2 Diagram



13.9.3.4.2.3 Fields

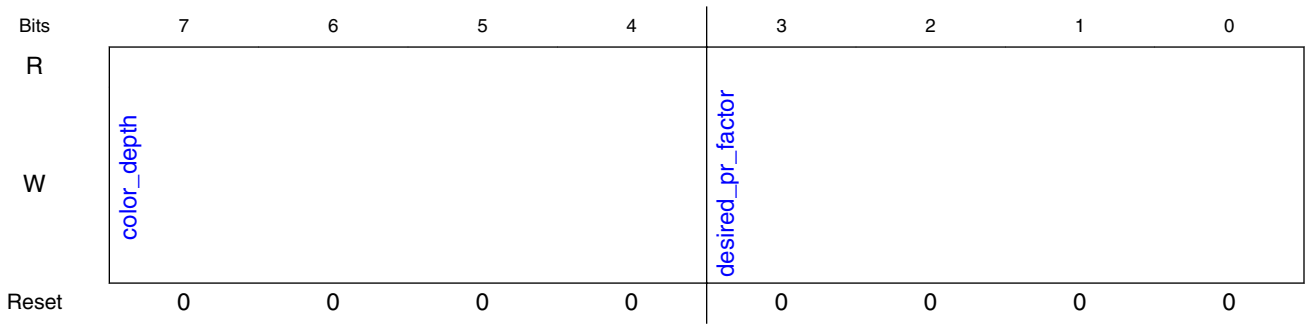
Field	Function
7-4 reserved_7_4	Reserved and read as zero
3-0 packing_phase	Read only register that holds the "packing phase" output of the Video Packetizer block. Read only register that holds the "packing phase" output of the Video Packetizer block. For more information about "packing" video data, refer to the HDMI 1.4b specification. The register is updated at TMDS clock rate.

13.9.3.4.3 Video Packetizer Pixel Repetition and Color Depth Register (vp_pr_cd)

13.9.3.4.3.1 Offset

Register	Offset
vp_pr_cd	1h

13.9.3.4.3.2 Diagram



13.9.3.4.3.3 Fields

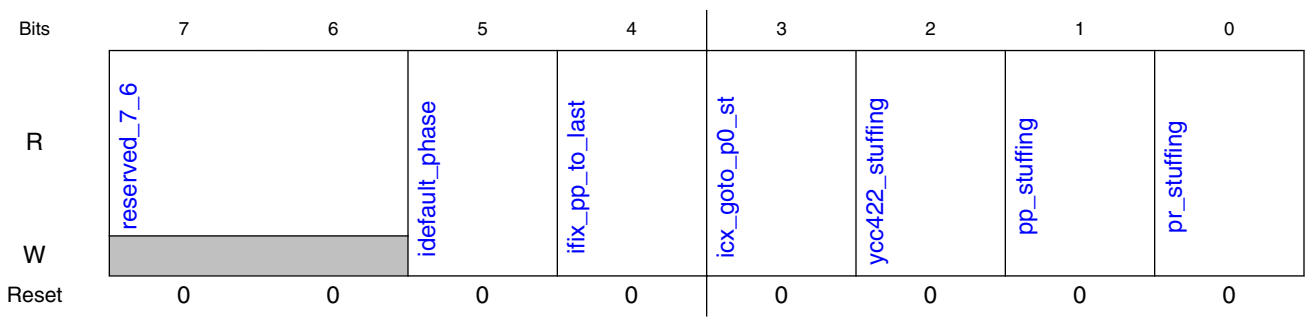
Field	Function
7-4 color_depth	The Color depth configuration is described as the following, with the action stated corresponding to color_depth[3:0]: - 0000b: 24 bits per pixel video (8 bits per component). The Color depth configuration is described as the following, with the action stated corresponding to color_depth[3:0]: - 0000b: 24 bits per pixel video (8 bits per component). 8-bit packing mode. - 0001b-0011b: Reserved. Not used. - 0100b: 24 bits per pixel video (8 bits per component). 8-bit packing mode. - 0101b: 30 bits per pixel video (10 bits per component). 10-bit packing mode. - 0110b: 36 bits per pixel video (12 bits per component). 12-bit packing mode. - 0111b: 48 bits per pixel video (16 bits per component). 16-bit packing mode. - Other: Reserved. Not used.
3-0 desired_pr_factor	Desired pixel repetition factor configuration. Desired pixel repetition factor configuration. The configured value sets H13T PHY PLL to multiply pixel clock by the factor in order to obtain the desired repetition clock. For the CEA modes some are already defined with pixel repetition in the input video. So for CEA modes this shall be always 0. Shall only be used if the user wants to do pixel repetition using H13TCTRL controller. The action is stated corresponding to desired_pr_factor[3:0]: - 0000b: No pixel repetition (pixel sent only once) - 0001b: Pixel sent two times (pixel repeated once) - 0010b: Pixel sent three times - 0011b: Pixel sent four times - 0100b: Pixel sent five times - 0101b: Pixel sent six times - 0110b: Pixel sent seven times - 0111b: Pixel sent eight times - 1000b: Pixel sent nine times - 1001b: Pixel sent 10 times - Other: Reserved. Not used

13.9.3.4.4 Video Packetizer Stuffing and Default Packing Phase Register (vp_stuff)

13.9.3.4.4.1 Offset

Register	Offset
vp_stuff	2h

13.9.3.4.4.2 Diagram



13.9.3.4.4.3 Fields

Field	Function
7-6 reserved_7_6	Reserved and read as zero
5 idefault_phase	Controls the default phase packing machine used according to HDMI 1. Controls the default phase packing machine used according to HDMI 1.4b specification: "If the transmitted video format has timing such that the phase of the first pixel of every Video Data Period corresponds to pixel packing phase 0 (e.g. 10P0, 12P0, 16P0), the Source may set the Default_Phase bit in the GCP. The Sink may use this bit to optimize its filtering or handling of the PP field." This means that videos that are in the below conditions: - 10bit mode, the Htotal dividable by 4, default_phase can be set 1 - 12bit mode, the Htotal dividable by 2, default_phase can be set 1
4 ifix_pp_to_last	Reserved Reserved. Controls packing machine strategy
3 icx_goto_p0_st	Reserved Reserved. Controls packing machine strategy
2 ycc422_stuffing	YCbCr 422 remap stuffing control.

Table continues on the next page...

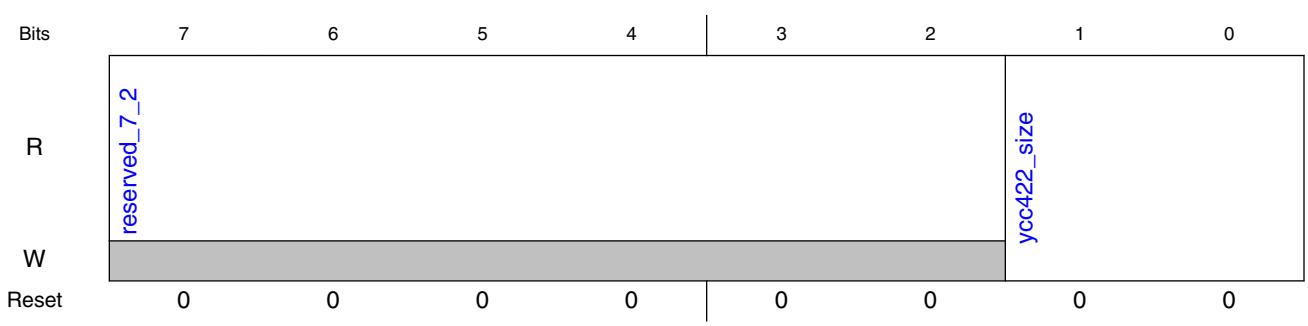
Field	Function
	YCbCr 422 remap stuffing control. For horizontal blanking, the action is stated corresponding to ycc422_stuffing: 0b: YCbCr 422 remap block in direct mode (input blanking data goes directly to output). 1b: YCbCr 422 remap block in stuffing mode. When "de" goes to low the outputs are fixed to 0x00.
1 pp_stuffing	Pixel packing stuffing control. Pixel packing stuffing control. The action is stated corresponding to pp_stuffing: 0b: Pixel packing block in direct mode (input blanking data goes directly to output). 1b: Pixel packing block in stuffing mode. When "de_rep" goes to low the outputs are fixed to 0x00.
0 pr_stuffing	Pixel repeater stuffing control. Pixel repeater stuffing control. The action is stated corresponding to pp_stuffing: 0b: Pixel repeater block in direct mode (input blanking data goes directly to output). 1b: Pixel repeater block in stuffing mode. When "de" goes to low the outputs are fixed to 0x00.

13.9.3.4.5 Video Packetizer YCbCr 422 Remapping Register (vp_remap)

13.9.3.4.5.1 Offset

Register	Offset
vp_remap	3h

13.9.3.4.5.2 Diagram



13.9.3.4.5.3 Fields

Field	Function
7-2 reserved_7_2	Reserved and read as zero

Table continues on the next page...

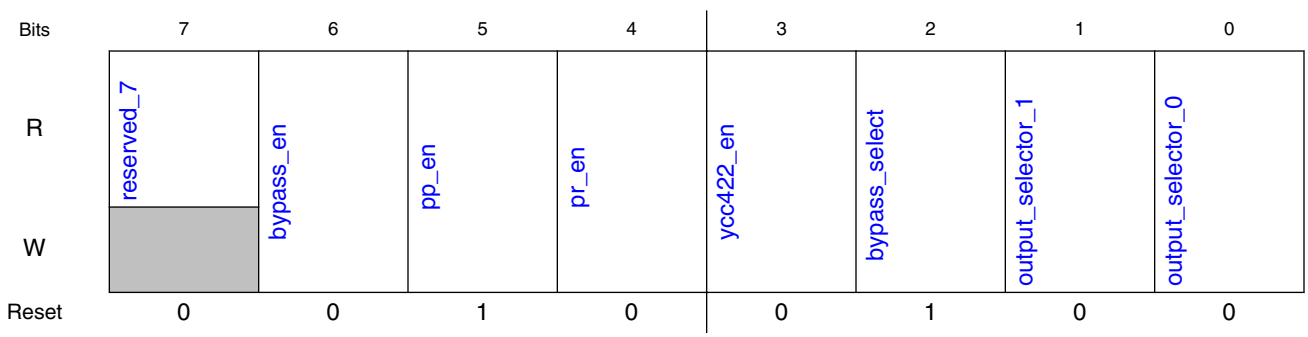
Field	Function
1-0 ycc422_size	YCbCr 422 remap input video size ycc422_size[1:0] 00b: YCbCr 422 16-bit input video (8 bits per component) 01b: YCbCr 422 20-bit input video (10 bits per component) 10b: YCbCr 422 24-bit input video (12 bits per component) 11b: Reserved YCbCr 422 remap input video size ycc422_size[1:0] 00b: YCbCr 422 16-bit input video (8 bits per component) 01b: YCbCr 422 20-bit input video (10 bits per component) 10b: YCbCr 422 24-bit input video (12 bits per component) 11b: Reserved. Not used

13.9.3.4.6 Video Packetizer Output and Enable Configuration Register (vp_conf)

13.9.3.4.6.1 Offset

Register	Offset
vp_conf	4h

13.9.3.4.6.2 Diagram



13.9.3.4.6.3 Fields

Field	Function
7 reserved_7	Reserved and read as zero
6 bypass_en	When set to 1'b1, Pixel packing enable. When set to 1'b1, Pixel packing enable. When set to 1b'0, the pixel packing block is controlled by pp_en.

Table continues on the next page...

Field	Function
	Note: the use of this field is deprecated
5 pp_en	Pixel packing enable. Pixel packing enable. When set to 0, the pixel packing block is disabled if bypass_en is 1'b0.
4 pr_en	Pixel repeater enable. Pixel repeater enable. When set to 0, the pixel repetition block is disabled.
3 ycc422_en	YCbCr 422 select enable. YCbCr 422 select enable. Disabling forces bypass module to output always zeros.
2 bypass_select	bypass_select 0b: Data from pixel repeater block 1b: Data from input of Video Packetizer block
1 output_selector_1	When set to 1'b1, Data from pixel packing block Note: the use of this field is deprecated
0 output_selector_0	Video Packetizer output selection 0b: Data from pixel packing block 1b: Data from YCbCr 422 remap block

13.9.3.4.7 Video Packetizer Interrupt Mask Register (vp_mask)

13.9.3.4.7.1 Offset

Register	Offset
vp_mask	7h

13.9.3.4.7.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	ointfullrepet	ointemptyrepet	ointfullpp	ointemptypp	ointfullremap	ointemptyremap	spare_2	spare_1
Reset	0	0	0	0	0	0	0	0

13.9.3.4.7.3 Fields

Field	Function
7 ointfullrepet	Mask bit for Video Packetizer pixel repeater FIFO full
6 ointemptyrepet	Mask bit for Video Packetizer pixel repeater FIFO empty
5 ointfullpp	Mask bit for Video Packetizer pixel packing FIFO full
4 ointemptypp	Mask bit for Video Packetizer pixel packing FIFO empty
3 ointfullremap	Mask bit for Video Packetizer pixel YCbCr 422 re-mapper FIFO full
2 ointemptyremap	Mask bit for Video Packetizer pixel YCbCr 422 re-mapper FIFO empty
1 spare_2	Reserved as "spare" bit with no associated functionality.
0 spare_1	Reserved as "spare" bit with no associated functionality.

13.9.3.5 FrameComposer register descriptions

13.9.3.5.1 Memory map

FrameComposer registers base address: 1000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Frame Composer Input Video Configuration Register (fc_invidconf)	8	RW	70h
1h	Frame Composer Input Video HActive Pixels Register 0 (fc_inhactiv0)	8	RW	00h
2h	Frame Composer Input Video HActive Pixels Register 1 (fc_inhactiv1)	8	RW	00h
3h	Frame Composer Input Video HBlank Pixels Register 0 (fc_inhblank0)	8	RW	00h
4h	Frame Composer Input Video HBlank Pixels Register 1 (fc_inhblank1)	8	RW	00h
5h	Frame Composer Input Video VActive Pixels Register 0 (fc_invactiv0)	8	RW	00h
6h	Frame Composer Input Video VActive Pixels Register 1 (fc_invactiv1)	8	RW	00h
7h	Frame Composer Input Video VBlank Pixels Register (fc_invblank)	8	RW	00h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8h	Frame Composer Input Video HSync Front Porch Register 0 (fc_hsyncindelay0)	8	RW	00h
9h	Frame Composer Input Video HSync Front Porch Register 1 (fc_hsyncindelay1)	8	RW	00h
Ah	Frame Composer Input Video HSync Width Register 0 (fc_hsyncinwidth0)	8	RW	00h
Bh	Frame Composer Input Video HSync Width Register 1 (fc_hsyncinwidth1)	8	RW	00h
Ch	Frame Composer Input Video VSync Front Porch Register (fc_vsyncindelay)	8	RW	00h
Dh	Frame Composer Input Video VSync Width Register (fc_vsyncinwidth)	8	RW	00h
Eh	Frame Composer Input Video Refresh Rate Register 0 (fc_infreq0)	8	RW	00h
Fh	Frame Composer Input Video Refresh Rate Register 1 (fc_infreq1)	8	RW	00h
10h	Frame Composer Input Video Refresh Rate Register 2 (fc_infreq2)	8	RW	00h
11h	Frame Composer Control Period Duration Register (fc_ctrldur)	8	RW	00h
12h	Frame Composer Extended Control Period Duration Register (fc_exctrldur)	8	RW	00h
13h	Frame Composer Extended Control Period Maximum Spacing Register (fc_exctrlspac)	8	RW	00h
14h	Frame Composer Channel 0 Non-Preamble Data Register (fc_ch0pream)	8	RW	00h
15h	Frame Composer Channel 1 Non-Preamble Data Register (fc_ch1pream)	8	RW	00h
16h	Frame Composer Channel 2 Non-Preamble Data Register (fc_ch2pream)	8	RW	00h
17h	Frame Composer AVI Packet Configuration Register 3 (fc_aviconf3)	8	RW	00h
18h	Frame Composer GCP Packet Configuration Register (fc_gcp)	8	RW	00h
19h	Frame Composer AVI Packet Configuration Register 0 (fc_aviconf0)	8	RW	00h
1Ah	Frame Composer AVI Packet Configuration Register 1 (fc_aviconf1)	8	RW	00h
1Bh	Frame Composer AVI Packet Configuration Register 2 (fc_aviconf2)	8	RW	00h
1Ch	Frame Composer AVI Packet VIC Register (fc_avivid)	8	RW	00h
25h	Frame Composer AUD Packet Configuration Register 0 (fc_audiconf0)	8	RW	00h
26h	Frame Composer AUD Packet Configuration Register 1 (fc_audiconf1)	8	RW	00h
27h	Frame Composer AUD Packet Configuration Register 2 (fc_audiconf2)	8	RW	00h
28h	Frame Composer AUD Packet Configuration Register 3 (fc_audiconf3)	8	RW	00h
29h	Frame Composer VSI Packet Data IEEE Register 0 (fc_vsdieeeid0)	8	RW	00h
2Ah	Frame Composer VSI Packet Data Size Register (fc_vsdsize)	8	RW	1Bh
30h	Frame Composer VSI Packet Data IEEE Register 1 (fc_vsdieeeid1)	8	RW	00h
31h	Frame Composer VSI Packet Data IEEE Register 2 (fc_vsdieeeid2)	8	RW	00h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
62h	Frame Composer SPD Packet Data Source Product Descriptor Register (fc_spddeviceinf)	8	RW	00h
63h	Frame Composer Audio Sample Flat and Layout Configuration Register (fc_audsconf)	8	RW	00h
64h	Frame Composer Audio Sample Flat and Layout Status Register (fc_audsstat)	8	RO	00h
65h	Frame Composer Audio Sample Validity Flag Register (fc_audsv)	8	RW	00h
66h	Frame Composer Audio Sample User Flag Register (fc_audsu)	8	RW	00h
67h	Frame Composer Audio Sample Channel Status Configuration Register 0 (fc_audschnl0)	8	RW	00h
68h	Frame Composer Audio Sample Channel Status Configuration Register 1 (fc_audschnl1)	8	RW	00h
69h	Frame Composer Audio Sample Channel Status Configuration Register 2 (fc_audschnl2)	8	RW	00h
6Ah	Frame Composer Audio Sample Channel Status Configuration Register 3 (fc_audschnl3)	8	RW	00h
6Bh	Frame Composer Audio Sample Channel Status Configuration Register 4 (fc_audschnl4)	8	RW	00h
6Ch	Frame Composer Audio Sample Channel Status Configuration Register 5 (fc_audschnl5)	8	RW	00h
6Dh	Frame Composer Audio Sample Channel Status Configuration Register 6 (fc_audschnl6)	8	RW	00h
6Eh	Frame Composer Audio Sample Channel Status Configuration Register 7 (fc_audschnl7)	8	RW	00h
6Fh	Frame Composer Audio Sample Channel Status Configuration Register 8 (fc_audschnl8)	8	RW	00h
73h	Frame Composer Number of High Priority Packets Attended Configuration Register (fc_ctrlqhigh)	8	RW	0Fh
74h	Frame Composer Number of Low Priority Packets Attended Configuration Register (fc_ctrlqlow)	8	RW	03h
75h	Frame Composer ACP Packet Type Configuration Register 0 (fc_acp0)	8	RW	00h
82h	Frame Composer ACP Packet Body Configuration Register 16 (fc_acp16)	8	RW	00h
83h	Frame Composer ACP Packet Body Configuration Register 15 (fc_acp15)	8	RW	00h
84h	Frame Composer ACP Packet Body Configuration Register 14 (fc_acp14)	8	RW	00h
85h	Frame Composer ACP Packet Body Configuration Register 13 (fc_acp13)	8	RW	00h
86h	Frame Composer ACP Packet Body Configuration Register 12 (fc_acp12)	8	RW	00h
87h	Frame Composer ACP Packet Body Configuration Register 11 (fc_acp11)	8	RW	00h
88h	Frame Composer ACP Packet Body Configuration Register 10 (fc_acp10)	8	RW	00h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
89h	Frame Composer ACP Packet Body Configuration Register 9 (fc_acp9)	8	RW	00h
8Ah	Frame Composer ACP Packet Body Configuration Register 8 (fc_acp8)	8	RW	00h
8Bh	Frame Composer ACP Packet Body Configuration Register 7 (fc_acp7)	8	RW	00h
8Ch	Frame Composer ACP Packet Body Configuration Register 6 (fc_acp6)	8	RW	00h
8Dh	Frame Composer ACP Packet Body Configuration Register 5 (fc_acp5)	8	RW	00h
8Eh	Frame Composer ACP Packet Body Configuration Register 4 (fc_acp4)	8	RW	00h
8Fh	Frame Composer ACP Packet Body Configuration Register 3 (fc_acp3)	8	RW	00h
90h	Frame Composer ACP Packet Body Configuration Register 2 (fc_acp2)	8	RW	00h
91h	Frame Composer ACP Packet Body Configuration Register 1 (fc_acp1)	8	RW	00h
92h	Frame Composer ISRC1 Packet Status, Valid, and Continue Configuration Register (fc_iscr1_0)	8	RW	00h
93h	Frame Composer ISRC1 Packet Body Register 16 (fc_iscr1_16)	8	RW	00h
94h	Frame Composer ISRC1 Packet Body Register 15 (fc_iscr1_15)	8	RW	00h
95h	Frame Composer ISRC1 Packet Body Register 14 (fc_iscr1_14)	8	RW	00h
96h	Frame Composer ISRC1 Packet Body Register 13 (fc_iscr1_13)	8	RW	00h
97h	Frame Composer ISRC1 Packet Body Register 12 (fc_iscr1_12)	8	RW	00h
98h	Frame Composer ISRC1 Packet Body Register 11 (fc_iscr1_11)	8	RW	00h
99h	Frame Composer ISRC1 Packet Body Register 10 (fc_iscr1_10)	8	RW	00h
9Ah	Frame Composer ISRC1 Packet Body Register 9 (fc_iscr1_9)	8	RW	00h
9Bh	Frame Composer ISRC1 Packet Body Register 8 (fc_iscr1_8)	8	RW	00h
9Ch	Frame Composer ISRC1 Packet Body Register 7 (fc_iscr1_7)	8	RW	00h
9Dh	Frame Composer ISRC1 Packet Body Register 6 (fc_iscr1_6)	8	RW	00h
9Eh	Frame Composer ISRC1 Packet Body Register 5 (fc_iscr1_5)	8	RW	00h
9Fh	Frame Composer ISRC1 Packet Body Register 4 (fc_iscr1_4)	8	RW	00h
A0h	Frame Composer ISRC1 Packet Body Register 3 (fc_iscr1_3)	8	RW	00h
A1h	Frame Composer ISRC1 Packet Body Register 2 (fc_iscr1_2)	8	RW	00h
A2h	Frame Composer ISRC1 Packet Body Register 1 (fc_iscr1_1)	8	RW	00h
A3h	Frame Composer ISRC2 Packet Body Register 15 (fc_iscr2_15)	8	RW	00h
A4h	Frame Composer ISRC2 Packet Body Register 14 (fc_iscr2_14)	8	RW	00h
A5h	Frame Composer ISRC2 Packet Body Register 13 (fc_iscr2_13)	8	RW	00h
A6h	Frame Composer ISRC2 Packet Body Register 12 (fc_iscr2_12)	8	RW	00h
A7h	Frame Composer ISRC2 Packet Body Register 11 (fc_iscr2_11)	8	RW	00h
A8h	Frame Composer ISRC2 Packet Body Register 10 (fc_iscr2_10)	8	RW	00h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
A9h	Frame Composer ISRC2 Packet Body Register 9 (fc_iscr2_9)	8	RW	00h
AAh	Frame Composer ISRC2 Packet Body Register 8 (fc_iscr2_8)	8	RW	00h
ABh	Frame Composer ISRC2 Packet Body Register 7 (fc_iscr2_7)	8	RW	00h
ACh	Frame Composer ISRC2 Packet Body Register 6 (fc_iscr2_6)	8	RW	00h
ADh	Frame Composer ISRC2 Packet Body Register 5 (fc_iscr2_5)	8	RW	00h
AEh	Frame Composer ISRC2 Packet Body Register 4 (fc_iscr2_4)	8	RW	00h
AFh	Frame Composer ISRC2 Packet Body Register 3 (fc_iscr2_3)	8	RW	00h
B0h	Frame Composer ISRC2 Packet Body Register 2 (fc_iscr2_2)	8	RW	00h
B1h	Frame Composer ISRC2 Packet Body Register 1 (fc_iscr2_1)	8	RW	00h
B2h	Frame Composer ISRC2 Packet Body Register 0 (fc_iscr2_0)	8	RW	00h
B3h	Frame Composer Data Island Auto Packet Scheduling Register 0 Configures the Frame Composer RDRB(1)/Manual(0) data island packet insertion for SPD, VSD, ISRC2, ISRC1 and ACP packets. (fc_datauto0)	8	RW	00h
B4h	Frame Composer Data Island Auto Packet Scheduling Register 1 Configures the Frame Composer (FC) RDRB frame interpolation for SPD, VSD, ISRC2, ISRC1 and ACP packet insertion on data island when FC is on RDRB mode for the listed packets. (fc_datauto1)	8	RW	00h
B5h	Frame Composer Data Island Auto packet scheduling Register 2 Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for SPD, VSD, ISRC2, ISRC1 and ACP packet insertion on data island when FC is on RDRB mode for the listed packets. (fc_datauto2)	8	RW	00h
B6h	Frame Composer Data Island Manual Packet Request Register Requests to the Frame Composer the data island packet insertion for NULL, SPD, VSD, ISRC2, ISRC1 and ACP packets when FC_DATAUTO0 bit is in manual mode for the packet requested. (fc_datman)	8	WO	00h
B7h	Frame Composer Data Island Auto Packet Scheduling Register 3 Configures the Frame Composer Automatic(1)/RDRB(0) data island packet insertion for AVI, GCP, AUDI and ACR packets. (fc_datauto3)	8	RW	7Fh
B8h	Frame Composer Round Robin ACR Packet Insertion Register 0 Configures the Frame Composer (FC) RDRB frame interpolation for ACR packet insertion on data island when FC is on RDRB mode for this packet. (fc_rdrb0)	8	RW	00h
B9h	Frame Composer Round Robin ACR Packet Insertion Register 1 Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the ACR packet insertion on data island when FC is on RDRB mode this packet. (fc_rdrb1)	8	RW	00h
BAh	Frame Composer Round Robin AUDI Packet Insertion Register 2 Configures the Frame Composer (FC) RDRB frame interpolation for AUDI packet insertion on data island when FC is on RDRB mode for this packet. (fc_rdrb2)	8	RW	00h
BBh	Frame Composer Round Robin AUDI Packet Insertion Register 3 Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the AUDI packet insertion on data island when FC is on RDRB mode this packet. (fc_rdrb3)	8	RW	00h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
BCh	Frame Composer Round Robin GCP Packet Insertion Register 4 Configures the Frame Composer (FC) RDRB frame interpolation for GCP packet insertion on data island when FC is on RDRB mode for this packet. (fc_rdrb4)	8	RW	00h
BDh	Frame Composer Round Robin GCP Packet Insertion Register 5 Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the GCP packet insertion on data island when FC is on RDRB mode this packet. (fc_rdrb5)	8	RW	00h
BEh	Frame Composer Round Robin AVI Packet Insertion Register 6 Configures the Frame Composer (FC) RDRB frame interpolation for AVI packet insertion on data island when FC is on RDRB mode for this packet. (fc_rdrb6)	8	RW	00h
BFh	Frame Composer Round Robin AVI Packet Insertion Register 7 Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the AVI packet insertion on data island when FC is on RDRB mode this packet. (fc_rdrb7)	8	RW	00h
C0h	Frame Composer Round Robin AMP Packet Insertion Register 8 (fc_rdrb8)	8	RW	00h
C1h	Frame Composer Round Robin AMP Packet Insertion Register 9 (fc_rdrb9)	8	RW	00h
C2h	Frame Composer Round Robin NTSC VBI Packet Insertion Register 10 (fc_rdrb10)	8	RW	00h
C3h	Frame Composer Round Robin NTSC VBI Packet Insertion Register 11 (fc_rdrb11)	8	RW	00h
C4h	Frame Composer Round Robin DRM Packet Insertion Register 12 (fc_rdrb12)	8	RW	00h
C5h	Frame Composer Round Robin DRM Packet Insertion Register 13 (fc_rdrb13)	8	RW	00h
D2h	Frame Composer Packet Interrupt Mask Register 0 (fc_mask0)	8	RW	3Dh
D6h	Frame Composer Packet Interrupt Mask Register 1 (fc_mask1)	8	RW	04h
DAh	Frame Composer High/Low Priority Overflow and DRM Interrupt Mask Register 2 (fc_mask2)	8	RW	10h
E0h	Frame Composer Pixel Repetition Configuration Register (fc_prconf)	8	RW	10h
E1h	Frame Composer Scrambler Control (fc_scrambler_ctrl)	8	RW	00h
E2h	Frame Composer Multi-Stream Audio Control (fc_multistream_ctrl)	8	RW	00h
E3h	Frame Composer Packet Transmission Control (fc_packet_tx_en)	8	RW	1Fh
E8h	Frame Composer Active Space Control (fc_actspc_hdlr_cfg)	8	RW	00h
E9h	Frame Composer Input Video 2D VActive Pixels Register 0 (fc_invact_2d_0)	8	RW	00h
EAh	Frame Composer Input Video VActive pixels Register 1 (fc_invact_2d_1)	8	RW	00h
100h	Frame Composer GMD Packet Status Register Gamut metadata packet status bit information for no_current_gmd, next_gmd_field, gmd_packet_sequence and current_gamut_seq_num. (fc_gmd_stat)	8	RO	00h
101h	Frame Composer GMD Packet Enable Register This register enables Gamut metadata (GMD) packet transmission. (fc_gmd_en)	8	RW	00h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
102h	Frame Composer GMD Packet Update Register This register performs an GMD packet content update according to the configured packet body (FC_GMD_PB0 to FC_GMD_PB27) and packet header (FC_GMD_HB). (fc_gmd_up)	8	WO	00h
103h	Frame Composer GMD Packet Schedule Configuration Register This register configures the number of GMD packets to be inserted per frame (starting always in the line where the active Vsync appears) and the line spacing between the transmitted GMD packets. (fc_g md_conf)	8	RW	10h
104h	Frame Composer GMD Packet Profile and Gamut Sequence Configuration Register This register configures the GMD packet header affected_gamut_seq_num and gmd_profile bits. (fc_gmd_hb)	8	RW	00h
128h	Frame Composer AMP Packet Header Register 1 (fc_amp_hb1)	8	RW	00h
129h	Frame Composer AMP Packet Header Register 2 (fc_amp_hb2)	8	RW	00h
148h	Frame Composer NTSC VBI Packet Header Register 1 (fc_nvbi_hb1)	8	RW	00h
149h	Frame Composer NTSC VBI Packet Header Register 2 (fc_nvbi_hb2)	8	RW	00h
167h	Frame Composer DRM Packet Update Register This register performs an DRM packet content update according to the configured packet body (FC_DRM_PB0 to FC_DRM_PB27) and packet header (FC_DRM_HB). (fc_drm_up)	8	WO	00h
200h	Frame Composer video/audio Force Enable Register This register allows to force the controller to output audio and video data the values configured in the FC_DBGAUD and FC_DBGTMDS registers. (fc_dbgforce)	8	RW	00h
201h	Frame Composer Audio Data Channel 0 Register 0 Configures the audio fixed data to be used in channel 0 when in fixed audio selection. (fc_dbgaud0ch0)	8	RW	00h
202h	Frame Composer Audio Data Channel 0 Register 1 Configures the audio fixed data to be used in channel 0 when in fixed audio selection. (fc_dbgaud1ch0)	8	RW	00h
203h	Frame Composer Audio Data Channel 0 Register 2 Configures the audio fixed data to be used in channel 0 when in fixed audio selection. (fc_dbgaud2ch0)	8	RW	00h
204h	Frame Composer Audio Data Channel 1 Register 0 Configures the audio fixed data to be used in channel 1 when in fixed audio selection. (fc_dbgaud0ch1)	8	RW	00h
205h	Frame Composer Audio Data Channel 1 Register 1 Configures the audio fixed data to be used in channel 1 when in fixed audio selection. (fc_dbgaud1ch1)	8	RW	00h
206h	Frame Composer Audio Data Channel 1 Register 2 Configures the audio fixed data to be used in channel 1 when in fixed audio selection. (fc_dbgaud2ch1)	8	RW	00h
207h	Frame Composer Audio Data Channel 2 Register 0 Configures the audio fixed data to be used in channel 2 when in fixed audio selection. (fc_dbgaud0ch2)	8	RW	00h
208h	Frame Composer Audio Data Channel 2 Register 1 Configures the audio fixed data to be used in channel 2 when in fixed audio selection. (fc_dbgaud1ch2)	8	RW	00h

Table continues on the next page...

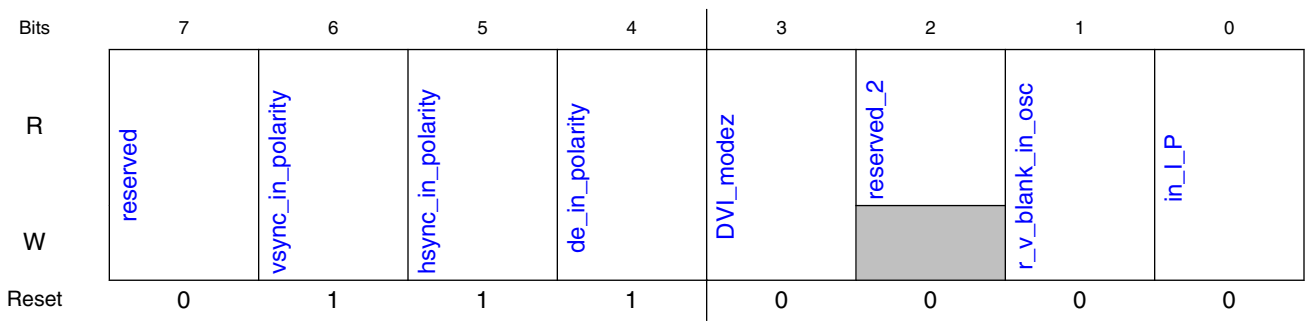
Offset	Register	Width (In bits)	Access	Reset value
209h	Frame Composer Audio Data Channel 2 Register 2 Configures the audio fixed data to be used in channel 2 when in fixed audio selection. (fc_dbgaud2ch2)	8	RW	00h
20Ah	Frame Composer Audio Data Channel 3 Register 0 Configures the audio fixed data to be used in channel 3 when in fixed audio selection. (fc_dbgaud0ch3)	8	RW	00h
20Bh	Frame Composer Audio Data Channel 3 Register 1 Configures the audio fixed data to be used in channel 3 when in fixed audio selection. (fc_dbgaud1ch3)	8	RW	00h
20Ch	Frame Composer Audio Data Channel 3 Register 2 Configures the audio fixed data to be used in channel 3 when in fixed audio selection. (fc_dbgaud2ch3)	8	RW	00h
20Dh	Frame Composer Audio Data Channel 4 Register 0 Configures the audio fixed data to be used in channel 4 when in fixed audio selection. (fc_dbgaud0ch4)	8	RW	00h
20Eh	Frame Composer Audio Data Channel 4 Register 1 Configures the audio fixed data to be used in channel 4 when in fixed audio selection. (fc_dbgaud1ch4)	8	RW	00h
20Fh	Frame Composer Audio Data Channel 4 Register 2 Configures the audio fixed data to be used in channel 4 when in fixed audio selection. (fc_dbgaud2ch4)	8	RW	00h
210h	Frame Composer Audio Data Channel 5 Register 0 Configures the audio fixed data to be used in channel 5 when in fixed audio selection. (fc_dbgaud0ch5)	8	RW	00h
211h	Frame Composer Audio Data Channel 5 Register 1 Configures the audio fixed data to be used in channel 5 when in fixed audio selection. (fc_dbgaud1ch5)	8	RW	00h
212h	Frame Composer Audio Data Channel 5 Register 2 Configures the audio fixed data to be used in channel 5 when in fixed audio selection. (fc_dbgaud2ch5)	8	RW	00h
213h	Frame Composer Audio Data Channel 6 Register 0 Configures the audio fixed data to be used in channel 6 when in fixed audio selection. (fc_dbgaud0ch6)	8	RW	00h
214h	Frame Composer Audio Data Channel 6 Register 1 Configures the audio fixed data to be used in channel 6 when in fixed audio selection. (fc_dbgaud1ch6)	8	RW	00h
215h	Frame Composer Audio Data Channel 6 Register 2 Configures the audio fixed data to be used in channel 6 when in fixed audio selection. (fc_dbgaud2ch6)	8	RW	00h
216h	Frame Composer Audio Data Channel 7 Register 0 Configures the audio fixed data to be used in channel 7 when in fixed audio selection. (fc_dbgaud0ch7)	8	RW	00h
217h	Frame Composer Audio Data Channel 7 Register 1 Configures the audio fixed data to be used in channel 7 when in fixed audio selection. (fc_dbgaud1ch7)	8	RW	00h
218h	Frame Composer Audio Data Channel 7 Register 2 Configures the audio fixed data to be used in channel 7 when in fixed audio selection. (fc_dbgaud2ch7)	8	RW	00h

13.9.3.5.2 Frame Composer Input Video Configuration Register (fc_invidconf)

13.9.3.5.2.1 Offset

Register	Offset
fc_invidconf	0h

13.9.3.5.2.2 Diagram



13.9.3.5.2.3 Fields

Field	Function
7 reserved	Reserved and read as zero
6 vsync_in_polarity	Vsync input polarity 1b: Active high 0b: Active low
5 hsync_in_polarity	Hsync input polarity 1b: Active high 0b: Active low
4 de_in_polarity	Data enable input polarity 1b: Active high 0b: Active low
3 DVI_modez	Active low 0b: DVI mode selected 1b: HDMI mode selected
2 reserved_2	Reserved and read as zero
1 r_v_blank_in_osc	Used for CEA861-D modes with fractional Vblank (for example, modes 5, 6, 7, 10, 11, 20, 21, and 22). Used for CEA861-D modes with fractional Vblank (for example, modes 5, 6, 7, 10, 11, 20, 21, and 22). For more modes, see the CEA861-D specification.

Table continues on the next page...

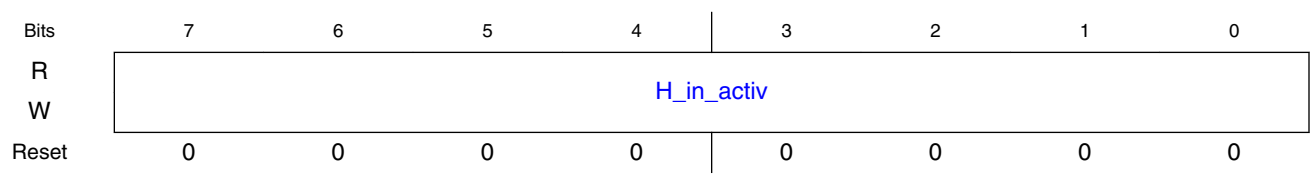
Field	Function
	Note: Set this field to 1 for video mode 39, although there is no Vblank oscillation. 1b: Active high
0 in_I_P	Input video mode: 1b: Interlaced 0b: Progressive

13.9.3.5.3 Frame Composer Input Video HActive Pixels Register 0 (fc_inhactiv0)

13.9.3.5.3.1 Offset

Register	Offset
fc_inhactiv0	1h

13.9.3.5.3.2 Diagram



13.9.3.5.3.3 Fields

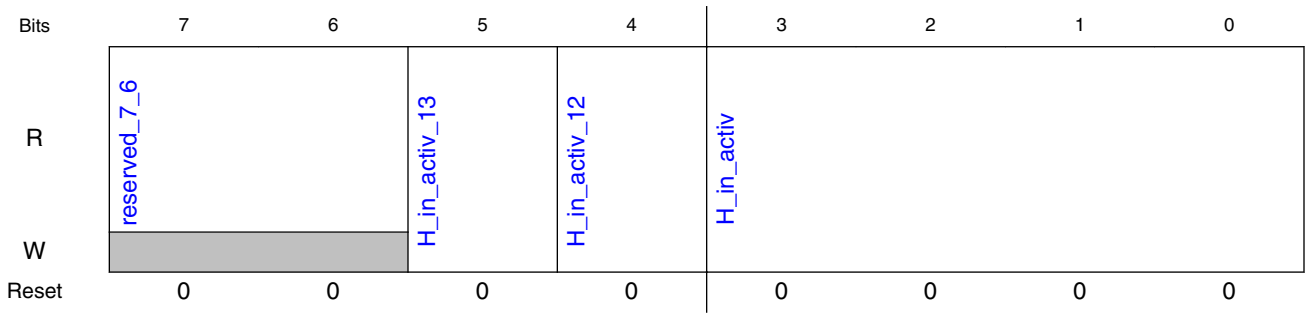
Field	Function
7-0 H_in_activ	Input video Horizontal active pixel region width. Input video Horizontal active pixel region width. Number of Horizontal active pixels [0...8191].

13.9.3.5.4 Frame Composer Input Video HActive Pixels Register 1 (fc_inhactiv1)

13.9.3.5.4.1 Offset

Register	Offset
fc_inhactiv1	2h

13.9.3.5.4.2 Diagram



13.9.3.5.4.3 Fields

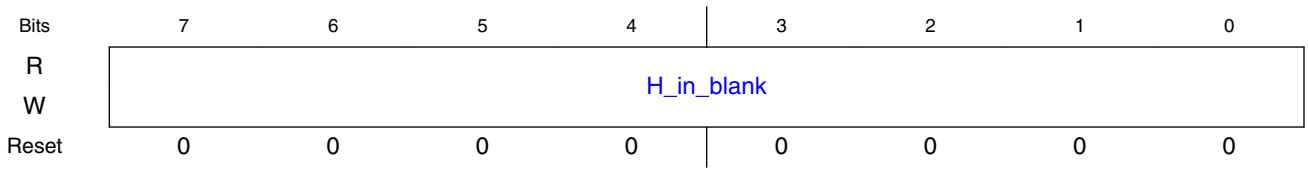
Field	Function
7-6 reserved_7_6	Reserved and read as zero
5 H_in_activ_13	Input video Horizontal active pixel region width (0 . Input video Horizontal active pixel region width (0 .. 16383) If the configuration parameter DWC_HDMI_TX_20 = True (1), this bit field holds bit 13.
4 H_in_activ_12	Input video Horizontal active pixel region width (0 . Input video Horizontal active pixel region width (0 . . 8191) If configuration parameter DWC_HDMI_TX_14 = True (1), this bit field holds bit 12.
3-0 H_in_activ	Input video Horizontal active pixel region width

13.9.3.5.5 Frame Composer Input Video HBlank Pixels Register 0 (fc_inhblank0)

13.9.3.5.5.1 Offset

Register	Offset
fc_inhblank0	3h

13.9.3.5.5.2 Diagram



13.9.3.5.5.3 Fields

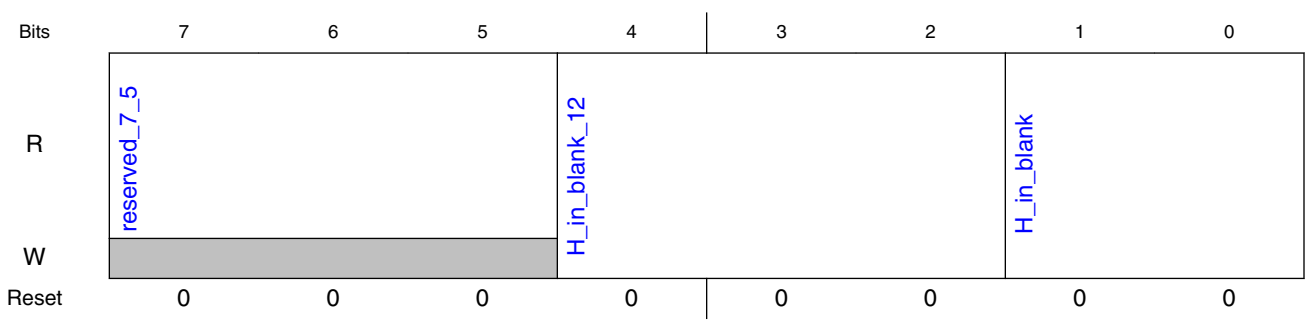
Field	Function
7-0 H_in_blank	Input video Horizontal blanking pixel region width. Input video Horizontal blanking pixel region width. Number of Horizontal blanking pixels [0...4095].

13.9.3.5.6 Frame Composer Input Video HBlank Pixels Register 1 (fc_inhblank1)

13.9.3.5.6.1 Offset

Register	Offset
fc_inhblank1	4h

13.9.3.5.6.2 Diagram



13.9.3.5.6.3 Fields

Field	Function
7-5 reserved_7_5	Reserved and read as zero

Table continues on the next page...

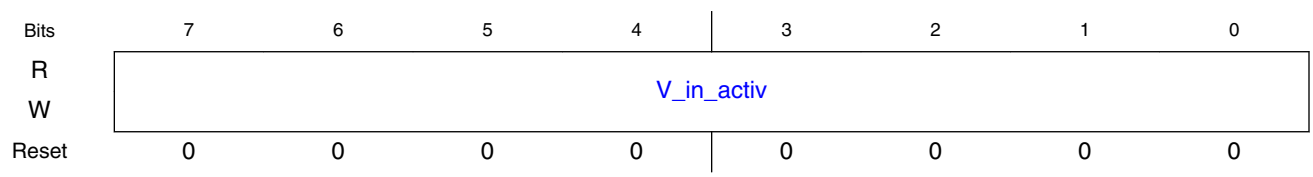
Field	Function
4-2 H_in_blank_12	Input video Horizontal blanking pixel region width If configuration parameter DWC_HDMI_TX_14 = True (1), this bit field holds bit 12:10 of number of horizontal blanking pixels.
1-0 H_in_blank	Input video Horizontal blanking pixel region width this bit field holds bits 9:8 of number of Horizontal blanking pixels.

13.9.3.5.7 Frame Composer Input Video VActive Pixels Register 0 (fc_invactiv0)

13.9.3.5.7.1 Offset

Register	Offset
fc_invactiv0	5h

13.9.3.5.7.2 Diagram



13.9.3.5.7.3 Fields

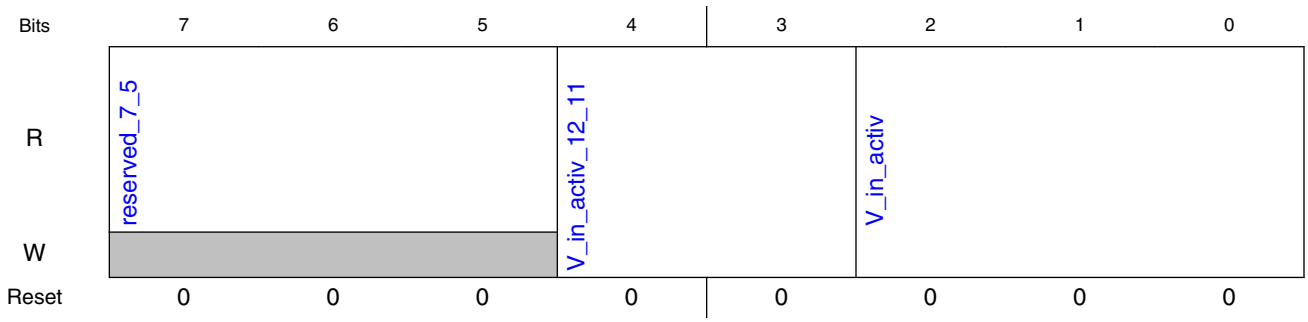
Field	Function
7-0 V_in_activ	Input video Vertical active pixel region width. Input video Vertical active pixel region width. This bit field holds bits 7:0 of number of Vertical active pixels.

13.9.3.5.8 Frame Composer Input Video VActive Pixels Register 1 (fc_invactiv1)

13.9.3.5.8.1 Offset

Register	Offset
fc_invactiv1	6h

13.9.3.5.8.2 Diagram



13.9.3.5.8.3 Fields

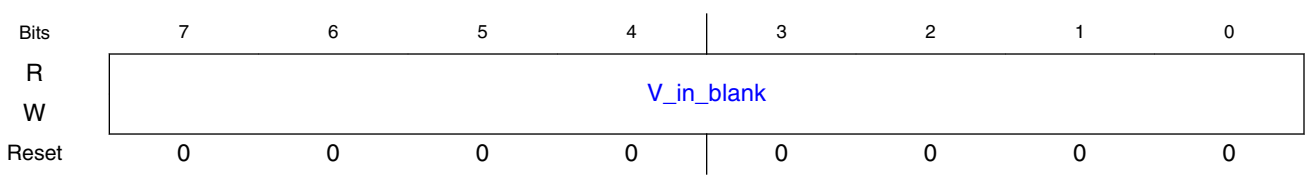
Field	Function
7-5 reserved_7_5	Reserved and read as zero
4-3 V_in_activ_12_11	Input video Vertical active pixel region width. Input video Vertical active pixel region width. If the configuration parameter DWC_HDMI_TX_14 = True (1), this bit field holds bits 12:10 of number of Vertical active pixels.
2-0 V_in_activ	Input video Vertical active pixel region width. Input video Vertical active pixel region width. This bit field holds bits 9:8 of number of Vertical active pixels.

13.9.3.5.9 Frame Composer Input Video VBlank Pixels Register (fc_invblank)

13.9.3.5.9.1 Offset

Register	Offset
fc_invblank	7h

13.9.3.5.9.2 Diagram



13.9.3.5.9.3 Fields

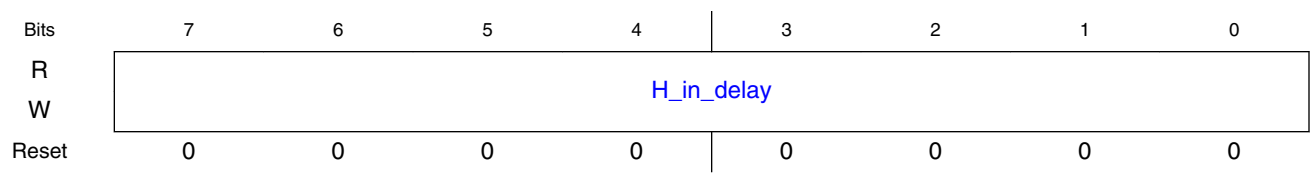
Field	Function
7-0 V_in_blank	Input video Vertical blanking pixel region width. Input video Vertical blanking pixel region width. Number of Vertical blanking lines [0...255].

13.9.3.5.10 Frame Composer Input Video HSync Front Porch Register 0 (fc_hsyncindelay0)

13.9.3.5.10.1 Offset

Register	Offset
fc_hsyncindelay0	8h

13.9.3.5.10.2 Diagram



13.9.3.5.10.3 Fields

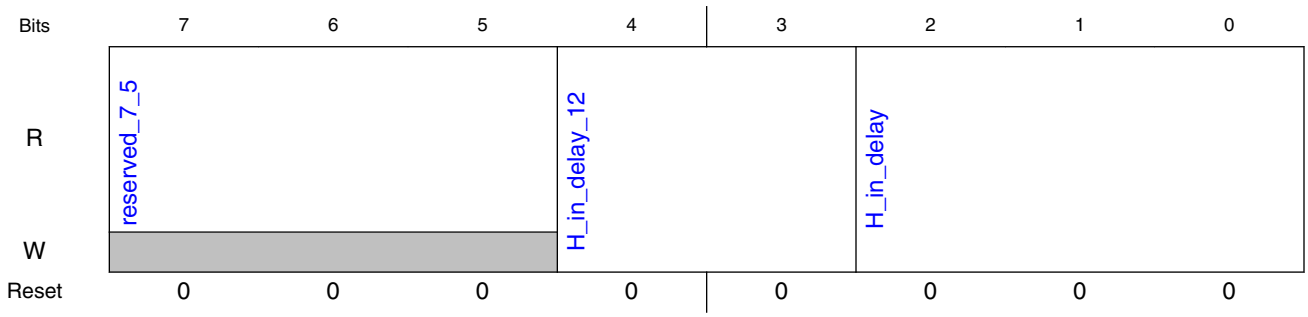
Field	Function
7-0 H_in_delay	Input video Hsync active edge delay. Input video Hsync active edge delay. Integer number of pixel clock cycles from "de" non active edge of the last "de" valid period [0...4095].

13.9.3.5.11 Frame Composer Input Video HSync Front Porch Register 1 (fc_hsyncindelay1)

13.9.3.5.11.1 Offset

Register	Offset
fc_hsyncindelay1	9h

13.9.3.5.11.2 Diagram



13.9.3.5.11.3 Fields

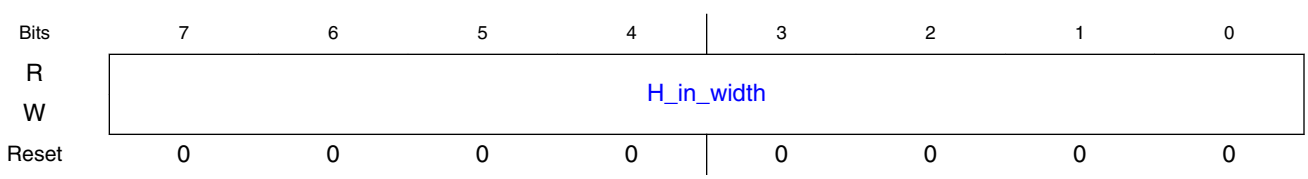
Field	Function
7-5 reserved_7_5	Reserved and read as zero
4-3 H_in_delay_12	Input video Horizontal active edge delay. Input video Horizontal active edge delay. If configuration parameter DWC_HDMI_TX_14 = True (1), this bit field holds bit 12. Integer number of pixel clock cycles from "de" non-active edge of the last "de" valid period [0...8191].
2-0 H_in_delay	Input video Horizontal active edge delay.

13.9.3.5.12 Frame Composer Input Video HSync Width Register 0 (fc_hsyncinwidth0)

13.9.3.5.12.1 Offset

Register	Offset
fc_hsyncinwidth0	Ah

13.9.3.5.12.2 Diagram



13.9.3.5.12.3 Fields

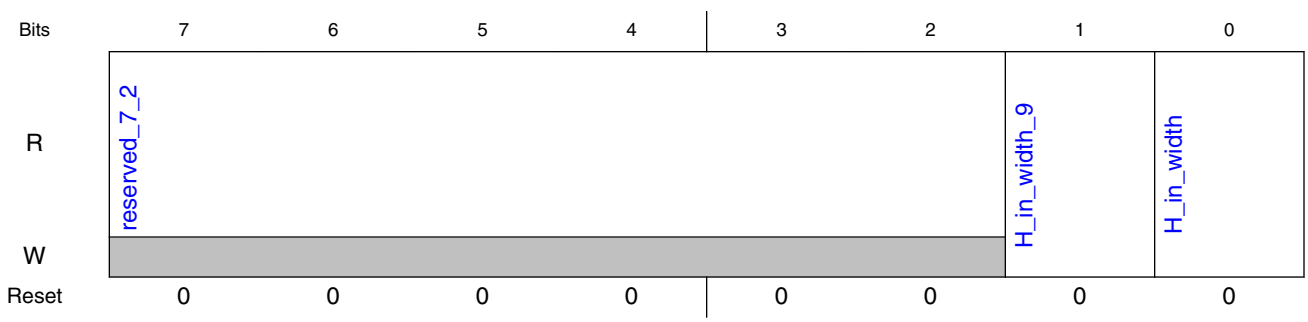
Field	Function
7-0 H_in_width	Input video Hsync active pulse width. Input video Hsync active pulse width. Integer number of pixel clock cycles [0...511].

13.9.3.5.13 Frame Composer Input Video HSync Width Register 1 (fc_hsyncinwidth1)

13.9.3.5.13.1 Offset

Register	Offset
fc_hsyncinwidth1	Bh

13.9.3.5.13.2 Diagram



13.9.3.5.13.3 Fields

Field	Function
7-2 reserved_7_2	Reserved and read as zero
1 H_in_width_9	Input video Hsync active pulse width. Input video Hsync active pulse width. If configuration parameter DWC_HDMI_TX_14 = True (1), then this bit field holds bit 9. Number of Horizontal active pixels [0...1024].
0 H_in_width	Input video Hsync active pulse width.

13.9.3.5.14 Frame Composer Input Video VSync Front Porch Register (fc_vsyncindelay)

13.9.3.5.14.1 Offset

Register	Offset
fc_vsyncindelay	Ch

13.9.3.5.14.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	V_in_delay							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.14.3 Fields

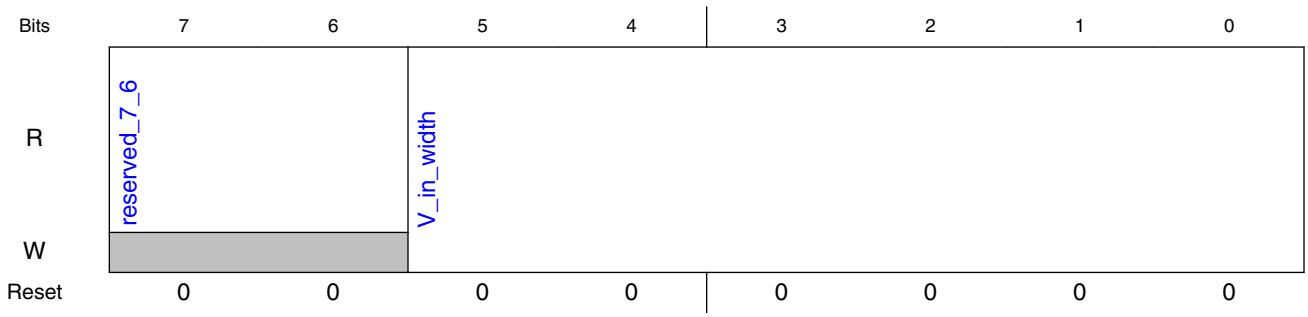
Field	Function
7-0 V_in_delay	Input video Vsync active edge delay. Input video Vsync active edge delay. Integer number of Hsync pulses from "de" non active edge of the last "de" valid period. [0...255].

13.9.3.5.15 Frame Composer Input Video VSync Width Register (fc_vsyncinwidth)

13.9.3.5.15.1 Offset

Register	Offset
fc_vsyncinwidth	Dh

13.9.3.5.15.2 Diagram



13.9.3.5.15.3 Fields

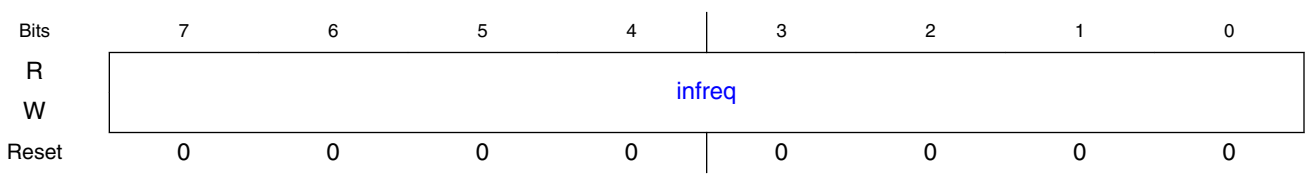
Field	Function
7-6 reserved_7_6	Reserved and read as zero
5-0 V_in_width	Input video Vsync active pulse width. Input video Vsync active pulse width. Integer number of video lines [0...63].

13.9.3.5.16 Frame Composer Input Video Refresh Rate Register 0 (fc_infreq0)

13.9.3.5.16.1 Offset

Register	Offset
fc_infreq0	Eh

13.9.3.5.16.2 Diagram



13.9.3.5.16.3 Fields

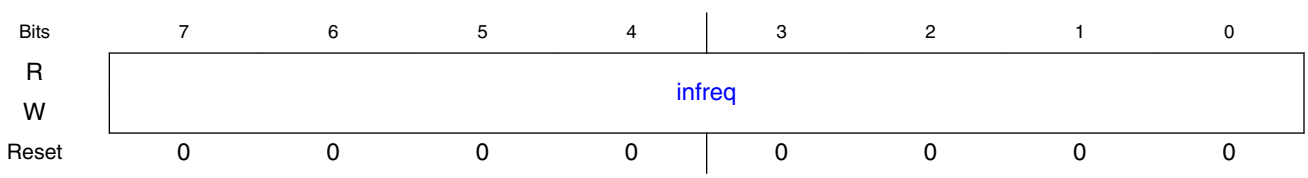
Field	Function
7-0 infreq	Video refresh rate in Hz*1E3 format. Video refresh rate in Hz*1E3 format. This register is provided for debug and informative purposes. The DWC_hdmi_tx does not write any data to this register; the data written by software is not used by the DWC_hdmi_tx.

13.9.3.5.17 Frame Composer Input Video Refresh Rate Register 1 (fc_infreq1)

13.9.3.5.17.1 Offset

Register	Offset
fc_infreq1	Fh

13.9.3.5.17.2 Diagram



13.9.3.5.17.3 Fields

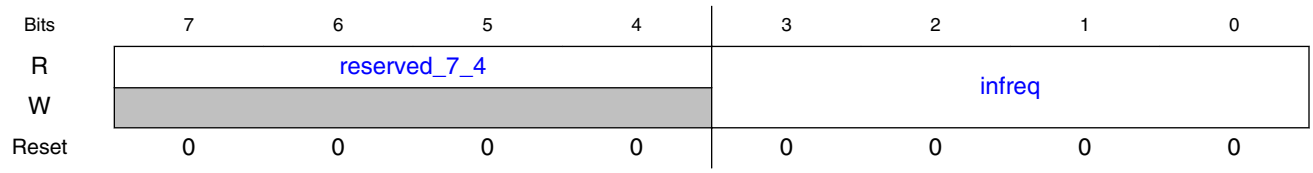
Field	Function
7-0 infreq	Video refresh rate in Hz*1E3 format. Video refresh rate in Hz*1E3 format. This register is provided for debug and informative purposes. The DWC_hdmi_tx does not write any data to this register; the data written by software is not used by the DWC_hdmi_tx.

13.9.3.5.18 Frame Composer Input Video Refresh Rate Register 2 (fc_infreq2)

13.9.3.5.18.1 Offset

Register	Offset
fc_infreq2	10h

13.9.3.5.18.2 Diagram



13.9.3.5.18.3 Fields

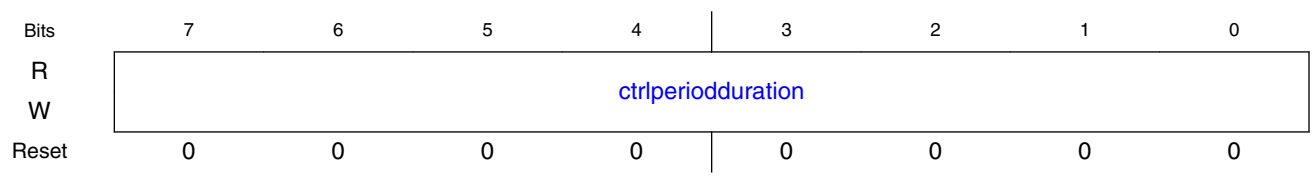
Field	Function
7-4 reserved_7_4	Reserved and read as zero
3-0 infreq	Video refresh rate in Hz*1E3 format. Video refresh rate in Hz*1E3 format. This register is provided for debug and informative purposes. The DWC_hdmi_tx does not write any data to this register; the data written by software is not used by the DWC_hdmi_tx.

13.9.3.5.19 Frame Composer Control Period Duration Register (fc_ctrldur)

13.9.3.5.19.1 Offset

Register	Offset
fc_ctrldur	11h

13.9.3.5.19.2 Diagram



13.9.3.5.19.3 Fields

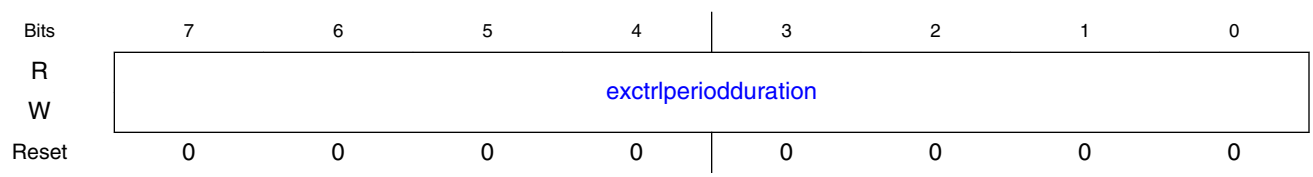
Field	Function
7-0 ctrlperioddduration	Configuration of the control period minimum duration (minimum of 12 pixel clock cycles; refer to HDMI 1. Configuration of the control period minimum duration (minimum of 12 pixel clock cycles; refer to HDMI 1.4b specification). Integer number of pixel clocks cycles [0..223].

13.9.3.5.20 Frame Composer Extended Control Period Duration Register (fc_exctrldur)

13.9.3.5.20.1 Offset

Register	Offset
fc_exctrldur	12h

13.9.3.5.20.2 Diagram



13.9.3.5.20.3 Fields

Field	Function
7-0 exctrlperioddduration	Configuration of the extended control period minimum duration (minimum of 32 pixel clock cycles; refer to HDMI 1. Configuration of the extended control period minimum duration (minimum of 32 pixel clock cycles; refer to HDMI 1.4b specification). Integer number of pixel clocks cycles [0..223].

13.9.3.5.21 Frame Composer Extended Control Period Maximum Spacing Register (fc_exctrlspac)

13.9.3.5.21.1 Offset

Register	Offset
fc_exctrlspac	13h

13.9.3.5.21.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	exctrlperiodspacing							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.21.3 Fields

Field	Function
7-0 exctrlperiodspacing	Configuration of the maximum spacing between consecutive extended control periods (maximum of 50ms; refer to the applicable HDMI specification). Configuration of the maximum spacing between consecutive extended control periods (maximum of 50ms; refer to the applicable HDMI specification). When using the HDMI 2.0 supported features (DWC_HDMI_TX_20 = 1): - generated spacing = (1/freq tmds clock)*256*512*(extctrlperiodspacing +1) else - generated spacing = (1/freq tmds clock)*256*256*(extctrlperiodspacing +1)

13.9.3.5.22 Frame Composer Channel 0 Non-Preamble Data Register (fc_ch0pream)

13.9.3.5.22.1 Offset

Register	Offset
fc_ch0pream	14h

13.9.3.5.22.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	ch0_preamble_filter							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.22.3 Fields

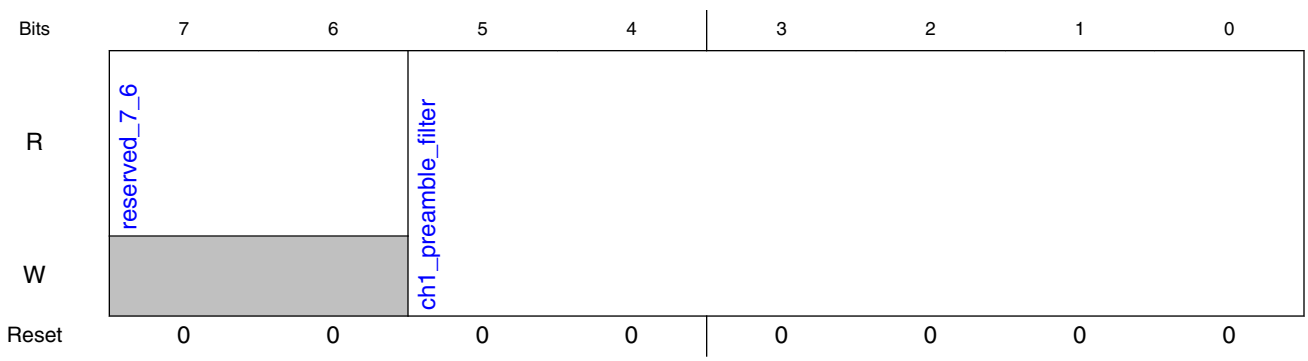
Field	Function
7-0 ch0_preamble_filter	When in control mode, configures 8 bits that fill the channel 0 data lines not used to transmit the preamble (for more clarification, refer to the HDMI 1. When in control mode, configures 8 bits that fill the channel 0 data lines not used to transmit the preamble (for more clarification, refer to the HDMI 1.4b specification).

13.9.3.5.23 Frame Composer Channel 1 Non-Preamble Data Register (fc_ch1pream)

13.9.3.5.23.1 Offset

Register	Offset
fc_ch1pream	15h

13.9.3.5.23.2 Diagram



13.9.3.5.23.3 Fields

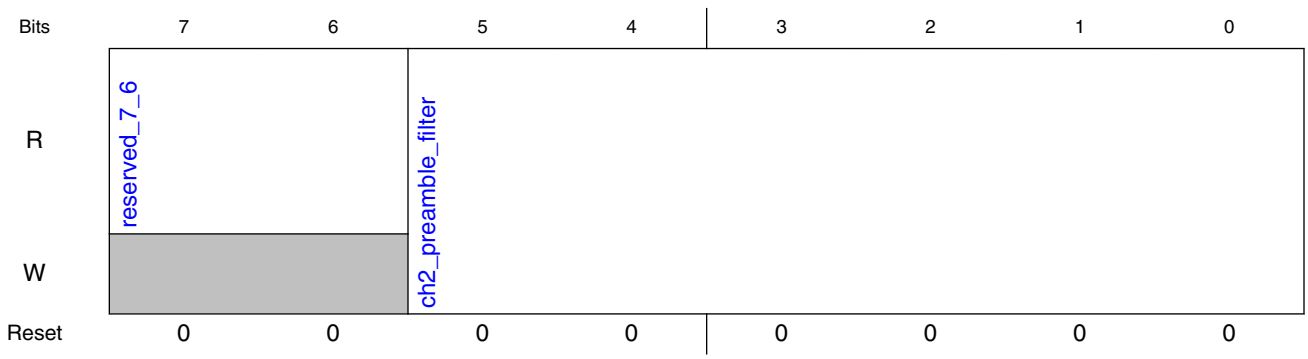
Field	Function
7-6 reserved_7_6	Reserved and read as zero
5-0 ch1_preamble_filter	When in control mode, configures 6 bits that fill the channel 1 data lines not used to transmit the preamble (for more clarification, refer to the HDMI 1. When in control mode, configures 6 bits that fill the channel 1 data lines not used to transmit the preamble (for more clarification, refer to the HDMI 1.4b specification).

13.9.3.5.24 Frame Composer Channel 2 Non-Preamble Data Register (fc_ch2pream)

13.9.3.5.24.1 Offset

Register	Offset
fc_ch2pream	16h

13.9.3.5.24.2 Diagram



13.9.3.5.24.3 Fields

Field	Function
7-6 reserved_7_6	Reserved and read as zero
5-0 ch2_preamble_filter	When in control mode, configures 6 bits that fill the channel 2 data lines not used to transmit the preamble (for more clarification, refer to the HDMI 1. When in control mode, configures 6 bits that fill the channel 2 data lines not used to transmit the preamble (for more clarification, refer to the HDMI 1.4b specification).

13.9.3.5.25 Frame Composer AVI Packet Configuration Register 3 (fc_aviconf3)

13.9.3.5.25.1 Offset

Register	Offset
fc_aviconf3	17h

13.9.3.5.25.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	reserved_7_4				YQ		CN	
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.5.25.3 Fields

Field	Function
7-4 reserved_7_4	Reserved and read as zero
3-2 YQ	YCbCr Quantization range according to the CEA specification
1-0 CN	IT content type according to CEA the specification

13.9.3.5.26 Frame Composer GCP Packet Configuration Register (fc_gcp)

13.9.3.5.26.1 Offset

Register	Offset
fc_gcp	18h

13.9.3.5.26.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	reserved_7_3					default_phase	set_avmute	clear_avmute
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.5.26.3 Fields

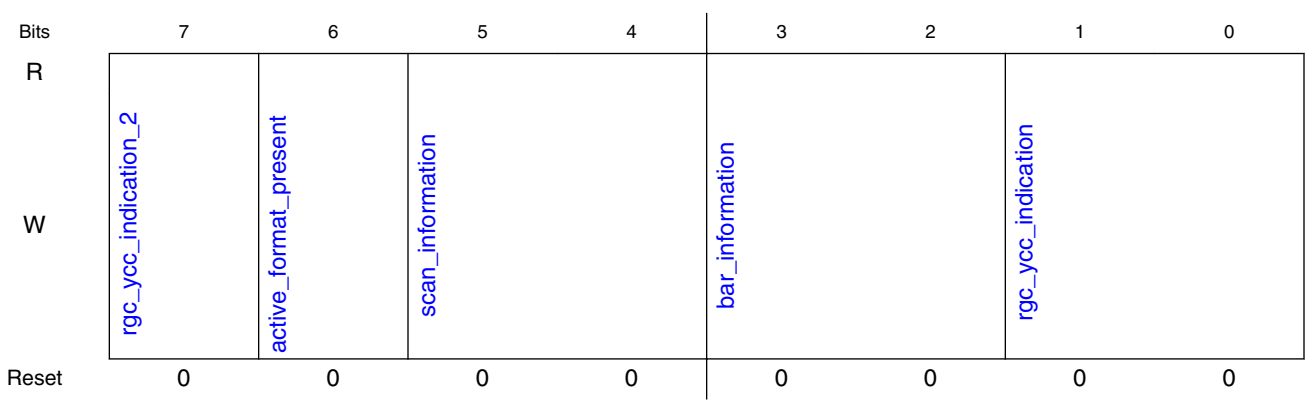
Field	Function
7-3 reserved_7_3	Reserved and read as zero
2 default_phase	Value of "default_phase" in the GCP packet. Value of "default_phase" in the GCP packet. This data must be equal to the default phase used at Video Packetizer packing machine.
1 set_avmute	Value of "set_avmute" in the GCP packet Once the AVmute is set, the frame composer schedules the GCP packet with AVmute set in the packet scheduler to be sent once (may only be transmitted between the active edge of VSYNC and 384 pixels following this edge).
0 clear_avmute	Value of "clear_avmute" in the GCP packet

13.9.3.5.27 Frame Composer AVI Packet Configuration Register 0 (fc_aviconf0)

13.9.3.5.27.1 Offset

Register	Offset
fc_aviconf0	19h

13.9.3.5.27.2 Diagram



13.9.3.5.27.3 Fields

Field	Function
7	Y2, Bit 2 of rgc_ycc_indication

Table continues on the next page...

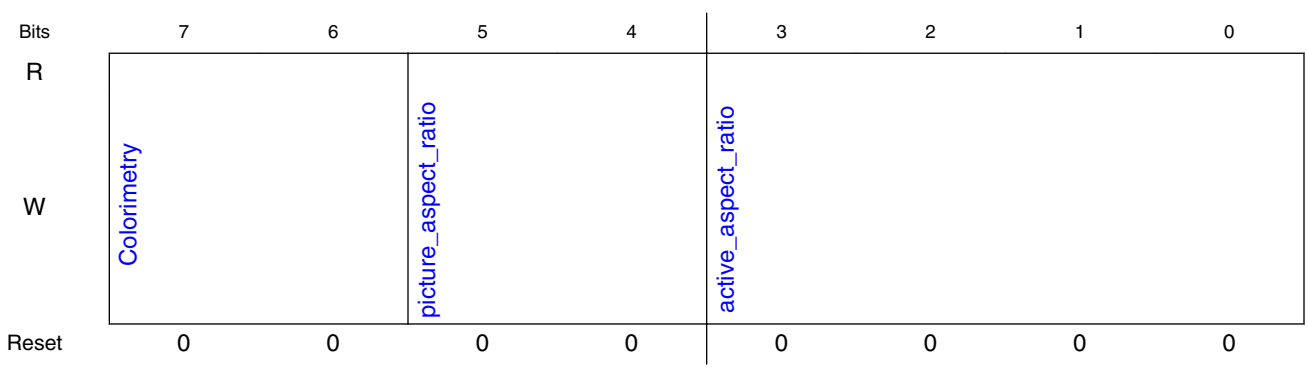
Field	Function
rgc_ycc_indication_2	
6 active_format_present	Active format present
5-4 scan_information	Scan information
3-2 bar_information	Bar information data valid
1-0 rgc_ycc_indication	Y1,Y0 RGB or YCbCr indicator

13.9.3.5.28 Frame Composer AVI Packet Configuration Register 1 (fc_aviconf1)

13.9.3.5.28.1 Offset

Register	Offset
fc_aviconf1	1Ah

13.9.3.5.28.2 Diagram



13.9.3.5.28.3 Fields

Field	Function
7-6	Colorimetry

Table continues on the next page...

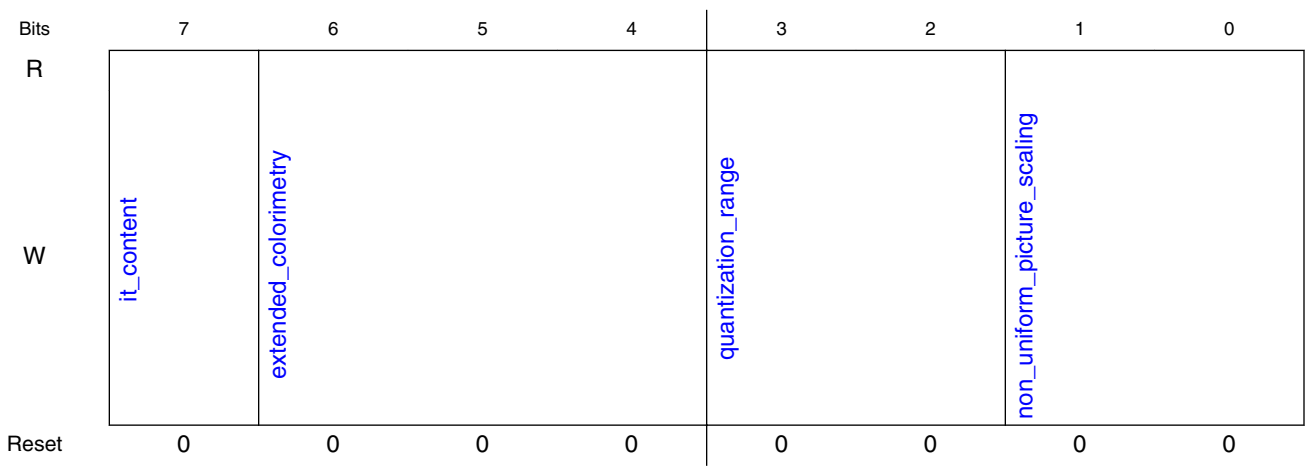
Field	Function
Colorimetry	
5-4 picture_aspect_ratio	Picture aspect ratio
3-0 active_aspect_ratio	Active aspect ratio

13.9.3.5.29 Frame Composer AVI Packet Configuration Register 2 (fc_aviconf2)

13.9.3.5.29.1 Offset

Register	Offset
fc_aviconf2	1Bh

13.9.3.5.29.2 Diagram



13.9.3.5.29.3 Fields

Field	Function
7 it_content	IT content
6-4	Extended colorimetry

Table continues on the next page...

Field	Function
extended_colorimetry	
3-2 quantization_range	Quantization range
1-0 non_uniform_picture_scaling	Non-uniform picture scaling

13.9.3.5.30 Frame Composer AVI Packet VIC Register (fc_avivid)

13.9.3.5.30.1 Offset

Register	Offset
fc_avivid	1Ch

13.9.3.5.30.2 Diagram

13.9.3.5.30.3 Fields

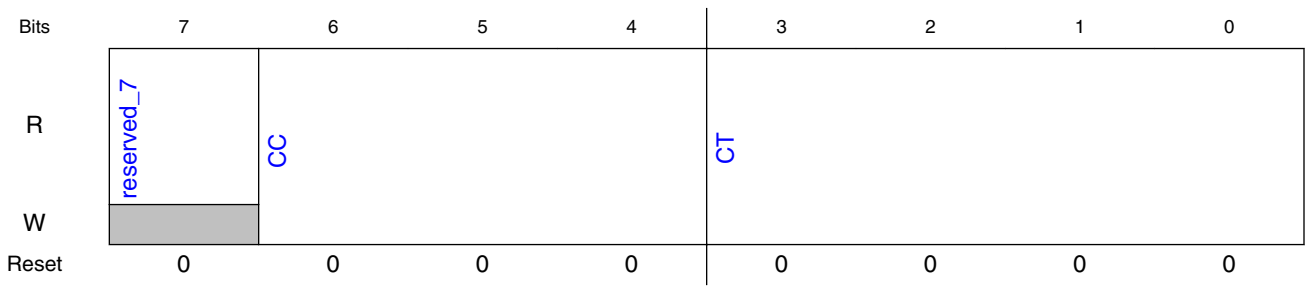
Field	Function
7 fc_avivid_7	Bit 7 of fc_avivid register
6-0 fc_avivid	Configures the AVI InfoFrame Video Identification code. Configures the AVI InfoFrame Video Identification code. For more information, refer to the CEA-861-E specification.

13.9.3.5.31 Frame Composer AUD Packet Configuration Register 0 (fc_audiconf0)

13.9.3.5.31.1 Offset

Register	Offset
fc_audiconf0	25h

13.9.3.5.31.2 Diagram



13.9.3.5.31.3 Fields

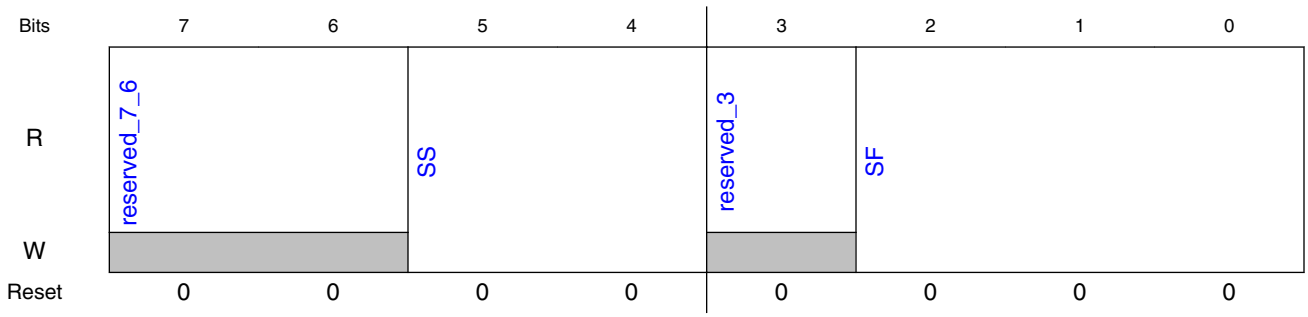
Field	Function
7 reserved_7	Reserved and read as zero
6-4 CC	Channel count
3-0 CT	Coding Type

13.9.3.5.32 Frame Composer AUD Packet Configuration Register 1 (fc_audiconf1)

13.9.3.5.32.1 Offset

Register	Offset
fc_audiconf1	26h

13.9.3.5.32.2 Diagram



13.9.3.5.32.3 Fields

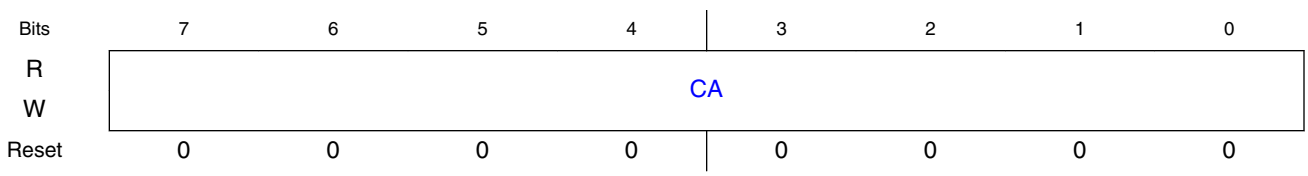
Field	Function
7-6 reserved_7_6	Reserved and read as zero
5-4 SS	Sampling size
3 reserved_3	Reserved and read as zero
2-0 SF	Sampling frequency

13.9.3.5.33 Frame Composer AUD Packet Configuration Register 2 (fc_audiconf2)

13.9.3.5.33.1 Offset

Register	Offset
fc_audiconf2	27h

13.9.3.5.33.2 Diagram



13.9.3.5.33.3 Fields

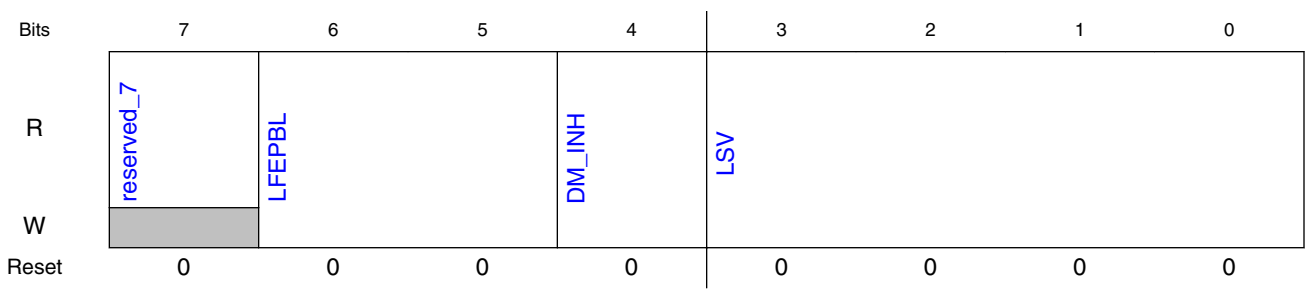
Field	Function
7-0 CA	Channel allocation

13.9.3.5.34 Frame Composer AUD Packet Configuration Register 3 (fc_audiconf3)

13.9.3.5.34.1 Offset

Register	Offset
fc_audiconf3	28h

13.9.3.5.34.2 Diagram



13.9.3.5.34.3 Fields

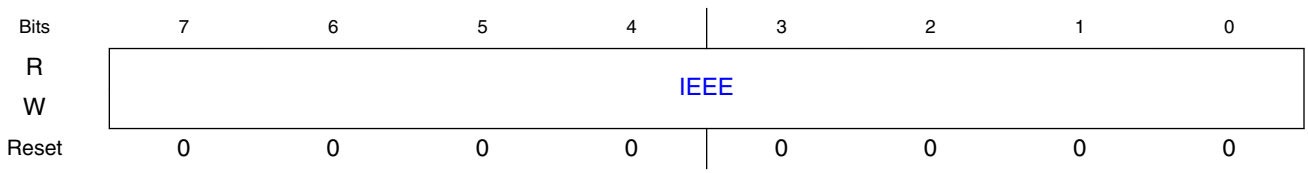
Field	Function
7 reserved_7	Reserved and read as zero
6-5 LFEPBL	LFE playback information LFEPBL1, LFEPBL0 LFE playback level as compared to the other channels.
4 DM_INH	Down mix enable
3-0 LSV	Level shift value (for down mixing)

13.9.3.5.35 Frame Composer VSI Packet Data IEEE Register 0 (fc_vsdieeeid0)

13.9.3.5.35.1 Offset

Register	Offset
fc_vsdieeeid0	29h

13.9.3.5.35.2 Diagram



13.9.3.5.35.3 Fields

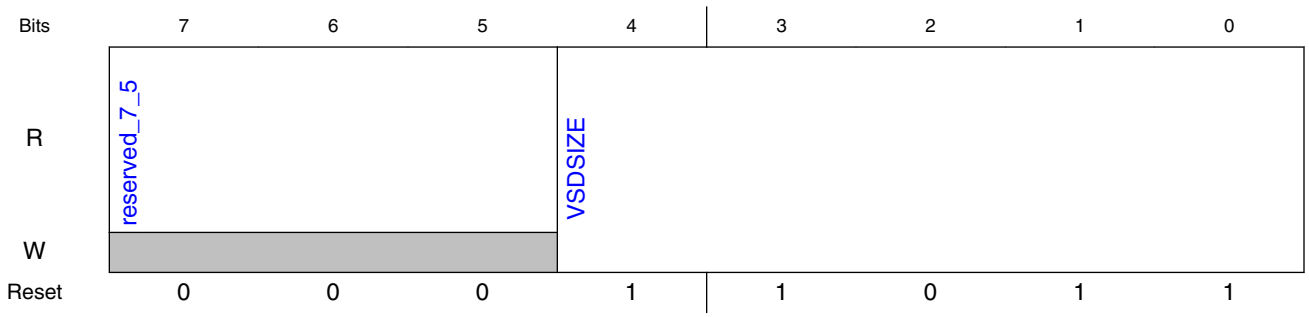
Field	Function
7-0 IEEE	This register configures the Vendor Specific InfoFrame IEEE registration identifier. This register configures the Vendor Specific InfoFrame IEEE registration identifier. For more information, refer to the CEA-861-E specification.

13.9.3.5.36 Frame Composer VSI Packet Data Size Register (fc_vsdsize)

13.9.3.5.36.1 Offset

Register	Offset
fc_vsdsize	2Ah

13.9.3.5.36.2 Diagram



13.9.3.5.36.3 Fields

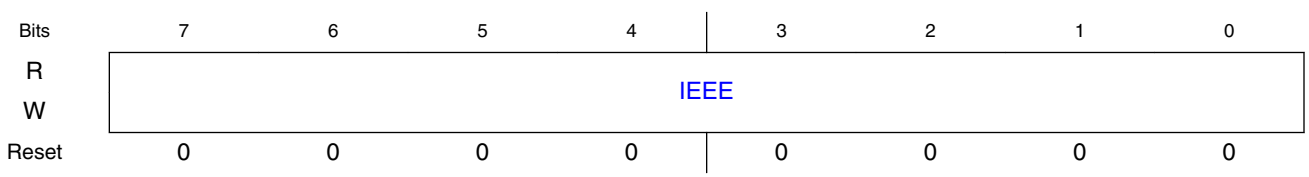
Field	Function
7-5 reserved_7_5	Reserved and read as zero
4-0 VSDSIZE	Packet size as described in the HDMI Vendor Specific InfoFrame (from the HDMI specification).

13.9.3.5.37 Frame Composer VSI Packet Data IEEE Register 1 (fc_vsdieeeid1)

13.9.3.5.37.1 Offset

Register	Offset
fc_vsdieeeid1	30h

13.9.3.5.37.2 Diagram



13.9.3.5.37.3 Fields

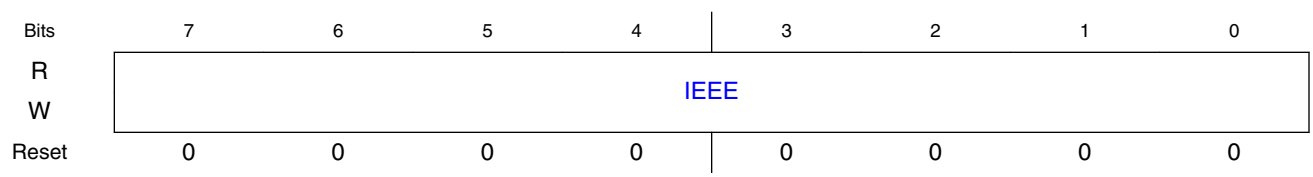
Field	Function
7-0 IEEE	This register configures the Vendor Specific InfoFrame IEEE registration identifier. This register configures the Vendor Specific InfoFrame IEEE registration identifier. For more information, refer to the CEA-861-E specification.

13.9.3.5.38 Frame Composer VSI Packet Data IEEE Register 2 (fc_vsdieeeid2)

13.9.3.5.38.1 Offset

Register	Offset
fc_vsdieeeid2	31h

13.9.3.5.38.2 Diagram



13.9.3.5.38.3 Fields

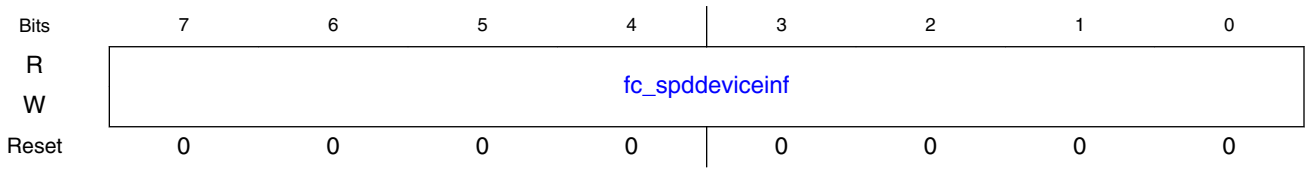
Field	Function
7-0 IEEE	This register configures the Vendor Specific InfoFrame IEEE registration identifier. This register configures the Vendor Specific InfoFrame IEEE registration identifier. For more information, refer to the CEA-861-E specification.

13.9.3.5.39 Frame Composer SPD Packet Data Source Product Descriptor Register (fc_spddeviceinf)

13.9.3.5.39.1 Offset

Register	Offset
fc_spddeviceinf	62h

13.9.3.5.39.2 Diagram



13.9.3.5.39.3 Fields

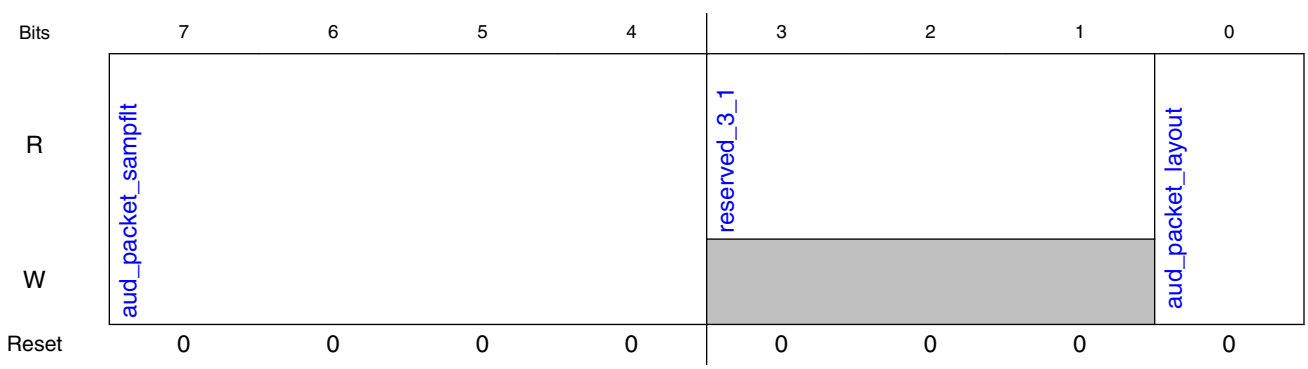
Field	Function
7-0 fc_spddeviceinf	Frame Composer SPD Packet Data Source Product Descriptor Register

13.9.3.5.40 Frame Composer Audio Sample Flat and Layout Configuration Register (fc_audsconf)

13.9.3.5.40.1 Offset

Register	Offset
fc_audsconf	63h

13.9.3.5.40.2 Diagram



13.9.3.5.40.3 Fields

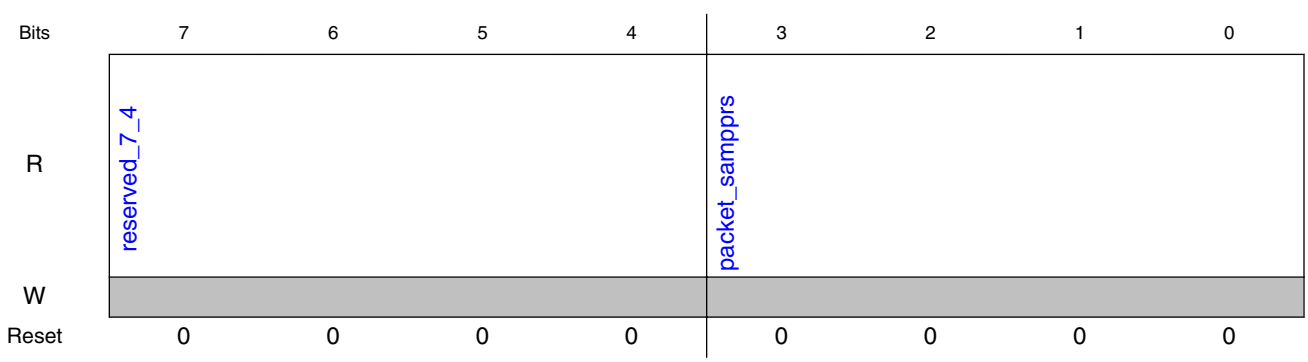
Field	Function
7-4 aud_packet_sampflt	Set the audio packet sample flat value to be sent on the packet.
3-1 reserved_3_1	Reserved and read as zero
0 aud_packet_layout	Set the audio packet layout to be sent in the packet: 1b: layout 1 0b: layout 0 If DWC_HDMI_TX_20 is defined and register field fc_multistream_ctrl. Set the audio packet layout to be sent in the packet: 1b: layout 1 0b: layout 0 If DWC_HDMI_TX_20 is defined and register field fc_multistream_ctrl.fc_mas_packet_en is active, this bit has no effect.

13.9.3.5.41 Frame Composer Audio Sample Flat and Layout Status Register (fc_audsstat)

13.9.3.5.41.1 Offset

Register	Offset
fc_audsstat	64h

13.9.3.5.41.2 Diagram



13.9.3.5.41.3 Fields

Field	Function
7-4 reserved_7_4	Reserved and read as zero
3-0 packet_sampprs	Shows the data sample present indication of the last Audio sample packet sent by the HDMI TX Controller. Shows the data sample present indication of the last Audio sample packet sent by the HDMI TX Controller. This register information is at TMDS clock rate.

13.9.3.5.42 Frame Composer Audio Sample Validity Flag Register (fc_audsv)

13.9.3.5.42.1 Offset

Register	Offset
fc_audsv	65h

13.9.3.5.42.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	V3r	V2r	V1r	V0r	V3l	V2l	V1l	V0l
Reset	0	0	0	0	0	0	0	0

13.9.3.5.42.3 Fields

Field	Function
7 V3r	Set validity bit "V" for Channel 3, Right
6 V2r	Set validity bit "V" for Channel 2, Right
5 V1r	Set validity bit "V" for Channel 1, Right
4 V0r	Set validity bit "V" for Channel 0, Right
3 V3l	Set validity bit "V" for Channel 3, Left

Table continues on the next page...

Field	Function
2 V2l	Set validity bit "V" for Channel 2, Left
1 V1l	Set validity bit "V" for Channel 1, Left
0 V0l	Set validity bit "V" for Channel 0, Left

13.9.3.5.43 Frame Composer Audio Sample User Flag Register (fc_audsu)

13.9.3.5.43.1 Offset

Register	Offset
fc_audsu	66h

13.9.3.5.43.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	U3r	U2r	U1r	U0r	U3l	U2l	U1l	U0l
Reset	0	0	0	0	0	0	0	0

13.9.3.5.43.3 Fields

Field	Function
7 U3r	Set user bit "U" for Channel 3, Right
6 U2r	Set user bit "U" for Channel 2, Right
5 U1r	Set user bit "U" for Channel 1, Right
4 U0r	Set user bit "U" for Channel 0, Right
3 U3l	Set user bit "U" for Channel 3, Left
2 U2l	Set user bit "U" for Channel 2, Left

Table continues on the next page...

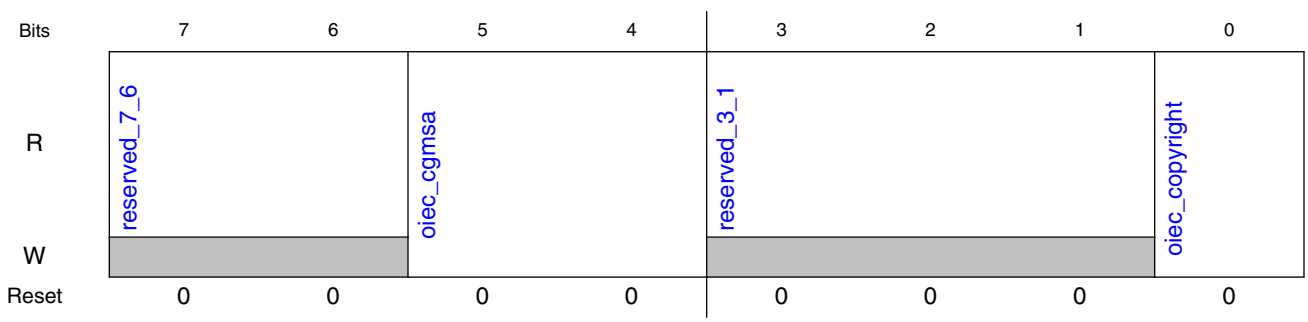
Field	Function
1 U1I	Set user bit "U" for Channel 1, Left
0 U0I	Set user bit "U" for Channel 0, Left

13.9.3.5.44 Frame Composer Audio Sample Channel Status Configuration Register 0 (fc_audschnl0)

13.9.3.5.44.1 Offset

Register	Offset
fc_audschnl0	67h

13.9.3.5.44.2 Diagram



13.9.3.5.44.3 Fields

Field	Function
7-6 reserved_7_6	Reserved and read as zero
5-4 oiec_cgmsa	CGMS-A
3-1 reserved_3_1	Reserved and read as zero
0 oiec_copyright	IEC Copyright indication

13.9.3.5.45 Frame Composer Audio Sample Channel Status Configuration Register 1 (fc_audschnl1)

13.9.3.5.45.1 Offset

Register	Offset
fc_audschnl1	68h

13.9.3.5.45.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	oiec_categorycode							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.45.3 Fields

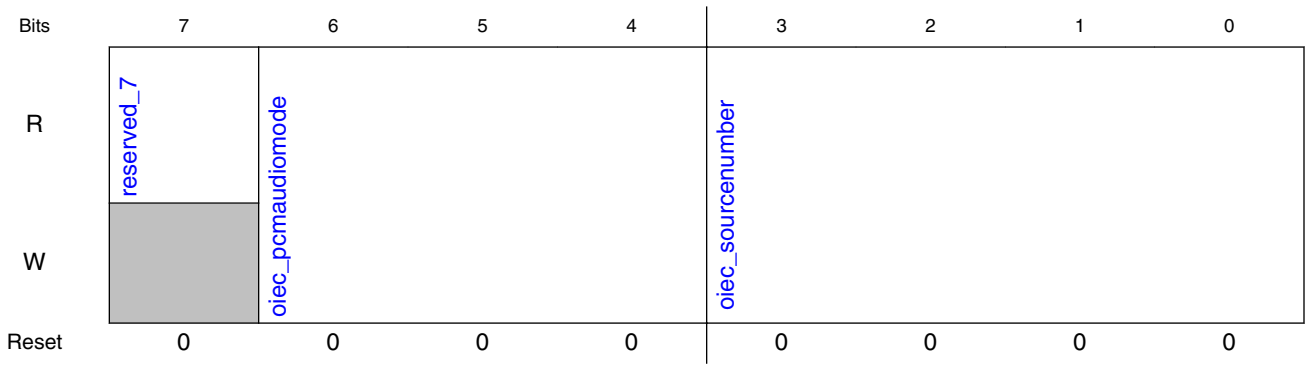
Field	Function
7-0 oiec_categorycode	Category code

13.9.3.5.46 Frame Composer Audio Sample Channel Status Configuration Register 2 (fc_audschnl2)

13.9.3.5.46.1 Offset

Register	Offset
fc_audschnl2	69h

13.9.3.5.46.2 Diagram



13.9.3.5.46.3 Fields

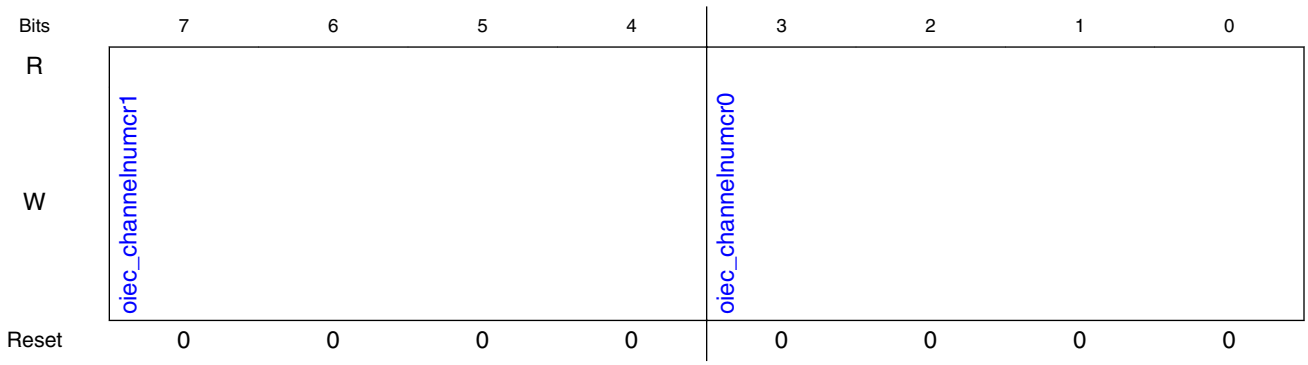
Field	Function
7 reserved_7	Reserved and read as zero
6-4 oiec_pcmaudiomode	PCM audio mode
3-0 oiec_sourcenumber	Source number

13.9.3.5.47 Frame Composer Audio Sample Channel Status Configuration Register 3 (fc_audschnl3)

13.9.3.5.47.1 Offset

Register	Offset
fc_audschnl3	6Ah

13.9.3.5.47.2 Diagram



13.9.3.5.47.3 Fields

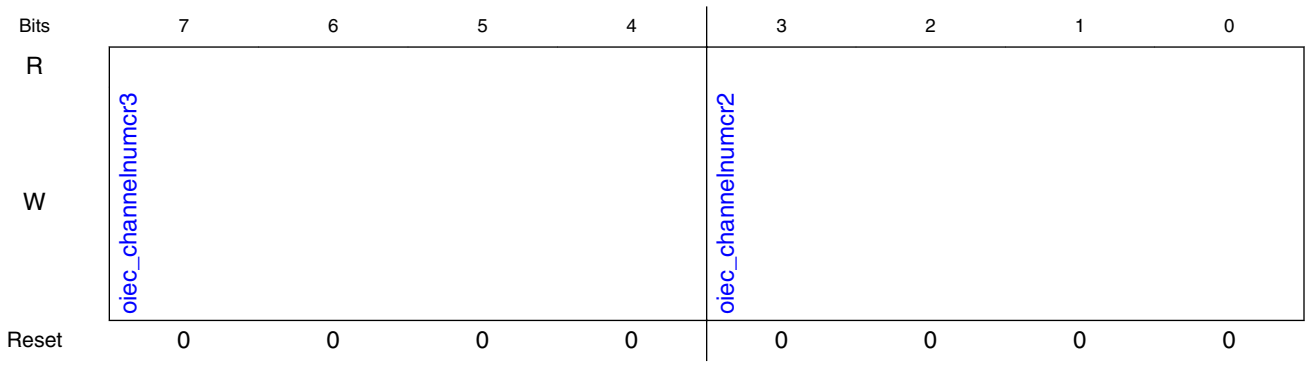
Field	Function
7-4 oiec_channelnumcr1	Channel number for second right sample
3-0 oiec_channelnumcr0	Channel number for first right sample

13.9.3.5.48 Frame Composer Audio Sample Channel Status Configuration Register 4 (fc_audschnl4)

13.9.3.5.48.1 Offset

Register	Offset
fc_audschnl4	6Bh

13.9.3.5.48.2 Diagram



13.9.3.5.48.3 Fields

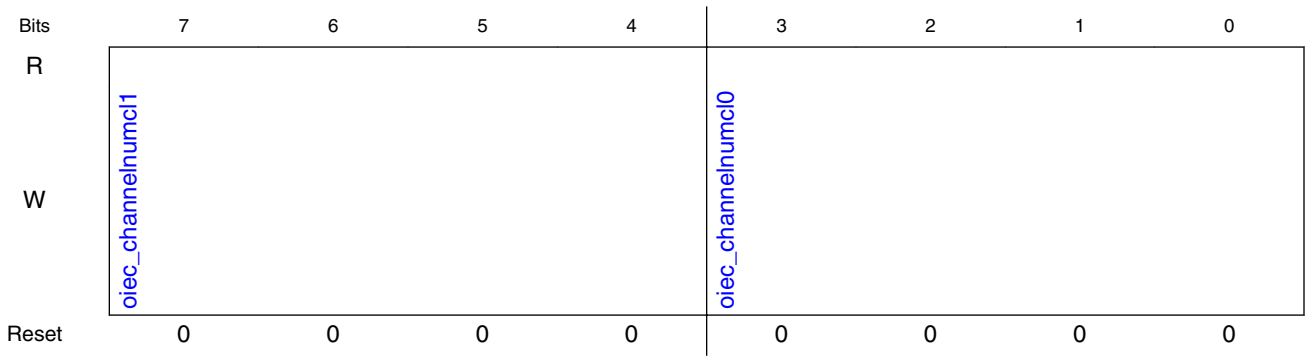
Field	Function
7-4 oiec_channelnumcr3	Channel number for fourth right sample
3-0 oiec_channelnumcr2	Channel number for third right sample

13.9.3.5.49 Frame Composer Audio Sample Channel Status Configuration Register 5 (fc_audschnl5)

13.9.3.5.49.1 Offset

Register	Offset
fc_audschnl5	6Ch

13.9.3.5.49.2 Diagram



13.9.3.5.49.3 Fields

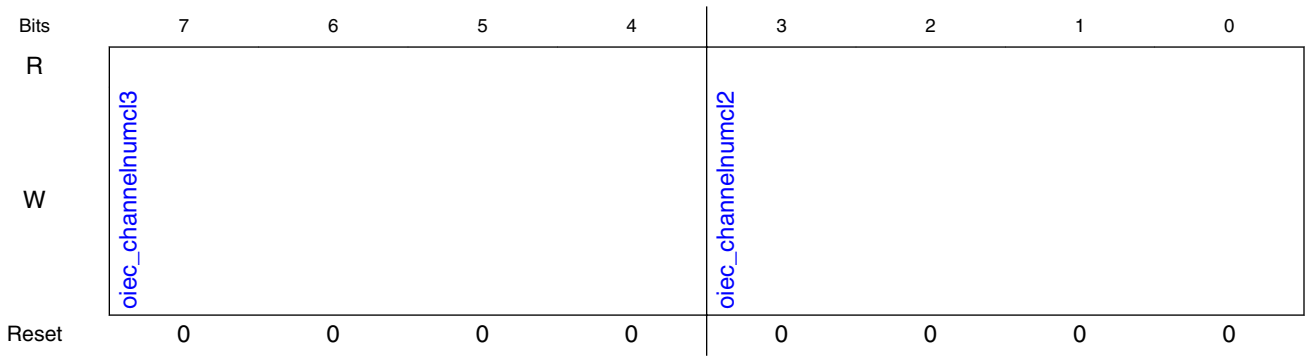
Field	Function
7-4 oiec_channelnumcl1	Channel number for second left sample
3-0 oiec_channelnumcl0	Channel number for first left sample

13.9.3.5.50 Frame Composer Audio Sample Channel Status Configuration Register 6 (fc_audschnl6)

13.9.3.5.50.1 Offset

Register	Offset
fc_audschnl6	6Dh

13.9.3.5.50.2 Diagram



13.9.3.5.50.3 Fields

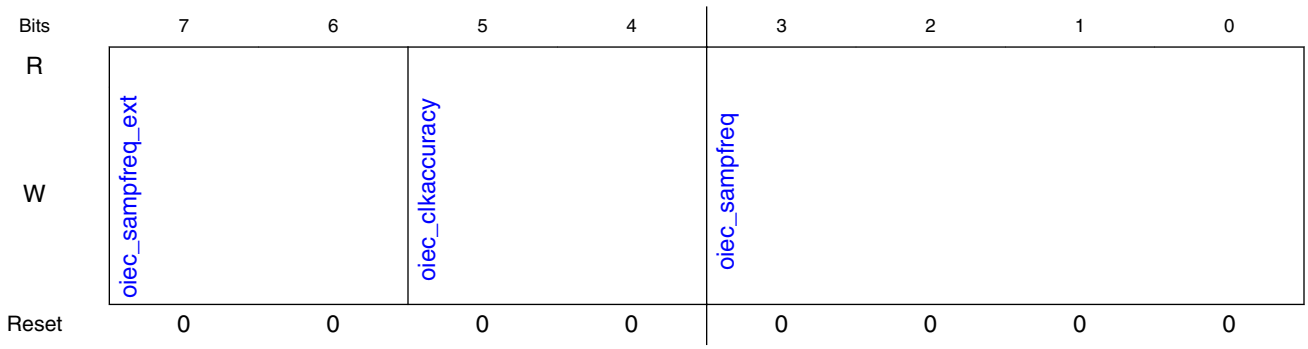
Field	Function
7-4 oiec_channelnumcl3	Channel number for fourth left sample
3-0 oiec_channelnumcl2	Channel number for third left sample

13.9.3.5.51 Frame Composer Audio Sample Channel Status Configuration Register 7 (fc_audschnl7)

13.9.3.5.51.1 Offset

Register	Offset
fc_audschnl7	6Eh

13.9.3.5.51.2 Diagram



13.9.3.5.51.3 Fields

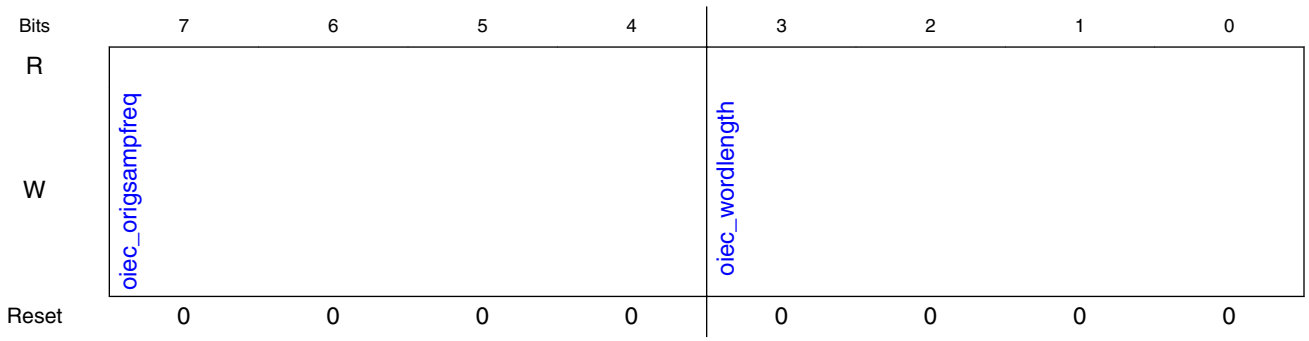
Field	Function
7-6 oiec_sampfreq_ext	Sampling frequency (channel status bits 31 and 30)
5-4 oiec_clkaccuracy	Clock accuracy
3-0 oiec_sampfreq	Sampling frequency

13.9.3.5.52 Frame Composer Audio Sample Channel Status Configuration Register 8 (fc_audschnl8)

13.9.3.5.52.1 Offset

Register	Offset
fc_audschnl8	6Fh

13.9.3.5.52.2 Diagram



13.9.3.5.52.3 Fields

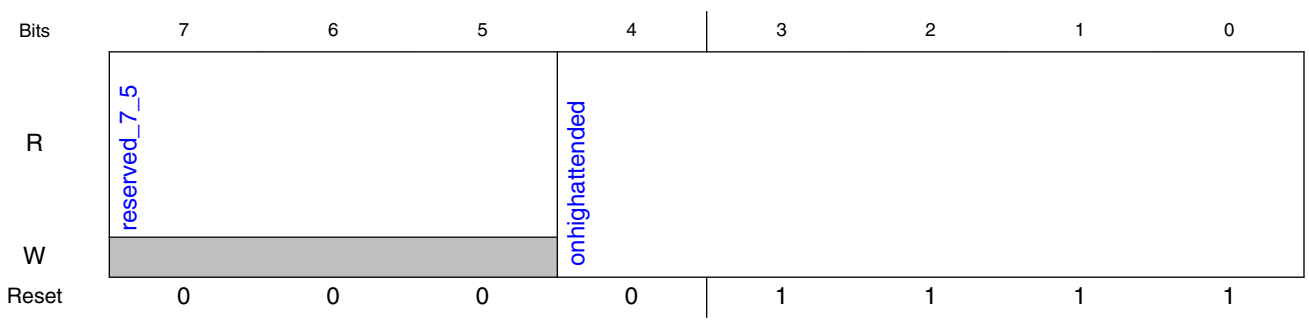
Field	Function
7-4 oiec_origsampfreq	Original sampling frequency
3-0 oiec_wordlength	Word length configuration

13.9.3.5.53 Frame Composer Number of High Priority Packets Attended Configuration Register (fc_ctrlqhigh)

13.9.3.5.53.1 Offset

Register	Offset
fc_ctrlqhigh	73h

13.9.3.5.53.2 Diagram



13.9.3.5.53.3 Fields

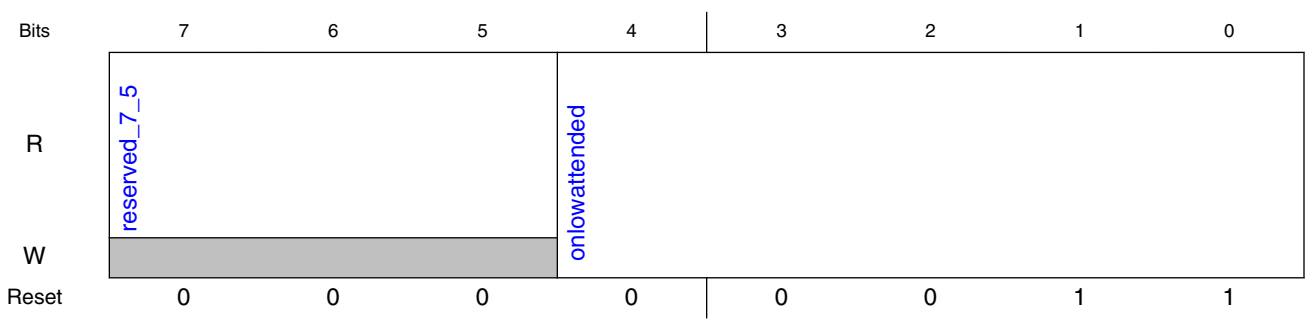
Field	Function
7-5 reserved_7_5	Reserved and read as zero
4-0 onhighattended	Configures the number of high priority packets or audio sample packets consecutively attended before checking low priority queue status. Configures the number of high priority packets or audio sample packets consecutively attended before checking low priority queue status. Valid range is from 5'd1 to 5'd31.

13.9.3.5.54 Frame Composer Number of Low Priority Packets Attended Configuration Register (fc_ctrlqlow)

13.9.3.5.54.1 Offset

Register	Offset
fc_ctrlqlow	74h

13.9.3.5.54.2 Diagram



13.9.3.5.54.3 Fields

Field	Function
7-5 reserved_7_5	Reserved and read as zero
4-0 onlowattended	Configures the number of low priority packets or null packets consecutively attended before checking high priority queue status or audio samples availability. Configures the number of low priority packets or null packets consecutively attended before checking high priority queue status or audio samples availability. Valid range is from 5'd1 to 5'd31.

13.9.3.5.55 Frame Composer ACP Packet Type Configuration Register 0 (fc_acp0)

13.9.3.5.55.1 Offset

Register	Offset
fc_acp0	75h

13.9.3.5.55.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	acptype							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.55.3 Fields

Field	Function
7-0 acptype	Configures the ACP packet type.

13.9.3.5.56 Frame Composer ACP Packet Body Configuration Register 16 (fc_acp16)

13.9.3.5.56.1 Offset

Register	Offset
fc_acp16	82h

13.9.3.5.56.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_acp16							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.56.3 Fields

Field	Function
7-0 fc_acp16	Frame Composer ACP Packet Body Configuration Register 16

13.9.3.5.57 Frame Composer ACP Packet Body Configuration Register 15 (fc_acp15)

13.9.3.5.57.1 Offset

Register	Offset
fc_acp15	83h

13.9.3.5.57.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_acp15							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.57.3 Fields

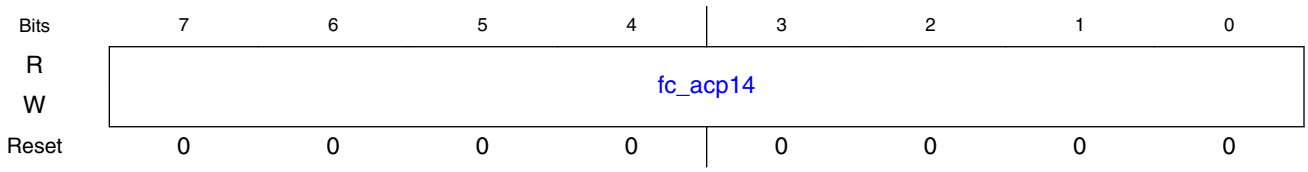
Field	Function
7-0 fc_acp15	Frame Composer ACP Packet Body Configuration Register 15

13.9.3.5.58 Frame Composer ACP Packet Body Configuration Register 14 (fc_acp14)

13.9.3.5.58.1 Offset

Register	Offset
fc_acp14	84h

13.9.3.5.58.2 Diagram



13.9.3.5.58.3 Fields

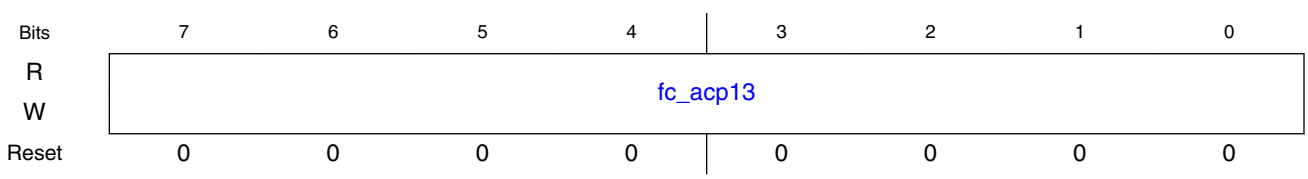
Field	Function
7-0 fc_acp14	Frame Composer ACP Packet Body Configuration Register 14

13.9.3.5.59 Frame Composer ACP Packet Body Configuration Register 13 (fc_acp13)

13.9.3.5.59.1 Offset

Register	Offset
fc_acp13	85h

13.9.3.5.59.2 Diagram



13.9.3.5.59.3 Fields

Field	Function
7-0 fc_acp13	Frame Composer ACP Packet Body Configuration Register 13

13.9.3.5.60 Frame Composer ACP Packet Body Configuration Register 12 (fc_acp12)

13.9.3.5.60.1 Offset

Register	Offset
fc_acp12	86h

13.9.3.5.60.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_acp12							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.60.3 Fields

Field	Function
7-0 fc_acp12	Frame Composer ACP Packet Body Configuration Register 12

13.9.3.5.61 Frame Composer ACP Packet Body Configuration Register 11 (fc_acp11)

13.9.3.5.61.1 Offset

Register	Offset
fc_acp11	87h

13.9.3.5.61.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_acp11							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.61.3 Fields

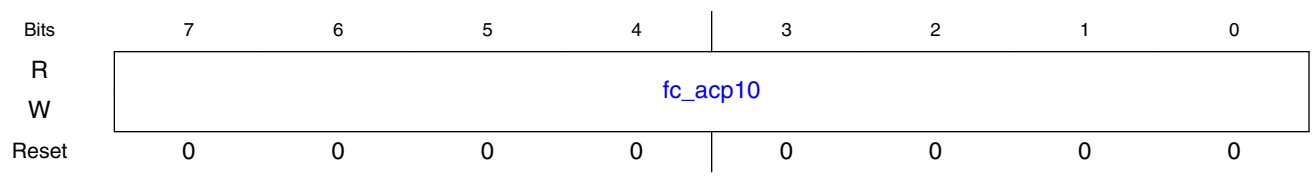
Field	Function
7-0 fc_acp11	Frame Composer ACP Packet Body Configuration Register 11

13.9.3.5.62 Frame Composer ACP Packet Body Configuration Register 10 (fc_acp10)

13.9.3.5.62.1 Offset

Register	Offset
fc_acp10	88h

13.9.3.5.62.2 Diagram



13.9.3.5.62.3 Fields

Field	Function
7-0 fc_acp10	Frame Composer ACP Packet Body Configuration Register 10

13.9.3.5.63 Frame Composer ACP Packet Body Configuration Register 9 (fc_acp9)

13.9.3.5.63.1 Offset

Register	Offset
fc_acp9	89h

13.9.3.5.63.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_acp9							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.63.3 Fields

Field	Function
7-0 fc_acp9	Frame Composer ACP Packet Body Configuration Register 9

13.9.3.5.64 Frame Composer ACP Packet Body Configuration Register 8 (fc_acp8)

13.9.3.5.64.1 Offset

Register	Offset
fc_acp8	8Ah

13.9.3.5.64.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_acp8							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.64.3 Fields

Field	Function
7-0 fc_acp8	Frame Composer ACP Packet Body Configuration Register 8

13.9.3.5.65 Frame Composer ACP Packet Body Configuration Register 7 (fc_acp7)

13.9.3.5.65.1 Offset

Register	Offset
fc_acp7	8Bh

13.9.3.5.65.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_acp7							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.65.3 Fields

Field	Function
7-0 fc_acp7	Frame Composer ACP Packet Body Configuration Register 7

13.9.3.5.66 Frame Composer ACP Packet Body Configuration Register 6 (fc_acp6)

13.9.3.5.66.1 Offset

Register	Offset
fc_acp6	8Ch

13.9.3.5.66.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_acp6							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.66.3 Fields

Field	Function
7-0 fc_acp6	Frame Composer ACP Packet Body Configuration Register 6

13.9.3.5.67 Frame Composer ACP Packet Body Configuration Register 5 (fc_acp5)

13.9.3.5.67.1 Offset

Register	Offset
fc_acp5	8Dh

13.9.3.5.67.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_acp5							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.67.3 Fields

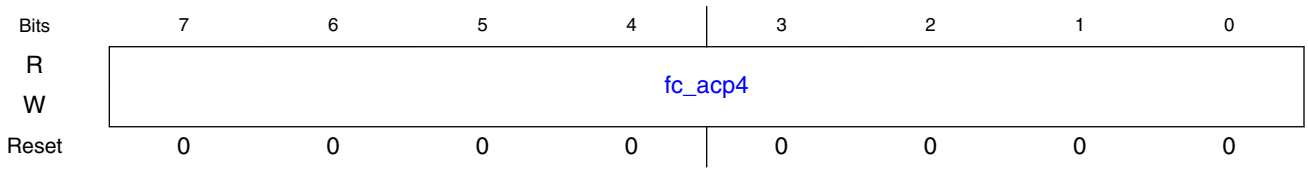
Field	Function
7-0 fc_acp5	Frame Composer ACP Packet Body Configuration Register 5

13.9.3.5.68 Frame Composer ACP Packet Body Configuration Register 4 (fc_acp4)

13.9.3.5.68.1 Offset

Register	Offset
fc_acp4	8Eh

13.9.3.5.68.2 Diagram



13.9.3.5.68.3 Fields

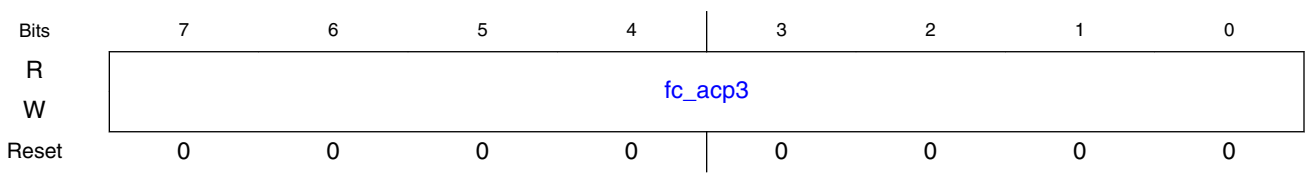
Field	Function
7-0 fc_acp4	Frame Composer ACP Packet Body Configuration Register 4

13.9.3.5.69 Frame Composer ACP Packet Body Configuration Register 3 (fc_acp3)

13.9.3.5.69.1 Offset

Register	Offset
fc_acp3	8Fh

13.9.3.5.69.2 Diagram



13.9.3.5.69.3 Fields

Field	Function
7-0 fc_acp3	Frame Composer ACP Packet Body Configuration Register 3

13.9.3.5.70 Frame Composer ACP Packet Body Configuration Register 2 (fc_acp2)

13.9.3.5.70.1 Offset

Register	Offset
fc_acp2	90h

13.9.3.5.70.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_acp2							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.70.3 Fields

Field	Function
7-0 fc_acp2	Frame Composer ACP Packet Body Configuration Register 2

13.9.3.5.71 Frame Composer ACP Packet Body Configuration Register 1 (fc_acp1)

13.9.3.5.71.1 Offset

Register	Offset
fc_acp1	91h

13.9.3.5.71.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_acp1							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.71.3 Fields

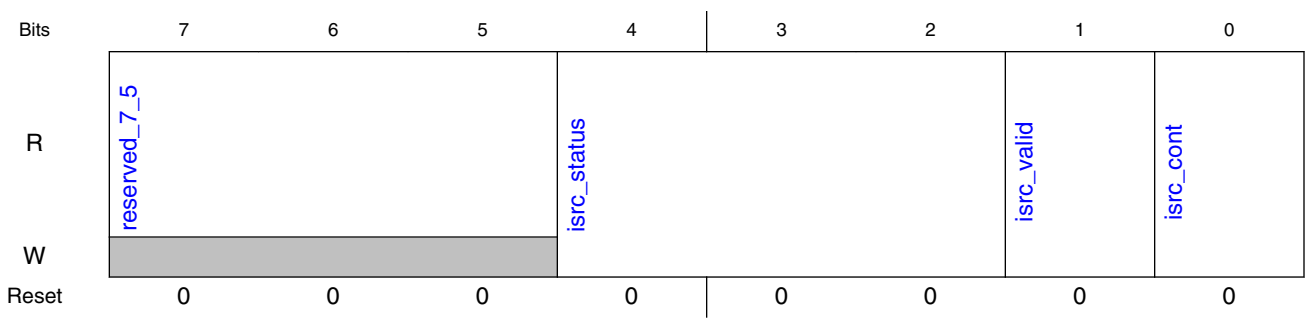
Field	Function
7-0 fc_acp1	Frame Composer ACP Packet Body Configuration Register 1

13.9.3.5.72 Frame Composer ISRC1 Packet Status, Valid, and Continue Configuration Register (fc_iscr1_0)

13.9.3.5.72.1 Offset

Register	Offset
fc_iscr1_0	92h

13.9.3.5.72.2 Diagram



13.9.3.5.72.3 Fields

Field	Function
7-5 reserved_7_5	Reserved and read as zero
4-2 isrc_status	ISRC1 Status signal
1 isrc_valid	ISRC1 Valid control signal
0 isrc_cont	ISRC1 Indication of packet continuation (ISRC2 will be transmitted)

13.9.3.5.73 Frame Composer ISRC1 Packet Body Register 16 (fc_iscr1_16)

13.9.3.5.73.1 Offset

Register	Offset
fc_iscr1_16	93h

13.9.3.5.73.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr1_16							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.73.3 Fields

Field	Function
7-0 fc_iscr1_16	Frame Composer ISRC1 Packet Body Register 16; configures ISRC1 packet body of the ISRC1 packet

13.9.3.5.74 Frame Composer ISRC1 Packet Body Register 15 (fc_iscr1_15)

13.9.3.5.74.1 Offset

Register	Offset
fc_iscr1_15	94h

13.9.3.5.74.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr1_15							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.74.3 Fields

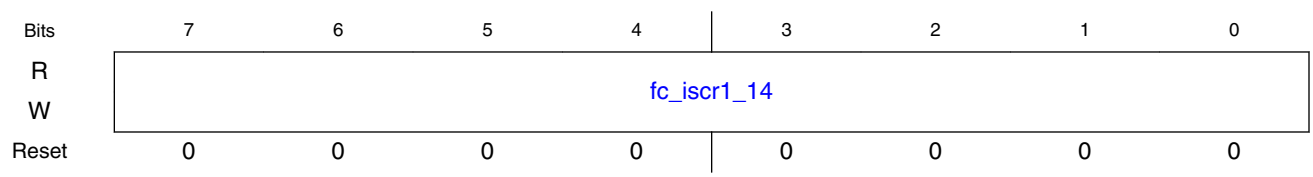
Field	Function
7-0 fc_iscr1_15	Frame Composer ISRC1 Packet Body Register 15

13.9.3.5.75 Frame Composer ISRC1 Packet Body Register 14 (fc_iscr1_14)

13.9.3.5.75.1 Offset

Register	Offset
fc_iscr1_14	95h

13.9.3.5.75.2 Diagram



13.9.3.5.75.3 Fields

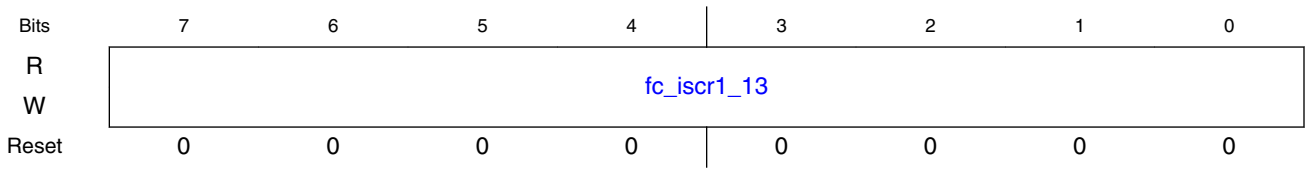
Field	Function
7-0 fc_iscr1_14	Frame Composer ISRC1 Packet Body Register 14

13.9.3.5.76 Frame Composer ISRC1 Packet Body Register 13 (fc_iscr1_13)

13.9.3.5.76.1 Offset

Register	Offset
fc_iscr1_13	96h

13.9.3.5.76.2 Diagram



13.9.3.5.76.3 Fields

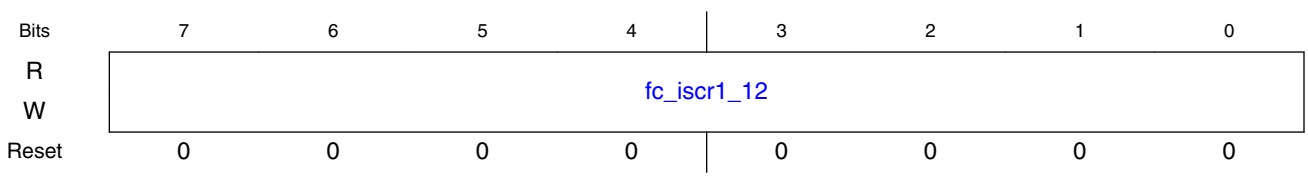
Field	Function
7-0 fc_iscr1_13	Frame Composer ISRC1 Packet Body Register 13

13.9.3.5.77 Frame Composer ISRC1 Packet Body Register 12 (fc_iscr1_12)

13.9.3.5.77.1 Offset

Register	Offset
fc_iscr1_12	97h

13.9.3.5.77.2 Diagram



13.9.3.5.77.3 Fields

Field	Function
7-0 fc_iscr1_12	Frame Composer ISRC1 Packet Body Register 12

13.9.3.5.78 Frame Composer ISRC1 Packet Body Register 11 (fc_iscr1_11)

13.9.3.5.78.1 Offset

Register	Offset
fc_iscr1_11	98h

13.9.3.5.78.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr1_11							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.78.3 Fields

Field	Function
7-0 fc_iscr1_11	Frame Composer ISRC1 Packet Body Register 11

13.9.3.5.79 Frame Composer ISRC1 Packet Body Register 10 (fc_iscr1_10)

13.9.3.5.79.1 Offset

Register	Offset
fc_iscr1_10	99h

13.9.3.5.79.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr1_10							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.79.3 Fields

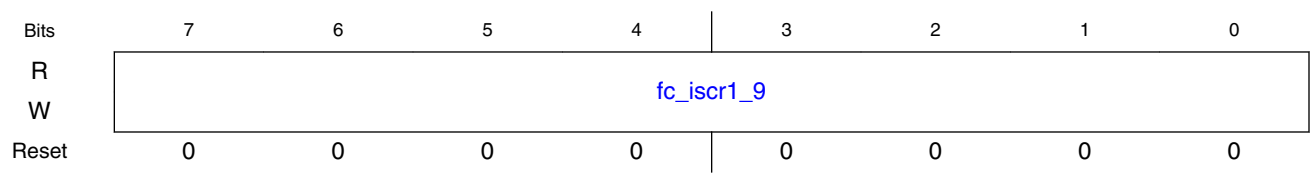
Field	Function
7-0 fc_iscr1_10	Frame Composer ISRC1 Packet Body Register 10

13.9.3.5.80 Frame Composer ISRC1 Packet Body Register 9 (fc_iscr1_9)

13.9.3.5.80.1 Offset

Register	Offset
fc_iscr1_9	9Ah

13.9.3.5.80.2 Diagram



13.9.3.5.80.3 Fields

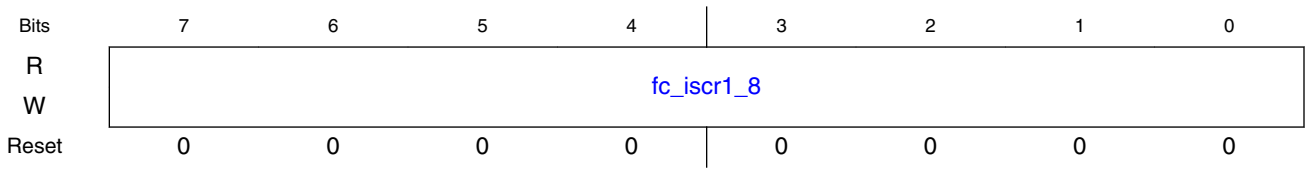
Field	Function
7-0 fc_iscr1_9	Frame Composer ISRC1 Packet Body Register 9

13.9.3.5.81 Frame Composer ISRC1 Packet Body Register 8 (fc_iscr1_8)

13.9.3.5.81.1 Offset

Register	Offset
fc_iscr1_8	9Bh

13.9.3.5.81.2 Diagram



13.9.3.5.81.3 Fields

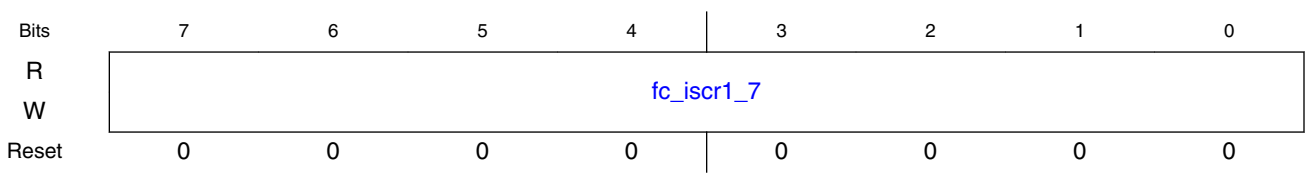
Field	Function
7-0 fc_iscr1_8	Frame Composer ISRC1 Packet Body Register 8

13.9.3.5.82 Frame Composer ISRC1 Packet Body Register 7 (fc_iscr1_7)

13.9.3.5.82.1 Offset

Register	Offset
fc_iscr1_7	9Ch

13.9.3.5.82.2 Diagram



13.9.3.5.82.3 Fields

Field	Function
7-0 fc_iscr1_7	Frame Composer ISRC1 Packet Body Register 7

13.9.3.5.83 Frame Composer ISRC1 Packet Body Register 6 (fc_iscr1_6)

13.9.3.5.83.1 Offset

Register	Offset
fc_iscr1_6	9Dh

13.9.3.5.83.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr1_6							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.83.3 Fields

Field	Function
7-0 fc_iscr1_6	Frame Composer ISRC1 Packet Body Register 6

13.9.3.5.84 Frame Composer ISRC1 Packet Body Register 5 (fc_iscr1_5)

13.9.3.5.84.1 Offset

Register	Offset
fc_iscr1_5	9Eh

13.9.3.5.84.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr1_5							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.84.3 Fields

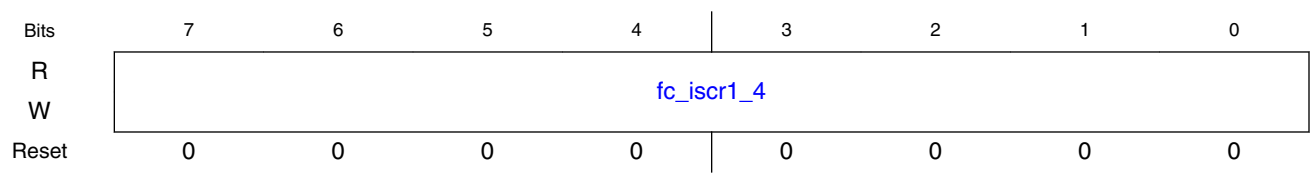
Field	Function
7-0 fc_iscr1_5	Frame Composer ISRC1 Packet Body Register 5

13.9.3.5.85 Frame Composer ISRC1 Packet Body Register 4 (fc_iscr1_4)

13.9.3.5.85.1 Offset

Register	Offset
fc_iscr1_4	9Fh

13.9.3.5.85.2 Diagram



13.9.3.5.85.3 Fields

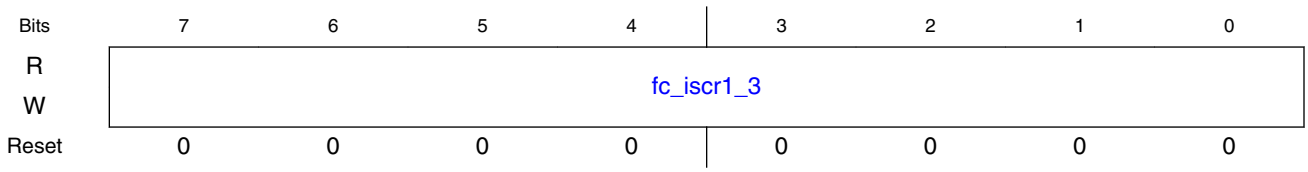
Field	Function
7-0 fc_iscr1_4	Frame Composer ISRC1 Packet Body Register 4

13.9.3.5.86 Frame Composer ISRC1 Packet Body Register 3 (fc_iscr1_3)

13.9.3.5.86.1 Offset

Register	Offset
fc_iscr1_3	A0h

13.9.3.5.86.2 Diagram



13.9.3.5.86.3 Fields

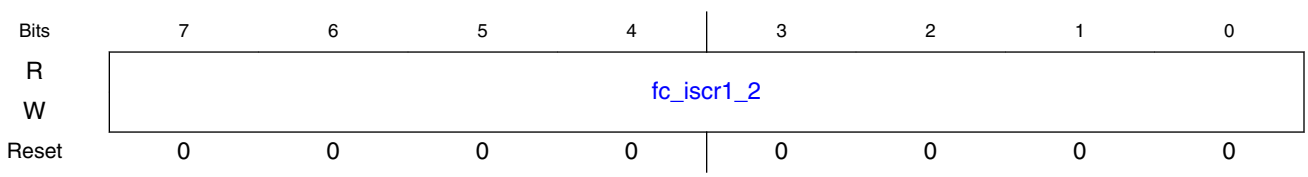
Field	Function
7-0 fc_iscr1_3	Frame Composer ISRC1 Packet Body Register 3

13.9.3.5.87 Frame Composer ISRC1 Packet Body Register 2 (fc_iscr1_2)

13.9.3.5.87.1 Offset

Register	Offset
fc_iscr1_2	A1h

13.9.3.5.87.2 Diagram



13.9.3.5.87.3 Fields

Field	Function
7-0 fc_iscr1_2	Frame Composer ISRC1 Packet Body Register 2

13.9.3.5.88 Frame Composer ISRC1 Packet Body Register 1 (fc_iscr1_1)

13.9.3.5.88.1 Offset

Register	Offset
fc_iscr1_1	A2h

13.9.3.5.88.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr1_1							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.88.3 Fields

Field	Function
7-0 fc_iscr1_1	Frame Composer ISRC1 Packet Body Register 1

13.9.3.5.89 Frame Composer ISRC2 Packet Body Register 15 (fc_iscr2_15)

13.9.3.5.89.1 Offset

Register	Offset
fc_iscr2_15	A3h

13.9.3.5.89.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr2_15							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.89.3 Fields

Field	Function
7-0 fc_iscr2_15	Frame Composer ISRC2 Packet Body Register 15; configures the ISRC2 packet body of the ISRC2 packet

13.9.3.5.90 Frame Composer ISRC2 Packet Body Register 14 (fc_iscr2_14)

13.9.3.5.90.1 Offset

Register	Offset
fc_iscr2_14	A4h

13.9.3.5.90.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr2_14							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.90.3 Fields

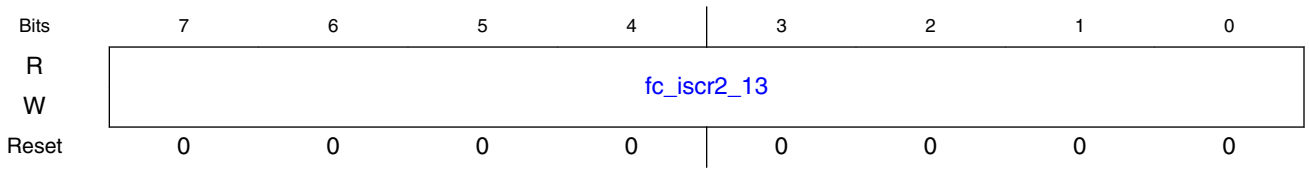
Field	Function
7-0 fc_iscr2_14	Frame Composer ISRC2 Packet Body Register 14

13.9.3.5.91 Frame Composer ISRC2 Packet Body Register 13 (fc_iscr2_13)

13.9.3.5.91.1 Offset

Register	Offset
fc_iscr2_13	A5h

13.9.3.5.91.2 Diagram



13.9.3.5.91.3 Fields

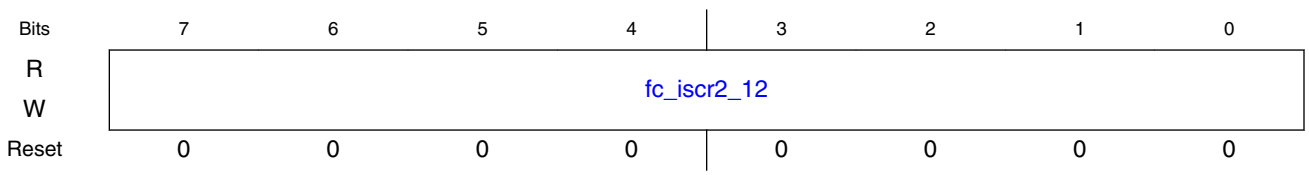
Field	Function
7-0 fc_iscr2_13	Frame Composer ISRC2 Packet Body Register 13

13.9.3.5.92 Frame Composer ISRC2 Packet Body Register 12 (fc_iscr2_12)

13.9.3.5.92.1 Offset

Register	Offset
fc_iscr2_12	A6h

13.9.3.5.92.2 Diagram



13.9.3.5.92.3 Fields

Field	Function
7-0 fc_iscr2_12	Frame Composer ISRC2 Packet Body Register 12

13.9.3.5.93 Frame Composer ISRC2 Packet Body Register 11 (fc_iscr2_11)

13.9.3.5.93.1 Offset

Register	Offset
fc_iscr2_11	A7h

13.9.3.5.93.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr2_11							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.93.3 Fields

Field	Function
7-0 fc_iscr2_11	Frame Composer ISRC2 Packet Body Register 11

13.9.3.5.94 Frame Composer ISRC2 Packet Body Register 10 (fc_iscr2_10)

13.9.3.5.94.1 Offset

Register	Offset
fc_iscr2_10	A8h

13.9.3.5.94.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr2_10							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.94.3 Fields

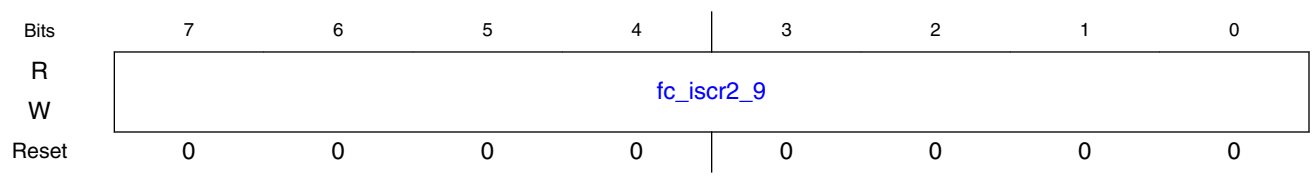
Field	Function
7-0 fc_iscr2_10	Frame Composer ISRC2 Packet Body Register 10

13.9.3.5.95 Frame Composer ISRC2 Packet Body Register 9 (fc_iscr2_9)

13.9.3.5.95.1 Offset

Register	Offset
fc_iscr2_9	A9h

13.9.3.5.95.2 Diagram



13.9.3.5.95.3 Fields

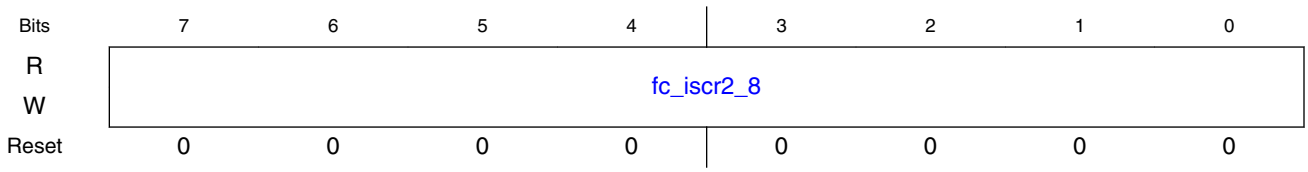
Field	Function
7-0 fc_iscr2_9	Frame Composer ISRC2 Packet Body Register 9

13.9.3.5.96 Frame Composer ISRC2 Packet Body Register 8 (fc_iscr2_8)

13.9.3.5.96.1 Offset

Register	Offset
fc_iscr2_8	AAh

13.9.3.5.96.2 Diagram



13.9.3.5.96.3 Fields

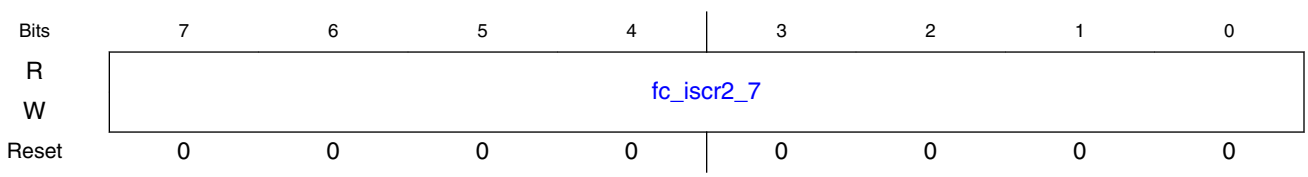
Field	Function
7-0 fc_iscr2_8	Frame Composer ISRC2 Packet Body Register 8

13.9.3.5.97 Frame Composer ISRC2 Packet Body Register 7 (fc_iscr2_7)

13.9.3.5.97.1 Offset

Register	Offset
fc_iscr2_7	ABh

13.9.3.5.97.2 Diagram



13.9.3.5.97.3 Fields

Field	Function
7-0 fc_iscr2_7	Frame Composer ISRC2 Packet Body Register 7

13.9.3.5.98 Frame Composer ISRC2 Packet Body Register 6 (fc_iscr2_6)

13.9.3.5.98.1 Offset

Register	Offset
fc_iscr2_6	ACh

13.9.3.5.98.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr2_6							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.98.3 Fields

Field	Function
7-0 fc_iscr2_6	Frame Composer ISRC2 Packet Body Register 6

13.9.3.5.99 Frame Composer ISRC2 Packet Body Register 5 (fc_iscr2_5)

13.9.3.5.99.1 Offset

Register	Offset
fc_iscr2_5	ADh

13.9.3.5.99.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr2_5							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.99.3 Fields

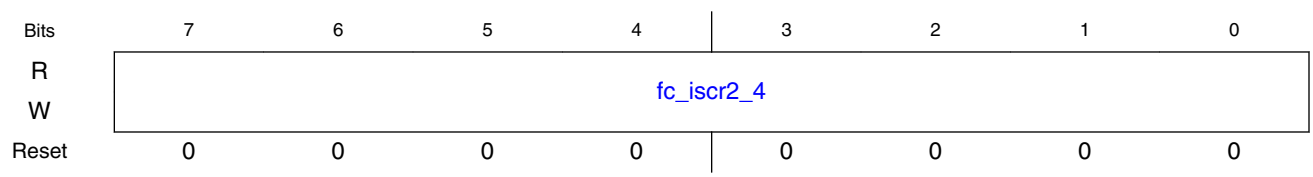
Field	Function
7-0 fc_iscr2_5	Frame Composer ISRC2 Packet Body Register 5

13.9.3.5.100 Frame Composer ISRC2 Packet Body Register 4 (fc_iscr2_4)

13.9.3.5.100.1 Offset

Register	Offset
fc_iscr2_4	AEh

13.9.3.5.100.2 Diagram



13.9.3.5.100.3 Fields

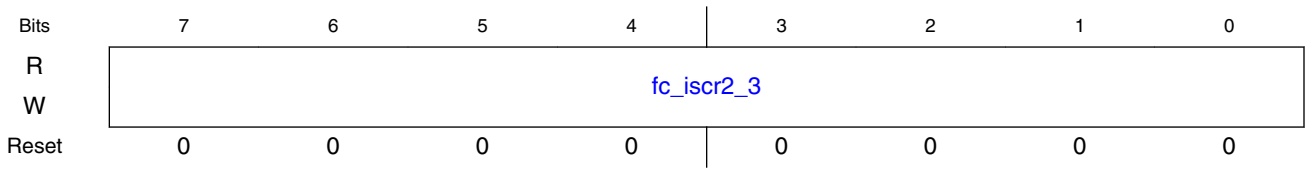
Field	Function
7-0 fc_iscr2_4	Frame Composer ISRC2 Packet Body Register 4

13.9.3.5.101 Frame Composer ISRC2 Packet Body Register 3 (fc_iscr2_3)

13.9.3.5.101.1 Offset

Register	Offset
fc_iscr2_3	AFh

13.9.3.5.101.2 Diagram



13.9.3.5.101.3 Fields

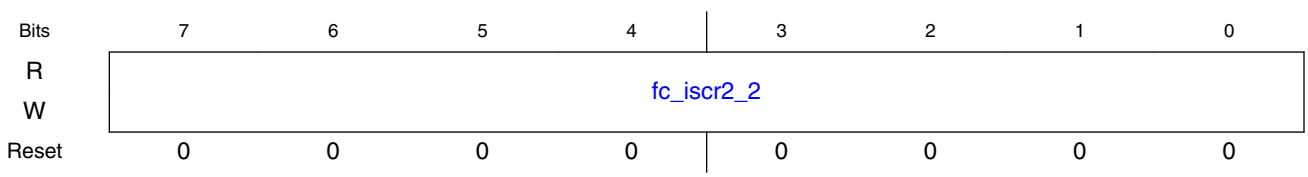
Field	Function
7-0 fc_iscr2_3	Frame Composer ISRC2 Packet Body Register 3

13.9.3.5.102 Frame Composer ISRC2 Packet Body Register 2 (fc_iscr2_2)

13.9.3.5.102.1 Offset

Register	Offset
fc_iscr2_2	B0h

13.9.3.5.102.2 Diagram



13.9.3.5.102.3 Fields

Field	Function
7-0 fc_iscr2_2	Frame Composer ISRC2 Packet Body Register 2

13.9.3.5.103 Frame Composer ISRC2 Packet Body Register 1 (fc_iscr2_1)

13.9.3.5.103.1 Offset

Register	Offset
fc_iscr2_1	B1h

13.9.3.5.103.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr2_1							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.103.3 Fields

Field	Function
7-0 fc_iscr2_1	Frame Composer ISRC2 Packet Body Register 1

13.9.3.5.104 Frame Composer ISRC2 Packet Body Register 0 (fc_iscr2_0)

13.9.3.5.104.1 Offset

Register	Offset
fc_iscr2_0	B2h

13.9.3.5.104.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_iscr2_0							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.104.3 Fields

Field	Function
7-0 fc_iscr2_0	Frame Composer ISRC2 Packet Body Register 0

13.9.3.5.105 Frame Composer Data Island Auto Packet Scheduling Register 0 Configures the Frame Composer RDRB(1)/ Manual(0) data island packet insertion for SPD, VSD, ISRC2, ISRC1 and ACP packets. (fc_datauto0)

13.9.3.5.105.1 Offset

Register	Offset
fc_datauto0	B3h

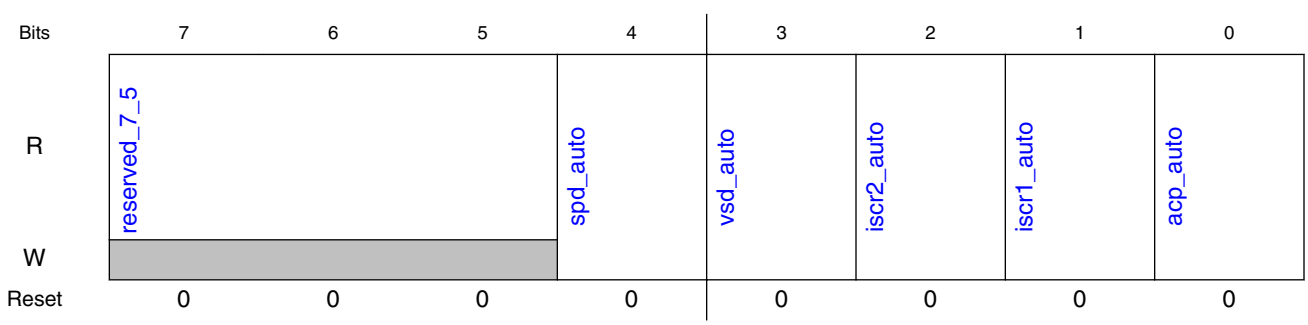
13.9.3.5.105.2 Function

Frame Composer Data Island Auto Packet Scheduling Register 0

Configures the Frame Composer RDRB(1)/Manual(0) data island packet insertion for SPD, VSD, ISRC2, ISRC1 and ACP packets. On RDRB mode the described packet scheduling is controlled by registers FC_DATAUTO1 and FC_DATAUTO2, while in Manual mode register FC_DATMAN requests to FC the insertion of the requested packet.

Note: When activating auto packet scheduling, there is a possibility of sending multiple vendor-specific infoframes and violating CEA-861F.

13.9.3.5.105.3 Diagram



13.9.3.5.105.4 Fields

Field	Function
7-5 reserved_7_5	Reserved and read as zero
4 spd_auto	Enables SPD automatic packet scheduling
3 vsd_auto	Enables VSD automatic packet scheduling
2 iscr2_auto	Enables ISRC2 automatic packet scheduling
1 iscr1_auto	Enables ISRC1 automatic packet scheduling
0 acp_auto	Enables ACP automatic packet scheduling

13.9.3.5.106 Frame Composer Data Island Auto Packet Scheduling Register 1 Configures the Frame Composer (FC) RDRB frame interpolation for SPD, VSD, ISRC2, ISRC1 and ACP packet insertion on data island when FC is on RDRB mode for the listed packets. (fc_datauto1)

13.9.3.5.106.1 Offset

Register	Offset
fc_datauto1	B4h

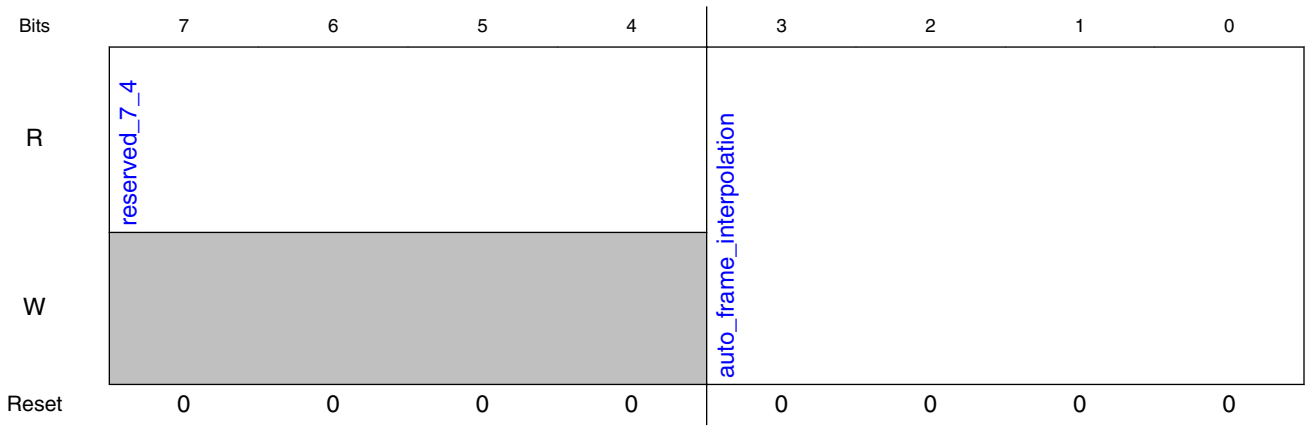
13.9.3.5.106.2 Function

Frame Composer Data Island Auto Packet Scheduling Register 1

Configures the Frame Composer (FC) RDRB frame interpolation for SPD, VSD, ISRC2, ISRC1 and ACP packet insertion on data island when FC is on RDRB mode for the listed packets.

Note: When activating auto packet scheduling, there is a possibility of sending multiple vendor-specific infoframes and violating CEA-861F.

13.9.3.5.106.3 Diagram



13.9.3.5.106.4 Fields

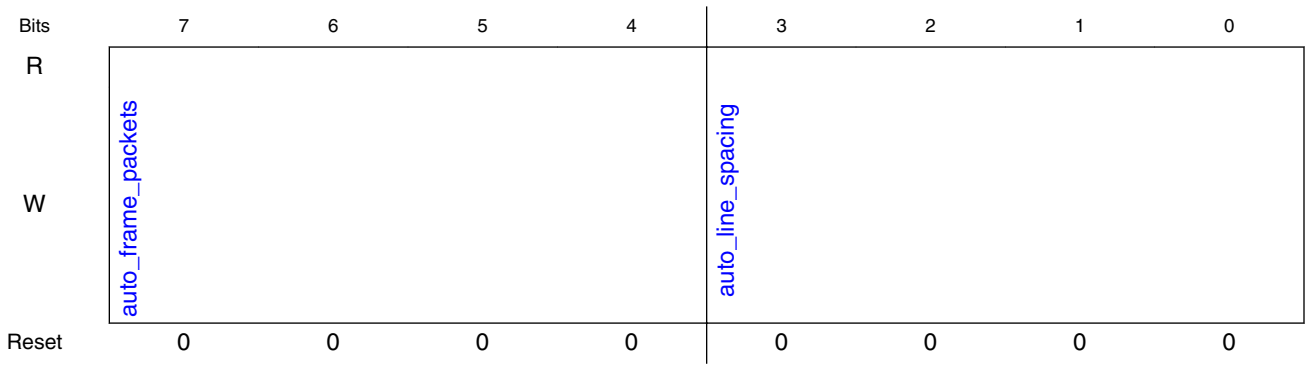
Field	Function
7-4 reserved_7_4	Reserved and read as zero
3-0 auto_frame_interpolation	Packet frame interpolation for automatic packet scheduling

13.9.3.5.107 Frame Composer Data Island Auto packet scheduling Register 2 Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for SPD, VSD, ISRC2, ISRC1 and ACP packet insertion on data island when FC is on RDRB mode for the listed packets. (fc_datauto2)

13.9.3.5.107.1 Offset

Register	Offset
fc_datauto2	B5h

13.9.3.5.107.2 Diagram



13.9.3.5.107.3 Fields

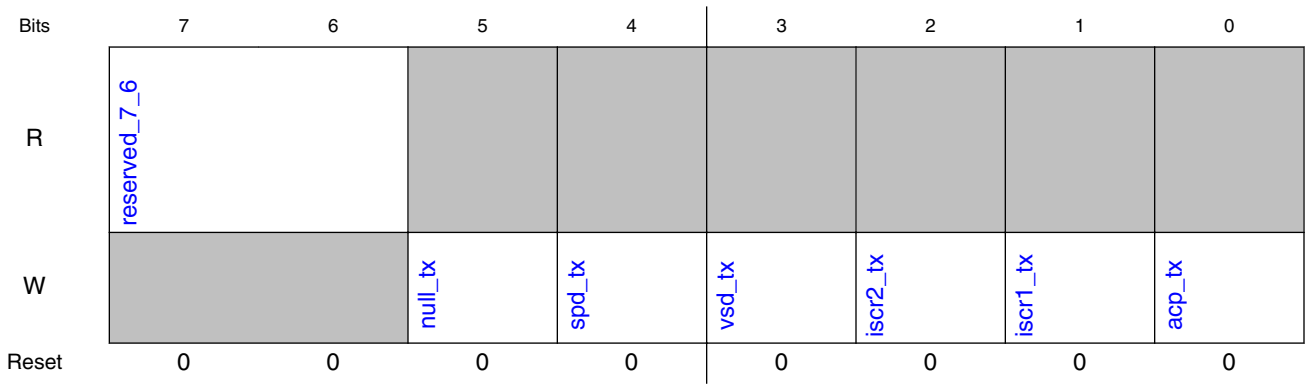
Field	Function
7-4 auto_frame_packets	Packets per frame, for automatic packet scheduling
3-0 auto_line_spacing	Packets line spacing, for automatic packet scheduling

13.9.3.5.108 Frame Composer Data Island Manual Packet Request Register Requests to the Frame Composer the data island packet insertion for NULL, SPD, VSD, ISRC2, ISRC1 and ACP packets when FC_DATAUTO0 bit is in manual mode for the packet requested. (fc_datman)

13.9.3.5.108.1 Offset

Register	Offset
fc_datman	B6h

13.9.3.5.108.2 Diagram



13.9.3.5.108.3 Fields

Field	Function
7-6 reserved_7_6	Reserved and read as zero
5 null_tx	Null packet
4 spd_tx	SPD packet
3 vsd_tx	VSD packet
2 iscr2_tx	ISRC2 packet
1 iscr1_tx	ISRC1 packet
0 acp_tx	ACP packet

13.9.3.5.109 Frame Composer Data Island Auto Packet Scheduling Register 3 Configures the Frame Composer Automatic(1)/ RDRB(0) data island packet insertion for AVI, GCP, AUDI and ACR packets. (fc_datauto3)

13.9.3.5.109.1 Offset

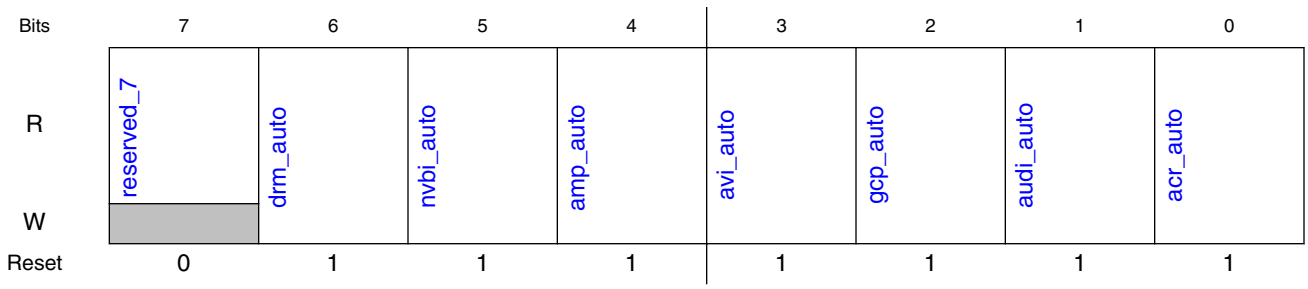
Register	Offset
fc_datauto3	B7h

13.9.3.5.109.2 Function

Frame Composer Data Island Auto Packet Scheduling Register 3

Configures the Frame Composer Automatic(1)/RDRB(0) data island packet insertion for AVI, GCP, AUDI and ACR packets. In Automatic mode, the packet is inserted on Vblanking when first line with active Vsync appears.

13.9.3.5.109.3 Diagram



13.9.3.5.109.4 Fields

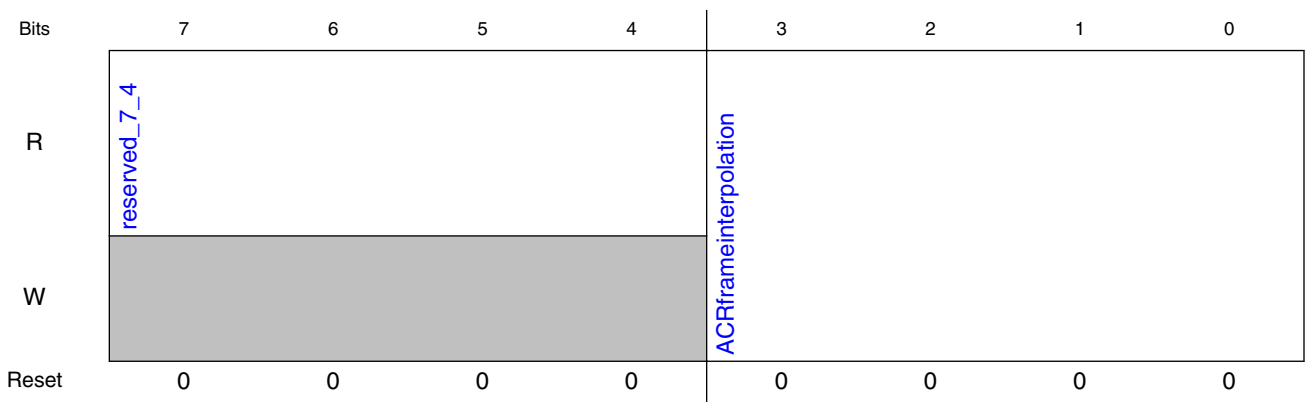
Field	Function
7 reserved_7	Reserved and read as zero
6 drm_auto	Enables DRM packet insertion
5 nvbi_auto	Enables NTSC VBI packet insertion
4 amp_auto	Enables AMP packet insertion
3 avi_auto	Enables AVI packet insertion
2 gcp_auto	Enables GCP packet insertion
1 audi_auto	Enables AUDI packet insertion
0 acr_auto	Enables ACR packet insertion

13.9.3.5.110 Frame Composer Round Robin ACR Packet Insertion Register 0 Configures the Frame Composer (FC) RDRB frame interpolation for ACR packet insertion on data island when FC is on RDRB mode for this packet. (fc_rdrb0)

13.9.3.5.110.1 Offset

Register	Offset
fc_rdrb0	B8h

13.9.3.5.110.2 Diagram



13.9.3.5.110.3 Fields

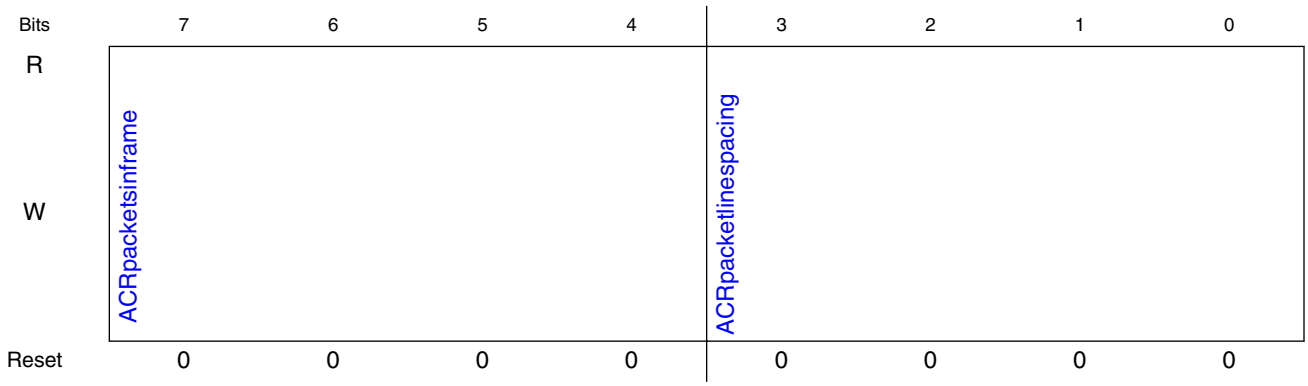
Field	Function
7-4 reserved_7_4	Reserved and read as zero
3-0 ACRframeinterpolation	ACR Frame interpolation

13.9.3.5.111 Frame Composer Round Robin ACR Packet Insertion Register 1 Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the ACR packet insertion on data island when FC is on RDRB mode this packet. (fc_rdrb1)

13.9.3.5.111.1 Offset

Register	Offset
fc_rdrb1	B9h

13.9.3.5.111.2 Diagram



13.9.3.5.111.3 Fields

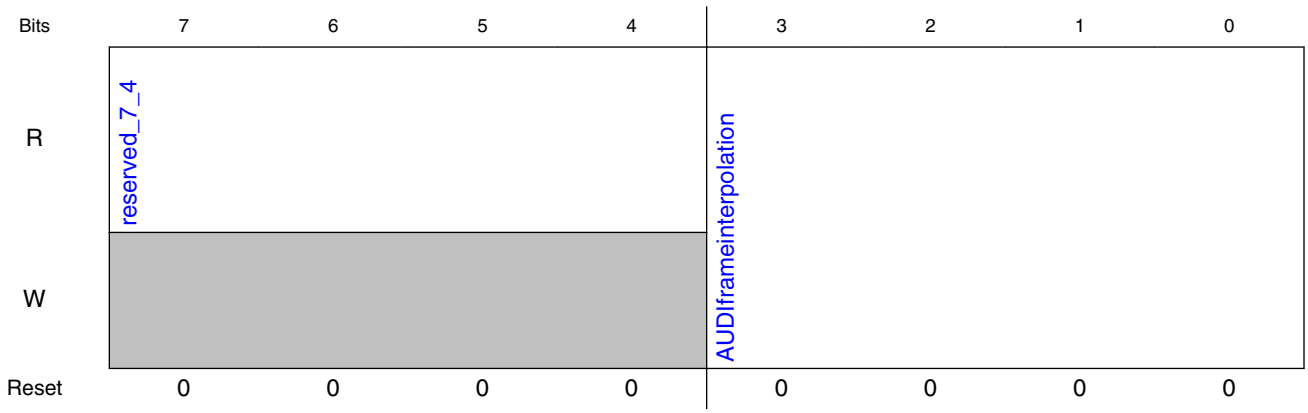
Field	Function
7-4 ACRpacketsinframe	ACR packets in frame
3-0 ACRpacketlinespacing	ACR packet line spacing

13.9.3.5.112 Frame Composer Round Robin AUDI Packet Insertion Register 2 Configures the Frame Composer (FC) RDRB frame interpolation for AUDI packet insertion on data island when FC is on RDRB mode for this packet. (fc_rdrb2)

13.9.3.5.112.1 Offset

Register	Offset
fc_rdrb2	BAh

13.9.3.5.112.2 Diagram



13.9.3.5.112.3 Fields

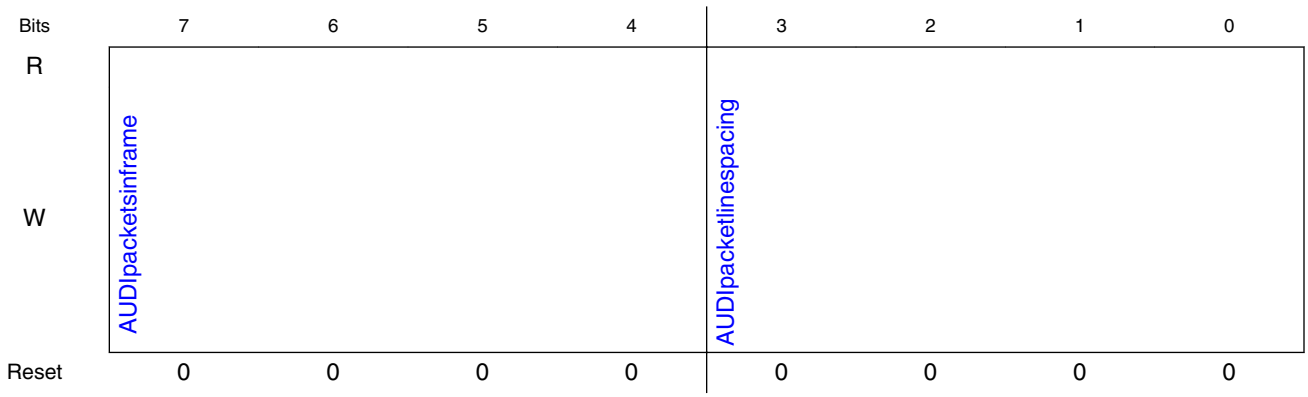
Field	Function
7-4 reserved_7_4	Reserved and read as zero
3-0 AUDIframeinterpolation	Audio frame interpolation

13.9.3.5.113 Frame Composer Round Robin AUDI Packet Insertion Register 3 Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the AUDI packet insertion on data island when FC is on RDRB mode this packet. (fc_rdrb3)

13.9.3.5.113.1 Offset

Register	Offset
fc_rdrb3	BBh

13.9.3.5.113.2 Diagram



13.9.3.5.113.3 Fields

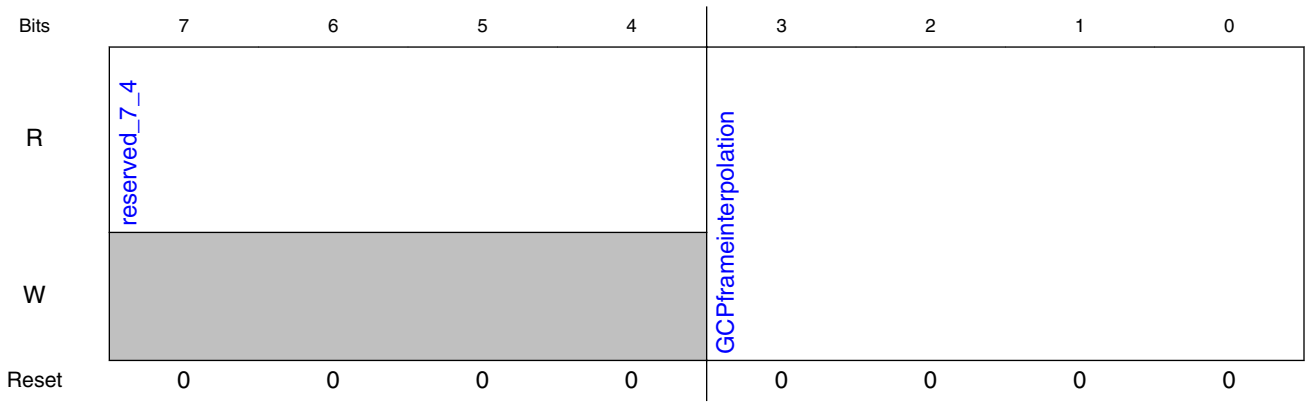
Field	Function
7-4 AUDIpacketsinframe	Audio packets per frame
3-0 AUDIpacketlinespacing	Audio packets line spacing

13.9.3.5.114 Frame Composer Round Robin GCP Packet Insertion Register 4 Configures the Frame Composer (FC) RDRB frame interpolation for GCP packet insertion on data island when FC is on RDRB mode for this packet. (fc_rdrb4)

13.9.3.5.114.1 Offset

Register	Offset
fc_rdrb4	BCh

13.9.3.5.114.2 Diagram



13.9.3.5.114.3 Fields

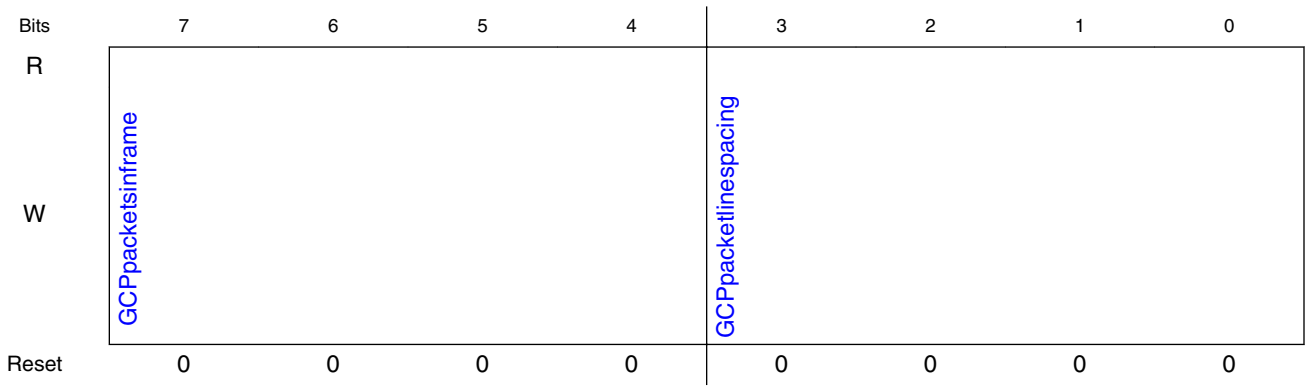
Field	Function
7-4 reserved_7_4	Reserved and read as zero
3-0 GCPframeinterpolation	Frames interpolated between GCP packets

13.9.3.5.115 Frame Composer Round Robin GCP Packet Insertion Register 5 Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the GCP packet insertion on data island when FC is on RDRB mode this packet. (fc_rdrb5)

13.9.3.5.115.1 Offset

Register	Offset
fc_rdrb5	BDh

13.9.3.5.115.2 Diagram



13.9.3.5.115.3 Fields

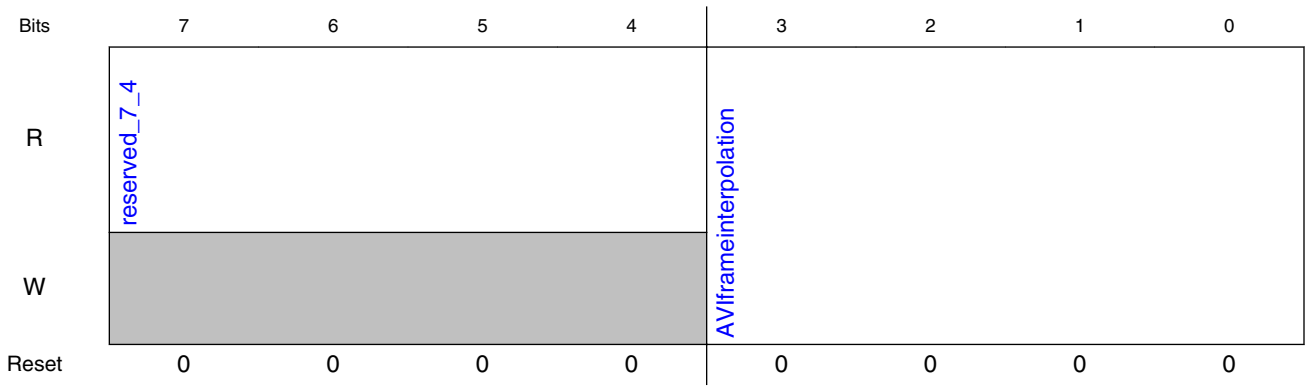
Field	Function
7-4 GCPpacketsinframe	GCP packets per frame
3-0 GCPpacketlinespacing	GCP packets line spacing

13.9.3.5.116 Frame Composer Round Robin AVI Packet Insertion Register 6 Configures the Frame Composer (FC) RDRB frame interpolation for AVI packet insertion on data island when FC is on RDRB mode for this packet. (fc_rdrb6)

13.9.3.5.116.1 Offset

Register	Offset
fc_rdrb6	BEh

13.9.3.5.116.2 Diagram



13.9.3.5.116.3 Fields

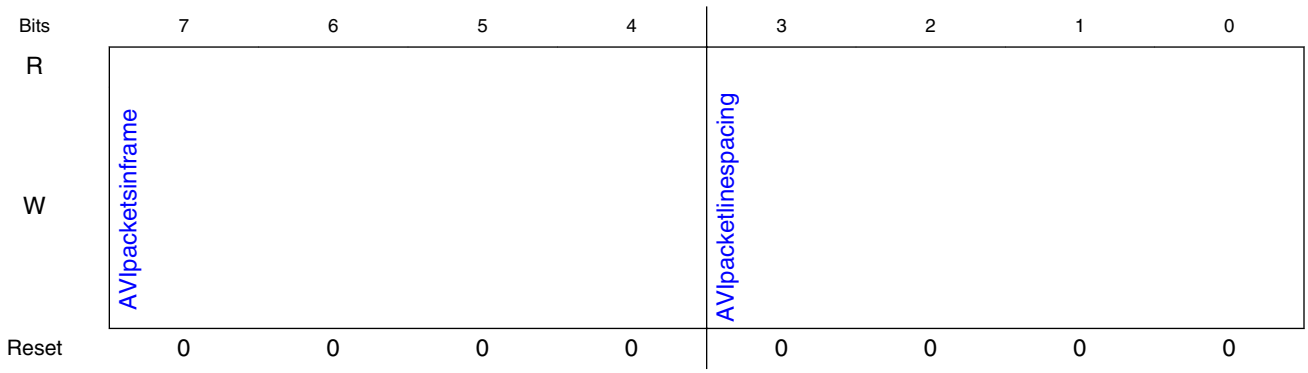
Field	Function
7-4 reserved_7_4	Reserved and read as zero
3-0 AVIframeinterpolation	Frames interpolated between AVI packets

13.9.3.5.117 Frame Composer Round Robin AVI Packet Insertion Register 7 Configures the Frame Composer (FC) RDRB line interpolation and number of packets in frame for the AVI packet insertion on data island when FC is on RDRB mode this packet. (fc_rdrb7)

13.9.3.5.117.1 Offset

Register	Offset
fc_rdrb7	BFh

13.9.3.5.117.2 Diagram



13.9.3.5.117.3 Fields

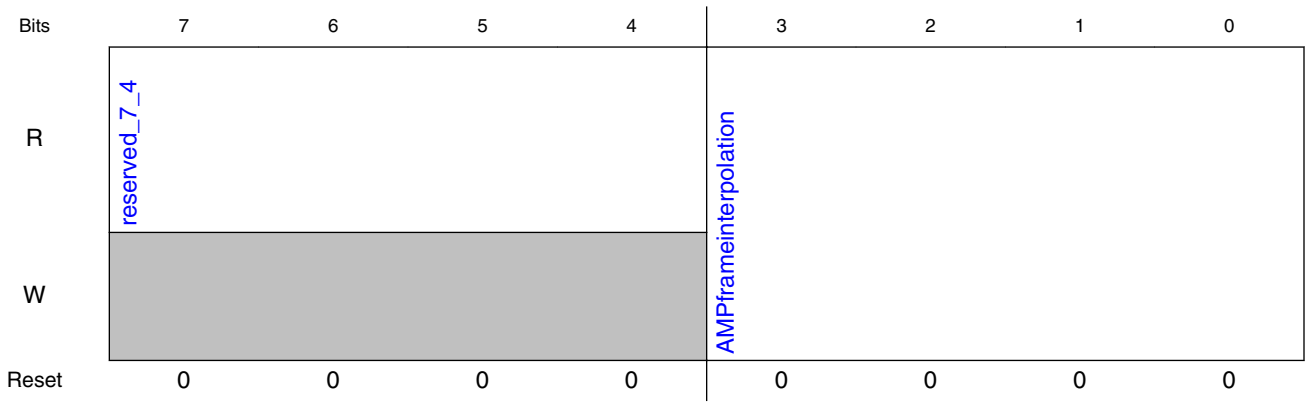
Field	Function
7-4 AVIpacketsinframe	AVI packets per frame
3-0 AVIpacketlinespacing	AVI packets line spacing

13.9.3.5.118 Frame Composer Round Robin AMP Packet Insertion Register 8 (fc_rdrb8)

13.9.3.5.118.1 Offset

Register	Offset
fc_rdrb8	C0h

13.9.3.5.118.2 Diagram



13.9.3.5.118.3 Fields

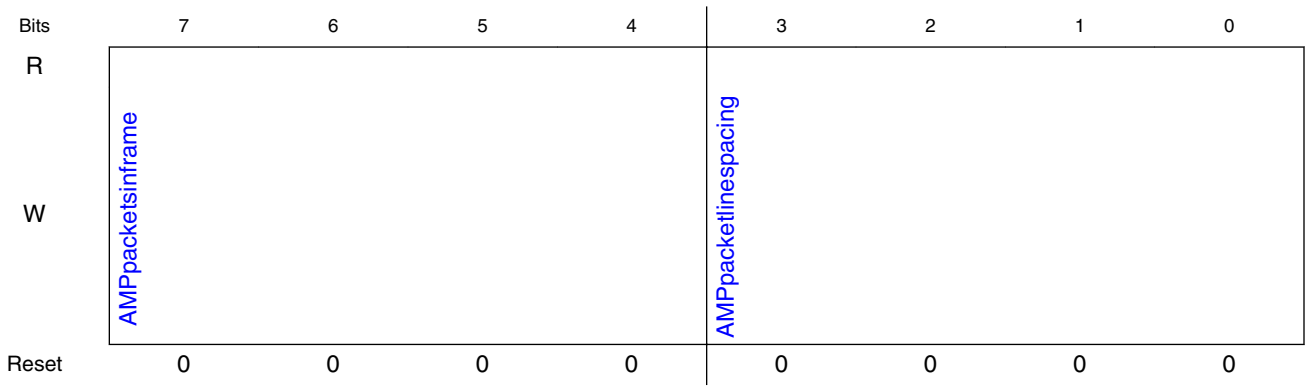
Field	Function
7-4 reserved_7_4	Reserved and read as zero
3-0 AMPframeinterpolation	AMP frame interpolation

13.9.3.5.119 Frame Composer Round Robin AMP Packet Insertion Register 9 (fc_rdrb9)

13.9.3.5.119.1 Offset

Register	Offset
fc_rdrb9	C1h

13.9.3.5.119.2 Diagram



13.9.3.5.119.3 Fields

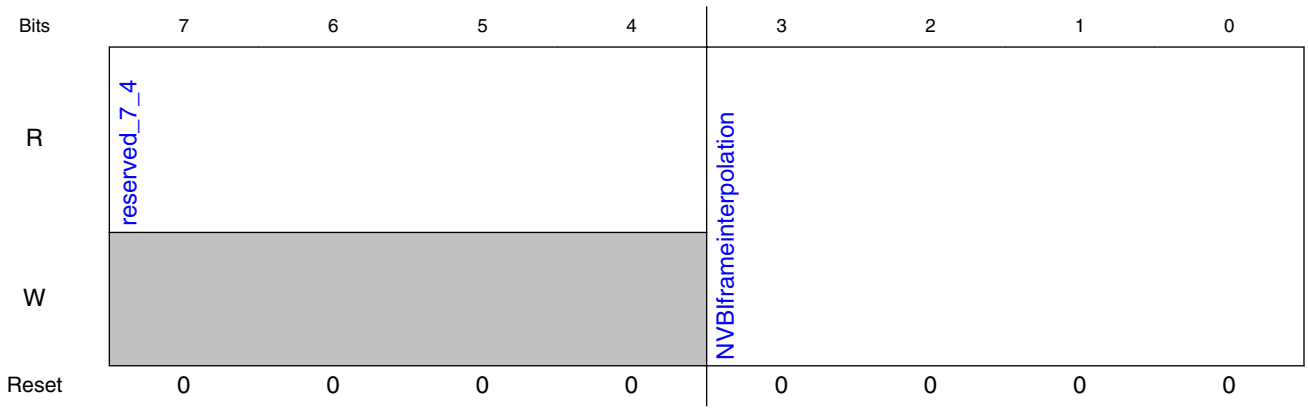
Field	Function
7-4 AMPpacketsinframe	AMP packets per frame
3-0 AMPpacketlinespacing	AMP packets line spacing

13.9.3.5.120 Frame Composer Round Robin NTSC VBI Packet Insertion Register 10 (fc_rdrb10)

13.9.3.5.120.1 Offset

Register	Offset
fc_rdrb10	C2h

13.9.3.5.120.2 Diagram



13.9.3.5.120.3 Fields

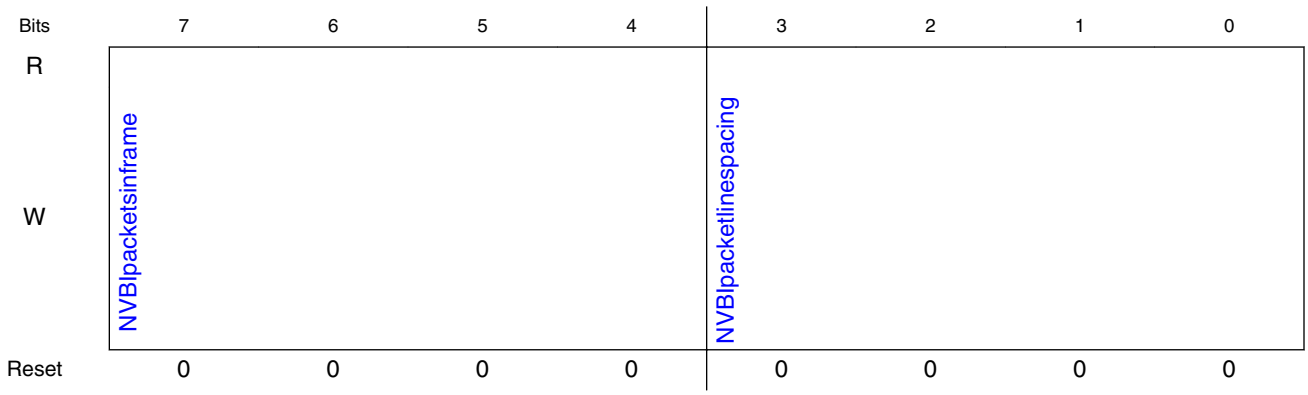
Field	Function
7-4 reserved_7_4	Reserved and read as zero
3-0 NVBIframeinterpolation	NTSC VBI frame interpolation

13.9.3.5.121 Frame Composer Round Robin NTSC VBI Packet Insertion Register 11 (fc_rdrb11)

13.9.3.5.121.1 Offset

Register	Offset
fc_rdrb11	C3h

13.9.3.5.121.2 Diagram



13.9.3.5.121.3 Fields

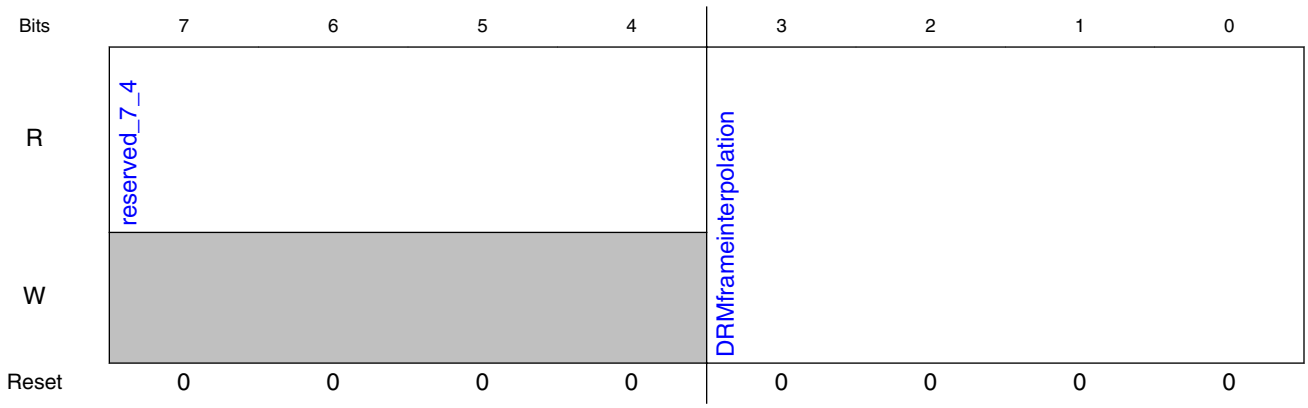
Field	Function
7-4 NVBIpacketsinframe	NTSC VBI packets per frame
3-0 NVBIpacketlinespacing	NTSC VBI packets line spacing

13.9.3.5.122 Frame Composer Round Robin DRM Packet Insertion Register 12 (fc_rdrb12)

13.9.3.5.122.1 Offset

Register	Offset
fc_rdrb12	C4h

13.9.3.5.122.2 Diagram



13.9.3.5.122.3 Fields

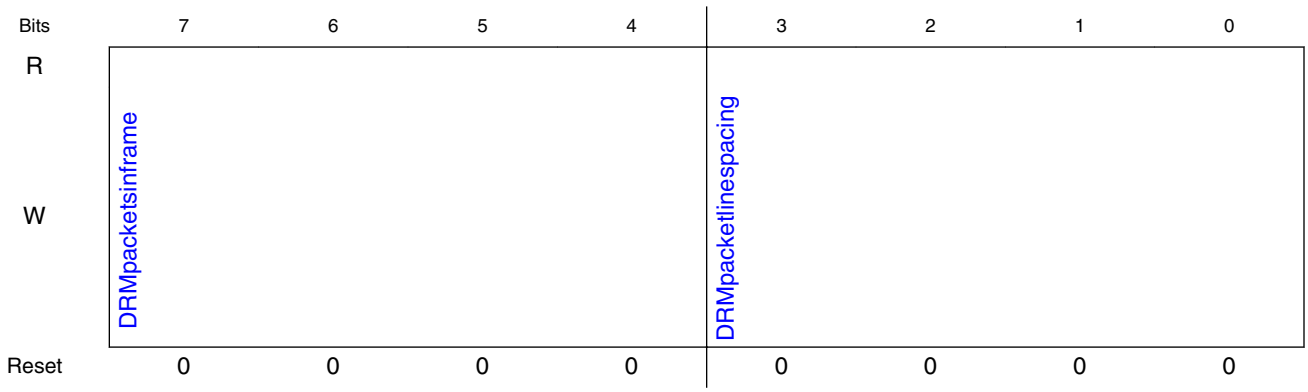
Field	Function
7-4 reserved_7_4	Reserved and read as zero
3-0 DRMframeinterpolation	DRM frame interpolation

13.9.3.5.123 Frame Composer Round Robin DRM Packet Insertion Register 13 (fc_rdrb13)

13.9.3.5.123.1 Offset

Register	Offset
fc_rdrb13	C5h

13.9.3.5.123.2 Diagram



13.9.3.5.123.3 Fields

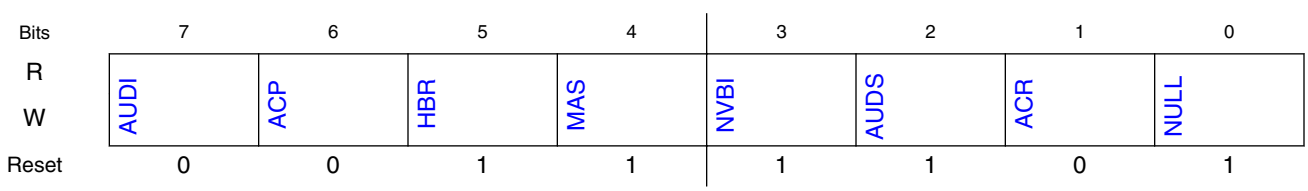
Field	Function
7-4 DRMpacketsinframe	DRM packets per frame
3-0 DRMpacketlinespacing	DRM packets line spacing

13.9.3.5.124 Frame Composer Packet Interrupt Mask Register 0 (fc_mask0)

13.9.3.5.124.1 Offset

Register	Offset
fc_mask0	D2h

13.9.3.5.124.2 Diagram



13.9.3.5.124.3 Fields

Field	Function
7 AUDI	Mask bit for FC_INT0. Mask bit for FC_INT0.AUDI interrupt bit
6 ACP	Mask bit for FC_INT0. Mask bit for FC_INT0.ACP interrupt bit
5 HBR	Mask bit for FC_INT0. Mask bit for FC_INT0.HBR interrupt bit
4 MAS	Mask bit for FC_INT0. Mask bit for FC_INT0.MAS interrupt bit.
3 NVBI	Mask bit for FC_INT0. Mask bit for FC_INT0.NVBI interrupt bit.
2 AUDS	Mask bit for FC_INT0. Mask bit for FC_INT0.AUDS interrupt bit
1 ACR	Mask bit for FC_INT0. Mask bit for FC_INT0.ACR interrupt bit
0 NULL	Mask bit for FC_INT0. Mask bit for FC_INT0.NULL interrupt bit

13.9.3.5.125 Frame Composer Packet Interrupt Mask Register 1 (fc_mask1)

13.9.3.5.125.1 Offset

Register	Offset
fc_mask1	D6h

13.9.3.5.125.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	GMD	ISCR1	ISCR2	VSD	SPD	AMP	AVI	GCP
Reset	0	0	0	0	0	1	0	0

13.9.3.5.125.3 Fields

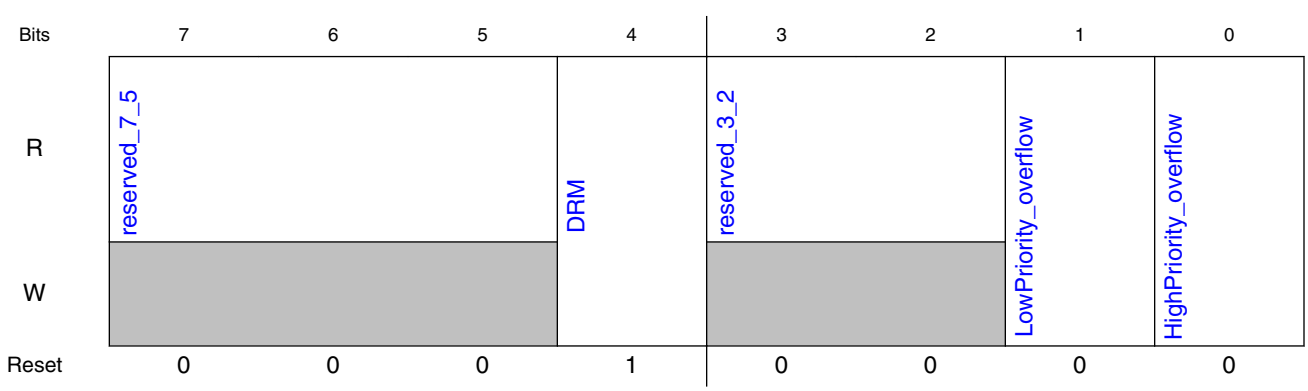
Field	Function
7 GMD	Mask bit for FC_INT1. Mask bit for FC_INT1.GMD interrupt bit
6 ISCR1	Mask bit for FC_INT1. Mask bit for FC_INT1.ISRC1 interrupt bit
5 ISCR2	Mask bit for FC_INT1. Mask bit for FC_INT1.ISRC2 interrupt bit
4 VSD	Mask bit for FC_INT1. Mask bit for FC_INT1.VSD interrupt bit
3 SPD	Mask bit for FC_INT1. Mask bit for FC_INT1.SPD interrupt bit
2 AMP	Mask bit for FC_INT1. Mask bit for FC_INT1.AMP interrupt bit.
1 AVI	Mask bit for FC_INT1. Mask bit for FC_INT1.AVI interrupt bit
0 GCP	Mask bit for FC_INT1. Mask bit for FC_INT1.GCP interrupt bit

13.9.3.5.126 Frame Composer High/Low Priority Overflow and DRM Interrupt Mask Register 2 (fc_mask2)

13.9.3.5.126.1 Offset

Register	Offset
fc_mask2	DAh

13.9.3.5.126.2 Diagram



13.9.3.5.126.3 Fields

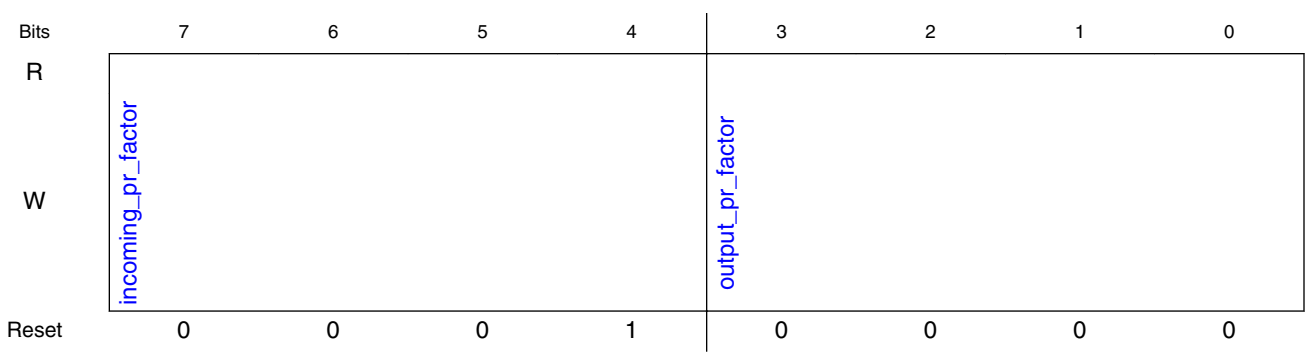
Field	Function
7-5 reserved_7_5	Reserved and read as zero
4 DRM	Mask bit for FC_INT2. Mask bit for FC_INT2.DRM interrupt bit.
3-2 reserved_3_2	Reserved and read as zero
1 LowPriority_overflow	Mask bit for FC_INT2. Mask bit for FC_INT2.LowPriority_overflow interrupt bit
0 HighPriority_overflow	Mask bit for FC_INT2. Mask bit for FC_INT2.HighPriority_overflow interrupt bit

13.9.3.5.127 Frame Composer Pixel Repetition Configuration Register (fc_prconf)

13.9.3.5.127.1 Offset

Register	Offset
fc_prconf	E0h

13.9.3.5.127.2 Diagram



13.9.3.5.127.3 Fields

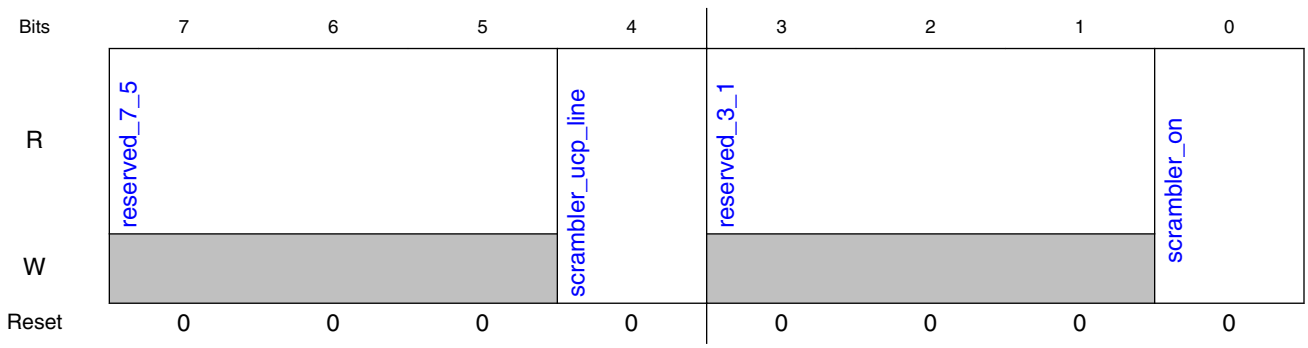
Field	Function
7-4 incoming_pr_factor	Configures the input video pixel repetition. Configures the input video pixel repetition. For CEA modes, this value must be extracted from the CEA specification for the video mode being input. incoming_pr_factor[3:0] 0000b: No action. Not used. 0001b: No pixel repetition (pixel sent only once) 0010b: Pixel sent two times (pixel repeated once) 0011b: Pixel sent three times 0100b: Pixel sent four times 0101b: Pixel sent five times 0110b: Pixel sent six times 0111b: Pixel sent seven times 1000b: Pixel sent eight times 1001b: Pixel sent nine times 1010b: Pixel sent 10 times Other: Reserved. Not used
3-0 output_pr_factor	Configures the video pixel repetition ratio to be sent on the AVI InfoFrame. Configures the video pixel repetition ratio to be sent on the AVI InfoFrame. This value must be valid according to the HDMI specification. The output_pr_factor = incoming_pr_factor * (desired_pr_factor + 1) - 1. output_pr_factor[3:0] 0000b: No action. Not used. 0001b: Pixel sent two times (pixel repeated once) 0010b: Pixel sent three times 0011b: Pixel sent four times 0100b: Pixel sent five times 0101b: Pixel sent six times 0110b: Pixel sent seven times 0111b: Pixel sent eight times 1000b: Pixel sent nine times 1001b: Pixel sent 10 times Other: Reserved. Not used Note: When working in YCbCr 422 video, the actual repetition of the stream is Incoming_pr_factor * (desired_pr_factor + 1). This calculation is done internally in the H13TCTRL and no hardware overflow protection is available. Care must be taken to avoid this result passes the maximum number of 10 pixels repeated because no HDMI support is available for this in the specification and the H13TPHY does not support this higher repetition values.

13.9.3.5.128 Frame Composer Scrambler Control (fc_scrambler_ctrl)

13.9.3.5.128.1 Offset

Register	Offset
fc_scrambler_ctrl	E1h

13.9.3.5.128.2 Diagram



13.9.3.5.128.3 Fields

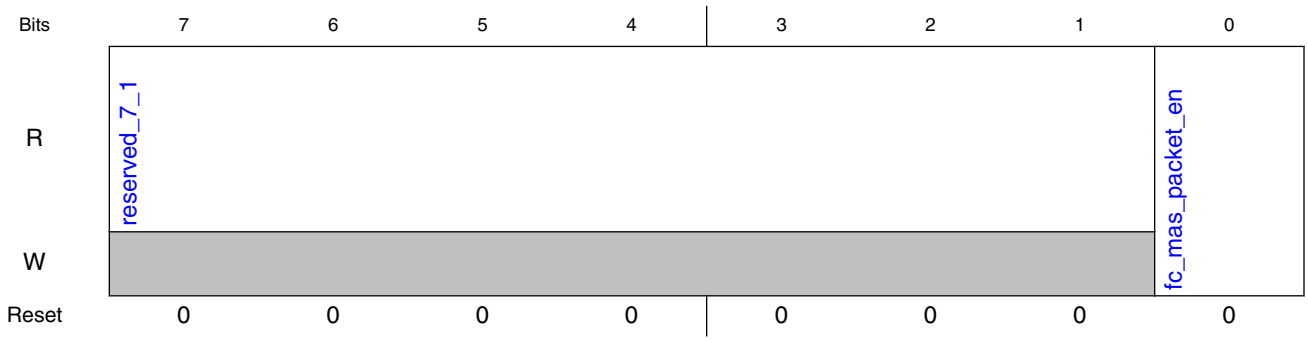
Field	Function
7-5 reserved_7_5	Reserved and read as zero
4 scrambler_ucp_line	Debug register. Debug register. When active (1'b1), the Unscrambled Control Period is generated after each active video line (non-compliant behavior). This is quasi-static field which requires a mc_swrstzreq_1.tmdsswrst_req reset request to be performed after the change of this configuration bit.
3-1 reserved_3_1	Reserved and read as zero
0 scrambler_on	When set (1'b1), this field activates the HDMI 2. When set (1'b1), this field activates the HDMI 2.0 scrambler feature. When disabled (1'b0) the scrambler feature is bypassed, placing DWC_hdmi_tx in HDMI 1.4b compatible mode. This is field can be changed in runtime.

13.9.3.5.129 Frame Composer Multi-Stream Audio Control (fc_multistream_ctrl)

13.9.3.5.129.1 Offset

Register	Offset
fc_multistream_ctrl	E2h

13.9.3.5.129.2 Diagram



13.9.3.5.129.3 Fields

Field	Function
7-1 reserved_7_1	Reserved and read as zero
0 fc_mas_packet_en	This field, when set (1'b1), activates the HDMI 2. This field, when set (1'b1), activates the HDMI 2.0 Multi-Stream support. The audio stream present at the input of the DWC_hdmi_tx controller is transported using Multi-Stream Audio Sample Packets.

13.9.3.5.130 Frame Composer Packet Transmission Control (fc_packet_tx_en)

13.9.3.5.130.1 Offset

Register	Offset
fc_packet_tx_en	E3h

13.9.3.5.130.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	drm_tx_en	nvbi_tx_en	amp_tx_en	aut_tx_en	audi_tx_en	avi_tx_en	gcp_tx_en	acr_tx_en
W								
Reset	0	0	0	1	1	1	1	1

13.9.3.5.130.3 Fields

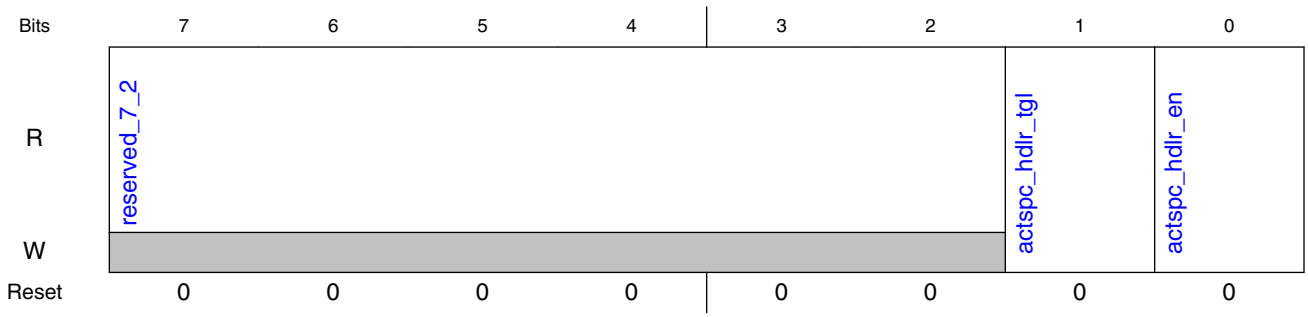
Field	Function
7 drm_tx_en	DRM transmission control 1b: Transmission enabled 0b: Transmission disabled
6 nvbi_tx_en	NTSC VBI transmission control 1b: Transmission enabled 0b: Transmission disabled
5 amp_tx_en	AMP transmission control 1b: Transmission enabled 0b: Transmission disabled
4 aut_tx_en	ACP, SPD, VSIF, ISRC1, and SRC2 packet transmission control 1b: Transmission enabled 0b: Transmission disabled
3 audi_tx_en	AUDI packet transmission control 1b: Transmission enabled 0b: Transmission disabled
2 avi_tx_en	AVI packet transmission control 1b: Transmission enabled 0b: Transmission disabled
1 gcp_tx_en	GCP transmission control 1b: Transmission enabled 0b: Transmission disabled
0 acr_tx_en	ACR packet transmission control 1b: Transmission enabled 0b: Transmission disabled

13.9.3.5.131 Frame Composer Active Space Control (fc_actspc_hdlr_cfg)

13.9.3.5.131.1 Offset

Register	Offset
fc_actspc_hdlr_cfg	E8h

13.9.3.5.131.2 Diagram



13.9.3.5.131.3 Fields

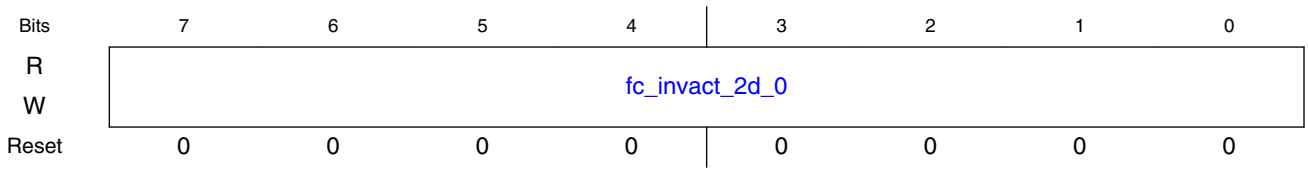
Field	Function
7-2 reserved_7_2	Reserved and read as zero
1 actspc_hdlr_tgl	Active Space handler control 1b: Active space 1 value is different from Active Space 2 value. Active Space handler control 1b: Active space 1 value is different from Active Space 2 value. Refer to Figure 8-4: 3D Structure of the HDMI 1.4b specification. 0b: Active space not oscillating
0 actspc_hdlr_en	Active Space Handler Control 1b: Fixed active space value mode enabled. Active Space Handler Control 1b: Fixed active space value mode enabled. During active space, a fixed value of 0xAA is applied to all TMDS channels. 0b: Fixed active space value mode disabled

13.9.3.5.132 Frame Composer Input Video 2D VActive Pixels Register 0 (fc_invact_2d_0)

13.9.3.5.132.1 Offset

Register	Offset
fc_invact_2d_0	E9h

13.9.3.5.132.2 Diagram



13.9.3.5.132.3 Fields

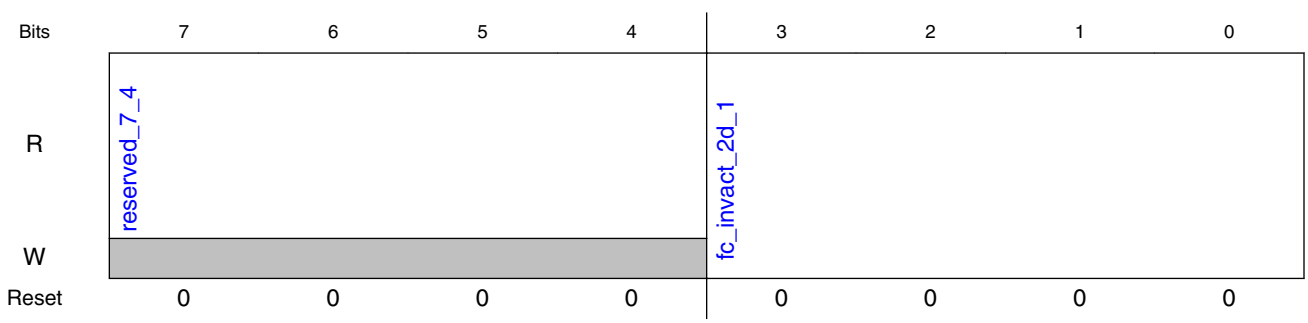
Field	Function
7-0 fc_invact_2d_0	2D Input video vertical active pixel region width. 2D Input video vertical active pixel region width. Number of 2D video vertical active lines [7:0].

13.9.3.5.133 Frame Composer Input Video VActive pixels Register 1 (fc_invact_2d_1)

13.9.3.5.133.1 Offset

Register	Offset
fc_invact_2d_1	EAh

13.9.3.5.133.2 Diagram



13.9.3.5.133.3 Fields

Field	Function
7-4 reserved_7_4	Reserved and read as zero

Table continues on the next page...

Field	Function
3-0	2D Input video vertical active pixel region width.
fc_invact_2d_1	2D Input video vertical active pixel region width. Number of 2D video vertical active lines [11:8].

13.9.3.5.134 Frame Composer GMD Packet Status Register Gamut metadata packet status bit information for no_current_gmd, next_gmd_field, gmd_packet_sequence and current_gamut_seq_num. (fc_gmd_stat)

13.9.3.5.134.1 Offset

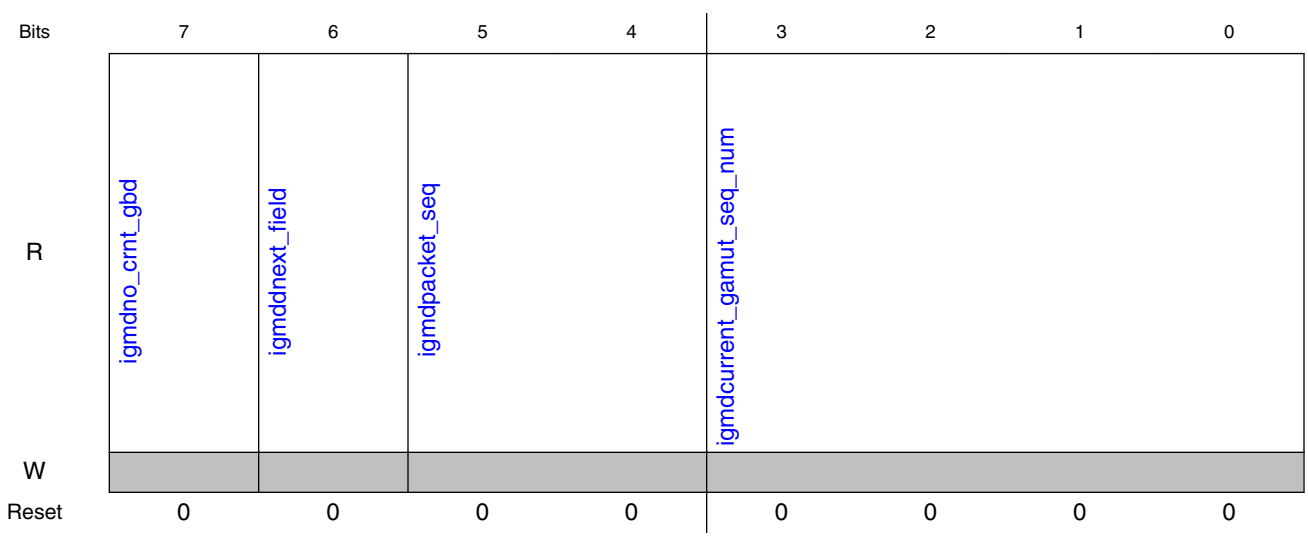
Register	Offset
fc_gmd_stat	100h

13.9.3.5.134.2 Function

Frame Composer GMD Packet Status Register

Gamut metadata packet status bit information for no_current_gmd, next_gmd_field, gmd_packet_sequence and current_gamut_seq_num. For more information, refer to the HDMI 1.4b specification.

13.9.3.5.134.3 Diagram



13.9.3.5.134.4 Fields

Field	Function
7 igmdno_crnt_gbd	Gamut scheduling: No current gamut data
6 igmddnext_field	Gamut scheduling: Gamut Next field
5-4 igmdpacket_seq	Gamut scheduling: Gamut packet sequence
3-0 igmdcurrent_gamut_seq_num	Gamut scheduling: Current Gamut packet sequence number

13.9.3.5.135 Frame Composer GMD Packet Enable Register This register enables Gamut metadata (GMD) packet transmission. (fc_gmd_en)

13.9.3.5.135.1 Offset

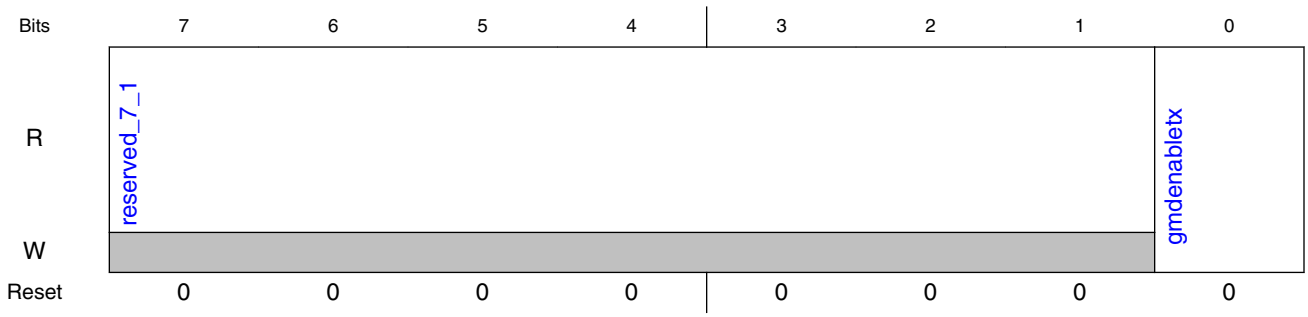
Register	Offset
fc_gmd_en	101h

13.9.3.5.135.2 Function

Frame Composer GMD Packet Enable Register

This register enables Gamut metadata (GMD) packet transmission. Packets are inserted in the incoming frame, starting in the line where active Vsync indication starts. After enable of GMD packets the outgoing packet is sent with no_current_gmd active indication until update GMD request is performed in the controller.

13.9.3.5.135.3 Diagram



13.9.3.5.135.4 Fields

Field	Function
7-1 reserved_7_1	Reserved and read as zero
0 gmdenabletx	Gamut Metadata packet transmission enable (1b)

13.9.3.5.136 Frame Composer GMD Packet Update Register This register performs an GMD packet content update according to the configured packet body (FC_GMD_PB0 to FC_GMD_PB27) and packet header (FC_GMD_HB). (fc_gmd_up)

13.9.3.5.136.1 Offset

Register	Offset
fc_gmd_up	102h

13.9.3.5.136.2 Function

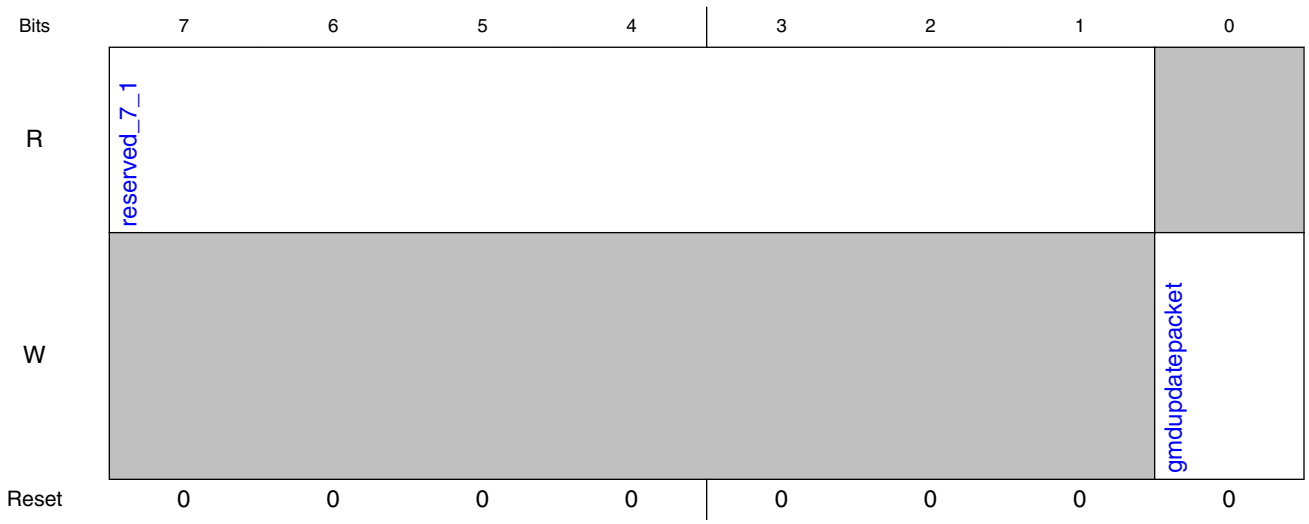
Frame Composer GMD Packet Update Register

This register performs an GMD packet content update according to the configured packet body (FC_GMD_PB0 to FC_GMD_PB27) and packet header (FC_GMD_HB). This active high auto clear register reflects the body and header configurations on the GMD packets sent arbitrating the current_gamut_seq_num, gmd_packet_sequence and

next_gmd_field bits on packet to correctly indicate to source the Gamut change to be performed. After enable GMD packets the first update request is also responsible for deactivating the no_current_gmd indication bit.

Attention packet update request must only be done after correct configuration of GMD packet body and header registers. Correct affected_gamut_seq_num and gmd_profile configuration is user responsibility and must convey with HDMI 1.4b standard gamut rules.

13.9.3.5.136.3 Diagram



13.9.3.5.136.4 Fields

Field	Function
7-1 reserved_7_1	Reserved and read as zero
0 gmdupdatepacket	Gamut Metadata packet update

13.9.3.5.137 Frame Composer GMD Packet Schedule Configuration Register This register configures the number of GMD packets to be inserted per frame (starting always in the line where the active Vsync appears) and the line spacing between the transmitted GMD packets. (fc_gmd_conf)

13.9.3.5.137.1 Offset

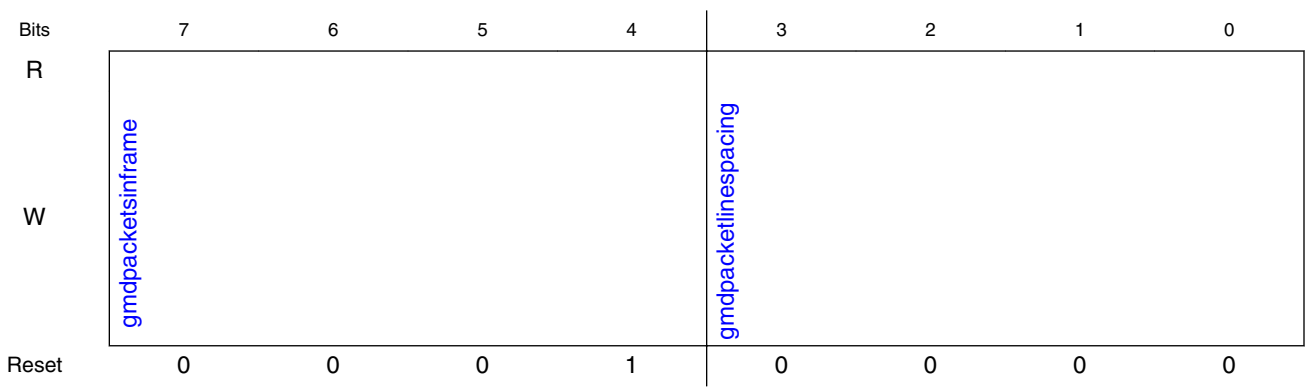
Register	Offset
fc_gmd_conf	103h

13.9.3.5.137.2 Function

Frame Composer GMD Packet Schedule Configuration Register

This register configures the number of GMD packets to be inserted per frame (starting always in the line where the active Vsync appears) and the line spacing between the transmitted GMD packets. Note that for profile P0 (refer to the HDMI 1.4b specification) this register should only indicate one GMD packet to be inserted per video field.

13.9.3.5.137.3 Diagram



13.9.3.5.137.4 Fields

Field	Function
7-4 gmdpacketsinframe	Number of GMD packets per frame or video field (profile P0)
3-0 gmdpacketlinespacing	Number of line spacing between the transmitted GMD packets

13.9.3.5.138 Frame Composer GMD Packet Profile and Gamut Sequence Configuration Register This register configures the GMD

packet header affected_gamut_seq_num and gmd_profile bits. (fc_gmd_hb)

13.9.3.5.138.1 Offset

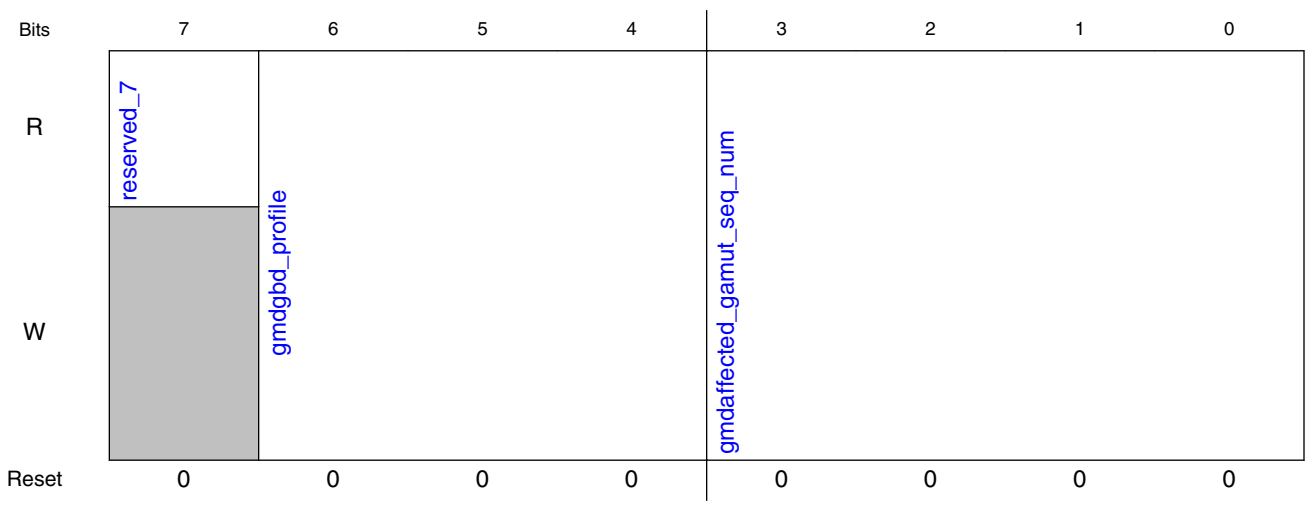
Register	Offset
fc_gmd_hb	104h

13.9.3.5.138.2 Function

Frame Composer GMD Packet Profile and Gamut Sequence Configuration Register

This register configures the GMD packet header affected_gamut_seq_num and gmd_profile bits. For more information, refer to the HDMI 1.4b specification.

13.9.3.5.138.3 Diagram



13.9.3.5.138.4 Fields

Field	Function
7 reserved_7	Reserved and read as zero
6-4 gmdgbd_profile	GMD profile bits. GMD profile bits. DWC_hdmi_tx only supports Profile 0 (P0) of the Gamut Boundary Description Metadata Profiles described in the HDMI 1.4 Specification (which defines four profiles, P0-P4).
3-0	Affected gamut sequence number

Field	Function
gmdaffected_gamut_seq_num	

13.9.3.5.139 Frame Composer AMP Packet Header Register 1 (fc_amp_hb1)

13.9.3.5.139.1 Offset

Register	Offset
fc_amp_hb1	128h

13.9.3.5.139.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_amp_hb0							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.139.3 Fields

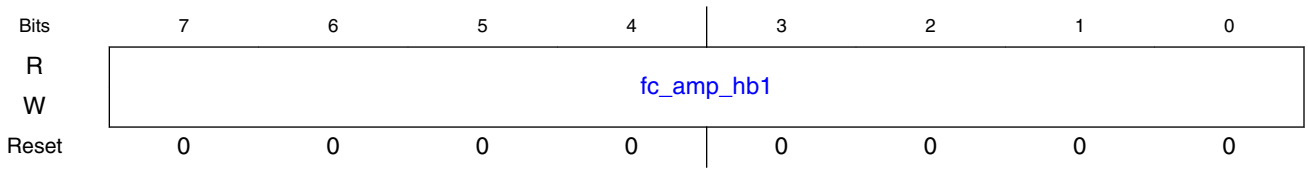
Field	Function
7-0 fc_amp_hb0	Frame Composer AMP Packet Header Register 1

13.9.3.5.140 Frame Composer AMP Packet Header Register 2 (fc_amp_hb2)

13.9.3.5.140.1 Offset

Register	Offset
fc_amp_hb2	129h

13.9.3.5.140.2 Diagram



13.9.3.5.140.3 Fields

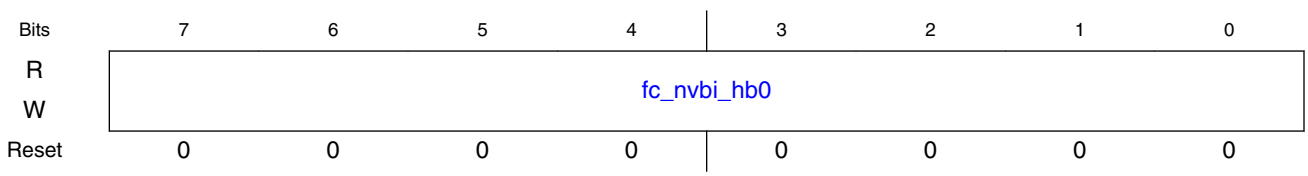
Field	Function
7-0 fc_amp_hb1	Frame Composer AMP Packet Header Register 2

13.9.3.5.141 Frame Composer NTSC VBI Packet Header Register 1 (fc_nvbi_hb1)

13.9.3.5.141.1 Offset

Register	Offset
fc_nvbi_hb1	148h

13.9.3.5.141.2 Diagram



13.9.3.5.141.3 Fields

Field	Function
7-0 fc_nvbi_hb0	Frame Composer NTSC VBI Packet Header Register 1

13.9.3.5.142 Frame Composer NTSC VBI Packet Header Register 2 (fc_nvbi_hb2)

13.9.3.5.142.1 Offset

Register	Offset
fc_nvbi_hb2	149h

13.9.3.5.142.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_nvbi_hb1							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.142.3 Fields

Field	Function
7-0 fc_nvbi_hb1	Frame Composer NTSC VBI Packet Header Register 2

13.9.3.5.143 Frame Composer DRM Packet Update Register This register performs an DRM packet content update according to the configured packet body (FC_DRM_PB0 to FC_DRM_PB27) and packet header (FC_DRM_HB). (fc_drm_up)

13.9.3.5.143.1 Offset

Register	Offset
fc_drm_up	167h

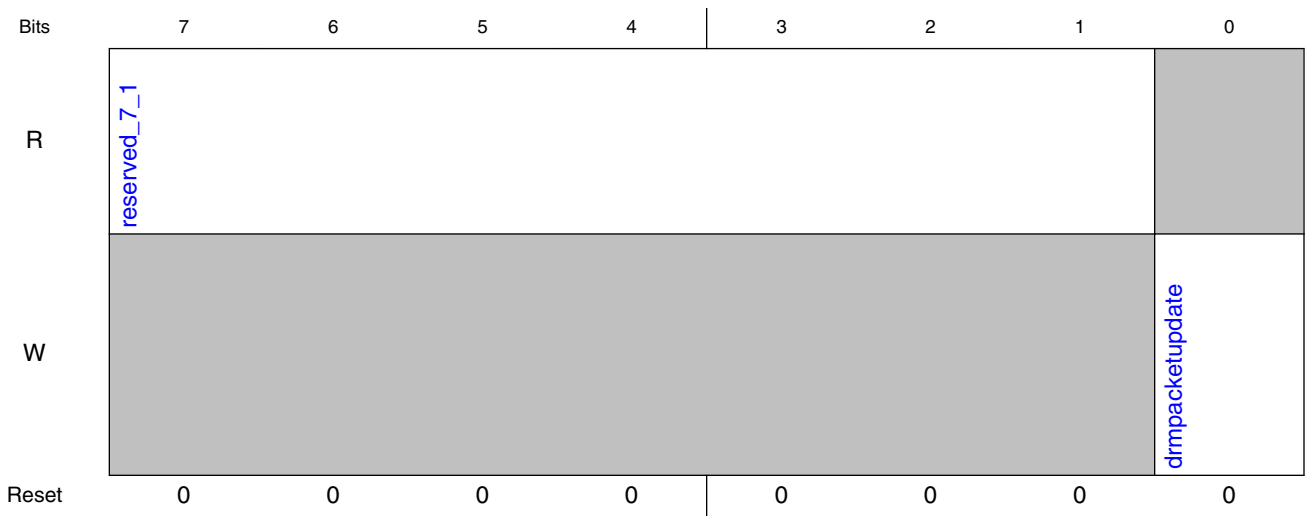
13.9.3.5.143.2 Function

Frame Composer DRM Packet Update Register

This register performs an DRM packet content update according to the configured packet body (FC_DRM_PB0 to FC_DRM_PB27) and packet header (FC_DRM_HB). This active high auto clear register reflects the body and header configurations on the DRM packets change to be performed.

Attention packet update request must only be done after correct configuration of DRM packet body and header registers.

13.9.3.5.143.3 Diagram



13.9.3.5.143.4 Fields

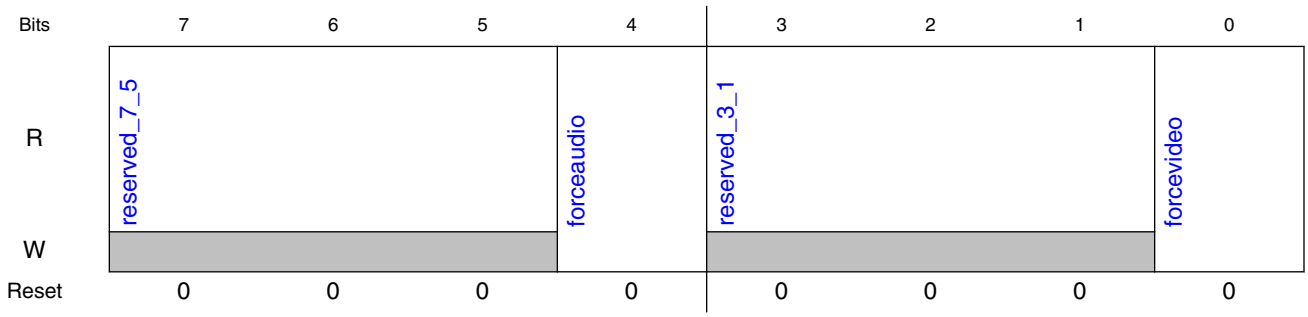
Field	Function
7-1 reserved_7_1	Reserved and read as zero
0 drmpacketupdate	DRM packet update

13.9.3.5.144 Frame Composer video/audio Force Enable Register This register allows to force the controller to output audio and video data the values configured in the FC_DBGAUD and FC_DBGTMDS registers. (fc_dbgforce)

13.9.3.5.144.1 Offset

Register	Offset
fc_dbgforce	200h

13.9.3.5.144.2 Diagram



13.9.3.5.144.3 Fields

Field	Function
7-5 reserved_7_5	Reserved and read as zero
4 forceaudio	Force fixed audio output with FC_DBGAUDxCHx register contents.
3-1 reserved_3_1	Reserved and read as zero
0 forcevideo	Force fixed video output with FC_DBGTMDSx register contents.

13.9.3.5.145 Frame Composer Audio Data Channel 0 Register 0 Configures the audio fixed data to be used in channel 0 when in fixed audio selection. (fc_dbgaud0ch0)

13.9.3.5.145.1 Offset

Register	Offset
fc_dbgaud0ch0	201h

13.9.3.5.145.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud0ch0							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.145.3 Fields

Field	Function
7-0 fc_dbgaud0ch0	Frame Composer Audio Data Channel 0 Register 0

13.9.3.5.146 Frame Composer Audio Data Channel 0 Register 1 Configures the audio fixed data to be used in channel 0 when in fixed audio selection. (fc_dbgaud1ch0)

13.9.3.5.146.1 Offset

Register	Offset
fc_dbgaud1ch0	202h

13.9.3.5.146.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud1ch0							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.146.3 Fields

Field	Function
7-0 fc_dbgaud1ch0	Frame Composer Audio Data Channel 0 Register 1

13.9.3.5.147 Frame Composer Audio Data Channel 0 Register 2 Configures the audio fixed data to be used in channel 0 when in fixed audio selection. (fc_dbgaud2ch0)

13.9.3.5.147.1 Offset

Register	Offset
fc_dbgaud2ch0	203h

13.9.3.5.147.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud2ch0							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.147.3 Fields

Field	Function
7-0 fc_dbgaud2ch0	Frame Composer Audio Data Channel 0 Register 2

13.9.3.5.148 Frame Composer Audio Data Channel 1 Register 0 Configures the audio fixed data to be used in channel 1 when in fixed audio selection. (fc_dbgaud0ch1)

13.9.3.5.148.1 Offset

Register	Offset
fc_dbgaud0ch1	204h

13.9.3.5.148.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud0ch1							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.148.3 Fields

Field	Function
7-0 fc_dbgaud0ch1	Frame Composer Audio Data Channel 1 Register 0

13.9.3.5.149 Frame Composer Audio Data Channel 1 Register 1 Configures the audio fixed data to be used in channel 1 when in fixed audio selection. (fc_dbgaud1ch1)

13.9.3.5.149.1 Offset

Register	Offset
fc_dbgaud1ch1	205h

13.9.3.5.149.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud1ch1							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.149.3 Fields

Field	Function
7-0 fc_dbgaud1ch1	Frame Composer Audio Data Channel 1 Register 1

13.9.3.5.150 Frame Composer Audio Data Channel 1 Register 2 Configures the audio fixed data to be used in channel 1 when in fixed audio selection. (fc_dbgaud2ch1)

13.9.3.5.150.1 Offset

Register	Offset
fc_dbgaud2ch1	206h

13.9.3.5.150.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud2ch1							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.150.3 Fields

Field	Function
7-0 fc_dbgaud2ch1	Frame Composer Audio Data Channel 1 Register 2

13.9.3.5.151 Frame Composer Audio Data Channel 2 Register 0 Configures the audio fixed data to be used in channel 2 when in fixed audio selection. (fc_dbgaud0ch2)

13.9.3.5.151.1 Offset

Register	Offset
fc_dbgaud0ch2	207h

13.9.3.5.151.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud0ch2							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.151.3 Fields

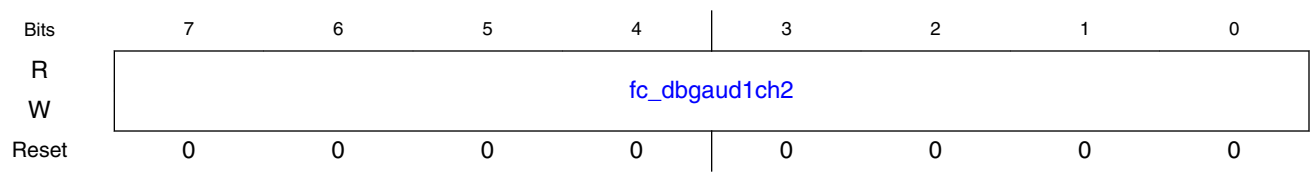
Field	Function
7-0 fc_dbgaud0ch2	Frame Composer Audio Data Channel 2 Register 0

13.9.3.5.152 Frame Composer Audio Data Channel 2 Register 1 Configures the audio fixed data to be used in channel 2 when in fixed audio selection. (fc_dbgaud1ch2)

13.9.3.5.152.1 Offset

Register	Offset
fc_dbgaud1ch2	208h

13.9.3.5.152.2 Diagram



13.9.3.5.152.3 Fields

Field	Function
7-0 fc_dbgaud1ch2	Frame Composer Audio Data Channel 2 Register 1

13.9.3.5.153 Frame Composer Audio Data Channel 2 Register 2 Configures the audio fixed data to be used in channel 2 when in fixed audio selection. (fc_dbgaud2ch2)

13.9.3.5.153.1 Offset

Register	Offset
fc_dbgaud2ch2	209h

13.9.3.5.153.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud2ch2							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.153.3 Fields

Field	Function
7-0 fc_dbgaud2ch2	Frame Composer Audio Data Channel 2 Register 2

13.9.3.5.154 Frame Composer Audio Data Channel 3 Register 0 Configures the audio fixed data to be used in channel 3 when in fixed audio selection. (fc_dbgaud0ch3)

13.9.3.5.154.1 Offset

Register	Offset
fc_dbgaud0ch3	20Ah

13.9.3.5.154.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud0ch3							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.154.3 Fields

Field	Function
7-0 fc_dbgaud0ch3	Frame Composer Audio Data Channel 3 Register 0

13.9.3.5.155 Frame Composer Audio Data Channel 3 Register 1 Configures the audio fixed data to be used in channel 3 when in fixed audio selection. (fc_dbgaud1ch3)

13.9.3.5.155.1 Offset

Register	Offset
fc_dbgaud1ch3	20Bh

13.9.3.5.155.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud1ch3							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.155.3 Fields

Field	Function
7-0 fc_dbgaud1ch3	Frame Composer Audio Data Channel 3 Register 1

13.9.3.5.156 Frame Composer Audio Data Channel 3 Register 2 Configures the audio fixed data to be used in channel 3 when in fixed audio selection. (fc_dbgaud2ch3)

13.9.3.5.156.1 Offset

Register	Offset
fc_dbgaud2ch3	20Ch

13.9.3.5.156.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud2ch3							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.156.3 Fields

Field	Function
7-0 fc_dbgaud2ch3	Frame Composer Audio Data Channel 3 Register 2

13.9.3.5.157 Frame Composer Audio Data Channel 4 Register 0 Configures the audio fixed data to be used in channel 4 when in fixed audio selection. (fc_dbgaud0ch4)

13.9.3.5.157.1 Offset

Register	Offset
fc_dbgaud0ch4	20Dh

13.9.3.5.157.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud0ch4							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.157.3 Fields

Field	Function
7-0 fc_dbgaud0ch4	Frame Composer Audio Data Channel 4 Register 0

13.9.3.5.158 Frame Composer Audio Data Channel 4 Register 1 Configures the audio fixed data to be used in channel 4 when in fixed audio selection. (fc_dbgaud1ch4)

13.9.3.5.158.1 Offset

Register	Offset
fc_dbgaud1ch4	20Eh

13.9.3.5.158.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud1ch4							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.158.3 Fields

Field	Function
7-0 fc_dbgaud1ch4	Frame Composer Audio Data Channel 4 Register 1

13.9.3.5.159 Frame Composer Audio Data Channel 4 Register 2 Configures the audio fixed data to be used in channel 4 when in fixed audio selection. (fc_dbgaud2ch4)

13.9.3.5.159.1 Offset

Register	Offset
fc_dbgaud2ch4	20Fh

13.9.3.5.159.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud2ch4							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.159.3 Fields

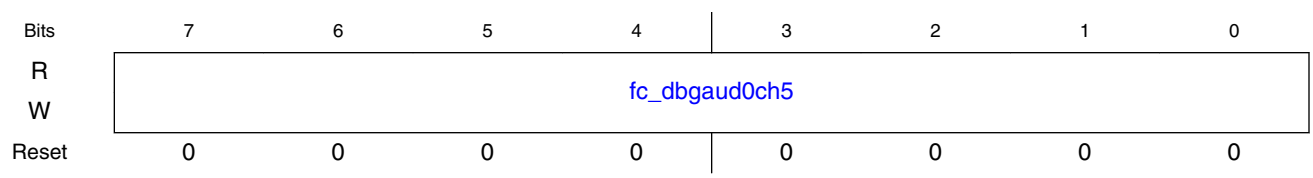
Field	Function
7-0 fc_dbgaud2ch4	Frame Composer Audio Data Channel 4 Register 2

13.9.3.5.160 Frame Composer Audio Data Channel 5 Register 0 Configures the audio fixed data to be used in channel 5 when in fixed audio selection. (fc_dbgaud0ch5)

13.9.3.5.160.1 Offset

Register	Offset
fc_dbgaud0ch5	210h

13.9.3.5.160.2 Diagram



13.9.3.5.160.3 Fields

Field	Function
7-0 fc_dbgaud0ch5	Frame Composer Audio Data Channel 5 Register 0

13.9.3.5.161 Frame Composer Audio Data Channel 5 Register 1 Configures the audio fixed data to be used in channel 5 when in fixed audio selection. (fc_dbgaud1ch5)

13.9.3.5.161.1 Offset

Register	Offset
fc_dbgaud1ch5	211h

13.9.3.5.161.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud1ch5							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.161.3 Fields

Field	Function
7-0 fc_dbgaud1ch5	Frame Composer Audio Data Channel 5 Register 1

13.9.3.5.162 Frame Composer Audio Data Channel 5 Register 2 Configures the audio fixed data to be used in channel 5 when in fixed audio selection. (fc_dbgaud2ch5)

13.9.3.5.162.1 Offset

Register	Offset
fc_dbgaud2ch5	212h

13.9.3.5.162.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud2ch5							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.162.3 Fields

Field	Function
7-0 fc_dbgaud2ch5	Frame Composer Audio Data Channel 5 Register 2

13.9.3.5.163 Frame Composer Audio Data Channel 6 Register 0 Configures the audio fixed data to be used in channel 6 when in fixed audio selection. (fc_dbgaud0ch6)

13.9.3.5.163.1 Offset

Register	Offset
fc_dbgaud0ch6	213h

13.9.3.5.163.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud0ch6							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.163.3 Fields

Field	Function
7-0 fc_dbgaud0ch6	Frame Composer Audio Data Channel 6 Register 0

13.9.3.5.164 Frame Composer Audio Data Channel 6 Register 1 Configures the audio fixed data to be used in channel 6 when in fixed audio selection. (fc_dbgaud1ch6)

13.9.3.5.164.1 Offset

Register	Offset
fc_dbgaud1ch6	214h

13.9.3.5.164.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud1ch6							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.164.3 Fields

Field	Function
7-0 fc_dbgaud1ch6	Frame Composer Audio Data Channel 6 Register 1

13.9.3.5.165 Frame Composer Audio Data Channel 6 Register 2 Configures the audio fixed data to be used in channel 6 when in fixed audio selection. (fc_dbgaud2ch6)

13.9.3.5.165.1 Offset

Register	Offset
fc_dbgaud2ch6	215h

13.9.3.5.165.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud2ch6							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.165.3 Fields

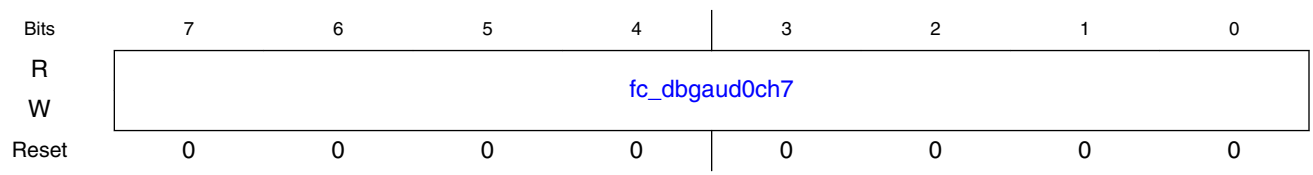
Field	Function
7-0 fc_dbgaud2ch6	Frame Composer Audio Data Channel 6 Register 2

13.9.3.5.166 Frame Composer Audio Data Channel 7 Register 0 Configures the audio fixed data to be used in channel 7 when in fixed audio selection. (fc_dbgaud0ch7)

13.9.3.5.166.1 Offset

Register	Offset
fc_dbgaud0ch7	216h

13.9.3.5.166.2 Diagram



13.9.3.5.166.3 Fields

Field	Function
7-0 fc_dbgaud0ch7	Frame Composer Audio Data Channel 7 Register 0

13.9.3.5.167 Frame Composer Audio Data Channel 7 Register 1 Configures the audio fixed data to be used in channel 7 when in fixed audio selection. (fc_dbgaud1ch7)

13.9.3.5.167.1 Offset

Register	Offset
fc_dbgaud1ch7	217h

13.9.3.5.167.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud1ch7							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.167.3 Fields

Field	Function
7-0 fc_dbgaud1ch7	Frame Composer Audio Data Channel 7 Register 1

13.9.3.5.168 Frame Composer Audio Data Channel 7 Register 2 Configures the audio fixed data to be used in channel 7 when in fixed audio selection. (fc_dbgaud2ch7)

13.9.3.5.168.1 Offset

Register	Offset
fc_dbgaud2ch7	218h

13.9.3.5.168.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	fc_dbgaud2ch7							
Reset	0	0	0	0	0	0	0	0

13.9.3.5.168.3 Fields

Field	Function
7-0 fc_dbgaud2ch7	Frame Composer Audio Data Channel 7 Register 2

13.9.3.6 PHYConfiguration register descriptions

13.9.3.6.1 Memory map

PHYConfiguration registers base address: 3000h

Offset	Register	Width (In bits)	Access	Reset value
0h	PHY Configuration Register This register holds the power down, data enable polarity, and interface control of the HDMI Source PHY control. (phy_conf0)	8	RW	06h
1h	PHY Test Interface Register 0 PHY TX mapped test interface (control). (phy_tst0)	8	RW	00h
2h	PHY Test Interface Register 1 PHY TX mapped text interface (data in). (phy_tst1)	8	RW	00h
3h	PHY Test Interface Register 2 PHY TX mapped text interface (data out). (phy_tst2)	8	RO	00h
4h	PHY RXSENSE, PLL Lock, and HPD Status Register This register contains the following active high packet sent status indications. (phy_stat0)	8	RO	00h
5h	PHY RXSENSE, PLL Lock, and HPD Interrupt Register This register contains the interrupt indication of the PHY_STAT0 status interrupts. (phy_int0)	8	RO	00h
6h	PHY RXSENSE, PLL Lock, and HPD Mask Register Mask register for generation of PHY_INT0 interrupts. (phy_mask0)	8	RW	00h
7h	PHY RXSENSE, PLL Lock, and HPD Polarity Register Polarity register for generation of PHY_INT0 interrupts. (phy_pol0)	8	RW	F3h
20h	PHY I2C Slave Address Configuration Register (phy_i2cm_slave)	8	RW	00h
21h	PHY I2C Address Configuration Register This register writes the address for read and write operations. (phy_i2cm_address)	8	RW	00h
22h	PHY I2C Data Write Register 1 (phy_i2cm_datao_1)	8	RW	00h
23h	PHY I2C Data Write Register 0 (phy_i2cm_datao_0)	8	RW	00h
24h	PHY I2C Data Read Register 1 (phy_i2cm_datai_1)	8	RO	00h
25h	PHY I2C Data Read Register 0 (phy_i2cm_datai_0)	8	RO	00h
26h	PHY I2C RD/RD_EXT/WR Operation Register This register requests read and write operations from the I2C Master PHY. (phy_i2cm_operation)	8	WO	00h
27h	PHY I2C Done Interrupt Register This register contains and configures I2C master PHY done interrupt. (phy_i2cm_int)	8	RW	08h
28h	PHY I2C error Interrupt Register This register contains and configures the I2C master PHY error interrupts. (phy_i2cm_ctlint)	8	RW	88h
29h	PHY I2C Speed control Register This register wets the I2C Master PHY to work in either Fast or Standard mode. (phy_i2cm_div)	8	RW	0Bh
2Ah	PHY I2C SW reset control register This register sets the I2C Master PHY software reset. (phy_i2cm_softrstz)	8	RW	01h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
2Bh	PHY I2C Slow Speed SCL High Level Control Register 1 (phy_i2cm_ss_scl_hcnt_1_addr)	8	RW	00h
2Ch	PHY I2C Slow Speed SCL High Level Control Register 0 (phy_i2cm_ss_scl_hcnt_0_addr)	8	RW	6Ch
2Dh	PHY I2C Slow Speed SCL Low Level Control Register 1 (phy_i2cm_ss_scl_lcnt_1_addr)	8	RW	00h
2Eh	PHY I2C Slow Speed SCL Low Level Control Register 0 (phy_i2cm_ss_scl_lcnt_0_addr)	8	RW	7Fh
2Fh	PHY I2C Fast Speed SCL High Level Control Register 1 (phy_i2cm_fs_scl_hcnt_1_addr)	8	RW	00h
30h	PHY I2C Fast Speed SCL High Level Control Register 0 (phy_i2cm_fs_scl_hcnt_0_addr)	8	RW	11h
31h	PHY I2C Fast Speed SCL Low Level Control Register 1 (phy_i2cm_fs_scl_lcnt_1_addr)	8	RW	00h
32h	PHY I2C Fast Speed SCL Low Level Control Register 0 (phy_i2cm_fs_scl_lcnt_0_addr)	8	RW	24h
33h	PHY I2C SDA HOLD Control Register (phy_i2cm_sda_hold)	8	RW	09h
34h	PHY I2C/JTAG I/O Configuration Control Register (jtag_phy_config)	8	RW	11h
35h	PHY JTAG Clock Control Register (jtag_phy_tap_tck)	8	RW	00h
36h	PHY JTAG TAP In Control Register (jtag_phy_tap_in)	8	RW	10h
37h	PHY JTAG TAP Out Control Register (jtag_phy_tap_out)	8	RO	00h
38h	PHY JTAG Address Control Register (jtag_phy_addr)	8	RW	00h

13.9.3.6.2 PHY Configuration Register This register holds the power down, data enable polarity, and interface control of the HDMI Source PHY control. (phy_conf0)

13.9.3.6.2.1 Offset

Register	Offset
phy_conf0	0h

13.9.3.6.2.2 Function

PHY Configuration Register

This register holds the power down, data enable polarity, and interface control of the HDMI Source PHY control. For more information, refer to the DesignWare Cores HDMI TX PHY Databook.

13.9.3.6.2.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	spares_2	spares_1	sparectrl	pddq	txpwron	enhpdrxsense	seldataenpol	seldipif
W								
Reset	0	0	0	0	0	1	1	0

13.9.3.6.2.4 Fields

Field	Function
7 spares_2	Reserved as "spare" register with no associated functionality.
6 spares_1	Reserved as "spare" register with no associated functionality.
5 sparectrl	Reserved as "spare" register with no associated functionality.
4 pddq	PHY PDDQ signal.
3 txpwron	PHY TXPWRON signal.
2 enhpdrxsense	PHY ENHPDRXSENSE signal.
1 seldataenpol	Select data enable polarity.
0 seldipif	Select interface control.

13.9.3.6.3 PHY Test Interface Register 0 PHY TX mapped test interface (control). (phy_tst0)

13.9.3.6.3.1 Offset

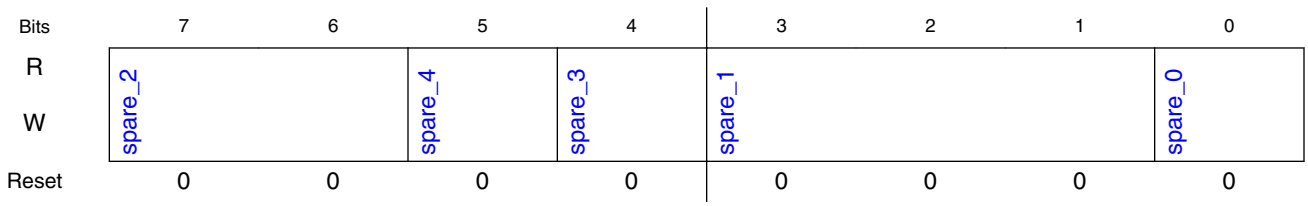
Register	Offset
phy_tst0	1h

13.9.3.6.3.2 Function

PHY Test Interface Register 0

PHY TX mapped test interface (control). For more information, refer to the DesignWare Cores HDMI TX PHY Databook.

13.9.3.6.3.3 Diagram



13.9.3.6.3.4 Fields

Field	Function
7-6 spare_2	Reserved as "spare" bit with no associated functionality.
5 spare_4	Reserved as "spare" register with no associated functionality.
4 spare_3	Reserved as "spare" register with no associated functionality.
3-1 spare_1	Reserved as "spare" bit with no associated functionality.
0 spare_0	Reserved as "spare" register with no associated functionality.

13.9.3.6.4 PHY Test Interface Register 1 PHY TX mapped text interface (data in). (phy_tst1)

13.9.3.6.4.1 Offset

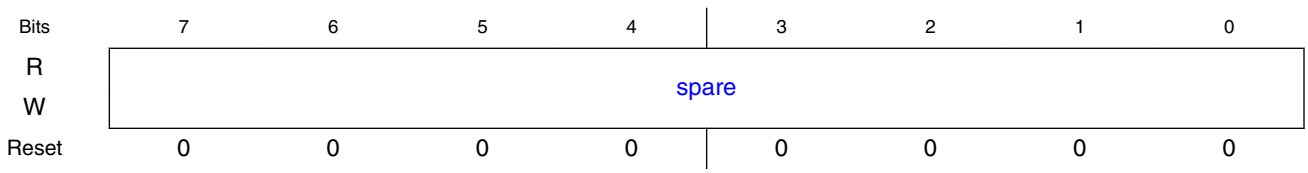
Register	Offset
phy_tst1	2h

13.9.3.6.4.2 Function

PHY Test Interface Register 1

PHY TX mapped text interface (data in). For more information, refer to the DesignWare Cores HDMI TX PHY Databook.

13.9.3.6.4.3 Diagram



13.9.3.6.4.4 Fields

Field	Function
7-0 spare	Reserved as "spare" register with no associated functionality.

13.9.3.6.5 PHY Test Interface Register 2 PHY TX mapped text interface (data out). (phy_tst2)

13.9.3.6.5.1 Offset

Register	Offset
phy_tst2	3h

13.9.3.6.5.2 Function

PHY Test Interface Register 2

PHY TX mapped text interface (data out). For more information, refer to the DesignWare Cores HDMI TX PHY Databook.

13.9.3.6.5.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	spare							
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.6.5.4 Fields

Field	Function
7-0 spare	Reserved as "spare" register with no associated functionality.

13.9.3.6.6 PHY RXSENSE, PLL Lock, and HPD Status Register This register contains the following active high packet sent status indications. (phy_stat0)

13.9.3.6.6.1 Offset

Register	Offset
phy_stat0	4h

13.9.3.6.6.2 Function

PHY RXSENSE, PLL Lock, and HPD Status Register

This register contains the following active high packet sent status indications. For more information, refer to the DesignWare Cores HDMI TX PHY Databook.

13.9.3.6.6.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	RX_SENSE_3	RX_SENSE_2	RX_SENSE_1	RX_SENSE_0	reserved_3_2		HPD	TX_PHY_LOCK
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.6.6.4 Fields

Field	Function
7 RX_SENSE_3	Status bit. Status bit. TX PHY RX_SENSE indication for TMDS channel 3 driver. You may need to mask or change polarity of this interrupt after it has became active.
6 RX_SENSE_2	Status bit. Status bit. TX PHY RX_SENSE indication for TMDS channel 2 driver. You may need to mask or change polarity of this interrupt after it has became active.
5 RX_SENSE_1	Status bit. Status bit. TX PHY RX_SENSE indication for TMDS channel 1 driver. You may need to mask or change polarity of this interrupt after it has became active.
4 RX_SENSE_0	Status bit. Status bit. TX PHY RX_SENSE indication for TMDS channel 0 driver. You may need to mask or change polarity of this interrupt after it has became active.
3-2 reserved_3_2	Reserved and read as zero
1 HPD	Status bit. Status bit. HDMI Hot Plug Detect indication. You may need to mask or change polarity of this interrupt after it has became active.
0 TX_PHY_LOCK	Status bit. Status bit. TX PHY PLL lock indication. For more information, refer to the PHY databook. You may need to mask or change polarity of this interrupt after it has became active.

13.9.3.6.7 PHY RXSENSE, PLL Lock, and HPD Interrupt Register This register contains the interrupt indication of the PHY_STAT0 status interrupts. (phy_int0)

13.9.3.6.7.1 Offset

Register	Offset
phy_int0	5h

13.9.3.6.7.2 Function

PHY RXSENSE, PLL Lock, and HPD Interrupt Register

This register contains the interrupt indication of the PHY_STAT0 status interrupts. Interrupt generation is accomplished in the following way:

interrupt = (mask == 1'b0) && (polarity == status);

All these interrupts are forwarded to the Interrupt Handler sticky bit register ih_phy_stat0 and after ORed to a single main interrupt line to micro controller. Assertion of this interrupt implies that data related with the corresponding packet has been sent through the HDMI interface.

13.9.3.6.7.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	RX_SENSE_3	RX_SENSE_2	RX_SENSE_1	RX_SENSE_0	reserved_3_2		HPD	TX_PHY_LOCK
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.6.7.4 Fields

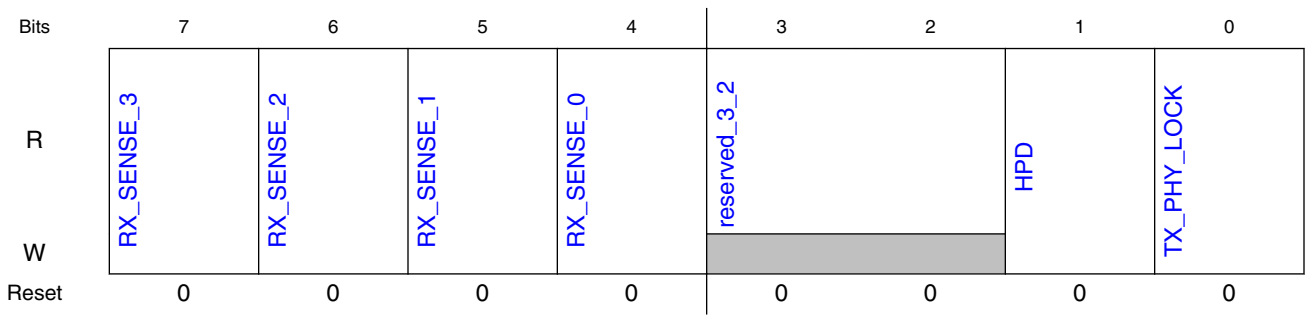
Field	Function
7 RX_SENSE_3	Interrupt indication bit. Interrupt indication bit. TX PHY RX_SENSE indication interruption for TMDS CLK driver.
6 RX_SENSE_2	Interrupt indication bit. Interrupt indication bit. TX PHY RX_SENSE indication interruption for TMDS channel 2 driver.
5 RX_SENSE_1	Interrupt indication bit. Interrupt indication bit. TX PHY RX_SENSE indication interruption for TMDS channel 1 driver.
4 RX_SENSE_0	Interrupt indication bit. Interrupt indication bit. TX PHY RX_SENSE indication interruption for TMDS channel 0 driver.
3-2 reserved_3_2	Reserved and read as zero
1 HPD	Interrupt indication bit. Interrupt indication bit. HDMI Hot Plug Detect indication interrupt.
0 TX_PHY_LOCK	Interrupt indication bit. Interrupt indication bit. TX PHY PLL lock indication interrupt.

13.9.3.6.8 PHY RXSENSE, PLL Lock, and HPD Mask Register Mask register for generation of PHY_INT0 interrupts. (phy_mask0)

13.9.3.6.8.1 Offset

Register	Offset
phy_mask0	6h

13.9.3.6.8.2 Diagram



13.9.3.6.8.3 Fields

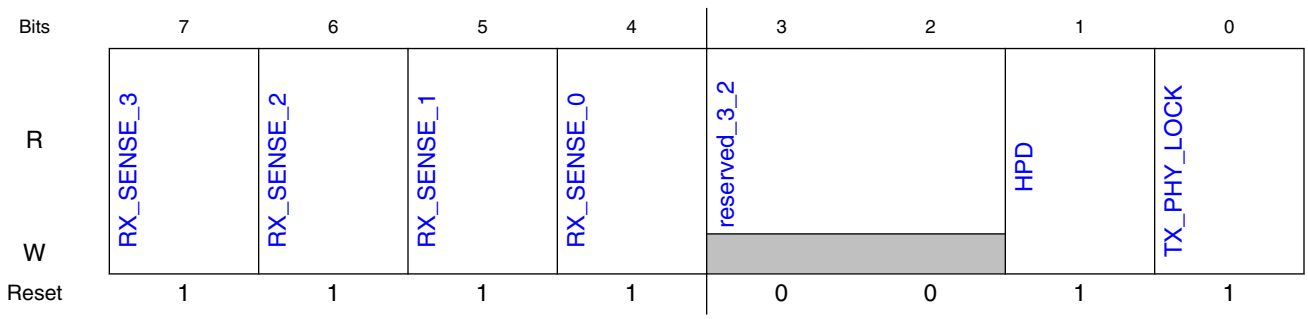
Field	Function
7 RX_SENSE_3	Mask bit for PHY_INT0. Mask bit for PHY_INT0.RX_SENSE[3] interrupt bit
6 RX_SENSE_2	Mask bit for PHY_INT0. Mask bit for PHY_INT0.RX_SENSE[2] interrupt bit
5 RX_SENSE_1	Mask bit for PHY_INT0. Mask bit for PHY_INT0.RX_SENSE[1] interrupt bit
4 RX_SENSE_0	Mask bit for PHY_INT0. Mask bit for PHY_INT0.RX_SENSE[0] interrupt bit
3-2 reserved_3_2	Reserved and read as zero
1 HPD	Mask bit for PHY_INT0. Mask bit for PHY_INT0.HPD interrupt bit
0 TX_PHY_LOCK	Mask bit for PHY_INT0. Mask bit for PHY_INT0.TX_PHY_LOCK interrupt bit

13.9.3.6.9 PHY RXSENSE, PLL Lock, and HPD Polarity Register Polarity register for generation of PHY_INT0 interrupts. (phy_pol0)

13.9.3.6.9.1 Offset

Register	Offset
phy_pol0	7h

13.9.3.6.9.2 Diagram



13.9.3.6.9.3 Fields

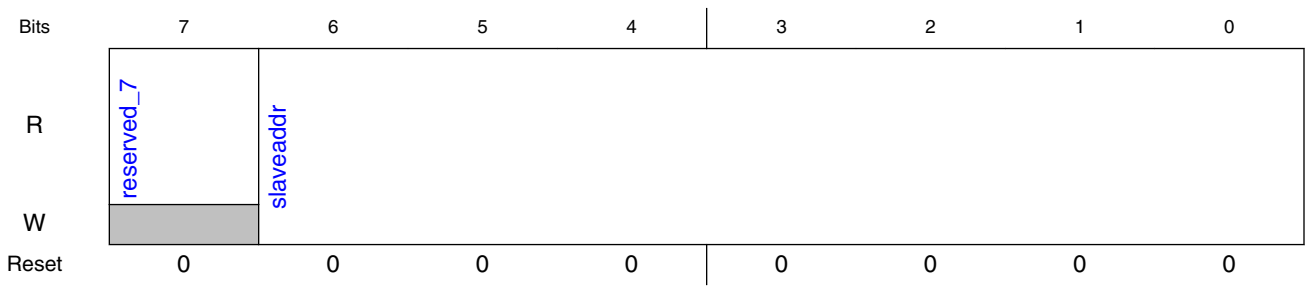
Field	Function
7 RX_SENSE_3	Polarity bit for PHY_INT0. Polarity bit for PHY_INT0.RX_SENSE[3] interrupt bit
6 RX_SENSE_2	Polarity bit for PHY_INT0. Polarity bit for PHY_INT0.RX_SENSE[2] interrupt bit
5 RX_SENSE_1	Polarity bit for PHY_INT0. Polarity bit for PHY_INT0.RX_SENSE[1] interrupt bit
4 RX_SENSE_0	Polarity bit for PHY_INT0. Polarity bit for PHY_INT0.RX_SENSE[0] interrupt bit
3-2 reserved_3_2	Reserved and read as zero
1 HPD	Polarity bit for PHY_INT0. Polarity bit for PHY_INT0.HPD interrupt bit
0 TX_PHY_LOCK	Polarity bit for PHY_INT0. Polarity bit for PHY_INT0.TX_PHY_LOCK interrupt bit

13.9.3.6.10 PHY I2C Slave Address Configuration Register (phy_i2cm_slave)

13.9.3.6.10.1 Offset

Register	Offset
phy_i2cm_slave	20h

13.9.3.6.10.2 Diagram



13.9.3.6.10.3 Fields

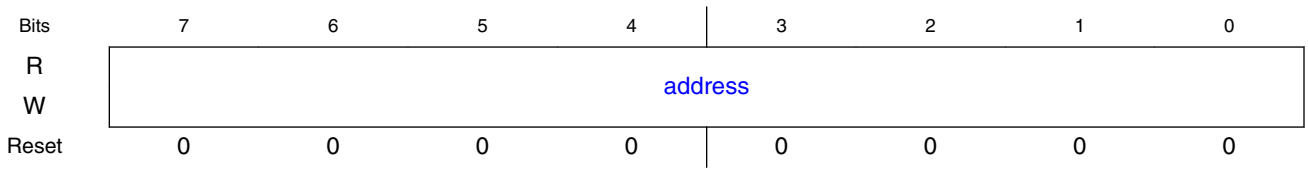
Field	Function
7 reserved_7	Reserved and read as zero
6-0 slaveaddr	Slave address to be sent during read and write operations. Slave address to be sent during read and write operations. - PHY Gen2 slave address: 7'h69 - HEAC PHY slave address: 7'h49

13.9.3.6.11 PHY I2C Address Configuration Register This register writes the address for read and write operations. (phy_i2cm_address)

13.9.3.6.11.1 Offset

Register	Offset
phy_i2cm_address	21h

13.9.3.6.11.2 Diagram



13.9.3.6.11.3 Fields

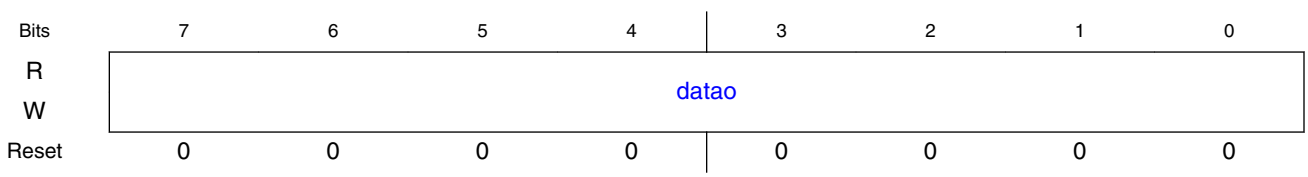
Field	Function
7-0 address	Register address for read and write operations

13.9.3.6.12 PHY I2C Data Write Register 1 (phy_i2cm_datao_1)

13.9.3.6.12.1 Offset

Register	Offset
phy_i2cm_datao_1	22h

13.9.3.6.12.2 Diagram



13.9.3.6.12.3 Fields

Field	Function
7-0 datao	Data MSB (datao[15:8]) to be written on register pointed by phy_i2cm_address [7:0].

13.9.3.6.13 PHY I2C Data Write Register 0 (phy_i2cm_datao_0)

13.9.3.6.13.1 Offset

Register	Offset
phy_i2cm_datao_0	23h

13.9.3.6.13.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	datao							
Reset	0	0	0	0	0	0	0	0

13.9.3.6.13.3 Fields

Field	Function
7-0 datao	Data LSB (datao[7:0]) to be written on register pointed by phy_i2cm_address [7:0].

13.9.3.6.14 PHY I2C Data Read Register 1 (phy_i2cm_datai_1)

13.9.3.6.14.1 Offset

Register	Offset
phy_i2cm_datai_1	24h

13.9.3.6.14.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	datai							
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.6.14.3 Fields

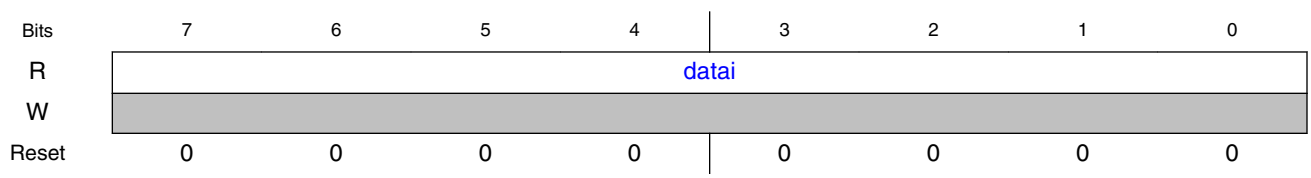
Field	Function
7-0 datai	Data MSB (datai[15:8]) read from register pointed by phy_i2cm_address[7:0].

13.9.3.6.15 PHY I2C Data Read Register 0 (phy_i2cm_datai_0)

13.9.3.6.15.1 Offset

Register	Offset
phy_i2cm_datai_0	25h

13.9.3.6.15.2 Diagram



13.9.3.6.15.3 Fields

Field	Function
7-0 datai	Data LSB (datai[7:0]) read from register pointed by phy_i2cm_address[7:0].

13.9.3.6.16 PHY I2C RD/RD_EXT/WR Operation Register This register requests read and write operations from the I2C Master PHY. (phy_i2cm_operation)

13.9.3.6.16.1 Offset

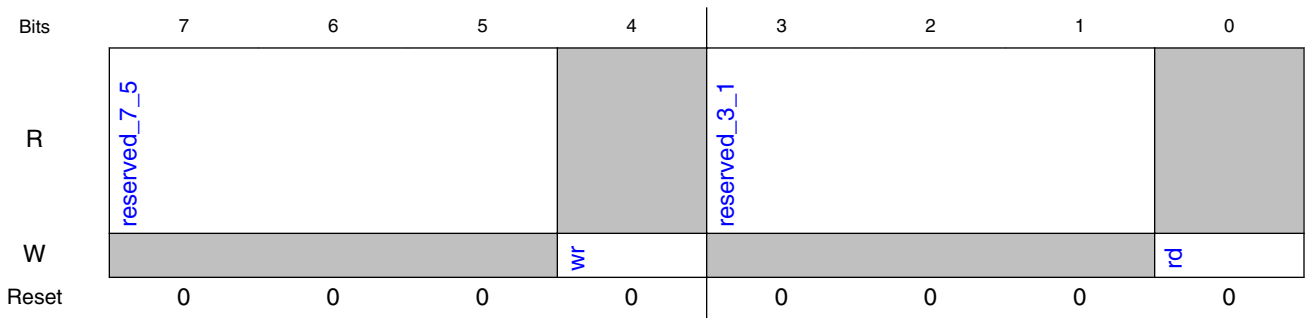
Register	Offset
phy_i2cm_operation	26h

13.9.3.6.16.2 Function

PHY I2C RD/RD_EXT/WR Operation Register

This register requests read and write operations from the I2C Master PHY. This register can only be written; reading this register always results in 00h. Writing 1'b1 simultaneously to read and write requests is considered a read request.

13.9.3.6.16.3 Diagram



13.9.3.6.16.4 Fields

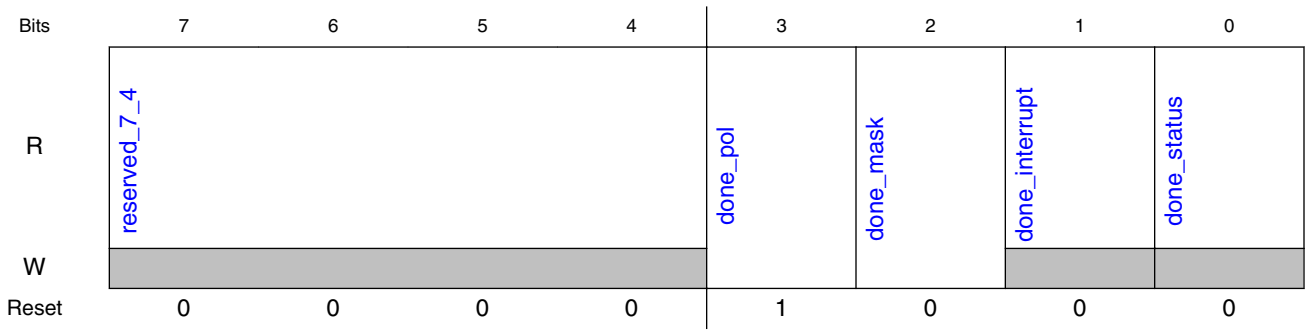
Field	Function
7-5 reserved_7_5	Reserved and read as zero
4 wr	Write operation request
3-1 reserved_3_1	Reserved and read as zero
0 rd	Read operation request

13.9.3.6.17 PHY I2C Done Interrupt Register This register contains and configures I2C master PHY done interrupt. (phy_i2cm_int)

13.9.3.6.17.1 Offset

Register	Offset
phy_i2cm_int	27h

13.9.3.6.17.2 Diagram



13.9.3.6.17.3 Fields

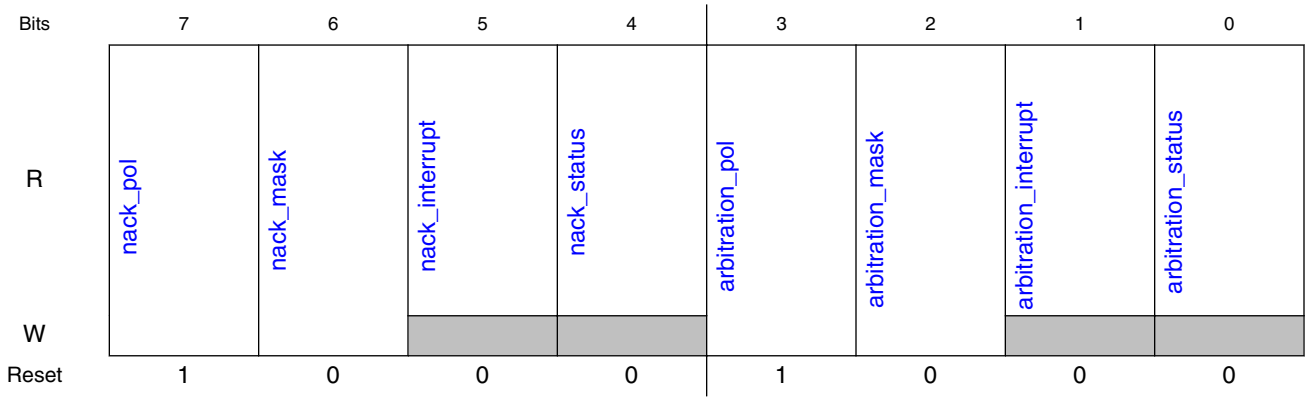
Field	Function
7-4 reserved_7_4	Reserved and read as zero
3 done_pol	Done interrupt polarity configuration
2 done_mask	Done interrupt mask signal
1 done_interrupt	Operation done interrupt bit. Operation done interrupt bit. Only lasts for 1 SFR clock cycle and is auto cleared after it. {done_interrupt = (done_mask==0b) && (done_status==done_pol)}
0 done_status	Operation done status bit. Operation done status bit. Marks the end of a read or write operation.

13.9.3.6.18 PHY I2C error Interrupt Register This register contains and configures the I2C master PHY error interrupts. (phy_i2cm_ctl int)

13.9.3.6.18.1 Offset

Register	Offset
phy_i2cm_ctlint	28h

13.9.3.6.18.2 Diagram



13.9.3.6.18.3 Fields

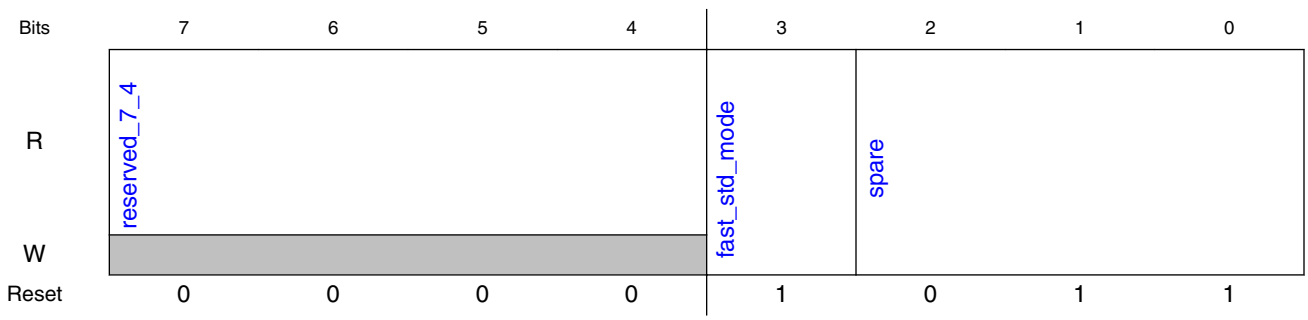
Field	Function
7 nack_pol	Not acknowledge error interrupt polarity configuration
6 nack_mask	Not acknowledge error interrupt mask signal
5 nack_interrupt	Not acknowledge error interrupt bit. Not acknowledge error interrupt bit. Only lasts for one SFR clock cycle and is auto cleared after it. {nack_interrupt = (nack_mask==0b) && (nack_status==nack_pol)}. Note: This bit field is read by the sticky bits present on the ih_i2cmphy_stat0 register.
4 nack_status	Not acknowledge error status bit. Not acknowledge error status bit. Error on I2C not acknowledge. Note: This bit field is read by the sticky bits present on the ih_i2cmphy_stat0 register.
3 arbitration_pol	Arbitration error interrupt polarity configuration.
2 arbitration_mask	Arbitration error interrupt mask signal.
1 arbitration_interrupt	Arbitration error interrupt bit {arbitration_interrupt = (arbitration_mask==0b) && (arbitration_status==arbitration_pol)} Note: This bit field is read by the sticky bits present on the ih_i2cmphy_stat0 register.
0 arbitration_status	Arbitration error status bit. Arbitration error status bit. Error on master I2C protocol arbitration. Only lasts for one SFR clock cycle and is auto cleared after it. Note: This bit field is read by the sticky bits present on the ih_i2cmphy_stat0 register.

13.9.3.6.19 PHY I2C Speed control Register This register wets the I2C Master PHY to work in either Fast or Standard mode. (phy_i2cm_div)

13.9.3.6.19.1 Offset

Register	Offset
phy_i2cm_div	29h

13.9.3.6.19.2 Diagram



13.9.3.6.19.3 Fields

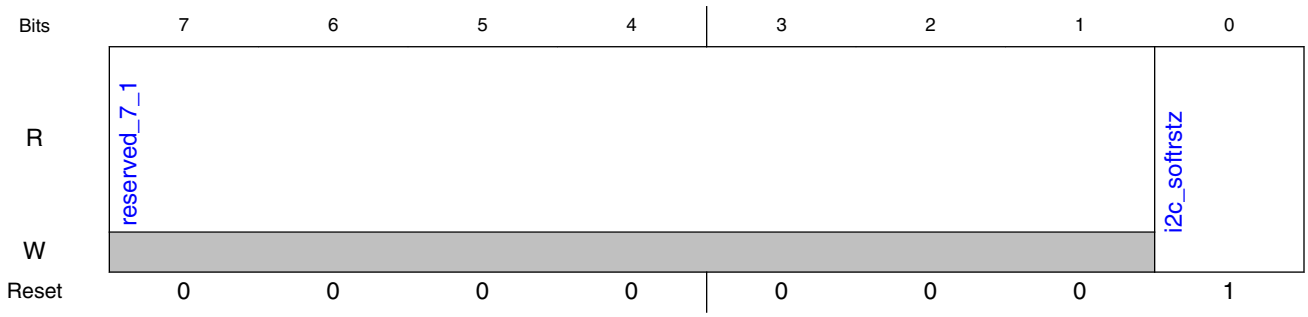
Field	Function
7-4 reserved_7_4	Reserved and read as zero
3 fast_std_mode	Sets the I2C Master to work in Fast Mode or Standard Mode: 1: Fast Mode 0: Standard Mode
2-0 spare	Reserved as "spare" register with no associated functionality.

13.9.3.6.20 PHY I2C SW reset control register This register sets the I2C Master PHY software reset. (phy_i2cm_softrstz)

13.9.3.6.20.1 Offset

Register	Offset
phy_i2cm_softrstz	2Ah

13.9.3.6.20.2 Diagram



13.9.3.6.20.3 Fields

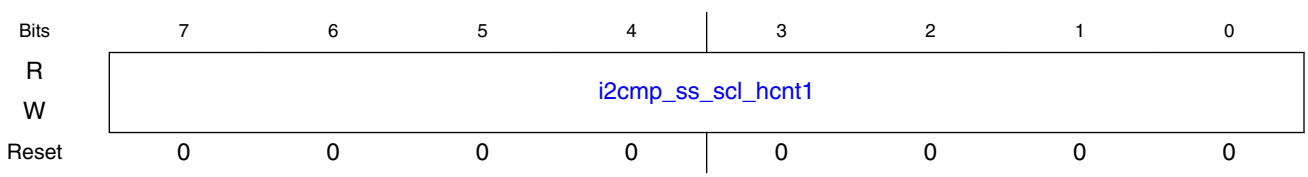
Field	Function
7-1 reserved_7_1	Reserved and read as zero
0 i2c_softrstz	I2C Master Software Reset. I2C Master Software Reset. Active by writing a zero and auto cleared to one in the following cycle.

13.9.3.6.21 PHY I2C Slow Speed SCL High Level Control Register 1 (phy_i2cm_ss_scl_hcnt_1_addr)

13.9.3.6.21.1 Offset

Register	Offset
phy_i2cm_ss_scl_hcnt_1_addr	2Bh

13.9.3.6.21.2 Diagram



13.9.3.6.21.3 Fields

Field	Function
7-0 i2cmp_ss_scl_hcnt1	PHY I2C Slow Speed SCL High Level Control Register 1

13.9.3.6.22 PHY I2C Slow Speed SCL High Level Control Register 0 (phy_i2cm_ss_scl_hcnt_0_addr)

13.9.3.6.22.1 Offset

Register	Offset
phy_i2cm_ss_scl_hcnt_0_addr	2Ch

13.9.3.6.22.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_ss_scl_hcnt0							
Reset	0	1	1	0	1	1	0	0

13.9.3.6.22.3 Fields

Field	Function
7-0 i2cmp_ss_scl_hcnt0	PHY I2C Slow Speed SCL High Level Control Register 0

13.9.3.6.23 PHY I2C Slow Speed SCL Low Level Control Register 1 (phy_i2cm_ss_scl_lcnt_1_addr)

13.9.3.6.23.1 Offset

Register	Offset
phy_i2cm_ss_scl_lcnt_1_addr	2Dh

13.9.3.6.23.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_ss_scl_lcnt1							
Reset	0	0	0	0	0	0	0	0

13.9.3.6.23.3 Fields

Field	Function
7-0 i2cmp_ss_scl_lcnt1	PHY I2C Slow Speed SCL Low Level Control Register 1

13.9.3.6.24 PHY I2C Slow Speed SCL Low Level Control Register 0 (phy_i2cm_ss_scl_lcnt_0_addr)

13.9.3.6.24.1 Offset

Register	Offset
phy_i2cm_ss_scl_lcnt_0_addr	2Eh

13.9.3.6.24.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_ss_scl_lcnt0							
Reset	0	1	1	1	1	1	1	1

13.9.3.6.24.3 Fields

Field	Function
7-0 i2cmp_ss_scl_lcnt0	PHY I2C Slow Speed SCL Low Level Control Register 0

13.9.3.6.25 PHY I2C Fast Speed SCL High Level Control Register 1 (phy_i2cm_fs_scl_hcnt_1_addr)

13.9.3.6.25.1 Offset

Register	Offset
phy_i2cm_fs_scl_hcnt_1_addr	2Fh

13.9.3.6.25.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_fs_scl_hcnt1							
Reset	0	0	0	0	0	0	0	0

13.9.3.6.25.3 Fields

Field	Function
7-0 i2cmp_fs_scl_hcnt1	PHY I2C Fast Speed SCL High Level Control Register 1

13.9.3.6.26 PHY I2C Fast Speed SCL High Level Control Register 0 (phy_i2cm_fs_scl_hcnt_0_addr)

13.9.3.6.26.1 Offset

Register	Offset
phy_i2cm_fs_scl_hcnt_0_addr	30h

13.9.3.6.26.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_fs_scl_hcnt0							
Reset	0	0	0	1	0	0	0	1

13.9.3.6.26.3 Fields

Field	Function
7-0 i2cmp_fs_scl_hcnt0	PHY I2C Fast Speed SCL High Level Control Register 0

13.9.3.6.27 PHY I2C Fast Speed SCL Low Level Control Register 1 (phy_i2cm_fs_scl_lcnt_1_addr)

13.9.3.6.27.1 Offset

Register	Offset
phy_i2cm_fs_scl_lcnt_1_addr	31h

13.9.3.6.27.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_fs_scl_lcnt1							
Reset	0	0	0	0	0	0	0	0

13.9.3.6.27.3 Fields

Field	Function
7-0 i2cmp_fs_scl_lcnt1	PHY I2C Fast Speed SCL Low Level Control Register 1

13.9.3.6.28 PHY I2C Fast Speed SCL Low Level Control Register 0 (phy_i2cm_fs_scl_lcnt_0_addr)

13.9.3.6.28.1 Offset

Register	Offset
phy_i2cm_fs_scl_lcnt_0_addr	32h

13.9.3.6.28.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_fs_scl_lcnt0							
Reset	0	0	1	0	0	1	0	0

13.9.3.6.28.3 Fields

Field	Function
7-0 i2cmp_fs_scl_lcnt0	PHY I2C Fast Speed SCL Low Level Control Register 0

13.9.3.6.29 PHY I2C SDA HOLD Control Register (phy_i2cm_sda_hold)

13.9.3.6.29.1 Offset

Register	Offset
phy_i2cm_sda_hold	33h

13.9.3.6.29.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	osda_hold							
Reset	0	0	0	0	1	0	0	1

13.9.3.6.29.3 Fields

Field	Function
7-0 osda_hold	Defines the number of SFR clock cycles to meet tHD:DAT (300 ns) osda_hold = round_to_high_integer (300 ns / (1/isfrclk_frequency))

13.9.3.6.30 PHY I2C/JTAG I/O Configuration Control Register (jtag_phy_config)

13.9.3.6.30.1 Offset

Register	Offset
jtag_phy_config	34h

13.9.3.6.30.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	reserved_7_5			i2c_jtagz	reserved_3_1			jtag_trst_n
W								
Reset	0	0	0	1	0	0	0	1

13.9.3.6.30.3 Fields

Field	Function
7-5	Reserved and read as zero

Table continues on the next page...

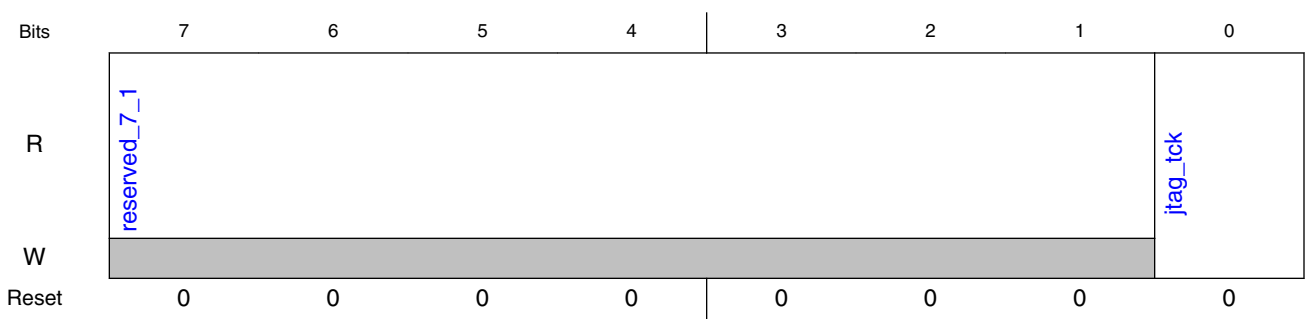
Field	Function
reserved_7_5	
4 i2c_jtagz	Configures the JTAG PHY interface output pin I2C_JTAGZ to select the PHY configuration interface when in internal control mode (iphy_ext_ctrl=1'b0) or ophyext_jtag_i2c_jtagz when PHY_EXTERNAL=1. Configures the JTAG PHY interface output pin I2C_JTAGZ to select the PHY configuration interface when in internal control mode (iphy_ext_ctrl=1'b0) or ophyext_jtag_i2c_jtagz when PHY_EXTERNAL=1. - 1'b0: JTAG configuration Interface - 1'b1: I2C configuration Interface
3-1 reserved_3_1	Reserved and read as zero
0 jtag_trst_n	Configures the JTAG PHY interface output pin JTAG_TRST_N when in internal control mode (iphy_ext_ctrl=1'b0) or ophyext_jtag_trst_n when PHY_EXTERNAL=1.

13.9.3.6.31 PHY JTAG Clock Control Register (jtag_phy_tap_tck)

13.9.3.6.31.1 Offset

Register	Offset
jtag_phy_tap_tck	35h

13.9.3.6.31.2 Diagram



13.9.3.6.31.3 Fields

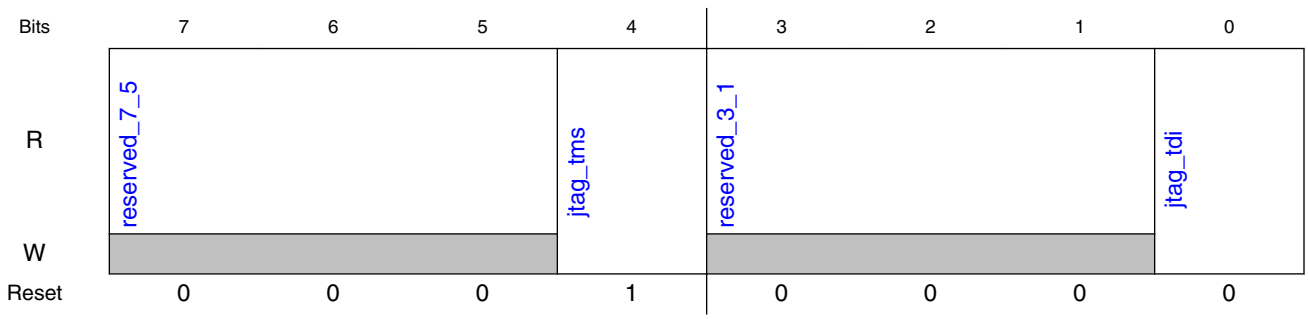
Field	Function
7-1 reserved_7_1	Reserved and read as zero
0 jtag_tck	Configures the JTAG PHY interface pin JTAG_TCK when in internal control mode (iphy_ext_ctrl=1'b0) or ophyext_jtag_tck when PHY_EXTERNAL=1.

13.9.3.6.32 PHY JTAG TAP In Control Register (jtag_phy_tap_in)

13.9.3.6.32.1 Offset

Register	Offset
jtag_phy_tap_in	36h

13.9.3.6.32.2 Diagram



13.9.3.6.32.3 Fields

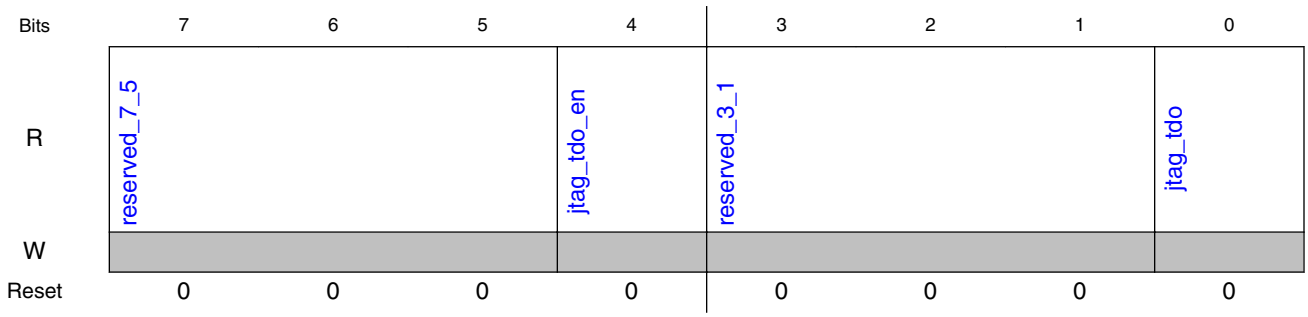
Field	Function
7-5 reserved_7_5	Reserved and read as zero
4 jtag_tms	Configures the JTAG PHY interface pin JTAG_TMS when in internal control mode (iphy_ext_ctrl=1'b0) or ophyext_jtag_tms when PHY_EXTERNAL=1.
3-1 reserved_3_1	Reserved and read as zero
0 jtag_tdi	Configures the JTAG PHY interface pin JTAG_TDI when in internal control mode (iphy_ext_ctrl=1'b0) or ophyext_jtag_tdi when PHY_EXTERNAL=1.

13.9.3.6.33 PHY JTAG TAP Out Control Register (jtag_phy_tap_out)

13.9.3.6.33.1 Offset

Register	Offset
jtag_phy_tap_out	37h

13.9.3.6.33.2 Diagram



13.9.3.6.33.3 Fields

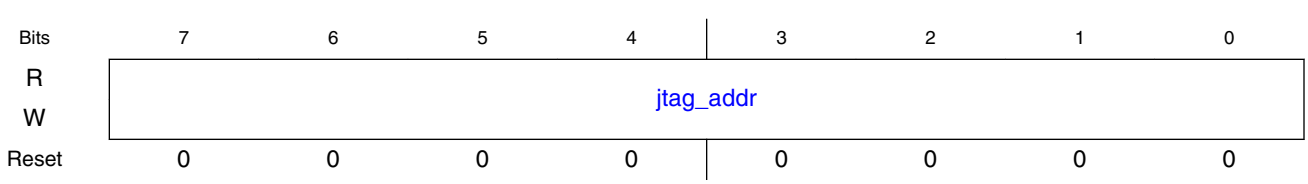
Field	Function
7-5 reserved_7_5	Reserved and read as zero
4 jtag_tdo_en	Read JTAG PHY interface input pin JTAG_TDO_EN when in internal control mode (iphy_ext_ctrl=1'b0) or iphyext_jtag_tdo_en when PHY_EXTERNAL=1
3-1 reserved_3_1	Reserved and read as zero
0 jtag_tdo	Read JTAG PHY interface input pin JTAG_TDO when in internal control mode (iphy_ext_ctrl=1'b0) or iphyext_jtag_tdo when PHY_EXTERNAL=1

13.9.3.6.34 PHY JTAG Address Control Register (jtag_phy_addr)

13.9.3.6.34.1 Offset

Register	Offset
jtag_phy_addr	38h

13.9.3.6.34.2 Diagram



13.9.3.6.34.3 Fields

Field	Function
7-0 jtag_addr	Configures the JTAG PHY interface pin JTAG_ADDR[7:0] when in internal control mode (iphy_ext_ctrl=1'b0) or iphyext_jtag_addr[7:0] when PHY_EXTERNAL=1

13.9.3.7 AudioPacketizer register descriptions

13.9.3.7.1 Memory map

AudioPacketizer base registers address: 3200h

Offset	Register	Width (In bits)	Access	Reset value
0h	Audio Clock Regenerator N Value Register 1 For N expected values, refer to the HDMI 1. (aud_n1)	8	RW	00h
1h	Audio Clock Regenerator N Value Register 2 For N expected values, refer to the HDMI 1. (aud_n2)	8	RW	00h
2h	Audio Clock Regenerator N Value Register 3 For N expected values, refer to the HDMI 1. (aud_n3)	8	RW	00h
3h	Audio Clock Regenerator CTS Value Register 1 For CTS expected values, refer to the HDMI 1. (aud_cts1)	8	RW	00h
4h	Audio Clock Regenerator CTS Register 2 For CTS expected values, refer to the HDMI 1. (aud_cts2)	8	RW	00h
5h	Audio Clock Regenerator CTS value Register 3. (aud_cts3)	8	RW	00h
6h	Audio Input Clock FS Factor Register (aud_inputclkfs)	8	RO	00h
7h	Audio CTS Dither Register (aud_cts_dither)	8	RW	11h

13.9.3.7.2 Audio Clock Regenerator N Value Register 1 For N expected values, refer to the HDMI 1. (aud_n1)

13.9.3.7.2.1 Offset

Register	Offset
aud_n1	0h

13.9.3.7.2.2 Function

Audio Clock Regenerator N Value Register 1

For N expected values, refer to the HDMI 1.4b specification.

13.9.3.7.2.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	AudN							
Reset	0	0	0	0	0	0	0	0

13.9.3.7.2.4 Fields

Field	Function
7-0 AudN	HDMI Audio Clock Regenerator N value

13.9.3.7.3 Audio Clock Regenerator N Value Register 2 For N expected values, refer to the HDMI 1. (aud_n2)

13.9.3.7.3.1 Offset

Register	Offset
aud_n2	1h

13.9.3.7.3.2 Function

Audio Clock Regenerator N Value Register 2

For N expected values, refer to the HDMI 1.4b specification.

13.9.3.7.3.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	AudN							
Reset	0	0	0	0	0	0	0	0

13.9.3.7.3.4 Fields

Field	Function
7-0 AudN	HDMI Audio Clock Regenerator N value

13.9.3.7.4 Audio Clock Regenerator N Value Register 3 For N expected values, refer to the HDMI 1. (aud_n3)

13.9.3.7.4.1 Offset

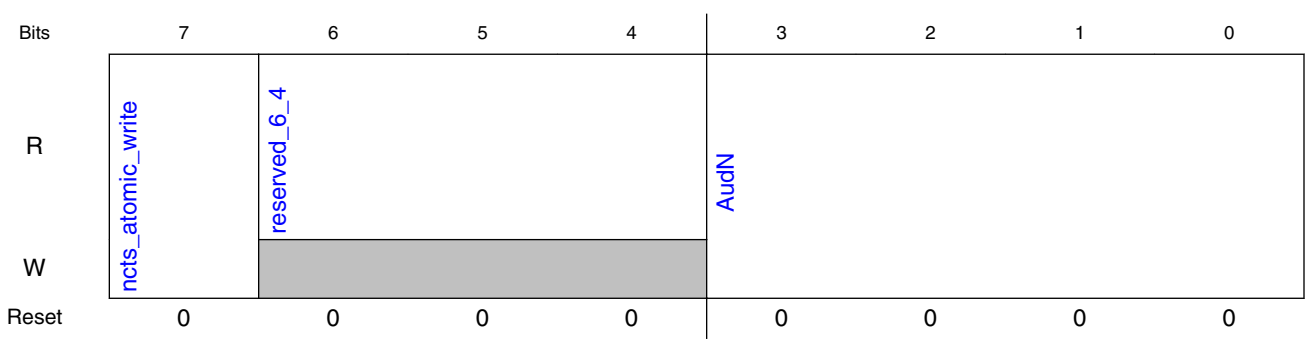
Register	Offset
aud_n3	2h

13.9.3.7.4.2 Function

Audio Clock Regenerator N Value Register 3

For N expected values, refer to the HDMI 1.4b specification.

13.9.3.7.4.3 Diagram



13.9.3.7.4.4 Fields

Field	Function
7 ncts_atomic_write	When set, the new N and CTS values are only used when aud_n1 register is written. When set, the new N and CTS values are only used when aud_n1 register is written. If clear, N and CTS data is updated each time a new N or CTS byte is written. The following write sequence is recommended:

Table continues on the next page...

Field	Function
	1. aud_n3 (set bit ncts_atomic_write if desired) 2. aud_cts3 (set CTS_manual and CTS value if desired/enabled) 3. aud_cts2 (required in CTS_manual) 4. aud_cts1 (required in CTS_manual) 5. aud_n3 (bit ncts_atomic_write with same value as in step 1.) 6. aud_n2 7. aud_n1 For dynamic N/CTS changes, perform only steps from 2-7 or 5-7 depending on the state of CTS_manual.
6-4 reserved_6_4	Reserved and read as zero
3-0 AudN	HDMI Audio Clock Regenerator N value

13.9.3.7.5 Audio Clock Regenerator CTS Value Register 1 For CTS expected values, refer to the HDMI 1. (aud_cts1)

13.9.3.7.5.1 Offset

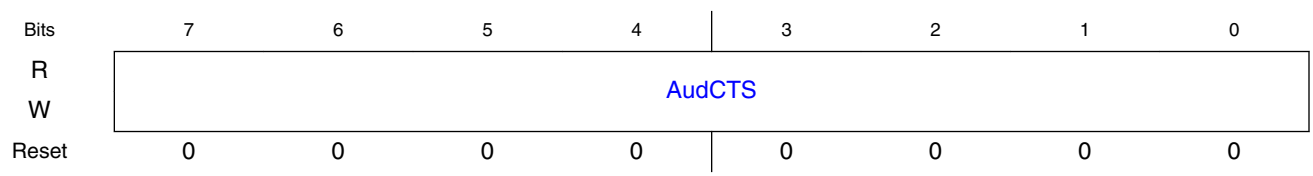
Register	Offset
aud_cts1	3h

13.9.3.7.5.2 Function

Audio Clock Regenerator CTS Value Register 1

For CTS expected values, refer to the HDMI 1.4b specification.

13.9.3.7.5.3 Diagram



13.9.3.7.5.4 Fields

Field	Function
7-0 AudCTS	HDMI Audio Clock Regenerator CTS calculated value. HDMI Audio Clock Regenerator CTS calculated value. This value can be manually set using the CTS_manual (AUD_CTS3) mechanism.

13.9.3.7.6 Audio Clock Regenerator CTS Register 2 For CTS expected values, refer to the HDMI 1. (aud_cts2)

13.9.3.7.6.1 Offset

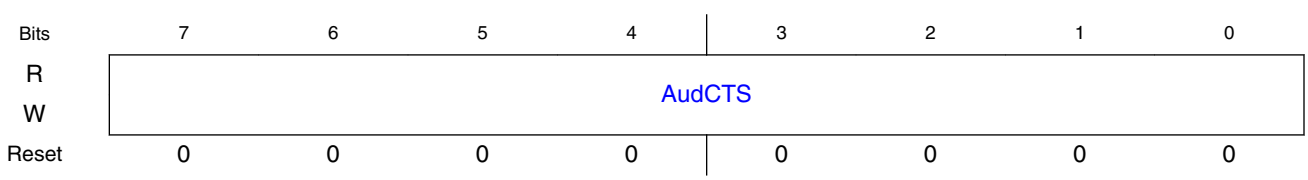
Register	Offset
aud_cts2	4h

13.9.3.7.6.2 Function

Audio Clock Regenerator CTS Register 2

For CTS expected values, refer to the HDMI 1.4b specification.

13.9.3.7.6.3 Diagram



13.9.3.7.6.4 Fields

Field	Function
7-0 AudCTS	HDMI Audio Clock Regenerator CTS calculated value. HDMI Audio Clock Regenerator CTS calculated value. This value can be manually set using the CTS_manual (AUD_CTS3) mechanism.

13.9.3.7.7 Audio Clock Regenerator CTS value Register 3. (aud_cts3)

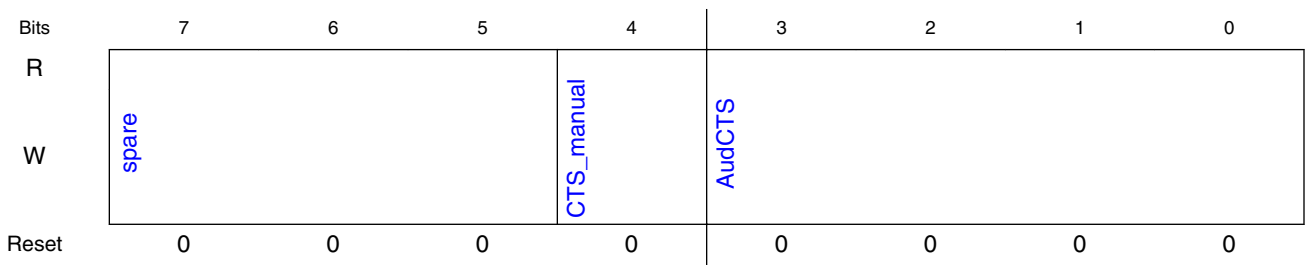
13.9.3.7.7.1 Offset

Register	Offset
aud_cts3	5h

13.9.3.7.7.2 Function

Audio Clock Regenerator CTS value Register 3. For CTS expected values, refer to the HDMI 1.4b specification.

13.9.3.7.7.3 Diagram



13.9.3.7.7.4 Fields

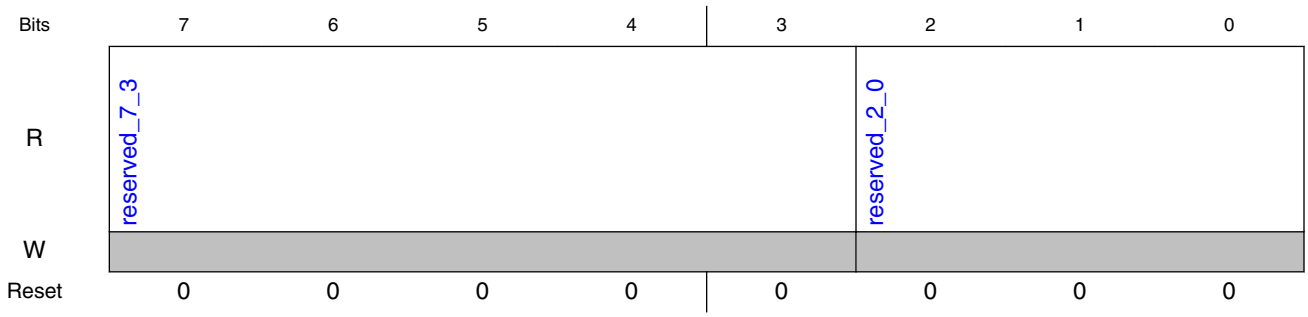
Field	Function
7-5 spare	Reserved as "spare" bit with no associated functionality.
4 CTS_manual	If the CTS_manual bit equals 0b, this registers contains audCTS[19:0] generated by the Cycle time counter according to the specified timing. If the CTS_manual bit equals 0b, this registers contains audCTS[19:0] generated by the Cycle time counter according to the specified timing. If the CTS_manual bit equals 1b, this register is configured with the audCTS[7:0] value that is output by the Audio Packetizer. Note: When the General Parallel Audio Interface (GPAUD) is enabled (AUDIO_IF = 6) or the AHB DMA Audio Interface is enabled (AUDIO_IF = 8), writing to these bits has no effect; reading these bits always return 0.
3-0 AudCTS	HDMI Audio Clock Regenerator CTS calculated value. HDMI Audio Clock Regenerator CTS calculated value. This value can be manually set using the CTS_manual (AUD_CTS3) mechanism.

13.9.3.7.8 Audio Input Clock FS Factor Register (aud_inputclkfs)

13.9.3.7.8.1 Offset

Register	Offset
aud_inputclkfs	6h

13.9.3.7.8.2 Diagram



13.9.3.7.8.3 Fields

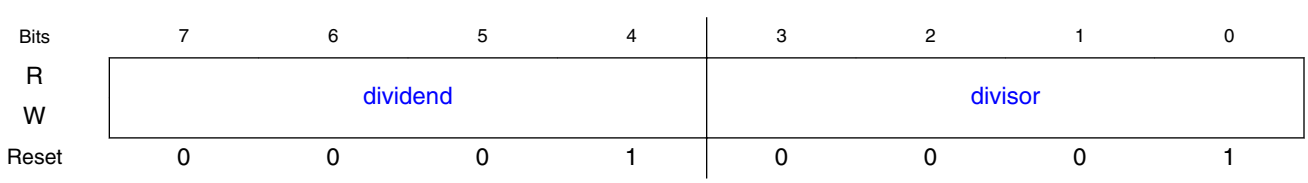
Field	Function
7-3 reserved_7_3	Reserved and read as zero
2-0 reserved_2_0	Reserved and read as zero

13.9.3.7.9 Audio CTS Dither Register (aud_cts_dither)

13.9.3.7.9.1 Offset

Register	Offset
aud_cts_dither	7h

13.9.3.7.9.2 Diagram



13.9.3.7.9.3 Fields

Field	Function
7-4 dividend	Dither dividend (4'd1 if no CTS Dither) This field should be configured with the value of dividend from the HDMI specification. Dither dividend (4'd1 if no CTS Dither) This field should be configured with the value of dividend from the HDMI specification. Note: This value is only meaningful when operating in N/CTS manual mode. Value 4'd0 is not supported.
3-0 divisor	Dither divisor (4'd1 if no CTS Dither) This field should be configured with the value of divisor from the HDMI specification. Dither divisor (4'd1 if no CTS Dither) This field should be configured with the value of divisor from the HDMI specification. Note: This value is only meaningful when operating in N/CTS manual mode. Value 4'd0 is not supported.

13.9.3.8 AudioSampleGP register descriptions

13.9.3.8.1 Memory map

AudioSampleGP registers base address: 3500h

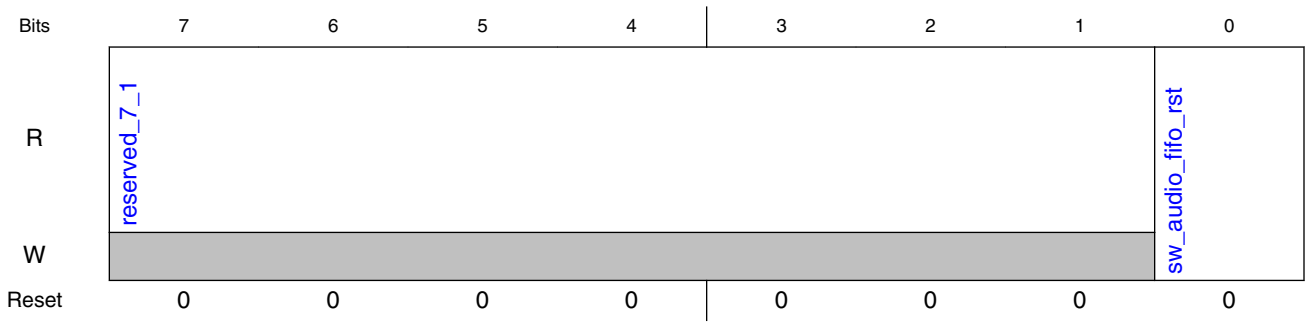
Offset	Register	Width (In bits)	Access	Reset value
0h	Audio GPA Software FIFO Reset Control Register 0 (gp_conf0)	8	RW	00h
1h	Audio GPA Channel Enable Configuration Register 1 (gp_conf1)	8	RW	00h
2h	Audio GPA HBR Enable Register 2 (gp_conf2)	8	RW	00h
6h	Audio GPA FIFO Full and Empty Mask Interrupt Register (gp_mask)	8	RW	10h

13.9.3.8.2 Audio GPA Software FIFO Reset Control Register 0 (gp_conf0)

13.9.3.8.2.1 Offset

Register	Offset
gp_conf0	0h

13.9.3.8.2.2 Diagram



13.9.3.8.2.3 Fields

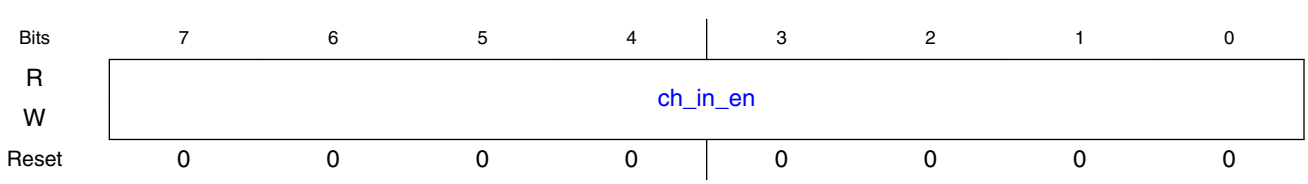
Field	Function
7-1 reserved_7_1	Reserved and read as zero
0 sw_audio_fifo_rst	Audio FIFOs software reset - Writing 0b: no action taken - Writing 1b: Resets all audio FIFOs Reading from this register always returns 0b. Audio FIFOs software reset - Writing 0b: no action taken - Writing 1b: Resets all audio FIFOs Reading from this register always returns 0b. Note: If a FIFO reset request (via register write command) lands in the middle of an GPAUD audio transaction, the samples become misaligned (left-right sequence lost). As a solution, for each FIFO reset, an associated SPDIF reset must be issued (writing 8'h7F to MC_SWRSTZ register).

13.9.3.8.3 Audio GPA Channel Enable Configuration Register 1 (gp_conf1)

13.9.3.8.3.1 Offset

Register	Offset
gp_conf1	1h

13.9.3.8.3.2 Diagram



13.9.3.8.3.3 Fields

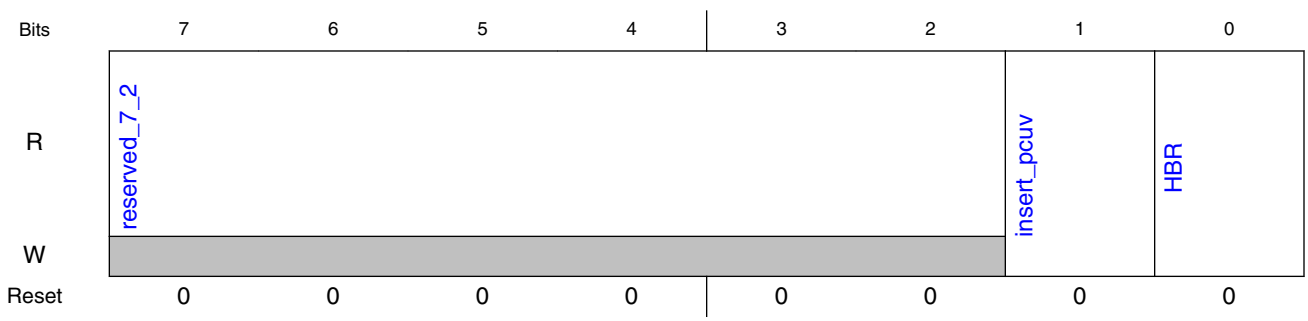
Field	Function
7-0 ch_in_en	Each bit controls the enabling of the respective audio channel. Each bit controls the enabling of the respective audio channel. For instance, bit 1, when set (1'b1), the audio Channel 1 is enabled. When cleared, the referred channel is disabled.

13.9.3.8.4 Audio GPA HBR Enable Register 2 (gp_conf2)

13.9.3.8.4.1 Offset

Register	Offset
gp_conf2	2h

13.9.3.8.4.2 Diagram



13.9.3.8.4.3 Fields

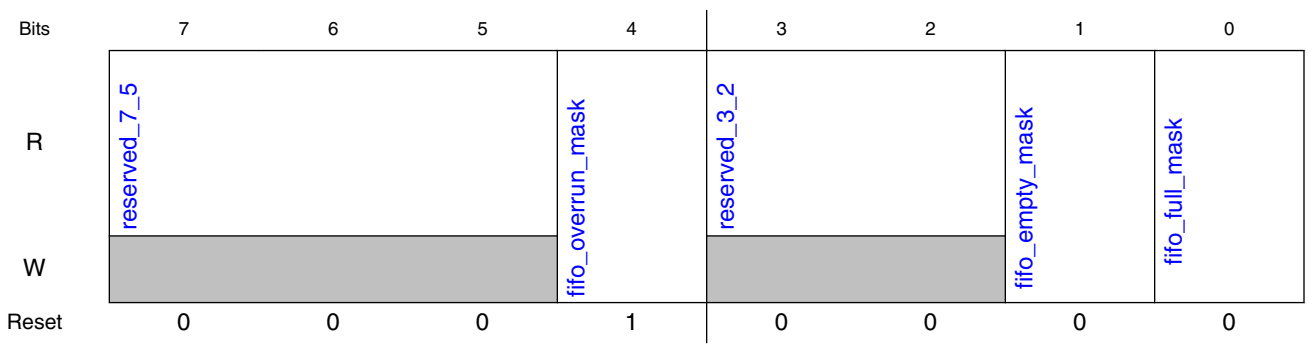
Field	Function
7-2 reserved_7_2	Reserved and read as zero
1 insert_pcuv	When set (1'b1), this bit enables the insertion of the PCUV (Parity, Channel Status, User bit and Validity) bits on the incoming audio stream (support limited to Linear PCM audio). When set (1'b1), this bit enables the insertion of the PCUV (Parity, Channel Status, User bit and Validity) bits on the incoming audio stream (support limited to Linear PCM audio). If disabled, the incoming audio stream must contain the PCUV bits, mapped according to 2.6.4.2 Data Mapping Examples.
0 HBR	HBR packets enable. HBR packets enable. The DWC_hdmi_tx sends the HBR packets. This bit is enabled when the audio frequency is higher than 192 kHz. If this bit is enabled, the number of channels configured in GP_CONF1 must be set to 8'hFF.

13.9.3.8.5 Audio GPA FIFO Full and Empty Mask Interrupt Register (gp_mask)

13.9.3.8.5.1 Offset

Register	Offset
gp_mask	6h

13.9.3.8.5.2 Diagram



13.9.3.8.5.3 Fields

Field	Function
7-5 reserved_7_5	Reserved and read as zero
4 fifo_overrun_mask	FIFO overrun mask
3-2 reserved_3_2	Reserved and read as zero
1 fifo_empty_mask	FIFO empty flag mask
0 fifo_full_mask	FIFO full flag mask

13.9.3.9 MainController register descriptions

13.9.3.9.1 Memory map

MainController registers base address: 4000h

Offset	Register	Width (In bits)	Access	Reset value
1h	Main Controller Synchronous Clock Domain Disable Register (mc_clkdis)	8	RW	Table 13-127
2h	Main Controller Software Reset Register Main controller software reset request per clock domain. (mc_swrstzreq_1)	8	RW	Table 13-127
4h	Main Controller Feed Through Control Register (mc_flowctrl)	8	RW	00h
5h	Main Controller PHY Reset Register (mc_phyrstz)	8	RW	00h
6h	Main Controller Clock Present Register (mc_lockonclock_1)	8	W1C	00h

13.9.3.9.2 Main Controller Synchronous Clock Domain Disable Register (mc_clkdis)

13.9.3.9.2.1 Offset

Register	Offset
mc_clkdis	1h

13.9.3.9.2.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	cecclk_disable	csccclk_disable	audclk_disable	prepclk_disable	tmdsclk_disable	pixelclk_disable
Reset	u	0	0	0	0	0	0	0

13.9.3.9.2.3 Fields

Field	Function
7 —	Reserved
6 —	Reserved
5 cecclk_disable	CEC Engine clock synchronous disable signal.
4 cscclk_disable	Color Space Converter clock synchronous disable signal.
3 audclk_disable	Audio Sampler clock synchronous disable signal.
2 prepclk_disable	Pixel Repetition clock synchronous disable signal.
1 tmdsclk_disable	TMDS clock synchronous disable signal. TMDS clock synchronous disable signal. It is required to perform a write action on one of the following registers: - fc_invidconf, fc_inhactiv0, fc_inhactiv1, fc_inhblank0, fc_inhblank1, fc_invactiv0 - fc_invactiv1, fc_invblank, fc_hsyncindelay0, fc_hsyncindelay1, fc_hsyncinwidth0 - fc_hsyncinwidth1, fc_vsyncindelay, fc_vsyncinwidth, fc_ctrldur, fc_exctrldur, fc_exctrlspac
0 pixelclk_disable	Pixel clock synchronous disable signal.

13.9.3.9.3 Main Controller Software Reset Register Main controller software reset request per clock domain. (mc_swrstzreq_1)

13.9.3.9.3.1 Offset

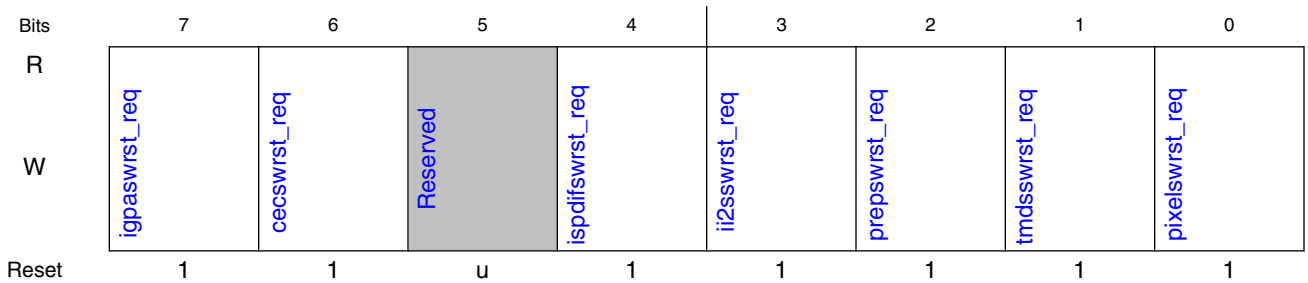
Register	Offset
mc_swrstzreq_1	2h

13.9.3.9.3.2 Function

Main Controller Software Reset Register

Main controller software reset request per clock domain. Writing zero to a bit of this register results in an NRZ signal toggle at sfrclk rate to an output signal that indicates a software reset request. This toggle must be used to generate a synchronized reset to de corresponding domain, with at least 1 clock cycle.

13.9.3.9.3.3 Diagram



13.9.3.9.3.4 Fields

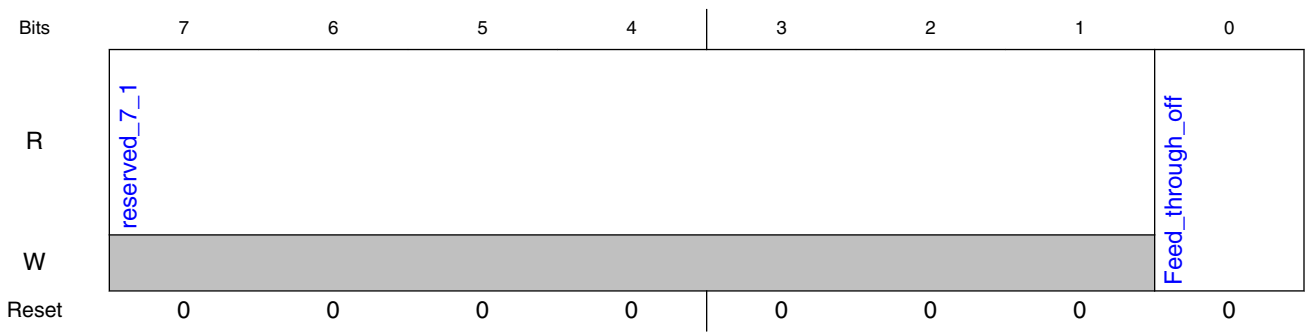
Field	Function
7 igpaswrst_req	GPAUD interface soft reset request. GPAUD interface soft reset request. This bit is enabled when the Generic Parallel Audio (GPAUD) interface is enabled (AUDIO_IF = 6). Otherwise, this bit returns zero.
6 cecswrst_req	CEC software reset request. CEC software reset request. Defaults back to 1b after reset request. Note: After you configure cecswrst_req, set the value of the bit csc_clk_disable of the register mc_clkdis to 1, 0, and then 1 again.
5 —	Reserved
4 ispdifswrst_req	SPDIF audio software reset request.
3 ii2sswrst_req	I2S audio software reset request.
2 prepswrst_req	Pixel Repetition software reset request.
1 tmdsswrst_req	TMDS software reset request. TMDS software reset request. It is required to perform a write action on one of the following registers: - fc_invidconf, fc_inhactiv0, fc_inhactiv1, fc_inhblank0, fc_inhblank1, fc_invactiv0 - fc_invactiv1, fc_invblank, fc_hsyncindelay0, fc_hsyncindelay1, fc_hsyncinwidth0 - fc_hsyncinwidth1, fc_vsyncindelay, fc_vsyncinwidth, fc_ctrldur, fc_exctrldur, fc_exctrlspac
0 pixelswrst_req	Pixel software reset request.

13.9.3.9.4 Main Controller Feed Through Control Register (mc_flowctrl)

13.9.3.9.4.1 Offset

Register	Offset
mc_flowctrl	4h

13.9.3.9.4.2 Diagram



13.9.3.9.4.3 Fields

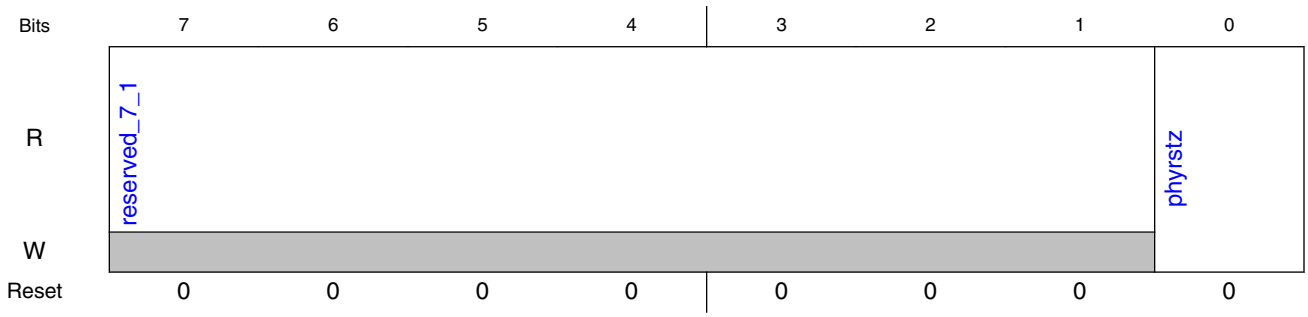
Field	Function
7-1 reserved_7_1	Reserved and read as zero
0 Feed_through_off	Video path Feed Through enable bit: - 1b: Color Space Converter is in the video data path. Video path Feed Through enable bit: - 1b: Color Space Converter is in the video data path. - 0b: Color Space Converter is bypassed (not in the video data path).

13.9.3.9.5 Main Controller PHY Reset Register (mc_phyrstz)

13.9.3.9.5.1 Offset

Register	Offset
mc_phyrstz	5h

13.9.3.9.5.2 Diagram



13.9.3.9.5.3 Fields

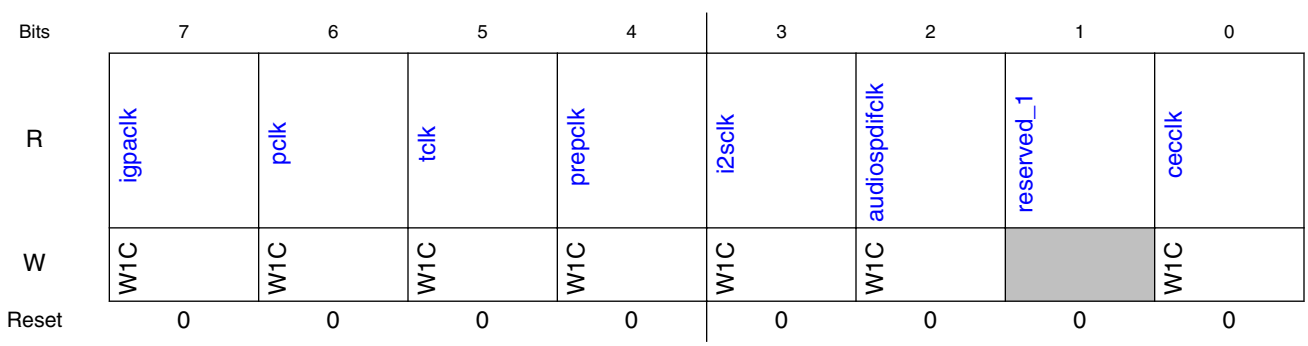
Field	Function
7-1 reserved_7_1	Reserved and read as zero
0 phyrstz	HDMI Source PHY active low reset control for PHY GEN1, active high reset control for PHY GEN2.

13.9.3.9.6 Main Controller Clock Present Register (mc_lockonclock_1)

13.9.3.9.6.1 Offset

Register	Offset
mc_lockonclock_1	6h

13.9.3.9.6.2 Diagram



13.9.3.9.6.3 Fields

Field	Function
7 igpaclk	GPAUD interface clock status. GPAUD interface clock status. This bit is enabled when the Generic Parallel Audio (GPAUD) interface is enabled (AUDIO_IF = 6). Otherwise, this bit returns zero. This bit indicates the clock is present in the system. It is cleared by writing 1 to this bit.
6 pclk	Pixel clock status. Pixel clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position.
5 tclk	TMDS clock status. TMDS clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position.
4 prepclk	Pixel Repetition clock status. Pixel Repetition clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position.
3 i2sclk	I2S clock status. I2S clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position.
2 audiospdifclk	SPDIF clock status. SPDIF clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position.
1 reserved_1	Reserved and read as zero
0 cecclk	CEC clock status. CEC clock status. Indicates that the clock is present in the system. Cleared by WR 1 to this position.

13.9.3.10 ColorSpaceConverter register descriptions

13.9.3.10.1 Memory map

ColorSpaceConverter registers base address: 4100h

Offset	Register	Width (In bits)	Access	Reset value
0h	Color Space Converter Interpolation and Decimation Configuration Register (csc_cfg)	8	RW	00h
1h	Color Space Converter Scale and Deep Color Configuration Register (csc_scale)	8	RW	01h
2h	Color Space Converter Matrix A1 Coefficient Register MSB Notes: - The coefficients used in the CSC matrix use only 15 bits for the internal computations. (csc_coef_a1_msb)	8	RW	20h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
3h	Color Space Converter Matrix A1 Coefficient Register LSB Notes: - The coefficients used in the CSC matrix use only 15 bits for the internal computations. (csc_coef_a1_lsb)	8	RW	00h
4h	Color Space Converter Matrix A2 Coefficient Register MSB Color Space Conversion A2 coefficient. (csc_coef_a2_msb)	8	RW	00h
5h	Color Space Converter Matrix A2 Coefficient Register LSB Color Space Conversion A2 coefficient. (csc_coef_a2_lsb)	8	RW	00h
6h	Color Space Converter Matrix A3 Coefficient Register MSB Color Space Conversion A3 coefficient. (csc_coef_a3_msb)	8	RW	00h
7h	Color Space Converter Matrix A3 Coefficient Register LSB Color Space Conversion A3 coefficient. (csc_coef_a3_lsb)	8	RW	00h
8h	Color Space Converter Matrix A4 Coefficient Register MSB Color Space Conversion A4 coefficient. (csc_coef_a4_msb)	8	RW	00h
9h	Color Space Converter Matrix A4 Coefficient Register LSB Color Space Conversion A4 coefficient. (csc_coef_a4_lsb)	8	RW	00h
Ah	Color Space Converter Matrix B1 Coefficient Register MSB Color Space Conversion B1 coefficient. (csc_coef_b1_msb)	8	RW	00h
Bh	Color Space Converter Matrix B1 Coefficient Register LSB Color Space Conversion B1 coefficient. (csc_coef_b1_lsb)	8	RW	00h
Ch	Color Space Converter Matrix B2 Coefficient Register MSB Color Space Conversion B2 coefficient. (csc_coef_b2_msb)	8	RW	20h
Dh	Color Space Converter Matrix B2 Coefficient Register LSB Color Space Conversion B2 coefficient. (csc_coef_b2_lsb)	8	RW	00h
Eh	Color Space Converter Matrix B3 Coefficient Register MSB Color Space Conversion B3 coefficient. (csc_coef_b3_msb)	8	RW	00h
Fh	Color Space Converter Matrix B3 Coefficient Register LSB Color Space Conversion B3 coefficient. (csc_coef_b3_lsb)	8	RW	00h
10h	Color Space Converter Matrix B4 Coefficient Register MSB Color Space Conversion B4 coefficient. (csc_coef_b4_msb)	8	RW	00h
11h	Color Space Converter Matrix B4 Coefficient Register LSB Color Space Conversion B4 coefficient. (csc_coef_b4_lsb)	8	RW	00h
12h	Color Space Converter Matrix C1 Coefficient Register MSB Color Space Conversion C1 coefficient. (csc_coef_c1_msb)	8	RW	00h
13h	Color Space Converter Matrix C1 Coefficient Register LSB Color Space Conversion C1 coefficient. (csc_coef_c1_lsb)	8	RW	00h
14h	Color Space Converter Matrix C2 Coefficient Register MSB Color Space Conversion C2 coefficient. (csc_coef_c2_msb)	8	RW	00h
15h	Color Space Converter Matrix C2 Coefficient Register LSB Color Space Conversion C2 coefficient. (csc_coef_c2_lsb)	8	RW	00h
16h	Color Space Converter Matrix C3 Coefficient Register MSB Color Space Conversion C3 coefficient. (csc_coef_c3_msb)	8	RW	20h
17h	Color Space Converter Matrix C3 Coefficient Register LSB Color Space Conversion C3 coefficient. (csc_coef_c3_lsb)	8	RW	00h
18h	Color Space Converter Matrix C4 Coefficient Register MSB Color Space Conversion C4 coefficient. (csc_coef_c4_msb)	8	RW	00h
19h	Color Space Converter Matrix C4 Coefficient Register LSB Color Space Conversion C4 coefficient. (csc_coef_c4_lsb)	8	RW	00h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1Ah	Color Space Converter Matrix Output Up Limit Register MSB (csc_limit_up_msb)	8	RW	FFh
1Bh	Color Space Converter Matrix output Up Limit Register LSB (csc_limit_up_lsb)	8	RW	FFh
1Ch	Color Space Converter Matrix output Down Limit Register MSB (csc_limit_dn_msb)	8	RW	00h
1Dh	Color Space Converter Matrix output Down Limit Register LSB (csc_limit_dn_lsb)	8	RW	00h

13.9.3.10.2 Color Space Converter Interpolation and Decimation Configuration Register (csc_cfg)

13.9.3.10.2.1 Offset

Register	Offset
csc_cfg	0h

13.9.3.10.2.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_limit	spare_2	intmode		spare_1		decmode	
Reset	0	0	0	0	0	0	0	0

13.9.3.10.2.3 Fields

Field	Function
7 csc_limit	When set (1'b1), the range limitation values defined in registers csc_mat_uplim and csc_mat_dnlim are applied to the output of the Color Space Conversion matrix. When set (1'b1), the range limitation values defined in registers csc_mat_uplim and csc_mat_dnlim are applied to the output of the Color Space Conversion matrix. This feature ensures that the video output range is always respected, independently of the matrix coefficient configuration or of the video input stream.
6 spare_2	Reserved as "spare" register with no associated functionality.

Table continues on the next page...

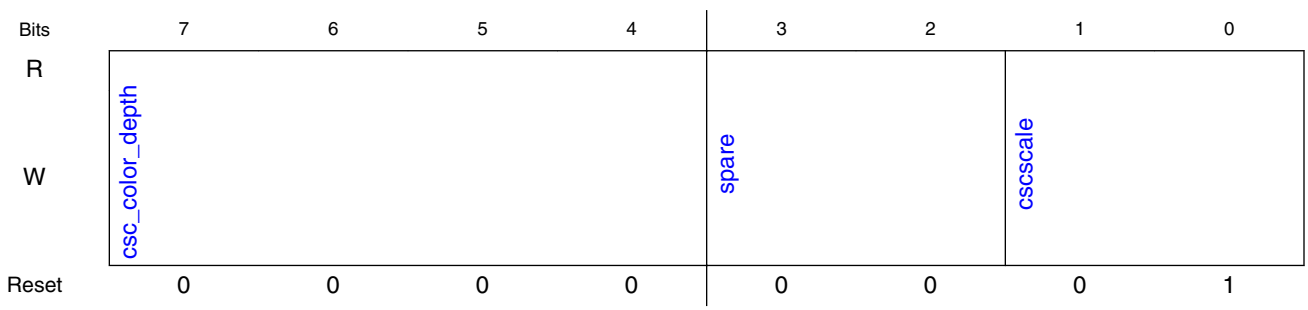
Field	Function
5-4 intmode	Chroma interpolation configuration: intmode[1:0] I Chroma Interpolation 00 I interpolation disabled 01 I Hu (z) =1 + z^(-1) 10 I Hu(z)=1/ 2 + z^(-11)+1/2 z^(-2) 11 I interpolation disabled
3-2 spare_1	Reserved as "spare" register with no associated functionality.
1-0 decmode	Chroma decimation configuration: decmode[1:0] I Chroma Decimation 00 I decimation disabled 01 I Hd (z) =1 10 I Hd(z)=1/ 4 + 1/2z^(-1)+1/4 z^(-2) 11 I Hd(z)x2^(11)= -5+12z^(-2) - 22z^(-4)+39z^(-8) +109z^(-10) -204z^(-12)+648z^(-14) + 1024z^(-15) +648z^(-16) -204z^(-18) +109z^(-20)- 65z^(-22) +39z^(-24) -22z^(-26) +12z^(-28)-5z^(-30)

13.9.3.10.3 Color Space Converter Scale and Deep Color Configuration Register (csc_scale)

13.9.3.10.3.1 Offset

Register	Offset
csc_scale	1h

13.9.3.10.3.2 Diagram



13.9.3.10.3.3 Fields

Field	Function
7-4 csc_color_depth	Color space converter color depth configuration: csc_colordepth[3:0] I Action 0000 I 24 bit per pixel video (8 bit per component). Color space converter color depth configuration: csc_colordepth[3:0] I Action 0000 I 24 bit per pixel video (8 bit per component). 0001-0011 I Reserved. Not used. 0100 I 24 bit per pixel video (8 bit per component).

Table continues on the next page...

Field	Function
	0101 | 30 bit per pixel video (10 bit per component). 0110 | 36 bit per pixel video (12 bit per component). 0111 | 48 bit per pixel video (16 bit per component). other | Reserved. Not used.
3-2 spare	The is a Reserved as "spare" register with no associated functionality.
1-0 cscscale	Defines the cscscale[1:0] scale factor to apply to all coefficients in Color Space Conversion. Defines the cscscale[1:0] scale factor to apply to all coefficients in Color Space Conversion. This scale factor is expressed in the number of left shifts to apply to each of the coefficients, ranging from 0 to 2.

13.9.3.10.4 Color Space Converter Matrix A1 Coefficient Register MSB Notes: - The coefficients used in the CSC matrix use only 15 bits for the internal computations. (csc_coef_a1_msb)

13.9.3.10.4.1 Offset

Register	Offset
csc_coef_a1_msb	2h

13.9.3.10.4.2 Function

Color Space Converter Matrix A1 Coefficient Register MSB

Notes:

- The coefficients used in the CSC matrix use only 15 bits for the internal computations.

- Coefficients are represented in 2's complementary format and stored in two registers:

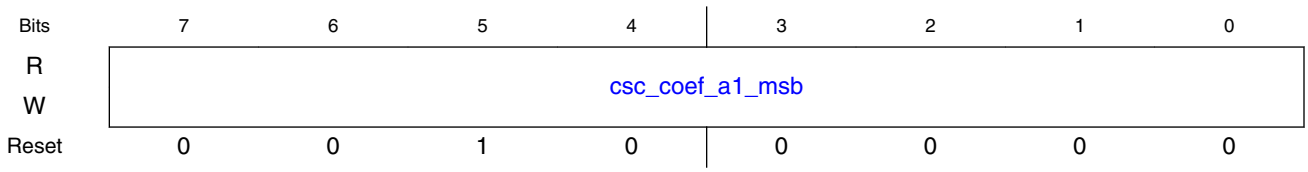
- csc_coef_*_lsb[7:0]: coefficient bits 7 to 0

- csc_coef_*_msb[7]: spare bit

- csc_coef_*_msb[6:0]: coefficient bits 14 to 8

- Examples for standard ITU601 and ITU709 RGB/YCbCr conversion CSC coefficients exist in the Video Datapath Application Note (available from the Synopsys web site).

13.9.3.10.4.3 Diagram



13.9.3.10.4.4 Fields

Field	Function
7-0 csc_coef_a1_msb	Color Space Converter Matrix A1 Coefficient Register MSB

13.9.3.10.5 Color Space Converter Matrix A1 Coefficient Register LSB Notes: - The coefficients used in the CSC matrix use only 15 bits for the internal computations. (csc_coef_a1_lsb)

13.9.3.10.5.1 Offset

Register	Offset
csc_coef_a1_lsb	3h

13.9.3.10.5.2 Function

Color Space Converter Matrix A1 Coefficient Register LSB

Notes:

- The coefficients used in the CSC matrix use only 15 bits for the internal computations.

- Coefficients are represented in 2's complementary format and stored in two registers:

- - csc_coef_*_lsb[7:0]: coefficient bits 7 to 0

- - csc_coef_*_msb[7]: spare bit

- - csc_coef_*_msb[6:0]: coefficient bits 14 to 8

- Examples for standard ITU601 and ITU709 RGB/YCbCr conversion CSC coefficients exist in the Video Datapath Application Note (available from the Synopsys web site).

13.9.3.10.5.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_a1_lsb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.5.4 Fields

Field	Function
7-0 csc_coef_a1_lsb	Color Space Converter Matrix A1 Coefficient Register LSB

13.9.3.10.6 Color Space Converter Matrix A2 Coefficient Register MSB Color Space Conversion A2 coefficient. (csc_coef_a2_msb)

13.9.3.10.6.1 Offset

Register	Offset
csc_coef_a2_msb	4h

13.9.3.10.6.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_a2_msb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.6.3 Fields

Field	Function
7-0 csc_coef_a2_msb	Color Space Converter Matrix A2 Coefficient Register MSB

13.9.3.10.7 Color Space Converter Matrix A2 Coefficient Register LSB Color Space Conversion A2 coefficient. (csc_coef_a2_lsb)

13.9.3.10.7.1 Offset

Register	Offset
csc_coef_a2_lsb	5h

13.9.3.10.7.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_a2_lsb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.7.3 Fields

Field	Function
7-0 csc_coef_a2_lsb	Color Space Converter Matrix A2 Coefficient Register LSB

13.9.3.10.8 Color Space Converter Matrix A3 Coefficient Register MSB Color Space Conversion A3 coefficient. (csc_coef_a3_msb)

13.9.3.10.8.1 Offset

Register	Offset
csc_coef_a3_msb	6h

13.9.3.10.8.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_a3_msb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.8.3 Fields

Field	Function
7-0 csc_coef_a3_msb	Color Space Converter Matrix A3 Coefficient Register MSB

13.9.3.10.9 Color Space Converter Matrix A3 Coefficient Register LSB Color Space Conversion A3 coefficient. (csc_coef_a3_lsb)

13.9.3.10.9.1 Offset

Register	Offset
csc_coef_a3_lsb	7h

13.9.3.10.9.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_a3_lsb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.9.3 Fields

Field	Function
7-0 csc_coef_a3_lsb	Color Space Converter Matrix A3 Coefficient Register LSB

13.9.3.10.10 Color Space Converter Matrix A4 Coefficient Register MSB Color Space Conversion A4 coefficient. (csc_coef_a4_msb)

13.9.3.10.10.1 Offset

Register	Offset
csc_coef_a4_msb	8h

13.9.3.10.10.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_a4_msb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.10.3 Fields

Field	Function
7-0 csc_coef_a4_msb	Color Space Converter Matrix A4 Coefficient Register MSB

13.9.3.10.11 Color Space Converter Matrix A4 Coefficient Register LSB Color Space Conversion A4 coefficient. (csc_coef_a4_lsb)

13.9.3.10.11.1 Offset

Register	Offset
csc_coef_a4_lsb	9h

13.9.3.10.11.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_a4_lsb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.11.3 Fields

Field	Function
7-0 csc_coef_a4_lsb	Color Space Converter Matrix A4 Coefficient Register LSB

13.9.3.10.12 Color Space Converter Matrix B1 Coefficient Register MSB Color Space Conversion B1 coefficient. (csc_coef_b1_msb)

13.9.3.10.12.1 Offset

Register	Offset
csc_coef_b1_msb	Ah

13.9.3.10.12.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_b1_msb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.12.3 Fields

Field	Function
7-0 csc_coef_b1_msb	Color Space Converter Matrix B1 Coefficient Register MSB

13.9.3.10.13 Color Space Converter Matrix B1 Coefficient Register LSB Color Space Conversion B1 coefficient. (csc_coef_b1_lsb)

13.9.3.10.13.1 Offset

Register	Offset
csc_coef_b1_lsb	Bh

13.9.3.10.13.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_b1_lsb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.13.3 Fields

Field	Function
7-0 csc_coef_b1_lsb	Color Space Converter Matrix B1 Coefficient Register LSB

13.9.3.10.14 Color Space Converter Matrix B2 Coefficient Register MSB Color Space Conversion B2 coefficient. (csc_coef_b2_msb)

13.9.3.10.14.1 Offset

Register	Offset
csc_coef_b2_msb	Ch

13.9.3.10.14.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_b2_msb							
Reset	0	0	1	0	0	0	0	0

13.9.3.10.14.3 Fields

Field	Function
7-0 csc_coef_b2_msb	Color Space Converter Matrix B2 Coefficient Register MSB

13.9.3.10.15 Color Space Converter Matrix B2 Coefficient Register LSB Color Space Conversion B2 coefficient. (csc_coef_b2_lsb)

13.9.3.10.15.1 Offset

Register	Offset
csc_coef_b2_lsb	Dh

13.9.3.10.15.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_b2_lsb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.15.3 Fields

Field	Function
7-0 csc_coef_b2_lsb	Color Space Converter Matrix B2 Coefficient Register LSB

13.9.3.10.16 Color Space Converter Matrix B3 Coefficient Register MSB Color Space Conversion B3 coefficient. (csc_coef_b3_msb)

13.9.3.10.16.1 Offset

Register	Offset
csc_coef_b3_msb	Eh

13.9.3.10.16.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_b3_msb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.16.3 Fields

Field	Function
7-0 csc_coef_b3_msb	Color Space Converter Matrix B3 Coefficient Register MSB

13.9.3.10.17 Color Space Converter Matrix B3 Coefficient Register LSB Color Space Conversion B3 coefficient. (csc_coef_b3_lsb)

13.9.3.10.17.1 Offset

Register	Offset
csc_coef_b3_lsb	Fh

13.9.3.10.17.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_b3_lsb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.17.3 Fields

Field	Function
7-0 csc_coef_b3_lsb	Color Space Converter Matrix B3 Coefficient Register LSB

13.9.3.10.18 Color Space Converter Matrix B4 Coefficient Register MSB Color Space Conversion B4 coefficient. (csc_coef_b4_msb)

13.9.3.10.18.1 Offset

Register	Offset
csc_coef_b4_msb	10h

13.9.3.10.18.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_b4_msb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.18.3 Fields

Field	Function
7-0 csc_coef_b4_msb	Color Space Converter Matrix B4 Coefficient Register MSB

13.9.3.10.19 Color Space Converter Matrix B4 Coefficient Register LSB Color Space Conversion B4 coefficient. (csc_coef_b4_lsb)

13.9.3.10.19.1 Offset

Register	Offset
csc_coef_b4_lsb	11h

13.9.3.10.19.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_b4_lsb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.19.3 Fields

Field	Function
7-0 csc_coef_b4_lsb	Color Space Converter Matrix B4 Coefficient Register LSB

13.9.3.10.20 Color Space Converter Matrix C1 Coefficient Register MSB Color Space Conversion C1 coefficient. (csc_coef_c1_msb)

13.9.3.10.20.1 Offset

Register	Offset
csc_coef_c1_msb	12h

13.9.3.10.20.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_c1_msb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.20.3 Fields

Field	Function
7-0 csc_coef_c1_msb	Color Space Converter Matrix C1 Coefficient Register MSB

13.9.3.10.21 Color Space Converter Matrix C1 Coefficient Register LSB Color Space Conversion C1 coefficient. (csc_coef_c1_lsb)

13.9.3.10.21.1 Offset

Register	Offset
csc_coef_c1_lsb	13h

13.9.3.10.21.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_c1_lsb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.21.3 Fields

Field	Function
7-0 csc_coef_c1_lsb	Color Space Converter Matrix C1 Coefficient Register LSB

13.9.3.10.22 Color Space Converter Matrix C2 Coefficient Register MSB Color Space Conversion C2 coefficient. (csc_coef_c2_msb)

13.9.3.10.22.1 Offset

Register	Offset
csc_coef_c2_msb	14h

13.9.3.10.22.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_c2_msb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.22.3 Fields

Field	Function
7-0 csc_coef_c2_msb	Color Space Converter Matrix C2 Coefficient Register MSB

13.9.3.10.23 Color Space Converter Matrix C2 Coefficient Register LSB Color Space Conversion C2 coefficient. (csc_coef_c2_lsb)

13.9.3.10.23.1 Offset

Register	Offset
csc_coef_c2_lsb	15h

13.9.3.10.23.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_c2_lsb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.23.3 Fields

Field	Function
7-0 csc_coef_c2_lsb	Color Space Converter Matrix C2 Coefficient Register LSB

13.9.3.10.24 Color Space Converter Matrix C3 Coefficient Register MSB Color Space Conversion C3 coefficient. (csc_coef_c3_msb)

13.9.3.10.24.1 Offset

Register	Offset
csc_coef_c3_msb	16h

13.9.3.10.24.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_c3_msb							
Reset	0	0	1	0	0	0	0	0

13.9.3.10.24.3 Fields

Field	Function
7-0 csc_coef_c3_msb	Color Space Converter Matrix C3 Coefficient Register MSB

13.9.3.10.25 Color Space Converter Matrix C3 Coefficient Register LSB Color Space Conversion C3 coefficient. (csc_coef_c3_lsb)

13.9.3.10.25.1 Offset

Register	Offset
csc_coef_c3_lsb	17h

13.9.3.10.25.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_c3_lsb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.25.3 Fields

Field	Function
7-0 csc_coef_c3_lsb	Color Space Converter Matrix C3 Coefficient Register LSB

13.9.3.10.26 Color Space Converter Matrix C4 Coefficient Register MSB Color Space Conversion C4 coefficient. (csc_coef_c4_msb)

13.9.3.10.26.1 Offset

Register	Offset
csc_coef_c4_msb	18h

13.9.3.10.26.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_c4_msb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.26.3 Fields

Field	Function
7-0 csc_coef_c4_msb	Color Space Converter Matrix C4 Coefficient Register MSB

13.9.3.10.27 Color Space Converter Matrix C4 Coefficient Register LSB Color Space Conversion C4 coefficient. (csc_coef_c4_lsb)

13.9.3.10.27.1 Offset

Register	Offset
csc_coef_c4_lsb	19h

13.9.3.10.27.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_coef_c4_lsb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.27.3 Fields

Field	Function
7-0 csc_coef_c4_lsb	Color Space Converter Matrix C4 Coefficient Register LSB

13.9.3.10.28 Color Space Converter Matrix Output Up Limit Register MSB For more details, refer to the HDMI 1. (csc_limit_up_msb)

13.9.3.10.28.1 Offset

Register	Offset
csc_limit_up_msb	1Ah

13.9.3.10.28.2 Function

Color Space Converter Matrix Output Up Limit Register MSB

For more details, refer to the HDMI 1.4 specification, paragraph 6.6 Video Quantization Ranges. For an RGB output of 8 bits, the expected limit is 254 (valid range column of Table 6-3), and this register must be configured with 0x00.

13.9.3.10.28.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_limit_up_msb							
Reset	1	1	1	1	1	1	1	1

13.9.3.10.28.4 Fields

Field	Function
7-0 csc_limit_up_msb	Color Space Converter Matrix Output Upper Limit Register MSB

13.9.3.10.29 Color Space Converter Matrix output Up Limit Register LSB For more details, refer to the HDMI 1. (csc_limit_up_lsb)

13.9.3.10.29.1 Offset

Register	Offset
csc_limit_up_lsb	1Bh

13.9.3.10.29.2 Function

Color Space Converter Matrix output Up Limit Register LSB

For more details, refer to the HDMI 1.4 specification, paragraph 6.6 Video Quantization Ranges. For an RGB output of 8 bits, the expected limit is 254 (valid range column of Table 6-3), and this register must be configured with 0xFE.

13.9.3.10.29.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_limit_up_lsb							
Reset	1	1	1	1	1	1	1	1

13.9.3.10.29.4 Fields

Field	Function
7-0 csc_limit_up_lsb	Color Space Converter Matrix Output Upper Limit Register LSB

13.9.3.10.30 Color Space Converter Matrix output Down Limit Register MSB For more details, refer to the HDMI 1. (csc_limit_dn_msb)

13.9.3.10.30.1 Offset

Register	Offset
csc_limit_dn_msb	1Ch

13.9.3.10.30.2 Function

Color Space Converter Matrix output Down Limit Register MSB

For more details, refer to the HDMI 1.4 specification, paragraph 6.6 Video Quantization Ranges. For an RGB output of 8 bits, the expected limit is 1 (valid range column of Table 6-3), and this register must be configured with 0x00.

13.9.3.10.30.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_limit_dn_msb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.30.4 Fields

Field	Function
7-0 csc_limit_dn_msb	Color Space Converter Matrix output Down Limit Register MSB

13.9.3.10.31 Color Space Converter Matrix output Down Limit Register LSB For more details, refer to the HDMI 1. (csc_limit_dn_lsb)

13.9.3.10.31.1 Offset

Register	Offset
csc_limit_dn_lsb	1Dh

13.9.3.10.31.2 Function

Color Space Converter Matrix output Down Limit Register LSB

For more details, refer to the HDMI 1.4 specification, paragraph 6.6 Video Quantization Ranges. For an RGB output of 8 bits, the expected limit is 1 (valid range column of Table 6-3), and this register must be configured with 0x01.

13.9.3.10.31.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	csc_limit_dn_lsb							
Reset	0	0	0	0	0	0	0	0

13.9.3.10.31.4 Fields

Field	Function
7-0 csc_limit_dn_lsb	Color Space Converter Matrix Output Down Limit Register LSB

13.9.3.11 CEC register descriptions

13.9.3.11.1 Memory map

CEC registers base address: 7D00h

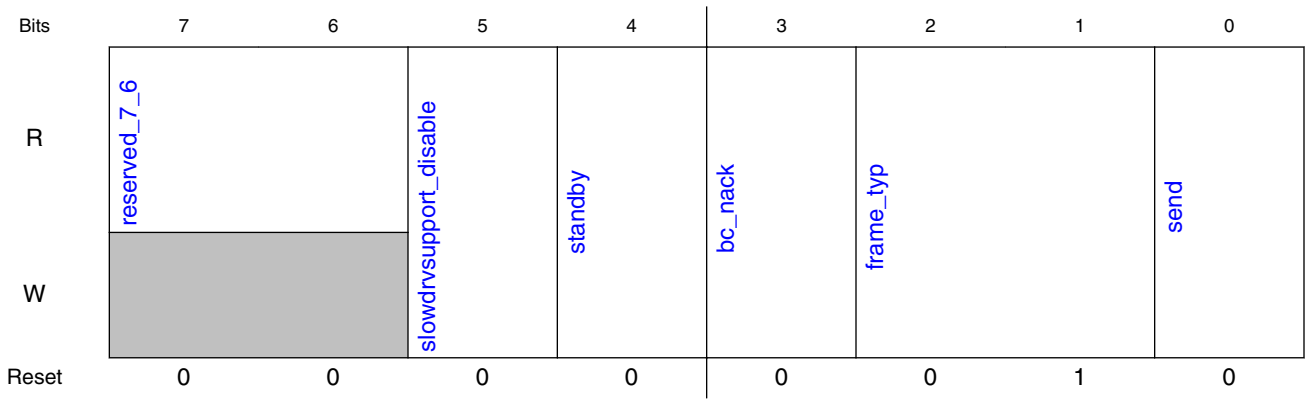
Offset	Register	Width (In bits)	Access	Reset value
0h	CEC Control Register This register handles the main control of the CEC initiator. (cec_ctrl)	8	RW	02h
2h	CEC Interrupt Mask Register This read/write register masks/unmasks the interrupt events. (cec_mask)	8	RW	00h
5h	CEC Logical Address Register Low This register indicates the logical address(es) allocated to the CEC device. (cec_addr_l)	8	RW	00h
6h	CEC Logical Address Register High This register indicates the logical address(es) allocated to the CEC device. (cec_addr_h)	8	RW	80h
7h	CEC TX Frame Size Register This register indicates the size of the frame in bytes (including header and data blocks), which are available in the transmitter data buffer. (cec_tx_cnt)	8	RW	00h
8h	CEC RX Frame Size Register This register indicates the size of the frame in bytes (including header and data blocks), which are available in the receiver data buffer. (cec_rx_cnt)	8	RO	00h
30h	CEC Buffer Lock Register (cec_lock)	8	RW	00h
31h	CEC Wake-up Control Register After receiving a message in the CEC_RX_DATA1 (OPCODE) registers, the CEC engine verifies the message opcode[7:0] against one of the previously defined values to generate the wake-up status: Wakeupstatus is 1 when: received opcode is 0x04 and opcode0x04en is 1 or received opcode is 0x0D and opcode0x0Den is 1 or received opcode is 0x41 and opcode0x41en is 1 or received opcode is 0x42 and opcode0x42en is 1 or received opcode is 0x44 and opcode0x44en is 1 or received opcode is 0x70 and opcode0x70en is 1 or received opcode is 0x82 and opcode0x82en is 1 or received opcode is 0x86 and opcode0x86en is 1 Wakeupstatus is 0 when none of the previous conditions are true. (cec_wakeupctrl)	8	RW	FFh

13.9.3.11.2 CEC Control Register This register handles the main control of the CEC initiator. (cec_ctrl)

13.9.3.11.2.1 Offset

Register	Offset
cec_ctrl	0h

13.9.3.11.2.2 Diagram



13.9.3.11.2.3 Fields

Field	Function
7-6 reserved_7_6	Reserved and read as zero
5 slowdrvsupport_disable	- 1: CEC controller supports IO drivers with risetime of up to 1 CEC clock period. - 1: CEC controller supports IO drivers with risetime of up to 1 CEC clock period. - 0: CEC controller supports IO drivers with risetime greater than 1 CEC clock period and where the IO driver risetime-falltime < 170us .
4 standby	- 1: CEC controller responds with a NACK to all messages and generates a wakeup status for opcode. - 1: CEC controller responds with a NACK to all messages and generates a wakeup status for opcode. It only responds with a NACK when the EOM is received. This means only the last block of a frame responds with NACK. The follower sends an ACK to the message when there is only one head block pointed to the follower, if the follower is in the standby mode. - 0: CEC controller responds the ACK to all messages.
3 bc_nack	- 1'b1: Set by software to NACK the received broadcast message. - 1'b1: Set by software to NACK the received broadcast message. This bit holds until software resets. The broadcasts is answered with 1'b0, indicating the follower reject the message. - 1'b0: Reset by software to ACK the received broadcast message.
2-1 frame_typ	- 2'b00: Signal Free Time = 3-bit periods. - 2'b00: Signal Free Time = 3-bit periods. Previous attempt to send frame is unsuccessful. - 2'b01: Signal Free Time = 5-bit periods. New initiator wants to send a frame. - 2'b10: Signal Free Time = 7-bit periods. Present initiator wants to send another frame immediately after its previous frame. (specification CEC 9.1). - 2'b11: Illegal value. If software writes this value, hardware sets the value to the default 2'b01.
0 send	- 1'b1: Set by software to trigger CEC sending a frame as an initiator. - 1'b1: Set by software to trigger CEC sending a frame as an initiator. This bit keeps at 1'b1 while the transmission is going on. - 1'b0: Reset to 1'b0 by hardware when the CEC transmission is done (no matter successful or failed). It can also work as an indicator checked by software to see whether the transmission is finished.

13.9.3.11.3 CEC Interrupt Mask Register This read/write register masks/unmasks the interrupt events. (cec_mask)

13.9.3.11.3.1 Offset

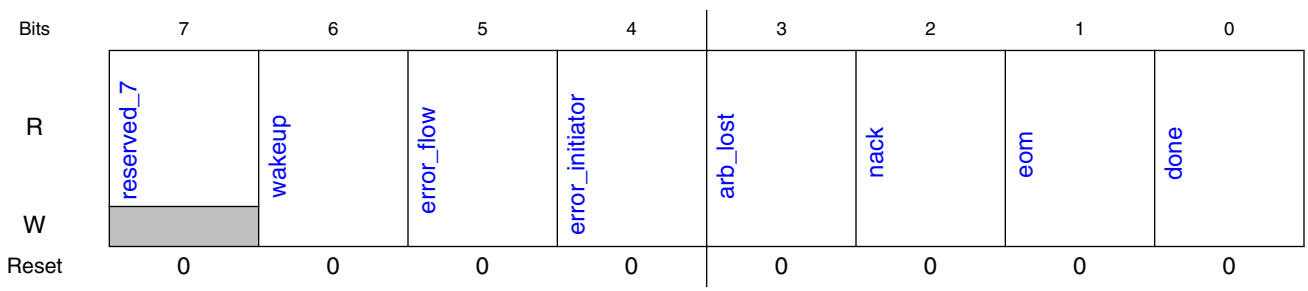
Register	Offset
cec_mask	2h

13.9.3.11.3.2 Function

CEC Interrupt Mask Register

This read/write register masks/unmasks the interrupt events. When the bit is set to 1 (masked), the corresponding event does not trigger an interrupt signal at the system interface. When the bit is reset to 0, the interrupt event is unmasked.

13.9.3.11.3.3 Diagram



13.9.3.11.3.4 Fields

Field	Function
7 reserved_7	Reserved and read as zero
6 wakeup	Follower wake-up signal mask
5 error_flow	An error is notified by a follower. An error is notified by a follower. Abnormal logic data bit error (for follower)
4 error_initiator	An error is detected on a CEC line (for initiator only).
3	The initiator losses the CEC line arbitration to a second initiator.

Table continues on the next page...

Field	Function
arb_lost	The initiator losses the CEC line arbitration to a second initiator. (specification CEC 9)
2 nack	A frame is not acknowledged in a directly addressed message. A frame is not acknowledged in a directly addressed message. Or a frame is negatively acknowledged in a broadcast message (for initiator only)
1 eom	EOM is detected so that the received data is ready in the receiver data buffer (for follower only)
0 done	The current transmission is successful (for initiator only)

13.9.3.11.4 CEC Logical Address Register Low This register indicates the logical address(es) allocated to the CEC device. (cec_addr_l)

13.9.3.11.4.1 Offset

Register	Offset
cec_addr_l	5h

13.9.3.11.4.2 Function

CEC Logical Address Register Low

This register indicates the logical address(es) allocated to the CEC device.

This register is written by software when the logical allocation is finished. Bit value 1 means the corresponding logical address is allocated to this device. Bit value 0 means the corresponding logical address is not allocated to this device.

13.9.3.11.4.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	cec_addr_l_7	cec_addr_l_6	cec_addr_l_5	cec_addr_l_4	cec_addr_l_3	cec_addr_l_2	cec_addr_l_1	cec_addr_l_0
Reset	0	0	0	0	0	0	0	0

13.9.3.11.4.4 Fields

Field	Function
7 cec_addr_l_7	Logical address 7 - Tuner 3
6 cec_addr_l_6	Logical address 6 - Tuner 2
5 cec_addr_l_5	Logical address 5 - Audio System
4 cec_addr_l_4	Logical address 4 - Playback Device 1
3 cec_addr_l_3	Logical address 3 - Tuner 1
2 cec_addr_l_2	Logical address 2 - Recording Device 2
1 cec_addr_l_1	Logical address 1 - Recording Device 1
0 cec_addr_l_0	Logical address 0 - Device TV

13.9.3.11.5 CEC Logical Address Register High This register indicates the logical address(es) allocated to the CEC device. (cec_addr_h)

13.9.3.11.5.1 Offset

Register	Offset
cec_addr_h	6h

13.9.3.11.5.2 Function

CEC Logical Address Register High

This register indicates the logical address(es) allocated to the CEC device.

This register is written by software when the logical allocation is finished. Bit value 1 means the corresponding logical address is allocated to this device. Bit value 0 means the corresponding logical address is not allocated to this device.

13.9.3.11.5.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	cec_addr_h_7	cec_addr_h_6	cec_addr_h_5	cec_addr_h_4	cec_addr_h_3	cec_addr_h_2	cec_addr_h_1	cec_addr_h_0
Reset	1	0	0	0	0	0	0	0

13.9.3.11.5.4 Fields

Field	Function
7 cec_addr_h_7	Logical address 15 - Unregistered (as initiator address), Broadcast (as destination address)
6 cec_addr_h_6	Logical address 14 - Free use
5 cec_addr_h_5	Logical address 13 - Reserved
4 cec_addr_h_4	Logical address 12 - Reserved
3 cec_addr_h_3	Logical address 11 - Playback Device 3
2 cec_addr_h_2	Logical address 10 - Tuner 4
1 cec_addr_h_1	Logical address 9 - Playback Device 3
0 cec_addr_h_0	Logical address 8 - Playback Device 2

13.9.3.11.6 CEC TX Frame Size Register This register indicates the size of the frame in bytes (including header and data blocks), which are available in the transmitter data buffer. (cec_tx_cnt)

13.9.3.11.6.1 Offset

Register	Offset
cec_tx_cnt	7h

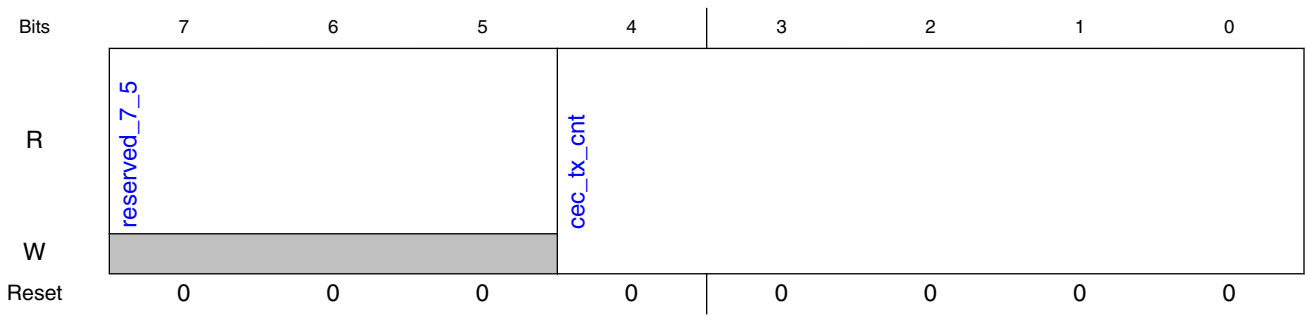
13.9.3.11.6.2 Function

CEC TX Frame Size Register

This register indicates the size of the frame in bytes (including header and data blocks), which are available in the transmitter data buffer.

Note: When the value is zero, the CEC controller ignores the send command triggered by software. When the transmission is done (no matter success or not), the current value is held until it is overwritten by software.

13.9.3.11.6.3 Diagram



13.9.3.11.6.4 Fields

Field	Function
7-5 reserved_7_5	Reserved and read as zero
4-0 cec_tx_cnt	CEC Transmitter Counter register 5'd0: No data needs to be transmitted 5'd1: Frame size is 1 byte . CEC Transmitter Counter register 5'd0: No data needs to be transmitted 5'd1: Frame size is 1 byte ... 5'd16: Frame size is 16 bytes

13.9.3.11.7 CEC RX Frame Size Register This register indicates the size of the frame in bytes (including header and data blocks), which are available in the receiver data buffer. (cec_rx_cnt)

13.9.3.11.7.1 Offset

Register	Offset
cec_rx_cnt	8h

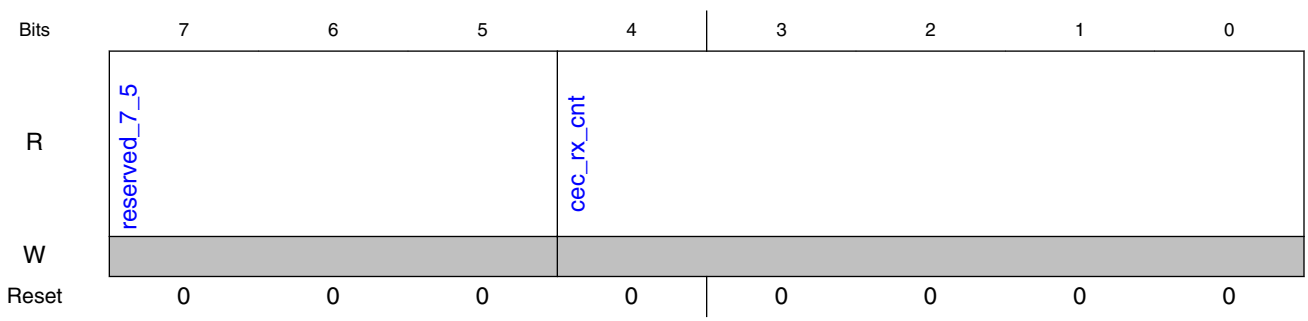
13.9.3.11.7.2 Function

CEC RX Frame Size Register

This register indicates the size of the frame in bytes (including header and data blocks), which are available in the receiver data buffer.

Note: Only after the whole receiving process is finished successfully, the counter is refreshed to the value which indicates the total number of data bytes in the Receiver Data Register.

13.9.3.11.7.3 Diagram



13.9.3.11.7.4 Fields

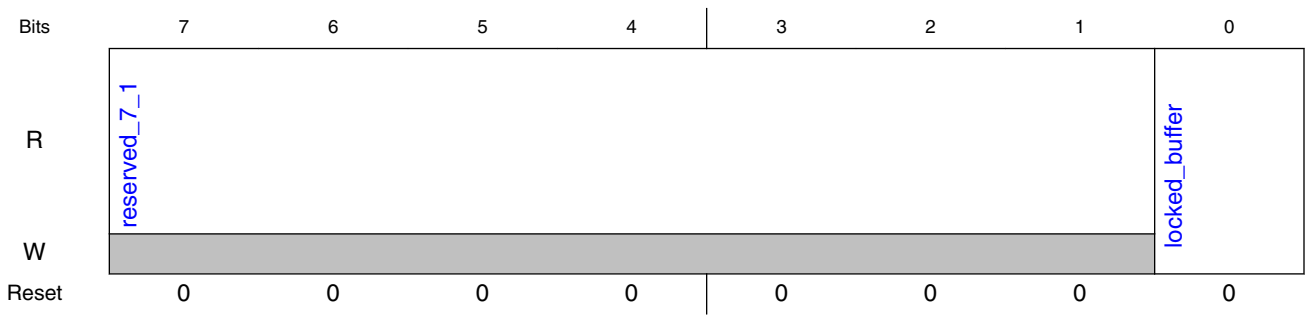
Field	Function
7-5 reserved_7_5	Reserved and read as zero
4-0 cec_rx_cnt	CEC Receiver Counter register: 5'd0: No data received 5'd1: 1-byte data is received . CEC Receiver Counter register: 5'd0: No data received 5'd1: 1-byte data is received ... 5'd16: 16-byte data is received

13.9.3.11.8 CEC Buffer Lock Register (cec_lock)

13.9.3.11.8.1 Offset

Register	Offset
cec_lock	30h

13.9.3.11.8.2 Diagram



13.9.3.11.8.3 Fields

Field	Function
7-1 reserved_7_1	Reserved and read as zero
0 locked_buffer	When a frame is received, this bit would be active. When a frame is received, this bit would be active. The CEC controller answers to all the messages with NACK until the CPU writes it to '0'.

13.9.3.11.9 CEC Wake-up Control Register After receiving a message in the CEC_RX_DATA1 (OPCODE) registers, the CEC engine verifies the message opcode[7:0] against one of the previously defined values to generate the wake-up status: Wakeupstatus is 1 when: received opcode is 0x04 and opcode0x04en is 1 or received opcode is 0x0D and opcode0x0Den is 1 or received opcode is 0x41 and opcode0x41en is 1 or received opcode is 0x42 and opcode0x42en is 1 or received opcode is 0x44 and opcode0x44en is 1 or received opcode is 0x70 and opcode0x70en is 1 or received opcode is 0x82 and opcode0x82en is 1 or received opcode is 0x86 and

opcode0x86en is 1 Wakeupstatus is 0 when none of the previous conditions are true. (cec_wakeupctrl)

13.9.3.11.9.1 Offset

Register	Offset
cec_wakeupctrl	31h

13.9.3.11.9.2 Function

CEC Wake-up Control Register

After receiving a message in the CEC_RX_DATA1 (OPCODE) registers, the CEC engine verifies the message opcode[7:0] against one of the previously defined values to generate the wake-up status:

Wakeupstatus is 1 when:

received opcode is 0x04 and opcode0x04en is 1 or

received opcode is 0x0D and opcode0x0Den is 1 or

received opcode is 0x41 and opcode0x41en is 1 or

received opcode is 0x42 and opcode0x42en is 1 or

received opcode is 0x44 and opcode0x44en is 1 or

received opcode is 0x70 and opcode0x70en is 1 or

received opcode is 0x82 and opcode0x82en is 1 or

received opcode is 0x86 and opcode0x86en is 1

Wakeupstatus is 0 when none of the previous conditions are true.

This formula means that the wake-up status (on CEC_STAT[6] register) is only '1' if the opcode[7:0] received is equal to one of the defined values and the corresponding enable bit of that defined value is set to '1'.

13.9.3.11.9.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	opcode0x86en	opcode0x82en	opcode0x70en	opcode0x44en	opcode0x42en	opcode0x41en	opcode0x0Den	opcode0x04en
W								
Reset	1	1	1	1	1	1	1	1

13.9.3.11.9.4 Fields

Field	Function
7 opcode0x86en	OPCODE 0x86 wake up enable
6 opcode0x82en	OPCODE 0x82 wake up enable
5 opcode0x70en	OPCODE 0x70 wake up enable
4 opcode0x44en	OPCODE 0x44 wake up enable
3 opcode0x42en	OPCODE 0x42 wake up enable
2 opcode0x41en	OPCODE 0x41 wake up enable
1 opcode0x0Den	OPCODE 0x0D wake up enable
0 opcode0x04en	OPCODE 0x04 wake up enable

13.9.3.12 EDDC register descriptions

13.9.3.12.1 Memory map

EDDC registers base address: 7E00h

Offset	Register	Width (In bits)	Access	Reset value
0h	I2C DDC Slave address Configuration Register (i2cm_slave)	8	RW	00h
1h	I2C DDC Address Configuration Register (i2cm_address)	8	RW	00h
2h	I2C DDC Data Write Register (i2cm_datao)	8	RW	00h
3h	I2C DDC Data read Register (i2cm_datai)	8	RO	00h
4h	I2C DDC RD/RD_EXT/WR Operation Register Read and write operation request. (i2cm_operation)	8	WO	00h
5h	I2C DDC Done Interrupt Register This register configures the I2C master interrupts. (i2cm_int)	8	RW	40h
6h	I2C DDC error Interrupt Register This register configures the I2C master arbitration lost and not acknowledge error interrupts. (i2cm_ctlint)	8	RW	00h
7h	I2C DDC Speed Control Register This register configures the division relation between master and scl clock. (i2cm_div)	8	RW	0Bh
8h	I2C DDC Segment Address Configuration Register This register configures the segment address for extended R/W destination and is used for EDID reading operations, particularly for the Extended Data Read Operation for Enhanced DDC. (i2cm_segaddr)	8	RW	00h
9h	I2C DDC Software Reset Control Register This register resets the I2C master. (i2cm_softrstz)	8	RW	01h
Ah	I2C DDC Segment Pointer Register This register configures the segment pointer for extended RD/WR request. (i2cm_segptr)	8	RW	00h
Bh	I2C DDC Slow Speed SCL High Level Control Register 1 (i2cm_ss_scl_hcnt_1_addr)	8	RW	00h
Ch	I2C DDC Slow Speed SCL High Level Control Register 0 (i2cm_ss_scl_hcnt_0_addr)	8	RW	6Ch
Dh	I2C DDC Slow Speed SCL Low Level Control Register 1 (i2cm_ss_scl_lcnt_1_addr)	8	RW	00h
Eh	I2C DDC Slow Speed SCL Low Level Control Register 0 (i2cm_ss_scl_lcnt_0_addr)	8	RW	7Fh
Fh	I2C DDC Fast Speed SCL High Level Control Register 1 (i2cm_fs_scl_hcnt_1_addr)	8	RW	00h
10h	I2C DDC Fast Speed SCL High Level Control Register 0 (i2cm_fs_scl_hcnt_0_addr)	8	RW	11h
11h	I2C DDC Fast Speed SCL Low Level Control Register 1 (i2cm_fs_scl_lcnt_1_addr)	8	RW	00h
12h	I2C DDC Fast Speed SCL Low Level Control Register 0 (i2cm_fs_scl_lcnt_0_addr)	8	RW	24h
13h	I2C DDC SDA Hold Register (i2cm_sda_hold)	8	RW	09h
14h	SCDC Control Register This register configures the SCDC update status read through the I2C master interface. (i2cm_scdc_read_update)	8	RW	00h
20h	I2C Master Sequential Read Buffer Register 0 (i2cm_read_buff0)	8	RO	00h
21h	I2C Master Sequential Read Buffer Register 1 (i2cm_read_buff1)	8	RO	00h
22h	I2C Master Sequential Read Buffer Register 2 (i2cm_read_buff2)	8	RO	00h
23h	I2C Master Sequential Read Buffer Register 3 (i2cm_read_buff3)	8	RO	00h

Table continues on the next page...

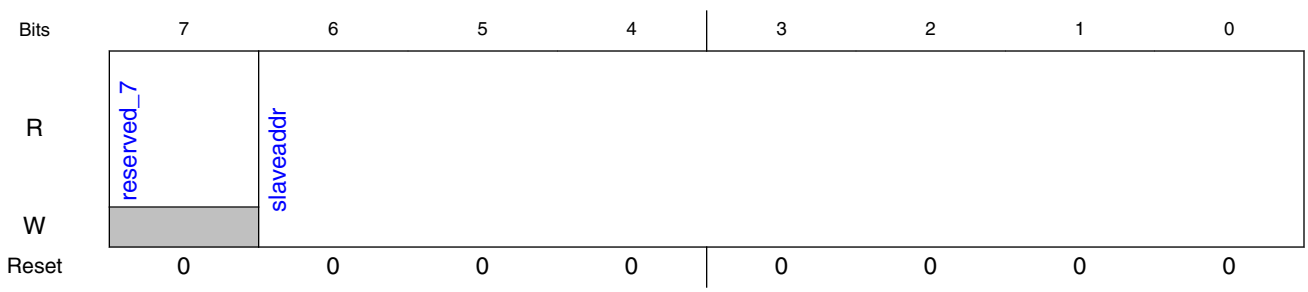
Offset	Register	Width (In bits)	Access	Reset value
24h	I2C Master Sequential Read Buffer Register 4 (i2cm_read_buff4)	8	RO	00h
25h	I2C Master Sequential Read Buffer Register 5 (i2cm_read_buff5)	8	RO	00h
26h	I2C Master Sequential Read Buffer Register 6 (i2cm_read_buff6)	8	RO	00h
27h	I2C Master Sequential Read Buffer Register 7 (i2cm_read_buff7)	8	RO	00h
30h	I2C SCDC Read Update Register 0 (i2cm_scdc_update0)	8	RO	00h
31h	I2C SCDC Read Update Register 1 (i2cm_scdc_update1)	8	RO	00h

13.9.3.12.2 I2C DDC Slave address Configuration Register (i2cm_slave)

13.9.3.12.2.1 Offset

Register	Offset
i2cm_slave	0h

13.9.3.12.2.2 Diagram



13.9.3.12.2.3 Fields

Field	Function
7 reserved_7	Reserved and read as zero
6-0 slaveaddr	Slave address to be sent during read and write normal operations.

13.9.3.12.3 I2C DDC Address Configuration Register (i2cm_address)

13.9.3.12.3.1 Offset

Register	Offset
i2cm_address	1h

13.9.3.12.3.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	address							
Reset	0	0	0	0	0	0	0	0

13.9.3.12.3.3 Fields

Field	Function
7-0 address	Register address for read and write operations

13.9.3.12.4 I2C DDC Data Write Register (i2cm_datao)

13.9.3.12.4.1 Offset

Register	Offset
i2cm_datao	2h

13.9.3.12.4.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	datao							
Reset	0	0	0	0	0	0	0	0

13.9.3.12.4.3 Fields

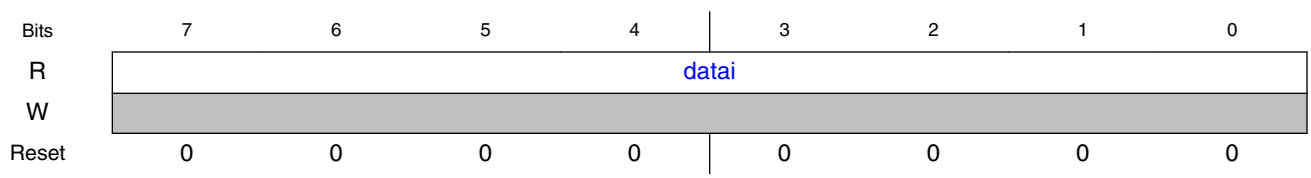
Field	Function
7-0 datao	Data to be written on register pointed by address[7:0].

13.9.3.12.5 I2C DDC Data read Register (i2cm_datai)

13.9.3.12.5.1 Offset

Register	Offset
i2cm_datai	3h

13.9.3.12.5.2 Diagram



13.9.3.12.5.3 Fields

Field	Function
7-0 datai	Data read from register pointed by address[7:0].

13.9.3.12.6 I2C DDC RD/RD_EXT/WR Operation Register Read and write operation request. (i2cm_operation)

13.9.3.12.6.1 Offset

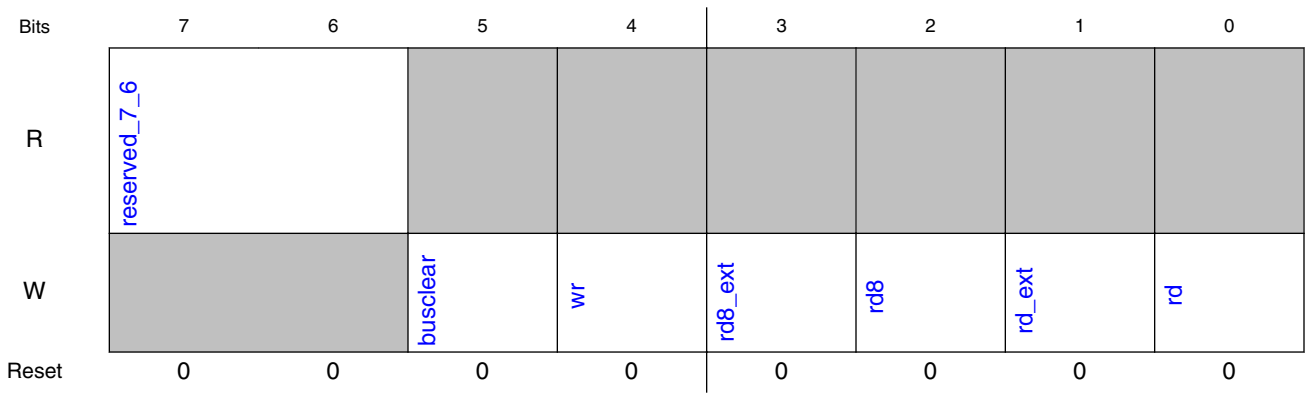
Register	Offset
i2cm_operation	4h

13.9.3.12.6.2 Function

I2C DDC RD/RD_EXT/WR Operation Register

Read and write operation request. This register can only be written; reading this register always results in 00h. Writing 1'b1 simultaneously to rd, rd_ext and wr requests is considered as a read (rd) request.

13.9.3.12.6.3 Diagram



13.9.3.12.6.4 Fields

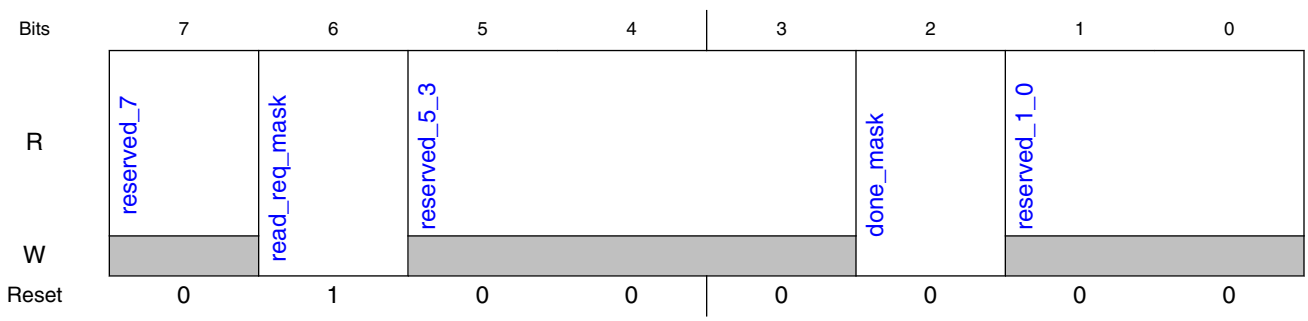
Field	Function
7-6 reserved_7_6	Reserved and read as zero
5 busclear	bus clear operation request.
4 wr	Single byte write operation request.
3 rd8_ext	Extended sequential read operation request. Extended sequential read operation request. Eight bytes are read starting at the address defined in register field i2cm_address.address and stored in registers i2cm_read_buffx. For more information on the extended read operation, refer to "I2C Master Interface Extended Read Mode" in the "Functional Description" chapter.
2 rd8	Sequential read operation request. Sequential read operation request. Eight bytes are read starting at the address defined in the i2cm_address.address register field and stored in the i2cm_read_buffx registers.
1 rd_ext	After writing 1'b1 to rd_ext bit a extended data read operation is started (E-DDC read operation). After writing 1'b1 to rd_ext bit a extended data read operation is started (E-DDC read operation). For more information on the extended read operation, refer to "I2C Master Interface Extended Read Mode" in the "Functional Description" chapter.
0 rd	Single byte read operation request

13.9.3.12.7 I2C DDC Done Interrupt Register This register configures the I2C master interrupts. (i2cm_int)

13.9.3.12.7.1 Offset

Register	Offset
i2cm_int	5h

13.9.3.12.7.2 Diagram



13.9.3.12.7.3 Fields

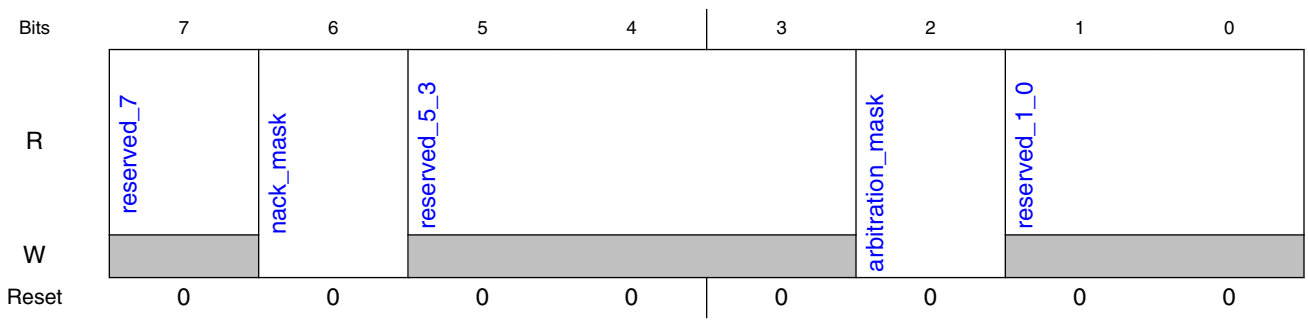
Field	Function
7 reserved_7	Reserved and read as zero
6 read_req_mask	Read request interruption mask signal.
5-3 reserved_5_3	Reserved and read as zero
2 done_mask	Done interrupt mask signal.
1-0 reserved_1_0	Reserved and read as zero

13.9.3.12.8 I2C DDC error Interrupt Register This register configures the I2C master arbitration lost and not acknowledge error interrupts. (i2cm_ctlint)

13.9.3.12.8.1 Offset

Register	Offset
i2cm_ctlint	6h

13.9.3.12.8.2 Diagram



13.9.3.12.8.3 Fields

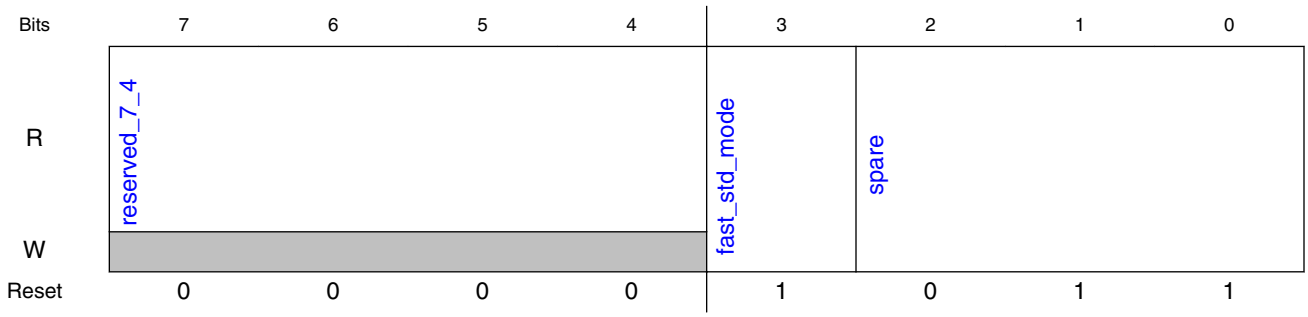
Field	Function
7 reserved_7	Reserved and read as zero
6 nack_mask	Not acknowledge error interrupt mask signal.
5-3 reserved_5_3	Reserved and read as zero
2 arbitration_mask	Arbitration error interrupt mask signal.
1-0 reserved_1_0	Reserved and read as zero

13.9.3.12.9 I2C DDC Speed Control Register This register configures the division relation between master and scl clock. (i2cm_div)

13.9.3.12.9.1 Offset

Register	Offset
i2cm_div	7h

13.9.3.12.9.2 Diagram



13.9.3.12.9.3 Fields

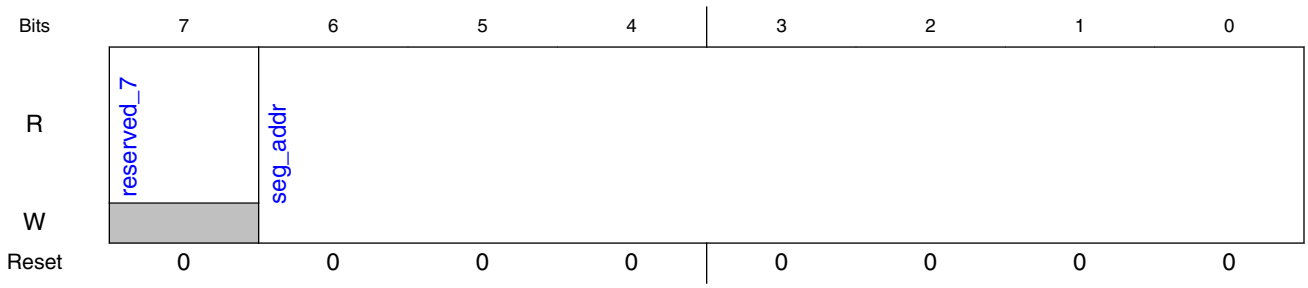
Field	Function
7-4 reserved_7_4	Reserved and read as zero
3 fast_std_mode	Sets the I2C Master to work in Fast Mode or Standard Mode: 1: Fast Mode 0: Standard Mode
2-0 spare	Reserved as "spare" bit with no associated functionality.

13.9.3.12.10 I2C DDC Segment Address Configuration Register This register configures the segment address for extended R/W destination and is used for EDID reading operations, particularly for the Extended Data Read Operation for Enhanced DDC. (i2cm_segaddr)

13.9.3.12.10.1 Offset

Register	Offset
i2cm_segaddr	8h

13.9.3.12.10.2 Diagram



13.9.3.12.10.3 Fields

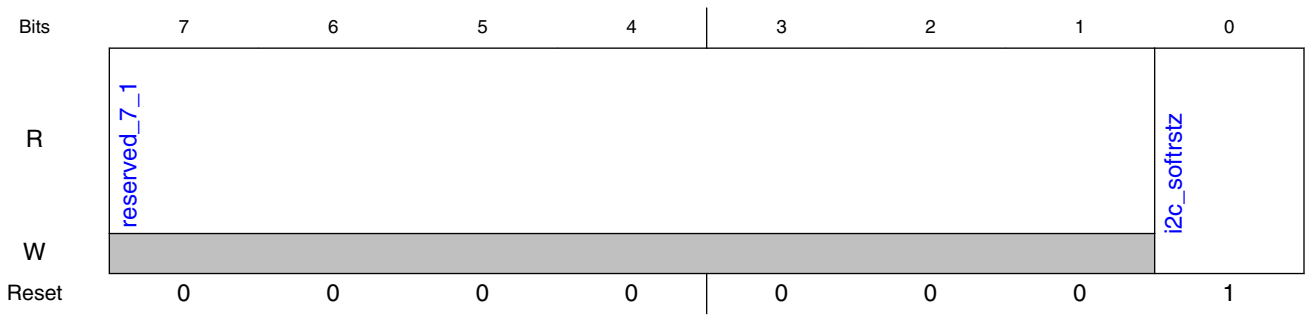
Field	Function
7 reserved_7	Reserved and read as zero
6-0 seg_addr	I2C DDC Segment Address Configuration Register

13.9.3.12.11 I2C DDC Software Reset Control Register This register resets the I2C master. (i2cm_softrstz)

13.9.3.12.11.1 Offset

Register	Offset
i2cm_softrstz	9h

13.9.3.12.11.2 Diagram



13.9.3.12.11.3 Fields

Field	Function
7-1 reserved_7_1	Reserved and read as zero
0 i2c_softrstz	I2C Master Software Reset. I2C Master Software Reset. Active by writing a zero and auto cleared to one in the following cycle.

13.9.3.12.12 I2C DDC Segment Pointer Register This register configures the segment pointer for extended RD/WR request. (i2cm_segptr)

13.9.3.12.12.1 Offset

Register	Offset
i2cm_segptr	Ah

13.9.3.12.12.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	segptr							
Reset	0	0	0	0	0	0	0	0

13.9.3.12.12.3 Fields

Field	Function
7-0 segptr	I2C DDC Segment Pointer Register

13.9.3.12.13 I2C DDC Slow Speed SCL High Level Control Register 1 (i2cm_ss_scl_hcnt_1_addr)

13.9.3.12.13.1 Offset

Register	Offset
i2cm_ss_scl_hcnt_1_addr	Bh

13.9.3.12.13.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_ss_scl_hcnt1							
Reset	0	0	0	0	0	0	0	0

13.9.3.12.13.3 Fields

Field	Function
7-0 i2cmp_ss_scl_hcnt1	I2C DDC Slow Speed SCL High Level Control Register 1

13.9.3.12.14 I2C DDC Slow Speed SCL High Level Control Register 0 (i2cm_ss_scl_hcnt_0_addr)

13.9.3.12.14.1 Offset

Register	Offset
i2cm_ss_scl_hcnt_0_addr	Ch

13.9.3.12.14.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_ss_scl_hcnt0							
Reset	0	1	1	0	1	1	0	0

13.9.3.12.14.3 Fields

Field	Function
7-0 i2cmp_ss_scl_hcnt0	I2C DDC Slow Speed SCL High Level Control Register 0

13.9.3.12.15 I2C DDC Slow Speed SCL Low Level Control Register 1 (i2cm_ss_scl_lcnt_1_addr)

13.9.3.12.15.1 Offset

Register	Offset
i2cm_ss_scl_lcnt_1_addr	Dh

13.9.3.12.15.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_ss_scl_lcnt1							
Reset	0	0	0	0	0	0	0	0

13.9.3.12.15.3 Fields

Field	Function
7-0 i2cmp_ss_scl_lcnt1	I2C DDC Slow Speed SCL Low Level Control Register 1

13.9.3.12.16 I2C DDC Slow Speed SCL Low Level Control Register 0 (i2cm_ss_scl_lcnt_0_addr)

13.9.3.12.16.1 Offset

Register	Offset
i2cm_ss_scl_lcnt_0_addr	Eh

13.9.3.12.16.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_ss_scl_lcnt0							
Reset	0	1	1	1	1	1	1	1

13.9.3.12.16.3 Fields

Field	Function
7-0 i2cmp_ss_scl_lcnt0	I2C DDC Slow Speed SCL Low Level Control Register 0

13.9.3.12.17 I2C DDC Fast Speed SCL High Level Control Register 1 (i2cm_fs_scl_hcnt_1_addr)

13.9.3.12.17.1 Offset

Register	Offset
i2cm_fs_scl_hcnt_1_addr	Fh

13.9.3.12.17.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_fs_scl_hcnt1							
Reset	0	0	0	0	0	0	0	0

13.9.3.12.17.3 Fields

Field	Function
7-0 i2cmp_fs_scl_hcnt1	I2C DDC Fast Speed SCL High Level Control Register 1

13.9.3.12.18 I2C DDC Fast Speed SCL High Level Control Register 0 (i2cm_fs_scl_hcnt_0_addr)

13.9.3.12.18.1 Offset

Register	Offset
i2cm_fs_scl_hcnt_0_addr	10h

13.9.3.12.18.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_fs_scl_hcnt0							
Reset	0	0	0	1	0	0	0	1

13.9.3.12.18.3 Fields

Field	Function
7-0 i2cmp_fs_scl_hcnt0	I2C DDC Fast Speed SCL High Level Control Register 0

13.9.3.12.19 I2C DDC Fast Speed SCL Low Level Control Register 1 (i2cm_fs_scl_lcnt_1_addr)

13.9.3.12.19.1 Offset

Register	Offset
i2cm_fs_scl_lcnt_1_addr	11h

13.9.3.12.19.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_fs_scl_lcnt1							
Reset	0	0	0	0	0	0	0	0

13.9.3.12.19.3 Fields

Field	Function
7-0 i2cmp_fs_scl_lcnt1	I2C DDC Fast Speed SCL Low Level Control Register 1

13.9.3.12.20 I2C DDC Fast Speed SCL Low Level Control Register 0 (i2cm_fs_scl_lcnt_0_addr)

13.9.3.12.20.1 Offset

Register	Offset
i2cm_fs_scl_lcnt_0_addr	12h

13.9.3.12.20.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	i2cmp_fs_scl_lcnt0							
Reset	0	0	1	0	0	1	0	0

13.9.3.12.20.3 Fields

Field	Function
7-0 i2cmp_fs_scl_lcnt0	I2C DDC Fast Speed SCL Low Level Control Register 0

13.9.3.12.21 I2C DDC SDA Hold Register (i2cm_sda_hold)

13.9.3.12.21.1 Offset

Register	Offset
i2cm_sda_hold	13h

13.9.3.12.21.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	osda_hold							
Reset	0	0	0	0	1	0	0	1

13.9.3.12.21.3 Fields

Field	Function
7-0 osda_hold	Defines the number of SFR clock cycles to meet tHD;DAT (300 ns) osda_hold = round_to_high_integer (300 ns / (1 / isfrclk_frequency))

13.9.3.12.22 SCDC Control Register This register configures the SCDC update status read through the I2C master interface. (i2cm_scdc_read_update)

13.9.3.12.22.1 Offset

Register	Offset
i2cm_scdc_read_update	14h

13.9.3.12.22.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	reserved_7_6		updtrd_vsyncpoll_en	read_request_en	reserved_3_1			
W								read_update
Reset	0	0	0	0	0	0	0	0

13.9.3.12.22.3 Fields

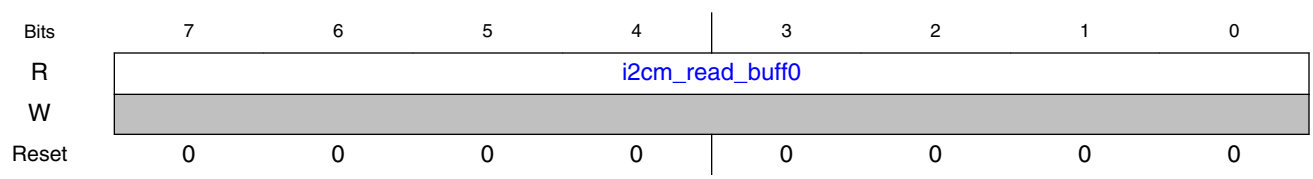
Field	Function
7-6 reserved_7_6	Reserved and read as zero
5 updtrd_vsyncpoll_en	Update read polling enabled. Update read polling enabled. When active (1'b1), an SCDC Update Read is performed at the fall of the active edge of the vertical sync pulse.
4 read_request_en	Read request enabled. Read request enabled. When active (1'b1) an SCDC Update Read shall be performed whenever a SCDC read request is detected.
3-1 reserved_3_1	Reserved and read as zero
0 read_update	When set to 1'b1, a SCDC Update Read is performed and the read data loaded into registers i2cm_scdc_update0 and i2cm_scdc_update1.

13.9.3.12.23 I2C Master Sequential Read Buffer Register 0 (i2cm_read_buff0)

13.9.3.12.23.1 Offset

Register	Offset
i2cm_read_buff0	20h

13.9.3.12.23.2 Diagram



13.9.3.12.23.3 Fields

Field	Function
7-0 i2cm_read_buff0	Byte 0 of a I2C read buffer sequential read (from address i2cm_address)

13.9.3.12.24 I2C Master Sequential Read Buffer Register 1 (i2cm_read_buff1)

13.9.3.12.24.1 Offset

Register	Offset
i2cm_read_buff1	21h

13.9.3.12.24.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	i2cm_read_buff1							
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.12.24.3 Fields

Field	Function
7-0 i2cm_read_buff1	Byte 1 of a I2C read buffer sequential read (from address i2cm_address+1)

13.9.3.12.25 I2C Master Sequential Read Buffer Register 2 (i2cm_read_buff2)

13.9.3.12.25.1 Offset

Register	Offset
i2cm_read_buff2	22h

13.9.3.12.25.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	i2cm_read_buff2							
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.12.25.3 Fields

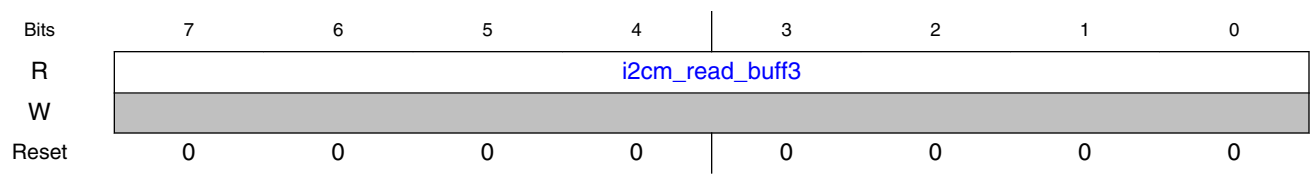
Field	Function
7-0 i2cm_read_buff2	Byte 2 of a I2C read buffer sequential read (from address i2cm_address+2)

13.9.3.12.26 I2C Master Sequential Read Buffer Register 3 (i2cm_read_buff3)

13.9.3.12.26.1 Offset

Register	Offset
i2cm_read_buff3	23h

13.9.3.12.26.2 Diagram



13.9.3.12.26.3 Fields

Field	Function
7-0 i2cm_read_buff3	Byte 3 of a I2C read buffer sequential read (from address i2cm_address+3)

13.9.3.12.27 I2C Master Sequential Read Buffer Register 4 (i2cm_read_buff4)

13.9.3.12.27.1 Offset

Register	Offset
i2cm_read_buff4	24h

13.9.3.12.27.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	i2cm_read_buff4							
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.12.27.3 Fields

Field	Function
7-0 i2cm_read_buff4	Byte 4 of a I2C read buffer sequential read (from address i2cm_address+4)

13.9.3.12.28 I2C Master Sequential Read Buffer Register 5 (i2cm_read_buff5)

13.9.3.12.28.1 Offset

Register	Offset
i2cm_read_buff5	25h

13.9.3.12.28.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	i2cm_read_buff5							
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.12.28.3 Fields

Field	Function
7-0 i2cm_read_buff5	Byte 5 of a I2C read buffer sequential read (from address i2cm_address+5)

13.9.3.12.29 I2C Master Sequential Read Buffer Register 6 (i2cm_read_buff6)

13.9.3.12.29.1 Offset

Register	Offset
i2cm_read_buff6	26h

13.9.3.12.29.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	i2cm_read_buff6							
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.12.29.3 Fields

Field	Function
7-0 i2cm_read_buff6	Byte 6 of a I2C read buffer sequential read (from address i2cm_address+6)

13.9.3.12.30 I2C Master Sequential Read Buffer Register 7 (i2cm_read_buff7)

13.9.3.12.30.1 Offset

Register	Offset
i2cm_read_buff7	27h

13.9.3.12.30.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	i2cm_read_buff7							
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.12.30.3 Fields

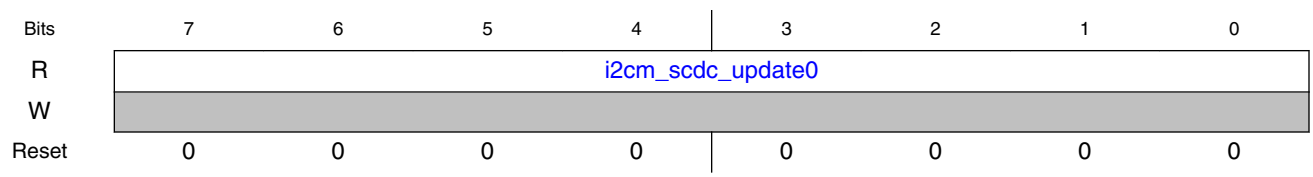
Field	Function
7-0 i2cm_read_buff7	Byte 7 of a I2C read buffer sequential read (from address i2cm_address+7)

13.9.3.12.31 I2C SCDC Read Update Register 0 (i2cm_scdc_update0)

13.9.3.12.31.1 Offset

Register	Offset
i2cm_scdc_update0	30h

13.9.3.12.31.2 Diagram



13.9.3.12.31.3 Fields

Field	Function
7-0 i2cm_scdc_update0	Byte 0 of a SCDC I2C update sequential read

13.9.3.12.32 I2C SCDC Read Update Register 1 (i2cm_scdc_update1)

13.9.3.12.32.1 Offset

Register	Offset
i2cm_scdc_update1	31h

13.9.3.12.32.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	i2cm_scdc_update1							
W								
Reset	0	0	0	0	0	0	0	0

13.9.3.12.32.3 Fields

Field	Function
7-0 i2cm_scdc_update1	Byte 1 of a SCDC I2C update sequential read

13.10 HDMI TX PHY

13.10.1 Overview

The HDMI (High Definition Multimedia Interface) TX PHY is compatible with the HDMI v1.4/v2.0 spec and supports 4Kp30 video resolution. It accepts Transition-Minimized Differential Signaling (TMDS) coded parallel data from the HDMI link layer and transmits them into HDMI cable with a serialized form.

13.10.1.1 Block Diagram

The high level block diagram for the HDMI TX PHY is shown below.

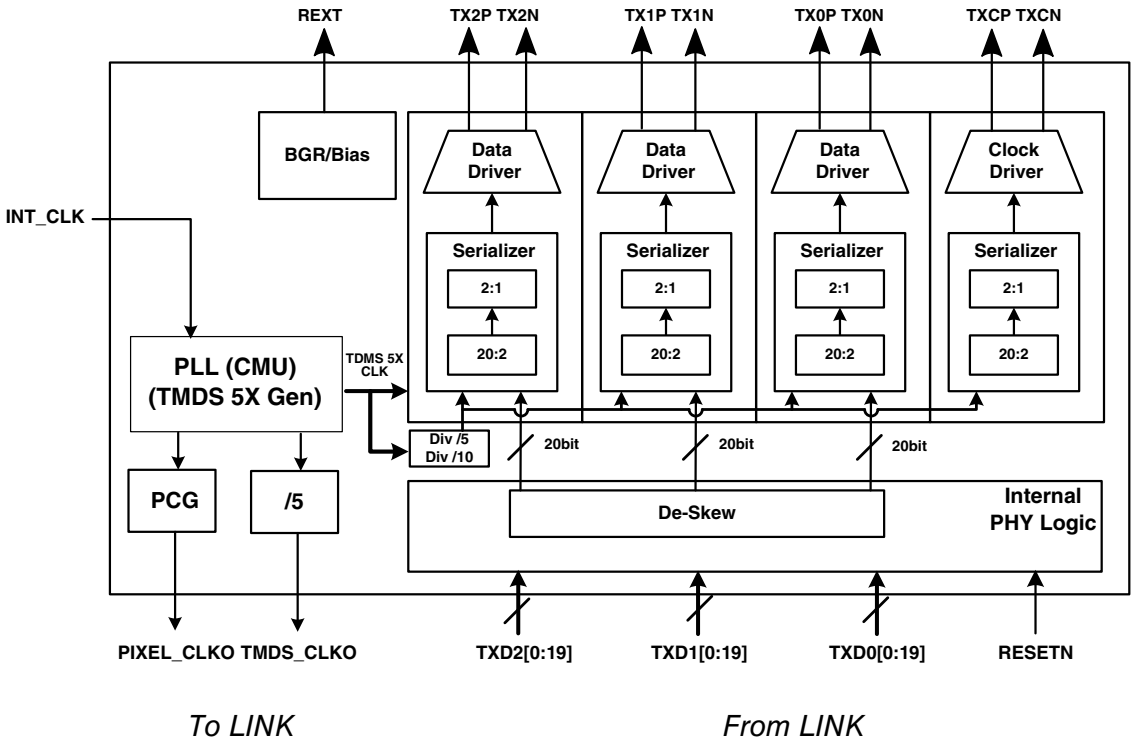


Figure 13-78. HDMI TX PHY Block Diagram

The PHY receives parallel data from the LINK. The data is serialized and transmitted on the TXnP/N differential outputs, which are driven from the TX open-drain type driver. The TX driver characteristics, such as amplitude and pre-emphasis are user-controlled. BIAS block can generate bias currents that are used for each lane using the band-gap reference and an external resistor through the REXT pad. The PLL generates high-speed clock, which is needed for TX serializer.

13.10.1.2 Features

HDMI TX PHY supports the following features:

- Supports 25 MHz to 594 MHz TMDS clock
- 20-bit parallel data interface for transmit up to 297 MHz
- Supports integrated pixel clock generator
- Supports internal loop-back for low cost at-speed test
- (All DTV Video formats up to 1080p/12-bit, 3D, 4K x 2K/60 Hz, and VGA/XGA/SXGA/UXGA format for PC)
- Supports on-chip source termination
- Supports programmable analog characteristics through JTAG, APB register control
- Supports 60-bit CMOS interface with LINK logic for relaxed timing constraint between LINK and PHY

- Supports 0.8 V/1.8 V power supply
- Supports 24 MHz or 26 MHz reference clock is required

13.10.2 Functional Description

13.10.2.1 Common Block

The Common block consists of a Band-gap Reference(BGR), Bias generator, and Phase Locked Loop (PLL). Additonal details on each block is provided in the following sections.

13.10.2.1.1 Band-gap Reference (BGR) and Bias Generator

The BGR circuit generates high-accuracy reference voltage of 820mV across Process, Voltage and Temperature (PVT) variation. A bias generator makes two kinds of current, external- and internal-resistor (RMRES)-referred current, and these currents are distributed to each block in Common and Lanes. The external resistor-referred (REXT) current is generated using an 8.2KΩ(+/-1%) resistor which is connected between the REXT pad and ground outside the chip. On the other hand, the internal resistor-referred current is generated using internally integrated replica resistor to compensate resistance error across PVT variation.

13.10.2.1.2 Phase Locked Loop (PLL)

The PLL in the Common synthesizes high-speed clock, which is used for TX serializer from the reference clock. The nominal reference frequency is 24MHz.

In the HDMI TX PHY, the high frequency clock from the PLL is used in the serialization of TX data. The following are the main features of the PLL:

- Supports ring VCO from 0.75 GHz to 3GHz clock
- Fractional PLL with sigma delta modulated divider.
- Automatic frequency calibrated (AFC) oscillator setting for the desired frequency of operation
- Programmable divider value control for pre-divider (P), main divider (M), and post divider (S)
- Programmable charge pump current and loop filter resistance allowing for wide range of programmable loop bandwidth and peaking
-

The output frequency of PLL in integer mode is derived with the following formula:

$$F_{PLL} = F_{REF} \times M/(P \times S)$$

P, M and S are pre-, main- and post-divider values, respectively.

The output frequency of PLL in fractional mode is derived with the following fomula:

$$TCLK5X = F_{ref} \times 1/P \times (M + K/LC \times 1/PI \times ((SDC \times P)/24)) \times 1/S$$

SDC is derived with the following fomula:

$$SDC = F_{ref} \times (M/P) / (N \times (LC_{sub} - K_{sub})/(LC_{sub}) + (N - 1) \times K_{sub})/LC_{sub})$$

K and LC are nominator and denominator of the fractional part, PI is the number of phase of the phase interpolator, N is the dividing-ratio for the Sigma-Delta Modulation (SDM) clock, respectively. The PLL clock is distributed to each lane by a clock distribution (CD) buffers.

13.10.2.2 Lane

Each Lane block consists of serializer, deserializer, and driver.

Figure 13-79. Lane Block Diagram

13.10.2.3 Serializer

The serializer accepts 20-bit data (TXD[0:19]) from the PHY digital via 20-bit bus which is synchronized to TX byte clock (TMDS_CLKHI). The parallel data is converted to the high-speed serial data.

NOTE

The LSB in the input parallel data is transmitted first and the MSB the last. This data will be serialized and transmitted on the TXnP/N differential outputs at 0.25~6Gb/s according to specified resolutions.

13.10.2.4 Driver

The driver generates baud-rate voltage waveform on TX output pads, TXnP/N. The driver comprises of driver controller, pre-driver and main-driver. The output data of serializer are delivered to the main-driver via the predriver and the driver controller. The main driver is based on open-drain type driver. The amplitude of output signaling can be controlled by TX_CH_AMP_LVL.

TX driver has 2-Tap de-emphasis function to compensate channel loss. TX equalization is controlled primarily through the TX_EMP_LVL bit bit. These pins control internal registers values.

13.10.2.5 Deserializer

In the BIST (Built-In Self-Test) mode, the serial data from the serializer is fed back to the deserializer. Deserializer converted serial data to 20-bit parallel data and transferred them into PHY digital circuit to check bit error. The main purpose of deserializer is to verify the functionality of the serializer.

13.10.3 Memory Map and register definition

This section includes the HDMI_TX PHY module memory map and descriptions of the registers.

13.10.3.1 HDMI_TX_PHY register descriptions

13.10.3.1.1 HDMI_TX_PHY memory map

HDMI_TX_PHY base address: 32FD_FF00h

Offset	Register	Width (In bits)	Access	Reset value
4h	HDMI TX PHY Register 1 (PHY_REG1)	8	RW	D1h
8h	HDMI TX PHY Register 2 (PHY_REG2)	8	RW	3Eh
Ch	HDMI TX PHY Register 3 (PHY_REG3)	8	RW	15h
10h	HDMI TX PHY Register 4 (PHY_REG4)	8	RW	C0h
14h	HDMI TX PHYRegister 5 (PHY_REG5)	8	RW	90h
18h	HDMI TX PHY Register 6 (PHY_REG6)	8	RW	88h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1Ch	HDMI TX PHY Register 7 (PHY_REG7)	8	RW	42h
20h	HDMI TX PHY Register 8 (PHY_REG8)	8	RW	47h
24h	HDMI TX PHY Register 9 (PHY_REG9)	8	RW	30h
28h	HDMI TX PHY Register 10 (PHY_REG10)	8	RW	33h
2Ch	HDMI TX PHY Register 11 (PHY_REG11)	8	RW	65h
30h	HDMI TX PHY Register 12 (PHY_REG12)	8	RW	20h
34h	HDMI TX PHY Register 13 (PHY_REG13)	8	RW	A6h
38h	HDMI TX PHY Register 14 (PHY_REG14)	8	RW	24h
3Ch	HDMI TX PHY Register 15 (PHY_REG15)	8	RW	80h
40h	HDMI TX PHY Register 16 (PHY_REG16)	8	RW	6Ch
44h	HDMI TX PHY Register 17 (PHY_REG17)	8	RW	F2h
48h	HDMI TX PHY Register 18 (PHY_REG18)	8	RW	67h
4Ch	HDMI TX PHY Register 19 (PHY_REG19)	8	RW	00h
50h	HDMI TX PHY Register 20 (PHY_REG20)	8	RW	10h
54h	HDMI TX PHY Register 21 (PHY_REG21)	8	RW	00h
58h	HDMI TX PHY Register 22 (PHY_REG22)	8	RW	00h
5Ch	HDMI TX PHY Register 23 (PHY_REG23)	8	RW	10h
60h	HDMI TX PHY Register 24 (PHY_REG24)	8	RW	60h
64h	HDMI TX PHY Register 25 (PHY_REG25)	8	RW	00h
68h	HDMI TX PHY Register 26 (PHY_REG26)	8	RW	00h
6Ch	HDMI TX PHY Register 27 (PHY_REG27)	8	RW	00h
70h	HDMI TX PHY Register 28 (PHY_REG28)	8	RW	00h
74h	HDMI TX PHY Register 29 (PHY_REG29)	8	RW	00h
78h	HDMI TX PHY Register 30 (PHY_REG30)	8	RW	00h
7Ch	HDMI TX PHY Register 31 (PHY_REG31)	8	RW	00h
80h	HDMI TX PHY Register 32 (PHY_REG32)	8	RW	00h
84h	HDMI TX PHY Register 33 (PHY_REG33)	8	RW	80h
88h	HDMI TX PHY Register 34 (PHY_REG34)	8	R	00h
8Ch	HDMI TX PHY Register 35 (PHY_REG35)	8	RW	00h
90h	HDMI TX PHY Register 36 (PHY_REG36)	8	RW	00h
94h	HDMI TX PHY Register 37 (PHY_REG37)	8	RW	00h
98h	HDMI TX PHY Register 38 (PHY_REG38)	8	RW	00h
9Ch	HDMI TX PHY Register 39 (PHY_REG39)	8	RW	00h
A0h	HDMI TX PHY Register 40 (PHY_REG40)	8	RW	00h
A4h	HDMI TX PHY Register 41 (PHY_REG41)	8	RW	E0h
A8h	HDMI TX PHY Register 42 (PHY_REG42)	8	RW	83h
ACh	HDMI TX PHY Register 43 (PHY_REG43)	8	RW	0Fh
B0h	HDMI TX PHY Register 44 (PHY_REG44)	8	RW	3Eh
B4h	HDMI TX PHY Register 45 (PHY_REG45)	8	RW	F8h
B8h	HDMI TX PHY Register 46 (PHY_REG46)	8	R	00h

Table continues on the next page...

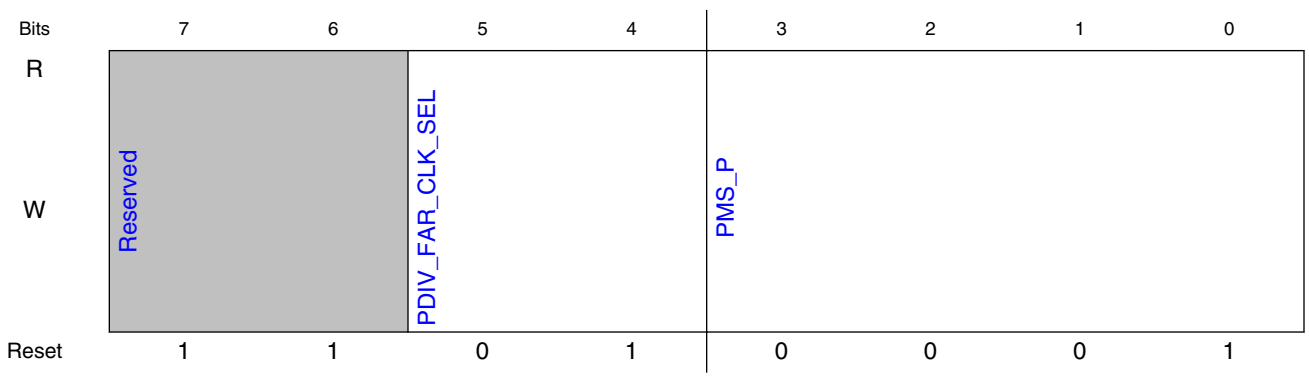
Offset	Register	Width (In bits)	Access	Reset value
BCh	HDMI TX PHY Register 47 (PHY_REG47)	8	RW	00h

13.10.3.1.2 HDMI TX PHY Register 1 (PHY_REG1)

13.10.3.1.2.1 Offset

Register	Offset
PHY_REG1	4h

13.10.3.1.2.2 Diagram



13.10.3.1.2.3 Fields

Field	Function
7-6 —	Reserved
5-4 PDIV_FAR_CLK_SEL	Clock source selection Clock Source Selection 00b-01b - When the REFCLK_SEL port is 0, selects none. When the REFCLK_SELport is 1, selects INT_CLK. 10b - Selects Pixel CLK. It is no longer used. 11b - selects none.
3-0 PMS_P	PLL Pre Divider Division Ratio PLL Pre Divider Division Ratio 0000b - div by 1 0001b - div by 1 0010b - div by 2

Field	Function
	0011b - div by 3 0100b - div by 4 0101b - div by 5 0110b - div by 6 0111b - div by 7 1000b - div by 8 1001b - div by 9 1010b - div by 10 1011b - div by 11 1100b - div by 11

13.10.3.1.3 HDMI TX PHY Register 2 (PHY_REG2)

13.10.3.1.3.1 Offset

Register	Offset
PHY_REG2	8h

13.10.3.1.3.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	PMS_M							
Reset	0	0	1	1	1	1	1	0

13.10.3.1.3.3 Fields

Field	Function
7-0 PMS_M	PLL Main Divider Division Ratio PLL Main Divider Division Ratio (Div by -Div by 255) 0011_1110b - 62

13.10.3.1.4 HDMI TX PHY Register 3 (PHY_REG3)

13.10.3.1.4.1 Offset

Register	Offset
PHY_REG3	Ch

13.10.3.1.4.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	PMS_S				SDC_N2	SDC_N		
W								
Reset	0	0	0	1	0	1	0	1

13.10.3.1.4.3 Fields

Field	Function
7-4 PMS_S	PLL post divider division ratio PLL post divider division ratio 0000b - 1 0001b - 2 0011b - 4 0101b - 6 0111b - 8 1001b - 10 1011b - 12 1111b - 16
3 SDC_N2	SDC gen pre-divider value SDC gen pre-divider value 0b - div by1 1b - div by 2
2-0 SDC_N	PLL N/N-1 divider division ratio PLL N/N-1 divider division ratio 000b - 4/3 001b - 5/4 010b - 6/5 100b - 7/6 101b - 8/7 110b - 9/8

13.10.3.1.5 HDMI TX PHY Register 4 (PHY_REG4)

13.10.3.1.5.1 Offset

Register	Offset
PHY_REG4	10h

13.10.3.1.5.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	SDM_K_EN	SDM_LC						
W								
Reset	1	1	0	0	0	0	0	0

13.10.3.1.5.3 Fields

Field	Function
7 SDM_K_EN	Enable PLL sigma-delta Enable PLL sigma-delta 0b - Integer 1b - Fractional-N
6-0 SDM_LC	PLL main sigma-delta fractional-N ratio denominator (K/LCPLL main sigma-delta fractional-N ratio denominator (K/LC) PLL main sigma-delta fractional-N ratio denominator (K/LC) 100_0000b - 64

13.10.3.1.6 HDMI TX PHYRegister 5 (PHY_REG5)

13.10.3.1.6.1 Offset

Register	Offset
PHY_REG5	14h

13.10.3.1.6.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	SDM_K_SIGN	SDM_K_CODE						
W								
Reset	1	0	0	1	0	0	0	0

13.10.3.1.6.3 Fields

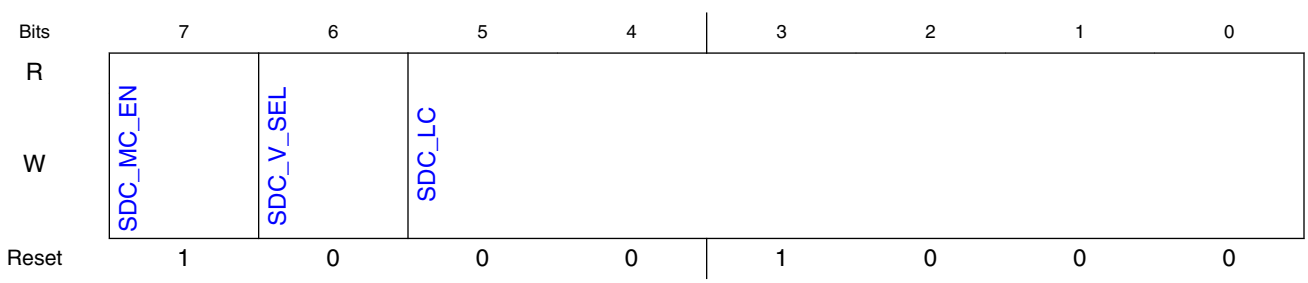
Field	Function
7 SDM_K_SIGN	Sign of the numerator Sign of the numerator 0b - Positive 1b - Negative
6-0 SDM_K_CODE	PLL main sigma-delta fractional-N ratio numerator(K/LC) PLL main sigma-delta fractional-N ratio numerator(K/LC) 000_1000b - 16

13.10.3.1.7 HDMI TX PHY Register 6 (PHY_REG6)

13.10.3.1.7.1 Offset

Register	Offset
PHY_REG6	18h

13.10.3.1.7.2 Diagram



13.10.3.1.7.3 Fields

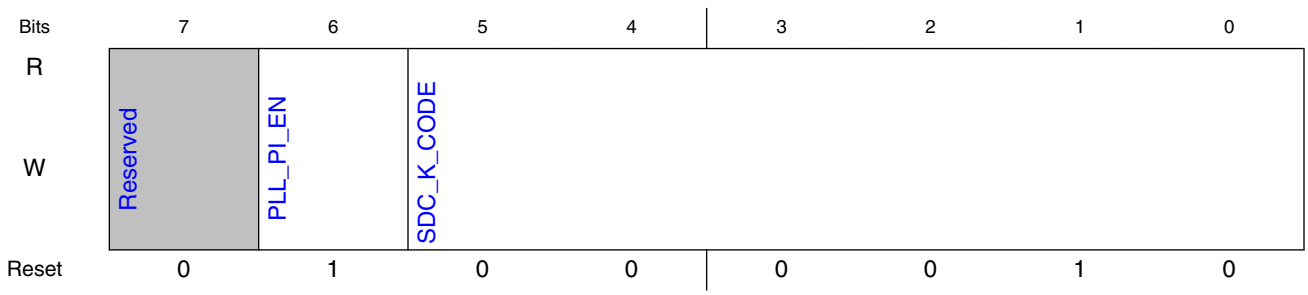
Field	Function
7 SDC_MC_EN	SDC_MC_EN 0b - SDC_GEN disable 1b - SDC_GEN enable
6 SDC_V_SEL	If SDC_MC_EN= 0, SDC_V_SEL= MC(SDC_GEN output) If SDC_MC_EN= 0, SDC_V_SEL= MC (SDC_GEN output)
5-0 SDC_LC	PLL SDC GEN Fractional-N ratio denominator (K/LC) PLL SDC GEN Fractional-N ratio denominator (K/LC) 00_1000b - 8

13.10.3.1.8 HDMI TX PHY Register 7 (PHY_REG7)

13.10.3.1.8.1 Offset

Register	Offset
PHY_REG7	1Ch

13.10.3.1.8.2 Diagram



13.10.3.1.8.3 Fields

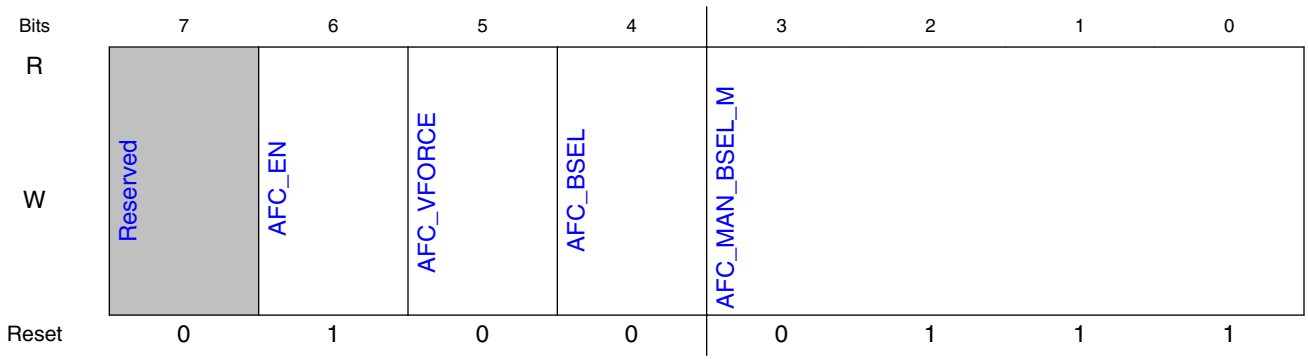
Field	Function
7 —	Reserved
6 PLL_PI_EN	PLL_PI_EN 0b - Integer N PLL 1b - Fractional-N PLL
5-0 SDC_K_CODE	PLL SDC GEN fractional-N ratio numerator (K/LC) PLL SDC GEN fractional-N ratio numerator (K/LC) 00_0010b - 2

13.10.3.1.9 HDMI TX PHY Register 8 (PHY_REG8)

13.10.3.1.9.1 Offset

Register	Offset
PHY_REG8	20h

13.10.3.1.9.2 Diagram



13.10.3.1.9.3 Fields

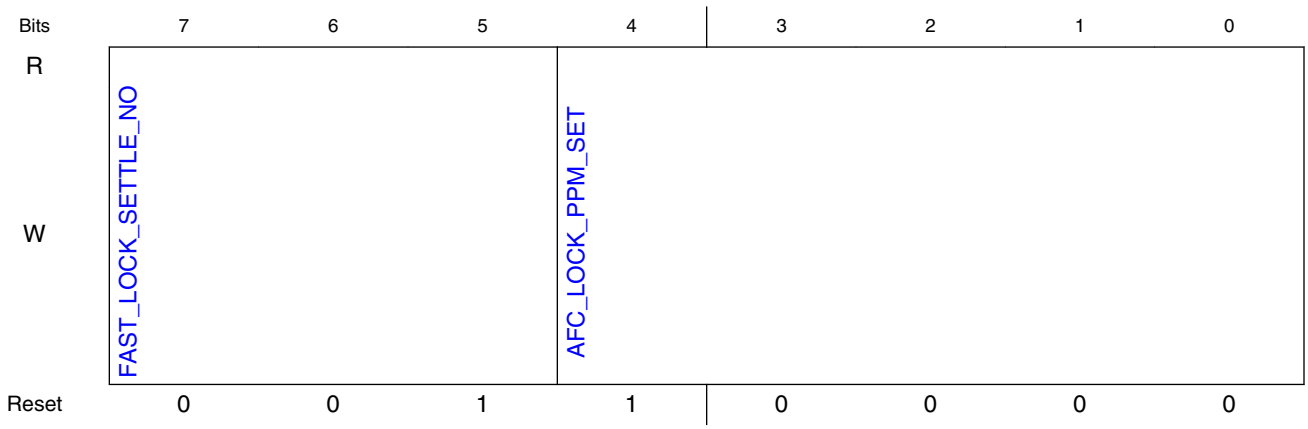
Field	Function
7 —	Reserved
6 AFC_EN	AFC_EN 0b - AFC Disable 1b - AFC Enable
5 AFC_VFORCE	To monitor VCO performance in an open loop PLL with a fixed Vctrl
4 AFC_BSEL	AFC_BSEL 0b - Find AFC 1b - Fix AFC code
3-0 AFC_MAN_BSEL_M	AFC_MAN_BSEL_M If AFC_BSEL=0. MAN_BSEL_M = initial start code, else MAN_BSEL_M = Fixed AFC MSB code.

13.10.3.1.10 HDMI TX PHY Register 9 (PHY_REG9)

13.10.3.1.10.1 Offset

Register	Offset
PHY_REG9	24h

13.10.3.1.10.2 Diagram



13.10.3.1.10.3 Fields

Field	Function
7-5 FAST_LOCK_SETTLE_NO	FAST_LOCK settlement number setting FAST_LOCK settlement number setting. Define LOCK continuous conditions for FAST_LOCK to be high.
4-0 AFC_LOCK_PPM_SET	PLL locking ppm range setting PLL locking ppm range setting. PLL_LOCK signal is combination logic of two kinds of lock signals: FAST_LOCK and SLOW_LOCK, respectively. PLL_LOCK becomes high under the following conditions: - FAST_LOCK = 1 or SLOW_LOCK = 1 when FAST/SLOW_LOCK_BYPASS =0 (default setting) - FAST_LOCK = 1 when FAST_LOCK_BYPASS =1 - SLOW _LOCK = 1 when SLOW_LOCK_BYPASS =1 Otherwise, PLL_LOCK is 0. The FAST_LOCK becomes high when the following 2 conditions are met. Condition1: If the difference between the VCO clock and the reference clock counter value is within PLL_LOCK_PPM_SET value for a certain period of time. This period is selected by shifting PLL PMS_M [7: 0] to the right until it becomes bigger than 1024. When PMS_M [7: 0] is h5c, if shift it 4 to the right or multiply it by 16. It will be h5c0 (1280). It means that the VCO clock counter must be 1280 during 16 reference clock period if PLL is locked. Condition2: If the number of consecutive lock of the previous condition1 is maintained above the setting number in FAST_LOCK_SETTLE_NO[2:0] (APB24, bit7-5) The slow lock becomes high if the difference between the counter value of PLL clock after M main divider during 1024 reference clock and the expected value 1024 is within the PLL_LOCK_PPM_SET error range.

13.10.3.1.11 HDMI TX PHY Register 10 (PHY_REG10)

13.10.3.1.11.1 Offset

Register	Offset
PHY_REG10	28h

13.10.3.1.11.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	AFC_TOL				AFC_STB_NUM			
Reset	0	0	1	1	0	0	1	1

13.10.3.1.11.3 Fields

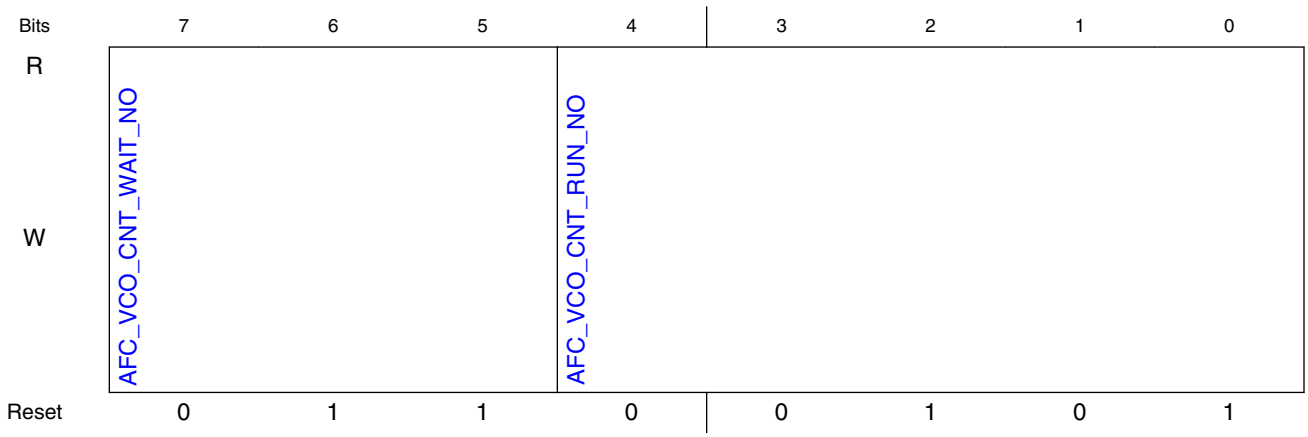
Field	Function
7-4 AFC_TOL	Tolerance number for stabilization check of VCO frequency during AFC
3-0 AFC_STB_NUM	Number of iterations for stabilization check of VCO frequency during

13.10.3.1.12 HDMI TX PHY Register 11 (PHY_REG11)

13.10.3.1.12.1 Offset

Register	Offset
PHY_REG11	2Ch

13.10.3.1.12.2 Diagram



13.10.3.1.12.3 Fields

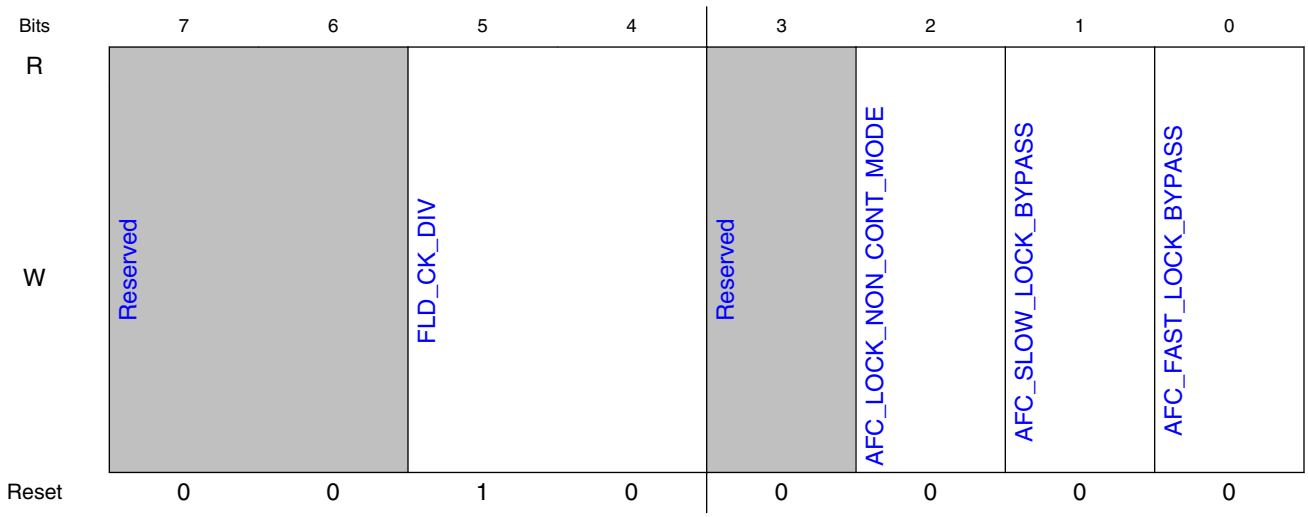
Field	Function
7-5 AFC_VCO_CNT_WAIT_NO	VCO counter waiting period before VCO_CNT running REF_CLK period * (VCO_CNT_WAIT_NO +1)
4-0 AFC_VCO_CNT_RUN_NO	VCO counter running period REF_CLK period * (VCO_CNT_RUN_NO +1)

13.10.3.1.13 HDMI TX PHY Register 12 (PHY_REG12)

13.10.3.1.13.1 Offset

Register	Offset
PHY_REG12	30h

13.10.3.1.13.2 Diagram



13.10.3.1.13.3 Fields

Field	Function
7-6 —	Reserved
5-4 FLD_CK_DIV	FLD_CK_DIV FLD internal frequency should be less than 50MHz 00b - MDS_CLK 01b - MDS_CLK/2 10b - TMDS_CLK/4 11b - TMDS_CLK/8
3 —	Reserved
2 AFC_LOCK_NON_CONT_MODE	AFC_LOCK_NON_CONT_MODE 0b - Check the lock of the PLL constantly 1b - Check the lock of PLL once
1 AFC_SLOW_LOCK_BYPASS	Slow PLL lock bypass 0b - PLL_LOCK is generated by SLOW_PLL_LOCK 1b - PLL_LOCK is generated only by FAST_PLL_LOCK, but SLOW_PLL_LOCK is not used
0 AFC_FAST_LOCK_BYPASS	Fast PLL lock bypass 0b - PLL_LOCK is generated by FAST_PLL_LOCK 1b - PLL_LOCK is generated only by SLOW_PLL_LOCK, but FAST_PLL_LOCK is not used

13.10.3.1.14 HDMI TX PHY Register 13 (PHY_REG13)

13.10.3.1.14.1 Offset

Register	Offset
PHY_REG13	34h

13.10.3.1.14.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	FLD_TG_CODE							
Reset	1	0	1	0	0	1	1	0

13.10.3.1.14.3 Fields

Field	Function
7-0 FLD_TG_CODE	FLD target code (FLD_TG_CODE[7:0]) If CK_DIV=00, Int_pllclk = TMDS_CLK If CK_DIV=01, Int_pllclk = TMDS_CLK/2 If CK_DIV=10, Int_pllclk = TMDS_CLK/4 If CK_DIV=11, Int_pllclk = TMDS_CLK/8 FLD_TG_CODE = (256/Freq. of Int_pllclk) * 24 Default : REFCLK=24MHz, TMDS_CLK=148.352 MHz, int_pllclk=37.0879MHz, FLD_TG_CODE = 166 (010100110)

13.10.3.1.15 HDMI TX PHY Register 14 (PHY_REG14)

13.10.3.1.15.1 Offset

Register	Offset
PHY_REG14	38h

13.10.3.1.15.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	FLD_TOL				FLD_RP_CODE			FLD_TG_CODE
W								
Reset	0	0	1	0	0	1	0	0

13.10.3.1.15.3 Fields

Field	Function
7-4 FLD_TOL	Fine FLD tolerance Tolerance : (TOL+1)/256, 1.17%
3-1 FLD_RP_CODE	Fine FLD RP code 010b - 2 (default)
0 FLD_TG_CODE	FLD_TG_CODE[8] Please see PHY_REG13 for definiton

13.10.3.1.16 HDMI TX PHY Register 15 (PHY_REG15)

13.10.3.1.16.1 Offset

Register	Offset
PHY_REG15	3Ch

13.10.3.1.16.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	ANA_VCO_PI_CTRL		SDM_PI_STEP		Reserved			
W								
Reset	1	0	0	0	0	0	0	0

13.10.3.1.16.3 Fields

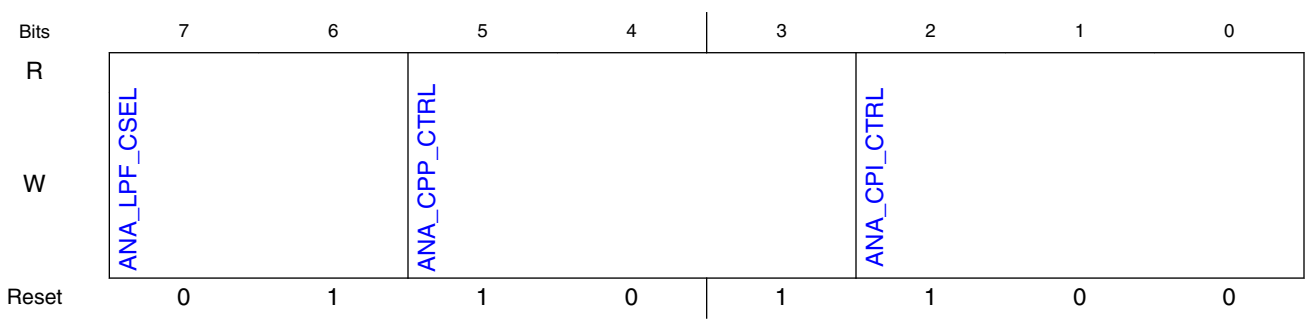
Field	Function
7-6 ANA_VCO_PI_CTRL	VCO Kvp/Kvi ratio control 00b - X 01b - 1:33 10b - 1:20 11b - 1:12.5
5-4 SDM_PI_STEP	The step of shift 00b - 1-step(PI=16) 01b - 2-step(PI=8) 10b-11b - Not Used
3-0 —	Reserved

13.10.3.1.17 HDMI TX PHY Register 16 (PHY_REG16)

13.10.3.1.17.1 Offset

Register	Offset
PHY_REG16	40h

13.10.3.1.17.2 Diagram



13.10.3.1.17.3 Fields

Field	Function
7-6 ANA_LPF_CSEL	LPF C control 00b - pF 01b - pF 10b - pF

Table continues on the next page...

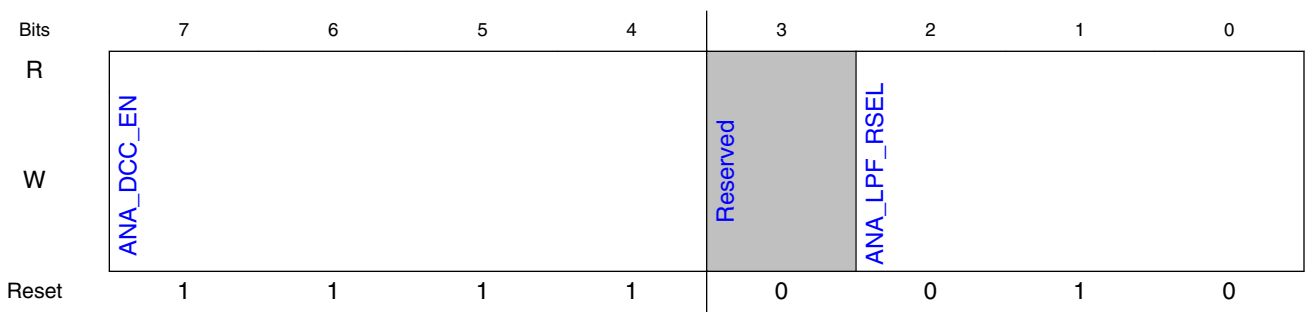
Field	Function
	11b - pF
5-3 ANA_CPP_CTRL	Proportional charge-pump current control 000b - 6.7uA 001b - 5.0uA 010b - 5.7uA 011b - 00uA 100b - 20uA 101b - 50uA 110b - 00uA 111b - 00uA
2-0 ANA_CPI_CTRL	Integral charge-pump current control 100b - .25uA 101b - .5625uA 110b - .0833uA 111b - .1250uA

13.10.3.1.18 HDMI TX PHY Register 17 (PHY_REG17)

13.10.3.1.18.1 Offset

Register	Offset
PHY_REG17	44h

13.10.3.1.18.2 Diagram



13.10.3.1.18.3 Fields

Field	Function
7-4 ANA_DCC_EN	DCC enable for each lane
3	Reserved

Table continues on the next page...

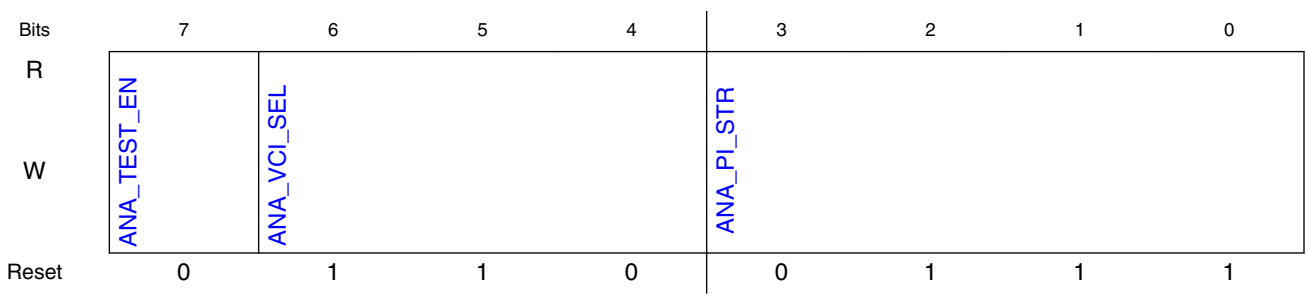
Field	Function
—	
2-0 ANA_LPF_RSEL	LPF R control 000b - K 001b - K 010b - 2K 011b - 6K 100b - 0K 101b - 4K 110b - 8K 111b - 2K

13.10.3.1.19 HDMI TX PHY Register 18 (PHY_REG18)

13.10.3.1.19.1 Offset

Register	Offset
PHY_REG18	48h

13.10.3.1.19.2 Diagram



13.10.3.1.19.3 Fields

Field	Function
7 ANA_TEST_EN	AFC Vctrl_I test mode enable
6-4 ANA_VCI_SEL	Initial AFC Vctrl_I voltage option 000b - 1.5V 001b - 1.4V 010b - 1.3V 011b - 1.2V 100b - 1.1V 101b - 1.0V 110b - 0.9V

Table continues on the next page...

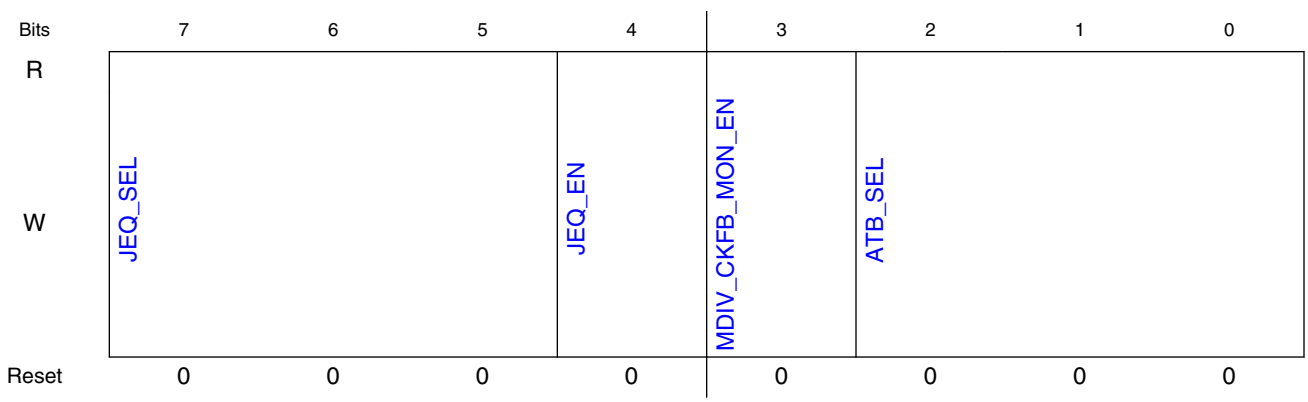
Field	Function
	111b - 0.8V
3-0 ANA_PI_STR	Phase blender buffer strength control

13.10.3.1.20 HDMI TX PHY Register 19 (PHY_REG19)

13.10.3.1.20.1 Offset

Register	Offset
PHY_REG19	4Ch

13.10.3.1.20.2 Diagram



13.10.3.1.20.3 Fields

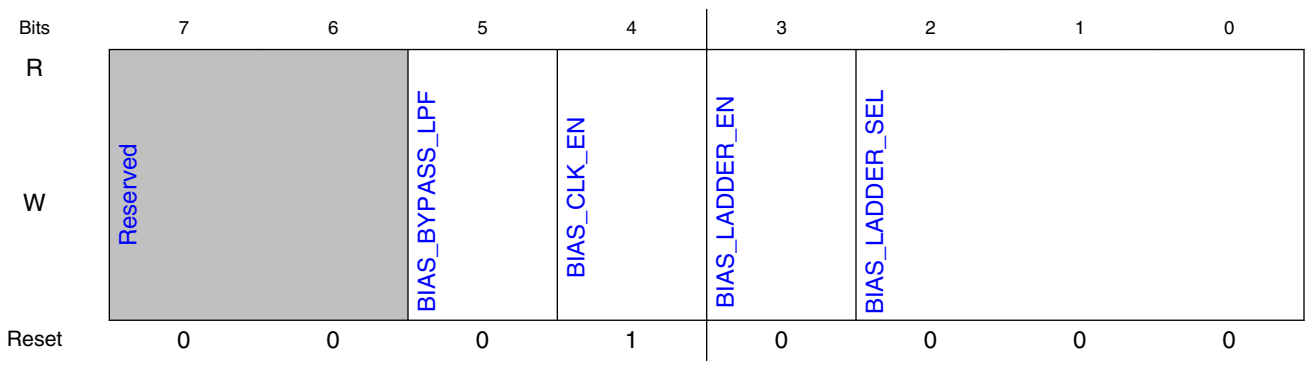
Field	Function
7-5 JEQ_SEL	Jitter EQ level control
4 JEQ_EN	Jitter EQ Enable
3 MDIV_CKFB_MON_EN	Feedback clock monitoring option
2-0 ATB_SEL	ATB selection control 000b - 001b - FC Vctrl_I 010b - CI_LPF 100b - CM

13.10.3.1.21 HDMI TX PHY Register 20 (PHY_REG20)

13.10.3.1.21.1 Offset

Register	Offset
PHY_REG20	50h

13.10.3.1.21.2 Diagram



13.10.3.1.21.3 Fields

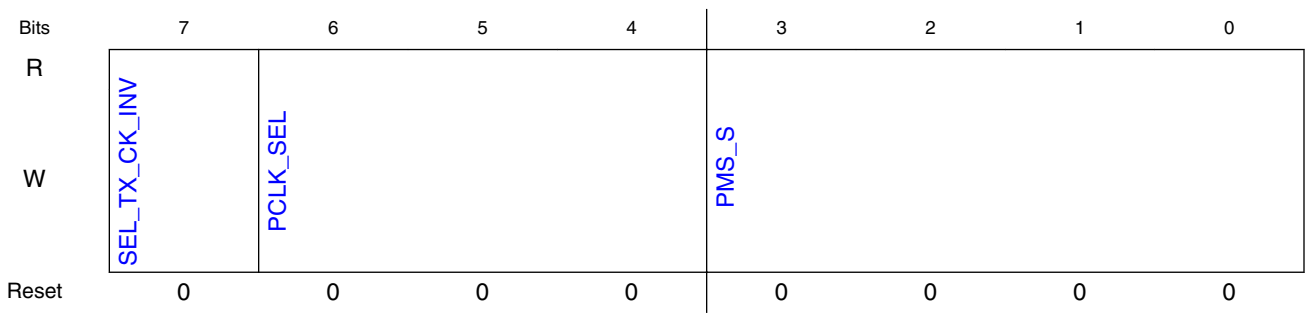
Field	Function
7-6 —	Reserved
5 BIAS_BYPASS_LPF	BIAS LPF bypass mode on
4 BIAS_CLK_EN	BGR amp. Chopper clock enable
3 BIAS_LADDER_EN	VREF voltage selection 0b - GR 1b - esistor ladder
2-0 BIAS_LADDER_SEL	BIAS_LADDER_SEL 000b - 1.0V 001b - 0.96V 010b - 0.92V 011b - 0.88V 100b - 0.84V 101b - 0.8V 110b - 0.76V 111b - 0.72V

13.10.3.1.22 HDMI TX PHY Register 21 (PHY_REG21)

13.10.3.1.22.1 Offset

Register	Offset
PHY_REG21	54h

13.10.3.1.22.2 Diagram



13.10.3.1.22.3 Fields

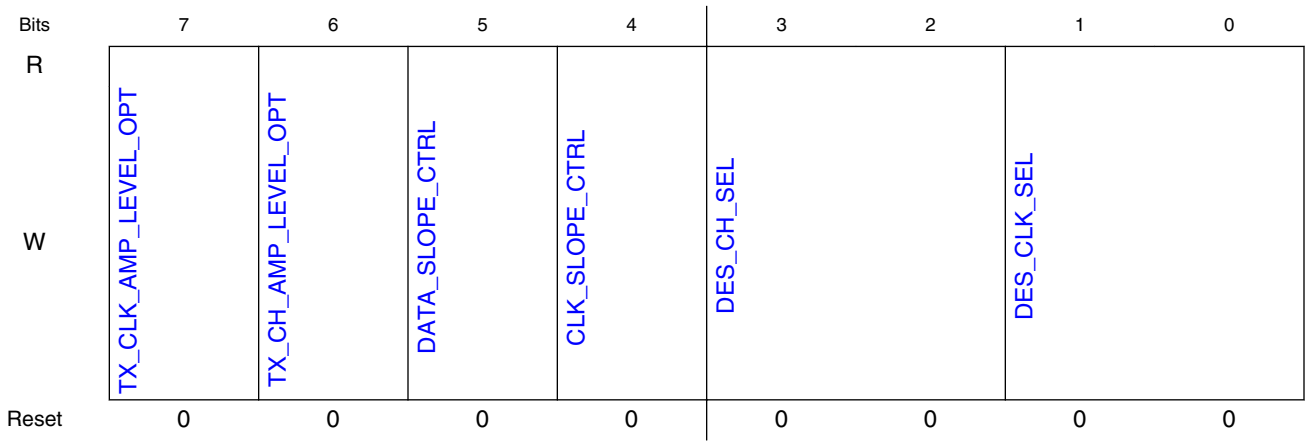
Field	Function
7 SEL_TX_CK_INV	Serializer parallel input clock inversion
6-4 PCLK_SEL	Pixel clock color depth 000b - 8bit 001b - 10bit 010b - 12bit
3-0 PMS_S	PCG post divider division ratio 0000b - 1 0001b - 2 0011b - 4 0101b - 6 0111b - 8 1001b - 10 1011b - 12 1111b - 16

13.10.3.1.23 HDMI TX PHY Register 22 (PHY_REG22)

13.10.3.1.23.1 Offset

Register	Offset
PHY_REG22	58h

13.10.3.1.23.2 Diagram



13.10.3.1.23.3 Fields

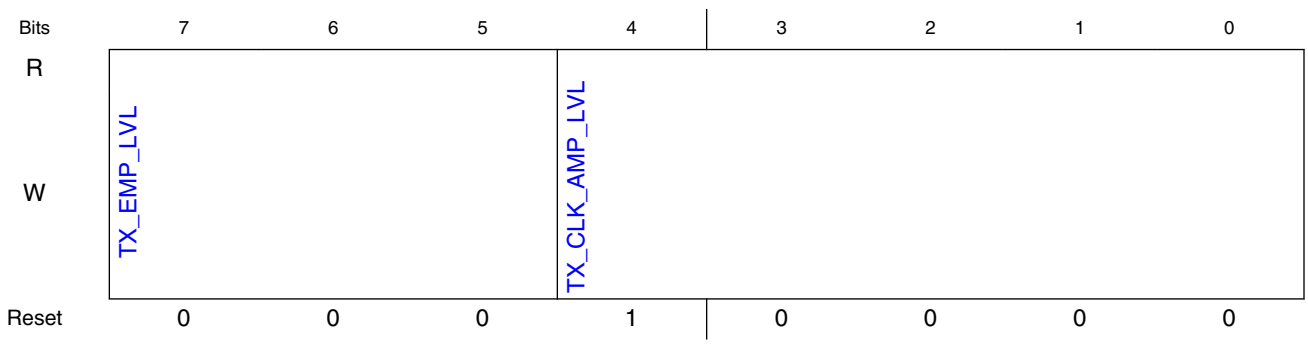
Field	Function
7 TX_CLK_AMP_LEVEL_OPT	Tx Clock channel option current cell enable(+8mA)
6 TX_CH_AMP_LEVEL_OPT	Tx Data channel option current cell enable(+8mA)
5 DATA_SLOPE_CTRL	Data channels slope control
4 CLK_SLOPE_CTRL	Clock channel slope control
3-2 DES_CH_SEL	Data channel selection for DFT 00b - CH0 01b - CH1 10b - CH2 11b - CLK
1-0 DES_CLK_SEL	Clock selection for DFT 00b - CLK_I 01b - CLK_IB_DEL 10b - CLK_QB_DEL 11b - CLK_Q

13.10.3.1.24 HDMI TX PHY Register 23 (PHY_REG23)

13.10.3.1.24.1 Offset

Register	Offset
PHY_REG23	5Ch

13.10.3.1.24.2 Diagram



13.10.3.1.24.3 Fields

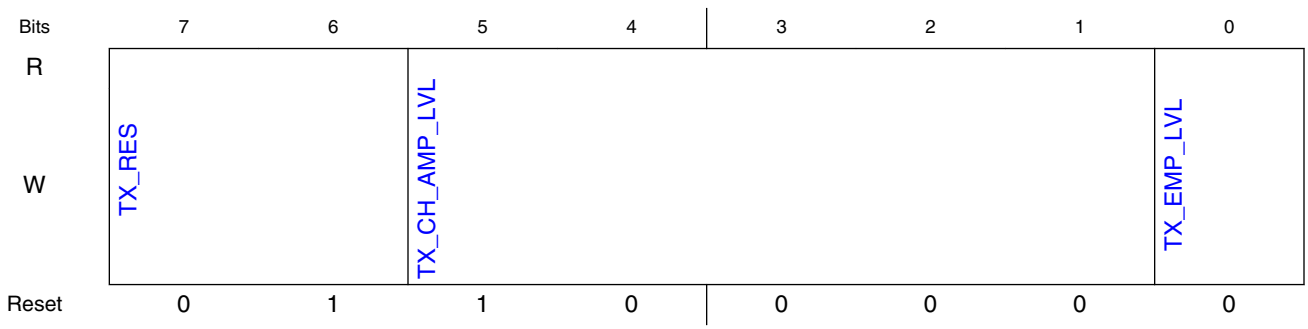
Field	Function
7-5 TX_EMP_LVL	TX data channel pre-emphasis control 0.5mA step: 0000 : 0mA (default) 1111=7.5mA For TX_EMP_LVL[3] bit, refer to HDMI TX PHY Register 24.
4-0 TX_CLK_AMP_LVL	TX clock channel amplitude control 0.5mA step: 00000 : 4mA ... 10000 : 12mA (default) ... 11111:19.5mA

13.10.3.1.25 HDMI TX PHY Register 24 (PHY_REG24)

13.10.3.1.25.1 Offset

Register	Offset
PHY_REG24	60h

13.10.3.1.25.2 Diagram



13.10.3.1.25.3 Fields

Field	Function
7-6 TX_RES	TX source termination resistor control 00b - OPEN 01b - 300Ohm 10b - 150Ohm 11b - 100Ohm
5-1 TX_CH_AMP_LVL	TX data channel amplitude control 0.5mA step: 00000 : 4mA 10000 : 12mA (Default) ... 11111: 19.5mA
0 TX_EMP_LVL	TX data channel pre-emphasis control Refer to HDMI TX PHY Register 23 for definiton.

13.10.3.1.26 HDMI TX PHY Register 25 (PHY_REG25)

13.10.3.1.26.1 Offset

Register	Offset
PHY_REG25	64h

13.10.3.1.26.2 Diagram

Bits	7	6	5	4	3	2	1	0
R								
W	SEL_RCLK_DIV10_PHASE	LB_EN			TX_CLK_SEL			
Reset	0	0	0	0	0	0	0	0

13.10.3.1.26.3 Fields

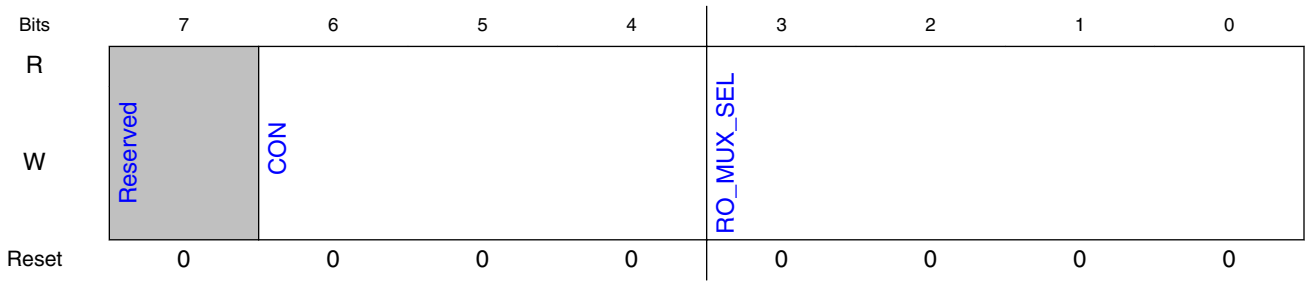
Field	Function
7 SEL_RCLK_DIV10_PHASE	Deserializer output clock inversion for BIST. Setting 1 to secure HVCC Margin during ATE BIST Shmoo Test (secure Hold Margin)
6-4 LB_EN	TX loopback Enable 000b - X 100b - CH0 101b - CH1 110b - CH2 111b - CLK
3-0 TX_CLK_SEL	TX main clock selection (I/IB or Q/QB) TX_CLK_SEL[3:0] = TXCP/CN, TX2P/2N, TX1P/1N, TX0P/0N For each bit: 0 : CKI/CKIB Sel 1 : CKQ/CKQB Sel Use Q to secure Shmoo margin during ATE BIST test

13.10.3.1.27 HDMI TX PHY Register 26 (PHY_REG26)

13.10.3.1.27.1 Offset

Register	Offset
PHY_REG26	68h

13.10.3.1.27.2 Diagram



13.10.3.1.27.3 Fields

Field	Function
7 —	Reserved
6-4 CON	
3-0 RO_MUX_SEL	

13.10.3.1.28 HDMI TX PHY Register 27 (PHY_REG27)

13.10.3.1.28.1 Offset

Register	Offset
PHY_REG27	6Ch

13.10.3.1.28.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	FLD_RSTN	SDC_RSTN	SDM_RSTN	NDIV_RSTN	MDIV_RSTN	PDIV_RSTN	AFC_PLL_RSTN	AFC_INIT_RSTN
W								
Reset	0	0	0	0	0	0	0	0

13.10.3.1.28.3 Fields

Field	Function
7 FLD_RSTN	Frequency Lock Detector RESETn
6 SDC_RSTN	SDC Gen. RESETn
5 SDM_RSTN	Sigma-Delta Modulation RESETn
4 NDIV_RSTN	SDC Gen. Divider RESETn
3 MDIV_RSTN	Main-Divider RESETn
2 PDIV_RSTN	Pre-Divider RESETn
1 AFC_PLL_RSTN	AFC PLL RESETn
0 AFC_INIT_RSTN	AFC RESETn

13.10.3.1.29 HDMI TX PHY Register 28 (PHY_REG28)

13.10.3.1.29.1 Offset

Register	Offset
PHY_REG28	70h

13.10.3.1.29.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	TXDRV_CH	TXDRV_CLK	SER_CH	SER_CLK	DES	PCG	BIAS	PLL
W								
Reset	0	0	0	0	0	0	0	0

13.10.3.1.29.3 Fields

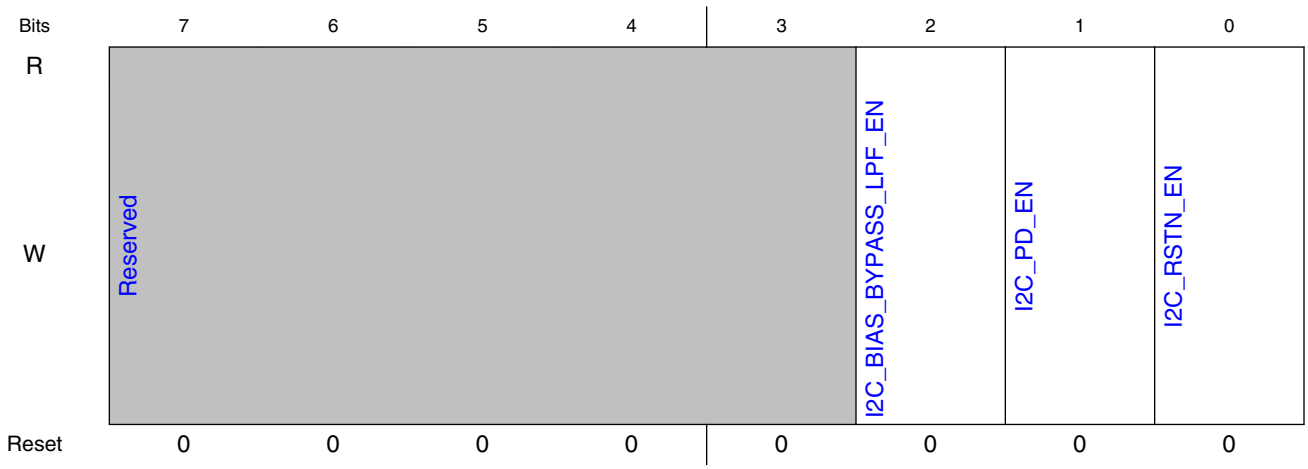
Field	Function
7 TXDRV_CH	TXDRV in Data Channel Power-Down
6 TXDRV_CLK	TXDRV in Clock Channel Power-Down
5 SER_CH	Serializer in Data Channel Power-Down
4 SER_CLK	Serializer in Clock Channel Power-Down
3 DES	Deserializer Power-Down
2 PCG	Pixel Clock Generator Power-Down
1 BIAS	BIAS Power-Down
0 PLL	PLL Power-Down

13.10.3.1.30 HDMI TX PHY Register 29 (PHY_REG29)

13.10.3.1.30.1 Offset

Register	Offset
PHY_REG29	74h

13.10.3.1.30.2 Diagram



13.10.3.1.30.3 Fields

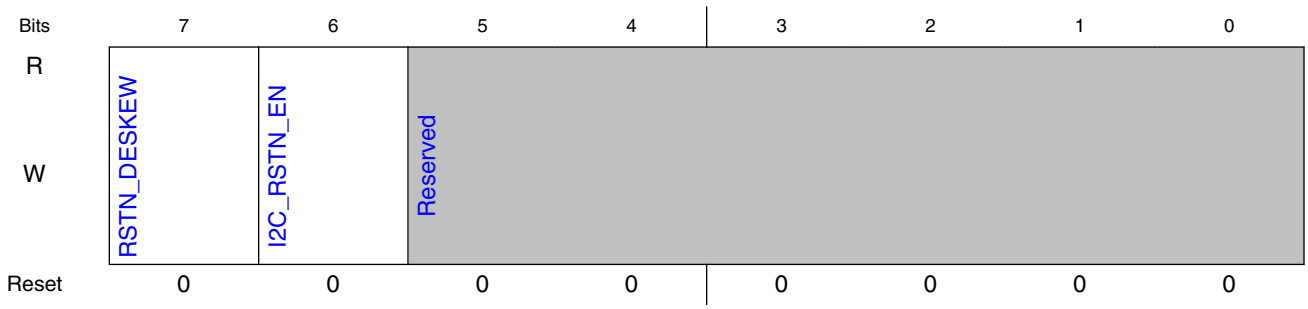
Field	Function
7-3 —	Reserved
2 I2C_BIAS_BYPASS_LPF_EN	BIAS_BYPASS_LPF_EN
1 I2C_PD_EN	PLL PD_EN
0 I2C_RSTN_EN	PLL RSTN_EN

13.10.3.1.31 HDMI TX PHY Register 30 (PHY_REG30)

13.10.3.1.31.1 Offset

Register	Offset
PHY_REG30	78h

13.10.3.1.31.2 Diagram



13.10.3.1.31.3 Fields

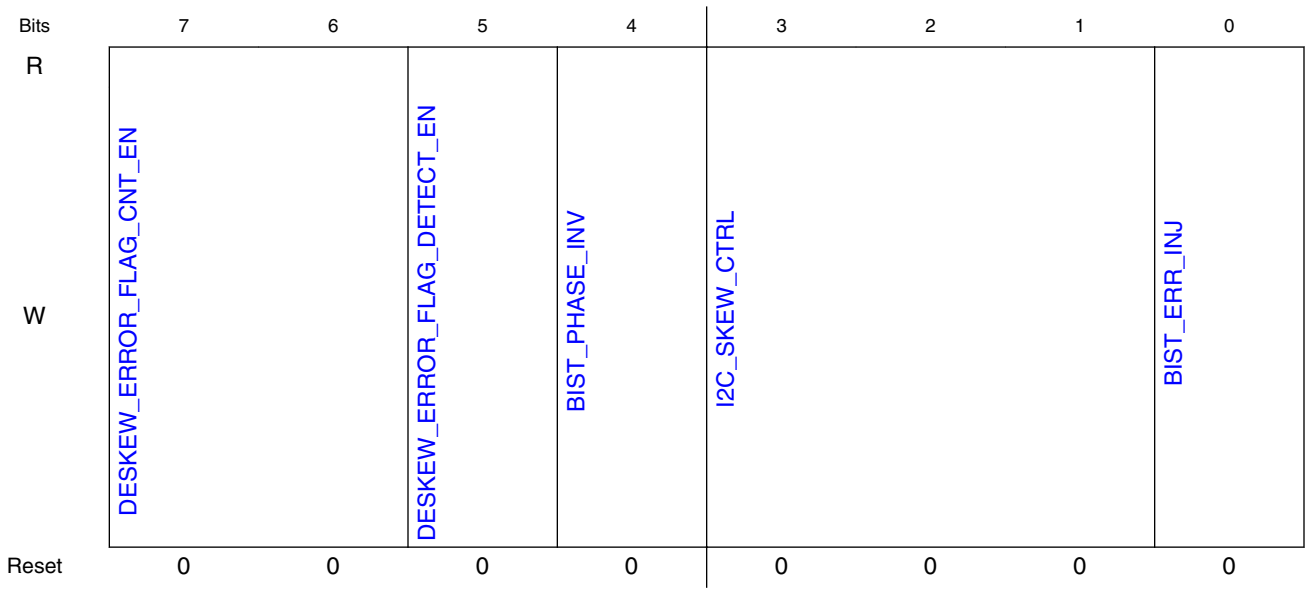
Field	Function
7 RSTN_DESKEW	DESKEW RSTN_DESKEW
6 I2C_RSTN_EN	DESKEW RSTN_EN 0b - normal reset sequence 1b - force reset assertion
5-0 —	Reserved

13.10.3.1.32 HDMI TX PHY Register 31 (PHY_REG31)

13.10.3.1.32.1 Offset

Register	Offset
PHY_REG31	7Ch

13.10.3.1.32.2 Diagram



13.10.3.1.32.3 Fields

Field	Function
7-6 DESKEW_ERROR_FLAG_CNT_EN	DESKEW_ERROR_FLAG_CNT_EN
5 DESKEW_ERROR_FLAG_DETECT_EN	DESKEW_ERROR_FLAG_DETECT_EN
4 BIST_PHASE_INV	BIST_PHASE_INV If the bit is 1'b1, BIST RX block will latch the de-serialized data from de-serializer with negative edge of sync clock. Else, de-serialized data will be latched by positive edge of clock.
3-1 I2C_SKEW_CTRL	Skew_CTRL for de-serializer if I2C_SKEW_CTRL[2] = 1'b1, I2C_SKEW_CTRL[1:0] will be automatically controlled by state machine in BIST logic. Else, user could control the value of phase control bit by access apb interface.
0 BIST_ERR_INJ	BIST_ERR_INJ Caution : User should not use 2 patterns (20'hF0F0F and 20'hCCCC) for BIST Operation with user defined pattern because it may cause false error, if the 20-bit user defined pattern and error injection pattern are matched. 0b - value of 20bit BIST TX pattern - 20'hf0f0f 1b - value of 20bit BIST TX pattern - 20'ccccc

13.10.3.1.33 HDMI TX PHY Register 32 (PHY_REG32)

13.10.3.1.33.1 Offset

Register	Offset
PHY_REG32	80h

13.10.3.1.33.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	TESTIN_EN	BIST_RX_EN	BIST_TX_EN	BIST_DATAPATH_EN	BIST_RX_RESETN	TESTOUT_EN	BIST_MODE	BIST_PATH_SEL
W								
Reset	0	0	0	0	0	0	0	0

13.10.3.1.33.3 Fields

Field	Function
7 TESTIN_EN	TESTIN_EN 0 : DOUT - 20bit data from TXD port 1 : DOUT - 20bit data from TESTIN port
6 BIST_RX_EN	BIST_RX_EN 0 : Disable BIST pattern comparison operation 1 : Enable BIST pattern comparison operation
5 BIST_TX_EN	BIST_TX_EN 0 : Disable BIST pattern generation operation 1 : Enable BIST pattern generation operation
4 BIST_DATAPATH_EN	BIST_DATAPATH_EN 0 : 20bit data from TXD port 1 : 20bit data from BIST pattern generation logic
3 BIST_RX_RESETN	BIST_RX_RESETN 0 : Assert reset for BIST pattern comparison logic 1 : Release reset for BIST pattern comparison logic

Table continues on the next page...

Field	Function
2 TESTOUT_EN	TESTOUT_EN 0 : TESOUT[10:11] - 2'b00 1 : TESTOUT[10] - Divided by 10 clock (source clock : PIXEL_CLK). TESTOUT[11] - Divided by 10 clock (source clock : TMDS_CLK).
1 BIST_MODE	BIST MODE 0 : PRBS7 1: User defined pattern
0 BIST_PATH_SEL	BIST_PATH_SEL 0 : bist_mode - mode[249] i2c_skew_ctrl - mode[243:241] bist_tx_en - mode[253] bist_rx_en - mode[254] bist_rx_resetn - mode[251] bist_data_path_en - mode[252] bist_err_inj - mode[240] bist_phase_inv - mode[244] 1 : bist_mode - TESTIN[0] i2c_skew_ctrl - TESTIN[3:1] bist_tx_en - TESTIN[4] bist_rx_en - TESTIN[5] bist_rx_resetn - TESTIN[6] bist_data_path_en - TESTIN[7] bist_err_inj - TESTIN[8] bist_phase_inv - TESTIN[9]

13.10.3.1.34 HDMI TX PHY Register 33 (PHY_REG33)

13.10.3.1.34.1 Offset

Register	Offset
PHY_REG33	84h

13.10.3.1.34.2 Diagram

Bits	7	6	5	4	3	2	1	0
R								
W	MODE_SET_DONE	TX_INV2	TX_INV1	TX_INV0	MON_RXD	FIX_DB	FIX_DA	ALTER
Reset	1	0	0	0	0	0	0	0

13.10.3.1.34.3 Fields

Field	Function
7 MODE_SET_DONE	MODE_SET_DONE 0b - Assert forced global reset 1b - Release forced global reset
6 TX_INV2	TX_INV2 0b - There are not any change in 20bit data from TXD2 port 1b - Invert polarity of 10bit data of 20bit data from TXD2 port
5 TX_INV1	TX_INV1 0b - There are not any change in 20bit data from TXD1 port 1b - Invert polarity of 10bit data of 20bit data from TXD1 port
4 TX_INV0	TX_INV0 0b - There are not any change in 20bit data from TXD0 port 1b - Invert polarity of 10bit data of 20bit data from TXD0 port
3 MON_RXD	MON_RXD MON_RXD 0 : TESTOUT[0] - BIST_ON TESTOUT[1] - BIST_ERROR TESTOUT[2] - DES_CLK_SEL[0] TESTOUT[3] - DES_CLK_SEL[1] TESTOUT[4] - TX_CLK_SEL[0] TESTOUT[5] - TX_CLK_SEL[1] TESTOUT[6] - TX_CLK_SEL[2] TESTOUT[7] - TX_CLK_SEL[3] TESTOUT[8] - PLL_COARSE_LOCK_DONE TESTOUT[9] - PLL_FINE_LOCK_DONE TESTOUT[10] - Divided by 10 clock (source clock : PIXEL_CLK) TESTOUT[11] - Divided by 10 clock (source clock : TMDS_CLK) 1 : TESTOUT[0:9] - 10bit of RXDATA TESTOUT[10] - Divided by 10 clock (source clock : PIXEL_CLK)

Table continues on the next page...

Field	Function
	TESTOUT[11] - Divided by 10 clock (source clock : TMDS_CLK)
2 FIX_DB	FIX_DB Disable de-skew function 1 - Select fixed phase 2
1 FIX_DA	FIX_DA Disable de-skew function 1 - Select fixed phase 1
0 ALTER	ALTER 0b - There are not any change in 20bit data from TESTIN port 1b - Invert polarity of 10bit data of 20bit data from TESTIN port

13.10.3.1.35 HDMI TX PHY Register 34 (PHY_REG34)

13.10.3.1.35.1 Offset

Register	Offset
PHY_REG34	88h

13.10.3.1.35.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	PHY_READY	PLL_LOCK	PHY_CLK_RDY	AFC_DONE	ATB_SEL			
W								
Reset	0	0	0	0	0	0	0	0

13.10.3.1.35.3 Fields

Field	Function
7 PHY_READY	Monitoring signal (PHY_READY)
6 PLL_LOCK	Monitoring signal (PLL_LOCK)
5 PHY_CLK_RDY	Monitoring signal (PHY_CLK_RDY)

Table continues on the next page...

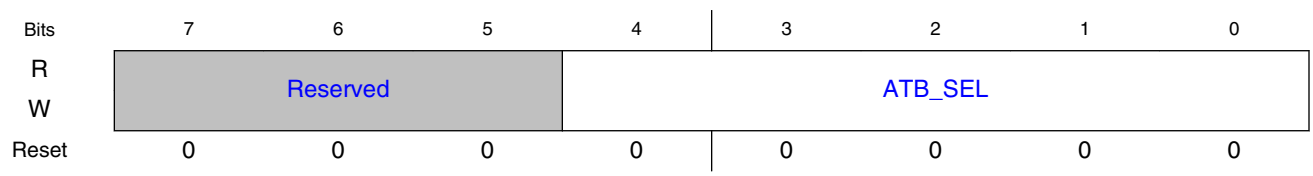
Field	Function
4 AFC_DONE	Monitoring signal (AFC_DONE)
3-0 ATB_SEL	Monitoring signal (AFC_CODE[3:0])

13.10.3.1.36 HDMI TX PHY Register 35 (PHY_REG35)

13.10.3.1.36.1 Offset

Register	Offset
PHY_REG35	8Ch

13.10.3.1.36.2 Diagram



13.10.3.1.36.3 Fields

Field	Function
7-5 —	Reserved
4-0 ATB_SEL	ATB_SEL

13.10.3.1.37 HDMI TX PHY Register 36 (PHY_REG36)

13.10.3.1.37.1 Offset

Register	Offset
PHY_REG36	90h

13.10.3.1.37.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	USR_PATTERN							
Reset	0	0	0	0	0	0	0	0

13.10.3.1.37.3 Fields

Field	Function
7-0 USR_PATTERN	USR_PATTERN[7:0] 40-bit user pattern[7:0]

13.10.3.1.38 HDMI TX PHY Register 37 (PHY_REG37)

13.10.3.1.38.1 Offset

Register	Offset
PHY_REG37	94h

13.10.3.1.38.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	USR_PATTERN							
Reset	0	0	0	0	0	0	0	0

13.10.3.1.38.3 Fields

Field	Function
7-0 USR_PATTERN	USR_PATTERN[15:8] 40-bit user pattern[15:8]

13.10.3.1.39 HDMI TX PHY Register 38 (PHY_REG38)

13.10.3.1.39.1 Offset

Register	Offset
PHY_REG38	98h

13.10.3.1.39.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	USR_PATTERN							
Reset	0	0	0	0	0	0	0	0

13.10.3.1.39.3 Fields

Field	Function
7-0 USR_PATTERN	USR_PATTERN[23:16] 40-bit user pattern[23:16]

13.10.3.1.40 HDMI TX PHY Register 39 (PHY_REG39)

13.10.3.1.40.1 Offset

Register	Offset
PHY_REG39	9Ch

13.10.3.1.40.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	USR_PATTERN							
Reset	0	0	0	0	0	0	0	0

13.10.3.1.40.3 Fields

Field	Function
7-0 USR_PATTERN	USR_PATTERN[31:24] 40-bit user pattern[31:24]

13.10.3.1.41 HDMI TX PHY Register 40 (PHY_REG40)

13.10.3.1.41.1 Offset

Register	Offset
PHY_REG40	A0h

13.10.3.1.41.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	USR_PATTERN							
Reset	0	0	0	0	0	0	0	0

13.10.3.1.41.3 Fields

Field	Function
7-0 USR_PATTERN	USR_PATTERN[39:32] 40-bit user pattern[39:32]]

13.10.3.1.42 HDMI TX PHY Register 41 (PHY_REG41)

13.10.3.1.42.1 Offset

Register	Offset
PHY_REG41	A4h

13.10.3.1.42.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	CLK_USR_PAT_10BIT							
Reset	1	1	1	0	0	0	0	0

13.10.3.1.42.3 Fields

Field	Function
7-0 CLK_USR_PAT_10BIT	CLK_USR_PAT_10BIT[7:0] 40bit clock user pattern[7:0]

13.10.3.1.43 HDMI TX PHY Register 42 (PHY_REG42)

13.10.3.1.43.1 Offset

Register	Offset
PHY_REG42	A8h

13.10.3.1.43.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	CLK_USR_PAT_10BIT							
Reset	1	0	0	0	0	0	1	1

13.10.3.1.43.3 Fields

Field	Function
7-0 CLK_USR_PAT_10BIT	CLK_USR_PAT_10BIT[15:8] 40bit clock user pattern[15:8]

13.10.3.1.44 HDMI TX PHY Register 43 (PHY_REG43)

13.10.3.1.44.1 Offset

Register	Offset
PHY_REG43	ACh

13.10.3.1.44.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	CLK_USR_PAT_10BIT							
Reset	0	0	0	0	1	1	1	1

13.10.3.1.44.3 Fields

Field	Function
7-0 CLK_USR_PAT_10BIT	CLK_USR_PAT_10BIT[23:16] 40bit clock user pattern[23:16]

13.10.3.1.45 HDMI TX PHY Register 44 (PHY_REG44)

13.10.3.1.45.1 Offset

Register	Offset
PHY_REG44	B0h

13.10.3.1.45.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	CLK_USR_PAT_10BIT							
Reset	0	0	1	1	1	1	1	0

13.10.3.1.45.3 Fields

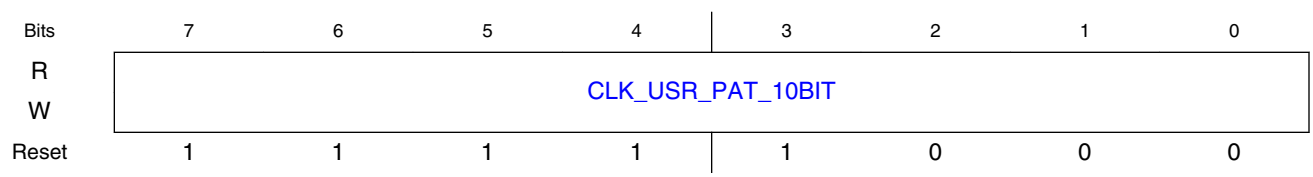
Field	Function
7-0 CLK_USR_PAT_10BIT	CLK_USR_PAT_10BIT[31:24] 40bit clock user pattern[31:24]

13.10.3.1.46 HDMI TX PHY Register 45 (PHY_REG45)

13.10.3.1.46.1 Offset

Register	Offset
PHY_REG45	B4h

13.10.3.1.46.2 Diagram



13.10.3.1.46.3 Fields

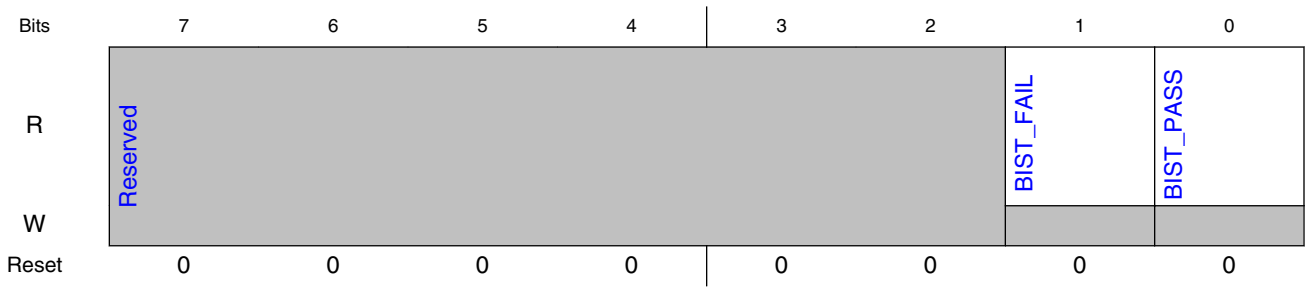
Field	Function
7-0 CLK_USR_PAT_10BIT	CLK_USR_PAT_10BIT[39:32] 40bit clock user pattern[39:32]

13.10.3.1.47 HDMI TX PHY Register 46 (PHY_REG46)

13.10.3.1.47.1 Offset

Register	Offset
PHY_REG46	B8h

13.10.3.1.47.2 Diagram



13.10.3.1.47.3 Fields

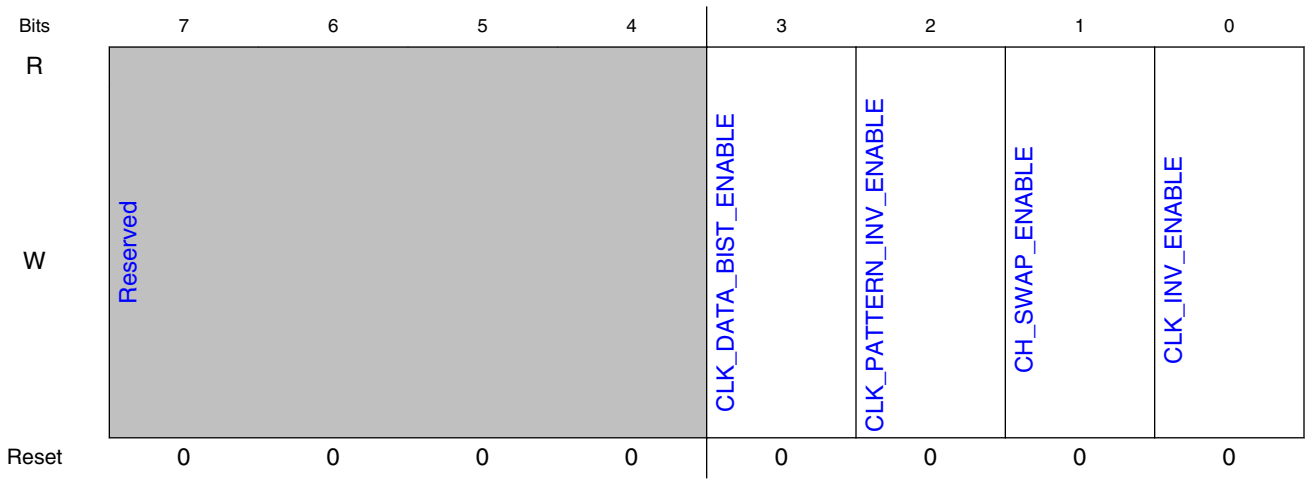
Field	Function
7-2 —	Reserved
1 BIST_FAIL	Monitoring signal(BIST_FAIL)
0 BIST_PASS	Monitoring signal(BIST_PASS)

13.10.3.1.48 HDMI TX PHY Register 47 (PHY_REG47)

13.10.3.1.48.1 Offset

Register	Offset
PHY_REG47	BCh

13.10.3.1.48.2 Diagram



13.10.3.1.48.3 Fields

Field	Function
7-4 —	Reserved
3 CLK_DATA_BIST_ENABLE	clock data bist enable
2 CLK_PATTERN_INV_ENABLE	Clock pattern inversion enable
1 CH_SWAP_ENABLE	TX channel swap enable
0 CLK_INV_ENABLE	TX deskew clock inversion enable

13.11 HDMI TX BLK_CTRL

13.11.1 Overview

The HDMIMIX domain contains control and status registers known as HDMI TX Block Control (HDMI TX BLK_CTRL). These registers have miscellaneous top-level controls for the HDMI peripherals within the HDMIMIX domain. Some of these controls includes

clocking, interrupt, and reset sources. Please see the clock control chapter and interrupt chapter for more information regarding the clock input sources and IRQ event mapping to HDMIMIX.

13.11.1.1 Block Diagram

The HDMI BLK_CTRL interface connects to the HDMIMIX domain peripherals. A block diagram illustrating the HDMIMIX domain is shown below.

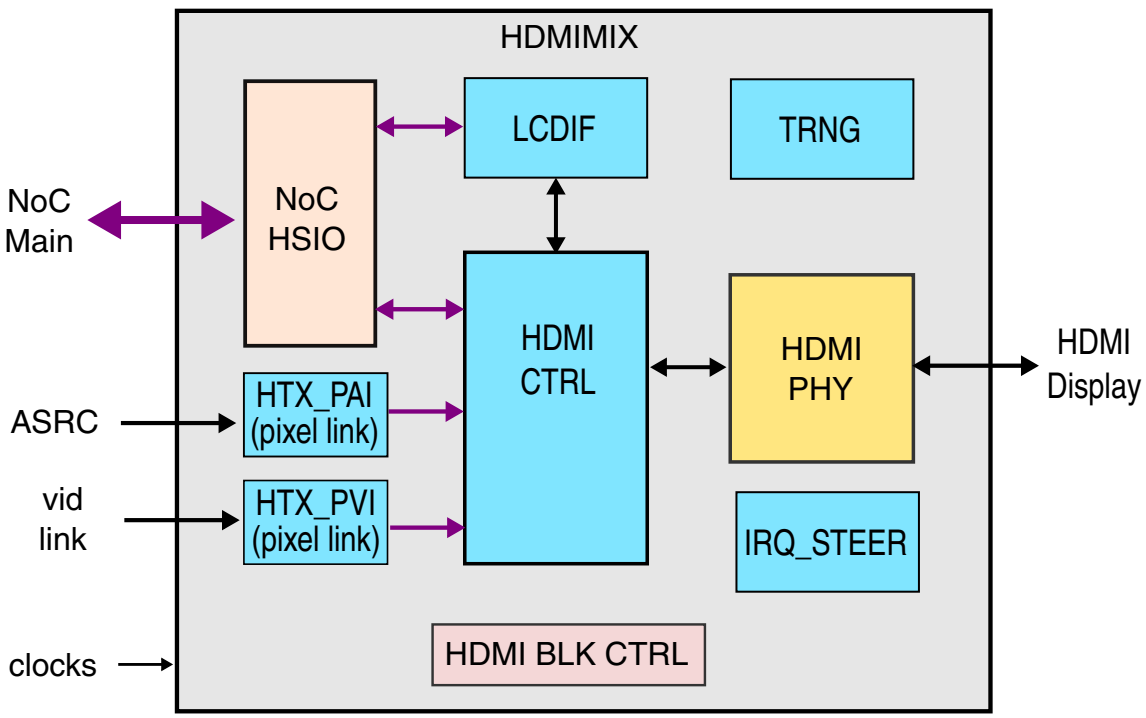


Figure 13-80. HDMIMIX Block Diagram

13.11.2 Clocks and Resets

The HDMI BLK_CTRL contains clock distribution and gating controls, as well as reset handling to several of the HDMIMIX peripherals. The clock gating controls include several layers of gating, which include Fail/Safe control. Please see related clock and reset controls in the BLK_CTRL register description. For more information about the input clocks, system-level resets, and IRQ to the domain, please see the related block guide and system-level assignments in the control blocks for clocks, reset, and IRQ assignments.

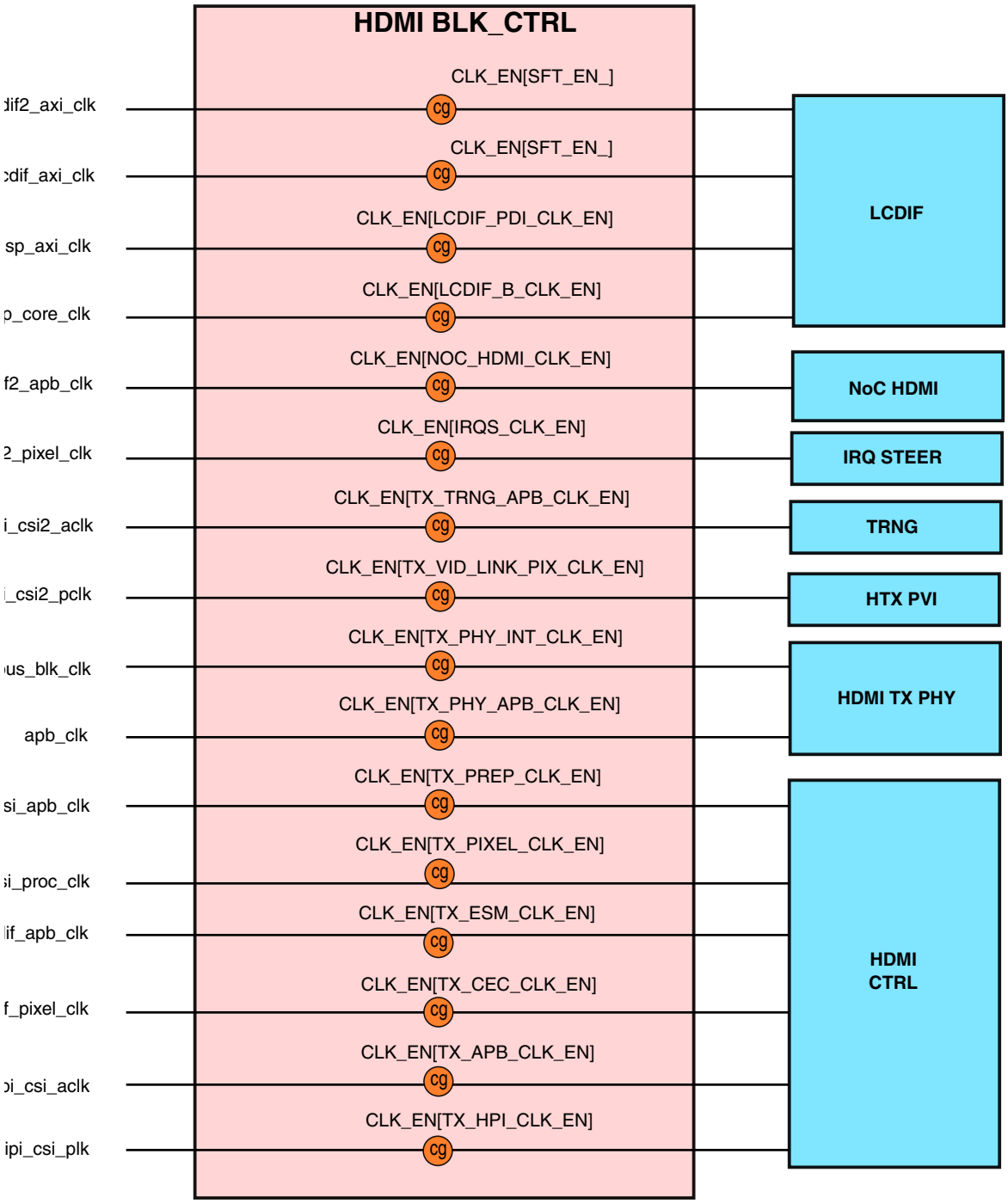


Figure 13-81. HDMI BLK_CTRL Clocks

13.11.3 Memory Map and Registers

13.11.3.1 register descriptions

13.11.3.1.1 HDMI20_TX_BLK_CTL memory map

HDMI_TX_BLK_CTRL base address: 32FC_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	HDMI_RTX_GENERAL CONFIG (RTX_GENERAL)	32	RW	0000_0000h
4h	HDMI_RTX_GENERAL CONFIG (RTX_GENERAL_SET)	32	RW	0000_0000h
8h	HDMI_RTX_GENERAL CONFIG (RTX_GENERAL_CLR)	32	RW	0000_0000h
Ch	HDMI_RTX_GENERAL CONFIG (RTX_GENERAL_TOG)	32	RW	0000_0000h
10h	HDMI_RTX_GENERAL CONFIG (RTX_GENERAL_1)	32	RW	0000_0000h
14h	HDMI_RTX_GENERAL CONFIG (RTX_GENERAL_1_SET)	32	RW	0000_0000h
18h	HDMI_RTX_GENERAL CONFIG (RTX_GENERAL_1_CLR)	32	RW	0000_0000h
1Ch	HDMI_RTX_GENERAL CONFIG (RTX_GENERAL_1_TOG)	32	RW	0000_0000h
20h	HDMI_RTX_RESET_CTL0 (RTX_RESET_CTL0)	32	RW	0000_0000h
24h	HDMI_RTX_RESET_CTL0 (RTX_RESET_CTL0_SET)	32	RW	0000_0000h
28h	HDMI_RTX_RESET_CTL0 (RTX_RESET_CTL0_CLR)	32	RW	0000_0000h
2Ch	HDMI_RTX_RESET_CTL0 (RTX_RESET_CTL0_TOG)	32	RW	0000_0000h
40h	HDMI_RTX_CLK_CTL0 (RTX_CLK_CTL0)	32	RW	0000_0000h
44h	HDMI_RTX_CLK_CTL0 (RTX_CLK_CTL0_SET)	32	RW	0000_0000h
48h	HDMI_RTX_CLK_CTL0 (RTX_CLK_CTL0_CLR)	32	RW	0000_0000h
4Ch	HDMI_RTX_CLK_CTL0 (RTX_CLK_CTL0_TOG)	32	RW	0000_0000h
50h	HDMI_RTX_CLK_CTL1 (RTX_CLK_CTL1)	32	RW	0000_0000h
54h	HDMI_RTX_CLK_CTL1 (RTX_CLK_CTL1_SET)	32	RW	0000_0000h
58h	HDMI_RTX_CLK_CTL1 (RTX_CLK_CTL1_CLR)	32	RW	0000_0000h
5Ch	HDMI_RTX_CLK_CTL1 (RTX_CLK_CTL1_TOG)	32	RW	0000_0000h
60h	RTX_CLK_CTL2 (RTX_CLK_CTL2)	32	RW	0000_0000h
64h	RTX_CLK_CTL2 (RTX_CLK_CTL2_SET)	32	RW	0000_0000h
68h	RTX_CLK_CTL2 (RTX_CLK_CTL2_CLR)	32	RW	0000_0000h
6Ch	RTX_CLK_CTL2 (RTX_CLK_CTL2_TOG)	32	RW	0000_0000h
70h	RTX_CLK_CTL3 (RTX_CLK_CTL3)	32	RW	0000_0000h
74h	RTX_CLK_CTL3 (RTX_CLK_CTL3_SET)	32	RW	0000_0000h
78h	RTX_CLK_CTL3 (RTX_CLK_CTL3_CLR)	32	RW	0000_0000h
7Ch	RTX_CLK_CTL3 (RTX_CLK_CTL3_TOG)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
80h	RTX_CLK_CTL4 (RTX_CLK_CTL4)	32	RW	0000_0000h
84h	RTX_CLK_CTL4 (RTX_CLK_CTL4_SET)	32	RW	0000_0000h
88h	RTX_CLK_CTL4 (RTX_CLK_CTL4_CLR)	32	RW	0000_0000h
8Ch	RTX_CLK_CTL4 (RTX_CLK_CTL4_TOG)	32	RW	0000_0000h
90h	HDMI_RX_Control (RTX_IRQ_MASK)	32	RW	0000_0000h
94h	HDMI_RX_Control (RTX_IRQ_MASK_SET)	32	RW	0000_0000h
98h	HDMI_RX_Control (RTX_IRQ_MASK_CLR)	32	RW	0000_0000h
9Ch	HDMI_RX_Control (RTX_IRQ_MASK_TOG)	32	RW	0000_0000h
A0h	HDMI_TX Masked Interrupt status (RTX_IRQ_MASKED_STATUS)	32	RO	0000_0000h
A4h	HDMI_TX Masked Interrupt status (RTX_IRQ_MASKED_STATUS_SET)	32	RO	0000_0000h
A8h	HDMI_TX Masked Interrupt status (RTX_IRQ_MASKED_STATUS_CLR)	32	RO	0000_0000h
ACh	HDMI_TX Masked Interrupt status (RTX_IRQ_MASKED_STATUS_TOG)	32	RO	0000_0000h
B0h	HDMI_RX_Control (RTX_IRQ_NONMASK_STATUS)	32	W1C	0000_0000h
B4h	HDMI_RX_Control (RTX_IRQ_NONMASK_STATUS_SET)	32	W1C	0000_0000h
B8h	HDMI_RX_Control (RTX_IRQ_NONMASK_STATUS_CLR)	32	W1C	0000_0000h
BCh	HDMI_RX_Control (RTX_IRQ_NONMASK_STATUS_TOG)	32	W1C	0000_0000h
200h	Miscellaneous Controls for the HDMI TX Controller (TX_CONTROL0)	32	RW	0000_0018h
204h	Miscellaneous Controls for the HDMI TX Controller (TX_CONTROL0_SET)	32	RW	0000_0018h
208h	Miscellaneous Controls for the HDMI TX Controller (TX_CONTROL0_CLR)	32	RW	0000_0018h
20Ch	Miscellaneous Controls for the HDMI TX Controller (TX_CONTROL0_TOG)	32	RW	0000_0018h
220h	TX Control (TX_CONTROL2)	32	RW	0000_0000h
224h	TX Control (TX_CONTROL2_SET)	32	RW	0000_0000h
228h	TX Control (TX_CONTROL2_CLR)	32	RW	0000_0000h
22Ch	TX Control (TX_CONTROL2_TOG)	32	RW	0000_0000h
230h	Status (TX_STATUS0)	32	RO	0000_0000h
234h	Status (TX_STATUS0_SET)	32	RO	0000_0000h
238h	Status (TX_STATUS0_CLR)	32	RO	0000_0000h
23Ch	Status (TX_STATUS0_TOG)	32	RO	0000_0000h

13.11.3.1.2 HDMI_RTX_GENERAL CONFIG (RTX_GENERAL)

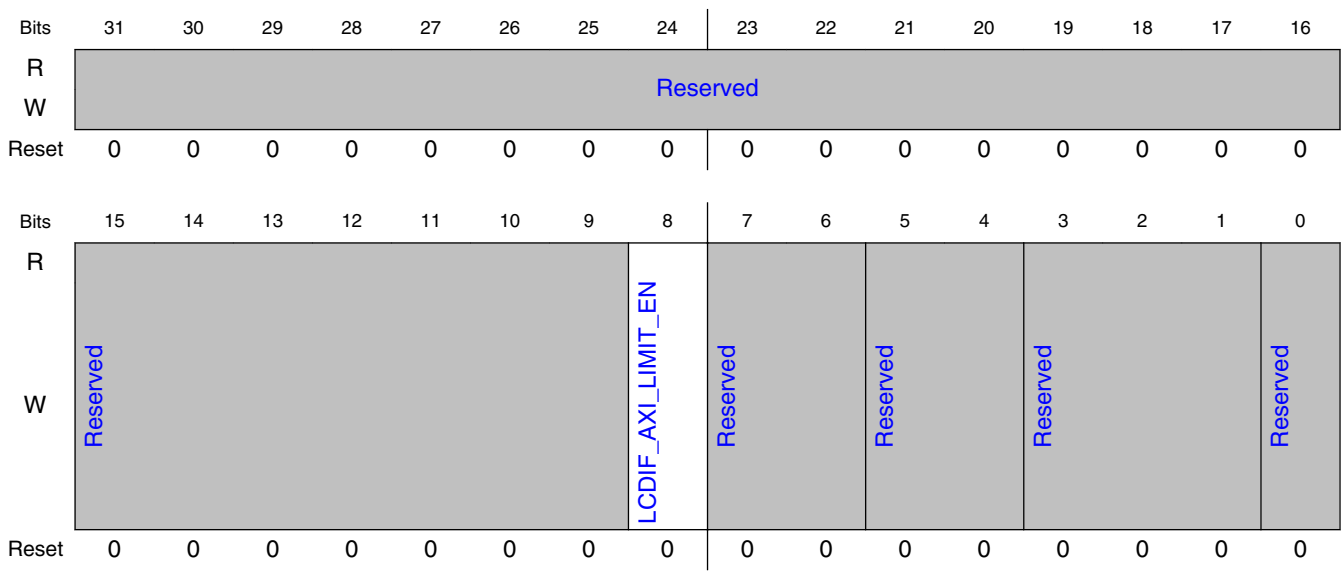
13.11.3.1.2.1 Offset

Register	Offset	Description
RTX_GENERAL	0h	HDMI_RTX_GENERAL CONFIG
RTX_GENERAL_SET	4h	Writing a 1 to a bit in this register sets the corresponding bit in RTX_GENERAL
RTX_GENERAL_CLR	8h	Writing a 1 to a bit in this register clears the corresponding bit in RTX_GENERAL
RTX_GENERAL_TOG	Ch	Writing a 1 to a bit in this register toggles the corresponding bit in RTX_GENERAL

13.11.3.1.2.2 Function

HDMI_RTX_GENERAL CONFIG

13.11.3.1.2.3 Diagram



13.11.3.1.2.4 Fields

Field	Function
31-9 —	Reserved
8 LCDIF_AXI_LIMIT_EN	Enables the AXI Read Beat count limiter; the beat limit value is given by the 16b value LCDIF_AXI_BEAT_LIMIT
7-6	Reserved

Table continues on the next page...

Field	Function
—	
5-4 —	Reserved
3-1 —	Reserved
0 —	Reserved

13.11.3.1.3 HDMI_RTX_GENERAL CONFIG (RTX_GENERAL_1)

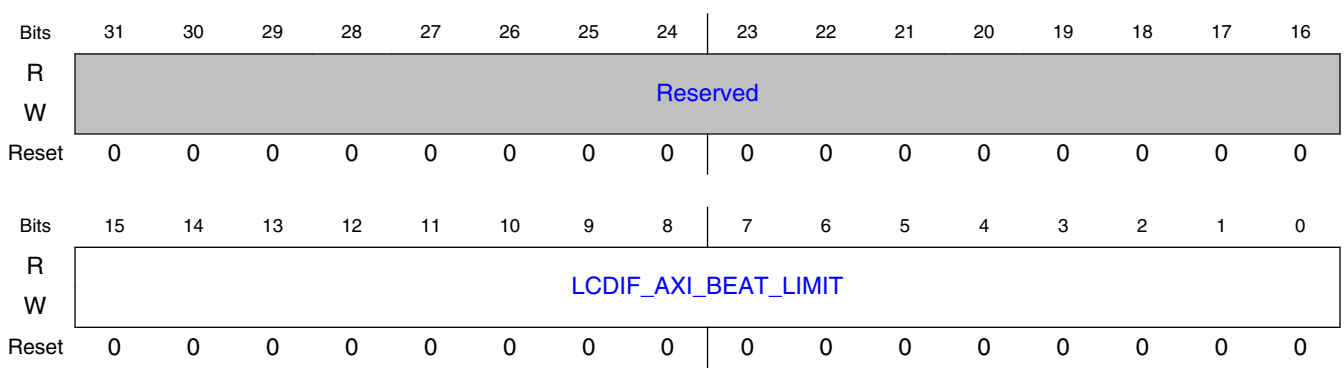
13.11.3.1.3.1 Offset

Register	Offset	Description
RTX_GENERAL_1	10h	HDMI_RTX_GENERAL CONFIG
RTX_GENERAL_1_SET	14h	Writing a 1 to a bit in this register sets the corresponding bit in RTX_GENERAL_1
RTX_GENERAL_1_CLR	18h	Writing a 1 to a bit in this register clears the corresponding bit in RTX_GENERAL_1
RTX_GENERAL_1_TOG	1Ch	Writing a 1 to a bit in this register toggles the corresponding bit in RTX_GENERAL_1

13.11.3.1.3.2 Function

HDMI_RTX_GENERAL CONFIG

13.11.3.1.3.3 Diagram



13.11.3.1.3.4 Fields

Field	Function
31-16 —	Reserved
15-0 LCDIF_AXI_BEAT_LIMIT	LCDIF_AXI_BEAT_LIMIT Count limit for the number of beats of read request that LCDIF master can issue This value should be programmed before enabling LCDIF_AXI_LIMIT_EN

13.11.3.1.4 HDMI_RTX_RESET_CTL0 (RTX_RESET_CTL0)

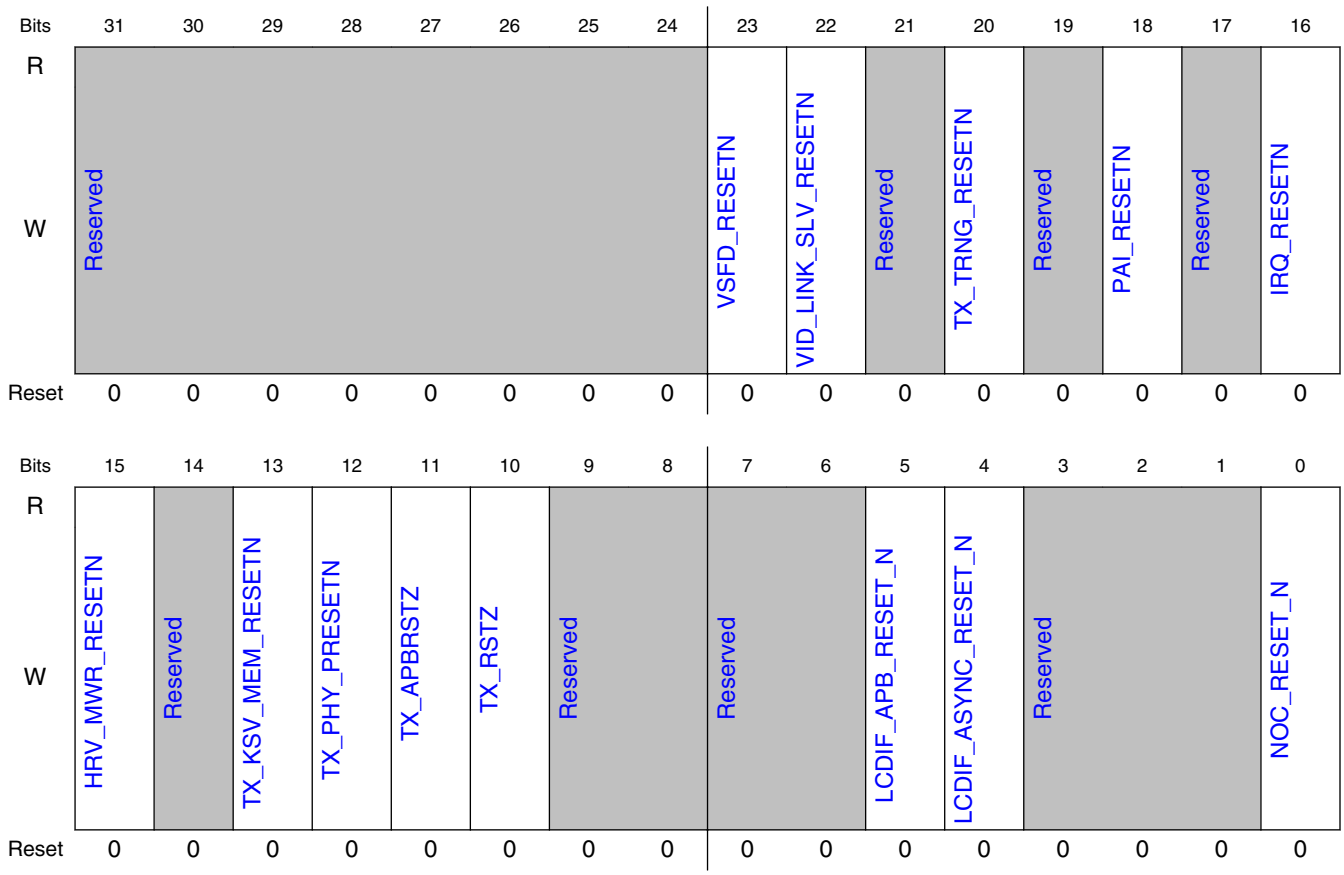
13.11.3.1.4.1 Offset

Register	Offset	Description
RTX_RESET_CTL0	20h	HDMI_RTX_RESET_CTL0
RTX_RESET_CTL0_SET	24h	Writing a 1 to a bit in this register sets the corresponding bit in RTX_RESET_CTL0
RTX_RESET_CTL0_CLR	28h	Writing a 1 to a bit in this register clears the corresponding bit in RTX_RESET_CTL0
RTX_RESET_CTL0_TOG	2Ch	Writing a 1 to a bit in this register toggles the corresponding bit in RTX_RESET_CTL0

13.11.3.1.4.2 Function

HDMI_RTX_RESET_CTL0

13.11.3.1.4.3 Diagram



13.11.3.1.4.4 Fields

Field	Function
31-24 —	Reserved
23 VSFD_RESETN	VSFD_RESETN control Active low soft reset to the VSFD block
22 VID_LINK_SLV_RESETN	VID_LINK_SLV_RESETN control Active low Asynchronous soft reset to the hdmi_tx_chip_reset_n input of PVI block
21 —	Reserved
20 TX_TRNG_RESETN	TX_TRNG_RESETN control Active low Asynchronous soft reset to the presetn input of TRNG block
19 —	Reserved

Table continues on the next page...

Field	Function
18 PAI_RESETN	PAI_RESETN control Activ low soft reset to the PAI block
17 —	Reserved
16 IRQ_RESETN	IRQ_RESETN control Active low soft reset to the IRQ_Steer block
15 HRV_MWR_RESETN	HRV_MWR_RESETN control Active low soft reset to the HRV_MWR block
14 —	Reserved
13 TX_KSV_MEM_RESETN	KSV Mem reset control This bit is used to control the reset of the KSV mem
12 TX_PHY_PRESETN	TX_PHY_PRESETN control Active low Soft reset to I_PRESETn input of HDMI TX PHY
11 TX_APBRSTZ	TX_APBRSTZ control Active low soft reset to iapbrstz input of HDMI20 TX Controller
10 TX_RSTZ	TX_RSTZ control Active low Soft reset to irstz input of HDMI20 TX Controller
9-8 —	Reserved
7-6 —	Reserved
5 LCDIF_APB_RESET_N	LCDIF_APB_RESET_N control Active low Soft reset to the apb_reset_n input of LCDIF
4 LCDIF_ASYNC_RESET_N	LCDIF_ASYNC_RESET_N control Active low Soft reset to the reset_n input of LCDIF
3-1 —	Reserved
0 NOC_RESET_N	NOC_RESET_N control Active low Soft reset to the reset_n input of NOC This same reset is synchronized to the respective clock domain and fed to NOC

13.11.3.1.5 HDMI_RTX_CLK_CTL0 (RTX_CLK_CTL0)

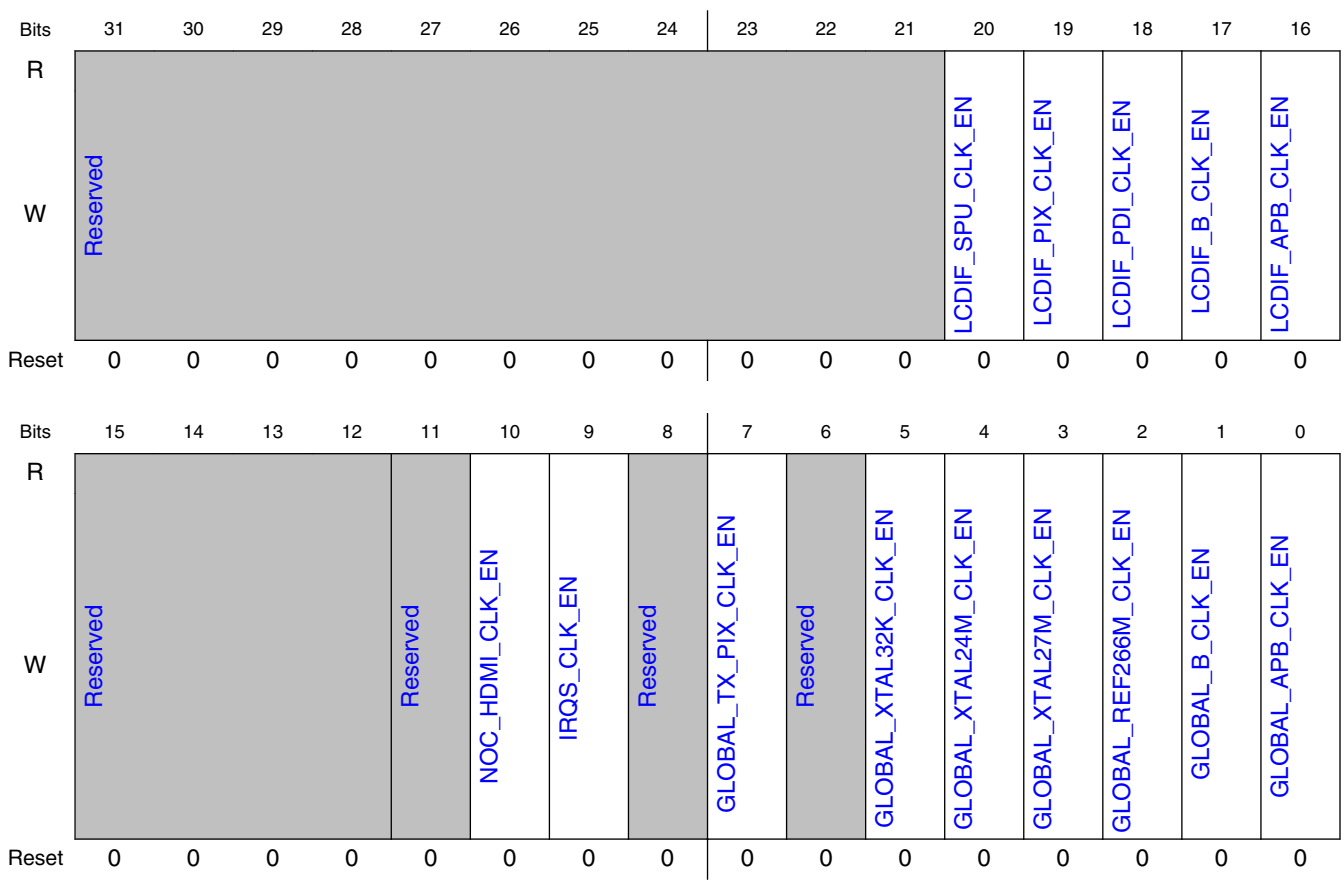
13.11.3.1.5.1 Offset

Register	Offset	Description
RTX_CLK_CTL0	40h	HDMI_RTX_CLK_CTL0
RTX_CLK_CTL0_SET	44h	Writing a 1 to a bit in this register sets the corresponding bit in RTX_CLK_CTL0
RTX_CLK_CTL0_CLR	48h	Writing a 1 to a bit in this register clears the corresponding bit in RTX_CLK_CTL0
RTX_CLK_CTL0_TOG	4Ch	Writing a 1 to a bit in this register toggles the corresponding bit in RTX_CLK_CTL0

13.11.3.1.5.2 Function

Fuse takes the highest priority in clock control; if a function is disabled through fuse all related clocks are gated (disabled) Second priority is given to always-on/always-off condition (registers related to this are mentioned below) Third priority is given to Level1 clock enable - global clock controls Fourth priority is given to Level2 clock enable which is programmed in BLK_CTL in registers mentioned below Fifth priority is given to Level3 clock enable provided by the corresponding blocks for their idle condition

13.11.3.1.5.3 Diagram



13.11.3.1.5.4 Fields

Field	Function
31-21 —	Reserved
20 LCDIF_SPU_CLK_EN	clock enable for lcdif spu_clk input
19 LCDIF_PIX_CLK_EN	clock enable for lcdif pix_clk input
18 LCDIF_PDI_CLK_EN	clock enable for lcdif pdi_clk input
17 LCDIF_B_CLK_EN	clock enable for lcdif bus_clk input
16 LCDIF_APB_CLK_EN	clock enable for lcdif apb_clk input
15-12 —	Reserved
11 —	Reserved
10 NOC_HDMI_CLK_EN	clock enable for the NOC bus_clk enable
9 IRQS_CLK_EN	clock control for the irq_steer block
8 —	Reserved
7 GLOBAL_TX_PIX_CLK_EN	TX pix clk control global pixel clock enable
6 —	Reserved
5 GLOBAL_XTAL32K_CLK_EN	GLOBAL_XTAL32K_CLK_EN control Active High Global clock enable for XTAL 32KHz clock
4 GLOBAL_XTAL24M_CLK_EN	GLOBAL_XTAL24M_CLK_EN control Active High Global clock enable for XTAL 24MHz clock

Table continues on the next page...

Field	Function
3 GLOBAL_XTAL27M_CLK_EN	GLOBAL_XTAL27M_CLK_EN control Active High Global clock enable for XTAL 27MHz clock
2 GLOBAL_REF266M_CLK_EN	GLOBAL_REF266M_CLK_EN control Active High Global clock enable for 266MHz reference clock
1 GLOBAL_B_CLK_EN	GLOBAL_B_CLK_EN control Active High Global clock enable for b_clk (bus clock) input
0 GLOBAL_APB_CLK_EN	GLOBAL_APB_CLK_EN control Active High Global clock Enable for apb_clk input of all blocks; BLK_CTL clock is not gated by this control since blk_ctl is required to be accessible at all times

13.11.3.1.6 HDMI_RTX_CLK_CTL1 (RTX_CLK_CTL1)

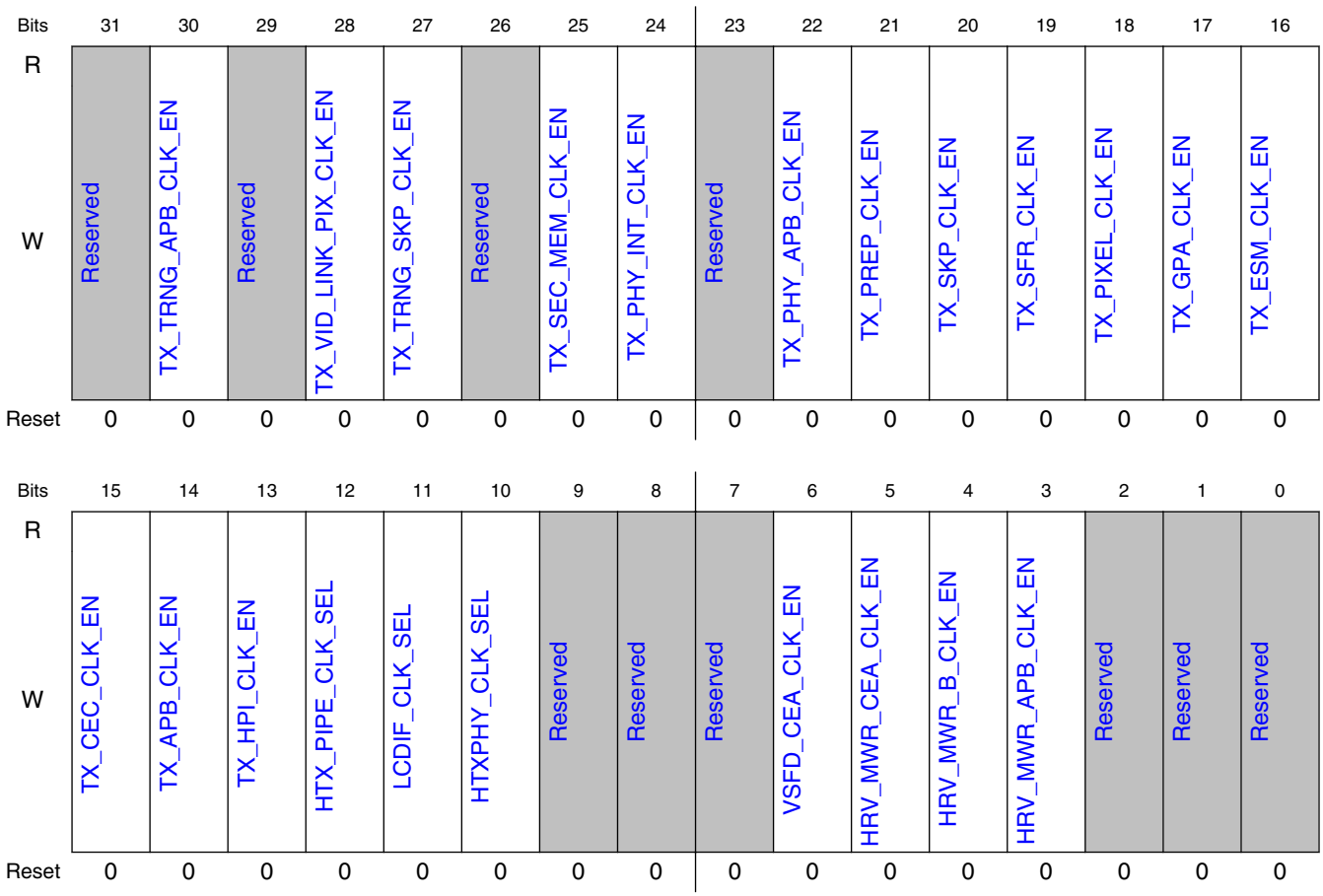
13.11.3.1.6.1 Offset

Register	Offset	Description
RTX_CLK_CTL1	50h	HDMI_RTX_CLK_CTL1
RTX_CLK_CTL1_SET	54h	Writing a 1 to a bit in this register sets the corresponding bit in RTX_CLK_CTL1
RTX_CLK_CTL1_CLR	58h	Writing a 1 to a bit in this register clears the corresponding bit in RTX_CLK_CTL1
RTX_CLK_CTL1_TOG	5Ch	Writing a 1 to a bit in this register toggles the corresponding bit in RTX_CLK_CTL1

13.11.3.1.6.2 Function

HDMI_RTX_CLK_CTL1

13.11.3.1.6.3 Diagram



13.11.3.1.6.4 Fields

Field	Function
31 —	Reserved
30 TX_TRNG_APB_CLK_EN	TX_TRNG_APB_CLK_EN control clock enable for the apb_clk input of the TRNG block
29 —	Reserved
28 TX_VID_LINK_PIX_CLK_EN	TX_VID_LINK_PIX_CLK_EN control video link (PVI) pixel clock enable
27 TX_TRNG_SKP_CLK_EN	TX_TRNG_SKP_CLK_EN control clock enable for the skp_clk input of the TRNG block
26	Reserved

Table continues on the next page...

Field	Function
—	
25 TX_SEC_MEM_CLK_EN	TX_SEC_MEM_CLK_EN control apb_clk enable for the HTX sec_mem region (outside the TX controller)
24 TX_PHY_INT_CLK_EN	TX_PHY_INT_CLK_EN control clock enable for the int_clk input of TX PHY
23 —	Reserved
22 TX_PHY_APB_CLK_EN	TX_PHY_APB_CLK_EN control clock enable for the apb_clk input of TX PHY
21 TX_PREP_CLK_EN	TX_PREP_CLK_EN control clock enable for the prep clock input of TX controller
20 TX_SKP_CLK_EN	TX_SKP_CLK_EN control clock enable for the skp clock input of TX controller
19 TX_SFR_CLK_EN	TX_SFR_CLK_EN control clock enable for the sfr clock input of TX controller
18 TX_PIXEL_CLK_EN	TX_PIXEL_CLK_EN control clock enable for the TX Controller input ipixelclk
17 TX_GPA_CLK_EN	TX_GPA_CLK_EN control clock enable for the gpa clock input of TX controller
16 TX_ESM_CLK_EN	TX_ESM_CLK_EN control clock enable for the esm clock input of TX controller
15 TX_CEC_CLK_EN	TX_CEC_CLK_EN control clock enable for the cec_clk input of TX controller
14 TX_APB_CLK_EN	TX_APB_CLK_EN control clock enable for the apb_clk input of TX controller
13 TX_HPI_CLK_EN	TX_HPI_CLK_EN control clock enable for the hpi_clk input of TX controller
12 HTX_PIPE_CLK_SEL	HTX_PIPE_CLK_SEL control this is to select between Frac PLL clock output and the PIXEL_CLK_OUT output from PHY
11	LCDIF_CLK_SEL control this is to select between output of the full integer divider clock and test_clk input

Table continues on the next page...

Field	Function
LCDIF_CLK_SEL	
10 HTXPHY_CLK_SEL	HTXPHY_CLK_SEL control this is to select between 24MHz clock and VPLL_PHY clock to drive INT_CLK input of PHY int_clk input of the phy is the pixel clock reference
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 VSFD_CEA_CLK_EN	VSFD_CEA_CLK_EN control clock enable for the cea clk input of PLB
5 HRV_MWR_CEA_CLK_EN	HRV_MWR_CEA_CLK_EN control clock enable for the cea clk input of HRV_MWR
4 HRV_MWR_B_CLK_EN	HRV_MWR_B_CLK_EN control clock enable for the bus clk input of HRV_MWR
3 HRV_MWR_APB_CLK_EN	HRV_MWR_APB_CLK_EN control clock enable for the apb_clk input of HRV_MWR block
2 —	Reserved
1 —	Reserved
0 —	Reserved

13.11.3.1.7 RTX_CLK_CTL2 (RTX_CLK_CTL2)

13.11.3.1.7.1 Offset

Register	Offset	Description
RTX_CLK_CTL2	60h	RTX_CLK_CTL2
RTX_CLK_CTL2_SET	64h	Writing a 1 to a bit in this register sets the corresponding bit in RTX_CLK_CTL2
RTX_CLK_CTL2_CLR	68h	Writing a 1 to a bit in this register clears the corresponding bit in RTX_CLK_CTL2

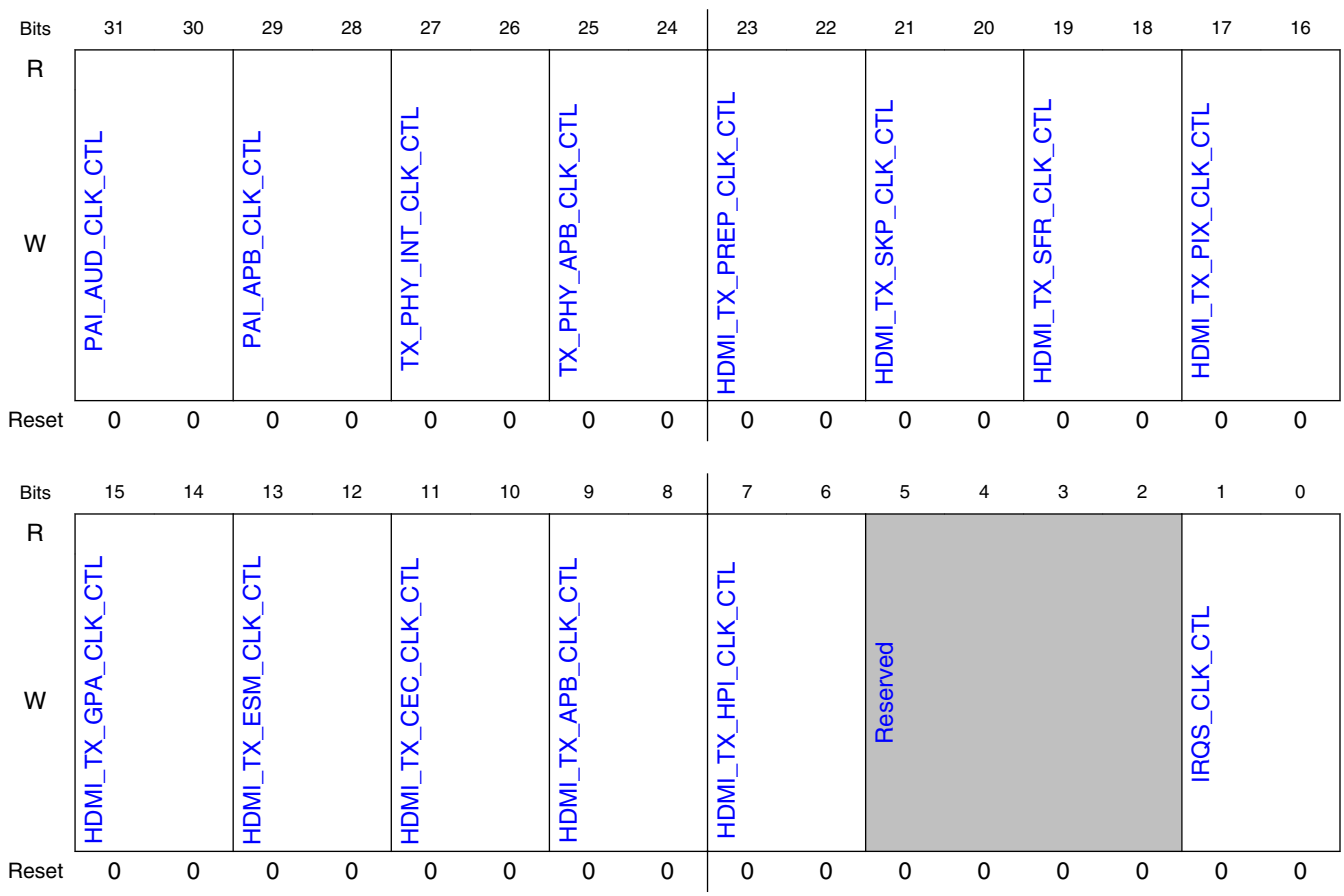
Table continues on the next page...

Register	Offset	Description
RTX_CLK_CTL2_TOG	6Ch	Writing a 1 to a bit in this register toggles the corresponding bit in RTX_CLK_CTL2

13.11.3.1.7.2 Function

This register is used to control the always-on/always-off condition of clocks as fail-safe Fuse takes the highest priority; Second priority is given to always-on/always-off condition Third priority is given to Level2 clock enable which is programmed in BLK_CTL in registers mentioned above Fourth (least) priority is given to Level3 clock enable provided by the corresponding blocks for their idle condition 00 - Default clock gating logic 01 - always clock ON 10 - always clock OFF 11 - Reserved

13.11.3.1.7.3 Diagram



13.11.3.1.7.4 Fields

Field	Function
31-30 PAI_AUD_CLK_CTL	Used to bypass the programmable clock controls for aud_clk input of HTX_PAI
29-28 PAI_APB_CLK_CTL	Used to bypass the programmable clock controls for apb_clk input of HTX_PAI
27-26 TX_PHY_INT_CLK_CTL	Used to bypass the programmable clock controls for int_clk input of the HDMI TX PHY
25-24 TX_PHY_APB_CLK_CTL	Used to bypass the programmable clock controls for apb_clk input of the HDMI TX PHY
23-22 HDMI_TX_PREP_CLK_CTL	Used to bypass the programmable clock controls for prep_clk input of HDMI TX
21-20 HDMI_TX_SKP_CLK_CTL	Used to bypass the programmable clock controls for skp_clk input of HDMI TX
19-18 HDMI_TX_SFR_CLK_CTL	Used to bypass the programmable clock controls for sfr_clk input of HDMI TX
17-16 HDMI_TX_PIX_CLK_CTL	Used to bypass the programmable clock controls for ipixel_clk input of HDMI TX
15-14 HDMI_TX_GPA_CLK_CTL	Used to bypass the programmable clock controls for gpa_clk input of HDMI TX
13-12 HDMI_TX_ESM_CLK_CTL	Used to bypass the programmable clock controls for esm_clk input of HDMI TX
11-10 HDMI_TX_CEC_CLK_CTL	Used to bypass the programmable clock controls for cec_clk input of HDMI TX
9-8 HDMI_TX_APB_CLK_CTL	Used to bypass the programmable clock controls for apb_clk input of HDMI TX
7-6 HDMI_TX_HPI_CLK_CTL	Used to bypass the programmable clock controls for hpi_clk input of HDMI TX
5-2 —	Reserved
1-0 IRQS_CLK_CTL	Used to bypass the programmable clock controls for IRQ_STEER

13.11.3.1.8 RTX_CLK_CTL3 (RTX_CLK_CTL3)

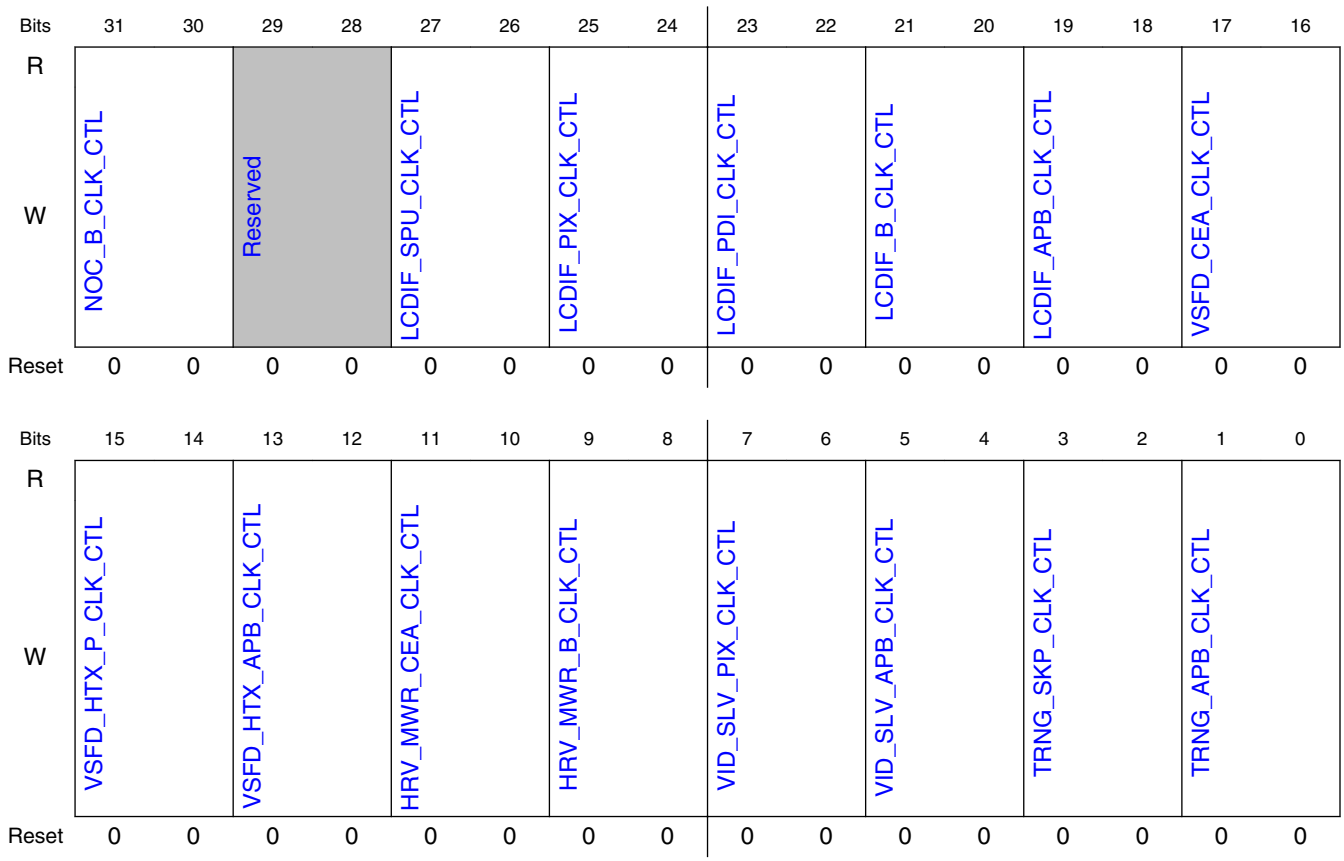
13.11.3.1.8.1 Offset

Register	Offset	Description
RTX_CLK_CTL3	70h	RTX_CLK_CTL3
RTX_CLK_CTL3_SET	74h	Writing a 1 to a bit in this register sets the corresponding bit in RTX_CLK_CTL3
RTX_CLK_CTL3_CLR	78h	Writing a 1 to a bit in this register clears the corresponding bit in RTX_CLK_CTL3
RTX_CLK_CTL3_TOG	7Ch	Writing a 1 to a bit in this register toggles the corresponding bit in RTX_CLK_CTL3

13.11.3.1.8.2 Function

This register is used to control the always-on/always-off condition of clocks as fail-safe Fuse takes the highest priority; Second priority is given to always-on/always-off condition Third priority is given to Level2 clock enable which is programmed in BLK_CTL in registers mentioned above Fourth (least) priority is given to Level3 clock enable provided by the corresponding blocks for their idle condition 00 - Default clock gating logic 01 - always clock ON 10 - always clock OFF 11 - Reserved

13.11.3.1.8.3 Diagram



13.11.3.1.8.4 Fields

Field	Function
31-30 NOC_B_CLK_CTL	Used to bypass the programmable clock controls for bus clock input of the HDMI NOC
29-28 —	Reserved
27-26 LCDIF_SPU_CLK_CTL	Used to bypass the programmable clock controls for spu_clk input of LCDIF
25-24 LCDIF_PIX_CLK_CTL	Used to bypass the programmable clock controls for pix_clk input of LCDIF
23-22 LCDIF_PDI_CLK_CTL	Used to bypass the programmable clock controls for pdi_clk input of LCDIF
21-20	Used to bypass the programmable clock controls for b_clk input of LCDIF

Table continues on the next page...

Field	Function
LCDIF_B_CLK_CTL	
19-18 LCDIF_APB_CLK_CTL	Used to bypass the programmable clock controls for apb_clk input of LCDIF
17-16 VSFD_CEA_CLK_CTL	Used to bypass the programmable clock controls for cea_clk input of vsfd
15-14 VSFD_HTX_P_CLK_CTL	Used to bypass the programmable clock controls for the htx_p_clk input used in VSFD (including portions of HRV_MWR logic)
13-12 VSFD_HTX_APB_CLK_CTL	Used to bypass the programmable clock controls for the apb clk input of VSFD (including portions of HRV_MWR logic)
11-10 HRV_MWR_CEA_CLK_CTL	Used to bypass the programmable clock controls for cea_clk input of HRV_MWR
9-8 HRV_MWR_B_CLK_CTL	Used to bypass the programmable clock controls for b_clk input of HRV_MWR
7-6 VID_SLV_PIX_CLK_CTL	Used to bypass the programmable clock controls for pix_clk input of video link slave
5-4 VID_SLV_APB_CLK_CTL	Used to bypass the programmable clock controls for apb_clk input of video link slave
3-2 TRNG_SKP_CLK_CTL	Used to bypass the programmable clock controls for skp_clk input of TRNG
1-0 TRNG_APB_CLK_CTL	Used to bypass the programmable clock controls for apb_clk input of TRNG

13.11.3.1.9 RTX_CLK_CTL4 (RTX_CLK_CTL4)

13.11.3.1.9.1 Offset

Register	Offset	Description
RTX_CLK_CTL4	80h	RTX_CLK_CTL4
RTX_CLK_CTL4_SET	84h	Writing a 1 to a bit in this register sets the corresponding bit in RTX_CLK_CTL4

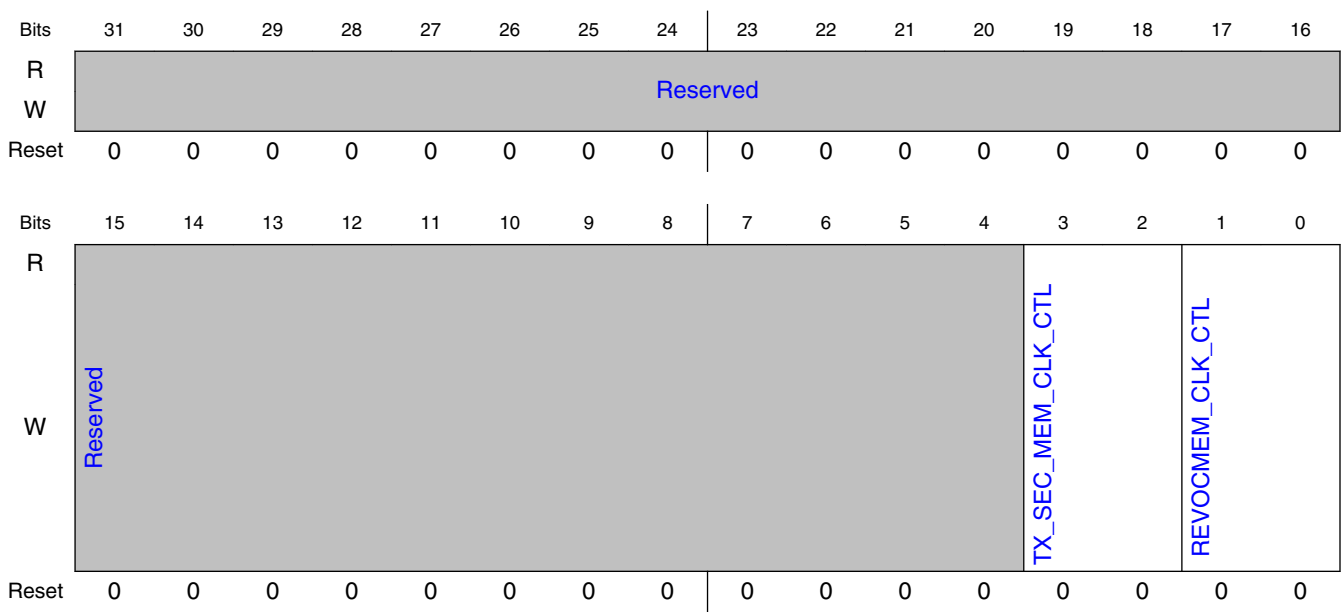
Table continues on the next page...

Register	Offset	Description
RTX_CLK_CTL4_CLR	88h	Writing a 1 to a bit in this register clears the corresponding bit in RTX_CLK_CTL4
RTX_CLK_CTL4_TOG	8Ch	Writing a 1 to a bit in this register toggles the corresponding bit in RTX_CLK_CTL4

13.11.3.1.9.2 Function

This register is used to control the always-on/always-off condition of clocks as fail-safe Fuse takes the highest priority; Second priority is given to always-on/always-off condition Third priority is given to Level2 clock enable which is programmed in BLK_CTL in registers mentioned above Fourth (least) priority is given to Level3 clock enable provided by the corresponding blocks for their idle condition 00 - Default clock gating logic 01 - always clock ON 10 - always clock OFF 11 - Reserved

13.11.3.1.9.3 Diagram



13.11.3.1.9.4 Fields

Field	Function
31-4 —	Reserved
3-2	Used to bypass the request based clock gating on tx_sec_mem

Table continues on the next page...

Field	Function
TX_SEC_MEM_CLK_CTL	
1-0 REVOCMEM_CLK_CTL	Used to bypass the request based clock gating on revocmem

13.11.3.1.10 HDMI_RX_Control (RTX_IRQ_MASK)

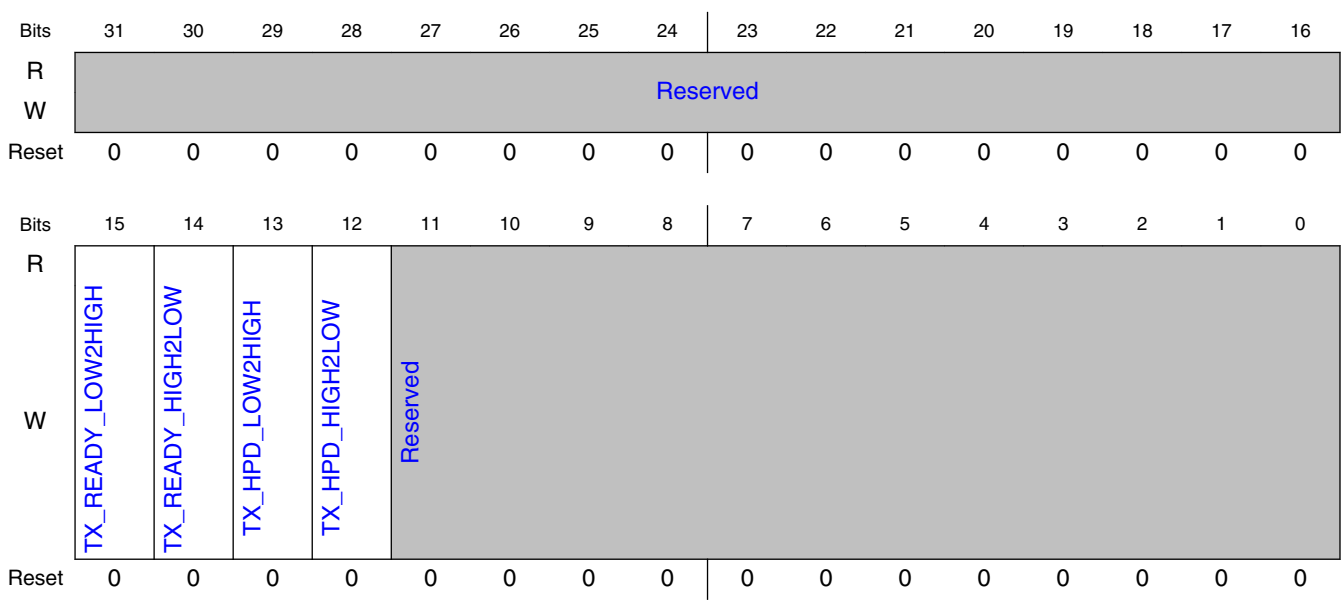
13.11.3.1.10.1 Offset

Register	Offset	Description
RTX_IRQ_MASK	90h	HDMI_RX_Control
RTX_IRQ_MASK_SET	94h	Writing a 1 to a bit in this register sets the corresponding bit in RTX_IRQ_MASK
RTX_IRQ_MASK_CLR	98h	Writing a 1 to a bit in this register clears the corresponding bit in RTX_IRQ_MASK
RTX_IRQ_MASK_TOG	9Ch	Writing a 1 to a bit in this register toggles the corresponding bit in RTX_IRQ_MASK

13.11.3.1.10.2 Function

HDMI_TX Interrupt Mask Control registers

13.11.3.1.10.3 Diagram



13.11.3.1.10.4 Fields

Field	Function
31-16 —	Reserved
15 TX_READY_LOW2HIGH	Interrupt Mask for TX_READY_LOW2HIGH
14 TX_READY_HIGH2LOW	Interrupt Mask for TX_READY_HIGH2LOW
13 TX_HPD_LOW2HIGH	Interrupt Mask for TX_HPD_LOW2HIGH
12 TX_HPD_HIGH2LOW	Interrupt Mask for TX_HPD_HIGH2LOW
11-0 —	Reserved

13.11.3.1.11 HDMI_TX Masked Interrupt status (RTX_IRQ_MASKED_STATUS)

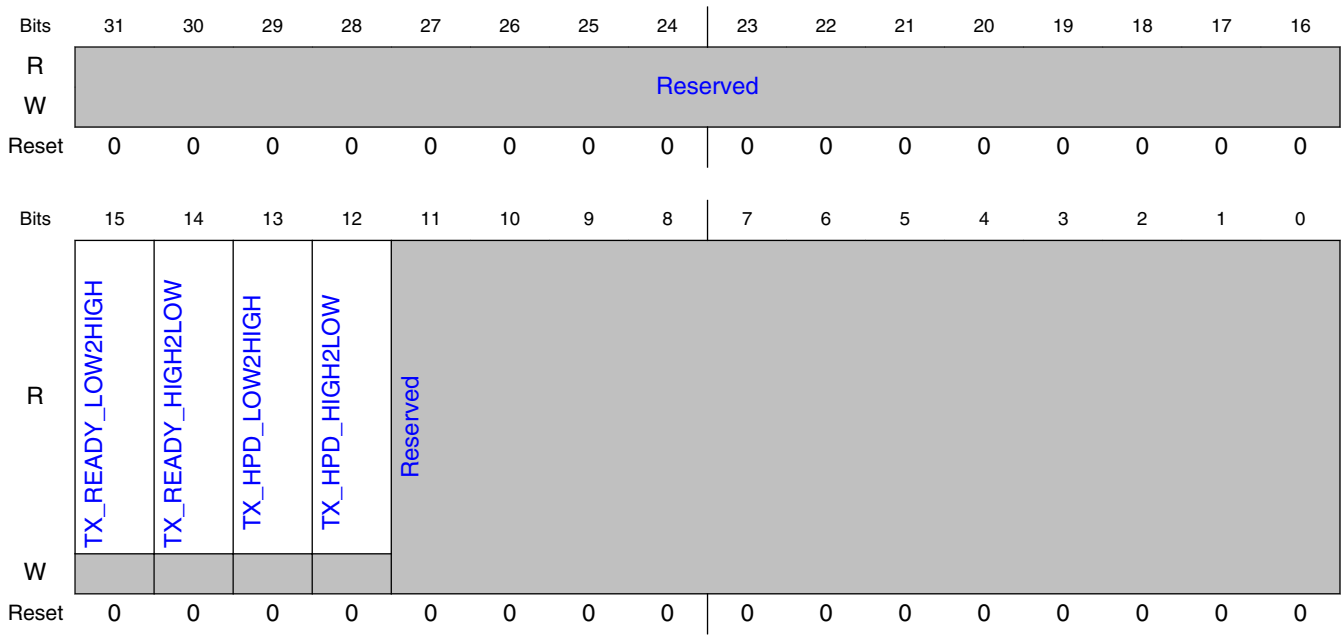
13.11.3.1.11.1 Offset

Register	Offset	Description
RTX_IRQ_MASKED_STATUS	A0h	HDMI_TX Masked Interrupt status
RTX_IRQ_MASKED_STATUS_SET	A4h	Writing a 1 to a bit in this register sets the corresponding bit in RTX_IRQ_MASKED_STATUS
RTX_IRQ_MASKED_STATUS_CLR	A8h	Writing a 1 to a bit in this register clears the corresponding bit in RTX_IRQ_MASKED_STATUS
RTX_IRQ_MASKED_STATUS_TOG	ACh	Writing a 1 to a bit in this register toggles the corresponding bit in RTX_IRQ_MASKED_STATUS

13.11.3.1.11.2 Function

Unmasked interrupt status for the HDMI SS interrupts

13.11.3.1.11.3 Diagram



13.11.3.1.11.4 Fields

Field	Function
31-16 —	Reserved
15 TX_READY_LOW2HIGH	Masked Interrupt status for TX_READY_LOW2HIGH
14 TX_READY_HIGH2LOW	Masked Interrupt status for TX_READY_HIGH2LOW
13 TX_HPD_LOW2HIGH	Masked Interrupt status for TX_HPD_LOW2HIGH
12 TX_HPD_HIGH2LOW	Masked Interrupt status for TX_HPD_HIGH2LOW
11-0 —	Reserved

13.11.3.1.12 HDMI_RX_Control (RTX_IRQ_NONMASK_STATUS)

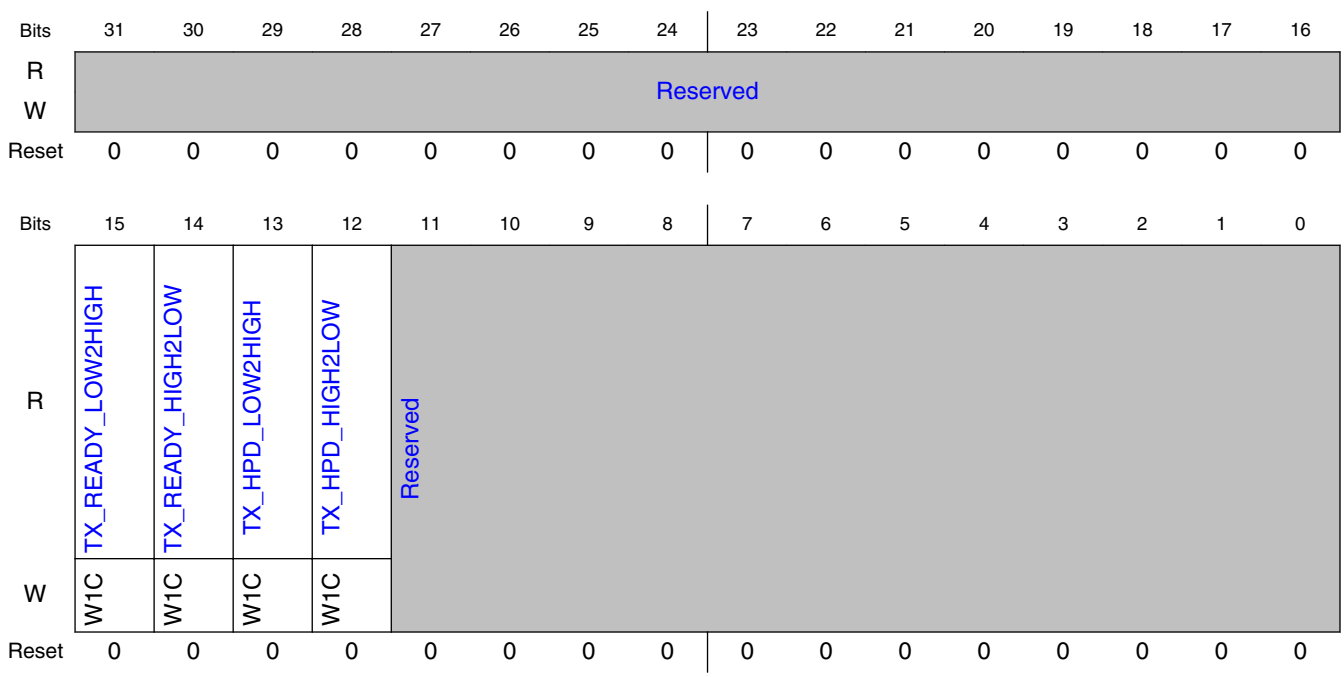
13.11.3.1.12.1 Offset

Register	Offset	Description
RTX_IRQ_NONMASK_STATUS	B0h	HDMI_RX_Control
RTX_IRQ_NONMASK_STATUS_SET	B4h	Writing a 1 to a bit in this register sets the corresponding bit in RTX_IRQ_NONMASK_STATUS
RTX_IRQ_NONMASK_STATUS_CLR	B8h	Writing a 1 to a bit in this register clears the corresponding bit in RTX_IRQ_NONMASK_STATUS
RTX_IRQ_NONMASK_STATUS_TOG	BCh	Writing a 1 to a bit in this register toggles the corresponding bit in RTX_IRQ_NONMASK_STATUS

13.11.3.1.12.2 Function

HDMI_TX unmasked IRQ Status registers

13.11.3.1.12.3 Diagram



13.11.3.1.12.4 Fields

Field	Function
31-16 —	Reserved
15	Unmasked Interrupt status for TX_READY_LOW2HIGH

Table continues on the next page...

Field	Function
TX_READY_LOW2HIGH	
14 TX_READY_HIGH2LOW	Unmasked Interrupt status for TX_READY_HIGH2LOW
13 TX_HPD_LOW2HIGH	Unmasked Interrupt status for TX_HPD_LOW2HIGH
12 TX_HPD_HIGH2LOW	Unmasked Interrupt status for TX_HPD_HIGH2LOW
11-0 —	Reserved

13.11.3.1.13 Miscellaneous Controls for the HDMI TX Controller (TX_CONTROL0)

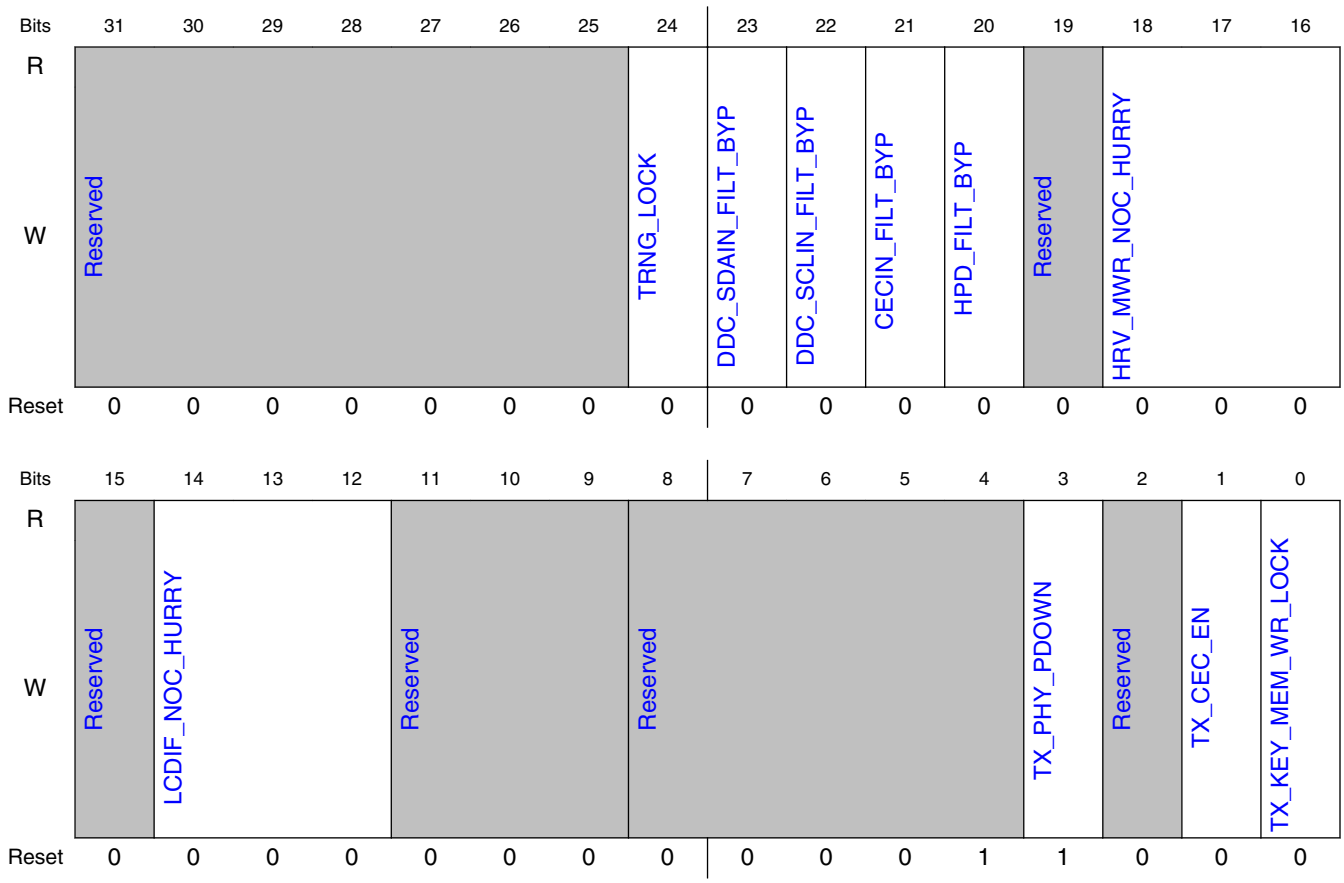
13.11.3.1.13.1 Offset

Register	Offset	Description
TX_CONTROL0	200h	Miscellaneous Controls for the HDMI TX Controller
TX_CONTROL0_SET	204h	Writing a 1 to a bit in this register sets the corresponding bit in TX_CONTROL0
TX_CONTROL0_CLR	208h	Writing a 1 to a bit in this register clears the corresponding bit in TX_CONTROL0
TX_CONTROL0_TOG	20Ch	Writing a 1 to a bit in this register toggles the corresponding bit in TX_CONTROL0

13.11.3.1.13.2 Function

Misc. Controls registers

13.11.3.1.13.3 Diagram



13.11.3.1.13.4 Fields

Field	Function
31-25 —	Reserved
24 TRNG_LOCK	TRNG_LOCK control TRNG_LOCK control This bit is used to disable writes to TRNG's SMODE.MISSION_MODE config bit. Once set TRNG cannot be set to TEST_MODE until a chip_por is asserted This is a 1-time write (in one reset cycle) bit. Once set a change of value on this bit is not considered Chip por needs to be asserted to reset the Lock state
23 DDC_SDAIN_FILT_BYP	DDC_SDAIN_FILT_BYP Glitch filter bypass for DDC_SDA_IN
22 DDC_SCLIN_FILT_BYP	DDC_SCLIN_FILT_BYP Glitch filter bypass for DDC_SCL_IN
21	CECIN_FILT_BYP Glitch filter bypass for CEC_IN

Table continues on the next page...

Field	Function
CECIN_FILT_BYP	
20 HPD_FILT_BYP	HPD_FILT_BYP Glitch filter bypass for HPD input
19 —	Reserved
18-16 HRV_MWR_NOC_HURRY	HRV_MWR_NOC_HURRY Program this bit to enable the panic trigger from HRV_MWR to be sent to NOC to initiate a faster DDR response
15 —	Reserved
14-12 LCDIF_NOC_HURRY	LCDIF_NOC_HURRY Program this bit to enable the panic trigger from LCDIF to be sent to NOC to initiate a faster DDR response
11-9 —	Reserved
8-4 —	Reserved
3 TX_PHY_PDOWN	TX_PHY_PDOWN control Drives PHY_PDOWN input of TX Phy
2 —	Reserved
1 TX_CEC_EN	TX_CEC_EN control CSR_HTX_CEC_EN control as shown in the ADD
0 TX_KEY_MEM_WR_LOCK	TX_KEY_MEM_WR_LOCK control TX_KEY_MEM_WR_LOCK control This is a 1-time write (in one reset cycle) bit. Once set a change of value on this bit is not considered Chip por needs to be asserted to reset the Lock state

13.11.3.1.14 TX Control (TX_CONTROL2)

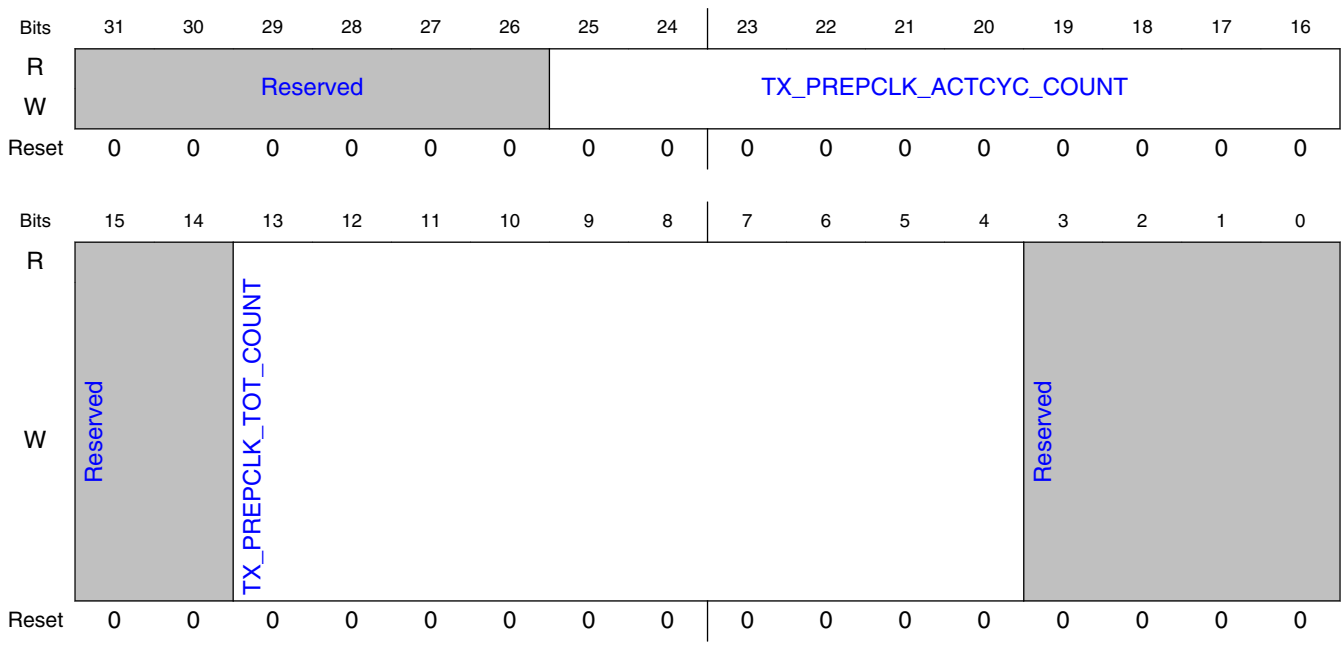
13.11.3.1.14.1 Offset

Register	Offset	Description
TX_CONTROL2	220h	TX Control
TX_CONTROL2_SET	224h	Writing a 1 to a bit in this register sets the corresponding bit in TX_CONTROL2
TX_CONTROL2_CLR	228h	Writing a 1 to a bit in this register clears the corresponding bit in TX_CONTROL2
TX_CONTROL2_TOG	22Ch	Writing a 1 to a bit in this register toggles the corresponding bit in TX_CONTROL2

13.11.3.1.14.2 Function

Misc. TX Control registers

13.11.3.1.14.3 Diagram



13.11.3.1.14.4 Fields

Field	Function
31-26 —	Reserved
25-16 TX_PREPCLK_ACTCYC_COUNT	TX_PREPCLK_ACTCYC_COUNT control prepclk_act_cycles - defines the number of active cycles
15-14 —	Reserved
13-4 TX_PREPCLK_TOT_COUNT	TX_PREPCLK_TOT_COUNT control prepclk_tot_cycles - defines the total number of tmds_clko cycles to iterate over
3-0 —	Reserved

13.11.3.1.15 Status (TX_STATUS0)

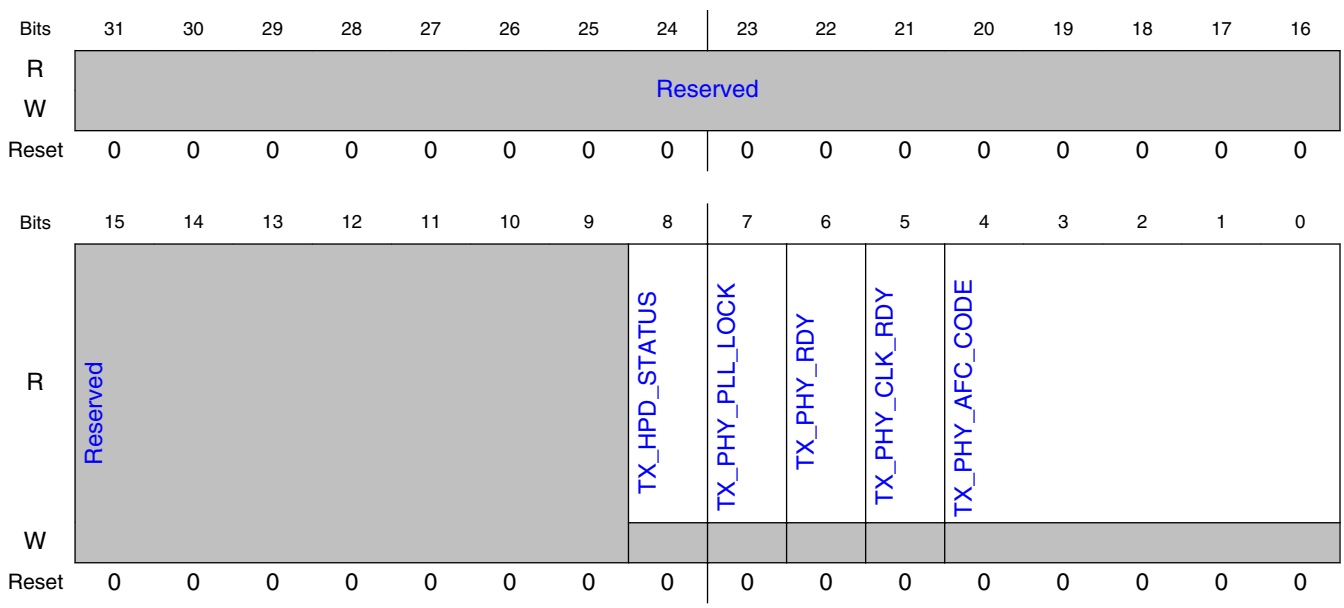
13.11.3.1.15.1 Offset

Register	Offset	Description
TX_STATUS0	230h	Status
TX_STATUS0_SET	234h	Writing a 1 to a bit in this register sets the corresponding bit in TX_STATUS0
TX_STATUS0_CLR	238h	Writing a 1 to a bit in this register clears the corresponding bit in TX_STATUS0
TX_STATUS0_TOG	23Ch	Writing a 1 to a bit in this register toggles the corresponding bit in TX_STATUS0

13.11.3.1.15.2 Function

TX Status

13.11.3.1.15.3 Diagram



13.11.3.1.15.4 Fields

Field	Function
31-9 —	Reserved
8	TX_HPD_STATUS status

Table continues on the next page...

Field	Function
TX_HPD_STATUS	TX_HPD_STATUS gives the current state of the HPD
7 TX_PHY_PLL_LOCK	TX_PHY_PLL_LOCK status TX_PHY_PLL_LOCK status
6 TX_PHY_RDY	TX_PHY_RDY status TX_PHY_RDY status
5 TX_PHY_CLK_RDY	TX_PHY_CLK_RDY status TX_PHY_CLK_RDY status
4-0 TX_PHY_AFC_CODE	TX_PHY_AFC_CODE status TX_PHY_AFC_CODE status

13.12 HDMI TX Parallel Audio Interface (HTX_PAI)

13.12.1 Overview

The HDMI TX Parallel Audio Interface (HTX_PAI) is a digital module that acts as the bridge between the Audio Subsystem to the HDMI TX Controller.

13.12.1.1 Block Diagram

The HTX_PAI block diagram is represented by the following figure:

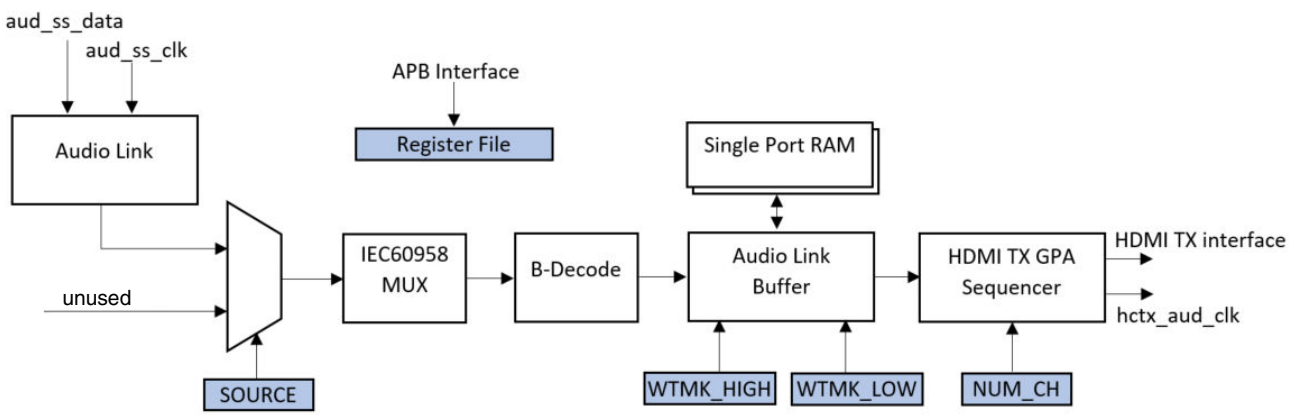


Figure 13-82. HTX_PAI Block Diagram

13.12.1.2 Features

The HTX_PAI has the following features:

- IEC60958 component muxing
- B-Preamble decoding
- Intermediate storage to queue results prior to being passed to the HDMI TX GPA Interface

13.12.2 Functional Description

The source of audio data is selected through the HTX_PAI_CTRL_EXT[SOURCE] register field.

If SOURCE = 0x0, audio samples are received from the Audio Subsystem using a source synchronous interface which is contained within Audio Link module. Note, data received by the Audio Link will be 32-bits wide and must contain the standard IEC60958 P,C,U,V,Data, and Preamble fields. However, data received from the audio subsystem can have an arbitrary component ordering. The HTX_PAI block has integrated muxing options to select which sections of the 32-bit input data word will be mapped to each IEC60958 field. The HTX_PAI_FIELD_CTRL register contains mux selects to individually select P,C,U,V,Data, and Preamble. An illustration of the IEC60958 mux select hardware is represented in the following diagram.

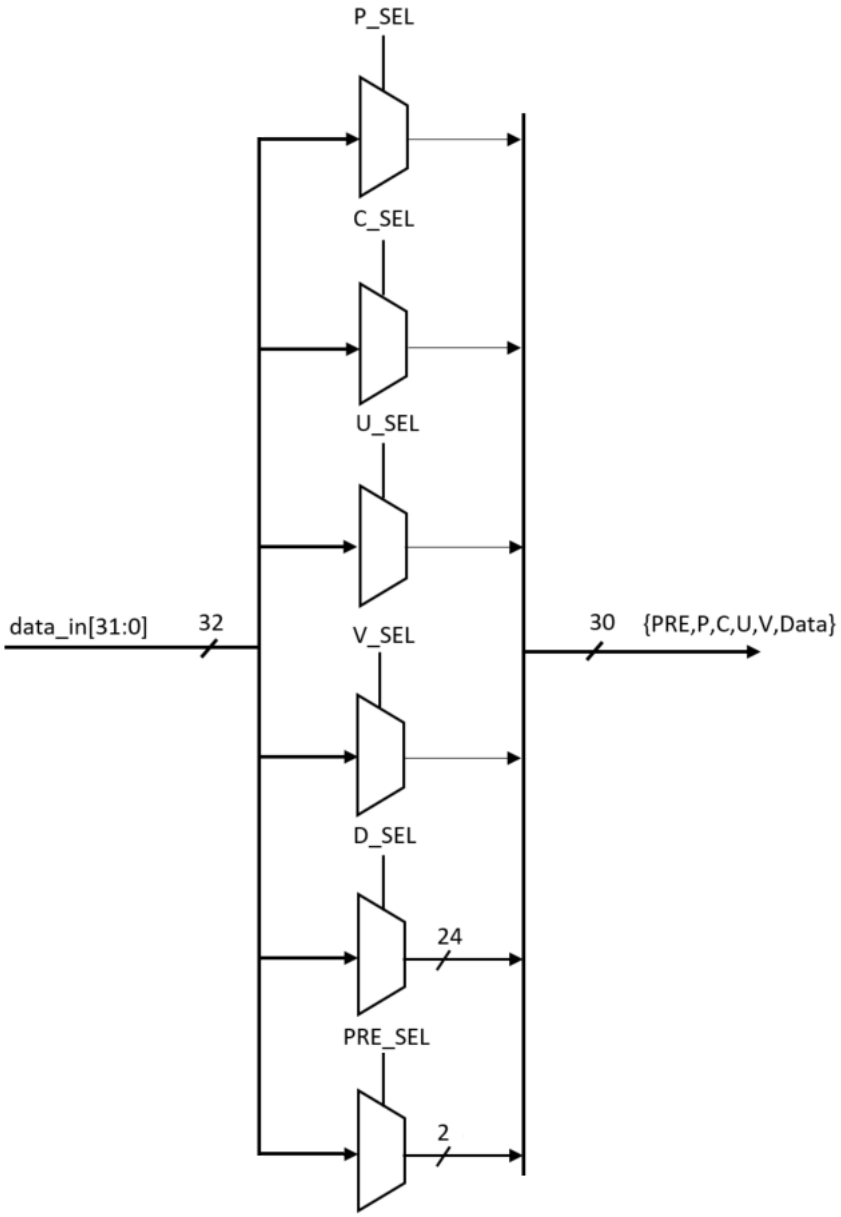


Figure 13-83. HTX_PAI IEC60958 Muxing

NOTE

The muxing procedure decodes the IEC60958 4-bit preamble into a synthetic 2-bit preamble to be used for the HTX_PAI and downstream logic. The final muxed output will be a 30-bit field organized in {PRE[1:0],P,C,U,V,Data[23:0]} order. After component muxing, data is decoded and placed into the Audio Link Buffer. The buffer contains storage for up to 256 samples.

There is a programmable watermark to detect when the Audio Link Buffer fills above or below a specific threshold. This will provide status information which can be used for audio data flow control. Maskable interrupt flags will be generated for underflow, overflow, and watermark based events in the audio link buffer. Back pressure will be provided to the audio subsystem if the Audio Link Buffer WTMK_HIGH is crossed.

NOTE

This watermark should always be programmed with a minimum 8 entry headroom in the buffer (WTMK_HIGH < 248) to allow backpressure to properly function.

WTMK_LOW can be adjusted as per the system requirements to get an IRQ at the desired levels.

NOTE

The HTX_PAI_CTRL_EXT register must be completely configured prior to running the HTX_PAI block. The HTX_PAI module is enabled by setting the HTX_PAI_CTRL[ENABLE] bit to 0x1.

13.12.2.1 HDMI TX GPA Sequencer

The HDMI TX GPA interface uses a simple handshake based protocol to retrieve data from the Audio Link Buffer. The following waveform represents the TX GPA protocol for the case of two channel LPCM audio:

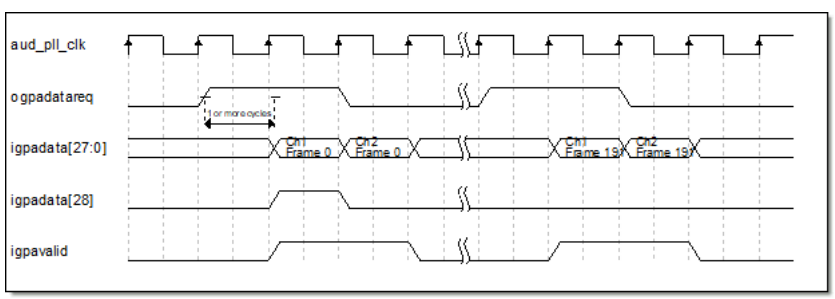


Figure 13-84. 2 Channel LPCM GPA Transaction

The HDMI TX Controller's GPA interface has a 29 bit audio packet, where the B-preamble bit is igpdata[28].Control logic will decode the B,Z – preamble, and set igpdata[28] high when detected.

Note, igpavalid must remain high throughout the duration of each data transfer. This means that 1 sample per channel must be queued in the Audio Link Buffer prior to accepting a new transaction. If the HDMI TX GPA module requests a transfer before the Audio Link Buffer contains enough samples to fulfill the request, then an underflow

interrupt request will be triggered in the HTX_PAI_IRQ*[UND] register. The number of channels enabled is programmable using the HTX_PAI_CTRL_CXT[NUM_CH] register field. The HDMI TX GPA Sequencer will ensure that the Audio Link Buffer contains NUM_CH+1 samples prior to transmitting data to the HDMI TX GPA module.

Overflow happens when the FIFO is full and there is a request to write into the FIFO by the audio source. Corresponding IRQ will be triggered.

13.12.3 Memory Map and Register Definition

The memory map and detailed descriptions of all its registers are as follows.

13.12.3.1 HTX_PAI register descriptions

13.12.3.1.1 HTX_PAI_ADDR memory map

d_ip_htx_pai_syn base address: 32FC_4800h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	HTX PAI Control (HTX_PAI_CTRL)	32	RW	0000_0000
4	HTX PAI Control Extended (HTX_PAI_CTRL_EXT)	32	RW	0000_0000
8	HTX PAI Field Control (HTX_PAI_FIELD_CTRL)	32	RW	0000_0000
C	HTX PAI Status (HTX_PAI_STAT)	32	RO	0000_0000
10	HTX PAI Nonmasked Interrupt Flags (HTX_PAI_IRQ_NOMASK)	32	W1C	0000_0000
14	HTX PAI Masked Interrupt Flags (HTX_PAI_IRQ_MASKED)	32	W1C	0000_0000
18	HTX PAI IRQ Masks (HTX_PAI_IRQ_MASK)	32	RW	0000_0000

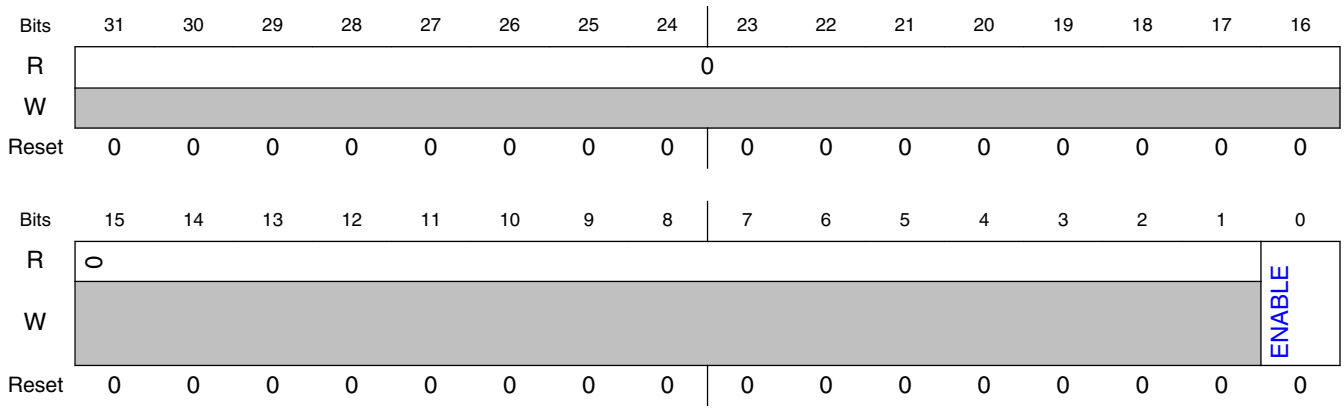
13.12.3.1.2 HTX PAI Control (HTX_PAI_CTRL)

The HTX PAI Control Register enables/disables the HTX PAI

13.12.3.1.2.1 Offset

Register	Offset
HTX_PAI_CTRL	0h

13.12.3.1.2.2 Diagram



13.12.3.1.2.3 Fields

Field	Description
31-1 —	Reserved
0 ENABLE	HTX PAI Enable Setting ENABLE to 1 will allow the HTX PAI to begin running. The ENABLE bit should only be set when all static control registers contain valid settings. Note, de-assertion of this register field will cause a soft-reset the HTX PAI control hardware. 0 - HTX PAI disabled 1 - HTX PAI enabled

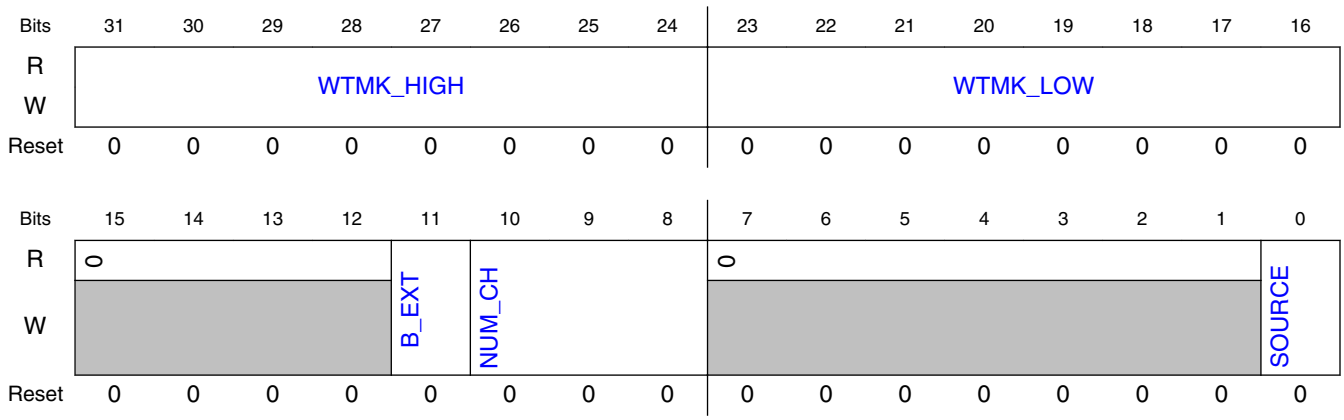
13.12.3.1.3 HTX PAI Control Extended (HTX_PAI_CTRL_EXT)

The HTX PAI Control Extended Register provides additional HTX PAI configuration/ control.

13.12.3.1.3.1 Offset

Register	Offset
HTX_PAI_CTRL_EXT	4h

13.12.3.1.3.2 Diagram



13.12.3.1.3.3 Fields

Field	Description
31-24 WTMK_HIGH	HTX PAI Watermark High The WTMK_HIGH level is used to detect if the number of entries in the link buffer is above a programmable threshold. If the number of entries is above WM_HIGH, a maskable interrupt flag will be generated, and the status bit WM_HIGH will be flaged in the HTX_PAI_STAT register. Note, WTMK_HIGH is also used for upstream flow control when operating in HTX_PAI_CTRL_CXT[SOURCE] = 0x0. In this mode, the maximum supported watermark level is WTMK_HIGH = 0xF7.
23-16 WTMK_LOW	HTX PAI Watermark Low The watermark low level is used to detect if the number of entries in the link buffer is below a programmable threshold. If the number of entries is below WTMK_LOW, a maskable interrupt flag will be generated, and the status bit WM_LOW will be flaged in the HTX_PAI_STAT register.
15-12 —	Reserved
11 B_EXT	B-Field Extension This bit is used to extend the B preamble. It will ensure that the B preamble is set for at least 2 consecutive samples. Note, if the B-preamble input to the HTX_PAI is already set for more than 1 cycle, B_EXT will have no impact. 0 - Use default B-Preamble timing (from selected input source). 1 - Extend B-Preamble timing to ensure it is set for at least 2 cycles.
10-8 NUM_CH	Number of Channels Per Packet This register defines the number of audio channels required for 1 full HDMI TX audio packet delivery. For example, in a stereo audio use case there will be 2 channels per packet, so the NUM_CH regiser field will be set to 0x1. 000 - There is 1 channel per packet. 001 - There are 2 channels per packet. 010 - There are 3 channels per packet. 011 - There are 4 channels per packet.

Table continues on the next page...

Field	Description
	100 - There are 5 channels per packet. 101 - There are 6 channels per packet. 110 - There are 7 channels per packet. 111 - There are 8 channels per packet.
7-1 —	Reserved
0 SOURCE	HTX PAI Source Select This bit selects the source for audio data input to the HTX_PAI. 0 - Normal operation. Data input from the audio subsystem. 1 - Reserved.

13.12.3.1.4 HTX PAI Field Control (HTX_PAI_FIELD_CTRL)

Audio frames must be stored in the Audio Link Buffer in the following format:

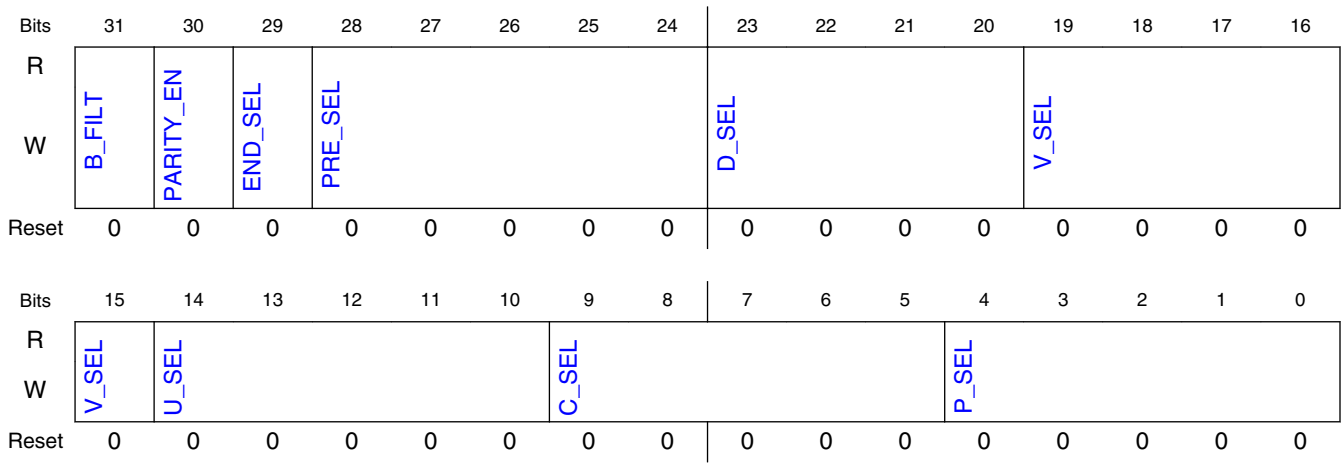
{PRE[1:0],P,C,U,V,Data[23:0]}

Writes to the HTX_PAI block are expected to contain all of the IEC60958 bit fields. However, the ordering of each field can be arbitrary in the input word. Bit field muxing is implemented in the HTX_PAI_FIELD_CTRL to provide a control mechanism to map any input word muxing to the output format required for the HDMI subsystem. MUX selects are implemented to select where each IEC60958 field is concatenated in the input word.

13.12.3.1.4.1 Offset

Register	Offset
HTX_PAI_FIELD_CTRL	8h

13.12.3.1.4.2 Diagram



13.12.3.1.4.3 Fields

Field	Description
31 B_FILT	B-Detect Filter This register field will enable the HTX_PAI to filter audio samples until the IEC60958 B preamble is detected. All samples written to the HTX_PAI will be dropped until the B preamble is detected if B_FILT = 1.
30 PARITY_EN	Parity Enable This bit is used to enable the HTX_PAI parity generator. If the input word already contains a pre-computed IEC60958 parity, then this bit can be set to 0. To ignore the input parity or generate the IEC60958 parity, set PARITY_EN to 1.
29 END_SEL	Endianness Select This bit is used to control the endianness of data input to the HTX_PAI. Set this bit to 0 if the input data is in little-endian format. Set this bit to 1 if the input data is in big-endian format.
28-24 PRE_SEL	IEC60958 Preamble Select This field is used to mux the Preamble field from the IEC60958 input data word. The P_SEL must be set equal to the bit position where the IEC60958 Preamble field is located (MSB-1). Note, the parity used by the HTX_PAI is compressed to two bits.
23-20 D_SEL	IEC60958 Data Select This field is used to select the position of the 24-bit data field input to the HTX_PAI. Note, the programing of this field will not be impacted by the endianness of the data. The D_SEL field should be programmed as the first bit position where valid data is located beginning from bit 0 and indexing up. The D_SEL determines the data position in the following way: data[23:0] = ((input_audio) >> D_SEL);
19-15 V_SEL	IEC60958 V Select This field is used to mux the Validity (V) field from the IEC60958 input data word. The V_SEL must be set equal to the bit position where the IEC60958 V field is located.
14-10 U_SEL	IEC60958 U Select This field is used to mux the User Data (U) field from the IEC60958 input data word. The U_SEL must be set equal to the bit position where the IEC60958 U field is located.

Table continues on the next page...

Field	Description
9-5 C_SEL	IEC60958 C Select This field is used to mux the Channel Status (C) field from the IEC60958 input data word. The C_SEL must be set equal to the bit position where the IEC60958 C field is located.
4-0 P_SEL	IEC60958 P Select This field is used to mux the Parity (P) field from the IEC60958 input data word. The P_SEL must be set equal to the bit position where the IEC60958 P field is located.

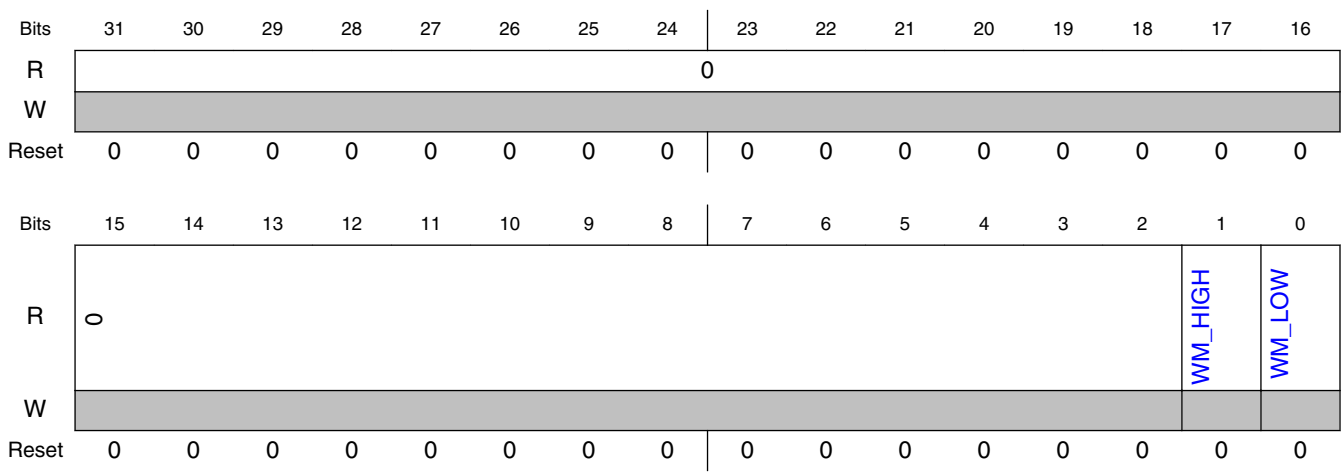
13.12.3.1.5 HTX PAI Status (HTX_PAI_STAT)

The HTX PAI Status Register reports the watermark status.

13.12.3.1.5.1 Offset

Register	Offset
HTX_PAI_STAT	Ch

13.12.3.1.5.2 Diagram



13.12.3.1.5.3 Fields

Field	Description
31-2 —	Reserved
1 WM_HIGH	Watermark High Flag

Table continues on the next page...

Field	Description
	This flag will be set when the number of entries in the audio link buffer exceeds the WTMK_HIGH threshold.
0 WM_LOW	Watermark Low Flag This flag will be set when the number of entries in the audio link buffer is below the WTMK_LOW threshold.

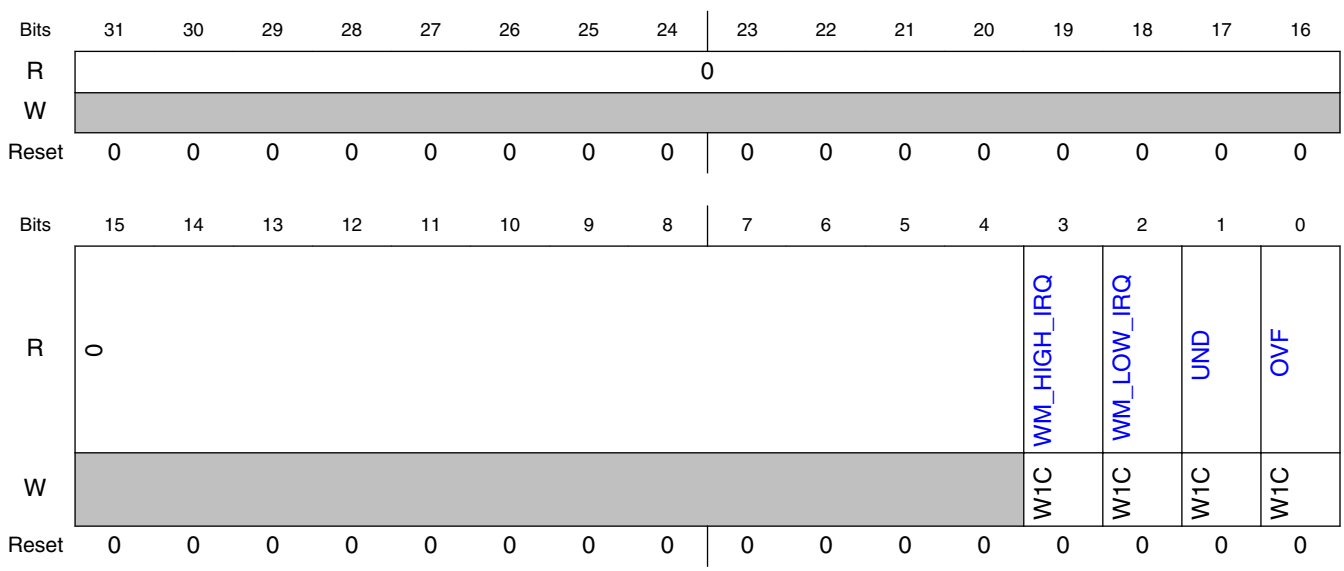
13.12.3.1.6 HTX PAI Nonmasked Interrupt Flags (HTX_PAI_IRQ_NOMASK)

This register shows the nonmasked interrupt flags. Write 1 to clear.

13.12.3.1.6.1 Offset

Register	Offset
HTX_PAI_IRQ_NOMASK	10h

13.12.3.1.6.2 Diagram



13.12.3.1.6.3 Fields

Field	Description
31-4 —	Reserved
3	Watermark High IRQ

Table continues on the next page...

Field	Description
WM_HIGH_IRQ	This IRQ will be flagged when the number of entries in the link buffer transitions above the threshold programmed in HTX_PAI_CTRL[WTMK_HIGH].
2 WM_LOW_IRQ	Watermark Low IRQ This IRQ will be flagged when the number of entries in the link buffer transitions from above the threshold programmed in HTX_PAI_CTRL[WTMK_LOW] to below the threshold.
1 UND	HTX PAI Buffer Underflow This IRQ will be flagged when a buffer underflow occurs in the HTX PAI. A buffer underflow indicates that the HDMI TX controller is requesting data faster than the HTX PAI is receiving it.
0 OVF	HTX PAI Buffer Overflow This IRQ will be flagged when a buffer oveflow occurs in the HTX PAI. A buffer overflow indicates that the HDMI TX controller is requesting data slower than the HTX PAI is receiving it.

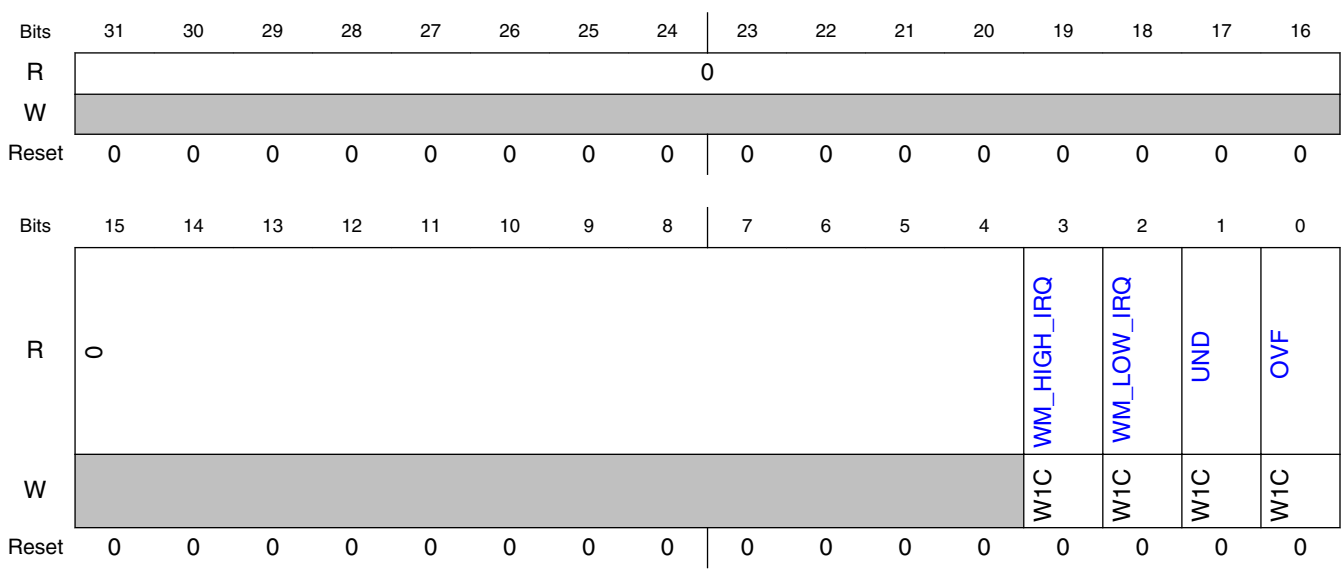
13.12.3.1.7 HTX PAI Masked Interrupt Flags (HTX_PAI_IRQ_MASKED)

This register shows the maskable interrupt flags. Write 1 to clear.

13.12.3.1.7.1 Offset

Register	Offset
HTX_PAI_IRQ_MASKED	14h

13.12.3.1.7.2 Diagram



13.12.3.1.7.3 Fields

Field	Description
31-4 —	Reserved
3 WM_HIGH_IRQ	Watermark High IRQ Masked Masked version of WM_HIGH_IRQ.
2 WM_LOW_IRQ	Watermark Low IRQ Masked Masked version of WM_LOW_IRQ.
1 UND	HTX PAI Buffer Underflow Masked version of UND IRQ.
0 OVF	HTX PAI Buffer Overflow Masked version of OVF IRQ.

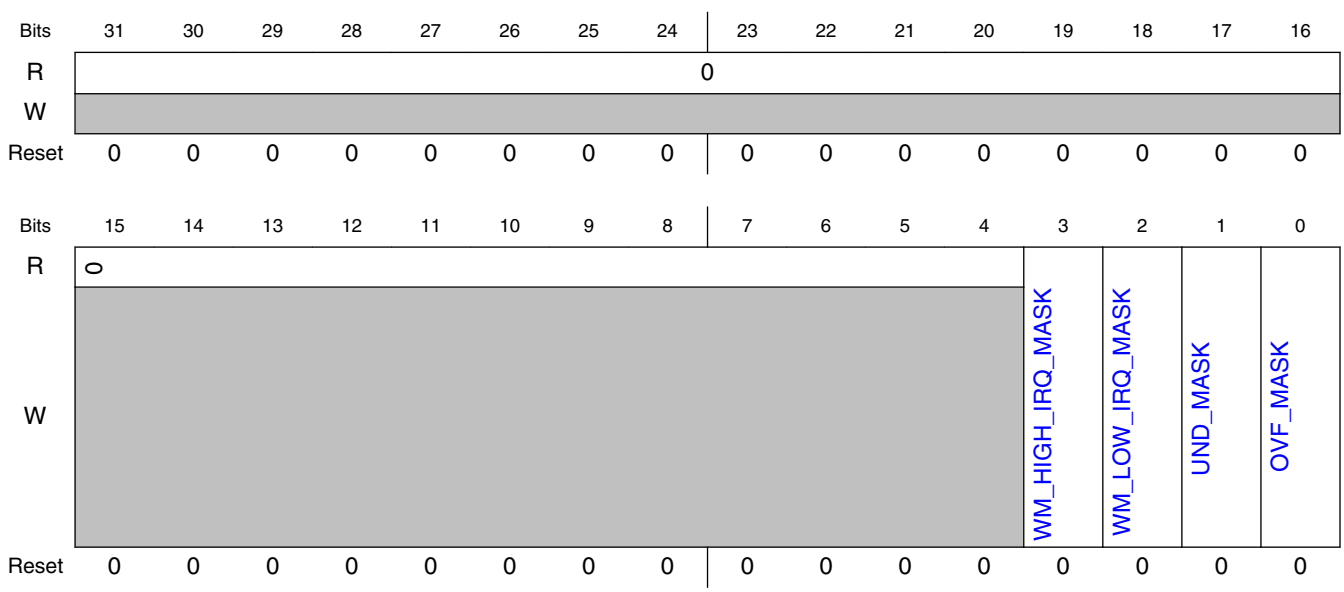
13.12.3.1.8 HTX PAI IRQ Masks (HTX_PAI_IRQ_MASK)

This register controls the IRQ Masks.

13.12.3.1.8.1 Offset

Register	Offset
HTX_PAI_IRQ_MASK	18h

13.12.3.1.8.2 Diagram



13.12.3.1.8.3 Fields

Field	Description
31-4 —	Reserved
3 WM_HIGH_IRQ_MASK	Watermark High IRQ Mask Set this bit to 1 to mask the WM_HIGH_IRQ.
2 WM_LOW_IRQ_MASK	Watermark Low IRQ Mask Set this bit to 1 to mask the WM_LOW_IRQ.
1 UND_MASK	HTX PAI Buffer Underflow Mask Set this bit to 1 to mask the UND IRQ.
0 OVF_MASK	HTX PAI Buffer Overflow Mask Set this bit to 1 to mask the OVF IRQ.

13.13 HDMI TX Parallel Video Interface (HTX_PVI)

13.13.1 Overview

The HDMI TX Parallel Video Interface (HTX_PVI) is a digital module that acts as the bridge between the Video Subsystem to the HDMI TX Controller.

13.13.1.1 Block Diagram

The HTX_PVI block diagram is represented by the following figure:

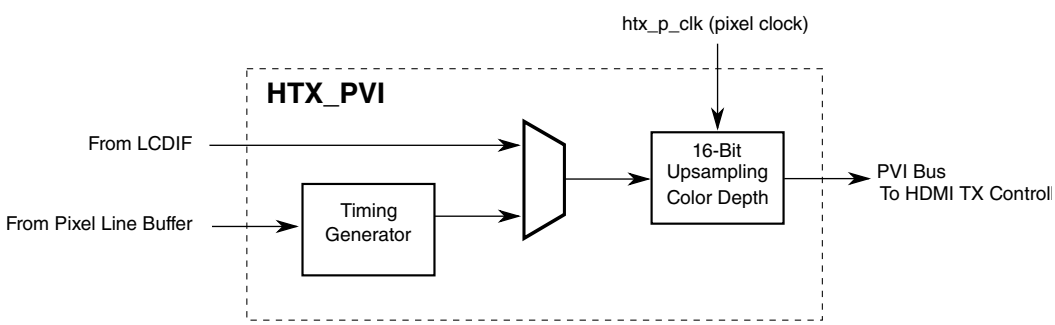


Figure 13-85. HTX_PVI Block Diagram

13.13.1.2 Features

The features for the HTX_PVI are as follows:

- Timing generator:
  - Implements full CEA-861-G compliant Hsync, Vsync, and DE timing strobes
  - Generates 4 independently programmable IRQs
- Supports input pixel formats RGB444, YCbCr444 and YCbCr422
- HTX_PVI interface supports 48-bits (16-bit color depth) of pixel data per htx_p_clk clock cycle
- The 16-bit color depth upsampling logic has software control to select between the following upsampling options:
  - csr_htc_upsample = 0x0 = lower 4 bits tied to 4'h0 (default)
  - csr_htc_upsample = 0x1 = lower 4 bits are replicated with the respective color component's bits[11:8]

13.13.2 Functional Description

13.13.2.1 Modes of Operation

The HTX_PVI operates in two different modes:

- LCDIF direct path to HDMI TX Controller
- Pixel Line Buffer path to HDMI TX Controller

13.13.2.2 Timing Generator

The timing generator provides fully compliant CEA-861-G timing strobes to the HDMI TX Controller. This feature is necessary to protect the HDMI TX incoming video stream from a HDMI RX frequency change such as video mode change or loss of clock due to sink cable disconnect. In order to protect the HDMI TX controller from HDMI RX change of frequency (or loss of clock), and pixel underflow conditions, the timing generator will implement the following functions:

- Fully functional timing generator and pixel background color insertion control - basic function of the Color Sub-sampler, when operating in pass-thru mode will be retain where the incoming pixel stream kicks off the timing generator and insert the incoming pixels with the regenerated timing strobes
- Timing generator implements full CEA-861-G compliant Hsync, Vsync, and DE timing strobes:

- Counters start counting based off of the incoming Vsync assertion
- Hsync counter with control register programming for Hsync rising and falling edge count values
- Vsync counter with control register programming for Vsync rising and falling edge
- For timing generated strobes, independent control registers used to select Hsync, Vsync and DE assertion active high or low
- For incoming video stream timing strobes, independent control registers used to select Hsync, Vsync and DE assertion active high or low

- The timing generator pixel insertion:
  - In the pixel data path just before the timing generator, pixels are popped off an async fifo. The pixel source is a streaming source, where valid data is always available. The async fifo data valid signal is used to:
    - If async fifo data is not empty – Pixel data is transmitted with the timing generator's generated timing strobes
    - If async fifo is empty – This represents an underflow condition. The timing generator performs two functions:
      - Assert an underflow error IRQ
      - Insert a background pixel
  - The timing generator will generate 4 independently programmable IRQs. Each IRQ will be programmable to trigger on any Frame row and pixel location within that row
  - In the case where the Pixel Line Buffer is the path to HDMI TX Controller, the timing generator will generate the HRV_MWR kick. The kick will be software configurable to any Frame row and pixel position within that row
    - The kick should never be generated if the HDMI TX datapath is not selected as the input source to the HRTX Pixel Line Buffer

13.13.2.3 16-bit Color Depth Upsampling

The HTX_PVI interface supports 48-bits (16-bit color depth) of pixel data per htx_p_clk clock cycle. The 16-bit color depth upsampling logic will have software control to select between the following upsampling options: csr_htc_upsample – 0x0 = lower 4 bits tied to 4'h0 (default). 0x1 = lower 4 bits are replicated with the respective color component's bits[11:8]. Based on the software configuration, the pixel data will be upsampled, and pack the pixel data for the HTX_PVI interface as shown in the figures below.

Video Format				Pixel Data [47:32]																Pixel Data [31:16]														Pixel Data [15:0]													
Color Space	Color Depth	Sample M. Code		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RGB444	8 - bit	1		R [7:0]																G [7:0]																B [7:0]													
	10 - bit	3		R[9:0]																G [9:0]																B [9:0]													
	12 -bit	5		R[11:0]																G [11:0]																B [11:0]													
	16 - bit	7		R{[11:0],4'h0}																G {[11:0],4'h0}																B {[11:0],4'h0}													
YCbCr 444	8 - bit	9		Cb [7:0]																Y [7:0]																Cr [7:0]													
	10 - bit	11		Cb [9:0]																Y [9:0]																Cr [9:0]													
	12 -bit	13		Cb [11:0]																Y [11:0]																Cr [11:0]													
	16 - bit	15		Cb {[11:0],4'h0}																Y {[11:0],4'h0}																Cr {[11:0],4'h0}													
YCbCr 422	8 - bit	22	even pixels	Cb (n)[7:0]																Y (n)[7:0]																													
			odd pixels	Cr (n)[7:0]																Y (n+1)[7:0]																													
	10 - bit	20	even pixels	Cb (n)[9:0]																Y (n)[9:0]																													
			odd pixels	Cr (n)[9:0]																Y (n+1)[9:0]																													
	12 - bit	18	even pixels	Cb (n)[11:0]																Y (n)[11:0]																													
			odd pixels	Cr (n)[11:0]																Y (n+1)[11:0]																													
YCbCr 420	8 - bit	9	even lines	Cb (L)(n)[7:0]																Y (L)(n)[7:0]																Y (L)(n+1)[7:0]													
			odd lines	Cr (L+1)(n)[7:0]																Y (L+1)(n)[7:0]																Y (L+1)(n+1)[7:0]													
	10 - bit	11	even lines	Cb (L)(n)[9:0]																Y (L)(n)[9:0]																Y (L)(n+1)[9:0]													
			odd lines	Cr (L+1)(n)[9:0]																Y (L+1)(n)[9:0]																Y (L+1)(n+1)[9:0]													
	12 -bit	13	even lines	Cb (L)(n)[11:0]																Y (L)(n)[11:0]																Y (L)(n+1)[11:0]													
			odd lines	Cr (L+1)(n)[11:0]																Y (L+1)(n)[11:0]																Y (L+1)(n+1)[11:0]													
	16 - bit	15	even lines	Cb (L)(n){[11:0],4'h0}																Y (L)(n){[11:0],4'h0}																Y (L)(n+1){[11:0],4'h0}													
			odd lines	Cr (L+1)(n){[11:0],4'h0}																Y (L+1)(n){[11:0],4'h0}																Y (L+1)(n+1){[11:0],4'h0}													
YCbCr 444 (IPI)	8 - bit	23		Y [7:0]																Cb [7:0]																Cr [7:0]													
	10 - bit	24		Y [9:0]																Cb [9:0]																Cr [9:0]													
	12 -bit	25		Y [11:0]																Cb [11:0]																Cr [11:0]													
	16 - bit	26		Y {[11:0],4'h0}																Cb {[11:0],4'h0}																Cr {[11:0],4'h0}													
YCbCr422 (IPI)	12 - bit	27	even pixels	Y (n)[11:0]																Cb (n)[11:0]																Cr (n)[11:0]													
			odd pixels	Y (n+1)[11:0]																Cb (n)[11:0]																Cr (n)[11:0]													
YCbCr 420 (IPI)	8 - bit	28	even lines	Y (L)(n)[7:0]																Y (L)(n+1)[7:0]																Cb (L)(n)[7:0]													
			odd lines	Y (L+1)(n)[7:0]																Y (L+1)(n+1)[7:0]																Cr (L+1)(n)[7:0]													
	10 - bit	29	even lines	Y (L)(n)[9:0]																Y (L)(n+1)[9:0]																Cb (L)(n)[9:0]													
			odd lines	Y (L+1)(n)[9:0]																Y (L+1)(n+1)[9:0]																Cr (L+1)(n)[9:0]													
	12 -bit	30	even lines	Y (L)(n)[11:0]																Y (L)(n+1)[11:0]																Cb (L)(n)[11:0]													
			odd lines	Y (L+1)(n)[11:0]																Y (L+1)(n+1)[11:0]																Cr (L+1)(n)[11:0]													
	16 - bit	31	even lines	Y (L)(n){[11:0],4'h0}																Y (L)(n+1){[11:0],4'h0}																Cb (L)(n){[11:0],4'h0}													
			odd lines	Y (L+1)(n){[11:0],4'h0}																Y (L+1)(n+1){[11:0],4'h0}																Cr (L+1)(n){[11:0],4'h0}													

Figure 13-86. HTX_PVI interface pixel formats with csr_htc_upsample = 0x0

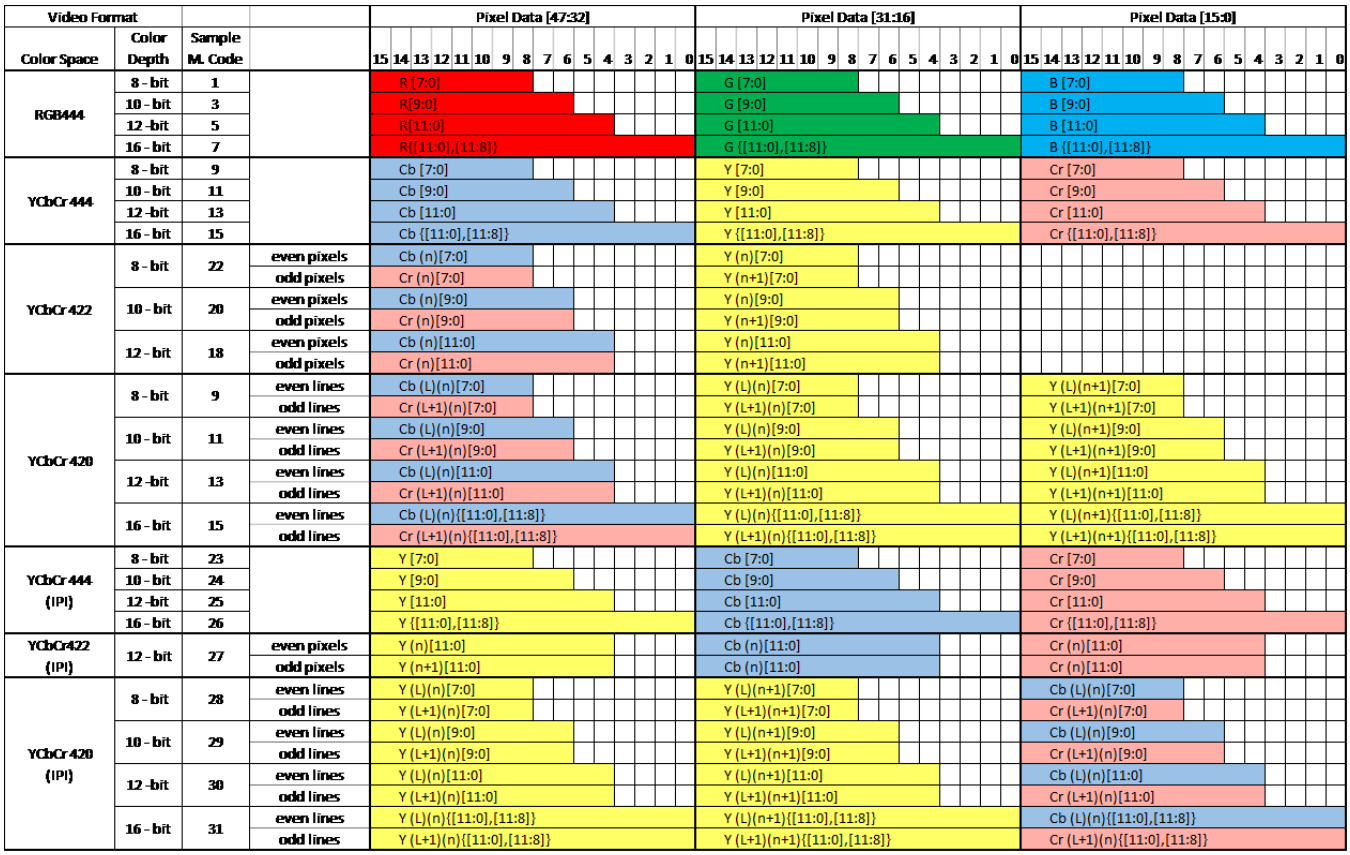


Figure 13-87. HTX_PVI interface pixel formats with csr_htc_upsample = 0x1

13.13.3 Memory Map and Registers

13.13.3.1 register descriptions

13.13.3.1.1 HTX_PVI memory map

d_ip_htx_pvi base address: 32FC_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	HTX_PVI Control Reg (HTX_PVI_CTRL)	32	RW	0037_7000
4	Masks off the Interrupts (HTX_PVI_IRQ_MASK)	32	RW	0000_003F
8	Interrupt Status (HTX_PVI_IRQ_STATUS)	32	RO	0000_0000
C	Interrupts (HTX_PVI_IRQ_CLR)	32	RW	0000_0000
10	Display Coordinates (HTX_TMG_GEN_DISP_LRC)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
14	Data Enable Coordinates (HTX_TMG_GEN_DE_ULC)	32	RW	0000_0000
18	Data Enable Coordinates (HTX_TMG_GEN_DE_LRC)	32	RW	0000_0000
1C	Hsync Start and End (HTX_TMG_GEN_HSYNC)	32	RW	0000_0000
20	Vsync Start and End (HTX_TMG_GEN_VSYNC)	32	RW	0000_0000
24	Controls the Position of first IRQ from Timing Generator (HTX_TMG_GEN_IRQ0)	32	RW	0000_0000
28	Controls the Position of Second IRQ from Timing Generator (HTX_TMG_GEN_IRQ1)	32	RW	0000_0000
2C	Controls the Position of Third IRQ from Timing Generator (HTX_TMG_GEN_IRQ2)	32	RW	0000_0000
30	Controls the Position of Fourth IRQ from Timing Generator (HTX_TMG_GEN_IRQ3)	32	RW	0000_0000
34	Background Color insertion for R or Y (HTX_TMG_GEN_BG0)	32	RW	0000_0000
38	Background Color insertion for G or Cb (HTX_TMG_GEN_BG1)	32	RW	0000_0000
3C	Background Color insertion for B or Cr (HTX_TMG_GEN_BG2)	32	RW	0000_0000
40	HStart and Vstart Delay Configuration (HTX_TMG_GEN_CFG)	32	RW	0000_0000

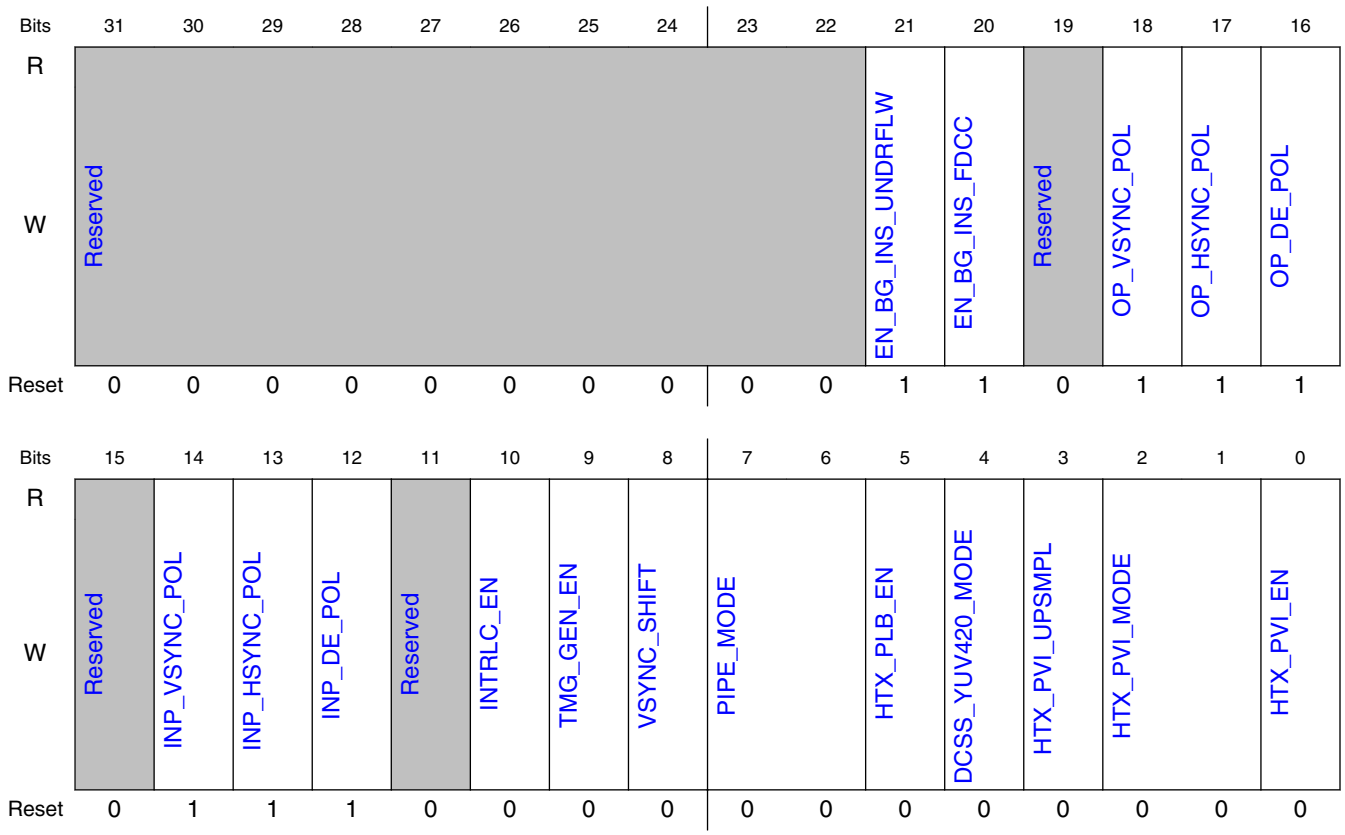
13.13.3.1.2 HTX_PVI Control Reg (HTX_PVI_CTRL)

Contains the Configuration for the HTX PVI

13.13.3.1.2.1 Offset

Register	Offset
HTX_PVI_CTRL	0h

13.13.3.1.2.2 Diagram



13.13.3.1.2.3 Fields

Field	Description
31-22 —	Reserved
21 EN_BG_INS_UNDRFLW	Enables the insertion of background color if the async FIFO underflows.
20 EN_BG_INS_FDCC	Enables the insertion of background color if FDCC detects the HDMI Rx reference clock is out of specified video mode frequency range.
19 —	Reserved
18 OP_VSYNC_POL	Controls the polarity of Vsync of the outgoing data. 0 : Active Low, 1: Active High
17 OP_HSYNC_POL	Controls the polarity of Hsync of the outgoing data. 0 : Active Low, 1: Active High

Table continues on the next page...

Field	Description
16 OP_DE_POL	Controls the polarity of DataEnable of the outgoing data. 0 : Active Low, 1: Active High
15 —	Reserved
14 INP_VSYNC_POL	Contains the information about the polarity of Vsync of the incoming data. 0 : Active Low, 1: Active High
13 INP_HSYNC_POL	Contains the information about the polarity of Hsync of the incoming data. 0 : Active Low, 1: Active High
12 INP_DE_POL	Contains the information about the polarity of DataEnable of the incoming data. 0 : Active Low, 1: Active High
11 —	Reserved
10 INTRLC_EN	Set this bit to 1 to enable interlaced HDMI timing.
9 TMG_GEN_EN	Set this bit to 1 to enable Timing Generator to insert the hsync and vsync.
8 VSYNC_SHIFT	Set 0 to run in general interlaced mode. Every other field the vsync is shifted forward 1/2 scan lines. In addition, the horizontal back porch is extended 1 scan line. Set 1 run in special interlaced format. This shifts vsync backwards 1/2 scan line every other field. Remaining timing won't change.
7-6 PIPE_MODE	Sets the Timing Generator mode. 00: bypass 01: 422 subsample 10: 420 subsample 11: bypass
5 HTX_PLB_EN	Enables Data write to PLB from HDMI Tx
4 DCSS_YUV420_MODE	Indicates the data coming in from DCSS is in yuv 420 mode
3 HTX_PVI_UPSMPL	Select the mode of upsample in case of 16bit output and 12bit input 0 -- Fill LSB with 4'b0, 1 -- Fill LSB with MSB 4 bits, i.e. [11:8]
2-1 HTX_PVI_MODE	Selects the mode of operation in HTX PVI 0 -- Select the DCSS Path, 1 -- Select the Bypass path from HDMI Rx 2 -- Select the LCDIF Path 3 -- Reserved
0 HTX_PVI_EN	Enable to block to receive incoming data

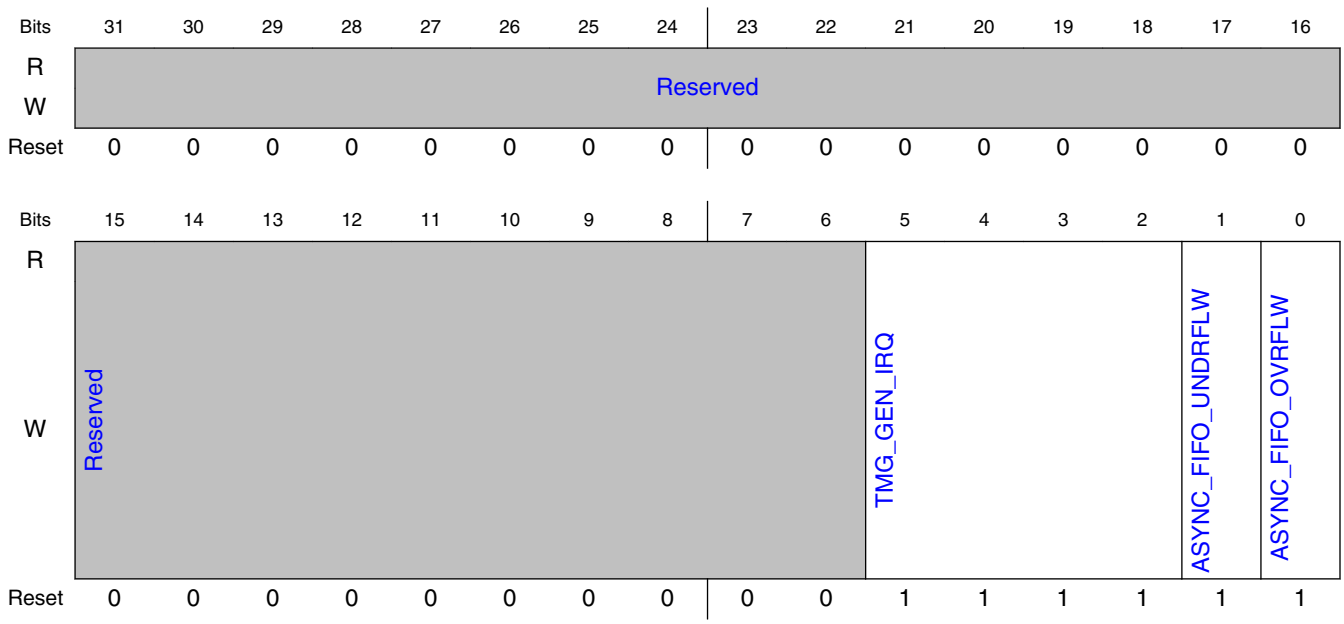
13.13.3.1.3 Masks off the Interrupts (HTX_PVI_IRQ_MASK)

Contains the Error and other interrupts from HTX PVI

13.13.3.1.3.1 Offset

Register	Offset
HTX_PVI_IRQ_MASK	4h

13.13.3.1.3.2 Diagram



13.13.3.1.3.3 Fields

Field	Description
31-6 —	Reserved
5-2 TMG_GEN_IRQ	4 IRQs that can be generated from Timing Generator
1 ASYNC_FIFO_UNDRFLW	Indicates the Async FIFO has underflow
0 ASYNC_FIFO_OVRFLW	Indicates the Async FIFO has overflow

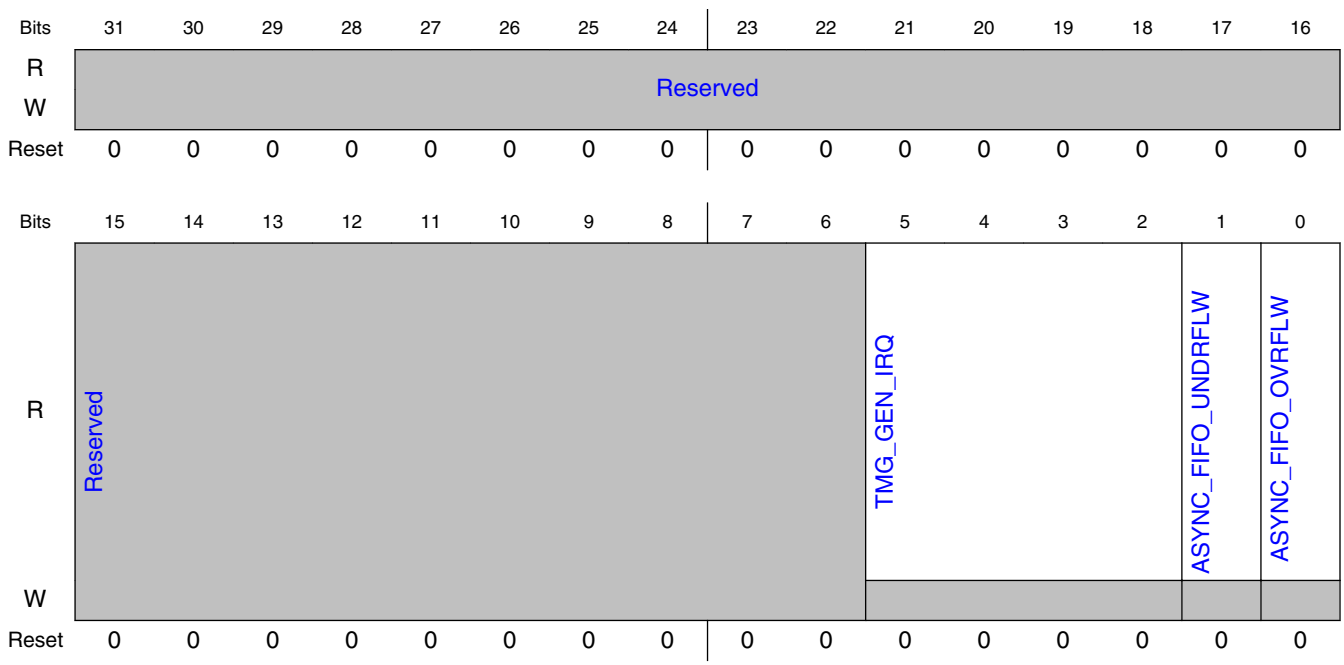
13.13.3.1.4 Interrupt Status (HTX_PVI_IRQ_STATUS)

Contains the Error and other interrupts from HTX PVI

13.13.3.1.4.1 Offset

Register	Offset
HTX_PVI_IRQ_STATUS	8h

13.13.3.1.4.2 Diagram



13.13.3.1.4.3 Fields

Field	Description
31-6 —	Reserved
5-2 TMG_GEN_IRQ	4 IRQs that can be generated from Timing Generator
1 ASYNC_FIFO_UNDRFLW	Indicates the Async FIFO has underflow

Table continues on the next page...

Field	Description
0 ASYNC_FIFO_OVRFLW	Indicates the Async FIFO has overflow

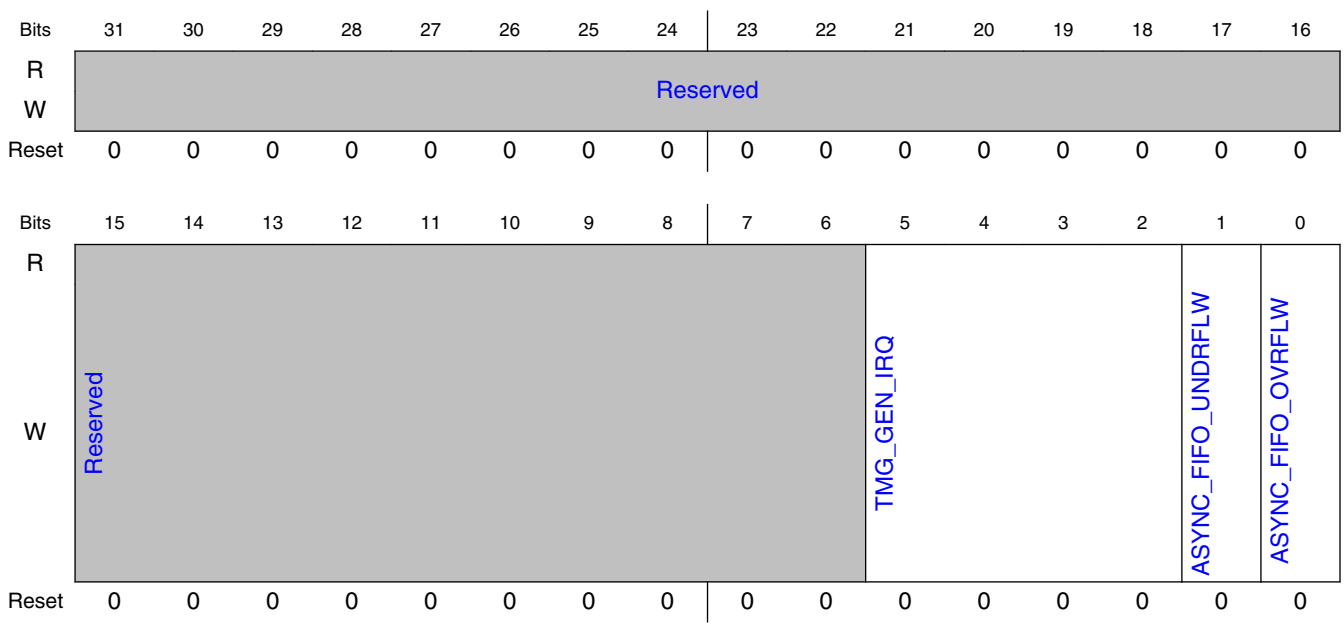
13.13.3.1.5 Interrupts (HTX_PVI_IRQ_CLR)

Contains the Error and other interrupts from HTX PVI

13.13.3.1.5.1 Offset

Register	Offset
HTX_PVI_IRQ_CLR	Ch

13.13.3.1.5.2 Diagram



13.13.3.1.5.3 Fields

Field	Description
31-6	Reserved

Table continues on the next page...

Field	Description
—	
5-2 TMG_GEN_IRQ	4 IRQs that can be generated from Timing Generator
1 ASYNC_FIFO_UNDRFLW	Indicates the Async FIFO has underflow
0 ASYNC_FIFO_OVRFLW	Indicates the Async FIFO has overflow

13.13.3.1.6 Display Coordinates (HTX_TMG_GEN_DISP_LRC)

Contains the Lower Right Corner coordinates of the display needed by the timing generator

13.13.3.1.6.1 Offset

Register	Offset
HTX_TMG_GEN_DISP_LRC	10h

13.13.3.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	POSX															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	POSY															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.13.3.1.6.3 Fields

Field	Description
31-16	Value of the Lower Right Corner X

Table continues on the next page...

Field	Description
POSX	
15-0 POSY	Value of the Lower Right Corner Y

13.13.3.1.7 Data Enable Coordinates (HTX_TMG_GEN_DE_ULC)

Contains the Upper Left Corner coordinates of the Data Enable needed by the Timing Generator

13.13.3.1.7.1 Offset

Register	Offset
HTX_TMG_GEN_DE_ULC	14h

13.13.3.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	POSX															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	POSY															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.13.3.1.7.3 Fields

Field	Description
31-16 POSX	Value of the Upper Left Corner X. The DE starts at this location
15-0 POSY	Value of the Upper Left Corner Y

13.13.3.1.8 Data Enable Coordinates (HTX_TMG_GEN_DE_LRC)

Contains the Lower Right Corner coordinates of the Data Enable needed by the Timing Generator

13.13.3.1.8.1 Offset

Register	Offset
HTX_TMG_GEN_DE_LRC	18h

13.13.3.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	POSX															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	POSY															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.13.3.1.8.3 Fields

Field	Description
31-16 POSX	Value of the Lower Right Corner X
15-0 POSY	Value of the Lower Right Corner Y

13.13.3.1.9 Hsync Start and End (HTX_TMG_GEN_HSYNC)

13.13.3.1.9.1 Offset

Register	Offset
HTX_TMG_GEN_HSYNC	1Ch

13.13.3.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	END															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.13.3.1.9.3 Fields

Field	Description
31-16 START	Value of the Lower Right Corner X
15-0 END	Value of the Lower Right Corner Y

13.13.3.1.10 Vsync Start and End (HTX_TMG_GEN_VSYNC)

13.13.3.1.10.1 Offset

Register	Offset
HTX_TMG_GEN_VSYNC	20h

13.13.3.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	END															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.13.3.1.10.3 Fields

Field	Description
31-16 START	Value of the Lower Right Corner X
15-0 END	Value of the Lower Right Corner Y

13.13.3.1.11 Controls the Position of first IRQ from Timing Generator (HTX_TMG_GEN_IRQ0)

13.13.3.1.11.1 Offset

Register	Offset
HTX_TMG_GEN_IRQ0	24h

13.13.3.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	POS_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	POS_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.13.3.1.11.3 Fields

Field	Description
31-16 POS_X	Xposition for the interrupt generation
15-0 POS_Y	Yposition for the interrupt generation

13.13.3.1.12 Controls the Position of Second IRQ from Timing Generator (HTX_TMG_GEN_IRQ1)

13.13.3.1.12.1 Offset

Register	Offset
HTX_TMG_GEN_IRQ1	28h

13.13.3.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	POS_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	POS_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.13.3.1.12.3 Fields

Field	Description
31-16 POS_X	Xposition for the interrupt generation
15-0 POS_Y	Yposition for the interrupt generation

13.13.3.1.13 Controls the Position of Third IRQ from Timing Generator (HTX_TMG_GEN_IRQ2)

13.13.3.1.13.1 Offset

Register	Offset
HTX_TMG_GEN_IRQ2	2Ch

13.13.3.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	POS_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	POS_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.13.3.1.13.3 Fields

Field	Description
31-16 POS_X	Xposition for the interrupt generation
15-0 POS_Y	Yposition for the interrupt generation

13.13.3.1.14 Controls the Position of Fourth IRQ from Timing Generator (HTX_TMG_GEN_IRQ3)

13.13.3.1.14.1 Offset

Register	Offset
HTX_TMG_GEN_IRQ3	30h

13.13.3.1.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	POS_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	POS_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.13.3.1.14.3 Fields

Field	Description
31-16 POS_X	Xposition for the interrupt generation
15-0 POS_Y	Yposition for the interrupt generation

13.13.3.1.15 Background Color insertion for R or Y (HTX_TMG_GEN_BG0)

13.13.3.1.15.1 Offset

Register	Offset
HTX_TMG_GEN_BG0	34h

13.13.3.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				DEF_VAL											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.13.3.1.15.3 Fields

Field	Description
31-12 —	Reserved
11-0 DEF_VAL	Default Value of the Background Color for R or Y

13.13.3.1.16 Background Color insertion for G or Cb (HTX_TMG_GEN_BG1)

13.13.3.1.16.1 Offset

Register	Offset
HTX_TMG_GEN_BG1	38h

13.13.3.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				DEF_VAL											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.13.3.1.16.3 Fields

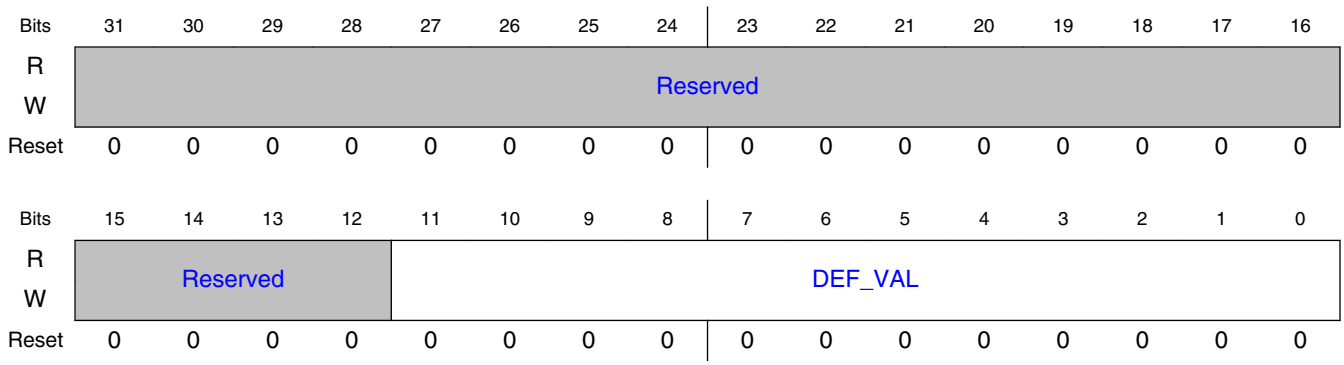
Field	Description
31-12 —	Reserved
11-0 DEF_VAL	Default Value of the Background color for G or Cb

13.13.3.1.17 Background Color insertion for B or Cr (HTX_TMG_GEN_BG2)

13.13.3.1.17.1 Offset

Register	Offset
HTX_TMG_GEN_BG2	3Ch

13.13.3.1.17.2 Diagram



13.13.3.1.17.3 Fields

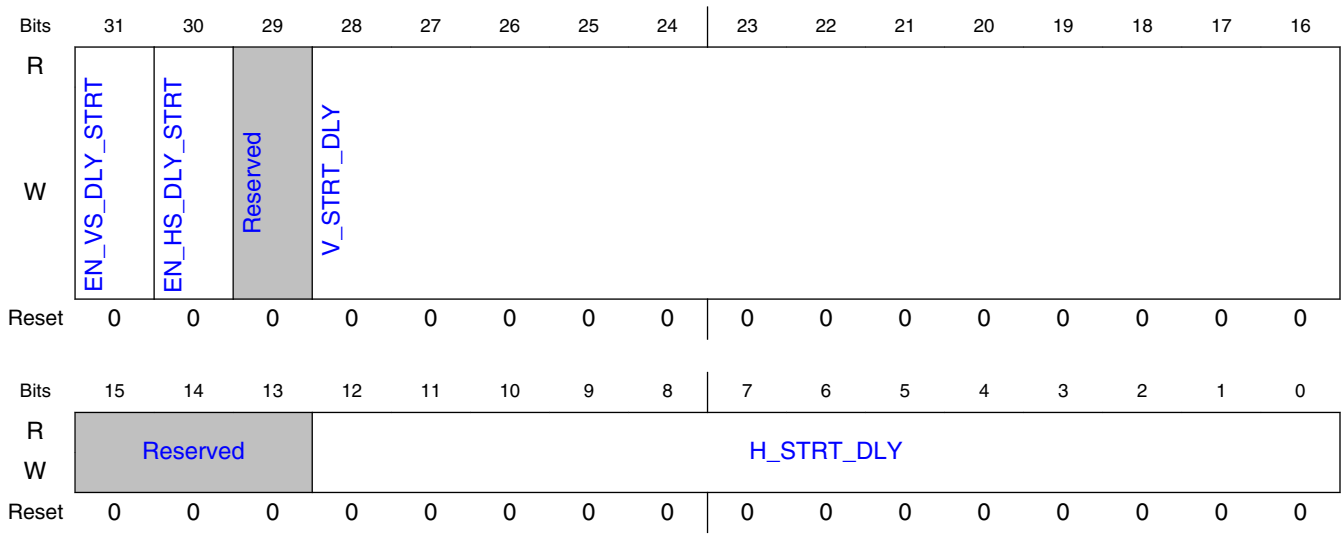
Field	Description
31-12 —	Reserved
11-0 DEF_VAL	Background Color Component for B or Cr

13.13.3.1.18 HStart and Vstart Delay Configuration (HTX_TMG_GEN_CFG)

13.13.3.1.18.1 Offset

Register	Offset
HTX_TMG_GEN_CFG	40h

13.13.3.1.18.2 Diagram



13.13.3.1.18.3 Fields

Field	Description
31 EN_VS_DLY_STRT	Controls the Vsync Start Generation. Enables to start either in VFront, Vsync or in Vback
30 EN_HS_DLY_STRT	Controls the Hsync Start Generation. Enables to start either in HFront, Hsync or in Hback
29 —	Reserved
28-16 V_STRT_DLY	Starting position of VCount to enable Starting in VBack, VSync or VFront
15-13 —	Reserved
12-0 H_STRT_DLY	Starting position of HCount to enable Starting in HBack, HSync or HFront

13.14 Image Signal Processor (ISP)

13.14.1 Overview

The Image Signal Processing (ISP) core is a complete video and still picture input block. It contains image processing and color space conversion (RAW Bayer to YUV) functions. The integrated image processing unit supports simple CMOS sensors delivering RGB Bayer pattern without any integrated image processing and also image sensors with integrated YCbCr processing.

13.14.1.1 Block Diagram

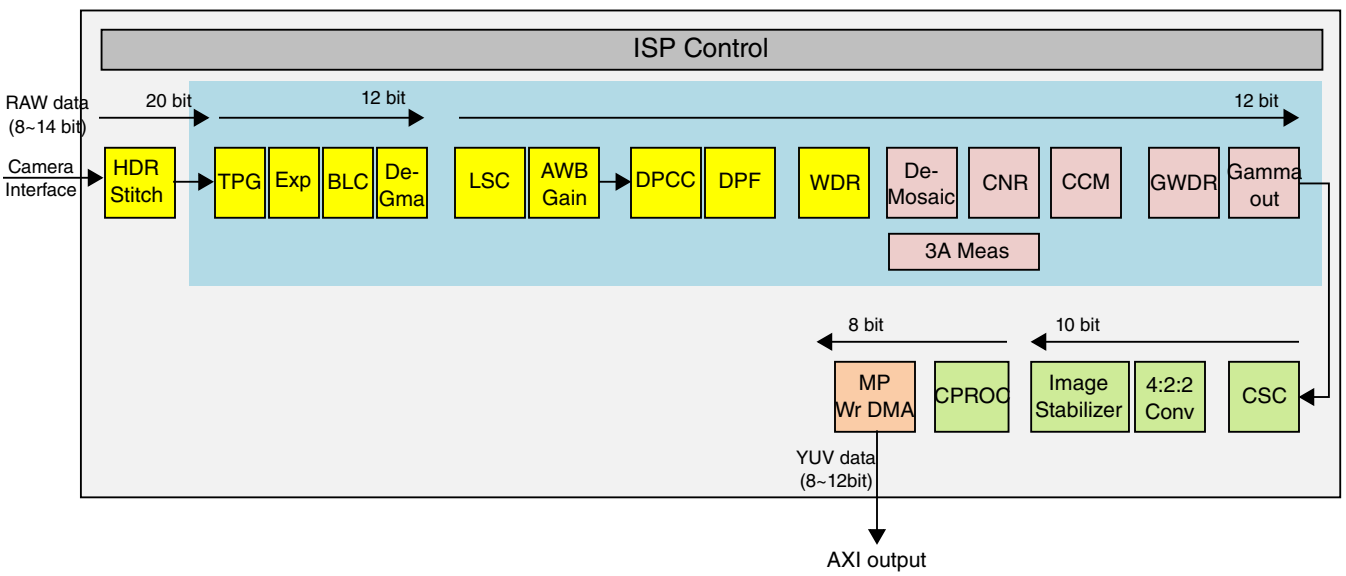


Figure 13-88. ISP High-level Block Diagram

13.14.1.2 Features

The ISP is connected to a MIPI-CSI interface and receives 1 pixel/clock from the CSI Rx Controller. Key features of the ISP include:

- Input Formats:
  - YCbCr420
  - YCbCr422
  - RAW8
  - RAW10

    - RAW12
    - RAW14
- Wide Dynamic Range (local tone mapping)
- High Dynamic Range (HDR)
    - Native HDR sensor with compand output and max 14-bit compressed input
    - 2-exposure and 3-exposure line-interleaved (staggered) HDR
        - Full resolution for multiple exposures is output by the image sensor, and the sensor outputs are interleaved on a per-line basis. No impact to vertical resolution, but frame rate is reduced by a factor of 2 (for 2-exposure HDR) or 3 (for 3-exposure HDR).
        - Referred to as DOL-HDR (Digital Overlap High Dynamic Range) and as Staggered HDR (12-bit x3)
- A separate Dewarp engine is also supported to adjust an image (i.e. fisheye lens) at a resolution up to the max resolution supported by the ISP.

NOTE

High Dynamic Range (HDR) is NOT supported with 2-exposure Line-interlaced HDR, referred to as iHDR and BME-HDR (Binning Multiple Exposure).

13.14.1.2.1 ISP Module

The following features are supported by the main ISP submodule:

- ISP input interface is ITU-R BT.601 compatible
- Variable sensor interface for RGB-Bayer Sensors
- Input sampling on positive or negative sample clocks
- Cropping of the output picture (to crop interpolation artifacts), also used for windowing
- Black level compensation
- Bad pixel detection/correction
- Lens shade correction
- Denoising pre-filter
- Bayer de-mosaic filter
- Chromatic Aberration Correction
- Filter (Noise reduction, Sharpness, Blurring)
- Programmable gamma correction for sensor adaptation and display correction
- Sensor crosstalk compensation
- Enhanced Chroma Noise Reduction (CNR)
- Automatic white balance measurements (AWB)
- Exposure measurement for AE (AEC/AGC)
- Auto focus measurement (AF)
- Histogram calculation

- Mechanical shutter control
- Flash light control

13.14.1.2.2 CPROC Module

The following features are provided by the color processing (CPROC) submodule:

- 2*10 bit data input interface
- 2*8 bit data output interface
- 10 bit resolution per color component
- Operates on 4:2:2 YCbCr format
- Full range and ITU-R BT.601 range are independently programmable for input and output.
- Contrast adjustment
- Brightness adjustment
- Saturation adjustment
- Hue adjustment

NOTE

Full range and ITU-R BT.601 range are independently programmable for input and output. Contrast adjustment has to be done with contrast register and brightness register settings.

13.14.2 Functional Description

13.14.2.1 Block Descriptions

The main functional blocks of the ISP are described in the table below.

Table 13-41. ISP Functional Blocks

ISP Block	Description
Input Port	The ISP connects to an external CMOS sensor. Supported sensors are MIPI CSI. It can support 8-14-bit data width. Different Bayer Pattern (R, Gr, Gb or B) can also be programmed.
HDR Stitch	If using DOL/staggered HDR sensor, the ISP can drive the sensor in different exposures and stitch it together. This allows for a linear sensor to be extended in dynamic range, less noise. This block supports 2-dol or 3-dol HDR and does not require external DRAM.
Test Patten Generator (TPG)	Used for ISP diagnostics. Can be programmed to generate a variety of color and random test patterns.
Expand LUT (Exp)	When a build-in HDR sensor with a compressed output is used (e.g. OV2775), this 64 bin programmable look-up table reverses the image sensor's compression function, producing a normalized data path for the following processing of the ISP.

Table continues on the next page...

Table 13-41. ISP Functional Blocks (continued)

ISP Block	Description
Black Level Compensation (BLC)	Adjusts for given black level from sensor. Normalizes black value for following processing of ISP pipeline.
De-gamma	This programmable look-up table is for making the sensor linearization.
LSC	Mesh shading correction: Programmable unequal interval per-color mesh look-up table with bi-linear interpolation. It is co-working with the AWB firmware algorithm.
Automatic white balance measurements (AWB)	AWB statistics values are provided per zone (32x32 zones, 4 Bayer Planar). Used by the AWB firmware algorithm to calculate per channel gains and color matrix values.
AE	AE statistics are provided on a per-zone basis (32x32 zones), which is employed for the AE algorithm (in firmware) to calculate exposure and gain.
Defect Pixel Cluster Correction (DPCC)	Both table-based static and dynamic defect pixel correction is included. Static defect pixel correction is managed by a locally stored location LUT while dynamic correction detects and removes defective pixels on-the-fly.
Denoising Pre-Filter (DPF)	2D 9x9 Bayer bilateral noise filter to remove sensor noise, and keep edge and details.
Wide Dynamic Range(WDR3)	VSI dynamic range compression / local tone mapping engine. This block is used to achieve natural visibility based on the local contrast improvement model for single-frame, stitched frame and sensor-build-in HDR frame. Support pixel-by-pixel enhancement auto-control.
Demosaic	Converts the RAW Bayer pipeline data into 12-bit RGB. The demosaic algorithm employed is an advanced 8 directional context-based interpolation, incorporating de-moiré, de-false color, de-purple and edge-aware blur/sharpness controls. Extreme care is taken to preserve details.
Chroma Noise Reduction (CNR)	Removes small color noise especially on mobile camera.
Color Correction Matrix (CCM)	Programmable 3x3 color matrix and 3x1 offset used by the AWB firmware algorithm to adjust color on-the-fly based on given light sources.
GWDR	Programmable LUT based global tone curve to improve global luminance.
Gamma out	Programmable 64-node piecewise linear look-up-table to provide gamma correction for data regulation, separated LUT planar for R,G,B plane.
Color Space Conversion (CSC)	3x3 programmable color matrix capable of converting RGB to YUV444 standard color spaces.
Conv422	Conversion from YUV444 to YUV422
IS (CROP)	The output can be cropped to a desired programmable size (Height x Width).
Color Processing (CPROC)	Implements global contrast, brightness, hue, saturation regulation.
Dynamic Memory Access (DMA) Writer	Each output of the ISP includes a DMA writer block which drives 128 Bytes AXI master(s) to write the output image data to memory on-the-fly. Supports RGB and YCbCr/YUV modes. In YUV modes, planar and semi-planar modes are supported with programmable Horizontal & Vertical down-sampling (e.g. YUV 4:2:2 and 4:2:0 are supported). Main path output also supports output Bayer data.

13.14.2.2 Input Format Support

13.14.2.2.1 Sensor Inputs

The sensor interface is implemented as an ITU-R BT.656 compliant video data interface with included timing reference codes or optionally as an ITU-R BT.601 interface with separate horizontal and vertical synchronization inputs. Supported video formats are:

- RGB BAYER data in all possible pattern layouts
- YCbCr 4:2:2 with the following luminance/chrominance orderings:
  - YCbYCr
  - YCrYCb
  - CrYCbY
  - CbYCrY

Horizontal and vertical synchronization signals can be interpreted as synchronization pulses or reference signals independently.

13.14.2.2.2 MIPI Inputs

The ISP can support the following image data formats through the MIPI interface (see the MIPI-CSI2 standard for details about the image data formats and byte sequences):

- YUV 4:2:2 8bit / 10bit (to YUV processing chain)
- non-legacy YUV 4:2:0 8bit / 10bit with cosited chroma sampling (direct to system memory)
- non-legacy YUV 4:2:0 8bit / 10bit with non-cosited chroma sampling (direct to system memory)
- legacy YUV 4:2:0 8bit (direct to system memory)
- RAW 6-bit / 7-bit image data (direct to system memory)
- RAW 8-bit / 10-bit / 12-bit / 14-bit image data (to RGB Bayer data processing chain)

13.14.2.3 Output Data Format Support

13.14.2.3.1 Main Path Format Support

The following output data formats are supported:

- YCbCr 4:2:2 or 4:2:0 either planar or semi-planar or interleaved.
- 8 to 12 bit RAW data with data end signaling.

13.14.2.4 Clock Domains

There are four independent clock domains in the ISP, and they are detailed in the table below.

Table 13-42. ISP Clocks

Clock	Signal name	Description
Core Clock	clk	Core clock is the main clock domain used by the ISP modules and is the biggest clock domain in the ISP.
Sensor Clock	sclk	The sensor clock domain is selected from the sensor interface and is the data input clock.
AHB Clock	hclk	The AHB Clock is the AHB Interface clock.
AXI Clock	m_hclk	The AXI Clock is the AXI Interface clock.

The communication between the different domains in ISP occurs mostly by way of asynchronous FIFOs. All clocks are asynchronous to each other. There is no communication between AHB and AXI clock domains.

13.14.2.5 Clock Gating

Automatic localized clock gating is employed in the ISP to minimize dynamic power consumption. An additional clock gating mechanism is employed in the ISP submodule. Each ISP submodule contains it's own local configuration registers. For optimized power consumption, these registers are clocked during read/write data transfers only. Clock gating is settable, but is handled automatically by default within the control unit.

13.14.2.6 Reset

The ISP supports an asynchronous reset. The reset is programmable via the Control Unit. An asynchronous reset for the sensor clock domain is generated from the system reset. A soft reset for all ISP registers is also supported, and behaves like the asynchronous hardware reset.

13.14.2.7 External Signals

Table 13-43. External Signals

Signal	Description
ISP_FL_TRIG_n	Flashlight trigger signal extern input

Table continues on the next page...

Table 13-43. External Signals (continued)

Signal	Description
ISP_FLASH_TRIG_n	Flashlight control trigger output
ISP_PRELIGHT_TRIG_n	Pre-light control trigger output
ISP_SHUTTER_OPEN_n	Shutter open
ISP_SHUTTER_TRIG_n	Shutter trigger input

13.14.2.8 Shadow Registers

Most of the configuration registers inside of ISP hold static programming information (e.g., horizontal and vertical synchronization polarities at the sensor interface). The registers are accessible by the CPU at any time. In some cases it is required to dynamically re-program the active processing data path with new parameters. All of these new parameters must not be changed during processing of a frame, but between frames or at the end of a frame. Because it cannot be ensured that the re-programming can be fully completed during the vertical blanking period between two frames, shadow registers are used.

These shadow registers are identified by the suffix _SHD. They hold the configuration values that are used for processing. Their corresponding configuration registers do not directly affect the hardware, but store the new values until the configuration update is triggered. The shadow registers get their value from the respective configuration register by an immediate or frame synchronous update procedure. Registers may contain a mixture of normal bits and bits which have corresponding shadow bits.

NOTE

ISP functionality can be accessed from Software Interface.

13.15 DeWarp

13.15.1 Overview

The Dewarp Engine provides high-performance dewarp processing for the correction of the distortion that is introduced in images produced by fisheye and wide angle lenses. It is implemented with a line/tile-cache based architecture. With configurable address mapping look up tables and per tile processing, it successfully generates a corrected output image.

13.15.1.1 Block Diagram

The block diagram of the Dewarp Engine is shown below.

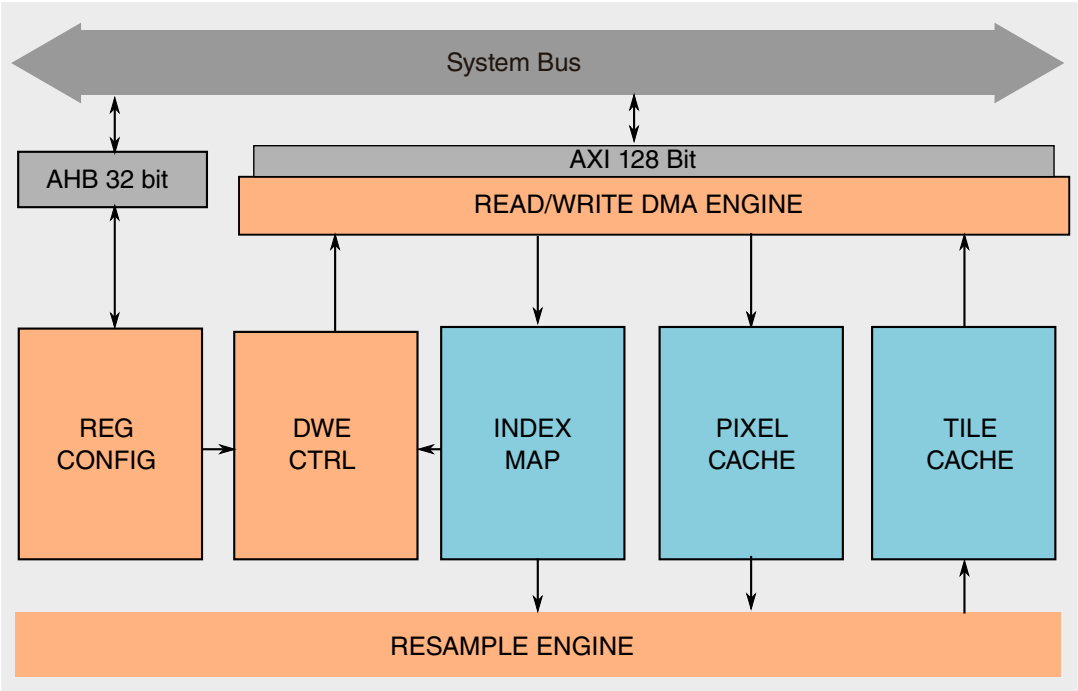


Figure 13-89. Dewarp Engine Block Diagram

13.15.1.1.1 Block Descriptions

The main functional blocks of the Dewarp Engine are described in the table below.

Table 13-44. Dewarp Engine Functional Blocks

Dewarp Engine Block	Description
Register Configuration	Generates configurable registers for dewarp control
DMA Fetch Engine	Fetches the LUT grid map and prefetches the distorted image
DWE Control	Provides configuration and execution controls for the resample engine
Index Map	Collects the image data that needs interpolation
Pixel Cache	Cache where the data manipulation occurs
Resample Engine	Performs bilinear interpolation for sets of four neighboring pixels and sends 4 byte data to the tile cache
Tile Cache	Cache for tiled output data

13.15.1.2 Features

The main features of the Dewarp Engine include:

- Line/Tile-cache based architecture and configurable map to support various lens distortions
- YUV422 and YUV420 inputs and outputs
- Configurable bilinear interpolation engine
- Grid map with 16x16 pixel macroblocks indexed using X, Y vertex coordinates
- Fisheye correction
- Wide Field of View (FOV) correction
- Keystone correction
- Scaling and 4PTZ view

13.15.2 Functional Description

The following sections provide functional information about the Dewarp Engine.

13.15.2.1 Pre-fetch and LUT Features

LUT (Look Up Table) content is fetched from DDR and used to generate the coordinates for the correct image. Macroblocks are prefetched as 16x16 MB tiles. When dewarp starts, data is fetched from the LUT according to an established fetch priority.

Table 13-45. Pre-fetch and LUT Support

LUT Data	DeWarp Support
LUT SRAM	1024 x128 bit
LUT max image size	4K image (4 lines)
Fetch Priority	Highest: Real Pixel Calculation Second Priority: Pre-fetch LUT read Lowest Priority: Pre-fetch LUT write
Macroblock (MB) Fetch	Pre-fetches image data in MBs (16 pixels x16 lines)

13.15.2.2 Pixel Calculation and Distortion Correction

The Dewarp Engine includes algorithms which provide correction for a variety of distortion effects typical of image output from pinhole cameras with fisheye or wide angled lenses.

Table 13-46. Distortion Correction Support

Distortion Correction Algorithm	Typical Visible Distortion Effects
Radial distortion correction	barrel distortion pincushion distortion mustache distortion fisheye distortion
Tangential distortion correction	parallax distortion wide field of view (FOV) distortion perspective transformation keystone effects
Scaling	size, 4PTZ(pan-tilt-zoom) view

13.15.2.3 Dewarping Remap

The image is divided into many small 16x16 blocks. If the width of the image is not divisible by 16, the size of the rightmost block is the remainder. The dewarping map only saves the vertex coordinates of the block.

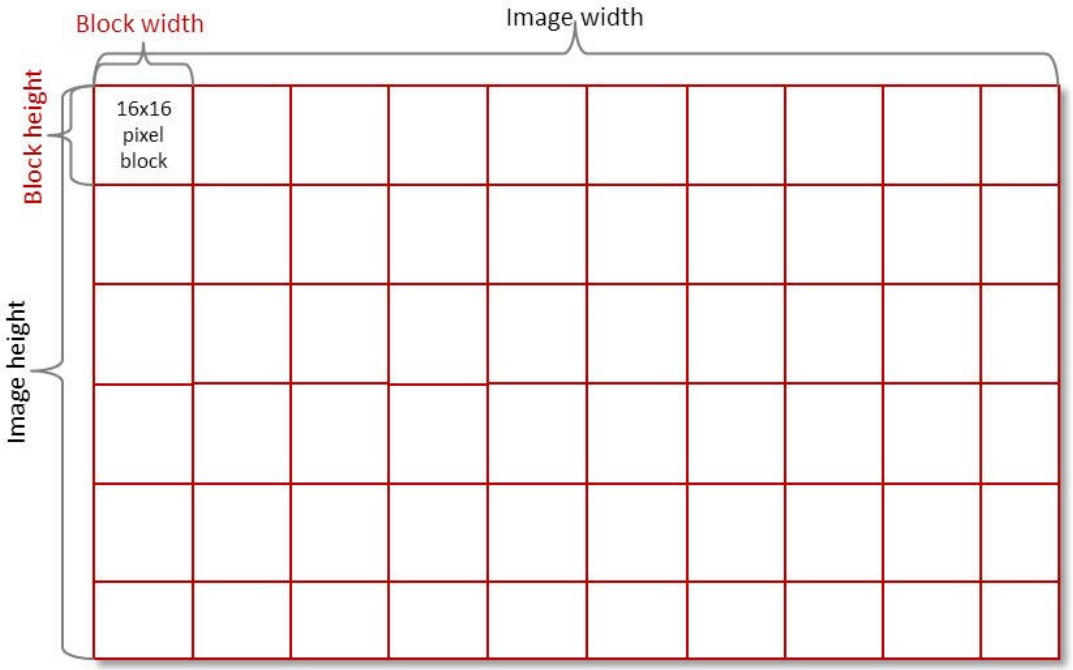


Figure 13-90. Grid of Image Blocks for Dewarping Map

The dewarping grid map is comprised of vertex coordinates for x and y. Each x, y coordinate register uses 16 bits to record the coordinate address, with the Y coordinate in the upper bits and X in the lower bits.

- 12 bits is the integer part and 4 bits is the fractional part.

The dewarping remap contains two interpolations: coordinate interpolation and pixel interpolation as shown in the figure below.

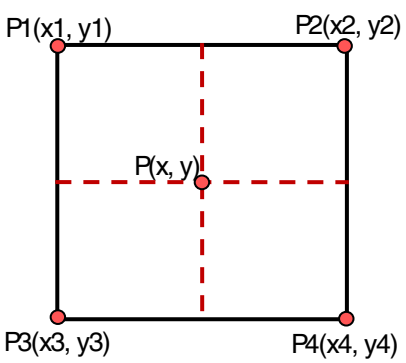


Figure 13-91. Remap with Coordinate and Pixel Interpolation

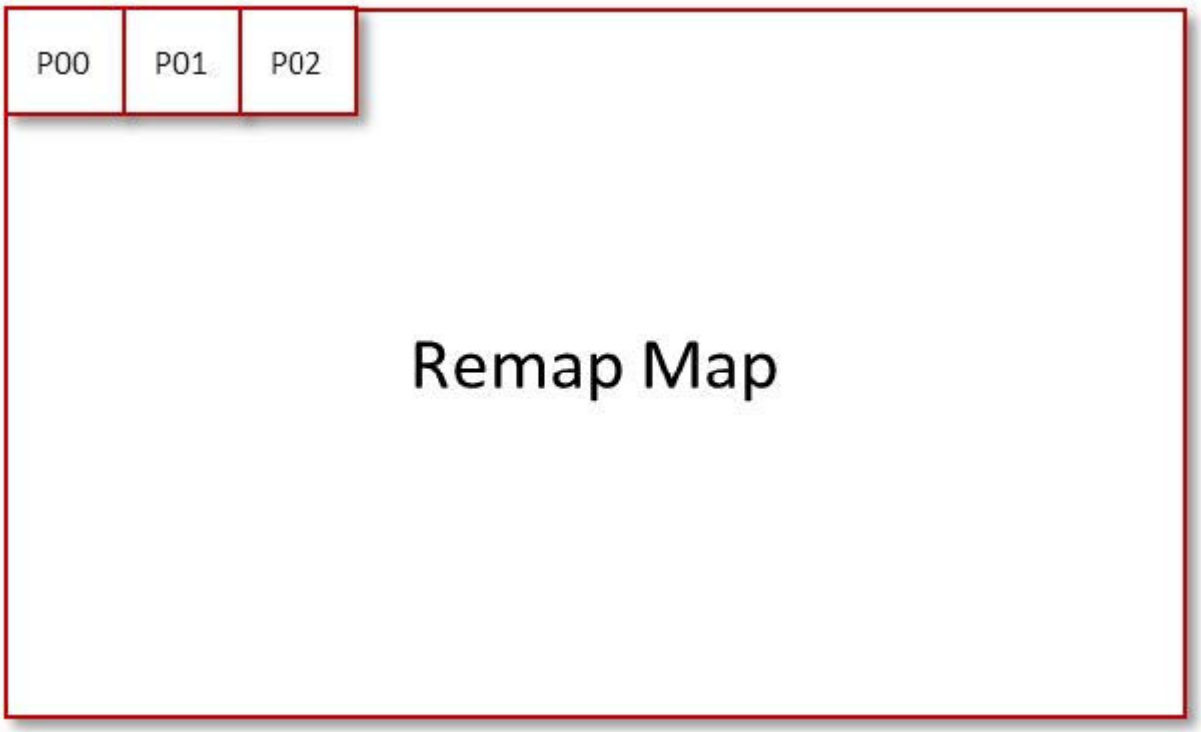


Figure 13-92. Remap Map

13.15.2.4 Output Generation

Once dewarp calculations are complete, the Dewarp Engine generates a write request for the image.

- If the mode is interleave, it generates one request, with burst length of 8 and data format YUYVY.
- If the mode is semi-planar, it generates two types of requests: one is a Y request with a burst length of 4 and data format YYYY; the second is a UV request with a burst length of 4 and data format UVUVUV.

13.15.2.5 Image Format Support

The following YUV formats are supported.

Table 13-47. YUV Input and Output Format Support

Format	Input Support	Output Support	Bits per pixel
YUV420 Semi-planar	YES	YES	12
YUV422 Interleave	YES	YES	16

Table continues on the next page...

Table 13-47. YUV Input and Output Format Support (continued)

Format	Input Support	Output Support	Bits per pixel
YUV422 Semi-planar	YES	YES (only when Input is YUV422)	16

13.15.3 External Signals

There are no external signals pinned out for this module.

13.15.4 Initialization and Application Information

The Dewarp Engine cannot be used as a standalone module. It can only be used in conjunction with the Image Signal Processor (ISP) module.

13.16 Machine Learning Neural Processing Unit (NPU)

13.16.1 Overview

The Neural Processing Unit (NPU) provides hardware acceleration for AI/ML workloads and vision functions. The IP provides enhanced performance for real-time use cases with hardware support for the OpenVX API.

13.16.2 Block Diagram

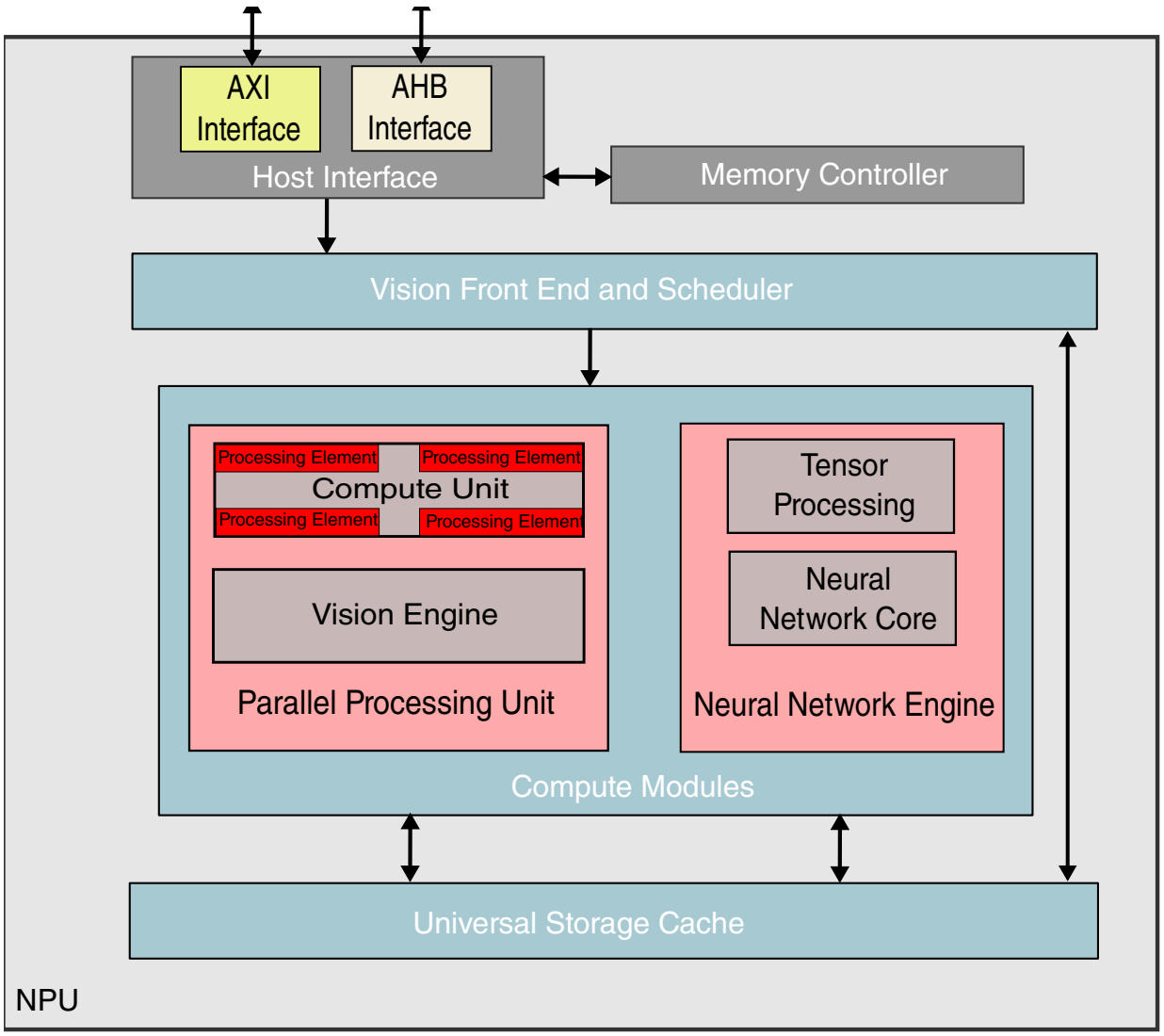


Figure 13-93. NPU High-level Block Diagram

13.16.2.1 Block Descriptions

The main functional blocks of the NPU core are described in the table below.

Table 13-48. NPU Functional Blocks

NPU Block	Description
Host Interface	Allows the NPU to communicate with external memory and the CPU through the AXI / AHB bus. In this block data crosses clock domain boundaries.

Table continues on the next page...

Table 13-48. NPU Functional Blocks (continued)

NPU Block	Description
Memory Controller	Internal memory management unit that controls the block-to-host memory request interface.
Vision Front End	Inserts high level primitives and commands into the vision pipeline.
Neural Network Core	Provides parallel convolution MAC for recognition functions using 8 or 16 bits integer.
Tensor Processing Fabric	Provides data preprocessing and supports compression and pruning for multi-dimensional array processing for Neural Nets.
Compute Unit	SIMD processor programmable execution unit that perform as a Compute Unit. The NPU block has 1 vec4 Parallel Processor Unit which also acts as 4 Processing Elements.
Vision Engine	Provides advanced image processing functions.
Universal Storage Cache	Cache shared between the Vision Front End and the Parallel Processing Unit.

13.16.3 Features

Key features of the NPU block include:

- OpenVX compliance, including extensions
- Convolutional Neural Network acceleration
- IEEE 32-bit floating-point pipeline in PPU shaders.
- Ultra-threaded parallel processing unit
- Low bandwidth at both high and low data rates
- Low CPU loading
- MMU functionality supported
- Performance Counters for DMA Profiling
- Data transfers between Neural Network Engines and the Parallel Processing Unit, with 256KB SRAM as local storage
- Neural Network Engine and Parallel Processing Unit synchronization with hardware semaphore

13.16.3.1 API Support and Architecture Features

The following table describes API support and architectural features of the NPU block.

Table 13-49. NPU Architecture Features

Feature	NPU Support
Primary APIs	OpenVX with Neural Network Extensions
Drivers	OpenVX

13.16.4 Functional Description

13.16.4.1 Parallel Processing Unit

The Parallel Processing Unit (PPU) features are noted below:

Table 13-50. Parallel Processing Unit Features

Feature	Supported
Parallel Processor Execution unit	1 compute unit, SIMD4, SFP32 Trans
FP denorm and rounding options	Denorms are flushed to zero. Supports round to nearest even and round to zero.
Maximum number of instructions	1 M (with I-cache)
Maximum number of streams	8
Maximum number of threads inflight per PPU core	256
Image Load/Store Instructions	Yes
Conditional branch support	GT, LT, EQ, GE, LE, NE, ISNAN, ISFINITY, ISINFINITY, ISNORMAL, AND, OR,XOR, NOT, ANYMSB, ALLMSB, SELMSB
Control flow instructions	Yes
Instruction rate	1-cycle throughput for all instructions
Standard derivatives	Yes
Integer pipeline	Support 8, 16, 32-bit integer operations
Universal Storage Cache	16 KB flexible allocation for both local shared memory and L1 cache

13.16.4.2 Neural Network Engine

The Neural Network Engine support features are in the table below:

Table 13-51. Neural Network Engine Features

Feature	Supported
MAC per cycle	1152
Applications	Classification, Detection, Segmentation
Network Topologies	Linear, MIMO, Fully Connected, Fully Convolutional
Network Size	2048 (Software driver limitation)
Inference Engine	TensorFlow Lite, Arm NN, ONNX, eIQ™ Inference with DeepViewRT
Configurable and Programmable	Yes
Network parameter deep compression	Yes
RNN Support	Yes

13.16.4.3 Tensor Processor

The Tensor Processor fabric support features are in the table below:

Table 13-52. Tensor Processor Features

Feature	Supported
Data format support	8/16-bit integer
Zero Weight Skipping	Yes, saving execution time
Intermediate Data Shuffling	Yes, saving CPU overhead
Local Memory for NN Weights andIntermediate Data Buffering	Yes
Local Buffer Configurations	0KB
Pooling	Max, average
Unpooling	Yes
Activation	ReLU, Leaky ReLU (LUT for other types)
Normalization	Yes
Region Proposal Support	Yes

13.16.4.4 Image Format Support

OpenVX image formats are supported. Packed formats are supported so that multiple pixels can be loaded in one shot.

13.16.4.4.1 OpenVX Support

Support for OpenVX includes the following capabilities:

Table 13-53. OpenVX Hardware Support

Feature	Supported
Enhanced Vision Instructions	EVIS 2 Instruction Set
Max kernel size	Up to 11x11, programmable
3x3 Image Filter modes	HorzMin3, HorzMed3, HorzMax3, VertMin3, VertMed3, VertMax3
Interpolation support	Lerp
Histogram support	IndexAdd, AtomicAdd
8-bit Dot Products	32x1, 16x2, 8x4, 4x8
16-bit Dot Products	16x1, 8x2 , 4x4, 2x8
Bit Manipulation modes	BitExtract
Packed Image Load/Store instructions	ImgLoad, ImgStore
Other EVIS instructions	AbsDiff, AccSq, MulShift, Clamp
Wide Uniform support	512-bit uniform reads per instruction

13.16.4.4.2 OpenVX Image Formats

The following OpenVX image formats do not require conversion.

Table 13-54. Vision Formats for OpenVX

OpenVX Formats	Description
VX_DF_IMAGE_VIRT	A virtual image of no defined type.
VX_DF_IMAGE_RGB	A single plane of 24-bit pixel as 3 interleaved 8-bit units of R then G then B data. This uses the BT709 full range by default.
VX_DF_IMAGE_RGBX	A single plane of 32-bit pixel as 4 interleaved 8-bit units of R then G then B data, then a don't care byte. This uses the BT709 full range by default.
VX_DF_IMAGE_NV12	A 2-plane YUV format of Luma (Y) and interleaved UV data at 4:2:0 sampling. This uses the BT709 full range by default.
VX_DF_IMAGE_NV21	A 2-lane YUV format of Luma (Y) and interleaved VU data at 4:2:0 sampling. This uses the BT709 full range by default.
VX_DF_IMAGE_UYVY	A single plane of 32-bit macro pixel of U0, Y0, V0, Y1 bytes. This uses the BT709 full range by default.
VX_DF_IMAGE_YUYV	A single plane of 32-bit macro pixel of Y0, U0, Y1, V0 bytes. This uses the BT709 full range by default.
VX_DF_IMAGE_IYUV	A 3 plane of 8-bit 4:2:0 sampled Y, U, V planes. This uses the BT709 full range by default.
VX_DF_IMAGE_YUV4	A 3 plane of 8 bit 4:4:4 sampled Y, U, V planes. This uses the BT709 full range by default.
VX_DF_IMAGE_U8	A single plane of unsigned 8-bit data. The range of data is not specified, as it may be extracted from a YUV or generated.
VX_DF_IMAGE_U16	A single plane of unsigned 16-bit data. The range of data is not specified, as it may be extracted from a YUV or generated.
VX_DF_IMAGE_S16	A single plane of signed 16-bit data. The range of data is not specified, as it may be extracted from a YUV or generated.
VX_DF_IMAGE_U32	A single plane of unsigned 32-bit data. The range of data is not specified, as it may be extracted from a YUV or generated.
VX_DF_IMAGE_S32	A single plane of unsigned 32-bit data. The range of data is not specified, as it may be extracted from a YUV or generated.

13.16.4.4.3 Vision Formats Requiring Conversion

Formats with eight (8) bits (or a multiple of 8 bits) per component typically do not require any conversion. Formats which do not have a component size which is a multiple of 8-bits usually require conversion. A Vision instruction can be used to do conversion if needed; conversion costs one instruction. Conversion will not be done if the application does not require it.

13.16.4.5 Power Modes

The NPU supports four (4) programmable, software-controlled power states that affect power consumption of the NPU. These states are: ON, IDLE, SUSPEND, and OFF.

Table 13-55. NPU Power States

Power State	Workload	Description
ON	Full	NPU runs at full clock speed. No clock gating.
IDLE	Minimal	NPU runs at 1/64 clock speed. No clock gating.
SUSPEND	None	NPU runs at 1/64 clock speed. AXI/core/SH are clock gated. FE stopped.
OFF	None	Same as SUSPEND, plus when this state is switched out then the NPU hardware is re-initialized. External power to the NPU can be disabled.

The kernel layer manages the power state transitions, and each state corresponds to an operating mode of the NPU:

1. Active mode
   - Power state is ON.
   - NPU is actively processing commands. One or more blocks are not in idle mode.
2. Idle mode
   - Power state is IDLE.
   - NPU is not processing any commands. All modules in the pipeline are in idle state.
   - The input clocks are on, but most of the NPU internal clocks are automatically gated off.
   - Frequency skipping is turned on with 1/64 pulse.
   - SoC reduces ACLK and HCLK frequency by 20x external to the NPU.
   - The NPU can start command processing instantly without any delay.
3. Standby mode
   - Power state is SUSPEND.
   - All input clocks to the NPU are shut off.
   - NPU clocks must be turned on and given enough time to reach idle state. Please refer to Resets section for information on the timing.
   - Once the NPU is in idle state, command processing can start.
4. Sleep mode
   - Power state is OFF.
   - The entire NPU is powered off through power gating.
   - The SoC must go through a power-on sequence in order to bring the NPU to idle state before starting command processing.

13.16.4.5.1 Low Power State Control

This NPU supports two low power state control schemes. One is the AXI Low Power state control scheme specified in the AXI spec. The other is the NPU Power Management scheme, controlled through a register which is accessible from the AHB.

13.16.4.5.1.1 AXI Low-Power Interface

The AXI Low-Power Interface is standardized in the AXI specification, and no interactions are required. However, it does not provide finer control over how the power management is configured. When exiting from the low-power state, the AQHiClock Control Register is initialized to the default value, enabling all functional blocks.

13.16.4.5.1.2 Power Management

There are two registers (AQHiClockControl and AQHiIdle) in the Host Interface (HI) module dedicated for power management. Software can control power management through these registers. Unlike the AXI Low-Power Interface, no automatic sequencing is involved in this scheme. The software needs to make sure that a block is in idle before shutting down the clock to the block. This scheme also offers a register interface to the clock scaling mechanism available in the graphics core.

13.16.4.6 Clock Domains

There are four independent clock domains in the NPU, and they are detailed in the table below.

Table 13-56. NPU Clocks

Clock	Signal name	Description
Core Clock	clkCore	Core clock is the main clock domain used by the NPU modules other than the PPU, NN, and USC.
PPU Clock	clkSh	The PPU clock domain contains the PPU, NN, and USC.
AHB Clock	hclk	The AHB Clock is the AHB Interface clock.
AXI Clock	aclk	The AXI Clock is the AXI Interface clock.

NOTE

The PPU Clock (clkSh) is the same source as the Core Clock (clkCore).

The communication between the different domains in NPU occurs mostly by way of asynchronous FIFOs. All clocks are asynchronous to each other. There is no communication between AHB and AXI clock domains.

The Core Clock in the NPU core can be scaled down dynamically without reprogramming its source. This scaling is controlled through the use of an internal register.

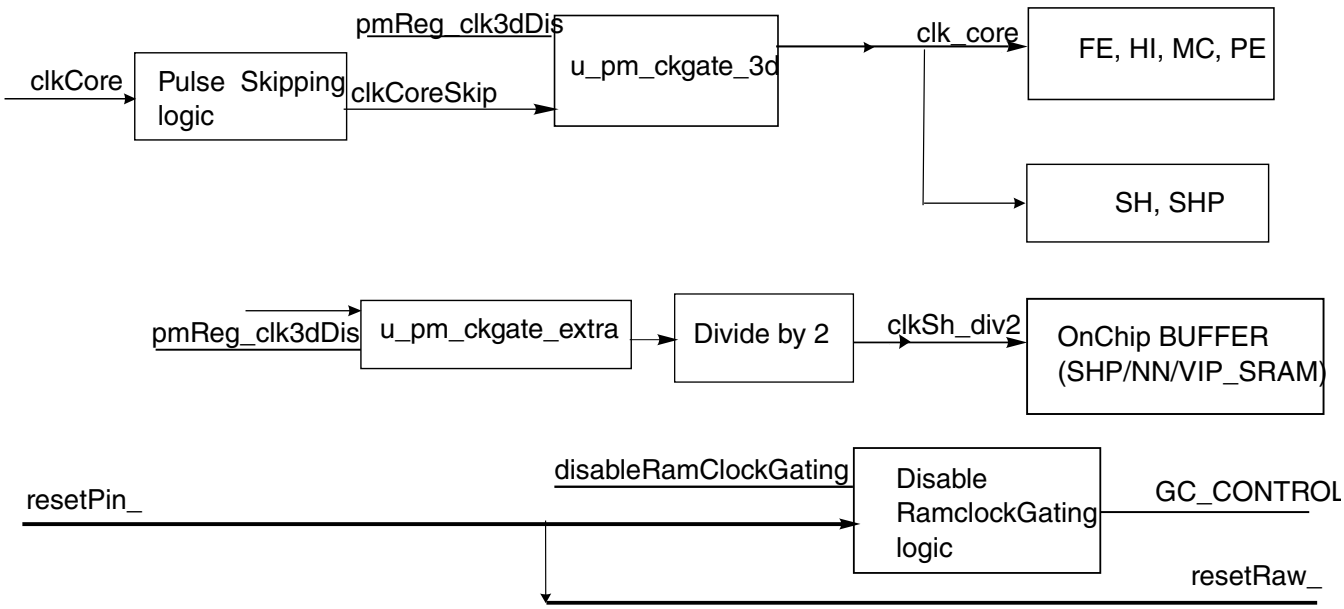


Figure 13-94. NPU Core Clock Logic Block Diagram

13.16.4.6.1 Clock Gating

Automatic localized clock gating is employed in the NPU to minimize dynamic power consumption.

13.16.4.6.1.1 Second Level Clock Gating

Block level clock gating is implemented in a majority of the blocks. If a block and the interface to the block are both idle, then the clock of that block will be gated automatically. This feature can be disabled by software.

13.16.4.6.2 Clock Disabling

The output of one of the clock gating blocks is for the Parallel Processing Unit (clkSh), and the output of the other clock gating block is for the rest of the NPU blocks (clkCore). The clock branches can be shut down independently through software control by setting the proper register bits. Please see Clock Control Register for more information.

13.16.4.6.3 Core Clock Frequency Scaling

A basic period of 64 clock cycles is chosen as the basis for clock skipping. Any number of clocks within that 64 clock period can be skipped. This yields a fine granularity of 1.6% (1/64).The scaling factor is a 7-bit value, which represents the number of clocks NOT to skip (the reset value is 64). Pulses will be skipped uniformly along the 64-cycle period to smooth out the power demand of the power supply.

Scaling is controlled by software through the use of control registers. The software determines the number of cycles not to skip within a 64-cycle period. The frequency scaling value is double-buffered as well, and it is updated at the beginning of a 64-cycle sequence. The following diagram shows an example of pulse skipping for the case of a 16 cycle period.(The actual period is 64 cycles, but 16 was chosen for ease of illustration).

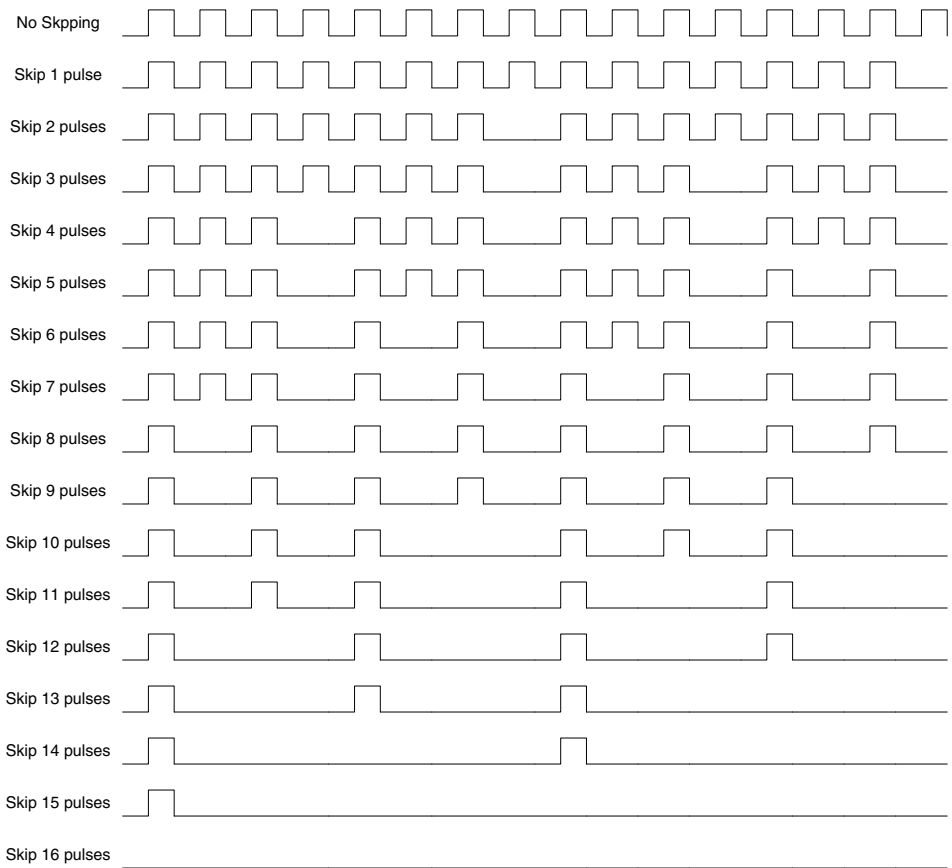


Figure 13-95. Pulse Skipping Example - Timing Diagram

Reducing the frequency of the core clock should not be a problem on the AXI side. The AXI interface is designed to handle any ratio of AXI clock vs. core clock. The AXI side can be stalled as well if the core clock side cannot accept the return data fast enough.

13.16.4.6.4 Clock Control Register

All clock control registers reside in the host clock domain.

Table 13-57. AQHiClockControl Register - Select Clock Control Bits

Bitfield	Access	Name	Description
0	R/W	CLK3D_DIS	Software core clock disable signal for core (clkCore) and 3D (clkSh) clocks. When set to “1”, these clocks are frozen. Reset value = 1’b0
1	R/W	CLK2D_DIS	Software core clock disable signal. For this core both bits CLK3D_DIS and CLK2D_DIS should be controlled by software. The AXI interface clock is the only block not stalled at that point. Reset value = 1’b0
8:2	R/W	FSCALE_VAL	Frequency scaling value. Reset value = 7’d64 (no pulse skipping)
9	R/W	FSCALE_CMD_LOAD	Frequency scaling load command. When writing a “1” to this bit, it updates the frequency scale factor with the value FSCALE_VAL[6:0]. The bit must be set back to “0” after that. Reset value = 1’b0

13.16.4.7 Resets

The NPU supports asynchronous resets that must be asserted for a minimum of of 32 core clock cycles (slowest clock). After deassertion, 128 core clock cycles (slowest clock) is required before starting any activity on AHB. During the 32 cycle assertion + 128 cycle wait, the core clock supplied into the NPU may be kept running. The clock may be gated before assertion, but because it is an asynchronous reset, it is not required.

13.16.4.8 Interrupts

The IP can send an interrupt signal to the host processor. The signal name is xaq2_intr, and it remains asserted until the host processor clears the interrupt by reading the interrupt acknowledge register (AQIntrAcknowledge). The interrupt is level triggered. The SOC should not use edge to detect the interrupt, as using edge to detect the GPU/NPU interrupt will miss some back to back interrupts. Each bit of AQIntrAcknowledge represents one of the 32 possible events that the NPU can signal to the host processor. By setting or clearing the bits of the interrupt enable register (AQIntrEnbl) the programmer can control which of those events will generate an interrupt. The bits of these registers have no predefined meaning, except for bit 31 which is used for AXI_BUS_ERROR. Instead, all of them are used by and controlled through the NPU driver. The NPU driver puts events with different event IDs (0~30) in the command buffer, and the NPU will generate an interrupt with the corresponding interrupt bit as set by the event ID in AQIntrAcknowledge. The NPU kernel driver and its interrupt

service routine (ISR) will process the interrupt accordingly to synchronize the driver/CPU with NPU. The meaning of the NPU interrupt bits are entirely private to the NPU driver. The event IDs (0~30) are arbitrarily used by the NPU driver thus the interrupt bits in AQIntrAcknowledge are also arbitrary. Chip designers need only to ensure that the xaq2_intr signal is connected to their interrupt controller. Since AQIntrEnbl is set to 0 at reset, the xaq2_intr signal will never be asserted during the chip validation phase.

13.16.5 Initialization

13.16.5.1 Power Off Sequence

The Power Off sequence for NPU is as follows:

- Chip software issues a request to go into "OFF" state or the NPU internal timer triggers a transition into the "OFF" state.
- NPU software then executes all the required preliminary steps to put the block into "OFF" state.
- NPU software issues a callback to gate the NPU clock and NPU power.
- Chip software initiates a power down sequence to the chip controller HW.
- Chip hardware asserts an isolation signal to turn on isolation from the NPU.
- Chip hardware asserts a power gate signal to turn off power to the NPU.
- Upon return of the hardware acknowledgement, chip software returns call-back status to NPU software.
- Upon receipt of a successful callback, NPU software puts the block into the "OFF" state.

13.16.5.2 Power On Sequence

The Power On sequence for NPU is as follows:

- Chip software requests a transition from the "OFF" state to the "ON" power state.
- NPU software issues a call back to enable clock and power for the NPU.
- Chip software initiates a power up sequence to the chip controller hardware.
- Chip hardware de-asserts the power gate signal to turn on power to the NPU block.
- Upon hardware acknowledgement, the chip Hardware turns on the clocks to the NPU block.
- Chip software performs an AHB domain reset.
- Chip Hardware de-asserts the isolation signal to turn off isolation from the NPU block.
- Chip software returns call-back status to NPU software.

- Upon a successful return, NPU Software performs an AHB write that will soft reset the core and AXI domains.
- NPU software enables FE and waits for commit

13.16.6 Memory Map and register definition

This section includes the NPU module memory map and detailed descriptions of key accessible registers.

The following are four modules that can be accessed via the AHB bus:

- Host Interface (HI)
- Power Management (PM)
- MMU Secure Zone (MMU)
- Memory Controller (MC)

The following table describes the registers, their corresponding AHB modules and addresses.

Table 13-58. Register Addresses and corresponding AHB Module

AHB Module	AHB Byte Offset	GPU DWord Offset
HI	0x000-0x0A8	0x0000-0x002A
PM	0x100-0x010C	0x0040-0x0043
TZ	0x388-0x3AC	0x00E2-0x00EA
MC	0x414-0x42C	0x0105-0x010B

Host Interface: The Host Interface is a bridge between the Memory Controller and the AXI interface. It also acts as a bridge between the core and the AHB bus. The Host Interface's register set contain the control settings for the clocks, AXI interface and interrupts. It also contains the identification registers and some counters that are used for debug. Users can access the Host Interface registers via the AHB bus.

Power Management Registers: The Power Management register set controls clock gating within the core. The graphics core allows the user to control the clock gating of each internal module independently of the other modules. Users can access the registers via the AHB Bus.

MMU Secure Zone Registers: These registers provide hardware mechanisms and base address information for access control.

Memory Controller Registers: All memory accesses that go to or from the AXI bus pass through the Memory Controller. It connects the Host Interface to the rest of the core. The Memory Controller's register set contains the controls for the memory interface, the

settings for the virtual memory page table, and the values of the offset registers. It also contains a variety of debug registers that are set by other blocks within the core. Users can access the Memory Controller's registers via the AHB bus.

NOTE

NPU functionality can be supported using SW interface.

13.16.6.1 NPU register descriptions

13.16.6.1.1 NPU memory map

NPU base address: 3850_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Clock Control Register (AQHiClockControl)	32	RW	0000_0900h
4h	Idle Status Register (AQHiIdle)	32	RO	7FFF_FFFFh
Ch	AXI Status Register (AQAxiStatus)	32	RO	0000_0000h
10h	Interrupt Acknowledge Register (AQIntrAcknowledge)	32	RO	0000_0000h
14h	Interrupt Enable Register (AQIntrEnbl)	32	RW	0000_0000h
78h	Total Cycles Register (gcTotalCycles)	32	RW	0000_7000h
100h	Module Power Level Control Register (gcModulePowerControls)	32	RW	0014_0020h
10Ch	Pulse eater Control Register (gcPulseEater)	32	RW	0159_0880h
388h	MMU Control Register (gcregMMUAHBControl)	32	RWONCE	0000_0000h
38Ch	MMU Table Array Base Lower 32-bit Address Register (gcregMMUAHBTableArrayBaseAddressLow)	32	RW	0000_0000h
390h	MMU Table Array Base Higher 32-bit Address Register (gcregMMUAHBTableArrayBaseAddressHigh)	32	RW	0000_0000h
394h	MMU Table Array Size Control Register (gcregMMUAHBTableArraySize)	32	RW	0000_FFFFh
398h	MMU NonSecure Address Register (gcregMMUAHBSafeNonSecureAddress)	32	RW	0000_0000h
39Ch	MMU Secure Address Register (gcregMMUAHBSafeSecureAddress)	32	RW	0000_0000h
3A4h	Command Buffer Control Register (gcregCmdBufferAHBCtrl)	32	WO	0000_0000h
3A8h	MMU Host Interface Control Register (gcregHiAHBControl)	32	RW	0000_0000h
3ACh	MMU AXI Configuration Register (gcregAxiAHBConfig)	32	RW	0022_2200h
414h	Memory Debug Register (AQMemoryDebug)	32	RW	3C00_0000h
42Ch	Register Timing Control Register (AQRegisterTimingControl)	32	RW	0003_0000h
654h	Command Buffer Base Address Register (AQCmdBufferAddr)	32	WO	0000_0000h
664h	Command Decoder Address Register (AQFEDebugCurCmdAdr)	32	RO	0000_0000h

13.16.6.1.2 Clock Control Register (AQHiClockControl)

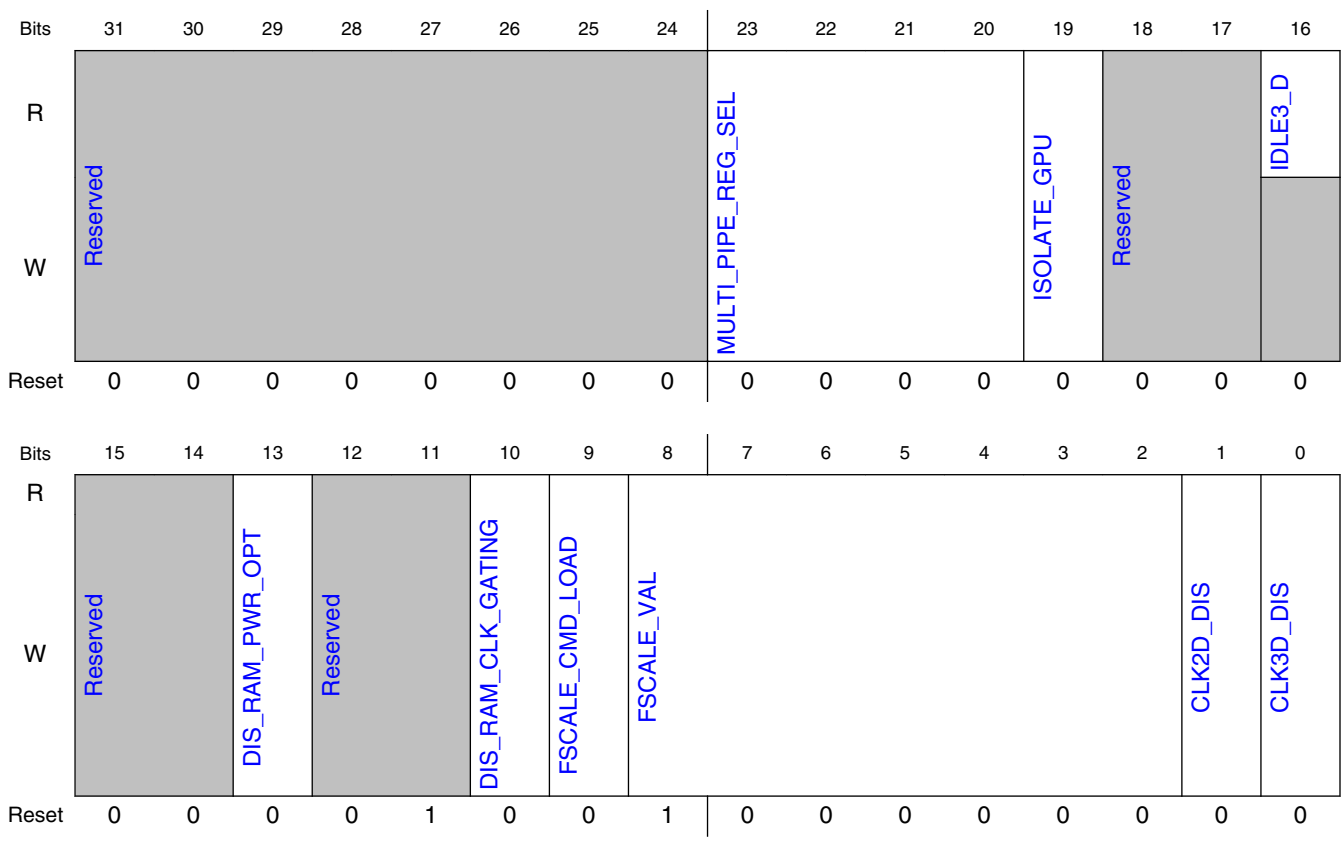
13.16.6.1.2.1 Offset

Register	Offset
AQHiClockControl	0h

13.16.6.1.2.2 Function

This register is used for clock control

13.16.6.1.2.3 Diagram



13.16.6.1.2.4 Fields

Field	Function
31-24	Reserved

Table continues on the next page...

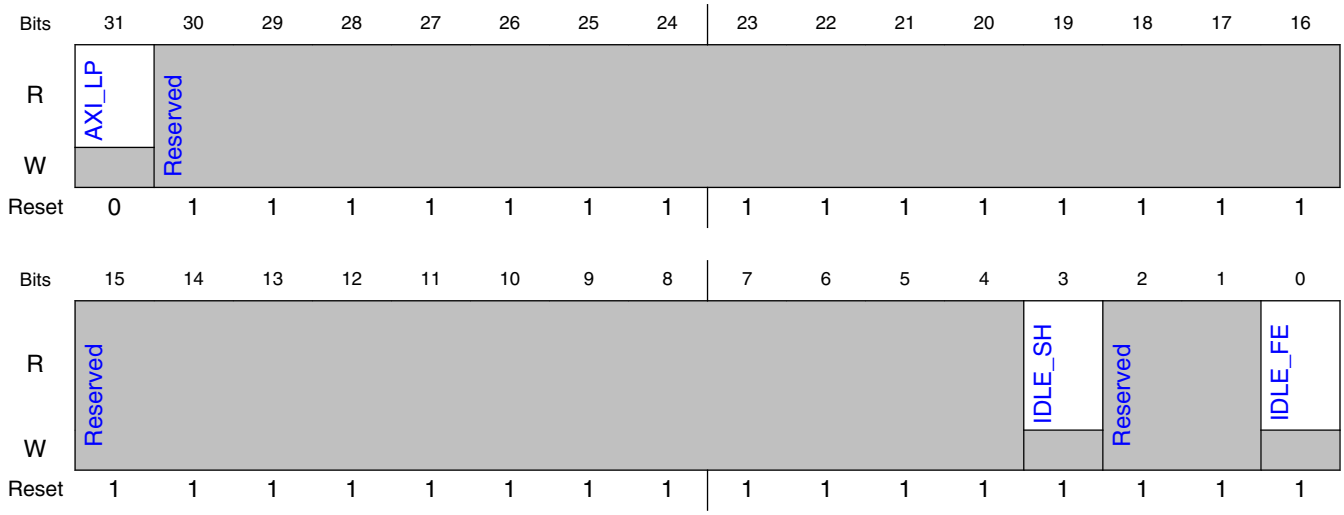
Field	Function
—	
23-20 MULTI_PIPE_REG_SEL	Multiple Pipe Register Select This bit is determines which HI/MC to use while reading registers.
19 ISOLATE_GPU	Isolate GPU This bit is used for power on/off, isolation only for multi-core GPUs. Please refer to Initialization section for information on its usage.
18-17 —	Reserved
16 IDLE3_D	3D pipe is idle
15-14 —	Reserved
13 DIS_RAM_PWR_OPT	Disable RAM power optimization
12-11 —	Reserved
10 DIS_RAM_CLK_GATING	Disable clock gating for RAMs
9 FSCALE_CMD_LOAD	Core clock frequency scale value enable If this bit is set and FSCALE_VAL=0 (an invalid combination), the HREADYOUT output signal will get stuck to 0. 1b - The frequency scale factor is updated with the value FSCALE_VAL[6:0]. The bit sets back to 0 after that.
8-2 FSCALE_VAL	Core clock frequency scale value
1 CLK2D_DIS	Disable 2D clock This bit is connected to the software clock disable signal. For this core, both bits CLK3D_DIS and CLK2D_DIS should be controlled by software. The AXI interface clock is the only block not stalled at that point.
0 CLK3D_DIS	Disable 3D clock Software core clock disable signal for 3D modules (clk_3d) clock. 1b - The clock is frozen

13.16.6.1.3 Idle Status Register (AQHildle)

13.16.6.1.3.1 Offset

Register	Offset
AQHiIdle	4h

13.16.6.1.3.2 Diagram



13.16.6.1.3.3 Fields

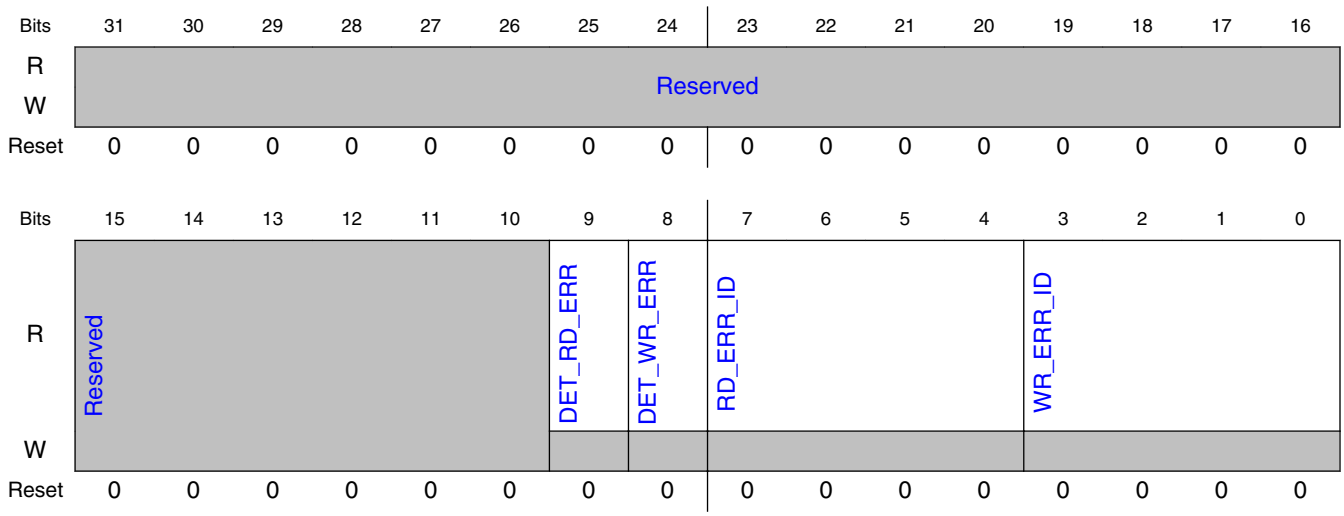
Field	Function
31 AXI_LP	AXI is in low power mode
30-4 —	Reserved
3 IDLE_SH	SH is idle
2-1 —	Reserved
0 IDLE_FE	FE is idle

13.16.6.1.4 AXI Status Register (AQAxiStatus)

13.16.6.1.4.1 Offset

Register	Offset
AQAxiStatus	Ch

13.16.6.1.4.2 Diagram



13.16.6.1.4.3 Fields

Field	Function
31-10 —	Reserved
9 DET_RD_ERR	Detect Read Error 1b - Detect read error
8 DET_WR_ERR	Detect Write Error 1b - Detect write error
7-4 RD_ERR_ID	Read Error ID
3-0 WR_ERR_ID	Write Error ID

13.16.6.1.5 Interrupt Acknowledge Register (AQIntrAcknowledge)

13.16.6.1.5.1 Offset

Register	Offset
AQIntrAcknowledge	10h

13.16.6.1.5.2 Function

Each bit represents a corresponding event being triggered. Reading from this register clears the outstanding interrupt.

13.16.6.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	INTR_VEC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INTR_VEC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.5.4 Fields

Field	Function
31-0 INTR_VEC	Interrupt Vector For each interrupt event, 0=Clear, 1=InterruptActive Bit 31 is AXI_BUS_ERROR, 0 = No Error

13.16.6.1.6 Interrupt Enable Register (AQIntrEnbl)

13.16.6.1.6.1 Offset

Register	Offset
AQIntrEnbl	14h

13.16.6.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INTR_ENBL_VEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INTR_ENBL_VEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.6.3 Fields

Field	Function
31-0 INTR_ENBL_VEC	Interrupt Vector Enable In this register, each bit enables a corresponding event. 0: Disable interrupt 1: Enable Interrupt

13.16.6.1.7 Total Cycles Register (gcTotalCycles)

13.16.6.1.7.1 Offset

Register	Offset
gcTotalCycles	78h

13.16.6.1.7.2 Function

This register is a free running counter. It can be reset by writing 0 to it.

13.16.6.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CYCLES															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CYCLES															
Reset	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.7.4 Fields

Field	Function
31-0 CYCLES	Cycles

13.16.6.1.8 Module Power Level Control Register (gcModulePowerControls)

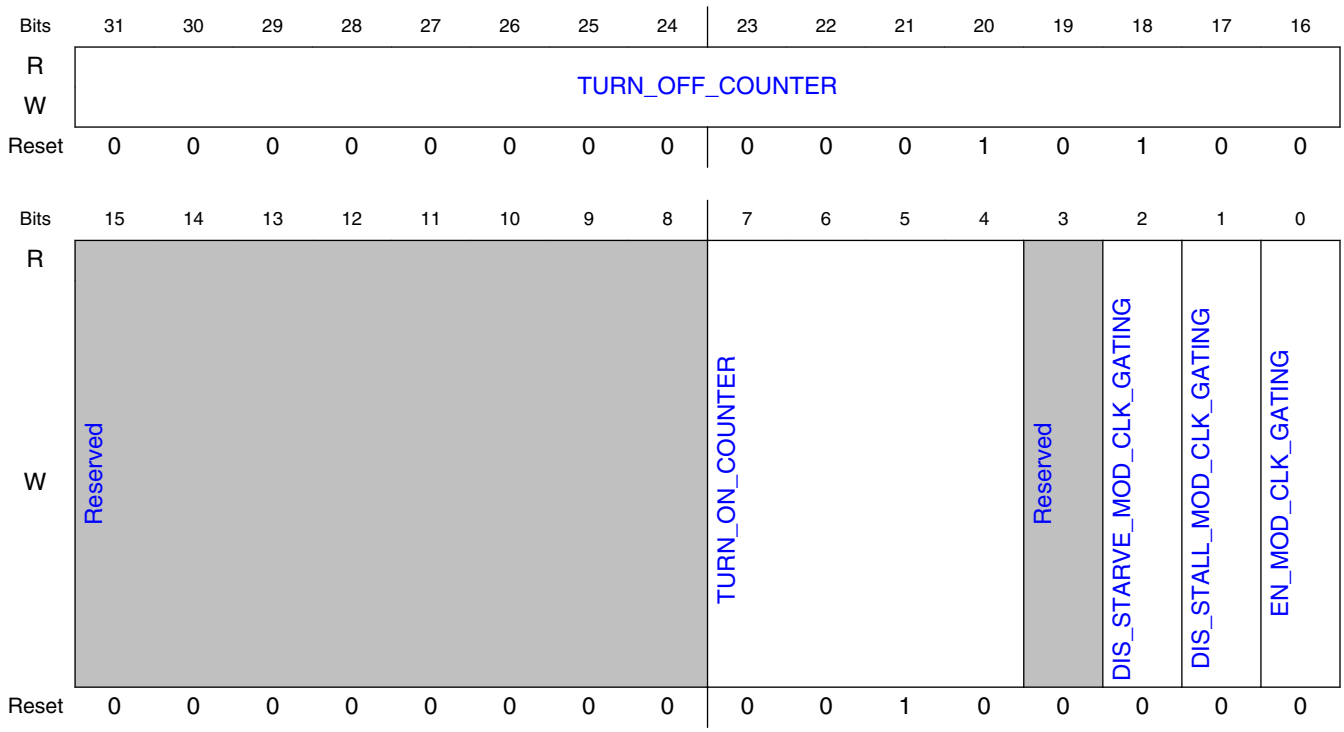
13.16.6.1.8.1 Offset

Register	Offset
gcModulePowerControls	100h

13.16.6.1.8.2 Function

This is a control register for module level power controls

13.16.6.1.8.3 Diagram



13.16.6.1.8.4 Fields

Field	Function
31-16 TURN_OFF_COUNTER	Counter value for clock gating the module if the module is idle for this amount of clock cycles
15-8 —	Reserved
7-4 TURN_ON_COUNTER	Number of clock cycles to wait after turning on the clock
3 —	Reserved
2 DIS_STARVE_MOD_CLK_GATING	Disables module level clock gating for starve/idle condition
1 DIS_STALL_MOD_CLK_GATING	Disables module level clock gating for stall condition
0	Enables module level clock gating

Field	Function
EN_MOD_CLK_GATING	

13.16.6.1.9 Pulse eater Control Register (gcPulseEater)

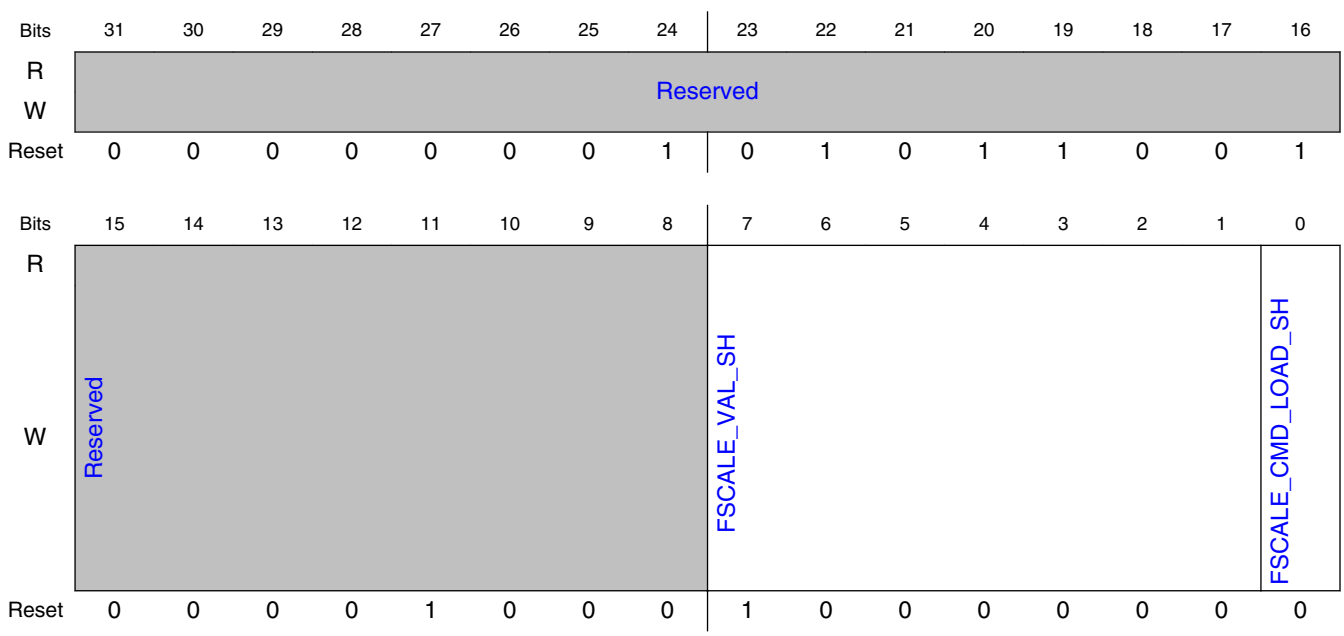
13.16.6.1.9.1 Offset

Register	Offset
gcPulseEater	10Ch

13.16.6.1.9.2 Function

This register shows the Pulse eater parameters

13.16.6.1.9.3 Diagram



13.16.6.1.9.4 Fields

Field	Function
31-8 —	Reserved

Table continues on the next page...

Field	Function
7-1 FSCALE_VAL_SH	Fscale value for shader
0 FSCALE_CMD_LOAD_SH	Fscale_cmd_load for shader

13.16.6.1.10 MMU Control Register (gcregMMUAHBControl)

13.16.6.1.10.1 Offset

Register	Offset
gcregMMUAHBControl	388h

13.16.6.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															MMU
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.10.3 Fields

Field	Function
31-1 —	Reserved
0 MMU	Enable the MMU For security reasons, once the MMU is enabled it cannot be disabled anymore 0b - Disable 1b - Enable

13.16.6.1.11 MMU Table Array Base Lower 32-bit Address Register (gcregMMUAHBTableArrayBaseAddressLow)

13.16.6.1.11.1 Offset

Register	Offset
gcregMMUAHBTableArrayBaseAddressLow	38Ch

13.16.6.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.11.3 Fields

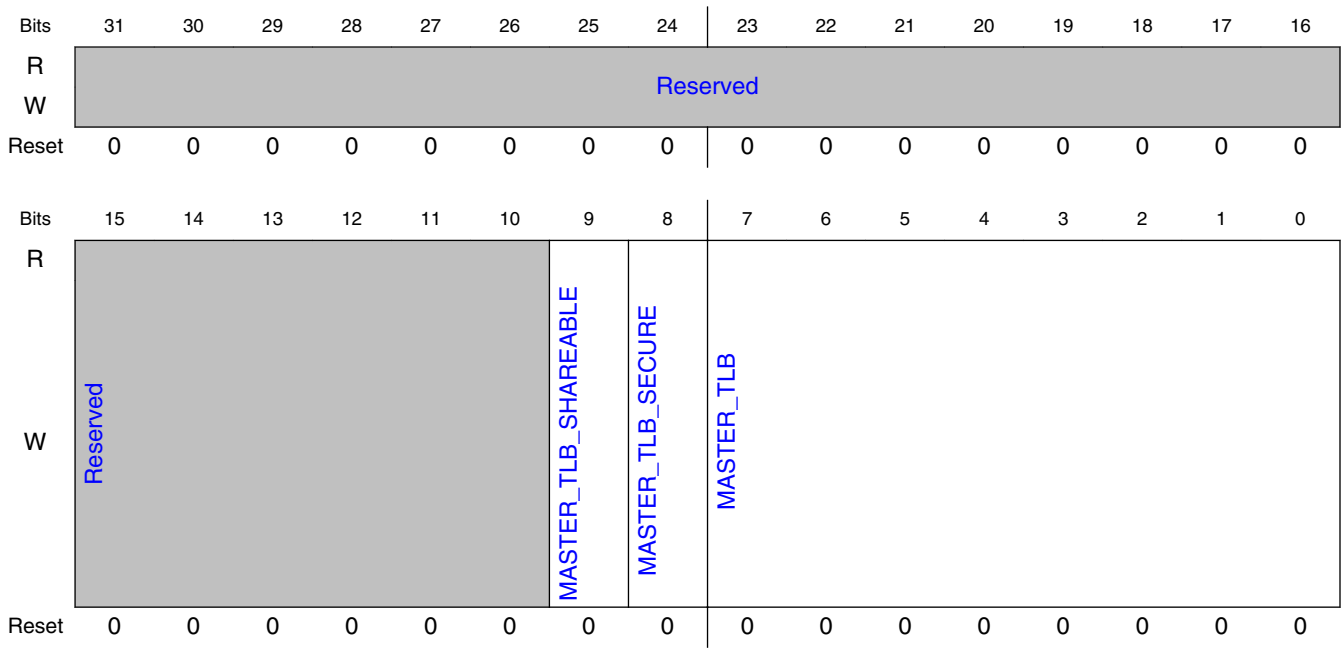
Field	Function
31-0 ADDRESS	Address

13.16.6.1.12 MMU Table Array Base Higher 32-bit Address Register (gcregMMUAHBTableArrayBaseAddressHigh)

13.16.6.1.12.1 Offset

Register	Offset
gcregMMUAHBTableArrayBaseAddressHigh	390h

13.16.6.1.12.2 Diagram



13.16.6.1.12.3 Fields

Field	Function
31-10 —	Reserved
9 MASTER_TLB_SHAREABLE	Bit that defines whether the master TLB address is shareable or not
8 MASTER_TLB_SECURE	Bit that defines whether the master TLB address is secure or not
7-0 MASTER_TLB	Upper 8-bits of the master TLB address to form a true 40-bit address

13.16.6.1.13 MMU Table Array Size Control Register (gcregMMUAHBTableArraySize)

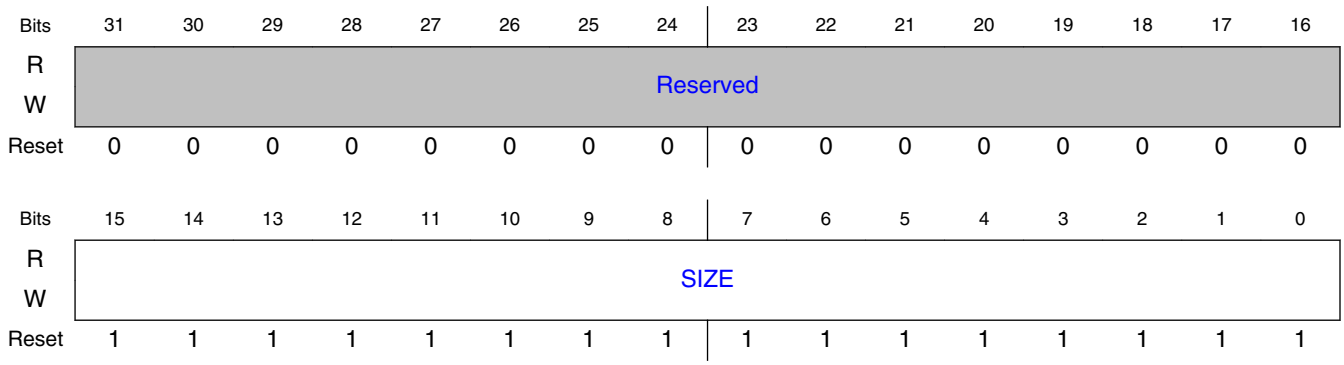
13.16.6.1.13.1 Offset

Register	Offset
gcregMMUAHBTableArraySize	394h

13.16.6.1.13.2 Function

16 bit MMUTableArraySize

13.16.6.1.13.3 Diagram



13.16.6.1.13.4 Fields

Field	Function
31-16 —	Reserved
15-0 SIZE	Size

13.16.6.1.14 MMU NonSecure Address Register (gcregMMUAHBSafeNonSecureAddress)

13.16.6.1.14.1 Offset

Register	Offset
gcregMMUAHBSafeNonSecureAddress	398h

13.16.6.1.14.2 Function

A 64-byte address that will acts as a 'safe' zone. Any address that would cause an exception is routed to this safe zone. Reads will happen and writes will go to this address, but with a write-enable of 0.

This register can only be programmed once after a reset - any attempt to write to this register after the initial write-after-reset will be ignored. This is in NonSecure memory.

13.16.6.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.14.4 Fields

Field	Function
31-0 ADDRESS	Address

13.16.6.1.15 MMU Secure Address Register (gcregMMUAHBSafeSecureAddress)

13.16.6.1.15.1 Offset

Register	Offset
gcregMMUAHBSafeSecureAddress	39Ch

13.16.6.1.15.2 Function

A 64-byte address that will acts as a 'safe' zone. Any address that would cause an exception is routed to this safe zone. Reads will happen and writes will go to this address, but with a write-enable of 0.

This register can only be programmed once after a reset - any attempt to write to this register after the initial write-after-reset will be ignored. This is in Secure memory.

13.16.6.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.15.4 Fields

Field	Function
31-0 ADDRESS	Address

13.16.6.1.16 Command Buffer Control Register (gcregCmdBufferAHBCtrl)

13.16.6.1.16.1 Offset

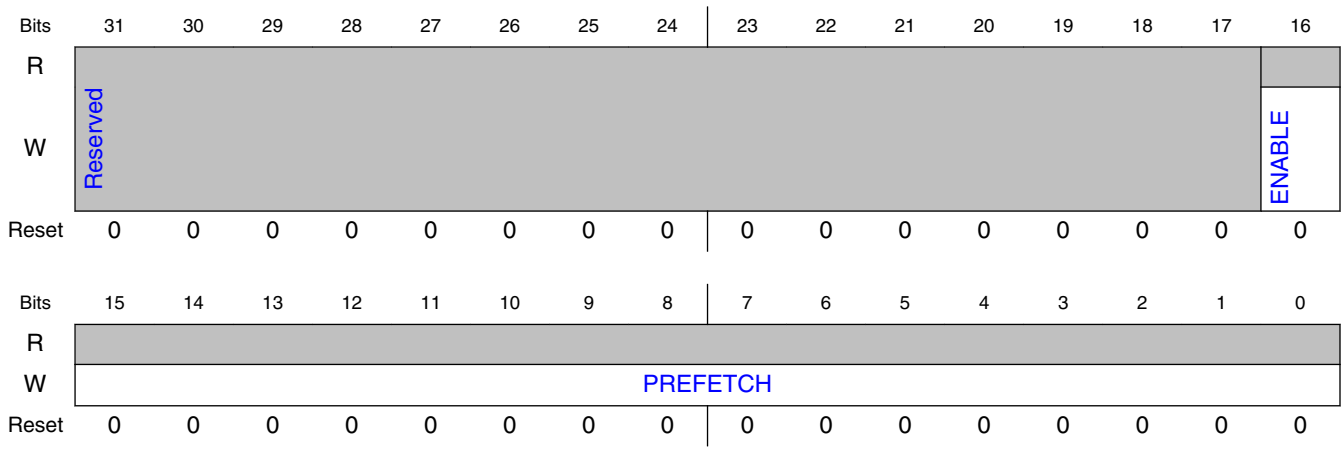
Register	Offset
gcregCmdBufferAHBCtrl	3A4h

13.16.6.1.16.2 Function

NOTE

Since this is a write only register, it has no reset value

13.16.6.1.16.3 Diagram



13.16.6.1.16.4 Fields

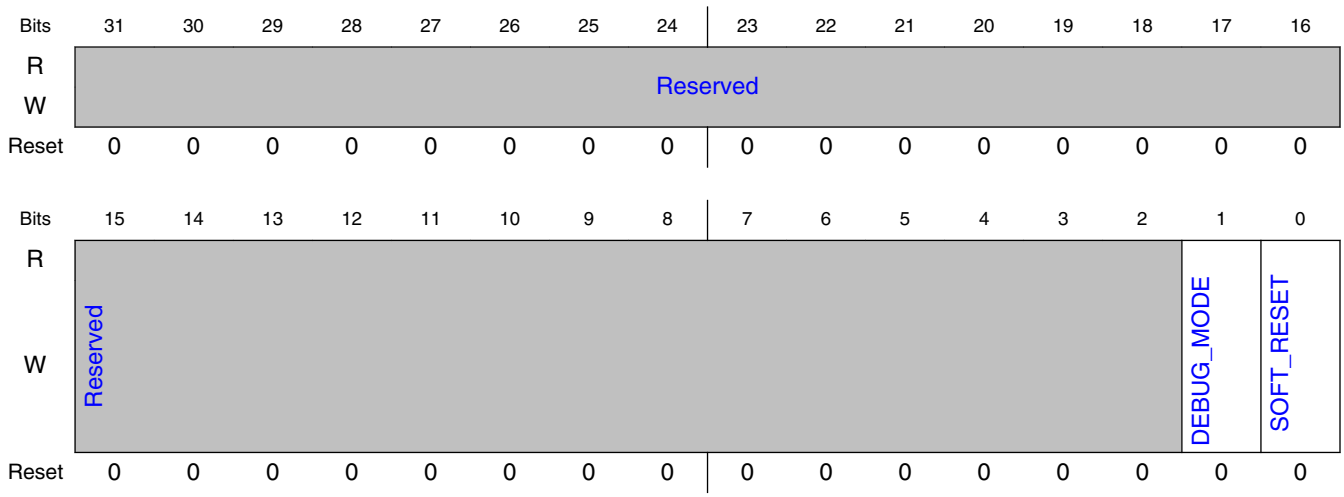
Field	Function
31-17 —	Reserved
16 ENABLE	Enable the command parser 0b - Disable 1b - Enable
15-0 PREFETCH	Prefetch Number of 64-bit words to fetch from the command buffer

13.16.6.1.17 MMU Host Interface Control Register (gcregHiAHBControl)

13.16.6.1.17.1 Offset

Register	Offset
gcregHiAHBControl	3A8h

13.16.6.1.17.2 Diagram



13.16.6.1.17.3 Fields

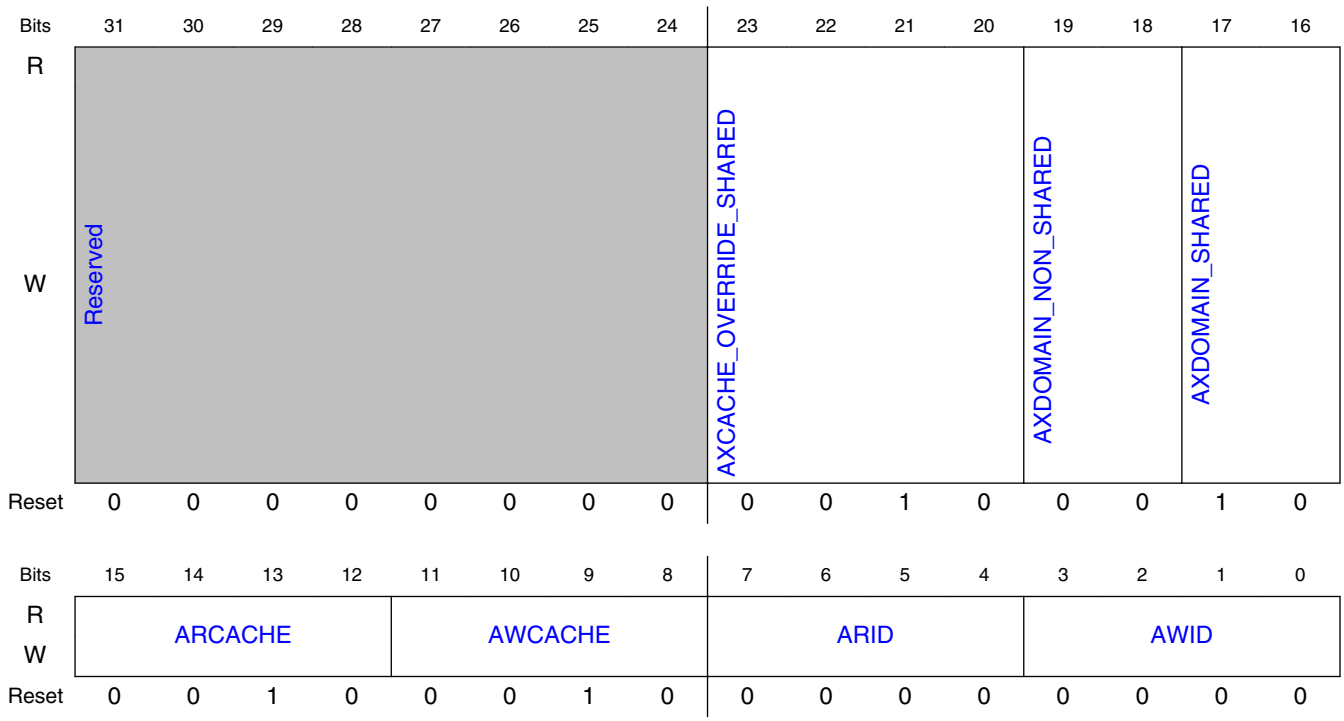
Field	Function
31-2 —	Reserved
1 DEBUG_MODE	Debug Mode Enable debug mode if disabled debug registers return 0xFFFF FFFF 0b - Disable 1b - Enable
0 SOFT_RESET	Soft Reset Soft resets the IP 0b - Disable 1b - Enable

13.16.6.1.18 MMU AXI Configuration Register (gcregAxiAHBConfig)

13.16.6.1.18.1 Offset

Register	Offset
gcregAxiAHBConfig	3ACh

13.16.6.1.18.2 Diagram



13.16.6.1.18.3 Fields

Field	Function
31-24 —	Reserved
23-20 AXCACHE_OVERRIDE_SHARED	Ax Cache value Configure AxCACHE walue for shareable request
19-18 AXDOMAIN_NON_SHARED	Ax Domain value Configure AxDOMAIN walue for non-shareable request
17-16 AXDOMAIN_SHARED	Ax Domain value Configure AxDOMAIN walue for shareable request
15-12 ARCACHE	AR Cache value Set ARCACHE[3:0] value
11-8 AWCACHE	AW Cache value Set AWCACHE[3:0] value
7-4 ARID	

Table continues on the next page...

Field	Function
3-0 AWID	

13.16.6.1.19 Memory Debug Register (AQMemoryDebug)

13.16.6.1.19.1 Offset

Register	Offset
AQMemoryDebug	414h

13.16.6.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								MAX_OUTSTANDING_READS							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.19.3 Fields

Field	Function
31-8 —	Reserved
7-0 MAX_OUTSTANDING_READS	Maximum Outstanding Reads Limits the total number of outstanding read requests

13.16.6.1.20 Register Timing Control Register (AQRegisterTimingControl)

13.16.6.1.20.1 Offset

Register	Offset
AQRegisterTimingControl	42Ch

13.16.6.1.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved									LIGHT_SLEEP	DEEP_SLEEP	POWER_DOWN	FAST_WTC		FAST_RTC	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FOR_RF2P								FOR_RF1P							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.20.3 Fields

Field	Function
31-23 —	Reserved
22 LIGHT_SLEEP	Light sleep Bit to allow SOC to manage sleep for embedded memories
21 DEEP_SLEEP	Deep sleep Bit to allow SOC to manage sleep for embedded memories
20 POWER_DOWN	Power Down Memory
19-18 FAST_WTC	WTC for fast RAM
17-16 FAST_RTC	RTC for fast RAM
15-8 FOR_RF2P	For 2 port RAM
7-0 FOR_RF1P	For 1 port RAM

13.16.6.1.21 Command Buffer Base Address Register (AQCmdBufferAddr)

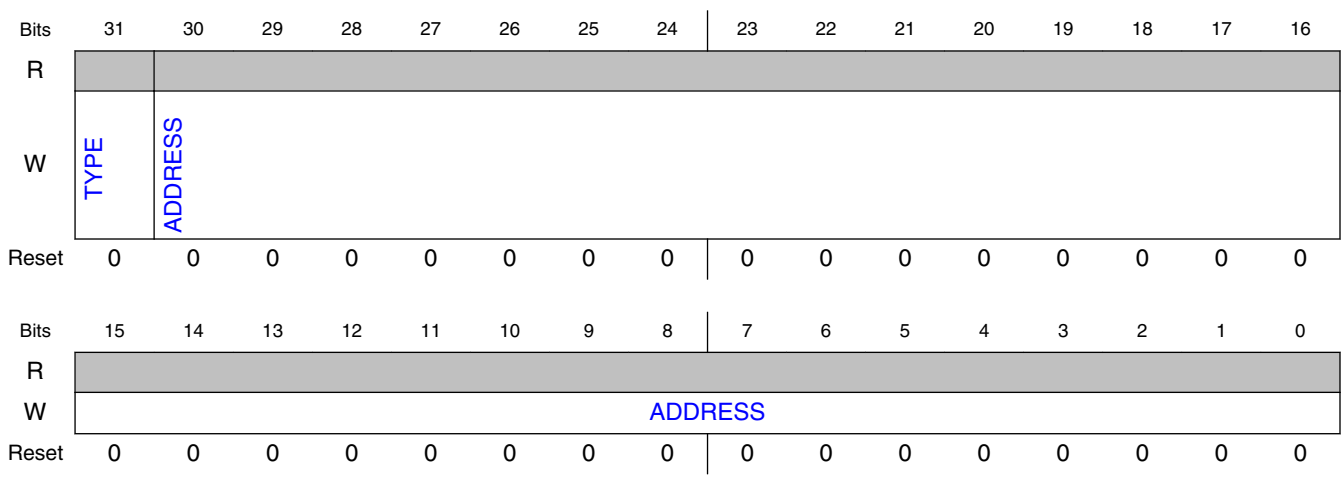
13.16.6.1.21.1 Offset

Register	Offset
AQCmdBufferAddr	654h

13.16.6.1.21.2 Function

The address must be 64-bit aligned, physical address. To check the value of the current fetch address use AQFEDebugCurCmdAdr. Since this is a write-only register, it has no set reset value.

13.16.6.1.21.3 Diagram



13.16.6.1.21.4 Fields

Field	Function
31 TYPE	Type 0b - System 1b - Virtual System
30-0 ADDRESS	Address

13.16.6.1.22 Command Decoder Address Register (AQFEDebugCurCmdAdr)

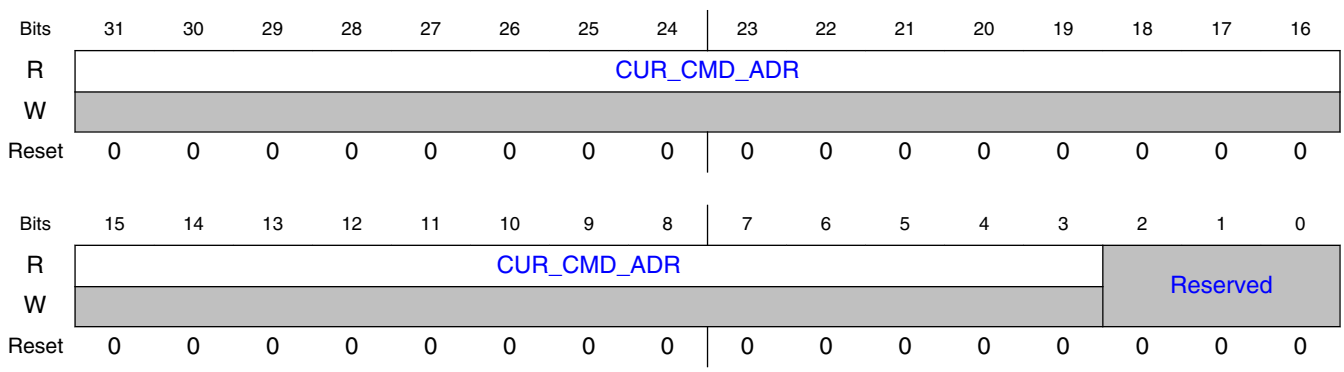
13.16.6.1.22.1 Offset

Register	Offset
AQFEDebugCurCmdAdr	664h

13.16.6.1.22.2 Function

This is a read-only register

13.16.6.1.22.3 Diagram



13.16.6.1.22.4 Fields

Field	Function
31-3 CUR_CMD_ADR	Command decoder Address
2-0 —	Reserved

Chapter 14
Audio

14.1 Audio Overview

14.1.1 Audio Interface

The device supports multiple audio interfaces as listed below:

Table 14-1. Audio Interface Summary

Interface	Function	RX Data Line	TX Data Line	Note
SAI1	External audio	8	8	Access to port is subject to pin mux constraints
SAI2	External audio	4	4	Access to port is subject to pin mux constraints
SAI3	External audio	2	2	Access to port is subject to pin mux constraints
SAI5	External audio	4	4	Access to port is subject to pin mux constraints
SAI6	External audio	1	1	Access to port is subject to pin mux constraints
SAI7	External audio	1	1	Access to port is subject to pin mux constraints
SPDIF1	External audio	1	1	The eARC interface provides SPDIF functionality
PDM	External audio	4		Supports up to 8-Microphones
HDMI Tx	External audio	N/A	0	Supported through HDMI PHY. More info can be found in the HDMI TX Controller and the HDMI TX Parallel Audio Interface (HTX_PAI) sections.
eARC	External audio	1	0	Supported through eARC PHY
ASRC	Internal audio	-	-	No external interface

In addition to the general audio input/output functions, the audio interfaces support the following features:

- SAI1 supports to up to 8 I2S/TDM Tx lanes and 8 I2S/TDM Rx lanes at 768kHz/32-bit
- SAI2/5 supports to up to 8-channels TX (4 lanes) and 8-channels RX (4 lanes) at 768kHz/32-bit
- SAI3 supports up to 4-channels TX (2 lanes) and 4-channels RX (2 lanes) at 768kHz/32-bit
- SAI6 supports to up to 2-channels TX (1 lanes) and 2-channels RX (1 lanes) at 768kHz/32-bit when multiplexed on SAI1, or up to 384kHz/32-bit when multiplexed on Ethernet primary pins
- SAI7 supports to up to 2-channels TX (1 lanes) and 2-channels RX (1 lanes) at 384kHz/32-bit
- SPDIF supports raw capture mode that can save all incoming bits into an audio buffer
- PDM supports up to 8-channels (4 lanes)

14.1.1.1 SAI Master Clock Inputs/Outputs

The MCLK pin on each SAI module can be configured as either an input or an output. When configured as an output, the SAIn_CLK_ROOT from the CCM or the MCLK_OUT from SAIn is routed to the pad output. Note that the SAIn_MCLK_OUT is always derived from the ipg_clk_sai_mclk (MCLK[1]) input. When configured as an input, the external input to the pad will be used as SAIn_MCLK and is routed to SAIn_MCLK_IN, which can be used as master clock for SAI. Below is a diagram showing both input and output options.

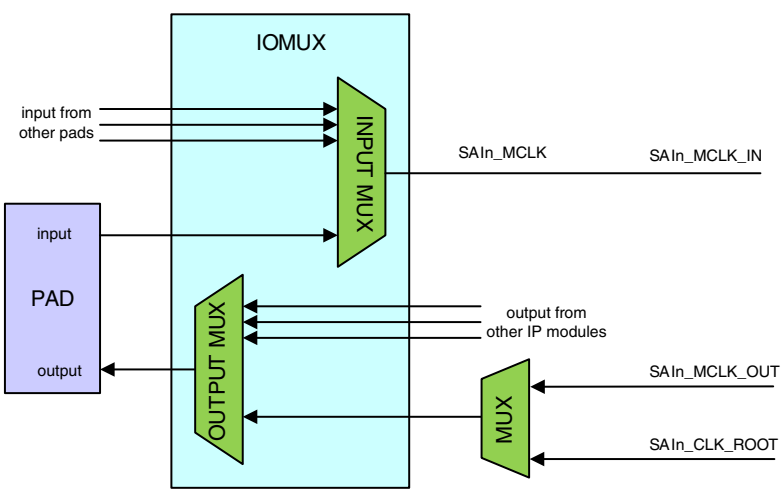


Figure 14-1. SAI MCLK Input and Output

Each SAI module supports up to 3 master clock inputs. The TX and RX sub-modules inside each SAI can independently select one of the clock inputs as their master clock. This allows the TX and RX of one SAI to run from different clock sources. The master clock inputs have the following options:

- SAIn_MCLK[1] can be selected from SAIn_CLK_ROOT from CCM or SAIn_MCLK from IOMUX. This is the most straight-forward clock routing in which SAIn only uses its own clock source from CCM or IO pad
- SAIn_MCLK[2] can be selected from the following clock sources:
  - Any of the SAIn_CLK_ROOT from CCM
  - Any of the SAIn_MCLK from IOMUX
  - Other clock sources from SPDIF
- SAIn_MCLK[3] is connected to a fixed /2 divider of the SAI CLK output of the Fractional PLL

The clock options for master clock on SAI are shown in the diagram below. The master clock options are identical for SAI1, SAI2, SAI3, SAI5,SAI6, and SAI7.

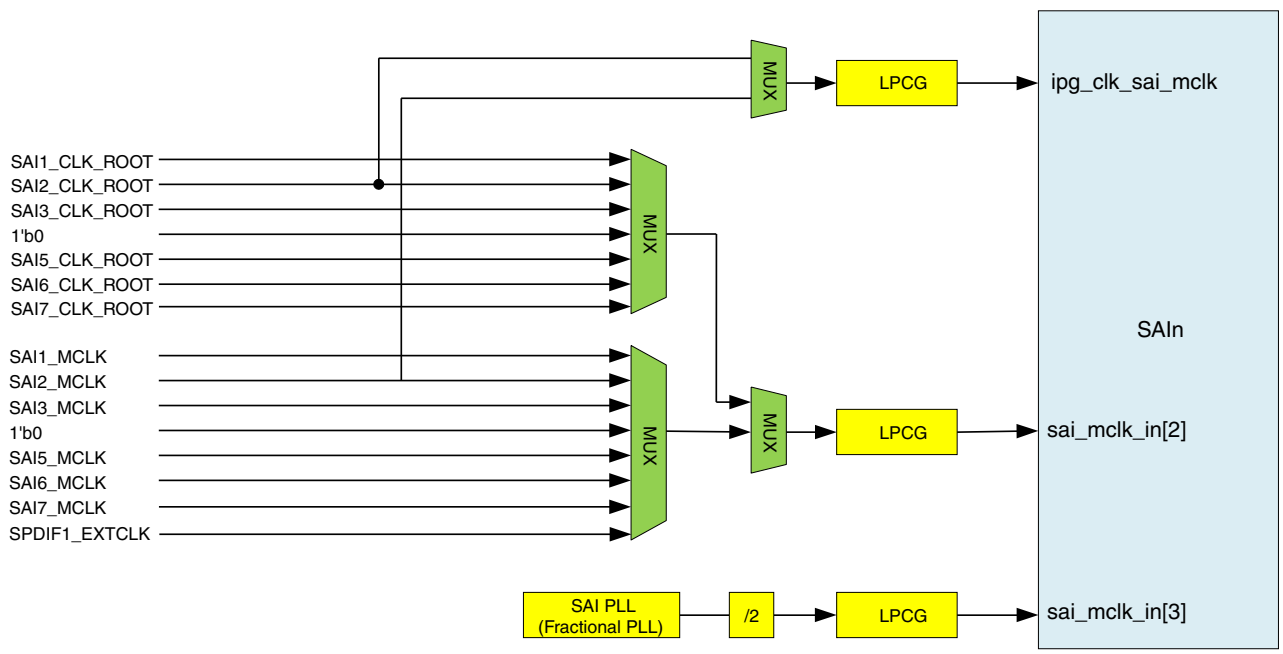


Figure 14-2. SAI Master Clock Options

The configuration of the MUX for master clock is controlled by the AUDIO BLK_CTRL registers. They should be configured before the SAI clock is enabled to avoid glitches on the clock.

14.1.1.2 SDMA Support

All of the SAI and SPDIF instances have SDMA support. In order to meet the audio data rate, two dedicated SDMA modules are used. The DMA requests from all audio interface modules will be routed to both SMDA modules, and both SDMA modules' IPG master port are connected to the SPBA. (SPBA has 3 master ports, one for chip-level IPG bus, two for the SDMAs). With this implementation, the two SDMA modules can be flexibly allocated to all the audio interfaces, based on the needs of audio use cases.

The SDMA2/3 target frequency is 400MHz IPG and 400MHz AHB, always 1:1 mode, to make sure there is enough throughput for all the audio use cases.

Table 14-2. SDMA Assignment for Audio Interface

SDMA	Audio Interface	TX and RX Data Lines
SDMA2/3 400MHz AHB, 400MHz IPG	SAI1	8 TX + 8 RX
	SAI2	4 TX + 4 RX
	SAI3	2 TX + 2 RX
	Audio Transceiver (eARC)	TX + RX
	PDM	4 RX
	SAI5	4 TX + 4 RX
	SAI6	1 TX + 1 RX
	SAI7	1 TX + 1 RX
	ASRC	-

14.1.1.3 eDMA Support

This chip also supports one eDMA for memory-to-memory copies (no DMA requests connected from any peripheral). It is generally controlled by the Audio DSP.

14.1.1.4 ASRC

The ASRC can process 4 groups of audio channels with an independent time-base, simultaneously. A group of channels with the same time-base (or resampling ratio) is referred to as a context. Each context has independent processing pipelines. Contexts can be configured to start and stop at any time without affecting the processing of other contexts. The ASRC supports up to 32 audio channels. The 32 audio channels can either be assigned to a single context or spread across multiple contexts.

The ASRC supports the following features:

- Simultaneous processing of up to 32 audio channels
- 4 independent contexts – groups of channels with an independent time base

- Fully independent and concurrent control of all 4 contexts
- Programmable filter characteristics for each context
- 32-bit floating and 32, 24, 20 and 16-bit fixed point audio sample support
- 8 kHz to 384 kHz sample rate
- 1/16 to 16x sample rate conversion ratio
- Software control of fine conversion ratio

14.1.2 Audio DSP (HiFi 4)

The HiFi 4 Audio Engine is a highly optimized audio processor geared for efficient execution of audio and voice codecs and pre- and post-processing modules.

See the Audio DSP (HiFi 4 DSP) section for more details.

NOTE

The Hifi4 DSP is integrated into the audio subsystem, which includes the AUDIO BLK_CTRL, MICFIL, SAI, ASRC and eARC instances. The subsystem shares one reset (controlled by the SRC), and the Hifi4 DSP doesn't have an individual reset. If the Hifi4 needs to be reset, then the entire audio subsystem must be reset.

14.2 AUDIO BLK_CTRL

14.2.1 Overview

The AUDIOMIX domain contains control and status registers known as Audio Block Control (AUDIO BLK_CTRL). These registers have miscellaneous top-level controls for peripherals within the AUDIOMIX domain. Some of these controls includes clocking, interrupt, and reset sources. Please see the clock control chapter and interrupt chapter for more information regarding the clock input sources and IRQ event mapping to AUDIOMIX.

14.2.1.1 Block Diagram

The AUDIO BLK_CTRL interface connects to the AUDIOMIX domain peripherals. A block diagram illustrating the AUDIOMIX domain is shown below.

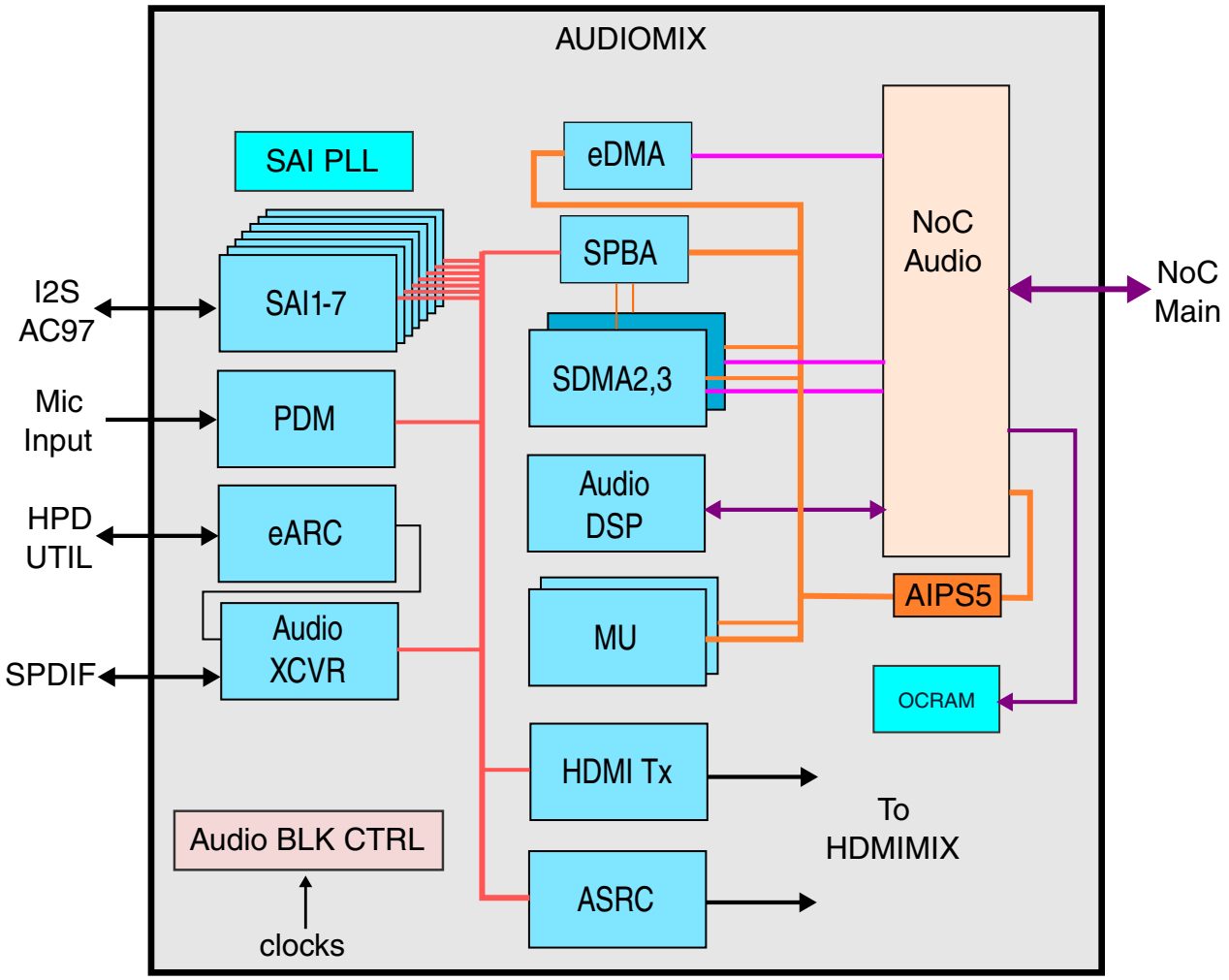


Figure 14-3. AUDIOMIX Block Diagram

14.2.2 Clocks and Resets

The Audio BLK_CTRL contains clock distribution and gating controls, as well as reset handling to several of the AUDIOMIX peripherals. The BLK_CTRL also handles the MCLK routing for SAI. A figure illustrating the supporting MCLK connections is shown below. Please see related clock and reset controls in the BLK_CTRL register description. For more information about the input clocks, system-level resets, and IRQ to the domain, please see the related block guide and system-level assignments in the control blocks for clocks, reset, and IRQ assignments.

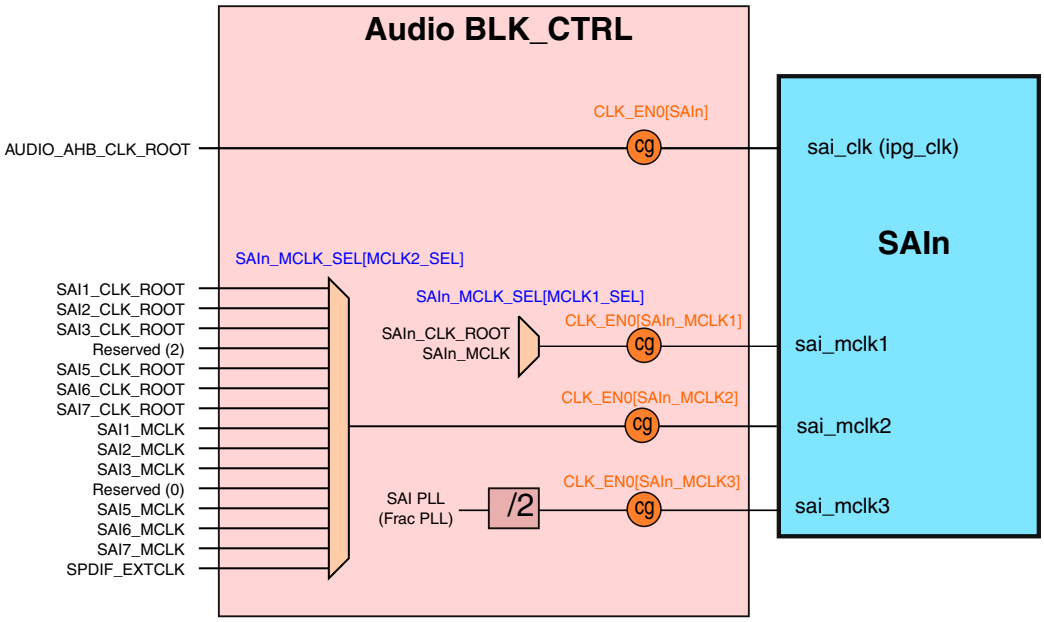


Figure 14-4. Audio BLK_CTRL MCLK Generation

14.2.3 Memory Map and register definition

This section includes the AUDIO BLK_CTRL module memory map and detailed descriptions of all registers.

14.2.3.1 register descriptions

14.2.3.1.1 memory map

AUDIO_BLK_CTRL base address: 30E2_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	IP Clock Enable Control Register 0 (CLKEN0)	32	RW	FF77_7777h
4h	IP Clock Enable Control Register 1 (CLKEN1)	32	RW	0000_003Fh
100h	AudioDSP EXPSTATE Register (AudioDSP_REG0)	32	RO	0000_0000h
104h	AudioDSP IMPWIRE Register (AudioDSP_REG1)	32	RW	0000_0000h
108h	AudioDSP XOCDMODE Register (AudioDSP_REG2)	32	RW	0000_0020h
10Ch	AudioDSP PID Register (AudioDSP_REG3)	32	RW	0000_0000h
200h	EARC Control Register (EARC)	32	RW	0000_0003h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
300h	SAI1 MCLK SELECT Register (SAI1_MCLK_SEL)	32	RW	0000_0000h
304h	SAI2 MCLK SELECT Register (SAI2_MCLK_SEL)	32	RW	0000_0000h
308h	SAI3 MCLK SELECT Register (SAI3_MCLK_SEL)	32	RW	0000_0000h
30Ch	SAI5 MCLK SELECT Register (SAI5_MCLK_SEL)	32	RW	0000_0000h
310h	SAI6 MCLK SELECT Register (SAI6_MCLK_SEL)	32	RW	0000_0000h
314h	SAI7 MCLK SELECT Register (SAI7_MCLK_SEL)	32	RW	0000_0000h
318h	PDM Root Clock Select Register (PDM_CLK)	32	RW	0000_0000h
400h	SAI PLL General control Register (SAI_PLL_GNRL_CTL)	32	RW	0000_2010h
404h	SAI PLL Frequency Divider control Register (SAI_PLL_FDIV_CTL0)	32	RW	0020_C046h
408h	SAI PLL DSM value Register (SAI_PLL_FDIV_CTL1)	32	RW	0000_49BBh
40Ch	SAI PLL SSCG control Register (SAI_PLL_SSCG_CTL)	32	RW	0000_0000h
410h	SAI PLL SSCG control Register (SAI_PLL_MNIT_CTL)	32	RW	0000_0103h
500h	AUDIOMIX Extra Addr Bits Register (AUDIO_EXT_ADDR)	32	RW	0000_0000h
504h	IPG Low Power Control Register (IPG_LP_CTRL)	32	RW	0000_0000h
508h	AUDIOMIX AXI LIMIT CTRL Register (AUDIO_AXI_LIMIT)	32	RW	0000_0000h

14.2.3.1.2 IP Clock Enable Control Register 0 (CLKEN0)

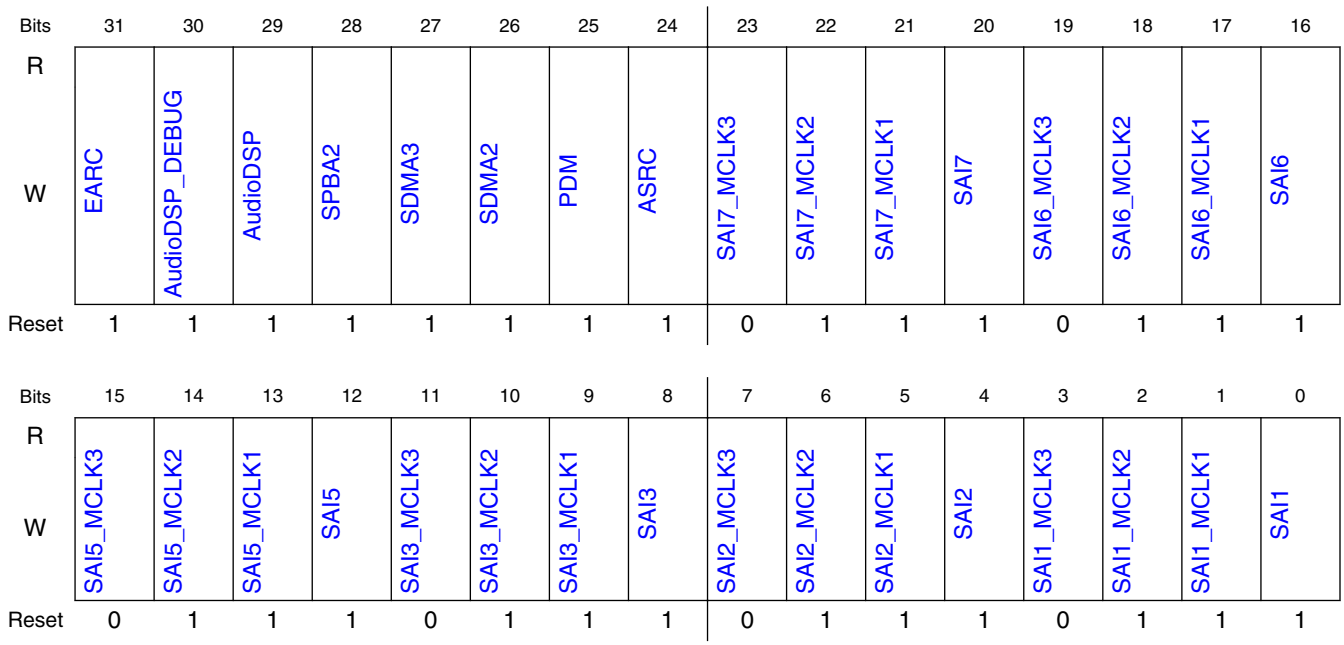
14.2.3.1.2.1 Offset

Register	Offset
CLKEN0	0h

14.2.3.1.2.2 Function

.

14.2.3.1.2.3 Diagram



14.2.3.1.2.4 Fields

Field	Function
31 EARC	EARC clock enable EARC clock enable control bit.
30 AudioDSP_DEBUG	AudioDSP DEBUG clock enable AudioDSP DEBUG clock enable control bit.
29 AudioDSP	AudioDSP core clock enable HFIF4 core clock enable control bit.
28 SPBA2	SPBA2 clock enable SPBA2 clock enable control bit.
27 SDMA3	SDMA3 clock enable SDMA3 clock enable control bit.
26 SDMA2	SDMA2 clock enable SDMA2 clock enable control bit.
25 PDM	PDM clock enable PDM clock enable control bit.
24 ASRC	ASRC clock enable ASRC clock enable control bit.
23 SAI7_MCLK3	SAI7 mclk3 clock enable SAI7 mclk3 enable control bit.

Table continues on the next page...

Field	Function
22	SAI7 mclk2 clock enable
SAI7_MCLK2	SAI7 mclk2 enable control bit.
21	SAI7 mclk1 clock enable
SAI7_MCLK1	SAI7 mclk1 enable control bit.
20	SAI7 clock enable
SAI7	SAI7 clock enable control bit. 0b - SAI7 sai clock disable 1b - SAI7 sai clock enable
19	SAI6 mclk3 clock enable
SAI6_MCLK3	SAI6 mclk3 enable control bit.
18	SAI6 mclk2 clock enable
SAI6_MCLK2	SAI6 mclk2 enable control bit.
17	SAI6 mclk1 clock enable
SAI6_MCLK1	SAI6 mclk1 enable control bit.
16	SAI6 clock enable
SAI6	SAI6 clock enable control bit. 0b - SAI6 IPG clock disable 1b - SAI6 IPG clock enable
15	SAI5 mclk3 clock enable
SAI5_MCLK3	SAI5 mclk3 enable control bit.
14	SAI5 mclk2 clock enable
SAI5_MCLK2	SAI5 mclk2 enable control bit.
13	SAI5 mclk1 clock enable
SAI5_MCLK1	SAI5 mclk1 enable control bit.
12	SAI5 clock enable
SAI5	SAI5 clock enable control bit. 0b - SAI5 sai clock disable 1b - SAI5 sai clock enable
11	SAI3 mclk3 clock enable
SAI3_MCLK3	SAI3 mclk3 enable control bit.
10	SAI3 mclk2 clock enable
SAI3_MCLK2	SAI3 mclk2 enable control bit.
9	SAI3 mclk1 clock enable
SAI3_MCLK1	SAI3 mclk1 enable control bit.
8	SAI3 clock enable
SAI3	SAI3 clock enable control bit. 0b - SAI3 sai clock disable 1b - SAI3 sai clock enable
7	SAI2 mclk3 clock enable
SAI2_MCLK3	SAI2 mclk3 enable control bit.
6	SAI2 mclk2 clock enable
SAI2_MCLK2	SAI2 mclk2 enable control bit.

Table continues on the next page...

Field	Function
5 SAI2_MCLK1	SAI2 mclk1 clock enable SAI2 mclk1 enable control bit.
4 SAI2	SAI2 clock enable SAI2 clock enable control bit. 0b - SAI2 sai clock disable 1b - SAI2 sai clock enable
3 SAI1_MCLK3	SAI1 mclk3 clock enable SAI1 mclk3 enable control bit.
2 SAI1_MCLK2	SAI1 mclk2 clock enable SAI1 mclk2 enable control bit.
1 SAI1_MCLK1	SAI1 mclk1 clock enable SAI1 mclk1 enable control bit.
0 SAI1	SAI1 clock enable SAI1 clock enable control bit. 0b - SAI1 sai clock disable 1b - SAI1 sai clock enable

14.2.3.1.3 IP Clock Enable Control Register 1 (CLKEN1)

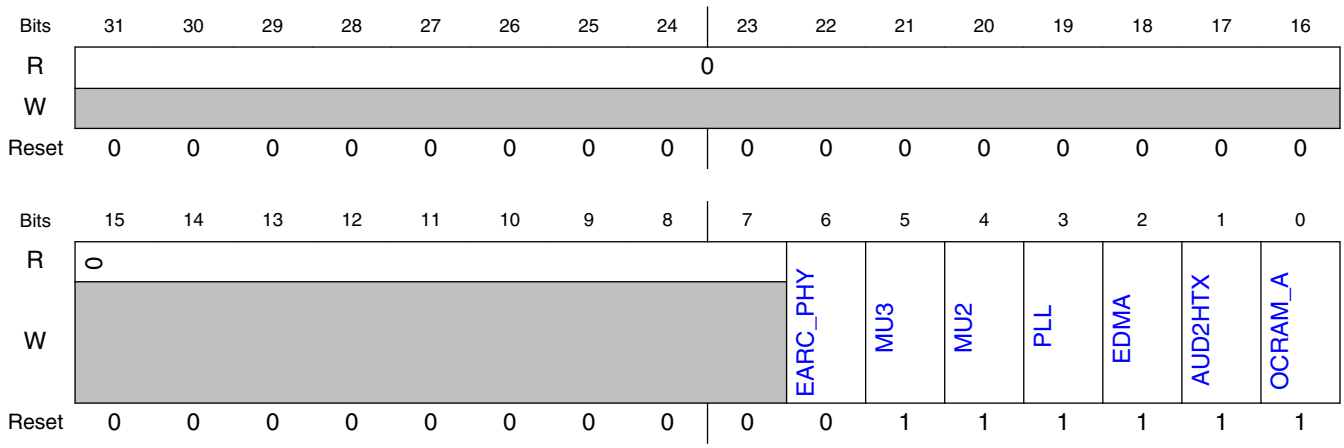
14.2.3.1.3.1 Offset

Register	Offset
CLKEN1	4h

14.2.3.1.3.2 Function

.

14.2.3.1.3.3 Diagram



14.2.3.1.3.4 Fields

Field	Function
31-7 —	Reserved
6 EARC_PHY	EARC PHY audio ss clock enable EARC PHY audio ss clock enable control bit.
5 MU3	MU3 clock enable MU3 clock enable control bit.
4 MU2	MU2 clock enable MU2 clock enable control bit.
3 PLL	PLL clock enable PLL clock enable control bit.
2 EDMA	EDMA clock enable EDMA clock enable control bit.
1 AUD2HTX	AUDIO LINK MASTER clock enable AUDIO LINK MASTER clock enable control bit.
0 OCRAM_A	OCRAM_A clock enable OCRAM_A clock enable control bit.

14.2.3.1.4 AudioDSP EXPSTATE Register (AudioDSP_REG0)

14.2.3.1.4.1 Offset

Register	Offset
AudioDSP_REG0	100h

14.2.3.1.4.2 Function

.

14.2.3.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EXPSTATE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EXPSTATE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.3.1.4.4 Fields

Field	Function
31-0 EXPSTATE	TIE_EXPSTATE output port of the AudioDSP Readonly

14.2.3.1.5 AudioDSP IMPWIRE Register (AudioDSP_REG1)

14.2.3.1.5.1 Offset

Register	Offset
AudioDSP_REG1	104h

14.2.3.1.5.2 Function

.

14.2.3.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IMPWIRE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMPWIRE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.3.1.5.4 Fields

Field	Function
31-0 IMPWIRE	TIE_IMPWIRE input port of the AudioDSP

14.2.3.1.6 AudioDSP XOCDMODE Register (AudioDSP_REG2)

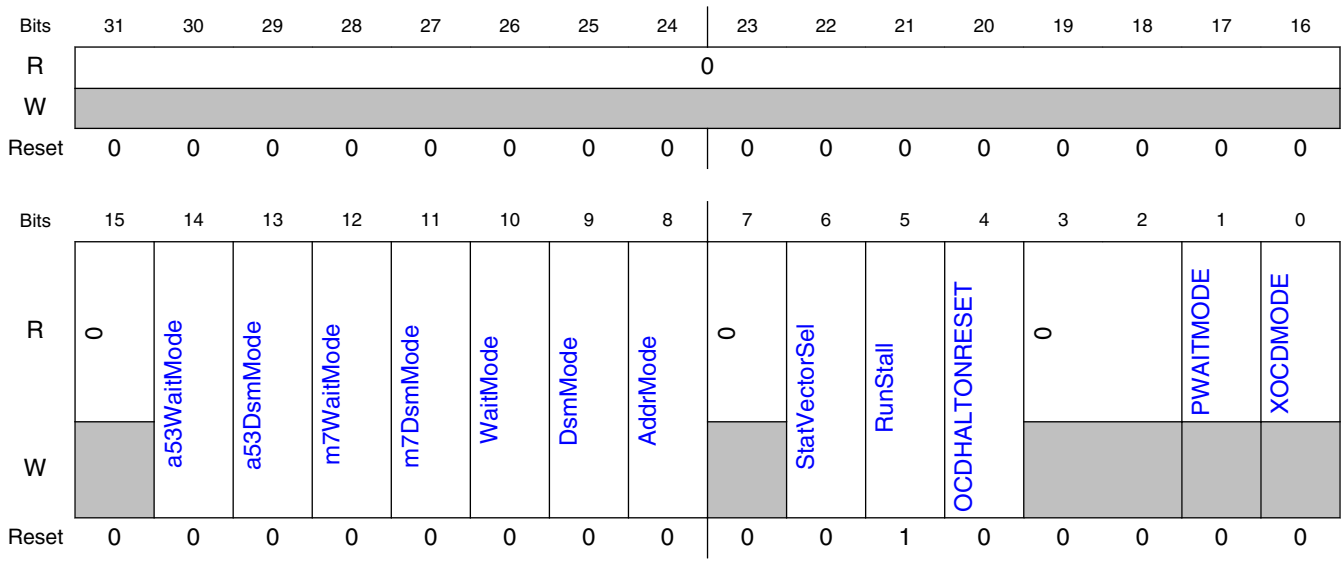
14.2.3.1.6.1 Offset

Register	Offset
AudioDSP_REG2	108h

14.2.3.1.6.2 Function

.

14.2.3.1.6.3 Diagram



14.2.3.1.6.4 Fields

Field	Function
31-15 —	Reserved
14 a53WaitMode	CA53 in Wait Mode for MU2 0b - not in wait mode 1b - in wait mode
13 a53DsmMode	CA53 in DSM Mode for MU2 0b - not in DSM mode 1b - in DSM mode
12 m7WaitMode	M7 in Wait Mode for MU3 0b - not in wait mode 1b - in wait mode
11 m7DsmMode	M7 in DSM Mode for MU3 0b - not in DSM mode 1b - in DSM mode
10 WaitMode	AudioDSP in Wait Mode for MU 0b - not in wait mode 1b - in wait mode
9 DsmMode	AudioDSP in DSM Mode for MU 0b - not in DSM mode 1b - in DSM mode
8 AddrMode	AudioDSP Addr Mode(Alias) Register. Refer to Chip memory map for details.
7 —	Reserved
6	Selects between one of two stationary vector bases

Table continues on the next page...

Field	Function
StatVectorSel	0b - default 1b - alternative
5 RunStall	AudioDSP RunStall control bit. 1b - stalls the processor
4 OCDHALTONRESET	AudioDSP enters OCDHaltMode if this signal is samped asserted on reset.
3-2 —	Reserved
1 PWAITMODE	Indicates that the AudioDSP is in sleep mode. The processor asserts this signal when it has executed a WAITI instruction and is waiting for an interrupt. Readonly.
0 XOCDMODE	Indicates that the AudioDSP is in OCD halt mode Readonly.

14.2.3.1.7 AudioDSP PID Register (AudioDSP_REG3)

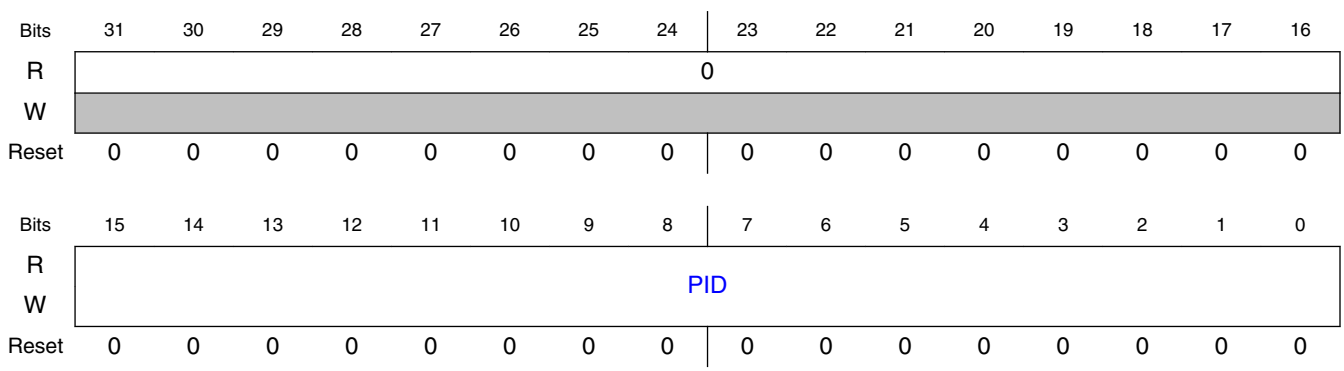
14.2.3.1.7.1 Offset

Register	Offset
AudioDSP_REG3	10Ch

14.2.3.1.7.2 Function

.

14.2.3.1.7.3 Diagram



14.2.3.1.7.4 Fields

Field	Function
31-16 —	Reserved
15-0 PID	AudioDSP PID Register. Input to the AudioDSP, latched at reset into the low-order bits of the PRID(processor ID) special register.

14.2.3.1.8 EARC Control Register (EARC)

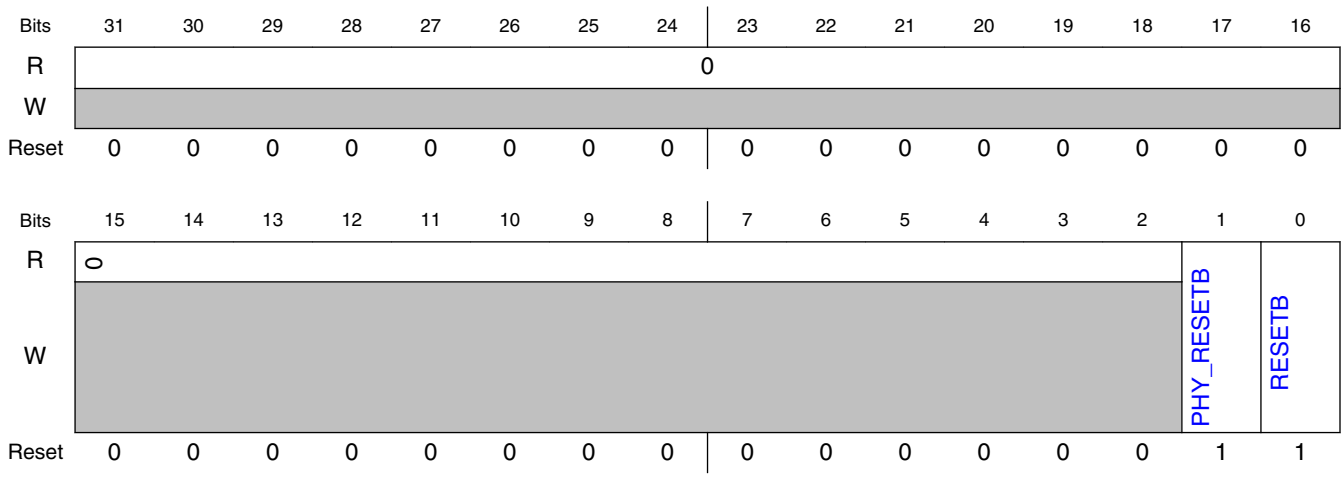
14.2.3.1.8.1 Offset

Register	Offset
EARC	200h

14.2.3.1.8.2 Function

.

14.2.3.1.8.3 Diagram



14.2.3.1.8.4 Fields

Field	Function
31-2	Reserved

Table continues on the next page...

Field	Function
—	
1 PHY_RESETB	Earc PHY Software Reset. 0b - provide a software reset for EARC PHY 1b - return from reset
0 RESETB	Earc Software Reset. 0b - provide a software reset for EARC controller 1b - return from reset

14.2.3.1.9 SAI1 MCLK SELECT Register (SAI1_MCLK_SEL)

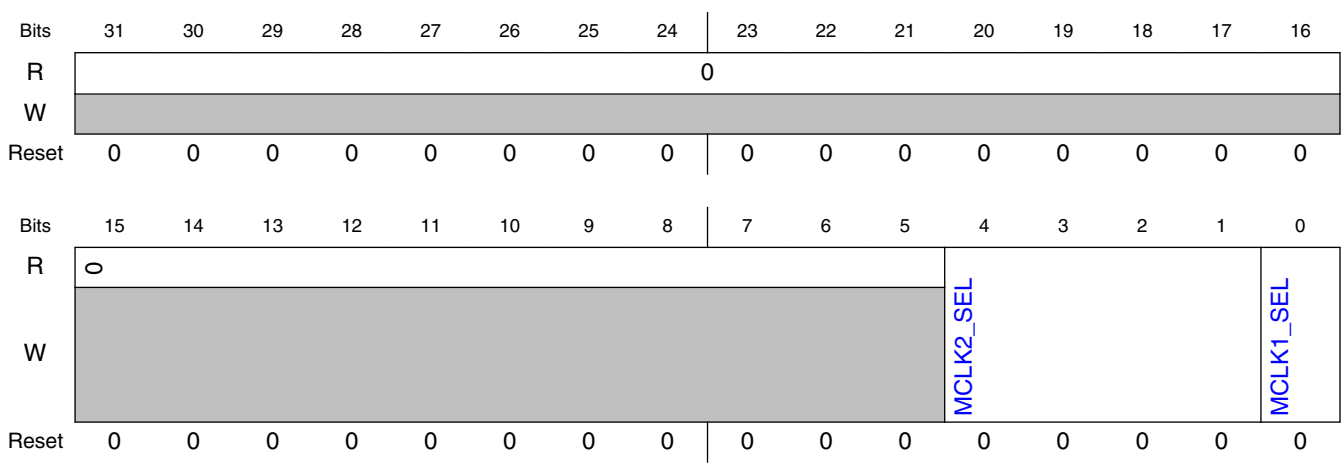
14.2.3.1.9.1 Offset

Register	Offset
SAI1_MCLK_SEL	300h

14.2.3.1.9.2 Function

SAI1 MCLK SELECT Register.

14.2.3.1.9.3 Diagram



14.2.3.1.9.4 Fields

Field	Function
31-5 —	Reserved

Table continues on the next page...

Field	Function
4-1 MCLK2_SEL	MCLK2 Select Register 0000b - SAI1_CLK_ROOT is selected 0001b - SAI2_CLK_ROOT is selected 0010b - SAI3_CLK_ROOT is selected 0011b - Reserved, MCLK2 is 0 0100b - SAI5_CLK_ROOT is selected 0101b - SAI6_CLK_ROOT is selected 0110b - SAI7_CLK_ROOT is selected 0111b - SAI1.MCLK is selected 1000b - SAI2.MCLK is selected 1001b - SAI3.MCLK is selected 1010b - Reserved, MCLK2 is 0 1011b - SAI5.MCLK is selected 1100b - SAI6.MCLK is selected 1101b - SAI7.MCLK is selected 1110b - SPDIF.ETXCLK is selected
0 MCLK1_SEL	MCLK1 Select Register 0b - SAI1_CLK_ROOT is selected 1b - SAI1.MCLK is selected

14.2.3.1.10 SAI2 MCLK SELECT Register (SAI2_MCLK_SEL)

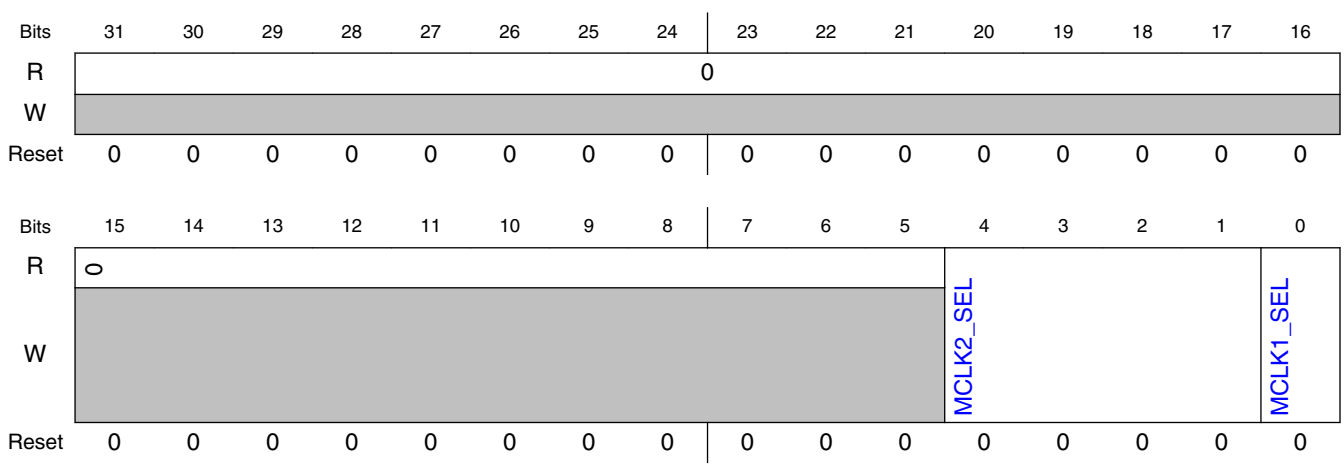
14.2.3.1.10.1 Offset

Register	Offset
SAI2_MCLK_SEL	304h

14.2.3.1.10.2 Function

SAI2 MCLK SELECT Register.

14.2.3.1.10.3 Diagram



14.2.3.1.10.4 Fields

Field	Function
31-5 —	Reserved
4-1 MCLK2_SEL	MCLK2 Select Register 0000b - SAI1_CLK_ROOT is selected 0001b - SAI2_CLK_ROOT is selected 0010b - SAI3_CLK_ROOT is selected 0011b - Reserved, MCLK2 is 0 0100b - SAI5_CLK_ROOT is selected 0101b - SAI6_CLK_ROOT is selected 0110b - SAI7_CLK_ROOT is selected 0111b - SAI1.MCLK is selected 1000b - SAI2.MCLK is selected 1001b - SAI3.MCLK is selected 1010b - Reserved, MCLK is 0 1011b - SAI5.MCLK is selected 1100b - SAI6.MCLK is selected 1101b - SAI7.MCLK is selected 1110b - SPDIF.ETXCLK is selected
0 MCLK1_SEL	MCLK1 Select Register 0b - SAI2_CLK_ROOT is selected 1b - SAI2.MCLK is selected

14.2.3.1.11 SAI3 MCLK SELECT Register (SAI3_MCLK_SEL)

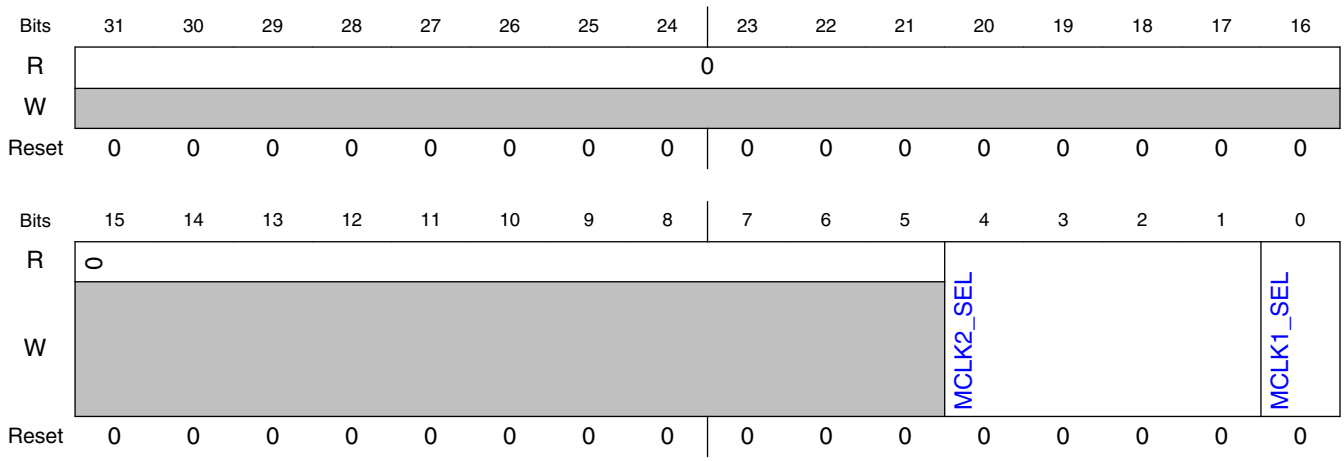
14.2.3.1.11.1 Offset

Register	Offset
SAI3_MCLK_SEL	308h

14.2.3.1.11.2 Function

SAI3 MCLK SELECT Register.

14.2.3.1.11.3 Diagram



14.2.3.1.11.4 Fields

Field	Function
31-5 —	Reserved
4-1 MCLK2_SEL	MCLK2 Select Register 0000b - SAI1_CLK_ROOT is selected 0001b - SAI2_CLK_ROOT is selected 0010b - SAI3_CLK_ROOT is selected 0011b - Reserved, MCLK2 is 0 0100b - SAI5_CLK_ROOT is selected 0101b - SAI6_CLK_ROOT is selected 0110b - SAI7_CLK_ROOT is selected 0111b - SAI1.MCLK is selected 1000b - SAI2.MCLK is selected 1001b - SAI3.MCLK is selected 1010b - Reserved, MCLK is 0 1011b - SAI5.MCLK is selected 1100b - SAI6.MCLK is selected 1101b - SAI7.MCLK is selected 1110b - SPDIF.ETXCLK is selected
0 MCLK1_SEL	MCLK1 Select Register 0b - SAI3_CLK_ROOT is selected 1b - SAI3.MCLK is selected

14.2.3.1.12 SAI5 MCLK SELECT Register (SAI5_MCLK_SEL)

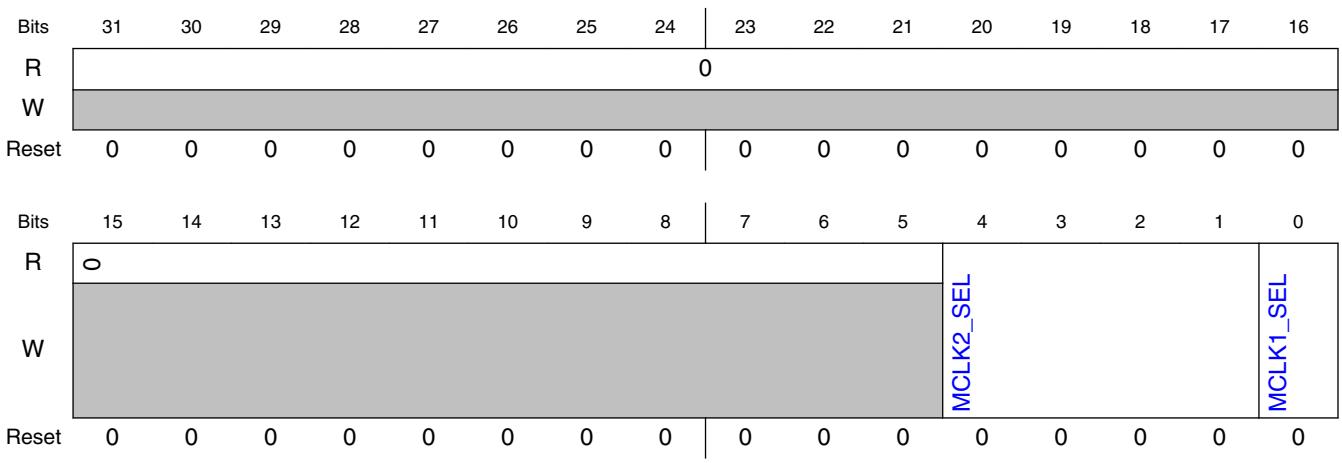
14.2.3.1.12.1 Offset

Register	Offset
SAI5_MCLK_SEL	30Ch

14.2.3.1.12.2 Function

SAI5 MCLK SELECT Register.

14.2.3.1.12.3 Diagram



14.2.3.1.12.4 Fields

Field	Function
31-5 —	Reserved
4-1 MCLK2_SEL	MCLK2 Select Register 0000b - SAI1_CLK_ROOT is selected 0001b - SAI2_CLK_ROOT is selected 0010b - SAI3_CLK_ROOT is selected 0011b - Reserved, MCLK2 is 0 0100b - SAI5_CLK_ROOT is selected 0101b - SAI6_CLK_ROOT is selected 0110b - SAI7_CLK_ROOT is selected 0111b - SAI1.MCLK is selected 1000b - SAI2.MCLK is selected 1001b - SAI3.MCLK is selected 1010b - Reserved, MCLK is 0 1011b - SAI5.MCLK is selected 1100b - SAI6.MCLK is selected 1101b - SAI7.MCLK is selected 1110b - SPDIF.ETXCLK is selected

Table continues on the next page...

Field	Function
0 MCLK1_SEL	MCLK1 Select Register 0b - SAI5_CLK_ROOT is selected 1b - SAI5.MCLK is selected

14.2.3.1.13 SAI6 MCLK SELECT Register (SAI6_MCLK_SEL)

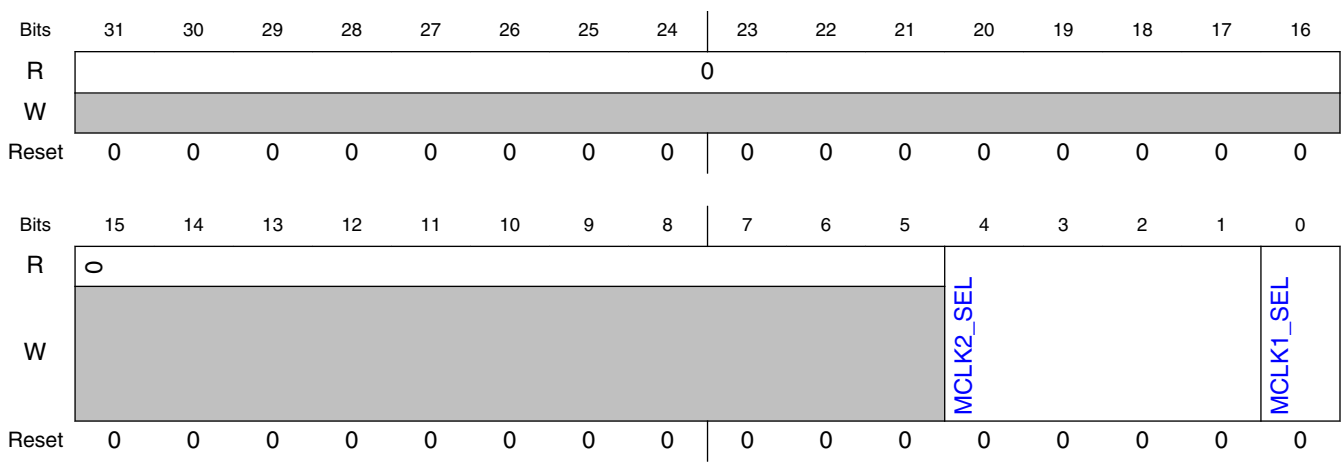
14.2.3.1.13.1 Offset

Register	Offset
SAI6_MCLK_SEL	310h

14.2.3.1.13.2 Function

SAI6 MCLK SELECT Register.

14.2.3.1.13.3 Diagram



14.2.3.1.13.4 Fields

Field	Function
31-5 —	Reserved
4-1 MCLK2_SEL	MCLK2 Select Register 0000b - SAI1_CLK_ROOT is selected 0001b - SAI2_CLK_ROOT is selected 0010b - SAI3_CLK_ROOT is selected

Table continues on the next page...

Field	Function
	0011b - Reserved, MCLK2 is 0 0100b - SAI5_CLK_ROOT is selected 0101b - SAI6_CLK_ROOT is selected 0110b - SAI7_CLK_ROOT is selected 0111b - SAI1.MCLK is selected 1000b - SAI2.MCLK is selected 1001b - SAI3.MCLK is selected 1010b - Reserved, MCLK is 0 1011b - SAI5.MCLK is selected 1100b - SAI6.MCLK is selected 1101b - SAI7.MCLK is selected 1110b - SPDIF.ETXCLK is selected
0 MCLK1_SEL	MCLK1 Select Register 0b - SAI6_CLK_ROOT is selected 1b - SAI6.MCLK is selected

14.2.3.1.14 SAI7 MCLK SELECT Register (SAI7_MCLK_SEL)

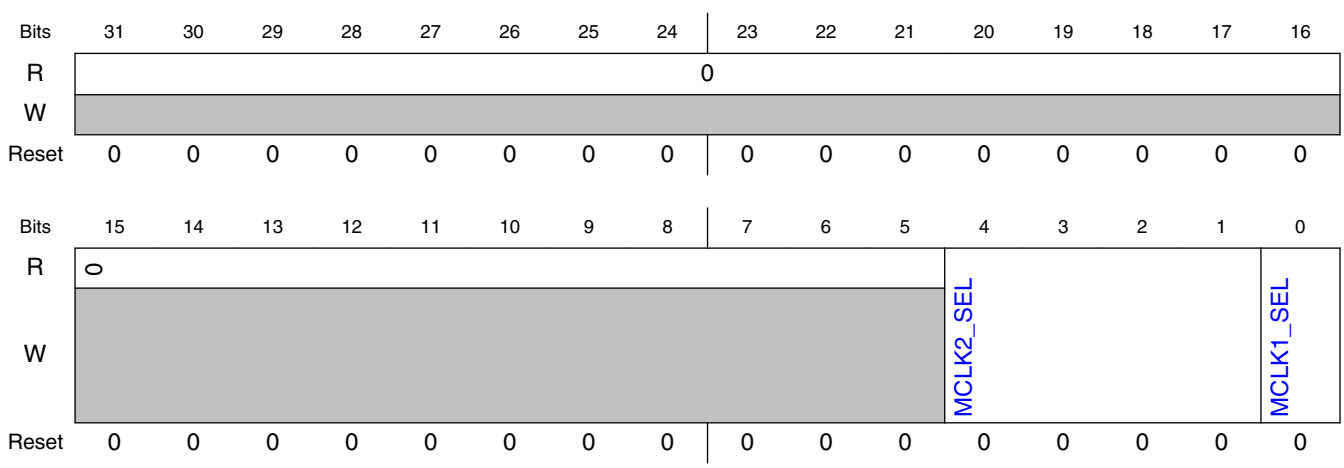
14.2.3.1.14.1 Offset

Register	Offset
SAI7_MCLK_SEL	314h

14.2.3.1.14.2 Function

SAI7 MCLK SELECT Register.

14.2.3.1.14.3 Diagram



14.2.3.1.14.4 Fields

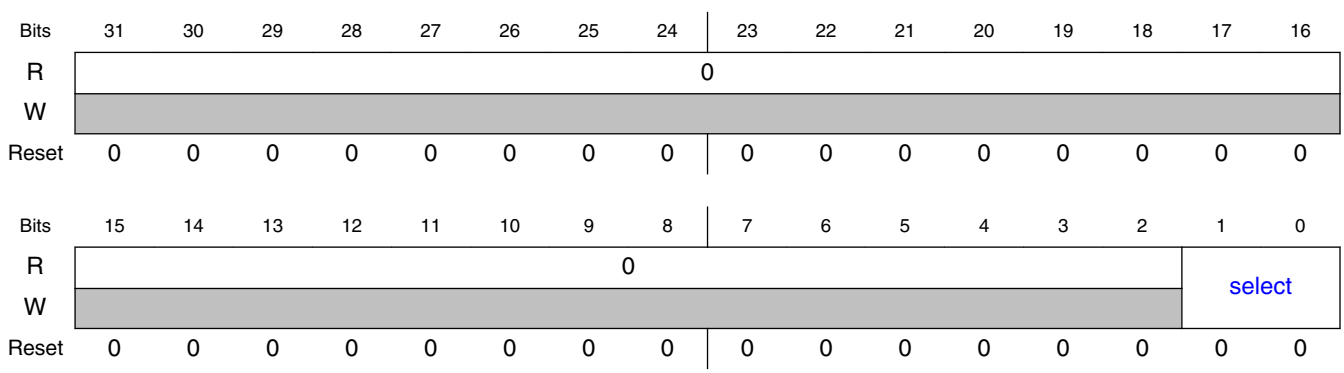
Field	Function
31-5 —	Reserved
4-1 MCLK2_SEL	MCLK2 Select Register 0000b - SAI1_CLK_ROOT is selected 0001b - SAI2_CLK_ROOT is selected 0010b - SAI3_CLK_ROOT is selected 0011b - Reserved, MCLK2 is 0 0100b - SAI5_CLK_ROOT is selected 0101b - SAI6_CLK_ROOT is selected 0110b - SAI7_CLK_ROOT is selected 0111b - SAI1.MCLK is selected 1000b - SAI2.MCLK is selected 1001b - SAI3.MCLK is selected 1010b - Reserved, MCLK is 0 1011b - SAI5.MCLK is selected 1100b - SAI6.MCLK is selected 1101b - SAI7.MCLK is selected 1110b - SPDIF.ETXCLK is selected
0 MCLK1_SEL	MCLK1 Select Register 0b - SAI7_CLK_ROOT is selected 1b - SAI7.MCLK is selected

14.2.3.1.15 PDM Root Clock Select Register (PDM_CLK)

14.2.3.1.15.1 Offset

Register	Offset
PDM_CLK	318h

14.2.3.1.15.2 Diagram



14.2.3.1.15.3 Fields

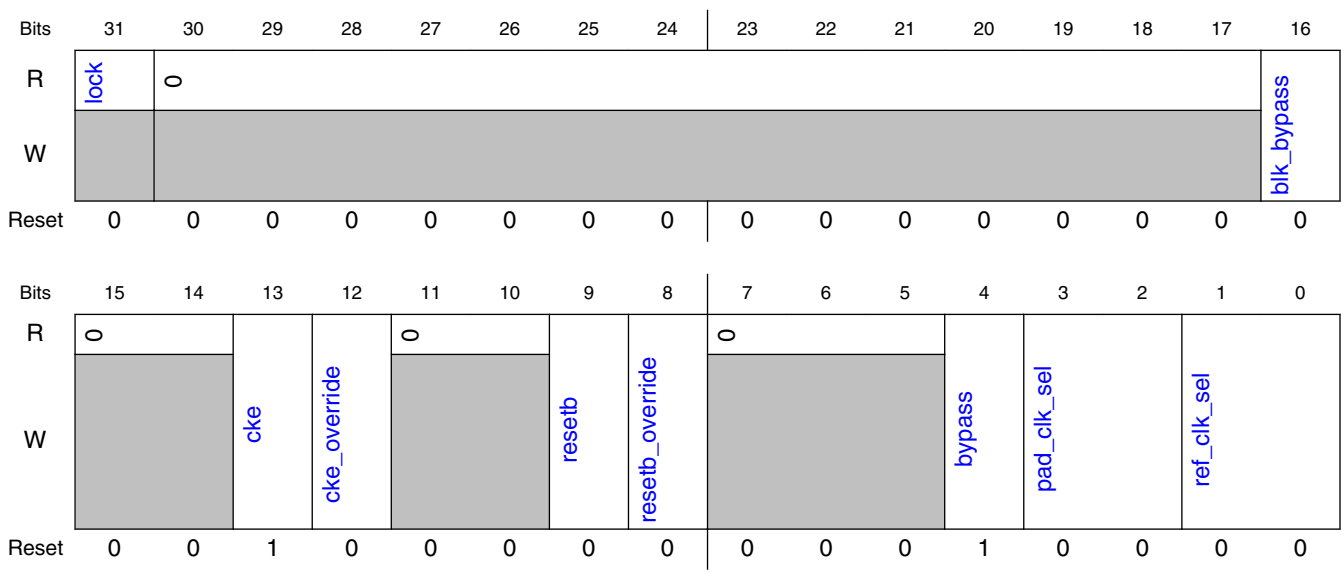
Field	Function
31-2 —	Reserved
1-0 select	PDM Root Clock Select Bits 00b - ccm pdm clock is selected 01b - sai_pll div2 is selected 10b - SAI1_MCLK is selected 11b - reserved.

14.2.3.1.16 SAI PLL General control Register (SAI_PLL_GNRL_CTL)

14.2.3.1.16.1 Offset

Register	Offset
SAI_PLL_GNRL_CTL	400h

14.2.3.1.16.2 Diagram



14.2.3.1.16.3 Fields

Field	Function
31	pll lock

Table continues on the next page...

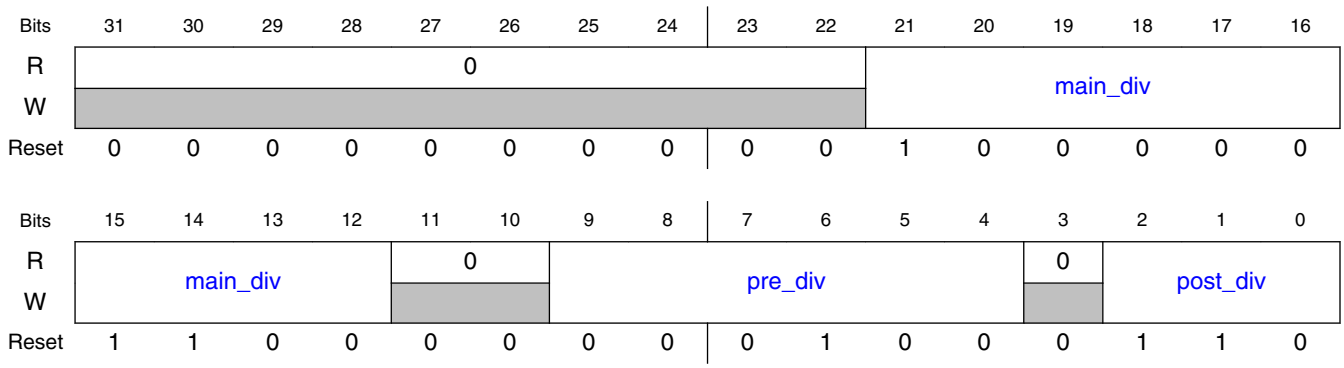
Field	Function
lock	
30-17 —	Reserved
16 blk_bypass	blk bypass
15-14 —	Reserved
13 cke	pll cke
12 cke_override	pll cke override.
11-10 —	Reserved
9 resetb	pll resetb
8 resetb_override	resetb override
7-5 —	Reserved
4 bypass	pll bypass
3-2 pad_clk_sel	pad clock select
1-0 ref_clk_sel	reference clock select

14.2.3.1.17 SAI PLL Frequency Divider control Register (SAI_PLL_FDIV_CTL0)

14.2.3.1.17.1 Offset

Register	Offset
SAI_PLL_FDIV_CTL0	404h

14.2.3.1.17.2 Diagram



14.2.3.1.17.3 Fields

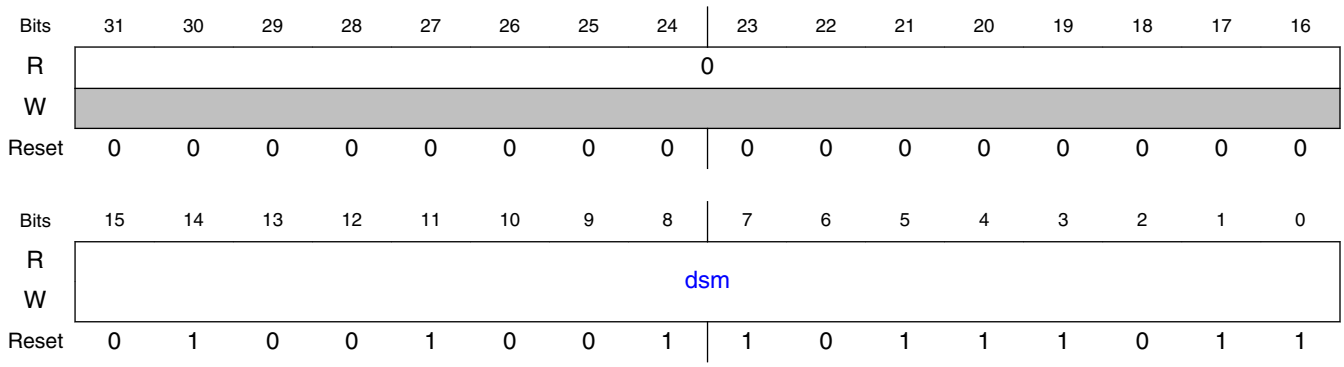
Field	Function
31-22 —	Reserved
21-12 main_div	main divider value
11-10 —	Reserved
9-4 pre_div	pre divider value
3 —	Reserved
2-0 post_div	post divider value

14.2.3.1.18 SAI PLL DSM value Register (SAI_PLL_FDIV_CTL1)

14.2.3.1.18.1 Offset

Register	Offset
SAI_PLL_FDIV_CTL1	408h

14.2.3.1.18.2 Diagram



14.2.3.1.18.3 Fields

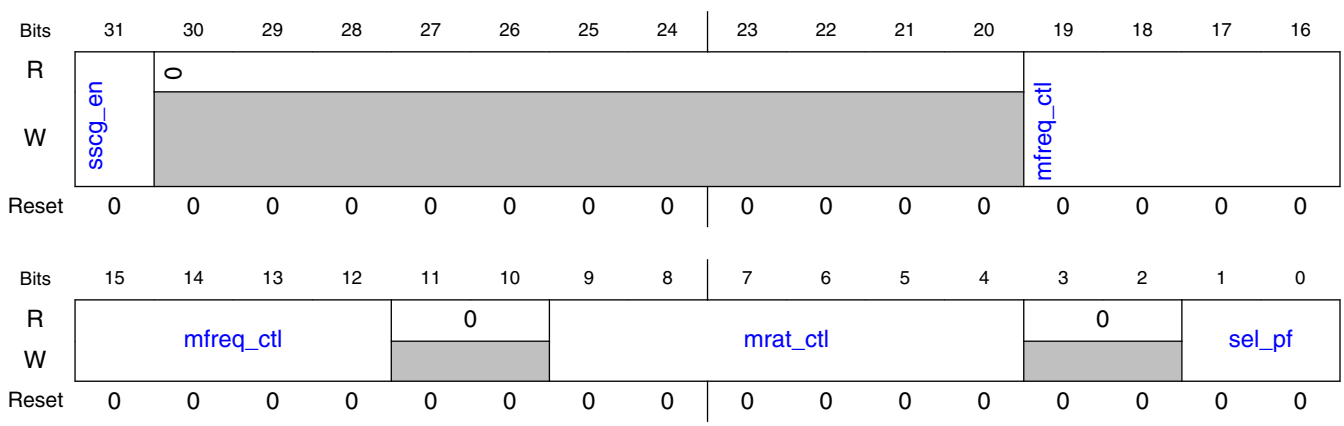
Field	Function
31-16 —	Reserved
15-0 dsm	pll DSM(K) value

14.2.3.1.19 SAI PLL SSCG control Register (SAI_PLL_SSCG_CTL)

14.2.3.1.19.1 Offset

Register	Offset
SAI_PLL_SSCG_CTL	40Ch

14.2.3.1.19.2 Diagram



14.2.3.1.19.3 Fields

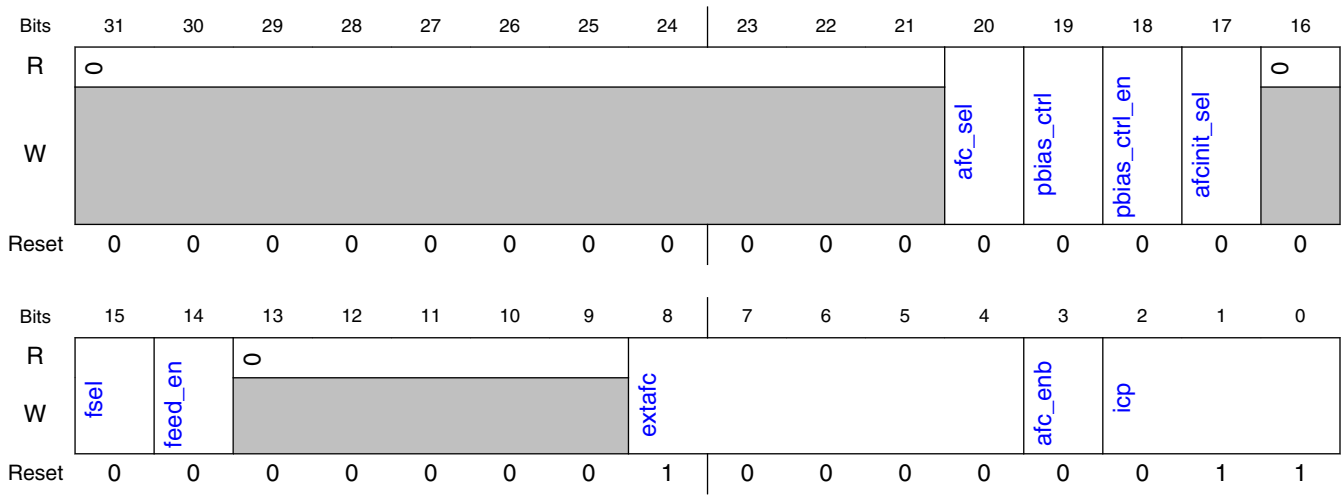
Field	Function
31 sscg_en	SSCG Enable Bit
30-20 —	Reserved
19-12 mfreq_ctl	pll modulation frequency control
11-10 —	Reserved
9-4 mrat_ctl	pll modulation rate control
3-2 —	Reserved
1-0 sel_pf	pll modulation method control

14.2.3.1.20 SAI PLL SSCG control Register (SAI_PLL_MNIT_CTL)

14.2.3.1.20.1 Offset

Register	Offset
SAI_PLL_MNIT_CTL	410h

14.2.3.1.20.2 Diagram



14.2.3.1.20.3 Fields

Field	Function
31-21 —	Reserved
20 afc_sel	PLL AFC SEL
19 pbias_ctrl	PLL PBIAS CTRL
18 pbias_ctrl_en	PLL PBIAS CTRL EN
17 afcinit_sel	PLL AFC INIT SEL
16 —	Reserved
15 fsel	PLL FEED SEL control
14 feed_en	PLL FEED Enable control
13-9 —	Reserved
8-4 extafc	AFC Enable control
3 afc_enb	AFC Enable control

Table continues on the next page...

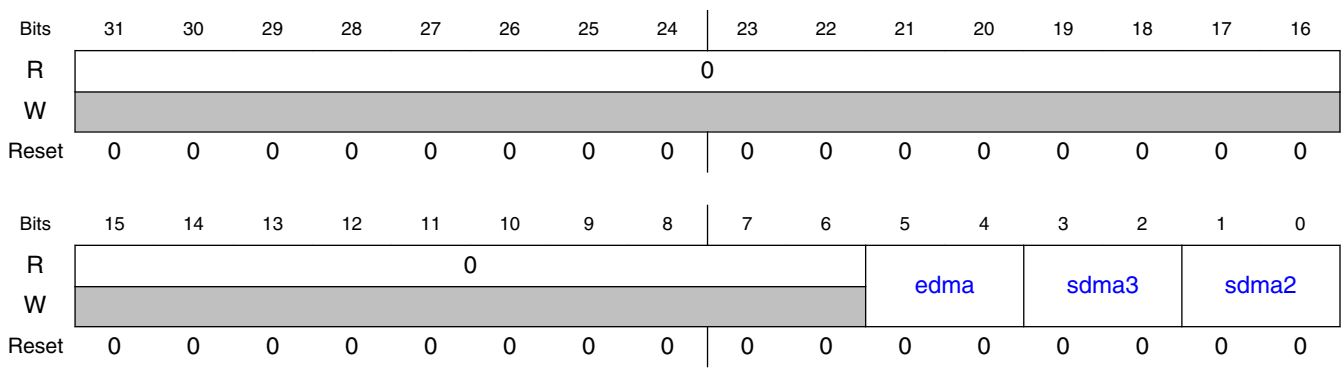
Field	Function
2-0 icp	Charge-pump current control

14.2.3.1.21 AUDIOMIX Extra Addr Bits Register (AUDIO_EXT_ADDR)

14.2.3.1.21.1 Offset

Register	Offset
AUDIO_EXT_ADDR	500h

14.2.3.1.21.2 Diagram



14.2.3.1.21.3 Fields

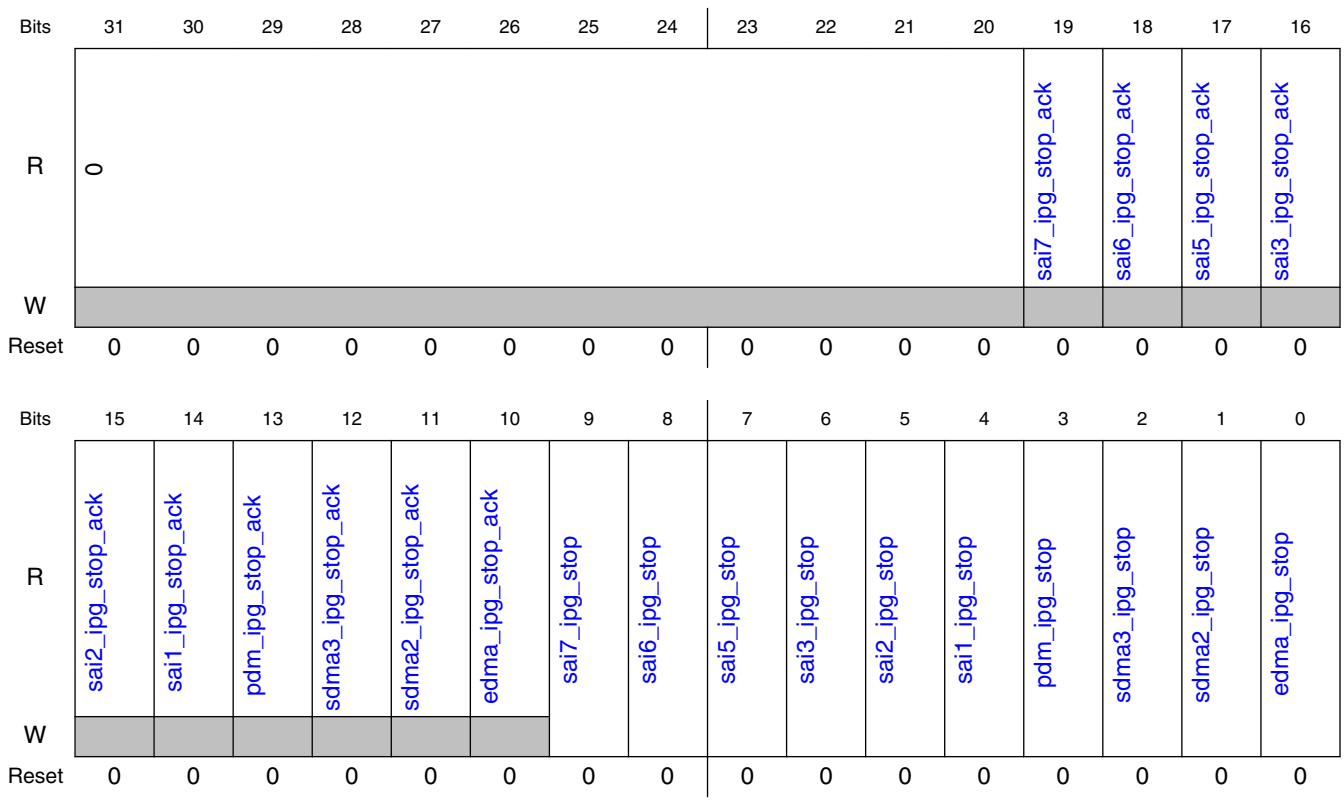
Field	Function
31-6 —	Reserved
5-4 edma	EDMA extra Addr Bits
3-2 sdma3	SDMA3 Extra Addr Bits
1-0 sdma2	SDMA2 Extra Addr Bits

14.2.3.1.22 IPG Low Power Control Register (IPG_LP_CTRL)

14.2.3.1.22.1 Offset

Register	Offset
IPG_LP_CTRL	504h

14.2.3.1.22.2 Diagram



14.2.3.1.22.3 Fields

Field	Function
31-20 —	Reserved
19 sai7_ipg_stop_ack	SAI1 IPG_STOP_ACK Bit
18	SAI6 IPG_STOP_ACK Bit

Table continues on the next page...

Field	Function
sai6_ipg_stop_ack	
17 sai5_ipg_stop_ack	SAI5 IPG_STOP_ACK Bit
16 sai3_ipg_stop_ack	SAI3 IPG_STOP_ACK Bit
15 sai2_ipg_stop_ack	SAI2 IPG_STOP_ACK Bit
14 sai1_ipg_stop_ack	SAI1 IPG_STOP_ACK Bit
13 pdm_ipg_stop_ack	PDM IPG_STOP_ACK Bit
12 sdma3_ipg_stop_ack	SDMA3 IPG_STOP_ACK Bit
11 sdma2_ipg_stop_ack	SDMA2 IPG_STOP_ACK Bit
10 edma_ipg_stop_ack	EDMA IPG_STOP_ACK Bit
9 sai7_ipg_stop	SAI1 IPG_STOP Bit
8 sai6_ipg_stop	SAI6 IPG_STOP Bit
7 sai5_ipg_stop	SAI5 IPG_STOP Bit
6 sai3_ipg_stop	SAI3 IPG_STOP Bit
5 sai2_ipg_stop	SAI2 IPG_STOP Bit
4 sai1_ipg_stop	SAI1 IPG_STOP Bit
3 pdm_ipg_stop	PDM IPG_STOP Bit
2 sdma3_ipg_stop	SDMA3 IPG_STOP Bit

Table continues on the next page...

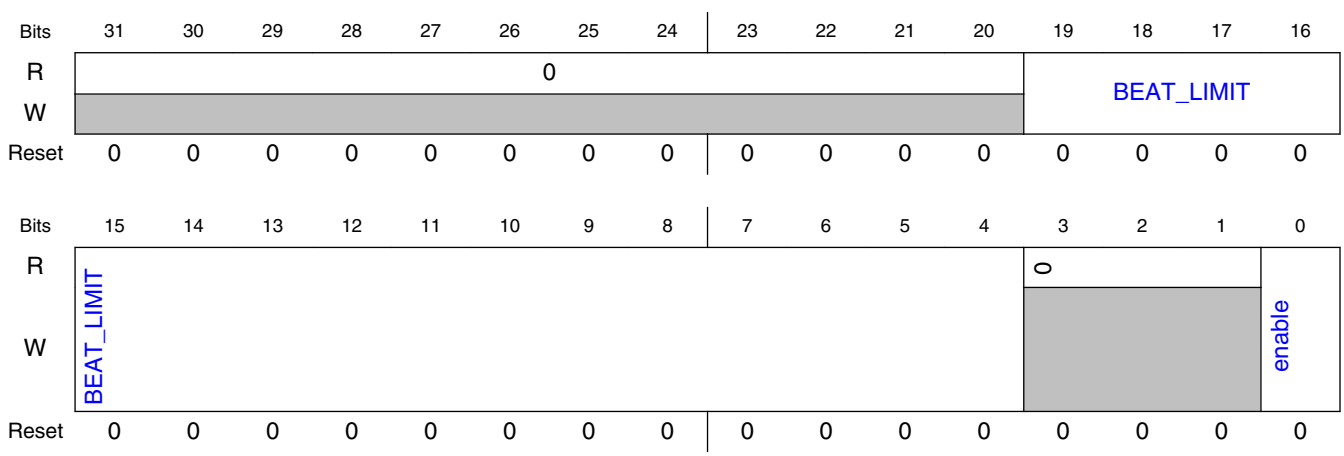
Field	Function
1 sdma2_ipg_stop	SDMA2 IPG_STOP Bit
0 edma_ipg_stop	EDMA IPG_STOP Bit

14.2.3.1.23 AUDIOMIX AXI LIMIT CTRL Register (AUDIO_AXI_LIMIT)

14.2.3.1.23.1 Offset

Register	Offset
AUDIO_AXI_LIMIT	508h

14.2.3.1.23.2 Diagram



14.2.3.1.23.3 Fields

Field	Function
31-20 —	Reserved
19-4 BEAT_LIMIT	Beat Limit. Limit the burst beat number from AudioDSP.
3-1 —	Reserved
0 enable	AXI Limit enable

14.3 PDM Microphone Interface (MICFIL)

14.3.1 Overview

The Pulse Density Modulated Microphone Interface (MICFIL) is a popular way to deliver audio from microphones to the processor in several applications, such as mobile telephones. However, current digital-audio systems use multibit audio signal (also known as multibit PCM) to represent the signal. This block implements the required digital interface to provide a 24-bits audio signal from a PDM microphone bitstream in a configurable output sampling rate.

The implementation of this digital interface is based on the application of digital signal processing techniques in hardware. The PDM Microphone Interface architecture was designed to save gate count and minimize power consumption. It implements a series of filters to transform a PDM bitstream to a 24-bit PCM signal in the audio band.

The whole module is implemented to work in a multichannel mode. All channels have the same configuration but each input channel could be turned on/off independently.

14.3.1.1 Block Diagram

Figure 14-5 shows the block diagram for the PDM Microphone Interface module. It consists of the following:

- an input interface for each pair of PDM microphones.
- a decimation filter by channel.
- a FIFO by channel.
- a time generation unit.
- shared interfaces to DMA, interrupts and SoC.
- A Hardware Voice Activity Detector (HWVAD).

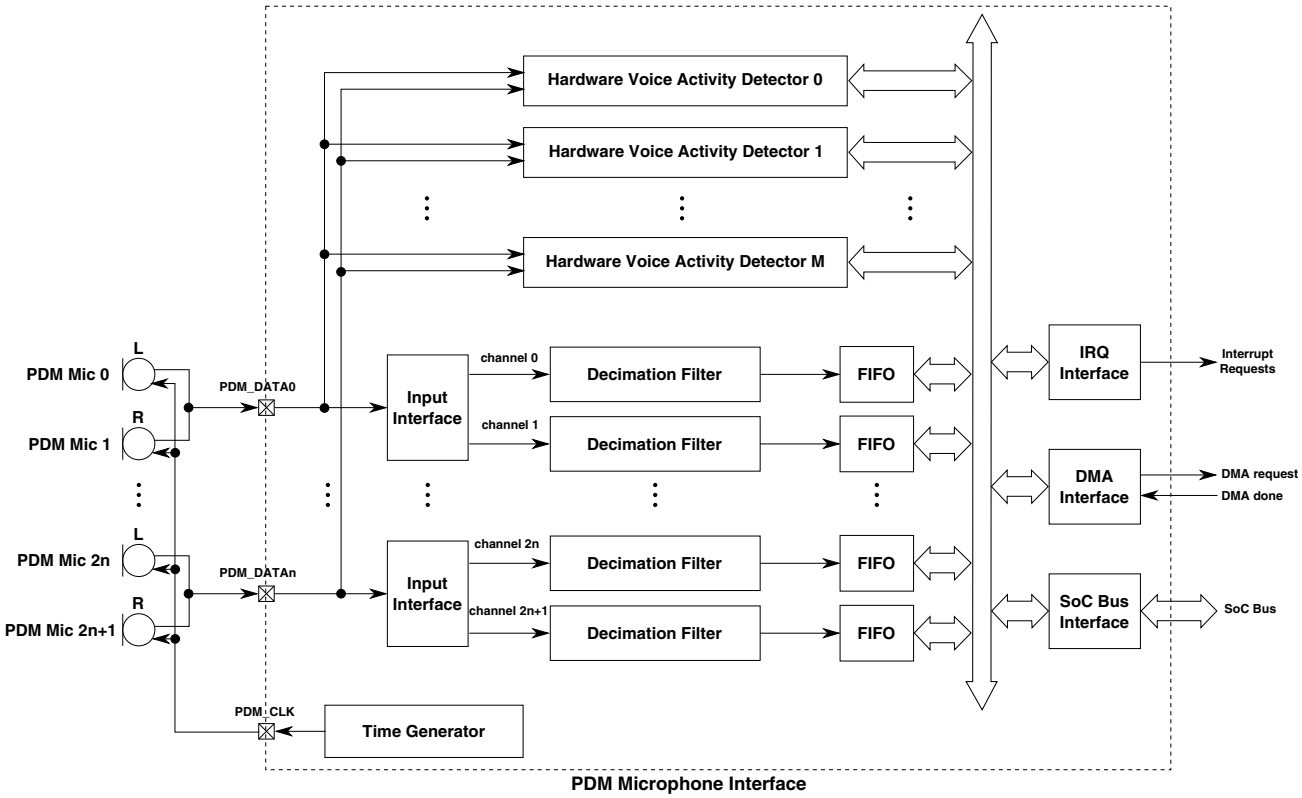


Figure 14-5. PDM Microphone Interface Block Diagram

The Time Generator unit delivers the PDM_CLK to all microphones. All microphones must operate at the same clock frequency. Each input interface receives a time multiplexed PDM bitstream from two PDM microphones and it separates the two audio information in two channels: left (0) and right (1). Each channel is processed separately.

The Decimation Filter is composed internally by a CIC filter, a DC remover and Half Band filters. The filtering results are stored in individual FIFOs; a FIFO by channel. The FIFOs have overflow and underflow detectors to deliver an error interrupt request.

An interrupt request or DMA request could be delivered by the MICFIL. Then the filters results stored in FIFOs could be accessed by the SoC or DMA respectively via the internal bus interface .

The HWVAD takes data from the input of a selected PDM microphone to detect if there is any voice activity. When a voice activity is detected, an interrupt could be delivered to the system.

14.3.1.2 Features

The PDM Microphone Interface (MICFIL) includes the following features:

- Decimation filters:

- Fixed-point filtering
- 24-bit PCM audio output
- Internal clock divider for a programmable PDM clock generation
- Full or partial set of channels operation with individual enable control
- Programmable decimation rate
- Programmable DC remover
- Range adjustement capability
- FIFOs with interrupt and DMA capability
    - Each FIFO with 32 entries length

- Hardware Voice Activity Detector (HWVAD)
    - Interrupt capability
    - Zero-Crossing Detection (ZCD) option

14.3.2 Functional Description

Figure 14-5 shows the PDM Microphone Interface (MICFIL) block diagram. Basically this module is composed by a Decimation Filter and a FIFO by channel, an Input Interface for each pair of microphones, a shared Time Generator unit, a shared bus interface, a shared DMA interface and a shared interrupt interface.

The Decimation Filter implements a low-pass filter in the audio band (20Hz-22.5kHz @48kHz output sampling rate by default) with a configurable decimation rate. It is implemented using a series of CIC, Half Band and DC remover filters.

The Time Generator unit generates the PDM_CLK to the microphones. This clock is the same for all the PDM microphones and it is active for all the microphones, i.e. there is not possiblity to turn off the PDM_CLK for one microphone only.

The output of each Decimation Filter is stored into a FIFO buffer. Each FIFO is mapped in the DATACHn registers. It is possible to generate either an interrupt or a DMA request when, in each FIFO of all enabled channels, the number of data stored surpasses a configured watermark.

Also, independently on decimation filters, there is a Hardware Voice Activity Detector(HWVAD) which implements voice-detection algorithms to generate wake-up interrupts.

14.3.2.1 Modes of operation

This section describes the MICFIL operation modes which depends on the SoC power mode , the SoC Doze mode , the SoC Debug mode and MDIS, DBGE, DBG and DOZEN bitfields as summarized in Table 14-3.

Table 14-3. Operation Mode Selection

Mode	MDIS	SoC Doze & DOZEN= 1	SoC Power Mode	DOZEN	SoC Debug | DBG	DBGE	CPU clock	MICFIL internal clock	Decimation Filter state	HWVAD[n] state
Normal	0	0	RUN	Any	0	Any	Enabled	Enabled	Run (Depends on PDMIEN and CHENn)	Run (Depends on PDMIEN and VADEN)
Debug					1	1				
						0			Stop	Stop
Low Power			SLEEP	0	Any	Any	Gated (Externally)		Run (Depends on PDMIEN and CHENn)	Run (Depends on PDMIEN and VADEN)[1]
			DEEPSLEEP						Stop	
Disable/Low Leakage			SLEEP/ DEEPSLEEP	1				Gated		Stop
		1	Any	Any			Enabled			
	1	Any					Gated (Internally)			

1. HWVAD[n] will run in DEEPSLEEP only if it is placed on an always-on power domain in the SoC.

14.3.2.1.1 Normal mode

This is the default MICFIL operational mode. In this mode the channels can be enabled to perform filtering operation.

14.3.2.1.2 Low Power mode

In Low Power mode the MICFIL is able to filter the PDM microphone data and perform voice detection such as in Normal mode and can generate a wake up event through asynchronous DMA/IRQ requests depending on CHENn, VADEN and PDMIEN values.

The Low Power mode is entered when the SoC enters to SLEEP or DEEPSLEEP modes and DOZEN =0. In this mode the CPU clock is gated externally and MICFIL could be kept running.

In case a SoC SLEEP mode is requested, the decimation filter is kept running such as Normal mode and the SoC SLEEP mode is acknowledged immediately.

In case a SoC DEEPSLEEP mode is requested it is waited until the last input samples are processed to acknowledge the SoC mode and stop the decimation filter.

During this mode, VAD will continue running if it was previously enabled (VADEN=1) during Normal mode.

The DMA and IRQ requests can be generated asynchronously to wake up the CPU. .

NOTE

The asynchronous DMA or asynchronous interrupt requests are generated only in this mode.

14.3.2.1.3 Debug mode

This mode is used to check the status of the IP when requested. In this mode the MICFIL can be stopped after the remaining data processing has finished depending on the DBGE bit value. This mode can be actived by a SoC debug request or asserting the DBG bitfield.

In DBGE=0 case, if a debug request is received or DBG is asserted while the module is running, it will be stopped after finish to process the remaining data. Otherwise, if DBGE=1, the module will be kept running.

The PDM Microphone Interface cannot enter Debug mode once in Disable/Low Leakage or Low Power mode.

The Debug mode request affects all channels in the MICFIL. Access to output buffers remain operational, as well as their related flags.

NOTE

Once a debug request is issued, it can not be aborted.

14.3.2.1.4 Disable/Low Leakage mode

In this mode, MICFIL is stopped after the remaining data processing has finished. MICFIL is in Disable/Low Leakage (DLL) mode when:

- The SoC is in SLEEP and DEEPSLEEP modes and DOZEN=1.
- The SoC is in Doze and DOZEN bit is asserted.
- MDIS is asserted.

When Disable/Low Leakage (DLL) mode is requested while the module is running, it will be stopped after finish to process the remaining data.

Writes only to CTRL_1 and CTRL_2 registers are allowed with the exception of writes to the DBG and SRES bits, which are ignored, but read is still available for these bits.

14.3.2.2 Output Rate

The output rate is the sampling rate at the decimation filter output. It depends on the sampling rate of the microphone input and the configured quality mode (refer to Quality Modes) . Common audio sampling rates are 48kHz and 16kHz. On this document the output rate is referred sometimes as FS.

14.3.2.3 Quality Modes

Independently on the power mode selected, if the decimation filter is running, the MICFIL can operate in 6 different quality modes, those can be selected by the QSEL bitfield. The decimation and interpolation rates in internal blocks are shown in Table 14-4 where OSR depends on the CICOSR as shown in Equation 1 on page 6025.

$$OSR = 16\text{-}CICOSR$$

Equation 1. Oversampling Rate (OSR)

The quality selected can be configured dinamically when the Decimation Filter is running, the output rate is mantained in every quality selected.

The passband of the Decimation Filter depends on the quality mode, refer to Overall Passband for more information.

In order to mantain the same output rate in the Decimation Filter when the quality is changed, the PDM_CLK rate changes depending on the quality selected. For more details about the PDM_CLK, refer to Time Generator.

The Sampler and the Half Band filters are whether enabled or not depending on the quality selected, achieving a low power consumption when they are disabled. The CIC filter decimation rate is also whether doubled or not depending on the quality selected. For more information, refer to Decimation Filter.

NOTE

In order to get a better audio performance, the RANGEADJn bitfield need to be adjusted depending on the quality mode selected as pointed out on its description. For more information, refer to Overall Filter Gain.

Table 14-4. Quality Modes

Quality Mode	QSEL bitfield	Sampler Interpolation	CIC Filter Decimation	1st Half Band Filter Decimation	2nd Half Band Filter Decimation	PDM_CLK Rate	Passband
High Quality	001	-	:(2OSR)	:2	:2	Output Rate x 8 x OSR	To ~0.5 x Output Rate
Medium Quality	000	-	:OSR	:2	:2	Output Rate x 4 x OSR	To ~0.5 x Output Rate
Low Quality	111	x2	:OSR	:2	:2	Output Rate x 2 x OSR	To ~0.5 x Output Rate
Very Low Quality 0	110	-	:(2OSR)	:2	-	Output Rate x 4 x OSR	To ~0.25 x Output Rate
Very Low Quality 1	101	-	:OSR	:2	-	Output Rate x 2 x OSR	To ~0.25 x Output Rate
Very Low Quality 2	100	x2	:OSR	:2	-	Output Rate x OSR	To ~0.25 x Output Rate

14.3.2.4 Time Generator

The Time Generator is the responsible to generate the PDM_CLK to the microphones.

NOTE

The Time Generator is affected by soft-reset. The PDM_CLK starts with a rising edge after a soft reset is issued.

14.3.2.4.1 Clock divider

The sampling rate of the microphone input depends on the PDM_CLK rate. The PDM_CLK rate can be trimmed using an internal clock divider, configuring the quality mode (refer to Table 14-4) or trimming the MICFIL's root clock frequency (MICFIL_CLK_ROOT rate).

The internal clock divider frequency is defined by the CLKDIV bitfield value which must be configured following Equation 2 on page 6026.

$$CLKDIV = \frac{\text{MICFIL_CLK_ROOT rate}}{8(16\text{-CICOSR})(\text{Output Rate})}$$

Equation 2. CLKDIV bitfield value

Once calculated the CLKDIV value, the generated PDM_CLK rate can be estimated as shown in Equation 3 on page 6026, where the K factor value depends on the quality mode as shown in Table 14-5 and floor(x) is the function that takes as input a real number x and gives as output the greatest integer less than or equal to x.

$$\text{PDM_CLK rate} = \frac{\text{MICFIL_CLK_ROOT rate}}{2 \cdot \text{floor}(K \cdot CLKDIV)}$$

Equation 3. PDM_CLK rate

Table 14-5. K factor value

Quality mode	K factor
High Quality	1/2
Medium Quality, Very Low Quality 0	1
Low Quality, Very Low Quality 1	2
Very Low Quality 2	4

14.3.2.4.1.1 Minimum Required CLKDIV

NOTE

This minimum CLKDIV condition is required only if any decimation filter channel is enabled and the clock divider is enabled, otherwise this condition should be ignored.

To work properly the CLKDIV bitfield value must have a minimum value according to the quality mode configured (see Quality Modes), the number of enabled channels (EC), Equation 1 on page 5977 and K factor (see Input/Output Timing Requirements). When this condition is not met the LOWFREQF bit is asserted and an error interrupt can be issued (in case ERREN=1).

The equations to estimate the minimum required CLKDIV bitfield value depending on the configured quality mode are shown in Equation 4 on page 6027 (for medium, high and low quality modes) and Equation 5 on page 6027 (for very low quality modes), on these equations floor(x) is the function that takes as input a real number x and gives as output the greatest integer less than or equal to x.

$$\text{floor}(\text{K x CLKDIV}) >= \text{K x } \frac{10 + 93EC}{8 \text{ x OSR}}$$

Equation 4. Minimum required CLKDIV value in Medium, High and Low Quality modes

$$\text{floor}(\text{K x CLKDIV}) >= \text{K x } \frac{10 + 43EC}{8 \text{ x OSR}}$$

Equation 5. Minimum required CLKDIV value in Very Low Quality modes

14.3.2.4.2 MICFIL_CLK_ROOT rate and CLKDIV calculation examples

Example 1:

If it is required an output rate=16kHz and enable 4 channels (EC=4). A valid configuration is following:

1. Configure High Quality mode and CIC decimation by 32 (CICOSR =0, then OSR=16), .
2. Check the PDM_CLK rate for High Quality mode in Table 14-4. PDM_CLK rate = 16kHz x 8 x OSR = 16kHz x 8 x 16 = 2.048MHz.
3. Check the K factor value for High Quality mode in Table 14-5. K factor is 1/2.
4. Calculate minimum CLKDIV value (see Equation 4 on page 5979): floor(K x CLKDIV) >= K x (10 + 93EC)/(8 x OSR) then floor(K x CLKDIV) >= 2.984375. Finally, CLKDIV >= 5.96875.
5. Calculate the minimum MICFIL_CLK_ROOT rate:
   - From Equation 2 on page 5978 : MICFIL_CLK_ROOT rate = 8 x (16-CICOSR) x CLKDIV x (Output rate)
   - From step 4 result: CLKDIV>=4, then MICFIL_CLK_ROOT rate >= 8 x (16-CICOSR) x 5.96875 x 16kHz.
   - Finally: MICFIL_CLK_ROOT rate >= 12224kHz
   - Select the MICFIL_CLK_ROOT rate in that range.
6. Check if selected MICFIL_CLK_ROOT rate and CLKDIV value satisfies the Table 14-6.
7. Finally, check again if the selected MICFIL_CLK_ROOT rate satisfies Equation 2 on page 5978. If not, adjust CLKDIV or change MICFIL_CLK_ROOT rate accordingly.

Example 2:

If it is required an output rate=48kHz and enable 4 channels (EC=4). A valid configuration is following:

1. Configure Medium Quality mode and CIC decimation by 16 (CICOSR =0, then OSR=16), .
2. Check the PDM_CLK rate for Medium Quality mode in Table 14-4. PDM_CLK rate = 48kHz x 4 x OSR = 48kHz x 4 x 16 = 3.072MHz.
3. Check the K factor value for Medium Quality mode in Table 14-5. K factor is 1.
4. Calculate minimum CLKDIV value (see Equation 4 on page 5979): floor(K x CLKDIV) >= K x (10 + 93EC)/(8 x OSR) then floor(K x CLKDIV) >= 2.984375. Finally, CLKDIV >= 2.984375.
5. Calculate the minimum MICFIL_CLK_ROOT rate:
   - From Equation 2 on page 5978 : MICFIL_CLK_ROOT rate = 8 x (16-CICOSR) x CLKDIV x (Output rate)
   - From step 4 result: CLKDIV>=2.984375, then MICFIL_CLK_ROOT rate >= 8 x (16-CICOSR) x 2.984375 x 48kHz.

- Finally: MICFIL_CLK_ROOT rate >= 18336kHz
- Select the MICFIL_CLK_ROOT rate in that range.

6. Check if selected MICFIL_CLK_ROOT rate and CLKDIV value satisfies the Table 14-6.
7. Finally, check again if the selected MICFIL_CLK_ROOT rate satisfies Equation 2 on page 5978. If not, adjust CLKDIV or change MICFIL_CLK_ROOT rate accordingly.

14.3.2.4.3 Input/Output Timing Requirements

The PDM microphones must meet the setup and hold timing requirements shown in Table 14-6 and Figure 14-6. The "k" factor value in Table 14-6 depends on the selected quality mode as shown in Table 14-5.

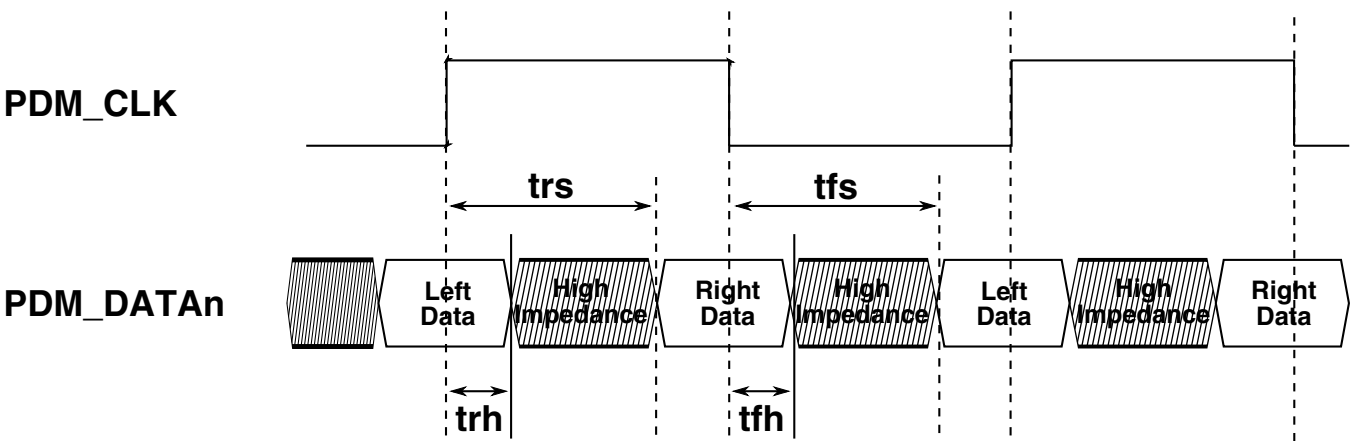


Figure 14-6. Input/Output TIming Requirements

Table 14-6. Timing parameters

Parameter	Value
trs, tfs	$<= \frac{\text{floor}(K \times CLKDIV) - 1}{\text{MICFIL_CLK_ROOT rate}}$ [1]
trh. tfh	>=0

1. Depending on K value, the user must make sure floor(K x CLKDIV) > 1 to avoid timing problems.

14.3.2.5 Decimation Filter

Figure 14-7 shows the block diagram of the Decimation Filter. This block is composed by a CIC filter, a DC remover, two Half Band filters for each channel.

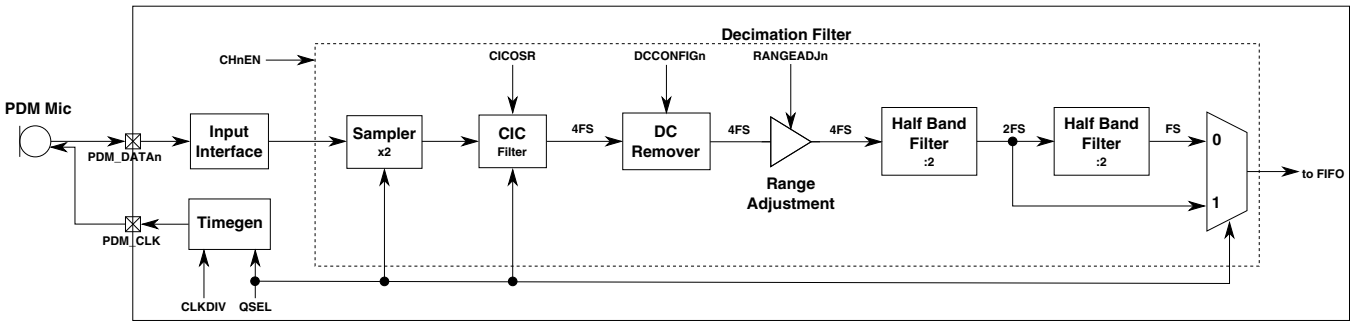


Figure 14-7. Decimation Filter Block Diagram

The Sampler block is activated depending on the quality mode selected (QSEL bitfield). This one samples twice the PDM_DATAn incoming from the Input Interface.

The CIC filter (Cascaded Integrator Comb filter) is a low pass filter commonly used to filter the sigma-delta modulator output signal.

The DC remover is a high pass filter used to remove the DC component of the processed signal with configurable cut-off frequency.

Each Half Band filter implements a low pass digital filter with decimation by 2. They are used to compensate the high CIC drop in passband and helps the FIR filter block to flat the overall passband response.

The Decimation Filter output depends on the quality mode selected by the QSEL bitfield, so it can come from any of the Half Band filters output. In case, the last Half Band filter stage is bypassed, power consumption should be reduced.

The range of each Decimation Filter can be adjusted by a Range Adjustment block so that, in case a adjustment overflows or underflows is detected. an error interrupt is raised.

14.3.2.5.1 Filter Performance

14.3.2.5.1.1 Overall Filter Gain

The overall filter gain depends on the quality mode, the Equation 8 on page 6032 and the RANGEADJn bitfield as shown in Equation 6 on page 6030.

$$OverallFilterGain\ (dB) = \begin{cases} 140 \cdot \log_{10}(32 - 2CICOSR) + 6.02RANGEADJn - 162.54 & ;if\ QSEL \in [HQ, VLQ0] \\ 140 \cdot \log_{10}(16 - CICOSR) + 6.02RANGEADJn - 162.54 & ;others \end{cases}$$

Equation 6. Overall Filter Gain

Finally, the input and output dBFS levels are related as shown in Equation 7 on page 6030.

$$Output(dBFS) = Input(dBFS) + OverallFilterGain(dB)$$

Equation 7. Filter input and output relationship

14.3.2.5.1.2 Overall Passband

The Decimation Filter passband depends on the quality mode and the output rate (FS). For instance, the overall filter passband for different output data rates when the MICFIL is in High, Medium and Low Quality modes is shown in Table 14-7.

Table 14-7. Filter Passband in HQ, MQ and LQ modes (DCCONFIGn=00)

Output Data Rate	Overall Filter Passband
FS	**0.0004FS - 0.4688FS**
16kHz	6.67Hz - 7.50kHz
48kHz	20Hz - 22.5kHz

The overall filter passband for different output data rates when the MICFIL is in Very Low Quality modes is shown in Table 14-8.

Table 14-8. Filter Passband in VLQn modes (DCCONFIGn=00)

Output Data Rate	Overall Filter Passband
FS	**0.0002FS - 0.2344FS**
16kHz	3.33Hz - 3.75kHz
48kHz	10.00Hz - 11.25kHz
96kHz	20Hz - 22.5kHz

14.3.2.5.2 Input Interface

The timing diagram of the input interface signals is shown in Figure 14-8. The bitstream incoming from the microphone data input "n" (PDM_DATAn) in the first half (Right microphone) of the PDM_CLK is directed to the Channel "2n+1" and the data generated during the last half (Left microphone) is directed to the Channel "2n".

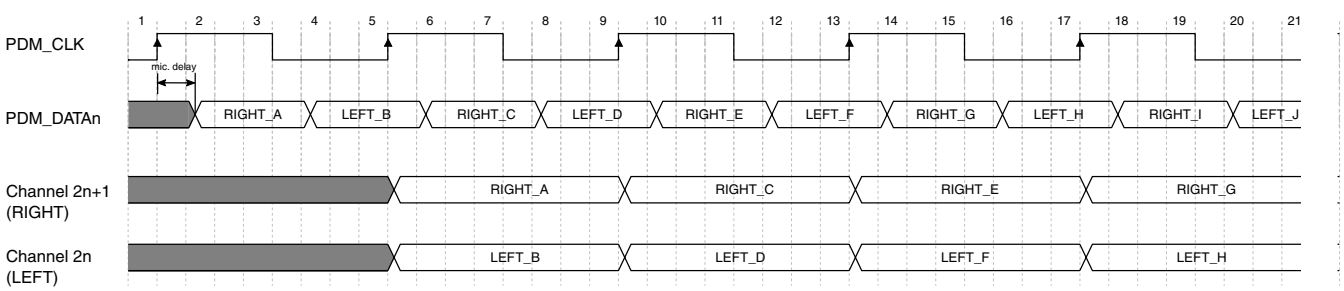


Figure 14-8. Input Interface signals

14.3.2.5.3 CIC Filter

The CIC filter is an optimized implementation of the low pass Sinc filter with multiple stages. It is widely used because it requires just adders and subtractors. The CIC filter is composed by sub-blocks called: integrator and comb. The CIC decimation rate is defined by the CICOSR bitfield and the selected quality mode as shown in Equation 8 on page 6032.

$$\text{CIC Decimation Rate} = \begin{cases} 2*(16-\text{CICOSR}) & ;\text{if HQ or VLQ0} \\ 16-\text{CICOSR} & ;\text{others} \end{cases}$$

Equation 8. CIC Decimation Rate

14.3.2.5.4 DC Remover

There are two DC removers in the filter chain, one in the input and another in the output. The former is integrated in the CIC filter, and the later is placed just before the FIFO. Both can be bypassed independently.

The block removes the DC level component of the signal using a high-pass filter with a very low cut-off frequency selected by the DCCONFIGn bitfield. This filter is bypassed when DCCONFIGn=11.

14.3.2.5.5 Half Band Filters

A half band filter divides the signal base-band into two equal subbands before to be decimated by 2. It also compensates the CIC filter drop in passband and helps to flatten the overall passband response. MICFIL has two half band filter stages, with the possibility of bypass the last one.

14.3.2.6 Hardware Voice Activity Detector

The Hardware Voice Activity Detector (HWVAD) is a block responsible for detect voice activity in a channel selected by the user as shown in the Figure 14-9 and Figure 14-10. It can be configured in Envelope-based or Energy-based mode and it should be configured as described in Initialization.

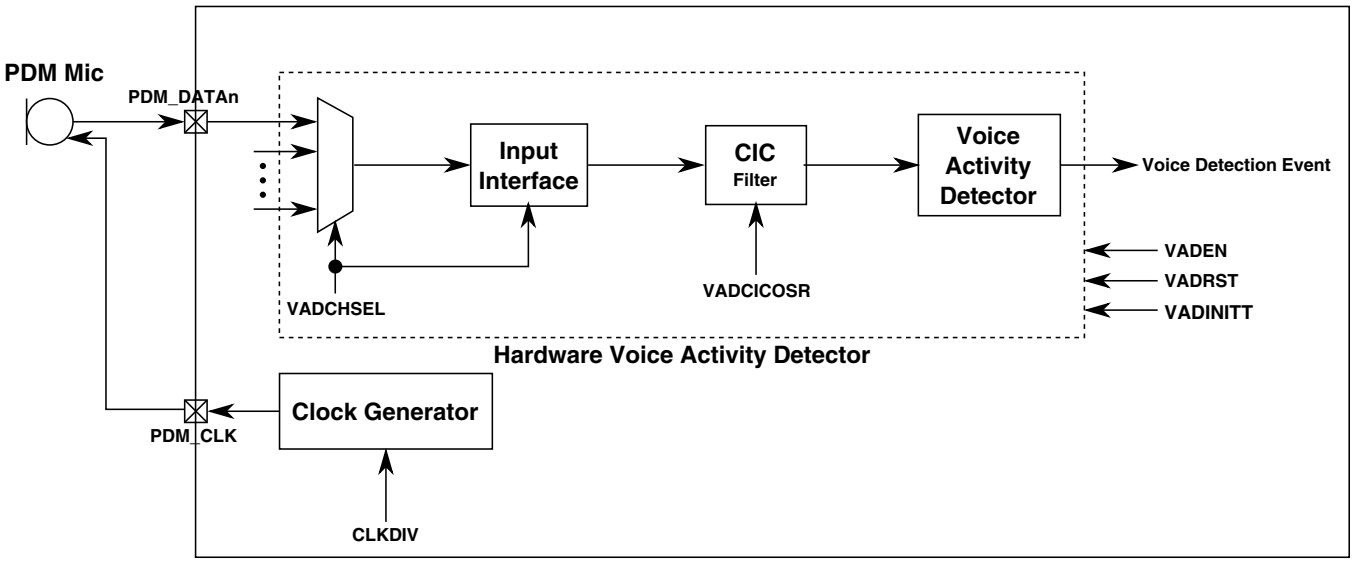


Figure 14-9. HWVAD Block Diagram

The HWVAD can divide the incoming PCM signal from its internal CIC in frames, the frame energy is used to estimate the background noise. Both frame energy and noise energy are examinated, a voice activity is detected when a specific delta between them is detected, then a interrupt will be issued. Optionally a Zero-Crossing Detection block (ZCD) could be enabled to avoid low energy voiced speech be missed, improving the voice detection performance.

The HWVAD detector can be enabled by the VADEN bit (VAD_CTRL_1 register) and could be reset by software through the VADRST bit (VAD_CTRL_1 register) . It is important to remark that the HWVAD detector could be enabled and/or reset only when the MICFIL isn't running i.e. when the BSY_FIL bit in STAT register is cleared.

To perform voice detection a Noise Filter and Signal Filter are used as shown Figure 14-10. In addition, a minimum block in the output of the Noise Filter and a maximum block in the ouput of the Signal Filter can be used. The result of both blocks can be scaled by gain blocks. Finally, the signal and noise estimated are compared, voice is detected when signal is greater than noise.

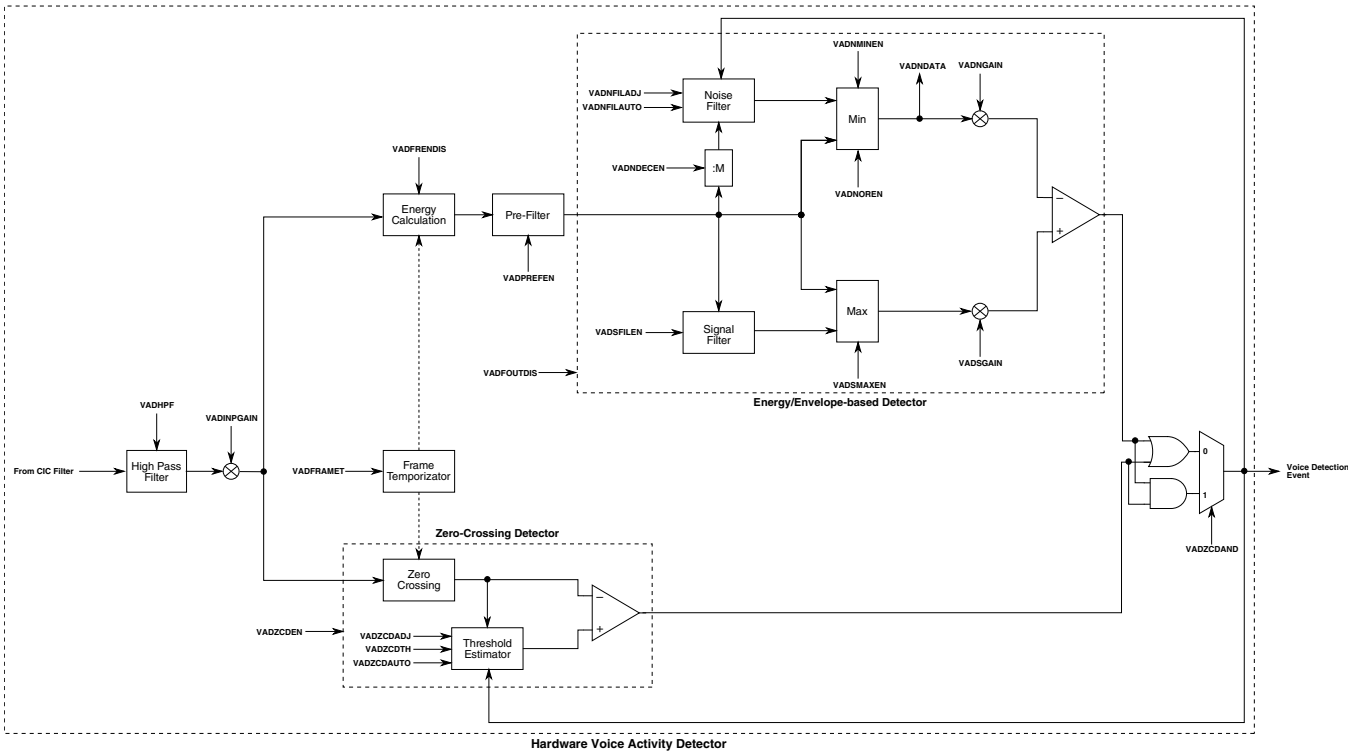


Figure 14-10. HWVAD Block Diagram (cont.)

14.3.2.6.1 High-Pass Filter

The purpose of the High-Pass Filter is to remove the low frequency components. The cut-off frequency is selected by the VADHPF bitfield. This filter is bypassed when VADHPF=00.

14.3.2.6.2 Input Setting

The HWVAD input has internally a user configurable High-Pass filter (see VADHPF bit description) and a user configurable gain (see VADINPGAIN). Both controls could be used to configure the module based on external noise conditions.

14.3.2.6.3 Frame Time Setting

In order to perform voice detection the signal frames incoming from the CIC filter can be divided in frames whose duration can be configured. The frame time will depend on the CIC output rate, the value of the VADFRAMET bitfield and a constant as is defined in Equation 9 on page 6034.

$$frame\ \ time = \frac{VADFRAMET \times 192}{CIC\ \ output\ \ rate}$$

Equation 9. Frame Time

For instance, in the typical case of the CIC filter output rate would be 192kHz, the frame time would be 10ms when VADFRAMET is 10.

14.3.2.6.4 Initialization Time

An initialization time can be configured by the VADINITT bitfield. During this initialization time no voice detection event is issued, all detectors are kept disabled.

$$initialization\ \ time = (\mathrm{VADINITT}+1) \times frame\ \ time$$

Equation 10. Initialization Time

14.3.2.6.5 Energy Calculation

In case VADFRENDIS = 0, the sum of the absolute values of all the samples of each frame are used to detect voice. Otherwise, if VADFRENDIS=1, the absolute value of the samples are used to detect voice i.e. the data incoming from CIC is not divided in frames.

14.3.2.6.6 Pre-Filter

The Pre-FIlter block is a low-pass filter which can be enabled when VADPREFEN=1 .

14.3.2.6.7 Energy/Envelope-based Detector

To perform voice detection a Noise Filter and Signal Filter are used as shown Figure 14-10. In addition, a minimum block in the output of the Noise Filter and a maximum block in the ouput of the Signal Filter can be used. The result of both blocks can be scaled by gain blocks. Finally, the signal and noise estimated are compared, voice is detected when signal is greater than noise.

14.3.2.6.7.1 Noise Filter

The Noise Filter is a low-pass filter . This filter could be adjusted by the VADNFILADJ bitfield . In case VADNFILAUTO=1 the noise filter is enabled only when voice is not detected, when voice is detected the filter output does not change. In case VADNFILAUTO=0 the filter is always enabled.

A decimation stage in the input of the Noise Filter can be enabled by the VADNDECEN bitfield.

14.3.2.6.7.2 Minimum Noise

In order to estimate the noise, a block which calculates the minimum value between the input and output of the noise filter is used. This block can be enabled by VADNMINEN, otherwise the output of the noise filter is passed directly.

The estimated noise of the output of this block is stored in the VADNDATA register for post-processing via software.

14.3.2.6.7.3 Signal Filter

The Signal Filter is a low-pass filter . This filter will be enabled when VADSFILEN=1, otherwise the filter will be bypassed.

14.3.2.6.7.4 Maximum Signal

In order to estimate the signal, a block which calculates the maximum value between the input and output of the signal filter can be used. This block can be enabled by VADSMAXEN.

14.3.2.6.7.5 Filter Result Gain Setting

There are two gain adjustment controls, one for the estimated noise (VADNGAIN) and another for signal estimated (VADSGAIN). Both gain adjustments outputs are compared, the comparison result determines whether or not there is voice in the channel.

14.3.2.6.8 Zero-Crossing Detector

Optionally, the HWVAD performance can be increased by the use of a Zero-Crossing Detection (ZCD) circuit. It is enabled by a VADZCDEN bit.

The threshold of the ZCD could be static or calculated dynamically depending on the VADZCDAUTO bit. When it is dynamically determined (VADZCDAUTO = 1) the threshold is calculated automatically taking the VADZCDTH bit-field as initial value, otherwise (VADZCDAUTO=0) the threshold is static in the VADZCDTH user configurable value.

The ZCD result can be AND'ed or OR'ed with the energy-based or envelope-based detection, it is selected by the VADZCDAND bit.The filter coefficient is equal to (VADZCDADJ+1)*2^(-4).

14.3.2.6.9 Interrupt

The HWVAD issues an interrupt when voice activity is detected and the VADIE bit in the VAD_CTRL_1 register is enabled. The voice activity event also will set the VADIF flag in the VAD_STAT register, this bit will be cleared when it is written 1.

14.3.2.6.10 Extended Operation

The VADIF bit could be used for a HWVAD extended operation i.e. using the VADNDATA bitfield value to determine a more accurate voice detection event by software. The procedure to follow will be:

1. An interrupt is issued by the HWVAD, the VADIF is set.
2. Leave the VADIF asserted while sofware is processing the VADNDATA bitfield content.
3. When software finishes to process the VADNDATA and detects voice activity:
   - If Envelope-based Detection is used, pulse VADST10 by more than 2-cycles and clear VADIF flag.
   - If Energy-based Detection is used, clear VADIF flag.

14.3.2.6.11 VAD Soft-reset

The VAD soft-reset is requested by writing 1 to the VADRST bit. It resets the following blocks and/or registers:

- VADIF, VADINSATF bitfields.
- All internal VAD buffers.

14.3.2.7 Bus Interface

The Bus Interface manages the Direct Memory Access (DMA) transactions, the interrupt requests and the user register interface with the SoC.

14.3.2.7.1 FIFO

The FIFO is a First-In, First-Out buffer that saves the overall filtering results. Each FIFO is 32 entries length. There is one FIFO for each channel. A push writes data into the FIFO, and a read pops data from FIFO. The push and pop operations from FIFO are in different clock domains, the push operation is made using the MICFIL clock while the pop operation uses the CPU clock . The FIFO has a configurable watermark FIFOWMK in the FIFO_CTRL register. When the amount of data in all the FIFOs of the enabled channels is greater than the watermark value (FIFO_CTRL[FIFOWMK]) a DMA/IRQ is requested depending on the DISEL field in the CTRL_1 register. After the filters are enabled by PDMIEN, the initial FIR results are discarded and not written into the FIFOs.

About 68 initial FIR results are not stored on FIFOs (FIR_RDY=0). After the STAT[FIR_RDY] register is set the data generated by the filter is valid and starts to push the results in FIFO.

A new filtered result is discarded (not stored) when the FIFO is full and overflow flag FIFOOVFn rises. Similarly, when trying a pop and FIFO is empty no valid data is read and underflow flag rises. In both cases, when overflow or underflow occurs they are flagged in the FIFO_STAT register to the respective channel, and an error interrupt is generated if the ERREN in the CTRL_1 register is asserted.

When the FIFO is reset by asserting the SRES soft-reset bit, zero values fills all FIFO positions, as well as push and pop pointers are set to initial position.

14.3.2.7.2 DMA

If DISEL=01 the FIFO stored samples will be read through DMA transactions. Then if there is at least one enabled channel and all FIFOs of all enabled channels reach their watermark the output interface makes a request for DMA transaction. When the PDMIEN is set and the filters start to operate, there is a period to wait the filter results to stabilize and no filter results are stored on the FIFOs. Therefore the first DMA request takes longer to occur than the following ones.

The CHnF flags in the STAT register are cleared when a DMA transaction is finished.

When the MICFIL is in Low Power mode and the watermark level is surpassed an asynchronous DMA request is delivered to the system, then the system wakes up the CPU clock, returning to Normal mode, to read FIFOs data.

14.3.2.7.3 Interrupt Requests

If DISEL=10 then an interrupt request is used to indicate that filtering results are stored in the FIFOs to be read by CPU. Also in this case the data can be read in the DATACHn registers and the flag CHnF of the STAT register are asserted for the enabled channels when the CHn FIFO surpasses the watermark level.

When the module is in Low Power mode the CPU clock is disabled, but the MICFIL clock remains enabled. In this case, an asynchronous interrupt is delivered to the system, it wakes up the CPU clock, returning to Normal mode. The system could read the FIFOs data and after should clear the CHnF flags to get a new interrupt request.

An error interrupt request can be generated also when an exceptional condition happens, such as a overflow or underflow in any channel output, an overflow or underflow of any FIFO, or the MICFIL clock rate is not the minimum value required to the MICFIL working (see Minimum Required CLKDIV). This one is enabled by the ERREN bit and

the type of error is signalized by its respective status flag: OUTOVFn and OUTUNFn for channel outputs errors, FIFOOVF and FIFOUND flags for FIFO errors, and LOWFREQF for MICFIL clock rate.

14.3.3 External Signals

Note

The PDM Microphone Interface module does not provide metastability protection or filtering for input data. The data must be generated using the provided PDM_CLK.

Table 14-9. PDM Microphone Interface External Signals

Signal	Description	Direction
PDM_DATAn	The data bitstreams received from *n*th PDM microphone pair throughout the input interface	I
PDM_CLK	The clock delivered to the PDM microphones from the clock generator	O

14.3.4 Initialization

Following are some simple initialization steps to be done before using the MICFIL block.

14.3.4.1 Initialization Procedure without HWVAD

The sequence of steps for the block initialization without enabling voice activity detector, i.e. using decimation filters only, is as follows:

1. Program the bit-fields as desired for your application (CLKDIV, CHnEN, FIFOWMK, DISEL, ERREN etc.). For more details see MICFIL_CLK_ROOT rate and CLKDIV calculation examples.
2. Start the overall module asserting the PDMIEN bit.
3. The module is ready to receive data from the input ports of the enabled channels.

14.3.4.2 Initialization Procedure with HWVAD

In this section is described the initialization procedures to initialize MICFIL with the voice activity detector.

The voice activity detector can work in two modes: Energy-based mode or Envelope-based mode. Optionally, a Zero-Crossing Detector can be enabled to improve the voice detection.

The Energy-based method separates the audio signal in frames and estimates the energy of these frames to detect voice activity. In the other hand, the Envelope-based method calculates the audio signal envelope to detect voice.

Both methods have the same detection performance, but the Energy-based is recommended for low power applications where is expected that the chip is asleep for long periods. Envelope-based detection is recommended for applications where the device require to wake faster and it is not asleep for long periods.

The Zero-Crossing Detector is recommended to improve detection with fricatives phonemes (/e.g. /f/ or /s/) in noisy environments.

14.3.4.2.1 HWVAD in Energy-based mode

The sequence of steps to initialize the voice activity detector in Energy-based mode is as follows:

1. Program the control bitfields as desired for your application (CLKDIV, CHnEN, FIFOWMK, DISEL, ERREN etc.). For more details see MICFIL_CLK_ROOT rate and CLKDIV calculation examples.
2. Configure general HWVAD parameters:
   a. CIC oversampling rate (VADCICOSR bitfield).
   b. Configure the voice source channel (VADCHSEL bitfield).
   c. Configure the internal initialization timer (VADINITT bitfield) if required. More details in Initialization Time.
   d. Assert VADEN bitfield.
3. Configure input, noise and signal gains according to the environmental noise conditions:
   a. Configure the input gain (VADINPGAIN bitfield).
   b. Configure the signal gain (VADSGAIN bitfield).
   c. Configure the noise gain (VADNGAIN bitfield).
4. Enable the high-pass filter if required (VADHPF bitfield).
5. Configure the VADNFILADJ bitfield according to environmental noise conditions (VADNFILADJ = 16 is recommended).
6. Put HWVAD in energy-based mode (default values):
   a. Keep the VADFRENDIS bitfield cleared.
   b. Keep the VADPREFEN bitfield cleared.
   c. Keep the VADSFILEN bitfield cleared.
   d. Keep the VADSMAXEN bitfield cleared.

e. Keep the VADNFILAUTO bitfield asserted.
f. Keep the VADNMINEN bitfield cleared.
g. Keep the VADNDECEN bitfield cleared.
h. Keep the VADNOREN bitfield cleared.

7. Initialize the Zero-Crossing Detector (Zero-Crossing Detector Initialization) if required.
8. Enable interrupts if required (VADIE and VADERIE bitfields).
9. Start the overall module asserting the PDMIEN bit.
10. The module is ready to receive data from the input ports of the enabled channels and to trigger interrupts when voice is detected.

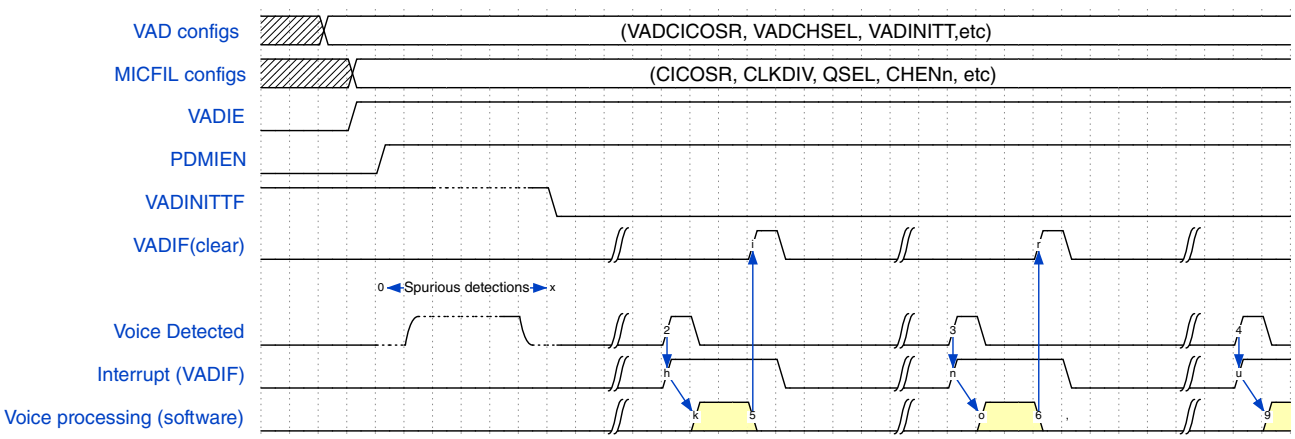


Figure 14-11. Energy-based initialization sequence

14.3.4.2.2 HWVAD in Envelope-based mode

The sequence of steps to initialize MICIL with the voice activity detector in Envelope-based mode is as follows:

1. Program the control bitfields as desired for your application (CLKDIV, CHnEN, FIFOWMK, DISEL, ERREN etc.). For more details see MICFIL_CLK_ROOT rate and CLKDIV calculation examples.
2. Configure general HWVAD parameters:
   a. CIC oversampling rate (VADCICOSR bitfield).
   b. Configure the voice source channel (VADCHSEL bitfield).
   c. Configure the internal initialization timer (VADINITT bitfield) if required. More details in Initialization Time.
   d. Assert VADEN bitfield.
3. Configure input, noise and signal gains according the environment noise conditions:
   a. Configure the input gain (VADINPGAIN bitfield).
   b. Configure the signal gain (VADSGAIN bitfield).
   c. Configure the noise gain (VADNGAIN bitfield).
4. Enable the high-pass filter if required (VADHPF bitfield).

5. Put HWVAD in envelope-based mode:
   a. Keep VADNFILADJ bitfield cleared.
   b. Assert the VADFRENDIS bitfield.
   c. Assert the VADPREFEN bitfield.
   d. Assert the VADSFILEN bitfield.
   e. Assert the VADSMAXEN bitfield.
   f. Clear the VADNFILAUTO bitfield.
   g. Assert the VADNMINEN bitfield.
   h. Assert the VADNDECEN bitfield.
   i. Assert the VADNOREN bitfield.
6. Initialize the Zero-Crossing Detector (Zero-Crossing Detector Initialization) if required.
7. Start the overall module asserting the PDMIEN bit.
8. Wait for VADINITF to be cleared.
9. Write VADST10 bitfield three times.
10. Enable interrupts if required (VADIE and VADERIE bitfields).
11. The module is ready to receive data from the input ports of the enabled channels and to trigger interrupts when voice is detected.

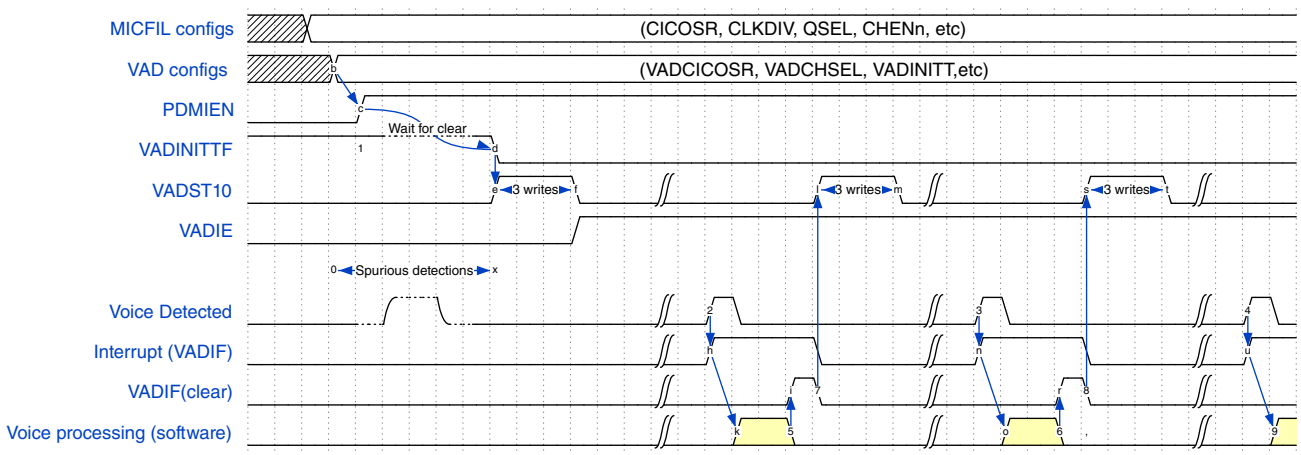


Figure 14-12. Envelope-based initialization sequence

14.3.4.2.3 Zero-Crossing Detector Initialization

The sequence of steps to initialize the Zero-Crossing Detector in Automatic mode is as follows:

1. Configure VADZCDADJ bitfield.
2. Configure an initial threshold in VADZCDTH according to the environmental noise conditions if required.
3. Configure VADZCDAND bitfield.
4. Keep VADZCDAUTO bitfield asserted.

5. Assert VADZCDEN.

The sequence of steps to initialize the Zero-Crossing Detector in Manual mode is as follows:

1. Configure a threshold in VADZCDTH according to the environmental noise conditions if required.
2. Configure VADZCDAND bitfield.
3. Clear VADZCDAUTO bitfield.
4. Assert VADZCDEN.

14.3.5 Application Information

14.3.5.1 Reconfiguration Procedure

The sequence of steps for the block reconfiguration is as follows:

1. Deassert PDMIEN bit.
2. Wait to BSY_FIL to be cleared.
3. Run a soft-reset cycle.
4. Run a initilialization procedure as described in Initialization Procedure without HWVAD or Initialization Procedure with HWVAD.

14.3.6 MICFIL register descriptions

This section provides the memory map and detailed description of all MICFIL registers. Any access to reserved areas can issue a slave bus error indication.

The registers allocated in this memory map are implemented for up to 4 pairs of microphones, that means 8 channels are available.

14.3.6.1 MICFIL memory map

MICFIL base address: 30CA_0000h

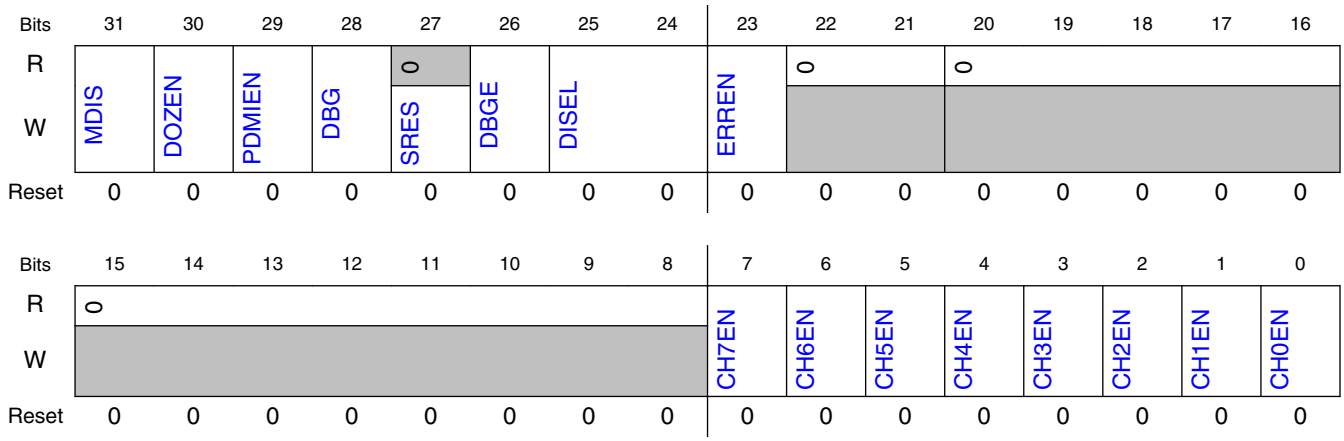
Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	MICFIL Control register 1 (CTRL_1)	32	RW	0000_0000
4	MICFIL Control register 2 (CTRL_2)	32	RW	0000_0000
8	MICFIL Status register (STAT)	32	W1C	0000_0000
10	MICFIL FIFO Control register (FIFO_CTRL)	32	RW	0000_001F
14	MICFIL FIFO Status register (FIFO_STAT)	32	W1C	0000_0000
24 - 40	MICFIL Output Result Register (DATACH0 - DATACH7)	32	RO	0000_0000
64	MICFIL DC Remover Control register (DC_CTRL)	32	RW	0000_0000
74	MICFIL Range Control register (RANGE_CTRL)	32	RW	0000_0000
7C	MICFIL Range Status register (RANGE_STAT)	32	W1C	0000_0000
90	Voice Activity Detector 0 Control register (VAD0_CTRL_1)	32	RW	0000_0000
94	Voice Activity Detector 0 Control register (VAD0_CTRL_2)	32	RW	000A_0000
98	Voice Activity Detector 0 Status register (VAD0_STAT)	32	W1C	8000_0000
9C	Voice Activity Detector 0 Signal Configuration (VAD0_SCONFIG)	32	RW	0000_0000
A0	Voice Activity Detector 0 Noise Configuration (VAD0_NCONFIG)	32	RW	8000_0000
A4	Voice Activity Detector 0 Noise Data (VAD0_NDATA)	32	RO	0000_0000
A8	Voice Activity Detector 0 Zero-Crossing Detector (VAD0_ZCD)	32	RW	0000_0004

14.3.6.2 MICFIL Control register 1 (CTRL_1)

14.3.6.2.1 Offset

Register	Offset
CTRL_1	0h

14.3.6.2.2 Diagram



14.3.6.2.3 Fields

Field	Description
31 MDIS	Module Disable Puts the whole MICFIL in Disable/Low Leakage mode. The generated PDM_CLK is stopped and consequently no data is generated by the microphones. **NOTE:** When MDIS=1, writes to CTRL_1 and CTRL_2 only are allowed with the exception of writes to the DBG and SRES bits. A slave bus error indication will be issued when any register except CTRL_1 and CTRL_2 is written when MDIS=1. 0 - Normal Mode 1 - Disable/Low Leakage Mode
30 DOZEN	DOZE enable When SoC is in Doze mode and DOZEN is active the module enters to Disable/Low Leakage mode.
29 PDMIEN	MICFIL Enable Starts the module operation. Only the channels enabled by CHnEN or VAD enabled by VADEN will start to operate. When it is cleared, the whole module is stopped after to finish the pending filtering. **NOTE:** This field must be the latest to be enabled, for more information refer to Initialization . 0 - MICFIL stopped 1 - MICFIL operation started
28 DBG	Debug Mode Enables the Debug mode. The module can run on this mode depending on the DBGE bit value. **NOTE:** This bit is writable only when MDIS is cleared. 0 - Normal Mode 1 - Debug Mode
27 SRES	Software-reset bit Provides the CPU with the capability to initialize the MICFIL module through the slave-bus interface. This bit always reads as zero, and this bit is only effective when MDIS is cleared. This bit is self-cleared.

Table continues on the next page...

Field	Description
	NOTE: This bit must be asserted only when BSY_FIL=0 This bit can be used to clear all data in the filters. It resets the following blocks and/or registers: • All internal buffers in decimation filters and Voice Activity Detectors. • All FIFOs. • All CHnF flags and LOWFREQF flag on the STAT register, and the FIFO_STAT register. • Time Generator. 0 - No action 1 - Software reset
26 DBGE	Module Enable in Debug Enables operation in Debug mode. 0 - Disabled after completing the current frame 1 - Enabled
25-24 DISEL	DMA Interrupt Selection Selects if the module performs DMA requests or filter interrupt requests when the number of FIFO elements surpasses the configured watermark. Note that requests are only generated from the channels whose CHnEN bits are asserted. **NOTE:** This bit mustn't change when BSY_FIL bit is asserted 00 - DMA and interrupt requests disabled 01 - DMA requests enabled 10 - Interrupt requests enabled 11 - Reserved
23 ERREN	Error Interruption Enable Enables error interrupts (exceptions). This error could be caused by: • an overflow or underflow in the FIFOs (FIFOOVF or FIFOUND bits), • an overflow or underflow in the decimations filters (the RANGEOVF or RANGEUNF bits), or • not enough MICFIL internal clock frequency (LOWFREQF bit). **NOTE:** This bit mustn't change when BSY_FIL bit is asserted 0 - Error Interrupts disabled 1 - Error Interrupts enabled
22-21 —	Reserved
20-16 —	Reserved
15-8 —	Reserved
7-0 CHnEN	Channel n Enable Enables the decimation filter channel n. Refer to Initialization for more information. **NOTE:** This bit mustn't change when BSY_FIL bit is asserted.

14.3.6.3 MICFIL Control register 2 (CTRL_2)

14.3.6.3.1 Offset

Register	Offset
CTRL_2	4h

14.3.6.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				QSEL				0				CICOSR			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CLKDIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.6.3.3 Fields

Field	Description
31-28 —	Reserved
27-25 QSEL	Quality Mode Defines the actual quality mode of the decimation filter. **NOTE:** This bitfield mustn't change when BSY_FIL bit is asserted. 000 - Medium quality mode 001 - High quality mode 100 - Very low quality 2 mode 101 - Very low quality 1 mode 110 - Very low quality 0 mode 111 - Low quality mode
24-20 —	Reserved
19-16 CICOSR	CIC Decimation Rate Helps to configure the Equation 8 on page 5984.

Table continues on the next page...

Field	Description
	NOTE: This bitfield mustn't change when BSY_FIL bit is asserted.
15-8 —	Reserved
7-0 CLKDIV	Clock Divider Configures the internal clock divider value. **NOTE:** This bitfield mustn't change when BSY_FIL bit is asserted. **NOTE:** CLKDIV=0 has the same effect than CLKDIV=1. CLKDIV=0 or 1 has the same effect than CLKDIV=2 during Medium Quality mode. CLKDIV=0 or 1 is not allowed during High Quality mode.

14.3.6.4 MICFIL Status register (STAT)

14.3.6.4.1 Offset

Register	Offset
STAT	8h

14.3.6.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BSY_FIL	FIR_RDY	LOWFREQF	0												
W			W1C													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CH7F	CH6F	CH5F	CH4F	CH3F	CH2F	CH1F	CH0F
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.6.4.3 Fields

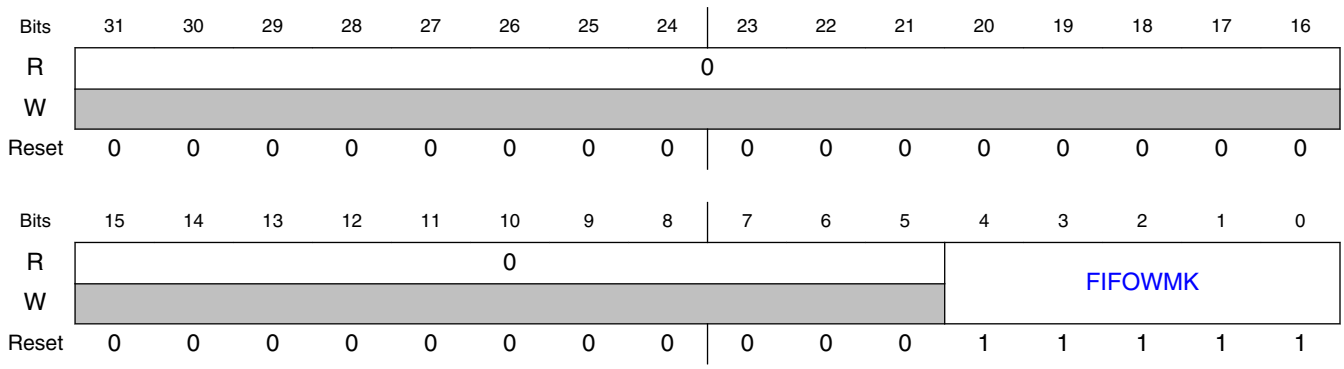
Field	Description
31 BSY_FIL	Busy Flag Signalizes when at least a decimation filter channel or at least a HWVAD is running and the PDM_CLK is being generated. This bit could be used as a confirmation that all decimation filters are stopped in a transition to Debug (when DBGE=0) or Disable/Low Leakage mode. 0 - MICFIL is stopped 1 - MICFIL is running
30 FIR_RDY	Filter Data Ready Indicates that all the decimation filter buffers have been updated, and all data provided from now on will be reliable. 0 - Filter data is not reliable 1 - Filter data is reliable
29 LOWFREQF	Low Frequency Flag Indicates that the CLKDIV bitfield value is not high enough to filter all enabled channels , and the half band filters didn't finish their operations before the next sample arrived. See Minimum Required CLKDIV . Write a logic 1 to this field to clear this flag. **NOTE:** The filtering results in this condition are unpredictable. 0 - CLKDIV value is OK 1 - CLKDIV value is too low
28-16 —	Reserved
15-8 —	Reserved
7-0 CHnF	Channel n Output Data Flag Indicates that the FIFO from this channel has surpassed its watermark value and the data is available at the DATACHn register. This flag contributes to generate an Interrupt or DMA request if enabled by DISEL in the CTRL_1. Write a logic 1 to this field to clear this flag. If DMA request is set (DISEL = 01) the CHnF flag will be set according to watermark level and will be cleared , in the other cases (IRQ included) the CHnF flag will be set according to the watermark level and will be cleared with the w1c functionality of the bit. **NOTE:** If DMA is enabled, this flag is self-cleared when the DMA transaction is finished . 0 - Channel's FIFO did not reach the number of elements configured in watermark bit-field 1 - Channel's FIFO reached the number of elements configured in watermark bit-field

14.3.6.5 MICFIL FIFO Control register (FIFO_CTRL)

14.3.6.5.1 Offset

Register	Offset
FIFO_CTRL	10h

14.3.6.5.2 Diagram



14.3.6.5.3 Fields

Field	Description
31-5 —	Reserved
4-0 FIFOWMK	FIFO Watermark Control Controls the watermark of the FIFO used to set CHnF. When the number of results in the FIFO is greater than FIFOWMK value then the CHnF flag is set. A DMA request or interrupt can be also generated according to DISEL bit-field. **NOTE:** This mustn't change when BSY_FIL is asserted. Before changing the watermark value service all filtered data and clear channel flags.

14.3.6.6 MICFIL FIFO Status register (FIFO_STAT)

14.3.6.6.1 Offset

Register	Offset
FIFO_STAT	14h

14.3.6.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFOUND7	FIFOUND6	FIFOUND5	FIFOUND4	FIFOUND3	FIFOUND2	FIFOUND1	FIFOUND0	FIFOOVF7	FIFOOVF6	FIFOOVF5	FIFOOVF4	FIFOOVF3	FIFOOVF2	FIFOOVF1	FIFOOVF0
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.6.6.3 Fields

Field	Description
31-16 —	Reserved
15-8 FIFOUNDn	FIFO Underflow Exception flag for Channel n Indicates an exceptional underflow condition in the FIFO. If the FIFO is empty and a read operation is requested then this flag is set. Write a logic 1 to this field to clear this flag. 0 - No exception by FIFO Underflow 1 - Exception by FIFO underflow
7-0 FIFOOVFn	FIFO Overflow Exception flag for Channel n Indicates an exceptional overflow condition in the FIFO. If the FIFO is full and an additional result is received to be written then this flag is set. Write a logic 1 to this field to clear this flag. 0 - No exception by FIFO overflow 1 - Exception by FIFO overflow

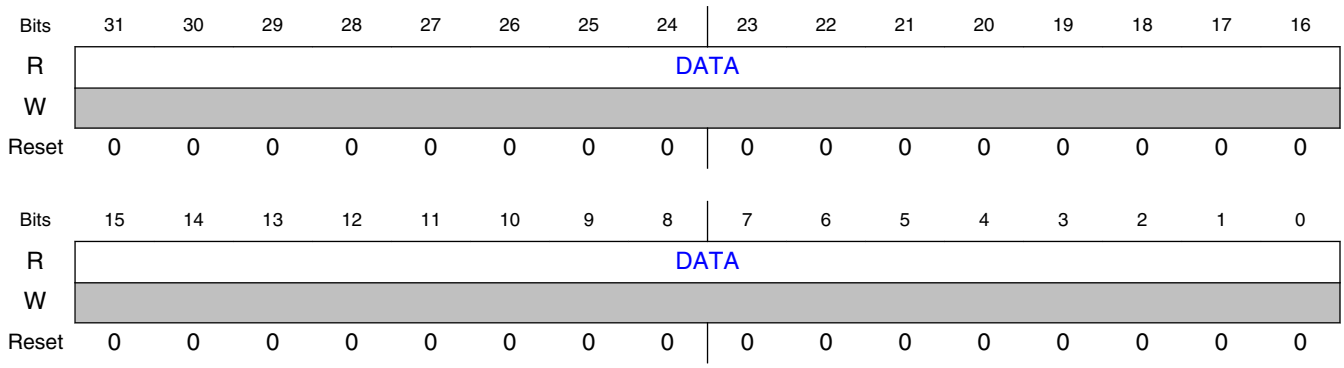
14.3.6.7 MICFIL Output Result Register (DATACH0 - DATACH7)

14.3.6.7.1 Offset

For a = 0 to 7:

Register	Offset
DATACHa	24h + (a × 4h)

14.3.6.7.2 Diagram



14.3.6.7.3 Fields

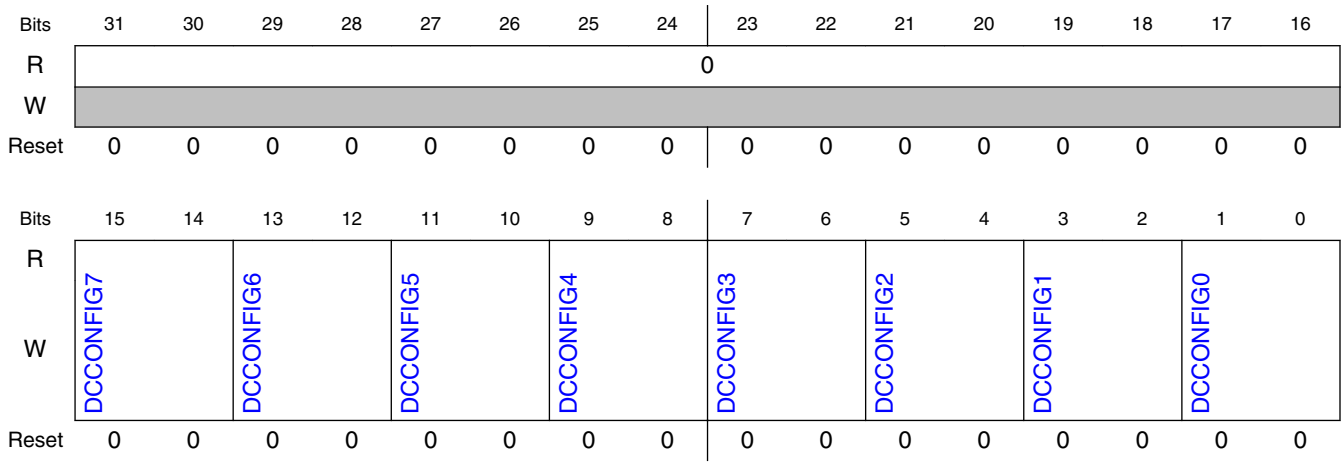
Field	Description
31-0 DATA	Channel n Data Reflects the word value stored at the top of the Channel's n FIFO. The samples are integer signed numbers in two's complement format. **NOTE:** Even FIFO data width is 32-bits, only the 24 more significative bits have information, the other bits are always zero.

14.3.6.8 MICFIL DC Remover Control register (DC_CTRL)

14.3.6.8.1 Offset

Register	Offset
DC_CTRL	64h

14.3.6.8.2 Diagram



14.3.6.8.3 Fields

Field	Description
31-16 —	Reserved
15-14: DCCONFIG7 13-12: DCCONFIG6 11-10: DCCONFIG5 9-8: DCCONFIG4 7-6: DCCONFIG3 5-4: DCCONFIG2 3-2: DCCONFIG1 1-0: DCCONFIG0	Channel n DC Remover Configuration Defines the value of the cut-off frequency of the DC Remover. 00 - DC Remover cut-off at 21Hz 01 - DC Remover cut-off at 83Hz 10 - DC Remover cut-off at 152Hz 11 - DC Remover is bypassed

14.3.6.9 MICFIL Range Control register (RANGE_CTRL)

14.3.6.9.1 Offset

Register	Offset
RANGE_CTRL	74h

14.3.6.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RANGEADJ7				RANGEADJ6				RANGEADJ5				RANGEADJ4			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RANGEADJ3				RANGEADJ2				RANGEADJ1				RANGEADJ0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.6.9.3 Fields

Field	Description
31-28: RANGEADJ7 27-24: RANGEADJ6 23-20: RANGEADJ5 19-16: RANGEADJ4 15-12: RANGEADJ3 11-8: RANGEADJ2 7-4: RANGEADJ1 3-0: RANGEADJ0	Channel n Range Adjustment Adjusts the dynamic range of the CIC filter. It is a unsigned positive value. This bitfield must be set within the following ranges: If QSEL in [HQ,LQ]: $RANGEADJn <= 27 - \text{ceil}(7\log_2(32 - 2CICOSR))$ If QSEL in [MQ,VLQ1]: $RANGEADJn <= 27 - \text{ceil}(7\log_2(16 - CICOSR))$ If QSEL in [LQ,VLQ2]: $RANGEADJn <= 26 - \text{ceil}(7\log_2(16 - CICOSR))$ where OSR is the Equation 1 on page 5977 and the ceiling function, ceil(x), maps to the least integer greater than or equal to x.

14.3.6.10 MICFIL Range Status register (RANGE_STAT)

14.3.6.10.1 Offset

Register	Offset
RANGE_STAT	7Ch

14.3.6.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								RANGEUNF7	RANGEUNF6	RANGEUNF5	RANGEUNF4	RANGEUNF3	RANGEUNF2	RANGEUNF1	RANGEUNF0
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RANGEOVF7	RANGEOVF6	RANGEOVF5	RANGEOVF4	RANGEOVF3	RANGEOVF2	RANGEOVF1	RANGEOVF0
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.6.10.3 Fields

Field	Description
31-24 —	Reserved
23-16 RANGEUNFn	Channel n Range Underflow Error Flag Indicates an exceptional underflow condition in MICFIL range. **NOTE:** If this error flag is asserted, channel n data is not reliable. Require to adjust the RANGEADJn value until this flag is not asserted anymore. 0 - No exception by range underflow 1 - Exception by range underflow
15-8 —	Reserved

Table continues on the next page...

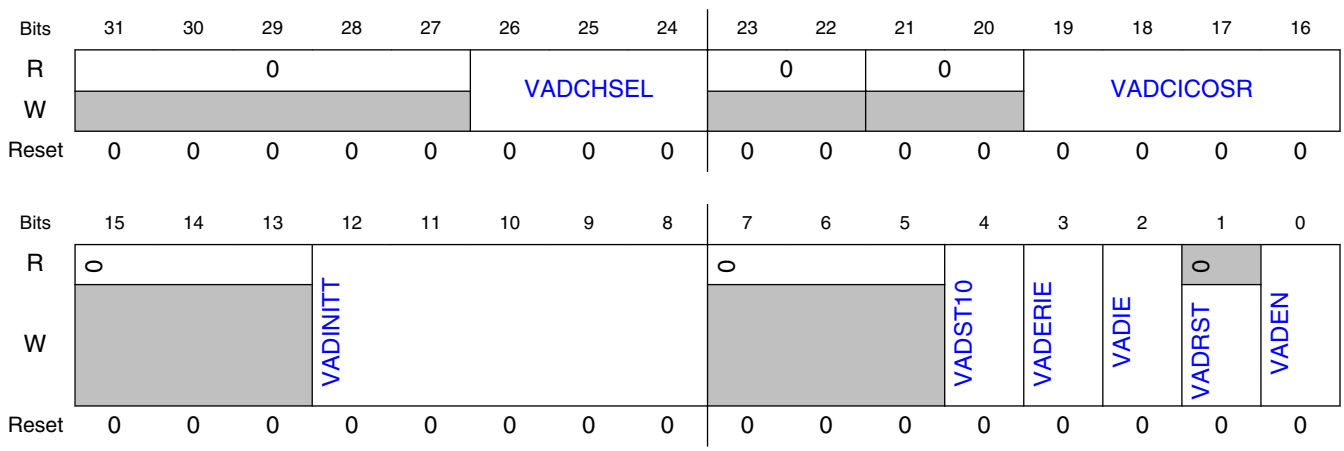
Field	Description
7-0 RANGEOVFn	Channel n Range Overflow Error Flag Indicates an exceptional overflow condition in MICFIL range. **NOTE:** If this error flag is asserted, channel n data is not reliable. Require to adjust the RANGEADJn value until this flag is not asserted anymore. 0 - No exception by range overflow 1 - Exception by range overflow

14.3.6.11 Voice Activity Detector 0 Control register (VAD0_CTRL_1)

14.3.6.11.1 Offset

Register	Offset
VAD0_CTRL_1	90h

14.3.6.11.2 Diagram



14.3.6.11.3 Fields

Field	Description
31-27 —	Reserved
26-24	Voice Activity Detector Channel Selector

Table continues on the next page...

Field	Description
VADCHSEL	Selects the number of channel which the hardware voice activity detector will work. **NOTE:** This bitfield mustn't change when BSY_FIL is asserted.
23-22 —	Reserved
21-20 —	Reserved
19-16 VADCICOSR	Voice Activity Detector CIC Oversampling Rate Defines the oversampling rate of the CIC filter. The oversampling rate is (16-CICOSR). **NOTE:** This bitfield mustn't change when BSY_FIL is asserted.
15-13 —	Reserved
12-8 VADINITT	Voice Activity Detector Initialization Time Selects the number of frames to be used to initialize the voice detection. During this period the output of the voice activity detector is forced to inactive. For more information, refer to Initialization Time . **NOTE:** The initialization time will be the time taken by [VADINITT+1] frames. **NOTE:** This bitfield mustn't change when BSY_FIL is asserted.
7-5 —	Reserved
4 VADST10	Voice Activity Detector Internal Filters Initialization Initializes the signal and noise filter with pre-filter value. **NOTE:** If the use of this bit is required, it should be stay pulsed by more than 2 cycles of system clock . **NOTE:** There is not voice activity detection when VADST10 is asserted. 0 - Normal operation. 1 - Filters are initialized.
3 VADERIE	Voice Activity Detector Error Interruption Enable Enables Error Interrupts in the MICFIL. This error could be caused by an overflow in the input of the HWVAD and it is signalized by the VADINSATF flag. 0 - HWVAD Error Interrupts disabled 1 - HWVAD Error Interrupts enabled
2 VADIE	Voice Activity Detector Interruption Enable Enables interrupts in the MICFIL when voice activity event has been detected by the Hardware Voice Activity Detector (HWVAD). See the VADIF for more information. 0 - HWVAD Interrupts disabled 1 - HWVAD Interrupts enabled
1 VADRST	Voice Activity Detector Reset Provides the CPU with the capability to initialize the VAD through the slave-bus interface. This bit always reads as zero. This bit is self-cleared. See VAD Soft-reset.
0 VADEN	Voice Activity Detector Enable Enables the use of the Hardware Voice Activity Detector. **NOTE:** This bitfield mustn't change when BSY_FIL is asserted.

Field	Description
	0 - The HWVAD is disabled 1 - The HWVAD is enabled

14.3.6.12 Voice Activity Detector 0 Control register (VAD0_CTRL_2)

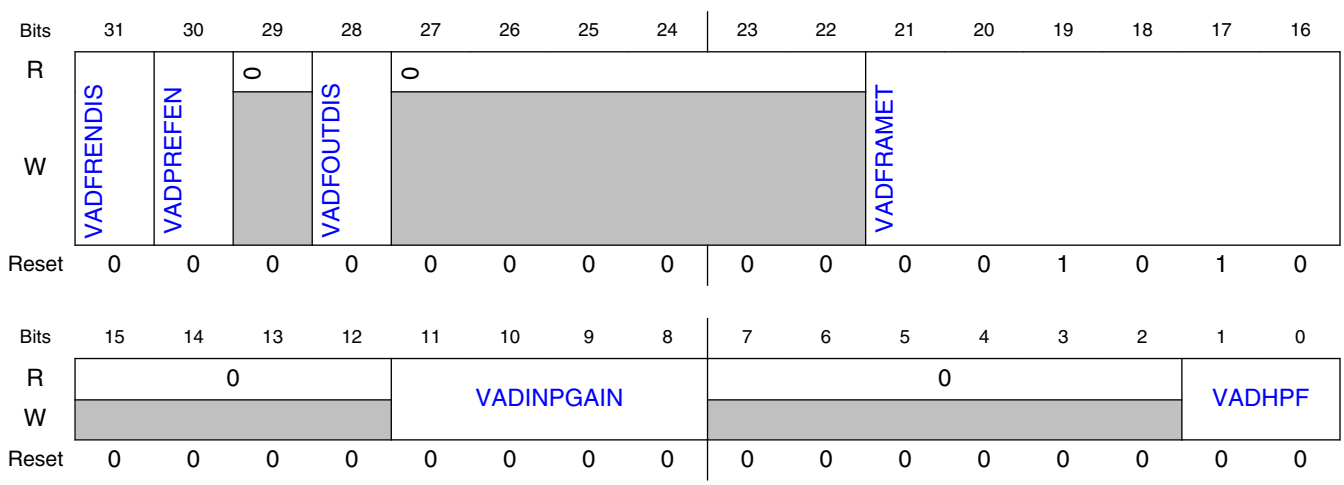
NOTE

This register mustn't change when BSY_FIL is asserted.

14.3.6.12.1 Offset

Register	Offset
VAD0_CTRL_2	94h

14.3.6.12.2 Diagram



14.3.6.12.3 Fields

Field	Description
31 VADFRENDIS	Voice Activity Detector Frame Energy Disable Disables the calculus of energy in a frame. For more information, refer to Noise Filter . 0 - Frame energy calculus enabled.

Table continues on the next page...

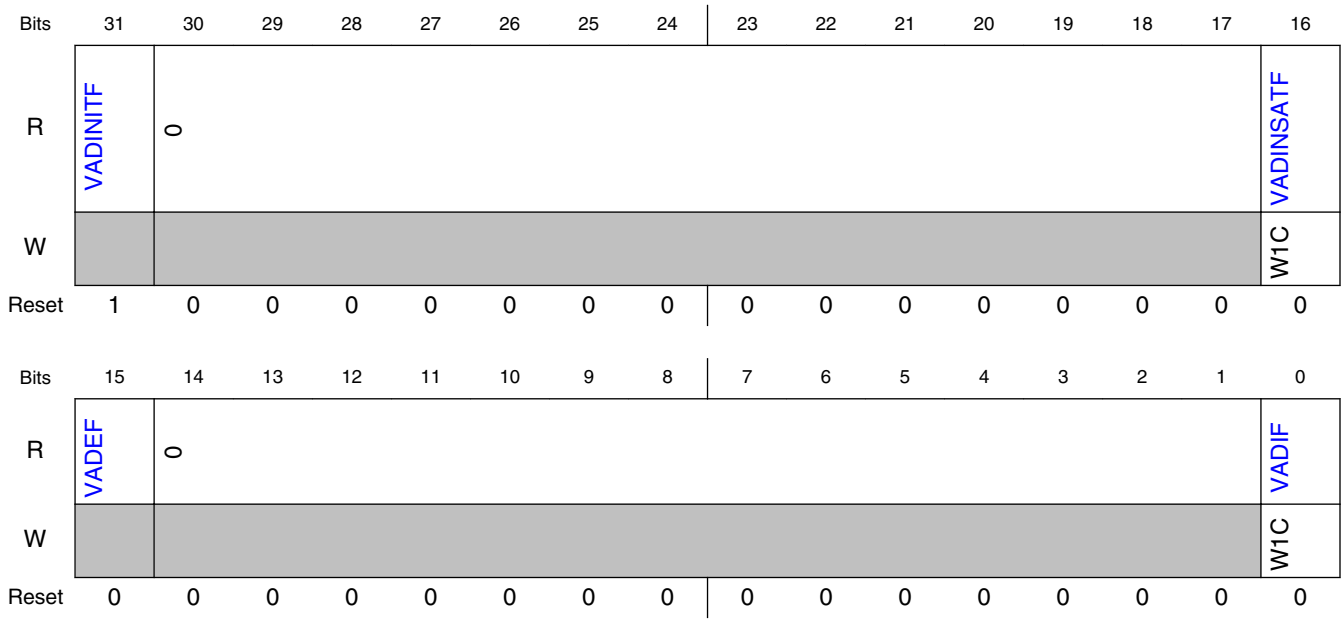
Field	Description
	1 - Frame energy calculus disabled.
30 VADPREFEN	Voice Activity Detector Pre Filter Enable Enables a pre-filter in the input of Noise and Signal filters. For more information, refer to Pre-Filter . 0 - Pre-filter is bypassed. 1 - Pre-filter is enabled.
29 —	Reserved
28 VADFOUTDIS	Voice Activity Detector Force Output Disable Disables the output of the Energy/Envelope-based Detector. 0 - Output is enabled. 1 - Output is disabled.
27-22 —	Reserved
21-16 VADFRAMET	Voice Activity Detector Frame Time Selects the scale value to control the frame time duration as is described in Frame Time Setting . **NOTE:** VADFRAMET=0 is not allowed.
15-12 —	Reserved
11-8 VADINPGAIN	Voice Activity Detector Input Gain Configures the gain value in the HWVAD input as a positive or negative (two's complement) number of bits to shift.
7-2 —	Reserved
1-0 VADHPF	Voice Activity Detector High-Pass Filter Configures the internal high-pass filter. 00 - Filter bypassed. 01 - Cut-off frequency at 1750Hz. 10 - Cut-off frequency at 215Hz. 11 - Cut-off frequency at 102Hz.

14.3.6.13 Voice Activity Detector 0 Status register (VAD0_STAT)

14.3.6.13.1 Offset

Register	Offset
VAD0_STAT	98h

14.3.6.13.2 Diagram



14.3.6.13.3 Fields

Field	Description
31 VADINITF	Voice Activity Detector Initialization Flag Signalizes when the HWVAD is being initialized according to the VADINITT bitfield value. 0 - HWVAD is not being initialized. 1 - HWVAD is being initialized.
30-17 —	Reserved
16 VADINSATF	Voice Activity Detector Input Saturation Flag Indicates an exceptional saturation condition by an overflow or underflow in the HWVAD input. 0 - No exception 1 - Exception
15 VADEF	Voice Activity Detector Event Flag Indicates when voice activity event has been detected. 0 - Voice activity not detected 1 - Voice activity detected
14-1 —	Reserved
0	Voice Activity Detector Interrupt Flag

Field	Description
VADIF	Indicates when voice activity has been detected by the HWVAD. Write a logic 1 to this field to clear this flag. 0 - Voice activity not detected 1 - Voice activity detected

14.3.6.14 Voice Activity Detector 0 Signal Configuration (VAD0_SCONFIG)

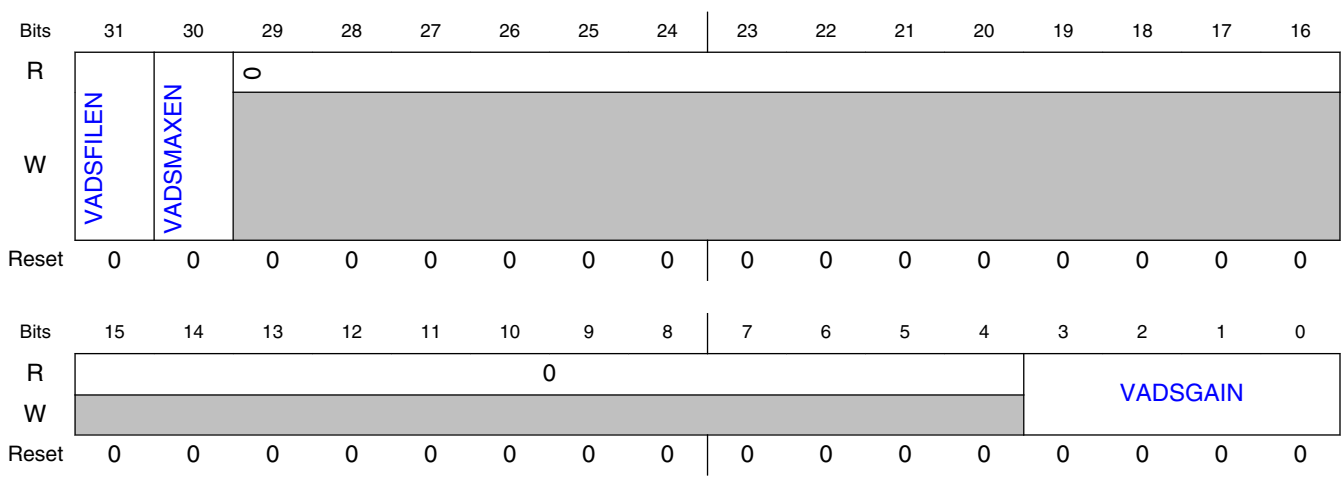
NOTE

These bit-fields must be configured when the MICFIL is not running.

14.3.6.14.1 Offset

Register	Offset
VAD0_SCONFIG	9Ch

14.3.6.14.2 Diagram



14.3.6.14.3 Fields

Field	Description
31 VADSFILEN	Voice Activity Detector Signal Filter Enable Enables the signal filter. For more information, refer to Signal Filter . 0 - Signal filter is disabled. 1 - Signal filter is enabled.
30 VADSMAXEN	Voice Activity Detector Signal Maximum Enable Selects whether or not a block to calculate the maximum value between the input and output of signal filter is used as signal estimation. For more information, refer to Signal Filter . 0 - Maximum block is bypassed. 1 - Maximum block is enabled.
29-4 —	Reserved
3-0 VADSGAIN	Voice Activity Detector Signal Gain Configures the gain value for the signal energy or envelope estimated. It is a unsigned value which is multiplied with the estimated signal data. **NOTE:** VADSGAIN=0 gain corresponds to 1.

14.3.6.15 Voice Activity Detector 0 Noise Configuration (VAD0_NCONFIG)

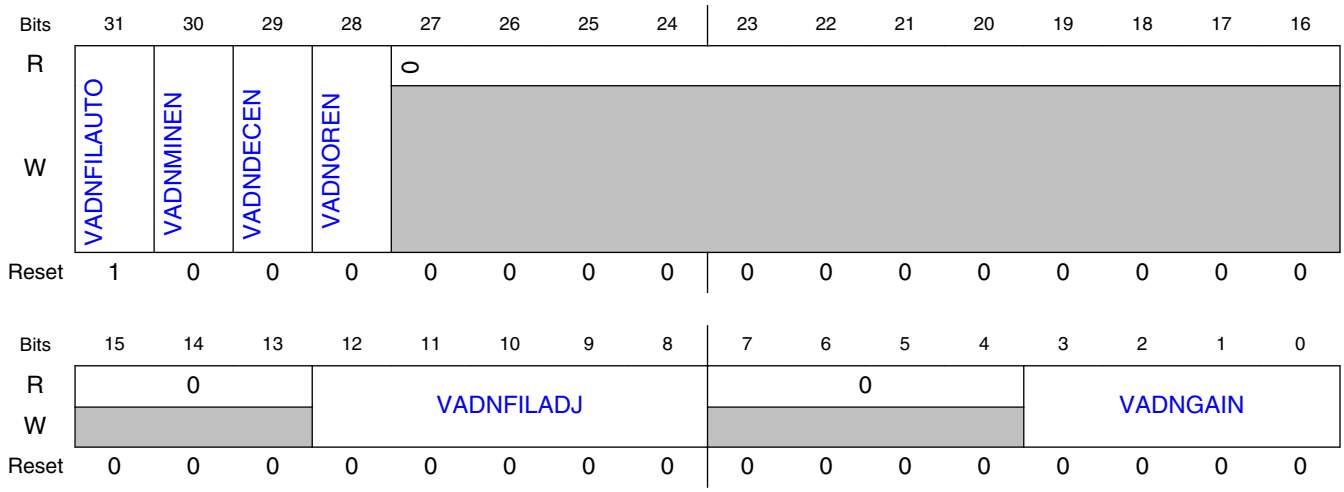
NOTE

These bit-fields must be configured when the MICFIL is not running.

14.3.6.15.1 Offset

Register	Offset
VAD0_NCONFIG	A0h

14.3.6.15.2 Diagram



14.3.6.15.3 Fields

Field	Description
31 VADNFILAUTO	Voice Activity Detector Noise Filter Auto Selects whether or not the noise filter is activated automatically based in voice activity information. For more information, refer to Noise Filter . 0 - Noise filter is always enabled. 1 - Noise filter is enabled/disabled based on voice activity information.
30 VADNMINEN	Voice Activity Detector Noise Minimum Enable Selects whether or not a block to calculate the minimum value between the input and output of noise filter is used as noise estimation. For more information, refer to Noise Filter . 0 - Minimum block is bypassed. 1 - Minimum block is enabled.
29 VADNDECEN	Voice Activity Detector Noise Decimation Enable Selects whether or not the input of the noise filter is decimated. For more information, refer to Noise Filter . 0 - Noise input is not decimated. 1 - Noise input is decimated.
28 VADNOREN	Voice Activity Detector Noise OR Enable Enables a OR logic in the output of minimum noise estimator block. For more information, refer to Filter Result Gain Setting . 0 - Noise input is not decimated. 1 - Noise input is decimated.
27-13 —	Reserved
12-8	Voice Activity Detector Noise Filter Adjustment

Table continues on the next page...

Field	Description
VADNFILADJ	Selects the adjustment value of the noise filter. For more information, refer to Noise Filter .
7-4 —	Reserved
3-0 VADNGAIN	Voice Activity Detector Noise Gain Configures the gain value for the noise energy or envelope estimated. It is an unsigned value which is multiplied with the estimated noise data. **NOTE:** VADNGAIN=0 gain corresponds to 1.

14.3.6.16 Voice Activity Detector 0 Noise Data (VAD0_NDATA)

14.3.6.16.1 Offset

Register	Offset
VAD0_NDATA	A4h

14.3.6.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VADNDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.6.16.3 Fields

Field	Description
31-16 —	Reserved
15-0 VADNDATA	Voice Activity Detector Noise Data

Field	Description
	Stores the noise energy or noise envelope calculated by the HWVAD. It can be used by software for a further voice activity detection. **NOTE:** This field value should be ignored when VADFOUTDIS=1 or VADST10=1.

14.3.6.17 Voice Activity Detector 0 Zero-Crossing Detector (VAD0_ZCD)

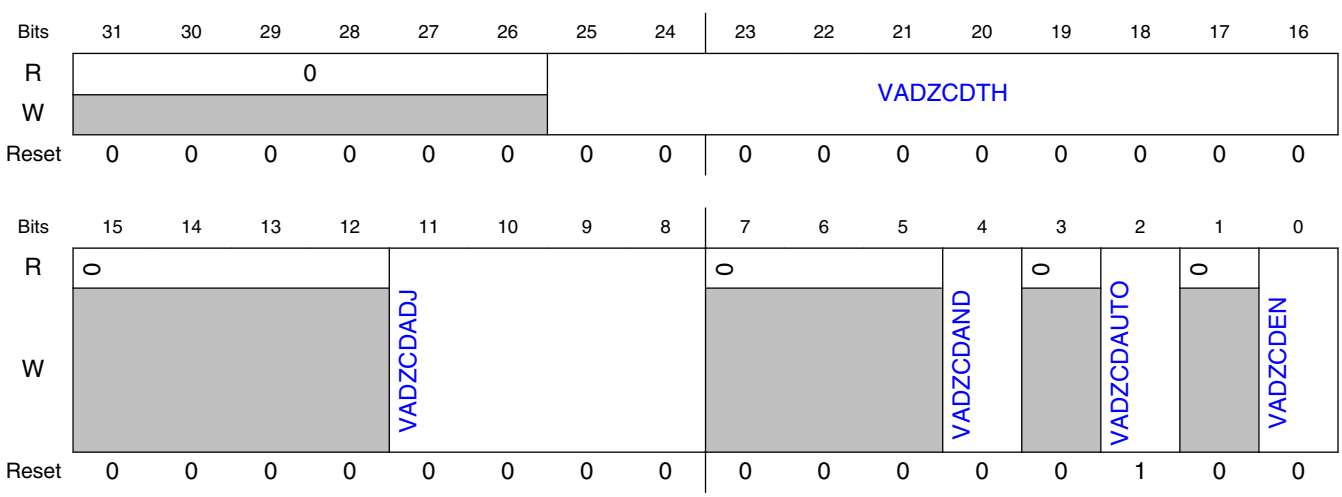
NOTE

These bit-fields must be configured when the MICFIL is not running.

14.3.6.17.1 Offset

Register	Offset
VAD0_ZCD	A8h

14.3.6.17.2 Diagram



14.3.6.17.3 Fields

Field	Description
31-26	Reserved

Table continues on the next page...

Field	Description
—	
25-16 VADZCDTH	Zero-Crossing Detector Threshold Configures the initial value taken to estimate the ZCD threshold when it is estimated automatically (VASDZCDAUTO = 1), otherwise (VADZCDAUTO = 0) this bit-field is the static value of the ZCD threshold.
15-12 —	Reserved
11-8 VADZCDADJ	Zero-Crossing Detector Adjustment Selects the adjustment value of the threshold estimator. For more information, refer to Zero-Crossing Detector .
7-5 —	Reserved
4 VADZCDAND	Zero-Crossing Detector AND Behavior Selects whether the Zero-Crossing detector results will be AND'ed or OR'ed with the energy-based detection to form the voice activity detection event. 0 - The ZCD result is OR'ed with the energy-based detection. 1 - The ZCD result is AND'ed with the energy-based detection.
3 —	Reserved
2 VADZCDAUTO	Zero-Crossing Detector Automatic Threshold Enables the automatic estimation of the ZCD threshold. 0 - The ZCD threshold is not estimated automatically 1 - The ZCD threshold is estimated automatically
1 —	Reserved
0 VADZCDEN	Zero-Crossing Detector Enable Enables the Zero-Crossing Detector (ZCD) in the Hardware Voice Activity Detector (HWVAD). 0 - The ZCD is disabled 1 - The ZCD is enabled

14.4 Synchronous Audio Interface (SAI)

14.4.1 Overview

The Synchronous Audio Interface (SAI) provides an interface that supports full-duplex serial interfaces with frame synchronization formats such as I^2S, AC97, TDM, and codec/DSP interfaces.

14.4.1.1 Block diagram

The following block diagram also shows the module clocks.

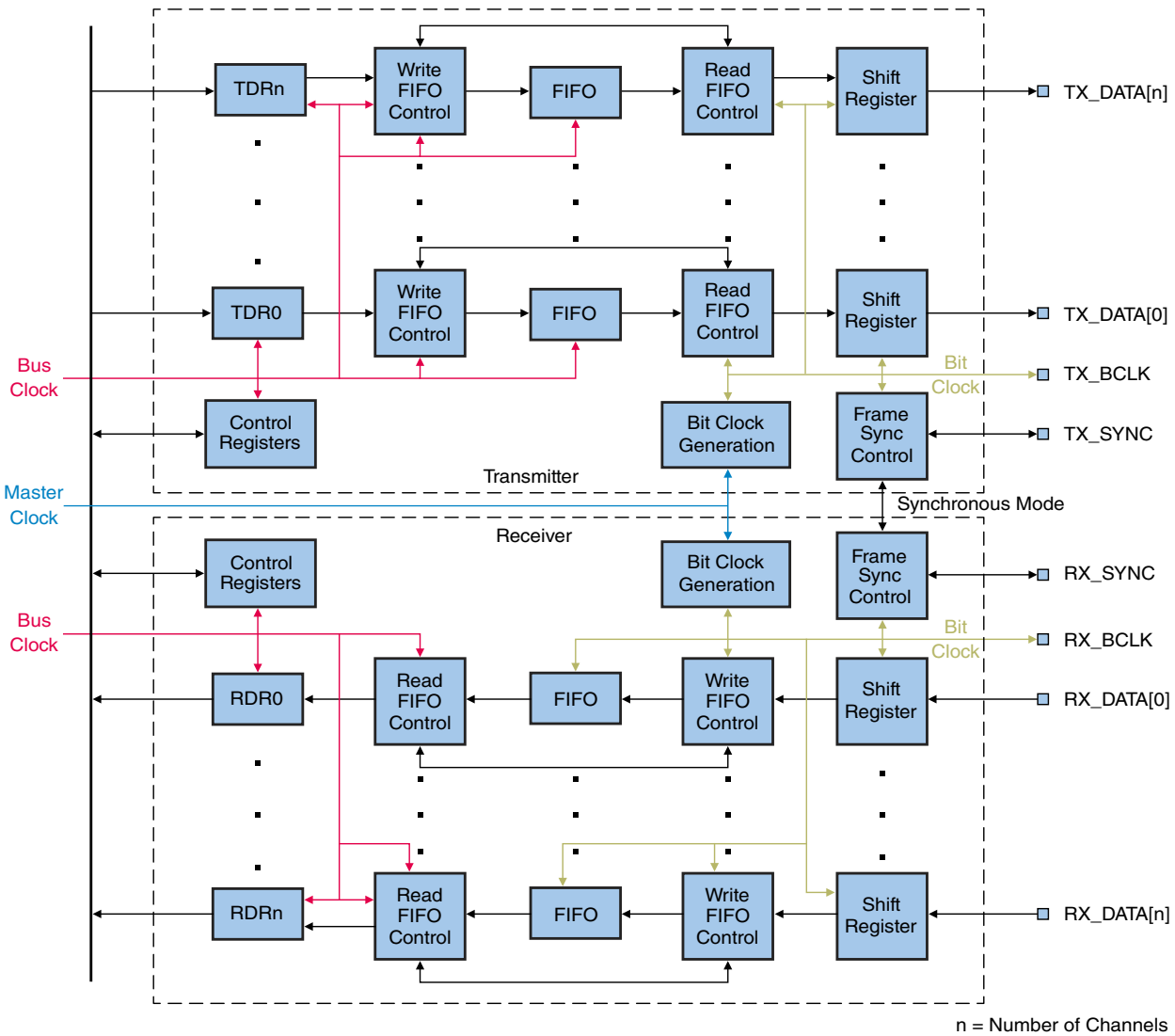


Figure 14-13. SAI block diagram

14.4.1.2 Features

Some of the features are not supported across all SAI instances; see the chip-specific SAI information in the first section of this chapter.

- Transmitter with independent bit clock and frame sync supporting 8 data lines
- Receiver with independent bit clock and frame sync supporting 8 data lines

- Each data line can support a maximum frame size of 32 words
- 8- to 32-bit word size
- Word size configured separately for the first word and remaining words in a frame
- Asynchronous 128 x 32-bit FIFO for each transmit and receive data line supporting:
  - Graceful restart after FIFO error
  - Automatic restart after FIFO error without software intervention
  - 8-bit and 16-bit data packing into each 32-bit FIFO word
  - Multiple data line FIFOs combining into single data line FIFO
- Independent 32-bit timestamp counters and bit counters for monitoring transmit and receive progress

14.4.2 Functional description

This section provides a complete functional description of the block.

14.4.2.1 Modes of operation

Module power modes include Run mode.

14.4.2.1.1 Run mode

In Run mode, the SAI transmitter and receiver operate normally.

14.4.2.2 Synchronous modes

The SAI transmitter and receiver can operate synchronously to each other or synchronously to other SAI peripheral instances.

14.4.2.2.1 Synchronous mode

The SAI transmitter and receiver can be configured to operate with synchronous bit clock and frame sync.

If both the transmitter and receiver use the transmitter bit clock and frame sync:

- Configure the transmitter for asynchronous operation and the receiver for synchronous operation.

- Enable the receiver in synchronous mode only after both the transmitter and receiver are enabled.
- Enable the transmitter last and disable the tranmitter first.

If both the transmitter and receiver use the receiver bit clock and frame sync:

- Configure the receiver for asynchronous operation and the transmitter for synchronous operation.
- Enable the transmitter in synchronous mode only after both the receiver and transmitter are enabled.
- Enable the receiver last and disable the receiver first.

When operating in synchronous mode, the transmitter and receiver share only the bit clock, frame sync, and transmitter/receiver enable. Otherwise, the transmitter and receiver operate independently, although the configuration registers must be configured consistently across both the transmitter and receiver.

14.4.2.2.2 Multiple SAI Synchronous mode

Synchronous operation between multiple SAI peripherals is not supported on all devices. This mode requires the source of the bit clock and frame sync to be configured for asynchronous operation and the remaining users of the bit clock and frame sync to be configured for synchronous operation.

Synchronous operation between multiple SAI transmitters or receivers also requires the source of the bit clock and frame sync to be enabled for any of the synchronous transmitters or receivers to also be enabled. It is recommended that the source of the bit clock and frame sync is the last enabled and the first disabled.

When operating in Synchronous mode, only the bit clock, frame sync, and transmitter/receiver enable are shared. The separate SAI peripherals otherwise operate independently, although configuration registers must be configured consistently across both the transmitter and receiver.

14.4.2.3 Frame sync configuration

When enabled, the SAI continuously transmits and/or receives frames of data. Each frame consists of a fixed number of words and each word consists of a fixed number of bits. Within each frame, any given word can be masked causing the receiver to ignore that word and the transmitter to tri-state for the duration of that word.

The frame sync signal indicates the start of each frame. A valid frame sync requires a rising edge (if active high) or falling edge (if active low) to be detected and the transmitter or receiver cannot be busy with a previous frame. A valid frame sync is also ignored (slave mode) or not generated (master mode) for the first four bit clock cycles after enabling the transmitter or receiver.

The transmitter and receiver frame sync can be configured independently with any of the following options:

- Generate externally or internally
- Require active high or active low
- Assert with the first bit in frame or assert one bit early
- Assert for a duration of between 1 bit clock and the first word length
- Set frame length from 1 to 32 words per frame
- Support word length of 8 to 32 bits per word
    - First word length and remaining word lengths can be configured separately
- Transmit/Receive words MSB first or LSB first

These configuration options cannot be changed after enabling the SAI transmitter or receiver.

14.4.2.4 Data FIFO

Each transmit and receive channel includes a 128 x 32-bit size FIFO. To access the FIFO data use the SAI Transmit/Receive Data Registers.

14.4.2.4.1 Data alignment

Data in the FIFO can be aligned anywhere within the 32-bit wide register through the use of the First Bit Shifted configuration field, which selects the bit index (between 31 and 0) of the first bit shifted.

Examples of supported data alignment and the required First Bit Shifted configuration are illustrated in Figure 14-14 for LSB First configurations and Figure 14-15 for MSB First configurations.

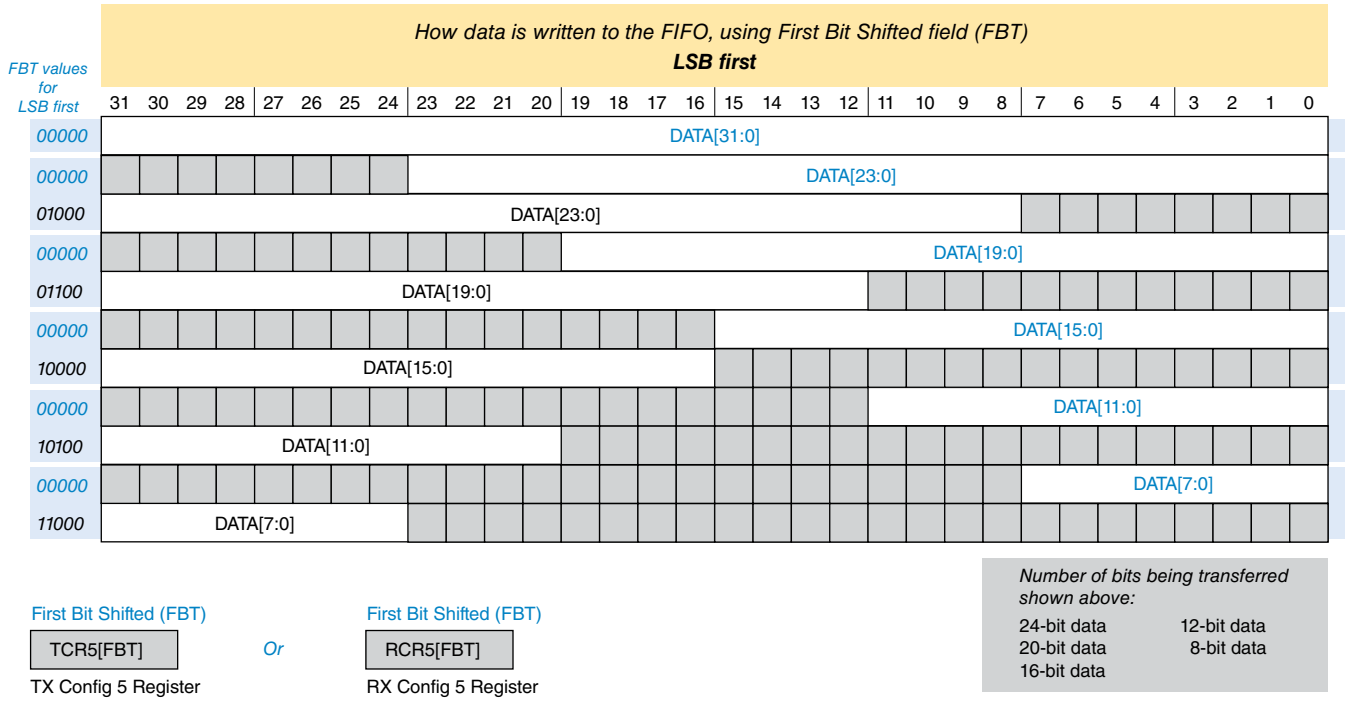


Figure 14-14. SAI first bit shifted, LSB first

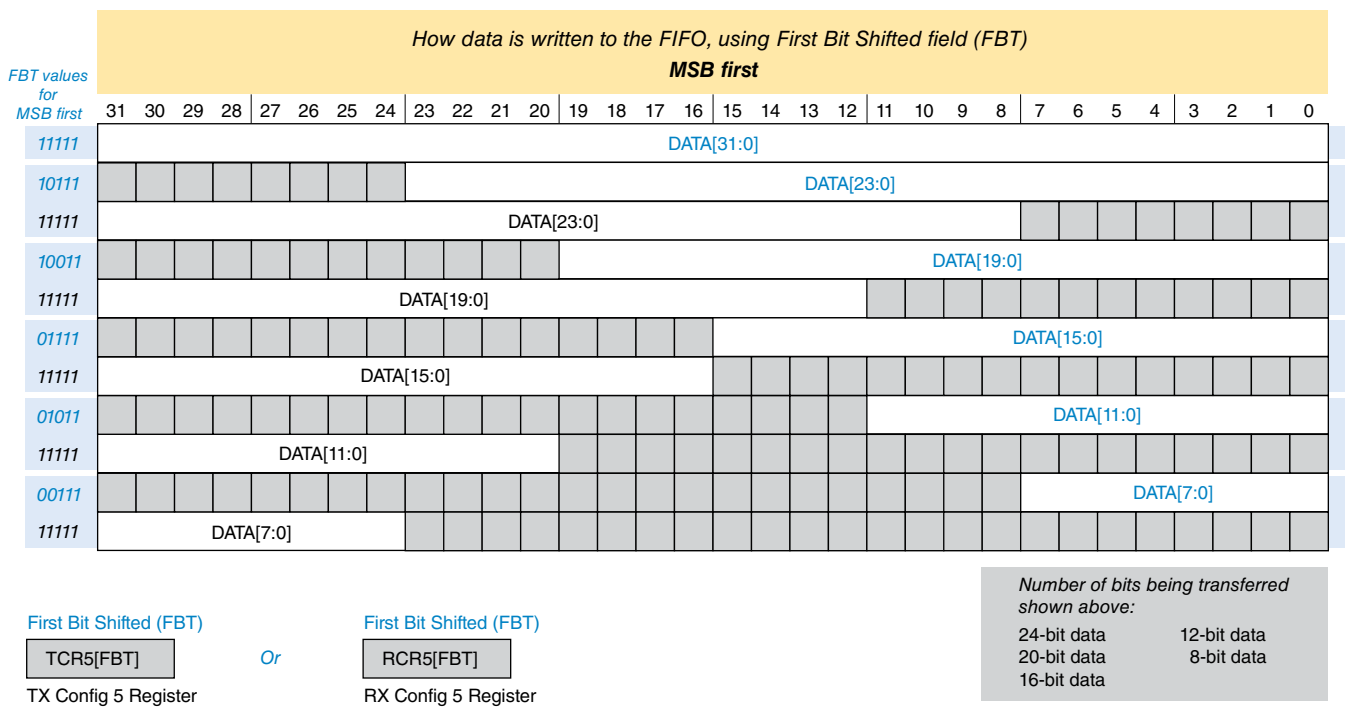


Figure 14-15. SAI first bit shifted, MSB first

14.4.2.4.2 FIFO pointers

When writing to a Transmit Data Register (TDR*n*), the Write FIFO Pointer (WFP) of the corresponding Transmit FIFO Register (TFR*n*) increments after each valid write. The SAI supports 8-bit, 16-bit, and 32-bit writes to the Transmit Data Register and the FIFO pointer increments after each individual write. Note that 8-bit writes should only be used when transmitting up to 8-bit data; 16-bit writes should only be used when transmitting up to 16-bit data.

- If the Transmit FIFO is full, writes to a Transmit Data Register are ignored.
- If the Transmit FIFO is empty, to avoid a FIFO underrun, write to the Transmit Data Register at least 3 bit clocks before the start of the next unmasked word. Before enabling the transmitter, initialize the Transmit FIFO with data (because after enabling the transmitter, the transmitter starts a new frame, and if no data is in the FIFO, then the transmitter immediately give an error).

When reading a Receive Data Register (RDR*n*), the Read FIFO Pointer (RFP) of the corresponding Receive FIFO Register (RFR*n*) increments after each valid read. The SAI supports 8-bit, 16-bit and 32-bit reads from the RDR and the FIFO pointer will increment after each individual read. Note that 8-bit reads should only be used when receiving up to 8-bit data; 16-bit reads should only be used when receiving up to 16-bit data.

- If the Receive FIFO is empty, reads from a Receive Data Register are ignored.
- If the Receive FIFO is full, to avoid a FIFO overrun, read the Receive Data Register at least 3 bit clocks before the end of an unmasked word.

14.4.2.4.3 FIFO packing

FIFO packing supports storing multiple 8-bit or 16-bit data words in one 32-bit FIFO word for the transmitter and/or receiver. While this can be emulated by adjusting the number of bits per word and number of words per frame (for example, one 32-bit word per frame versus two 16-bit words per frame), FIFO packing does not require even multiples of words per frame and fully supports word masking. When FIFO packing is enabled, the FIFO pointers only increment when the full 32-bit FIFO word has been written (transmit) or read (receive) by software, supporting scenarios where different words within each frame are loaded/stored in different areas of memory.

Using 16-bit FIFO packing for transmit, the transmit shift register loads at the start of each frame and after every second unmasked transmit word. The first word transmitted is taken from 16-bit word at byte offset 0x0 (the first bit is selected by TCR5[FBT] and must be configured within this 16-bit word) and the second word transmitted is taken from the 16-bit word at byte offset 0x2 (the first bit is selected by TCR5[FBT][3:0]). The transmitter transmits logic zero until the start of the next word once the 16-bit word has been transmitted.

Using 16-bit FIFO packing for receive, the receive shift register is stored after every second unmasked received word, and at the end of each frame if there is an odd number of unmasked received words in each frame. The first word received is stored in the 16-bit word at byte offset 0x0 (the first bit is selected by RCR5[FBT] and must be configured within this 16-bit word) and the second word received is stored in the 16-bit word at byte offset 0x2 (the first bit is selected by RCR5[FBT][3:0]). The receiver ignores received data until the start of the next word once the 16-bit word has been received.

The 8-bit FIFO packing is similar to 16-bit packing except four words are loaded or stored into each 32-bit FIFO word. The first word is loaded/stored in byte offset 0x0, second word in byte offset 0x1, third word in byte offset 0x2 and fourth word in byte offset 0x3. The TCR5[FBT] and/or RCR5[FBT] must be configured within byte offset 0x0.

14.4.2.4.4 FIFO Combine

FIFO Combining mode allows the separate FIFOs for multiple data channels to be used as a single FIFO for either software accesses or a single data channel or both. Note that the enabled data channels must be contiguous and data channel 0 must be enabled when using FIFO Combine mode.

Combining FIFOs for software access (writing transmit FIFO registers, reading receive FIFO registers) allows a DMA controller or software to read or write multiple FIFOs without incrementing the address that is accessed. Once enabled, the first software access to a FIFO register accesses the first enabled channel FIFO, while the second access to a FIFO register accesses the second enabled channel FIFO. This continues until software accesses the last enabled channel FIFO and the pointer resets back to the first enabled channel FIFO. To reset the pointer manually, software can reset the FIFOs or disable FIFO combining on software accesses.

Combining FIFOs for transmit data channels allows one data channel to use the FIFOs of all enabled channel FIFOs, with identical data output on each enabled data channel. The transmit shift registers for all enabled data channels load at the start of each frame and every N unmasked words (where N is the number of enabled data channels). The first word transmitted loads from the first enabled channel FIFO, while the second word transmitted loads from the second enabled channel FIFO, and so on until the end of the frame. Because the first word in each frame always loads from the first enabled data channel, it is recommended that the number of unmasked words in each frame is evenly divisible by the number of enabled data channels.

Combining FIFOs for receive data channels allows one data channel to use the FIFOs of all enabled channel FIFOs, with received data from channel 0 stored into each enabled data channel. The receive shift register for all enabled data channels are stored after every

N unmasked words (where N is the number of enabled data channels). The first word received stores to the first enabled channel FIFO, while the second word received stores to the second enabled channel FIFO, and so on until the end of the frame. Because the first word in each frame always stores to the first enabled data channel, it is recommended that the number of unmasked words in each frame is evenly divisible by the number of enabled data channels.

Note that combining FIFOs for data channels loads or stores each channel FIFO at the same time. This means that FIFO error conditions are only checked every N words (where N is the number of enabled data channels) and that the FIFO warning and request flags assert if any of the enabled data channel meets the warning flag or request flag conditions.

14.4.2.5 Word mask register

The SAI transmitter and receiver each contain a word mask register (TMR and RMR) that can be used to mask any word in the frame. Because the word mask register is double buffered, software can update it before the end of each frame to mask a particular word in the next frame.

The TMR causes the Transmit Data pin to be tri-stated for the length of each selected word and the transmit FIFO is not read for masked words.

The RMR causes the received data for each selected word to be discarded and not written to the receive FIFO.

14.4.2.6 Timestamp Counter

The transmitter and receiver implement separate timestamp counters and bit counters. The bit counter increments at the end of each bit in a frame whenever the transmitter or receiver is enabled. The bit counter continues to increment if a word is masked or a FIFO underrun or overrun condition occurs, but does not increment if an external frame sync is configured and no frame sync is detected. The bit counter can be reset by software, due to synchronization delays it can take up to 3 bit clock cycles for the bit clock counter to continue incrementing after a bit counter reset.

The timestamp counter increments on the bus interface clock whenever it is enabled, but can be configured to synchronize to the bit counter by waiting for the bit counter to increment first (if configured, this synchronization is performed whenever the time counter is reset). The current value of the timestamp counter is latched whenever the bit counter increments. Reading the bit counter register will cause the latched timestamp

value to be saved in the bit counter timestamp register. The timestamp counter can be reset by software, this also resets the latched timestamp value and the bit counter timestamp register.

The timestamp counter and bit counter can be used by software to track the progress of the transmitter and receiver. It can also be used to calculate the relative frequency of the bit clock against the bus interface clock.

14.4.2.7 Interrupts and DMA requests

The SAI transmitter and receiver generate separate interrupts and separate DMA requests, but support the same status flags. Asynchronous interrupts are only generated when the system clock is gated provided the corresponding BCLK continues.

14.4.2.7.1 FIFO request flag

The FIFO request flag sets based on the number of entries in the FIFO and the FIFO watermark configuration.

The transmit FIFO request flag sets when the number of entries in any of the enabled transmit FIFOs is less than or equal to the transmit FIFO watermark configuration and clears when the number of entries in each enabled transmit FIFO is greater than the transmit FIFO watermark configuration.

The receive FIFO request flag sets when the number of entries in any of the enabled receive FIFOs is greater than the receive FIFO watermark configuration and clears when the number of entries in each enabled receive FIFO is less than or equal to the receive FIFO watermark configuration.

The FIFO request flag can generate an interrupt or a DMA request.

14.4.2.7.2 FIFO warning flag

The FIFO warning flag sets based on the number of entries in the FIFO.

The transmit warning flag sets when the number of entries in any of the enabled transmit FIFOs is empty and clears when the number of entries in each enabled transmit FIFO is not empty.

The receive warning flag sets when the number of entries in any of the enabled receive FIFOs is full and clears when the number of entries in each enabled receive FIFO is not full.

The FIFO warning flag can generate an Interrupt or a DMA request.

14.4.2.7.3 FIFO error flag

The Transmit FIFO Error Flag (TCSR[FEF] = 1]) sets when any of the enabled transmit FIFOs underrun. After the error flag sets, all enabled transmit channels transmit zero data before TCSR[FEF] clears.

When FIFO Continue on Error is enabled (TCR4[FCONT] = 1), the FIFO continues transmitting data following an underrun without software intervention. To ensure that data transmits in the correct order, the transmitter continues from the same word number in the frame that caused the FIFO to underrun, but only after new data writes to the transmit FIFO. Software should still clear the TCSR[FEF] flag, but without reinitializing the transmit FIFOs.

The Receive FIFO Error Flag sets (RCSR[FEF] = 1) when any of the enabled receive FIFOs overflow. After the flag is set, all enabled receive channels discard received data until RCSR[FEF] clears and the next receive frame starts. Empty all enabled receive FIFOs before RCSR[FEF] clears.

When Receive FIFO Continue on Error is enabled (RCR4[FCONT] = 1) the FIFO continues receiving data following an overflow without software intervention. To ensure that data is received in the correct order, the receiver continues from the same word number in the frame that caused the FIFO to overflow, but only after data has been read from the receive FIFO. Software should still clear the RCSR[FEF] flag, but without emptying the receive FIFOs.

The FIFO Error Flag can generate only an interrupt.

14.4.2.7.4 Sync error flag

The Sync Error Flag, TCSR[SEF] or RCSR[SEF], sets when configured for an externally generated frame sync and the external frame sync asserts when the transmitter or receiver is busy with the previous frame. The external frame sync assertion is ignored and the sync error flag sets. When the Sync Error Flag sets, the transmitter or receiver continues checking for frame sync assertion when idle or at the end of each frame.

The Sync Error Flag can only generate an interrupt.

14.4.2.7.5 Word start flag

The Word Start Flag sets (TCSR[WSF] = 1) at the start of the second bit clock for the selected word, as configured by the Word Flag Configuration field (TCR3[WDFL]).

The Word Start Flag can generate an interrupt only.

14.4.2.8 SAI clocking

The SAI clocks include:

- Audio master clock
- Bit clock
- Bus clock

14.4.2.8.1 Audio master clock

The audio master clock generates the bit clock when the receiver or transmitter is configured for an internally generated bit clock. The transmitter and receiver can independently select between the bus clock and up to three audio master clocks to generate the bit clock.

The SAI peripheral controls the pin direction and post-divider of an audio master clock. The post-divide can be used to divide down the clock output on the MCLK signal pin, without affecting the audio master clock used by the transmitter and receiver.

The audio master clock generation and selection is chip-specific. Refer to chip-specific clocking information about how the audio master clocks are generated.

14.4.2.8.2 Bit clock

The SAI transmitter and receiver support asynchronous free-running bit clocks that can be generated internally from an audio master clock or supplied externally. There is also the option for synchronous bit clock and frame sync operation between the receiver and transmitter or between multiple SAI peripherals.

- If both transmitter and receiver are configured for asynchronous operation, then the transmitter and receiver will each use *their own* bit clock and frame sync.
- If the *transmitter* is configured for asynchronous mode and the receiver is configured for synchronous mode, then both transmitter and receiver will use the *transmitter* bit clock and frame sync.
- If the *receiver* is configured for asynchronous mode and the transmitter is configured for synchronous mode, then both transmitter and receiver will use the *receiver* bit clock and frame sync.

Note that the software configures synchronous or asynchronous mode, and that choice selects the bit clock/frame sync used.

Externally generated bit clocks must be:

- Enabled before the SAI transmitter or receiver is enabled
- Disabled after the SAI transmitter or receiver is disabled and completes its current frames

If the SAI transmitter or receiver uses an externally generated bit clock in asynchronous mode and that bit clock is generated by an SAI that is disabled in stop mode, then the transmitter or receiver should be disabled by software before entering stop mode. This issue does not apply when the transmitter or receiver is in a synchronous mode because all synchronous SAIs are enabled and disabled simultaneously.

14.4.2.8.3 Bus clock

The bus clock is used by the control and configuration registers and to generate synchronous interrupts and DMA requests.

NOTE

Although there is no minimum bus clock frequency specified, the bus clock frequency must be fast enough (relative to the bit clock frequency) to ensure that the FIFOs can be serviced, without generating either a transmitter FIFO underrun or receiver FIFO overflow condition.

14.4.2.9 SAI resets

The SAI is asynchronously reset on system reset. The SAI has a software reset and a FIFO reset.

14.4.2.9.1 Software reset

The SAI transmitter includes a software reset that resets all transmitter internal logic, including the bit clock generation, status flags, and FIFO pointers. This software reset does not reset the configuration registers. The software reset remains asserted until cleared by software.

The SAI receiver includes a software reset that resets all receiver internal logic, including the bit clock generation, status flags and FIFO pointers. This software reset does not reset the configuration registers. The software reset remains asserted until cleared by software.

14.4.2.9.2 FIFO reset

The SAI transmitter includes a FIFO reset that synchronizes the FIFO write pointer to the same value as the FIFO read pointer. This FIFO reset empties the FIFO contents and is used after TCSR[FEF] is set, and before the FIFO is re-initialized and TCSR[FEF] is cleared. The FIFO reset is asserted for one cycle only.

The SAI transmitter can also reset the FIFO of individual data channels by setting the appropriate TCR3[CFR] bit. This should only be done when the corresponding TCR3[TCE] bit is clear.

The SAI receiver includes a FIFO reset that synchronizes the FIFO read pointer to the same value as the FIFO write pointer. This empties the FIFO contents and is to be used after the RCSR[FEF] is set and any remaining data has been read from the FIFO, and before the RCSR[FEF] is cleared. The FIFO reset is asserted for one cycle only.

The SAI receiver can also reset the FIFO of individual data channels by setting the appropriate RCR3[CFR] bit. This should only be done when the corresponding RCR3[RCE] bit is clear.

14.4.3 External signals

Name	Function	I/O
TX_BCLK	Transmit Bit Clock. The bit clock is an input when externally generated and an output when internally generated.	I/O
TX_SYNC	Transmit Frame Sync. The frame sync is an input sampled synchronously by the bit clock when externally generated and an output generated synchronously by the bit clock when internally generated.	I/O
TX_DATA[7:0]	Transmit Data. The transmit data is generated synchronously by the bit clock and is tristated whenever not transmitting a word.	O
RX_BCLK	Receive Bit Clock. The bit clock is an input when externally generated and an output when internally generated.	I/O
RX_SYNC	Receive Frame Sync. The frame sync is an input sampled synchronously by the bit clock when externally generated and an output generated synchronously by the bit clock when internally generated.	I/O
RX_DATA[7:0]	Receive Data. The receive data is sampled synchronously by the bit clock.	I
MCLK	Audio Master Clock. The audio master clock is an input when externally generated and an output when internally generated.	I/O

14.4.4 Memory map and register definition

A read or write access to an address from offset 0x108 and above will result in a bus error.

14.4.4.1 SAI register descriptions

14.4.4.1.1 SAI memory map

SAI1 base address: 30C1_0000h

SAI2 base address: 30C2_0000h

SAI3 base address: 30C3_0000h

SAI5 base address: 30C5_0000h

SAI6 base address: 30C6_0000h

SAI7 base address: 30C8_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID (VERID)	32	RO	0301_0002
4	Parameter (PARAM)	32	RO	0005_0708
8	Transmit Control (TCSR)	32	RW	0000_0000
C	Transmit Configuration 1 (TCR1)	32	RW	0000_0000
10	Transmit Configuration 2 (TCR2)	32	RW	0000_0000
14	Transmit Configuration 3 (TCR3)	32	RW	0000_0000
18	Transmit Configuration 4 (TCR4)	32	RW	0000_0000
1C	Transmit Configuration 5 (TCR5)	32	RW	0000_0000
20 - 3C	Transmit Data (TDR0 - TDR7)	32	WORZ	0000_0000
40 - 5C	Transmit FIFO (TFR0 - TFR7)	32	RO	0000_0000
60	Transmit Mask (TMR)	32	RW	0000_0000
70	Transmit Timestamp Control (TTCR)	32	RW	0000_0000
74	Transmit Timestamp (TTSR)	32	RO	0000_0000
78	Transmit Bit Count (TBCR)	32	RO	0000_0000
7C	Transmit Bit Count Timestamp (TBCTR)	32	RO	0000_0000
88	Receive Control (RCSR)	32	RW	0000_0000
8C	Receive Configuration 1 (RCR1)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
90	Receive Configuration 2 (RCR2)	32	RW	0000_0000
94	Receive Configuration 3 (RCR3)	32	RW	0000_0000
98	Receive Configuration 4 (RCR4)	32	RW	0000_0000
9C	Receive Configuration 5 (RCR5)	32	RW	0000_0000
A0 - BC	Receive Data (RDR0 - RDR7)	32	RO	0000_0000
C0 - DC	Receive FIFO (RFR0 - RFR7)	32	RO	0000_0000
E0	Receive Mask (RMR)	32	RW	0000_0000
F0	Receive Timestamp Control (RTCR)	32	RW	0000_0000
F4	Receive Timestamp (RTSR)	32	RO	0000_0000
F8	Receive Bit Count (RBCR)	32	RO	0000_0000
FC	Receive Bit Count Timestamp (RBCTR)	32	RO	0000_0000
100	MCLK Control (MCR)	32	RW	0000_0000

14.4.4.1.2 Version ID (VERID)

Contains version numbers for the module design and feature set.

14.4.4.1.2.1 Offset

Register	Offset
VERID	0h

14.4.4.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

14.4.4.1.2.3 Fields

Field	Description
31-24 MAJOR	Major Version Number This read only field returns the major version number for the specification.
23-16 MINOR	Minor Version Number This read only field returns the minor version number for the specification.
15-0 FEATURE	Feature Specification Number This read only field returns the feature set number. 0000000000000000 - Standard feature set. 0000000000000010 - Standard feature set with Timestamp Registers.

14.4.4.1.3 Parameter (PARAM)

Contains parameter values that were implemented in the module.

14.4.4.1.3.1 Offset

Register	Offset
PARAM	4h

14.4.4.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												FRAME			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				FIFO				0				DATALINE			
W																
Reset	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0

14.4.4.1.3.3 Fields

Field	Description
31-20 —	Reserved

Table continues on the next page...

Field	Description
19-16 FRAME	Frame Size The maximum number of slots per frame is 2^FRAME.
15-12 —	Reserved
11-8 FIFO	FIFO Size The number of words in each FIFO is 2^FIFO.
7-4 —	Reserved
3-0 DATALINE	Number of Datalines The number of datalines implemented.

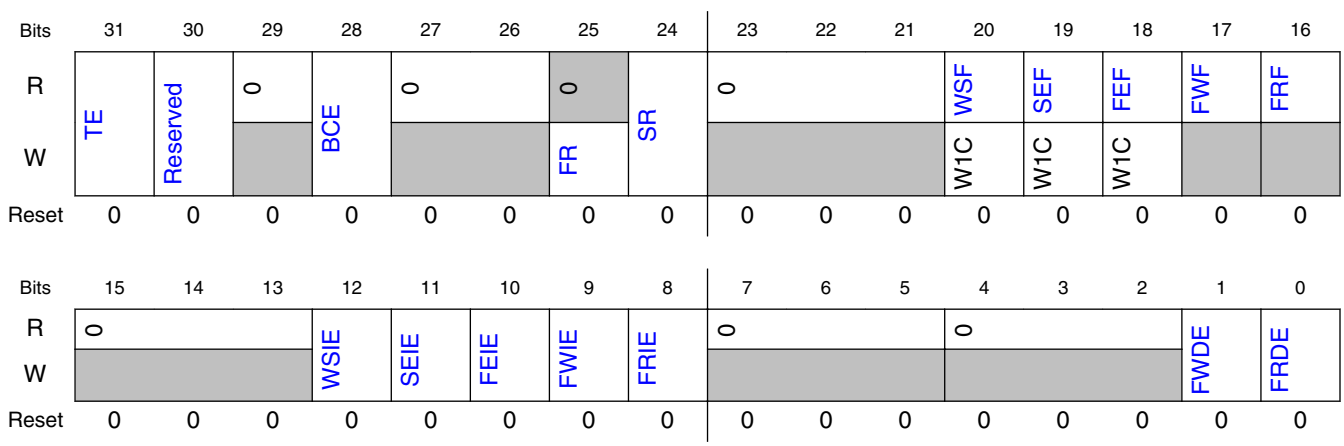
14.4.4.1.4 Transmit Control (TCSR)

Contains transmitter enable fields including resets, error and interrupt enable fields, and error flag fields.

14.4.4.1.4.1 Offset

Register	Offset
TCSR	8h

14.4.4.1.4.2 Diagram



14.4.4.1.4.3 Fields

Field	Description
31 TE	Transmitter Enable Enables/disables the transmitter. When software clears this field, the transmitter remains enabled, and this bit remains set, until the end of the current frame. 0 - Transmitter is disabled. 1 - Transmitter is enabled, or transmitter has been disabled and has not yet reached end of frame.
30 —	Reserved Software should only write zero to this reserved bit.
29 —	Reserved
28 BCE	Bit Clock Enable Enables the transmit bit clock, separately from the TE. This field is automatically set whenever TE is set. When software clears this field, the transmit bit clock remains enabled, and this bit remains set, until the end of the current frame. 0 - Transmit bit clock is disabled. 1 - Transmit bit clock is enabled.
27-26 —	Reserved
25 FR	FIFO Reset Empties the FIFO, and sets the FIFO read and write pointers to the same value, which may or may not be zero. Reading this field will always return zero. FIFO pointers should only be reset when the transmitter is disabled or the FIFO error flag is set. 0 - No effect. 1 - FIFO reset.
24 SR	Software Reset When set, resets the internal transmitter logic including the FIFO read and write pointers. Software-visible registers are not affected, except for the status registers. 0 - No effect. 1 - Software reset.
23-21 —	Reserved
20 WSF	Word Start Flag Indicates that the start of the configured word has been detected. Write a logic 1 to this field to clear this flag. 0 - Start of word not detected. 1 - Start of word detected.
19 SEF	Sync Error Flag Indicates that an error in the externally-generated frame sync has been detected. Write a logic 1 to this field to clear this flag. 0 - Sync error not detected. 1 - Frame sync error detected.

Table continues on the next page...

Field	Description
18 FEF	FIFO Error Flag Indicates that an enabled transmit FIFO has underrun. Write a logic 1 to this field to clear this flag. 0 - Transmit underrun not detected. 1 - Transmit underrun detected.
17 FWF	FIFO Warning Flag Indicates that an enabled transmit FIFO is empty. 0 - No enabled transmit FIFO is empty. 1 - Enabled transmit FIFO is empty.
16 FRF	FIFO Request Flag Indicates that the number of words in an enabled transmit channel FIFO is less than or equal to the transmit FIFO watermark. 0 - Transmit FIFO watermark has not been reached. 1 - Transmit FIFO watermark has been reached.
15-13 —	Reserved
12 WSIE	Word Start Interrupt Enable Enables/disables word start interrupts. 0 - Disables interrupt. 1 - Enables interrupt.
11 SEIE	Sync Error Interrupt Enable Enables/disables sync error interrupts. 0 - Disables interrupt. 1 - Enables interrupt.
10 FEIE	FIFO Error Interrupt Enable Enables/disables FIFO error interrupts. 0 - Disables the interrupt. 1 - Enables the interrupt.
9 FWIE	FIFO Warning Interrupt Enable Enables/disables FIFO warning interrupts. 0 - Disables the interrupt. 1 - Enables the interrupt.
8 FRIE	FIFO Request Interrupt Enable Enables/disables FIFO request interrupts. 0 - Disables the interrupt. 1 - Enables the interrupt.
7-5 —	Reserved
4-2 —	Reserved

Table continues on the next page...

Field	Description
1 FWDE	FIFO Warning DMA Enable Enables/disables DMA requests. 0 - Disables the DMA request. 1 - Enables the DMA request.
0 FRDE	FIFO Request DMA Enable Enables/disables DMA requests. 0 - Disables the DMA request. 1 - Enables the DMA request.

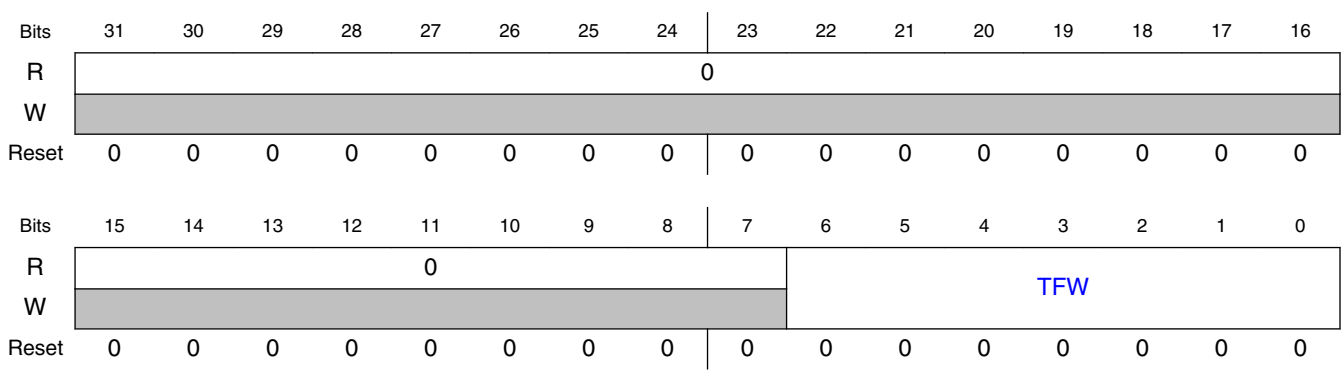
14.4.4.1.5 Transmit Configuration 1 (TCR1)

Configures the watermark level for all enabled transmit channels.

14.4.4.1.5.1 Offset

Register	Offset
TCR1	Ch

14.4.4.1.5.2 Diagram



14.4.4.1.5.3 Fields

Field	Description
31-7 —	Reserved

Table continues on the next page...

Field	Description
6-0 TFW	Transmit FIFO Watermark

14.4.4.1.6 Transmit Configuration 2 (TCR2)

Contains the SYNC mode and clock setting fields.

NOTE

This register must not be altered when TCSR[TE] is set.

14.4.4.1.6.1 Offset

Register	Offset
TCR2	10h

14.4.4.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SYNC		BCS	BCI	MSEL		BCP	BCD	BYP	0						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.6.3 Fields

Field	Description
31-30 SYNC	Synchronous Mode Configures between asynchronous and synchronous modes of operation. When configured for a synchronous mode of operation, the receiver or other SAI peripheral must be configured for asynchronous operation. 00 - Asynchronous mode. 01 - Synchronous with receiver. 10 - Synchronous with another SAI transmitter

Table continues on the next page...

Field	Description
	11 - Synchronous with another SAI receiver
29 BCS	Bit Clock Swap This field swaps the bit clock used by the transmitter. When the transmitter is configured in asynchronous mode and this bit is set, the transmitter is clocked by the receiver bit clock (RX_BCLK). This allows the transmitter and receiver to share the same bit clock, but the transmitter continues to use the transmit frame sync (TX_SYNC). When the transmitter is configured in synchronous mode, the transmitter BCS field and receiver BCS field must be set to the same value. When both are set, the transmitter and receiver are both clocked by the transmitter bit clock (TX_BCLK) but use the receiver frame sync (RX_SYNC). This field has no effect when synchronous to another SAI peripheral. 0 - Use the normal bit clock source. 1 - Swap the bit clock source.
28 BCI	Bit Clock Input When this field is set and using an internally generated bit clock in either synchronous or asynchronous mode, the bit clock actually used by the transmitter is delayed by the pad output delay (the transmitter is clocked by the pad input as if the clock was externally generated). This has the effect of decreasing the data input setup time, but increasing the data output valid time. The slave mode timing from the datasheet should be used for the transmitter when this bit is set. In synchronous mode, this bit allows the transmitter to use the slave mode timing from the datasheet, while the receiver uses the master mode timing. This field has no effect when configured for an externally generated bit clock or when synchronous to another SAI peripheral . 0 - No effect. 1 - Internal logic is clocked as if bit clock was externally generated.
27-26 MSEL	MCLK Select Selects the audio Master Clock option used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock. **NOTE:** Depending on the device, some Master Clock options might not be available. See the chip-specific information for the meaning of each option. 00 - Bus Clock selected. 01 - Master Clock (MCLK) 1 option selected. 10 - Master Clock (MCLK) 2 option selected. 11 - Master Clock (MCLK) 3 option selected.
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. 0 - Bit clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1 - Bit clock is active low with drive outputs on falling edge and sample inputs on rising edge.
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0 - Bit clock is generated externally in Slave mode. 1 - Bit clock is generated internally in Master mode.
23 BYP	Bit Clock Bypass Bypasses the bit clock divider, internal bit clock is divide by 1 of the audio master clock. 0 - Internal bit clock is generated from bit clock divider.

Table continues on the next page...

Field	Description
	1 - Internal bit clock is divide by one of the audio master clock.
22-8 —	Reserved
7-0 DIV	Bit Clock Divide Divides down the audio master clock to generate the bit clock when configured for an internal bit clock. The division value is (DIV + 1) * 2.

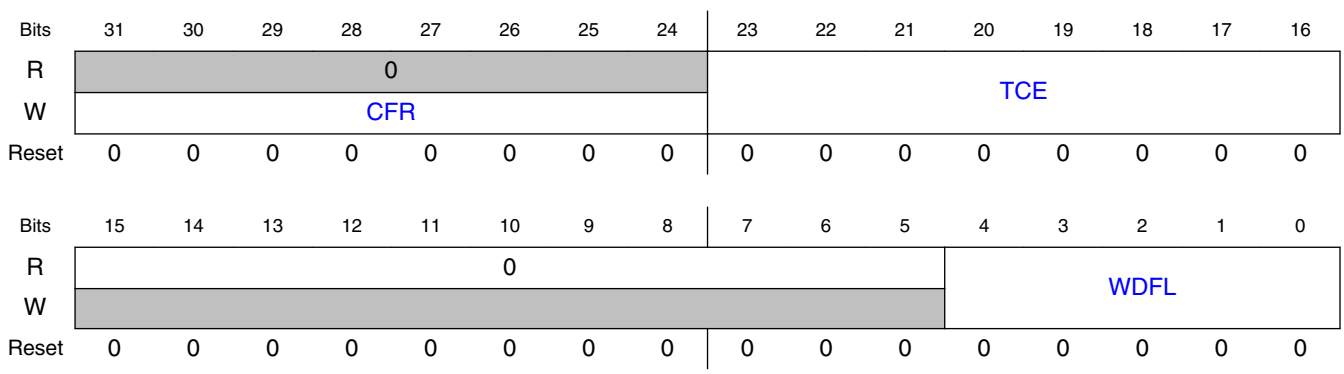
14.4.4.1.7 Transmit Configuration 3 (TCR3)

Contains the transmit channel settings.

14.4.4.1.7.1 Offset

Register	Offset
TCR3	14h

14.4.4.1.7.2 Diagram



14.4.4.1.7.3 Fields

Field	Description
31-24 CFR	Channel FIFO Reset Resets the FIFO pointers for a specific channel. Reading this field always returns zero. FIFO pointers should only be reset when a channel is disabled or the FIFO error flag is set.

Table continues on the next page...

Field	Description
	The width of CFR field = the number of transmit channels (call it N). For example, if CFR is 2 bits wide, then bit position 24 refers to transmit channel 1 FIFO pointer and bit position 25 refers to transmit channel 2 FIFO pointer. Setting bit 24 resets transmit channel 1 FIFO pointer, and setting bit 25 enables transmit channel 2 FIFO pointer. Setting bit N will reset transmit channel N FIFO pointer. **NOTE:** When there is only a single channel, there is no need for individual channel FIFO reset (TCR3[CFR]). In the case of a single channel, use the global FIFO reset (TCSR[FR]). 0b - No effect. 1b - Transmit data channel N FIFO is reset.
23-16 TCE	Transmit Channel Enable Enables the corresponding data channel for transmit operation. Changing TCE field will take effect immediately for generating the FIFO request and warning flags, but at the end of each frame for transmit operation. The width of TCE field = the number of transmit channels (call it N). For example, if TCE field is 2 bits wide, then bit position 16 refers to transmit channel 1 and bit position 17 refers to transmit channel 2. Setting bit 16 enables transmit channel 1, and setting bit 17 enables transmit channel 2. Setting bit N will enable transmit channel N. 0b - Transmit data channel N is disabled. 1b - Transmit data channel N is enabled.
15-5 —	Reserved
4-0 WDFL	Word Flag Configuration Configures which word sets the start of word flag. The value written must be one less than the word number. For example, writing 0 configures the first word in the frame. When configured to a value greater than TCR4[FRSZ], then the start of word flag is never set.

14.4.4.1.8 Transmit Configuration 4 (TCR4)

Contains the transmit fields for FIFO Combine Mode, FIFO Packing Mode, and frame sync settings.

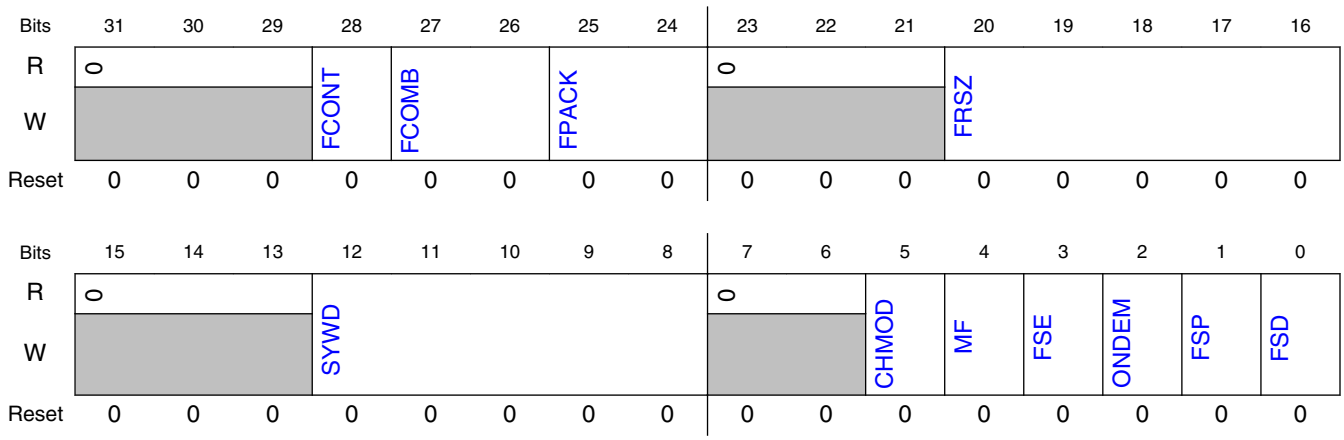
NOTE

This register must not be altered when TCSR[TE] is set.

14.4.4.1.8.1 Offset

Register	Offset
TCR4	18h

14.4.4.1.8.2 Diagram



14.4.4.1.8.3 Fields

Field	Description
31-29 —	Reserved
28 FCONT	FIFO Continue on Error Configures when the SAI will continue transmitting after a FIFO error has been detected. 0 - On FIFO error, the SAI will continue from the start of the next frame after the FIFO error flag has been cleared. 1 - On FIFO error, the SAI will continue from the same word that caused the FIFO error to set after the FIFO warning flag has been cleared.
27-26 FCOMB	FIFO Combine Mode When FIFO combine mode is enabled for FIFO writes, software writing to any FIFO data register will alternate the write among the enabled data channel FIFOs. For example, if two data channels are enabled then the first write will be performed to the first enabled data channel FIFO and the second write will be performed to the second enabled data channel FIFO. Reseting the FIFO or disabling FIFO combine mode for FIFO writes will reset the pointer back to the first enabled data channel. When FIFO combine mode is enabled for FIFO reads from the transmit shift registers, the transmit data channel output will alternate between the enabled data channel FIFOs. For example, if two data channels are enabled then the first unmasked word will be transmitted from the first enabled data channel FIFO and the second unmasked word will be transmitted from the second enabled data channel FIFO. Since the first word of the frame is always transmitted from the first enabled data channel FIFO, it is recommended that the number of unmasked words per frame is evenly divisible by the number of enabled data channels. 00 - FIFO combine mode disabled. 01 - FIFO combine mode enabled on FIFO reads (from transmit shift registers). 10 - FIFO combine mode enabled on FIFO writes (by software). 11 - FIFO combine mode enabled on FIFO reads (from transmit shift registers) and writes (by software).
25-24 FPACK	FIFO Packing Mode

Table continues on the next page...

Field	Description
	Enables packing of 8-bit data or 16-bit data into each 32-bit FIFO word. If the word size is greater than 8-bit or 16-bit then only the first 8-bit or 16-bits are loaded from the FIFO. The first word in each frame always starts with a new 32-bit FIFO word and the first bit shifted must be configured within the first packed word. When FIFO packing is enabled, the FIFO write pointer will only increment when the full 32-bit FIFO word has been written by software. 00 - FIFO packing is disabled. 01 - Reserved 10 - 8-bit FIFO packing is enabled. 11 - 16-bit FIFO packing is enabled.
23-21 —	Reserved
20-16 FRSZ	Frame size Configures the number of words in each frame. The value written must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 32 words.
15-13 —	Reserved
12-8 SYWD	Sync Width Configures the length of the frame sync in number of bit clocks. The value written must be one less than the number of bit clocks. For example, write 0 for the frame sync to assert for one bit clock only. The sync width cannot be configured longer than the first word of the frame.
7-6 —	Reserved
5 CHMOD	Channel Mode Configures if transmit data pins are configured for TDM mode or Output mode. 0 - TDM mode, transmit data pins are tri-stated when slots are masked or channels are disabled. 1 - Output mode, transmit data pins are never tri-stated and will output zero when slots are masked or channels are disabled.
4 MF	MSB First Configures whether the LSB or the MSB is transmitted first. 0 - LSB is transmitted first. 1 - MSB is transmitted first.
3 FSE	Frame Sync Early 0 - Frame sync asserts with the first bit of the frame. 1 - Frame sync asserts one bit before the first bit of the frame.
2 ONDEM	On Demand Mode When set, and the frame sync is generated internally, a frame sync is only generated when the FIFO warning flag is clear. 0 - Internal frame sync is generated continuously. 1 - Internal frame sync is generated when the FIFO warning flag is clear.
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync.

Table continues on the next page...

Field	Description
	0 - Frame sync is active high. 1 - Frame sync is active low.
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0 - Frame sync is generated externally in Slave mode. 1 - Frame sync is generated internally in Master mode.

14.4.4.1.9 Transmit Configuration 5 (TCR5)

Contains transmit word width and bit index settings.

NOTE

This register must not be altered when TCSR[TE] is set.

14.4.4.1.9.1 Offset

Register	Offset
TCR5	1Ch

14.4.4.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			WNW					0			W0W				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			FBT					0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.9.3 Fields

Field	Description
31-29	Reserved

Table continues on the next page...

Field	Description
—	
28-24 WNW	Word N Width Configures the number of bits in each word, for each word except the first in the frame. The value written must be one less than the number of bits per word. Word width of less than 8 bits is not supported.
23-21 —	Reserved
20-16 W0W	Word 0 Width Configures the number of bits in the first word in each frame. The value written must be one less than the number of bits in the first word. Word width of less than 8 bits is not supported if there is only one word per frame.
15-13 —	Reserved
12-8 FBT	First Bit Shifted Configures the bit index for the first bit transmitted for each word in the frame. If configured for MSB First, the index of the next bit transmitted is one less than the current bit transmitted. If configured for LSB First, the index of the next bit transmitted is one more than the current bit transmitted. The value written must be greater than or equal to the word width when configured for MSB First. The value written must be less than or equal to 31-word width when configured for LSB First.
7-0 —	Reserved

14.4.4.1.10 Transmit Data (TDR0 - TDR7)

Writes to this register when the transmit FIFO is not full will push the data written into the transmit data FIFO. Writes to this register when the transmit FIFO is full are ignored.

14.4.4.1.10.1 Offset

For a = 0 to 7:

Register	Offset
TDRa	20h + (a × 4h)

14.4.4.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	TDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	TDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.10.3 Fields

Field	Description
31-0 TDR	Transmit Data Register

14.4.4.1.11 Transmit FIFO (TFR0 - TFR7)

The MSB of the read and write pointers is used to distinguish between FIFO full and empty conditions. If the read and write pointers are identical, then the FIFO is empty. If the read and write pointers are identical except for the MSB, then the FIFO is full.

14.4.4.1.11.1 Offset

For a = 0 to 7:

Register	Offset
TFRa	40h + (a × 4h)

14.4.4.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WCP	0							WFP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RFP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.11.3 Fields

Field	Description
31 WCP	Write Channel Pointer When FIFO Combine mode is enabled for writes, indicates that this data channel is the next FIFO to be written. 0 - No effect. 1 - FIFO combine is enabled for FIFO writes and this FIFO will be written on the next FIFO write.
30-24 —	Reserved
23-16 WFP	Write FIFO Pointer FIFO write pointer for transmit data channel.
15-8 —	Reserved
7-0 RFP	Read FIFO Pointer FIFO read pointer for transmit data channel.

14.4.4.1.12 Transmit Mask (TMR)

This register is double-buffered and updates:

1. When TCSR[TE] is first set
2. At the end of each frame

This allows the masked words in each frame to change from frame to frame.

14.4.4.1.12.1 Offset

Register	Offset
TMR	60h

14.4.4.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TWM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TWM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.12.3 Fields

Field	Description
31-0 TWM	Transmit Word Mask Configures whether the transmit word is masked (transmit data pins are tri-stated or drive zero and transmit data not read from FIFO) for the corresponding word in the frame. 00000000000000000000000000000000 - Word N is enabled. 00000000000000000000000000000001 - Word N is masked. The transmit data pins are tri-stated or drive zero when masked.

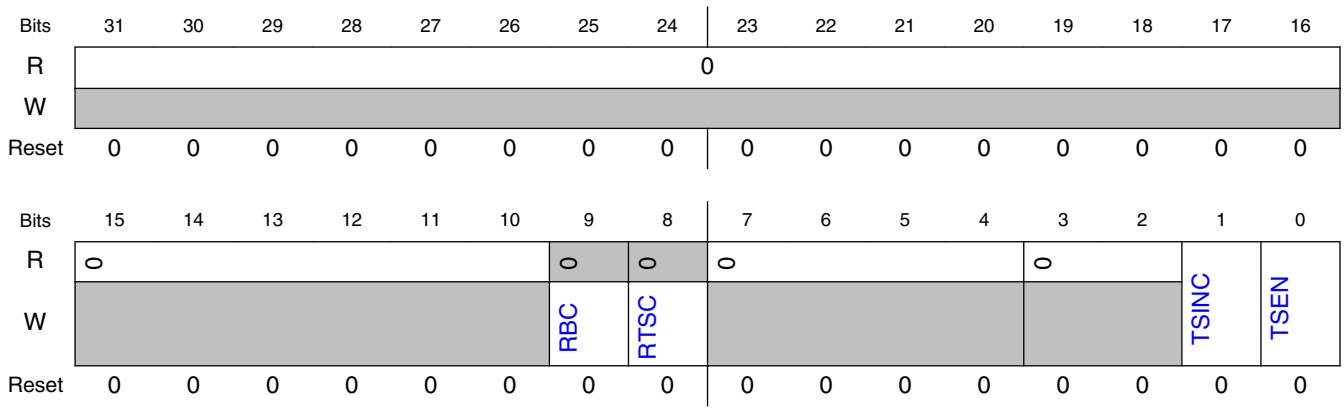
14.4.4.1.13 Transmit Timestamp Control (TTCR)

This register controls the transmit timestamp counter and bit clock counter.

14.4.4.1.13.1 Offset

Register	Offset
TTCR	70h

14.4.4.1.13.2 Diagram



14.4.4.1.13.3 Fields

Field	Description
31-10 —	Reserved
9 RBC	Reset Bit Counter Reset the bit counter. 0 - Bit counter is not reset. 1 - Bit counter is reset.
8 RTSC	Reset Timestamp Counter Reset the timestamp counter. 0 - Timestamp counter is not reset. 1 - Timestamp counter is reset.
7-4 —	Reserved
3-2 —	Reserved
1 TSINC	Timestamp Increment Configures when the timestamp counter starts to increment. 0 - Timestamp counter starts to increment when enabled and the bit counter has incremented. 1 - Timestamp counter starts to increment when enabled.
0 TSEN	Timestamp Enable Enables the timestamp counter to increment on the bus interface clock. 0 - Timestamp counter is disabled. 1 - Timestamp counter is enabled.

14.4.4.1.14 Transmit Timestamp (TTSR)

This register returns the current value of the transmit timestamp counter.

14.4.4.1.14.1 Offset

Register	Offset
TTSR	74h

14.4.4.1.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.14.3 Fields

Field	Description
31-0 TSC	Timestamp Counter Current value of the timestamp counter.

14.4.4.1.15 Transmit Bit Count (TBCR)

This register returns the current value of the transmit bit counter. The bit counter increments for each bit in a frame when the SAI transmitter is enabled. Reading this register causes the Bit Count Timestamp Regsiter to update with the value of the timestamp counter that correponds to the bit count register value.

14.4.4.1.15.1 Offset

Register	Offset
TBCR	78h

14.4.4.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.15.3 Fields

Field	Description
31-0 BCNT	Bit Counter Returns the current value of the bit counter.

14.4.4.1.16 Transmit Bit Count Timestamp (TBCTR)

Returns the value of the timestamp register that corresponds to the value last read from the bit count register.

14.4.4.1.16.1 Offset

Register	Offset
TBCTR	7Ch

14.4.4.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.16.3 Fields

Field	Description
31-0 BCTS	Bit Timestamp Returns the timestamp counter value that matches the last read of the bit count register.

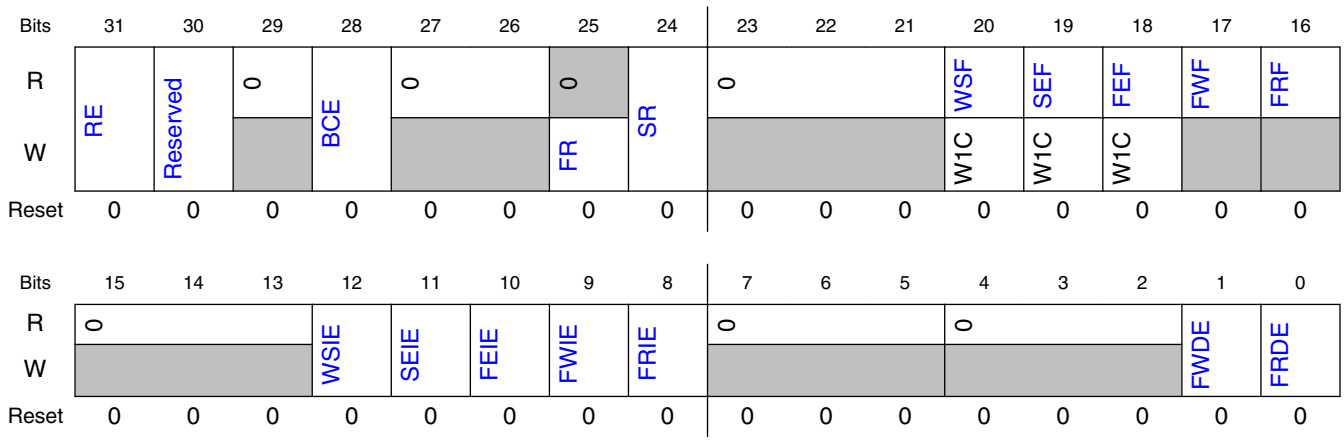
14.4.4.1.17 Receive Control (RCSR)

Contains receiver enable fields including resets, error and interrupt enable fields, and error flag fields.

14.4.4.1.17.1 Offset

Register	Offset
RCSR	88h

14.4.4.1.17.2 Diagram



14.4.4.1.17.3 Fields

Field	Description
31 RE	Receiver Enable Enables/disables the receiver. When software clears this field, the receiver remains enabled, and this bit remains set, until the end of the current frame. 0 - Receiver is disabled. 1 - Receiver is enabled, or receiver has been disabled and has not yet reached end of frame.
30 —	Reserved Software should only write zero to this reserved bit.
29 —	Reserved
28 BCE	Bit Clock Enable Enables the receive bit clock, separately from RE. This field is automatically set whenever RE is set. When software clears this field, the receive bit clock remains enabled, and this field remains set, until the end of the current frame. 0 - Receive bit clock is disabled. 1 - Receive bit clock is enabled.
27-26 —	Reserved
25 FR	FIFO Reset Empties the FIFO, and sets the FIFO read and write pointers to the same value, which may or may not be zero. Reading this field will always return zero. FIFO pointers should only be reset when the receiver is disabled or the FIFO error flag is set. 0 - No effect. 1 - FIFO reset.
24 SR	Software Reset

Table continues on the next page...

Field	Description
	Resets the internal receiver logic including the FIFO pointers. Software-visible registers are not affected, except for the status registers. 0 - No effect. 1 - Software reset.
23-21 —	Reserved
20 WSF	Word Start Flag Indicates that the start of the configured word has been detected. Write a logic 1 to this field to clear this flag. 0 - Start of word not detected. 1 - Start of word detected.
19 SEF	Sync Error Flag Indicates that an error in the externally-generated frame sync has been detected. Write a logic 1 to this field to clear this flag. 0 - Sync error not detected. 1 - Frame sync error detected.
18 FEF	FIFO Error Flag Indicates that an enabled receive FIFO has overflowed. Write a logic 1 to this field to clear this flag. 0 - Receive overflow not detected. 1 - Receive overflow detected.
17 FWF	FIFO Warning Flag Indicates that an enabled receive FIFO is full. 0 - No enabled receive FIFO is full. 1 - Enabled receive FIFO is full.
16 FRF	FIFO Request Flag Indicates that the number of words in an enabled receive channel FIFO is greater than the receive FIFO watermark. 0 - Receive FIFO watermark not reached. 1 - Receive FIFO watermark has been reached.
15-13 —	Reserved
12 WSIE	Word Start Interrupt Enable Enables/disables word start interrupts. 0 - Disables interrupt. 1 - Enables interrupt.
11 SEIE	Sync Error Interrupt Enable Enables/disables sync error interrupts. 0 - Disables interrupt. 1 - Enables interrupt.
10	FIFO Error Interrupt Enable

Table continues on the next page...

Field	Description
FEIE	Enables/disables FIFO error interrupts. 0 - Disables the interrupt. 1 - Enables the interrupt.
9 FWIE	FIFO Warning Interrupt Enable Enables/disables FIFO warning interrupts. 0 - Disables the interrupt. 1 - Enables the interrupt.
8 FRIE	FIFO Request Interrupt Enable Enables/disables FIFO request interrupts. 0 - Disables the interrupt. 1 - Enables the interrupt.
7-5 —	Reserved
4-2 —	Reserved
1 FWDE	FIFO Warning DMA Enable Enables/disables DMA requests. 0 - Disables the DMA request. 1 - Enables the DMA request.
0 FRDE	FIFO Request DMA Enable Enables/disables DMA requests. 0 - Disables the DMA request. 1 - Enables the DMA request.

14.4.4.1.18 Receive Configuration 1 (RCR1)

Configures the watermark level for all enabled receiver channels.

14.4.4.1.18.1 Offset

Register	Offset
RCR1	8Ch

14.4.4.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RFW							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.18.3 Fields

Field	Description
31-7 —	Reserved
6-0 RFW	Receive FIFO Watermark

14.4.4.1.19 Receive Configuration 2 (RCR2)

Contains the SYNC mode and clock setting fields.

NOTE

This register must not be altered when RCSR[RE] is set.

14.4.4.1.19.1 Offset

Register	Offset
RCR2	90h

14.4.4.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SYNC		BCS	BCI	MSEL		BCP	BCD	BYP	0						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.19.3 Fields

Field	Description
31-30 SYNC	Synchronous Mode Configures between asynchronous and synchronous modes of operation. When configured for a synchronous mode of operation, the transmitter or other SAI peripheral must be configured for asynchronous operation. 00 - Asynchronous mode. 01 - Synchronous with transmitter. 10 - Synchronous with another SAI receiver. 11 - Synchronous with another SAI transmitter.
29 BCS	Bit Clock Swap This field swaps the bit clock used by the receiver. When the receiver is configured in asynchronous mode and this bit is set, the receiver is clocked by the transmitter bit clock (TX_BCLK). This allows the transmitter and receiver to share the same bit clock, but the receiver continues to use the receiver frame sync (RX_SYNC). When the receiver is configured in synchronous mode, the transmitter BCS field and receiver BCS field must be set to the same value. When both are set, the transmitter and receiver are both clocked by the receiver bit clock (RX_BCLK) but use the transmitter frame sync (TX_SYNC). This field has no effect when synchronous to another SAI peripheral. 0 - Use the normal bit clock source. 1 - Swap the bit clock source.
28 BCI	Bit Clock Input When this field is set and using an internally generated bit clock in either synchronous or asynchronous mode, the bit clock actually used by the receiver is delayed by the pad output delay (the receiver is clocked by the pad input as if the clock was externally generated). This has the effect of decreasing the data input setup time, but increasing the data output valid time. The slave mode timing from the datasheet should be used for the receiver when this bit is set. In synchronous mode, this bit allows the receiver to use the slave mode timing from the datasheet, while the transmitter uses the master mode timing. This field has no effect when configured for an externally generated bit clock or when synchronous to another SAI peripheral . 0 - No effect.

Table continues on the next page...

Field	Description
	1 - Internal logic is clocked as if bit clock was externally generated.
27-26 MSEL	MCLK Select Selects the audio Master Clock option used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock. **NOTE:** Depending on the device, some Master Clock options might not be available. See the chip-specific information for the availability and chip-specific meaning of each option. 00 - Bus Clock selected. 01 - Master Clock (MCLK) 1 option selected. 10 - Master Clock (MCLK) 2 option selected. 11 - Master Clock (MCLK) 3 option selected.
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. 0 - Bit Clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1 - Bit Clock is active low with drive outputs on falling edge and sample inputs on rising edge.
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0 - Bit clock is generated externally in Slave mode. 1 - Bit clock is generated internally in Master mode.
23 BYP	Bit Clock Bypass Bypasses the bit clock divider, internal bit clock is divide by 1 of the audio master clock. 0 - Internal bit clock is generated from bit clock divider. 1 - Internal bit clock is divide by one of the audio master clock.
22-8 —	Reserved
7-0 DIV	Bit Clock Divide Divides down the audio master clock to generate the bit clock when configured for an internal bit clock. The division value is (DIV + 1) * 2.

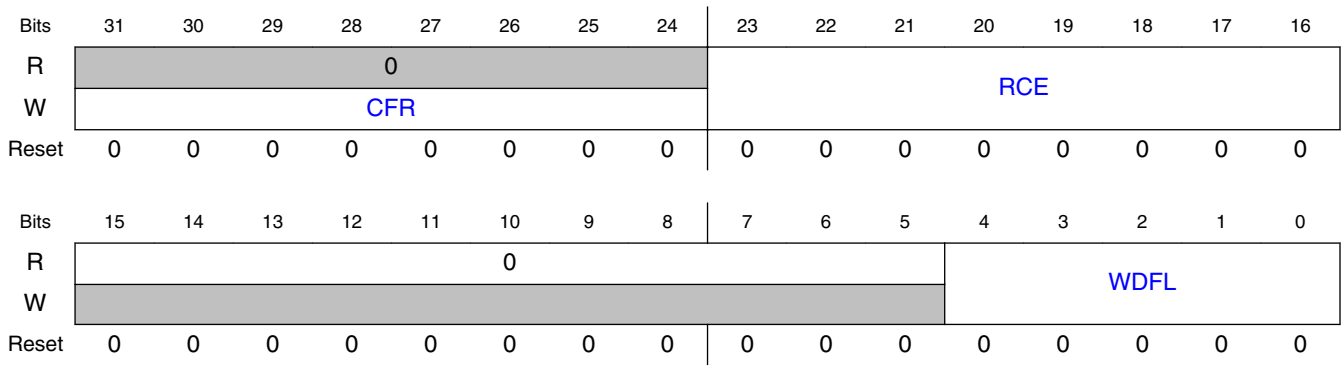
14.4.4.1.20 Receive Configuration 3 (RCR3)

Contains the receive channel settings.

14.4.4.1.20.1 Offset

Register	Offset
RCR3	94h

14.4.4.1.20.2 Diagram



14.4.4.1.20.3 Fields

Field	Description
31-24 CFR	Channel FIFO Reset Resets the FIFO pointers for a specific channel. Reading this field will always return zero. FIFO pointers should only be reset when a channel is disabled or the FIFO error flag is set. The width of CFR field = the number of receive channels (call it N). For example, if CFR is 2 bits wide, then bit position 24 refers to receive channel 1 FIFO pointer and bit position 25 refers to receive channel 2 FIFO pointer. Setting bit 24 resets receive channel 1 FIFO pointer, and setting bit 25 enables receive channel 2 FIFO pointer. Setting bit N will reset receive channel N FIFO pointer. 0b - No effect. 1b - Receive data channel N FIFO is reset.
23-16 RCE	Receive Channel Enable Enables the corresponding data channel for receive operation. Changing this field will take effect immediately for generating the FIFO request and warning flags, but at the end of each frame for receive operation. The width of RCE field = the number of receive channels (call it N). For example, if RCE field is 2 bits wide, then bit position 16 refers to receive channel 1 and bit position 17 refers to receive channel 2. Setting bit 16 enables receive channel 1, and setting bit 17 enables receive channel 2. Setting bit N will enable receive channel N. **NOTE:** When there is only a single channel, there is no need for individual channel FIFO reset (RCR3[CFR]). In the case of a single channel, use the global FIFO reset (RCSR[FR]). 0b - Receive data channel N is disabled. 1b - Receive data channel N is enabled.
15-5 —	Reserved
4-0 WDFL	Word Flag Configuration Configures which word the start of word flag is set. The value written should be one less than the word number (for example, write zero to configure for the first word in the frame). When configured to a value greater than the Frame Size field, then the start of word flag is never set.

14.4.4.1.21 Receive Configuration 4 (RCR4)

Contains the receive fields for FIFO Combine Mode, FIFO Packing Mode, and frame sync settings.

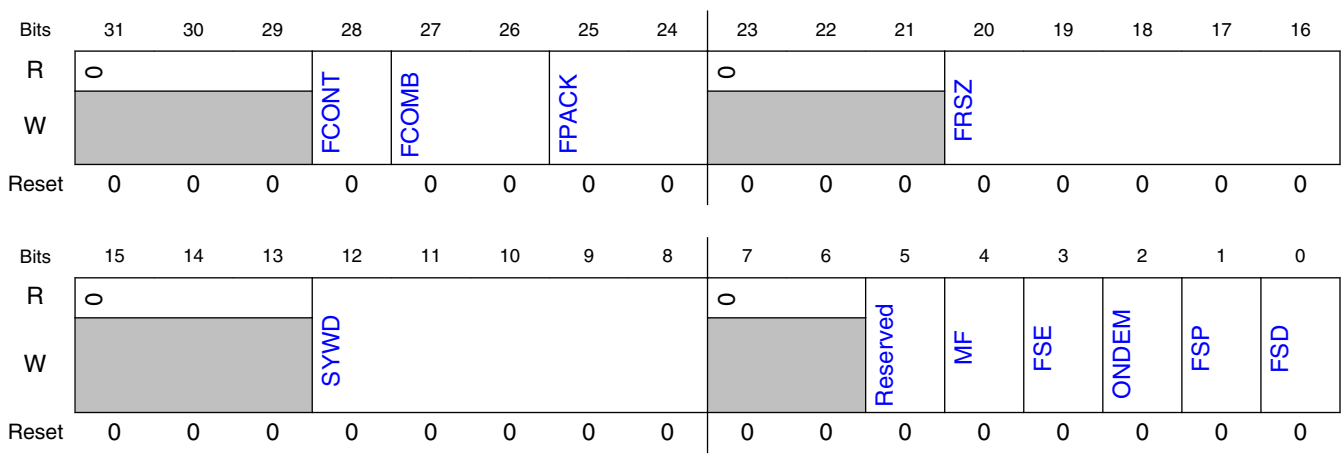
NOTE

This register must not be altered when RCSR[RE] is set.

14.4.4.1.21.1 Offset

Register	Offset
RCR4	98h

14.4.4.1.21.2 Diagram



14.4.4.1.21.3 Fields

Field	Description
31-29 —	Reserved
28 FCONT	FIFO Continue on Error Configures when the SAI will continue receiving after a FIFO error has been detected. 0 - On FIFO error, the SAI will continue from the start of the next frame after the FIFO error flag has been cleared. 1 - On FIFO error, the SAI will continue from the same word that caused the FIFO error to set after the FIFO warning flag has been cleared.
27-26	FIFO Combine Mode

Table continues on the next page...

Field	Description
FCOMB	When FIFO combine mode is enabled for FIFO reads, software reading any FIFO data register will alternate the read among the enabled data channel FIFOs. For example, if two data channels are enabled then the first read will be performed to the first enabled data channel FIFO and the second read will be performed to the second enabled data channel FIFO. Reseting the FIFO or disabling FIFO combine mode for FIFO reads will reset the pointer back to the first enabled data channel. When FIFO combine mode is enabled for FIFO writes from the receive shift registers, the first enabled data channel input will alternate between the enabled data channel FIFOs. For example, if two data channels are enabled then the first unmasked received word will be stored in the first enabled data channel FIFO and the second unmasked received word will be stored in the second enabled data channel FIFO. Since the first word of the frame is always stored in the first enabled data channel FIFO, it is recommended that the number of unmasked words per frame is evenly divisible by the number of enabled data channels. 00 - FIFO combine mode disabled. 01 - FIFO combine mode enabled on FIFO writes (from receive shift registers). 10 - FIFO combine mode enabled on FIFO reads (by software). 11 - FIFO combine mode enabled on FIFO writes (from receive shift registers) and reads (by software).
25-24 FPACK	FIFO Packing Mode Enables packing of 8-bit data or 16-bit data into each 32-bit FIFO word. If the word size is greater than 8-bit or 16-bit then only the first 8-bit or 16-bits are stored to the FIFO. The first word in each frame always starts with a new 32-bit FIFO word and the first bit shifted must be configured within the first packed word. When FIFO packing is enabled, the FIFO read pointer will only increment when the full 32-bit FIFO word has been read by software. 00 - FIFO packing is disabled 01 - Reserved. 10 - 8-bit FIFO packing is enabled 11 - 16-bit FIFO packing is enabled
23-21 —	Reserved
20-16 FRSZ	Frame Size Configures the number of words in each frame. The value written must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 32 words.
15-13 —	Reserved
12-8 SYWD	Sync Width Configures the length of the frame sync in number of bit clocks. The value written must be one less than the number of bit clocks. For example, write 0 for the frame sync to assert for one bit clock only. The sync width cannot be configured longer than the first word of the frame.
7-6 —	Reserved
5 —	Reserved Software should only write zero to this bit.
4 MF	MSB First Configures whether the LSB or the MSB is received first. 0 - LSB is received first.

Table continues on the next page...

Field	Description
	1 - MSB is received first.
3 FSE	Frame Sync Early 0 - Frame sync asserts with the first bit of the frame. 1 - Frame sync asserts one bit before the first bit of the frame.
2 ONDEM	On Demand Mode When set, and the frame sync is generated internally, a frame sync is only generated when the FIFO warning flag is clear. 0 - Internal frame sync is generated continuously. 1 - Internal frame sync is generated when the FIFO warning flag is clear.
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0 - Frame sync is active high. 1 - Frame sync is active low.
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0 - Frame Sync is generated externally in Slave mode. 1 - Frame Sync is generated internally in Master mode.

14.4.4.1.22 Receive Configuration 5 (RCR5)

This register must not be altered when RCSR[RE] is set.

14.4.4.1.22.1 Offset

Register	Offset
RCR5	9Ch

14.4.4.1.22.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			WNW					0			W0W				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			FBT					0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.22.3 Fields

Field	Description
31-29 —	Reserved
28-24 WNW	Word N Width Configures the number of bits in each word, for each word except the first in the frame. The value written must be one less than the number of bits per word. Word width of less than 8 bits is not supported.
23-21 —	Reserved
20-16 W0W	Word 0 Width Configures the number of bits in the first word in each frame. The value written must be one less than the number of bits in the first word. Word width of less than 8 bits is not supported if there is only one word per frame.
15-13 —	Reserved
12-8 FBT	First Bit Shifted Configures the bit index for the first bit received for each word in the frame. If configured for MSB First, the index of the next bit received is one less than the current bit received. If configured for LSB First, the index of the next bit received is one more than the current bit received. The value written must be greater than or equal to the word width when configured for MSB First. The value written must be less than or equal to 31-word width when configured for LSB First.
7-0 —	Reserved

14.4.4.1.23 Receive Data (RDR0 - RDR7)

Reads from this register when the receive FIFO is not empty return the data from the top of the receive FIFO. Reads from this register when the receive FIFO is empty are ignored.

Reading this register introduces one additional peripheral clock wait state on each read.

14.4.4.1.23.1 Offset

For a = 0 to 7:

Register	Offset
RDRa	A0h + (a × 4h)

14.4.4.1.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.23.3 Fields

Field	Description
31-0 RDR	Receive Data Register

14.4.4.1.24 Receive FIFO (RFR0 - RFR7)

The MSB of the read and write pointers is used to distinguish between FIFO full and empty conditions. If the read and write pointers are identical, then the FIFO is empty. If the read and write pointers are identical except for the MSB, then the FIFO is full.

14.4.4.1.24.1 Offset

For a = 0 to 7:

Register	Offset
RFRa	C0h + (a × 4h)

14.4.4.1.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								WFP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RCP	0							RFP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.24.3 Fields

Field	Description
31-24 —	Reserved
23-16 WFP	Write FIFO Pointer FIFO write pointer for receive data channel.
15 RCP	Receive Channel Pointer When FIFO Combine mode is enabled for reads, indicates that this data channel is the next FIFO to be read. 0 - No effect. 1 - FIFO combine is enabled for FIFO reads and this FIFO will be read on the next FIFO read.
14-8 —	Reserved
7-0 RFP	Read FIFO Pointer FIFO read pointer for receive data channel.

14.4.4.1.25 Receive Mask (RMR)

This register is double-buffered and updates:

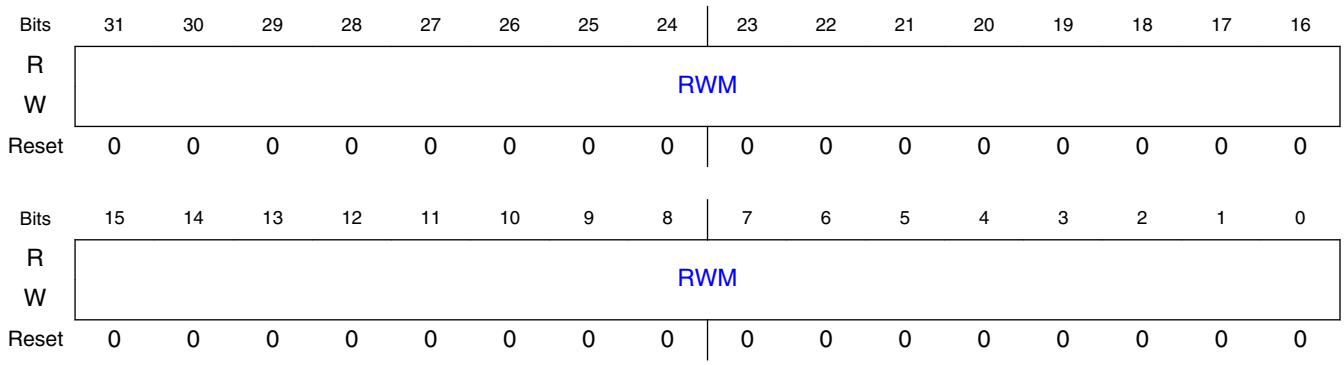
1. When RCSR[RE] is first set
2. At the end of each frame

This allows the masked words in each frame to change from frame to frame.

14.4.4.1.25.1 Offset

Register	Offset
RMR	E0h

14.4.4.1.25.2 Diagram



14.4.4.1.25.3 Fields

Field	Description
31-0 RWM	Receive Word Mask Configures whether the receive word is masked (received data ignored and not written to receive FIFO) for the corresponding word in the frame. 00000000000000000000000000000000 - Word N is enabled. 00000000000000000000000000000001 - Word N is masked.

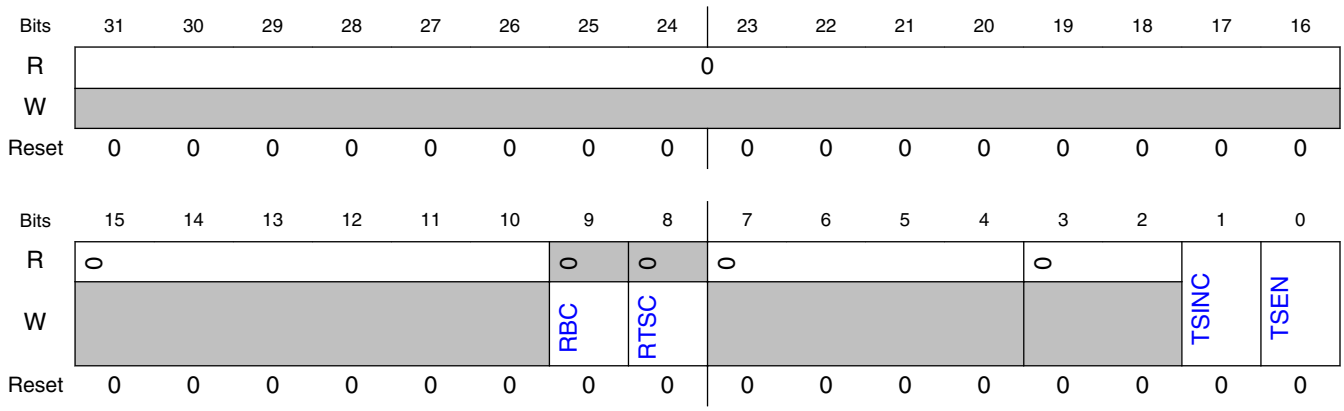
14.4.4.1.26 Receive Timestamp Control (RTCR)

Controls the receive timestamp counter and bit clock counter.

14.4.4.1.26.1 Offset

Register	Offset
RTCR	F0h

14.4.4.1.26.2 Diagram



14.4.4.1.26.3 Fields

Field	Description
31-10 —	Reserved
9 RBC	Reset Bit Counter Reset the receive bit counter. 0 - Bit counter is not reset. 1 - Bit counter is reset.
8 RTSC	Reset Timestamp Counter Reset the receive timestamp counter. 0 - Timestamp counter is not reset. 1 - Timestamp counter is reset.
7-4 —	Reserved
3-2 —	Reserved
1 TSINC	Timestamp Increment Configures when the timestamp counter starts to increment. 0 - Timestamp counter starts to increment when enabled and the bit counter has incremented. 1 - Timestamp counter starts to increment when enabled.
0 TSEN	Timestamp Enable Enables the timestamp counter to increment on the bus interface clock. 0 - Timestamp counter is disabled. 1 - Timestamp counter is enabled.

14.4.4.1.27 Receive Timestamp (RTSR)

Returns the current value of the receive timestamp counter.

14.4.4.1.27.1 Offset

Register	Offset
RTSR	F4h

14.4.4.1.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.27.3 Fields

Field	Description
31-0 TSC	Timestamp Counter Current value of the timestamp counter.

14.4.4.1.28 Receive Bit Count (RBCR)

Returns the current value of the receive bit counter. The bit counter increments for each bit in a frame when the SAI receiver is enabled. Reading this register causes the Bit Count Timestamp Regsiter to update with the value of the timestamp counter that correponds to the bit count register value.

14.4.4.1.28.1 Offset

Register	Offset
RBCR	F8h

14.4.4.1.28.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.28.3 Fields

Field	Description
31-0 BCNT	Bit Counter Returns the current value of the bit counter.

14.4.4.1.29 Receive Bit Count Timestamp (RBCTR)

Returns the value of the timestamp register that corresponds to the value last read from the bit count register.

14.4.4.1.29.1 Offset

Register	Offset
RBCTR	FCh

14.4.4.1.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.29.3 Fields

Field	Description
31-0 BCTS	Bit Timestamp Returns the timestamp counter value that matches the last read of the bit count register.

14.4.4.1.30 MCLK Control (MCR)

Controls the audio master clock.

14.4.4.1.30.1 Offset

Register	Offset
MCR	100h

14.4.4.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MOE	0				0		DIVEN	0						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.4.4.1.30.3 Fields

Field	Description
31 —	Reserved
30 MOE	MCLK Output Enable Configures MCLK signal pin as an input or an output. 0 - MCLK signal pin is an input. 1 - MCLK signal pin is an output.
29-26 —	Reserved
25-24 —	Reserved
23 DIVEN	MCLK Post Divide Enable Enables the post-divider that is output on the MCLK signal pin, the output divider does not impact the audio master clock used by the transmitter or receiver. 0 - Output on MCLK signal pin is the audio master clock. 1 - Output on MCLK signal pin is a post-divided version of audio master clock.
22-8 —	Reserved
7-0 DIV	MCLK Post Divide Post-divide for the audio master clock that only divides the output to the MCLK signal pin. The division value is (DIV + 1) * 2.

14.5 Asynchronous Sample Rate Converter (ASRC)

14.5.1 Overview

The Asynchronous Sample Rate Converter (ASRC) is a digital module that converts audio from a source sample rate to a destination sample rate.

14.5.1.1 Block Diagram

The following diagram gives a high-level overview of the ASRC.

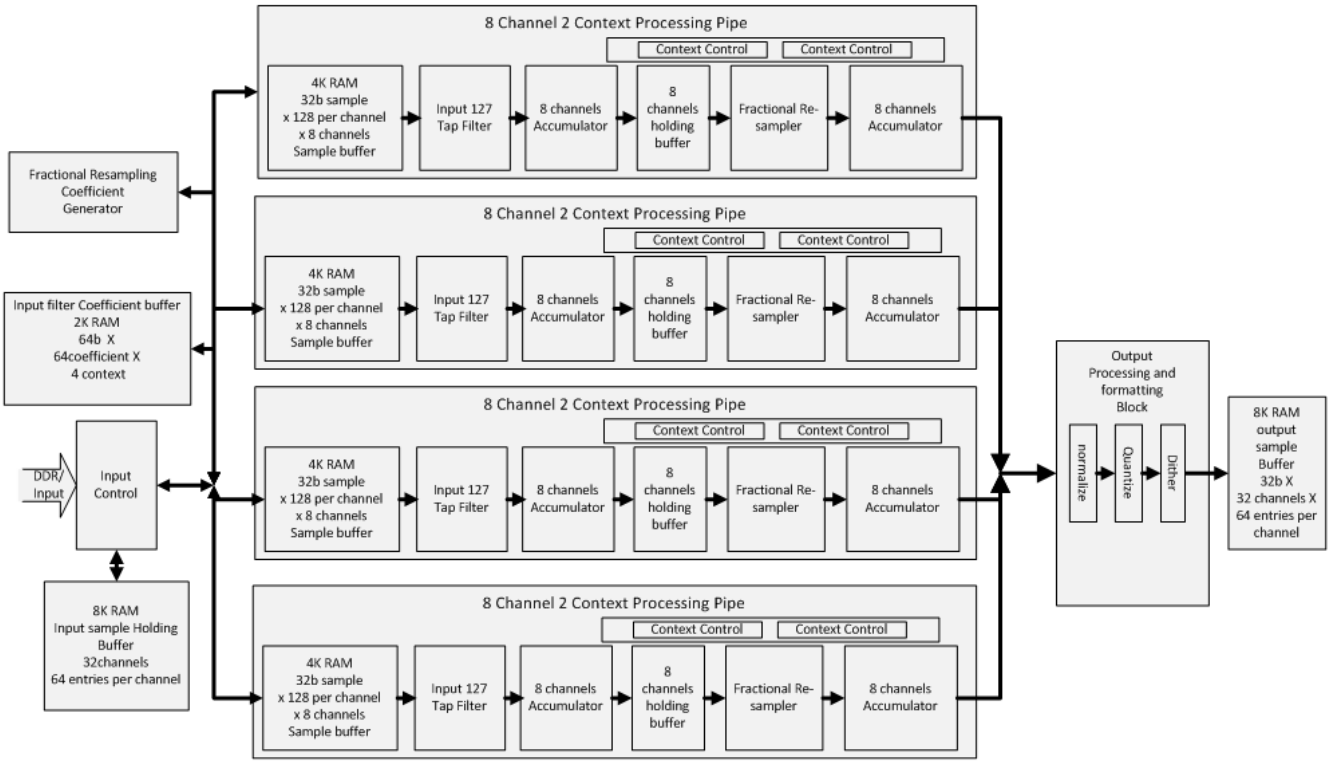


Figure 14-16. ASRC Block Diagram

14.5.1.2 Features

The primary features for the ASRC are as follows:

- 4 Contexts - groups of channels with an independent time base
- Fully independent and concurrent context control
- Simultaneous processing of up to 32 audio channels
- Programmable filter charachteristics for each context
- 32, 24, 20, and 16-bit fixed point audio sample support
- 32-bit floating point audio sample support
- 8kHz to 384 kHz sample rate
- 1/16 to 16x sample rate conversion ratio
- Software control of fine conversion ratio

14.5.2 Functional description

14.5.2.1 ASRC Datapath Overview

The ASRC can process 4 groups of audio channels with an independent time-base simultaneously. A group of channels with the same time-base (or resampling ratio) is referred to as a context. Each context has an independent input/output sample memory, resampling ratio, prefilter coefficient FIFO, number of channels, and assigned datapath resources. A context can be configured to start and stop at any time without affecting the processing of other contexts in flight. The maximum number of channels which can run on the ASRC is 32. These 32 channels can either be associated with a single context or spread across multiple contexts. The ASRC datapath is comprised of 4 independent context processing pipes capable of running the prefilter and resampling algorithms.

The following sections will describe the contents of each major component within the block diagram.

14.5.2.1.1 ASRC Input Control

The input control block acts as the interface between the ASRC and external sources of audio data. This block contains a circular queue to store sample data for each context before it is processed. In addition, each circular queue has interleaving hardware to support fetching data from a variety of sample sources. The circular queues are described in detail throughout section, ASRC FIFO Organization. The ASRC runs all internal calculations on a unified sample format. The input control block can convert a variety of input formats to the internal standard used by the ASRC by utilizing register fields within the ASRC_CTX_CTRL register. Note, all input samples must be written to the ASRC in a 32-bit packed format (1 sample contained within a 32-bit field). The input control block supports 16, 20, 24, and 32-bits per sample (ASRC_CTX_CTRL[BITS_PER_SAMPLE]), fixed point and floating-point data (ASRC_CTX_CTRL[FLOAT_FMT]), bit-reversal of sample data (ASRC_CTX_CTRL[BIT_REV]), and a variable sample position within each packed 32-bit write (ASRC_CTX_CTRL[SAMPLE_POSITION]). The main purpose of the input control block is to format samples and forward them to the context processing pipes when requested.

14.5.2.1.2 ASRC Context Processing Pipe

There are four 8-Channel 2-Context Processing Pipes in the ASRC design. Each processing pipe contains a prefilter and a fractional-resampler which can process a maximum of 8 audio channels at a time. The prefilter configuration is flexible and can be used to implement an FIR or IFIR filter architecture. The number of prefilter taps, stages, and IFIR expansion factor is programmable for each context. The resampler is configurable for 32, 64, or 128 taps. Note, the resampler configuration applies to every

context in the design. The 8 audio channels in each datapath processing pipe can be associated with up to 2 contexts (slots). A single context control within a datapath processing unit it is referred to as a slot. If a single context is configured to contain more than 8 channels then it must be distributed across multiple context processing pipe slots. The resources for a single context processing pipe can be configured through the ASRC_PROC_CTRL_SLOTx_Rxx registers. These registers contain the control fields for both slots in a processor SLOT0 and SLOT1. Each slot has control fields to select the context which will run (SLOTx_CTX_NUM), the number of channels associated with each slot (SLOTx_NUM_CH), which channels from the context will run in this slot (SLOTx_MIN_CH to SLOTx_MAX_CH), and an enable bit (SLOTx_EN). The context processing pipes act as the master control blocks throughout the datapath. They control fetching sample data, coefficients, performing computations, and managing available resources. When a resource is unavailable they will stall processing of samples and insert back pressure to the beginning of the pipeline.

14.5.2.1.3 ASRC Input Filter Coefficient Buffer

The input filter coefficient buffer manages addressing and fetching of the prefilter coefficients. Note, each context supports a programmable length prefilter which can be configured using the registers ASRC_CTX_CTRL_EXT1x[ST1_NUM_TAPS] and ASRC_CTX_CTRL_EXT2x[ST2_NUM_TAPS]. The prefilter can run in a single-stage (ST1 only) or two-stage (ST1 and ST2) mode by controlling the bit ASRC_CTX_CTRL_EXT1x[PF_TWO_STAGE_EN]. Both prefilter stages must be symmetric about the center tap. The coefficient buffer is designed to store one half of the filter coefficients, and it can store up to 192 filter coefficients per context. The number of coefficients places a limit on the maximum number of filter taps which can run per context (192x2=384). The coefficient memory is shared between the ST1 and ST2 filter stages, so the cumulative number of filter taps between these stages must remain below 384. The coefficients are stored in IEEE 64-bit floating point format by writing to the ASRC_PRE_COEFF_FIFO[COEFF_DATA] register. After each write to this register, the coefficient buffer write address will be automatically incremented. Writes to the coefficient buffer should be performed from tap 0 to the center tap in incrementing order. To reset the write pointer back to zero, the register field ASRC_CTX_CTRL_EXT1x[PF_COEFF_MEM_RST] can be toggled from 0x0 to 0x1 to 0x0. Stage1 coefficients are written by default, and stage2 coefficients can be written by setting the register field ASRC_CTX_CTRL_EXT1x[PF_COEFF_STAGE_WR]. When switching between stages, the address pointer should be reset back to 0x0 before performing a write.

14.5.2.1.4 ASRC Fractional Resampling Coefficient Generator

The fractional resampling coefficient generator is responsible for generating the resampling interpolation filters for each context. The resampling coefficient generator will calculate a new coefficient for each request from the context processing datapath. This process is computed on the fly for each sample group in flight. In addition, it will calculate the normalization factor which will be used to cancel out any dc gain incurred throughout the resampling process. Each coefficient represents an interpolated value from the coefficient RAM which is managed by this block. The format of the coefficient RAM is dependent on the number of taps used by the resampling process. Note, there is 1 coefficient RAM in the design. Any resampling filter characteristics will be applied to all contexts in the ASRC; however, each context can have an independent resampling ratio. The resampling filter can be configured to operate in 32, 64, or 128-tap mode by configuring the register field ASRC_CTX_RS_COEFF_CTRL[NUM_RES_TAPS]. In 32-tap mode the resampling coefficient RAM is organized as 16 taps x 128 phases/tap x 4 coefficients/phase x 64-bits/coefficient (64 kB). In 64-tap mode, the resampling coefficient RAM is organized as 32 taps x 64 phases/tap x 4 coefficients/phase x 64-bits/coefficient (64 kB). In 128-tap mode, the resampling coefficient RAM is organized as 64 taps x 32 phases/tap x 4 coefficients/phase x 64-bits/coefficient (64 kB). The coefficient RAM must be configured prior to beginning any context processing within the ASRC, and the RAM can be accessed by writing to ASRC_CTX_RS_COEFF_CTRL[RS_COEFF_WDATA].

14.5.2.1.5 ASRC Output Processing and Formatting

The output processing and formatting block serves the following purposes: fetching data from the context processing datapath, normalizing each sample with data provided by the resampling coefficient generator, implementing the TPDF dither algorithm, formatting samples according to user specifications, and managing the output sample circular queues. When a complete sample group from timestamp N has been processed by the context processing data paths, this block will perform a burst fetch to grab every output sample associated with the given context at time N. Each sample group has an associated normalization coefficient norm[CTX][N] which was generated during the resampling process. This coefficient is computed for every sample group output from the ASRC, and it must be applied to every sample associated with context the CTX at timestamp N. After the normalization procedure has been completed, the samples are then passed through the TPDF dither block. The dither function is enabled using the configuration register ASRC_CTX_OUT_CTRL[DITHER_EN]. Output formatting logic is implemented to allow for the following sample modifications: variable bits-per-sample ASRC_CTX_OUT_CTRL[BITS_PER_SAMPLE], bit-reversal of sample data ASRC_CTX_OUT_CTRL[BIT_REV], floating-point or fixed-point output ASRC_CTX_OUT_CTRL[FLOAT_FMT], and a variable sample position within a 32-bit

packed sample ASRC_CTX_OUT_CTRL[SAMPLE_POSITION]. Each sample group is written to the output circular queue which can be accessed through the ASRC_RDFIFOa registers.

14.5.2.2 Data Formatting and Dither

The ASRC can process signed fixed-point sample data which is represented as 16, 20, 24, or 32-bits per sample, and floating-point sample data formatted in IEEE single-precision floating point format (1 sign, 8 exponent, 23 mantissa). Note, all samples associated with a given context must have the same sample format. The ASRC will re-format all input samples to IEEE double-precision floating point format prior to running any calculations. The ASRC input sample Datapath is represented in the following block diagram. The sample position must be set to 0x0 and bits per sample must be set to 32 when operating on floating point input data.

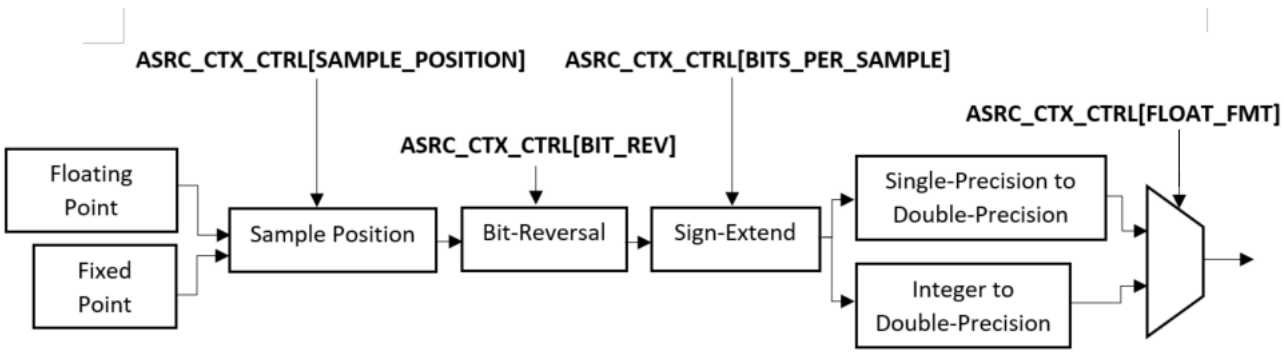


Figure 14-17. Input Sample Formatting Datapath

Samples output from the ASRC can be selected as 16, 20, 24, or 32-bits per sample with fixed or floating-point format. The input and output data formats are completely independent and can be selected in an arbitrary combination. This is due to the unified double-precision sample format which is utilized throughout the ASRC Datapath. There are 3 fixed point cases of interest which should be considered when determining how samples will be conditioned using different input and output bit-widths: up-conversion, down-conversion, and equivalent bit-width. When the input sample data has a lower number of bits per sample than the output data, this case is considered an up-conversion. During an up-conversion, the final output sample data will be multiplied by a constant to scale it up to the output bit width. As an example, if the input sample data is configured to contain 20 bits per sample, and the output data is configured to contain 32 bits per sample, then the ASRC output data will be multiplied by a constant factor of 2^(32-20) = 2^12. Clipping will be performed to ensure that output samples will not overflow beyond the output bit-width after this conversion. Note, any fractional resolution which was tracked throughout the ASRC Datapath will be retained during the up-conversion. When

the input sample data has a higher number of bits per sample than the output data, this case is considered a down-conversion. To perform a down-conversion, the output data LSBs will be truncated and optionally dithered.

TPDF dither can be optionally applied to the output samples of a context. To generate the TPDF dither values, two independent signed pseudo-random numbers are added together to form the final dither value. The dither is then scaled to one-half of the output LSB size and added to the audio sample data. Overflow detection is built into the hardware to ensure that an overflow doesn't occur after adding the TPDF dither to the final sample. A block diagram of the TPDF dither function is provided below. Note, dither is unsupported in 32 bit floating point output format.

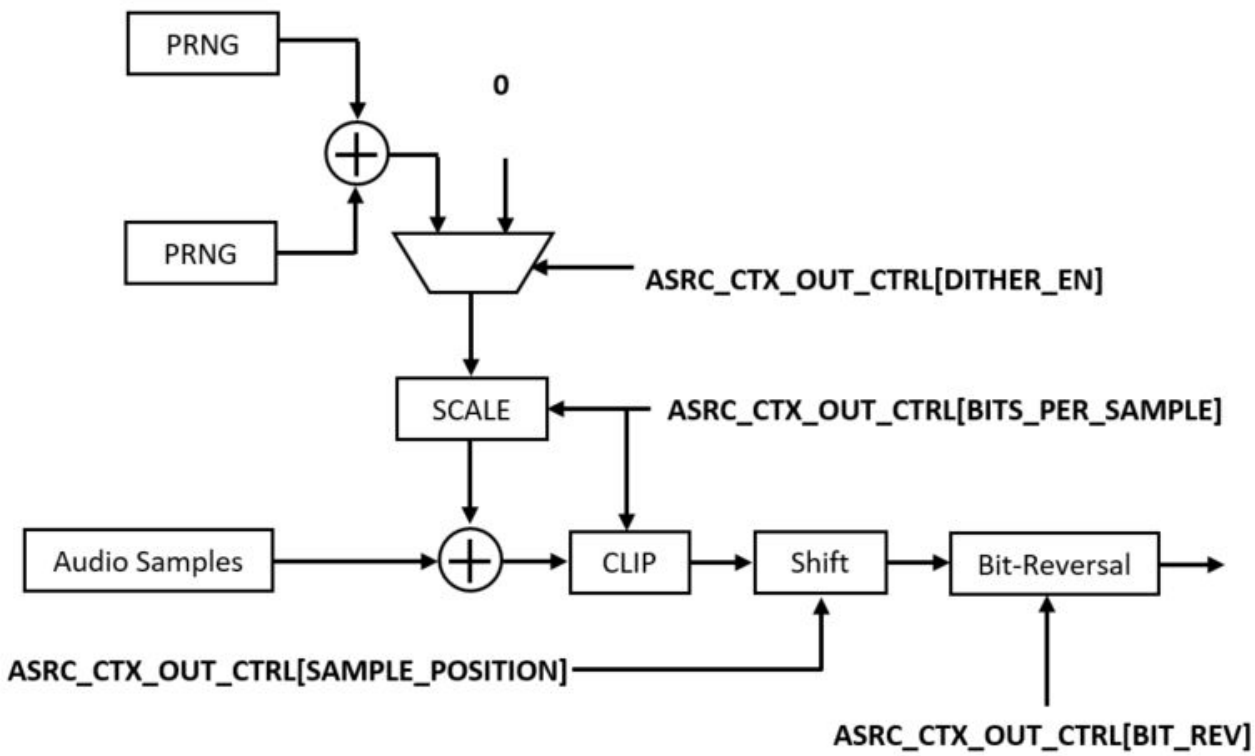


Figure 14-18. TPDF Dither and Output Formatting

14.5.2.3 ASRC FIFO ORGANIZATION

There are independent circular queues for each context in the ASRC. Each context is provided with an input circular queue (accessed through ASRC_WRFIFOa) and an output circular queue (accessed through ASRC_RDFIFOa). When writing data to be processed by the ASRC, each sample will be written to the ASRC_WRFIFOa register corresponding to the context which controls processing for the sample selected. The address for each ASRC_WRFIFO is automatically managed by the ASRC. The ASRC

formats each circular queue using a consistent sample interleaving pattern. This pattern ensures that all samples from a single time stamp for a given context (referred to as a Sample Group) are grouped together in sequential memory locations. This pattern is illustrated in the following figure. Note, this example shows the circular queue for Context0 with 6 channels enabled.

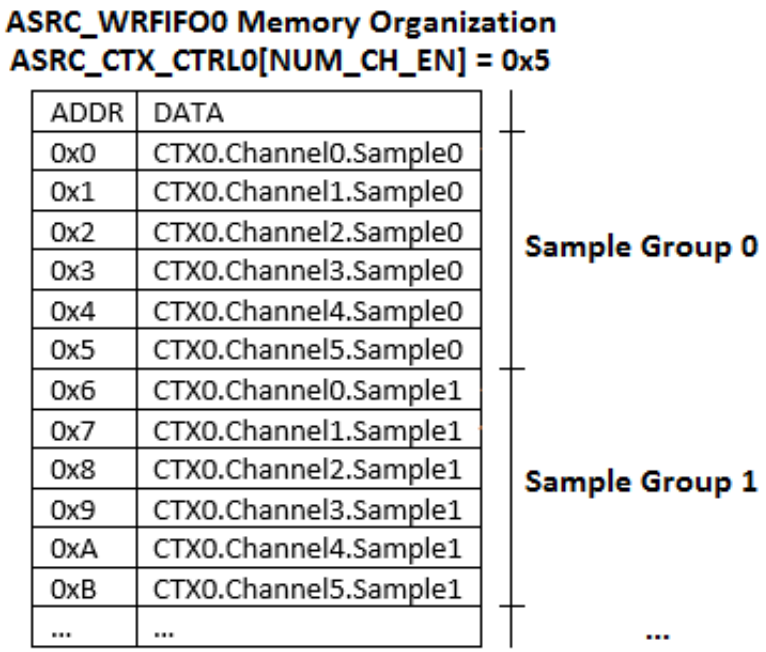


Figure 14-19. ASRC WRFIFO Organization

As shown in the figure above, Context0 is configured to contain 6 channels. Each channel will contain a single sample at time N (denoted CTX0.ChannelX.SampleN). The context circular queues are formatted to ensure that all samples corresponding to time N are contained in sequential memory locations. For example, channel 0-5 sample 0 will be contained in addresses 0x0-0x5 (Sample Group 0), channel 0-5 sample 1 will be contained in address 0x6-0xB (Sample Group 1), and so on in this order.

The ASRC provides interleaving support in hardware to ensure that a variety of sample sources can be internally combined to conform with this format. Interleaving parameters are accessed through the ASRC_CTRL_IN_ACCESSa and ASRC_CTRL_OUT_ACCESSa registers. The interleaving pattern is controlled using 3 register fields: GROUP_LENGTH, ACCESS_LENGTH, and ITERATIONS. This is intended to support hardware configurations which distribute a single context across samples from multiple audio sources. The following figure illustrates a use case where 3 audio sources are combined to be processed using a single ASRC context.

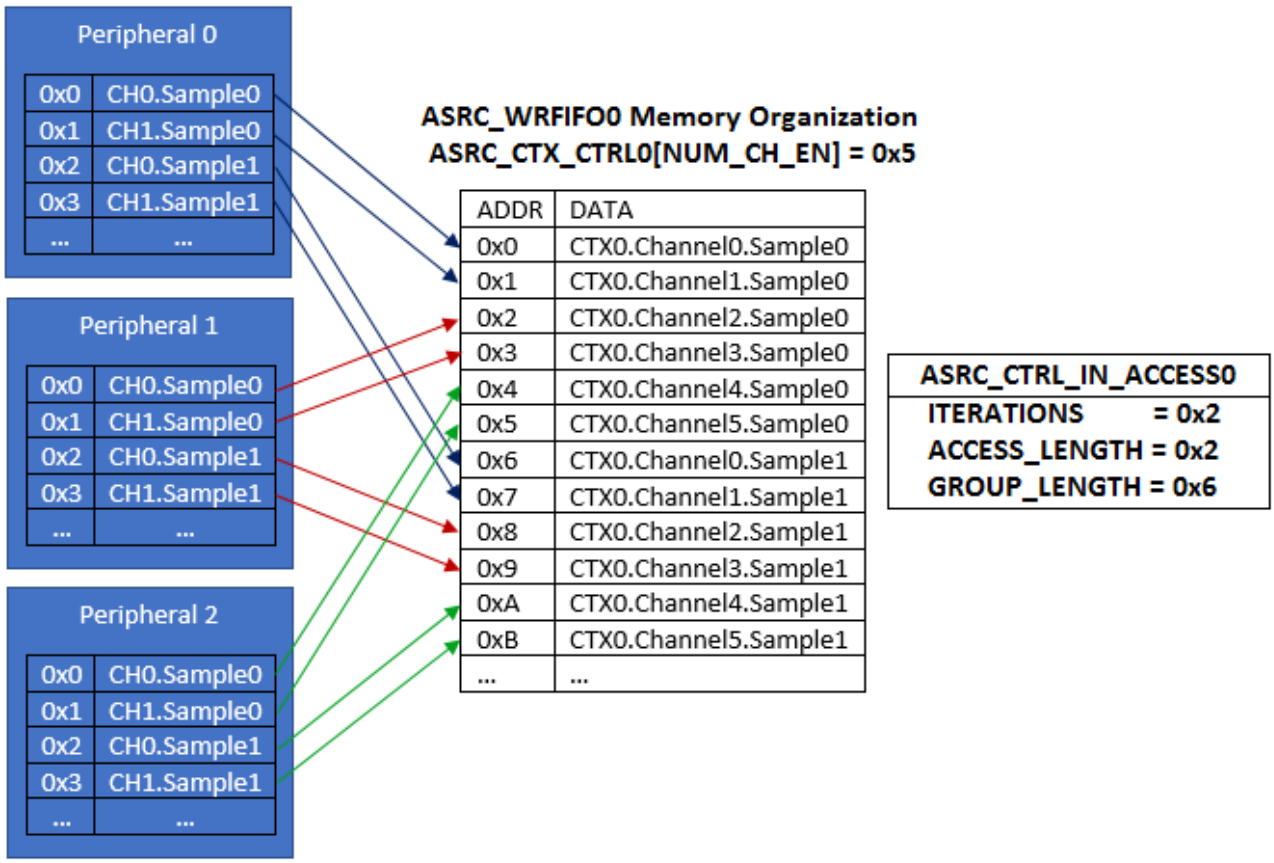


Figure 14-20. ASRC WRFIFO Interleaving Example

In the figure above, 3 audio sources (Peripheral0-2) will be processed by the ASRC using Context0. Each peripheral contains 2 channels of audio data (ACCESS_LENGTH = 0x2). The total number of channels contained across these 3 sources is 6 ((2 channels/source)*(3 sources); GROUP_LENGTH = 0x6). To make memory accesses more efficient, the burst length was configured to fetch 4 samples per access to a peripheral. This pattern will fetch 2 samples per channel on each peripheral access (ITERATIONS = 0x2). The ASRC input interleaving hardware will control the ASRC_WRFIFO0 memory address to increment in the following order: 0x0, 0x1, 0x6, 0x7, 0x2, 0x3, 0x8, 0x9, 0x4, 0x5, 0xA, 0xB. This will ensure that sample groups are separated sequentially in the WRFIFO memory space.

The ASRC processes data on a sample group granularity. Prior to running any SRC calculations, the ASRC will ensure that a complete sample group has been written to the WRFIFO. The ASRC_FIFO_STATUSa registers provide visibility into how many sample groups are stored in the WRFIFO. This can be read using the field NUM_SAMPLE_GROUPS_IN. Note, this register field will only be incremented once a complete sample group has been written to the FIFO. Utilizing the example above, this would occur after a write to addresses 0x5 and 0xB. As illustrated with this example, the number of writes required to increment the NUM_SAMPLE_GROUPS_IN status is

dependent on the ASRC_CTRL_IN_ACCESS register parameters. Each WRFIFO has a watermark status with dma request support. The watermark is configured with ASRC_CTX_CTRL[FIFO_WTMK], and the dma request can be enabled with ASRC_CTX_CTRL[FWMDE]. If the number of sample groups stored in the WRFIFO is less than the FIFO_WTMK then the ASRC_SAMPLE_FIFO_STATUS[INFIFO_WTMK] will be set, and a dma transfer will be requested (if enabled).

The examples outlined in this section were illustrated for the input WRFIFO; however, the same concepts can be applied to the output RDFIFO. The output RDFIFO contains a sample group count (ASRC_FIFO_STATUSa[NUM_SAMPLE_GROUPS_OUT]), a watermark (ASRC_CTX_OUT_CTRL[FIFO_WTMK]) with associated dma request (ASRC_CTX_OUT_CTRL[FWMDE]) and status flag (ASRC_FIFO_STATUSa[OUTFIFO_WTMK]), and output interleaving parameters (ASRC_CTRL_OUT_ACCESS). The RDFIFO address is automatically managed by the ASRC output interleaving hardware, and the NUM_SAMPLE_GROUPS_OUT is only updated when a complete sample group is written or read. Note, data is written to the RDFIFO one sample group at a time by the ASRC. However, data may be read across sample groups depending on the output interleaving parameters. This makes it possible to request data from an empty RDFIFO element by indexing the read address beyond the number of sample groups stored in the RDFIFO. If a read is requested from an empty element, then the error flag ASRC_FIFO_STATUSa[OUTFIFO_EMPTY_RD] will be triggered. The following example illustrates the case where an empty RDFIFO element is accessed.

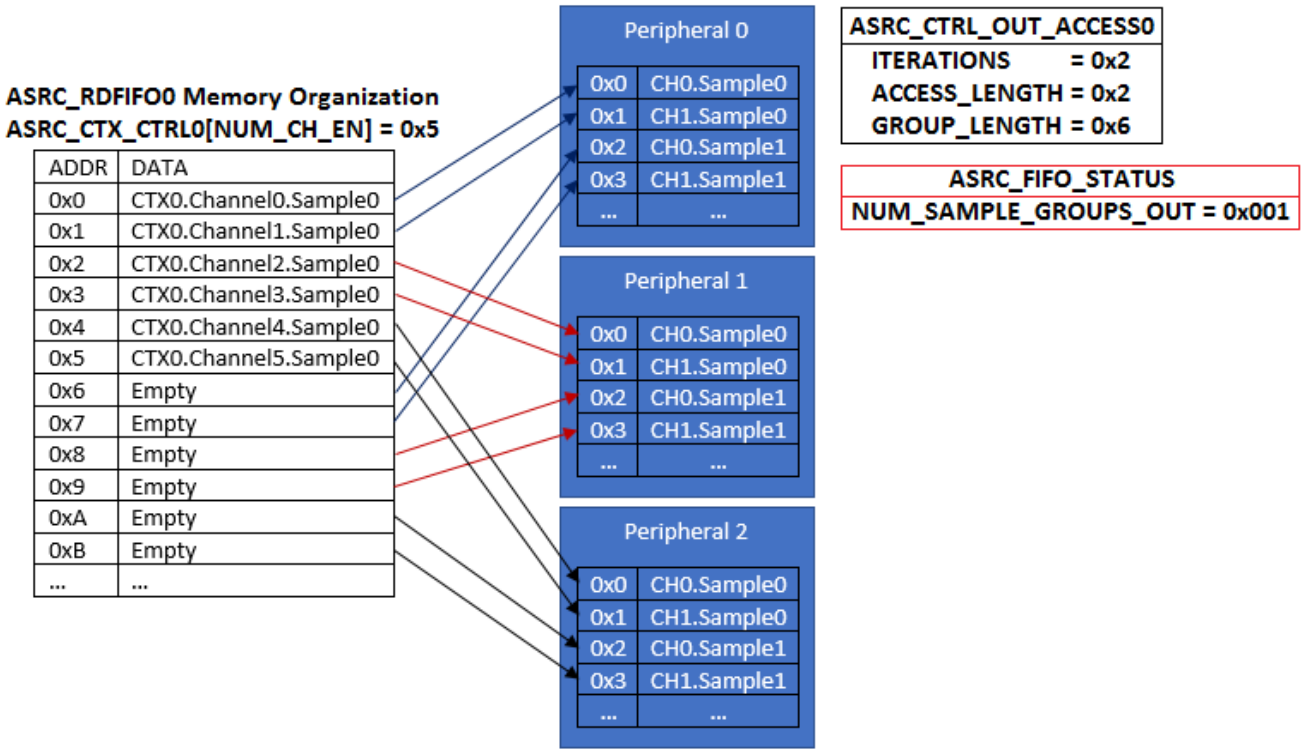


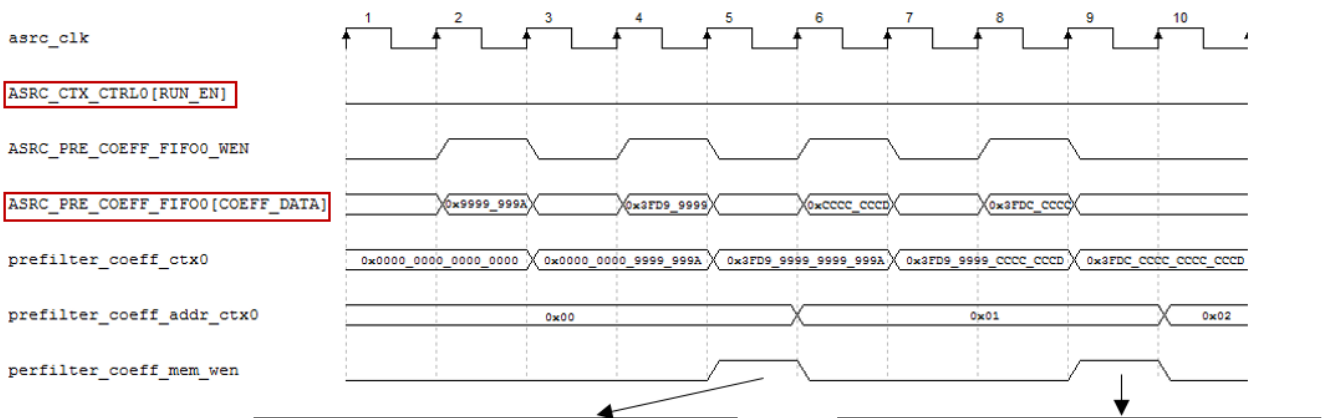
Figure 14-21. ASRC RDFIFO Empty Request

In the above example, the ASRC has completed processing for sample group 0 and has written this data to the RDFIFO in addresses 0x0-0x5. In addition, the ASRC_FIFO_STATUS is updated to indicate that 0x001 sample groups are stored in the RDFIFO. The RDFIFO access parameters are configured with 0x2 ITERATIONS and 0x2 ACCESS_LENGTH. For a complete burst read to be fulfilled, the RDFIFO would need to contain 0x002 sample groups. If a burst access is attempted under these conditions, then the first empty request will be made from address 0x6 after 2 completed reads from 0x0 and 0x1. When the empty read request is made, the OUTFIFO_EMPTY_RD flag will be asserted for Context0, stale data will be read, and the RDFIFO internal read address will remain unmodified. The internal read address will only change when valid data has been read from the RDFIFO. Care should be taken to avoid this scenario by utilizing the RDFIFO watermark and dma request feature.

14.5.2.4 ASRC Coefficient Memory

The ASRC contains memory instances to store coefficients for both the prefilter and resampler. All coefficients within the ASRC are stored in 64-bit IEEE floating point format (1-sign, 11-exponent, 52-mantissa).

The number of prefilter taps is configurable from 1 to 384 taps using the registers ASRC_CTX_CTRL_EXT2x[ST1_NUM_TAPS] and ASRC_CTX_CTRL_EXT2x[ST2_NUM_TAPS]. Both prefilter stages must be symmetric around the center tap, so the prefilter coefficient memory is sized to contain a maximum of 192 entries. The prefilter coefficients can be programmed individually for each context in the design. To modify the value of a prefilter coefficient, the user must perform a write to the register ASRC_PRE_COEFF_FIFOn[COEFF_DATA] while the respective context n RUN_EN bit is set to 0b0. The register bit ASRC_CTX_CTRL_EXT1x[PF_COEFF_STAGE_WR] can be controlled to select which filter stage coefficients will be modified (ST1 or ST2). Two 32-bit writes are required to update a single 64-bit coefficient value. After each coefficient write has been performed, the memory index will be automatically incremented. The following example illustrates 4 writes to the ASRC_PRE_COEFF_FIFO for context 0 ST1 to configure the coefficient values for tap 0 and tap 1. Software controlled signals are outlined in red.



Address	Data
0x00	0x3FD9_9999_9999_999A
0x01	X
...	...
0x3F	X

Address	Data
0x00	0x3FD9_9999_9999_999A
0x01	0x3FDC_CCCC_CCCC_CCCD
...	...
0x3F	X

Figure 14-22. Prefilter Coefficient Access

The above example illustrates the programming sequence required to initialize the first two taps of the ST1 prefilter coefficient FIFO for CTX0 (ASRC_PRE_COEFF_FIFO0[COEFF_DATA]). Note, it is assumed that the write which occurs on asrc_clk cycle 2 is the first write to the context prefilter FIFO out of reset. To indicate this, the prefilter_coeff_addr_ctx0 is initialized to 0x00, which acts as the pointer to the prefilter coefficient for tap 0. After the 4th write has occurred (post clock cycle 9), the context0 prefilter coefficient FIFO will contain the floating-point representation of 0.4 in tap0 and 0.45 in tap1. On clock cycle 10, the coefficient fifo address will be equal to 0x02. This indicates that the next write sequence will be used to modify tap2 of the prefilter. To reset this address back to 0x00, the user must toggle the associated prefilter

pointer reset (this example would use ASRC_CTX_CRTL_EXT10[PF_COEFF_MEM_RST]) or continue performing writes until the address rolls over back to 0x00. By default, the coefficients for prefilter ST1 will be updated. To perform a write for the ST2 coefficients, set the PF_COEFF_STAGE_WR bit to 0x1 for the context to be modified. The prefilter coefficient address pointer should always be reset using PF_COEFF_MEM_RST prior to writing any coefficient values.

There is one resampling coefficient memory within the ASRC design. This memory is designed to store the coefficients used to perform a cubic spline interpolation of a sinc function with NUM_RES_TAPS/2 zero crossings between tap 0 and tap NUM_RES_TAPS/2 inclusive. The organization of this memory is dependent on the number of taps used in the resampling stage (ASRC_CTX_RS_COEFF_CTRL[NUM_RES_TAPS]). The peak of the sinc function should occur at tap NUM_RES_TAPS/2. It should be documented that the center tap of the sinc function lies outside of the coefficient RAM, and it will be programmed directly to the registers ASRC_RS_CT_HIGH and ASRC_RS_CT_LOW. Note, the coefficient RAM is sized to store one half of the full windowed sinc function under the assumption that the sinc function is symmetric around tap NUM_RES_TAPS/2. When the filter is programmed to run in 32-tap mode, the memory will be organized with 16-taps, 128-phases, and 4-coefficients per phase. This organization is illustrated in the image below.

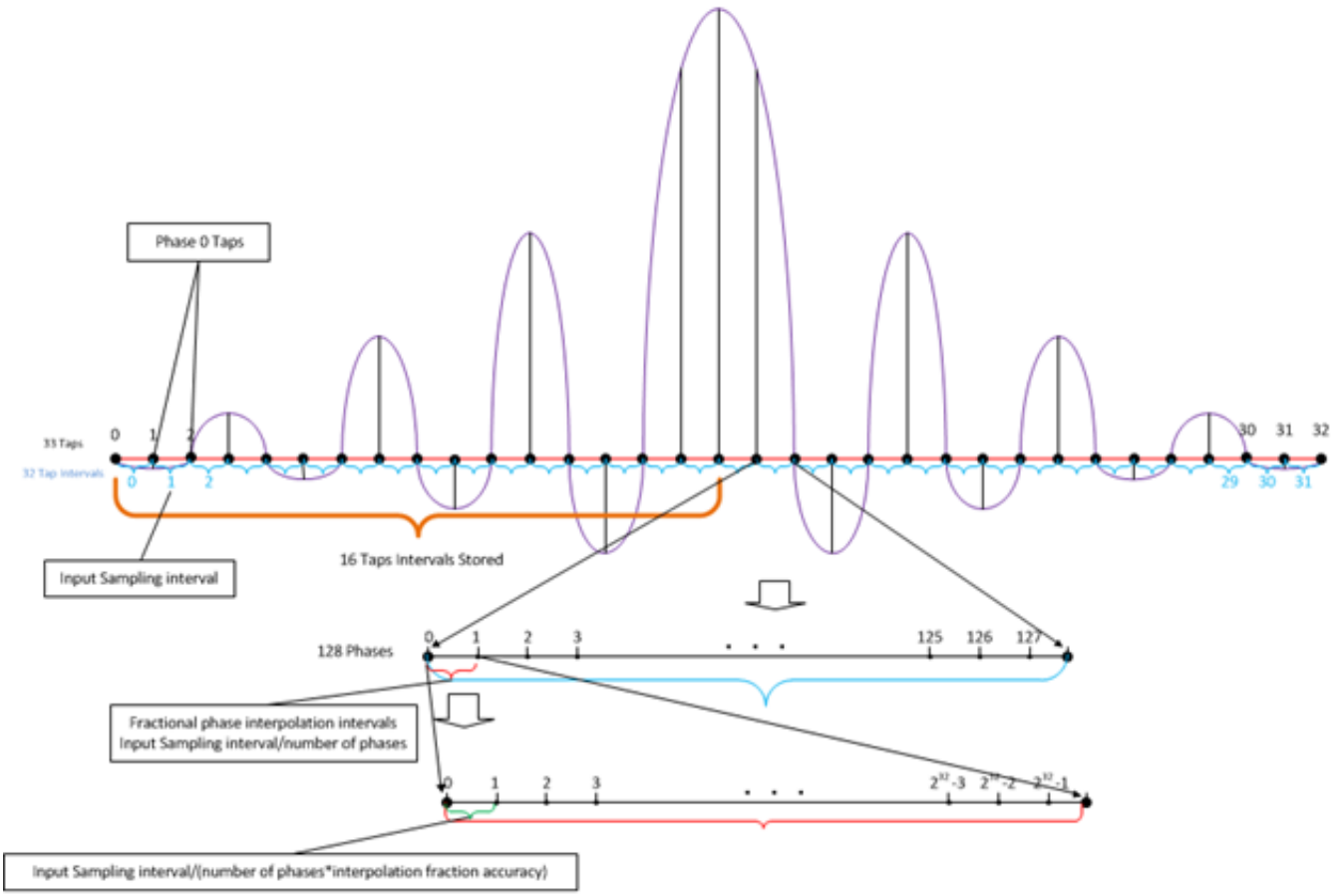


Figure 14-23. Resampling Coefficient Organization

In 32-tap mode, software will need to program 16x128 elements of RAM, where each element contains the 4 64-bit floating point coefficients used to perform the cubic spline interpolation. This requires a total of 16x128x8 writes to the ASRC_CTX_RS_COEFF register. Each element in RAM is organized as follows:

Table 14-10. Resampling Coefficient RAM Element

Address	coeff3[63:0]	coeff2[63:0]	coeff1[63:0]	coeff0[63:0]

The resampling coefficient generator will perform the cubic spline interpolation using the following equation:

resampler_coeff_out(x) = coeff0*x^3 + coeff1*x^2 + coeff2*x + coeff3

In this case, x will be equal to a number between 0 and 2^32-1, where 0 indicates that x lies exactly on the lower interpolation phase and 2^32-1 lies directly to the left of the higher interpolation phase. In this equation, x essentially represents the distance between the two nearest phases in the resampling coefficient RAM. In 64-tap mode, the resampling coefficient ram is organized as 32-taps, 64-phases, and 4-coefficients per phase. In this mode software will need to program 32x64 elements of RAM, and the

coefficient values will be calculated according to the new 64-tap organization. In 128-tap mode, the resampling coefficient ram is organized as 64-taps, 32-phases, and 4-coefficients per phase. The following timing diagram provides an example of programming the first element of the resampling coefficient RAM.

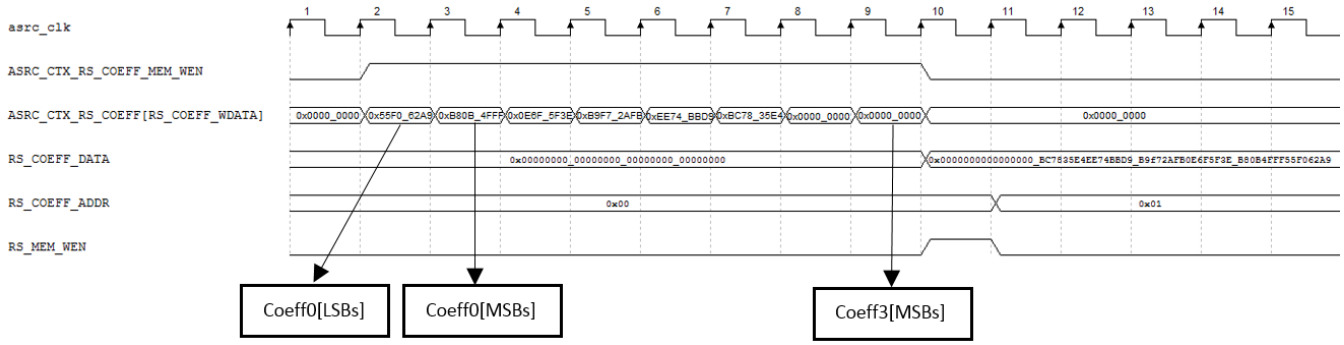


Figure 14-24. ASRC RDFIFO Empty Request

This example shows 8 consecutive writes being performed to the ASRC_CTX_RS_COEFF[RS_COEFF_WDATA] register field. It is assumed that this example shows the first writes to the resampling coefficient memory out of reset. This is illustrated by the RS_COEFF_ADDR being initialized to 0x00. After the 8 writes have been completed, element 0x00 of the resampling coefficient memory will be stored, and the write address will increment to 0x01. The writes are always performed from coeff0 to coeff3 with LSBs leading the MSBs. The RS_COEFF_ADDR can be reset by toggling the state of ASRC_CTX_RS_COEFF_CTRL[RS_COEFF_PTR_RST] or by performing the full 16x128x8 writes to RS_COEFF_WDATA until the address rolls over back to 0x00.

14.5.2.5 Resampling and Normalization

The resampling process works by performing a sinc interpolation of the input samples to generate the output samples. To accomplish this, the fractional resampling coefficient generator computes an interpolation filter for every sample group which is output from the ASRC. The interpolation filter generated is dependent on the following factors: the resampling ratio, the timestamp for the output sample being generated, the cubic spline interpolation coefficients, and the number of taps used in the resampling stage. The resampling ratio is calculated by dividing the input sample frequency Fsin by the output sample frequency Fsout. In 32 tap mode, this ratio is represented as a 44-bit fixed point number with 5 integer bits and 39 fractional bits. The resampling ratio is broken up internally into three component parts:

1. Whole number portion (INT) : rs_ratio[43:39]
2. Integer Phase (IP) : rs_ratio[38:32]

3. Fractional Phase (FP) : rs_ratio[31:0]

To understand how the resampling coefficient generator functions, these three component parts must first be understood. The INT portion of the resampling ratio is used to determine how many input samples need to be shifted into/out of the resampling filter per every output sample generated. For example, if the resampling ratio is selected to be 2.0 (Fsin = 2*Fsout) then the INT will be equal to 2, and the resampler will need to shift in 2 new samples per every output sample generated. The INT is passed from the resampling coefficient generator to the resampler within the context processing datapath for every sample group output to indicate the number of new samples to shift into/out of the buffer.

When the IP portion and FP portion are concatenated {IP,FP}, this value represents the fractional time difference between the output sample being generated and the nearest input sample. This is represented by the following diagram.

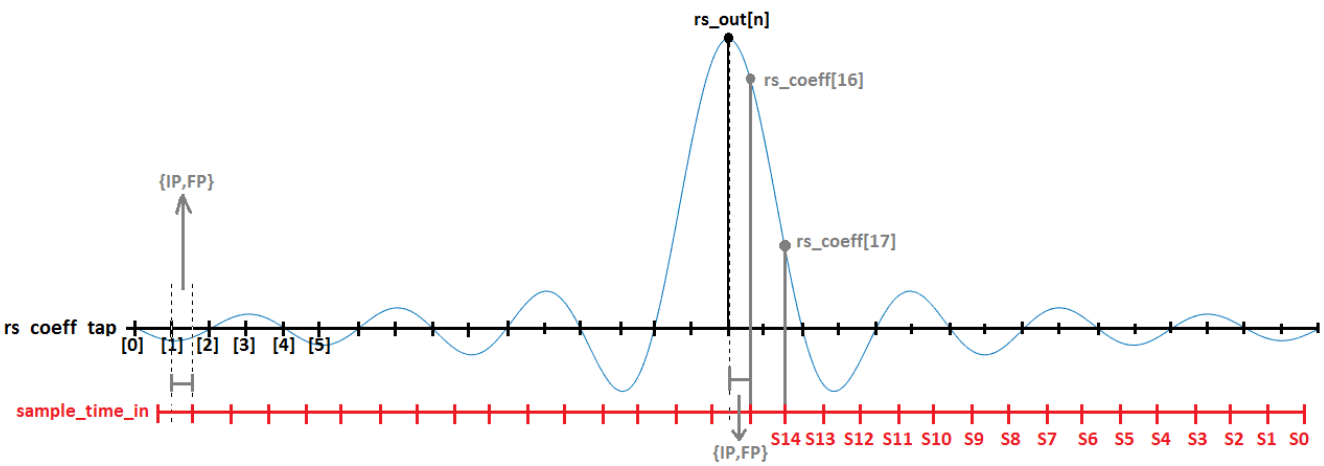


Figure 14-25. Resampling Coefficient Generation

In this example, the resampler is performing an iteration to compute the output sample rs_out[n] (center of the sinc function). This sample lies on a fractional time offset from the input sample time, and this offset is represented by {IP,FP}. Ideally, the resampler would compute the coefficients rs_coeff[x] exactly by sampling the sinc interpolation function at the time offset represented by {IP,FP}. To store all possible samples of the sinc function required for the 39-bit fractional time offset {IP,FP} would require 2^(39) sample points between each of the rs coeff tap points. To implement this practically a coefficient RAM is used to store cubic spline coefficients for the sinc function. Each interval between the rs coeff tap points is divided into 128 sections which are indexed using the IP portion of the fractional time difference. The IP portion is split from this fraction and used to determine which phase of the filter tap is being fetched from the resampling coefficient RAM. Note, the section "ASRC Coefficient Memory" provides an overview of how the RAM is organized. In summary, this RAM is split into the following

component sections when operating in 32-tap mode: 16 taps x 128 phases per tap. The cubic spline coefficients are stored in RAM for each of the 16x128 knots in the sinc function.

Once the IP portion of the fractional time difference is utilized to fetch the proper cubic spline coefficients, the FP portion is used to interpolate the rs_coeff[x] value. There are 4 cubic spline coefficients stored per phase in the resampling coefficient RAM (coeff[3],coeff[2],coeff[1],coeff[0]). The final rs_coeff[x] value is interpolated by the following equation:

rs_coeff[x] = coeff[0] * FP^3 + coeff[1]* FP^2 + coeff[2] * FP + coeff[3]

The IP portion is used to divide the interval between resampling coefficient taps into 128 phases, and the FP portion is used to divide the phases into 2^32 sections by using cubic spline interpolation. This method provides a high degree of accuracy for computing the proper coefficient value at time offset {IP,FP}.

After each output sample is calculated, the internal resampling ratio is accumulated, and the coefficient generation process is repeated. This is equivalent to incrementing the time base of the resampler to the location of the next output sample. If this accumulation results in the INT being greater than 0 then INT sample groups are shifted into/out of the resampling buffer for the next computation. In addition, the fractional time offset {IP,FP} will increase to the fractional location of the next output sample. To prevent the resampling process from generating a dc gain on the output sample data, the resampling coefficients must accumulate to a value which is equal to 1. However, due to minor errors which may be produced in the interpolation process, the accumulated coefficient values could potentially result in a value which is less than or equal to 1. This requires a normalization stage which divides the resampler output data by the sum of all resampling coefficients used to generate each sample. This normalization occurs in the output processing and formatting block.

The initial resampling ratio configured through the ASRC_RS_RATIO_LOWx and ASRC_RS_RATIO_HIGHx registers must be static while the target context is actively processing data. Updates to the internal resampling ratio can be made to an active context through the ASRC_RS_UPDATE_CTRLx registers. Note, any modification to the internal resampling ratio is applied during the next available output sample time. These registers provide software control over the resampling ratio ramp rate and fine adjustment amount. To prevent distortion when making large adjustments to the resampling ratio, an upper limit can be placed on the resampling ratio adjustment amount through the RS_RATIO_RAMP_RATE register field. This will place a cap on the maximum amount of change which can be applied to the resampling ratio per output sample time (this is an absolute value of the maximum change allowed). The desired resampling rate adjustment is programmed to the RS_RATIO_MOD register field. Pending updates are accumulated and applied to the resampling ratio depending on the ramp rate and output sample time.

The following flowcharts illustrate how RS_RATIO_MOD and RS_RATIO_RAMP_RATE are utilized to adjust the internal resampling ratio RS_RATIO. Note RS_RATIO is initialized to the concatenation of {RS_RATIO_HIGH,RA_RATIO_LOW} while the context is in reset. Sample Out RDY is a flag which triggers after every output sample time.

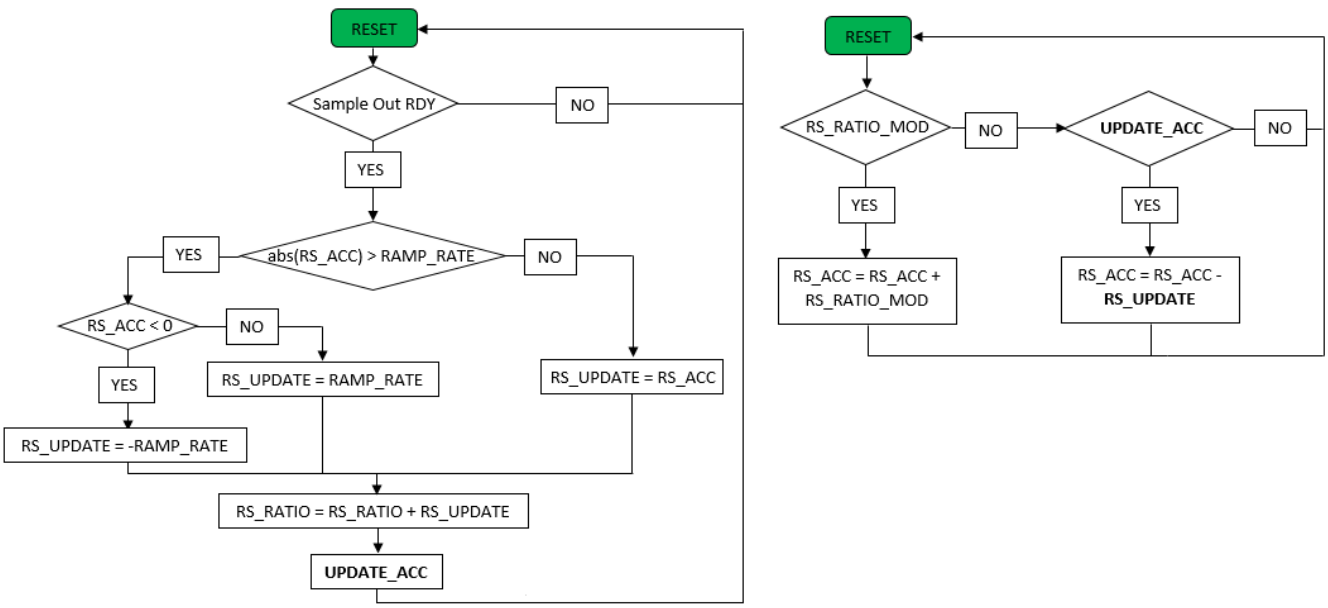


Figure 14-26. Resampling Coefficient Generation

14.5.2.6 ASRC Context Processor Initialization

The following section provides a programming example which initializes 4 contexts to run on the ASRC context processing datapath. The configuration of prefilter and resampling coefficient RAMs will not be included in this example. This configuration step is outlined in section ASRC Coefficient Memory and must be completed prior to running any conversions with the ASRC. The prefilter configuration registers will be outlined in the section ASRC Prefilter Configuration and Performance.

In this example, context 0 (CTX0) will be configured to run 4 channels, CTX1 will be configured to run 6 channels, CTX2 will be configured to run 5 channels, and CTX3 will be configured to run 12 channels

One possible first step to initialize a context is to determine where to distribute the channels within the design. A single context can be split amongst any of the 4 context processing pipes in the design. The user can poll the ASRC_PROC_CTRL_SLOTx_R0x registers to see what is available when determining where to place a context. Two conditions must be met for a context processor to be considered available:

1. At least one of the following two bits must be set to zero: SLOT1_EN or SLOT0_EN, or the context configured in one of the slots is not running. To determine if a slot context is running, first read the ASRC_PROC_CTRL_SLOT1_R0x[SLOT1_CTX_NUM] and ASRC_PROC_CTRL_SLOT0_R0x[SLOT0_CTX_NUM] registers to see which contexts are selected for each slot. Next, read the associated ASRC_CTX_CTRLx[RUN_EN] bits to determine if each context is active.
2. The total number of channels consumed within the context processor must be less than or equal to 8. To determine how many channels are in use, SLOT0_NUM_CH and SLOT1_NUM_CH can be accumulated (note, these are zero indexed). The number of channels consumed by a slot with SLOTx_EN = 0 is always 0.

In this example, the ASRC is unused at the time of configuring context 0. This means that every context processor in the design is available (All SLOTx_EN bits are low and all RUN_EN bits are low). The initial configuration of the ASRC datapath is represented in the following diagram.

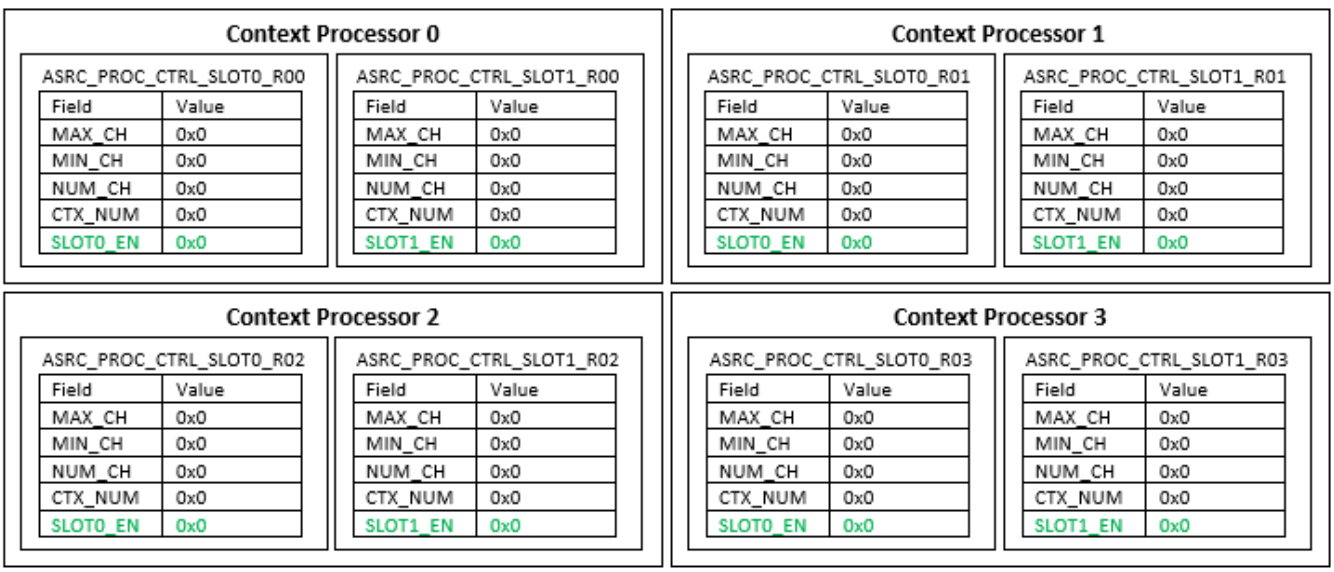


Figure 14-27. Context Processor Initialization State

The placement of a context is arbitrary if the resources are available. However, the minimum number of SLOTs per context should be chosen to ensure maximum utilization of available resources. In this case, context 0 will be placed on datapath processor 3 SLOT0. To program this configuration, the following register settings will be selected:

1. ASRC_PROC_CTRL_SLOT0_R03[SLOT0_EN] = 0x1: This slot is in use
2. ASRC_PROC_CTRL_SLOT0_R03[SLOT0_CTX_NUM] = 0x0 : CTX0 will run here
3. ASRC_PROC_CTRL_SLOT0_R03[SLOT0_NUM_CH] = 0x3 : 4 channels will be used by this slot

4. ASRC_PROC_CTRL_SLOT0_R03[SLOT0_MIN_CH] = 0x0 : CTX0 channel range lower boundary set to 0
5. ASRC_PROC_CTRL_SLOT0_R03[SLOT0_MAX_CH] = 0x3 : CTX0 channel 0 (SLOT0_MIN_CH) to channel 3 will run here

After programming the ASRC_PROC_CTRL_SLOT*_R0* registers, the prefilter memories will need to be allocated for each SLOT in the processor. This allocation is dependent on the number of prefilter stages, the number of taps per stage, and the prefilter ST1 expansion factor selected. A detailed programming example will cover prefilter initialization in the "ASRC Prefilter Initialization" section. Note, the following top-level Context 0 registers will need to be programmed prior to running the context:

1. ASRC_CTX_CTRL_EXT10
2. ASRC_CTX_CTRL_EXT20
3. ASRC_CTRL_IN_ACCESS0
4. ASRC_CTX_OUT_CTRL0
5. ASRC_CTRL_OUT_ACCESS0
6. ASRC_CHANNEL_STATUSx0
7. ASRC_RS_RATIO_LOW0
8. ASRC_RS_RATIO_HIGH0

After each of these registers has been configured, the last step is to configure the register fields within ASRC_CTX_CTRL0, and set the ASRC_CTX_CTRL0[RUN_EN] = 0x1. This will allow CTX0 to begin processing any samples which are written to ASRC_WRFIFO0[CTX_WR_DATA]. This same process should be followed when setting up the remaining channels from CTX1, CTX2, and CTX3. The following diagram illustrates one possible programming for this context set:

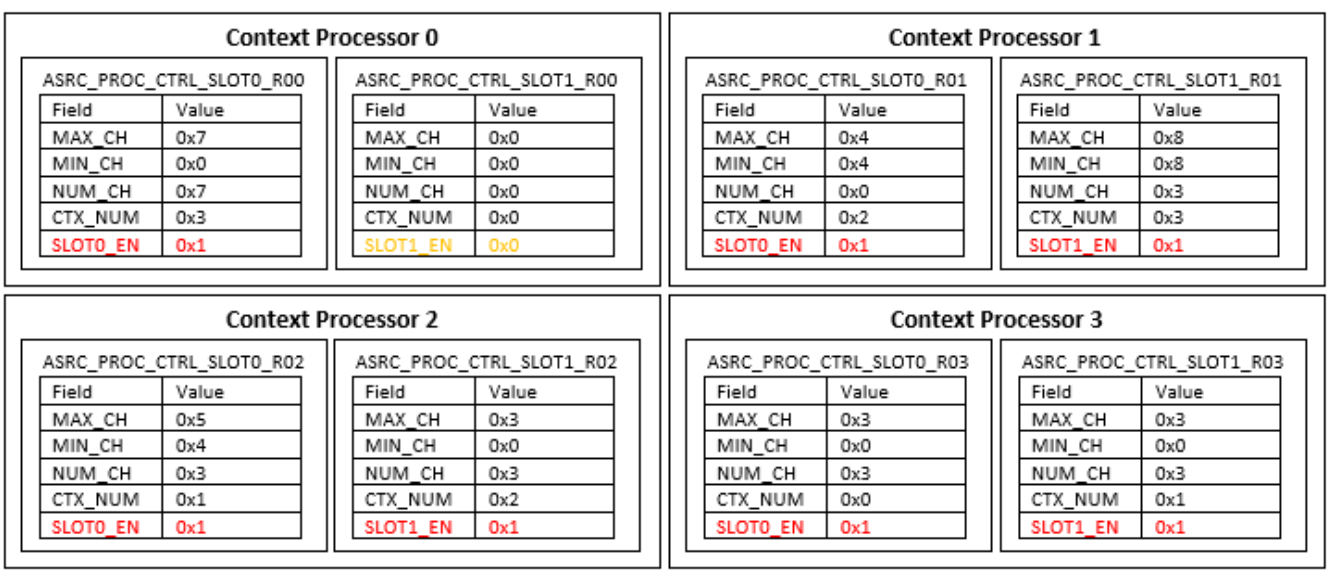


Figure 14-28. Context Processor Configuration Result

To summarize, CTX0 is configured to run 4 channels on processor 3 (PROC3) SLOT0, CTX1 is configured to run 4 channels on PROC3 SLOT1 and 2 channels on PROC2 SLOT0, CTX2 is configured to run 4 channels on PROC2 SLOT1 and 1 channel on PROC1 SLOT0, and CTX3 is configured to run 8 channels on PROC0 SLOT0 and 4 channels on PROC1 SLOT0. As a final note, when a context is shut down, all the context processor slots which were in use must be initialized back to 0x0 to ensure resources are available for the next configuration.

14.5.2.7 ASRC Prefilter Configuration and Performance

Each ASRC context processor contains a prefilter which can be configured to run in a single or double stage mode. The single stage prefilter configuration will be implemented using a FIR structure, and the double stage prefilter configuration will be implemented using an IFIR structure. Both the stage1 (ST1) and stage2 (ST2) prefilter share a single sample memory. This memory structure (PF Sample Memory) contains a total of 6144 elements (1 element = 1 sample).

Samples for the ST1 stage are received directly from the ASRC Input Write FIFO, and samples for the ST2 stage are received from the ST1 Writeback path. The ST1 prefilter has a programmable number of taps (ASRC_CTX_CTRL_EXT2[ST1_NUM_TAPS]), and a programmable expansion factor (ASRC_CTX_CTRL1[PF_EXPANSION_FACTOR]). The ST2 prefilter has a programmable number of taps (ASRC_CTX_CTRL_EXT2[ST2_NUM_TAPS]) with an implied expansion factor of 0. A simplified block diagram for the ASRC Prefilter is shown in the following block diagram. Note, this structure is instantiated one time in each context processor.

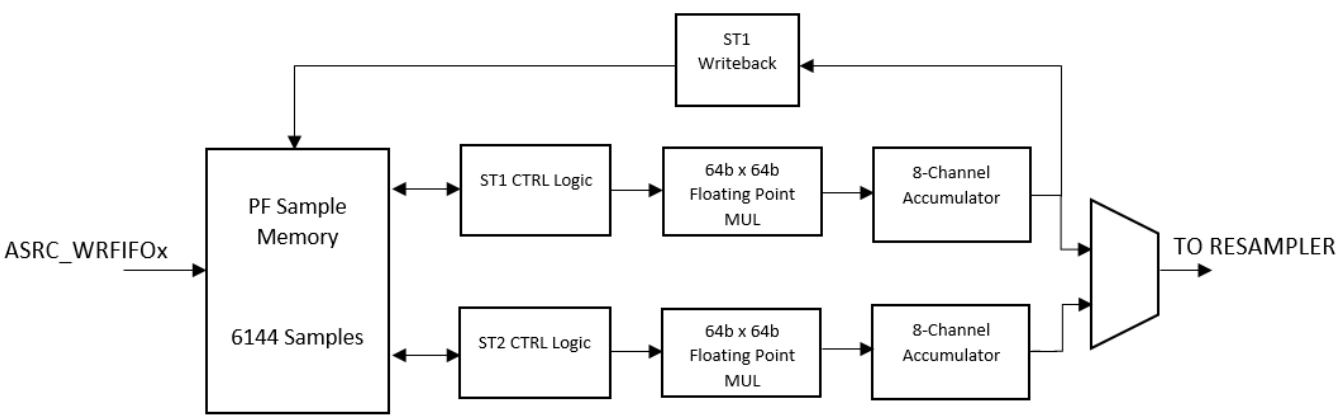


Figure 14-29. Prefilter Block Diagram

The PF Sample Memory partitions must be specified by software using the ASRC_PROC_CTRL registers. The following procedure can be followed to determine the amount of memory which will be used by each filter stage. This example illustrates the case for a two stage prefilter. In addition, it is assumed that the context being programed has already been placed in a context processor. Since there is 1 prefilter per context processor, this procedure must be followed by every processor where the context is running.

1. Program the context to use a two layer prefilter:

   ASRC_CTX_CTRL_EXT1x[PF_TWO_STAGE_EN] = 0x1

2. Select the number of taps used by ST1:

   ASRC_CTX_CTRL_EXT2x[ST1_NUM_TAPS]

3. Select the expansion factor used by ST1:

   ASRC_CTX_CTRL_EXT1x[PF_EXPANSION_FACTOR]

4. Determine the number of channels which will be using the prefilter in each context processor where the selected context will be enabled. If the context is in SLOT0, the number of channels will be equal to:

   NUM_CH = SLOT0_NUM_CH + 1

5. Calculate the amount of prefilter memory which will be used by ST1:

   EXP = ASRC_CTX_CTRL_EXT1x[PF_EXPANSION_FACTOR] + 1

   NUM_S1_TAPS = ASRC_CTX_CTRL_EXT2x[ST1_NUM_TAPS]

   ST1_MEM_ALLOC = NUM_S1_TAPS*NUM_CH*EXP + NUM_CH

   ASRC_PROC_CTRL_SLOTx_R3x[SLOTx_ST1_MEM_ALLOC] = ST1_MEM_ALLOC

6. Compute the CHANxEXP parameter which is used by the prefilter ST1 addressing logic:

   CHANxEXP =NUM_CH * ASRC_CTX_CTRL_EXT1x[PF_EXPANSION_FACTOR]

   ASRC_PROC_CTRL_SLOTx_R2x[SLOTx_ST1_CHANxEXP] = CHANxEXP

7. Calculate the amount of prefilter memory which will be used by ST2:

   NUM_S2_TAPS = ASRC_CTX_CTRL_EXT2x[ST2_NUM_TAPS] + 1

   ST2_MEM_ALLOC = NUM_CH * NUM_S2_TAPS

ASRC_PROC_CTRL_SLOTx_R3x[SLOTx_ST2_MEM_ALLOC] = ST2_MEM_ALLOC

After SLOTx[ST1_MEM_ALLOC] and SLOTx[ST2_MEM_ALLOC] have been calculated, a base address will need to be specified to determine the location to place each queue in the Prefilter Sample Memory. The base address for each queue is programmed through the ASRC_PROC_CTRL_SLOTx_R2x[SLOTx_ST1_ST_ADDR] and ASRC_PROC_CTRL_SLOTx_R3x[SLOTx_ST2_ST_ADDR] registers. Software must ensure that queues do not overlap within the memory. One method to determine the base address placement is to keep SLOT0 queues incrementing from the min address (0x00) and SLOT1 queues decrementing from the max address (6143). For example, the SLOT0 ST1 base address can be set to 0x0, and the SLOT0 ST2 base address can be set equal to SLOT0[ST1_MEM_ALLOC]. Next, the base address for SLOT1 ST1 can be placed at (6144 - SLOT1[ST1_MEM_ALLOC]), and the base address for SLOT1 ST2 can be placed at (6144 - SLOT1[ST1_MEM_ALLOC] - SLOT1[ST2_MEM_ALLOC]). The following diagram illustrates the one potential Prefilter Sample Memory queue partition.

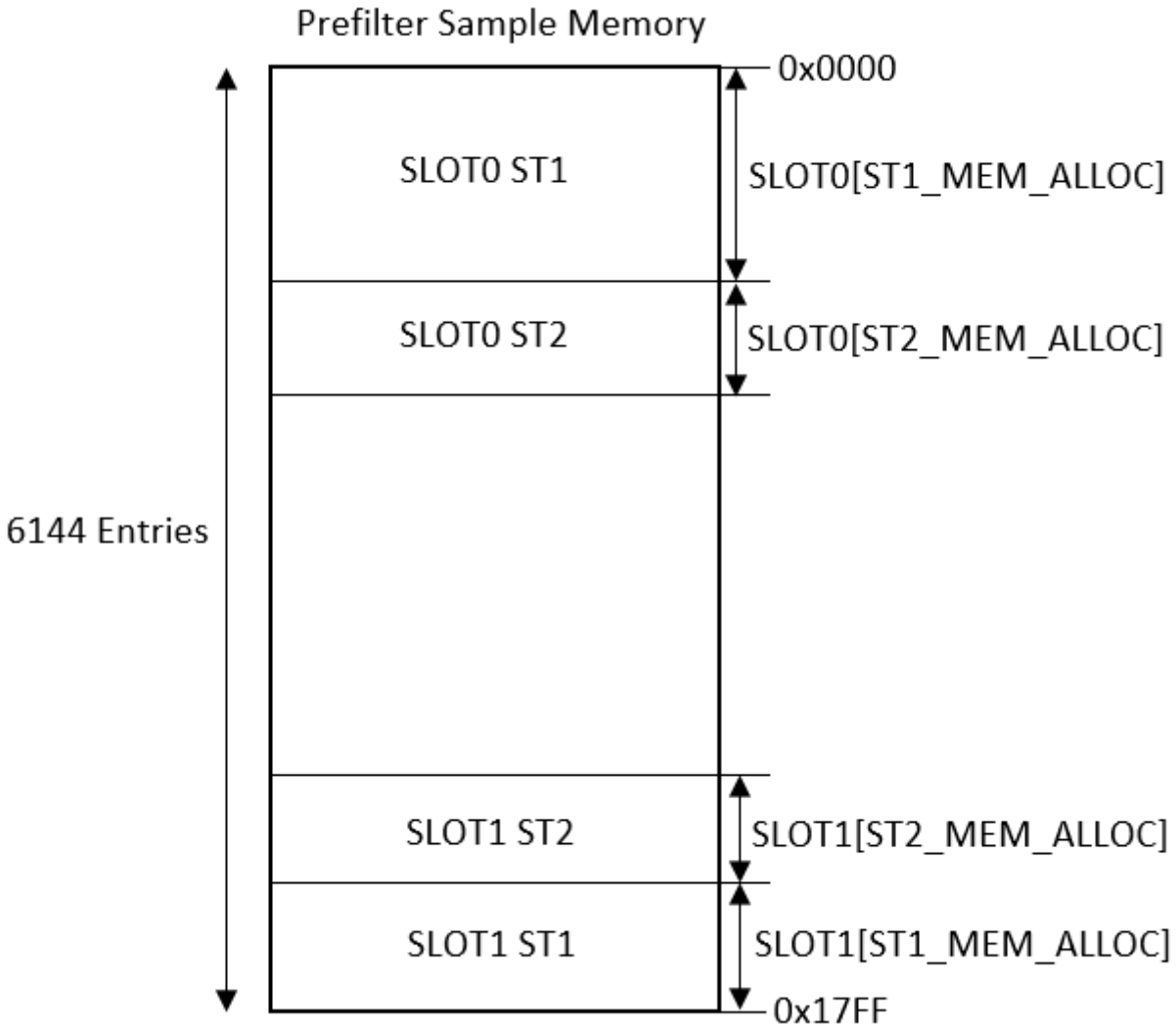


Figure 14-30. ASRC Prefilter Memory Partition

Three factors must be simultaneously optimized when determining how many taps can be ran on the prefilter:

1. The amount of memory available in the prefilter sample memory.
2. The performance limitations of the prefilter (coefficient fetch and MAC rate).
3. The amount of coefficient memory available in ASRC_PRE_COEFF_FIFO.

The Prefilter Sample Memory size places a direct restriction on the number of taps and the max expansion factor for the ST1 and ST2 prefilter. The prefilter coefficient memory places a restriction on the number of taps shared between the ST1 and ST2 prefilter. The cumulative number of taps between ST1 and ST2 must remain below 384 because the prefilter coefficient FIFO is sized to store a maximum of 192 coefficients per context. In addition, the ASRC has a performance requirement which places a limit on the number of filter taps shared between ST1 and ST2 prefilter stages. This performance requirement

exists because the ASRC prefilter can compute a maximum of 786.5*10^6 floating-point multiplications per second. If the prefilter is being utilized by a single context, then the maximum number of filter taps which can run per context processor is equal to:

MAX_TAPS = (786.5*10^6/Fs)/NUM_CH

For example, if the Fs = 384kHz, then the MAX_TAPS will be equal to 2048.17/ NUM_CH. This number of taps represents the total number of multiplies the ASRC can perform for a given number of channels. If the processor from this example is using all 8 channels, then the maximum number of taps per channel will be equal to 2048.17/8 = 256. This performance limitation requires that the ASRC remain under an 8-channel 256-tap prefilter at Fs=384kHz. If the processor from this example if only using 4 channels, then the maximum number of taps will be 512 (Note, this is beyond the coefficient memory boundary of 384). The maximum number of taps can be distributed in any organization across ST1 and ST2 if the cumulative number of taps between ST1 and ST2 remains below the maximum of 384.

All contexts within the design share a single prefilter coefficient memory which is capable of delivering 400*10^6 coefficients per second. The bendiwdh consumed by all prefilters within the ASRC must not exceed this limit. For example, if there are 4 contexts enabled with an Fsin = 384 kHz, then the maximum number of prefilter taps per context is limited to:

MAX_TAPS(384kHz,4-context) = 400*10^6/(4 * 384 kHz) = 260 taps

The number of prefilter taps available to each context is dependent on the sample rate of input data and the number of contexts running within the design. To compute the total bandwidth used by the coefficient memory, the following equation can be used: (Note, fs_in[i] is the input sample rate of context i, and num_pf_taps[i] is the number of prefilter taps used for context i)

int bandwidth = 0; for (int i=0; i<4; i++) { bandwidth = bandwidth + fs_in[i] * num_pf_taps[i]; }

Multiple contexts can run on a single prefilter, and each context can have a different Fs. The maximum number of taps for each context can be calculated by considering each individually. For example, if both CTX0 and CTX1 are sharing Prefilter0, CTX0 Fs=384kHz, CTX1 Fs=48kHz, CTX0 is using 5 channels, CTX1 is using 3 channels, and CTX0 needs to run a 256 tap prefilter. In this case, the number of taps which can run on CTX1 will be calculated as follows:

N = 786.5*10^6 - (384kHz*5*256) = 294.98*10^6

(N/(48kHz))/3 = 2048-taps

From this example, CTX0 can run a 5-channel 256-tap prefilter at 384kHz, while CTX1 can run a 3-channel 2048-tap prefilter at 48kHz on the same processor. Note, however, that this configuration would exceed the available memory in the context processor as well as the coefficient FIFO.

MEM_USED = 5*246+2048*3 =7424 entries

To remain within the limitations of the ASRC prefilter, CTX1 would need to reduce the number of taps to fit within the boundaries of the PF Sample Memory and coefficient FIFO. When determining the capabilities of the prefilter, both the number of taps as well as the performance of the system must be considered.

The prefilter can also be enabled to run in a bypass mode ASRC_CTX_CTRL_EXT1x[PF_BYPASS_MODE]. When bypass mode is enabled, both the prefilter ST1 and ST2 stages will be unused, and input samples will be forwarded directly to the resampling stage. Bypass mode can be used efficiently for resampling options which are performing an up-conversion of the sampling frequency. When a context is placed in prefilter bypass mode, it will not consume any of the PF Sample Memory entries, and will not have a performance impact on the prefilter. If two contexts are sharing a context processor and one of them is in prefilter bypass mode, then the other context can use all the PF Sample Memory entries and max out the performance without impacting the context which is being bypassed. Note, when a context is configured in bypass mode, the ASRC_CTX_CTRL_EXT2x[ST1_NUM_TAPS] and ASRC_CTX_CTRL_EXT1x[PF_EXPANSION_FACTOR] must be set to 0x0.

14.5.2.8 Filter Initialization and Shut Down

When a context is configured to begin processing on the ASRC, the associated prefilter and resampling filters can be initialized to begin in one of 3 possible states. The prefilter and resampling initialization options are selected using the ASRC_CTX_CTRL_EXT1x[RS_INIT_MODE] and ASRC_CTX_CTRL_EXT1x[PF_INIT_MODE]. These options allow the user to select which tap the first sample will be multiplied by and how the ASRC will initialize the remaining filter taps. If ASRC_CTX_CTRL_EXT1x[PF_INIT_MODE] is set to 0x2, then the prefilter will be initialized to run in zero-fill mode. In zero-fill mode, the right half of the prefilter will be filled with zeros, and the first sample written to the ASRC will be multiplied by the center tap of the prefilter. Note, the prefilter initialization mode selected will be applied to both stages ST1 and ST2 of the prefilter. The following diagram represents the zero-fill mode for the case of a 127-tap single stage prefilter. In this diagram S0 represents the sample group written at time 0, S1 represents the sample group written at time 1, or more generally SN is the sample group written at time N.

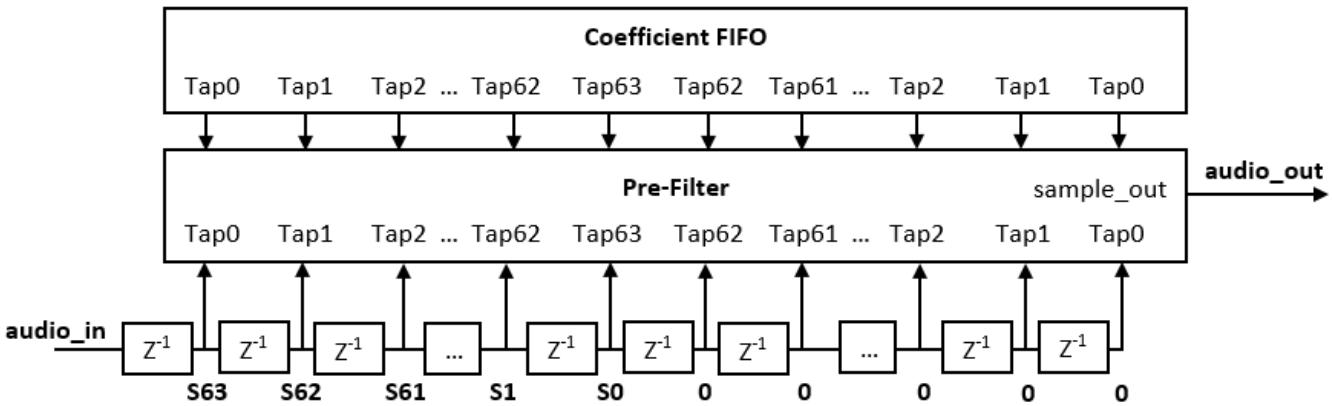


Figure 14-31. Single-Stage Prefilter Zero-Fill Initialization

When operating in zero-fill mode, the first prefilter output sample will be computed after sample data has been written to every tap on the left half of the filter. For the case of a 127 tap prefilter, the first output won't occur until 64 sample groups have been written to the ASRC (S63 to S0). The prefilter output in this mode will be can be represented by the following equation (note pf_tap[n] is the prefilter coefficient at tap n, and audio_in[x] = 0 if x < 0):

$$\text{audio_out}[n] = \sum_{k=0}^{62} \text{audio_in}[n - 63 + k] * \text{pf_tap}[k] + \sum_{k=0}^{63} \text{audio_in}[n + k] * \text{pf_tap}[63 - k]$$

Figure 14-32. Single-Stage Prefilter Zero-Fill Output Equation

If the prefilter is configured to run in a two-stage mode, then the ASRC_CTX_CTRL_EXT1x[PF_INIT_MODE] setting will be applied to both the ST1 and ST2 prefilter. The following image illustrates the zero-fill process for the case of a two-stage prefilter with 127 ST1 taps and 127 ST2 taps. The ST1 prefilter has a programmable expansion factor to place additional delay elements between adjacent filter taps. The expansion factor allows the prefilter to be configured in an IFIR architecture. This will effectively increase the number of samples which must be written to ST1 before an output will be computed. The number of samples needed to calculate the first ST1 output (ST1[0]) is equal to: floor(TAPS/2)*(EXP+1)+1. Note, in this example, S0 corresponds to the first sample group input for this context, S1 corresponds to the sample group input at sample time 1*(EXP+1) (EXP = expansion factor), S2 corresponds to the sample group input at sample time 2*(EXP+1), and more generally SN corresponds to the sample group input at sample time N*(EXP+1).

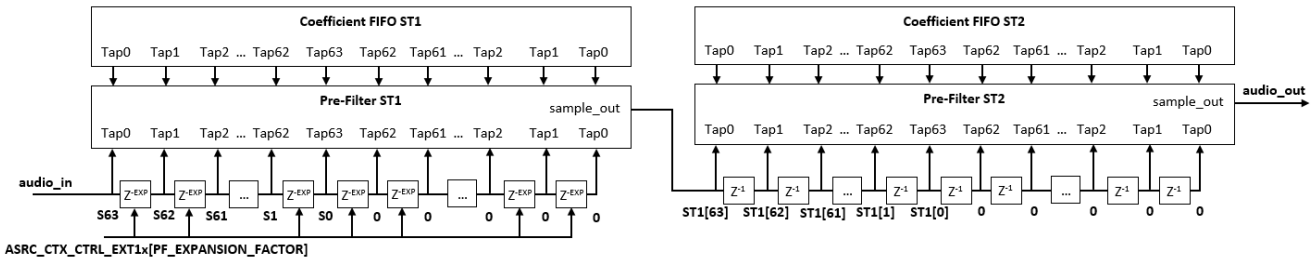


Figure 14-33. Two-Stage Prefilter Zero-Fill Initialization

When ASRC_CTX_CTRL_EXTx[PF_INIT_MODE] is set to 0x1, the prefilter will be initialized in replication mode. In replication mode, the first sample group written to the ASRC will be replicated to fill the right-half of the prefilter taps. In the case of 127 taps, the first output from the prefilter will occur once 64 sample groups have been written to the ASRC. Replication mode is represented in the following diagram.

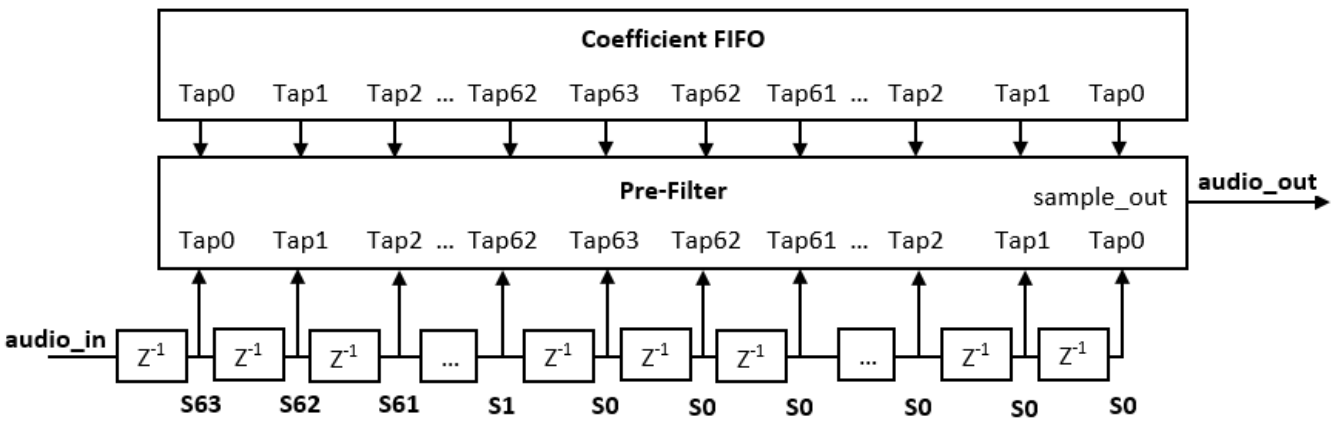


Figure 14-34. Single-Stage Prefilter Replication Initialization

The output data in replication mode can be represented by the same equation as zero-fill mode with the exception that audio_in[x] = S0 if x < 0.

$$\text{audio_out}[n] = \sum_{k=0}^{62} \text{audio_in}[n - 63 + k] * \text{pf_tap}[k] + \sum_{k=0}^{63} \text{audio_in}[n + k] * \text{pf_tap}[63 - k]$$

Figure 14-35. Single-Stage Prefilter Replication Output Equation

If the prefilter is running in a two-stage mode, then the replication initialization mode will be applied to both ST1 and ST2. The ST2 prefilter will replicate the first output from the ST1 prefilter for half of the filter taps enabled. This process is represented in the following image. Note, in this example, S0 corresponds to the first sample group input for this context, S1 corresponds to the sample group input at sample time 1*(EXP+1)

(EXP = expansion factor), S2 corresponds to the sample group input at sample time 2*(EXP+1), and more generally SN corresponds to the sample group input at sample time N*(EXP+1).

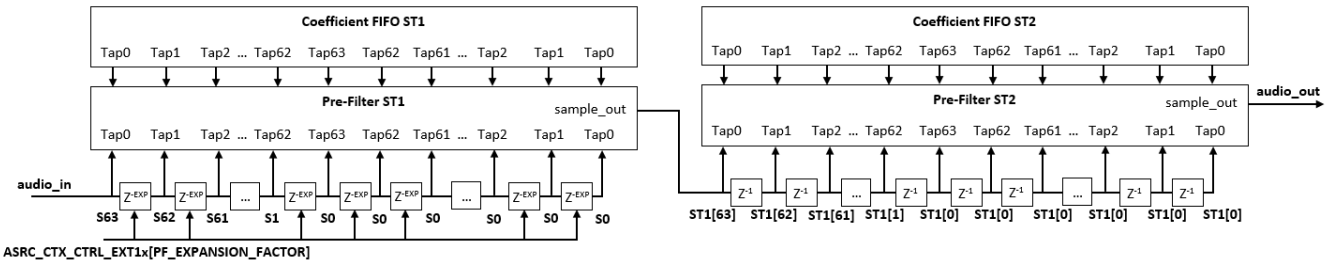


Figure 14-36. Two-Stage Prefilter Replication Initialization

When ASRC_CTX_CTRL_EXTx[PF_INIT_MODE] is set to 0x0, the prefilter initialization will be under software control. There will be no hardware mechanism enabled to fill any of the prefilter taps. The first sample group written to the ASRC will correspond to the rightmost tap of the prefilter, and an output will not be computed until all prefilter taps have been filled with input data. For the case of a 127 tap prefilter, 127 samples must be written to the ASRC before a single prefilter output is computed. This mode is represented in the following diagram.

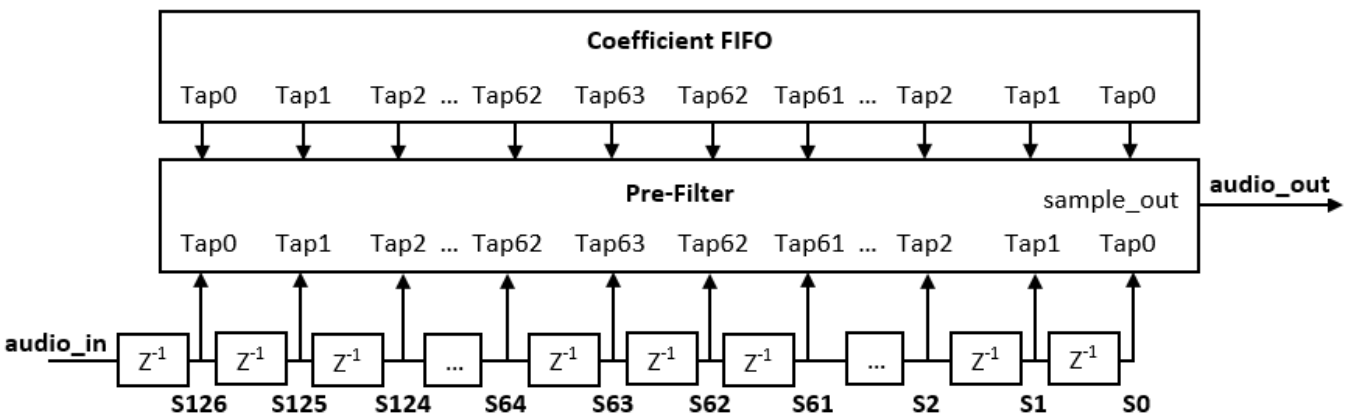


Figure 14-37. Prefilter Software-Initialization Mode

The prefilter output in software control mode is equal to:

$$\text{audio_out}[n] = \sum_{k=0}^{62} \text{audio_in}[n+k] * \text{pf_tap}[k] + \sum_{k=0}^{63} \text{audio_in}[n+k+63] * \text{pf_tap}[63-k]$$

Figure 14-38. Prefilter Software-Initialization Mode

The resampling filter has an equivalent set of initialization modes: software-control, replication, and zero-fill. The resampling initialization mode can be configured using the ASRC_CTX_CTRL_EXTx[RS_INIT_MODE] register field. The only difference to note here is that the initial value written to the resampler is equal to the first output from the prefilter stage.

Each iteration of the prefilter algorithm will fetch 1 new sample from the input audio stream and retire the oldest sample (this is equivalent to a shift by 1 sample operation). If there is no data to fetch then the prefilter will stall until the next sample is received. After the final sample is written to the ASRC, the user must indicate that the audio data is complete to flush the remaining samples from the pipeline. If this stop condition is not indicated to the ASRC, then the remining process will stall under the assumption that there is still more data pending to be received. The bit field ASRC_CTX_CTRLx[RUN_STOP] can be set to indicate that processing for a context will complete, and the pipeline will be flushed. Once the RUN_STOP bit has been triggered, there are two configuration options which can be selected to determine how to flush data from the prefilter and resampling filters. The RS_STOP_MODE and PF_STOP_MODE bit fields in the ASRC_CTX_CTRL_EXTx registers are used to select the RUN_STOP configuration options for the resampler and prefilter respectively. These modes operate similarly to the initialization options. On RUN_STOP, hardware will automatically fill the left-half of the prefilter and resampler with data. This data can either be selected to run in a replication mode or zero-fill mode. In replication mode, the final sample which was written to the ASRC will be replicated for the left-half of the prefilter taps. Also, the final sample from the prefilter will be replicated for the left-half of the resampler taps. In zero fill mode, once all samples have been read from the input WRFIFO, zeros will be written to the prefilter to fill the left-half. Next, zeros will be written to fill the left-half of the resampler once the final calculation has been performed by the prefilter.

14.5.2.9 ASRC IEC60958 Data Insertion

The ASRC contains IEC60958 bit field insertion hardware integrated into the output processing and formatting block. Note, the IEC60958 bit fields are logically ORed with the ASRC output sample data. The user must shift each output sample to the Audio Sample Word boundary defined in the IEC60958 packet specification to ensure data isn't corrupted by this process. This is accomplished using the ASRC_CTX_OUT_CTRLx[SAMPLE_POSITION] bit field. IEC60958 data insertion is only supported when the output is configured to have 16, 20, or 24 bits per sample. To enable IEC60958 data insertion, the ASRC_CTX_OUT_CTRLx[IEC_EN] bit field must

be set on a per-context basis. IEC60958 data is formatted into blocks, frames, and sub-frames. Each sub-frame contains a single sample of audio data along with sample related metadata. The IEC60958 sub-frame is represented in the following diagram.

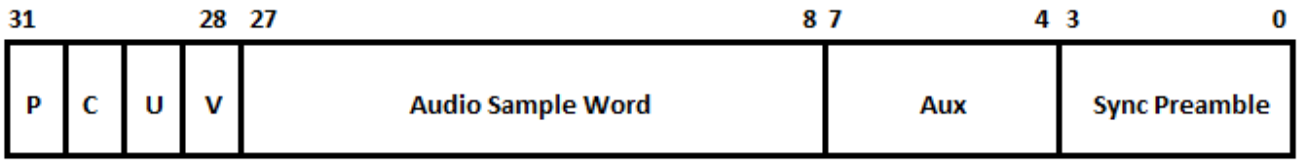


Figure 14-39. IEC60958 Sub-Frame

A sub-frame is composed of the following fields:

1. Parity (P): This is used to give the sub-frame (minus preamble) an even parity
2. Channel Status (C): This contains one bit from a 192-bit channel status packet which is transmitted once per block.
3. User Data (U): The user data field is unsupported by the ASRC and is always set to 0b0.
4. Validity Bit (V): This bit is controlled through the ASRC_CTX_OUT_CTRLx[IEC_V_DATA] register field per context.
5. Audio Sample Word: This field holds the sub-frame sample data (left justified to align the MSB to position 27).
6. Aux: This bit field is unused by the ASRC unless in 24-bits per sample mode. It will contain the lowest 4-bits of the sample data in 24-bit per sample mode.
7. Sync Preamble: This field carries one of the three permitted IEC60958 preambles M, W, or B.

Each frame is comprised of two sub-frames, and each block is comprised of 192 frames (384 sub-frames). If a context contains more than 2 channels, then a single sample group will span across multiple frames. For example, if a context contains 4 channels, then each sample group will span across 2 frames (4 sub-frames). To conform with IEC60958 specifications, the number of channels running on a context with IEC_EN must be an even divisor of 384. This places a limitation to on the number of channels which can run on a context with IEC_EN to: 1, 2, 4, 6, 8, 12, 16, 24, and 32. Each sub-frame contains a preamble which is used for identification and synchronization of blocks, frames, and sub-frames. The preambles are automatically inserted by the ASRC when IEC_EN is set. The B preamble is used for block synchronization, and is only inserted on the first sub-frame of every block. The W preamble is inserted on the second sub-frame of every frame, and the M preamble is inserted on the first sub-frame of every frame, unless it is the first in a block. The preamble insertion pattern is illustrated in the following diagram:

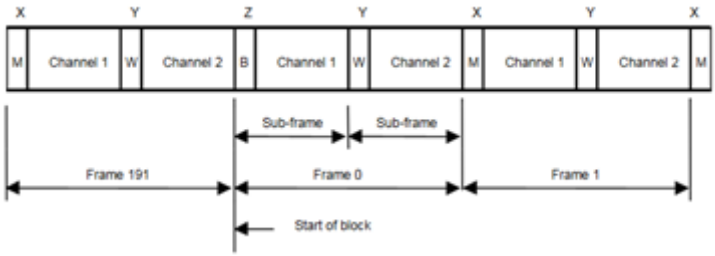


Figure 14-40. ASRC Preamble Insertion

The channel status information is contained in a 192-bit data packet which is transmitted with 1 unique bit per frame (W sub-frames will contain the same channel status as the M sub-frame in the same frame). Channel status data can be configured through the ASRC_CHANNEL_STATUS_xx registers. Each context contains enough storage for a full 192-bit channel status contained within the ASRC. The IEC60958 data insertion logic will cycle through this 192-bit packet starting from the LSB of ASRC_CHANNEL_STATUS0x[CHN_STAT] to the MSB of ASRC_CHANNEL_STATUS_5x[CHN_STAT]. Once the complete packet has been transmitted, the ASRC will start transmitting again repeating from the LSB of ASRC_CHANNEL_STATUS0x[CHN_STAT].

14.5.3 Memory Map and Register Definition

The ASRC Digital memory map and detailed descriptions of all its registers are as follows.

14.5.3.1 D_IP_ASRC_SYN register descriptions

14.5.3.1.1 ASRC_ADDR memory map

d_ip_asrc_syn base address: 30C9_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0 - C	ASRC Input Write FIFO (ASRC_WRFIFO0 - ASRC_WRFIFO3)	32	WO	0000_0000
10 - 1C	ASRC Output Read FIFO (ASRC_RDFIFO0 - ASRC_RDFIFO3)	32	RO	0000_0000
20 - 2C	ASRC Context Control (ASRC_CTX_CTRL0 - ASRC_CTX_CTRL3)	32	RW	0000_0000
30 - 3C	ASRC Context Control Extended 1 (ASRC_CTX_CTRL_EXT10 - ASRC_CTX_CTRL_EXT13)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
40 - 4C	ASRC Context Control Extended 2 (ASRC_CTX_CTRL_EXT20 - ASRC_CTX_CTRL_EXT23)	32	RW	0000_0000
50 - 5C	ASRC Control Input Access (ASRC_CTRL_IN_ACCESS0 - ASRC_CTRL_IN_ACCESS3)	32	RW	0000_0000
60 - 6C	ASRC Datapath Processor Control Slot0 Register0 (ASRC_PROC_CTRL_SLOT0_R00 - ASRC_PROC_CTRL_SLOT0_R03)	32	RW	0000_0000
70 - 7C	ASRC Datapath Processor Control Slot0 Register1 (ASRC_PROC_CTRL_SLOT0_R10 - ASRC_PROC_CTRL_SLOT0_R13)	32	RW	0000_0000
80 - 8C	ASRC Datapath Processor Control Slot0 Register2 (ASRC_PROC_CTRL_SLOT0_R20 - ASRC_PROC_CTRL_SLOT0_R23)	32	RW	0000_0000
90 - 9C	ASRC Datapath Processor Control Slot0 Register3 (ASRC_PROC_CTRL_SLOT0_R30 - ASRC_PROC_CTRL_SLOT0_R33)	32	RW	0000_0000
A0 - AC	ASRC Datapath Processor Control Slot1 Register0 (ASRC_PROC_CTRL_SLOT1_R00 - ASRC_PROC_CTRL_SLOT1_R03)	32	RW	0000_0000
B0 - BC	ASRC Datapath Processor Control SLOT1 Register1 (ASRC_PROC_CTRL_SLOT1_R10 - ASRC_PROC_CTRL_SLOT1_R13)	32	RW	0000_0000
C0 - CC	ASRC Datapath Processor Control SLOT1 Register2 (ASRC_PROC_CTRL_SLOT1_R20 - ASRC_PROC_CTRL_SLOT1_R23)	32	RW	0000_0000
D0 - DC	ASRC Datapath Processor Control SLOT1 Register3 (ASRC_PROC_CTRL_SLOT1_R30 - ASRC_PROC_CTRL_SLOT1_R33)	32	RW	0000_0000
E0 - EC	ASRC Context Output Control (ASRC_CTX_OUT_CTRL0 - ASRC_CTX_OUT_CTRL3)	32	RW	0000_0000
F0 - FC	ASRC Control Output Access (ASRC_CTRL_OUT_ACCESS0 - ASRC_CTRL_OUT_ACCESS3)	32	RW	0000_0000
100 - 10C	ASRC Sample FIFO Status (ASRC_SAMPLE_FIFO_STATUS0 - ASRC_SAMPLE_FIFO_STATUS3)	32	RO	0000_0000
110	ASRC Resampling Ratio Low (ASRC_RS_RATIO_LOW0)	32	RW	0000_0000
114	ASRC Resampling Ratio High (ASRC_RS_RATIO_HIGH0)	32	RW	0000_0000
118	ASRC Resampling Ratio Low (ASRC_RS_RATIO_LOW1)	32	RW	0000_0000
11C	ASRC Resampling Ratio High (ASRC_RS_RATIO_HIGH1)	32	RW	0000_0000
120	ASRC Resampling Ratio Low (ASRC_RS_RATIO_LOW2)	32	RW	0000_0000
124	ASRC Resampling Ratio High (ASRC_RS_RATIO_HIGH2)	32	RW	0000_0000
128	ASRC Resampling Ratio Low (ASRC_RS_RATIO_LOW3)	32	RW	0000_0000
12C	ASRC Resampling Ratio High (ASRC_RS_RATIO_HIGH3)	32	RW	0000_0000
130 - 13C	ASRC Resampling Ratio Update Control (ASRC_RS_UPDATE_CTRL0 - ASRC_RS_UPDATE_CTRL3)	32	RW	0000_0000
140 - 14C	ASRC Resampling Ratio Update Rate (ASRC_RS_UPDATE_RATE0 - ASRC_RS_UPDATE_RATE3)	32	RW	7FFF_FFFF

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
150	ASRC Resampling Center Tap Coefficient Low (ASRC_RS_CT_LOW)	32	RW	0000_0000
154	ASRC Resampling Center Tap Coefficient High (ASRC_RS_CT_HIGH)	32	RW	0000_0000
160 - 16C	ASRC Prefilter Coefficient FIFO (ASRC_PRE_COEFF_FIFO0 - ASRC_PRE_COEFF_FIFO3)	32	RW	0000_0000
170	ASRC Context Resampling Coefficient Memory (ASRC_CTX_RS_COEFF_MEM)	32	WO	0000_0000
174	ASRC Context Resampling Coefficient Control (ASRC_CTX_RS_COEFF_CTRL)	32	RW	0000_0000
178	ASRC Interrupt Control (ASRC_IRQ_CTRL)	32	RW	0000_0FFF
17C	ASRC Interrupt Status Flags (ASRC_IRQ_FLAGS)	32	W1C	0000_0000
180 - 18C	ASRC Channel Status 0 (ASRC_CHANNEL_STATUS_00 - ASRC_CHANNEL_STATUS_03)	32	RW	0000_0000
190 - 19C	ASRC Channel Status 1 (ASRC_CHANNEL_STATUS_10 - ASRC_CHANNEL_STATUS_13)	32	RW	0000_0000
1A0 - 1AC	ASRC Channel Status 2 (ASRC_CHANNEL_STATUS_20 - ASRC_CHANNEL_STATUS_23)	32	RW	0000_0000
1B0 - 1BC	ASRC Channel Status 3 (ASRC_CHANNEL_STATUS_30 - ASRC_CHANNEL_STATUS_33)	32	RW	0000_0000
1C0 - 1CC	ASRC Channel Status 4 (ASRC_CHANNEL_STATUS_40 - ASRC_CHANNEL_STATUS_43)	32	RW	0000_0000
1D0 - 1DC	ASRC Channel Status 5 (ASRC_CHANNEL_STATUS_50 - ASRC_CHANNEL_STATUS_53)	32	RW	0000_0000

14.5.3.1.2 ASRC Input Write FIFO (ASRC_WRFIFO0 - ASRC_WRFIFO3)

14.5.3.1.2.1 Offset

Register	Offset
ASRC_WRFIFO0	0h
ASRC_WRFIFO1	4h
ASRC_WRFIFO2	8h
ASRC_WRFIFO3	Ch

14.5.3.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	CTX_WR_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	CTX_WR_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.2.3 Fields

Field	Description
31-0 CTX_WR_DATA	Write Data For CTX Input FIFO Audio samples for CTX[N] should be written here to be processed by the ASRC.

14.5.3.1.3 ASRC Output Read FIFO (ASRC_RDFIFO0 - ASRC_RDFIFO3)

14.5.3.1.3.1 Offset

Register	Offset
ASRC_RDFIFO0	10h
ASRC_RDFIFO1	14h
ASRC_RDFIFO2	18h
ASRC_RDFIFO3	1Ch

14.5.3.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CTX_RD_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CTX_RD_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.3.3 Fields

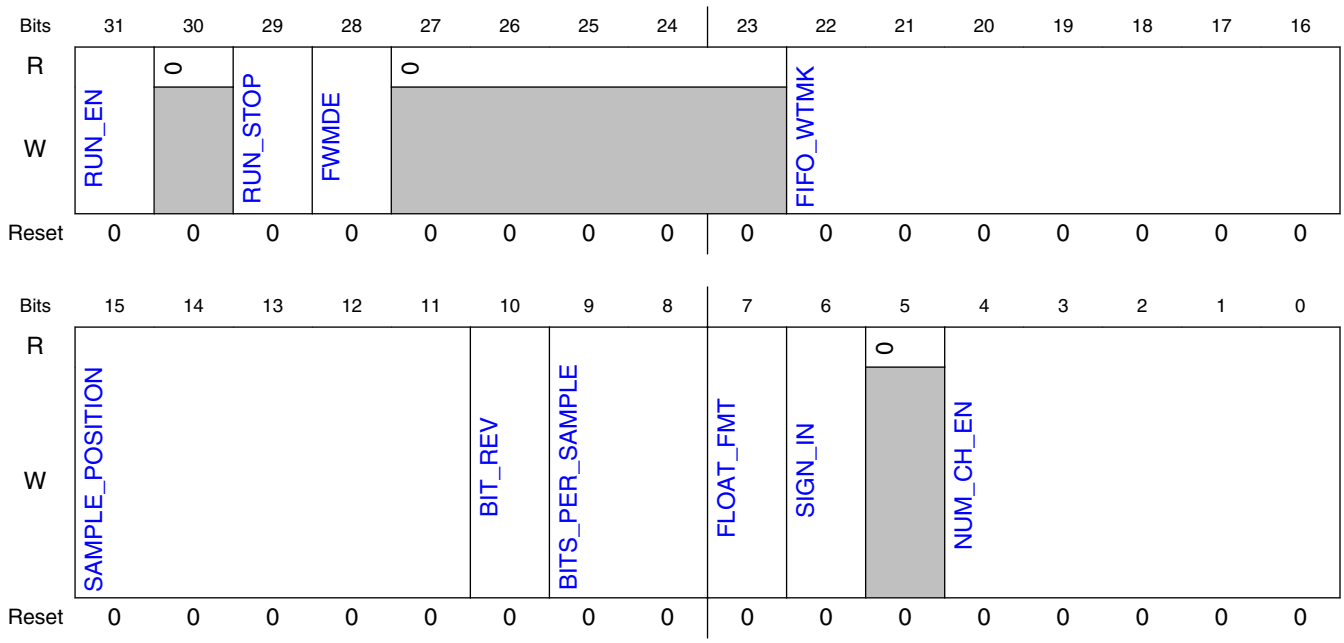
Field	Description
31-0 CTX_RD_DATA	Read Data For CTX Output FIFO Audio samples for CTX[N] will be stored in this FIFO until read by the user. This FIFO contains the final processed audio data for each context.

14.5.3.1.4 ASRC Context Control (ASRC_CTX_CTRL0 - ASRC_CTX_CTRL3)

14.5.3.1.4.1 Offset

Register	Offset
ASRC_CTX_CTRL0	20h
ASRC_CTX_CTRL1	24h
ASRC_CTX_CTRL2	28h
ASRC_CTX_CTRL3	2Ch

14.5.3.1.4.2 Diagram



14.5.3.1.4.3 Fields

Field	Description
31 RUN_EN	Context Run Enable The RUN_EN bit informs the datapath to begin processing sample data for the context. Note, setting RUN_EN to 1'b0 acts as a soft-reset for the context. Data processing will immediately halt upon de-assertion. All control fields associated with a context must be stable prior to setting RUN_EN.
30 —	Reserved
29 RUN_STOP	Context Run Stop The RUN_STOP bit is used to flush the ASRC pipeline and completely end processing for a context. When this bit is asserted, the ASRC will complete processing of all outstanding sample groups in the input FIFO. Note, partial sample groups will be ignored. If samples are written to the FIFO while this bit is set they will be ignored. The last sample (per channel) will be automatically written to the prefilter for half the number of taps of each filter respectively. Next, the final output from the prefilter will be written to the resampler for half the number of resampling taps enabled. Once all outstanding sample conversions have been written to the output FIFO, the associated ASRC_IRQ_FLAGS[RUN_STOP_DONE] flag will be asserted. Note, the OUTFIFO may still contain samples to read when the ASRC_IRQ_FLAGS[RUN_STOP_DONE] bit is set. The RUN_EN bit must be de-asserted for a full soft-reset of the context.
28 FWMDE	FIFO Watermark DMA Enable This bit is used to enable DMA requests for more sample data in the input FIFO. 0 - Input DMA Requests Not Enabled for This Context 1 - Input DMA Requests Enabled for This Context
27-23	Reserved

Table continues on the next page...

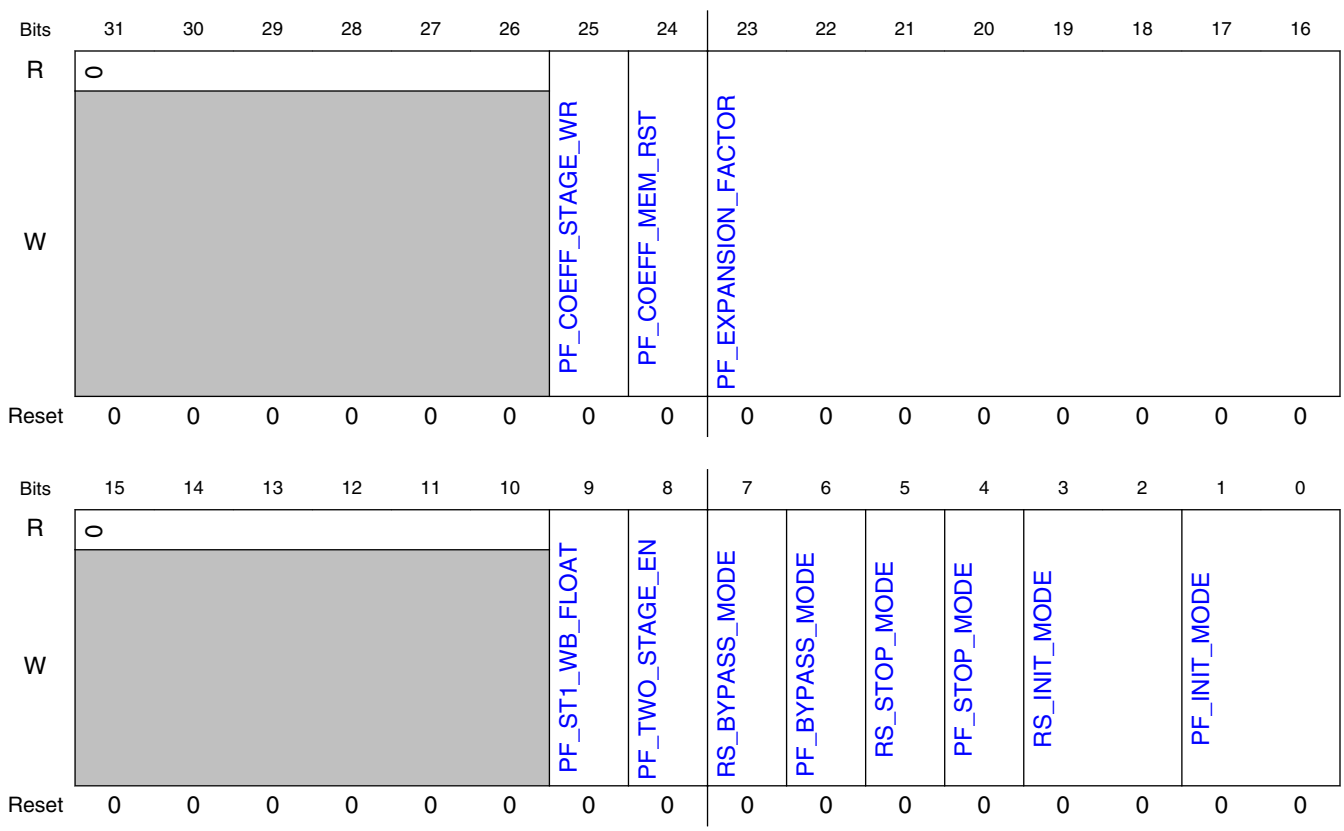
Field	Description
—	
22-16 FIFO_WTMK	Context Input FIFO Watermark The FIFO watermark will be used when generating DMA requests. If the number of sample groups in the input FIFO is < the FIFO_WTMK and the DMA request is enabled with FWMDE then a DMA request will be made for more sample data. Note, the maximum number of sample groups per context is limited to 128.
15-11 SAMPLE_POSITION	Sample Position This register field is used to specify the position of the first valid sample bit within a 32 bit boundary. The position is indexed starting from 0. For example, if the input sample is 16-bits wide and left-justified on a 32-bit boundary, then this register should be set to 16. If the input sample is 16-bits wide and right-justified on a 32-bit boundary, then this register should be set to 0. Bit reversal of the sample data should not be considered when programming this field.
10 BIT_REV	Sample Bit Reversal This register field is used to reverse the bit ordering of input sample data. Note, the bit-reversal operation is computed after shifting (SAMPLE_POSITION) is applied. 0 - Keep Input Ordering 1 - Reverse Bit Ordering
9-8 BITS_PER_SAMPLE	Number of Bits Per Audio Sample This register field is used to specify the number of bits per sample within this context. Note, this width doesn't account for metadata associated with a sample. 00 - 16-bits Per Sample 01 - 20-bits Per Sample 10 - 24-bits Per Sample 11 - 32-bits Per Sample
7 FLOAT_FMT	Context Input Floating Point Format This register field is used to select the input sample data format. The ASRC can process signed integer samples or 32-bit floating point samples. Note, if FLOAT_FMT is set to 0b1 then BITS_PER_SAMPLE must be set to 0b11, and SAMPLE_POSITION must be set to 0x00. 0 - Integer Format 1 - Single Precision Floating Point Format
6 SIGN_IN	Input Data Sign This bit is used to select if the input data is in a signed or unsigned format. The ASRC will provide an internal offset to convert unsigned to signed data when SIGN_IN is set. Note, to convert back to unsigned format at the output of the ASRC, the ASRC_CTX_OUT_CTRLx[SIGN_OUT] bit must also be set for the associated context. This field is only active in 32, 24, 20, or 16-bit integer modes. 0 - Signed Format 1 - Unsigned Format
5 —	Reserved
4-0 NUM_CH_EN	Number of Channels In Context This register field is used to specify the number of channels associated with a context. NUM_CH_EN is zero indexed, so the number of channels associated with the context is (NUM_CH_EN + 1). There are 32 total channels in the design. The sum of all NUM_CH_EN registers in the design must not exceed the total number of channels supported in the design.

14.5.3.1.5 ASRC Context Control Extended 1 (ASRC_CTX_CTRL_EXT10 - ASRC_CTX_CTRL_EXT13)

14.5.3.1.5.1 Offset

Register	Offset
ASRC_CTX_CTRL_EXT10	30h
ASRC_CTX_CTRL_EXT11	34h
ASRC_CTX_CTRL_EXT12	38h
ASRC_CTX_CTRL_EXT13	3Ch

14.5.3.1.5.2 Diagram



14.5.3.1.5.3 Fields

Field	Description
31-26 —	Reserved
25 PF_COEFF_STAGE_WR	Prefilter Coefficient Write Select Select which stage to write prefilter coefficients to. 1: Write to S2 coefficient memory. 0: Write to S1 coefficient memory.
24 PF_COEFF_MEM_RST	Prefilter Coefficient Memory Reset If PF_COEFF_MEM_RST is set to 0b1, the coefficient write address will be reset to 0. This should be done before any write operation to the prefilter coefficient memory begins.
23-16 PF_EXPANSION_FACTOR	Prefilter IFIR Expansion Factor When the prefilter is running in a two stage mode, the first stage (ST1) can be configured with an optional expansion factor. This expansion factor selects the number of delay stages placed between adjacent taps of the prefilter. For example, if PF_EXPANSION_FACTOR is set to 0x3 then 3 delay stages will be placed between each filter tap. This means that every 4th sample will be multiplied by a coefficient. If the PF_EXPANSION_FACTOR is set to 0x5, then 5 delay stages will be placed between each filter tap. This means that every 6th sample will be multiplied by a coefficient.
15-10 —	Reserved
9 PF_ST1_WB_FLOAT	Prefilter Stage1 Writeback Floating Point This register bit is used to select the format for prefilter stage 1 results in two stage mode. The prefilter results ftom stage1 are written back to memory prior to being used by stage2. By default, results are written back to memory in 32-bit integer format. If the PF_ST1_WB_FLOAT bit is set, results from stage1 will be written to memroy in a 32-bit floating point format. 0 - The pre-filter stage1 results are stored in 32-bit integer format. 1 - The pre-filter stage1 results are stored in 32-bit floating point format.
8 PF_TWO_STAGE_EN	Prefilter Two-Stage Enable This register bit is used to select the prefilter configuration. The prefilter can run in a single or double stage mode. When operating in a double-stage mode, the prefilter can be configured to perform an IFIR operation. In a single-stage mode, the prefilter can be configured to perform a FIR operation. 0 - The pre-filter will run in single stage mode (ST1 only) 1 - The pre-filter will run in two stage mode (ST1 and ST2)
7 RS_BYPASS_MODE	Resampler Bypass Mode This register field is used to place the resampler in bypass mode. Bypass mode will force all resampling coefficients to zero except for the center tap. When operating in bypass mode, the resampling ratio for the associated context must be set to Fsin/Fsout = 1.0. The center tap coefficient must be set to a non-zero value. Note, the RS_STOP_MODE and RS_INIT_MODE mode fields will still be active. For a true 1-to-1 input vs output, set RS_INIT_MODE to a non-zero value. When a context is placed in resampler bypass mode, the prefilter is still active. 0 - Run the resampler in normal operation. 1 - Run the resampler in bypass mode.
6 PF_BYPASS_MODE	Prefilter Bypass Mode

Table continues on the next page...

Field	Description
	This register field is used to place the prefilter in bypass mode. When operating in bypass mode, samples for this context will be passed directly to the resampler. Both the ST1 and ST2 prefilter stages will be unused for this context. The prefilter sample memory will be unused by this context. The PF_EXPANSION_FACTOR must be set to 0x0 when operating in bypass mode. The ST1_NUM_TAPS must be set to 0x0 when operating in bypass mode. 0 - Run the prefilter in normal operation. 1 - Run the prefilter in bypass mode.
5 RS_STOP_MODE	Resampler Stop Mode This register field selects the shut-down procedure for the resampling sample buffer. When the ASRC_CTX_CTRL[RUN_STOP] bit is set, ASRC will perform a shut-down option to fill the last (NUM_RES_TAPS/2) taps of the resampling filter with the last output from the prefilter (on a per channel basis) or zeros. 0 - Replicate the final prefilter output for the left-half of the resampler on RUN_STOP. 1 - Zero-Fill the left-half of the resampler on RUN_STOP.
4 PF_STOP_MODE	Pre-Filter Stop Mode This register field selects the shut-down procedure for the Pre-Filter sample buffer. When the ASRC_CTX_CTRL[RUN_STOP] bit is set, ASRC will perform a shut-down option to fill the last (NUM_LPF_TAPS/2) taps of the pre-filter with the last sample input to the ASRC_WRFIFO (on a per channel basis) or zeros. 0 - Replicate the last sample input to the ASRC_WRFIFO for the left-half of the pre-filter on RUN_STOP. 1 - Zero-Fill the left-half of the pre-filter on RUN_STOP.
3-2 RS_INIT_MODE	Resampler Initialization Mode This register field selects the initialization procedure for the resampling sample buffer. The ASRC contains an initialization option to fill the first (NUM_RES_TAPS/2) taps of the resampling filter with the first sample output from the prefilter (on a per channel basis) or zeros. Otherwise the ASRC will require NUM_RES_TAPS samples to be written to the resampler prior to running a resampling calculation. 00 - Do not pre-fill any resampler taps. The first sample output from the prefilter corresponds to the highest index resampling filter tap. 01 - Replicate the first prefilter output sample to fill the right half of the resampler. 10 - Fill the right half of the re-sampler with zeros. 11 - N/A
1-0 PF_INIT_MODE	Prefilter Initialization Mode This register field selects the initialization procedure for the prefilter sample buffer. The ASRC contains an initialization option to fill the first (NUM_LPF_TAPS/2) taps of the prefilter with zeros or the first sample written to the input FIFO (on a per channel basis). Otherwise the ASRC will require NUM_LPF_TAPS samples to be written to the input FIFO prior to running a filter calculation. 00 - Do not pre-fill any prefilter taps. The first sample written to the ASRC corresponds to the highest index prefilter filter tap. 01 - Replicate the first sample to fill the right half of the prefilter. 10 - Zero fill the right half of the prefilter. 11 - N/A

14.5.3.1.6 ASRC Context Control Extended 2 (ASRC_CTX_CTRL_EXT20 - ASRC_CTX_CTRL_EXT23)

14.5.3.1.6.1 Offset

Register	Offset
ASRC_CTX_CTRL_EXT 20	40h
ASRC_CTX_CTRL_EXT 21	44h
ASRC_CTX_CTRL_EXT 22	48h
ASRC_CTX_CTRL_EXT 23	4Ch

14.5.3.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								ST2_NUM_TAPS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ST1_NUM_TAPS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.6.3 Fields

Field	Description
31-25 —	Reserved
24-16 ST2_NUM_TAPS	Prefilter Stage2 Number of Taps Number of taps used by the prefilter stage 2. This register is zero-indexed. If ST2_NUM_TAPS is set to 0x0 then the number of taps S2_TAPS will be equal to 1.
15-9 —	Reserved
8-0 ST1_NUM_TAPS	Prefilter Stage1 Number of Taps Number of taps used by the prefilter stage 1. This register is zero-indexed. If ST2_NUM_TAPS is set to 0x0 then the number of taps S2_TAPS will be equal to 1.

14.5.3.1.7 ASRC Control Input Access (ASRC_CTRL_IN_ACCESS0 - ASRC_CTRL_IN_ACCESS3)

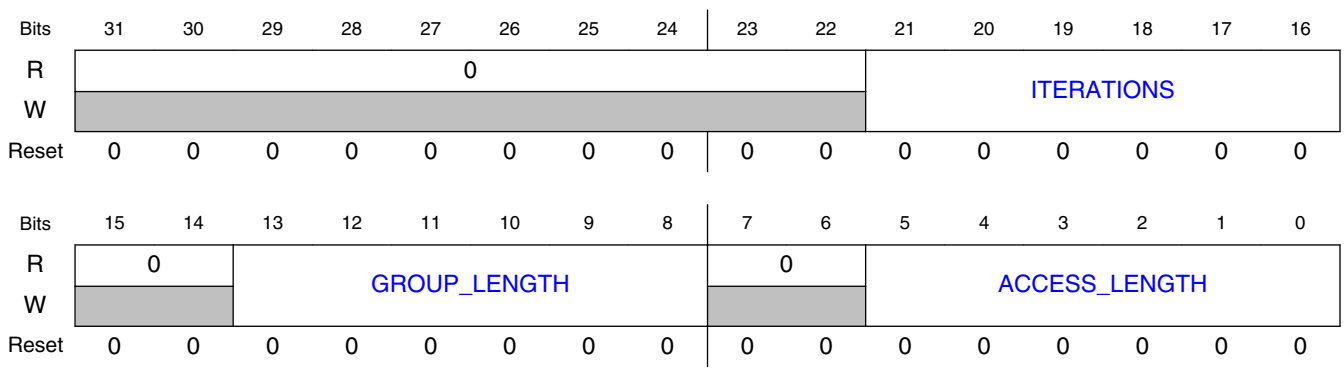
This register is used to describe how input samples are interleaved when being written to the ASRC. The ASRC has general de-interleaving hardware built into the input stage. This interleaving format supports fetching data from multiple sources per burst, a programmable number of channels per source, and variable length burst accesses.

A single burst transaction will iterate across (GROUP_LENGTH/ACCESS_LENGTH) number of sources. Each source access will consist of a burst transaction which fetches ITERATIONS*ACCESS_LENGTH samples. If all channels associated with a given context are contained within a single source, then the GROUP_LENGTH will be equal to the ACCESS_LENGTH. The total number of samples input to the ASRC in a burst access will be equal to GROUP_LENGTH*ITERATIONS. Note, it is assumed that channels are interleaved within a single source.

14.5.3.1.7.1 Offset

Register	Offset
ASRC_CTRL_IN_ACCESS0	50h
ASRC_CTRL_IN_ACCESS1	54h
ASRC_CTRL_IN_ACCESS2	58h
ASRC_CTRL_IN_ACCESS3	5Ch

14.5.3.1.7.2 Diagram



14.5.3.1.7.3 Fields

Field	Description
31-22 —	Reserved
21-16 ITERATIONS	Number of Sequential Fetches Per Source This register selects the number of sequential ACCESS_LENGTH reads from a single source prior to iterating to the next source. For example, if one audio source contains two channels of data (ACCESS_LENGTH == 0x2), and every access to this source will read 32 samples, then the ITERATIONS parameter should be set to 16. ITERATIONS is indexed from one. If ITERATIONS is set to 0x0 this will result in an invalid configuration.
15-14 —	Reserved
13-8 GROUP_LENGTH	Number of Channels in a Context This register has units of channels/context. When combined with ACCESS_LENGTH, these parameters will inform the ASRC how many independent sources will be combined to provide data in a burst write. The number of sources per burst is (GROUP_LENGTH/ACCESS_LENGTH). This register is indexed from one. If GROUP_LENGTH == 0x0 this will result in an invalid configuration. For example, if the context has 2 channels associated with it then GROUP_LENGTH should be programmed to 0x2. The maximum valid GROUP_LENGTH value is 0x20. GROUP_LENGTH must be programmed to an integer multiple of ACCESS_LENGTH.
7-6 —	Reserved
5-0 ACCESS_LENGTH	Number Of Channels Per Source This register has units of channels/source. When fetching samples from a single source the ASRC will assume data is formatted with ACCESS_LENGTH interleaved samples. The total number of samples fetched from a single source before iterating to the next source is ACCESS_LENGTH*ITERATIONS. For example, if a single audio source contains two channels, then ACCESS_LENGTH should be set to 0x2. Note 0x0 is not a valid ACCESS_LENGTH configuration.

14.5.3.1.8 ASRC Datapath Processor Control Slot0 Register0 (ASRC_PROC_CTRL_SLOT0_R00 - ASRC_PROC_CTRL_SLOT0_R03)

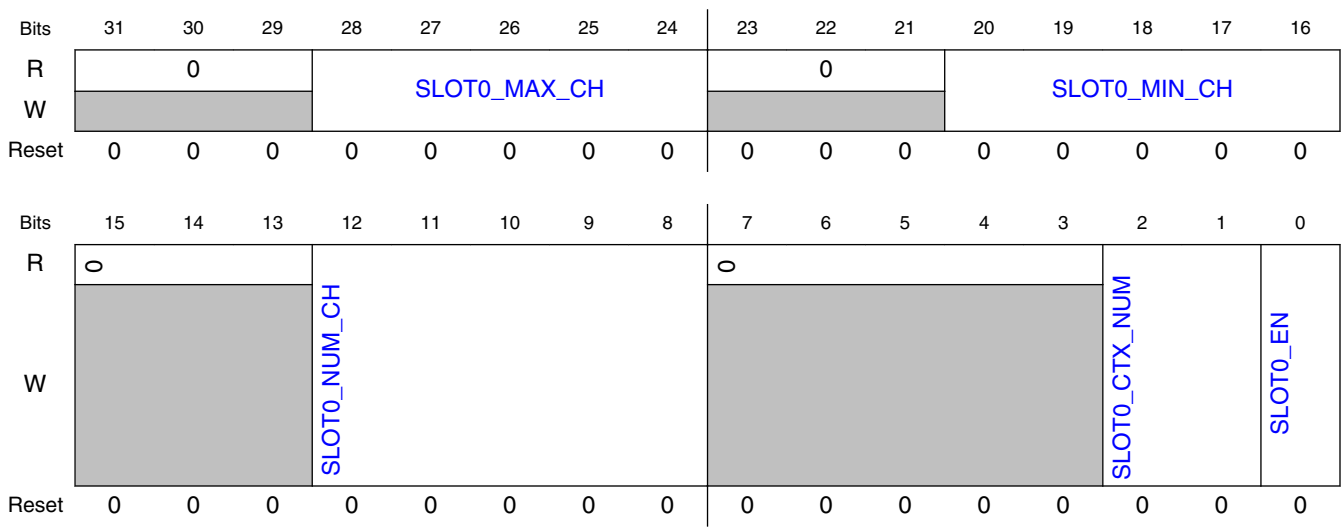
This register contains control fields to configure Slot0 of each datapath processing unit within the design. Each processing unit contains control registers to run up to 2 independent contexts (called slots). There are a total of 8 channels supported per datapath processing unit. If a single context contains more than 8 channels, it must be split between multiple datapath processing units. Note, setting the SLOTx_EN bit will not begin processing for the associated context. To begin processing data, the global ASRC_CTX_CTRLa[RUN_EN] must be set. Prior to setting ASRC_CTX_CTRLa[RUN_EN] for a given context, all of the slots associated with it

must be statically configured in the ASRC_PROCESSOR_CTRLa registers. Each register in the ASRC_PROC_CTRL_SLOT0_R0 register array corresponds to a different context datapath processor.

14.5.3.1.8.1 Offset

Register	Offset
ASRC_PROC_CTRL_SLOT0_R00	60h
ASRC_PROC_CTRL_SLOT0_R01	64h
ASRC_PROC_CTRL_SLOT0_R02	68h
ASRC_PROC_CTRL_SLOT0_R03	6Ch

14.5.3.1.8.2 Diagram



14.5.3.1.8.3 Fields

Field	Description
31-29 —	Reserved
28-24 SLOT0_MAX_CH	SLOT0 Maximum Global Channel Number The global ASRC_CTX_CTRLN[NUM_CH_EN] registers are used to select how many global channels correspond to a single context (0 to 32). The SLOT0_MIN_CH and SLOT0_MAX_CH are used to break down this number of channels into the values which will run on SLOT0. For example, if CONTEXT0 owns

Table continues on the next page...

Field	Description
	16 channels (ASRC_CTX_CTRL0[NUM_CH_EN] = 0xF), and SLOT0 is used to process channels 0 through 5, then the SLOT0_MIN_CH will be set to 0x0 and SLOT0_MAX_CH will be set to 0x5. An alternative implementation could have SLOT0 processing channels 7 through 12. In this case, SLOT0_MIN_CH will be set to 0x7 and SLOT0_MAX_CH will be set to 0xC.
23-21 —	Reserved
20-16 SLOT0_MIN_CH	SLOT0 Minimum Global Channel Number The global ASRC_CTX_CTRLN[NUM_CH_EN] registers are used to select how many global channels correspond to a single context (0 to 32). The SLOT0_MIN_CH and SLOT0_MAX_CH are used to break down this number of channels into the values which will run on SLOT0. For example, if CONTEXT0 owns 16 channels (ASRC_CTX_CTRL0[NUM_CH_EN] = 0xF), and SLOT0 is used to process channels 0 through 5, then the SLOT0_MIN_CH will be set to 0x0 and SLOT0_MAX_CH will be set to 0x5. An alternative implementation could have SLOT0 processing channels 7 through 12. In this case, SLOT0_MIN_CH will be set to 0x7 and SLOT0_MAX_CH will be set to 0xC.
15-13 —	Reserved
12-8 SLOT0_NUM_CH	SLOT0 Number of Channels This register field selects the number of context channels used by SLOT0. Note, a single datapath processing unit supports up to 8 channels. One slot can use anywhere from 0 to 8 channels. The total number of channels used by the combination of SLOT0_NUM_CH and SLOT1_NUM_CH must be less than or equal to 8 for valid operation. If SLOT0_EN is set to 0b0, then SLOT0_NUM_CH will use 0 channels. 00000 - Context SLOT0 owns 1 of 8 channels 00001 - Context SLOT0 owns 2 of 8 channels 00010 - Context SLOT0 owns 3 of 8 channels 00011-00111 - Context SLOT0 owns N of 8 channels
7-3 —	Reserved
2-1 SLOT0_CTX_NUM	Context SLOT0 Selection This register selects the context which will run in SLOT0. Note, one context can be enabled to run in multiple slots across multiple datapath processing units. However, SLOT0 and SLOT1 within a single datapath processor should not be set to the same context.
0 SLOT0_EN	SLOT0 Enable Set this bit to 0b1 to enable SLOT0 for data processing. Note, the context running in this slot (SLOT0_CTX_NUM) will not begin processing data until the global ASRC_CTX_CTRL[RUN_EN] bit is set to 0b1. This register field must be static prior to setting the associated ASRC_CTX_CTRL[RUN_EN]. 0 - Context SLOT0 is disabled 1 - Context SLOT0 is enabled

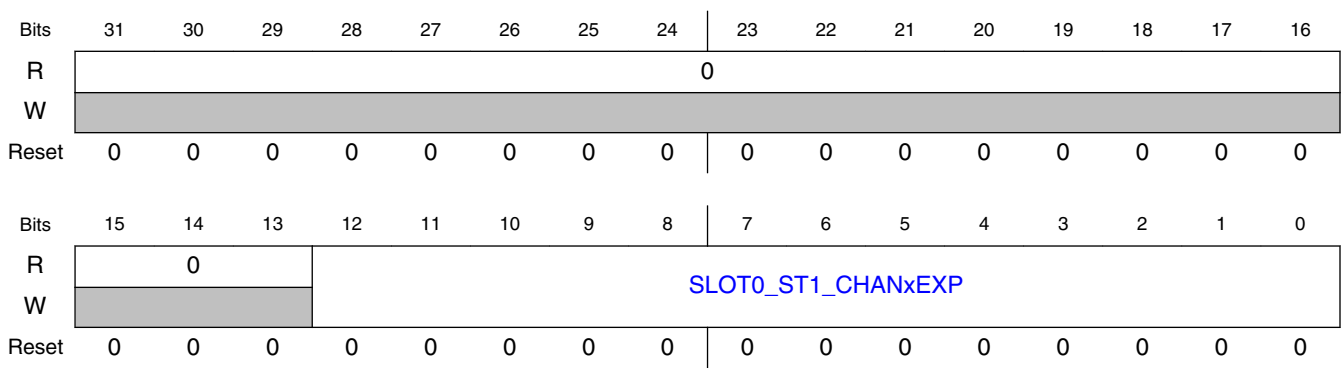
14.5.3.1.9 ASRC Datapath Processor Control Slot0 Register1 (ASRC_PROC_CTRL_SLOT0_R10 - ASRC_PROC_CTRL_SLOT0_R13)

This register contains control fields to configure Slot0 of each datapath processing unit within the design. Each processing unit contains control registers to run up to 2 independent contexts (called slots). There are a total of 8 channels supported per datapath processing unit. If a single context contains more than 8 channels, it must be split between multiple datapath processing units. Note, setting the SLOTx_EN bit will not begin processing for the associated context. To begin processing data, the global ASRC_CTX_CTRLa[RUN_EN] must be set. Prior to setting ASRC_CTX_CTRLa[RUN_EN] for a given context, all of the slots associated with it must be statically configured in the ASRC_PROCESSOR_CTRLa registers. Each register in the ASRC_PROC_CTRL_SLOT0_R1 register array corresponds to a different context datapath processor.

14.5.3.1.9.1 Offset

Register	Offset
ASRC_PROC_CTRL_SLOT0_R10	70h
ASRC_PROC_CTRL_SLOT0_R11	74h
ASRC_PROC_CTRL_SLOT0_R12	78h
ASRC_PROC_CTRL_SLOT0_R13	7Ch

14.5.3.1.9.2 Diagram



14.5.3.1.9.3 Fields

Field	Description
31-13	Reserved

Table continues on the next page...

Field	Description
—	
12-0 SLOT0_ST1_CHANxEXP	SLOT0 Stage1 Channels x Expansion Factor When the ASRC prefilter is configured to run in two stage mode, the first stage can be programmed to have an expansion factor ASRC_CTX_CTRL_EXT1x[PF_EXPANSION_FACTOR]. The expansion factor specifies the number of delay units between each filter tap in the Stage1 filter. For example, if the Stage1 expansion factor is set to 1, then 1 unit of delay will be placed between adjacent filter taps. If the Stage1 expansion factor is set to 2, then 2 units of delay will be placed between adjacent filter taps. When the prefilter is running in a 2 stage mode, this register must be programmed with the following value: `SLOT0_ST1_CHANxEXP = (ASRC_PROC_CTRL_S0_R0x[SLOT0_NUM_CH]+1) * (ASRC_CTX_CTRL_EXT1x[PF_EXPANSION_FACTOR])` When the ASRC prefilter is configured to run in a single stage mode, this register must be set to 0x00.

14.5.3.1.10 ASRC Datapath Processor Control Slot0 Register2 (ASRC_PROC_CTRL_SLOT0_R20 - ASRC_PROC_CTRL_SLOT0_R23)

This register contains control fields to configure Slot0 of each datapath processing unit within the design. Each processing unit contains control registers to run up to 2 independent contexts (called slots). There are a total of 8 channels supported per datapath processing unit. If a single context contains more than 8 channels, it must be split between multiple datapath processing units. Note, setting the SLOTx_EN bit will not begin processing for the associated context. To begin processing data, the global ASRC_CTX_CTRLa[RUN_EN] must be set. Prior to setting ASRC_CTX_CTRLa[RUN_EN] for a given context, all of the slots associated with it must be statically configured in the ASRC_PROCESSOR_CTRLa registers. Each register in the ASRC_PROC_CTRL_SLOT0_R2 register array corresponds to a different context datapath processor.

14.5.3.1.10.1 Offset

Register	Offset
ASRC_PROC_CTRL_SLOT0_R20	80h
ASRC_PROC_CTRL_SLOT0_R21	84h
ASRC_PROC_CTRL_SLOT0_R22	88h
ASRC_PROC_CTRL_SLOT0_R23	8Ch

14.5.3.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			SLOT0_ST1_MEM_ALLOC												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SLOT0_ST1_ST_ADDR												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.10.3 Fields

Field	Description
31-29 —	Reserved

28-16
SLOT0_ST1_MEM_ALLOC
	SLOT0 Stage1 Memory Allocation
	This register will specify the amount of memory to be used by the SLOT0 ST1 prefilter. The amount of memory used here is specific to the ST1 prefilter number of taps (ASRC_CTX_CTRL_EXT2x[ST1_NUM_TAPS]), expansion factor (ASRC_CTX_CTRL_EXT1x[PF_EXPANSION_FACTOR]), and number of channels (ASRC_PROC_CTRL_SLOT0_R0x[SLOT0_NUM_CH]). To compute the amount of memory to use for the ST1 prefilter, the following equation can be used:
	`SLOT0_ST1_MEM_ALLOC = [ ASRC_CTX_CTRL_EXT2x[ST1_NUM_TAPS] * (ASRC_PROC_CTRL_SLOT0_R0x[SLOT0_NUM_CH]+1) * (ASRC_CTX_CTRL_EXT1x[PF_EXPANSION_FACTOR]+1) ] + (ASRC_PROC_CTRL_SLOT0_R0x[SLOT0_NUM_CH]+1)`
	When operating in a single stage prefilter, the amount of memory can be calculated through the following equation:
	`SLOT0_ST1_MEM_ALLOC = (ASRC_PROC_CTRL_SLOT0_R0x[SLOT0_NUM_CH]+1) * (ASRC_CTX_CTRL_EXT2x[ST1_NUM_TAPS]+1)`

15-13 —	Reserved
12-0 SLOT0_ST1_ST_ADDR	SLOT0 Stage1 Start Address This register will specify the prefilter RAM base address for the SLOT0 ST1 prefilter. The value selected here must prevent SLOT0 ST1, SLOT0 ST2, SLOT1 ST1, and SLOT1 ST2 from overlapping in RAM. For example, if SLOT0_ST1_ST_ADDR = 0x0, and SLOT0_ST1_MEM_ALLOC = 0x10, then the address range of 0x0 - 0x0F is consumed by SLOT0 ST1, and no other masters in this context processor can occupy this address range.

14.5.3.1.11 ASRC Datapath Processor Control Slot0 Register3 (ASRC_PROC_CTRL_SLOT0_R30 - ASRC_PROC_CTRL_SLOT0_R33)

This register contains control fields to configure Slot0 of each datapath processing unit within the design. Each processing unit contains control registers to run up to 2 independent contexts (called slots). There are a total of 8 channels supported per datapath processing unit. If a single context contains more than 8 channels, it must be split between multiple datapath processing units. Note, setting the SLOTx_EN bit will not begin processing for the associated context. To begin processing data, the global ASRC_CTX_CTRLa[RUN_EN] must be set. Prior to setting ASRC_CTX_CTRLa[RUN_EN] for a given context, all of the slots associated with it must be statically configured in the ASRC_PROCESSOR_CTRLa registers. Each register in the ASRC_PROC_CTRL_SLOT0_R3 register array corresponds to a different context datapath processor.

14.5.3.1.11.1 Offset

Register	Offset
ASRC_PROC_CTRL_SLOT0_R30	90h
ASRC_PROC_CTRL_SLOT0_R31	94h
ASRC_PROC_CTRL_SLOT0_R32	98h
ASRC_PROC_CTRL_SLOT0_R33	9Ch

14.5.3.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			SLOT0_ST2_MEM_ALLOC												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SLOT0_ST2_ST_ADDR												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.11.3 Fields

Field	Description
31-29 —	Reserved
28-16 SLOT0_ST2_MEM_ALLOC	SLOT0 Stage2 Memory Allocation This register will specify the amount of memory to be used by the SLOT0 ST2 prefilter. The amount of memory used here is specific to the ST2 prefilter number of taps (ASRC_CTX_CTRL_EXT2x[ST2_NUM_TAPS]+1), and number of channels (ASRC_PROC_CTRL_SLOT0_R0x[SLOT0_NUM_CH]+1). To compute the amount of memory to use for the ST2 prefilter, the following equation can be used: `SLOT0_ST2_MEM_ALLOC = (ASRC_PROC_CTRL_SLOT0_R0x[SLOT0_NUM_CH]+1) * (ASRC_CTX_CTRL_EXT2x[ST2_NUM_TAPS]+1)`
15-13 —	Reserved
12-0 SLOT0_ST2_ST_ADDR	SLOT0 Stage2 Start Address This register will specify the prefilter RAM base address for the SLOT0 ST2 prefilter. The value selected here must prevent SLOT0 ST1, SLOT0 ST2, SLOT1 ST1, and SLOT1 ST2 from overlapping in RAM. For example, if SLOT0_ST2_ST_ADDR = 0x0, and SLOT0_ST2_MEM_ALLOC = 0x10, then the address range of 0x0 - 0x0F is consumed by SLOT0 ST2, and no other masters in this context processor can occupy this address range.

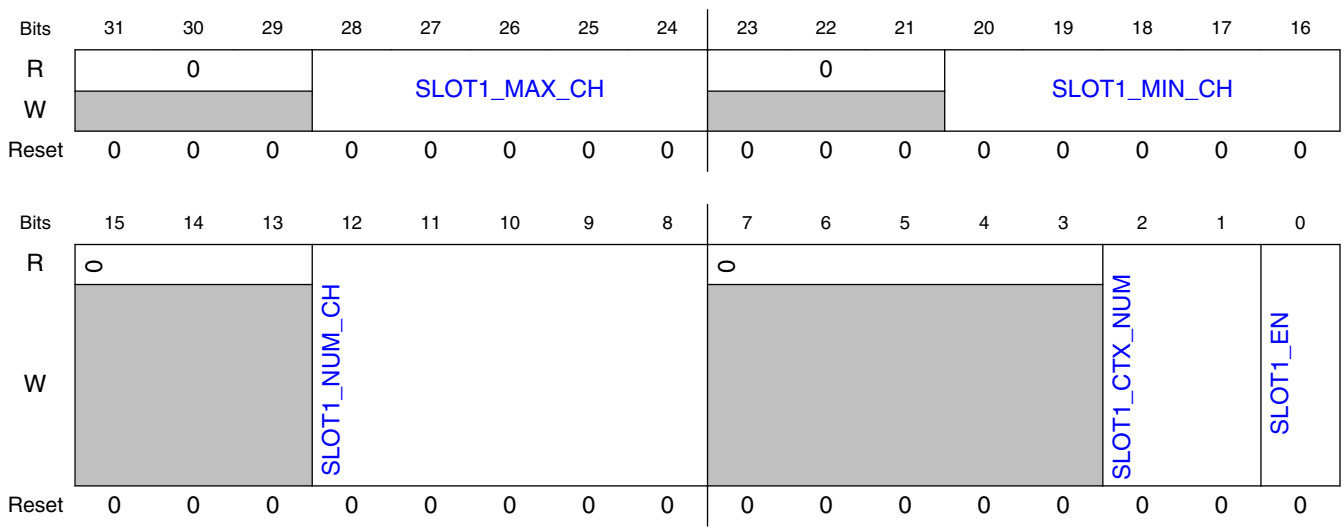
14.5.3.1.12 ASRC Datapath Processor Control Slot1 Register0 (ASRC_PROC_CTRL_SLOT1_R00 - ASRC_PROC_CTRL_SLOT1_R03)

This register contains control fields to configure Slot0 of each datapath processing unit within the design. Each processing unit contains control registers to run up to 2 independent contexts (called slots). There are a total of 8 channels supported per datapath processing unit. If a single context contains more than 8 channels, it must be split between multiple datapath processing units. Note, setting the SLOTx_EN bit will not begin processing for the associated context. To begin processing data, the global ASRC_CTX_CTRLa[RUN_EN] must be set. Prior to setting ASRC_CTX_CTRLa[RUN_EN] for a given context, all of the slots associated with it must be statically configured in the ASRC_PROCESSOR_CTRLa registers. Each register in the ASRC_PROC_CTRL_SLOT1_R0 register array corresponds to a different context datapath processor.

14.5.3.1.12.1 Offset

Register	Offset
ASRC_PROC_CTRL_SLOT1_R00	A0h
ASRC_PROC_CTRL_SLOT1_R01	A4h
ASRC_PROC_CTRL_SLOT1_R02	A8h
ASRC_PROC_CTRL_SLOT1_R03	ACh

14.5.3.1.12.2 Diagram



14.5.3.1.12.3 Fields

Field	Description
31-29 —	Reserved
28-24 SLOT1_MAX_CH	Slot1 Maximum Global Channel Number The global ASRC_CTX_CTRLN[NUM_CH_EN] registers are used to select how many global channels correspond to a single context (0 to 32). The SLOT1_MIN_CH and SLOT1_MAX_CH are used to break down this number of channels into the values which will run on Slot1. For example, if CONTEXT0 owns 16 channels (ASRC_CTX_CTRL0[NUM_CH_EN] = 0xF), and Slot1 is used to process channels 0 through 5, then the SLOT1_MIN_CH will be set to 0x0 and SLOT1_MAX_CH will be set to 0x5. An alternative implementation could have Slot1 processing channels 7 through 12. In this case, SLOT1_MIN_CH will be set to 0x7 and SLOT1_MAX_CH will be set to 0xC.
23-21	Reserved

Table continues on the next page...

Field	Description
—	
20-16 SLOT1_MIN_CH	Slot1 Minimum Global Channel Number The global ASRC_CTX_CTRLN[NUM_CH_EN] registers are used to select how many global channels correspond to a single context (0 to 32). The SLOT1_MIN_CH and SLOT1_MAX_CH are used to break down this number of channels into the values which will run on Slot1. For example, if CONTEXT0 owns 16 channels (ASRC_CTX_CTRL0[NUM_CH_EN] = 0xF), and Slot1 is used to process channels 0 through 5, then the SLOT1_MIN_CH will be set to 0x0 and SLOT1_MAX_CH will be set to 0x5. An alternative implementation could have Slot1 processing channels 7 through 12. In this case, SLOT1_MIN_CH will be set to 0x7 and SLOT1_MAX_CH will be set to 0xC.
15-13 —	Reserved
12-8 SLOT1_NUM_CH	SLOT1 Number of Channels This register field selects the number of context channels used by SLOT1. Note, a single datapath processing unit supports up to 8 channels. One slot can use anywhere from 0 to 8 channels. The total number of channels used by the combination of SLOT1_NUM_CH and SLOT1_NUM_CH must be less than or equal to 8 for valid operation. If SLOT1_EN is set to 0b0, then SLOT1_NUM_CH will use 0 channels. 00000 - Context SLOT1 owns 1 of 8 channels 00001 - Context SLOT1 owns 2 of 8 channels 00010 - Context SLOT1 owns 3 of 8 channels 00011-00111 - Context SLOT1 owns N of 8 channels
7-3 —	Reserved
2-1 SLOT1_CTX_NUM	Context SLOT1 Selection This register selects the context which will run in SLOT1. Note, one context can be enabled to run in multiple slots across multiple datapath processing units. However, SLOT1 and SLOT1 within a single datapath processor should not be set to the same context.
0 SLOT1_EN	SLOT1 Enable Set this bit to 0b1 to enable SLOT1 for data processing. Note, the context running in this slot (SLOT1_CTX_NUM) will not begin processing data until the global ASRC_CTX_CTRL[RUN_EN] bit is set to 0b1. This register field must be static prior to setting the associated ASRC_CTX_CTRL[RUN_EN]. 0 - Context SLOT1 is disabled 1 - Context SLOT1 is enabled

14.5.3.1.13 ASRC Datapath Processor Control SLOT1 Register1 (ASRC_PROC_CTRL_SLOT1_R10 - ASRC_PROC_CTRL_SLOT1_R13)

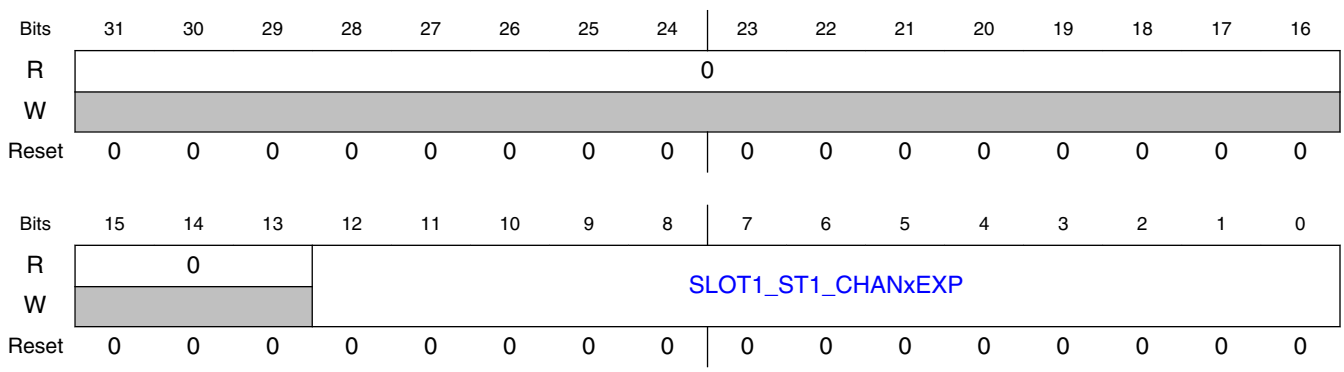
This register contains control fields to configure SLOT1 of each datapath processing unit within the design. Each processing unit contains control registers to run up to 2 independent contexts (called slots). There are a total of 8 channels supported per datapath processing unit. If a single context contains more than 8 channels, it must be split

between multiple datapath processing units. Note, setting the SLOTx_EN bit will not begin processing for the associated context. To begin processing data, the global ASRC_CTX_CTRLa[RUN_EN] must be set. Prior to setting ASRC_CTX_CTRLa[RUN_EN] for a given context, all of the slots associated with it must be statically configured in the ASRC_PROCESSOR_CTRLa registers. Each register in the ASRC_PROC_CTRL_SLOT1_R1 register array corresponds to a different context datapath processor.

14.5.3.1.13.1 Offset

Register	Offset
ASRC_PROC_CTRL_SLOT1_R10	B0h
ASRC_PROC_CTRL_SLOT1_R11	B4h
ASRC_PROC_CTRL_SLOT1_R12	B8h
ASRC_PROC_CTRL_SLOT1_R13	BCh

14.5.3.1.13.2 Diagram



14.5.3.1.13.3 Fields

Field	Description
31-13 —	Reserved
12-0 SLOT1_ST1_CHANxEXP	SLOT1 Stage1 Channels x Expansion Factor

Field	Description
	When the ASRC prefilter is configured to run in two stage mode, the first stage can be programmed to have an expansion factor (ASRC_CTX_CTRL_EXT1x[PF_EXPANSION_FACTOR]). The expansion factor specifies the number of delay units between each filter tap in the ST1 filter. For example, if the ST1 expansion factor is set to 1, then 1 unit of delay will be placed between adjacent filter taps. If the ST1 expansion factor is set to 2, then 2 units of delay will be placed between adjacent filter taps. When the prefilter is running in a 2 stage mode, this register must be programmed with the following value: `SLOT1_ST1_CHANxEXP = (ASRC_PROC_CTRL_S1_R0[SLOT1_NUM_CH]+1) * (ASRC_CTX_CTRL_EXT1x[PF_EXPANSION_FACTOR])` When the ASRC prefilter is configured to run in a single stage mode, this register must be set to 0x00.

14.5.3.1.14 ASRC Datapath Processor Control SLOT1 Register2 (ASRC_PROC_CTRL_SLOT1_R20 - ASRC_PROC_CTRL_SLOT1_R23)

This register contains control fields to configure SLOT1 of each datapath processing unit within the design. Each processing unit contains control registers to run up to 2 independent contexts (called slots). There are a total of 8 channels supported per datapath processing unit. If a single context contains more than 8 channels, it must be split between multiple datapath processing units. Note, setting the SLOTx_EN bit will not begin processing for the associated context. To begin processing data, the global ASRC_CTX_CTRLa[RUN_EN] must be set. Prior to setting ASRC_CTX_CTRLa[RUN_EN] for a given context, all of the slots associated with it must be statically configured in the ASRC_PROCESSOR_CTRLa registers. Each register in the ASRC_PROC_CTRL_SLOT1_R2 register array corresponds to a different context datapath processor.

14.5.3.1.14.1 Offset

Register	Offset
ASRC_PROC_CTRL_SLOT1_R20	C0h
ASRC_PROC_CTRL_SLOT1_R21	C4h
ASRC_PROC_CTRL_SLOT1_R22	C8h
ASRC_PROC_CTRL_SLOT1_R23	CCh

14.5.3.1.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			SLOT1_ST1_MEM_ALLOC												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SLOT1_ST1_ST_ADDR												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.14.3 Fields

Field	Description
31-29 —	Reserved

28-16
SLOT1_ST1_MEM_ALLOC
	SLOT1 Stage1 Memory Allocation
	This register will specify the amount of memory to be used by the SLOT1 ST1 prefilter. The amount of memory used here is specific to the ST1 prefilter number of taps (S1_TAPS), expansion factor (EXP), and number of channels (SLOT1_NUM_CH). To compute the amount of memory to use for the S1 prefilter, the following equation can be used:
	`SLOT1_ST1_MEM_ALLOC = [ ASRC_CTX_CTRL_EXT2x[ST1_NUM_TAPS] * (ASRC_PROC_CTRL_SLOT1_R0x[SLOT1_NUM_CH]+1) * (ASRC_CTX_CTRL_EXT1x[PF_EXPANSION_FACTOR]+1) ] + (ASRC_PROC_CTRL_SLOT1_R0x[SLOT1_NUM_CH]+1)`
	When operating in a single stage prefilter, the amount of memory can be calculated through the following equation:
	`SLOT1_ST1_MEM_ALLOC = (ASRC_PROC_CTRL_SLOT1_R0x[SLOT1_NUM_CH]+1) * (ASRC_CTX_CTRL_EXT2x[ST1_NUM_TAPS]+1)`

15-13 —	Reserved
12-0 SLOT1_ST1_ST_ADDR	SLOT1 Stage1 Start Address This register will specify the prefilter RAM base address for the SLOT1 ST1 prefilter. The value selected here must prevent SLOT0 ST1, SLOT0 ST2, SLOT1 ST1, and SLOT1 ST2 from overlapping in RAM. For example, if SLOT1_ST1_ST_ADDR = 0x0, and SLOT1_ST1_MEM_ALLOC = 0x10, then the address range of 0x0 - 0x0F is consumed by SLOT1 S1, and no other masters in this context processor can occupy this address range.

14.5.3.1.15 ASRC Datapath Processor Control SLOT1 Register3 (ASRC_PROC_CTRL_SLOT1_R30 - ASRC_PROC_CTRL_SLOT1_R33)

This register contains control fields to configure SLOT1 of each datapath processing unit within the design. Each processing unit contains control registers to run up to 2 independent contexts (called slots). There are a total of 8 channels supported per datapath processing unit. If a single context contains more than 8 channels, it must be split between multiple datapath processing units. Note, setting the SLOTx_EN bit will not begin processing for the associated context. To begin processing data, the global ASRC_CTX_CTRLa[RUN_EN] must be set. Prior to setting ASRC_CTX_CTRLa[RUN_EN] for a given context, all of the slots associated with it must be statically configured in the ASRC_PROCESSOR_CTRLa registers. Each register in the ASRC_PROC_CTRL_SLOT1_R3 register array corresponds to a different context datapath processor.

14.5.3.1.15.1 Offset

Register	Offset
ASRC_PROC_CTRL_SLOT1_R30	D0h
ASRC_PROC_CTRL_SLOT1_R31	D4h
ASRC_PROC_CTRL_SLOT1_R32	D8h
ASRC_PROC_CTRL_SLOT1_R33	DCh

14.5.3.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			SLOT1_ST2_MEM_ALLOC												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SLOT1_ST2_ST_ADDR												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.15.3 Fields

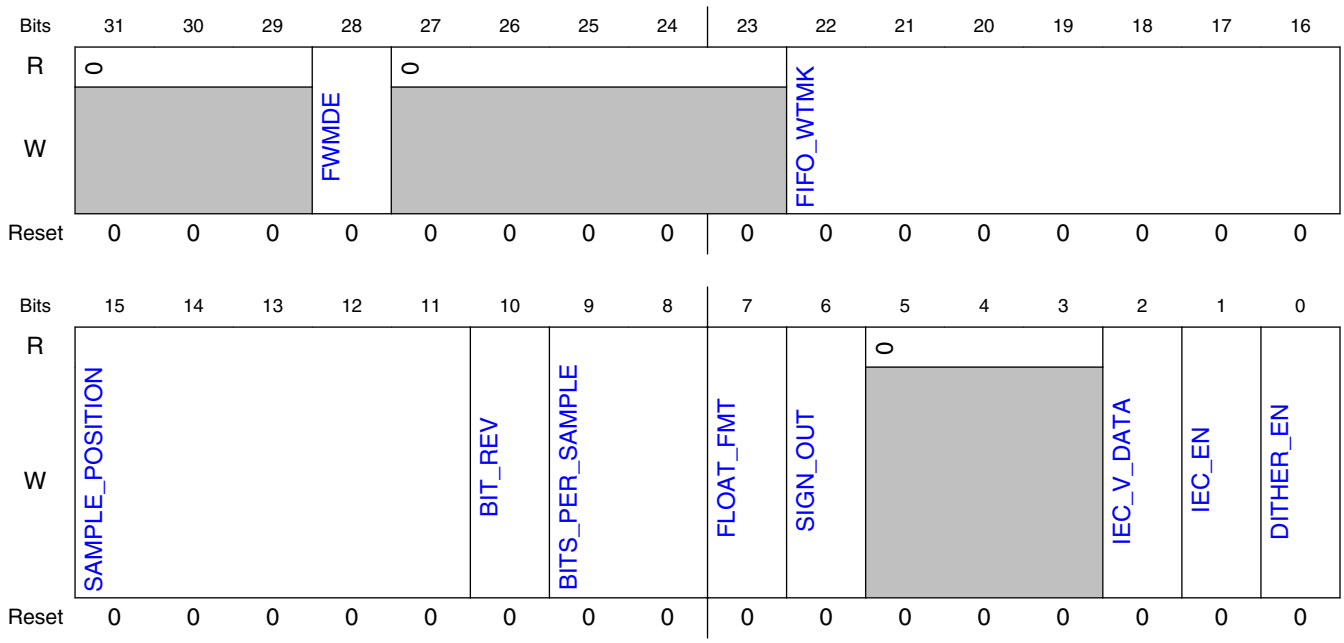
Field	Description
31-29 —	Reserved
28-16 SLOT1_ST2_MEM_ALLOC	SLOT1 Stage2 Memory Allocation This register will specify the amount of memory to be used by the SLOT1 ST2 prefilter. The amount of memory used here is specific to the ST2 prefilter number of taps (ASRC_CTX_CTRL_EXT2x[ST2_NUM_TAPS]+1), and number of channels (ASRC_PROC_CTRL_SLOT1_R0x[SLOT1_NUM_CH]+1). To compute the amount of memory to use for the ST2 prefilter, the following equation can be used: `SLOT1_ST2_MEM_ALLOC = (ASRC_PROC_CTRL_SLOT1_R0x[SLOT1_NUM_CH]+1) * (ASRC_CTX_CTRL_EXT2x[ST2_NUM_TAPS]+1)` Note, this value is only used if the Stage2 prefilter is enabled for the context programmed to the SLOT1_CTX_NUM field.
15-13 —	Reserved
12-0 SLOT1_ST2_ST_ADDR	SLOT1 Stage2 Start Address This register will specify the prefilter RAM base address for the SLOT1 ST2 prefilter. The value selected here must prevent SLOT0 ST1, SLOT0 ST2, SLOT1 ST1, and SLOT1 ST2 from overlapping in RAM. For example, if SLOT1_ST2_ST_ADDR = 0x0, and SLOT1_ST2_MEM_ALLOC = 0x10, then the address range of 0x0 - 0x0F is consumed by SLOT1 S2, and no other masters in this context processor can occupy this address range.

14.5.3.1.16 ASRC Context Output Control (ASRC_CTX_OUT_CTRL0 - ASRC_CTX_OUT_CTRL3)

14.5.3.1.16.1 Offset

Register	Offset
ASRC_CTX_OUT_CTRL0	E0h
ASRC_CTX_OUT_CTRL1	E4h
ASRC_CTX_OUT_CTRL2	E8h
ASRC_CTX_OUT_CTRL3	ECh

14.5.3.1.16.2 Diagram



14.5.3.1.16.3 Fields

Field	Description
31-29 —	Reserved
28 FWMDE	Output FIFO Watermark DMA Enable This bit is used to enable DMA requests for FIFO_WTMK. A DMA request will be generated if the number of sample groups stored in the output FIFO is > the FIFO_WTMK and FWMDE is set. 0 - Output DMA Requests Not Enabled for This Context 1 - Output DMA Requests Enabled for This Context
27-23 —	Reserved
22-16 FIFO_WTMK	Context Output FIFO Watermark The FIFO watermark level will be used to generate an output FIFO status signal. This status signal can be utilized when servicing DMA requests for sample data. If the number of sample groups in the output FIFO is > the FIFO_WTMK then the associated ASRC_CTRL_FIFO_STATUSa[OUTFIFO_WTMK] will be flagged as 0b1. Note, the output FIFO is sized to support a maximum of 64 sample groups per context.
15-11 SAMPLE_POSITION	Sample Position This register field is used to specify the position of the first valid sample bit within a 32 bit boundary. The position is indexed starting from 0. For example, if the sample is 16-bits wide and left-justified on a 32-bit boundary, then this register should be set to 16. If the sample is 16-bits wide and right-justified on a 32-bit boundary, then this register should be set to 0. Bit-Reversal of the sample data should not be considered when programming this field.
10	Sample Bit-Reversal

Table continues on the next page...

Field	Description
BIT_REV	This register field is used to reverse the bit ordering of output sample data. The BIT_REV operation only operates on sample data and will not impact iec60958 bit field insertion. 0 - No change. 1 - Bit-reverse sample data.
9-8 BITS_PER_SAMPLE	Number of Bits Per Audio Sample This register field is used to specify the number of bits per sample within this context. Note, this width doesn't account for meta-data associated with a sample. 00 - 16-bits Per Sample 01 - 20-bits Per Sample 10 - 24-bits Per Sample 11 - 32-bits Per Sample
7 FLOAT_FMT	Context Output Floating Point Format This register field is used to select the output sample data format. The ASRC can output signed integer samples or 32-bit floating point samples. Note, if FLOAT_FMT is set to 0b1 then BITS_PER_SAMPLE must be set to 0b11, and SAMPLE_POSITION must be set to 0x00. 0 - Integer Format 1 - Single Precision Floating Point Format
6 SIGN_OUT	Output Data Sign This bit is used to convert output data to an unsigned format. SIGN_OUT is only active in 32, 24, 20, or 16-bit integer modes. Note, an unsigned conversion should only be made when the ASRC_CTX_CTRLx[SIGN_IN] is set. 0 - Signed Format 1 - Convert to Unsigned
5-3 —	Reserved
2 IEC_V_DATA	IEC60958 Validity Flag This register is used directly as the IEC60958 validity flag for the associated context.
1 IEC_EN	IEC60958 Bit-Field Insertion Enable This register is used to enable IEC60958 field insertion on the output sample data. This mode is only supported when BITS_PER_SAMPLE != 0b11. SAMPLE_POSITION must be adjusted to ensure that output sample data is properly aligned in the sample bit-field defined by IEC60958. SAMPLE_POSITION must be programmed to shift the sample MSB to position 27 of the output sample word. If BITS_PER_SAMPLE = 0b00, SAMPLE_POSITION = 12. If BITS_PER_SAMPLE = 0b01, SAMPLE_POSITION = 8. If BITS_PER_SAMPLE = 0b10, SAMPLE_POSITION = 4. If BITS_PER_SAMPLE = 0b11, IEC60958 data insertion is not supported. 0 - No Data Insertion Enabled. 1 - IEC60958 Bit-Field Insertion Enabled.
0 DITHER_EN	Output Dither Enable The output path contains a TPDF dither function. To enable this function, this bit must be set to 0b1. Dither is supported in all fixed point output modes (16, 20, 24, and 32-bit). Dither is not supported in 32bit floating point output mode.

14.5.3.1.17 ASRC Control Output Access (ASRC_CTRL_OUT_ACCESS0 - ASRC_CTRL_OUT_ACCESS3)

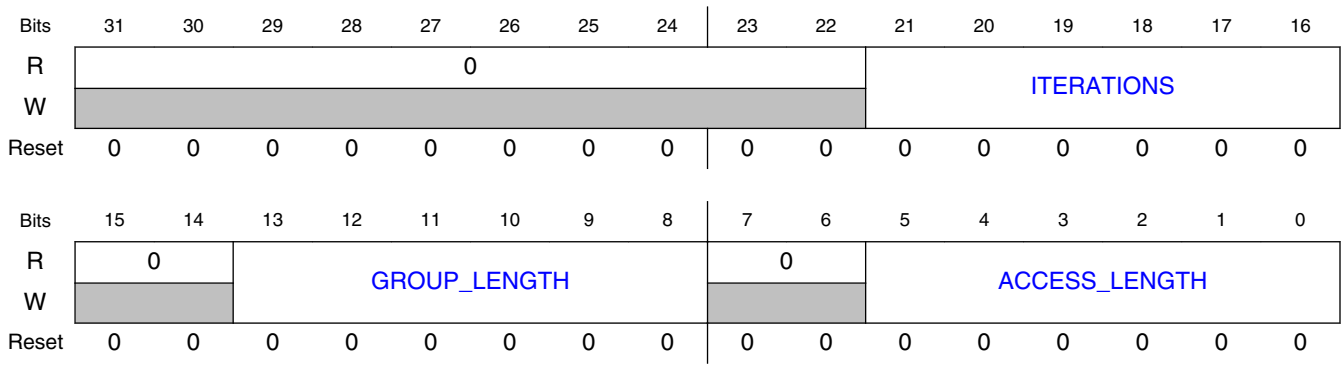
This register is used to describe how output samples are interleaved when being read from the ASRC. The ASRC has general interleaving hardware built into the output stage. This interleaving format supports writing data to multiple sources per burst, a programmable number of channels per source, and variable length burst accesses.

A single burst transaction can be configured to split channels across (GROUP_LENGTH/ ACCESS_LENGTH) number of destinations. Note, the channels output to a single destination will be in an interleaved format. The ASRC can be configured for a burst transaction which fetches ITERATIONS*ACCESS_LENGTH samples to be routed to one destination. The number of channels per destination is ACCESS_LENGTH. If all channels associated with a given context are contained within a single destination, then the GROUP_LENGTH should be set equal to the ACCESS_LENGTH. The total number of samples which need to be read from the ASRC in a complete burst access will be equal to GROUP_LENGTH*ITERATIONS.

14.5.3.1.17.1 Offset

Register	Offset
ASRC_CTRL_OUT_ACCESS0	F0h
ASRC_CTRL_OUT_ACCESS1	F4h
ASRC_CTRL_OUT_ACCESS2	F8h
ASRC_CTRL_OUT_ACCESS3	FCh

14.5.3.1.17.2 Diagram



14.5.3.1.17.3 Fields

Field	Description
31-22 —	Reserved
21-16 ITERATIONS	Number of Sequential Fetches Per Channel Group This register selects the number of sequential ACCESS_LENGTH reads from a single set of channels prior to iterating to the next set. For example, if one audio source destination will contain two channels of data (ACCESS_LENGTH == 0x2), and every access to this source will write 32 samples, then the ITERATIONS parameter should be set to 0x10. ITERATIONS is indexed from one. If ITERATIONS is set to 0x0 this will result in an invalid configuration.
15-14 —	Reserved
13-8 GROUP_LENGTH	Number of Channels in a Context This register has units of channels/context. When combined with ACCESS_LENGTH, these parameters will inform the ASRC how many independent destinations will be used in a burst read. The number of destinations per burst is (GROUP_LENGTH/ACCESS_LENGTH). This register is indexed from one. If GROUP_LENGTH == 0x0 this will result in an invalid configuration. For example, if the context has 2 channels associated with it then GROUP_LENGTH should be programmed to 0x2. The maximum valid GROUP_LENGTH value is 0x20. GROUP_LENGTH must be programmed to an integer multiple of ACCESS_LENGTH.
7-6 —	Reserved
5-0 ACCESS_LENGTH	Number Of Channels Per Destination This register field allows a context containing many channels to be split into sub-groups of channels to be forwarded to separate destinations. For example, if a single context contains 6 channels, but these results are to be output to 3 independent destinations then the ACCESS_LENGTH parameter can be set to 0x2 (2 channels/destination). This allows sequential reads from a context output FIFO to only fetch channel data associated with one destination at a time. This register has units of channels/destination. The total number of samples output to a single destination before iterating to the next is ACCESS_LENGTH*ITERATIONS. For example, if a single destination will contain audio data for two channels, then ACCESS_LENGTH should be set to 0x2. Note 0x0 is not a valid ACCESS_LENGTH configuration.

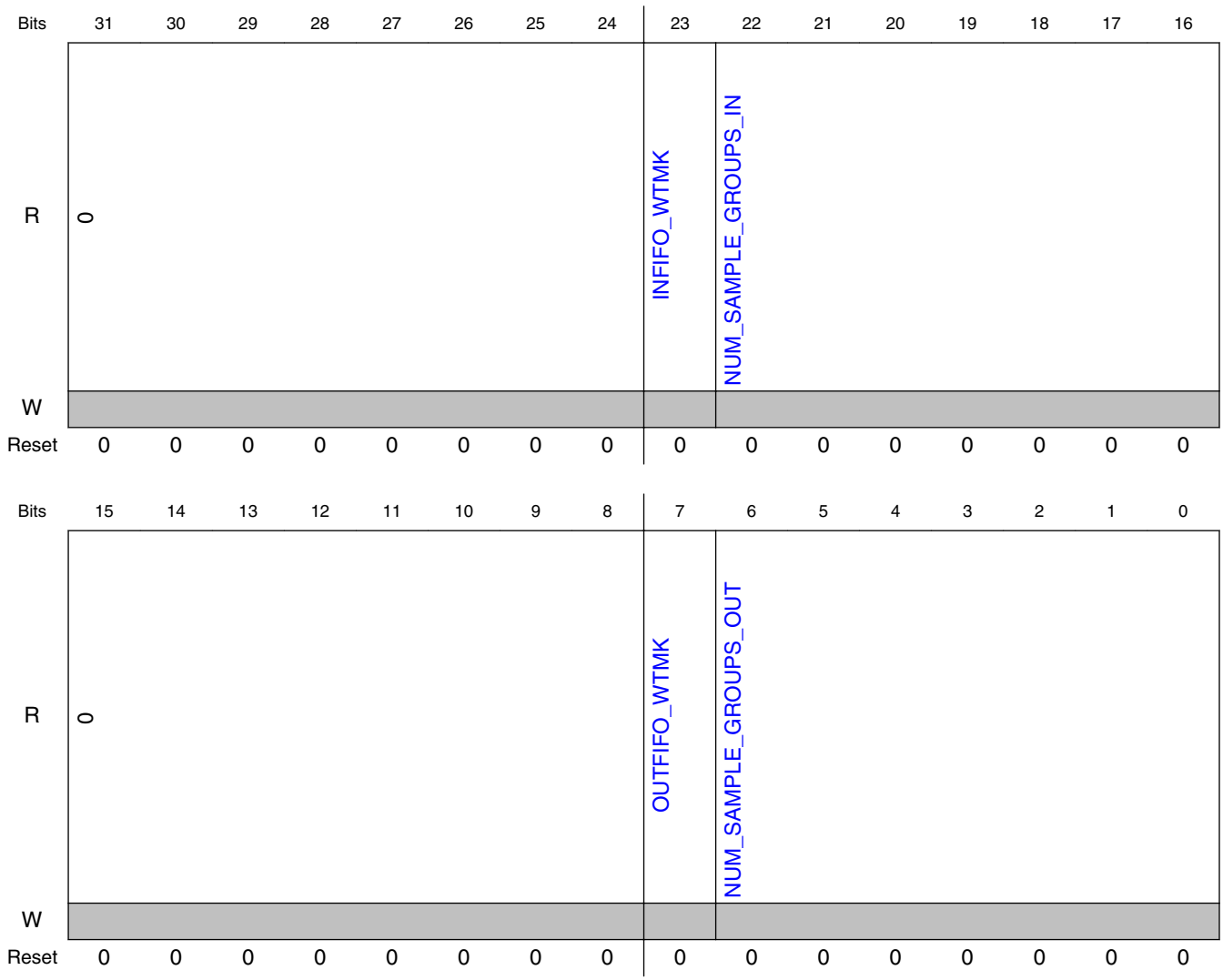
14.5.3.1.18 ASRC Sample FIFO Status (ASRC_SAMPLE_FIFO_STATUS0 - ASRC_SAMPLE_FIFO_STATUS3)

This register provides a status for the number of samples stored in the input and output FIFOs. Note, the status fields track the number of sample groups associated with each context. All data is handled on a sample group granularity internal to the ASRC.

14.5.3.1.18.1 Offset

Register	Offset
ASRC_SAMPLE_FIFO_STATUS0	100h
ASRC_SAMPLE_FIFO_STATUS1	104h
ASRC_SAMPLE_FIFO_STATUS2	108h
ASRC_SAMPLE_FIFO_STATUS3	10Ch

14.5.3.1.18.2 Diagram



14.5.3.1.18.3 Fields

Field	Description
31-24 —	Reserved
23 INFIFO_WTMK	Input FIFO Watermark Flag This flag will be set to 0b1 when the number of sample groups contained within the input FIFO is less than the associated ASRC_CTX_CTRLa[FIFO_WTMK]. Note, the associated context ASRC_CTX_CTRLa[RUN_EN] muse be set to 0b1 for this field to be active.
22-16 NUM_SAMPLE_GROUPS_IN	Number Of Sample Groups Stored in Input FIFO This register holds the number of sample groups per context contained in the input FIFO. Note, when performing a burst write which has non-equivalent ACCESS_LENGTH and GROUP_LENGTH, this pointer will only update once a complete sample group has been written to the FIFO. This could take

Table continues on the next page...

Field	Description
	multiple ITERATIONS*ACCESS_LENGTH cycles until the first sample group has been completely written to the FIFO.
15-8 —	Reserved
7 OUTFIFO_WTMK	Output FIFO Watermark Flag This flag will be set to 0b1 when the number of sample groups contained within the output FIFO is greater than the associated ASRC_CTX_OUT_CTRLa[FIFO_WTMK]. Note, the associated context ASRC_CTX_CTRLa[RUN_EN] must be set to 0b1 for this field to be active.
6-0 NUM_SAMPLE_GROUPS_OUT	Number Of Sample Groups Stored in the output FIFO This register holds the number of sample groups per context stored in the output FIFO. Note, when performing a burst read which has non-equivalent ACCESS_LENGTH and GROUP_LENGTH, this pointer will only update once a complete sample group has been read from the FIFO. This could take multiple ITERATIONS*ACCESS_LENGTH cycles until the first sample group has been completely read from the FIFO.

14.5.3.1.19 ASRC Resampling Ratio Low (ASRC_RS_RATIO_LOW0 - ASRC_RS_RATIO_LOW3)

The ASRC_RS_RATIO_LOW register can only be modified before a context has began processing data. A write to ASRC_RS_RATIO_LOW will only be accepted if the respective ASRC_CTX_CTRL[RUN_EN] = 0x0, or if the context has not began processing samples yet. Note, when starting up a new context, the ASRC will stall processing of data until an initial write to {ASRC_RS_RATIO_HIGH,ASRC_RS_RATIO_LOW} has occurred. Subsequent writes to this register will be ignored while the context is actively processing data. Updates to the resampling ratio can be made while the a context is active through the ASRC_RS_UPDATE_CTRL register. The order of write operations should always follow ASRC_RS_RATIO_LOW followed by ASRC_RS_RATIO_HIGH. Note, writes to ASRC_RS_RATIO_HIGH without a previous write to ASRC_RS_RATIO_LOW will be considered invalid. In addition, a write to ASRC_RS_RATIO_LOW without a subsequent write to ASRC_RS_RATIO_HIGH will be considered invalid. This register is 32-bit accessible only.

14.5.3.1.19.1 Offset

Register	Offset
ASRC_RS_RATIO_LOW0	110h
ASRC_RS_RATIO_LOW1	118h
ASRC_RS_RATIO_LOW2	120h

Table continues on the next page...

Register	Offset
ASRC_RS_RATIO_LOW3	128h

14.5.3.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RS_RATIO_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RS_RATIO_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.19.3 Fields

Field	Description
31-0 RS_RATIO_LOW	Resampling Ratio Low This register contains the lower 32-bits of the 44-bit fixed point resampling ratio between input to output sources. The concatenation of {RS_RATIO_HIGH,RS_RATIO_LOW} represents the 44-bit unsigned integer and fractional phase offset between input samples to output samples. The resampling ratio is calculated by dividing the input sample frequency by the output sample frequency Fs(in)/Fs(out). The format of the resampling ratio is dependent on the number of taps in the resampler (ASRC_CTX_RS_COEFF_CTRL[NUM_RES_TAPS]). If the resampler is in 32-tap mode, the resampling ratio is formatted with with 5 integer bits and 39 fractional bits (5.39 format). If the resampler is in 64-tap mode, the resampling ratio is formatted with 6 integer bits and 38 fractional bits (6.38 format). If the resampler is in 128-tap mode, the resampling ratio is formatted with 7 integer bits and 37 fractional bits (7.37 format). Note, the maximum up-sampling ratio supported is Fs(in)/Fs(out) = 1/8. The maximum down-sampling ratio supported is Fs(in)/Fs(out) = 16.

14.5.3.1.20 ASRC Resampling Ratio High (ASRC_RS_RATIO_HIGH0 - ASRC_RS_RATIO_HIGH3)

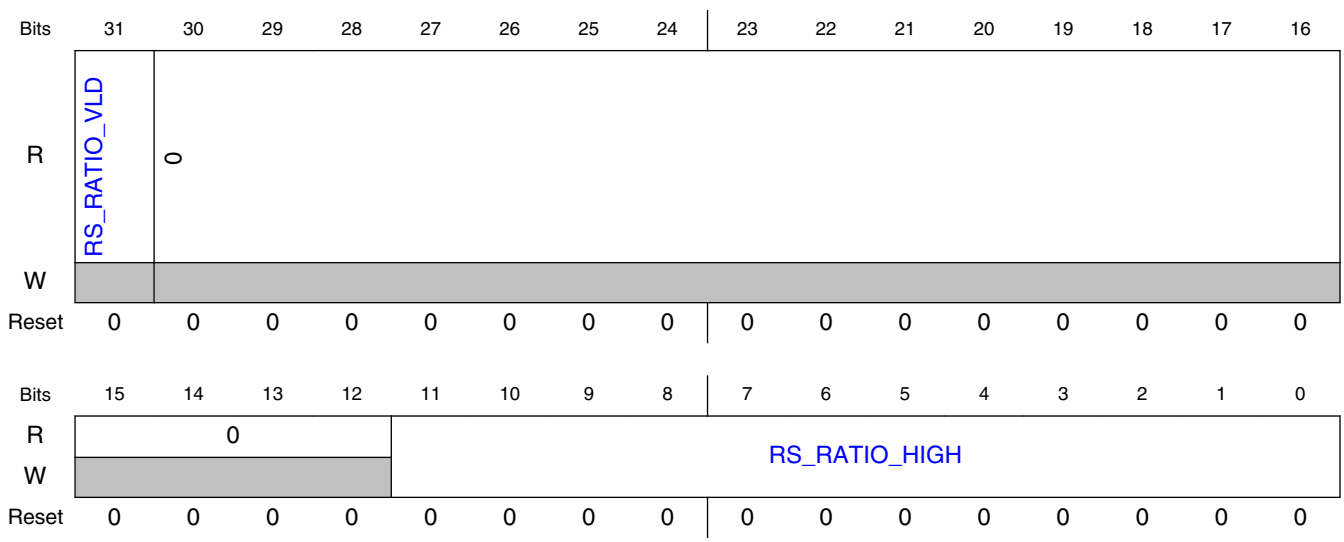
The ASRC_RS_RATIO_HIGH register can only be modified before a context has began processing data. A write to ASRC_RS_RATIO_HIGH will only be accepted if the respective ASRC_CTX_CTRL[RUN_EN] = 0x0, or if the respective context has not began processing samples. Note, when starting up a new context, the ASRC will stall processing of data until an initial write to {ASRC_RS_RATIO_HIGH,ASRC_RS_RATIO_LOW} has occurred. Subsequent writes to this register will be ignored while the context is actively processing data. Updates to

the resampling ratio can be made while the a context is active through the ASRC_RS_UPDATE_CTRL register. The order of write operations should always follow ASRC_RS_RATIO_LOW followed by ASRC_RS_RATIO_HIGH. Note, writes to ASRC_RS_RATIO_HIGH without a previous write to ASRC_RS_RATIO_LOW will be considered invalid. In addition, a write to ASRC_RS_RATIO_LOW without a subsequent write to ASRC_RS_RATIO_HIGH will be considered invalid. This register is 32-bit accessible only.

14.5.3.1.20.1 Offset

Register	Offset
ASRC_RS_RATIO_HIGH0	114h
ASRC_RS_RATIO_HIGH1	11Ch
ASRC_RS_RATIO_HIGH2	124h
ASRC_RS_RATIO_HIGH3	12Ch

14.5.3.1.20.2 Diagram



14.5.3.1.20.3 Fields

Field	Description
31	Resampling Ratio Valid

Table continues on the next page...

Field	Description
RS_RATIO_VLD	This register indicates that a valid resampling ratio has been programmed to {RS_RATIO_HIGH,RS_RATIO_LOW}. This bit will only be set to 1'b1 if a valid write sequence has been executed. For a sequence to be considered valid, a write to RS_RATIO_LOW must precede a write to RS_RATIO_HIGH.
30-12 —	Reserved
11-0 RS_RATIO_HIGH	Resampling Ratio High This register contains the upper 12-bits of the 44-bit fixed point resampling ratio between input to output sources. The concatenation of {RS_RATIO_HIGH,RS_RATIO_LOW} represents the 44-bit unsigned integer and fractional phase offset between input samples to output samples. This value is calculated by dividing the input sample frequency by the output sample frequency Fs(in)/Fs(out). The format of the resampling ratio is dependent on the number of taps in the resampler (ASRC_CTX_RS_COEFF_CTRL[NUM_RES_TAPS]). If the resampler is in 32-tap mode, the resampling ratio is formatted with with 5 integer bits and 39 fractional bits. If the resampler is in 64-tap mode, the resampling ratio is formatted with 6 integer bits and 38 fractional bits. If the resampler is in 128-tap mode, the resampling ratio is formatted with 7 integer bits and 37 fractional bits. Note, the maximum up-sampling ratio supported is Fs(in)/Fs(out) = 1/8. The maximum down-sampling ratio supported is Fs(in)/Fs(out) = 16.

14.5.3.1.21 ASRC Resampling Ratio Update Control (ASRC_RS_UPDATE_CTRL0 - ASRC_RS_UPDATE_CTRL3)

The resampling ratio update control register is used make modifications to the resampling ratio while a context is active processing samples.

14.5.3.1.21.1 Offset

Register	Offset
ASRC_RS_UPDATE_CTRL0	130h
ASRC_RS_UPDATE_CTRL1	134h
ASRC_RS_UPDATE_CTRL2	138h
ASRC_RS_UPDATE_CTRL3	13Ch

14.5.3.1.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RS_RATIO_MOD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RS_RATIO_MOD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.21.3 Fields

Field	Description
31-0 RS_RATIO_MOD	Resampling Ratio Modifier The RS_RATIO_MOD register field is a 32-bit two's compliment signed value which is used to adjust the resampling ratio in the ASRC. When a write is performed to this register, the RS_RATIO_MOD will be automatically added to an accumulator containing pending resampling ratio updates. Pending resampling ratio updates are added to the lowest 31-bits of the 44-bit resampling ratio after each output sample time. Note, the rate of change of the resampling ratio is limited by RS_RATIO_RAMP_RATE. If the value programmed in RS_RATIO_MOD is > RS_RATIO_RAMP_RATE then the update will be distributed over RS_RATIO_MOD / RS_RATIO_RAMP_RATE output sample times.

14.5.3.1.22 ASRC Resampling Ratio Update Rate (ASRC_RS_UPDATE_RATE0 - ASRC_RS_UPDATE_RATE3)

The resampling ratio update rate register is used limit the resampling ratio update rate per output sample time.

14.5.3.1.22.1 Offset

Register	Offset
ASRC_RS_UPDATE_RATE0	140h
ASRC_RS_UPDATE_RATE1	144h
ASRC_RS_UPDATE_RATE2	148h
ASRC_RS_UPDATE_RATE3	14Ch

14.5.3.1.22.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	RS_RATIO_RAMP_RATE														
W																
Reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RS_RATIO_RAMP_RATE															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

14.5.3.1.22.3 Fields

Field	Description
31 —	Reserved
30-0 RS_RATIO_RAMP_RATE	Resampling Ratio Ramp Rate The RS_RATIO_RAMP_RATE defines the maximum rate of change for the internal resampling ratio. This is an unsigned value which represents the magnitude of the maximum change to the resampling ratio per output sample time. If the value programmed in RS_RATIO_MOD is > RS_RATIO_RAMP_RATE then the update will be distributed across (RS_RATIO_MOD/RS_RATIO_RAMP_RATE) output sample times. Note, RS_RATIO_RAMP_RATE applies to the lower 31-bits of the 44-bit resampling ratio. This caps the maximum rate of change for the resampling ratio. Note, the specific rate is dependent on the number or taps in the resampler. If there are 128 taps, the max rate of change is max(dFsin/dFsout) = (2^31)/(2^38), in 64 tap mode the limit is (2^31)/(2^39), and in 32-tap mode the limit is (2^31)/(2^40).

14.5.3.1.23 ASRC Resampling Center Tap Coefficient Low (ASRC_RS_CT_LOW)

Contains lower 32-bits of a 64-bit double precision floating point value representing the center tap coefficient used by the resampling stage.

14.5.3.1.23.1 Offset

Register	Offset
ASRC_RS_CT_LOW	150h

14.5.3.1.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RS_CT_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RS_CT_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.23.3 Fields

Field	Description
31-0 RS_CT_LOW	Resampling Center Tap Coefficient LSBs

14.5.3.1.24 ASRC Resampling Center Tap Coefficient High (ASRC_RS_CT_HIGH)

This register contains the upper 32-bits of a 64-bit double precision floating point value representing the center tap coefficient used by the resampling stage. Note, the double precision floating point format contains 1 sign bit, 11 exponent bits, and 52 mantissa bits. This register will hold the sign bit, 11 exponent bits, and the upper 20 mantissa bits concatenated.

14.5.3.1.24.1 Offset

Register	Offset
ASRC_RS_CT_HIGH	154h

14.5.3.1.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RS_CT_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RS_CT_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.24.3 Fields

Field	Description
31-0 RS_CT_HIGH	Resampling Center Tap Coefficient MSBs This register field contains the upper 32-bits of a 64-bit double precision floating point value representing the center tap coefficient used by the resampling stage

14.5.3.1.25 ASRC Prefilter Coefficient FIFO (ASRC_PRE_COEFF_FIFO0 - ASRC_PRE_COEFF_FIFO3)

The prefilter coefficient FIFOs store the coefficients used by the ASRC prefilter stage1 (ST1) and stage2 (ST2). Both the ST1 and ST2 filters are designed to be symmetric around the center tap. Each coefficient FIFO contains 192 entries to be shared between the ST1 and ST2 filters. This places a maximum of 394 taps on the number of cumulative filter taps which can be shared between the ST1 and ST2 filter stages. The length of each filter is programmable using the ASRC_CTX_CTRL_EXT2a[STx_NUM_TAPS] register fields. Note, the coefficients are stored in 64-bit floating point format. This requires 2 32-bit register writes to update each coefficient. After a coefficient has been updated, the write pointer will be automatically indexed to the next entry in the FIFO. To reset the FIFO write pointer, toggle the state of ASRC_CTX_CTRL_EXT1a[PF_COEFF_MEM_RST]. By default, a write to this register will update the coefficients for the ST1 filter stage. The ASRC_CTX_CTRL_EXT1a[PF_COEFF_STAGE_WR] bit must be set to 0x1 to write ST2 coefficients for the selected context.

14.5.3.1.25.1 Offset

Register	Offset
ASRC_PRE_COEFF_FIFO0	160h
ASRC_PRE_COEFF_FIFO1	164h
ASRC_PRE_COEFF_FIFO2	168h
ASRC_PRE_COEFF_FIFO3	16Ch

14.5.3.1.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	COEFF_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COEFF_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.25.3 Fields

Field	Description
31-0 COEFF_DATA	Coefficient Value For Prefilter Each coefficient FIFO contains up to 192 entries. The coefficients are stored in 64-bit IEEE floating point format. This format contains 1 sign bit, 11 biased exponent bits, and 52 mantissa bits. To modify a single element, two register writes are required starting with the LSBs of the value to be modified. The first access to this register should contain the lower 32-bits of the mantissa, and the second access should contain the sign bit, 11 exponent bits, and upper 20 mantissa bits concatenated. The FIFO pointers are automatically indexed after each access. To reset a FIFO write pointer back to zero, toggle the state of ASRC_CTX_CTRL_EXT1a[PF_COEFF_MEM_RST]. Note, this can also be used to cancel an outstanding 32-bit write operation which isn't complete. The FIFOs can only be written when the associated ASRC_CTX_CTRLa[RUN_EN] is set to 0b0. Writes to this register take lower priority than internal accesses. Wait states will be added on the bus if multiple masters are attempting to access this FIFO at the same time.

14.5.3.1.26 ASRC Context Resampling Coefficient Memory (ASRC_CTX_RS_COEFF_MEM)

14.5.3.1.26.1 Offset

Register	Offset
ASRC_CTX_RS_COEFF_MEM	170h

14.5.3.1.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	RS_COEFF_WDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	RS_COEFF_WDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.26.3 Fields

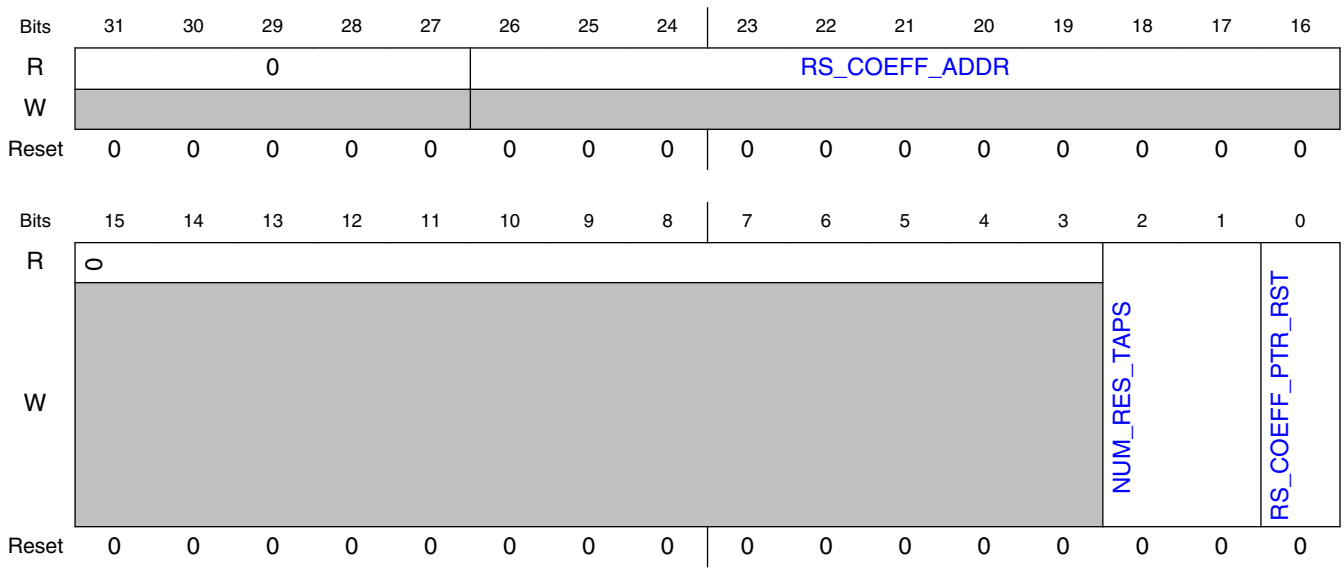
Field	Description
31-0 RS_COEFF_WDATA	Resampling Coefficient Write Data If NUM_RES_TAPS = 0b00, the resampling coefficient memory contains 16 taps and 128 phase entries per tap. If NUM_RES_TAPS = 0b01, the resampling coefficient memory contains 32 taps and 64 phase entries per tap. If NUM_RES_TAPS = 0b10, the resampling coefficient memory contains 64 taps and 32 phase entries per tap. Each phase entry contains 4 64-bit floating point coefficients. The coefficients can be modified by writing to ASRC_CTX_RS_COEFF_MEM[RS_COEFF_WDATA]. The FIFO pointer (RS_COEFF_ADDR) is automatically indexed after each 256-bit access. To reset the FIFO write pointer back to zero, toggle the state of ASRC_CTX_RS_COEFF_CTRL[RS_COEFF_PTR_RST] from 0b0 to 0b1 and back to 0b0. The FIFO write pointer will be held in reset as long as ASRC_CTX_RS_COEFF_CTRL[RS_COEFF_PTR_RST] is set to 0b1. Note, this can also be used to cancel an outstanding 32-bit write operation which isn't complete. The FIFOs can only be written when all ASRC_CTX_CTRL[RUN_EN] bits in the design are set to 0b0. Each entry in the RAM is 256-bits wide. 8 writes to RS_COEFF_WDATA must be completed prior to a single entry being updated. Writes will be ordered internally from LSB to MSB.

14.5.3.1.27 ASRC Context Resampling Coefficient Control (ASRC_CTX_RS_COEFF_CTRL)

14.5.3.1.27.1 Offset

Register	Offset
ASRC_CTX_RS_COEFF_CTRL	174h

14.5.3.1.27.2 Diagram



14.5.3.1.27.3 Fields

Field	Description
31-27 —	Reserved
26-16 RS_COEFF_ADDR	Resampling Coefficient Address This is the current resampling coefficient access address being written to with RS_COEFF_WDATA. Note, this address is comprised of two concatenated fields: If NUM_RES_TAPS == 0b00, the address is comprised of {filter_tap[3:0],phase[6:0]} If NUM_RES_TAPS == 0b01, the address is comprised of {filter_tap[4:0],phase[5:0]} If NUM_RES_TAPS == 0b10, the address is comprised of {filter_tap[5:0],phase[4:0]}
15-3	Reserved

Table continues on the next page...

Field	Description
—	
2-1 NUM_RES_TAPS	Number of Resampling Coefficient Taps The second stage data processing pipeline includes a fractional resampling filter. The number of filter taps and filter coefficients can be selected as 32, 64, or 128. 00 - 32-Tap Resampling Filter 01 - 64-Tap Resampling Filter 10 - 128-Tap Resampling Filter 11 - N/A
0 RS_COEFF_PTR_RST	Resampling Coefficient Write Pointer Reset The FIFO pointers are automatically indexed after each write access to RS_COEFF_WDATA. To reset a FIFO write pointer back to zero, toggle the state of RS_COEFF_PTR_RST. Note, this can also be used to cancel an outstanding 32-bit write operation which isn't complete.

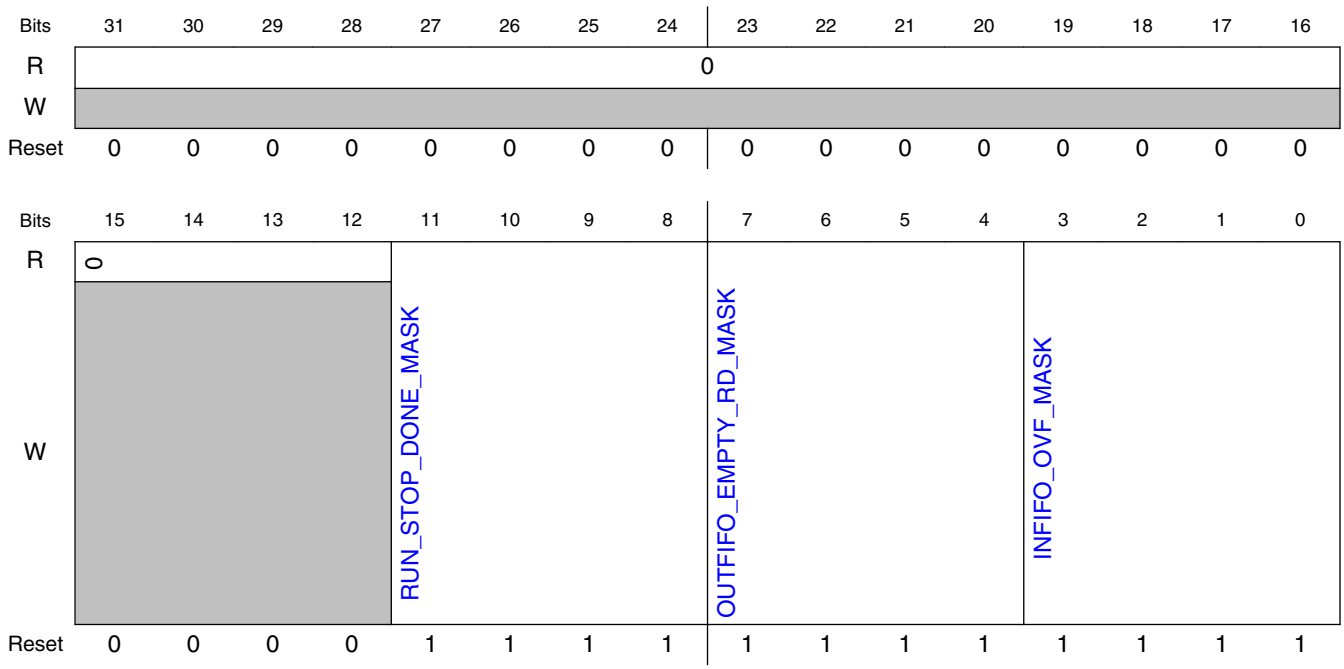
14.5.3.1.28 ASRC Interrupt Control (ASRC_IRQ_CTRL)

The ASRC_IRQ_CTRL register provides the mask bits for all interrupt sources within the ASRC. By default, all interrupt sources are disabled. To enable an interrupt source, the respective mask bit should be clear.

14.5.3.1.28.1 Offset

Register	Offset
ASRC_IRQ_CTRL	178h

14.5.3.1.28.2 Diagram



14.5.3.1.28.3 Fields

Field	Description
31-12 —	Reserved
11-8 RUN_STOP_DONE_MASK	ASRC RUN STOP DONE MASK The RUN_STOP_DONE flag indicates the previous ASRC_CTX_CTRL[RUN_STOP] operation has been completed. The ASRC context will be IDLE and all output samples will have been written to the output FIFO. Each bit in this register corresponds to a single context. Bit 8 corresponds to Context0, Bit 9 corresponds to Context1, ... The IRQ associated with each context is enabled when it's respective bit position is set to 0b0. 0000 - The RUN_STOP_DONE interrupt is enabled for Context 0 to 3. 0001 - The RUN_STOP_DONE interrupt is disabled for Context 0 and enabled for Context 1 to 3. 0010 - The RUN_STOP_DONE interrupt is disabled for Context 1 and enabled for Context 0, 2, and 3. 0011-1110 - The RUN_STOP_DONE interrupt is enabled for any context with a 1'b0 bit field. 1111 - The RUN_STOP_DONE interrupt is disabled for Context 0 to 3.
7-4 OUTFIFO_EMPTY_RD_MASK	ASRC Output FIFO Empty Read Mask The OUTFIFO_EMPTY_RD flags are triggered if a read is requested from an empty output FIFO. In this case, the return data from the read will be invalid. Note, the ASRC tracks conversions on a sample group granularity. This flag triggers if the sample group being accessed has not been completely written to the FIFO prior to a read being requested. Each bit in this register corresponds to a single context. Bit 4 corresponds to Context0, Bit 5 corresponds to Context1, ... The IRQ associated with each context is enabled when it's respective bit position is set to 0b0.

Table continues on the next page...

Field	Description
	0000 - The OUTFIFO_EMPTY_RD interrupt is enabled for Context 0 to 3. 0001 - The OUTFIFO_EMPTY_RD interrupt is disabled for Context 0 and enabled for Context 1 to 3. 0010 - The OUTFIFO_EMPTY_RD interrupt is disabled for Context 1 and enabled for Context 0, 2, and 3. 0011-1110 - The OUTFIFO_EMPTY_RD interrupt is enabled for any context with a 1'b0 bit field. 1111 - The OUTFIFO_EMPTY_RD interrupt is disabled for Context 0 to 3.
3-0 INFIFO_OVF_MASK	ASRC Input FIFO Overflow Mask The INFIFO_OVF flags are triggered if an input FIFO has overflown. This condition may occur if writes are issued to the ASRC_WRFIFO registers faster then the input samples can be processed by the ASRC internal logic or the ASRC_RDFIFO consumer. Each bit in this register corresponds to a single context. Bit 0 corresponds to Context0, Bit 1 corresponds to Context1, ... The IRQ associated with each context is enabled when it's respective bit position is set to 0b0. 0000 - The INFIFO_OVF interrupt is enabled for Context 0 to 3. 0001 - The INFIFO_OVF interrupt is disabled for Context 0 and enabled for Context 1 to 3. 0010 - The INFIFO_OVF interrupt is disabled for Context 1 and enabled for Context 0, 2, and 3. 0011-1110 - The INFIFO_OVF interrupt is enabled for any context with a 1'b0 bit field. 1111 - The INFIFO_OVF interrupt is disabled for Context 0 to 3.

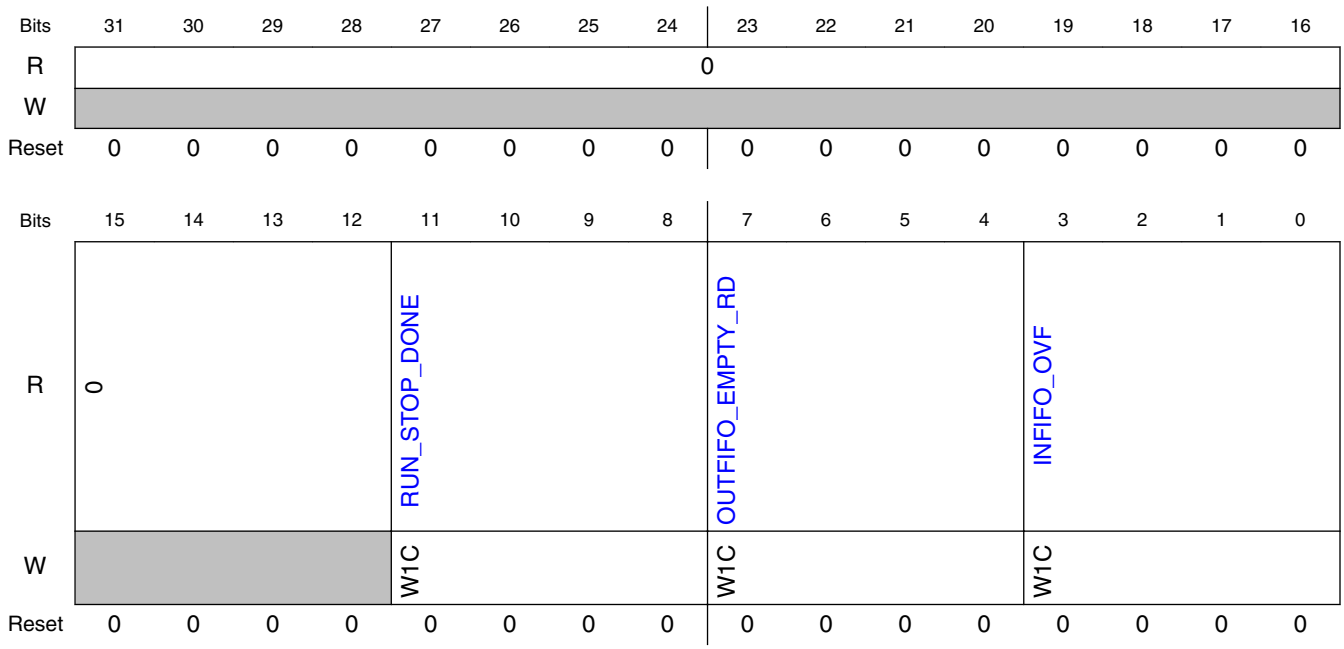
14.5.3.1.29 ASRC Interrupt Status Flags (ASRC_IRQ_FLAGS)

The ASRC_IRQ_FLAGS register provides the unmasked status flags for all interrupt sources within the ASRC. All status flags are implemented as W1C registers and should be cleared after servicing the interrupt.

14.5.3.1.29.1 Offset

Register	Offset
ASRC_IRQ_FLAGS	17Ch

14.5.3.1.29.2 Diagram



14.5.3.1.29.3 Fields

Field	Description
31-12 —	Reserved
11-8 RUN_STOP_DONE	ASRC RUN STOP DONE FLAG The RUN_STOP_DONE flag indicates the previous RUN_STOP operation has been completed. The ASRC context will be IDLE and all output samples will have been written to the output FIFO. Each bit in this register corresponds to a single context. Bit 8 corresponds to Context0, Bit 9 corresponds to Context1, ... An interrupt will be requested if one of the RUN_STOP_DONE flags are set and the respective RUN_STOP_DONE_MASK bit is cleared. 0000 - No RUN_STOP operations have been completed. 0001 - The RUN_STOP operation for Context 0 has completed. 0010 - The RUN_STOP operation for Context 1 has completed. 0011-1110 - The RUN_STOP operation has completed for any context with a 1'b1 bit field. 1111 - The RUN_STOP operation has completed for Context 0 to 3.
7-4 OUTFIFO_EMPTY_RD	ASRC Output FIFO Empty Read Flag The OUTFIFO_EMPTY_RD flags are triggered if a read is requested from an empty output FIFO. In this case, the return data from the read will be invalid. Note, the ASRC tracks conversions on a sample group granularity. This flag triggers if the sample group being accessed has not been completely written to the FIFO prior to a read being requested. Each bit in this register corresponds to a single context. Bit 4 corresponds to Context0, Bit 5 corresponds to Context1, ...

Table continues on the next page...

Field	Description
	An interrupt will be requested if one of the OUTFIFO_EMPTY_RD flags are set and the respective OUTFIFO_EMPTY_RD_MASK bit is cleared. 0000 - No reads have been requested from an empty ASRC_RDFIFO. 0001 - A read has been requested from ASRC_RDFIFO0 when it was empty. 0010 - A read has been requested from ASRC_RDFIFO1 when it was empty. 0011-1110 - A read has been requested from ASRC_RDFIFOn when it was empty. n = any bit position with a 0b1. 1111 - A read has been requested from ASRC_RDFIFO0, ASRC_RDFIFO1, ASRC_RDFIFO2, and ASRC_RDFIFO3 while empty.
3-0 INFIFO_OVF	ASRC Input FIFO Overflow Flag The INFIFO_OVF flags are triggered if an input FIFO has overflown. This condition may occur if writes are issued to the ASRC_WRFIFO registers faster then the input samples can be processed by the ASRC internal logic or the ASRC_RDFIFO consumer. Each bit in this register corresponds to a single context. Bit 0 corresponds to Context0, Bit 1 corresponds to Context1, ... An interrupt will be requested if one of the INFIFO_OVF flags are set and the respective INFIFO_OVF_MASK bit is cleared. 0000 - No INFIFO_OVF errors have been recorded. 0001 - The ASRC_WRFIFO0 has overflown. 0010 - The ASRC_WRFIFO1 has overflown. 0011-1110 - The ASRC_WRFIFOn has overflown. Where n = any bit position set to 0b1. 1111 - ASRC_WRFIFO0, ASRC_WRFIFO1, ASRC_WRFIFO2, and ASRC_WRFIFO3 have overflown.

14.5.3.1.30 ASRC Channel Status 0 (ASRC_CHANNEL_STATUS_00 - ASRC_CHANNEL_STATUS_03)

The ASRC Channel Status registers are used to specify the channel status field used when operating with ASRC_CTX_CTRL[IEC_EN] = 0b1. The channel status registers will be shifted out 1 bit per frame starting from the LSB of ASRC_CHANNEL_STATUS_0 to the MSB of ASRC_CHANNEL_STATUS_5. Note, there is one ASRC_CHANNEL_STATUS_x register per context in the design. ASRC_CHANNEL_STATUS_00 corresponds to context 0, ASRC_CHANNEL_STATUS_01 corresponds to context 1, and so in in this format.

14.5.3.1.30.1 Offset

Register	Offset
ASRC_CHANNEL_STATUS_00	180h
ASRC_CHANNEL_STATUS_01	184h

Table continues on the next page...

Register	Offset
ASRC_CHANNEL_STATUS_02	188h
ASRC_CHANNEL_STATUS_03	18Ch

14.5.3.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CHN_STAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CHN_STAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.30.3 Fields

Field	Description
31-0 CHN_STAT	Channel Status Data This register contains the channel status information which will be shifted out to the C data field defined in an IEC60958 sample. Data will be shifted out from LSB to MSB iterating from ASRC_CHANNEL_STATUS_0 to ASRC_CHANNEL_STATUS_5. Note, the channel status specified for an entire frame is the concatenation of {ASRC_CHANNEL_STATUS_5,ASRC_CHANNEL_STATUS_4,ASRC_CHANNEL_STATUS_3, ASRC_CHANNEL_STATUS_2,ASRC_CHANNEL_STATUS_1,ASRC_CHANNEL_STATUS_0}.

14.5.3.1.31 ASRC Channel Status 1 (ASRC_CHANNEL_STATUS_10 - ASRC_CHANNEL_STATUS_13)

The ASRC Channel Status registers are used to specify the channel status field used when operating with ASRC_CTX_CTRL[IEC_EN] = 0b1. The channel status registers will be shifted out 1 bit per frame starting from the LSB of ASRC_CHANNEL_STATUS_0 to the MSB of ASRC_CHANNEL_STATUS_5.

14.5.3.1.31.1 Offset

Register	Offset
ASRC_CHANNEL_STATUS_10	190h
ASRC_CHANNEL_STATUS_11	194h
ASRC_CHANNEL_STATUS_12	198h
ASRC_CHANNEL_STATUS_13	19Ch

14.5.3.1.31.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CHN_STAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CHN_STAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.31.3 Fields

Field	Description
31-0 CHN_STAT	Channel Status Data This register contains the channel status information which will be shifted out to the C data field defined in an IEC60958 sample. Data will be shifted out from LSB to MSB iterating from ASRC_CHANNEL_STATUS_0 to ASRC_CHANNEL_STATUS_5. Note, the channel status specified for an entire frame is the concatenation of {ASRC_CHANNEL_STATUS_5,ASRC_CHANNEL_STATUS_4,ASRC_CHANNEL_STATUS_3, ASRC_CHANNEL_STATUS_2,ASRC_CHANNEL_STATUS_1,ASRC_CHANNEL_STATUS_0}.

14.5.3.1.32 ASRC Channel Status 2 (ASRC_CHANNEL_STATUS_20 - ASRC_CHANNEL_STATUS_23)

The ASRC Channel Status registers are used to specify the channel status field used when operating with ASRC_CTX_CTRL[IEC_EN] = 0b1. The channel status registers will be shifted out 1 bit per frame starting from the LSB of ASRC_CHANNEL_STATUS_0 to the MSB of ASRC_CHANNEL_STATUS_5.

14.5.3.1.32.1 Offset

Register	Offset
ASRC_CHANNEL_STATUS_20	1A0h
ASRC_CHANNEL_STATUS_21	1A4h
ASRC_CHANNEL_STATUS_22	1A8h
ASRC_CHANNEL_STATUS_23	1ACh

14.5.3.1.32.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CHN_STAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CHN_STAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.32.3 Fields

Field	Description
31-0 CHN_STAT	Channel Status Data This register contains the channel status information which will be shifted out to the C data field defined in an IEC60958 sample. Data will be shifted out from LSB to MSB iterating from ASRC_CHANNEL_STATUS_0 to ASRC_CHANNEL_STATUS_5. Note, the channel status specified for an entire frame is the concatenation of {ASRC_CHANNEL_STATUS_5,ASRC_CHANNEL_STATUS_4,ASRC_CHANNEL_STATUS_3, ASRC_CHANNEL_STATUS_2,ASRC_CHANNEL_STATUS_1,ASRC_CHANNEL_STATUS_0}.

14.5.3.1.33 ASRC Channel Status 3 (ASRC_CHANNEL_STATUS_30 - ASRC_CHANNEL_STATUS_33)

The ASRC Channel Status registers are used to specify the channel status field used when operating with ASRC_CTX_CTRL[IEC_EN] = 0b1. The channel status registers will be

shifted out 1 bit per frame starting from the LSB of ASRC_CHANNEL_STATUS_0 to the MSB of ASRC_CHANNEL_STATUS_5.

14.5.3.1.33.1 Offset

Register	Offset
ASRC_CHANNEL_STATUS_30	1B0h
ASRC_CHANNEL_STATUS_31	1B4h
ASRC_CHANNEL_STATUS_32	1B8h
ASRC_CHANNEL_STATUS_33	1BCh

14.5.3.1.33.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CHN_STAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CHN_STAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.33.3 Fields

Field	Description
31-0 CHN_STAT	Channel Status Data This register contains the channel status information which will be shifted out to the C data field defined in an IEC60958 sample. Data will be shifted out from LSB to MSB iterating from ASRC_CHANNEL_STATUS_0 to ASRC_CHANNEL_STATUS_5. Note, the channel status specified for an entire frame is the concatenation of {ASRC_CHANNEL_STATUS_5,ASRC_CHANNEL_STATUS_4,ASRC_CHANNEL_STATUS_3, ASRC_CHANNEL_STATUS_2,ASRC_CHANNEL_STATUS_1,ASRC_CHANNEL_STATUS_0}.

14.5.3.1.34 ASRC Channel Status 4 (ASRC_CHANNEL_STATUS_40 - ASRC_CHANNEL_STATUS_43)

The ASRC Channel Status registers are used to specify the channel status field used when operating with ASRC_CTX_CTRL[IEC_EN] = 0b1. The channel status registers will be shifted out 1 bit per frame starting from the LSB of ASRC_CHANNEL_STATUS_0 to the MSB of ASRC_CHANNEL_STATUS_5.

14.5.3.1.34.1 Offset

Register	Offset
ASRC_CHANNEL_STATUS_40	1C0h
ASRC_CHANNEL_STATUS_41	1C4h
ASRC_CHANNEL_STATUS_42	1C8h
ASRC_CHANNEL_STATUS_43	1CCh

14.5.3.1.34.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CHN_STAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CHN_STAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.34.3 Fields

Field	Description
31-0 CHN_STAT	Channel Status Data This register contains the channel status information which will be shifted out to the C data field defined in an IEC60958 sample. Data will be shifted out from LSB to MSB iterating from ASRC_CHANNEL_STATUS_0 to ASRC_CHANNEL_STATUS_5. Note, the channel status specified for an entire frame is the concatenation of {ASRC_CHANNEL_STATUS_5,ASRC_CHANNEL_STATUS_4,ASRC_CHANNEL_STATUS_3, ASRC_CHANNEL_STATUS_2,ASRC_CHANNEL_STATUS_1,ASRC_CHANNEL_STATUS_0}.

14.5.3.1.35 ASRC Channel Status 5 (ASRC_CHANNEL_STATUS_50 - ASRC_CHANNEL_STATUS_53)

The ASRC Channel Status registers are used to specify the channel status field used when operating with ASRC_CTX_CTRL[IEC_EN] = 0b1. The channel status registers will be shifted out 1 bit per frame starting from the LSB of ASRC_CHANNEL_STATUS_0 to the MSB of ASRC_CHANNEL_STATUS_5.

14.5.3.1.35.1 Offset

Register	Offset
ASRC_CHANNEL_STATUS_50	1D0h
ASRC_CHANNEL_STATUS_51	1D4h
ASRC_CHANNEL_STATUS_52	1D8h
ASRC_CHANNEL_STATUS_53	1DCh

14.5.3.1.35.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CHN_STAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CHN_STAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.5.3.1.35.3 Fields

Field	Description
31-0 CHN_STAT	Channel Status Data This register contains the channel status information which will be shifted out to the C data field defined in an IEC60958 sample. Data will be shifted out from LSB to MSB iterating from ASRC_CHANNEL_STATUS_0 to ASRC_CHANNEL_STATUS_5. Note, the channel status specified for an entire frame is the concatenation of {ASRC_CHANNEL_STATUS_5,ASRC_CHANNEL_STATUS_4,ASRC_CHANNEL_STATUS_3, ASRC_CHANNEL_STATUS_2,ASRC_CHANNEL_STATUS_1,ASRC_CHANNEL_STATUS_0}.

14.6 Enhanced Audio Return Channel (eARC)

14.6.1 Overview

The Audio XCVR module is compliant to the HDMI Specification (version 2.1, dated November 13, 2017). This module supports eARC, ARC and SPDIF modes.

14.6.1.1 Block diagram

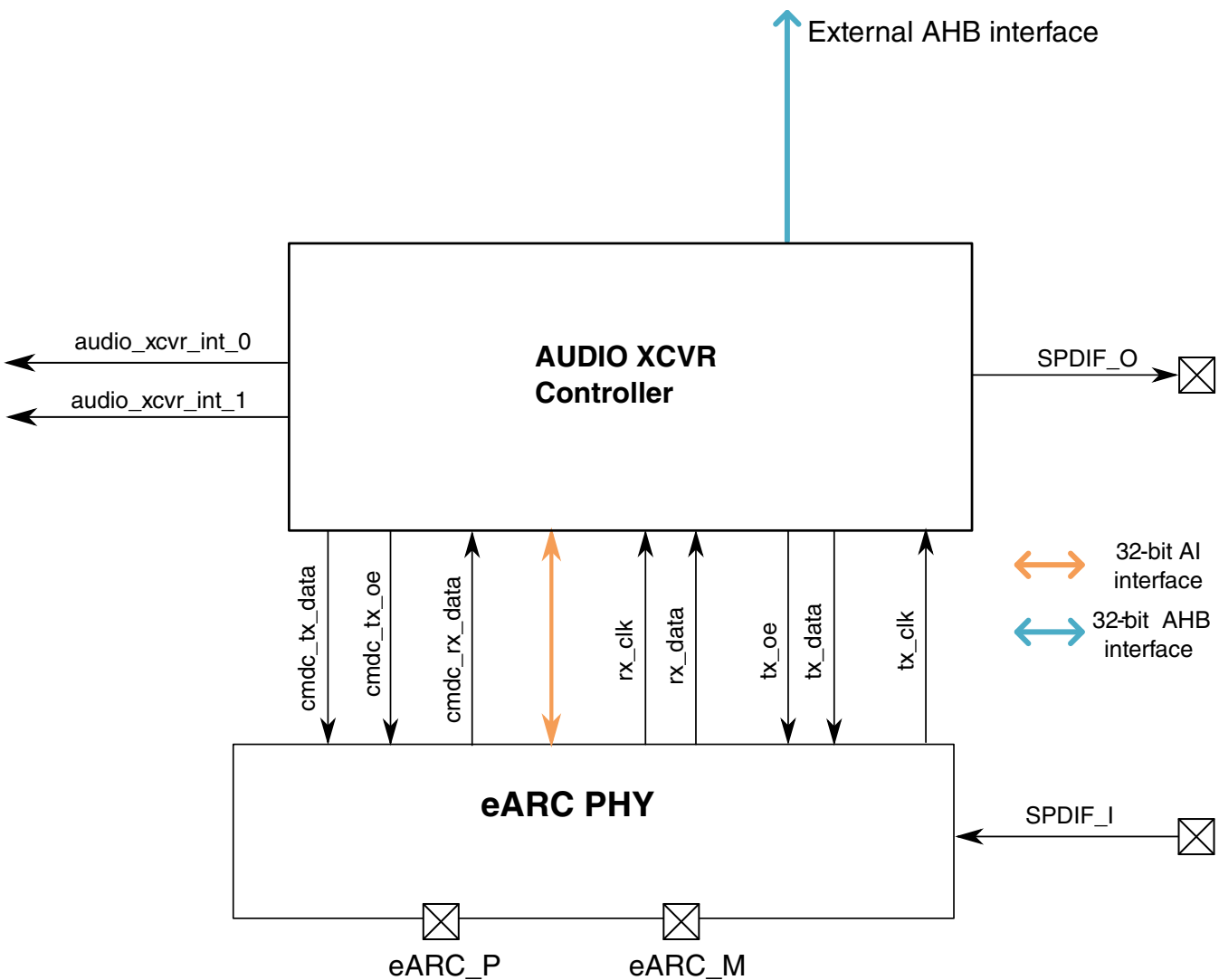


Figure 14-41. EARC block diagram

The main building blocks are:

1. Audio XCVR PHY: Used to receive eARC, SPDIF and ARC data from the +/- eARC pins and from the SPDIF input.

2. Audio XCVR Controller : Used to process the received audio data or the audio data that needs to be transmitted (SPDIF). It also receives and transmits the Common mode data used for establishing an eARC channel.

14.6.1.2 Features

The Audio XCVR module can operate in one of the following modes:

- eARC Receiver:
  - Full support for the eARC specification including all defined audio modes
  - No support for content protection
  - Support for error detection and recovery of audio
- ARC Receiver:
  - Support for up to 192kHz sample rate
  - Support for CEC for ARC backwards compatibility. DDC will be handled by the HDMI TX SS
  - CEC provides a potential system wake-up source
- SPDIF Receiver:
  - Support for 32kHz - 192kHz sample rate
- SPDIF Transmitter:
  - Support for 32kHz - 192kHz data tx rate

14.6.2 Functional description

The following sections describe functional details of the EARC module.

14.6.2.1 Audio data format support

All eARC/ARC/SPDIF message formats use IEC60958 framing. Data is stored in the FIFO in Big endian in 2 possible formats :

1. IEC60958 data that does not need ECC protection.
   - 31:28 - Preamble. For TX, the preamble can be added by HW or stored in the FIFO with the correct value. For RX, the 3rd bit will be replaced by MUTE and 4th bit will be replaced by Parity error if either one is detected.

- 27:4 - TX data
- 3:0 - V,U,C and P. For TX, these bits can be added by HW or stored in the FIFO with the correct value.

Each sub-frame can be stored in memory in the reverse order, bit 31 is stored in position 0 and so on using register bit : FRM_FMT in TX_DATAPATH_CTRL and STORE_FMT in RX_DATAPATH_CTRL.

2. IEC61937 data that requires ECC protection
   - 31:28 - Preamble. For TX, the preamble can be added by HW or stored in the FIFO with the correct value. For RX, the 3rd bit will be replaced by MUTE and 4th bit will be replaced by Parity error.
   - 27:12 - 16 bits of data. (MSB is stored in bit 27)
   - 11:4 - Not used. 8bit's 0.
   - 3:0 - V,U,C and P. For TX, these bits can be added by HW or stored in the FIFO with the correct value.

   Each sub-frame can be stored in memory in the reverse order, bit 31 is stored in position 0 and so on using register bit : FRM_FMT in TX_DATAPATH_CTRL and STORE_FMT in RX_dATAPATH_CTRL.

eARC supports multiple audio formats within the IEC60958-1 framing structure. SPDIF supports L-PCM and compressed audio within the IEC60958-1 framing structure.

Table 14-11. Supported data formats

Interface	Type	Channels	Payload Size	Frame Format
eARC	2-ch LPCM	2	24	IEC60958
	Multi-ch LPCM	2	24	
	Multi-ch LPCM	8	24	
	Multi-ch LPCM	16	24	
	Multi-ch LPCM	32	24	
	One-bit	6	24	
	One-bit	12	24	
	IEC61937 Compressed	Layout A	16 (+8 ECC)	
	IEC61937 Compressed	Layout B	16 (+8 ECC)	
SPDIF	2-ch LPCM	2	16, 20, or 24	IEC60958
	IEC61937 Compressed	Layout A	16 (+8 ECC)	

14.6.2.2 eARC Data Rates

EARC supports high audio bit rates

- Uncompressed audio at 2.048Mbps for 2-ch 32/16

- Uncompressed audio at 36.864Mbps for 8-ch 192/24
- Compressed audio up to 24.576Mbps

In all cases the maximum is 49.152Mbps, or a 768kHz IEC60958-1 Frame rate. Biphase-mark coding and FEM doubles the DMAC frequency to a maximum of 98.304Mbps.

NOTE

- The eARC specification does not specifically identify supported Fs, except for 32, 44.1 and 48kHz 2-ch LPCM.
- The table uses permissible Fs values for HDMI2.1
- L-PCM audio sample rates are maximum 192kHz
- eARC cannot meet the theoretical HDMI support for one-bit (up to 192kHz * 64) due to the 6 or 12-ch grouping

Table 14-12. Supported eARC Data Rates

Type	Channels	Fs (kHz)	Data (Mbps)
2-ch LPCM	2	32	2.048
		44.1	2.8224
		48	3.072
		64	4.096
		88.2	5.6448
		96	6.144
		128	8.192
		176.4	11.2896
		192	12.288
Multi-channel LPCM	8	32	8.192
		44.1	11.2896
		48	12.288
		64	16.384
		88.2	22.5792
		96	24.576
		128	32.768
		176.4	45.1584
		192	49.152
	16	32	16.384
		44.1	22.5792
		48	24.576
		64	32.768
		176.4	45.1584
		192	49.152
	32	32	32.768
		44.1	45.1584

Table continues on the next page...

Table 14-12. Supported eARC Data Rates (continued)

Type	Channels	Fs (kHz)	Data (Mbps)
		48	49.152
One-bit	6	32	16.384
		44.1	22.5792
		48	24.576
		64	32.768
		88.2	45.1584
		96	49.152
	12	32	32.768
		44.1	45.1584
		48	49.152
IEC61937 Compressed	Layout A	32	2.048
		44.1	2.8224
		48	3.072
		64	4.096
		88.2	5.6448
		96	6.144
		128	8.192
		176.4	11.2896
		192	12.288
		256	16.384
		352.8	22.5792
		384	24.576
		512	32.768
		705.6	45.1584
		768	49.152
IEC61937 Compressed	Layout B	32	8.192
		44.1	11.2896
		48	12.288
		64	16.384
		88.2	22.5792
		96	24.576
		128	32.786
		176.4	45.1584
		192	49.152

14.6.2.3 Functional Operating mode

The AUDIO transceiver hardware can be used without leveraging the M0+ or the code for it. However, the intended use is to use the M0+ to set up the HW and use it to process the data.

The interface is interrupt driven and when the corresponding interrupt is received, FW will release HW from reset and set up the PHY.

The AUDIO XCVR can be operated in eARC RX, ARC RX or SPDIF TX and/or RX. In order to set up the Controller in the various modes, set the following interrupts.

- eARC RX mode: In order to set up the AUDIO_XCVR in eARC_RX mode, load the M0 code and release it from reset. Once HPD is received, FW will enable the PHY and set up the PLL as needed.
- ARC RX mode: In order to set up the PHY and controller in ARC RX mode, assert the ARC single ended mode (IER[23]) or ARC common mode (IER[22]) interrupt. When either of these interrupts are asserted, FW will program the PHY and Controller controller registers.
- SPDIF TX mode: This mode is the same as eARC TX mode. This mode can be set up by asserting IER[21]. Host core will need to program the PHY PLL to output a 2*TX data rate clock.
- SPDIF RX mode: This mode can be set up using IER[20]. This will set up the PHY and Controller registers as needed. SPDIF TX and RX can be operated simultaneously. If this is desired, use IER[19]. The HDMI TX SS PLL will still need to be set up by the host core.

14.6.2.4 Low power modes

The M0+ core can be put on hold / in reset, reducing it's power consumption. Each of the data paths, DATA TX, DATA RX, and CMDC in eARC RX can be put in reset when not being used. This will reduce power consumption from the corresponding data paths.

14.6.2.5 Test modes

The main test modes that are supported by the Audio XCVR controller are the loopback modes for Audio transceiver Data path and eARC CMDC. Loopback tests for all sets of HW can be run separately or a combination of DMAC with CMDC HW. The loopback test is slightly different for CMDC compared to DMAC.

14.6.2.5.1 CMDC Loopback Mode

The below steps indicate how the CMDC can be enabled in loopback mode. Special PHY programming is also required and is listed below.

Start out by enabling the PHY. This is done by programming the PHY using the AI interface. Refer to AI interface for PHY and PLL programming to see how this is done. The PHY can be enabled by first enabling the bandgap in the PLL eARC.PLL - (0x54) = 0x00000001, and then enabling the PHY eARC.PHY (0x4) = 0x00000001. The PHY will take 2 ms to start up. In the meanwhile, program the PLL and the controller using the below steps:

- Release CMDC RX HW from reset. eARC.CTRL (0x18) = 0x60000000 for RX HW
- Set CMDC in loopback mode: EXT_CTRL (0xC4) = 0x80000000 for RX HW

Once the 2 ms are up, program the PHY to TX and RX data simultaneously, and enable data tx in the controller. Read CMDC RX data and compare against data transmitted. If they match, the test passes. Repeat the below steps with different TX_DATA to check for different values

1. Program the PHY to TX and RX CMDC data: eARC.PHY (0x4) = 0x00000012
2. Write CMDC data to be transmit eARC.CTRL (0xE0) = {0x83F0,TX_DATA} for RX datapath and enable the transmit of CMDC data in the controller.
3. Wait for 60 us and read CMDC RX data eARC.CTRL (0xF0) for RX.
4. If the lower 16 bits from step 1 match the received data, the test passes.

Repeat the above steps for a few more patterns. Suggested patterns are: TX_DATA = 0x0000 / 0xFFFF / 0xAAAA

14.6.2.5.2 DMAC loopback mode

DMAC loopback does not need any special modes to be set. The DMAC TX and RX datapaths can be set up as required. The registers don't necessarily need to be set and can be used in the out of reset mode, where the transmitter will insert parity, and preamble before transmitting data. The PHY needs to be programmed to enable receiving data and the HDMI_TX_AVPLL needs to be programmed to send out the max transmit frequency.

Start out by enabling the PHY. This is done by programming the PHY using the AI interface. Refer to AI interface for PHY and PLL programming and follow the steps. The PHY can be enabled by first enabling the bandgap in the PLL eARC.PLL - (0x54) = 0x00000001, and then enabling the PHY eARC.PHY (0x4) = 0x00000001. The PHY will take 2 ms to start up. In the meanwhile, program the PLL and the controller using the below steps:

- Set up the PHY PLL to output Clock 1. Refer to section: Application Information: Programming the PLL

- Set up PLL in HDMI_TX SS to output a frequency of 491.52MHz. The steps for programming the PLL are the same as above. Refer to the HDMI TX SS on registers to write to, to program the PLL. Follow MFI, MFN and MFD values for DMAC TX in PLL programming sequence.
- Do not enable the output clocks until the MUX has been set to select the right clock source
- Once both the PLLs have locked, set the TX clock source in the PHY to use the output of HDMI SS AVPLL eARC.PHY (0x4) = 0x02000000
- Enable the output clocks from the PLL.

Once the 2 ms are up, program the PHY to TX and RX DMAC data simultaneously, (eARC.PHY(0x4) = 0x00000012) and enable data tx in the controller. Once RX data is available, compare against transmit data.

1. Release DMAC TX and RX datapath from reset. eARC.CTRL (0x18) = 0x18000000.
2. Enable DMAC TX and RX datapath in the PHY. eARC.PHY (0x4) = 0x00000012 and eARC.PHY(0x74) = 0x00004000.
3. Load the transmit FIFO with transmit data, by writing to location eARC.CTRL (0xE00). This location will accept upto 128 entries of transmit data. If you wish to transmit more data, dma_wr_req can be used to add more data to the transmit FIFO.
4. Once the data is written to the transmit FIFO, start transmit of data using eARC.CTRL (0x224) = 0x00004000
5. Once 64 entries are available in the receive FIFO, SDMA_RD_REQ will be asserted and the received data can be checked against the transmitted data, by reading it from address 12'hC00. If fewer entries are transmitted, either wait the required amount of time and read entries from the RX FIFO.

At a minimum, test at least 64 entries to make sure enough data is sent and received on the interface.

14.6.2.6 Operating modes

The AUDIO XCVR can be operated in eARC RX, ARC RX or SPDIF TX and/or RX.

In order to set up the Controller in the various modes, set the following interrupts.

- eARC RX mode: In order to set up the AUDIO_XCVR in eARC_RX mode, load the M0 code and release it from reset. Once HPD is received, FW will enable the PHY and set up the PLL as needed.
- ARC RX mode: In order to set up the PHY and controller in ARC RX mode, assert the ARC single ended mode (IER[23]) or ARC common mode (IER[22]) interrupt. When either of these interrupts are asserted, FW will program the PHY and Controller controller registers.

- SPDIF TX mode: Host core will need to program the PHY PLL to output a 2*TX bit rate clock or 10*TX bit rate clock based on selection programmed in TX_DATAPATH_CTRL[10].
- SPDIF RX mode: This mode can be set up using IER[20]. This will set up the PHY and Controller registers as needed. SPDIF TX and RX can be operated simultaneously. If this is desired, use IER[19]. The HDMI TX SS PLL will still need to be set up by the host core.

14.6.2.7 Interrupts

This section describes all the interrupts that the AUDIO XCVR module generates.

1. CMDC interrupts:
   - CMDC_RESP_TO_ERR - This interrupt is asserted if the CMDC response was not sent out in the programmed time frames. Details about which timer expired can be found from EXT_STATUS register bits [26:23]
   - HB_STATE_CHG - Heartbeat state changed. (If heartbeat was first established or if we lose it after it was established)
   - CMDC_STATUS_UPDATE - Set if on read of CMDC STAT reg, status change bit was set (Can be used to read the latency)
   - OHPD - Set if output HPD is set.
2. DMAC Interrupts:
   - RX_NEW_CH_STAT - New channel status was received (Set by M0+)
   - RX_NEW_USR_DATA - New user data was received which passed CRC check. (Set by M0+). EXT_STATUS register holds additional information if it was UD type 4, 5 or 6
   - MUTE_DET - HW Mute detect
   - ECC_ERR - Asserted when an uncorrectable ECC error is received in compressed data mode
   - PREAMBLE_MISMATCH - Asserted when there is an unexpected preamble received / when a preamble was expected but one was not received.
   - FIFO_OFLOW_UFLOW_ERR - This interrupt is asserted if the RX / TX FIFO overflows / underflows.
   - DMAC_NO_DATA_REC - Set after valid frames were received, but then valid data was not received for over 1us.
   - FMT_CHG_DET - No data was received for ~60 ms. Format change is detected.

- PREAMBLE_MATCH_INT - This interrupt is asserted every time a PA PB / CD match is received in compressed data. The information for packet count when this was received and which match occurred can be read from the 4KB data memory location.
- UNEXP_PRE_REC - This interrupt is set if an unexpected preamble is received. If this is received, Host software should reset the RX datapath and wait for the next B preamble to be received.

3. DMA Interrupts:
   - DMA_RD_REQ - Asserted when RX FIFO entries reaches the programmed threshold. (Default = 64 entries, half full)
   - DMA_WR_REQ - Asserted when TX FIFO entries reaches the programmed threshold. (Default = 64 entries, half empty)

14.6.3 Application information

This section describes the applications supported by the Audio XCVR and how to program the registers to support these features.

14.6.3.1 Memory Access

There are 4 sets of memories that can be accessed.

- 16KB of Program Memory
- 4 KB of Data memory
- Register access space and
- TX / RX Data access space.

Since externally only 4 KB of memory space is allocated to the eARC controller, a paging mechanism implemented in the eARC Controller. The first two memories can be accessed as a chunk of 2KB each, while the registers have their own KB of space, and so does the TX/RX data space.

Memory region access can be paged through using the control register EXT_CTRL[19:16]. Bit[19] Separates 16KB memory(= 0) from 4KB memory(= 1). The rest of the bits are used to page through the memories. 16KB memory has 8 pages, while the 2KB memory has 2 pages.

Register access can be achieved when the address to the eARC controller is : 12'b10xx_xxxx_xxxx , where x can be any value.

RX data can be accessed by reading from 12'h110x_xxxx_xxxx. Since the data is read from a FIFO, the actual address does not matter.

TX data can be accessed by reading from 12'h111x_xxxx_xxxx. Since the data is written to a FIFO, the actual address does not matter.

14.6.3.2 Data RAM organization

The 4KB data RAM is allocated for external host access. It stores a range of information listed below:

- CMDC status and latency registers
- Channel status and user data status information
- Double buffered AUDIO channel status
- Double buffered user data information for ACP, ISRC1 and ISRC2 type
- Capability data structure for CMDC
- PA PB preamble match circular FIFO
- Logging

Table 14-13. Data RAM organization

Function	Address	Size (Bytes)	Comment
CMDC Control / Status	0x0	8	Includes CMDC TX and RX status registers along with the Latency and latency request register. Each register occupies a byte in memory. The first 4 bytes are for CMDC in eARC RX mode and the next 4 bytes are for CMDC in eARC TX mode.
Channel Status Control	0x20	8	These two words hold the control register for channel status buffers. These control registers should be set to 0 by host once read. They will only be updated by M0 if cleared. Buffer 0 is always checked first. Only if buffer 0 has not been cleared will buffer 1 be updated. If both buffers are full, the new status will be dropped and an interrupt indicating new channel status value will not be asserted. The non-zero value has 2 fields: • [31] - Update flag • [30:0] - Packet count after the channel status was processed.
User Data (ACP) Control	0x28	8	These two words hold the control register for ACP user data. These control registers should be set to 0 by host once read. They will only be updated by M0 if cleared. Buffer 0 is always checked first. Only if buffer 0 has not been cleared will buffer 1 be updated. If both buffers are full, the new status will be dropped and an interrupt indicating new user data value will not be asserted. The non-zero value has 2 fields: • [31] - Update flag • [30:0] - Packet count after the channel status was processed.
User Data (ISRC1) Control	0x30	8	These two words hold the control register for ISRC1 user data. These control registers should be set to 0 by host once read. They will only be updated by M0 if cleared. Buffer 0 is

Table continues on the next page...

Table 14-13. Data RAM organization (continued)

Function	Address	Size (Bytes)	Comment
			always checked first. Only if buffer 0 has not been cleared will buffer 1 be updated. If both buffers are full, the new status will be dropped and an interrupt indicating new user data value will not be asserted. The non-zero value has 2 fields: • [31] - Update flag • [30:0] - Packet count after the channel status was processed.
User Data (ISRC2) Control	0x38	8	These two words hold the control register for ISRC2 user data. These control registers should be set to 0 by host once read. They will only be updated by M0 if cleared. Buffer 0 is always checked first. Only if buffer 0 has not been cleared will buffer 1 be updated. If both buffers are full, the new status will be dropped and an interrupt indicating new user data value will not be asserted. The non-zero value has 2 fields: • [31] - Update flag • [30:0] - Packet count after the channel status was processed.
Channel Status Buffers	0x80	64	The channel status buffers are aligned to 32-bytes. The last 8 bytes are empty. Each buffer corresponds with the associated channel status control field. When a new block of data has a channel status update, the buffer that is empty is updated and an interrupt is asserted to the host. Buffer 0 is always checked first.
User data (ACP) Buffers	0xC0	64	The user data buffers are aligned to 32 bytes. IU1,2,3 and byte 1 of the remainder of the IUs are not stored. Each buffer corresponds with the associated user data control field. When a new User data message is received, and it passes CRC checks, the buffer that is empty is updated and an interrupt is asserted to the host. Buffer 0 is always checked first.
User data (ISRC1) Buffers	0x100	64	The user data buffers are aligned to 32 bytes. IU1,2,3 and byte 1 of the remainder of the IUs are not stored. Each buffer corresponds with the associated user data control field. When a new User data message is received, and it passes CRC checks, the buffer that is empty is updated and an interrupt is asserted to the host. Buffer 0 is always checked first.
User data (ISRC2) Buffers	0x140	64	The user data buffers are aligned to 32 bytes. IU1,2,3 and byte 1 of the remainder of the IUs are not stored. Each buffer corresponds with the associated user data control field. When a new User data message is received, and it passes CRC checks, the buffer that is empty is updated and an interrupt is asserted to the host. Buffer 0 is always checked first.
Capabilities data structure	0x300	256	Capability data structure is maximum 256 bytes.

14.6.3.3 PLL Programming Sequence

The pll programmed frequency value will be: PLL frequency = 24 * I (Integer) + (N (Numerator) / D (Denominator))

PLL can be programmed using the below sequence. If programming the PLL for eARC PHY, the first step can be skipped as it is included with enabling the PHY.

1. Enable the bandgap: eARC.PLL (0x54) = 0x00000001
2. Wait for at least 50 us to allow for the bandgap to stabilize the voltage and current reference for the PLL. In the meanwhile, program the PLL I, N and D to desired values
   - Numerator can be programmed by writing to eARC.PLL (0x20)
   - Denominator can be programmed by writing to eARC.PLL (0x30)
   - Integer can be programmed by writing to eARC.PLL (0x4)[6:0]. In order to update this field, always clear the 7 bits first and then set them with the desired value.
3. Assert power up in the PLL eARC.PLL (0x4) = 0x00003000.
4. Wait for 25 us to allow the ring oscillator to settle down.
5. Disable hold ring by clearing the hold bit: eARC.PLL (0x8) = 0x00002000.
6. Wait for 75 us for the PLL to lock. In the mean while, set up the post dividers. eARC.PLL (0x40) bits [2:0] - for clk0, bits [6:4] - for clk1, and bits [10:8] - for clk2.
7. Once the PLL s have locked, make sure to select the right clock source in the PHY before enabling the PLL clocks using bits [26:24] in eARC.PLL (0x4) for clock 2,1 and 0 respectively.

Typical PLL programming values for different use cases:

- For TX Datapath - If only SPDIF TX function is desired, select Clock 0. Set the frequency to 10*(data TX rate) or 2*(data TX rate) for SPDIF TX. If Simultaneous SPDIF TX and RX is desired, use an alternate clock source of HDMI TX AVPLL / Audio SS PLL for SPDIF TX. Set eARC.CTRL(0x0) = 0x02000000 for HDMI TX SS and 0x04000000 for AUDIO SS PLL clock source.
- For DMAC RX / Loopback mode - Select Clock 1. The desired frequency is 1.3GHz. I = 0x36, N = 0x1, D = 0x6. Post divider value [6:4] = 0x0.
- For SPDIF RX - Select clock 2. The desired frequency is 175 MHz. I = 0x1D, N = 0x1, D = 0x6. post Divider value [10:8] = 0x2.

14.6.3.4 Common Mode Data Channel (CMDC)

CMDC is a bi-directional protocol that is used to establish the earc channel, transmit and receive capabilities data structure and exchange latency information. CMDC functions are different for source / sink modes. CMDC in source mode can be brought out of reset using EXT_CTRL.CMDC_RX_REST. The clock to 1us ratio needs to be programmed into the CLK_CTRL register. The default value is for a 400MHz clock. This value is used as a reference for various timers needed by CMDC protocol.

When CMDC is used as the source, host core should update the value in CAP_CHG every time it updates the capabilities data structure. If host core has an update in the RX latency value, it should update the value of STAT_CHG in 4KB memory to let the sink know there is a new value that should be read.

14.6.3.5 Audio Receiver Overview

The Audio RX datapath is held in reset until host core decides to release the HW out of reset to start receiving IEC60958 data. This feature helps save power when specific datapaths are not being used. The PHY transmits DMAC clock and DMAC data over to the controller. SPDIF and ARC data is muxed over on the same signals along with an enable strobe. The data sent over is Bipphase Mark Encoded (BME) and needs to be decoded into subframes.

CMDC interface sees the HPD asserted and responds by sending out the comma pattern. Once heartbeat is achieved, data can be received. There are no HW gates for this. As soon as the RX datapath is released from reset, it is capable of receiving data. The RX datapath can be controled by RX_DATAPATH_CTRL register. Configurations are listed below:

- FSM: The input data stream can be received when aligned to either the B preamble, M preamble, or the M/W preambles. By default, the input stream is set to sync to the B preamble. Change this register setting to align to M/W input preamble.
- COMP and PRC: Compressed data mode can either be detected by HW or enabled by SW. COMP bit selects between HW / SW control of PRC bit HW detection of compressed mode is selected. In this mode, compressed data is detected by matching the first 3 Channel status bits with CBITE and mask CBITM. Once a match is detected, PRC bit is set by HW. If SW mode is selected, SW can set PRC to enable ECC correction and de-interleaving of data frames once the corresponding channel status bits match compressed mode.
- STORE_FMT: Format to store the received data in memory. If this bit is 0, the first bit received in a sub frame is stored in bit position 31. Else, it is stored in bit position 0. The rest of the bits follow the first bit.
- EN_PARITY_CALC: This bit enables parity evaluation. If Parity evaluation fails, bit 4 of the preamble is replaced with 1.
- MUTE_MODE: Mute mode selects between HW detection of MUTE mode and FW enabling Mute. By default, HW mode is set. In this mode, mute is detected by looking at channel status of frame number programmed in MBIT. If SW control is enabled, mute is asserted using MUTE_CTRL. Preamble bit 1 is replaced by

- BLKS: This bit enables blocking audio data that is in compressed mode. Compressed data output can be blocked if Audio XCVR is connected to ASRC and processing L-PCM. If this bit is set, compressed data is blocked and RX audio data sent to the RX fifo is all 0's. Preamble and VUCP bits are preserved.
- LAYB: Layout B handler. This bit is set by HW when compressed data is indicated by PRC. When layout B is detected by parsing the Channel Status registers, only %4 frames are stored starting from the next block of data.
- PABS and DTS_CDS: Pa Pb and DTS CD 14 search can be enabled using these register bits. When enabled, compressed mode data is parsed for a frame match of the values programmed in DMAC_PRE_MATCH_VAL and DMAC_DTS_PRE_MATCH_VAL. The match reference values can be found in registers DMAC_PRE_MATCH_VAL and DMAC_DTS_PRE_MATCH_VAL respectively. When a match is detected, the packet count along with the match type is committed to a FIFO and can be read from DMAC_PRE_MATCH_OFFSET.
- Performance counter: The Audio Receiver path supports a performance/progress counter. This is essentially 2 counters, one running at the highest clock speed available to the block and the other at the audio bit rate. The high speed counter is latched twice, once on every bit clock transition and again when the bit clock counter is read to guarantee a synchronized read.
- Packet counter: The packet counter keeps track of all the sub-frames written to the output FIFO. It is a rolling counter used to keep a reference of the number of frames received.

Channel status and user data buffers are stored per block. Once the data is processed, an interrupt to the host core is asserted if the Channel status does not match the last channel status value. For user data, the type of user data is compared against the previously received user data of that type. If a difference is noticed, CRC is calculated and if the user data chunk of 47 IUs passes CRC, the host core is interrupted. The host core can read this channel status / user data from the 4KB memory with the offsets listed below.

- Channel Status : Offset 0x80
- User Data type 4 : Offset 0xC0
- User Data type 5 : Offset 0x100
- User Data type 6 : Offset 0x140

Channel status and user data values can be reset by FW using RX_DATAPATH_CTRL.CSR and RX_DATAPATH_CTRL.UDR.

14.6.3.6 Audio Transmitter Overview

The Audio transmitter can output data in the SPDIF format. In order to transmit data, the PLL Frequency for clk root 0, needs to be programmed to 5* data TX rate or 1* data TX rate, depending on the mode selected. If transmit datapath needs to be enabled, it should be released from reset using EXT_CTRL.TX_DPATH_RESET. If TX datapath is not released from reset, data entries should not be written to the TX FIFO. TX can now be enabled using TX_DATAPATH_CTRL.STRT_DATA_TX. Once this bit is set, HW will start transmitting data in the TX FIFO. If the FIFO does not have watermark number of entries, HW will assert a dma_wr_req. Care should be taken to make sure the FIFO does not underrun.

The Audio transmitter consists of the following features that can be enabled in TX_DATAPATH_CTRL register.

- EN_PREAMBLE: If this bit is set, preamble matching the frame count is inserted in the TX sub-frame. TX always starts with a B preamble.
- CS_MOD and UD_MOD: These register bits are used to modify the channel status and user data values programmed in the TX FIFO. The channel status and user data replacement values should be programmed in TX_CS_DATA_BITS and TX_USER_DATA_BITS buffers on a block boundary. These registers have an ACK value in UD_ACK and CS_ACK. FW should not write to the data buffers if the ACK bits are not 0. Once FW writes to the registers, the ACK bits should be set so HW can read the new values when it starts transmitting the next block of data. These bits will be cleared by HW when the value in TX_CS_DATA_BITS and TX_USER_DATA_BITS have been moved to a secondary register.
- VLD_MOD and FRM_VLD: These register bits are used to modify the valid field programmed in the TX FIFO. If VLD_MOD is set, Valid bit in the TX frames is replaced by the value in FRM_VLD.
- EN_PARITY: This bit is used to calculate the parity over the data and VUC bits and replace the parity bit of the frame.
- EN_ECC_INTER: If this bit is set, Error correction bits are inserted into the TX frame and then a pair of sub-frames are interleaved.
- TX_CLK_RATE: If this bit is set, output data will be generated by using a clock that is 2* BME bit rate. If this bit is not set, output data will be generated by using a clock that is 10* BME bit rate.
- FRM_FMT: This bit indicates if the preamble is stored in bits 3:0 or 31:28

14.6.4 Memory Map and register definition

This section includes the EARC module memory map and detailed descriptions of all registers.

14.6.4.1 EARC register descriptions

14.6.4.1.1 EARC memory map

eARC base address: 30CC_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version control register (VERSION)	32	RW	0000_0000
10	External control register (EXT_CTRL)	32	RW	F820_4040
14	External control register (EXT_CTRL_SET)	32	RW	F820_4040
18	External control register (EXT_CTRL_CLR)	32	RW	F820_4040
1C	External control register (EXT_CTRL_TOG)	32	RW	F820_4040
20	External Status register (EXT_STATUS)	32	RW	0000_0000
24	External Status register (EXT_STATUS_SET)	32	RW	0000_0000
28	External Status register (EXT_STATUS_CLR)	32	RW	0000_0000
2C	External Status register (EXT_STATUS_TOG)	32	RW	0000_0000
30	Interrupt enables for interrupt 0 (EXT_IER0)	32	RW	0000_0000
34	Interrupt enables for interrupt 0 (EXT_IER0_SET)	32	RW	0000_0000
38	Interrupt enables for interrupt 0 (EXT_IER0_CLR)	32	RW	0000_0000
3C	Interrupt enables for interrupt 0 (EXT_IER0_TOG)	32	RW	0000_0000
40	Interrupt enables for interrupt 1 (EXT_IER1)	32	RW	0000_0000
44	Interrupt enables for interrupt 1 (EXT_IER1_SET)	32	RW	0000_0000
48	Interrupt enables for interrupt 1 (EXT_IER1_CLR)	32	RW	0000_0000
4C	Interrupt enables for interrupt 1 (EXT_IER1_TOG)	32	RW	0000_0000
50	External Interrupt Status register (EXT_ISR)	32	RW	0000_0000
54	External Interrupt Status register (EXT_ISR_SET)	32	RW	0000_0000
58	External Interrupt Status register (EXT_ISR_CLR)	32	RW	0000_0000
5C	External Interrupt Status register (EXT_ISR_TOG)	32	RW	0000_0000
90	AI interface control register (PHY_AI_CTRL)	32	RW	0000_0000
94	AI interface control register (PHY_AI_CTRL_SET)	32	RW	0000_0000
98	AI interface control register (PHY_AI_CTRL_CLR)	32	RW	0000_0000
9C	AI interface control register (PHY_AI_CTRL_TOG)	32	RW	0000_0000
A0	AI interface WDATA register (PHY_AI_WDATA)	32	RW	0000_0000
A4	AI interface RDATA register (PHY_AI_RDATA)	32	RO	0000_0000
A8	Audio XCVR datapath status (DPATH_STATUS)	32	RO	0000_0000
C0	CMDC receiver control register (RX_CMDC_CTRL)	32	RW	0028_1B02
C4	CMDC receiver control register (RX_CMDC_CTRL_SET)	32	RW	0028_1B02
C8	CMDC receiver control register (RX_CMDC_CTRL_CLR)	32	RW	0028_1B02
CC	CMDC receiver control register (RX_CMDC_CTRL_TOG)	32	RW	0028_1B02
D0	eARC receiver CMDC status (RX_CMDC_STATUS)	32	RO	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
E0	CMDC transmit data register (RX_CMDC_TX_DATA)	32	RW	0000_0000
F0	CMDC receive data register (RX_CMDC_RX_DATA)	32	RO	0000_0000
180	Data path control register (RX_DATAPATH_CTRL)	32	RW	0000_2C89
184	Data path control register (RX_DATAPATH_CTRL_SET)	32	RW	0000_2C89
188	Data path control register (RX_DATAPATH_CTRL_CLR)	32	RW	0000_2C89
18C	Data path control register (RX_DATAPATH_CTRL_TOG)	32	RW	0000_2C89
190 - 1A4	Channel staus bits (RX_CS_DATA_BITS_0 - RX_CS_DATA_BITS_5)	32	RO	0000_0000
1A8 - 1BC	User data bits (RX_USER_DATA_BITS_0 - RX_USER_DATA_BITS_5)	32	RO	0000_0000
1C0	DMAC counter control register (RX_DPATH_CNTR_CTRL)	32	RW	0000_0000
1C4	DMAC counter control register (RX_DPATH_CNTR_CTRL_SET)	32	RW	0000_0000
1C8	DMAC counter control register (RX_DPATH_CNTR_CTRL_CLR)	32	RW	0000_0000
1CC	DMAC counter control register (RX_DPATH_CNTR_CTRL_TOG)	32	RW	0000_0000
1D0	Receive Datapath Timestamp Counter Register (RX_DPATH_TSCR)	32	RO	0000_0000
1D4	Receive Datapath Bit counter register (RX_DPATH_BCR)	32	RO	0000_0000
1D8	Receive datapath Bit count timestamp register. (RX_DPATH_BCTR)	32	RO	0000_0000
1DC	Receive datapath Bit read timestamp register. (RX_DPATH_BCRR)	32	RO	0000_0000
1E0	Preamble match value register (DMAC_PRE_MATCH_VAL)	32	RW	4E1F_F872
1E4	Preamble match value register (DMAC_PRE_MATCH_VAL_SET)	32	RW	4E1F_F872
1E8	Preamble match value register (DMAC_PRE_MATCH_VAL_CLR)	32	RW	4E1F_F872
1EC	Preamble match value register (DMAC_PRE_MATCH_VAL_TOG)	32	RW	4E1F_F872
1F0	Preamble match value register (DMAC_DTS_PRE_MATCH_VAL)	32	RW	1387_FE1C
1F4	Preamble match value register (DMAC_DTS_PRE_MATCH_VAL_SET)	32	RW	1387_FE1C
1F8	Preamble match value register (DMAC_DTS_PRE_MATCH_VAL_CLR)	32	RW	1387_FE1C
1FC	Preamble match value register (DMAC_DTS_PRE_MATCH_VAL_TOG)	32	RW	1387_FE1C
200	Error count for IEC60958-1 Block Synchronization. (RX_DPATH_PRE_ERR)	32	RW	0000_0000
204	Parity Error count for IEC60958-1 Blocks. (RX_DPATH_PARITY_ERR)	32	RW	0000_0000
210	Receive Data packet count. (RX_DPATH_PKT_CNT)	32	RO	0000_0000
214	Receive Data packet Corrected error count. (RX_DPATH_ONE_BIT_ERR_CNT)	32	RO	0000_0000
218	Preamble match offset value register (DMAC_PRE_MATCH_OFFSET)	32	RO	0000_0000
220	Transmit Data path control register (TX_DATAPATH_CTRL)	32	RW	0000_0000
224	Transmit Data path control register (TX_DATAPATH_CTRL_SET)	32	RW	0000_0000
228	Transmit Data path control register (TX_DATAPATH_CTRL_CLR)	32	RW	0000_0000
22C	Transmit Data path control register (TX_DATAPATH_CTRL_TOG)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
230 - 244	Channel staus bits (TX_CS_DATA_BITS_0 - TX_CS_DATA_BITS_5)	32	RW	0000_0000
248 - 25C	User data bits (TX_USER_DATA_BITS_0 - TX_USER_DATA_BITS_5)	32	RW	0000_0000
260	DMAC counter control register (TX_DPATH_CNTR_CTRL)	32	RW	0000_0000
264	DMAC counter control register (TX_DPATH_CNTR_CTRL_SET)	32	RW	0000_0000
268	DMAC counter control register (TX_DPATH_CNTR_CTRL_CLR)	32	RW	0000_0000
26C	DMAC counter control register (TX_DPATH_CNTR_CTRL_TOG)	32	RW	0000_0000
270	Transmit Datapath Timestamp Counter Register (TX_DPATH_TSCR)	32	RO	0000_0000
274	Transmit Datapath Bit counter register (TX_DPATH_BCR)	32	RO	0000_0000
278	Transmit datapath Bit count timestamp register. (TX_DPATH_BCTR)	32	RO	0000_0000
27C	Transmmit datapath Bit read timestamp register. (TX_DPATH_BCRR)	32	RO	0000_0000
2A0	HPD Debounce Control Register (HPD_DBNC_CTRL)	32	RW	0003_0D40
2A4	HPD Debounce Control Register (HPD_DBNC_CTRL_SET)	32	RW	0003_0D40
2A8	HPD Debounce Control Register (HPD_DBNC_CTRL_CLR)	32	RW	0003_0D40
2AC	HPD Debounce Control Register (HPD_DBNC_CTRL_TOG)	32	RW	0003_0D40

14.6.4.1.2 Version control register (VERSION)

Indicates the version of Audio XCVR

14.6.4.1.2.1 Offset

Register	Offset
VERSION	0h

14.6.4.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VERID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VERID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.2.3 Fields

Field	Description
31-0 VERID	Version ID Indicates the version of Audio XCVR

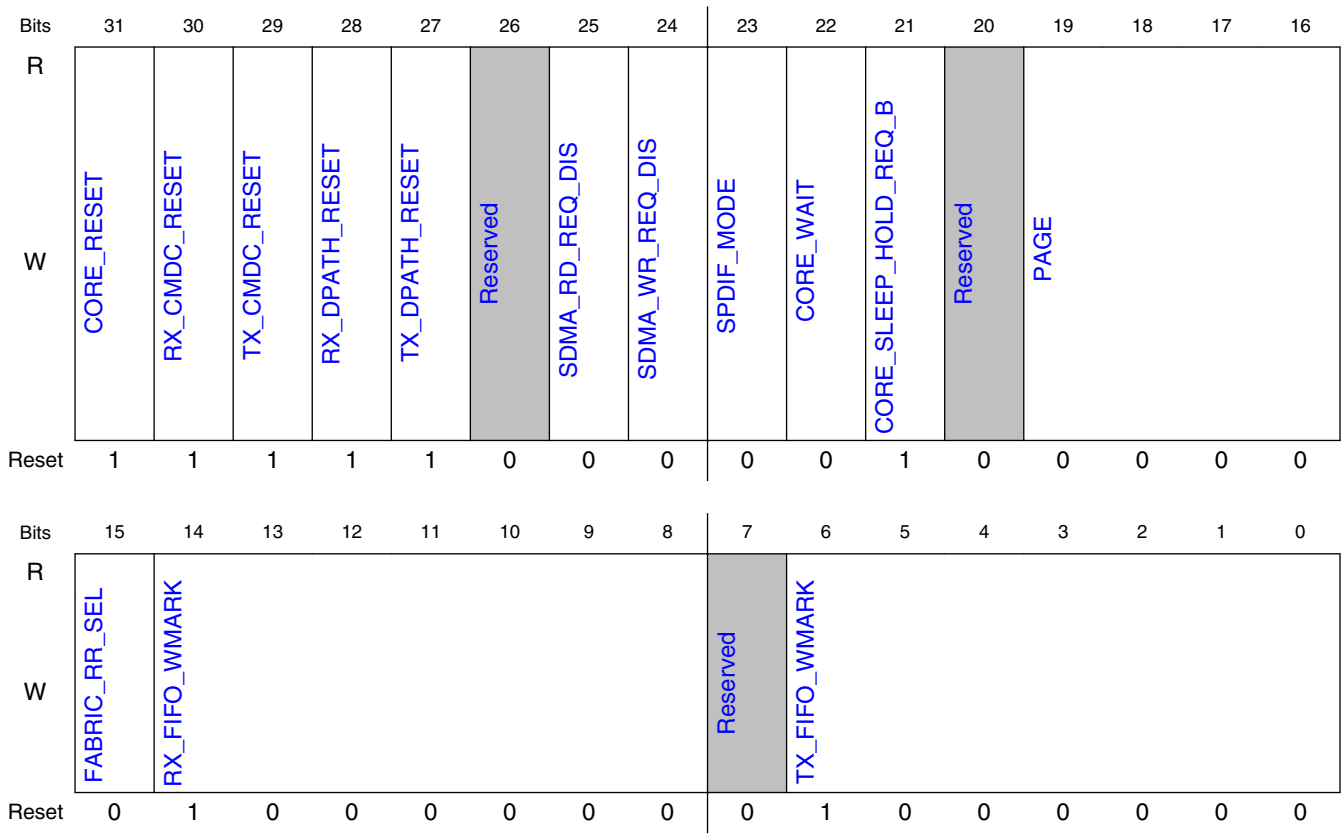
14.6.4.1.3 External control register (EXT_CTRL)

This register provides host control registers for the Audio XCVR

14.6.4.1.3.1 Offset

Register	Offset	Description
EXT_CTRL	10h	External control register
EXT_CTRL_SET	14h	Writing a 1 to a bit in this register sets the corresponding bit in EXT_CTRL
EXT_CTRL_CLR	18h	Writing a 1 to a bit in this register clears the corresponding bit in EXT_CTRL
EXT_CTRL_TOG	1Ch	Writing a 1 to a bit in this register toggles the corresponding bit in EXT_CTRL

14.6.4.1.3.2 Diagram



14.6.4.1.3.3 Fields

Field	Description
31 CORE_RESET	M0+ Reset Controls the reset to the M0+ core. Allows external host to keep M0+ under reset until memory and control registers are configured. Reset is asserted when '1' is written to this bit. Software must allow 8 or more clock cycles before clearing this bit to allow for reset synchronization and propagation.
30 RX_CMDC_RESET	Soft reset to the eARC Common mode Receiver Receive datapath is reset when '1' is written to this bit. Software must allow 8 or more clock cycles before writing a '0' to this bit to allow for reset synchronization and propagation. In SPDIF and ARC modes, this reset can be asserted to save some power.
29 TX_CMDC_RESET	Soft reset to the eARC Common mode Transmitter Transmit datapath is reset when '1' is written to this bit. Software must allow 16 or more clock cycles before writing a '0' to this bit to allow for reset synchronization and propagation. In SPDIF and ARC modes, this reset can be asserted to save some power.
28 RX_DPATH_RESET	Soft reset to the eARC Differential data Receiver Receive datapath is reset when '1' is written to this bit. Software must allow 8 or more clock cycles before writing a '0' to this bit to allow for reset synchronization and propagation.
27	Soft reset to the Datapath for Transmit

Table continues on the next page...

Field	Description
TX_DPATH_RESET	Transmit datapath is reset when '1' is written to this bit. Software must allow 8 or more clock cycles before writing a '0' to this bit to allow for reset synchronization and propagation.
26 —	Reserved
25 SDMA_RD_REQ_DIS	SDMA RD REQ disable This bit field is used to disable SDMA read requests.
24 SDMA_WR_REQ_DIS	SDMA WR REQ disable This bit field is used to disable SDMA write requests.
23 SPDIF_MODE	Indicates SPDIF output mode. If Set, SPDIF data is transmitted.
22 CORE_WAIT	Stop executing code If Set, the core will not execute code, but rather go into WAIT mode.
21 CORE_SLEEP_HOLD_REQ_B	Hold core from going to sleep mode when 0. If not set, the core will not go into sleep mode.
20 —	Reserved
19-16 PAGE	Page Select. Selects 2KB memory that can be accessed by the external interface. Bit 19 is used to pick between the 16KB memory VS the 4KB memory. • 0000 - 16 KB memory, where the 1st 2 KBs are accessible • 0001 - 16 KB memory, where the 2nd 2 KBs are accessible • ... • 0111 - 16 KB memory, where the 8th 2 KBs are accessible • 1000 - 4 KB memory, where the 1st 2 KBs are accessible • 1001 - 4 KB memory, where the 2nd 2 KBs are accessible • 1010 - 1111 - Reserved
15 FABRIC_RR_SEL	Selects Arbitration mode of crossbar switch. This bit selects the Arbitration mode of the cross bar switch. • 0 - M0 core has priority over external core. • 1 - Round Robin Arbitration between M0 core and external master
14-8 RX_FIFO_WMARK	Audio Receive FIFO Watermark Level Indicates the watermark level (number of 32-bit words) at and above which a FIFO read request will be generated Reset to 1/2 the FIFO size. • 0 - Not Applicable • 1 - 127 - Valid values
7 —	Reserved
6-0 TX_FIFO_WMARK	Audio Transmit FIFO Watermark Level Indicates the watermark level (number of 32-bit words) at and below which a FIFO write request will be generated. Reset to 1/2 the FIFO size. • 0 - Not Applicable • 1 - 127 - Valid values

14.6.4.1.4 External Status register (EXT_STATUS)

This register provides status to the host controller

14.6.4.1.4.1 Offset

Register	Offset	Description
EXT_STATUS	20h	External Status register
EXT_STATUS_SET	24h	Writing a 1 to a bit in this register sets the corresponding bit in EXT_STATUS
EXT_STATUS_CLR	28h	Writing a 1 to a bit in this register clears the corresponding bit in EXT_STATUS
EXT_STATUS_TOG	2Ch	Writing a 1 to a bit in this register toggles the corresponding bit in EXT_STATUS

14.6.4.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HPD_I	NEW_UD6_REC	NEW_UD5_REC	NEW_UD4_REC	HEARTBEAT_STATUS	Reserved	RX_CMDC_COMMA_TO	Reserved	RX_CMDC_RESP_TO	Reserved	TX_PIPE_EMPTY	Reserved		CM0_SLEEP_HOLD_ACK_B	CM0_DEEP_SLEEP	CM0_SLEEPING
W									W1C							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NO_RX_FIFO_ENTRIES								NO_TX_FIFO_ENTRIES							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.4.3 Fields

Field	Description
31	HPD Input status

Table continues on the next page...

Field	Description
HPD_I	This status provides the read of the HPD input.
30 NEW_UD6_REC	New user data This bit indicates new user data of type 6
29 NEW_UD5_REC	New user data This bit indicates new user data of type 5
28 NEW_UD4_REC	New user data This bit indicates new user data of type 4
27 HEARTBEAT_STATUS	Earc Connection Status This bit indicates a heartbeat was successfully exchanged within the given timeframe. If this bit is not set, check DEBUG register for CMDC state.
26 —	Reserved
25 RX_CMDC_COMMA_TO	Receiver CMDC comma timeout Interrupt This interrupt indicates the receiver CMDC comma sequence did not result in a successful heartbeat. This interrupt is sent to the host core on CMDC_TO_ERR.
24 —	Reserved
23 RX_CMDC_RESP_TO	CMDC Response not sent in programmed time This bit indicates RX CMDC respose could not be sent out by M0 core in the required timeframe. This interrupt is sent to the host core on CMDC_TO_ERR.
22 —	Reserved
21 TX_PIPE_EMPTY	Indicates TX pipe status. This status bit is used to indicate TX pipeline is empty. This can be used to determine if new block of data can be sent / Frequency can be updated.
20-19 —	Reserved
18 CM0_SLEEP_HOLD_ACK_B	Sleep extension acknowledge.
17 CM0_DEEP_SLEEP	CM0 is in deep sleep mode. Using this bit, the eARC clock into the core can be gated for power savings.
16 CM0_SLEEPING	CM0 is in Sleep mode. Using this bit, the eARC clock into the core can be gated for power savings.
15-8 NO_RX_FIFO_ENTRIES	RX FIFO entries This status provides the number of entries in the receiver FIFO
7-0	TX FIFO entries

Field	Description
NO_TX_FIFO_ENTRIES	This status provides the number of entries in the transmit FIFO

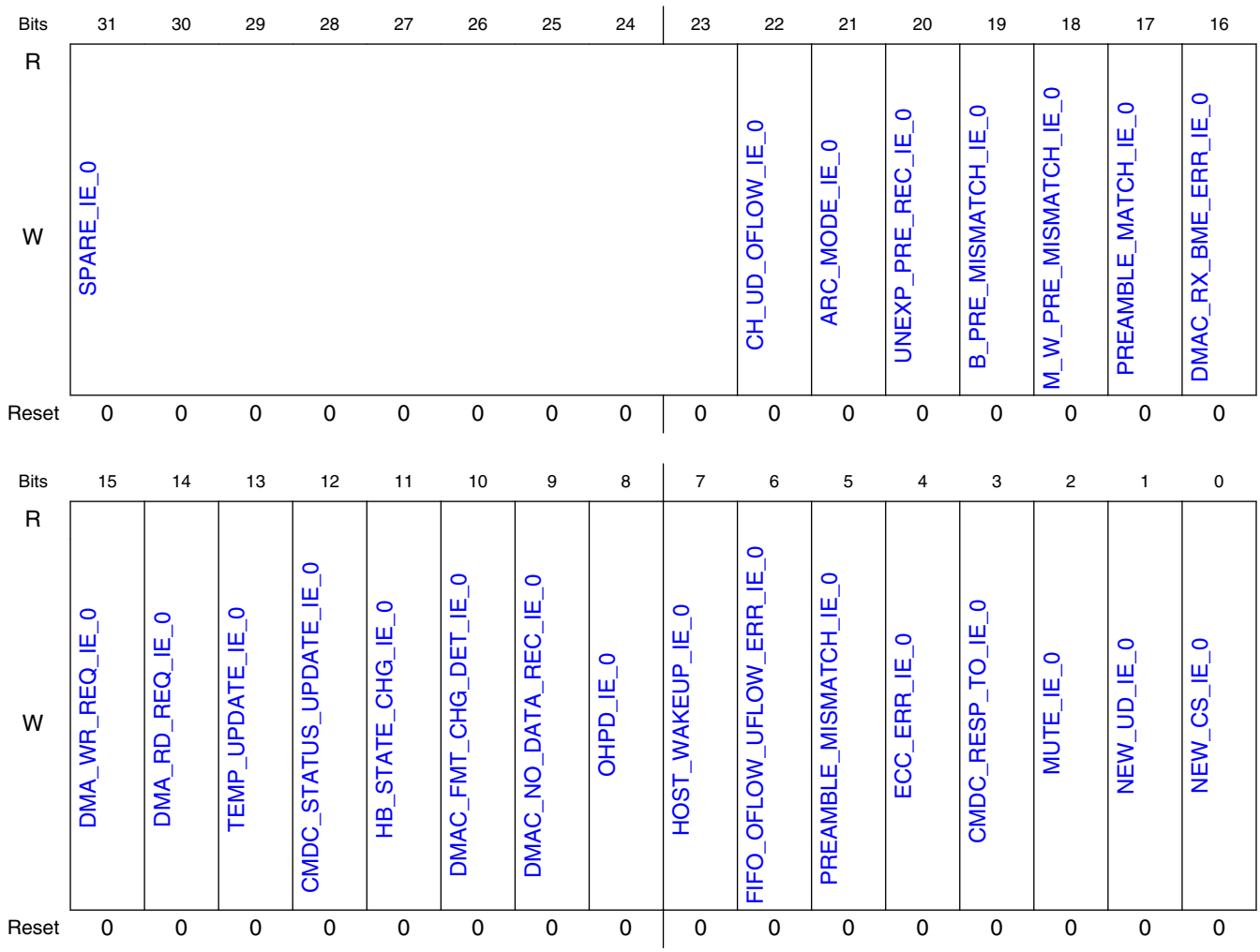
14.6.4.1.5 Interrupt enables for interrupt 0 (EXT_IER0)

Provides enable bits to enable interrupt 0 generation. All enables are active high. Interrupt enables for bits 22 - 31 are reserved for future use

14.6.4.1.5.1 Offset

Register	Offset	Description
EXT_IER0	30h	Interrupt enables for interrupt 0
EXT_IER0_SET	34h	Writing a 1 to a bit in this register sets the corresponding bit in EXT_IER0
EXT_IER0_CLR	38h	Writing a 1 to a bit in this register clears the corresponding bit in EXT_IER0
EXT_IER0_TOG	3Ch	Writing a 1 to a bit in this register toggles the corresponding bit in EXT_IER0

14.6.4.1.5.2 Diagram



14.6.4.1.5.3 Fields

Field	Description
31-23	Spare interrupts
SPARE_IE_0	These spare interrupts can be set by M0+.
22	Channel status or used data could not be stored.
CH_UD_OFLOW_IE_0	This interrupt enables channel status or user data overflow interrupt. This interrupt is asserted when FW is unable to store user data or channel status because the previous entries were not cleared.
21	Interrupt to indicate ARC mode setup.
ARC_MODE_IE_0	This interrupt enables SEL_ARC_MODE interrupt indicating eARC RX state machine could not establish a heartbeat. Check to see if ARC mode needs to be set up.
20	Interrupt enable for Unexpected preamble received.
UNEXP_PRE_REC_IE_0	This enables the interrupt to be asserted everytime HW detects a preamble was not received when expected or more than 8 bits were received before a preamble was detected.

Table continues on the next page...

Field	Description
19 B_PRE_MISMATCH_IE_0	Interrupt enable for sub-frame B preamble mismatch received. This enables the interrupt to be asserted everytime HW detects a B sub-frame preamble mismatch.
18 M_W_PRE_MISMATCH_IE_0	Interrupt enable for sub-frame M/W preamble mismatch received. This enables the interrupt to be asserted everytime HW detects a M/W sub-frame preamble mismatch.
17 PREAMBLE_MATCH_IE_0	Interrupt enable for preamble match received. This enables the interrupt to be asserted everytime HW detects a preamble match and FW moves the data to memory and asserts the interrupt.
16 DMAC_RX_BME_ERR_IE_0	Bi-phase mark encoding error This enables the interrupt to be asserted everytime Bi-phase mark encoded data received has an error. i.e. No toggle after 2 bits.
15 DMA_WR_REQ_IE_0	Request to write data to FIFO. This enables the interrupt to be asserted everytime Watermark number of entries are available to be filled with tx sub-frames.
14 DMA_RD_REQ_IE_0	Request to read data from FIFO. This enables the interrupt to be asserted everytime Watermark number of entries are available to be read.
13 TEMP_UPDATE_IE_0	Update request for chip temperature value. This enables the interrupt to be asserted every 5s, requesting an update in the temperature value for PHY settings.
12 CMDC_STATUS_UPDATE_IE_0	Interrupt enable for CMDC status register update. This enables the interrupt to be asserted when there is a change in Capabilities data structure / Latency request of the CMDC Status register.
11 HB_STATE_CHG_IE_0	Interrupt enable for Heartbeat status change This enables the interrupt to be asserted when there is a change in heartbeat status. If heartbeat is either achieved for the first time / heartbeat is lost after the channel was established.
10 DMAC_FMT_CHG_DET_IE_0	Indicates DMAC format change was detected No DMAC input data for over 65 ms. This interrupt can be set by HW / FW.
9 DMAC_NO_DATA_REC_IE_0	Indicates no DMAC data is received. No DMAC input data for over 2 ms.
8 OHPD_IE_0	Output HPD interrupt enable. Enables assertion of interrupt when the output HPD is set.
7 HOST_WAKEUP_IE_0	Host wakeup on CEC match interrupt enable. Enables wakeup interrupt to the host from CEC commands.
6 FIFO_OFLOW_UFLOW_ERR_IE_0	Receive FIFO overflow error interrupt enable. Enables assertion of interrupt when FIFO underflows (in TX mode) / overflows (in RX mode).
5	Preamble mismatch interrupt enable.

Table continues on the next page...

Field	Description
PREAMBLE_MISMATCH_IE_0	Enables assertion of interrupt when the expected preamble is not received. Usually when a W does not follow an M / an unexpected B preamble is received. Once an unexpected preamble is received, the curent block of data is dropped and sub-frame data is synced to the next input B preamble.
4 ECC_ERR_IE_0	60958 Compressed data uncorrectable error interrupt enable Enables assertion of interrupt when 60958 compressed data is received with uncorrectable error.
3 CMDC_RESP_TO_IE_0	Receiver CMDC data response timeout interrupt enable Enables assertion of an interrupt to indicate response in the programmed CMDC window was missed.
2 MUTE_IE_0	Enable for Mute detected interrupt Enables assertion of interrupt when HW / FW mute bit is set for receive data.
1 NEW_UD_IE_0	Enable for new user data received interrupt Enables assertion of interrupt when a new user data block is received.
0 NEW_CS_IE_0	Enable for New channel status block received interrupt Enables assertion of interrupt when a new channel status block is received.

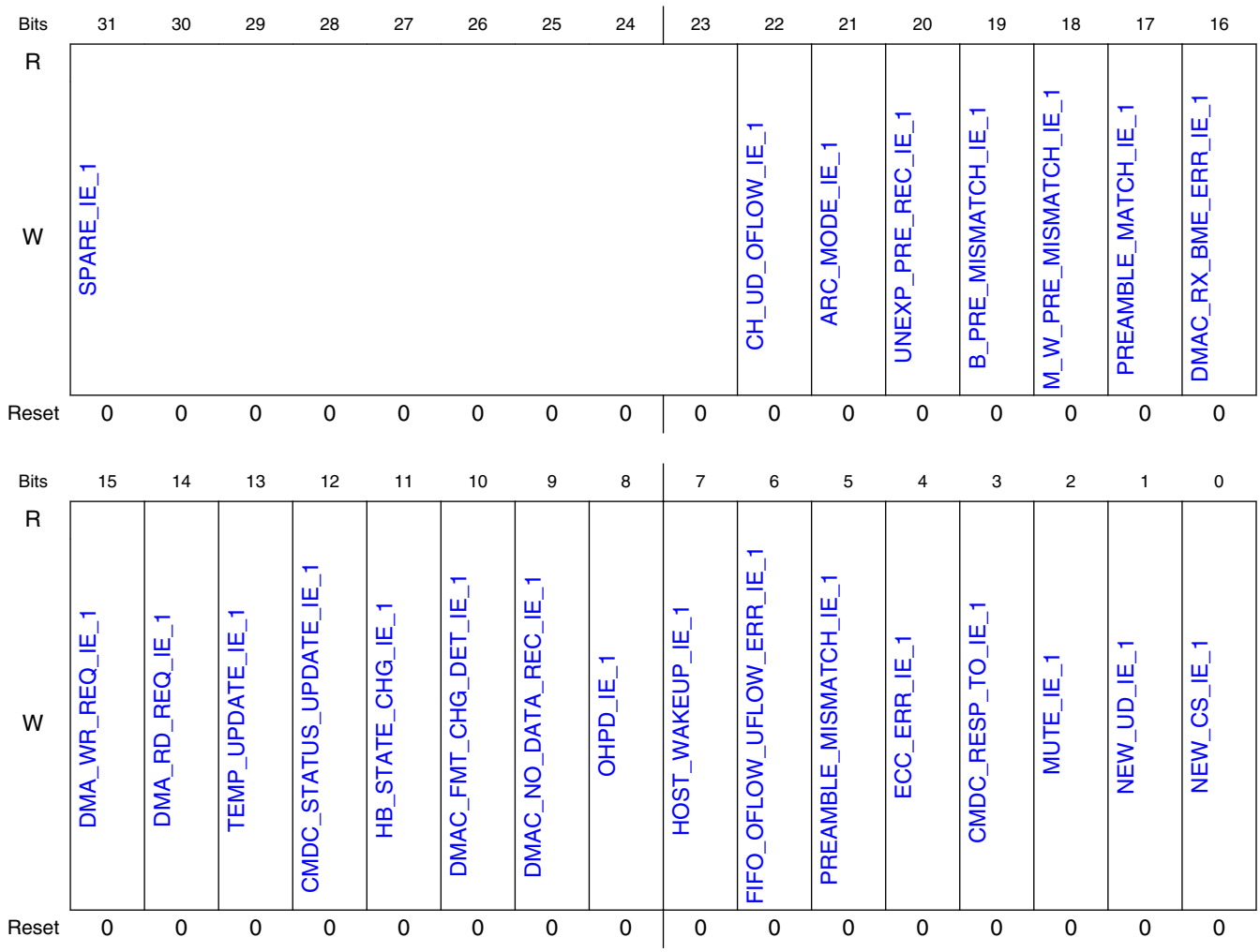
14.6.4.1.6 Interrupt enables for interrupt 1 (EXT_IER1)

Provides enable bits to enable interrupt 1 generation. All enables are active high. Interrupt enables 22 - 31 are reserved for future use

14.6.4.1.6.1 Offset

Register	Offset	Description
EXT_IER1	40h	Interrupt enables for interrupt 1
EXT_IER1_SET	44h	Writing a 1 to a bit in this register sets the corresponding bit in EXT_IER1
EXT_IER1_CLR	48h	Writing a 1 to a bit in this register clears the corresponding bit in EXT_IER1
EXT_IER1_TOG	4Ch	Writing a 1 to a bit in this register toggles the corresponding bit in EXT_IER1

14.6.4.1.6.2 Diagram



14.6.4.1.6.3 Fields

Field	Description
31-23 SPARE_IE_1	Spare interrupt enables. Enables assertion of spare interrupts that cna be set by M0+.
22 CH_UD_OFLOW_IE_1	Channel status or used data could not be stored. This interrupt enables channel status or user data overflow interrupt. This interrupt is asserted when FW is unable to store user data or channel status because the previous entries were not cleared.
21 ARC_MODE_IE_1	Interrupt to indicate ARC mode setup. This interrupt enables SEL_ARC_MODE interrupt indicating eARC RX state machine could not establish a heartbeat. Check to see if ARC mode needs to be set up.
20 UNEXP_PRE_REC_IE_1	Interrupt enable for Unexpected preamble received. This enables the interrupt to be asserted everytime HW detects a preamble was not received when expected or more than 8 bits were received before a preamble was detected.

Table continues on the next page...

Field	Description
19 B_PRE_MISMATCH_IE_1	Interrupt enable for sub-frame B preamble mismatch received. This enables the interrupt to be asserted everytime HW detects a B sub-frame preamble mismatch.
18 M_W_PRE_MISMATCH_IE_1	Interrupt enable for sub-frame M/W preamble mismatch received. This enables the interrupt to be asserted everytime HW detects a M/W sub-frame preamble mismatch.
17 PREAMBLE_MATCH_IE_1	Interrupt enable for preamble match received. This enables the interrupt to be asserted everytime HW detects a preamble match and FW moves the data to memory and asserts the interrupt.
16 DMAC_RX_BME_ERR_IE_1	Bi-phase mark encoding error This enables the interrupt to be asserted everytime Bi-phase mark encoded data received has an error. i.e. No toggle after 2 bits.
15 DMA_WR_REQ_IE_1	Request to write data to FIFO. This enables the interrupt to be asserted everytime Watermark number of entries are available to be filled with tx sub-frames.
14 DMA_RD_REQ_IE_1	Request to read data from FIFO. This enables the interrupt to be asserted everytime Watermark number of entries are available to be read.
13 TEMP_UPDATE_IE_1	Update request for chip temperature value. This enables the interrupt to be asserted every 5s, requesting an update in the temperature value for PHY settings.
12 CMDC_STATUS_UPDATE_IE_1	Interrupt enable for CMDC status register update. This enables the interrupt to be asserted when there is a change in Capabilities data structure / Latency request of the CMDC Status register.
11 HB_STATE_CHG_IE_1	Interrupt enable for Heartbeat status change This enables the interrupt to be asserted when there is a change in heartbeat status. If heartbeat is either achieved for the first time / heartbeat is lost after the channel was established.
10 DMAC_FMT_CHG_DET_IE_1	Indicates DMAC format change was detected No DMAC input data for over 65 ms. This interrupt can be set by HW / FW.
9 DMAC_NO_DATA_REC_IE_1	Indicates no DMAC data is received. No DMAC input data for over 2 ms.
8 OHPD_IE_1	Output HPD interrupt enable. Enables assertion of interrupt when the output HPD is set.
7 HOST_WAKEUP_IE_1	Host wakeup on CEC match interrupt enable. Enables wakeup interrupt to the host from CEC commands.
6 FIFO_OFLOW_UFLOW_ERR_IE_1	Receive FIFO overflow error interrupt enable. Enables assertion of interrupt when FIFO underflows (in TX mode) / overflows (in RX mode).
5	Preamble mismatch interrupt enable.

Table continues on the next page...

Field	Description
PREAMBLE_MISMATCH_IE_1	Enables assertion of interrupt when the expected preamble is not received. Usually when a W does not follow an M / an unexpected B preamble is received. Once an unexpected preamble is received, the curent block of data is dropped and sub-frame data is synced to the next input B preamble.
4 ECC_ERR_IE_1	60958 Compressed data uncorrectable error interrupt enable Enables assertion of interrupt when 60958 compressed data is received with uncorrectable error.
3 CMDC_RESP_TO_IE_1	Receiver CMDC data response timeout interrupt enable Enables assertion of an interrupt to indicate response in the programmed CMDC window was missed.
2 MUTE_IE_1	Enable for Mute detected interrupt Enables assertion of interrupt when HW / FW mute bit is set for receive data.
1 NEW_UD_IE_1	Enable for new user data received interrupt Enables assertion of interrupt when a new user data block is received.
0 NEW_CS_IE_1	Enable for New channel status block received interrupt Enables assertion of interrupt when a new channel status block is received.

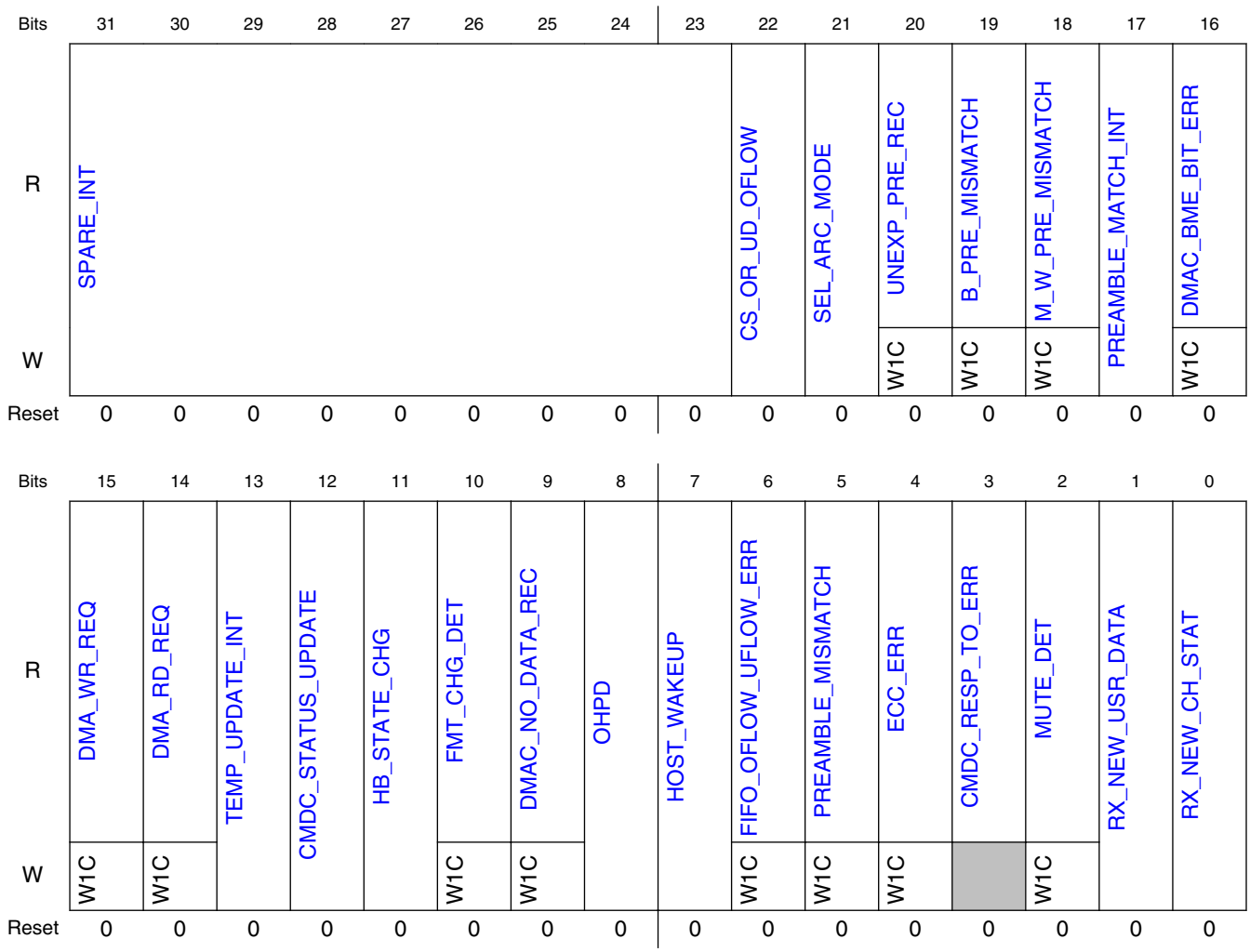
14.6.4.1.7 External Interrupt Status register (EXT_ISR)

Interrupt status for the host core

14.6.4.1.7.1 Offset

Register	Offset	Description
EXT_ISR	50h	External Interrupt Status register
EXT_ISR_SET	54h	Writing a 1 to a bit in this register sets the corresponding bit in EXT_ISR
EXT_ISR_CLR	58h	Writing a 1 to a bit in this register clears the corresponding bit in EXT_ISR
EXT_ISR_TOG	5Ch	Writing a 1 to a bit in this register toggles the corresponding bit in EXT_ISR

14.6.4.1.7.2 Diagram



14.6.4.1.7.3 Fields

Field	Description
31-23 SPARE_INT	Extra interrupt. Currently not driven. Can be set by M0
22 CS_OR_UD_OFLOW	Channel status or used data could not be stored. This interrupt indicates channel status or user data should have been stored in data ram, but since the previous entries were not cleared, FW was unable to store it.
21 SEL_ARC_MODE	Set when CMDC SM falls out of eARC mode. This interrupt indicates eARC RX state machine could not establish a heartbeat. Check to see if ARC mode needs to be set up.
20 UNEXP_PRE_REC	Set when DMAC preamble was received after unexpected number of input bits. This interrupt indicates DMAC subframe received has a B preamble error. i.e, B preamble expected and was not received / B preamble received and was not expected.

Table continues on the next page...

Field	Description
19	Set when DMAC preamble of B has an error.
B_PRE_MISMATCH	This interrupt indicates DMAC subframe received has a B preamble error. i.e, B preamble expected and was not received / B preamble received and was not expected.
18	Set when DMAC preamble of M/W has an error.
M_W_PRE_MISMATCH	This interrupt indicates DMAC subframe received has a preamble error.
17	Interrupt to indicate PA PB / DTC CD preamble match was detected.
PREAMBLE_MATCH_INT	This interrupt is asserted every time a PA PB match is detected by HW. Host core will neeed to go read the Circular buffer in 4KB memory.
16	Set when DMAC BME data has an error.
DMAC_BME_BIT_ERR	This interrupt indicates BME data received has an error. After a preamble was received, the input data did not toggle after every 2 bits. This interrupt will have to be cleared.
15	Set when DMA write request is asserted.
DMA_WR_REQ	This interrupt indicates Watermark number of entries are free to be filled with new sub-frames. This interrupt will have to be cleared.
14	Set when DMA read request is asserted.
DMA_RD_REQ	This interrupt indicates Watermark number of entries are avaialble to be read. This interrupt will have to be cleared.
13	Interrupt to get the new temperature value.
TEMP_UPDATE_INT	This interrupt is asserted every 5s for an update of the temperature reading.
12	Interrupt enable for CMDC status register update.
CMDC_STATUS_UPDATE	This interrupt is asserted when eARC in TX mode and new capabilities data structure or latency information is available. This should be read from the data RAM at offset 0x300 for the capabilities data structure and byte at offset 0x6 for the latency update.
11	Interrupt enable for Heartbeat status change
HB_STATE_CHG	This interrupt indicates a change in heartbeat status. The link was either established for the first time or an established link lost heartbeat.
10	Format change detect interrupt
FMT_CHG_DET	This interrupt indicates no input DMAC data is received for over 65ms. This interrupt can be set by HW / FW.
9	No DMAC data is received for 1us.
DMAC_NO_DATA_REC	This interrupt indicates input DMAC hasn't toggled in 1us. This is 4* the slowest data rate. This can either indicate a format change / so other issue in the data pipe.
8	HPD output driver
OHPD	This register drives the output HPD value. It should not be cleared by the external host. If the host core does not want to see this interrupt, it should mask the interrupt in IE0 / IE1 registers.
7	Host wakeup on CEC OPCODE match interrupt.
HOST_WAKEUP	This interrupt indicates a wakeup source to the host core.
6	Receive FIFO overflow error interrupt
FIFO_OFLOW_UFLOW_ERR	This interrupt indicates the data receive FIFO is full, or the TX fifo is empty. In RX mode, the new frames will be dropped until the fifo has room for more sub-frames. In TX mode, no data will be sent out on the TX path.

Table continues on the next page...

Field	Description
5 PREAMBLE_MISMATCH	Preamble mismatch interrupt This interrupt indicates an unexpected preamble is received. When this interrupt is set, HW clears the RX pipe and waits for a new B preamble to start receiving a fresh block of data.
4 ECC_ERR	60958 Compressed data uncorrectable error interrupt This bit indicates there was an uncorrectable error in received compressed data.
3 CMDC_RESP_TO_ERR	CMDC response timeout interrupt This interrupt indicates no CMDC response or comma was received / sent in the CMDC response window. Refer to EXT_STATUS register to see which protocol failed. This interrupt can be cleared in the EXT_STATUS register.
2 MUTE_DET	Interrupt to indicate HW mute bit was detected. This interrupt indicates mute bit was detected in HW or set by FW.
1 RX_NEW_USR_DATA	Received new User data Information This interrupt indicates a unique user data type was received. Refer to the External status register to see if it was IU type 4/5/6.
0 RX_NEW_CH_STAT	Received new channel status block This interrupt indicates a unique channel status block was received.

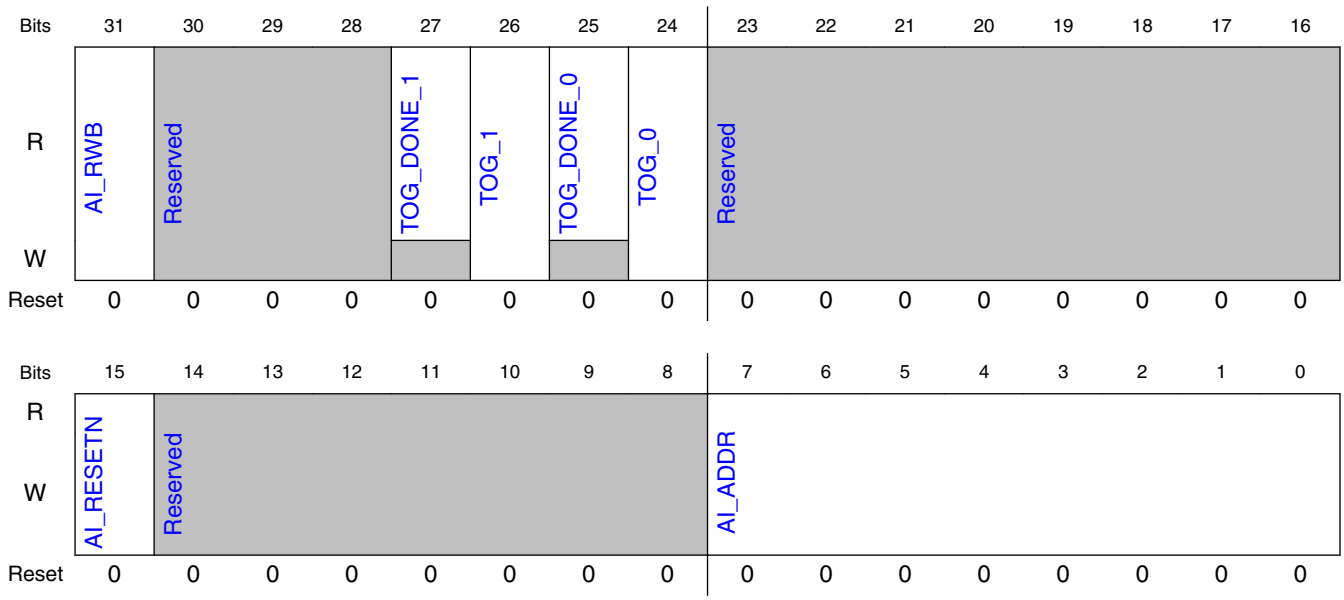
14.6.4.1.8 AI interface control register (PHY_AI_CTRL)

Provides address and toggle bit controls for the AI interface to the PHY.

14.6.4.1.8.1 Offset

Register	Offset	Description
PHY_AI_CTRL	90h	AI interface control register
PHY_AI_CTRL_SET	94h	Writing a 1 to a bit in this register sets the corresponding bit in PHY_AI_CTRL
PHY_AI_CTRL_CLR	98h	Writing a 1 to a bit in this register clears the corresponding bit in PHY_AI_CTRL
PHY_AI_CTRL_TOG	9Ch	Writing a 1 to a bit in this register toggles the corresponding bit in PHY_AI_CTRL

14.6.4.1.8.2 Diagram



14.6.4.1.8.3 Fields

Field	Description
31 AI_RWB	AI Read / write control bit This register field holds the control to read / write the AI interface.
30-28 —	Reserved
27 TOG_DONE_1	AI toggle done bit This register field holds the Toggle done value for the audio XVCR PHY. When this bit matches TOG_0 the read / write operation has successfully completed.
26 TOG_1	AI toggle bit This register field holds the Toggle value for the audio XVCR PHY. When this bit is set, the AI_ADDR filed and AI_RWB take effect. This bit should not be toggled unless it matches TOG_DONE_1. and TOG_0 matches TOG_DONE_0
25 TOG_DONE_0	AI toggle done bit This register field holds the Toggle done value for the audio PLL in the XCVR PHY. When this bit matches TOG_0 the read / write operation has successfully completed.
24 TOG_0	AI toggle bit This register field holds the Toggle value for the audio PLL in the PHY. When this bit is set, the AI_ADDR filed and AI_RWB take effect. This bit should not be toggled unless it matches TOG_DONE_0 and TOG_1 matches TOG_DONE_1
23-16 —	Reserved
15 AI_RESETN	AI reset bit This register field controls the reset of the AI interface in the eARC controller.

Table continues on the next page...

Field	Description
14-8 —	Reserved
7-0 AI_ADDR	AI ADDR value This register field holds the ADDR value to be sent out on the AI bus. This value can correspond to the PLL or the Audio XVCR registers. The toggle bit decides where the address will be accepted.

14.6.4.1.9 AI interface WDATA register (PHY_AI_WDATA)

Provides the data to be written using the AI interface to the PHY.

14.6.4.1.9.1 Offset

Register	Offset
PHY_AI_WDATA	A0h

14.6.4.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.9.3 Fields

Field	Description
31-0 WDATA	Write data Write data for the AI interface. If performing an AI write, this field should be set before the TOG_* bit is toggled.

14.6.4.1.10 AI interface RDATA register (PHY_AI_RDATA)

Provides the data read using the AI interface to the PHY.

14.6.4.1.10.1 Offset

Register	Offset
PHY_AI_RDATA	A4h

14.6.4.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.10.3 Fields

Field	Description
31-0 RDATA	Read data Read data from the AI interface. If performing an AI read, this field should be read after TOG_* matches the relevent TOG_DONE_*.

14.6.4.1.11 Audio XCVR datapath status (DPATH_STATUS)

Provides the status for datapath

14.6.4.1.11.1 Offset

Register	Offset
DPATH_STATUS	A8h

14.6.4.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX_FRM_CNT								RX_FRM_CNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.11.3 Fields

Field	Description
31-16 —	Reserved
15-8 TX_FRM_CNT	Count of transmitted frames in a block This register holds the count of transmitted frames in a block of data.
7-0 RX_FRM_CNT	Count of received frames in a block This register holds the count of received frames in a block of data.

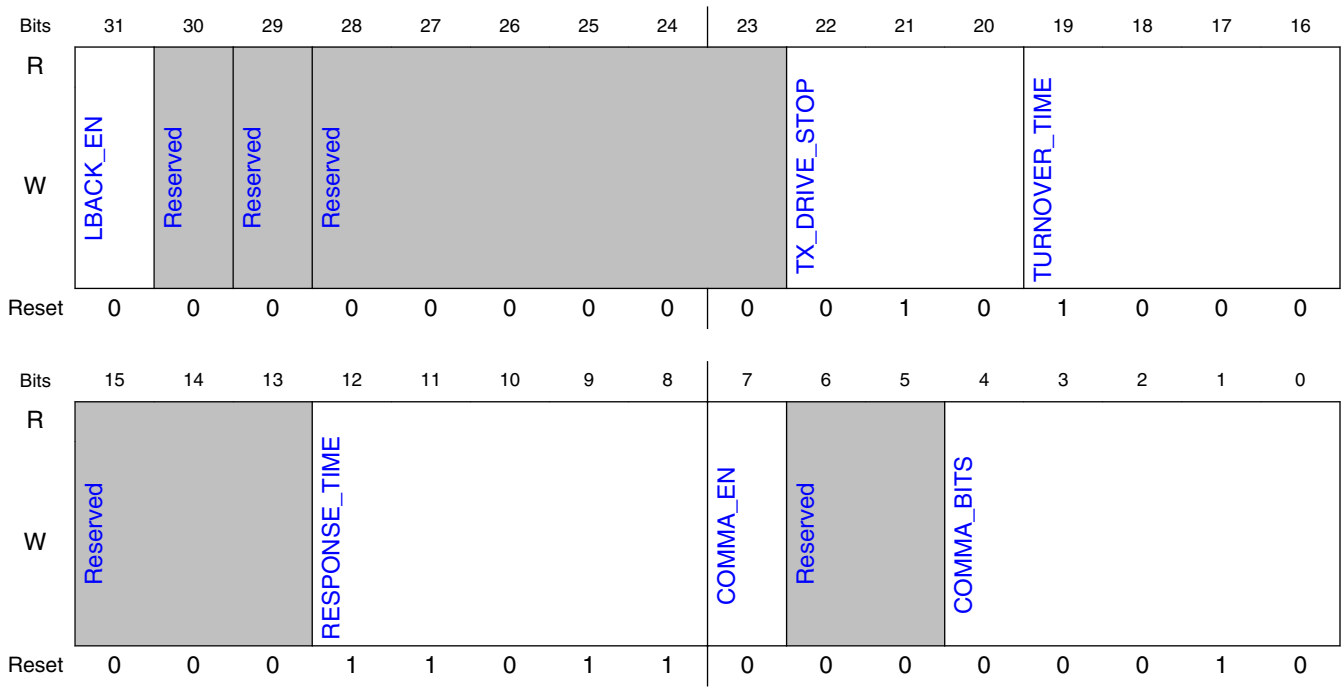
14.6.4.1.12 CMDC receiver control register (RX_CMDC_CTRL)

Configures and controls operations for the RX CMDC block.

14.6.4.1.12.1 Offset

Register	Offset	Description
RX_CMDC_CTRL	C0h	CMDC receiver control register
RX_CMDC_CTRL_SET	C4h	Writing a 1 to a bit in this register sets the corresponding bit in RX_CMDC_CTRL
RX_CMDC_CTRL_CLR	C8h	Writing a 1 to a bit in this register clears the corresponding bit in RX_CMDC_CTRL
RX_CMDC_CTRL_TOG	CCh	Writing a 1 to a bit in this register toggles the corresponding bit in RX_CMDC_CTRL

14.6.4.1.12.2 Diagram



14.6.4.1.12.3 Fields

Field	Description
31 LBACK_EN	Loopback enable If set, enables loopback between CMDC_TX and CDMC_RX blocks. For test purposes only.
30 —	reserved
29 —	Reserved
28-23 —	reserved
22-20 TX_DRIVE_STOP	Transmitter bus release time This is the time in "us" measured from end of transmission of message (i.e. transmission of parity bit), during which the transmitter shall drive the bus low before releasing it. This field is programmed in steps of 0.5 us. A value of 3 represents 1.5 us time. • 0 - This implies bypass drive stop timer. Transmitter will stop driving the output immediately after the last bit is sent. • 1 - 7 Valid values. Default is 3.
19-16 TURNOVER_TIME	Minimum time before a response is generated Time in "us" - microseconds, which is measured from the end of a received message (i.e. reception of parity bit). M0+ core or external host can provide a response before this time has expired but the message will be ignored until turnover time has expired.

Table continues on the next page...

Field	Description
	• 0 - This implies bypass mode for turnover timer. Response will be sent / accepted when ready depending on response time. • 1 - 15 Valid values. Default is 7.
15-13 —	Reserved
12-8 RESPONSE_TIME	Transmitter response timeout to a received message This field defines the timeout time in "us" - microseconds, which is measured from the end of a received message (i.e. reception of parity bit). If a message response from M0+ or external host is not available within this window, the response is dropped. • 0 - This implies bypass mode for response timer. Response will be accepted whenever it is available • 1 - 31 Valid values. Default is 27.
7 COMMA_EN	Enables COMMA pattern generation Enables generation of the COMMA pattern as defined in the CMDC_TX_DATA register, and COMMA_BITS bit field (Default: 2 bits = 0x2). The combination of CMDC_TX_DATA register and CMDC_CTRL[COMMA_BITS] determines the comma pattern. • 0 - Disable Comma transmission • 1 - Enable comma transmission.
6-5 —	Reserved
4-0 COMMA_BITS	Number of repeating bits in COMMA pattern Determines the number of bits (LSBs) of the CMDC_TX_DATA register that will be used as the repeating COMMA pattern. This field takes effect only when COMMA_EN = '1'. • 0, 27-31 - Invalid value.. • 1 - 26 Valid values. Default is 2.

14.6.4.1.13 eARC receiver CMDC status (RX_CMDC_STATUS)

Provides the status for CMDC in receiver mode.

14.6.4.1.13.1 Offset

Register	Offset
RX_CMDC_STATUS	D0h

14.6.4.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved												CMDC_STATE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.13.3 Fields

Field	Description
31-20 —	Reserved
19-16 CMDC_STATE	Current state of the RX CDMC control state machine Provides the current state of CMDC control state machine for M0+ or external host. If M0+ writes to this field, the field will be cleared. • 0 - Idle state. No transactions in flight • 1 - Pending state. Data is received, waiting for M0+ to respond. • 2 - Ready state. M0+ has responded. Waiting for turnover time to expore. • 3 - Send state. Response message is being transmitted • 4 - Send state. Response message is being transmitted • 5 - Timeout state. M0+ did not respond within the required time. • 6 - Compplete. Response sent / received.
15-0 —	Reserved

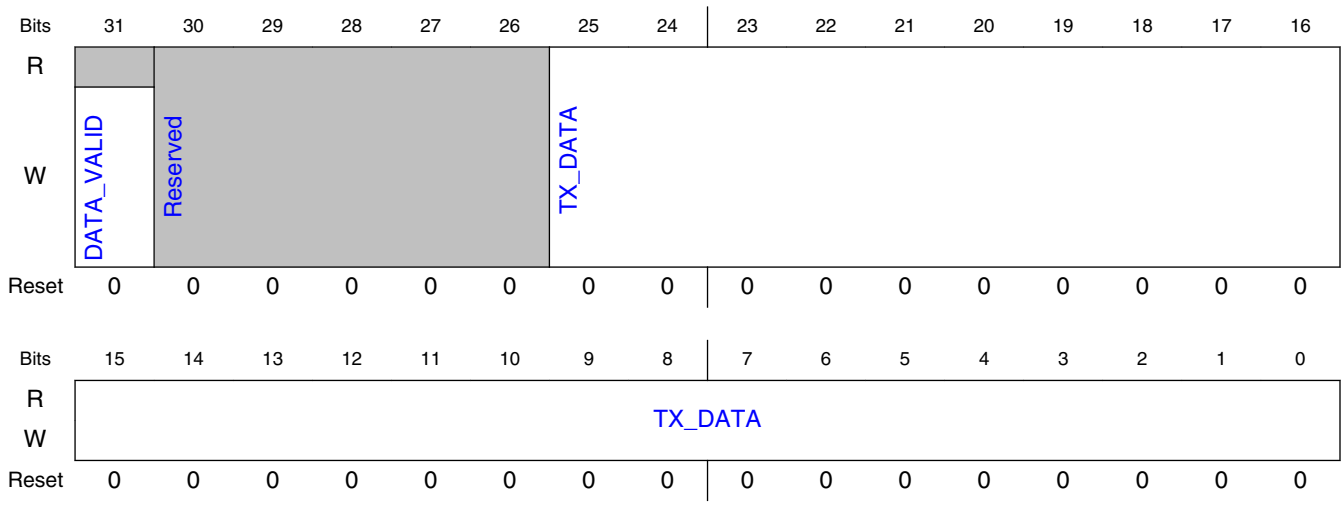
14.6.4.1.14 CMDC transmit data register (RX_CMDC_TX_DATA)

Provides the data for tranmission and control to indicate a new valid data.

14.6.4.1.14.1 Offset

Register	Offset
RX_CMDC_TX_DATA	E0h

14.6.4.1.14.2 Diagram



14.6.4.1.14.3 Fields

Field	Description
31 DATA_VALID	Transmit Data Valid Write only bit. Read will return zeros. User must write '1' to this bit to indicate new valid data for transmission. **NOTE:** Writing a '1' to this bit causes the bit to toggle internally which is used as an indication for new data available for transmission.
30-26 —	Reserved
25-0 TX_DATA	Transmit data This register is used to provide the COMMA pattern (when CMDC_CTRL[COMMA_EN] ='1') and data message during normal data transfer operation.

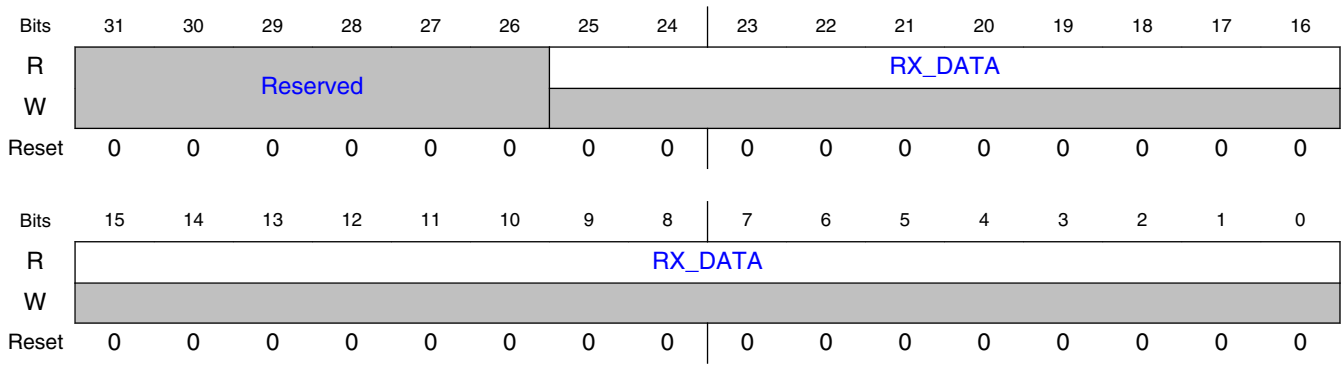
14.6.4.1.15 CMDC receive data register (RX_CMDC_RX_DATA)

Provides the recevied data packet for processing by software.

14.6.4.1.15.1 Offset

Register	Offset
RX_CMDC_RX_DATA	F0h

14.6.4.1.15.2 Diagram



14.6.4.1.15.3 Fields

Field	Description
31-26 —	Reserved
25-0 RX_DATA	Receive data Receive data after being qualified with CMDC_RX_MASK and CMDC_RX_MATCH registers is stored in this register.

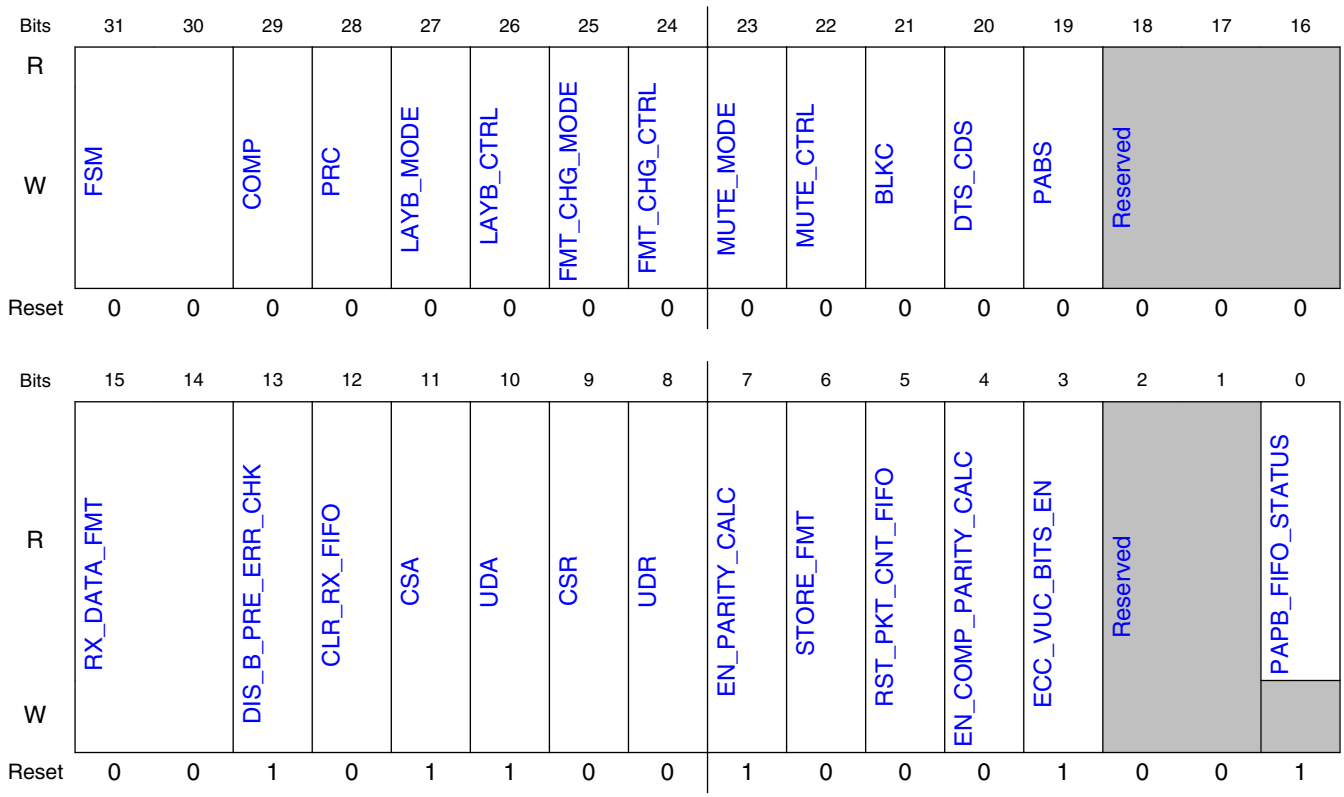
14.6.4.1.16 Data path control register (RX_DATAPATH_CTRL)

Configuration and control bist for the Receive audio block.

14.6.4.1.16.1 Offset

Register	Offset	Description
RX_DATAPATH_CTRL	180h	Data path control register
RX_DATAPATH_CTRL_SET	184h	Writing a 1 to a bit in this register sets the corresponding bit in RX_DATAPATH_CTRL
RX_DATAPATH_CTRL_CLR	188h	Writing a 1 to a bit in this register clears the corresponding bit in RX_DATAPATH_CTRL
RX_DATAPATH_CTRL_TOG	18Ch	Writing a 1 to a bit in this register toggles the corresponding bit in RX_DATAPATH_CTRL

14.6.4.1.16.2 Diagram



14.6.4.1.16.3 Fields

Field	Description
31-30 FSM	IEC60958-1 Frame Synchronization Mode Selects between frame sync or immediate capture of data. • 00 - Normal, Sync to block using B preamble • 01 - Reserved • 10 - Sync to B or M preamble • 11 - Start with the first received preamble.
29 COMP	Compressed data search mode Selects between M0+ control or hardware control for enabling compressed data search. • 0 - Compressed format autodetect • 1 - M0+ control
28 PRC	Process Compressed Selects between De-interleaving, and ECC correcting compressed data and passing through the data as received to the FIFO. COMP value selects if this bit is set by FW / HW. • 1 - Process interleved, ECC protected IEC61937 data • 0 - pass though data to FIFO
27 LAYB_MODE	Layout B Indicates compressed data layout B is being used • 0 - Not Layout B • 1 - Layout B

Table continues on the next page...

Field	Description
26 LAYB_CTRL	Layout B mode control Selects between M0+ control or hardware control for enabling Layout B mode. • 0 - HW control. Layout B is detected using CS bits 47:44 • 1 - M0+ Control
25 FMT_CHG_MODE	Format change detected. Reset HW for next frame Set by M0+ to indicate format change detected. It is cleared by HW to indicate format information was reset.
24 FMT_CHG_CTRL	Format Change detection control. Selects between M0+ control or hardware control for detecting format change. • 0 - HW control. When no input data is received for 65 ms, format change register is set. • 1 - M0+ Control
23 MUTE_MODE	Mute mode control Selects between M0+ control or hardware control for enabling Mute detection. • 0 - Channel Status control. MBIT field is used to select the bit that will control the mute request • 1 - M0+ Control
22 MUTE_CTRL	M0+ mute request If M0+ is used to control MUTE assertion, this bit is used to set mute mode. • 0 - Normal operation • 1 - Assert mute request
21 BLKC	Block Compressed data If this bit is set, compressed data is blocked. It won't be stored in the FIFO and CS and UD won't be processed.
20 DTS_CDS	Enable DTS CD 14 preamble search Enable Preamble Pa Pb search which matches DTS CD 14 format.
19 PABS	Enable preamble search Enable Preamble Pa,Pb search
18-16 —	reserved
15-14 RX_DATA_FMT	Indicates format of data stored in memory. • 00 - Store complete sub frame • 01 - Store 24 audio data bits of sub-frame. • 10 - Store 20 audio data bits of sub-frame. • 11 - Store 16 audio data bits of sub-frame.
13 DIS_B_PRE_ERR_CHK	RX_DATAPATH: Disable B preamble error check. This bit when enabled, disables dropping frames of data once a bad B preamble is received, • 0 - A new B preamble will clear the pipe and wait for the following B preamble. • 1 - A new B preamble will start receiving a new block of data.
12 CLR_RX_FIFO	Clear Receive FIFO Clear receive FIFO. This is a self clear register bit.
11 CSA	Channel Status Acknowledge channel status acknowledge that should be set once channel status is processed.
10 UDA	User Data Acknowledge User data acknowledge that should be set once user data is processed.
9	Channel Status reset

Table continues on the next page...

Field	Description
CSR	Reset channel status gathering and storage. This is an active high reset.
8 UDR	User data reset Reset user data gathering and storage. This is an active high reset.
7 EN_PARITY_CALC	Enable Parity calculation. This bit enables parity calculation. • 0 - Do not calculate Parity for un-compressed data • 1 - Calculate Parity for un-compressed data
6 STORE_FMT	Receive Data store format. This bit describes the format to store the received data in memory. • 0 - Store data with bit 0 of subframe at bit position 31 in sub-frame. • 1 - Store data with bit 0 of subframe at bit position 0 in sub-frame.
5 RST_PKT_CNT_FIFO	Resets the packet count fifo. This bit when enabled, resets the preamble match packet number FIFO • 0 - Do not reset the FIFO • 1 - Reset FIFO.
4 EN_COMP_PARITY_CALC	RX_DATAPATH: Enable Compressed mode Parity calculation. This bit enables parity calculation for compressed mode. • 0 - Do not calculate Parity for compressed data • 1 - Calculate Parity for compressed data
3 ECC_VUC_BITS_EN	RX_DATAPATH: Enable VUC bit replacement after ECC correction. This bit enables VUC replacement for compressed mode with ECC corrected data • 0 - Do not replace VUC bits after ECC correction • 1 - Repalce VUC bits after ECC correction
2-1 —	reserved
0 PAPB_FIFO_STATUS	This bit is an empty fifo indicator for PaPb match.

14.6.4.1.17 Channel staus bits (RX_CS_DATA_BITS_0 - RX_CS_DATA_BITS_5)

Holds the 192 bit channel status for the last received block. It is only updated if M0+ acknowledges reading these registers using RX_DATAPATH_CTRL.CSA.

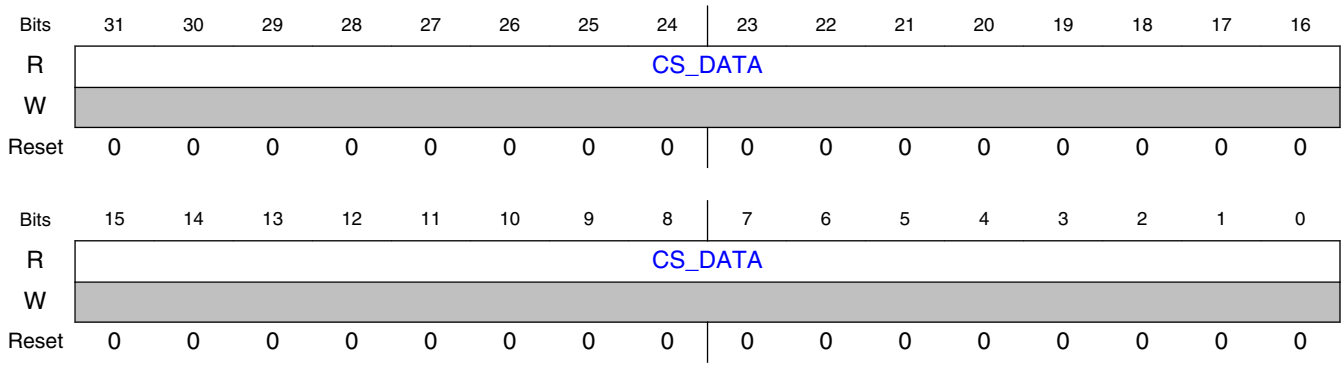
14.6.4.1.17.1 Offset

Register	Offset
RX_CS_DATA_BITS_0	190h
RX_CS_DATA_BITS_1	194h
RX_CS_DATA_BITS_2	198h
RX_CS_DATA_BITS_3	19Ch

Table continues on the next page...

Register	Offset
RX_CS_DATA_BITS_4	1A0h
RX_CS_DATA_BITS_5	1A4h

14.6.4.1.17.2 Diagram



14.6.4.1.17.3 Fields

Field	Description
31-0 CS_DATA	Channel Status bits 192 bit channel status / block

14.6.4.1.18 User data bits (RX_USER_DATA_BITS_0 - RX_USER_DATA_BITS_5)

Holds the 192 bit user data for the last received block. It is only updated if M0+ acknowledges reading these registers using RX_DATAPATH_CTRL.UDA.

14.6.4.1.18.1 Offset

Register	Offset
RX_USER_DATA_BITS_0	1A8h
RX_USER_DATA_BITS_1	1ACh
RX_USER_DATA_BITS_2	1B0h
RX_USER_DATA_BITS_3	1B4h

Table continues on the next page...

Register	Offset
RX_USER_DATA_BITS_4	1B8h
RX_USER_DATA_BITS_5	1BCh

14.6.4.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	U_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	U_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.18.3 Fields

Field	Description
31-0 U_DATA	User data bits 192 bit user data / block

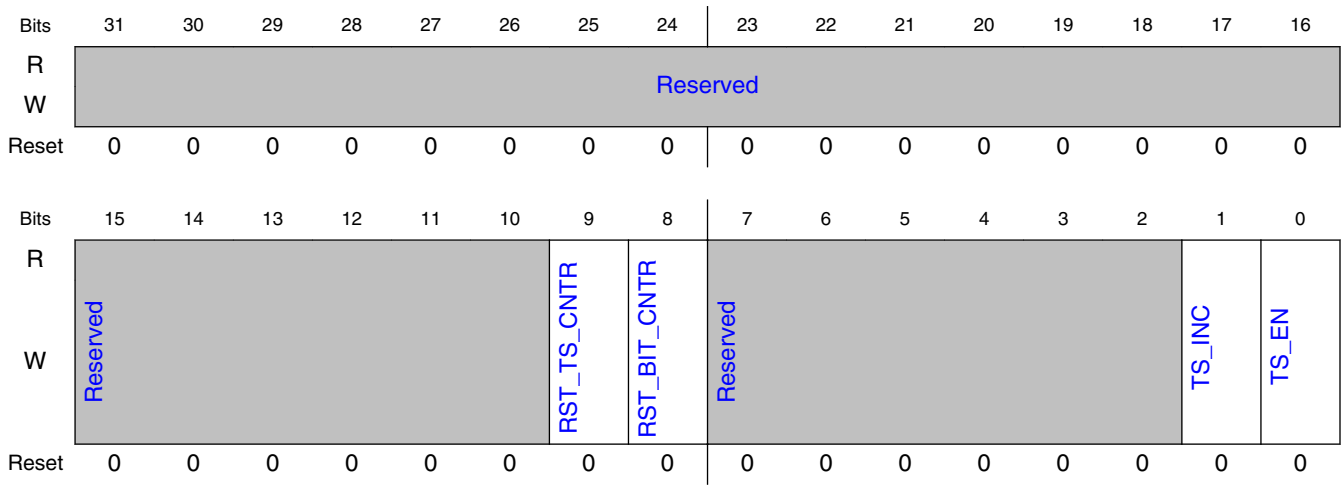
14.6.4.1.19 DMAC counter control register (RX_DPATH_CNTR_CTRL)

Controls the functioning of the DMAC internal counters for time stamp and bit count.

14.6.4.1.19.1 Offset

Register	Offset	Description
RX_DPATH_CNTR_CTRL	1C0h	DMAC counter control register
RX_DPATH_CNTR_CTRL_SET	1C4h	Writing a 1 to a bit in this register sets the corresponding bit in RX_DPATH_CNTR_CTRL
RX_DPATH_CNTR_CTRL_CLR	1C8h	Writing a 1 to a bit in this register clears the corresponding bit in RX_DPATH_CNTR_CTRL
RX_DPATH_CNTR_CTRL_TOG	1CCh	Writing a 1 to a bit in this register toggles the corresponding bit in RX_DPATH_CNTR_CTRL

14.6.4.1.19.2 Diagram



14.6.4.1.19.3 Fields

Field	Description
31-10 —	reserved
9 RST_TS_CNTR	Reset timestamp counter. • 0 - Timestamp counter is not reset. • 1 - Timestamp counter is reset.
8 RST_BIT_CNTR	Reset bit counter. • 0 - Bit counter is not reset. • 1 - Bit counter is reset.
7-2 —	reserved
1 TS_INC	Timestamp Increment • 0 - Timestamp counter starts to increment when enabled and the bit counter has incremented. • 1 - Timestamp counter increments when enabled.
0 TS_EN	Timestamp counter enable • 0 - Timestamp counter is disabled • 1 - Timestamp counter is enabled.

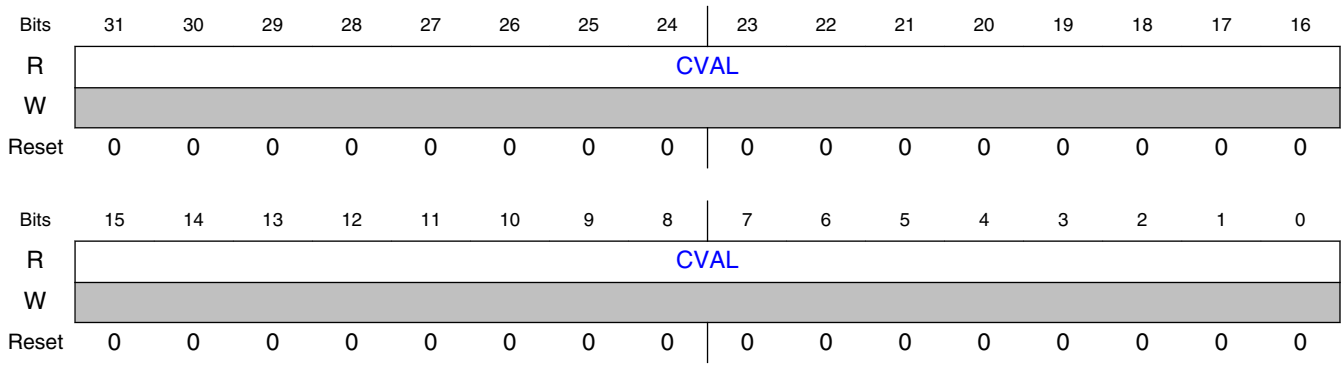
14.6.4.1.20 Receive Datapath Timestamp Counter Register (RX_DPATH_TSCR)

Current value of the timestamp counter.

14.6.4.1.20.1 Offset

Register	Offset
RX_DPATH_TSCR	1D0h

14.6.4.1.20.2 Diagram



14.6.4.1.20.3 Fields

Field	Description
31-0 CVAL	Timestamp counter value This counter is incremented based on the timestamp control register

14.6.4.1.21 Receive Datapath Bit counter register (RX_DPATH_BCR)

This register returns the current value of the bit counter.

14.6.4.1.21.1 Offset

Register	Offset
RX_DPATH_BCR	1D4h

14.6.4.1.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CVAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CVAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.21.3 Fields

Field	Description
31-0 CVAL	Bit count value The bit counter increments based on a bit count valid sgnal from the audio clock divider when eARC is active.

14.6.4.1.22 Receive datapath Bit count timestamp register. (RX_DPATH_BCTR)

Returns the value of the timestamp counter that corresponds to the value last update of the bit count register.

14.6.4.1.22.1 Offset

Register	Offset
RX_DPATH_BCTR	1D8h

14.6.4.1.22.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCT_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCT_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.22.3 Fields

Field	Description
31-0 BCT_VAL	Bit count timestamp value This counter is updated every time the the bit counter is updated.

14.6.4.1.23 Receive datapath Bit read timestamp register. (RX_DPATH_BCRR)

Returns the value of the timestamp counter that corresponds to the value when bit counter was last read.

14.6.4.1.23.1 Offset

Register	Offset
RX_DPATH_BCRR	1DCh

14.6.4.1.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCT_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCT_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.23.3 Fields

Field	Description
31-0 BCT_VAL	Bit count timestamp value This counter is updated every time the the bit counter is read.

14.6.4.1.24 Preamble match value register (DMAC_PRE_MATCH_VAL)

Holds the values for IEC61937 Preamble match.

14.6.4.1.24.1 Offset

Register	Offset	Description
DMAC_PRE_MATCH_VAL	1E0h	Preamble match value register
DMAC_PRE_MATCH_VAL_SET	1E4h	Writing a 1 to a bit in this register sets the corresponding bit in DMAC_PRE_MATCH_VAL
DMAC_PRE_MATCH_VAL_CLR	1E8h	Writing a 1 to a bit in this register clears the corresponding bit in DMAC_PRE_MATCH_VAL
DMAC_PRE_MATCH_VAL_TOG	1ECh	Writing a 1 to a bit in this register toggles the corresponding bit in DMAC_PRE_MATCH_VAL

14.6.4.1.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PA_VAL															
Reset	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PB_VAL															
Reset	1	1	1	1	1	0	0	0	0	1	1	1	0	0	1	0

14.6.4.1.24.3 Fields

Field	Description
31-16 PA_VAL	Preamble PA value Holds the PA value for PA PB match.
15-0 PB_VAL	Preamble PB value Holds the PB value for PA PB match.

14.6.4.1.25 Preamble match value register (DMAC_DTS_PRE_MATCH_VAL)

Holds the values for DTS CD 14 IEC61937 Preamble match.

14.6.4.1.25.1 Offset

Register	Offset	Description
DMAC_DTS_PRE_MATCH_VAL	1F0h	Preamble match value register
DMAC_DTS_PRE_MATCH_VAL_SET	1F4h	Writing a 1 to a bit in this register sets the corresponding bit in DMAC_DTS_PRE_MATCH_VAL
DMAC_DTS_PRE_MATCH_VAL_CLR	1F8h	Writing a 1 to a bit in this register clears the corresponding bit in DMAC_DTS_PRE_MATCH_VAL
DMAC_DTS_PRE_MATCH_VAL_TOG	1FCh	Writing a 1 to a bit in this register toggles the corresponding bit in DMAC_DTS_PRE_MATCH_VAL

14.6.4.1.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DTS_PA_VAL															
Reset	0	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DTS_PB_VAL															
Reset	1	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0

14.6.4.1.25.3 Fields

Field	Description
31-16 DTS_PA_VAL	Preamble PA value Holds the PA value for PA PB match.
15-0 DTS_PB_VAL	Preamble PB value Holds the PB value for PA PB match.

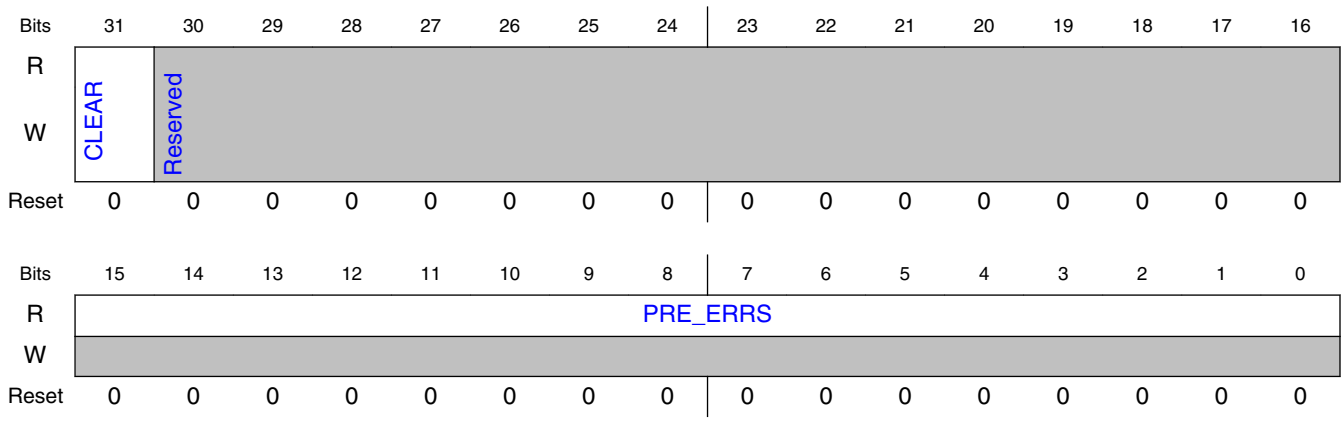
14.6.4.1.26 Error count for IEC60958-1 Block Synchronization. (RX_DPATH_PRE_ERR)

Error count for IEC60958-1 Block Synchronization. It is incremented when an unexpected B,M or W preamble is detected. Writing to bit 31 will clear this register

14.6.4.1.26.1 Offset

Register	Offset
RX_DPATH_PRE_ERR	200h

14.6.4.1.26.2 Diagram



14.6.4.1.26.3 Fields

Field	Description
31 CLEAR	Clear bit for error counter. When this bit is set, the preamble error counter will be reset.
30-16 —	reserved
15-0 PRE_ERRS	Preamble Error counter This counter is incremented when an unexpected Preamble is detected

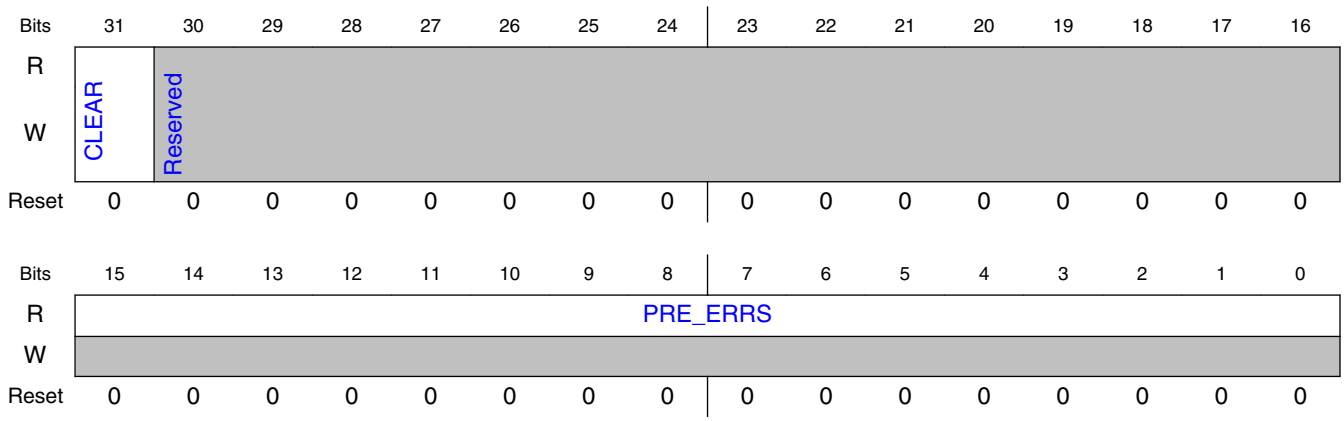
14.6.4.1.27 Parity Error count for IEC60958-1 Blocks. (RX_DPATH_PARITY_ERR)

Parity error count for IEC60958-1 Block Synchronization. It is incremented when a parity error is received. Writing a 1 to bit 31 will clear this register. SW will have to clear bit 31 after writing to it.

14.6.4.1.27.1 Offset

Register	Offset
RX_DPATH_PARITY_ERR	204h

14.6.4.1.27.2 Diagram



14.6.4.1.27.3 Fields

Field	Description
31 CLEAR	Clear bit for error counter. When this bit is set, the preamble error counter will be reset.
30-16 —	reserved
15-0 PRE_ERRS	Preamble Error counter This counter is incremented when an unexpected Preamble is detected

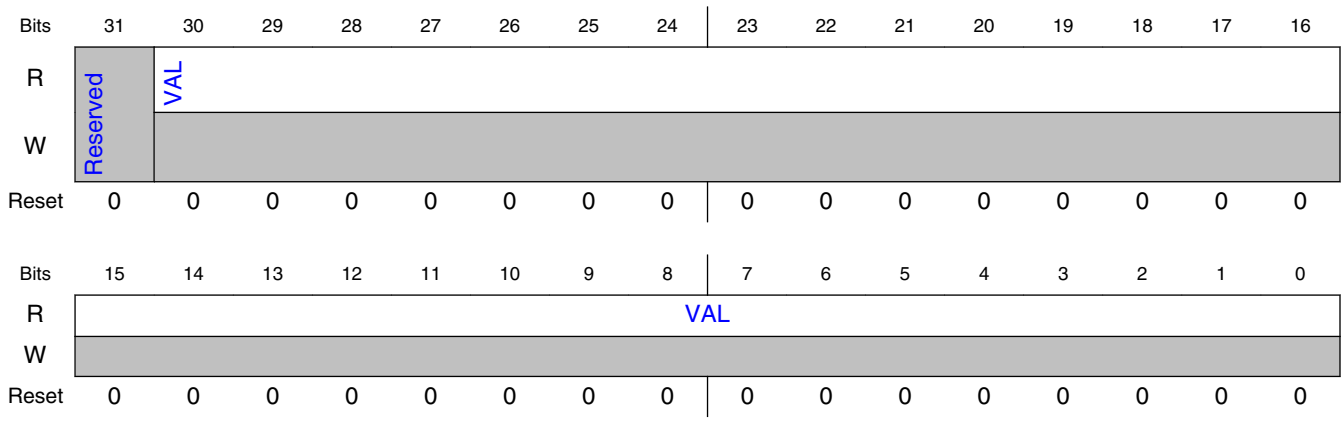
14.6.4.1.28 Receive Data packet count. (RX_DPATH_PKT_CNT)

This register provides the Receive data packet count.

14.6.4.1.28.1 Offset

Register	Offset
RX_DPATH_PKT_CNT	210h

14.6.4.1.28.2 Diagram



14.6.4.1.28.3 Fields

Field	Description
31 —	Reserved
30-0 VAL	Data packet counter This status provides the number of received packets.

14.6.4.1.29 Receive Data packet Corrected error count. (RX_DPATH_ONE_BIT_ERR_CNT)

This register provides the Receive data single bit BCH error count.

14.6.4.1.29.1 Offset

Register	Offset
RX_DPATH_ONE_BIT_ERR_CNT	214h

14.6.4.1.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.29.3 Fields

Field	Description
31-16 —	Reserved
15-0 VAL	This status provides the number of single bit BCH errors corrected.

14.6.4.1.30 Preamble match offset value register (DMAC_PRE_MATCH_OFFSET)

When an IEC61937 burst preamble is matched the current sample count shall be committed to a FIFO read from this register.

14.6.4.1.30.1 Offset

Register	Offset
DMAC_PRE_MATCH_OFFSET	218h

14.6.4.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PA_OFFSET															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PA_OFFSET															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.30.3 Fields

Field	Description
31-0 PA_OFFSET	Sample count value for PA offset match This reflects the value of a 32 bit counter that will count all samples received.

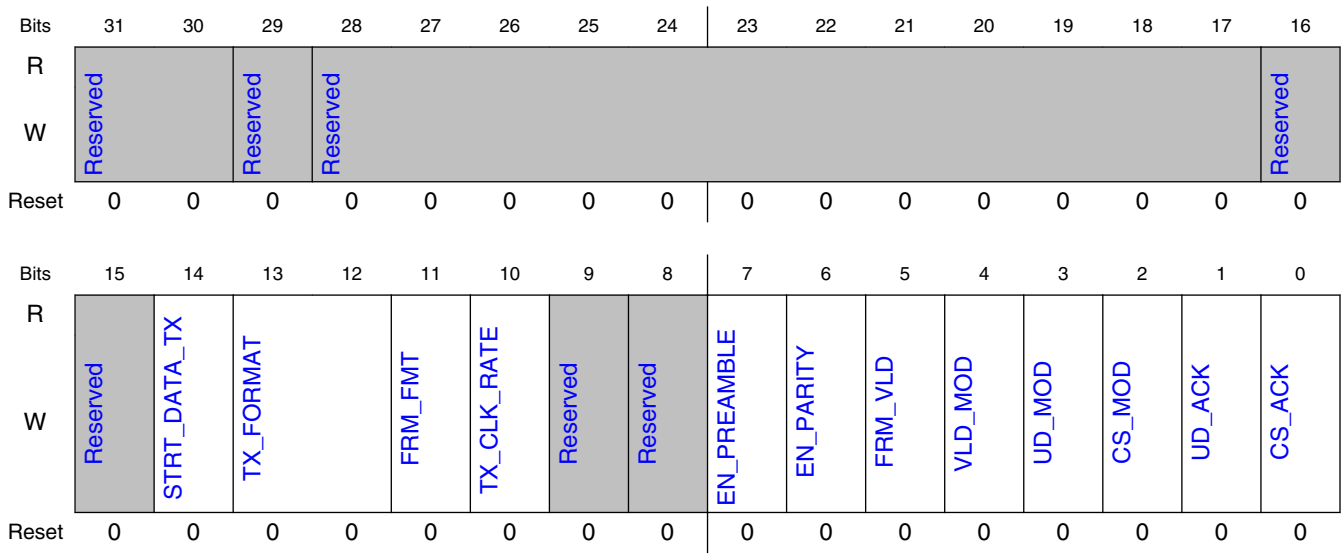
14.6.4.1.31 Transmit Data path control register (TX_DATAPATH_CTRL)

Configures and controls operations for the transmit of 60958 audio block.

14.6.4.1.31.1 Offset

Register	Offset	Description
TX_DATAPATH_CTRL	220h	Transmit Data path control register
TX_DATAPATH_CTRL_SET	224h	Writing a 1 to a bit in this register sets the corresponding bit in TX_DATAPATH_CTRL
TX_DATAPATH_CTRL_CLR	228h	Writing a 1 to a bit in this register clears the corresponding bit in TX_DATAPATH_CTRL
TX_DATAPATH_CTRL_TOG	22Ch	Writing a 1 to a bit in this register toggles the corresponding bit in TX_DATAPATH_CTRL

14.6.4.1.31.2 Diagram



14.6.4.1.31.3 Fields

Field	Description
31-30 —	reserved
29 —	Reserved
28-17 —	reserved
16-15 —	reserved
14 STRT_DATA_TX	Once Comma pattern is successively received, and heartbeat is detected, start TX of DMAC data. • 0 - Even though TX FIFO has data, do not transmit it right now. • 1 - As soon as TX_FIFO has data, enable DMAC TX datapath.
13-12 TX_FORMAT	Transmit data format This field describes the number of bits to be transmitted / audio sub-frame. Each FIFO entry only represents one sub-frame worth of data. Bit reversal is indicated by TX_DATAPATH_CTRL.FRM_FMT • 0 - 32 bit data format. The audio data will be positioned within the subframe by SW. • 1 - 24 bit input data stored in bits 23:0 of TX FIFO • 2 - 20 bit input data stored in bits 19:0 of TX FIFO • 3 - 16 bit input data stored in bits 15:0 of TX FIFO
11 FRM_FMT	Frame format of input data • 0 - Frame data will be considered to be aligned to the MSB. i.e, 60958 frame bit 0 is at bit position 31 of the TX data FIFO. • 1 - Output 60958 frame format data will be considered to be aligned to the LSB. i.e, 60958 frame bit 0 is at bit position 0 of the TX data FIFO.
10	This bit controls the TX clock rate.

Table continues on the next page...

Field	Description
TX_CLK_RATE	If this bit is set, TX datapath clock should be = 2* TX bit rate. If this bit is not set, TX datapath clock should be 10* TX bit rate.
9 —	reserved
8 —	Reserved
7 EN_PREAMBLE	Enable preamble insertion If this bit is set, Preamble matching the frame count will be inserted in the TX data path.
6 EN_PARITY	Enable parity insertion If this bit is set, parity will be calculaed and inserted in the TX data path.
5 FRM_VLD	Valid bit value If VLD_MOD bit is set, Valid bit will be replaced in the TX data path with the value in this bit.
4 VLD_MOD	Enable Valid bit insertion If this bit is set, Valid bit will be inserted in the TX data path from the FRM_VLD register bit.
3 UD_MOD	Enable User Data insertion If this bit is set, User data will be inserted in the TX data path from the UD buffer.
2 CS_MOD	Enable Channel Status insertion If this bit is set, Channel status will be inserted in the TX data path from the CS buffer.
1 UD_ACK	User Data ACK Once the user data registers are double buffered, set this ack to let the host know that new values can be written in the UD buffer. Once the Buffer is updated, host should clear this value.
0 CS_ACK	Channel Status ACK Once the Channel status registers are double buffered, set this ack to let the host know that new values can be written in the CS buffer. Once the Buffer is updated, host should clear this value.

14.6.4.1.32 Channel staus bits (TX_CS_DATA_BITS_0 - TX_CS_DATA_BITS_5)

Holds the 192 bit channel status for the current block to transmit. It is updated only when format change needs to be communicated. It can be updated in the time when no data sub-frames will be added to the TX fifo.

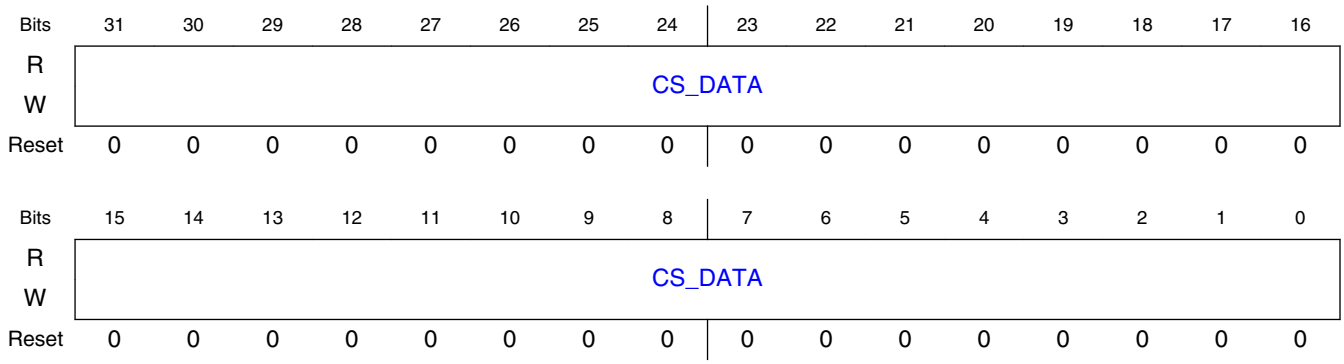
14.6.4.1.32.1 Offset

Register	Offset
TX_CS_DATA_BITS_0	230h
TX_CS_DATA_BITS_1	234h
TX_CS_DATA_BITS_2	238h
TX_CS_DATA_BITS_3	23Ch

Table continues on the next page...

Register	Offset
TX_CS_DATA_BITS_4	240h
TX_CS_DATA_BITS_5	244h

14.6.4.1.32.2 Diagram



14.6.4.1.32.3 Fields

Field	Description
31-0 CS_DATA	Channel Status bits / block 192-bit channel status field / block of audio data.

14.6.4.1.33 User data bits (TX_USER_DATA_BITS_0 - TX_USER_DATA_BITS_5)

Holds the 192 bit user data for the current block to transmit. It can be updated / block of data after the acknowledge field TX_DATAPATH_CTRL.UD_ACK is cleared. Setting this field indicates new user data block is available.

14.6.4.1.33.1 Offset

Register	Offset
TX_USER_DATA_BITS_0	248h
TX_USER_DATA_BITS_1	24Ch
TX_USER_DATA_BITS_2	250h

Table continues on the next page...

Register	Offset
TX_USER_DATA_BITS_3	254h
TX_USER_DATA_BITS_4	258h
TX_USER_DATA_BITS_5	25Ch

14.6.4.1.33.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	U_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	U_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.33.3 Fields

Field	Description
31-0 U_DATA	User data bits/block 192-bit User Data field / block of audio data.

14.6.4.1.34 DMAC counter control register (TX_DPATH_CNTR_CTRL)

Controls the functioning of the DMAC internal counters for time stamp and bit count.

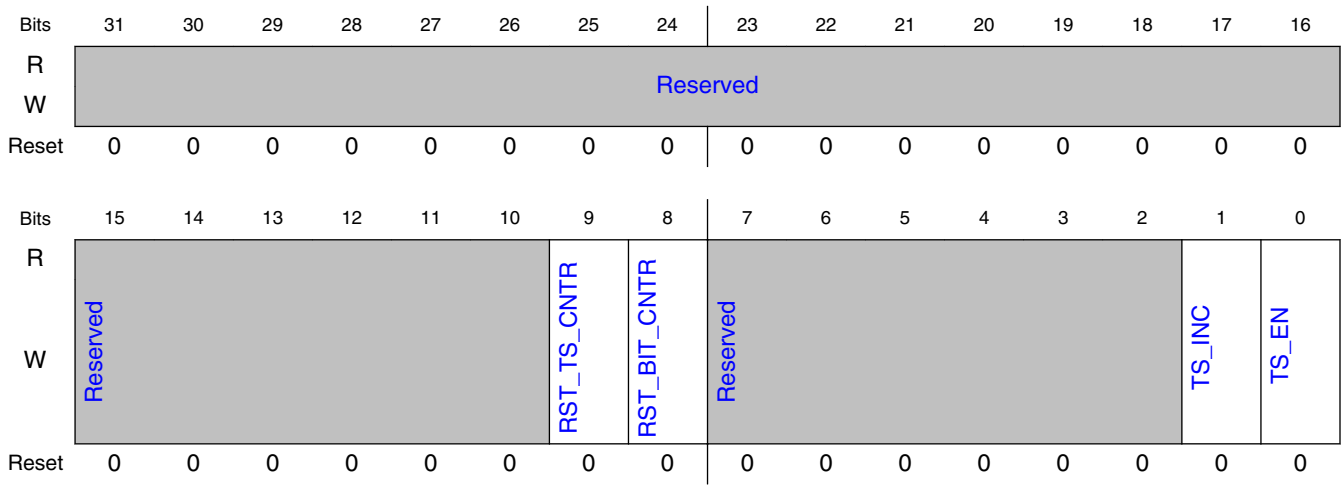
14.6.4.1.34.1 Offset

Register	Offset	Description
TX_DPATH_CNTR_CTRL	260h	DMAC counter control register
TX_DPATH_CNTR_CTRL_SET	264h	Writing a 1 to a bit in this register sets the corresponding bit in TX_DPATH_CNTR_CTRL
TX_DPATH_CNTR_CTRL_CLR	268h	Writing a 1 to a bit in this register clears the corresponding bit in TX_DPATH_CNTR_CTRL

Table continues on the next page...

Register	Offset	Description
TX_DPATH_CNTR_CTRL_TOG	26Ch	Writing a 1 to a bit in this register toggles the corresponding bit in TX_DPATH_CNTR_CTRL

14.6.4.1.34.2 Diagram



14.6.4.1.34.3 Fields

Field	Description
31-10 —	reserved
9 RST_TS_CNTR	Reset timestamp counter. • 0 - Timestamp counter is not reset. • 1 - Timestamp counter is reset.
8 RST_BIT_CNTR	Reset bit counter. • 0 - Bit counter is not reset. • 1 - Bit counter is reset.
7-2 —	reserved
1 TS_INC	Timestamp Increment • 0 - Timestamp counter starts to increment when enabled and the bit counter has incremented. • 1 - Timestamp counter increments when enabled.
0 TS_EN	Timestamp counter enable • 0 - Timestamp counter is disabled • 1 - Timestamp counter is enabled.

14.6.4.1.35 Transmit Datapath Timestamp Counter Register (TX_DPATH_TSCR)

Returns the current value of the timestamp counter. This counter is incremented based on the timestamp control register

14.6.4.1.35.1 Offset

Register	Offset
TX_DPATH_TSCR	270h

14.6.4.1.35.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CVAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CVAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.35.3 Fields

Field	Description
31-0 CVAL	Timestamp counter value

14.6.4.1.36 Transmit Datapath Bit counter register (TX_DPATH_BCR)

This register returns the current value of the bit counter. The bit counter increments based on a bit count signal from the audio clock divider when eARC is active.

14.6.4.1.36.1 Offset

Register	Offset
TX_DPATH_BCR	274h

14.6.4.1.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CVAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CVAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.36.3 Fields

Field	Description
31-0 CVAL	Bit count value

14.6.4.1.37 Transmit datapath Bit count timestamp register. (TX_DPATH_BCTR)

Returns the value of the timestamp counter that corresponds to the value last update of the bit count register.

14.6.4.1.37.1 Offset

Register	Offset
TX_DPATH_BCTR	278h

14.6.4.1.37.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCT_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCT_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.37.3 Fields

Field	Description
31-0 BCT_VAL	Bit count timestamp value This counter is updated every time the the bit counter is updated.

14.6.4.1.38 Transmmit datapath Bit read timestamp register. (TX_DPATH_BCRR)

Returns the value of the timestamp counter that corresponds to the value when bit counter was last read.

14.6.4.1.38.1 Offset

Register	Offset
TX_DPATH_BCRR	27Ch

14.6.4.1.38.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BCT_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCT_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.6.4.1.38.3 Fields

Field	Description
31-0 BCT_VAL	Bit count timestamp value This counter is updated every time the the bit counter is read.

14.6.4.1.39 HPD Debounce Control Register (HPD_DBNC_CTRL)

This register implements control for functions we are not clear about.

14.6.4.1.39.1 Offset

Register	Offset	Description
HPD_DBNC_CTRL	2A0h	HPD Debounce Control Register
HPD_DBNC_CTRL_SET	2A4h	Writing a 1 to a bit in this register sets the corresponding bit in HPD_DBNC_CTRL
HPD_DBNC_CTRL_CLR	2A8h	Writing a 1 to a bit in this register clears the corresponding bit in HPD_DBNC_CTRL
HPD_DBNC_CTRL_TOG	2ACh	Writing a 1 to a bit in this register toggles the corresponding bit in HPD_DBNC_CTRL

14.6.4.1.39.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL															
Reset	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0

14.6.4.1.39.3 Fields

Field	Description
31-0 VAL	HDP pin debounce interval Specifies the decounce interval on the HPD pin. If set, HW will wait until the timer is up before asserting HPD toggle interrupt to the core. If set to 0, this counter will be ignored. Reset value of the counter is used to support the PHY requirement of waiting for 500 us after HPD before it is enabled.

14.7 Audio DSP (HiFi 4 DSP)

14.7.1 Overview

The HiFi 4 Audio Engine is a highly optimized audio processor geared for efficient execution of audio and voice codecs and pre- and post-processing modules.

The HiFi 4 Audio Engine consists of three main components: a DSP subsystem, floating point unit, and a subsystem to assist with bit stream access and variable-length (Huffman) encoding and decoding.

The HiFi 4 Audio Engine, a SIMD (single-instruction/multiple-data) processor, has the ability to work in parallel on two 32-bit data items or four 16-bit data items. For example, it allows for one operation to perform two 32-bit additions in parallel, with each addition occupying half of a 64-bit AE_DR register.

The HiFi 4 multipliers support multiplication of four, 32-bit operands per cycle. There are operations for single, dual, and quad multiplication. Single or dual multiply operations can be dual issued using VLIW instructions. Special 32x16 and 16x16-bit dot-

product instructions can be dual issued, allowing for a peak of eight multiply operations per cycle. Quad multiply operations cannot otherwise be issued together with other multiplies.

The HiFi 4 Audio Engine can only be configured to use a little-endian byte ordering.

In general, 16-bit support is geared towards efficient support of the ITU-T/ETSI intrinsic model, while 32x16-bit and 32-bit support is provided for both integer and fixed-point computation.

HiFi 4 is a Very-Long Instruction Word (VLIW) architecture, supporting the execution of four operations in parallel. DSP loads, bit-stream and Huffman operations, and core operations are available in slot 0 and 1 of a VLIW instruction. Stores are available only in slot 0. DSP MAC (multiplier–accumulator), ALU, and optional floating point operations are typically available in slots 2 and 3.

HiFi 4 supports either caches or local memories.

14.7.1.1 Block Diagram

The following diagram illustrates the main custom state, register file, and execution units added by the HiFi 4 Audio Engine.

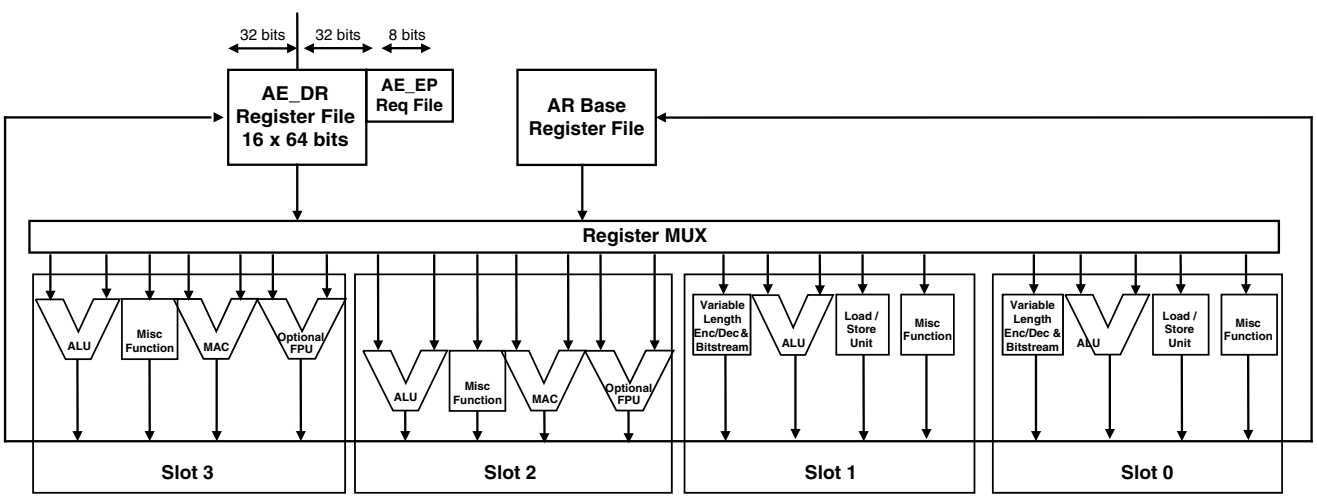


Figure 14-42. HiFi 4 Components

The main hardware resources in the DSP subsystem are two 2/4-multiplier multiply/ accumulate units, an option for two copies of a 2-way SIMD single-precision IEEE floating point unit, a 16-entry register file AE_DR to hold 64-bit, pairs of 32-bit, or quads

of 16-bit data items, an arithmetic/logic unit, an 8-entry register file, AE_EP, to hold eight extra guard bits for AE_DR registers and a shift unit to operate on the AE_DR values.

The multiplier units support four 32x32, up to eight 16x32 or 16x16-bit MACs per cycle. The multiplies are supported through single-instruction quad multiplies or through parallel-issued dual, single, or quad multiply instructions.

The load/store unit is capable of loading or storing up to two 24-bit or 32-bit SIMD elements, four 16-bit SIMD elements, or single elements up to 64 bits in size. 24-bit data can either be contained inside 32-bit envelopes, or can be packed together into 24 bits of memory. Eight packed elements can be loaded or stored in three instructions. The load/store unit supports unaligned accesses, whereby a stream is first primed and afterwards 64 unaligned bits can be loaded or stored in every cycle.

The DSP subsystem can be issued in several VLIW formats. A few formats are encoded in 88 bits and support up to four slots. A few formats are encoded in 48 bits and can support up to two slots in each execution cycle, zero or one operation from each slot can be executed independently according to the static bundling expressed in the machine code.

14.7.1.2 Features

The HiFi 4 Audio Engine includes the following features:

- Four 32x32-bit MAC per cycle (2.7 GMACs)
- Instruction cache, Data cache and prefetching supported
- Instruction and Data Tightly Coupled Memories
- Access to external SDRAM
- 128-bit local AXI bus
- Local access to audio interfaces (eSAI, SAI, SPDIF) to reduce latency
- Support for EC/NR, Audio Codecs, Audio/Voice enhancements algorithms

14.7.2 Functional Description

The sections below describe the HiFi 4 functionality and architecture.

14.7.2.1 HiFi 4 Configuration

The table below provides the configuration details for HiFi 4:

Table 14-14. HiFi 4 Configuration Details

Feature	Value
DSP Frequency	800 MHz at overdrive voltage 600 MHz at nominal voltage
Instruction Cache size	32 KB
Data Cache size	48 KB
Prefetch supported	Yes
Instruction RAM TCM	2 KB
Data RAM TCM	64 KB
AXI Bus width	128-bit
On-Chip RAM (OCRAM)	256 KB

14.7.2.2 Instruction Set Architecture (ISA)

The table below provides the ISA details for HiFi 4:

Table 14-15. HiFi 4 Instruction Set Architecture Details

	HiFi 4
Instruction Set Architecture (ISA)	V32 (vf)
VLIW Slots	4
VLIW bundle Instruction Sizes	48 / 88-bit
Fixed Point MACs per Cycle – 32x32	4
Fixed Point MACs per Cycle – 24x24	4
Fixed Point MACs per Cycle – 16x16	8
Accumulator	64 / 72-bit
Circular buffer support	2
Load/store units	2

14.7.2.3 HiFi 4 Instruction Set Overview

A wide variety of load/store operations support multiple addressing modes, with support for 16/24/32-bit scalar and vector data types together with 56/64-bit scalar. Vector data management is supported with select operations and shifting.

Multiply operations include 32x32-bit, 32x16-bit, and 16x16-bit. Multiply operations come in fixed-point and integer variants. They come in high precision and low precision variants. High-precision multiplies utilize a 64-bit accumulator and in some cases an additional 8-bit extended precision accumulator. Since an accumulator can only hold one result, HiFi 4 supports dual multiplies where the results of two multiplies are added or subtracted together before being added into the accumulator.

Low-precision multiplies accumulate in 32bits or 16 bits. Since each register can hold two, 32-bit accumulators (or four 16-bit accumulators), these instructions can perform two or four independent SIMD multiplies.

A set of bit-stream and variable length instructions allow for efficient access of serial bit streams, including Huffman encode and decode.

The optional floating point unit supports two 2-way SIMD units of IEEE-754 single-precision floating point operations.

14.7.2.4 AE_DR Register

The HiFi 4 Audio Engine contains a 16-entry, 64-bit register file, AE_DR. Each register can hold one or two, 24- or 32-bit operands, one or four 16-bit operands or one 56- or 64-bit operand as shown in Figure 2-1. 24-bit and 56-bit operands are sign extended to fill their 32- or 64-bit container. The separate halves or quarters of the register are always separate data items. For example, if you shift a 32-bit element to the left, the L element does not spill over into the high element.

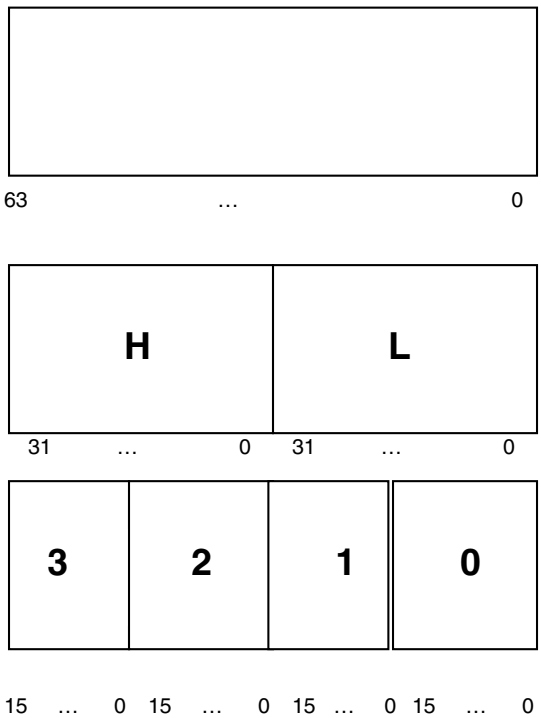


Figure 14-43. AE_DR Register

When a register is stored to memory, the high half of the register is always stored in the lower memory address. Operations that access individual 24- or 32-bit elements of AE_DR registers refer to the elements with selectors L and H in the mnemonics. Operations that access individual 16-bit elements refer to the elements with sectors 3, 2, 1 and 0 in the mnemonics.

14.7.2.5 AE_EP Register

HiFi 4 supports an 8-entry, 8-bit extra precision register, AE_EP. This register is used in conjunction with some multiply, add, shift, and saturate instructions to provide a total of 72 bits of precision. This register is accessed one cycle later than the AE_DR register. Accumulations can still happen every cycle, but there is an extra cycle of delay when trying to shift or saturate the extra accumulator back down to 64 bits.

14.7.2.6 Prefetching

HiFi 4 includes a prefetch option geared for systems with long memory latency. When the HiFi 4 processor detects a positive stride-1 stream of cache misses (either data or instruction), it can speculatively prefetch ahead up to four cache lines and place them in a

buffer close to the processor, or on the data side optionally into the L1 data cache (there is no support for prefetching directly into the L1 instruction cache). In addition, prefetch instructions can be manually issued.

Hardware prefetching is enabled by default. By default, on configurations that support it, data prefetches are placed into the L1 data cache. HAL calls can be used to explicitly disable prefetching or to increase its aggressiveness in different sections of the code. With more aggressive prefetching, the hardware will prefetch earlier when detecting a stream and will prefetch more lines ahead. With aggressive prefetch, the system will require more bandwidth. Prefetching instructions and data can be controlled separately.

14.7.2.7 VLIW Slots and Formats

HiFi 4 can issue up to four operations in a single 88-bit instruction bundle or two operations in a 48-bit bundle. HiFi 4 supports several different formats with different slotting constraints. Every instruction belongs to one format but different formats may pack different numbers of operations in a single instruction.

In general, the first slot supports loads and stores and core operations, the second slot supports loads and core operations and the last two slots support ALU and multiply operations. One of the 48-bit formats uses a dummy second slot to allow multiply operations in the third nominal slot.

For the optional floating point unit, most floating point operations are available in the third and fourth slot, allowing the machine to issue, for example, two SIMD floating point loads in parallel with two, SIMD (2-way), multiply-accumulation operations, allowing for an aggregate of four multiply-accumulation operations per cycle.

A subset of the operations as well as all the bit-stream operations are available in a single issue, 24-bit format called Inst. The compiler will automatically use the 24-bit format when it is not possible (or beneficial) to bundle a relevant operation together with an operation that can go in another slot. Core instructions are all available in Inst or an even smaller 16-bit format.

14.7.2.8 Load and Store Operations

HiFi 4 supports loading and storing scalars or vectors of 16, 24, 32 and 64 bits. Each scalar load/store accesses 16, 24, 32, or 64 bits. Each vector accesses 64 bits or 48 bits for packed 24-bit data. For vector loads and stores, the high address in memory is always stored in the least significant bits in the register. This enables the same source code to work on both little and big endian systems.

Reverse vector loads and stores reverse the elements in a register so that the low address in memory is stored in the least significant bits in the register. This way, whether accessing data in a stride one or stride negative one fashion, the earliest data to be accessed is always in the same position in the register.

HiFi 4 also supports two circular buffers that can be used with either aligned or unaligned data.

14.7.2.9 Multiply and Accumulate Operations

The HiFi 4 ISA supports a rich collection of single, dual, and quad multiply/accumulate operations with different input and output precision, scaling, rounding and saturation modes. HiFi 4 supports four 32-bit or 16-bit multiplies per cycle. Eight 32x16-bit multiplies per cycle are supported but only using four way, dot-product instructions that can only be scheduled together with loads. Otherwise, four 32x16-bit multiplies can be performed per cycle. Individual operations perform one, two, or four multiplies. Single or dual-multiply operations can typically be dual-issued in a VLIW bundle.

Chapter 15
Graphics Processing Unit (GPU)

15.1 GPU Overview

15.1.1 Overview

This section provides a high-level overview of the Graphics Processing Unit (GPU) subsystem in the chip. This chip uses two GPU cores, one for 3D processing and the other to provide 2D and Composition acceleration.

15.1.1.1 Block Diagram

The high level block diagram for the GPU subsystem is shown below:

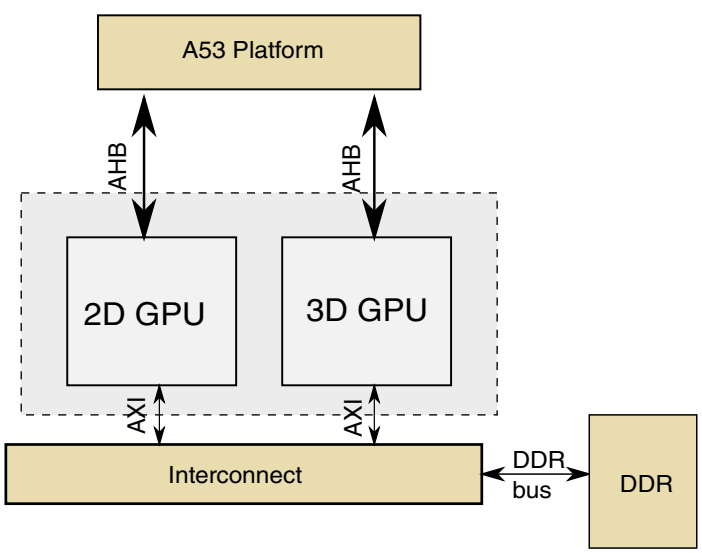


Figure 15-1. GPU Subsystem High Level Block Diagram

The 3D GPU and 2D GPU are designed in such a way that if only one core is required, the other can be gated off.

15.1.2 Features

The key features of the GPUs are described below:

- 3D GPU Core:
  - Two Shader Execution Units
  - Supports OpenGL ES 1.1, 2.0, 3.0, 3.1
  - Supports OpenCL 3.0
  - Supports OpenVG 1.1
  - Supports OpenGL 4.0
  - Supports EGL 1.5
  - Supports Vulkan 1.1
  - Supports tile format
- 2D GPU Core:
  - Supports multi-source composition
  - Supports one-pass filter
  - Supports tile format from 3D GPU

15.2 2D Graphics Processing Unit (GPU2D)

15.2.1 Overview

This 2D Graphics Processing Unit (GPU) is a high-performance, multi-pipe 2D raster graphics core that accelerates the 2D graphics display and provides advanced composition capabilities.

15.2.1.1 Block Diagram

The block diagram of the 2D GPU and the description of the functional blocks is provided below.

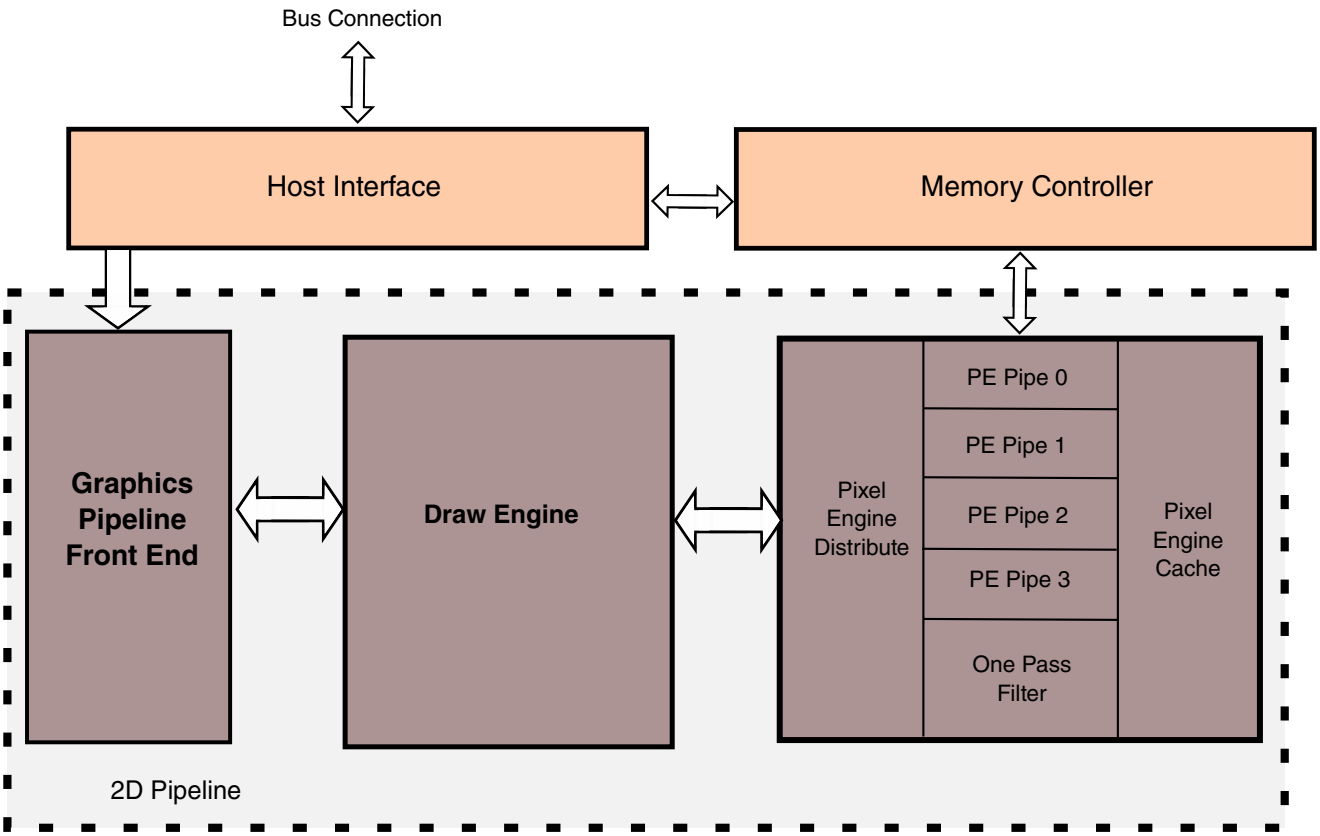


Figure 15-2. 2D GPU Block Diagram

- **Host Interface** - Allows the core to communicate with external memory and the CPU through the AXI or the AHB bus. In this block, data crosses clock domain boundaries.
- **Memory Controller** - Internal memory management unit that controls the block-to-host memory requests.
- **Graphics Pipeline Front End** - Inserts high level primitives and commands into the graphics pipeline.
- **2D Drawing and Scaling Engine** - Draws 2D graphics primitives and rasterizes 2D images.
- **Pixel Engine** - Pixel manipulation and filtering on rendered images. The GPU has four pixel pipes.

15.2.1.2 Features

The features of the GPU include:

- API Support:
  - NXP G2D API
- 2D GPU Unit:
  - Bit Blit

    - Stretch Blit
    - Rectangle fill and clear
    - Line drawing
    - Filter Blit
    - Mono expansion for text rendering
    - ROP2, ROP3 and ROP4
    - Alpha blending, including Java 2 Porter-Duff compositing blending rules
    - 32K x 32K coordinate system
    - 90 / 180 / 270 degree rotation
    - Transparency by monochrome mask, chroma key, or pattern mask
    - Support 2x2 in 4x4 tile format
    - A8 output with rotation in filter Blit and bit Blit
    - Src/Dest color key full bypass support
- Multi Source Blending:
    - Full support for Multi source blending with variable block size to improve BW and reduce SW overhead
    - Up to 8 sources are supported
    - Programmable block size guarantees cache efficiency so each source is read once and each destination is written once
    - Supports 90, 180, 270 degree rotation with different block size for higher cache efficiency
- YUV Support:
    - Full Multi destination support for converting non-planar YUV formats to planar YUV. It is used in extracting various components from input color into different destination planes
    - YUV422 output with alpha blending supported
- Clock disabling:
    - The core clock enters the frequency scaling block, and a clock with missing pulses is output, buffered, and sent to clock gating logic. The clock gate output will be used by the 2D GPU blocks. The core clock can be shut down through software by setting the proper register bits
- AXI Bus:
    - AXI 4-bit ID, such that all Read and Write transactions have a unique AXI ID
    - AXI out of order read return supported
- Additional Enhancements:
    - Full functional MMU with variable page size support
- Rotation:
    - 90° / 180° / 270° / Mirror rotation is supported for all primitives
    - Independent source rotation
- Transparency Mode:
    - For Monochrome Expansion:

    - Opaque
    - Conditional transparency - Transparent if the current pixel matches the specified value
  - For blits:
    - Opaque
    - Masked transparency - Transparent if the mask for the current pixel or pattern is zero
    - Source Conditional transparency - Transparent if the source pixel is within the specified value range
    - Destination Conditional transparency - Transparent if the destination pixel is not within the specified value range
- Clipping:
  - One clipping rectangle is supported for all bit blit primitives

15.2.1.2.1 GPU Host and Memory Interface Features

Table 15-1. Host and memory interface features

Feature	GPU Support
AHB interface	32-bit
Interface for External memory access	one 128-bit AXI
Virtual memory support	Yes
Out of order return support	Yes, supports out of order return for different clients
Code and data memory location restrictions	Unrestricted; arbitrary memory reads and writes
System address space	256 Kbytes (64K 32-bit words); however, the range of addresses used is only 4 Kbytes
Physical address	32 bits
Resource locks with CPU	Semaphore lock
Read write request size support	Memory controller supports 16, 32, 64, 128 byte read and write requests
MMU with variable page size support	MMU supports variable page sizes (4KB, 64KB, 1MB, 16MB)
GPU Cache Flush	Auto-flush at programmable intervals

15.2.1.2.2 GPU Power Management Features

Table 15-2. Power management features

Feature	Support
Low power CMOS technology compatible	Yes
Dynamic clock scaling	Yes
Automatic clock gating of flip flops and rams	Yes
Global clock gating of unused macro blocks	Yes
Active(ON), Idle, Standby and Sleep(OFF) Programmable Power Modes	Yes

15.2.1.2.3 Composition and 2D Hardware Features

This following table provides additional details on the hardware features supported by 2D GPU.

Table 15-3. Composition and 2D Hardware Features

Feature	Support
Programmable OPs	ROP2, ROP3, ROP4 full alpha blending and transparency
Fixed function	Line draw,Rectangle fill, Rectangle Clear, Bit blit, Stretch blit, Filter blit
Blit support	Copy (Bit), Filter, Monochrome Mask, Stretch/Shrink
Alpha blending modes	Java2 Porter-Duff, Chroma Key, Pattern Mask
Alpha blending during filter blit	Not supported
Transparency	Supported by monochrome mask, chroma key or pattern mask
Dithering	2D dither support
Image scaling	Filter Blit FIR resampling filter with up to 9 taps, One Pass Filter with 3/5 taps
Rotation	90 / 180 / 270 degrees with mirror
Clipping	one clipping rectangle supported for all bit blit primitives.
Text rendering	Monochrome expansion
Rendering size	32k x 32k raster 2D coordinate system
Compression	Lossless, tiled compression support

15.2.1.2.3.1 Source and Destination formats

The graphics engine supports the following source and destination formats for data, bit blits and filter blits. In addition to these source and destination RGB formats, their swizzle formats (ARGB, RGBA, ABGR, BGRA) are also supported. For YUV formats, the GPU supports their U/V swap formats.

Table key:

- Y = Yes
- N = No

- MD = Multi-Destinatin Support
- Yv = Only possible when vertical filter blit is done on YUV input

Table 15-4. Supported Source (Src) and Destination (Dest) formats

Format	Bit Blit Src	Bit Blit Dest	Stretch Blit Src	Stretch Blit Dest	Filter Blit Src	Filter Blit Dest	OPF Src	OPF Dest	Multi-Source Src	Multi-Source Dest
A1R5G5B5	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
A4R4G4B4	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
A8R8G8B8	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
X1R5G5B5	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
X4R4G4B4	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
X8R8G8B8	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
R5G6B5	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
A8	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
YUY2 (packed YUV422)	Y	Y	Y	N	Y	Yv	Y	Yv	Y	Y
UYVY (packed YUV422)	Y	Y	Y	N	Y	N	Y	Y	Y	Y
YV12 (planar YUV420)	Y	MD	Y	N	Y	N	Y	N	Y	N
NV12 (semi-planar YUV420)	Y	MD	Y	N	Y	N	Y	N	Y	N
NV16 (semi-planar YUV422)	Y	MD	Y	N	Y	N	Y	N	Y	N
8-bit color index	N	N	Y	N	Y	N	N	N	Y	N
1-bit monochrome	Y	N	N	N	N	N	N	N	N	N

15.2.2 Functional description

15.2.2.1 Composition and 2D Operations

15.2.2.1.1 Line

The LINE operation draws a line. The coordinates for two points are given - start point and end point and the end point is not drawn.

Lines are rendered using the Bresenham algorithm. The Bresenham algorithm has the advantage of using integer arithmetic and has no accumulation of rounding errors.

In the case of line, only ROP2 and ROP4 are supported. It operates on pattern and destination. The pattern should have a transparency mask in order to use ROP4.

Clipping is supported for lines on a per pixel basis.

the figure below shows the operation in more detail.

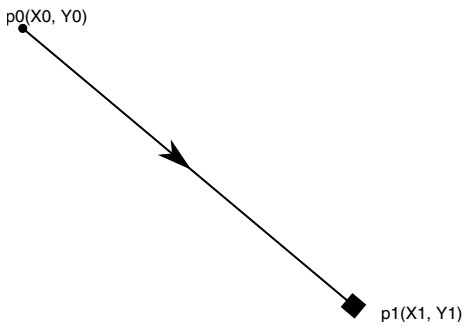


Figure 15-3. Line

15.2.2.1.2 Rectangle Fill and Clear

Rectangle fill suffuses a rectangle area with a given color. Essentially rectangle fill is a pattern fill, where an 8x8 pattern is initialized with the specified color. It supports ROP2 and ROP4 with the pattern and destination as its inputs. If ROP4 is used, the pattern should have a transparency mask.

Clear is similar to rectangle fill except that it does not use a pattern. A 32-bit clear value with 4-bit byte mask is used to fill the entire rectangle area.

Both rectangle fill and clear support clipping, which is performed on a per primitive basis.

15.2.2.1.3 Bit Blit

Bit blit transfers data from one area of a memory (source) to another area of the memory (destination). The source and destination can be from the same or from different memory locations. Both source and destination must be described by a rectangular area. The source and destination rectangles can be the same size (most bit blits are of this nature) or they can be different sizes, in which case the operation becomes a stretch or shrink blit.

Bit blit supports ROP2, ROP3, and ROP4 which includes source, destination and pattern, and an optional transparency color.

Clipping can be performed on a primitive basis.

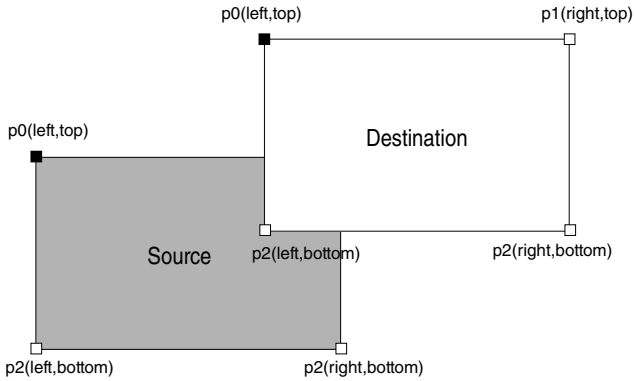


Figure 15-4. Bit Blit

15.2.2.1.4 Stretch Blit

The stretch blit primitive performs a bit blit operation with stretch or shrink. The modified Bresenham algorithm is used to generate corresponding coordinates for fast stretching. The stretch factor is specified in a 15.0 fixed-point format. Stretch blit is not allowed to overlap, that is, no part of source and destination can share any piece of memory. Non-stretch blits can overlap. For stretch blit clipping is performed on a per pixel basis.

15.2.2.1.5 Monochrome Expansion and Mask Blit

Monochrome expansion and mask blit are different operations, although both use the bit stream from the command buffer. And both can be the source for ROP4 source selection. This means that each output pixel can be a combination of source, pattern, monochrome mask (for masked blits) and destination.

Monochrome expansion

For monochrome expansion, the bit from the stream is used to switch on/off a solid color that is defined in a register. This mechanism enables the use of just one bit per pixel to represent colors. In effect, the MONO EXPANSION primitive increases color representation from one bit per pixel to multiple bits per pixel. A typical application for mono color is font drawing.

Monochrome expansion does not support overlapping of the source and destination. It is the responsibility of the driver to make sure that the command will never be executed on overlapping source and destination.

Mask Blit

For mask blit, the bit from the stream is used to toggle on/off a color in the source frame buffer. Mask blit takes its color source from memory and its monochrome mask from the command stream. Clipping is supported and is performed on a per pixel basis.

15.2.2.1.6 YUV Output for 2D Operations

YUV output is available for packed or planar YUV input formats per the following table.

Table 15-5. YUV Output Support for Packed or Planar YUV Source per 2D Operation

2D Operation	Destination
Bit blit	Packed YUV
One Pass Filter	Packed YUV
Multi-source	Packed YUV (no scaling)
Multi-destination	Planar YUV
Stretch blit	NO
Two Pass Filter	NO

15.2.2.1.7 ARGB Data Conversion

The pixels read from source or destination will be expanded into A8R8G8B8 format to maintain lossless pixel operations. The resulting pixels will be converted into the destination format.

15.2.2.1.8 YUV to RGB Conversion

YUV data can be converted into 8-bit per component RGB format at the output of the cache only. Once converted, there is no way back to YUV format. GPU supports BT.601 and BT.709 YUV to RGB color conversion standards.

In BT.601, the YUV to RGB conversion is done using the following approximation:

$$16 \le Y \le 235$$
$$16 \le U \le 240$$
$$16 \le V \le 240$$

$$A = Y - 16 \quad B = U - 128 \quad C = V - 128$$
$$R = clip((298*A + 410*C + 128) >> 8)$$
$$G = clip((298*A - 101*B - 209*C + 128) >> 8)$$
$$B = clip((298*A + 519*B + 128) >> 8)$$

The Y, U and V components are clamped prior to the conversion. Y is clamped between 16 and 235, inclusively. U and V are clamped between 16 and 240, inclusively.

In BT.709, the R, G, B equations are slightly changed to:

R = clip((298*A + 461*C + 128) >> 8)
G = clip((298*A – 55*B – 137*C + 128) >> 8)
B = clip((298*A + 543*B + 128) >> 8)

15.2.2.1.9 Color Index Input Conversion Support

Color index is supported for source data only. A look-up table with 256 entries is provided for indexing the data. The table is fully programmable. The conversion is done when pixels are read out of the cache.

15.2.2.1.10 Source/Destination Pre-multiply Support

The GPU supports pre-multiply source alpha or global alpha, or source global color for global colorizing.

On destination, the GPU supports destination pre-multiply and destination alpha.

On destination, the GPU supports destination pre-multiply destination alpha, destination de-multiply alpha.

Table 15-6. Pre-multiplied modes

Microsoft Alpha Blending	Equation
S:Fix Alpha	D.C=A*S.C+(1-A)*D.C
	D.A=A*S.A+(1-A)*D.A
S:Per-Pixel	D.C=S.C+(1-S.A)*D.C
	D.A=S.A+(1-S.A)*D.A
S:Fix+Per-Pixel	D.C=A*S.C+(1-S.A)*D.C
	D.A=A*S.A+(1-S.A)*D.A
S:Per-Pixel	D.C=S.A*S.C+(1-S.A)*D.C
	D.A=S.A+(1-S.A)*D.A
S:Fix+Per-Pixel	D.C=A*S.A*S.C+(1-S.A)*D.C
	D.A=A*S.A+(1-S.A)*D.A

NOTE

D: Destination

S: Source

A: Alpha

C: Color

15.2.2.1.11 Source/Destination Pre-multiply Support

The GPU supports alpha blending together with ROP. Alpha blending function is performed on ROP function result source.

The general alpha blending equations are:

$$Cd = Fs * Cs' + Fd * Cd'$$
$$Ad = Fs * As'' + Fd * Ad''$$

Where, Cs' is the source color component (adjusted for NPM if necessary)

Cd' is the destination color component (adjusted for NPM if necessary)

As'' is the modified source alpha component

Ad'' is the modified destination alpha component

Fs is fraction of the source that contributes to the final value

Fd is fraction of the destination that contributes to the final value

The blending is done in four logical stages:

1. Transparent/opaque conversion - In this stage, the incoming alpha (source or destination independently) can be inverted if needed to match the internal alpha rule. Internally, an alpha of 0 means transparent, while an alpha of "0xFF" means opaque. External content might follow the opposite rule. The output of the block is either As (Ad for destination) or 1-As (1-Ad for destination).
2. Global value substitution - A global alpha value from a register can be used to substitute or scaled the incoming alpha. An incoming alpha As can pass-through, be directly substituted by Ags (global alpha) or scaled by the global alpha value (As * Ags). The source and destination have distinct global alpha values.
3. Blending factor generation - At this stages, the blending factors are generated (refer to table below). Each alpha can take the values 0, 1, A or 1-A depending on the blending mode.
4. Final blending - This is the final stage which implements the operations described by equations.
5.

The stages are shown in the figure below:

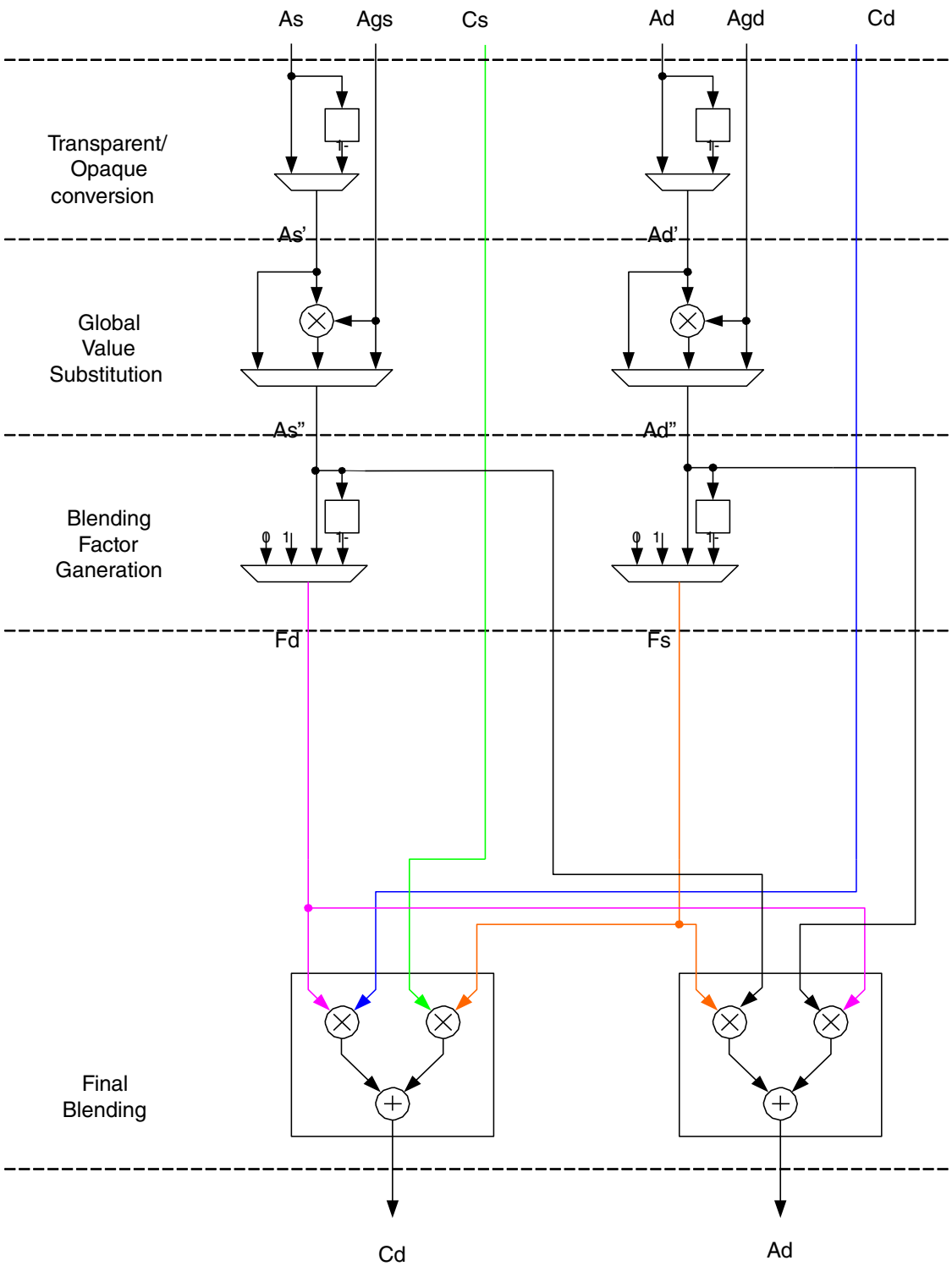


Figure 15-5. Alpha Blender for One Pixel

The fractions take the values described in the following table, depending on the blending mode:

Table 15-7. Blending Modes Fractions Description

Blending Mode	Fs	Fd
Clear	0	0
SRC	1	0
DST	0	1
SRC_OVER	1	1 – As”
DST_OVER	1 – Ad”	1
SRC_IN	Ad”	0
DST_IN	0	As”
SRC_OUT	1 – Ad”	0
DST_OUT	0	1 – As”
SRC_ATOP	Ad”	1 – As”
DST_ATOP	1 – Ad”	As”
XOR	1 – Ad”	1 – As”

15.2.2.1.12 Dither

Color intensity at any point of an image can be thought of as a real number between 0.0 and 1.0 at each location. However, in reality only a discrete number of intensity levels can be represented. This discretization leads to a total error in the color intensities of an image equal to the square root of the sum of the squares from each point.

This error results in contouring. In an image where intensities change slowly, this will cause noticeable jumps. Dithering can be used to diffuse the intensity across neighboring pixels.

In dithering mode pixels are scanned in order, and errors in calculating a pixel's intensity are distributed (i.e., diffused) to neighboring pixels to keep the overall intensity of the image closer to the input intensity.

Dithering is supported for 16-bit color destination formats. The dither table is programmable by software.

15.2.2.1.13 Multi-Source Blending

Up to 8 sources are supported. Programmable block size guarantees cache efficiency so each source is read once and each destination is written once. It supports 90, 180, 270 degree destination and source rotation and mirror with different block size to have high

cache efficiency. It also supports ROP2, ROP3, and ROP4 which includes source, destination and pattern, and an optional transparency color. Alpha blending between every source is supported.

15.2.2.1.14 GPU Cache Management

SW cache flush is supported to flush the GPU cache to memory. Auto-flush of GPU cache is also supported. SW sets up the auto-flush interval, and HW will do the cache flush automatically at programmable intervals.

15.2.2.2 Clocks

The following table describes the clock sources for the 2D GPU. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 15-8. 2D GPU Clocks

Clock name	Description
clk2x	GPU core clock
HCLK	AHB interface clock
ACLK	AXI interface clock

15.2.3 External Signals

There are no external signals pinned out for this module.

15.2.4 Initialization and Application Information

The GPU should be programmed through the NXP provided driver. NXP does not provide support for software that directly programs the GPU registers. APIs for programming the GPU through the software driver are described in separate driver documentation.

15.3 3D Graphics Processing Unit (GPU3D)

15.3.1 Overview

The 3D Graphics Processing Unit (GPU) provides high performance, high quality graphics, and low power consumption.

The 3D GPU supports modern API features and full support for Android, Linux, and a variety of other embedded development platforms.

15.3.1.1 Block Diagram

The block diagram of the GPU and description of the functional blocks is provided below.

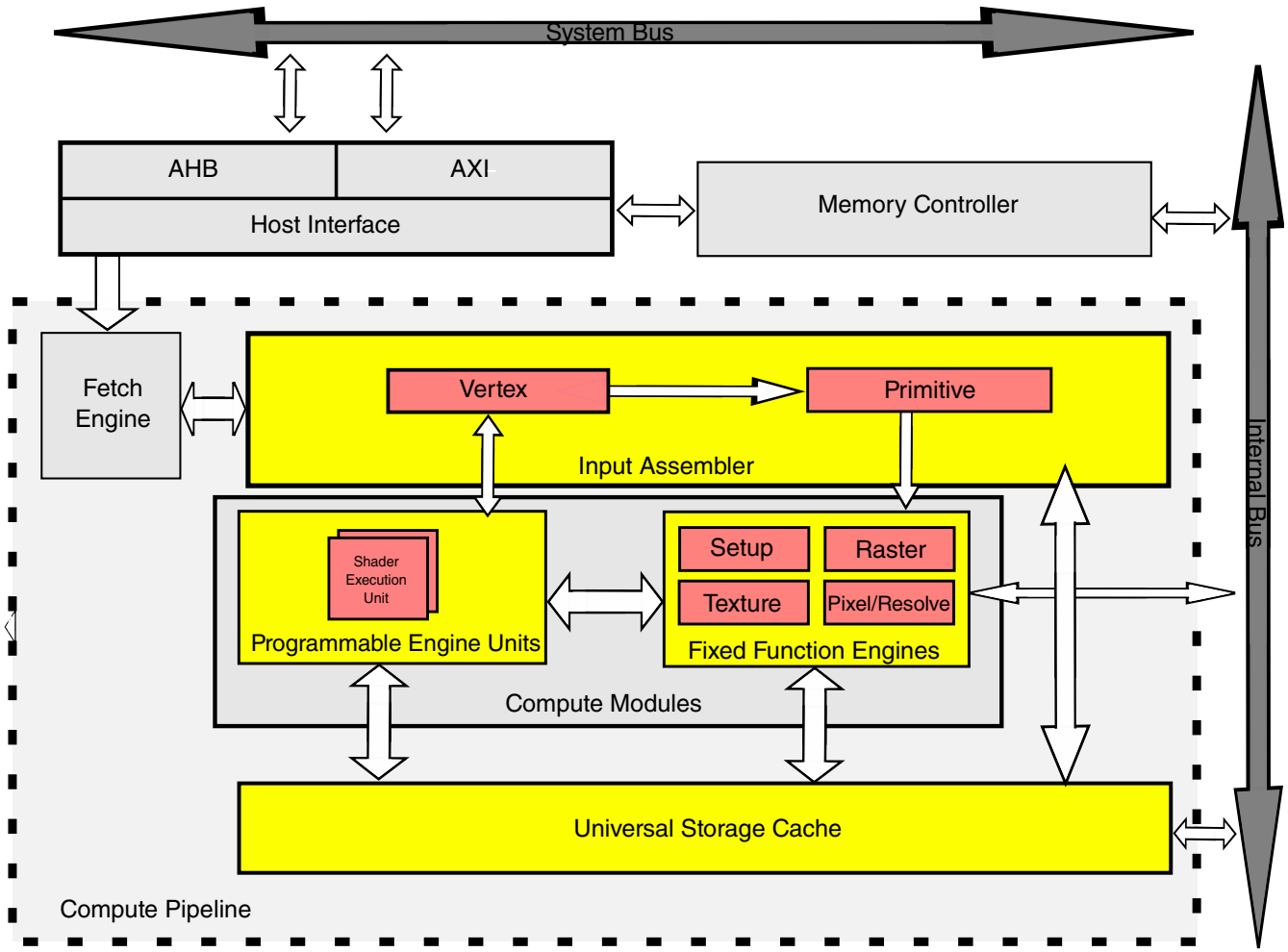


Figure 15-6. GPU Block Diagram

- Host Interface - Allows GCCORE to communicate with external memory and the CPU through AXI or AHB bus. In this block data crosses clock domain boundaries.
- Memory Controller - Internal memory management unit that controls the block-to-host memory request interface.
- Fetch Engine - Inserts high level primitives and commands into the graphics pipeline.
- Input Assembly - Assembles vertices and primitives for further processing.
- Compute Modules:
  - Fixed Function 3D Rendering Engine Units:
    - Setup/Raster - Converts triangles and lines into pixels. Computes slopes of color attributes and texture coordinates. Performs clipping.

- Texture Engine - Retrieves texture information from memory upon request by the vertex shader or fragment shader. Performs interpolation and filtering, and transfers the computed value to the fragment shader or the vertex shader. The texture unit can process one pixel or one vertice/cycle.
- Pixel Engine/Resolve - The Pixel Engine does alpha blending and visible surface determination. Resolve does resolve, clear, tiling and de-tiling as well as FSAA filtering. The Pixel Engine can process one pixel/cycle.

- Programmable Engine Units:
  - Shader Execution Units - SIMD processor that performs as vertex shader and fragment shader. When used as a vertex shader, it performs geometry transformations and lighting computations. When used as a fragment shader it applies texture data and computes color values for each pixel. The shaders also act as Compute Units for OpenCL. The GPU has 2 such vec4 shaders.
- Universal Storage Cache - Cache shared between the Fetch Engine and the Compute Modules. A portion of this cache can be locked to stay on-chip.

15.3.1.2 Features

The features of the GPU 3D unit include:

- IEEE 32-bit floating-point pipeline
- Ultra-threaded, unified vertex and fragment (pixel) shaders
- Low bandwidth at both high and low data rates
- Low CPU loading
- Up to 16 programmable elements per vertex
- Dependent texture operation with high-performance
- Alpha blending
- Depth and stencil compare
- Point sampling, bi-linear sampling, tri-linear filtering, and cubic textures
- Resolve and fast clear
- 8k x 8k texture size and 8k x 8k rendering target
- 8 Vertex DMA streams
- Generates 1 texel and 1 pixel per cycle.
- Supports YUV formats for display output (YUV422, YUYV 4:2:2)
- MMU functionality supported
- Wide AA line support
- Primitive restart support

15.3.1.2.1 API Support

The following table describes API support and select architectural features of the GPU:

Table 15-9. GPU Architecture Features

Feature	GPU Support
Primary API	OpenGL ES 3.1 / 3.0 / 2.0 / 1.1 OpenCL 1.1 / 1.2 / 3.0
Additional APIs	OpenVG 1.1 DirectX 11 (9_3) OpenGL 4.0 Vulkan 1.1
Other graphics support	EGL 1.5
Drivers	OpenGL ES 3.1 / 3.0 / 2.0 / 1.1 OpenVG 1.1 EGL 1.5 DirectX 11 (9_3) OpenGL 4.0 OpenCL 1.1 / 1.2 / 3.0
Operating systems	Windows Embedded Compact / Embedded CE 6 Embedded Linux Android 8.0/7.x / 6.0 /5.1 / 4.x
Z (depth)	Early Z support included
Stencil	Early stencil support included
GLSL ES Shader languages	3.1 / 3.0 / 1.0
DirectX Shader model compatibility	SM3.0
Shader types and execution units	Two programmable Scalable Ultra-threaded Unified Shaders (SIMD4:transcendental,ctl-flow,tx-load) one instruction issue per shader per clock; IEEE 32-bit floating-point pipeline supports long shader instructions
FSAA anti-aliasing mechanisms	High quality MSAA 4x

15.3.1.2.2 GPU Host and Memory Interface Features

Table 15-10. GPU host and memory interface features

Feature	GPU Support
AHB interface	32-bit
AXI interface	1 128-bit AXI / ACE-Lite interfaces for external memory access
Virtual memory support	Yes
Code and data memory location restrictions	Unrestricted; arbitrary memory reads and writes
Physical address	32 bits
Resource locks with CPU	Semaphore lock

15.3.1.2.3 GPU Power Management Features

Table 15-11. Power management features

Feature	GPU Support
Low power CMOS technology compatible	Yes
Dynamic frequency scaling based on current load	Yes
Automatic clock gating of flip flops and rams	Yes
Global clock gating of unused macro blocks	Yes
Independent clock gating for the shader core	Yes
Software controlled effective clock frequency without changing the PLL	Yes

15.3.1.2.4 GPU Command Processor Features

Table 15-12. GPU command processor features

Feature	GPU Support
Counters	Variety of hardware counters for performance profiling

15.3.1.2.5 OpenCL Support

Support for OpenCL includes the following capabilities in a full memory configuration:

Table 15-13. OpenCL Support

OpenCL Support	GPU Support
Profile	Full Profile with Global memory
Compute Devices (GPGPU cores)	1
Compute Units per device	1
Processing Elements per compute unit (vec4 sh cores)	2
Preferred work-group/ thread group size	8
Max count global work-items each dim	64KB
Max count work-items each dim per work-group	1 KB
Instruction Cache Size	512

15.3.1.2.6 Texture processing

Table 15-14. Texture Processing Features

Feature	GPU Support
Fixed-point input texture formats	A8, L8, A8L8, ARGB4, XRGB4, ARGB8, XRGB8, ABGR8, XBGR8, R5G6B5, A1RGB5, X1RGB5, YUY2, UYVY, D16, D24X8, A8_OES, DXT1, DXT2, DXT3, DXT4, DXT5, HDR7E3, HDR6E4, HDR5E5,

Table continues on the next page...

Table 15-14. Texture Processing Features (continued)

<table>
<tr><th>Feature</th><th>GPU Support</th></tr>
<tr><td></td><td>HDR6E5, RGBE8, RGBE8F, RGB9E5, RGB9E5F, ETC1, ETC2_RGB8, ETC2_RGB8A1, ETC2_RGB8A8, EAC_R11_UNSIGNED, EAC__RG11_UNSIGNED, EAC__RG11_SIGNED, R8G8, RF16, RF16GF16, RF32, R10G10B10A2, EAC__R11_SIGNED, R8__SNORM, RG8__SNORM, RGBX8__SNORM, RGBA8__SNORM. All fixed-point formats are filtered. sRGB conversion and Swizzling supported for color channels.
<table>
<tr><th>Bits</th><th>Format</th><th>Alpha</th><th>R</th><th>B</th><th>G</th></tr>
<tr><td>16</td><td>ARGB4444</td><td>4</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>XRGB4444</td><td>4 don't care</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>ARGB1555</td><td>1</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>16</td><td>XRGB1555</td><td>1 don't care</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>16</td><td>RGB565</td><td>0</td><td>5</td><td>6</td><td>5</td></tr>
<tr><td>32</td><td>ARGB8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XRGB8888</td><td>8 don't care</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>ABGR8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XBGR8888</td><td>8 don't care</td><td>8</td><td>8</td><td>8</td></tr>
</table>
<table>
<tr><th>Planes</th><th>Format</th><th>Mode</th><th>Y</th><th>U</th><th>V</th><th>UV</th><th>YUYV</th><th>UYVY</th></tr>
<tr><td>1</td><td>YUY2</td><td>4:2:2</td><td></td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>1</td><td>UYVY</td><td>4:2:2</td><td></td><td></td><td></td><td></td><td></td><td>1</td></tr>
</table></td></tr>
<tr><td>Additional texture formats supported through Resolve conversion</td><td>Resolve converts planar to YUV 4:2:2 packed:
<table>
<tr><th>Planes</th><th>Format</th><th>Mode</th><th>Y</th><th>U</th><th>V</th><th>UV</th><th>YUYV</th><th>UYVY</th></tr>
<tr><td>3</td><td>YV12</td><td>4:2:0</td><td>1</td><td>1</td><td>1</td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>NV12</td><td>4:2:0</td><td>1</td><td></td><td></td><td>1</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>Texture compression</td><td>4 bits and 8 bits per texel</td></tr>
<tr><td>Compressed texture formats</td><td>DXT1, DXT2, DXT3, DXT4, DXT5, ETC1, ETC2, EAC11. All compressed formats are filtered.</td></tr>
<tr><td>Texture Formats for Optimal Performance</td><td>Formats with 4 bits per texel: DXT1, ETC1, EXT2_RGB8A1, EAC_R11_UNSIGNED, EAC_R11_SIGNED</td></tr>
<tr><td>Texture size maximum</td><td>8k x 8k</td></tr>
<tr><td>Addressing modes</td><td>wrap, mirror, clamp</td></tr>
<tr><td>Mipmap support</td><td>14 mipmap levels; programmable LOD biasing and replacement</td></tr>
<tr><td>Shadow texture</td><td>Depth texture PCF filtering</td></tr>
<tr><td>Texture coordinate fraction bits</td><td>5 bits</td></tr>
<tr><td>Texture samplers</td><td>32 samplers, indexable</td></tr>
<tr><td>Textures per fragment maximum</td><td>16 texture samplers</td></tr>
</table>

Table continues on the next page...

Table 15-14. Texture Processing Features (continued)

Feature	GPU Support
Dependent texture operation	High performance; unlimited dependent texture reads
Dependent tx per fragment max, relative sampling	No Limit
Texture repeat max	256
Texture types	2D, seamless cube map, 1D, projected, depth, bump map, displacement map, PCF, 3D, texture array
Texture filters	Point sample, bi-linear, fast tri-linear (1 pixel/cycle), quad-linear, Anisotropic
Texture component mapping: D3D, OGL ES options	Supports both D3D and OES options
Texture size types	Power-of-2, Non-square, Non-power-of-2
Texture swizzle	Yes
OES3 half-float support	Yes

15.3.1.2.7 Fragment Processing and Render Targets

Table 15-15. Fragment Processing and Render Target Features

<table>
<tr><th>Feature</th><th>GPU Support</th></tr>
<tr><td>FSAA mechanisms</td><td>MSAA 4x, SSAA 4x</td></tr>
<tr><td>Fragment color, alpha, Z, stencil precision</td><td>
<table>
<tr><th>Bits</th><th>Format</th><th>Alpha</th><th>R</th><th>B</th><th>G</th></tr>
<tr><td>16</td><td>ARGB4444</td><td>4</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>XRGB4444</td><td>4 don't care</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>ARGB1555</td><td>1</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>16</td><td>XRGB1555</td><td>1 don't care</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>16</td><td>RGB565</td><td>0</td><td>5</td><td>6</td><td>5</td></tr>
<tr><td>16</td><td>RF16</td><td>0</td><td>16</td><td>0</td><td>0</td></tr>
<tr><td>32</td><td>ARGB8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XRGB8888</td><td>8 don't care</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>ABGR8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XBGR8888</td><td>8 don't care</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>RGB10_a2ui</td><td>2</td><td>10</td><td>10</td><td>10</td></tr>
<tr><td>32</td><td>RF16GF16</td><td>0</td><td>16</td><td>16</td><td>0</td></tr>
<tr><td>32</td><td>RF32</td><td>0</td><td>32</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><th>Bits</th><th>Format</th><th>Depth</th><th>Stencil</th></tr>
<tr><td>16</td><td>D16</td><td>16</td><td>0</td></tr>
<tr><td>32</td><td>D24S8</td><td>24</td><td>8</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

Table 15-15. Fragment Processing and Render Target Features (continued)

Feature	GPU Support
	Render Target no blend formats: A8, R8G8, RI8, RI16, RI32, RI8GI8, RI16GI16, RI32GI32, RI8GI8BI8AI8, RI16GI16BI16AI16, RI10GI10BI10AI2, RF11GF11BF10
Fragment storage / Render Target formats	16-bit color and z, 32-bit color and z for each fragment. Lossless compression, no storage reduction.
Individual fragment alpha masking	Yes
Two sided stencil support	Yes
Compressed formats	Yes, 30% - 70%
Fast clear and initialization options	Yes
Multiple Render Target support	Yes, 4 Targets

15.3.1.2.8 Destination / Alpha Blending

Table 15-16. Destination / Alpha Blending Features

Feature	GPU Support					
Destination / Resolve color formats	Bits	Format	Alpha	R	B	G
	16	ARGB4444	4	4	4	4
	16	ARGB1555	1	5	5	5
	16	RGB565	0	5	6	5
	32	ARGB8888	8	8	8	8
	32	ABGR8888	8	8	8	8
	32	RGB10_a2ui	2	10	10	10

15.3.2 Clocks

The following table describes the clock sources for the GPU. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 15-17. GPU Clocks

Clock name	Description
clkCore	GPU core clock for modules except Shader
clkSh	Core clock for Shader and Universal Storage Cache (USC)
HCLK	AHB interface clock
ACLK	AXI interface clock

15.3.3 External Signals

There are no external signals pinned out for this module.

15.3.4 Initialization and Application Information

The GPU should be programmed through the NXP provided driver. NXP does not provide support for software that directly programs the GPU registers. APIs for programming the GPU through the software driver are described in separate driver documentation.

Chapter 16
Video Processing Unit (VPU)

16.1 VPU G1 Decoder

16.1.1 Overview

The Video Processing Unit (VPU) G1 hardware-based decoder decodes H.264, and VP8 formats. The decoder conforms to the H.264 Baseline, Main and High profile, and VP8 Main Profile

16.1.1.1 Block Diagram

The block diagram of the VPU G1 decoder is shown below.

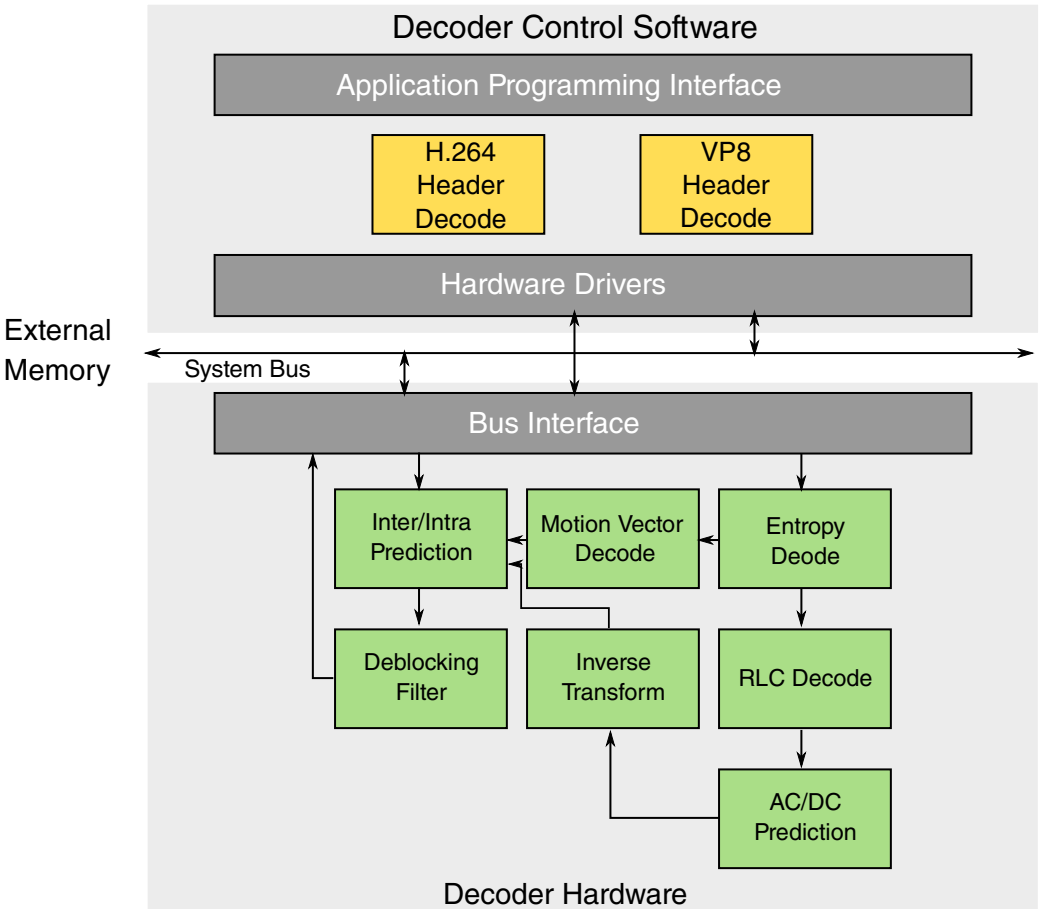


Figure 16-1. VPU G1 Decoder Block Diagram

16.1.1.2 Features

The VPU G1 decoder includes the following features:

- H.264 baseline profile, levels 1-4.2 decoding
  - Byte stream input
  - NAL unit input
- H.264 main and high profile, levels 1-4.2 decoding
- VP8 decoding

16.1.2 Functional description

16.1.2.1 Video Frame Storage Format

The following sections describe the different input and output picture storage formats supported by the decoder.

Table 16-1. Input and output picture storage formats

Input data format	Output data format
H.264 byte or NAL unit stream	YCbCr 4:2:0 semi-planar raster-scan YCbCr 4:2:0 semi-planar 8x4 tiled YCbCr 4:0:0 (monochrome)
VP8 stream	YCbCr 4:2:0 semi-planar raster-scan YCbCr 4:2:0 semi-planar 8x4 tiled

16.1.2.1.1 Raster-scan format

The output picture of the decoder is in semi-planar YCbCr 4:2:0 format, i.e. luminance data forms one plane in memory, and chrominance data forms another. The output picture has to be stored linearly and contiguously in the memory. The interleaved chrominance block has to be stored right after the luminance block in external memory as shown in the following figure. The number of luminance pixels in one row must be divisible by 16.

The interleaved chrominance is stored raster-scanned in memory in Cb0Cr0Cb1Cr1Cb2Cr2Cb3… format.

The decoder output picture for interlaced sequence is always stored as an interlaced frame where even lines belongs to top field and odd lines belongs to bottom field of the frame.

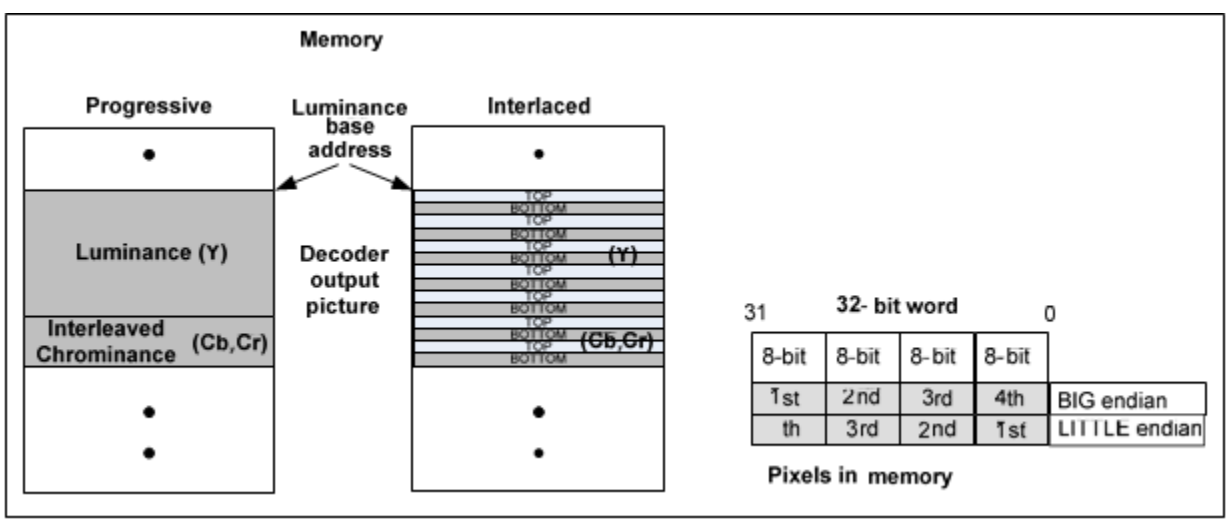


Figure 16-2. Raster-scan Format External Memory Usage

16.1.2.1.2 Tiled format

The output picture of the decoder is in semi-planar YCbCr 4:2:0 format, i.e. luminance data forms one plane in memory, and chrominance data forms another. The distinction from raster scan format is that the output picture is grouped into tiles, which are then stored linearly and contiguously in the memory. The chrominance tiles have to be stored right after the luminance tiles in external memory as shown in Figure 16-3. The number of luminance pixels in one row must be divisible by 16.

The chrominance is stored into tiles in interleaved Cb0Cr0Cb1Cr1Cb2Cr2Cb3… format.

The following figure is an example of external memory usage when using 8x4 tiles.

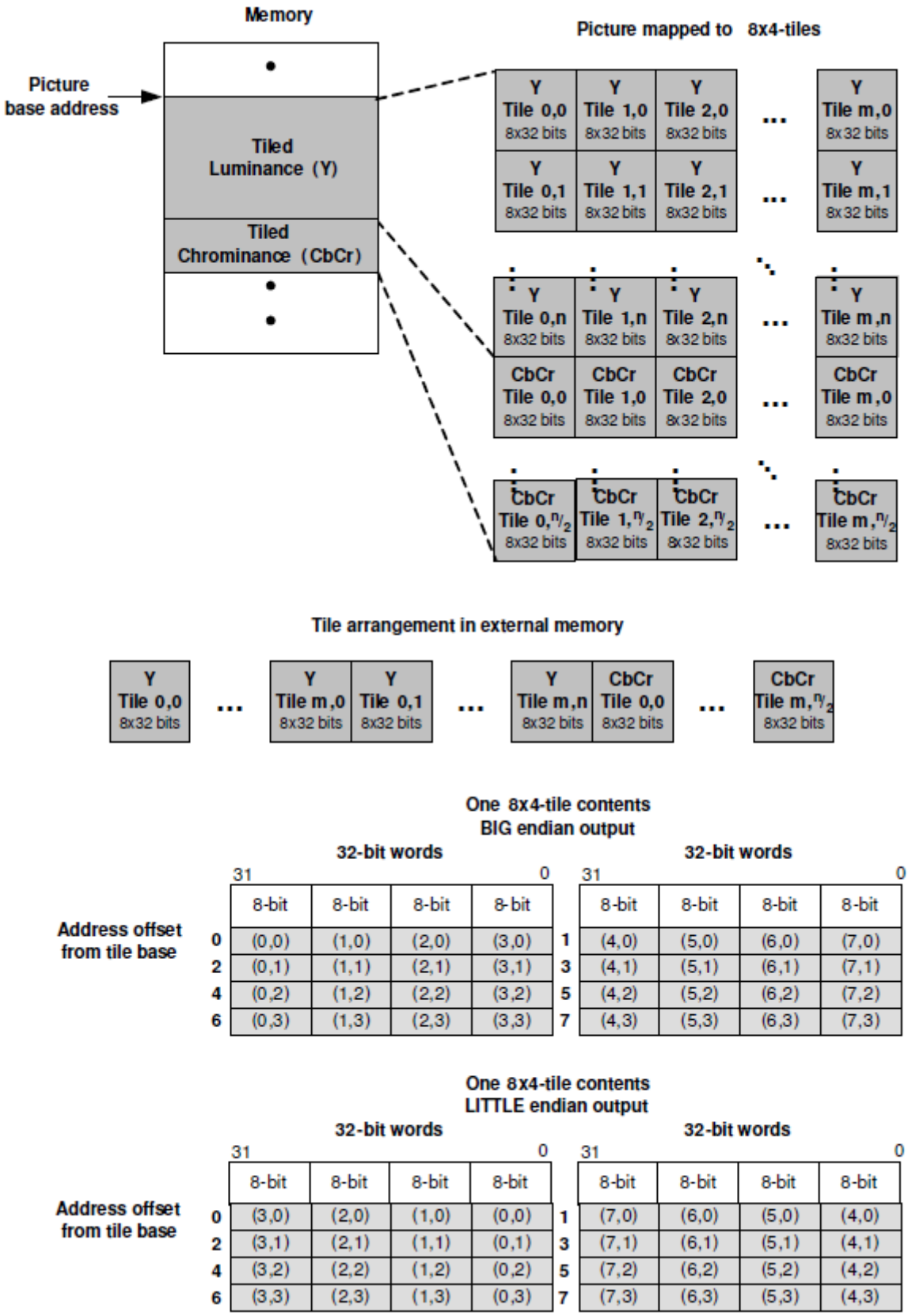


Figure 16-3. External Memory Usage in 8x4 Tiled Reference Picture Mode

16.1.2.2 Clocks

The following table describes the clock sources for the VPU G1 decoder. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 16-2. VPU G1 Clocks

Clock name	Description
core_clk	VPU G1 core clock

16.1.2.3 Reset

The VPU G1 decoder has one software reset (g1_sft_rstn).

16.1.3 External Signals

There are no external signals pinned out for this module.

16.1.4 Initialization and Application Information

The VPU G1 decoder is operated via the VPU Application Programming Interface (API). Details are provided in separate API documentation.

16.1.5 VPU G1 Register Decoder

The definition/function of a register's bit field(s) may differ depending on the decoder mode. The following table maps the bit fields in a register that are applicable for each decoder mode.

NOTE

The VPU G1 decoder supports H.264 and VP8 decoding only.

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
swreg0			
31:16	-	x	x
15:12	-	x	x

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
11:4	-	x	x
3	-	x	x
2:0	-	x	x
swreg1			
31:25	-		
24	sw_dec_pic_inf	x	
23:19	-		
18	sw_dec_timeout	x	x
17	sw_dec_slice_int		
16	sw_dec_error_int	x	x
15	sw_dec_aso_int	x	x
14	sw_dec_buffer_int	x	x
13	sw_dec_bus_int	x	x
12	sw_dec_rdy_int	x	x
11	sw_dec_abort_int	x	x
10:9	-		
8	sw_dec_irq	x	x
7:6	-		
5	sw_dec_abort_e	x	x
4	sw_dec_irq_dis	x	x
3:1	-		
0	sw_dec_e	x	x
swreg2			
31:24	sw_dec_axi_rd_id	x	x
23	sw_dec_timeout_e	x	x
22	sw_dec_strswap32_e	x	x
21	sw_dec_strendian_e	x	x
20	sw_dec_inswap32_e	x	x
19	sw_dec_outswap32_e	x	x
18	sw_dec_data_disc_e	x	x
18	sw_dec_2chan_dis	x	x
17	sw_tiled_mode_msb	x	x
17	sw_dec_out_tiled_e	x	
16:11	sw_dec_latency	x	x
10	sw_dec_clk_gate_e	x	x
9	sw_dec_in_endian	x	x
8	sw_dec_out_endian	x	x
7:5	sw_priority_mode	x	
7	sw_tiled_mode_lsb	x	x
6	sw_dec_adv_pre_dis	x	x

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
5	sw_dec_scmd_dis	x	x
4:0	sw_dec_max_burst	x	x
swreg3			
31:28	sw_dec_mode	x	x
27	sw_rlc_mode_e	x	
26	sw_skip_mode		x
25	sw_divx3_e		
24	sw_pjpeg_e		
23	sw_pic_interlace_e	x	
22	sw_pic_fieldmode_e	x	
21	sw_pic_b_e		
20	sw_pic_inter_e		x
19	sw_pic_topfield_e	x	
18	sw_fwd_interlace_e		
17	sw_sorenson_e		
16	sw_ref_topfield_e		
15	sw_dec_out_dis	x	x
14	sw_filtering_dis	x	
13	sw_webp_e		x
13	sw_mvc_e	x	
13	sw_pic_fixed_quant		
12	sw_write_mvs_e	x	x
11	sw_reftopfirst_e		
10	sw_seq_mbaff_e	x	
9	sw_picord_count_e	x	
8	sw_dec_ahb_hlock_e	x	x
7:0	sw_dec_axi_wr_id	x	x
swreg4			
31:23	sw_pic_mb_width	x	x
22:19	sw_mb_width_off		
18:11	sw_pic_mb_height_p	x	x
10:7	sw_mb_height_off		
6	sw_alt_scan_e		
5	sw_topfieldfirst_e		
4:0	sw_ref_frames	x	
5:3	sw_pic_mb_w_ext		x
2:0	sw_pic_mb_h_ext		x
0	sw_pic_refer_flag		
swreg5			
31:26	sw_strm_start_bit	x	x

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
25	sw_sync_marker_e		
24	sw_type1_quant_e	x	
23:19	sw_ch_qp_offset	x	
18:14	sw_ch_qp_offset2	x	
13:1	-	x	
0	sw_fieldpic_flag_e	x	
18:16	sw_intradc_vlc_thr		
15:0	sw_vop_time_incr		
24	sw_dq_profile		
23	sw_dqbi_level		
22	sw_range_red_frm_e		
21	-		
20	sw_fast_uvmc_e		
19:18	-		
17	sw_transdctab		
16:15	sw_transacfrm		
14:13	sw_transacfrm2		
12:10	sw_mb_mode_tab		
9:7	sw_mvtab		
6:4	sw_cbptab		
3:2	sw_2mv_blk_pat_tab		
1:0	sw_4mv_blk_pat_tab		
24	sw_qscale_type		
23:5	-		
4	sw_con_mv_e		
3:2	sw_intra_dc_prec		
1	sw_intra_vlc_tab		
0	sw_frame_pred_dct		
12:11	sw_jpeg_qtables		
10:8	sw_jpeg_mode		
7	sw_jpeg_filright_e		
6	sw_jpeg_stream_all		
5	sw_cr_ac_vlctable		
4	sw_cb_ac_vlctable		
3	sw_cr_dc_vlctable		
2	sw_cb_dc_vlctable		
1	sw_cr_dc_vlctable3		
0	sw_cb_dc_vlctable3		
23:18	sw_strm1_start_bit		x
17	sw_huffman_e		

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
16	sw_multistream_e		
15:8	sw_boolean_value		x
7:0	sw_boolean_range		x
9:5	sw_alpha_offset		
4:0	sw_beta_offset		
swreg6			
31	sw_start_code_e	x	
30:25	sw_init_qp	x	
24	sw_ch_8pix_ileav_e	x	
31:24	sw_stream_len_ext		x
23:0	sw_stream_len	x	x
swreg7			
31	sw_cabac_e	x	
30	sw_blackwhite_e	x	
29	sw_dir_8x8_infer_e	x	
28	sw_weight_pred_e	x	
27:26	sw_weight_bipr_idc	x	
25	sw_avs_h264_h_ext	x	
24:21	-	x	
20:16	sw_framenum_len	x	
15:0	sw_framenum	x	
31	sw_bitplane0_e		
30	sw_bitplane1_e		
29	sw_bitplane2_e		
28:24	sw_alt_pquant		
23:20	sw_dq_edges		
19	sw_ttmbf		
18:14	sw_pqindex		
13	sw_vc1_height_ext		
12	sw_bilin_mc_e		x
11	sw_uniqp_e		
10	sw_halfqp_e		
9:8	sw_ttfrm		
7	sw_2nd_byte_emul_e		
6	sw_dquant_e		
5	sw_vc1_adv_e		
31:26	sw_dct1_start_bit		x
25:20	sw_dct2_start_bit		x
13	sw_ch_mv_res		x
11:9	sw_init_dc_match0		x

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
8:6	sw_init_dc_match1		x
5	sw_vp7_version		x
swreg8			
31	sw_const_intra_e	x	
30	sw_filt_ctrl_pres	x	
29	sw_rdpic_cnt_pres	x	
28	sw_8x8trans_flag_e	x	
27:17	sw_refpic_mk_len	x	
16	sw_idr_pic_e	x	
15:0	sw_idr_pic_id	x	
31:24	sw_mv_scalefactor		
23:19	sw_ref_dist_fwd		
18:14	sw_ref_dist_bwd		
17:14	sw_loop_filt_limit		
13	sw_variance_test_e		
12:10	sw_mv_threshold		
9:0	sw_var_threshold		
8	sw_divx_idct_e		
7:0	sw_divx3_slice_size		
31:30	sw_rv_profile		
29:28	sw_rv_osv_quant		
27:14	sw_rv_fwd_scale		
13:0	sw_rv_bwd_scale		
31:16	sw_init_dc_comp0		x
15:0	sw_init_dc_comp1		x
swreg9			
31:24	sw_pps_id	x	
23:19	sw_refidx1_active	x	
18:14	sw_refidx0_active	x	
7:0	sw_poc_length	x	
24	sw_icomp0_e		
23:16	sw_iscale0		
15:0	sw_ishift0		
23:0	sw_stream1_len		x
31:2	sw_mb_ctrl_base	x	
12:0	sw_pic_slice_am		
27:24	sw_coeffs_part_am		x
swreg10			
31:2	sw_diff_mv_base	x	
29:25	sw_pinit_rlist_f9	x	

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
24:20	sw_pinit_rlist_f8	x	
19:15	sw_pinit_rlist_f7	x	
14:10	sw_pinit_rlist_f6	x	
9:5	sw_pinit_rlist_f5	x	
4:0	sw_pinit_rlist_f4	x	
24	sw_icomp1_e		
23:16	sw_iscale1		
15:0	sw_ishift1		
31:2	sw_segment_base		x
1	sw_segment_upd_e		x
0	sw_segment_e		x
swreg11			
31:2	sw_i4x4_or_dc_base	x	
29:25	sw_pinit_rlist_f15	x	
24:20	sw_pinit_rlist_f14	x	
19:15	sw_pinit_rlist_f13	x	
14:10	sw_pinit_rlist_f12	x	
9:5	sw_pinit_rlist_f11	x	
4:0	sw_pinit_rlist_f10	x	
24	sw_icomp2_e		
23:16	sw_iscale2		
15:0	sw_ishift2		
29:24	sw_dct3_start_bit		x
23:18	sw_dct4_start_bit		x
17:12	sw_dct5_start_bit		x
11:6	sw_dct6_start_bit		x
5:0	sw_dct7_start_bit		x
swreg12			
31:0	sw_rlc_vlc_base	x	x
swreg13			
31:2	sw_dec_out_base	x	x
1	sw_dpb_ilace_mode	x	
swreg14			
31:2	sw_refer0_base	x	x
1	sw_refer0_field_e	x	
0	sw_refer0_topc_e	x	
31:2	sw_jpg_ch_out_base		x
swreg15			
31:2	sw_refer1_base	x	x
1	sw_refer1_field_e	x	

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
0	sw_refer1_topc_e	x	
7:0	sw_jpeg_slice_h		x
swreg16			
31:2	sw_refer2_base	x	
1	sw_refer2_field_e	x	
0	sw_refer2_topc_e	x	
31	-		
30:24	sw_ac1_code6_cnt		
23:22	-		
21:16	sw_ac1_code5_cnt		
15:11	sw_ac1_code4_cnt		
10:7	sw_ac1_code3_cnt		
6	-		
5:3	sw_ac1_code2_cnt		
2	-		
1:0	sw_ac1_code1_cnt		
swreg17			
31:2	sw_refer3_base	x	
1	sw_refer3_field_e	x	
0	sw_refer3_topc_e	x	
31:24	sw_ac1_code10_cnt		
23:16	sw_ac1_code9_cnt		
15:8	sw_ac1_code8_cnt		
7:0	sw_ac1_code7_cnt		
swreg18			
31:2	sw_refer4_base	x	x
1	sw_refer4_field_e	x	
0	sw_refer4_topc_e	x	
31:16	sw_pic_header_len		
15:14	-		
13	sw_pic_4mv_e		
12	-		
11	sw_range_red_ref_e		
10:9	sw_vc1_difmv_range		
8	-		
7:6	sw_mv_range		
5	sw_overlap_e		
4:3	sw_overlap_method		
19	sw_alt_scan_flag_e		
18:15	sw_fcode_fwd_hor		

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
14:11	sw_fcode_fwd_ver		
10:7	sw_fcode_bwd_hor		
6:3	sw_fcode_bwd_ver		
2	sw_mv_accuracy_fwd		
1	sw_mv_accuracy_bwd		
1	sw_mpeg4_vc1_rc		
0	sw_prev_anc_type		
31:24	sw_ac1_code14_cnt		
23:16	sw_ac1_code13_cnt		
15:8	sw_ac1_code12_cnt		
7:0	sw_ac1_code11_cnt		
0	sw_gref_sign_bias		x
swreg19			
31:2	sw_refer5_base	x	x
1	sw_refer5_field_e	x	
0	sw_refer5_topc_e	x	
26:0	sw_trb_per_trd_d0		
24	sw_icomp3_e		
23:16	sw_iscale3		
15:0	sw_ishift3		
31:27	sw_ac2_code4_cnt		
26:23	sw_ac2_code3_cnt		
22	-		
21:19	sw_ac2_code2_cnt		
18	-		
17:16	sw_ac2_code1_cnt		
15:8	sw_ac1_code16_cnt		
7:0	sw_ac1_code15_cnt		
29:24	sw_scan_map_1		x
23:18	sw_scan_map_2		x
17:12	sw_scan_map_3		x
11:6	sw_scan_map_4		x
5:0	sw_scan_map_5		x
0	sw_aref_sign_bias		x
swreg20			
31:2	sw_refer6_base	x	
31:2	sw_vp8_dec_ch_base	x	
1	sw_vp8_stride_e		x
0	sw_vp8_ch_base_e		x
1	sw_refer6_field_e	x	

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
0	sw_refer6_topc_e	x	
26:0	sw_trb_per_trd_dm1		
24	sw_icomp4_e		
23:16	sw_iscale4		
15:0	sw_ishift4		
31:24	sw_ac2_code8_cnt		
23:16	sw_ac2_code7_cnt		
15			
14:8	sw_ac2_code6_cnt		
7:6	-		
5:0	sw_ac2_code5_cnt		
29:24	sw_scan_map_6		x
23:18	sw_scan_map_7		x
17:12	sw_scan_map_8		x
11:6	sw_scan_map_9		x
5:0	sw_scan_map_10		x
swreg21			
31:2	sw_refer7_base	x	
31:27	sw_y_stride_pow2		x
26:22	sw_c_stride_pow2		x
1	sw_refer7_field_e	x	
0	sw_refer7_topc_e	x	
26:0	sw_trb_per_trd_d1		
31:24	sw_ac2_code12_cnt		
23:16	sw_ac2_code11_cnt		
15:8	sw_ac2_code10_cnt		
7:0	sw_ac2_code9_cnt		
29:24	sw_scan_map_11		x
23:18	sw_scan_map_12		x
17:12	sw_scan_map_13		x
11:6	sw_scan_map_14		x
5:0	sw_scan_map_15		x
swreg22			
31:2	sw_refer8_base	x	
31:2	sw_dct_strm1_base		x
1	sw_refer8_field_e	x	
0	sw_refer8_topc_e	x	
31:24	sw_ac2_code16_cnt		
23:16	sw_ac2_code15_cnt		
15:8	sw_ac2_code14_cnt		

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
7:0	sw_ac2_code13_cnt		
29:24	sw_scan_map_16		
23:18	sw_scan_map_17		
17:12	sw_scan_map_18		
11:6	sw_scan_map_19		
5:0	sw_scan_map_20		
swreg23			
31:2	sw_refer9_base	x	
31:2	sw_dct_strm2_base		x
1	sw_refer9_field_e	x	
0	sw_refer9_topc_e	x	
31:28	sw_dc1_code8_cnt		
27:24	sw_dc1_code7_cnt		
23:20	sw_dc1_code6_cnt		
19:16	sw_dc1_code5_cnt		
15:12	sw_dc1_code4_cnt		
11:8	sw_dc1_code3_cnt		
7	-		
6:4	sw_dc1_code2_cnt		
3:2	-		
1:0	sw_dc1_code1_cnt		
29:24	sw_scan_map_21		
23:18	sw_scan_map_22		
17:12	sw_scan_map_23		
11:6	sw_scan_map_24		
5:0	sw_scan_map_25		
swreg24			
31:2	sw_refer10_base	x	
31:2	sw_dct_strm3_base		x
1	sw_refer10_field_e	x	
0	sw_refer10_topc_e	x	
31:28	sw_dc1_code16_cnt		
27:24	sw_dc1_code15_cnt		
23:20	sw_dc1_code14_cnt		
19:16	sw_dc1_code13_cnt		
15:12	sw_dc1_code12_cnt		
11:8	sw_dc1_code11_cnt		
7:4	sw_dc1_code10_cnt		
3:0	sw_dc1_code9_cnt		
29:24	sw_scan_map_26		

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
23:18	sw_scan_map_27		
17:12	sw_scan_map_28		
11:6	sw_scan_map_29		
5:0	sw_scan_map_30		
swreg25			
31:2	sw_refer11_base	x	
31:2	sw_dct_strm4_base		x
1	sw_refer11_field_e	x	
0	sw_refer11_topc_e	x	
31:28	sw_dc2_code8_cnt		
27:24	sw_dc2_code7_cnt		
23:20	sw_dc2_code6_cnt		
19:16	sw_dc2_code5_cnt		
15:12	sw_dc2_code4_cnt		
11:8	sw_dc2_code3_cnt		
7	-		
6:4	sw_dc2_code2_cnt		
3:2	-		
1:0	sw_dc2_code1_cnt		
29:24	sw_scan_map_31		
23:18	sw_scan_map_32		
17:12	sw_scan_map_33		
11:6	sw_scan_map_34		
5:0	sw_scan_map_35		
swreg26			
31:2	sw_refer12_base	x	
31:2	sw_dct_strm5_base		x
1	sw_refer12_field_e	x	
0	sw_refer12_topc_e	x	
31:28	sw_dc2_code16_cnt		
27:24	sw_dc2_code15_cnt		
23:20	sw_dc2_code14_cnt		
19:16	sw_dc2_code13_cnt		
15:12	sw_dc2_code12_cnt		
11:8	sw_dc2_code11_cnt		
7:4	sw_dc2_code10_cnt		
3:0	sw_dc2_code9_cnt		
29:24	sw_scan_map_36		
23:18	sw_scan_map_37		
17:12	sw_scan_map_38		

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
11:6	sw_scan_map_39		
5:0	sw_scan_map_40		
swreg27			
31:2	sw_refer13_base	x	
1	sw_refer13_field_e	x	
0	sw_refer13_topc_e	x	
31:28	sw_dc3_code8_cnt		
27:24	sw_dc3_code7_cnt		
23:20	sw_dc3_code6_cnt		
19:16	sw_dc3_code5_cnt		
15:12	sw_dc3_code4_cnt		
11:8	sw_dc3_code3_cnt		
7	-		
6:4	sw_dc3_code2_cnt		
3:2	-		
1:0	sw_dc3_code1_cnt		
31:2	sw_bitpl_ctrl_base		x
swreg28			
31:2	sw_refer14_base	x	
31:2	sw_dct_strm6_base		x
1	sw_refer14_field_e	x	
0	sw_refer14_topc_e	x	
31:16	sw_ref_invd_cur_1		
15:0	sw_ref_invd_cur_0		
31:28	sw_dc3_code16_cnt		
27:24	sw_dc3_code15_cnt		
23:20	sw_dc3_code14_cnt		
19:16	sw_dc3_code13_cnt		
15:12	sw_dc3_code12_cnt		
11:8	sw_dc3_code11_cnt		
7:4	sw_dc3_code10_cnt		
3:0	sw_dc3_code9_cnt		
29:24	sw_scan_map_41		
23:18	sw_scan_map_42		
17:12	sw_scan_map_43		
11:6	sw_scan_map_44		
5:0	sw_scan_map_45		
swreg29			
31:2	sw_refer15_base	x	
31:2	sw_dct_strm7_base		x

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
1	sw_refer15_field_e	x	
0	sw_refer15_topc_e	x	
31:16	sw_ref_invd_cur_3		
15:0	sw_ref_invd_cur_2		
29:24	sw_scan_map_46		
23:18	sw_scan_map_47		
17:12	sw_scan_map_48		
11:6	sw_scan_map_49		
5:0	sw_scan_map_50		
swreg30			
31:16	sw_refer1_nbr	x	
15:0	sw_refer0_nbr	x	
31:16	sw_ref_dist_cur_1		
15:0	sw_ref_dist_cur_0		
31	sw_filt_type		x
30:28	sw_filt_sharpness		x
27:21	sw_filt_mb_adj_0		x
20:14	sw_filt_mb_adj_1		x
13:7	sw_filt_mb_adj_2		x
6:0	sw_filt_mb_adj_3		x
swreg31			
31:16	sw_refer3_nbr	x	
15:0	sw_refer2_nbr	x	
29:24	sw_scan_map_51		
23:18	sw_scan_map_52		
17:12	sw_scan_map_53		
11:6	sw_scan_map_54		
5:0	sw_scan_map_55		
31:16	sw_ref_dist_cur_3		
15:0	sw_ref_dist_cur_2		
27:21	sw_filt_ref_adj_0		x
20:14	sw_filt_ref_adj_1		x
13:7	sw_filt_ref_adj_2		x
6:0	sw_filt_ref_adj_3		x
swreg32			
31:16	sw_refer5_nbr	x	
15:0	sw_refer4_nbr	x	
29:24	sw_scan_map_56		
23:18	sw_scan_map_57		
17:12	sw_scan_map_58		

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
11:6	sw_scan_map_59		
5:0	sw_scan_map_60		
31:16	sw_ref_invd_col_1		
15:0	sw_ref_invd_col_0		
23:18	sw_filt_level_0		x
17:12	sw_filt_level_1		x
11:6	sw_filt_level_2		x
5:0	sw_filt_level_3		x
swreg33			
31:16	sw_refer7_nbr	x	
15:0	sw_refer6_nbr	x	
29:24	sw_scan_map_61		
23:18	sw_scan_map_62		
17:12	sw_scan_map_63		
31:16	sw_ref_invd_col_3		
15:0	sw_ref_invd_col_2		
31:27	sw_quant_delta_0		x
26:22	sw_quant_delta_1		x
21:11	sw_quant_0		x
10:0	sw_quant_1		x
swreg34			
31:16	sw_refer9_nbr	x	
15:0	sw_refer8_nbr	x	
31:22	sw_pred_bc_tap_0_3		x
21:12	sw_pred_bc_tap_1_0		x
11:2	sw_pred_bc_tap_1_1		x
swreg35			
31:16	sw_refer11_nbr	x	
15:0	sw_refer10_nbr	x	
31:22	sw_pred_bc_tap_1_2		x
21:12	sw_pred_bc_tap_1_3		x
11:2	sw_pred_bc_tap_2_0		x
swreg36			
31:16	sw_refer13_nbr	x	
15:0	sw_refer12_nbr	x	
31:22	sw_pred_bc_tap_2_1		x
21:12	sw_pred_bc_tap_2_2		x
11:2	sw_pred_bc_tap_2_3		x
swreg37			
31:16	sw_refer15_nbr	x	

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
15:0	sw_refer14_nbr	x	
31:22	sw_pred_bc_tap_3_0		x
21:12	sw_pred_bc_tap_3_1		x
11:2	sw_pred_bc_tap_3_2		x
swreg38			
31:0	sw_refer_lterm_e	x	
31:22	sw_pred_bc_tap_3_3		x
21:12	sw_pred_bc_tap_4_0		x
11:2	sw_pred_bc_tap_4_1		x
swreg39			
31:0	sw_refer_valid_e	x	
31:22	sw_pred_bc_tap_4_2		x
21:12	sw_pred_bc_tap_4_3		x
11:2	sw_pred_bc_tap_5_0		x
swreg40			
31:2	sw_qtable_base	x	x
swreg41			
31:2	sw_dir_mv_base	x	x
swreg42			
29:25	sw_binit_rlist_b2	x	
24:20	sw_binit_rlist_f2	x	
19:15	sw_binit_rlist_b1	x	
14:10	sw_binit_rlist_f1	x	
9:5	sw_binit_rlist_b0	x	
4:0	sw_binit_rlist_f0	x	
31:22	sw_pred_bc_tap_5_1		x
21:12	sw_pred_bc_tap_5_2		x
11:2	sw_pred_bc_tap_5_3		x
31:24	sw_weight_qp_1		
23:21	sw_ref_delta_col_0		
20:18	sw_ref_delta_col_1		
17:15	sw_ref_delta_col_2		
14:12	sw_ref_delta_col_3		
11:9	sw_ref_delta_cur_0		
8:6	sw_ref_delta_cur_1		
5:3	sw_ref_delta_cur_2		
2:0	sw_ref_delta_cur_3		
swreg43			
29:25	sw_binit_rlist_b5	x	
24:20	sw_binit_rlist_f5	x	

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
19:15	sw_binit_rlist_b4	x	
14:10	sw_binit_rlist_f4	x	
9:5	sw_binit_rlist_b3	x	
4:0	sw_binit_rlist_f3	x	
31:22	sw_pred_bc_tap_6_0		x
21:12	sw_pred_bc_tap_6_1		x
11:2	sw_pred_bc_tap_6_2		x
31:24	sw_weight_qp_2		
23:16	sw_weight_qp_3		
15:8	sw_weight_qp_4		
7:0	sw_weight_qp_5		
swreg44			
29:25	sw_binit_rlist_b8	x	
24:20	sw_binit_rlist_f8	x	
19:15	sw_binit_rlist_b7	x	
14:10	sw_binit_rlist_f7	x	
9:5	sw_binit_rlist_b6	x	
4:0	sw_binit_rlist_f6	x	
31:22	sw_pred_bc_tap_6_3		x
21:12	sw_pred_bc_tap_7_0		x
11:2	sw_pred_bc_tap_7_1		x
26	sw_dec_avsp_ena		
25	sw_weight_qp_e		
24:23	sw_weight_qp_model		
22	sw_avs_aec_e		
21	sw_no_fwd_ref_e		
20	sw_pb_field_enhanced_e		
19:14	sw_qp_delta_cb		
13:8	sw_qp_delta_cr		
7:0	sw_weight_qp_0		
swreg45			
29:25	sw_binit_rlist_b11	x	
24:20	sw_binit_rlist_f11	x	
19:15	sw_binit_rlist_b10	x	
14:10	sw_binit_rlist_f10	x	
9:5	sw_binit_rlist_b9	x	
4:0	sw_binit_rlist_f9	x	
31:22	sw_pred_bc_tap_7_2		x
21:12	sw_pred_bc_tap_7_3		x
11:10	sw_pred_tap_2_m1		x

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
9:8	sw_pred_tap_2_4		x
7:6	sw_pred_tap_4_m1		x
5:4	sw_pred_tap_4_4		x
3:2	sw_pred_tap_6_m1		x
1:0	sw_pred_tap_6_4		x
31:0	sw_dir_mv_base2		
swreg46			
31:30			
29:25	sw_binit_rlist_b14	x	
24:20	sw_binit_rlist_f14	x	
19:15	sw_binit_rlist_b13	x	
14:10	sw_binit_rlist_f13	x	
9:5	sw_binit_rlist_b12	x	
4:0	sw_binit_rlist_f12	x	
31:27	sw_quant_delta_2		x
26:22	sw_quant_delta_3		x
21:11	sw_quant_2		x
10:0	sw_quant_3		x
swreg47			
31:30	-		
29:25	sw_pinit_rlist_f3	x	
24:20	sw_pinit_rlist_f2	x	
19:15	sw_pinit_rlist_f1	x	
14:10	sw_pinit_rlist_f0	x	
9:5	sw_binit_rlist_b15	x	
4:0	sw_binit_rlist_f15	x	
31:27	sw_quant_delta_4		x
21:11	sw_quant_4		x
10:0	sw_quant_5		x
swreg48			
31:23	sw_startmb_x		x
22:14	sw_startmb_y		x
13:12	sw_error_conc_mode		x
11:0	-		
swreg49			
31:22	sw_pred_bc_tap_0_0	x	
21:12	sw_pred_bc_tap_0_1	x	
11:2	sw_pred_bc_tap_0_2	x	
1:0	-		
swreg50			

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
31	SW_DEC_MPEG2_PROF		
30:29	SW_DEC_VC1_PROF		
28	SW_DEC_JPEG_PROF		
27:26	SW_DEC_MPEG4_PROF		
25:24	SW_DEC_H264_PROF	x	
23	SW_DEC_VP6_PROF		
22	SW_DEC_PJPEG_EXIST		
21	SW_DEC_OBUFF_LEVEL	x	x
20	SW_REF_BUFF_EXIST	x	x
19:16	SW_DEC_BUS_STRD	x	x
15:14	SW_DEC_SYNTH_LAN	x	x
13:12	SW_DEC_BUS_WIDTH	x	x
11	SW_DEC_SOREN_PROF		
10:0	SW_DEC_MAX_OWIDTH	x	x
swreg51			
31	sw_refbu_e	x	x
30:19	sw_refbu_thr	x	x
18:14	sw_refbu_picid	x	x
13	sw_refbu_eval_e	x	x
12	sw_refbu_fparmod_e	x	x
11:9	-		
8:0	sw_refbu_y_offset	x	x
swreg52			
31:16	sw_refbu_hit_sum	x	x
15:0	sw_refbu_intra_sum	x	x
swreg53			
31:22	-		
21:0	sw_refbu_y_mv_sum	x	x
swreg54			
31	SW_DEC_JPEG_EXTENS		
30	SW_DEC_REFBU_ILACE	x	
29	SW_DEC_DIVX_PROF		
28:0	SW_REF_BUFF2_EXIST	x	x
27:26	SW_DEC_RV_PROF		
25	SW_DEC_RTL_ROM		
24	SW_DEC_VP7_PROF		x
23	SW_DEC_VP8_PROF		x
22	SW_DEC_AVS_PROF		
21:20	SW_DEC_MVC_PROF	x	
19	SW_DEC_WEBP_E		x

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
18:17	SW_DEC_TILED_L	x	x
16	SW_DEC_VP8S_ARCH		x
15:14	SW_DEC_MAX_OW_EXT	x	x
13:12	SW_DEC_ERRCO_LEVEL		x
11	SW_VP8_STRIDE_E		x
10	SW_DPB_FIELD_E	x	x
9:7	SW_DEC_CORE_AM	x	x
6:0	-		
swreg55			
31	sw_refbu2_buf_e	x	x
30:19	sw_refbu2_thr	x	x
18:14	sw_refbu2_picid	x	x
13:0	sw_apf_threshold	x	x
swreg56			
31:16	sw_refbu_top_sum	x	x
15:0	sw_refbu_bot_sum	x	x
swreg57			
31	fuse_dec_h264	x	
30	fuse_dec_mpeg4		
29	fuse_dec_mpeg2		
28	fuse_dec_sorenson		
27	fuse_dec_jpeg		
26	fuse_dec_vp6		
25	fuse_dec_vc1		
24	fuse_dec_pjpeg		
23	fuse_dec_divx		
22	fuse_dec_rv		
21	fuse_dec_vp7		x
20	fuse_dec_vp8		x
19	fuse_dec_avs		
18	fuse_dec_mvc	x	
17	-		
16	fuse_dec_maxw_4k	x	x
15	fuse_dec_maxw_1920	x	x
14	fuse_dec_maxw_1280	x	x
13	fuse_dec_maxw_720	x	x
12	fuse_dec_maxw_352	x	x
11:8	-		
7	fuse_dec_refbuffer	x	x
6:0	-		

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
swreg58			
31	sw_serv_merge_dis	x	x
30	sw_dec_multicore_e	x	x
29	sw_dec_writestat_e	x	x
28:27	sw_dec_mc_pollmode	x	x
26:17	sw_dec_mc_polltime	x	x
16:0	-		
swreg59			
31:2	sw_dec_ch8pix_base	x	
1:0	-		
swreg102			
31:2	sw_dec_ch_base	x	
1	-		
0	sw_ch_base_e	x	
swreg103			
31:2	sw_refer0_ch_base	x	
1:0	-		
swreg104			
31:2	sw_refer1_ch_base	x	
1:0	-		
swreg105			
31:2	sw_refer2_ch_base	x	
1:0	-		
swreg106			
31:2	sw_refer3_ch_base	x	
1:0	-		
swreg107			
31:2	sw_refer4_ch_base	x	
1:0	-		
swreg108			
31:2	sw_refer5_ch_base	x	
1:0	-		
swreg109			
31:2	sw_refer6_ch_base	x	
1:0	-		
swreg110			
31:2	sw_refer7_ch_base	x	
1:0	-		
swreg111			
31:2	sw_refer8_ch_base	x	

Table continues on the next page...

Bit	Name	Dec Modes	
		H.264	VP7 / VP8
1:0	-		
swreg112			
31:2	sw_refer9_ch_base	x	
1:0	-		
swreg113			
31:2	sw_refer10_ch_base	x	
1:0	-		
swreg114			
31:2	sw_refer11_ch_base	x	
1:0	-		
swreg115			
31:2	sw_refer12_ch_base	x	
1:0	-		
swreg116			
31:2	sw_refer13_ch_base	x	
1:0	-		
swreg117			
31:2	sw_refer14_ch_base	x	
1:0	-		
swreg118			
31:2	sw_refer15_ch_base	x	
1:0	-		

16.1.6 VPU G1 Memory Map/Register Definition

16.1.6.1 VPU_G1 common registers descriptions

16.1.6.1.1 VPU_G1 common registers memory map

VPU_G1 base address: 3830_0000h

Offset	Register	Width (In bits)	Access	Reset value
4h	Interrupt register decoder (SWREG1)	32	RW	0000_0000h
8h	Device configuration register decoder (SWREG2)	32	RW	0000_0000h

Table continues on the next page...

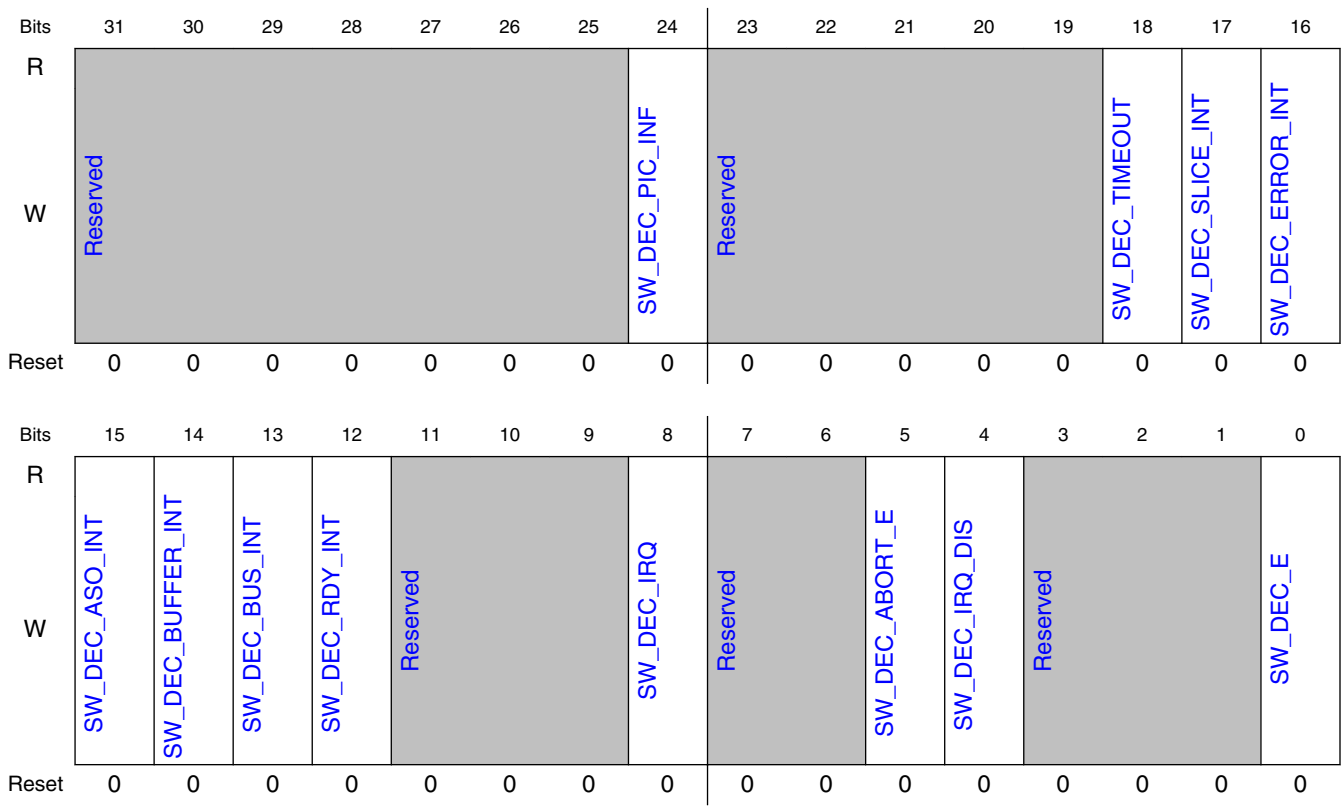
Offset	Register	Width (In bits)	Access	Reset value
Ch	Decoder control register 0 (decmode,picture type etc) (SWREG3)	32	RW	0000_0000h
30h	Base address for RLC data (RLC) / stream start address/decoded end addr register (VLC) (SWREG12)	32	RW	0000_0000h
34h	Base address for decoded picture (SWREG13)	32	RW	0000_0000h
A0h	Base address for standard dependent tables (SWREG40)	32	RW	0000_0000h
A4h	Base address for direct mode motion vectors (SWREG41)	32	RW	0000_0000h
C0h	Error concealment register (SWREG48)	32	RW	0000_0000h
C4h	Prediction filter tap register for H264 (SWREG49)	32	RW	0000_0000h
C8h	Synthesis configuration register decoder 0 (SWREG50)	32	RO	Table 16-2
CCh	Reference picture buffer control register (SWREG51)	32	RW	0000_0000h
D0h	Reference picture buffer information register 1 (SWREG52)	32	RO	0000_0000h
D4h	Reference picture buffer information register 2 (SWREG53)	32	RO	0000_0000h
D8h	Synthesis configuration register decoder 1 (SWREG54)	32	RO	0000_0000h
DCh	Reference picture buffer 2 / Advanced prefetch control register (SWREG55)	32	RW	0000_0000h
E0h	Reference buffer information register 3 (SWREG56)	32	RO	0000_0000h
E4h	Decoder fuse register (SWREG57)	32	RO	Table 16-2
E8h	Device configuration register decoder 2 + Multi core control register (SWREG58)	32	RW	0000_0000h
ECh	H264 Chrominance 8 pixel interleaved data base (SWREG59)	32	RW	0000_0000h
198h	Base address for H264 decoded chroma picture (SWREG102)	32	RW	0000_0000h
19Ch	Base address for reference chroma picture index 0 (SWREG103)	32	RW	0000_0000h
1A0h	Base address for reference chroma picture index 1 (SWREG104)	32	RW	0000_0000h
1A4h	Base address for reference chroma picture index 2 (SWREG105)	32	RW	0000_0000h
1A8h	Base address for reference chroma picture index 3 (SWREG106)	32	RW	0000_0000h
1ACh	Base address for reference chroma picture index 4 (SWREG107)	32	RW	0000_0000h
1B0h	Base address for reference chroma picture index 5 (SWREG108)	32	RW	0000_0000h
1B4h	Base address for reference chroma picture index 6 (SWREG109)	32	RW	0000_0000h
1B8h	Base address for reference chroma picture index 7 (SWREG110)	32	RW	0000_0000h
1BCh	Base address for reference chroma picture index 8 (SWREG111)	32	RW	0000_0000h
1C0h	Base address for reference chroma picture index 9 (SWREG112)	32	RW	0000_0000h
1C4h	Base address for reference chroma picture index 10 (SWREG113)	32	RW	0000_0000h
1C8h	Base address for reference chroma picture index 11 (SWREG114)	32	RW	0000_0000h
1CCh	Base address for reference chroma picture index 12 (SWREG115)	32	RW	0000_0000h
1D0h	Base address for reference chroma picture index 13 (SWREG116)	32	RW	0000_0000h
1D4h	Base address for reference chroma picture index 14 (SWREG117)	32	RW	0000_0000h
1D8h	Base address for reference chroma picture index 15 (SWREG118)	32	RW	0000_0000h

16.1.6.1.2 Interrupt register decoder (SWREG1)

16.1.6.1.2.1 Offset

Register	Offset
SWREG1	4h

16.1.6.1.2.2 Diagram



16.1.6.1.2.3 Fields

Field	Function
31-25 —	Reserved.
24 SW_DEC_PIC_INF	B slice detected. This signal is driven high during picture ready interrupt if B-type slice is found. This bit does not launch interrupt but is used to inform SW about h264 tools.
23-19 —	Reserved.

Table continues on the next page...

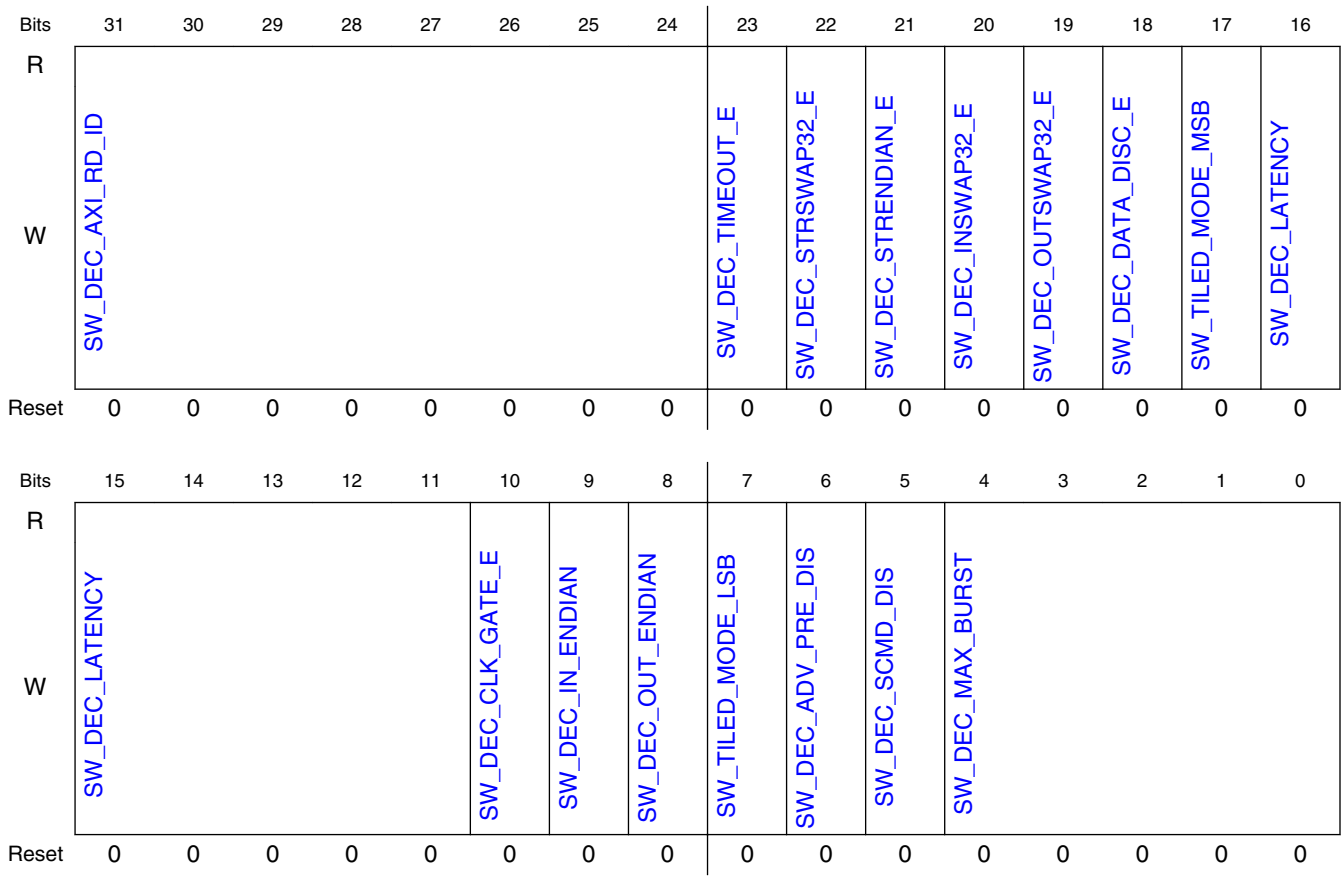
Field	Function
18 SW_DEC_TIMEOUT	Interrupt status bit decoder timeout. When high the decoder has been idling for too long. HW will self reset. Possible only if timeout interrupt is enabled
17 SW_DEC_SLICE_INT	Interrupt status bit dec_slice_decoded. When high SW must set new base addresses for sw_dec_out_base and sw_jpg_ch_out_base before resetting this status bit. Used for VP8 web-p modes
16 SW_DEC_ERROR_INT	Interrupt status bit input stream error. When high, an error is found in input data stream decoding. HW will self reset.
15 SW_DEC_ASO_INT	H264: Interrupt status bit ASO (Arbitrary Slice Ordering) detected. When high, ASO detected in input data stream decoding. HW will self reset. VP8: Error detected in Residual data. HW returns MB number in error concealment register for MB it detected it
14 SW_DEC_BUFFER_INT	Interrupt status bit input buffer empty. When high, input stream buffer is empty but picture is not ready. HW will not self reset.
13 SW_DEC_BUS_INT	Interrupt status bit bus. Error response from bus.
12 SW_DEC_RDY_INT	Interrupt status bit decoder. When this bit is high decoder has decoded a picture. HW will self reset.
11-9 —	Reserved.
8 SW_DEC_IRQ	Decoder IRQ. When high, decoder requests an interrupt. SW will reset this after interrupt is handled.
7-6 —	Reserved.
5 SW_DEC_ABORT_E	Abort decoding enable. Setting this bit high will cause HW to abort decoding and safely to reset itself down. After abort is complete the corresponding interrupt status is set and this bit is set low as well as the decoder enable.
4 SW_DEC_IRQ_DIS	Decoder IRQ disable. When high, there are no interrupts concerning decoder from HW. Polling must be used to see the interrupt statuses.
3-1 —	Reserved.
0 SW_DEC_E	Decoder enable. Setting this bit high will start the decoding operation. HW will reset this when picture is processed or ASO or stream error is detected or bus error or timeout interrupt is given.

16.1.6.1.3 Device configuration register decoder (SWREG2)

16.1.6.1.3.1 Offset

Register	Offset
SWREG2	8h

16.1.6.1.3.2 Diagram



16.1.6.1.3.3 Fields

Field	Function
31-24 SW_DEC_AXI_RD_ID	Read ID used for decoder reading services in AXI bus (if connected to AXI).
23 SW_DEC_TIMEOUT_E	Timeout interrupt enable. If enabled HW may return timeout interrupt in case HW gets stuck while decoding picture.
22	Decoder input 32bit data swap for stream data (may be used for 64 bit environment): 0b - no swapping of 32 bit words

Table continues on the next page...

Field	Function
SW_DEC_STR SWAP32_E	1b - 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
21 SW_DEC_STR ENDIAN_E	Decoder input endian mode for stream data: 0b - Big endian (0-1-2-3 order) 1b - Little endian (3-2-1-0 order)
20 SW_DEC_INSW AP32_E	Decoder input 32bit data swap for other than stream data (may be used for 64 bit environment): 0b - no swapping of 32 bit words 1b - 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
19 SW_DEC_OUT SWAP32_E	Decoder output 32bit data swap (may be used for 64 bit environment): 0b - no swapping of 32 bit words 1b - 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
18 SW_DEC_DAT A_DISC_E	Data discard enable. Precise burst lengths are used with reading services. Extra data is discarded internally. Note. If AHB maxburst 17 is used data discard cannot be enabled (causes conflict)
17 SW_TILED_MO DE_MSB	Tiled mode msb. Concatenated to Tiled mode lsb which form 2 bit tiled mode. 0b - Tiled mode not enabled 1b - Tiled mode enabled for 8x4 tile size
16-11 SW_DEC_LATE NCY	Decoder master interface additional latency. Can be used to slow down decoder HW between services in steps of 8 clock cycles: 0 = no latency 1 = minimum 8 cycles of IDLE between services 2 = minimum 16 cycles of IDLE between services . . . 63 = minimum latency of 504 cycles of IDLE between services
10 SW_DEC_CLK_ GATE_E	Decoder dynamic clock gating enable: 0b - Clock is running for all structures 1b - Clock is gated for decoder structures that are not used. Note: Clock gating value can be changed only when decoder is disabled.
9 SW_DEC_IN_E NDIAN	Decoder input endian mode for other than stream data: 0b - Big endian (0-1-2-3 order) 1b - Little endian (3-2-1-0 order)
8 SW_DEC_OUT _ENDIAN	Decoder output endian mode: 0b - Big endian (0-1-2-3 order) 1b - Little endian (3-2-1-0 order)
7 SW_TILED_MO DE_LSB	Tiled mode lsb. Concatenated to Tiled mode msb which form 2 bit tiled mode. Defined in tiled_mode_msb
6 SW_DEC_ADV_ PRE_DIS	Advanced PREFETCH mode disable (advanced reference picture reading mode for video)

Table continues on the next page...

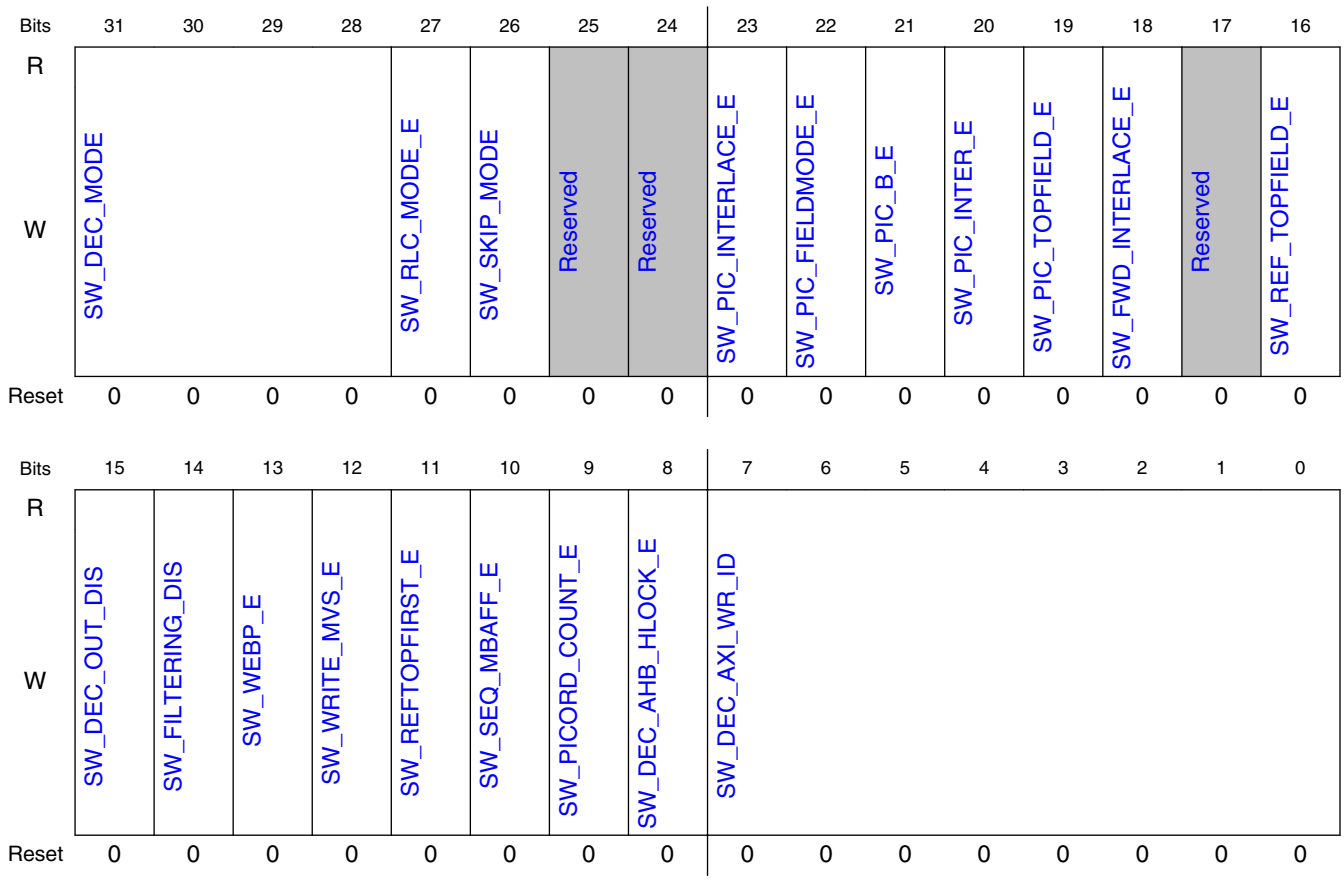
Field	Function
5 SW_DEC_SCMD_DIS	9170 decoder and later->: AXI Single Command Multiple Data disable. 9170 axi wrapper supports this mode by default (where only the first addresses of the burst are given from address generator). This bit is used to disable the feature (possible SW workaround if something is not working correctly)
4-0 SW_DEC_MAX_BURST	Maximum burst length for decoder bus transactions. Valid values: AHB: 1, 4, 8, 16 and 17. Other values will result in INCR type (undefined length) for all transfers. INCR used for lengths 2 and 3. 17 = fixed bursts are used when possible. INCR bursts are used for lengths 2, 3, 5, 6, 7, 9, 10, 11, 12, 13, 14 and 15. OCP: 16-31 AXI: 1-16

16.1.6.1.4 Decoder control register 0 (decmode,picture type etc) (SWREG3)

16.1.6.1.4.1 Offset

Register	Offset
SWREG3	Ch

16.1.6.1.4.2 Diagram



16.1.6.1.4.3 Fields

Field	Function
31-28 SW_DEC_MODE	Decoding mode: 0000b - H.264 0001b - Reserved 0010b - Reserved 0011b - Reserved 0100b - Reserved 0101b - Reserved 0110b - Reserved 0111b - Reserved 1000b - Reserved 1001b - Reserved 1010b - VP8 1011b - Reserved 1100b - Reserved 1101b - Reserved 1110b - Reserved 1111b - Reserved
27	RLC mode enable: 0b - HW decodes video from bit stream (VLC mode) + side information

Table continues on the next page...

Field	Function
SW_RLC_MODE_E	1b - HW decodes video from RLC input data + side information (Differential MV's, separate DC coeffs, Intra 4x4 modes, MB control). Valid only for H.264 Baseline.
26 SW_SKIP_MODE	VP8: • 0 : HW decodes mb_coeff_skip -flag • 1 : HW does not decode mb_coeff_skip -flag
25 —	Reserved.
24 —	Reserved.
23 SW_PIC_INTERLACE_E	Coding mode of the current picture: 0b - progressive 1b - interlaced
22 SW_PIC_FIELDMODE_E	Structure of the current picture (residual structure) 0b - Frame structure. For H264, this means MBAFF structured picture for interlaced sequence 1b - Field structure
21 SW_PIC_B_E	B picture enable for current picture: 0b - picture type is I or P depending on sw_pic_inter_e 1b - picture type is B depending on sw_pic_inter_e (not valid for H264 since it is slice based information)
20 SW_PIC_INTER_E	Picture type. Please also see SW_PIC_B_E. 0b - Intra type (I) 1b - Inter type (P)
19 SW_PIC_TOPFIELD_E	If field structure is enabled, this bit informs which one of the fields is being decoded: 0b - bottom field 1b - top field
18 SW_FWD_INTERLACE_E	Coding mode of forward reference picture **NOTE:** For backward reference picture, the coding mode is always the same as for current picture. 0b - progressive 1b - interlaced
17 —	Reserved.
16 SW_REF_TOPFIELD_E	Indicates which field should be used as reference if sw_ref_frames = '0': 0b - bottom field 1b - top field
15 SW_DEC_OUT_DIS	Disable decoder output picture writing: 0b - Decoder output picture is written to external memory 1b - Decoder output picture is not written to external memory
14 SW_FILTERING_DIS	De-block filtering disable: 0b - filtering is enabled for current picture 1b - filtering is disabled for current picture
13 SW_WEBP_E	SW_WEBP_E: Webp enable for VP8: • '0' = Normal VP8 • '1' = WEBP picture SW_MVC_E: Multi View Coding enable. Possible for H264 only.

Table continues on the next page...

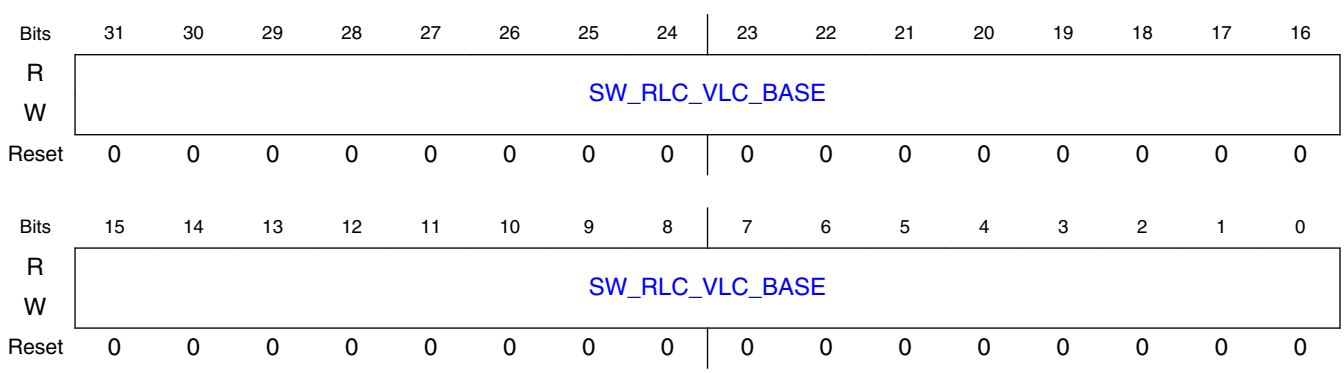
Field	Function
12 SW_WRITE_MVS_E	Direct mode motion vector write enable for current picture / VPX motion vector write enable for error concealment purposes: 0b - Writing disabled for current picture 1b - The direct mode motion vectors are written to external memory. H264 direct mode motion vectors are written to DPB aside with the corresponding reference picture. Other decoding mode dir mode mvs are written to external memory starting from sw_dir_mv_base.
11 SW_REFTOPFIRST_E	Indicates which FWD reference field has been decoded first. 0b - FWD reference bottom field 1b - FWD reference top field
10 SW_SEQ_MBAFF_E	Sequence includes MBAFF coded pictures
9 SW_PICORD_COUNT_E	h264_high config: Picture order count table read enable. If enabled HW will read picture order counts from memory in the beginning of picture
8 SW_DEC_AHB_HLOCK_E	AHB master HLOCK enable. When high the service is locked to decoder as long as it needs the bus (whenever decoder requests the bus it will be granted)
7-0 SW_DEC_AXI_WR_ID	Write ID used for decoder writing services in AXI bus (if connected to AXI)

16.1.6.1.5 Base address for RLC data (RLC) / stream start address/ decoded end addr register (VLC) (SWREG12)

16.1.6.1.5.1 Offset

Register	Offset
SWREG12	30h

16.1.6.1.5.2 Diagram



16.1.6.1.5.3 Fields

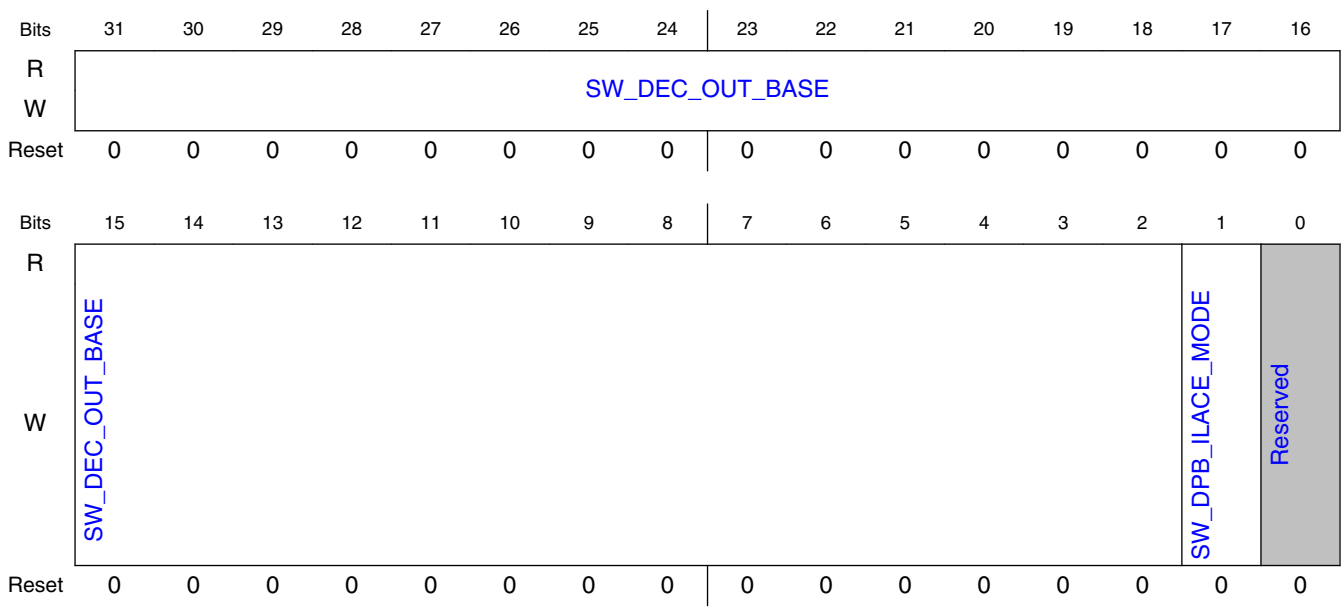
Field	Function
31-0 SW_RLC_VLC_BASE	RLC mode: Base address for RLC data (swreg3.sw_rlc_mode_e = 1). VLC mode: Stream start address / end addr+I288ess with byte precision (swreg4.rlc_mode_en = 0), start bit number in swreg5.stream_start_bit. When sw_dec_buffer_int is high or sw_dec_e is low this register contains HW return value of last_byte_address where stream has been read (and used) in accuracy of byte. For debug purposes the last_byte_address is also written when stream error/ASO is detected even though it may not be accurate. VP8: This base address is used as sw_dct_strm0_base including DCT stream for MB rows 0,2n.

16.1.6.1.6 Base address for decoded picture (SWREG13)

16.1.6.1.6.1 Offset

Register	Offset
SWREG13	34h

16.1.6.1.6.2 Diagram



16.1.6.1.6.3 Fields

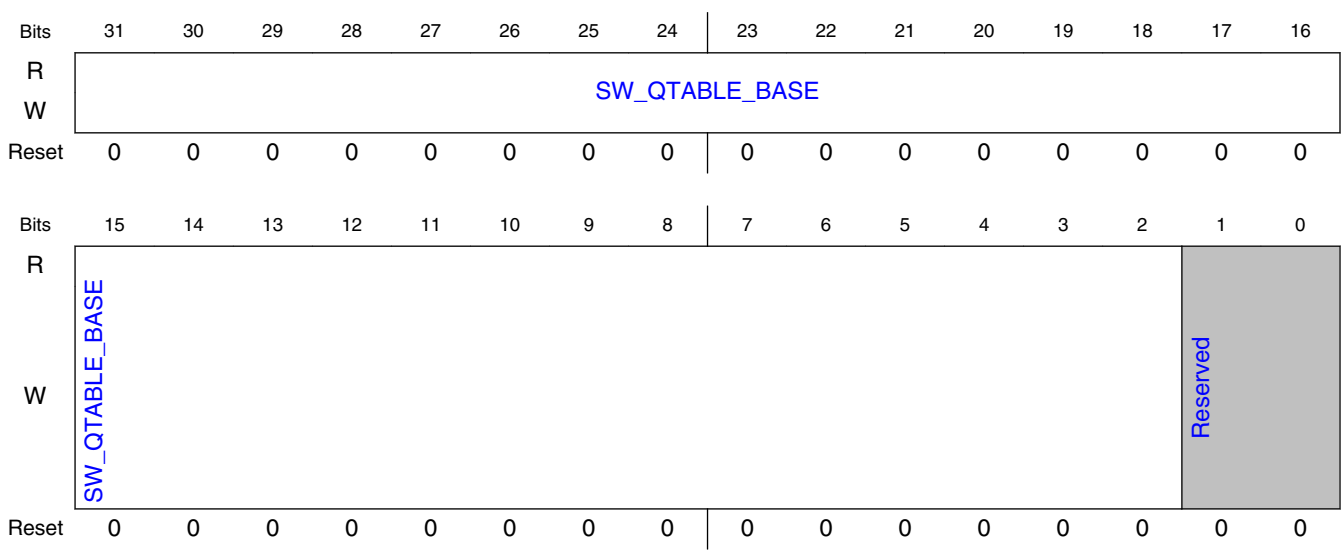
Field	Function
31-2 SW_DEC_OUT_BASE	Video: Base address for decoder output picture. Points directly to start of decoder output picture or field.
1 SW_DPB_ILACE_MODE	DPB ilaced mode: '0' : DPB consist of ilaced/progressive frames '1' : DPB consist of progressive frames / separate fields. This mode requires config support from HW
0 —	Reserved.

16.1.6.1.7 Base address for standard dependent tables (SWREG40)

16.1.6.1.7.1 Offset

Register	Offset
SWREG40	A0h

16.1.6.1.7.2 Diagram



16.1.6.1.7.3 Fields

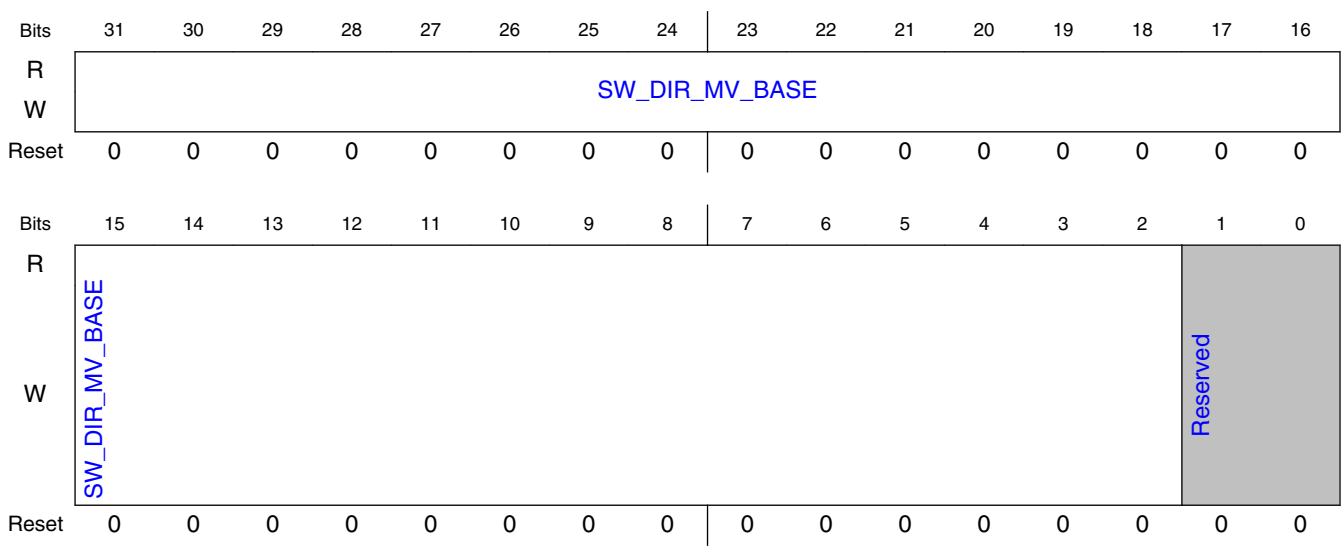
Field	Function
31-2 SW_QTABLE_BASE	Base address for standard dependent tables: H.264: base address for various tables VP8: base address for stream decoding tables
1-0 —	Reserved.

16.1.6.1.8 Base address for direct mode motion vectors (SWREG41)

16.1.6.1.8.1 Offset

Register	Offset
SWREG41	A4h

16.1.6.1.8.2 Diagram



16.1.6.1.8.3 Fields

Field	Function
31-2	Direct mode motion vector write/read base address. For H264 this is used only for direct mode motion vector write base. VP8: Motion vectors are written for error concealment purposes if sw_write_mvs is high. In error concealment mode motion vectors are read from this base address

Table continues on the next page...

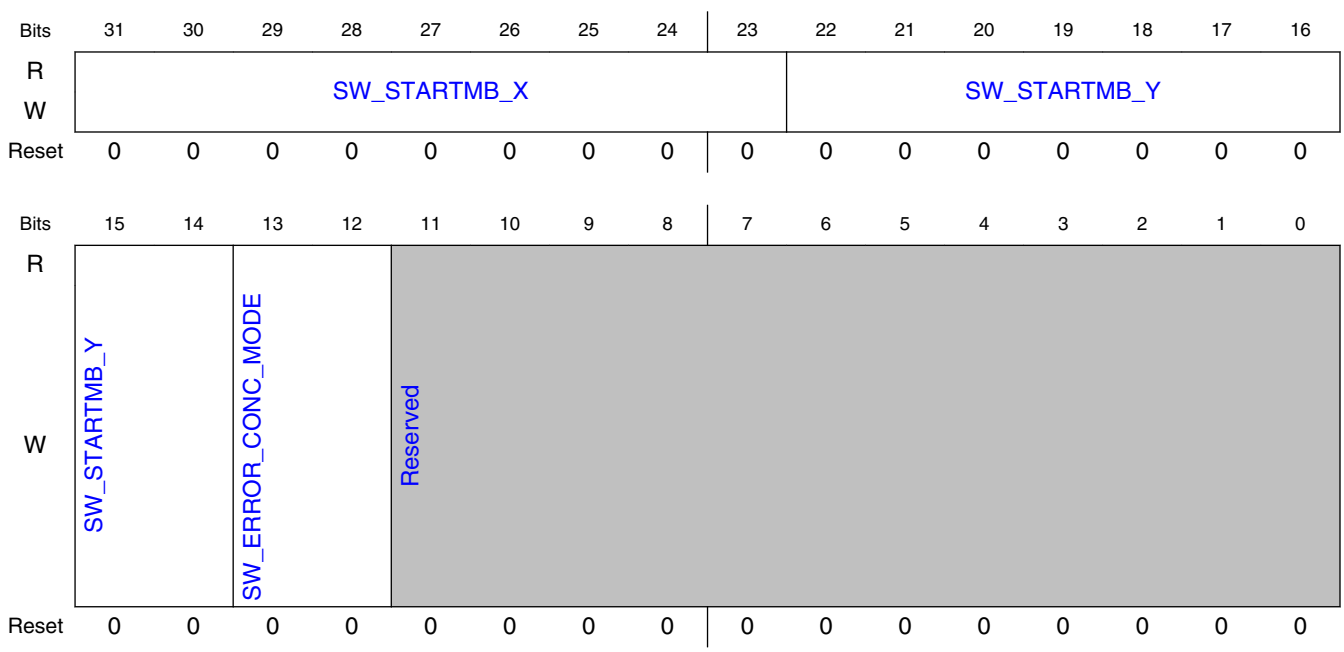
Field	Function
SW_DIR_MV_BASE	
1-0 —	Reserved.

16.1.6.1.9 Error concealment register (SWREG48)

16.1.6.1.9.1 Offset

Register	Offset
SWREG48	C0h

16.1.6.1.9.2 Diagram



16.1.6.1.9.3 Fields

Field	Function
31-23 SW_STARTMB_X	Start MB from SW for X dimension. Used in error concealment case as HW return value if HW founds an error or in HW init mb for error concealment if SW enables error concealment

Table continues on the next page...

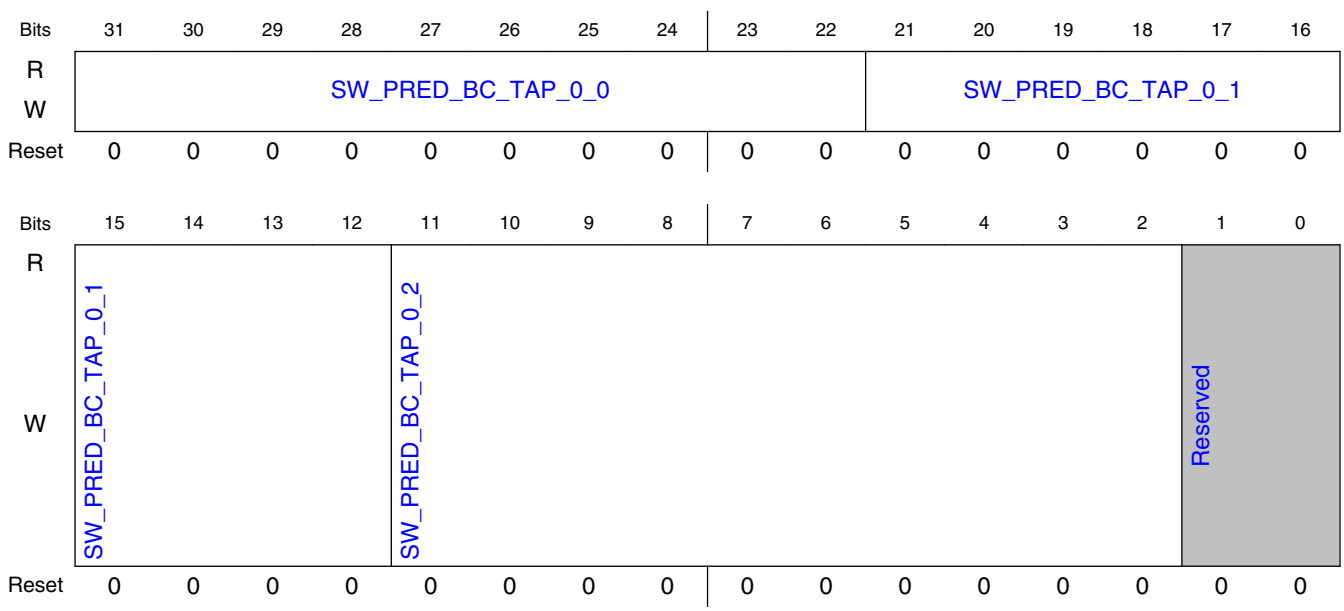
Field	Function
22-14 SW_STARTMB_Y	Start MB from SW for Y dimension. Used in error concealment case as HW return value if HW founds an error or in HW init mb for error concealment if SW enables error concealment
13-12 SW_ERROR_CONC_MODE	Error concealment mode: 00b - disabled (normal decoding mode) 01b - enabled for direct mode MV usage starting from MB defined by sw_startmb_x, sw_startmb_y
11-0 —	Reserved.

16.1.6.1.10 Prediction filter tap register for H264 (SWREG49)

16.1.6.1.10.1 Offset

Register	Offset
SWREG49	C4h

16.1.6.1.10.2 Diagram



16.1.6.1.10.3 Fields

Field	Function
31-22	Prediction filter set 0, tap 0

Table continues on the next page...

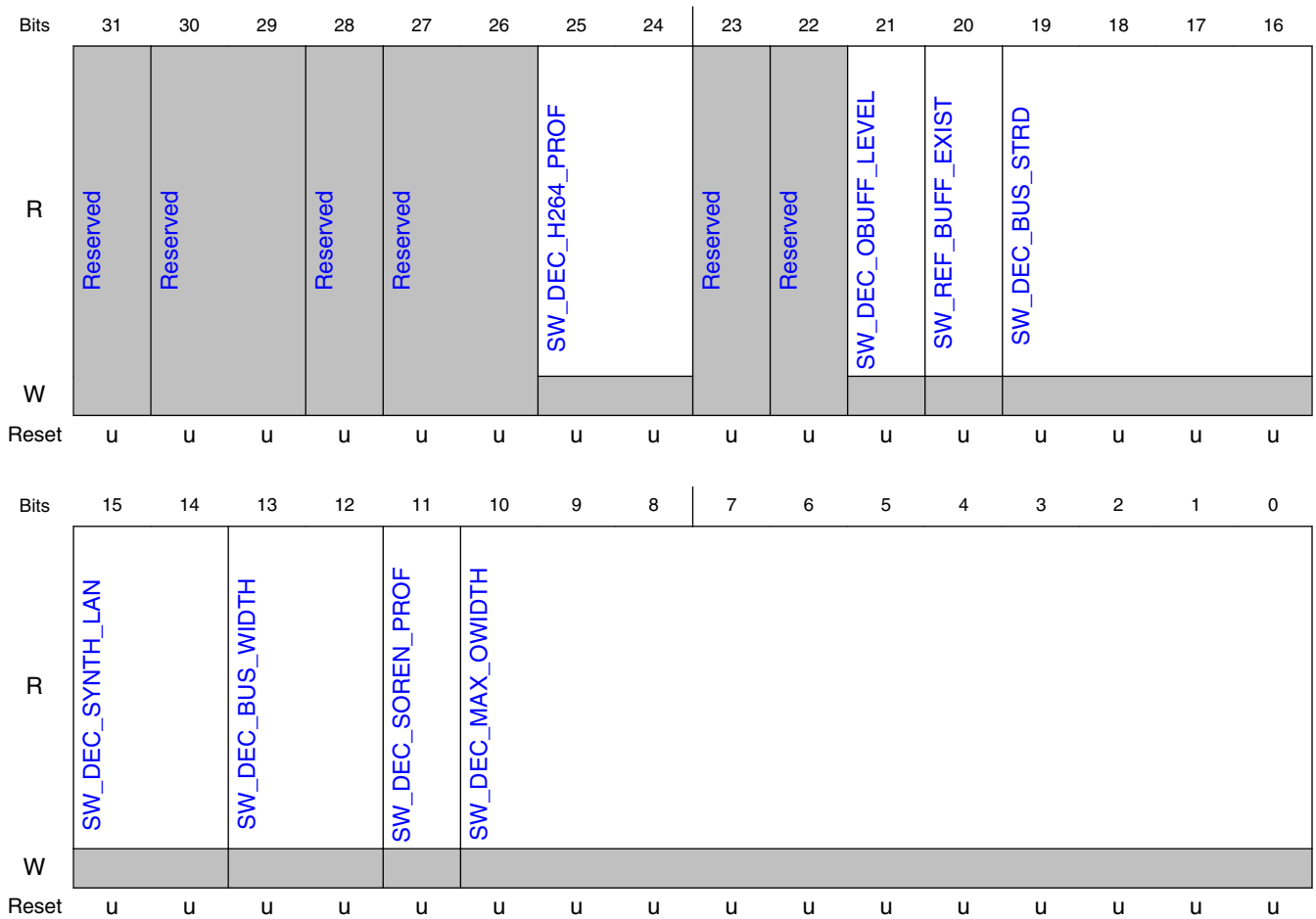
Field	Function
SW_PRED_BC_TAP_0_0	
21-12 SW_PRED_BC_TAP_0_1	Prediction filter set 0, tap 1
11-2 SW_PRED_BC_TAP_0_2	Prediction filter set 0, tap 2
1-0 —	Reserved.

16.1.6.1.11 Synthesis configuration register decoder 0 (SWREG50)

16.1.6.1.11.1 Offset

Register	Offset
SWREG50	C8h

16.1.6.1.11.2 Diagram



16.1.6.1.11.3 Fields

Field	Function
31 —	Reserved.
30-29 —	Reserved.
28 —	Reserved.
27-26 —	Reserved.
25-24 SW_DEC_H264_PROF	Decoding format support, H.264 00b - not supported 01b - supported up to baseline profile 10b - supported up to high profile labeled stream with restricted high profile tools (Tools that are used in Hantro 7280, 8270 encoder)

Table continues on the next page...

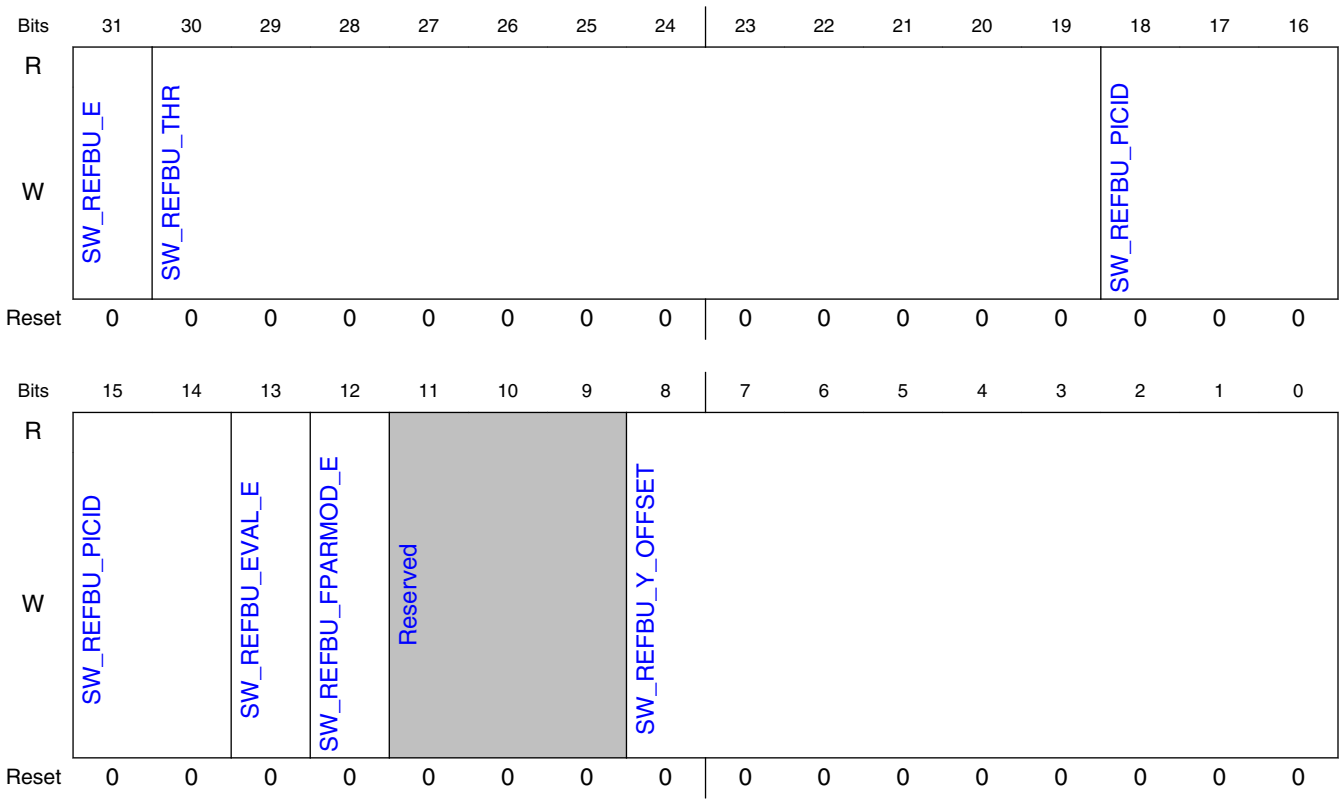
Field	Function
	11b - supported up to high profile
23 —	Reserved.
22 —	Reserved.
21 SW_DEC_OBUFF_LEVEL	Decoder output buffer level: 0b - 1 MB buffering is used 1b - 4 MB buffering is used
20 SW_REF_BUFF_EXIST	Reference picture buffer usage: 0b - not supported 1b - reference buffer is used
19-16 SW_DEC_BUS_STRD	Connected to standard bus: 0000b - error 0001b - AHB master, AHB slave 0010b - OCP master, OCP slave 0011b - AXI master, AXI slave 0100b - AXI master, APB slave 0101b - AXI master, AHB slave
15-14 SW_DEC_SYNTH_LAN	00b - error 01b - vhdl 10b - verilog
13-12 SW_DEC_BUS_WIDTH	00b - error 01b - 32 bit bus 10b - 64 bit bus 11b - 128 bit bus
11 SW_DEC_SOREN_PROF	Decoding format support, Sorenson 0b - not supported 1b - supported
10-0 SW_DEC_MAX_OWIDTH	Max configured decoder video resolution that can be decoded. Informed as width of the picture in pixels.

16.1.6.1.12 Reference picture buffer control register (SWREG51)

16.1.6.1.12.1 Offset

Register	Offset
SWREG51	CCh

16.1.6.1.12.2 Diagram



16.1.6.1.12.3 Fields

Field	Function
31 SW_REFBU_E	Refer picture buffer enable: 0b - refer picture buffer disabled 1b - refer picture buffer enabled. Valid if picture size is QVGA or more.
30-19 SW_REFBU_THR	Reference buffer disable threshold value (cache miss amount). Used to buffer shut down (if more misses than allowed)
18-14 SW_REFBU_PICID	The used reference picture ID for reference buffer usage
13 SW_REFBU_EVAL_E	Enable for HW internal reference ID calculation. If given threshold level is reached by any picture_id after first MB row, that picture_id is used for reference buffer fill for rest of the picture
12 SW_REFBU_FPARMOD_E	Field parity mode enable. Used in refbufferd evaluation mode. 0b - use the result field of the evaluation 1b - use the parity mode field
11-9 —	Reserved.

Table continues on the next page...

Field	Function
8-0 SW_REFBU_Y_OFFSET	Y offset for refbufferd. This coordinate is used to compensate the global motion of the video for better buffer hit rate

16.1.6.1.13 Reference picture buffer information register 1 (SWREG52)

16.1.6.1.13.1 Offset

Register	Offset
SWREG52	D0h

16.1.6.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_REFBU_HIT_SUM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_REFBU_INTRA_SUM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.1.13.3 Fields

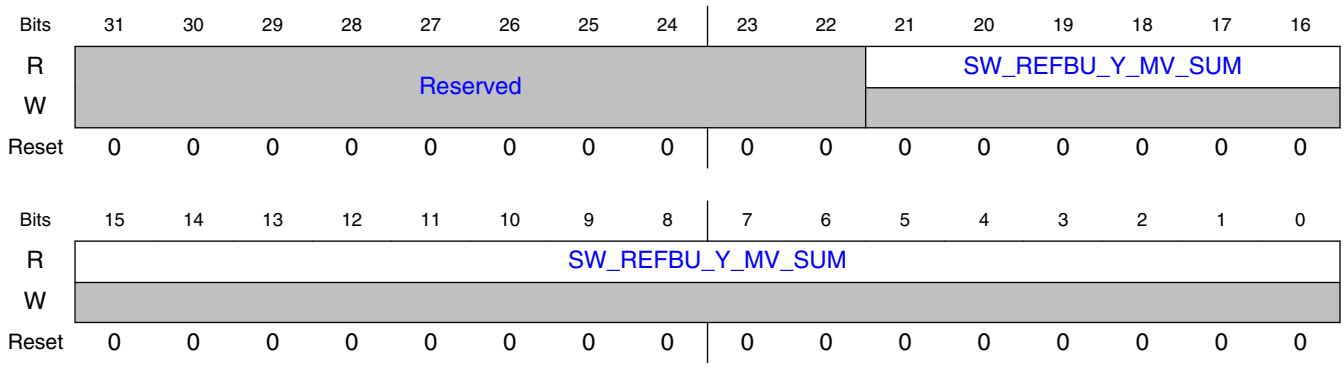
Field	Function
31-16 SW_REFBU_HIT_SUM	The sum of the refbufferd hits of the picture. Determined for each 8x8 luminance partition of the picture. The proceeding of the HW calculation can be read during HW decoding.
15-0 SW_REFBU_INTRA_SUM	The sum of the luminance 8x8 intra partitions of the picture. The proceeding of the HW calculation can be read during HW decoding.

16.1.6.1.14 Reference picture buffer information register 2 (SWREG53)

16.1.6.1.14.1 Offset

Register	Offset
SWREG53	D4h

16.1.6.1.14.2 Diagram



16.1.6.1.14.3 Fields

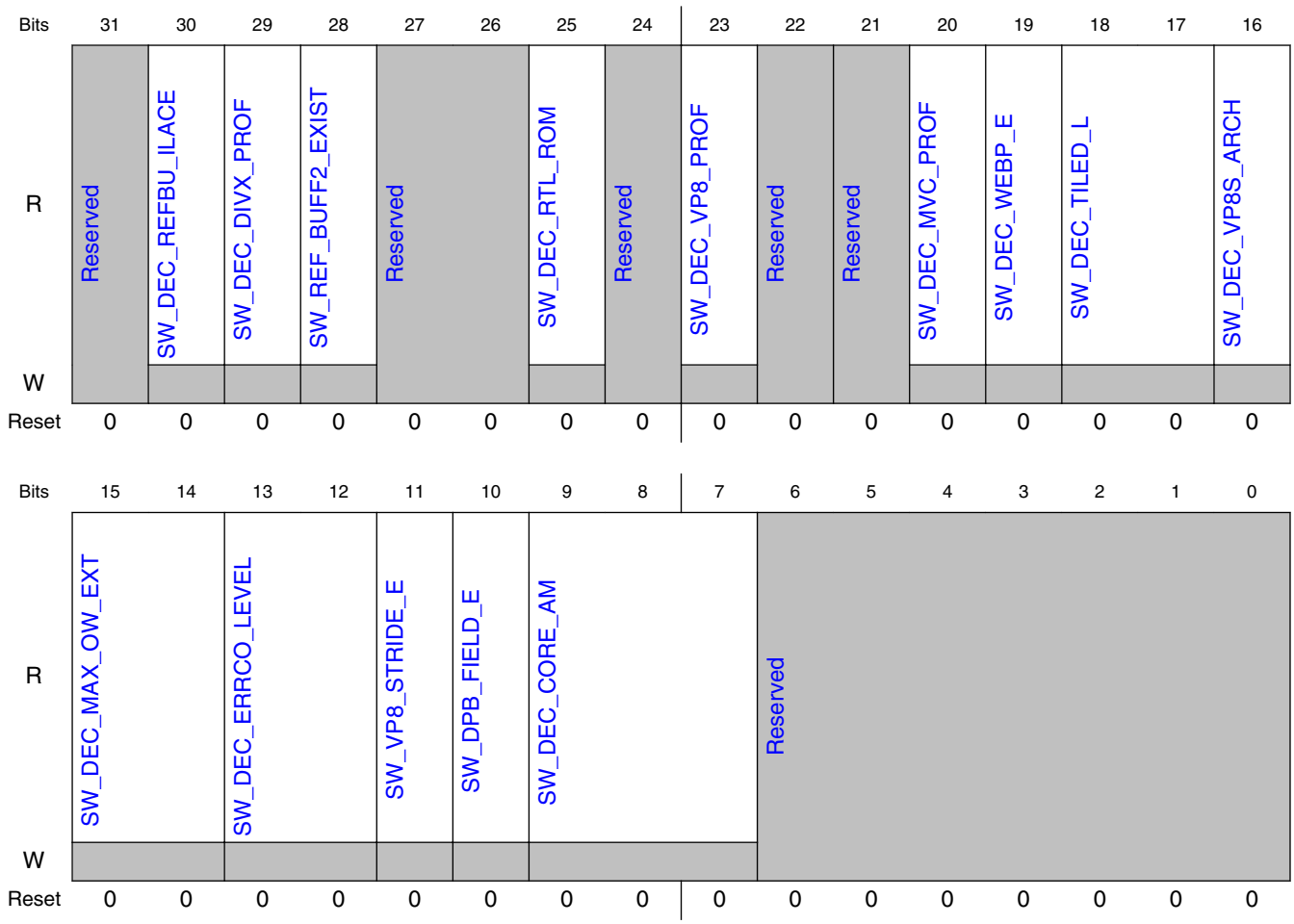
Field	Function
31-22 —	Reserved.
21-0 SW_REFBU_Y_MV_SUM	The sum of the decoded motion vector y-components of the picture. The first luminance motion vector of each MB is used in calculation. Other motion vectors of the MB are discarded. Each motion vector is saturated between -256 - 255 before calculation. The proceeding of the HW calculation can be read during HW decoding.

16.1.6.1.15 Synthesis configuration register decoder 1 (SWREG54)

16.1.6.1.15.1 Offset

Register	Offset
SWREG54	D8h

16.1.6.1.15.2 Diagram



16.1.6.1.15.3 Fields

Field	Function
31 —	Reserved.
30 SW_DEC_REFBU_ILACE	Refbufferd support for interlaced content: 0b - not supported 1b - supported
29 SW_DEC_DIVX_PROF	DIVX Support: 0b - not supported 1b - supported
28 SW_REF_BUFF2_EXIST	Reference picture buffer 2 usage: 0b - not supported 1b - reference buffer 2 is used
27-26	Reserved.

Table continues on the next page...

Field	Function
—	
25 SW_DEC_RTL_ROM	ROM implementation type (If design includes ROMs) 0b - ROMs are implemented from actual ROM units 1b - ROMs are implemented from RTL
24 —	Reserved.
23 SW_DEC_VP8_PROF	Decoding format support, VP8 0b - not supported 1b - supported
22 —	Reserved.
21 —	Reserved.
20 SW_DEC_MVC_PROF	Decoding format support, MVC 0b - not supported 1b - supported
19 SW_DEC_WEBP_E	Decoding format support, Web-p 0b - not supported bigger than 1080p resolution 1b - supported upto 16kx16k pixel resolution (defined max)
18-17 SW_DEC_TILED_L	Tiled mode support level 00b - not supported 01b - supported with 8x4 tile size for progressive content 10b - supported with 8x4 tile size for progressive/ilaced content
16 SW_DEC_VP8S_ARCH	VP8 Architecture type (for prediction) 0b - Same prediction architecture as for other decoding formats 1b - Dedicated small architecture for VP8 (refbuffer cannot be used either)
15-14 SW_DEC_MAX_OW_EXT	Max configured decoder video resolution that can be decoded. This is the MSB part of the configuration signal
13-12 SW_DEC_ERRCO_LEVEL	Decoder error concealment support level: 00b - Error concealment not supported (only error detection) 01b - VP8 direct mode motion vector error concealment supported
11 SW_VP8_STRIDE_E	Decoder output stride support for VP8. Separate base addresses for Y/C data and possibility to set scanline bigger than picture width: 0b - not supported, Y and C tables attached. 1b - supported, Y and C tables can be set freely.
10 SW_DPB_FIELD_E	DPB field separate mode support for ilaced content: 0b - Not supported. For ilaced content, DPB is ilaced frame order. 1b - Supported. For ilaced content, DPB can consist of ilaced frames or separate fields (TOP/BOT).
9-7 SW_DEC_CORE_AM	Decoder core amount. If other than 0, the multicore can be used. Each individual cores can be identified from corresponding core ID register: 000b - single core decoder 001b - dual core decoder 010b - 3 core decoder 011b - 4 core decoder 100b - 5 core decoder

Table continues on the next page...

Field	Function
	101b - 6 core decoder 110b - 7 core decoder 111b - 8 core decoder
6-0 —	Reserved.

16.1.6.1.16 Reference picture buffer 2 / Advanced prefetch control register (SWREG55)

16.1.6.1.16.1 Offset

Register	Offset
SWREG55	DCh

16.1.6.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_REFBU2_BUF_E	SW_REFBU2_THR												SW_REFBU2_PICID		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_REFBU2_PICID		SW_APF_THRESHOLD													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.1.16.3 Fields

Field	Function
31 SW_REFBU2_BUF_E	Refer picture buffer 2 enable: 0b - refer picture buffer disabled 1b - refer picture buffer enabled. Valid if picture size is QVGA or more (can be turned of by HW if threshold value reached).
30-19 SW_REFBU2_THR	Reference buffer disable threshold value (buffer miss amount). Used to buffer shut down (if more misses than allowed)
18-14 SW_REFBU2_PICID	The used reference picture ID for reference buffer usage
13-0 SW_APF_THRESHOLD	G1 decoder and later :Advanced prefetch threshold value. If current MB exceeds the threshold the advanced mode is not used. Value 0 disables threshold usage and advanced prefetch usage is restricted by internal memory limitation only

16.1.6.1.17 Reference buffer information register 3 (SWREG56)

16.1.6.1.17.1 Offset

Register	Offset
SWREG56	E0h

16.1.6.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_REFBU_TOP_SUM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_REFBU_BOT_SUM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.1.17.3 Fields

Field	Function
31-16	The sum of the top partitions of the picture

Table continues on the next page...

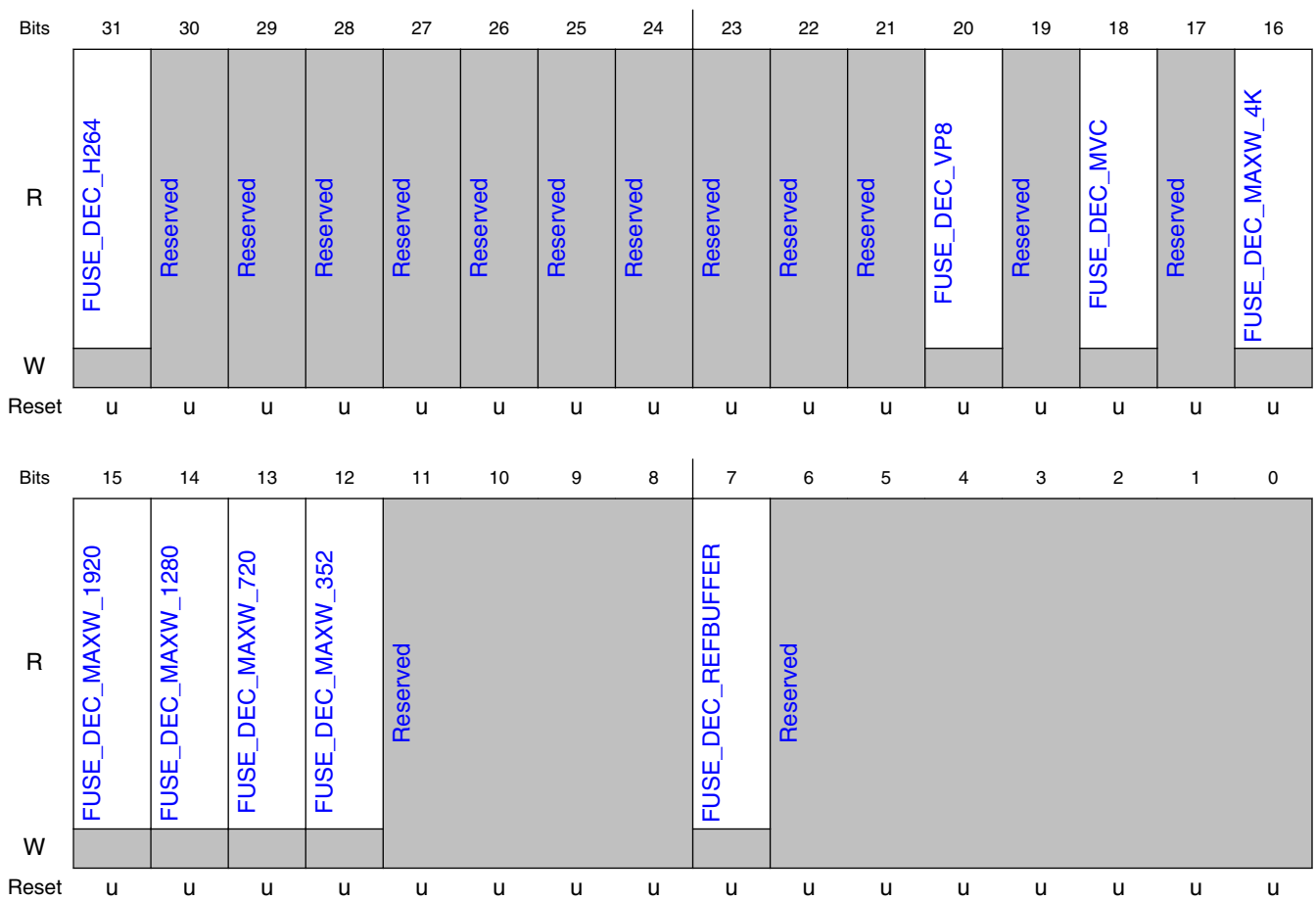
Field	Function
SW_REFBU_TOP_SUM	
15-0 SW_REFBU_BOT_SUM	The sum of the bottom partitions of the picture

16.1.6.1.18 Decoder fuse register (SWREG57)

16.1.6.1.18.1 Offset

Register	Offset
SWREG57	E4h

16.1.6.1.18.2 Diagram



16.1.6.1.18.3 Fields

Field	Function
31 FUSE_DEC_H264	1 = H.264 enabled
30 —	Reserved.
29 —	Reserved.
28 —	Reserved.
27 —	Reserved.
26 —	Reserved.
25 —	Reserved.
24 —	Reserved.
23 —	Reserved.
22 —	Reserved.
21 —	Reserved.
20 FUSE_DEC_VP8	1 = VP8 enabled
19 —	Reserved.
18 FUSE_DEC_MVC	1 = MVC enabled (requires also H264 to be enabled)
17 —	Reserved.
16 FUSE_DEC_MAXW_4K	1 = Max video width up to 4096 pixels enabled. Priority coded with priority 1.
15	1 = Max video width up to 1920 pixels enabled. Priority coded with priority 2.

Table continues on the next page...

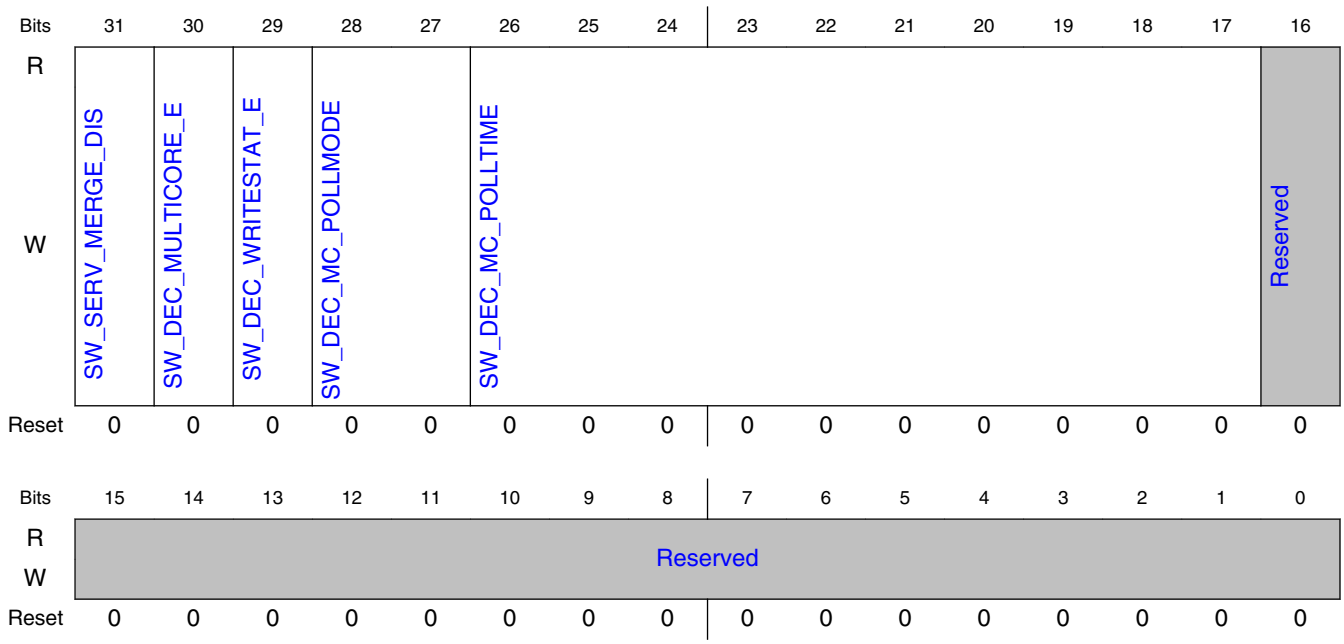
Field	Function
FUSE_DEC_MAXW_1920	
14 FUSE_DEC_MAXW_1280	1 = Max video width up to 1280 pixels enabled. Priority coded with priority 3.
13 FUSE_DEC_MAXW_720	1 = Max video width up to 720 pixels enabled. Priority coded with priority 4.
12 FUSE_DEC_MAXW_352	1 = Max video width up to 352 pixels enabled. Priority coded with priority 5.
11-8 —	Reserved.
7 FUSE_DEC_REFBUFFER	1 = reference buffer used
6-0 —	Reserved.

16.1.6.1.19 Device configuration register decoder 2 + Multi core control register (SWREG58)

16.1.6.1.19.1 Offset

Register	Offset
SWREG58	E8h

16.1.6.1.19.2 Diagram



16.1.6.1.19.3 Fields

Field	Function
31 SW_SERV_MERGE_DIS	Decoder service merge disable: 0b - HW merges simultaneous sub-block requests internally if they are same type (read or write). 1b - decoder serves one sub-block per service and merging is disabled.
30 SW_DEC_MULTICORE_E	Decoder multi core enable: 0b - Multi core disabled or only one core exists in design. 1b - Multi core enable. Each reference picture status must be verified from external memory status field before usage. 128 bits status word exists after each reference picture and include picture proceeding coordinates Y and X.
29 SW_DEC_WRITESTAT_E	Decoder write statusword enable. Must be high if multi core decoding enabled. HW writes output picture data proceeding to external memory after picture data (and after H264 direct mode MVS if they exist)
28-27 SW_DEC_MC_POLLMODE	Decoder multicore status reading mode: 00b - HW internal status polling mechanism is used. Status of reference picture is read only when required coordinate for the reference picture is not big enough. If the status is still not big enough after reading it the HW waits N clock cycles per pixel from the coordinate difference. The N is defined by the sw_dec_mc_polltime (range 0...4). 01b - Dummy status polling mechanism is used for all reference pictures. HW reads status of all reference pictures at frequency defined by sw_dec_mc_polltime.
26-17 SW_DEC_MC_POLLTIME	sw_dec_mc_polltime definition depends on sw_dec_mc_mode. if mode is 0 (HW internal mechanism) the sw_dec_mc_polltime: • 0 = 1/4 cycles per pixel • 1 = wait 1/2 cycles per pixel • 2 = wait one cycle per pixel

Table continues on the next page...

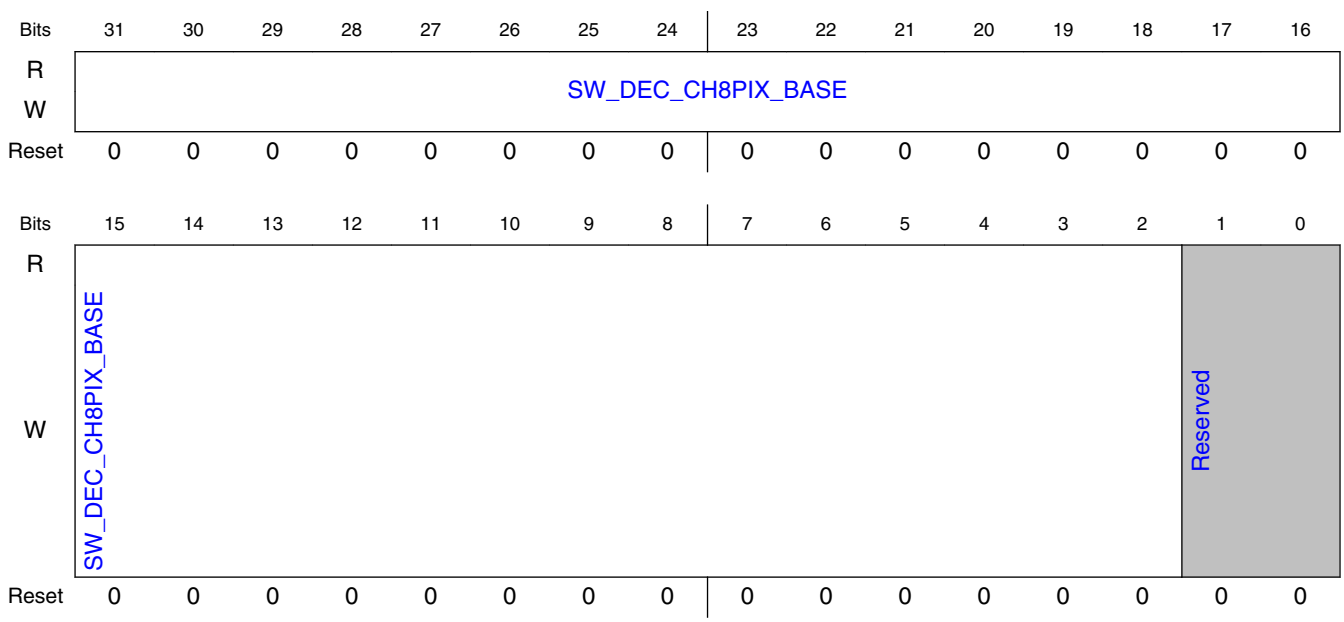
Field	Function
	• 3 = wait 2 cycles per pixel • 4=wait 4 cycles per pixel For dummy polling mechanism the sw_dec_mc_polltime defines the amount cycles between status reads x1024: • 0 = wait 1024 cycles between status reading • 1 = wait 2048 cycles between status reading • 2 = wait 3072 cycles between status reading • N = wait (N+1)x1024 cycles
16-0 —	Reserved.

16.1.6.1.20 H264 Chrominance 8 pixel interleaved data base (SWREG59)

16.1.6.1.20.1 Offset

Register	Offset
SWREG59	ECh

16.1.6.1.20.2 Diagram



16.1.6.1.20.3 Fields

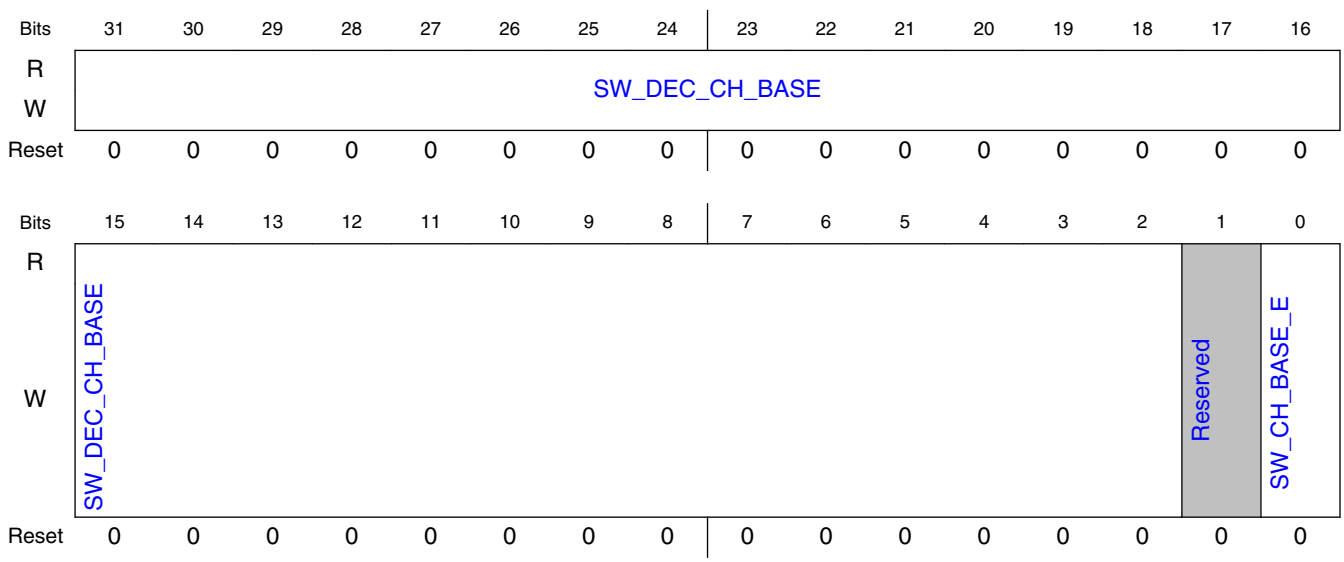
Field	Function
31-2 SW_DEC_CH8PIX_BASE	Base address for additional chrominance data format where chrominance is interleaved in group of 8 pixels. The usage is enabled by sw_ch_8pix_ileav_e
1-0 —	Reserved.

16.1.6.1.21 Base address for H264 decoded chroma picture (SWREG102)

16.1.6.1.21.1 Offset

Register	Offset
SWREG102	198h

16.1.6.1.21.2 Diagram



16.1.6.1.21.3 Fields

Field	Function
31-2	Valid only if chroma address separate mode is enabled. H264: Base address for decoder output chroma picture. Points directly to start of decoder output chroma picture or field.

Table continues on the next page...

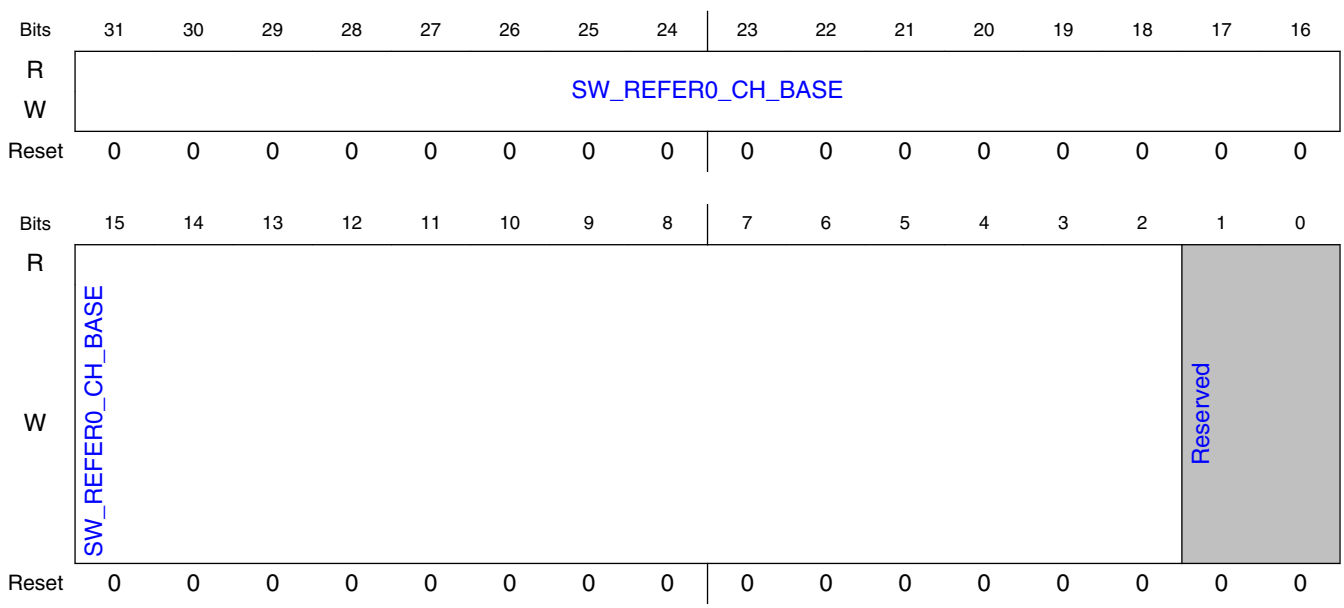
Field	Function
SW_DEC_CH_BASE	
1 —	Reserved.
0 SW_CH_BASE_E	chroma address separate mode enable: 0b - HW outputs decoded chroma picture to the end of decoded luma picture. HW calculates the chroma picture address according to sw_dec_base and luma data length. 1b - HW outputs decoded chroma picture to independent memory address

16.1.6.1.22 Base address for reference chroma picture index 0 (SWREG103)

16.1.6.1.22.1 Offset

Register	Offset
SWREG103	19Ch

16.1.6.1.22.2 Diagram



16.1.6.1.22.3 Fields

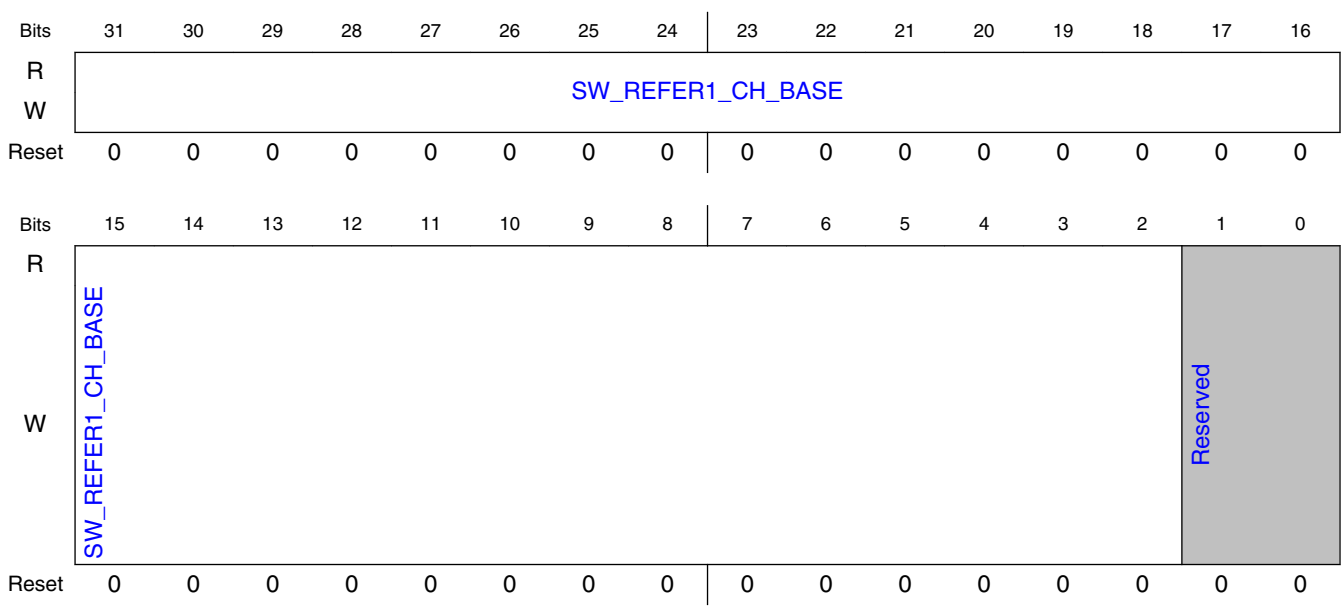
Field	Function
31-2 SW_REFER0_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 0.
1-0 —	Reserved.

16.1.6.1.23 Base address for reference chroma picture index 1 (SWREG104)

16.1.6.1.23.1 Offset

Register	Offset
SWREG104	1A0h

16.1.6.1.23.2 Diagram



16.1.6.1.23.3 Fields

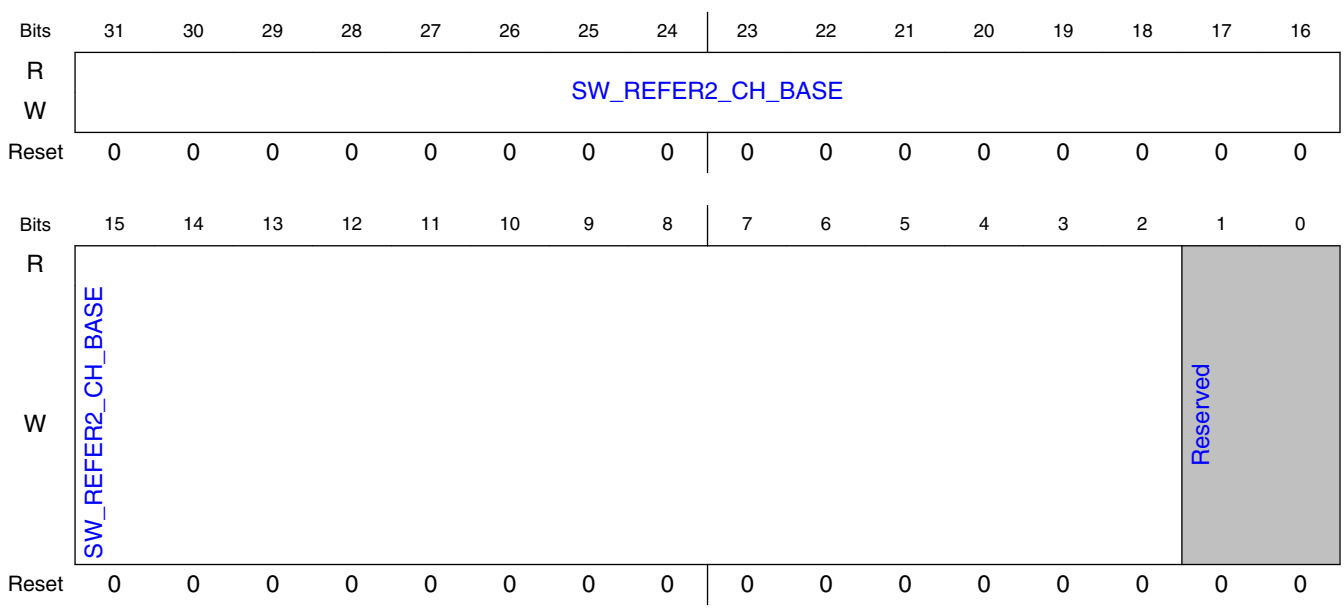
Field	Function
31-2 SW_REFER1_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 1.
1-0 —	Reserved.

16.1.6.1.24 Base address for reference chroma picture index 2 (SWREG105)

16.1.6.1.24.1 Offset

Register	Offset
SWREG105	1A4h

16.1.6.1.24.2 Diagram



16.1.6.1.24.3 Fields

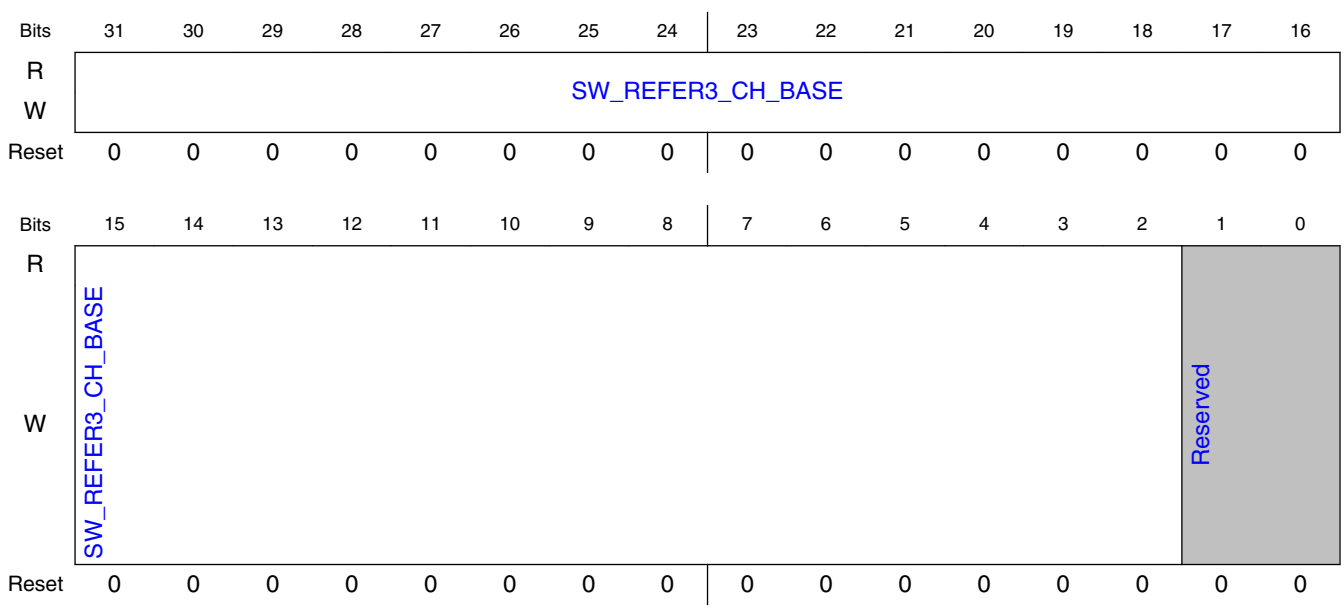
Field	Function
31-2 SW_REFER2_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 2.
1-0 —	Reserved.

16.1.6.1.25 Base address for reference chroma picture index 3 (SWREG106)

16.1.6.1.25.1 Offset

Register	Offset
SWREG106	1A8h

16.1.6.1.25.2 Diagram



16.1.6.1.25.3 Fields

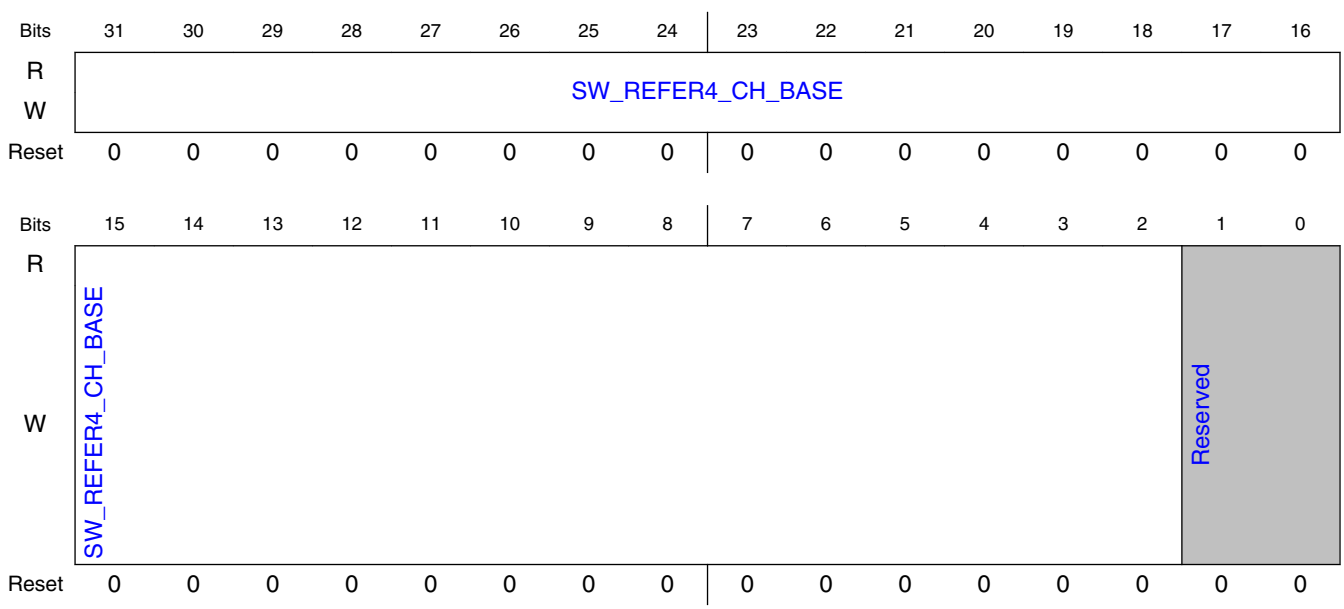
Field	Function
31-2 SW_REFER3_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 3.
1-0 —	Reserved.

16.1.6.1.26 Base address for reference chroma picture index 4 (SWREG107)

16.1.6.1.26.1 Offset

Register	Offset
SWREG107	1ACh

16.1.6.1.26.2 Diagram



16.1.6.1.26.3 Fields

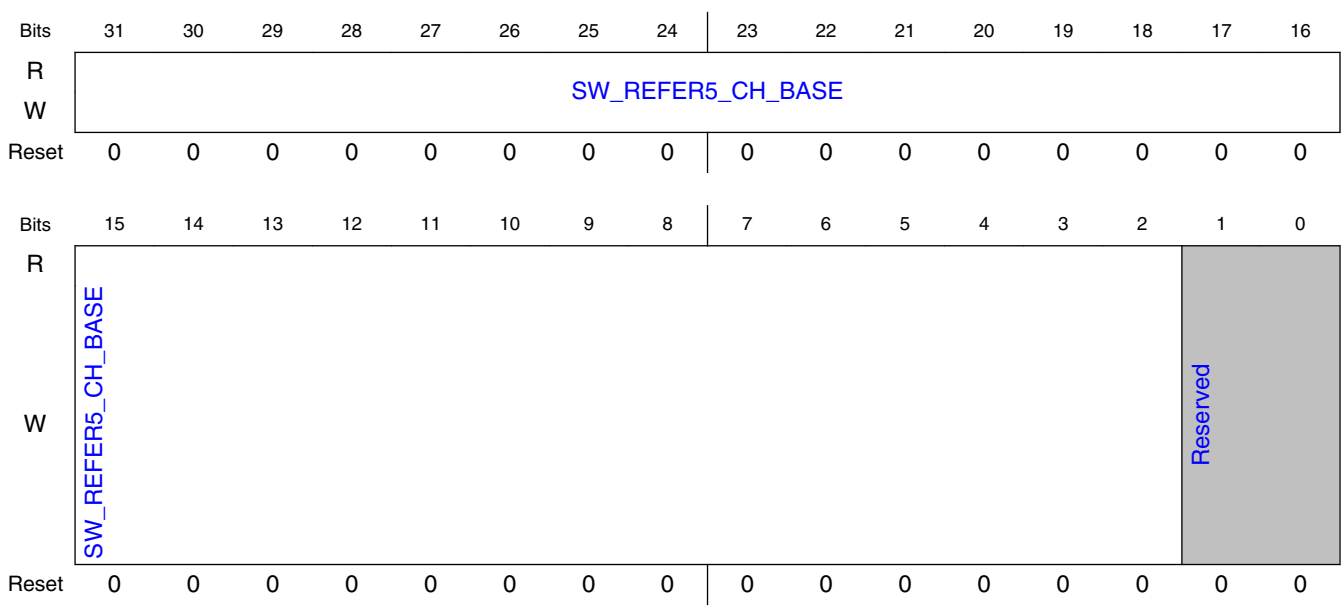
Field	Function
31-2 SW_REFER4_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 4.
1-0 —	Reserved.

16.1.6.1.27 Base address for reference chroma picture index 5 (SWREG108)

16.1.6.1.27.1 Offset

Register	Offset
SWREG108	1B0h

16.1.6.1.27.2 Diagram



16.1.6.1.27.3 Fields

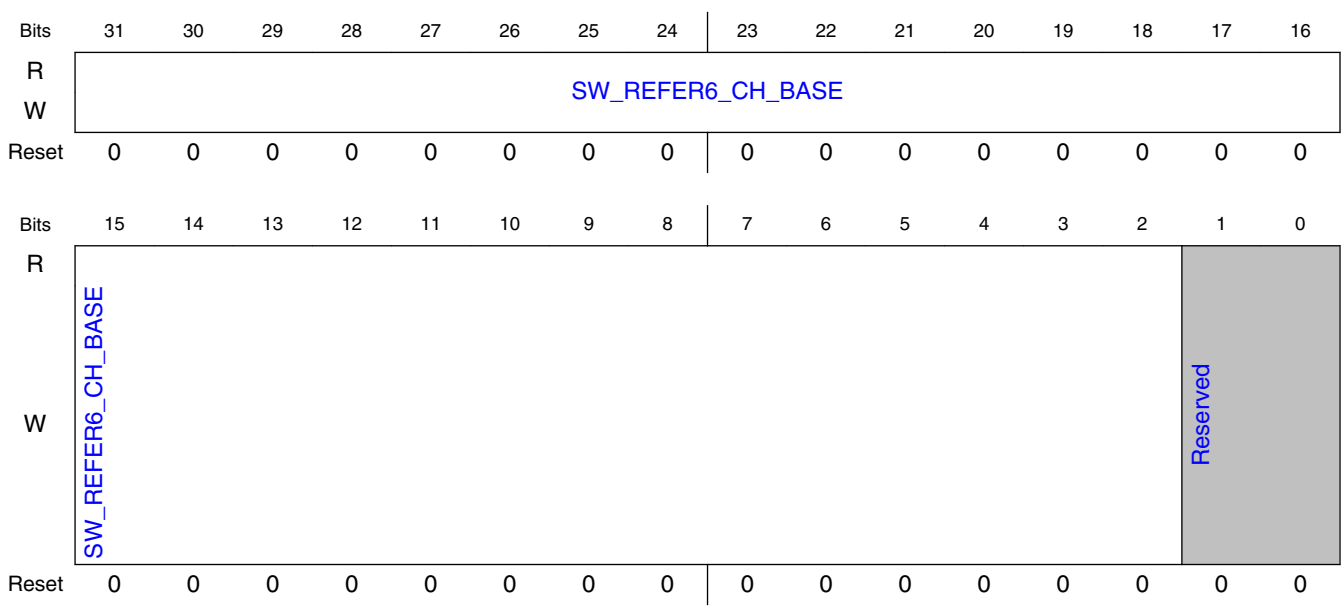
Field	Function
31-2 SW_REFER5_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 5.
1-0 —	Reserved.

16.1.6.1.28 Base address for reference chroma picture index 6 (SWREG109)

16.1.6.1.28.1 Offset

Register	Offset
SWREG109	1B4h

16.1.6.1.28.2 Diagram



16.1.6.1.28.3 Fields

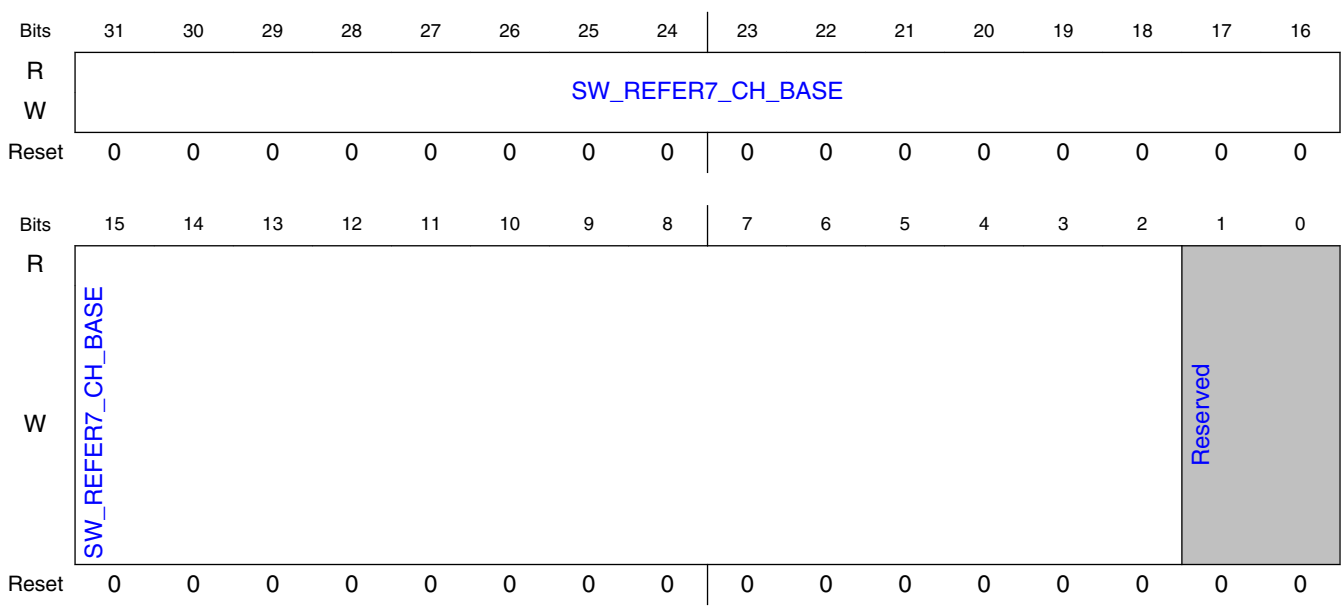
Field	Function
31-2 SW_REFER6_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 6.
1-0 —	Reserved.

16.1.6.1.29 Base address for reference chroma picture index 7 (SWREG110)

16.1.6.1.29.1 Offset

Register	Offset
SWREG110	1B8h

16.1.6.1.29.2 Diagram



16.1.6.1.29.3 Fields

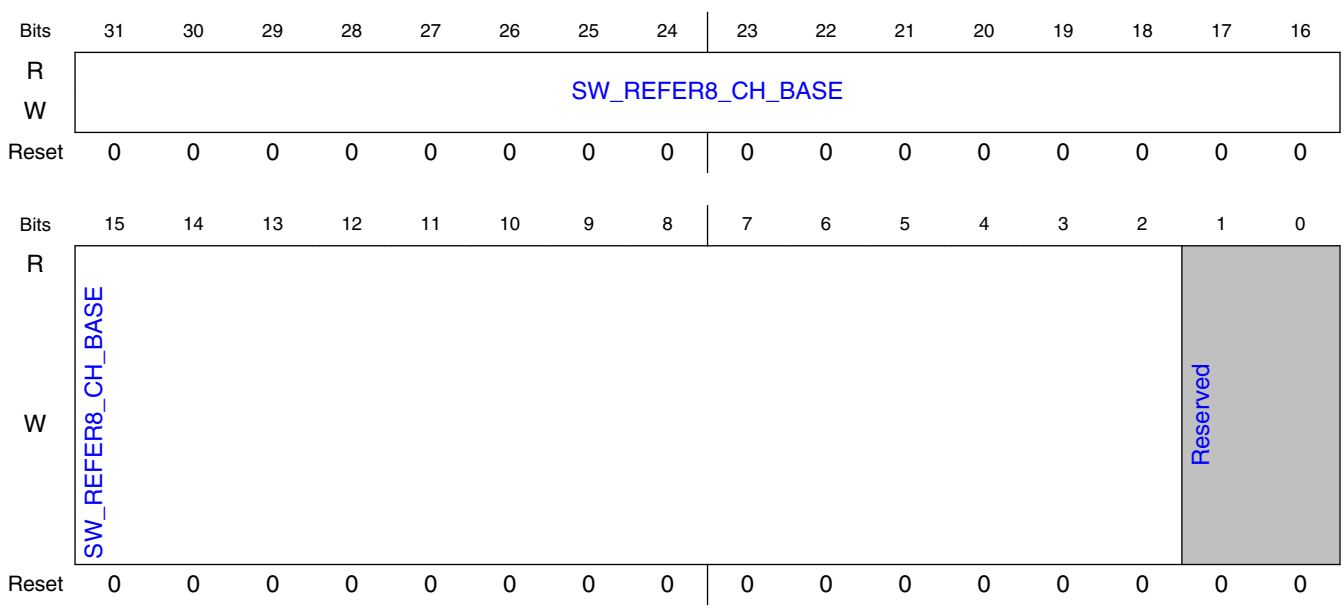
Field	Function
31-2 SW_REFER7_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 7.
1-0 —	Reserved.

16.1.6.1.30 Base address for reference chroma picture index 8 (SWREG111)

16.1.6.1.30.1 Offset

Register	Offset
SWREG111	1BCh

16.1.6.1.30.2 Diagram



16.1.6.1.30.3 Fields

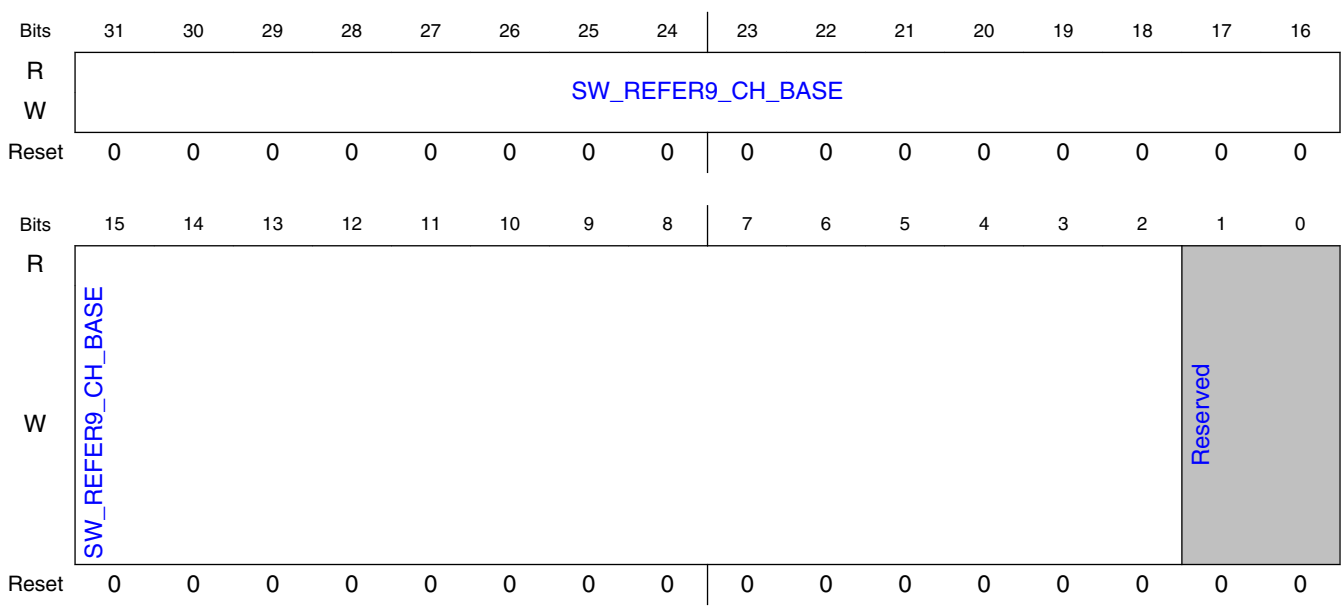
Field	Function
31-2 SW_REFER8_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 8.
1-0 —	Reserved.

16.1.6.1.31 Base address for reference chroma picture index 9 (SWREG112)

16.1.6.1.31.1 Offset

Register	Offset
SWREG112	1C0h

16.1.6.1.31.2 Diagram



16.1.6.1.31.3 Fields

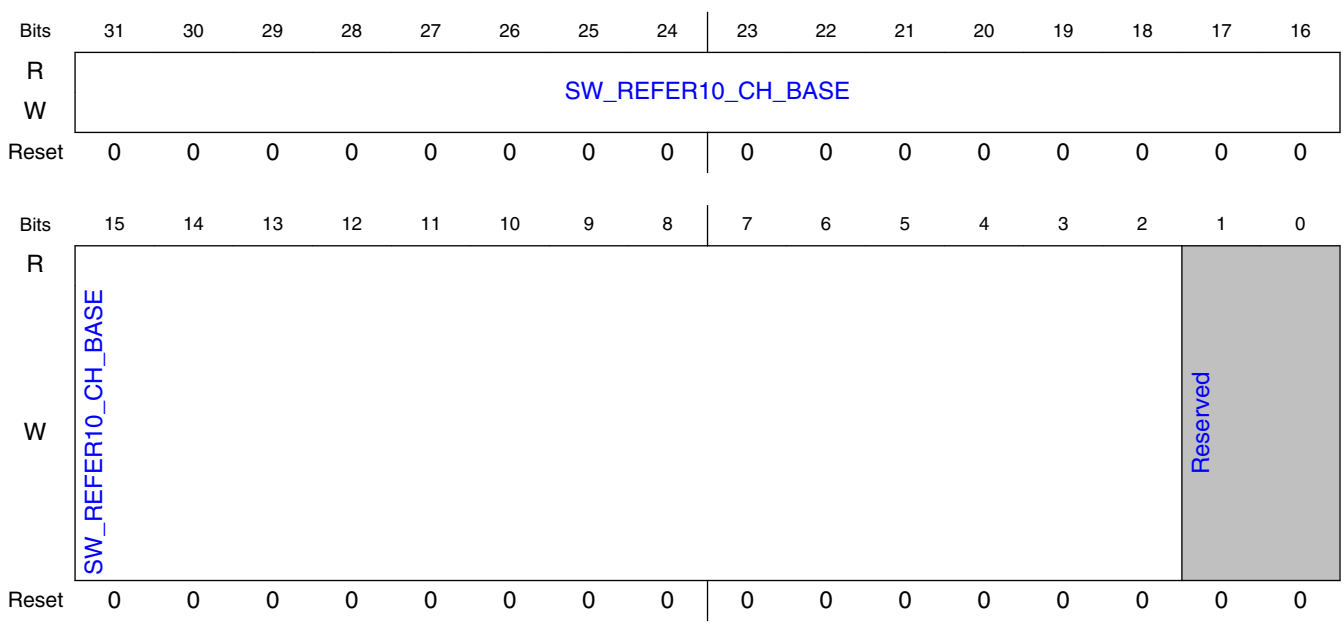
Field	Function
31-2 SW_REFER9_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 9.
1-0 —	Reserved.

16.1.6.1.32 Base address for reference chroma picture index 10 (SWREG113)

16.1.6.1.32.1 Offset

Register	Offset
SWREG113	1C4h

16.1.6.1.32.2 Diagram



16.1.6.1.32.3 Fields

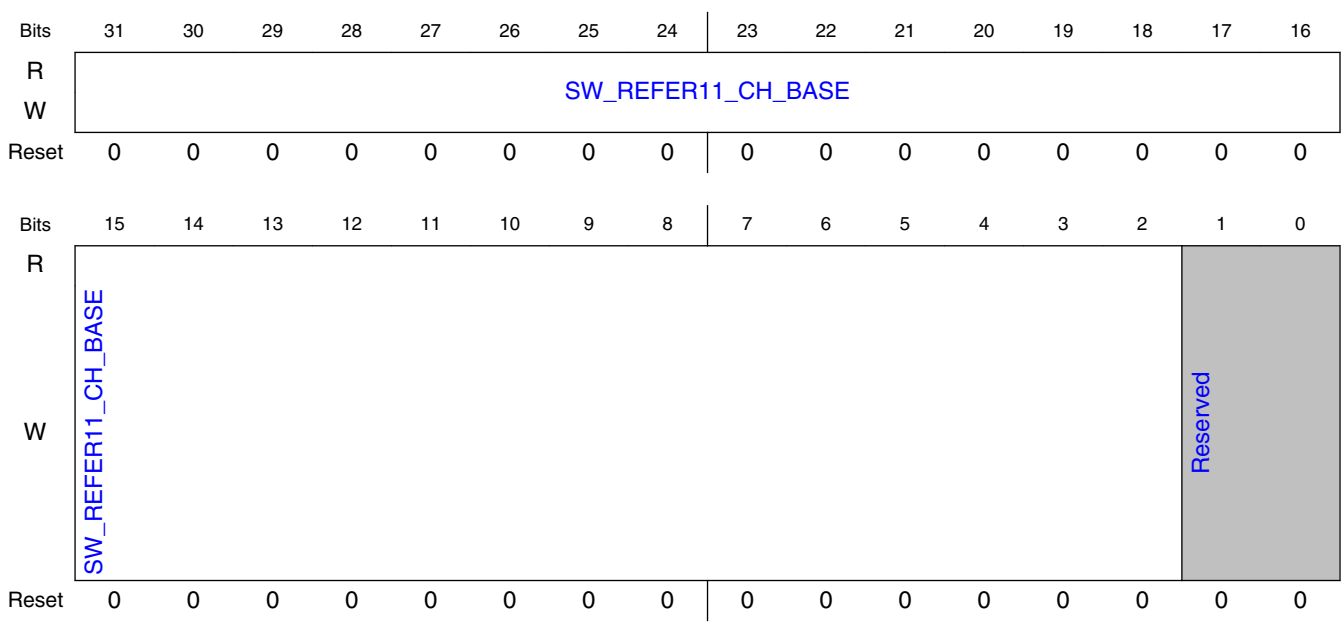
Field	Function
31-2 SW_REFER10_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 10.
1-0 —	Reserved.

16.1.6.1.33 Base address for reference chroma picture index 11 (SWREG114)

16.1.6.1.33.1 Offset

Register	Offset
SWREG114	1C8h

16.1.6.1.33.2 Diagram



16.1.6.1.33.3 Fields

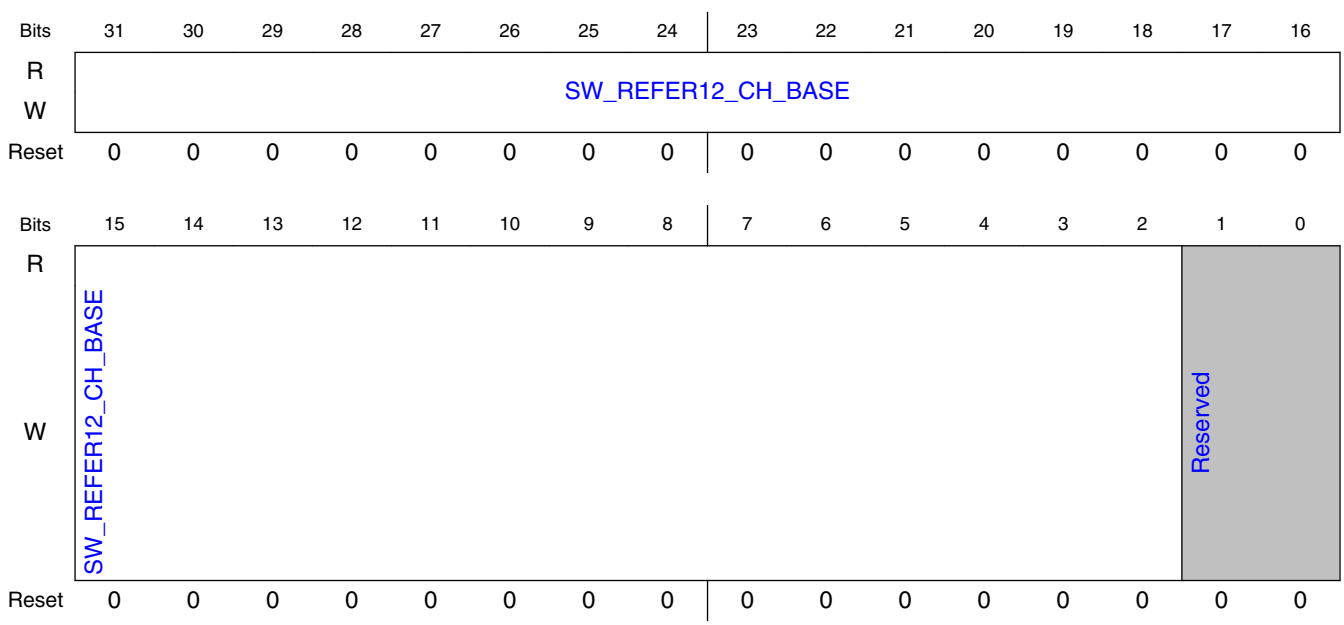
Field	Function
31-2 SW_REFER11_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 11.
1-0 —	Reserved.

16.1.6.1.34 Base address for reference chroma picture index 12 (SWREG115)

16.1.6.1.34.1 Offset

Register	Offset
SWREG115	1CCh

16.1.6.1.34.2 Diagram



16.1.6.1.34.3 Fields

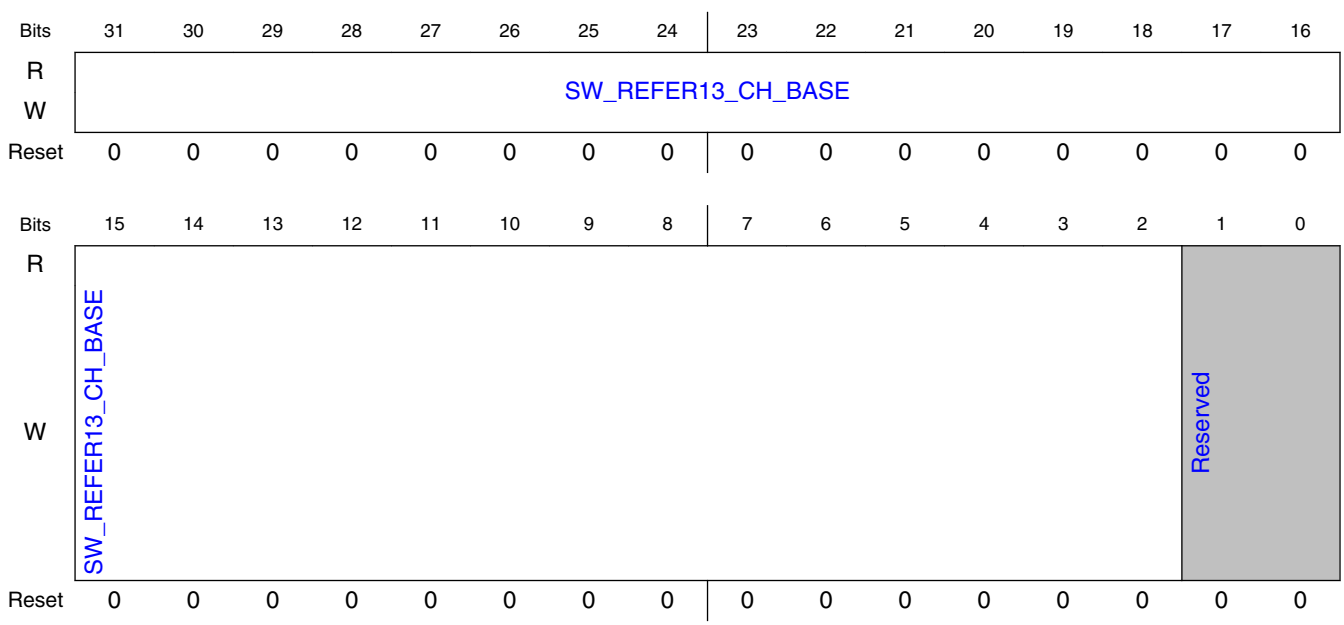
Field	Function
31-2 SW_REFER12_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 12.
1-0 —	Reserved.

16.1.6.1.35 Base address for reference chroma picture index 13 (SWREG116)

16.1.6.1.35.1 Offset

Register	Offset
SWREG116	1D0h

16.1.6.1.35.2 Diagram



16.1.6.1.35.3 Fields

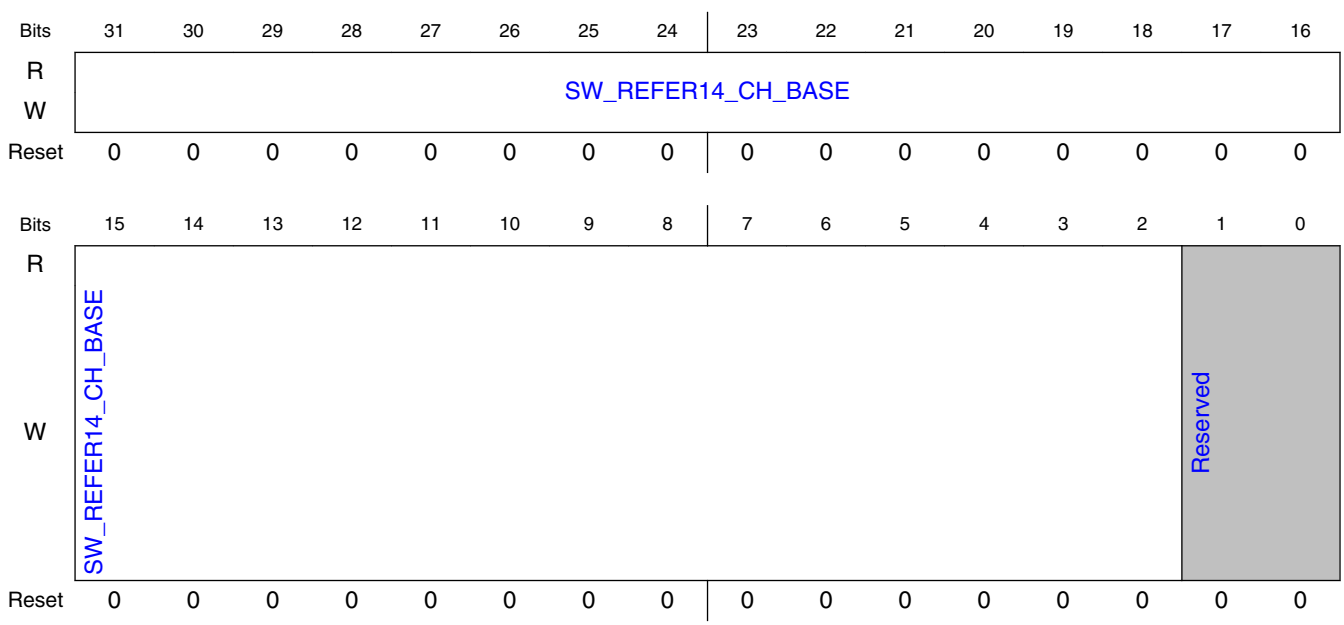
Field	Function
31-2 SW_REFER13_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 13.
1-0 —	Reserved.

16.1.6.1.36 Base address for reference chroma picture index 14 (SWREG117)

16.1.6.1.36.1 Offset

Register	Offset
SWREG117	1D4h

16.1.6.1.36.2 Diagram



16.1.6.1.36.3 Fields

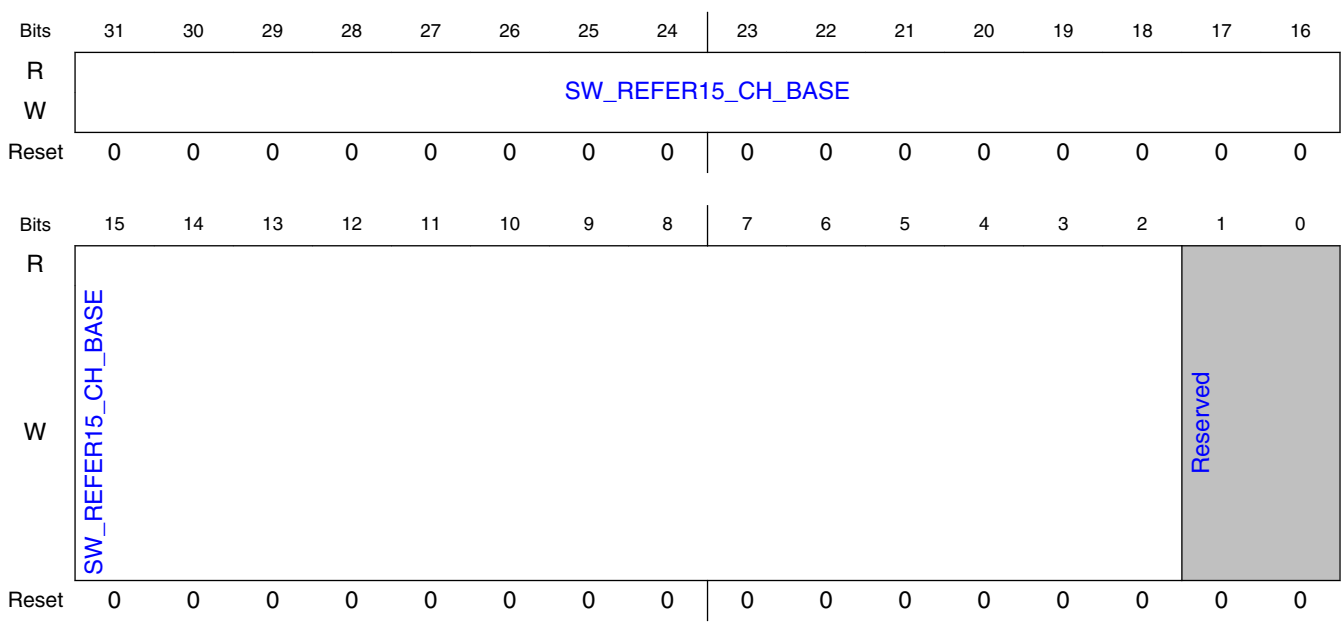
Field	Function
31-2 SW_REFER14_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 14.
1-0 —	Reserved.

16.1.6.1.37 Base address for reference chroma picture index 15 (SWREG118)

16.1.6.1.37.1 Offset

Register	Offset
SWREG118	1D8h

16.1.6.1.37.2 Diagram



16.1.6.1.37.3 Fields

Field	Function
31-2 SW_REFER15_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 15.
1-0 —	Reserved.

16.1.6.2 VPU_G1 H264 decoder register descriptions

16.1.6.2.1 VPU_G1 H264 decoder memory map

VPU_G1_H264 base address: 3830_0000h

Offset	Register	Width (In bits)	Access	Reset value
10h	Decoder control register 1 (picture parameters) (SWREG4)	32	RW	0000_0000h
14h	Decoder control register 2 (stream decoding table selects) (SWREG5)	32	RW	0000_0000h
18h	Decoder control register 3 (stream buffer information) (SWREG6)	32	RW	0000_0000h
1Ch	Decoder control register 4 (H264, VC-1, VP6 and progressive JPEG control) (SWREG7)	32	RW	0000_0000h
20h	Decoder control register 5 (H264, VC-1, VP6, Progressive JPEG and RV control) (SWREG8)	32	RW	0000_0000h
24h	Decoder control register 6 / base address for MB-control (RLC) / VC-1 intensity control 0/ VP6,VP7,VP8 ctrl-stream length/ RV pic slice amount (SWREG9)	32	RW	0000_0000h
28h	Base address for differential motion vector base address (RLC-mode) /H264 P initial fwd ref pic list register (4-9)/ VC-1 intensity control 1/ VP7 and VP8 segmentation base register (SWREG10)	32	RW	0000_0000h
2Ch	Decoder control register 7 (VLC) / base address for H.264 intra prediction 4x4 / base address for MPEG-4 DC component (RLC) / H264 P initial fwd ref pic list register (10-15) / VC-1 intensity control 2 (SWREG11)	32	RW	0000_0000h
38h	Base address for reference picture index 0 / base address for JPEG decoder output chrominance picture (SWREG14)	32	RW	0000_0000h
3Ch	Base address for reference picture index 1 / JPEG control (SWREG15)	32	RW	0000_0000h
40h	Base address for reference picture index 2 / List of VLC code lengths in first JPEG AC table (SWREG16)	32	RW	0000_0000h
44h	Base address for reference picture index 3 / List of VLC code lengths in first JPEG AC table (SWREG17)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
48h	Base address for reference picture index 4 / VC1 control / MPEG4 MVD control/ List of VLC code lengths in first JPEG AC table / VC-1 intensity control 4 / VP6/VP7, VP8 Golden refer picture base (SWREG18)	32	RW	0000_0000h
4Ch	Base address for reference picture index 5 / MPEG4 TRB/TRD delta 0 / VC-1 intensity control 3 List of VLC code lengths in first/second JPEG AC table / VP6/VP7 scan maps (SWREG19)	32	RW	0000_0000h
50h	Base address for reference picture index 6 / / MPEG4 TRB/TRD delta -1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG20)	32	RW	0000_0000h
54h	Base address for reference picture index 7 / MPEG4 TRB/TRD delta 1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG21)	32	RW	0000_0000h
58h	Base address for reference picture index 8 / List of VLC code lengths in second JPEG AC table / VP6 scan maps / VP7,VP8 DCT stream 1 base (SWREG22)	32	RW	0000_0000h
5Ch	Base address for reference picture index 9 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 2 base (SWREG23)	32	RW	0000_0000h
60h	Base address for reference picture index 10 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 3 base (SWREG24)	32	RW	0000_0000h
64h	Base address for reference picture index 11 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 4 base (SWREG25)	32	RW	0000_0000h
68h	Base address for reference picture index 12 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 5 base (SWREG26)	32	RW	0000_0000h
6Ch	Base address for reference picture index 13 / VC-1 bitpl mbctrl or VP6,VP7,VP8 ctrl stream base /Progressive JPEG DC table (SWREG27)	32	RW	0000_0000h
70h	Base address for reference picture index 14 / VP6 scan maps / Progressive JPEG DC table / VP7,VP8 DCT stream 6 base (SWREG28)	32	RW	0000_0000h
74h	Base address for reference picture index 15 / VP6 scan maps / VP7,VP8 DCT stream 7 base (SWREG29)	32	RW	0000_0000h
78h	Reference picture numbers for index 0 and 1 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter mb level adjusts (SWREG30)	32	RW	0000_0000h
7Ch	Reference picture numbers for index 2 and 3 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter ref pic level adjusts (SWREG31)	32	RW	0000_0000h
80h	Reference picture numbers for index 4 and 5 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter levels (SWREG32)	32	RW	0000_0000h
84h	Reference picture numbers for index 6 and 7 (H264 VLC) / VP6 scan maps / VP7,VP8 quantization values (SWREG33)	32	RW	0000_0000h
88h	Reference picture numbers for index 8 and 9 (H264 VLC) / MPEG4, VC1, VPx prediction filter taps (SWREG34)	32	RW	0000_0000h
8Ch	Reference picture numbers for index 10 and 11 (H264 VLC) / VC1, VPx prediction filter taps (SWREG35)	32	RW	0000_0000h

Table continues on the next page...

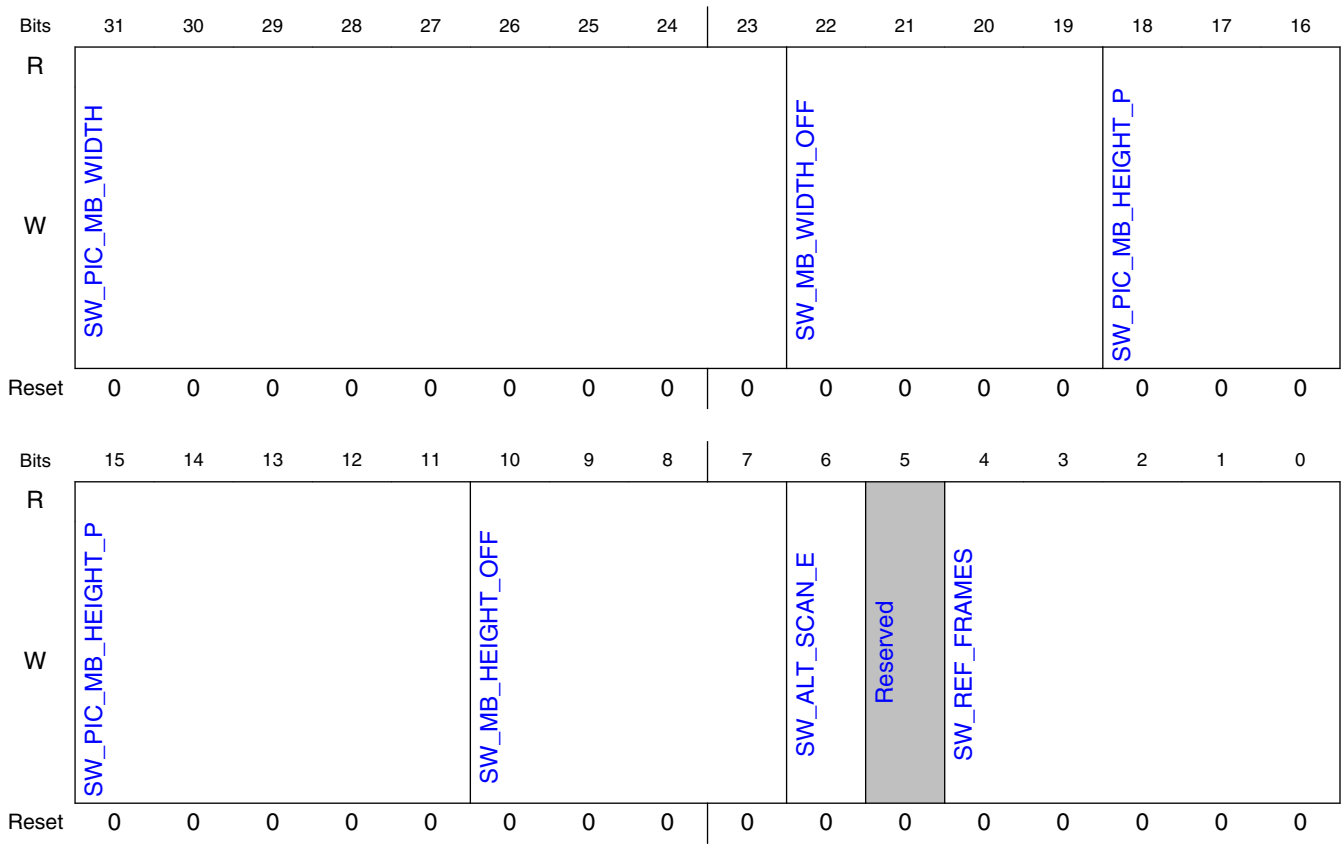
Offset	Register	Width (In bits)	Access	Reset value
90h	Reference picture numbers for index 12 and 13 (H264 VLC) / VC1, VPx prediction filter taps (SWREG36)	32	RW	0000_0000h
94h	Reference picture numbers for index 14 and 15 (H264 VLC) / VPx prediction filter taps (SWREG37)	32	RW	0000_0000h
98h	Reference picture long term flags (H264 VLC) / VPx prediction filter taps (SWREG38)	32	RW	0000_0000h
9Ch	Reference picture valid flags (H264 VLC) / VPx prediction filter taps (SWREG39)	32	RW	0000_0000h
A8h	bi_dir initial ref pic list register (0-2) / VP6 prediction filter taps / Progressive JPEG Cb ACDC coefficient base (SWREG42_H264)	32	RW	0000_0000h
ACh	bi-dir initial ref pic list register (3-5) / VP6 prediction filter taps / Progressive JPEG Cr ACDC coefficient base (SWREG43_H264)	32	RW	0000_0000h
B0h	bi-dir initial ref pic list register (6-8) / VP6 prediction filter taps (SWREG44_H264)	32	RW	0000_0000h
B4h	bi-dir initial ref pic list register (9-11) / VP6 prediction filter taps (SWREG45)	32	RW	0000_0000h
B8h	bi-dir initial ref pic list register (12-14) / VP7,VP8 quantization values (SWREG46)	32	RW	0000_0000h
BCh	bi-dir and P fwd initial ref pic list register (15 and P 0-3) / VP7,VP8 quantization values (SWREG47)	32	RW	0000_0000h

16.1.6.2.2 Decoder control register 1 (picture parameters) (SWREG4)

16.1.6.2.2.1 Offset

Register	Offset
SWREG4	10h

16.1.6.2.2.2 Diagram



16.1.6.2.2.3 Fields

Field	Function
31-23 SW_PIC_MB_WIDTH	Picture width in macroblocks = ((width in pixels + 15) /16)
22-19 SW_MB_WIDTH_OFF	The amount of meaningfull horizontal pixels in last MB (width offset) 0 if exactly 16 pixels multiple picture and all the horizontal pixels in last MB are meaningfull
18-11 SW_PIC_MB_HEIGHT_P	Picture height in macroblocks =((height in pixels+15)/16). Picture height is informed as size of the (progressive) frame also for single field (of interlaced content) is being decoded
10-7 SW_MB_HEIGHT_OFF	The amount of meaningful vertical pixels in last MB (height offset 0 if exactly 16 pixels multiple picture and all the vertical pixels in last MB are meaningfull
6 SW_ALT_SCAN_E	indicates alternative vertical scan method used for interlaced frames
5	Reserved.

Table continues on the next page...

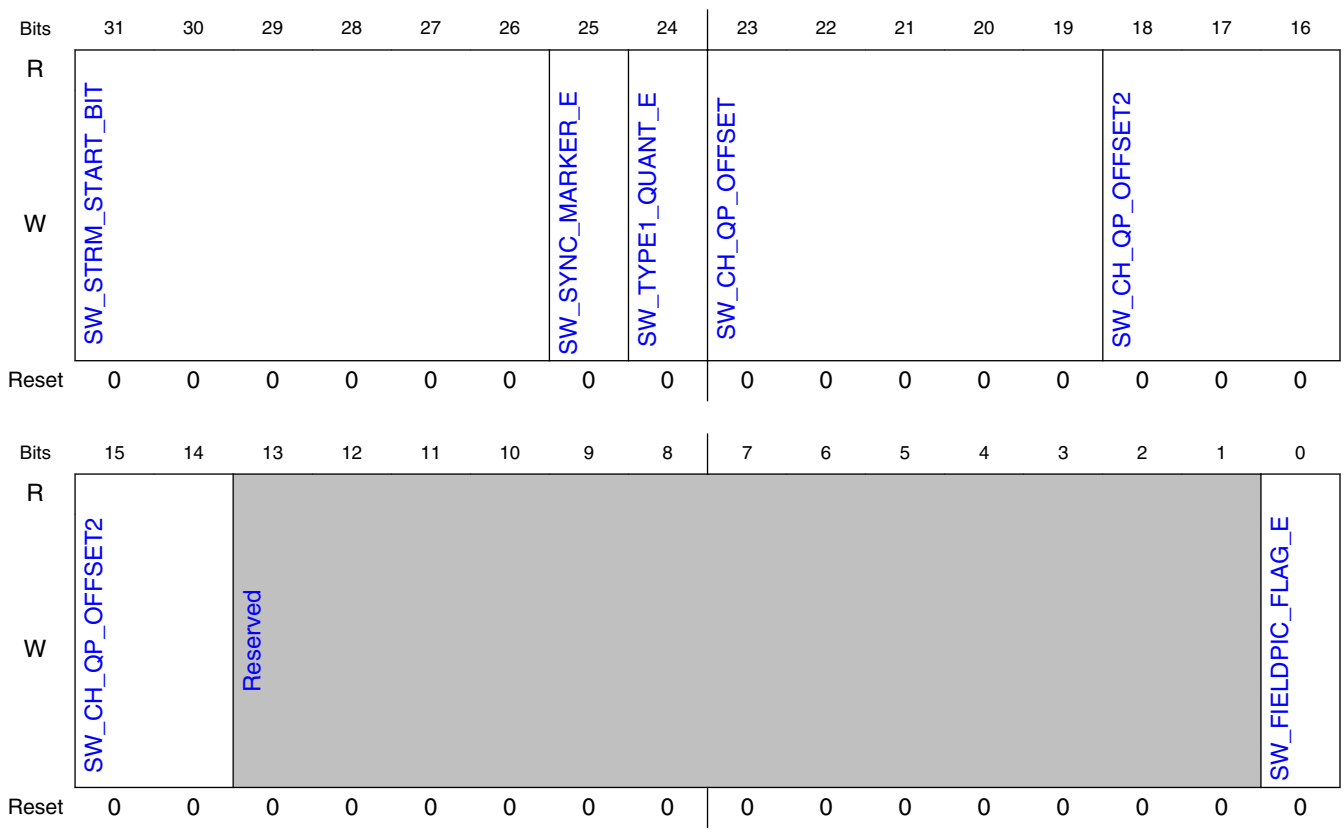
Field	Function
—	
4-0 SW_REF_FRAMES	H.264: num_ref_frames, maximum number of short and long term reference frames in decoded picture buffer VC-1: num_ref semantics

16.1.6.2.3 Decoder control register 2 (stream decoding table selects) (SWREG5)

16.1.6.2.3.1 Offset

Register	Offset
SWREG5	14h

16.1.6.2.3.2 Diagram



16.1.6.2.3.3 Fields

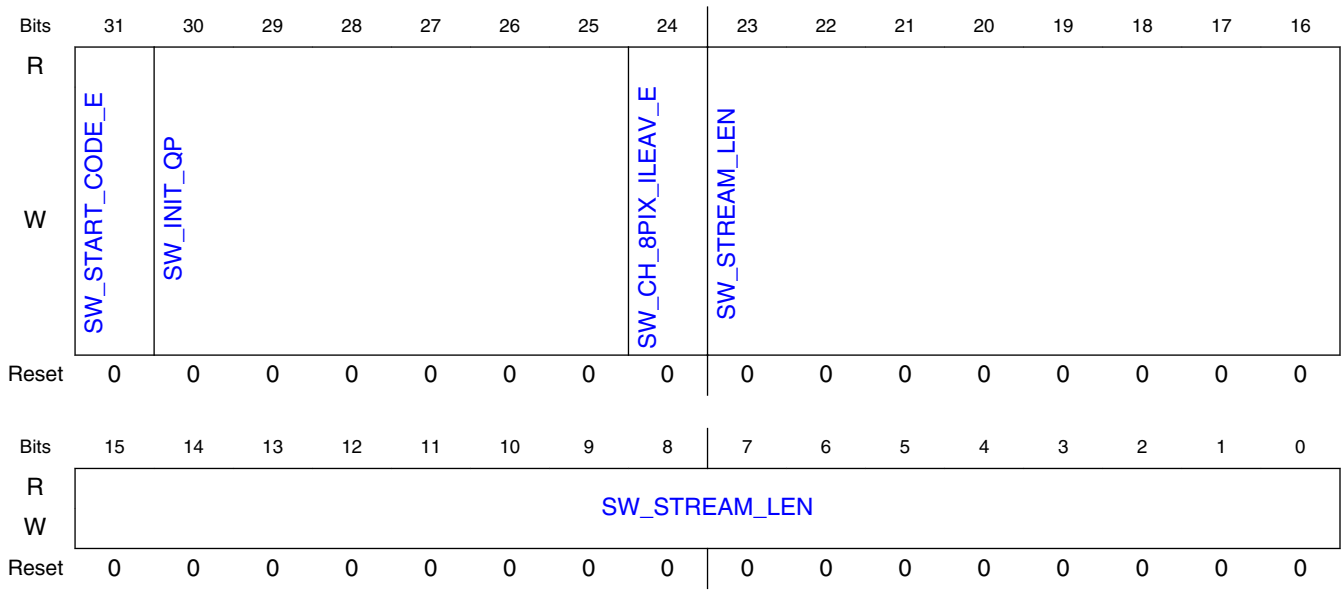
Field	Function
31-26 SW_STRM_START_BIT	Exact bit of stream start word where decoding can be started (assosiates with sw_rlc_vlc_base)
25 SW_SYNC_MARKER_E	Sync markers enable For progressive JPEG, this indicates that there are restart markers in the stream after restart interval steps. 0b - synch markers are not used 1b - synch markers are used.
24 SW_TYPE1_QUANT_E	MPEG4: Type 1 quantization enable • '0' = type 2 inverse Q method • '1' = type 1 inverse Q method (Q-tables used) H264 (h264_high config): scaling matrix enable • '0' = normal transform • '1' = use scaling matrix for transform (read from external memory)
23-19 SW_CH_QP_OFFSET	Chroma Qp filter offset. (For H.264 this offset concerns Cb only)
18-14 SW_CH_QP_OFFSET2	Chroma Qp filter offset for cr type
13-1 —	Reserved.
0 SW_FIELDPIC_FLAG_E	Flag for streamd that field_pic_flag exists in stream

16.1.6.2.4 Decoder control register 3 (stream buffer information) (SWREG6)

16.1.6.2.4.1 Offset

Register	Offset
SWREG6	18h

16.1.6.2.4.2 Diagram



16.1.6.2.4.3 Fields

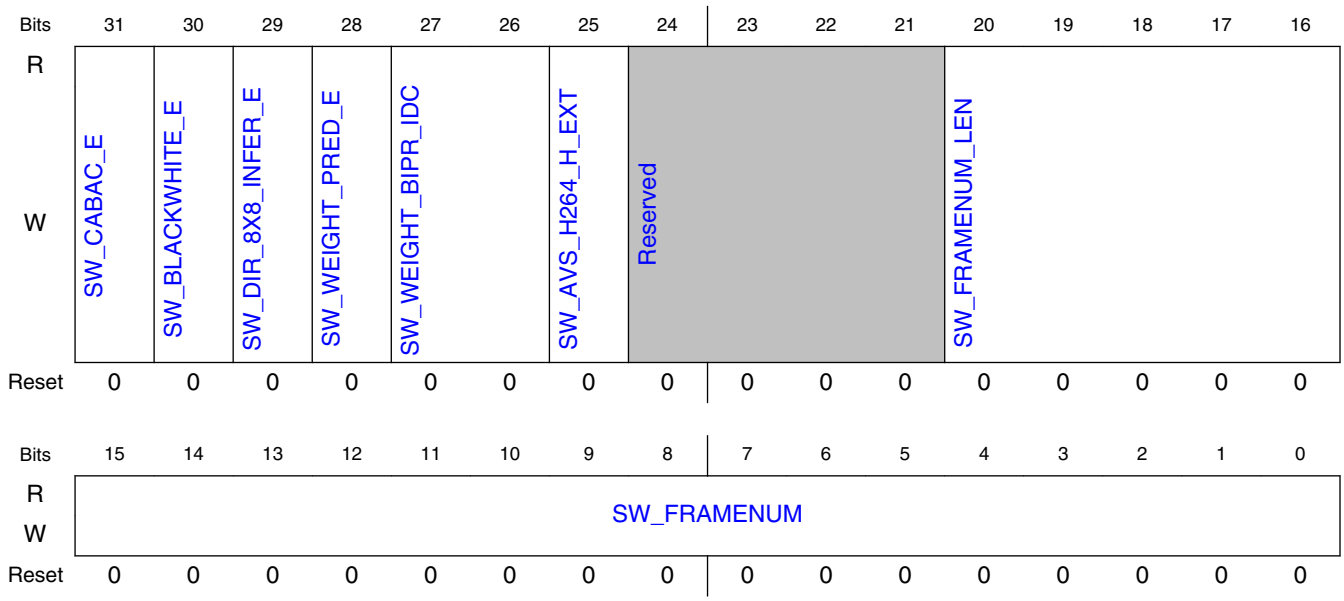
Field	Function
31 SW_START_CODE_E	Bit for indicating stream start code existence: 0b - stream doesn't contain start codes 1b - stream contains start codes
30-25 SW_INIT_QP	Initial value for quantization parameter (picture quantizer).
24 SW_CH_8PIX_ILEAV_E	Enable for additional chrominance data format writing where decoder writes chrominance in group of 8 pixels of Cb and then corresponding 8 pixels of Cr. Data is written to sw_dec_ch8pix_base. Cannot be used if tiled mode is enabled
23-0 SW_STREAM_LEN	Amount of stream data bytes in input buffer. If the given buffer size is not enough for finishing the picture the corresponding interrupt is given and new stream buffer base address and stream buffer size information should be given (assosiates with sw_rlc_vlc_base). For VC-1/VP6 the buffer must include data for one picture/slice of the picture For H264/MPEG4/H263/MPEG2/MPEG1 the buffer must include at least data for one slice/VP of the picture For JPEG the buffer size must be a multiple of 256 bytes or the amount of data for one picture.

16.1.6.2.5 Decoder control register 4 (H264, VC-1, VP6 and progressive JPEG control) (SWREG7)

16.1.6.2.5.1 Offset

Register	Offset
SWREG7	1Ch

16.1.6.2.5.2 Diagram



16.1.6.2.5.3 Fields

Field	Function
31 SW_CABAC_E	CABAC enable
30 SW_BLACKWHITE_E	0b - 4:2:0 sampling format 1b - 4:0:0 sampling format (H264 monochroma)
29 SW_DIR_8X8_INFER_E	Specifies the method to use to derive luma motion vectors in B_skip, B_Direct_16x16 and B_direct_8x8_inference_flag (see direct_8x8_inference flag)
28 SW_WEIGHT_PRED_E	Weighted prediction enable for P slices
27-26 SW_WEIGHT_BIPR_IDC	weighted prediction specification for B slices: 00b - default weighted prediction is applied to B slices 01b - explicit weighted prediction shall be applied to B slices 10b - implicit weighted prediction shall be applied to B slices
25	Resolution extension to support 4k resolution for AVS/H264. Used as MSB of sw_pic_mb_height

Table continues on the next page...

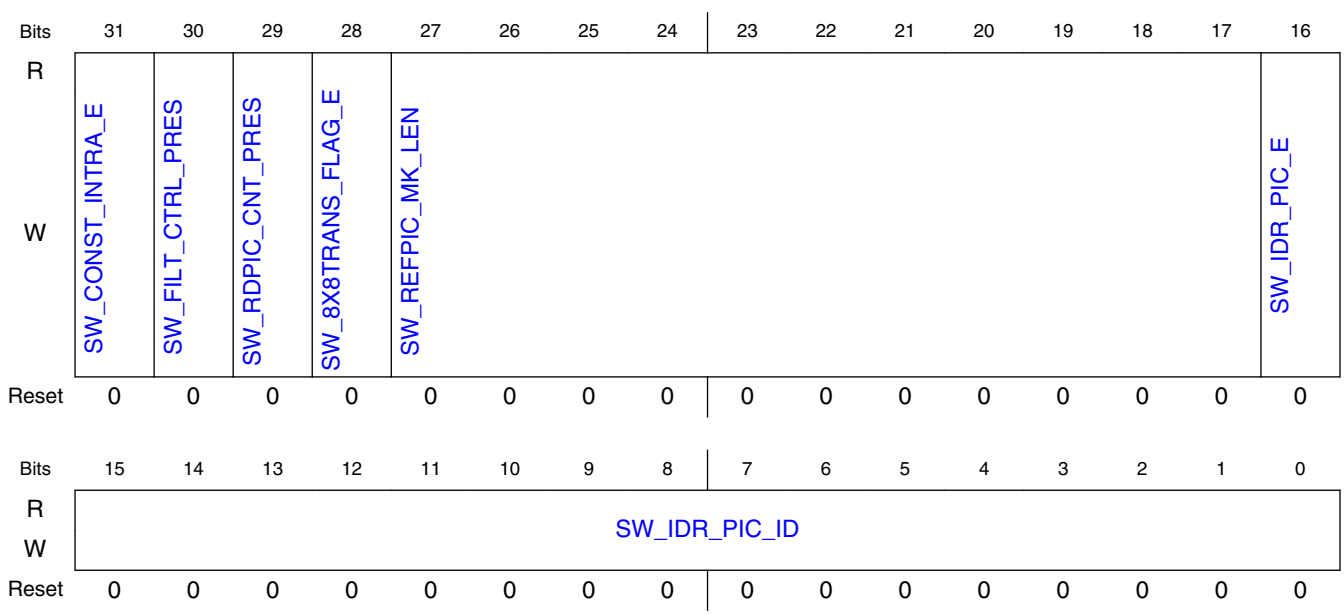
Field	Function
SW_AVS_H264_H_EXT	
24-21 —	Reserved.
20-16 SW_FRAMENUM_LEN	H.264: Bit length of frame_num in data stream RV: frame size length. Informs how many bits in stream are used for frame size (HW discards these bits)
15-0 SW_FRAMENUM	current frame_num, used to identify short-term reference frames. Used in reference picture reordering

16.1.6.2.6 Decoder control register 5 (H264, VC-1, VP6, Progressive JPEG and RV control) (SWREG8)

16.1.6.2.6.1 Offset

Register	Offset
SWREG8	20h

16.1.6.2.6.2 Diagram



16.1.6.2.6.3 Fields

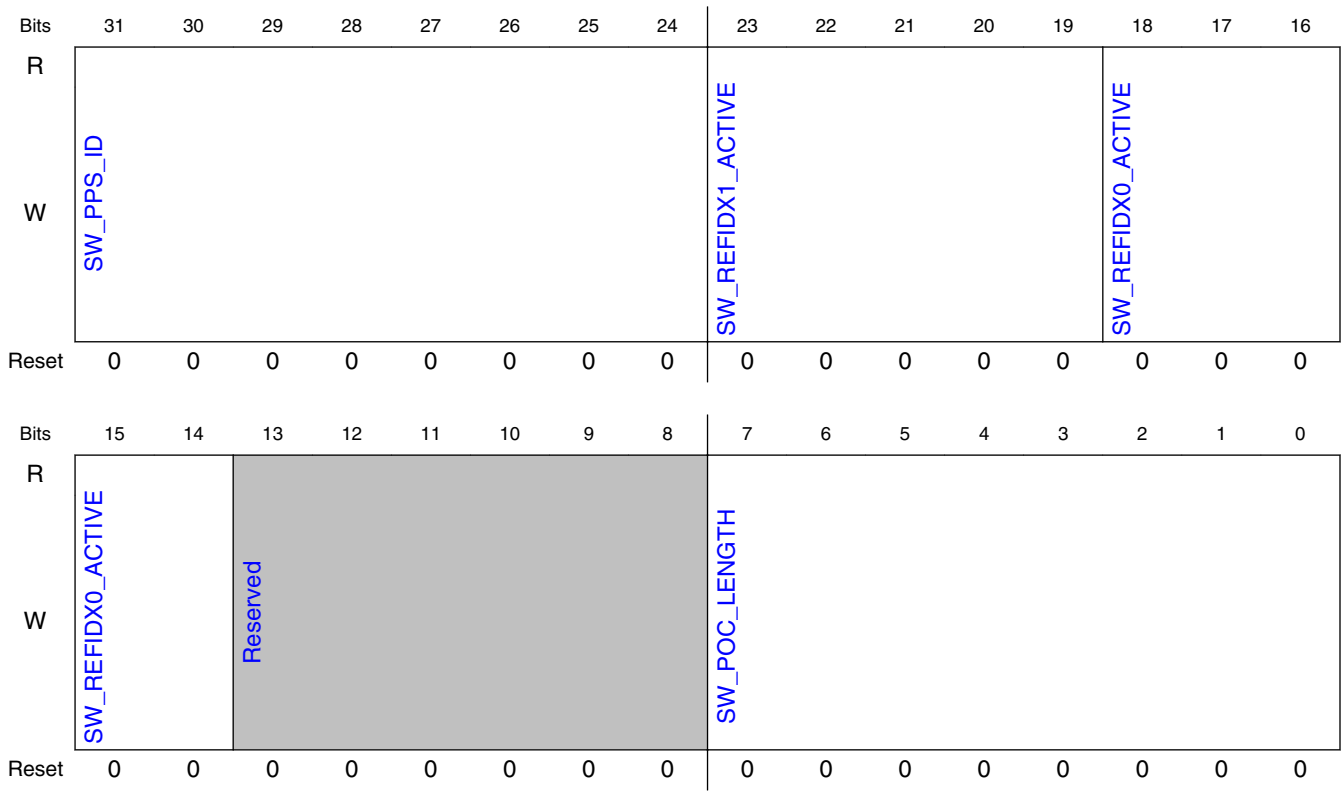
Field	Function
31 SW_CONST_INTRA_E	constrained_intra_pred_flag equal to 1 specifies that intra prediction uses only neighbouring intra macroblocks in prediction. When equal to 0 also neighbouring inter macroblocks are used in intra prediction process.
30 SW_FILT_CTRL_PRES	deblocking_filter_control_present_flag indicates whether extra variables controlling characteristics of the deblocking filter are present in the slice header.
29 SW_RDPIC_CNT_PRES	redundant_pic_cnt_present_flag specifies whether redundant_pic_cnt syntax elements are present in the slice header.
28 SW_8X8TRANS_FLAG_E	8x8 transform flag enable for stream decoding
27-17 SW_REFPIC_MK_LEN	Length of decoded reference picture marking bits
16 SW_IDR_PIC_E	IDR (instantaneous decoding refresh) picture flag.
15-0 SW_IDR_PIC_ID	idr_pic_id, identifies IDR (instantaneous decoding refresh) picture

16.1.6.2.7 Decoder control register 6 / base address for MB-control (RLC) / VC-1 intensity control 0/ VP6,VP7,VP8 ctrl-stream length/ RV pic slice amount (SWREG9)

16.1.6.2.7.1 Offset

Register	Offset
SWREG9	24h

16.1.6.2.7.2 Diagram



16.1.6.2.7.3 Fields

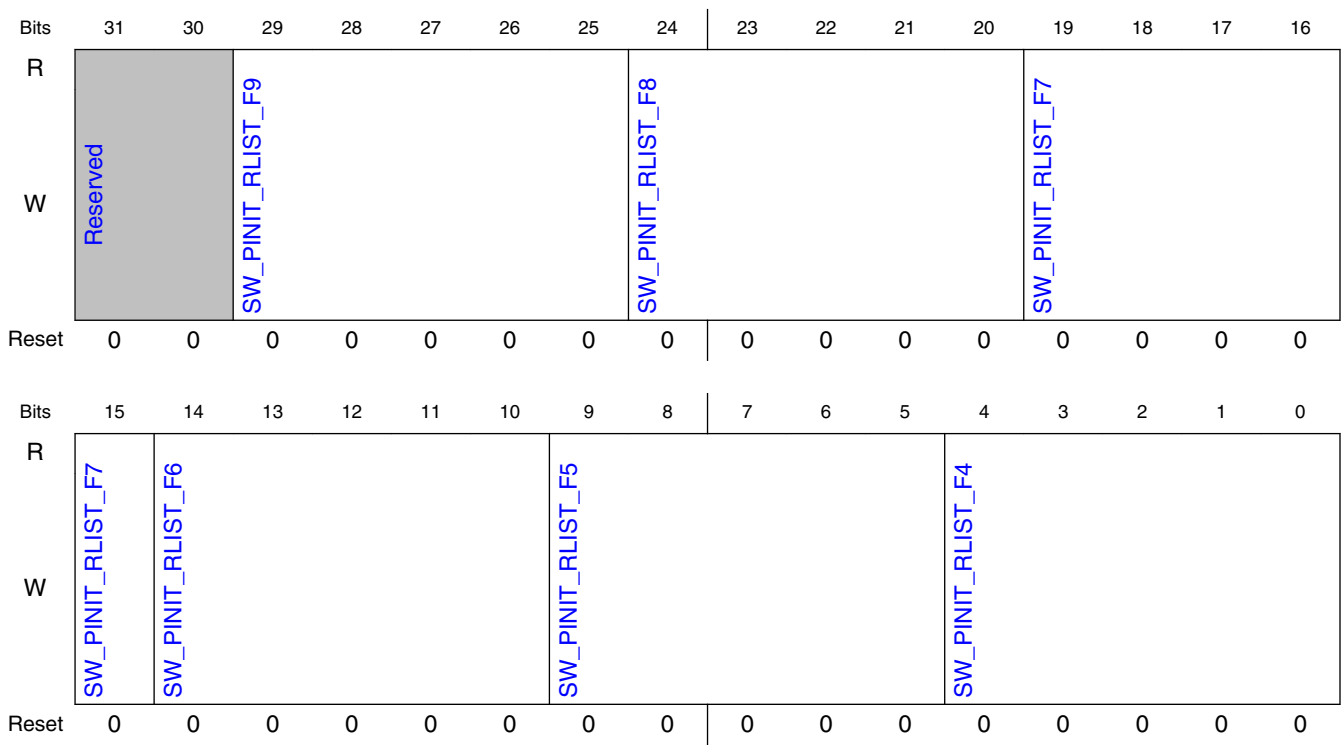
Field	Function
31-24 SW_PPS_ID	pic_parameter_set_id, identifies the picture parameter set that is referred to in the slice header.
23-19 SW_REFIDX1_ACTIVE	Specifies the maximum reference index that can be used while decoding inter predicted macro blocks.
18-14 SW_REFIDX0_ACTIVE	Specifies the maximum reference index that can be used while decoding inter predicted macro blocks. This is same as in previous decoders (width increased with q bit)
13-8 —	Reserved.
7-0 SW_POC_LENGTH	Length of picture order count field in stream

16.1.6.2.8 Base address for differential motion vector base address (RLC-mode) /H264 P initial fwd ref pic list register (4-9)/ VC-1 intensity control 1/ VP7 and VP8 segmentation base register (SWREG10)

16.1.6.2.8.1 Offset

Register	Offset
SWREG10	28h

16.1.6.2.8.2 Diagram



16.1.6.2.8.3 Fields

Field	Function
31-30 —	Reserved.
29-25 SW_PINIT_RLIST_F9	Initial reference picture list for P forward picid 9
24-20	Initial reference picture list for P forward picid 8

Table continues on the next page...

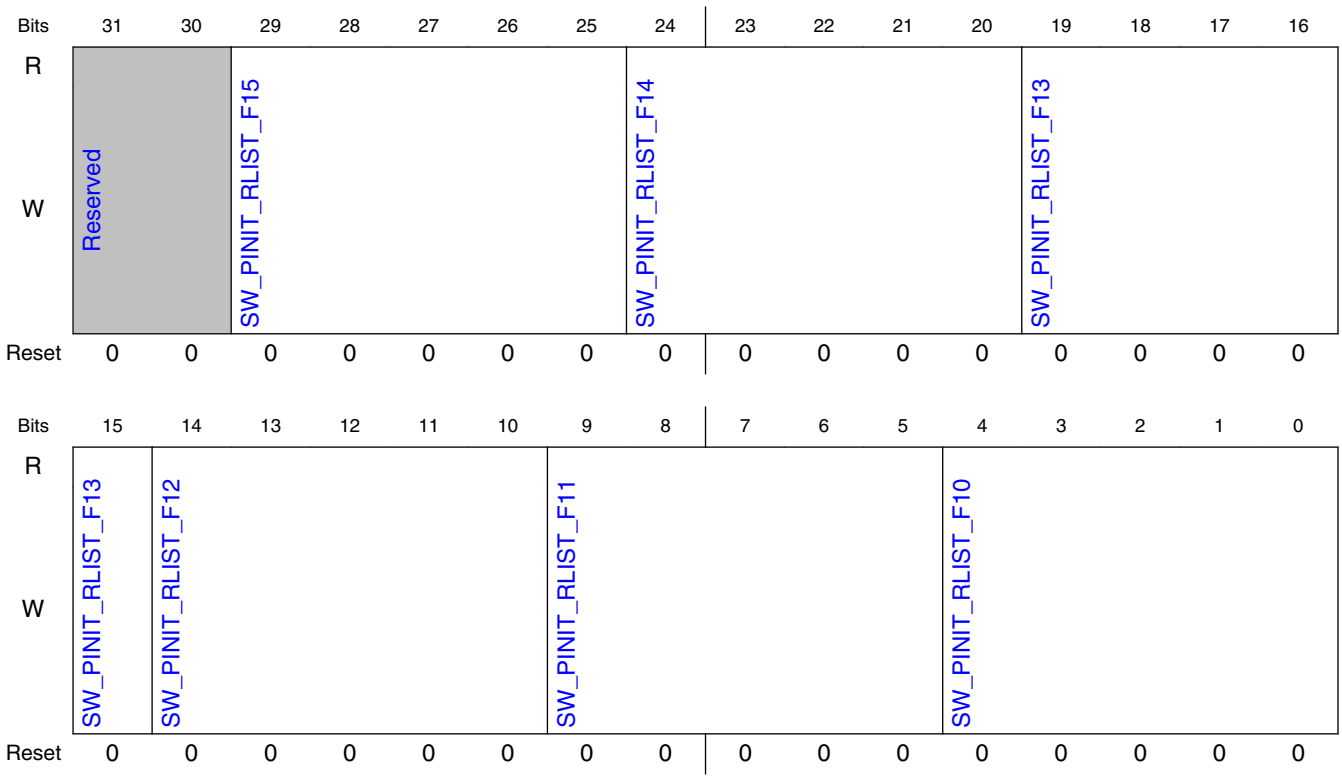
Field	Function
SW_PINIT_RLIST_F8	
19-15 SW_PINIT_RLIST_F7	Initial reference picture list for P forward picid 7
14-10 SW_PINIT_RLIST_F6	Initial reference picture list for P forward picid 6
9-5 SW_PINIT_RLIST_F5	Initial reference picture list for P forward picid 5
4-0 SW_PINIT_RLIST_F4	Initial reference picture list for P forward picid 4

16.1.6.2.9 Decoder control register 7 (VLC) / base address for H.264 intra prediction 4x4 / base address for MPEG-4 DC component (RLC) / H264 P initial fwd ref pic list register (10-15) / VC-1 intensity control 2 (SWREG11)

16.1.6.2.9.1 Offset

Register	Offset
SWREG11	2Ch

16.1.6.2.9.2 Diagram



16.1.6.2.9.3 Fields

Field	Function
31-30 —	Reserved.
29-25 SW_PINIT_RLIST_F15	Initial reference picture list for P forward picid 15
24-20 SW_PINIT_RLIST_F14	Initial reference picture list for P forward picid 14
19-15 SW_PINIT_RLIST_F13	Initial reference picture list for P forward picid 13
14-10 SW_PINIT_RLIST_F12	Initial reference picture list for P forward picid 12
9-5 SW_PINIT_RLIST_F11	Initial reference picture list for P forward picid 11

Table continues on the next page...

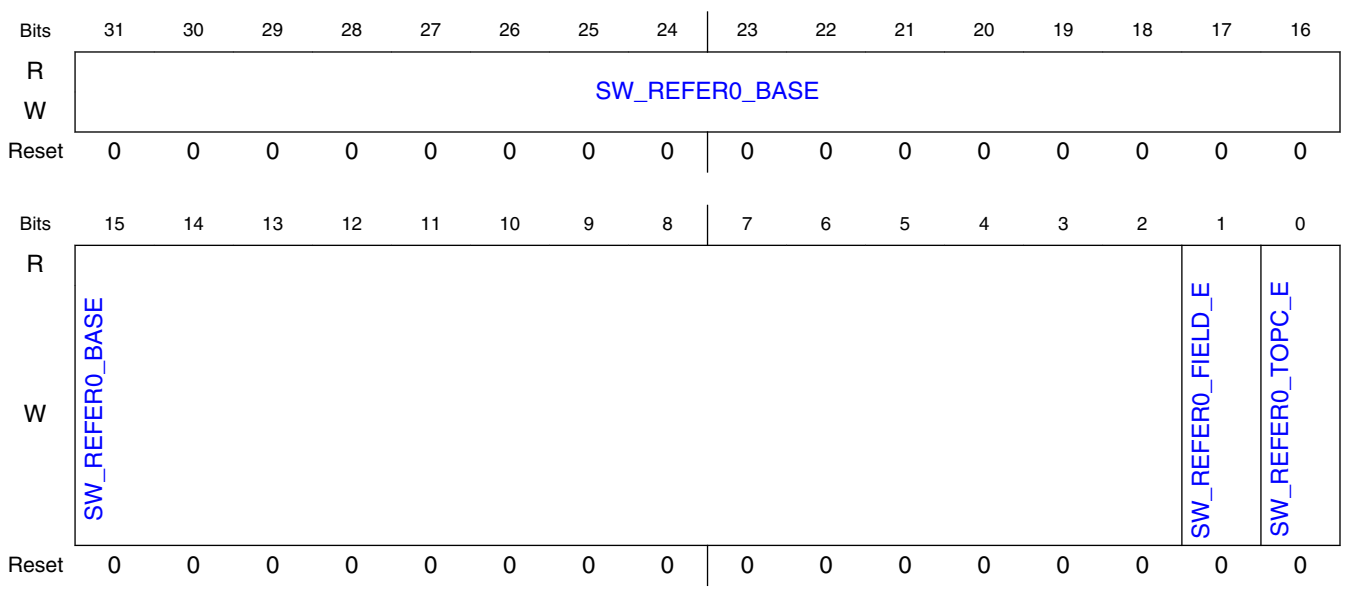
Field	Function
4-0 SW_PINIT_RLIST_F10	Initial reference picture list for P forward picid 10

16.1.6.2.10 Base address for reference picture index 0 / base address for JPEG decoder output chrominance picture (SWREG14)

16.1.6.2.10.1 Offset

Register	Offset
SWREG14	38h

16.1.6.2.10.2 Diagram



16.1.6.2.10.3 Fields

Field	Function
31-2 SW_REFER0_BASE	Base address for reference picture index 0. See picture index definition from toplevel_sp
1	Refer picture consist of single fields or frame: 0b - reference picture consists of frame

Table continues on the next page...

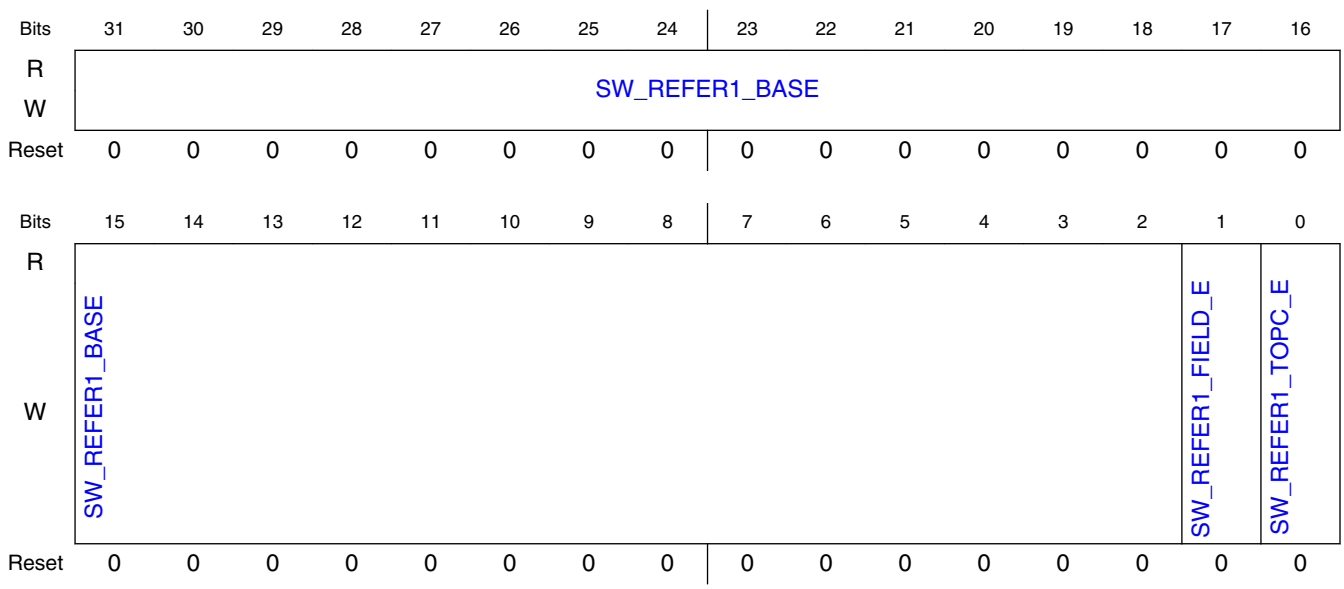
Field	Function
SW_REFER0_FIELD_E	1b - reference picture consists of fields
0 SW_REFER0_TOPC_E	Which field of reference picture is closer to current picture: 0b - Bottom field is closer to current picture 1b - Top field is closer to current picture

16.1.6.2.11 Base address for reference picture index 1 / JPEG control (SWREG15)

16.1.6.2.11.1 Offset

Register	Offset
SWREG15	3Ch

16.1.6.2.11.2 Diagram



16.1.6.2.11.3 Fields

Field	Function
31-2 SW_REFER1_BASE	Base address for reference picture index 1. See picture index definition from toplevel_sp. For VP8 this base address is used as Chrominance base address for reference picture 0 (if vp8 stride configuration is enabled)

Table continues on the next page...

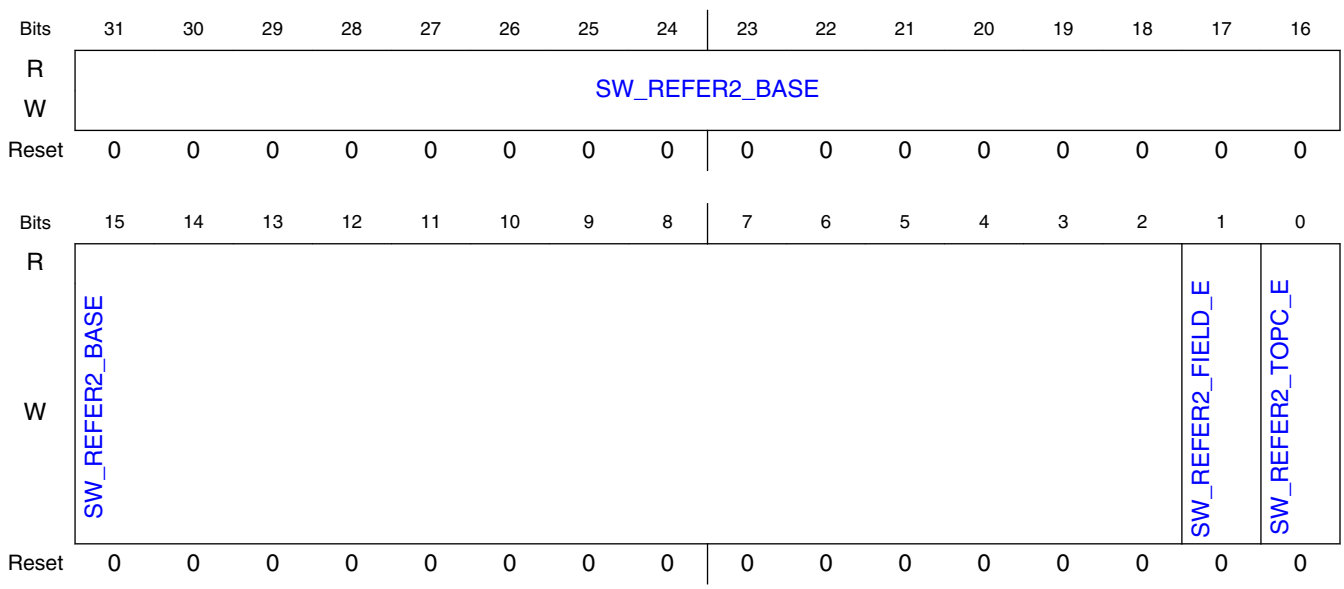
Field	Function
1 SW_REFER1_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER1_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.12 Base address for reference picture index 2 / List of VLC code lengths in first JPEG AC table (SWREG16)

16.1.6.2.12.1 Offset

Register	Offset
SWREG16	40h

16.1.6.2.12.2 Diagram



16.1.6.2.12.3 Fields

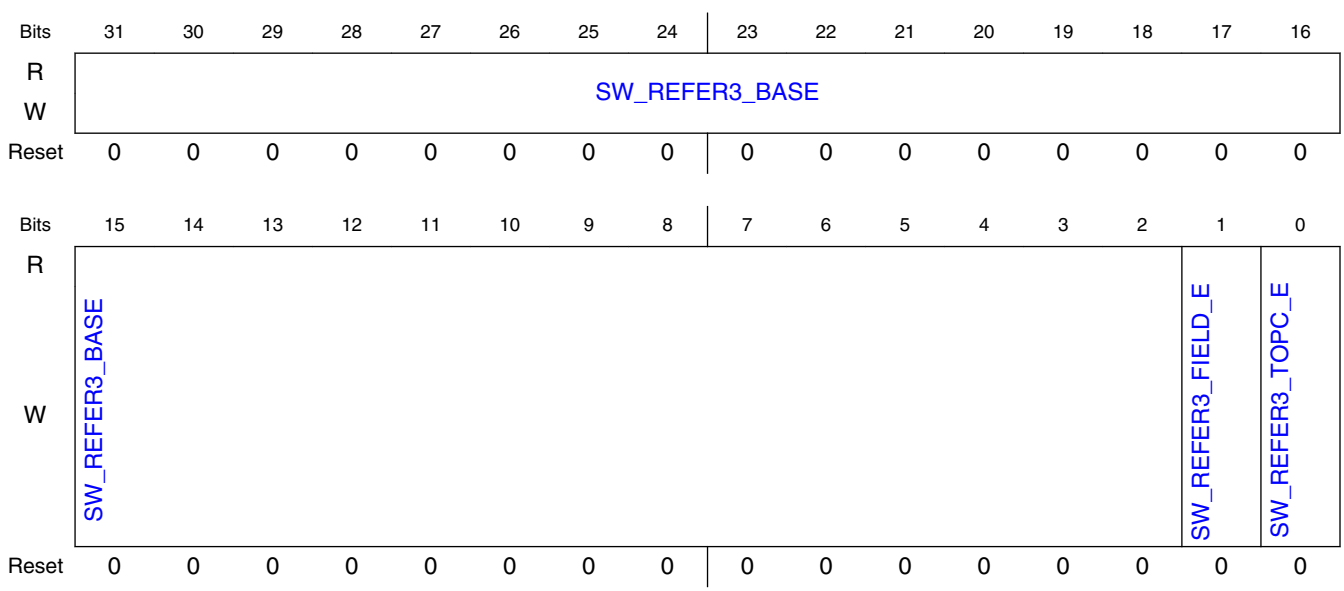
Field	Function
31-2 SW_REFER2_BASE	Base address for reference picture index 2. See picture index definition from toplevel_sp. For VP8 video this base address is used as Golden reference chrominance base address (if vp8 stride configuration is enabled)
1 SW_REFER2_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER2_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.13 Base address for reference picture index 3 / List of VLC code lengths in first JPEG AC table (SWREG17)

16.1.6.2.13.1 Offset

Register	Offset
SWREG17	44h

16.1.6.2.13.2 Diagram



16.1.6.2.13.3 Fields

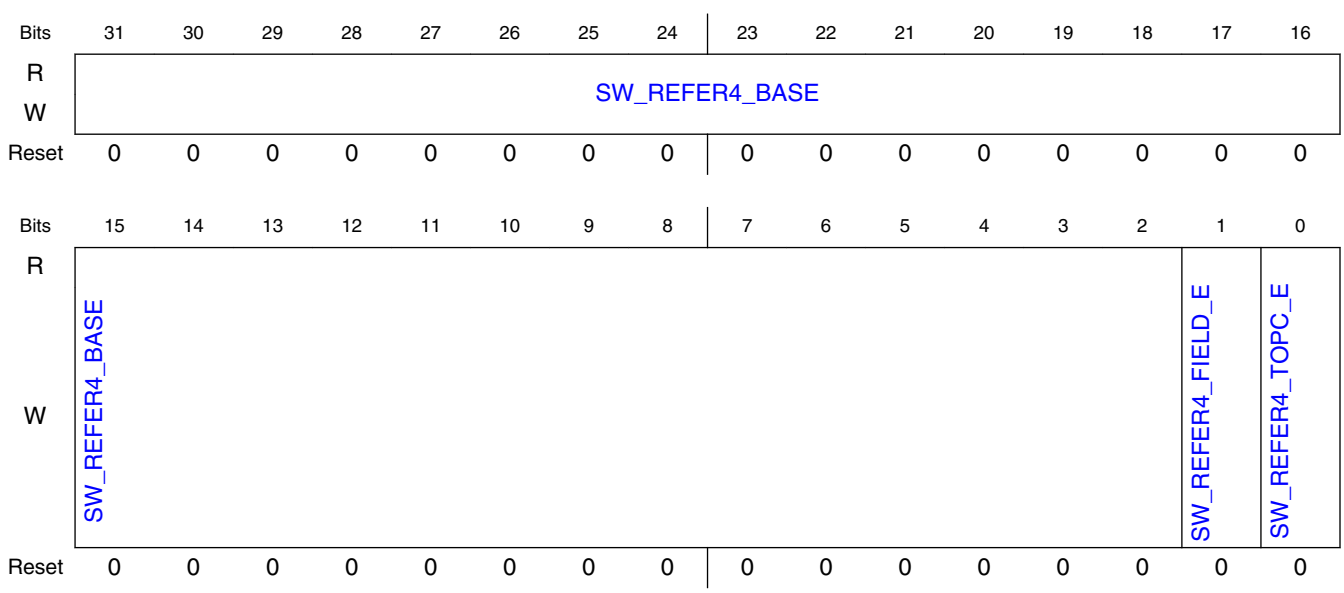
Field	Function
31-2 SW_REFER3_BASE	Base address for reference picture index 3. See picture index definition from toplevel_sp. For VP8 video this base address is used as Alternate reference chrominance base address (if vp8 stride configuration is enabled)
1 SW_REFER3_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER3_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.14 Base address for reference picture index 4 / VC1 control / MPEG4 MVD control/ List of VLC code lengths in first JPEG AC table / VC-1 intensity control 4 / VP6/VP7, VP8 Golden refer picture base (SWREG18)

16.1.6.2.14.1 Offset

Register	Offset
SWREG18	48h

16.1.6.2.14.2 Diagram



16.1.6.2.14.3 Fields

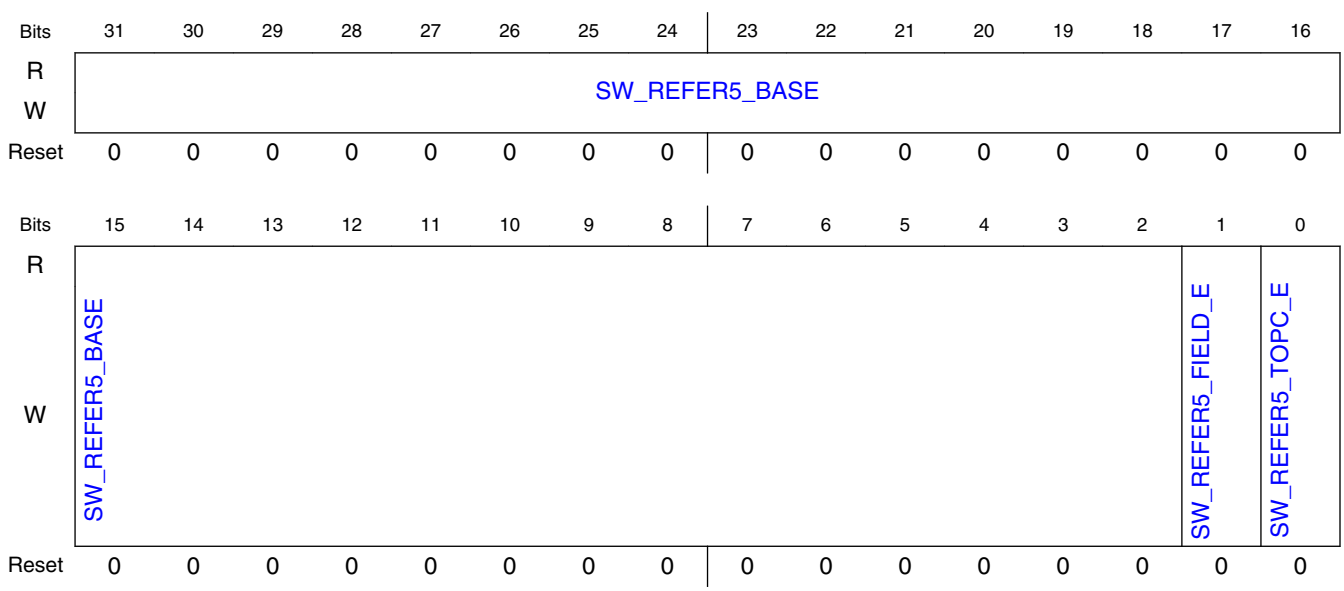
Field	Function
31-2 SW_REFER4_BASE	H264: Base address for reference picture index 4 VP6/VP7/VP8: Base address for Golden reference picture (corresponds picid 4)
1 SW_REFER4_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER4_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.15 Base address for reference picture index 5 / MPEG4 TRB/TRD delta 0 / VC-1 intensity control 3 List of VLC code lengths in first/second JPEG AC table / VP6/VP7 scan maps (SWREG19)

16.1.6.2.15.1 Offset

Register	Offset
SWREG19	4Ch

16.1.6.2.15.2 Diagram



16.1.6.2.15.3 Fields

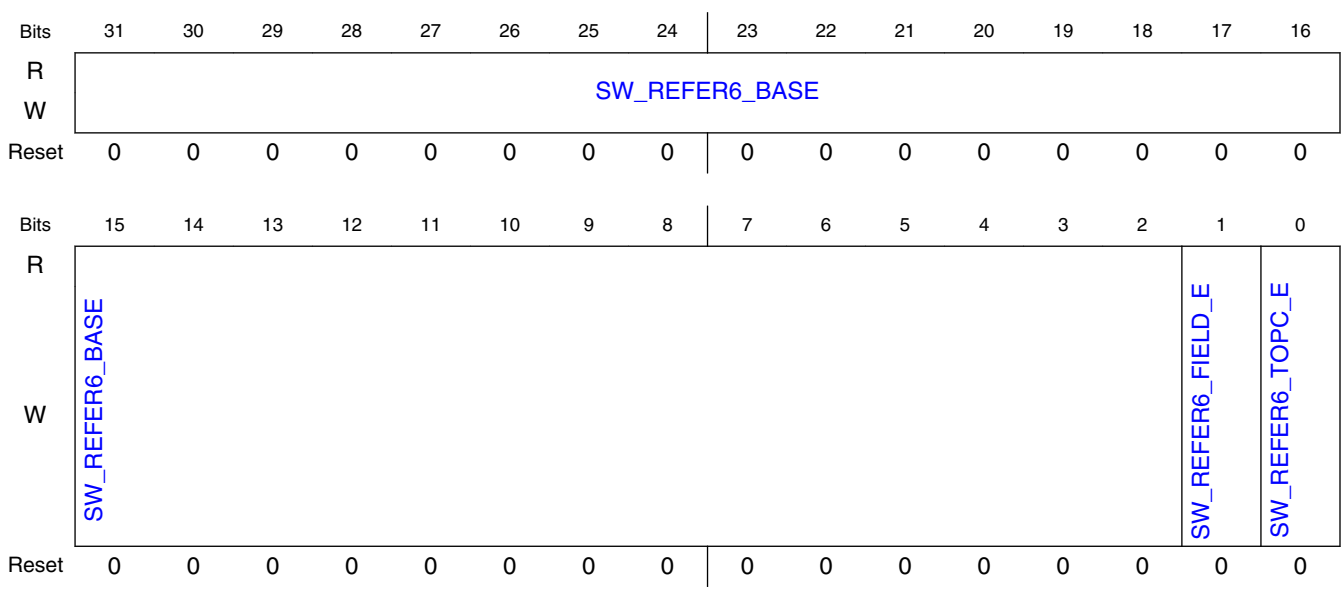
Field	Function
31-2 SW_REFER5_BASE	H.264: Base address for reference picture index 5 VP8: Base address for alternate reference picture (corresponds picid 5)
1 SW_REFER5_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER5_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.16 Base address for reference picture index 6 / / MPEG4 TRB/TRD delta -1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG20)

16.1.6.2.16.1 Offset

Register	Offset
SWREG20	50h

16.1.6.2.16.2 Diagram



16.1.6.2.16.3 Fields

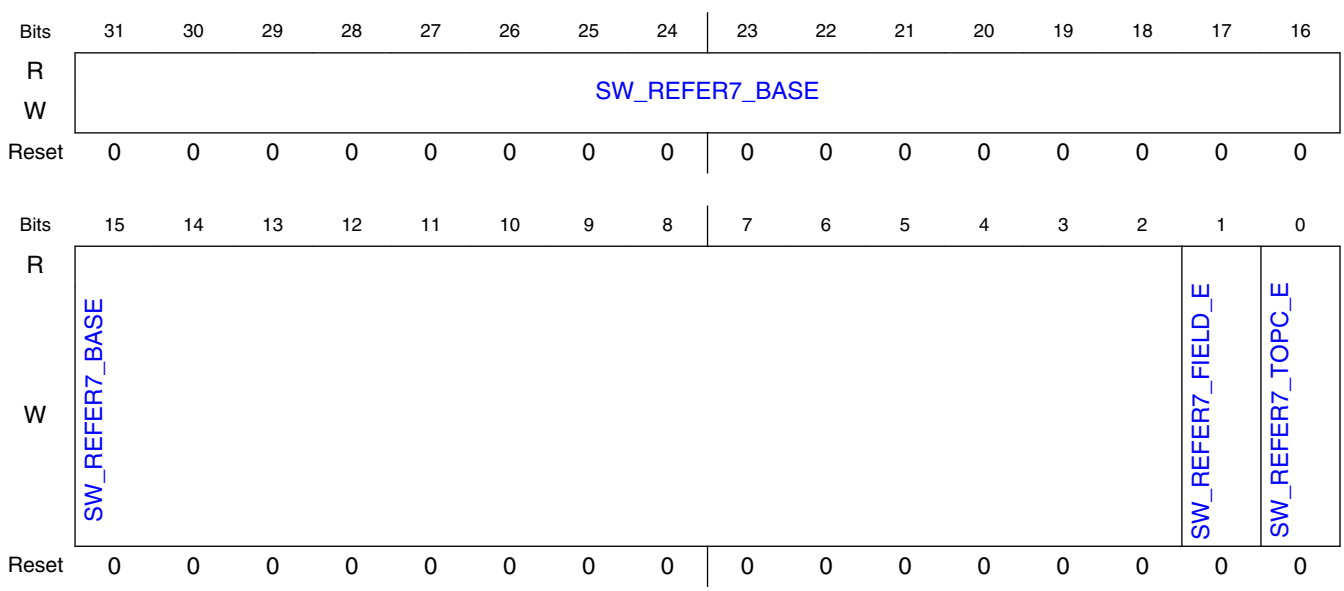
Field	Function
31-2 SW_REFER6_BASE	Base address for reference picture index 6. For VP8 video this base address is used as decoder output chrominance base address (if vp8 stride configuration is enabled)
1 SW_REFER6_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER6_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.17 Base address for reference picture index 7 / MPEG4 TRB/TRD delta 1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG21)

16.1.6.2.17.1 Offset

Register	Offset
SWREG21	54h

16.1.6.2.17.2 Diagram



16.1.6.2.17.3 Fields

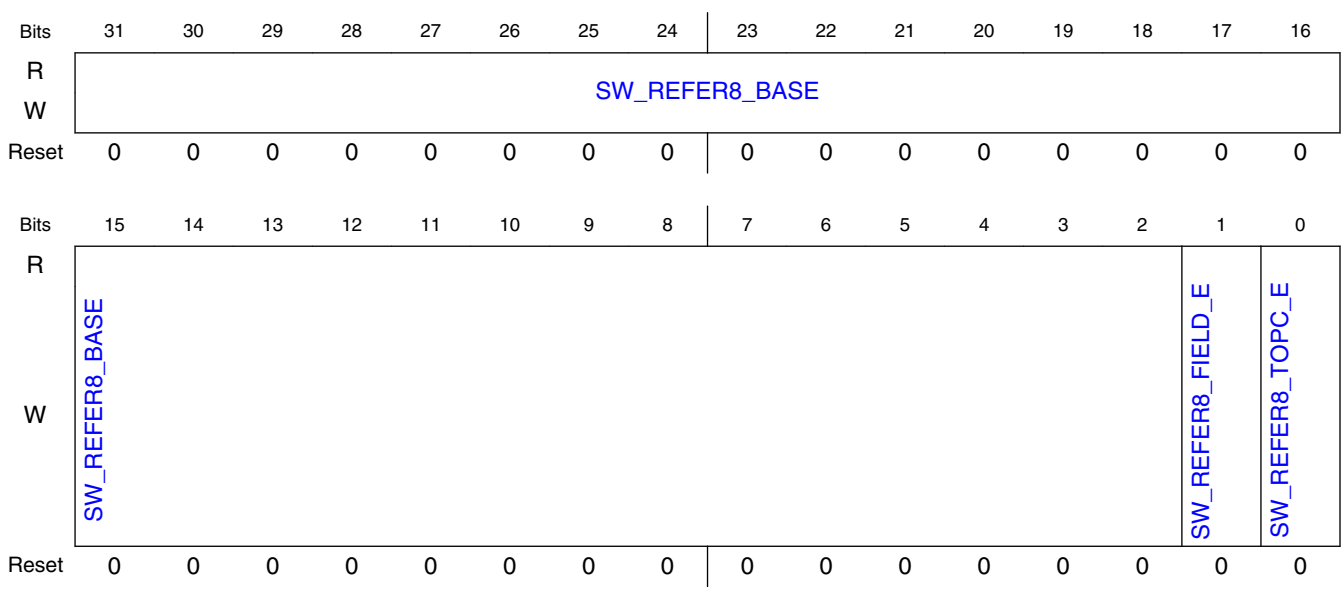
Field	Function
31-2 SW_REFER7_BASE	Base address for reference picture index 7
1 SW_REFER7_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER7_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.18 Base address for reference picture index 8 / List of VLC code lengths in second JPEG AC table / VP6 scan maps / VP7,VP8 DCT stream 1 base (SWREG22)

16.1.6.2.18.1 Offset

Register	Offset
SWREG22	58h

16.1.6.2.18.2 Diagram



16.1.6.2.18.3 Fields

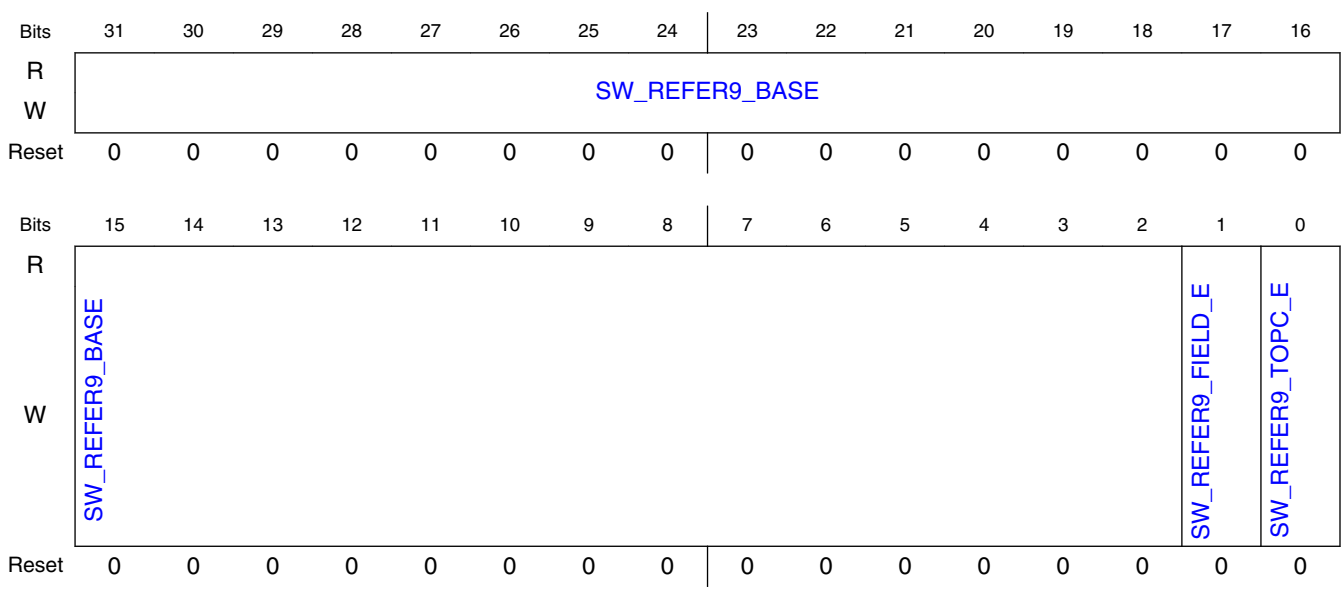
Field	Function
31-2 SW_REFER8_BASE	Base address for reference picture index 8
1 SW_REFER8_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER8_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.19 Base address for reference picture index 9 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 2 base (SWREG23)

16.1.6.2.19.1 Offset

Register	Offset
SWREG23	5Ch

16.1.6.2.19.2 Diagram



16.1.6.2.19.3 Fields

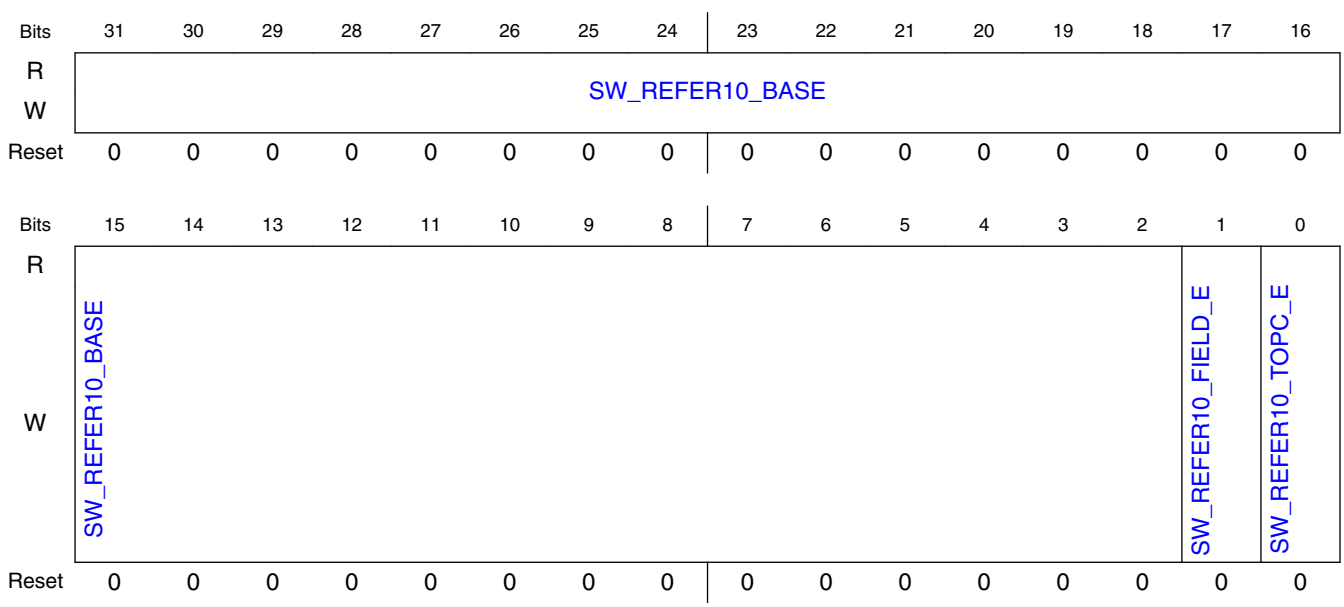
Field	Function
31-2 SW_REFER9_BASE	Base address for reference picture index 9
1 SW_REFER9_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER9_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.20 Base address for reference picture index 10 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 3 base (SWREG24)

16.1.6.2.20.1 Offset

Register	Offset
SWREG24	60h

16.1.6.2.20.2 Diagram



16.1.6.2.20.3 Fields

Field	Function
31-2 SW_REFER10_BASE	Base address for reference picture index 10
1 SW_REFER10_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER10_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.21 Base address for reference picture index 11 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 4 base (SWREG25)

16.1.6.2.21.1 Offset

Register	Offset
SWREG25	64h

16.1.6.2.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_BASE														SW_REFER11_FIELD_E	SW_REFER11_TOPC_E
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.2.21.3 Fields

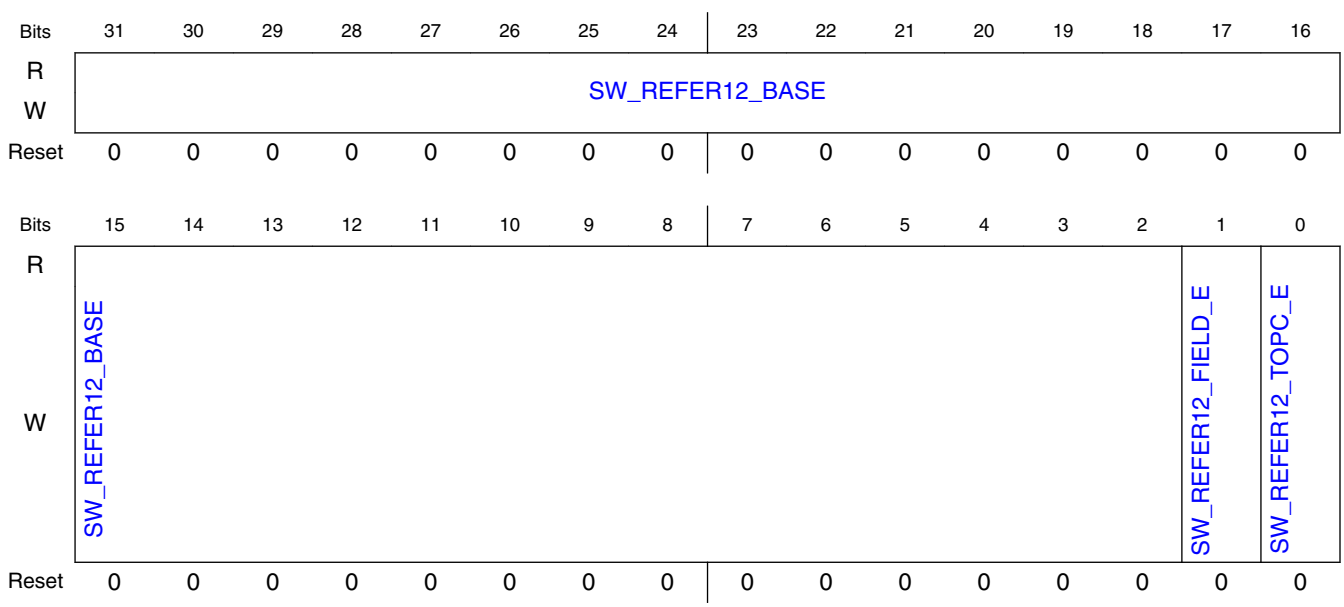
Field	Function
31-2 SW_REFER11_BASE	Base address for reference picture index 11
1 SW_REFER11_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER11_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.22 Base address for reference picture index 12 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 5 base (SWREG26)

16.1.6.2.22.1 Offset

Register	Offset
SWREG26	68h

16.1.6.2.22.2 Diagram



16.1.6.2.22.3 Fields

Field	Function
31-2 SW_REFER12_BASE	Base address for reference picture index 12
1 SW_REFER12_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER12_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.23 Base address for reference picture index 13 / VC-1 bitpl mbctrl or VP6,VP7,VP8 ctrl stream base /Progressive JPEG DC table (SWREG27)

16.1.6.2.23.1 Offset

Register	Offset
SWREG27	6Ch

16.1.6.2.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_BASE														SW_REFER13_FIELD_E	SW_REFER13_TOPC_E
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.2.23.3 Fields

Field	Function
31-2 SW_REFER13_BASE	Base address for reference picture index 13
1 SW_REFER13_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER13_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.24 Base address for reference picture index 14 / VP6 scan maps / Progressive JPEG DC table / VP7,VP8 DCT stream 6 base (SWREG28)

16.1.6.2.24.1 Offset

Register	Offset
SWREG28	70h

16.1.6.2.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_REFER14_BASE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_REFER14_BASE														SW_REFER14_FIELD_E	SW_REFER14_TOPC_E
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.2.24.3 Fields

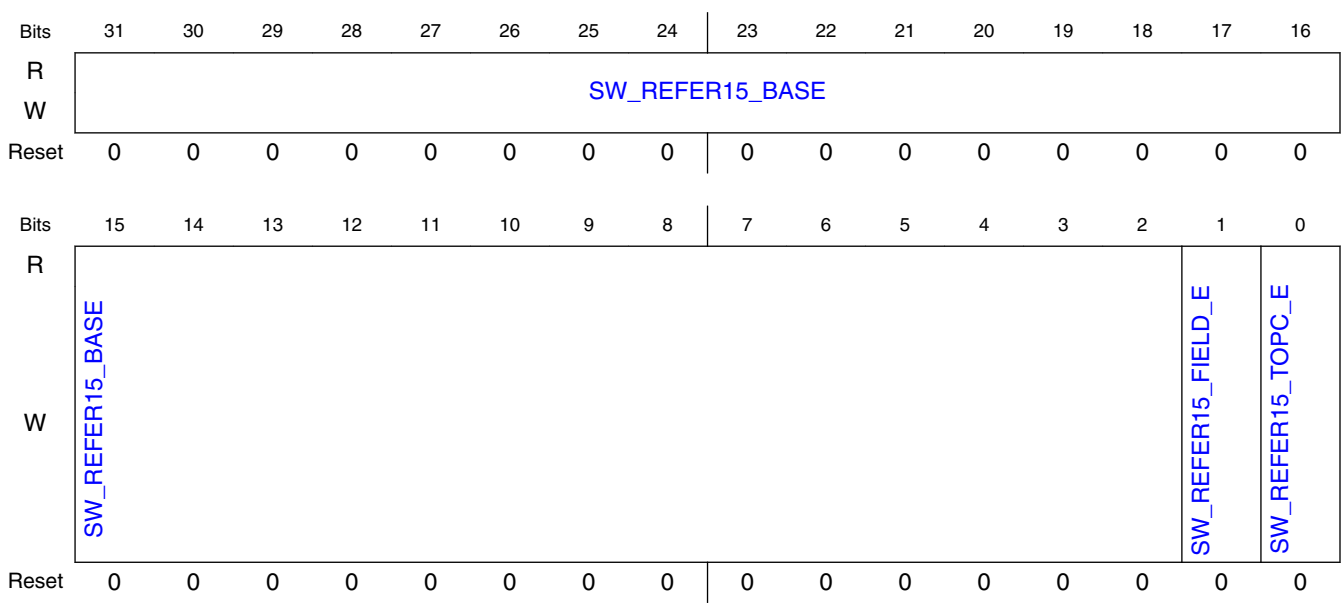
Field	Function
31-2 SW_REFER14_BASE	Base address for reference picture index 14
1 SW_REFER14_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER14_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.25 Base address for reference picture index 15 / VP6 scan maps / VP7,VP8 DCT stream 7 base (SWREG29)

16.1.6.2.25.1 Offset

Register	Offset
SWREG29	74h

16.1.6.2.25.2 Diagram



16.1.6.2.25.3 Fields

Field	Function
31-2 SW_REFER15_BASE	Base address for reference picture index 15. For Multi View Coding this base address refers to inter view base address
1 SW_REFER15_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER15_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

16.1.6.2.26 Reference picture numbers for index 0 and 1 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter mb level adjusts (SWREG30)

16.1.6.2.26.1 Offset

Register	Offset
SWREG30	78h

16.1.6.2.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.2.26.3 Fields

Field	Function
31-16	Number for reference picture index 1

Table continues on the next page...

Field	Function
SW_REFER1_NBR	
15-0 SW_REFER0_NBR	Number for reference picture index 0

16.1.6.2.27 Reference picture numbers for index 2 and 3 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter ref pic level adjusts (SWREG31)

16.1.6.2.27.1 Offset

Register	Offset
SWREG31	7Ch

16.1.6.2.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.2.27.3 Fields

Field	Function
31-16 SW_REFER3_NBR	Number for reference picture index 3
15-0 SW_REFER2_NBR	Number for reference picture index 2

16.1.6.2.28 Reference picture numbers for index 4 and 5 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter levels (SWREG32)

16.1.6.2.28.1 Offset

Register	Offset
SWREG32	80h

16.1.6.2.28.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.2.28.3 Fields

Field	Function
31-16 SW_REFER5_NBR	Number for reference picture index 5
15-0 SW_REFER4_NBR	Number for reference picture index 4

16.1.6.2.29 Reference picture numbers for index 6 and 7 (H264 VLC) / VP6 scan maps / VP7,VP8 quantization values (SWREG33)

16.1.6.2.29.1 Offset

Register	Offset
SWREG33	84h

16.1.6.2.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.2.29.3 Fields

Field	Function
31-16 SW_REFER7_NBR	Number for reference picture index 7
15-0 SW_REFER6_NBR	Number for reference picture index 6

16.1.6.2.30 Reference picture numbers for index 8 and 9 (H264 VLC) / MPEG4, VC1, VPx prediction filter taps (SWREG34)

16.1.6.2.30.1 Offset

Register	Offset
SWREG34	88h

16.1.6.2.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.2.30.3 Fields

Field	Function
31-16 SW_REFER9_NBR	Number for reference picture index 9
15-0 SW_REFER8_NBR	Number for reference picture index 8

16.1.6.2.31 Reference picture numbers for index 10 and 11 (H264 VLC) / VC1, VPx prediction filter taps (SWREG35)

16.1.6.2.31.1 Offset

Register	Offset
SWREG35	8Ch

16.1.6.2.31.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.2.31.3 Fields

Field	Function
31-16 SW_REFER11_NBR	Number for reference picture index 11
15-0 SW_REFER10_NBR	Number for reference picture index 10

16.1.6.2.32 Reference picture numbers for index 12 and 13 (H264 VLC) / VC1, VPx prediction filter taps (SWREG36)

16.1.6.2.32.1 Offset

Register	Offset
SWREG36	90h

16.1.6.2.32.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.2.32.3 Fields

Field	Function
31-16 SW_REFER13_NBR	Number for reference picture index 13
15-0 SW_REFER12_NBR	Number for reference picture index 12

16.1.6.2.33 Reference picture numbers for index 14 and 15 (H264 VLC) / VPx prediction filter taps (SWREG37)

16.1.6.2.33.1 Offset

Register	Offset
SWREG37	94h

16.1.6.2.33.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.2.33.3 Fields

Field	Function
31-16 SW_REFER15_NBR	Number for reference picture index 15
15-0 SW_REFER14_NBR	Number for reference picture index 14

16.1.6.2.34 Reference picture long term flags (H264 VLC) / VPx prediction filter taps (SWREG38)

16.1.6.2.34.1 Offset

Register	Offset
SWREG38	98h

16.1.6.2.34.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER_LTERM_E															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER_LTERM_E															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.2.34.3 Fields

Field	Function
31-0 SW_REFER_LTERM_E	Long term flag for reference picture index [31:0]. Definition: If frame is being decoded the bits 31:15 are used, Bit 31 for picture index 0, Bit 30 for picture index 1 etc… IF field is being decoded the bits 31:0 are used, Bit 31 for reference picture 0 top field, bit 30 for reference picture 0 bottom field etc...

16.1.6.2.35 Reference picture valid flags (H264 VLC) / VPx prediction filter taps (SWREG39)

16.1.6.2.35.1 Offset

Register	Offset
SWREG39	9Ch

16.1.6.2.35.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER_VALID_E															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER_VALID_E															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.2.35.3 Fields

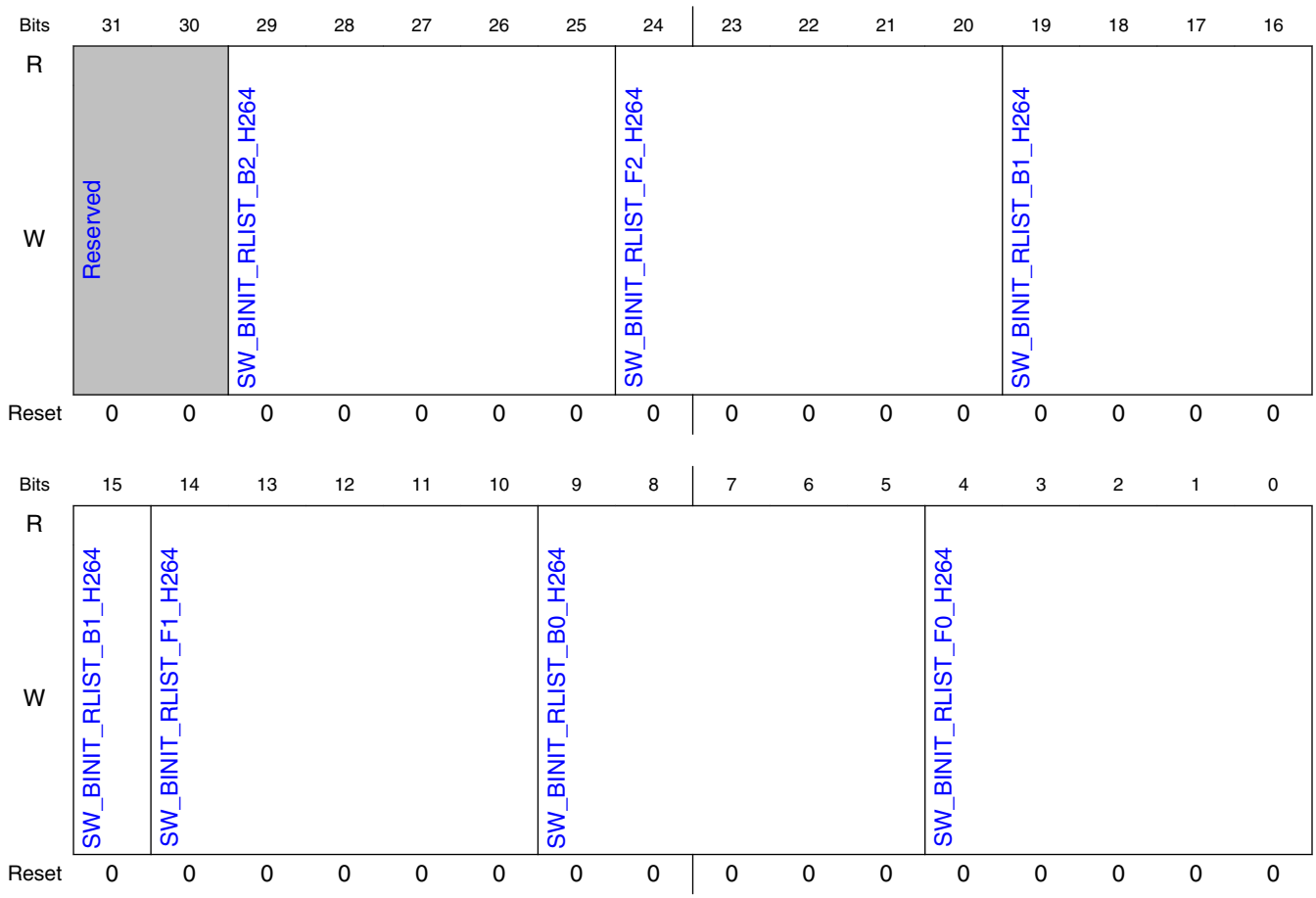
Field	Function
31-0 SW_REFER_VALID_E	Valid flag for reference picture index [31:0].Definition: If frame is being decoded the bits 31:15 are used, Bit 31 for picture index 0, Bit 30 for picture index 1 etc… IF field is being decoded the bits 31:0 are used, Bit 31 for reference picture 0 top field, bit 30 for reference picture 0 bottom field etc...

16.1.6.2.36 bi_dir initial ref pic list register (0-2) / VP6 prediction filter taps / Progressive JPEG Cb ACDC coefficient base (SWREG42_H264)

16.1.6.2.36.1 Offset

Register	Offset
SWREG42_H264	A8h

16.1.6.2.36.2 Diagram



16.1.6.2.36.3 Fields

Field	Function
31-30 —	Reserved.
29-25 SW_BINIT_RLIST_B2_H264	Initial reference picture list for bi-direct backward picid 2
24-20 SW_BINIT_RLIST_F2_H264	Initial reference picture list for bi-direct forward picid 2
19-15 SW_BINIT_RLIST_B1_H264	Initial reference picture list for bi-direct backward picid 1
14-10 SW_BINIT_RLIST_F1_H264	Initial reference picture list for bi-direct forward picid 1

Table continues on the next page...

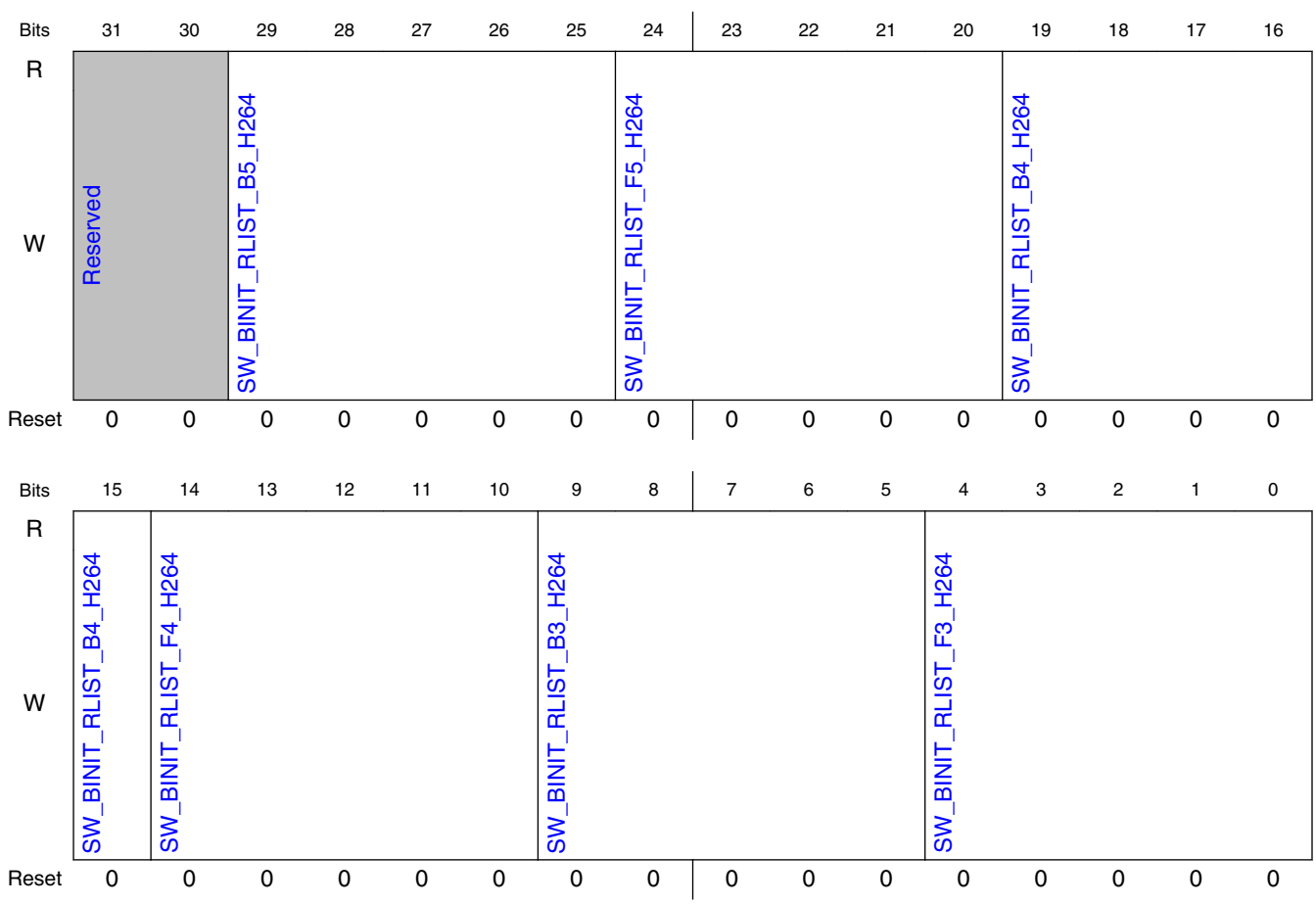
Field	Function
9-5 SW_BINIT_RLIST_B0_H264	Initial reference picture list for bi-direct backward picid 0
4-0 SW_BINIT_RLIST_F0_H264	Initial reference picture list for bi-direct forward picid 0

16.1.6.2.37 bi-dir initial ref pic list register (3-5) / VP6 prediction filter taps / Progressive JPEG Cr ACDC coefficient base (SWREG43_H264)

16.1.6.2.37.1 Offset

Register	Offset
SWREG43_H264	ACh

16.1.6.2.37.2 Diagram



16.1.6.2.37.3 Fields

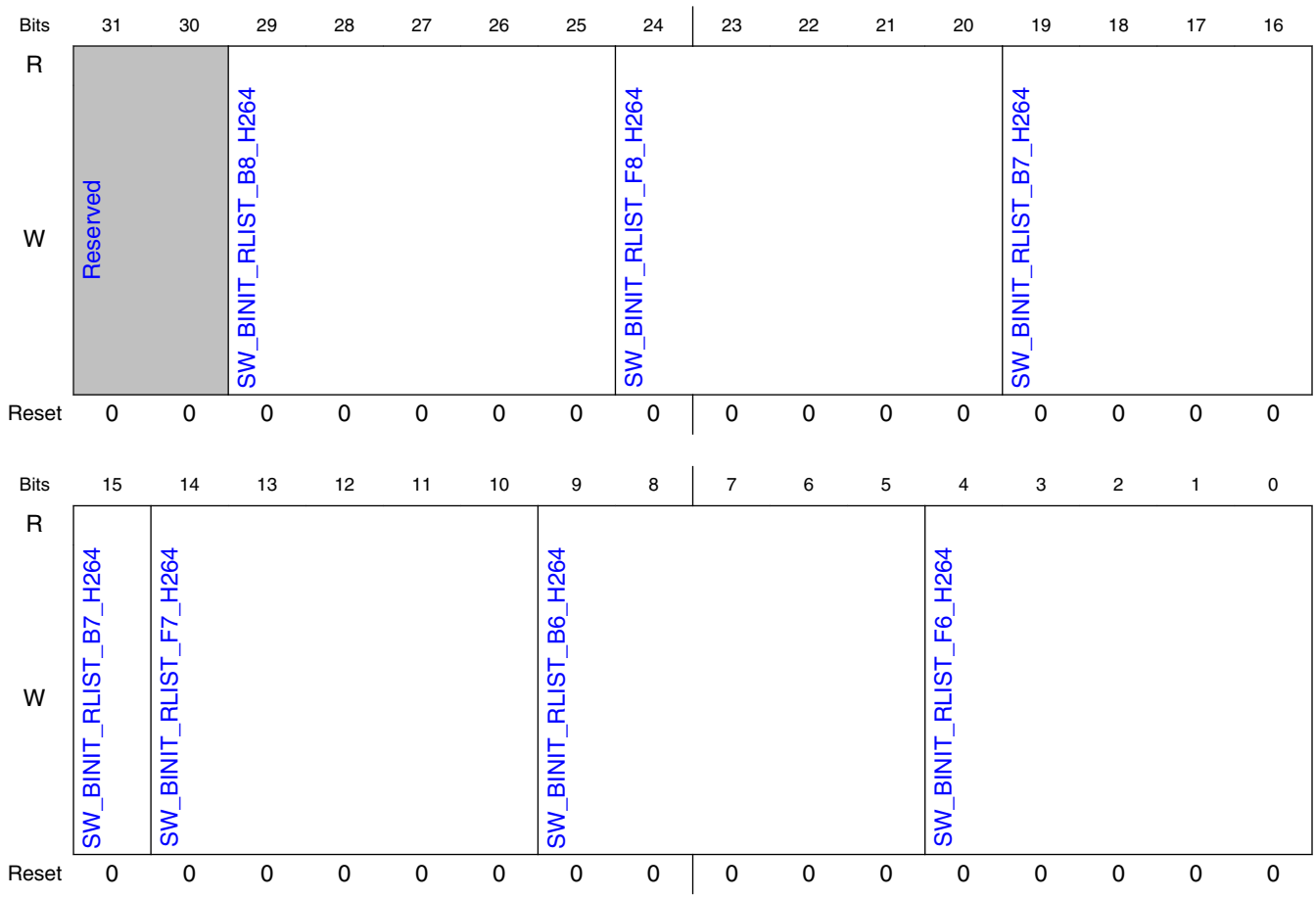
Field	Function
31-30 —	Reserved.
29-25 SW_BINIT_RLIST_B5_H264	Initial reference picture list for bi-direct backward picid 5
24-20 SW_BINIT_RLIST_F5_H264	Initial reference picture list for bi-direct forward picid 5
19-15 SW_BINIT_RLIST_B4_H264	Initial reference picture list for bi-direct backward picid 4
14-10 SW_BINIT_RLIST_F4_H264	Initial reference picture list for bi-direct forward picid 4
9-5 SW_BINIT_RLIST_B3_H264	Initial reference picture list for bi-direct backward picid 3
4-0 SW_BINIT_RLIST_F3_H264	Initial reference picture list for bi-direct forward picid 3

16.1.6.2.38 bi-dir initial ref pic list register (6-8) / VP6 prediction filter taps (SWREG44_H264)

16.1.6.2.38.1 Offset

Register	Offset
SWREG44_H264	B0h

16.1.6.2.38.2 Diagram



16.1.6.2.38.3 Fields

Field	Function
31-30 —	Reserved.
29-25 SW_BINIT_RLIST_B8_H264	Initial reference picture list for bi-direct backward picid 8
24-20 SW_BINIT_RLIST_F8_H264	Initial reference picture list for bi-direct forward picid 8
19-15 SW_BINIT_RLIST_B7_H264	Initial reference picture list for bi-direct backward picid 7
14-10 SW_BINIT_RLIST_F7_H264	Initial reference picture list for bi-direct forward picid 7

Table continues on the next page...

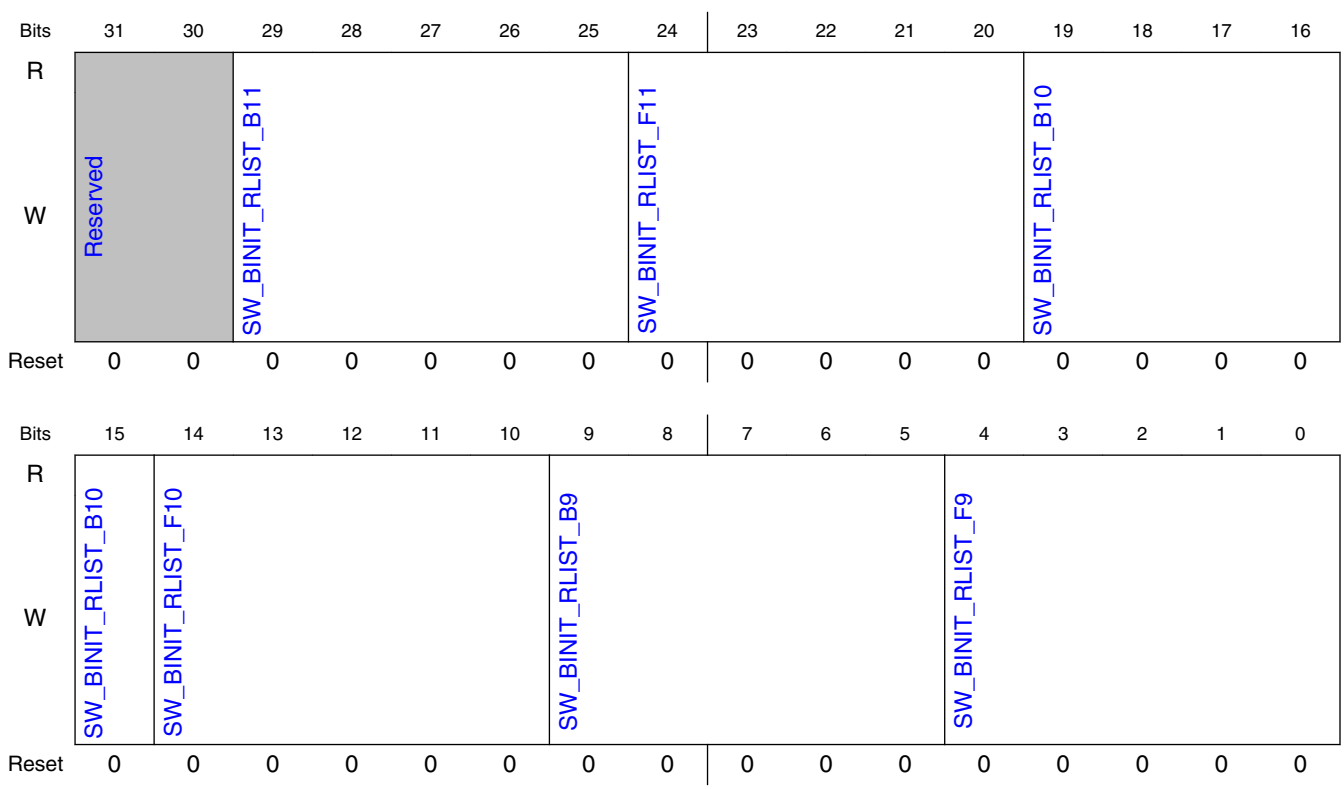
Field	Function
9-5 SW_BINIT_RLIST_B6_H264	Initial reference picture list for bi-direct backward picid 6
4-0 SW_BINIT_RLIST_F6_H264	Initial reference picture list for bi-direct forward picid 6

16.1.6.2.39 bi-dir initial ref pic list register (9-11) / VP6 prediction filter taps (SWREG45)

16.1.6.2.39.1 Offset

Register	Offset
SWREG45	B4h

16.1.6.2.39.2 Diagram



16.1.6.2.39.3 Fields

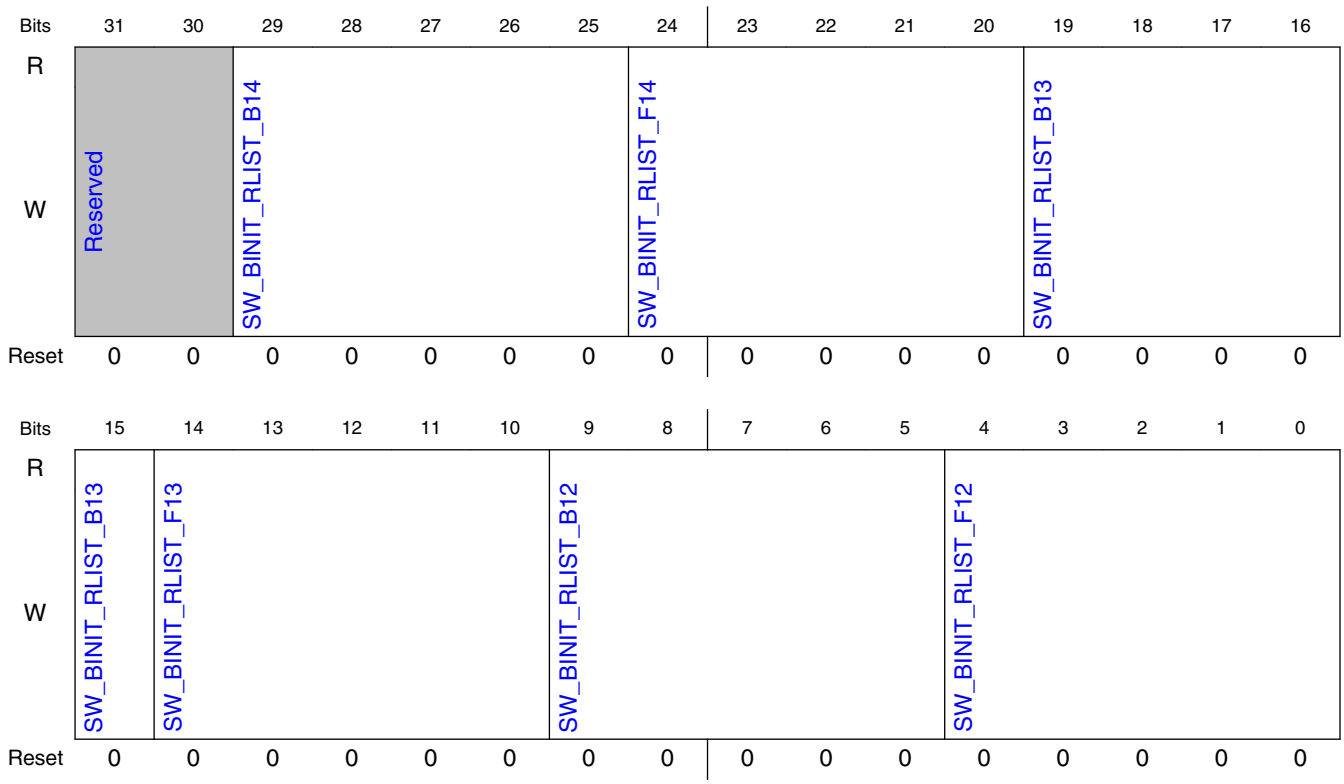
Field	Function
31-30 —	Reserved.
29-25 SW_BINIT_RLIST_B11	Initial reference picture list for bi-direct backward picid 11
24-20 SW_BINIT_RLIST_F11	Initial reference picture list for bi-direct forward picid 11
19-15 SW_BINIT_RLIST_B10	Initial reference picture list for bi-direct backward picid 10
14-10 SW_BINIT_RLIST_F10	Initial reference picture list for bi-direct forward picid 10
9-5 SW_BINIT_RLIST_B9	Initial reference picture list for bi-direct backward picid 9
4-0 SW_BINIT_RLIST_F9	Initial reference picture list for bi-direct forward picid 9

16.1.6.2.40 bi-dir initial ref pic list register (12-14) / VP7,VP8 quantization values (SWREG46)

16.1.6.2.40.1 Offset

Register	Offset
SWREG46	B8h

16.1.6.2.40.2 Diagram



16.1.6.2.40.3 Fields

Field	Function
31-30 —	Reserved.
29-25 SW_BINIT_RLIST_B14	Initial reference picture list for bi-direct backward picid 14
24-20 SW_BINIT_RLIST_F14	Initial reference picture list for bi-direct forward picid 14
19-15 SW_BINIT_RLIST_B13	Initial reference picture list for bi-direct backward picid 13
14-10 SW_BINIT_RLIST_F13	Initial reference picture list for bi-direct forward picid 13
9-5 SW_BINIT_RLIST_B12	Initial reference picture list for bi-direct backward picid 12

Table continues on the next page...

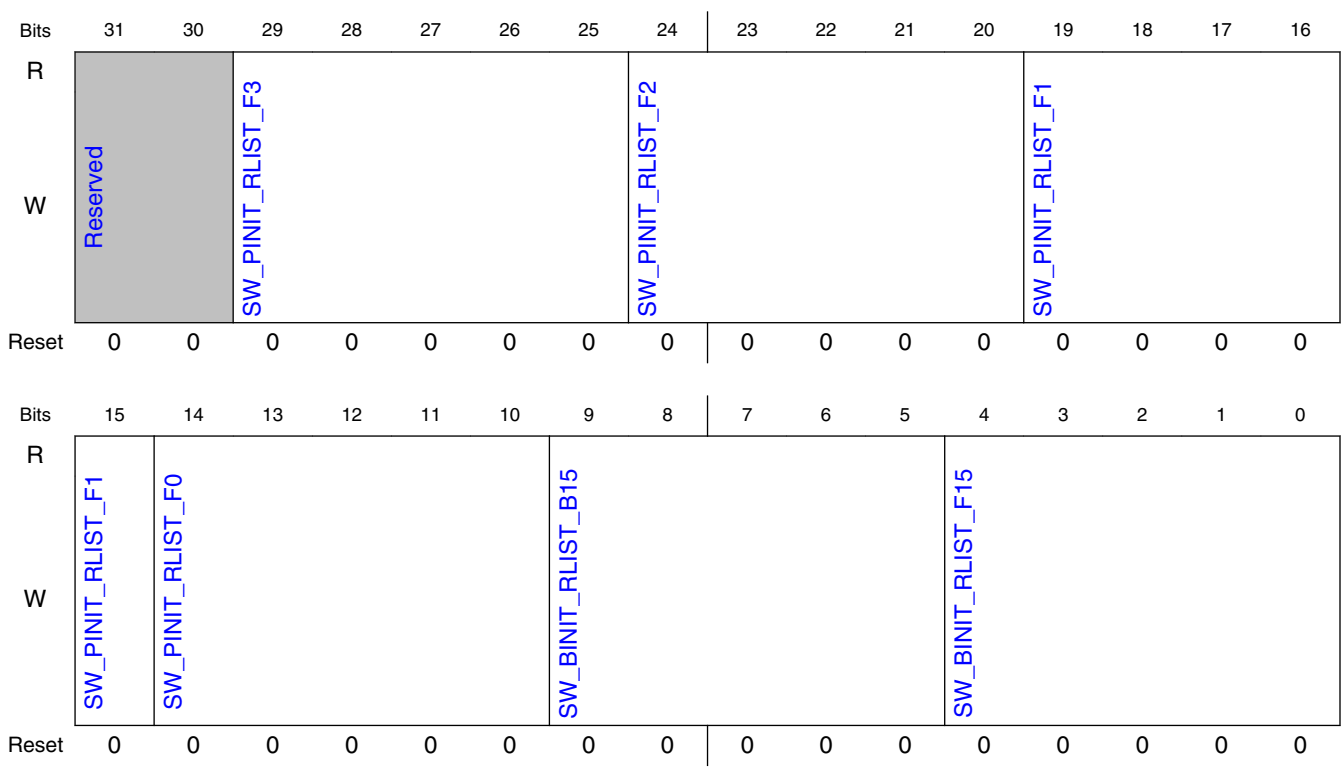
Field	Function
4-0 SW_BINIT_RLIST_F12	Initial reference picture list for bi-direct forward picid 12

16.1.6.2.41 bi-dir and P fwd initial ref pic list register (15 and P 0-3) / VP7,VP8 quantization values (SWREG47)

16.1.6.2.41.1 Offset

Register	Offset
SWREG47	BCh

16.1.6.2.41.2 Diagram



16.1.6.2.41.3 Fields

Field	Function
31-30	Reserved.

Table continues on the next page...

Field	Function
—	
29-25 SW_PINIT_RLIST_F3	Initial reference picture list for P forward picid 3
24-20 SW_PINIT_RLIST_F2	Initial reference picture list for P forward picid 2
19-15 SW_PINIT_RLIST_F1	Initial reference picture list for P forward picid 1
14-10 SW_PINIT_RLIST_F0	Initial reference picture list for P forward picid 0
9-5 SW_BINIT_RLIST_B15	Initial reference picture list for bi-direct backward picid 15
4-0 SW_BINIT_RLIST_F15	Initial reference picture list for bi-direct forward picid 15

16.1.6.3 VPU_G1 VP8 decoder register descriptions

16.1.6.3.1 VPU_G1 VP8 decoder memory map

VPU_G1_VP8 base address: 3830_0000h

Offset	Register	Width (In bits)	Access	Reset value
4Ch	Base address for reference picture index 5 / MPEG4 TRB/TRD delta 0 / VC-1 intensity control 3 List of VLC code lengths in first/second JPEG AC table / VP6/VP7 scan maps (SWREG19_VP8)	32	RW	0000_0000h
50h	Base address for reference picture index 6 / / MPEG4 TRB/TRD delta -1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG20_VP8)	32	RW	0000_0000h
54h	Base address for reference picture index 7 / MPEG4 TRB/TRD delta 1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG21_VP8)	32	RW	0000_0000h

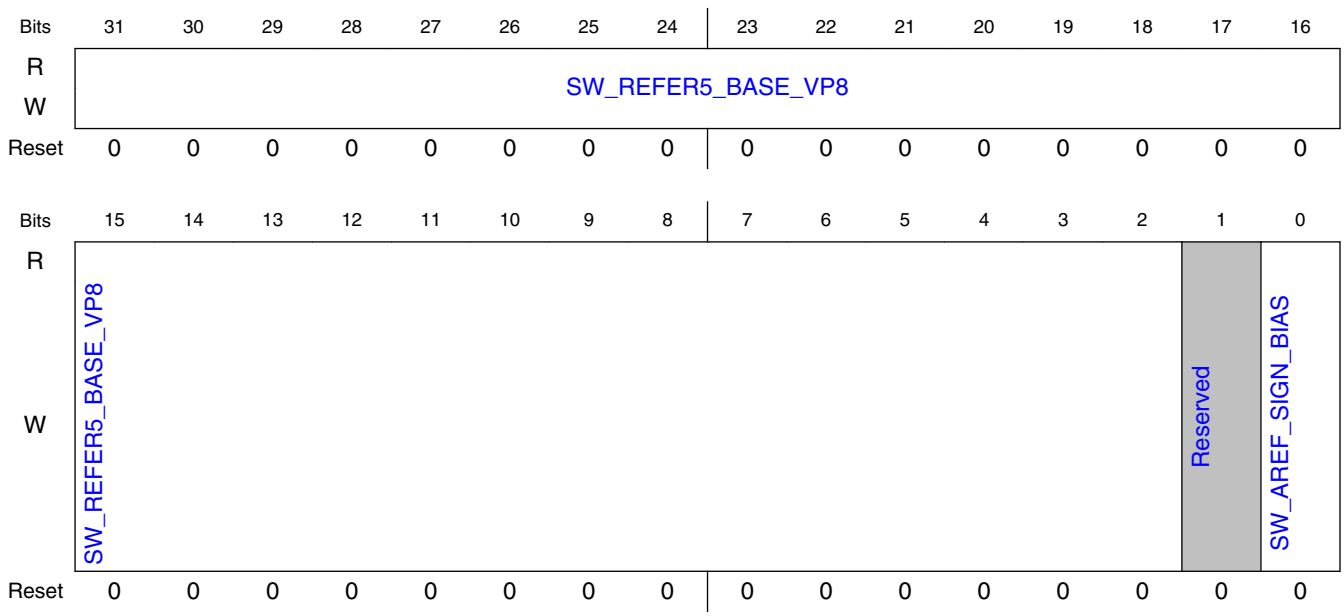
16.1.6.3.2 Base address for reference picture index 5 / MPEG4 TRB/TRD delta 0 / VC-1 intensity control 3 List of VLC code lengths in

first/second JPEG AC table / VP6/VP7 scan maps (SWREG19_VP8)

16.1.6.3.2.1 Offset

Register	Offset
SWREG19_VP8	4Ch

16.1.6.3.2.2 Diagram



16.1.6.3.2.3 Fields

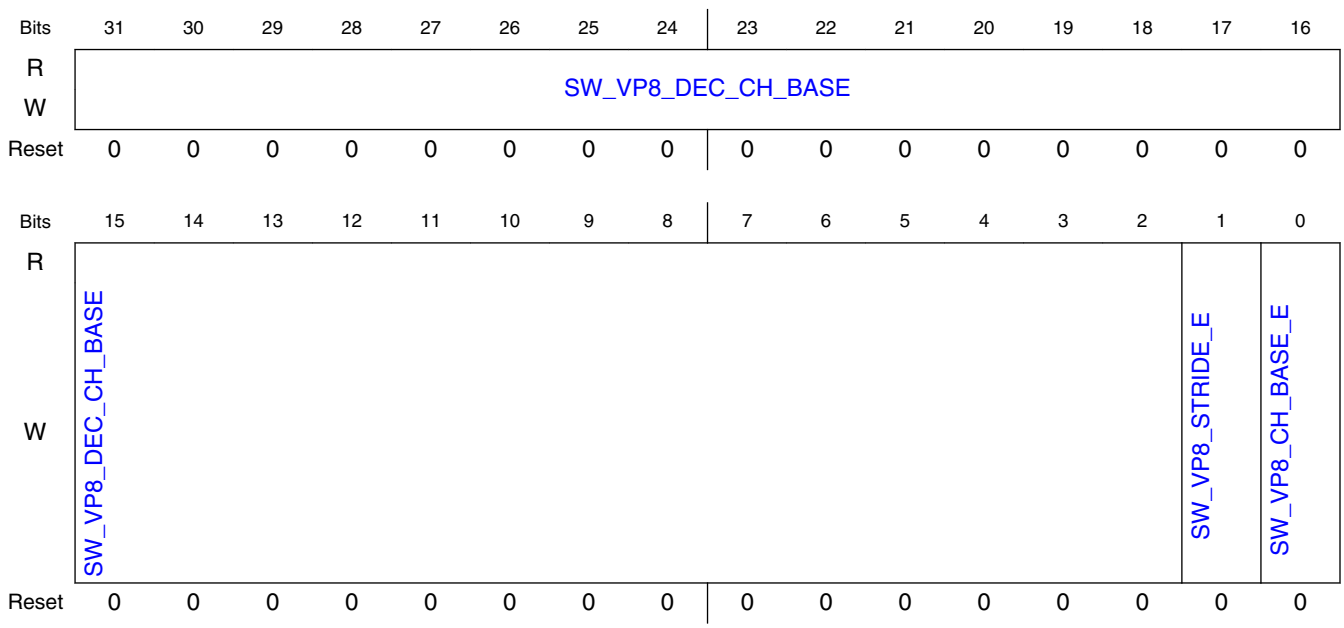
Field	Function
31-2 SW_REFER5_BASE_VP8	H.264: Base address for reference picture index 5 VP8: Base address for alternate reference picture (corresponds picid 5)
1 —	Reserved.
0 SW_AREF_SIGN_BIAS	VP8 only: Reference picture sign bias for Alternate reference frame

16.1.6.3.3 Base address for reference picture index 6 / / MPEG4 TRB/TRD delta -1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG20_VP8)

16.1.6.3.3.1 Offset

Register	Offset
SWREG20_VP8	50h

16.1.6.3.3.2 Diagram



16.1.6.3.3.3 Fields

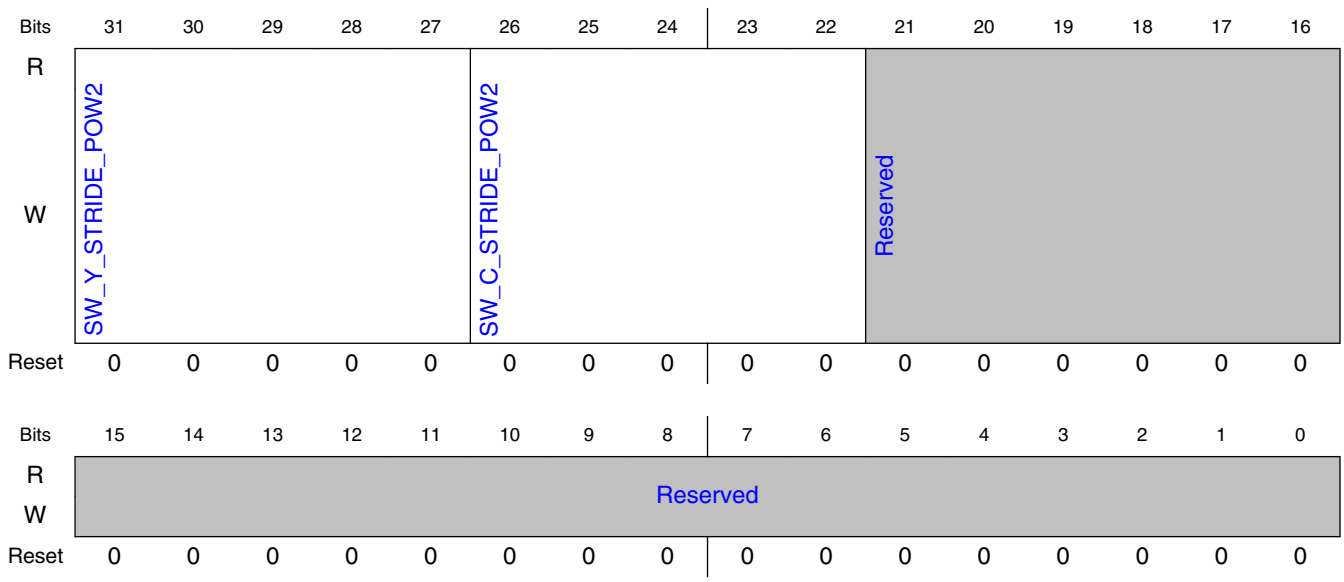
Field	Function
31-2 SW_VP8_DEC_CH_BASE	VP8 video base address for decoder output chrominance data (if vp8 stride configuration is enabled)
1 SW_VP8_STRIDE_E	VP8 stride enable. Can be set high only if HW configuration supports strides. Y and C strides are used instead of picture width. Separate chrominance base addresses are used instead of internal chrominance offsets. 0b - Not enabled 1b - Enabled
0 SW_VP8_CH_BASE_E	VP8 separate chrominance enable: 0b - Write/Read chrominance data from internal offset after the luminance data 1b - Write/Read chrominance data from separate base addresses given by SW

16.1.6.3.4 Base address for reference picture index 7 / MPEG4 TRB/TRD delta 1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG21_VP8)

16.1.6.3.4.1 Offset

Register	Offset
SWREG21_VP8	54h

16.1.6.3.4.2 Diagram



16.1.6.3.4.3 Fields

Field	Function
31-27 SW_Y_STRIDE_POW2	VP8 Y stride length informed by 2^n (n=sw_y_stride_pow2). Valid range 10-17 for 32 bit bus and 10-18 for 64 bit bus
26-22 SW_C_STRIDE_POW2	VP8 C stride length informed by 2^n (n=sw_c_stride_pow2). Valid range 10-17 for 32 bit bus and 10-18 for 64 bit bus
21-0 —	Reserved.

16.1.6.4 VPU_G1 VP7/VP8 decoder register descriptions

16.1.6.4.1 VPU_G1 VP7/VP8 decoder memory map

VPU_G1_VP7_VP8 base address: 3830_0000h

Offset	Register	Width (In bits)	Access	Reset value
10h	Decoder control register 1 (picture parameters) (SWREG4_JPEG_VP7_VP8)	32	RW	0000_0000h
14h	Decoder control register 2 (stream decoding table selects) (SWREG5_VP7_VP8)	32	RW	0000_0000h
18h	Decoder control register 3 (stream buffer information) (SWREG6_VP7_VP8)	32	RW	0000_0000h
1Ch	Decoder control register 4 (H264, VC-1, VP6 and progressive JPEG control) (SWREG7_VP7_VP8)	32	RW	0000_0000h
28h	Base address for differential motion vector base address (RLC-mode) /H264 P initial fwd ref pic list register (4-9)/ VC-1 intensity control 1/ VP7 and VP8 segmentation base register (SWREG10_VP7_VP8)	32	RW	0000_0000h
2Ch	Decoder control register 7 (VLC) / base address for H.264 intra prediction 4x4 / base address for MPEG-4 DC component (RLC) / H264 P initial fwd ref pic list register (10-15) / VC-1 intensity control 2 (SWREG11_VP7_VP8)	32	RW	0000_0000h
38h	Base address for reference picture index 0 / base address for JPEG decoder output chrominance picture (SWREG14_VP7_VP8)	32	RW	0000_0000h
3Ch	Base address for reference picture index 1 / JPEG control (SWREG15_VP7_VP8)	32	RW	0000_0000h
48h	Base address for reference picture index 4 / VC1 control / MPEG4 MVD control/ List of VLC code lengths in first JPEG AC table / VC-1 intensity control 4 / VP6/VP7, VP8 Golden refer picture base (SWREG18_VP7_VP8)	32	RW	0000_0000h
58h	Base address for reference picture index 8 / List of VLC code lengths in second JPEG AC table / VP6 scan maps / VP7,VP8 DCT stream 1 base (SWREG22_VP7_VP8)	32	RW	0000_0000h
5Ch	Base address for reference picture index 9 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 2 base (SWREG23_VP7_VP8)	32	RW	0000_0000h
60h	Base address for reference picture index 10 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 3 base (SWREG24_VP7_VP8)	32	RW	0000_0000h
64h	Base address for reference picture index 11 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 4 base (SWREG25_VP7_VP8)	32	RW	0000_0000h

Table continues on the next page...

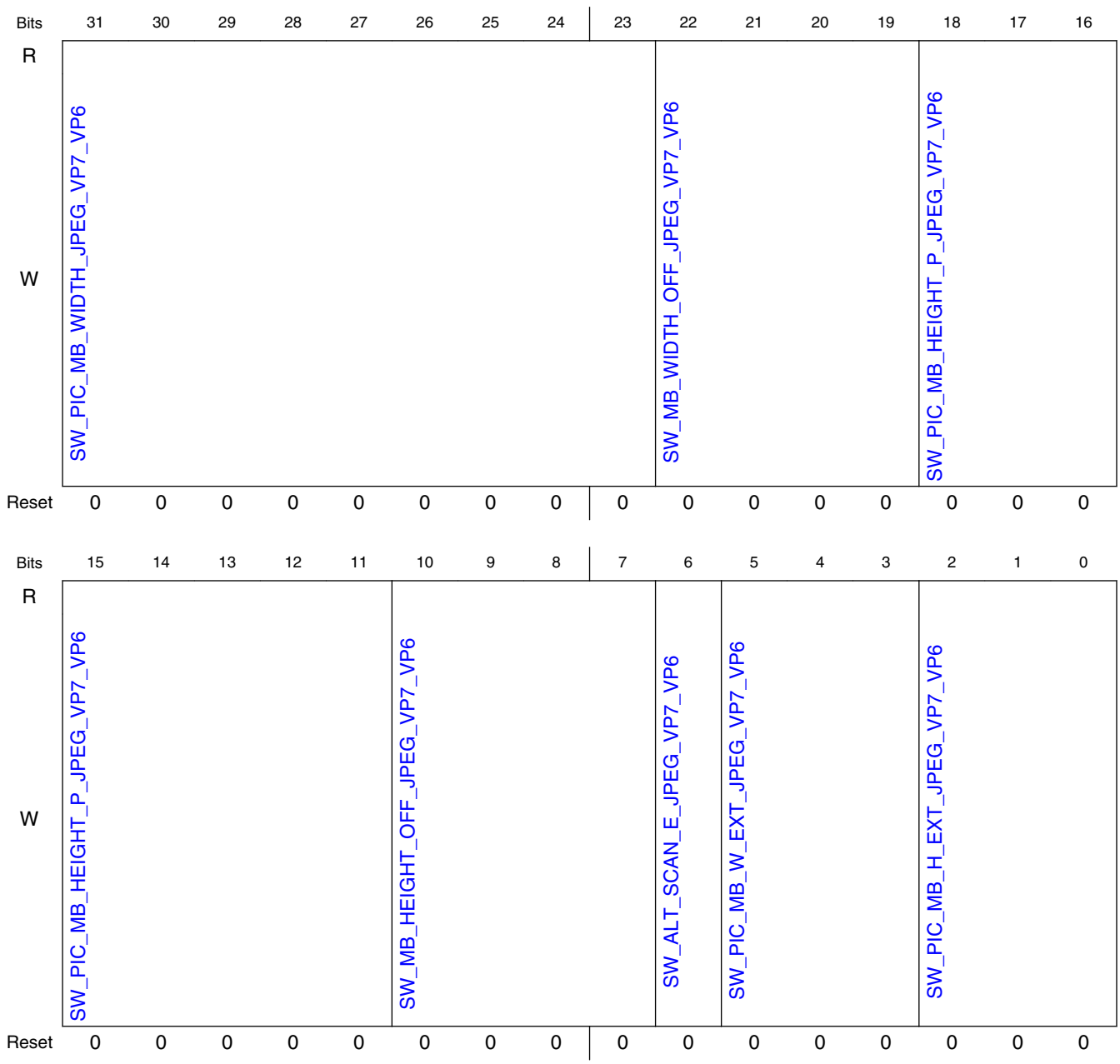
Offset	Register	Width (In bits)	Access	Reset value
68h	Base address for reference picture index 12 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 5 base (SWREG26_VP7_VP8)	32	RW	0000_0000h
6Ch	Base address for reference picture index 13 / VC-1 bitpl mbctrl or VP6,VP7,VP8 ctrl stream base /Progressive JPEG DC table (SWREG27_VC1)	32	RW	0000_0000h
70h	Base address for reference picture index 14 / VP6 scan maps / Progressive JPEG DC table / VP7,VP8 DCT stream 6 base (SWREG28_VP7_VP8)	32	RW	0000_0000h
74h	Base address for reference picture index 15 / VP6 scan maps / VP7,VP8 DCT stream 7 base (SWREG29_VP7_VP8)	32	RW	0000_0000h
78h	Reference picture numbers for index 0 and 1 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter mb level adjusts (SWREG30_VP7_VP8)	32	RW	0000_0000h
7Ch	Reference picture numbers for index 2 and 3 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter ref pic level adjusts (SWREG31_VP7_VP8)	32	RW	0000_0000h
80h	Reference picture numbers for index 4 and 5 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter levels (SWREG32_VP7_VP8)	32	RW	0000_0000h
84h	Reference picture numbers for index 6 and 7 (H264 VLC) / VP6 scan maps / VP7,VP8 quantization values (SWREG33_VP7_VP8)	32	RW	0000_0000h
88h	Reference picture numbers for index 8 and 9 (H264 VLC) / MPEG4, VC1, VPx prediction filter taps (SWREG34_H263)	32	RW	0000_0000h
8Ch	Reference picture numbers for index 10 and 11 (H264 VLC) / VC1, VPx prediction filter taps (SWREG35_VC1)	32	RW	0000_0000h
90h	Reference picture numbers for index 12 and 13 (H264 VLC) / VC1, VPx prediction filter taps (SWREG36_VC1)	32	RW	0000_0000h
94h	Reference picture numbers for index 14 and 15 (H264 VLC) / VPx prediction filter taps (SWREG37_VP6_VP7_VP8)	32	RW	0000_0000h
98h	Reference picture long term flags (H264 VLC) / VPx prediction filter taps (SWREG38_VP6_VP7_VP8)	32	RW	0000_0000h
9Ch	Reference picture valid flags (H264 VLC) / VPx prediction filter taps (SWREG39_VP6_VP7_VP8)	32	RW	0000_0000h
A8h	bi_dir initial ref pic list register (0-2) / VP6 prediction filter taps / Progressive JPEG Cb ACDC coefficient base (SWREG42_VP6)	32	RW	0000_0000h
ACh	bi-dir initial ref pic list register (3-5) / VP6 prediction filter taps / Progressive JPEG Cr ACDC coefficient base (SWREG43_VP7_VP8)	32	RW	0000_0000h
B0h	bi-dir initial ref pic list register (6-8) / VP6 prediction filter taps (SWREG44_VP7_VP8)	32	RW	0000_0000h
B4h	bi-dir initial ref pic list register (9-11) / VP6 prediction filter taps (SWREG45_VP7_VP8)	32	RW	0000_0000h
B8h	bi-dir initial ref pic list register (12-14) / VP7,VP8 quantization values (SWREG46_VP7_VP8)	32	RW	0000_0000h
BCh	bi-dir and P fwd initial ref pic list register (15 and P 0-3) / VP7,VP8 quantization values (SWREG47_VP7_VP8)	32	RW	0000_0000h

16.1.6.4.2 Decoder control register 1 (picture parameters) (SWREG4_JPEG_VP7_VP8)

16.1.6.4.2.1 Offset

Register	Offset
SWREG4_JPEG_VP7_VP8	10h

16.1.6.4.2.2 Diagram



16.1.6.4.2.3 Fields

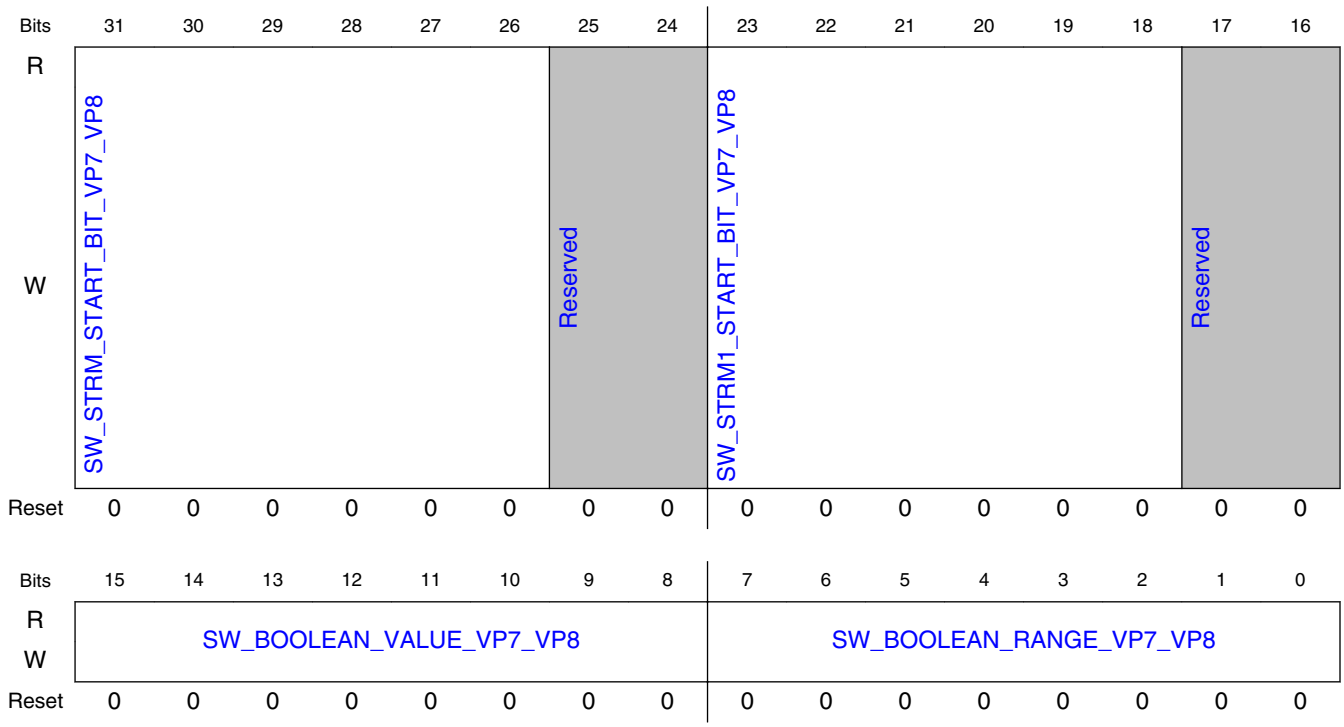
Field	Function
31-23 SW_PIC_MB_WIDTH_JPEG_VP7_VP6	Picture width in macroblocks = ((width in pixels + 15) /16)
22-19 SW_MB_WIDTH_OFF_JPEG_VP7_VP6	The amount of meaningfull horizontal pixels in last MB (width offset) 0 if exactly 16 pixels multiple picture and all the horizontal pixels in last MB are meaningfull
18-11 SW_PIC_MB_HEIGHT_P_JPEG_VP7_VP6	Picture height in macroblocks =((height in pixels+15)/16). Picture height is informed as size of the (progressive) frame also for single field (of interlaced content) is being decoded
10-7 SW_MB_HEIGHT_OFF_JPEG_VP7_VP6	The amount of meaningful vertical pixels in last MB (height offset 0 if exactly 16 pixels multiple picture and all the vertical pixels in last MB are meaningfull
6 SW_ALT_SCAN_E_JPEG_VP7_VP6	indicates alternative vertical scan method used for interlaced frames
5-3 SW_PIC_MB_W_EXT_JPEG_VP7_VP6	Picture mb width extension. If sw_pic_mb_width does not fit to 9 bits then these bits are used to increase the range upto 11 bits (used as 3 msb)
2-0 SW_PIC_MB_H_EXT_JPEG_VP7_VP6	Picture mb height extension. If sw_pic_mb_height_p does not fit to 9 bits then these bits are used to increase the range upto 11 bits (used as 3 msb). For 4k video one bit is used for extension (bit 0)

16.1.6.4.3 Decoder control register 2 (stream decoding table selects) (SWREG5_VP7_VP8)

16.1.6.4.3.1 Offset

Register	Offset
SWREG5_VP7_VP8	14h

16.1.6.4.3.2 Diagram



16.1.6.4.3.3 Fields

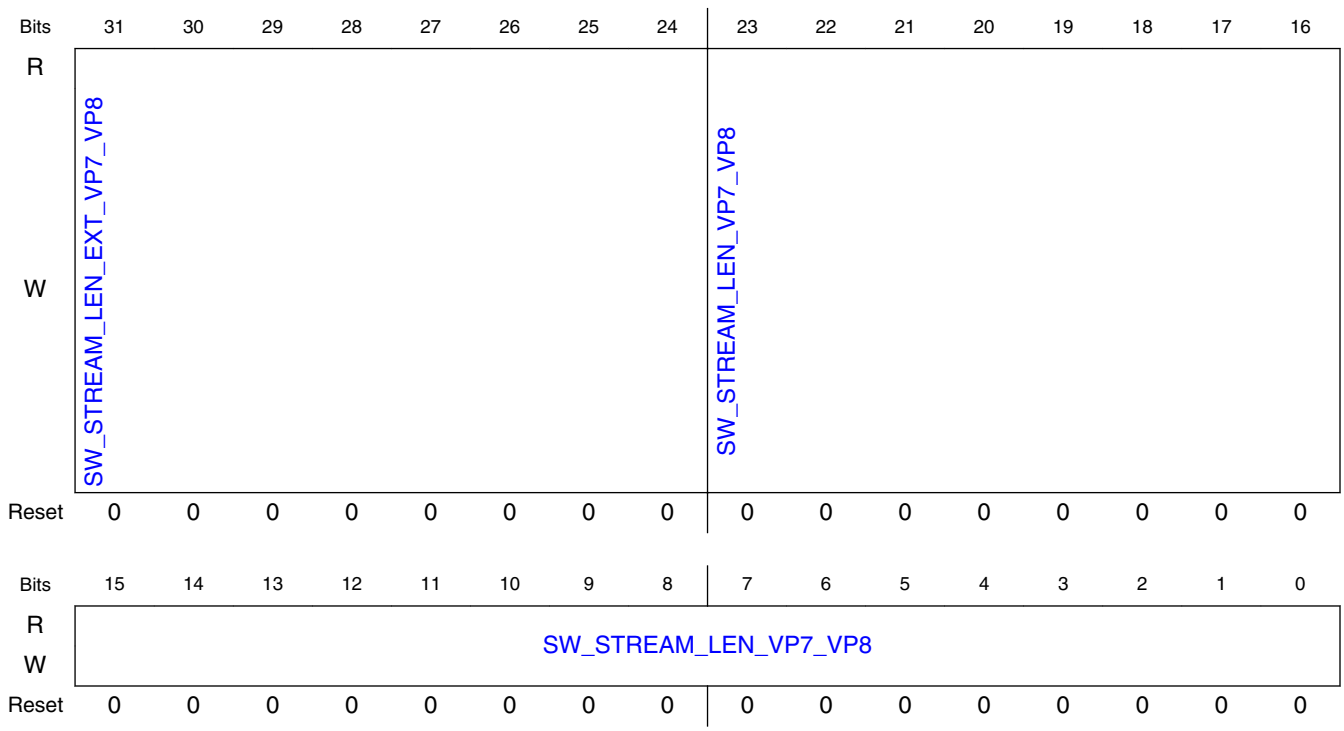
Field	Function
31-26 SW_STRM_START_BIT_VP7_VP8	Exact bit of stream start word where decoding can be started (assosiates with sw_rlc_vlc_base)
25-24 —	Reserved.
23-18 SW_STRM1_START_BIT_VP7_VP8	Start bit for ctrl-stream (needed if multistream is enabled, assosiates with sw_bitpl_ctrl_base)
17-16 —	Reserved.
15-8 SW_BOOLEAN_VALUE_VP7_VP8	Initial value for boolean dec
7-0 SW_BOOLEAN_RANGE_VP7_VP8	Initial range for boolean dec

16.1.6.4.4 Decoder control register 3 (stream buffer information) (SWREG6_VP7_VP8)

16.1.6.4.4.1 Offset

Register	Offset
SWREG6_VP7_VP8	18h

16.1.6.4.4.2 Diagram



16.1.6.4.4.3 Fields

Field	Function
31-24 SW_STREAM_LEN_EXT_VP7_VP8	Extended stream length for WEBP/VP8
23-0	Amount of stream data bytes in input buffer. If the given buffer size is not enough for finishing the picture the corresponding interrupt is given and new stream buffer base address and stream buffer size information should be given (assosiates with sw_rlc_vlc_base). For VC-1/VP6 the buffer must include

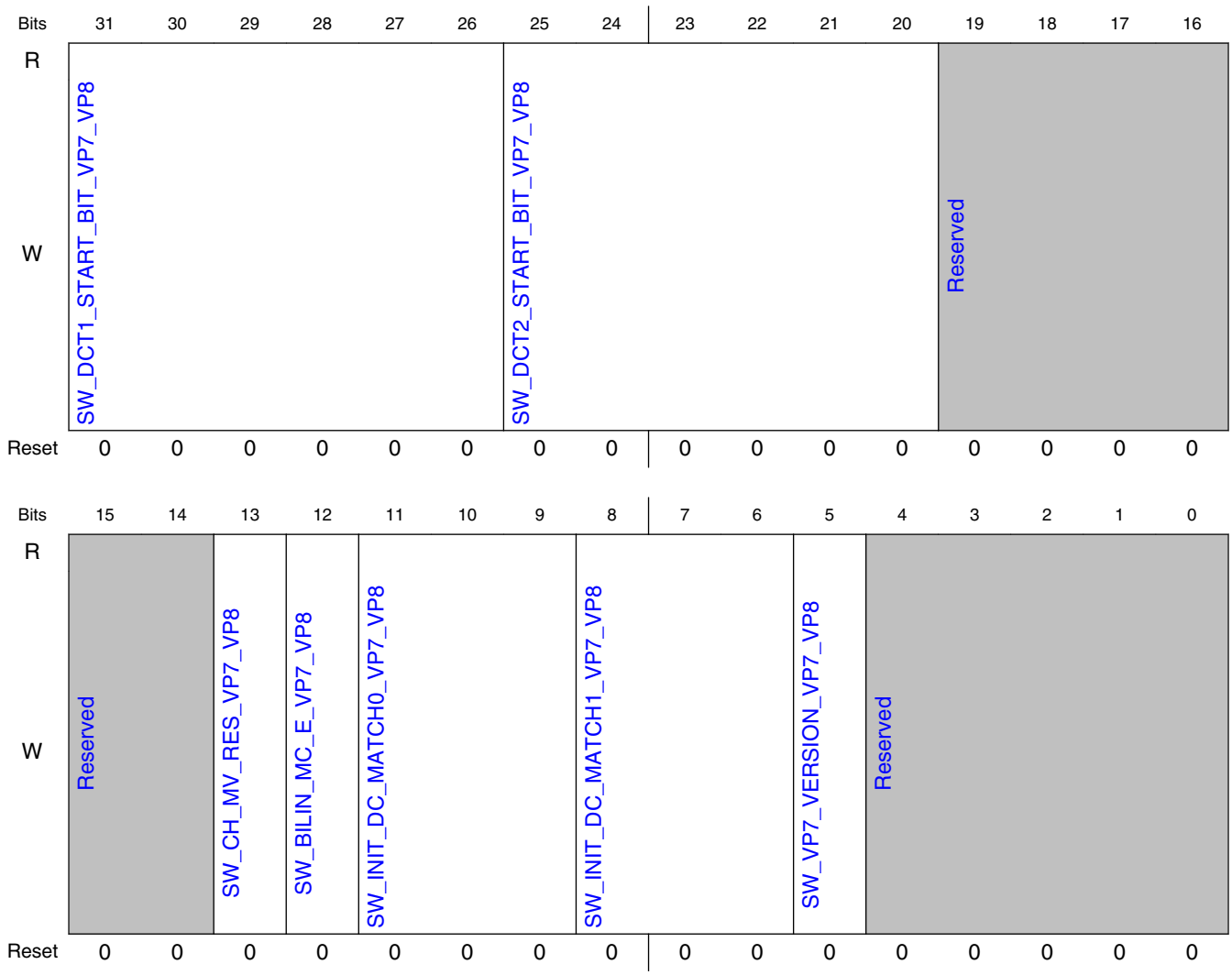
Field	Function
SW_STREAM_LEN_VP7_VP8	data for one picture/slice of the picture For H264/MPEG4/H263/MPEG2/MPEG1 the buffer must include at least data for one slice/VP of the picture For JPEG the buffer size must be a multiple of 256 bytes or the amount of data for one picture.

16.1.6.4.5 Decoder control register 4 (H264, VC-1, VP6 and progressive JPEG control) (SWREG7_VP7_VP8)

16.1.6.4.5.1 Offset

Register	Offset
SWREG7_VP7_VP8	1Ch

16.1.6.4.5.2 Diagram



16.1.6.4.5.3 Fields

Field	Function
31-26 SW_DCT1_START_BIT_VP7_VP8	Start bit for VP7/VP8 DCT stream partition index 1
25-20 SW_DCT2_START_BIT_VP7_VP8	Start bit for VP7/VP8 DCT stream partition index 2
19-14 —	Reserved.

Table continues on the next page...

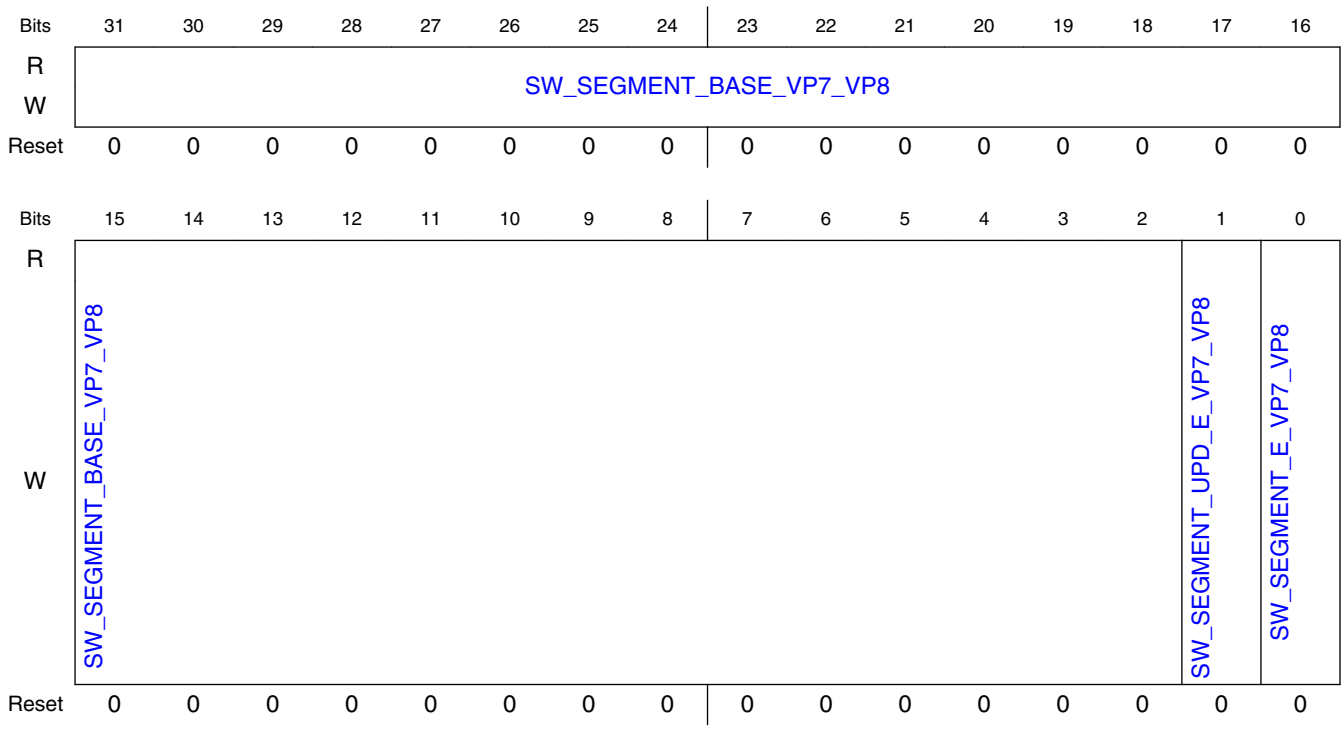
Field	Function
13 SW_CH_MV_RES_VP7_VP8	VP7/VP8 Chrominance motion vector resolution: 0b - Full pixel 1b - 1/8 pixel
12 SW_BILIN_MC_E_VP7_VP8	Bilinear motion compensation enable: 0b - Bicubic interpolation used 1b - Bilinear interpolation used
11-9 SW_INIT_DC_MATCH0_VP7_VP8	Initial DC prediction mach count 0. After HW has decoded a picture HW returns the final match count0 information which is read by SW
8-6 SW_INIT_DC_MATCH1_VP7_VP8	Initial DC prediction mach count 1. After HW has decoded a picture HW returns the final match count1 information which is read by SW
5 SW_VP7_VERSION_VP7_VP8	VP7 version information to streamd: 0b - VP7 version 7.0 1b - VP7 version 7.1 or better
4-0 —	Reserved.

16.1.6.4.6 Base address for differential motion vector base address (RLC-mode) /H264 P initial fwd ref pic list register (4-9)/ VC-1 intensity control 1/ VP7 and VP8 segmentation base register (SWREG10_VP7_VP8)

16.1.6.4.6.1 Offset

Register	Offset
SWREG10_VP7_VP8	28h

16.1.6.4.6.2 Diagram



16.1.6.4.6.3 Fields

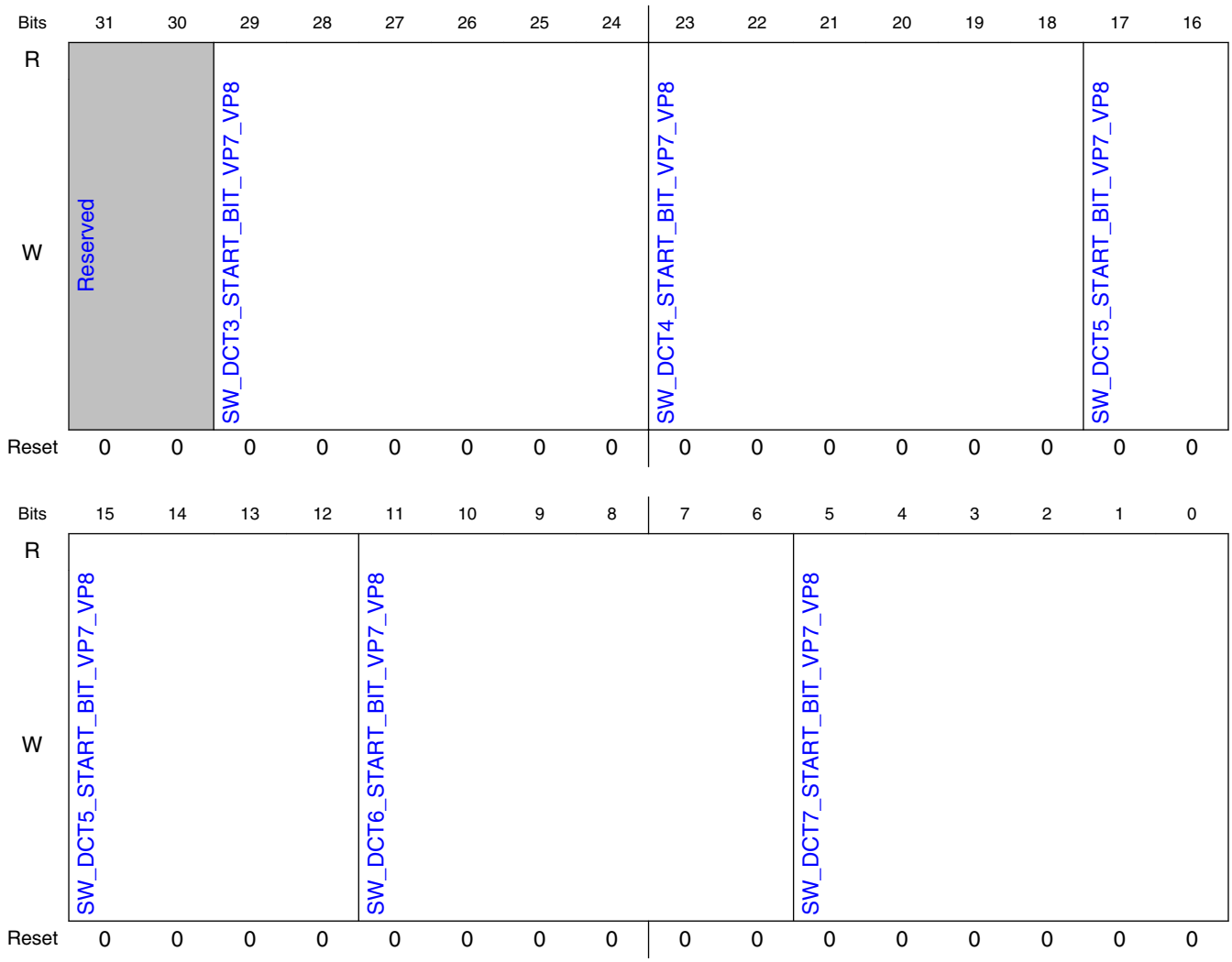
Field	Function
31-2 SW_SEGMENT_BASE_VP7_VP8	VP7/VP8: base address for segmentation map values
1 SW_SEGMENT_UPD_E_VP7_VP8	VP7/VP8 Segmentation map update enable: '0': segmentation values are read from external memory (from segment_base) '1': segmentation update is included in stream
0 SW_SEGMENT_E_VP7_VP8	Segmentation enable: '0': segmentation is not enabled '1': segmentation is enabled (sw_segment_upd_e value is used)

16.1.6.4.7 Decoder control register 7 (VLC) / base address for H.264 intra prediction 4x4 / base address for MPEG-4 DC component (RLC) / H264 P initial fwd ref pic list register (10-15) / VC-1 intensity control 2 (SWREG11_VP7_VP8)

16.1.6.4.7.1 Offset

Register	Offset
SWREG11_VP7_VP8	2Ch

16.1.6.4.7.2 Diagram



16.1.6.4.7.3 Fields

Field	Function
31-30 —	Reserved.
29-24	Start bit for VP7/VP8 DCT stream partition index 3

Table continues on the next page...

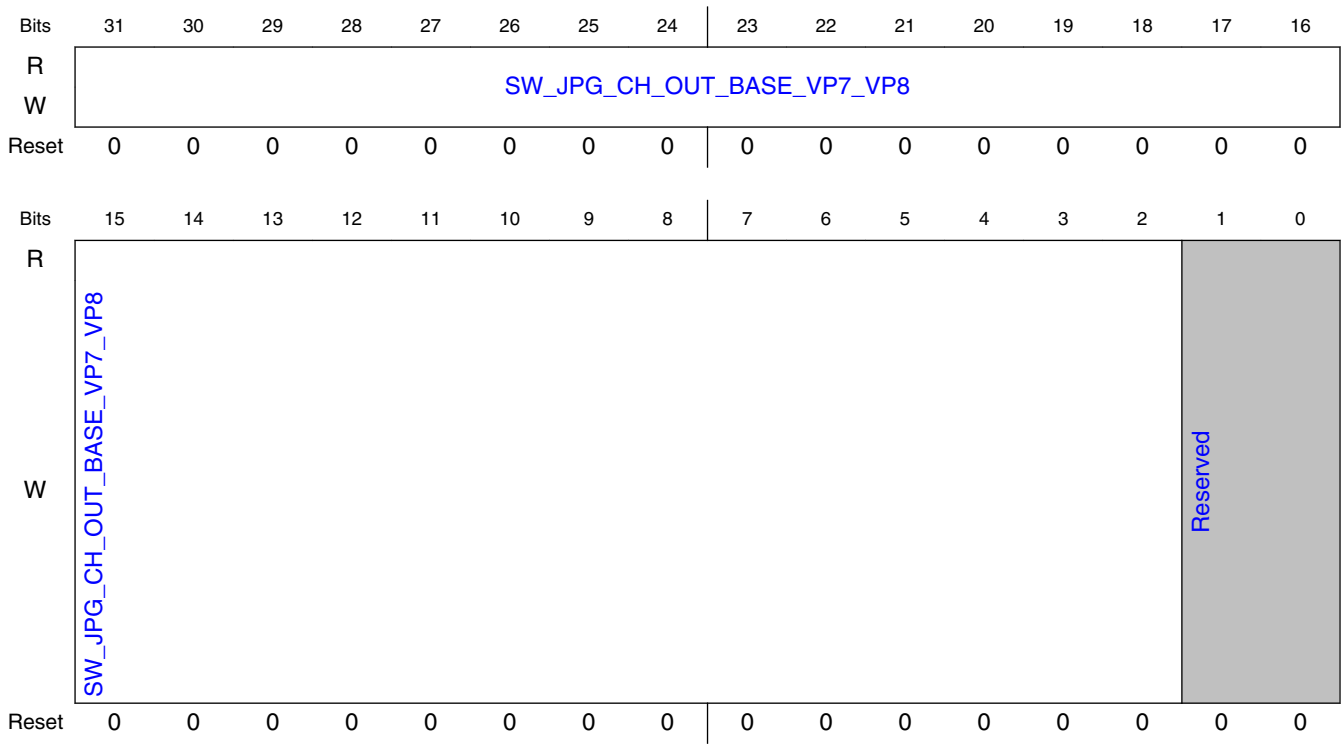
Field	Function
SW_DCT3_START_BIT_VP7_VP8	
23-18 SW_DCT4_START_BIT_VP7_VP8	Start bit for VP7/VP8 DCT stream partition index 4
17-12 SW_DCT5_START_BIT_VP7_VP8	Start bit for VP7/VP8 DCT stream partition index 5
11-6 SW_DCT6_START_BIT_VP7_VP8	Start bit for VP7/VP8 DCT stream partition index 6
5-0 SW_DCT7_START_BIT_VP7_VP8	Start bit for VP7/VP8 DCT stream partition index 7

16.1.6.4.8 Base address for reference picture index 0 / base address for JPEG decoder output chrominance picture (SWREG14_VP7_VP8)

16.1.6.4.8.1 Offset

Register	Offset
SWREG14_VP7_VP8	38h

16.1.6.4.8.2 Diagram



16.1.6.4.8.3 Fields

Field	Function
31-2 SW_JPG_CH_OUT_BASE_VP7_VP8	Base address for decoder output chrominance picture. Used in JPEG and web-p picture mode (not needed if decoder output is not written)
1-0 —	Reserved.

16.1.6.4.9 Base address for reference picture index 1 / JPEG control (SWREG15_VP7_VP8)

16.1.6.4.9.1 Offset

Register	Offset
SWREG15_VP7_VP8	3Ch

16.1.6.4.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								SW_JPEG_SLICE_H_VP7_VP8							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.1.6.4.9.3 Fields

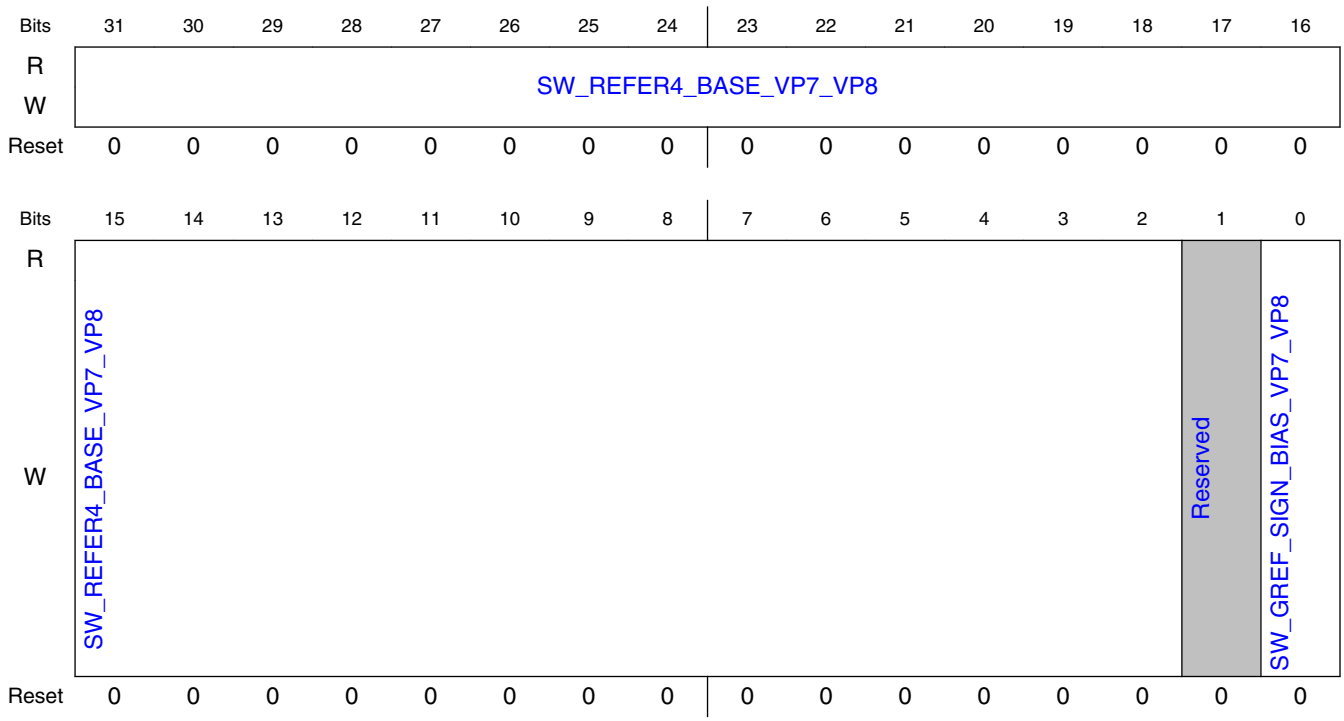
Field	Function
31-8 —	Reserved.
7-0 SW_JPEG_SLICE_H_VP7_VP8	JPEG/Web-p. Height of the slice (multiple of 16 pixels) that HW decodes before interrupt. When slice is decoded HW will rise an interrupt and reset external addresses back to base address. Note, value 0 disables slice mode. Slice mode must be used if picture size is more than 16 Mpixels. However for bigger than 4096 MBs the slice mode usage is recommended.

16.1.6.4.10 Base address for reference picture index 4 / VC1 control / MPEG4 MVD control/ List of VLC code lengths in first JPEG AC table / VC-1 intensity control 4 / VP6/VP7, VP8 Golden refer picture base (SWREG18_VP7_VP8)

16.1.6.4.10.1 Offset

Register	Offset
SWREG18_VP7_VP8	48h

16.1.6.4.10.2 Diagram



16.1.6.4.10.3 Fields

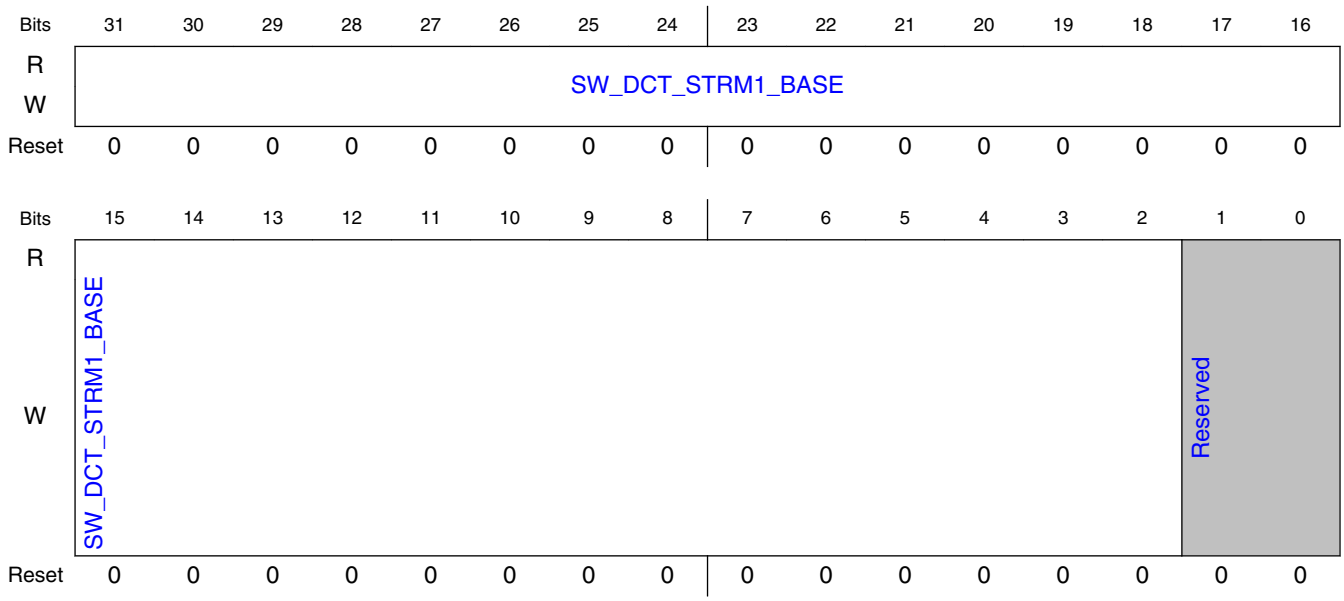
Field	Function
31-2 SW_REFER4_BASE_VP7_VP8	H264: Base address for reference picture index 4 VP6/VP7/VP8: Base address for Golden reference picture (corresponds picid 4)
1 —	Reserved.
0 SW_GREF_SIGN_BIAS_VP7_VP8	Reference picture sign bias for Golden reference frame

16.1.6.4.11 Base address for reference picture index 8 / List of VLC code lengths in second JPEG AC table / VP6 scan maps / VP7,VP8 DCT stream 1 base (SWREG22_VP7_VP8)

16.1.6.4.11.1 Offset

Register	Offset
SWREG22_VP7_VP8	58h

16.1.6.4.11.2 Diagram



16.1.6.4.11.3 Fields

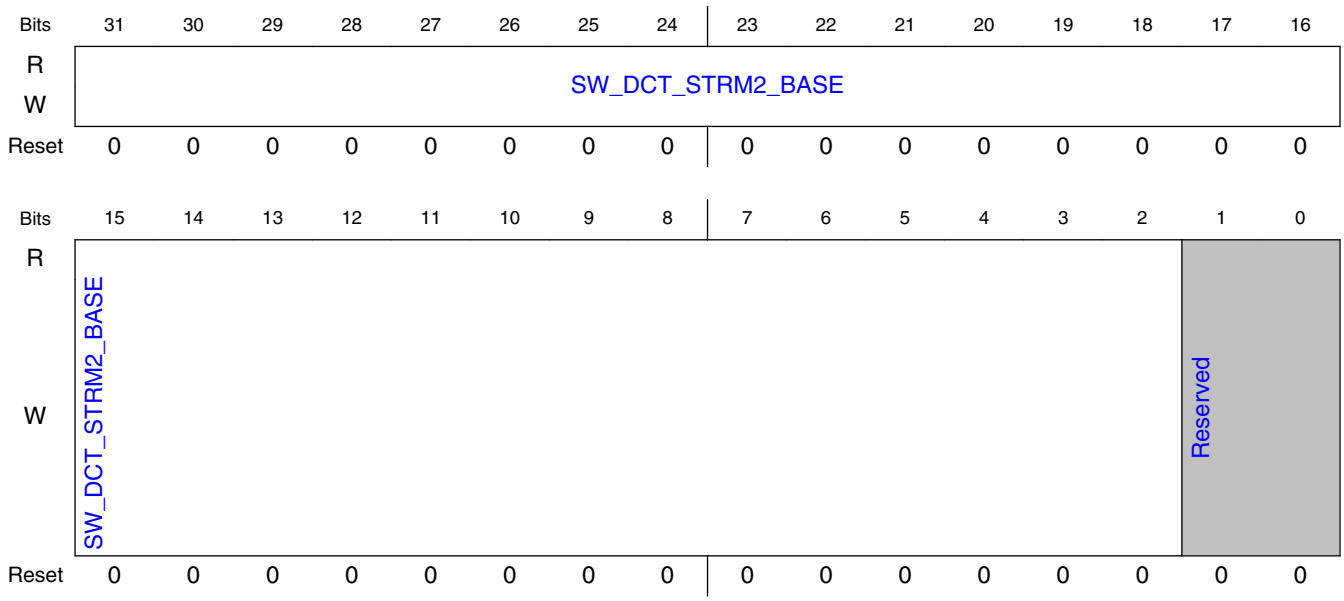
Field	Function
31-2 SW_DCT_STRM1_BASE	Base address for VP7/VP8 DCT stream MB row 1,2n+1
1-0 —	Reserved.

16.1.6.4.12 Base address for reference picture index 9 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 2 base (SWREG23_VP7_VP8)

16.1.6.4.12.1 Offset

Register	Offset
SWREG23_VP7_VP8	5Ch

16.1.6.4.12.2 Diagram



16.1.6.4.12.3 Fields

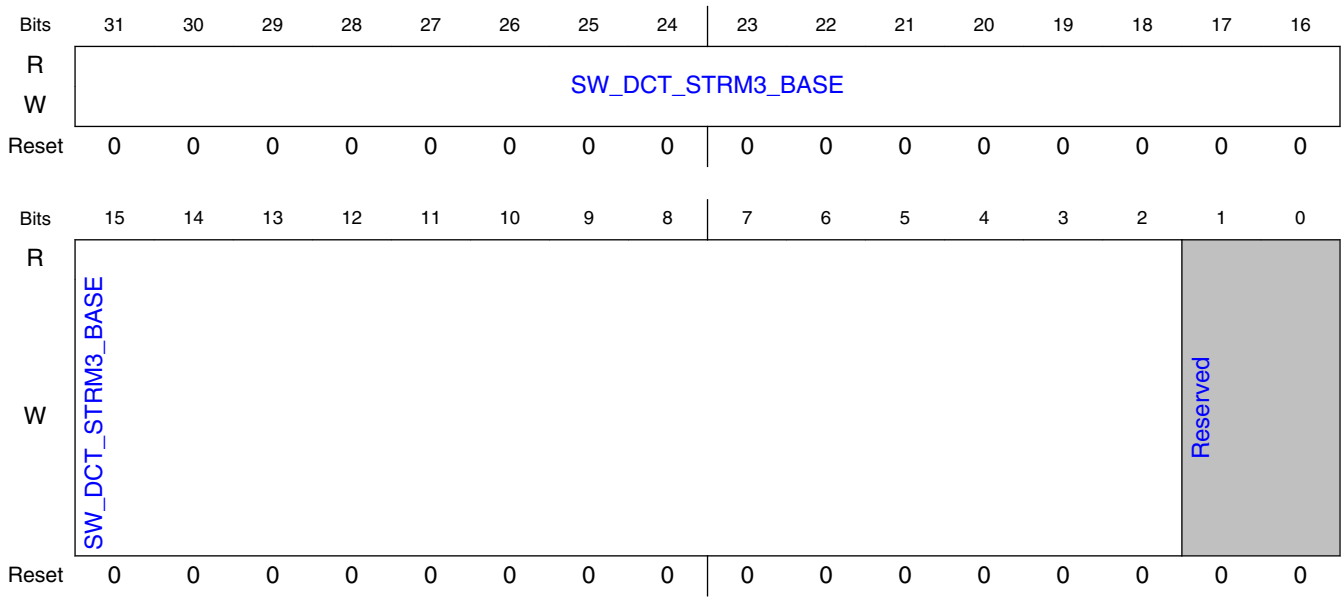
Field	Function
31-2 SW_DCT_STRM2_BASE	Base address for VP7/VP8 DCT stream MB row 2,2n+2
1-0 —	Reserved.

16.1.6.4.13 Base address for reference picture index 10 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 3 base (SWREG24_VP7_VP8)

16.1.6.4.13.1 Offset

Register	Offset
SWREG24_VP7_VP8	60h

16.1.6.4.13.2 Diagram



16.1.6.4.13.3 Fields

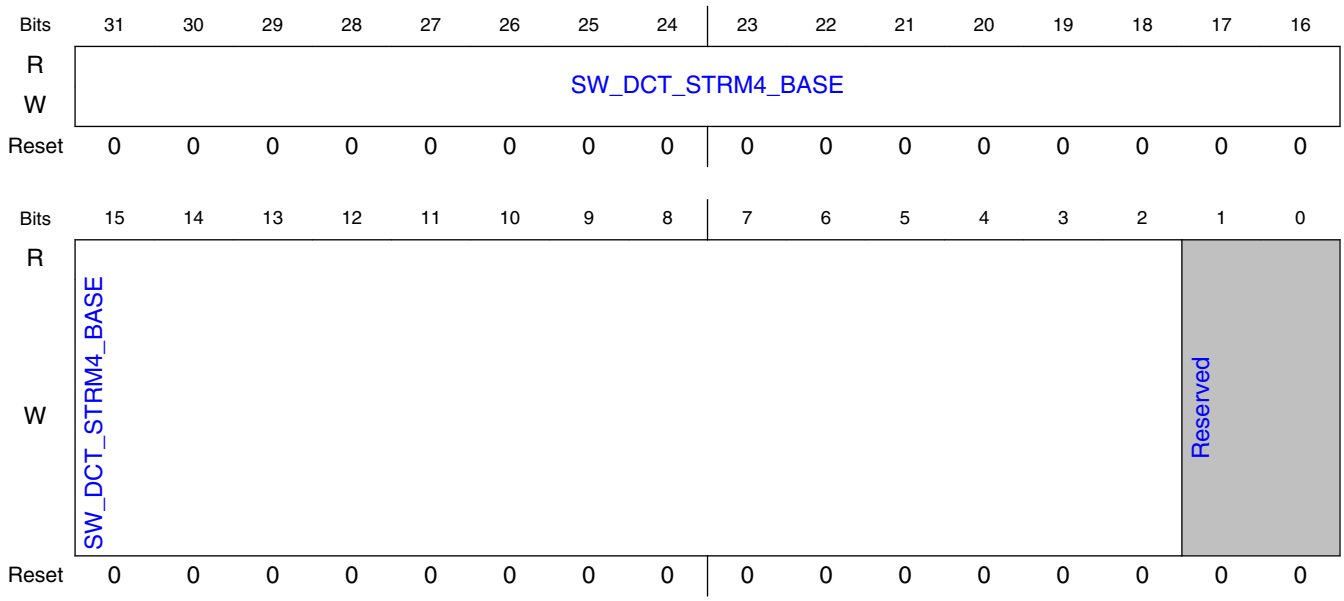
Field	Function
31-2 SW_DCT_STRM3_BASE	Base address for VP7/VP8 DCT stream MB row 3,2n+3
1-0 —	Reserved.

16.1.6.4.14 Base address for reference picture index 11 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 4 base (SWREG25_VP7_VP8)

16.1.6.4.14.1 Offset

Register	Offset
SWREG25_VP7_VP8	64h

16.1.6.4.14.2 Diagram



16.1.6.4.14.3 Fields

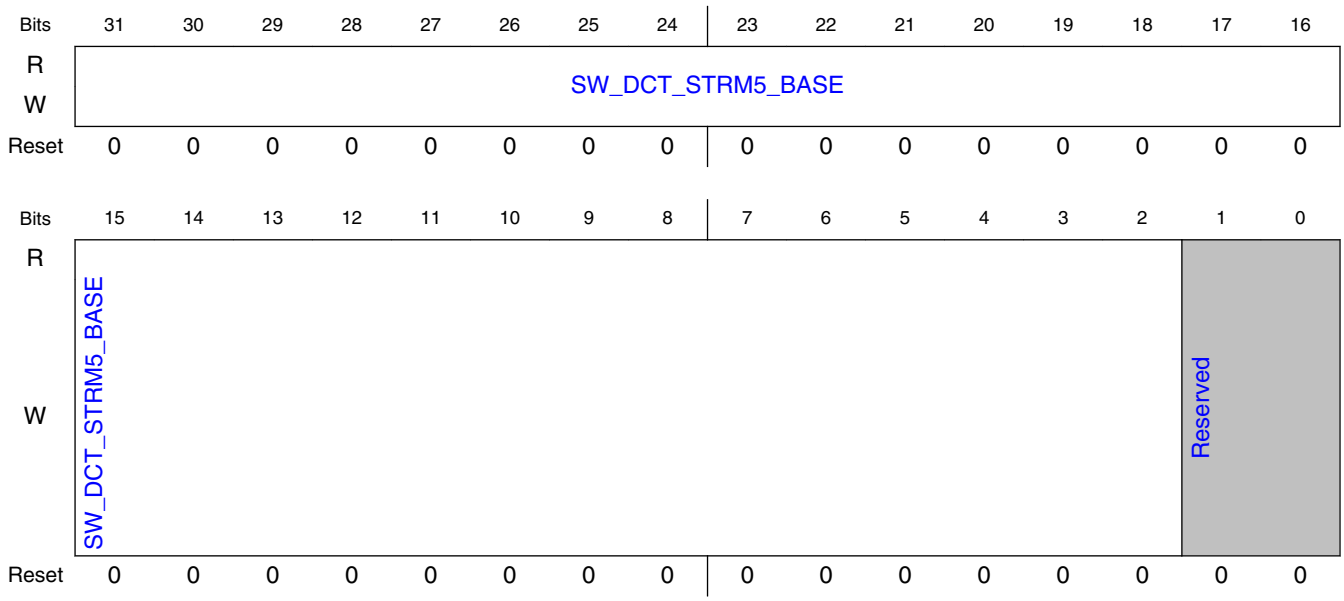
Field	Function
31-2 SW_DCT_STRM4_BASE	Base address for VP7/VP8 DCT stream MB row 4,2n+4
1-0 —	Reserved.

16.1.6.4.15 Base address for reference picture index 12 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 5 base (SWREG26_VP7_VP8)

16.1.6.4.15.1 Offset

Register	Offset
SWREG26_VP7_VP8	68h

16.1.6.4.15.2 Diagram



16.1.6.4.15.3 Fields

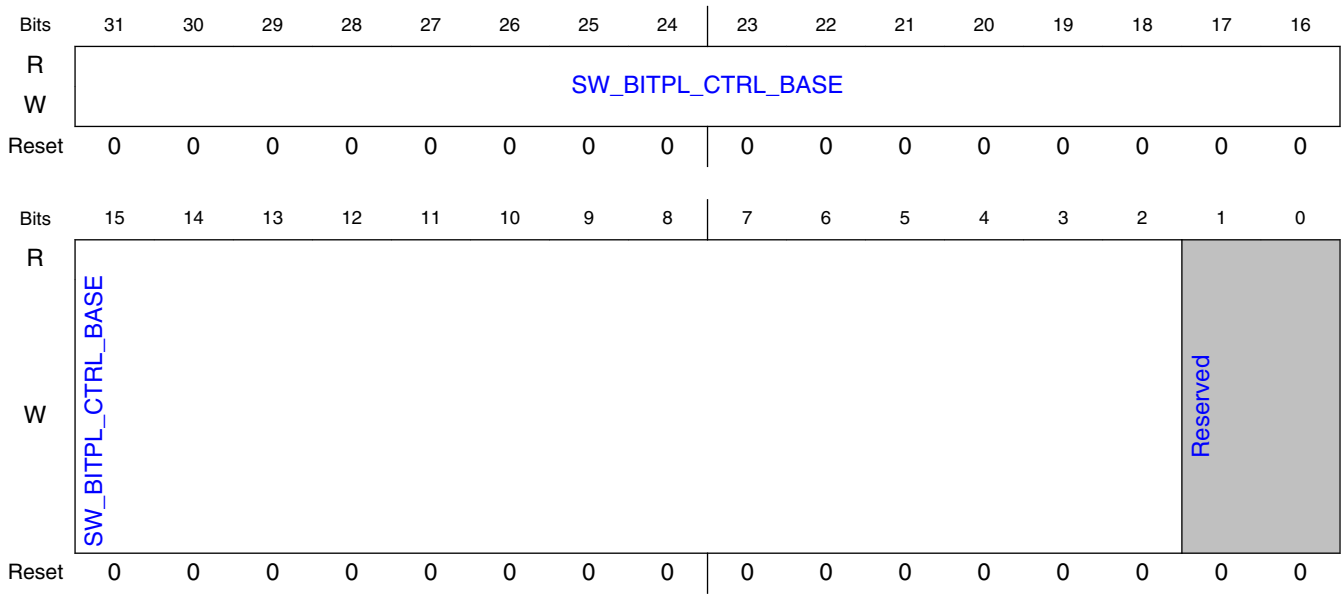
Field	Function
31-2 SW_DCT_STRM5_BASE	Base address for VP7/VP8 DCT stream MB row 5,2n+5
1-0 —	Reserved.

16.1.6.4.16 Base address for reference picture index 13 / VC-1 bitpl mbctrl or VP6,VP7,VP8 ctrl stream base /Progressive JPEG DC table (SWREG27_VC1)

16.1.6.4.16.1 Offset

Register	Offset
SWREG27_VC1	6Ch

16.1.6.4.16.2 Diagram



16.1.6.4.16.3 Fields

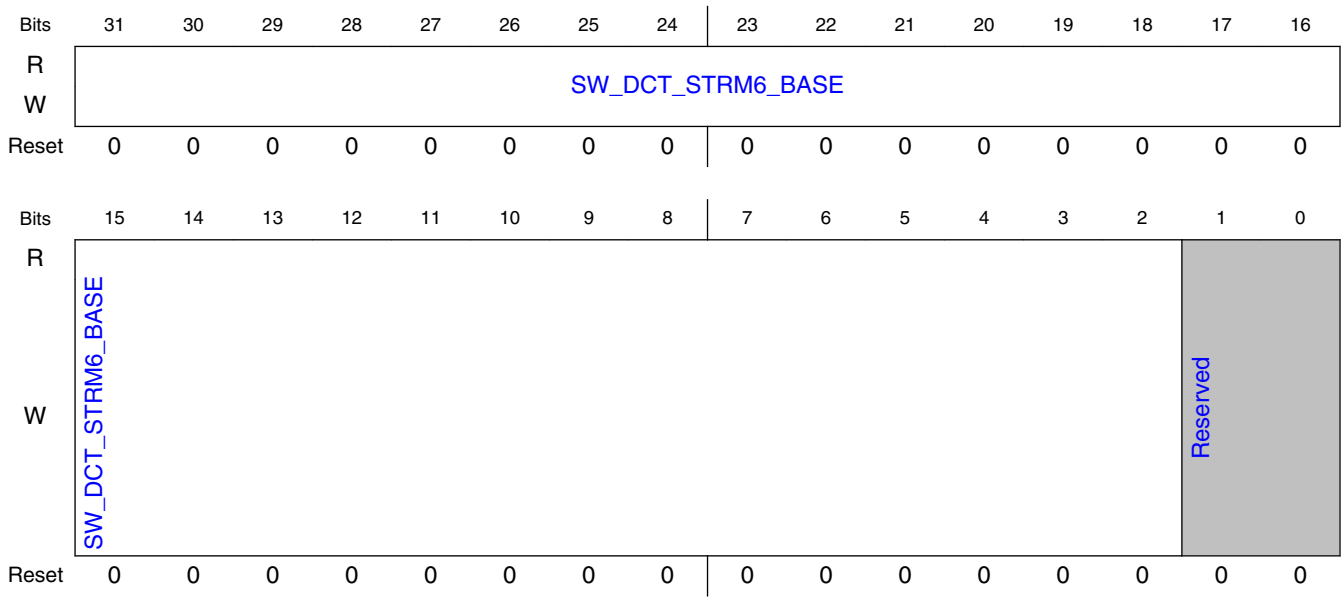
Field	Function
31-2 SW_BITPL_CTRL_BASE	VC-1: Base address for bitplane mb control VP6/VP7/VP8 : Base address for ctrl data stream. Used if multistream is enabled
1-0 —	Reserved.

16.1.6.4.17 Base address for reference picture index 14 / VP6 scan maps / Progressive JPEG DC table / VP7,VP8 DCT stream 6 base (SWREG28_VP7_VP8)

16.1.6.4.17.1 Offset

Register	Offset
SWREG28_VP7_VP8	70h

16.1.6.4.17.2 Diagram



16.1.6.4.17.3 Fields

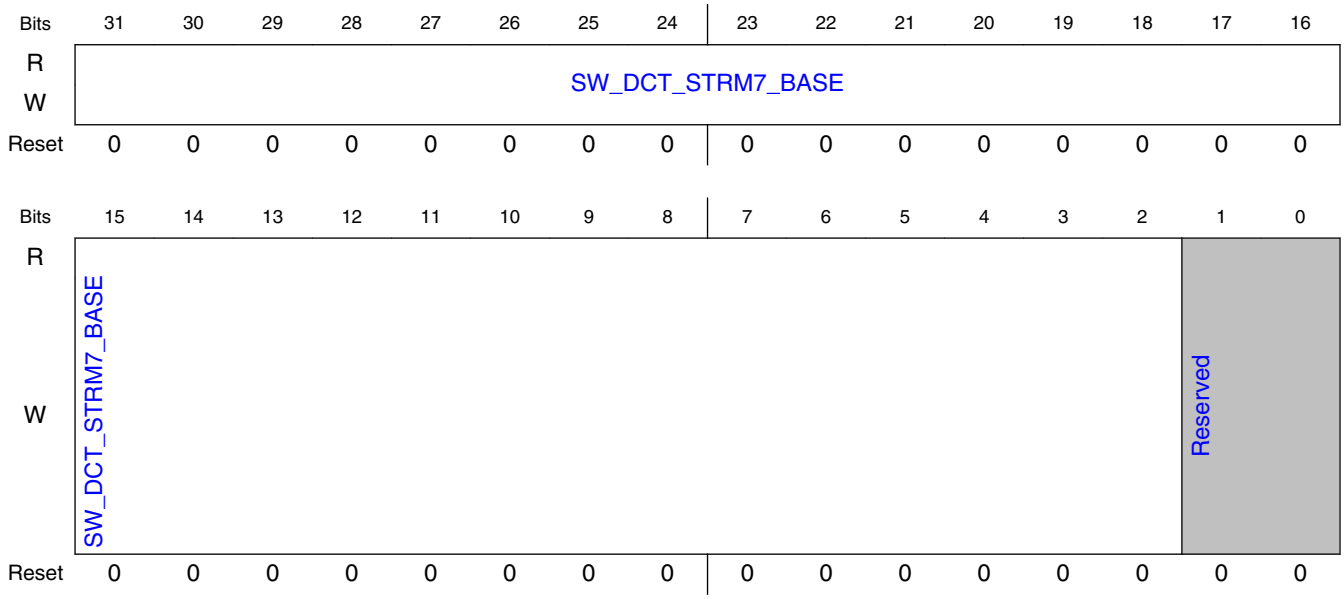
Field	Function
31-2 SW_DCT_STRM6_BASE	Base address for VP7/VP8 DCT stream MB row 6,2n+6
1-0 —	Reserved.

16.1.6.4.18 Base address for reference picture index 15 / VP6 scan maps / VP7,VP8 DCT stream 7 base (SWREG29_VP7_VP8)

16.1.6.4.18.1 Offset

Register	Offset
SWREG29_VP7_VP8	74h

16.1.6.4.18.2 Diagram



16.1.6.4.18.3 Fields

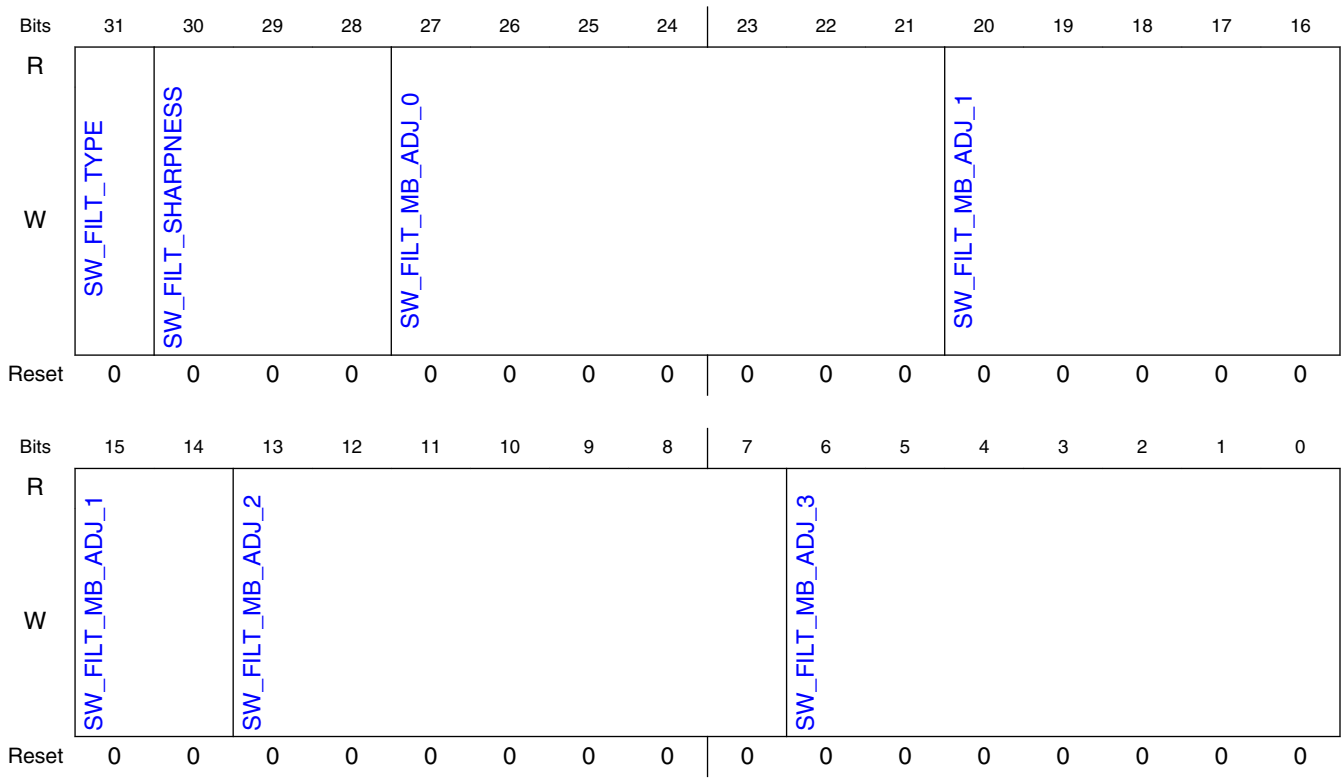
Field	Function
31-2 SW_DCT_STRM7_BASE	Base address for VP7/VP8 DCT stream MB row 7,2n+7
1-0 —	Reserved.

16.1.6.4.19 Reference picture numbers for index 0 and 1 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter mb level adjusts (SWREG30_VP7_VP8)

16.1.6.4.19.1 Offset

Register	Offset
SWREG30_VP7_VP8	78h

16.1.6.4.19.2 Diagram



16.1.6.4.19.3 Fields

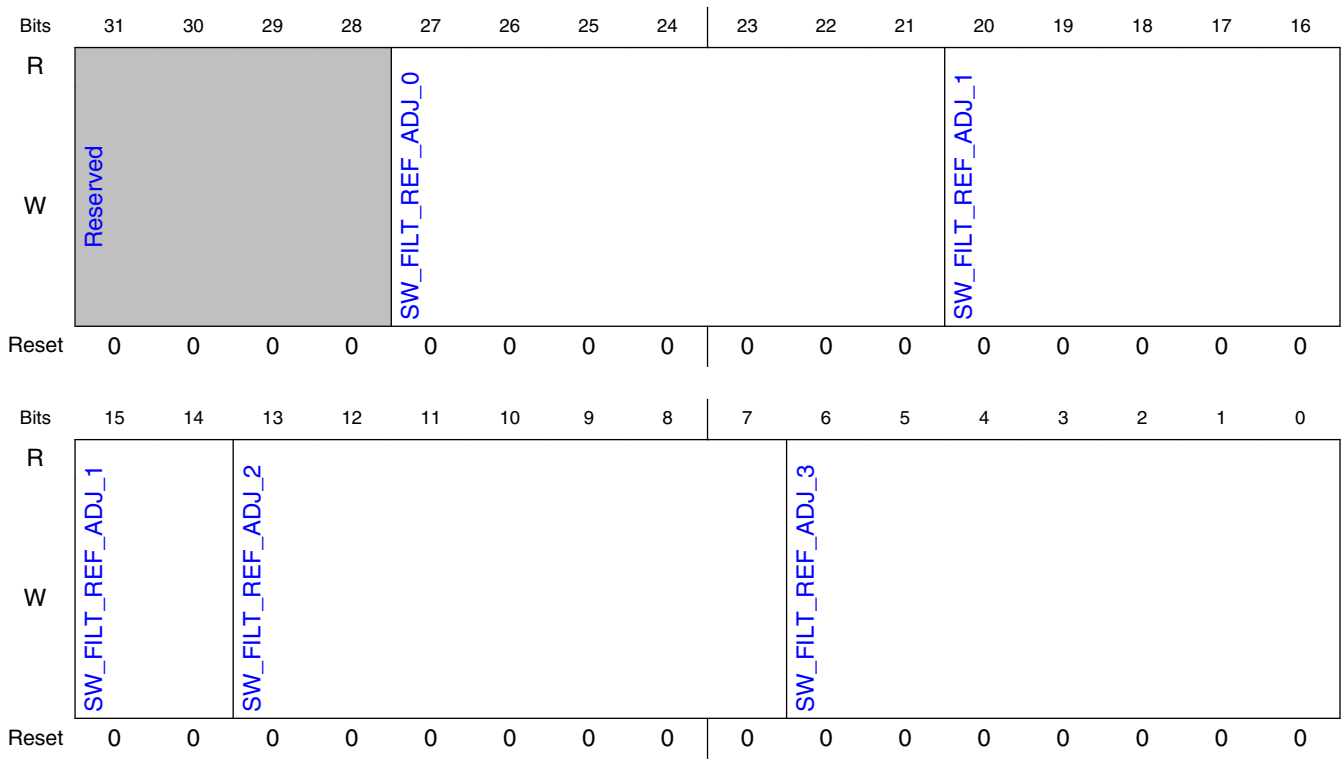
Field	Function
31 SW_FILT_TYPE	VP7/VP8 loop filter type
30-28 SW_FILT_SHARPNESS	VP7/VP8 loop filter sharpness
27-21 SW_FILT_MB_ADJ_0	VP7/VP8 filter level adjustment for MB type 0
20-14 SW_FILT_MB_ADJ_1	VP7/VP8 filter level adjustment for MB type 1
13-7 SW_FILT_MB_ADJ_2	VP7/VP8 filter level adjustment for MB type 2
6-0 SW_FILT_MB_ADJ_3	VP7/VP8 filter level adjustment for MB type 3

16.1.6.4.20 Reference picture numbers for index 2 and 3 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter ref pic level adjusts (SWREG31_VP7_VP8)

16.1.6.4.20.1 Offset

Register	Offset
SWREG31_VP7_VP8	7Ch

16.1.6.4.20.2 Diagram



16.1.6.4.20.3 Fields

Field	Function
31-28 —	Reserved.
27-21 SW_FILT_REF_ADJ_0	VP7/VP8 filter level adjustment for reference frame type 0

Table continues on the next page...

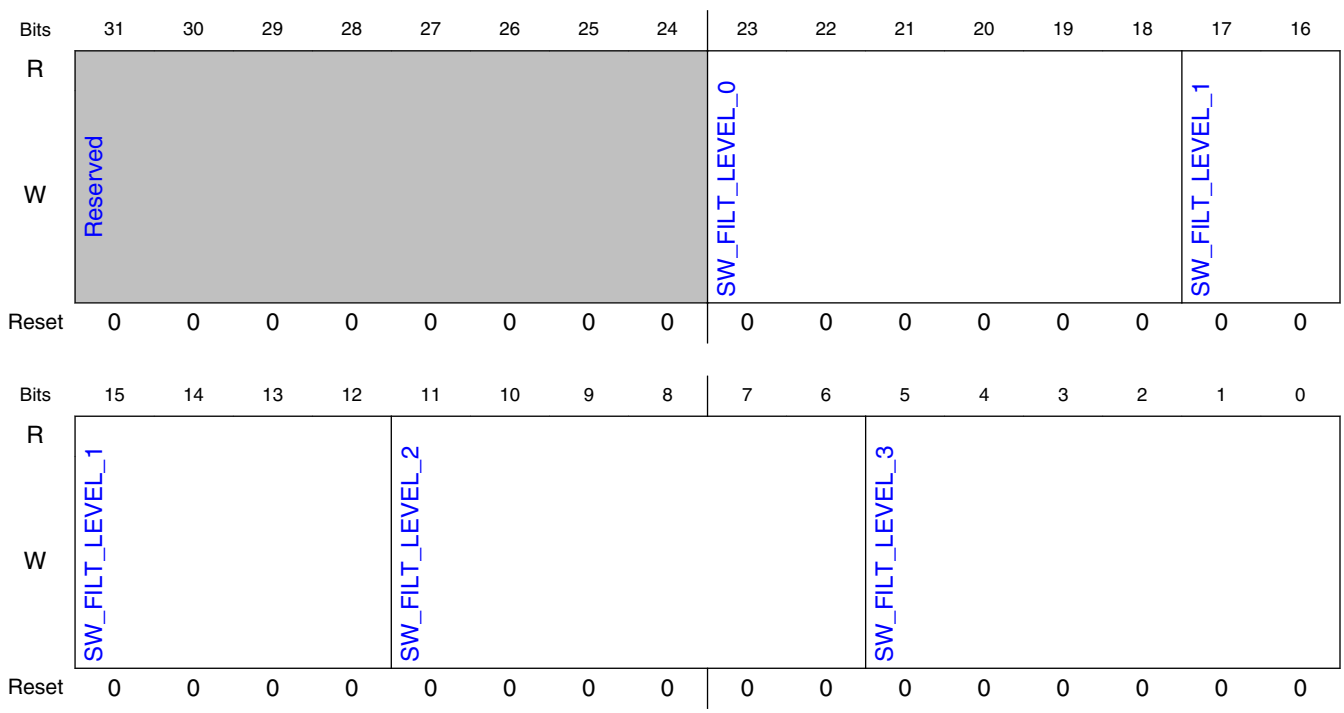
Field	Function
20-14 SW_FILT_REF_ADJ_1	VP7/VP8 filter level adjustment for reference frame type 1
13-7 SW_FILT_REF_ADJ_2	VP7/VP8 filter level adjustment for reference frame type 2
6-0 SW_FILT_REF_ADJ_3	VP7/VP8 filter level adjustment for reference frame type 3

16.1.6.4.21 Reference picture numbers for index 4 and 5 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter levels (SWREG32_VP7_VP8)

16.1.6.4.21.1 Offset

Register	Offset
SWREG32_VP7_VP8	80h

16.1.6.4.21.2 Diagram



16.1.6.4.21.3 Fields

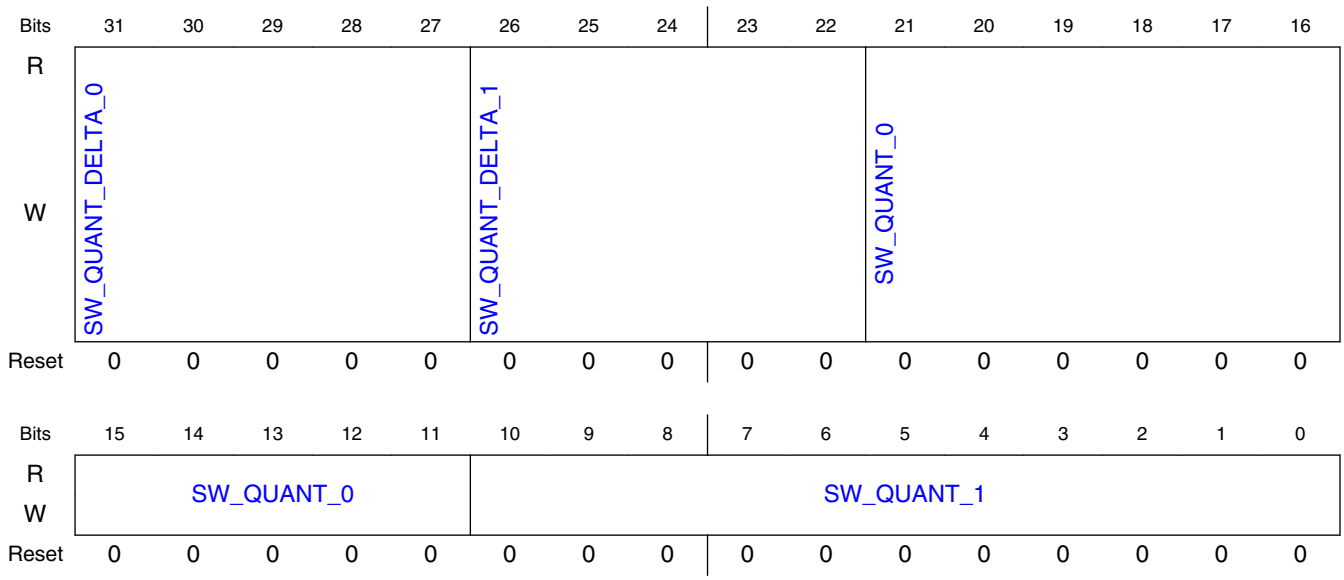
Field	Function
31-24 —	Reserved.
23-18 SW_FILT_LEVEL_0	VP7/VP8 filter level value for reference frame type 0
17-12 SW_FILT_LEVEL_1	VP7/VP8 filter level value for reference frame type 1
11-6 SW_FILT_LEVEL_2	VP7/VP8 filter level value for reference frame type 2
5-0 SW_FILT_LEVEL_3	VP7/VP8 filter level value for reference frame type 3

16.1.6.4.22 Reference picture numbers for index 6 and 7 (H264 VLC) / VP6 scan maps / VP7,VP8 quantization values (SWREG33_VP7_VP8)

16.1.6.4.22.1 Offset

Register	Offset
SWREG33_VP7_VP8	84h

16.1.6.4.22.2 Diagram



16.1.6.4.22.3 Fields

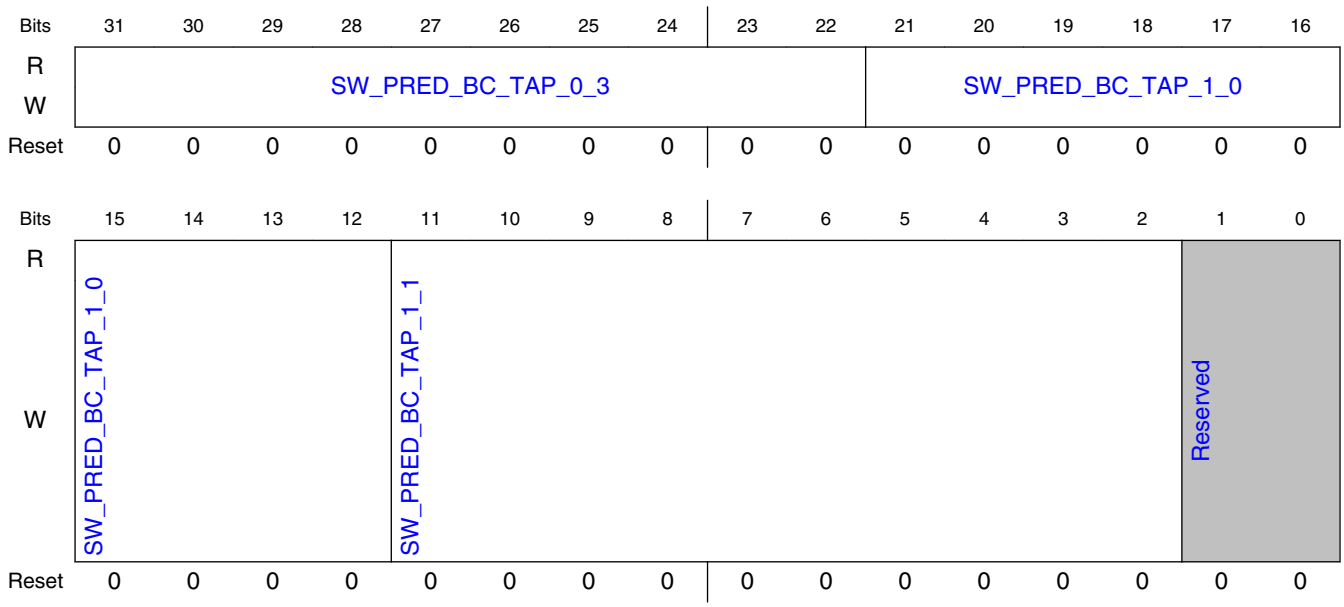
Field	Function
31-27 SW_QUANT_DELTA_0	VP8 quantisizer delta 0
26-22 SW_QUANT_DELTA_1	VP8 quantisizer delta 1
21-11 SW_QUANT_0	VP7: QP (11 bit) VP8: quantisizer value for LUT (7 bit)
10-0 SW_QUANT_1	VP7: QP (11 bit) VP8: quantisizer value for LUT (7 bit)

16.1.6.4.23 Reference picture numbers for index 8 and 9 (H264 VLC) / MPEG4, VC1, VPx prediction filter taps (SWREG34_H263)

16.1.6.4.23.1 Offset

Register	Offset
SWREG34_H263	88h

16.1.6.4.23.2 Diagram



16.1.6.4.23.3 Fields

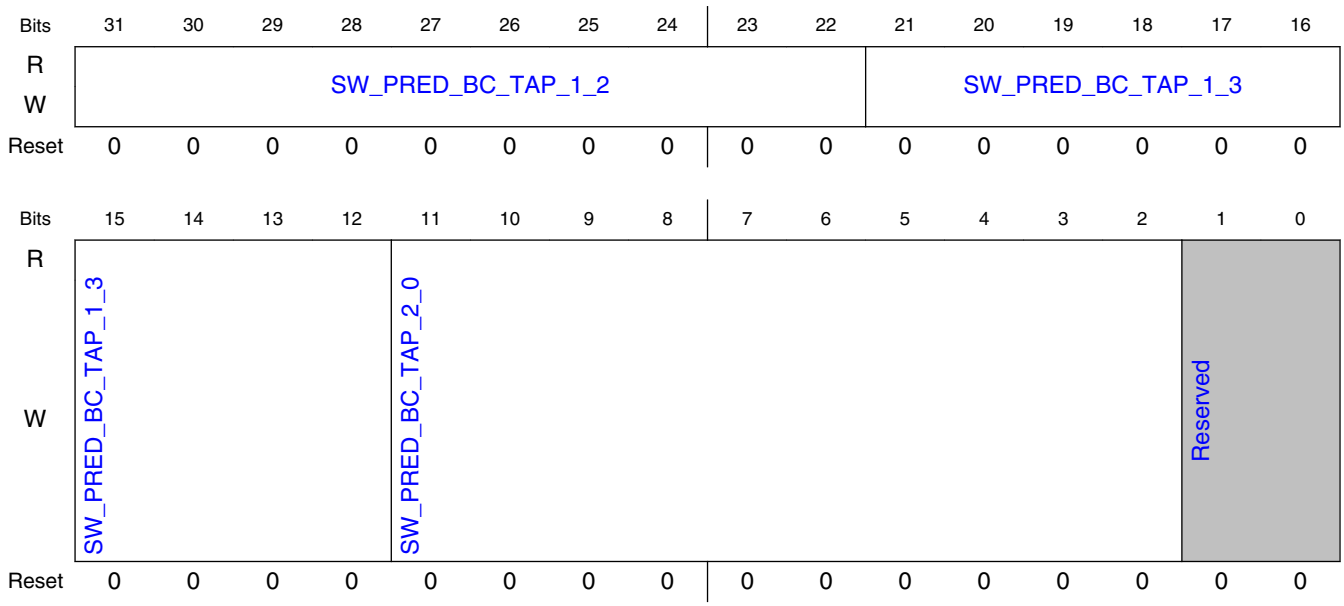
Field	Function
31-22 SW_PRED_BC_TAP_0_3	Prediction filter set 0, tap 3
21-12 SW_PRED_BC_TAP_1_0	Prediction filter set 1, tap 0
11-2 SW_PRED_BC_TAP_1_1	Prediction filter set 1, tap 1
1-0 —	Reserved.

16.1.6.4.24 Reference picture numbers for index 10 and 11 (H264 VLC) / VC1, VPx prediction filter taps (SWREG35_VC1)

16.1.6.4.24.1 Offset

Register	Offset
SWREG35_VC1	8Ch

16.1.6.4.24.2 Diagram



16.1.6.4.24.3 Fields

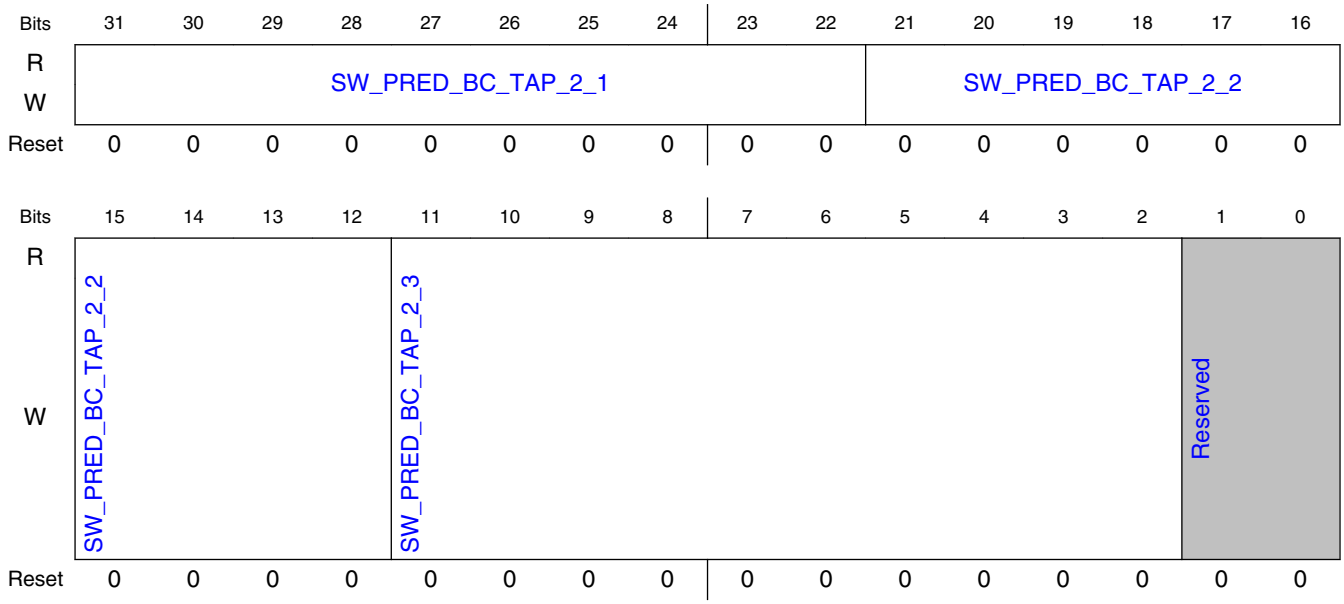
Field	Function
31-22 SW_PRED_BC_TAP_1_2	Prediction filter set 1, tap 2
21-12 SW_PRED_BC_TAP_1_3	Prediction filter set 1, tap 3
11-2 SW_PRED_BC_TAP_2_0	Prediction filter set 2, tap 0
1-0 —	Reserved.

16.1.6.4.25 Reference picture numbers for index 12 and 13 (H264 VLC) / VC1, VPx prediction filter taps (SWREG36_VC1)

16.1.6.4.25.1 Offset

Register	Offset
SWREG36_VC1	90h

16.1.6.4.25.2 Diagram



16.1.6.4.25.3 Fields

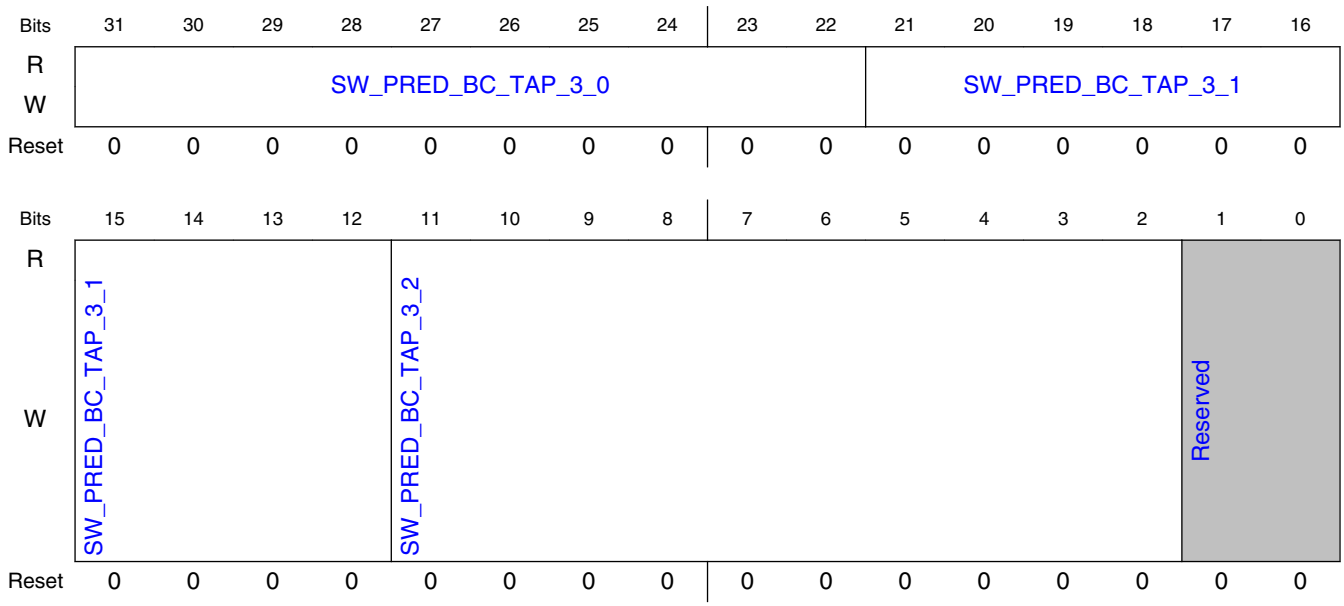
Field	Function
31-22 SW_PRED_BC_TAP_2_1	Prediction filter set 2, tap 1
21-12 SW_PRED_BC_TAP_2_2	Prediction filter set 2, tap 2
11-2 SW_PRED_BC_TAP_2_3	Prediction filter set 2, tap 3
1-0 —	Reserved.

16.1.6.4.26 Reference picture numbers for index 14 and 15 (H264 VLC) / VPx prediction filter taps (SWREG37_VP6_VP7_VP8)

16.1.6.4.26.1 Offset

Register	Offset
SWREG37_VP6_VP7_VP8	94h

16.1.6.4.26.2 Diagram



16.1.6.4.26.3 Fields

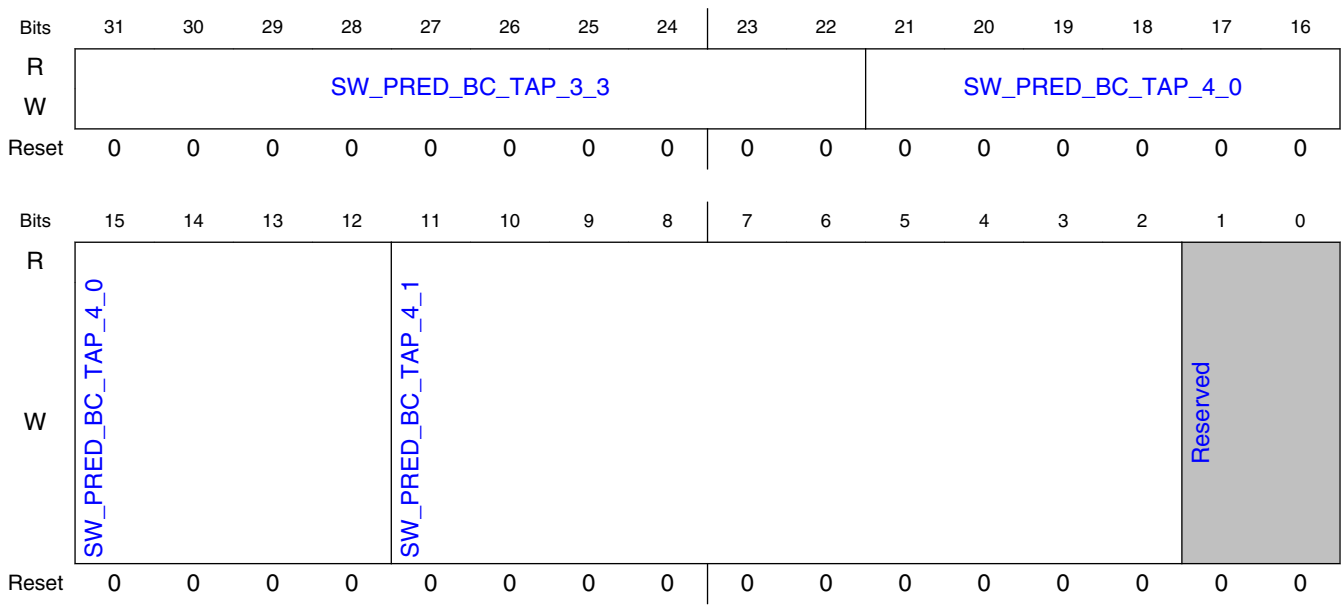
Field	Function
31-22 SW_PRED_BC_TAP_3_0	Prediction filter set 3, tap 0
21-12 SW_PRED_BC_TAP_3_1	Prediction filter set 3, tap 1
11-2 SW_PRED_BC_TAP_3_2	Prediction filter set 3, tap 2
1-0 —	Reserved.

16.1.6.4.27 Reference picture long term flags (H264 VLC) / VPx prediction filter taps (SWREG38_VP6_VP7_VP8)

16.1.6.4.27.1 Offset

Register	Offset
SWREG38_VP6_VP7_VP8	98h

16.1.6.4.27.2 Diagram



16.1.6.4.27.3 Fields

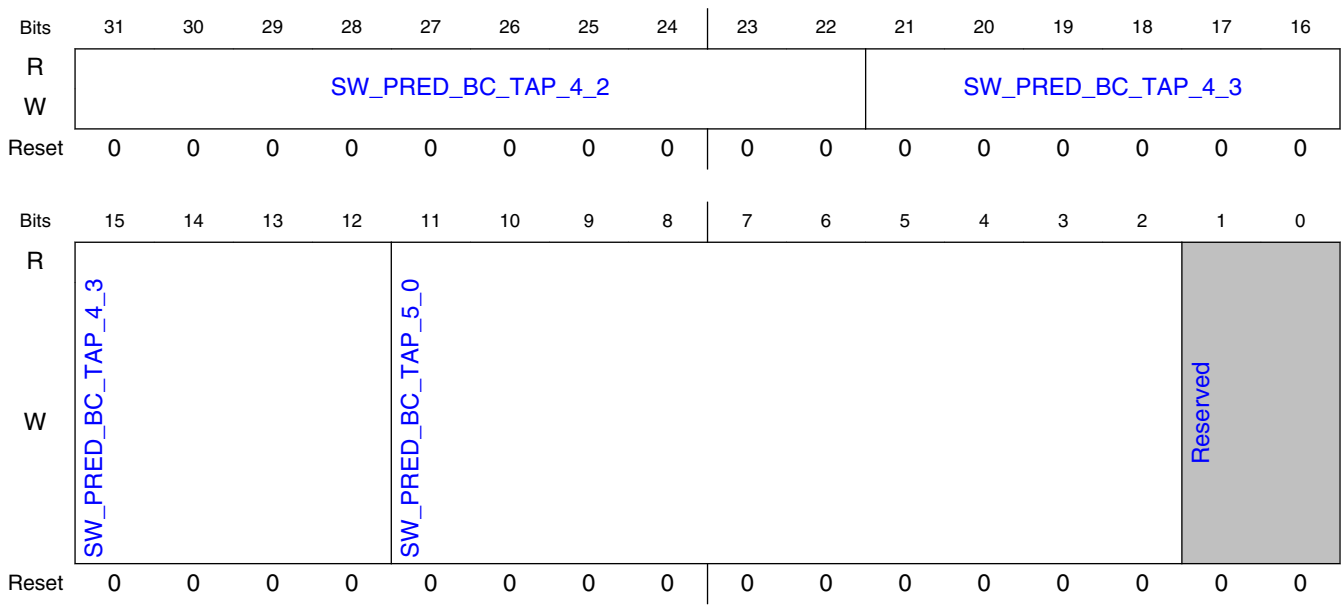
Field	Function
31-22 SW_PRED_BC_TAP_3_3	Prediction filter set 3, tap 3
21-12 SW_PRED_BC_TAP_4_0	Prediction filter set 4, tap 0
11-2 SW_PRED_BC_TAP_4_1	Prediction filter set 4, tap 1
1-0 —	Reserved.

16.1.6.4.28 Reference picture valid flags (H264 VLC) / VPx prediction filter taps (SWREG39_VP6_VP7_VP8)

16.1.6.4.28.1 Offset

Register	Offset
SWREG39_VP6_VP7_VP8	9Ch

16.1.6.4.28.2 Diagram



16.1.6.4.28.3 Fields

Field	Function
31-22 SW_PRED_BC_TAP_4_2	Prediction filter set 4, tap 2
21-12 SW_PRED_BC_TAP_4_3	Prediction filter set 4, tap 3
11-2 SW_PRED_BC_TAP_5_0	Prediction filter set 5, tap 0

Table continues on the next page...

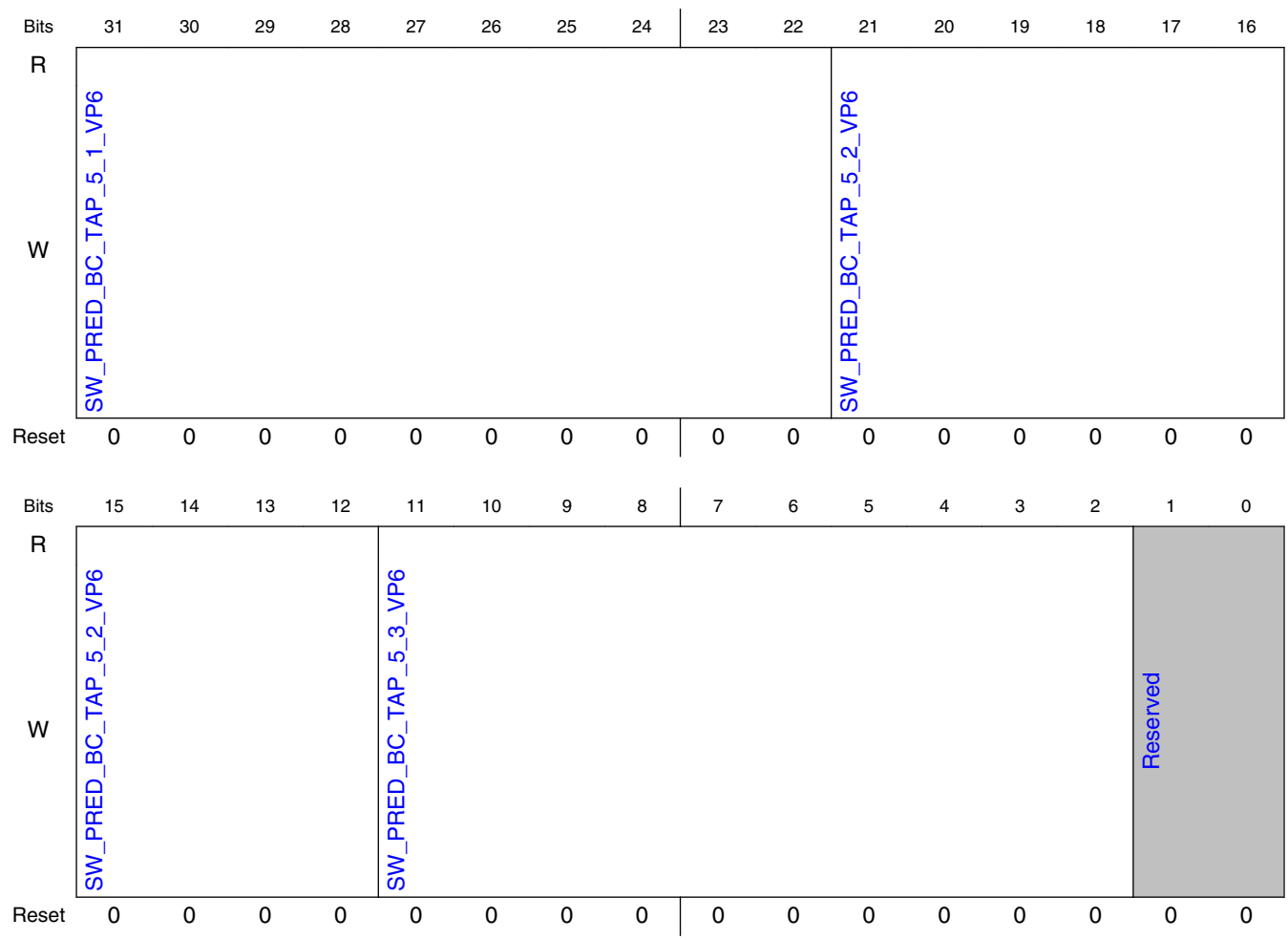
Field	Function
1-0 —	Reserved.

16.1.6.4.29 bi_dir initial ref pic list register (0-2) / VP6 prediction filter taps / Progressive JPEG Cb ACDC coefficient base (SWREG42_VP6)

16.1.6.4.29.1 Offset

Register	Offset
SWREG42_VP6	A8h

16.1.6.4.29.2 Diagram



16.1.6.4.29.3 Fields

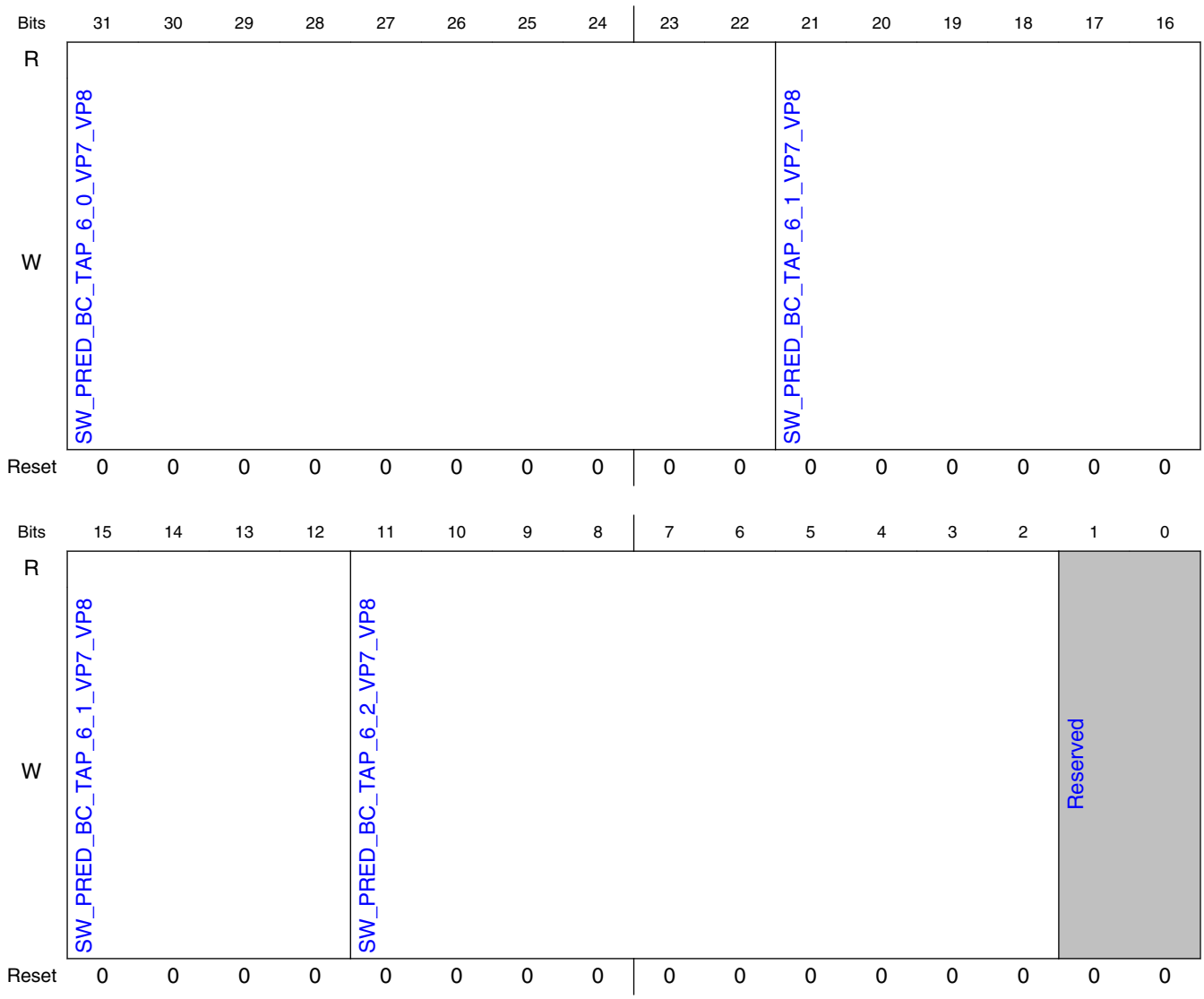
Field	Function
31-22 SW_PRED_BC_TAP_5_1_VP6	Prediction filter set 5, tap 1
21-12 SW_PRED_BC_TAP_5_2_VP6	Prediction filter set 5, tap 2
11-2 SW_PRED_BC_TAP_5_3_VP6	Prediction filter set 5, tap 3
1-0 —	Reserved.

16.1.6.4.30 bi-dir initial ref pic list register (3-5) / VP6 prediction filter taps / Progressive JPEG Cr ACDC coefficient base (SWREG43_VP7_VP8)

16.1.6.4.30.1 Offset

Register	Offset
SWREG43_VP7_VP8	ACh

16.1.6.4.30.2 Diagram



16.1.6.4.30.3 Fields

Field	Function
31-22 SW_PRED_BC_TAP_6_0_VP7_VP8	Prediction filter set 6, tap 0
21-12 SW_PRED_BC_TAP_6_1_VP7_VP8	Prediction filter set 6, tap 1
11-2	Prediction filter set 6, tap 2

Table continues on the next page...

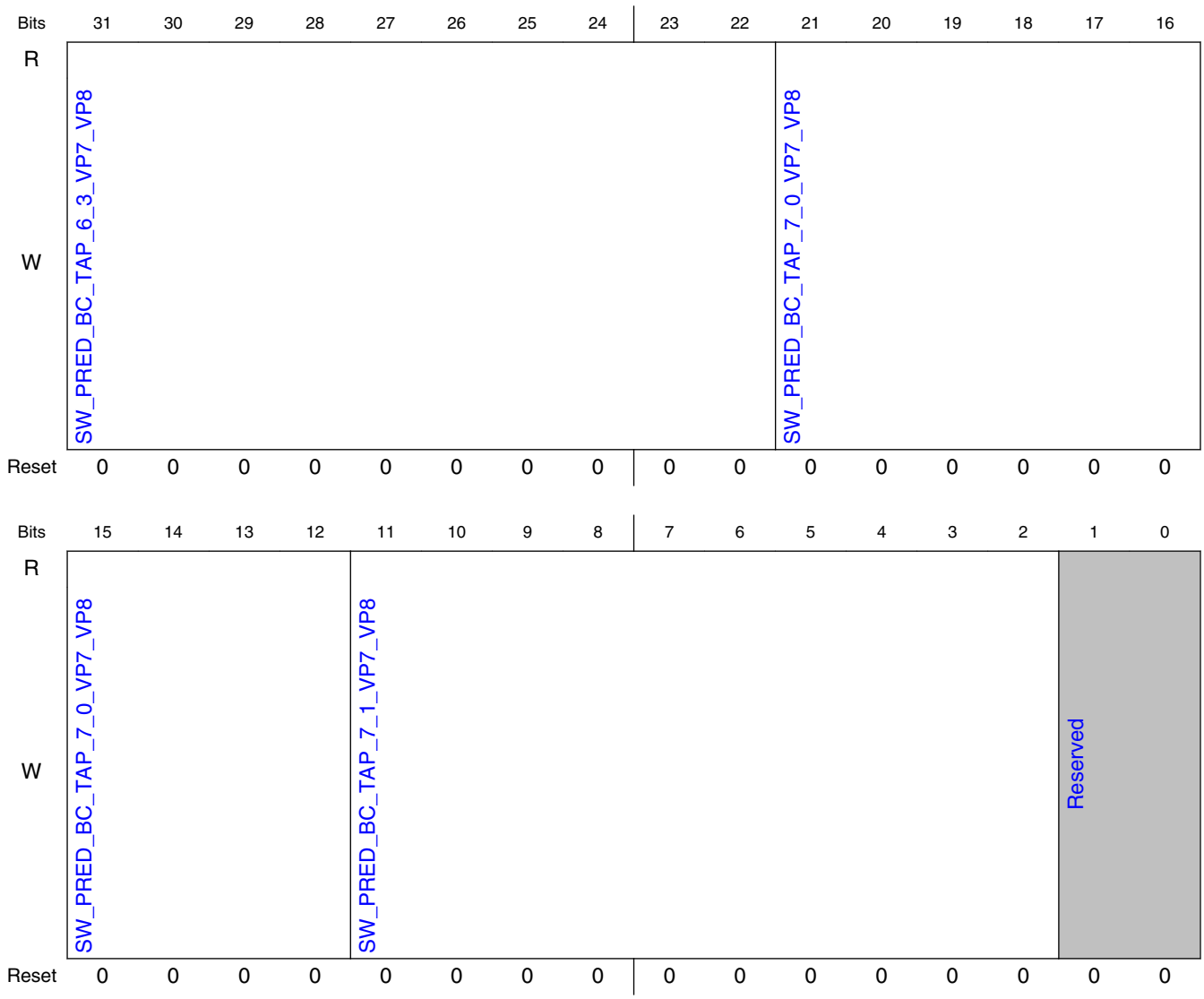
Field	Function
SW_PRED_BC_TAP_6_2_VP7_VP8	
1-0 —	Reserved.

16.1.6.4.31 bi-dir initial ref pic list register (6-8) / VP6 prediction filter taps (SWREG44_VP7_VP8)

16.1.6.4.31.1 Offset

Register	Offset
SWREG44_VP7_VP8	B0h

16.1.6.4.31.2 Diagram



16.1.6.4.31.3 Fields

Field	Function
31-22 SW_PRED_BC_TAP_6_3_VP7_VP8	Prediction filter set 6, tap 3
21-12 SW_PRED_BC_TAP_7_0_VP7_VP8	Prediction filter set 7, tap 0
11-2	Prediction filter set 7, tap 1

Table continues on the next page...

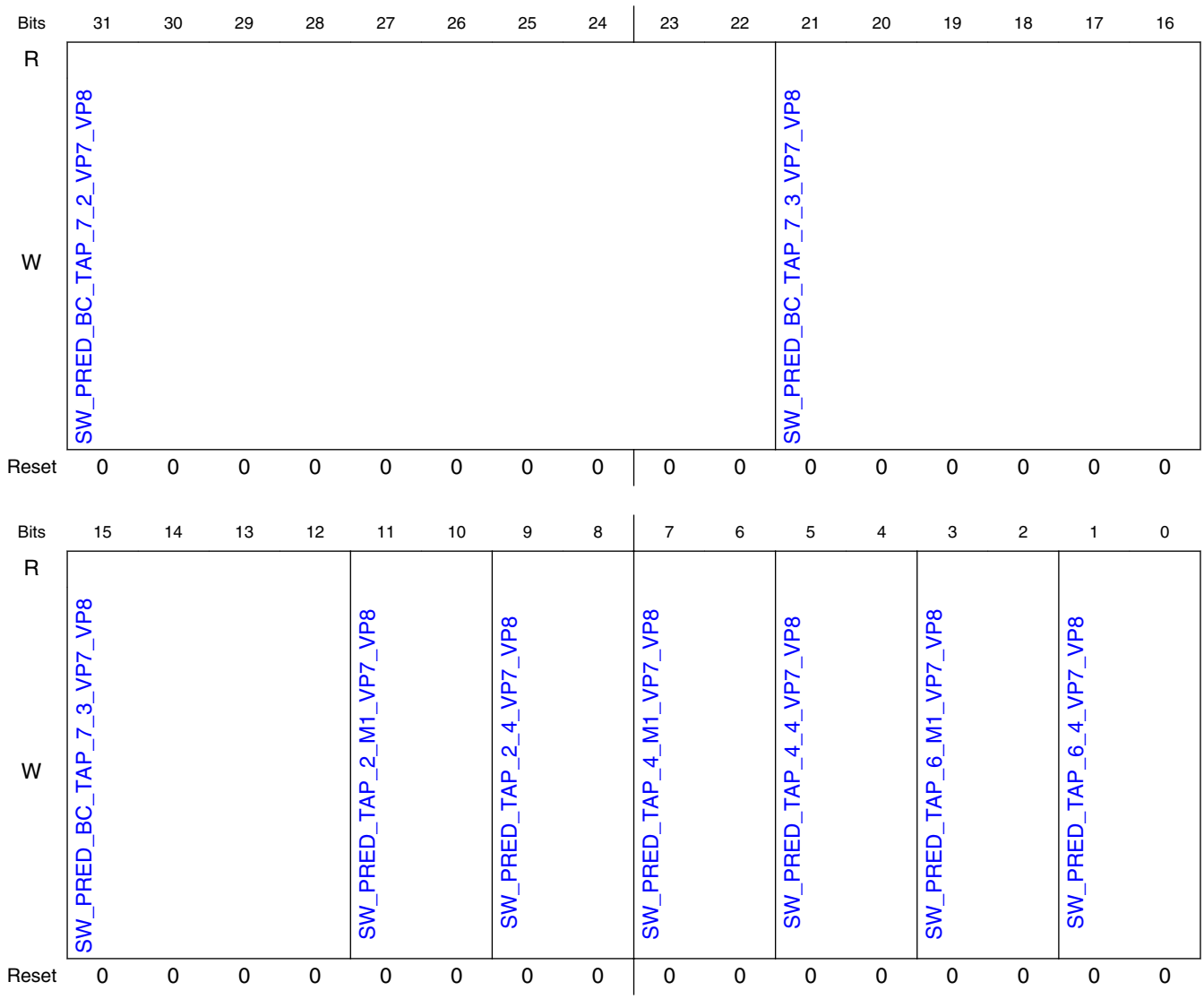
Field	Function
SW_PRED_BC_TAP_7_1_VP7_VP8	
1-0 —	Reserved.

16.1.6.4.32 bi-dir initial ref pic list register (9-11) / VP6 prediction filter taps (SWREG45_VP7_VP8)

16.1.6.4.32.1 Offset

Register	Offset
SWREG45_VP7_VP8	B4h

16.1.6.4.32.2 Diagram



16.1.6.4.32.3 Fields

Field	Function
31-22 SW_PRED_BC_TAP_7_2_VP7_VP8	Prediction filter set 7, tap 2
21-12 SW_PRED_BC_TAP_7_3_VP7_VP8	Prediction filter set 7, tap 3
11-10	Additional Prediction filter tap -1 for set 2

Table continues on the next page...

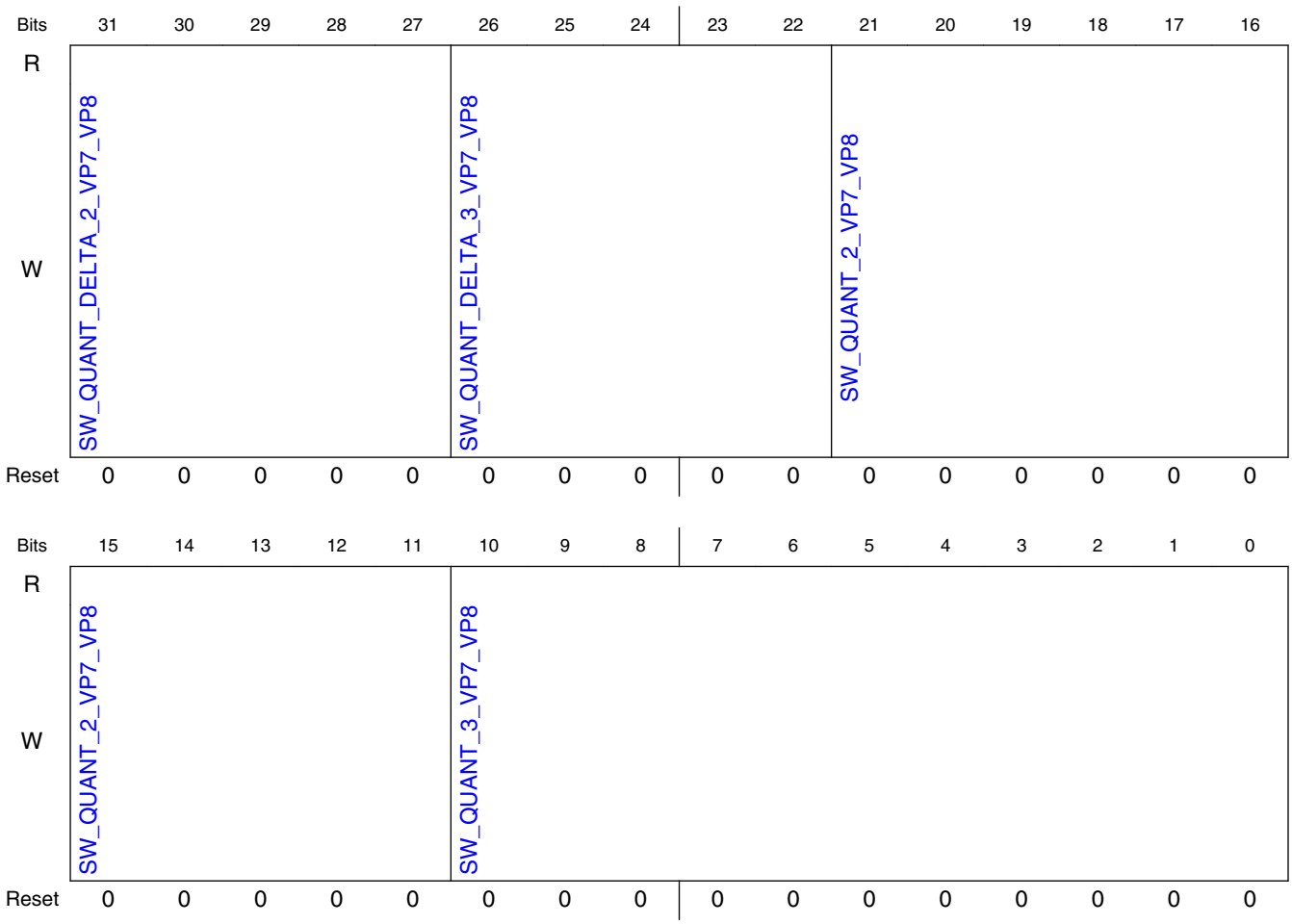
Field	Function
SW_PRED_TAP_2_M1_VP7_VP8	
9-8 SW_PRED_TAP_2_4_VP7_VP8	Additional Prediction filter tap 4 for set 2
7-6 SW_PRED_TAP_4_M1_VP7_VP8	Additional Prediction filter tap -1 for set 4
5-4 SW_PRED_TAP_4_4_VP7_VP8	Additional Prediction filter tap 4 for set 4
3-2 SW_PRED_TAP_6_M1_VP7_VP8	Additional Prediction filter tap -1 for set 6
1-0 SW_PRED_TAP_6_4_VP7_VP8	Additional Prediction filter tap 4 for set 6

16.1.6.4.33 bi-dir initial ref pic list register (12-14) / VP7,VP8 quantization values (SWREG46_VP7_VP8)

16.1.6.4.33.1 Offset

Register	Offset
SWREG46_VP7_VP8	B8h

16.1.6.4.33.2 Diagram



16.1.6.4.33.3 Fields

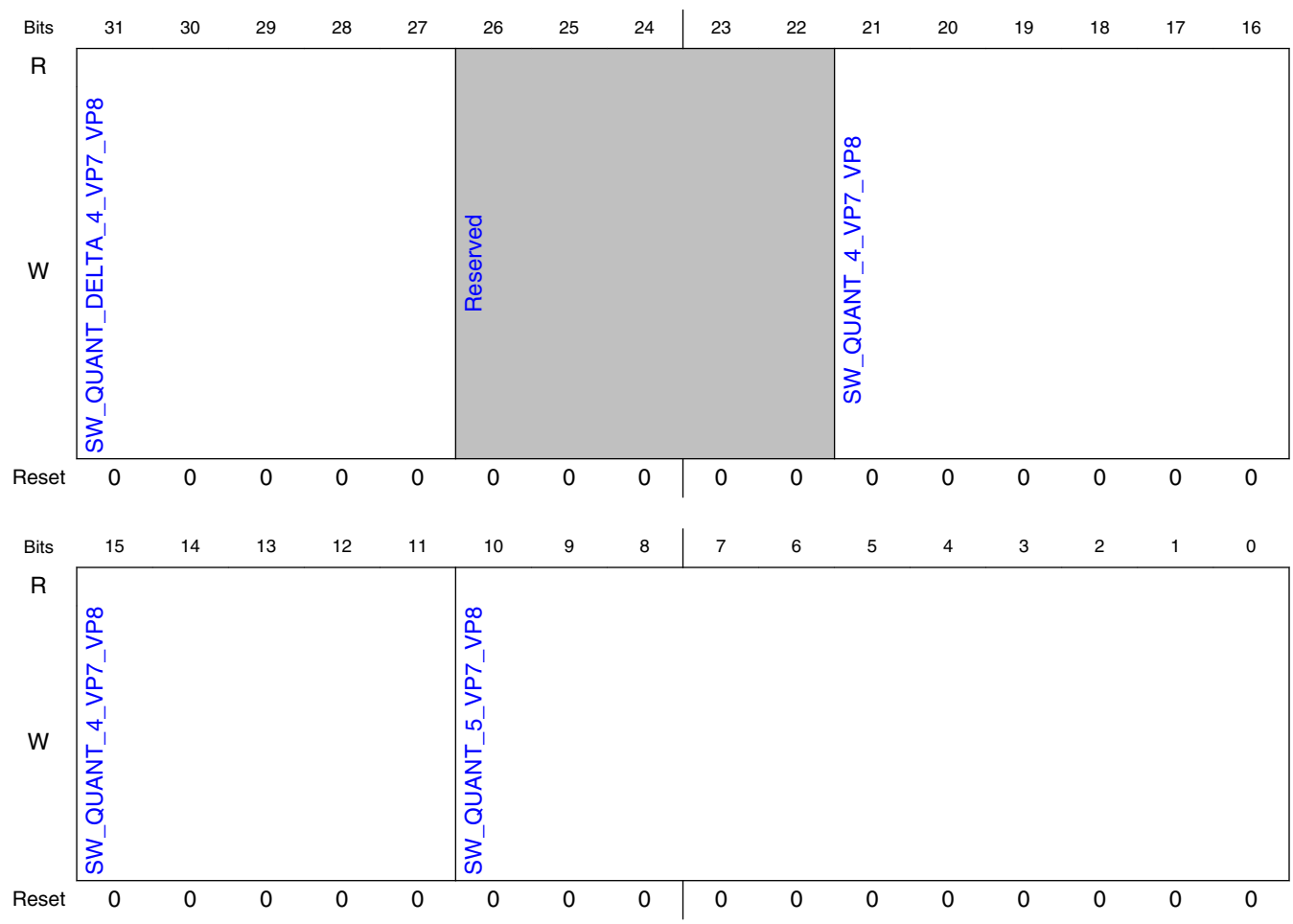
Field	Function
31-27 SW_QUANT_DELTA_2_VP7_VP8	VP8 quantisizer delta 2
26-22 SW_QUANT_DELTA_3_VP7_VP8	VP8 quantisizer delta 3
21-11 SW_QUANT_2_VP7_VP8	VP7: QP (11 bit) VP8: quantisizer value for LUT (7 bit)
10-0 SW_QUANT_3_VP7_VP8	VP7: QP (11 bit) VP8: quantisizer value for LUT (7 bit)

16.1.6.4.34 bi-dir and P fwd initial ref pic list register (15 and P 0-3) / VP7,VP8 quantization values (SWREG47_VP7_VP8)

16.1.6.4.34.1 Offset

Register	Offset
SWREG47_VP7_VP8	BCh

16.1.6.4.34.2 Diagram



16.1.6.4.34.3 Fields

Field	Function
31-27	VP8 quantisizer delta 4

Table continues on the next page...

Field	Function
SW_QUANT_DELTA_4_VP7_VP8	
26-22 —	Reserved.
21-11 SW_QUANT_4_VP7_VP8	VP7 QP (11 bit)
10-0 SW_QUANT_5_VP7_VP8	VP7 QP (11 bit)

16.2 VPU G2 Decoder

16.2.1 Overview

The Video Processing Unit (VPU) G2 hardware-based decoder decodes HEVC and VP9 compatible video streams.

16.2.1.1 Block Diagram

The block diagram of the VPU G2 decoder is shown below.

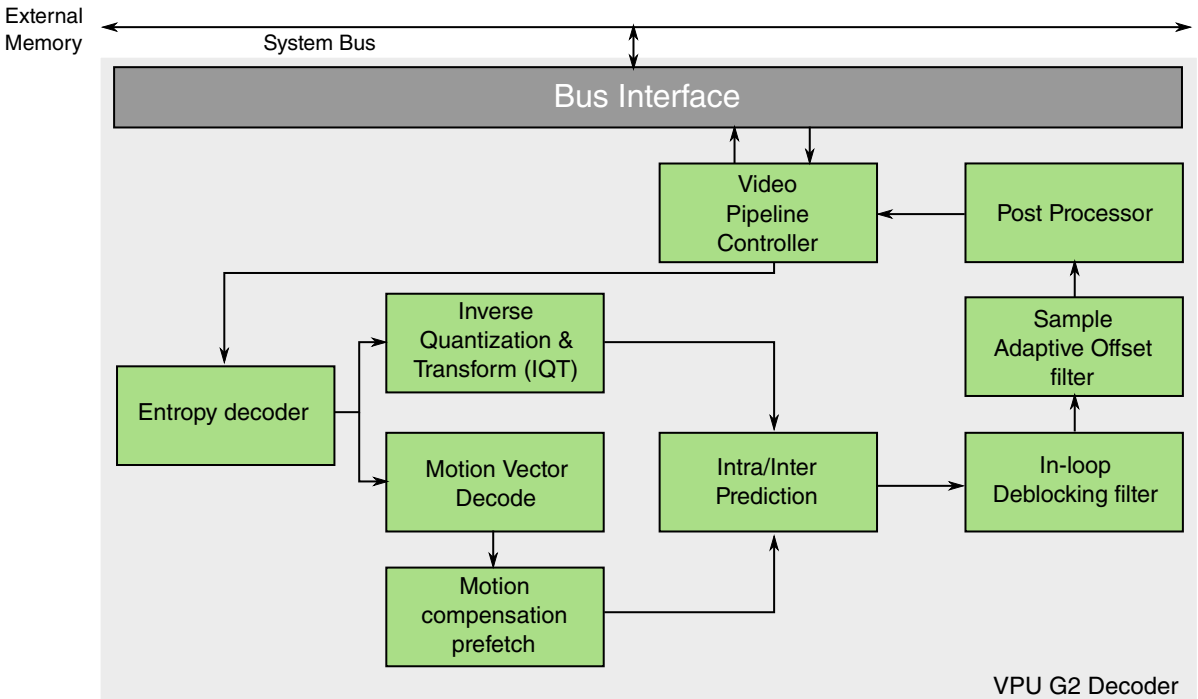


Figure 16-4. VPU G2 Decoder Block Diagram

16.2.1.2 Features

The VPU G2 decoder includes the following features:

- HEVC standard Main/Main 10 profile (up to level 5.1) decoding
- VP9 Profile 0 decoding
- Tiled to raster scan conversion

16.2.2 Functional description

16.2.2.1 Video Frame Storage Format

The decoder supports two storage formats for frames, raster-scan, tiled without being compressed and compressed output. The application using the decoder may specify upon initialization which format is preferred

16.2.2.1.1 Raster-scan format

The output picture of the decoder is in semi-planar YCbCr 4:2:0 format, i.e. luminance data forms one plane in memory, and chrominance data forms the other. The output picture has to be stored linearly and contiguously in the memory. The interleaved

chrominance block has to be stored right after the luminance block in external memory as shown in Figure 16-5 . The number of bytes in one luminance row must be divisible by 16, which means there will be padded 0 in the right edge when necessary.

The interleaved chrominance is stored raster-scanned in memory in Cb0Cr0Cb1Cr1Cb2Cr2Cb3… format.

The decoder output picture for interlaced sequence is always stored as an interlaced frame where even lines belongs to top field and odd lines belongs to bottom field of the frame.

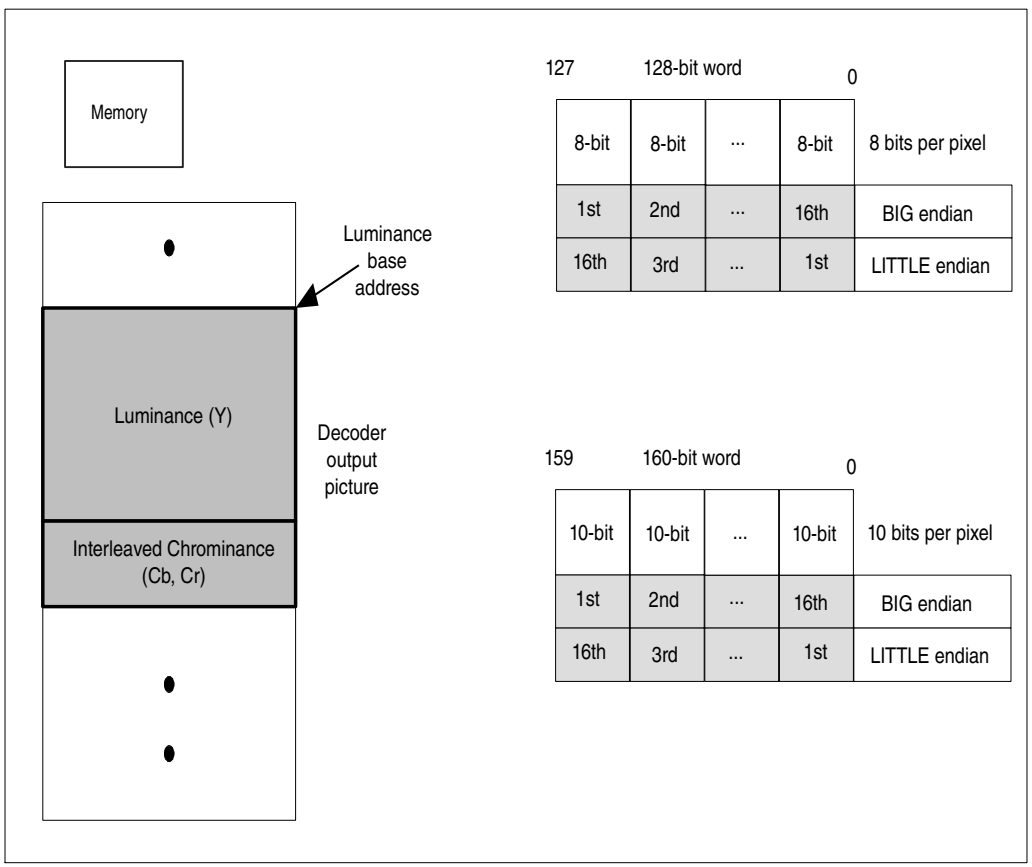


Figure 16-5. External Memory Usage

As an example, the luminance pixels of a QCIF sized (176 x 144) frame with each pixel in 8 bits are numbered as presented in Table 16-3. Table 16-4 shows the storage of the luminance pixel data of the QCIF sized video frame in raster-scan order (in little endian mode).

Table 16-3. Pixel numbers of the luminance data of the QCIF video frame

0	1	2	3	4	5	6	7	…	172	173	174	175
176	177	178	179	180	181	182	183	…	348	349	350	351
::												::

Table continues on the next page...

Table 16-3. Pixel numbers of the luminance data of the QCIF video frame (continued)

25200	25201	25202	25203	25204	25205	25206	25207	…	25340	25341	25342	25343

The storage of the luminance pixel data of a QCIF video frame in raster-scan order. All pixels of a pixel row are stored in consecutive memory locations.

Table 16-4. Pixel Memory Locations

Address offset (decimal)	128-bit words(Pixel data of the luminance component. Each byte contains data for one pixel. Number in table are pixel numbers)			
0	15	...	1	0
16	31	...	17	16
32	47	...	33	32
48	63	...	49	48
64	79	...	65	64
:	:			
25296	25311	...	25297	25296
25312	25327	...	25313	25312
25328	25343	...	25329	25328

16.2.2.1.2 Tiled format

The output picture of the decoder is in semi-planar YCbCr 4:2:0 4x4 tiled format, i.e. luminance data forms one plane in memory, and chrominance data forms another. The distinction from raster scan format is that the output picture is grouped into tiles, which are then stored linearly and contiguously in the memory. The chrominance tiles have to be stored right after the luminance tiles in external memory as shown in Figure . The number of luminance pixels in one row must be divisible by 16.

The chrominance is stored into tiles in interleaved Cb0Cr0Cb1Cr1Cb2Cr2Cb3… format.

Figure 16-6 contains example external memory usage when using 4x4 tiles, which is used in G2 decoder.

Below is an example code of converting tiled YCbCr 4:2:0 output picture back to raster scan format. The code is in generalized form, i.e. can function on arbitrary tile sizes as long as the picture dimensions are multiple of tile sizes. In the example, chrominance data is interleaved into 4x4 tile so that one 4x4 tile contains one 2x4 block of Cb and Cr each. Output is YCbCr 4:2:0 semiplanar.

```
void ConvertTiledToRaster( DecPicture decPicture, u32 tileWidth,
            u32 tileHeight, u8 *pOutput )
            {
            u32 i, j;
            u32 k, l;
```

```
u32 s, t;
u8 *pIn;
pIn = decPicture.pOutputPicture;
/* loop all tile rows for luminance */
for( i = 0 ; i < decPicture.frameHeight ; i += tileHeight )
{
t = 0;
s = 0;
/* loop all tiles in one row */
for( j = 0 ; j < decPicture.frameWidth ; j += tileWidth )
{
/* copy one tile */
for( k = 0 ; k < tileHeight ; ++k )
{
for( l = 0 ; l < tileWidth ; ++l )
{
pOutput[ k*decPicture.frameWidth + l + s ] = pIn[t++];
}
}
/* move to next horizontal tile */
s += tileWidth;
}
/* move to next tile row */
pOutput += decPicture.frameWidth * tileHeight;
pIn += decPicture.frameWidth * tileHeight;
}
/* setup pointers to chrominance data */
pIn = decPicture.pOutputPicture +
decPicture.frameWidth * decPicture.frameHeight;
/* loop all tile rows for chrominance */
for( i = 0 ; i < decPicture.frameHeight/2 ; i += tileHeight )
{
t = 0;
s = 0;
/* loop all tiles in one row */
for( j = 0 ; j < decPicture.frameWidth ; j += tileWidth )
{
/* copy one tile */
for( k = 0 ; k < tileHeight ; ++k )
{
for( l = 0 ; l < tileWidth ; ++l )
{
pOutput[ k*decPicture.frameWidth + l + s ] = pIn[t++];
}
}
/* move to next horizontal tile */
s += tileWidth;
}
/* move to next tile row */
pOutput += decPicture.frameWidth * tileHeight;
pIn += decPicture.frameWidth * tileHeight;
}
}
```

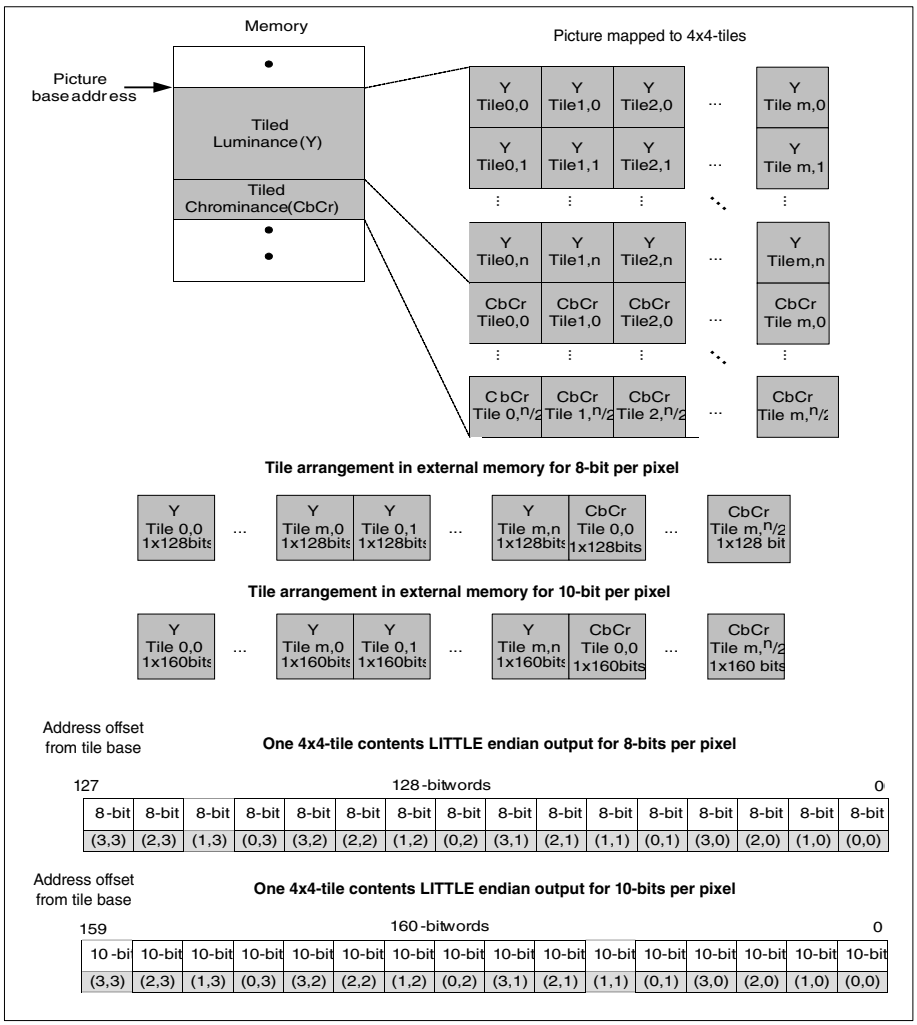


Figure 16-6. External Memory Usage in 4x4 Tiled Reference Picture Mode

16.2.2.1.3 Compressed format

G2 decoder supports compressed tiled output, when reference buffer compression (RFC) is enabled when synthesis, and user allows compressed reference frame output directly.

Reference frame compression is a feature added in G2 decoder to compress frame buffer so that the bandwidth of storing/loading reference frame can be reduced, especially when the resolution of decoded stream is of high definition.

When compression is enabled, the picture is divided into CBS rows, which is further divided into continuous CBS groups. Each CBS group is composed of 16 CBS. The luminance CBS is composed of 1 8x8 coded block (CB), which one chrominance CBS is composed of two 8x4 coded blocks, with cb CB first then cr CB following. The compression is performed to each CB in the CBS. The first CB's compressed data is

saved from the same offset of that CBS in raster scan buffer. And the compressed output of following CBs in the CBS row is saved continuously. That means there may be gaps between the compressed data of each CBS row.

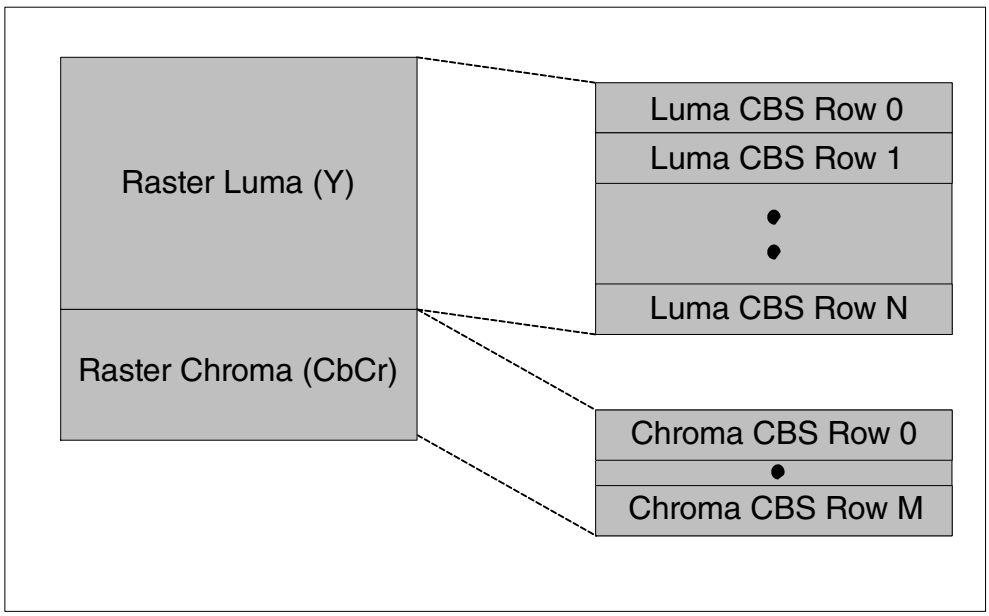


Figure 16-7. Raster scan picture mapped to CBS rows

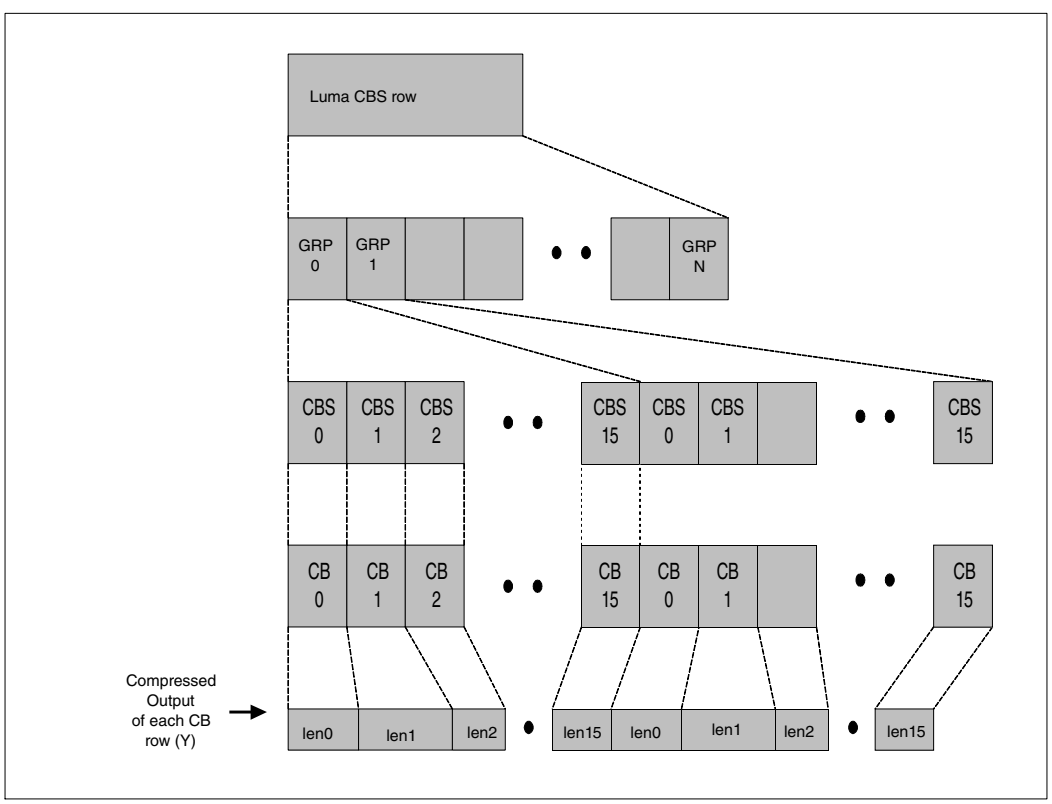


Figure 16-8. Compression performed to each Luma CBS row

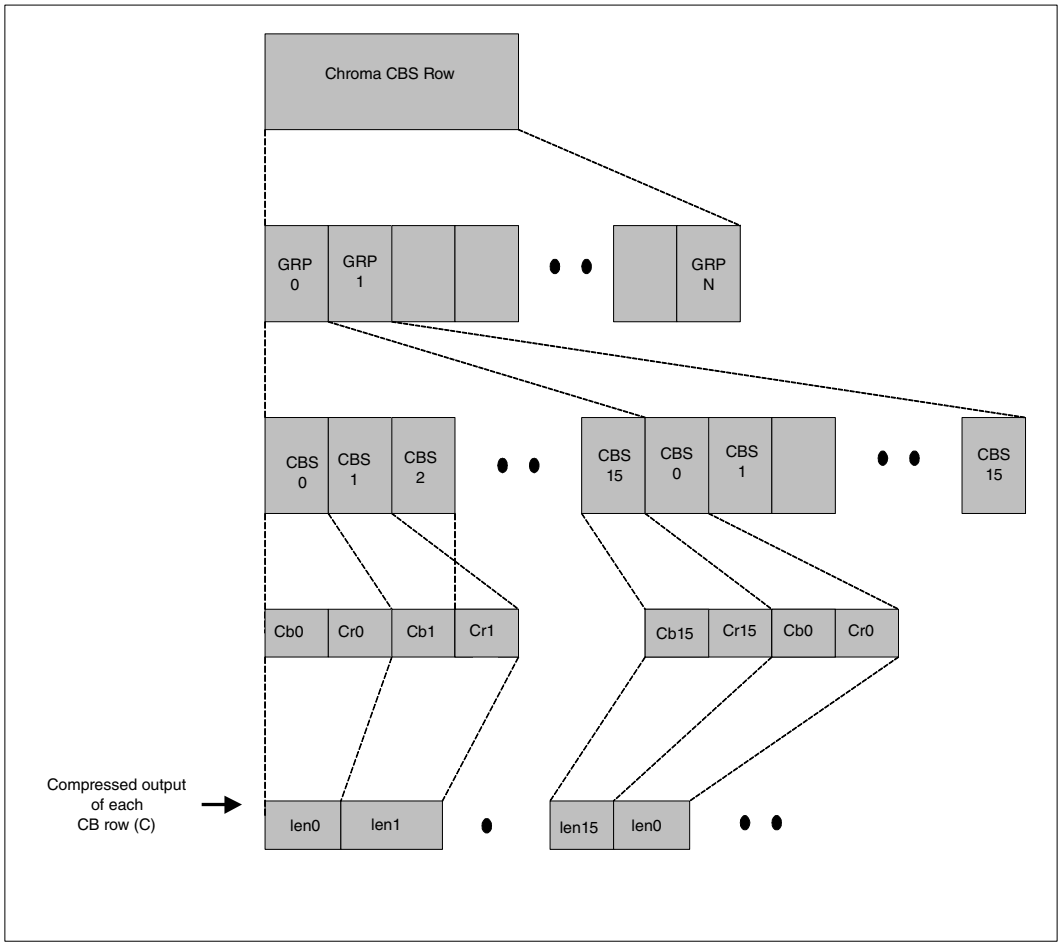


Figure 16-9. Compression performed to each Chroma CBS row

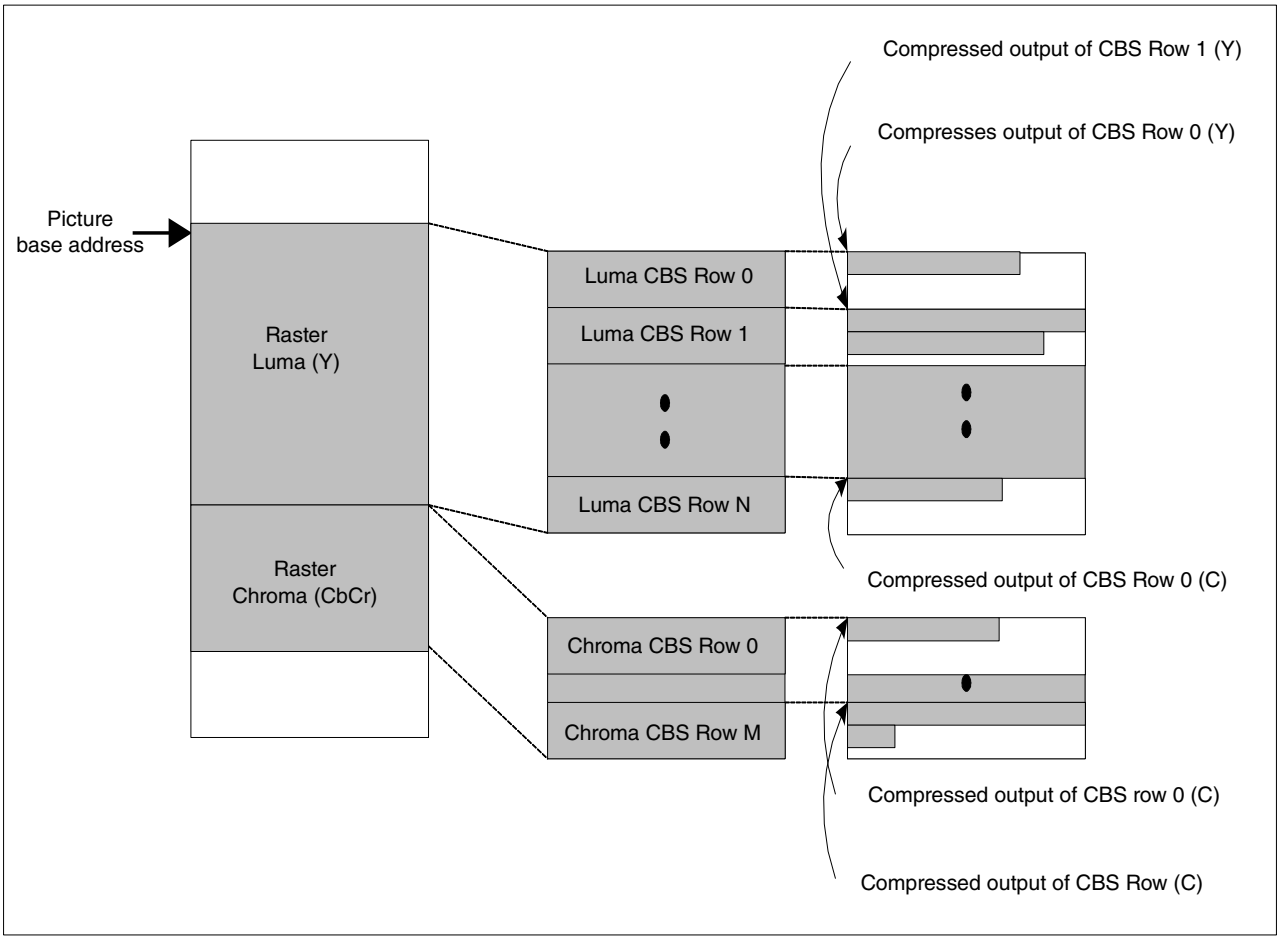


Figure 16-10. Map of external memory in RFC output buffer

Corresponding to each CBS group the starting offset of first compressed CBS output is saved in two bytes in CBS table, followed by the length of each CBS compress output, which is saved in 7 bits. So the table length of each CBS group information is 2+7*16/8=16 bytes.

This 16 bytes for each CBS group in compression table is mapped as shown in the following table. Here, X[n] means the bit-n of the length of CBS X in CBS group.

Table 16-5. Storage of the compression cache table for each CBS group

Byte	Bit7	Bit6	Bit5	Bit4	Bit 3	Bit 2	Bit 1	Bit 0
0	Offset[7:0]							
1	Offset[15:8]							
2	1[0]	0[6]	0[5]	0[4]	0[3]	0[2]	0[1]	0[0]
3	2[1]	2[0]	1[6]	1[5]	1[4]	1[3]	1[2]	1[1]
4	3[2]	3[1]	3[0]	2[6]	2[5]	2[4]	2[3]	2[2]
5	4[3]	4[2]	4[1]	4[0]	3[6]	3[5]	3[4]	3[3]
6	5[4]	5[3]	5[2]	5[1]	5[0]	4[6]	4[5]	4[4]

Table continues on the next page...

Table 16-5. Storage of the compression cache table for each CBS group (continued)

Byte	Bit7	Bit6	Bit5	Bit4	Bit 3	Bit 2	Bit 1	Bit 0
7	6[5]	6[4]	6[3]	6[2]	6[1]	6[0]	5[6]	5[5]
8	7[6]	7[5]	7[4]	7[3]	7[2]	7[1]	7[0]	6[6]
9	9[0]	8[6]	8[5]	8[4]	8[3]	8[2]	8[1]	8[0]
10	10[1]	10[0]	9[6]	9[5]	9[4]	9[3]	9[2]	9[1]
11	11[2]	11[1]	11[0]	10[6]	10[5]	10[4]	10[3]	10[2]
12	12[3]	12[2]	12[1]	12[0]	11[6]	11[5]	11[4]	11[3]
13	13[4]	13[3]	13[2]	13[1]	13[0]	12[6]	12[5]	12[4]
14	14[5]	14[4]	14[3]	14[2]	14[1]	14[0]	13[6]	13[5]
15	15[6]	15[5]	15[4]	15[3]	15[2]	15[1]	15[0]	14[6]

16.2.2.2 Clocks

The following table describes the clock sources for the VPU G2 decoder. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 16-6. VPU G2 Clocks

Clock name	Description
core_clk	VPU G2 core clock

16.2.2.3 Reset

The VPU G2 decoder has one software reset (g2_sft_rstn).

16.2.3 External Signals

There are no external signals pinned out for this module.

16.2.4 Initialization and Application Information

The VPU G2 decoder is operated via the VPU Application Programming Interface (API). Details are provided in separate API documentation.

16.2.5 VPU G2 Memory Map/Register Definition

16.2.5.1 VPU_G2 register descriptions

16.2.5.1.1 VPU Memory map

VPU_G2 base address: 3831_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	ID register (read only) (SWREG0)	32	RO	Table 16-6
4h	Interrupt register decoder (SWREG1)	32	RW	0000_0000h
8h	Data configuration register decoder (SWREG2)	32	RW	0000_0000h
Ch	Decoder control register 0 (SWREG3)	32	RW	0000_0000h
10h	Decoder control register 1 (SWREG4)	32	RW	0000_0000h
14h	Decoder control register 2 (SWREG5)	32	RW	0000_0000h
18h	Decoder control register 3 (SWREG6)	32	RW	0000_0000h
1Ch	Decoder control register 4 (SWREG7)	32	RW	0000_0000h
20h	Decoder control register 5 (SWREG8)	32	RW	0000_0000h
24h	Decoder control register 6 (SWREG9)	32	RW	0000_0000h
28h	Decoder control register 7 (SWREG10)	32	RW	0000_0000h
2Ch	Decoder control register 8 (SWREG11)	32	RW	0000_0000h
30h	Decoder control register 9 (SWREG12)	32	RW	0000_0000h
34h	Decoder control register 10 (SWREG13)	32	RW	0000_0000h
38h	Initial ref pic list register (0-2) (SWREG14)	32	RW	0000_0000h
3Ch	Initial ref pic list register (3-5) (SWREG15)	32	RW	0000_0000h
40h	Initial ref pic list register (6-8) (SWREG16)	32	RW	0000_0000h
44h	Initial ref pic list register (9-11) (SWREG17)	32	RW	0000_0000h
48h	Initial ref pic list register (12-14) (SWREG18)	32	RW	0000_0000h
4Ch	Initial ref pic list register (15 and P 0-3) (SWREG19)	32	RW	0000_0000h
50h	Decoder control register 11 (SWREG20)	32	RW	0000_0000h
54h	Not used (SWREG21)	32	RU	0000_0000h
58h	Not used (SWREG22)	32	RU	0000_0000h
5Ch	Decoder configure status register (SWREG23)	32	RO	0000_0000h
60h	Not used (SWREG24)	32	RU	0000_0000h
64h	Not used (SWREG25)	32	RU	0000_0000h
68h	Not used (SWREG26)	32	RU	0000_0000h
6Ch	Not used (SWREG27)	32	RU	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
70h	Not used (SWREG28)	32	RU	0000_0000h
74h	Not used (SWREG29)	32	RU	0000_0000h
78h	Not used (SWREG30)	32	RU	0000_0000h
7Ch	VP9 segmentation values (SWREG31)	32	RW	0000_0000h
80h	VP9 segmentation values (SWREG32)	32	RW	0000_0000h
84h	VP9 reference picture scaling register 0 (SWREG33)	32	RW	0000_0000h
88h	VP9 reference picture scaling register 1 (SWREG34)	32	RW	0000_0000h
8Ch	VP9 reference picture scaling register 2 (SWREG35)	32	RW	0000_0000h
90h	VP9 reference picture scaling register 3 (SWREG36)	32	RW	0000_0000h
94h	VP9 reference picture scaling register 4 (SWREG37)	32	RW	0000_0000h
98h	VP9 reference picture scaling register 5 (SWREG38)	32	RW	0000_0000h
9Ch	Not used (SWREG39)	32	RU	0000_0000h
A0h	Not used (SWREG40)	32	RU	0000_0000h
A4h	Not used (SWREG41)	32	RU	0000_0000h
A8h	Not used (SWREG42)	32	RU	0000_0000h
ACh	Not used (SWREG43)	32	RU	0000_0000h
B0h	Not used (SWREG44)	32	RU	0000_0000h
B4h	Timeout control register (SWREG45)	32	RW	0000_0000h
B8h	Picture order count from current pictures for index 0-3 (SWREG46)	32	RW	0000_0000h
BCh	Picture order count from current pictures for index 4-7 (SWREG47)	32	RW	0000_0000h
C0h	Picture order count from current pictures for index 8-11 (SWREG48)	32	RW	0000_0000h
C4h	Picture order count from current pictures for index 12-15 (SWREG49)	32	RW	0000_0000h
C8h	Synthesis configuration register decoder 0 (read only) (SWREG50)	32	RO	Table 16-6
CCh	Reference picture buffer control register (SWREG51)	32	RU	0000_0000h
D0h	Reference picture buffer information register 1 (read only) (SWREG52)	32	RU	0000_0000h
D4h	Reference picture buffer information register 2 (read only) (SWREG53)	32	RU	0000_0000h
D8h	Synthesis configuration register decoder 1 (read only) (SWREG54)	32	RO	0000_0000h
DCh	Advanced prefetch control register (SWREG55)	32	RW	0000_0000h
E0h	Synthesis configuration register decoder 2 (read only) (SWREG56)	32	RO	0000_0000h
E4h	Decoder fuse register (read only) (SWREG57)	32	RU	0000_0000h
E8h	Device configuration register decoder 2 + Multi core control register (SWREG58)	32	RW	0000_0000h
ECh	Device configuration register AXI ID (SWREG59)	32	RW	0000_0000h
F0h	Synthesis configuration register decoder 3 for PP (read only) (SWREG60)	32	RO	0000_0000h
F4h	Not used (SWREG61)	32	RU	0000_0000h
F8h	HW proceed register (CU location) (SWREG62)	32	RW	0000_0000h
FCh	HW performance register (cycles running) (SWREG63)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
100h	Base address MSB (bits 63:32) for decoded luminance picture (SWREG64)	32	RW	0000_0000h
104h	Base address LSB (bits 31:0) for decoded luminance picture (SWREG65)	32	RW	0000_0000h
108h	Base address MSB (bits 63:32) for reference luminance picture index 0 (SWREG66)	32	RW	0000_0000h
10Ch	Base address LSB (bits 31:0) for reference luminance picture index 0 (SWREG67)	32	RW	0000_0000h
110h	Base address MSB (bits 63:32) for reference luminance picture index 1 (SWREG68)	32	RW	0000_0000h
114h	Base address LSB (bits 31:0) for reference luminance picture index 1 (SWREG69)	32	RW	0000_0000h
118h	Base address MSB (bits 63:32) for reference luminance picture index 2 (SWREG70)	32	RW	0000_0000h
11Ch	Base address LSB (bits 31:0) for reference luminance picture index 2 (SWREG71)	32	RW	0000_0000h
120h	Base address MSB (bits 63:32) for reference luminance picture index 3 (SWREG72)	32	RW	0000_0000h
124h	Base address LSB (bits 31:0) for reference luminance picture index 3 (SWREG73)	32	RW	0000_0000h
128h	Base address MSB (bits 63:32) for reference luminance picture index 4 (SWREG74)	32	RW	0000_0000h
12Ch	Base address LSB (bits 31:0) for reference luminance picture index 4 (SWREG75)	32	RW	0000_0000h
130h	Base address MSB (bits 63:32) for reference luminance picture index 5 (SWREG76)	32	RW	0000_0000h
134h	Base address LSB (bits 31:0) for reference luminance picture index 5 (SWREG77)	32	RW	0000_0000h
138h	Base address MSB (bits 63:32) for reference luminance picture index 6 /VP9 segment write base MSB (SWREG78)	32	RW	0000_0000h
13Ch	Base address LSB (bits 31:0) for reference luminance picture index 6 /VP9 segment write base LSB (SWREG79)	32	RW	0000_0000h
140h	Base address MSB (bits 63:32) for reference luminance picture index 7 /VP9 segment read base MSB (SWREG80)	32	RW	0000_0000h
144h	Base address LSB (bits 31:0) for reference luminance picture index 7 /VP9 segment read base LSB (SWREG81)	32	RW	0000_0000h
148h	Base address MSB (bits 63:32) for reference luminance picture index 8 (SWREG82)	32	RW	0000_0000h
14Ch	Base address LSB (bits 31:0) for reference luminance picture index 8 (SWREG83)	32	RW	0000_0000h
150h	Base address MSB (bits 63:32) for reference luminance picture index 9 (SWREG84)	32	RW	0000_0000h
154h	Base address LSB (bits 31:0) for reference luminance picture index 9 (SWREG85)	32	RW	0000_0000h
158h	Base address MSB (bits 63:32) for reference luminance picture index 10 (SWREG86)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
15Ch	Base address LSB (bits 31:0) for reference luminance picture index 10 (SWREG87)	32	RW	0000_0000h
160h	Base address MSB (bits 63:32) for reference luminance picture index 11 (SWREG88)	32	RW	0000_0000h
164h	Base address LSB (bits 31:0) for reference luminance picture index 11 (SWREG89)	32	RW	0000_0000h
168h	Base address MSB (bits 63:32) for reference luminance picture index 12 (SWREG90)	32	RW	0000_0000h
16Ch	Base address LSB (bits 31:0) for reference luminance picture index 12 (SWREG91)	32	RW	0000_0000h
170h	Base address MSB (bits 63:32) for reference luminance picture index 13 (SWREG92)	32	RW	0000_0000h
174h	Base address LSB (bits 31:0) for reference luminance picture index 13 (SWREG93)	32	RW	0000_0000h
178h	Base address MSB (bits 63:32) for reference luminance picture index 14 (SWREG94)	32	RW	0000_0000h
17Ch	Base address LSB (bits 31:0) for reference luminance picture index 14 (SWREG95)	32	RW	0000_0000h
180h	Base address MSB (bits 63:32) for reference luminance picture index 15 (SWREG96)	32	RW	0000_0000h
184h	Base address LSB (bits 31:0) for reference luminance picture index 15 (SWREG97)	32	RW	0000_0000h
188h	Base address MSB (bits 63:32) for decoded chrominance picture (SWREG98)	32	RW	0000_0000h
18Ch	Base address LSB (bits 31:0) for decoded chrominance picture (SWREG99)	32	RW	0000_0000h
190h	Base address MSB (bits 63:32) for reference chrominance picture index 0 (SWREG100)	32	RW	0000_0000h
194h	Base address LSB (bits 31:0) for reference chrominance picture index 0 (SWREG101)	32	RW	0000_0000h
198h	Base address MSB (bits 63:32) for reference chrominance picture index 1 (SWREG102)	32	RW	0000_0000h
19Ch	Base address LSB (bits 31:0) for reference chrominance picture index 1 (SWREG103)	32	RW	0000_0000h
1A0h	Base address MSB (bits 63:32) for reference chrominance picture index 2 (SWREG104)	32	RW	0000_0000h
1A4h	Base address LSB (bits 31:0) for reference chrominance picture index 2 (SWREG105)	32	RW	0000_0000h
1A8h	Base address MSB (bits 63:32) for reference chrominance picture index 3 (SWREG106)	32	RW	0000_0000h
1ACh	Base address LSB (bits 31:0) for reference chrominance picture index 3 (SWREG107)	32	RW	0000_0000h
1B0h	Base address MSB (bits 63:32) for reference chrominance picture index 4 (SWREG108)	32	RW	0000_0000h
1B4h	Base address LSB (bits 31:0) for reference chrominance picture index 4 (SWREG109)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1B8h	Base address MSB (bits 63:32) for reference chrominance picture index 5 (SWREG110)	32	RW	0000_0000h
1BCh	Base address LSB (bits 31:0) for reference chrominance picture index 5 (SWREG111)	32	RW	0000_0000h
1C0h	Base address MSB (bits 63:32) for reference chrominance picture index 6 (SWREG112)	32	RW	0000_0000h
1C4h	Base address LSB (bits 31:0) for reference chrominance picture index 6 (SWREG113)	32	RW	0000_0000h
1C8h	Base address MSB (bits 63:32) for reference chrominance picture index 7 (SWREG114)	32	RW	0000_0000h
1CCh	Base address LSB (bits 31:0) for reference chrominance picture index 7 (SWREG115)	32	RW	0000_0000h
1D0h	Base address MSB (bits 63:32) for reference chrominance picture index 8 (SWREG116)	32	RW	0000_0000h
1D4h	Base address LSB (bits 31:0) for reference chrominance picture index 8 (SWREG117)	32	RW	0000_0000h
1D8h	Base address MSB (bits 63:32) for reference chrominance picture index 9 (SWREG118)	32	RW	0000_0000h
1DCh	Base address LSB (bits 31:0) for reference chrominance picture index 9 (SWREG119)	32	RW	0000_0000h
1E0h	Base address MSB (bits 63:32) for reference chrominance picture index 10 (SWREG120)	32	RW	0000_0000h
1E4h	Base address LSB (bits 31:0) for reference chrominance picture index 10 (SWREG121)	32	RW	0000_0000h
1E8h	Base address MSB (bits 63:32) for reference chrominance picture index 11 (SWREG122)	32	RW	0000_0000h
1ECh	Base address LSB (bits 31:0) for reference chrominance picture index 11 (SWREG123)	32	RW	0000_0000h
1F0h	Base address MSB (bits 63:32) for reference chrominance picture index 12 (SWREG124)	32	RW	0000_0000h
1F4h	Base address LSB (bits 31:0) for reference chrominance picture index 12 (SWREG125)	32	RW	0000_0000h
1F8h	Base address MSB (bits 63:32) for reference chrominance picture index 13 (SWREG126)	32	RW	0000_0000h
1FCh	Base address LSB (bits 31:0) for reference chrominance picture index 13 (SWREG127)	32	RW	0000_0000h
200h	Base address MSB (bits 63:32) for reference chrominance picture index 14 (SWREG128)	32	RW	0000_0000h
204h	Base address LSB (bits 31:0) for reference chrominance picture index 14 (SWREG129)	32	RW	0000_0000h
208h	Base address MSB (bits 63:32) for reference chrominance picture index 15 (SWREG130)	32	RW	0000_0000h
20Ch	Base address LSB (bits 31:0) for reference chrominance picture index 15 (SWREG131)	32	RW	0000_0000h
210h	Base address MSB (bits 63:32) for decoded direct mode MVS (SWREG132)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
214h	Base address LSB (bits 31:0) for decoded direct mode MVS (SWREG133)	32	RW	0000_0000h
218h	Base address MSB (bits 63:32) for reference direct mode MVS index 0 (SWREG134)	32	RW	0000_0000h
21Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 0 (SWREG135)	32	RW	0000_0000h
220h	Base address MSB (bits 63:32) for reference direct mode MVS index 1 (SWREG136)	32	RW	0000_0000h
224h	Base address LSB (bits 31:0) for reference direct mode MVS index 1 (SWREG137)	32	RW	0000_0000h
228h	Base address MSB (bits 63:32) for reference direct mode MVS index 2 (SWREG138)	32	RW	0000_0000h
22Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 2 (SWREG139)	32	RW	0000_0000h
230h	Base address MSB (bits 63:32) for reference direct mode MVS index 3 (SWREG140)	32	RW	0000_0000h
234h	Base address LSB (bits 31:0) for reference direct mode MVS index 3 (SWREG141)	32	RW	0000_0000h
238h	Base address MSB (bits 63:32) for reference direct mode MVS index 4 (SWREG142)	32	RW	0000_0000h
23Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 4 (SWREG143)	32	RW	0000_0000h
240h	Base address MSB (bits 63:32) for reference direct mode MVS index 5 (SWREG144)	32	RW	0000_0000h
244h	Base address LSB (bits 31:0) for reference direct mode MVS index 5 (SWREG145)	32	RW	0000_0000h
248h	Base address MSB (bits 63:32) for reference direct mode MVS index 6 (SWREG146)	32	RW	0000_0000h
24Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 6 (SWREG147)	32	RW	0000_0000h
250h	Base address MSB (bits 63:32) for reference direct mode MVS index 7 (SWREG148)	32	RW	0000_0000h
254h	Base address LSB (bits 31:0) for reference direct mode MVS index 7 (SWREG149)	32	RW	0000_0000h
258h	Base address MSB (bits 63:32) for reference direct mode MVS index 8 (SWREG150)	32	RW	0000_0000h
25Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 8 (SWREG151)	32	RW	0000_0000h
260h	Base address MSB (bits 63:32) for reference direct mode mode MVS index 9 (SWREG152)	32	RW	0000_0000h
264h	Base address LSB (bits 31:0) for reference direct mode mode MVS index 9 (SWREG153)	32	RW	0000_0000h
268h	Base address MSB (bits 63:32) for reference direct mode MVS index 10 (SWREG154)	32	RW	0000_0000h
26Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 10 (SWREG155)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
270h	Base address MSB (bits 63:32) for reference direct mode MVS index 11 (SWREG156)	32	RW	0000_0000h
274h	Base address LSB (bits 31:0) for reference direct mode MVS index 11 (SWREG157)	32	RW	0000_0000h
278h	Base address MSB (bits 63:32) for reference direct mode MVS index 12 (SWREG158)	32	RW	0000_0000h
27Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 12 (SWREG159)	32	RW	0000_0000h
280h	Base address MSB (bits 63:32) for reference direct mode MVS index 13 (SWREG160)	32	RW	0000_0000h
284h	Base address LSB (bits 31:0) for reference direct mode MVS index 13 (SWREG161)	32	RW	0000_0000h
288h	Base address MSB (bits 63:32) for reference direct mode MVS index 14 (SWREG162)	32	RW	0000_0000h
28Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 14 (SWREG163)	32	RW	0000_0000h
290h	Base address MSB (bits 63:32) for reference direct mode MVS index 15 (SWREG164)	32	RW	0000_0000h
294h	Base address LSB (bits 31:0) for reference direct mode MVS index 15 (SWREG165)	32	RW	0000_0000h
298h	Base address MSB (bits 63:32) for tile sizes (SWREG166)	32	RW	0000_0000h
29Ch	Base address LSB (bits 31:0) for tile sizes (SWREG167)	32	RW	0000_0000h
2A0h	Base address MSB (bits 63:32) for / stream start address/decoded end addr register (SWREG168)	32	RW	0000_0000h
2A4h	Base address LSB (bits 31:0) for / stream start address/decoded end addr register (SWREG169)	32	RW	0000_0000h
2A8h	Base address MSB (bits 63:32) for scaling lists / VP9 CTX counter values (SWREG170)	32	RW	0000_0000h
2ACh	Base address LSB (bits 31:0) for scaling lists / VP9 CTX counter values (SWREG171)	32	RW	0000_0000h
2B0h	Base address MSB (bits 63:32) for stream propability tables (SWREG172)	32	RW	0000_0000h
2B4h	Base address LSB (bits 31:0) for stream propability tables (SWREG173)	32	RW	0000_0000h
2B8h	Base address MSB (bits 63:32) for decoder output raster scan Y picture (SWREG174)	32	RW	0000_0000h
2BCh	Base address LSB (bits 31:0) for decoder output raster scan Y picture (SWREG175)	32	RW	0000_0000h
2C0h	Base address MSB (bits 63:32) for decoder output raster scan C picture (SWREG176)	32	RW	0000_0000h
2C4h	Base address LSB (bits 31:0) for decoder output raster scan C picture (SWREG177)	32	RW	0000_0000h
2C8h	Base address MSB (bits 63:32) for tile border coeffients of filter (SWREG178)	32	RW	0000_0000h
2CCh	Base address LSB (bits 31:0) for tile border coeffients of filter (SWREG179)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
2D0h	Base address MSB (bits 63:32) for tile border coeffients of sao (SWREG180)	32	RW	0000_0000h
2D4h	Base address LSB (bits 31:0) for tile border coeffients of sao (SWREG181)	32	RW	0000_0000h
2D8h	Base address MSB (bits 63:32) for tile border bsd control data (SWREG182)	32	RW	0000_0000h
2DCh	Base address LSB (bits 31:0) for tile border bsd control data (SWREG183)	32	RW	0000_0000h
2E0h	Raster scan down scale control register MSM (SWREG184)	32	RW	0000_0000h
2E4h	Base address MSB (bits 63:32) for decoder output raster scan down scale Y picture (SWREG185)	32	RW	0000_0000h
2E8h	Base address LSB (bits 31:0) for decoder output raster scan down scale Y picture (SWREG186)	32	RW	0000_0000h
2ECh	Base address MSB (bits 63:32) for decoder output raster scan down scale C picture (SWREG187)	32	RW	0000_0000h
2F0h	Base address LSB (bits 31:0) for decoder output raster scan down scale C picture (SWREG188)	32	RW	0000_0000h
2F4h	Base address MSB (bits 63:32) for decoder output compress luminance table (SWREG189)	32	RW	0000_0000h
2F8h	Base address LSB (bits 31:0) for decoder output compress luminance table (SWREG190)	32	RW	0000_0000h
2FCh	Base address MSB (bits 63:32) for reference compress luminance table index 0 (SWREG191)	32	RW	0000_0000h
300h	Base address LSB (bits 31:0) for reference compress luminance table index 0 (SWREG192)	32	RW	0000_0000h
304h	Base address MSB (bits 63:32) for reference compress luminance table index 1 (SWREG193)	32	RW	0000_0000h
308h	Base address LSB (bits 31:0) for reference compress luminance table index 1 (SWREG194)	32	RW	0000_0000h
30Ch	Base address MSB (bits 63:32) for reference compress luminance table index 2 (SWREG195)	32	RW	0000_0000h
310h	Base address LSB (bits 31:0) for reference compress luminance table index 2 (SWREG196)	32	RW	0000_0000h
314h	Base address MSB (bits 63:32) for reference compress luminance table index 3 (SWREG197)	32	RW	0000_0000h
318h	Base address LSB (bits 31:0) for reference compress luminance table index 3 (SWREG198)	32	RW	0000_0000h
31Ch	Base address MSB (bits 63:32) for reference compress luminance table index 4 (SWREG199)	32	RW	0000_0000h
320h	Base address LSB (bits 31:0) for reference compress luminance table index 4 (SWREG200)	32	RW	0000_0000h
324h	Base address MSB (bits 63:32) for reference compress luminance table index 5 (SWREG201)	32	RW	0000_0000h
328h	Base address LSB (bits 31:0) for reference compress luminance table index 5 (SWREG202)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
32Ch	Base address MSB (bits 63:32) for reference compress luminance table index 6 (SWREG203)	32	RW	0000_0000h
330h	Base address LSB (bits 31:0) for reference compress luminance table index 6 (SWREG204)	32	RW	0000_0000h
334h	Base address MSB (bits 63:32) for reference compress luminance table index 7 (SWREG205)	32	RW	0000_0000h
338h	Base address LSB (bits 31:0) for reference compress luminance table index 7 (SWREG206)	32	RW	0000_0000h
33Ch	Base address MSB (bits 63:32) for reference compress luminance table index 8 (SWREG207)	32	RW	0000_0000h
340h	Base address LSB (bits 31:0) for reference compress luminance table index 8 (SWREG208)	32	RW	0000_0000h
344h	Base address MSB (bits 63:32) for reference compress luminance table index 9 (SWREG209)	32	RW	0000_0000h
348h	Base address LSB (bits 31:0) for reference compress luminance table index 9 (SWREG210)	32	RW	0000_0000h
34Ch	Base address MSB (bits 63:32) for reference compress luminance table index 10 (SWREG211)	32	RW	0000_0000h
350h	Base address LSB (bits 31:0) for reference compress luminance table index 10 (SWREG212)	32	RW	0000_0000h
354h	Base address MSB (bits 63:32) for reference compress luminance table index 11 (SWREG213)	32	RW	0000_0000h
358h	Base address LSB (bits 31:0) for reference compress luminance table index 11 (SWREG214)	32	RW	0000_0000h
35Ch	Base address MSB (bits 63:32) for reference compress luminance table index 12 (SWREG215)	32	RW	0000_0000h
360h	Base address LSB (bits 31:0) for reference compress luminance table index 12 (SWREG216)	32	RW	0000_0000h
364h	Base address MSB (bits 63:32) for reference compress luminance table index 13 (SWREG217)	32	RW	0000_0000h
368h	Base address LSB (bits 31:0) for reference compress luminance table index 13 (SWREG218)	32	RW	0000_0000h
36Ch	Base address MSB (bits 63:32) for reference compress luminance table index 14 (SWREG219)	32	RW	0000_0000h
370h	Base address LSB (bits 31:0) for reference compress luminance table index 14 (SWREG220)	32	RW	0000_0000h
374h	Base address MSB (bits 63:32) for reference compress luminance table index 15 (SWREG221)	32	RW	0000_0000h
378h	Base address LSB (bits 31:0) for reference compress luminance table index 15 (SWREG222)	32	RW	0000_0000h
37Ch	Base address MSB (bits 63:32) for decoder output compress chrominance table (SWREG223)	32	RW	0000_0000h
380h	Base address LSB (bits 31:0) for decoder output compress chrominance table (SWREG224)	32	RW	0000_0000h
384h	Base address MSB (bits 63:32) for reference compress chrominance table index 0 (SWREG225)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
388h	Base address LSB (bits 31:0) for reference compress chrominance table index 0 (SWREG226)	32	RW	0000_0000h
38Ch	Base address MSB (bits 63:32) for reference compress chrominance table index 1 (SWREG227)	32	RW	0000_0000h
390h	Base address LSB (bits 31:0) for reference compress chrominance table index 1 (SWREG228)	32	RW	0000_0000h
394h	Base address MSB (bits 63:32) for reference compress chrominance table index 2 (SWREG229)	32	RW	0000_0000h
398h	Base address LSB (bits 31:0) for reference compress chrominance table index 2 (SWREG230)	32	RW	0000_0000h
39Ch	Base address MSB (bits 63:32) for reference compress chrominance table index 3 (SWREG231)	32	RW	0000_0000h
3A0h	Base address LSB (bits 31:0) for reference compress chrominance table index 3 (SWREG232)	32	RW	0000_0000h
3A4h	Base address MSB (bits 63:32) for reference compress chrominance table index 4 (SWREG233)	32	RW	0000_0000h
3A8h	Base address LSB (bits 31:0) for reference compress chrominance table index 4 (SWREG234)	32	RW	0000_0000h
3ACh	Base address MSB (bits 63:32) for reference compress chrominance table index 5 (SWREG235)	32	RW	0000_0000h
3B0h	Base address LSB (bits 31:0) for reference compress chrominance table index 5 (SWREG236)	32	RW	0000_0000h
3B4h	Base address MSB (bits 63:32) for reference compress chrominance table index 6 (SWREG237)	32	RW	0000_0000h
3B8h	Base address LSB (bits 31:0) for reference compress chrominance table index 6 (SWREG238)	32	RW	0000_0000h
3BCh	Base address MSB (bits 63:32) for reference compress chrominance table index 7 (SWREG239)	32	RW	0000_0000h
3C0h	Base address LSB (bits 31:0) for reference compress chrominance table index 7 (SWREG240)	32	RW	0000_0000h
3C4h	Base address MSB (bits 63:32) for reference compress chrominance table index 8 (SWREG241)	32	RW	0000_0000h
3C8h	Base address LSB (bits 31:0) for reference compress chrominance table index 8 (SWREG242)	32	RW	0000_0000h
3CCh	Base address MSB (bits 63:32) for reference compress chrominance table index 9 (SWREG243)	32	RW	0000_0000h
3D0h	Base address LSB (bits 31:0) for reference compress chrominance table index 9 (SWREG244)	32	RW	0000_0000h
3D4h	Base address MSB (bits 63:32) for reference compress chrominance table index 10 (SWREG245)	32	RW	0000_0000h
3D8h	Base address LSB (bits 31:0) for reference compress chrominance table index 10 (SWREG246)	32	RW	0000_0000h
3DCh	Base address MSB (bits 63:32) for reference compress chrominance table index 11 (SWREG247)	32	RW	0000_0000h
3E0h	Base address LSB (bits 31:0) for reference compress chrominance table index 11 (SWREG248)	32	RW	0000_0000h

Table continues on the next page...

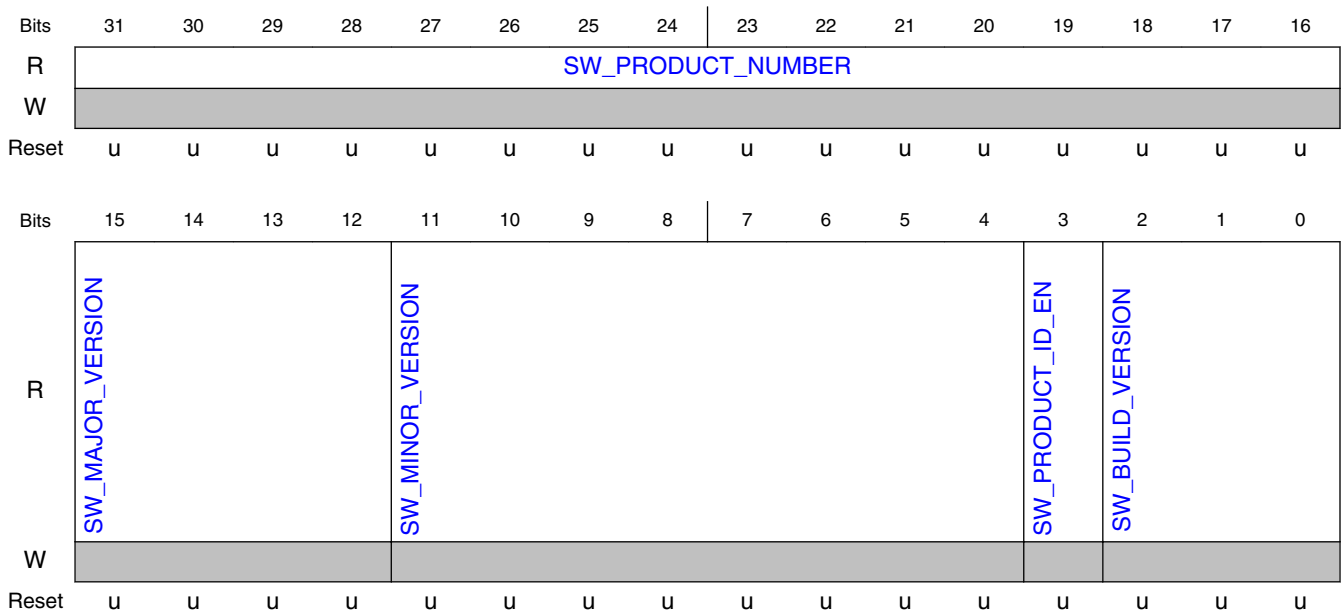
Offset	Register	Width (In bits)	Access	Reset value
3E4h	Base address MSB (bits 63:32) for reference compress chrominance table index 12 (SWREG249)	32	RW	0000_0000h
3E8h	Base address LSB (bits 31:0) for reference compress chrominance table index 12 (SWREG250)	32	RW	0000_0000h
3ECh	Base address MSB (bits 63:32) for reference compress chrominance table index 13 (SWREG251)	32	RW	0000_0000h
3F0h	Base address LSB (bits 31:0) for reference compress chrominance table index 13 (SWREG252)	32	RW	0000_0000h
3F4h	Base address MSB (bits 63:32) for reference compress chrominance table index 14 (SWREG253)	32	RW	0000_0000h
3F8h	Base address LSB (bits 31:0) for reference compress chrominance table index 14 (SWREG254)	32	RW	0000_0000h
3FCh	Base address MSB (bits 63:32) for reference compress chrominance table index 15 (SWREG255)	32	RW	0000_0000h
400h	Base address LSB (bits 31:0) for reference compress chrominance table index 15 (SWREG256)	32	RW	0000_0000h
404h	Not used (SWREG257)	32	RU	0000_0000h
408h	input stream buffer length (SWREG258)	32	RW	0000_0000h
40Ch	input stream buffer start offset (SWREG259)	32	RW	0000_0000h

16.2.5.1.2 ID register (read only) (SWREG0)

16.2.5.1.2.1 Offset

Register	Offset
SWREG0	0h

16.2.5.1.2.2 Diagram



16.2.5.1.2.3 Fields

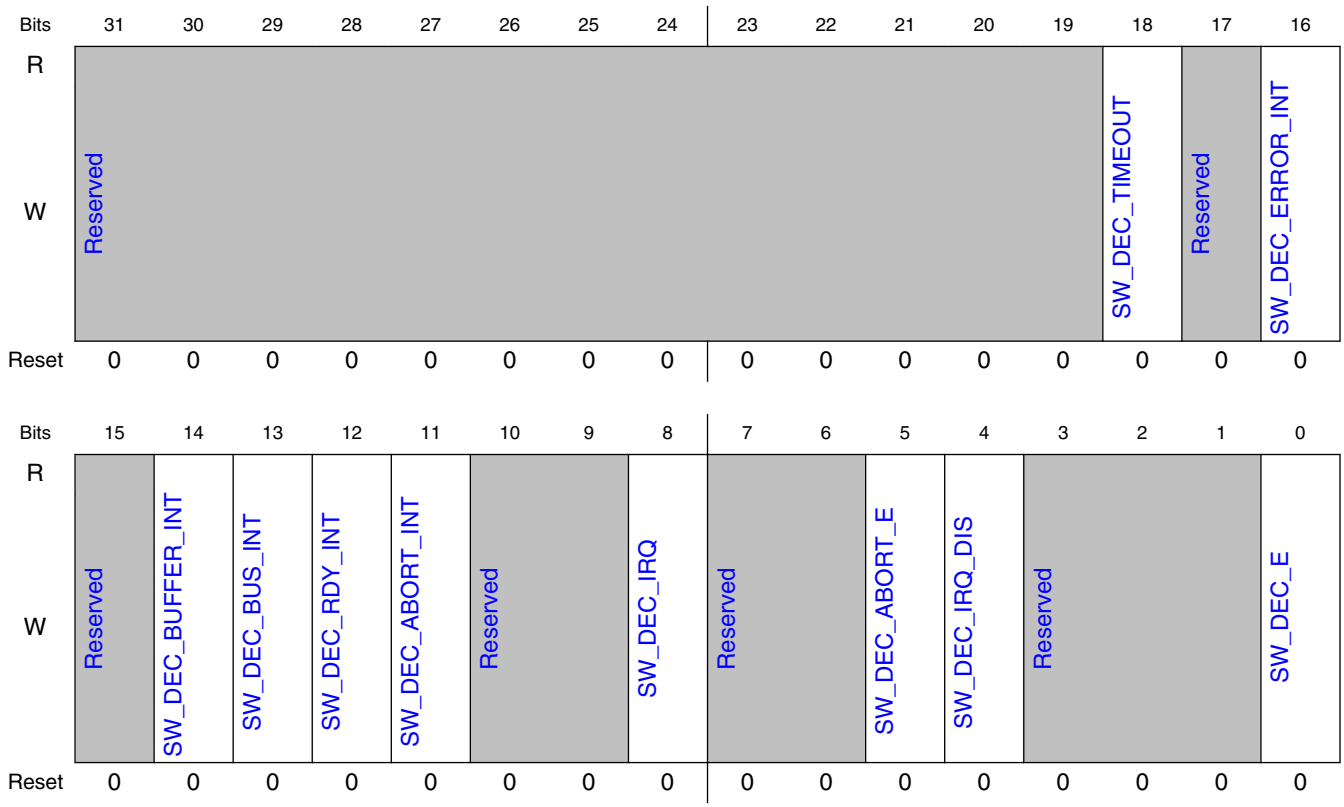
Field	Function
31-16 SW_PRODUCT_NUMBER	Product number (g2)
15-12 SW_MAJOR_VERSION	Major version
11-4 SW_MINOR_VERSION	Minor version
3 SW_PRODUCT_ID_EN	ASCII type product ID enable
2-0 SW_BUILD_VERSION	Build version (core number)

16.2.5.1.3 Interrupt register decoder (SWREG1)

16.2.5.1.3.1 Offset

Register	Offset
SWREG1	4h

16.2.5.1.3.2 Diagram



16.2.5.1.3.3 Fields

Field	Function
31-19 —	Reserved.
18 SW_DEC_TIMEOUT	Interrupt status bit decoder timeout. When high the decoder has been idling for too long. HW will self reset. Possible only if timeout interrupt is enabled
17 —	Reserved.
16	Interrupt status bit input stream error. When high an error is found in input data stream decoding. HW will self reset.

Table continues on the next page...

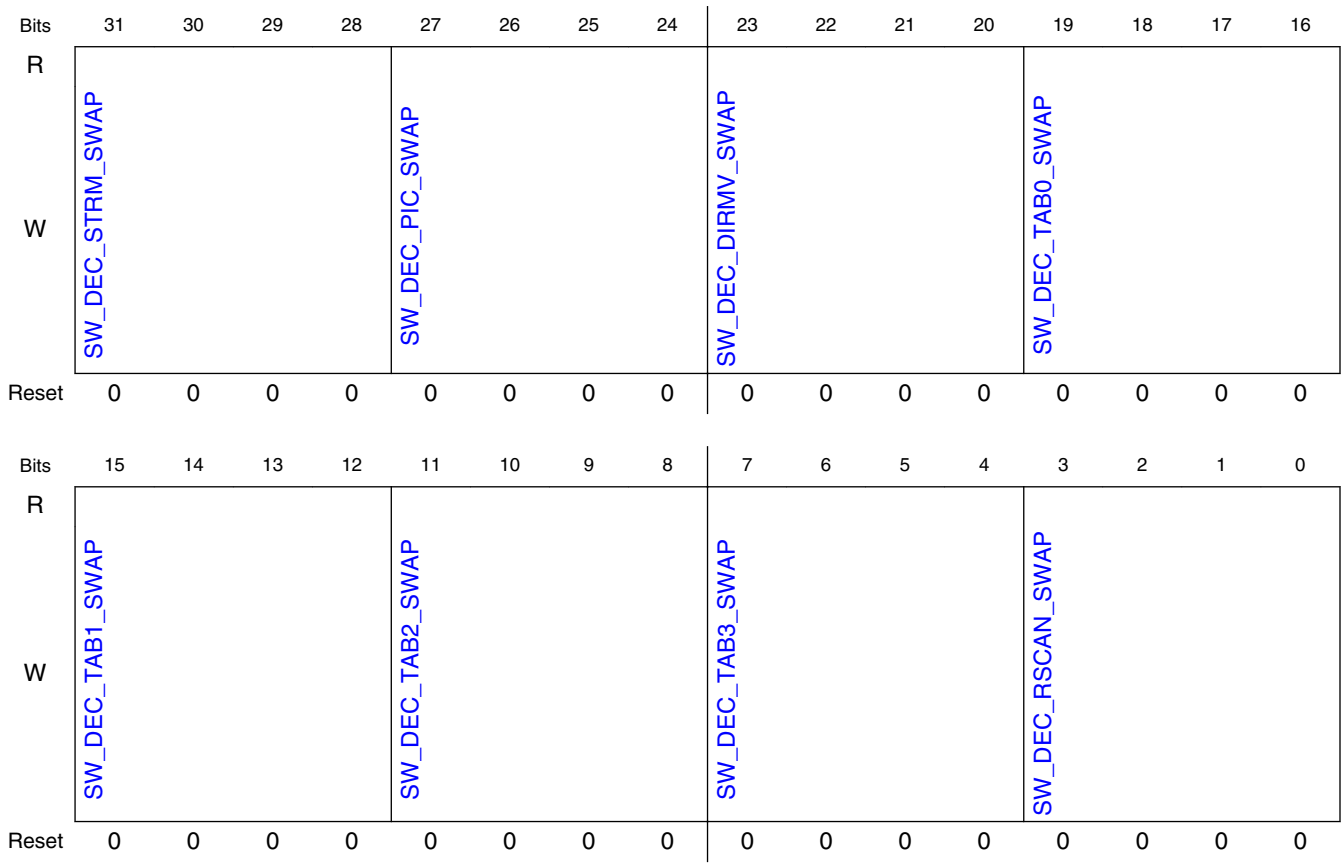
Field	Function
SW_DEC_ERROR_INT	
15 —	Reserved.
14 SW_DEC_BUFFER_INT	Interrupt status bit input buffer empty. When high input stream buffer is empty but picture is not ready. HW will not self reset.
13 SW_DEC_BUS_INT	Interrupt status bit bus. Error response from bus. HW will self reset.
12 SW_DEC_RDY_INT	Interrupt status bit decoder. When this bit is high decoder has decoded a picture. HW will self reset.
11 SW_DEC_ABORT_INT	Interrupt status bit decoding aborted. When this bit is high decoder has aborted the current picture decoding as SW requested (sw_dec_abort_e). Decoder self reset and sw_dec_abort_e written low
10-9 —	Reserved.
8 SW_DEC_IRQ	Decoder IRQ. When high decoder requests an interrupt. SW will reset this after interrupt is handled.
7-6 —	Reserved.
5 SW_DEC_ABORT_E	Abort decoding enable. Setting this bit high will cause HW to abort decoding and safely to reset itself down. After abort is complete the corresponding interrupt status is set and this bit is set low as well as the decoder enable.
4 SW_DEC_IRQ_DIS	Decoder IRQ disable. When high there are no interrupts concerning decoder from HW. Polling must be used to see the interrupt statuses.
3-1 —	Reserved.
0 SW_DEC_E	Decoder enable. Setting this bit high will start the decoding operation. HW will reset this when picture is processed or ASO or stream error is detected or bus error or timeout interrupt is given.

16.2.5.1.4 Data configuration register decoder (SWREG2)

16.2.5.1.4.1 Offset

Register	Offset
SWREG2	8h

16.2.5.1.4.2 Diagram



16.2.5.1.4.3 Fields

Field	Function
31-28 SW_DEC_STRM_SWAP	Byte swap configuration for stream data 4 Bit byte order vector to control byte locations inside HW internal 128 bit data vector. For 64 and 32 bit external bus widths, the data is first gathered to 128 bit width and then bytes swapped accordingly: Bit 0: 8 bit swap Bit 1: 16 bit swap Bit 2: 32 bit swap Bit 3: 64 bit swap '0000' = 0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15 '0001' = 1-0-3-2-5-4-7-6... '0010' = 2-3-0-1-6-7-4-5... '0011' = 3-2-1-0-7-6-5-4... '0100' = 4-5-6-7-0-1-2-3... ... '0111' = 7-6-5-4-3-2-1-0

Table continues on the next page...

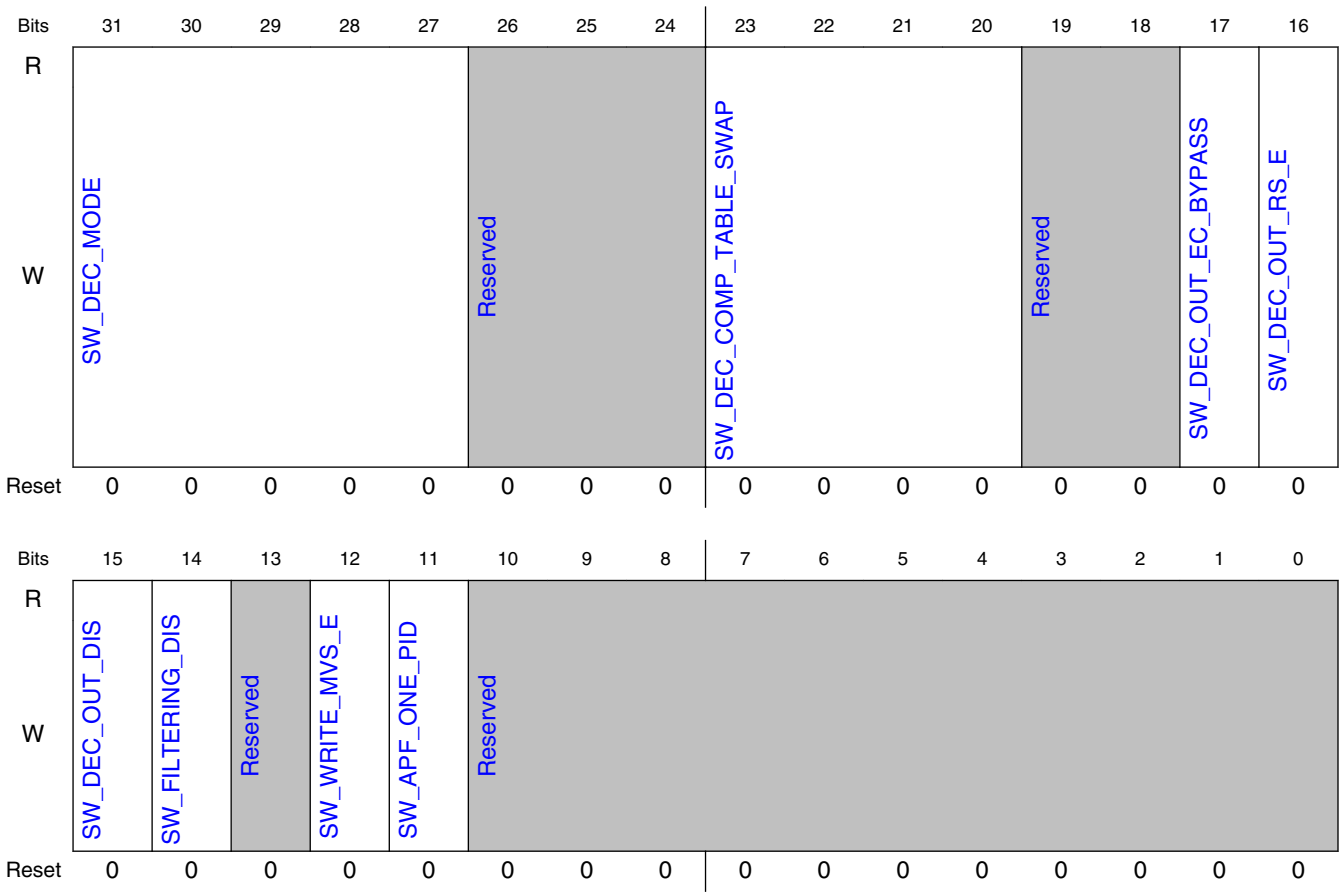
Field	Function
	'1000' = 8-9-10-11-12-13-14-15-0-1-2-3-4-5-6-7... ... '1111'=15-14-13-12-11-10-9-8-7-6-5-4-3-2-1-0
27-24 SW_DEC_PIC_SWAP	Byte swap configuration for decoder reference output picture data
23-20 SW_DEC_DIRMV_SWAP	Byte swap configuration for direct mode MV data (read/write)
19-16 SW_DEC_TAB0_SWAP	Byte swap configuration for VP9 stream propability tables
15-12 SW_DEC_TAB1_SWAP	Byte swap configuration for HEVC scaling lists / VP9 segmentation map read/write
11-8 SW_DEC_TAB2_SWAP	Byte swap configuration for VP9 CTX counter values
7-4 SW_DEC_TAB3_SWAP	Byte swap configuration for tile sizes
3-0 SW_DEC_RSCAN_SWAP	Byte swap for raster scan output picture data

16.2.5.1.5 Decoder control register 0 (SWREG3)

16.2.5.1.5.1 Offset

Register	Offset
SWREG3	Ch

16.2.5.1.5.2 Diagram



16.2.5.1.5.3 Fields

Field	Function
31-27 SW_DEC_MODE	Decoding mode: 00000-01011b - Reserved 01100b - HEVC 01101b - VP9 01110-11111b - Reserved
26-24 —	Reserved.
23-20 SW_DEC_COMP_TABLE_SWAP	Byte swap configuration for compress table data
19-18 —	Reserved.
17	Compress bypass

Table continues on the next page...

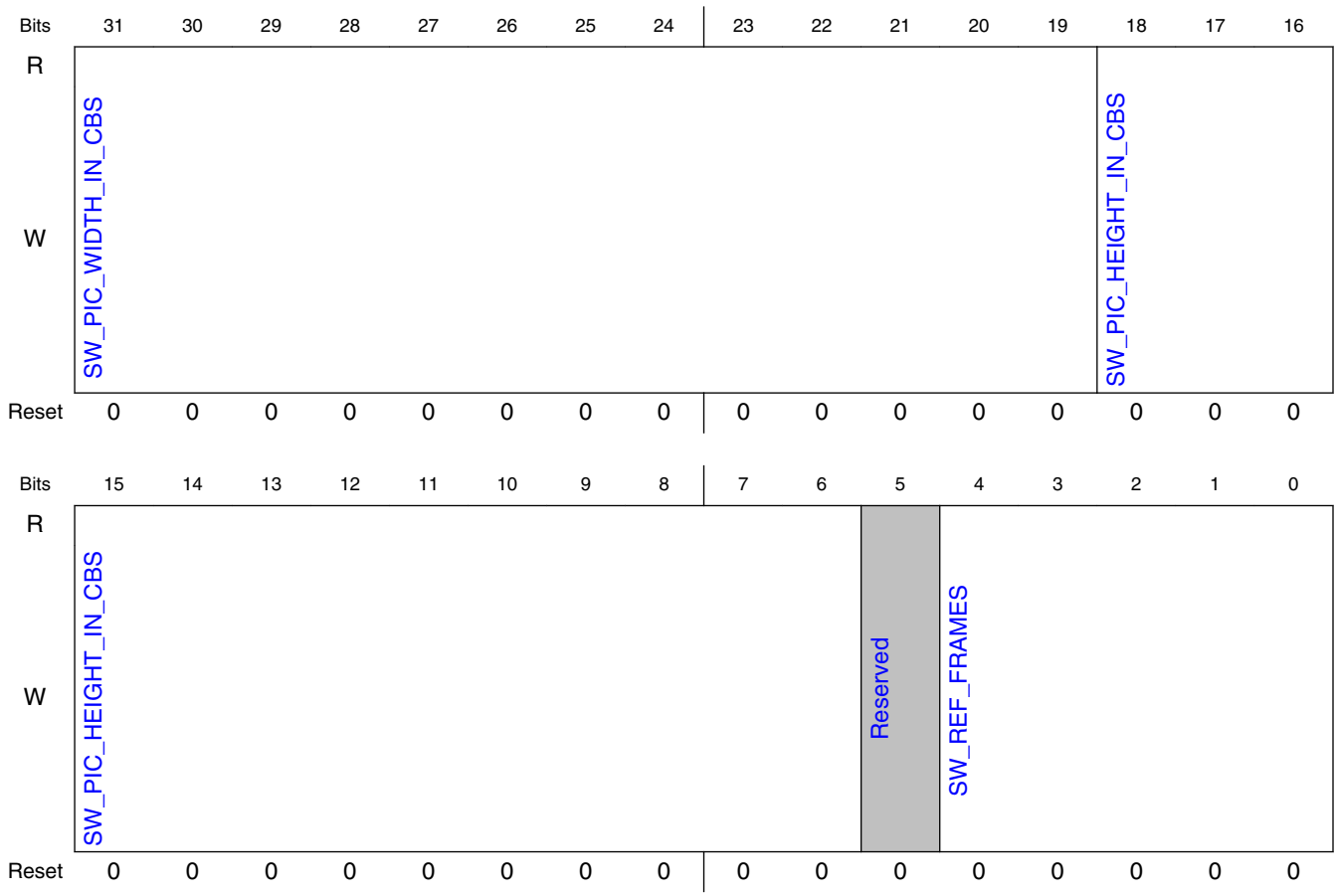
Field	Function
SW_DEC_OUT_EC_BYPASS	
16 SW_DEC_OUT_RS_E	Raster scan output enable. If high decoder writes the raster scan output if the configuration of Decoder includes PP raster scan output
15 SW_DEC_OUT_DIS	Disable decoder output picture writing 0b - Decoder output picture is written to external memory 1b - Decoder output picture is not written to external memory
14 SW_FILTERING_DIS	De-block filtering disable 0b - Filtering is enabled for current picture 1b - Filtering is disabled for current picture
13 —	Reserved.
12 SW_WRITE_MVS_E	Direct mode motion vector write enable for current picture 0b - Writing disabled for current picture. 1b - The direct mode motion vectors are written to external memory. HEVC/VP9 direct mode motion vectors are written to DPB aside with the corresponding reference picture. Other decoding mode dir mode mvs are written to external memory starting from sw_dir_mv_base.
11 SW_APF_ONE_PID	Prefetch partitions that have the same pic_id together
10-0 —	Reserved.

16.2.5.1.6 Decoder control register 1 (SWREG4)

16.2.5.1.6.1 Offset

Register	Offset
SWREG4	10h

16.2.5.1.6.2 Diagram



16.2.5.1.6.3 Fields

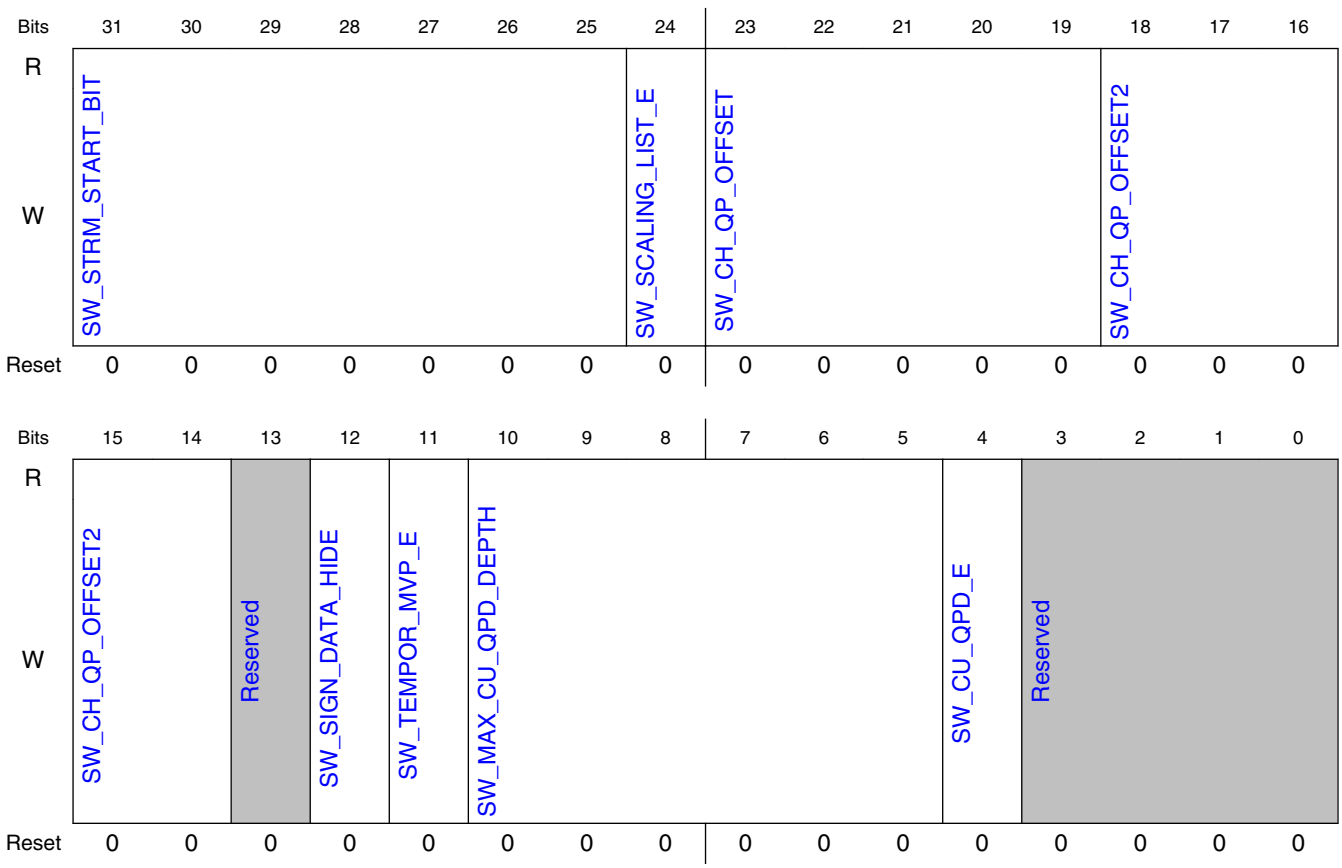
Field	Function
31-19 SW_PIC_WIDTH_IN_CBS	Picture width in min coded blocks (min = 8pix)
18-6 SW_PIC_HEIGHT_IN_CBS	Picture height in min coded blocks (min = 8pix)
5 —	Reserved.
4-0 SW_REF_FRAMES	HEVC: num_ref_frames maximum number of short and long term reference frames in decoded picture buffer

16.2.5.1.7 Decoder control register 2 (SWREG5)

16.2.5.1.7.1 Offset

Register	Offset
SWREG5	14h

16.2.5.1.7.2 Diagram



16.2.5.1.7.3 Fields

Field	Function
31-25 SW_STRM_START_BIT	Exact bit of stream start word where decoding can be started (assosiates with sw_rlc_vlc_base)
24	Scaling matrix enable 0b - Normal transform

Table continues on the next page...

Field	Function
SW_SCALING_LIST_E	1b - Use scaling matrix for transform (read from external memory)
23-19 SW_CH_QP_OFFSET	Chroma Qp filter offset. (For HEVC this offset concerns Cb only)
18-14 SW_CH_QP_OFFSET2	Chroma Qp filter offset for cr type
13 —	Reserved.
12 SW_SIGN_DATA_HIDE	Flag for stream decoding
11 SW_TEMPOR_MVP_E	Temporal mvp enable
10-5 SW_MAX_CU_QPD_DEPTH	Max CU qp delta depth
4 SW_CU_QPD_E	CU qp delta enable
3-0 —	Reserved.

16.2.5.1.8 Decoder control register 3 (SWREG6)

16.2.5.1.8.1 Offset

Register	Offset
SWREG6	18h

16.2.5.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_STREAM_LEN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_STREAM_LEN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.8.3 Fields

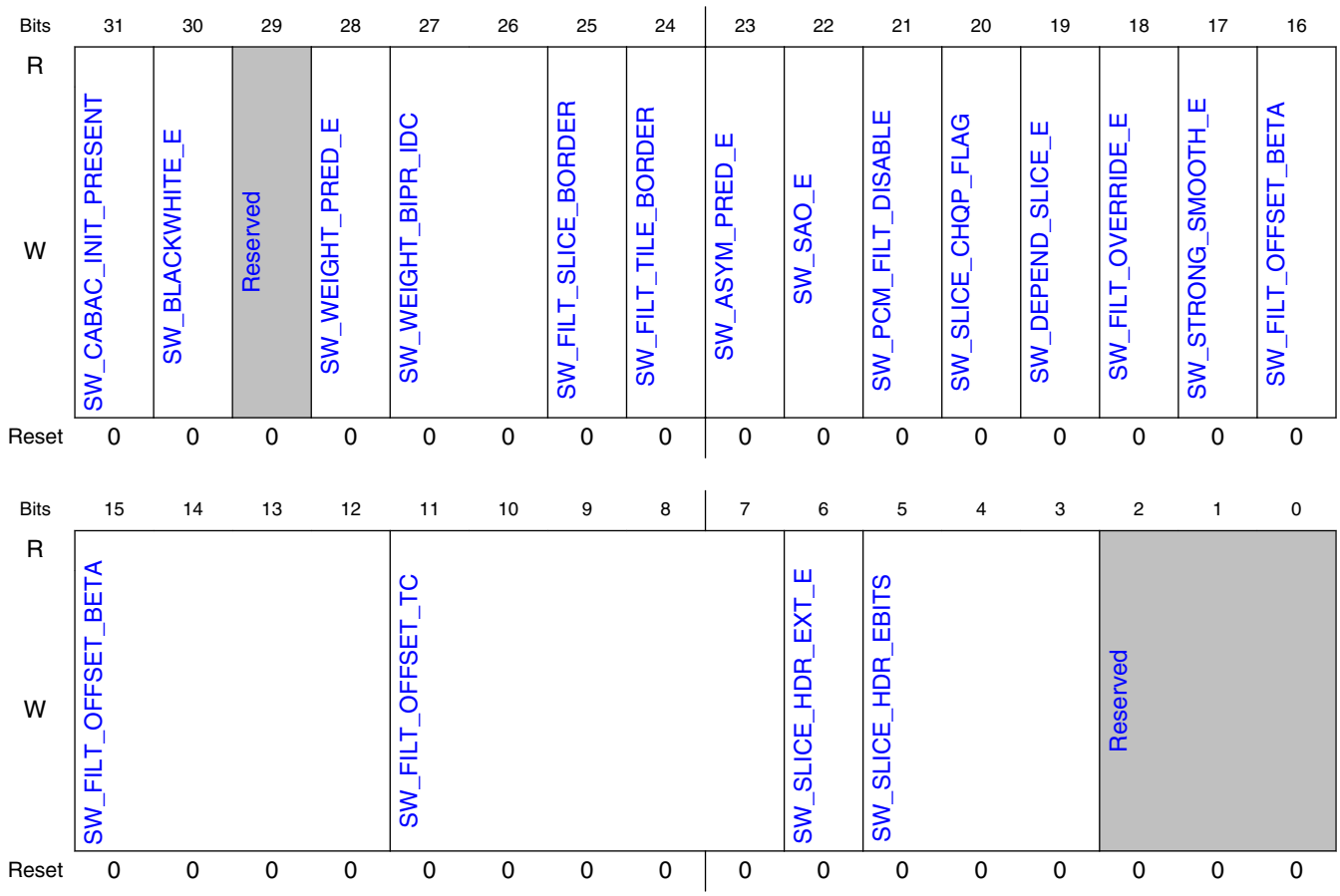
Field	Function
31-0 SW_STREAM_LEN	Amount of stream data bytes in input buffer. If the given buffer size is not enough for finishing the picture the corresponding interrupt is given and new stream buffer base address and stream buffer size information should be given (assosiates with sw_rlc_vlc_base). For HEVC the buffer must include at least data for one slice/VP of the picture

16.2.5.1.9 Decoder control register 4 (SWREG7)

16.2.5.1.9.1 Offset

Register	Offset
SWREG7	1Ch

16.2.5.1.9.2 Diagram



16.2.5.1.9.3 Fields

Field	Function
31 SW_CABAC_INIT_PRESENT	CABAC init present enable for stream decoding
30 SW_BLACKWHITE_E	Sampling 0b - 4:2:0 sampling format 1b - 4:0:0 sampling format (H264 monochroma)
29 —	Reserved.
28 SW_WEIGHT_PRED_E	Weighted prediction enable for P slices
27-26 SW_WEIGHT_BIPR_IDC	Weighted prediction specification 00b - Default weighted prediction is applied to B slices 01b - Explicit weighted prediction shall be applied to B slices 10b - NA

Table continues on the next page...

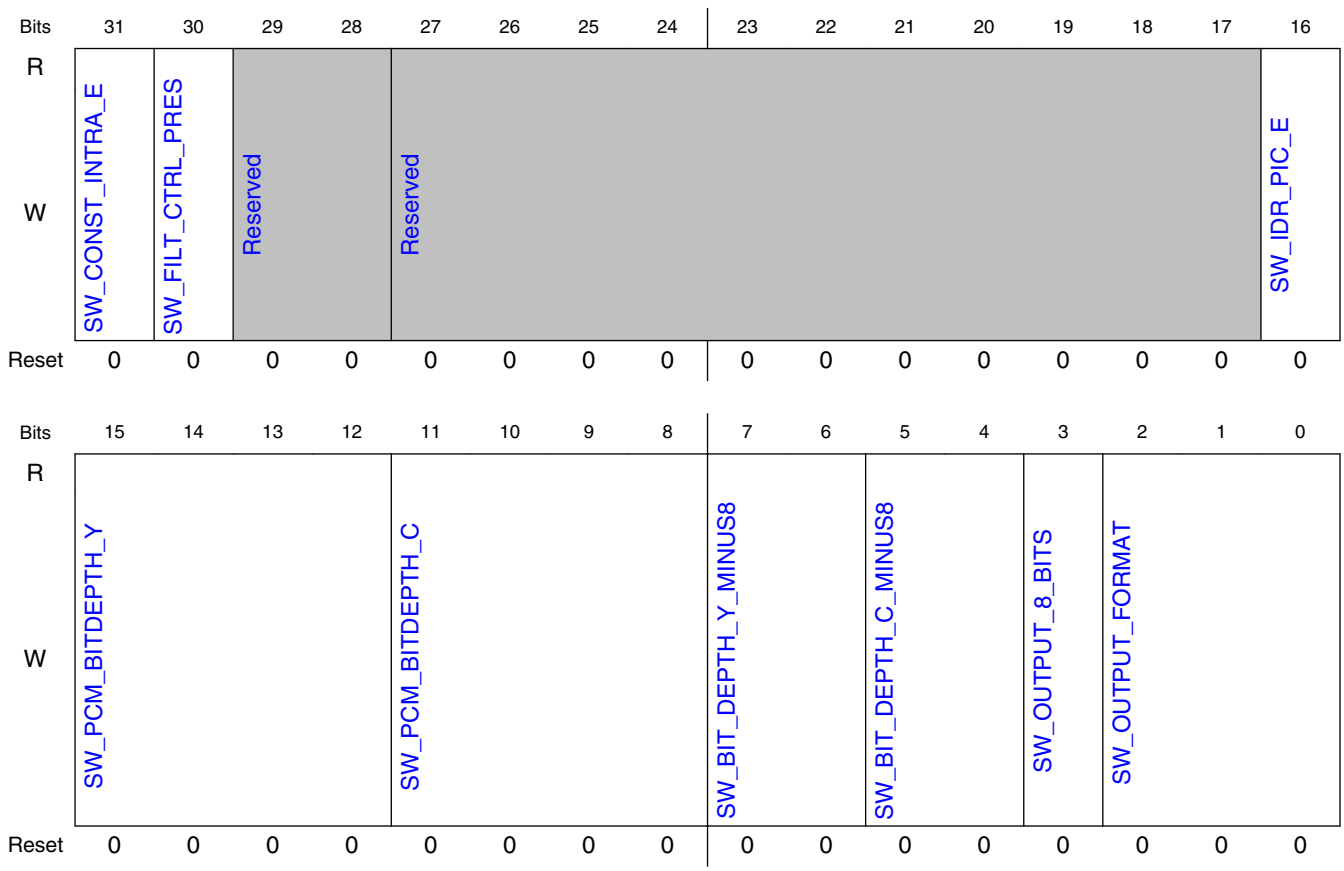
Field	Function
	11b - NA
25 SW_FILT_SLICE_BORDER	Filter enable over slice border
24 SW_FILT_TILE_BORDER	Filter enable over tile border
23 SW_ASYM_PRED_E	Asymmetric prediction flag for stream decoding
22 SW_SAO_E	Sample Adaptive Offset enable for stream decoding
21 SW_PCM_FILT_DISABLE	Disable for PCM loop filtering
20 SW_SLICE_CHQP_FLAG	Slice header flag for chroma QP present (if it is included in slice header)
19 SW_DEPEND_SLICE_E	Dependent slice enable
18 SW_FILT_OVERRIDE_E	Filter override enable
17 SW_STRONG_SMOOTH_E	Strong smoothing enable
16-12 SW_FILT_OFFSET_BETA	Filter beta offset (declared as div2)
11-7 SW_FILT_OFFSET_TC	Filter tc offset (declared as div2)
6 SW_SLICE_HDR_EXT_E	Slice header extension enable. Reserved for future use
5-3 SW_SLICE_HDR_EBITS	Number of extra slice header bits (if enabled slice header extension)
2-0 —	Reserved.

16.2.5.1.10 Decoder control register 5 (SWREG8)

16.2.5.1.10.1 Offset

Register	Offset
SWREG8	20h

16.2.5.1.10.2 Diagram



16.2.5.1.10.3 Fields

Field	Function
31 SW_CONST_INTRA_E	constrained_intra_pred_flag equal to 1 specifies that intra prediction uses only neighbouring intra macroblocks in prediction. When equal to 0 also neighbouring inter macroblocks are used in intra prediction process.
30	deblocking_filter_control_present_flag indicates whether extra variables controlling characteristics of the deblocking filter are present in the slice header.

Table continues on the next page...

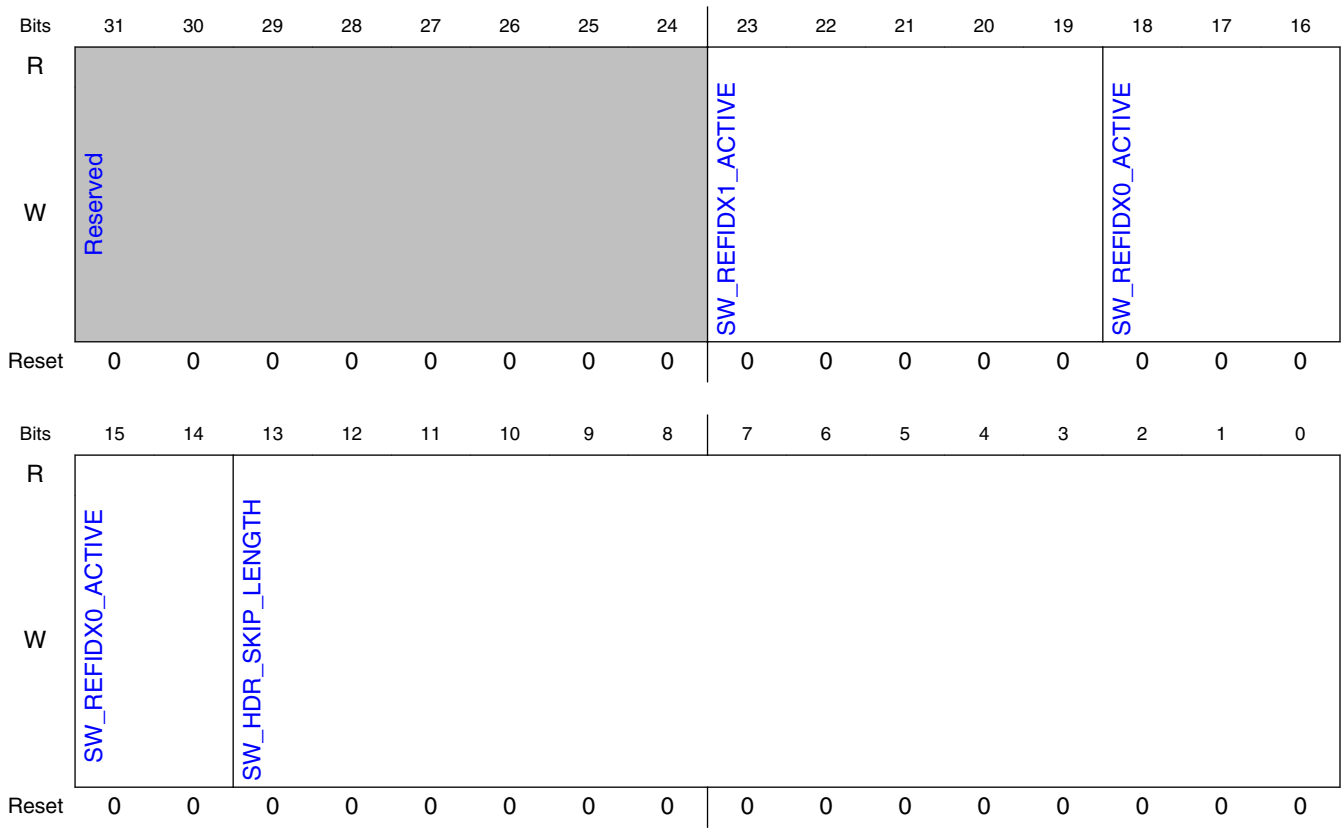
Field	Function
SW_FILT_CTRL_PRES	
29-28 —	Reserved.
27-17 —	Reserved.
16 SW_IDR_PIC_E	IDR (instantaneous decoding refresh) picture flag.
15-12 SW_PCM_BITDEPTH_Y	Bit depth for PCM Y data
11-8 SW_PCM_BITDEPTH_C	Bit depth for PCM C data
7-6 SW_BIT_DEPTH_Y_MINUS8	Bit depth of luma samples minus 8
5-4 SW_BIT_DEPTH_C_MINUS8	Bit depth of chroma samples minus 8
3 SW_OUTPUT_8_BITS	enable rasterscan output force to 8 bit(only for hevc main10 and vp9 10bit)
2-0 SW_OUTPUT_FORMAT	Raster scan and down scale output data format 000b - Each pixel in 10 bits when luma or chroma pixel bit depth is larger than 8; or 8 bits when both luma and chroma pixel bit depth are 8 bits. (default) 001b - Store in P010 format when luma or chroma pixel bit depth is larger than 8. 010b - A customized format: please refer to register SWREG23[6].

16.2.5.1.11 Decoder control register 6 (SWREG9)

16.2.5.1.11.1 Offset

Register	Offset
SWREG9	24h

16.2.5.1.11.2 Diagram



16.2.5.1.11.3 Fields

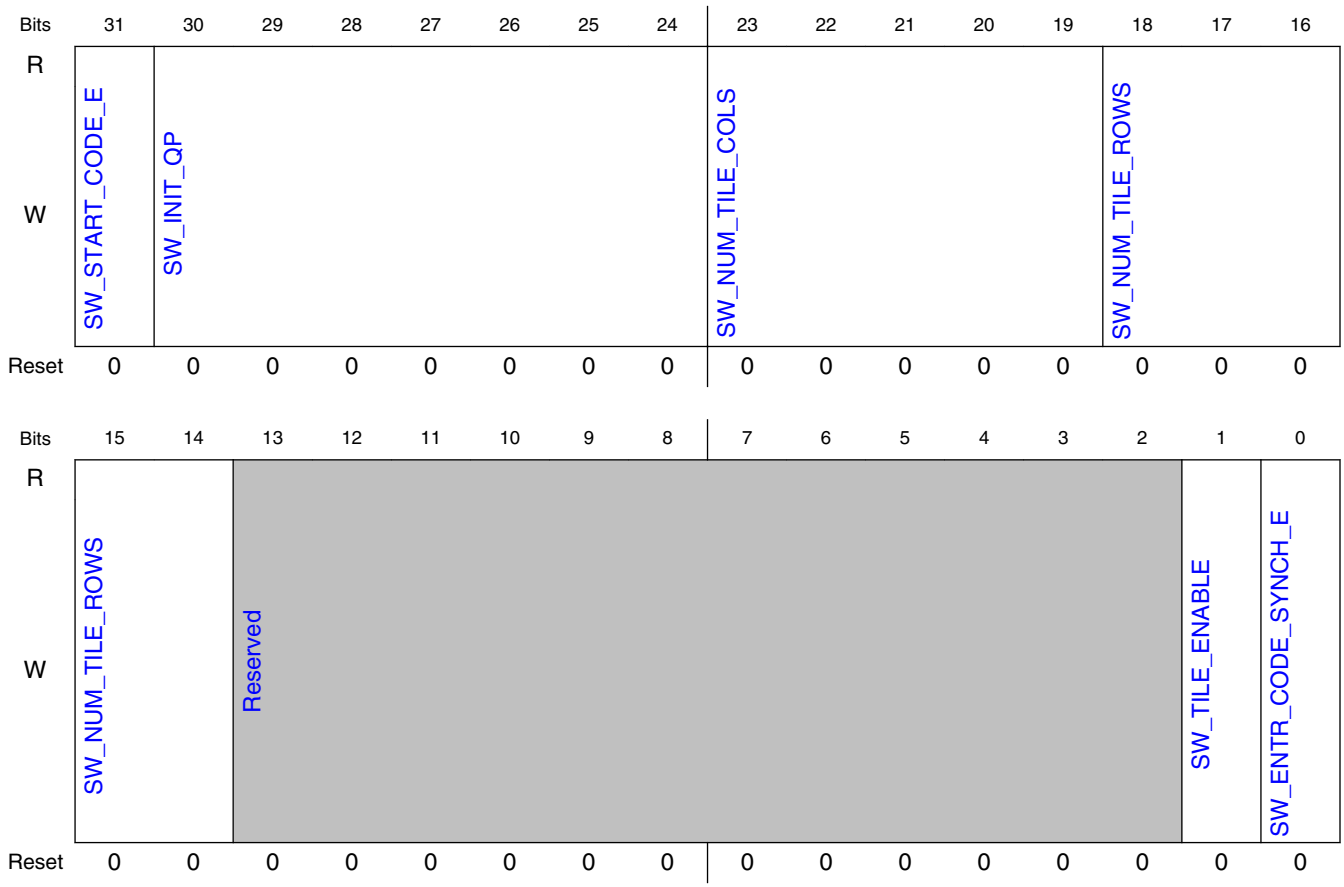
Field	Function
31-24 —	Reserved.
23-19 SW_REFIDX1_ACTIVE	Specifies the maximum reference index that can be used while decoding inter predicted macro blocks.
18-14 SW_REFIDX0_ACTIVE	Specifies the maximum reference index that can be used while decoding inter predicted macro blocks. This is same as in previous decoders (width increased with q bit)
13-0 SW_HDR_SKIP_LENGTH	Length of slice header skip length (bytes used by sw)

16.2.5.1.12 Decoder control register 7 (SWREG10)

16.2.5.1.12.1 Offset

Register	Offset
SWREG10	28h

16.2.5.1.12.2 Diagram



16.2.5.1.12.3 Fields

Field	Function
31 SW_START_CODE_E	Bit for indicating stream start code existence 0b - Stream does not contain start codes 1b - Stream contains start codes
30-24 SW_INIT_QP	Initial value for quantization parameter (picture quantizer).
23-19	Number of tile columns in picture

Table continues on the next page...

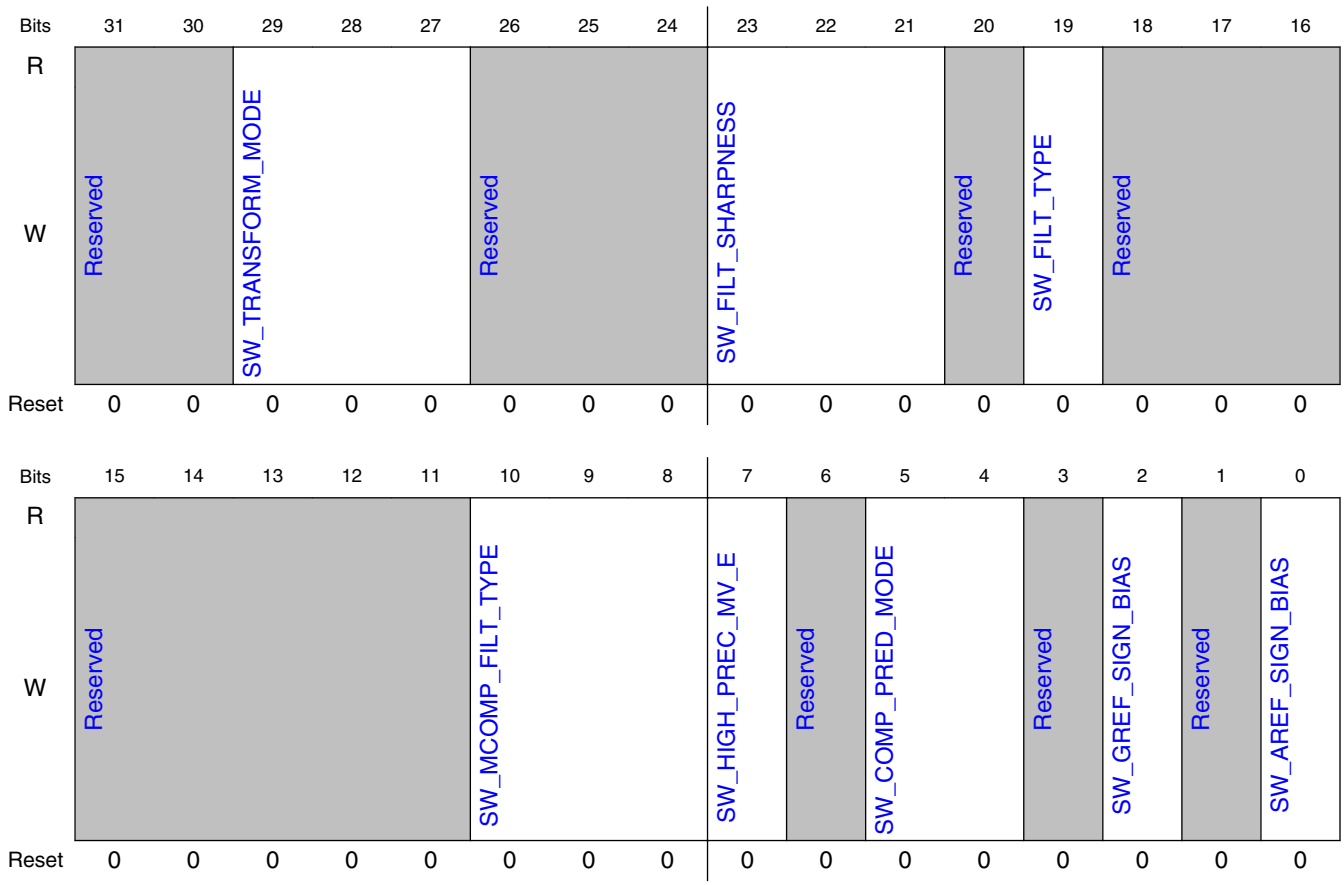
Field	Function
SW_NUM_TILE_COLS	
18-14 SW_NUM_TILE_ROWS	Number of tile rows in picture
13-2 —	Reserved.
1 SW_TILE_ENABLE	Tile enable
0 SW_ENTR_CODE_SYNCH_E	Entropy coding synchronization enable (Possible parallel cabac decoding)

16.2.5.1.13 Decoder control register 8 (SWREG11)

16.2.5.1.13.1 Offset

Register	Offset
SWREG11	2Ch

16.2.5.1.13.2 Diagram



16.2.5.1.13.3 Fields

Field	Function
31-30 —	Reserved.
29-27 SW_TRANSFORM_MODE	Transform modes 000b - 4x4 only 001b - Allow 8x8 010b - Allow 16x16 011b - Allow 32x32 100b - TX mode select
26-24 —	Reserved.
23-21 SW_FILT_SHARPNESS	Filter sharpness value
20 —	Reserved.

Table continues on the next page...

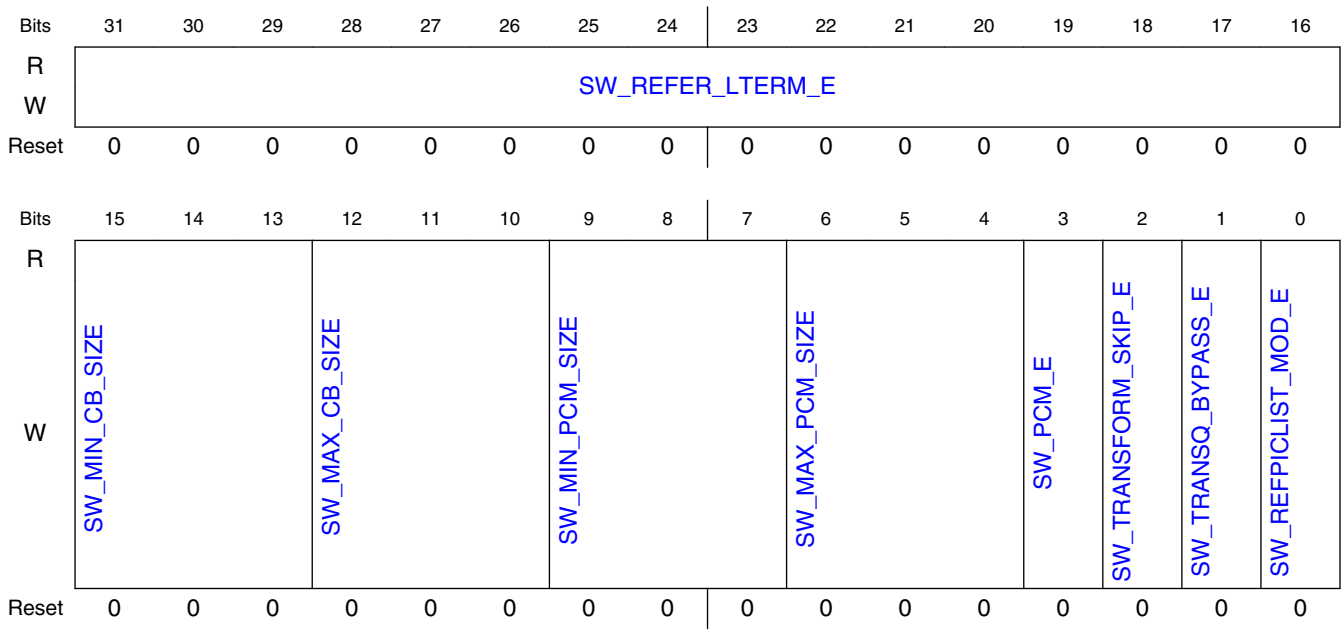
Field	Function
19 SW_FILT_TYPE	Filter Type
18-11 —	Reserved.
10-8 SW_MCOMP_FILT_TYPE	Inter prediction filter type to stream decoder 000b - Eight tap smooth 001b - Eight tap 010b - Eight tap sharp 011b - Bilinear 100b - Switchable
7 SW_HIGH_PREC_MV_E	High precision MV prediction enable
6 —	Reserved.
5-4 SW_COMP_PRED_MODE	Prediction Comp Type 00b - Single prediction only 01b - COMP prediction only 10b - Hybrid prediction
3 —	Reserved.
2 SW_GREF_SIGN_BIAS	Golden reference picture sign bias used for motion vector decoding
1 —	Reserved.
0 SW_AREF_SIGN_BIAS	Alternate reference picture sign bias used for motion vector decoding

16.2.5.1.14 Decoder control register 9 (SWREG12)

16.2.5.1.14.1 Offset

Register	Offset
SWREG12	30h

16.2.5.1.14.2 Diagram



16.2.5.1.14.3 Fields

Field	Function
31-16 SW_REFER_LTERM_E	Long term flag for reference picture index Definition: Bit 31 for picture index 0 Bit 30 for picture index 1 etc.
15-13 SW_MIN_CB_SIZE	CodedBlock min size (2^N): 011b - 8 pix 100b - 16 pix 101b - 32 pix 110b - 64 pix
12-10 SW_MAX_CB_SIZE	CodedBlock max size (2^N): 011b - 8 pix 100b - 16 pix 101b - 32 pix 110b - 64 pix
9-7 SW_MIN_PCM_SIZE	PCM min size (2^N): 011b - 8 pix 100b - 16 pix 101b - 32 pix 110b - 64 pix
6-4 SW_MAX_PCM_SIZE	PCM max size (2^N): 011b - 8 pix 100b - 16 pix 101b - 32 pix 110b - 64 pix
3 SW_PCM_E	IPCM MBs flag

Table continues on the next page...

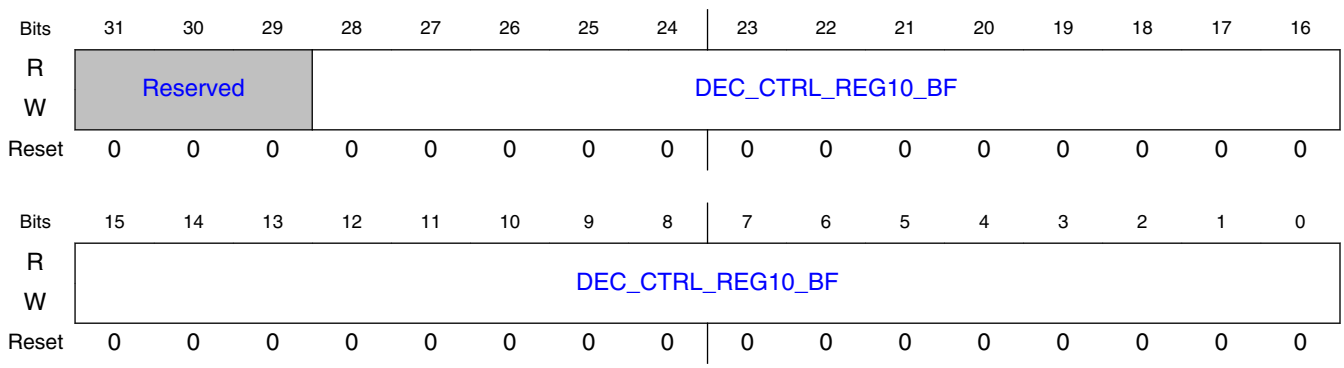
Field	Function
2 SW_TRANSFORM_SKIP_E	Transform skipping flag
1 SW_TRANSQ_BYPASS_E	Transform bypass flag (lossless mode)
0 SW_REFPICLIST_MOD_E	Refpic list reordering flag

16.2.5.1.15 Decoder control register 10 (SWREG13)

16.2.5.1.15.1 Offset

Register	Offset
SWREG13	34h

16.2.5.1.15.2 Diagram



16.2.5.1.15.3 Fields

Field	Function
31-29 —	Reserved.
28-0 DEC_CTRL_REG10_BF	For HEVC: [31:16] - Reserved - Not used [15:13] - sw_min_trb_size - Transform Block min size (2^N): 3 = 8 pix, 4 = 16 pix, 5 = 32 pix, 6 = 64 pix.

Field	Function
	[12:10] - sw_max_trb_size - Transform Block max size (2^N): 3 = 8 pix, 4 = 16 pix, 5 = 32 pix, 6 = 64 pix. [9:7] - sw_max_intra_hierdepth - Intra max hierarchy dept [6:4] - sw_max_inter_hierdepth - Inter max hierarchy depth [3:0] - sw_parallel_merge - Information about differential MV exist inside of coded block. For VP9: [31:29] - Reserved - Not used [28:23] - sw_qp_delta_y_dc - QP delta value for Y DC [22:17] - sw_qp_delta_ch_dc - QP delta value for C DC [16:11] - sw_qp_delta_ch_ac - QP delta value for C AC [10] - sw_last_sign_bias - Previous reference picture sign bias for motion vector decoding [9] - sw_lossless_e - Lossless transform enable [8:7] - sw_comp_pred_var_ref1 - Compaund prediction parameter to select correct sign bias from bin decoding [6:5] - sw_comp_pred_var_ref0 - Compaund prediction parameter to select correct sign bias from bin decoding [4:3] - sw_comp_pred_fixed_ref - Compaund prediction parameter to select last picture sign bias [2] - sw_segment_temp_upd_e - temporal segmentation update enable [1] - sw_segment_upd_e - segmentation update enable. If high new segmentation values are delivered in stream (and will be written out). If 0 the segmentation values are read from external memory [0] - sw_segment_e - segmentation enable

16.2.5.1.16 Initial ref pic list register (0-2) (SWREG14)

16.2.5.1.16.1 Offset

Register	Offset
SWREG14	38h

16.2.5.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_REF_PIC_0_2_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_REF_PIC_0_2_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.16.3 Fields

Field	Function
31-0 INIT_REF_PIC_0_2_BF	For HEVC: [31:30] - Reserved - Not used [29:25] - sw_init_rlist_b2 - Initial reference picture list for backward picid 2 [24:20] - sw_init_rlist_f2 - Initial reference picture list for forward picid 2 [19:15] - sw_init_rlist_b1 - Initial reference picture list for backward picid 1 [14:10] - sw_init_rlist_f1 - Initial reference picture list for forward picid 1 [9:5] - sw_init_rlist_b0 - Initial reference picture list for backward picid 0 [4:0] - sw_init_rlist_f0 - Initial reference picture list for forward picid 0 For VP9: [31:24] - Reserved - Not used [23:18] - sw_filt_level - Frame filtering level [17:15] - sw_refpic_seg0 - Segment refer picture [14] - sw_skip_seg0 - Segment skip enable [13:8] - sw_filt_level_seg0 - Segment filter level [7:0] - sw_quant_seg0 - Segment quantization parameter

16.2.5.1.17 Initial ref pic list register (3-5) (SWREG15)

16.2.5.1.17.1 Offset

Register	Offset
SWREG15	3Ch

16.2.5.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_REF_PIC_3_5_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_REF_PIC_3_5_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.17.3 Fields

Field	Function
31-0	For HEVC:

Field	Function
INIT_REF_PIC_3_5_BF	[31:30] - Reserved - Not used [29:25] - sw_init_rlist_b5 - Initial reference picture list for backward picid 5 [24:20] - sw_init_rlist_f5 - Initial reference picture list for forward picid 5 [19:15] - sw_init_rlist_b4 - Initial reference picture list for backward picid 4 [14:10] - sw_init_rlist_f4 - Initial reference picture list for forward picid 4 [9:5] - sw_init_rlist_b3 - Initial reference picture list for backward picid 3 [4:0] - sw_init_rlist_f3 - Initial reference picture list for forward picid 3 For VP9: [31:18] - Reserved - Not used [17:15] - sw_refpic_seg1 - Segment refer picture [14] - sw_skip_seg1 - Segment skip enable [13:8] - sw_filt_level_seg1 - Segment filter level [7:0] - sw_quant_seg1 - Segment quantization parameter

16.2.5.1.18 Initial ref pic list register (6-8) (SWREG16)

16.2.5.1.18.1 Offset

Register	Offset
SWREG16	40h

16.2.5.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_REF_PIC_6_8_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_REF_PIC_6_8_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.18.3 Fields

Field	Function
31-0 INIT_REF_PIC_6_8_BF	For HEVC: [31:30] - Reserved - Not used [29:25] - sw_init_rlist_b8 - Initial reference picture list for backward picid 8 [24:20] - sw_init_rlist_f8 - Initial reference picture list for forward picid 8 [19:15] - sw_init_rlist_b7 - Initial reference picture list for backward picid 7 [14:10] - sw_init_rlist_f7 - Initial reference picture list for forward picid 7

Field	Function
	[9:5] - sw_init_rlist_b6 - Initial reference picture list for backward picid 6 [4:0] - sw_init_rlist_f6 - Initial reference picture list for forward picid 6 For VP9: [31:18] - Reserved - Not used [17:15] - sw_refpic_seg2 - Segment refer picture [14] - sw_skip_seg2 - Segment skip enable [13:8] - sw_filt_level_seg2 - Segment filter level [7:0] - sw_quant_seg2 - Segment quantization parameter

16.2.5.1.19 Initial ref pic list register (9-11) (SWREG17)

16.2.5.1.19.1 Offset

Register	Offset
SWREG17	44h

16.2.5.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_REF_PIC_9_11_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_REF_PIC_9_11_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.19.3 Fields

Field	Function
31-0 INIT_REF_PIC_9_11_BF	For HEVC: [31:30] - Reserved - Not used [29:25] - sw_init_rlist_b11 - Initial reference picture list for backward picid 11 [24:20] - sw_init_rlist_f11 - Initial reference picture list for forward picid 11 [19:15] - sw_init_rlist_b10 - Initial reference picture list for backward picid 10 [14:10] - sw_init_rlist_f10 - Initial reference picture list for forward picid 10 [9:5] - sw_init_rlist_b9 - Initial reference picture list for backward picid 9 [4:0] - sw_init_rlist_f9 - Initial reference picture list for forward picid 9 For VP9: [31:18] - Reserved - Not used

Field	Function
	[17:15] - sw_refpic_seg3 - Segment refer picture [14] - sw_skip_seg3 - Segment skip enable [13:8] - sw_filt_level_seg3 - Segment filter level [7:0] - sw_quant_seg3 - Segment quantization parameter

16.2.5.1.20 Initial ref pic list register (12-14) (SWREG18)

16.2.5.1.20.1 Offset

Register	Offset
SWREG18	48h

16.2.5.1.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_REF_PIC_12_14_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_REF_PIC_12_14_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.20.3 Fields

Field	Function
31-0 INIT_REF_PIC_12_14_BF	For HEVC: [31:30] - Reserved - Not used [29:25] - sw_init_rlist_b14 - Initial reference picture list for backward picid 14 [24:20] - sw_init_rlist_f14 - Initial reference picture list for forward picid 14 [19:15] - sw_init_rlist_b13 - Initial reference picture list for backward picid 13 [14:10] - sw_init_rlist_f13 - Initial reference picture list for forward picid 13 [9:5] - sw_init_rlist_b12 - Initial reference picture list for backward picid 12 [4:0] - sw_init_rlist_f12 - Initial reference picture list for forward picid 12 For VP9: [31:18] - Reserved - Not used [17:15] - sw_refpic_seg4 - Segment refer picture [14] - sw_skip_seg4 - Segment skip enable [13:8] - sw_filt_level_seg4 - Segment filter level [7:0] - sw_quant_seg4 - Segment quantization parameter

16.2.5.1.21 Initial ref pic list register (15 and P 0-3) (SWREG19)

16.2.5.1.21.1 Offset

Register	Offset
SWREG19	4Ch

16.2.5.1.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_REF_PIC_15_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_REF_PIC_15_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.21.3 Fields

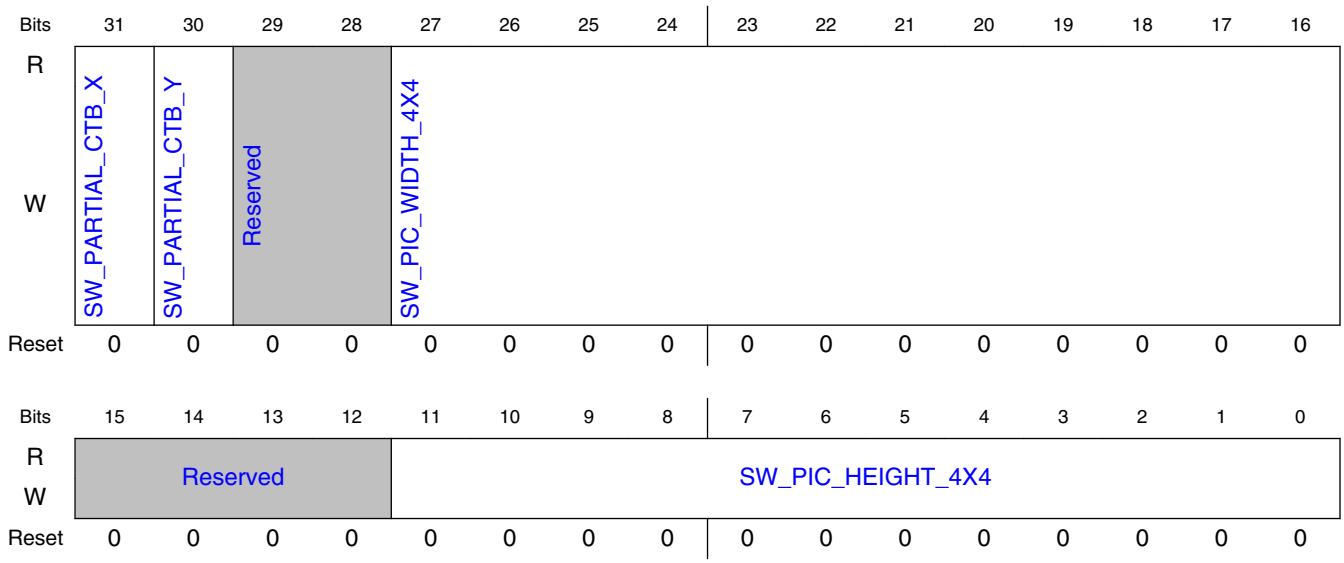
Field	Function
31-0 INIT_REF_PIC_15_BF	For HEVC: [31:10] - Reserved - Not used [9:5] - sw_init_rlist_b15 - Initial reference picture list for backward picid 15 [4:0] - sw_init_rlist_f15 - Initial reference picture list for forward picid 15 For VP9: [31:18] - Reserved - Not used [17:15] - sw_refpic_seg5 - Segment refer picture [14] - sw_skip_seg5 - Segment skip enable [13:8] - sw_filt_level_seg5 - Segment filter level [7:0] - sw_quant_seg5 - Segment quantization parameter

16.2.5.1.22 Decoder control register 11 (SWREG20)

16.2.5.1.22.1 Offset

Register	Offset
SWREG20	50h

16.2.5.1.22.2 Diagram



16.2.5.1.22.3 Fields

Field	Function
31 SW_PARTIAL_CTB_X	Picture width not multiple of CTB size
30 SW_PARTIAL_CTB_Y	Picture height not multiple of CTB size
29-28 —	Reserved.
27-16 SW_PIC_WIDTH_4X4	Current picture width in 4x4 blocks (Needed to reduce overlapping HW conditions in various blocks)
15-12 —	Reserved.
11-0 SW_PIC_HEIGHT_4X4	Current picture height in 4x4 blocks (Needed to reduce overlapping HW conditions in various blocks)

16.2.5.1.23 Not used (SWREG21)

16.2.5.1.23.1 Offset

Register	Offset
SWREG21	54h

16.2.5.1.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.23.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.24 Not used (SWREG22)

16.2.5.1.24.1 Offset

Register	Offset
SWREG22	58h

16.2.5.1.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.24.3 Fields

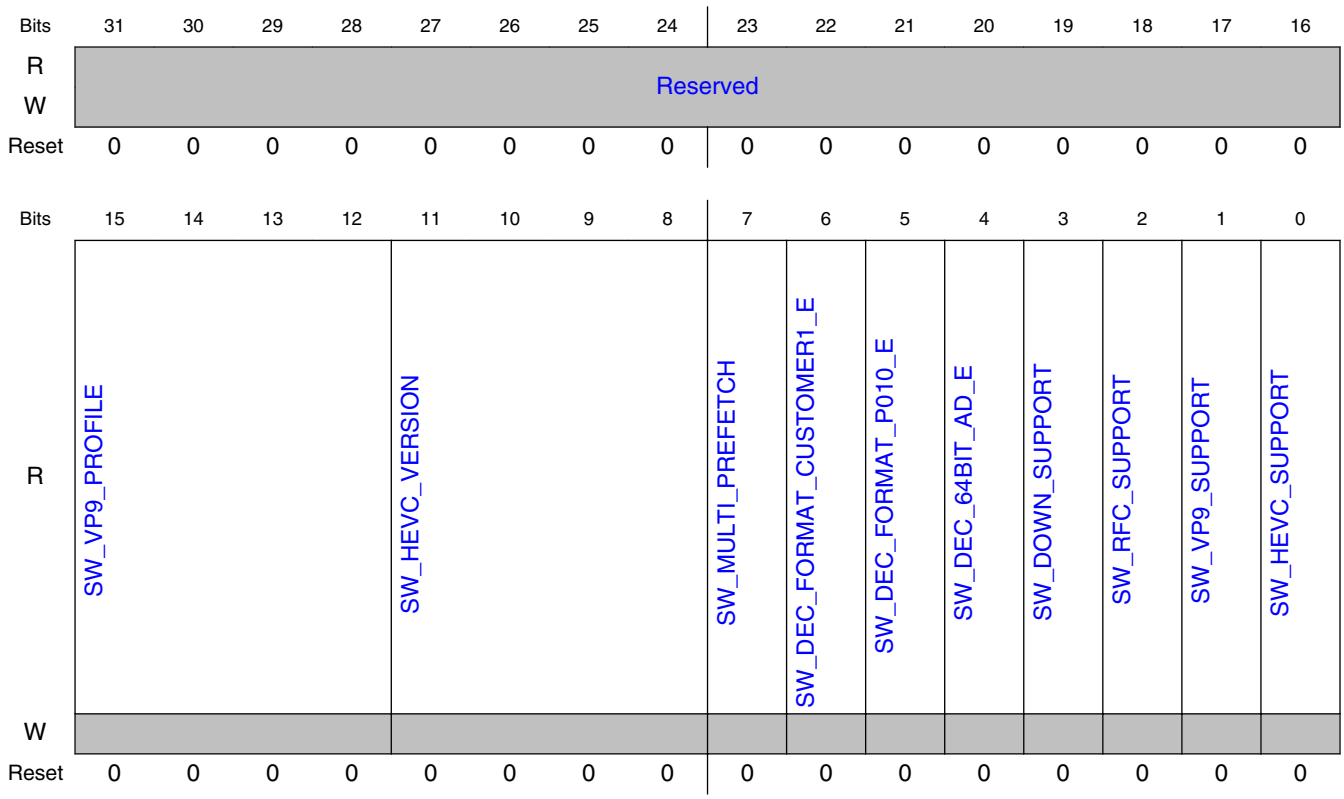
Field	Function
31-0 —	Reserved.

16.2.5.1.25 Decoder configure status register (SWREG23)

16.2.5.1.25.1 Offset

Register	Offset
SWREG23	5Ch

16.2.5.1.25.2 Diagram



16.2.5.1.25.3 Fields

Field	Function
31-16 —	Reserved.
15-12 SW_VP9_PROFILE	VP9 version 0000b - vp9 profile 0 0001b - vp9 profile 2 - 10bits
11-8 SW_HEVC_VERSION	HEVC version 0000b - main8 0001b - main10
7 SW_MULTI_PREFETCH	Multi-Reference Blocks Prefetch 0b - Not supported 1b - Supported

6
SW_DEC_FORMAT_CUSTOMER1_E
	Customized output format support
	Demonstrated as following - from MSB->LSB in a 128-bit burst.
	• 10-bit pixel
	  • 128-bit burst0: Y0|Y1|Y2|Y3|...|Y11|Y12[9:2]
	  • 128-bit burst1: Y12[1:0]|Y13|Y14|Y15|...|Y24|Y25[9:4]
	  • 128-bit burst2: Y25[3:0]|Y26|Y27|Y28|...|Y37|Y38[9:6]

Table continues on the next page...

Field	Function
	• 128-bit burst3: Y38[5:0]|Y39|Y40|Y41|...|Y50|Y51[9:8] • 128-bit burst4: Y51[7:0]|Y52|Y53|Y54|...|Y62|Y63 • 8-bit pixel • 128-bit burst: Y0|Y1|Y2|...|Y15 • 128-bit burst: Y16|Y17|Y18|...|Y31 0b - Not supported 1b - Supported
5 SW_DEC_FORMAT_P010_E	P010 output format support 0b - Not supported 1b - Supported
4 SW_DEC_64BIT_AD_E	64 bit addressing of master interface support 0b - Not supported (32 bit addressing) 1b - Supported
3 SW_DOWN_SUPPORT	Downscale support 0b - Do not support downscale 1b - Support downscale
2 SW_RFC_SUPPORT	RFC support 0b - Do not support RFC 1b - Support RFC
1 SW_VP9_SUPPORT	VP9 support 0b - Do not support VP9 1b - Support VP9
0 SW_HEVC_SUPPORT	HEVC support 0b - Do not support HEVC 1b - Support HEVC

16.2.5.1.26 Not used (SWREG24)

16.2.5.1.26.1 Offset

Register	Offset
SWREG24	60h

16.2.5.1.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.26.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.27 Not used (SWREG25)

16.2.5.1.27.1 Offset

Register	Offset
SWREG25	64h

16.2.5.1.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.27.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.28 Not used (SWREG26)

16.2.5.1.28.1 Offset

Register	Offset
SWREG26	68h

16.2.5.1.28.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.28.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.29 Not used (SWREG27)

16.2.5.1.29.1 Offset

Register	Offset
SWREG27	6Ch

16.2.5.1.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.29.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.30 Not used (SWREG28)

16.2.5.1.30.1 Offset

Register	Offset
SWREG28	70h

16.2.5.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.30.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.31 Not used (SWREG29)

16.2.5.1.31.1 Offset

Register	Offset
SWREG29	74h

16.2.5.1.31.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.31.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.32 Not used (SWREG30)

16.2.5.1.32.1 Offset

Register	Offset
SWREG30	78h

16.2.5.1.32.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.32.3 Fields

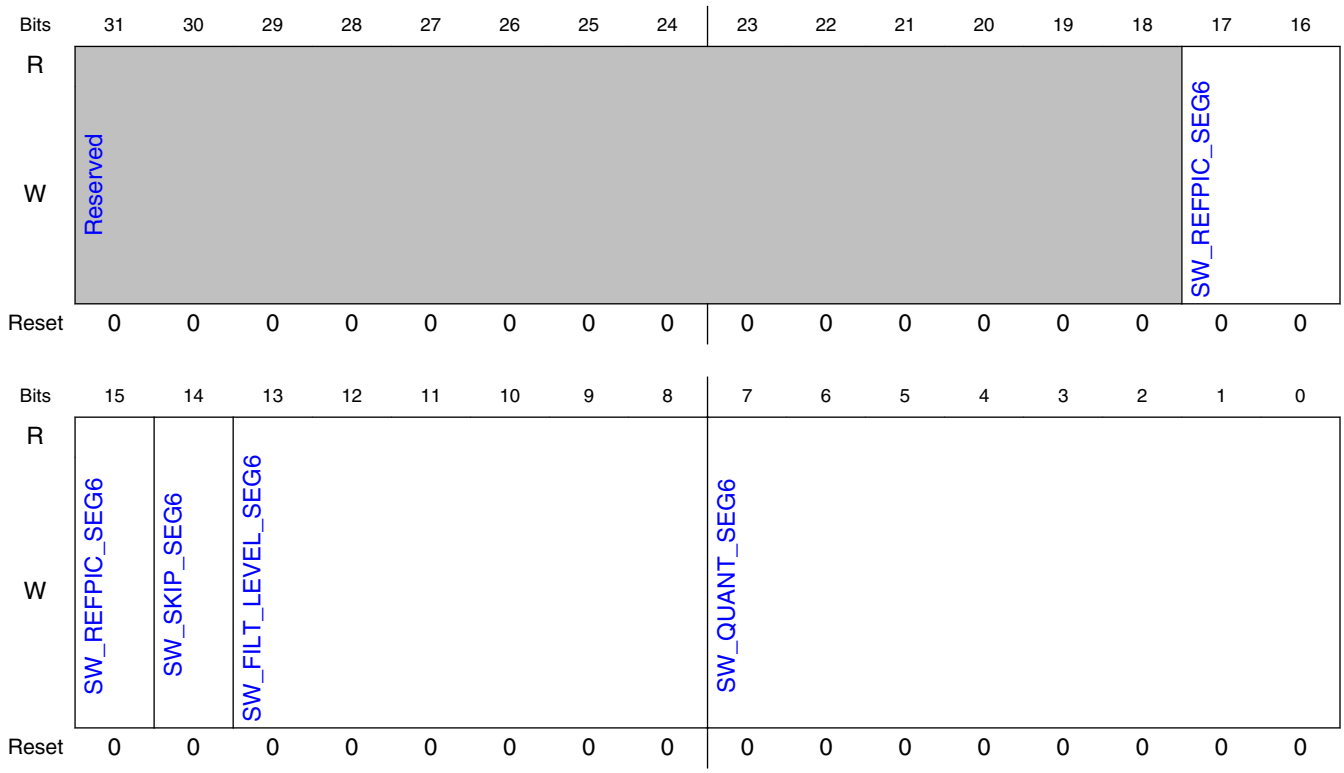
Field	Function
31-0 —	Reserved.

16.2.5.1.33 VP9 segmentation values (SWREG31)

16.2.5.1.33.1 Offset

Register	Offset
SWREG31	7Ch

16.2.5.1.33.2 Diagram



16.2.5.1.33.3 Fields

Field	Function
31-18 —	Reserved.
17-15 SW_REFPIC_SEG6	Segment refer picture
14 SW_SKIP_SEG6	Segment skip enable
13-8	Segment filter level

Table continues on the next page...

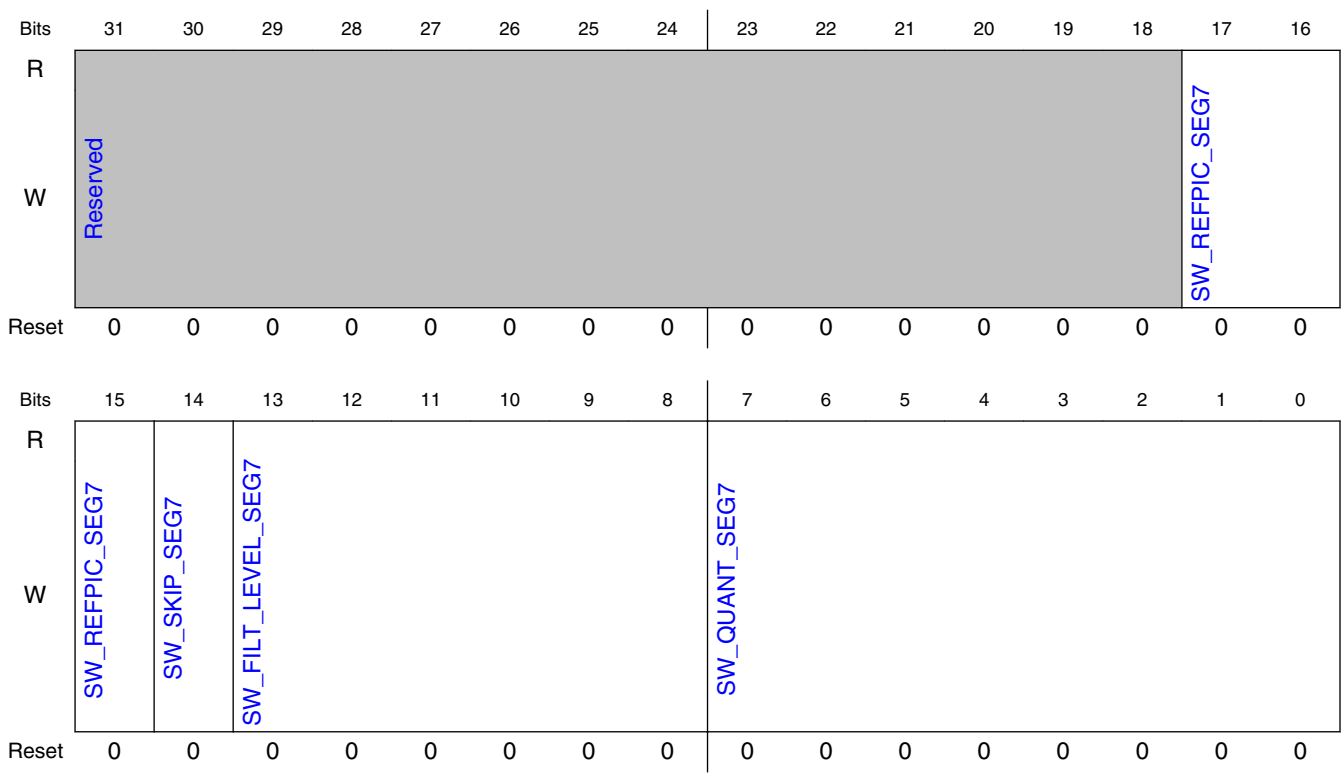
Field	Function
SW_FILT_LEVEL_SEG6	
7-0 SW_QUANT_SEG6	Segment quantization parameter

16.2.5.1.34 VP9 segmentation values (SWREG32)

16.2.5.1.34.1 Offset

Register	Offset
SWREG32	80h

16.2.5.1.34.2 Diagram



16.2.5.1.34.3 Fields

Field	Function
31-18 —	Reserved.
17-15 SW_REFPIC_SEG7	Segment refer picture
14 SW_SKIP_SEG7	Segment skip enable
13-8 SW_FILT_LEVEL_SEG7	Segment filter level
7-0 SW_QUANT_SEG7	Segment quantization parameter

16.2.5.1.35 VP9 reference picture scaling register 0 (SWREG33)

16.2.5.1.35.1 Offset

Register	Offset
SWREG33	84h

16.2.5.1.35.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_LREF_WIDTH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_LREF_HEIGHT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.35.3 Fields

Field	Function
31-16 SW_LREF_WIDTH	Accurate width of last (previous) reference picture in pixels
15-0 SW_LREF_HEIGHT	Accurate height of last (previous) reference picture in pixels

16.2.5.1.36 VP9 reference picture scaling register 1 (SWREG34)

16.2.5.1.36.1 Offset

Register	Offset
SWREG34	88h

16.2.5.1.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_GREF_WIDTH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_GREF_HEIGHT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.36.3 Fields

Field	Function
31-16 SW_GREF_WIDTH	Accurate width of golden reference picture in pixels
15-0 SW_GREF_HEIGHT	Accurate height of golden reference picture in pixels

16.2.5.1.37 VP9 reference picture scaling register 2 (SWREG35)

16.2.5.1.37.1 Offset

Register	Offset
SWREG35	8Ch

16.2.5.1.37.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_AREF_WIDTH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_AREF_HEIGHT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.37.3 Fields

Field	Function
31-16 SW_AREF_WIDTH	Accurate width of alternate reference picture in pixels
15-0 SW_AREF_HEIGHT	Accurate height of alternate reference picture in pixels

16.2.5.1.38 VP9 reference picture scaling register 3 (SWREG36)

16.2.5.1.38.1 Offset

Register	Offset
SWREG36	90h

16.2.5.1.38.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_LREF_HOR_SCALE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_LREF_VER_SCALE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.38.3 Fields

Field	Function
31-16 SW_LREF_HOR_SCALE	Horizontal scaling factor for last (previous) reference picture
15-0 SW_LREF_VER_SCALE	Vertical scaling factor for last (previous) reference picture

16.2.5.1.39 VP9 reference picture scaling register 4 (SWREG37)

16.2.5.1.39.1 Offset

Register	Offset
SWREG37	94h

16.2.5.1.39.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_GREF_HOR_SCALE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_GREF_VER_SCALE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.39.3 Fields

Field	Function
31-16 SW_GREF_HOR_SCALE	Horizontal scaling factor for golden reference picture
15-0 SW_GREF_VER_SCALE	Vertical scaling factor for golden reference picture

16.2.5.1.40 VP9 reference picture scaling register 5 (SWREG38)

16.2.5.1.40.1 Offset

Register	Offset
SWREG38	98h

16.2.5.1.40.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_AREF_HOR_SCALE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_AREF_VER_SCALE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.40.3 Fields

Field	Function
31-16 SW_AREF_HOR_SCALE	Horizontal scaling factor for alternate reference picture
15-0 SW_AREF_VER_SCALE	Vertical scaling factor for alternate reference picture

16.2.5.1.41 Not used (SWREG39)

16.2.5.1.41.1 Offset

Register	Offset
SWREG39	9Ch

16.2.5.1.41.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.41.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.42 Not used (SWREG40)

16.2.5.1.42.1 Offset

Register	Offset
SWREG40	A0h

16.2.5.1.42.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.42.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.43 Not used (SWREG41)

16.2.5.1.43.1 Offset

Register	Offset
SWREG41	A4h

16.2.5.1.43.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.43.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.44 Not used (SWREG42)

16.2.5.1.44.1 Offset

Register	Offset
SWREG42	A8h

16.2.5.1.44.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.44.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.45 Not used (SWREG43)

16.2.5.1.45.1 Offset

Register	Offset
SWREG43	ACh

16.2.5.1.45.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.45.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.46 Not used (SWREG44)

16.2.5.1.46.1 Offset

Register	Offset
SWREG44	B0h

16.2.5.1.46.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.46.3 Fields

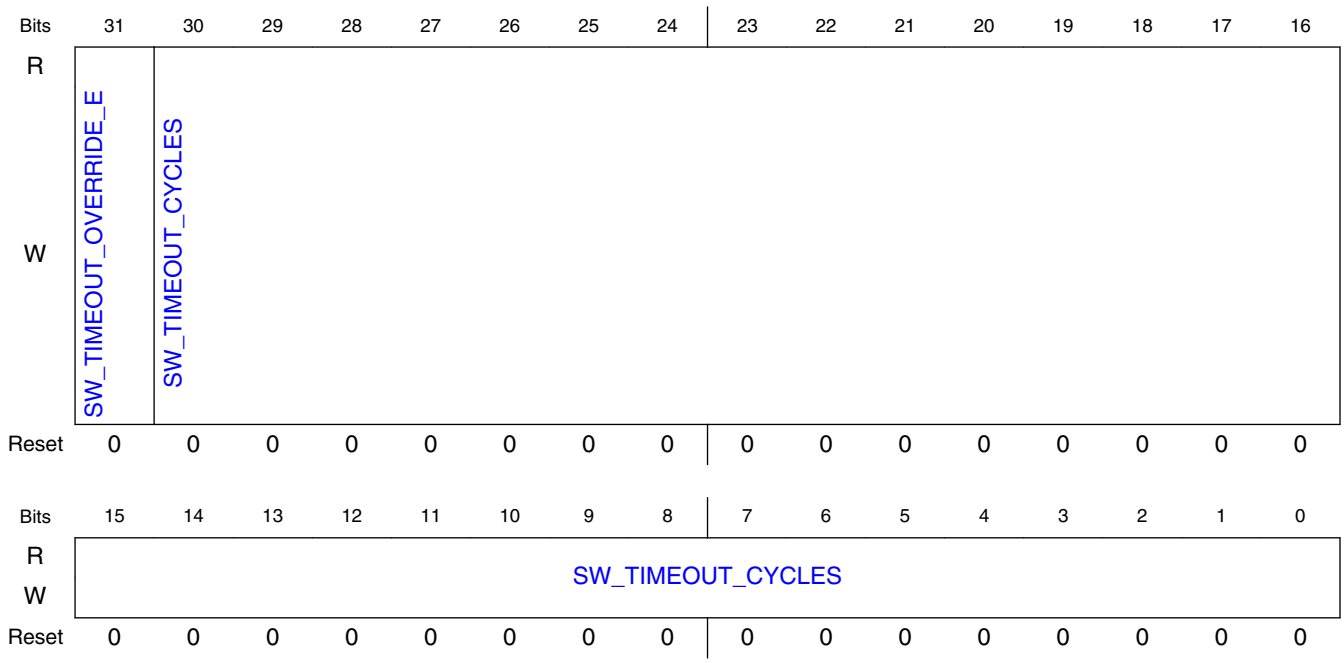
Field	Function
31-0 —	Reserved.

16.2.5.1.47 Timeout control register (SWREG45)

16.2.5.1.47.1 Offset

Register	Offset
SWREG45	B4h

16.2.5.1.47.2 Diagram



16.2.5.1.47.3 Fields

Field	Function
31 SW_TIMEOUT_OVERRIDE_E	Enable for SW controlled timeout. If enabled the sw_timeout_cycles is used to detect HW timeout instead of hard coded HW value
30-0 SW_TIMEOUT_CYCLES	Amount of clock cycles to trigger timeout interrupt if no external master activity acknowledged. Used if sw_timeout_override_e is set

16.2.5.1.48 Picture order count from current pictures for index 0-3 (SWREG46)

16.2.5.1.48.1 Offset

Register	Offset
SWREG46	B8h

16.2.5.1.48.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PIC_ORD_0_3_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PIC_ORD_0_3_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.48.3 Fields

Field	Function
31-0 PIC_ORD_0_3_BF	For HEVC: [31:24] - sw_cur_poc_00 - Picture order count from current picture 0 [23:16] - sw_cur_poc_01 - Picture order count from current picture 1 [15:8] - sw_cur_poc_02 - Picture order count from current picture 2 [7:0] - sw_cur_poc_03 - Picture order count from current picture 3 For VP9: [31] - Reserved - Not used [30:24] - sw_filt_ref_adj_0 - Filter level intra adjustment [23] - Reserved - Not used [22:16] - sw_filt_ref_adj_1 - Filter level last ref pic adjustment [15] - Reserved - Not used [14:8] - sw_filt_ref_adj_2 - Filter level golden pic adjustment [7] - Reserved - Not used [6:0] - sw_filt_ref_adj_3 - Filter level alt ref pic adjustment

16.2.5.1.49 Picture order count from current pictures for index 4-7 (SWREG47)

16.2.5.1.49.1 Offset

Register	Offset
SWREG47	BCh

16.2.5.1.49.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PIC_ORD_4_7_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PIC_ORD_4_7_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.49.3 Fields

Field	Function
31-0 PIC_ORD_4_7_BF	For HEVC: [31:24] - sw_cur_poc_04 - Picture order count from current picture 4 [23:16] - sw_cur_poc_05 - Picture order count from current picture 5 [15:8] - sw_cur_poc_06 - Picture order count from current picture 6 [7:0] - sw_cur_poc_07 - Picture order count from current picture 7 For VP9: [31] - Reserved - Not used [30:24] - sw_filt_mb_adj_0 - Filter level ZERO mv adjustment [23] - Reserved - Not used [22:16] - sw_filt_mb_adj_1 - Filter level adjustment [15] - Reserved - Not used [14:8] - sw_filt_mb_adj_2 - Filter level adjustment (Not used) [7] - Reserved - Not used [6:0] - sw_filt_mb_adj_3 - Filter level adjustment (Not used)

16.2.5.1.50 Picture order count from current pictures for index 8-11 (SWREG48)

16.2.5.1.50.1 Offset

Register	Offset
SWREG48	C0h

16.2.5.1.50.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_CUR_POC_08								SW_CUR_POC_09							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_CUR_POC_10								SW_CUR_POC_11							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.50.3 Fields

Field	Function
31-24 SW_CUR_POC_08	Picture order count from current picture 8
23-16 SW_CUR_POC_09	Picture order count from current picture 9
15-8 SW_CUR_POC_10	Picture order count from current picture 10
7-0 SW_CUR_POC_11	Picture order count from current picture 11

16.2.5.1.51 Picture order count from current pictures for index 12-15 (SWREG49)

16.2.5.1.51.1 Offset

Register	Offset
SWREG49	C4h

16.2.5.1.51.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_CUR_POC_12								SW_CUR_POC_13							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_CUR_POC_14								SW_CUR_POC_15							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.51.3 Fields

Field	Function
31-24 SW_CUR_POC_12	Picture order count from current picture 12
23-16 SW_CUR_POC_13	Picture order count from current picture 13
15-8 SW_CUR_POC_14	Picture order count from current picture 14
7-0 SW_CUR_POC_15	Picture order count from current picture 15

16.2.5.1.52 Synthesis configuration register decoder 0 (read only) (SWREG50)

16.2.5.1.52.1 Offset

Register	Offset
SWREG50	C8h

16.2.5.1.52.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					SW_DEC_MAX_OWIDTH										
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

16.2.5.1.52.3 Fields

Field	Function
31-11 —	Reserved.
10-0 SW_DEC_MAX_OWIDTH	Max configured decoder video resolution that can be decoded. Informed as width of the picture in pixels

16.2.5.1.53 Reference picture buffer control register (SWREG51)

16.2.5.1.53.1 Offset

Register	Offset
SWREG51	CCh

16.2.5.1.53.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.53.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.54 Reference picture buffer information register 1 (read only) (SWREG52)

16.2.5.1.54.1 Offset

Register	Offset
SWREG52	D0h

16.2.5.1.54.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.54.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.55 Reference picture buffer information register 2 (read only) (SWREG53)

16.2.5.1.55.1 Offset

Register	Offset
SWREG53	D4h

16.2.5.1.55.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.55.3 Fields

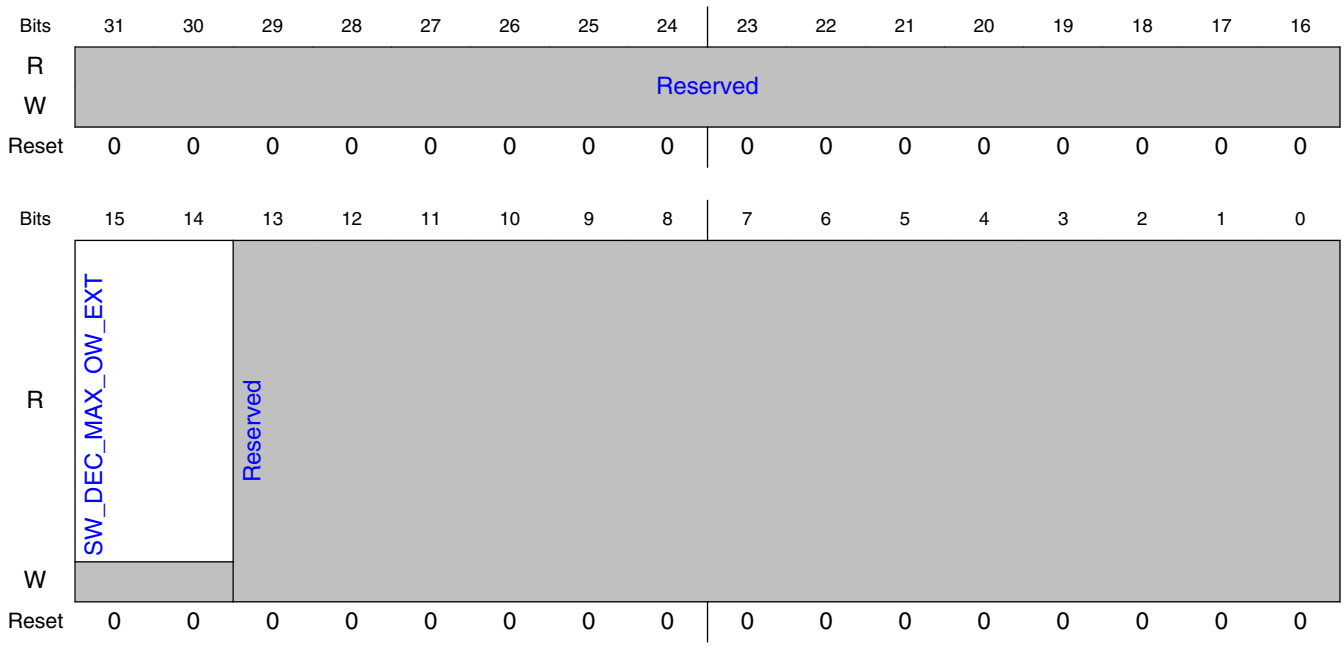
Field	Function
31-0 —	Reserved.

16.2.5.1.56 Synthesis configuration register decoder 1 (read only) (SWREG54)

16.2.5.1.56.1 Offset

Register	Offset
SWREG54	D8h

16.2.5.1.56.2 Diagram



16.2.5.1.56.3 Fields

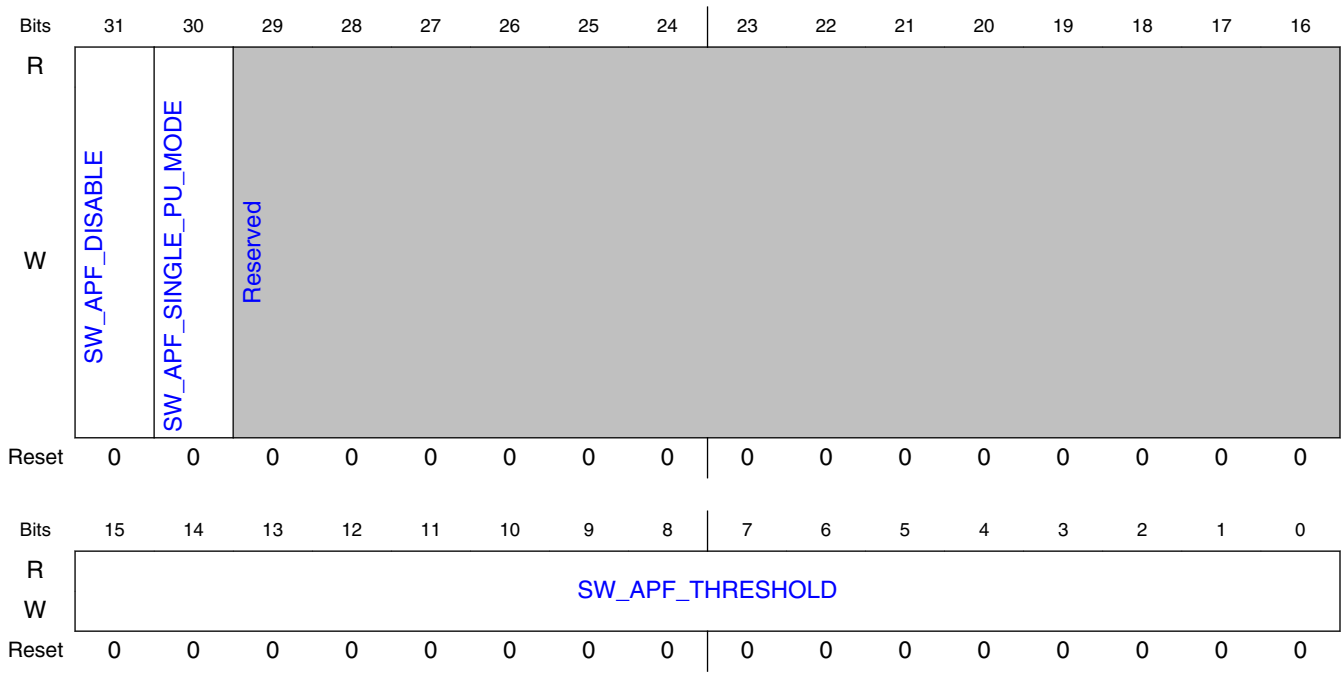
Field	Function
31-16 —	Reserved.
15-14 SW_DEC_MAX_OW_EXT	Max configured decoder video resolution that can be decoded. This is the MSB part of the configuration signal
13-0 —	Reserved.

16.2.5.1.57 Advanced prefetch control register (SWREG55)

16.2.5.1.57.1 Offset

Register	Offset
SWREG55	DCh

16.2.5.1.57.2 Diagram



16.2.5.1.57.3 Fields

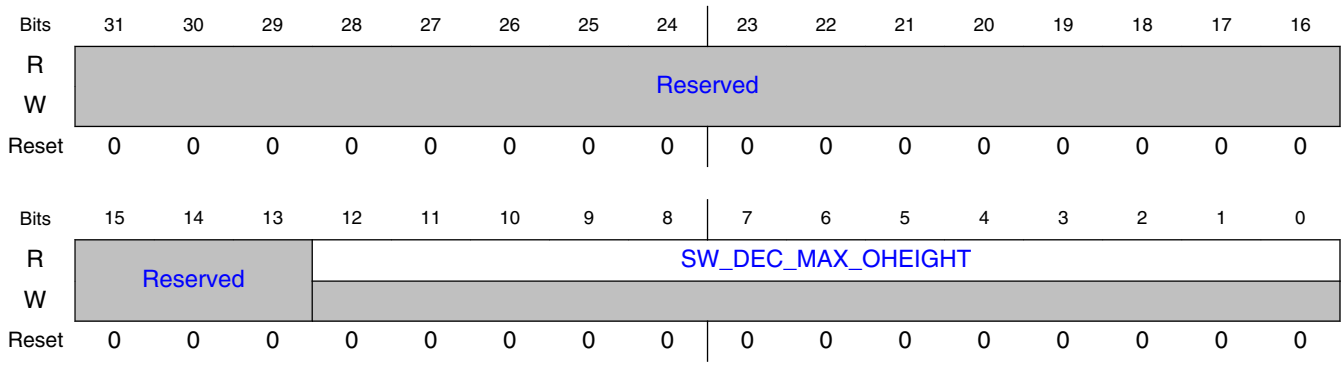
Field	Function
31 SW_APF_DISABLE	Advanced prefetch disable. If hight each partition is read separately
30 SW_APF_SINGLE_PU_MODE	APF amount of buffered Pus: can be restricted to buffer one PU at a time
29-16 —	Reserved.
15-0 SW_APF_THRESHOLD	Advanced prefetch threshold. If current buffered unit exceeds the threshold the advanced mode is not used. Value 0 disables threshold usage and advanced prefetch usage is restricted by internal memory limitation only

16.2.5.1.58 Synthesis configuration register decoder 2 (read only) (SWREG56)

16.2.5.1.58.1 Offset

Register	Offset
SWREG56	E0h

16.2.5.1.58.2 Diagram



16.2.5.1.58.3 Fields

Field	Function
31-13 —	Reserved.
12-0 SW_DEC_MAX_OHEIGHT	Max supported picture height in pixels

16.2.5.1.59 Decoder fuse register (read only) (SWREG57)

16.2.5.1.59.1 Offset

Register	Offset
SWREG57	E4h

16.2.5.1.59.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.59.3 Fields

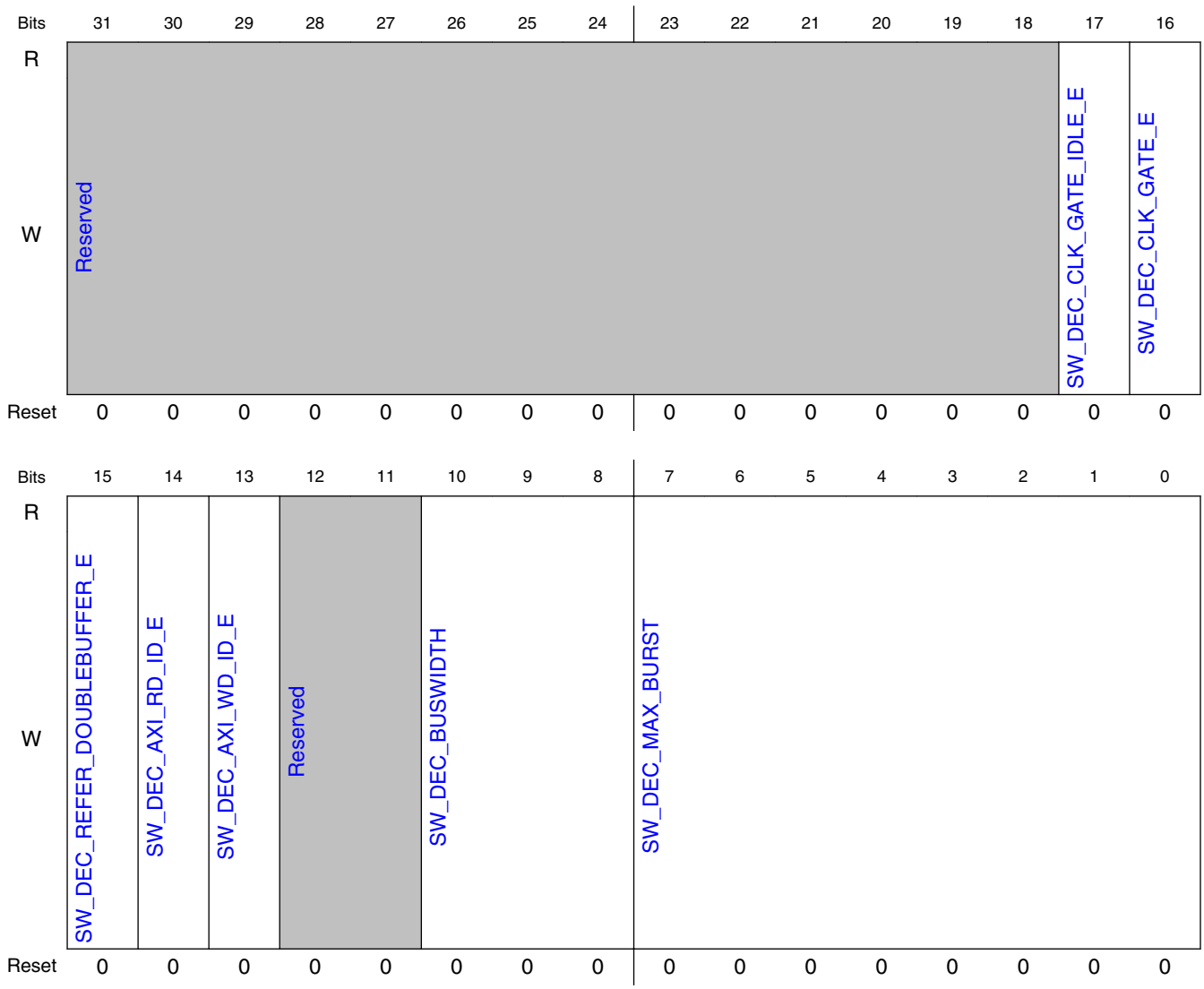
Field	Function
31-0 —	Reserved.

16.2.5.1.60 Device configuration register decoder 2 + Multi core control register (SWREG58)

16.2.5.1.60.1 Offset

Register	Offset
SWREG58	E8h

16.2.5.1.60.2 Diagram



16.2.5.1.60.3 Fields

Field	Function
31-18 —	Reserved.
17 SW_DEC_CLK_GATE_IDLE_E	Clock gating enable for decoder run-time. Generated separate clocks for each block by its own IDLE signal.
16 SW_DEC_CLK_GATE_E	Clock gating enable for picture-wise/decoding format clock gating. Between each picture the clock is gated from HW if this bit is high

Table continues on the next page...

Field	Function
15 SW_DEC_REFER_DOUBLEBUFFER_E	HW internal double buffering enable for reference data. This enable requires that there are two buffers available at the configured decoder (see configuration register values)
14 SW_DEC_AXI_RD_ID_E	SW axi ID enable. When enabled the given sw_dec_axi_rd_id is used as ID base and each sub-block will use offsets 0...max
13 SW_DEC_AXI_WD_ID_E	SW axi ID enable. When enabled the given sw_dec_axi_wd_id is used as ID base and each sub-block will use offsets 0...max
12-11 —	Reserved.
10-8 SW_DEC_BUSWIDTH	Decoder master interface buswidth 000b - 32 bit bus 001b - 64 bit bus 010b - 128 bit bus
7-0 SW_DEC_MAX_BURST	Maximum burst length for decoder bus transactions. Valid values: AXI: 1-256

16.2.5.1.61 Device configuration register AXI ID (SWREG59)

16.2.5.1.61.1 Offset

Register	Offset
SWREG59	ECh

16.2.5.1.61.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_AXI_WR_ID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_AXI_RD_ID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.61.3 Fields

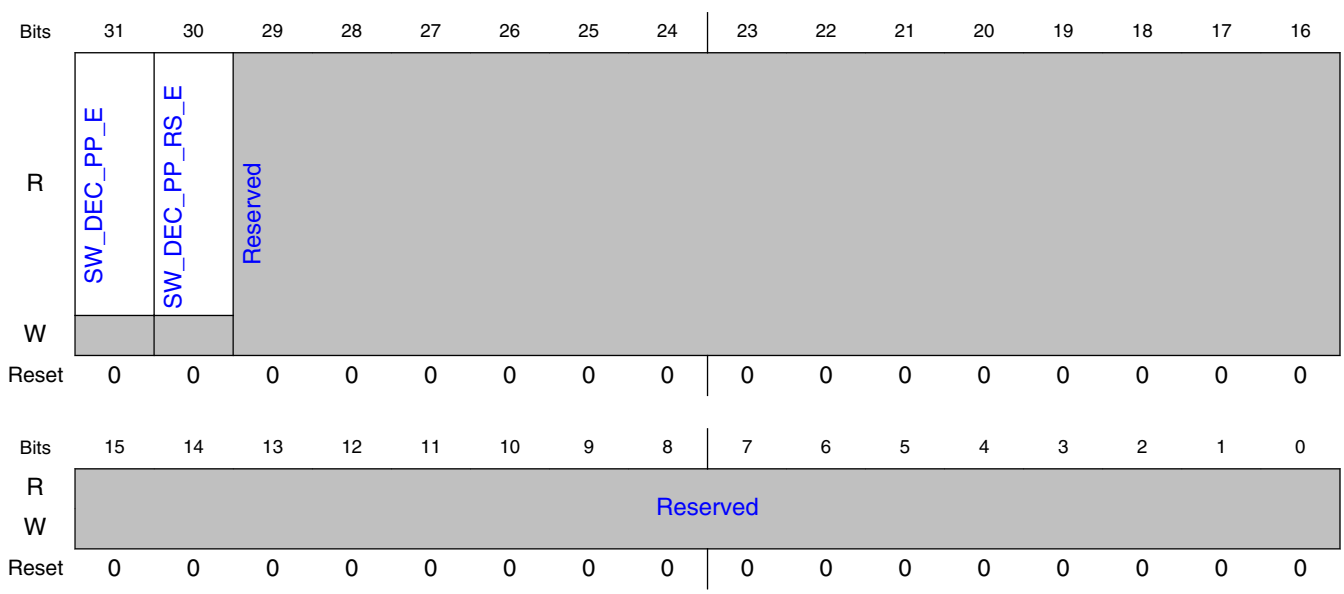
Field	Function
31-16 SW_DEC_AXI_WR_ID	Read ID base for HW write accesses. Each writing device use AXI ID of base+deviceoffset (where device offset is 0 1 2 3...Number of reading sub-blocks)
15-0 SW_DEC_AXI_RD_ID	Write ID base for HW write accesses. Each writing device use AXI ID of base+deviceoffset (where device offset is 0 1 2 3...Number of writing sub-blocks)

16.2.5.1.62 Synthesis configuration register decoder 3 for PP (read only) (SWREG60)

16.2.5.1.62.1 Offset

Register	Offset
SWREG60	F0h

16.2.5.1.62.2 Diagram



16.2.5.1.62.3 Fields

Field	Function
31	Decoder include PP

Table continues on the next page...

Field	Function
SW_DEC_PP_E	0b - PP does not exist. None of the PP features can be enabled. 1b - PP exists
30 SW_DEC_PP_RS_E	Decoder PP raster scan output support 0b - Raster scan output not supported 1b - Raster scan output supported
29-0 —	Reserved.

16.2.5.1.63 Not used (SWREG61)

16.2.5.1.63.1 Offset

Register	Offset
SWREG61	F4h

16.2.5.1.63.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.63.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.64 HW proceed register (CU location) (SWREG62)

16.2.5.1.64.1 Offset

Register	Offset
SWREG62	F8h

16.2.5.1.64.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_CU_LOCATION_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_CU_LOCATION_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.64.3 Fields

Field	Function
31-16 SW_CU_LOCATION_X	Cu horizontal start location X in pixels (returned HW internal position during interrupt)
15-0 SW_CU_LOCATION_Y	Cu vertical start location Y in pixels (returned HW internal position during interrupt)

16.2.5.1.65 HW performance register (cycles running) (SWREG63)

16.2.5.1.65.1 Offset

Register	Offset
SWREG63	FCh

16.2.5.1.65.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_PERF_CYCLE_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_PERF_CYCLE_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.65.3 Fields

Field	Function
31-0 SW_PERF_CYCLE_COUNT	HW clock cycle counter return value. Amount of consumed clock cycles returned to this register when interrupt is being made (any kind of interrupt)

16.2.5.1.66 Base address MSB (bits 63:32) for decoded luminance picture (SWREG64)

16.2.5.1.66.1 Offset

Register	Offset
SWREG64	100h

16.2.5.1.66.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_DEC_OUT_YBASE_MSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_DEC_OUT_YBASE_MSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.66.3 Fields

Field	Function
31-0 SW_DEC_OUT_YBASE_MSB	Base address MSB (bits 63:32) for decoded luminance picture

16.2.5.1.67 Base address LSB (bits 31:0) for decoded luminance picture (SWREG65)

16.2.5.1.67.1 Offset

Register	Offset
SWREG65	104h

16.2.5.1.67.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.67.3 Fields

Field	Function
31-0 SW_DEC_OUT_YBASE_LSB	Base address LSB (bits 31:0) for decoded luminance picture

16.2.5.1.68 Base address MSB (bits 63:32) for reference luminance picture index 0 (SWREG66)

16.2.5.1.68.1 Offset

Register	Offset
SWREG66	108h

16.2.5.1.68.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.68.3 Fields

Field	Function
31-0 SW_REFER0_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 0

16.2.5.1.69 Base address LSB (bits 31:0) for reference luminance picture index 0 (SWREG67)

16.2.5.1.69.1 Offset

Register	Offset
SWREG67	10Ch

16.2.5.1.69.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.69.3 Fields

Field	Function
31-0 SW_REFER0_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 0

16.2.5.1.70 Base address MSB (bits 63:32) for reference luminance picture index 1 (SWREG68)

16.2.5.1.70.1 Offset

Register	Offset
SWREG68	110h

16.2.5.1.70.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.70.3 Fields

Field	Function
31-0 SW_REFER1_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 1

16.2.5.1.71 Base address LSB (bits 31:0) for reference luminance picture index 1 (SWREG69)

16.2.5.1.71.1 Offset

Register	Offset
SWREG69	114h

16.2.5.1.71.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.71.3 Fields

Field	Function
31-0 SW_REFER1_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 1

16.2.5.1.72 Base address MSB (bits 63:32) for reference luminance picture index 2 (SWREG70)

16.2.5.1.72.1 Offset

Register	Offset
SWREG70	118h

16.2.5.1.72.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.72.3 Fields

Field	Function
31-0 SW_REFER2_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 2

16.2.5.1.73 Base address LSB (bits 31:0) for reference luminance picture index 2 (SWREG71)

16.2.5.1.73.1 Offset

Register	Offset
SWREG71	11Ch

16.2.5.1.73.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.73.3 Fields

Field	Function
31-0 SW_REFER2_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 2

16.2.5.1.74 Base address MSB (bits 63:32) for reference luminance picture index 3 (SWREG72)

16.2.5.1.74.1 Offset

Register	Offset
SWREG72	120h

16.2.5.1.74.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.74.3 Fields

Field	Function
31-0 SW_REFER3_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 3

16.2.5.1.75 Base address LSB (bits 31:0) for reference luminance picture index 3 (SWREG73)

16.2.5.1.75.1 Offset

Register	Offset
SWREG73	124h

16.2.5.1.75.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.75.3 Fields

Field	Function
31-0 SW_REFER3_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 3

16.2.5.1.76 Base address MSB (bits 63:32) for reference luminance picture index 4 (SWREG74)

16.2.5.1.76.1 Offset

Register	Offset
SWREG74	128h

16.2.5.1.76.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.76.3 Fields

Field	Function
31-0 SW_REFER4_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 4

16.2.5.1.77 Base address LSB (bits 31:0) for reference luminance picture index 4 (SWREG75)

16.2.5.1.77.1 Offset

Register	Offset
SWREG75	12Ch

16.2.5.1.77.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.77.3 Fields

Field	Function
31-0 SW_REFER4_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 4

16.2.5.1.78 Base address MSB (bits 63:32) for reference luminance picture index 5 (SWREG76)

16.2.5.1.78.1 Offset

Register	Offset
SWREG76	130h

16.2.5.1.78.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.78.3 Fields

Field	Function
31-0 SW_REFER5_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 5

16.2.5.1.79 Base address LSB (bits 31:0) for reference luminance picture index 5 (SWREG77)

16.2.5.1.79.1 Offset

Register	Offset
SWREG77	134h

16.2.5.1.79.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.79.3 Fields

Field	Function
31-0 SW_REFER5_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 5

16.2.5.1.80 Base address MSB (bits 63:32) for reference luminance picture index 6 /VP9 segment write base MSB (SWREG78)

16.2.5.1.80.1 Offset

Register	Offset
SWREG78	138h

16.2.5.1.80.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR_6_MSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR_6_MSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.80.3 Fields

Field	Function
31-0 BASE_ADDR_6_MSB_BF	For HEVC: [31:0] - sw_refer6_ybase_msb - Base address MSB (bits 63:32) for reference luminance picture index 6 For VP9: [31:0] - sw_segment_write_base_msb - VP9 segment write base MSB

16.2.5.1.81 Base address LSB (bits 31:0) for reference luminance picture index 6 /VP9 segment write base LSB (SWREG79)

16.2.5.1.81.1 Offset

Register	Offset
SWREG79	13Ch

16.2.5.1.81.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR_6_LSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR_6_LSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.81.3 Fields

Field	Function
31-0 BASE_ADDR_6_LSB_BF	For HEVC: [31:0] - sw_refer6_ybase_lsb - Base address LSB (bits 31:0) for reference luminance picture index 6 For VP9: [31:0] - sw_segment_write_base_lsb - VP9 segment write base LSB

16.2.5.1.82 Base address MSB (bits 63:32) for reference luminance picture index 7 /VP9 segment read base MSB (SWREG80)

16.2.5.1.82.1 Offset

Register	Offset
SWREG80	140h

16.2.5.1.82.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR_7_MSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR_7_MSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.82.3 Fields

Field	Function
31-0 BASE_ADDR_7_MSB_BF	For HEVC: [31:0] - sw_refer7_ybase_msb - Base address MSB (bits 63:32) for reference luminance picture index 7 For VP9: [31:0] - sw_segment_read_base_msb - VP9 segment read base MSB

16.2.5.1.83 Base address LSB (bits 31:0) for reference luminance picture index 7 /VP9 segment read base LSB (SWREG81)

16.2.5.1.83.1 Offset

Register	Offset
SWREG81	144h

16.2.5.1.83.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR_7_LSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR_7_LSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.83.3 Fields

Field	Function
31-0 BASE_ADDR_7_LSB_BF	For HEVC: [31:0] - sw_refer7_ybase_lsb - Base address LSB (bits 31:0) for reference luminance picture index 7 For VP9: [31:0] - sw_segment_read_base_lsb - VP9 segment read base LSB

16.2.5.1.84 Base address MSB (bits 63:32) for reference luminance picture index 8 (SWREG82)

16.2.5.1.84.1 Offset

Register	Offset
SWREG82	148h

16.2.5.1.84.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.84.3 Fields

Field	Function
31-0 SW_REFER8_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 8

16.2.5.1.85 Base address LSB (bits 31:0) for reference luminance picture index 8 (SWREG83)

16.2.5.1.85.1 Offset

Register	Offset
SWREG83	14Ch

16.2.5.1.85.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.85.3 Fields

Field	Function
31-0 SW_REFER8_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 8

16.2.5.1.86 Base address MSB (bits 63:32) for reference luminance picture index 9 (SWREG84)

16.2.5.1.86.1 Offset

Register	Offset
SWREG84	150h

16.2.5.1.86.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.86.3 Fields

Field	Function
31-0 SW_REFER9_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 9

16.2.5.1.87 Base address LSB (bits 31:0) for reference luminance picture index 9 (SWREG85)

16.2.5.1.87.1 Offset

Register	Offset
SWREG85	154h

16.2.5.1.87.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.87.3 Fields

Field	Function
31-0 SW_REFER9_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 9

16.2.5.1.88 Base address MSB (bits 63:32) for reference luminance picture index 10 (SWREG86)

16.2.5.1.88.1 Offset

Register	Offset
SWREG86	158h

16.2.5.1.88.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.88.3 Fields

Field	Function
31-0 SW_REFER10_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 10

16.2.5.1.89 Base address LSB (bits 31:0) for reference luminance picture index 10 (SWREG87)

16.2.5.1.89.1 Offset

Register	Offset
SWREG87	15Ch

16.2.5.1.89.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.89.3 Fields

Field	Function
31-0 SW_REFER10_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 10

16.2.5.1.90 Base address MSB (bits 63:32) for reference luminance picture index 11 (SWREG88)

16.2.5.1.90.1 Offset

Register	Offset
SWREG88	160h

16.2.5.1.90.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.90.3 Fields

Field	Function
31-0 SW_REFER11_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 11

16.2.5.1.91 Base address LSB (bits 31:0) for reference luminance picture index 11 (SWREG89)

16.2.5.1.91.1 Offset

Register	Offset
SWREG89	164h

16.2.5.1.91.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.91.3 Fields

Field	Function
31-0 SW_REFER11_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 11

16.2.5.1.92 Base address MSB (bits 63:32) for reference luminance picture index 12 (SWREG90)

16.2.5.1.92.1 Offset

Register	Offset
SWREG90	168h

16.2.5.1.92.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.92.3 Fields

Field	Function
31-0 SW_REFER12_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 12

16.2.5.1.93 Base address LSB (bits 31:0) for reference luminance picture index 12 (SWREG91)

16.2.5.1.93.1 Offset

Register	Offset
SWREG91	16Ch

16.2.5.1.93.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.93.3 Fields

Field	Function
31-0 SW_REFER12_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 12

16.2.5.1.94 Base address MSB (bits 63:32) for reference luminance picture index 13 (SWREG92)

16.2.5.1.94.1 Offset

Register	Offset
SWREG92	170h

16.2.5.1.94.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.94.3 Fields

Field	Function
31-0 SW_REFER13_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 13

16.2.5.1.95 Base address LSB (bits 31:0) for reference luminance picture index 13 (SWREG93)

16.2.5.1.95.1 Offset

Register	Offset
SWREG93	174h

16.2.5.1.95.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.95.3 Fields

Field	Function
31-0 SW_REFER13_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 13

16.2.5.1.96 Base address MSB (bits 63:32) for reference luminance picture index 14 (SWREG94)

16.2.5.1.96.1 Offset

Register	Offset
SWREG94	178h

16.2.5.1.96.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.96.3 Fields

Field	Function
31-0 SW_REFER14_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 14

16.2.5.1.97 Base address LSB (bits 31:0) for reference luminance picture index 14 (SWREG95)

16.2.5.1.97.1 Offset

Register	Offset
SWREG95	17Ch

16.2.5.1.97.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.97.3 Fields

Field	Function
31-0 SW_REFER14_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 14

16.2.5.1.98 Base address MSB (bits 63:32) for reference luminance picture index 15 (SWREG96)

16.2.5.1.98.1 Offset

Register	Offset
SWREG96	180h

16.2.5.1.98.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.98.3 Fields

Field	Function
31-0 SW_REFER15_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 15

16.2.5.1.99 Base address LSB (bits 31:0) for reference luminance picture index 15 (SWREG97)

16.2.5.1.99.1 Offset

Register	Offset
SWREG97	184h

16.2.5.1.99.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.99.3 Fields

Field	Function
31-0 SW_REFER15_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 15

16.2.5.1.100 Base address MSB (bits 63:32) for decoded chrominance picture (SWREG98)

16.2.5.1.100.1 Offset

Register	Offset
SWREG98	188h

16.2.5.1.100.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.100.3 Fields

Field	Function
31-0 SW_DEC_OUT_CBASE_MSB	Base address MSB (bits 64:32) for decoded chrominance picture

16.2.5.1.101 Base address LSB (bits 31:0) for decoded chrominance picture (SWREG99)

16.2.5.1.101.1 Offset

Register	Offset
SWREG99	18Ch

16.2.5.1.101.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.101.3 Fields

Field	Function
31-0 SW_DEC_OUT_CBASE_LSB	Base address LSB (bits 31:0) for decoded chrominance picture

16.2.5.1.102 Base address MSB (bits 63:32) for reference chrominance picture index 0 (SWREG100)

16.2.5.1.102.1 Offset

Register	Offset
SWREG100	190h

16.2.5.1.102.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.102.3 Fields

Field	Function
31-0 SW_REFER0_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 0

16.2.5.1.103 Base address LSB (bits 31:0) for reference chrominance picture index 0 (SWREG101)

16.2.5.1.103.1 Offset

Register	Offset
SWREG101	194h

16.2.5.1.103.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.103.3 Fields

Field	Function
31-0 SW_REFER0_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 0

16.2.5.1.104 Base address MSB (bits 63:32) for reference chrominance picture index 1 (SWREG102)

16.2.5.1.104.1 Offset

Register	Offset
SWREG102	198h

16.2.5.1.104.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.104.3 Fields

Field	Function
31-0 SW_REFER1_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 1

16.2.5.1.105 Base address LSB (bits 31:0) for reference chrominance picture index 1 (SWREG103)

16.2.5.1.105.1 Offset

Register	Offset
SWREG103	19Ch

16.2.5.1.105.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.105.3 Fields

Field	Function
31-0 SW_REFER1_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 1

16.2.5.1.106 Base address MSB (bits 63:32) for reference chrominance picture index 2 (SWREG104)

16.2.5.1.106.1 Offset

Register	Offset
SWREG104	1A0h

16.2.5.1.106.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.106.3 Fields

Field	Function
31-0 SW_REFER2_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 2

16.2.5.1.107 Base address LSB (bits 31:0) for reference chrominance picture index 2 (SWREG105)

16.2.5.1.107.1 Offset

Register	Offset
SWREG105	1A4h

16.2.5.1.107.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.107.3 Fields

Field	Function
31-0 SW_REFER2_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 2

16.2.5.1.108 Base address MSB (bits 63:32) for reference chrominance picture index 3 (SWREG106)

16.2.5.1.108.1 Offset

Register	Offset
SWREG106	1A8h

16.2.5.1.108.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.108.3 Fields

Field	Function
31-0 SW_REFER3_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 3

16.2.5.1.109 Base address LSB (bits 31:0) for reference chrominance picture index 3 (SWREG107)

16.2.5.1.109.1 Offset

Register	Offset
SWREG107	1ACh

16.2.5.1.109.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.109.3 Fields

Field	Function
31-0 SW_REFER3_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 3

16.2.5.1.110 Base address MSB (bits 63:32) for reference chrominance picture index 4 (SWREG108)

16.2.5.1.110.1 Offset

Register	Offset
SWREG108	1B0h

16.2.5.1.110.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.110.3 Fields

Field	Function
31-0 SW_REFER4_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 4

16.2.5.1.111 Base address LSB (bits 31:0) for reference chrominance picture index 4 (SWREG109)

16.2.5.1.111.1 Offset

Register	Offset
SWREG109	1B4h

16.2.5.1.111.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.111.3 Fields

Field	Function
31-0 SW_REFER4_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 4

16.2.5.1.112 Base address MSB (bits 63:32) for reference chrominance picture index 5 (SWREG110)

16.2.5.1.112.1 Offset

Register	Offset
SWREG110	1B8h

16.2.5.1.112.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.112.3 Fields

Field	Function
31-0 SW_REFER5_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 5

16.2.5.1.113 Base address LSB (bits 31:0) for reference chrominance picture index 5 (SWREG111)

16.2.5.1.113.1 Offset

Register	Offset
SWREG111	1BCh

16.2.5.1.113.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.113.3 Fields

Field	Function
31-0 SW_REFER5_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 5

16.2.5.1.114 Base address MSB (bits 63:32) for reference chrominance picture index 6 (SWREG112)

16.2.5.1.114.1 Offset

Register	Offset
SWREG112	1C0h

16.2.5.1.114.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.114.3 Fields

Field	Function
31-0 SW_REFER6_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 6

16.2.5.1.115 Base address LSB (bits 31:0) for reference chrominance picture index 6 (SWREG113)

16.2.5.1.115.1 Offset

Register	Offset
SWREG113	1C4h

16.2.5.1.115.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.115.3 Fields

Field	Function
31-0 SW_REFER6_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 6

16.2.5.1.116 Base address MSB (bits 63:32) for reference chrominance picture index 7 (SWREG114)

16.2.5.1.116.1 Offset

Register	Offset
SWREG114	1C8h

16.2.5.1.116.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.116.3 Fields

Field	Function
31-0 SW_REFER7_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 7

16.2.5.1.117 Base address LSB (bits 31:0) for reference chrominance picture index 7 (SWREG115)

16.2.5.1.117.1 Offset

Register	Offset
SWREG115	1CCh

16.2.5.1.117.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.117.3 Fields

Field	Function
31-0 SW_REFER7_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 7

16.2.5.1.118 Base address MSB (bits 63:32) for reference chrominance picture index 8 (SWREG116)

16.2.5.1.118.1 Offset

Register	Offset
SWREG116	1D0h

16.2.5.1.118.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.118.3 Fields

Field	Function
31-0 SW_REFER8_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 8

16.2.5.1.119 Base address LSB (bits 31:0) for reference chrominance picture index 8 (SWREG117)

16.2.5.1.119.1 Offset

Register	Offset
SWREG117	1D4h

16.2.5.1.119.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.119.3 Fields

Field	Function
31-0 SW_REFER8_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 8

16.2.5.1.120 Base address MSB (bits 63:32) for reference chrominance picture index 9 (SWREG118)

16.2.5.1.120.1 Offset

Register	Offset
SWREG118	1D8h

16.2.5.1.120.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.120.3 Fields

Field	Function
31-0 SW_REFER9_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 9

16.2.5.1.121 Base address LSB (bits 31:0) for reference chrominance picture index 9 (SWREG119)

16.2.5.1.121.1 Offset

Register	Offset
SWREG119	1DCh

16.2.5.1.121.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.121.3 Fields

Field	Function
31-0 SW_REFER9_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 9

16.2.5.1.122 Base address MSB (bits 63:32) for reference chrominance picture index 10 (SWREG120)

16.2.5.1.122.1 Offset

Register	Offset
SWREG120	1E0h

16.2.5.1.122.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.122.3 Fields

Field	Function
31-0 SW_REFER10_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 10

16.2.5.1.123 Base address LSB (bits 31:0) for reference chrominance picture index 10 (SWREG121)

16.2.5.1.123.1 Offset

Register	Offset
SWREG121	1E4h

16.2.5.1.123.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.123.3 Fields

Field	Function
31-0 SW_REFER10_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 10

16.2.5.1.124 Base address MSB (bits 63:32) for reference chrominance picture index 11 (SWREG122)

16.2.5.1.124.1 Offset

Register	Offset
SWREG122	1E8h

16.2.5.1.124.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.124.3 Fields

Field	Function
31-0 SW_REFER11_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 11

16.2.5.1.125 Base address LSB (bits 31:0) for reference chrominance picture index 11 (SWREG123)

16.2.5.1.125.1 Offset

Register	Offset
SWREG123	1ECh

16.2.5.1.125.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.125.3 Fields

Field	Function
31-0 SW_REFER11_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 11

16.2.5.1.126 Base address MSB (bits 63:32) for reference chrominance picture index 12 (SWREG124)

16.2.5.1.126.1 Offset

Register	Offset
SWREG124	1F0h

16.2.5.1.126.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.126.3 Fields

Field	Function
31-0 SW_REFER12_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 12

16.2.5.1.127 Base address LSB (bits 31:0) for reference chrominance picture index 12 (SWREG125)

16.2.5.1.127.1 Offset

Register	Offset
SWREG125	1F4h

16.2.5.1.127.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.127.3 Fields

Field	Function
31-0 SW_REFER12_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 12

16.2.5.1.128 Base address MSB (bits 63:32) for reference chrominance picture index 13 (SWREG126)

16.2.5.1.128.1 Offset

Register	Offset
SWREG126	1F8h

16.2.5.1.128.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.128.3 Fields

Field	Function
31-0 SW_REFER13_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 13

16.2.5.1.129 Base address LSB (bits 31:0) for reference chrominance picture index 13 (SWREG127)

16.2.5.1.129.1 Offset

Register	Offset
SWREG127	1FCh

16.2.5.1.129.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.129.3 Fields

Field	Function
31-0 SW_REFER13_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 13

16.2.5.1.130 Base address MSB (bits 63:32) for reference chrominance picture index 14 (SWREG128)

16.2.5.1.130.1 Offset

Register	Offset
SWREG128	200h

16.2.5.1.130.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.130.3 Fields

Field	Function
31-0 SW_REFER14_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 14

16.2.5.1.131 Base address LSB (bits 31:0) for reference chrominance picture index 14 (SWREG129)

16.2.5.1.131.1 Offset

Register	Offset
SWREG129	204h

16.2.5.1.131.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.131.3 Fields

Field	Function
31-0 SW_REFER14_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 14

16.2.5.1.132 Base address MSB (bits 63:32) for reference chrominance picture index 15 (SWREG130)

16.2.5.1.132.1 Offset

Register	Offset
SWREG130	208h

16.2.5.1.132.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.132.3 Fields

Field	Function
31-0 SW_REFER15_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 15

16.2.5.1.133 Base address LSB (bits 31:0) for reference chrominance picture index 15 (SWREG131)

16.2.5.1.133.1 Offset

Register	Offset
SWREG131	20Ch

16.2.5.1.133.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.133.3 Fields

Field	Function
31-0 SW_REFER15_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 15

16.2.5.1.134 Base address MSB (bits 63:32) for decoded direct mode MVS (SWREG132)

16.2.5.1.134.1 Offset

Register	Offset
SWREG132	210h

16.2.5.1.134.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.134.3 Fields

Field	Function
31-0 SW_DEC_OUT_DBASE_MSB	Base address MSB (bits 63:32) for decoded direct mode MVS

16.2.5.1.135 Base address LSB (bits 31:0) for decoded direct mode MVS (SWREG133)

16.2.5.1.135.1 Offset

Register	Offset
SWREG133	214h

16.2.5.1.135.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.135.3 Fields

Field	Function
31-0 SW_DEC_OUT_DBASE_LSB	Base address LSB (bits 31:0) for decoded direct mode MVS

16.2.5.1.136 Base address MSB (bits 63:32) for reference direct mode MVS index 0 (SWREG134)

16.2.5.1.136.1 Offset

Register	Offset
SWREG134	218h

16.2.5.1.136.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.136.3 Fields

Field	Function
31-0 SW_REFER0_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 0

16.2.5.1.137 Base address LSB (bits 31:0) for reference direct mode MVS index 0 (SWREG135)

16.2.5.1.137.1 Offset

Register	Offset
SWREG135	21Ch

16.2.5.1.137.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.137.3 Fields

Field	Function
31-0 SW_REFER0_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 0

16.2.5.1.138 Base address MSB (bits 63:32) for reference direct mode MVS index 1 (SWREG136)

16.2.5.1.138.1 Offset

Register	Offset
SWREG136	220h

16.2.5.1.138.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.138.3 Fields

Field	Function
31-0 SW_REFER1_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 1

16.2.5.1.139 Base address LSB (bits 31:0) for reference direct mode MVS index 1 (SWREG137)

16.2.5.1.139.1 Offset

Register	Offset
SWREG137	224h

16.2.5.1.139.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.139.3 Fields

Field	Function
31-0 SW_REFER1_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 1

16.2.5.1.140 Base address MSB (bits 63:32) for reference direct mode MVS index 2 (SWREG138)

16.2.5.1.140.1 Offset

Register	Offset
SWREG138	228h

16.2.5.1.140.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.140.3 Fields

Field	Function
31-0 SW_REFER2_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 2

16.2.5.1.141 Base address LSB (bits 31:0) for reference direct mode MVS index 2 (SWREG139)

16.2.5.1.141.1 Offset

Register	Offset
SWREG139	22Ch

16.2.5.1.141.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.141.3 Fields

Field	Function
31-0 SW_REFER2_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 2

16.2.5.1.142 Base address MSB (bits 63:32) for reference direct mode MVS index 3 (SWREG140)

16.2.5.1.142.1 Offset

Register	Offset
SWREG140	230h

16.2.5.1.142.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.142.3 Fields

Field	Function
31-0 SW_REFER3_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 3

16.2.5.1.143 Base address LSB (bits 31:0) for reference direct mode MVS index 3 (SWREG141)

16.2.5.1.143.1 Offset

Register	Offset
SWREG141	234h

16.2.5.1.143.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.143.3 Fields

Field	Function
31-0 SW_REFER3_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 3

16.2.5.1.144 Base address MSB (bits 63:32) for reference direct mode MVS index 4 (SWREG142)

16.2.5.1.144.1 Offset

Register	Offset
SWREG142	238h

16.2.5.1.144.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.144.3 Fields

Field	Function
31-0 SW_REFER4_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 4

16.2.5.1.145 Base address LSB (bits 31:0) for reference direct mode MVS index 4 (SWREG143)

16.2.5.1.145.1 Offset

Register	Offset
SWREG143	23Ch

16.2.5.1.145.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.145.3 Fields

Field	Function
31-0 SW_REFER4_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 4

16.2.5.1.146 Base address MSB (bits 63:32) for reference direct mode MVS index 5 (SWREG144)

16.2.5.1.146.1 Offset

Register	Offset
SWREG144	240h

16.2.5.1.146.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.146.3 Fields

Field	Function
31-0 SW_REFER5_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 5

16.2.5.1.147 Base address LSB (bits 31:0) for reference direct mode MVS index 5 (SWREG145)

16.2.5.1.147.1 Offset

Register	Offset
SWREG145	244h

16.2.5.1.147.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.147.3 Fields

Field	Function
31-0 SW_REFER5_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 5

16.2.5.1.148 Base address MSB (bits 63:32) for reference direct mode MVS index 6 (SWREG146)

16.2.5.1.148.1 Offset

Register	Offset
SWREG146	248h

16.2.5.1.148.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.148.3 Fields

Field	Function
31-0 SW_REFER6_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 6

16.2.5.1.149 Base address LSB (bits 31:0) for reference direct mode MVS index 6 (SWREG147)

16.2.5.1.149.1 Offset

Register	Offset
SWREG147	24Ch

16.2.5.1.149.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.149.3 Fields

Field	Function
31-0 SW_REFER6_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 6

16.2.5.1.150 Base address MSB (bits 63:32) for reference direct mode MVS index 7 (SWREG148)

16.2.5.1.150.1 Offset

Register	Offset
SWREG148	250h

16.2.5.1.150.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.150.3 Fields

Field	Function
31-0 SW_REFER7_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 7

16.2.5.1.151 Base address LSB (bits 31:0) for reference direct mode MVS index 7 (SWREG149)

16.2.5.1.151.1 Offset

Register	Offset
SWREG149	254h

16.2.5.1.151.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.151.3 Fields

Field	Function
31-0 SW_REFER7_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 7

16.2.5.1.152 Base address MSB (bits 63:32) for reference direct mode MVS index 8 (SWREG150)

16.2.5.1.152.1 Offset

Register	Offset
SWREG150	258h

16.2.5.1.152.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.152.3 Fields

Field	Function
31-0 SW_REFER8_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 8

16.2.5.1.153 Base address LSB (bits 31:0) for reference direct mode MVS index 8 (SWREG151)

16.2.5.1.153.1 Offset

Register	Offset
SWREG151	25Ch

16.2.5.1.153.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.153.3 Fields

Field	Function
31-0 SW_REFER8_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode mode MVS index 8

16.2.5.1.154 Base address MSB (bits 63:32) for reference direct mode mode MVS index 9 (SWREG152)

16.2.5.1.154.1 Offset

Register	Offset
SWREG152	260h

16.2.5.1.154.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.154.3 Fields

Field	Function
31-0 SW_REFER9_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 9

16.2.5.1.155 Base address LSB (bits 31:0) for reference direct mode mode MVS index 9 (SWREG153)

16.2.5.1.155.1 Offset

Register	Offset
SWREG153	264h

16.2.5.1.155.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.155.3 Fields

Field	Function
31-0 SW_REFER9_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode mode MVS index 9

16.2.5.1.156 Base address MSB (bits 63:32) for reference direct mode MVS index 10 (SWREG154)

16.2.5.1.156.1 Offset

Register	Offset
SWREG154	268h

16.2.5.1.156.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.156.3 Fields

Field	Function
31-0 SW_REFER10_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 10

16.2.5.1.157 Base address LSB (bits 31:0) for reference direct mode MVS index 10 (SWREG155)

16.2.5.1.157.1 Offset

Register	Offset
SWREG155	26Ch

16.2.5.1.157.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.157.3 Fields

Field	Function
31-0 SW_REFER10_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 10

16.2.5.1.158 Base address MSB (bits 63:32) for reference direct mode MVS index 11 (SWREG156)

16.2.5.1.158.1 Offset

Register	Offset
SWREG156	270h

16.2.5.1.158.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.158.3 Fields

Field	Function
31-0 SW_REFER11_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 11

16.2.5.1.159 Base address LSB (bits 31:0) for reference direct mode MVS index 11 (SWREG157)

16.2.5.1.159.1 Offset

Register	Offset
SWREG157	274h

16.2.5.1.159.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.159.3 Fields

Field	Function
31-0 SW_REFER11_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 11

16.2.5.1.160 Base address MSB (bits 63:32) for reference direct mode MVS index 12 (SWREG158)

16.2.5.1.160.1 Offset

Register	Offset
SWREG158	278h

16.2.5.1.160.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.160.3 Fields

Field	Function
31-0 SW_REFER12_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 12

16.2.5.1.161 Base address LSB (bits 31:0) for reference direct mode MVS index 12 (SWREG159)

16.2.5.1.161.1 Offset

Register	Offset
SWREG159	27Ch

16.2.5.1.161.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.161.3 Fields

Field	Function
31-0 SW_REFER12_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 12

16.2.5.1.162 Base address MSB (bits 63:32) for reference direct mode MVS index 13 (SWREG160)

16.2.5.1.162.1 Offset

Register	Offset
SWREG160	280h

16.2.5.1.162.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.162.3 Fields

Field	Function
31-0 SW_REFER13_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 13

16.2.5.1.163 Base address LSB (bits 31:0) for reference direct mode MVS index 13 (SWREG161)

16.2.5.1.163.1 Offset

Register	Offset
SWREG161	284h

16.2.5.1.163.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.163.3 Fields

Field	Function
31-0 SW_REFER13_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 13

16.2.5.1.164 Base address MSB (bits 63:32) for reference direct mode MVS index 14 (SWREG162)

16.2.5.1.164.1 Offset

Register	Offset
SWREG162	288h

16.2.5.1.164.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.164.3 Fields

Field	Function
31-0 SW_REFER14_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 14

16.2.5.1.165 Base address LSB (bits 31:0) for reference direct mode MVS index 14 (SWREG163)

16.2.5.1.165.1 Offset

Register	Offset
SWREG163	28Ch

16.2.5.1.165.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.165.3 Fields

Field	Function
31-0 SW_REFER14_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 14

16.2.5.1.166 Base address MSB (bits 63:32) for reference direct mode MVS index 15 (SWREG164)

16.2.5.1.166.1 Offset

Register	Offset
SWREG164	290h

16.2.5.1.166.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.166.3 Fields

Field	Function
31-0 SW_REFER15_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 15

16.2.5.1.167 Base address LSB (bits 31:0) for reference direct mode MVS index 15 (SWREG165)

16.2.5.1.167.1 Offset

Register	Offset
SWREG165	294h

16.2.5.1.167.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.167.3 Fields

Field	Function
31-0 SW_REFER15_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 15

16.2.5.1.168 Base address MSB (bits 63:32) for tile sizes (SWREG166)

16.2.5.1.168.1 Offset

Register	Offset
SWREG166	298h

16.2.5.1.168.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_TILE_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_TILE_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.168.3 Fields

Field	Function
31-0 SW_TILE_BASE_MSB	Base address MSB (bits 63:32) for tile sizes

16.2.5.1.169 Base address LSB (bits 31:0) for tile sizes (SWREG167)

16.2.5.1.169.1 Offset

Register	Offset
SWREG167	29Ch

16.2.5.1.169.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_TILE_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_TILE_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.169.3 Fields

Field	Function
31-0 SW_TILE_BASE_LSB	Base address LSB (bits 31:0) for tile sizes

16.2.5.1.170 Base address MSB (bits 63:32) for / stream start address/ decoded end addr register (SWREG168)

16.2.5.1.170.1 Offset

Register	Offset
SWREG168	2A0h

16.2.5.1.170.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_STREAM_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_STREAM_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.170.3 Fields

Field	Function
31-0 SW_STREAM_BASE_MSB	Base address MSB (bits 63:32) for / stream start address/decoded end addr register

16.2.5.1.171 Base address LSB (bits 31:0) for / stream start address/ decoded end addr register (SWREG169)

16.2.5.1.171.1 Offset

Register	Offset
SWREG169	2A4h

16.2.5.1.171.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_STREAM_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_STREAM_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.171.3 Fields

Field	Function
31-0 SW_STREAM_BASE_LSB	Base address LSB (bits 31:0) for / stream start address/decoded end addr register

16.2.5.1.172 Base address MSB (bits 63:32) for scaling lists / VP9 CTX counter values (SWREG170)

16.2.5.1.172.1 Offset

Register	Offset
SWREG170	2A8h

16.2.5.1.172.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_SCALE_LIST_CTX_COUNTER_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_SCALE_LIST_CTX_COUNTER_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.172.3 Fields

Field	Function
31-0 SW_SCALE_LIST_CTX_COUNTER_BASE_MSB	HEVC: Base address MSB (bits 63:32) for scaling lists VP9: CTX counter values

16.2.5.1.173 Base address LSB (bits 31:0) for scaling lists / VP9 CTX counter values (SWREG171)

16.2.5.1.173.1 Offset

Register	Offset
SWREG171	2ACh

16.2.5.1.173.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_SCALE_LISTT_CTX_COUNTER_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_SCALE_LISTT_CTX_COUNTER_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.173.3 Fields

Field	Function
31-0 SW_SCALE_LISTT_CTX_COUNTER_BASE_LSB	HEVC: Base address LSB (bits 31:0) for scaling lists VP9: CTX counter values

16.2.5.1.174 Base address MSB (bits 63:32) for stream propability tables (SWREG172)

16.2.5.1.174.1 Offset

Register	Offset
SWREG172	2B0h

16.2.5.1.174.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_PROB_TAB_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_PROB_TAB_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.174.3 Fields

Field	Function
31-0 SW_PROB_TAB_BASE_MSB	Base address MSB (bits 63:32) for stream propability tables

16.2.5.1.175 Base address LSB (bits 31:0) for stream propability tables (SWREG173)

16.2.5.1.175.1 Offset

Register	Offset
SWREG173	2B4h

16.2.5.1.175.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_PROB_TAB_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_PROB_TAB_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.175.3 Fields

Field	Function
31-0 SW_PROB_TAB_BASE_LSB	Base address LSB (bits 31:0) for stream propability tables

16.2.5.1.176 Base address MSB (bits 63:32) for decoder output raster scan Y picture (SWREG174)

16.2.5.1.176.1 Offset

Register	Offset
SWREG174	2B8h

16.2.5.1.176.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_RSY_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_RSY_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.176.3 Fields

Field	Function
31-0 SW_DEC_RSY_BASE_MSB	Base address MSB (bits 63:32) for decoder output raster scan Y picture

16.2.5.1.177 Base address LSB (bits 31:0) for decoder output raster scan Y picture (SWREG175)

16.2.5.1.177.1 Offset

Register	Offset
SWREG175	2BCh

16.2.5.1.177.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_RSY_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_RSY_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.177.3 Fields

Field	Function
31-0 SW_DEC_RSY_BASE_LSB	Base address LSB (bits 31:0) for decoder output raster scan Y picture

16.2.5.1.178 Base address MSB (bits 63:32) for decoder output raster scan C picture (SWREG176)

16.2.5.1.178.1 Offset

Register	Offset
SWREG176	2C0h

16.2.5.1.178.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_RSC_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_RSC_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.178.3 Fields

Field	Function
31-0 SW_DEC_RSC_BASE_MSB	Base address MSB (bits 63:32) for decoder output raster scan C picture

16.2.5.1.179 Base address LSB (bits 31:0) for decoder output raster scan C picture (SWREG177)

16.2.5.1.179.1 Offset

Register	Offset
SWREG177	2C4h

16.2.5.1.179.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_RSC_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_RSC_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.179.3 Fields

Field	Function
31-0 SW_DEC_RSC_BASE_LSB	Base address LSB (bits 31:0) for decoder output raster scan C picture

16.2.5.1.180 Base address MSB (bits 63:32) for tile border coeffients of filter (SWREG178)

16.2.5.1.180.1 Offset

Register	Offset
SWREG178	2C8h

16.2.5.1.180.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_VERT_FILT_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_VERT_FILT_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.180.3 Fields

Field	Function
31-0 SW_DEC_VERT_FILT_BASE_MSB	Base address MSB to store/read filtering coeffients of current picture at tile border.

16.2.5.1.181 Base address LSB (bits 31:0) for tile border coeffients of filter (SWREG179)

16.2.5.1.181.1 Offset

Register	Offset
SWREG179	2CCh

16.2.5.1.181.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_VERT_FILT_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_VERT_FILT_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.181.3 Fields

Field	Function
31-0 SW_DEC_VERT_FILT_BASE_LSB	Base address LSB to store/read filtering coeffients of current picture at tile border.

16.2.5.1.182 Base address MSB (bits 63:32) for tile border coeffients of sao (SWREG180)

16.2.5.1.182.1 Offset

Register	Offset
SWREG180	2D0h

16.2.5.1.182.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_VERT_SAO_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_VERT_SAO_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.182.3 Fields

Field	Function
31-0 SW_DEC_VERT_SAO_BASE_MSB	Base address MSB to store/read sao coeffients of current picture at tile border.

16.2.5.1.183 Base address LSB (bits 31:0) for tile border coeffients of sao (SWREG181)

16.2.5.1.183.1 Offset

Register	Offset
SWREG181	2D4h

16.2.5.1.183.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_VERT_SAO_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_VERT_SAO_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.183.3 Fields

Field	Function
31-0 SW_DEC_VERT_SAO_BASE_LSB	Base address LSB to store/read sao coeffients of current picture at tile border.

16.2.5.1.184 Base address MSB (bits 63:32) for tile border bsd control data (SWREG182)

16.2.5.1.184.1 Offset

Register	Offset
SWREG182	2D8h

16.2.5.1.184.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_BSD_CTRL_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_BSD_CTRL_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.184.3 Fields

Field	Function
31-0 SW_DEC_BSD_CTRL_BASE_MSB	Base address MSB to store/read BSD control data of current picture at tile border.

16.2.5.1.185 Base address LSB (bits 31:0) for tile border bsd control data (SWREG183)

16.2.5.1.185.1 Offset

Register	Offset
SWREG183	2DCh

16.2.5.1.185.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_BSD_CTRL_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_BSD_CTRL_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.185.3 Fields

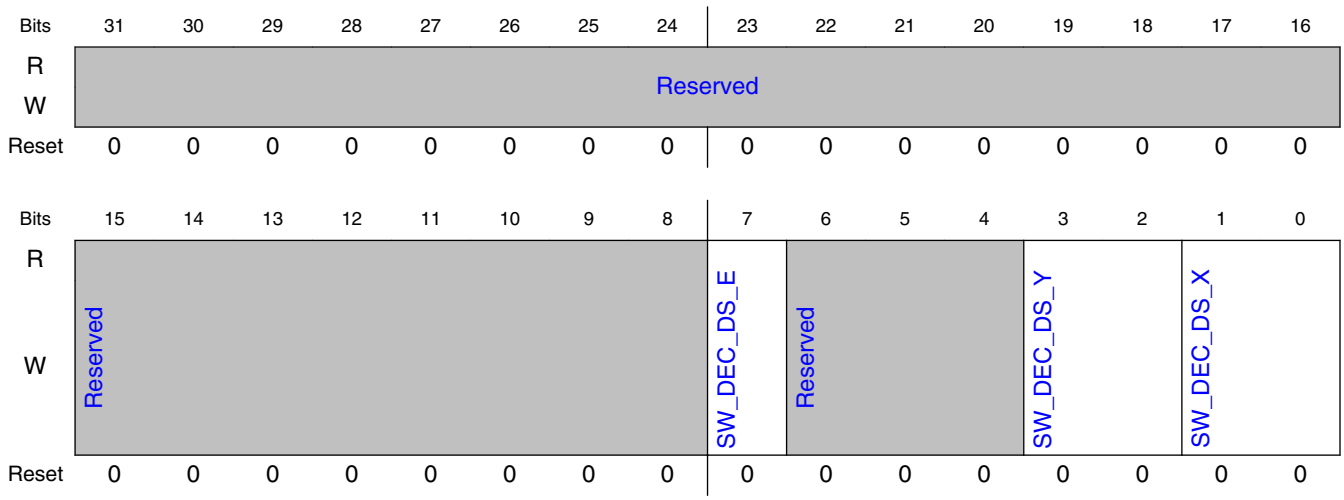
Field	Function
31-0 SW_DEC_BSD_CTRL_BASE_LSB	Base address LSB to store/read BSD control data of current picture at tile border.

16.2.5.1.186 Raster scan down scale control register MSM (SWREG184)

16.2.5.1.186.1 Offset

Register	Offset
SWREG184	2E0h

16.2.5.1.186.2 Diagram



16.2.5.1.186.3 Fields

Field	Function
31-8 —	Reserved.
7 SW_DEC_DS_E	Raster scan down scale enable 0b - Disable 1b - Enable
6-4 —	Reserved.
3-2 SW_DEC_DS_Y	Y coordinate down scale times for raster scan output picture data 00b - 1/2 01b - 1/4 10b - 1/8
1-0 SW_DEC_DS_X	X coordinate down scale times for raster scan output picture data 00b - 1/2 01b - 1/4 10b - 1/8

16.2.5.1.187 Base address MSB (bits 63:32) for decoder output raster scan down scale Y picture (SWREG185)

16.2.5.1.187.1 Offset

Register	Offset
SWREG185	2E4h

16.2.5.1.187.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_DSY_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_DSY_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.187.3 Fields

Field	Function
31-0 SW_DEC_DSY_BASE_MSB	Base address MSB (bits 63:32) for decoder output raster scan down scale Y picture

16.2.5.1.188 Base address LSB (bits 31:0) for decoder output raster scan down scale Y picture (SWREG186)

16.2.5.1.188.1 Offset

Register	Offset
SWREG186	2E8h

16.2.5.1.188.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_DSY_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_DSY_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.188.3 Fields

Field	Function
31-0 SW_DEC_DSY_BASE_LSB	Base address LSB (bits 31:0) for decoder output raster scan down scale Y picture

16.2.5.1.189 Base address MSB (bits 63:32) for decoder output raster scan down scale C picture (SWREG187)

16.2.5.1.189.1 Offset

Register	Offset
SWREG187	2ECh

16.2.5.1.189.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_DSC_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_DSC_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.189.3 Fields

Field	Function
31-0 SW_DEC_DSC_BASE_MSB	Base address MSB (bits 63:32) for decoder output raster scan down scale C picture

16.2.5.1.190 Base address LSB (bits 31:0) for decoder output raster scan down scale C picture (SWREG188)

16.2.5.1.190.1 Offset

Register	Offset
SWREG188	2F0h

16.2.5.1.190.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_DSC_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_DSC_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.190.3 Fields

Field	Function
31-0 SW_DEC_DSC_BASE_LSB	Base address LSB (bits 31:0) for decoder output raster scan down scale C picture

16.2.5.1.191 Base address MSB (bits 63:32) for decoder output compress luminance table (SWREG189)

16.2.5.1.191.1 Offset

Register	Offset
SWREG189	2F4h

16.2.5.1.191.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.191.3 Fields

Field	Function
31-0 SW_DEC_OUT_TYBASE_MSB	Base address MSB (bits 63:32) for decoder output compress luminance table

16.2.5.1.192 Base address LSB (bits 31:0) for decoder output compress luminance table (SWREG190)

16.2.5.1.192.1 Offset

Register	Offset
SWREG190	2F8h

16.2.5.1.192.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.192.3 Fields

Field	Function
31-0 SW_DEC_OUT_TYBASE_LSB	Base address LSB (bits 31:0) for decoder output compress luminance table

16.2.5.1.193 Base address MSB (bits 63:32) for reference compress luminance table index 0 (SWREG191)

16.2.5.1.193.1 Offset

Register	Offset
SWREG191	2FCh

16.2.5.1.193.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.193.3 Fields

Field	Function
31-0 SW_REFER0_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 0

16.2.5.1.194 Base address LSB (bits 31:0) for reference compress luminance table index 0 (SWREG192)

16.2.5.1.194.1 Offset

Register	Offset
SWREG192	300h

16.2.5.1.194.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.194.3 Fields

Field	Function
31-0 SW_REFER0_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 0

16.2.5.1.195 Base address MSB (bits 63:32) for reference compress luminance table index 1 (SWREG193)

16.2.5.1.195.1 Offset

Register	Offset
SWREG193	304h

16.2.5.1.195.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.195.3 Fields

Field	Function
31-0 SW_REFER1_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 1

16.2.5.1.196 Base address LSB (bits 31:0) for reference compress luminance table index 1 (SWREG194)

16.2.5.1.196.1 Offset

Register	Offset
SWREG194	308h

16.2.5.1.196.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.196.3 Fields

Field	Function
31-0 SW_REFER1_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 1

16.2.5.1.197 Base address MSB (bits 63:32) for reference compress luminance table index 2 (SWREG195)

16.2.5.1.197.1 Offset

Register	Offset
SWREG195	30Ch

16.2.5.1.197.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.197.3 Fields

Field	Function
31-0 SW_REFER2_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 2

16.2.5.1.198 Base address LSB (bits 31:0) for reference compress luminance table index 2 (SWREG196)

16.2.5.1.198.1 Offset

Register	Offset
SWREG196	310h

16.2.5.1.198.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.198.3 Fields

Field	Function
31-0 SW_REFER2_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 2

16.2.5.1.199 Base address MSB (bits 63:32) for reference compress luminance table index 3 (SWREG197)

16.2.5.1.199.1 Offset

Register	Offset
SWREG197	314h

16.2.5.1.199.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.199.3 Fields

Field	Function
31-0 SW_REFER3_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 3

16.2.5.1.200 Base address LSB (bits 31:0) for reference compress luminance table index 3 (SWREG198)

16.2.5.1.200.1 Offset

Register	Offset
SWREG198	318h

16.2.5.1.200.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.200.3 Fields

Field	Function
31-0 SW_REFER3_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 3

16.2.5.1.201 Base address MSB (bits 63:32) for reference compress luminance table index 4 (SWREG199)

16.2.5.1.201.1 Offset

Register	Offset
SWREG199	31Ch

16.2.5.1.201.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.201.3 Fields

Field	Function
31-0 SW_REFER4_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 4

16.2.5.1.202 Base address LSB (bits 31:0) for reference compress luminance table index 4 (SWREG200)

16.2.5.1.202.1 Offset

Register	Offset
SWREG200	320h

16.2.5.1.202.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.202.3 Fields

Field	Function
31-0 SW_REFER4_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 4

16.2.5.1.203 Base address MSB (bits 63:32) for reference compress luminance table index 5 (SWREG201)

16.2.5.1.203.1 Offset

Register	Offset
SWREG201	324h

16.2.5.1.203.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.203.3 Fields

Field	Function
31-0 SW_REFER5_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 5

16.2.5.1.204 Base address LSB (bits 31:0) for reference compress luminance table index 5 (SWREG202)

16.2.5.1.204.1 Offset

Register	Offset
SWREG202	328h

16.2.5.1.204.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.204.3 Fields

Field	Function
31-0 SW_REFER5_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 5

16.2.5.1.205 Base address MSB (bits 63:32) for reference compress luminance table index 6 (SWREG203)

16.2.5.1.205.1 Offset

Register	Offset
SWREG203	32Ch

16.2.5.1.205.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.205.3 Fields

Field	Function
31-0 SW_REFER6_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 6

16.2.5.1.206 Base address LSB (bits 31:0) for reference compress luminance table index 6 (SWREG204)

16.2.5.1.206.1 Offset

Register	Offset
SWREG204	330h

16.2.5.1.206.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.206.3 Fields

Field	Function
31-0 SW_REFER6_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 6

16.2.5.1.207 Base address MSB (bits 63:32) for reference compress luminance table index 7 (SWREG205)

16.2.5.1.207.1 Offset

Register	Offset
SWREG205	334h

16.2.5.1.207.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.207.3 Fields

Field	Function
31-0 SW_REFER7_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 7

16.2.5.1.208 Base address LSB (bits 31:0) for reference compress luminance table index 7 (SWREG206)

16.2.5.1.208.1 Offset

Register	Offset
SWREG206	338h

16.2.5.1.208.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.208.3 Fields

Field	Function
31-0 SW_REFER7_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 7

16.2.5.1.209 Base address MSB (bits 63:32) for reference compress luminance table index 8 (SWREG207)

16.2.5.1.209.1 Offset

Register	Offset
SWREG207	33Ch

16.2.5.1.209.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.209.3 Fields

Field	Function
31-0 SW_REFER8_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 8

16.2.5.1.210 Base address LSB (bits 31:0) for reference compress luminance table index 8 (SWREG208)

16.2.5.1.210.1 Offset

Register	Offset
SWREG208	340h

16.2.5.1.210.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.210.3 Fields

Field	Function
31-0 SW_REFER8_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 8

16.2.5.1.211 Base address MSB (bits 63:32) for reference compress luminance table index 9 (SWREG209)

16.2.5.1.211.1 Offset

Register	Offset
SWREG209	344h

16.2.5.1.211.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.211.3 Fields

Field	Function
31-0 SW_REFER9_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 9

16.2.5.1.212 Base address LSB (bits 31:0) for reference compress luminance table index 9 (SWREG210)

16.2.5.1.212.1 Offset

Register	Offset
SWREG210	348h

16.2.5.1.212.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.212.3 Fields

Field	Function
31-0 SW_REFER9_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 9

16.2.5.1.213 Base address MSB (bits 63:32) for reference compress luminance table index 10 (SWREG211)

16.2.5.1.213.1 Offset

Register	Offset
SWREG211	34Ch

16.2.5.1.213.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.213.3 Fields

Field	Function
31-0 SW_REFER10_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 10

16.2.5.1.214 Base address LSB (bits 31:0) for reference compress luminance table index 10 (SWREG212)

16.2.5.1.214.1 Offset

Register	Offset
SWREG212	350h

16.2.5.1.214.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.214.3 Fields

Field	Function
31-0 SW_REFER10_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 10

16.2.5.1.215 Base address MSB (bits 63:32) for reference compress luminance table index 11 (SWREG213)

16.2.5.1.215.1 Offset

Register	Offset
SWREG213	354h

16.2.5.1.215.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.215.3 Fields

Field	Function
31-0 SW_REFER11_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 11

16.2.5.1.216 Base address LSB (bits 31:0) for reference compress luminance table index 11 (SWREG214)

16.2.5.1.216.1 Offset

Register	Offset
SWREG214	358h

16.2.5.1.216.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.216.3 Fields

Field	Function
31-0 SW_REFER11_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 11

16.2.5.1.217 Base address MSB (bits 63:32) for reference compress luminance table index 12 (SWREG215)

16.2.5.1.217.1 Offset

Register	Offset
SWREG215	35Ch

16.2.5.1.217.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.217.3 Fields

Field	Function
31-0 SW_REFER12_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 12

16.2.5.1.218 Base address LSB (bits 31:0) for reference compress luminance table index 12 (SWREG216)

16.2.5.1.218.1 Offset

Register	Offset
SWREG216	360h

16.2.5.1.218.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.218.3 Fields

Field	Function
31-0 SW_REFER12_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 12

16.2.5.1.219 Base address MSB (bits 63:32) for reference compress luminance table index 13 (SWREG217)

16.2.5.1.219.1 Offset

Register	Offset
SWREG217	364h

16.2.5.1.219.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.219.3 Fields

Field	Function
31-0 SW_REFER13_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 13

16.2.5.1.220 Base address LSB (bits 31:0) for reference compress luminance table index 13 (SWREG218)

16.2.5.1.220.1 Offset

Register	Offset
SWREG218	368h

16.2.5.1.220.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.220.3 Fields

Field	Function
31-0 SW_REFER13_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 13

16.2.5.1.221 Base address MSB (bits 63:32) for reference compress luminance table index 14 (SWREG219)

16.2.5.1.221.1 Offset

Register	Offset
SWREG219	36Ch

16.2.5.1.221.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.221.3 Fields

Field	Function
31-0 SW_REFER14_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 14

16.2.5.1.222 Base address LSB (bits 31:0) for reference compress luminance table index 14 (SWREG220)

16.2.5.1.222.1 Offset

Register	Offset
SWREG220	370h

16.2.5.1.222.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.222.3 Fields

Field	Function
31-0 SW_REFER14_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 14

16.2.5.1.223 Base address MSB (bits 63:32) for reference compress luminance table index 15 (SWREG221)

16.2.5.1.223.1 Offset

Register	Offset
SWREG221	374h

16.2.5.1.223.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.223.3 Fields

Field	Function
31-0 SW_REFER15_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 15

16.2.5.1.224 Base address LSB (bits 31:0) for reference compress luminance table index 15 (SWREG222)

16.2.5.1.224.1 Offset

Register	Offset
SWREG222	378h

16.2.5.1.224.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.224.3 Fields

Field	Function
31-0 SW_REFER15_ TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 15

16.2.5.1.225 Base address MSB (bits 63:32) for decoder output compress chrominance table (SWREG223)

16.2.5.1.225.1 Offset

Register	Offset
SWREG223	37Ch

16.2.5.1.225.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.225.3 Fields

Field	Function
31-0 SW_DEC_OUT _TCBASE_MSB	Base address MSB (bits 63:32) for decoder output compress chrominance table

16.2.5.1.226 Base address LSB (bits 31:0) for decoder output compress chrominance table (SWREG224)

16.2.5.1.226.1 Offset

Register	Offset
SWREG224	380h

16.2.5.1.226.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.226.3 Fields

Field	Function
31-0 SW_DEC_OUT_TCBASE_LSB	Base address LSB (bits 31:0) for decoder output compress chrominance table

16.2.5.1.227 Base address MSB (bits 63:32) for reference compress chrominance table index 0 (SWREG225)

16.2.5.1.227.1 Offset

Register	Offset
SWREG225	384h

16.2.5.1.227.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.227.3 Fields

Field	Function
31-0 SW_REFER0_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 0

16.2.5.1.228 Base address LSB (bits 31:0) for reference compress chrominance table index 0 (SWREG226)

16.2.5.1.228.1 Offset

Register	Offset
SWREG226	388h

16.2.5.1.228.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.228.3 Fields

Field	Function
31-0 SW_REFER0_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 0

16.2.5.1.229 Base address MSB (bits 63:32) for reference compress chrominance table index 1 (SWREG227)

16.2.5.1.229.1 Offset

Register	Offset
SWREG227	38Ch

16.2.5.1.229.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.229.3 Fields

Field	Function
31-0 SW_REFER1_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 1

16.2.5.1.230 Base address LSB (bits 31:0) for reference compress chrominance table index 1 (SWREG228)

16.2.5.1.230.1 Offset

Register	Offset
SWREG228	390h

16.2.5.1.230.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.230.3 Fields

Field	Function
31-0 SW_REFER1_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 1

16.2.5.1.231 Base address MSB (bits 63:32) for reference compress chrominance table index 2 (SWREG229)

16.2.5.1.231.1 Offset

Register	Offset
SWREG229	394h

16.2.5.1.231.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.231.3 Fields

Field	Function
31-0 SW_REFER2_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 2

16.2.5.1.232 Base address LSB (bits 31:0) for reference compress chrominance table index 2 (SWREG230)

16.2.5.1.232.1 Offset

Register	Offset
SWREG230	398h

16.2.5.1.232.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.232.3 Fields

Field	Function
31-0 SW_REFER2_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 2

16.2.5.1.233 Base address MSB (bits 63:32) for reference compress chrominance table index 3 (SWREG231)

16.2.5.1.233.1 Offset

Register	Offset
SWREG231	39Ch

16.2.5.1.233.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.233.3 Fields

Field	Function
31-0 SW_REFER3_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 3

16.2.5.1.234 Base address LSB (bits 31:0) for reference compress chrominance table index 3 (SWREG232)

16.2.5.1.234.1 Offset

Register	Offset
SWREG232	3A0h

16.2.5.1.234.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.234.3 Fields

Field	Function
31-0 SW_REFER3_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 3

16.2.5.1.235 Base address MSB (bits 63:32) for reference compress chrominance table index 4 (SWREG233)

16.2.5.1.235.1 Offset

Register	Offset
SWREG233	3A4h

16.2.5.1.235.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.235.3 Fields

Field	Function
31-0 SW_REFER4_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 4

16.2.5.1.236 Base address LSB (bits 31:0) for reference compress chrominance table index 4 (SWREG234)

16.2.5.1.236.1 Offset

Register	Offset
SWREG234	3A8h

16.2.5.1.236.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.236.3 Fields

Field	Function
31-0 SW_REFER4_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 4

16.2.5.1.237 Base address MSB (bits 63:32) for reference compress chrominance table index 5 (SWREG235)

16.2.5.1.237.1 Offset

Register	Offset
SWREG235	3ACh

16.2.5.1.237.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.237.3 Fields

Field	Function
31-0 SW_REFER5_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 5

16.2.5.1.238 Base address LSB (bits 31:0) for reference compress chrominance table index 5 (SWREG236)

16.2.5.1.238.1 Offset

Register	Offset
SWREG236	3B0h

16.2.5.1.238.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.238.3 Fields

Field	Function
31-0 SW_REFER5_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 5

16.2.5.1.239 Base address MSB (bits 63:32) for reference compress chrominance table index 6 (SWREG237)

16.2.5.1.239.1 Offset

Register	Offset
SWREG237	3B4h

16.2.5.1.239.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.239.3 Fields

Field	Function
31-0 SW_REFER6_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 6

16.2.5.1.240 Base address LSB (bits 31:0) for reference compress chrominance table index 6 (SWREG238)

16.2.5.1.240.1 Offset

Register	Offset
SWREG238	3B8h

16.2.5.1.240.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.240.3 Fields

Field	Function
31-0 SW_REFER6_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 6

16.2.5.1.241 Base address MSB (bits 63:32) for reference compress chrominance table index 7 (SWREG239)

16.2.5.1.241.1 Offset

Register	Offset
SWREG239	3BCh

16.2.5.1.241.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.241.3 Fields

Field	Function
31-0 SW_REFER7_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 7

16.2.5.1.242 Base address LSB (bits 31:0) for reference compress chrominance table index 7 (SWREG240)

16.2.5.1.242.1 Offset

Register	Offset
SWREG240	3C0h

16.2.5.1.242.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.242.3 Fields

Field	Function
31-0 SW_REFER7_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 7

16.2.5.1.243 Base address MSB (bits 63:32) for reference compress chrominance table index 8 (SWREG241)

16.2.5.1.243.1 Offset

Register	Offset
SWREG241	3C4h

16.2.5.1.243.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.243.3 Fields

Field	Function
31-0 SW_REFER8_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 8

16.2.5.1.244 Base address LSB (bits 31:0) for reference compress chrominance table index 8 (SWREG242)

16.2.5.1.244.1 Offset

Register	Offset
SWREG242	3C8h

16.2.5.1.244.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.244.3 Fields

Field	Function
31-0 SW_REFER8_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 8

16.2.5.1.245 Base address MSB (bits 63:32) for reference compress chrominance table index 9 (SWREG243)

16.2.5.1.245.1 Offset

Register	Offset
SWREG243	3CCh

16.2.5.1.245.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.245.3 Fields

Field	Function
31-0 SW_REFER9_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 9

16.2.5.1.246 Base address LSB (bits 31:0) for reference compress chrominance table index 9 (SWREG244)

16.2.5.1.246.1 Offset

Register	Offset
SWREG244	3D0h

16.2.5.1.246.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.246.3 Fields

Field	Function
31-0 SW_REFER9_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 9

16.2.5.1.247 Base address MSB (bits 63:32) for reference compress chrominance table index 10 (SWREG245)

16.2.5.1.247.1 Offset

Register	Offset
SWREG245	3D4h

16.2.5.1.247.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.247.3 Fields

Field	Function
31-0 SW_REFER10_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 10

16.2.5.1.248 Base address LSB (bits 31:0) for reference compress chrominance table index 10 (SWREG246)

16.2.5.1.248.1 Offset

Register	Offset
SWREG246	3D8h

16.2.5.1.248.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.248.3 Fields

Field	Function
31-0 SW_REFER10_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 10

16.2.5.1.249 Base address MSB (bits 63:32) for reference compress chrominance table index 11 (SWREG247)

16.2.5.1.249.1 Offset

Register	Offset
SWREG247	3DCh

16.2.5.1.249.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.249.3 Fields

Field	Function
31-0 SW_REFER11_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 11

16.2.5.1.250 Base address LSB (bits 31:0) for reference compress chrominance table index 11 (SWREG248)

16.2.5.1.250.1 Offset

Register	Offset
SWREG248	3E0h

16.2.5.1.250.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.250.3 Fields

Field	Function
31-0 SW_REFER11_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 11

16.2.5.1.251 Base address MSB (bits 63:32) for reference compress chrominance table index 12 (SWREG249)

16.2.5.1.251.1 Offset

Register	Offset
SWREG249	3E4h

16.2.5.1.251.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.251.3 Fields

Field	Function
31-0 SW_REFER12_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 12

16.2.5.1.252 Base address LSB (bits 31:0) for reference compress chrominance table index 12 (SWREG250)

16.2.5.1.252.1 Offset

Register	Offset
SWREG250	3E8h

16.2.5.1.252.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.252.3 Fields

Field	Function
31-0 SW_REFER12_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 12

16.2.5.1.253 Base address MSB (bits 63:32) for reference compress chrominance table index 13 (SWREG251)

16.2.5.1.253.1 Offset

Register	Offset
SWREG251	3ECh

16.2.5.1.253.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.253.3 Fields

Field	Function
31-0 SW_REFER13_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 13

16.2.5.1.254 Base address LSB (bits 31:0) for reference compress chrominance table index 13 (SWREG252)

16.2.5.1.254.1 Offset

Register	Offset
SWREG252	3F0h

16.2.5.1.254.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.254.3 Fields

Field	Function
31-0 SW_REFER13_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 13

16.2.5.1.255 Base address MSB (bits 63:32) for reference compress chrominance table index 14 (SWREG253)

16.2.5.1.255.1 Offset

Register	Offset
SWREG253	3F4h

16.2.5.1.255.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.255.3 Fields

Field	Function
31-0 SW_REFER14_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 14

16.2.5.1.256 Base address LSB (bits 31:0) for reference compress chrominance table index 14 (SWREG254)

16.2.5.1.256.1 Offset

Register	Offset
SWREG254	3F8h

16.2.5.1.256.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.256.3 Fields

Field	Function
31-0 SW_REFER14_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 14

16.2.5.1.257 Base address MSB (bits 63:32) for reference compress chrominance table index 15 (SWREG255)

16.2.5.1.257.1 Offset

Register	Offset
SWREG255	3FCh

16.2.5.1.257.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.257.3 Fields

Field	Function
31-0 SW_REFER15_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 15

16.2.5.1.258 Base address LSB (bits 31:0) for reference compress chrominance table index 15 (SWREG256)

16.2.5.1.258.1 Offset

Register	Offset
SWREG256	400h

16.2.5.1.258.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.258.3 Fields

Field	Function
31-0 SW_REFER15_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 15

16.2.5.1.259 Not used (SWREG257)

16.2.5.1.259.1 Offset

Register	Offset
SWREG257	404h

16.2.5.1.259.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.259.3 Fields

Field	Function
31-0 —	Reserved.

16.2.5.1.260 input stream buffer length (SWREG258)

16.2.5.1.260.1 Offset

Register	Offset
SWREG258	408h

16.2.5.1.260.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_STRM_BUFFER_LEN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_STRM_BUFFER_LEN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.260.3 Fields

Field	Function
31-0 SW_STRM_BUFFER_LEN	input stream buffer length

16.2.5.1.261 input stream buffer start offset (SWREG259)

16.2.5.1.261.1 Offset

Register	Offset
SWREG259	40Ch

16.2.5.1.261.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_STRM_START_OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_STRM_START_OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.2.5.1.261.3 Fields

Field	Function
31-0 SW_STRM_START_OFFSET	input stream buffer start offset

16.3 VPU VC8000E Encoder

16.3.1 Overview

The Video Processing Unit (VPU) VC8000E encoder provides encoding support for AVC/H.264 and HEVC/H.265.

NOTE

Encoder Line buffer is limited to 1920 pixel width. So Encoder cannot encode image with width higher than 1920.

16.3.1.1 Block Diagram

The block diagram of the VPU encoder is shown below.

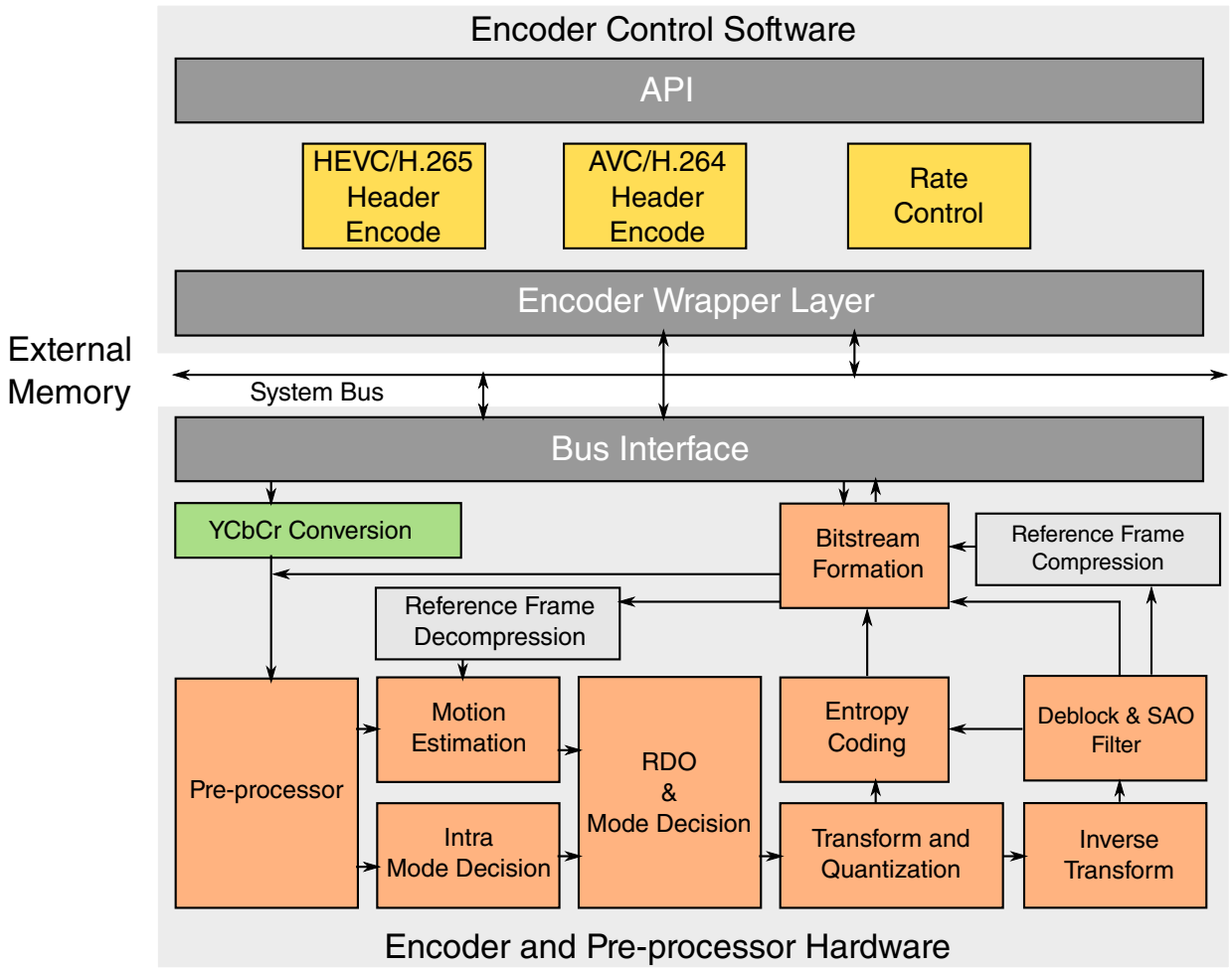


Figure 16-11. VPU Encoder Block Diagram

16.3.1.1.1 Block Descriptions

The main functional blocks of the VPU encoder are described in the table below.

Table 16-7. VPU Encoder Functional Blocks

Functional block	Description
Video Pipeline Controller	Inserts high level commands into the video pipeline.
Pre-processor	Performs conversions (e.g., RGB to YUV, YUV 4:2:2 to 4:2:0, between 10-bit and 8-bit) Cropping Rotation
Motion estimation	Motion vector prediction to determine which motion vectors will be transformed from one frame to another.
Mode selection	Determines the prediction mode (inter or intra) to use when encoding.
Intra Prediction	Creates prediction blocks based on samples from the current frame that were previously encoded and reconstructed.

Table continues on the next page...

Table 16-7. VPU Encoder Functional Blocks (continued)

Functional block	Description
Inter Prediction	Creates prediction blocks based on prior encoded frames, where samples are shifted in the reference frames.
Transform & Quantization	Manipulations performed on the prediction macroblock prior to encoding.
Entropy Encoder	Provides lossless coding which uses fewer bits to represent frequently occurring patterns.
Bitstream Generation	Creates the encoded bitstream of encoded data output.
Inverse Transform	Manipulations performed on the encoded macroblock.
Deblocking	Filter that reduces the visible blocking artifacts at block boundaries of decoded compressed video.
Sample Adaptive Offset	Encoding and filtering applied after deblocking to reduce banding effects and improve video quality.
Reference Frame Compression	Compresses the reference frame for use in future frame prediction.

16.3.1.2 Features

The VPU VC8000E encoder includes the following features:

- HEVC (H.265)
  - Main Profile,
  - Main10 profile, High Tier
  - Main Still Profile
- AVC (H.264)
  - Baseline Profile
  - Main Profile
  - High Profile
  - High 10 Profile
- Pre-processing
  - Frame cropping
  - Frame rotation
  - Color space conversion

16.3.2 Functional description

16.3.2.1 HEVC (H.265)

Additional details for the supported HEVC (H.265) features are provided in the tables below.

Table 16-8. HEVC (H.265) Feature Support

HEVC feature	Encoder support
Profiles	Main Profile, High Tier Main10 profile, High Tier Main Still Profile
Slices	I and P slices
Entropy encoding	CABAC
Error resilience	Slices
Maximum MV range	Horizontal (P slice) in pixels: +/-139 Vertical (P) in pixels: • Config1: +/-37 • Config2: +/-45 (planned) • Config3: +/-61 (planned) (=Search Window Size -3 pixels)
MV accuracy	1/4 pixel
Supported block sizes	CTU size: 64x64 CU size: 32x32/16x16/8x8 Inter PU: 2Nx2N, 2NxN, Nx2N, AMP Intra PU:2Nx2N, NxN TU size: 32x32/16x16/8x8/4x4
Intra-prediction modes	33 directional modes DC and planar modes
SAO	Supported
Maximum number of reference frames	2
Temporal scalable video coding	Up to 5 layers including the base layer
IPCM encoding	IPCM rectangle mode
ROI/ROI_map	Absolute QP and qpoffset mode (-32 ~ 31) User controllable CU coded as IPCM CU or skip CU

16.3.2.2 AVC (H.264)

Additional details for the supported AVC (H.264) features are provided in the table below.

Table 16-9. AVC (H.264) Feature Support

AVC feature	Encoder support
AVC Standard	Baseline Profile Main Profile High Profile High 10 Profile
Profiles	Main Profile High Profile High10 Profile Baseline Profile
Maximum luma pixel sample rate	1920x1080@60fps (BP, MP, HP)
Slices	I and P slices
Frame Types	Progressive
Entropy encoding	CABAC CAVLC
Error resilience	Slices
Maximum MV range	Horizontal (P slice) in pixels: +/-139 Vertical (P) in pixels: • Config1: +/-13 (planned) • Config2: +/-21 • Config3: +/-29 (planned) • Config4: +/-45 (planned) • Config5: +/-61 (planned) (= Search Window Size -3 pixels)
MV accuracy	1/4 pixel
Supported block sizes	Macroblock and sub-macroblock partitions: • Intra PU:16x16/8x8/4x4 • Inter PU: 16x16/8x16/16x8 • TU: 4x4 and 8x8 transforms
Intra-prediction modes	16x16: 4 modes 8x8: 9 modes 4x4: 9 modes
Maximum number of reference frames	2
Encoding picture type	Only progressive frame
IPCM encoding	Supported
Temporal scalable video coding	Up to 5 layers including the base layer
IPCM	IPCM rectangle mode
ROI/ROI_map	Absolute QP and qpoffset mode (-32 ~ 31) User controllable CU coded as IPCM CU or skip CU

16.3.2.3 Pre-processor

Additional details for the supported preprocessing features are provided in the table below.

Table 16-10. Pre-Processing Features

Pre-processor feature	Encoder support
8-bit YCbCr 4:2:0 color space	YCbCr 4:2:0 planar(YV12) YCbCr 4:2:0 semi-planar(NV12) YCrCb 4:2:0 semi-planar(NV21)
8-bit YCbCr 4:2:2 to YCbCr 4:2:0 color space conversion	YCbYCr 4:2:2 interleaved (YUY2) CbYCrY 4:2:2 interleaved (UYVY)
10-bit YCbCr color	I010 P010 PAK10 (VeriSilicon format) Y0L2 (TIE10) YCbCr 4:2:0 semi-planar 10
RGB to YCbCr 4:2:0 color space conversion	BT.601, BT.709, BT.2020(HDR10) or user defined coefficient conversion for RGB: • XRGB4444 and XBGR4444 • XRGB1555 and XBGR1555 • RGB565 and BGR565 • XRGB8888 and XBGR8888 • X2R10G10B10 and X2B10G10R10
Cropping	Supported, from 16384x16384 to any supported encoding size
Rotation	90, 270 degrees

16.3.2.4 Clocks

The following table describes the clock sources for the VPU VC8000E encoder. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 16-11. VPU Encoder Clocks

Clock name	Description
core_clk	VPU encoder core clock

16.3.2.5 Reset

The VPU encoder has one software reset (vc8000e_sft_rstn).

16.3.3 External Signals

There are no external signals pinned out for this module.

16.3.4 Initialization and Application Information

The VPU encoder is operated via the VPU Application Programming Interface (API). Details are provided in separate API documentation.

16.3.5 Memory map and register definition

This section includes the VC8000E encoder memory map and descriptions of the registers.

16.3.5.1 VC8000E_H264 register descriptions

16.3.5.1.1 VC8000E_H264 memory map

VC8000E_H264 base address: 3832_0000h

Offset	Register	Width (In bits)	Access	Reset value
4h	Interrupt register encoder (swreg1)	32	RW	0000_0800h
8h	Data configuration register0 (swreg2)	32	RW	0000_0000h
Ch	Data configuration register1 (swreg3)	32	RW	FF00_0000h
10h	control register 0 (swreg4)	32	RW	2190_0014h
14h	control register 1 (swreg5)	32	RW	Table 16-11
18h	control register 2 (swreg6)	32	RW	0000_0000h
1Ch	control register 3 (swreg7)	32	RW	6800_1A00h
20h	stream output buffer0 address (swreg8)	32	RW	0000_0000h
24h	stream output buffer0 limit size (swreg9)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
28h	sizeTblBase (swreg10)	32	RW	0000_0000h
2Ch	encoded Picture order count (swreg11)	32	RW	0000_0000h
30h	input lum base address (swreg12)	32	RW	0000_0000h
34h	input cb base address (swreg13)	32	RW	0000_0000h
38h	input cr base address (swreg14)	32	RW	0000_0000h
3Ch	recon image luma base address (swreg15)	32	RW	0000_0000h
40h	recon image chroma base address (swreg16)	32	RW	0000_0000h
48h	reference picture reconstructed list0 luma0 (swreg18)	32	RW	0000_0000h
4Ch	reference picture reconstructed list0 chroma0 (swreg19)	32	RW	0000_0000h
58h	Cyclic Intra (swreg22)	32	RW	0000_0000h
5Ch	intra Area (swreg23)	32	RW	0000_0000h
60h	ROI1 Area (swreg24)	32	RW	0000_0000h
64h	ROI2 Area (swreg25)	32	RW	0000_0000h
68h	intra penalty for picture0. For H2V1 only. (swreg26_H2V1)	32	RW	0000_0000h
68h	intra size factors. For H2V2 or later version. (swreg26_H2V2)	32	RW	0000_0000h
6Ch	intra penalty for picture1. For H2V1 only. (swreg27_H2V1)	32	RW	0000_0000h
6Ch	intra mode factors . For H2V2 or later version. (swreg27_H2V2)	32	RW	0000_0000h
70h	intra penalty for picture2 (swreg28)	32	RW	0000_0000h
70h	inter me SATD lambda config 0. For H2V2 or later version. (swreg28_H2V2)	32	RW	0000_0000h
70h	inter me SATD lambda config 0. For H2V5 or later version. (swreg28_H2V5)	32	RW	0000_0000h
74h	intra penalty for ROI10. For H2V1 only. (swreg29_H2V1)	32	RW	0000_0000h
74h	inter me SATD lambda config 1.For H2V2 or later version. (swreg29_H2V2)	32	RW	0000_0000h
74h	inter me SATD lambda config 1.For H2V5 or later version. (swreg29_H2V5)	32	RW	0000_0000h
78h	intra penalty for ROI11. For H2V1 only. (swreg30_H2V1)	32	RW	0000_0000h
78h	inter me SATD lambda config 2. For H2V2 or later version. (swreg30_H2V2)	32	RW	0000_0000h
78h	inter me SATD lambda config 2. For H2V5 or later version. (swreg30_H2V5)	32	RW	0000_0000h
7Ch	intra penalty for ROI12. For H2V1 only. (swreg31_H2V1)	32	RW	0000_0000h
7Ch	inter me SATD lambda config 3. For H2V2 or later version. (swreg31_H2V2)	32	RW	0000_0000h
7Ch	inter me SATD lambda config 3. For H2V5 or later version. (swreg31_H2V5)	32	RW	0000_0000h
80h	intra penalty for ROI20. For H2V1 only. (swreg32_H2V1)	32	RW	0000_0000h
80h	inter me SSE lambda config 0. For H2V2 or later version. (swreg32_H2V2)	32	RW	0000_0000h
80h	inter me SATD lambda config 4. For H2V5 or later version. (swreg32_H2V5)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
84h	intra penalty for ROI21. For H2V1 only. (swreg33_H2V1)	32	RW	0000_0000h
84h	inter me SSE lambda config 1. For H2V2 or later version. (swreg33_H2V2)	32	RW	0000_0000h
84h	inter me SATD lambda config 5. For H2V5 or later version. (swreg33_H2V5)	32	RW	0000_0000h
88h	intra penalty for ROI22. For H2V1 only. (swreg34_H2V1)	32	RW	0000_0000h
88h	inter me SSE lambda config 2. For H2V2 or later version. (swreg34_H2V2)	32	RW	0000_0000h
88h	inter me SATD lambda config 6. For H2V5 or later version. (swreg34_H2V5)	32	RW	0000_0000h
8Ch	inter prediction parameters1 (swreg35)	32	RW	0000_0000h
90h	inter prediction parameters2 (swreg36)	32	RW	0000_0000h
94h	SAO lambda parameter (swreg37)	32	RW	0000_0000h
98h	Pre-processor configuration (swreg38)	32	RW	0000_0000h
9Ch	Pre-processor color conversion parameters0 (swreg39)	32	RW	0000_0000h
A0h	Pre-processor color conversion parameters1 (swreg40)	32	RW	0000_0000h
A4h	Pre-processor color conversion parameters2 (swreg41)	32	RW	0000_0000h
B8h	compressed coefficients base address for SAN module. (swreg46)	32	RW	0000_0000h
F0h	Base address for recon luma compress table LSB. (swreg60)	32	RW	0000_0000h
F8h	Base address for recon Chroma compress table LSB (swreg62)	32	RW	0000_0000h
100h	Base address for list 0 ref 0 luma compress table LSB. (swreg64)	32	RW	0000_0000h
108h	Base address for list 0 ref 0 Chroma compress table LSB. (swreg66)	32	RW	0000_0000h
120h	Base address for recon luma 4n base LSB. (swreg72)	32	RW	0000_0000h
128h	reference picture reconstructed list0 4n 0 (swreg74)	32	RW	0000_0000h
138h	inter prediction parameters3. For H2V1 only. (swreg78_H2V1)	32	RW	0000_0002h
138h	inter me SSE lambda config 3. For H2V2 or later version. (swreg78_H2V2)	32	RW	0000_0000h
138h	inter me SATD lambda config 7. For H2V5 or later version. (swreg78_H2V5)	32	RW	0000_0000h
13Ch	inter prediction parameters4. For H2V1 only. (swreg79_H2V1)	32	RW	0000_0000h
13Ch	inter me SSE lambda config 4. For H2V2 or later version. (swreg79_H2V2)	32	RW	0000_0000h
13Ch	inter me SSE lambda config 0. For H2V5 or later version. (swreg79_H2V5)	32	RW	0000_0000h
140h	HW synthesis config register, read-only (swreg80)	32	RO	F8BF_D000h
144h	hardware configuation 0 (swreg81)	32	RW	2000_0000h
148h	record hardware performance (swreg82)	32	RO	0000_0000h
14Ch	reference picture reconstructed list1 luma0 (swreg83)	32	RW	0000_0000h
150h	reference picture reconstructed list1 chroma0 (swreg84)	32	RW	0000_0000h
16Ch	reference pictures list1 config (swreg91)	32	RW	0000_0000h
170h	reference picture reconstructed list1 4n 0 (swreg92)	32	RW	0000_0000h
180h	Base address for list 1 ref 0 luma compress table LSB. (swreg96)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
188h	Base address for list 1 ref 0 Chroma compress table LSB. (swreg98)	32	RW	0000_0000h
1A8h	Min picture size (swreg106)	32	RW	0000_0000h
1ACh	Max picture size (swreg107)	32	RW	0000_0000h
1B4h	Qp delta map (swreg109)	32	RW	0000_0000h
1BCh	adaptive GOP configuration1 (swreg111)	32	RO	0000_0000h
1C0h	adaptive GOP configuration2 (swreg112)	32	RO	0000_0000h
1C4h	adaptive GOP configuration3 (swreg113)	32	RW	0000_0000h
1C8h	ctb rate control bit memory address of current frame (swreg114)	32	RW	0000_0000h
1D0h	ctb rate control bit memory address of previous frame (swreg116)	32	RW	0000_0000h
1DCh	min/max lcu bits number of last picture (swreg119)	32	RW	0000_0000h
1E0h	total bits number of all lcus of last picture not including slice header bits (swreg120)	32	RW	0000_0000h
1E8h	inter me SSE lambda config 5 (swreg122)	32	RW	0000_0000h
1E8h	inter me SSE lambda config 1. For H2V5 or later version. (swreg122_H2V5)	32	RW	0000_0000h
1ECh	inter me SSE lambda config 6 (swreg123)	32	RW	0000_0000h
1ECh	inter me SSE lambda config 2. For H2V5 or later version. (swreg123_H2V5)	32	RW	0000_0000h
1F0h	inter me SSE lambda config 7 (swreg124)	32	RW	0000_0000h
1F0h	inter me SSE lambda config 3. For H2V5 or later version. (swreg124_H2V5)	32	RW	0000_0000h
1F4h	intra SATD lambda config 0 (swreg125)	32	RW	0000_0000h
1F8h	intra SATD lambda config 1 (swreg126)	32	RW	0000_0000h
1FCh	intra SATD lambda config 2 (swreg127)	32	RW	0000_0000h
200h	intra SATD lambda config 3 (swreg128)	32	RW	0000_0000h
204h	intra SATD lambda config 4 (swreg129)	32	RW	0000_0000h
208h	intra SATD lambda config 5 (swreg130)	32	RW	0000_0000h
20Ch	intra SATD lambda config 6 (swreg131)	32	RW	0000_0000h
210h	intra SATD lambda config 7 (swreg132)	32	RW	0000_0000h
214h	SSE devide 256 (swreg133)	32	RW	0000_0000h
228h	inter me SSE lambda config 4. For H2V5 or later version. (swreg138_H2V5)	32	RW	0000_0000h
22Ch	inter me SSE lambda config 5. For H2V5 or later version. (swreg139_H2V5)	32	RW	0000_0000h
230h	inter me SSE lambda config 6. For H2V5 or later version. (swreg140_H2V5)	32	RW	0000_0000h
234h	inter me SSE lambda config 7. For H2V5 or later version. (swreg141_H2V5)	32	RW	0000_0000h
238h	inter me SSE lambda config 8. For H2V5 or later version. (swreg142_H2V5)	32	RW	0000_0000h
23Ch	inter me SSE lambda config 9. For H2V5 or later version. (swreg143_H2V5)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
240h	inter me SSE lambda config 10. For H2V5 or later version. (swreg144_H2V5)	32	RW	0000_0000h
244h	inter me SSE lambda config 11. For H2V5 or later version. (swreg145_H2V5)	32	RW	0000_0000h
248h	inter me SSE lambda config 12. For H2V5 or later version. (swreg146_H2V5)	32	RW	0000_0000h
24Ch	inter me SSE lambda config 13. For H2V5 or later version. (swreg147_H2V5)	32	RW	0000_0000h
250h	inter me SSE lambda config 14. For H2V5 or later version. (swreg148_H2V5)	32	RW	0000_0000h
254h	inter me SSE lambda config 15. For H2V5 or later version. (swreg149_H2V5)	32	RW	0000_0000h
258h	inter me SATD lambda config 8 (swreg150)	32	RW	0000_0000h
25Ch	inter me SATD lambda config 9 (swreg151)	32	RW	0000_0000h
260h	inter me SATD lambda config 10 (swreg152)	32	RW	0000_0000h
264h	inter me SATD lambda config 11 (swreg153)	32	RW	0000_0000h
268h	inter me SATD lambda config 12 (swreg154)	32	RW	0000_0000h
26Ch	inter me SATD lambda config 13 (swreg155)	32	RW	0000_0000h
270h	inter me SATD lambda config 14 (swreg156)	32	RW	0000_0000h
274h	inter me SATD lambda config 15 (swreg157)	32	RW	0000_0000h
278h	inter me SSE lambda config 16 (swreg158)	32	RW	0000_0000h
27Ch	inter me SSE lambda config 17 (swreg159)	32	RW	0000_0000h
280h	inter me SSE lambda config 18 (swreg160)	32	RW	0000_0000h
284h	inter me SSE lambda config 19 (swreg161)	32	RW	0000_0000h
288h	inter me SSE lambda config 20 (swreg162)	32	RW	0000_0000h
28Ch	inter me SSE lambda config 21 (swreg163)	32	RW	0000_0000h
290h	inter me SSE lambda config 22 (swreg164)	32	RW	0000_0000h
294h	inter me SSE lambda config 23 (swreg165)	32	RW	0000_0000h
298h	inter me SSE lambda config 24 (swreg166)	32	RW	0000_0000h
29Ch	inter me SSE lambda config 25 (swreg167)	32	RW	0000_0000h
2A0h	inter me SSE lambda config 26 (swreg168)	32	RW	0000_0000h
2A4h	inter me SSE lambda config 27 (swreg169)	32	RW	0000_0000h
2B0h	inter me SSE lambda config 30 (swreg172)	32	RW	0000_0000h
2B4h	inter me SSE lambda config 31 (swreg173)	32	RW	0000_0660h
2B8h	intra SATD lambda config 8 (swreg174)	32	RW	0000_0000h
2BCh	intra SATD lambda config 9 (swreg175)	32	RW	0000_0000h
2C0h	intra SATD lambda config 10 (swreg176)	32	RW	0000_0000h
2C4h	intra SATD lambda config 11 (swreg177)	32	RW	0000_0000h
2C8h	intra SATD lambda config 12 (swreg178)	32	RW	0000_0000h
2CCh	intra SATD lambda config 13 (swreg179)	32	RW	0000_0000h
2D0h	intra SATD lambda config 14 (swreg180)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
2D4h	intra SATD lambda config 15 (swreg181)	32	RW	0000_0000h
2D8h	qp fractional part (swreg182)	32	RW	0000_0000h
2DCh	qp sum (swreg183)	32	RO	0000_0000h
2E0h	qp num (swreg184)	32	RO	0000_0000h
2E4h	picture complexity. Timeout cycles MSB. (swreg185)	32	RW	0000_0000h
2F8h	Long-term reference pictures config (swreg190)	32	RW	0000_0000h
2FCh	Temporal scalable config (swreg191)	32	RW	0000_0000h
300h	encoded Picture frame number (for H.264) (swreg192)	32	RW	0000_0000h
304h	reference pictures list0 config (for H.264) (swreg193)	32	RW	0000_0001h
308h	reference pictures list1 config (for H.264) (swreg194)	32	RW	0000_0000h
30Ch	register extension for ctu_size=16 (swreg195)	32	RW	0000_0000h
310h	Low Latency Controls (swreg196)	32	RW	0000_0000h
314h	Delta POC extension (swreg197)	32	RW	0000_0000h
318h	Long Term Reference Control (swreg198)	32	RW	0000_0000h
31Ch	Hash Code Control (swreg199)	32	RW	0000_0000h
320h	Hash Code Value (swreg200)	32	RW	0000_0000h
324h	Background SKIP Control 0 (swreg201)	32	RW	0505_0505h
32Ch	Background SKIP Control 2 (swreg203)	32	RW	0500_0101h
340h	Background SKIP Control 7 (swreg208)	32	RW	7800_0000h
344h	IPCM Control 0 (swreg209)	32	RW	0000_0000h
348h	IPCM Control 1 (swreg210)	32	RW	0000_0000h
34Ch	IPCM Control 2 (swreg211)	32	RW	0000_0000h
350h	IPCM Control 3 (swreg212)	32	RW	0000_0000h
354h	IPCM Control 4 (swreg213)	32	RW	0000_0000h
358h	HW synthesis config register 2, read-only (swreg214)	32	RO	C440_0000h
35Ch	AXI Information 0 (swreg215)	32	RO	0000_0000h
360h	AXI Information 1 (swreg216)	32	RO	0000_0000h
364h	AXI Information 2 (swreg217)	32	RO	0000_0000h
368h	AXI Information 3 (swreg218)	32	RO	0000_0000h
36Ch	AXI Information 4 (swreg219)	32	RO	0000_0000h
370h	AXI Information 5 (swreg220)	32	RO	0000_0000h
374h	AXI Information 6 (swreg221)	32	RO	0000_0000h
378h	AXI Information 7 (swreg222)	32	RO	0000_0000h
37Ch	AXI Information 8 (swreg223)	32	RO	0000_0000h
380h	control register 4 (swreg224)	32	RW	0002_0080h
384h	Tile Control (swreg225)	32	RW	0000_0000h
388h	HW synthesis config register 3, read-only (swreg226)	32	RO	0000_0000h
3ACh	RPS encoding control 0 (swreg235)	32	RW	0000_0000h
3B0h	RPS encoding control 1 (swreg236)	32	RW	0000_2000h
3B4h	Stride Control (swreg237)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
3B8h	Dummy Read (swreg238)	32	RW	0000_0000h
3BCh	Base Address LSB of CTB MADs of current frame. (swreg239)	32	RW	0000_0000h
3C4h	Base Address LSB of CTB MADs of previous frame. (swreg241)	32	RW	0000_0000h
3CCh	CTB RC Control 0 (swreg243)	32	RW	0000_0000h
3D0h	CTB RC Control 1 (swreg244)	32	RW	0000_0000h
3D4h	CTB RC Control 2 (swreg245)	32	RW	0000_0000h
3D8h	CTB RC Control 3 (swreg246)	32	RW	0000_1010h
3DCh	CTB RC Control 4 (swreg247)	32	RW	0000_0000h
3E0h	CTB RC Control 5 (swreg248)	32	RW	0000_0000h
3E4h	register extension for 8K width (swreg249)	32	RW	0000_0000h
3E8h	Global MV Control 0 (swreg250)	32	RW	0000_0000h
3ECh	Global MV Control 1 (swreg251)	32	RW	0000_0000h
3F0h	ROI3 Area (swreg252)	32	RW	0000_0000h
3F4h	ROI3&4 Area (swreg253)	32	RW	0000_0000h
3F8h	ROI4&5 Area (swreg254)	32	RW	0000_0000h
3FCh	ROI5 Area (swreg255)	32	RW	0000_0000h
400h	ROI6 Area (swreg256)	32	RW	0000_0000h
404h	ROI6&7 Area (swreg257)	32	RW	0000_0000h
408h	ROI7&8 Area (swreg258)	32	RW	0000_0000h
40Ch	ROI8 Area (swreg259)	32	RW	0000_0000h
410h	ROI qp (swreg260)	32	RW	0000_0000h
414h	Stride Control (swreg261)	32	RW	0000_0404h
424h	Multicore sync ctrl (swreg265)	32	RW	0000_0000h
428h	Multicore sync address L0 LSB (swreg266)	32	RW	0000_0000h
42Ch	Multicore sync address L0 MSB (swreg267)	32	RW	0000_0000h
430h	Multicore sync address L1 LSB (swreg268)	32	RW	0000_0000h
434h	Multicore sync address L1 MSB (swreg269)	32	RW	0000_0000h
438h	Multicore sync address recon LSB (swreg270)	32	RW	0000_0000h
43Ch	Multicore sync address recon MSB (swreg271)	32	RW	0000_0000h
440h	Programmable AXI urgent sideband signals (swreg272)	32	RW	FFFF_FFFFh
444h	roimap cu ctrl index address LSB (swreg273)	32	RW	0000_0000h
448h	roimap cu ctrl index address MSB (swreg274)	32	RW	0000_0000h
44Ch	roimap cu ctrl address LSB (swreg275)	32	RW	0000_0000h
450h	roimap cu ctrl address MSB (swreg276)	32	RW	0000_0000h
454h	poc type/bits setting (swreg277)	32	RW	0000_0000h
458h	stream output buffer1 address (swreg278)	32	RW	0000_0000h
460h	stream output buffer1 limit size (swreg280)	32	RW	0000_0000h
464h	poc type/bits setting (swreg281)	32	RW	0000_0000h
47Ch	HW synthesis config register 4, read-only (swreg287)	32	RO	0000_0000h
484h	Pre-processor color conversion parameters1 (swreg289)	32	RW	0000_0000h

16.3.5.1.2 Interrupt register encoder (swreg1)

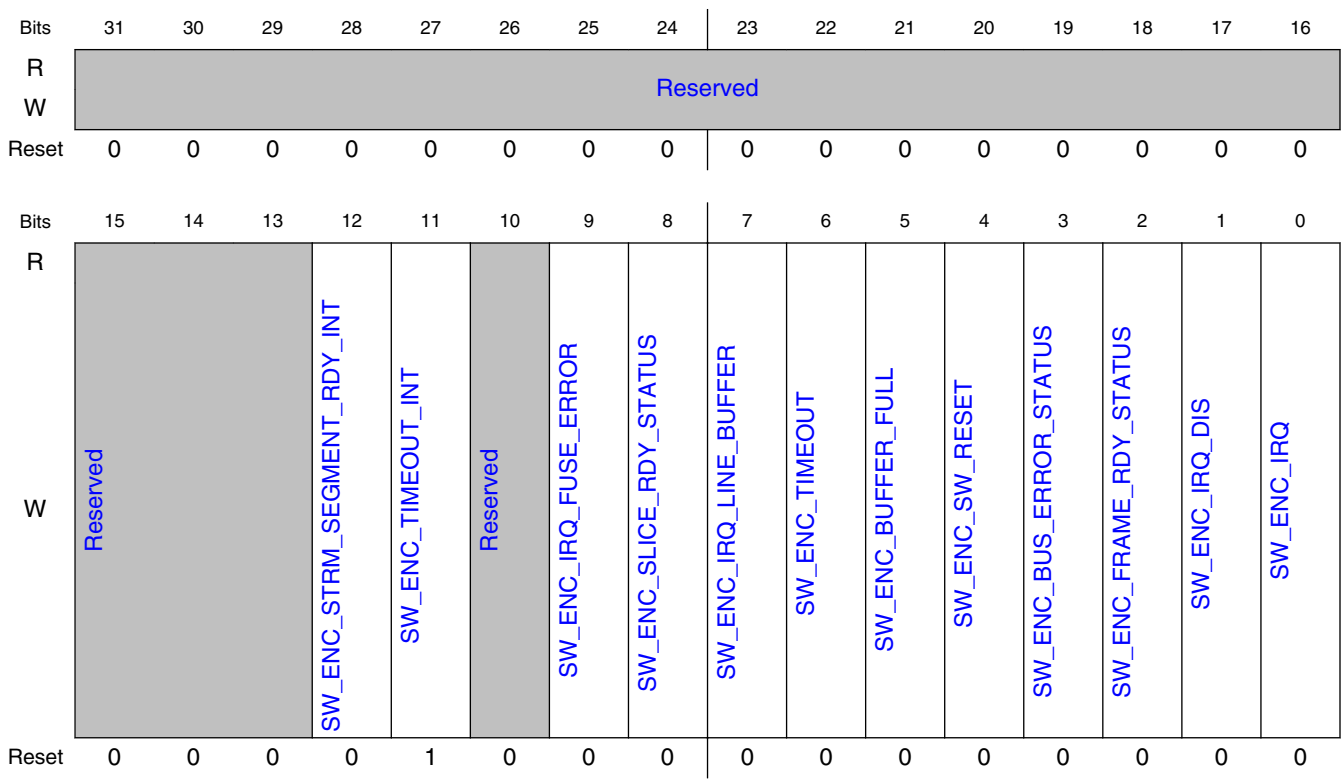
16.3.5.1.2.1 Offset

Register	Offset
swreg1	4h

16.3.5.1.2.2 Function

Interrupt register encoder

16.3.5.1.2.3 Diagram



16.3.5.1.2.4 Fields

Field	Function
31-13 —	Reserved

Table continues on the next page...

Field	Function
12 SW_ENC_STRM_SEGMENT_RDY_INT	Output stream segment IRQ. When high stream segment requests an interrupt. SW will reset this after interrupt is handled.
11 SW_ENC_TIMEOUT_INT	enable timeout interrupt when 1
10 —	Reserved
9 SW_ENC_IRQ_FUSE_ERROR	Interrupt Interrupt status bit encoder. When this bit is high encoder has a FUSE error. SW try to setup feature but HW miss the feature set
8 SW_ENC_SLICE_RDY_STATUS	Interrupt status bit encoder. When this bit is high encoder has encoded a slice.
7 SW_ENC_IRQ_LINE_BUFFER	Interrupt status bit encoder linebuffer empty. When setting. Encoder has finished encoding input buffers.
6 SW_ENC_TIMEOUT	Interrupt status bit encoder timeout. When high the encoder has been idling for too long. Possible only if timeout interrupt is enabled
5 SW_ENC_BUFFER_FULL	IRQ buffer full status bit. bufferFullInterrupt
4 SW_ENC_SW_RESET	IRQ SW reset status bit.
3 SW_ENC_BUS_ERROR_STATUS	Interrupt status bit bus. Error response from bus.
2 SW_ENC_FRAME_RDY_STATUS	Interrupt status bit encoder. When this bit is high encoder has encoded a picture.
1 SW_ENC_IRQ_DIS	Encoder IRQ disable. When high there are no interrupts concerning encoder from HW. Polling must be used to see the interrupt statuses.
0 SW_ENC_IRQ	Encoder IRQ. When high encoder requests an interrupt. SW will reset this after interrupt is handled.

16.3.5.1.3 Data configuration register0 (swreg2)

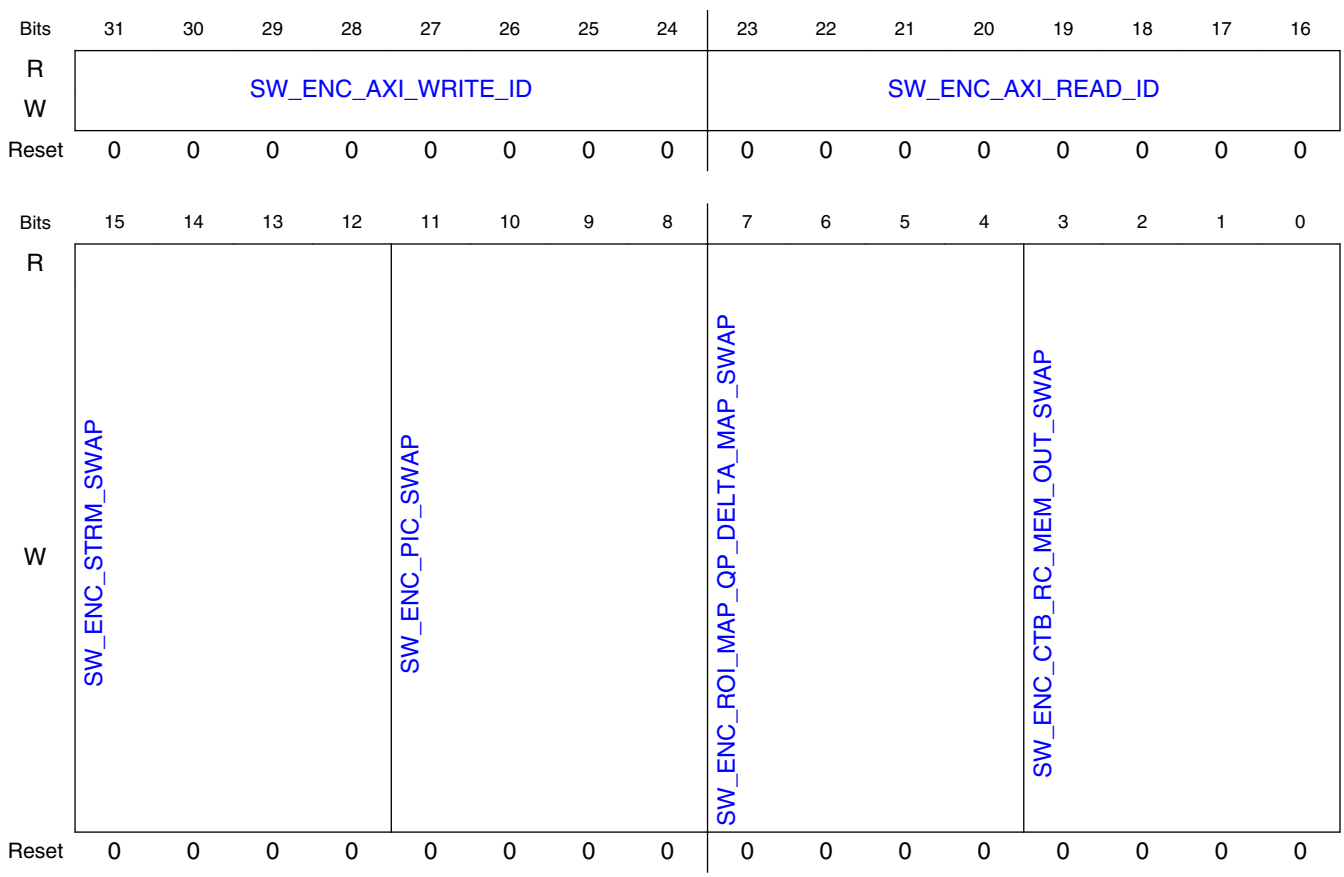
16.3.5.1.3.1 Offset

Register	Offset
swreg2	8h

16.3.5.1.3.2 Function

Data configuration register0

16.3.5.1.3.3 Diagram



16.3.5.1.3.4 Fields

Field	Function
31-24	AXI Write ID

Table continues on the next page...

Field	Function
SW_ENC_AXI_WRITE_ID	
23-16 SW_ENC_AXI_READ_ID	AXI Read ID
15-12 SW_ENC_STRM_SWAP	byte swap config for output stream data.4 Bit byte order vector to control byte locations inside HW internal 128 bit data vector. For 64 and 32 bit external bus widths the data is first gathered to 128 bit width and then bytes swapped accordingly
11-8 SW_ENC_PIC_SWAP	Byte swap configuration for picture data (encoder input)
7-4 SW_ENC_ROI_MAP_QP_DELTA_MAP_SWAP	Byte swap configuration for qp delta of ROI map
3-0 SW_ENC_CTB_RC_MEM_OUT_SWAP	Byte swap configuration for ctb rate control memory out

16.3.5.1.4 Data configuration register1 (swreg3)

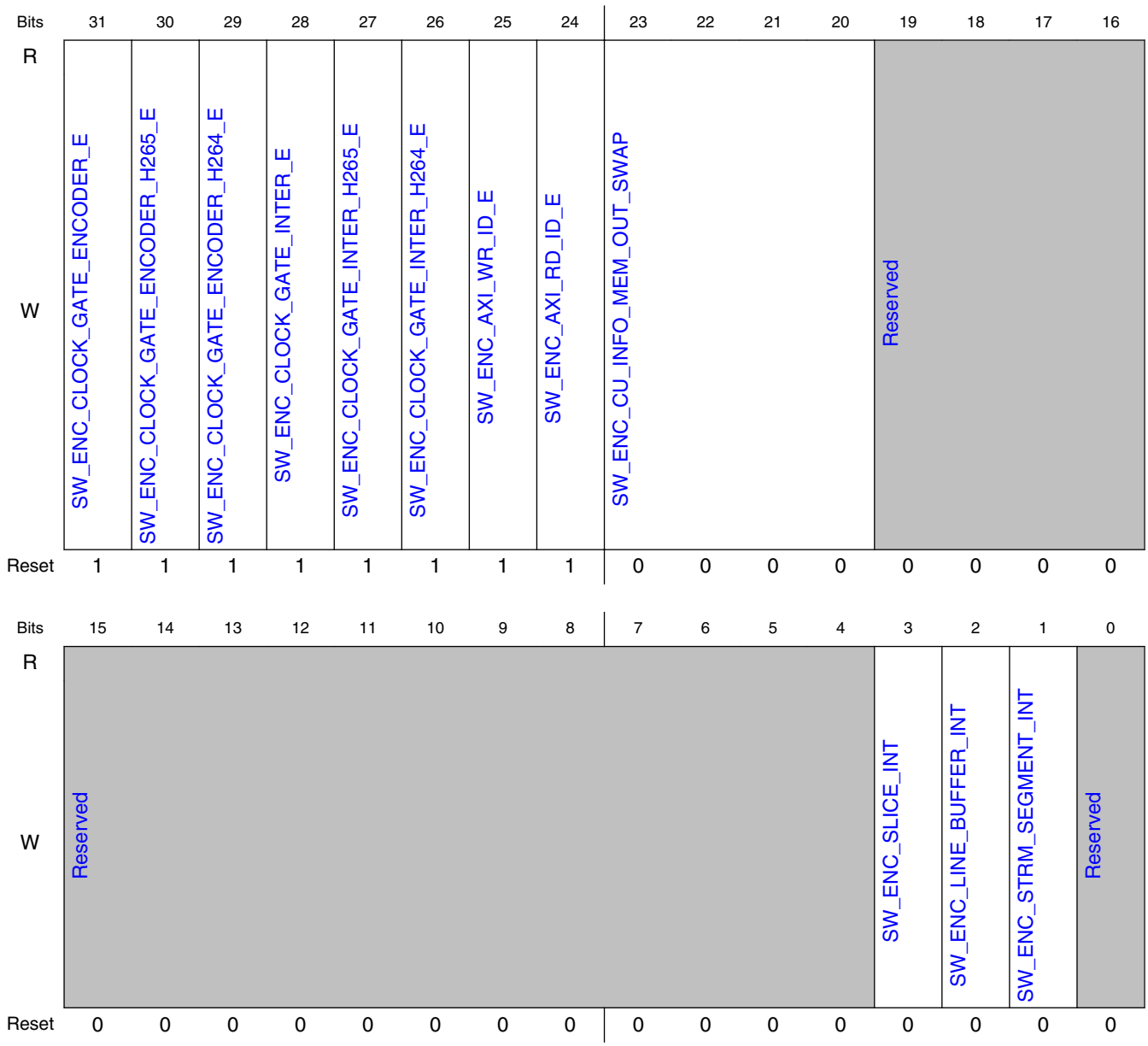
16.3.5.1.4.1 Offset

Register	Offset
swreg3	Ch

16.3.5.1.4.2 Function

Data configuration register1

16.3.5.1.4.3 Diagram



16.3.5.1.4.4 Fields

Field	Function
31 SW_ENC_CLOCK_GATE_ENCODER_E	ASIC encoder clock gating control. 0b - clock always on. 1b - hardware clock gating control
30	ASIC encoder clock gating control for h265. 0b - clock always on. 1b - hardware clock gating control

Table continues on the next page...

Field	Function
SW_ENC_CLOCK_GATE_ENCODER_H265_E	
29 SW_ENC_CLOCK_GATE_ENCODER_H264_E	ASIC encoder clock gating control for h264. 0b - clock always on. 1b - hardware clock gating control
28 SW_ENC_CLOCK_GATE_INTER_E	ASIC inter clock gating control. 0b - clock always on. 1b - hardware clock gating control
27 SW_ENC_CLOCK_GATE_INTER_H265_E	ASIC inter clock gating control for h265. 0b - clock always on. 1b - hardware clock gating control
26 SW_ENC_CLOCK_GATE_INTER_H264_E	ASIC inter clock gating control for h264. 0b - clock always on. 1b - hardware clock gating control
25 SW_ENC_AXI_WR_ID_E	axi write enable. 0b - disable. 1b - enable.
24 SW_ENC_AXI_RD_ID_E	axi read enable. 0b - disable. 1b - enable.
23-20 SW_ENC_CU_INFO_MEM_OUT_SWAP	Byte swap configuration for cu infomation memory out
19-4 —	Reserved
3 SW_ENC_SLICE_INT	enable slice interrupt when 1
2 SW_ENC_LINE_BUFFER_INT	enable line buffer interrupt when 1
1 SW_ENC_STRM_SEGMENT_INT	enable stream segment interrupt when 1
0 —	Reserved

16.3.5.1.5 control register 0 (swreg4)

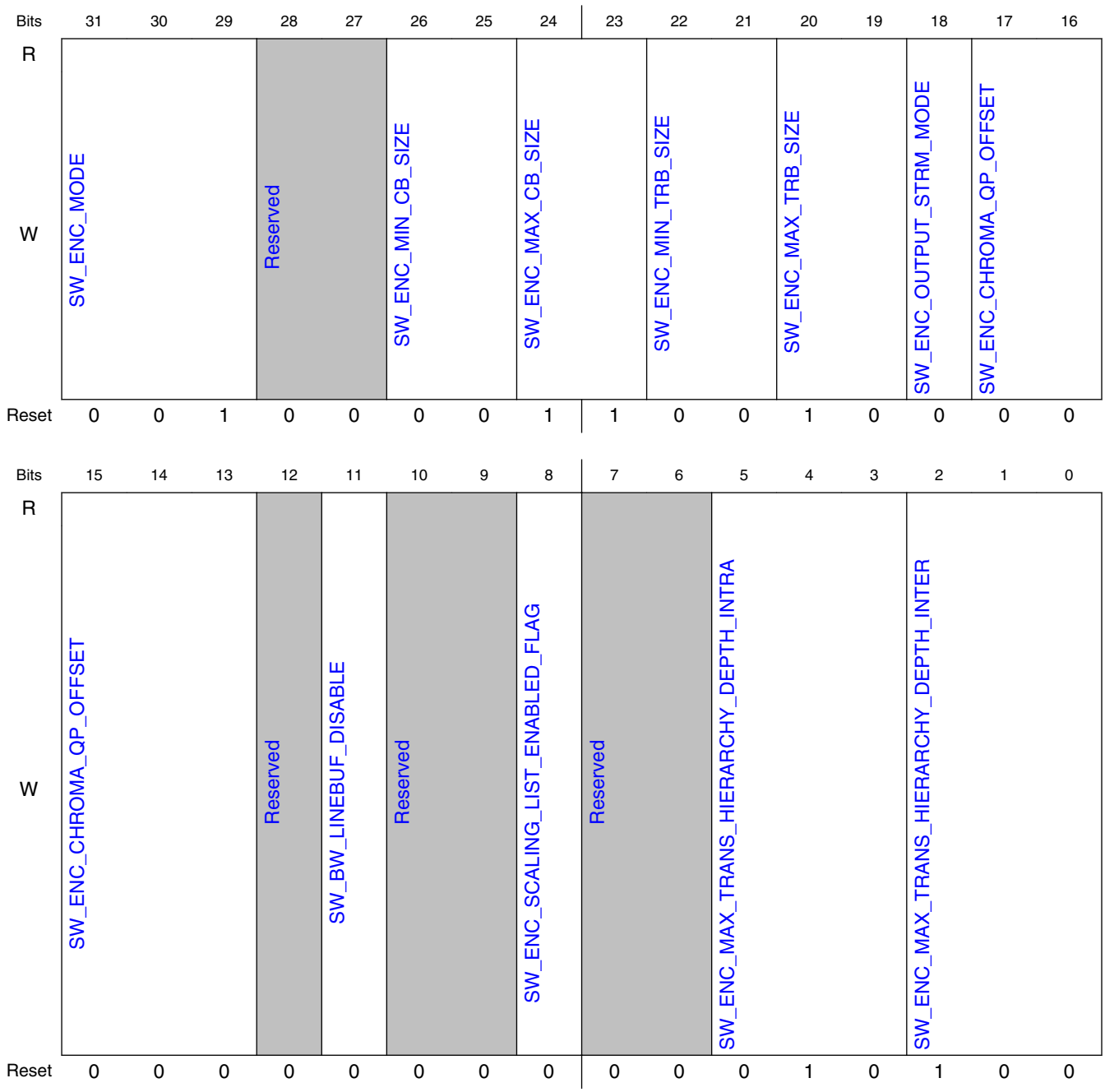
16.3.5.1.5.1 Offset

Register	Offset
swreg4	10h

16.3.5.1.5.2 Function

control register 0

16.3.5.1.5.3 Diagram



16.3.5.1.5.4 Fields

Field	Function
31-29 SW_ENC_MODE	Encoding mode. streamType. 001b - hevc. 010b - h264.

Table continues on the next page...

Field	Function
28-27 —	Reserved
26-25 SW_ENC_MIN_CB_SIZE	min cb size (we only support 8x8): 00b - 8x8 01b - 16x16 10b - 32x32 11b - 64x64
24-23 SW_ENC_MAX_CB_SIZE	max cb size (we only support 64x64): 00b - 8x8 01b - 16x16 10b - 32x32 11b - 64x64
22-21 SW_ENC_MIN_TRB_SIZE	min tr block size (we only support 4x4) 00b - 4x4 01b - 8x8 10b - 16x16 11b - 32x32
20-19 SW_ENC_MAX_TRB_SIZE	max tr block size (we only support 16x16) 00b - 4x4 01b - 8x8 10b - 16x16 11b - 32x32
18 SW_ENC_OUTPUT_STRM_MODE	output stream mode: 0b - byte stream 1b - Nal stream
17-13 SW_ENC_CHROMA_QP_OFFSET	chroma qp offset[-12~12]
12 —	Reserved
11 SW_BW_LINEBUF_DISABLE	BW line buffer disable
10-9 —	Reserved
8 SW_ENC_SCALING_LIST_ENABLED_FLAG	scaling_list_enabled_flag
7-6 —	Reserved
5-3 SW_ENC_MAX_TRANS_HIERARCHY_DEPTH_INTRA	max transform hierarchy depth of intra

Table continues on the next page...

Field	Function
2-0 SW_ENC_MAX_TRANS_HIERARCHY_DEPTH_INTER	max transform hierarchy depth of inter

16.3.5.1.6 control register 1 (swreg5)

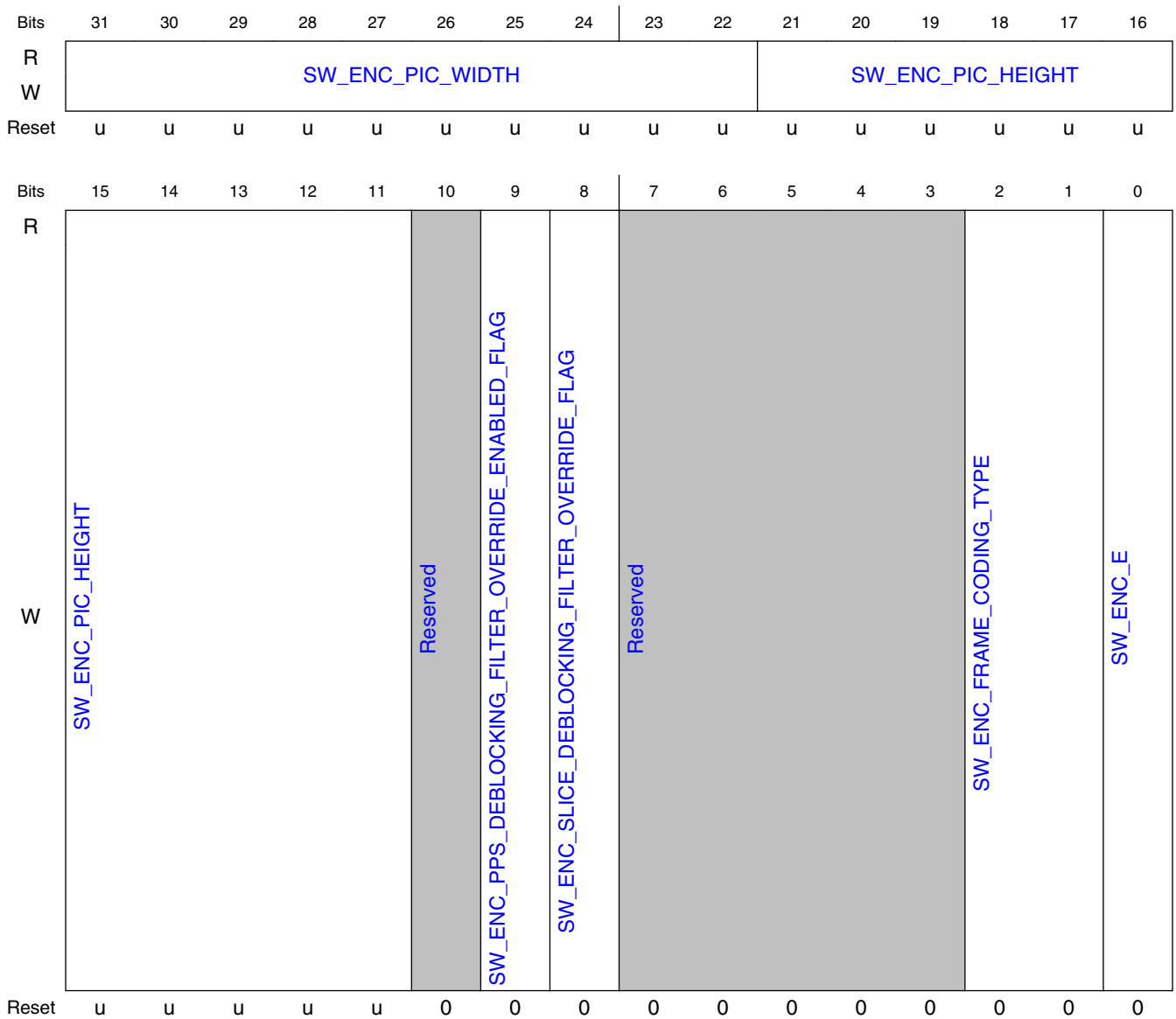
16.3.5.1.6.1 Offset

Register	Offset
swreg5	14h

16.3.5.1.6.2 Function

control register 1

16.3.5.1.6.3 Diagram



16.3.5.1.6.4 Fields

Field	Function
31-22 SW_ENC_PIC_WIDTH	Encoded width. lumWidthLsb(unit 8 pixels). bit[9:0]
21-11 SW_ENC_PIC_HEIGHT	Encoded height. lumHeight(unit 8 pixels):Max width x height is 4096 x 4096
10	Reserved

Table continues on the next page...

Field	Function
—	
9 SW_ENC_PPS_DEBLOCKING_FILTER_OVERRIDE_ENABLED_FLAG	deblocking filter override enable flag. 0b - disable 1b - enable
8 SW_ENC_SLICE_DEBLOCKING_FILTER_OVERRIDE_FLAG	slice deblocking filter override flag. 0b - no 1b - yes
7-3 —	Reserved
2-1 SW_ENC_FRAME_CODING_TYPE	Encoded picture type. frameType.:1-I:0-P:2:B
0 SW_ENC_E	encoder enable. Setting this bit high will start the encoding operation. HW will reset this when picture is processed or bus error or timeout interrupt is given.

16.3.5.1.7 control register 2 (swreg6)

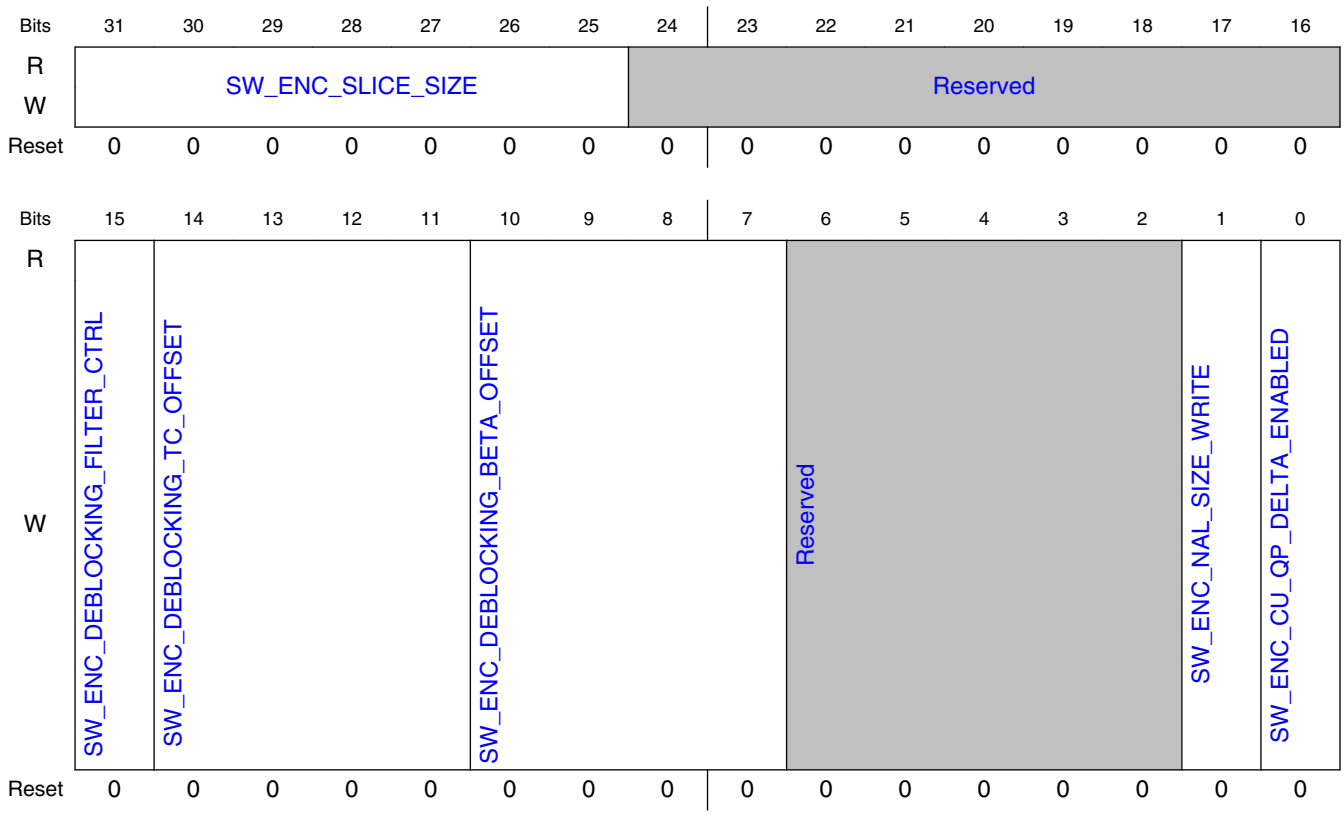
16.3.5.1.7.1 Offset

Register	Offset
swreg6	18h

16.3.5.1.7.2 Function

control register 2

16.3.5.1.7.3 Diagram



16.3.5.1.7.4 Fields

Field	Function
31-25 SW_ENC_SLICE_SIZE	slice size in ctu row for HEVC and MB row for H264
24-16 —	Reserved
15 SW_ENC_DEBLOCKING_FILTER_CTRL	deblocking filter control.De-block filtering disable. 0b - filtering is enabled for current picture. 1b - filtering is disabled for current picture.
14-11 SW_ENC_DEBLOCKING_TC_OFFSET	deblocking tc offset:-6~6
10-7 SW_ENC_DEBLOCKING_BETA_OFFSET	deblocking beta offset:-6~6

Table continues on the next page...

Field	Function
6-2 —	Reserved
1 SW_ENC_NAL_SIZE_WRITE	Enable writing size of each NAL unit to BaseControl
0 SW_ENC_CU_QP_DELTA_ENABLED	cu qp delta encoding is enabled:used for ROI

16.3.5.1.8 control register 3 (swreg7)

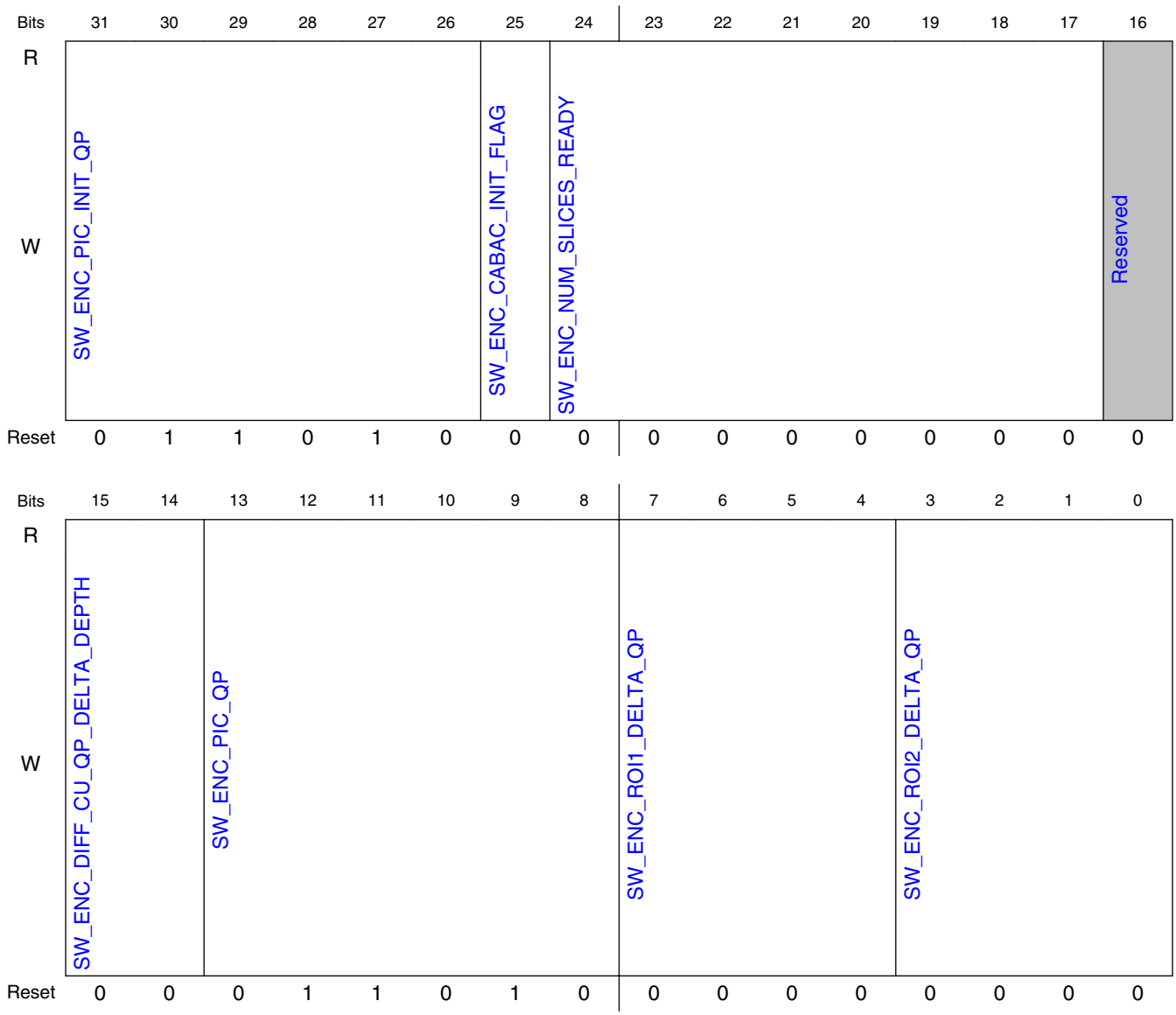
16.3.5.1.8.1 Offset

Register	Offset
swreg7	1Ch

16.3.5.1.8.2 Function

control register 3

16.3.5.1.8.3 Diagram



16.3.5.1.8.4 Fields

Field	Function
31-26 SW_ENC_PIC_INIT_QP	picture header qp
25 SW_ENC_CABAC_INIT_FLAG	cabac init flag
24-17	HEVC amount of completed slices.

Table continues on the next page...

Field	Function
SW_ENC_NUM_SLICES_READY	
16 —	Reserved
15-14 SW_ENC_DIFF_CU_QP_DELTA_DEPTH	difference of cu qp delta depth
13-8 SW_ENC_PIC_QP	qp of current picture
7-4 SW_ENC_ROI1_DELTA_QP	ROI1 delta qp
3-0 SW_ENC_ROI2_DELTA_QP	ROI2 delta qp

16.3.5.1.9 stream output buffer0 address (swreg8)

16.3.5.1.9.1 Offset

Register	Offset
swreg8	20h

16.3.5.1.9.2 Function

stream output buffer0 address

16.3.5.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_OUTPUT_STRM_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_OUTPUT_STRM_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.9.4 Fields

Field	Function
31-0 SW_ENC_OUTPUT_STRM_BASE	stream output buffer0 base address

16.3.5.1.10 stream output buffer0 limit size (swreg9)

16.3.5.1.10.1 Offset

Register	Offset
swreg9	24h

16.3.5.1.10.2 Function

stream output buffer0 limit size

16.3.5.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_OUTPUT_STRM_BUFFER_LIMIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_OUTPUT_STRM_BUFFER_LIMIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.10.4 Fields

Field	Function
31-0 SW_ENC_OUTPUT_STRM_BUFFER_LIMIT	Stream buffer0 limit / Output stream size (bytes). If buffer0 limit is reached and sw_enc_HWStreamBufChain=0 buffer_full_IRQ will be generated. Output stream size will be updated here when a slice or frame is finished

16.3.5.1.11 sizeTblBase (swreg10)

16.3.5.1.11.1 Offset

Register	Offset
swreg10	28h

16.3.5.1.11.2 Function

sizeTblBase

16.3.5.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_SIZE_TBL_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_SIZE_TBL_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.11.4 Fields

Field	Function
31-0 SW_ENC_SIZE_TBL_BASE	sizeTblBase

16.3.5.1.12 encoded Picture order count (swreg11)

16.3.5.1.12.1 Offset

Register	Offset
swreg11	2Ch

16.3.5.1.12.2 Function

encoded Picture order count

16.3.5.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_POC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_POC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.12.4 Fields

Field	Function
31-0 SW_ENC_POC	encoded Picture order count

16.3.5.1.13 input lum base address (swreg12)

16.3.5.1.13.1 Offset

Register	Offset
swreg12	30h

16.3.5.1.13.2 Function

input lum base address

16.3.5.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_INPUT_Y_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_INPUT_Y_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.13.4 Fields

Field	Function
31-0 SW_ENC_INPUT_Y_BASE	input image lum base address

16.3.5.1.14 input cb base address (swreg13)

16.3.5.1.14.1 Offset

Register	Offset
swreg13	34h

16.3.5.1.14.2 Function

input cb base address

16.3.5.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_INPUT_CB_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_INPUT_CB_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.14.4 Fields

Field	Function
31-0 SW_ENC_INPUT_CB_BASE	input image cb base address

16.3.5.1.15 input cr base address (swreg14)

16.3.5.1.15.1 Offset

Register	Offset
swreg14	38h

16.3.5.1.15.2 Function

input cr base address

16.3.5.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_INPUT_CR_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_INPUT_CR_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.15.4 Fields

Field	Function
31-0 SW_ENC_INPUT_CR_BASE	input image cr base address

16.3.5.1.16 recon image luma base address (swreg15)

16.3.5.1.16.1 Offset

Register	Offset
swreg15	3Ch

16.3.5.1.16.2 Function

recon image luma base address

16.3.5.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RECON_Y_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RECON_Y_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.16.4 Fields

Field	Function
31-0 SW_ENC_RECON_Y_BASE	recon image lum base address

16.3.5.1.17 recon image chroma base address (swreg16)

16.3.5.1.17.1 Offset

Register	Offset
swreg16	40h

16.3.5.1.17.2 Function

recon image chroma base address

16.3.5.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RECON_CHROMA_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RECON_CHROMA_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.17.4 Fields

Field	Function
31-0 SW_ENC_RECON_CHROMA_BASE	recon image chroma base address

16.3.5.1.18 reference picture reconstructed list0 luma0 (swreg18)

16.3.5.1.18.1 Offset

Register	Offset
swreg18	48h

16.3.5.1.18.2 Function

reference picture reconstructed list0 luma0

16.3.5.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REFPIC_RECON_L0_Y0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_REFPIC_RECON_L0_Y0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.18.4 Fields

Field	Function
31-0 SW_ENC_REFPIC_RECON_L0_Y0	reference picture reconstructed list0 luma0

16.3.5.1.19 reference picture reconstructed list0 chroma0 (swreg19)

16.3.5.1.19.1 Offset

Register	Offset
swreg19	4Ch

16.3.5.1.19.2 Function

reference picture reconstructed list0 chroma0

16.3.5.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REFPIC_RECON_L0_CHROMA0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_REFPIC_RECON_L0_CHROMA0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.19.4 Fields

Field	Function
31-0 SW_ENC_REFPIC_RECON_L0_CHROMA0	reference picture reconstructed list0 chroma0

16.3.5.1.20 Cyclic Intra (swreg22)

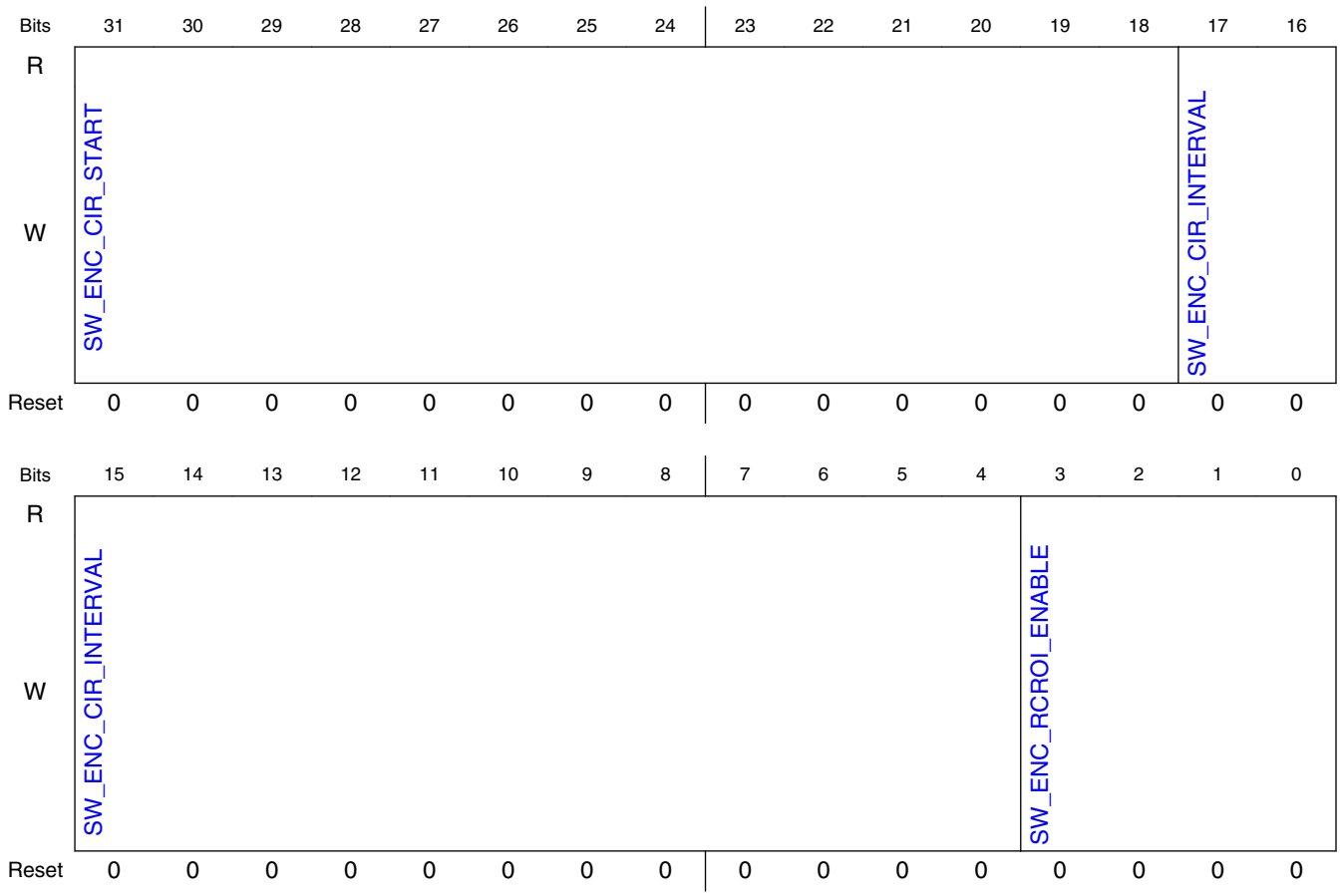
16.3.5.1.20.1 Offset

Register	Offset
swreg22	58h

16.3.5.1.20.2 Function

Cyclic Intra

16.3.5.1.20.3 Diagram



16.3.5.1.20.4 Fields

Field	Function
31-18 SW_ENC_CIR_START	cir start
17-4 SW_ENC_CIR_INTERVAL	Cyclic Intra Refresh
3-0 SW_ENC_RCROI_ENABLE	bit3:RC enable; bit2:Quatily adjustment enable; bit1:roi map enable; bit0:roi area enable.

16.3.5.1.21 intra Area (swreg23)

16.3.5.1.21.1 Offset

Register	Offset
swreg23	5Ch

16.3.5.1.21.2 Function

intra Area

16.3.5.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_INTRA_AREA_LEFT								SW_ENC_INTRA_AREA_RIGHT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_INTRA_AREA_TOP								SW_ENC_INTRA_AREA_BOTTOM							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.21.4 Fields

Field	Function
31-24 SW_ENC_INTRA_AREA_LEFT	intra Area Left
23-16 SW_ENC_INTRA_AREA_RIGHT	intra Area Right
15-8 SW_ENC_INTRA_AREA_TOP	intra Area Top
7-0 SW_ENC_INTRA_AREA_BOTTOM	intra Area Bottom

16.3.5.1.22 ROI1 Area (swreg24)

16.3.5.1.22.1 Offset

Register	Offset
swreg24	60h

16.3.5.1.22.2 Function

ROI1 Area

16.3.5.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROI1_LEFT								SW_ENC_ROI1_RIGHT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROI1_TOP								SW_ENC_ROI1_BOTTOM							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.22.4 Fields

Field	Function
31-24 SW_ENC_ROI1_LEFT	ROI1 Area Left
23-16 SW_ENC_ROI1_RIGHT	ROI1 Area Right
15-8 SW_ENC_ROI1_TOP	ROI1 Area Top
7-0 SW_ENC_ROI1_BOTTOM	ROI1 Area Bottom

16.3.5.1.23 ROI2 Area (swreg25)

16.3.5.1.23.1 Offset

Register	Offset
swreg25	64h

16.3.5.1.23.2 Function

ROI2 Area

16.3.5.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROI2_LEFT								SW_ENC_ROI2_RIGHT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROI2_TOP								SW_ENC_ROI2_BOTTOM							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.23.4 Fields

Field	Function
31-24 SW_ENC_ROI2_LEFT	ROI2 Area Left
23-16 SW_ENC_ROI2_RIGHT	ROI2 Area Right
15-8 SW_ENC_ROI2_TOP	ROI2 Area Top
7-0 SW_ENC_ROI2_BOTTOM	ROI2 Area Bottom

16.3.5.1.24 intra penalty for picture0. For H2V1 only. (swreg26_H2V1)

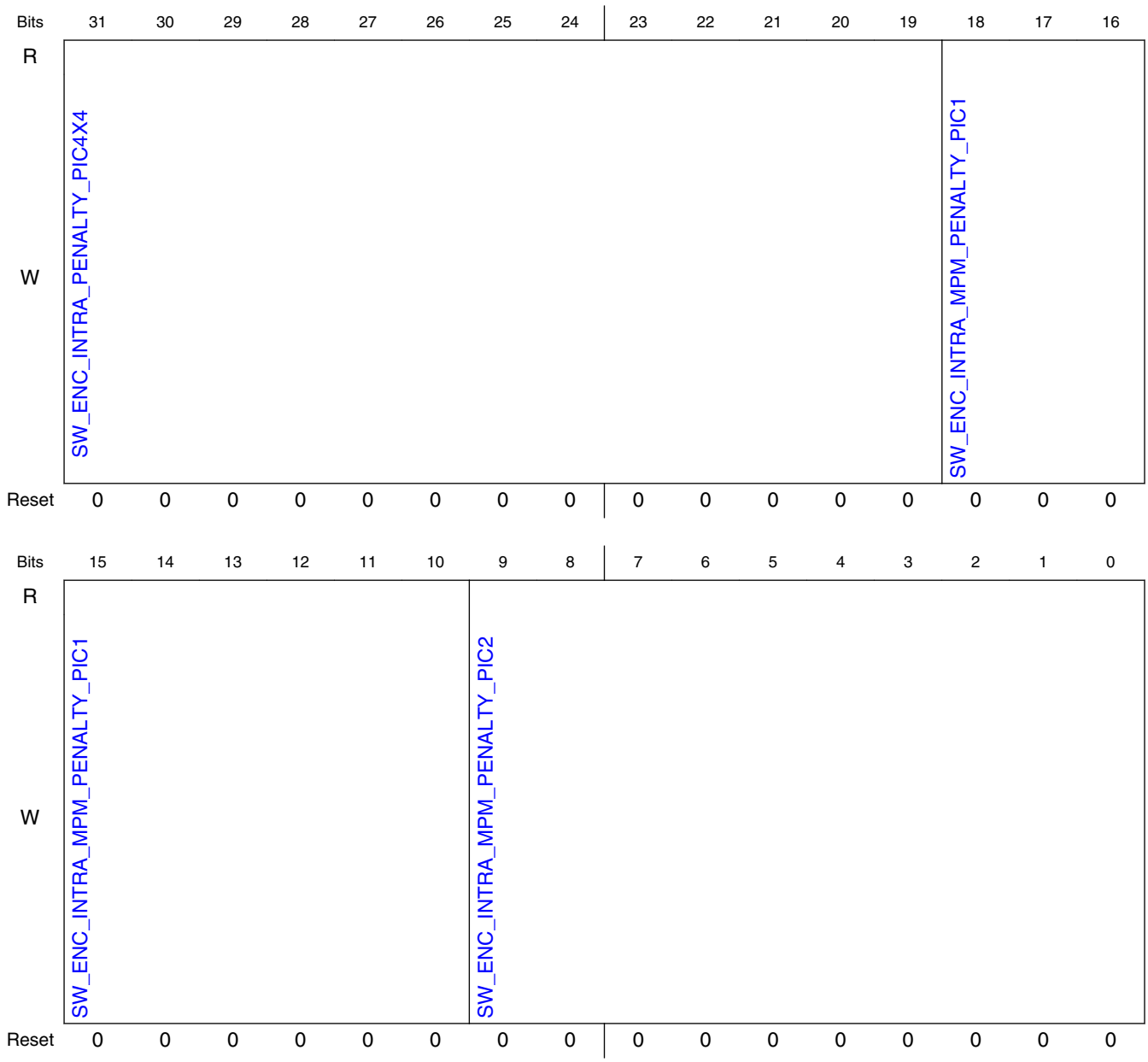
16.3.5.1.24.1 Offset

Register	Offset
swreg26_H2V1	68h

16.3.5.1.24.2 Function

intra penalty for picture0. For H2V1 only.

16.3.5.1.24.3 Diagram



16.3.5.1.24.4 Fields

Field	Function
31-19 SW_ENC_INTRA_PENALTY_PIC4X4	intra penalty for pic4x4
18-10	intra MPM penalty for pic1

Table continues on the next page...

Field	Function
SW_ENC_INTRA_MPM_PENALTY_PIC1	
9-0 SW_ENC_INTRA_MPM_PENALTY_PIC2	intra MPM penalty for pic2

16.3.5.1.25 intra size factors. For H2V2 or later version. (swreg26_H2V2)

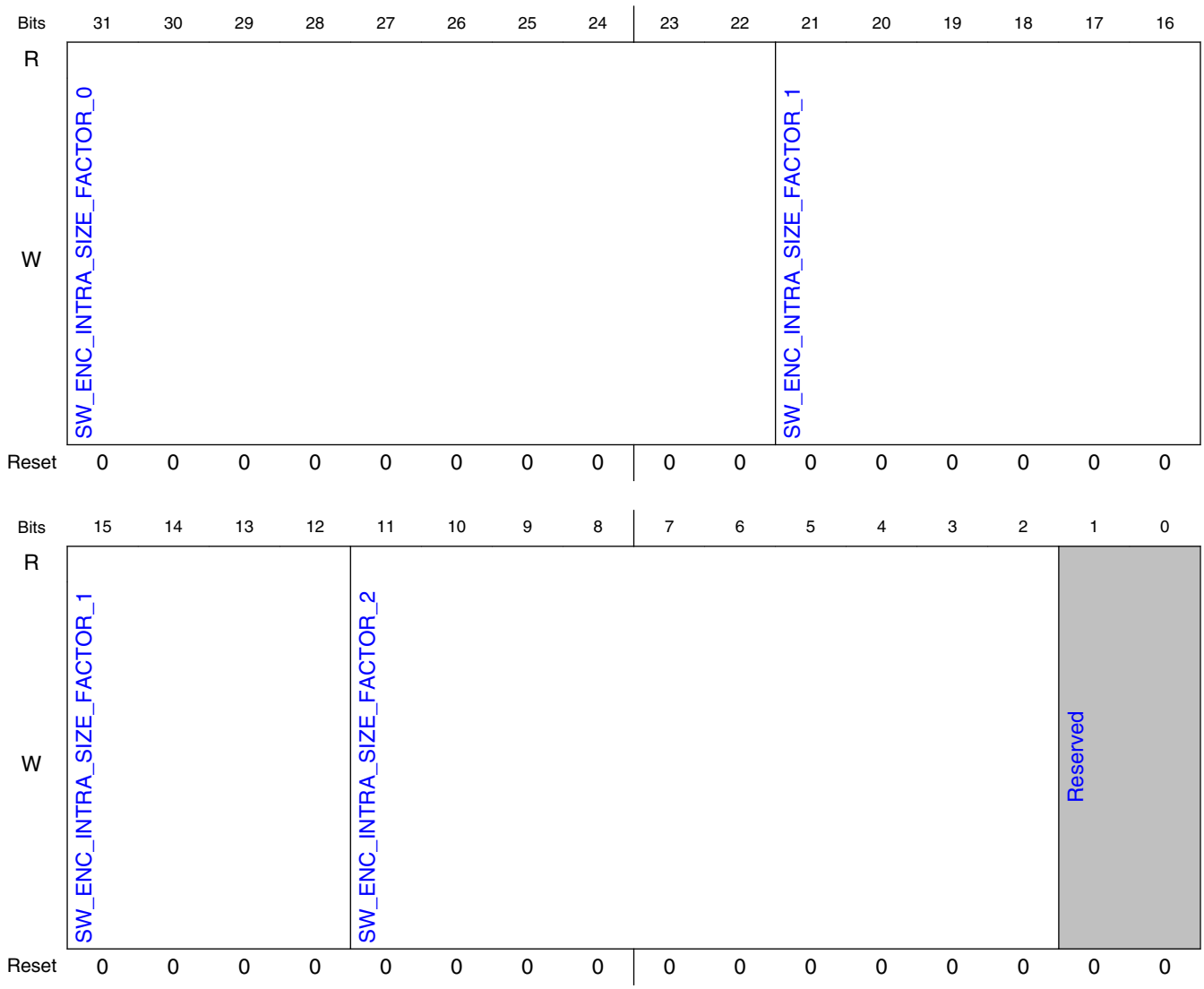
16.3.5.1.25.1 Offset

Register	Offset
swreg26_H2V2	68h

16.3.5.1.25.2 Function

intra size factors. For H2V2 or later version.

16.3.5.1.25.3 Diagram



16.3.5.1.25.4 Fields

Field	Function
31-22 SW_ENC_INTRA_SIZE_FACTOR_0	intra size factor 0
21-12 SW_ENC_INTRA_SIZE_FACTOR_1	intra size factor 1
11-2	intra size factor 2

Table continues on the next page...

Field	Function
SW_ENC_INTRA_SIZE_FACTOR_2	
1-0 —	Reserved

16.3.5.1.26 intra penalty for picture1. For H2V1 only. (swreg27_H2V1)

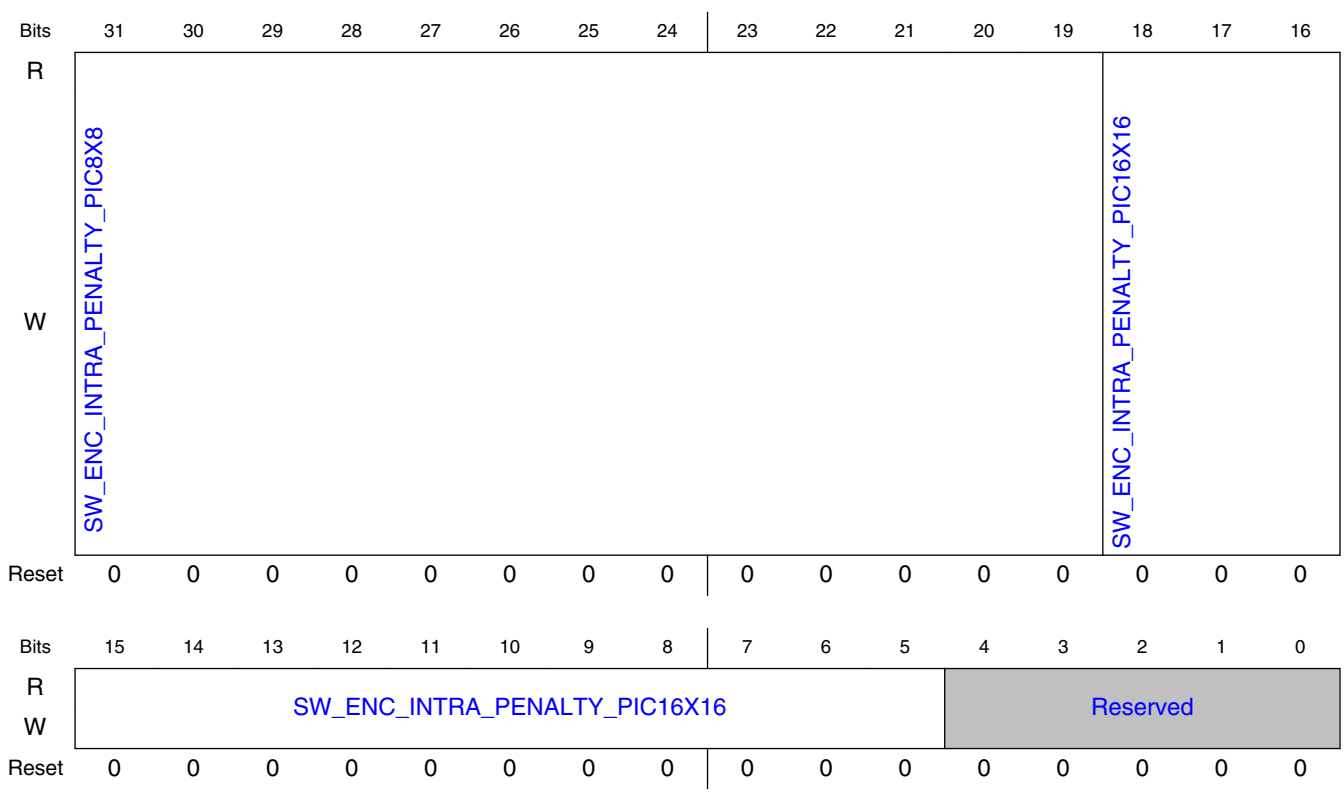
16.3.5.1.26.1 Offset

Register	Offset
swreg27_H2V1	6Ch

16.3.5.1.26.2 Function

intra penalty for picture1. For H2V1 only.

16.3.5.1.26.3 Diagram



16.3.5.1.26.4 Fields

Field	Function
31-19 SW_ENC_INTRA_PENALTY_PIC8X8	intra penalty for pic8x8
18-5 SW_ENC_INTRA_PENALTY_PIC16X16	intra penalty for pic16x16
4-0 —	Reserved

16.3.5.1.27 intra mode factors . For H2V2 or later version. (swreg27_H2V2)

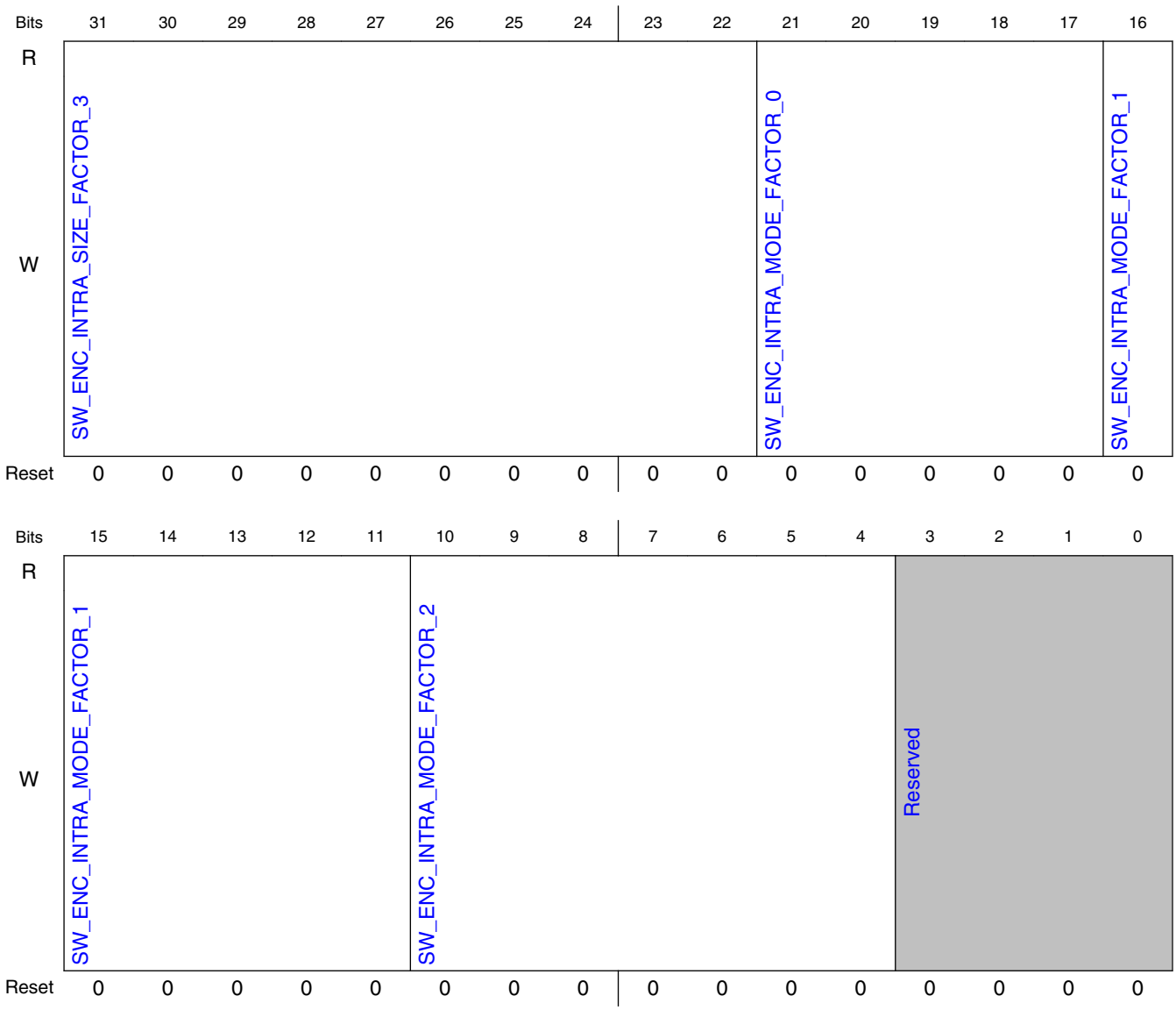
16.3.5.1.27.1 Offset

Register	Offset
swreg27_H2V2	6Ch

16.3.5.1.27.2 Function

intra mode factors . For H2V2 or later version.

16.3.5.1.27.3 Diagram



16.3.5.1.27.4 Fields

Field	Function
31-22 SW_ENC_INTRA_SIZE_FACTOR_3	intra size factor 3
21-17 SW_ENC_INTRA_MODE_FACTOR_0	intra mode factor 0

Table continues on the next page...

Field	Function
16-11 SW_ENC_INTRA_MODE_FACTOR_1	intra mode factor 1
10-4 SW_ENC_INTRA_MODE_FACTOR_2	intra mode factor 2
3-0 —	Reserved

16.3.5.1.28 intra penalty for picture2 (swreg28)

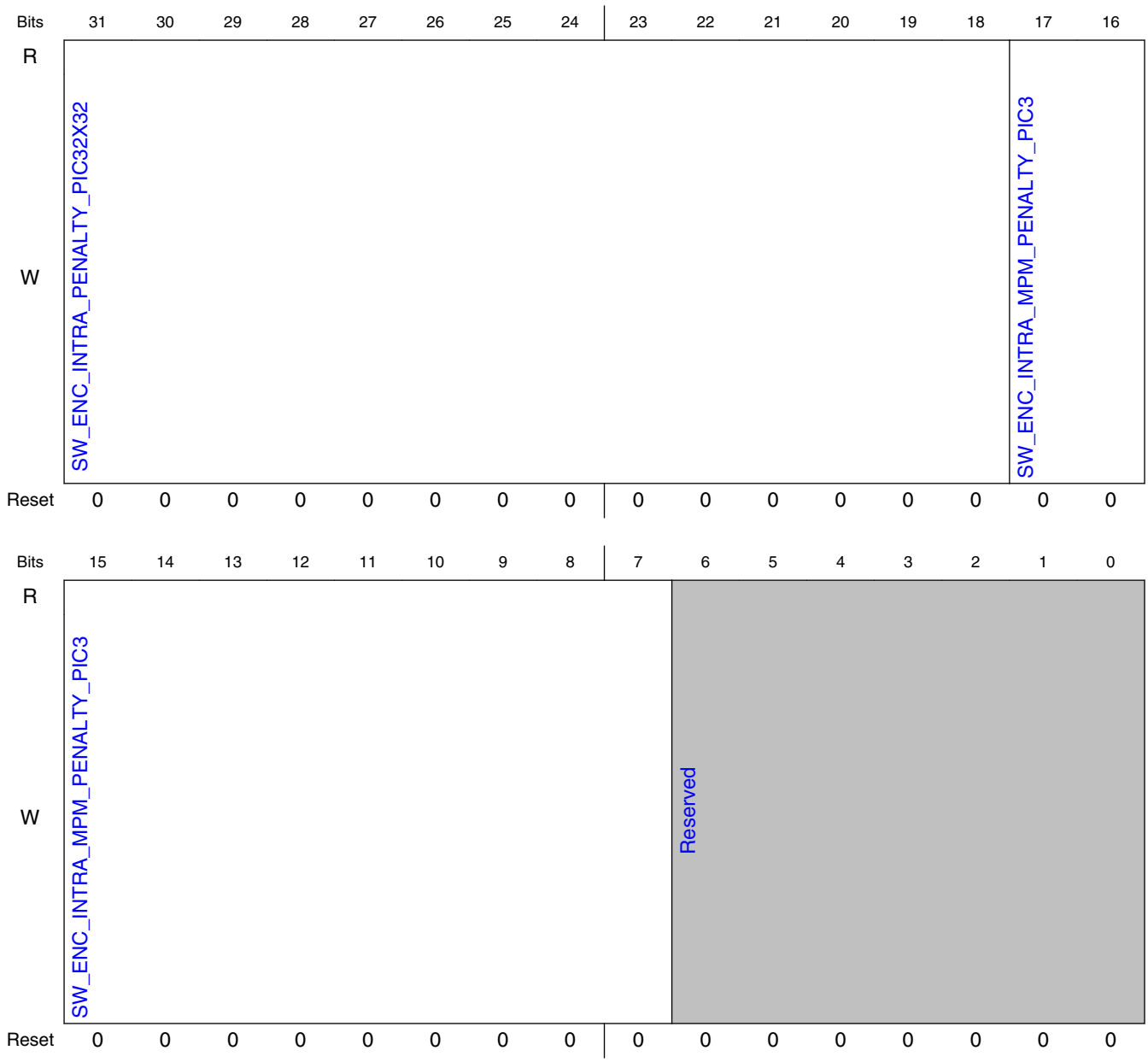
16.3.5.1.28.1 Offset

Register	Offset
swreg28	70h

16.3.5.1.28.2 Function

intra penalty for picture2

16.3.5.1.28.3 Diagram



16.3.5.1.28.4 Fields

Field	Function
31-18 SW_ENC_INTRA_PENALTY_PIC32X32	intra penalty for pic32x32
17-7	intra MPM penalty for pic3

Table continues on the next page...

Field	Function
SW_ENC_INTRA_MPM_PENALTY_PIC3	
6-0 —	Reserved

16.3.5.1.29 inter me SATD lambda config 0. For H2V2 or later version. (swreg28_H2V2)

16.3.5.1.29.1 Offset

Register	Offset
swreg28_H2V2	70h

16.3.5.1.29.2 Function

inter me SATD lambda config 0. For H2V2 or later version.

16.3.5.1.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_LAMDA_SATD_ME_0								SW_ENC_LAMDA_SATD_ME_1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_LAMDA_SATD_ME_2								SW_ENC_LAMDA_SATD_ME_3							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.29.4 Fields

Field	Function
31-24 SW_ENC_LAMDA_SATD_ME_0	lambda satd me 0
23-16	lambda satd me 1

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_1	
15-8 SW_ENC_LAMDA_SATD_ME_2	lambda satd me 2
7-0 SW_ENC_LAMDA_SATD_ME_3	lambda satd me 3

16.3.5.1.30 inter me SATD lambda config 0. For H2V5 or later version. (swreg28_H2V5)

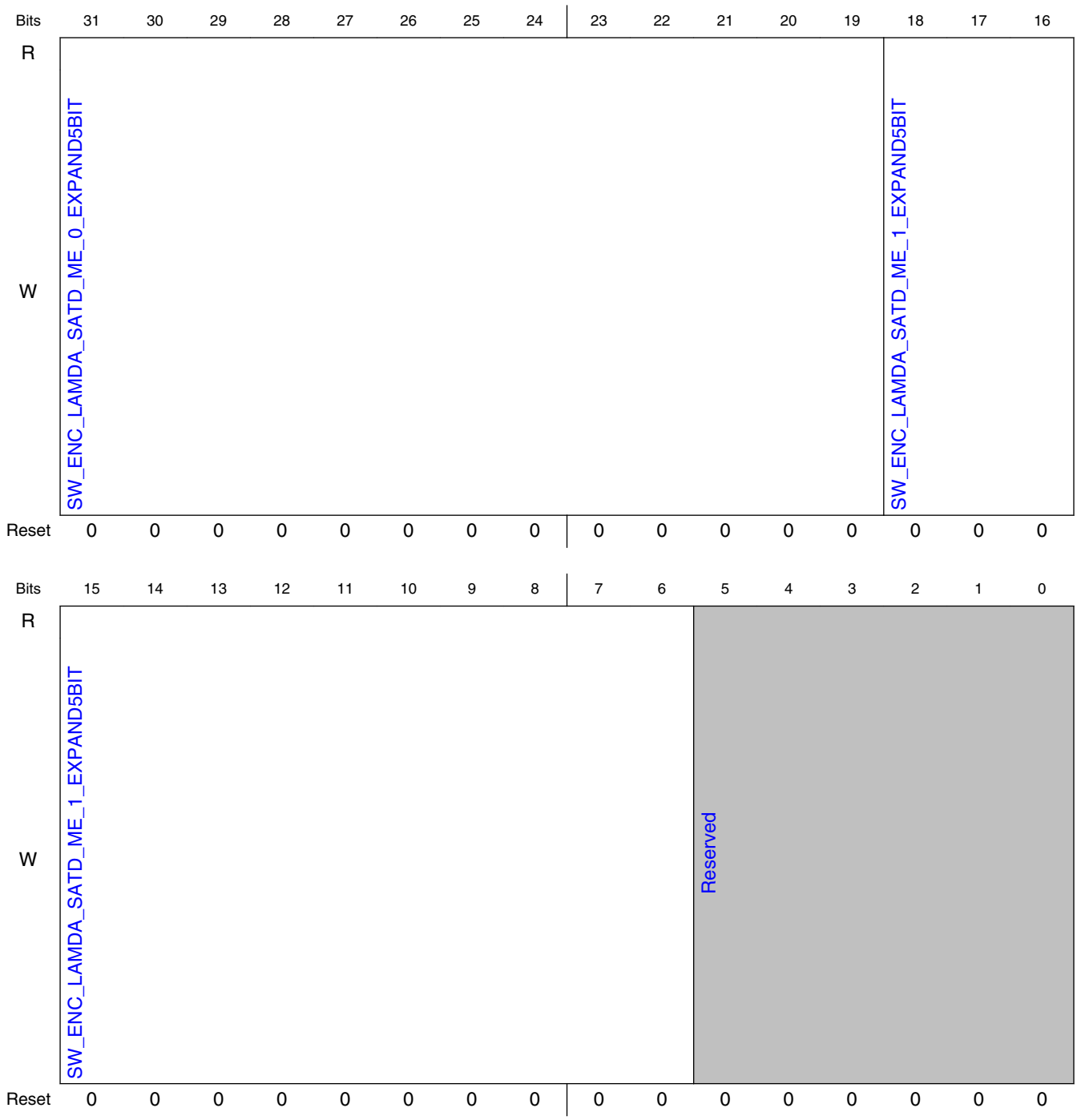
16.3.5.1.30.1 Offset

Register	Offset
swreg28_H2V5	70h

16.3.5.1.30.2 Function

inter me SATD lambda config 0. For H2V5 or later version.

16.3.5.1.30.3 Diagram



16.3.5.1.30.4 Fields

Field	Function
31-19	lambda satd me 0

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_0_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_1_EXPAND5BIT	lambda satd me 1
5-0 —	Reserved

16.3.5.1.31 intra penalty for ROI10. For H2V1 only. (swreg29_H2V1)

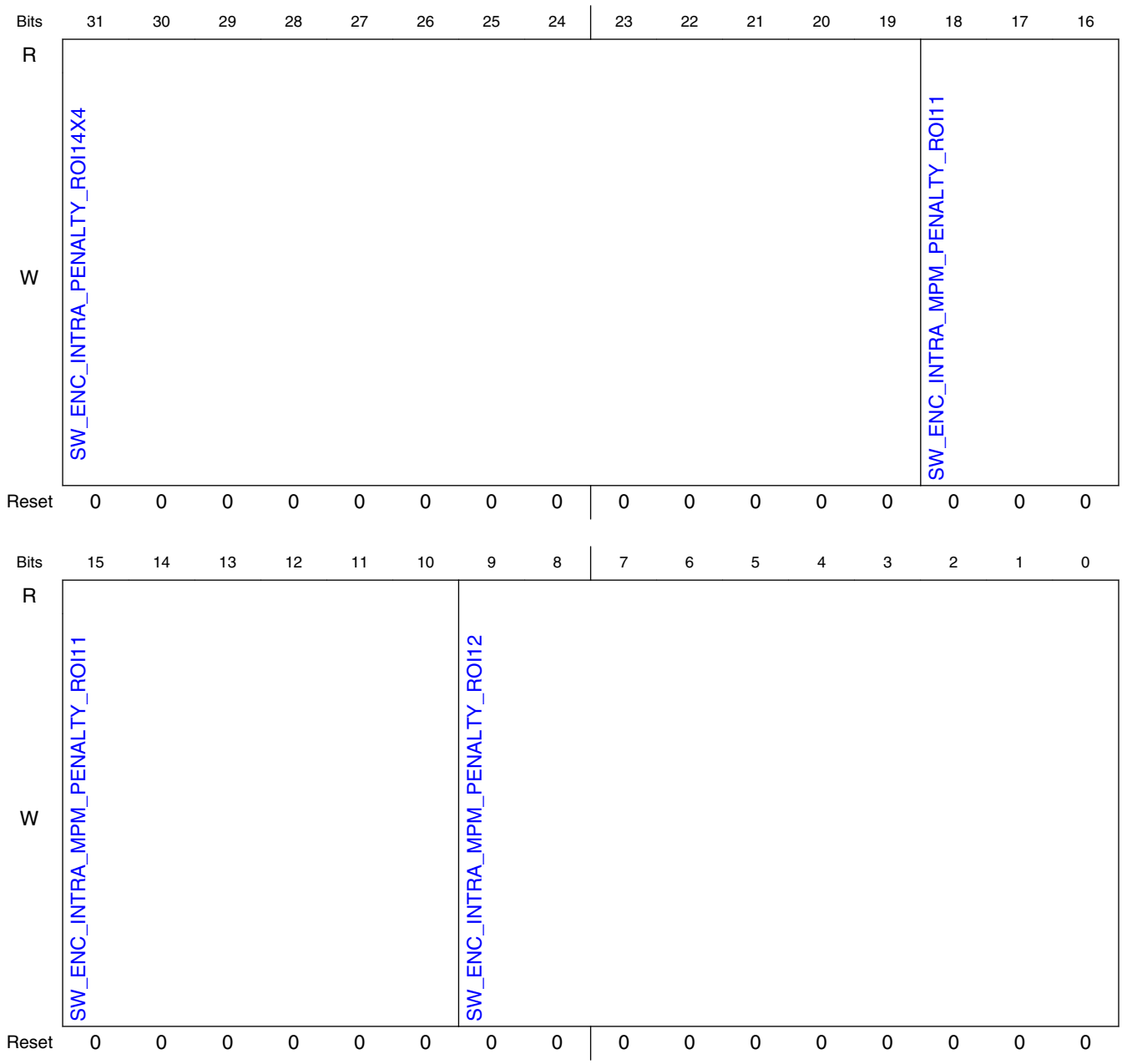
16.3.5.1.31.1 Offset

Register	Offset
swreg29_H2V1	74h

16.3.5.1.31.2 Function

intra penalty for ROI10. For H2V1 only.

16.3.5.1.31.3 Diagram



16.3.5.1.31.4 Fields

Field	Function
31-19 SW_ENC_INTRA_PENALTY_ROI14X4	intra penalty for roi14x4
18-10	intra MPM penalty for roi11

Table continues on the next page...

Field	Function
SW_ENC_INTRA_MPM_PENALTY_ROI11	
9-0 SW_ENC_INTRA_MPM_PENALTY_ROI12	intra MPM penalty for roi12

16.3.5.1.32 inter me SATD lambda config 1.For H2V2 or later version. (swreg29_H2V2)

16.3.5.1.32.1 Offset

Register	Offset
swreg29_H2V2	74h

16.3.5.1.32.2 Function

inter me SATD lambda config 1.For H2V2 or later version.

16.3.5.1.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_LAMDA_SATD_ME_4								SW_ENC_LAMDA_SATD_ME_5							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_LAMDA_SATD_ME_6								SW_ENC_LAMDA_SATD_ME_7							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.32.4 Fields

Field	Function
31-24 SW_ENC_LAMDA_SATD_ME_4	lambda satd me 4

Table continues on the next page...

Field	Function
23-16 SW_ENC_LAMDA_SATD_ME_5	lambda satd me 5
15-8 SW_ENC_LAMDA_SATD_ME_6	lambda satd me 6
7-0 SW_ENC_LAMDA_SATD_ME_7	lambda satd me 7

16.3.5.1.33 inter me SATD lambda config 1.For H2V5 or later version. (swreg29_H2V5)

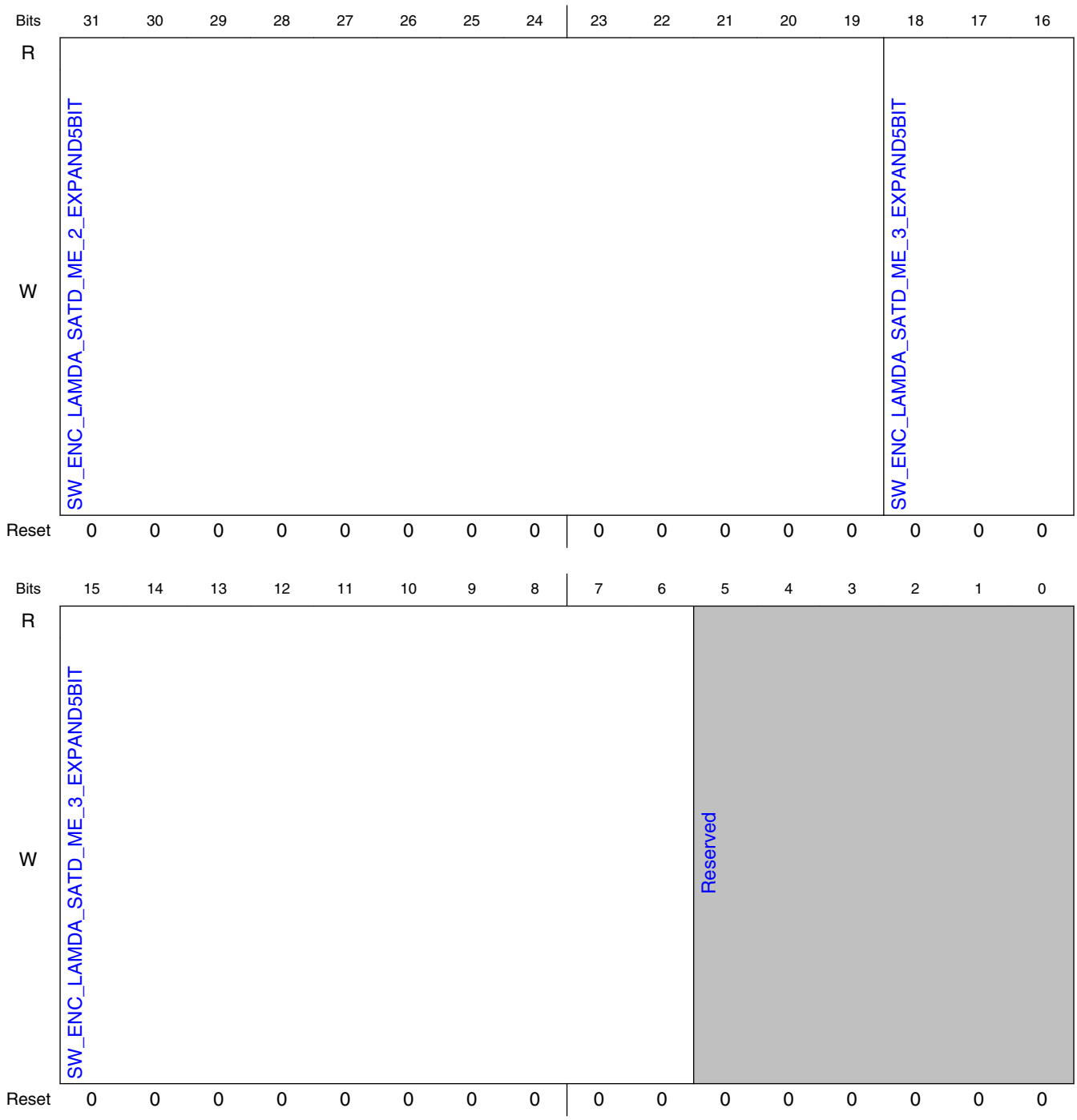
16.3.5.1.33.1 Offset

Register	Offset
swreg29_H2V5	74h

16.3.5.1.33.2 Function

inter me SATD lambda config 1.For H2V5 or later version.

16.3.5.1.33.3 Diagram



16.3.5.1.33.4 Fields

Field	Function
31-19	lambda satd me 2

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_2_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_3_EXPAND5BIT	lambda satd me 3
5-0 —	Reserved

16.3.5.1.34 intra penalty for ROI11. For H2V1 only. (swreg30_H2V1)

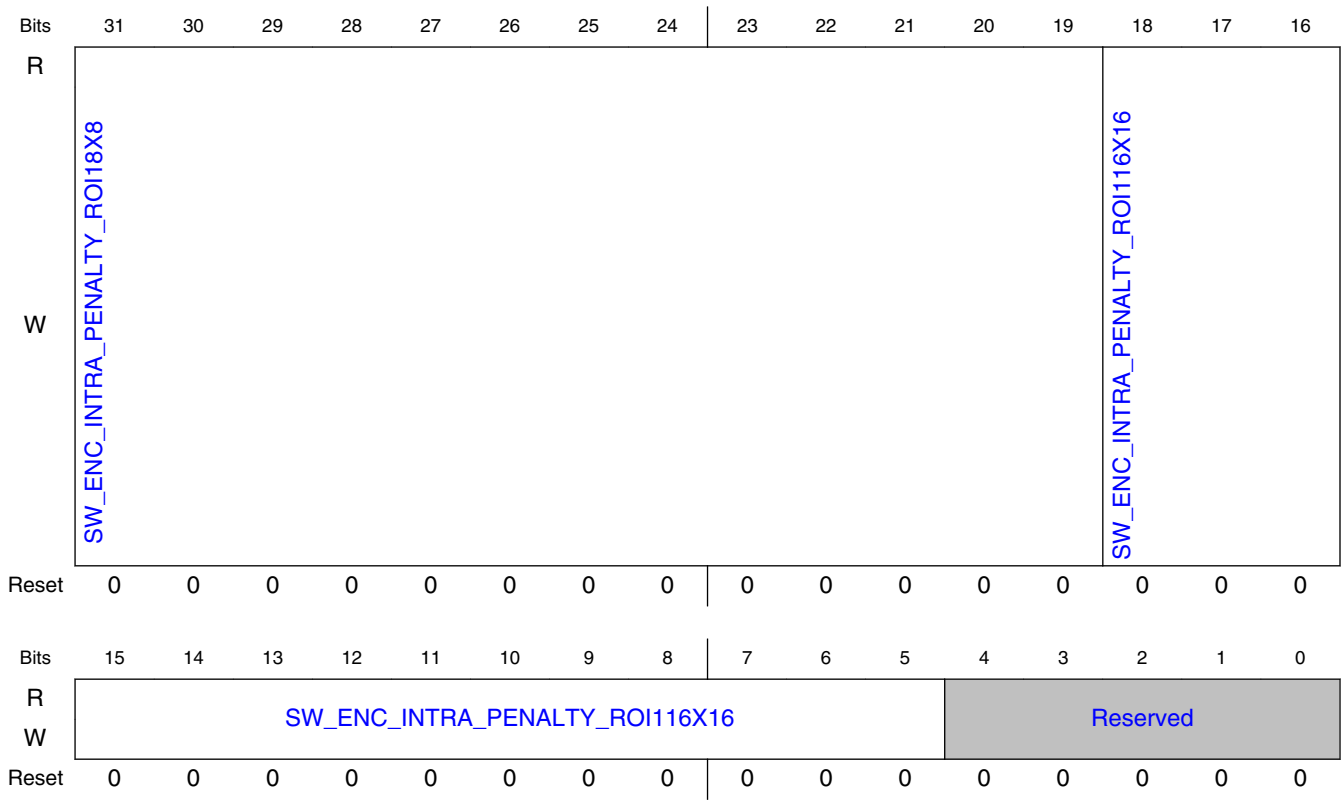
16.3.5.1.34.1 Offset

Register	Offset
swreg30_H2V1	78h

16.3.5.1.34.2 Function

intra penalty for ROI11. For H2V1 only.

16.3.5.1.34.3 Diagram



16.3.5.1.34.4 Fields

Field	Function
31-19 SW_ENC_INTRA_PENALTY_ROI18X8	intra penalty for roi18x8
18-5 SW_ENC_INTRA_PENALTY_ROI116X16	intra penalty for roi116x16
4-0 —	Reserved

16.3.5.1.35 inter me SATD lambda config 2. For H2V2 or later version. (swreg30_H2V2)

16.3.5.1.35.1 Offset

Register	Offset
swreg30_H2V2	78h

16.3.5.1.35.2 Function

inter me SATD lambda config 2. For H2V2 or later version.

16.3.5.1.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_LAMDA_SATD_ME_8								SW_ENC_LAMDA_SATD_ME_9							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_LAMDA_SATD_ME_10								SW_ENC_LAMDA_SATD_ME_11							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.35.4 Fields

Field	Function
31-24 SW_ENC_LAMDA_SATD_ME_8	lambda satd me 8
23-16 SW_ENC_LAMDA_SATD_ME_9	lambda satd me 9
15-8 SW_ENC_LAMDA_SATD_ME_10	lambda satd me 10
7-0 SW_ENC_LAMDA_SATD_ME_11	lambda satd me 11

16.3.5.1.36 inter me SATD lambda config 2. For H2V5 or later version. (swreg30_H2V5)

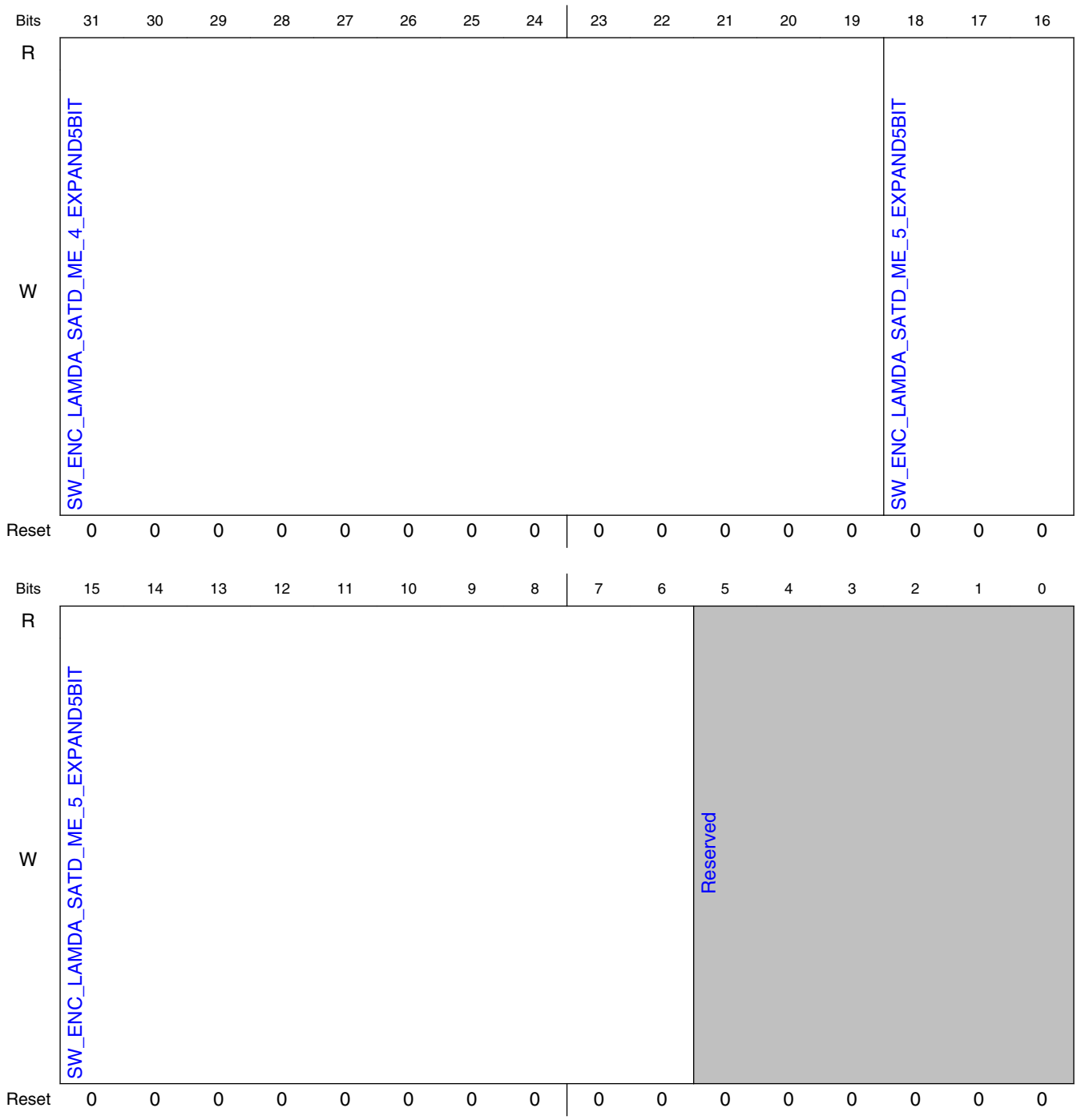
16.3.5.1.36.1 Offset

Register	Offset
swreg30_H2V5	78h

16.3.5.1.36.2 Function

inter me SATD lambda config 2. For H2V5 or later version.

16.3.5.1.36.3 Diagram



16.3.5.1.36.4 Fields

Field	Function
31-19	lambda satd me 4

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_4_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_5_EXPAND5BIT	lambda satd me 5
5-0 —	Reserved

16.3.5.1.37 intra penalty for ROI12. For H2V1 only. (swreg31_H2V1)

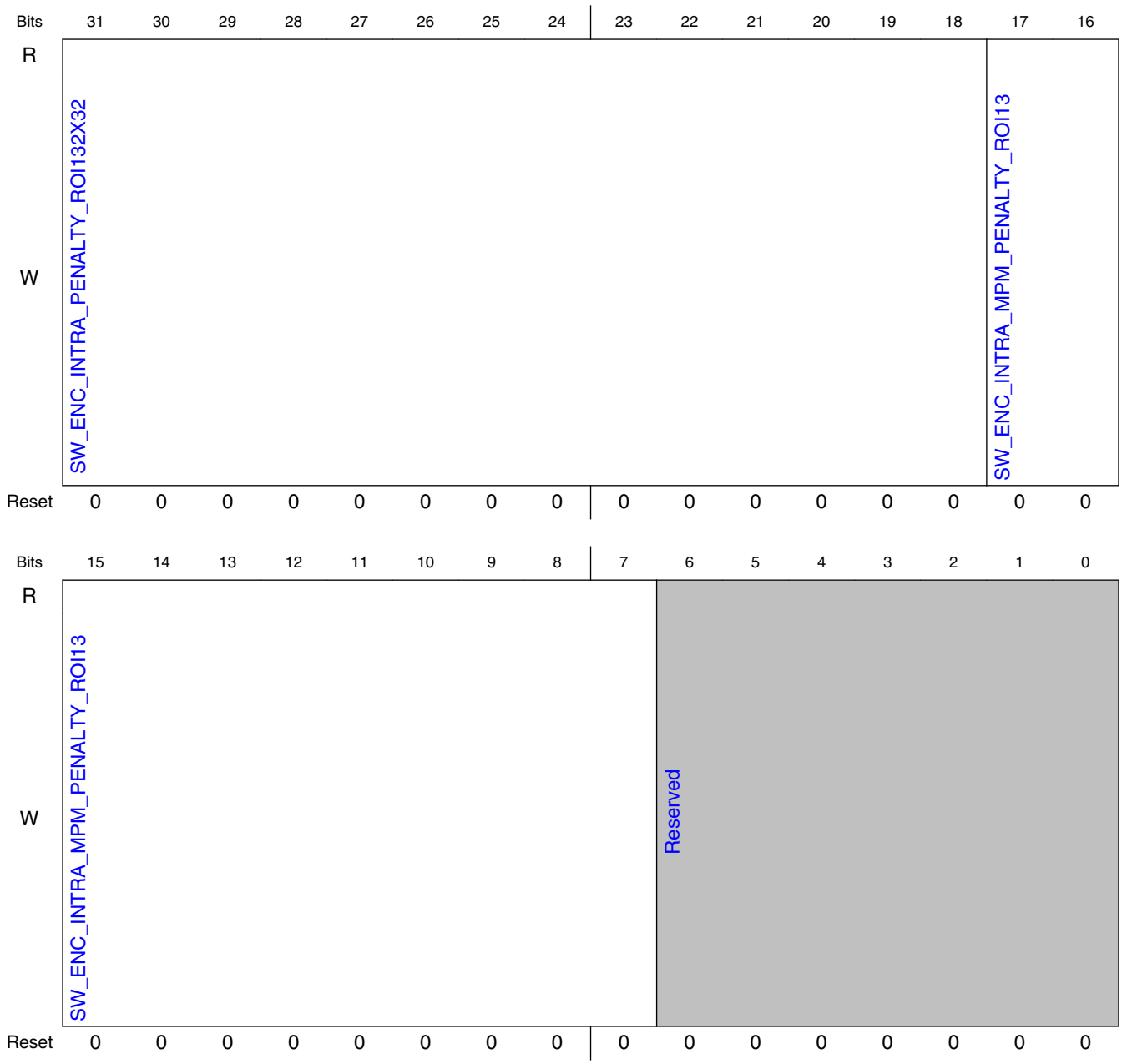
16.3.5.1.37.1 Offset

Register	Offset
swreg31_H2V1	7Ch

16.3.5.1.37.2 Function

intra penalty for ROI12. For H2V1 only.

16.3.5.1.37.3 Diagram



16.3.5.1.37.4 Fields

Field	Function
31-18 SW_ENC_INTRA_PENALTY_ROI132X32	intra penalty for roi132x32
17-7	intra MPM penalty for roi13

Table continues on the next page...

Field	Function
SW_ENC_INTRA_MPM_PENALTY_ROI13	
6-0 —	Reserved

16.3.5.1.38 inter me SATD lambda config 3. For H2V2 or later version. (swreg31_H2V2)

16.3.5.1.38.1 Offset

Register	Offset
swreg31_H2V2	7Ch

16.3.5.1.38.2 Function

inter me SATD lambda config 3. For H2V2 or later version.

16.3.5.1.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_LAMDA_SATD_ME_12								SW_ENC_LAMDA_SATD_ME_13							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_LAMDA_SATD_ME_14								SW_ENC_LAMDA_SATD_ME_15							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.38.4 Fields

Field	Function
31-24 SW_ENC_LAMDA_SATD_ME_12	lambda satd me 12
23-16	lambda satd me 13

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_13	
15-8 SW_ENC_LAMDA_SATD_ME_14	lambda satd me 14
7-0 SW_ENC_LAMDA_SATD_ME_15	lambda satd me 15

16.3.5.1.39 inter me SATD lambda config 3. For H2V5 or later version. (swreg31_H2V5)

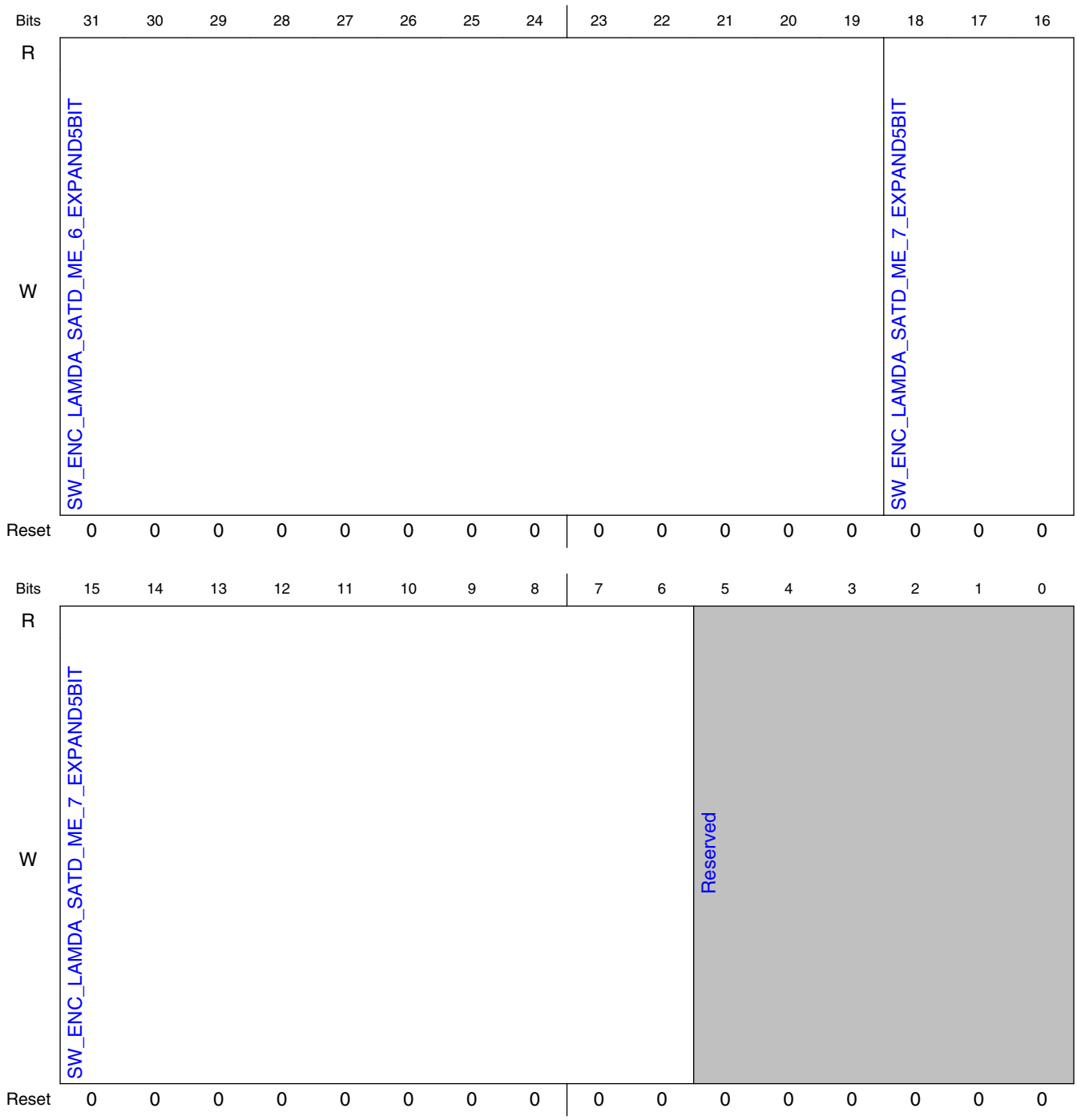
16.3.5.1.39.1 Offset

Register	Offset
swreg31_H2V5	7Ch

16.3.5.1.39.2 Function

inter me SATD lambda config 3. For H2V5 or later version.

16.3.5.1.39.3 Diagram



16.3.5.1.39.4 Fields

Field	Function
31-19	lambda satd me 6

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_6_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_7_EXPAND5BIT	lambda satd me 7
5-0 —	Reserved

16.3.5.1.40 intra penalty for ROI20. For H2V1 only. (swreg32_H2V1)

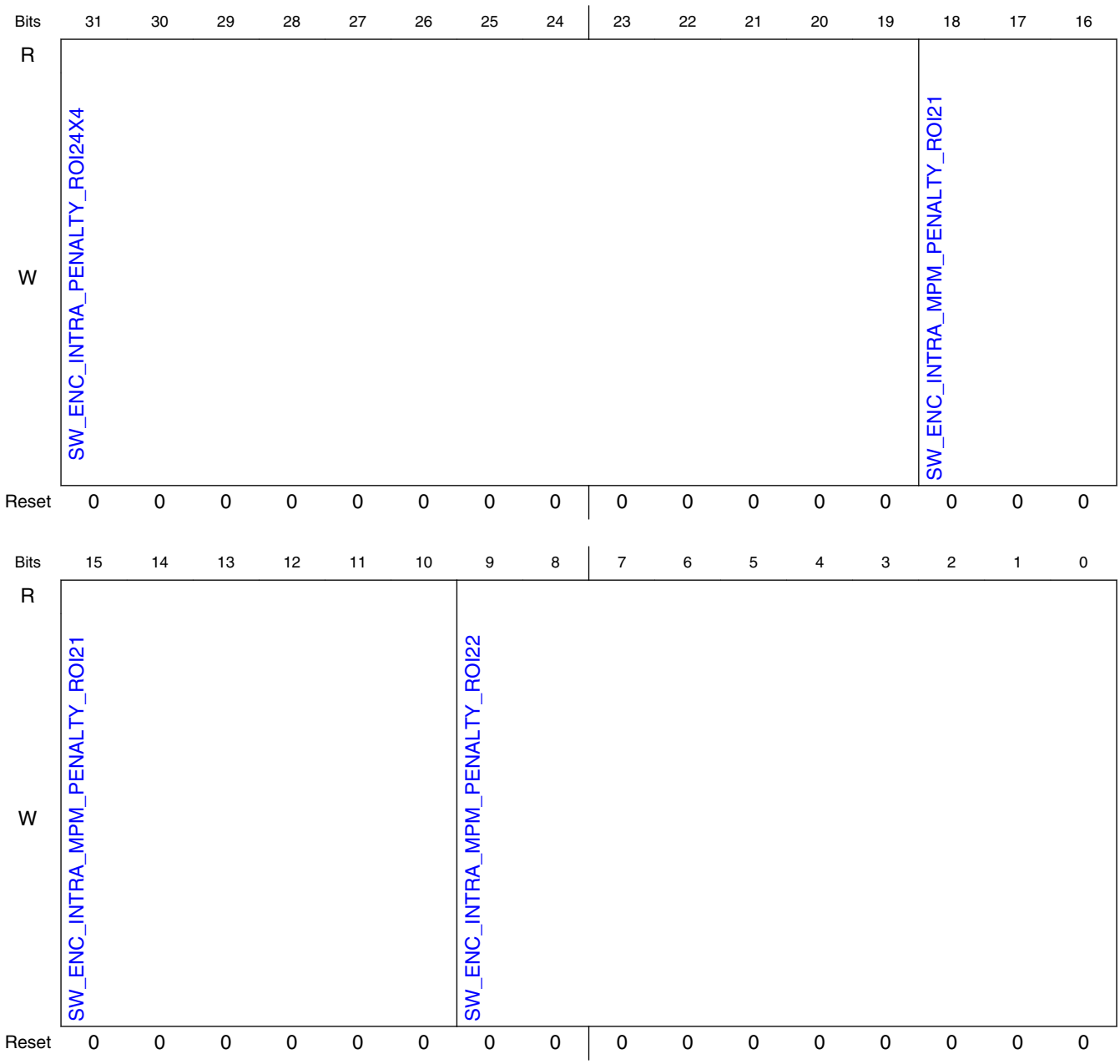
16.3.5.1.40.1 Offset

Register	Offset
swreg32_H2V1	80h

16.3.5.1.40.2 Function

intra penalty for ROI20. For H2V1 only.

16.3.5.1.40.3 Diagram



16.3.5.1.40.4 Fields

Field	Function
31-19 SW_ENC_INTRA_PENALTY_ROI24X4	intra penalty for roi24x4
18-10	intra MPM penalty for roi21

Table continues on the next page...

Field	Function
SW_ENC_INTRA_MPM_PENALTY_ROI21	
9-0 SW_ENC_INTRA_MPM_PENALTY_ROI22	intra MPM penalty for roi22

16.3.5.1.41 inter me SSE lambda config 0. For H2V2 or later version. (swreg32_H2V2)

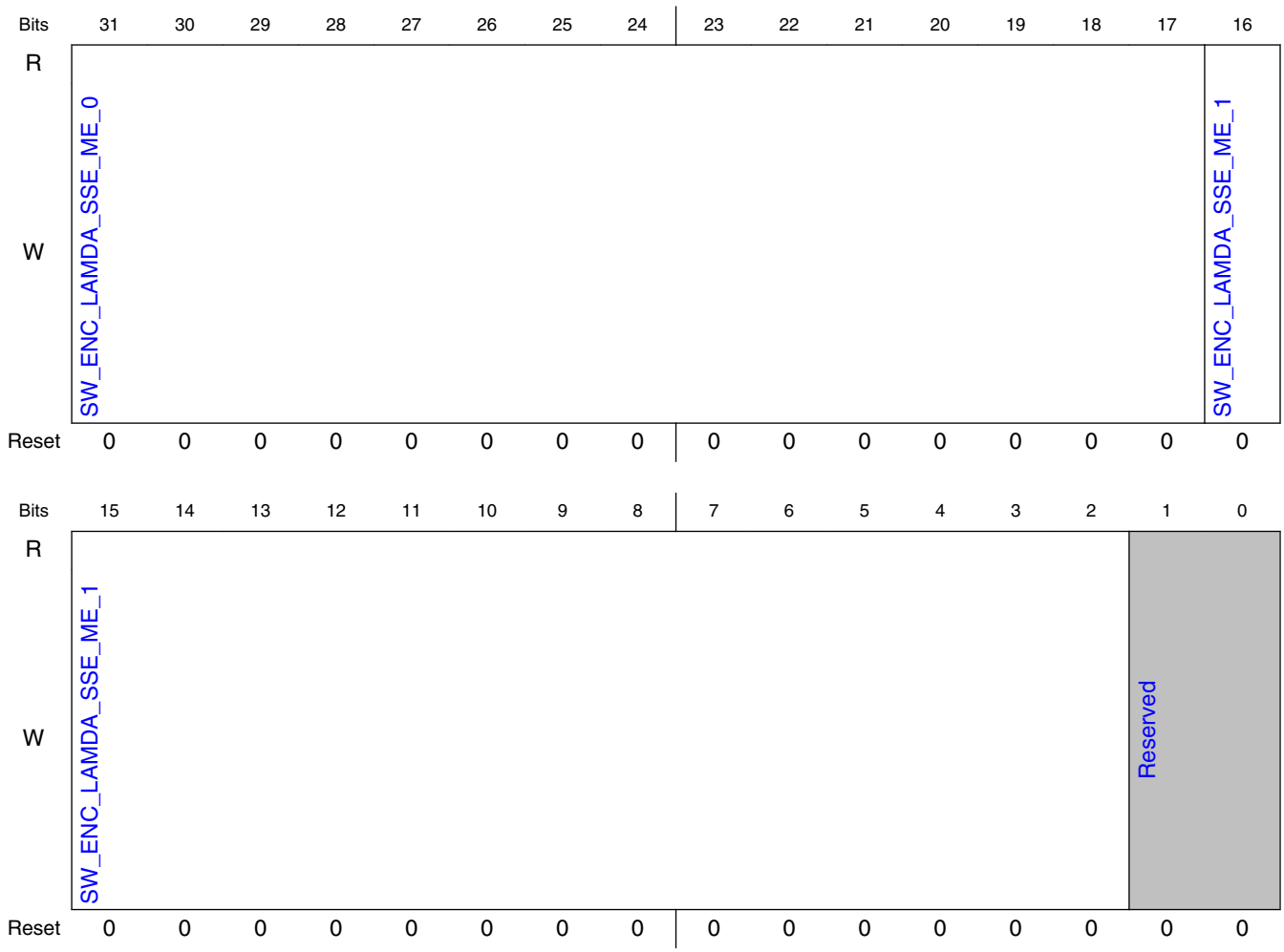
16.3.5.1.41.1 Offset

Register	Offset
swreg32_H2V2	80h

16.3.5.1.41.2 Function

inter me SSE lambda config 0. For H2V2 or later version.

16.3.5.1.41.3 Diagram



16.3.5.1.41.4 Fields

Field	Function
31-17 SW_ENC_LAMDA_SSE_ME_0	inter me SSE lambda 0
16-2 SW_ENC_LAMDA_SSE_ME_1	inter me SSE lambda 1
1-0 —	Reserved

16.3.5.1.42 inter me SATD lambda config 4. For H2V5 or later version. (swreg32_H2V5)

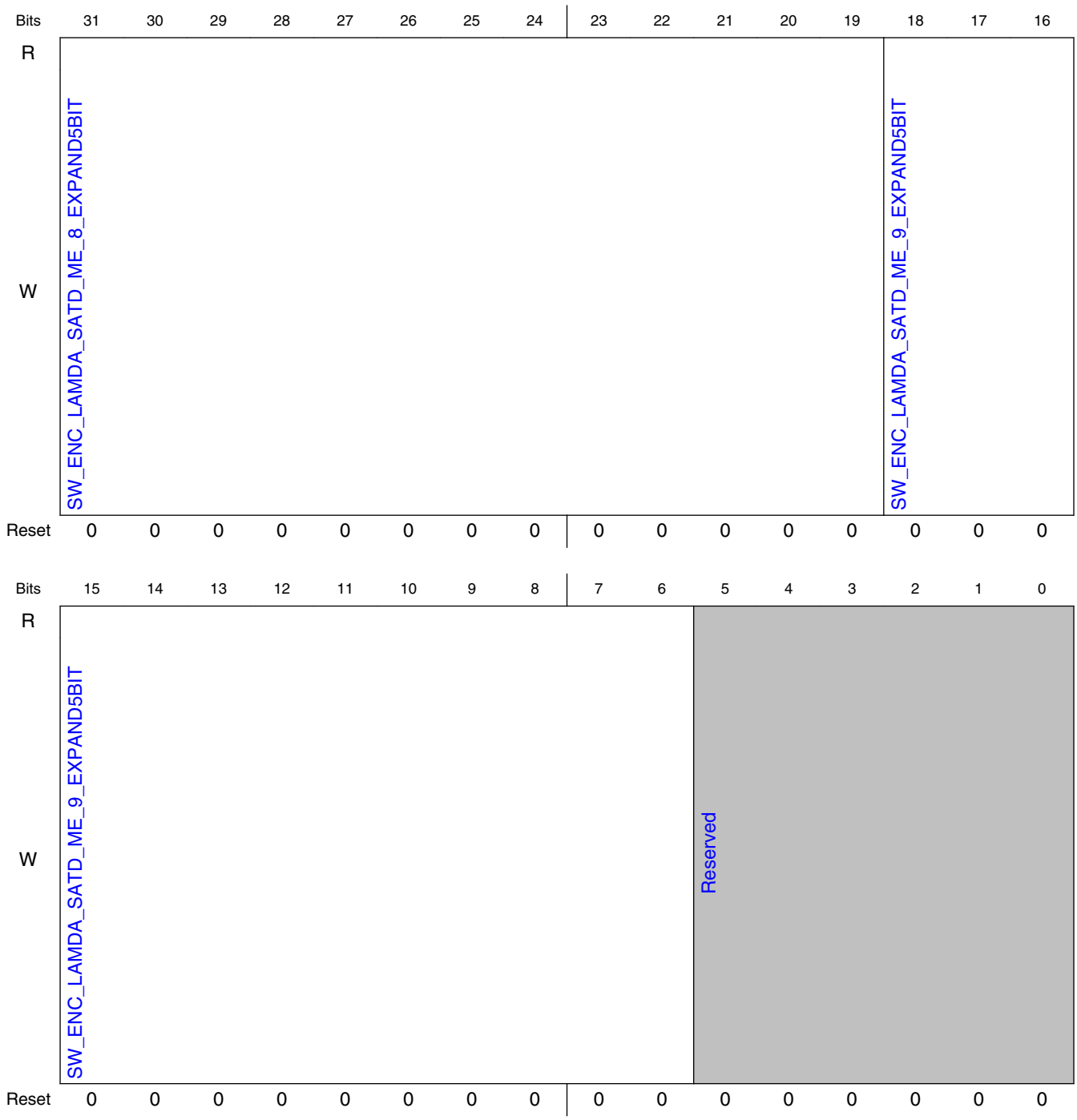
16.3.5.1.42.1 Offset

Register	Offset
swreg32_H2V5	80h

16.3.5.1.42.2 Function

inter me SATD lambda config 4. For H2V5 or later version.

16.3.5.1.42.3 Diagram



16.3.5.1.42.4 Fields

Field	Function
31-19	lambda satd me 8

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_8_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_9_EXPAND5BIT	lambda satd me 9
5-0 —	Reserved

16.3.5.1.43 intra penalty for ROI21. For H2V1 only. (swreg33_H2V1)

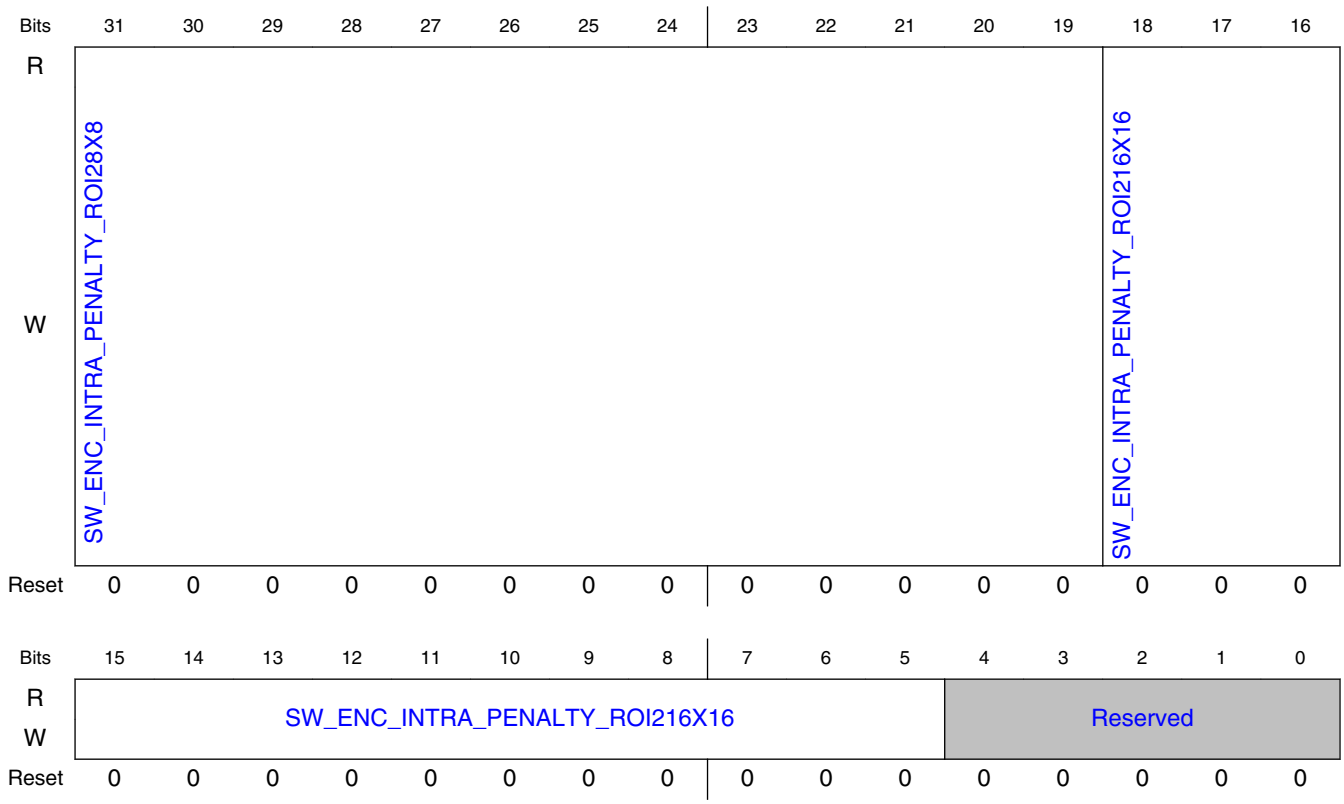
16.3.5.1.43.1 Offset

Register	Offset
swreg33_H2V1	84h

16.3.5.1.43.2 Function

intra penalty for ROI21. For H2V1 only.

16.3.5.1.43.3 Diagram



16.3.5.1.43.4 Fields

Field	Function
31-19 SW_ENC_INTRA_PENALTY_ROI28X8	intra penalty for roi28x8
18-5 SW_ENC_INTRA_PENALTY_ROI216X16	intra penalty for roi216x16
4-0 —	Reserved

16.3.5.1.44 inter me SSE lambda config 1. For H2V2 or later version. (swreg33_H2V2)

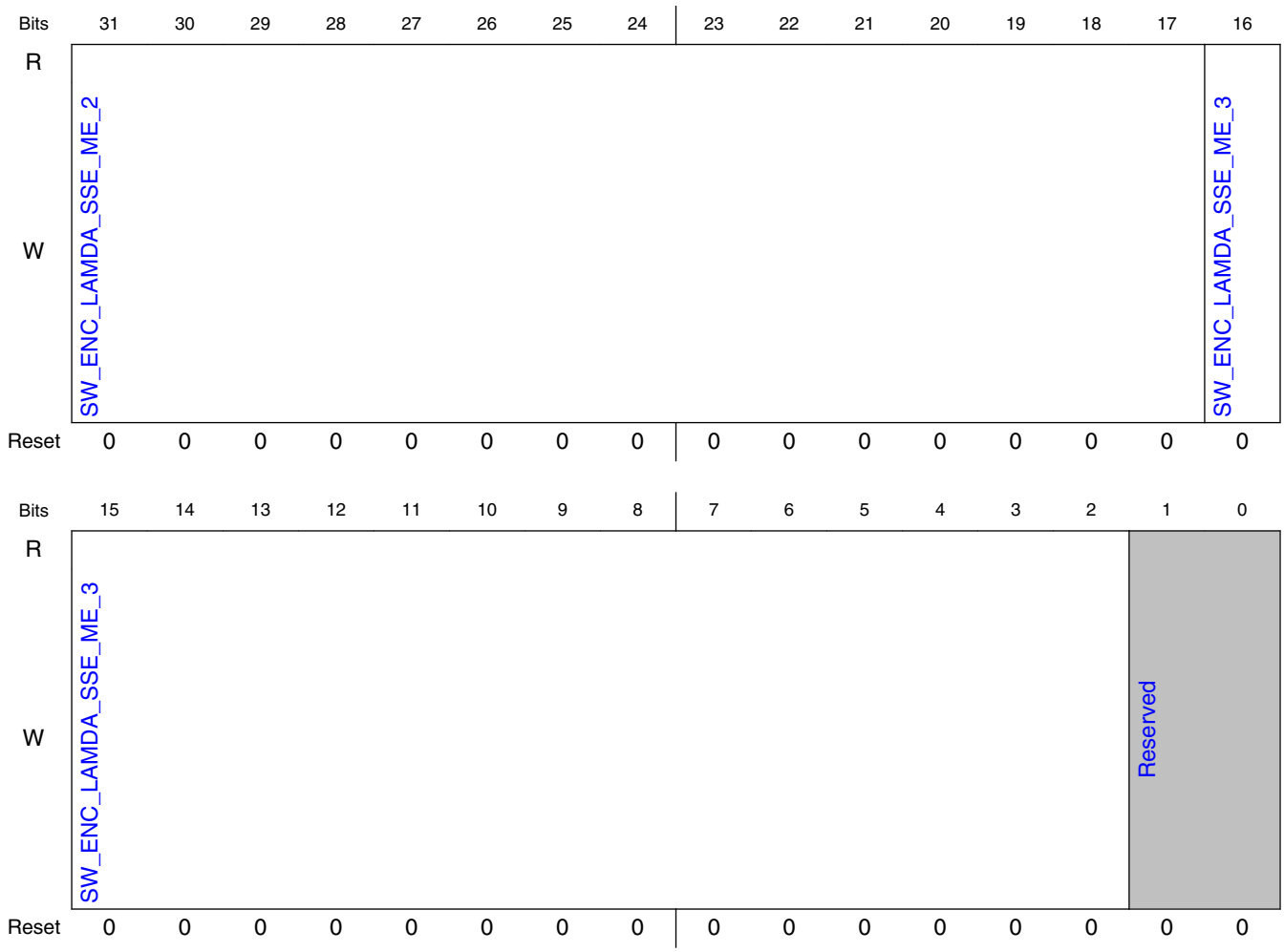
16.3.5.1.44.1 Offset

Register	Offset
swreg33_H2V2	84h

16.3.5.1.44.2 Function

inter me SSE lambda config 1. For H2V2 or later version.

16.3.5.1.44.3 Diagram



16.3.5.1.44.4 Fields

Field	Function
31-17	inter me SSE lambda 2

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SSE_ME_2	
16-2 SW_ENC_LAMDA_SSE_ME_3	inter me SSE lambda 3
1-0 —	Reserved

16.3.5.1.45 inter me SATD lambda config 5. For H2V5 or later version. (swreg33_H2V5)

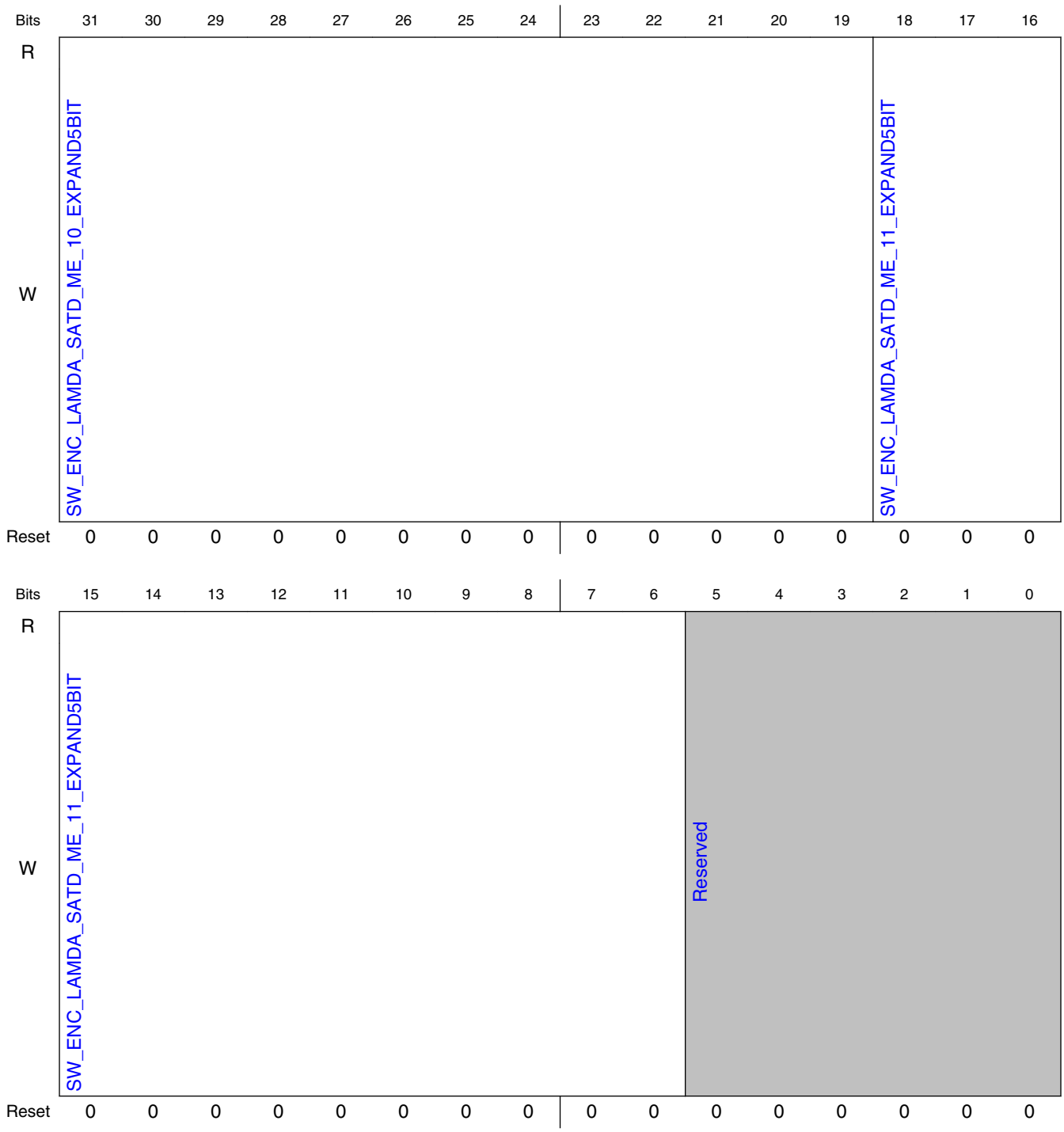
16.3.5.1.45.1 Offset

Register	Offset
swreg33_H2V5	84h

16.3.5.1.45.2 Function

inter me SATD lambda config 5. For H2V5 or later version.

16.3.5.1.45.3 Diagram



16.3.5.1.45.4 Fields

Field	Function
31-19	lambda satd me 10

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_10_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_11_EXPAND5BIT	lambda satd me 11
5-0 —	Reserved

16.3.5.1.46 intra penalty for ROI22. For H2V1 only. (swreg34_H2V1)

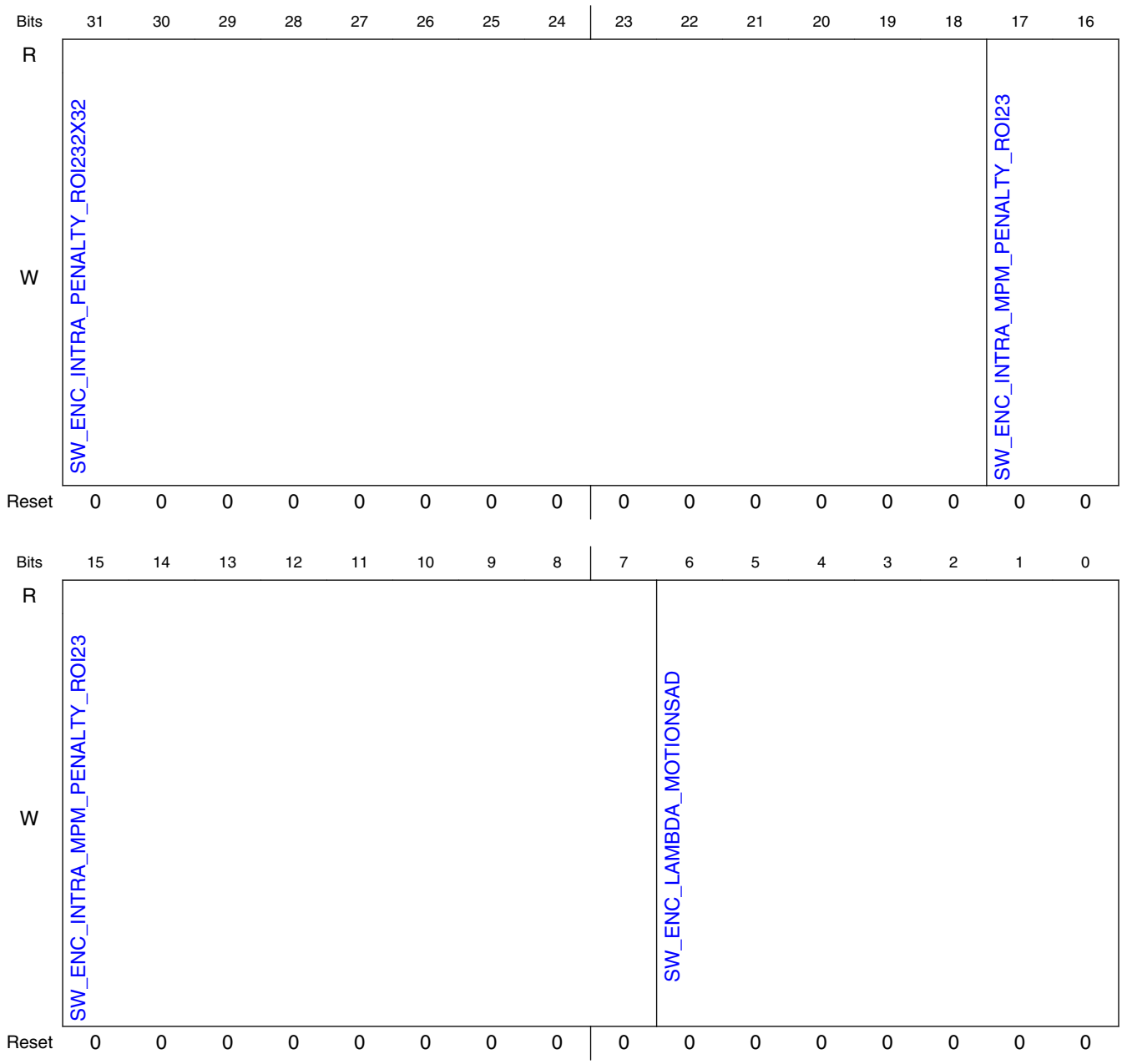
16.3.5.1.46.1 Offset

Register	Offset
swreg34_H2V1	88h

16.3.5.1.46.2 Function

intra penalty for ROI22. For H2V1 only.

16.3.5.1.46.3 Diagram



16.3.5.1.46.4 Fields

Field	Function
31-18 SW_ENC_INTRA_PENALTY_ROI232X32	intra penalty for roi232x32
17-7	intra MPM penalty for roi23

Table continues on the next page...

Field	Function
SW_ENC_INTRA_MPM_PENALTY_ROI23	
6-0 SW_ENC_LAMBDA_MOTIONSAD	lamda for motion SAD

16.3.5.1.47 inter me SSE lambda config 2. For H2V2 or later version. (swreg34_H2V2)

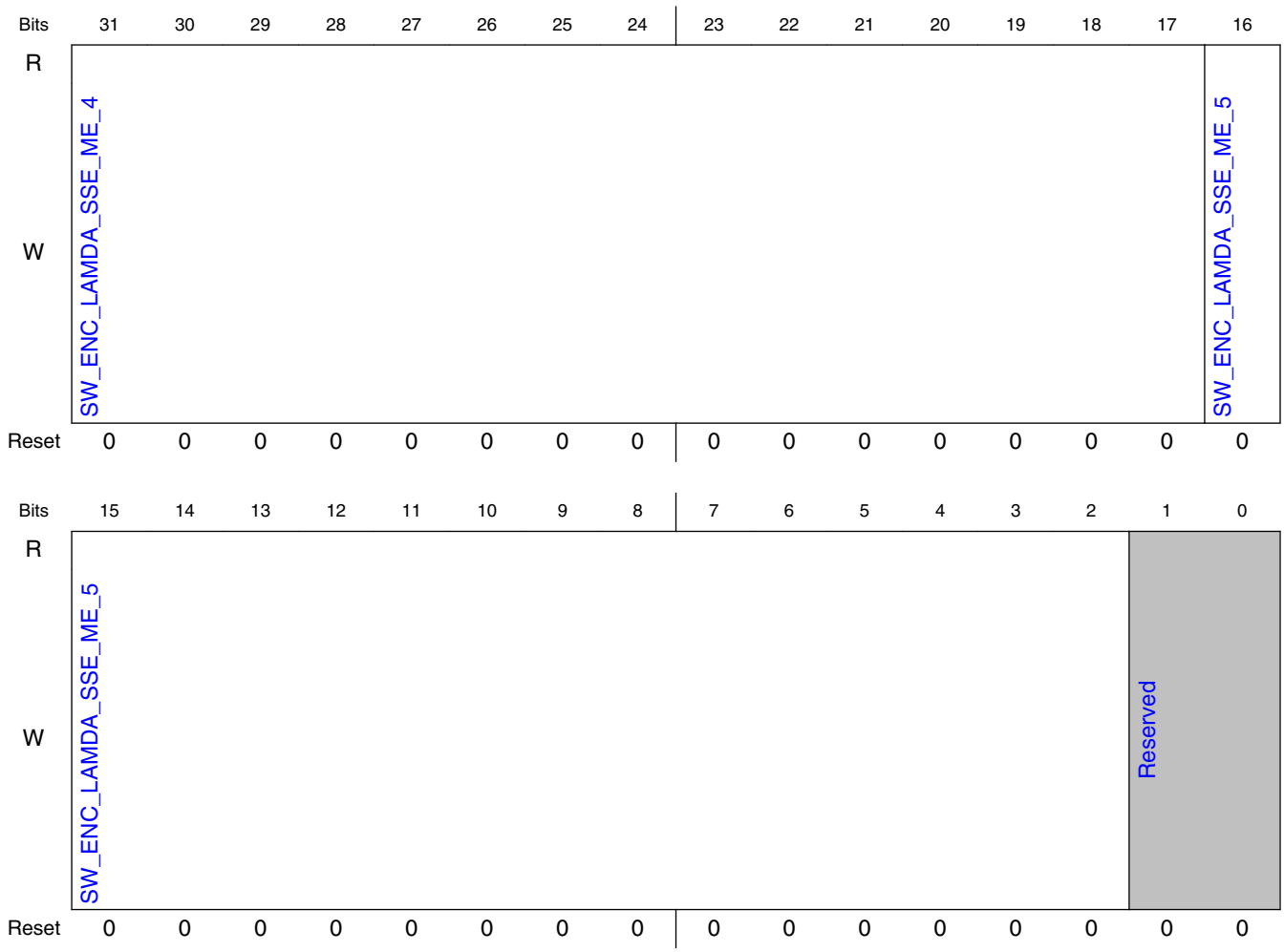
16.3.5.1.47.1 Offset

Register	Offset
swreg34_H2V2	88h

16.3.5.1.47.2 Function

inter me SSE lambda config 2. For H2V2 or later version.

16.3.5.1.47.3 Diagram



16.3.5.1.47.4 Fields

Field	Function
31-17 SW_ENC_LAMDA_SSE_ME_4	inter me SSE lambda 4
16-2 SW_ENC_LAMDA_SSE_ME_5	inter me SSE lambda 5
1-0 —	Reserved

16.3.5.1.48 inter me SATD lambda config 6. For H2V5 or later version. (swreg34_H2V5)

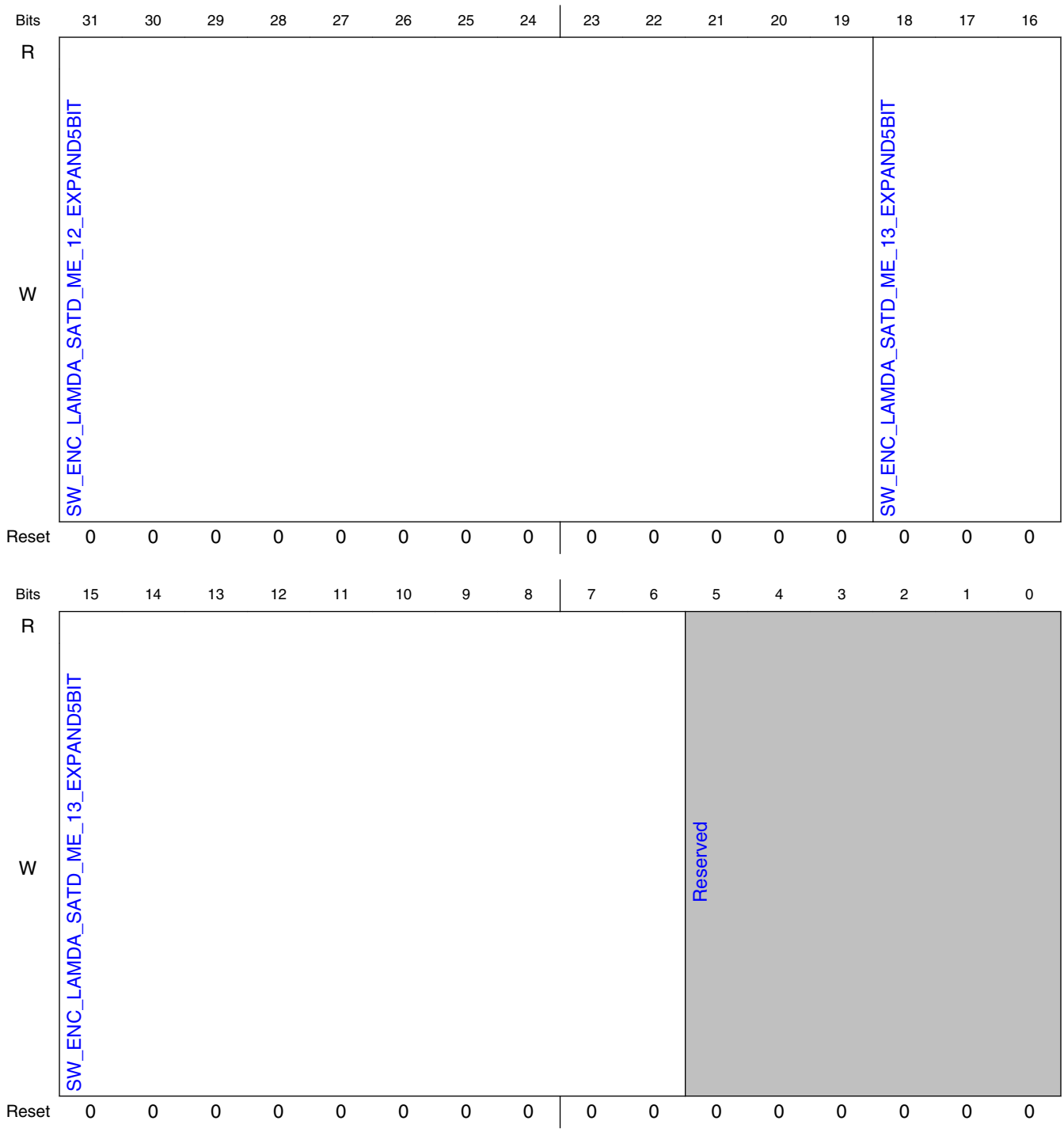
16.3.5.1.48.1 Offset

Register	Offset
swreg34_H2V5	88h

16.3.5.1.48.2 Function

inter me SATD lambda config 6. For H2V5 or later version.

16.3.5.1.48.3 Diagram



16.3.5.1.48.4 Fields

Field	Function
31-19	lambda satd me 12

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_12_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_13_EXPAND5BIT	lambda satd me 13
5-0 —	Reserved

16.3.5.1.49 inter prediction parameters1 (swreg35)

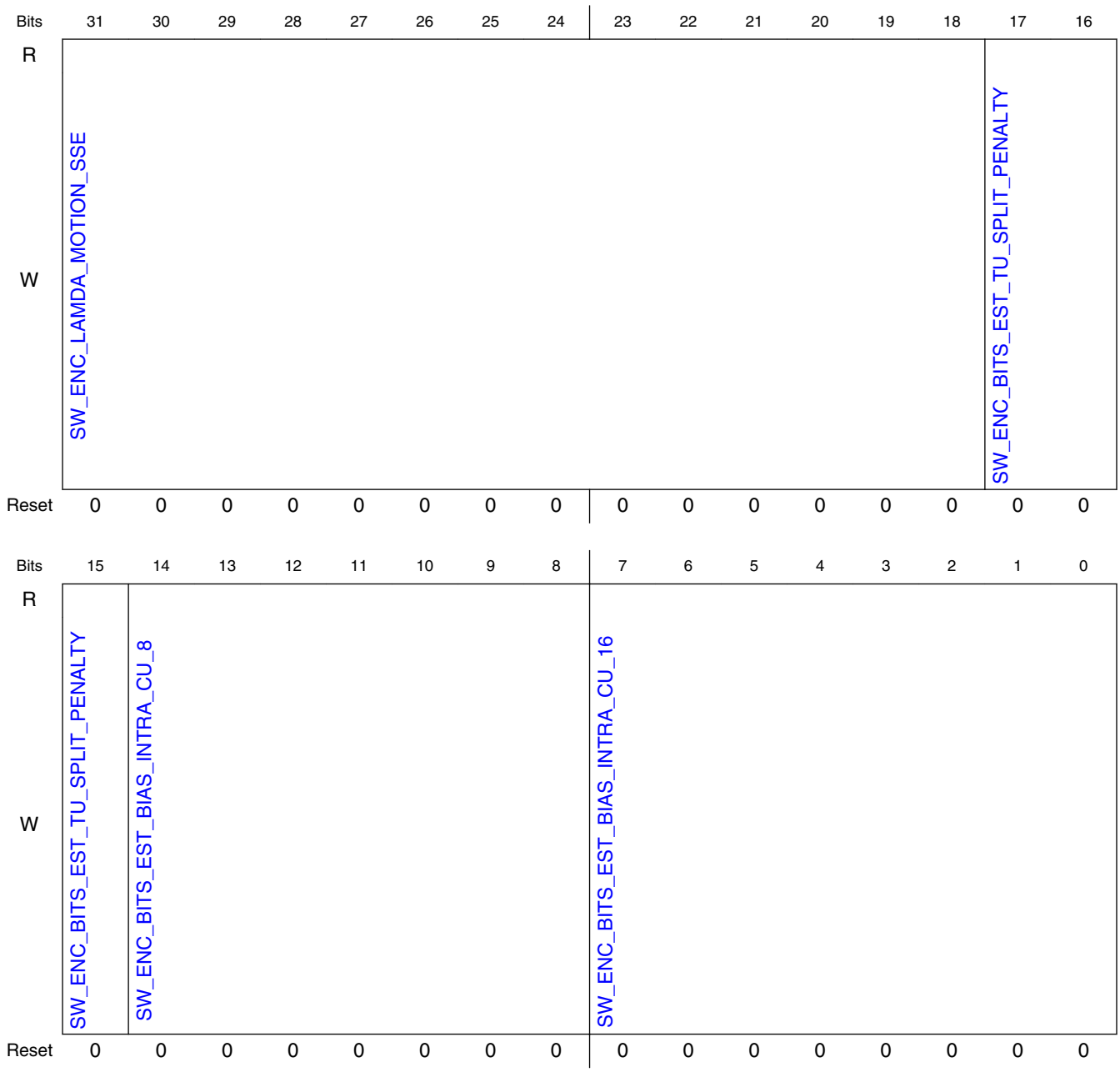
16.3.5.1.49.1 Offset

Register	Offset
swreg35	8Ch

16.3.5.1.49.2 Function

inter prediction parameters1

16.3.5.1.49.3 Diagram



16.3.5.1.49.4 Fields

Field	Function
31-18 SW_ENC_LAMDA_MOTION_SSE	lambda for motion SSE

Table continues on the next page...

Field	Function
17-15 SW_ENC_BITS_EST_TU_SPLIT_PENALTY	bits estimation for tu split penalty
14-8 SW_ENC_BITS_EST_BIAS_INTRA_CU_8	bits estimation bias for intra cu 8
7-0 SW_ENC_BITS_EST_BIAS_INTRA_CU_16	bits estimation bias for intra cu 16

16.3.5.1.50 inter prediction parameters2 (swreg36)

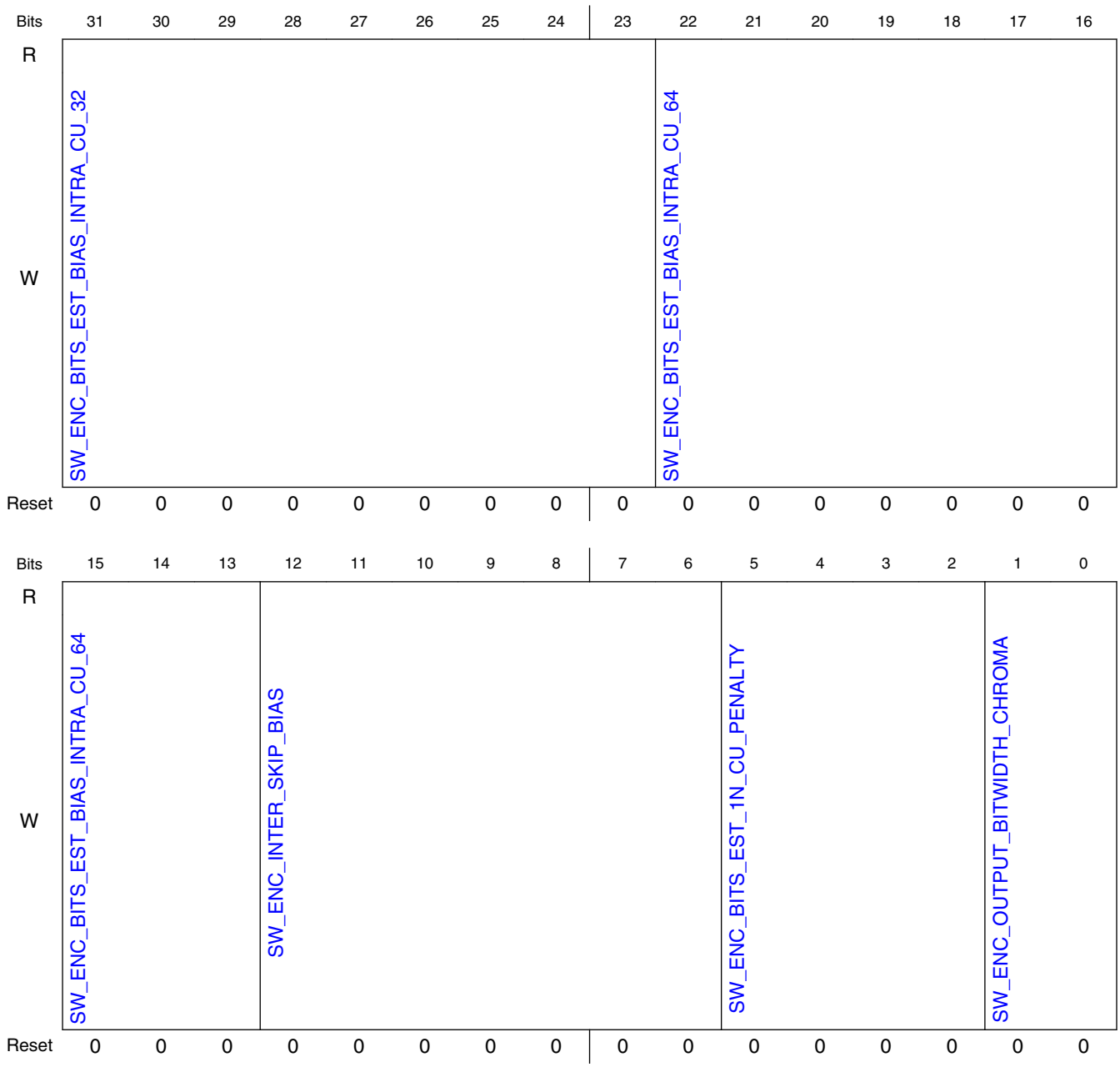
16.3.5.1.50.1 Offset

Register	Offset
swreg36	90h

16.3.5.1.50.2 Function

inter prediction parameters2

16.3.5.1.50.3 Diagram



16.3.5.1.50.4 Fields

Field	Function
31-23 SW_ENC_BITS_EST_BIAS_INTRA_CU_32	bits estimation bias for intra cu 32
22-13	bits estimation bias for intra cu 64

Table continues on the next page...

Field	Function
SW_ENC_BITS_EST_BIAS_INTRA_CU_64	
12-6 SW_ENC_INTER_SKIP_BIAS	inter skip bias
5-2 SW_ENC_BITS_EST_1N_CU_PENALTY	bits estimation 1N cu penalty
1-0 SW_ENC_OUTPUT_BITWIDTH_CHROMA	chroma output bitwidth 00b - 8 bit. 01b - 9 bit. 10b - 10 bit.

16.3.5.1.51 SAO lambda parameter (swreg37)

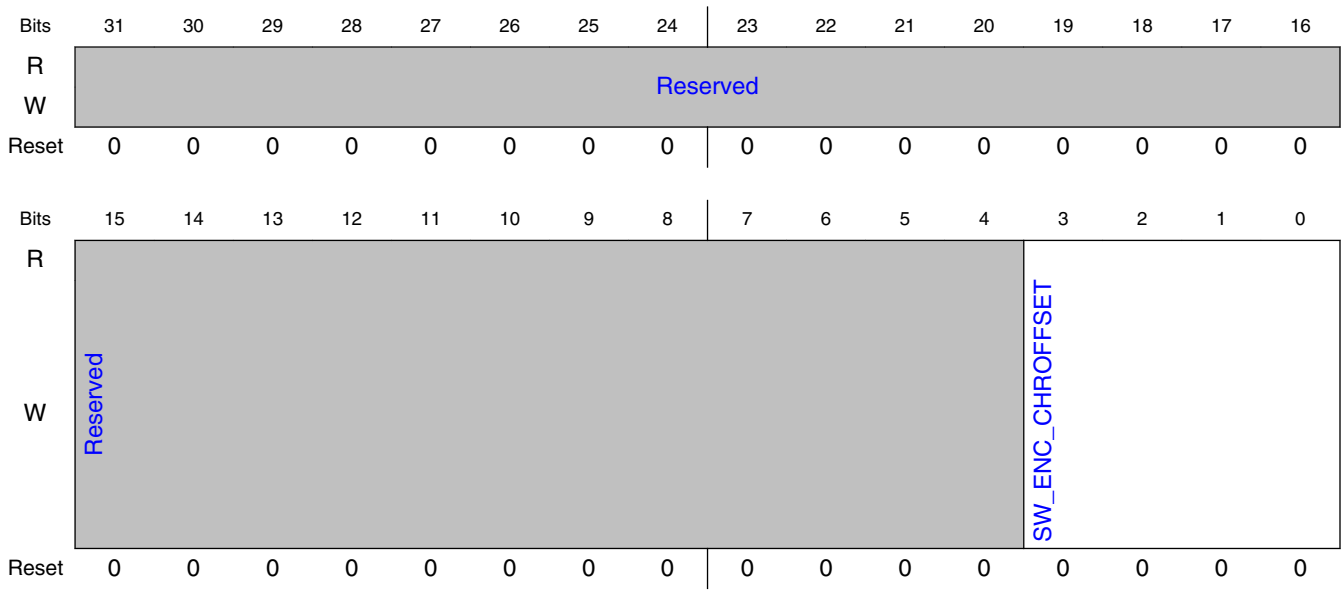
16.3.5.1.51.1 Offset

Register	Offset
swreg37	94h

16.3.5.1.51.2 Function

SAO lambda parameter

16.3.5.1.51.3 Diagram



16.3.5.1.51.4 Fields

Field	Function
31-4 —	Reserved
3-0 SW_ENC_CHR OFFSET	Input chrominance offset (bytes) [0..15]

16.3.5.1.52 Pre-processor configuration (swreg38)

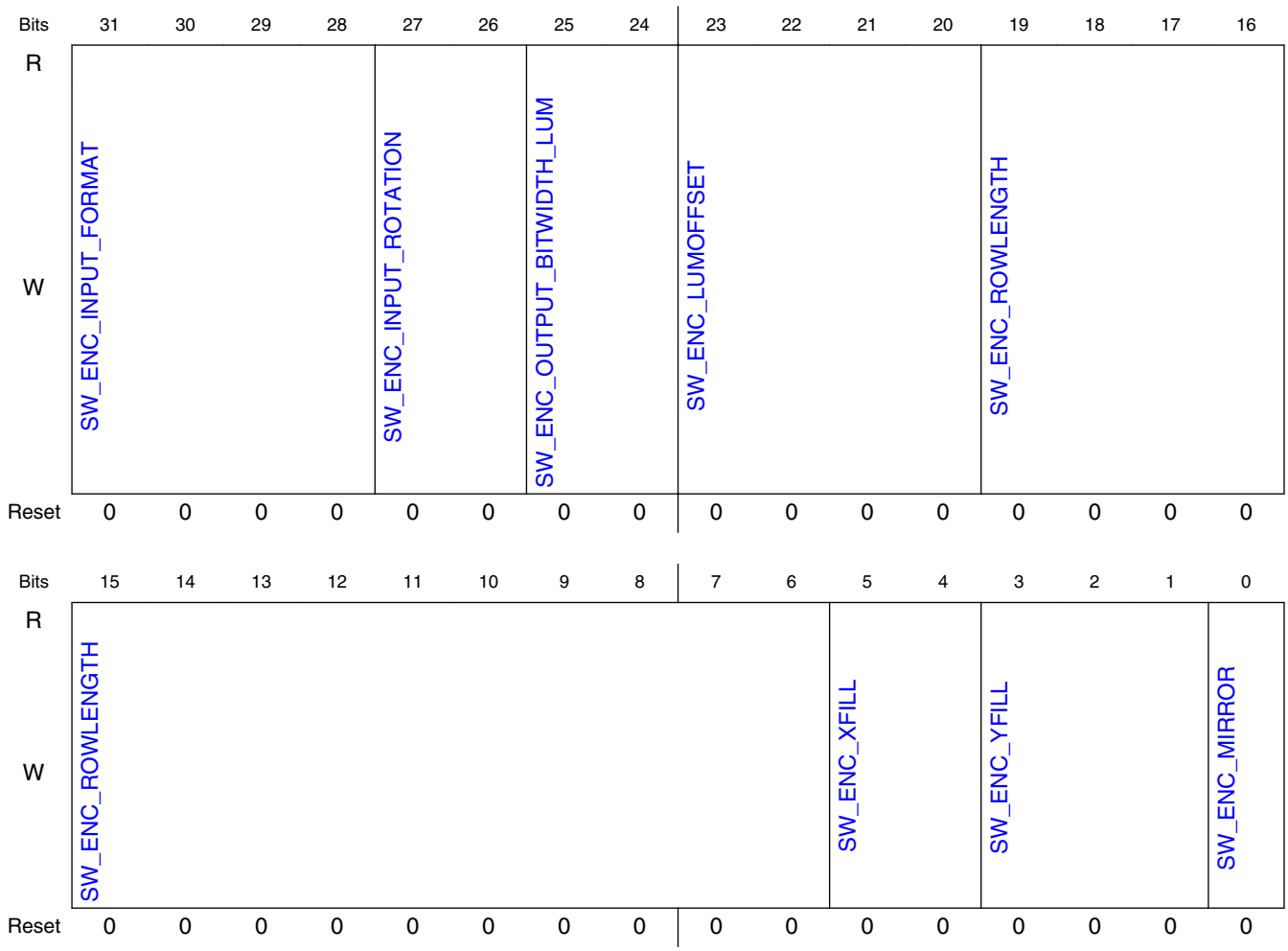
16.3.5.1.52.1 Offset

Register	Offset
swreg38	98h

16.3.5.1.52.2 Function

Pre-processor configuration

16.3.5.1.52.3 Diagram



16.3.5.1.52.4 Fields

Field	Function
31-28 SW_ENC_INPUT_FORMAT	Input image format. inputFormat. 0=YUV420P 0001b - YUV420SP 0010b - YUYV422 0011b - UYVY422 0100b - RGB565 0101b - RGB555 0110b - RGB444 0111b - RGB888 1000b - RGB101010 1001b - I010 1010b - P010 1011b - PACKED10BITPLANAR 1100b - Y0L2 1101b - DAHUAHEVC 1110b - DAHUAH264

Table continues on the next page...

Field	Function
27-26 SW_ENC_INPUT_ROTATION	Input image rotation. 00b - disabled. 01b - 90 degrees right. 10b - 90 degrees left. 11b - Reserved
25-24 SW_ENC_OUTPUT_BITWIDTH_LUM	luma output bitwidth 00b - 8 bit. 01b - 9 bit. 10b - 10 bit.
23-20 SW_ENC_LUMOFFSET	Input luminance offset (bytes) [0..15]
19-6 SW_ENC_ROWLENGTH	Input luminance row length. lumWidthSrc (pixels) [96..8192]
5-4 SW_ENC_XFILL	Overfill pixels on right edge of image div2 [0.1.2.3]
3-1 SW_ENC_YFILL	Overfill pixels on bottom edge of image. YFill. [0...7]
0 SW_ENC_MIRROR	mirror or not. [0.1]

16.3.5.1.53 Pre-processor color conversion parameters0 (swreg39)

16.3.5.1.53.1 Offset

Register	Offset
swreg39	9Ch

16.3.5.1.53.2 Function

Pre-processor color conversion parameters0

16.3.5.1.53.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RGBCOEFFA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RGBCOEFFB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.53.4 Fields

Field	Function
31-16 SW_ENC_RGB COEFFA	RGB to YUV conversion coefficient A
15-0 SW_ENC_RGB COEFFB	RGB to YUV conversion coefficient B

16.3.5.1.54 Pre-processor color conversion parameters1 (swreg40)

16.3.5.1.54.1 Offset

Register	Offset
swreg40	A0h

16.3.5.1.54.2 Function

Pre-processor color conversion parameters1

16.3.5.1.54.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RGBCOEFFC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RGBCOEFFE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.54.4 Fields

Field	Function
31-16 SW_ENC_RGB COEFFC	RGB to YUV conversion coefficient C
15-0 SW_ENC_RGB COEFFE	RGB to YUV conversion coefficient D

16.3.5.1.55 Pre-processor color conversion parameters2 (swreg41)

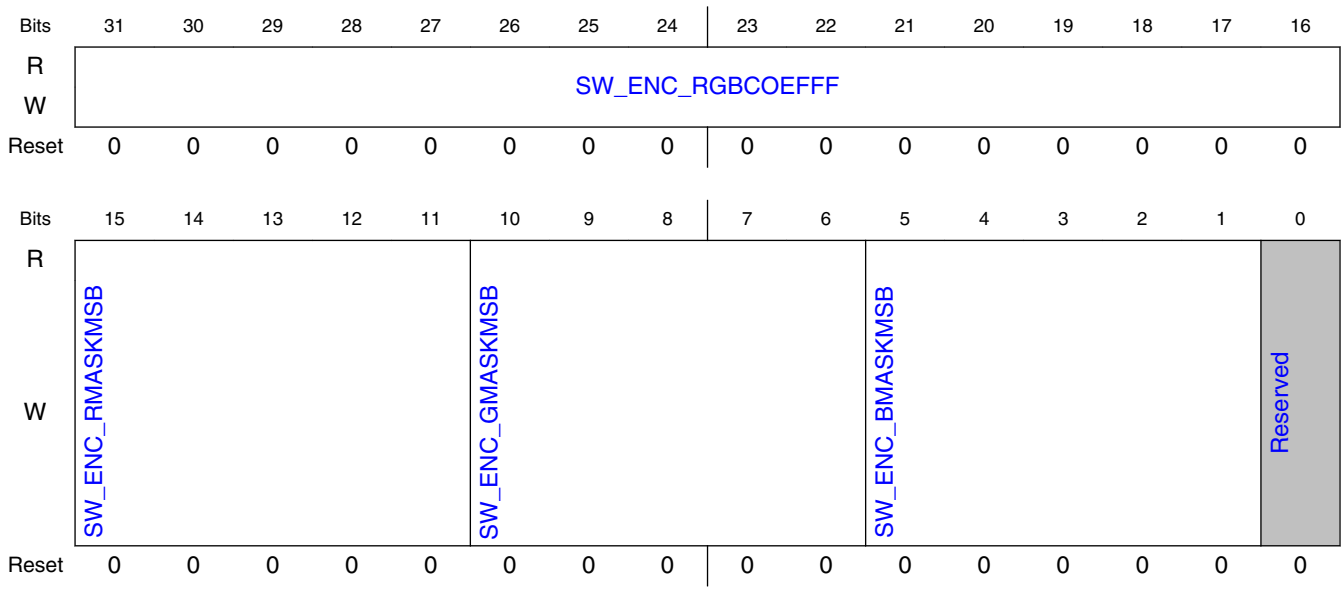
16.3.5.1.55.1 Offset

Register	Offset
swreg41	A4h

16.3.5.1.55.2 Function

Pre-processor color conversion parameters2

16.3.5.1.55.3 Diagram



16.3.5.1.55.4 Fields

Field	Function
31-16 SW_ENC_RGBCOEFFF	RGB to YUV conversion coefficient E
15-11 SW_ENC_RMASKMSB	RGB R-component mask MSB bit position [0..31]
10-6 SW_ENC_GMASKMSB	RGB G-component mask MSB bit position [0..31]
5-1 SW_ENC_BMASKMSB	RGB B-component mask MSB bit position [0..31]
0 —	Reserved

16.3.5.1.56 compressed coefficients base address for SAN module. (swreg46)

16.3.5.1.56.1 Offset

Register	Offset
swreg46	B8h

16.3.5.1.56.2 Function

compressed coefficients base address for SAN module.

16.3.5.1.56.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_COMPRESSEDCOEFF_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_COMPRESSEDCOEFF_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.56.4 Fields

Field	Function
31-0 SW_ENC_COMPRESSEDCOEFF_BASE	Base address for compressed coefficients.

16.3.5.1.57 Base address for recon luma compress table LSB. (swreg60)

16.3.5.1.57.1 Offset

Register	Offset
swreg60	F0h

16.3.5.1.57.2 Function

Base address for recon luma compress table LSB.

16.3.5.1.57.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RECON_LUMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RECON_LUMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.57.4 Fields

Field	Function
31-0 SW_ENC_RECON_LUMA_COMPRESS_TABLE_BASE	Base address for recon luma compress table LSB.

16.3.5.1.58 Base address for recon Chroma compress table LSB (swreg62)

16.3.5.1.58.1 Offset

Register	Offset
swreg62	F8h

16.3.5.1.58.2 Function

Base address for recon Chroma compress table LSB

16.3.5.1.58.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RECON_CHROMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RECON_CHROMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.58.4 Fields

Field	Function
31-0 SW_ENC_RECON_CHROMA_COMPRESS_TABLE_BASE	Base address for recon Chroma compress table LSB.

16.3.5.1.59 Base address for list 0 ref 0 luma compress table LSB. (swreg64)

16.3.5.1.59.1 Offset

Register	Offset
swreg64	100h

16.3.5.1.59.2 Function

Base address for list 0 ref 0 luma compress table LSB.

16.3.5.1.59.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_L0_REF0_LUMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_L0_REF0_LUMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.59.4 Fields

Field	Function
31-0 SW_ENC_L0_REF0_LUMA_COMPRESS_TABLE_BASE	Base address for list 0 ref 0 luma compress table LSB.

16.3.5.1.60 Base address for list 0 ref 0 Chroma compress table LSB. (swreg66)

16.3.5.1.60.1 Offset

Register	Offset
swreg66	108h

16.3.5.1.60.2 Function

Base address for list 0 ref 0 Chroma compress table LSB.

16.3.5.1.60.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_L0_REF0_CHROMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_L0_REF0_CHROMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.60.4 Fields

Field	Function
31-0 SW_ENC_L0_REF0_CHROMA_COMPRESS_TABLE_BASE	Base address for list 0 ref 0 Chroma compress table LSB.

16.3.5.1.61 Base address for recon luma 4n base LSB. (swreg72)

16.3.5.1.61.1 Offset

Register	Offset
swreg72	120h

16.3.5.1.61.2 Function

Base address for recon luma 4n base LSB.

16.3.5.1.61.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RECON_LUMA_4N_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RECON_LUMA_4N_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.61.4 Fields

Field	Function
31-0 SW_ENC_RECON_LUMA_4N_BASE	Base address for recon luma 4n base LSB.

16.3.5.1.62 reference picture reconstructed list0 4n 0 (swreg74)

16.3.5.1.62.1 Offset

Register	Offset
swreg74	128h

16.3.5.1.62.2 Function

reference picture reconstructed list0 4n 0

16.3.5.1.62.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REFPIC_RECON_L0_4N0_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_REFPIC_RECON_L0_4N0_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.62.4 Fields

Field	Function
31-0 SW_ENC_REFPIC_RECON_L0_4N0_BASE	reference picture reconstructed list0 4n 0

16.3.5.1.63 inter prediction parameters3. For H2V1 only. (swreg78_H2V1)

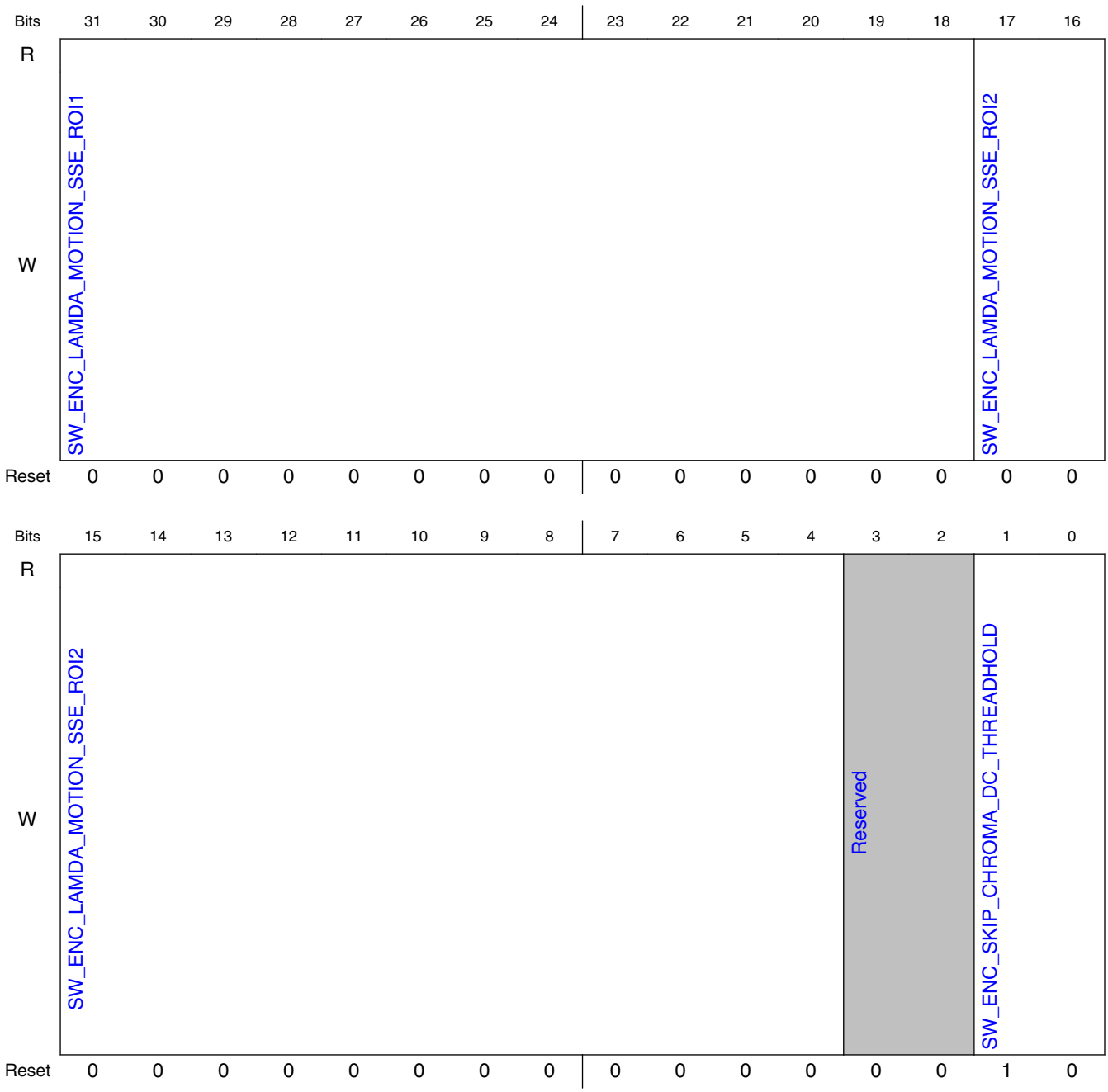
16.3.5.1.63.1 Offset

Register	Offset
swreg78_H2V1	138h

16.3.5.1.63.2 Function

inter prediction parameters3. For H2V1 only.

16.3.5.1.63.3 Diagram



16.3.5.1.63.4 Fields

Field	Function
31-18 SW_ENC_LAMDA_MOTION_SSE_ROI1	roi1 lambda for motion SSE

Table continues on the next page...

Field	Function
17-4 SW_ENC_LAMDA_MOTION_SSE_ROI2	roi2 lambda for motion SSE
3-2 —	Reserved
1-0 SW_ENC_SKIP_CHROMA_DC_THREADHOLD	skip chroma dc threadhold

16.3.5.1.64 inter me SSE lambda config 3. For H2V2 or later version. (swreg78_H2V2)

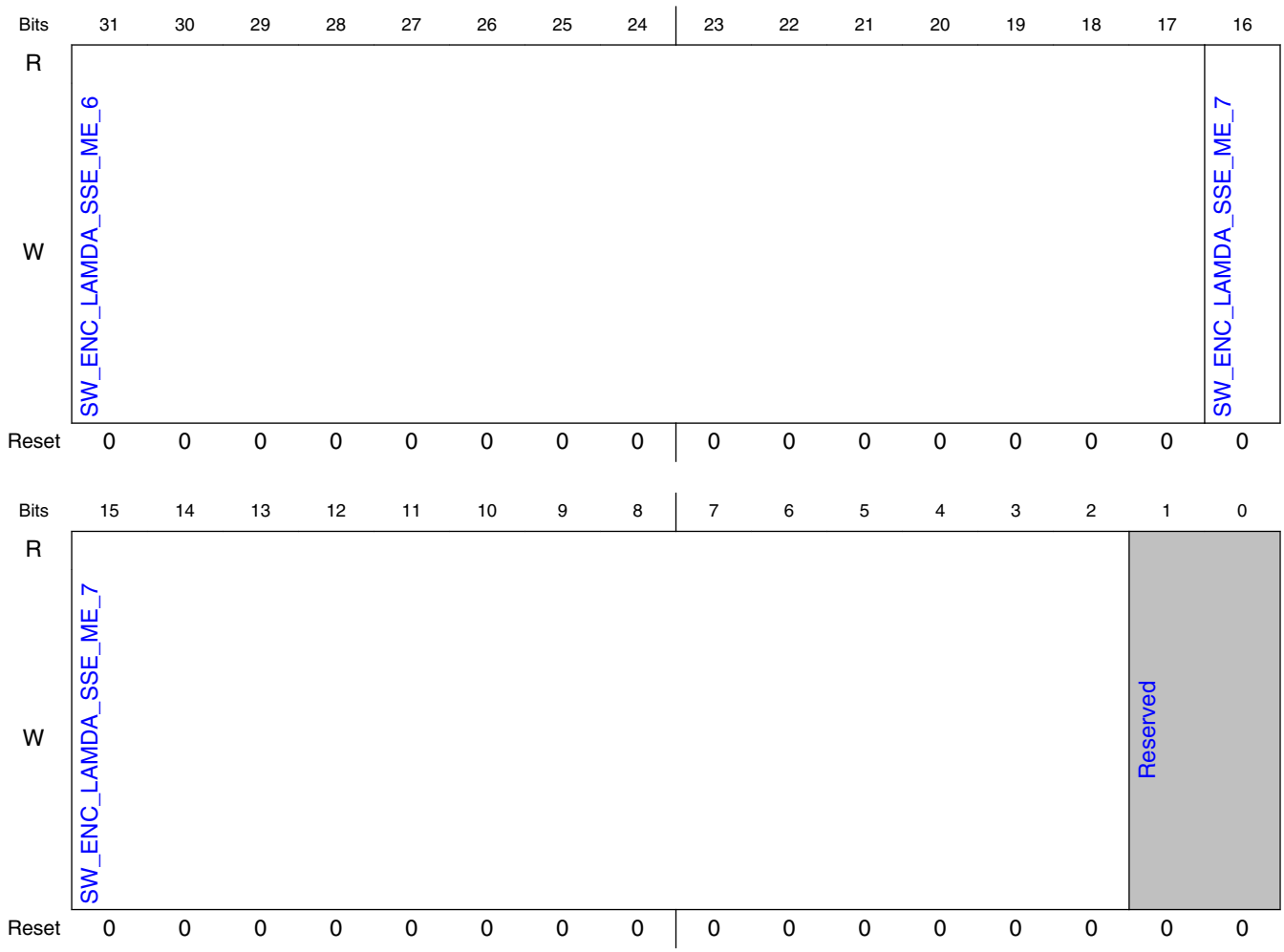
16.3.5.1.64.1 Offset

Register	Offset
swreg78_H2V2	138h

16.3.5.1.64.2 Function

inter me SSE lambda config 3. For H2V2 or later version.

16.3.5.1.64.3 Diagram



16.3.5.1.64.4 Fields

Field	Function
31-17 SW_ENC_LAMDA_SSE_ME_6	inter me SSE lambda 6
16-2 SW_ENC_LAMDA_SSE_ME_7	inter me SSE lambda 7
1-0 —	Reserved

16.3.5.1.65 inter me SATD lambda config 7. For H2V5 or later version. (swreg78_H2V5)

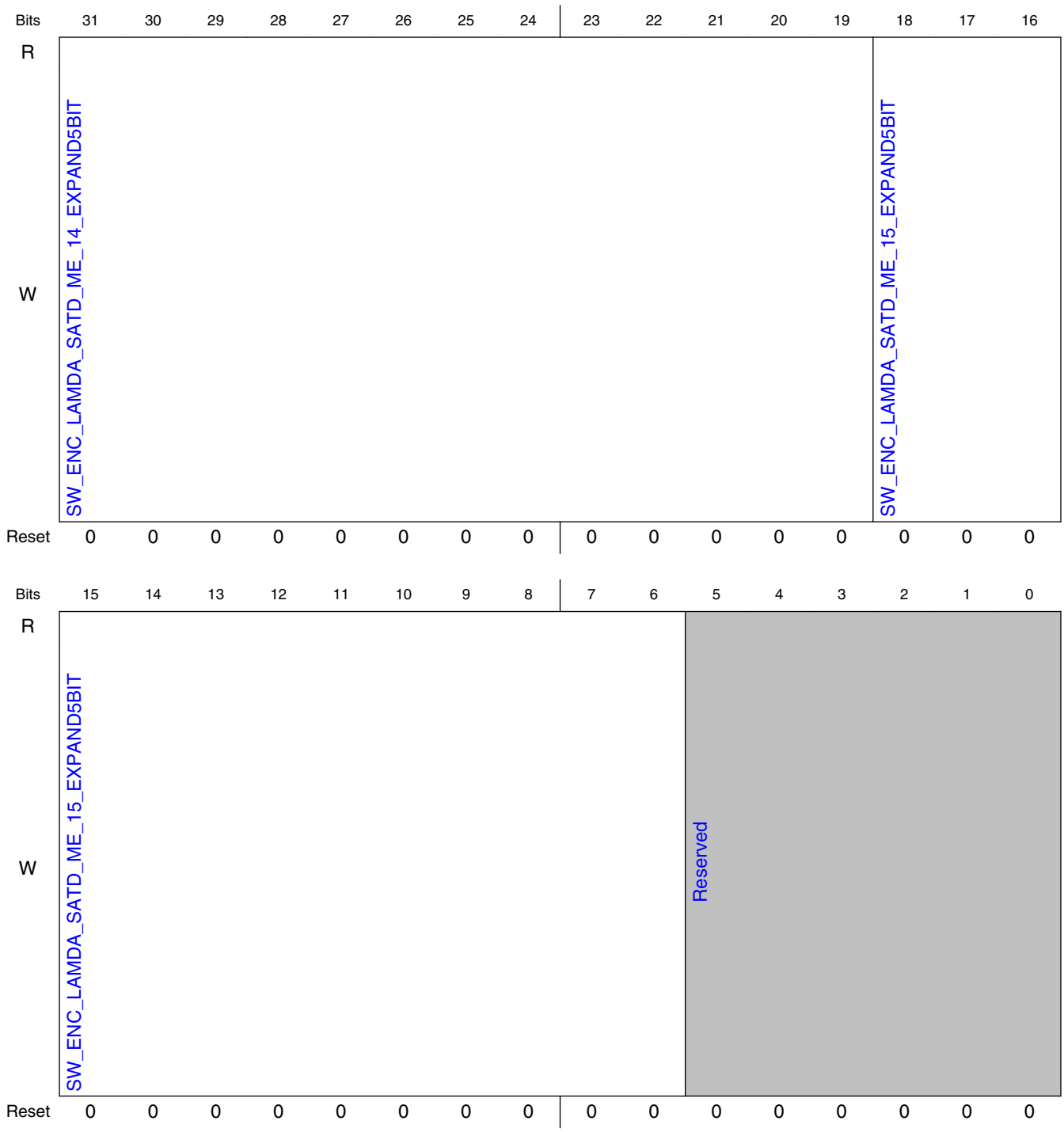
16.3.5.1.65.1 Offset

Register	Offset
swreg78_H2V5	138h

16.3.5.1.65.2 Function

inter me SATD lambda config 7. For H2V5 or later version.

16.3.5.1.65.3 Diagram



16.3.5.1.65.4 Fields

Field	Function
31-19	lambda satd me 14

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_14_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_15_EXPAND5BIT	lambda satd me 15
5-0 —	Reserved

16.3.5.1.66 inter prediction parameters4. For H2V1 only. (swreg79_H2V1)

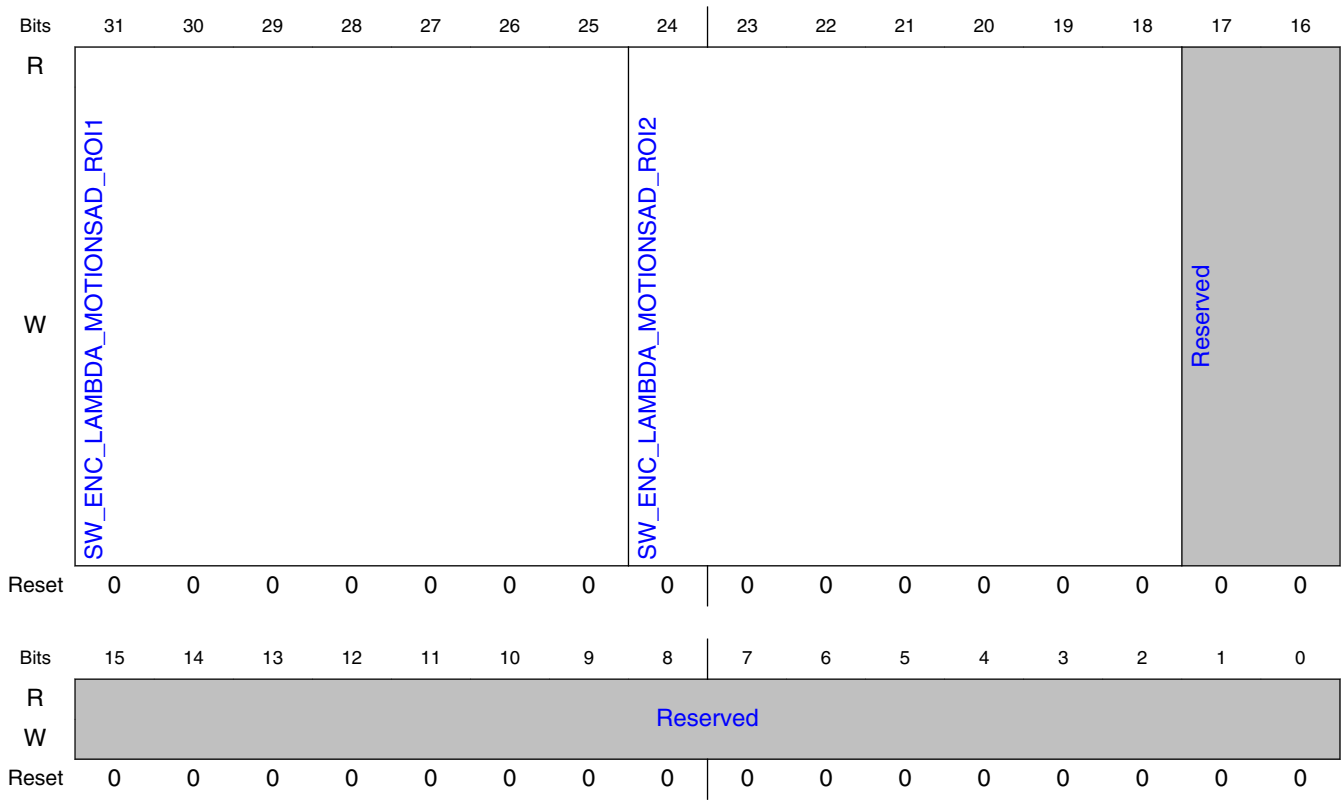
16.3.5.1.66.1 Offset

Register	Offset
swreg79_H2V1	13Ch

16.3.5.1.66.2 Function

inter prediction parameters4. For H2V1 only.

16.3.5.1.66.3 Diagram



16.3.5.1.66.4 Fields

Field	Function
31-25 SW_ENC_LAMBDA_MOTIONSAD_ROI1	roi1 lamda for motion SAD
24-18 SW_ENC_LAMBDA_MOTIONSAD_ROI2	roi2 lamda for motion SAD
17-0 —	Reserved

16.3.5.1.67 inter me SSE lambda config 4. For H2V2 or later version. (swreg79_H2V2)

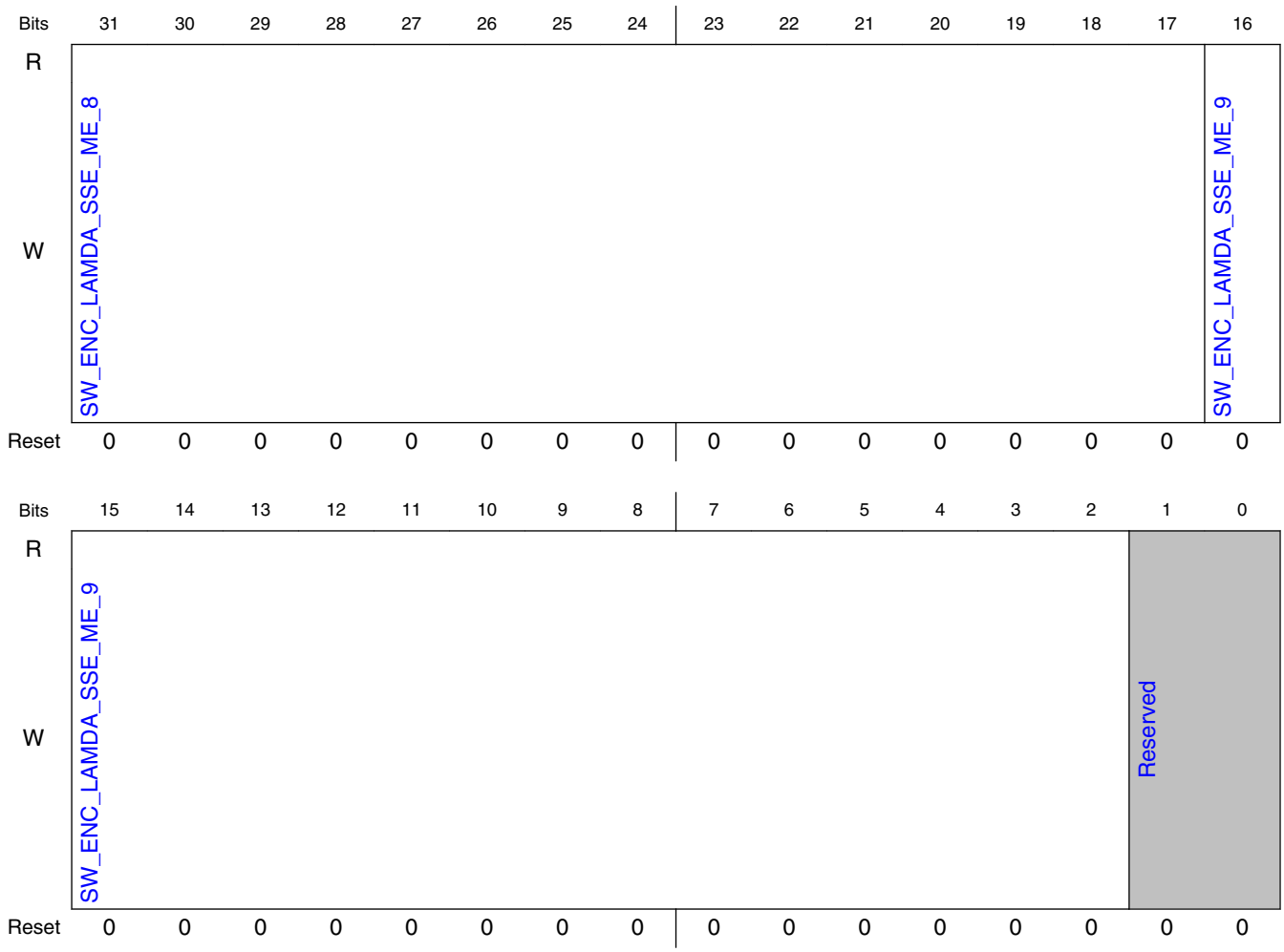
16.3.5.1.67.1 Offset

Register	Offset
swreg79_H2V2	13Ch

16.3.5.1.67.2 Function

inter me SSE lambda config 4. For H2V2 or later version.

16.3.5.1.67.3 Diagram



16.3.5.1.67.4 Fields

Field	Function
31-17	inter me SSE lambda 8

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SSE_ME_8	
16-2 SW_ENC_LAMDA_SSE_ME_9	inter me SSE lambda 9
1-0 —	Reserved

16.3.5.1.68 inter me SSE lambda config 0. For H2V5 or later version. (swreg79_H2V5)

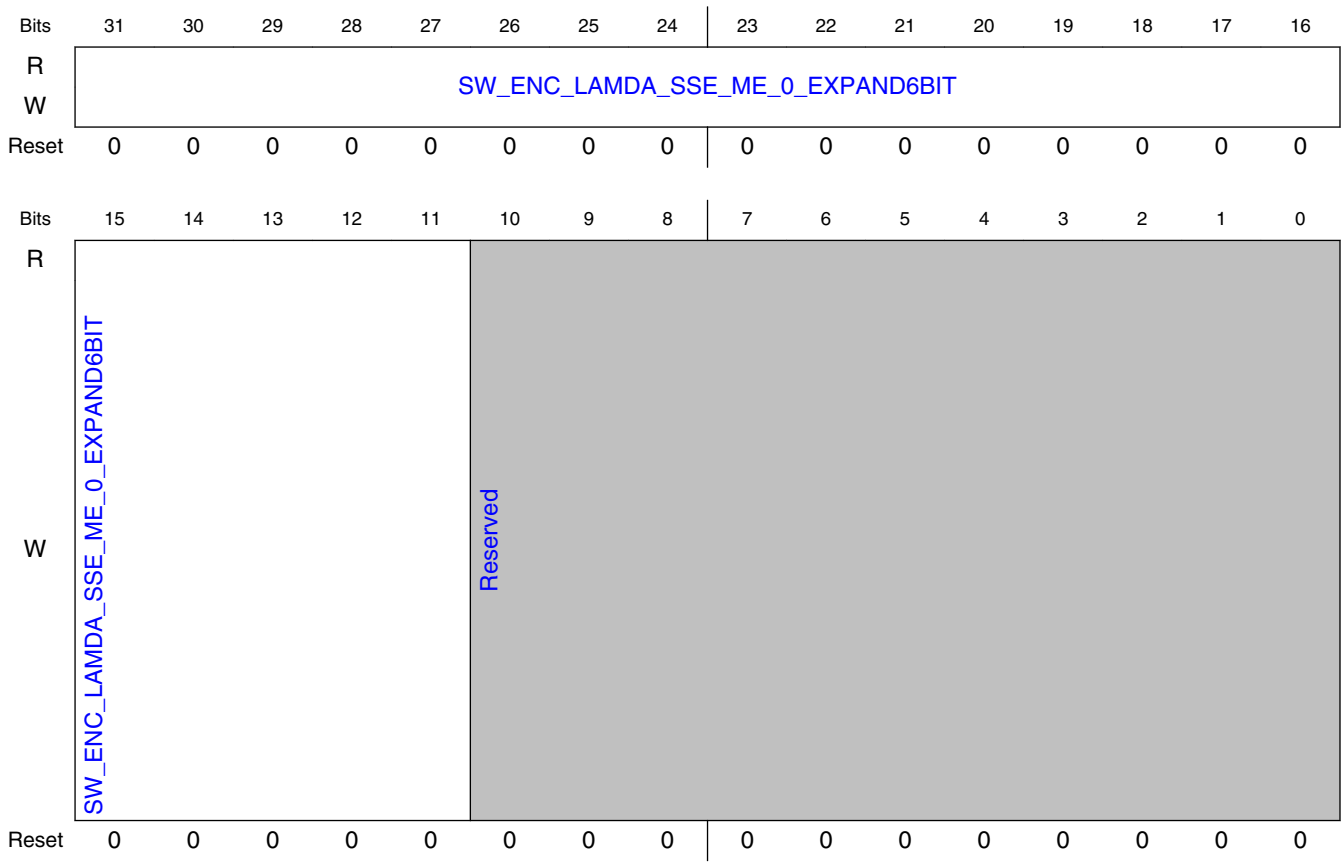
16.3.5.1.68.1 Offset

Register	Offset
swreg79_H2V5	13Ch

16.3.5.1.68.2 Function

inter me SSE lambda config 0. For H2V5 or later version.

16.3.5.1.68.3 Diagram



16.3.5.1.68.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_0_EXPAND6BIT	inter me SSE lambda 0
10-0 —	Reserved

16.3.5.1.69 HW synthesis config register, read-only (swreg80)

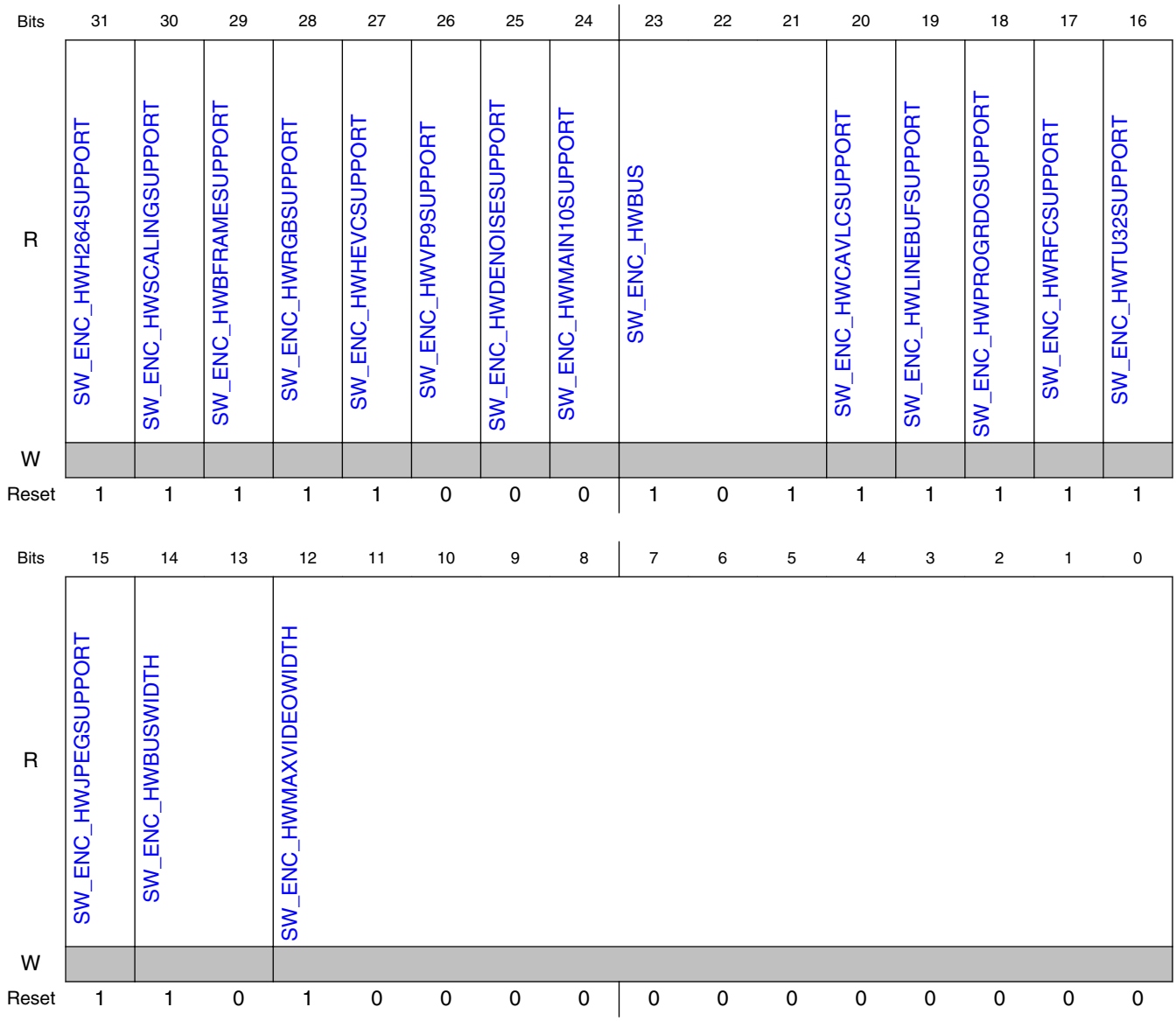
16.3.5.1.69.1 Offset

Register	Offset
swreg80	140h

16.3.5.1.69.2 Function

HW synthesis config register, read-only

16.3.5.1.69.3 Diagram



16.3.5.1.69.4 Fields

Field	Function
31	H264 encoding supported by HW. 0b - not supported.

Table continues on the next page...

Field	Function
SW_ENC_HWH264SUPPORT	1b - supported
30 SW_ENC_HWSCALINGSUPPORT	Down-scaling supported by HW. 0b - not supported. 1b - supported
29 SW_ENC_HWBFRAMESUPPORT	HW bframe support. 0b - not support bframe. 1b - support bframe
28 SW_ENC_HWRGBSUPPORT	RGB to YUV conversion supported by HW. 0b - not supported. 1b - supported
27 SW_ENC_HWHEVCSUPPORT	HEVC encoding supported by HW. 0b - not supported. 1b - supported
26 SW_ENC_HWVP9SUPPORT	VP9 encoding supported by HW. 0b - not supported. 1b - supported
25 SW_ENC_HWDENOISESUPPORT	denoise supported by HW. 0b - not supported. 1b - supported
24 SW_ENC_HWMAIN10SUPPORT	main10 supported by HW. 0b - main8 supported. 1b - main10 supported
23-21 SW_ENC_HWBUS	Bus connection of HW. 001b - AHB. 010b - OCP. 011b - AXI. 100b - PCI. 101b - AXIAHB. 110b - AXIAPB.
20 SW_ENC_HWCAVLCSUPPORT	CAVLC supported by HW. 0b - not supported. 1b - supported
19 SW_ENC_HWLINEBUFSUPPORT	LineBuffer input mode supported by HW. 0b - not supported. 1b - supported
18 SW_ENC_HWPROGRDOSUPPORT	Prog Rdo supported by HW. 0b - not supported. 1b - supported
17	Reference frame compression supported by HW. 0b - not supported.

Table continues on the next page...

Field	Function
SW_ENC_HWRFCSUPPORT	1b - supported
16 SW_ENC_HWTU32SUPPORT	TU32 supported by HW. 0b - not supported. 1b - supported
15 SW_ENC_HWJPEGSUPPORT	JPEG encoder supported by HW. 0b - not supported. 1b - supported
14-13 SW_ENC_HWBUSWIDTH	Bus width of HW. 00b - 32b. 01b - 64b. 10b - 128b
12-0 SW_ENC_HWMAXVIDEOWIDTH	Maximum video width supported by HW (pixels)

16.3.5.1.70 hardware configuation 0 (swreg81)

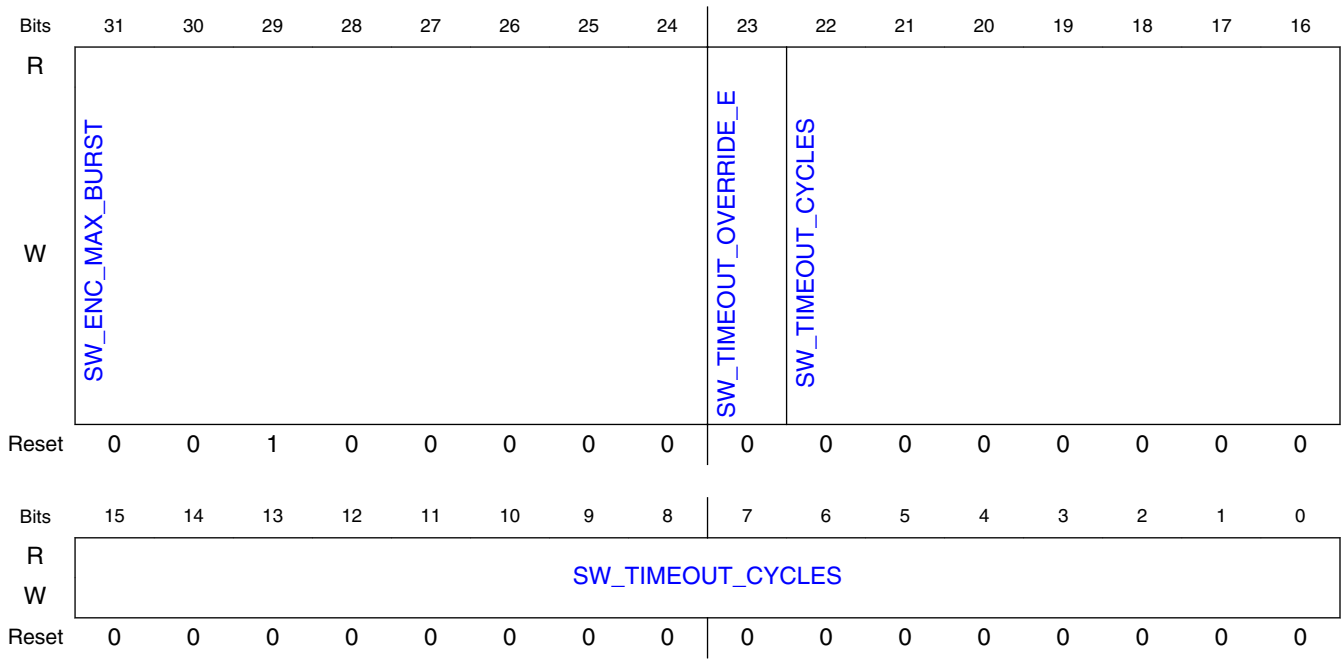
16.3.5.1.70.1 Offset

Register	Offset
swreg81	144h

16.3.5.1.70.2 Function

hardware configuation 0

16.3.5.1.70.3 Diagram



16.3.5.1.70.4 Fields

Field	Function
31-24 SW_ENC_MAX_BURST	for support AXI4.0 burst length is programmable Default value:0x20
23 SW_TIMEOUT_OVERRIDE_E	enable signal if timeout period is controlled by software Default value: 0x0
22-0 SW_TIMEOUT_CYCLES	timeout cycles number default value: 0x0

16.3.5.1.71 record hardware performance (swreg82)

16.3.5.1.71.1 Offset

Register	Offset
swreg82	148h

16.3.5.1.71.2 Function

record hardware performance

16.3.5.1.71.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_HW_PERFORMANCE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_HW_PERFORMANCE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.71.4 Fields

Field	Function
31-0 SW_ENC_HW_PERFORMANCE	record hardware performance(cycles) of current picture

16.3.5.1.72 reference picture reconstructed list1 luma0 (swreg83)

16.3.5.1.72.1 Offset

Register	Offset
swreg83	14Ch

16.3.5.1.72.2 Function

reference picture reconstructed list1 luma0

16.3.5.1.72.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REFPIC_RECON_L1_Y0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_REFPIC_RECON_L1_Y0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.72.4 Fields

Field	Function
31-0 SW_ENC_REFPIC_RECON_L1_Y0	reference picture reconstructed list1 luma0

16.3.5.1.73 reference picture reconstructed list1 chroma0 (swreg84)

16.3.5.1.73.1 Offset

Register	Offset
swreg84	150h

16.3.5.1.73.2 Function

reference picture reconstructed list1 chroma0

16.3.5.1.73.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REFPIC_RECON_L1_CHROMA0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_REFPIC_RECON_L1_CHROMA0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.73.4 Fields

Field	Function
31-0 SW_ENC_REFPIC_RECON_L1_CHROMA0	reference picture reconstructed list1 chroma0

16.3.5.1.74 reference pictures list1 config (swreg91)

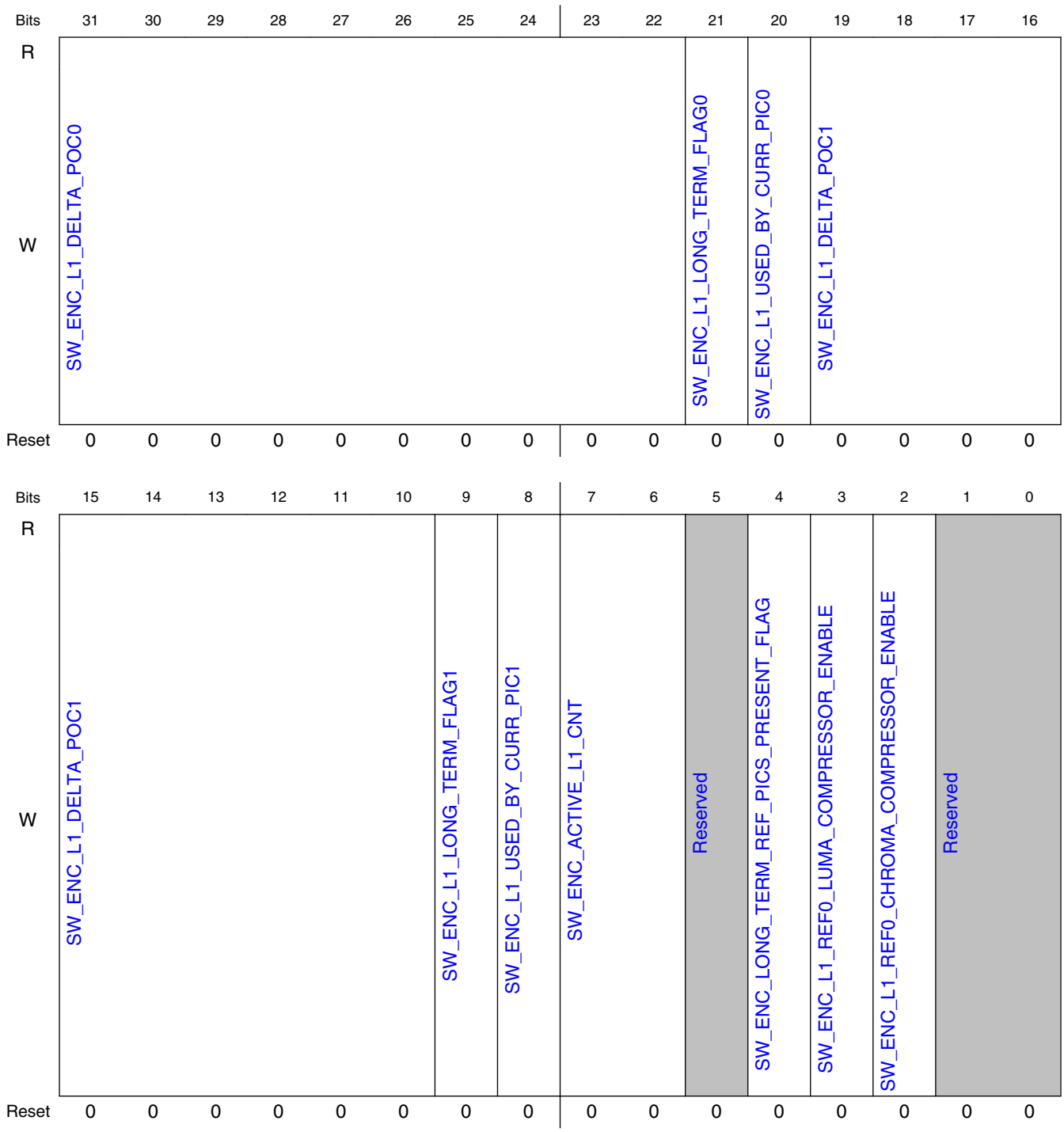
16.3.5.1.74.1 Offset

Register	Offset
swreg91	16Ch

16.3.5.1.74.2 Function

reference pictures list1 config

16.3.5.1.74.3 Diagram



16.3.5.1.74.4 Fields

Field	Function
31-22	delta poc of list1 pic0

Table continues on the next page...

Field	Function
SW_ENC_L1_DELTA_POC0	
21 SW_ENC_L1_LONG_TERM_FLAG0	list1 pic0 is long term
20 SW_ENC_L1_USED_BY_CURR_PIC0	list1 pic0 used by current
19-10 SW_ENC_L1_DELTA_POC1	delta poc of list1 pic1
9 SW_ENC_L1_LONG_TERM_FLAG1	list1 pic1 is long term
8 SW_ENC_L1_USED_BY_CURR_PIC1	list1 pic1 used by current
7-6 SW_ENC_ACTIVE_L1_CNT	active l0 count
5 —	Reserved
4 SW_ENC_LONG_TERM_REF_PICS_PRESENT_FLAG	long-term reference pictures may be used for inter prediction
3 SW_ENC_L1_REF0_LUMA_COMPRESSOR_ENABLE	list1 ref0 frame luma compressor enable flag 0b - disable 1b - enable.
2 SW_ENC_L1_REF0_CHROMA_COMPRESSOR_ENABLE	list1 ref0 frame chroma compressor enable flag 0b - disable 1b - enable.
1-0 —	Reserved

16.3.5.1.75 reference picture reconstructed list1 4n 0 (swreg92)

16.3.5.1.75.1 Offset

Register	Offset
swreg92	170h

16.3.5.1.75.2 Function

reference picture reconstructed list1 4n 0

16.3.5.1.75.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REFPIC_RECON_L1_4N0_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_REFPIC_RECON_L1_4N0_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.75.4 Fields

Field	Function
31-0 SW_ENC_REFPIC_RECON_L1_4N0_BASE	reference picture reconstructed list1 4n 0

16.3.5.1.76 Base address for list 1 ref 0 luma compress table LSB. (swreg96)

16.3.5.1.76.1 Offset

Register	Offset
swreg96	180h

16.3.5.1.76.2 Function

Base address for list 1 ref 0 luma compress table LSB.

16.3.5.1.76.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_L1_REF0_LUMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_L1_REF0_LUMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.76.4 Fields

Field	Function
31-0 SW_ENC_L1_REF0_LUMA_COMPRESS_TABLE_BASE	Base address for list 1 ref 0 luma compress table LSB.

16.3.5.1.77 Base address for list 1 ref 0 Chroma compress table LSB. (swreg98)

16.3.5.1.77.1 Offset

Register	Offset
swreg98	188h

16.3.5.1.77.2 Function

Base address for list 1 ref 0 Chroma compress table LSB.

16.3.5.1.77.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_L1_REF0_CHROMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_L1_REF0_CHROMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.77.4 Fields

Field	Function
31-0 SW_ENC_L1_REF0_CHROMA_COMPRESS_TABLE_BASE	Base address for list 1 ref 0 Chroma compress table LSB.

16.3.5.1.78 Min picture size (swreg106)

16.3.5.1.78.1 Offset

Register	Offset
swreg106	1A8h

16.3.5.1.78.2 Function

Min picture size

16.3.5.1.78.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MINPICSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MINPICSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.78.4 Fields

Field	Function
31-0 SW_ENC_MINPICSIZE	Allowed minimum picture size for CTB rate control.

16.3.5.1.79 Max picture size (swreg107)

16.3.5.1.79.1 Offset

Register	Offset
swreg107	1ACh

16.3.5.1.79.2 Function

Max picture size

16.3.5.1.79.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MAXPICSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MAXPICSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.79.4 Fields

Field	Function
31-0 SW_ENC_MAX PICSIZE	Allowed minimum picture size for CTB rate control.

16.3.5.1.80 Qp delta map (swreg109)

16.3.5.1.80.1 Offset

Register	Offset
swreg109	1B4h

16.3.5.1.80.2 Function

Qp delta map

16.3.5.1.80.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROIMAPDELTAQPADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROIMAPDELTAQPADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.80.4 Fields

Field	Function
31-0 SW_ENC_ROIMAPDELTAQPADDR	Qp delta map

16.3.5.1.81 adaptive GOP configuration1 (swreg111)

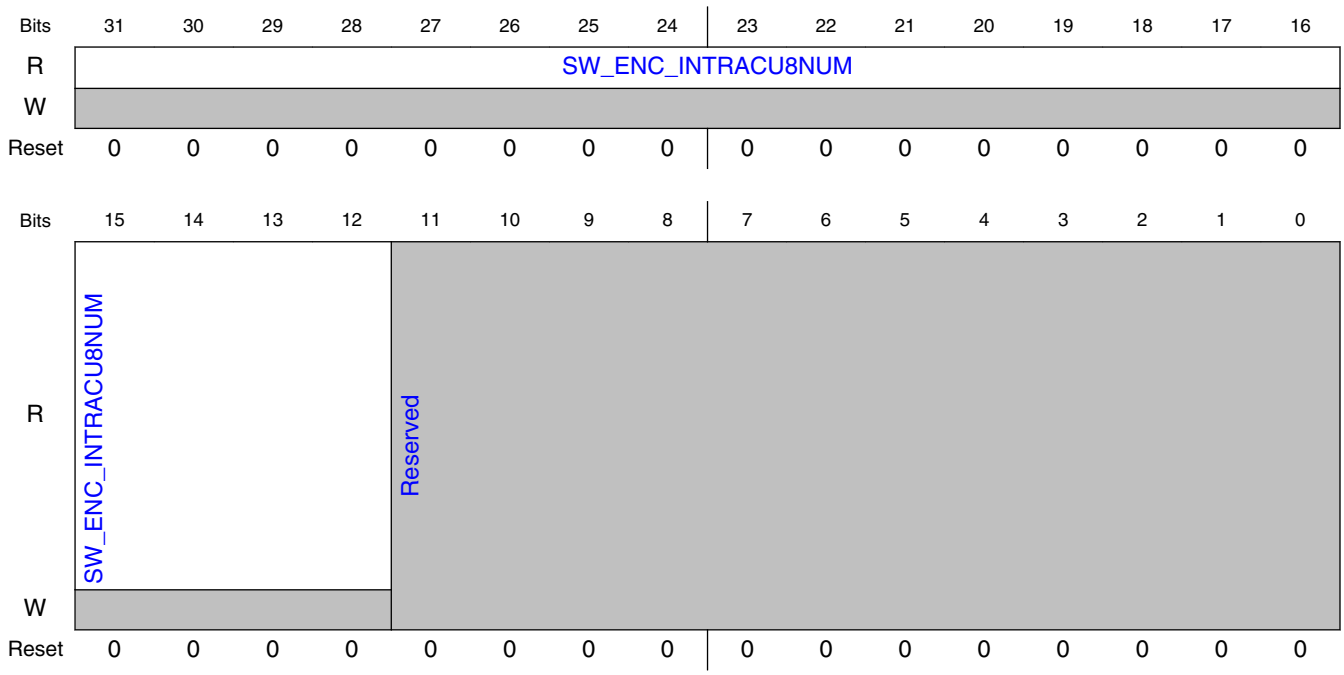
16.3.5.1.81.1 Offset

Register	Offset
swreg111	1BCh

16.3.5.1.81.2 Function

adaptive GOP configuration1

16.3.5.1.81.3 Diagram



16.3.5.1.81.4 Fields

Field	Function
31-12 SW_ENC_INTRACU8NUM	The number of block8x8 with type INTRA.
11-0 —	Reserved

16.3.5.1.82 adaptive GOP configuration2 (swreg112)

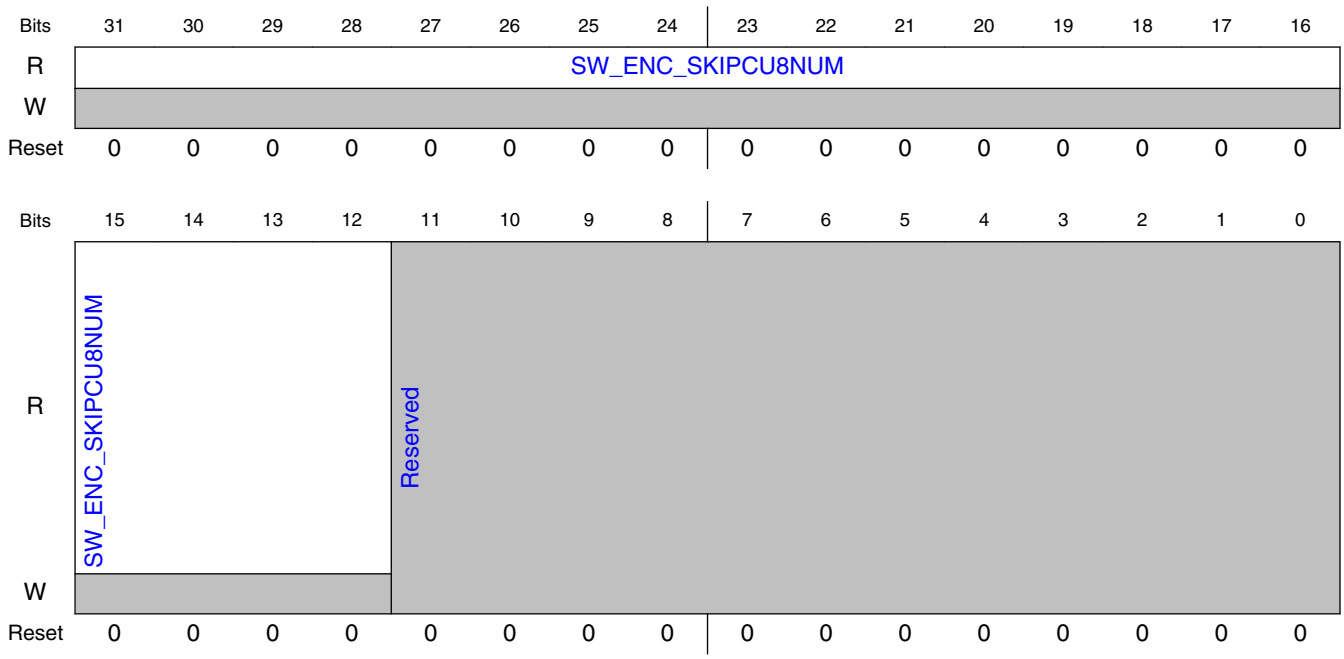
16.3.5.1.82.1 Offset

Register	Offset
swreg112	1C0h

16.3.5.1.82.2 Function

adaptive GOP configuration2

16.3.5.1.82.3 Diagram



16.3.5.1.82.4 Fields

Field	Function
31-12 SW_ENC_SKIPCU8NUM	The number of block8x8 with type SKIP.
11-0 —	Reserved

16.3.5.1.83 adaptive GOP configuration3 (swreg113)

16.3.5.1.83.1 Offset

Register	Offset
swreg113	1C4h

16.3.5.1.83.2 Function

adaptive GOP configuration3

16.3.5.1.83.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_PBFRAME4NRDCOST															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_PBFRAME4NRDCOST															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.83.4 Fields

Field	Function
31-0 SW_ENC_PBFRAME4NRDCOST	PBFrame4NRdCost

16.3.5.1.84 ctb rate control bit memory address of current frame (swreg114)

16.3.5.1.84.1 Offset

Register	Offset
swreg114	1C8h

16.3.5.1.84.2 Function

ctb rate control bit memory address of current frame

16.3.5.1.84.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_COLCTBS_STORE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_COLCTBS_STORE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.84.4 Fields

Field	Function
31-0 SW_ENC_COLCTBS_STORE_BASE	H.264 collocated mb memory address of current frame (for store)

16.3.5.1.85 ctb rate control bit memory address of previous frame (swreg116)

16.3.5.1.85.1 Offset

Register	Offset
swreg116	1D0h

16.3.5.1.85.2 Function

ctb rate control bit memory address of previous frame

16.3.5.1.85.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_COLCTBS_LOAD_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_COLCTBS_LOAD_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.85.4 Fields

Field	Function
31-0 SW_ENC_COLCTBS_LOAD_BASE	H.264 collocated mb memory address of reference frame (for load)

16.3.5.1.86 min/max lcu bits number of last picture (swreg119)

16.3.5.1.86.1 Offset

Register	Offset
swreg119	1DCh

16.3.5.1.86.2 Function

min/max lcu bits number of last picture

16.3.5.1.86.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_CTBBITSMIN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_CTBBITSMAX															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.86.4 Fields

Field	Function
31-16 SW_ENC_CTB BITSMIN	minimum lcu bits number of last picture
15-0 SW_ENC_CTB BITSMAX	maximum lcu bits number of last picture

16.3.5.1.87 total bits number of all lcus of last picture not including slice header bits (swreg120)

16.3.5.1.87.1 Offset

Register	Offset
swreg120	1E0h

16.3.5.1.87.2 Function

total bits number of all lcus of last picture not including slice header bits

16.3.5.1.87.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_TOTALLCUBITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_TOTALLCUBITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.87.4 Fields

Field	Function
31-0 SW_ENC_TOTALLCUBITS	total bits number of all lcus of last picture not including slice header bits

16.3.5.1.88 inter me SSE lambda config 5 (swreg122)

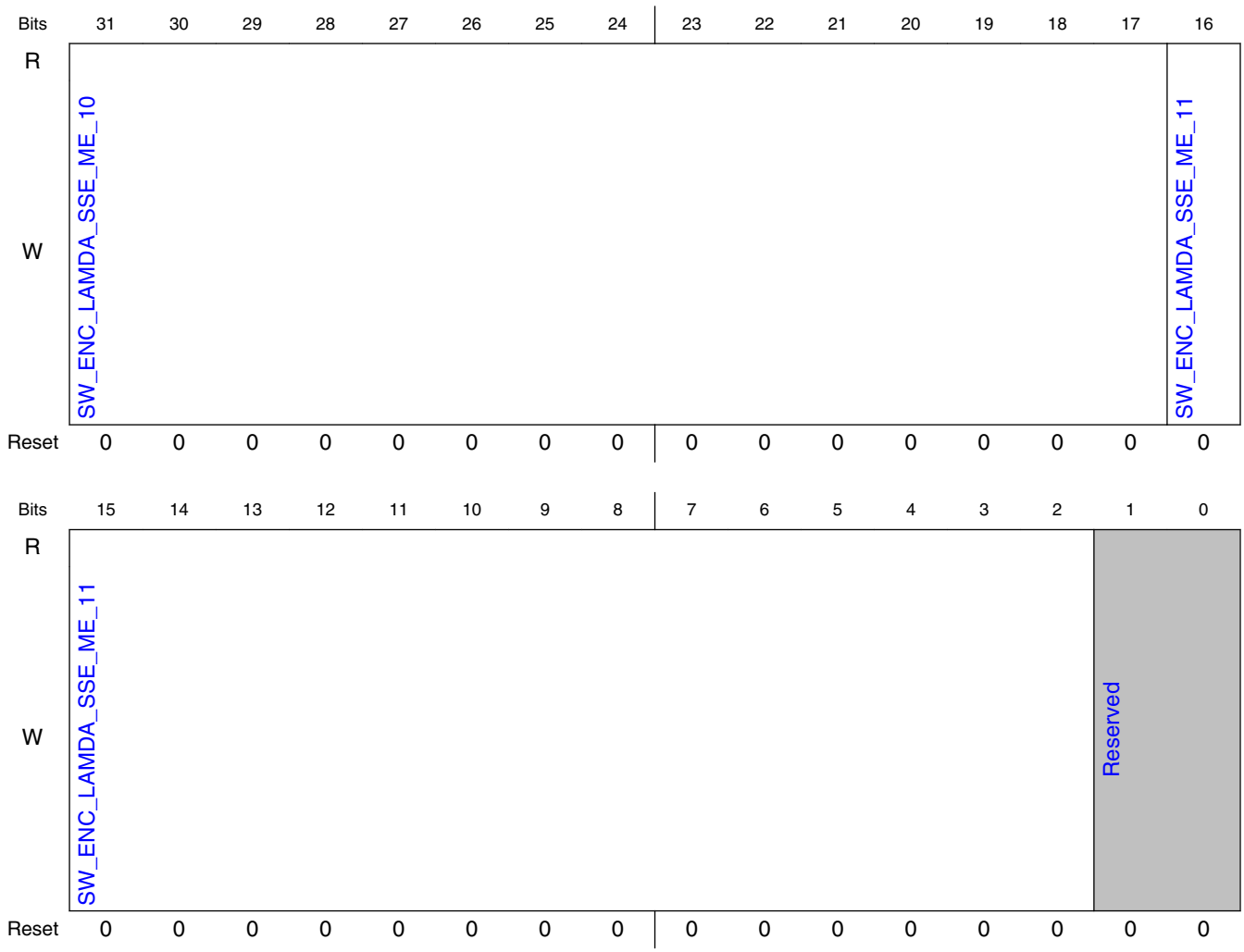
16.3.5.1.88.1 Offset

Register	Offset
swreg122	1E8h

16.3.5.1.88.2 Function

inter me SSE lambda config 5

16.3.5.1.88.3 Diagram



16.3.5.1.88.4 Fields

Field	Function
31-17 SW_ENC_LAMDA_SSE_ME_10	inter me SSE lambda 10
16-2 SW_ENC_LAMDA_SSE_ME_11	inter me SSE lambda 11
1-0 —	Reserved

16.3.5.1.89 inter me SSE lambda config 1. For H2V5 or later version. (swreg122_H2V5)

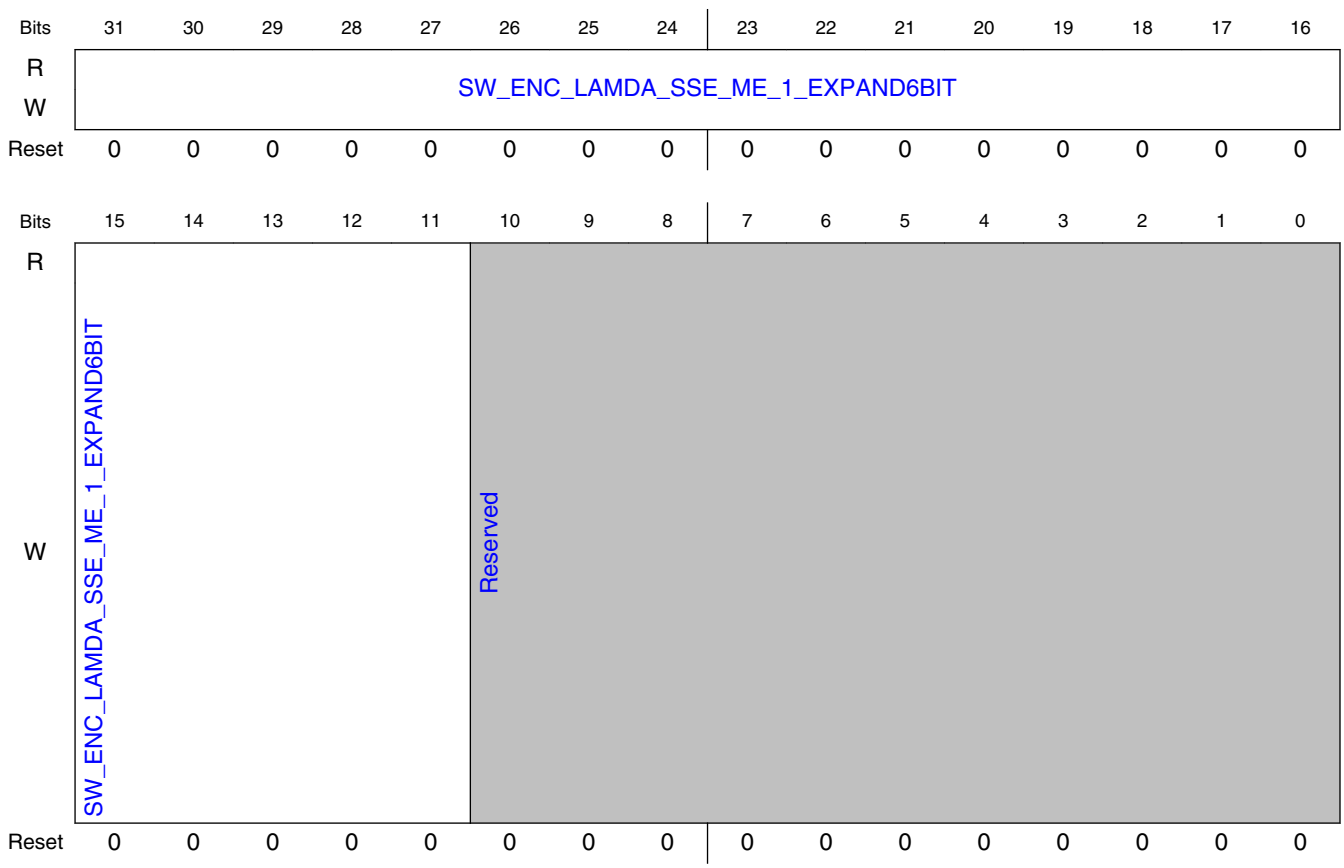
16.3.5.1.89.1 Offset

Register	Offset
swreg122_H2V5	1E8h

16.3.5.1.89.2 Function

inter me SSE lambda config 1. For H2V5 or later version.

16.3.5.1.89.3 Diagram



16.3.5.1.89.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_1_EXPAND6BIT	inter me SSE lambda 1
10-0 —	Reserved

16.3.5.1.90 inter me SSE lambda config 6 (swreg123)

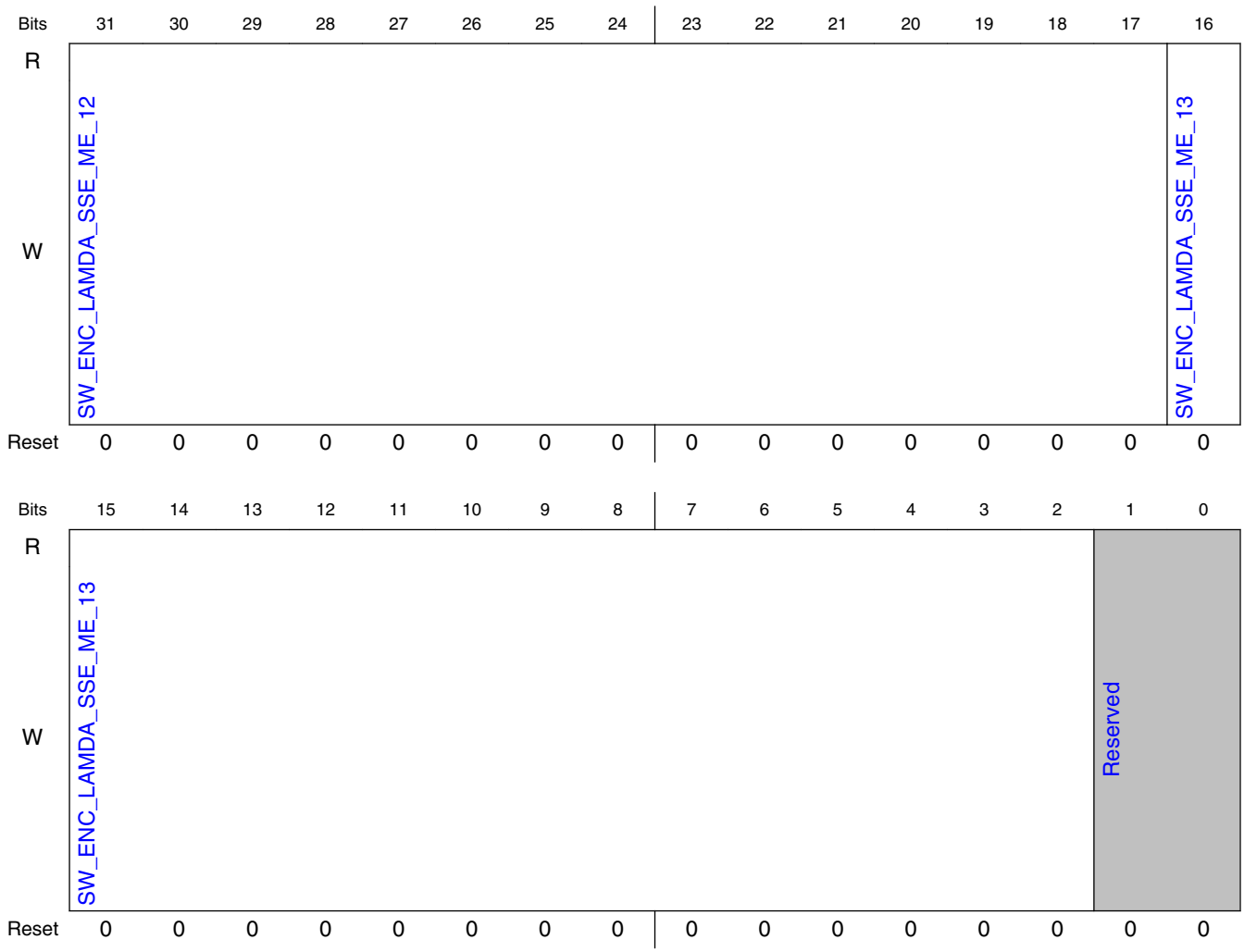
16.3.5.1.90.1 Offset

Register	Offset
swreg123	1ECh

16.3.5.1.90.2 Function

inter me SSE lambda config 6

16.3.5.1.90.3 Diagram



16.3.5.1.90.4 Fields

Field	Function
31-17 SW_ENC_LAMDA_SSE_ME_12	inter me SSE lambda 12
16-2 SW_ENC_LAMDA_SSE_ME_13	inter me SSE lambda 13
1-0 —	Reserved

16.3.5.1.91 inter me SSE lambda config 2. For H2V5 or later version. (swreg123_H2V5)

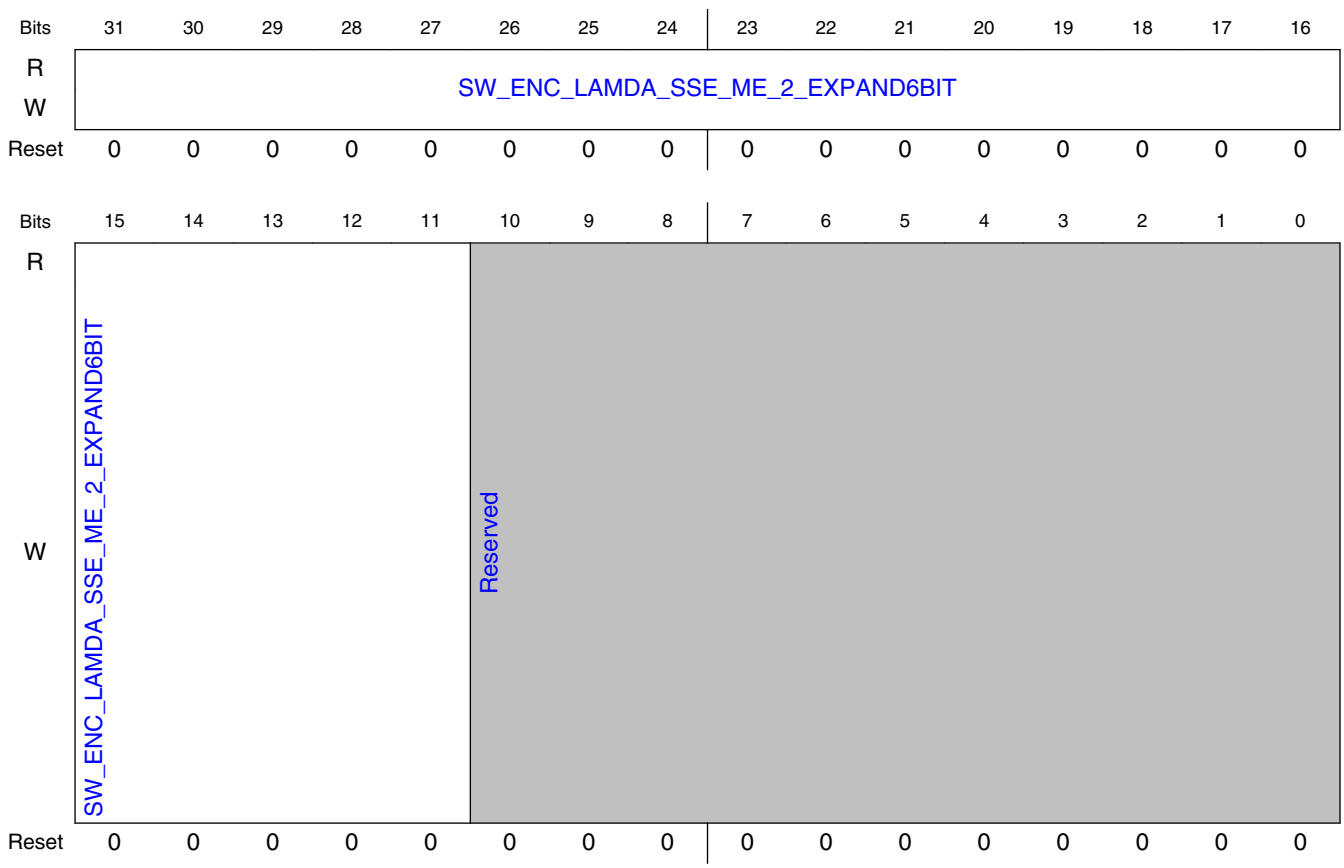
16.3.5.1.91.1 Offset

Register	Offset
swreg123_H2V5	1ECh

16.3.5.1.91.2 Function

inter me SSE lambda config 2. For H2V5 or later version.

16.3.5.1.91.3 Diagram



16.3.5.1.91.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_2_EXPAND6BIT	inter me SSE lambda 2
10-0 —	Reserved

16.3.5.1.92 inter me SSE lambda config 7 (swreg124)

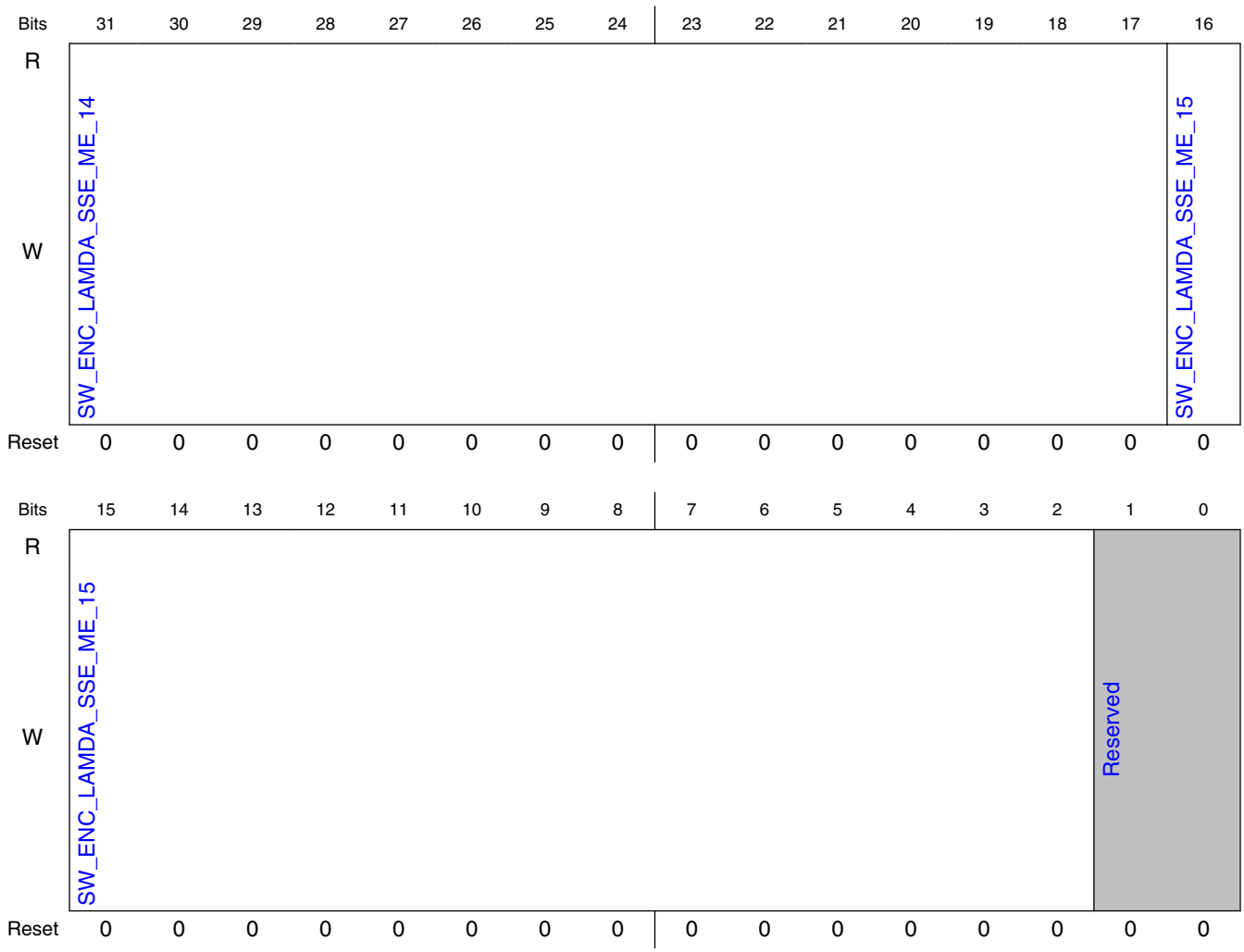
16.3.5.1.92.1 Offset

Register	Offset
swreg124	1F0h

16.3.5.1.92.2 Function

inter me SSE lambda config 7

16.3.5.1.92.3 Diagram



16.3.5.1.92.4 Fields

Field	Function
31-17 SW_ENC_LAMDA_SSE_ME_14	inter me SSE lambda 14
16-2 SW_ENC_LAMDA_SSE_ME_15	inter me SSE lambda 15
1-0 —	Reserved

16.3.5.1.93 inter me SSE lambda config 3. For H2V5 or later version. (swreg124_H2V5)

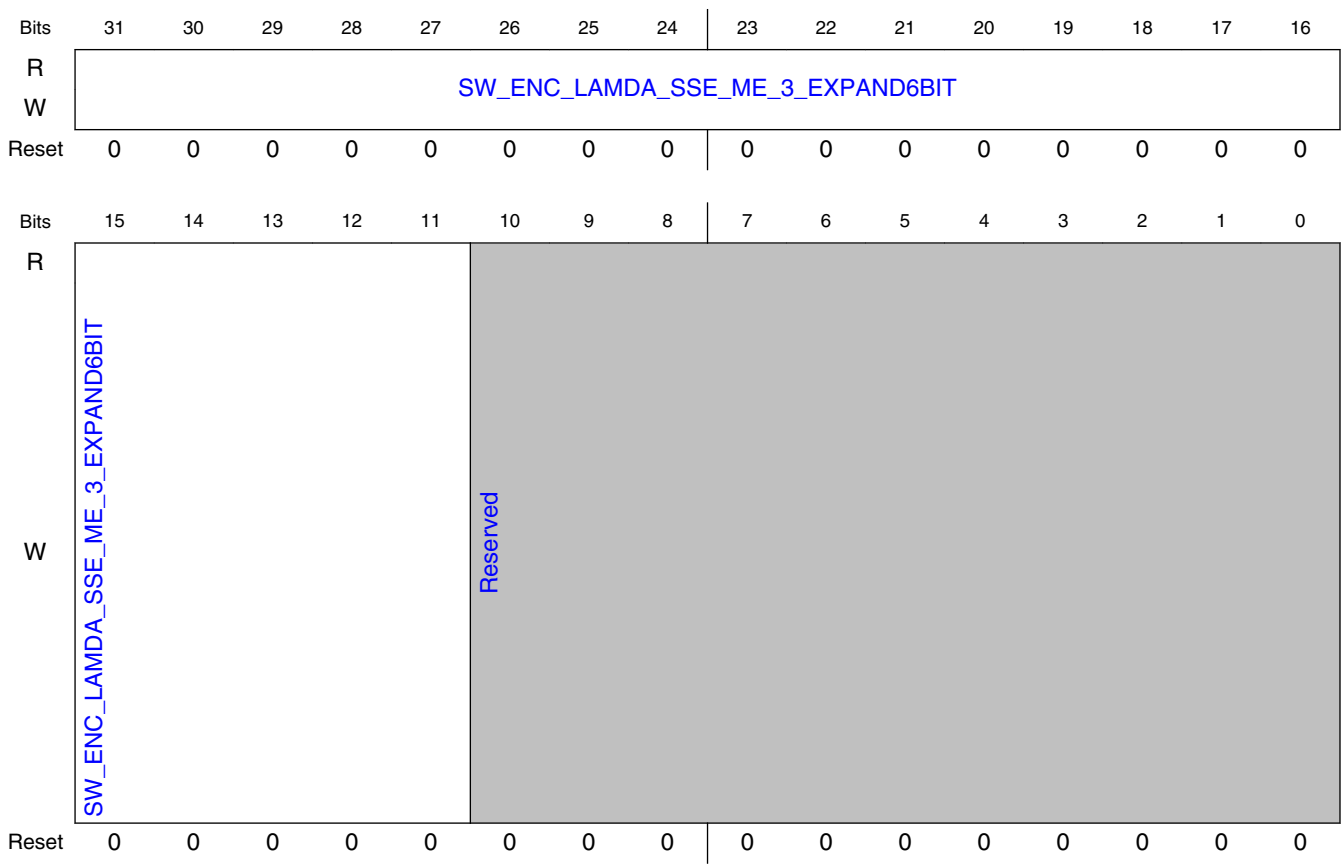
16.3.5.1.93.1 Offset

Register	Offset
swreg124_H2V5	1F0h

16.3.5.1.93.2 Function

inter me SSE lambda config 3. For H2V5 or later version.

16.3.5.1.93.3 Diagram



16.3.5.1.93.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_3_EXPAND6BIT	inter me SSE lambda 3
10-0 —	Reserved

16.3.5.1.94 intra SATD lambda config 0 (swreg125)

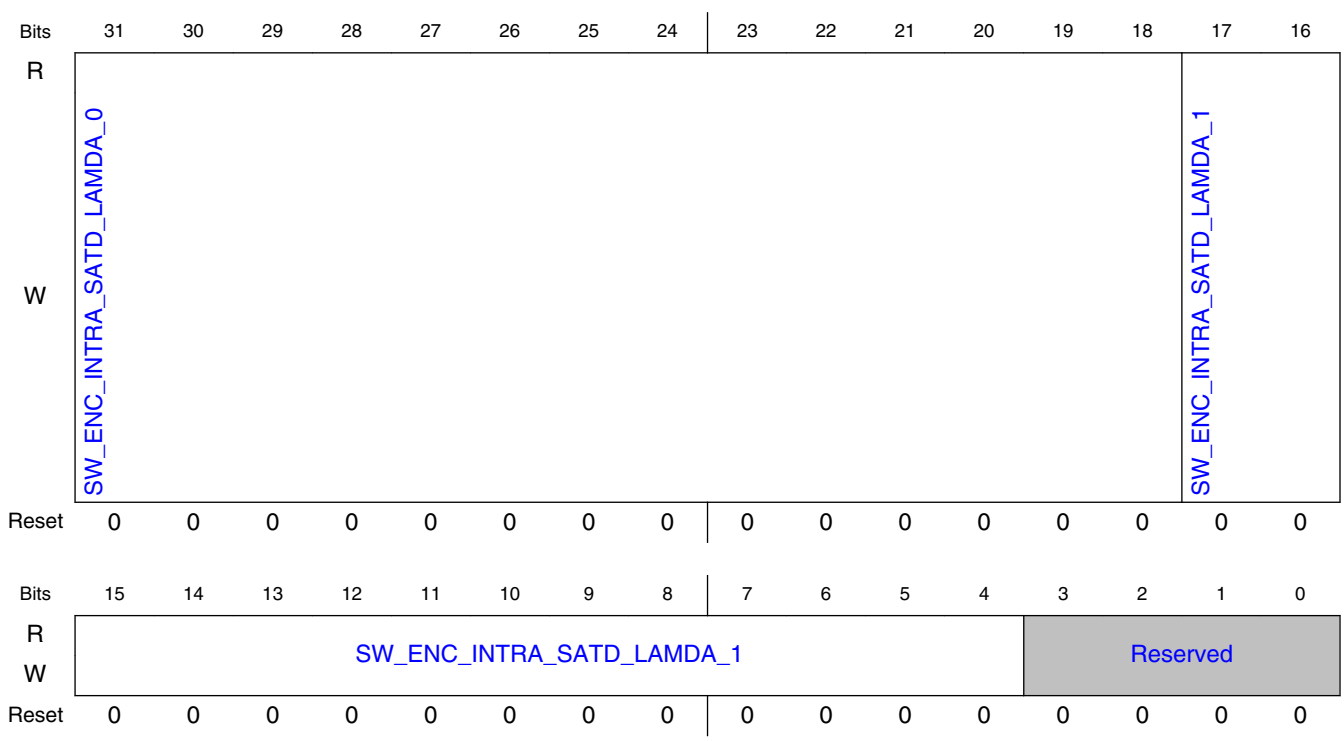
16.3.5.1.94.1 Offset

Register	Offset
swreg125	1F4h

16.3.5.1.94.2 Function

intra SATD lambda config 0

16.3.5.1.94.3 Diagram



16.3.5.1.94.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_0	intra SATD lambda 0
17-4 SW_ENC_INTRA_SATD_LAMDA_1	intra SATD lambda 1
3-0 —	Reserved

16.3.5.1.95 intra SATD lambda config 1 (swreg126)

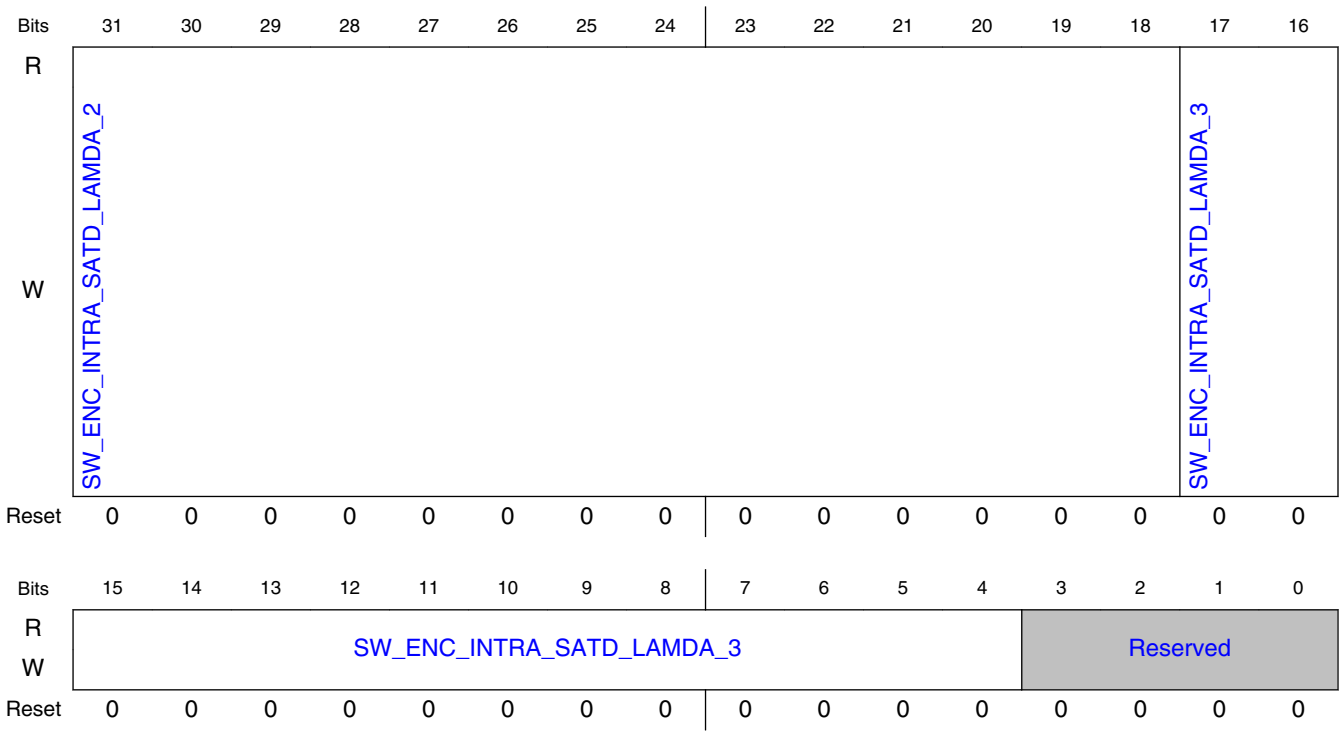
16.3.5.1.95.1 Offset

Register	Offset
swreg126	1F8h

16.3.5.1.95.2 Function

intra SATD lambda config 1

16.3.5.1.95.3 Diagram



16.3.5.1.95.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_2	intra SATD lambda 2
17-4 SW_ENC_INTRA_SATD_LAMDA_3	intra SATD lambda 3
3-0 —	Reserved

16.3.5.1.96 intra SATD lambda config 2 (swreg127)

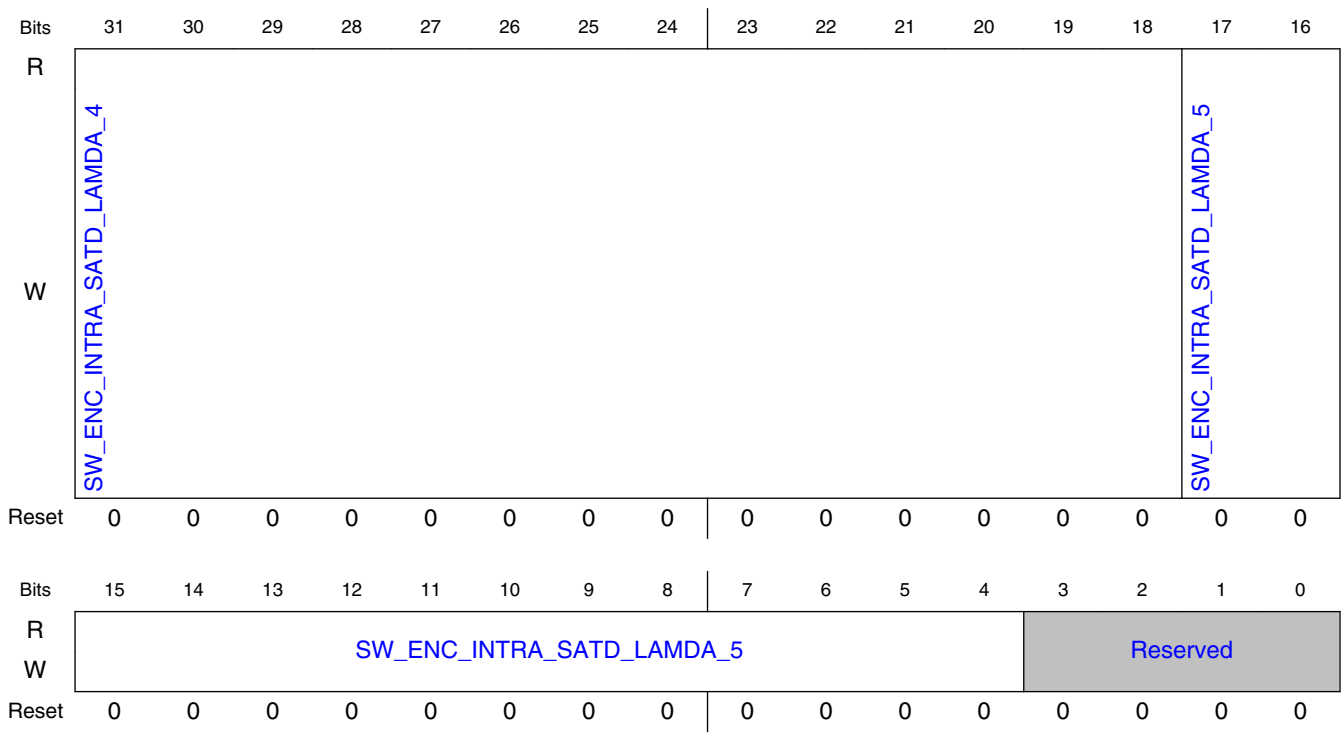
16.3.5.1.96.1 Offset

Register	Offset
swreg127	1FCh

16.3.5.1.96.2 Function

intra SATD lambda config 2

16.3.5.1.96.3 Diagram



16.3.5.1.96.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_4	intra SATD lambda 4
17-4 SW_ENC_INTRA_SATD_LAMDA_5	intra SATD lambda 5
3-0 —	Reserved

16.3.5.1.97 intra SATD lambda config 3 (swreg128)

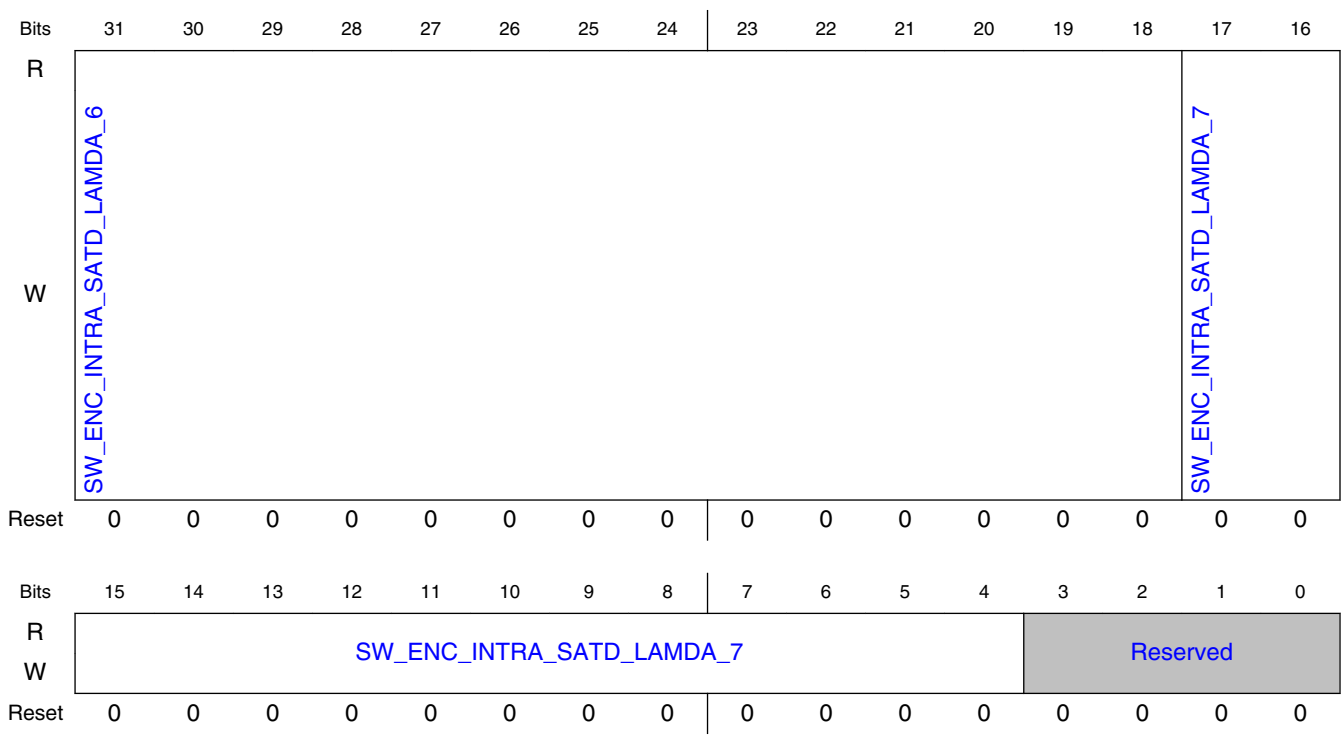
16.3.5.1.97.1 Offset

Register	Offset
swreg128	200h

16.3.5.1.97.2 Function

intra SATD lambda config 3

16.3.5.1.97.3 Diagram



16.3.5.1.97.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_6	intra SATD lambda 6
17-4	intra SATD lambda 7

Table continues on the next page...

Field	Function
SW_ENC_INTRA_SATD_LAMDA_7	
3-0 —	Reserved

16.3.5.1.98 intra SATD lambda config 4 (swreg129)

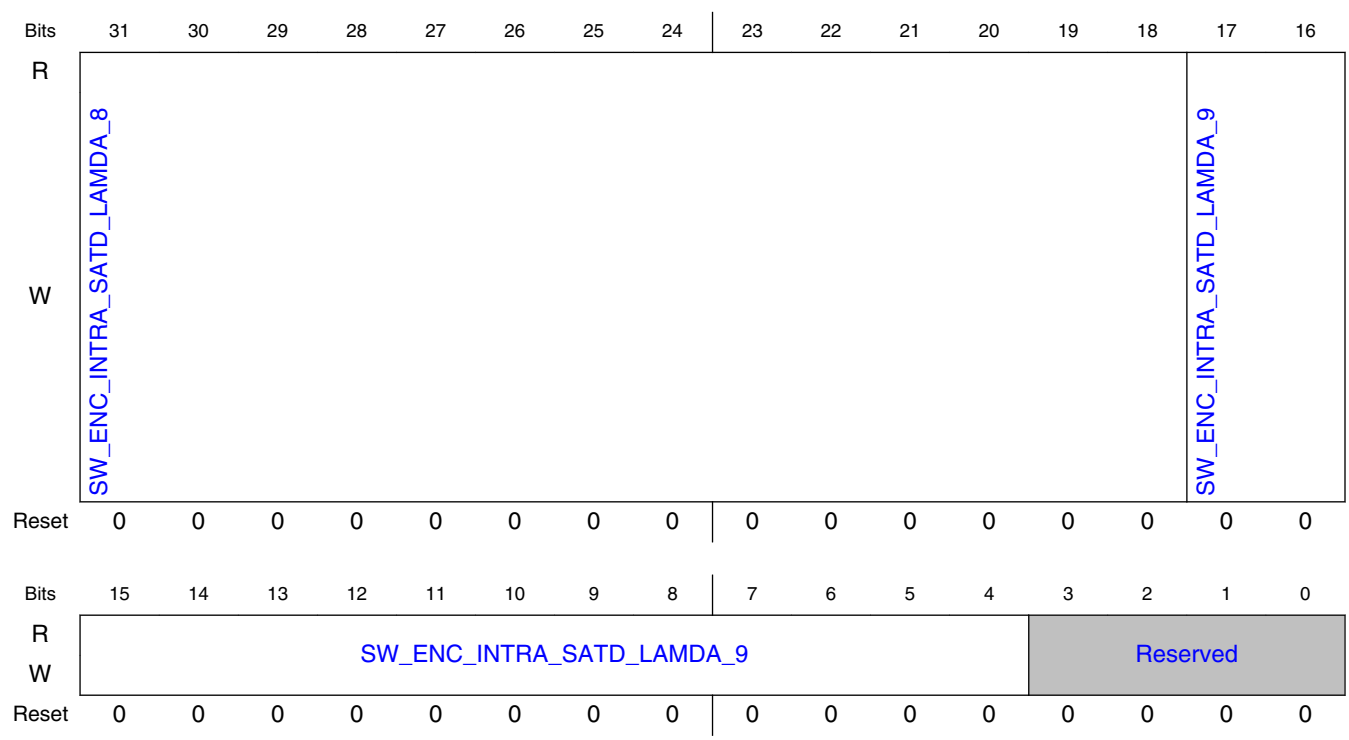
16.3.5.1.98.1 Offset

Register	Offset
swreg129	204h

16.3.5.1.98.2 Function

intra SATD lambda config 4

16.3.5.1.98.3 Diagram



16.3.5.1.98.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_8	intra SATD lambda 8
17-4 SW_ENC_INTRA_SATD_LAMDA_9	intra SATD lambda 9
3-0 —	Reserved

16.3.5.1.99 intra SATD lambda config 5 (swreg130)

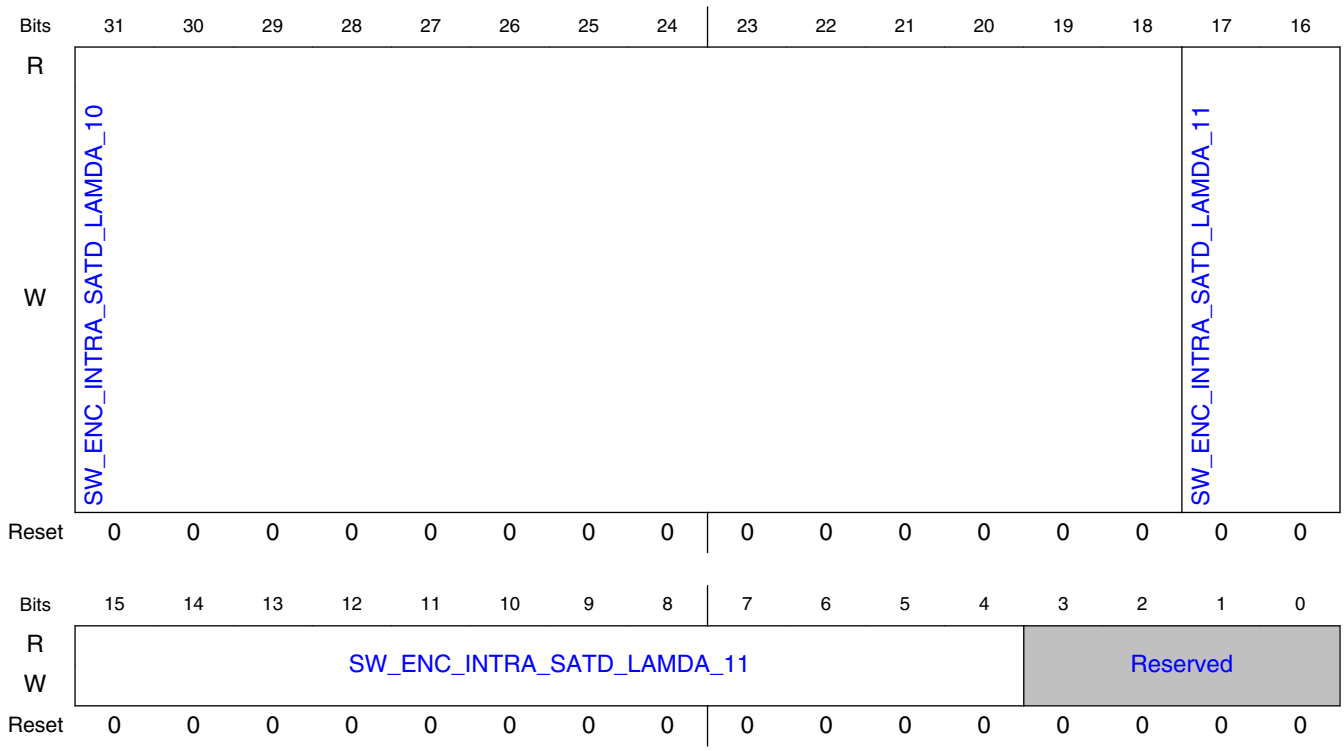
16.3.5.1.99.1 Offset

Register	Offset
swreg130	208h

16.3.5.1.99.2 Function

intra SATD lambda config 5

16.3.5.1.99.3 Diagram



16.3.5.1.99.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_10	intra SATD lambda 10
17-4 SW_ENC_INTRA_SATD_LAMDA_11	intra SATD lambda 11
3-0 —	Reserved

16.3.5.1.100 intra SATD lambda config 6 (swreg131)

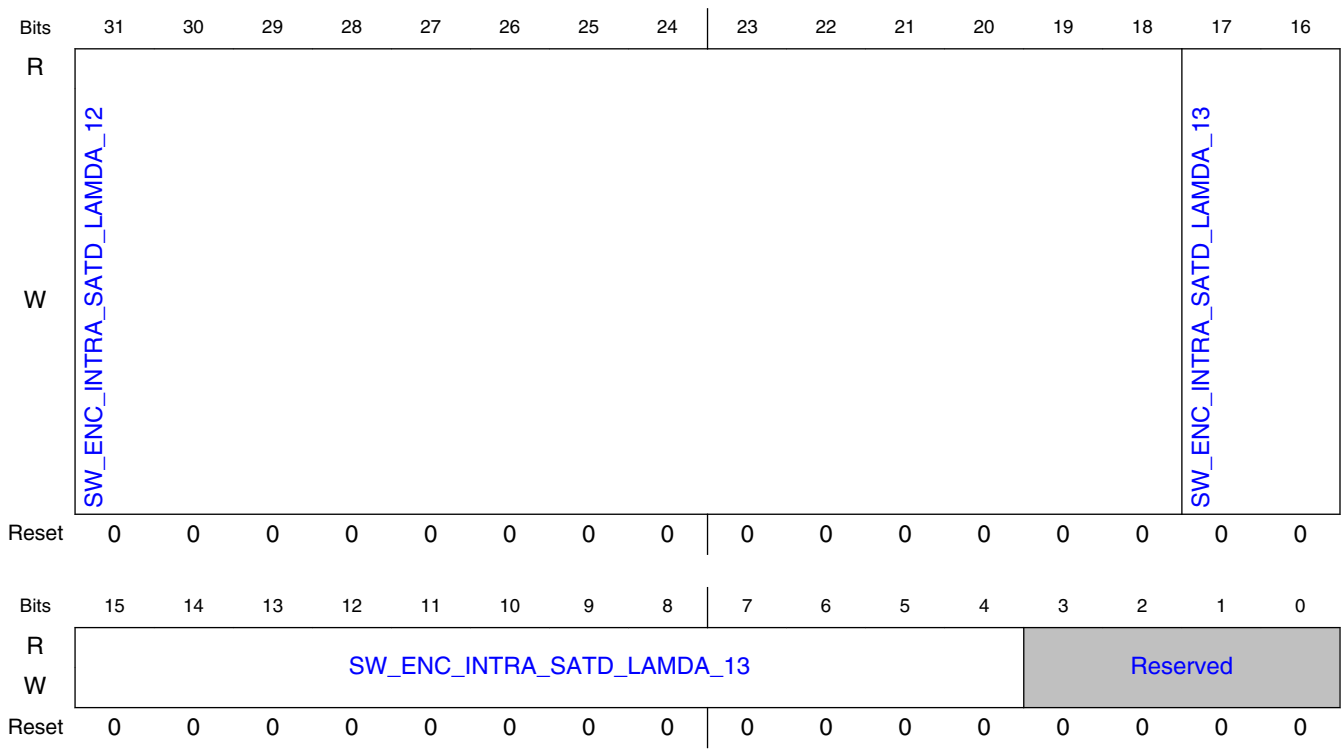
16.3.5.1.100.1 Offset

Register	Offset
swreg131	20Ch

16.3.5.1.100.2 Function

intra SATD lambda config 6

16.3.5.1.100.3 Diagram



16.3.5.1.100.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_12	intra SATD lambda 12
17-4 SW_ENC_INTRA_SATD_LAMDA_13	intra SATD lambda 13
3-0 —	Reserved

16.3.5.1.101 intra SATD lambda config 7 (swreg132)

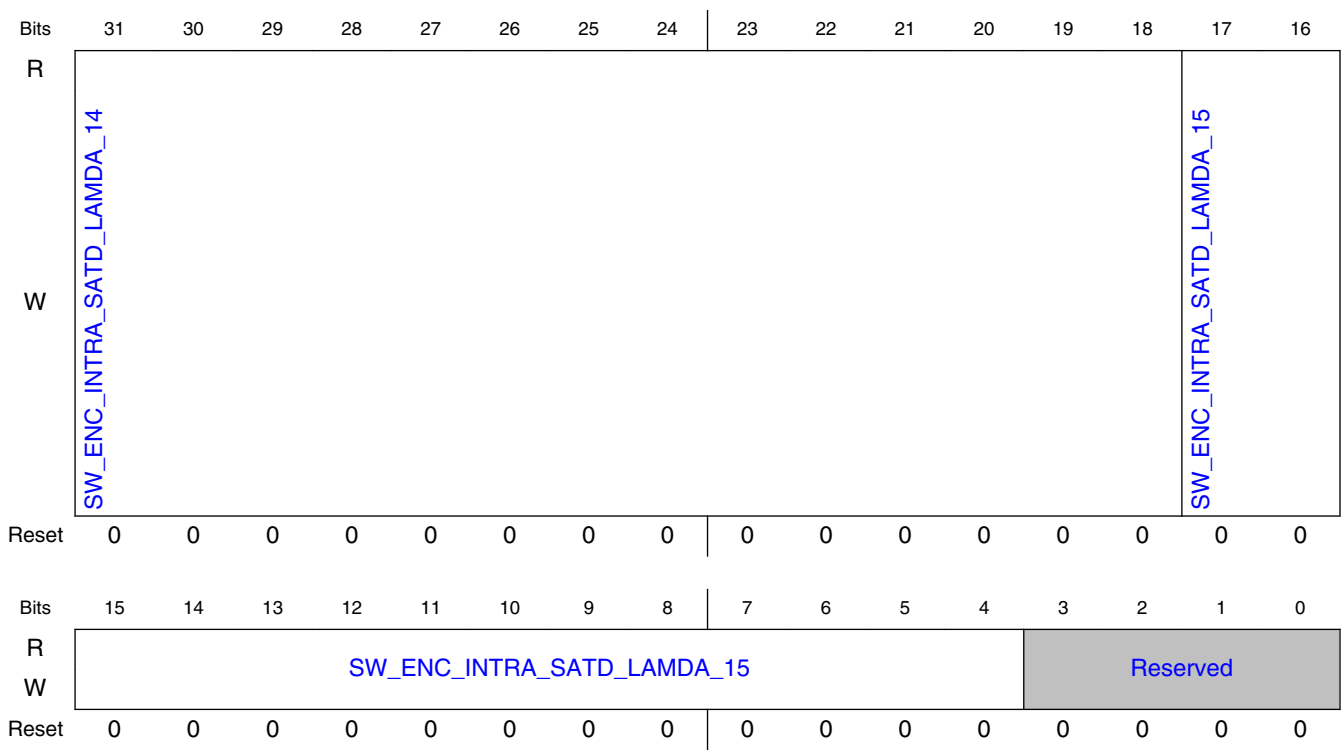
16.3.5.1.101.1 Offset

Register	Offset
swreg132	210h

16.3.5.1.101.2 Function

intra SATD lambda config 7

16.3.5.1.101.3 Diagram



16.3.5.1.101.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_14	intra SATD lambda 14
17-4	intra SATD lambda 15

Table continues on the next page...

Field	Function
SW_ENC_INTRA_SATD_LAMDA_15	
3-0 —	Reserved

16.3.5.1.102 SSE devide 256 (swreg133)

16.3.5.1.102.1 Offset

Register	Offset
swreg133	214h

16.3.5.1.102.2 Function

SSE devide 256

16.3.5.1.102.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_SSE_DIV_256															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_SSE_DIV_256															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.102.4 Fields

Field	Function
31-0 SW_ENC_SSE_DIV_256	sse divide 256

16.3.5.1.103 inter me SSE lambda config 4. For H2V5 or later version. (swreg138_H2V5)

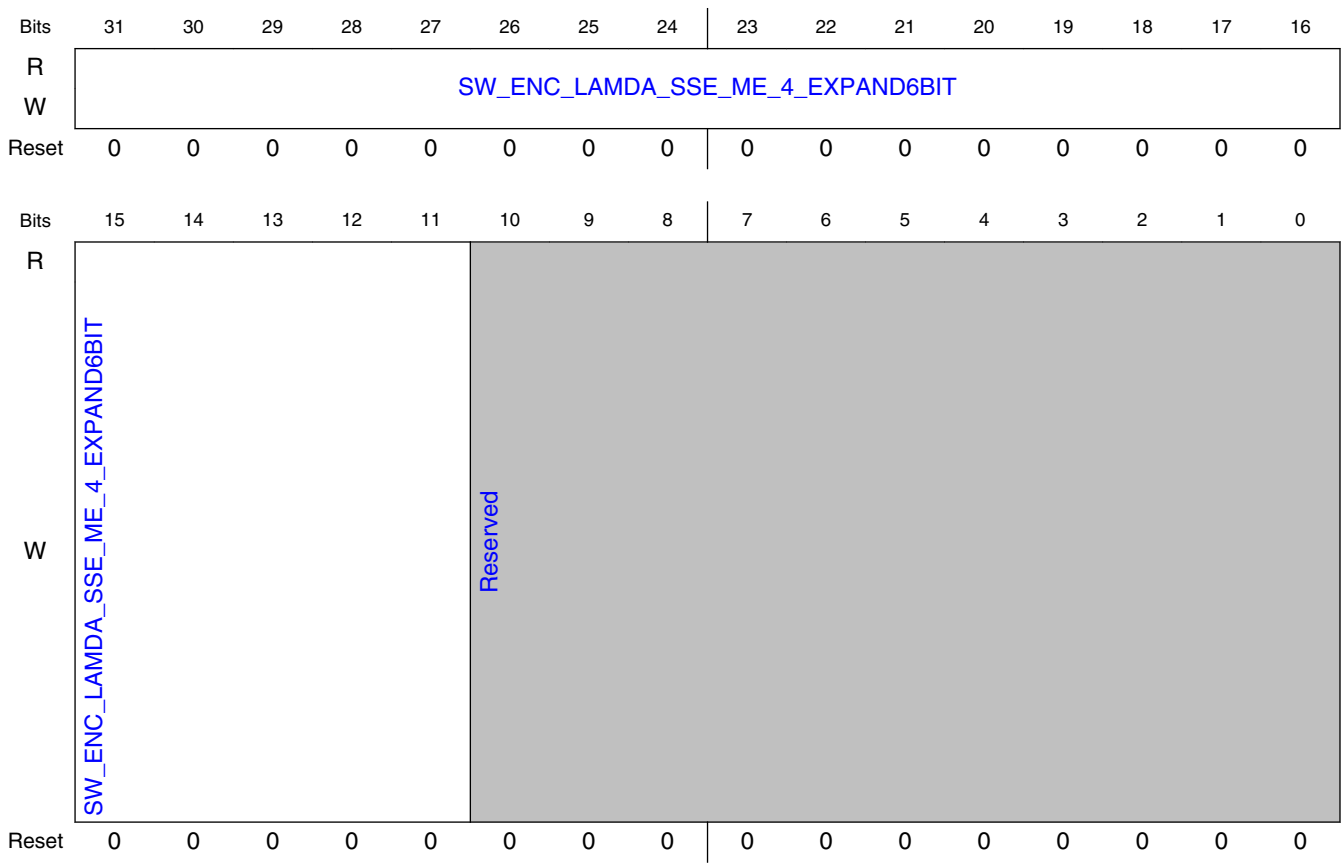
16.3.5.1.103.1 Offset

Register	Offset
swreg138_H2V5	228h

16.3.5.1.103.2 Function

inter me SSE lambda config 4. For H2V5 or later version.

16.3.5.1.103.3 Diagram



16.3.5.1.103.4 Fields

Field	Function
31-11	inter me SSE lambda 3

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SSE_ME_4_EXPAND6BIT	
10-0 —	Reserved

16.3.5.1.104 inter me SSE lambda config 5. For H2V5 or later version. (swreg139_H2V5)

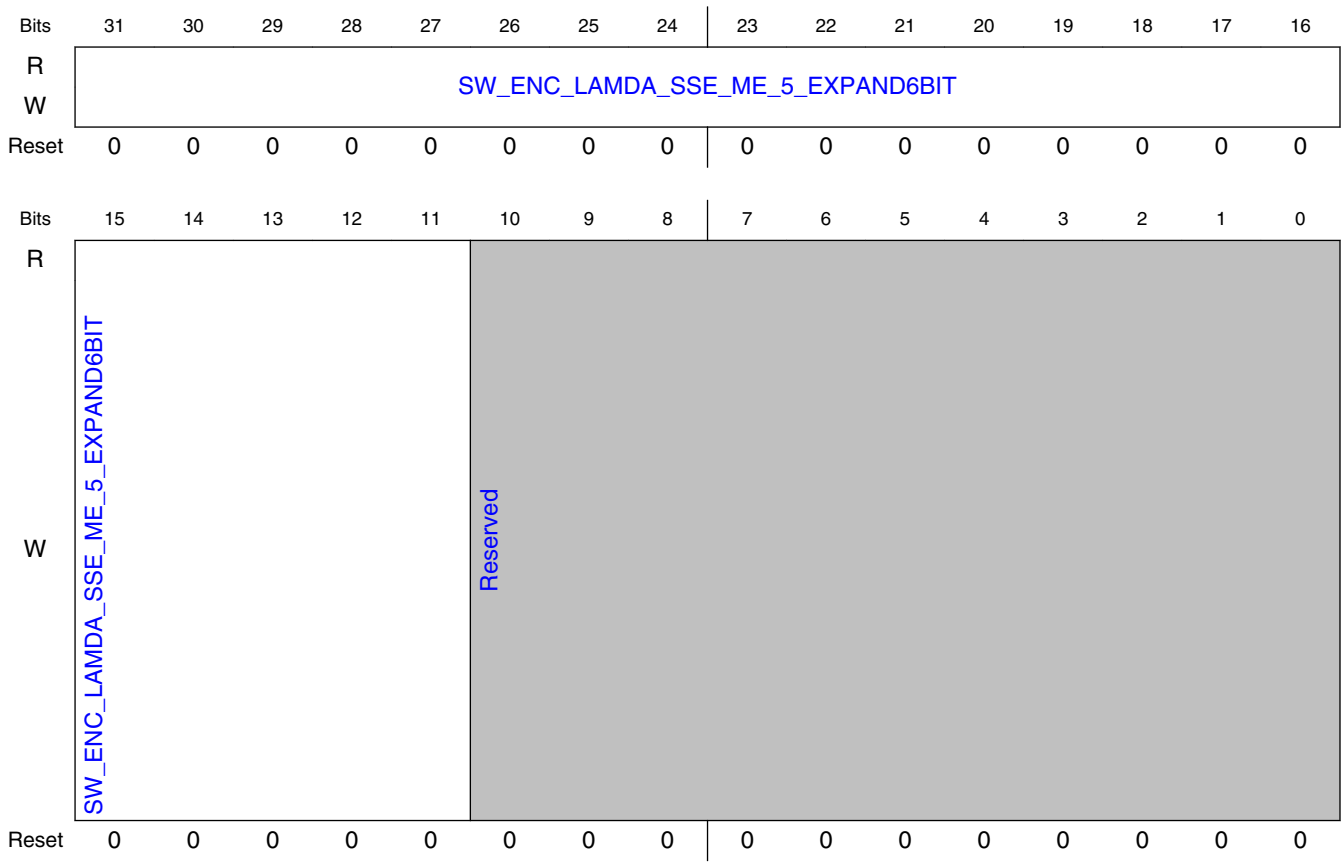
16.3.5.1.104.1 Offset

Register	Offset
swreg139_H2V5	22Ch

16.3.5.1.104.2 Function

inter me SSE lambda config 5. For H2V5 or later version.

16.3.5.1.104.3 Diagram



16.3.5.1.104.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_5_EXPAND6BIT	inter me SSE lambda 5
10-0 —	Reserved

16.3.5.1.105 inter me SSE lambda config 6. For H2V5 or later version. (swreg140_H2V5)

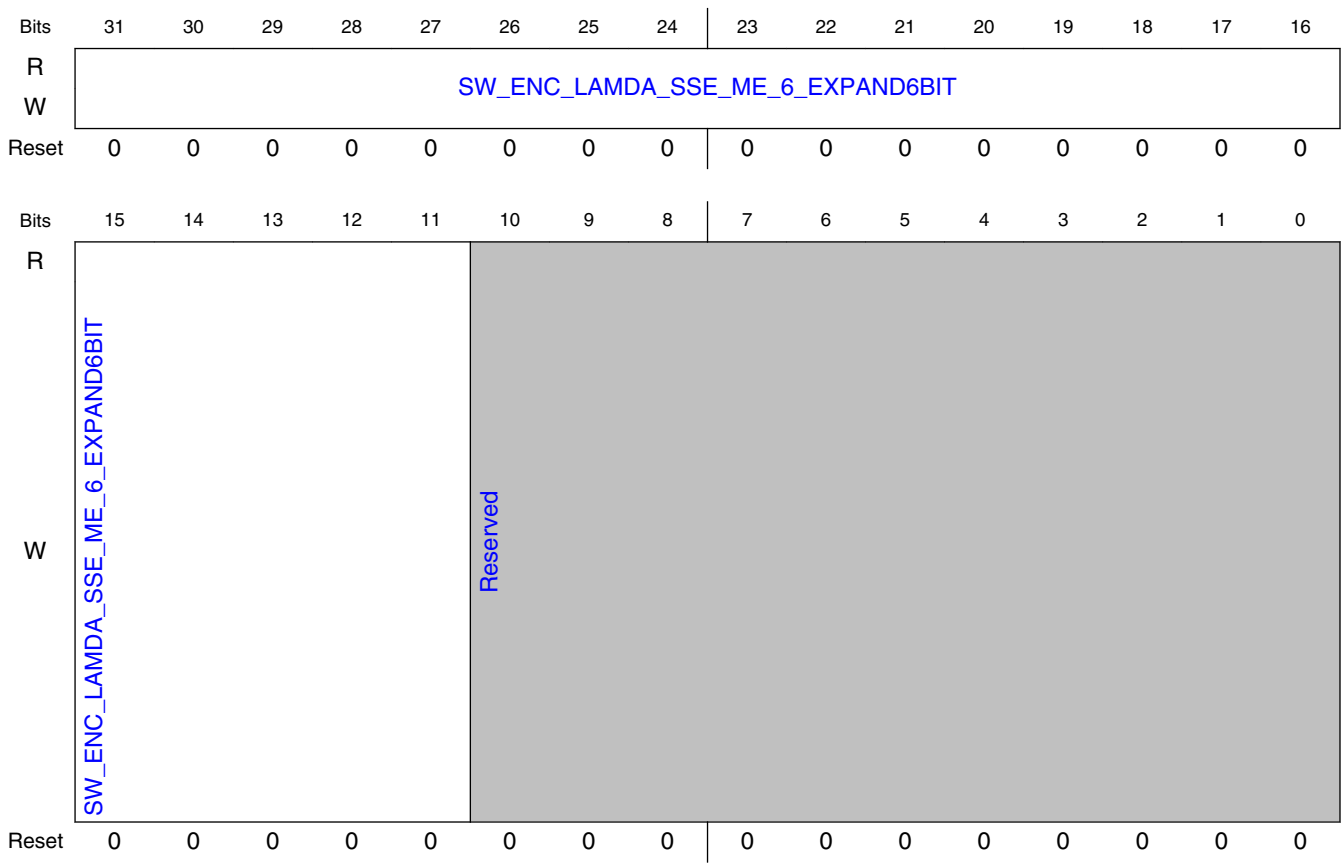
16.3.5.1.105.1 Offset

Register	Offset
swreg140_H2V5	230h

16.3.5.1.105.2 Function

inter me SSE lambda config 6. For H2V5 or later version.

16.3.5.1.105.3 Diagram



16.3.5.1.105.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_6_EXPAND6BIT	inter me SSE lambda 6
10-0	Reserved

Field	Function
—	

16.3.5.1.106 inter me SSE lambda config 7. For H2V5 or later version. (swreg141_H2V5)

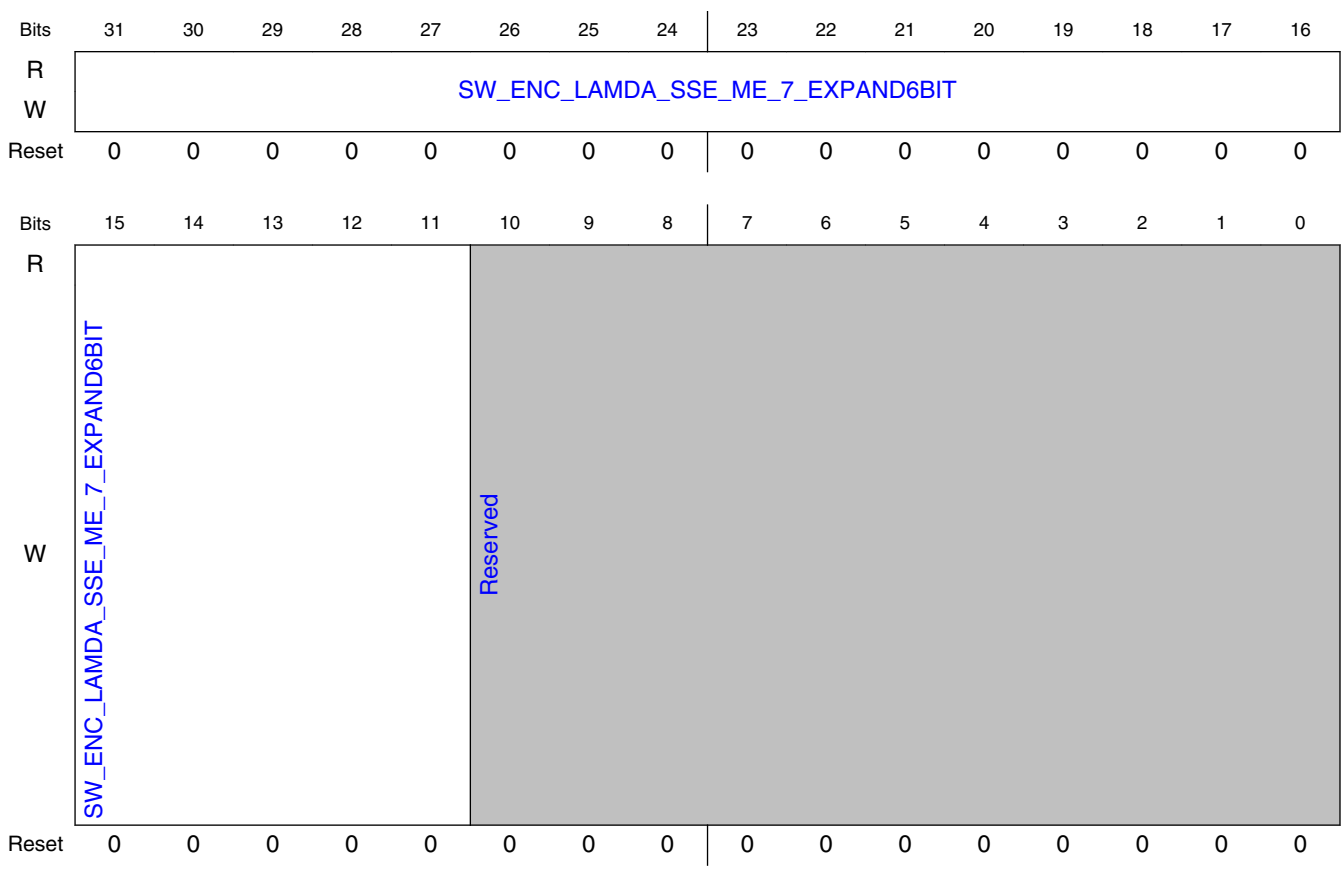
16.3.5.1.106.1 Offset

Register	Offset
swreg141_H2V5	234h

16.3.5.1.106.2 Function

inter me SSE lambda config 7. For H2V5 or later version.

16.3.5.1.106.3 Diagram



16.3.5.1.106.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_7_EXPAND6BIT	inter me SSE lambda 7
10-0 —	Reserved

16.3.5.1.107 inter me SSE lambda config 8. For H2V5 or later version. (swreg142_H2V5)

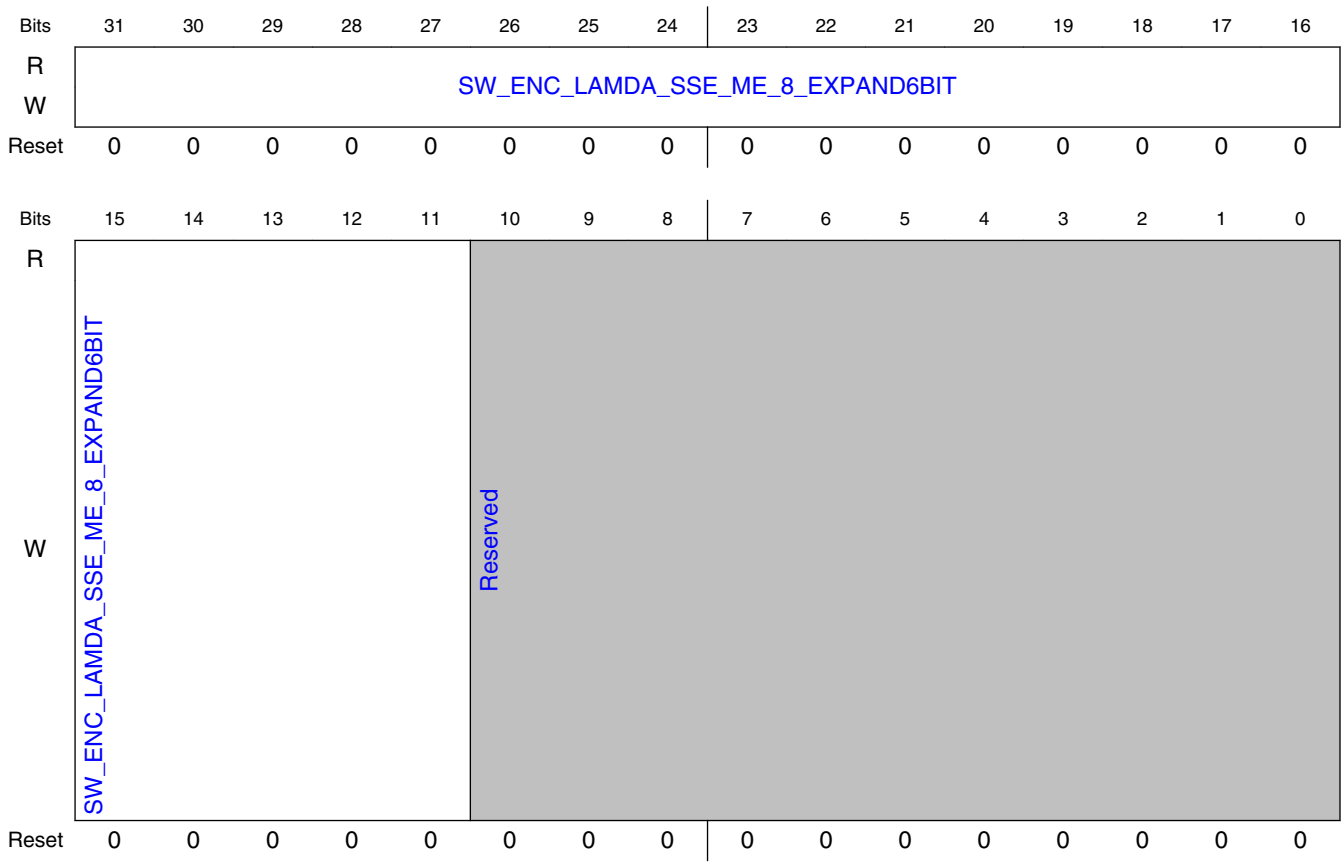
16.3.5.1.107.1 Offset

Register	Offset
swreg142_H2V5	238h

16.3.5.1.107.2 Function

inter me SSE lambda config 8. For H2V5 or later version.

16.3.5.1.107.3 Diagram



16.3.5.1.107.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_8_EXPAND6BIT	inter me SSE lambda 8
10-0 —	Reserved

16.3.5.1.108 inter me SSE lambda config 9. For H2V5 or later version. (swreg143_H2V5)

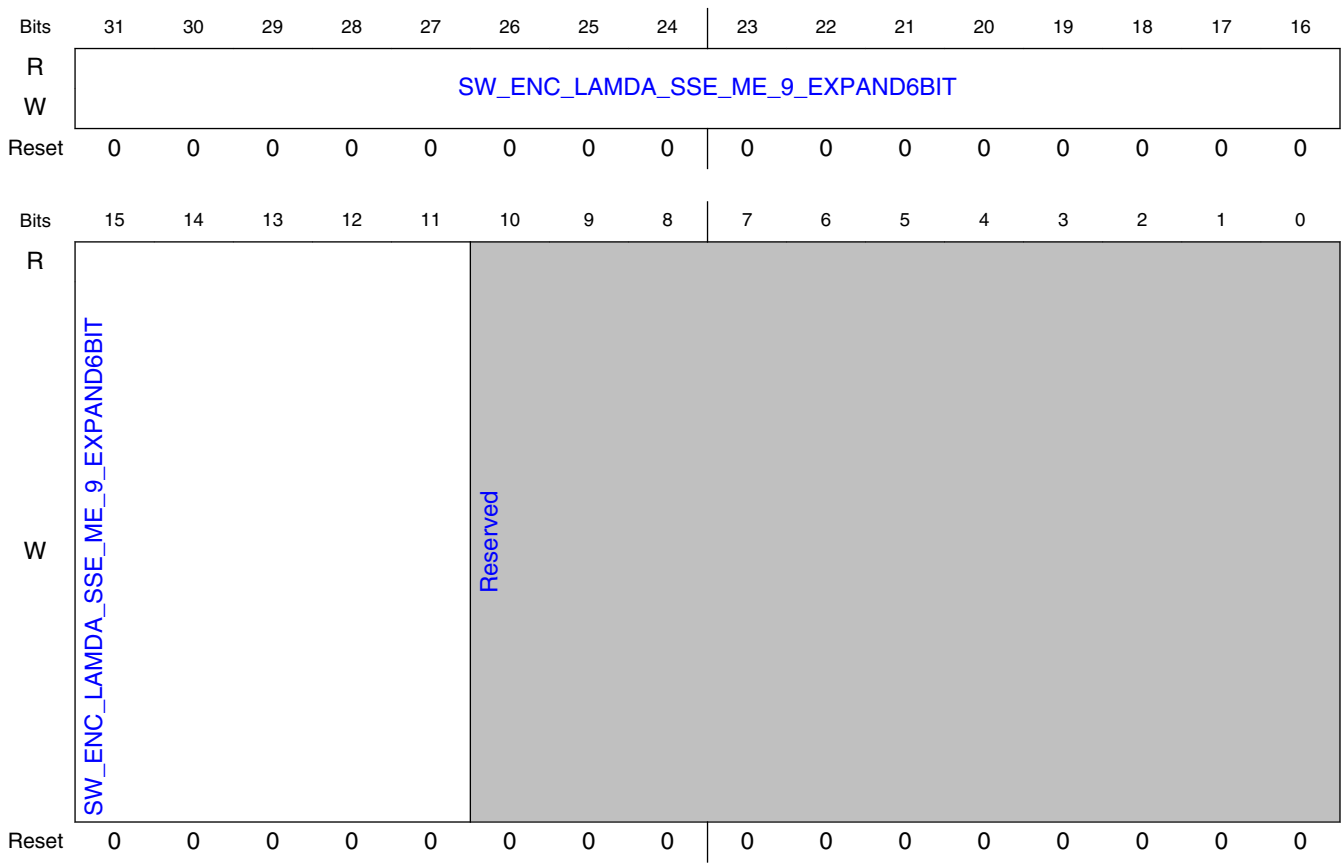
16.3.5.1.108.1 Offset

Register	Offset
swreg143_H2V5	23Ch

16.3.5.1.108.2 Function

inter me SSE lambda config 9. For H2V5 or later version.

16.3.5.1.108.3 Diagram



16.3.5.1.108.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_9_EXPAND6BIT	inter me SSE lambda 9
10-0	Reserved

Field	Function
—	

16.3.5.1.109 inter me SSE lambda config 10. For H2V5 or later version. (swreg144_H2V5)

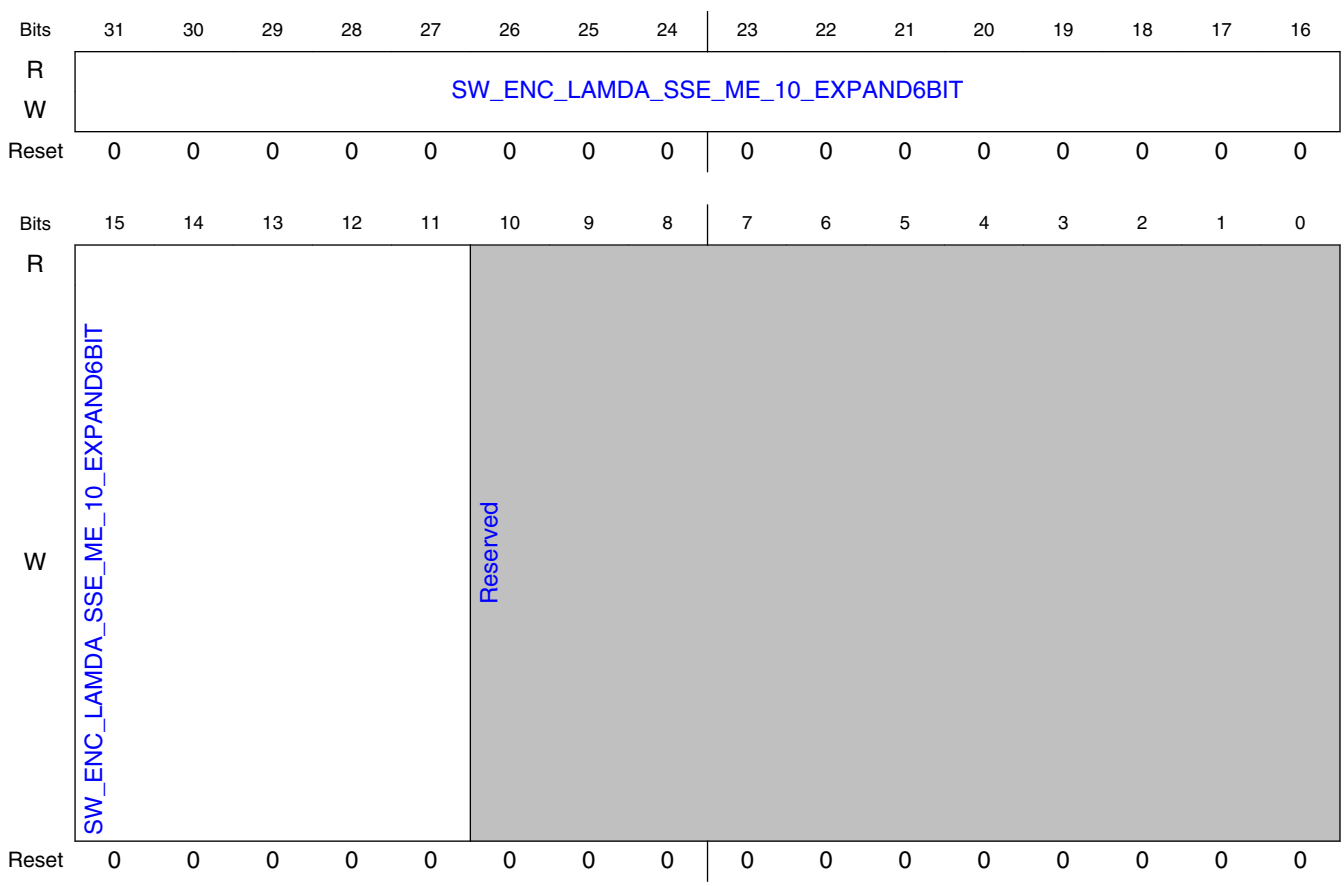
16.3.5.1.109.1 Offset

Register	Offset
swreg144_H2V5	240h

16.3.5.1.109.2 Function

inter me SSE lambda config 10. For H2V5 or later version.

16.3.5.1.109.3 Diagram



16.3.5.1.109.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_10_EXPAND6BIT	inter me SSE lambda 10
10-0 —	Reserved

16.3.5.1.110 inter me SSE lambda config 11. For H2V5 or later version. (swreg145_H2V5)

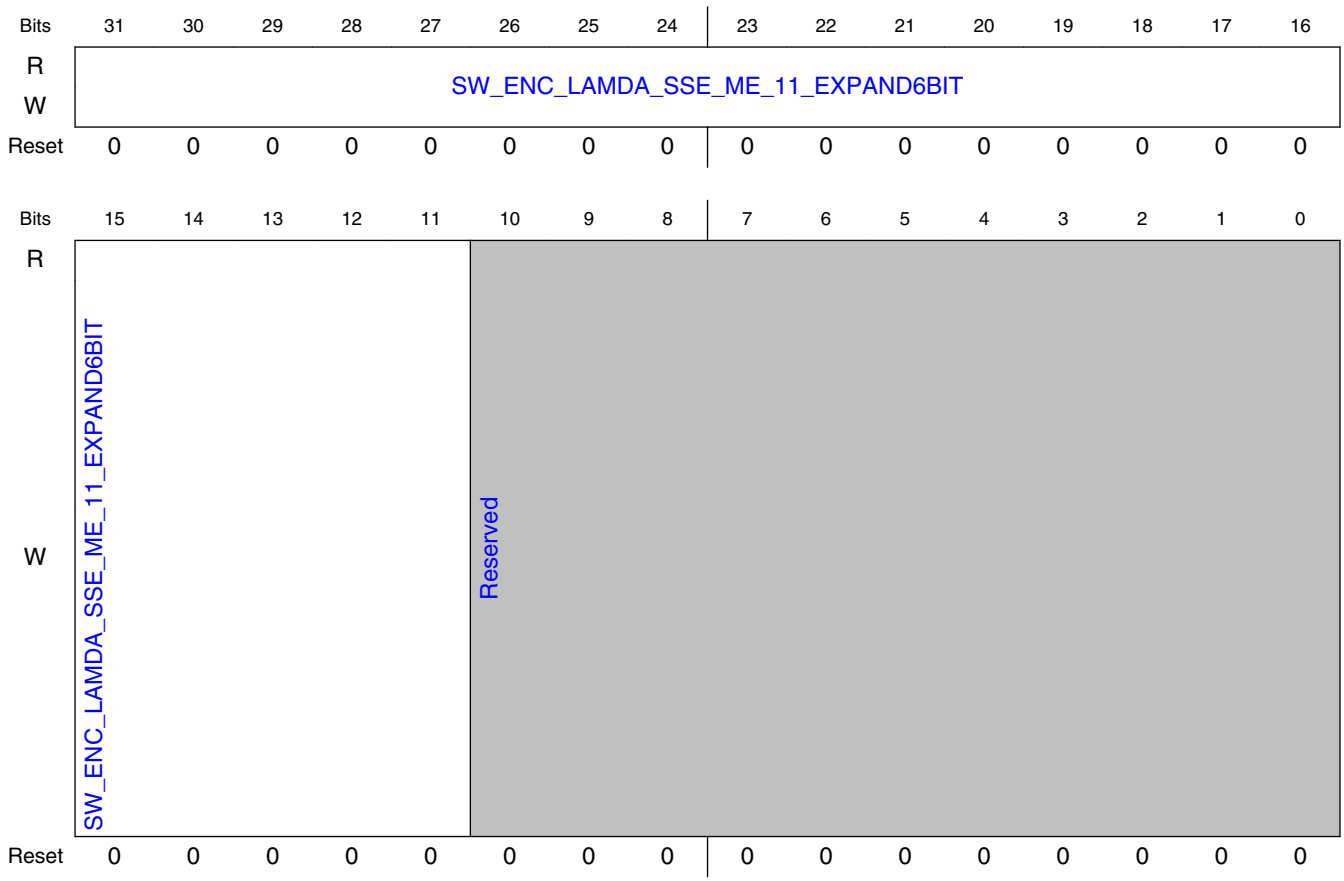
16.3.5.1.110.1 Offset

Register	Offset
swreg145_H2V5	244h

16.3.5.1.110.2 Function

inter me SSE lambda config 11. For H2V5 or later version.

16.3.5.1.110.3 Diagram



16.3.5.1.110.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_11_EXPAND6BIT	inter me SSE lambda 11
10-0 —	Reserved

16.3.5.1.111 inter me SSE lambda config 12. For H2V5 or later version. (swreg146_H2V5)

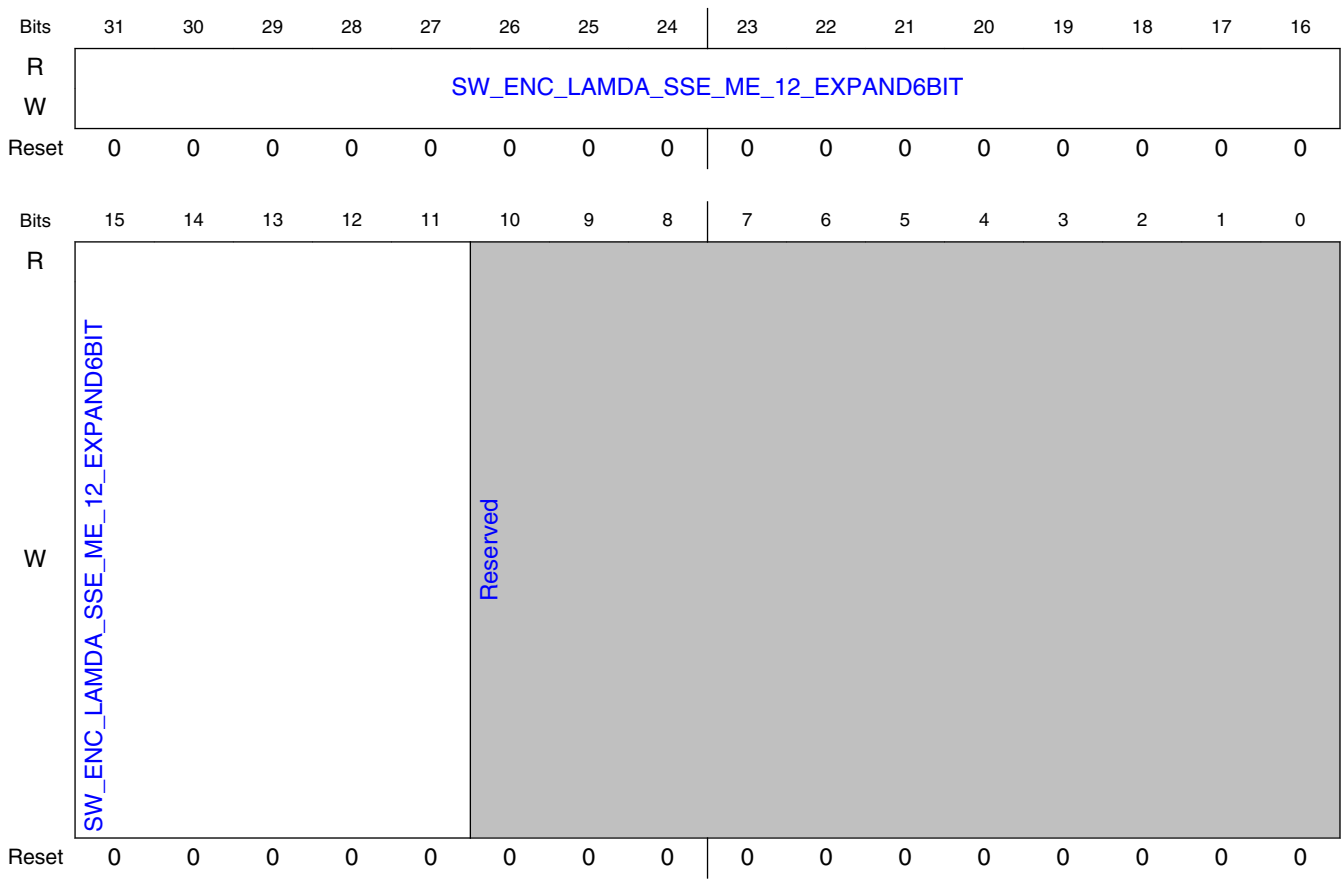
16.3.5.1.111.1 Offset

Register	Offset
swreg146_H2V5	248h

16.3.5.1.111.2 Function

inter me SSE lambda config 12. For H2V5 or later version.

16.3.5.1.111.3 Diagram



16.3.5.1.111.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_12_EXPAND6BIT	inter me SSE lambda 12

Table continues on the next page...

Field	Function
10-0 —	Reserved

16.3.5.1.112 inter me SSE lambda config 13. For H2V5 or later version. (swreg147_H2V5)

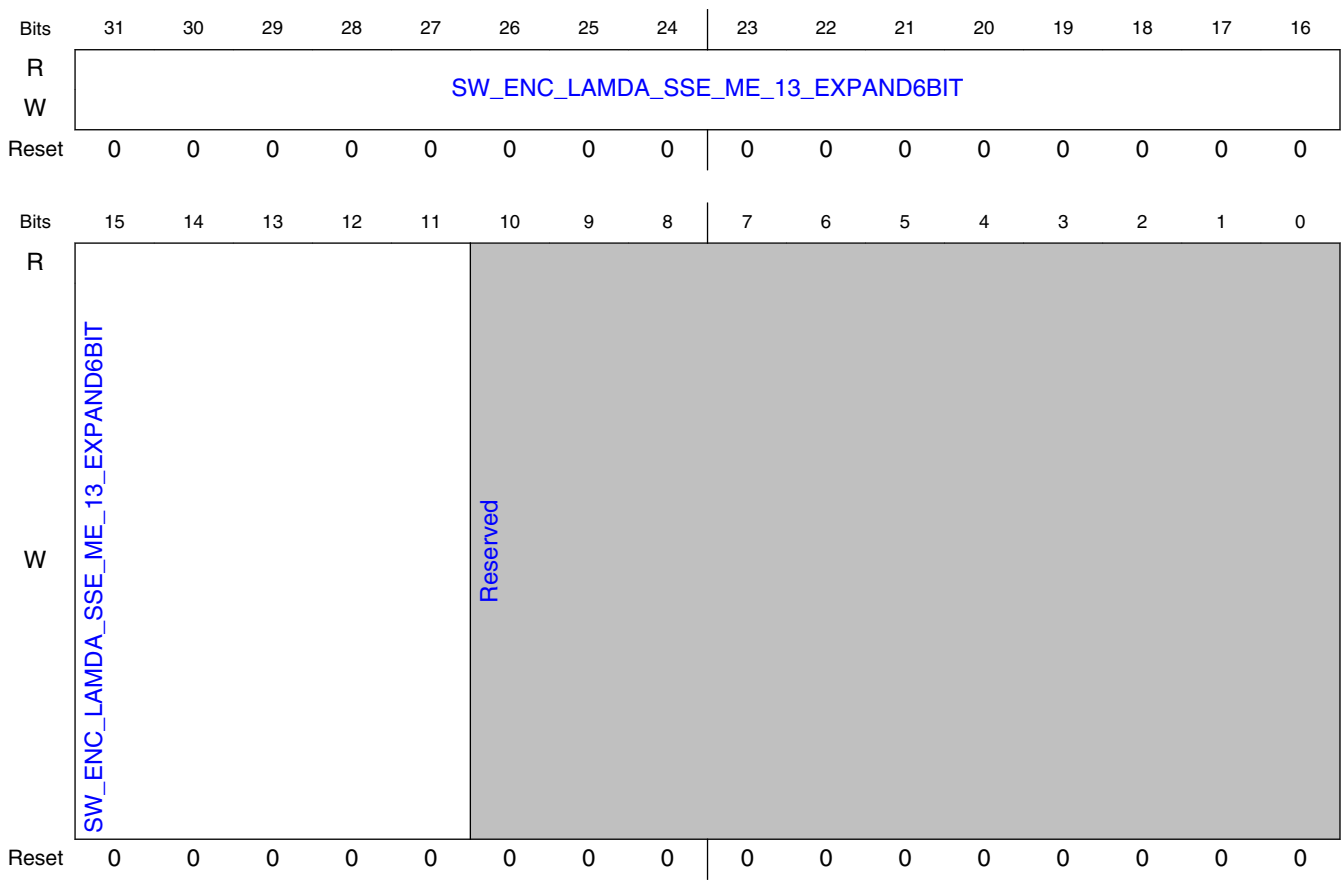
16.3.5.1.112.1 Offset

Register	Offset
swreg147_H2V5	24Ch

16.3.5.1.112.2 Function

inter me SSE lambda config 13. For H2V5 or later version.

16.3.5.1.112.3 Diagram



16.3.5.1.112.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_13_EXPAND6BIT	inter me SSE lambda 13
10-0 —	Reserved

16.3.5.1.113 inter me SSE lambda config 14. For H2V5 or later version. (swreg148_H2V5)

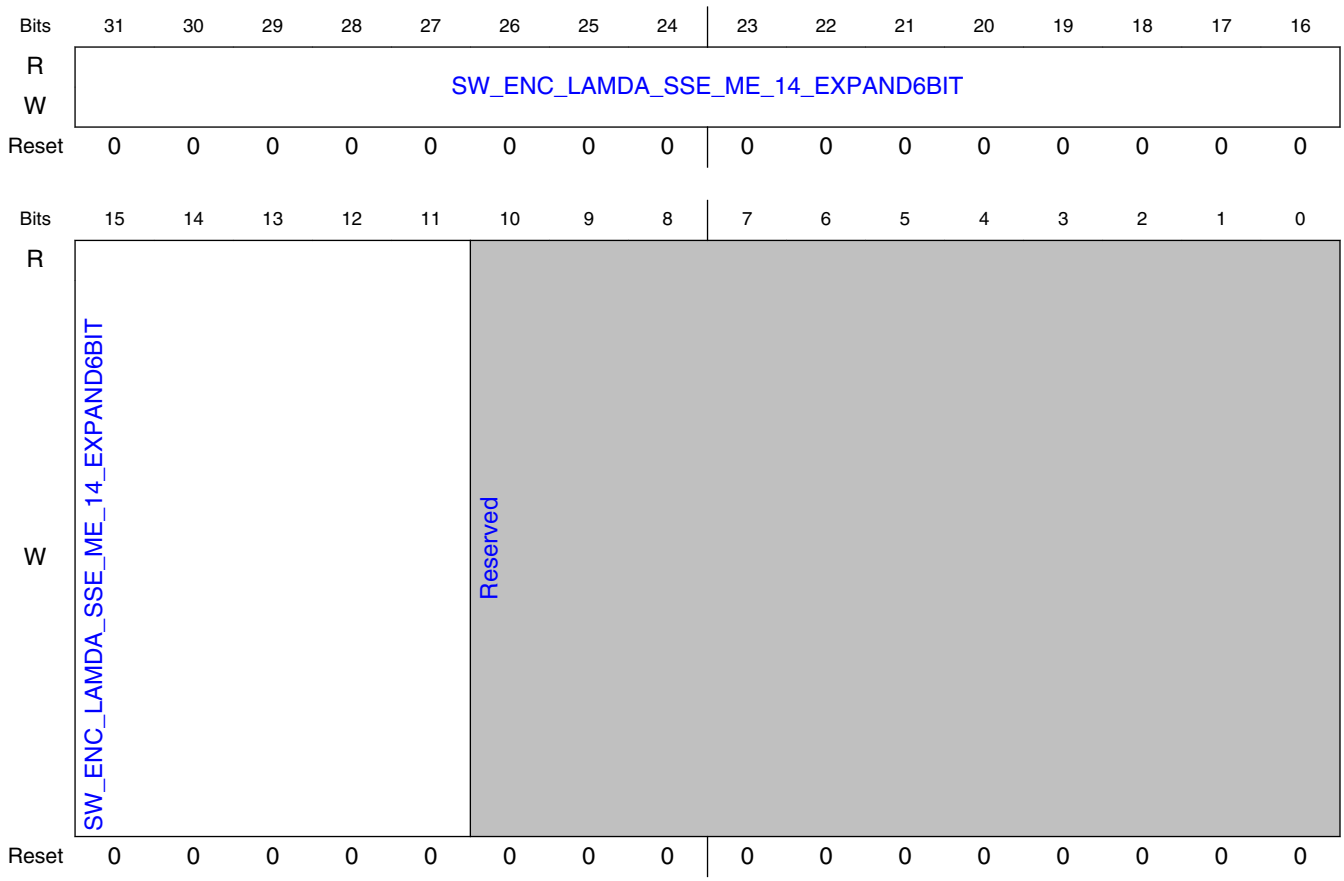
16.3.5.1.113.1 Offset

Register	Offset
swreg148_H2V5	250h

16.3.5.1.113.2 Function

inter me SSE lambda config 14. For H2V5 or later version.

16.3.5.1.113.3 Diagram



16.3.5.1.113.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_14_EXPAND6BIT	inter me SSE lambda 14
10-0 —	Reserved

16.3.5.1.114 inter me SSE lambda config 15. For H2V5 or later version. (swreg149_H2V5)

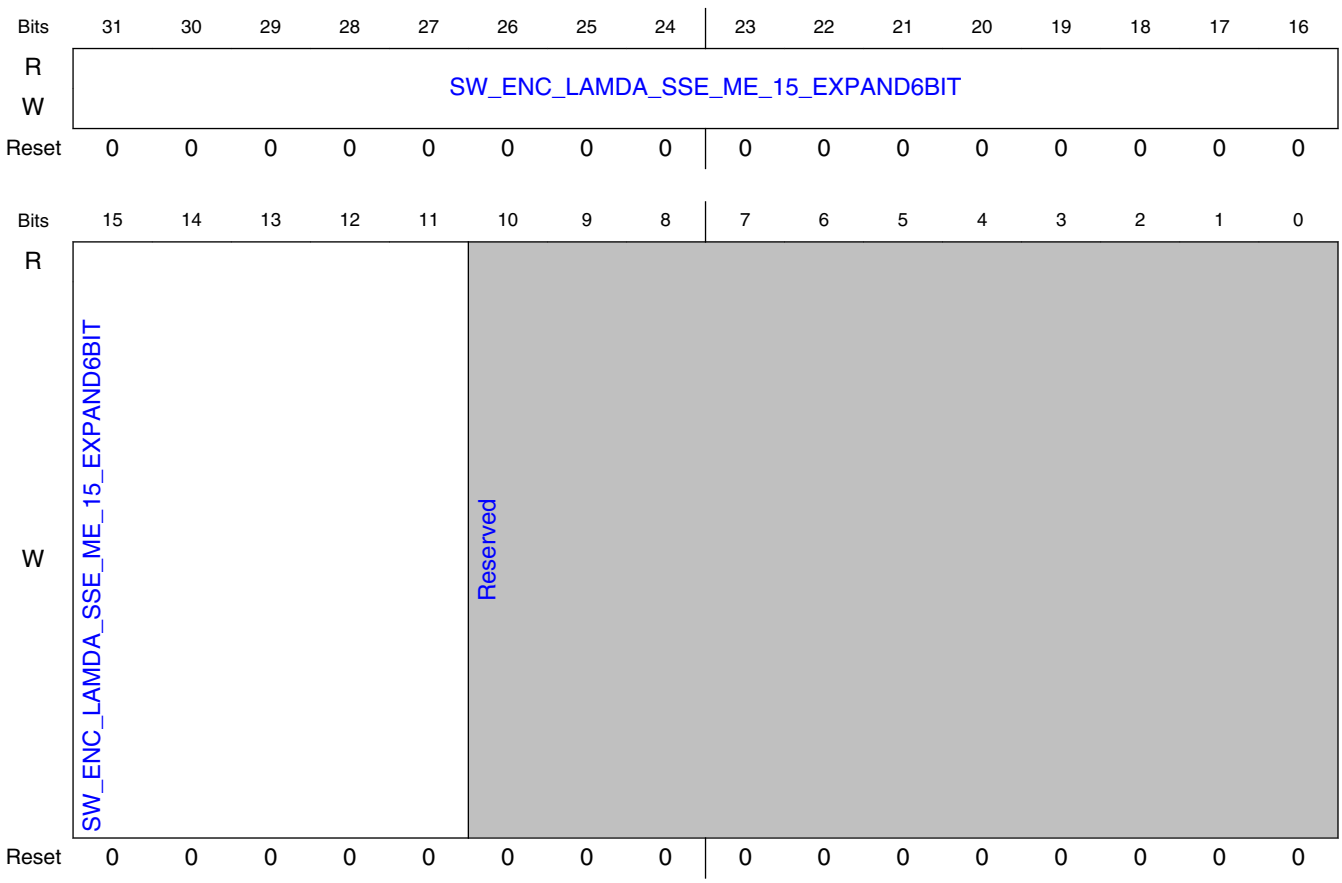
16.3.5.1.114.1 Offset

Register	Offset
swreg149_H2V5	254h

16.3.5.1.114.2 Function

inter me SSE lambda config 15. For H2V5 or later version.

16.3.5.1.114.3 Diagram



16.3.5.1.114.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_15_EXPAND6BIT	inter me SSE lambda 15

Table continues on the next page...

Field	Function
10-0 —	Reserved

16.3.5.1.115 inter me SATD lambda config 8 (swreg150)

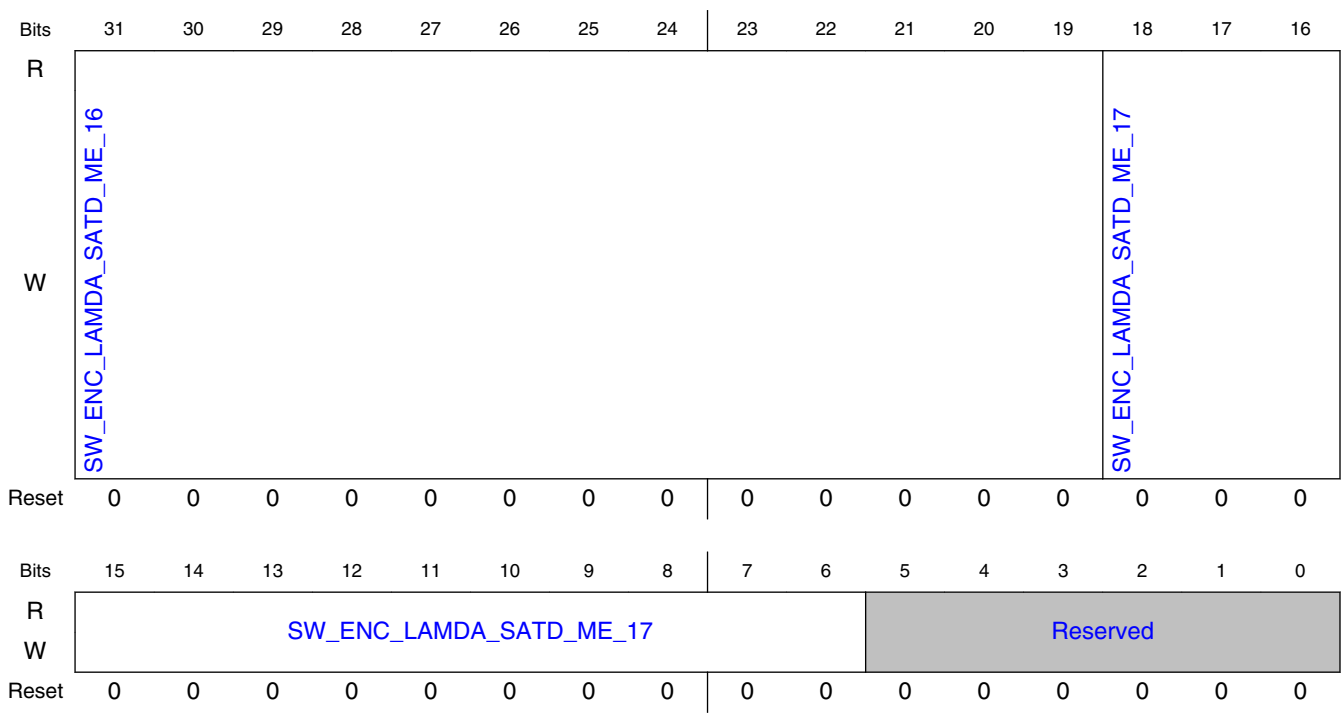
16.3.5.1.115.1 Offset

Register	Offset
swreg150	258h

16.3.5.1.115.2 Function

inter me SATD lambda config 8

16.3.5.1.115.3 Diagram



16.3.5.1.115.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_16	lambda satd me 16
18-6 SW_ENC_LAMDA_SATD_ME_17	lambda satd me 17
5-0 —	Reserved

16.3.5.1.116 inter me SATD lambda config 9 (swreg151)

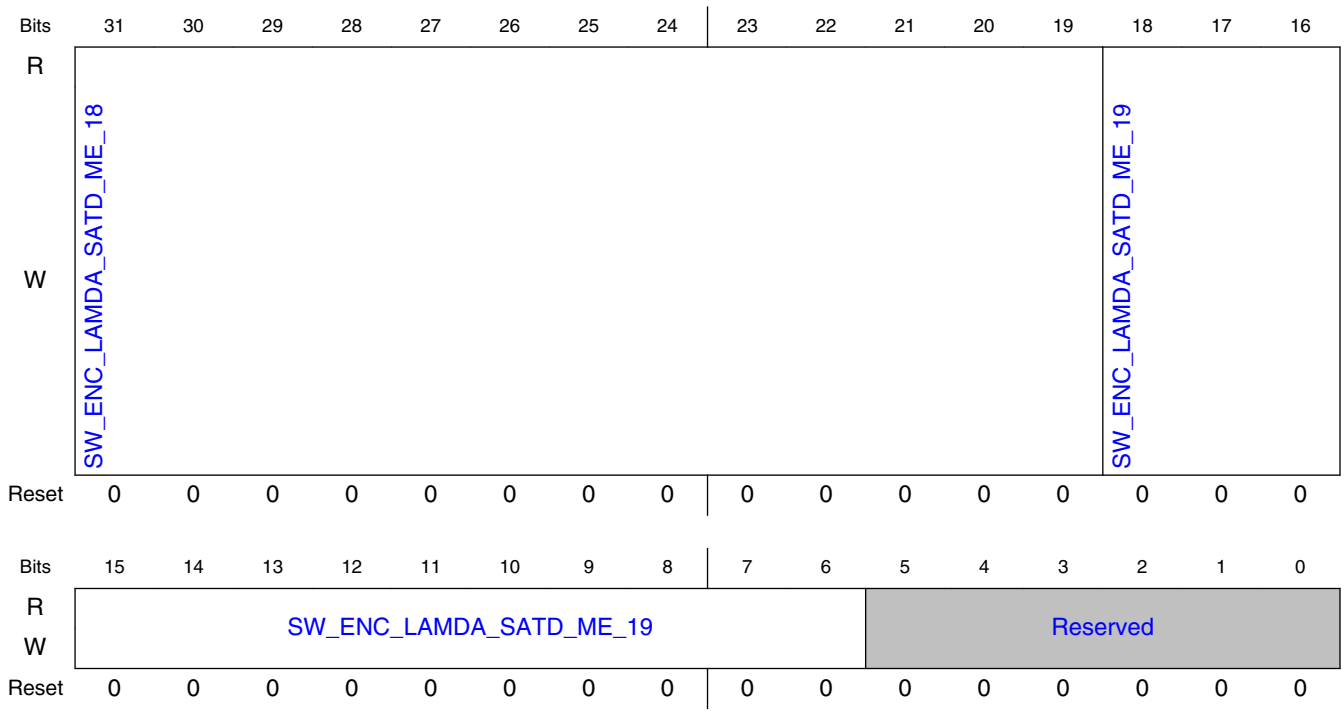
16.3.5.1.116.1 Offset

Register	Offset
swreg151	25Ch

16.3.5.1.116.2 Function

inter me SATD lambda config 9

16.3.5.1.116.3 Diagram



16.3.5.1.116.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_18	lambda satd me 18
18-6 SW_ENC_LAMDA_SATD_ME_19	lambda satd me 19
5-0 —	Reserved

16.3.5.1.117 inter me SATD lambda config 10 (swreg152)

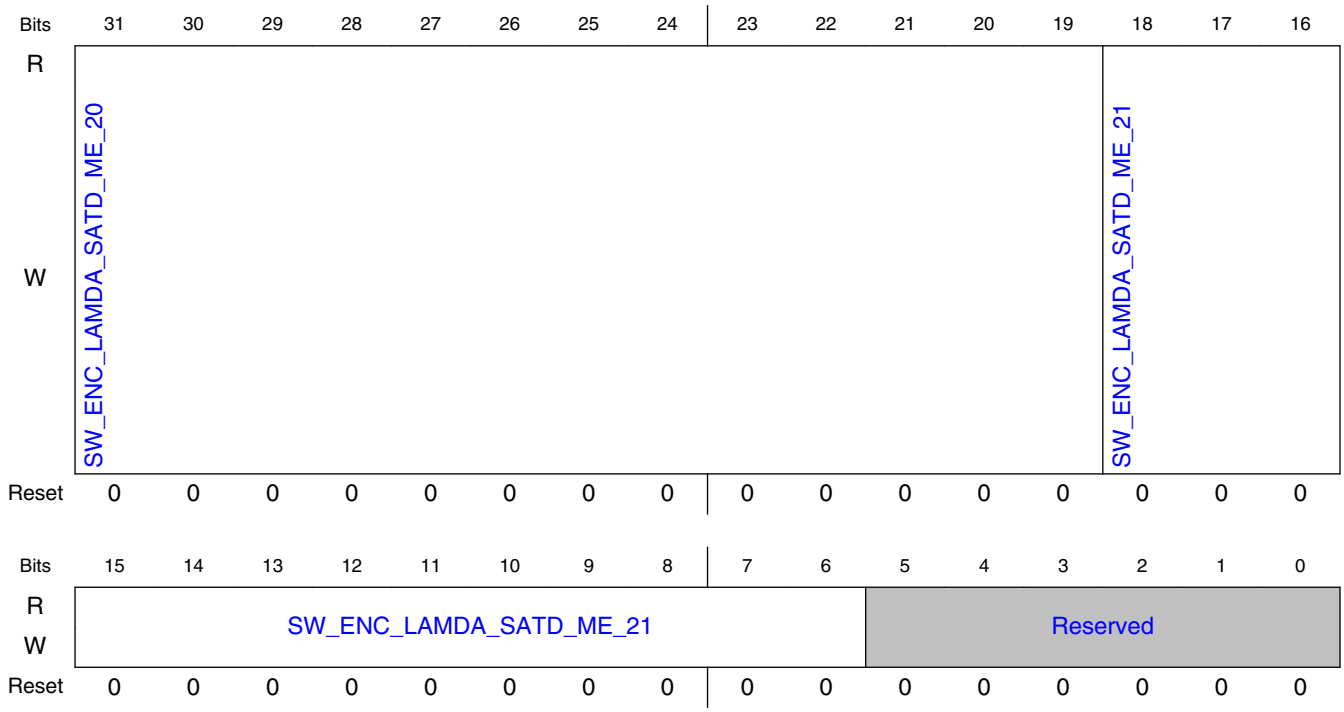
16.3.5.1.117.1 Offset

Register	Offset
swreg152	260h

16.3.5.1.117.2 Function

inter me SATD lambda config 10

16.3.5.1.117.3 Diagram



16.3.5.1.117.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_20	lambda satd me 20
18-6 SW_ENC_LAMDA_SATD_ME_21	lambda satd me 21
5-0 —	Reserved

16.3.5.1.118 inter me SATD lambda config 11 (swreg153)

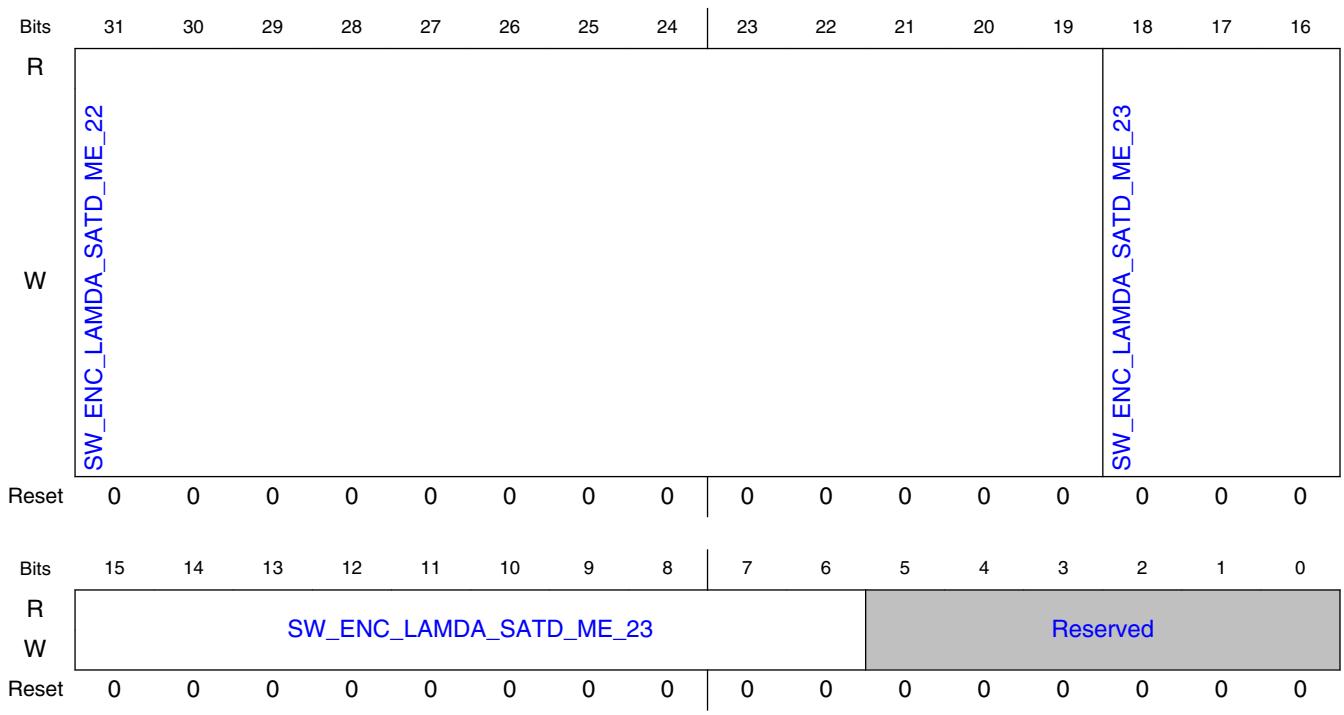
16.3.5.1.118.1 Offset

Register	Offset
swreg153	264h

16.3.5.1.118.2 Function

inter me SATD lambda config 11

16.3.5.1.118.3 Diagram



16.3.5.1.118.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_22	lambda satd me 22
18-6 SW_ENC_LAMDA_SATD_ME_23	lambda satd me 23
5-0	Reserved

Field	Function
—	

16.3.5.1.119 inter me SATD lambda config 12 (swreg154)

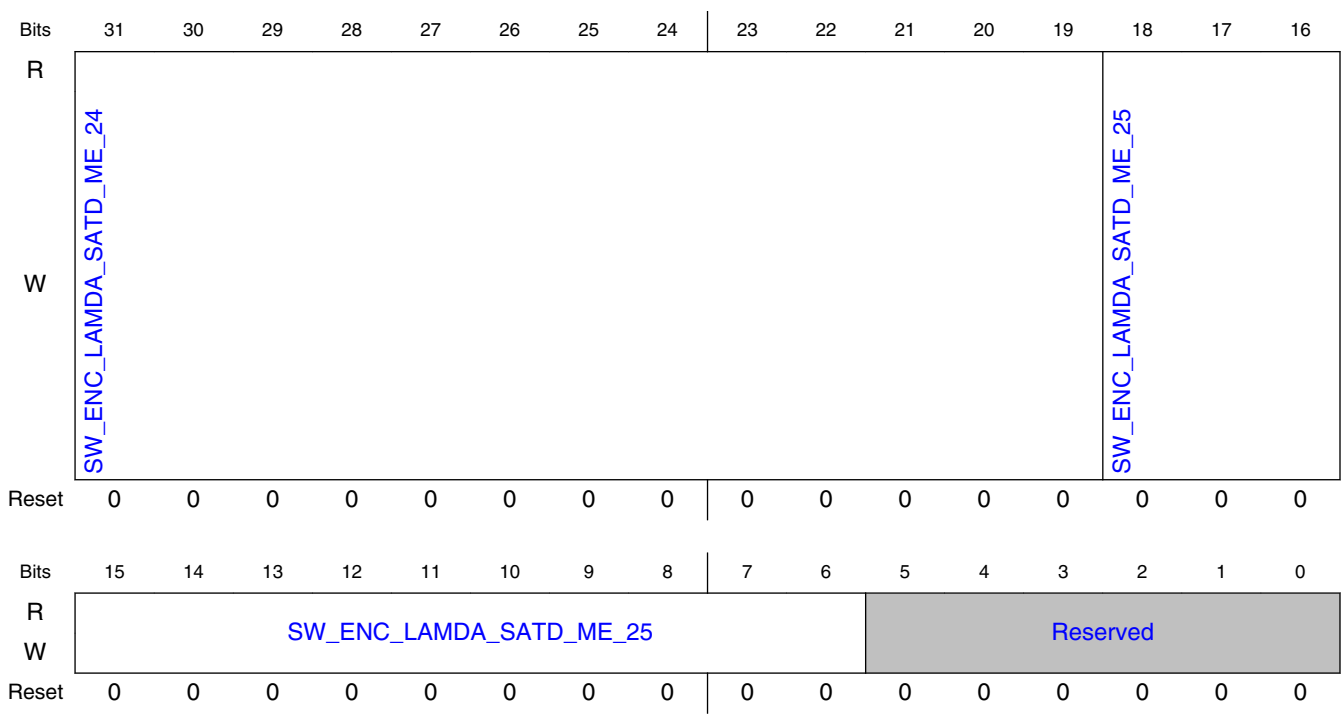
16.3.5.1.119.1 Offset

Register	Offset
swreg154	268h

16.3.5.1.119.2 Function

inter me SATD lambda config 12

16.3.5.1.119.3 Diagram



16.3.5.1.119.4 Fields

Field	Function
31-19	lambda satd me 24

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_24	
18-6 SW_ENC_LAMDA_SATD_ME_25	lambda satd me 25
5-0 —	Reserved

16.3.5.1.120 inter me SATD lambda config 13 (swreg155)

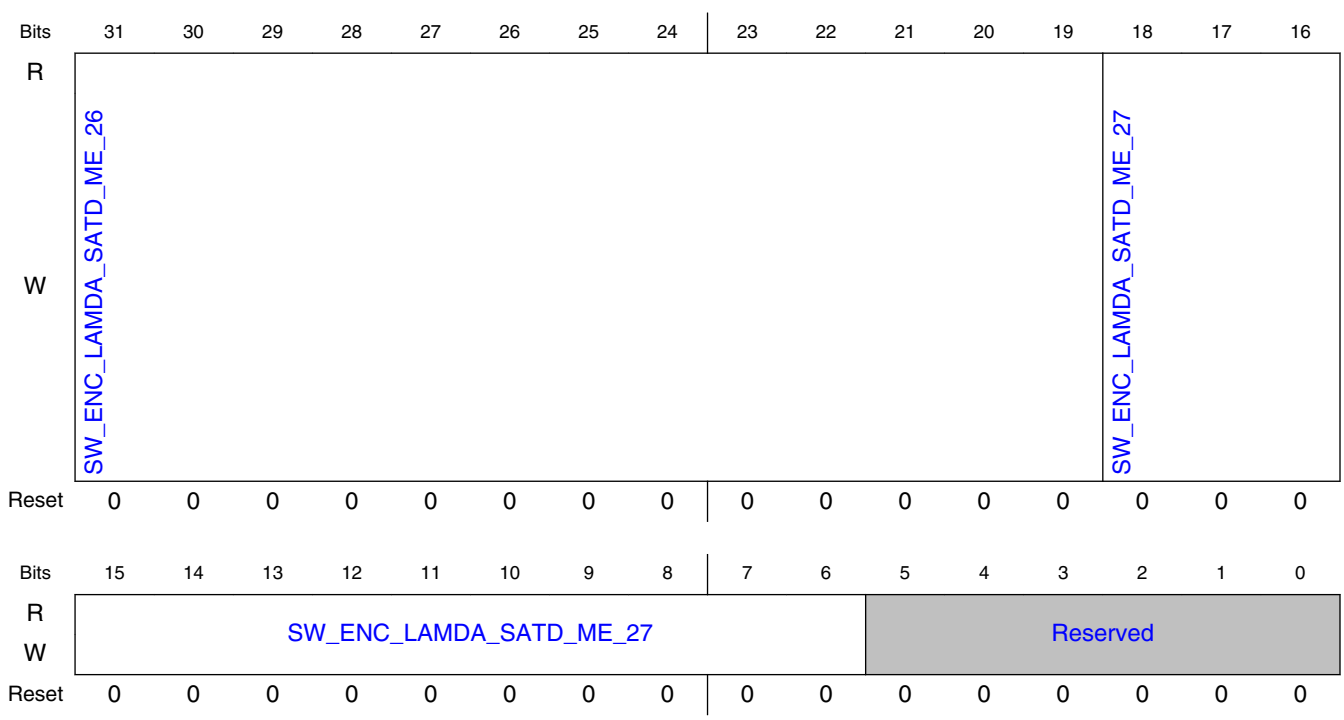
16.3.5.1.120.1 Offset

Register	Offset
swreg155	26Ch

16.3.5.1.120.2 Function

inter me SATD lambda config 13

16.3.5.1.120.3 Diagram



16.3.5.1.120.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_26	lambda satd me 26
18-6 SW_ENC_LAMDA_SATD_ME_27	lambda satd me 27
5-0 —	Reserved

16.3.5.1.121 inter me SATD lambda config 14 (swreg156)

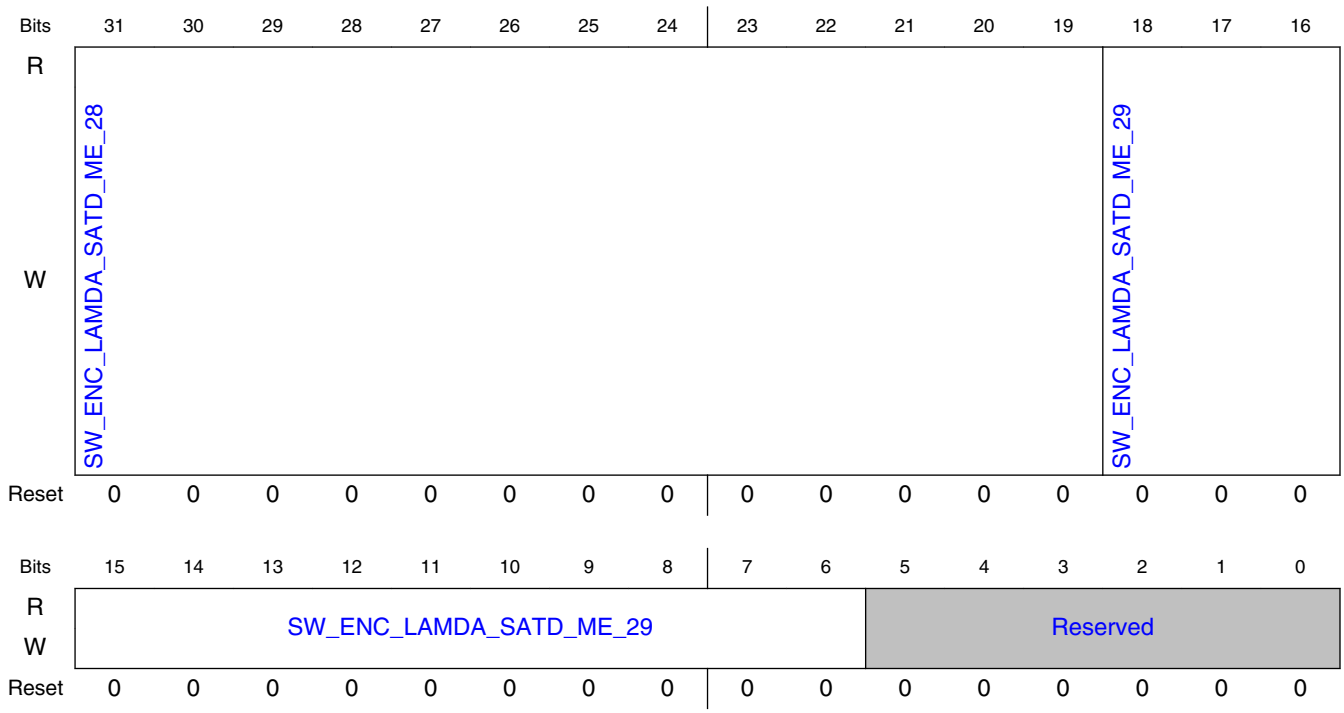
16.3.5.1.121.1 Offset

Register	Offset
swreg156	270h

16.3.5.1.121.2 Function

inter me SATD lambda config 14

16.3.5.1.121.3 Diagram



16.3.5.1.121.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_28	lambda satd me 28
18-6 SW_ENC_LAMDA_SATD_ME_29	lambda satd me 29
5-0 —	Reserved

16.3.5.1.122 inter me SATD lambda config 15 (swreg157)

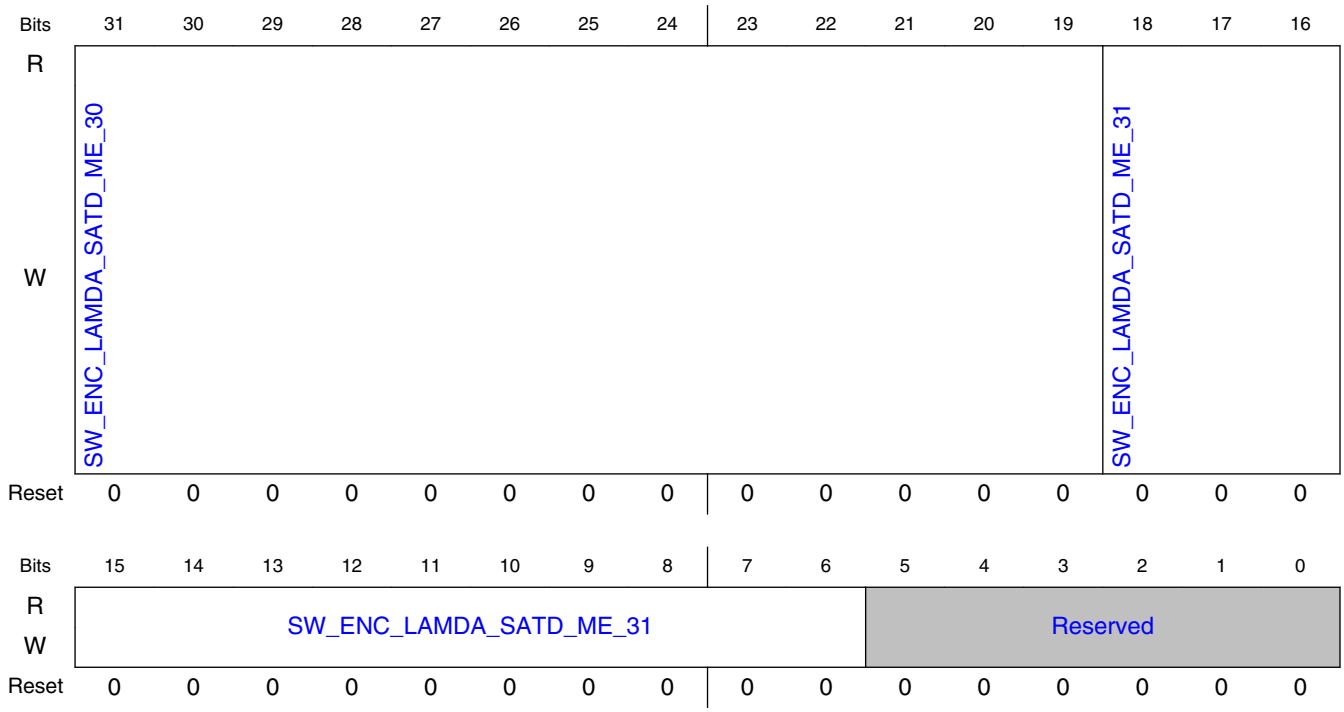
16.3.5.1.122.1 Offset

Register	Offset
swreg157	274h

16.3.5.1.122.2 Function

inter me SATD lambda config 15

16.3.5.1.122.3 Diagram



16.3.5.1.122.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_30	lambda satd me 30
18-6 SW_ENC_LAMDA_SATD_ME_31	lambda satd me 31
5-0 —	Reserved

16.3.5.1.123 inter me SSE lambda config 16 (swreg158)

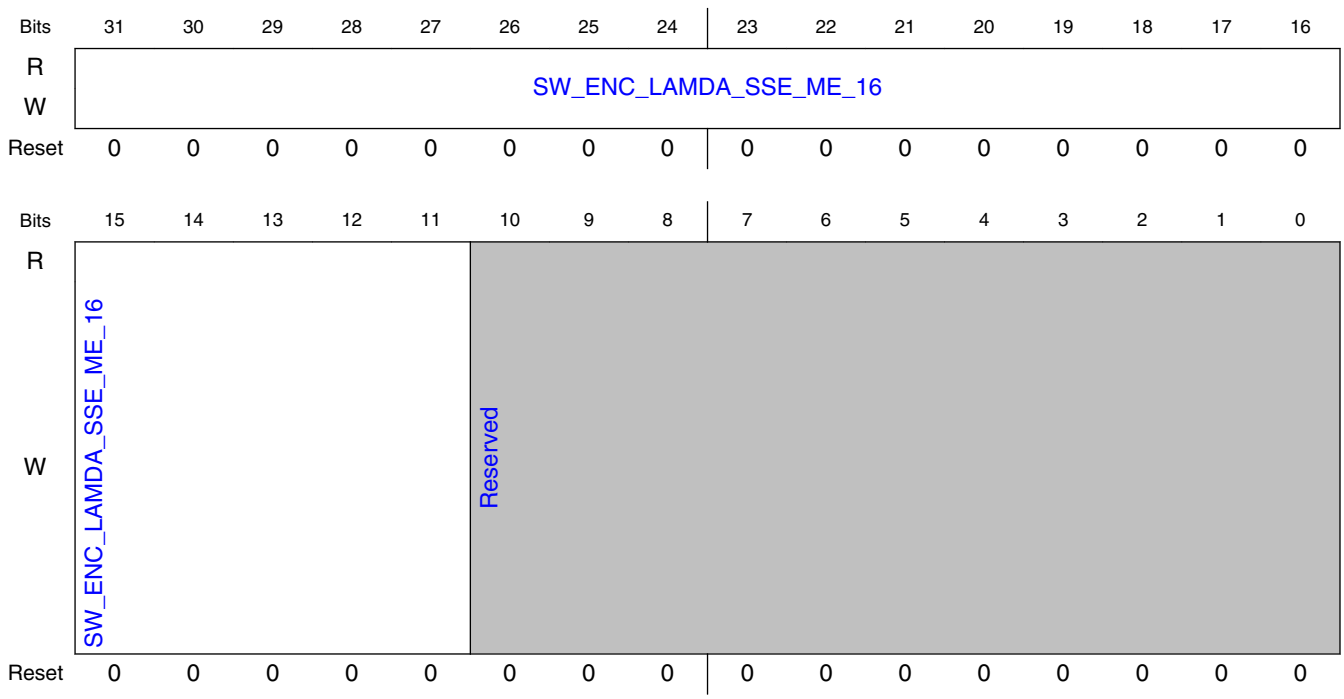
16.3.5.1.123.1 Offset

Register	Offset
swreg158	278h

16.3.5.1.123.2 Function

inter me SSE lambda config 16

16.3.5.1.123.3 Diagram



16.3.5.1.123.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_16	inter me SSE lambda 16
10-0 —	Reserved

16.3.5.1.124 inter me SSE lambda config 17 (swreg159)

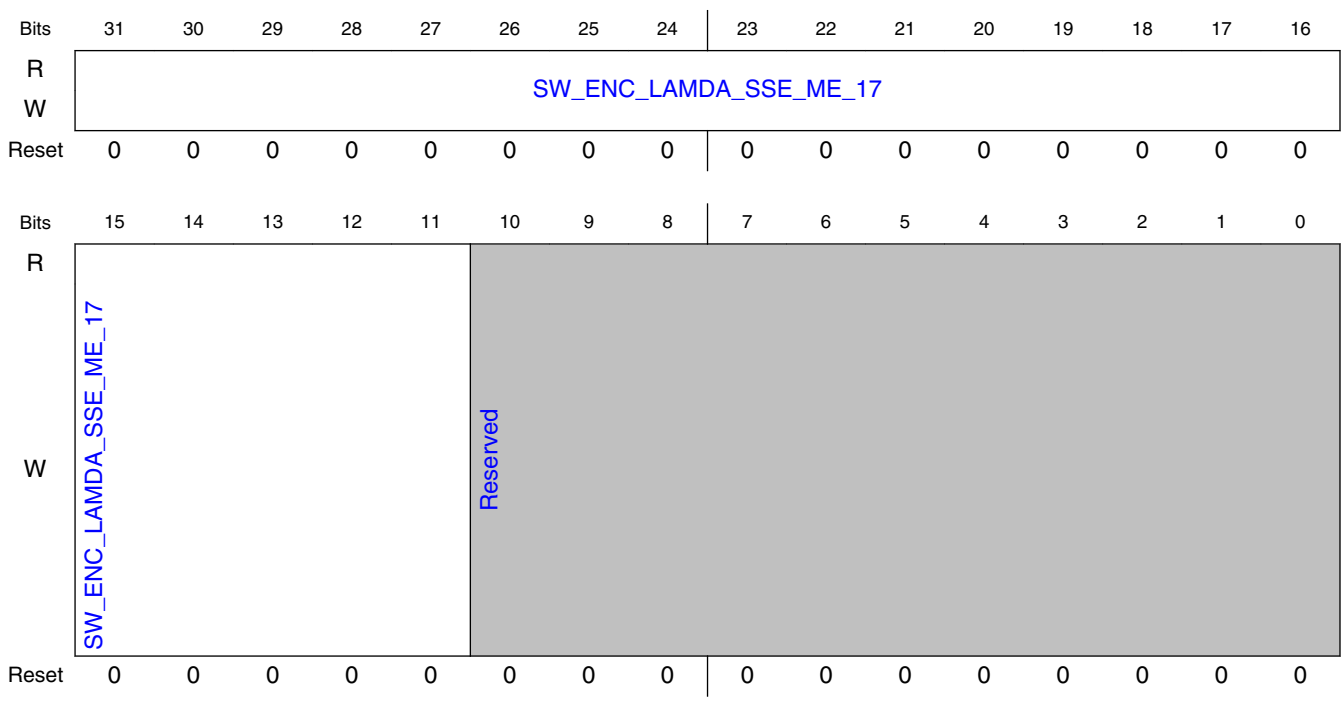
16.3.5.1.124.1 Offset

Register	Offset
swreg159	27Ch

16.3.5.1.124.2 Function

inter me SSE lambda config 17

16.3.5.1.124.3 Diagram



16.3.5.1.124.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_17	inter me SSE lambda 17
10-0 —	Reserved

16.3.5.1.125 inter me SSE lambda config 18 (swreg160)

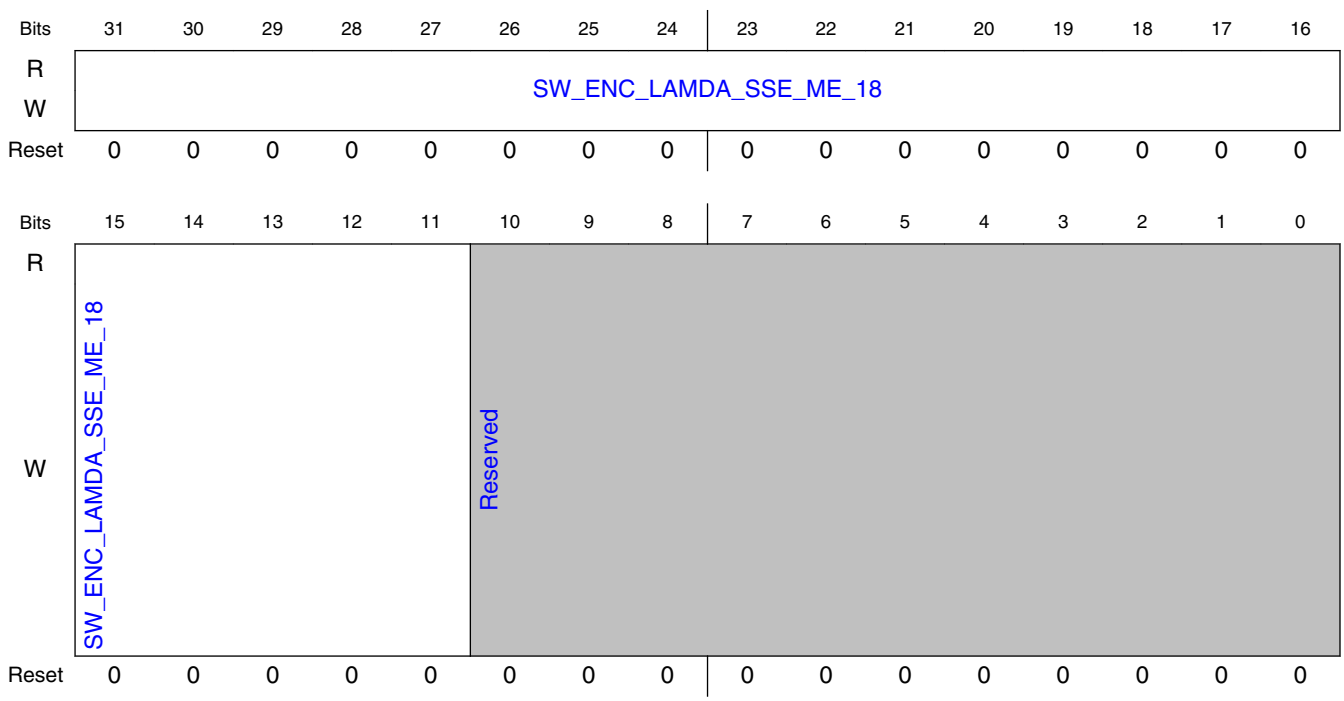
16.3.5.1.125.1 Offset

Register	Offset
swreg160	280h

16.3.5.1.125.2 Function

inter me SSE lambda config 18

16.3.5.1.125.3 Diagram



16.3.5.1.125.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_18	inter me SSE lambda 18
10-0 —	Reserved

16.3.5.1.126 inter me SSE lambda config 19 (swreg161)

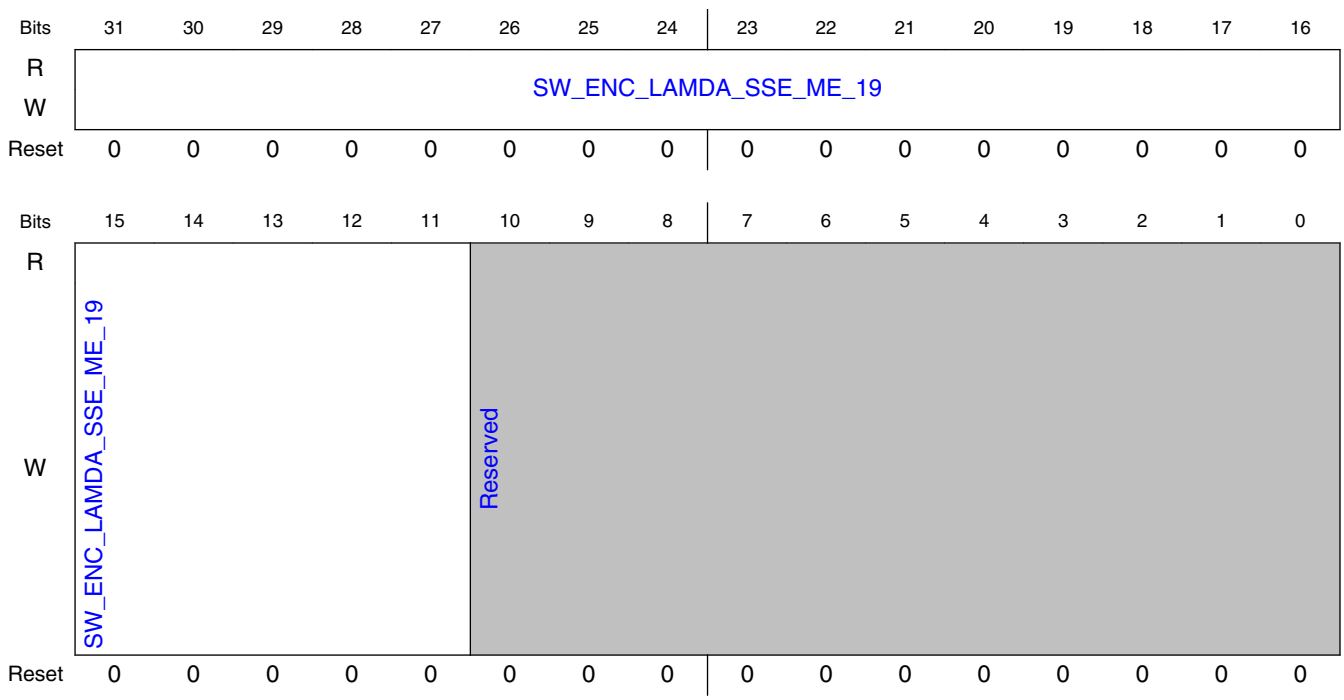
16.3.5.1.126.1 Offset

Register	Offset
swreg161	284h

16.3.5.1.126.2 Function

inter me SSE lambda config 19

16.3.5.1.126.3 Diagram



16.3.5.1.126.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_19	inter me SSE lambda 19

Table continues on the next page...

Field	Function
10-0 —	Reserved

16.3.5.1.127 inter me SSE lambda config 20 (swreg162)

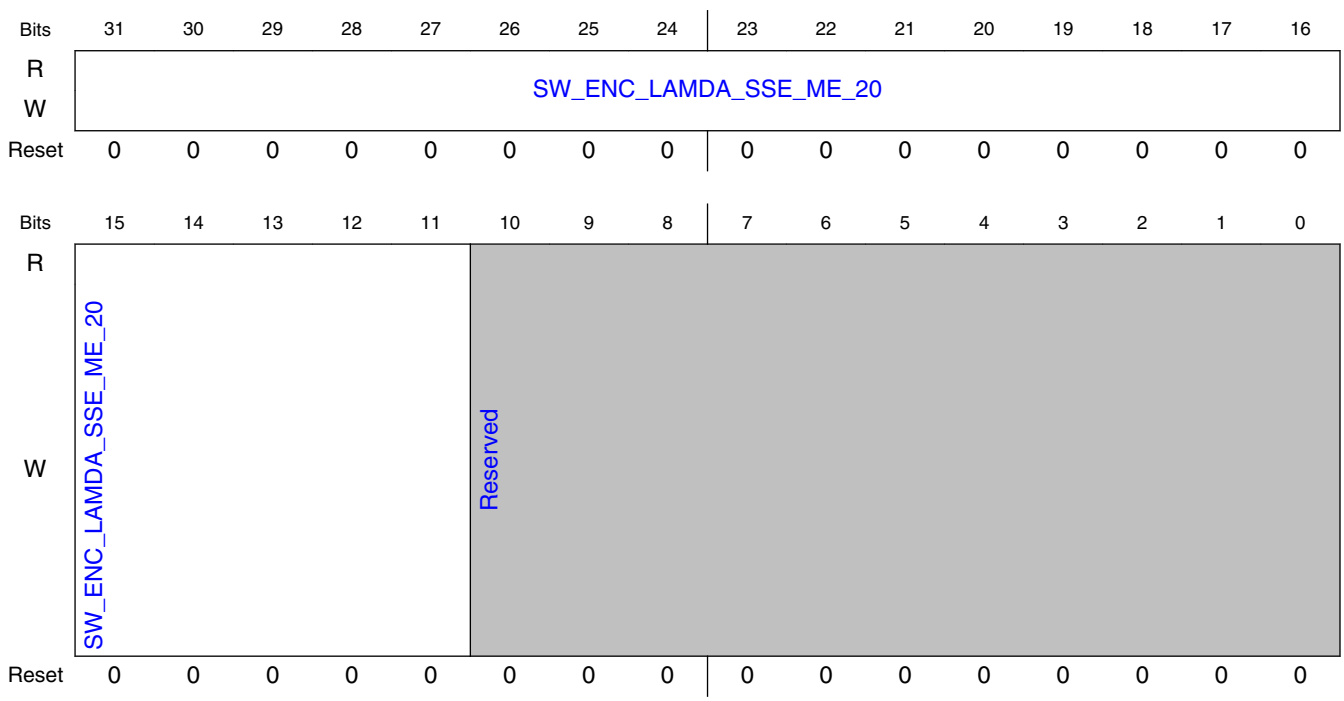
16.3.5.1.127.1 Offset

Register	Offset
swreg162	288h

16.3.5.1.127.2 Function

inter me SSE lambda config 20

16.3.5.1.127.3 Diagram



16.3.5.1.127.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_20	inter me SSE lambda 20
10-0 —	Reserved

16.3.5.1.128 inter me SSE lambda config 21 (swreg163)

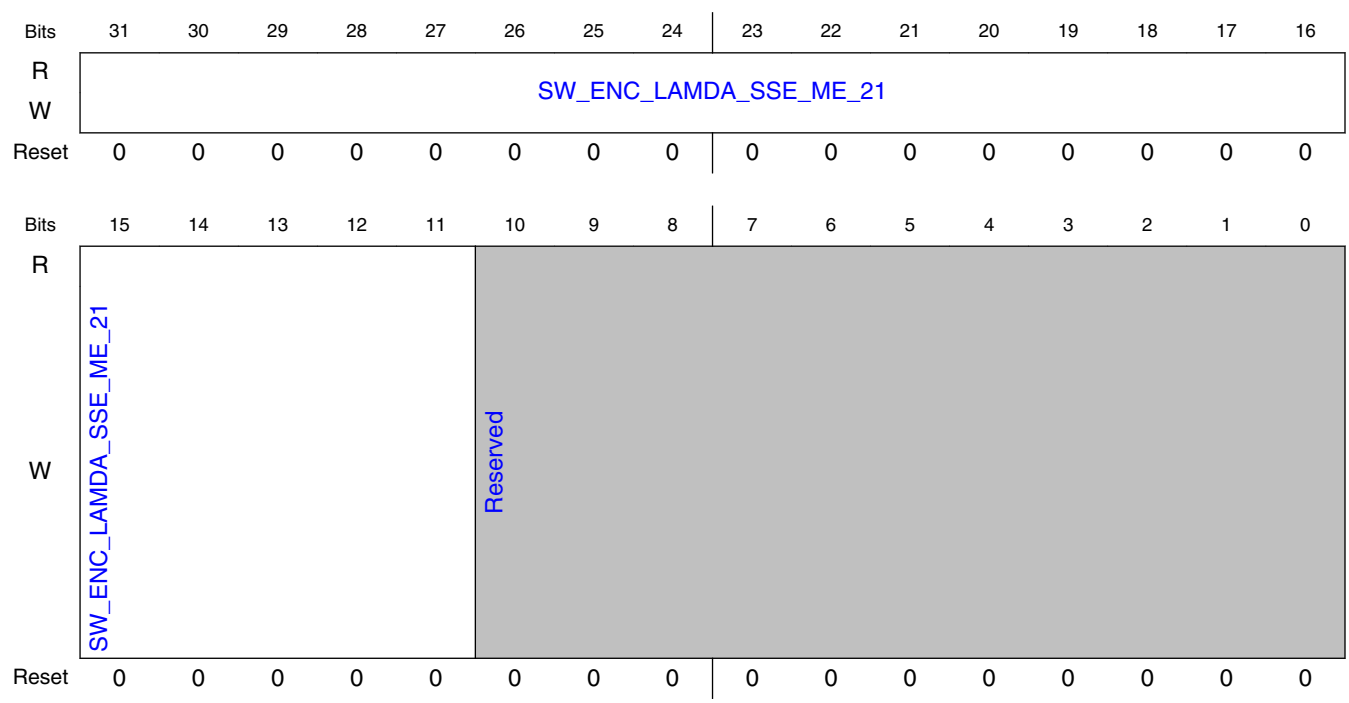
16.3.5.1.128.1 Offset

Register	Offset
swreg163	28Ch

16.3.5.1.128.2 Function

inter me SSE lambda config 21

16.3.5.1.128.3 Diagram



16.3.5.1.128.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_21	inter me SSE lambda 21
10-0 —	Reserved

16.3.5.1.129 inter me SSE lambda config 22 (swreg164)

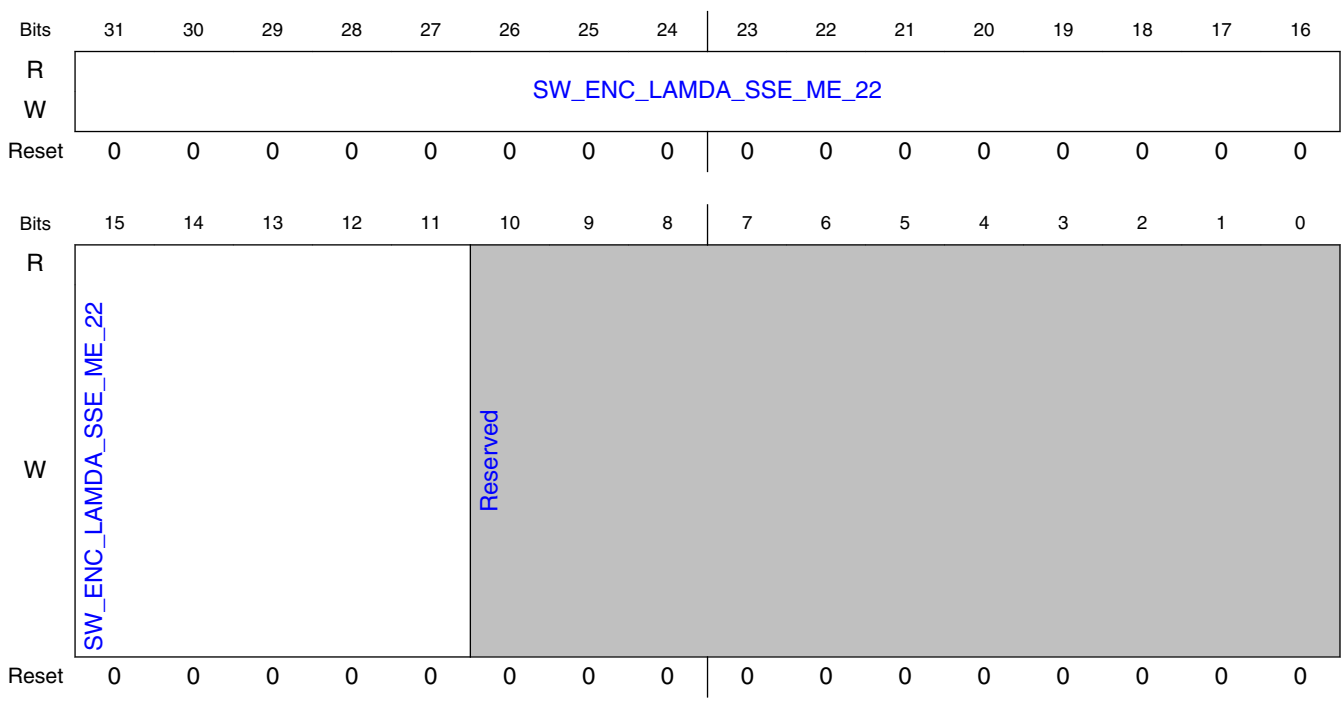
16.3.5.1.129.1 Offset

Register	Offset
swreg164	290h

16.3.5.1.129.2 Function

inter me SSE lambda config 22

16.3.5.1.129.3 Diagram



16.3.5.1.129.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_22	inter me SSE lambda 22
10-0 —	Reserved

16.3.5.1.130 inter me SSE lambda config 23 (swreg165)

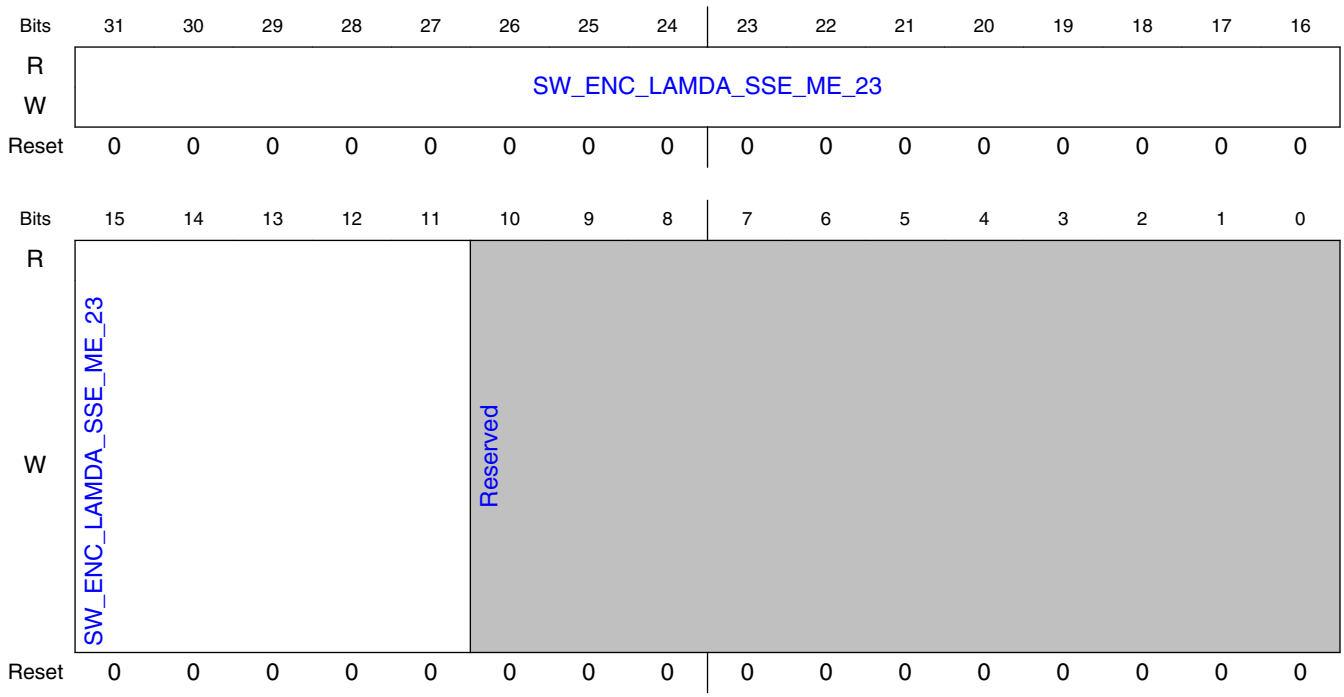
16.3.5.1.130.1 Offset

Register	Offset
swreg165	294h

16.3.5.1.130.2 Function

inter me SSE lambda config 23

16.3.5.1.130.3 Diagram



16.3.5.1.130.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_23	inter me SSE lambda 23
10-0 —	Reserved

16.3.5.1.131 inter me SSE lambda config 24 (swreg166)

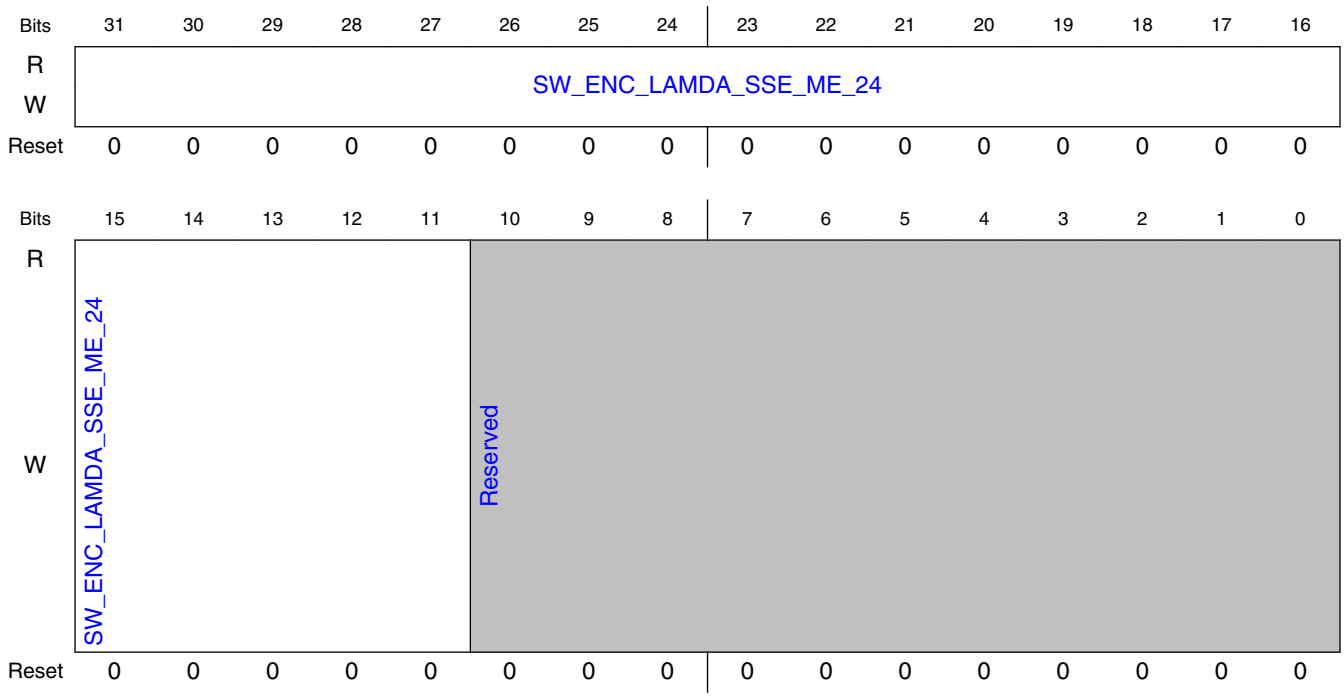
16.3.5.1.131.1 Offset

Register	Offset
swreg166	298h

16.3.5.1.131.2 Function

inter me SSE lambda config 24

16.3.5.1.131.3 Diagram



16.3.5.1.131.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_24	inter me SSE lambda 24
10-0 —	Reserved

16.3.5.1.132 inter me SSE lambda config 25 (swreg167)

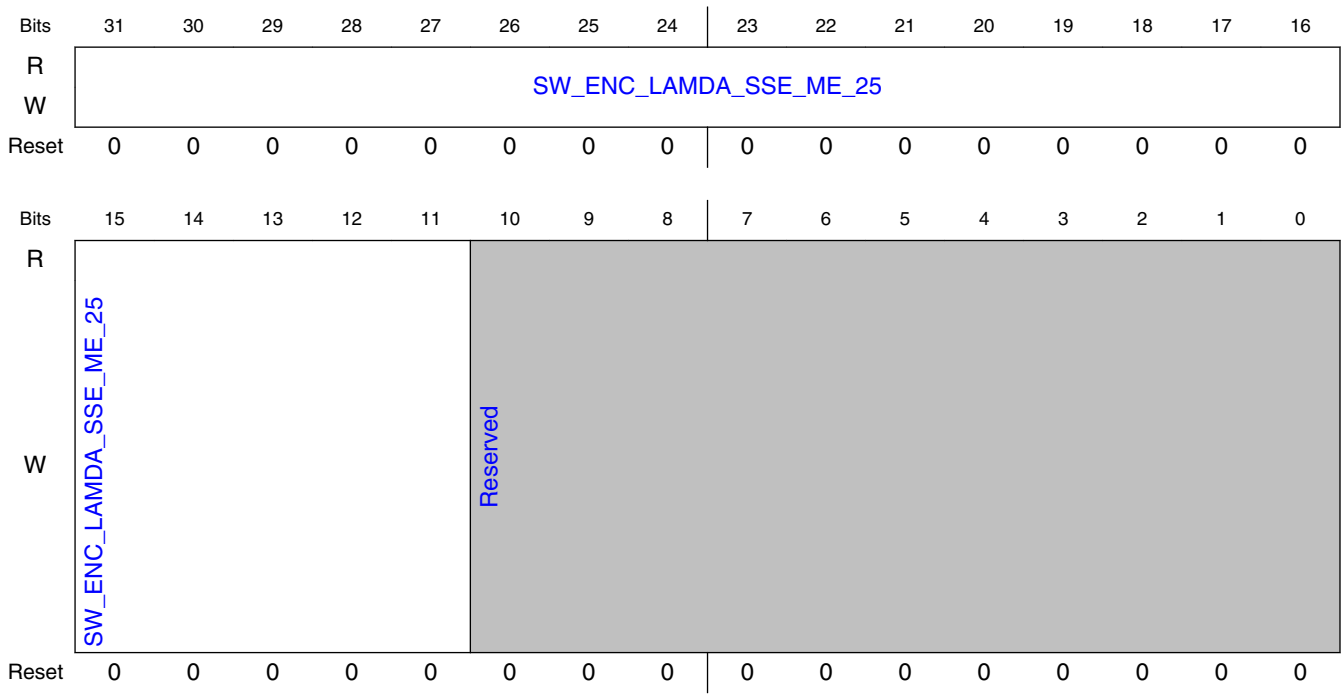
16.3.5.1.132.1 Offset

Register	Offset
swreg167	29Ch

16.3.5.1.132.2 Function

inter me SSE lambda config 25

16.3.5.1.132.3 Diagram



16.3.5.1.132.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_25	inter me SSE lambda 25
10-0 —	Reserved

16.3.5.1.133 inter me SSE lambda config 26 (swreg168)

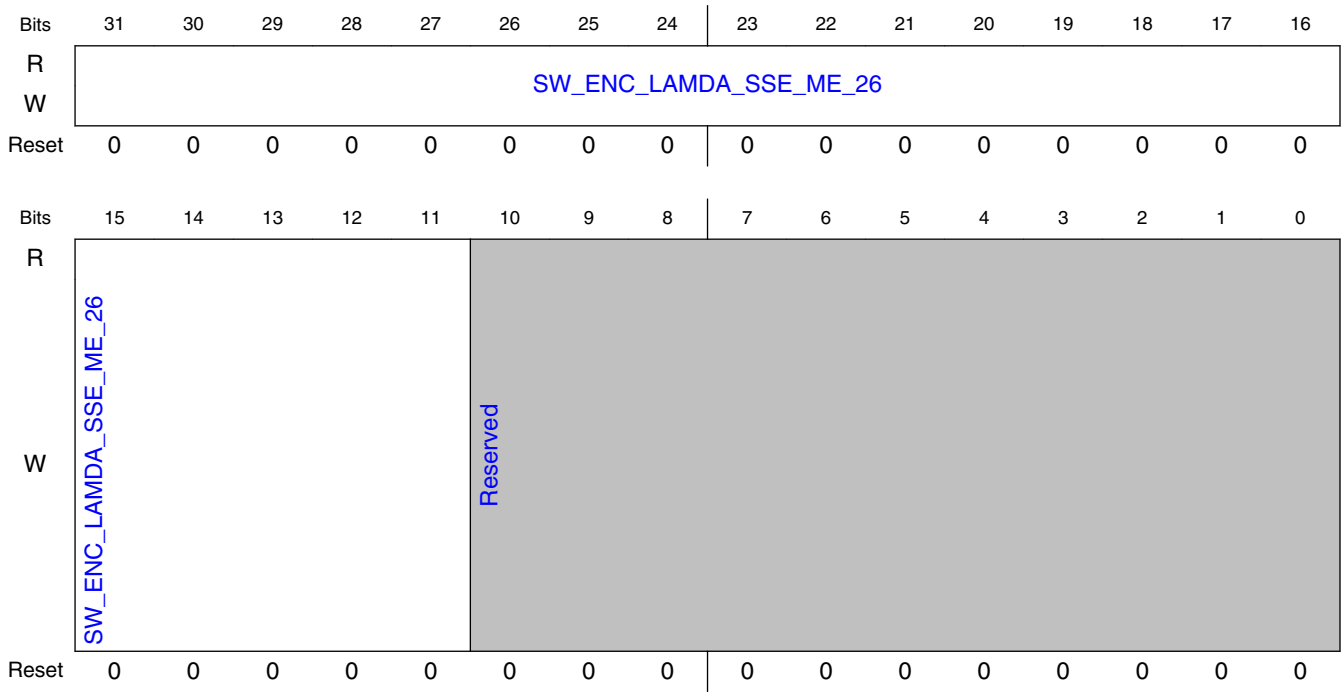
16.3.5.1.133.1 Offset

Register	Offset
swreg168	2A0h

16.3.5.1.133.2 Function

inter me SSE lambda config 26

16.3.5.1.133.3 Diagram



16.3.5.1.133.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_26	inter me SSE lambda 26
10-0 —	Reserved

16.3.5.1.134 inter me SSE lambda config 27 (swreg169)

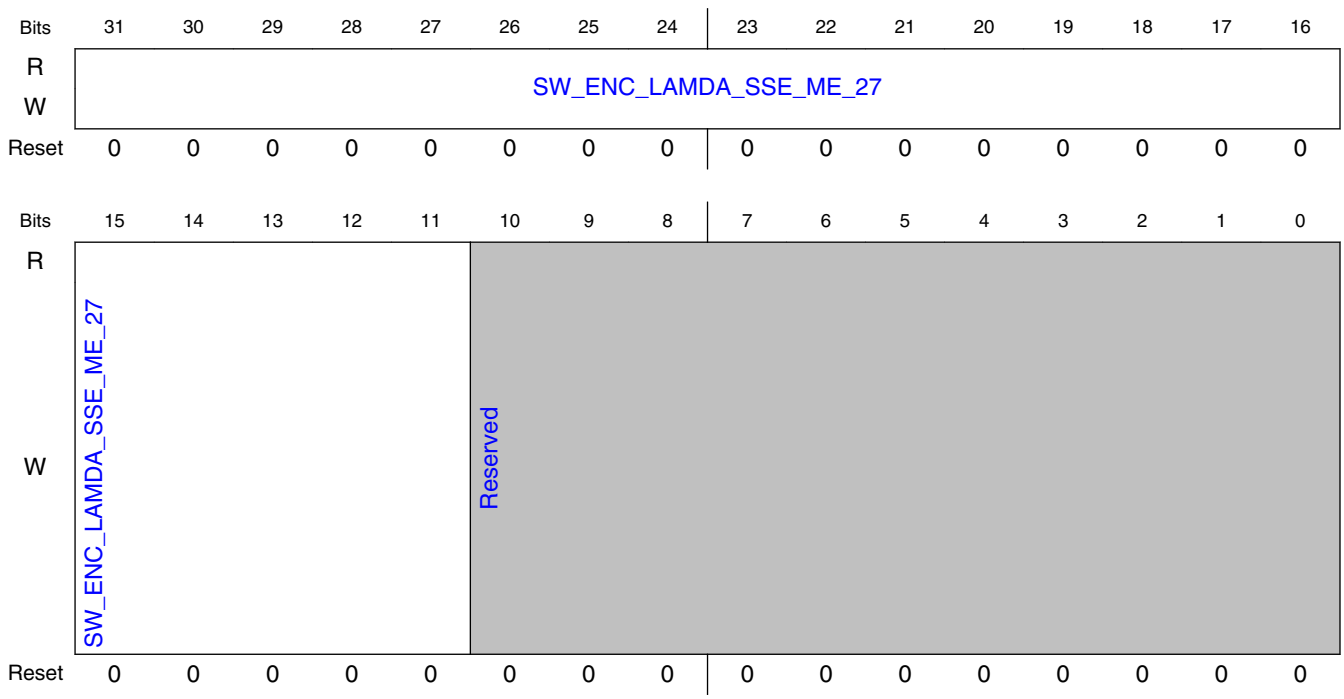
16.3.5.1.134.1 Offset

Register	Offset
swreg169	2A4h

16.3.5.1.134.2 Function

inter me SSE lambda config 27

16.3.5.1.134.3 Diagram



16.3.5.1.134.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_27	inter me SSE lambda 27
10-0 —	Reserved

16.3.5.1.135 inter me SSE lambda config 30 (swreg172)

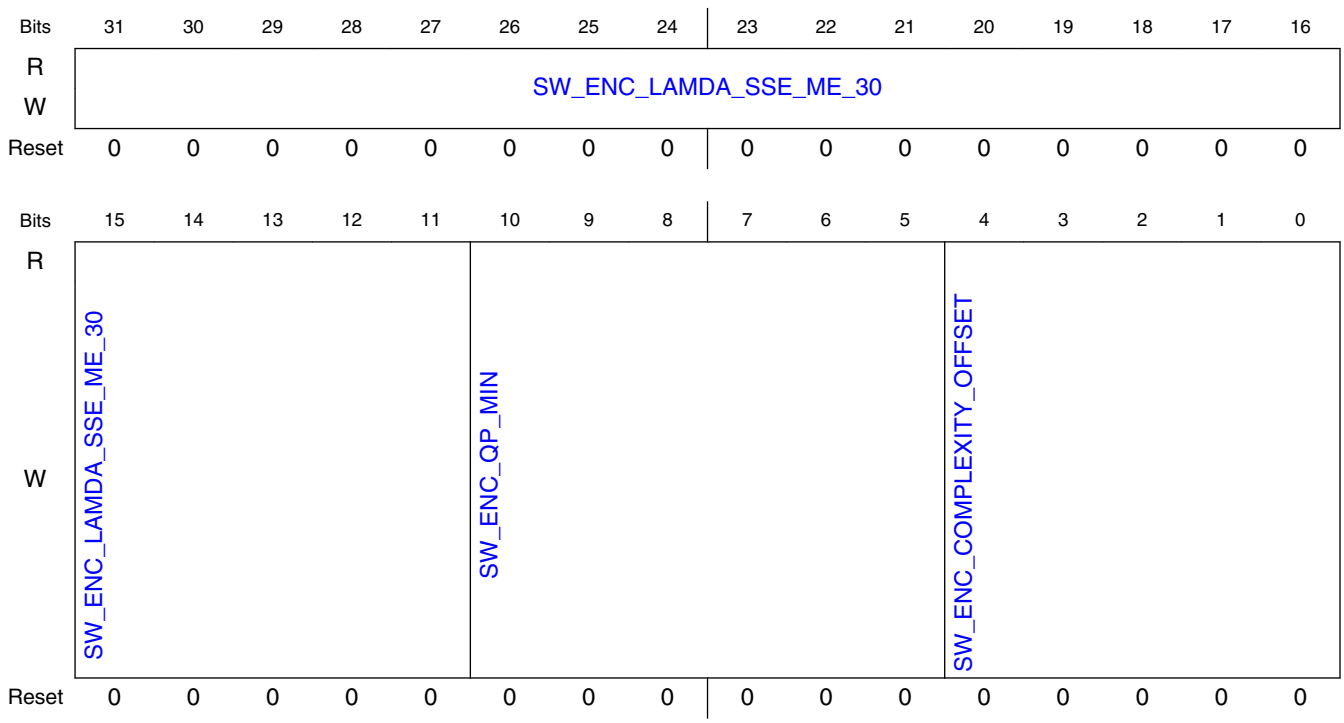
16.3.5.1.135.1 Offset

Register	Offset
swreg172	2B0h

16.3.5.1.135.2 Function

inter me SSE lambda config 30

16.3.5.1.135.3 Diagram



16.3.5.1.135.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_30	inter me SSE lambda 30
10-5 SW_ENC_QP_MIN	min value of qp
4-0	block rc complexity offset

Field	Function
SW_ENC_COMPLEXITY_OFFSET	

16.3.5.1.136 inter me SSE lambda config 31 (swreg173)

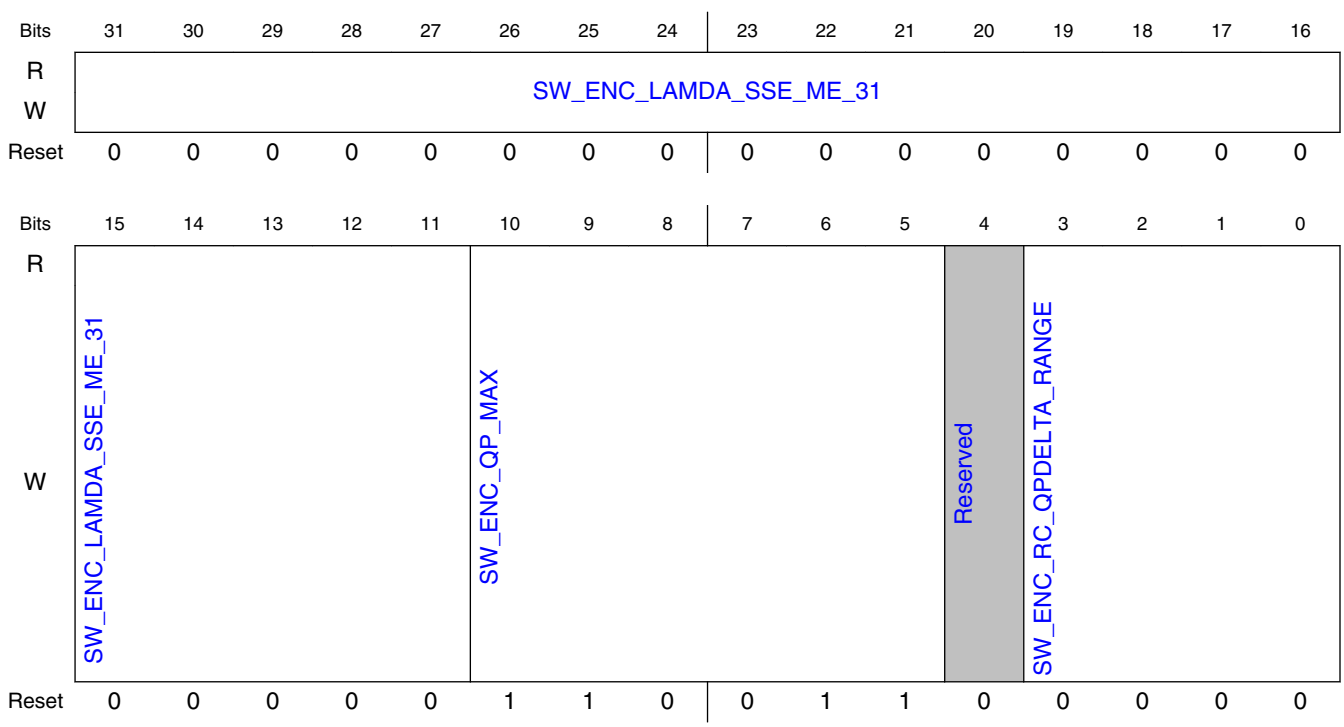
16.3.5.1.136.1 Offset

Register	Offset
swreg173	2B4h

16.3.5.1.136.2 Function

inter me SSE lambda config 31

16.3.5.1.136.3 Diagram



16.3.5.1.136.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_31	inter me SSE lambda 31
10-5 SW_ENC_QP_MAX	max value of qp
4 —	Reserved
3-0 SW_ENC_RC_QPDELTA_RANGE	rc qp delta range

16.3.5.1.137 intra SATD lambda config 8 (swreg174)

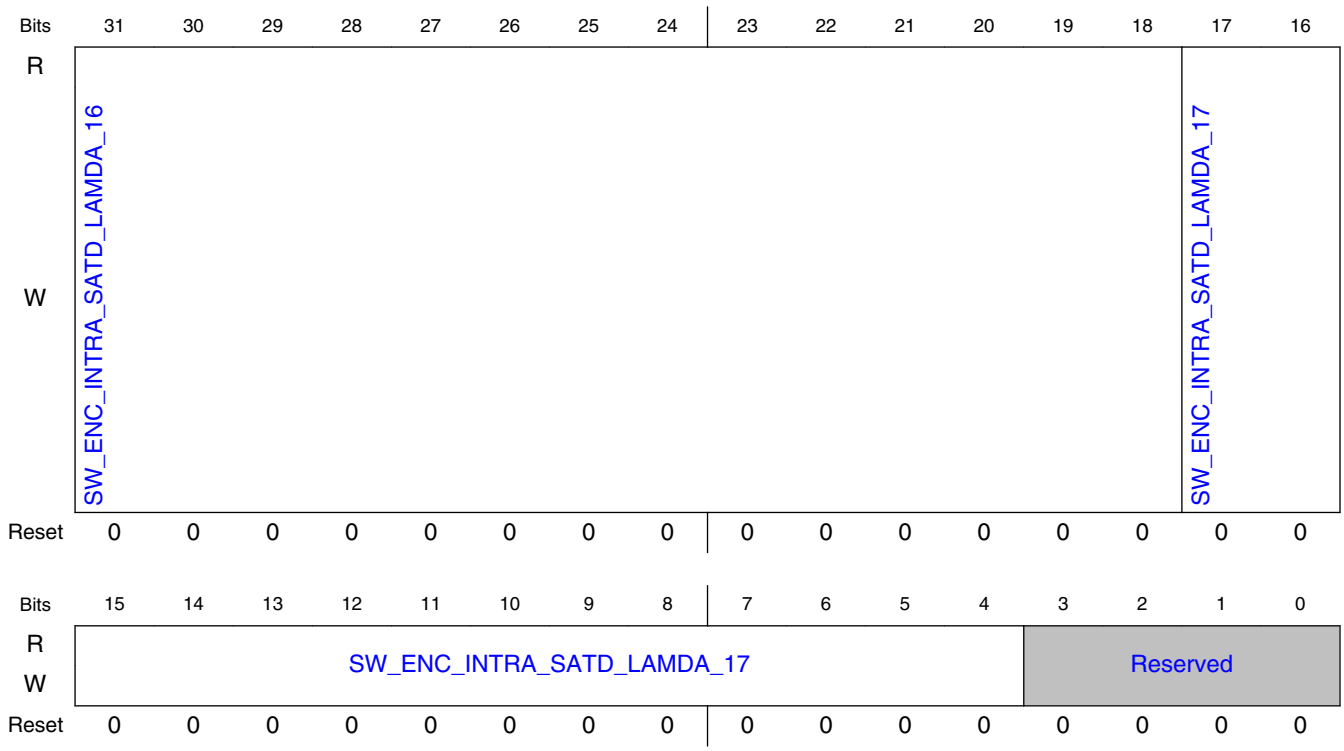
16.3.5.1.137.1 Offset

Register	Offset
swreg174	2B8h

16.3.5.1.137.2 Function

intra SATD lambda config 8

16.3.5.1.137.3 Diagram



16.3.5.1.137.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_16	intra SATD lambda 16
17-4 SW_ENC_INTRA_SATD_LAMDA_17	intra SATD lambda 17
3-0 —	Reserved

16.3.5.1.138 intra SATD lambda config 9 (swreg175)

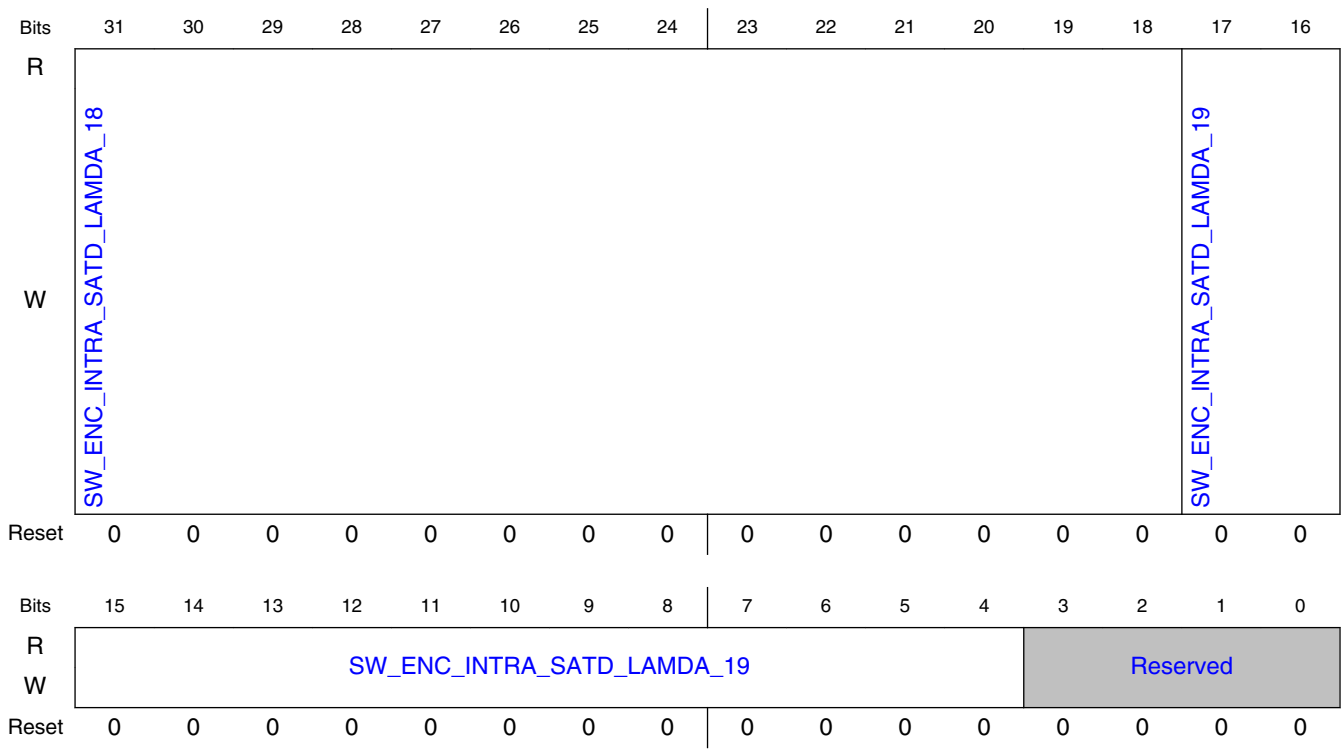
16.3.5.1.138.1 Offset

Register	Offset
swreg175	2BCh

16.3.5.1.138.2 Function

intra SATD lambda config 9

16.3.5.1.138.3 Diagram



16.3.5.1.138.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_18	intra SATD lambda 18
17-4 SW_ENC_INTRA_SATD_LAMDA_19	intra SATD lambda 19
3-0 —	Reserved

16.3.5.1.139 intra SATD lambda config 10 (swreg176)

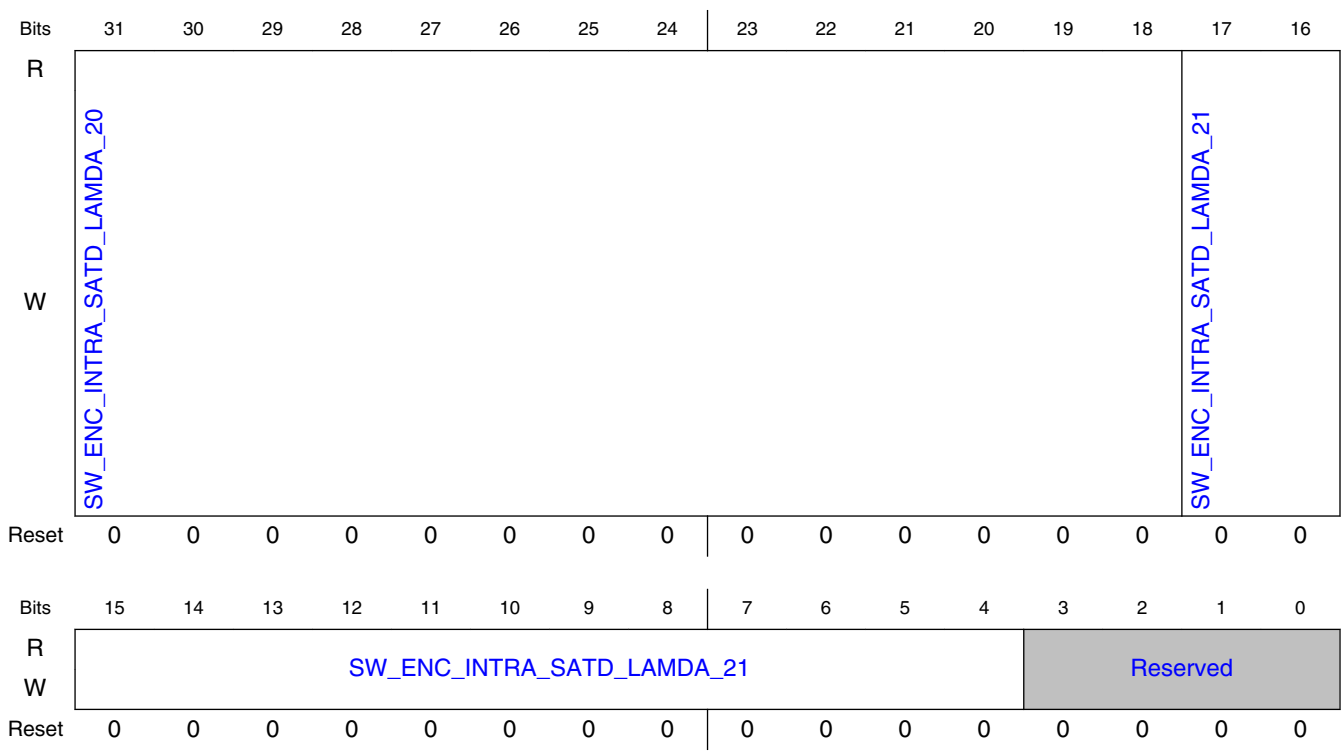
16.3.5.1.139.1 Offset

Register	Offset
swreg176	2C0h

16.3.5.1.139.2 Function

intra SATD lambda config 10

16.3.5.1.139.3 Diagram



16.3.5.1.139.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_20	intra SATD lambda 20
17-4	intra SATD lambda 21

Table continues on the next page...

Field	Function
SW_ENC_INTRA_SATD_LAMDA_21	
3-0 —	Reserved

16.3.5.1.140 intra SATD lambda config 11 (swreg177)

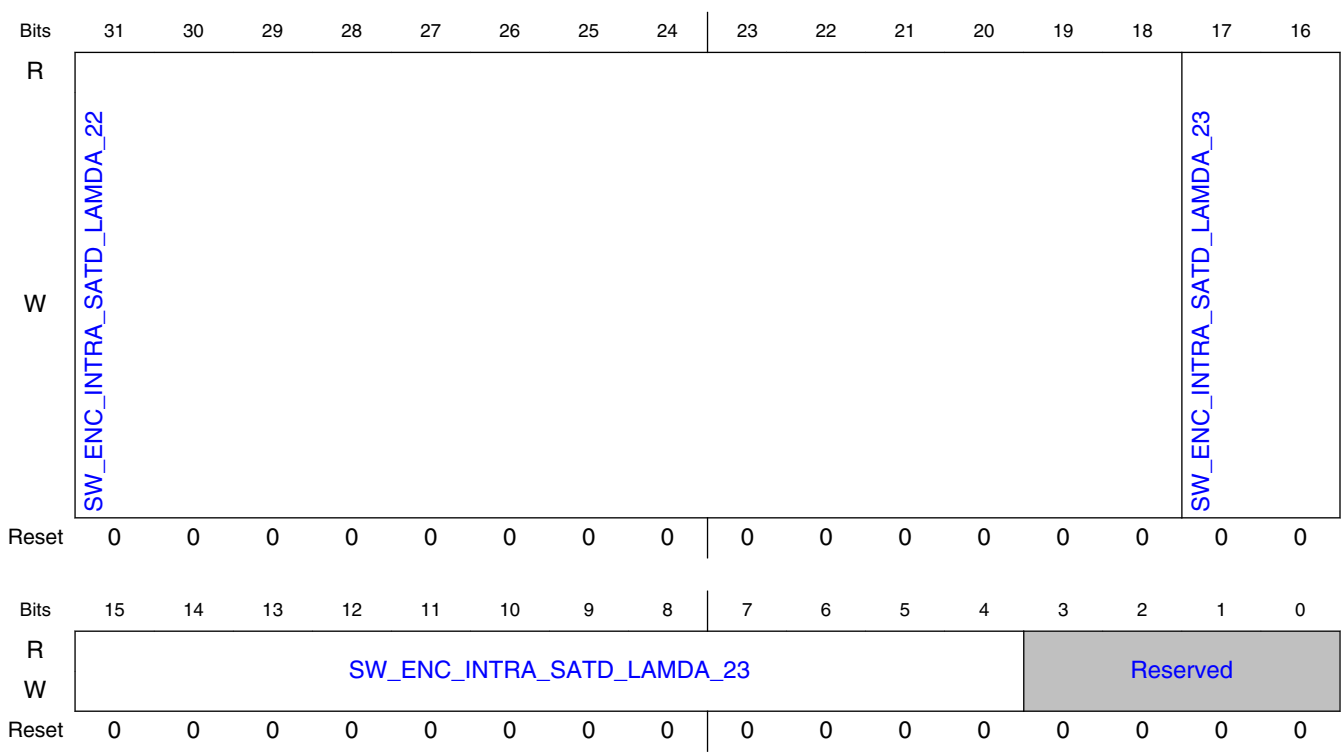
16.3.5.1.140.1 Offset

Register	Offset
swreg177	2C4h

16.3.5.1.140.2 Function

intra SATD lambda config 11

16.3.5.1.140.3 Diagram



16.3.5.1.140.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_22	intra SATD lambda 22
17-4 SW_ENC_INTRA_SATD_LAMDA_23	intra SATD lambda 23
3-0 —	Reserved

16.3.5.1.141 intra SATD lambda config 12 (swreg178)

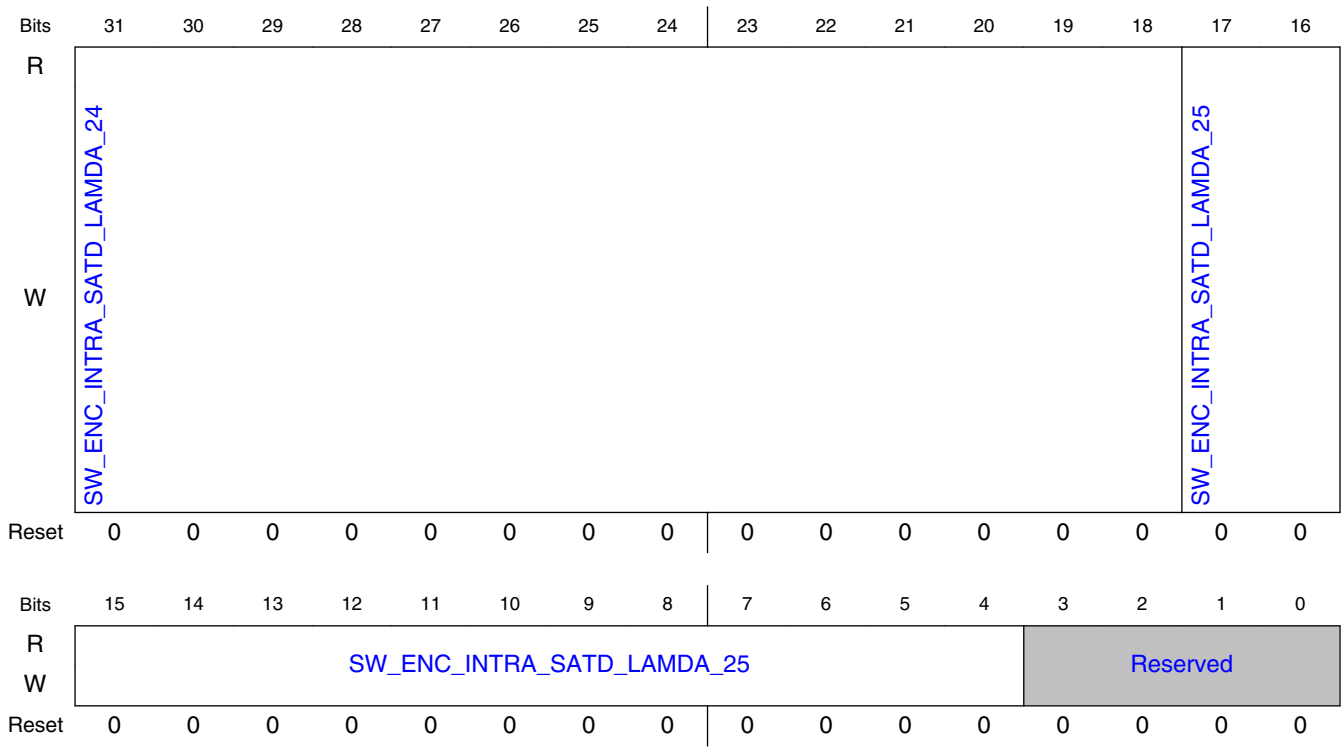
16.3.5.1.141.1 Offset

Register	Offset
swreg178	2C8h

16.3.5.1.141.2 Function

intra SATD lambda config 12

16.3.5.1.141.3 Diagram



16.3.5.1.141.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_24	intra SATD lambda 24
17-4 SW_ENC_INTRA_SATD_LAMDA_25	intra SATD lambda 25
3-0 —	Reserved

16.3.5.1.142 intra SATD lambda config 13 (swreg179)

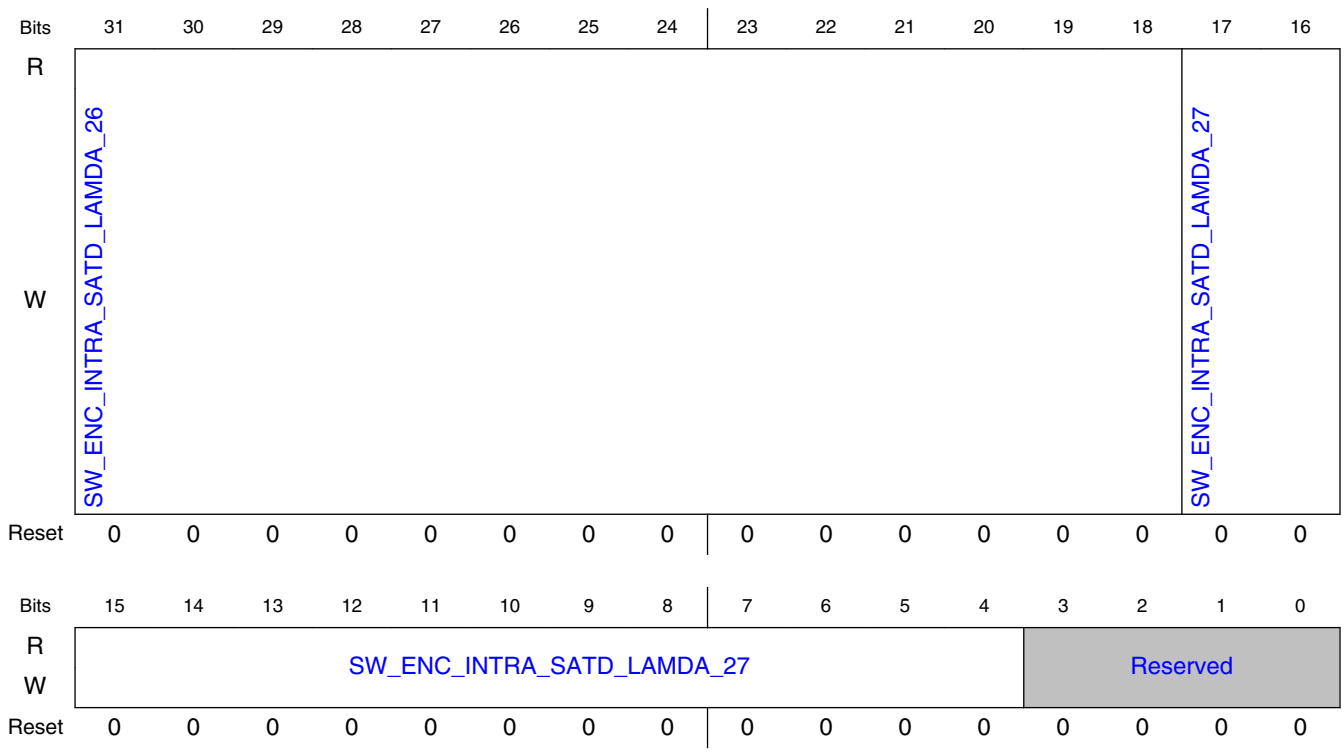
16.3.5.1.142.1 Offset

Register	Offset
swreg179	2CCh

16.3.5.1.142.2 Function

intra SATD lambda config 13

16.3.5.1.142.3 Diagram



16.3.5.1.142.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_26	intra SATD lambda 26
17-4 SW_ENC_INTRA_SATD_LAMDA_27	intra SATD lambda 27
3-0 —	Reserved

16.3.5.1.143 intra SATD lambda config 14 (swreg180)

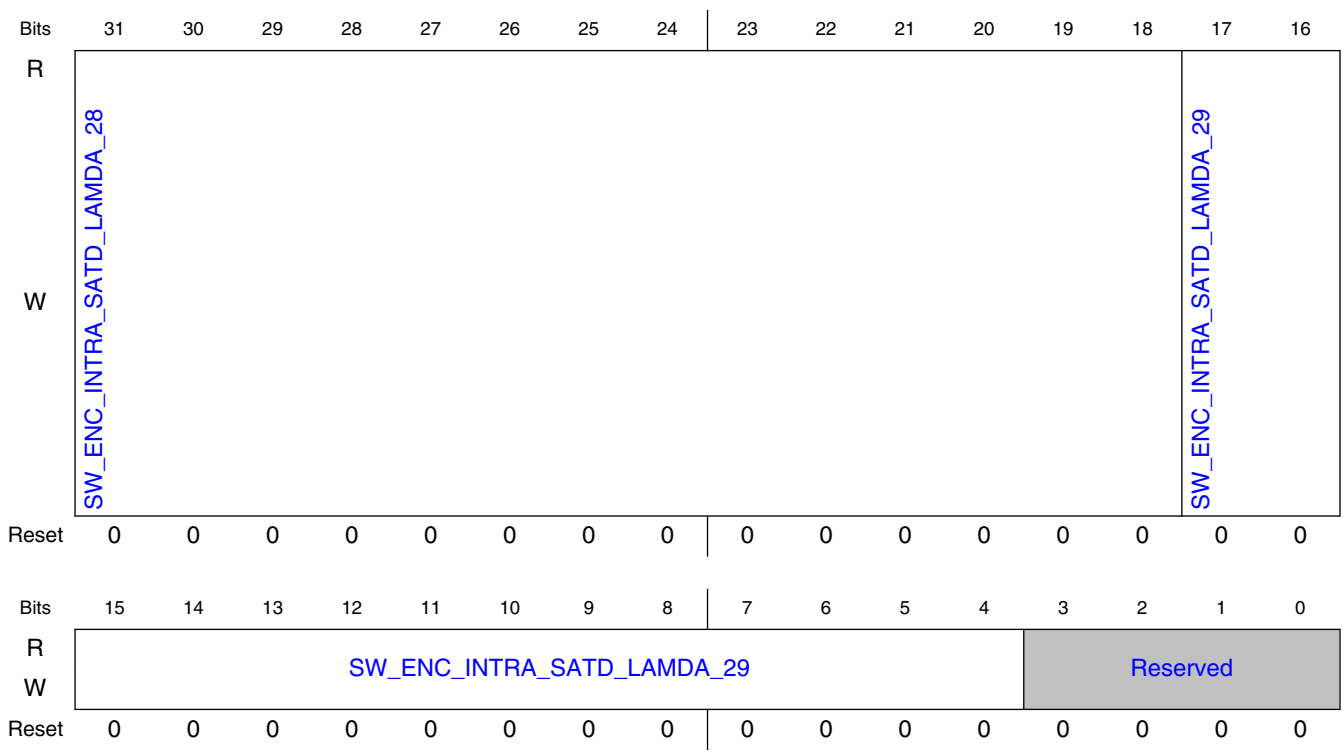
16.3.5.1.143.1 Offset

Register	Offset
swreg180	2D0h

16.3.5.1.143.2 Function

intra SATD lambda config 14

16.3.5.1.143.3 Diagram



16.3.5.1.143.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_28	intra SATD lambda 28
17-4	intra SATD lambda 29

Table continues on the next page...

Field	Function
SW_ENC_INTRA_SATD_LAMDA_29	
3-0 —	Reserved

16.3.5.1.144 intra SATD lambda config 15 (swreg181)

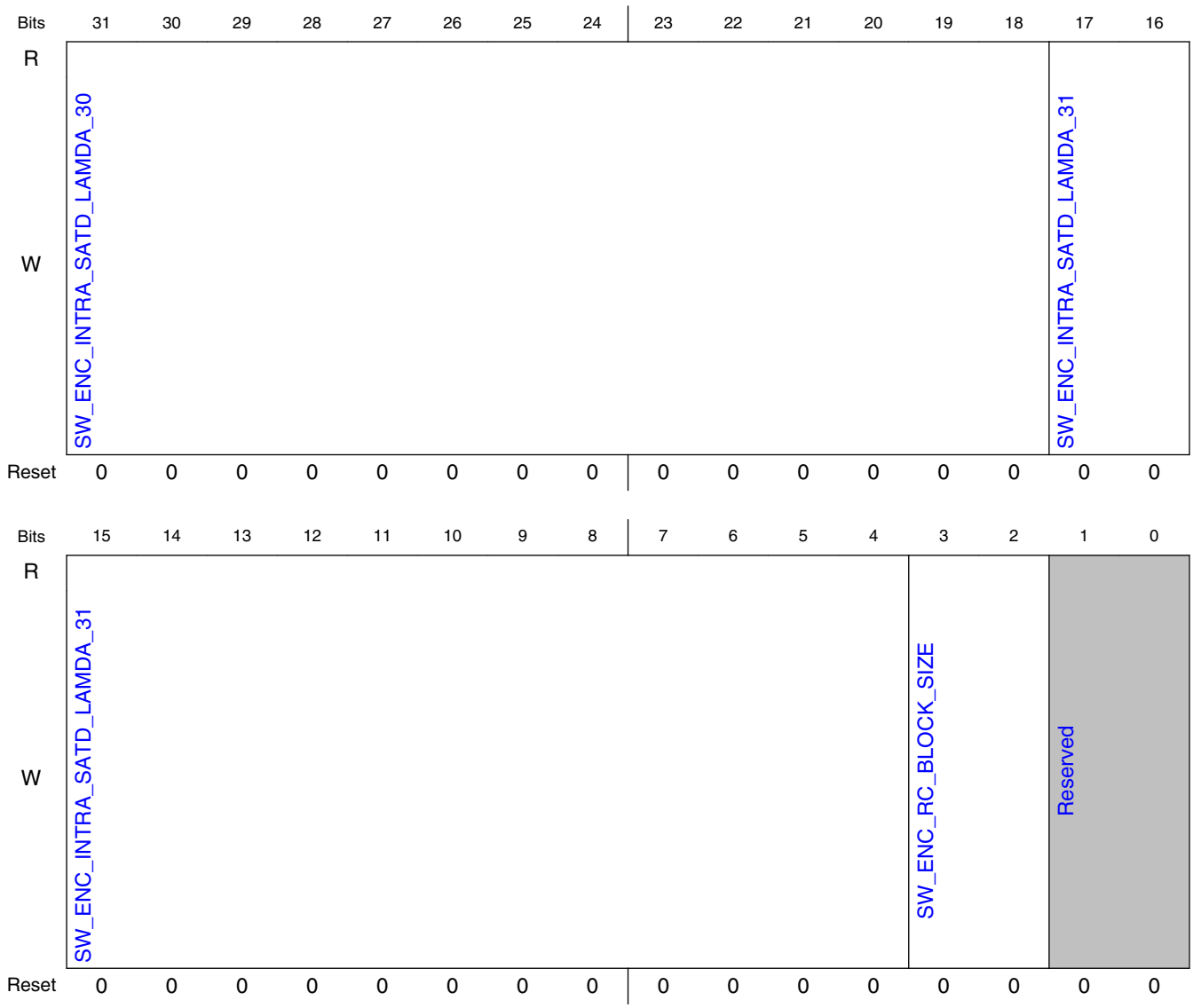
16.3.5.1.144.1 Offset

Register	Offset
swreg181	2D4h

16.3.5.1.144.2 Function

intra SATD lambda config 15

16.3.5.1.144.3 Diagram



16.3.5.1.144.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_30	intra SATD lambda 30
17-4 SW_ENC_INTRA_SATD_LAMDA_31	intra SATD lambda 31
3-2	block rc block size

Table continues on the next page...

Field	Function
SW_ENC_RC_BLOCK_SIZE	00b - 64x64. 01b - 32x32. 10b - 16x16
1-0 —	Reserved

16.3.5.1.145 qp fractional part (swreg182)

16.3.5.1.145.1 Offset

Register	Offset
swreg182	2D8h

16.3.5.1.145.2 Function

qp fractional part

16.3.5.1.145.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_QP_FRACTIONAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_QP_DELTA_GAIN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.145.4 Fields

Field	Function
31-16 SW_ENC_QP_FRACTIONAL	qp fractional part
15-0 SW_ENC_QP_DELTA_GAIN	qp delta gain

16.3.5.1.146 qp sum (swreg183)

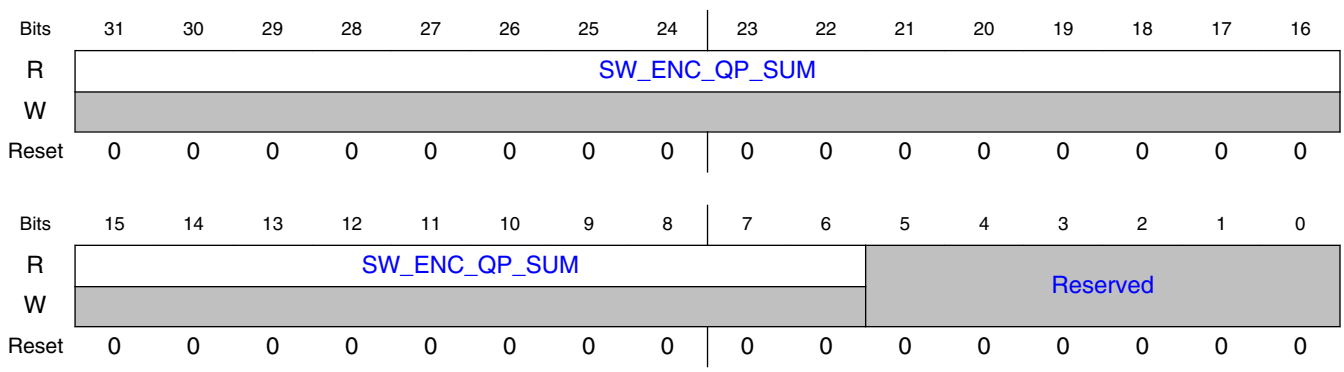
16.3.5.1.146.1 Offset

Register	Offset
swreg183	2DCh

16.3.5.1.146.2 Function

qp sum

16.3.5.1.146.3 Diagram



16.3.5.1.146.4 Fields

Field	Function
31-6 SW_ENC_QP_SUM	Block qp sum
5-0 —	Reserved

16.3.5.1.147 qp num (swreg184)

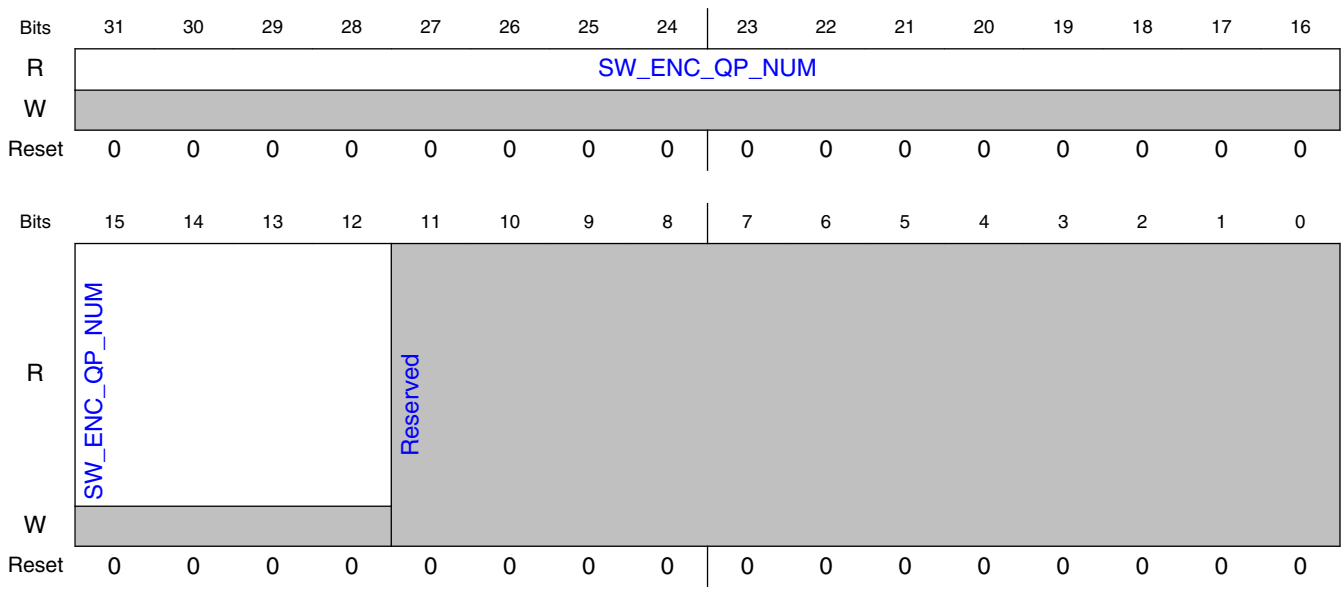
16.3.5.1.147.1 Offset

Register	Offset
swreg184	2E0h

16.3.5.1.147.2 Function

qp num

16.3.5.1.147.3 Diagram



16.3.5.1.147.4 Fields

Field	Function
31-12 SW_ENC_QP_NUM	Block qp number
11-0 —	Reserved

16.3.5.1.148 picture complexity. Timeout cycles MSB. (swreg185)

16.3.5.1.148.1 Offset

Register	Offset
swreg185	2E4h

16.3.5.1.148.2 Function

picture complexity. Timeout cycles MSB.

16.3.5.1.148.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_PIC_COMPLEXITY															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_PIC_COMPLEXITY								SW_TIMEOUT_CYCLES_MSB							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.148.4 Fields

Field	Function
31-9 SW_ENC_PIC_COMPLEXITY	Picture complexity
8-0 SW_TIMEOUT_CYCLES_MSB	Extend swreg81. Timeout cycles number. Bits 31:23.

16.3.5.1.149 Long-term reference pictures config (swreg190)

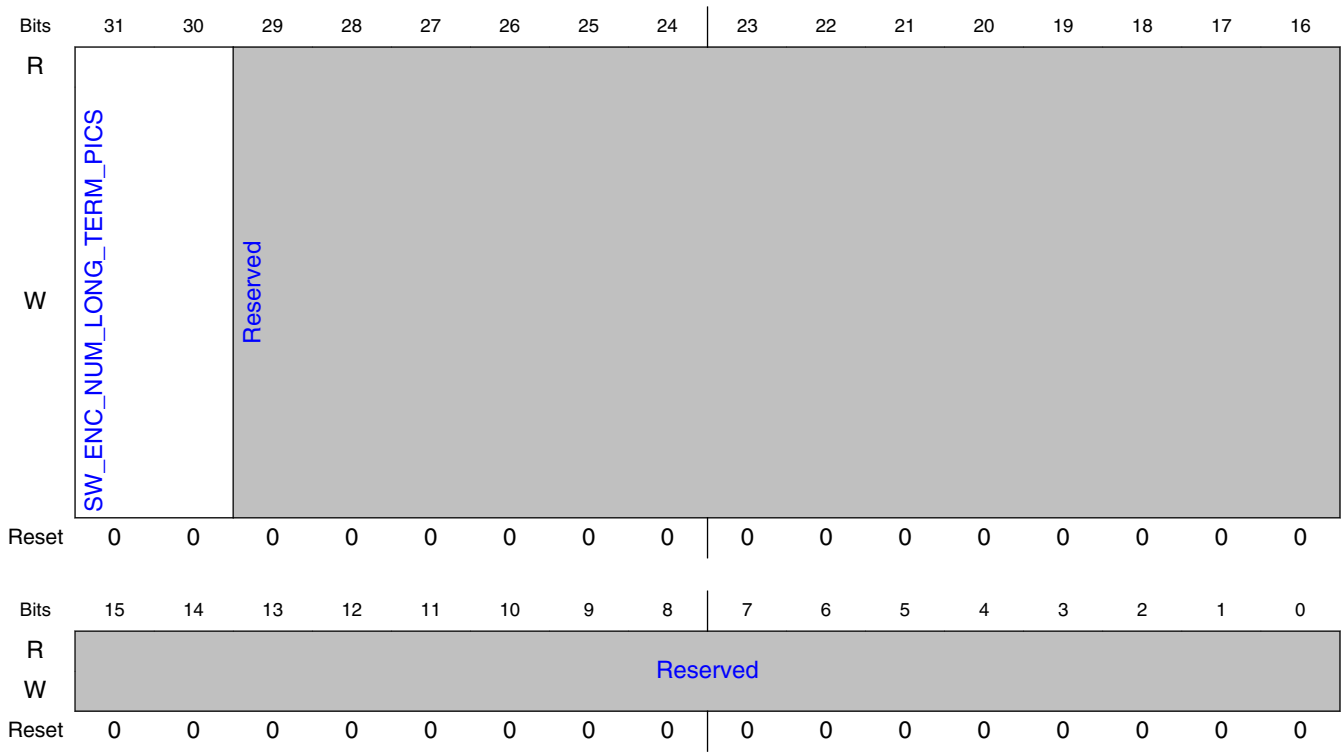
16.3.5.1.149.1 Offset

Register	Offset
swreg190	2F8h

16.3.5.1.149.2 Function

Long-term reference pictures config

16.3.5.1.149.3 Diagram



16.3.5.1.149.4 Fields

Field	Function
31-30 SW_ENC_NUM_LONG_TERM_PICS	Number of long-term reference pictures directly signaled in the slice header
29-0 —	Reserved

16.3.5.1.150 Temporal scalable config (swreg191)

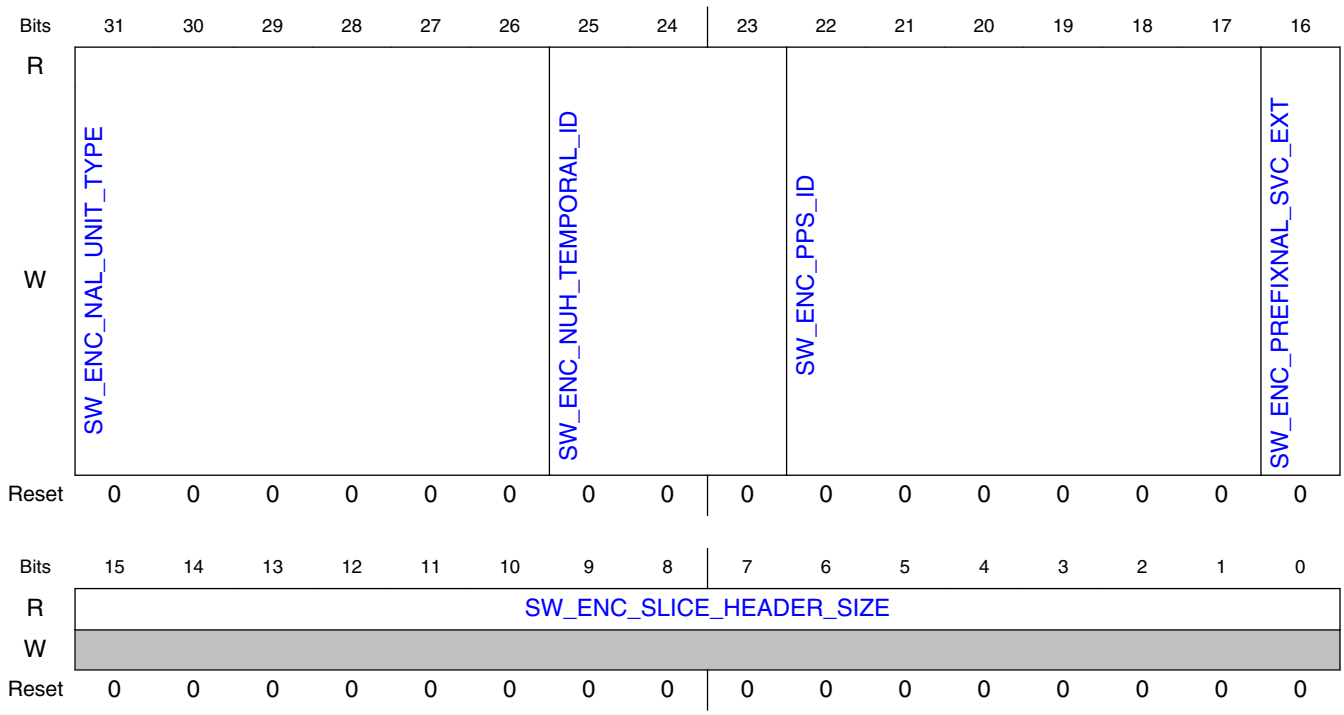
16.3.5.1.150.1 Offset

Register	Offset
swreg191	2FCh

16.3.5.1.150.2 Function

Temporal scalable config

16.3.5.1.150.3 Diagram



16.3.5.1.150.4 Fields

Field	Function
31-26 SW_ENC_NAL_UNIT_TYPE	NAL unit type
25-23 SW_ENC_NUH_TEMPORAL_ID	NAL temporal id
22-17 SW_ENC_PPS_ID	PPS id
16 SW_ENC_PREFIXNAL_SVC_EXT	H264 SVCT enable. 0b - disabled 1b - enabled (insert H264Scalability SEI).
15-0	Slice header size

Field	Function
SW_ENC_SLICE_HEADER_SIZE	

16.3.5.1.151 encoded Picture frame number (for H.264) (swreg192)

16.3.5.1.151.1 Offset

Register	Offset
swreg192	300h

16.3.5.1.151.2 Function

encoded Picture frame number (for H.264)

16.3.5.1.151.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_FRAMENUM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_FRAMENUM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.151.4 Fields

Field	Function
31-0 SW_ENC_FRAMENUM	encoded Picture frame number

16.3.5.1.152 reference pictures list0 config (for H.264) (swreg193)

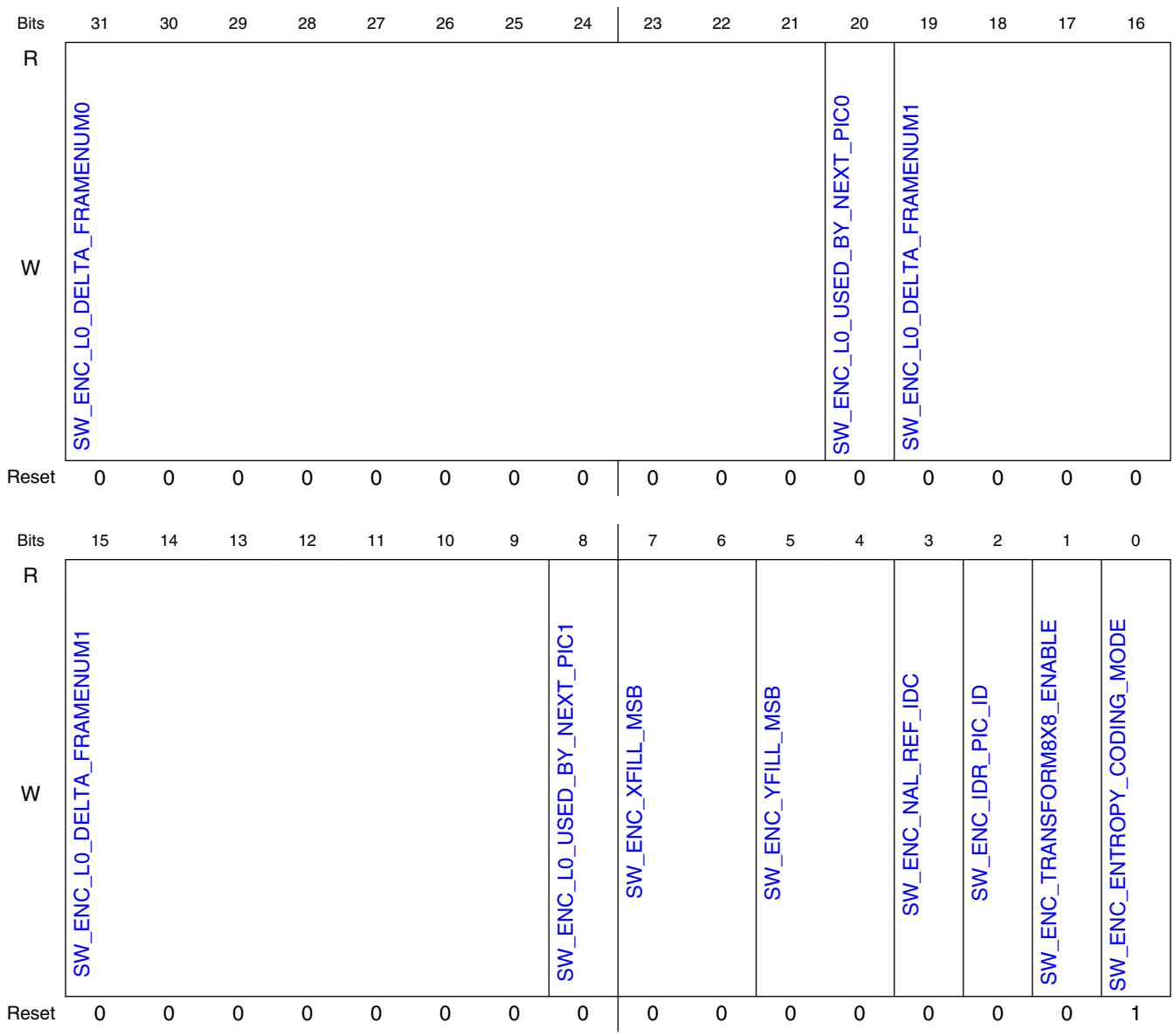
16.3.5.1.152.1 Offset

Register	Offset
swreg193	304h

16.3.5.1.152.2 Function

reference pictures list0 config (for H.264)

16.3.5.1.152.3 Diagram



16.3.5.1.152.4 Fields

Field	Function
31-21 SW_ENC_L0_DELTA_FRAMENUM0	delta frame_num of list0 pic0
20 SW_ENC_L0_USED_BY_NEXT_PIC0	list0 pic0 used by later frames
19-9 SW_ENC_L0_DELTA_FRAMENUM1	delta frame_num of list0 pic1
8 SW_ENC_L0_USED_BY_NEXT_PIC1	list0 pic1 used by later frames
7-6 SW_ENC_XFILL_MSB	Overfill pixels on right edge of image div2. [0.1.2.3] (bits[2:3] for H.264)
5-4 SW_ENC_YFILL_MSB	Overfill pixels on bottom edge of image. [0.1.2.3] (bits[3:4] for H.264)
3 SW_ENC_NAL_REF_IDC	current pic is used as reference
2 SW_ENC_IDR_PIC_ID	picture id for IDR frame
1 SW_ENC_TRANSFORM8X8_ENABLE	Enable transform8x8
0 SW_ENC_ENTROPY_CODING_MODE	Entropy coding mode flag. [0.1]

16.3.5.1.153 reference pictures list1 config (for H.264) (swreg194)

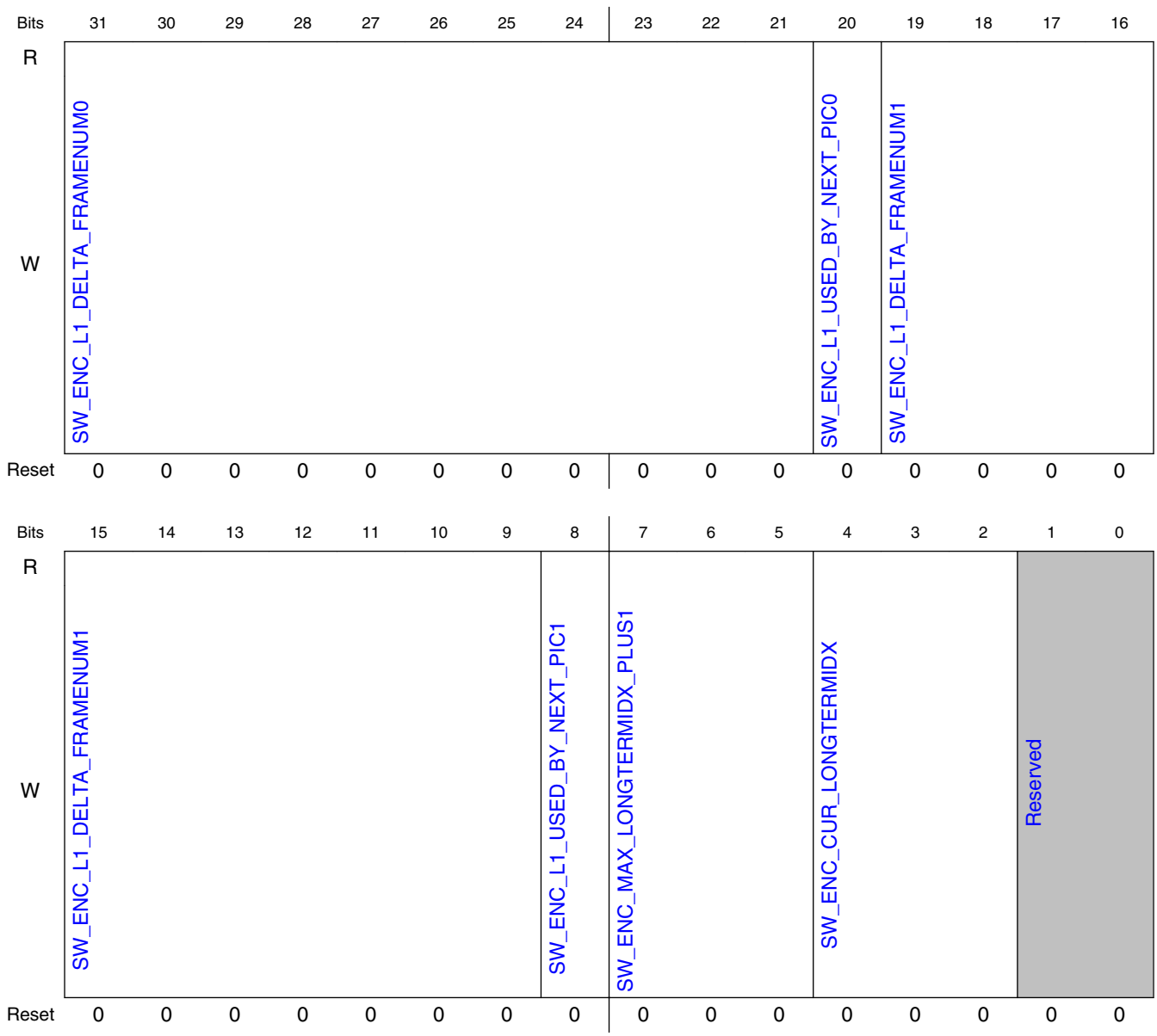
16.3.5.1.153.1 Offset

Register	Offset
swreg194	308h

16.3.5.1.153.2 Function

reference pictures list1 config (for H.264)

16.3.5.1.153.3 Diagram



16.3.5.1.153.4 Fields

Field	Function
31-21 SW_ENC_L1_DELTA_FRAMENUM0	delta frame_num of list1 pic0
20 SW_ENC_L1_USED_BY_NEXT_PIC0	list1 pic0 used by later frames
19-9 SW_ENC_L1_DELTA_FRAMENUM1	delta frame_num of list1 pic1
8 SW_ENC_L1_USED_BY_NEXT_PIC1	list1 pic1 used by later frames
7-5 SW_ENC_MAX_LONGTERMIDX_PLUS1	max number of long term frames (for H.264)
4-2 SW_ENC_CUR_LONGTERMIDX	long term frame idx of current frame (for H.264)
1-0 —	Reserved

16.3.5.1.154 register extension for ctu_size=16 (swreg195)

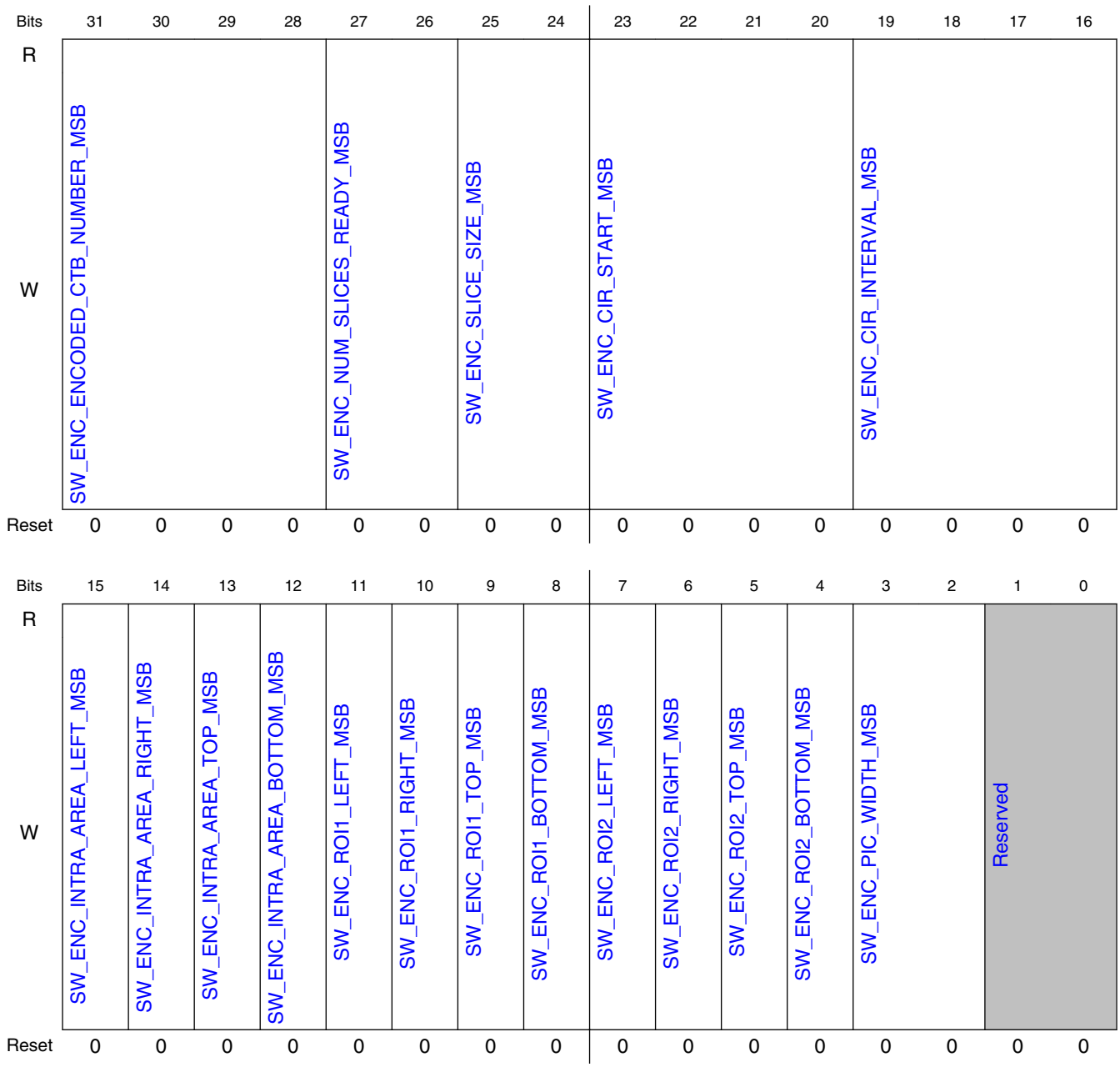
16.3.5.1.154.1 Offset

Register	Offset
swreg195	30Ch

16.3.5.1.154.2 Function

register extension for ctu_size=16

16.3.5.1.154.3 Diagram



16.3.5.1.154.4 Fields

Field	Function
31-28 SW_ENC_ENCODED_CTB_NUMBER_MSB	MB count output. Max 512*256. bit[13:16]
27-26	HEVC amount of completed slices. bit[8:9]

Table continues on the next page...

Field	Function
SW_ENC_NUM_SLICES_READY_MSB	
25-24 SW_ENC_SLICE_SIZE_MSB	slice size in ctu row. bit[7:8]
23-20 SW_ENC_CIR_START_MSB	cir start. bit[14:17]
19-16 SW_ENC_CIR_INTERVAL_MSB	Cyclic Intra Refresh. bit[14:17]
15 SW_ENC_INTRA_AREA_LEFT_MSB	intra Area Left. bit[8]
14 SW_ENC_INTRA_AREA_RIGHT_MSB	intra Area Right. bit[8]
13 SW_ENC_INTRA_AREA_TOP_MSB	intra Area Top. bit[8]
12 SW_ENC_INTRA_AREA_BOTTOM_MSB	intra Area Bottom. bit[8]
11 SW_ENC_ROI1_LEFT_MSB	ROI1 Area Left. bit[8]
10 SW_ENC_ROI1_RIGHT_MSB	ROI1 Area Right. bit[8]
9 SW_ENC_ROI1_TOP_MSB	ROI1 Area Top. bit[8]
8 SW_ENC_ROI1_BOTTOM_MSB	ROI1 Area Bottom. bit[8]
7 SW_ENC_ROI2_LEFT_MSB	ROI2 Area Left. bit[8]
6	ROI2 Area Right. bit[8]

Table continues on the next page...

Field	Function
SW_ENC_ROI2_RIGHT_MSB	
5 SW_ENC_ROI2_TOP_MSB	ROI2 Area Top. bit[8]
4 SW_ENC_ROI2_BOTTOM_MSB	ROI2 Area Bottom. bit[8]
3-2 SW_ENC_PIC_WIDTH_MSB	Encoded width. lumWidthMsb(unit 8 pixels). bit[11:10]
1-0 —	Reserved

16.3.5.1.155 Low Latency Controls (swreg196)

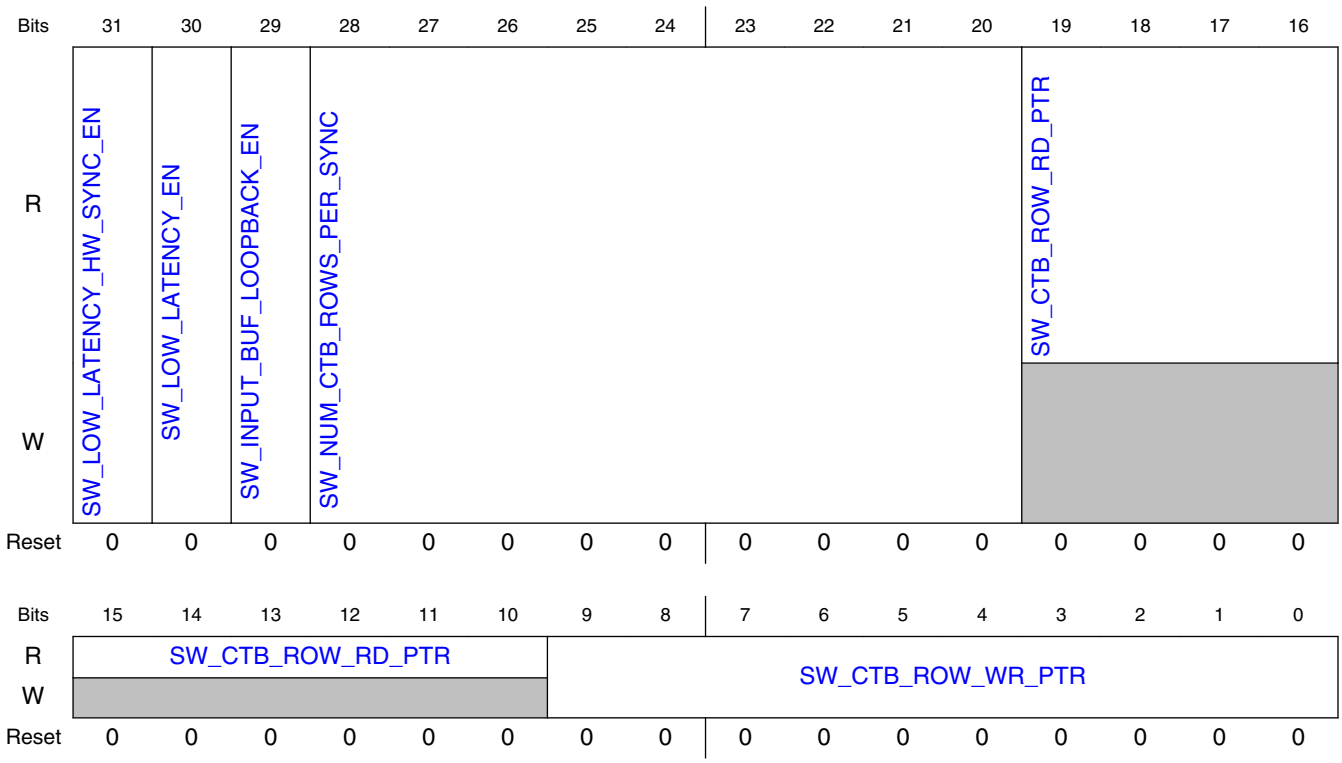
16.3.5.1.155.1 Offset

Register	Offset
swreg196	310h

16.3.5.1.155.2 Function

Low Latency Controls

16.3.5.1.155.3 Diagram



16.3.5.1.155.4 Fields

Field	Function
31 SW_LOW_LATENCY_HW_SYNC_EN	Low Latency Hardware Interface Enable. Use hardware handshaking.
30 SW_LOW_LATENCY_EN	Low Latency Enable
29 SW_INPUT_BUF_LOOPBACK_EN	Input buffer loopback Enable
28-20 SW_NUM_CTB_ROWS_PER_SYNC	Number of CTB rows for every HW sync. Unit is CTB
19-10 SW_CTB_ROW_RD_PTR	The number of CTB rows that has been fetched from Input buffer by encoder
9-0	The number of CTB rows that has been filled into the input buffer by external application

Field	Function
SW_CTB_ROW_WR_PTR	

16.3.5.1.156 Delta POC extension (swreg197)

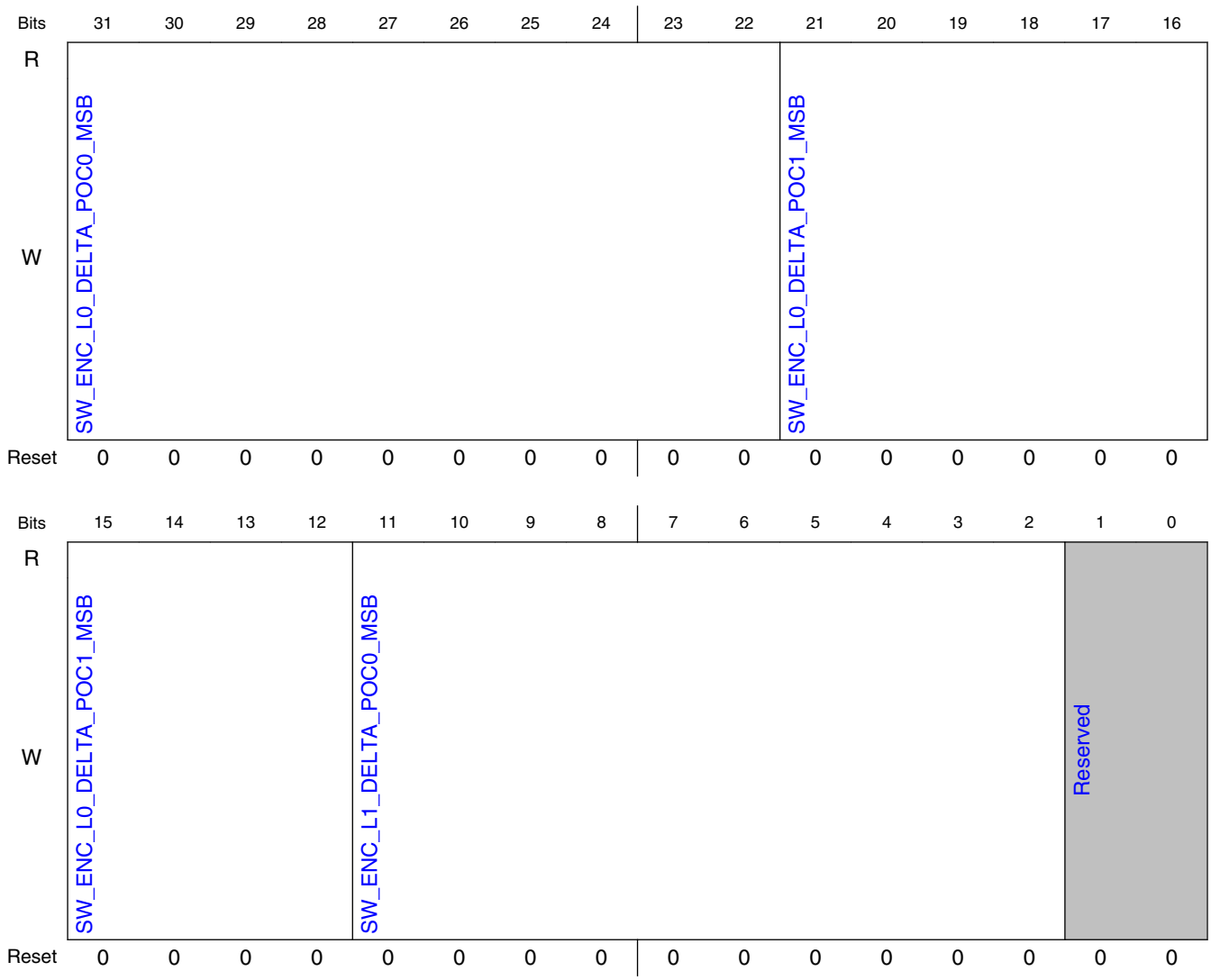
16.3.5.1.156.1 Offset

Register	Offset
swreg197	314h

16.3.5.1.156.2 Function

Delta POC extension

16.3.5.1.156.3 Diagram



16.3.5.1.156.4 Fields

Field	Function
31-22 SW_ENC_L0_DELTA_POC0_MSB	delta poc of list0 pic0 (bit[19..10])
21-12 SW_ENC_L0_DELTA_POC1_MSB	delta poc of list0 pic1 (bit[19..10])
11-2	delta poc of list1 pic0 (bit[19..10])

Table continues on the next page...

Field	Function
SW_ENC_L1_DELTA_POC0_MSB	
1-0 —	Reserved

16.3.5.1.157 Long Term Reference Control (swreg198)

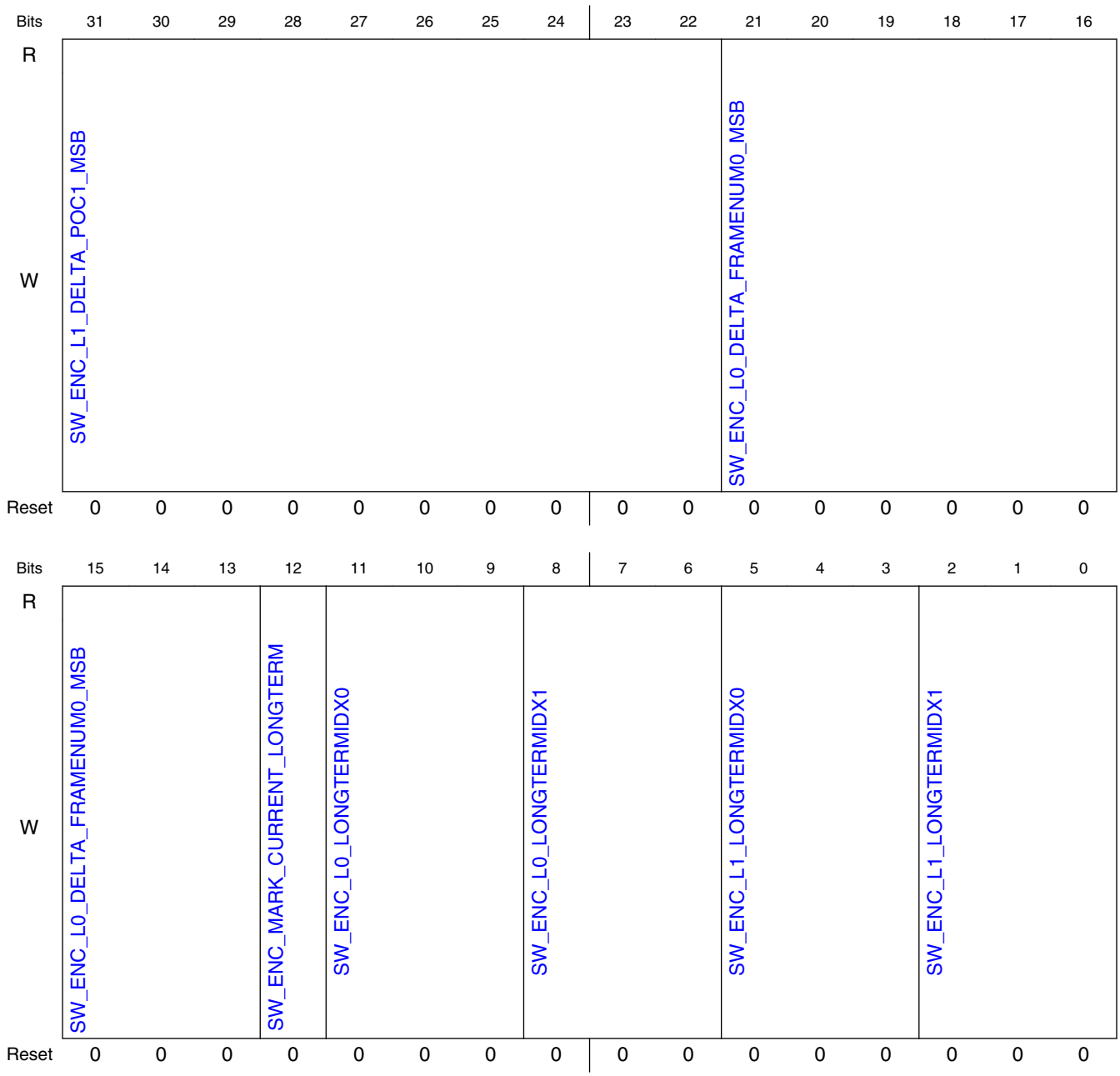
16.3.5.1.157.1 Offset

Register	Offset
swreg198	318h

16.3.5.1.157.2 Function

Long Term Reference Control

16.3.5.1.157.3 Diagram



16.3.5.1.157.4 Fields

Field	Function
31-22 SW_ENC_L1_DELTA_POC1_MSB	delta poc of list1 pic1 (bit[19..10])

Table continues on the next page...

Field	Function
21-13 SW_ENC_L0_DELTA_FRAMENUM0_MSB	delta frame_num of list0 pic0 (bit[19..11])
12 SW_ENC_MARK_CURRENT_LONGTERM	mark current frame as long term reference (for H.264)
11-9 SW_ENC_L0_LONGTERMIDX0	long term frame idx of list0 pic 0 (for H.264)
8-6 SW_ENC_L0_LONGTERMIDX1	long term frame idx of list0 pic 1 (for H.264)
5-3 SW_ENC_L1_LONGTERMIDX0	long term frame idx of list1 pic 0 (for H.264)
2-0 SW_ENC_L1_LONGTERMIDX1	long term frame idx of list1 pic 1 (for H.264)

16.3.5.1.158 Hash Code Control (swreg199)

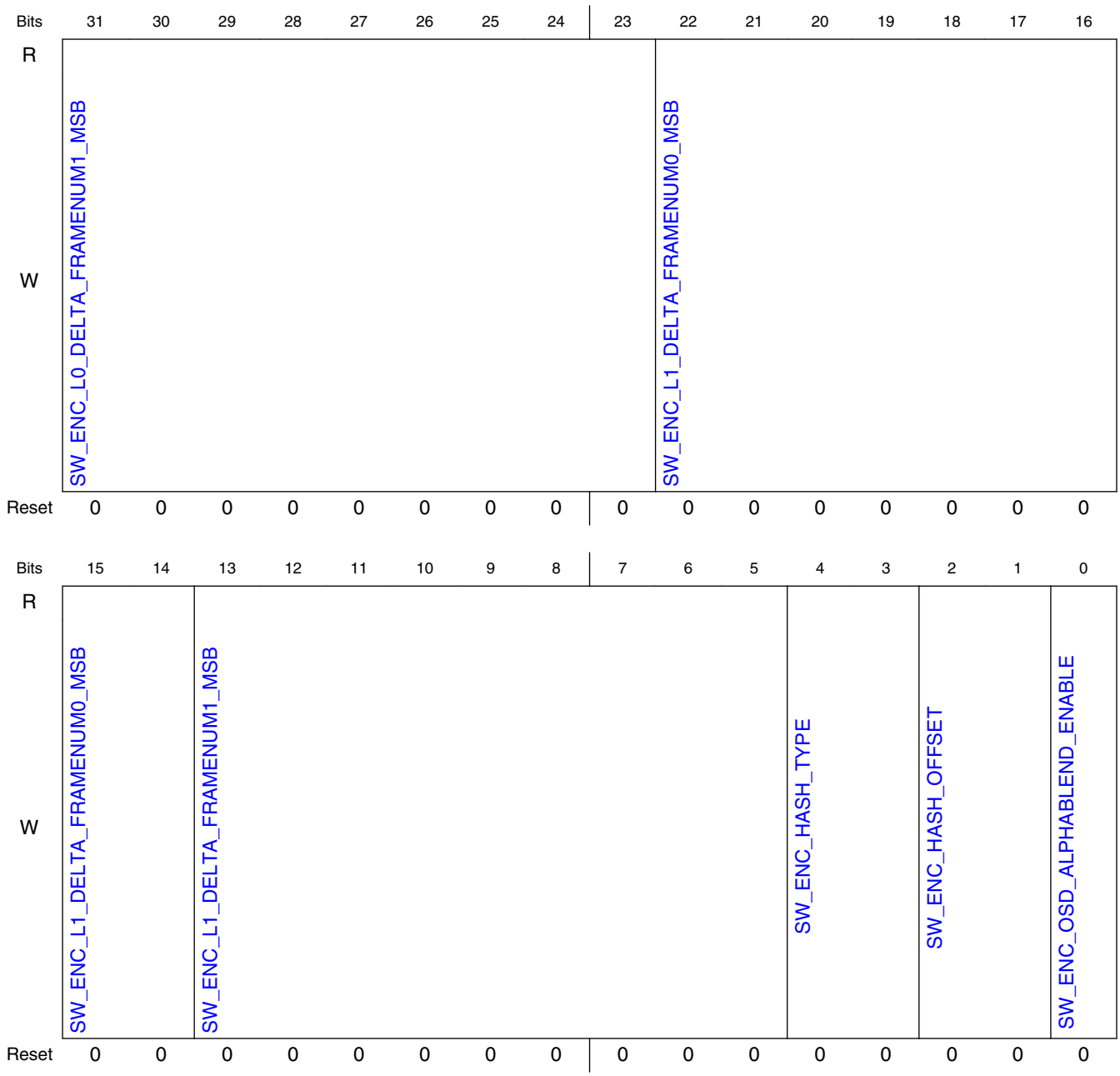
16.3.5.1.158.1 Offset

Register	Offset
swreg199	31Ch

16.3.5.1.158.2 Function

Hash Code Control

16.3.5.1.158.3 Diagram



16.3.5.1.158.4 Fields

Field	Function
31-23 SW_ENC_L0_DELTA_FRAMENUM1_MSB	delta frame_num of list0 pic1 (bit[19..11])

Table continues on the next page...

Field	Function
22-14 SW_ENC_L1_DELTA_FRAMENUM0_MSB	delta frame_num of list1 pic0 (bit[19..11])
13-5 SW_ENC_L1_DELTA_FRAMENUM1_MSB	delta frame_num of list1 pic1 (bit[19..11])
4-3 SW_ENC_HASH_TYPE	hash type of frame data. 00b - none. 01b - crc32. 10b - checksum32
2-1 SW_ENC_HASH_OFFSET	hash offset (byte offset of processed hashdata)
0 SW_ENC_OSD_ALPHABLEND_ENABLE	enable OSD Alpha Blending. 0b - disable. 1b - enable.

16.3.5.1.159 Hash Code Value (swreg200)

16.3.5.1.159.1 Offset

Register	Offset
swreg200	320h

16.3.5.1.159.2 Function

Hash Code Value

16.3.5.1.159.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_HASH_VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_HASH_VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.159.4 Fields

Field	Function
31-0 SW_ENC_HASH_VAL	hash value of frame data

16.3.5.1.160 Background SKIP Control 0 (swreg201)

16.3.5.1.160.1 Offset

Register	Offset
swreg201	324h

16.3.5.1.160.2 Function

Background SKIP Control 0

16.3.5.1.160.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MEAN_THR0								SW_ENC_MEAN_THR1							
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MEAN_THR2								SW_ENC_MEAN_THR3							
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

16.3.5.1.160.4 Fields

Field	Function
31-24 SW_ENC_MEAN_THR0	A mean value threshold of out circle used for foreground decision
23-16	A mean value threshold of second circle for foreground decision

Table continues on the next page...

Field	Function
SW_ENC_MEAN_THR1	
15-8 SW_ENC_MEAN_THR2	A mean value threshold of third circle
7-0 SW_ENC_MEAN_THR3	A mean value threshold of the smallest circle

16.3.5.1.161 Background SKIP Control 2 (swreg203)

16.3.5.1.161.1 Offset

Register	Offset
swreg203	32Ch

16.3.5.1.161.2 Function

Background SKIP Control 2

16.3.5.1.161.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_LUM_DC_SUM_THR								Reserved							
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_CB_DC_SUM_THR								SW_ENC_CR_DC_SUM_THR							
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

16.3.5.1.161.4 Fields

Field	Function
31-24 SW_ENC_LUM_DC_SUM_THR	A threshold value of DC for Y (for H.264)

Table continues on the next page...

Field	Function
23-16 —	Reserved
15-8 SW_ENC_CB_DC_SUM_THR	A threshold value of DC for U (for H.264)
7-0 SW_ENC_CR_DC_SUM_THR	A threshold value of DC for V (for H.264)

16.3.5.1.162 Background SKIP Control 7 (swreg208)

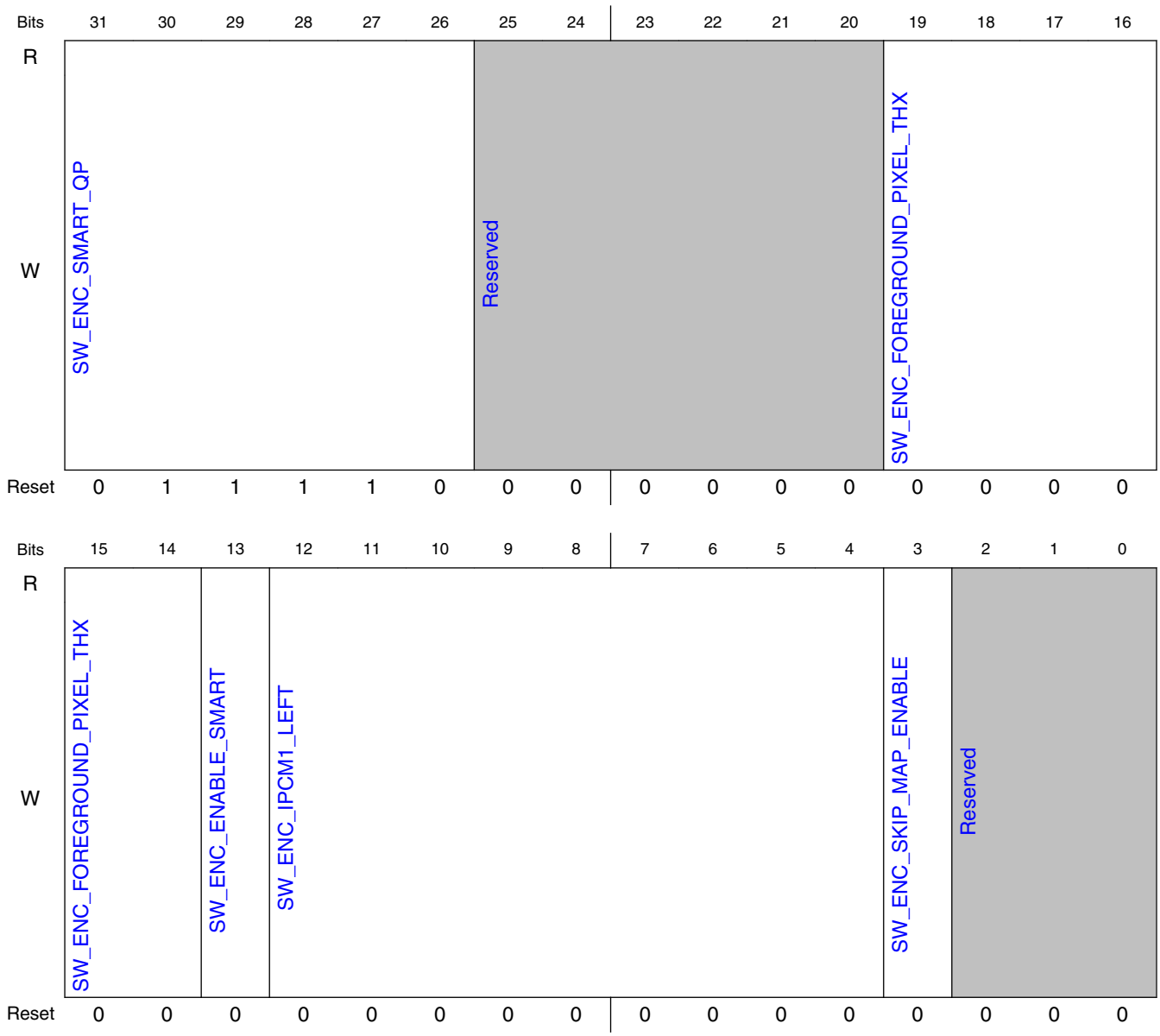
16.3.5.1.162.1 Offset

Register	Offset
swreg208	340h

16.3.5.1.162.2 Function

Background SKIP Control 7

16.3.5.1.162.3 Diagram



16.3.5.1.162.4 Fields

Field	Function
31-26 SW_ENC_SMART_QP	A luma QP value for skip decision (for H.264)
25-20 —	Reserved
19-14	Foreground threshold number of pixel (0 ~ 63) used for foreground decision

Table continues on the next page...

Field	Function
SW_ENC_FOREGROUND_PIXEL_THX	
13 SW_ENC_ENABLE_SMART	enable smart featrue
12-4 SW_ENC_IPCM1_LEFT	IPCM1 Area Left
3 SW_ENC_SKIP_MAP_ENABLE	enable skip map mode
2-0 —	Reserved

16.3.5.1.163 IPCM Control 0 (swreg209)

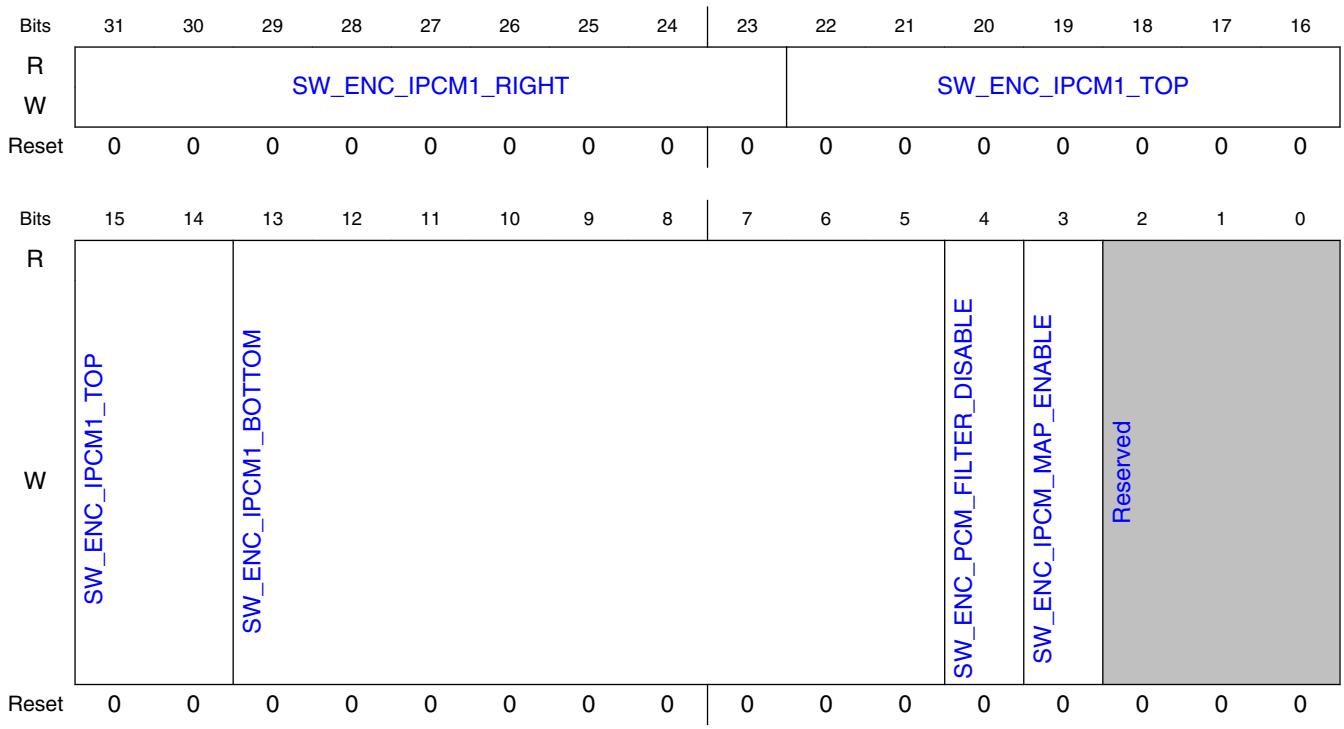
16.3.5.1.163.1 Offset

Register	Offset
swreg209	344h

16.3.5.1.163.2 Function

IPCM Control 0

16.3.5.1.163.3 Diagram



16.3.5.1.163.4 Fields

Field	Function
31-23 SW_ENC_IPCM1_RIGHT	IPCM1 Area Right
22-14 SW_ENC_IPCM1_TOP	IPCM1 Area Top
13-5 SW_ENC_IPCM1_BOTTOM	IPCM1 Area Bottom
4 SW_ENC_PCM_FILTER_DISABLE	disable deblock filter for IPCM
3 SW_ENC_IPCM_MAP_ENABLE	enable pcm map mode
2-0 —	Reserved

16.3.5.1.164 IPCM Control 1 (swreg210)

16.3.5.1.164.1 Offset

Register	Offset
swreg210	348h

16.3.5.1.164.2 Function

IPCM Control 1

16.3.5.1.164.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				SW_ENC_IPCM2_LEFT									Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.164.4 Fields

Field	Function
31-12 —	Reserved
11-3 SW_ENC_IPCM2_LEFT	IPCM2 Area Left
2-0 —	Reserved

16.3.5.1.165 IPCM Control 2 (swreg211)

16.3.5.1.165.1 Offset

Register	Offset
swreg211	34Ch

16.3.5.1.165.2 Function

IPCM Control 2

16.3.5.1.165.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				SW_ENC_IPCM2_RIGHT									Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.165.4 Fields

Field	Function
31-12 —	Reserved
11-3 SW_ENC_IPCM2_RIGHT	IPCM2 Area Right
2-0 —	Reserved

16.3.5.1.166 IPCM Control 3 (swreg212)

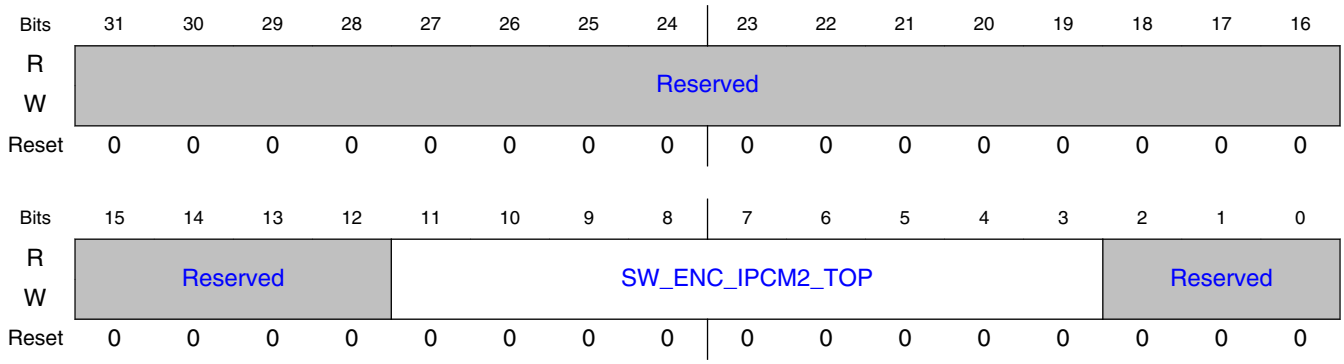
16.3.5.1.166.1 Offset

Register	Offset
swreg212	350h

16.3.5.1.166.2 Function

IPCM Control 3

16.3.5.1.166.3 Diagram



16.3.5.1.166.4 Fields

Field	Function
31-12 —	Reserved
11-3 SW_ENC_IPCM2_TOP	IPCM2 Area Top
2-0 —	Reserved

16.3.5.1.167 IPCM Control 4 (swreg213)

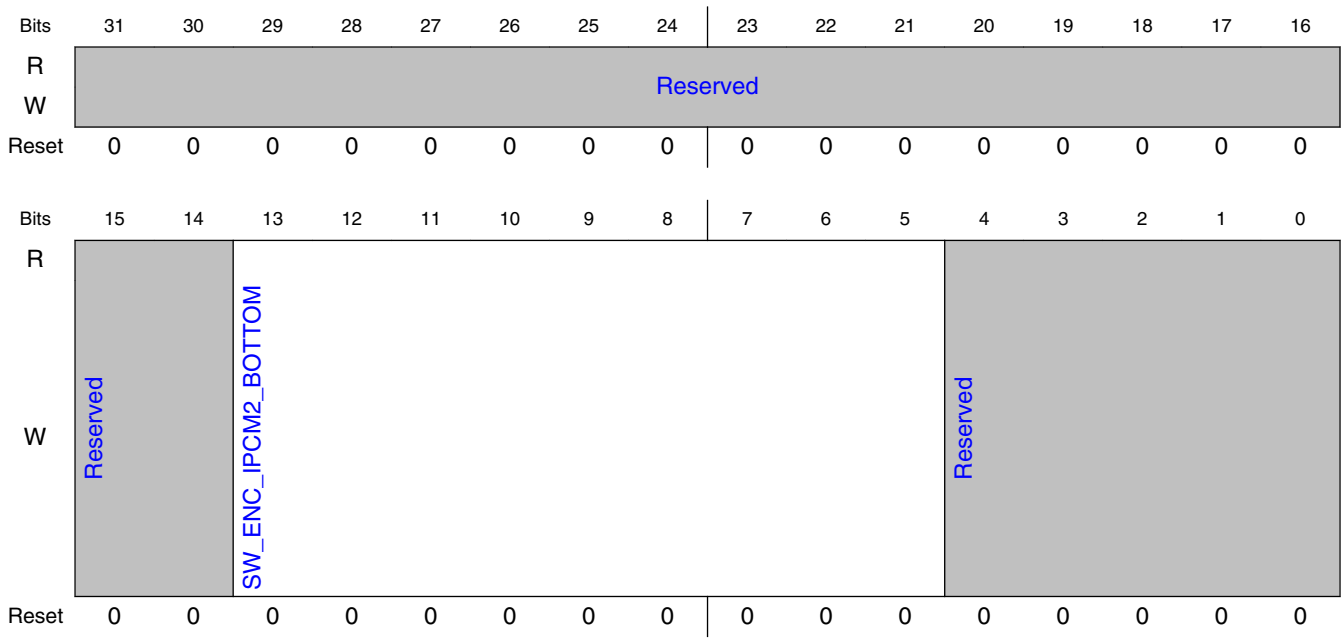
16.3.5.1.167.1 Offset

Register	Offset
swreg213	354h

16.3.5.1.167.2 Function

IPCM Control 4

16.3.5.1.167.3 Diagram



16.3.5.1.167.4 Fields

Field	Function
31-14 —	Reserved
13-5 SW_ENC_IPCM2_BOTTOM	IPCM2 Area Bottom
4-0 —	Reserved

16.3.5.1.168 HW synthesis config register 2, read-only (swreg214)

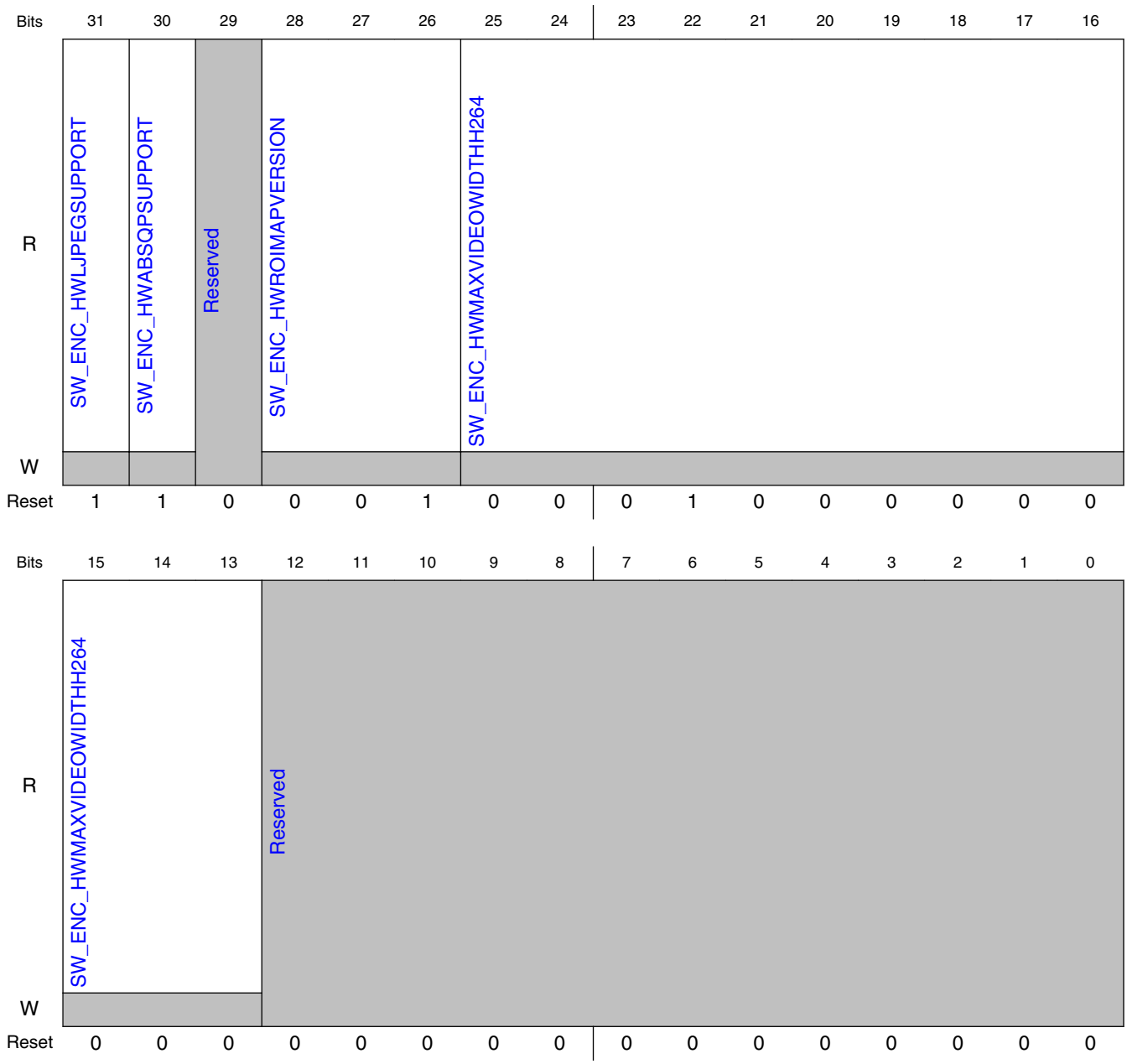
16.3.5.1.168.1 Offset

Register	Offset
swreg214	358h

16.3.5.1.168.2 Function

HW synthesis config register 2, read-only

16.3.5.1.168.3 Diagram



16.3.5.1.168.4 Fields

Field	Function
31 SW_ENC_HWL JPEGSUPPORT	Lossless JPEG supported by HW. 0b - not supported. 1b - supported
30	Absolute QP in ROI/ROI map supported by HW.

Table continues on the next page...

Field	Function
SW_ENC_HWABSQPSUPPORT	0b - not supported. 1b - supported
29 —	Reserved
28-26 SW_ENC_HWROIMAPVERSION	ROI map buffer format version. 000b - 4 bit per pixel. 001b - 8 bit per pixel
25-13 SW_ENC_HWMAXVIDEOWIDTHH264	Maximum video width supported by HW for H264 encoding (unit 8 pixels)
12-0 —	Reserved

16.3.5.1.169 AXI Information 0 (swreg215)

16.3.5.1.169.1 Offset

Register	Offset
swreg215	35Ch

16.3.5.1.169.2 Function

AXI Information 0

16.3.5.1.169.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALARLEN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALARLEN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.169.4 Fields

Field	Function
31-0 SW_ENC_TOTALARLEN	[AXI] Accumulated ARLEN+1

16.3.5.1.170 AXI Information 1 (swreg216)

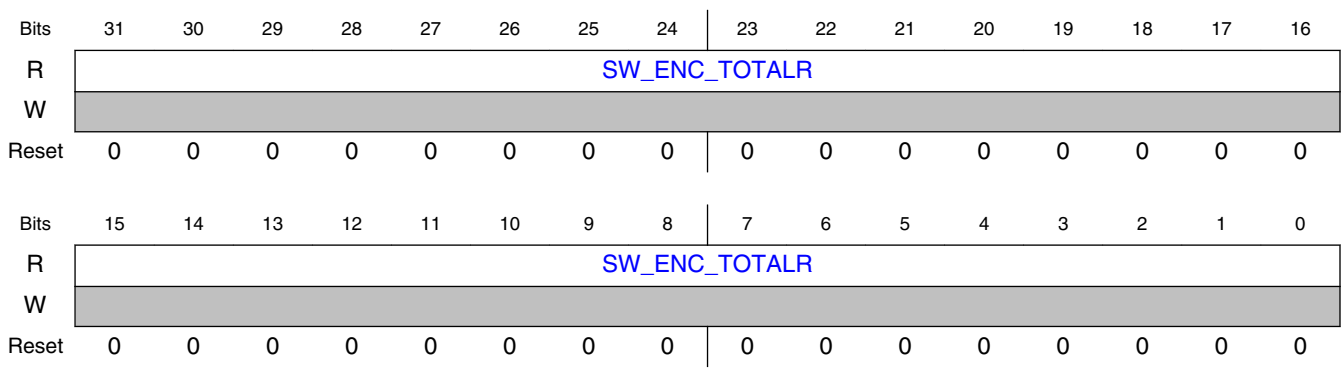
16.3.5.1.170.1 Offset

Register	Offset
swreg216	360h

16.3.5.1.170.2 Function

AXI Information 1

16.3.5.1.170.3 Diagram



16.3.5.1.170.4 Fields

Field	Function
31-0 SW_ENC_TOTALR	[AXI] RVALID & RREADY

16.3.5.1.171 AXI Information 2 (swreg217)

16.3.5.1.171.1 Offset

Register	Offset
swreg217	364h

16.3.5.1.171.2 Function

AXI Information 2

16.3.5.1.171.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALAR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALAR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.171.4 Fields

Field	Function
31-0 SW_ENC_TOTALAR	[AXI] ARVALID & ARREADY

16.3.5.1.172 AXI Information 3 (swreg218)

16.3.5.1.172.1 Offset

Register	Offset
swreg218	368h

16.3.5.1.172.2 Function

AXI Information 3

16.3.5.1.172.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALRLAST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALRLAST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.172.4 Fields

Field	Function
31-0 SW_ENC_TOTALRLAST	[AXI] RVALID & RREADY & RLAST

16.3.5.1.173 AXI Information 4 (swreg219)

16.3.5.1.173.1 Offset

Register	Offset
swreg219	36Ch

16.3.5.1.173.2 Function

AXI Information 4

16.3.5.1.173.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALAWLEN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALAWLEN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.173.4 Fields

Field	Function
31-0 SW_ENC_TOTALAWLEN	[AXI] Accumulated AWLEN+1

16.3.5.1.174 AXI Information 5 (swreg220)

16.3.5.1.174.1 Offset

Register	Offset
swreg220	370h

16.3.5.1.174.2 Function

AXI Information 5

16.3.5.1.174.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.174.4 Fields

Field	Function
31-0 SW_ENC_TOTALW	[AXI] WVALID & WREADY

16.3.5.1.175 AXI Information 6 (swreg221)

16.3.5.1.175.1 Offset

Register	Offset
swreg221	374h

16.3.5.1.175.2 Function

AXI Information 6

16.3.5.1.175.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALAW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALAW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.175.4 Fields

Field	Function
31-0 SW_ENC_TOTALAW	[AXI] AWVALID & AWREADY

16.3.5.1.176 AXI Information 7 (swreg222)

16.3.5.1.176.1 Offset

Register	Offset
swreg222	378h

16.3.5.1.176.2 Function

AXI Information 7

16.3.5.1.176.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALWLAST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALWLAST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.176.4 Fields

Field	Function
31-0 SW_ENC_TOTALWLAST	[AXI] WVALID & WREADY & WLAST

16.3.5.1.177 AXI Information 8 (swreg223)

16.3.5.1.177.1 Offset

Register	Offset
swreg223	37Ch

16.3.5.1.177.2 Function

AXI Information 8

16.3.5.1.177.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.177.4 Fields

Field	Function
31-0 SW_ENC_TOTALB	[AXI] BVALID & BREADY

16.3.5.1.178 control register 4 (swreg224)

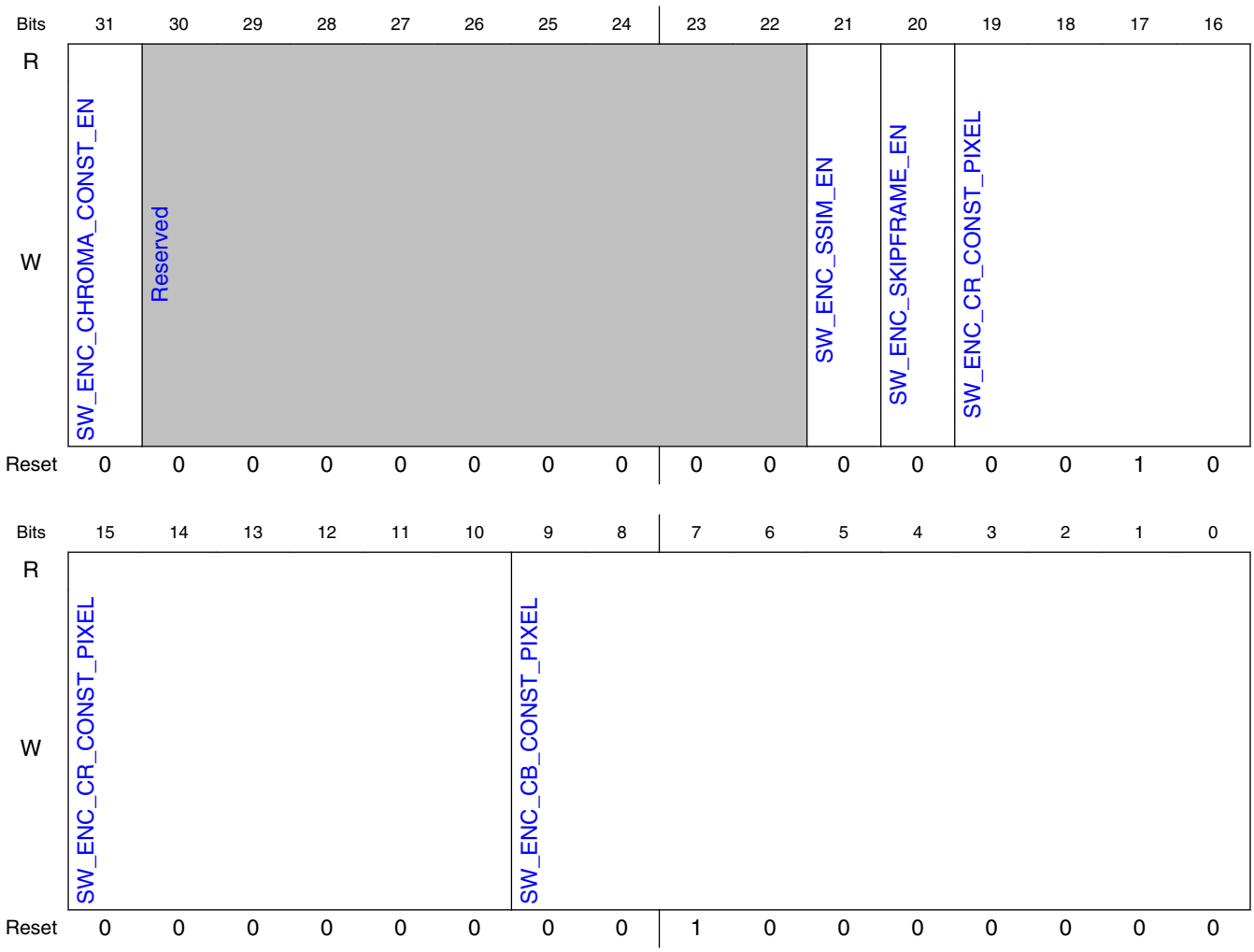
16.3.5.1.178.1 Offset

Register	Offset
swreg224	380h

16.3.5.1.178.2 Function

control register 4

16.3.5.1.178.3 Diagram



16.3.5.1.178.4 Fields

Field	Function
31 SW_ENC_CHROMA_CONST_EN	Force Chroma to be a constant pixel or not. 0b - no. 1b - yes.
30-22 —	Reserved
21 SW_ENC_SSIM_EN	Enable SSIM calculation. 0b - Disable. 1b - Enable
20	Force frame encode as SKIPFRAME. 0b - no.

Table continues on the next page...

Field	Function
SW_ENC_SKIP FRAME_EN	1b - yes
19-10 SW_ENC_CR_ CONST_PIXEL	The constant pixel of CR if sw_enc_chroma_const_en=1
9-0 SW_ENC_CB_ CONST_PIXEL	The constant pixel of CB if sw_enc_chroma_const_en=1

16.3.5.1.179 Tile Control (swreg225)

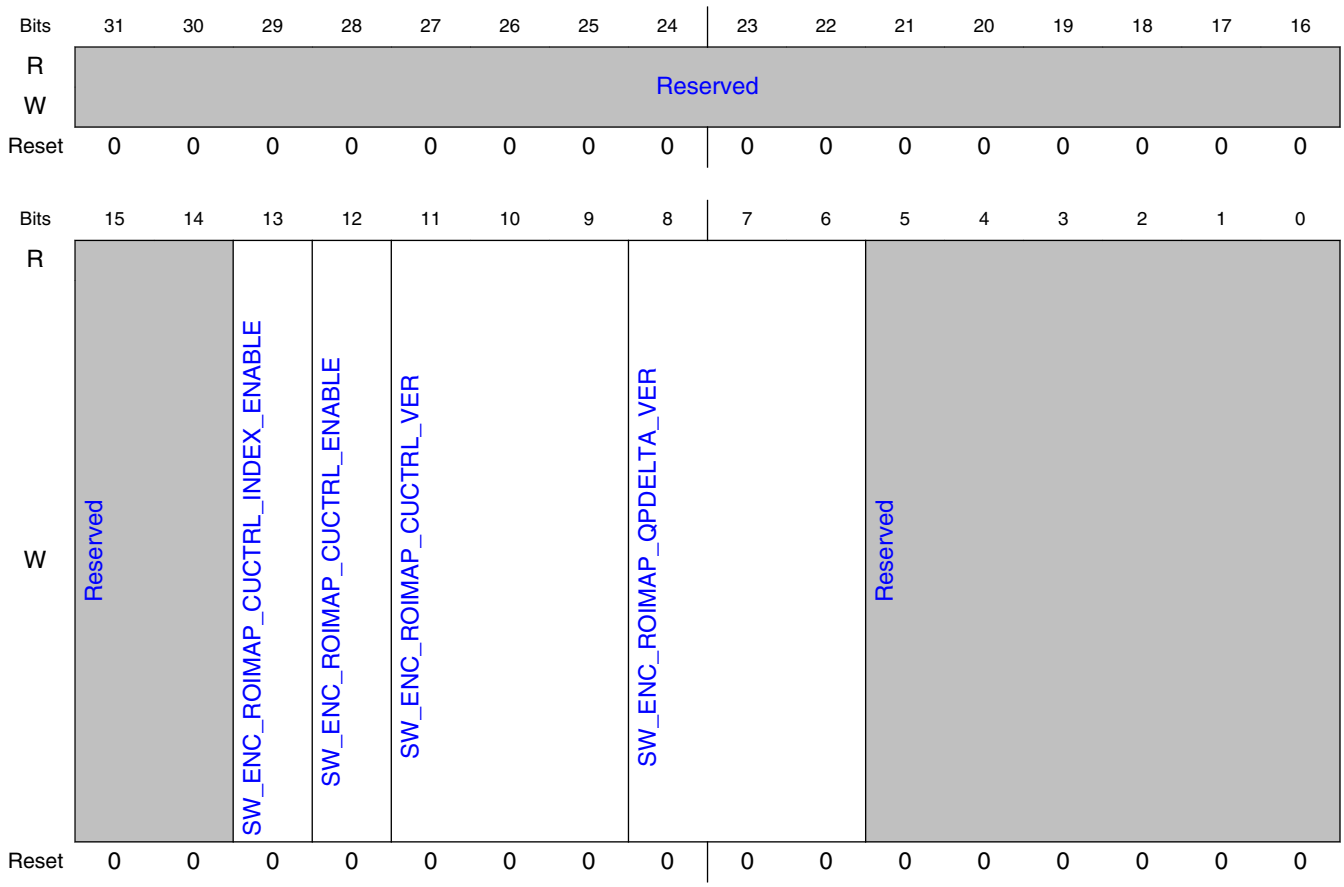
16.3.5.1.179.1 Offset

Register	Offset
swreg225	384h

16.3.5.1.179.2 Function

Tile Control

16.3.5.1.179.3 Diagram



16.3.5.1.179.4 Fields

Field	Function
31-14 —	Reserved
13 SW_ENC_ROIMAP_CUCTRL_INDEX_ENABLE	roi map cu ctrl index enable. 0b - Disable. 1b - Enable
12 SW_ENC_ROIMAP_CUCTRL_ENABLE	roi map cu ctrl enable. 0b - Disable. 1b - Enable
11-9 SW_ENC_ROIMAP_CUCTRL_VER	roi map cu ctrl info version. 3~7
8-6	roi map qp delta info version. 1~3

Table continues on the next page...

Field	Function
SW_ENC_ROIMAP_QPDELTA_VER	
5-0 —	Reserved

16.3.5.1.180 HW synthesis config register 3, read-only (swreg226)

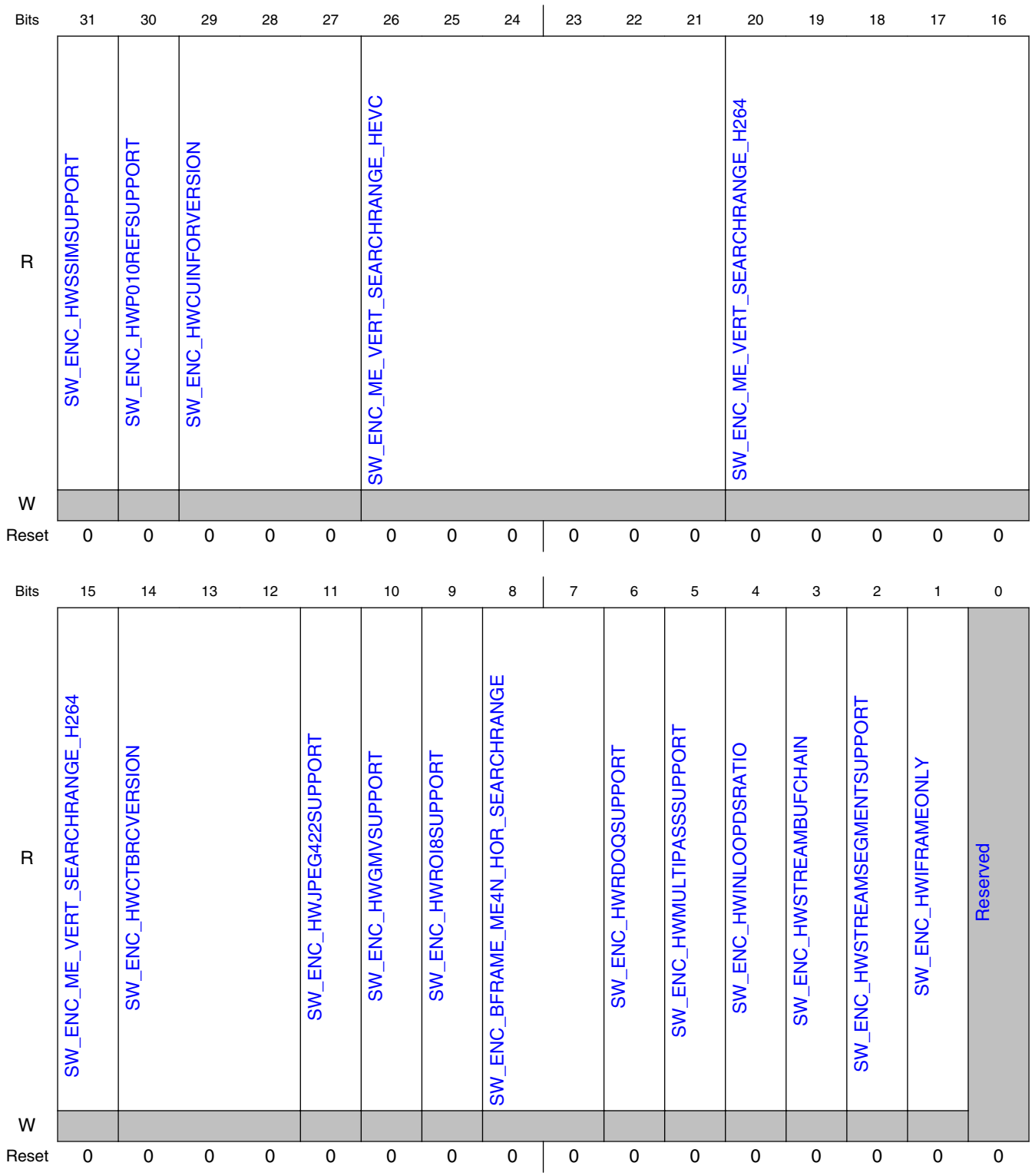
16.3.5.1.180.1 Offset

Register	Offset
swreg226	388h

16.3.5.1.180.2 Function

HW synthesis config register 3, read-only

16.3.5.1.180.3 Diagram



16.3.5.1.180.4 Fields

Field	Function
31 SW_ENC_HWSSIMSUPPORT	SSIM calculation supported by HW. 0b - not supported. 1b - supported
30 SW_ENC_HWP010REFSUPPORT	Reference buffer format for 10-bit encoding. 0b - normal format 1b - Reserved
29-27 SW_ENC_HWCUINFORVERSION	Version of the output CU information format.
26-21 SW_ENC_ME_VERT_SEARCHRANGE_HEVC	ME vertical search range in 8 pixel unit for HEVC
20-15 SW_ENC_ME_VERT_SEARCHRANGE_H264	ME vertical search range in 8 pixel unit for H264
14-12 SW_ENC_HWCTBRCVERSION	CTB Rate Control Version.
11 SW_ENC_HWJPEG422SUPPORT	Jpeg422 supported by HW. 0b - not supported. 1b - supported
10 SW_ENC_HWGMVSUPPORT	Global MV supported by HW. 0b - not supported. 1b - supported
9 SW_ENC_HWROI8SUPPORT	support 8 ROIs
8-7 SW_ENC_BFRAME_ME4N_HOR_SEARCHRANGE	ME4N horizontal search range in 64 pixel unit 00b - 64. 01b - 128. 10b - 192. 11b - 256
6 SW_ENC_HWRDOQSUPPORT	RDOQ supported by HW. 0b - not supported. 1b - supported
5 SW_ENC_HWMULTIPASSSUPPORT	Multipass Encoding supported by HW. 0b - not supported. 1b - supported

Table continues on the next page...

Field	Function
4 SW_ENC_HWINLOOPDSRATIO	in-loop ds ratio supported by HW 0b - 1:1 1b - 1:2
3 SW_ENC_HWSTREAMBUFCHAIN	Stream Buffer Chain supported by HW. 0b - not supported. 1b - supported
2 SW_ENC_HWSTREAMSEGMENTSUPPORT	Stream segment supported by HW. 0b - not supported. 1b - supported
1 SW_ENC_HWIFRAMEONLY	Only support I frame. 0b - support I/P/B frame. 1b - only support I frame
0 —	Reserved

16.3.5.1.181 RPS encoding control 0 (swreg235)

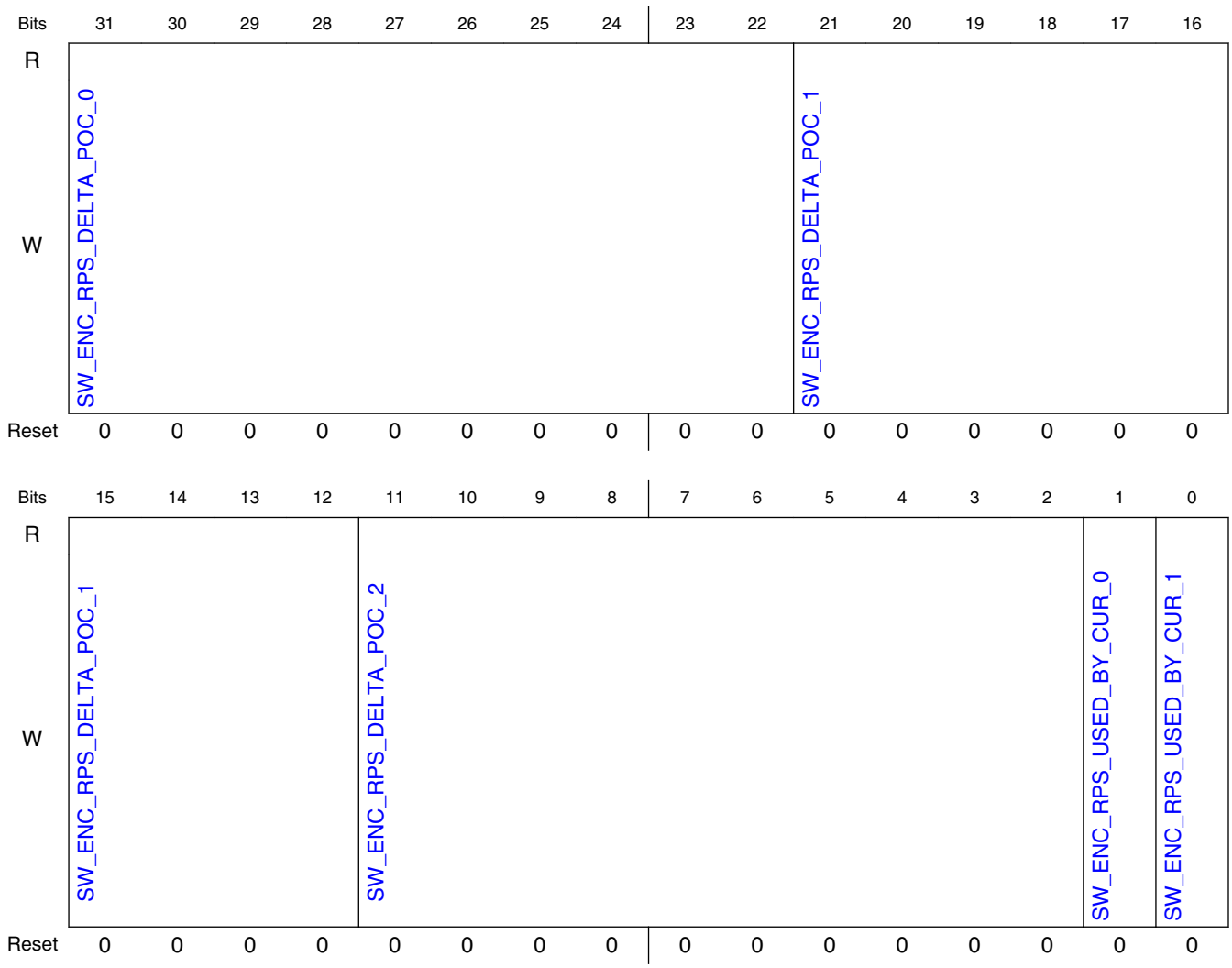
16.3.5.1.181.1 Offset

Register	Offset
swreg235	3ACh

16.3.5.1.181.2 Function

RPS encoding control 0

16.3.5.1.181.3 Diagram



16.3.5.1.181.4 Fields

Field	Function
31-22 SW_ENC_RPS_DELTA_POC_0	delta poc of pic0 in current rps
21-12 SW_ENC_RPS_DELTA_POC_1	delta poc of pic1 in current rps
11-2 SW_ENC_RPS_DELTA_POC_2	delta poc of pic2 in current rps
1	pic0 in current rps is used for reference by current slice

Table continues on the next page...

Field	Function
SW_ENC_RPS_USED_BY_CUR_0	
0 SW_ENC_RPS_USED_BY_CUR_1	pic1 in current rps is used for reference by current slice

16.3.5.1.182 RPS encoding control 1 (swreg236)

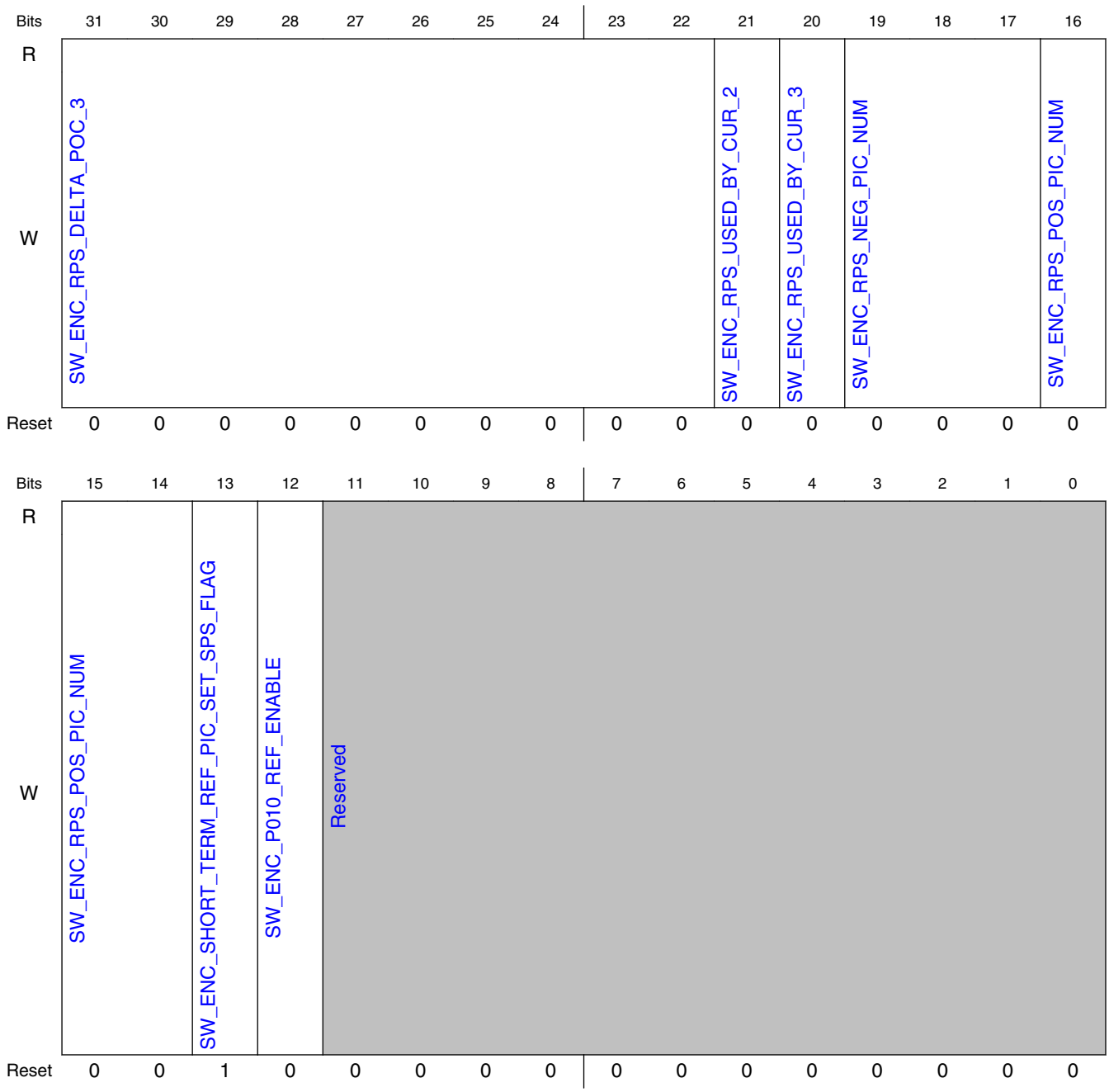
16.3.5.1.182.1 Offset

Register	Offset
swreg236	3B0h

16.3.5.1.182.2 Function

RPS encoding control 1

16.3.5.1.182.3 Diagram



16.3.5.1.182.4 Fields

Field	Function
31-22 SW_ENC_RPS_DELTA_POC_3	delta poc of pic3 in current rps

Table continues on the next page...

Field	Function
21 SW_ENC_RPS_USED_BY_CUR_2	pic2 in current rps is used for reference by current slice
20 SW_ENC_RPS_USED_BY_CUR_3	pic3 in current rps is used for reference by current slice
19-17 SW_ENC_RPS_NEG_PIC_NUM	number of negative pictures in current rps
16-14 SW_ENC_RPS_POS_PIC_NUM	number of positive pictures in current rps
13 SW_ENC_SHORT_TERM_REF_PIC_SET_SPS_FLAG	short term rps for current slice is encoded in sps
12 SW_ENC_P010_REF_ENABLE	Enable/Disable P010 Reference format. P010RefEnable[0]. 0b - not supported. 1b - supported.
11-0 —	Reserved

16.3.5.1.183 Stride Control (swreg237)

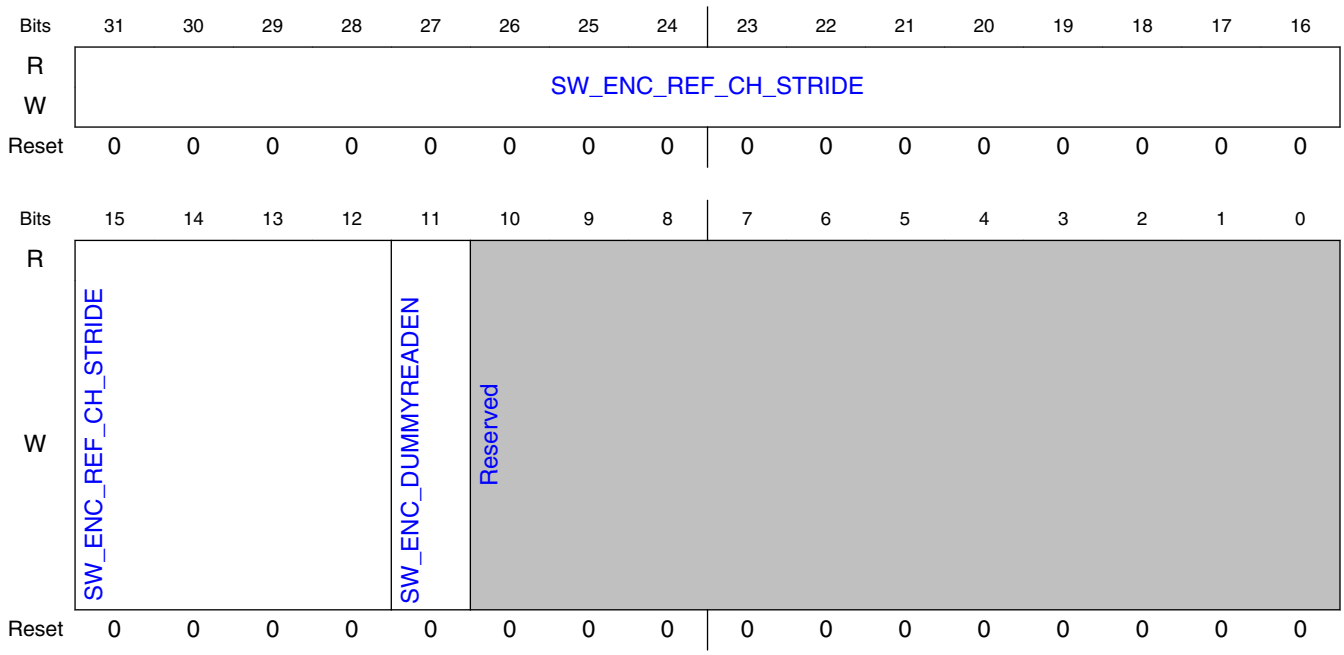
16.3.5.1.183.1 Offset

Register	Offset
swreg237	3B4h

16.3.5.1.183.2 Function

Stride Control

16.3.5.1.183.3 Diagram



16.3.5.1.183.4 Fields

Field	Function
31-12 SW_ENC_REF_CH_STRIDE	Chroma stride of reference frame
11 SW_ENC_DUMMYREADEN	enable dummy read when frame height not align to 64 or other possible condition
10-0 —	Reserved

16.3.5.1.184 Dummy Read (swreg238)

16.3.5.1.184.1 Offset

Register	Offset
swreg238	3B8h

16.3.5.1.184.2 Function

Dummy Read

16.3.5.1.184.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_DUMMYREADADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_DUMMYREADADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.184.4 Fields

Field	Function
31-0 SW_ENC_DUMMYREADADDR	Dummy read address

16.3.5.1.185 Base Address LSB of CTB MADs of current frame. (swreg239)

16.3.5.1.185.1 Offset

Register	Offset
swreg239	3BCh

16.3.5.1.185.2 Function

Base Address LSB of CTB MADs of current frame.

16.3.5.1.185.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_CURRENT_CTB_MAD_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_CURRENT_CTB_MAD_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.185.4 Fields

Field	Function
31-0 SW_ENC_CURRENT_CTB_MAD_BASE	Base Address LSB of CTB MADs of current frame used for CTB RC.

16.3.5.1.186 Base Address LSB of CTB MADs of previous frame. (swreg241)

16.3.5.1.186.1 Offset

Register	Offset
swreg241	3C4h

16.3.5.1.186.2 Function

Base Address LSB of CTB MADs of previous frame.

16.3.5.1.186.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_PREVIOUS_CTB_MAD_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_PREVIOUS_CTB_MAD_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.186.4 Fields

Field	Function
31-0 SW_ENC_PREVIOUS_CTB_MAD_BASE	Base Address LSB of CTB MADs of previous frame used for CTB RC.

16.3.5.1.187 CTB RC Control 0 (swreg243)

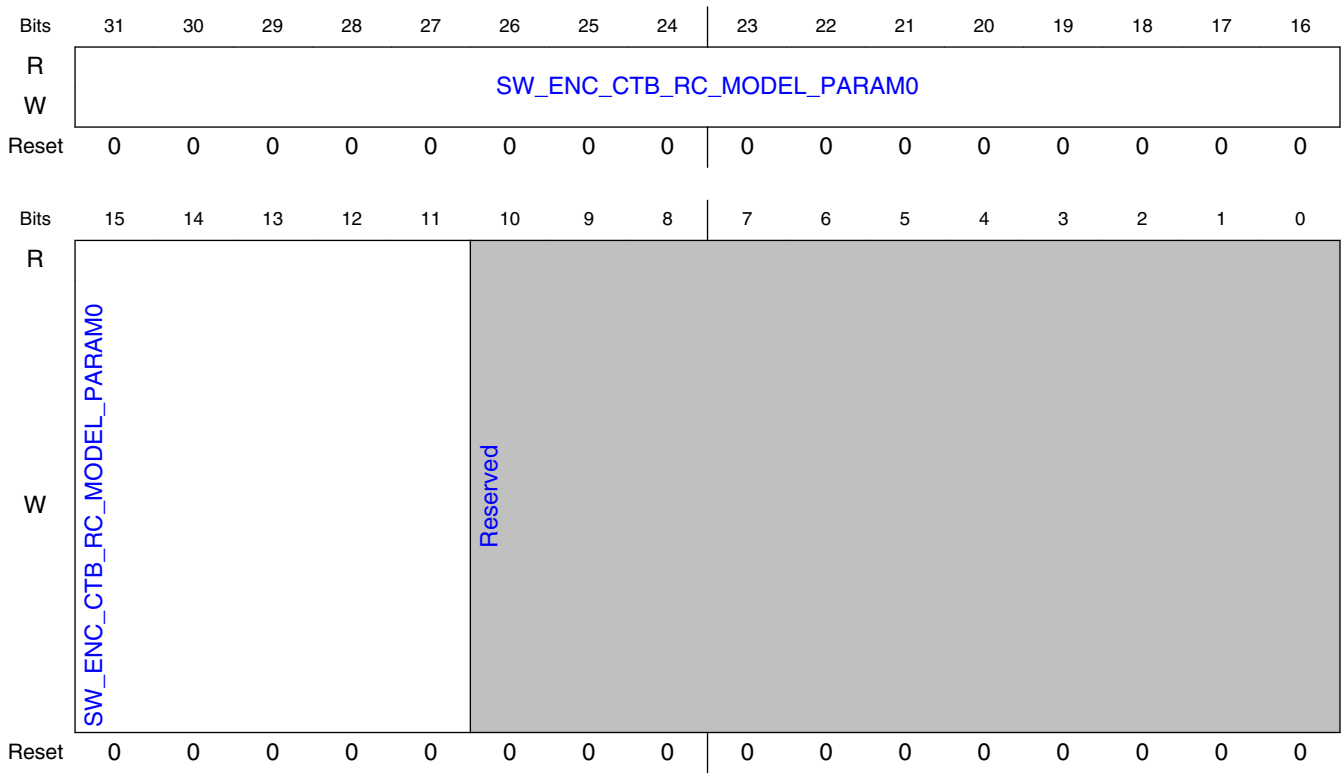
16.3.5.1.187.1 Offset

Register	Offset
swreg243	3CCh

16.3.5.1.187.2 Function

CTB RC Control 0

16.3.5.1.187.3 Diagram



16.3.5.1.187.4 Fields

Field	Function
31-11 SW_ENC_CTB_RC_MODEL_PARAM0	Parameter0 of CTB RC model
10-0 —	Reserved

16.3.5.1.188 CTB RC Control 1 (swreg244)

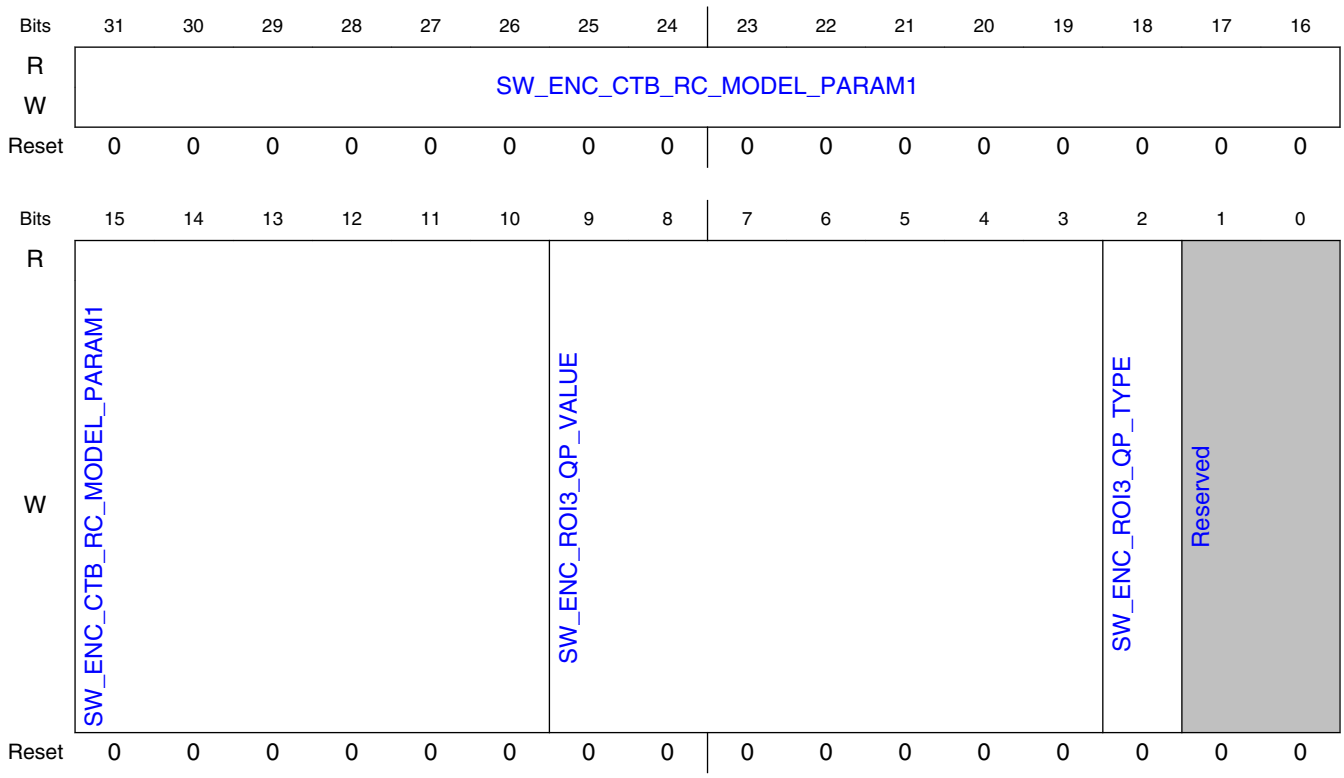
16.3.5.1.188.1 Offset

Register	Offset
swreg244	3D0h

16.3.5.1.188.2 Function

CTB RC Control 1

16.3.5.1.188.3 Diagram



16.3.5.1.188.4 Fields

Field	Function
31-10 SW_ENC_CTB_RC_MODEL_PARAM1	Parameter1 of CTB RC model
9-3 SW_ENC_ROI3_QP_VALUE	ROI3 qp value. delta or absolute value
2 SW_ENC_ROI3_QP_TYPE	ROI3 qp type. 0b - delta 1b - Absolute value
1-0 —	Reserved

16.3.5.1.189 CTB RC Control 2 (swreg245)

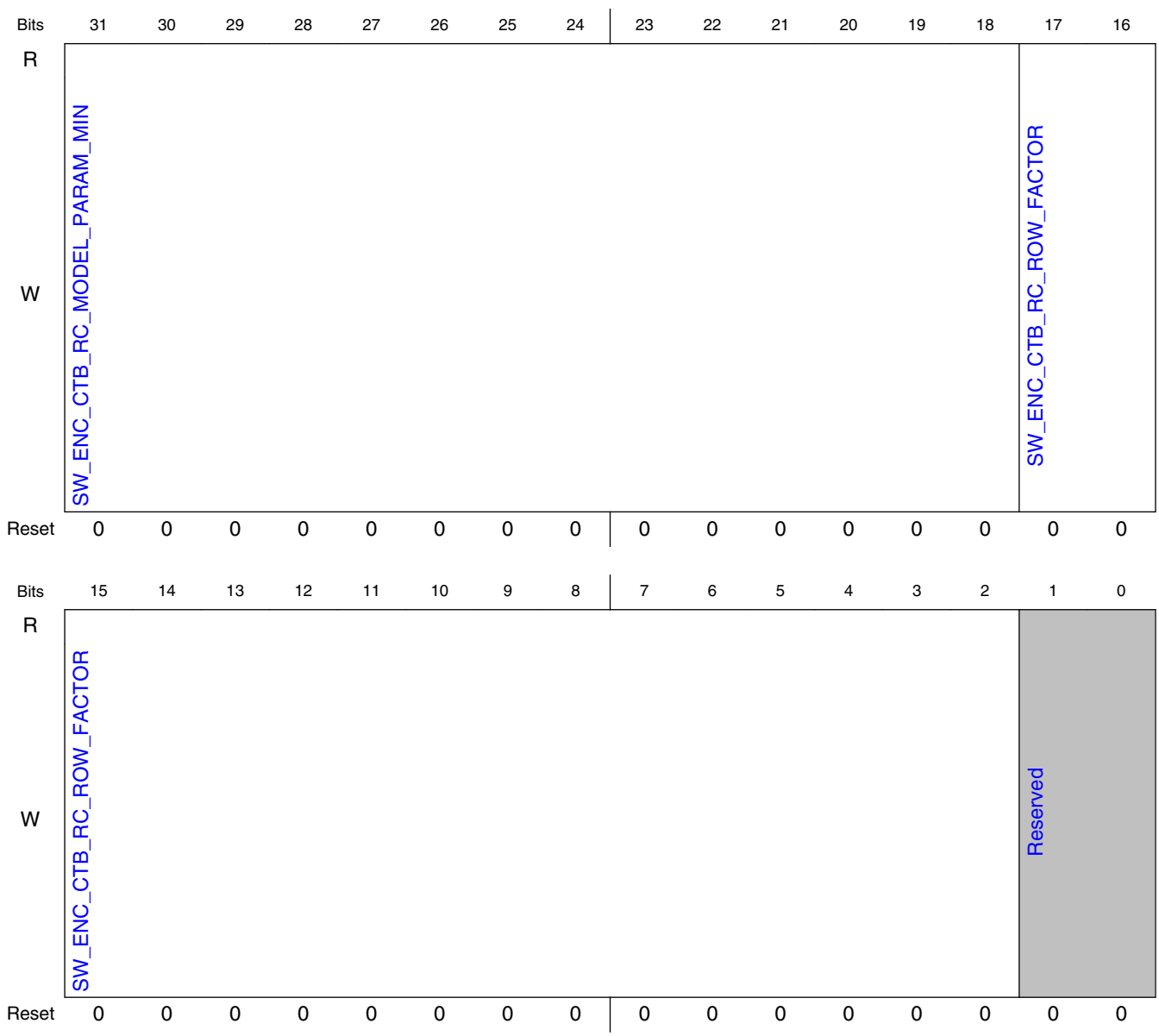
16.3.5.1.189.1 Offset

Register	Offset
swreg245	3D4h

16.3.5.1.189.2 Function

CTB RC Control 2

16.3.5.1.189.3 Diagram



16.3.5.1.189.4 Fields

Field	Function
31-18 SW_ENC_CTB_RC_MODEL_PARAM_MIN	Minimum parameter value of CTB RC model
17-2 SW_ENC_CTB_RC_ROW_FACTOR	Reciprocal of ctb per row. Q16
1-0 —	Reserved

16.3.5.1.190 CTB RC Control 3 (swreg246)

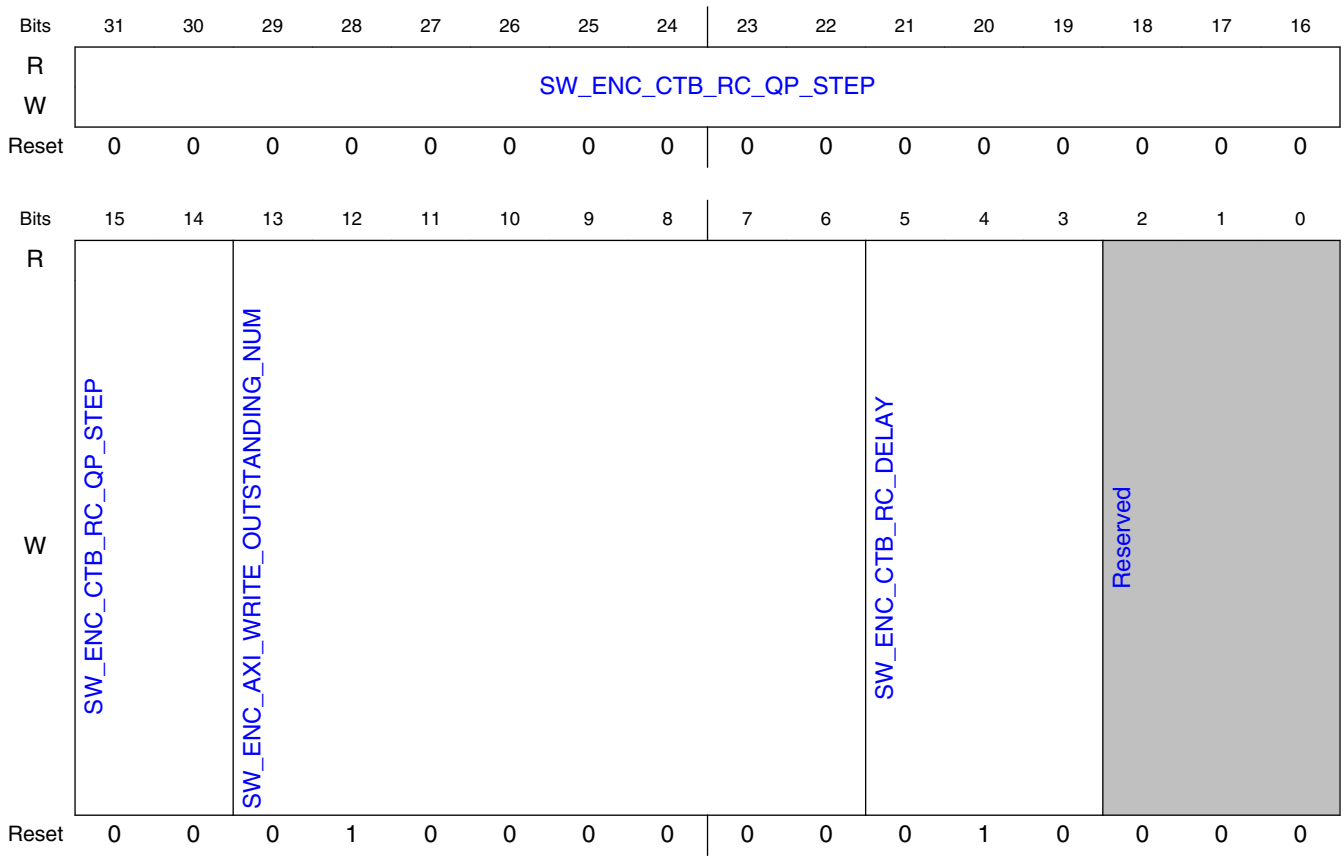
16.3.5.1.190.1 Offset

Register	Offset
swreg246	3D8h

16.3.5.1.190.2 Function

CTB RC Control 3

16.3.5.1.190.3 Diagram



16.3.5.1.190.4 Fields

Field	Function
31-14 SW_ENC_CTB_RC_QP_STEP	Maximum QP step per CTB for purpose of rate control. Q16.
13-6 SW_ENC_AXI_WRITE_OUTSTANDING_NUM	AXI outstanding number of write operation
5-3 SW_ENC_CTB_RC_DELAY	Feedback delay of information for CTB RC
2-0 —	Reserved

16.3.5.1.191 CTB RC Control 4 (swreg247)

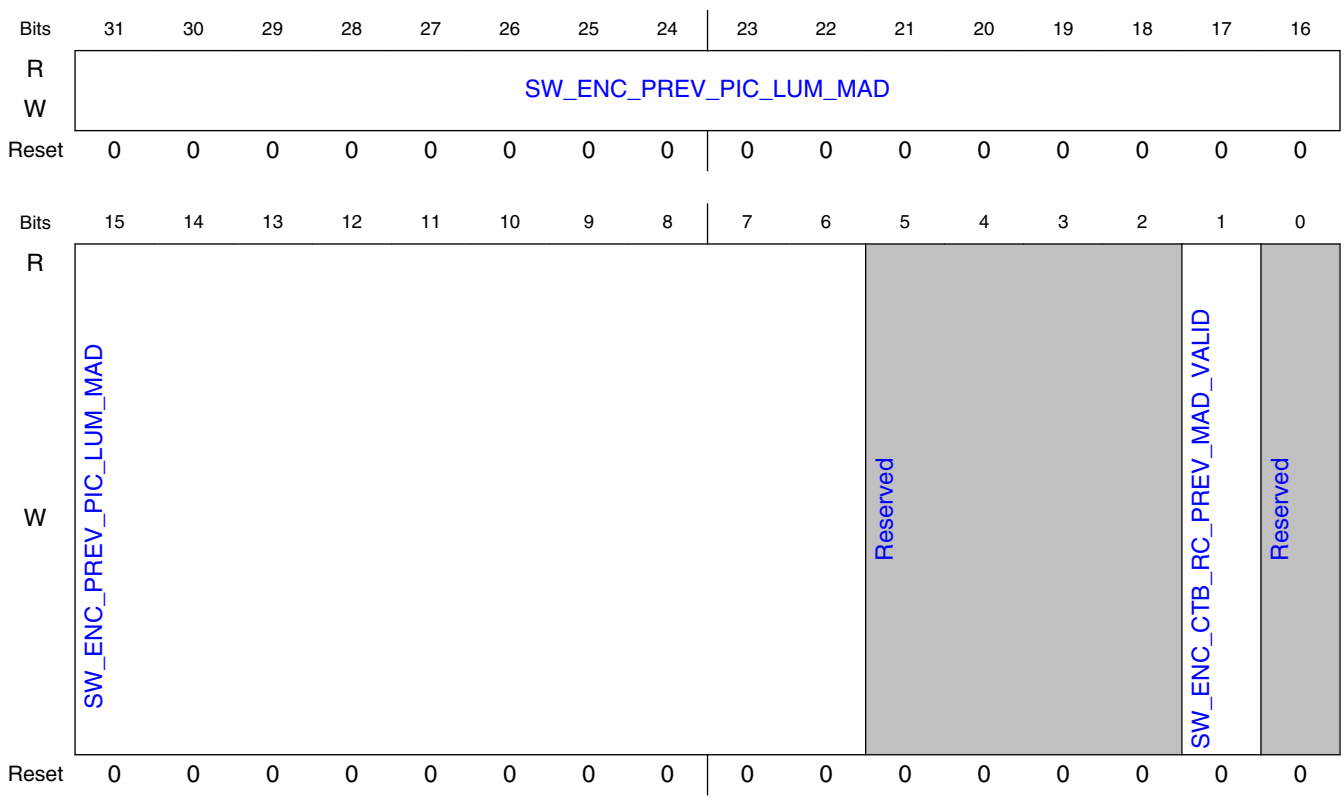
16.3.5.1.191.1 Offset

Register	Offset
swreg247	3DCh

16.3.5.1.191.2 Function

CTB RC Control 4

16.3.5.1.191.3 Diagram



16.3.5.1.191.4 Fields

Field	Function
31-6 SW_ENC_PREV_PIC_LUM_MAD	Luma MAD of the predict picture. Input for previous pic. Output for current pic.

Table continues on the next page...

Field	Function
5-2 —	Reserved
1 SW_ENC_CTB_RC_PREV_MAD_VALID	Prev luma MADs are valid or not for ctb rc. 0b - no 1b - yes.
0 —	Reserved

16.3.5.1.192 CTB RC Control 5 (swreg248)

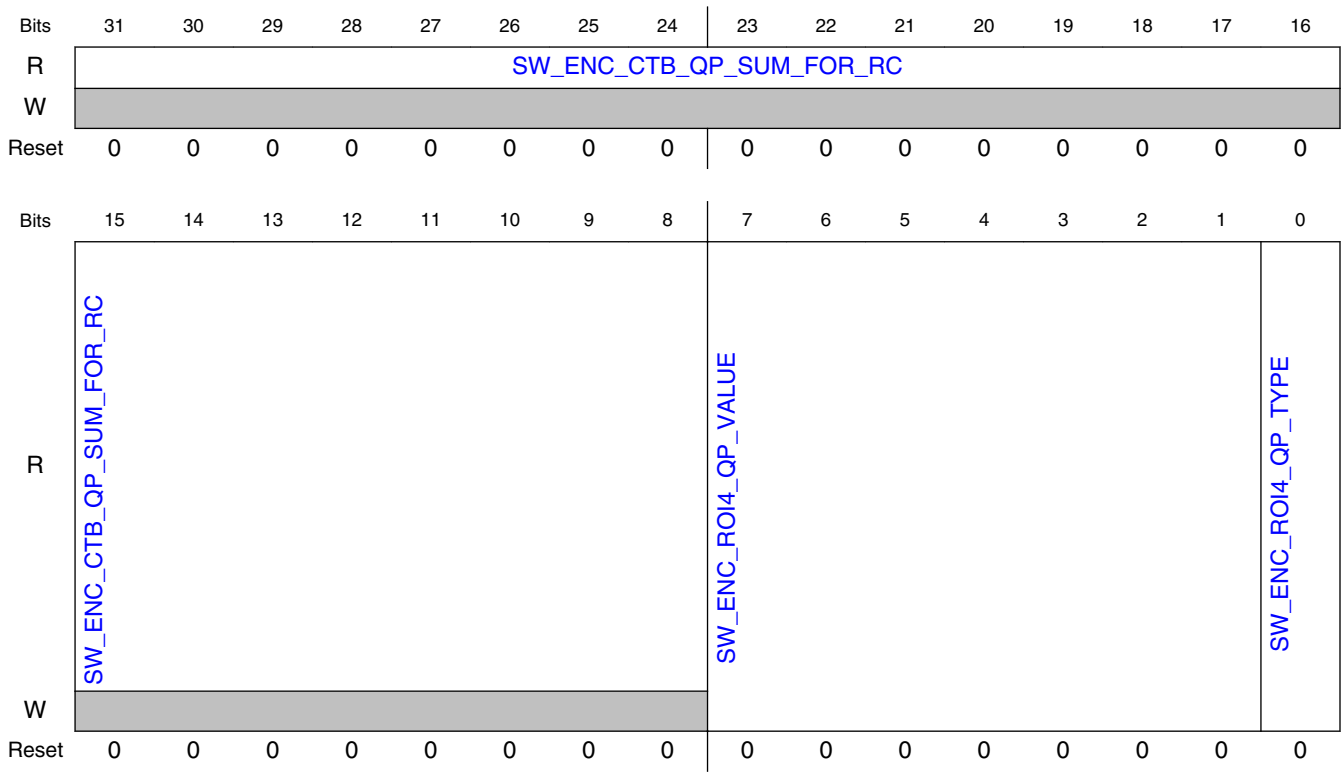
16.3.5.1.192.1 Offset

Register	Offset
swreg248	3E0h

16.3.5.1.192.2 Function

CTB RC Control 5

16.3.5.1.192.3 Diagram



16.3.5.1.192.4 Fields

Field	Function
31-8 SW_ENC_CTB_QP_SUM_FOR_RC	Sum of CTB QP used for rate control. not include QP delta for quality adjust or ROI.
7-1 SW_ENC_ROI4_QP_VALUE	ROI4 qp value. delta or absolute value
0 SW_ENC_ROI4_QP_TYPE	ROI4 qp type. 0b - delta 1b - Absolute value

16.3.5.1.193 register extension for 8K width (swreg249)

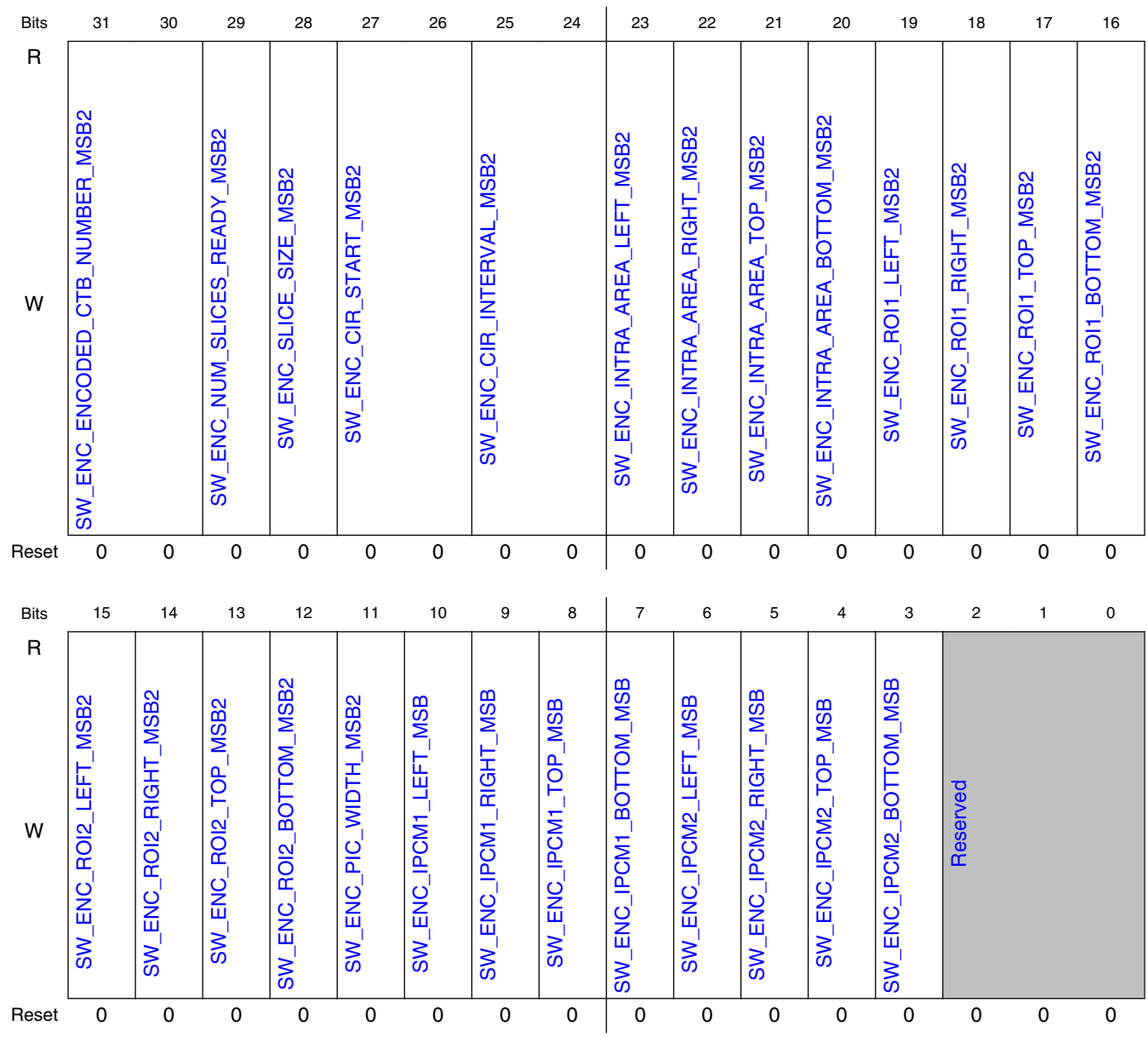
16.3.5.1.193.1 Offset

Register	Offset
swreg249	3E4h

16.3.5.1.193.2 Function

register extension for 8K width

16.3.5.1.193.3 Diagram



16.3.5.1.193.4 Fields

Field	Function
31-30 SW_ENC_ENCODED_CTB_NUMBER_MSB2	MB count output. Max 512*256. bit[17:18]
29 SW_ENC_NUM_SLICES_READY_MSB2	HEVC amount of completed slices. bit[10]
28 SW_ENC_SLICE_SIZE_MSB2	slice size in ctu row. bit[9]
27-26 SW_ENC_CIR_START_MSB2	cir start. bit[18:19]
25-24 SW_ENC_CIR_INTERVAL_MSB2	Cyclic Intra Refresh. bit[18:19]
23 SW_ENC_INTRA_AREA_LEFT_MSB2	intra Area Left. bit[9]
22 SW_ENC_INTRA_AREA_RIGHT_MSB2	intra Area Right. bit[9]
21 SW_ENC_INTRA_AREA_TOP_MSB2	intra Area Top. bit[9]
20 SW_ENC_INTRA_AREA_BOTTOM_MSB2	intra Area Bottom. bit[9]
19 SW_ENC_ROI1_LEFT_MSB2	ROI1 Area Left. bit[9]
18 SW_ENC_ROI1_RIGHT_MSB2	ROI1 Area Right. bit[9]
17 SW_ENC_ROI1_TOP_MSB2	ROI1 Area Top. bit[9]

Table continues on the next page...

Field	Function
16 SW_ENC_ROI1_BOTTOM_MSB2	ROI1 Area Bottom. bit[9]
15 SW_ENC_ROI2_LEFT_MSB2	ROI2 Area Left. bit[9]
14 SW_ENC_ROI2_RIGHT_MSB2	ROI2 Area Right. bit[9]
13 SW_ENC_ROI2_TOP_MSB2	ROI2 Area Top. bit[9]
12 SW_ENC_ROI2_BOTTOM_MSB2	ROI2 Area Bottom. bit[9]
11 SW_ENC_PIC_WIDTH_MSB2	Encoded width. lumWidthMsb2(unit 8 pixels). bit[12]
10 SW_ENC_IPCM1_LEFT_MSB	IPCM1 Area Left bit[9]
9 SW_ENC_IPCM1_RIGHT_MSB	IPCM1 Area Right bit[9]
8 SW_ENC_IPCM1_TOP_MSB	IPCM1 Area Top bit[9]
7 SW_ENC_IPCM1_BOTTOM_MSB	IPCM2 Area Bottom bit[9]
6 SW_ENC_IPCM2_LEFT_MSB	IPCM2 Area Left bit[9]
5 SW_ENC_IPCM2_RIGHT_MSB	IPCM2 Area Right bit[9]
4 SW_ENC_IPCM2_TOP_MSB	IPCM2 Area Top bit[9]
3	IPCM1 Area Bottom bit[9]

Table continues on the next page...

Field	Function
SW_ENC_IPCM2_BOTTOM_MSB	
2-0 —	Reserved

16.3.5.1.194 Global MV Control 0 (swreg250)

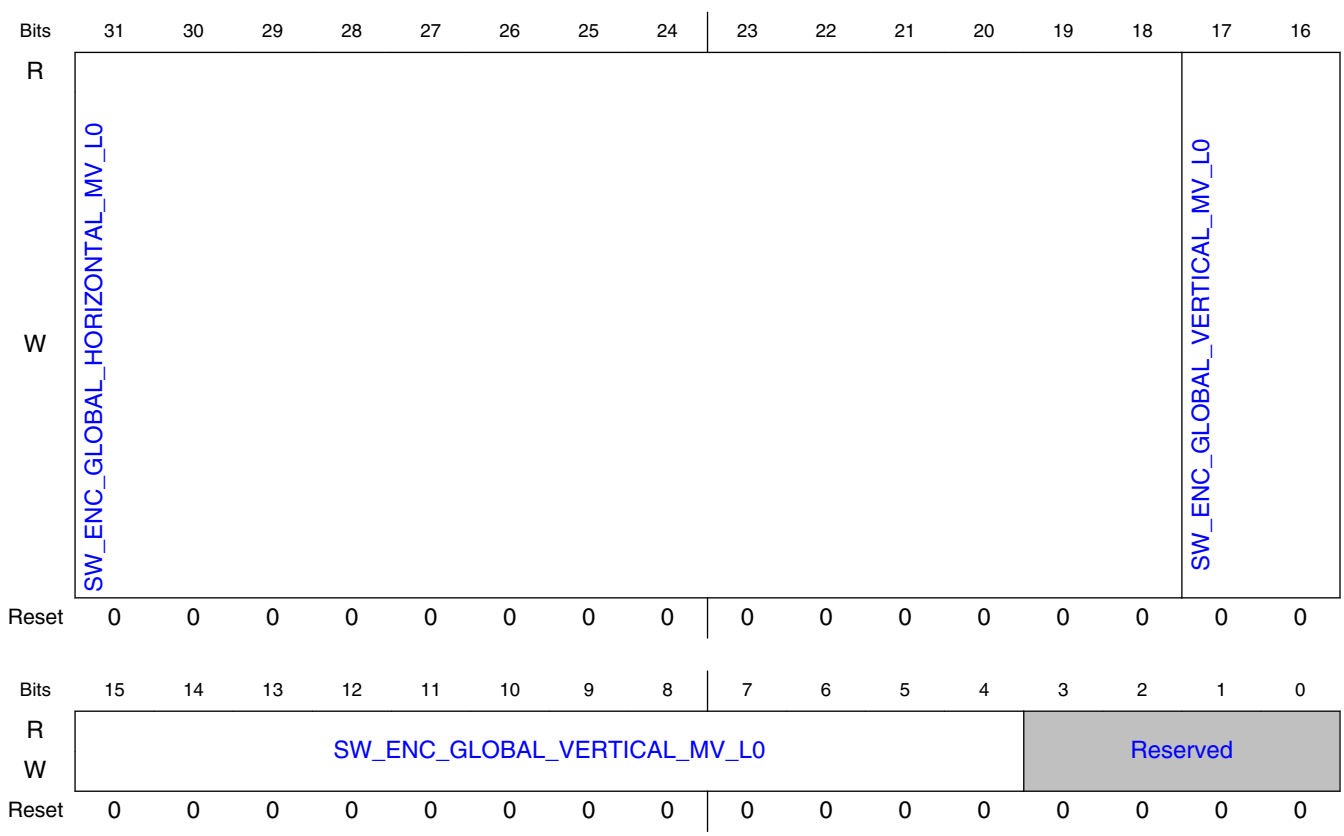
16.3.5.1.194.1 Offset

Register	Offset
swreg250	3E8h

16.3.5.1.194.2 Function

Global MV Control 0

16.3.5.1.194.3 Diagram



16.3.5.1.194.4 Fields

Field	Function
31-18 SW_ENC_GLOBAL_HORIZONTAL_MV_L0	Global horizontal MV for LIST0 in integer pixel.
17-4 SW_ENC_GLOBAL_VERTICAL_MV_L0	Global vertical MV for LIST0 in integer pixel.
3-0 —	Reserved

16.3.5.1.195 Global MV Control 1 (swreg251)

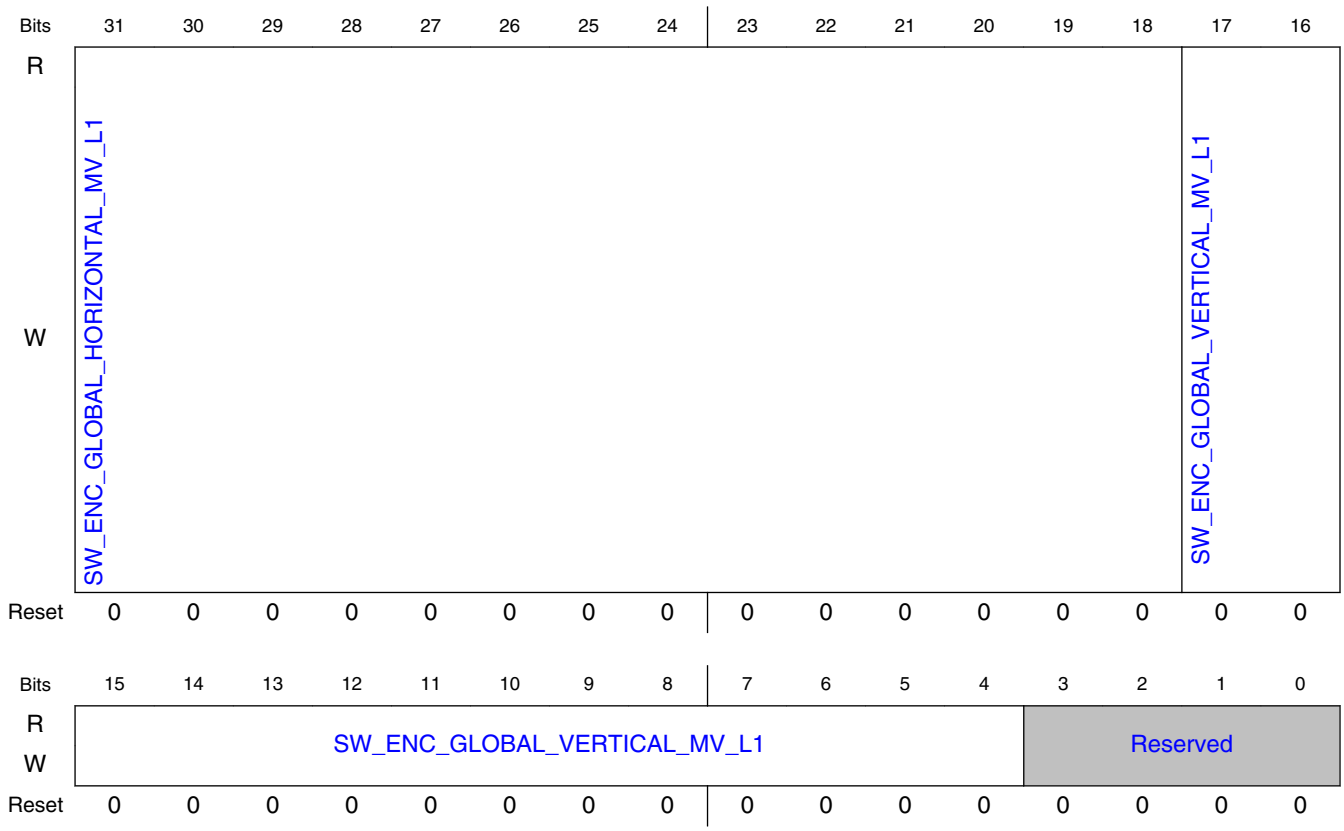
16.3.5.1.195.1 Offset

Register	Offset
swreg251	3ECh

16.3.5.1.195.2 Function

Global MV Control 1

16.3.5.1.195.3 Diagram



16.3.5.1.195.4 Fields

Field	Function
31-18 SW_ENC_GLOBAL_HORIZONTAL_MV_L1	Global horizontal MV for LIST1 in integer pixel.
17-4 SW_ENC_GLOBAL_VERTICAL_MV_L1	Global vertical MV for LIST1 in integer pixel.
3-0 —	Reserved

16.3.5.1.196 ROI3 Area (swreg252)

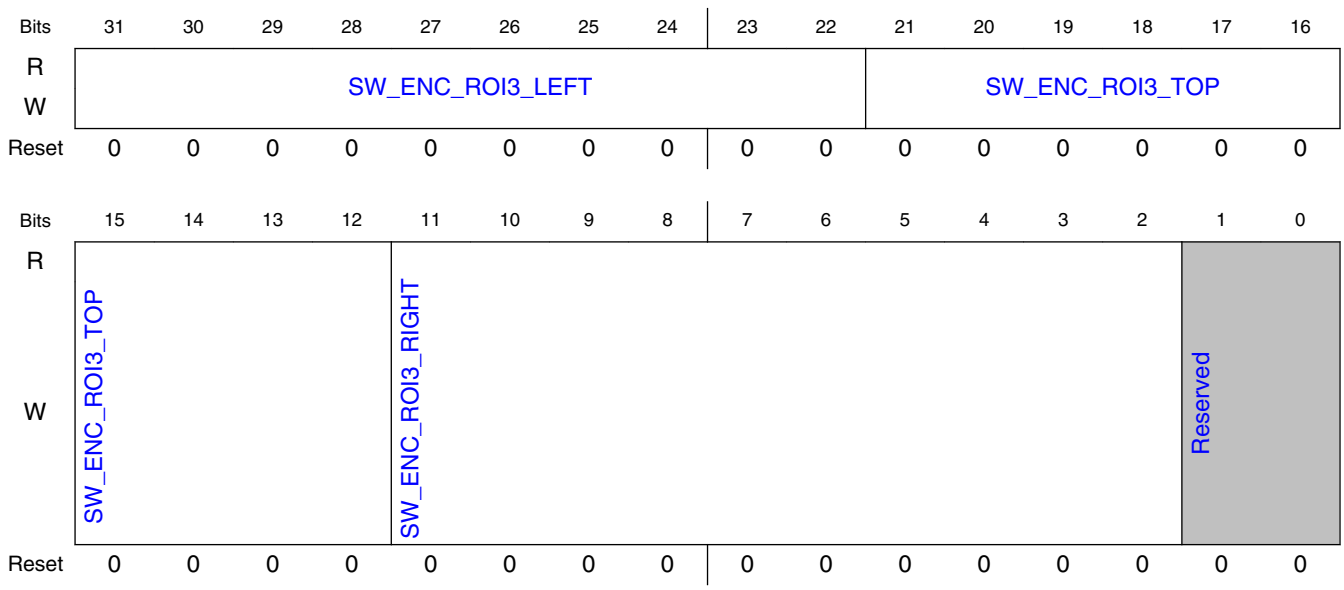
16.3.5.1.196.1 Offset

Register	Offset
swreg252	3F0h

16.3.5.1.196.2 Function

ROI3 Area

16.3.5.1.196.3 Diagram



16.3.5.1.196.4 Fields

Field	Function
31-22 SW_ENC_ROI3_LEFT	ROI3 Area Left
21-12 SW_ENC_ROI3_TOP	ROI3 Area Top
11-2 SW_ENC_ROI3_RIGHT	ROI3 Area Right
1-0 —	Reserved

16.3.5.1.197 ROI3&4 Area (swreg253)

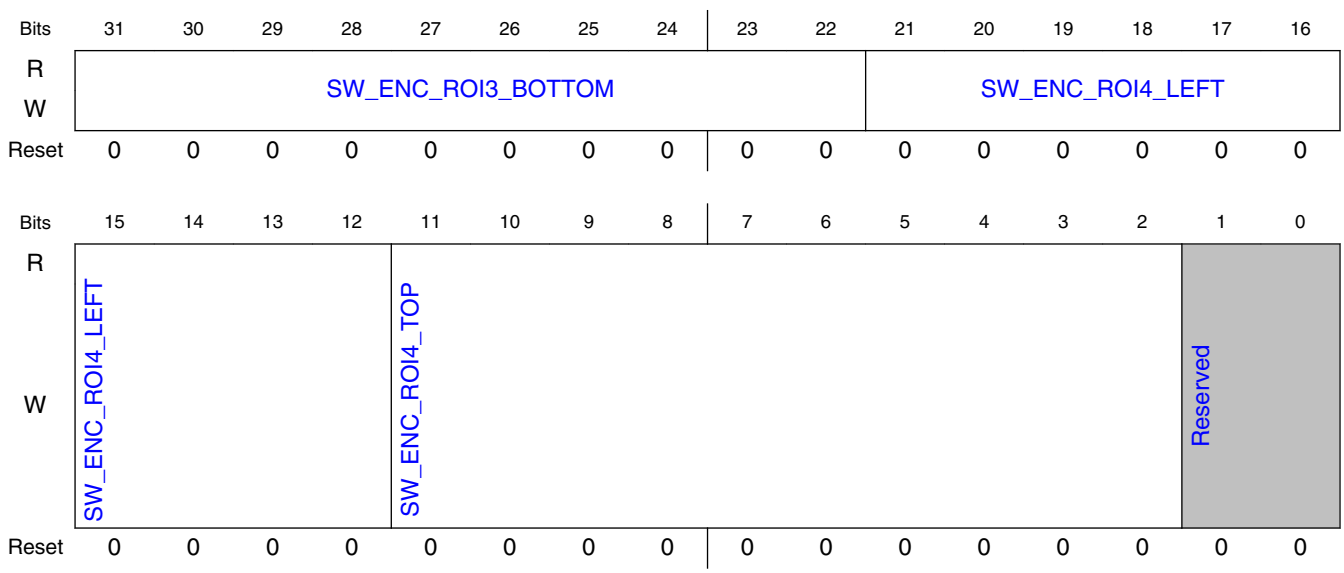
16.3.5.1.197.1 Offset

Register	Offset
swreg253	3F4h

16.3.5.1.197.2 Function

ROI3&4 Area

16.3.5.1.197.3 Diagram



16.3.5.1.197.4 Fields

Field	Function
31-22 SW_ENC_ROI3_BOTTOM	ROI3 Area Bottom
21-12 SW_ENC_ROI4_LEFT	ROI4 Area Left
11-2	ROI4 Area Top

Table continues on the next page...

Field	Function
SW_ENC_ROI4_TOP	
1-0 —	Reserved

16.3.5.1.198 ROI4&5 Area (swreg254)

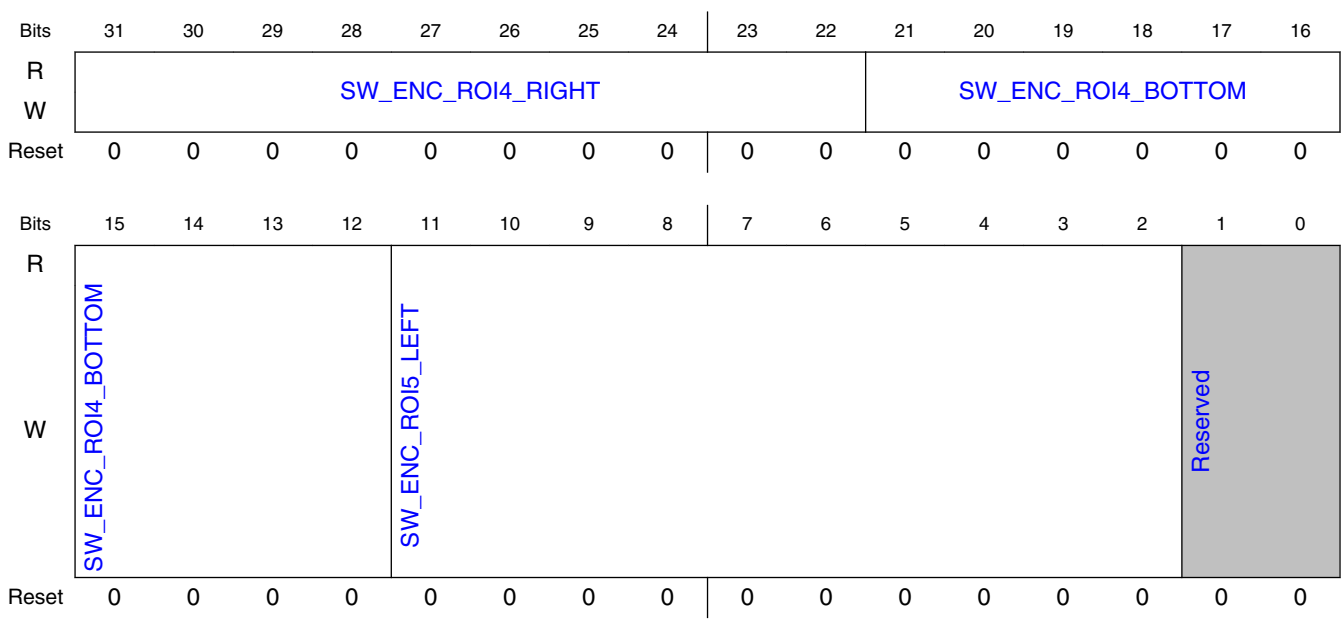
16.3.5.1.198.1 Offset

Register	Offset
swreg254	3F8h

16.3.5.1.198.2 Function

ROI4&5 Area

16.3.5.1.198.3 Diagram



16.3.5.1.198.4 Fields

Field	Function
31-22	ROI4 Area Right

Table continues on the next page...

Field	Function
SW_ENC_ROI4_RIGHT	
21-12 SW_ENC_ROI4_BOTTOM	ROI4 Area Bottom
11-2 SW_ENC_ROI5_LEFT	ROI5 Area Left
1-0 —	Reserved

16.3.5.1.199 ROI5 Area (swreg255)

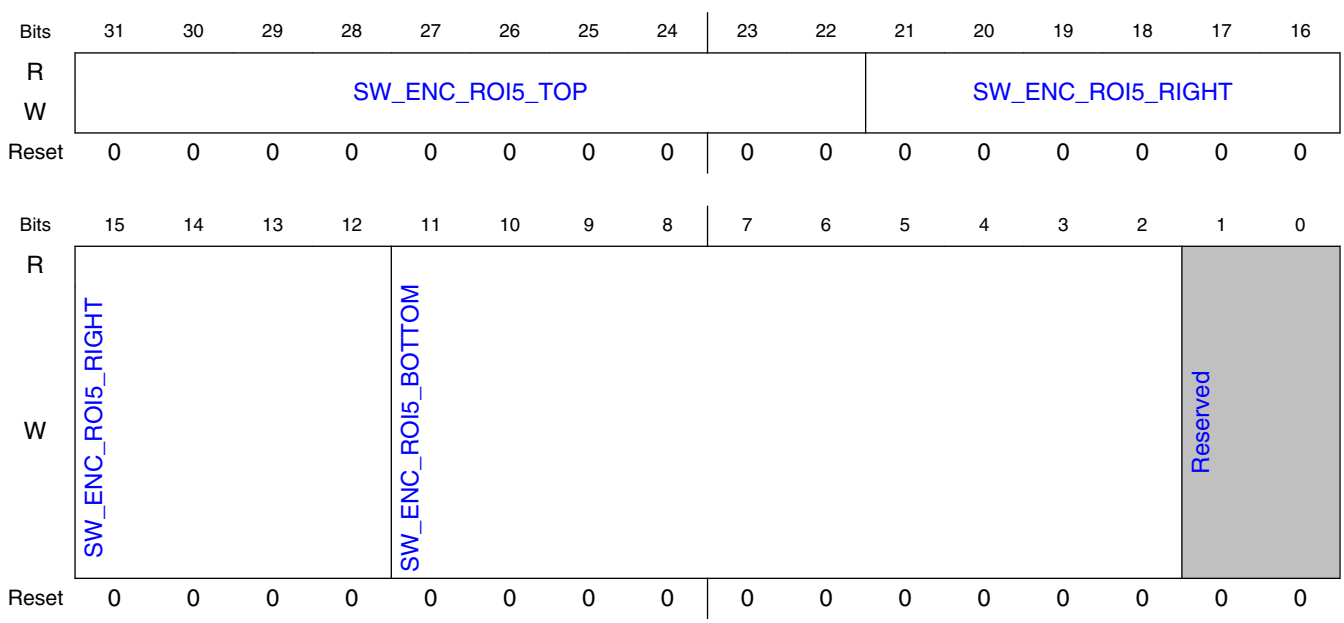
16.3.5.1.199.1 Offset

Register	Offset
swreg255	3FCh

16.3.5.1.199.2 Function

ROI5 Area

16.3.5.1.199.3 Diagram



16.3.5.1.199.4 Fields

Field	Function
31-22 SW_ENC_ROI5_TOP	ROI5 Area Top
21-12 SW_ENC_ROI5_RIGHT	ROI5 Area Right
11-2 SW_ENC_ROI5_BOTTOM	ROI5 Area Bottom
1-0 —	Reserved

16.3.5.1.200 ROI6 Area (swreg256)

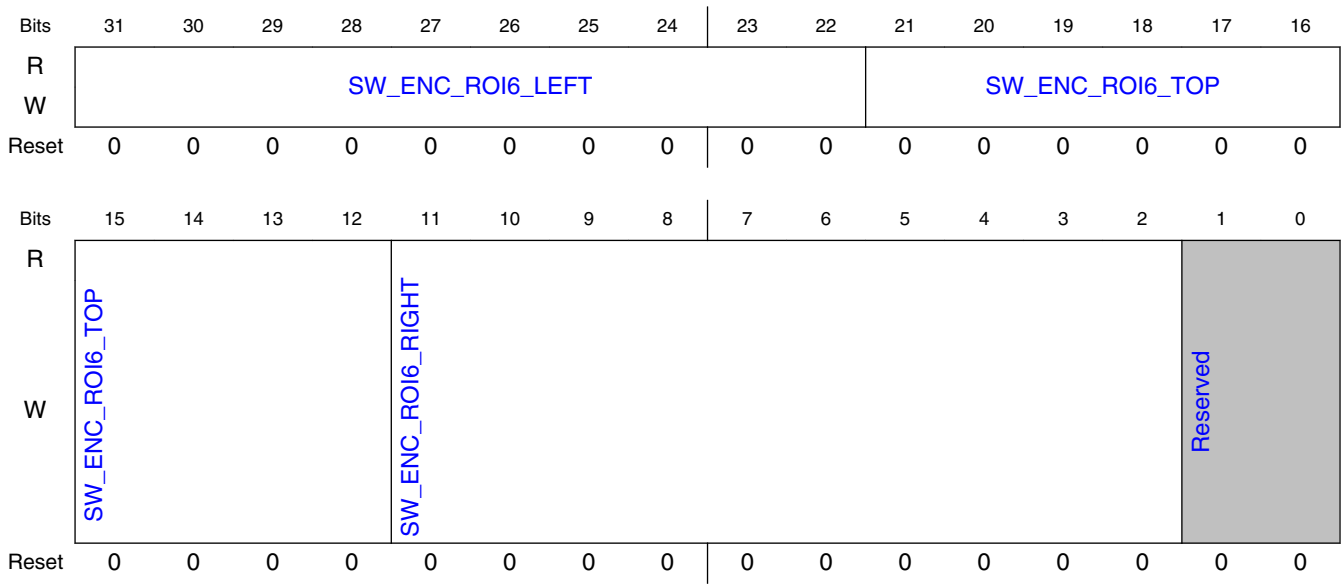
16.3.5.1.200.1 Offset

Register	Offset
swreg256	400h

16.3.5.1.200.2 Function

ROI6 Area

16.3.5.1.200.3 Diagram



16.3.5.1.200.4 Fields

Field	Function
31-22 SW_ENC_ROI6_LEFT	ROI6 Area Left
21-12 SW_ENC_ROI6_TOP	ROI6 Area Top
11-2 SW_ENC_ROI6_RIGHT	ROI6 Area Right
1-0 —	Reserved

16.3.5.1.201 ROI6&7 Area (swreg257)

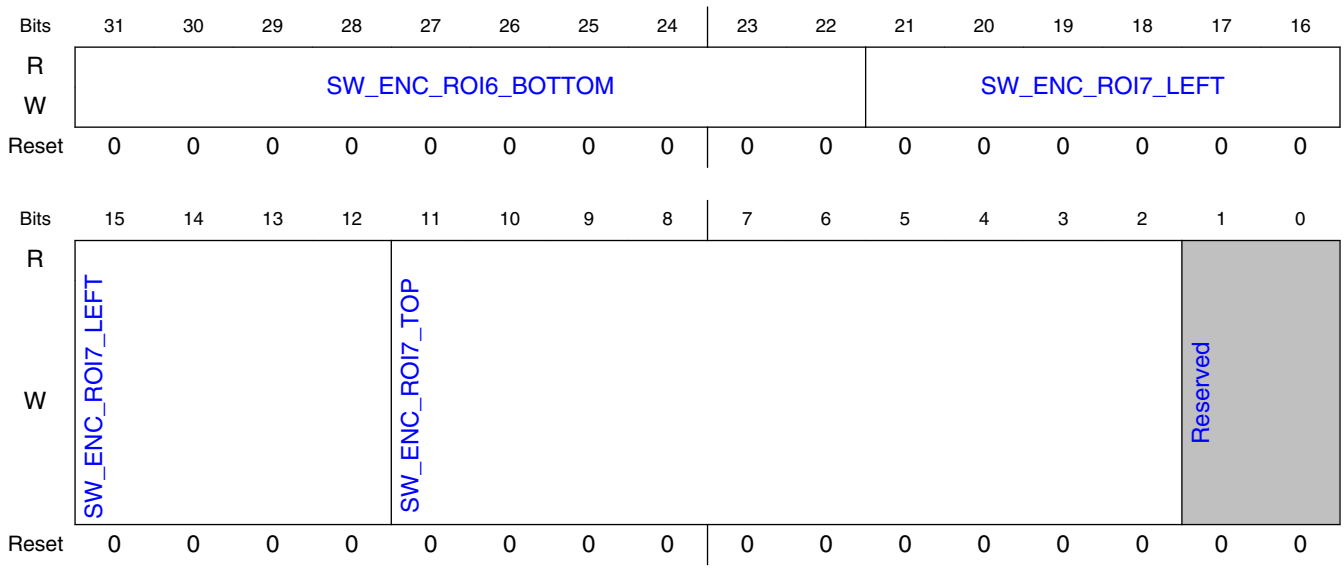
16.3.5.1.201.1 Offset

Register	Offset
swreg257	404h

16.3.5.1.201.2 Function

ROI6&7 Area

16.3.5.1.201.3 Diagram



16.3.5.1.201.4 Fields

Field	Function
31-22 SW_ENC_ROI6_BOTTOM	ROI6 Area Bottom
21-12 SW_ENC_ROI7_LEFT	ROI7 Area Left
11-2 SW_ENC_ROI7_TOP	ROI7 Area Top
1-0 —	Reserved

16.3.5.1.202 ROI7&8 Area (swreg258)

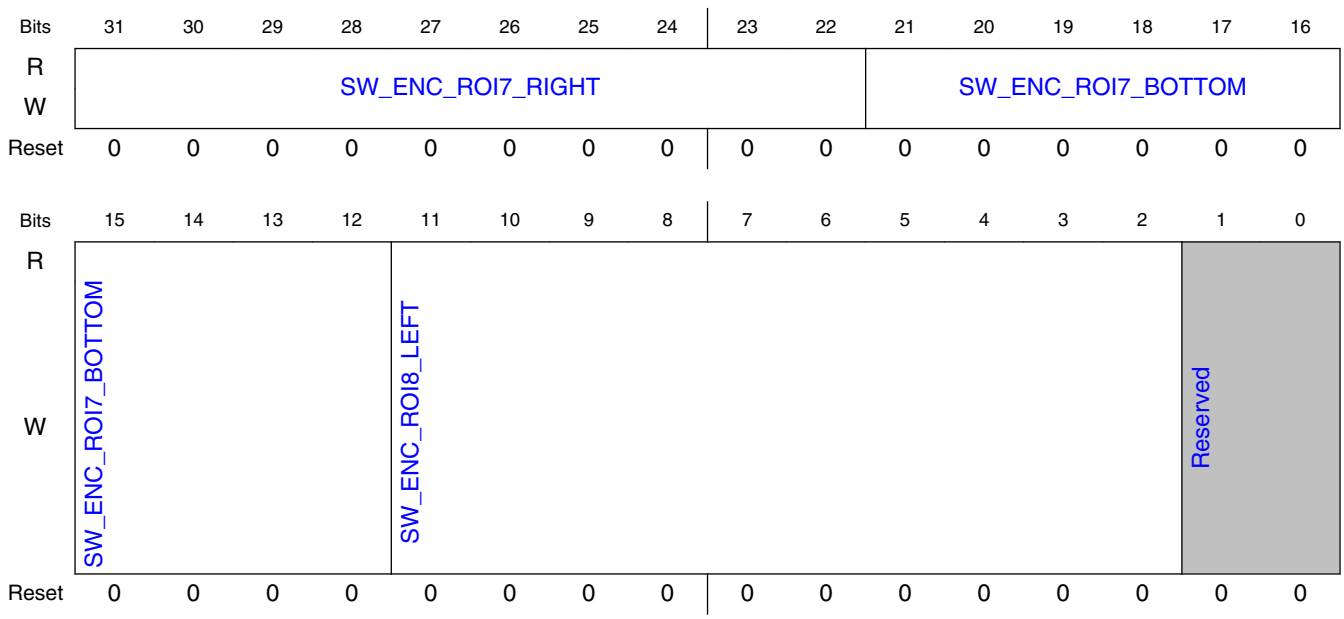
16.3.5.1.202.1 Offset

Register	Offset
swreg258	408h

16.3.5.1.202.2 Function

ROI7&8 Area

16.3.5.1.202.3 Diagram



16.3.5.1.202.4 Fields

Field	Function
31-22 SW_ENC_ROI7_RIGHT	ROI7 Area Right
21-12 SW_ENC_ROI7_BOTTOM	ROI7 Area Bottom
11-2 SW_ENC_ROI8_LEFT	ROI8 Area Left
1-0 —	Reserved

16.3.5.1.203 ROI8 Area (swreg259)

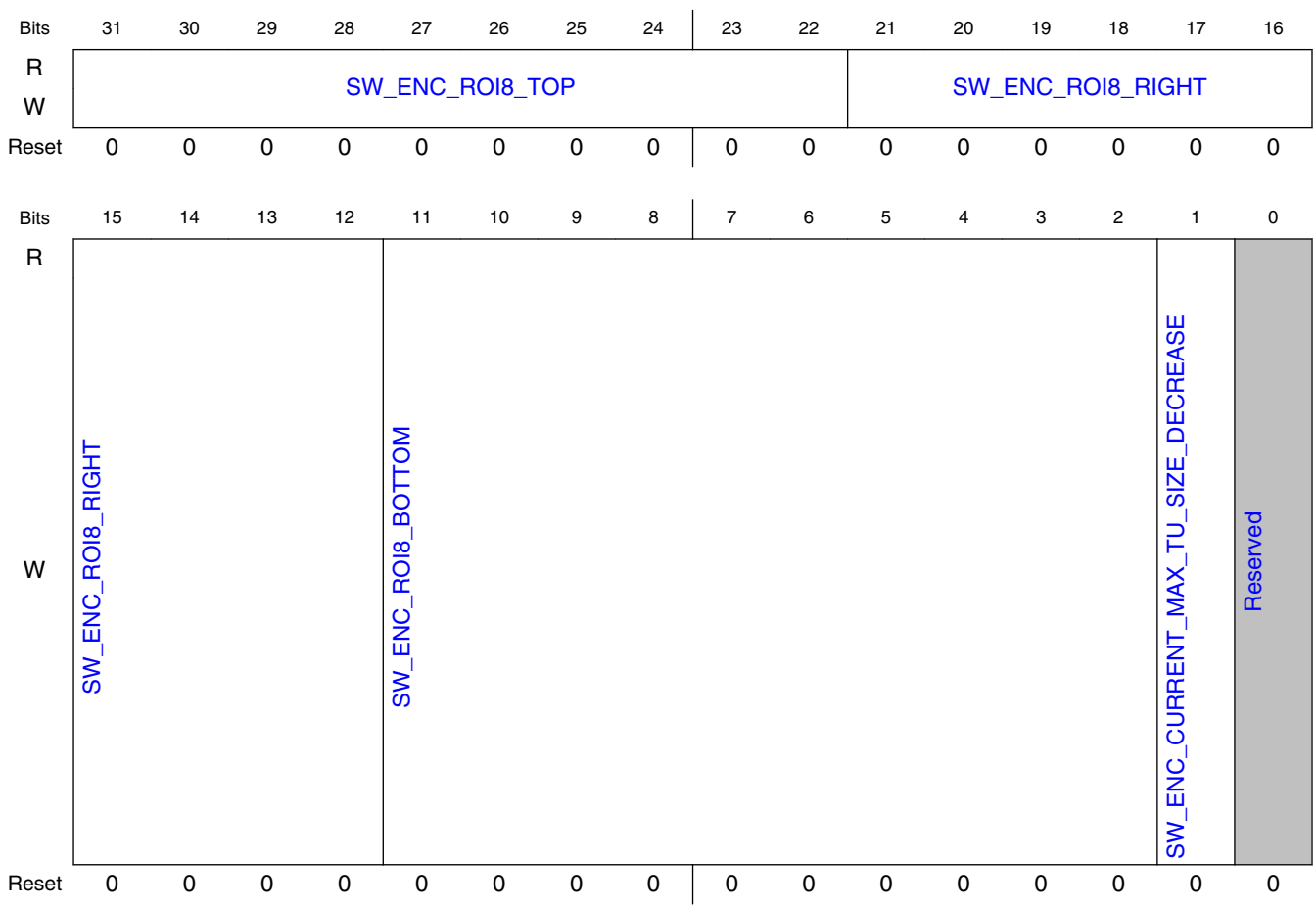
16.3.5.1.203.1 Offset

Register	Offset
swreg259	40Ch

16.3.5.1.203.2 Function

ROI8 Area

16.3.5.1.203.3 Diagram



16.3.5.1.203.4 Fields

Field	Function
31-22 SW_ENC_ROI8_TOP	ROI8 Area Top
21-12 SW_ENC_ROI8_RIGHT	ROI8 Area Right
11-2 SW_ENC_ROI8_BOTTOM	ROI8 Area Bottom
1 SW_ENC_CURRENT_MAX_TU_SIZE_DECREASE	Decrease max tu size in current frame.[0] 0b - max tu size 32. 1b - max tu size 16.
0 —	Reserved

16.3.5.1.204 ROI qp (swreg260)

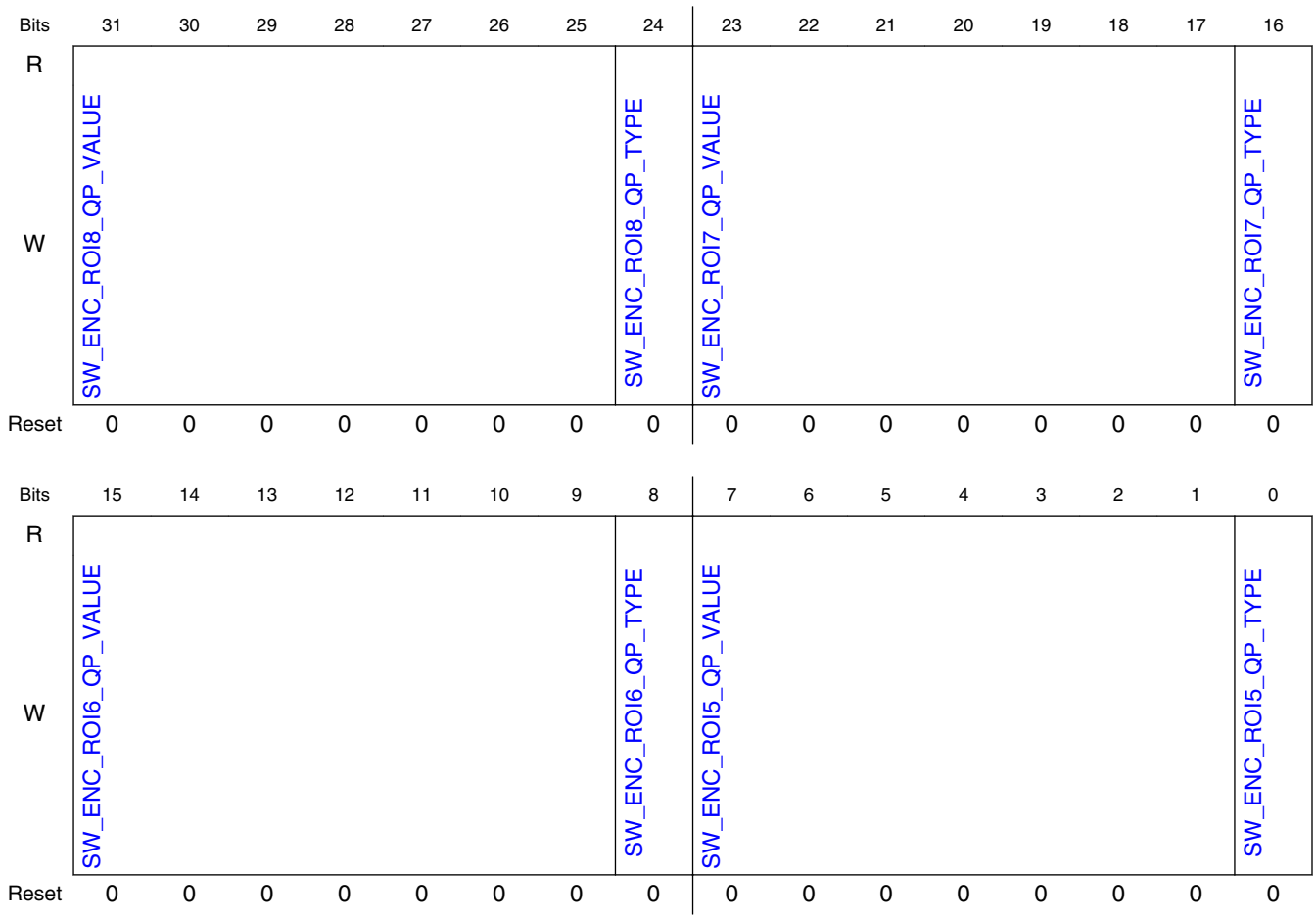
16.3.5.1.204.1 Offset

Register	Offset
swreg260	410h

16.3.5.1.204.2 Function

ROI qp

16.3.5.1.204.3 Diagram



16.3.5.1.204.4 Fields

Field	Function
31-25 SW_ENC_ROI8_QP_VALUE	ROI8 qp value. delta or absolute value
24 SW_ENC_ROI8_QP_TYPE	ROI8 qp type. 0b - delta 1b - Absolute value
23-17 SW_ENC_ROI7_QP_VALUE	ROI7 qp value. delta or absolute value
16 SW_ENC_ROI7_QP_TYPE	ROI7 qp type. 0b - delta 1b - Absolute value
15-9	ROI6 qp value. delta or absolute value

Table continues on the next page...

Field	Function
SW_ENC_ROI6 _QP_VALUE	
8 SW_ENC_ROI6 _QP_TYPE	ROI6 qp type. 0b - delta 1b - Absolute value
7-1 SW_ENC_ROI5 _QP_VALUE	ROI5 qp value. delta or absolute value
0 SW_ENC_ROI5 _QP_TYPE	ROI5 qp type. 0b - delta 1b - Absolute value

16.3.5.1.205 Stride Control (swreg261)

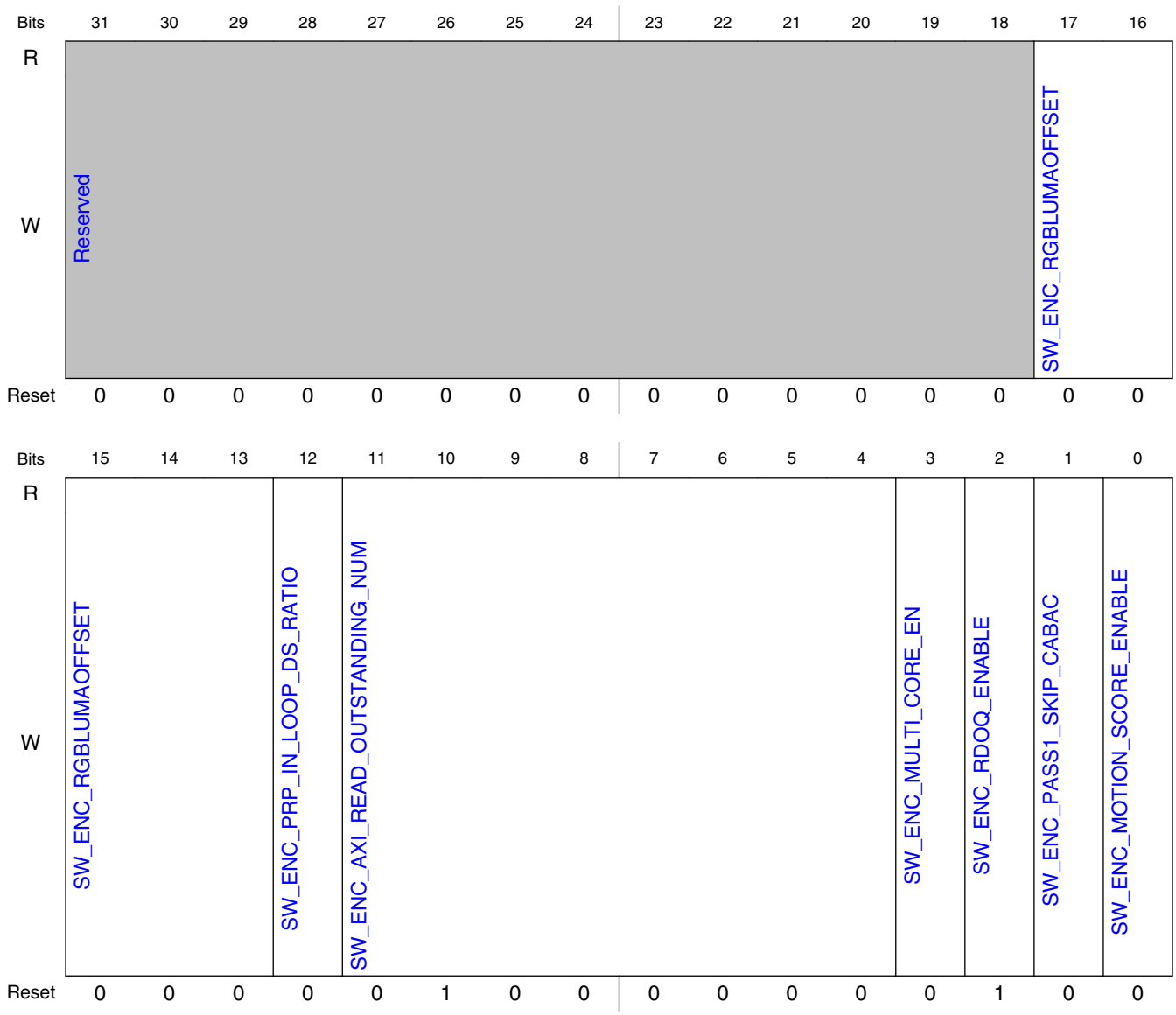
16.3.5.1.205.1 Offset

Register	Offset
swreg261	414h

16.3.5.1.205.2 Function

Stride Control

16.3.5.1.205.3 Diagram



16.3.5.1.205.4 Fields

Field	Function
31-18 —	Reserved
17-13 SW_ENC_RGBLUMAOFFSET	RGB to YUV conversion luma offset
12	down-scaling ratio of In-loop downscaler in prp

Table continues on the next page...

Field	Function
SW_ENC_PRP_IN_LOOP_DS_RATIO	
11-4 SW_ENC_AXI_READ_OUTSTANDING_NUM	AXI outstanding number of read operation
3 SW_ENC_MULTI_CORE_EN	Enable multi-core encoding mode
2 SW_ENC_RDOQ_ENABLE	Enable RDOQ
1 SW_ENC_PASS1_SKIP_CABAC	CABAC SKIP for pass one.
0 SW_ENC_MOTION_SCORE_ENABLE	Enable motion score computing for 2-pass Agop

16.3.5.1.206 Multicore sync ctrl (swreg265)

16.3.5.1.206.1 Offset

Register	Offset
swreg265	424h

16.3.5.1.206.2 Function

Multicore sync ctrl

16.3.5.1.206.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REF_READY_THRESHOLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_DDR_POLLING_INTERVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.206.4 Fields

Field	Function
31-16 SW_ENC_REF_READY_THRESHOLD	EXTRA_LINES_NUM bigger vertical search window ready
15-0 SW_ENC_DDR_POLLING_INTERVAL	Interval cycles for HW reading the DDR RECON_PIXELLINES_READY when value not bigger than search window.

16.3.5.1.207 Multicore sync address L0 LSB (swreg266)

16.3.5.1.207.1 Offset

Register	Offset
swreg266	428h

16.3.5.1.207.2 Function

Multicore sync address L0 LSB

16.3.5.1.207.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MULTICORE_SYNC_L0_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MULTICORE_SYNC_L0_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.207.4 Fields

Field	Function
31-0 SW_ENC_MULTICORE_SYNC_L0_ADDR	For multicore sync. Address LSB of RECON_PIXELLINES_READY for ref buffer L0

16.3.5.1.208 Multicore sync address L0 MSB (swreg267)

16.3.5.1.208.1 Offset

Register	Offset
swreg267	42Ch

16.3.5.1.208.2 Function

Multicore sync address L0 MSB

16.3.5.1.208.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MULTICORE_SYNC_L0_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MULTICORE_SYNC_L0_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.208.4 Fields

Field	Function
31-0 SW_ENC_MULTICORE_SYNC_L0_ADDR_MSB	For multicore sync. Address MSB of RECON_PIXELLINES_READY for ref buffer L0

16.3.5.1.209 Multicore sync address L1 LSB (swreg268)

16.3.5.1.209.1 Offset

Register	Offset
swreg268	430h

16.3.5.1.209.2 Function

Multicore sync address L1 LSB

16.3.5.1.209.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MULTICORE_SYNC_L1_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MULTICORE_SYNC_L1_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.209.4 Fields

Field	Function
31-0 SW_ENC_MULTICORE_SYNC_L1_ADDR	For multicore sync. Address LSB of RECON_PIXELLINES_READY for ref buffer L1

16.3.5.1.210 Multicore sync address L1 MSB (swreg269)

16.3.5.1.210.1 Offset

Register	Offset
swreg269	434h

16.3.5.1.210.2 Function

Multicore sync address L1 MSB

16.3.5.1.210.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MULTICORE_SYNC_L1_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MULTICORE_SYNC_L1_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.210.4 Fields

Field	Function
31-0 SW_ENC_MULTICORE_SYNC_L1_ADDR_MSB	For multicore sync. Address MSB of RECON_PIXELLINES_READY for ref buffer L1

16.3.5.1.211 Multicore sync address recon LSB (swreg270)

16.3.5.1.211.1 Offset

Register	Offset
swreg270	438h

16.3.5.1.211.2 Function

Multicore sync address recon LSB

16.3.5.1.211.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MULTICORE_SYNC_REC_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MULTICORE_SYNC_REC_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.211.4 Fields

Field	Function
31-0 SW_ENC_MULTICORE_SYNC_REC_ADDR	For multicore sync. Address LSB of RECON_PIXELLINES_READY for recon buffer

16.3.5.1.212 Multicore sync address recon MSB (swreg271)

16.3.5.1.212.1 Offset

Register	Offset
swreg271	43Ch

16.3.5.1.212.2 Function

Multicore sync address recon MSB

16.3.5.1.212.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MULTICORE_SYNC_REC_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MULTICORE_SYNC_REC_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.212.4 Fields

Field	Function
31-0 SW_ENC_MULTICORE_SYNC_REC_ADDR_MSB	For multicore sync. Address MSB of RECON_PIXELLINES_READY for recon buffer

16.3.5.1.213 Programmable AXI urgent sideband signals (swreg272)

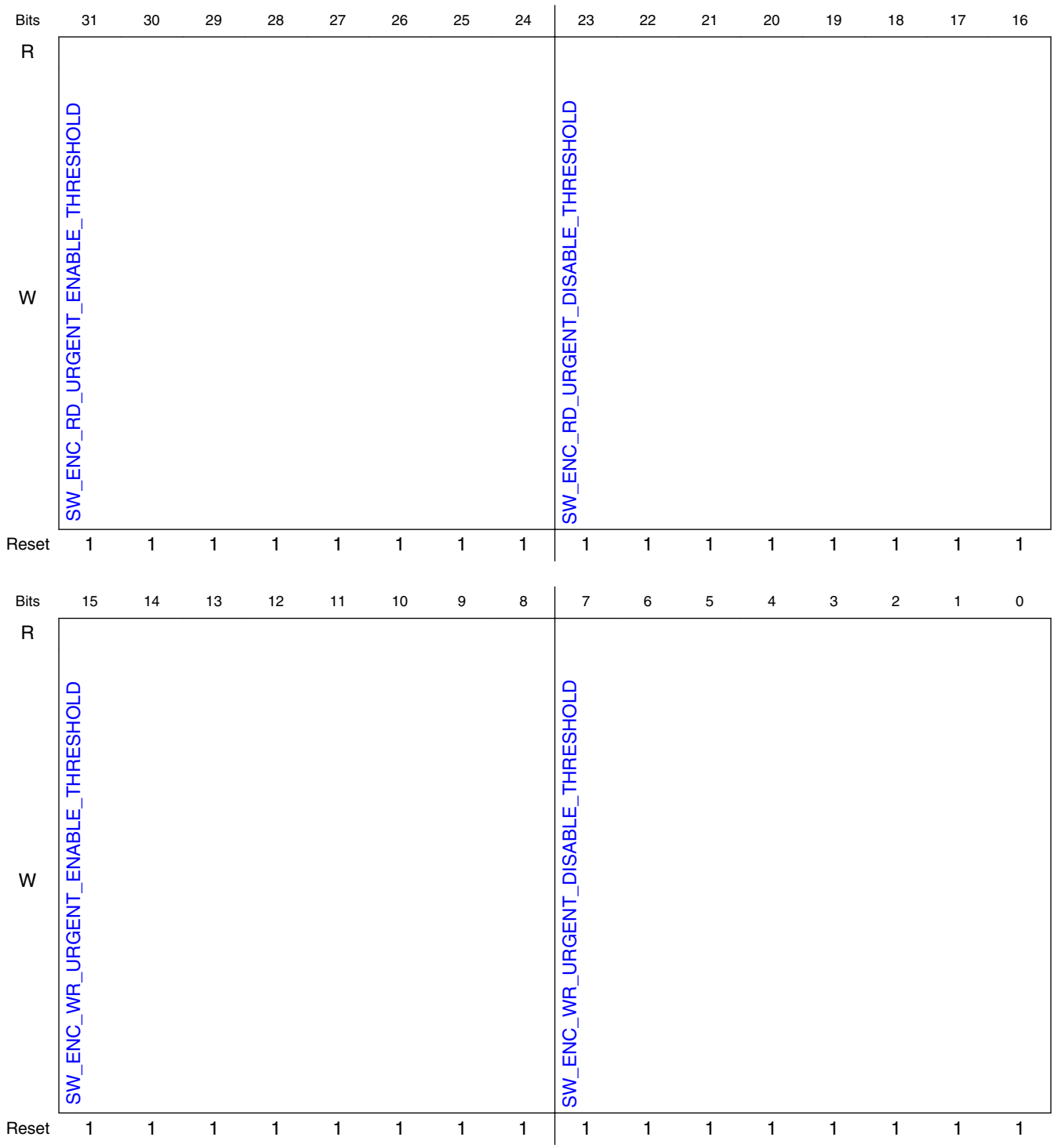
16.3.5.1.213.1 Offset

Register	Offset
swreg272	440h

16.3.5.1.213.2 Function

Programmable AXI urgent sideband signals

16.3.5.1.213.3 Diagram



16.3.5.1.213.4 Fields

Field	Function
31-24 SW_ENC_RD_URGENT_ENABLE_THRESHOLD	axi_rd_urgent=1 when total number of rd cmds is bigger or equal to threshold.255=disable
23-16 SW_ENC_RD_URGENT_DISABLE_THRESHOLD	axi_rd_urgent=0 when total number of rd cmds is smaller than threshold.255=disable
15-8 SW_ENC_WR_URGENT_ENABLE_THRESHOLD	axi_wr_urgent=1 when total number of wr cmds is bigger or equal to threshold.255=disable
7-0 SW_ENC_WR_URGENT_DISABLE_THRESHOLD	axi_wr_urgent=0 when total number of wr cmds is smaller than threshold.255=disable

16.3.5.1.214 roimap cu ctrl index address LSB (swreg273)

16.3.5.1.214.1 Offset

Register	Offset
swreg273	444h

16.3.5.1.214.2 Function

roimap cu ctrl index address LSB

16.3.5.1.214.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROIMAP_CUCTRL_INDEX_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROIMAP_CUCTRL_INDEX_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.214.4 Fields

Field	Function
31-0 SW_ENC_ROIMAP_CUCTRL_INDEX_ADDR	Address LSB of roi map cu ctrlindex buffer

16.3.5.1.215 roimap cu ctrl index address MSB (swreg274)

16.3.5.1.215.1 Offset

Register	Offset
swreg274	448h

16.3.5.1.215.2 Function

roimap cu ctrl index address MSB

16.3.5.1.215.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROIMAP_CUCTRL_INDEX_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROIMAP_CUCTRL_INDEX_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.215.4 Fields

Field	Function
31-0 SW_ENC_ROIMAP_CUCTRL_INDEX_ADDR_MSB	Address MSB of roi map cu ctrl index buffer

16.3.5.1.216 roimap cu ctrl address LSB (swreg275)

16.3.5.1.216.1 Offset

Register	Offset
swreg275	44Ch

16.3.5.1.216.2 Function

roimap cu ctrl address LSB

16.3.5.1.216.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROIMAP_CUCTRL_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROIMAP_CUCTRL_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.216.4 Fields

Field	Function
31-0 SW_ENC_ROIMAP_CUCTRL_ADDR	Address LSB of roi map cu ctrl buffer

16.3.5.1.217 roimap cu ctrl address MSB (swreg276)

16.3.5.1.217.1 Offset

Register	Offset
swreg276	450h

16.3.5.1.217.2 Function

roimap cu ctrl address MSB

16.3.5.1.217.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROIMAP_CUCTRL_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROIMAP_CUCTRL_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.217.4 Fields

Field	Function
31-0 SW_ENC_ROIMAP_CUCTRL_ADDR_MSB	Address MSB of roi map cu ctrl buffer

16.3.5.1.218 poc type/bits setting (swreg277)

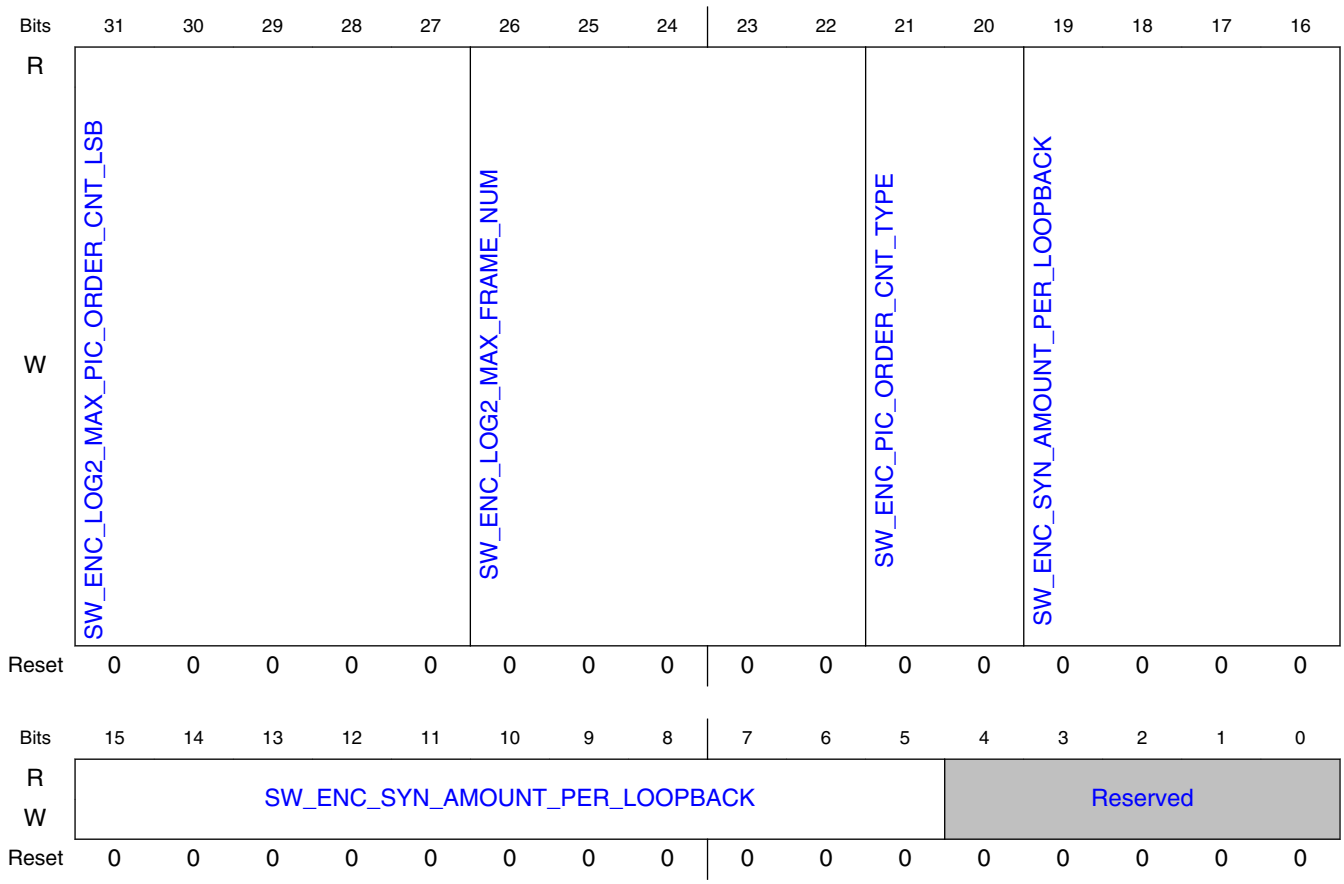
16.3.5.1.218.1 Offset

Register	Offset
swreg277	454h

16.3.5.1.218.2 Function

poc type/bits setting

16.3.5.1.218.3 Diagram



16.3.5.1.218.4 Fields

Field	Function
31-27 SW_ENC_LOG2_MAX_PIC_ORDER_CNT_LSB	number of bits in pic_order_cnt_lsb
26-22 SW_ENC_LOG2_MAX_FRAME_NUM	number of bits in frameNum
21-20 SW_ENC_PIC_ORDER_CNT_TYPE	poc type
19-5	Handshake sync amount for every loopback

Table continues on the next page...

Field	Function
SW_ENC_SYN_AMOUNT_PER_LOOPBACK	
4-0 —	Reserved

16.3.5.1.219 stream output buffer1 address (swreg278)

16.3.5.1.219.1 Offset

Register	Offset
swreg278	458h

16.3.5.1.219.2 Function

stream output buffer1 address

16.3.5.1.219.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_OUTPUT_STRM_BUF1_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_OUTPUT_STRM_BUF1_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.219.4 Fields

Field	Function
31-0 SW_ENC_OUTPUT_STRM_BUF1_BASE	stream output buffer1 base address. Only valid if sw_enc_HWStreamBufChain=1.

16.3.5.1.220 stream output buffer1 limit size (swreg280)

16.3.5.1.220.1 Offset

Register	Offset
swreg280	460h

16.3.5.1.220.2 Function

stream output buffer1 limit size

16.3.5.1.220.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_OUTPUT_STRM_BUFFER1_LIMIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_OUTPUT_STRM_BUFFER1_LIMIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.220.4 Fields

Field	Function
31-0 SW_ENC_OUTPUT_STRM_BUFFER1_LIMIT	Stream buffer1 limit size (bytes). Only valid if sw_enc_HWStreamBufChain=1. If buffer1 limit is reached buffer_full_IRQ will be generated.

16.3.5.1.221 poc type/bits setting (swreg281)

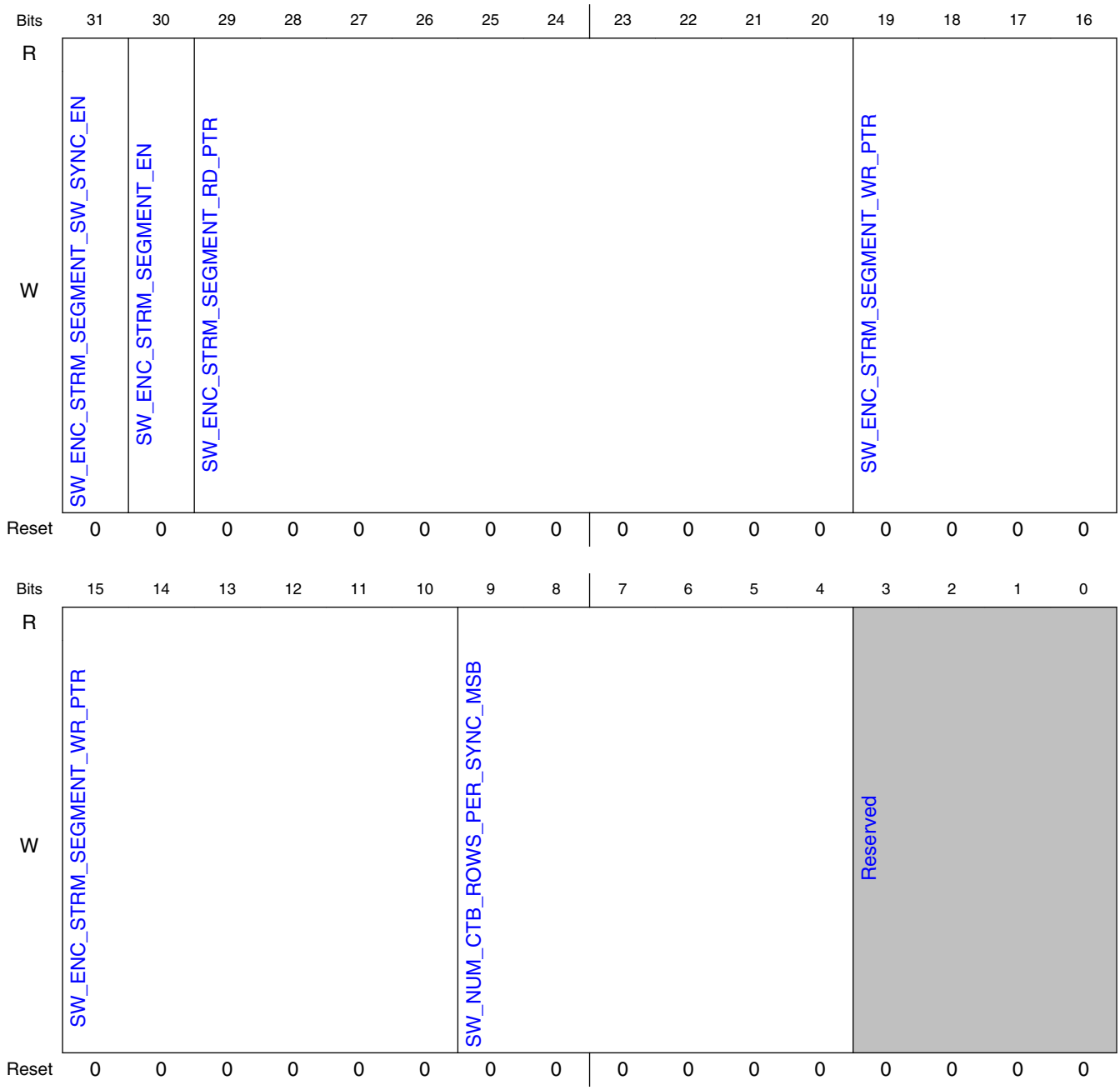
16.3.5.1.221.1 Offset

Register	Offset
swreg281	464h

16.3.5.1.221.2 Function

poc type/bits setting

16.3.5.1.221.3 Diagram



16.3.5.1.221.4 Fields

Field	Function
31 SW_ENC_STRM_SEGMENT_SW_SYNC_EN	Stream segment software handshake enable. Set 0 will disable this handshake
30 SW_ENC_STRM_SEGMENT_EN	Stream segment function enable
29-20 SW_ENC_STRM_SEGMENT_RD_PTR	the number of segments that have been read out from output buffer
19-10 SW_ENC_STRM_SEGMENT_WR_PTR	the number of segments that have been filled into the output buffer
9-4 SW_NUM_CTB_ROWS_PER_SYNC_MSB	Number MSB of CTB rows for every HW sync. Bit[14:9]. Unit is CTB
3-0 —	Reserved

16.3.5.1.222 HW synthesis config register 4, read-only (swreg287)

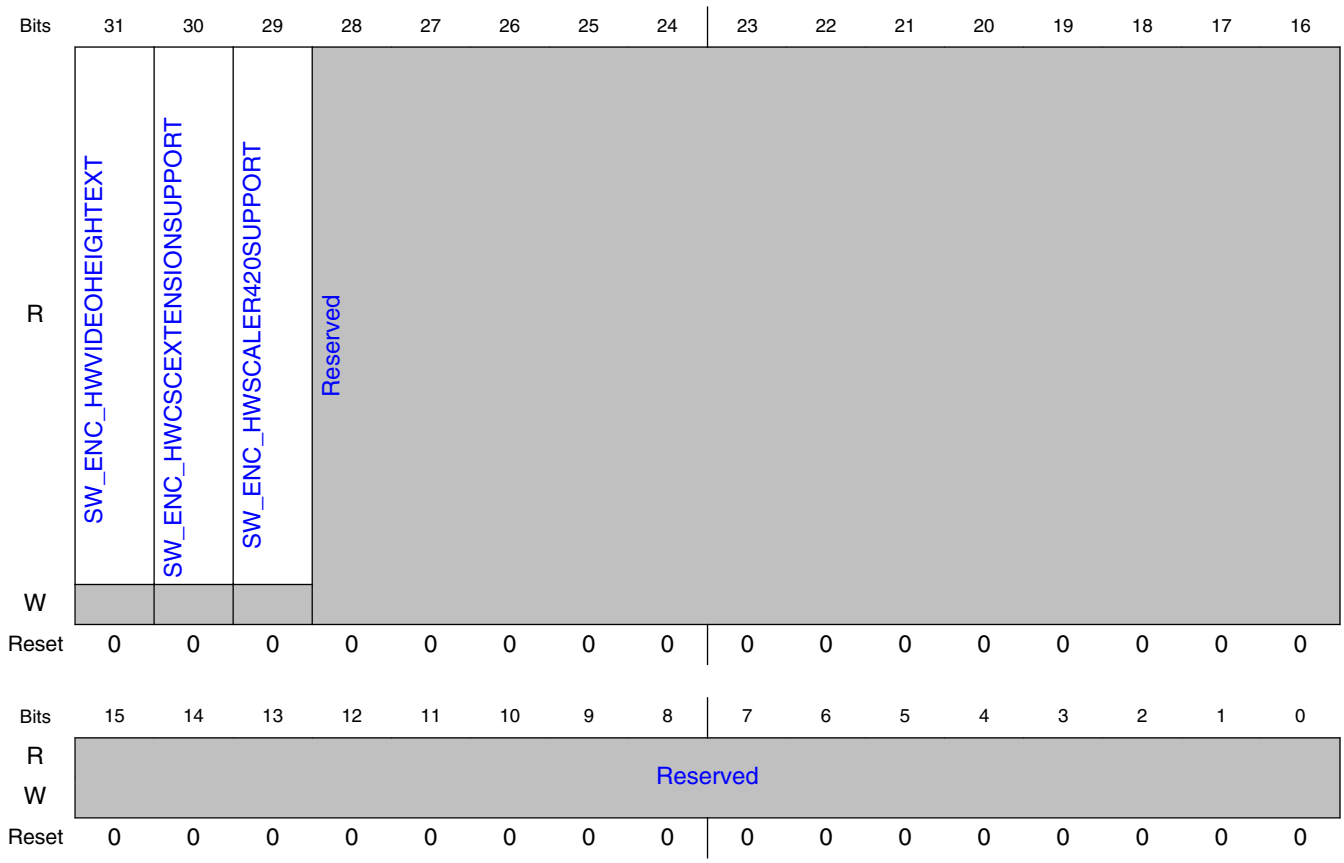
16.3.5.1.222.1 Offset

Register	Offset
swreg287	47Ch

16.3.5.1.222.2 Function

HW synthesis config register 4, read-only

16.3.5.1.222.3 Diagram



16.3.5.1.222.4 Fields

Field	Function
31 SW_ENC_HWVIDEOHEIGHTEXT	Maximum allowed video height extended from 8192 to 8640. 0b - Not. 1b - Yes.
30 SW_ENC_HWCSCEXTENSIONSUPPORT	RGB to YUV conversion extension. 0b - not supported. 1b - supported
29 SW_ENC_HWSCALER420SUPPORT	out-loop scaler output YUV420SP. 0b - not supported. 1b - supported
28-0 —	Reserved

16.3.5.1.223 Pre-processor color conversion parameters1 (swreg289)

16.3.5.1.223.1 Offset

Register	Offset
swreg289	484h

16.3.5.1.223.2 Function

Pre-processor color conversion parameters1

16.3.5.1.223.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RGBCOEFFG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RGBCOEFFH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.1.223.4 Fields

Field	Function
31-16 SW_ENC_RGBCOEFFG	RGB to YUV conversion coefficient G
15-0 SW_ENC_RGBCOEFFH	RGB to YUV conversion coefficient H

16.3.5.2 VC8000E_HEVC register descriptions

16.3.5.2.1 VC8000E_HEVC memory map

VC8000E_HEVC: 3832_0000h

Offset	Register	Width (In bits)	Access	Reset value
4h	Interrupt register encoder (swreg1)	32	RW	0000_0800h
8h	Data configuration register0 (swreg2)	32	RW	0000_0000h
Ch	Data configuration register1 (swreg3)	32	RW	FF00_0000h
10h	control register 0 (swreg4)	32	RW	2190_0014h
14h	control register 1 (swreg5)	32	RW	Table 16-11
1Ch	control register 3 (swreg7)	32	RW	6800_1A00h
20h	stream output buffer0 address (swreg8)	32	RW	0000_0000h
24h	stream output buffer0 limit size (swreg9)	32	RW	0000_0000h
28h	sizeTblBase (swreg10)	32	RW	0000_0000h
2Ch	encoded Picture order count (swreg11)	32	RW	0000_0000h
30h	input lum base address (swreg12)	32	RW	0000_0000h
34h	input cb base address (swreg13)	32	RW	0000_0000h
38h	input cr base address (swreg14)	32	RW	0000_0000h
3Ch	recon image luma base address (swreg15)	32	RW	0000_0000h
40h	recon image chroma base address (swreg16)	32	RW	0000_0000h
48h	reference picture reconstructed list0 luma0 (swreg18)	32	RW	0000_0000h
4Ch	reference picture reconstructed list0 chroma0 (swreg19)	32	RW	0000_0000h
58h	Cyclic Intra (swreg22)	32	RW	0000_0000h
5Ch	intra Area (swreg23)	32	RW	0000_0000h
60h	ROI1 Area (swreg24)	32	RW	0000_0000h
64h	ROI2 Area (swreg25)	32	RW	0000_0000h
68h	intra penalty for picture0. For H2V1 only. (swreg26_H2V1)	32	RW	0000_0000h
68h	intra size factors. For H2V2 or later version. (swreg26_H2V2)	32	RW	0000_0000h
6Ch	intra penalty for picture1. For H2V1 only. (swreg27_H2V1)	32	RW	0000_0000h
6Ch	intra mode factors . For H2V2 or later version. (swreg27_H2V2)	32	RW	0000_0000h
70h	intra penalty for picture2 (swreg28)	32	RW	0000_0000h
70h	inter me SATD lambda config 0. For H2V2 or later version. (swreg28_H2V2)	32	RW	0000_0000h
70h	inter me SATD lambda config 0. For H2V5 or later version. (swreg28_H2V5)	32	RW	0000_0000h
74h	intra penalty for ROI10. For H2V1 only. (swreg29_H2V1)	32	RW	0000_0000h
74h	inter me SATD lambda config 1.For H2V2 or later version. (swreg29_H2V2)	32	RW	0000_0000h
74h	inter me SATD lambda config 1.For H2V5 or later version. (swreg29_H2V5)	32	RW	0000_0000h
78h	intra penalty for ROI11. For H2V1 only. (swreg30_H2V1)	32	RW	0000_0000h
78h	inter me SATD lambda config 2. For H2V2 or later version. (swreg30_H2V2)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
78h	inter me SATD lambda config 2. For H2V5 or later version. (swreg30_H2V5)	32	RW	0000_0000h
7Ch	intra penalty for ROI12. For H2V1 only. (swreg31_H2V1)	32	RW	0000_0000h
7Ch	inter me SATD lambda config 3. For H2V2 or later version. (swreg31_H2V2)	32	RW	0000_0000h
7Ch	inter me SATD lambda config 3. For H2V5 or later version. (swreg31_H2V5)	32	RW	0000_0000h
80h	intra penalty for ROI20. For H2V1 only. (swreg32_H2V1)	32	RW	0000_0000h
80h	inter me SSE lambda config 0. For H2V2 or later version. (swreg32_H2V2)	32	RW	0000_0000h
80h	inter me SATD lambda config 4. For H2V5 or later version. (swreg32_H2V5)	32	RW	0000_0000h
84h	intra penalty for ROI21. For H2V1 only. (swreg33_H2V1)	32	RW	0000_0000h
84h	inter me SSE lambda config 1. For H2V2 or later version. (swreg33_H2V2)	32	RW	0000_0000h
84h	inter me SATD lambda config 5. For H2V5 or later version. (swreg33_H2V5)	32	RW	0000_0000h
88h	intra penalty for ROI22. For H2V1 only. (swreg34_H2V1)	32	RW	0000_0000h
88h	inter me SSE lambda config 2. For H2V2 or later version. (swreg34_H2V2)	32	RW	0000_0000h
88h	inter me SATD lambda config 6. For H2V5 or later version. (swreg34_H2V5)	32	RW	0000_0000h
8Ch	inter prediction parameters1 (swreg35)	32	RW	0000_0000h
90h	inter prediction parameters2 (swreg36)	32	RW	0000_0000h
94h	SAO lambda parameter (swreg37)	32	RW	0000_0000h
98h	Pre-processor configuration (swreg38)	32	RW	0000_0000h
9Ch	Pre-processor color conversion parameters0 (swreg39)	32	RW	0000_0000h
A0h	Pre-processor color conversion parameters1 (swreg40)	32	RW	0000_0000h
A4h	Pre-processor color conversion parameters2 (swreg41)	32	RW	0000_0000h
B8h	compressed coefficients base address for SAN module. (swreg46)	32	RW	0000_0000h
F0h	Base address for recon luma compress table LSB. (swreg60)	32	RW	0000_0000h
F8h	Base address for recon Chroma compress table LSB (swreg62)	32	RW	0000_0000h
100h	Base address for list 0 ref 0 luma compress table LSB. (swreg64)	32	RW	0000_0000h
108h	Base address for list 0 ref 0 Chroma compress table LSB. (swreg66)	32	RW	0000_0000h
120h	Base address for recon luma 4n base LSB. (swreg72)	32	RW	0000_0000h
128h	reference picture reconstructed list0 4n 0 (swreg74)	32	RW	0000_0000h
138h	inter prediction parameters3. For H2V1 only. (swreg78_H2V1)	32	RW	0000_0002h
138h	inter me SSE lambda config 3. For H2V2 or later version. (swreg78_H2V2)	32	RW	0000_0000h
138h	inter me SATD lambda config 7. For H2V5 or later version. (swreg78_H2V5)	32	RW	0000_0000h
13Ch	inter prediction parameters4. For H2V1 only. (swreg79_H2V1)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
13Ch	inter me SSE lambda config 4. For H2V2 or later version. (swreg79_H2V2)	32	RW	0000_0000h
13Ch	inter me SSE lambda config 0. For H2V5 or later version. (swreg79_H2V5)	32	RW	0000_0000h
144h	hardware configuation 0 (swreg81)	32	RW	2000_0000h
148h	record hardware performance (swreg82)	32	RO	0000_0000h
14Ch	reference picture reconstructed list1 luma0 (swreg83)	32	RW	0000_0000h
150h	reference picture reconstructed list1 chroma0 (swreg84)	32	RW	0000_0000h
16Ch	reference pictures list1 config (swreg91)	32	RW	0000_0000h
170h	reference picture reconstructed list1 4n 0 (swreg92)	32	RW	0000_0000h
180h	Base address for list 1 ref 0 luma compress table LSB. (swreg96)	32	RW	0000_0000h
188h	Base address for list 1 ref 0 Chroma compress table LSB. (swreg98)	32	RW	0000_0000h
1A0h	reference picture lists modification (swreg104)	32	RW	0000_0000h
1A8h	Min picture size (swreg106)	32	RW	0000_0000h
1ACh	Max picture size (swreg107)	32	RW	0000_0000h
1B4h	Qp delta map (swreg109)	32	RW	0000_0000h
1BCh	adaptive GOP configuration1 (swreg111)	32	RO	0000_0000h
1C0h	adaptive GOP configuration2 (swreg112)	32	RO	0000_0000h
1C4h	adaptive GOP configuration3 (swreg113)	32	RW	0000_0000h
1DCh	min/max lcu bits number of last picture (swreg119)	32	RW	0000_0000h
1E0h	total bits number of all lcus of last picture not including slice header bits (swreg120)	32	RW	0000_0000h
1E8h	inter me SSE lambda config 5 (swreg122)	32	RW	0000_0000h
1E8h	inter me SSE lambda config 1. For H2V5 or later version. (swreg122_H2V5)	32	RW	0000_0000h
1ECh	inter me SSE lambda config 6 (swreg123)	32	RW	0000_0000h
1ECh	inter me SSE lambda config 2. For H2V5 or later version. (swreg123_H2V5)	32	RW	0000_0000h
1F0h	inter me SSE lambda config 7 (swreg124)	32	RW	0000_0000h
1F0h	inter me SSE lambda config 3. For H2V5 or later version. (swreg124_H2V5)	32	RW	0000_0000h
1F4h	intra SATD lambda config 0 (swreg125)	32	RW	0000_0000h
1F8h	intra SATD lambda config 1 (swreg126)	32	RW	0000_0000h
1FCh	intra SATD lambda config 2 (swreg127)	32	RW	0000_0000h
200h	intra SATD lambda config 3 (swreg128)	32	RW	0000_0000h
204h	intra SATD lambda config 4 (swreg129)	32	RW	0000_0000h
208h	intra SATD lambda config 5 (swreg130)	32	RW	0000_0000h
20Ch	intra SATD lambda config 6 (swreg131)	32	RW	0000_0000h
210h	intra SATD lambda config 7 (swreg132)	32	RW	0000_0000h
214h	SSE devide 256 (swreg133)	32	RW	0000_0000h
218h	noise reduction (swreg134)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
21Ch	noise reduction 1 (swreg135)	32	RW	0000_0000h
220h	noise reduction 2 (swreg136)	32	RW	0000_0000h
224h	noise reduction 3 (swreg137)	32	RW	0000_0000h
228h	inter me SSE lambda config 4. For H2V5 or later version. (swreg138_H2V5)	32	RW	0000_0000h
22Ch	inter me SSE lambda config 5. For H2V5 or later version. (swreg139_H2V5)	32	RW	0000_0000h
230h	inter me SSE lambda config 6. For H2V5 or later version. (swreg140_H2V5)	32	RW	0000_0000h
234h	inter me SSE lambda config 7. For H2V5 or later version. (swreg141_H2V5)	32	RW	0000_0000h
238h	inter me SSE lambda config 8. For H2V5 or later version. (swreg142_H2V5)	32	RW	0000_0000h
23Ch	inter me SSE lambda config 9. For H2V5 or later version. (swreg143_H2V5)	32	RW	0000_0000h
240h	inter me SSE lambda config 10. For H2V5 or later version. (swreg144_H2V5)	32	RW	0000_0000h
244h	inter me SSE lambda config 11. For H2V5 or later version. (swreg145_H2V5)	32	RW	0000_0000h
248h	inter me SSE lambda config 12. For H2V5 or later version. (swreg146_H2V5)	32	RW	0000_0000h
24Ch	inter me SSE lambda config 13. For H2V5 or later version. (swreg147_H2V5)	32	RW	0000_0000h
250h	inter me SSE lambda config 14. For H2V5 or later version. (swreg148_H2V5)	32	RW	0000_0000h
254h	inter me SSE lambda config 15. For H2V5 or later version. (swreg149_H2V5)	32	RW	0000_0000h
258h	inter me SATD lambda config 8 (swreg150)	32	RW	0000_0000h
25Ch	inter me SATD lambda config 9 (swreg151)	32	RW	0000_0000h
260h	inter me SATD lambda config 10 (swreg152)	32	RW	0000_0000h
264h	inter me SATD lambda config 11 (swreg153)	32	RW	0000_0000h
268h	inter me SATD lambda config 12 (swreg154)	32	RW	0000_0000h
26Ch	inter me SATD lambda config 13 (swreg155)	32	RW	0000_0000h
270h	inter me SATD lambda config 14 (swreg156)	32	RW	0000_0000h
274h	inter me SATD lambda config 15 (swreg157)	32	RW	0000_0000h
278h	inter me SSE lambda config 16 (swreg158)	32	RW	0000_0000h
27Ch	inter me SSE lambda config 17 (swreg159)	32	RW	0000_0000h
280h	inter me SSE lambda config 18 (swreg160)	32	RW	0000_0000h
284h	inter me SSE lambda config 19 (swreg161)	32	RW	0000_0000h
288h	inter me SSE lambda config 20 (swreg162)	32	RW	0000_0000h
28Ch	inter me SSE lambda config 21 (swreg163)	32	RW	0000_0000h
290h	inter me SSE lambda config 22 (swreg164)	32	RW	0000_0000h
294h	inter me SSE lambda config 23 (swreg165)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
298h	inter me SSE lambda config 24 (swreg166)	32	RW	0000_0000h
29Ch	inter me SSE lambda config 25 (swreg167)	32	RW	0000_0000h
2A0h	inter me SSE lambda config 26 (swreg168)	32	RW	0000_0000h
2A4h	inter me SSE lambda config 27 (swreg169)	32	RW	0000_0000h
2B0h	inter me SSE lambda config 30 (swreg172)	32	RW	0000_0000h
2B4h	inter me SSE lambda config 31 (swreg173)	32	RW	0000_0660h
2B8h	intra SATD lambda config 8 (swreg174)	32	RW	0000_0000h
2BCh	intra SATD lambda config 9 (swreg175)	32	RW	0000_0000h
2C0h	intra SATD lambda config 10 (swreg176)	32	RW	0000_0000h
2C4h	intra SATD lambda config 11 (swreg177)	32	RW	0000_0000h
2C8h	intra SATD lambda config 12 (swreg178)	32	RW	0000_0000h
2CCh	intra SATD lambda config 13 (swreg179)	32	RW	0000_0000h
2D0h	intra SATD lambda config 14 (swreg180)	32	RW	0000_0000h
2D4h	intra SATD lambda config 15 (swreg181)	32	RW	0000_0000h
2D8h	qp fractional part (swreg182)	32	RW	0000_0000h
2DCh	qp sum (swreg183)	32	RO	0000_0000h
2E0h	qp num (swreg184)	32	RO	0000_0000h
2E4h	picture complexity. Timeout cycles MSB. (swreg185)	32	RW	0000_0000h
2E8h	Base address for CU information table LSB (swreg186)	32	RW	0000_0000h
2F0h	Base address for CU information LSB (swreg188)	32	RW	0000_0000h
2F8h	Long-term reference pictures config (swreg190)	32	RW	0000_0000h
2FCh	Temporal scalable config (swreg191)	32	RW	0000_0000h
30Ch	register extension for ctu_size=16 (swreg195)	32	RW	0000_0000h
310h	Low Latency Controls (swreg196)	32	RW	0000_0000h
314h	Delta POC extension (swreg197)	32	RW	0000_0000h
318h	Long Term Reference Control (swreg198)	32	RW	0000_0000h
31Ch	Hash Code Control (swreg199)	32	RW	0000_0000h
320h	Hash Code Value (swreg200)	32	RW	0000_0000h
324h	Background SKIP Control 0 (swreg201)	32	RW	0505_0505h
328h	Background SKIP Control 1 (swreg202)	32	RW	0002_0002h
32Ch	Background SKIP Control 2 (swreg203)	32	RW	0002_0002h
330h	Background SKIP Control 3 (swreg204)	32	RW	0002_0002h
334h	Background SKIP Control 4 (swreg205)	32	RW	000C_0003h
338h	Background SKIP Control 5 (swreg206)	32	RW	0033_000Ch
33Ch	Background SKIP Control 6 (swreg207)	32	RW	00CC_0033h
340h	Background SKIP Control 7 (swreg208)	32	RW	79E0_0000h
344h	IPCM Control 0 (swreg209)	32	RW	0000_0000h
348h	IPCM Control 1 (swreg210)	32	RW	0000_0000h
34Ch	IPCM Control 2 (swreg211)	32	RW	0000_0000h
350h	IPCM Control 3 (swreg212)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
354h	IPCM Control 4 (swreg213)	32	RW	0000_0000h
358h	HW synthesis config register 2, read-only (swreg214)	32	RO	E400_0000h
35Ch	AXI Information 0 (swreg215)	32	RO	0000_0000h
360h	AXI Information 1 (swreg216)	32	RO	0000_0000h
364h	AXI Information 2 (swreg217)	32	RO	0000_0000h
368h	AXI Information 3 (swreg218)	32	RO	0000_0000h
36Ch	AXI Information 4 (swreg219)	32	RO	0000_0000h
370h	AXI Information 5 (swreg220)	32	RO	0000_0000h
374h	AXI Information 6 (swreg221)	32	RO	0000_0000h
378h	AXI Information 7 (swreg222)	32	RO	0000_0000h
37Ch	AXI Information 8 (swreg223)	32	RO	0000_0000h
380h	control register 4 (swreg224)	32	RW	0002_0080h
384h	Tile Control (swreg225)	32	RW	0000_0000h
388h	HW synthesis config register 3, read-only (swreg226)	32	RO	0000_0000h
3ACh	RPS encoding control 0 (swreg235)	32	RW	0000_0000h
3B0h	RPS encoding control 1 (swreg236)	32	RW	0000_2000h
3B4h	Stride Control (swreg237)	32	RW	0000_0000h
3B8h	Dummy Read (swreg238)	32	RW	0000_0000h
3BCh	Base Address LSB of CTB MADs of current frame. (swreg239)	32	RW	0000_0000h
3C4h	Base Address LSB of CTB MADs of previous frame. (swreg241)	32	RW	0000_0000h
3CCh	CTB RC Control 0 (swreg243)	32	RW	0000_0000h
3D0h	CTB RC Control 1 (swreg244)	32	RW	0000_0000h
3D4h	CTB RC Control 2 (swreg245)	32	RW	0000_0000h
3D8h	CTB RC Control 3 (swreg246)	32	RW	0000_1010h
3DCh	CTB RC Control 4 (swreg247)	32	RW	0000_0000h
3E0h	CTB RC Control 5 (swreg248)	32	RW	0000_0000h
3E4h	register extension for 8K width (swreg249)	32	RW	0000_0000h
3E8h	Global MV Control 0 (swreg250)	32	RW	0000_0000h
3ECh	Global MV Control 1 (swreg251)	32	RW	0000_0000h
3F0h	ROI3 Area (swreg252)	32	RW	0000_0000h
3F4h	ROI3&4 Area (swreg253)	32	RW	0000_0000h
3F8h	ROI4&5 Area (swreg254)	32	RW	0000_0000h
3FCh	ROI5 Area (swreg255)	32	RW	0000_0000h
400h	ROI6 Area (swreg256)	32	RW	0000_0000h
404h	ROI6&7 Area (swreg257)	32	RW	0000_0000h
408h	ROI7&8 Area (swreg258)	32	RW	0000_0000h
40Ch	ROI8 Area (swreg259)	32	RW	0000_0000h
410h	ROI qp (swreg260)	32	RW	0000_0000h
414h	Stride Control (swreg261)	32	RW	0000_0404h
424h	Multicore sync ctrl (swreg265)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
428h	Multicore sync address L0 LSB (swreg266)	32	RW	0000_0000h
42Ch	Multicore sync address L0 MSB (swreg267)	32	RW	0000_0000h
430h	Multicore sync address L1 LSB (swreg268)	32	RW	0000_0000h
434h	Multicore sync address L1 MSB (swreg269)	32	RW	0000_0000h
438h	Multicore sync address recon LSB (swreg270)	32	RW	0000_0000h
43Ch	Multicore sync address recon MSB (swreg271)	32	RW	0000_0000h
440h	Programmable AXI urgent sideband signals (swreg272)	32	RW	FFFF_FFFFh
444h	roimap cu ctrl index address LSB (swreg273)	32	RW	0000_0000h
448h	roimap cu ctrl index address MSB (swreg274)	32	RW	0000_0000h
44Ch	roimap cu ctrl address LSB (swreg275)	32	RW	0000_0000h
450h	roimap cu ctrl address MSB (swreg276)	32	RW	0000_0000h
454h	poc type/bits setting (swreg277)	32	RW	0000_0000h
458h	stream output buffer1 address (swreg278)	32	RW	0000_0000h
460h	stream output buffer1 limit size (swreg280)	32	RW	0000_0000h
464h	poc type/bits setting (swreg281)	32	RW	0000_0000h
47Ch	HW synthesis config register 4, read-only (swreg287)	32	RO	0000_0000h
484h	Pre-processor color conversion parameters1 (swreg289)	32	RW	0000_0000h

16.3.5.2.2 Interrupt register encoder (swreg1)

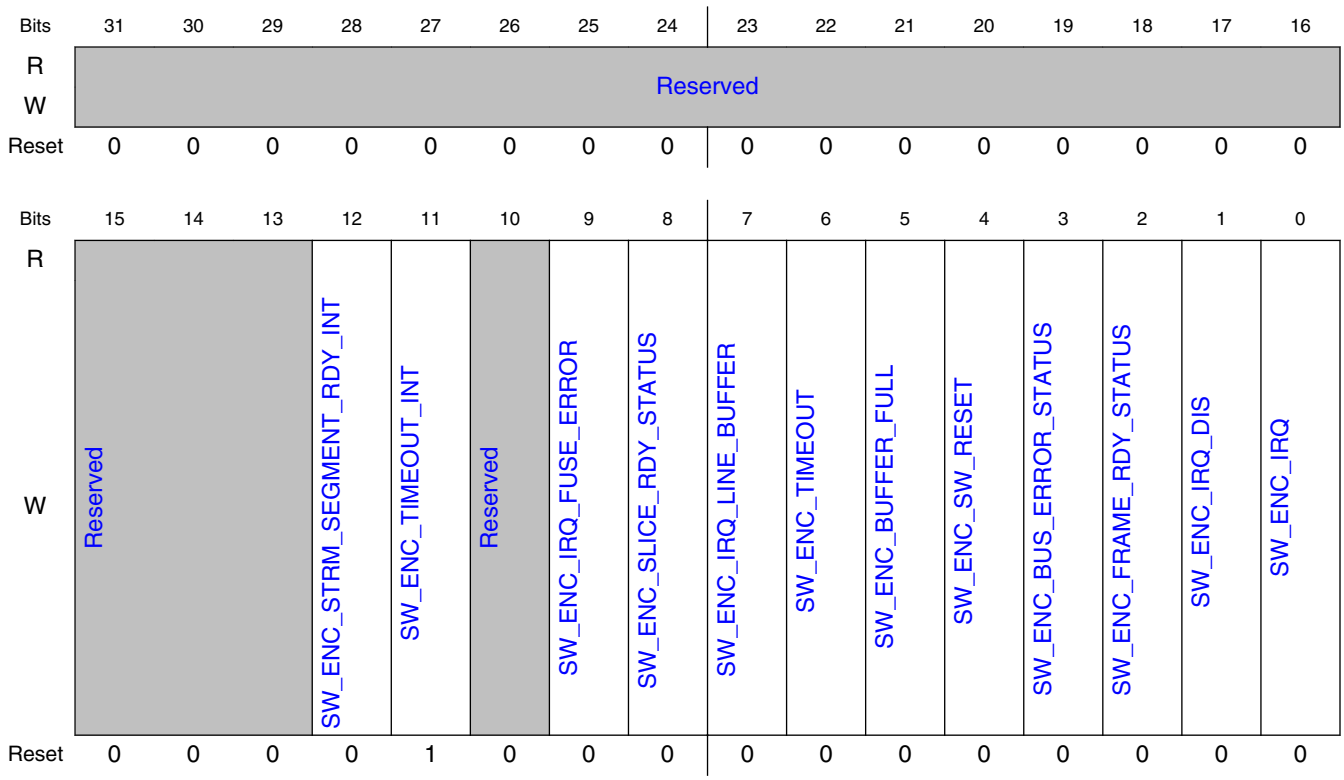
16.3.5.2.2.1 Offset

Register	Offset
swreg1	4h

16.3.5.2.2.2 Function

Interrupt register encoder

16.3.5.2.2.3 Diagram



16.3.5.2.2.4 Fields

Field	Function
31-13 —	Reserved
12 SW_ENC_STRM_SEGMENT_RDY_INT	Output stream segment IRQ. When high stream segment requests an interrupt. SW will reset this after interrupt is handled.
11 SW_ENC_TIMEOUT_INT	enable timeout interrupt when 1
10 —	Reserved
9 SW_ENC_IRQ_FUSE_ERROR	Interrupt Interrupt status bit encoder. When this bit is high encoder has a FUSE error. SW try to setup feature but HW miss the feature set
8 SW_ENC_SLICE_RDY_STATUS	Interrupt status bit encoder. When this bit is high encoder has encoded a slice.

Table continues on the next page...

Field	Function
7 SW_ENC_IRQ_LINE_BUFFER	Interrupt status bit encoder linebuffer empty. When setting. Encoder has finished encoding input buffers.
6 SW_ENC_TIMEOUT	Interrupt status bit encoder timeout. When high the encoder has been idling for too long. Possible only if timeout interrupt is enabled
5 SW_ENC_BUFFER_FULL	IRQ buffer full status bit. bufferFullInterrupt
4 SW_ENC_SW_RESET	IRQ SW reset status bit.
3 SW_ENC_BUS_ERROR_STATUS	Interrupt status bit bus. Error response from bus.
2 SW_ENC_FRAME_RDY_STATUS	Interrupt status bit encoder. When this bit is high encoder has encoded a picture.
1 SW_ENC_IRQ_DIS	Encoder IRQ disable. When high there are no interrupts concerning encoder from HW. Polling must be used to see the interrupt statuses.
0 SW_ENC_IRQ	Encoder IRQ. When high encoder requests an interrupt. SW will reset this after interrupt is handled.

16.3.5.2.3 Data configuration register0 (swreg2)

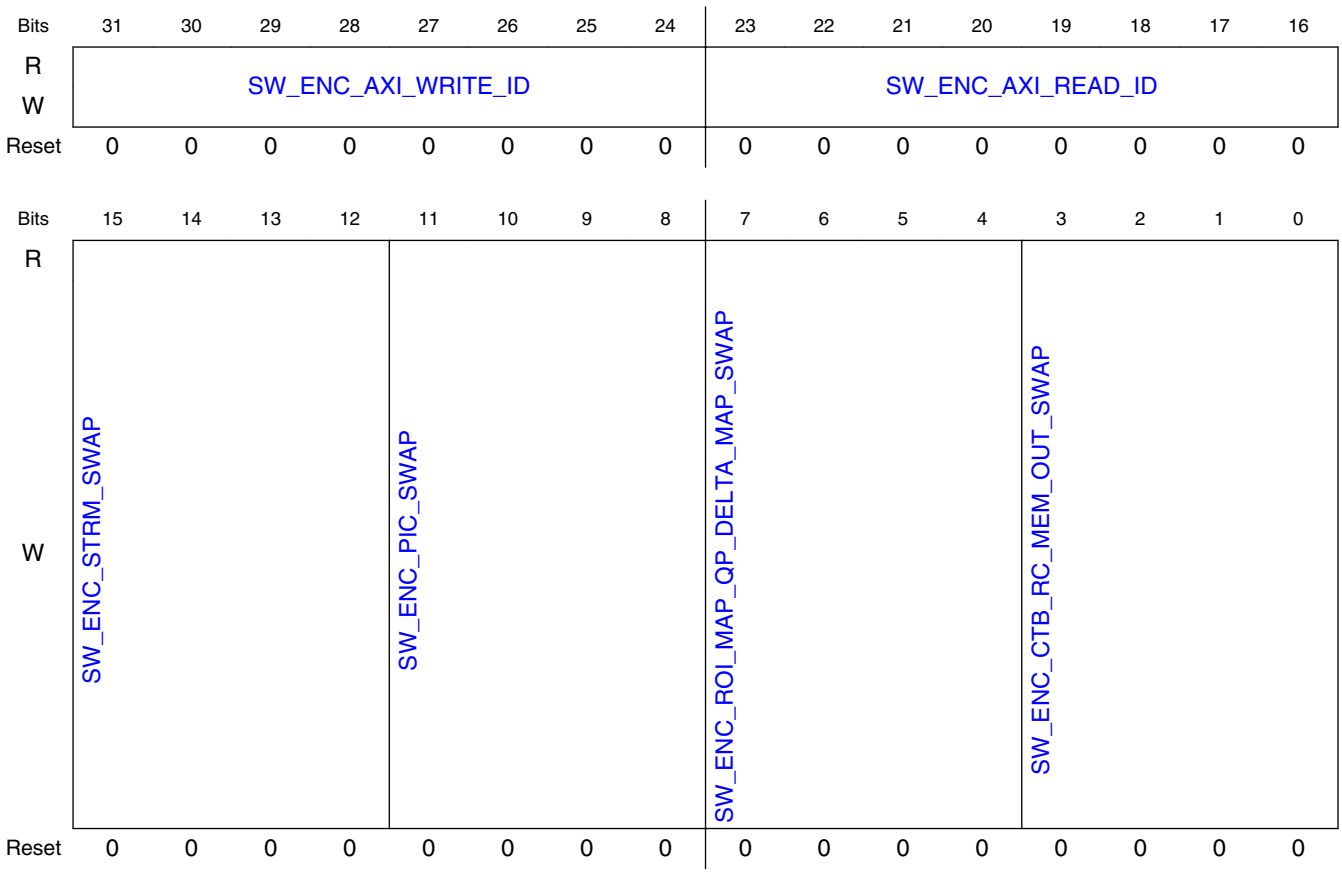
16.3.5.2.3.1 Offset

Register	Offset
swreg2	8h

16.3.5.2.3.2 Function

Data configuration register0

16.3.5.2.3.3 Diagram



16.3.5.2.3.4 Fields

Field	Function
31-24 SW_ENC_AXI_WRITE_ID	AXI Write ID
23-16 SW_ENC_AXI_READ_ID	AXI Read ID
15-12 SW_ENC_STRM_SWAP	byte swap config for output stream data.4 Bit byte order vector to control byte locations inside HW internal 128 bit data vector. For 64 and 32 bit external bus widths the data is first gathered to 128 bit width and then bytes swapped accordingly
11-8 SW_ENC_PIC_SWAP	Byte swap configuration for picture data (encoder input)
7-4	Byte swap configuration for qp delta of ROI map

Table continues on the next page...

Field	Function
SW_ENC_ROI_MAP_QP_DELTA_MAP_SWAP	
3-0 SW_ENC_CTB_RC_MEM_OUT_SWAP	Byte swap configuration for ctb rate control memory out

16.3.5.2.4 Data configuration register1 (swreg3)

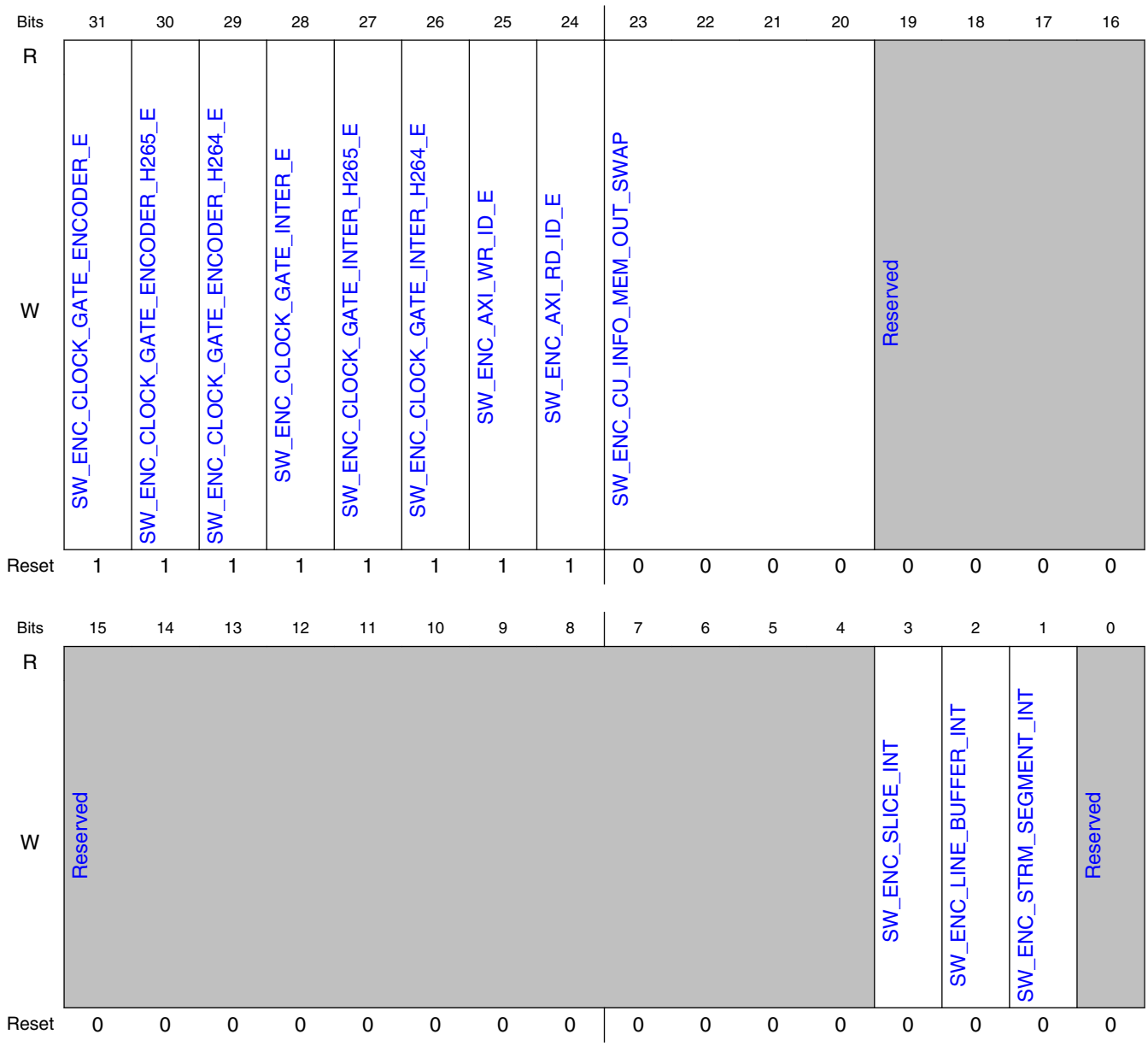
16.3.5.2.4.1 Offset

Register	Offset
swreg3	Ch

16.3.5.2.4.2 Function

Data configuration register1

16.3.5.2.4.3 Diagram



16.3.5.2.4.4 Fields

Field	Function
31 SW_ENC_CLOCK_GATE_ENCODER_E	ASIC encoder clock gating control. 0b - clock always on. 1b - hardware clock gating control
30	ASIC encoder clock gating control for h265. 0b - clock always on. 1b - hardware clock gating control

Table continues on the next page...

Field	Function
SW_ENC_CLOCK_GATE_ENCODER_H265_E	
29 SW_ENC_CLOCK_GATE_ENCODER_H264_E	ASIC encoder clock gating control for h264. 0b - clock always on. 1b - hardware clock gating control
28 SW_ENC_CLOCK_GATE_INTER_E	ASIC inter clock gating control. 0b - clock always on. 1b - hardware clock gating control
27 SW_ENC_CLOCK_GATE_INTER_H265_E	ASIC inter clock gating control for h265. 0b - clock always on. 1b - hardware clock gating control
26 SW_ENC_CLOCK_GATE_INTER_H264_E	ASIC inter clock gating control for h264. 0b - clock always on. 1b - hardware clock gating control
25 SW_ENC_AXI_WR_ID_E	axi write enable. 0b - disable. 1b - enable.
24 SW_ENC_AXI_RD_ID_E	axi read enable. 0b - disable. 1b - enable.
23-20 SW_ENC_CU_INFO_MEM_OUT_SWAP	Byte swap configuration for cu infomation memory out
19-4 —	Reserved
3 SW_ENC_SLICE_INT	enable slice interrupt when 1
2 SW_ENC_LINE_BUFFER_INT	enable line buffer interrupt when 1
1 SW_ENC_STRM_SEGMENT_INT	enable stream segment interrupt when 1
0 —	Reserved

16.3.5.2.5 control register 0 (swreg4)

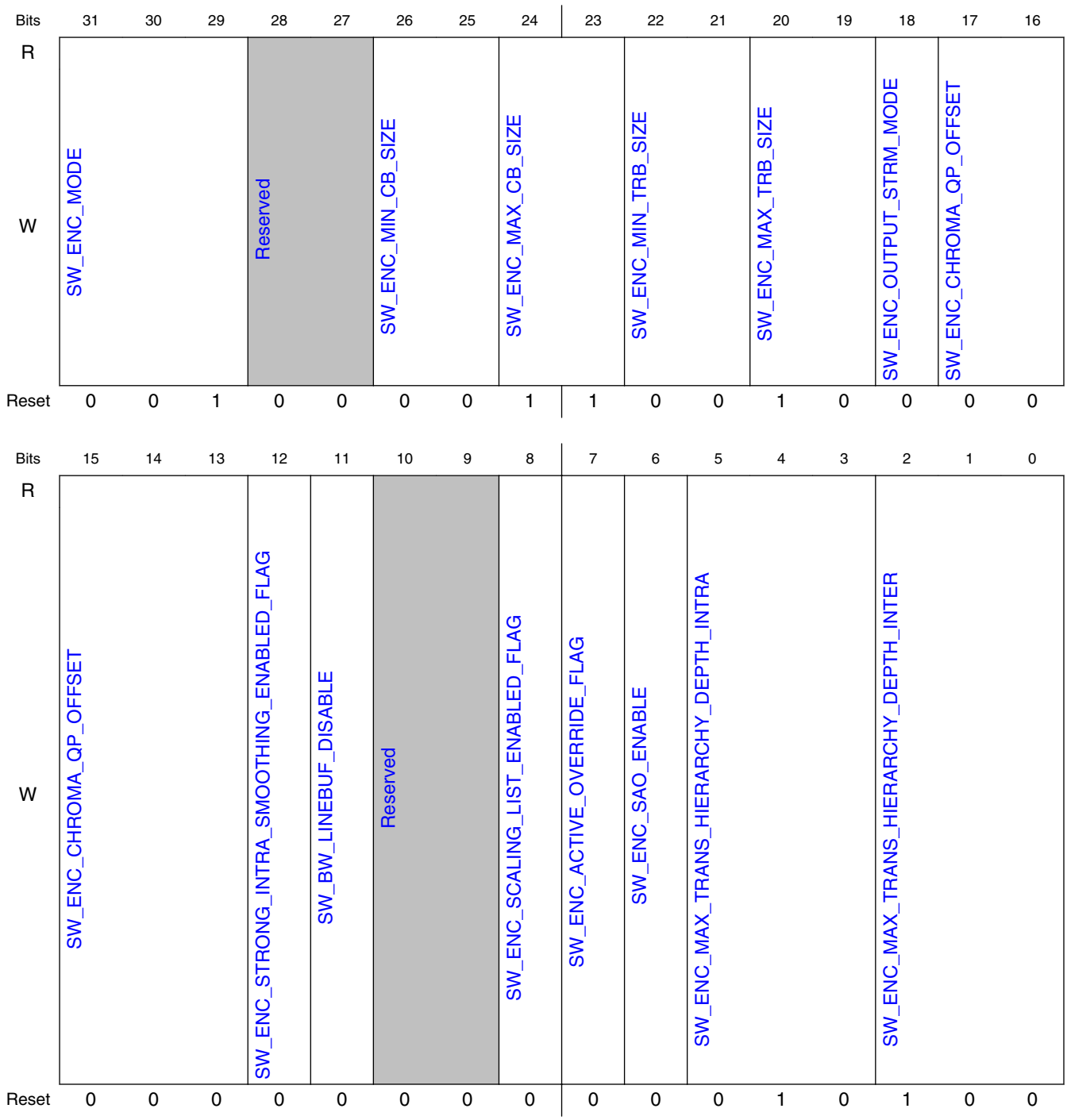
16.3.5.2.5.1 Offset

Register	Offset
swreg4	10h

16.3.5.2.5.2 Function

control register 0

16.3.5.2.5.3 Diagram



16.3.5.2.5.4 Fields

Field	Function
31-29	Encoding mode. streamType. 001b - hevc.

Table continues on the next page...

Field	Function
SW_ENC_MODE	010b - h264.
28-27 —	Reserved
26-25 SW_ENC_MIN_CB_SIZE	min cb size (we only support 8x8): 00b - 8x8 01b - 16x16 10b - 32x32 11b - 64x64
24-23 SW_ENC_MAX_CB_SIZE	max cb size (we only support 64x64): 00b - 8x8 01b - 16x16 10b - 32x32 11b - 64x64
22-21 SW_ENC_MIN_TRB_SIZE	min tr block size (we only support 4x4) 00b - 4x4 01b - 8x8 10b - 16x16 11b - 32x32
20-19 SW_ENC_MAX_TRB_SIZE	max tr block size (we only support 16x16) 00b - 4x4 01b - 8x8 10b - 16x16 11b - 32x32
18 SW_ENC_OUTPUT_STRM_MODE	output stream mode: 0b - byte stream 1b - Nal stream
17-13 SW_ENC_CHROMA_QP_OFFSET	chroma qp offset[-12~12]
12 SW_ENC_STRONG_INTRA_SMOOTHING_ENABLED_FLAG	HEVC IntraTU32x32 strong intra smoothing filter enable flag
11 SW_BW_LINEBUF_DISABLE	BW line buffer disable
10-9 —	Reserved
8 SW_ENC_SCALING_LIST_ENABLED_FLAG	scaling_list_enabled_flag
7	active override flag

Table continues on the next page...

Field	Function
SW_ENC_ACTIVE_OVERRIDE_FLAG	
6 SW_ENC_SAO_ENABLE	SAO enable
5-3 SW_ENC_MAX_TRANS_HIERARCHY_DEPTH_INTRA	max transform hierarchy depth of intra
2-0 SW_ENC_MAX_TRANS_HIERARCHY_DEPTH_INTER	max transform hierarchy depth of inter

16.3.5.2.6 control register 1 (swreg5)

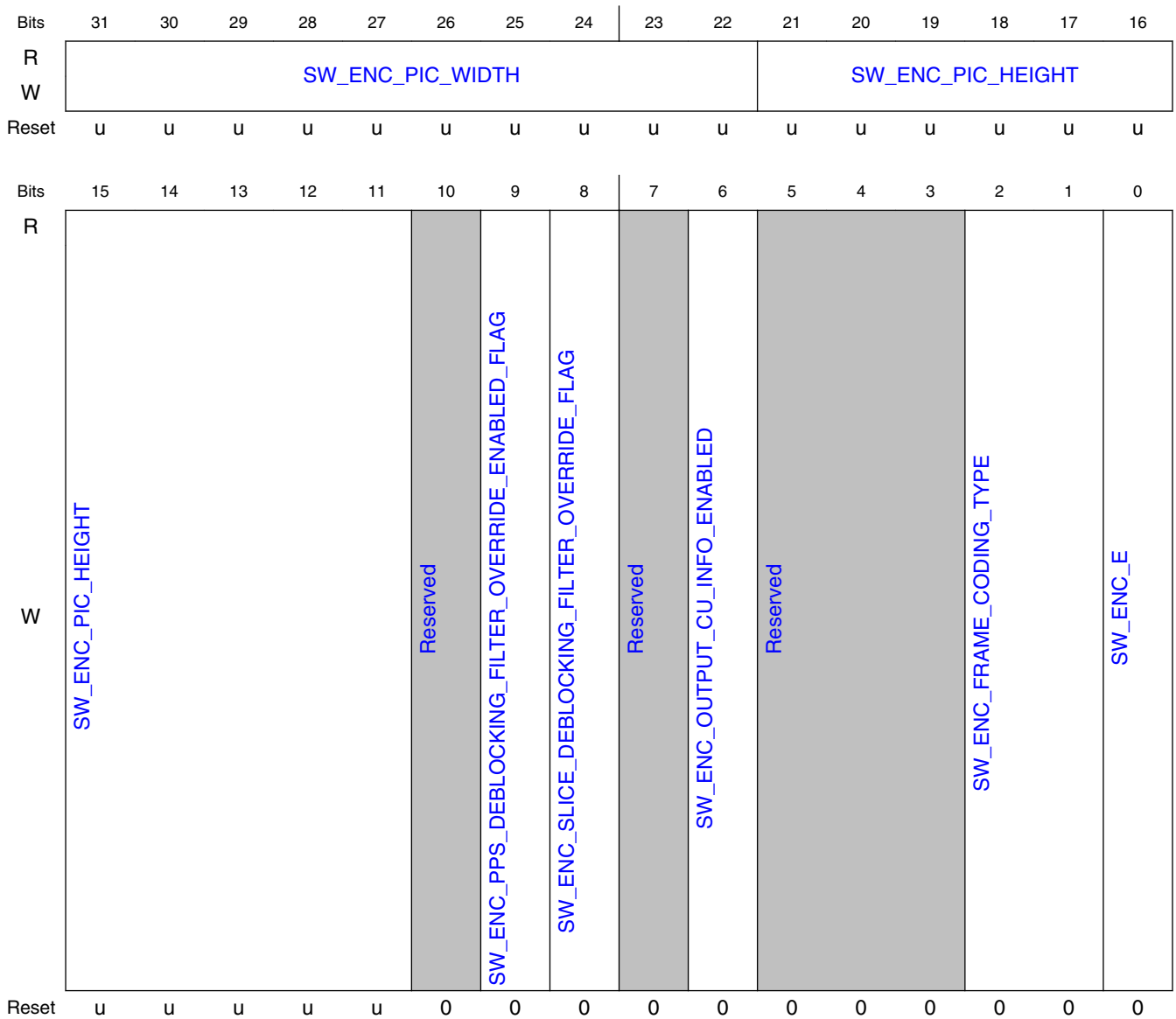
16.3.5.2.6.1 Offset

Register	Offset
swreg5	14h

16.3.5.2.6.2 Function

control register 1

16.3.5.2.6.3 Diagram



16.3.5.2.6.4 Fields

Field	Function
31-22 SW_ENC_PIC_WIDTH	Encoded width. lumWidthLsb(unit 8 pixels). bit[9:0]
21-11 SW_ENC_PIC_HEIGHT	Encoded height. lumHeight(unit 8 pixels):Max width x height is 4096 x 4096
10	Reserved

Table continues on the next page...

Field	Function
—	
9 SW_ENC_PPS_DEBLOCKING_FILTER_OVERRIDE_ENABLED_FLAG	deblocking filter override enable flag. 0b - disable 1b - enable
8 SW_ENC_SLICE_DEBLOCKING_FILTER_OVERRIDE_FLAG	slice deblocking filter override flag. 0b - no 1b - yes
7 —	Reserved
6 SW_ENC_OUTPUT_CU_INFO_ENABLED	Enable dumping cu information to external memory. 0b - disable 1b - enable
5-3 —	Reserved
2-1 SW_ENC_FRAME_CODING_TYPE	Encoded picture type. frameType.:1-I:0-P:2:B
0 SW_ENC_E	encoder enable. Setting this bit high will start the encoding operation. HW will reset this when picture is processed or bus error or timeout interrupt is given.

16.3.5.2.7 control register 3 (swreg7)

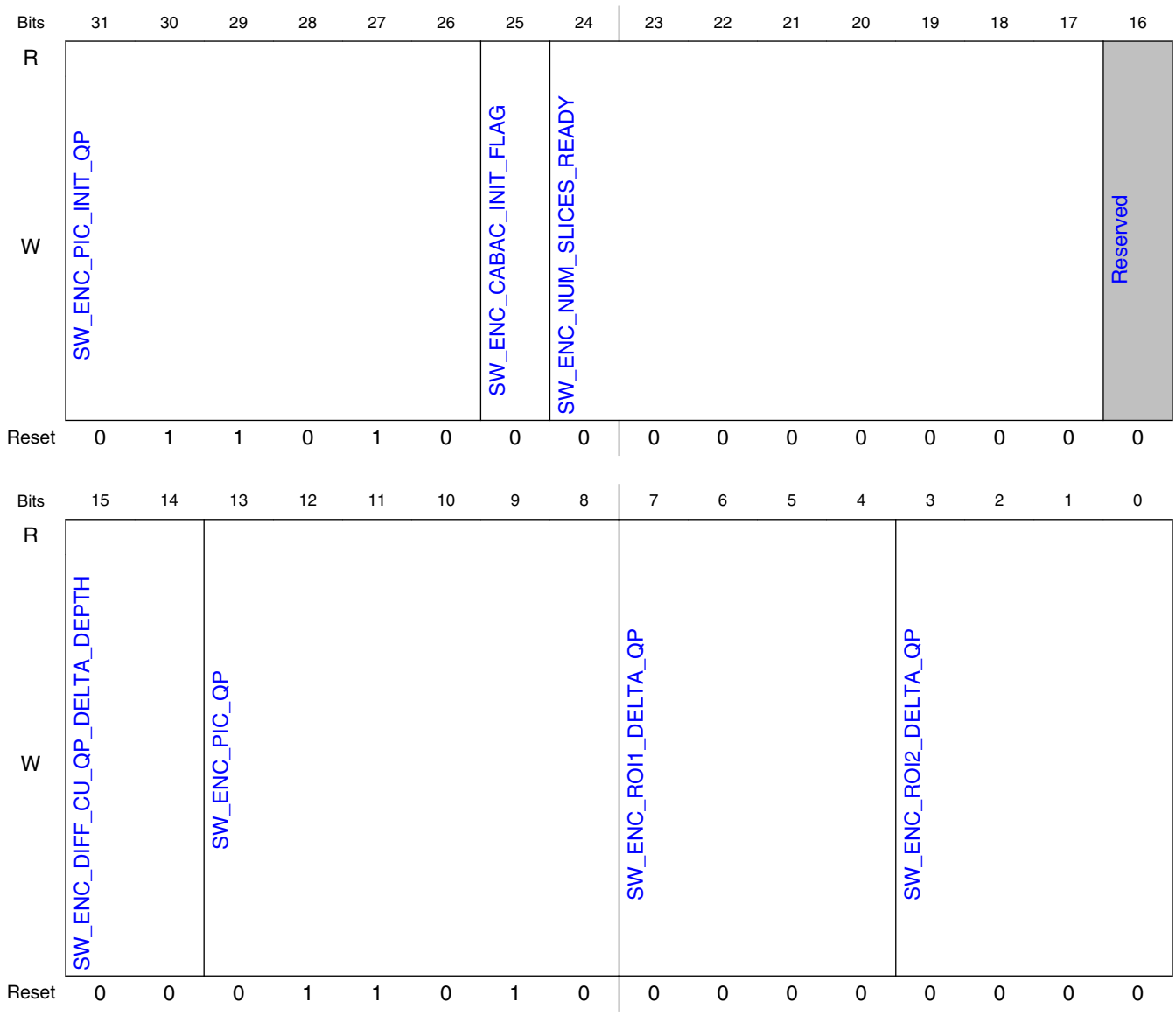
16.3.5.2.7.1 Offset

Register	Offset
swreg7	1Ch

16.3.5.2.7.2 Function

control register 3

16.3.5.2.7.3 Diagram



16.3.5.2.7.4 Fields

Field	Function
31-26 SW_ENC_PIC_INIT_QP	picture header qp
25 SW_ENC_CABAC_INIT_FLAG	cabac init flag
24-17	HEVC amount of completed slices.

Table continues on the next page...

Field	Function
SW_ENC_NUM_SLICES_READY	
16 —	Reserved
15-14 SW_ENC_DIFF_CU_QP_DELTA_DEPTH	difference of cu qp delta depth
13-8 SW_ENC_PIC_QP	qp of current picture
7-4 SW_ENC_ROI1_DELTA_QP	ROI1 delta qp
3-0 SW_ENC_ROI2_DELTA_QP	ROI2 delta qp

16.3.5.2.8 stream output buffer0 address (swreg8)

16.3.5.2.8.1 Offset

Register	Offset
swreg8	20h

16.3.5.2.8.2 Function

stream output buffer0 address

16.3.5.2.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_OUTPUT_STRM_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_OUTPUT_STRM_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.8.4 Fields

Field	Function
31-0 SW_ENC_OUTPUT_STRM_BASE	stream output buffer0 base address

16.3.5.2.9 stream output buffer0 limit size (swreg9)

16.3.5.2.9.1 Offset

Register	Offset
swreg9	24h

16.3.5.2.9.2 Function

stream output buffer0 limit size

16.3.5.2.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_OUTPUT_STRM_BUFFER_LIMIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_OUTPUT_STRM_BUFFER_LIMIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.9.4 Fields

Field	Function
31-0 SW_ENC_OUTPUT_STRM_BUFFER_LIMIT	Stream buffer0 limit / Output stream size (bytes). If buffer0 limit is reached and sw_enc_HWStreamBufChain=0 buffer_full_IRQ will be generated. Output stream size will be updated here when a slice or frame is finished

16.3.5.2.10 sizeTblBase (swreg10)

16.3.5.2.10.1 Offset

Register	Offset
swreg10	28h

16.3.5.2.10.2 Function

sizeTblBase

16.3.5.2.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_SIZE_TBL_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_SIZE_TBL_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.10.4 Fields

Field	Function
31-0 SW_ENC_SIZE_TBL_BASE	sizeTblBase

16.3.5.2.11 encoded Picture order count (swreg11)

16.3.5.2.11.1 Offset

Register	Offset
swreg11	2Ch

16.3.5.2.11.2 Function

encoded Picture order count

16.3.5.2.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_POC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_POC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.11.4 Fields

Field	Function
31-0 SW_ENC_POC	encoded Picture order count

16.3.5.2.12 input lum base address (swreg12)

16.3.5.2.12.1 Offset

Register	Offset
swreg12	30h

16.3.5.2.12.2 Function

input lum base address

16.3.5.2.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_INPUT_Y_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_INPUT_Y_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.12.4 Fields

Field	Function
31-0 SW_ENC_INPUT_Y_BASE	input image lum base address

16.3.5.2.13 input cb base address (swreg13)

16.3.5.2.13.1 Offset

Register	Offset
swreg13	34h

16.3.5.2.13.2 Function

input cb base address

16.3.5.2.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_INPUT_CB_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_INPUT_CB_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.13.4 Fields

Field	Function
31-0 SW_ENC_INPUT_CB_BASE	input image cb base address

16.3.5.2.14 input cr base address (swreg14)

16.3.5.2.14.1 Offset

Register	Offset
swreg14	38h

16.3.5.2.14.2 Function

input cr base address

16.3.5.2.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_INPUT_CR_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_INPUT_CR_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.14.4 Fields

Field	Function
31-0 SW_ENC_INPUT_CR_BASE	input image cr base address

16.3.5.2.15 recon image luma base address (swreg15)

16.3.5.2.15.1 Offset

Register	Offset
swreg15	3Ch

16.3.5.2.15.2 Function

recon image luma base address

16.3.5.2.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RECON_Y_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RECON_Y_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.15.4 Fields

Field	Function
31-0 SW_ENC_RECON_Y_BASE	recon image lum base address

16.3.5.2.16 recon image chroma base address (swreg16)

16.3.5.2.16.1 Offset

Register	Offset
swreg16	40h

16.3.5.2.16.2 Function

recon image chroma base address

16.3.5.2.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RECON_CHROMA_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RECON_CHROMA_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.16.4 Fields

Field	Function
31-0 SW_ENC_RECON_CHROMA_BASE	recon image chroma base address

16.3.5.2.17 reference picture reconstructed list0 luma0 (swreg18)

16.3.5.2.17.1 Offset

Register	Offset
swreg18	48h

16.3.5.2.17.2 Function

reference picture reconstructed list0 luma0

16.3.5.2.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REFPIC_RECON_L0_Y0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_REFPIC_RECON_L0_Y0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.17.4 Fields

Field	Function
31-0 SW_ENC_REFPIC_RECON_L0_Y0	reference picture reconstructed list0 luma0

16.3.5.2.18 reference picture reconstructed list0 chroma0 (swreg19)

16.3.5.2.18.1 Offset

Register	Offset
swreg19	4Ch

16.3.5.2.18.2 Function

reference picture reconstructed list0 chroma0

16.3.5.2.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REFPIC_RECON_L0_CHROMA0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_REFPIC_RECON_L0_CHROMA0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.18.4 Fields

Field	Function
31-0 SW_ENC_REFPIC_RECON_L0_CHROMA0	reference picture reconstructed list0 chroma0

16.3.5.2.19 Cyclic Intra (swreg22)

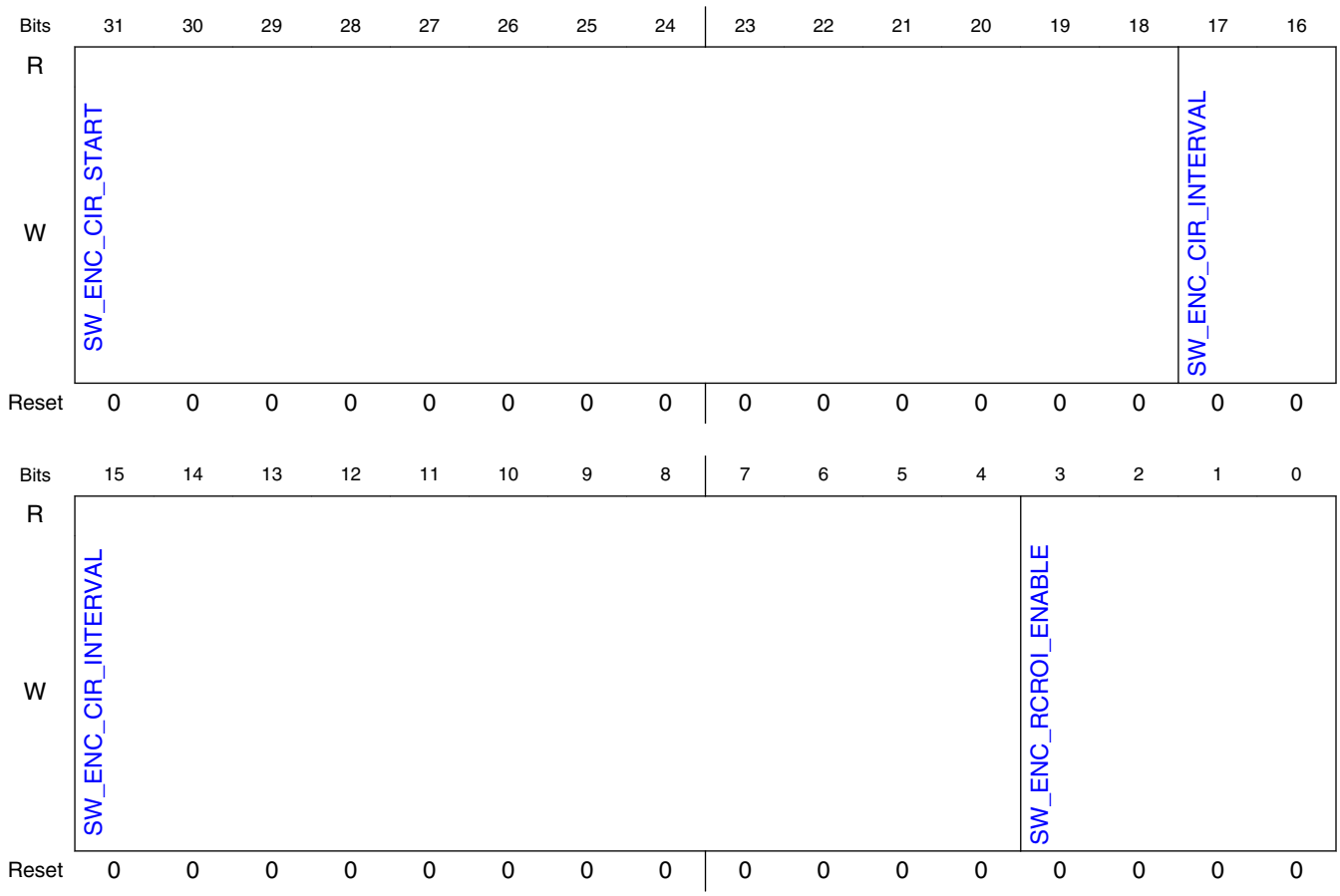
16.3.5.2.19.1 Offset

Register	Offset
swreg22	58h

16.3.5.2.19.2 Function

Cyclic Intra

16.3.5.2.19.3 Diagram



16.3.5.2.19.4 Fields

Field	Function
31-18 SW_ENC_CIR_START	cir start
17-4 SW_ENC_CIR_INTERVAL	Cyclic Intra Refresh
3-0 SW_ENC_RCROI_ENABLE	bit3:RC enable; bit2:Quatily adjustment enable; bit1:roi map enable; bit0:roi area enable.

16.3.5.2.20 intra Area (swreg23)

16.3.5.2.20.1 Offset

Register	Offset
swreg23	5Ch

16.3.5.2.20.2 Function

intra Area

16.3.5.2.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_INTRA_AREA_LEFT								SW_ENC_INTRA_AREA_RIGHT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_INTRA_AREA_TOP								SW_ENC_INTRA_AREA_BOTTOM							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.20.4 Fields

Field	Function
31-24 SW_ENC_INTRA_AREA_LEFT	intra Area Left
23-16 SW_ENC_INTRA_AREA_RIGHT	intra Area Right
15-8 SW_ENC_INTRA_AREA_TOP	intra Area Top
7-0 SW_ENC_INTRA_AREA_BOTTOM	intra Area Bottom

16.3.5.2.21 ROI1 Area (swreg24)

16.3.5.2.21.1 Offset

Register	Offset
swreg24	60h

16.3.5.2.21.2 Function

ROI1 Area

16.3.5.2.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROI1_LEFT								SW_ENC_ROI1_RIGHT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROI1_TOP								SW_ENC_ROI1_BOTTOM							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.21.4 Fields

Field	Function
31-24 SW_ENC_ROI1_LEFT	ROI1 Area Left
23-16 SW_ENC_ROI1_RIGHT	ROI1 Area Right
15-8 SW_ENC_ROI1_TOP	ROI1 Area Top
7-0 SW_ENC_ROI1_BOTTOM	ROI1 Area Bottom

16.3.5.2.22 ROI2 Area (swreg25)

16.3.5.2.22.1 Offset

Register	Offset
swreg25	64h

16.3.5.2.22.2 Function

ROI2 Area

16.3.5.2.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROI2_LEFT								SW_ENC_ROI2_RIGHT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROI2_TOP								SW_ENC_ROI2_BOTTOM							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.22.4 Fields

Field	Function
31-24 SW_ENC_ROI2_LEFT	ROI2 Area Left
23-16 SW_ENC_ROI2_RIGHT	ROI2 Area Right
15-8 SW_ENC_ROI2_TOP	ROI2 Area Top
7-0 SW_ENC_ROI2_BOTTOM	ROI2 Area Bottom

16.3.5.2.23 intra penalty for picture0. For H2V1 only. (swreg26_H2V1)

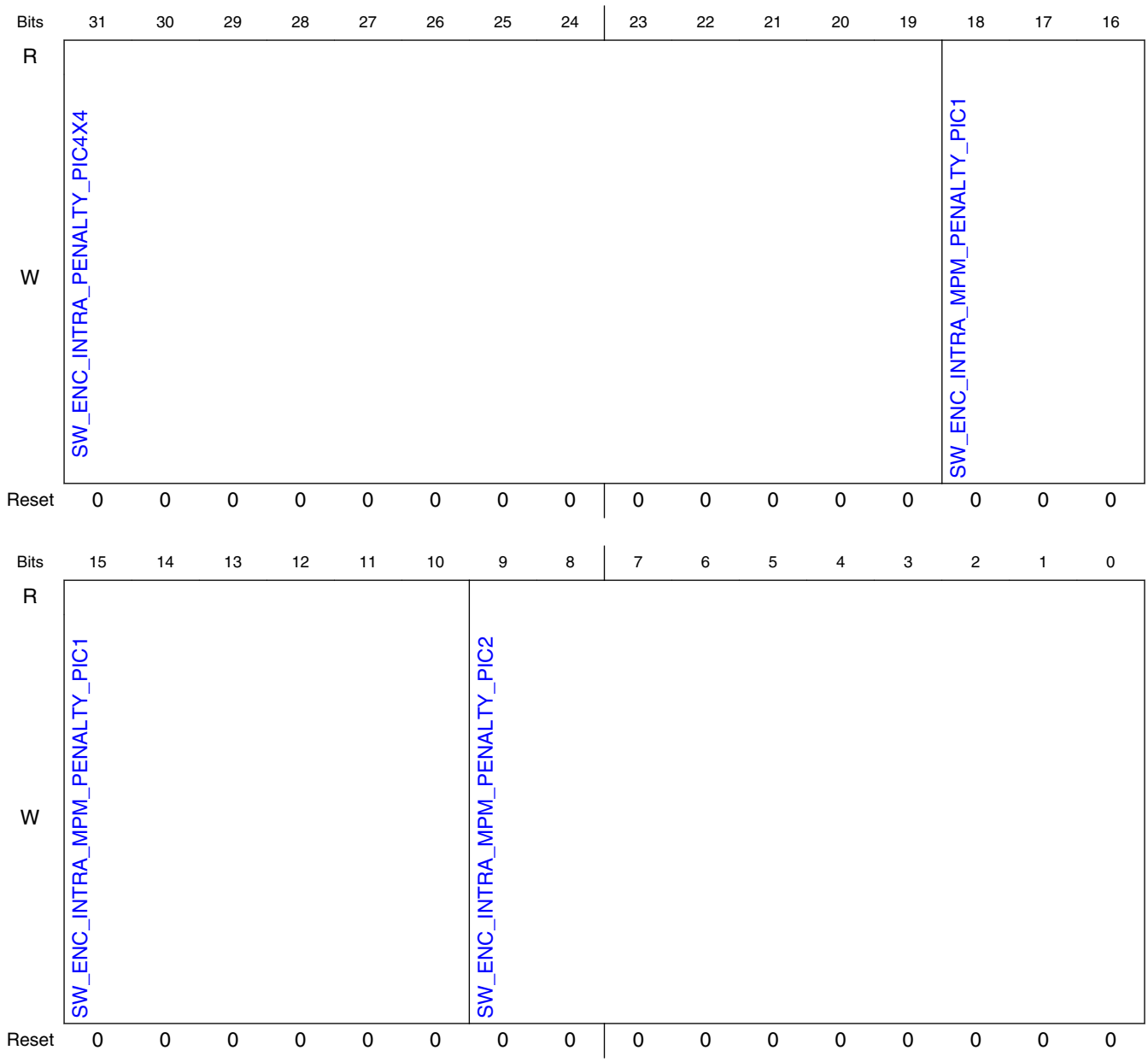
16.3.5.2.23.1 Offset

Register	Offset
swreg26_H2V1	68h

16.3.5.2.23.2 Function

intra penalty for picture0. For H2V1 only.

16.3.5.2.23.3 Diagram



16.3.5.2.23.4 Fields

Field	Function
31-19 SW_ENC_INTRA_PENALTY_PIC4X4	intra penalty for pic4x4
18-10	intra MPM penalty for pic1

Table continues on the next page...

Field	Function
SW_ENC_INTRA_MPM_PENALTY_PIC1	
9-0 SW_ENC_INTRA_MPM_PENALTY_PIC2	intra MPM penalty for pic2

16.3.5.2.24 intra size factors. For H2V2 or later version. (swreg26_H2V2)

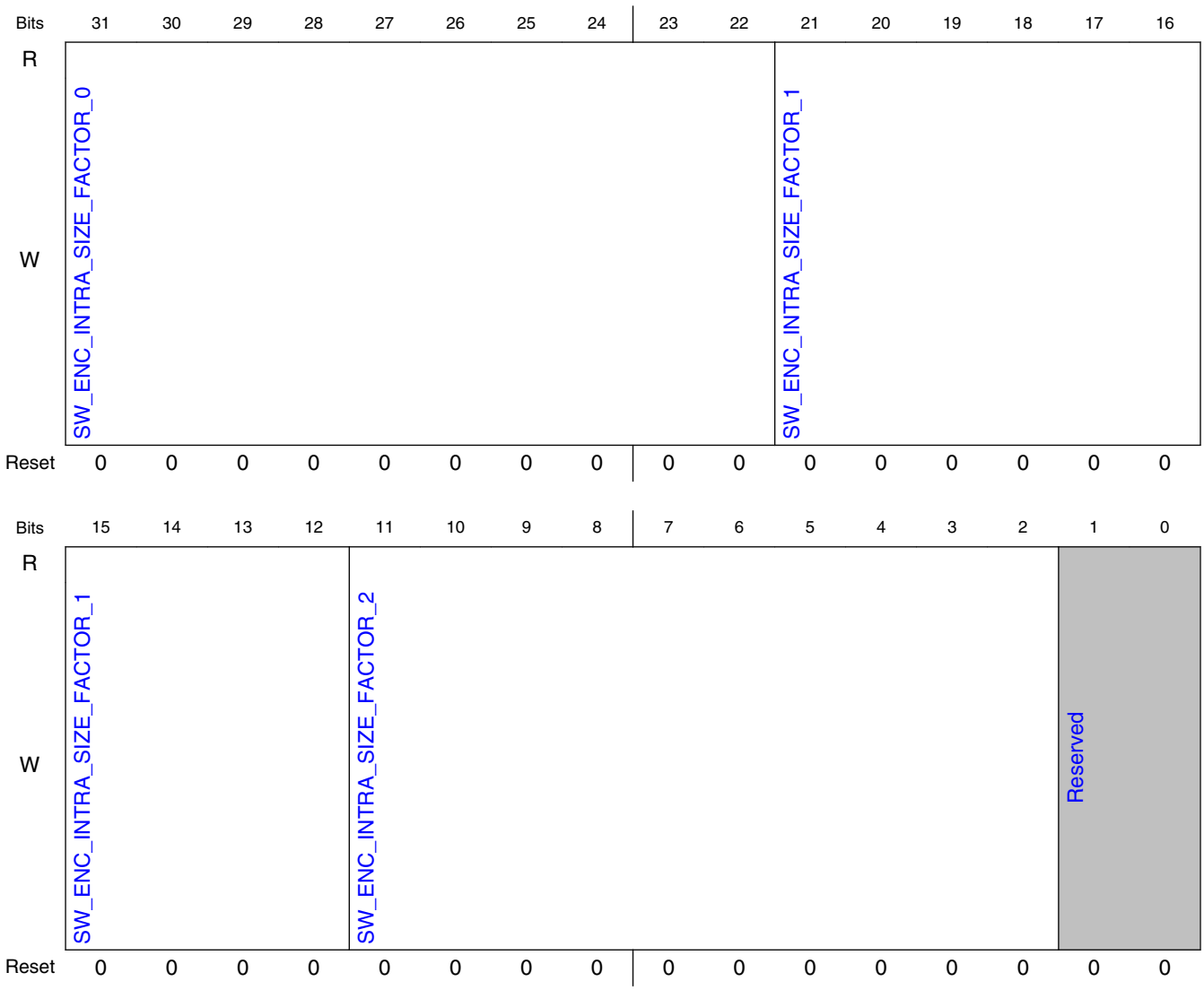
16.3.5.2.24.1 Offset

Register	Offset
swreg26_H2V2	68h

16.3.5.2.24.2 Function

intra size factors. For H2V2 or later version.

16.3.5.2.24.3 Diagram



16.3.5.2.24.4 Fields

Field	Function
31-22 SW_ENC_INTRA_SIZE_FACTOR_0	intra size factor 0
21-12 SW_ENC_INTRA_SIZE_FACTOR_1	intra size factor 1
11-2	intra size factor 2

Table continues on the next page...

Field	Function
SW_ENC_INTRA_SIZE_FACTOR_2	
1-0 —	Reserved

16.3.5.2.25 intra penalty for picture1. For H2V1 only. (swreg27_H2V1)

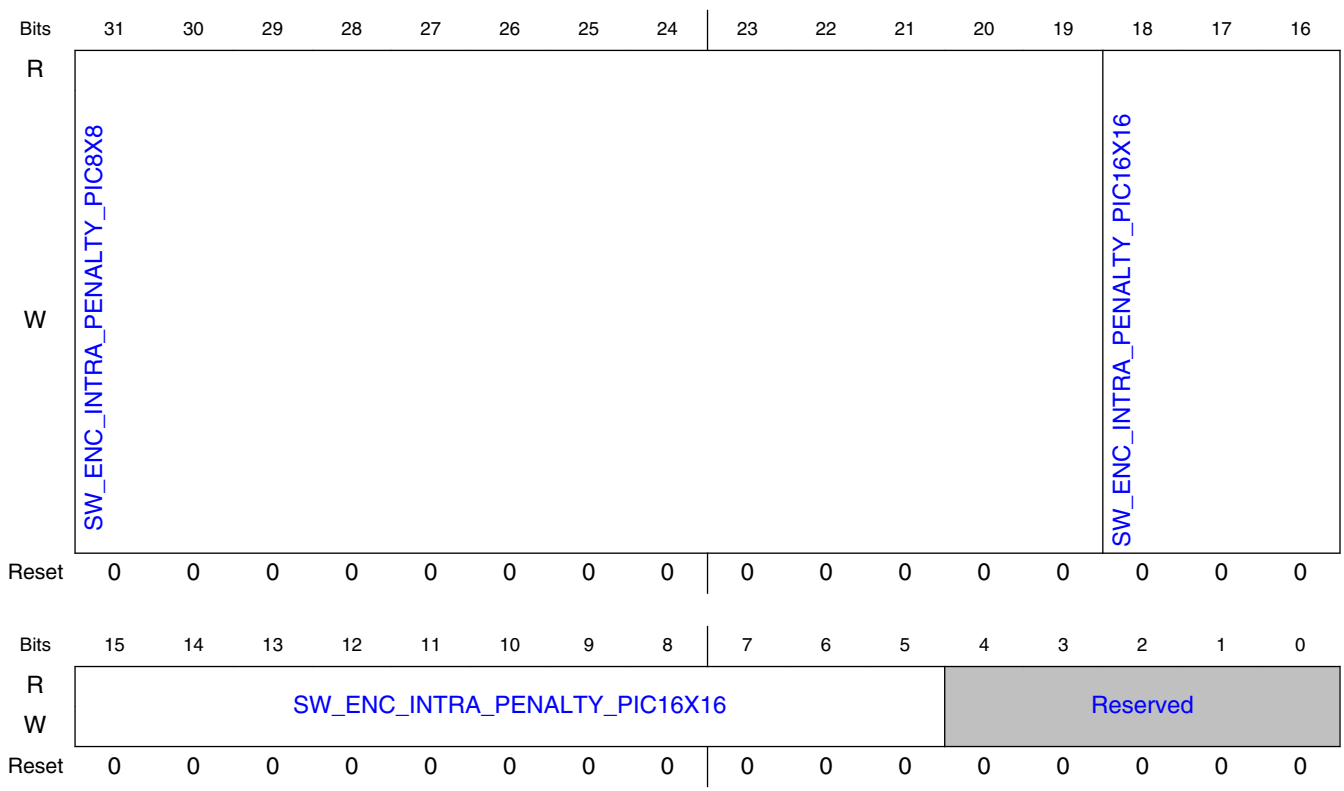
16.3.5.2.25.1 Offset

Register	Offset
swreg27_H2V1	6Ch

16.3.5.2.25.2 Function

intra penalty for picture1. For H2V1 only.

16.3.5.2.25.3 Diagram



16.3.5.2.25.4 Fields

Field	Function
31-19 SW_ENC_INTRA_PENALTY_PIC8X8	intra penalty for pic8x8
18-5 SW_ENC_INTRA_PENALTY_PIC16X16	intra penalty for pic16x16
4-0 —	Reserved

16.3.5.2.26 intra mode factors . For H2V2 or later version. (swreg27_H2V2)

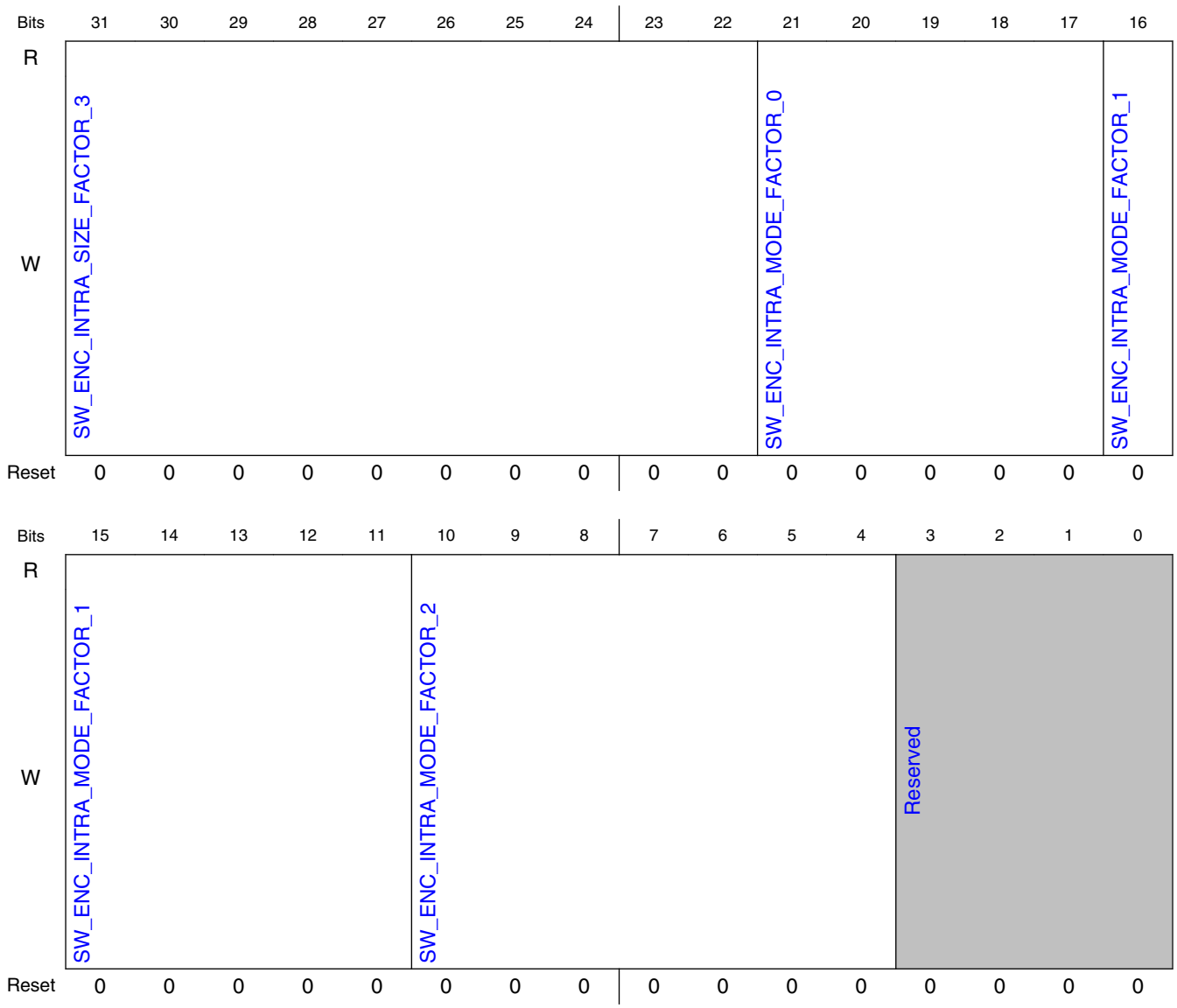
16.3.5.2.26.1 Offset

Register	Offset
swreg27_H2V2	6Ch

16.3.5.2.26.2 Function

intra mode factors . For H2V2 or later version.

16.3.5.2.26.3 Diagram



16.3.5.2.26.4 Fields

Field	Function
31-22 SW_ENC_INTRA_SIZE_FACTOR_3	intra size factor 3
21-17 SW_ENC_INTRA_MODE_FACTOR_0	intra mode factor 0

Table continues on the next page...

Field	Function
16-11 SW_ENC_INTRA_MODE_FACTOR_1	intra mode factor 1
10-4 SW_ENC_INTRA_MODE_FACTOR_2	intra mode factor 2
3-0 —	Reserved

16.3.5.2.27 intra penalty for picture2 (swreg28)

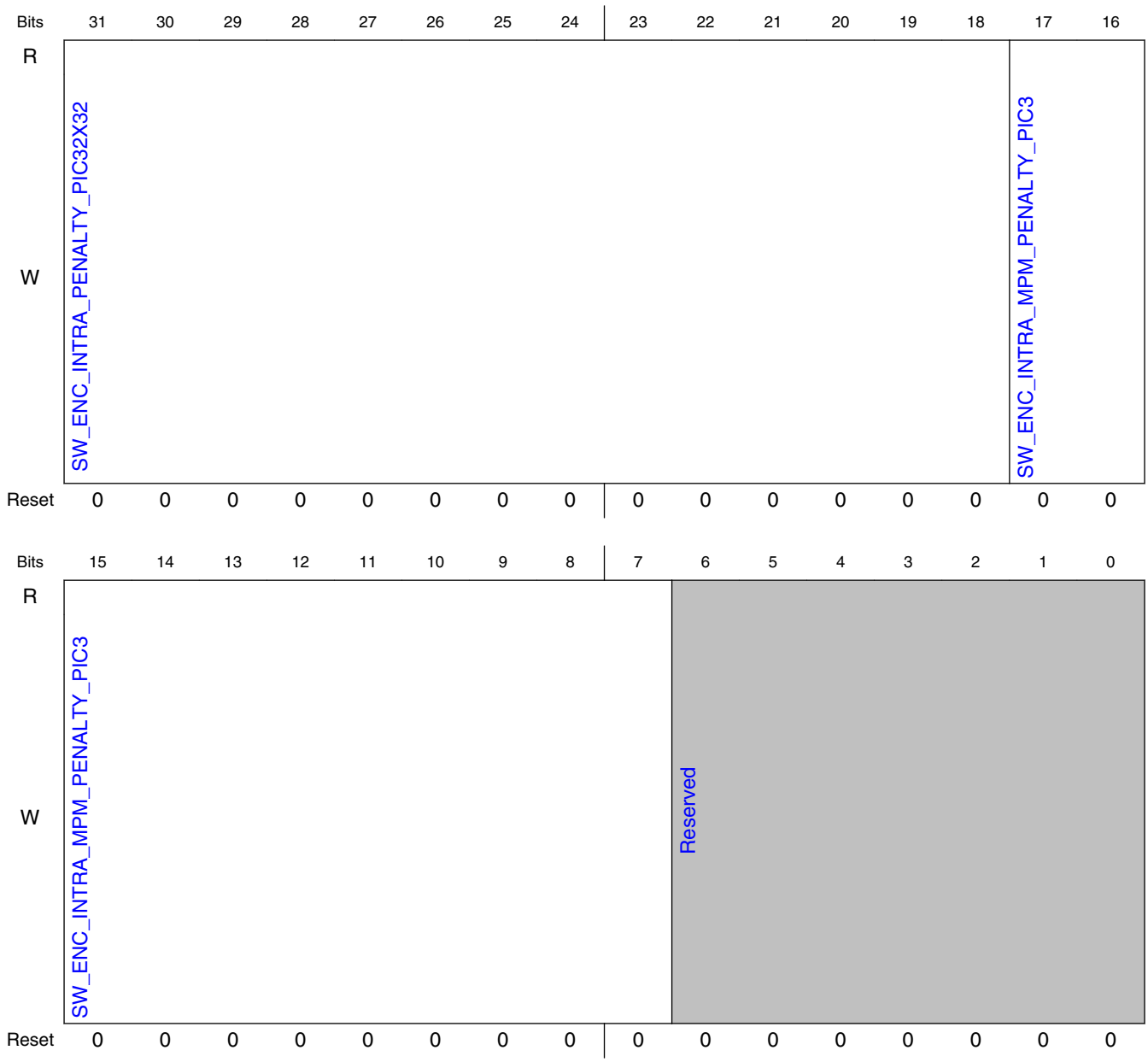
16.3.5.2.27.1 Offset

Register	Offset
swreg28	70h

16.3.5.2.27.2 Function

intra penalty for picture2

16.3.5.2.27.3 Diagram



16.3.5.2.27.4 Fields

Field	Function
31-18 SW_ENC_INTRA_PENALTY_PIC32X32	intra penalty for pic32x32
17-7	intra MPM penalty for pic3

Table continues on the next page...

Field	Function
SW_ENC_INTRA_MPM_PENALTY_PIC3	
6-0 —	Reserved

16.3.5.2.28 inter me SATD lambda config 0. For H2V2 or later version. (swreg28_H2V2)

16.3.5.2.28.1 Offset

Register	Offset
swreg28_H2V2	70h

16.3.5.2.28.2 Function

inter me SATD lambda config 0. For H2V2 or later version.

16.3.5.2.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_LAMDA_SATD_ME_0								SW_ENC_LAMDA_SATD_ME_1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_LAMDA_SATD_ME_2								SW_ENC_LAMDA_SATD_ME_3							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.28.4 Fields

Field	Function
31-24 SW_ENC_LAMDA_SATD_ME_0	lambda satd me 0
23-16	lambda satd me 1

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_1	
15-8 SW_ENC_LAMDA_SATD_ME_2	lambda satd me 2
7-0 SW_ENC_LAMDA_SATD_ME_3	lambda satd me 3

16.3.5.2.29 inter me SATD lambda config 0. For H2V5 or later version. (swreg28_H2V5)

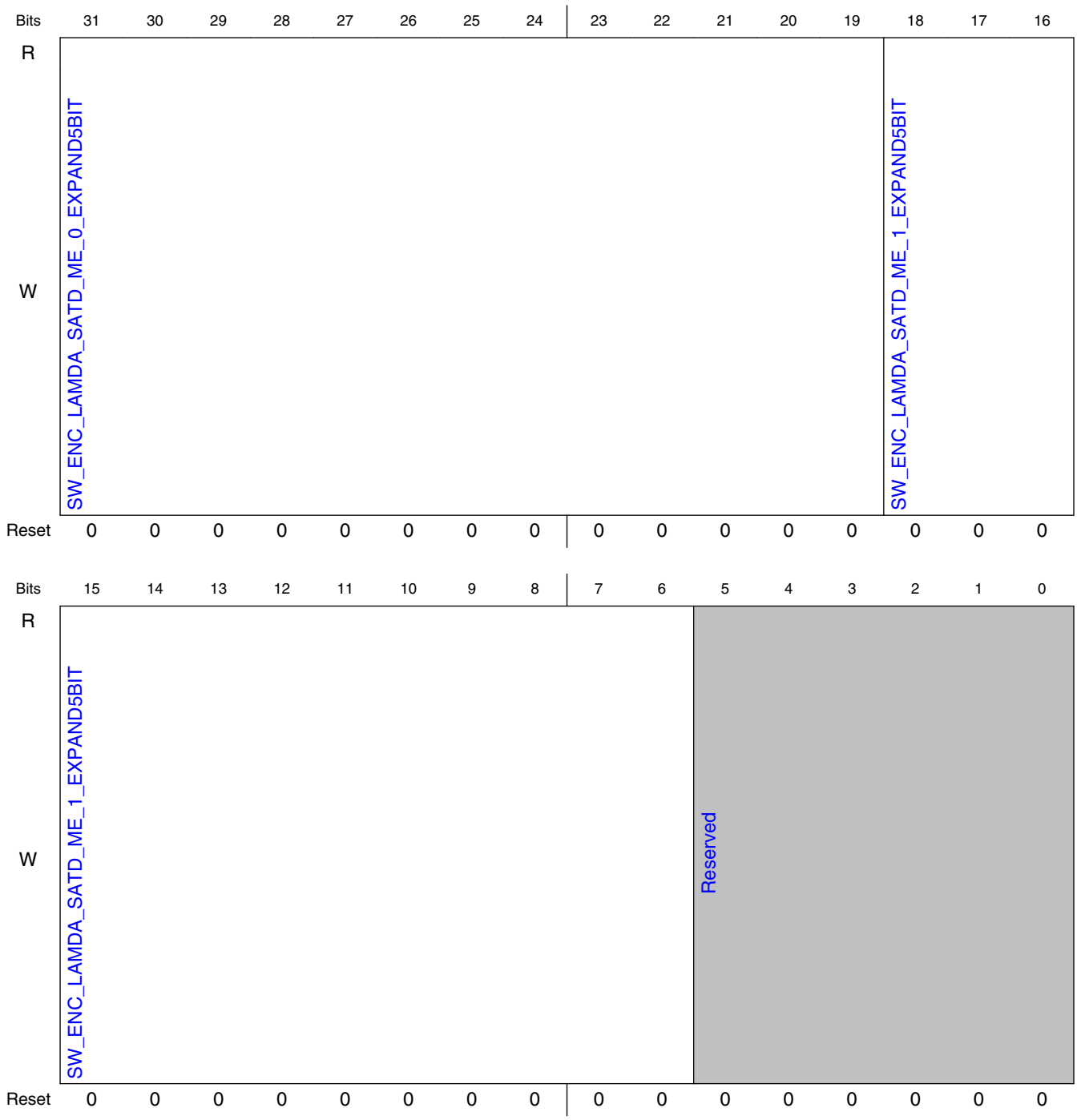
16.3.5.2.29.1 Offset

Register	Offset
swreg28_H2V5	70h

16.3.5.2.29.2 Function

inter me SATD lambda config 0. For H2V5 or later version.

16.3.5.2.29.3 Diagram



16.3.5.2.29.4 Fields

Field	Function
31-19	lambda satd me 0

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_0_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_1_EXPAND5BIT	lambda satd me 1
5-0 —	Reserved

16.3.5.2.30 intra penalty for ROI10. For H2V1 only. (swreg29_H2V1)

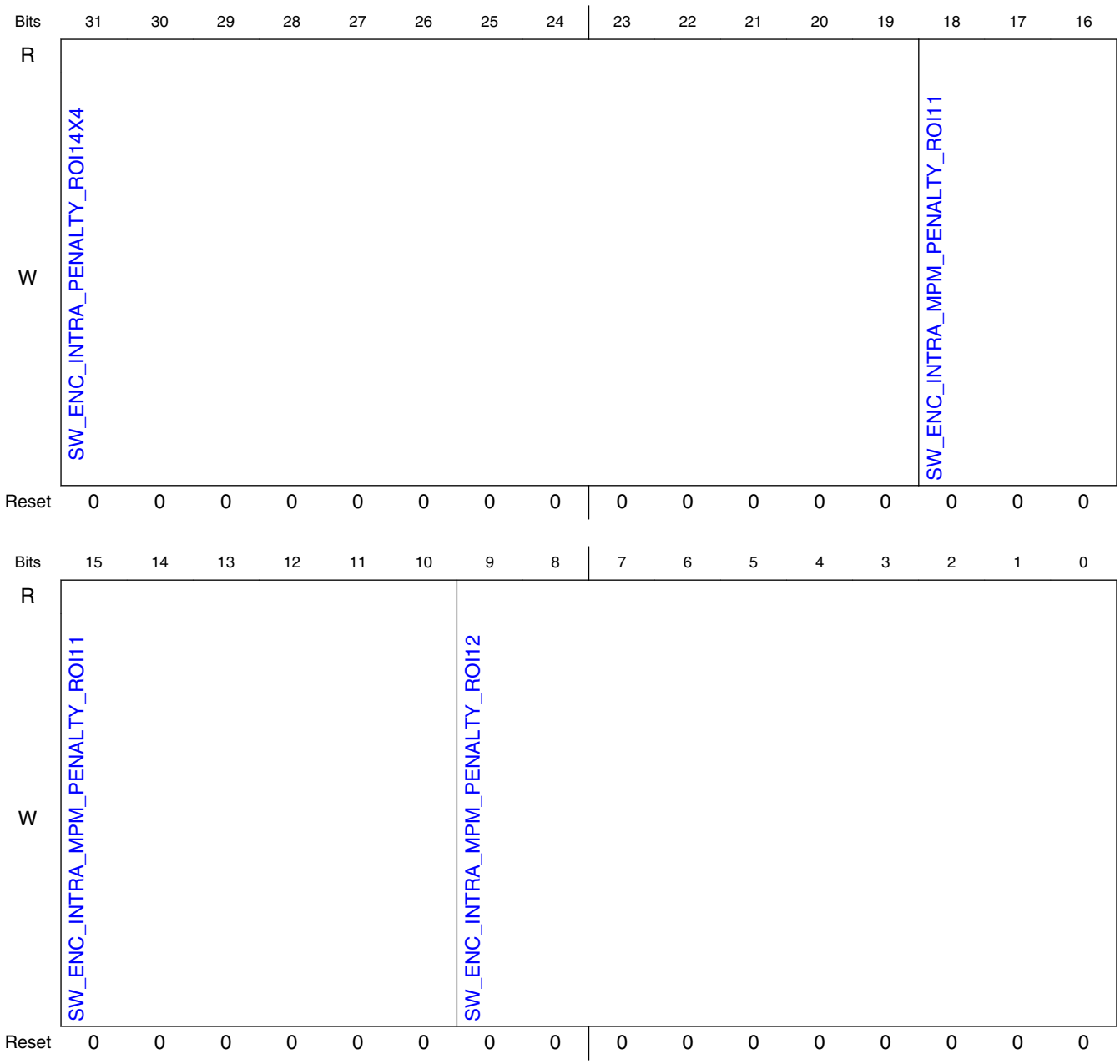
16.3.5.2.30.1 Offset

Register	Offset
swreg29_H2V1	74h

16.3.5.2.30.2 Function

intra penalty for ROI10. For H2V1 only.

16.3.5.2.30.3 Diagram



16.3.5.2.30.4 Fields

Field	Function
31-19 SW_ENC_INTRA_PENALTY_ROI14X4	intra penalty for roi14x4
18-10	intra MPM penalty for roi11

Table continues on the next page...

Field	Function
SW_ENC_INTRA_MPM_PENALTY_ROI11	
9-0 SW_ENC_INTRA_MPM_PENALTY_ROI12	intra MPM penalty for roi12

16.3.5.2.31 inter me SATD lambda config 1.For H2V2 or later version. (swreg29_H2V2)

16.3.5.2.31.1 Offset

Register	Offset
swreg29_H2V2	74h

16.3.5.2.31.2 Function

inter me SATD lambda config 1.For H2V2 or later version.

16.3.5.2.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_LAMDA_SATD_ME_4								SW_ENC_LAMDA_SATD_ME_5							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_LAMDA_SATD_ME_6								SW_ENC_LAMDA_SATD_ME_7							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.31.4 Fields

Field	Function
31-24 SW_ENC_LAMDA_SATD_ME_4	lambda satd me 4

Table continues on the next page...

Field	Function
23-16 SW_ENC_LAMDA_SATD_ME_5	lambda satd me 5
15-8 SW_ENC_LAMDA_SATD_ME_6	lambda satd me 6
7-0 SW_ENC_LAMDA_SATD_ME_7	lambda satd me 7

16.3.5.2.32 inter me SATD lambda config 1.For H2V5 or later version. (swreg29_H2V5)

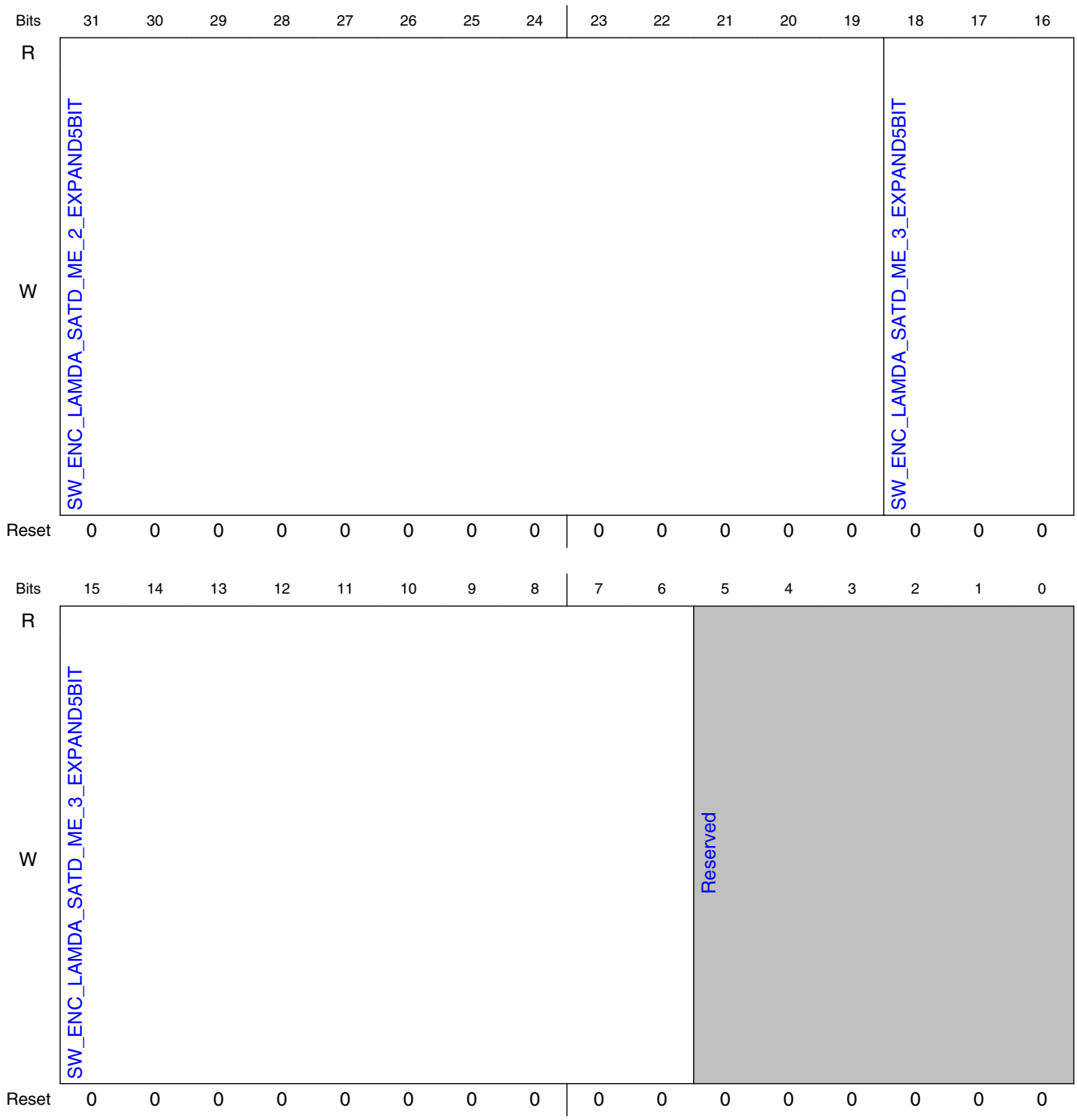
16.3.5.2.32.1 Offset

Register	Offset
swreg29_H2V5	74h

16.3.5.2.32.2 Function

inter me SATD lambda config 1.For H2V5 or later version.

16.3.5.2.32.3 Diagram



16.3.5.2.32.4 Fields

Field	Function
31-19	lambda satd me 2

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_2_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_3_EXPAND5BIT	lambda satd me 3
5-0 —	Reserved

16.3.5.2.33 intra penalty for ROI11. For H2V1 only. (swreg30_H2V1)

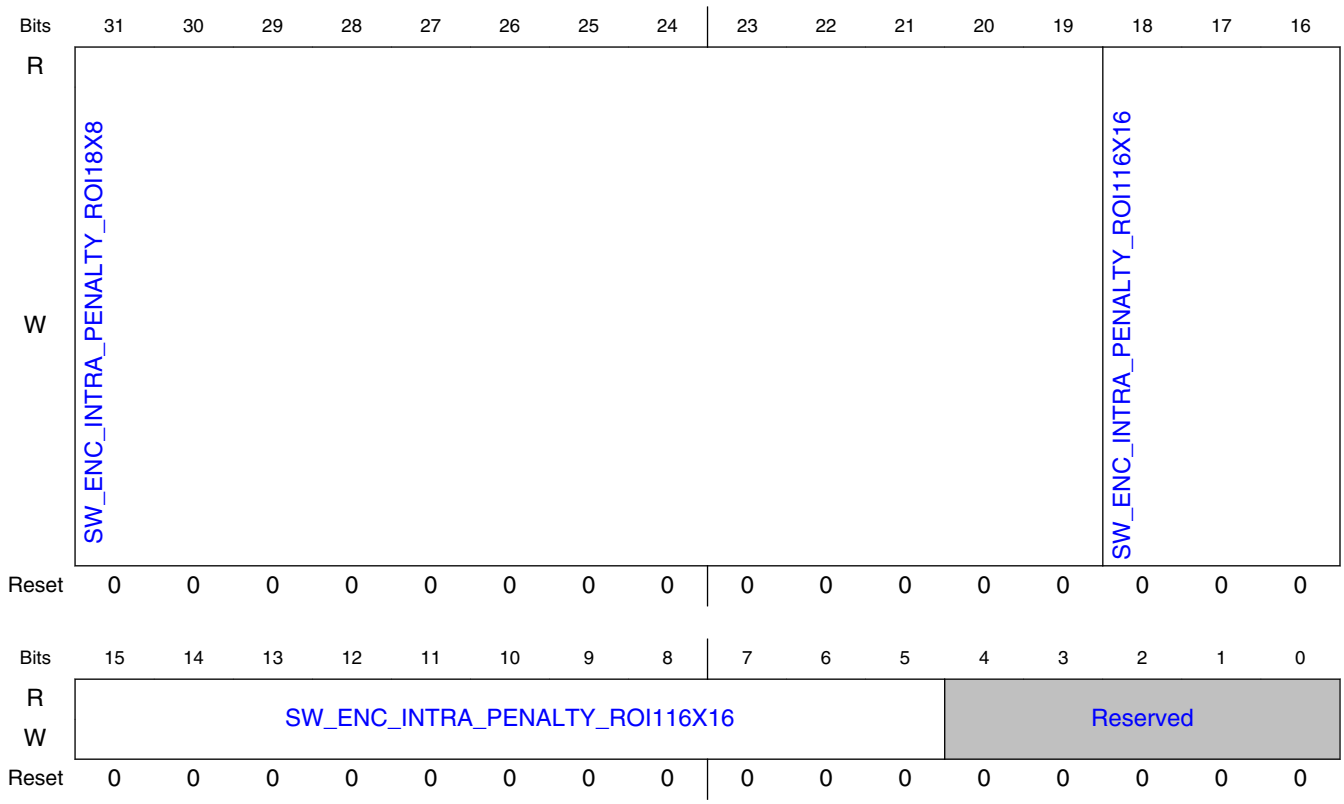
16.3.5.2.33.1 Offset

Register	Offset
swreg30_H2V1	78h

16.3.5.2.33.2 Function

intra penalty for ROI11. For H2V1 only.

16.3.5.2.33.3 Diagram



16.3.5.2.33.4 Fields

Field	Function
31-19 SW_ENC_INTRA_PENALTY_ROI18X8	intra penalty for roi18x8
18-5 SW_ENC_INTRA_PENALTY_ROI116X16	intra penalty for roi116x16
4-0 —	Reserved

16.3.5.2.34 inter me SATD lambda config 2. For H2V2 or later version. (swreg30_H2V2)

16.3.5.2.34.1 Offset

Register	Offset
swreg30_H2V2	78h

16.3.5.2.34.2 Function

inter me SATD lambda config 2. For H2V2 or later version.

16.3.5.2.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_LAMDA_SATD_ME_8								SW_ENC_LAMDA_SATD_ME_9							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_LAMDA_SATD_ME_10								SW_ENC_LAMDA_SATD_ME_11							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.34.4 Fields

Field	Function
31-24 SW_ENC_LAMDA_SATD_ME_8	lambda satd me 8
23-16 SW_ENC_LAMDA_SATD_ME_9	lambda satd me 9
15-8 SW_ENC_LAMDA_SATD_ME_10	lambda satd me 10
7-0 SW_ENC_LAMDA_SATD_ME_11	lambda satd me 11

16.3.5.2.35 inter me SATD lambda config 2. For H2V5 or later version. (swreg30_H2V5)

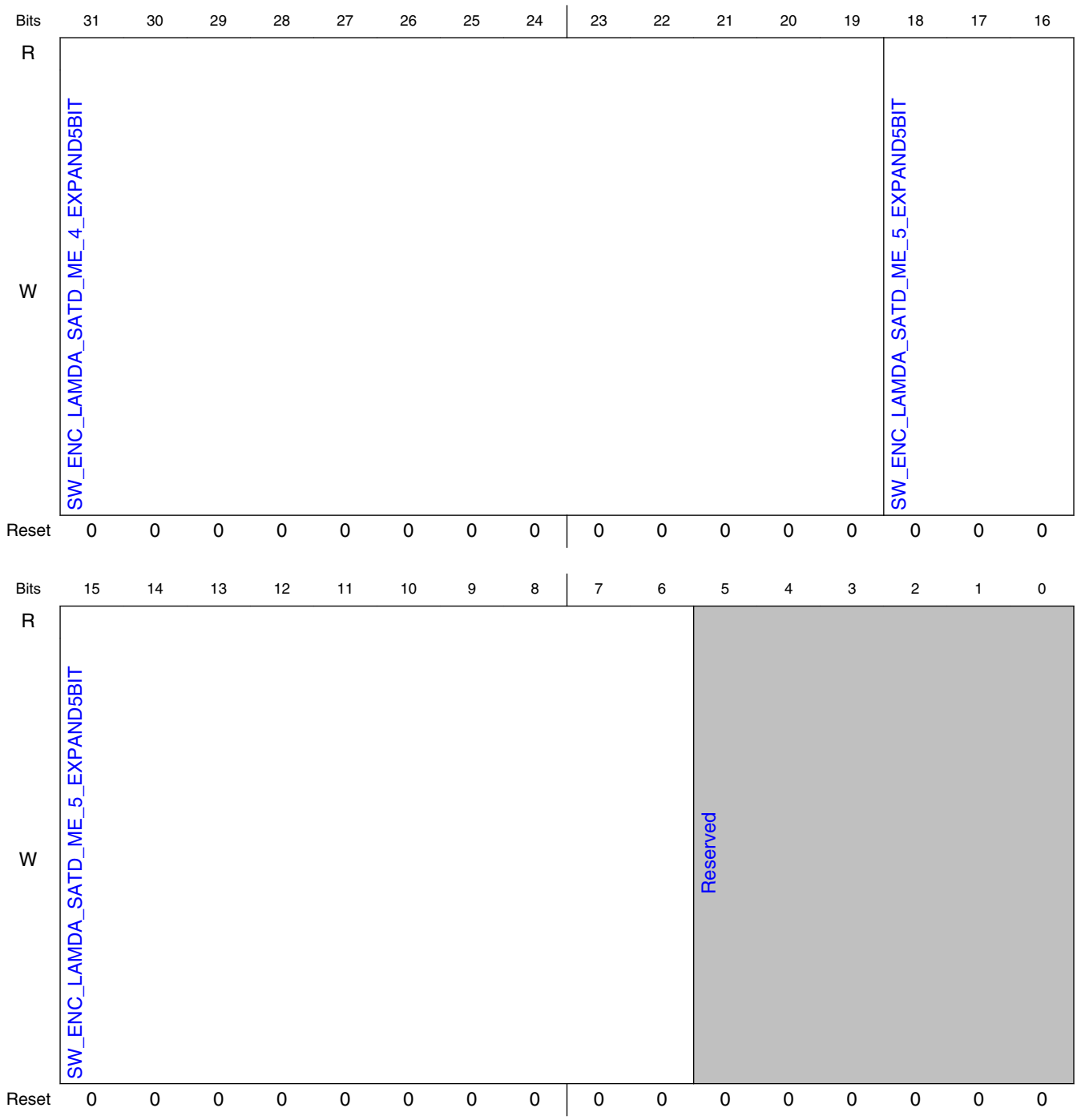
16.3.5.2.35.1 Offset

Register	Offset
swreg30_H2V5	78h

16.3.5.2.35.2 Function

inter me SATD lambda config 2. For H2V5 or later version.

16.3.5.2.35.3 Diagram



16.3.5.2.35.4 Fields

Field	Function
31-19	lambda satd me 4

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_4_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_5_EXPAND5BIT	lambda satd me 5
5-0 —	Reserved

16.3.5.2.36 intra penalty for ROI12. For H2V1 only. (swreg31_H2V1)

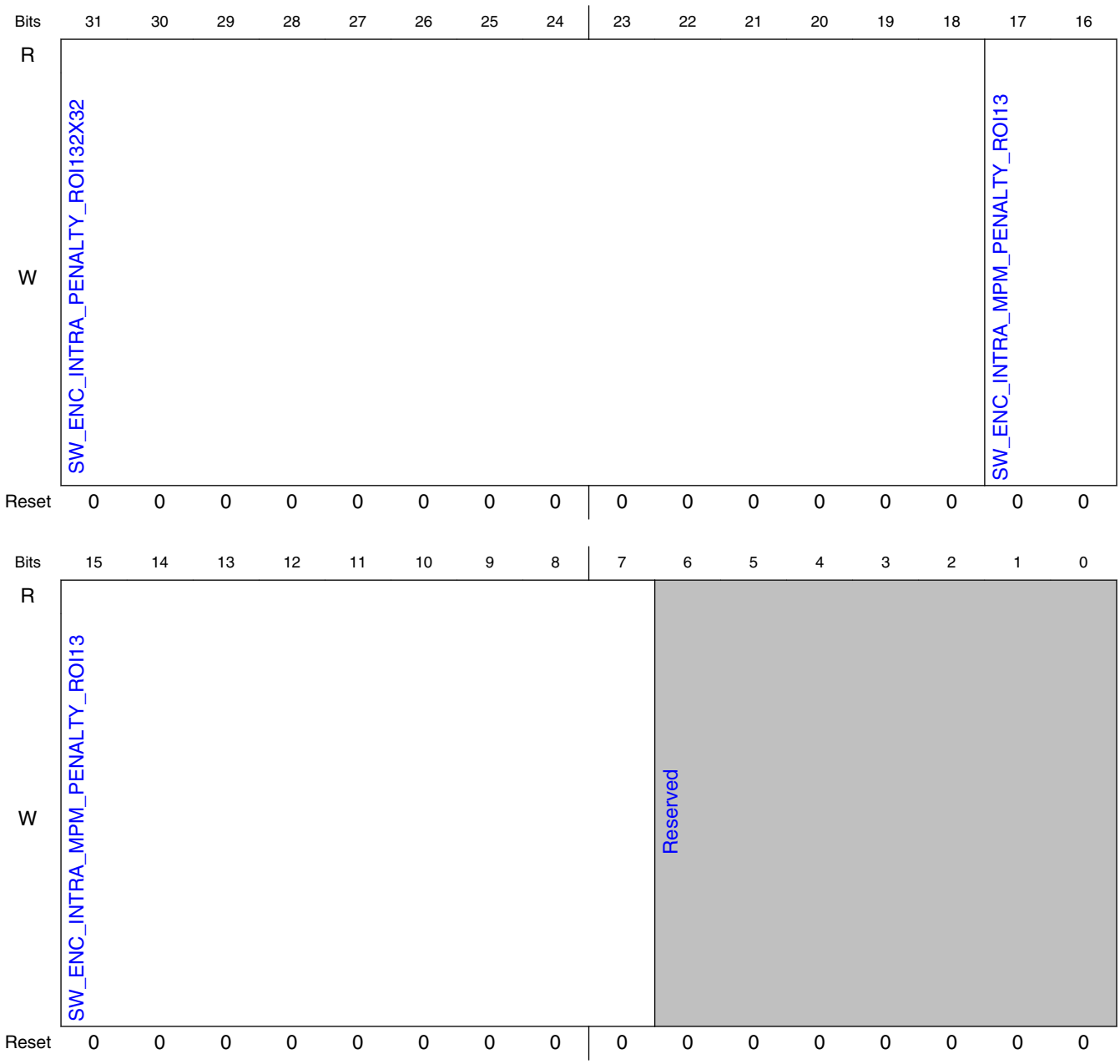
16.3.5.2.36.1 Offset

Register	Offset
swreg31_H2V1	7Ch

16.3.5.2.36.2 Function

intra penalty for ROI12. For H2V1 only.

16.3.5.2.36.3 Diagram



16.3.5.2.36.4 Fields

Field	Function
31-18 SW_ENC_INTRA_PENALTY_ROI132X32	intra penalty for roi132x32
17-7	intra MPM penalty for roi13

Table continues on the next page...

Field	Function
SW_ENC_INTRA_MPM_PENALTY_ROI13	
6-0 —	Reserved

16.3.5.2.37 inter me SATD lambda config 3. For H2V2 or later version. (swreg31_H2V2)

16.3.5.2.37.1 Offset

Register	Offset
swreg31_H2V2	7Ch

16.3.5.2.37.2 Function

inter me SATD lambda config 3. For H2V2 or later version.

16.3.5.2.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_LAMDA_SATD_ME_12								SW_ENC_LAMDA_SATD_ME_13							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_LAMDA_SATD_ME_14								SW_ENC_LAMDA_SATD_ME_15							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.37.4 Fields

Field	Function
31-24 SW_ENC_LAMDA_SATD_ME_12	lambda satd me 12
23-16	lambda satd me 13

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_13	
15-8 SW_ENC_LAMDA_SATD_ME_14	lambda satd me 14
7-0 SW_ENC_LAMDA_SATD_ME_15	lambda satd me 15

16.3.5.2.38 inter me SATD lambda config 3. For H2V5 or later version. (swreg31_H2V5)

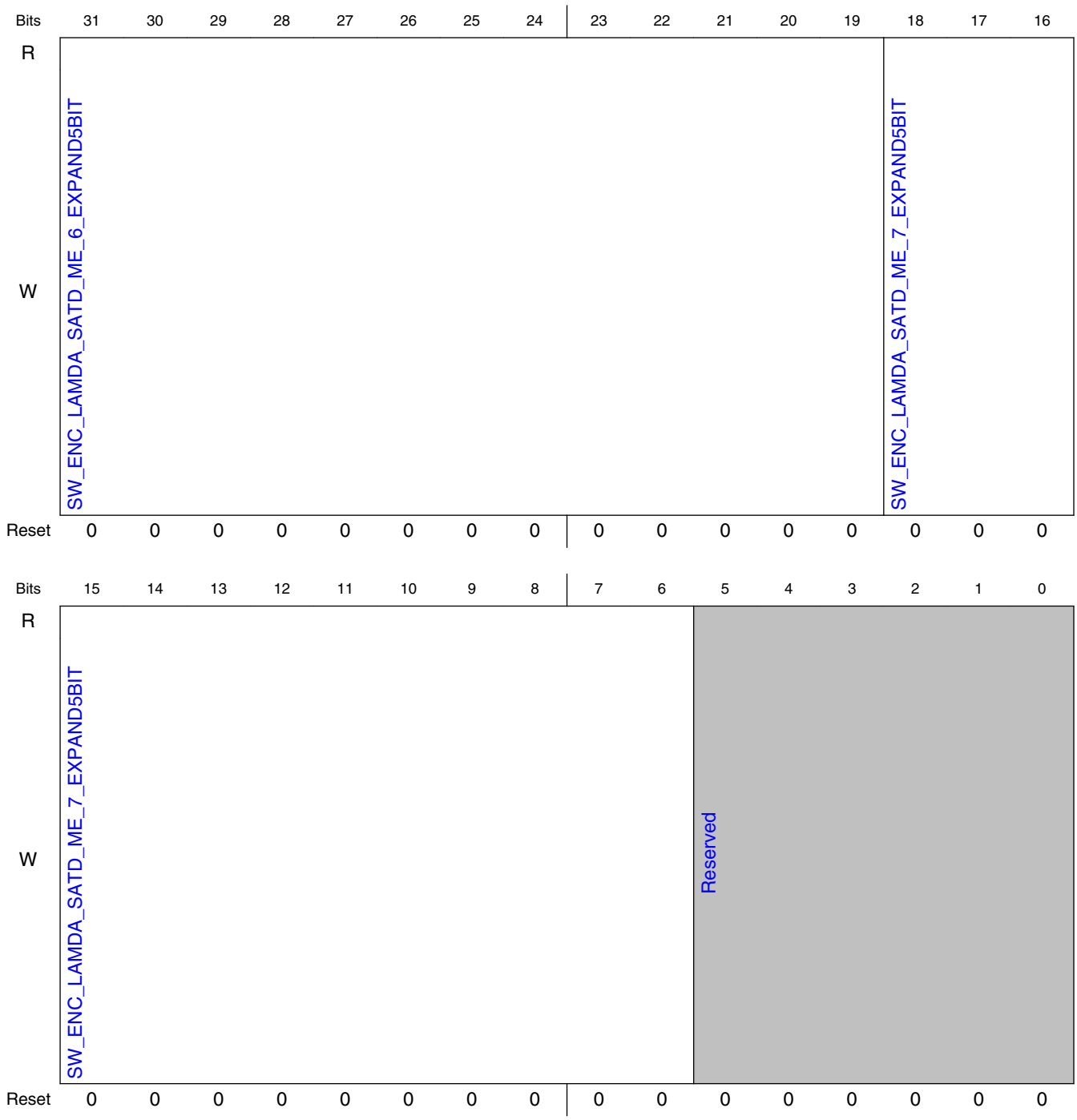
16.3.5.2.38.1 Offset

Register	Offset
swreg31_H2V5	7Ch

16.3.5.2.38.2 Function

inter me SATD lambda config 3. For H2V5 or later version.

16.3.5.2.38.3 Diagram



16.3.5.2.38.4 Fields

Field	Function
31-19	lambda satd me 6

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_6_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_7_EXPAND5BIT	lambda satd me 7
5-0 —	Reserved

16.3.5.2.39 intra penalty for ROI20. For H2V1 only. (swreg32_H2V1)

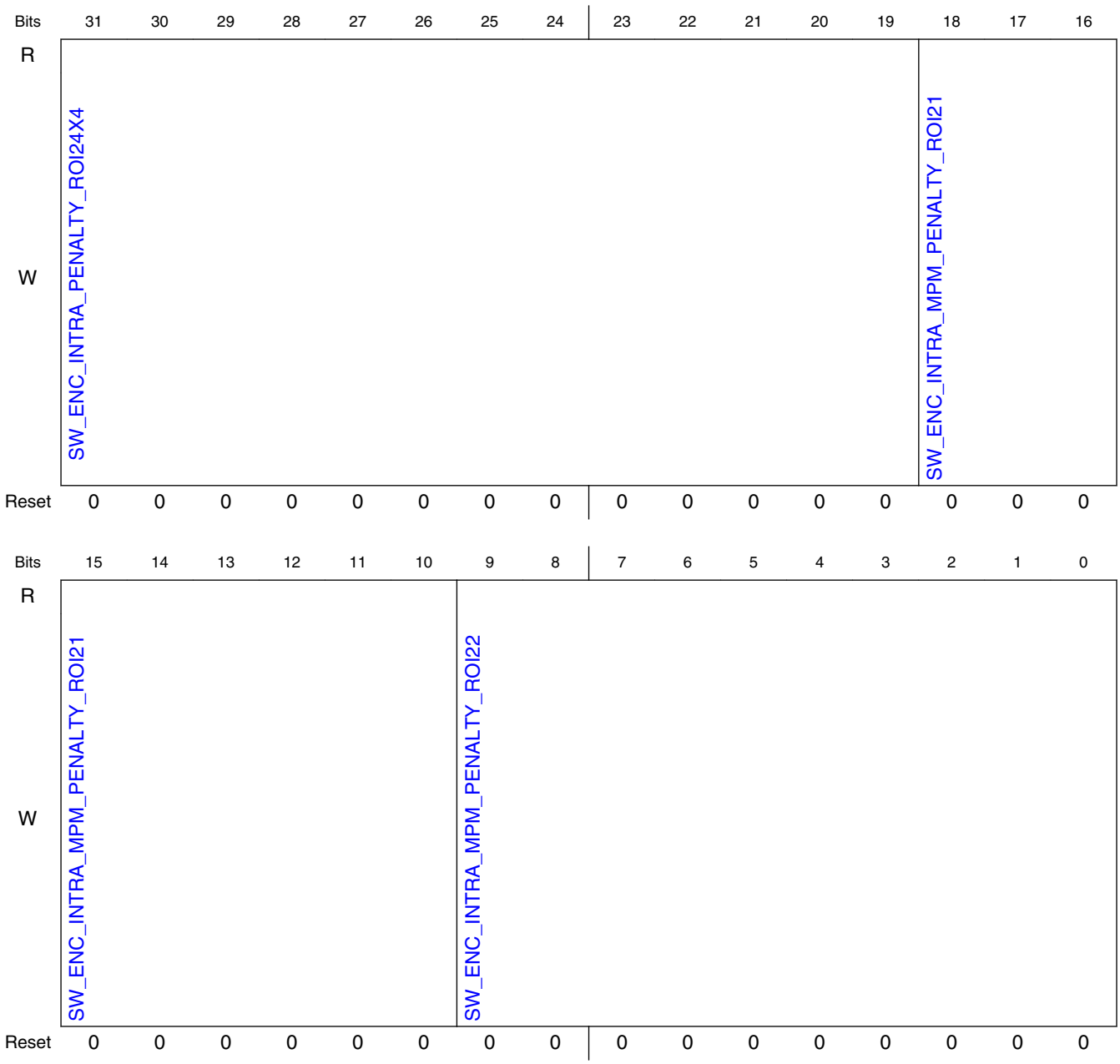
16.3.5.2.39.1 Offset

Register	Offset
swreg32_H2V1	80h

16.3.5.2.39.2 Function

intra penalty for ROI20. For H2V1 only.

16.3.5.2.39.3 Diagram



16.3.5.2.39.4 Fields

Field	Function
31-19 SW_ENC_INTRA_PENALTY_ROI24X4	intra penalty for roi24x4
18-10	intra MPM penalty for roi21

Table continues on the next page...

Field	Function
SW_ENC_INTRA_MPM_PENALTY_ROI21	
9-0 SW_ENC_INTRA_MPM_PENALTY_ROI22	intra MPM penalty for roi22

16.3.5.2.40 inter me SSE lambda config 0. For H2V2 or later version. (swreg32_H2V2)

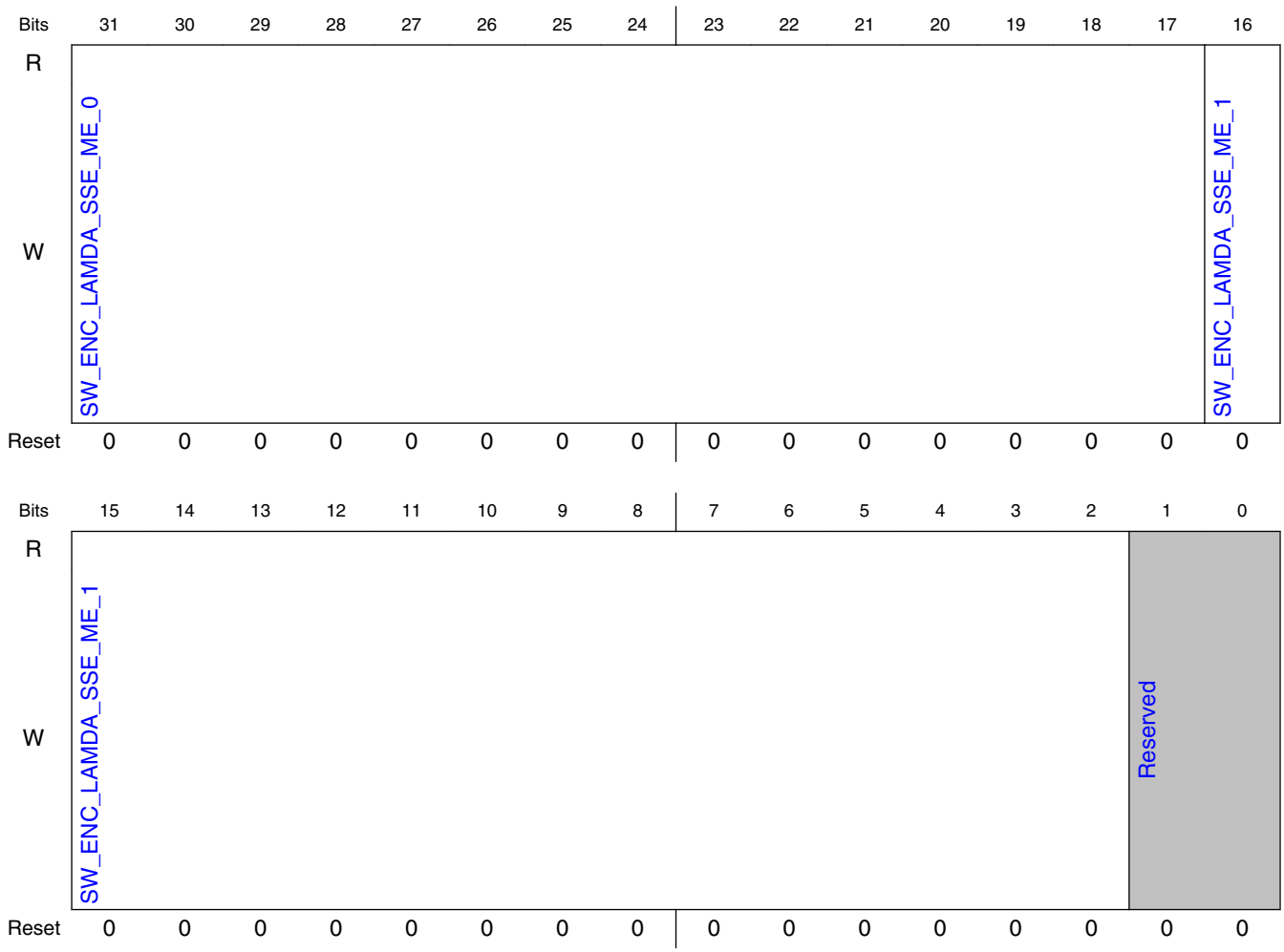
16.3.5.2.40.1 Offset

Register	Offset
swreg32_H2V2	80h

16.3.5.2.40.2 Function

inter me SSE lambda config 0. For H2V2 or later version.

16.3.5.2.40.3 Diagram



16.3.5.2.40.4 Fields

Field	Function
31-17 SW_ENC_LAMDA_SSE_ME_0	inter me SSE lambda 0
16-2 SW_ENC_LAMDA_SSE_ME_1	inter me SSE lambda 1
1-0 —	Reserved

16.3.5.2.41 inter me SATD lambda config 4. For H2V5 or later version. (swreg32_H2V5)

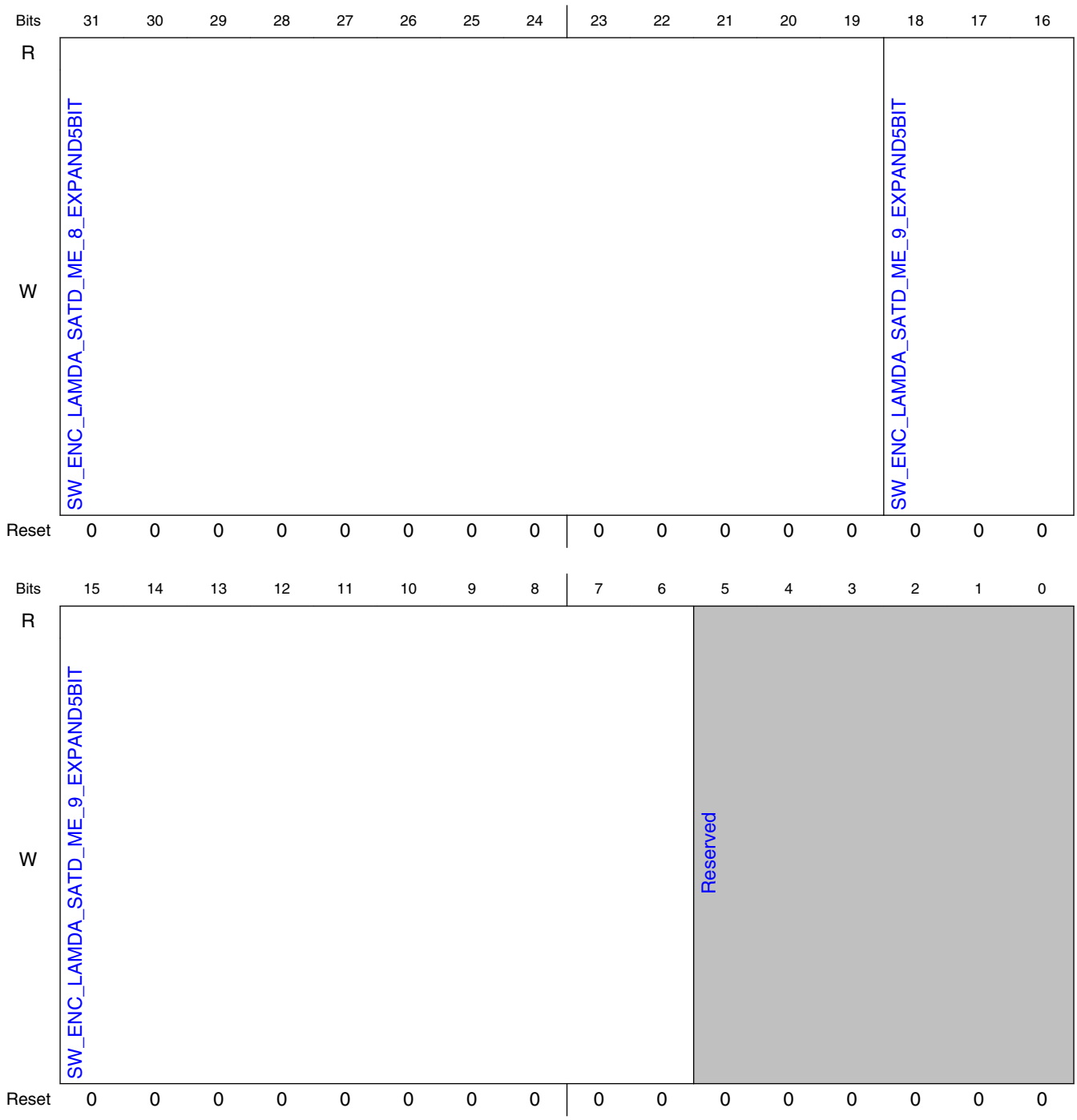
16.3.5.2.41.1 Offset

Register	Offset
swreg32_H2V5	80h

16.3.5.2.41.2 Function

inter me SATD lambda config 4. For H2V5 or later version.

16.3.5.2.41.3 Diagram



16.3.5.2.41.4 Fields

Field	Function
31-19	lambda satd me 8

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_8_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_9_EXPAND5BIT	lambda satd me 9
5-0 —	Reserved

16.3.5.2.42 intra penalty for ROI21. For H2V1 only. (swreg33_H2V1)

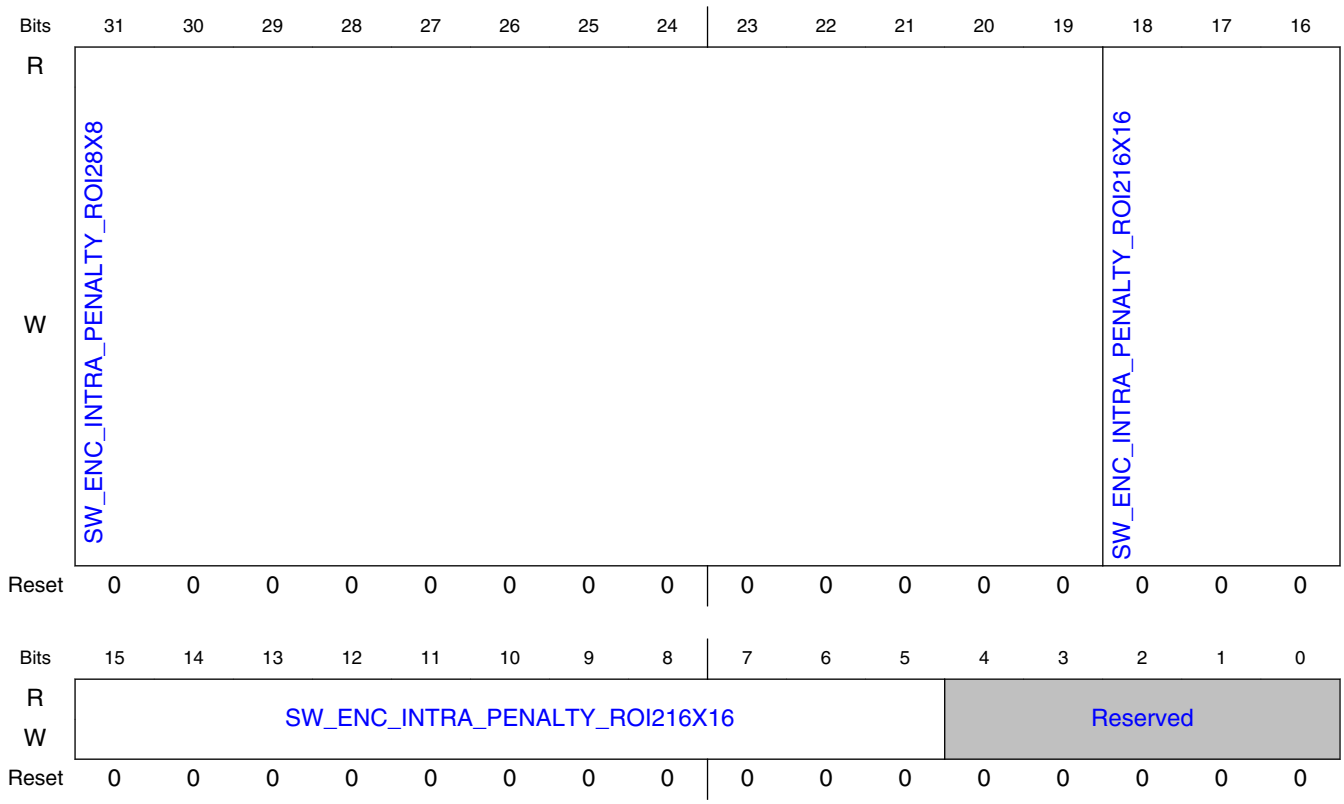
16.3.5.2.42.1 Offset

Register	Offset
swreg33_H2V1	84h

16.3.5.2.42.2 Function

intra penalty for ROI21. For H2V1 only.

16.3.5.2.42.3 Diagram



16.3.5.2.42.4 Fields

Field	Function
31-19 SW_ENC_INTRA_PENALTY_ROI28X8	intra penalty for roi28x8
18-5 SW_ENC_INTRA_PENALTY_ROI216X16	intra penalty for roi216x16
4-0 —	Reserved

16.3.5.2.43 inter me SSE lambda config 1. For H2V2 or later version. (swreg33_H2V2)

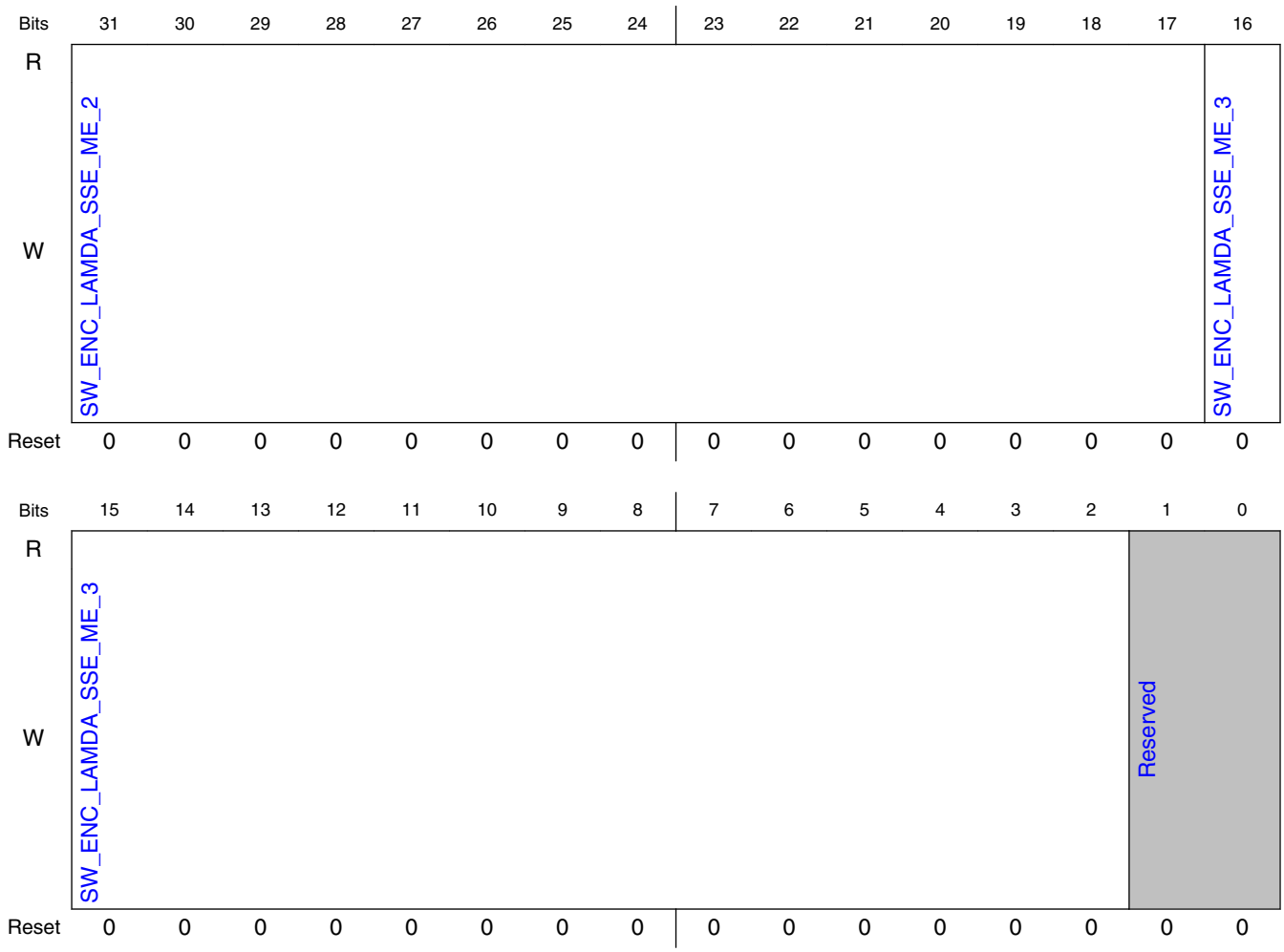
16.3.5.2.43.1 Offset

Register	Offset
swreg33_H2V2	84h

16.3.5.2.43.2 Function

inter me SSE lambda config 1. For H2V2 or later version.

16.3.5.2.43.3 Diagram



16.3.5.2.43.4 Fields

Field	Function
31-17	inter me SSE lambda 2

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SSE_ME_2	
16-2 SW_ENC_LAMDA_SSE_ME_3	inter me SSE lambda 3
1-0 —	Reserved

16.3.5.2.44 inter me SATD lambda config 5. For H2V5 or later version. (swreg33_H2V5)

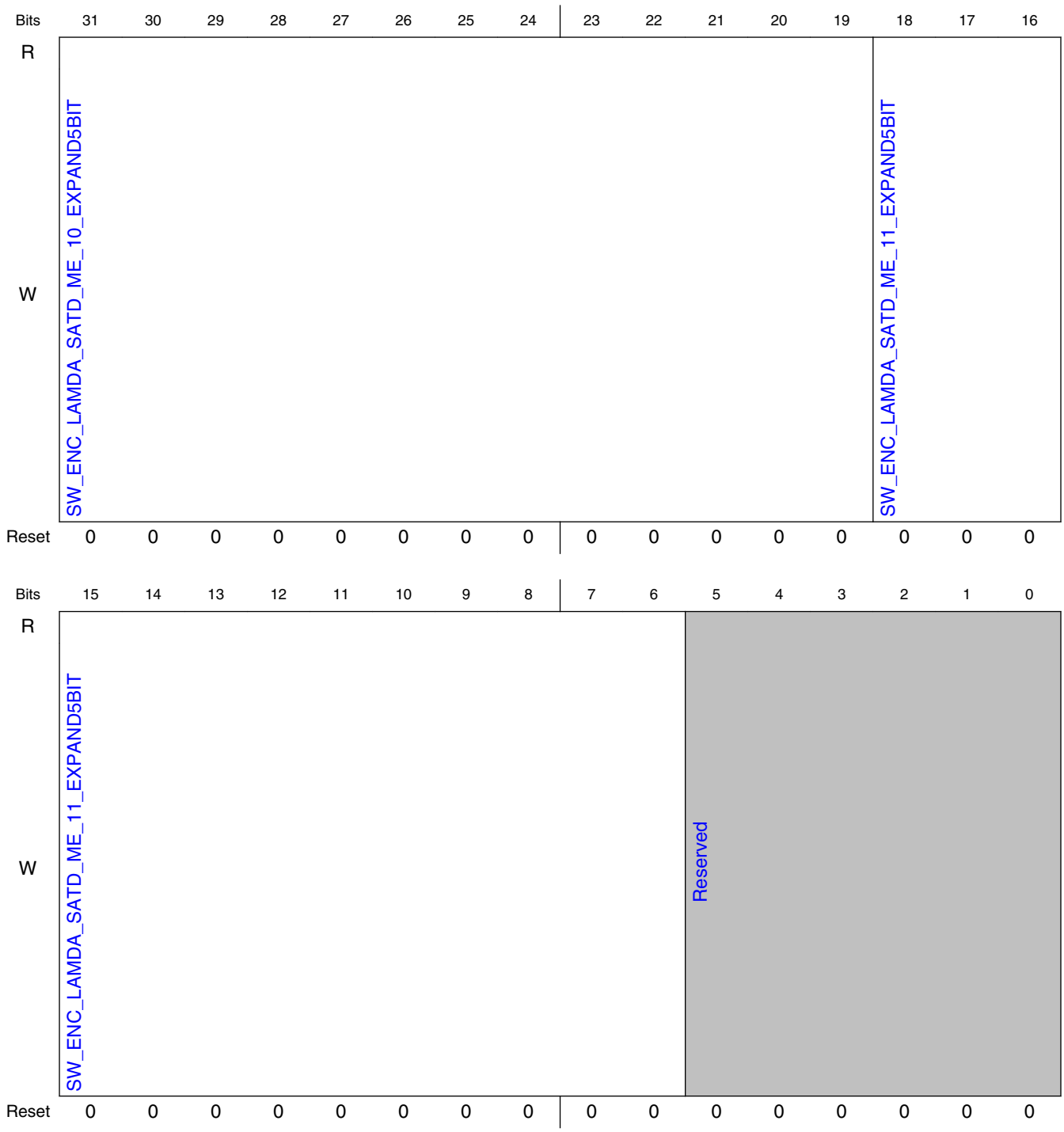
16.3.5.2.44.1 Offset

Register	Offset
swreg33_H2V5	84h

16.3.5.2.44.2 Function

inter me SATD lambda config 5. For H2V5 or later version.

16.3.5.2.44.3 Diagram



16.3.5.2.44.4 Fields

Field	Function
31-19	lambda satd me 10

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_10_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_11_EXPAND5BIT	lambda satd me 11
5-0 —	Reserved

16.3.5.2.45 intra penalty for ROI22. For H2V1 only. (swreg34_H2V1)

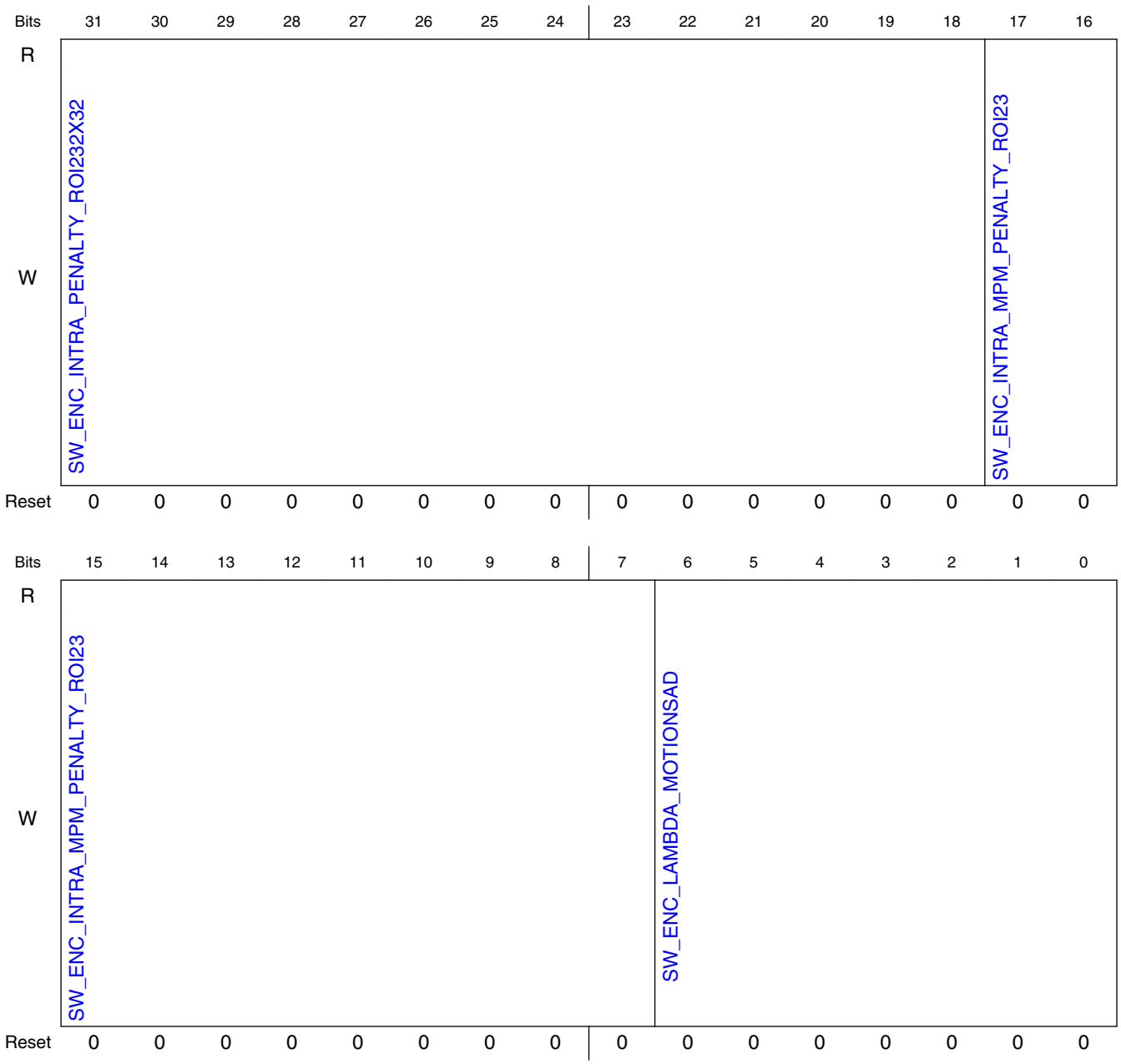
16.3.5.2.45.1 Offset

Register	Offset
swreg34_H2V1	88h

16.3.5.2.45.2 Function

intra penalty for ROI22. For H2V1 only.

16.3.5.2.45.3 Diagram



16.3.5.2.45.4 Fields

Field	Function
31-18 SW_ENC_INTRA_PENALTY_ROI232X32	intra penalty for roi232x32
17-7	intra MPM penalty for roi23

Table continues on the next page...

Field	Function
SW_ENC_INTRA_MPM_PENALTY_ROI23	
6-0 SW_ENC_LAMBDA_MOTIONSAD	lamda for motion SAD

16.3.5.2.46 inter me SSE lambda config 2. For H2V2 or later version. (swreg34_H2V2)

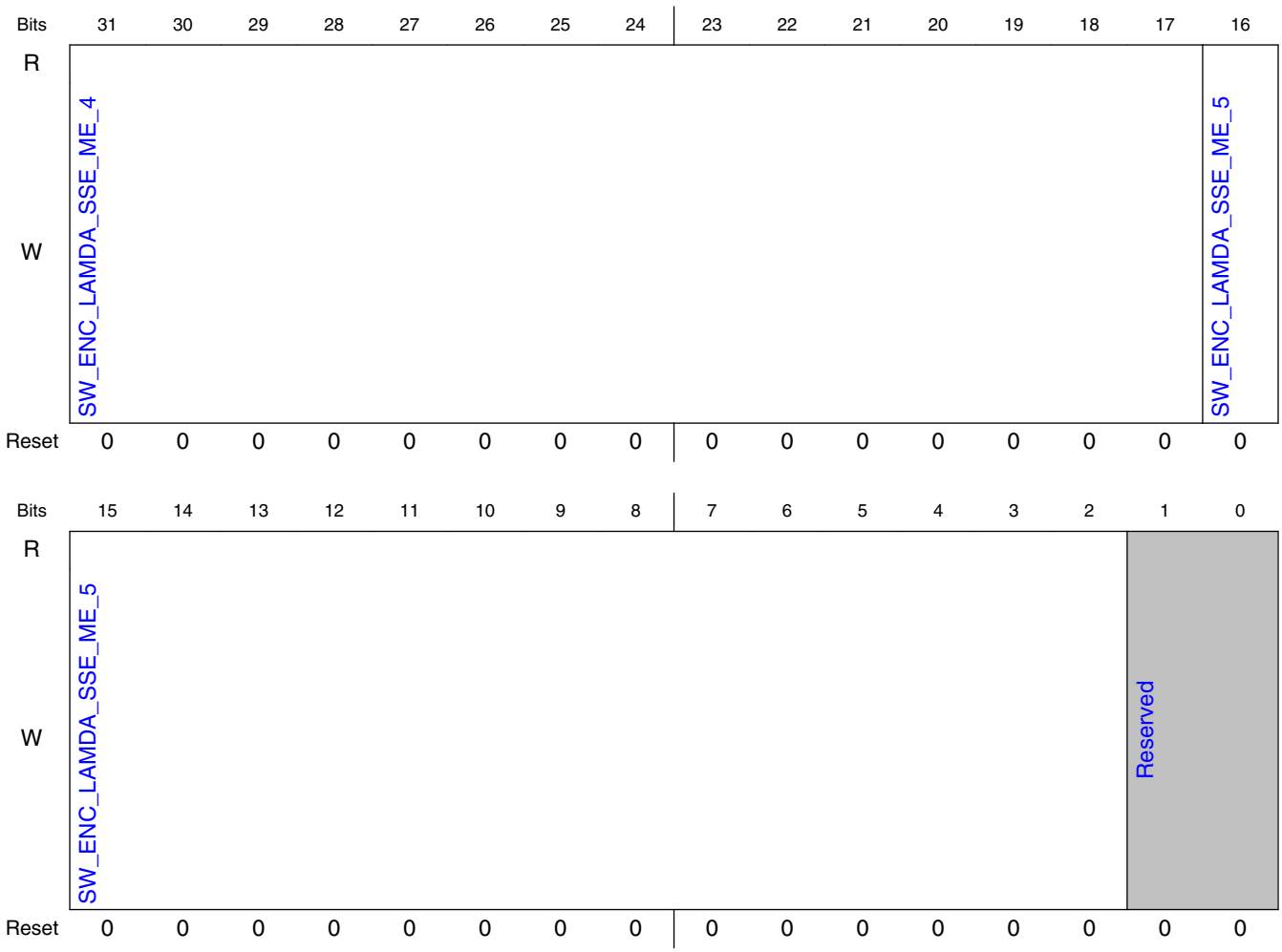
16.3.5.2.46.1 Offset

Register	Offset
swreg34_H2V2	88h

16.3.5.2.46.2 Function

inter me SSE lambda config 2. For H2V2 or later version.

16.3.5.2.46.3 Diagram



16.3.5.2.46.4 Fields

Field	Function
31-17 SW_ENC_LAMDA_SSE_ME_4	inter me SSE lambda 4
16-2 SW_ENC_LAMDA_SSE_ME_5	inter me SSE lambda 5
1-0 —	Reserved

16.3.5.2.47 inter me SATD lambda config 6. For H2V5 or later version. (swreg34_H2V5)

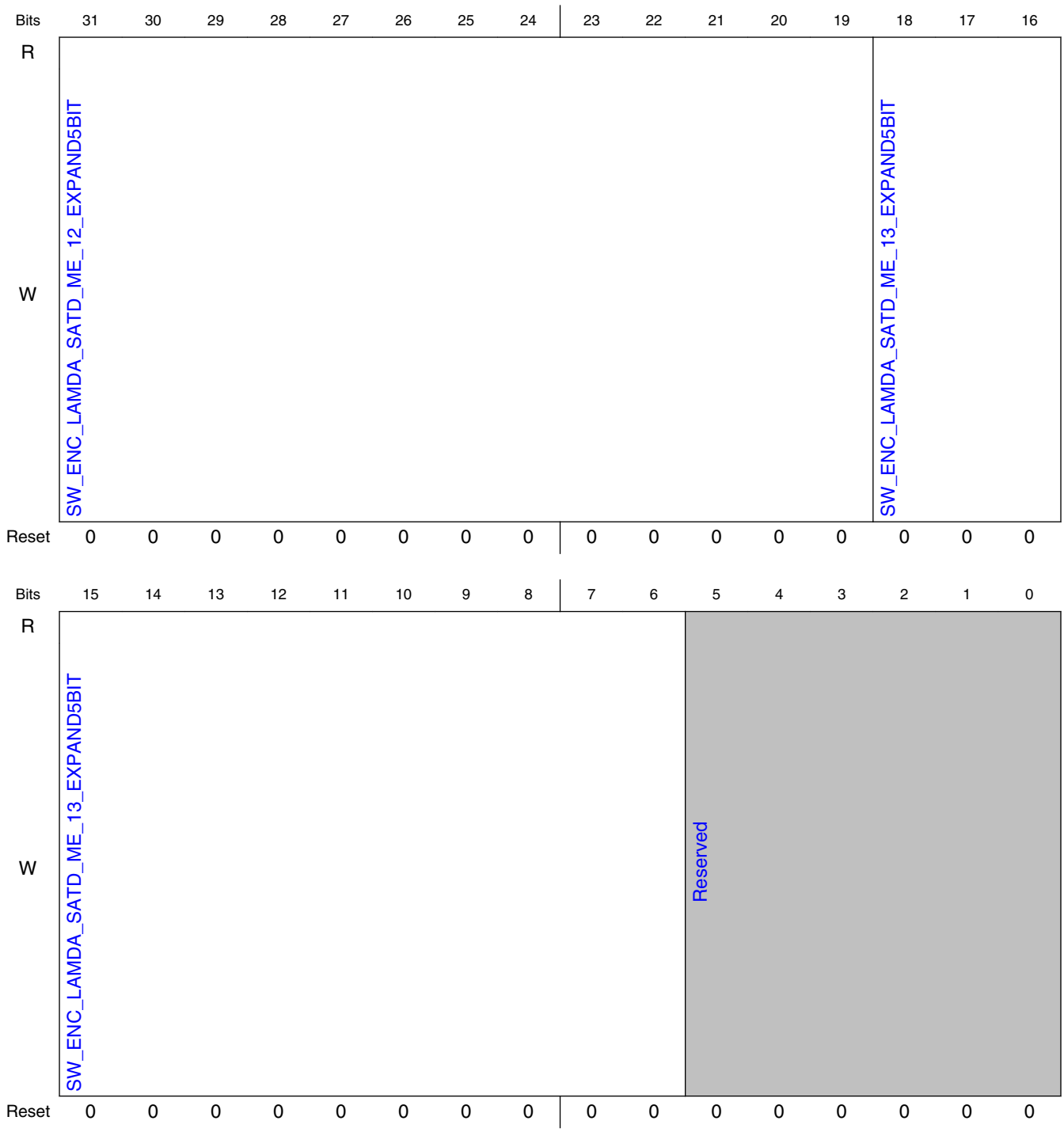
16.3.5.2.47.1 Offset

Register	Offset
swreg34_H2V5	88h

16.3.5.2.47.2 Function

inter me SATD lambda config 6. For H2V5 or later version.

16.3.5.2.47.3 Diagram



16.3.5.2.47.4 Fields

Field	Function
31-19	lambda satd me 12

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_12_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_13_EXPAND5BIT	lambda satd me 13
5-0 —	Reserved

16.3.5.2.48 inter prediction parameters1 (swreg35)

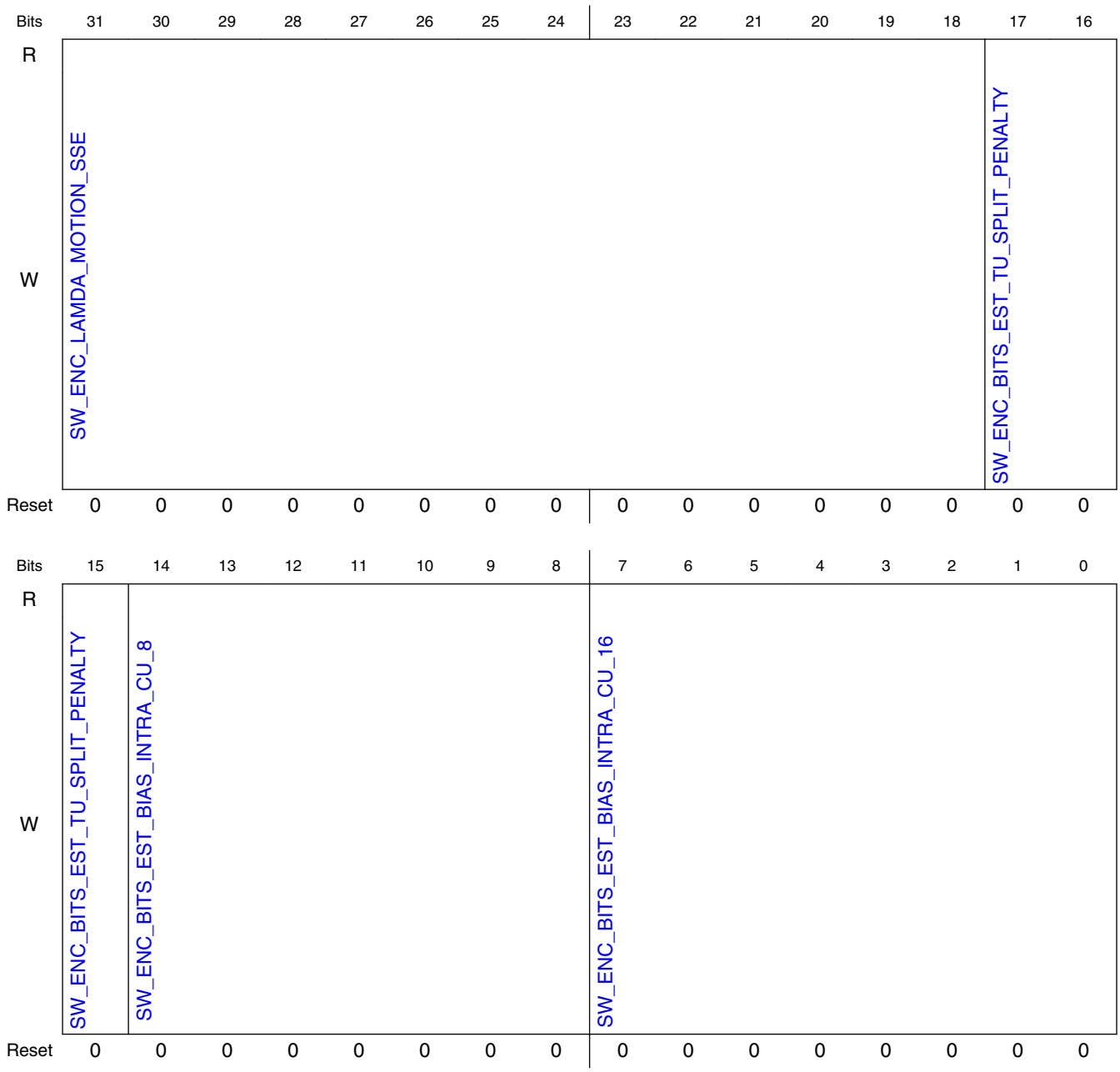
16.3.5.2.48.1 Offset

Register	Offset
swreg35	8Ch

16.3.5.2.48.2 Function

inter prediction parameters1

16.3.5.2.48.3 Diagram



16.3.5.2.48.4 Fields

Field	Function
31-18 SW_ENC_LAMDA_MOTION_SSE	lambda for motion SSE

Table continues on the next page...

Field	Function
17-15 SW_ENC_BITS_EST_TU_SPLIT_PENALTY	bits estimation for tu split penalty
14-8 SW_ENC_BITS_EST_BIAS_INTRA_CU_8	bits estimation bias for intra cu 8
7-0 SW_ENC_BITS_EST_BIAS_INTRA_CU_16	bits estimation bias for intra cu 16

16.3.5.2.49 inter prediction parameters2 (swreg36)

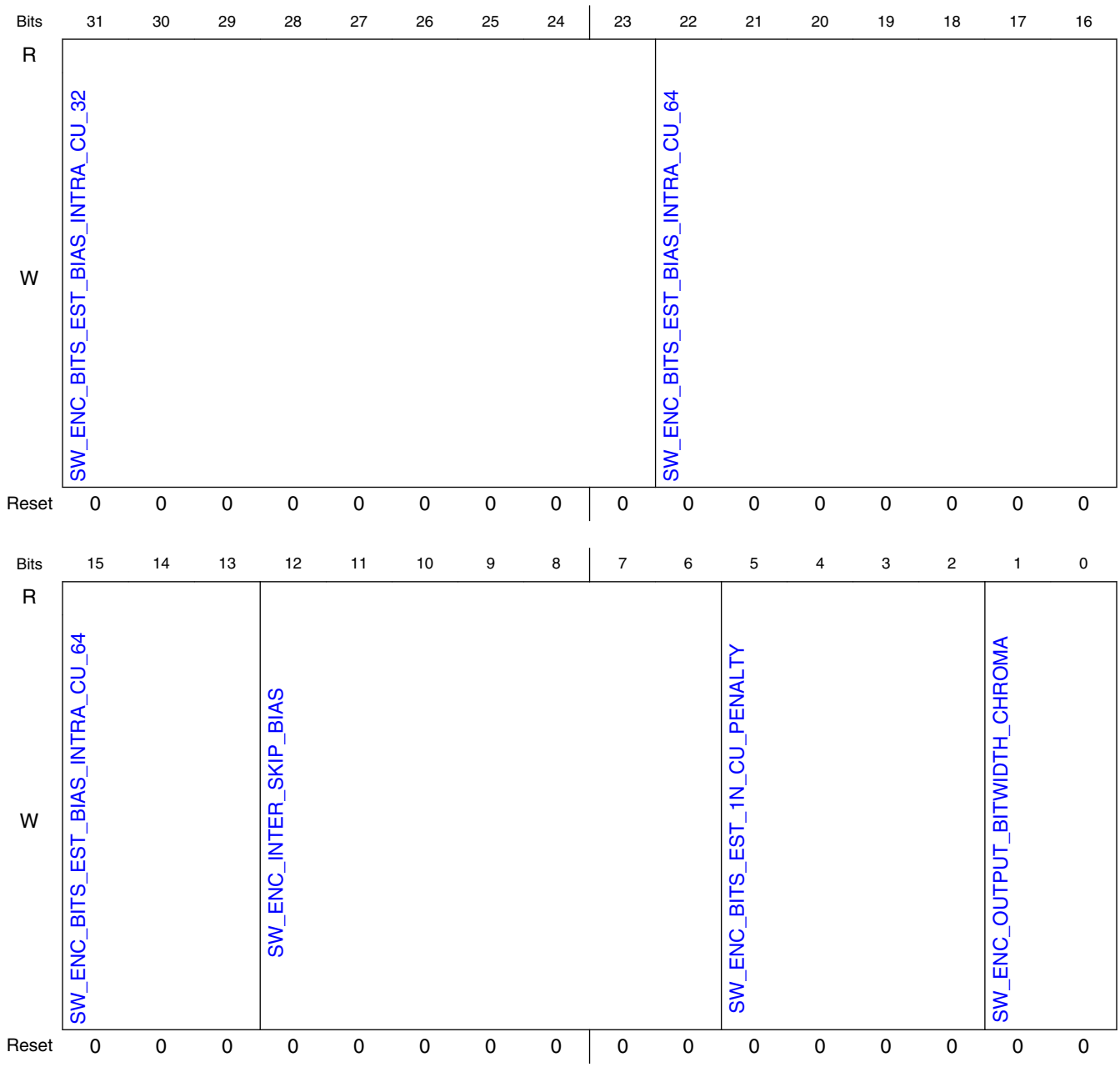
16.3.5.2.49.1 Offset

Register	Offset
swreg36	90h

16.3.5.2.49.2 Function

inter prediction parameters2

16.3.5.2.49.3 Diagram



16.3.5.2.49.4 Fields

Field	Function
31-23 SW_ENC_BITS_EST_BIAS_INTRA_CU_32	bits estimation bias for intra cu 32
22-13	bits estimation bias for intra cu 64

Table continues on the next page...

Field	Function
SW_ENC_BITS_EST_BIAS_INTRA_CU_64	
12-6 SW_ENC_INTER_SKIP_BIAS	inter skip bias
5-2 SW_ENC_BITS_EST_1N_CU_PENALTY	bits estimation 1N cu penalty
1-0 SW_ENC_OUTPUT_BITWIDTH_CHROMA	chroma output bitwidth 00b - 8 bit. 01b - 9 bit. 10b - 10 bit.

16.3.5.2.50 SAO lambda parameter (swreg37)

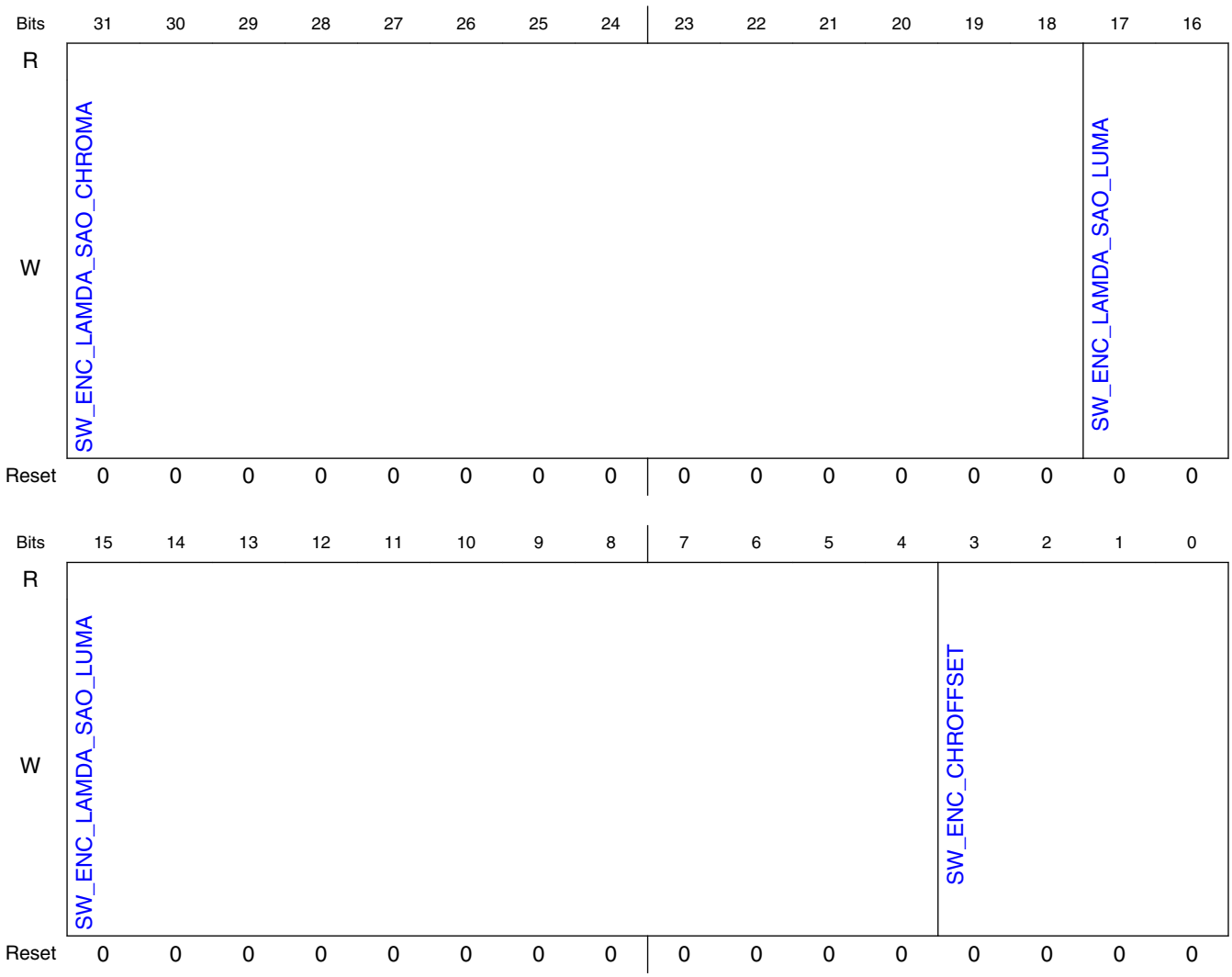
16.3.5.2.50.1 Offset

Register	Offset
swreg37	94h

16.3.5.2.50.2 Function

SAO lambda parameter

16.3.5.2.50.3 Diagram



16.3.5.2.50.4 Fields

Field	Function
31-18 SW_ENC_LAMDA_SAO_CHROMA	lambda for SAO chroma
17-4 SW_ENC_LAMDA_SAO_LUMA	lambda for SAO luma
3-0 SW_ENC_CHROFFSET	Input chrominance offset (bytes) [0..15]

16.3.5.2.51 Pre-processor configuration (swreg38)

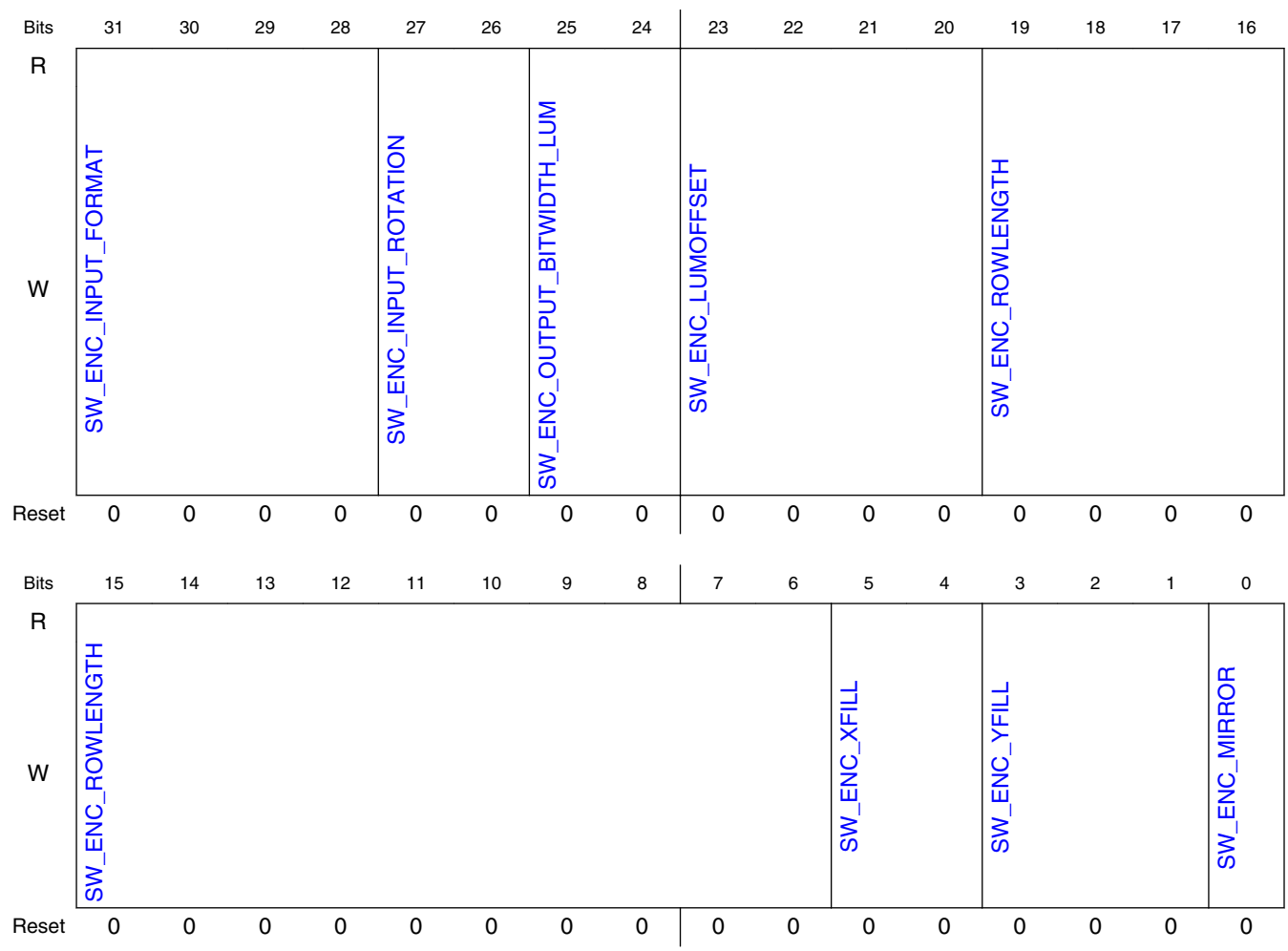
16.3.5.2.51.1 Offset

Register	Offset
swreg38	98h

16.3.5.2.51.2 Function

Pre-processor configuration

16.3.5.2.51.3 Diagram



16.3.5.2.51.4 Fields

Field	Function
31-28 SW_ENC_INPUT_FORMAT	Input image format. inputFormat. 0=YUV420P 0001b - YUV420SP 0010b - YUYV422 0011b - UYVY422 0100b - RGB565 0101b - RGB555 0110b - RGB444 0111b - RGB888 1000b - RGB101010 1001b - I010 1010b - P010 1011b - PACKED10BITPLANAR 1100b - Y0L2 1101b - DAHUAHEVC 1110b - DAHUAH264
27-26 SW_ENC_INPUT_ROTATION	Input image rotation. 00b - disabled. 01b - 90 degrees right. 10b - 90 degrees left. 11b - Reserved.
25-24 SW_ENC_OUTPUT_BITWIDTH_LUM	luma output bitwidth 00b - 8 bit. 01b - 9 bit. 10b - 10 bit.
23-20 SW_ENC_LUM_OFFSET	Input luminance offset (bytes) [0..15]
19-6 SW_ENC_ROW_LENGTH	Input luminance row length. lumWidthSrc (pixels) [96..8192]
5-4 SW_ENC_XFILL	Overfill pixels on right edge of image div2 [0.1.2.3]
3-1 SW_ENC_YFILL	Overfill pixels on bottom edge of image. YFill. [0...7]
0 SW_ENC_MIRROR	mirror or not. [0.1]

16.3.5.2.52 Pre-processor color conversion parameters0 (swreg39)

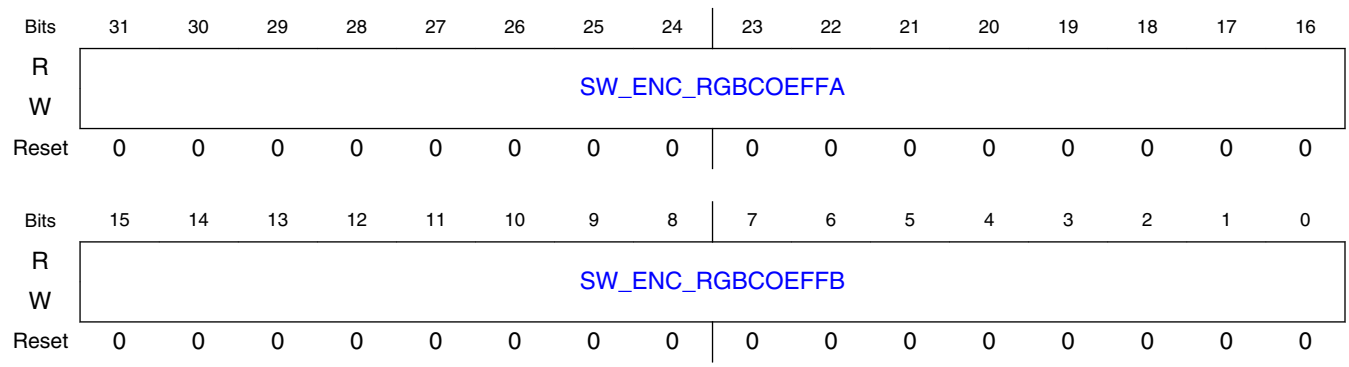
16.3.5.2.52.1 Offset

Register	Offset
swreg39	9Ch

16.3.5.2.52.2 Function

Pre-processor color conversion parameters0

16.3.5.2.52.3 Diagram



16.3.5.2.52.4 Fields

Field	Function
31-16 SW_ENC_RGBCOEFFA	RGB to YUV conversion coefficient A
15-0 SW_ENC_RGBCOEFFB	RGB to YUV conversion coefficient B

16.3.5.2.53 Pre-processor color conversion parameters1 (swreg40)

16.3.5.2.53.1 Offset

Register	Offset
swreg40	A0h

16.3.5.2.53.2 Function

Pre-processor color conversion parameters1

16.3.5.2.53.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RGBCOEFFC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RGBCOEFFE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.53.4 Fields

Field	Function
31-16 SW_ENC_RGBCOEFFC	RGB to YUV conversion coefficient C
15-0 SW_ENC_RGBCOEFFE	RGB to YUV conversion coefficient D

16.3.5.2.54 Pre-processor color conversion parameters2 (swreg41)

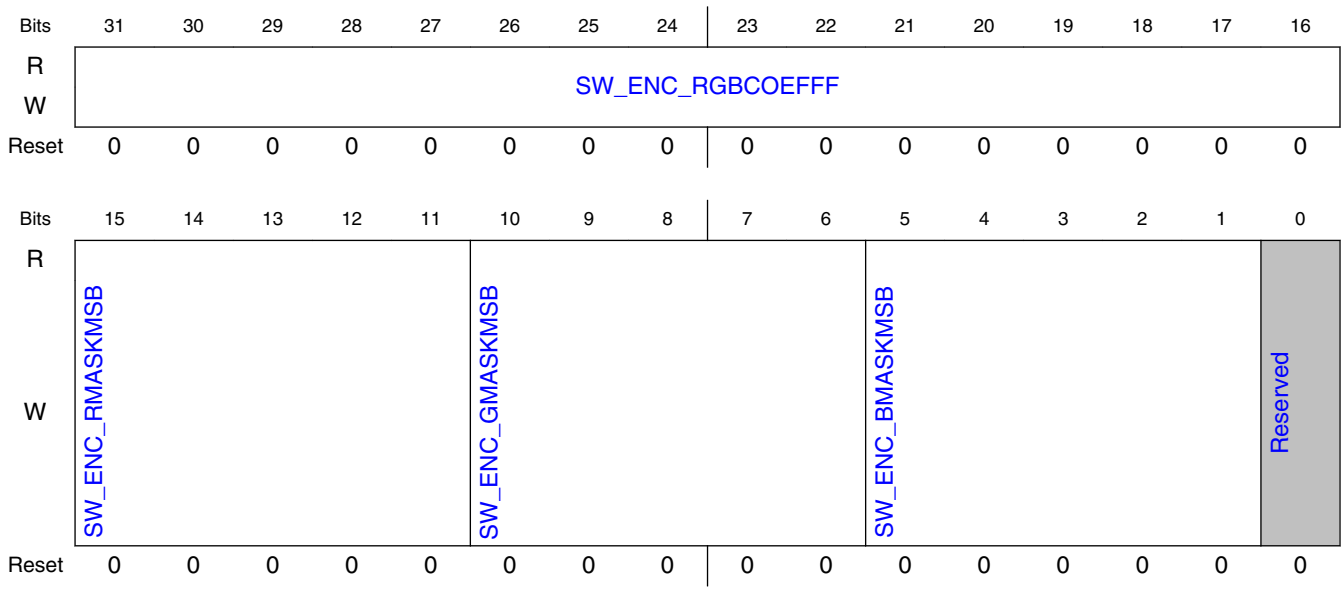
16.3.5.2.54.1 Offset

Register	Offset
swreg41	A4h

16.3.5.2.54.2 Function

Pre-processor color conversion parameters2

16.3.5.2.54.3 Diagram



16.3.5.2.54.4 Fields

Field	Function
31-16 SW_ENC_RGBCOEFFF	RGB to YUV conversion coefficient E
15-11 SW_ENC_RMASKMSB	RGB R-component mask MSB bit position [0..31]
10-6 SW_ENC_GMASKMSB	RGB G-component mask MSB bit position [0..31]
5-1 SW_ENC_BMASKMSB	RGB B-component mask MSB bit position [0..31]
0 —	Reserved

16.3.5.2.55 compressed coefficients base address for SAN module. (swreg46)

16.3.5.2.55.1 Offset

Register	Offset
swreg46	B8h

16.3.5.2.55.2 Function

compressed coefficients base address for SAN module.

16.3.5.2.55.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_COMPRESSEDCOEFF_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_COMPRESSEDCOEFF_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.55.4 Fields

Field	Function
31-0 SW_ENC_COMPRESSEDCOEFF_BASE	Base address for compressed coefficients.

16.3.5.2.56 Base address for recon luma compress table LSB. (swreg60)

16.3.5.2.56.1 Offset

Register	Offset
swreg60	F0h

16.3.5.2.56.2 Function

Base address for recon luma compress table LSB.

16.3.5.2.56.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RECON_LUMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RECON_LUMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.56.4 Fields

Field	Function
31-0 SW_ENC_RECON_LUMA_COMPRESS_TABLE_BASE	Base address for recon luma compress table LSB.

16.3.5.2.57 Base address for recon Chroma compress table LSB (swreg62)

16.3.5.2.57.1 Offset

Register	Offset
swreg62	F8h

16.3.5.2.57.2 Function

Base address for recon Chroma compress table LSB

16.3.5.2.57.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RECON_CHROMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RECON_CHROMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.57.4 Fields

Field	Function
31-0 SW_ENC_RECON_CHROMA_COMPRESS_TABLE_BASE	Base address for recon Chroma compress table LSB.

16.3.5.2.58 Base address for list 0 ref 0 luma compress table LSB. (swreg64)

16.3.5.2.58.1 Offset

Register	Offset
swreg64	100h

16.3.5.2.58.2 Function

Base address for list 0 ref 0 luma compress table LSB.

16.3.5.2.58.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_L0_REF0_LUMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_L0_REF0_LUMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.58.4 Fields

Field	Function
31-0 SW_ENC_L0_REF0_LUMA_COMPRESS_TABLE_BASE	Base address for list 0 ref 0 luma compress table LSB.

16.3.5.2.59 Base address for list 0 ref 0 Chroma compress table LSB. (swreg66)

16.3.5.2.59.1 Offset

Register	Offset
swreg66	108h

16.3.5.2.59.2 Function

Base address for list 0 ref 0 Chroma compress table LSB.

16.3.5.2.59.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_L0_REF0_CHROMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_L0_REF0_CHROMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.59.4 Fields

Field	Function
31-0 SW_ENC_L0_REF0_CHROMA_COMPRESS_TABLE_BASE	Base address for list 0 ref 0 Chroma compress table LSB.

16.3.5.2.60 Base address for recon luma 4n base LSB. (swreg72)

16.3.5.2.60.1 Offset

Register	Offset
swreg72	120h

16.3.5.2.60.2 Function

Base address for recon luma 4n base LSB.

16.3.5.2.60.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RECON_LUMA_4N_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RECON_LUMA_4N_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.60.4 Fields

Field	Function
31-0 SW_ENC_RECON_LUMA_4N_BASE	Base address for recon luma 4n base LSB.

16.3.5.2.61 reference picture reconstructed list0 4n 0 (swreg74)

16.3.5.2.61.1 Offset

Register	Offset
swreg74	128h

16.3.5.2.61.2 Function

reference picture reconstructed list0 4n 0

16.3.5.2.61.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REFPIC_RECON_L0_4N0_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_REFPIC_RECON_L0_4N0_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.61.4 Fields

Field	Function
31-0 SW_ENC_REFPIC_RECON_L0_4N0_BASE	reference picture reconstructed list0 4n 0

16.3.5.2.62 inter prediction parameters3. For H2V1 only. (swreg78_H2V1)

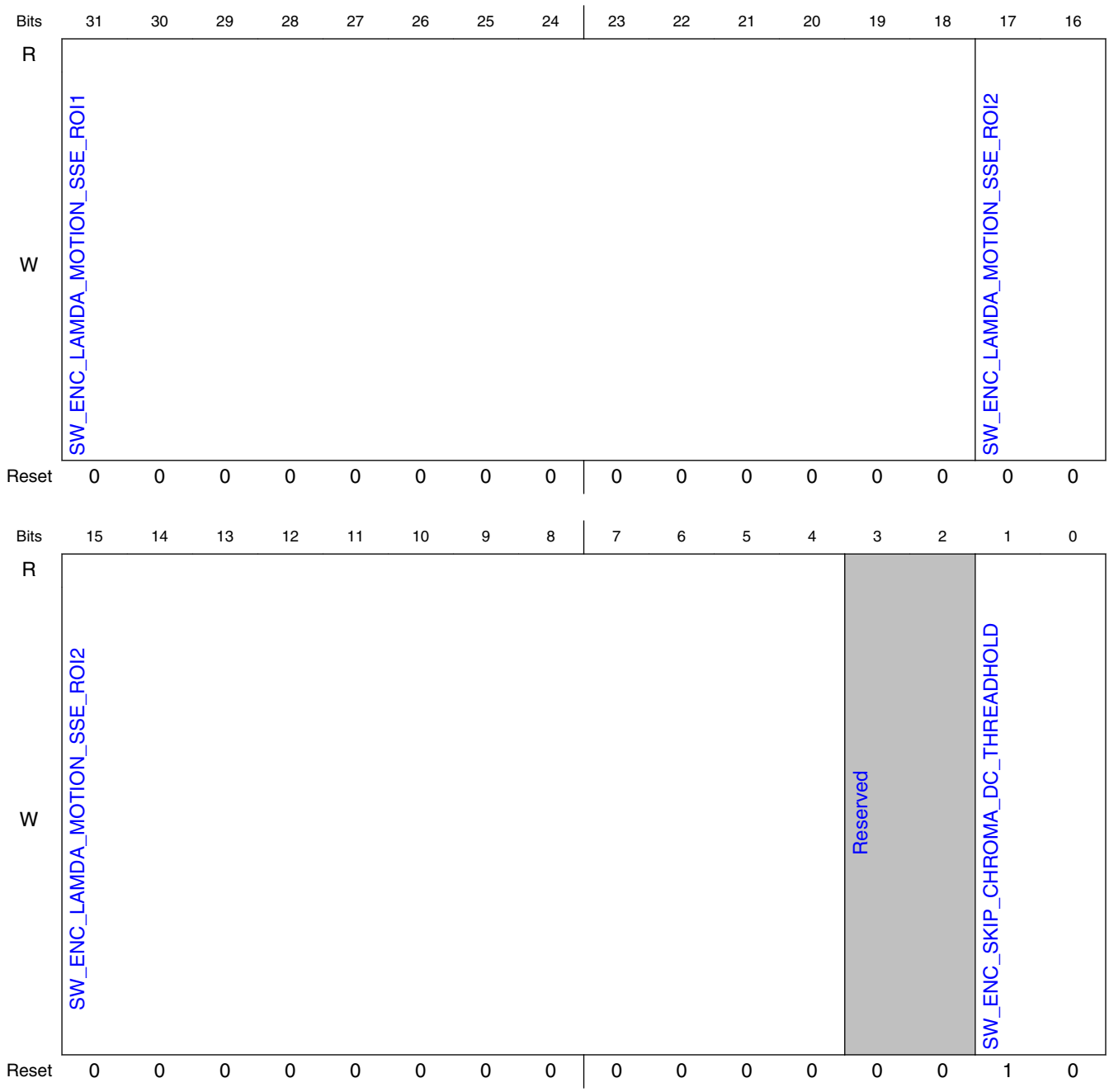
16.3.5.2.62.1 Offset

Register	Offset
swreg78_H2V1	138h

16.3.5.2.62.2 Function

inter prediction parameters3. For H2V1 only.

16.3.5.2.62.3 Diagram



16.3.5.2.62.4 Fields

Field	Function
31-18 SW_ENC_LAMDA_MOTION_SSE_ROI1	roi1 lambda for motion SSE

Table continues on the next page...

Field	Function
17-4 SW_ENC_LAMDA_MOTION_SSE_ROI2	roi2 lambda for motion SSE
3-2 —	Reserved
1-0 SW_ENC_SKIP_CHROMA_DC_THREADHOLD	skip chroma dc threadhold

16.3.5.2.63 inter me SSE lambda config 3. For H2V2 or later version. (swreg78_H2V2)

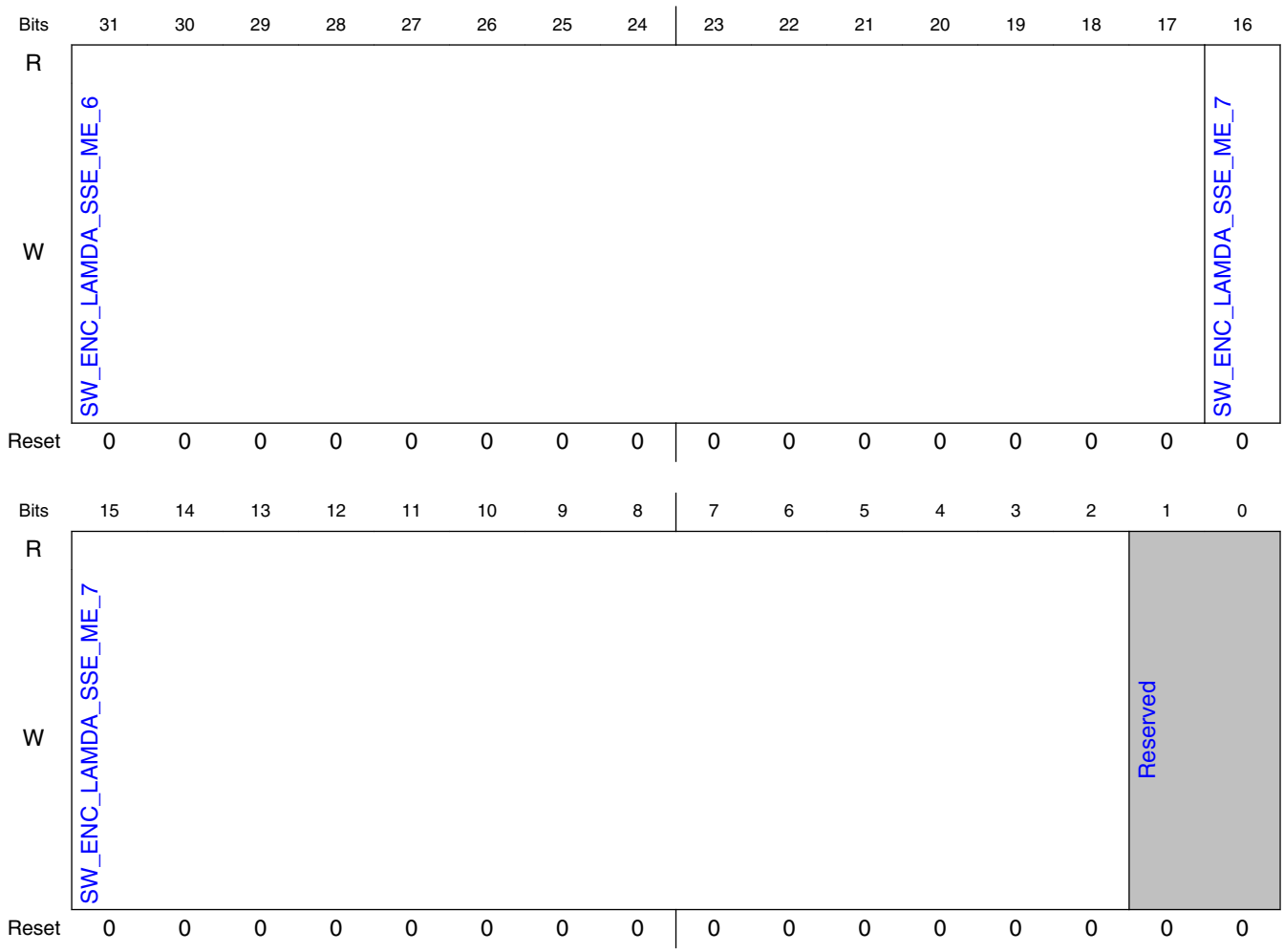
16.3.5.2.63.1 Offset

Register	Offset
swreg78_H2V2	138h

16.3.5.2.63.2 Function

inter me SSE lambda config 3. For H2V2 or later version.

16.3.5.2.63.3 Diagram



16.3.5.2.63.4 Fields

Field	Function
31-17 SW_ENC_LAMDA_SSE_ME_6	inter me SSE lambda 6
16-2 SW_ENC_LAMDA_SSE_ME_7	inter me SSE lambda 7
1-0 —	Reserved

16.3.5.2.64 inter me SATD lambda config 7. For H2V5 or later version. (swreg78_H2V5)

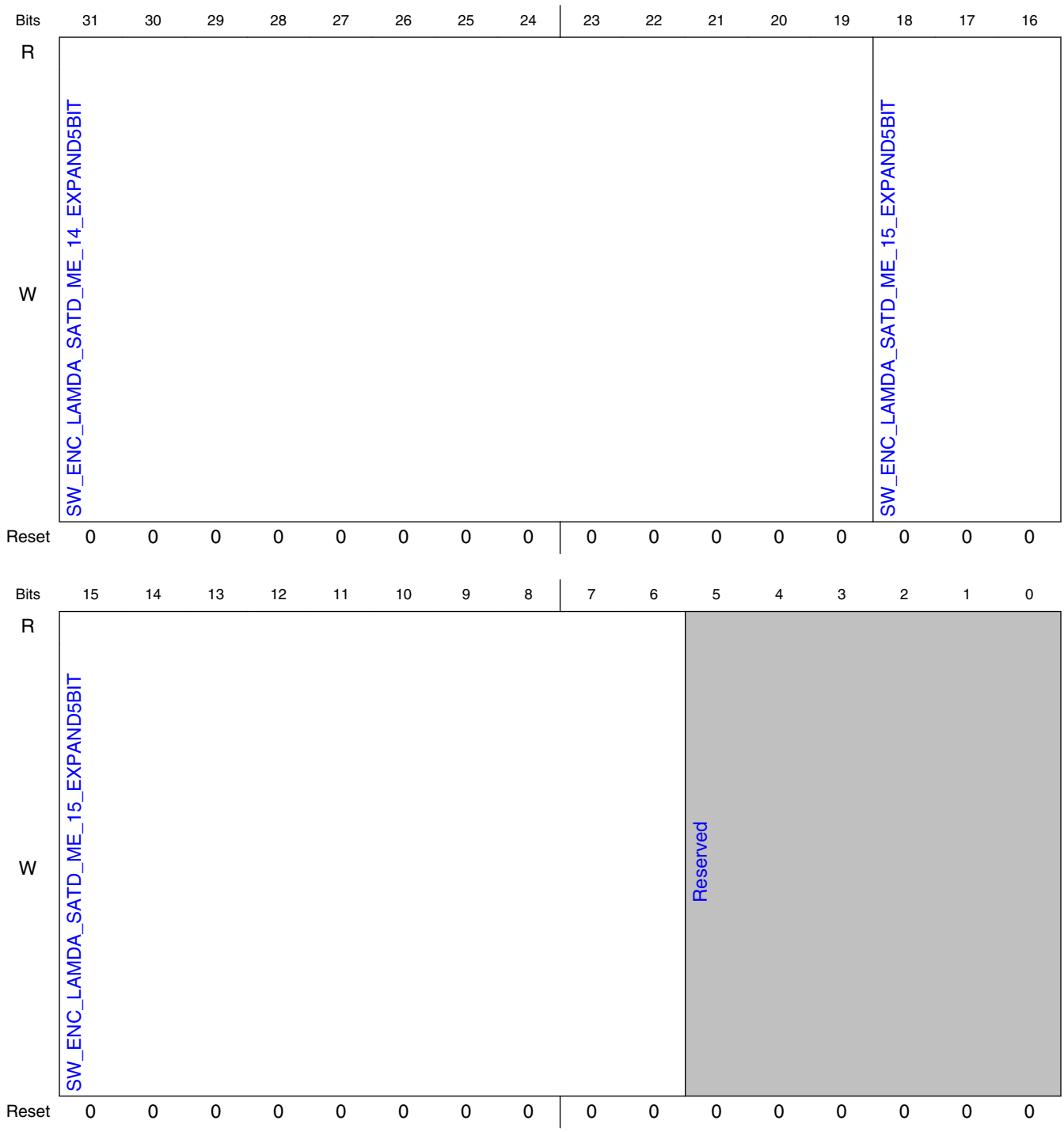
16.3.5.2.64.1 Offset

Register	Offset
swreg78_H2V5	138h

16.3.5.2.64.2 Function

inter me SATD lambda config 7. For H2V5 or later version.

16.3.5.2.64.3 Diagram



16.3.5.2.64.4 Fields

Field	Function
31-19	lambda satd me 14

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_14_EXPAND5BIT	
18-6 SW_ENC_LAMDA_SATD_ME_15_EXPAND5BIT	lambda satd me 15
5-0 —	Reserved

16.3.5.2.65 inter prediction parameters4. For H2V1 only. (swreg79_H2V1)

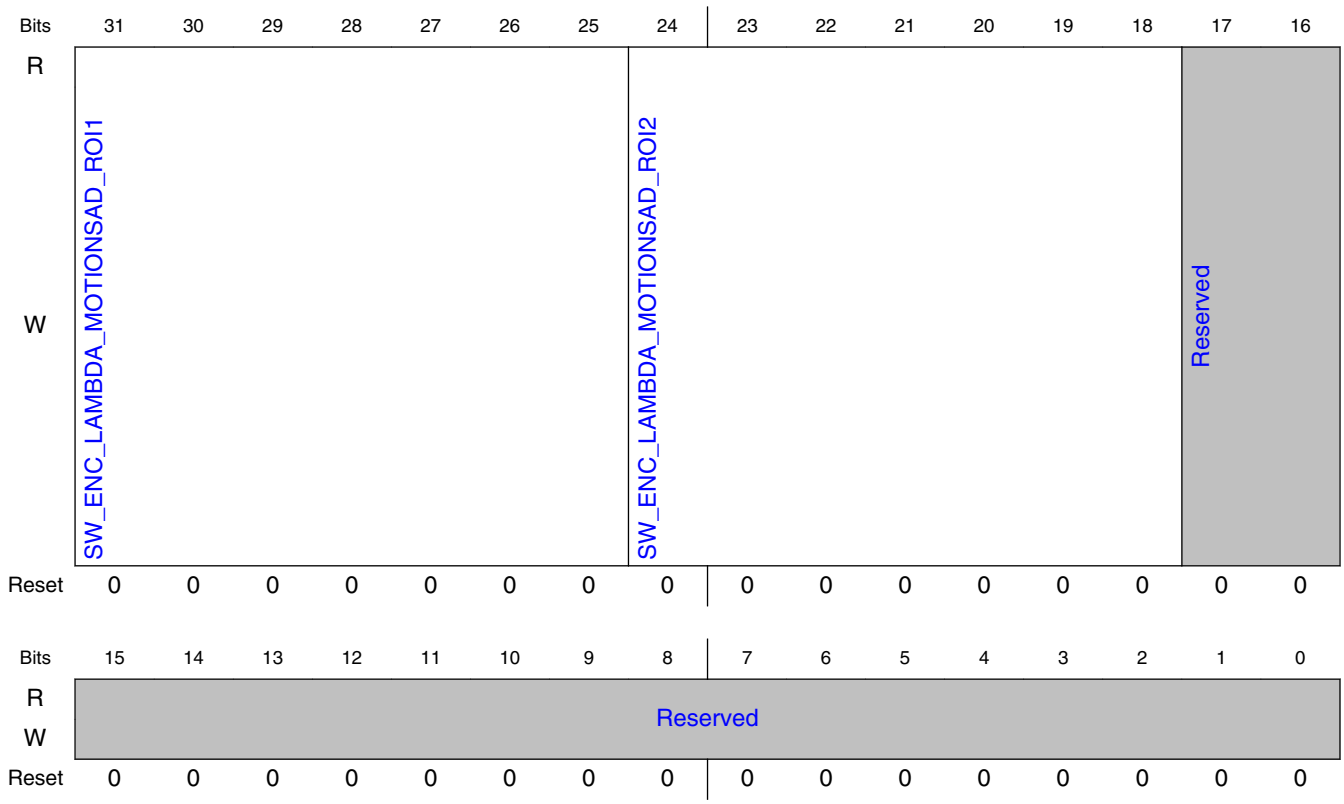
16.3.5.2.65.1 Offset

Register	Offset
swreg79_H2V1	13Ch

16.3.5.2.65.2 Function

inter prediction parameters4. For H2V1 only.

16.3.5.2.65.3 Diagram



16.3.5.2.65.4 Fields

Field	Function
31-25 SW_ENC_LAMBDA_MOTIONSAD_ROI1	roi1 lamda for motion SAD
24-18 SW_ENC_LAMBDA_MOTIONSAD_ROI2	roi2 lamda for motion SAD
17-0 —	Reserved

16.3.5.2.66 inter me SSE lambda config 4. For H2V2 or later version. (swreg79_H2V2)

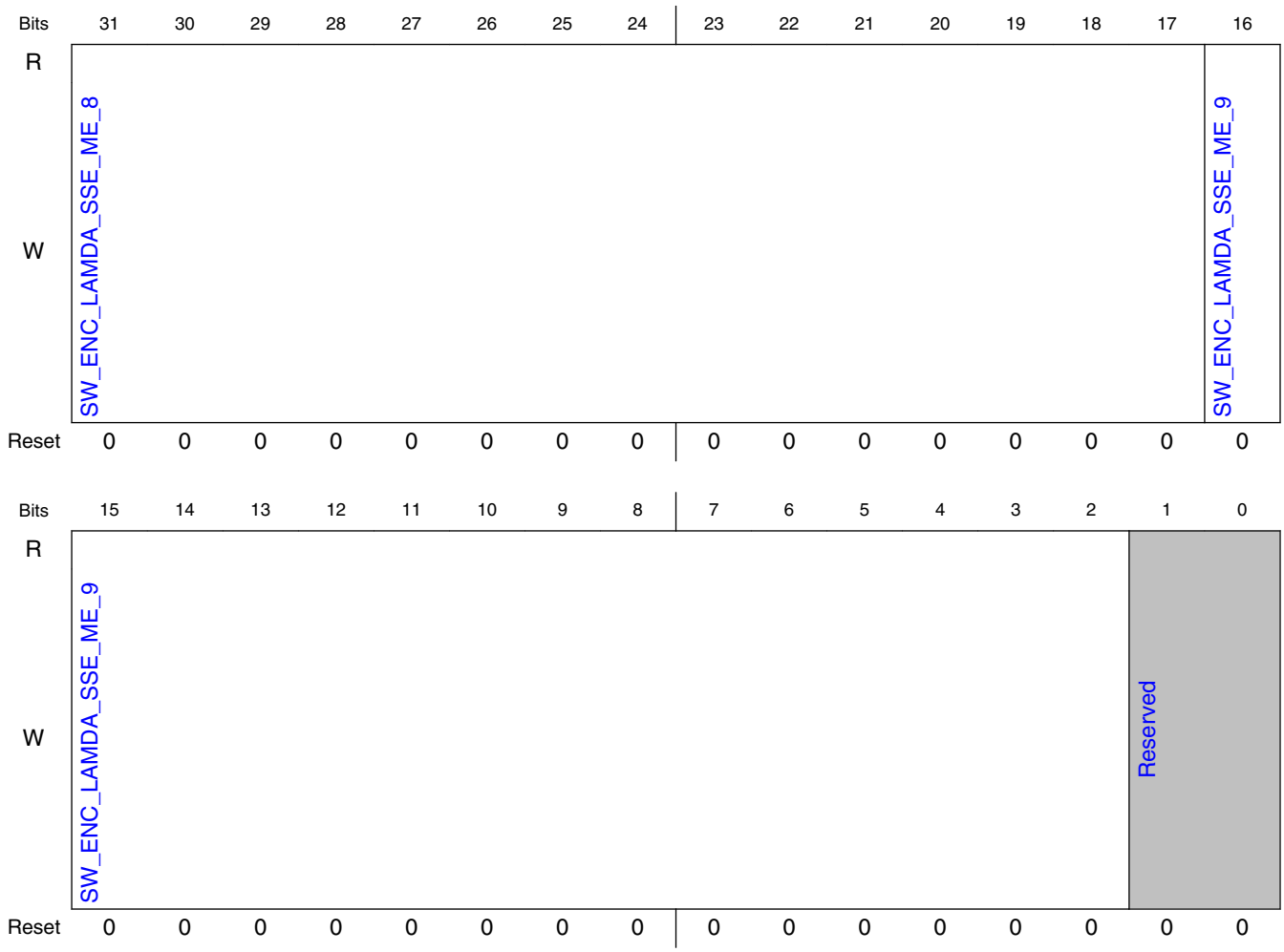
16.3.5.2.66.1 Offset

Register	Offset
swreg79_H2V2	13Ch

16.3.5.2.66.2 Function

inter me SSE lambda config 4. For H2V2 or later version.

16.3.5.2.66.3 Diagram



16.3.5.2.66.4 Fields

Field	Function
31-17	inter me SSE lambda 8

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SSE_ME_8	
16-2 SW_ENC_LAMDA_SSE_ME_9	inter me SSE lambda 9
1-0 —	Reserved

16.3.5.2.67 inter me SSE lambda config 0. For H2V5 or later version. (swreg79_H2V5)

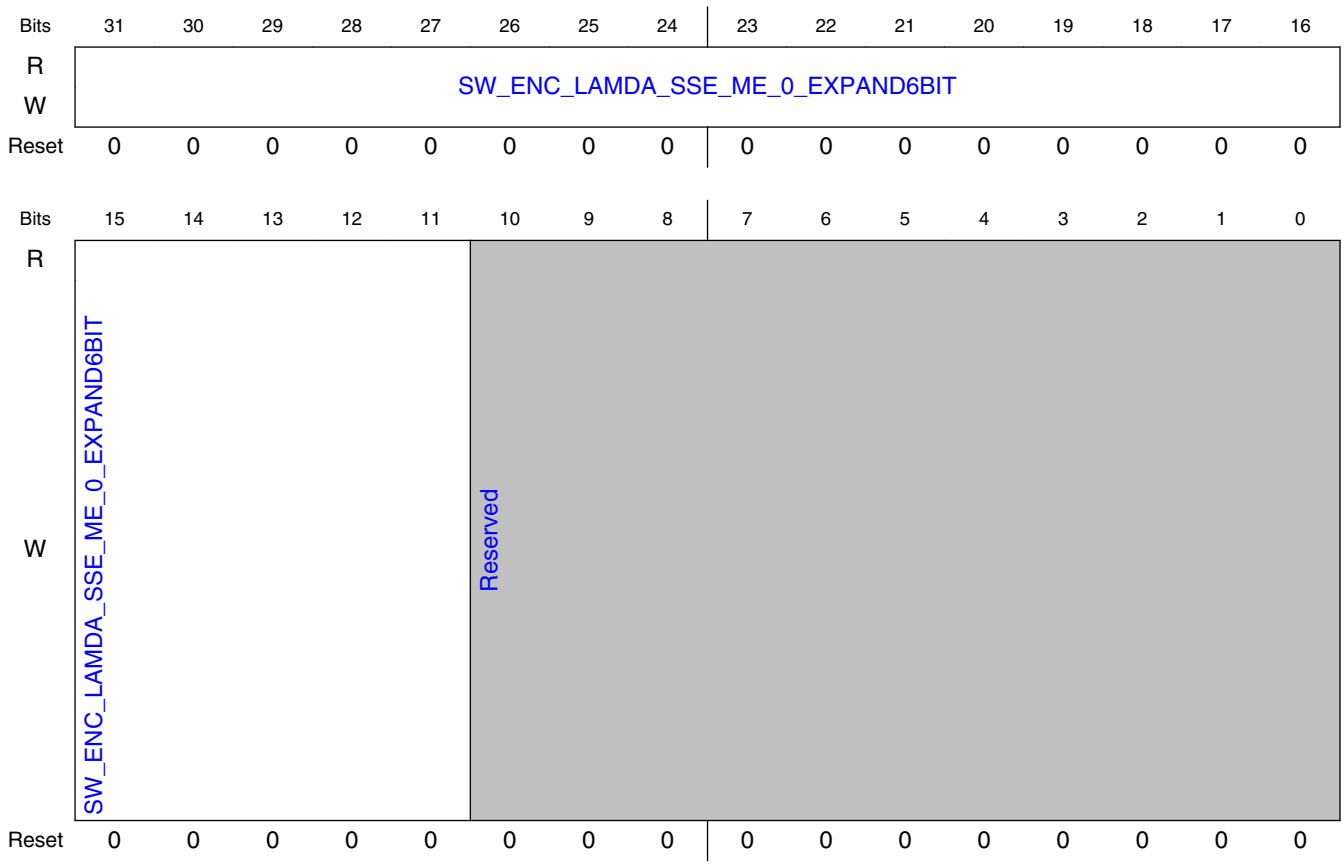
16.3.5.2.67.1 Offset

Register	Offset
swreg79_H2V5	13Ch

16.3.5.2.67.2 Function

inter me SSE lambda config 0. For H2V5 or later version.

16.3.5.2.67.3 Diagram



16.3.5.2.67.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_0_EXPAND6BIT	inter me SSE lambda 0
10-0 —	Reserved

16.3.5.2.68 hardware configuation 0 (swreg81)

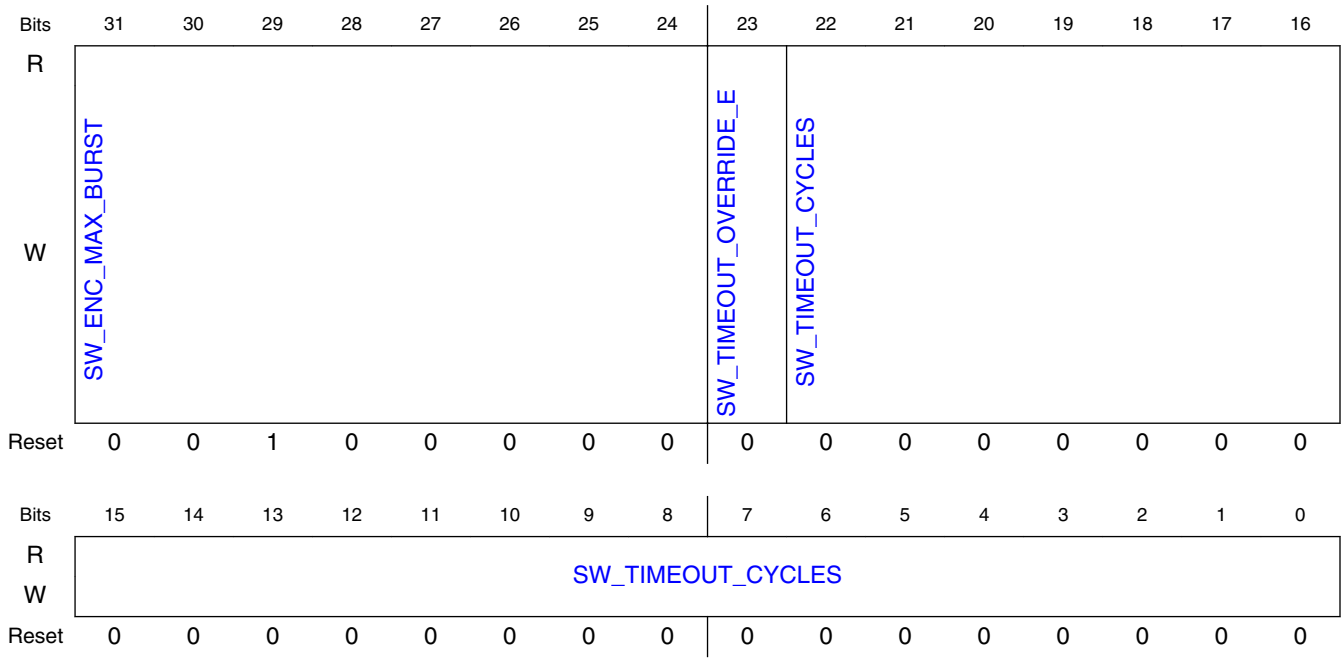
16.3.5.2.68.1 Offset

Register	Offset
swreg81	144h

16.3.5.2.68.2 Function

hardware configuation 0

16.3.5.2.68.3 Diagram



16.3.5.2.68.4 Fields

Field	Function
31-24 SW_ENC_MAX_BURST	for support AXI4.0 burst length is programmable Default value:0x20
23 SW_TIMEOUT_OVERRIDE_E	enable signal if timeout period is controlled by software Default value: 0x0
22-0 SW_TIMEOUT_CYCLES	timeout cycles number default value: 0x0

16.3.5.2.69 record hardware performance (swreg82)

16.3.5.2.69.1 Offset

Register	Offset
swreg82	148h

16.3.5.2.69.2 Function

record hardware performance

16.3.5.2.69.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_HW_PERFORMANCE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_HW_PERFORMANCE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.69.4 Fields

Field	Function
31-0 SW_ENC_HW_PERFORMANCE	record hardware performance(cycles) of current picture

16.3.5.2.70 reference picture reconstructed list1 luma0 (swreg83)

16.3.5.2.70.1 Offset

Register	Offset
swreg83	14Ch

16.3.5.2.70.2 Function

reference picture reconstructed list1 luma0

16.3.5.2.70.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REFPIC_RECON_L1_Y0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_REFPIC_RECON_L1_Y0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.70.4 Fields

Field	Function
31-0 SW_ENC_REFPIC_RECON_L1_Y0	reference picture reconstructed list1 luma0

16.3.5.2.71 reference picture reconstructed list1 chroma0 (swreg84)

16.3.5.2.71.1 Offset

Register	Offset
swreg84	150h

16.3.5.2.71.2 Function

reference picture reconstructed list1 chroma0

16.3.5.2.71.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REFPIC_RECON_L1_CHROMA0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_REFPIC_RECON_L1_CHROMA0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.71.4 Fields

Field	Function
31-0 SW_ENC_REFPIC_RECON_L1_CHROMA0	reference picture reconstructed list1 chroma0

16.3.5.2.72 reference pictures list1 config (swreg91)

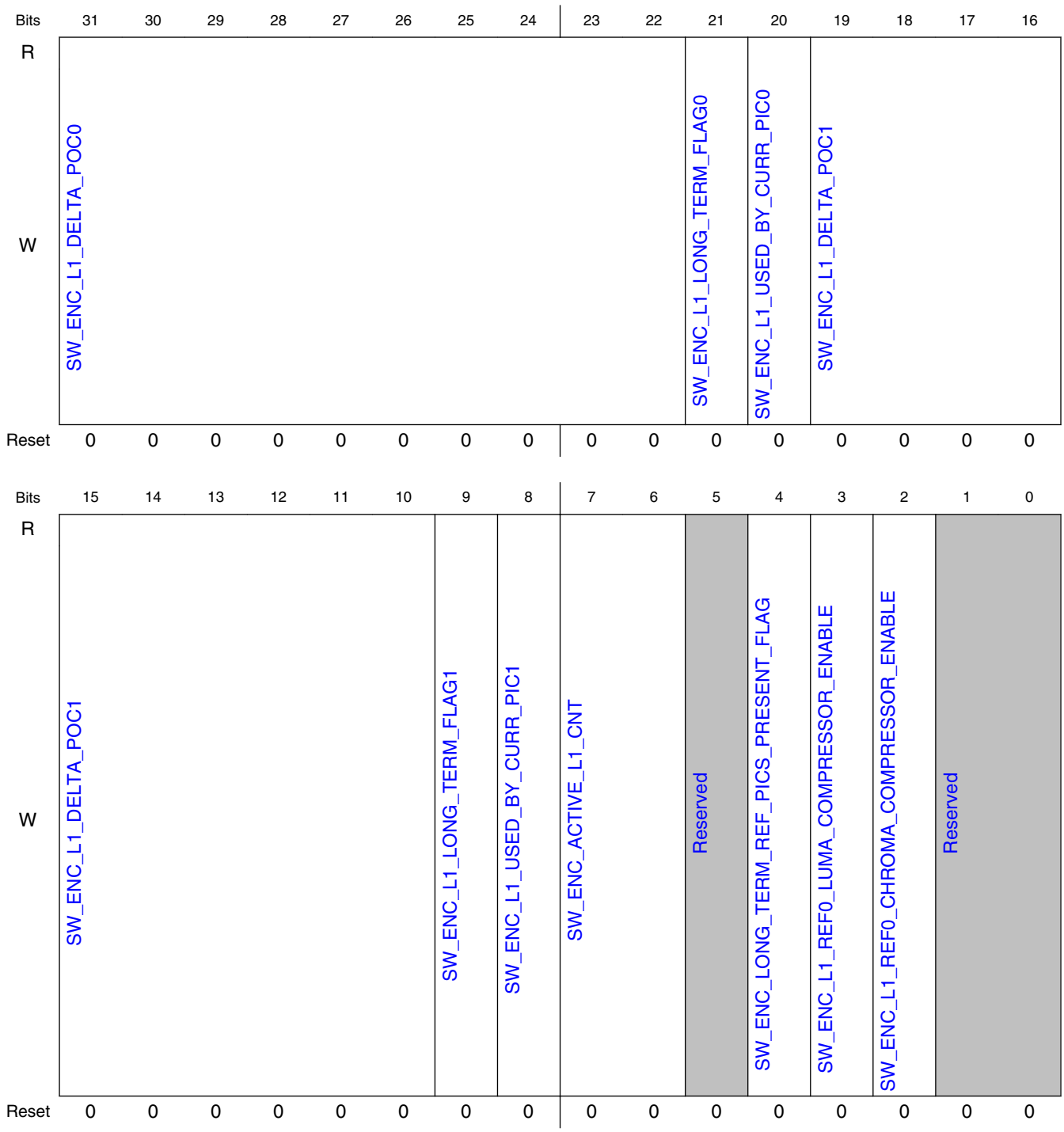
16.3.5.2.72.1 Offset

Register	Offset
swreg91	16Ch

16.3.5.2.72.2 Function

reference pictures list1 config

16.3.5.2.72.3 Diagram



16.3.5.2.72.4 Fields

Field	Function
31-22	delta poc of list1 pic0

Table continues on the next page...

Field	Function
SW_ENC_L1_DELTA_POC0	
21 SW_ENC_L1_LONG_TERM_FLAG0	list1 pic0 is long term
20 SW_ENC_L1_USED_BY_CURR_PIC0	list1 pic0 used by current
19-10 SW_ENC_L1_DELTA_POC1	delta poc of list1 pic1
9 SW_ENC_L1_LONG_TERM_FLAG1	list1 pic1 is long term
8 SW_ENC_L1_USED_BY_CURR_PIC1	list1 pic1 used by current
7-6 SW_ENC_ACTIVE_L1_CNT	active l0 count
5 —	Reserved
4 SW_ENC_LONG_TERM_REF_PICS_PRESENT_FLAG	long-term reference pictures may be used for inter prediction
3 SW_ENC_L1_REF0_LUMA_COMPRESSOR_ENABLE	list1 ref0 frame luma compressor enable flag 0b - disable 1b - enable.
2 SW_ENC_L1_REF0_CHROMA_COMPRESSOR_ENABLE	list1 ref0 frame chroma compressor enable flag 0b - disable 1b - enable.
1-0 —	Reserved

16.3.5.2.73 reference picture reconstructed list1 4n 0 (swreg92)

16.3.5.2.73.1 Offset

Register	Offset
swreg92	170h

16.3.5.2.73.2 Function

reference picture reconstructed list1 4n 0

16.3.5.2.73.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REFPIC_RECON_L1_4N0_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_REFPIC_RECON_L1_4N0_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.73.4 Fields

Field	Function
31-0 SW_ENC_REFPIC_RECON_L1_4N0_BASE	reference picture reconstructed list1 4n 0

16.3.5.2.74 Base address for list 1 ref 0 luma compress table LSB. (swreg96)

16.3.5.2.74.1 Offset

Register	Offset
swreg96	180h

16.3.5.2.74.2 Function

Base address for list 1 ref 0 luma compress table LSB.

16.3.5.2.74.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_L1_REF0_LUMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_L1_REF0_LUMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.74.4 Fields

Field	Function
31-0 SW_ENC_L1_REF0_LUMA_COMPRESS_TABLE_BASE	Base address for list 1 ref 0 luma compress table LSB.

16.3.5.2.75 Base address for list 1 ref 0 Chroma compress table LSB. (swreg98)

16.3.5.2.75.1 Offset

Register	Offset
swreg98	188h

16.3.5.2.75.2 Function

Base address for list 1 ref 0 Chroma compress table LSB.

16.3.5.2.75.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_L1_REF0_CHROMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_L1_REF0_CHROMA_COMPRESS_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.75.4 Fields

Field	Function
31-0 SW_ENC_L1_REF0_CHROMA_COMPRESS_TABLE_BASE	Base address for list 1 ref 0 Chroma compress table LSB.

16.3.5.2.76 reference picture lists modification (swreg104)

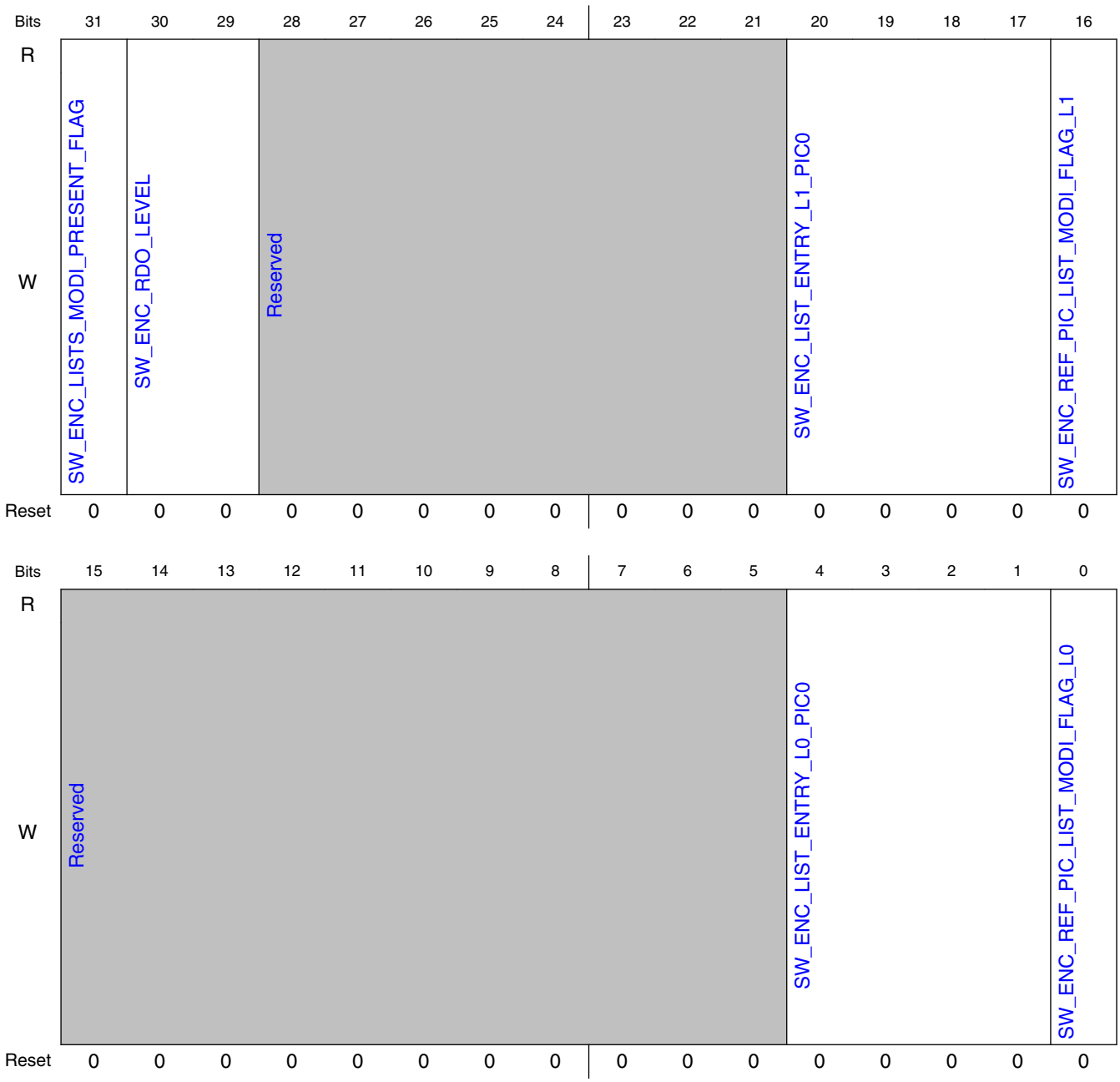
16.3.5.2.76.1 Offset

Register	Offset
swreg104	1A0h

16.3.5.2.76.2 Function

reference picture lists modification

16.3.5.2.76.3 Diagram



16.3.5.2.76.4 Fields

Field	Function
31 SW_ENC_LISTS_MODI_PRESENT_FLAG	lists_modification_present_flag from pps

Table continues on the next page...

Field	Function
30-29 SW_ENC_RDO_LEVEL	RDO level
28-21 —	Reserved
20-17 SW_ENC_LIST_ENTRY_L1_PIC0	list1 picture0 index
16 SW_ENC_REF_PIC_LIST_MODI_FLAG_L1	reference picture list1 modification flag
15-5 —	Reserved
4-1 SW_ENC_LIST_ENTRY_L0_PIC0	list0 picture0 index
0 SW_ENC_REF_PIC_LIST_MODI_FLAG_L0	reference picture list0 modification flag

16.3.5.2.77 Min picture size (swreg106)

16.3.5.2.77.1 Offset

Register	Offset
swreg106	1A8h

16.3.5.2.77.2 Function

Min picture size

16.3.5.2.77.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MINPICSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MINPICSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.77.4 Fields

Field	Function
31-0 SW_ENC_MINPICSIZE	Allowed minimum picture size for CTB rate control.

16.3.5.2.78 Max picture size (swreg107)

16.3.5.2.78.1 Offset

Register	Offset
swreg107	1ACh

16.3.5.2.78.2 Function

Max picture size

16.3.5.2.78.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MAXPICSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MAXPICSIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.78.4 Fields

Field	Function
31-0 SW_ENC_MAX PICSIZE	Allowed minimum picture size for CTB rate control.

16.3.5.2.79 Qp delta map (swreg109)

16.3.5.2.79.1 Offset

Register	Offset
swreg109	1B4h

16.3.5.2.79.2 Function

Qp delta map

16.3.5.2.79.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROIMAPDELTAQPADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROIMAPDELTAQPADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.79.4 Fields

Field	Function
31-0 SW_ENC_ROIMAPDELTAQPADDR	Qp delta map

16.3.5.2.80 adaptive GOP configuration1 (swreg111)

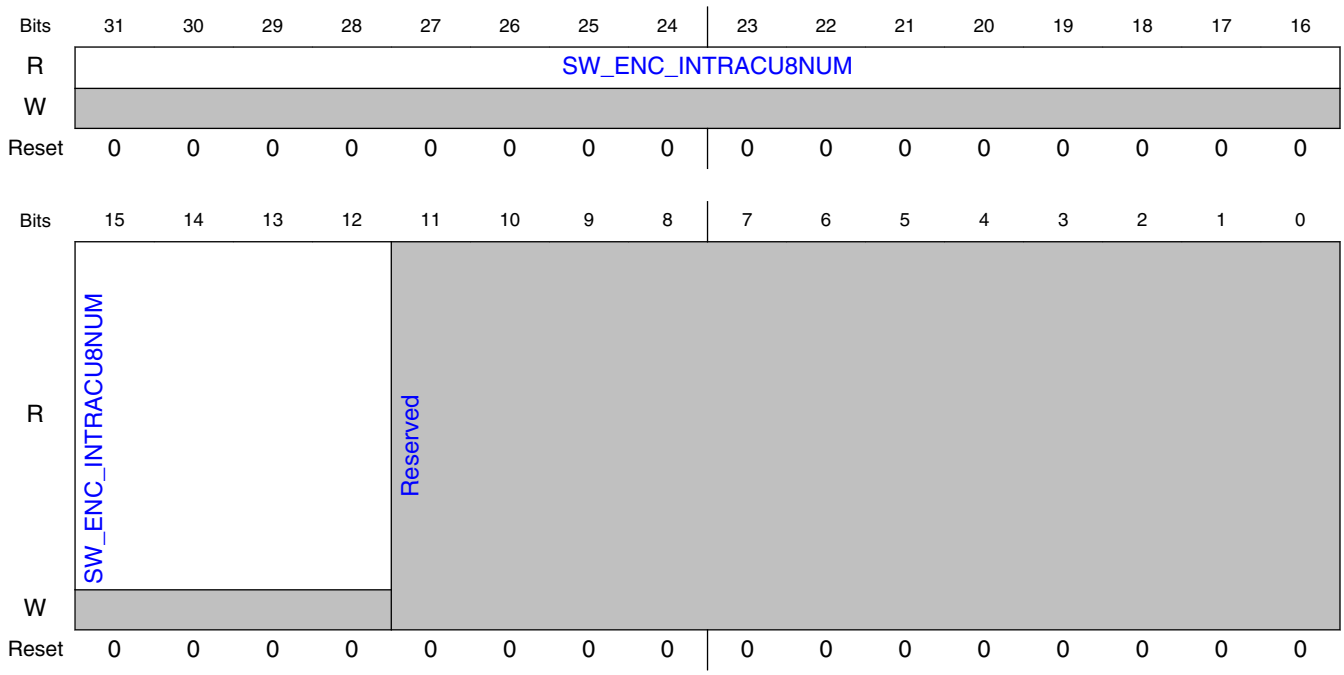
16.3.5.2.80.1 Offset

Register	Offset
swreg111	1BCh

16.3.5.2.80.2 Function

adaptive GOP configuration1

16.3.5.2.80.3 Diagram



16.3.5.2.80.4 Fields

Field	Function
31-12 SW_ENC_INTRACU8NUM	The number of block8x8 with type INTRA.
11-0 —	Reserved

16.3.5.2.81 adaptive GOP configuration2 (swreg112)

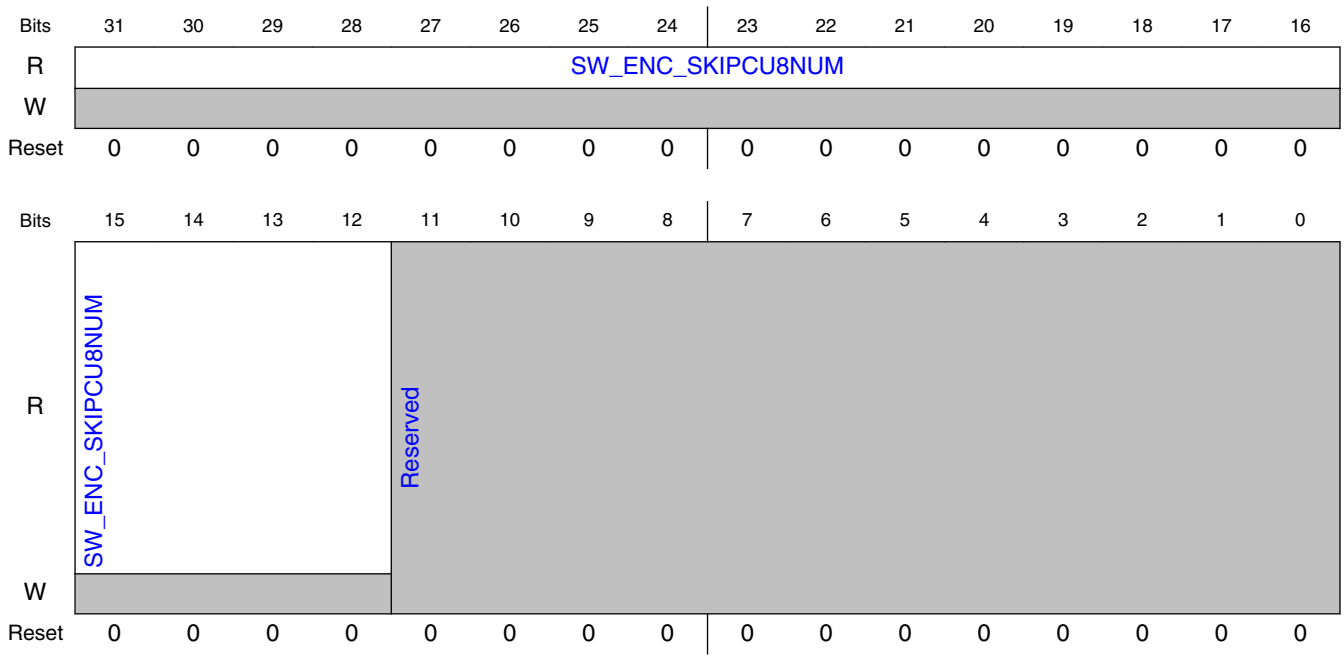
16.3.5.2.81.1 Offset

Register	Offset
swreg112	1C0h

16.3.5.2.81.2 Function

adaptive GOP configuration2

16.3.5.2.81.3 Diagram



16.3.5.2.81.4 Fields

Field	Function
31-12 SW_ENC_SKIPCU8NUM	The number of block8x8 with type SKIP.
11-0 —	Reserved

16.3.5.2.82 adaptive GOP configuration3 (swreg113)

16.3.5.2.82.1 Offset

Register	Offset
swreg113	1C4h

16.3.5.2.82.2 Function

adaptive GOP configuration3

16.3.5.2.82.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_PBFRAME4NRDCOST															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_PBFRAME4NRDCOST															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.82.4 Fields

Field	Function
31-0 SW_ENC_PBFRAME4NRDCOST	PBFrame4NRdCost

16.3.5.2.83 min/max lcu bits number of last picture (swreg119)

16.3.5.2.83.1 Offset

Register	Offset
swreg119	1DCh

16.3.5.2.83.2 Function

min/max lcu bits number of last picture

16.3.5.2.83.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_CTBBITSMIN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_CTBBITSMAX															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.83.4 Fields

Field	Function
31-16 SW_ENC_CTB BITSMIN	minimum lcu bits number of last picture
15-0 SW_ENC_CTB BITSMAX	maximum lcu bits number of last picture

16.3.5.2.84 total bits number of all lcus of last picture not including slice header bits (swreg120)

16.3.5.2.84.1 Offset

Register	Offset
swreg120	1E0h

16.3.5.2.84.2 Function

total bits number of all lcus of last picture not including slice header bits

16.3.5.2.84.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_TOTALLCUBITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_TOTALLCUBITS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.84.4 Fields

Field	Function
31-0 SW_ENC_TOTALLCUBITS	total bits number of all lcus of last picture not including slice header bits

16.3.5.2.85 inter me SSE lambda config 5 (swreg122)

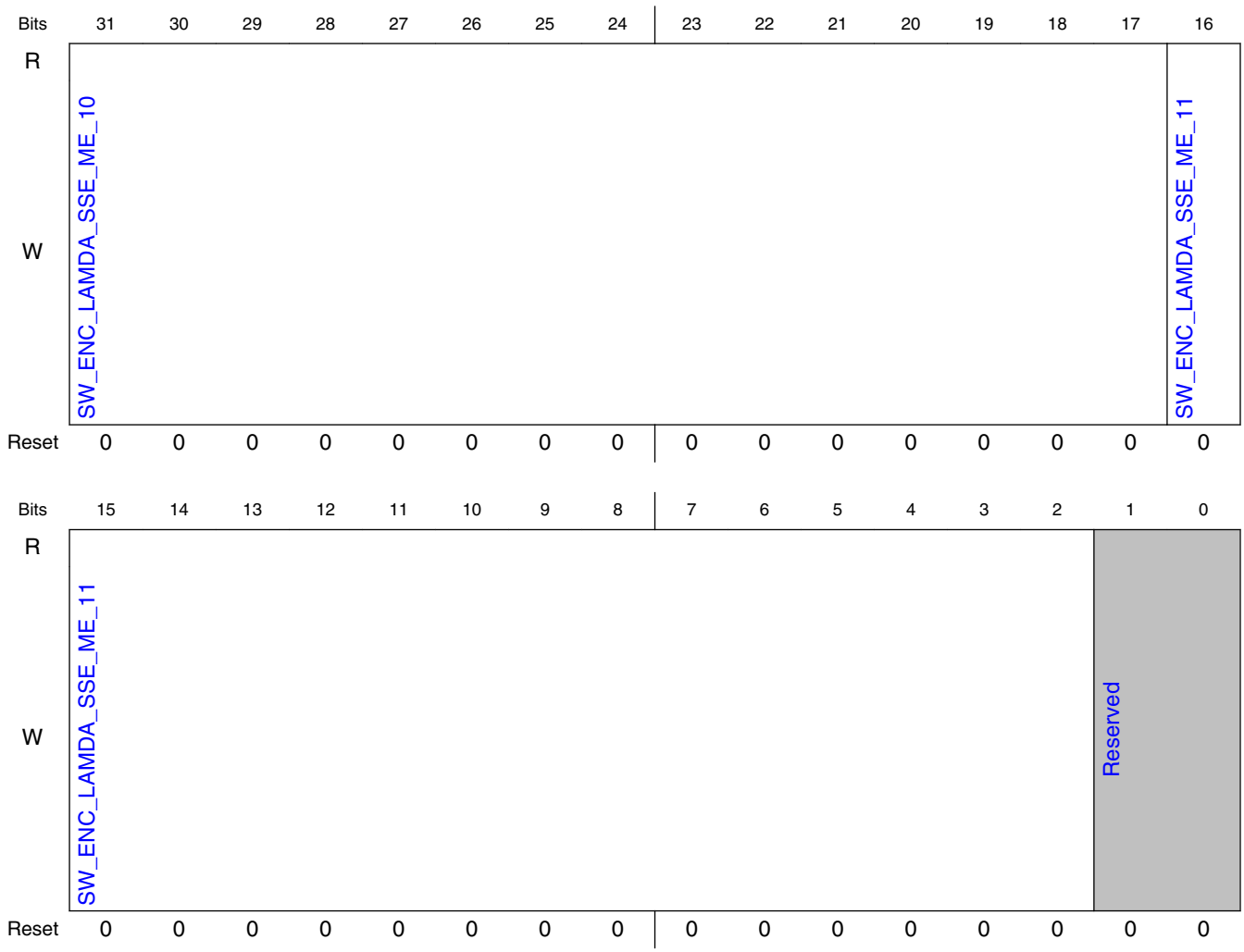
16.3.5.2.85.1 Offset

Register	Offset
swreg122	1E8h

16.3.5.2.85.2 Function

inter me SSE lambda config 5

16.3.5.2.85.3 Diagram



16.3.5.2.85.4 Fields

Field	Function
31-17 SW_ENC_LAMDA_SSE_ME_10	inter me SSE lambda 10
16-2 SW_ENC_LAMDA_SSE_ME_11	inter me SSE lambda 11
1-0 —	Reserved

16.3.5.2.86 inter me SSE lambda config 1. For H2V5 or later version. (swreg122_H2V5)

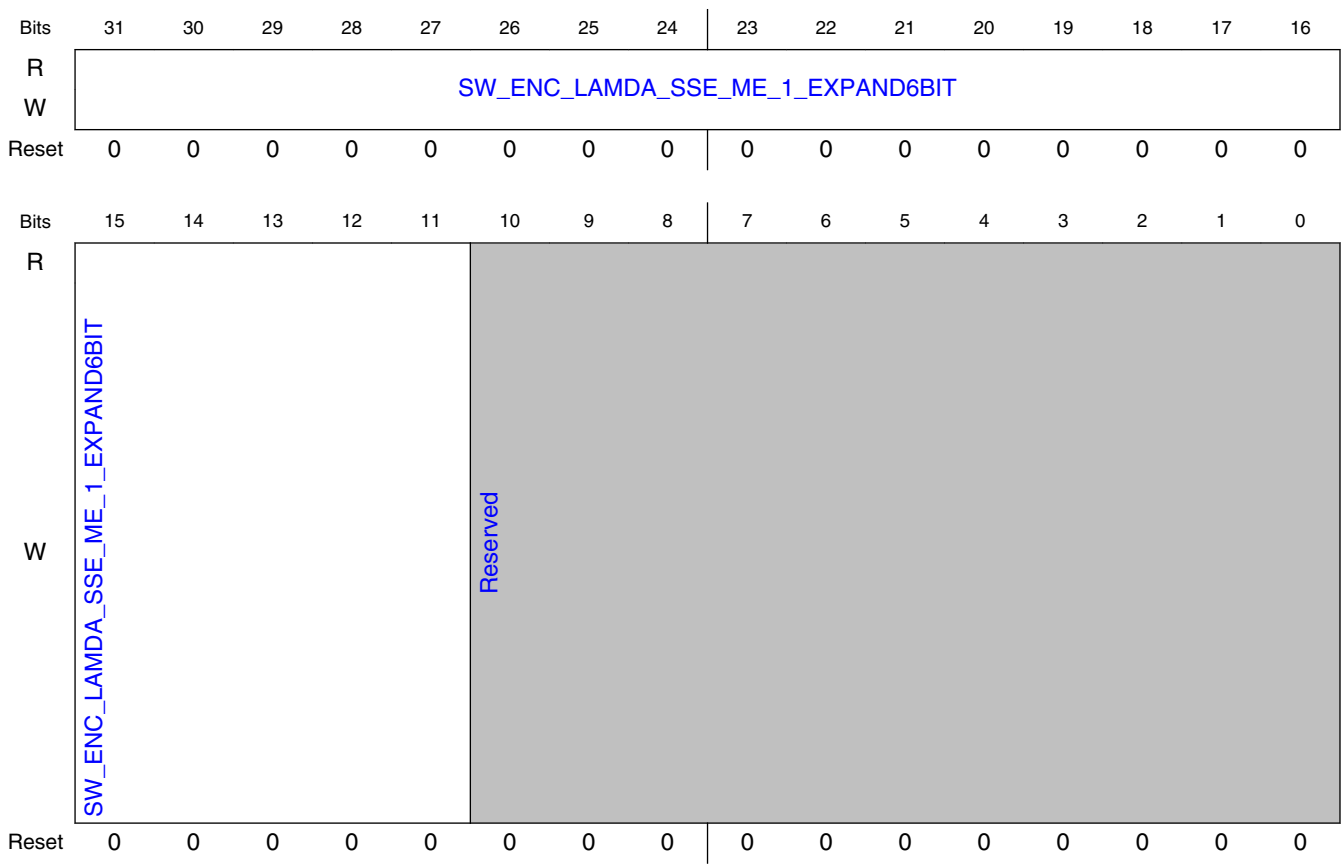
16.3.5.2.86.1 Offset

Register	Offset
swreg122_H2V5	1E8h

16.3.5.2.86.2 Function

inter me SSE lambda config 1. For H2V5 or later version.

16.3.5.2.86.3 Diagram



16.3.5.2.86.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_1_EXPAND6BIT	inter me SSE lambda 1
10-0 —	Reserved

16.3.5.2.87 inter me SSE lambda config 6 (swreg123)

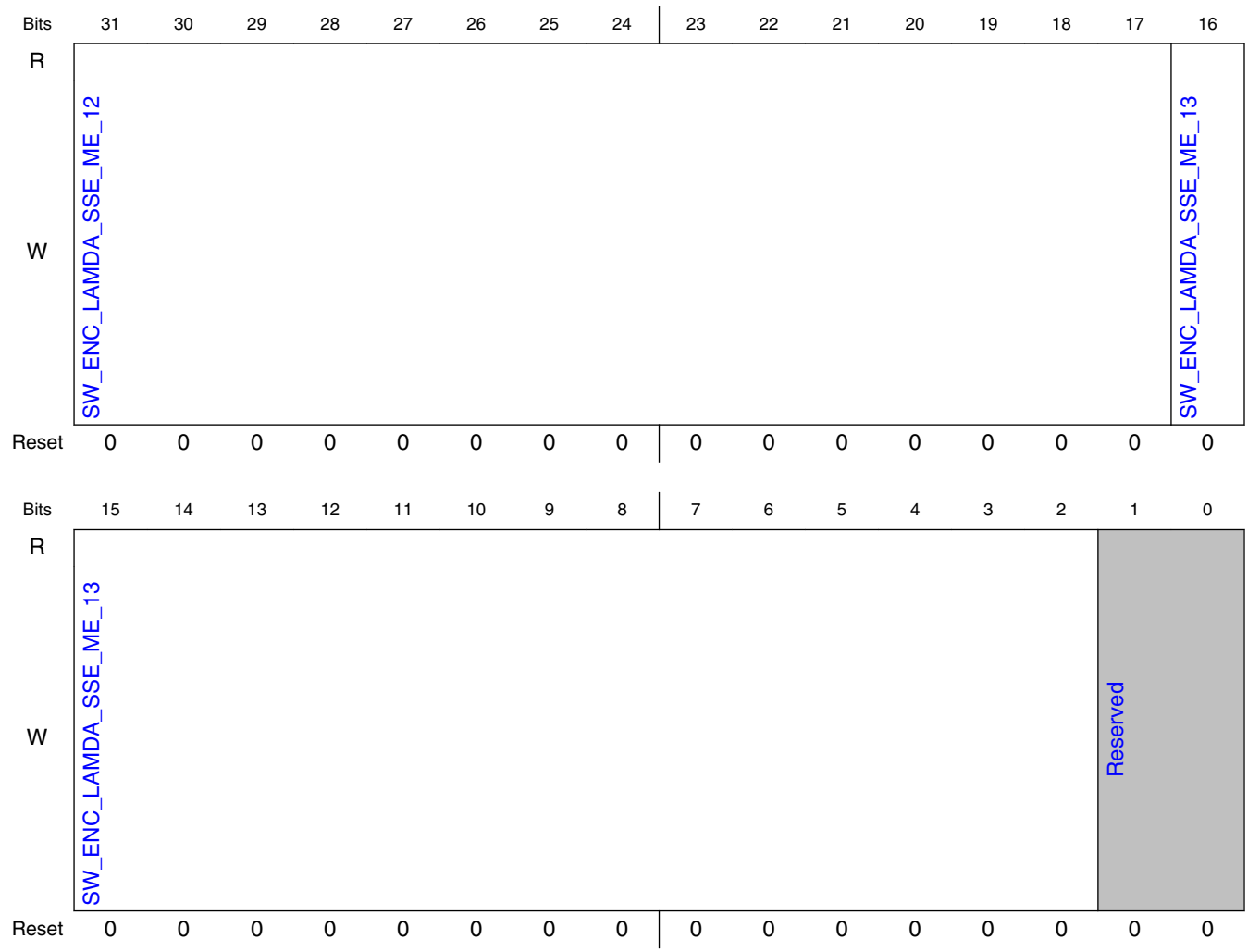
16.3.5.2.87.1 Offset

Register	Offset
swreg123	1ECh

16.3.5.2.87.2 Function

inter me SSE lambda config 6

16.3.5.2.87.3 Diagram



16.3.5.2.87.4 Fields

Field	Function
31-17 SW_ENC_LAMDA_SSE_ME_12	inter me SSE lambda 12
16-2 SW_ENC_LAMDA_SSE_ME_13	inter me SSE lambda 13
1-0 —	Reserved

16.3.5.2.88 inter me SSE lambda config 2. For H2V5 or later version. (swreg123_H2V5)

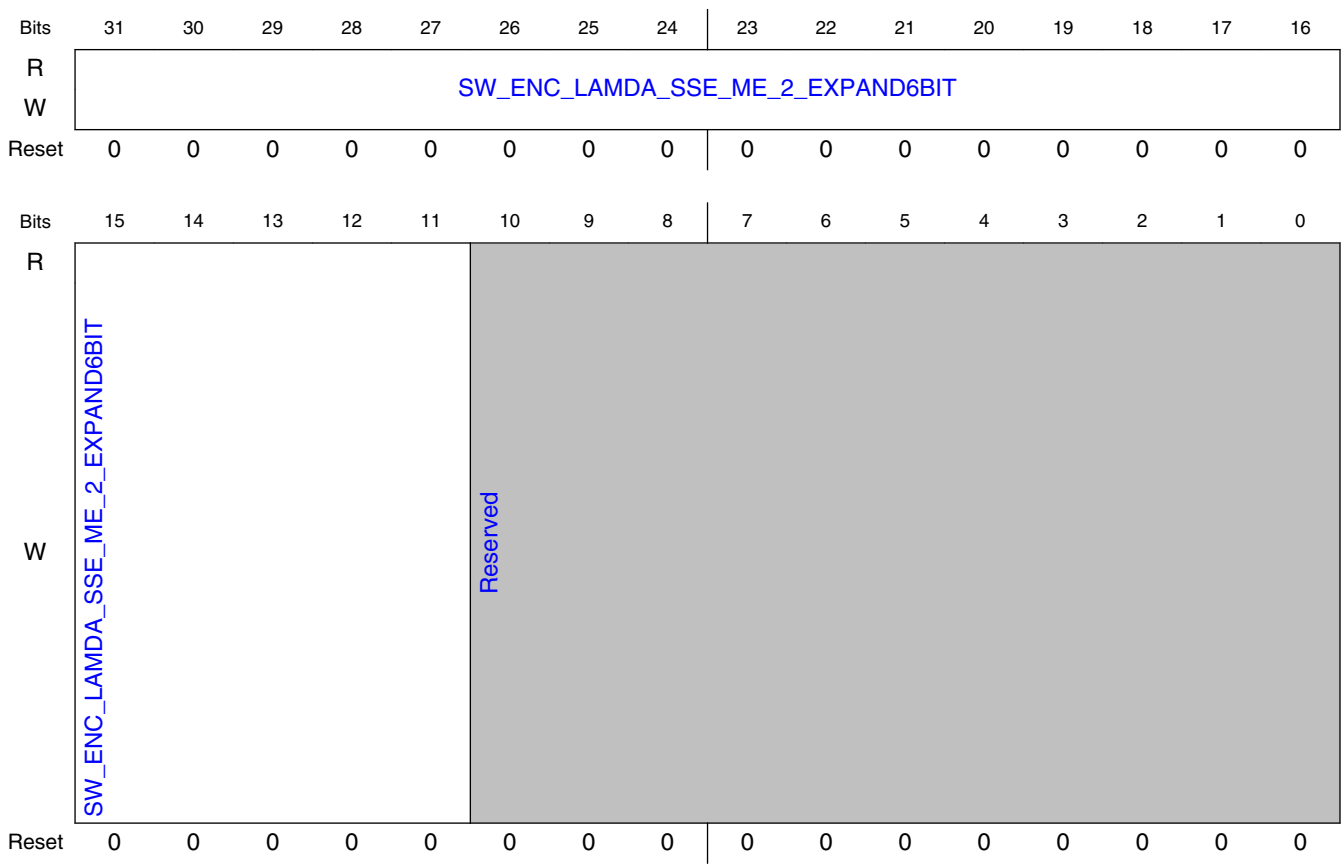
16.3.5.2.88.1 Offset

Register	Offset
swreg123_H2V5	1ECh

16.3.5.2.88.2 Function

inter me SSE lambda config 2. For H2V5 or later version.

16.3.5.2.88.3 Diagram



16.3.5.2.88.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_2_EXPAND6BIT	inter me SSE lambda 2
10-0 —	Reserved

16.3.5.2.89 inter me SSE lambda config 7 (swreg124)

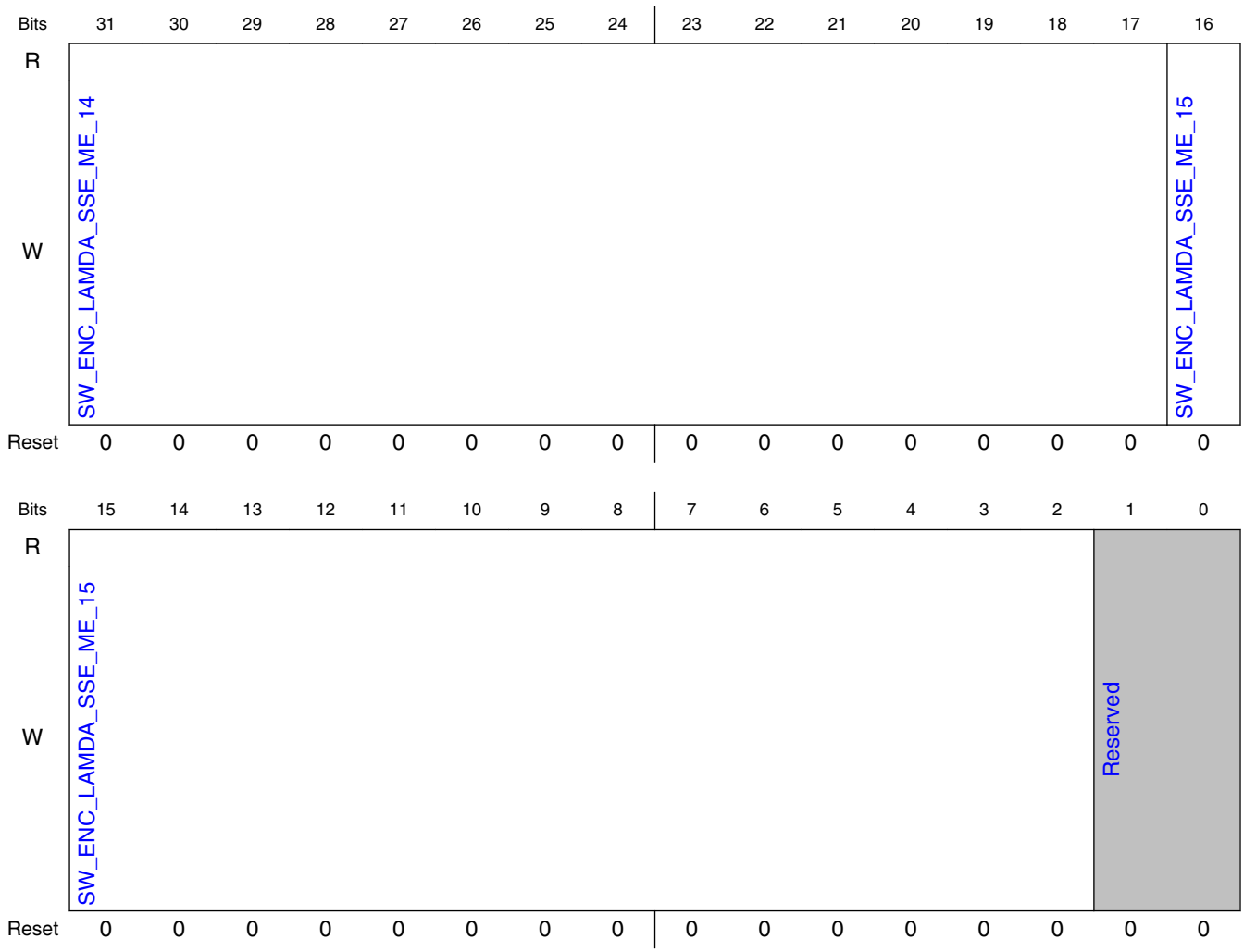
16.3.5.2.89.1 Offset

Register	Offset
swreg124	1F0h

16.3.5.2.89.2 Function

inter me SSE lambda config 7

16.3.5.2.89.3 Diagram



16.3.5.2.89.4 Fields

Field	Function
31-17 SW_ENC_LAMDA_SSE_ME_14	inter me SSE lambda 14
16-2 SW_ENC_LAMDA_SSE_ME_15	inter me SSE lambda 15
1-0 —	Reserved

16.3.5.2.90 inter me SSE lambda config 3. For H2V5 or later version. (swreg124_H2V5)

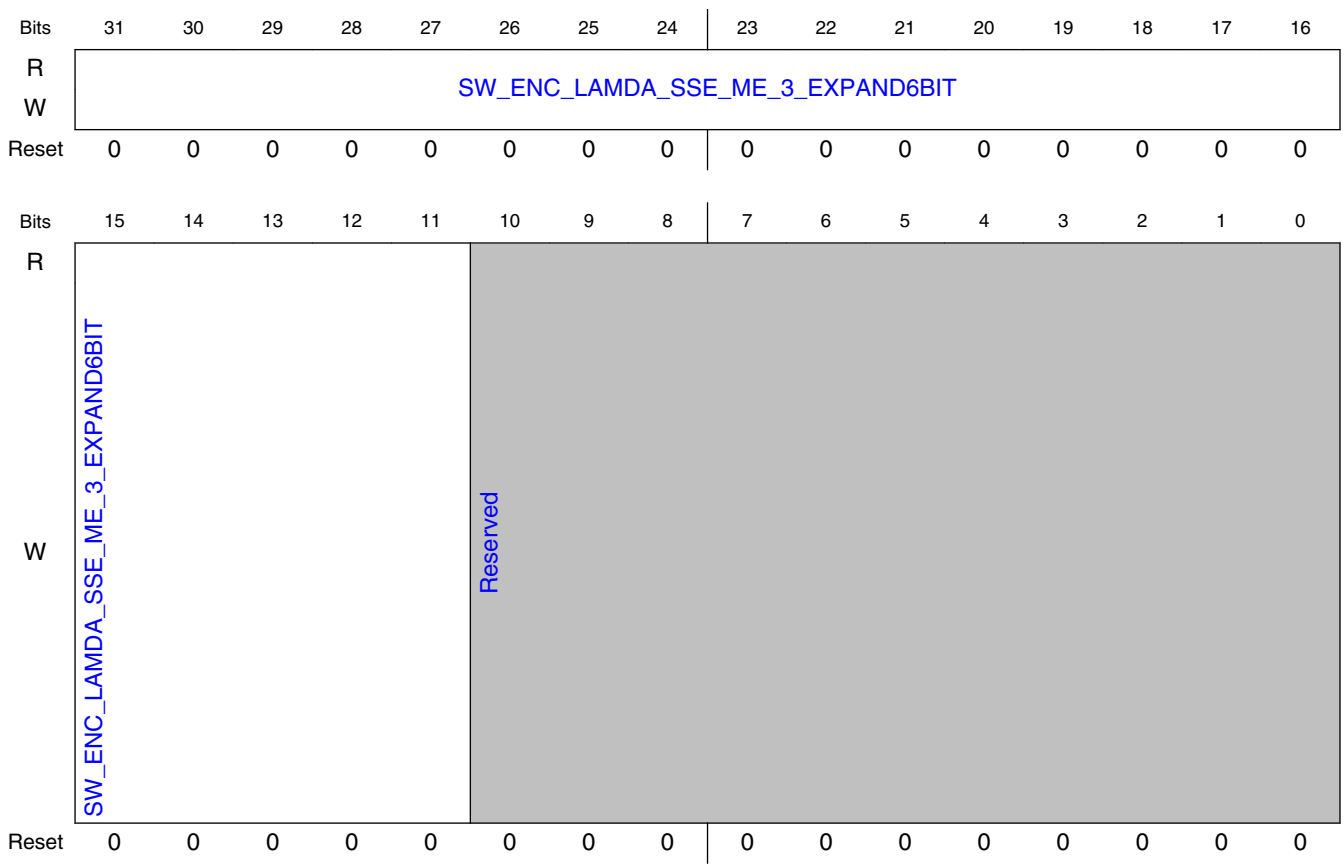
16.3.5.2.90.1 Offset

Register	Offset
swreg124_H2V5	1F0h

16.3.5.2.90.2 Function

inter me SSE lambda config 3. For H2V5 or later version.

16.3.5.2.90.3 Diagram



16.3.5.2.90.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_3_EXPAND6BIT	inter me SSE lambda 3
10-0 —	Reserved

16.3.5.2.91 intra SATD lambda config 0 (swreg125)

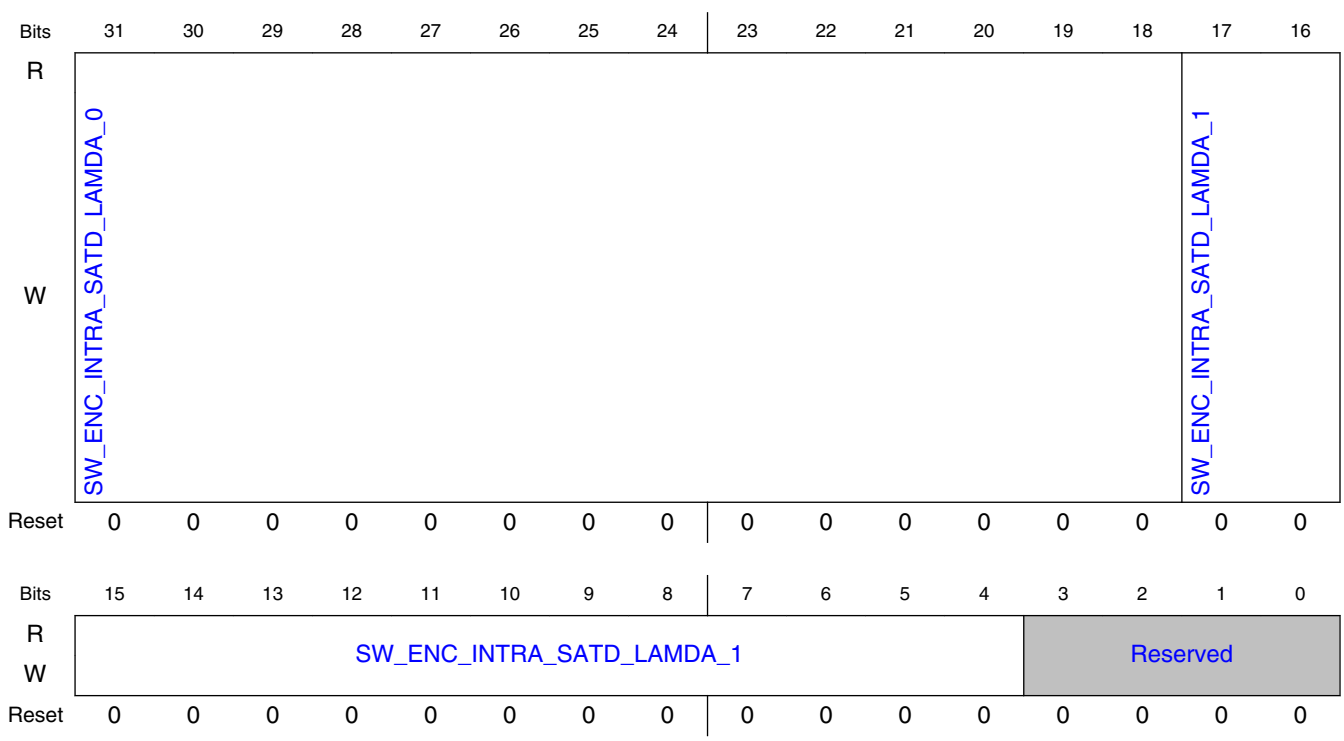
16.3.5.2.91.1 Offset

Register	Offset
swreg125	1F4h

16.3.5.2.91.2 Function

intra SATD lambda config 0

16.3.5.2.91.3 Diagram



16.3.5.2.91.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_0	intra SATD lambda 0
17-4 SW_ENC_INTRA_SATD_LAMDA_1	intra SATD lambda 1
3-0 —	Reserved

16.3.5.2.92 intra SATD lambda config 1 (swreg126)

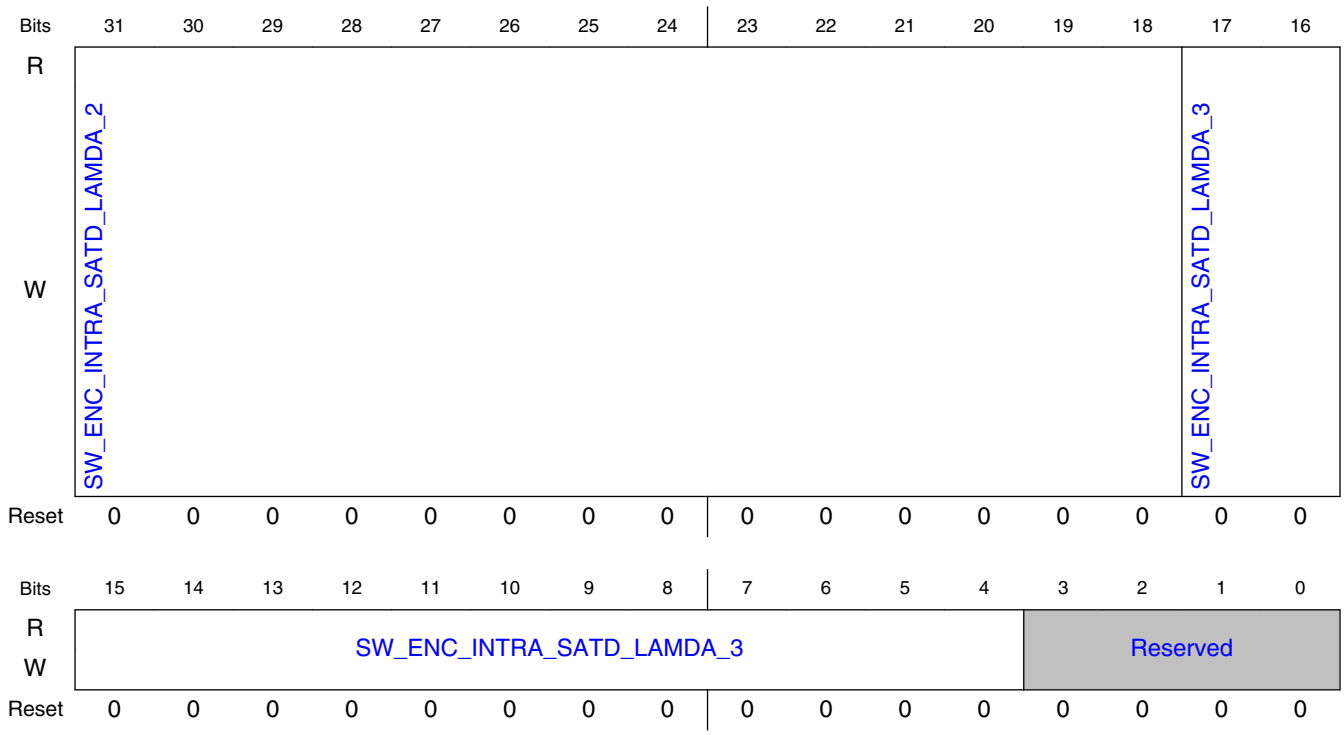
16.3.5.2.92.1 Offset

Register	Offset
swreg126	1F8h

16.3.5.2.92.2 Function

intra SATD lambda config 1

16.3.5.2.92.3 Diagram



16.3.5.2.92.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_2	intra SATD lambda 2
17-4 SW_ENC_INTRA_SATD_LAMDA_3	intra SATD lambda 3
3-0 —	Reserved

16.3.5.2.93 intra SATD lambda config 2 (swreg127)

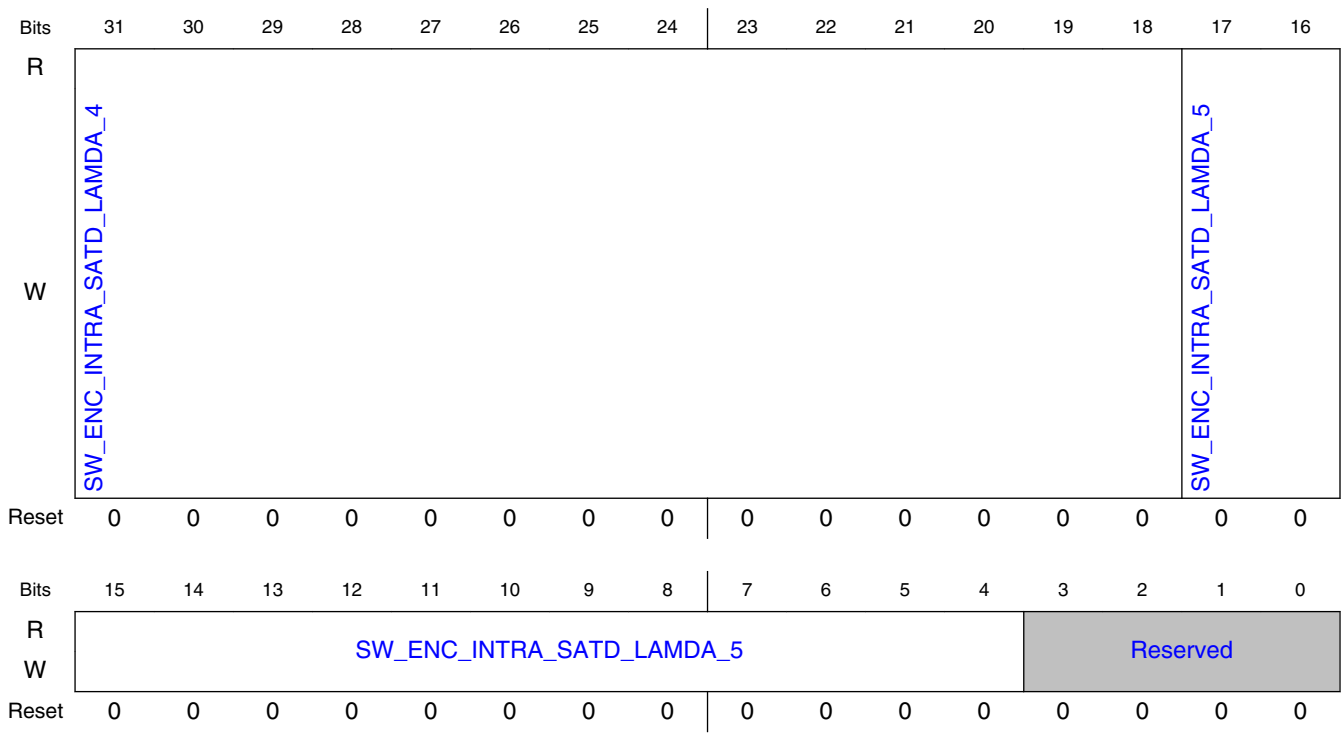
16.3.5.2.93.1 Offset

Register	Offset
swreg127	1FCh

16.3.5.2.93.2 Function

intra SATD lambda config 2

16.3.5.2.93.3 Diagram



16.3.5.2.93.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_4	intra SATD lambda 4
17-4 SW_ENC_INTRA_SATD_LAMDA_5	intra SATD lambda 5
3-0 —	Reserved

16.3.5.2.94 intra SATD lambda config 3 (swreg128)

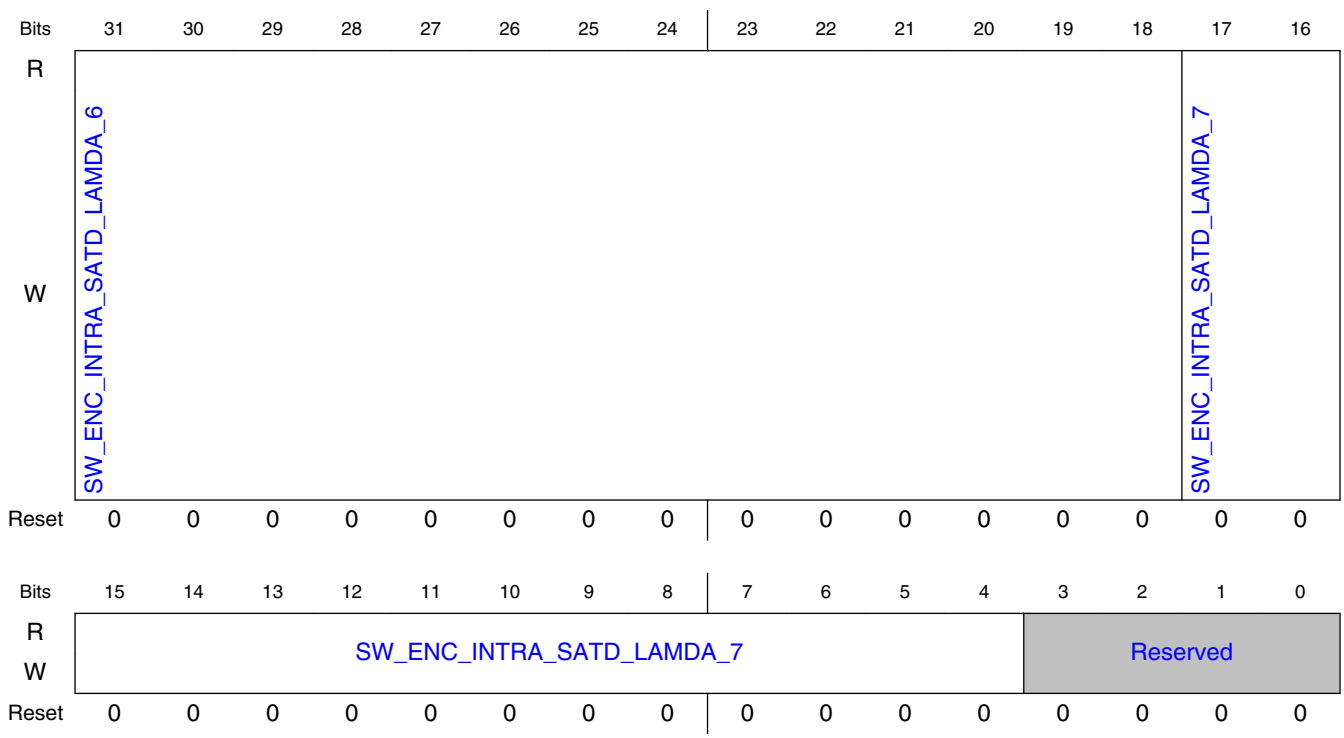
16.3.5.2.94.1 Offset

Register	Offset
swreg128	200h

16.3.5.2.94.2 Function

intra SATD lambda config 3

16.3.5.2.94.3 Diagram



16.3.5.2.94.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_6	intra SATD lambda 6
17-4	intra SATD lambda 7

Table continues on the next page...

Field	Function
SW_ENC_INTRA_SATD_LAMDA_7	
3-0 —	Reserved

16.3.5.2.95 intra SATD lambda config 4 (swreg129)

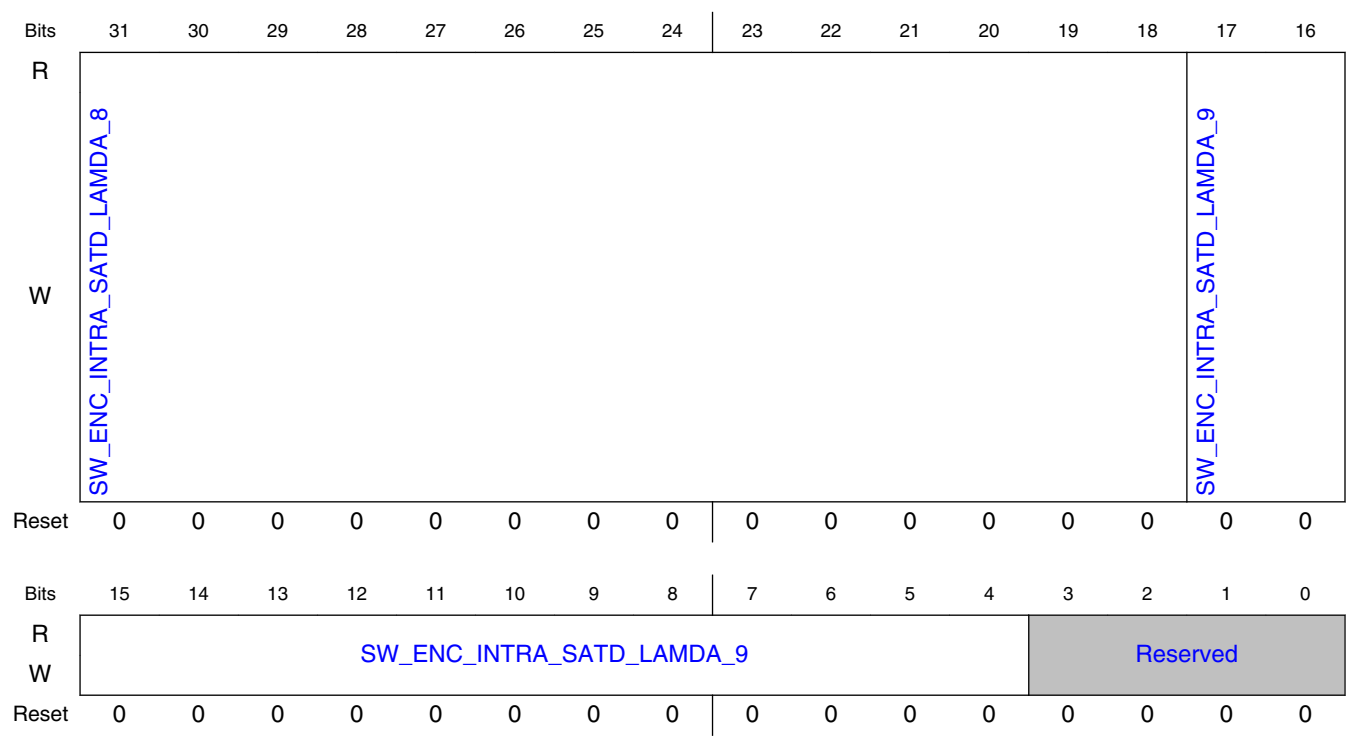
16.3.5.2.95.1 Offset

Register	Offset
swreg129	204h

16.3.5.2.95.2 Function

intra SATD lambda config 4

16.3.5.2.95.3 Diagram



16.3.5.2.95.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_8	intra SATD lambda 8
17-4 SW_ENC_INTRA_SATD_LAMDA_9	intra SATD lambda 9
3-0 —	Reserved

16.3.5.2.96 intra SATD lambda config 5 (swreg130)

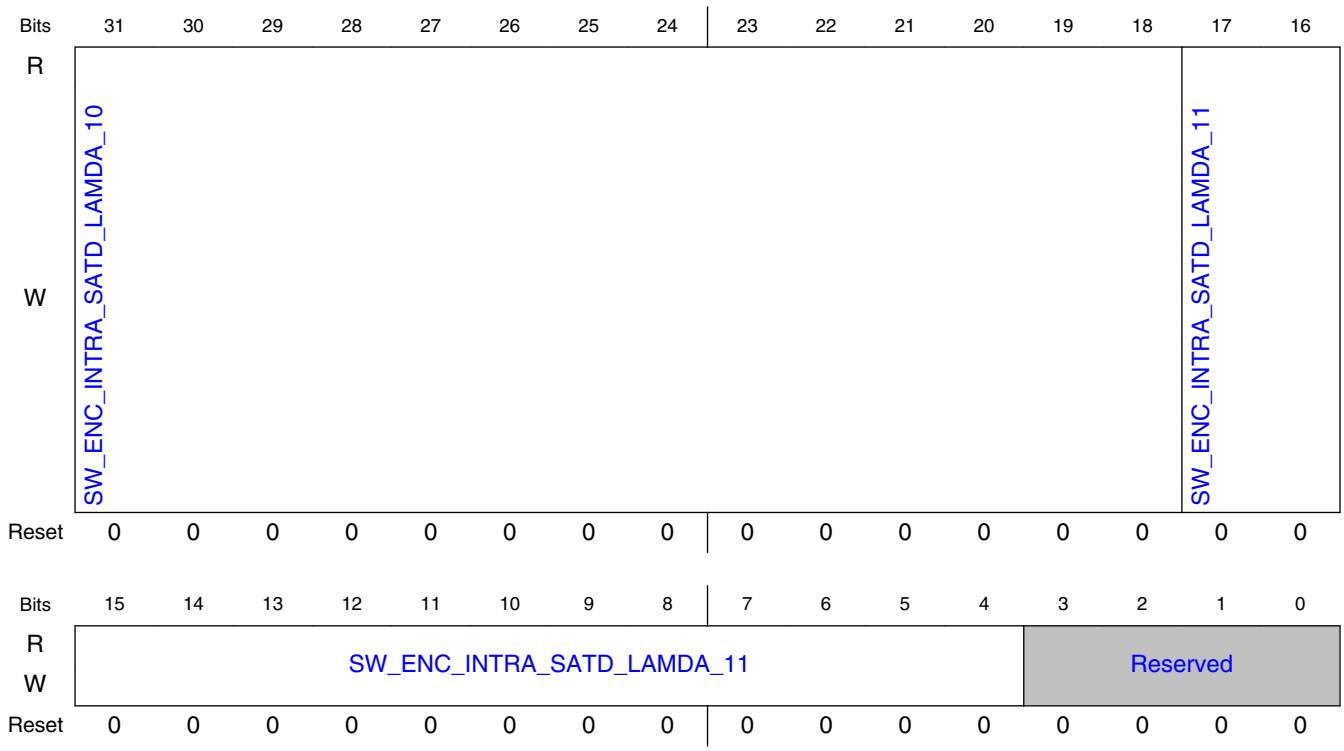
16.3.5.2.96.1 Offset

Register	Offset
swreg130	208h

16.3.5.2.96.2 Function

intra SATD lambda config 5

16.3.5.2.96.3 Diagram



16.3.5.2.96.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_10	intra SATD lambda 10
17-4 SW_ENC_INTRA_SATD_LAMDA_11	intra SATD lambda 11
3-0 —	Reserved

16.3.5.2.97 intra SATD lambda config 6 (swreg131)

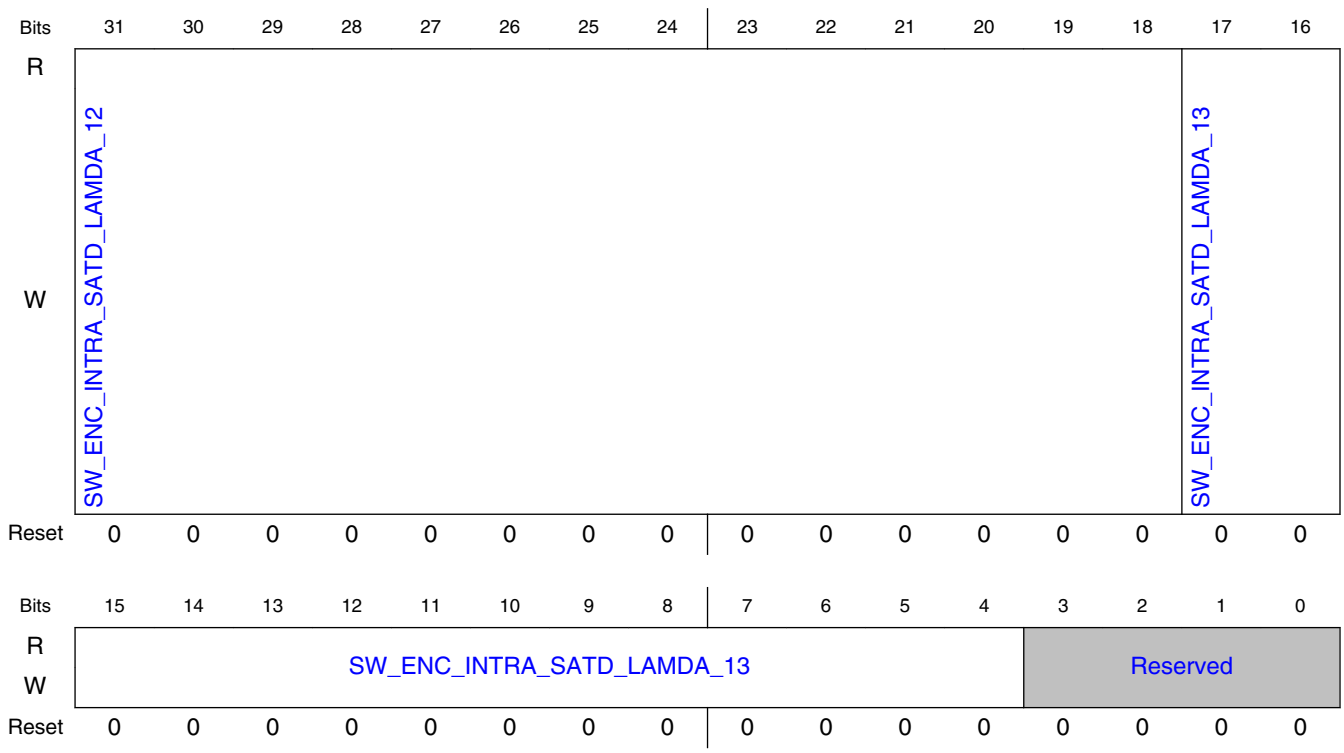
16.3.5.2.97.1 Offset

Register	Offset
swreg131	20Ch

16.3.5.2.97.2 Function

intra SATD lambda config 6

16.3.5.2.97.3 Diagram



16.3.5.2.97.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_12	intra SATD lambda 12
17-4 SW_ENC_INTRA_SATD_LAMDA_13	intra SATD lambda 13
3-0 —	Reserved

16.3.5.2.98 intra SATD lambda config 7 (swreg132)

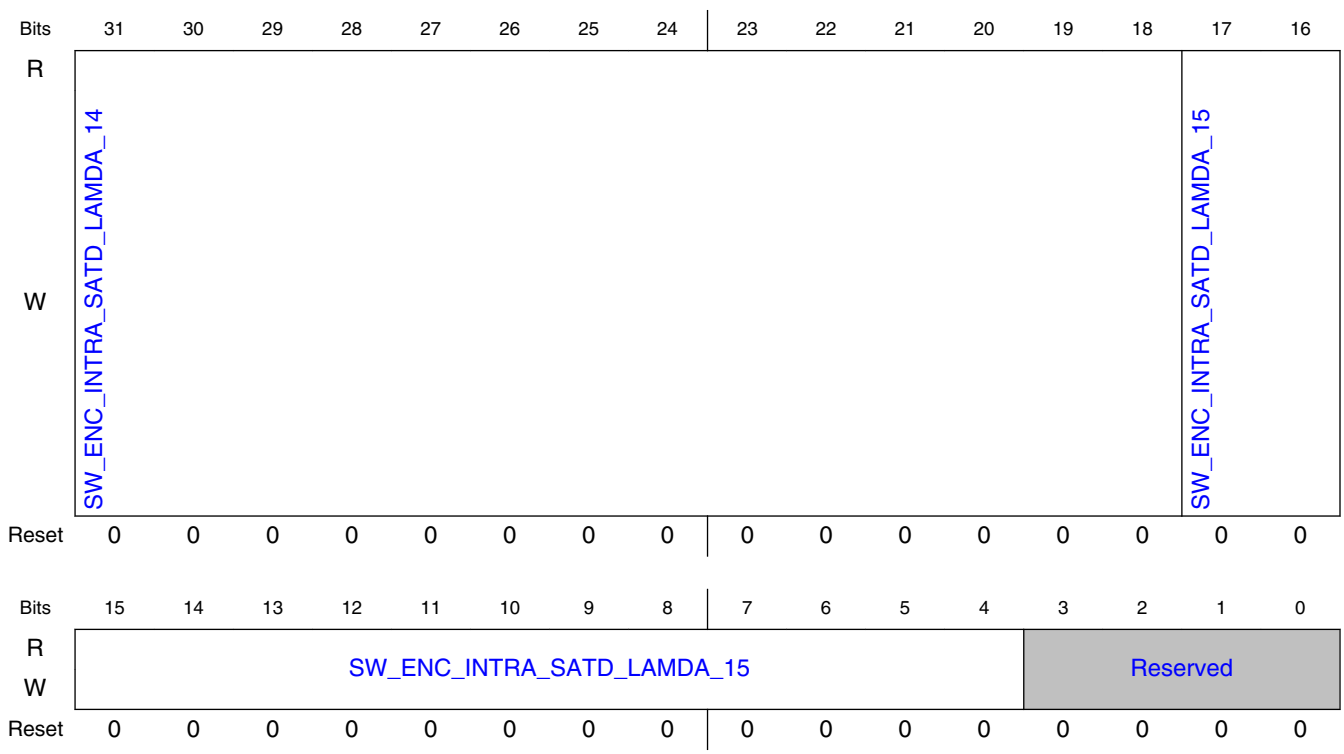
16.3.5.2.98.1 Offset

Register	Offset
swreg132	210h

16.3.5.2.98.2 Function

intra SATD lambda config 7

16.3.5.2.98.3 Diagram



16.3.5.2.98.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_14	intra SATD lambda 14
17-4	intra SATD lambda 15

Table continues on the next page...

Field	Function
SW_ENC_INTRA_SATD_LAMDA_15	
3-0 —	Reserved

16.3.5.2.99 SSE devide 256 (swreg133)

16.3.5.2.99.1 Offset

Register	Offset
swreg133	214h

16.3.5.2.99.2 Function

SSE devide 256

16.3.5.2.99.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_SSE_DIV_256															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_SSE_DIV_256															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.99.4 Fields

Field	Function
31-0 SW_ENC_SSE_DIV_256	sse divide 256

16.3.5.2.100 noise reduction (swreg134)

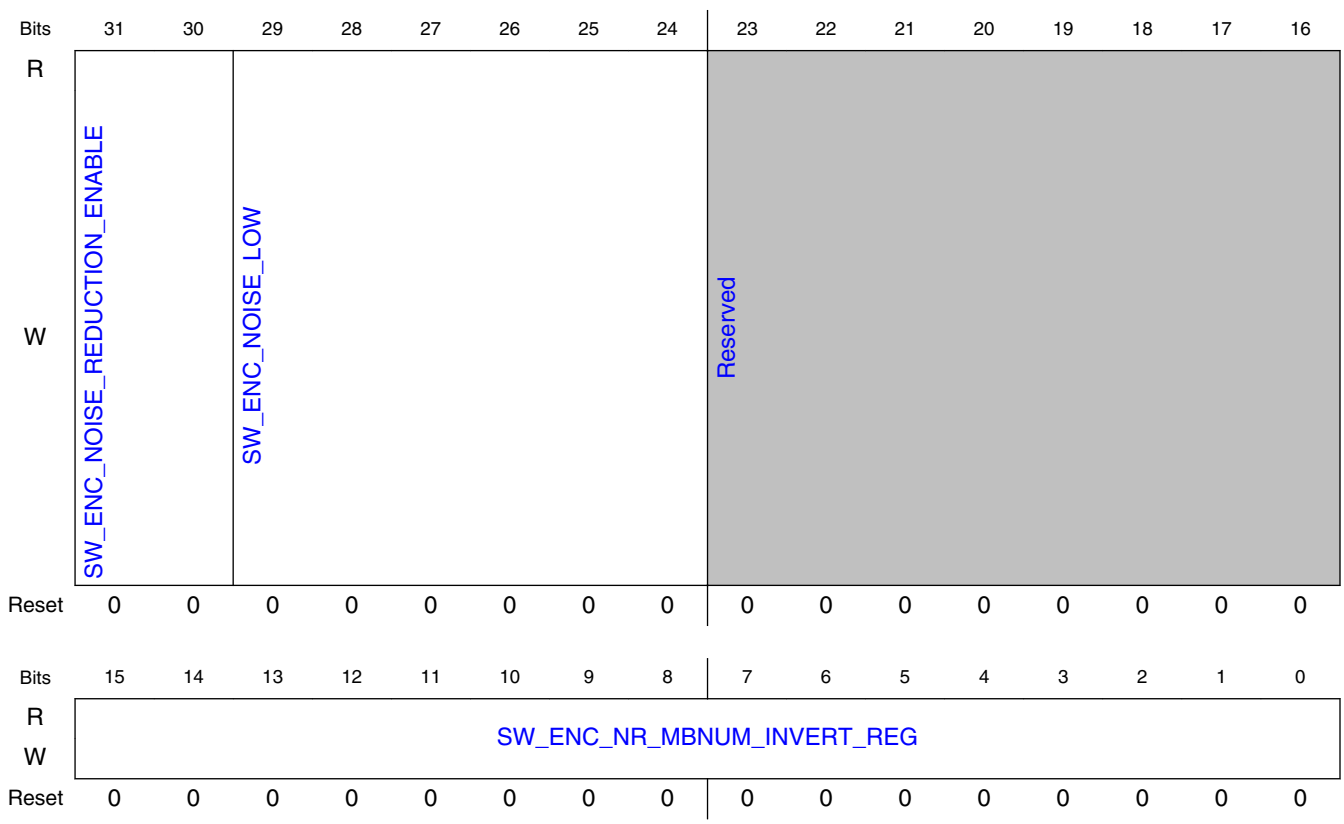
16.3.5.2.100.1 Offset

Register	Offset
swreg134	218h

16.3.5.2.100.2 Function

noise reduction

16.3.5.2.100.3 Diagram



16.3.5.2.100.4 Fields

Field	Function
31-30	enable/disable the de-noise function

Table continues on the next page...

Field	Function
SW_ENC_NOISE_REDUCTION_ENABLE	
29-24 SW_ENC_NOISE_LOW	the low boundary for noise estimation
23-16 —	Reserved
15-0 SW_ENC_NR_MBNUM_INVERT_REG	inverter of luma16 num *N

16.3.5.2.101 noise reduction 1 (swreg135)

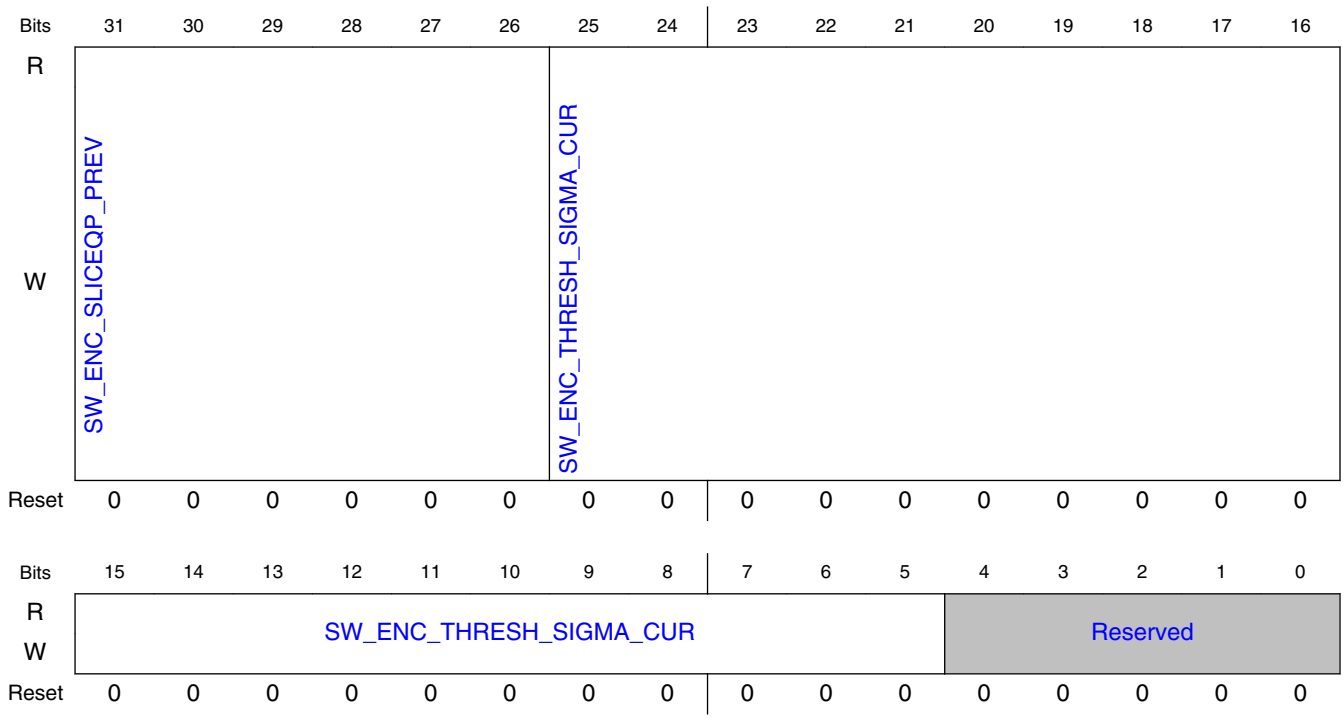
16.3.5.2.101.1 Offset

Register	Offset
swreg135	21Ch

16.3.5.2.101.2 Function

noise reduction 1

16.3.5.2.101.3 Diagram



16.3.5.2.101.4 Fields

Field	Function
31-26 SW_ENC_SLICEQP_PREV	previous frame slice QP
25-5 SW_ENC_THRESH_SIGMA_CUR	sigma threshold for current frame
4-0 —	Reserved

16.3.5.2.102 noise reduction 2 (swreg136)

16.3.5.2.102.1 Offset

Register	Offset
swreg136	220h

16.3.5.2.102.2 Function

noise reduction 2

16.3.5.2.102.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_SIGMA_CUR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_FRAME_SIGMA_CALCED															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.102.4 Fields

Field	Function
31-16 SW_ENC_SIGMA_CUR	noise sigma for current frame
15-0 SW_ENC_FRAME_SIGMA_CALCED	calculated sigma of coding frame

16.3.5.2.103 noise reduction 3 (swreg137)

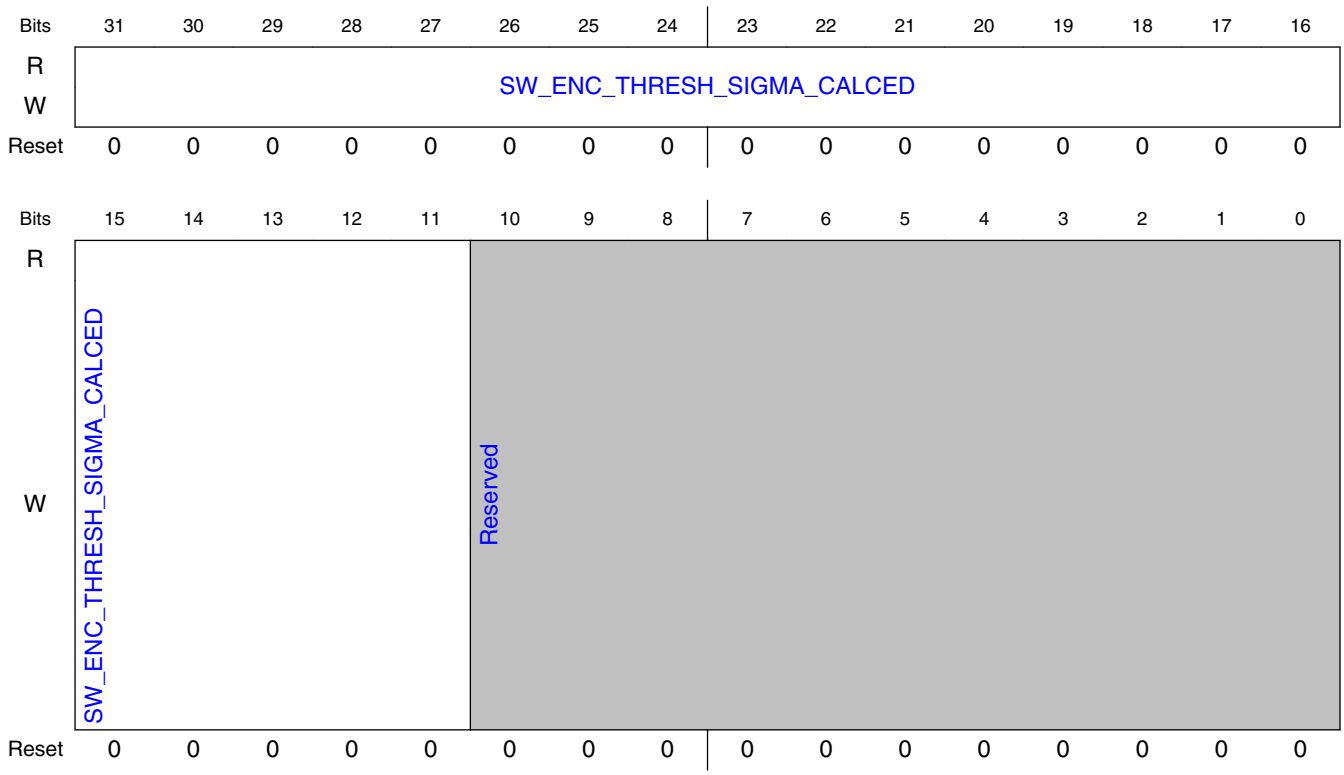
16.3.5.2.103.1 Offset

Register	Offset
swreg137	224h

16.3.5.2.103.2 Function

noise reduction 3

16.3.5.2.103.3 Diagram



16.3.5.2.103.4 Fields

Field	Function
31-11 SW_ENC_THRESH_SIGMA_CALCED	noise sigma for current frame
10-0 —	Reserved

16.3.5.2.104 inter me SSE lambda config 4. For H2V5 or later version. (swreg138_H2V5)

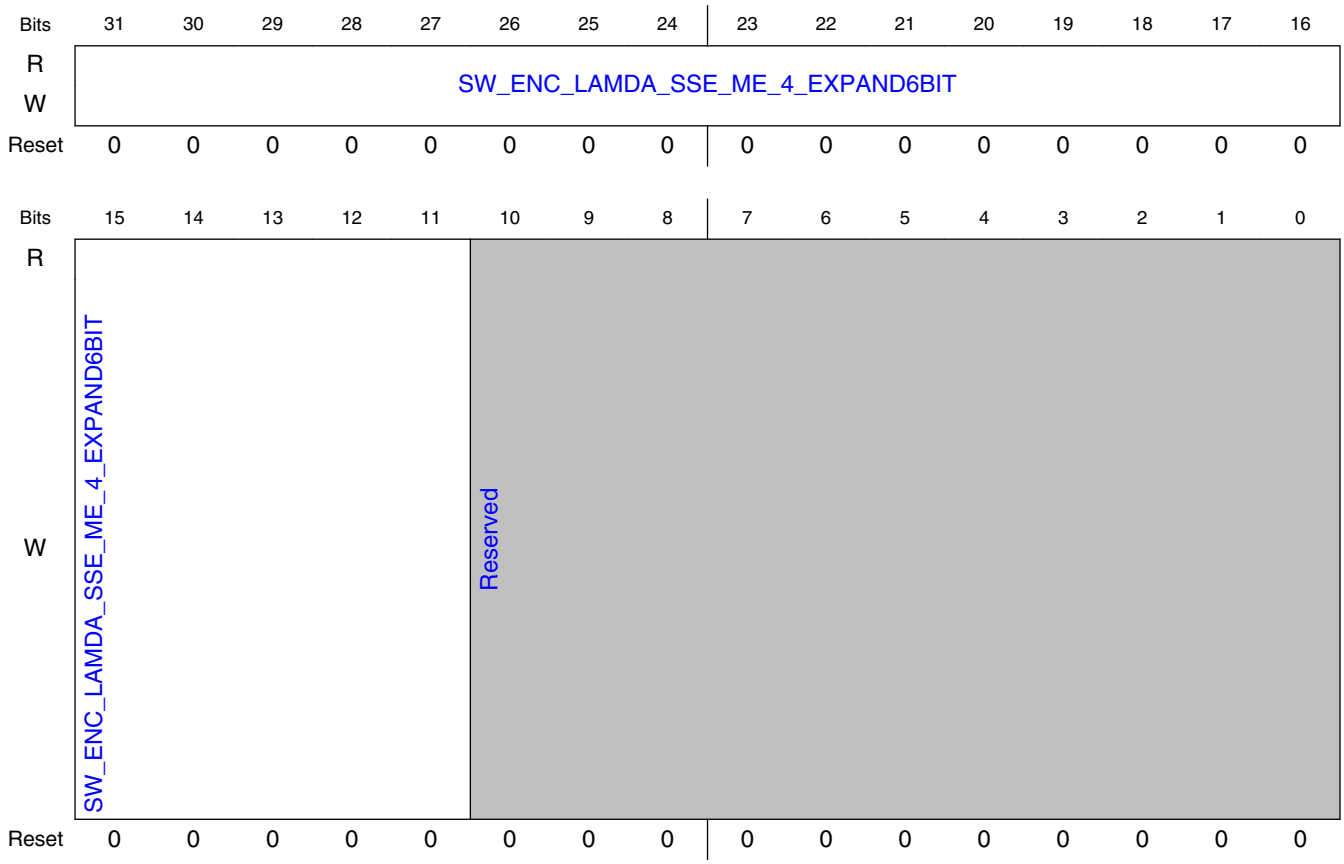
16.3.5.2.104.1 Offset

Register	Offset
swreg138_H2V5	228h

16.3.5.2.104.2 Function

inter me SSE lambda config 4. For H2V5 or later version.

16.3.5.2.104.3 Diagram



16.3.5.2.104.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_4_EXPAND6BIT	inter me SSE lambda 3
10-0 —	Reserved

16.3.5.2.105 inter me SSE lambda config 5. For H2V5 or later version. (swreg139_H2V5)

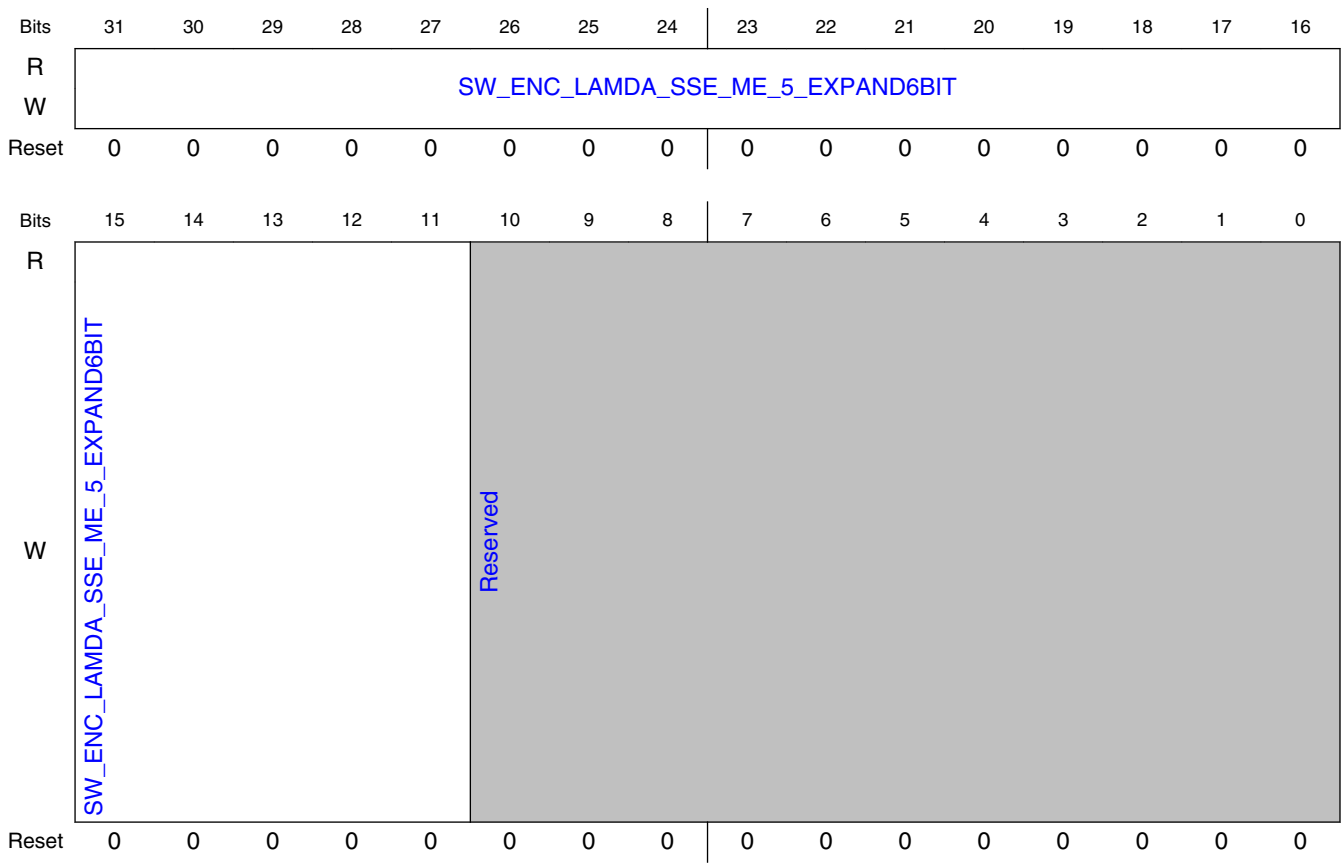
16.3.5.2.105.1 Offset

Register	Offset
swreg139_H2V5	22Ch

16.3.5.2.105.2 Function

inter me SSE lambda config 5. For H2V5 or later version.

16.3.5.2.105.3 Diagram



16.3.5.2.105.4 Fields

Field	Function
31-11	inter me SSE lambda 5

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SSE_ME_5_EXPAND6BIT	
10-0 —	Reserved

16.3.5.2.106 inter me SSE lambda config 6. For H2V5 or later version. (swreg140_H2V5)

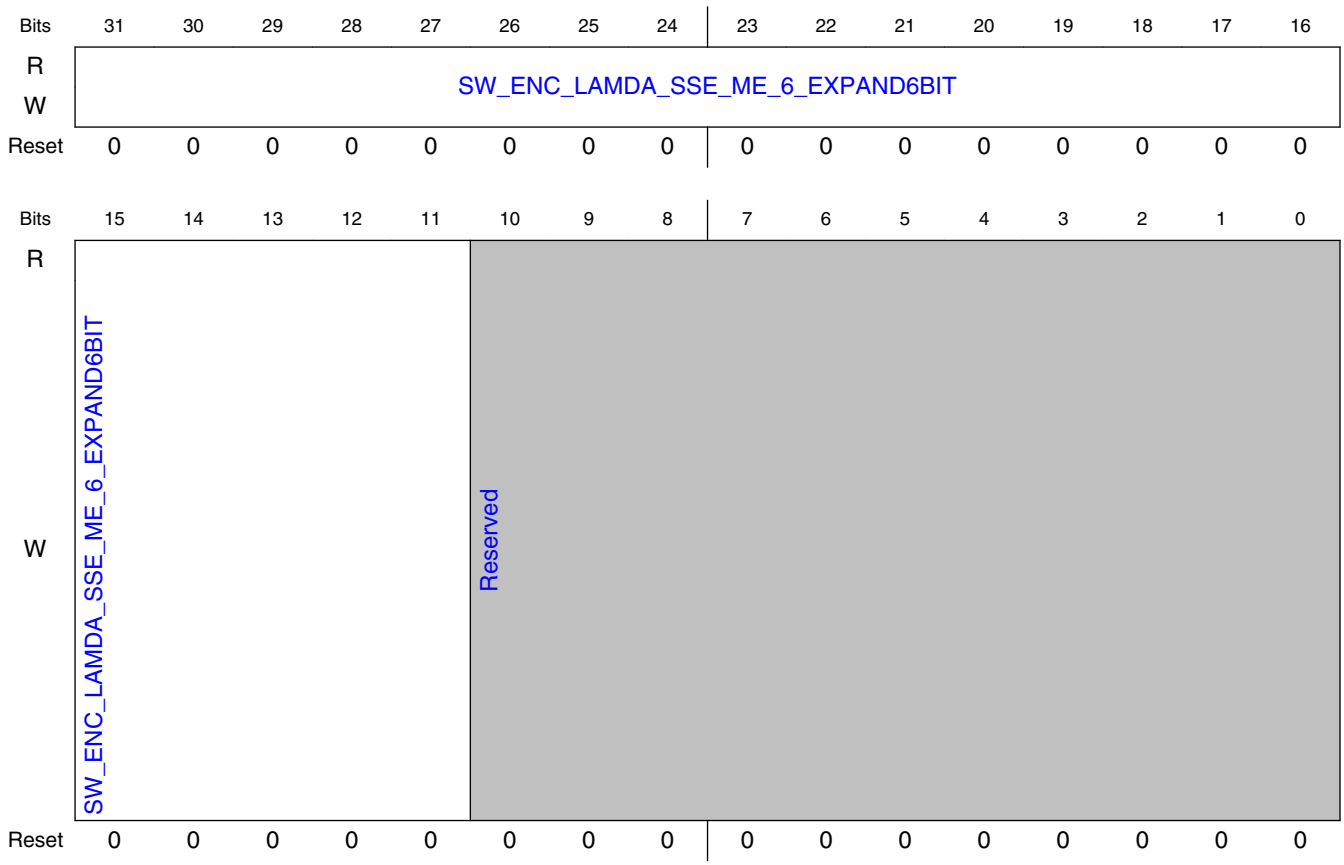
16.3.5.2.106.1 Offset

Register	Offset
swreg140_H2V5	230h

16.3.5.2.106.2 Function

inter me SSE lambda config 6. For H2V5 or later version.

16.3.5.2.106.3 Diagram



16.3.5.2.106.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_6_EXPAND6BIT	inter me SSE lambda 6
10-0 —	Reserved

16.3.5.2.107 inter me SSE lambda config 7. For H2V5 or later version. (swreg141_H2V5)

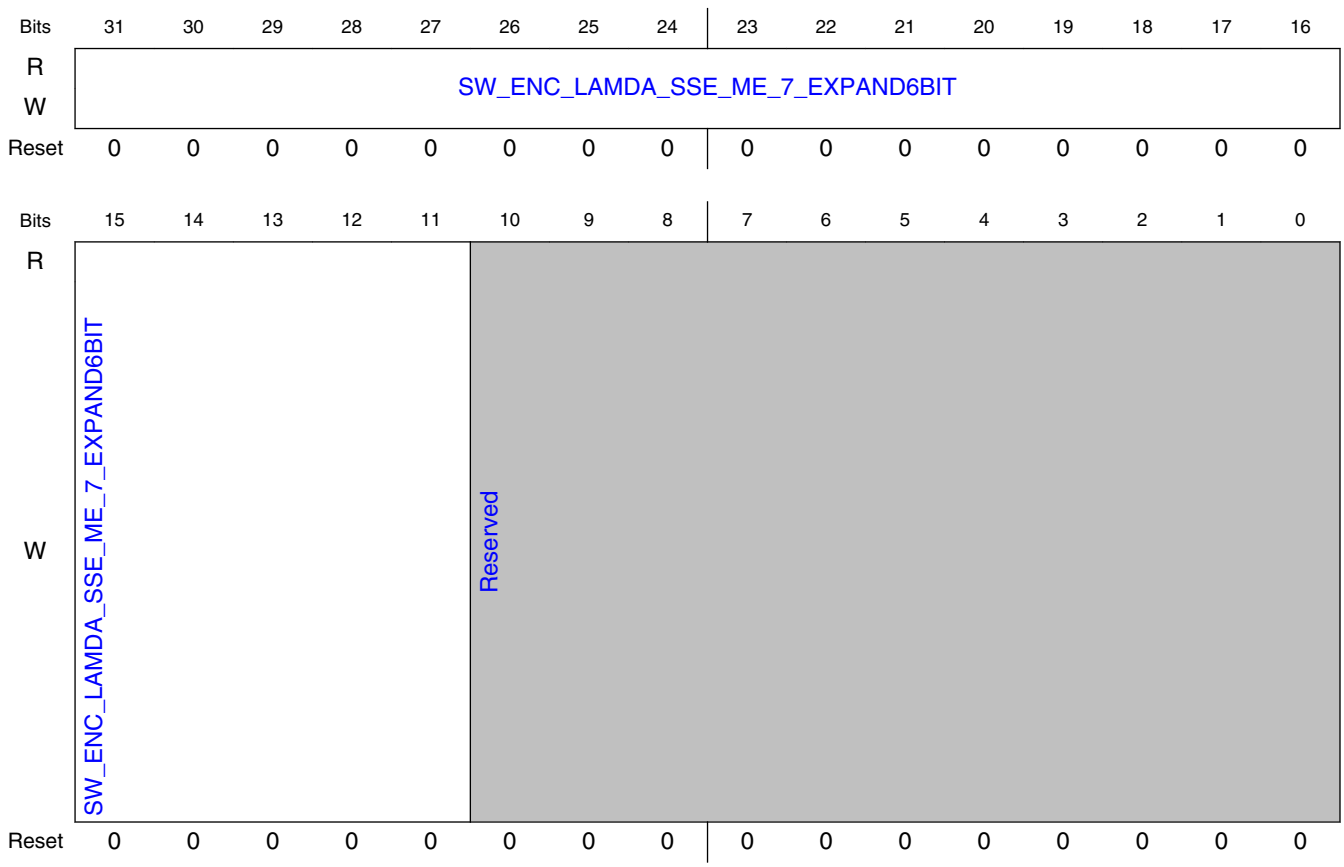
16.3.5.2.107.1 Offset

Register	Offset
swreg141_H2V5	234h

16.3.5.2.107.2 Function

inter me SSE lambda config 7. For H2V5 or later version.

16.3.5.2.107.3 Diagram



16.3.5.2.107.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_7_EXPAND6BIT	inter me SSE lambda 7
10-0	Reserved

Field	Function
—	

16.3.5.2.108 inter me SSE lambda config 8. For H2V5 or later version. (swreg142_H2V5)

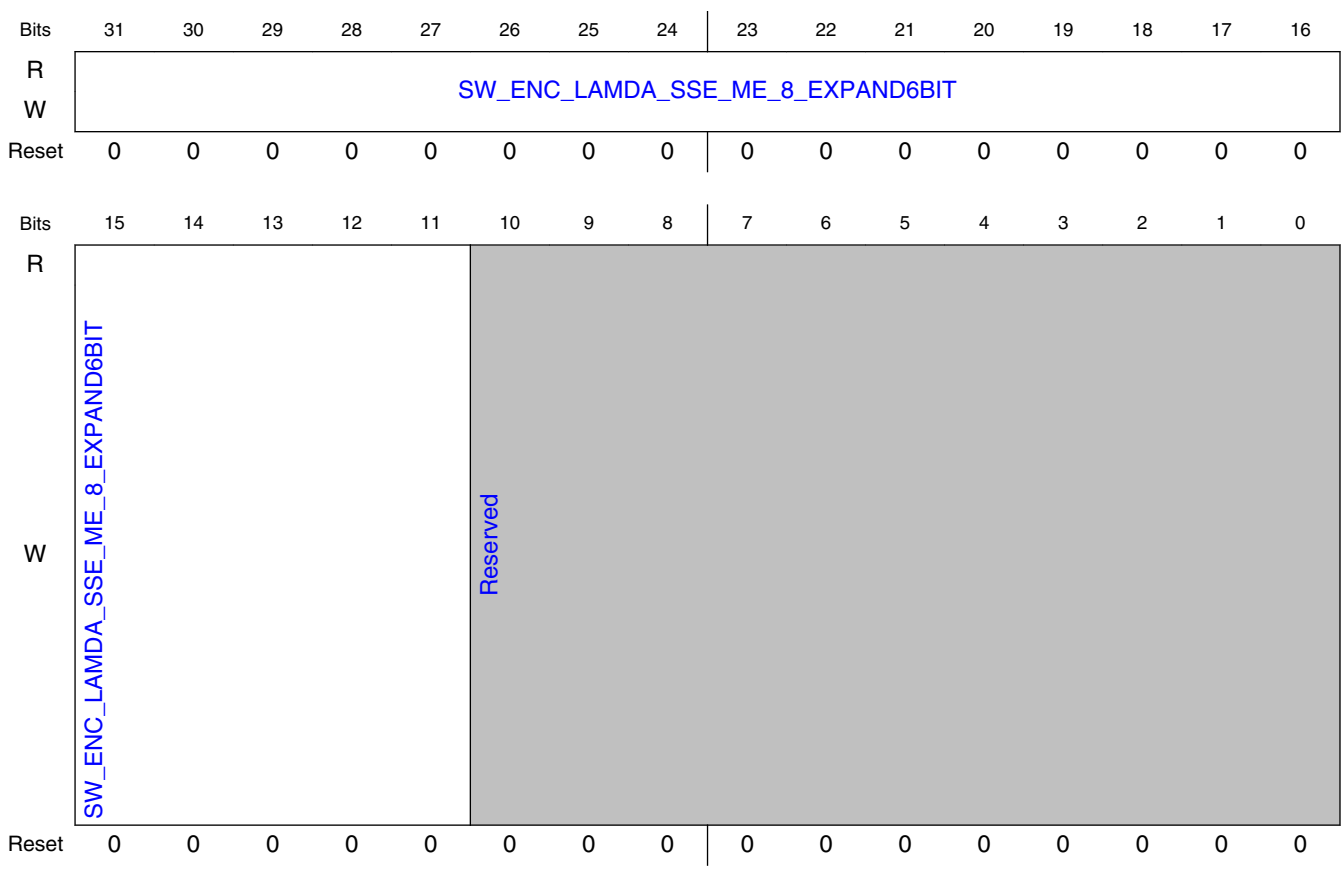
16.3.5.2.108.1 Offset

Register	Offset
swreg142_H2V5	238h

16.3.5.2.108.2 Function

inter me SSE lambda config 8. For H2V5 or later version.

16.3.5.2.108.3 Diagram



16.3.5.2.108.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_8_EXPAND6BIT	inter me SSE lambda 8
10-0 —	Reserved

16.3.5.2.109 inter me SSE lambda config 9. For H2V5 or later version. (swreg143_H2V5)

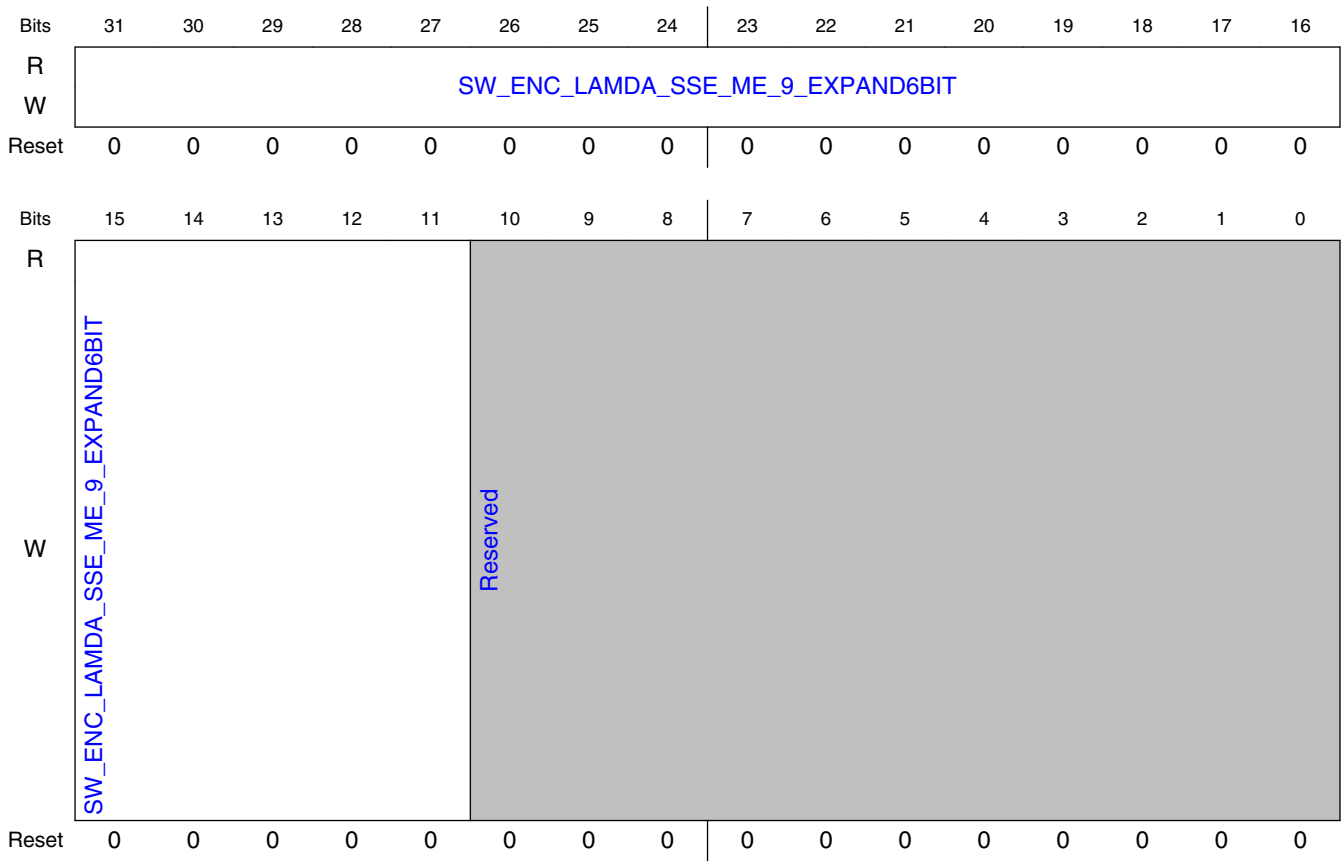
16.3.5.2.109.1 Offset

Register	Offset
swreg143_H2V5	23Ch

16.3.5.2.109.2 Function

inter me SSE lambda config 9. For H2V5 or later version.

16.3.5.2.109.3 Diagram



16.3.5.2.109.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_9_EXPAND6BIT	inter me SSE lambda 9
10-0 —	Reserved

16.3.5.2.110 inter me SSE lambda config 10. For H2V5 or later version. (swreg144_H2V5)

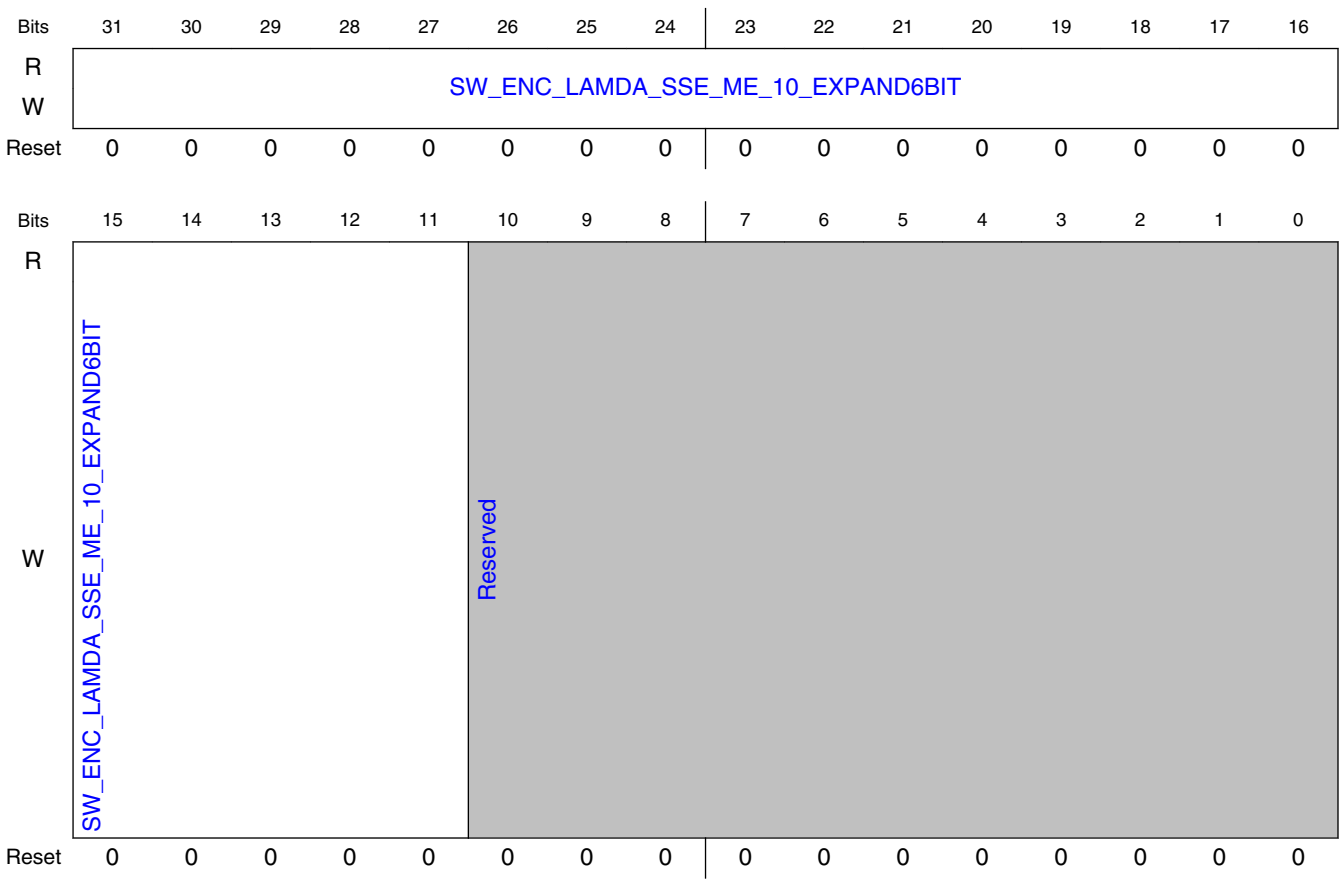
16.3.5.2.110.1 Offset

Register	Offset
swreg144_H2V5	240h

16.3.5.2.110.2 Function

inter me SSE lambda config 10. For H2V5 or later version.

16.3.5.2.110.3 Diagram



16.3.5.2.110.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_10_EXPAND6BIT	inter me SSE lambda 10

Table continues on the next page...

Field	Function
10-0 —	Reserved

16.3.5.2.111 inter me SSE lambda config 11. For H2V5 or later version. (swreg145_H2V5)

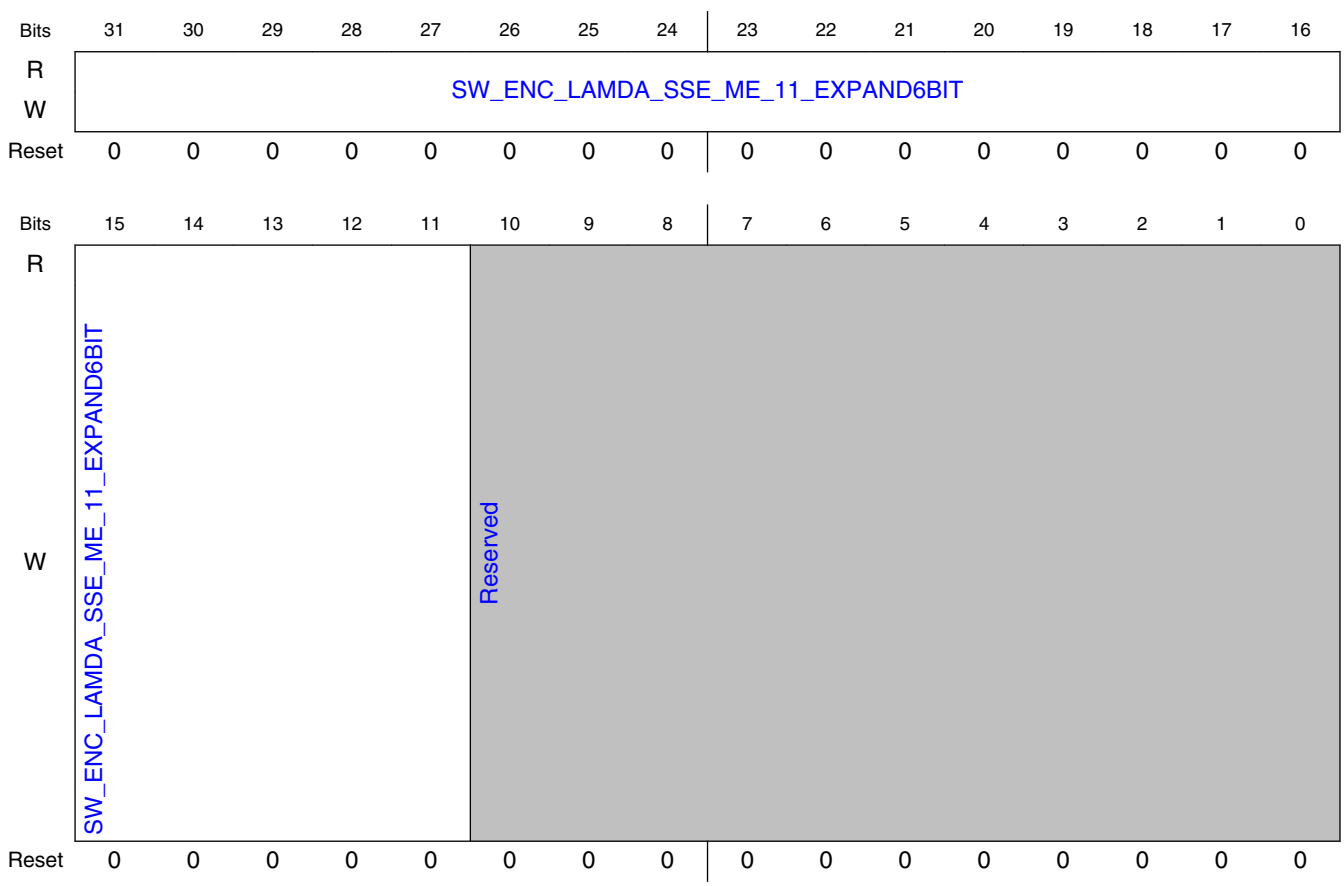
16.3.5.2.111.1 Offset

Register	Offset
swreg145_H2V5	244h

16.3.5.2.111.2 Function

inter me SSE lambda config 11. For H2V5 or later version.

16.3.5.2.111.3 Diagram



16.3.5.2.111.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_11_EXPAND6BIT	inter me SSE lambda 11
10-0 —	Reserved

16.3.5.2.112 inter me SSE lambda config 12. For H2V5 or later version. (swreg146_H2V5)

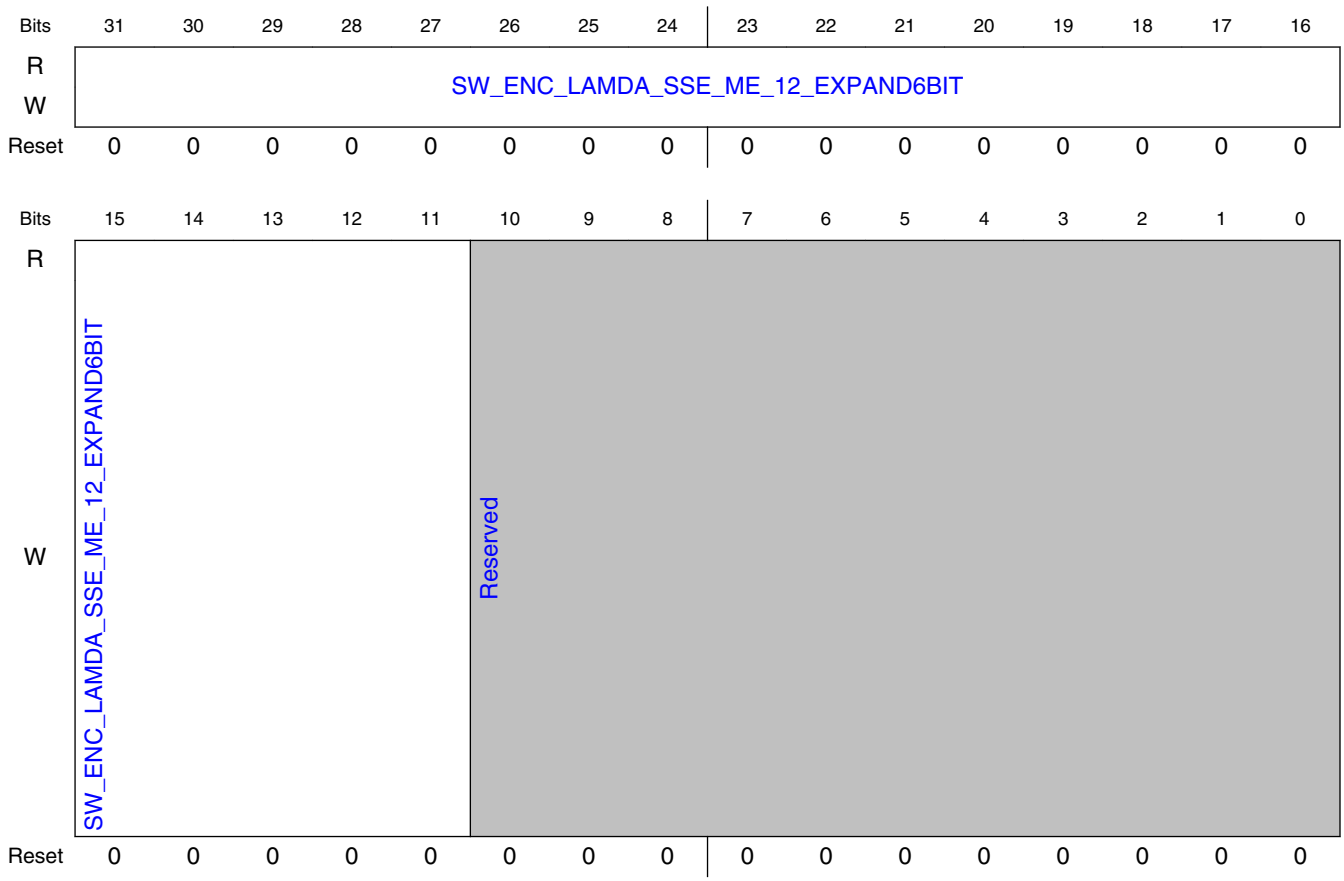
16.3.5.2.112.1 Offset

Register	Offset
swreg146_H2V5	248h

16.3.5.2.112.2 Function

inter me SSE lambda config 12. For H2V5 or later version.

16.3.5.2.112.3 Diagram



16.3.5.2.112.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_12_EXPAND6BIT	inter me SSE lambda 12
10-0 —	Reserved

16.3.5.2.113 inter me SSE lambda config 13. For H2V5 or later version. (swreg147_H2V5)

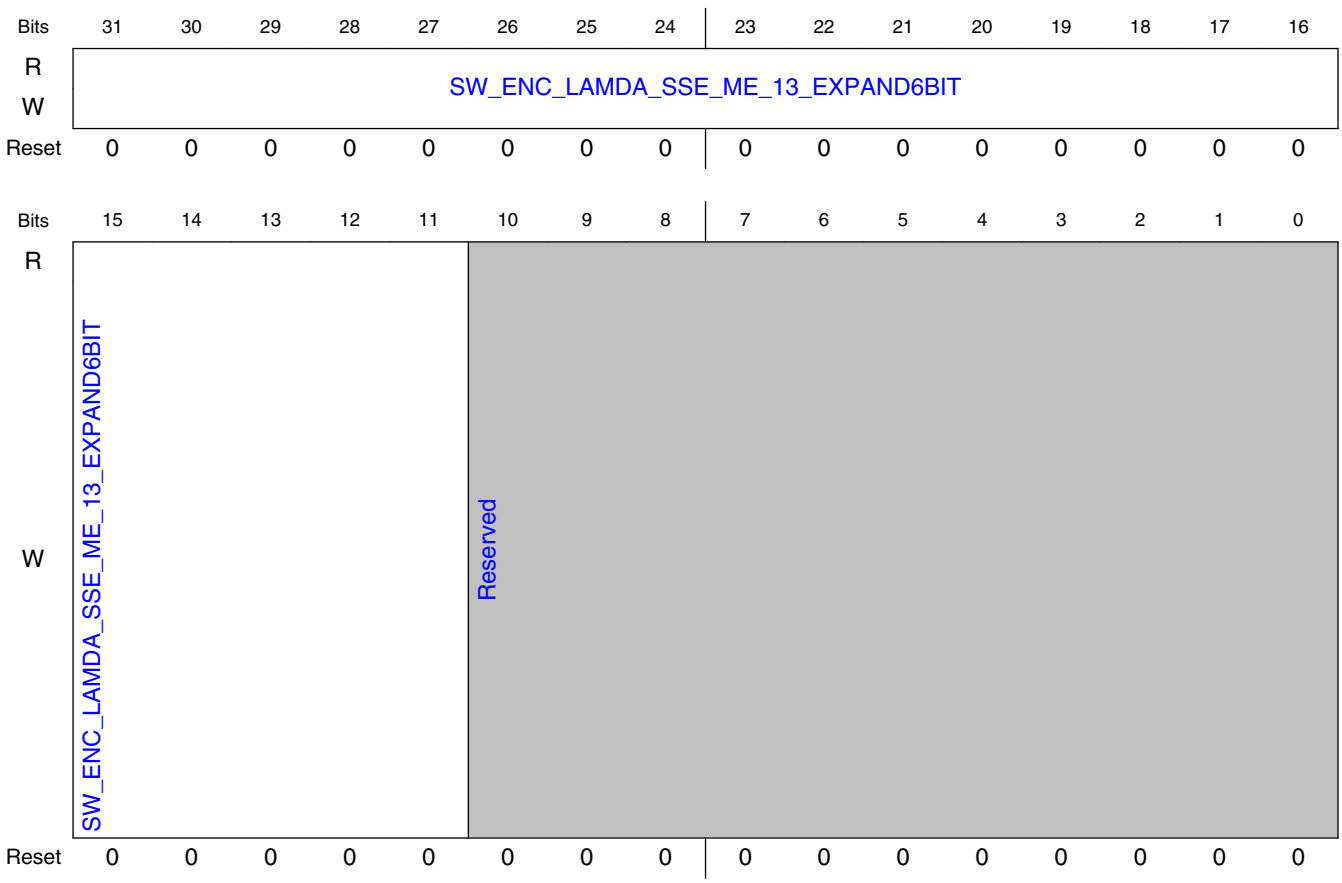
16.3.5.2.113.1 Offset

Register	Offset
swreg147_H2V5	24Ch

16.3.5.2.113.2 Function

inter me SSE lambda config 13. For H2V5 or later version.

16.3.5.2.113.3 Diagram



16.3.5.2.113.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_13_EXPAND6BIT	inter me SSE lambda 13

Table continues on the next page...

Field	Function
10-0 —	Reserved

16.3.5.2.114 inter me SSE lambda config 14. For H2V5 or later version. (swreg148_H2V5)

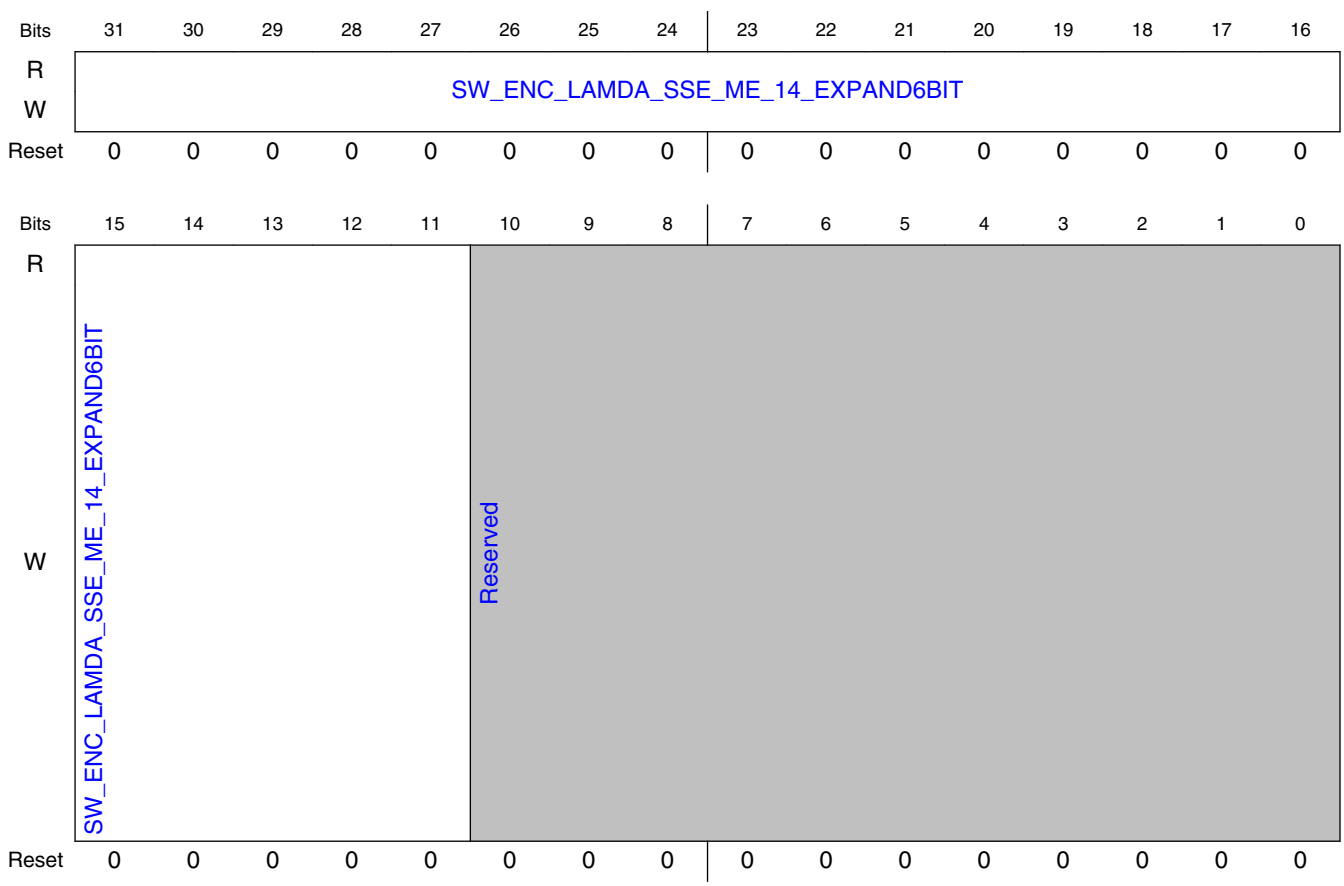
16.3.5.2.114.1 Offset

Register	Offset
swreg148_H2V5	250h

16.3.5.2.114.2 Function

inter me SSE lambda config 14. For H2V5 or later version.

16.3.5.2.114.3 Diagram



16.3.5.2.114.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_14_EXPAND6BIT	inter me SSE lambda 14
10-0 —	Reserved

16.3.5.2.115 inter me SSE lambda config 15. For H2V5 or later version. (swreg149_H2V5)

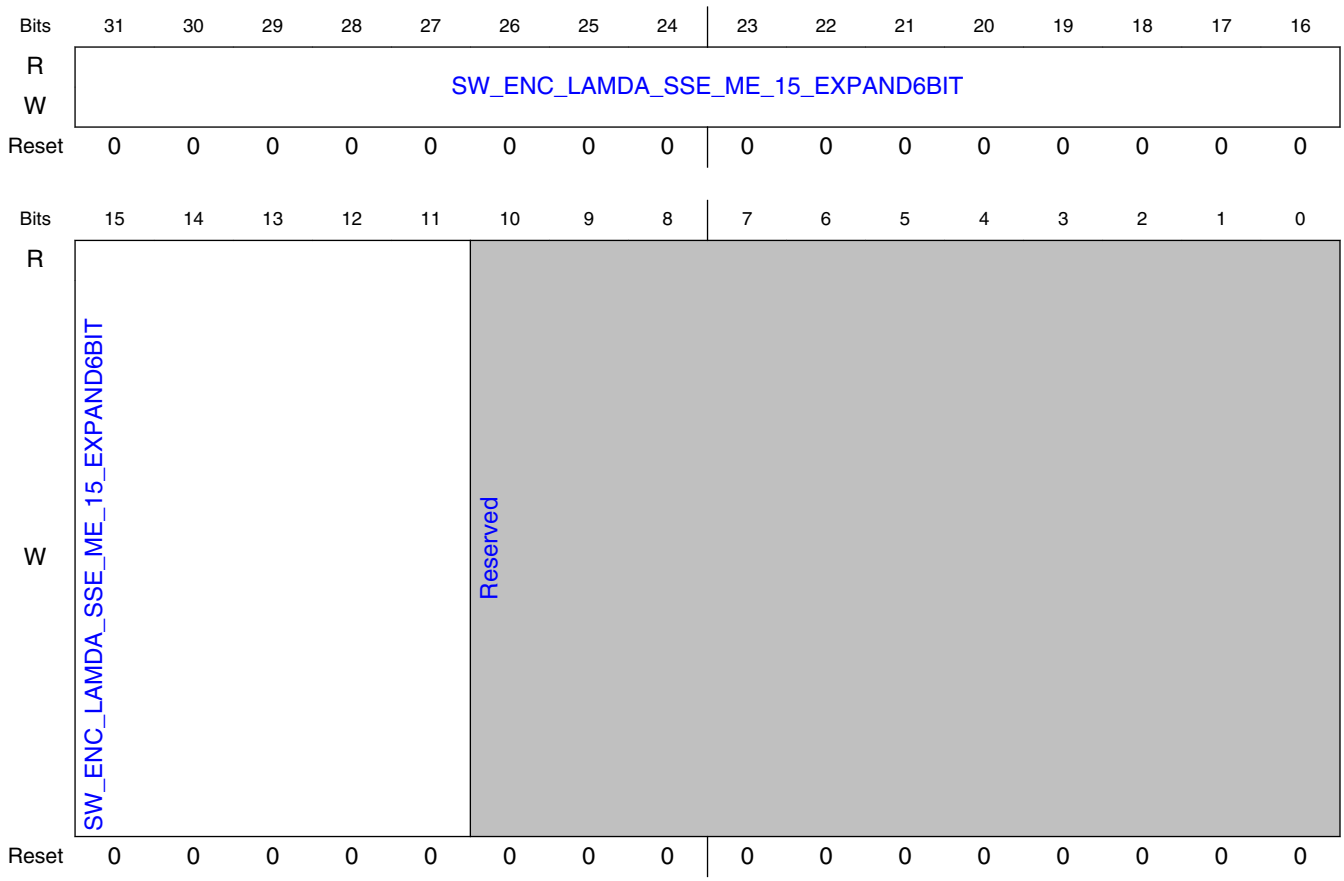
16.3.5.2.115.1 Offset

Register	Offset
swreg149_H2V5	254h

16.3.5.2.115.2 Function

inter me SSE lambda config 15. For H2V5 or later version.

16.3.5.2.115.3 Diagram



16.3.5.2.115.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_15_EXPAND6BIT	inter me SSE lambda 15
10-0 —	Reserved

16.3.5.2.116 inter me SATD lambda config 8 (swreg150)

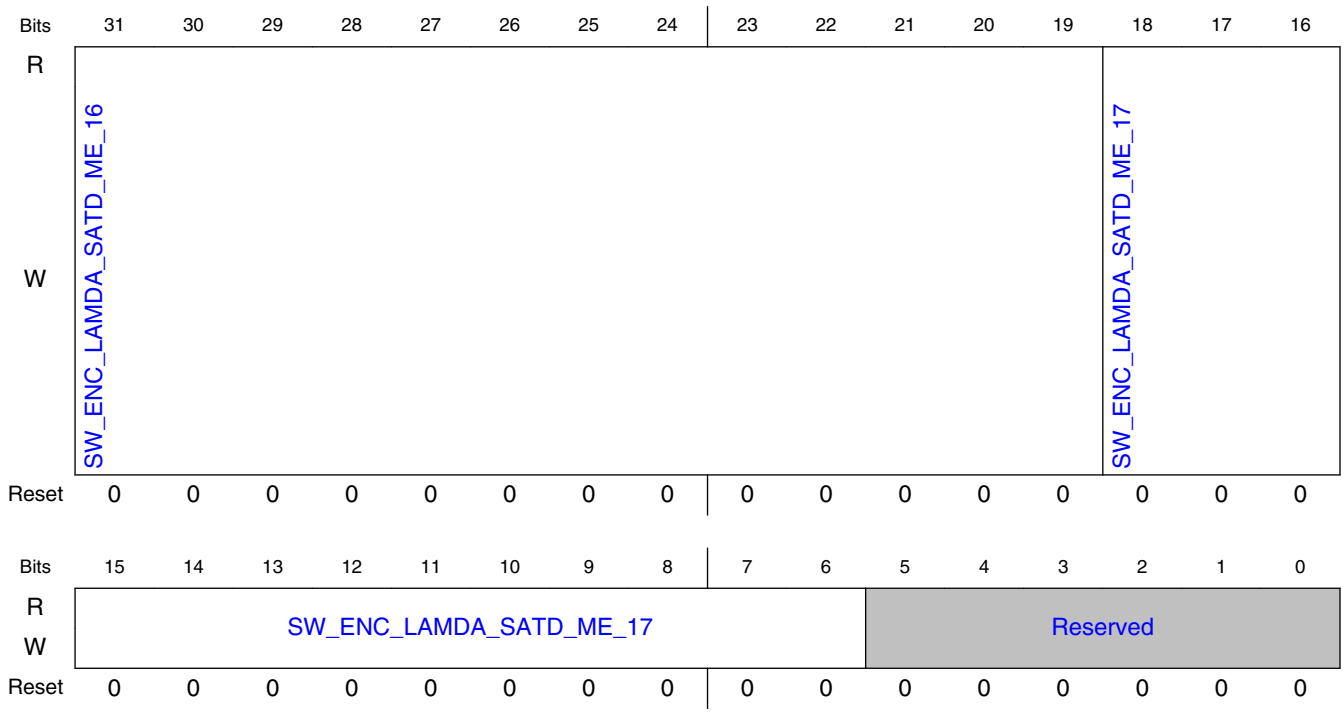
16.3.5.2.116.1 Offset

Register	Offset
swreg150	258h

16.3.5.2.116.2 Function

inter me SATD lambda config 8

16.3.5.2.116.3 Diagram



16.3.5.2.116.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_16	lambda satd me 16
18-6 SW_ENC_LAMDA_SATD_ME_17	lambda satd me 17
5-0 —	Reserved

16.3.5.2.117 inter me SATD lambda config 9 (swreg151)

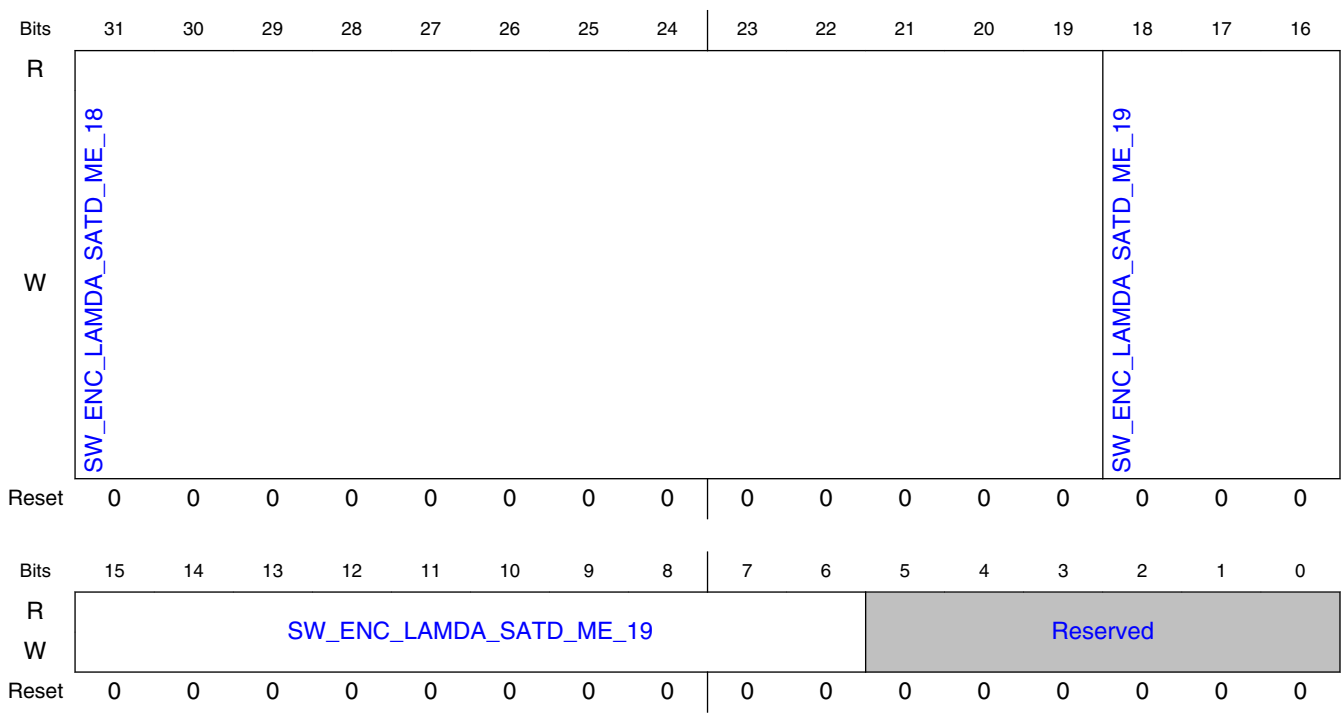
16.3.5.2.117.1 Offset

Register	Offset
swreg151	25Ch

16.3.5.2.117.2 Function

inter me SATD lambda config 9

16.3.5.2.117.3 Diagram



16.3.5.2.117.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_18	lambda satd me 18
18-6	lambda satd me 19

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_19	
5-0 —	Reserved

16.3.5.2.118 inter me SATD lambda config 10 (swreg152)

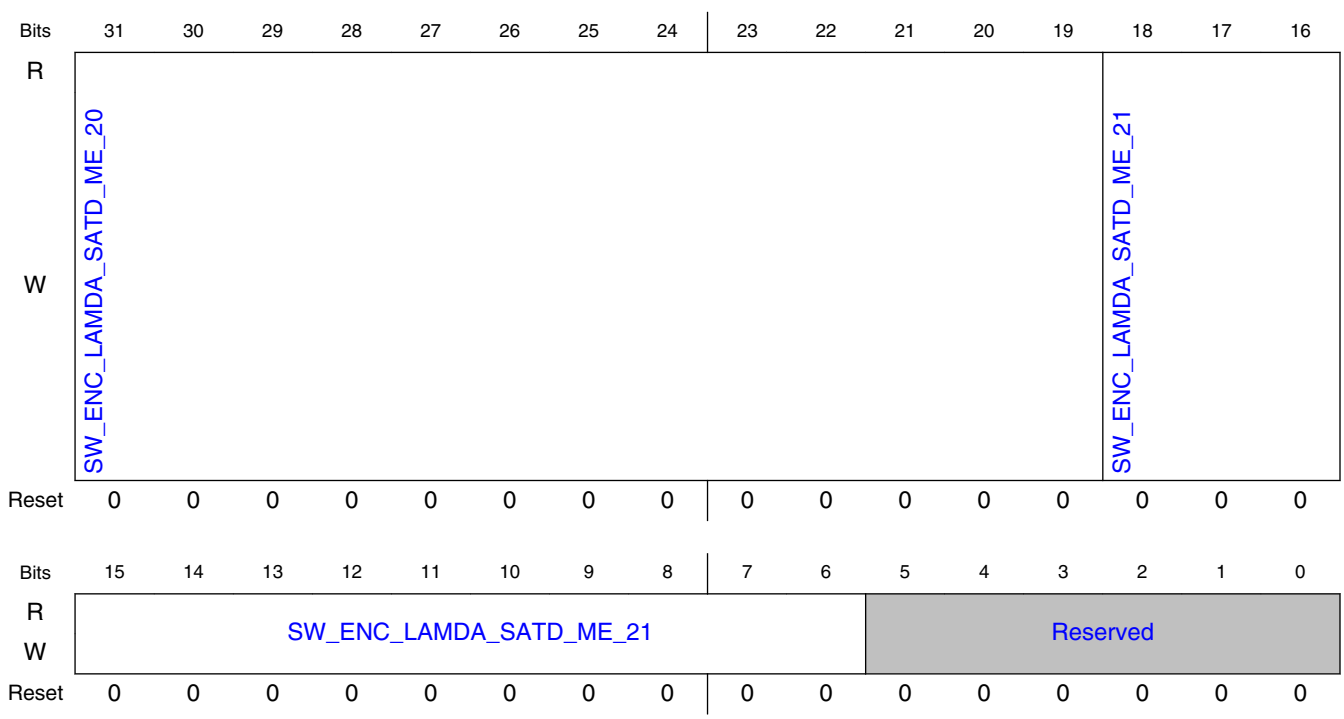
16.3.5.2.118.1 Offset

Register	Offset
swreg152	260h

16.3.5.2.118.2 Function

inter me SATD lambda config 10

16.3.5.2.118.3 Diagram



16.3.5.2.118.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_20	lambda satd me 20
18-6 SW_ENC_LAMDA_SATD_ME_21	lambda satd me 21
5-0 —	Reserved

16.3.5.2.119 inter me SATD lambda config 11 (swreg153)

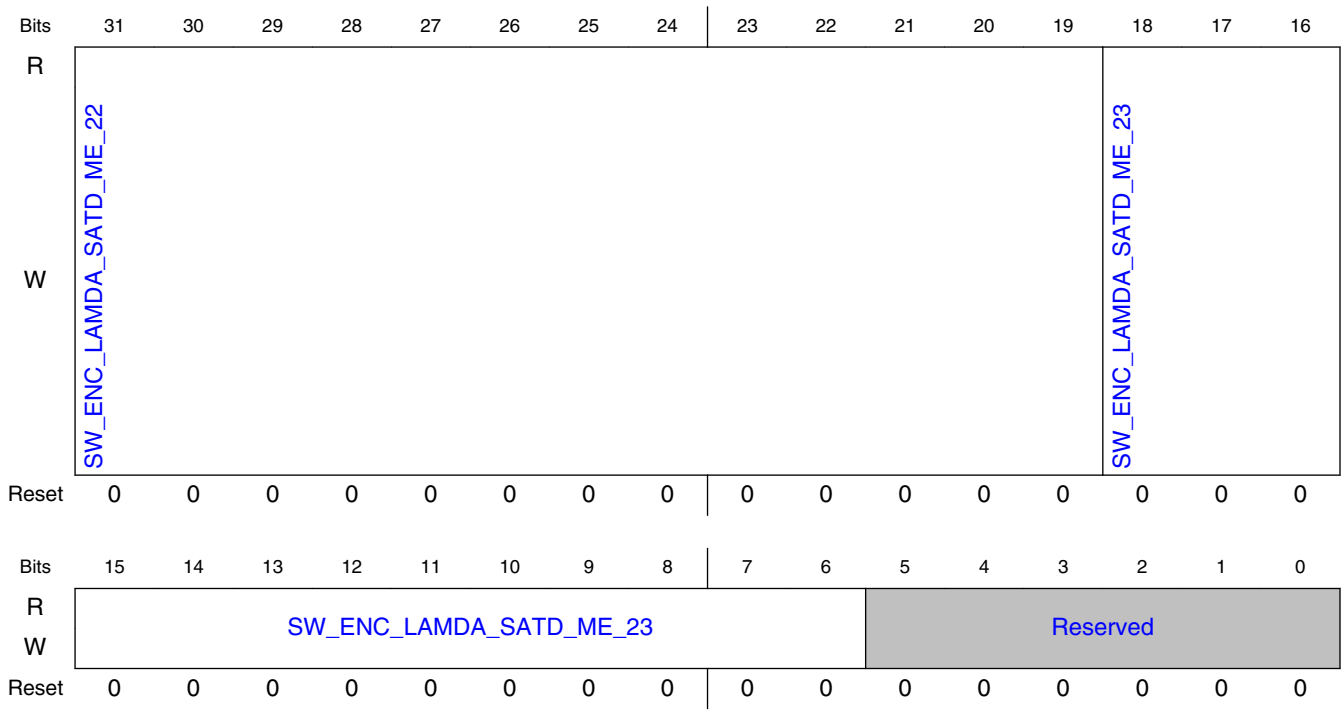
16.3.5.2.119.1 Offset

Register	Offset
swreg153	264h

16.3.5.2.119.2 Function

inter me SATD lambda config 11

16.3.5.2.119.3 Diagram



16.3.5.2.119.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_22	lambda satd me 22
18-6 SW_ENC_LAMDA_SATD_ME_23	lambda satd me 23
5-0 —	Reserved

16.3.5.2.120 inter me SATD lambda config 12 (swreg154)

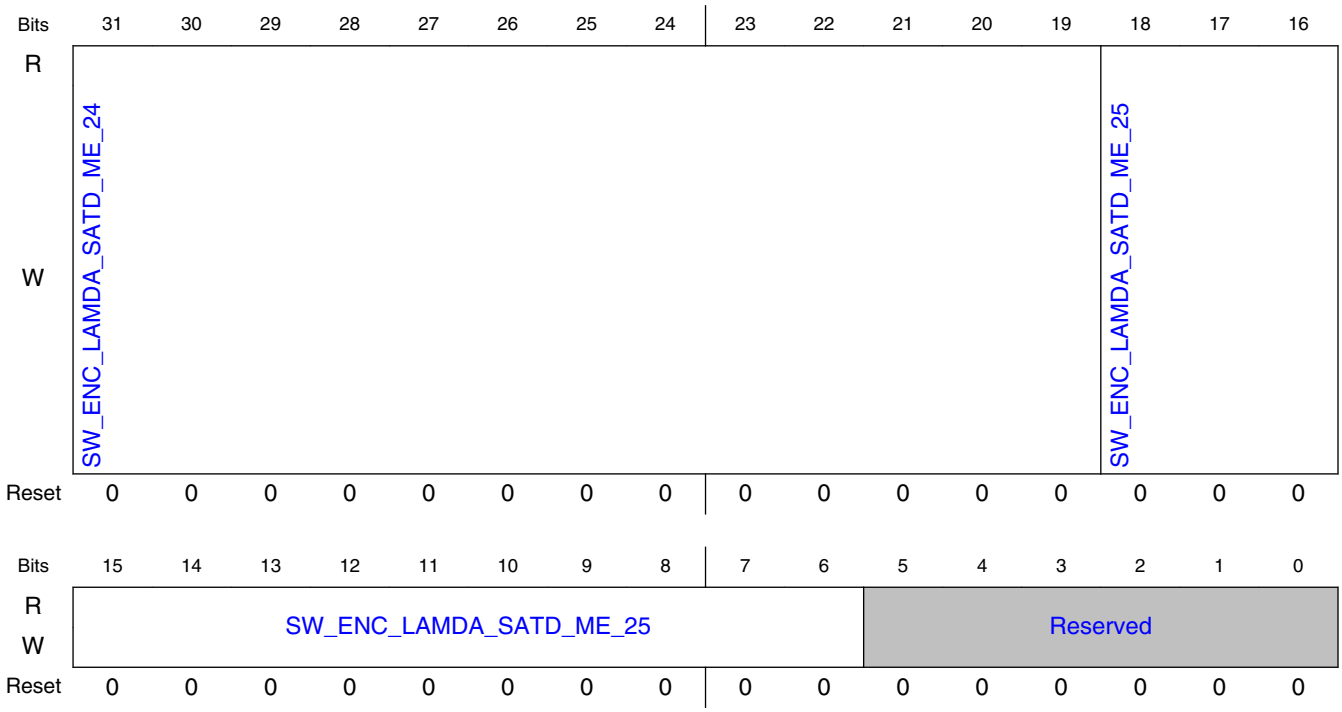
16.3.5.2.120.1 Offset

Register	Offset
swreg154	268h

16.3.5.2.120.2 Function

inter me SATD lambda config 12

16.3.5.2.120.3 Diagram



16.3.5.2.120.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_24	lambda satd me 24
18-6 SW_ENC_LAMDA_SATD_ME_25	lambda satd me 25
5-0 —	Reserved

16.3.5.2.121 inter me SATD lambda config 13 (swreg155)

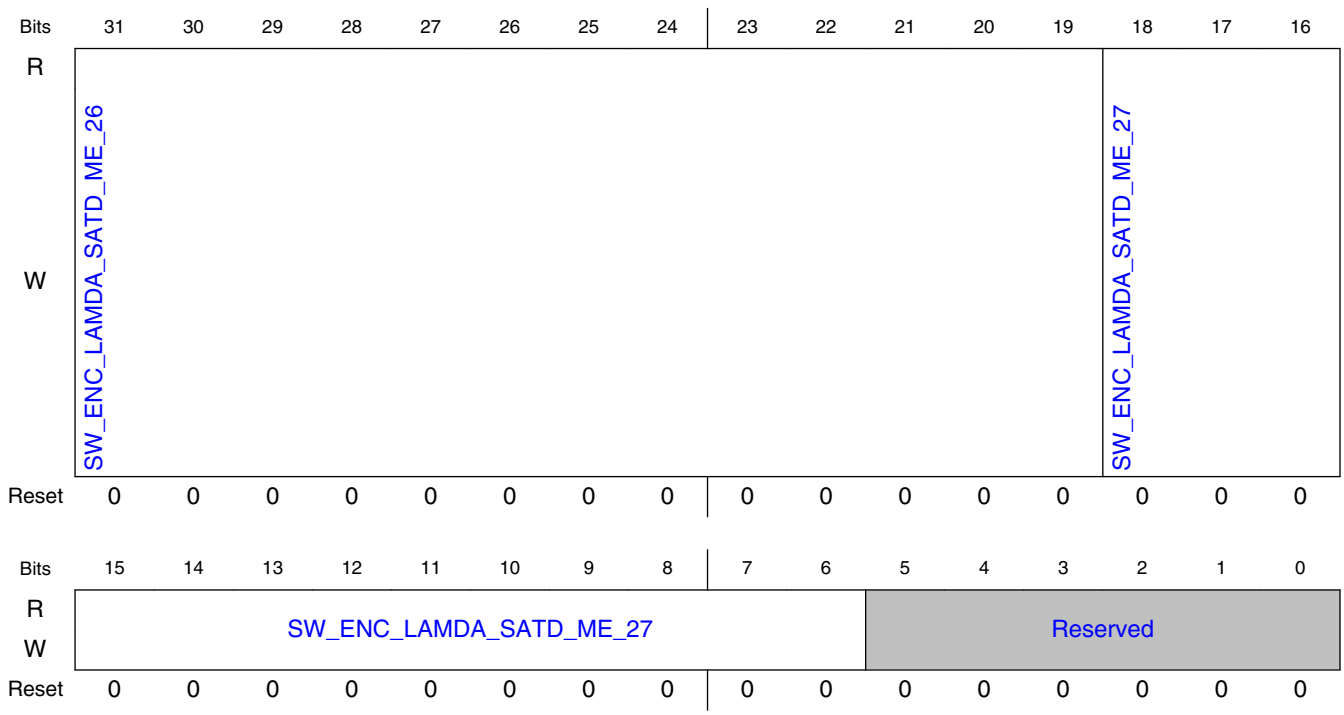
16.3.5.2.121.1 Offset

Register	Offset
swreg155	26Ch

16.3.5.2.121.2 Function

inter me SATD lambda config 13

16.3.5.2.121.3 Diagram



16.3.5.2.121.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_26	lambda satd me 26
18-6 SW_ENC_LAMDA_SATD_ME_27	lambda satd me 27
5-0	Reserved

Field	Function
—	

16.3.5.2.122 inter me SATD lambda config 14 (swreg156)

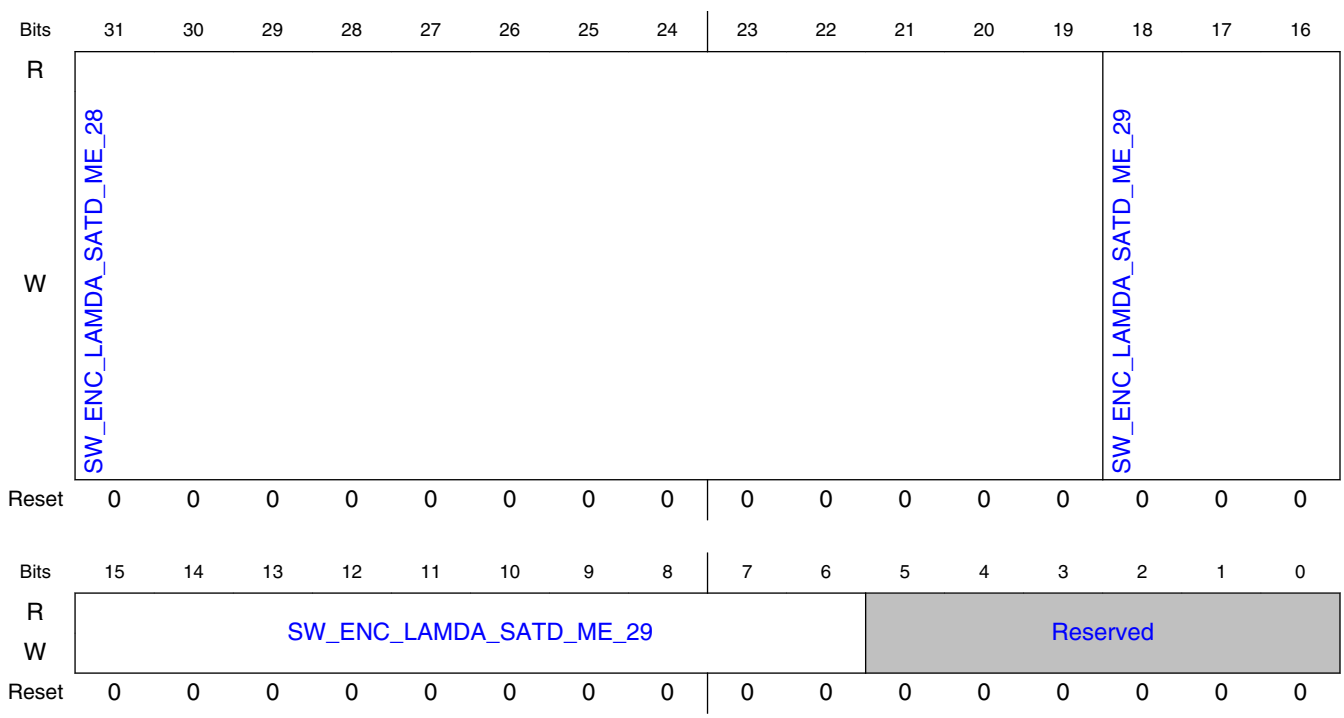
16.3.5.2.122.1 Offset

Register	Offset
swreg156	270h

16.3.5.2.122.2 Function

inter me SATD lambda config 14

16.3.5.2.122.3 Diagram



16.3.5.2.122.4 Fields

Field	Function
31-19	lambda satd me 28

Table continues on the next page...

Field	Function
SW_ENC_LAMDA_SATD_ME_28	
18-6 SW_ENC_LAMDA_SATD_ME_29	lambda satd me 29
5-0 —	Reserved

16.3.5.2.123 inter me SATD lambda config 15 (swreg157)

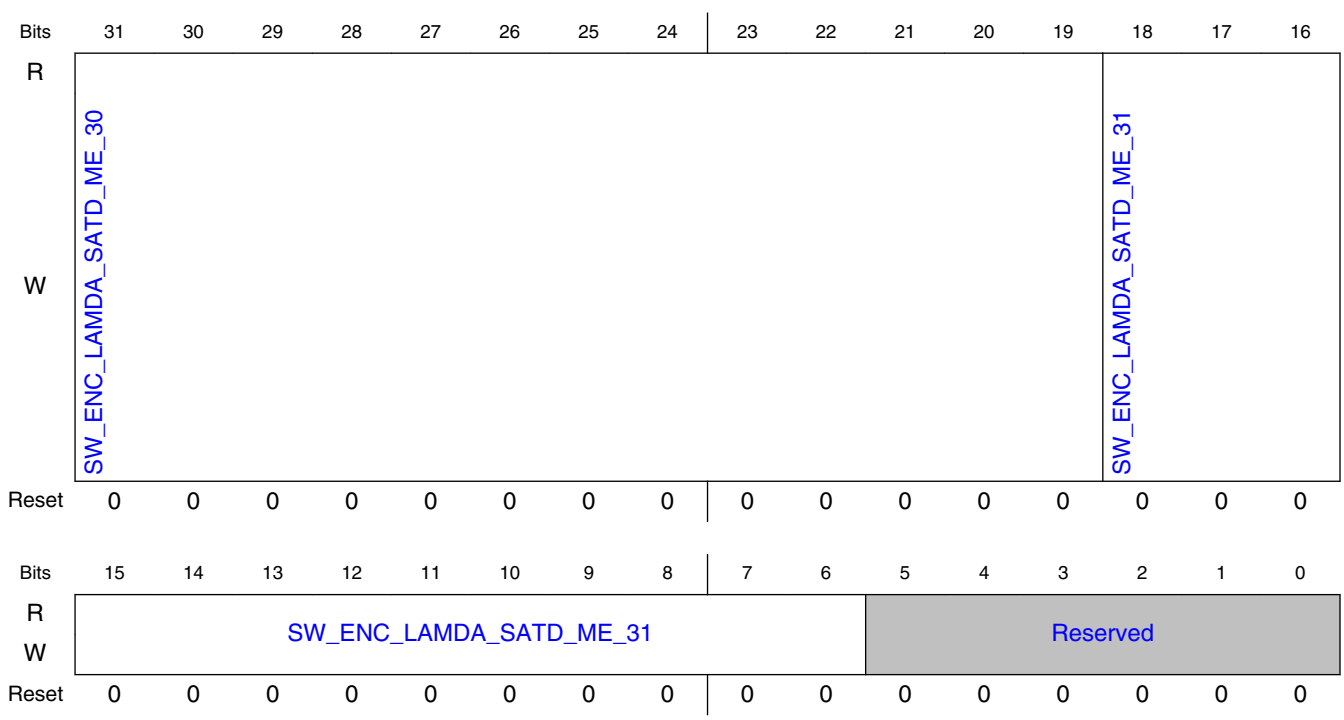
16.3.5.2.123.1 Offset

Register	Offset
swreg157	274h

16.3.5.2.123.2 Function

inter me SATD lambda config 15

16.3.5.2.123.3 Diagram



16.3.5.2.123.4 Fields

Field	Function
31-19 SW_ENC_LAMDA_SATD_ME_30	lambda satd me 30
18-6 SW_ENC_LAMDA_SATD_ME_31	lambda satd me 31
5-0 —	Reserved

16.3.5.2.124 inter me SSE lambda config 16 (swreg158)

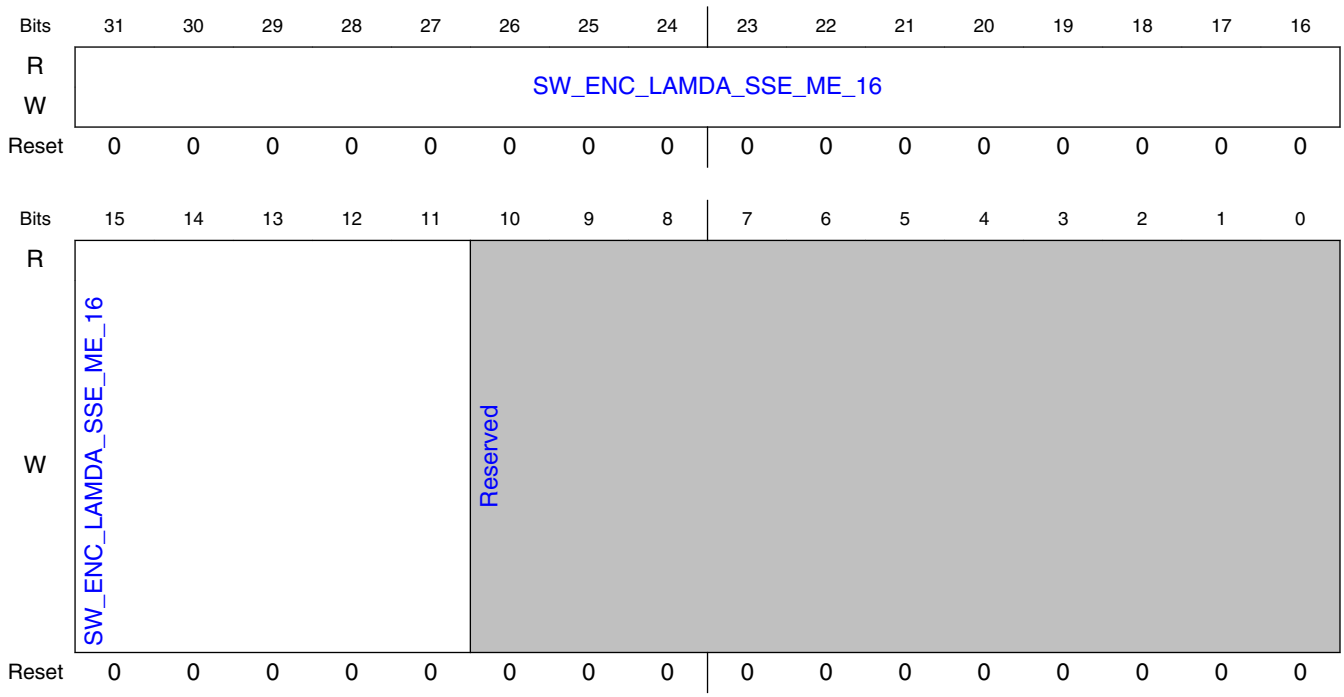
16.3.5.2.124.1 Offset

Register	Offset
swreg158	278h

16.3.5.2.124.2 Function

inter me SSE lambda config 16

16.3.5.2.124.3 Diagram



16.3.5.2.124.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_16	inter me SSE lambda 16
10-0 —	Reserved

16.3.5.2.125 inter me SSE lambda config 17 (swreg159)

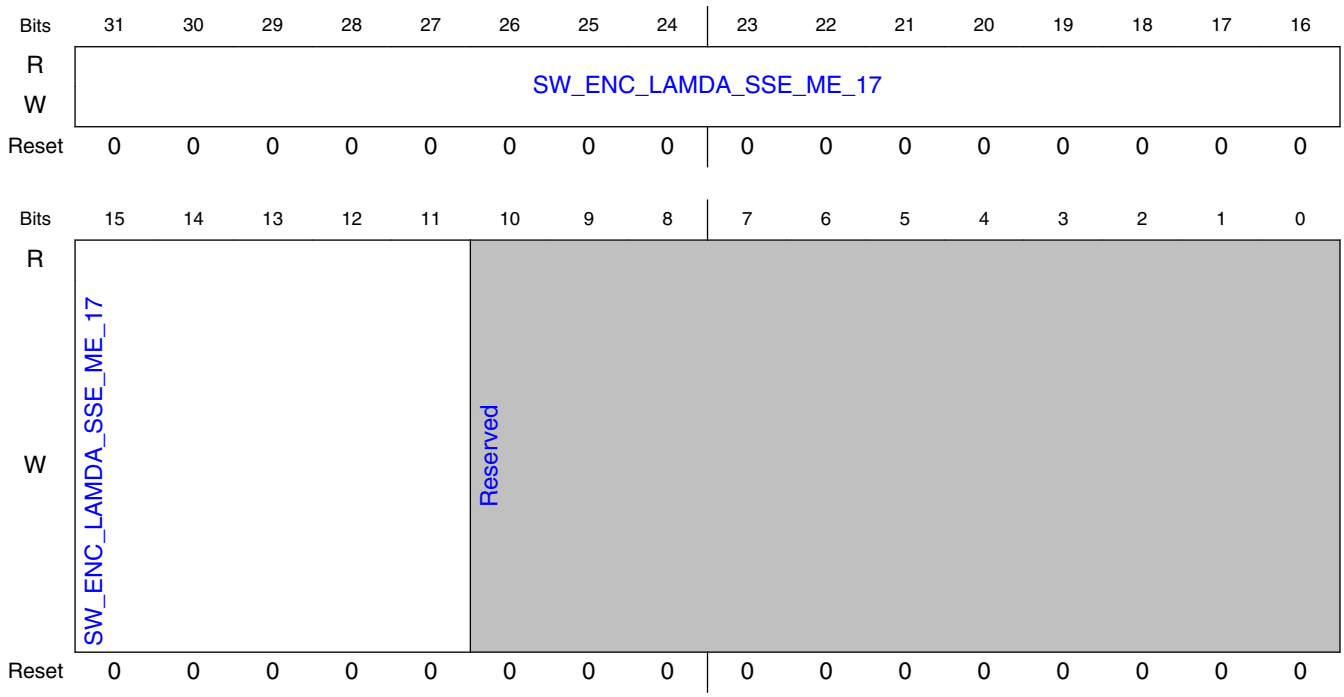
16.3.5.2.125.1 Offset

Register	Offset
swreg159	27Ch

16.3.5.2.125.2 Function

inter me SSE lambda config 17

16.3.5.2.125.3 Diagram



16.3.5.2.125.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_17	inter me SSE lambda 17
10-0 —	Reserved

16.3.5.2.126 inter me SSE lambda config 18 (swreg160)

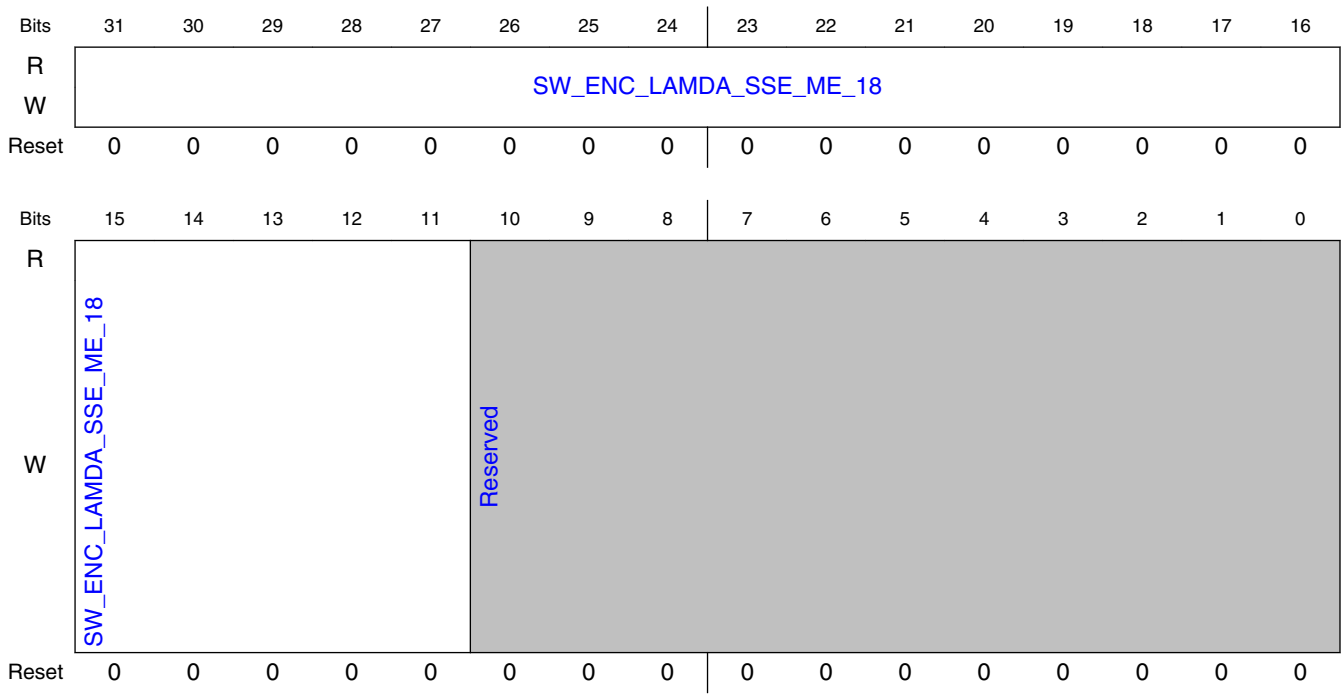
16.3.5.2.126.1 Offset

Register	Offset
swreg160	280h

16.3.5.2.126.2 Function

inter me SSE lambda config 18

16.3.5.2.126.3 Diagram



16.3.5.2.126.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_18	inter me SSE lambda 18
10-0 —	Reserved

16.3.5.2.127 inter me SSE lambda config 19 (swreg161)

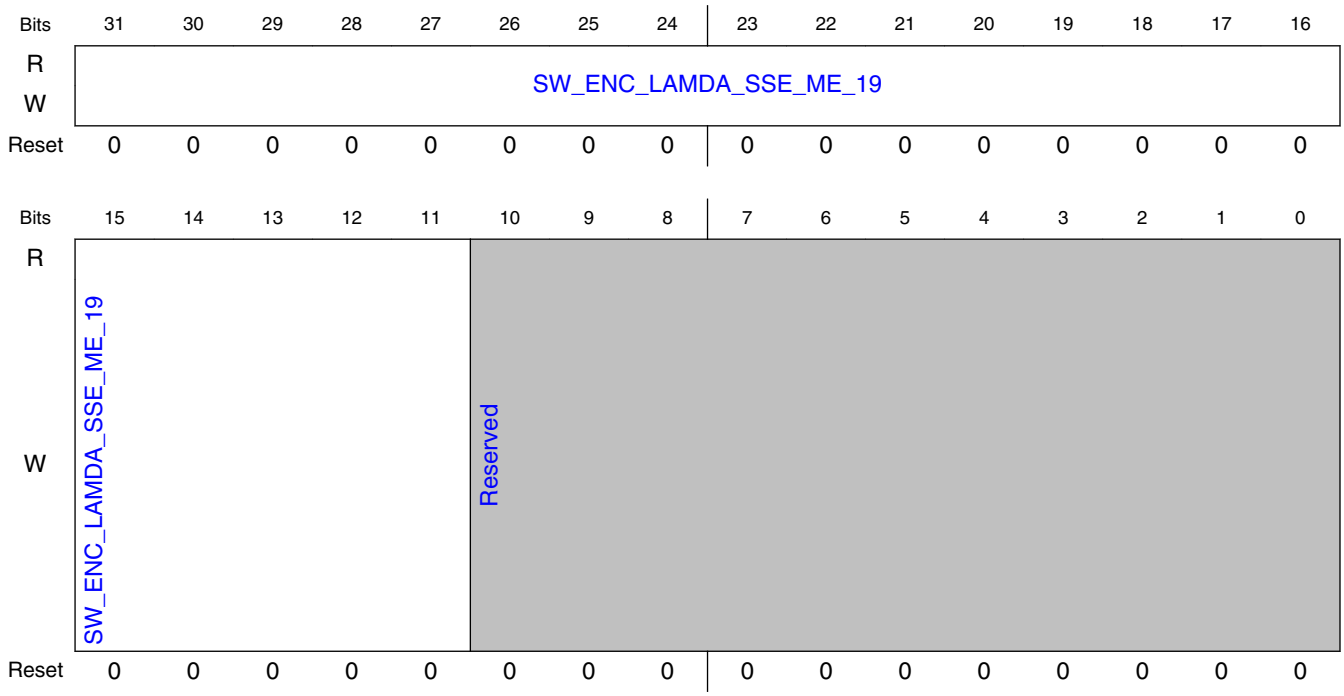
16.3.5.2.127.1 Offset

Register	Offset
swreg161	284h

16.3.5.2.127.2 Function

inter me SSE lambda config 19

16.3.5.2.127.3 Diagram



16.3.5.2.127.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_19	inter me SSE lambda 19
10-0 —	Reserved

16.3.5.2.128 inter me SSE lambda config 20 (swreg162)

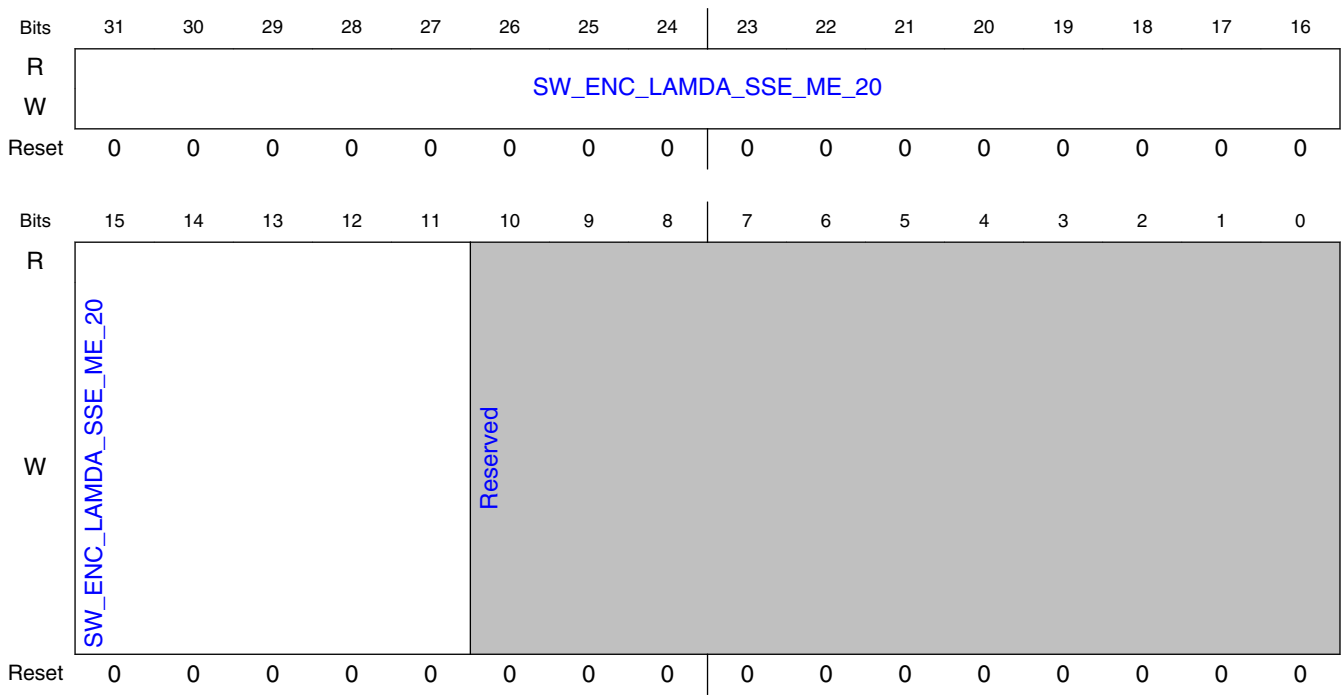
16.3.5.2.128.1 Offset

Register	Offset
swreg162	288h

16.3.5.2.128.2 Function

inter me SSE lambda config 20

16.3.5.2.128.3 Diagram



16.3.5.2.128.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_20	inter me SSE lambda 20
10-0 —	Reserved

16.3.5.2.129 inter me SSE lambda config 21 (swreg163)

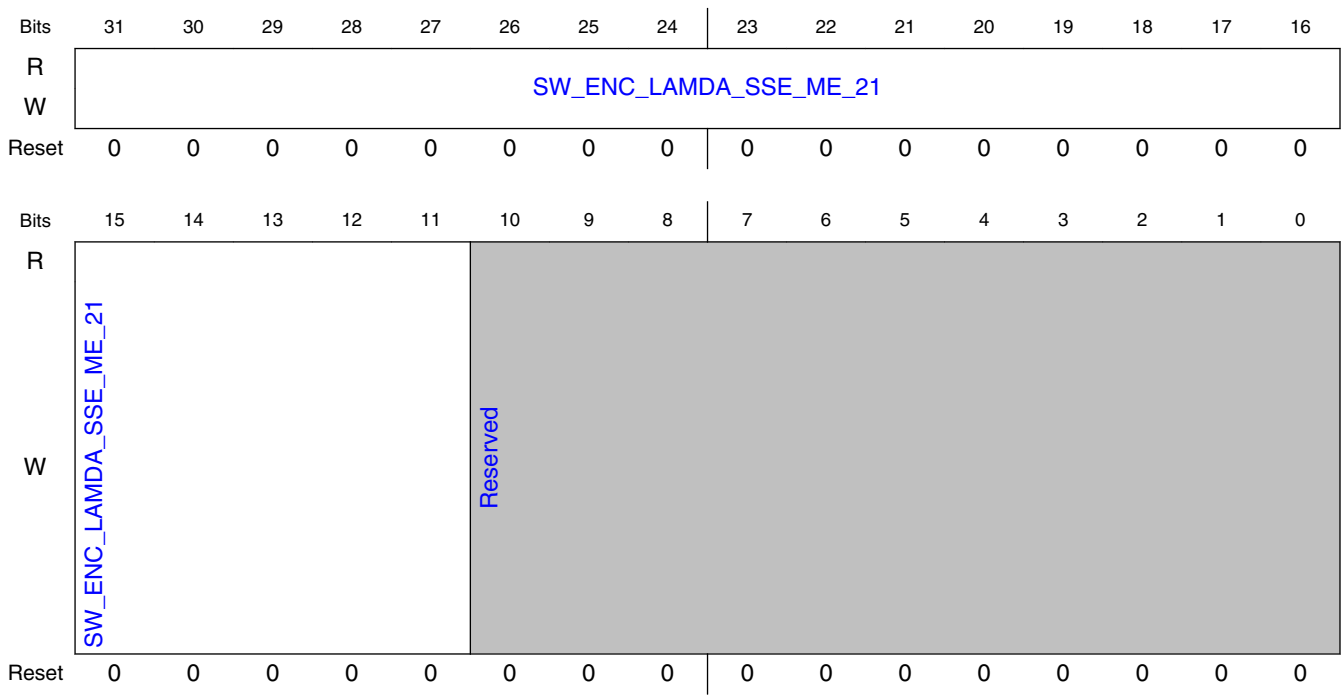
16.3.5.2.129.1 Offset

Register	Offset
swreg163	28Ch

16.3.5.2.129.2 Function

inter me SSE lambda config 21

16.3.5.2.129.3 Diagram



16.3.5.2.129.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_21	inter me SSE lambda 21
10-0 —	Reserved

16.3.5.2.130 inter me SSE lambda config 22 (swreg164)

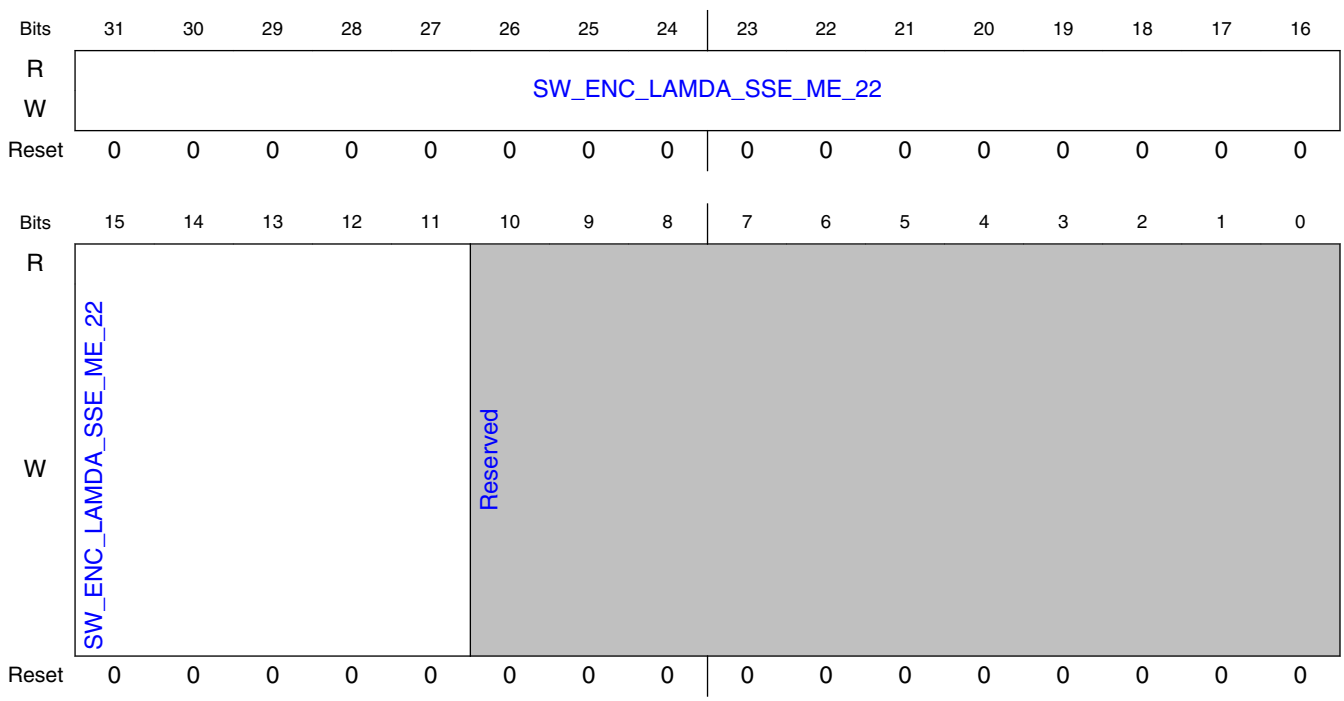
16.3.5.2.130.1 Offset

Register	Offset
swreg164	290h

16.3.5.2.130.2 Function

inter me SSE lambda config 22

16.3.5.2.130.3 Diagram



16.3.5.2.130.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_22	inter me SSE lambda 22
10-0 —	Reserved

16.3.5.2.131 inter me SSE lambda config 23 (swreg165)

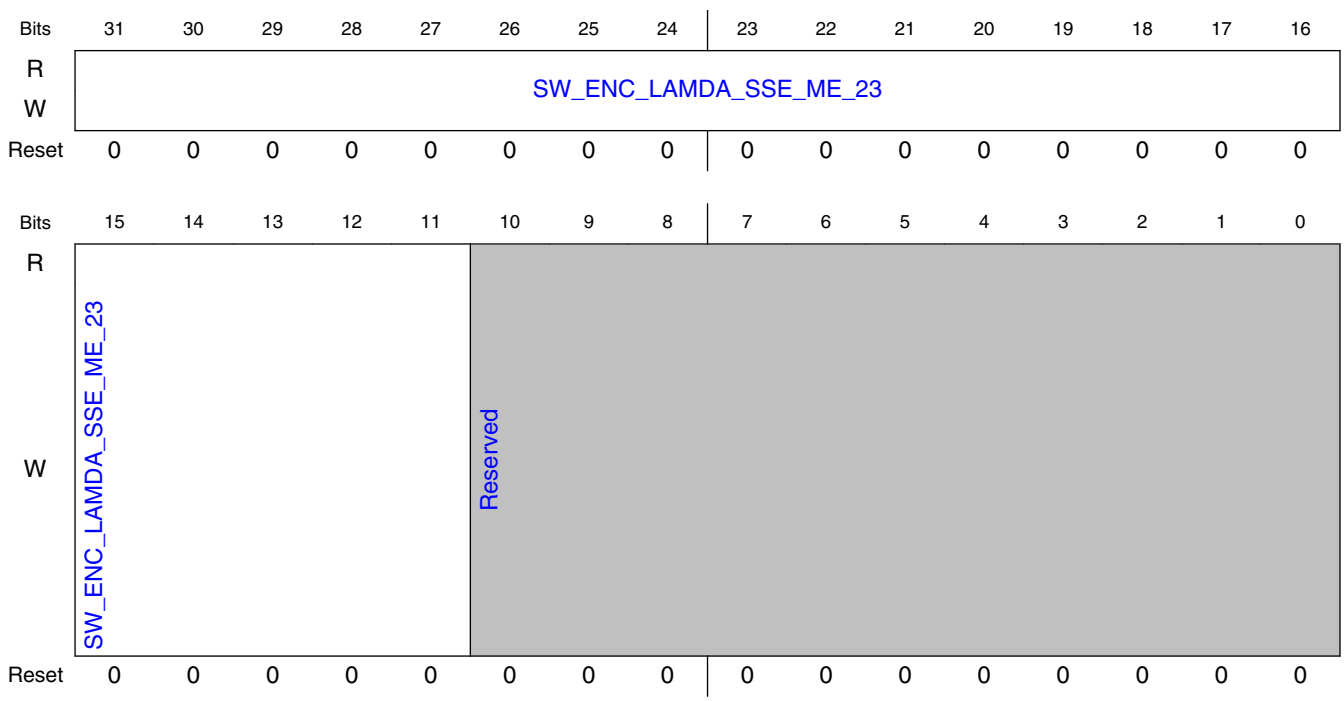
16.3.5.2.131.1 Offset

Register	Offset
swreg165	294h

16.3.5.2.131.2 Function

inter me SSE lambda config 23

16.3.5.2.131.3 Diagram



16.3.5.2.131.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_23	inter me SSE lambda 23
10-0 —	Reserved

16.3.5.2.132 inter me SSE lambda config 24 (swreg166)

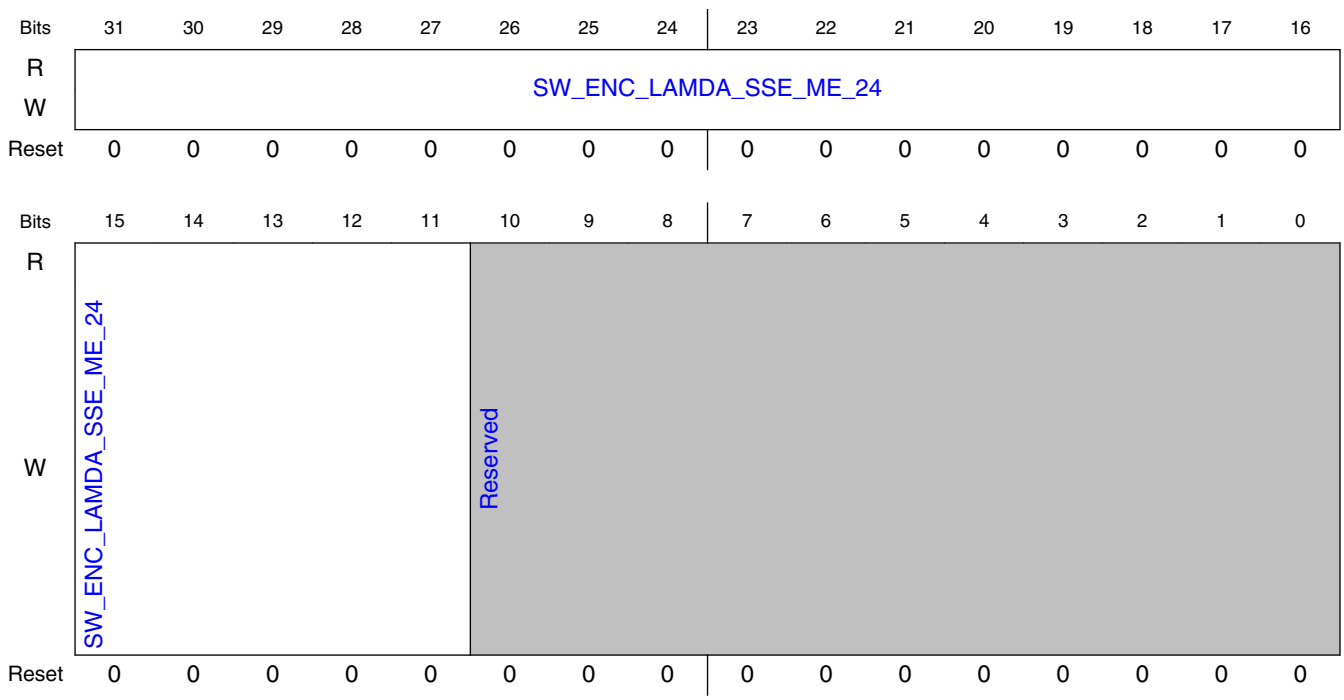
16.3.5.2.132.1 Offset

Register	Offset
swreg166	298h

16.3.5.2.132.2 Function

inter me SSE lambda config 24

16.3.5.2.132.3 Diagram



16.3.5.2.132.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_24	inter me SSE lambda 24

Table continues on the next page...

Field	Function
10-0 —	Reserved

16.3.5.2.133 inter me SSE lambda config 25 (swreg167)

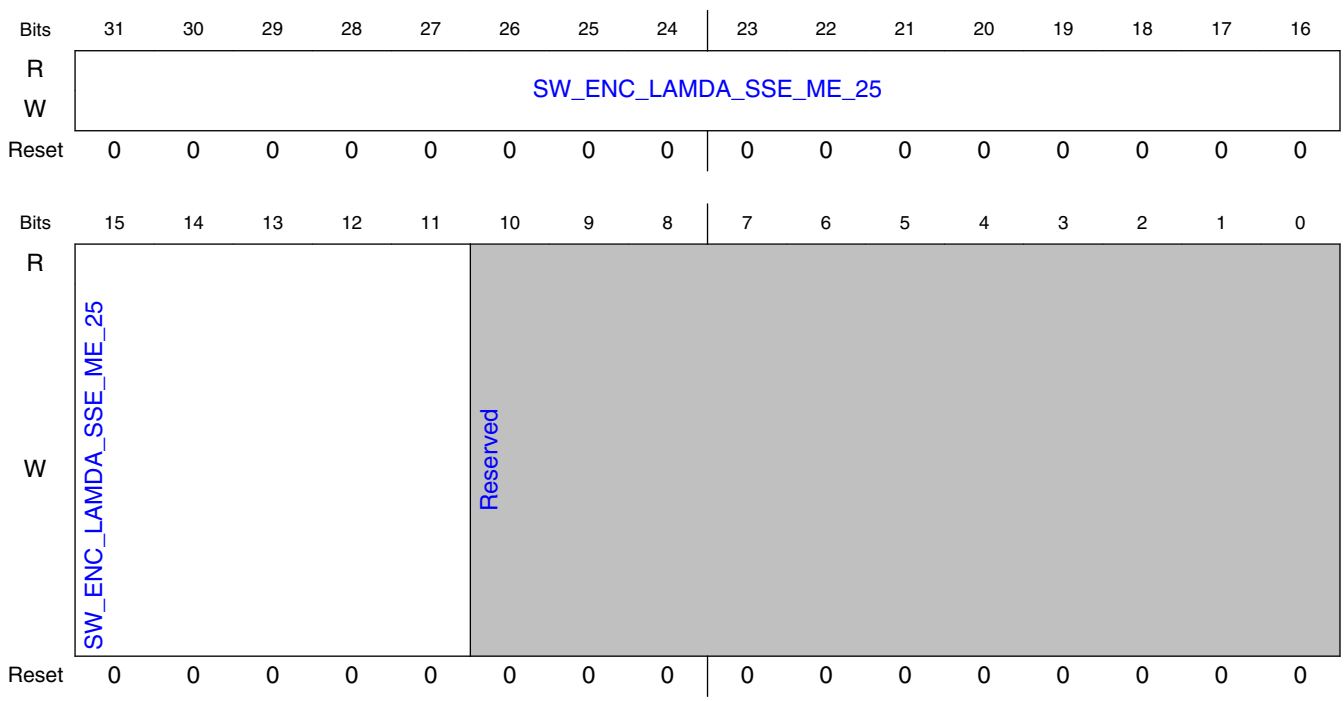
16.3.5.2.133.1 Offset

Register	Offset
swreg167	29Ch

16.3.5.2.133.2 Function

inter me SSE lambda config 25

16.3.5.2.133.3 Diagram



16.3.5.2.133.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_25	inter me SSE lambda 25
10-0 —	Reserved

16.3.5.2.134 inter me SSE lambda config 26 (swreg168)

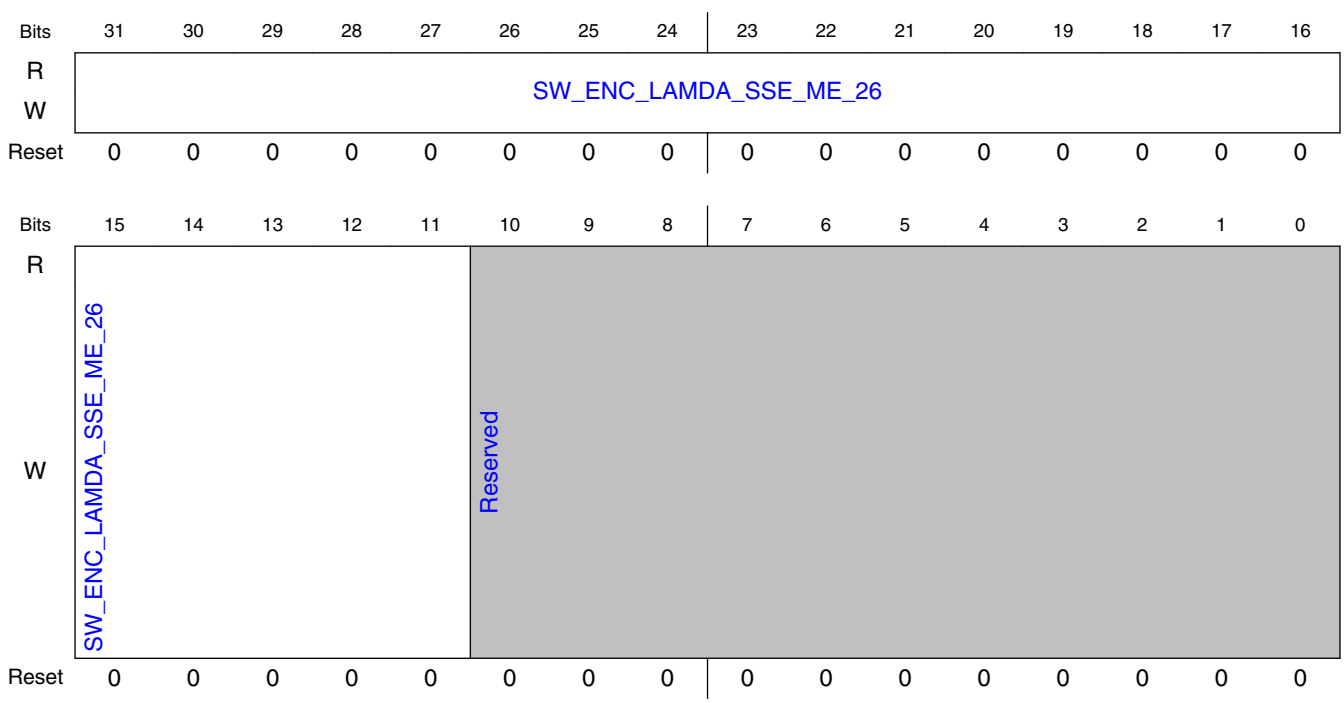
16.3.5.2.134.1 Offset

Register	Offset
swreg168	2A0h

16.3.5.2.134.2 Function

inter me SSE lambda config 26

16.3.5.2.134.3 Diagram



16.3.5.2.134.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_26	inter me SSE lambda 26
10-0 —	Reserved

16.3.5.2.135 inter me SSE lambda config 27 (swreg169)

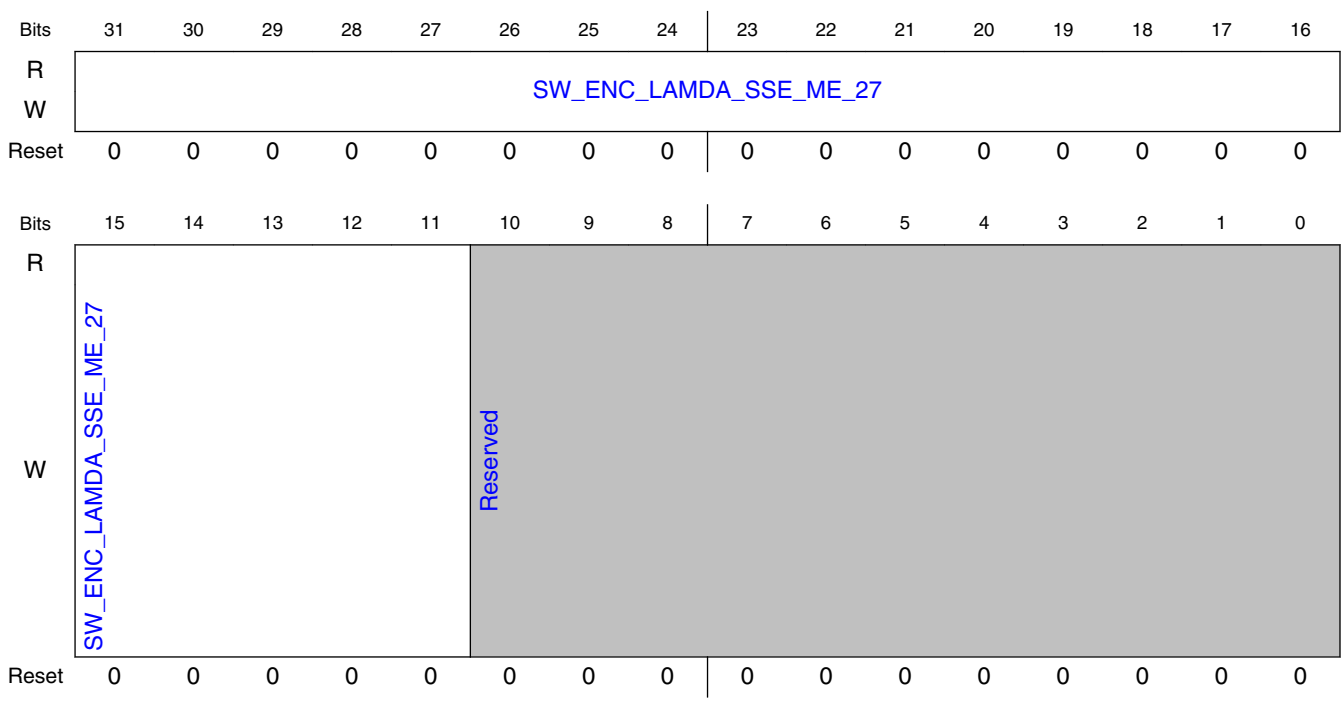
16.3.5.2.135.1 Offset

Register	Offset
swreg169	2A4h

16.3.5.2.135.2 Function

inter me SSE lambda config 27

16.3.5.2.135.3 Diagram



16.3.5.2.135.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_27	inter me SSE lambda 27
10-0 —	Reserved

16.3.5.2.136 inter me SSE lambda config 30 (swreg172)

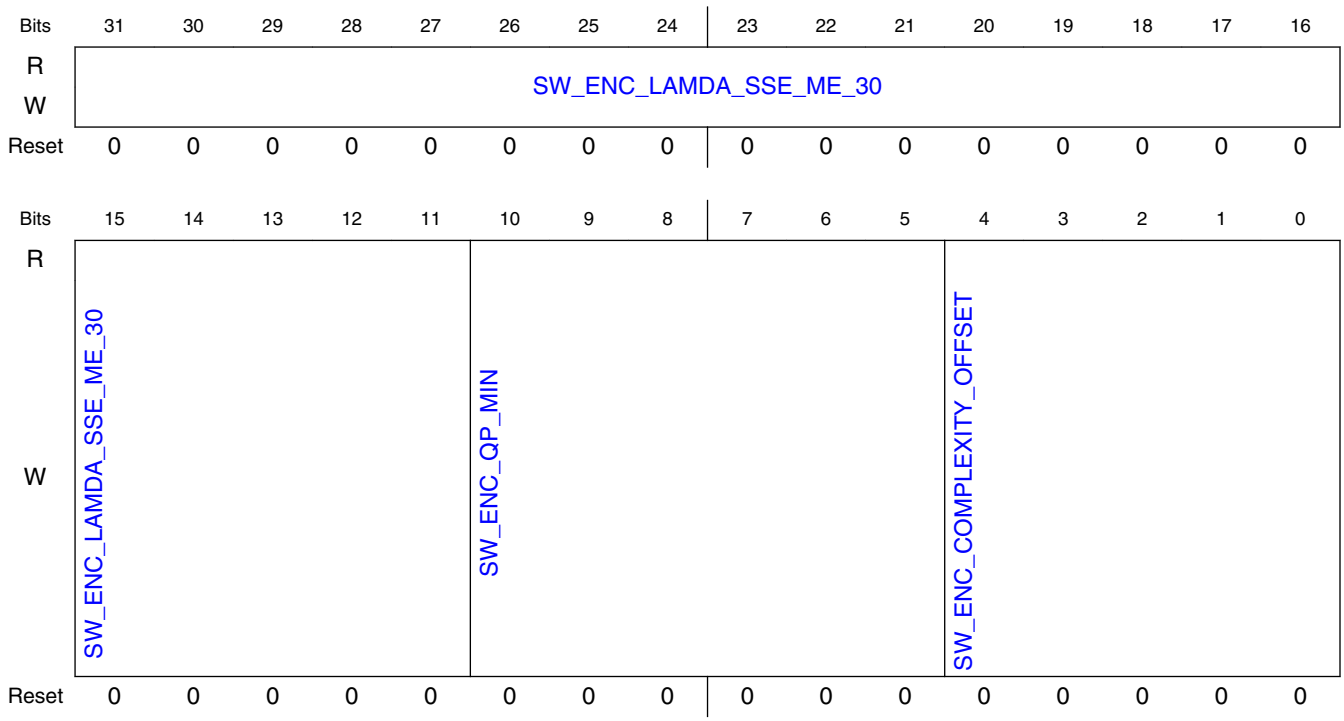
16.3.5.2.136.1 Offset

Register	Offset
swreg172	2B0h

16.3.5.2.136.2 Function

inter me SSE lambda config 30

16.3.5.2.136.3 Diagram



16.3.5.2.136.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_30	inter me SSE lambda 30
10-5 SW_ENC_QP_MIN	min value of qp
4-0 SW_ENC_COMPLEXITY_OFFSET	block rc complexity offset

16.3.5.2.137 inter me SSE lambda config 31 (swreg173)

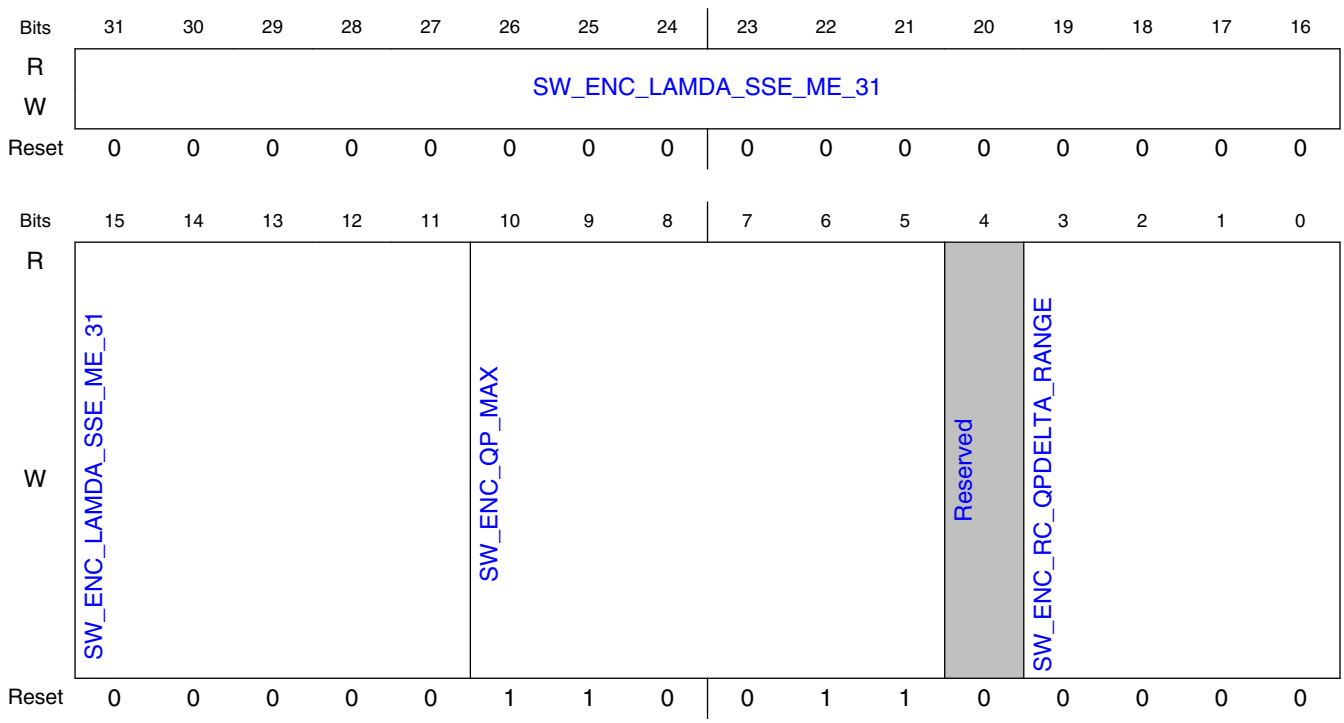
16.3.5.2.137.1 Offset

Register	Offset
swreg173	2B4h

16.3.5.2.137.2 Function

inter me SSE lambda config 31

16.3.5.2.137.3 Diagram



16.3.5.2.137.4 Fields

Field	Function
31-11 SW_ENC_LAMDA_SSE_ME_31	inter me SSE lambda 31
10-5 SW_ENC_QP_MAX	max value of qp
4 —	Reserved
3-0 SW_ENC_RC_QPDELTA_RANGE	rc qp delta range

16.3.5.2.138 intra SATD lambda config 8 (swreg174)

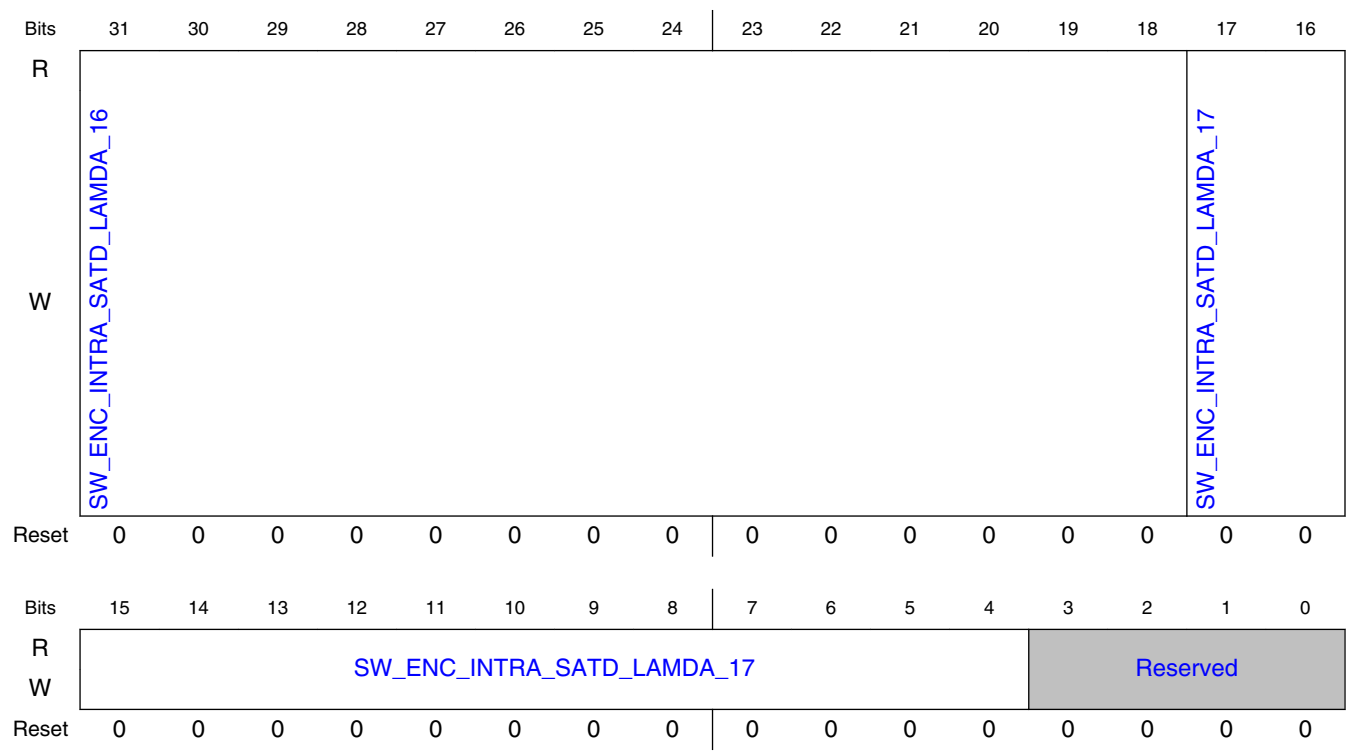
16.3.5.2.138.1 Offset

Register	Offset
swreg174	2B8h

16.3.5.2.138.2 Function

intra SATD lambda config 8

16.3.5.2.138.3 Diagram



16.3.5.2.138.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_16	intra SATD lambda 16

Table continues on the next page...

Field	Function
17-4 SW_ENC_INTRA_SATD_LAMDA_17	intra SATD lambda 17
3-0 —	Reserved

16.3.5.2.139 intra SATD lambda config 9 (swreg175)

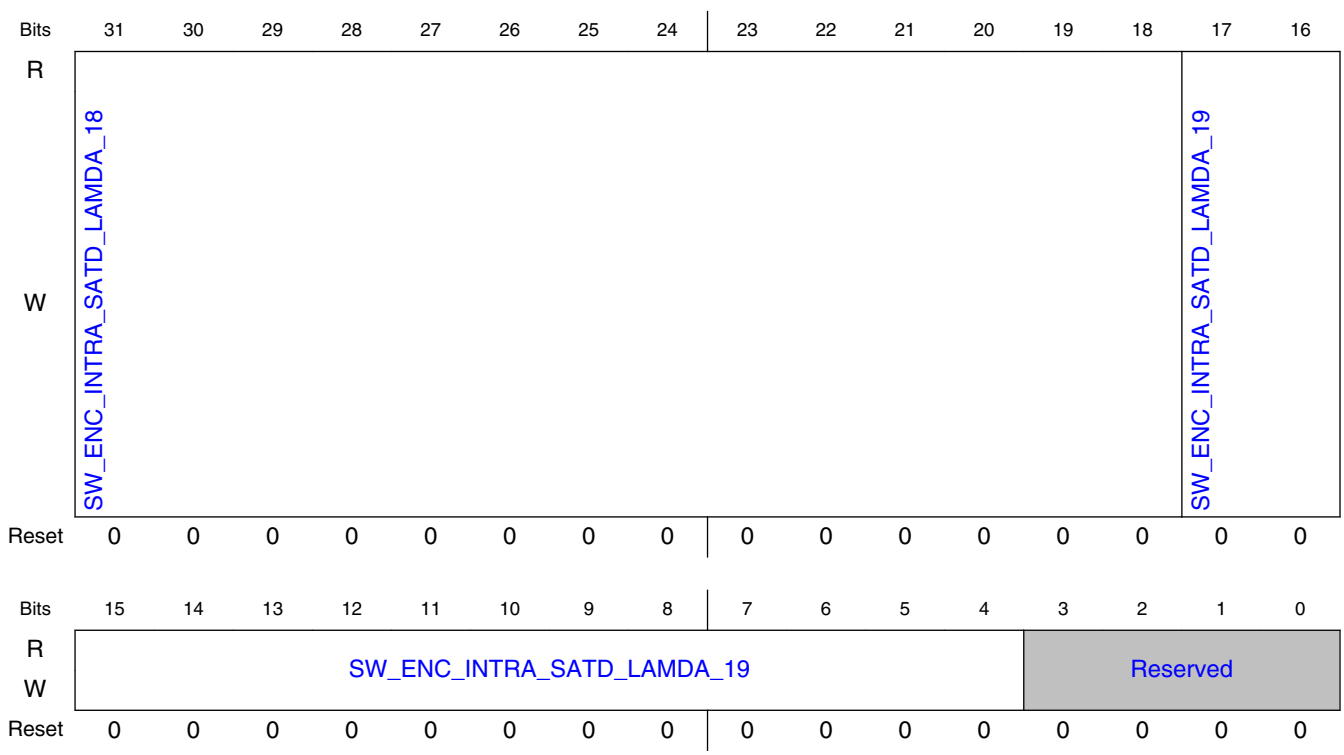
16.3.5.2.139.1 Offset

Register	Offset
swreg175	2BCh

16.3.5.2.139.2 Function

intra SATD lambda config 9

16.3.5.2.139.3 Diagram



16.3.5.2.139.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_18	intra SATD lambda 18
17-4 SW_ENC_INTRA_SATD_LAMDA_19	intra SATD lambda 19
3-0 —	Reserved

16.3.5.2.140 intra SATD lambda config 10 (swreg176)

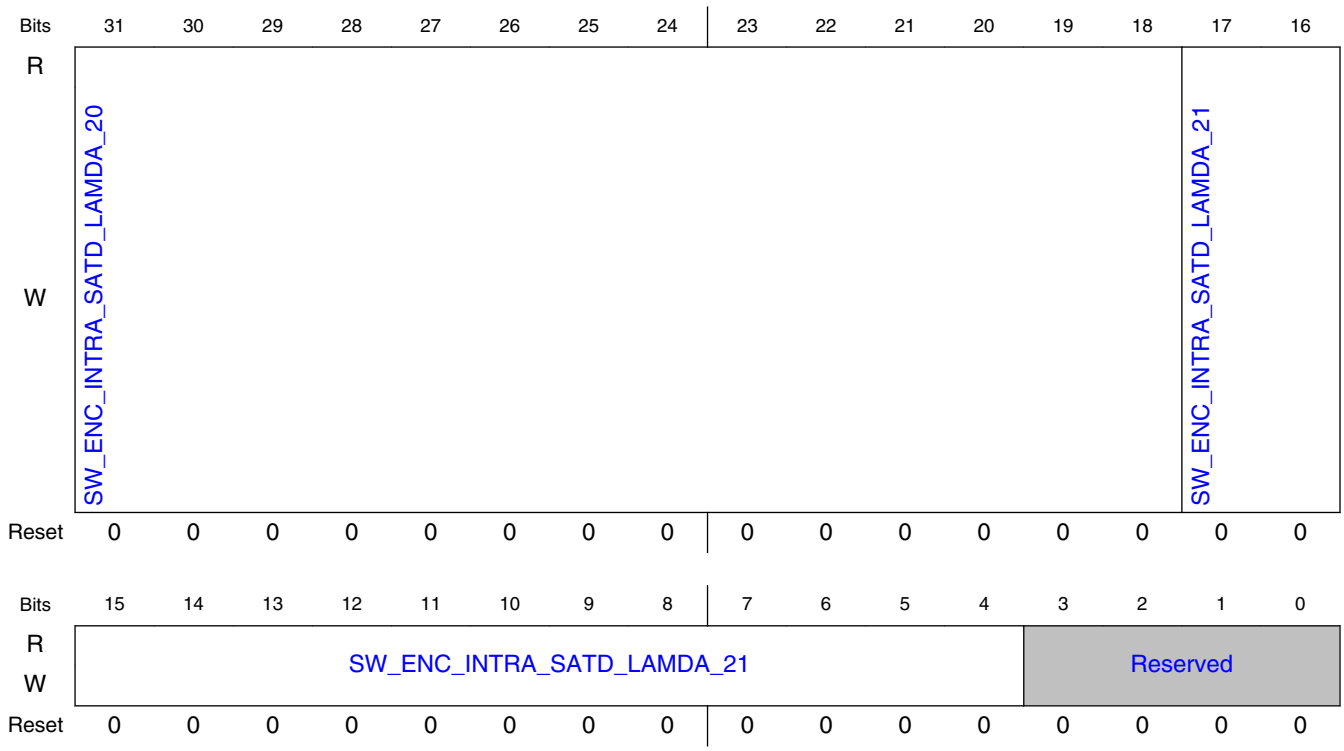
16.3.5.2.140.1 Offset

Register	Offset
swreg176	2C0h

16.3.5.2.140.2 Function

intra SATD lambda config 10

16.3.5.2.140.3 Diagram



16.3.5.2.140.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_20	intra SATD lambda 20
17-4 SW_ENC_INTRA_SATD_LAMDA_21	intra SATD lambda 21
3-0 —	Reserved

16.3.5.2.141 intra SATD lambda config 11 (swreg177)

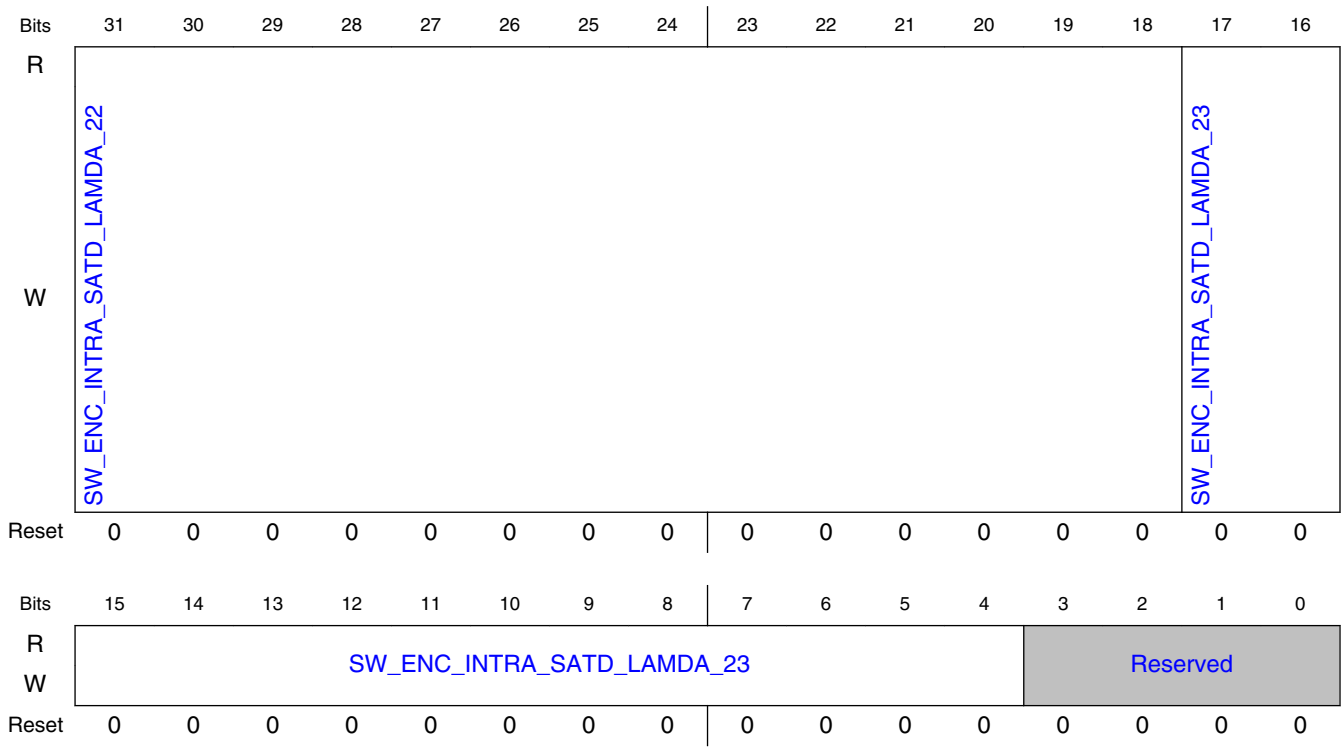
16.3.5.2.141.1 Offset

Register	Offset
swreg177	2C4h

16.3.5.2.141.2 Function

intra SATD lambda config 11

16.3.5.2.141.3 Diagram



16.3.5.2.141.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_22	intra SATD lambda 22
17-4 SW_ENC_INTRA_SATD_LAMDA_23	intra SATD lambda 23
3-0 —	Reserved

16.3.5.2.142 intra SATD lambda config 12 (swreg178)

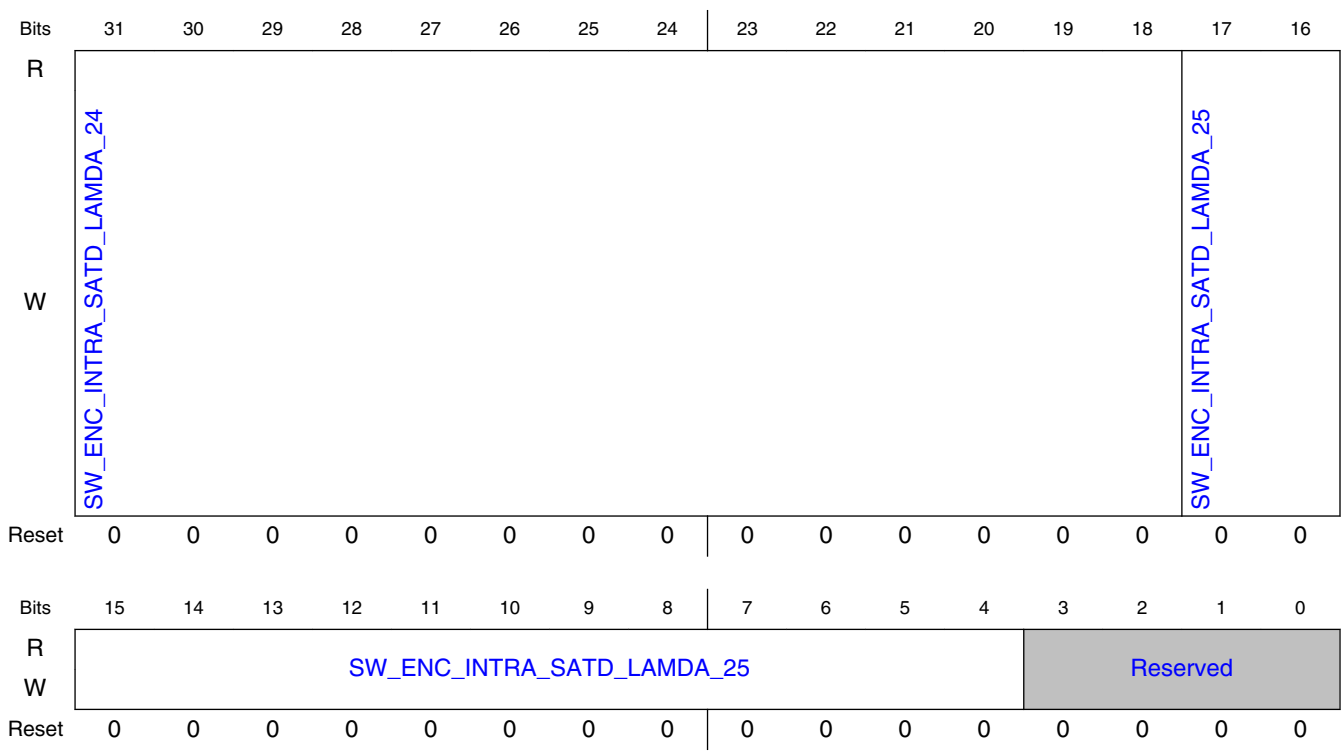
16.3.5.2.142.1 Offset

Register	Offset
swreg178	2C8h

16.3.5.2.142.2 Function

intra SATD lambda config 12

16.3.5.2.142.3 Diagram



16.3.5.2.142.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_24	intra SATD lambda 24
17-4	intra SATD lambda 25

Table continues on the next page...

Field	Function
SW_ENC_INTRA_SATD_LAMDA_25	
3-0 —	Reserved

16.3.5.2.143 intra SATD lambda config 13 (swreg179)

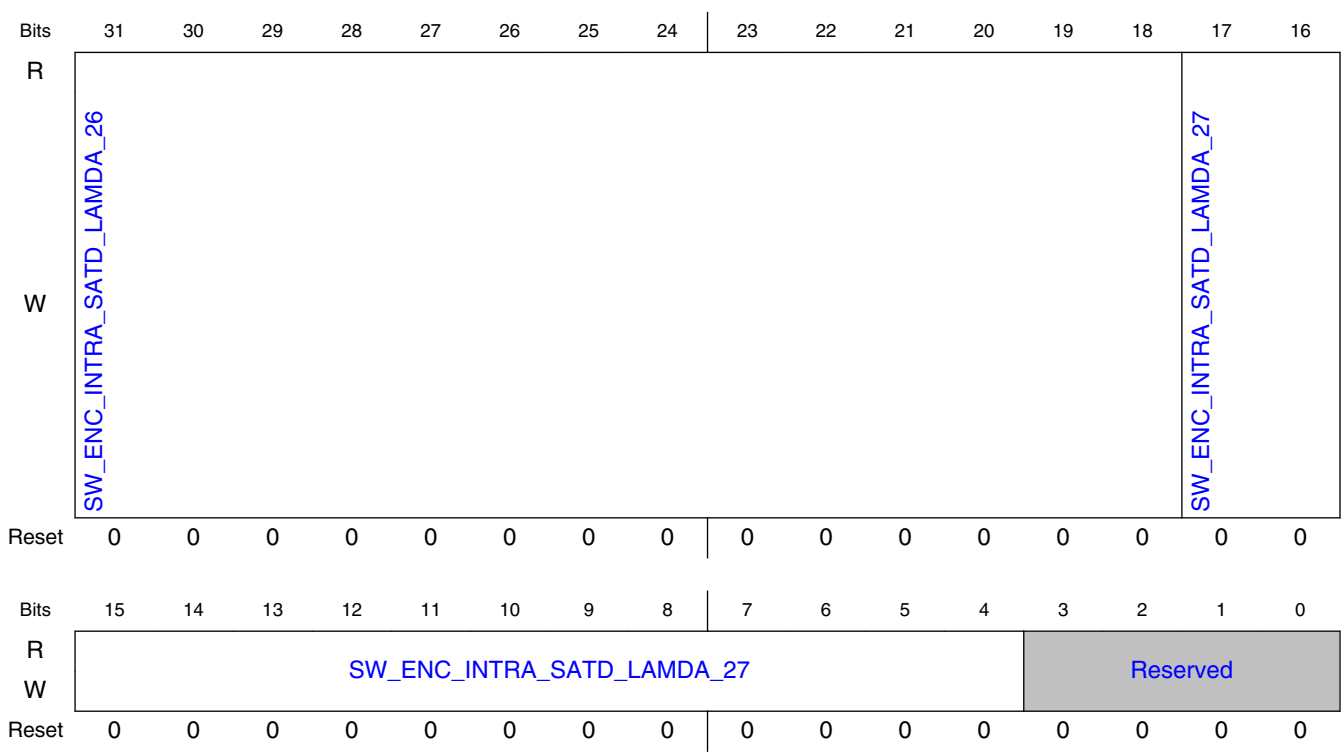
16.3.5.2.143.1 Offset

Register	Offset
swreg179	2CCh

16.3.5.2.143.2 Function

intra SATD lambda config 13

16.3.5.2.143.3 Diagram



16.3.5.2.143.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_26	intra SATD lambda 26
17-4 SW_ENC_INTRA_SATD_LAMDA_27	intra SATD lambda 27
3-0 —	Reserved

16.3.5.2.144 intra SATD lambda config 14 (swreg180)

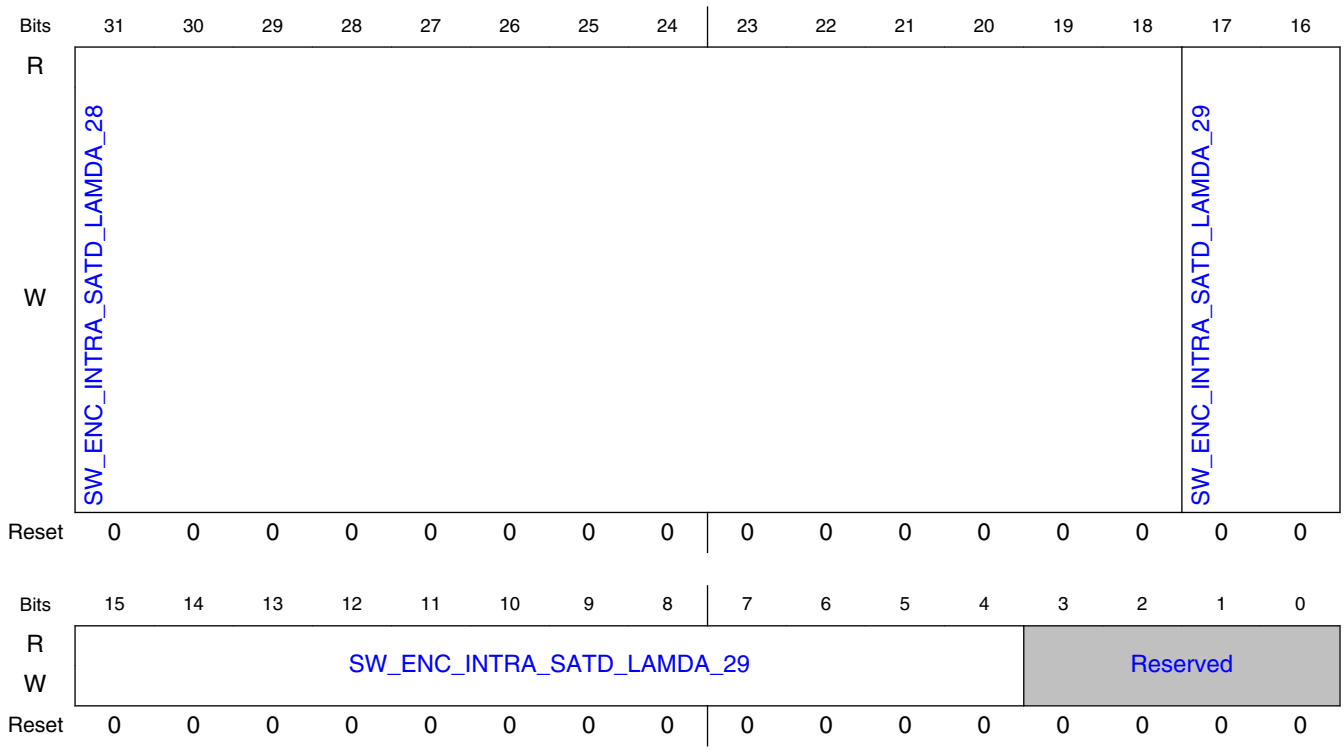
16.3.5.2.144.1 Offset

Register	Offset
swreg180	2D0h

16.3.5.2.144.2 Function

intra SATD lambda config 14

16.3.5.2.144.3 Diagram



16.3.5.2.144.4 Fields

Field	Function
31-18 SW_ENC_INTRA_SATD_LAMDA_28	intra SATD lambda 28
17-4 SW_ENC_INTRA_SATD_LAMDA_29	intra SATD lambda 29
3-0 —	Reserved

16.3.5.2.145 intra SATD lambda config 15 (swreg181)

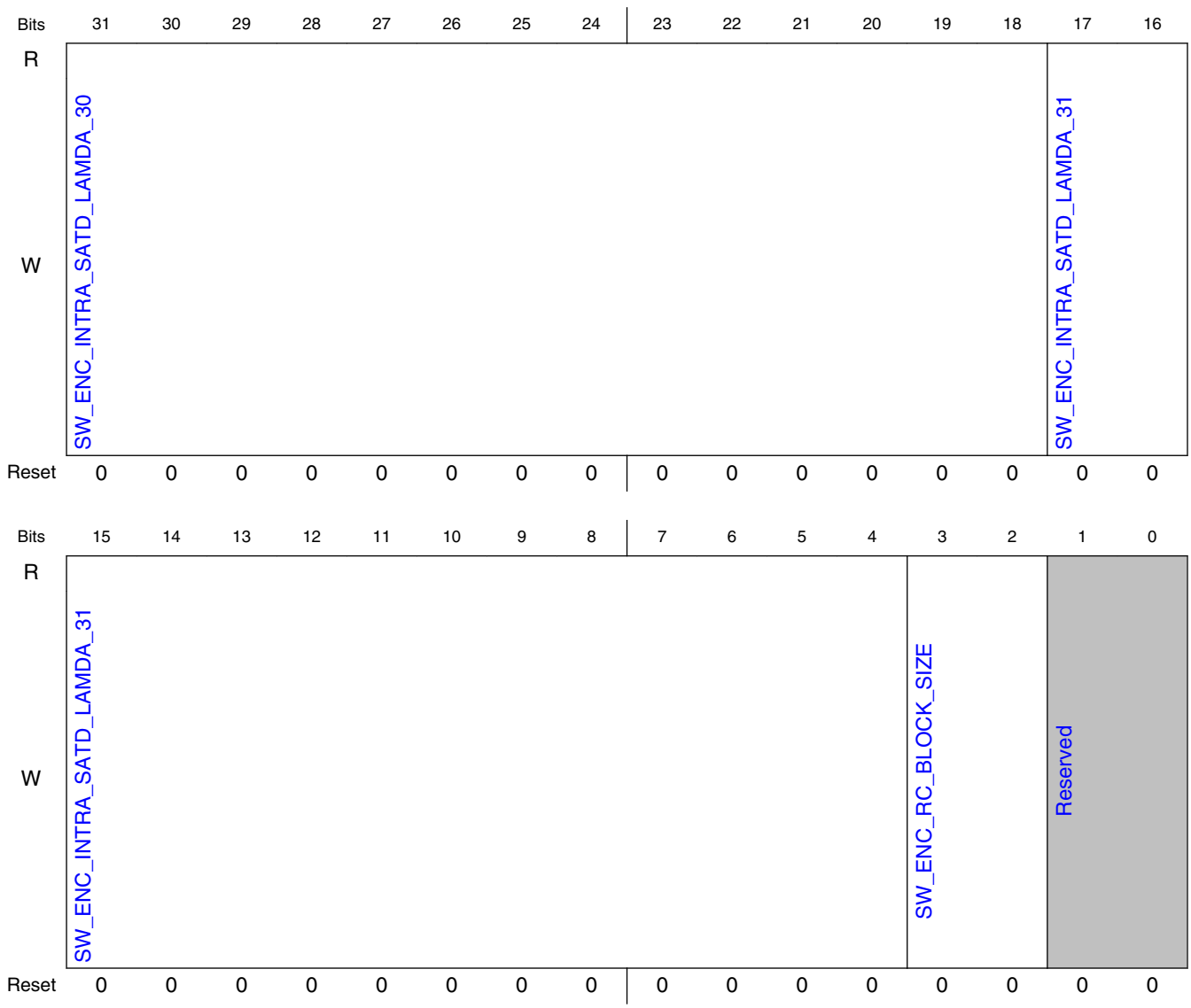
16.3.5.2.145.1 Offset

Register	Offset
swreg181	2D4h

16.3.5.2.145.2 Function

intra SATD lambda config 15

16.3.5.2.145.3 Diagram



16.3.5.2.145.4 Fields

Field	Function
31-18	intra SATD lambda 30

Table continues on the next page...

Field	Function
SW_ENC_INTRA_SATD_LAMDA_30	
17-4 SW_ENC_INTRA_SATD_LAMDA_31	intra SATD lambda 31
3-2 SW_ENC_RC_BLOCK_SIZE	block rc block size 00b - 64x64. 01b - 32x32. 10b - 16x16
1-0 —	Reserved

16.3.5.2.146 qp fractional part (swreg182)

16.3.5.2.146.1 Offset

Register	Offset
swreg182	2D8h

16.3.5.2.146.2 Function

qp fractional part

16.3.5.2.146.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_QP_FRACTIONAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_QP_DELTA_GAIN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.146.4 Fields

Field	Function
31-16 SW_ENC_QP_FRACTIONAL	qp fractional part
15-0 SW_ENC_QP_DELTA_GAIN	qp delta gain

16.3.5.2.147 qp sum (swreg183)

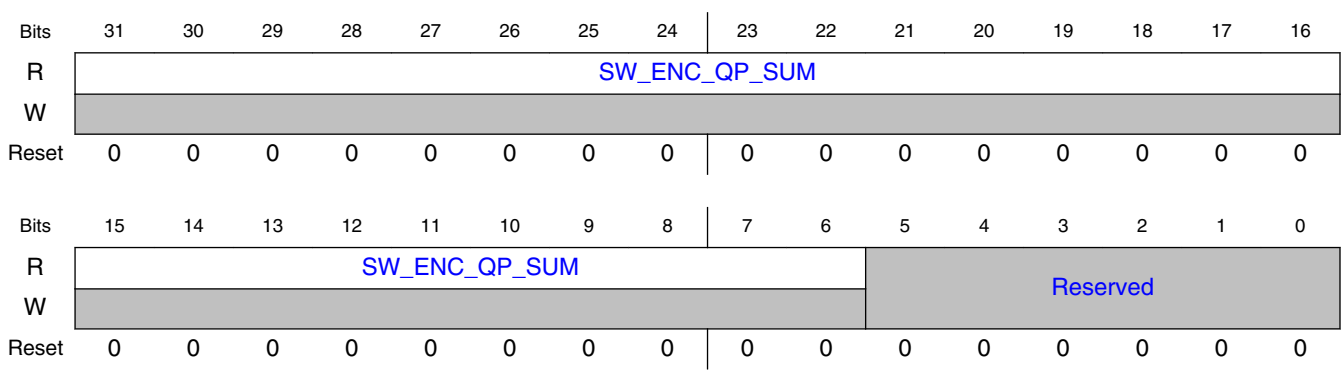
16.3.5.2.147.1 Offset

Register	Offset
swreg183	2DCh

16.3.5.2.147.2 Function

qp sum

16.3.5.2.147.3 Diagram



16.3.5.2.147.4 Fields

Field	Function
31-6 SW_ENC_QP_SUM	Block qp sum

Table continues on the next page...

Field	Function
5-0 —	Reserved

16.3.5.2.148 qp num (swreg184)

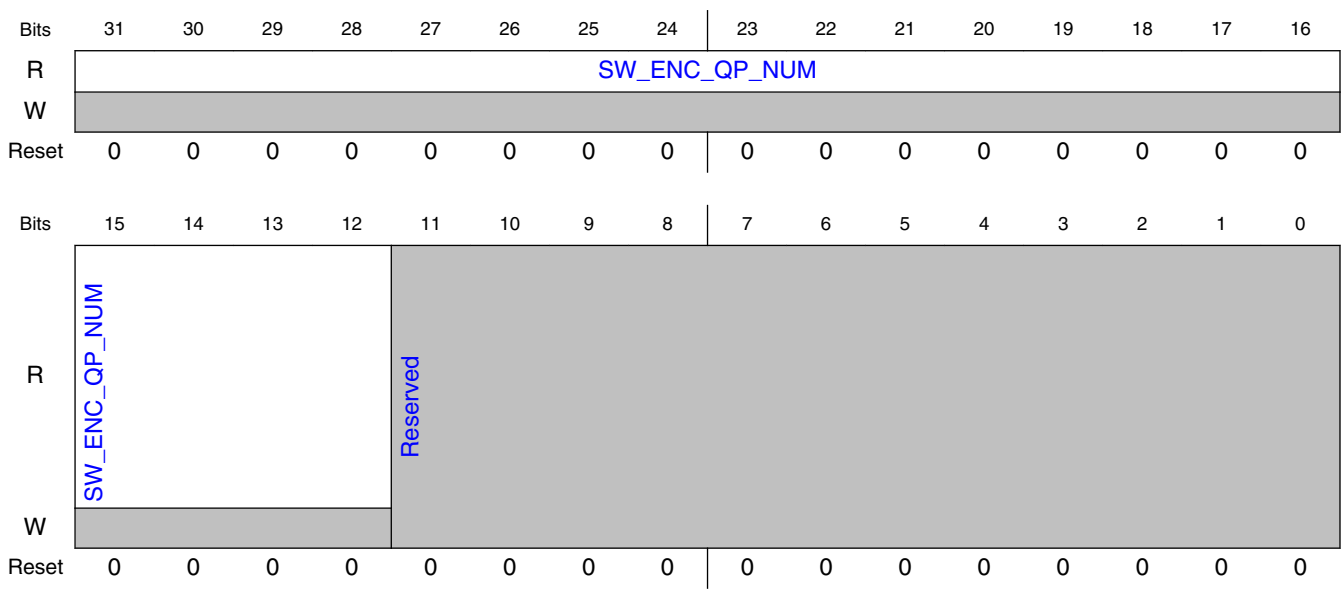
16.3.5.2.148.1 Offset

Register	Offset
swreg184	2E0h

16.3.5.2.148.2 Function

qp num

16.3.5.2.148.3 Diagram



16.3.5.2.148.4 Fields

Field	Function
31-12 SW_ENC_QP_NUM	Block qp number

Table continues on the next page...

Field	Function
11-0 —	Reserved

16.3.5.2.149 picture complexity. Timeout cycles MSB. (swreg185)

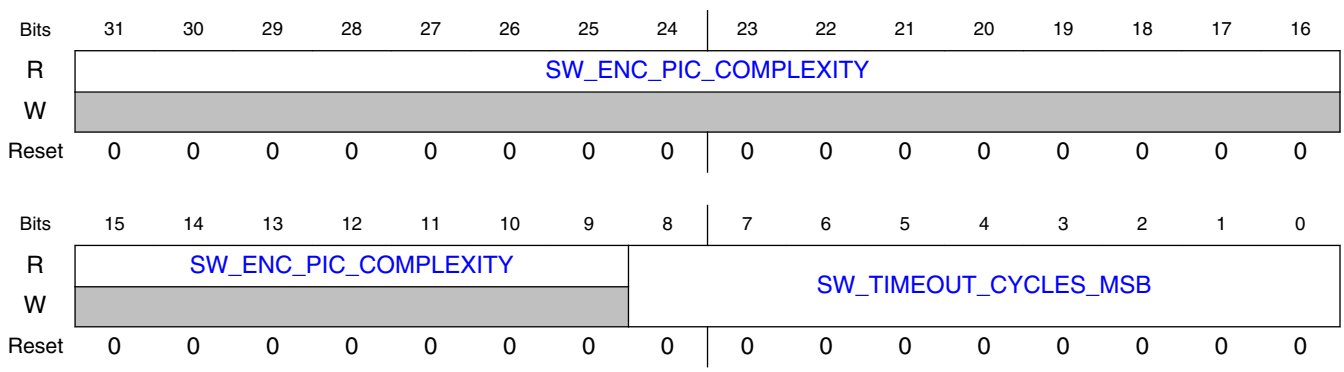
16.3.5.2.149.1 Offset

Register	Offset
swreg185	2E4h

16.3.5.2.149.2 Function

picture complexity. Timeout cycles MSB.

16.3.5.2.149.3 Diagram



16.3.5.2.149.4 Fields

Field	Function
31-9 SW_ENC_PIC_COMPLEXITY	Picture complexity
8-0 SW_TIMEOUT_CYCLES_MSB	Extend swreg81. Timeout cycles number. Bits 31:23.

16.3.5.2.150 Base address for CU information table LSB (swreg186)

16.3.5.2.150.1 Offset

Register	Offset
swreg186	2E8h

16.3.5.2.150.2 Function

Base address for CU information table LSB

16.3.5.2.150.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_CU_INFORMATION_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_CU_INFORMATION_TABLE_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.150.4 Fields

Field	Function
31-0 SW_ENC_CU_INFORMATION_TABLE_BASE	Base address for CU information table LSB

16.3.5.2.151 Base address for CU information LSB (swreg188)

16.3.5.2.151.1 Offset

Register	Offset
swreg188	2F0h

16.3.5.2.151.2 Function

Base address for CU information LSB

16.3.5.2.151.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_CU_INFORMATION_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_CU_INFORMATION_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.151.4 Fields

Field	Function
31-0 SW_ENC_CU_INFORMATION_BASE	Base address for CU information LSB

16.3.5.2.152 Long-term reference pictures config (swreg190)

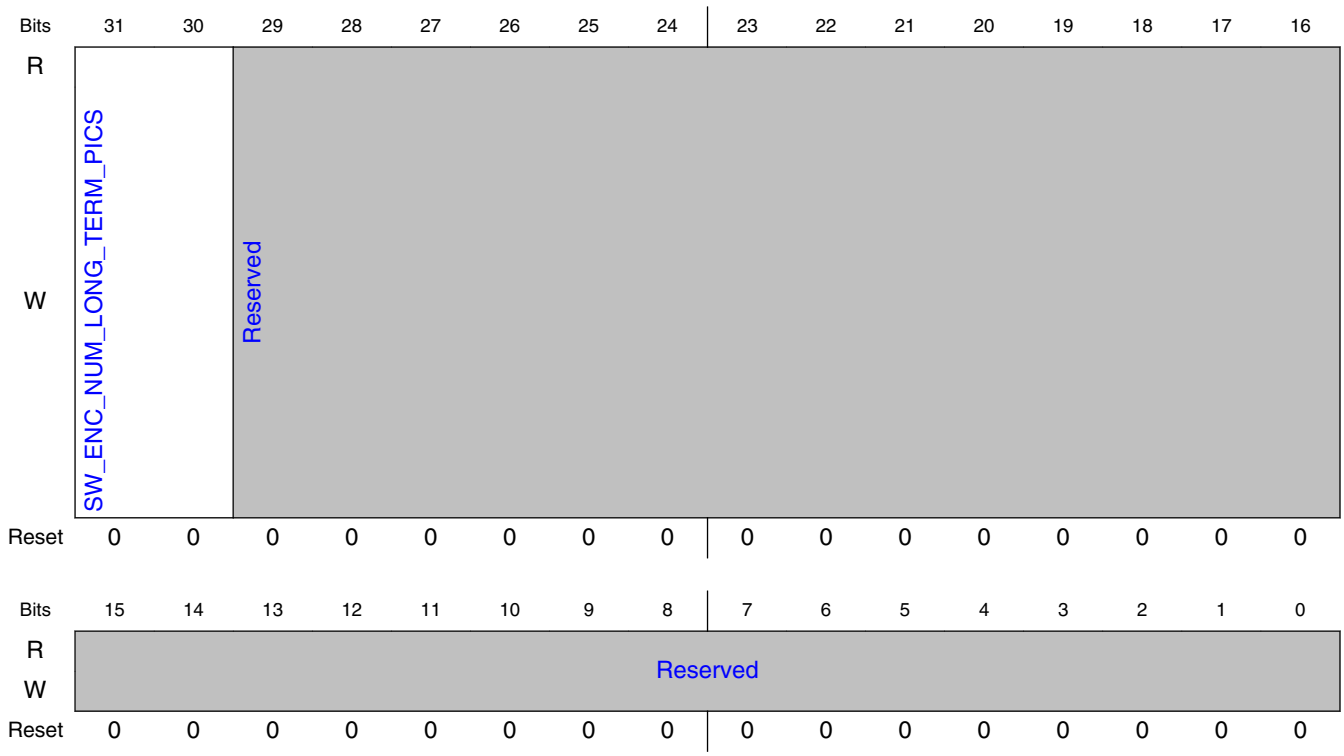
16.3.5.2.152.1 Offset

Register	Offset
swreg190	2F8h

16.3.5.2.152.2 Function

Long-term reference pictures config

16.3.5.2.152.3 Diagram



16.3.5.2.152.4 Fields

Field	Function
31-30 SW_ENC_NUM_LONG_TERM_PICS	Number of long-term reference pictures directly signaled in the slice header
29-0 —	Reserved

16.3.5.2.153 Temporal scalable config (swreg191)

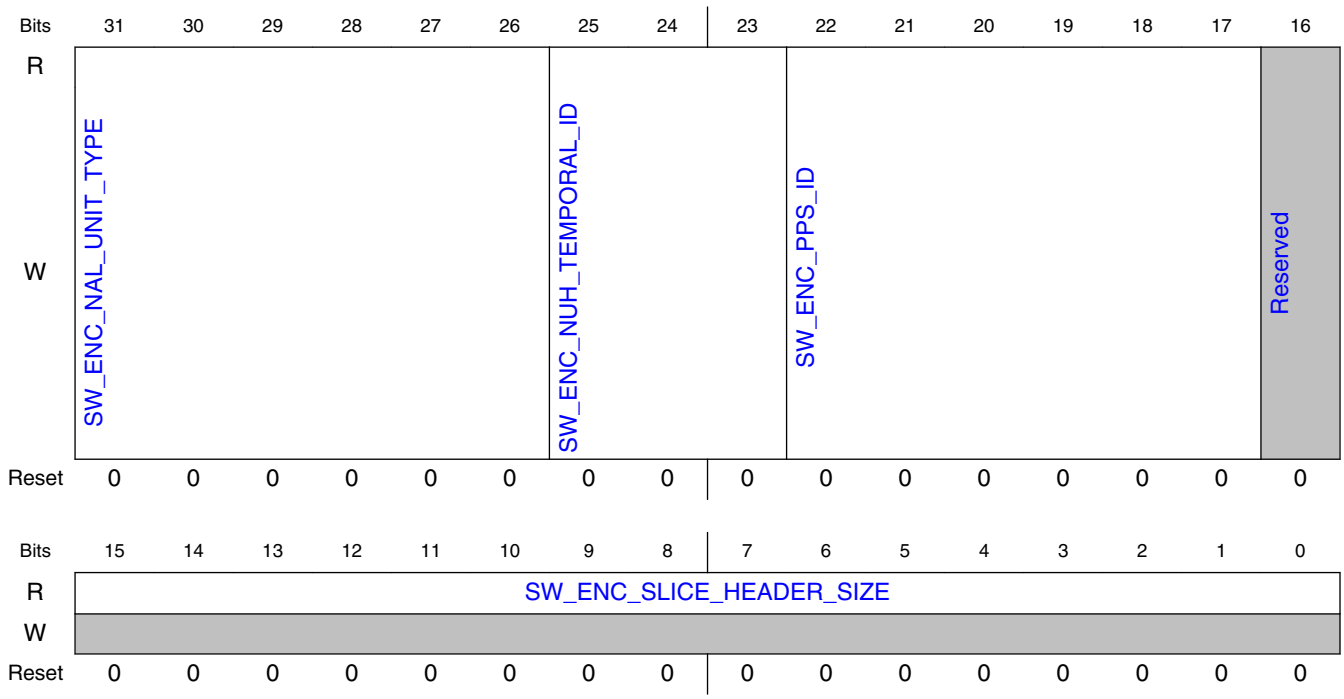
16.3.5.2.153.1 Offset

Register	Offset
swreg191	2FCh

16.3.5.2.153.2 Function

Temporal scalable config

16.3.5.2.153.3 Diagram



16.3.5.2.153.4 Fields

Field	Function
31-26 SW_ENC_NAL_UNIT_TYPE	NAL unit type
25-23 SW_ENC_NUH_TEMPORAL_ID	NAL temporal id
22-17 SW_ENC_PPS_ID	PPS id
16 —	Reserved
15-0 SW_ENC_SLICE_HEADER_SIZE	Slice header size

16.3.5.2.154 register extension for ctu_size=16 (swreg195)

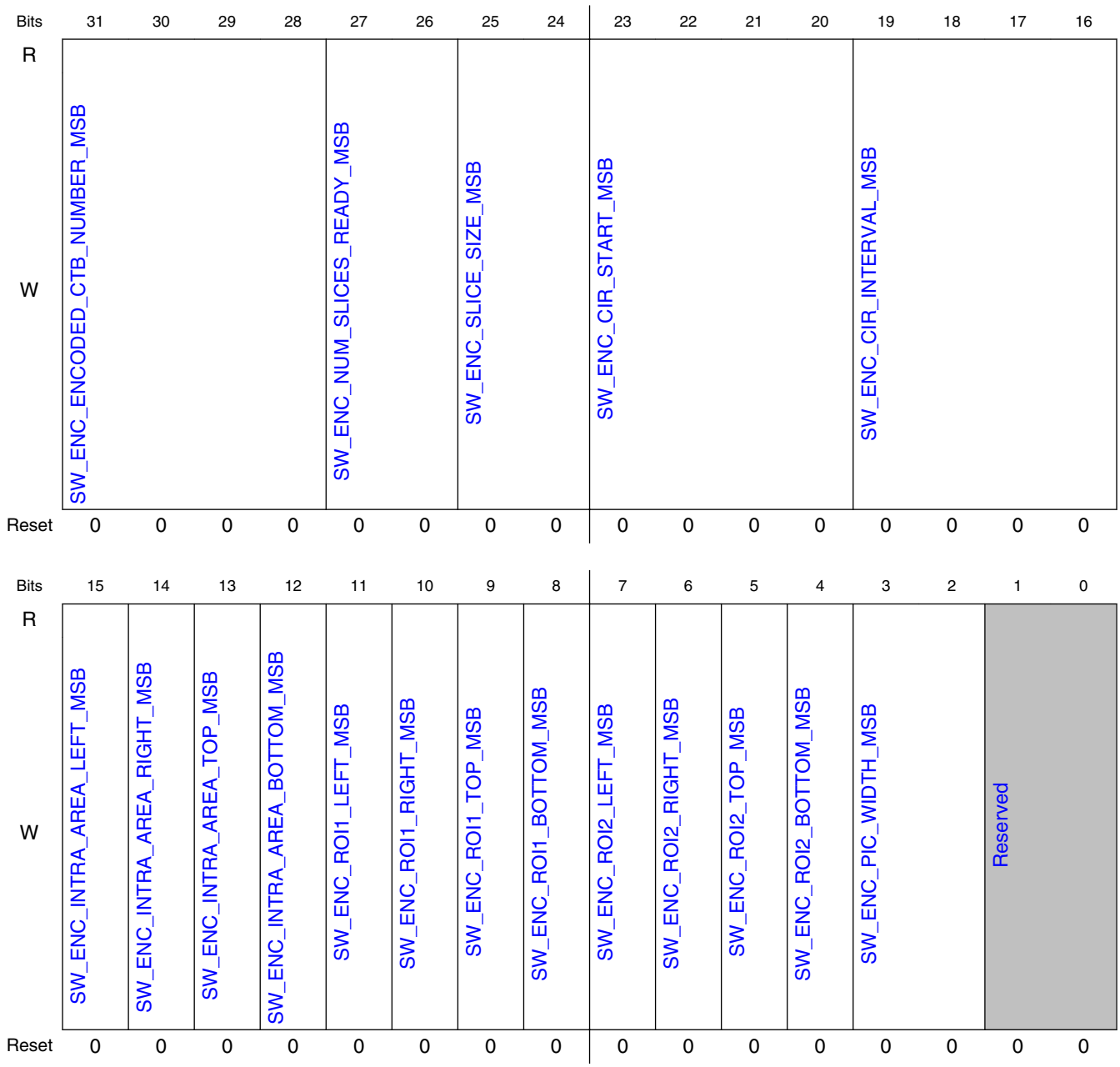
16.3.5.2.154.1 Offset

Register	Offset
swreg195	30Ch

16.3.5.2.154.2 Function

register extension for ctu_size=16

16.3.5.2.154.3 Diagram



16.3.5.2.154.4 Fields

Field	Function
31-28 SW_ENC_ENCODED_CTB_NUMBER_MSB	MB count output. Max 512*256. bit[13:16]
27-26	HEVC amount of completed slices. bit[8:9]

Table continues on the next page...

Field	Function
SW_ENC_NUM_SLICES_READY_MSB	
25-24 SW_ENC_SLICE_SIZE_MSB	slice size in ctu row. bit[7:8]
23-20 SW_ENC_CIR_START_MSB	cir start. bit[14:17]
19-16 SW_ENC_CIR_INTERVAL_MSB	Cyclic Intra Refresh. bit[14:17]
15 SW_ENC_INTRA_AREA_LEFT_MSB	intra Area Left. bit[8]
14 SW_ENC_INTRA_AREA_RIGHT_MSB	intra Area Right. bit[8]
13 SW_ENC_INTRA_AREA_TOP_MSB	intra Area Top. bit[8]
12 SW_ENC_INTRA_AREA_BOTTOM_MSB	intra Area Bottom. bit[8]
11 SW_ENC_ROI1_LEFT_MSB	ROI1 Area Left. bit[8]
10 SW_ENC_ROI1_RIGHT_MSB	ROI1 Area Right. bit[8]
9 SW_ENC_ROI1_TOP_MSB	ROI1 Area Top. bit[8]
8 SW_ENC_ROI1_BOTTOM_MSB	ROI1 Area Bottom. bit[8]
7 SW_ENC_ROI2_LEFT_MSB	ROI2 Area Left. bit[8]
6	ROI2 Area Right. bit[8]

Table continues on the next page...

Field	Function
SW_ENC_ROI2_RIGHT_MSB	
5 SW_ENC_ROI2_TOP_MSB	ROI2 Area Top. bit[8]
4 SW_ENC_ROI2_BOTTOM_MSB	ROI2 Area Bottom. bit[8]
3-2 SW_ENC_PIC_WIDTH_MSB	Encoded width. lumWidthMsb(unit 8 pixels). bit[11:10]
1-0 —	Reserved

16.3.5.2.155 Low Latency Controls (swreg196)

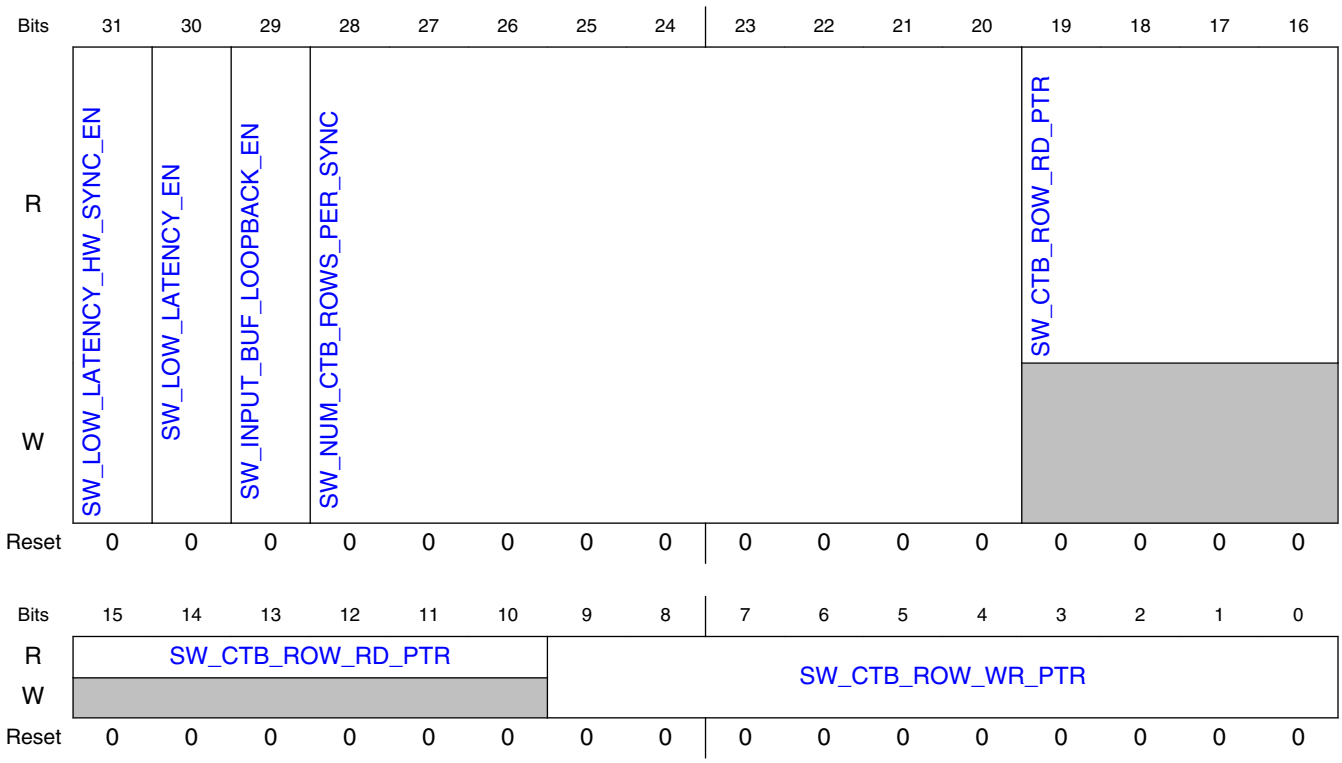
16.3.5.2.155.1 Offset

Register	Offset
swreg196	310h

16.3.5.2.155.2 Function

Low Latency Controls

16.3.5.2.155.3 Diagram



16.3.5.2.155.4 Fields

Field	Function
31 SW_LOW_LATENCY_HW_SYNC_EN	Low Latency Hardware Interface Enable. Use hardware handshaking.
30 SW_LOW_LATENCY_EN	Low Latency Enable
29 SW_INPUT_BUF_LOOPBACK_EN	Input buffer loopback Enable
28-20 SW_NUM_CTB_ROWS_PER_SYNC	Number of CTB rows for every HW sync. Unit is CTB
19-10 SW_CTB_ROW_RD_PTR	The number of CTB rows that has been fetched from Input buffer by encoder
9-0	The number of CTB rows that has been filled into the input buffer by external application

Field	Function
SW_CTB_ROW _WR_PTR	

16.3.5.2.156 Delta POC extension (swreg197)

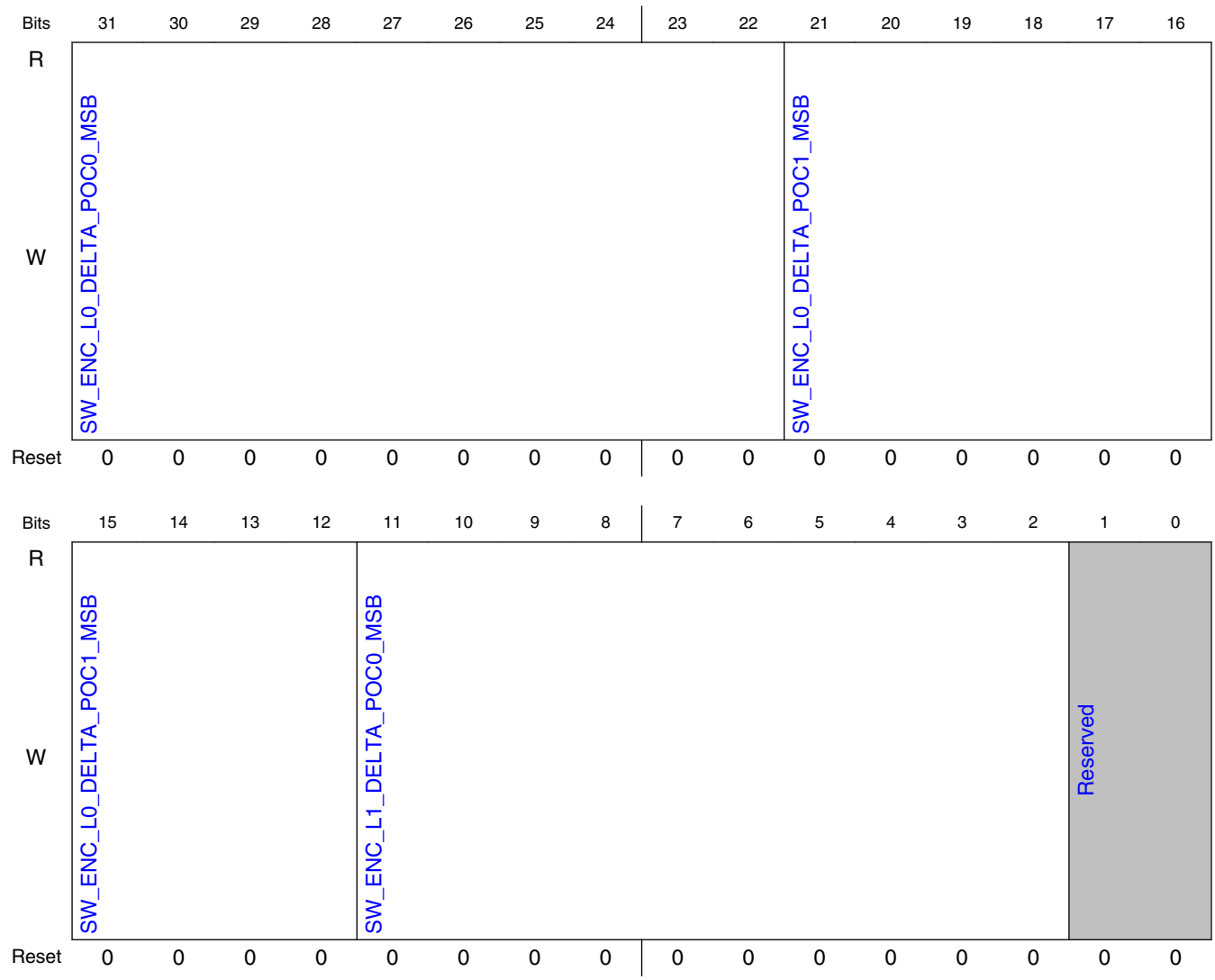
16.3.5.2.156.1 Offset

Register	Offset
swreg197	314h

16.3.5.2.156.2 Function

Delta POC extension

16.3.5.2.156.3 Diagram



16.3.5.2.156.4 Fields

Field	Function
31-22 SW_ENC_L0_DELTA_POC0_MSB	delta poc of list0 pic0 (bit[19..10])
21-12 SW_ENC_L0_DELTA_POC1_MSB	delta poc of list0 pic1 (bit[19..10])
11-2	delta poc of list1 pic0 (bit[19..10])

Table continues on the next page...

Field	Function
SW_ENC_L1_DELTA_POC0_MSB	
1-0 —	Reserved

16.3.5.2.157 Long Term Reference Control (swreg198)

16.3.5.2.157.1 Offset

Register	Offset
swreg198	318h

16.3.5.2.157.2 Function

Long Term Reference Control

16.3.5.2.157.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_L1_DELTA_POC1_MSB										Reserved					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.157.4 Fields

Field	Function
31-22 SW_ENC_L1_DELTA_POC1_MSB	delta poc of list1 pic1 (bit[19..10])
21-0 —	Reserved

16.3.5.2.158 Hash Code Control (swreg199)

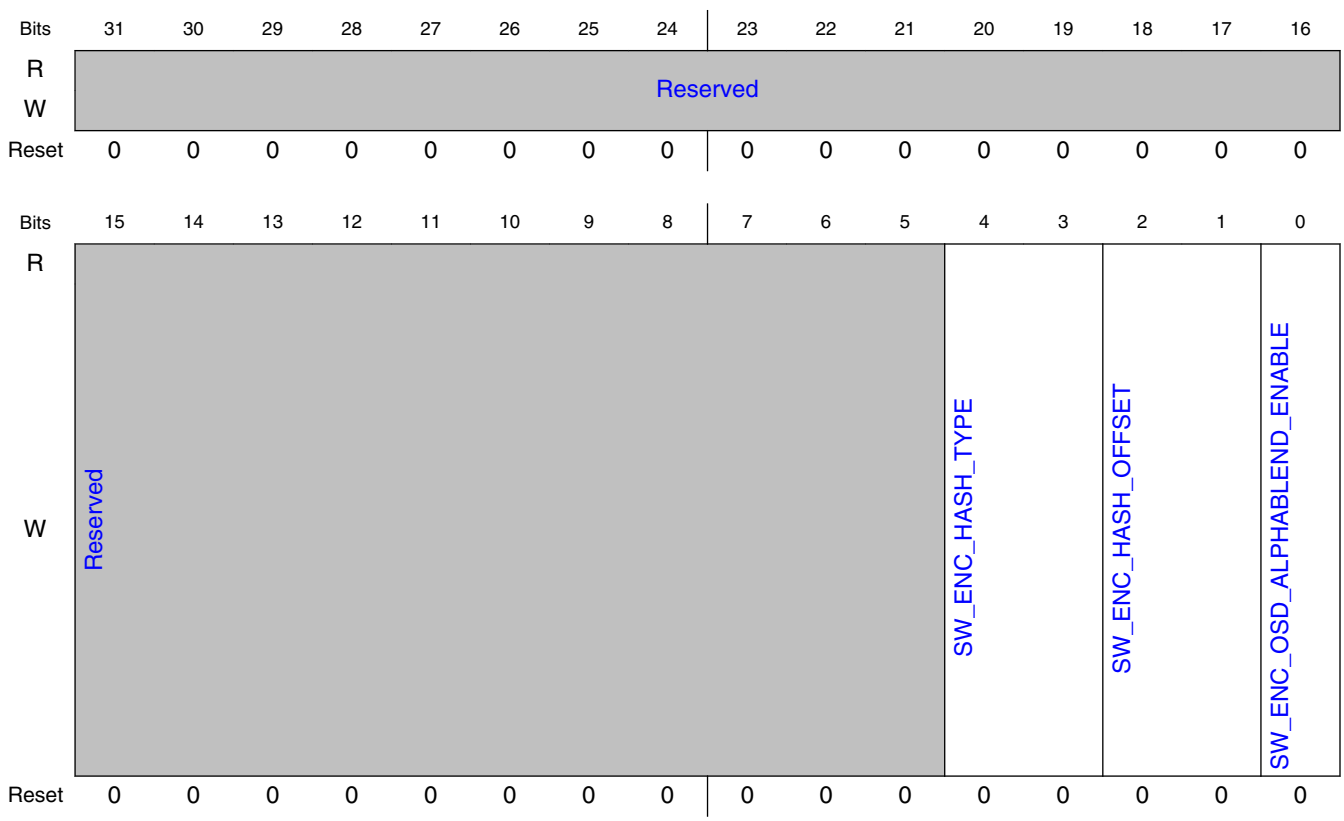
16.3.5.2.158.1 Offset

Register	Offset
swreg199	31Ch

16.3.5.2.158.2 Function

Hash Code Control

16.3.5.2.158.3 Diagram



16.3.5.2.158.4 Fields

Field	Function
31-5 —	Reserved

Table continues on the next page...

Field	Function
4-3 SW_ENC_HASH_TYPE	hash type of frame data. 00b - none. 01b - crc32. 10b - checksum32
2-1 SW_ENC_HASH_OFFSET	hash offset (byte offset of processed hashdata)
0 SW_ENC_OSD_ALPHABLEND_ENABLE	enable OSD Alpha Blending. 0b - disable. 1b - enable.

16.3.5.2.159 Hash Code Value (swreg200)

16.3.5.2.159.1 Offset

Register	Offset
swreg200	320h

16.3.5.2.159.2 Function

Hash Code Value

16.3.5.2.159.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_HASH_VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_HASH_VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.159.4 Fields

Field	Function
31-0	hash value of frame data

Field	Function
SW_ENC_HASH_VAL	

16.3.5.2.160 Background SKIP Control 0 (swreg201)

16.3.5.2.160.1 Offset

Register	Offset
swreg201	324h

16.3.5.2.160.2 Function

Background SKIP Control 0

16.3.5.2.160.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MEAN_THR0								SW_ENC_MEAN_THR1							
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MEAN_THR2								SW_ENC_MEAN_THR3							
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

16.3.5.2.160.4 Fields

Field	Function
31-24 SW_ENC_MEAN_THR0	A mean value threshold of out circle used for foreground decision
23-16 SW_ENC_MEAN_THR1	A mean value threshold of second circle for foreground decision
15-8 SW_ENC_MEAN_THR2	A mean value threshold of third circle

Table continues on the next page...

Field	Function
7-0 SW_ENC_MEAN_THR3	A mean value threshold of the smallest circle

16.3.5.2.161 Background SKIP Control 1 (swreg202)

16.3.5.2.161.1 Offset

Register	Offset
swreg202	328h

16.3.5.2.161.2 Function

Background SKIP Control 1

16.3.5.2.161.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_THR_DC_LUM_8X8															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_THR_DC_CHROMA_8X8															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

16.3.5.2.161.4 Fields

Field	Function
31-16 SW_ENC_THR_DC_LUM_8X8	A threshold value of DC for CU08 luma
15-0 SW_ENC_THR_DC_CHROMA_8X8	A threshold value of DC for CU08 chroma

16.3.5.2.162 Background SKIP Control 2 (swreg203)

16.3.5.2.162.1 Offset

Register	Offset
swreg203	32Ch

16.3.5.2.162.2 Function

Background SKIP Control 2

16.3.5.2.162.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_THR_DC_LUM_16X16															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_THR_DC_CHROMA_16X16															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

16.3.5.2.162.4 Fields

Field	Function
31-16 SW_ENC_THR_DC_LUM_16X16	A threshold value of DC for CU16 luma
15-0 SW_ENC_THR_DC_CHROMA_16X16	A threshold value of DC for CU16 chroma

16.3.5.2.163 Background SKIP Control 3 (swreg204)

16.3.5.2.163.1 Offset

Register	Offset
swreg204	330h

16.3.5.2.163.2 Function

Background SKIP Control 3

16.3.5.2.163.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_THR_DC_LUM_32X32															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_THR_DC_CHROMA_32X32															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

16.3.5.2.163.4 Fields

Field	Function
31-16 SW_ENC_THR_DC_LUM_32X32	A threshold value of DC for CU32 luma
15-0 SW_ENC_THR_DC_CHROMA_32X32	A threshold value of DC for CU32 chroma

16.3.5.2.164 Background SKIP Control 4 (swreg205)

16.3.5.2.164.1 Offset

Register	Offset
swreg205	334h

16.3.5.2.164.2 Function

Background SKIP Control 4

16.3.5.2.164.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_THR_AC_NUM_LUM_8X8															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_THR_AC_NUM_CHROMA_8X8															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

16.3.5.2.164.4 Fields

Field	Function
31-16 SW_ENC_THR_AC_NUM_LUM_8X8	A threshold value of number of non-zero AC for CU08 luma
15-0 SW_ENC_THR_AC_NUM_CHROMA_8X8	A threshold value of number of non-zero AC for CU08 chroma

16.3.5.2.165 Background SKIP Control 5 (swreg206)

16.3.5.2.165.1 Offset

Register	Offset
swreg206	338h

16.3.5.2.165.2 Function

Background SKIP Control 5

16.3.5.2.165.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_THR_AC_NUM_LUM_16X16															
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_THR_AC_NUM_CHROMA_16X16															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

16.3.5.2.165.4 Fields

Field	Function
31-16 SW_ENC_THR_AC_NUM_LUM_16X16	A threshold value of number of non-zero AC for CU16 luma
15-0 SW_ENC_THR_AC_NUM_CHROMA_16X16	A threshold value of number of non-zero AC for CU16 chroma

16.3.5.2.166 Background SKIP Control 6 (swreg207)

16.3.5.2.166.1 Offset

Register	Offset
swreg207	33Ch

16.3.5.2.166.2 Function

Background SKIP Control 6

16.3.5.2.166.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_THR_AC_NUM_LUM_32X32															
Reset	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_THR_AC_NUM_CHROMA_32X32															
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1

16.3.5.2.166.4 Fields

Field	Function
31-16 SW_ENC_THR_AC_NUM_LUM_32X32	A threshold value of number of non-zero AC for CU32 luma
15-0 SW_ENC_THR_AC_NUM_CHROMA_32X32	A threshold value of number of non-zero AC for CU32 chroma

16.3.5.2.167 Background SKIP Control 7 (swreg208)

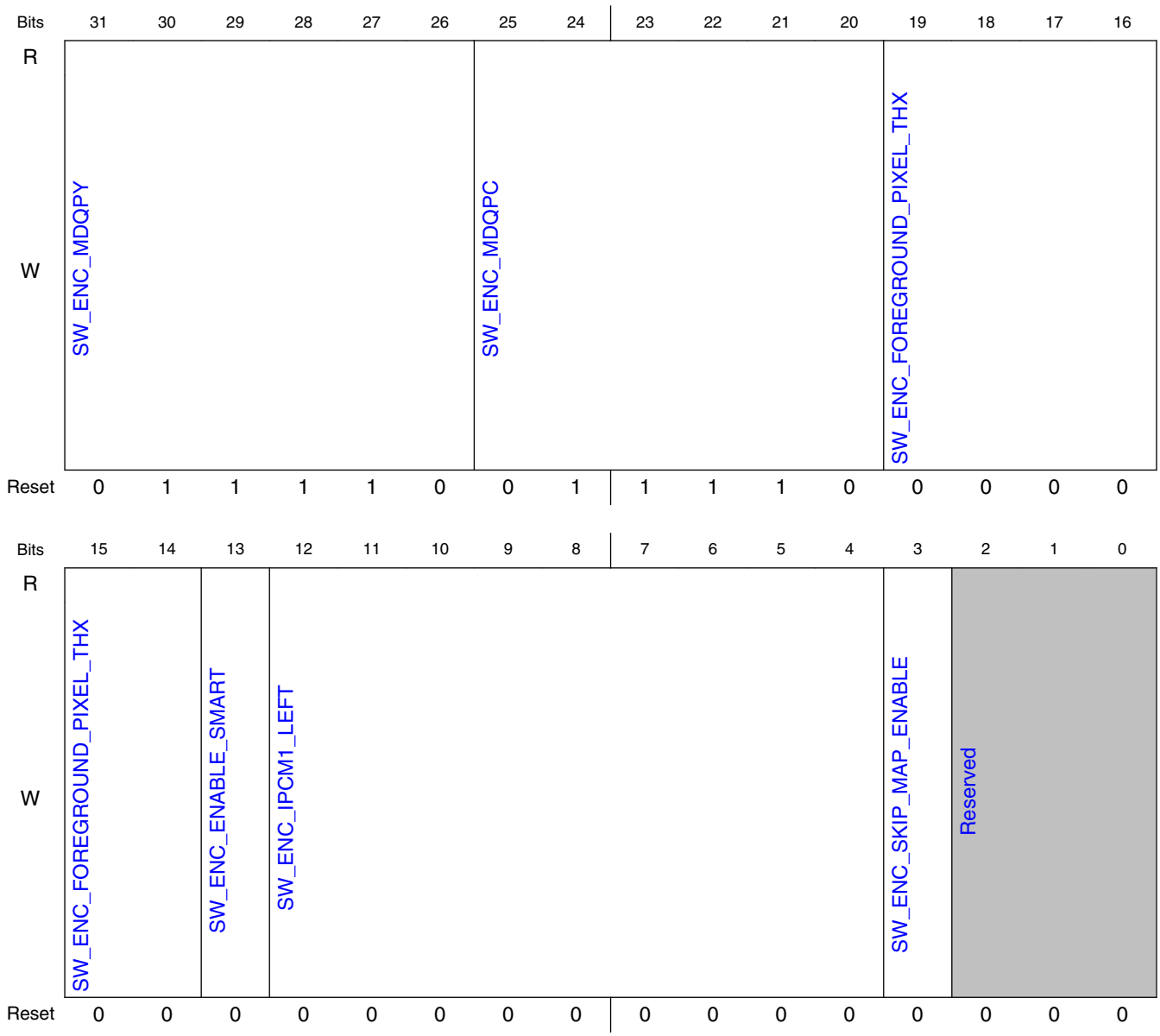
16.3.5.2.167.1 Offset

Register	Offset
swreg208	340h

16.3.5.2.167.2 Function

Background SKIP Control 7

16.3.5.2.167.3 Diagram



16.3.5.2.167.4 Fields

Field	Function
31-26 SW_ENC_MDQPY	A luma QP value for skip decision
25-20 SW_ENC_MDQPC	A chroma QP value for skip decision

Table continues on the next page...

Field	Function
19-14 SW_ENC_FOREGROUND_PIXEL_THX	Foreground threshold number of pixel (0 ~ 63) used for foreground decision
13 SW_ENC_ENABLE_SMART	enable smart featrue
12-4 SW_ENC_IPCM1_LEFT	IPCM1 Area Left
3 SW_ENC_SKIP_MAP_ENABLE	enable skip map mode
2-0 —	Reserved

16.3.5.2.168 IPCM Control 0 (swreg209)

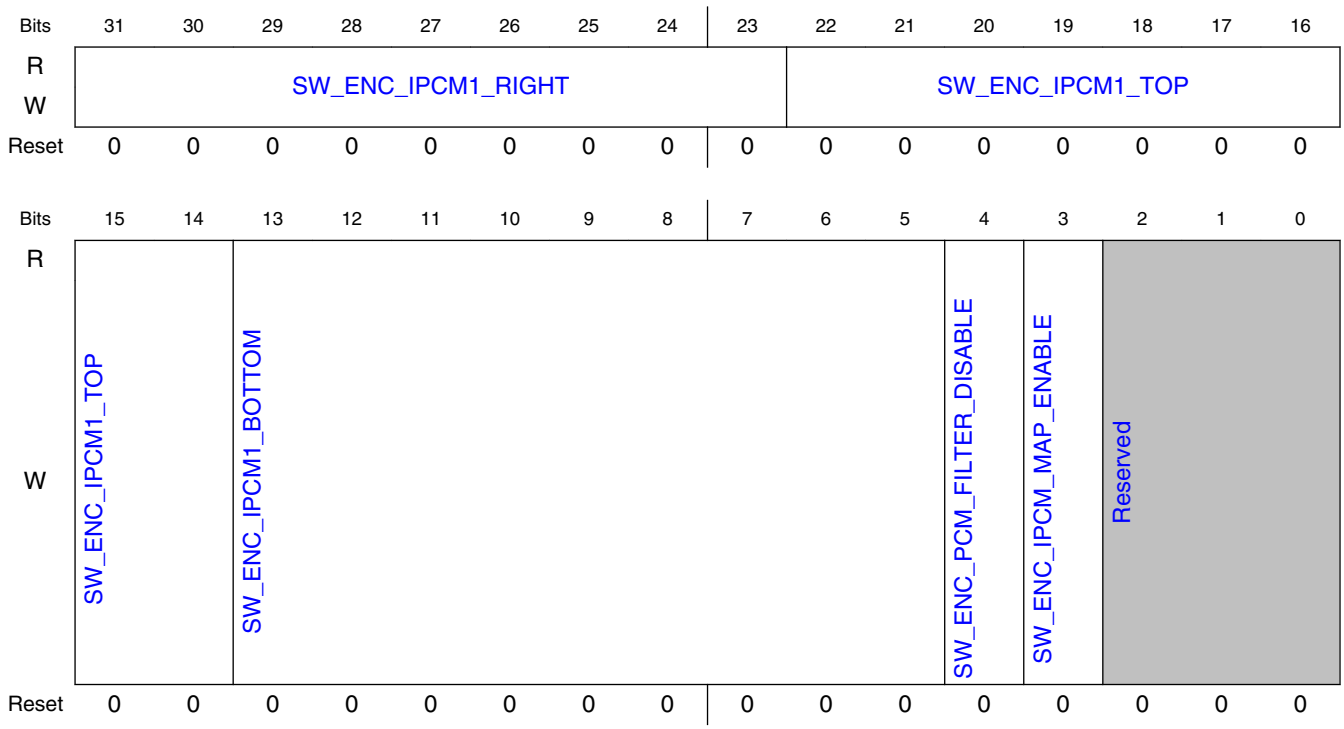
16.3.5.2.168.1 Offset

Register	Offset
swreg209	344h

16.3.5.2.168.2 Function

IPCM Control 0

16.3.5.2.168.3 Diagram



16.3.5.2.168.4 Fields

Field	Function
31-23 SW_ENC_IPCM1_RIGHT	IPCM1 Area Right
22-14 SW_ENC_IPCM1_TOP	IPCM1 Area Top
13-5 SW_ENC_IPCM1_BOTTOM	IPCM1 Area Bottom
4 SW_ENC_PCM_FILTER_DISABLE	disable deblock filter for IPCM
3 SW_ENC_IPCM_MAP_ENABLE	enable pcm map mode
2-0 —	Reserved

16.3.5.2.169 IPCM Control 1 (swreg210)

16.3.5.2.169.1 Offset

Register	Offset
swreg210	348h

16.3.5.2.169.2 Function

IPCM Control 1

16.3.5.2.169.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				SW_ENC_IPCM2_LEFT									Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.169.4 Fields

Field	Function
31-12 —	Reserved
11-3 SW_ENC_IPCM2_LEFT	IPCM2 Area Left
2-0 —	Reserved

16.3.5.2.170 IPCM Control 2 (swreg211)

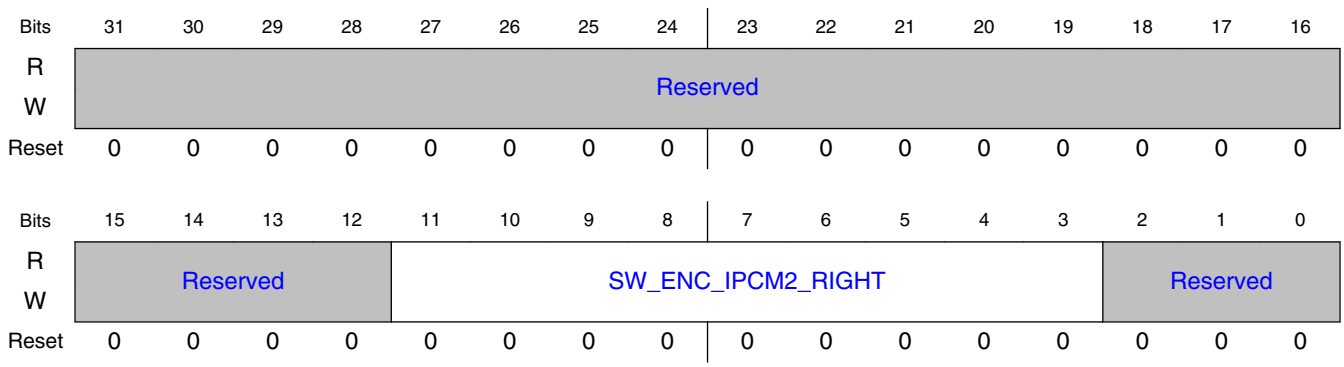
16.3.5.2.170.1 Offset

Register	Offset
swreg211	34Ch

16.3.5.2.170.2 Function

IPCM Control 2

16.3.5.2.170.3 Diagram



16.3.5.2.170.4 Fields

Field	Function
31-12 —	Reserved
11-3 SW_ENC_IPCM2_RIGHT	IPCM2 Area Right
2-0 —	Reserved

16.3.5.2.171 IPCM Control 3 (swreg212)

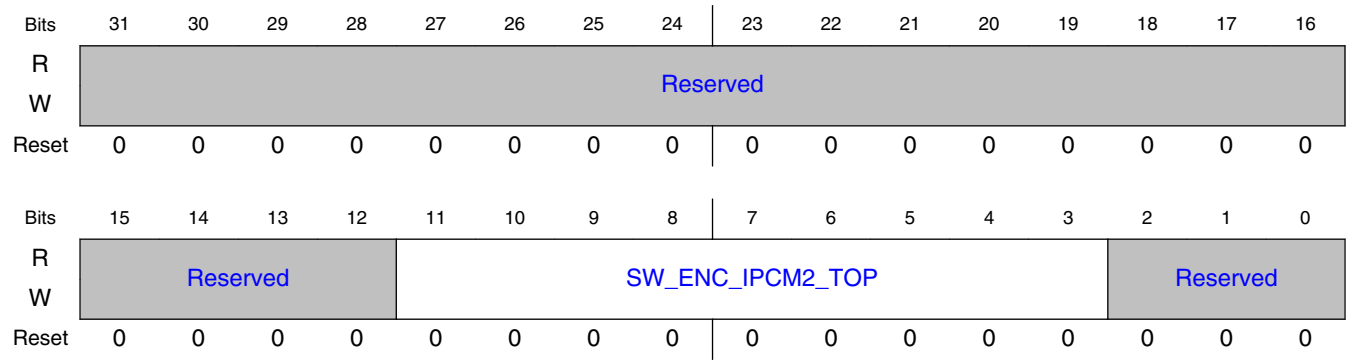
16.3.5.2.171.1 Offset

Register	Offset
swreg212	350h

16.3.5.2.171.2 Function

IPCM Control 3

16.3.5.2.171.3 Diagram



16.3.5.2.171.4 Fields

Field	Function
31-12 —	Reserved
11-3 SW_ENC_IPCM2_TOP	IPCM2 Area Top
2-0 —	Reserved

16.3.5.2.172 IPCM Control 4 (swreg213)

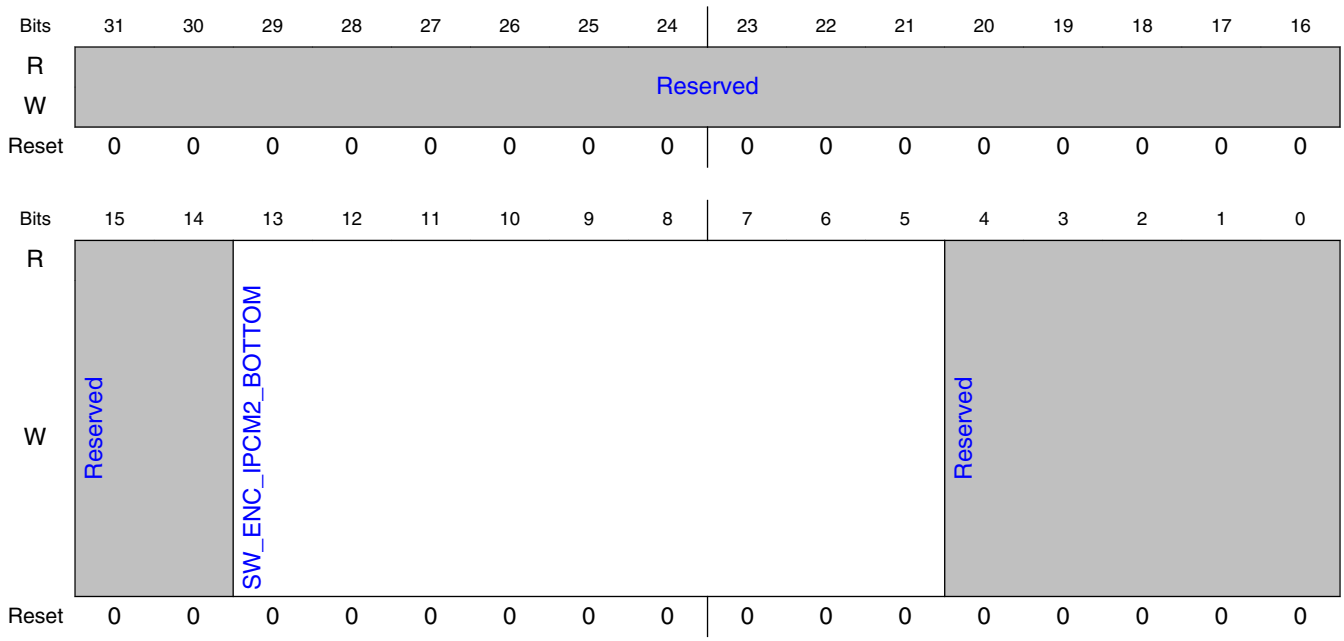
16.3.5.2.172.1 Offset

Register	Offset
swreg213	354h

16.3.5.2.172.2 Function

IPCM Control 4

16.3.5.2.172.3 Diagram



16.3.5.2.172.4 Fields

Field	Function
31-14 —	Reserved
13-5 SW_ENC_IPCM2_BOTTOM	IPCM2 Area Bottom
4-0 —	Reserved

16.3.5.2.173 HW synthesis config register 2, read-only (swreg214)

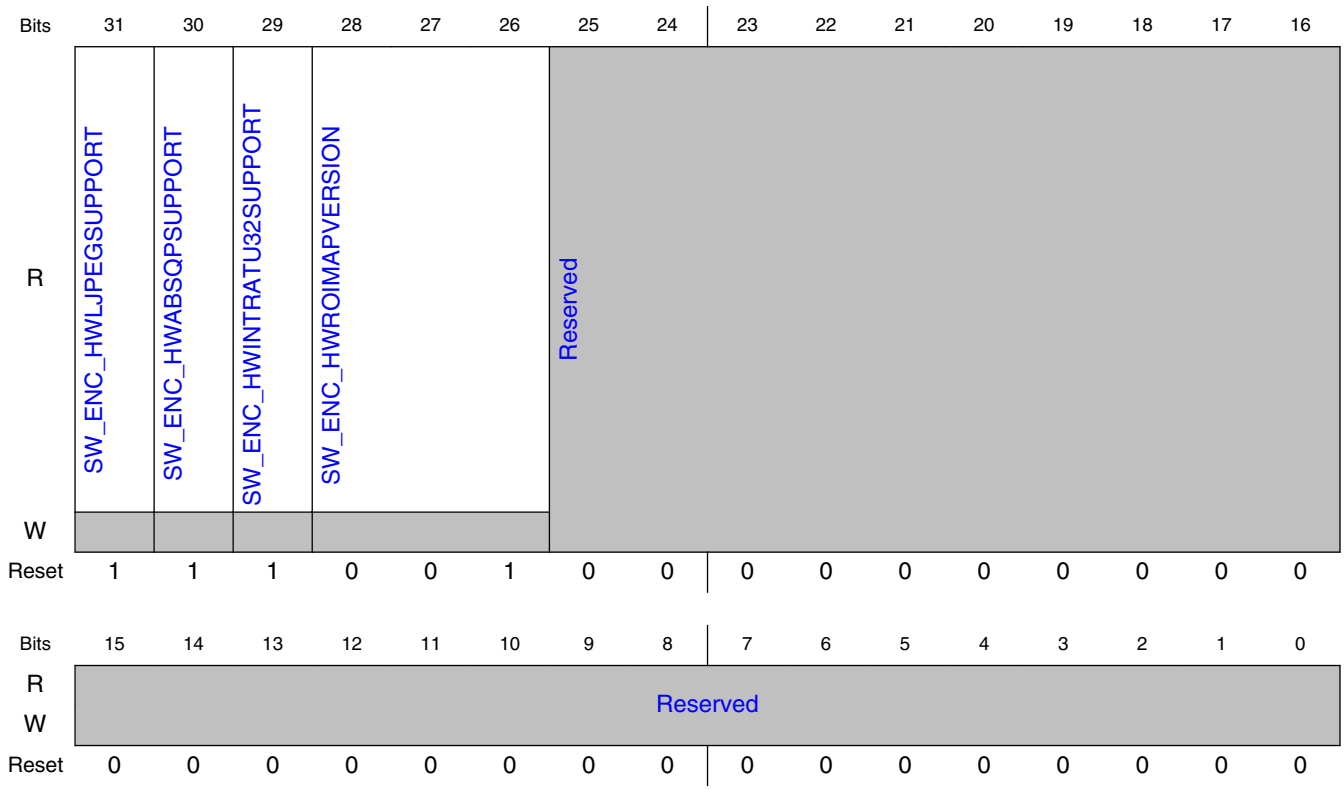
16.3.5.2.173.1 Offset

Register	Offset
swreg214	358h

16.3.5.2.173.2 Function

HW synthesis config register 2, read-only

16.3.5.2.173.3 Diagram



16.3.5.2.173.4 Fields

Field	Function
31 SW_ENC_HWLJPEGSUPPORT	Lossless JPEG supported by HW. 0b - not supported. 1b - supported
30 SW_ENC_HWABSQPSUPPORT	Absolute QP in ROI/ROI map supported by HW. 0b - not supported. 1b - supported
29 SW_ENC_HWINTRATU32SUPPORT	IntraPath TU32x32 supported by HW. 0b - not supported. 1b - supported
28-26 SW_ENC_HWROIMAPVERSION	ROI map buffer format version. 000b - 4 bit per pixel. 001b - 8 bit per pixel
25-0 —	Reserved

16.3.5.2.174 AXI Information 0 (swreg215)

16.3.5.2.174.1 Offset

Register	Offset
swreg215	35Ch

16.3.5.2.174.2 Function

AXI Information 0

16.3.5.2.174.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALARLEN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALARLEN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.174.4 Fields

Field	Function
31-0 SW_ENC_TOTALARLEN	[AXI] Accumulated ARLEN+1

16.3.5.2.175 AXI Information 1 (swreg216)

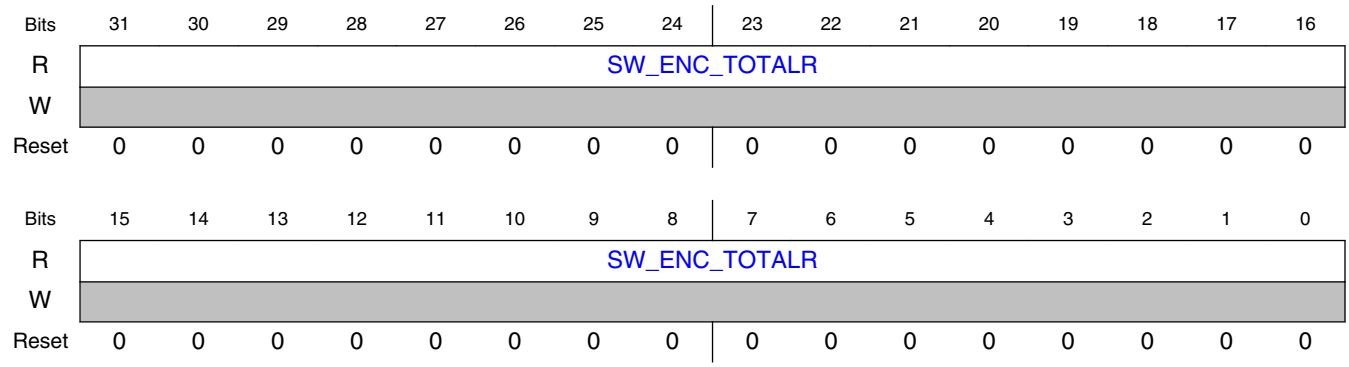
16.3.5.2.175.1 Offset

Register	Offset
swreg216	360h

16.3.5.2.175.2 Function

AXI Information 1

16.3.5.2.175.3 Diagram



16.3.5.2.175.4 Fields

Field	Function
31-0 SW_ENC_TOTALR	[AXI] RVALID & RREADY

16.3.5.2.176 AXI Information 2 (swreg217)

16.3.5.2.176.1 Offset

Register	Offset
swreg217	364h

16.3.5.2.176.2 Function

AXI Information 2

16.3.5.2.176.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALAR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALAR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.176.4 Fields

Field	Function
31-0 SW_ENC_TOTALAR	[AXI] ARVALID & ARREADY

16.3.5.2.177 AXI Information 3 (swreg218)

16.3.5.2.177.1 Offset

Register	Offset
swreg218	368h

16.3.5.2.177.2 Function

AXI Information 3

16.3.5.2.177.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALRLAST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALRLAST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.177.4 Fields

Field	Function
31-0 SW_ENC_TOTALRLAST	[AXI] RVALID & RREADY & RLAST

16.3.5.2.178 AXI Information 4 (swreg219)

16.3.5.2.178.1 Offset

Register	Offset
swreg219	36Ch

16.3.5.2.178.2 Function

AXI Information 4

16.3.5.2.178.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALAWLEN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALAWLEN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.178.4 Fields

Field	Function
31-0 SW_ENC_TOTALAWLEN	[AXI] Accumulated AWLEN+1

16.3.5.2.179 AXI Information 5 (swreg220)

16.3.5.2.179.1 Offset

Register	Offset
swreg220	370h

16.3.5.2.179.2 Function

AXI Information 5

16.3.5.2.179.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.179.4 Fields

Field	Function
31-0 SW_ENC_TOTALW	[AXI] WVALID & WREADY

16.3.5.2.180 AXI Information 6 (swreg221)

16.3.5.2.180.1 Offset

Register	Offset
swreg221	374h

16.3.5.2.180.2 Function

AXI Information 6

16.3.5.2.180.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALAW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALAW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.180.4 Fields

Field	Function
31-0 SW_ENC_TOTALAW	[AXI] AWVALID & AWREADY

16.3.5.2.181 AXI Information 7 (swreg222)

16.3.5.2.181.1 Offset

Register	Offset
swreg222	378h

16.3.5.2.181.2 Function

AXI Information 7

16.3.5.2.181.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALWLAST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALWLAST															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.181.4 Fields

Field	Function
31-0 SW_ENC_TOTALWLAST	[AXI] WVALID & WREADY & WLAST

16.3.5.2.182 AXI Information 8 (swreg223)

16.3.5.2.182.1 Offset

Register	Offset
swreg223	37Ch

16.3.5.2.182.2 Function

AXI Information 8

16.3.5.2.182.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_ENC_TOTALB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_ENC_TOTALB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.182.4 Fields

Field	Function
31-0 SW_ENC_TOTALB	[AXI] BVALID & BREADY

16.3.5.2.183 control register 4 (swreg224)

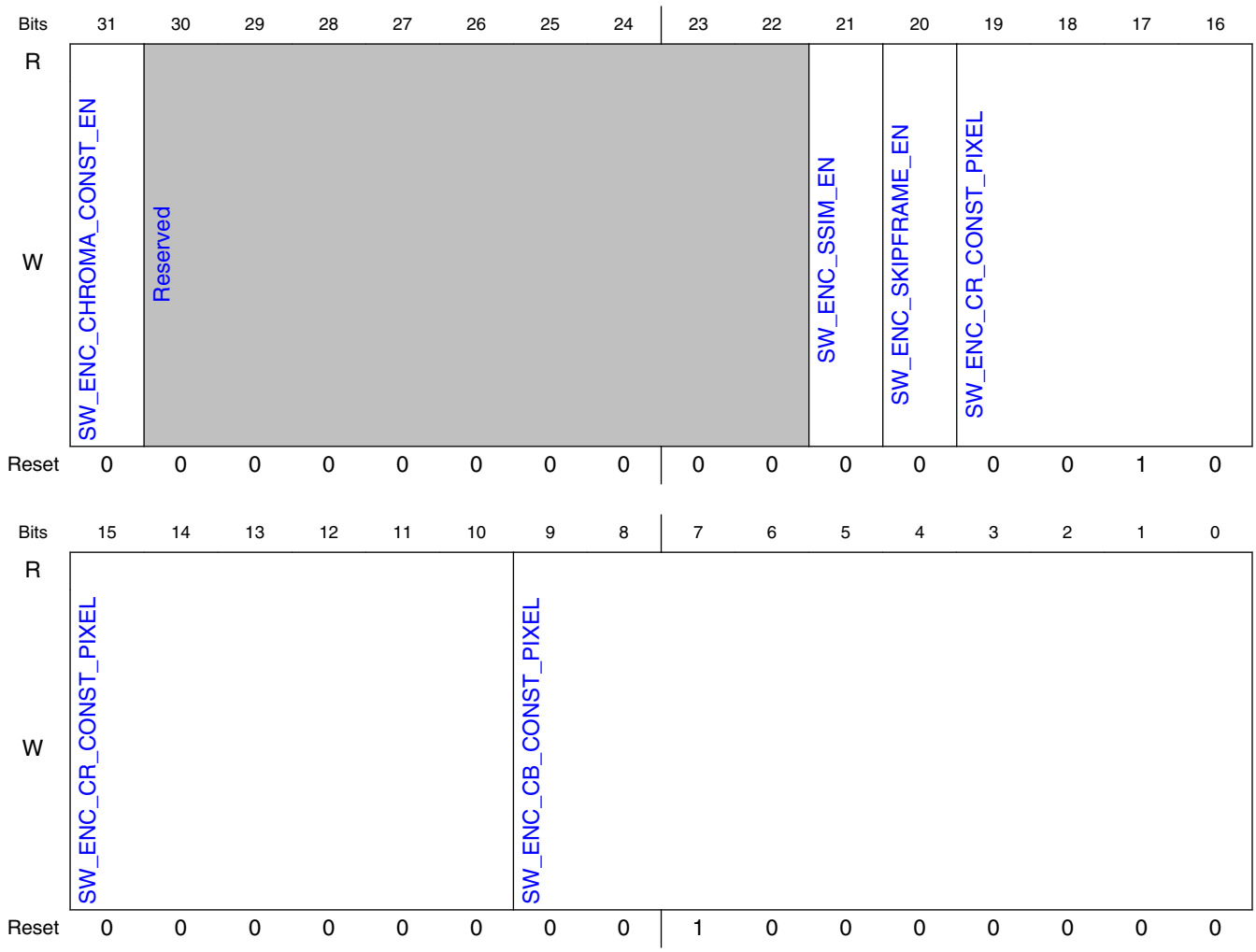
16.3.5.2.183.1 Offset

Register	Offset
swreg224	380h

16.3.5.2.183.2 Function

control register 4

16.3.5.2.183.3 Diagram



16.3.5.2.183.4 Fields

Field	Function
31 SW_ENC_CHROMA_CONST_EN	Force Chroma to be a constant pixel or not. 0b - no. 1b - yes.
30-22 —	Reserved
21 SW_ENC_SSIM_EN	Enable SSIM calculation. 0b - Disable. 1b - Enable
20	Force frame encode as SKIPFRAME. 0b - no.

Table continues on the next page...

Field	Function
SW_ENC_SKIP FRAME_EN	1b - yes
19-10 SW_ENC_CR_ CONST_PIXEL	The constant pixel of CR if sw_enc_chroma_const_en=1
9-0 SW_ENC_CB_ CONST_PIXEL	The constant pixel of CB if sw_enc_chroma_const_en=1

16.3.5.2.184 Tile Control (swreg225)

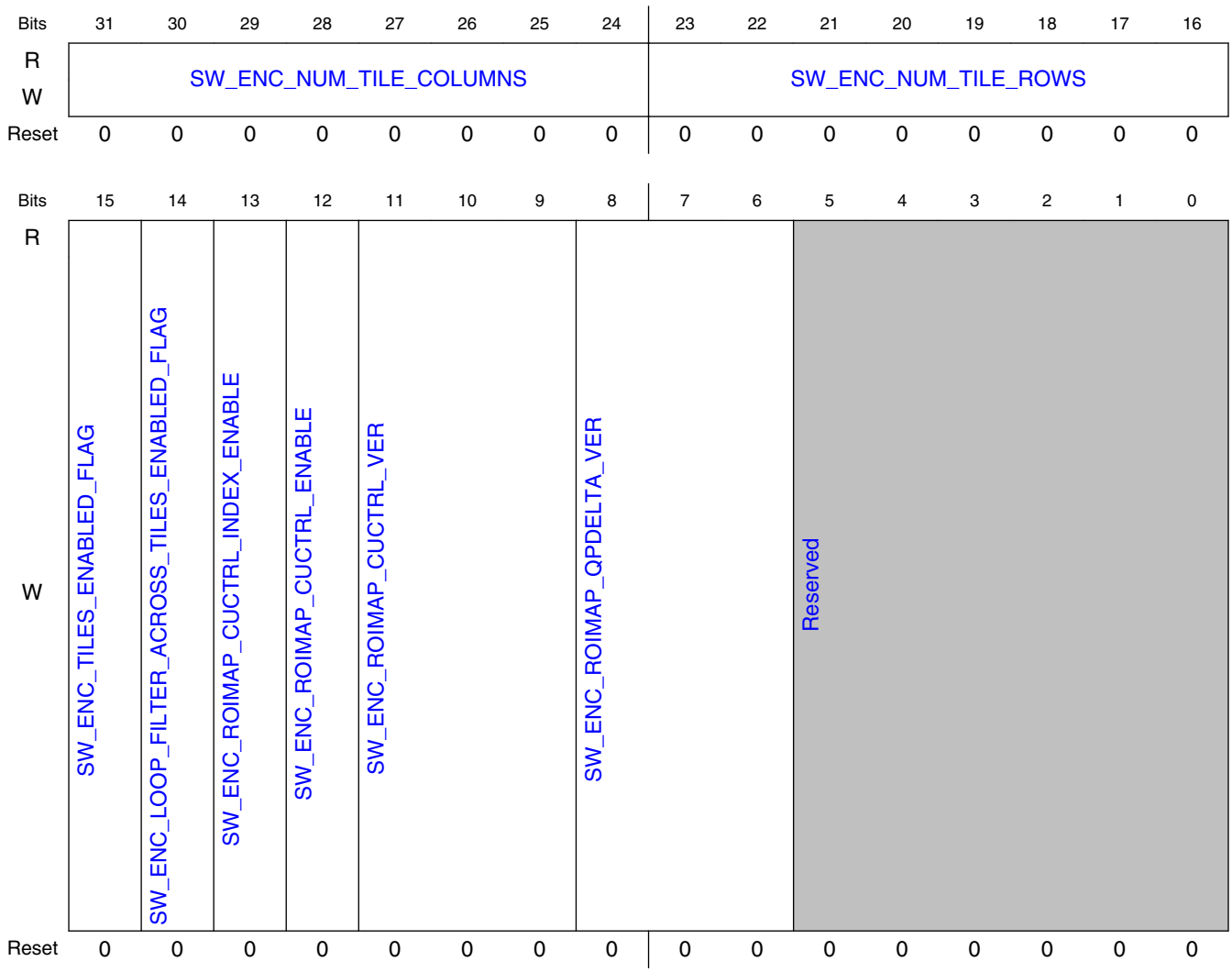
16.3.5.2.184.1 Offset

Register	Offset
swreg225	384h

16.3.5.2.184.2 Function

Tile Control

16.3.5.2.184.3 Diagram



16.3.5.2.184.4 Fields

Field	Function
31-24 SW_ENC_NUM_TILE_COLUMNS	the number of tile columns partitioning the picture.
23-16 SW_ENC_NUM_TILE_ROWS	the number of tile rows partitioning the picture.
15 SW_ENC_TILES_ENABLED_FLAG	enable HEVC tile partition. 0b - disabled. 1b - enabled.

Table continues on the next page...

Field	Function
14 SW_ENC_LOOP_FILTER_ACROSS_TILES_ENABLED_FLAG	enable in-loop filtering operations across tile boundaries. 0b - disabled. 1b - enabled.
13 SW_ENC_ROIMAP_CUCTRL_INDEX_ENABLE	roi map cu ctrl index enable. 0b - Disable. 1b - Enable
12 SW_ENC_ROIMAP_CUCTRL_ENABLE	roi map cu ctrl enable. 0b - Disable. 1b - Enable
11-9 SW_ENC_ROIMAP_CUCTRL_VER	roi map cu ctrl info version. 3~7
8-6 SW_ENC_ROIMAP_QPDELTA_VER	roi map qp delta info version. 1~3
5-0 —	Reserved

16.3.5.2.185 HW synthesis config register 3, read-only (swreg226)

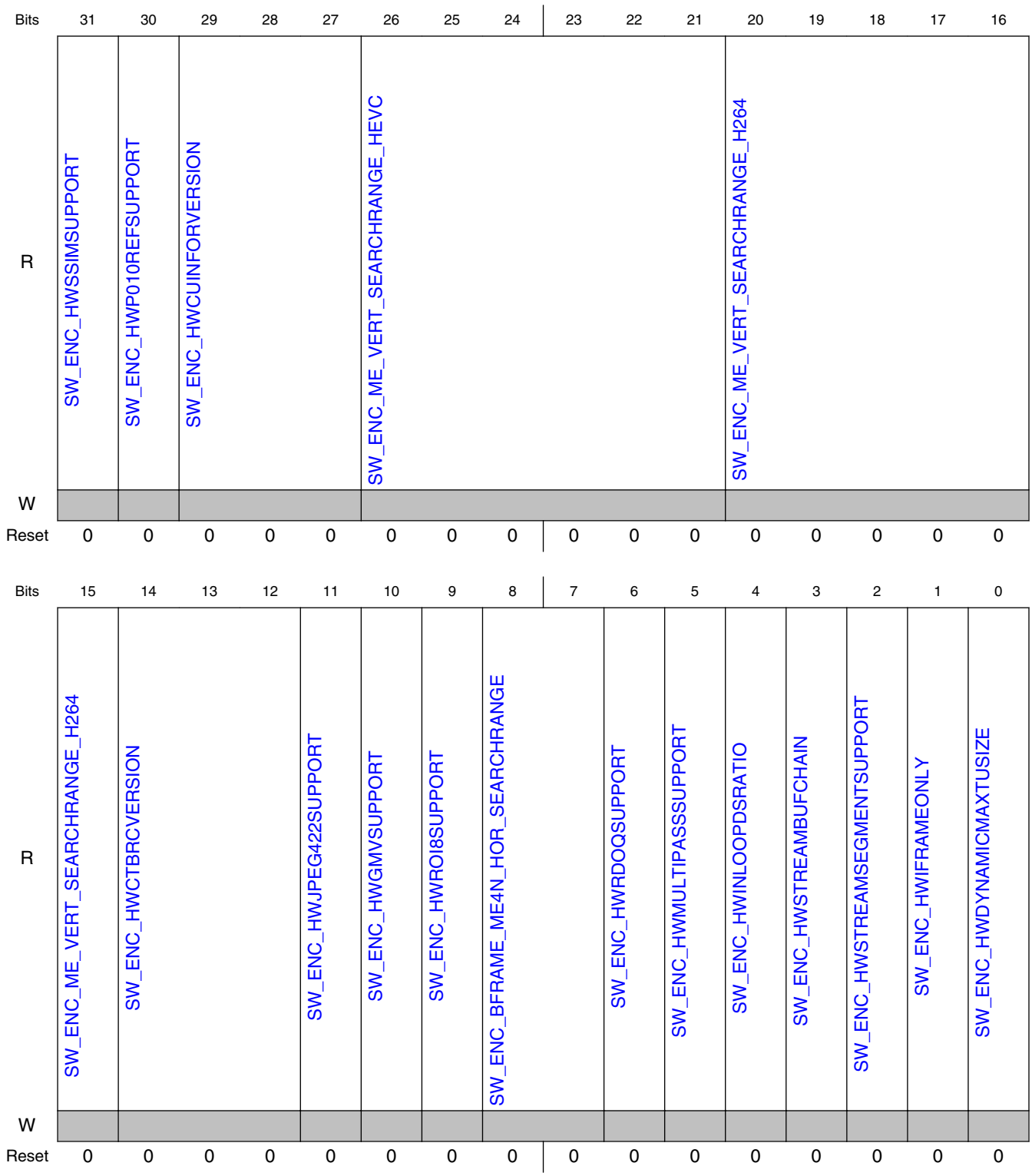
16.3.5.2.185.1 Offset

Register	Offset
swreg226	388h

16.3.5.2.185.2 Function

HW synthesis config register 3, read-only

16.3.5.2.185.3 Diagram



16.3.5.2.185.4 Fields

Field	Function
31 SW_ENC_HWSSIMSUPPORT	SSIM calculation supported by HW. 0b - not supported. 1b - supported
30 SW_ENC_HWP010REFSUPPORT	Reference buffer format for 10-bit encoding. 0b - normal format 1b - Reserved
29-27 SW_ENC_HWCUINFORVERSION	Version of the output CU information format.
26-21 SW_ENC_ME_VERT_SEARCHRANGE_HEVC	ME vertical search range in 8 pixel unit for HEVC
20-15 SW_ENC_ME_VERT_SEARCHRANGE_H264	ME vertical search range in 8 pixel unit for H264
14-12 SW_ENC_HWCTBRCVERSION	CTB Rate Control Version.
11 SW_ENC_HWJPEG422SUPPORT	Jpeg422 supported by HW. 0b - not supported. 1b - supported
10 SW_ENC_HWGMVSUPPORT	Global MV supported by HW. 0b - not supported. 1b - supported
9 SW_ENC_HWROI8SUPPORT	support 8 ROIs
8-7 SW_ENC_BFRAME_ME4N_HOR_SEARCHRANGE	ME4N horizontal search range in 64 pixel unit 00b - 64. 01b - 128. 10b - 192. 11b - 256
6 SW_ENC_HWRDOQSUPPORT	RDOQ supported by HW. 0b - not supported. 1b - supported
5 SW_ENC_HWMULTIPASSSUPPORT	Multipass Encoding supported by HW. 0b - not supported. 1b - supported

Table continues on the next page...

Field	Function
4 SW_ENC_HWINLOOPDSRATIO	in-loop ds ratio supported by HW 0b - 1:1 1b - 1:2
3 SW_ENC_HWSTREAMBUFCHAIN	Stream Buffer Chain supported by HW. 0b - not supported. 1b - supported
2 SW_ENC_HWSTREAMSEGMENTSUPPORT	Stream segment supported by HW. 0b - not supported. 1b - supported
1 SW_ENC_HWIFRAMEONLY	Only support I frame. 0b - support I/P/B frame. 1b - only support I frame
0 SW_ENC_HWDYNAMICMAXTUSIZE	HW support dynamic max TU size change per frame. 0b - not supported. 1b - supported

16.3.5.2.186 RPS encoding control 0 (swreg235)

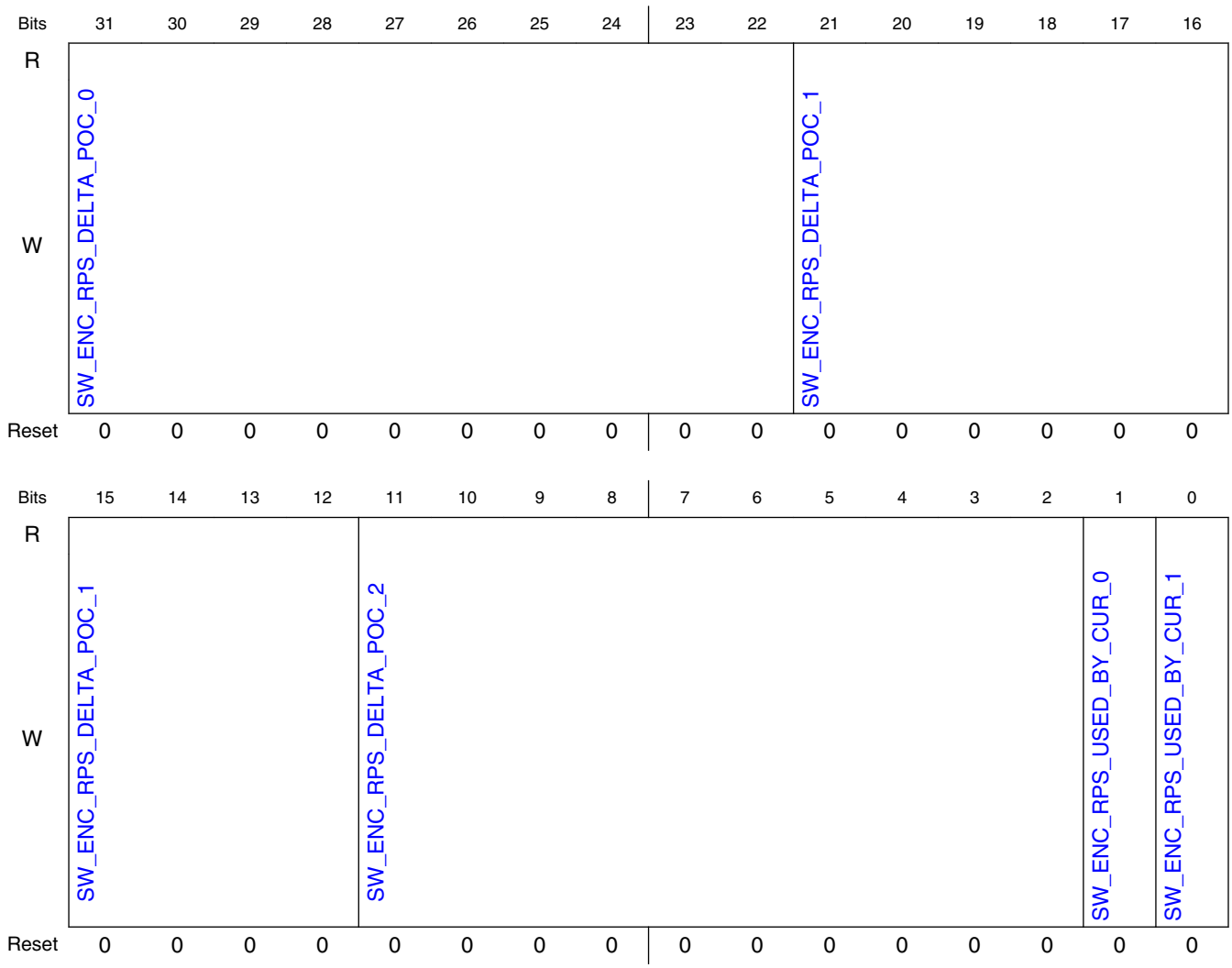
16.3.5.2.186.1 Offset

Register	Offset
swreg235	3ACh

16.3.5.2.186.2 Function

RPS encoding control 0

16.3.5.2.186.3 Diagram



16.3.5.2.186.4 Fields

Field	Function
31-22 SW_ENC_RPS_DELTA_POC_0	delta poc of pic0 in current rps
21-12 SW_ENC_RPS_DELTA_POC_1	delta poc of pic1 in current rps
11-2 SW_ENC_RPS_DELTA_POC_2	delta poc of pic2 in current rps
1	pic0 in current rps is used for reference by current slice

Table continues on the next page...

Field	Function
SW_ENC_RPS_USED_BY_CUR_0	
0 SW_ENC_RPS_USED_BY_CUR_1	pic1 in current rps is used for reference by current slice

16.3.5.2.187 RPS encoding control 1 (swreg236)

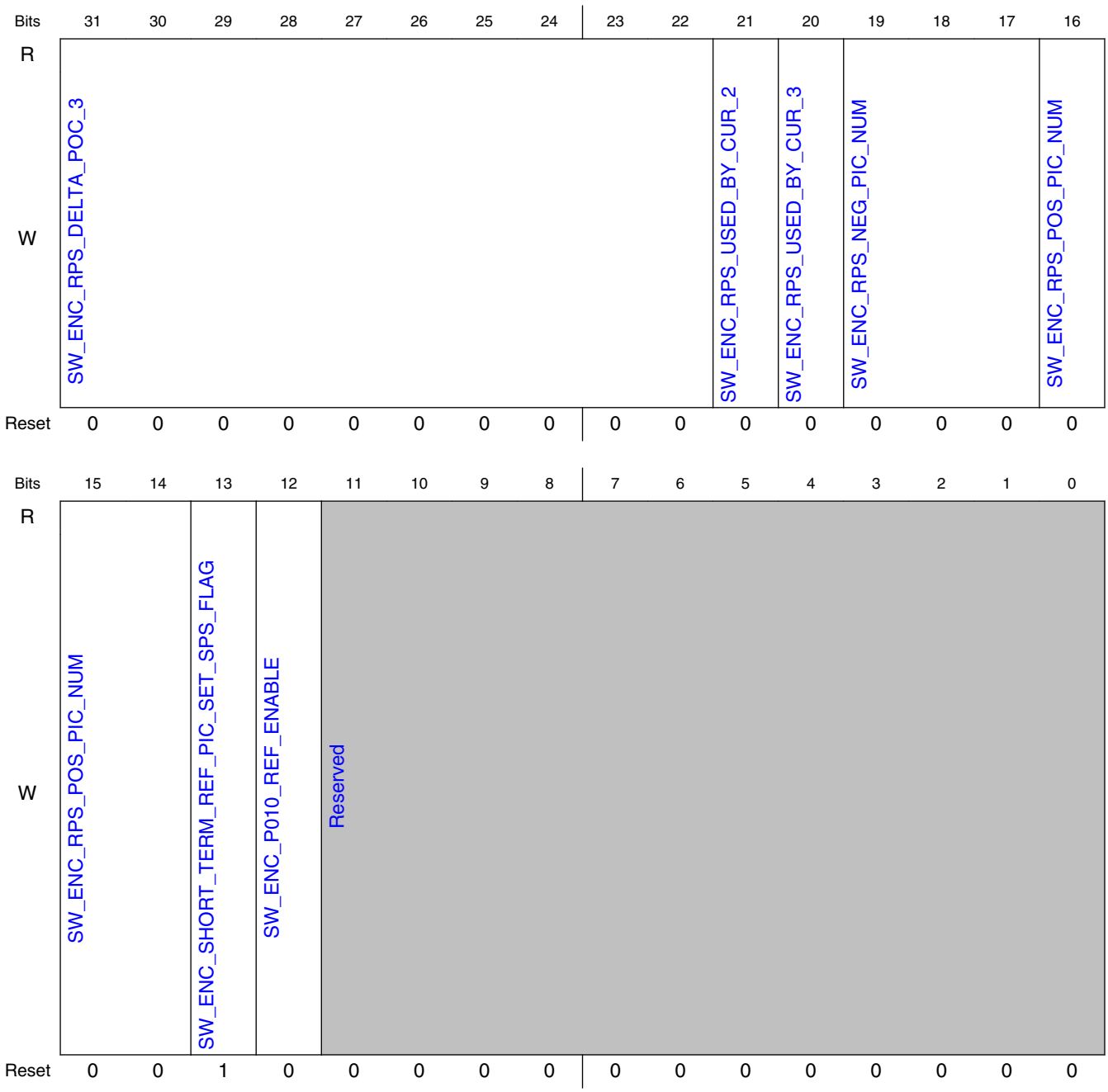
16.3.5.2.187.1 Offset

Register	Offset
swreg236	3B0h

16.3.5.2.187.2 Function

RPS encoding control 1

16.3.5.2.187.3 Diagram



16.3.5.2.187.4 Fields

Field	Function
31-22 SW_ENC_RPS_DELTA_POC_3	delta poc of pic3 in current rps

Table continues on the next page...

Field	Function
21 SW_ENC_RPS_USED_BY_CUR_2	pic2 in current rps is used for reference by current slice
20 SW_ENC_RPS_USED_BY_CUR_3	pic3 in current rps is used for reference by current slice
19-17 SW_ENC_RPS_NEG_PIC_NUM	number of negative pictures in current rps
16-14 SW_ENC_RPS_POS_PIC_NUM	number of positive pictures in current rps
13 SW_ENC_SHORT_TERM_REF_PIC_SET_SPS_FLAG	short term rps for current slice is encoded in sps
12 SW_ENC_P010_REF_ENABLE	Enable/Disable P010 Reference format. P010RefEnable[0]. 0b - not supported. 1b - supported.
11-0 —	Reserved

16.3.5.2.188 Stride Control (swreg237)

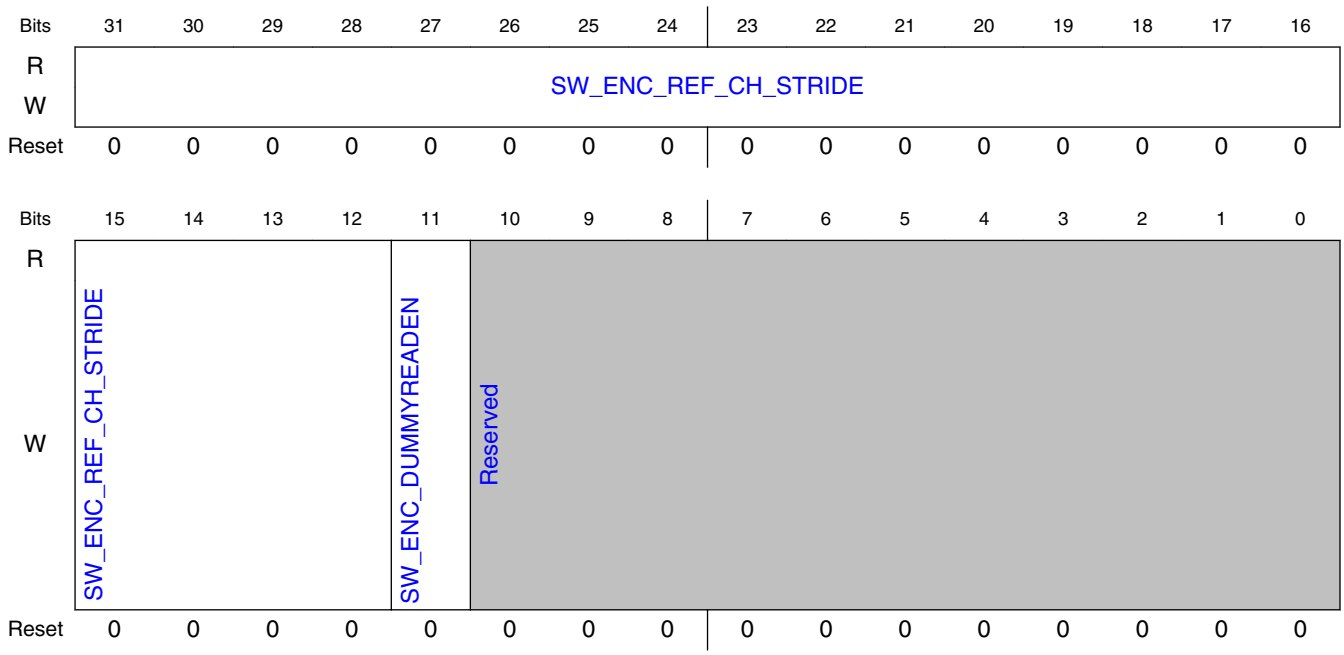
16.3.5.2.188.1 Offset

Register	Offset
swreg237	3B4h

16.3.5.2.188.2 Function

Stride Control

16.3.5.2.188.3 Diagram



16.3.5.2.188.4 Fields

Field	Function
31-12 SW_ENC_REF_CH_STRIDE	Chroma stride of reference frame
11 SW_ENC_DUMMYREADEN	enable dummy read when frame height not align to 64 or other possible condition
10-0 —	Reserved

16.3.5.2.189 Dummy Read (swreg238)

16.3.5.2.189.1 Offset

Register	Offset
swreg238	3B8h

16.3.5.2.189.2 Function

Dummy Read

16.3.5.2.189.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_DUMMYREADADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_DUMMYREADADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.189.4 Fields

Field	Function
31-0 SW_ENC_DUMMYREADADDR	Dummy read address

16.3.5.2.190 Base Address LSB of CTB MADs of current frame. (swreg239)

16.3.5.2.190.1 Offset

Register	Offset
swreg239	3BCh

16.3.5.2.190.2 Function

Base Address LSB of CTB MADs of current frame.

16.3.5.2.190.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_CURRENT_CTB_MAD_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_CURRENT_CTB_MAD_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.190.4 Fields

Field	Function
31-0 SW_ENC_CURRENT_CTB_MAD_BASE	Base Address LSB of CTB MADs of current frame used for CTB RC.

16.3.5.2.191 Base Address LSB of CTB MADs of previous frame. (swreg241)

16.3.5.2.191.1 Offset

Register	Offset
swreg241	3C4h

16.3.5.2.191.2 Function

Base Address LSB of CTB MADs of previous frame.

16.3.5.2.191.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_PREVIOUS_CTB_MAD_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_PREVIOUS_CTB_MAD_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.191.4 Fields

Field	Function
31-0 SW_ENC_PREVIOUS_CTB_MAD_BASE	Base Address LSB of CTB MADs of previous frame used for CTB RC.

16.3.5.2.192 CTB RC Control 0 (swreg243)

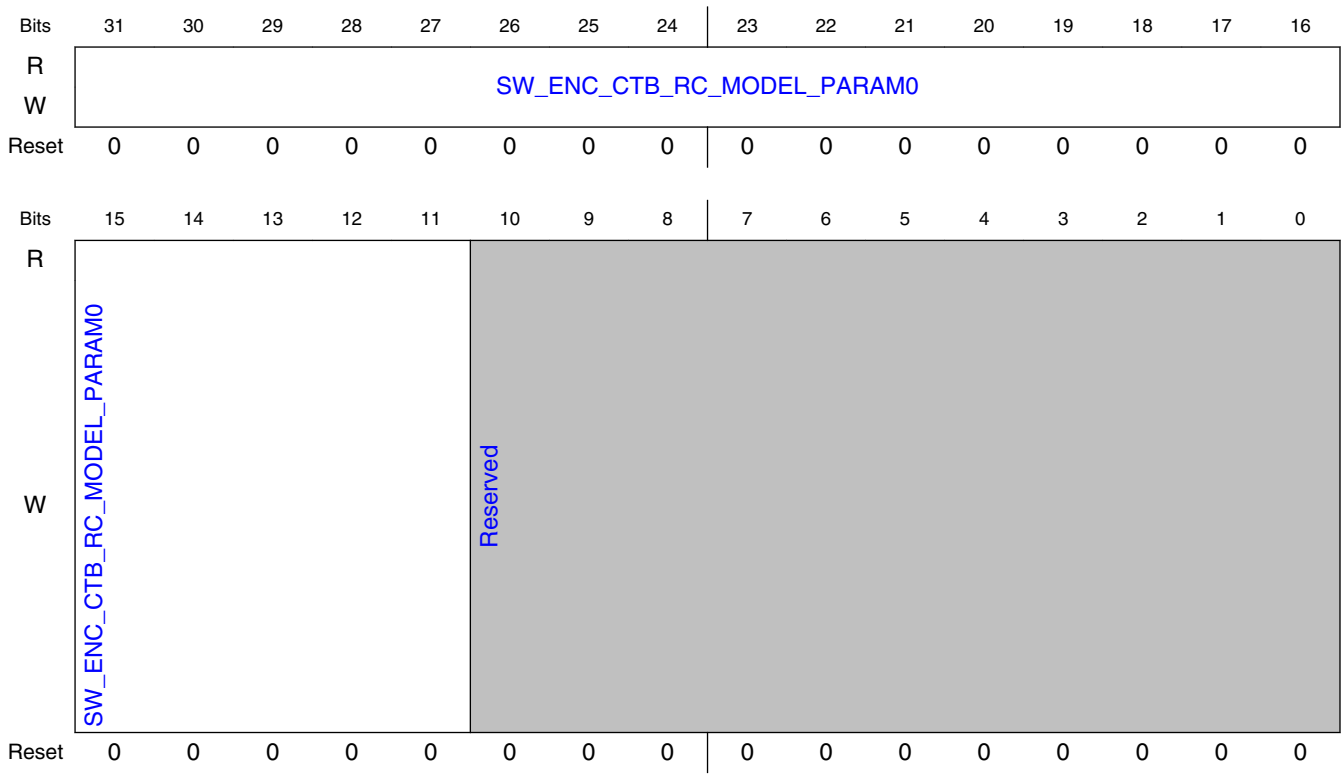
16.3.5.2.192.1 Offset

Register	Offset
swreg243	3CCh

16.3.5.2.192.2 Function

CTB RC Control 0

16.3.5.2.192.3 Diagram



16.3.5.2.192.4 Fields

Field	Function
31-11 SW_ENC_CTB_RC_MODEL_PARAM0	Parameter0 of CTB RC model
10-0 —	Reserved

16.3.5.2.193 CTB RC Control 1 (swreg244)

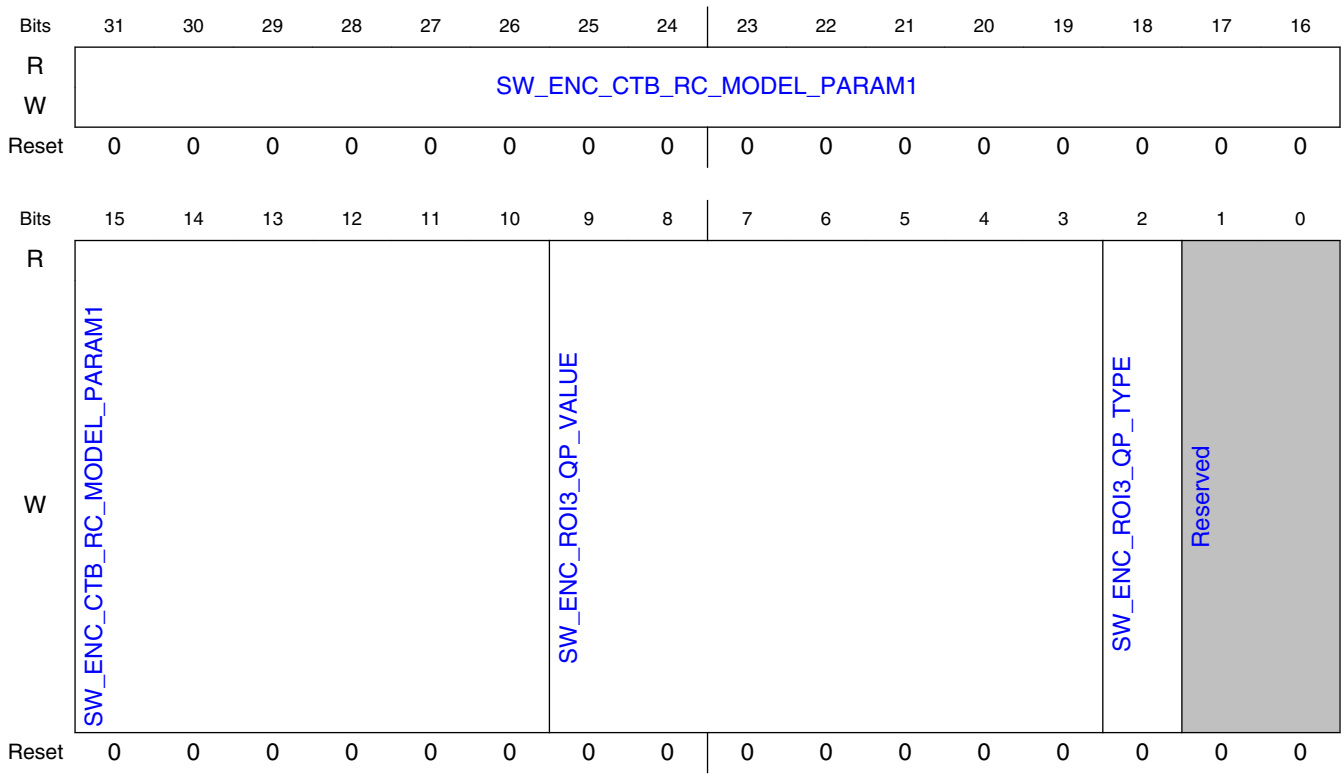
16.3.5.2.193.1 Offset

Register	Offset
swreg244	3D0h

16.3.5.2.193.2 Function

CTB RC Control 1

16.3.5.2.193.3 Diagram



16.3.5.2.193.4 Fields

Field	Function
31-10 SW_ENC_CTB_RC_MODEL_PARAM1	Parameter1 of CTB RC model
9-3 SW_ENC_ROI3_QP_VALUE	ROI3 qp value. delta or absolute value
2 SW_ENC_ROI3_QP_TYPE	ROI3 qp type. 0b - delta 1b - Absolute value
1-0 —	Reserved

16.3.5.2.194 CTB RC Control 2 (swreg245)

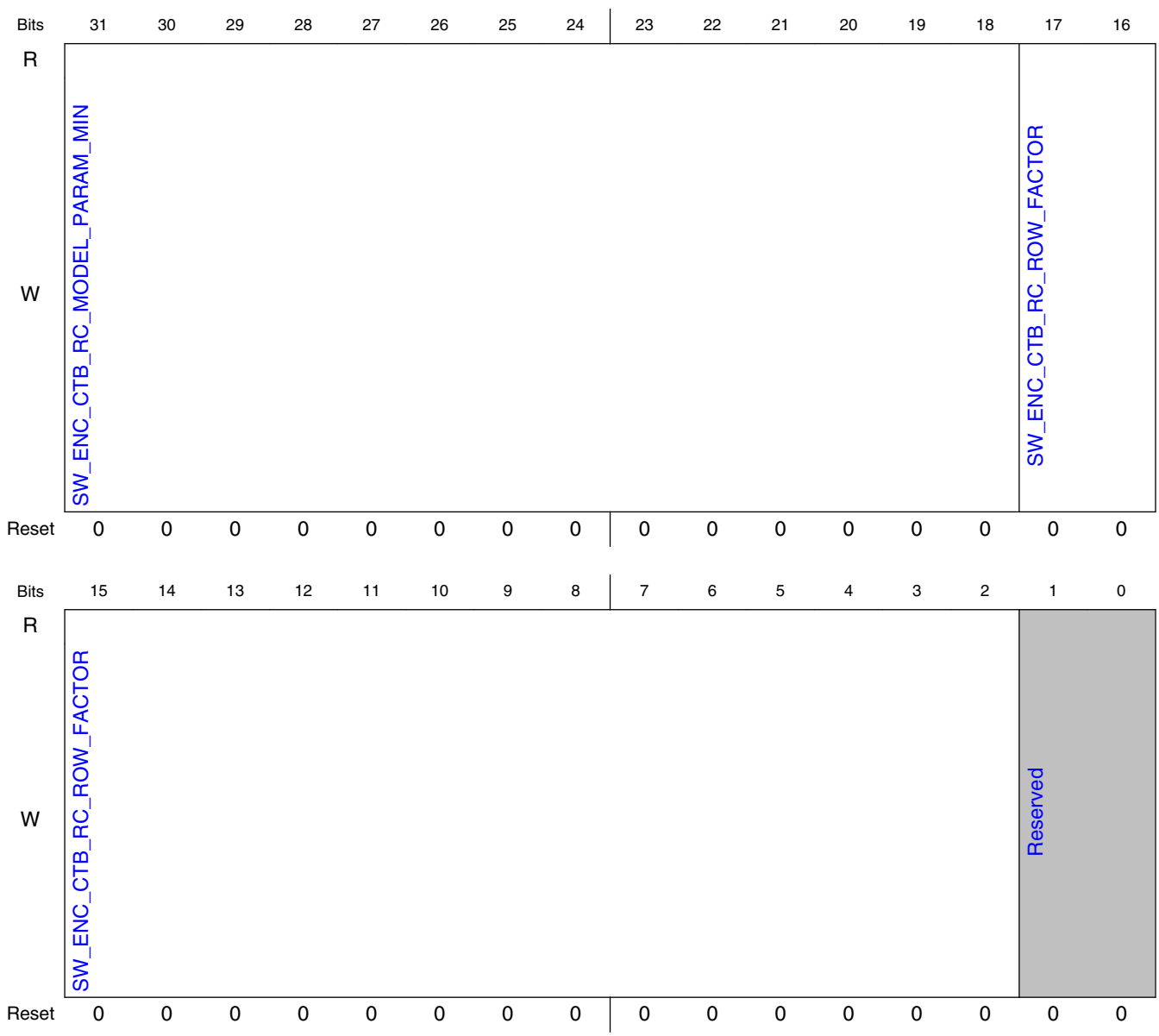
16.3.5.2.194.1 Offset

Register	Offset
swreg245	3D4h

16.3.5.2.194.2 Function

CTB RC Control 2

16.3.5.2.194.3 Diagram



16.3.5.2.194.4 Fields

Field	Function
31-18 SW_ENC_CTB_RC_MODEL_PARAM_MIN	Minimum parameter value of CTB RC model
17-2 SW_ENC_CTB_RC_ROW_FACTOR	Reciprocal of ctb per row. Q16
1-0 —	Reserved

16.3.5.2.195 CTB RC Control 3 (swreg246)

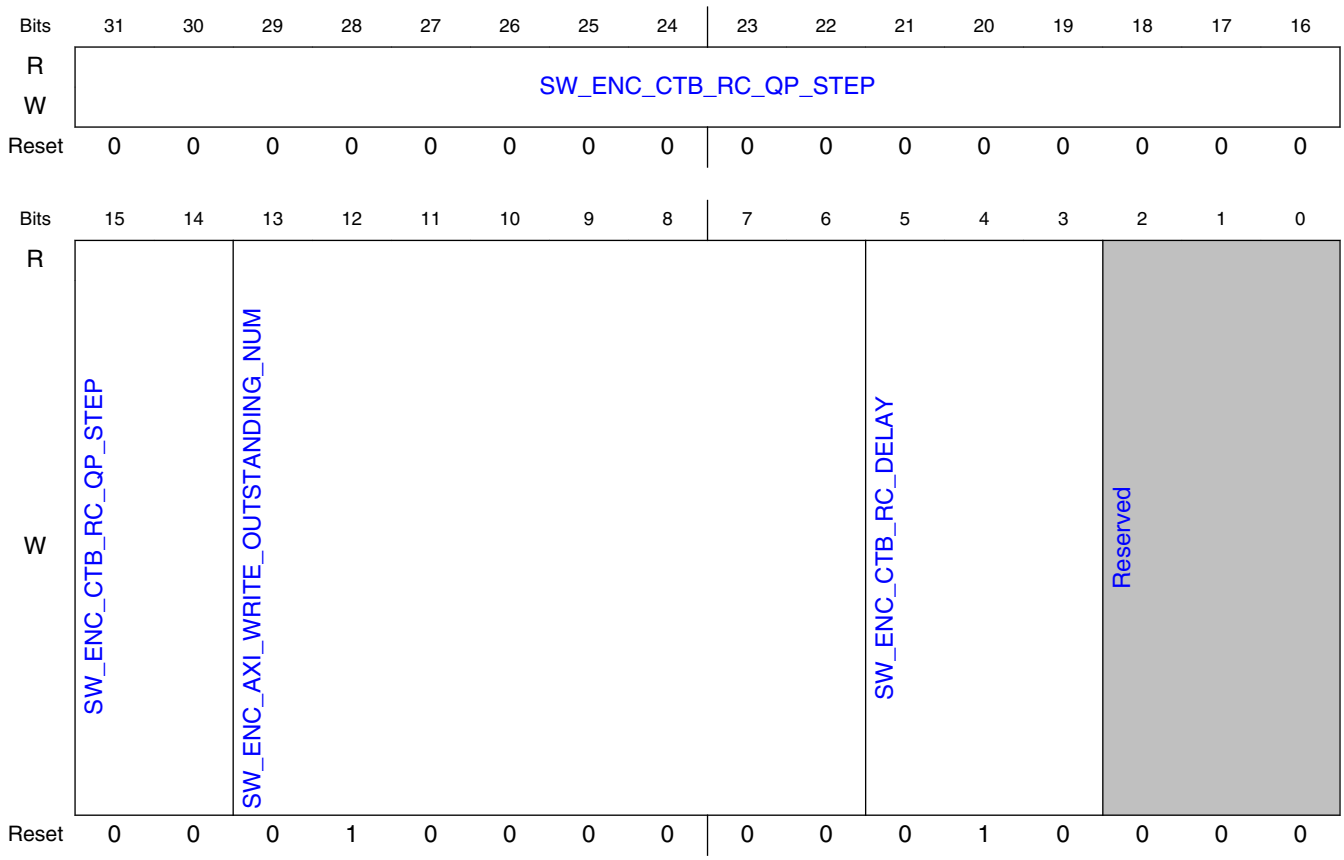
16.3.5.2.195.1 Offset

Register	Offset
swreg246	3D8h

16.3.5.2.195.2 Function

CTB RC Control 3

16.3.5.2.195.3 Diagram



16.3.5.2.195.4 Fields

Field	Function
31-14 SW_ENC_CTB_RC_QP_STEP	Maximum QP step per CTB for purpose of rate control. Q16.
13-6 SW_ENC_AXI_WRITE_OUTSTANDING_NUM	AXI outstanding number of write operation
5-3 SW_ENC_CTB_RC_DELAY	Feedback delay of information for CTB RC
2-0 —	Reserved

16.3.5.2.196 CTB RC Control 4 (swreg247)

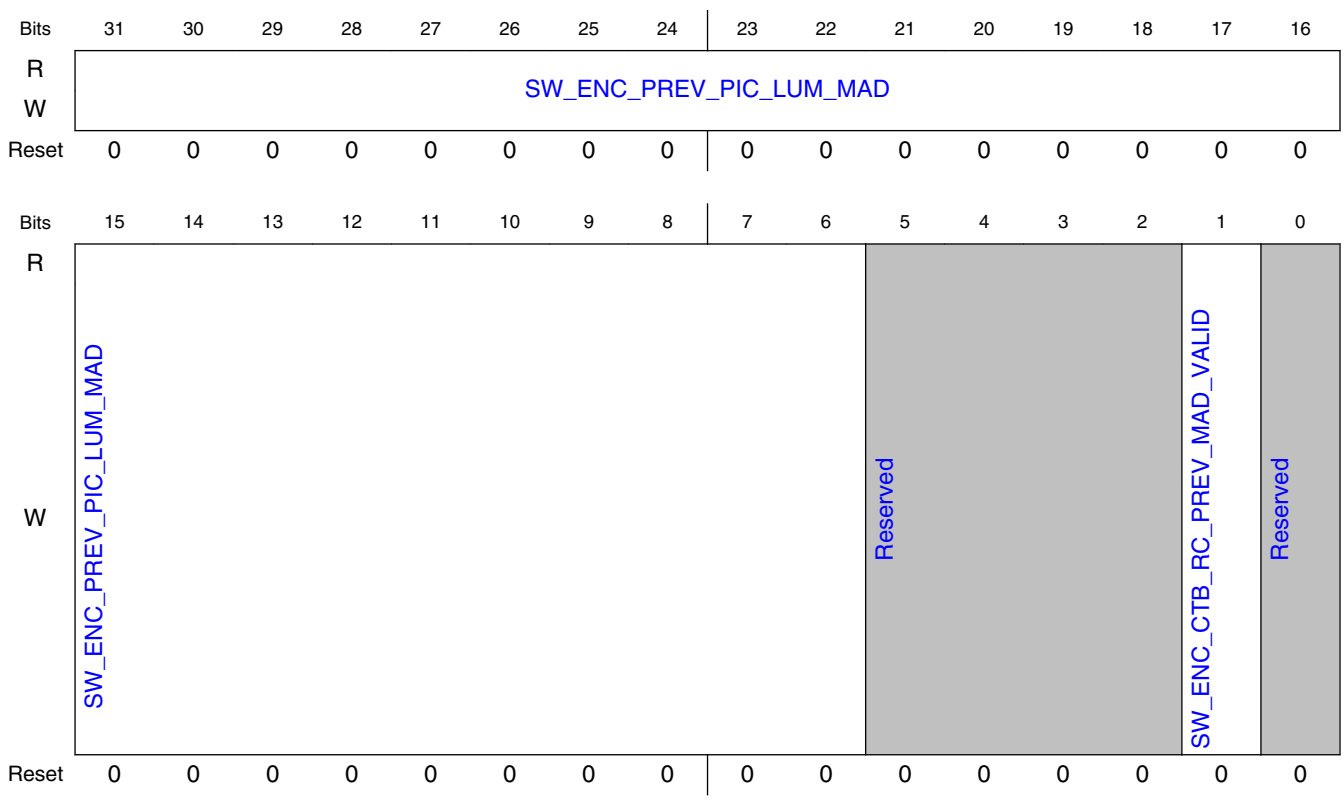
16.3.5.2.196.1 Offset

Register	Offset
swreg247	3DCh

16.3.5.2.196.2 Function

CTB RC Control 4

16.3.5.2.196.3 Diagram



16.3.5.2.196.4 Fields

Field	Function
31-6 SW_ENC_PREV_PIC_LUM_MAD	Luma MAD of the predict picture. Input for previous pic. Output for current pic.

Table continues on the next page...

Field	Function
5-2 —	Reserved
1 SW_ENC_CTB_RC_PREV_MAD_VALID	Prev luma MADs are valid or not for ctb rc. 0b - no 1b - yes.
0 —	Reserved

16.3.5.2.197 CTB RC Control 5 (swreg248)

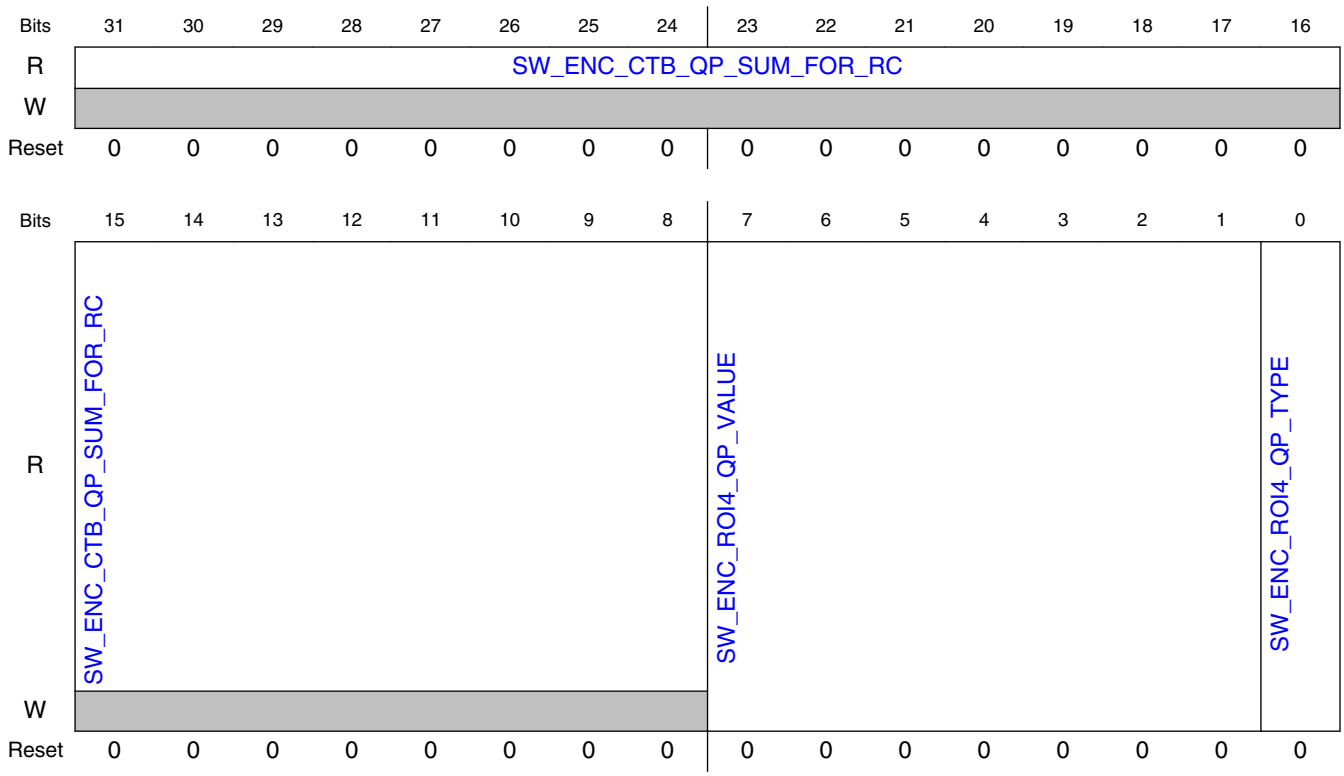
16.3.5.2.197.1 Offset

Register	Offset
swreg248	3E0h

16.3.5.2.197.2 Function

CTB RC Control 5

16.3.5.2.197.3 Diagram



16.3.5.2.197.4 Fields

Field	Function
31-8 SW_ENC_CTB_QP_SUM_FOR_RC	Sum of CTB QP used for rate control. not include QP delta for quality adjust or ROI.
7-1 SW_ENC_ROI4_QP_VALUE	ROI4 qp value. delta or absolute value
0 SW_ENC_ROI4_QP_TYPE	ROI4 qp type. 0b - delta 1b - Absolute value

16.3.5.2.198 register extension for 8K width (swreg249)

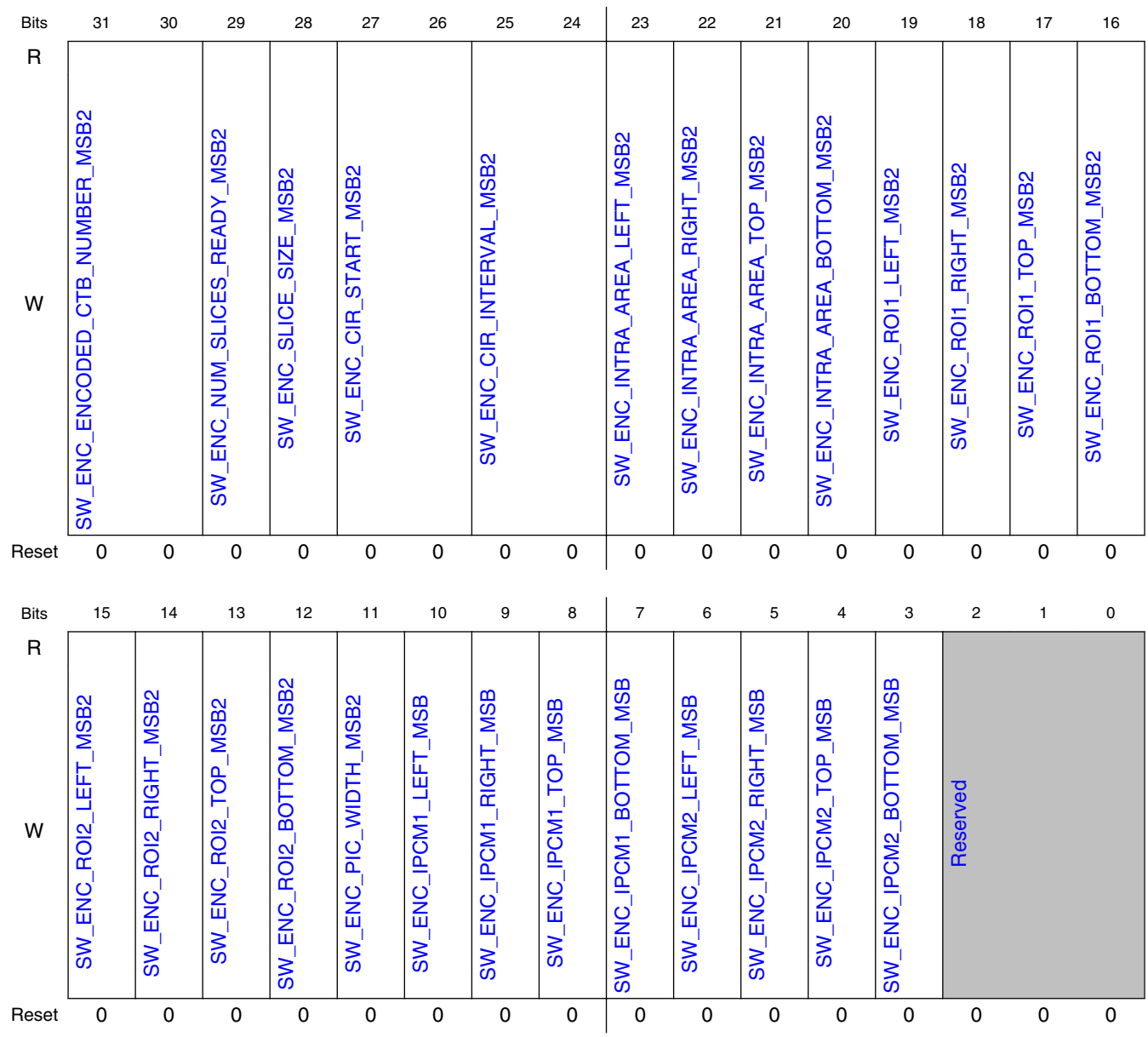
16.3.5.2.198.1 Offset

Register	Offset
swreg249	3E4h

16.3.5.2.198.2 Function

register extension for 8K width

16.3.5.2.198.3 Diagram



16.3.5.2.198.4 Fields

Field	Function
31-30 SW_ENC_ENCODED_CTB_NUMBER_MSB2	MB count output. Max 512*256. bit[17:18]
29 SW_ENC_NUM_SLICES_READY_MSB2	HEVC amount of completed slices. bit[10]
28 SW_ENC_SLICE_SIZE_MSB2	slice size in ctu row. bit[9]
27-26 SW_ENC_CIR_START_MSB2	cir start. bit[18:19]
25-24 SW_ENC_CIR_INTERVAL_MSB2	Cyclic Intra Refresh. bit[18:19]
23 SW_ENC_INTRA_AREA_LEFT_MSB2	intra Area Left. bit[9]
22 SW_ENC_INTRA_AREA_RIGHT_MSB2	intra Area Right. bit[9]
21 SW_ENC_INTRA_AREA_TOP_MSB2	intra Area Top. bit[9]
20 SW_ENC_INTRA_AREA_BOTTOM_MSB2	intra Area Bottom. bit[9]
19 SW_ENC_ROI1_LEFT_MSB2	ROI1 Area Left. bit[9]
18 SW_ENC_ROI1_RIGHT_MSB2	ROI1 Area Right. bit[9]
17 SW_ENC_ROI1_TOP_MSB2	ROI1 Area Top. bit[9]

Table continues on the next page...

Field	Function
16 SW_ENC_ROI1_BOTTOM_MSB2	ROI1 Area Bottom. bit[9]
15 SW_ENC_ROI2_LEFT_MSB2	ROI2 Area Left. bit[9]
14 SW_ENC_ROI2_RIGHT_MSB2	ROI2 Area Right. bit[9]
13 SW_ENC_ROI2_TOP_MSB2	ROI2 Area Top. bit[9]
12 SW_ENC_ROI2_BOTTOM_MSB2	ROI2 Area Bottom. bit[9]
11 SW_ENC_PIC_WIDTH_MSB2	Encoded width. lumWidthMsb2(unit 8 pixels). bit[12]
10 SW_ENC_IPCM1_LEFT_MSB	IPCM1 Area Left bit[9]
9 SW_ENC_IPCM1_RIGHT_MSB	IPCM1 Area Right bit[9]
8 SW_ENC_IPCM1_TOP_MSB	IPCM1 Area Top bit[9]
7 SW_ENC_IPCM1_BOTTOM_MSB	IPCM2 Area Bottom bit[9]
6 SW_ENC_IPCM2_LEFT_MSB	IPCM2 Area Left bit[9]
5 SW_ENC_IPCM2_RIGHT_MSB	IPCM2 Area Right bit[9]
4 SW_ENC_IPCM2_TOP_MSB	IPCM2 Area Top bit[9]
3	IPCM1 Area Bottom bit[9]

Table continues on the next page...

Field	Function
SW_ENC_IPCM2_BOTTOM_MSB	
2-0 —	Reserved

16.3.5.2.199 Global MV Control 0 (swreg250)

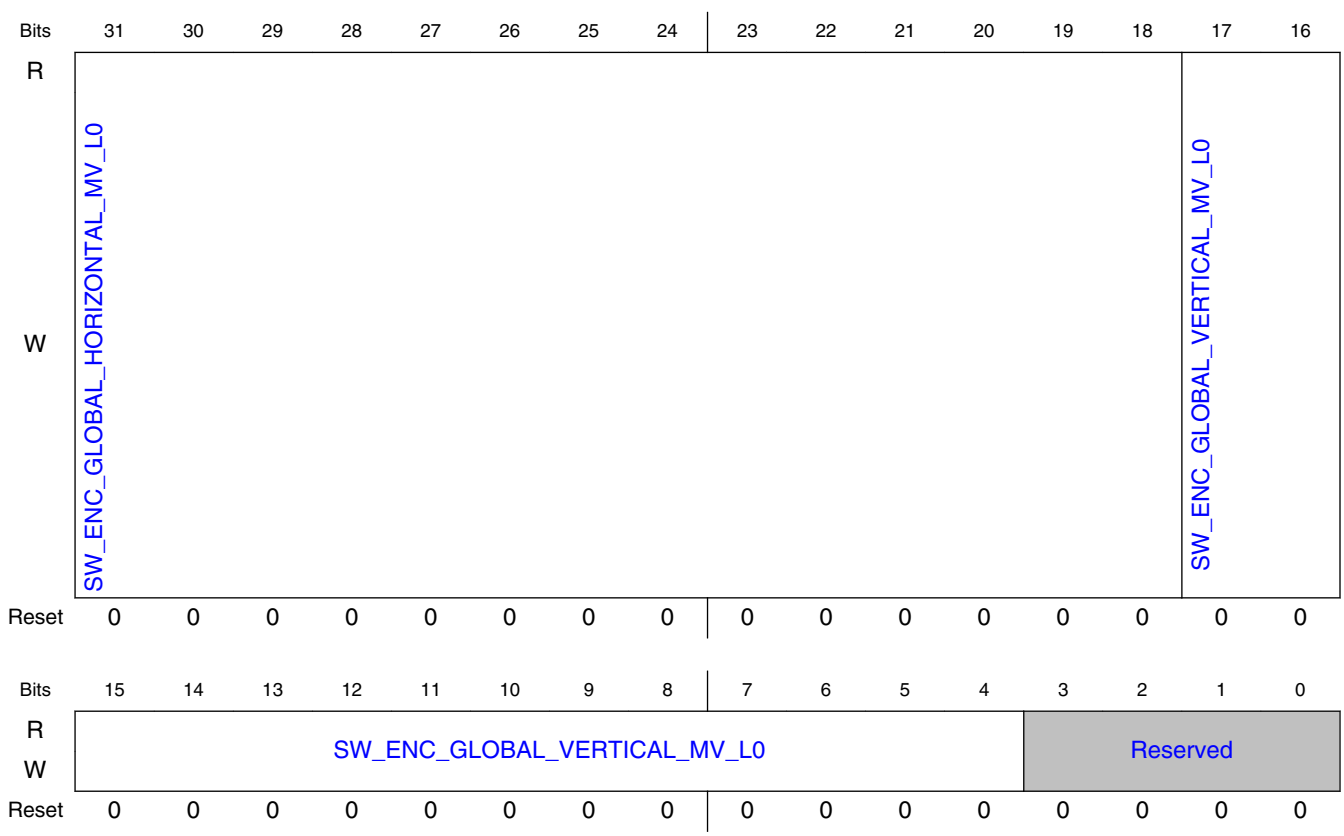
16.3.5.2.199.1 Offset

Register	Offset
swreg250	3E8h

16.3.5.2.199.2 Function

Global MV Control 0

16.3.5.2.199.3 Diagram



16.3.5.2.199.4 Fields

Field	Function
31-18 SW_ENC_GLOBAL_HORIZONTAL_MV_L0	Global horizontal MV for LIST0 in integer pixel.
17-4 SW_ENC_GLOBAL_VERTICAL_MV_L0	Global vertical MV for LIST0 in integer pixel.
3-0 —	Reserved

16.3.5.2.200 Global MV Control 1 (swreg251)

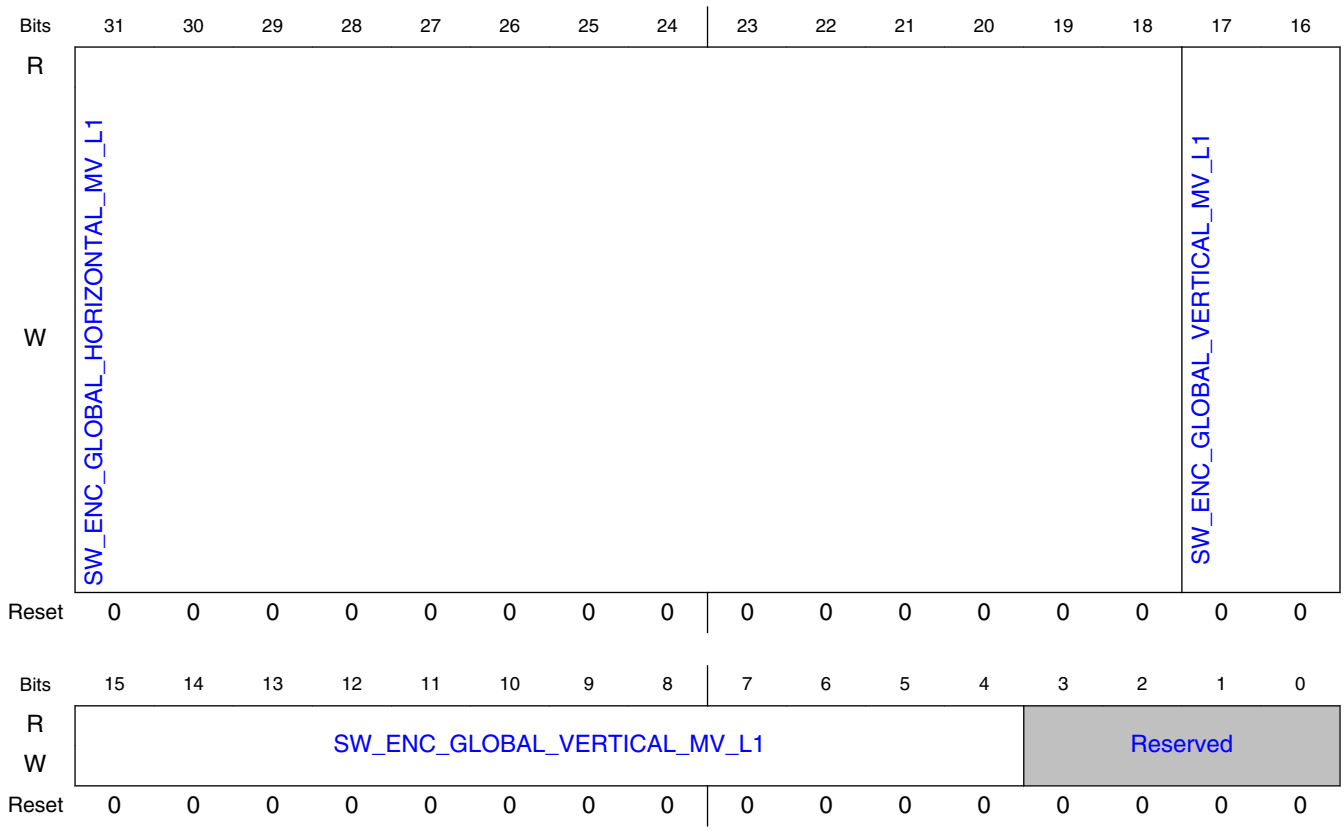
16.3.5.2.200.1 Offset

Register	Offset
swreg251	3ECh

16.3.5.2.200.2 Function

Global MV Control 1

16.3.5.2.200.3 Diagram



16.3.5.2.200.4 Fields

Field	Function
31-18 SW_ENC_GLOBAL_HORIZONTAL_MV_L1	Global horizontal MV for LIST1 in integer pixel.
17-4 SW_ENC_GLOBAL_VERTICAL_MV_L1	Global vertical MV for LIST1 in integer pixel.
3-0 —	Reserved

16.3.5.2.201 ROI3 Area (swreg252)

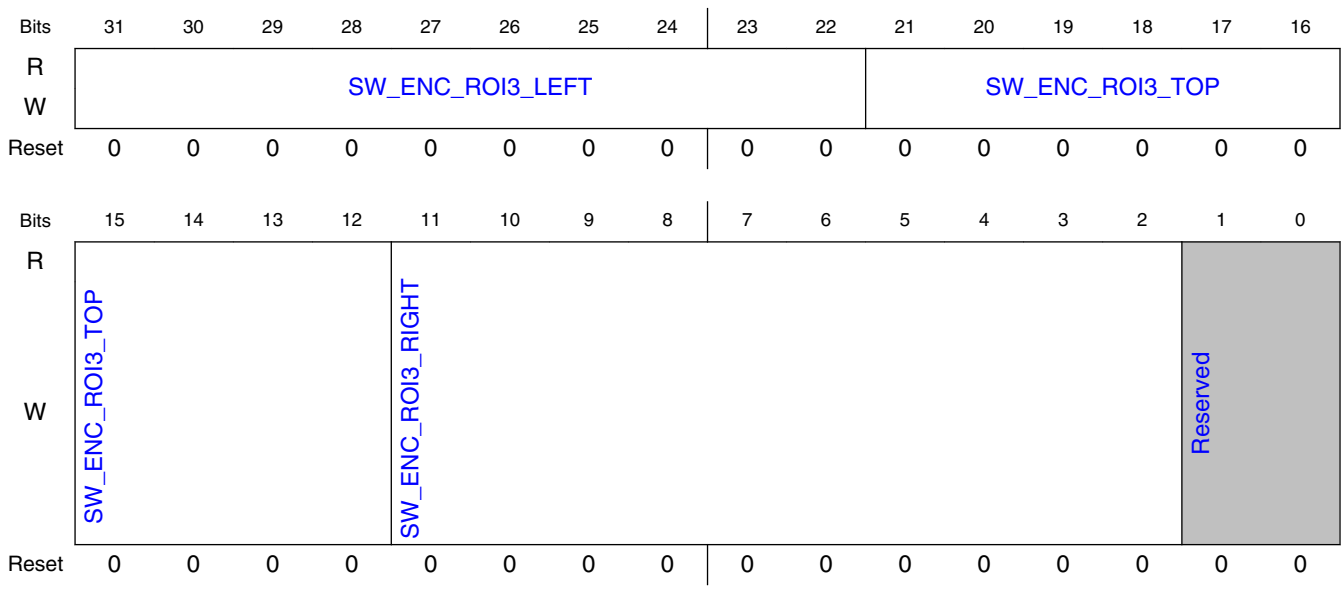
16.3.5.2.201.1 Offset

Register	Offset
swreg252	3F0h

16.3.5.2.201.2 Function

ROI3 Area

16.3.5.2.201.3 Diagram



16.3.5.2.201.4 Fields

Field	Function
31-22 SW_ENC_ROI3_LEFT	ROI3 Area Left
21-12 SW_ENC_ROI3_TOP	ROI3 Area Top
11-2 SW_ENC_ROI3_RIGHT	ROI3 Area Right
1-0 —	Reserved

16.3.5.2.202 ROI3&4 Area (swreg253)

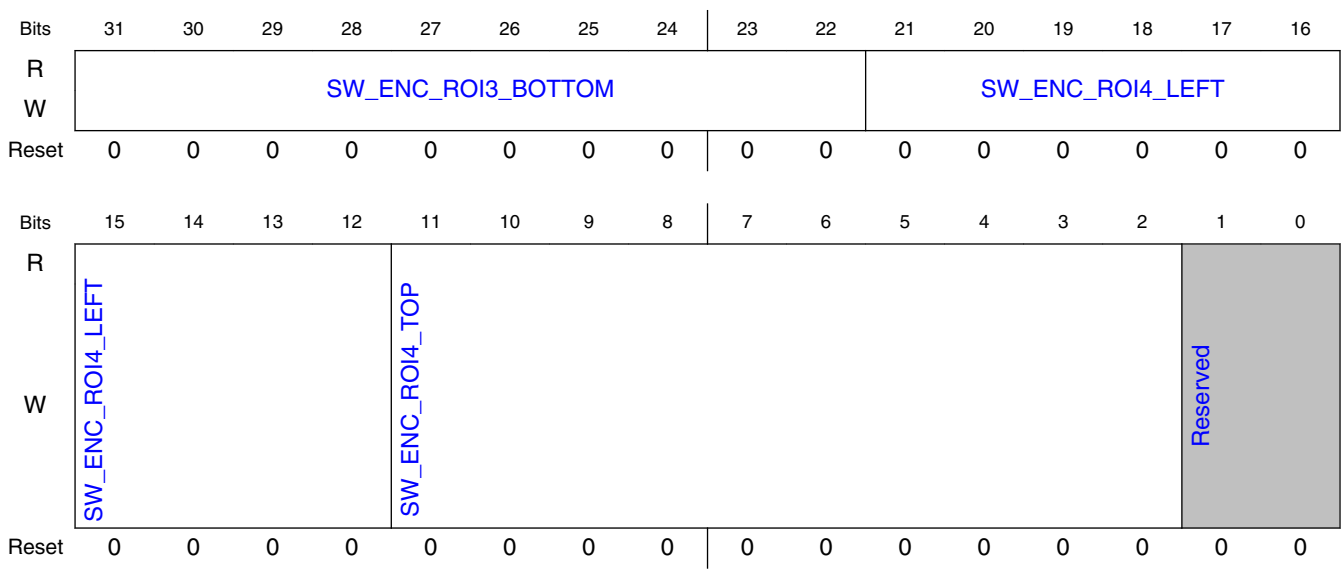
16.3.5.2.202.1 Offset

Register	Offset
swreg253	3F4h

16.3.5.2.202.2 Function

ROI3&4 Area

16.3.5.2.202.3 Diagram



16.3.5.2.202.4 Fields

Field	Function
31-22 SW_ENC_ROI3_BOTTOM	ROI3 Area Bottom
21-12 SW_ENC_ROI4_LEFT	ROI4 Area Left
11-2	ROI4 Area Top

Table continues on the next page...

Field	Function
SW_ENC_ROI4_TOP	
1-0 —	Reserved

16.3.5.2.203 ROI4&5 Area (swreg254)

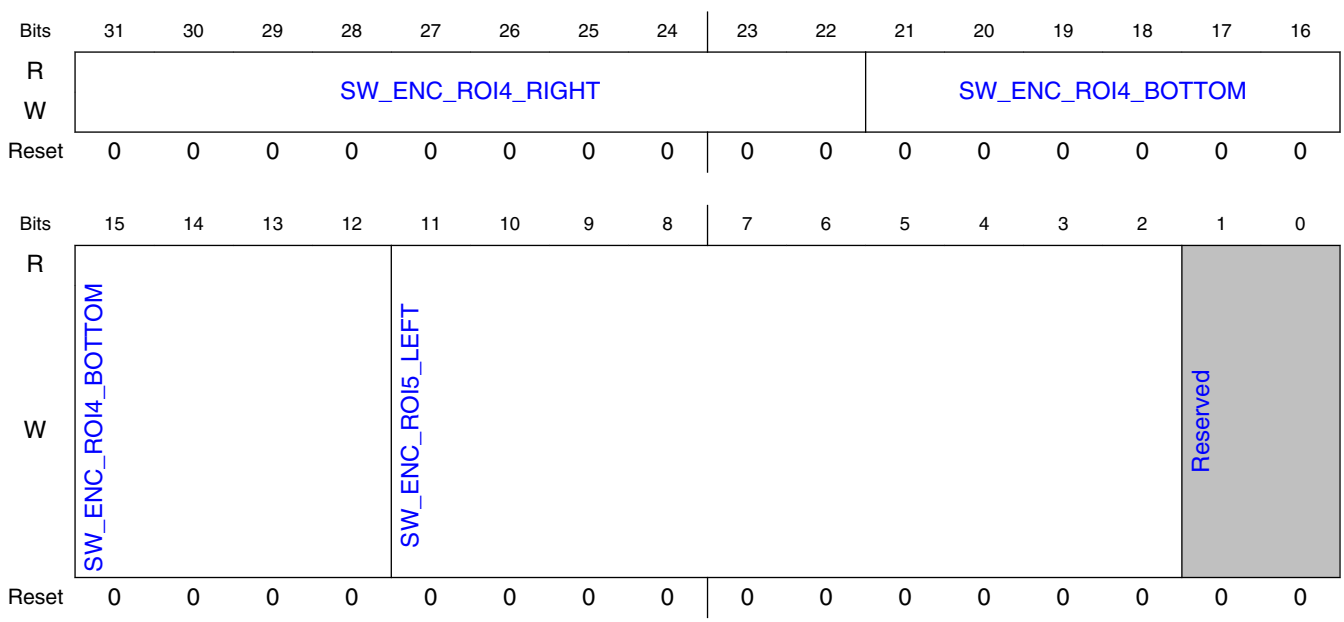
16.3.5.2.203.1 Offset

Register	Offset
swreg254	3F8h

16.3.5.2.203.2 Function

ROI4&5 Area

16.3.5.2.203.3 Diagram



16.3.5.2.203.4 Fields

Field	Function
31-22	ROI4 Area Right

Table continues on the next page...

Field	Function
SW_ENC_ROI4_RIGHT	
21-12 SW_ENC_ROI4_BOTTOM	ROI4 Area Bottom
11-2 SW_ENC_ROI5_LEFT	ROI5 Area Left
1-0 —	Reserved

16.3.5.2.204 ROI5 Area (swreg255)

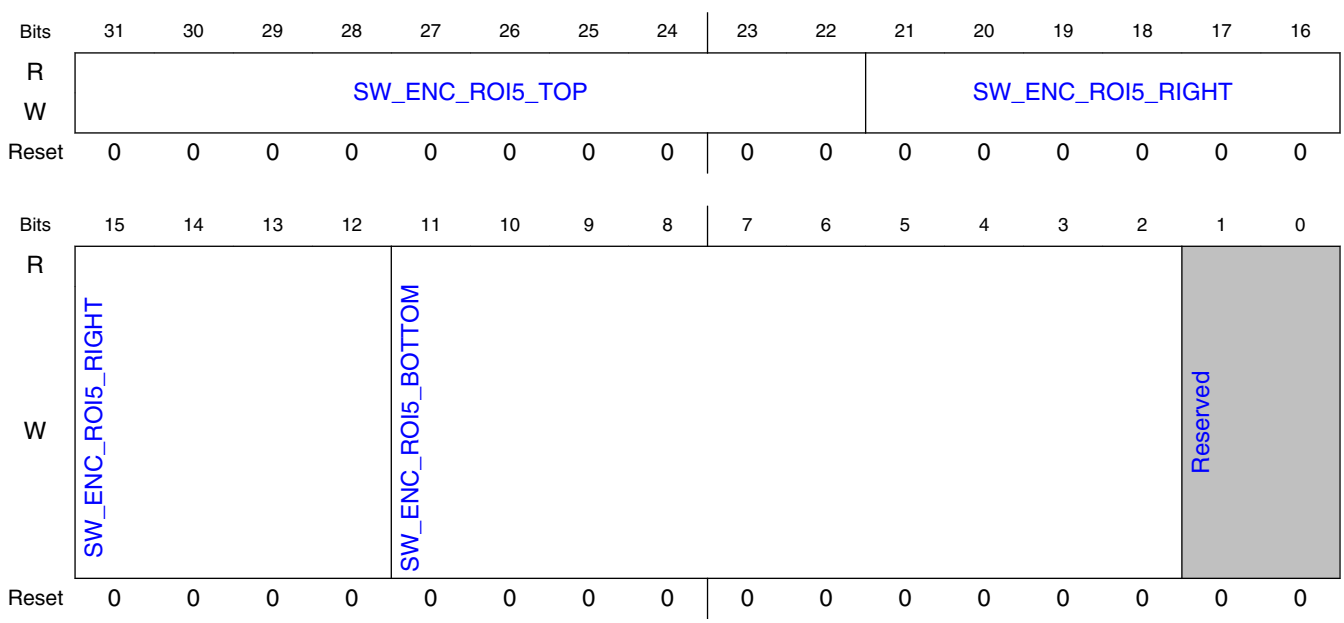
16.3.5.2.204.1 Offset

Register	Offset
swreg255	3FCh

16.3.5.2.204.2 Function

ROI5 Area

16.3.5.2.204.3 Diagram



16.3.5.2.204.4 Fields

Field	Function
31-22 SW_ENC_ROI5_TOP	ROI5 Area Top
21-12 SW_ENC_ROI5_RIGHT	ROI5 Area Right
11-2 SW_ENC_ROI5_BOTTOM	ROI5 Area Bottom
1-0 —	Reserved

16.3.5.2.205 ROI6 Area (swreg256)

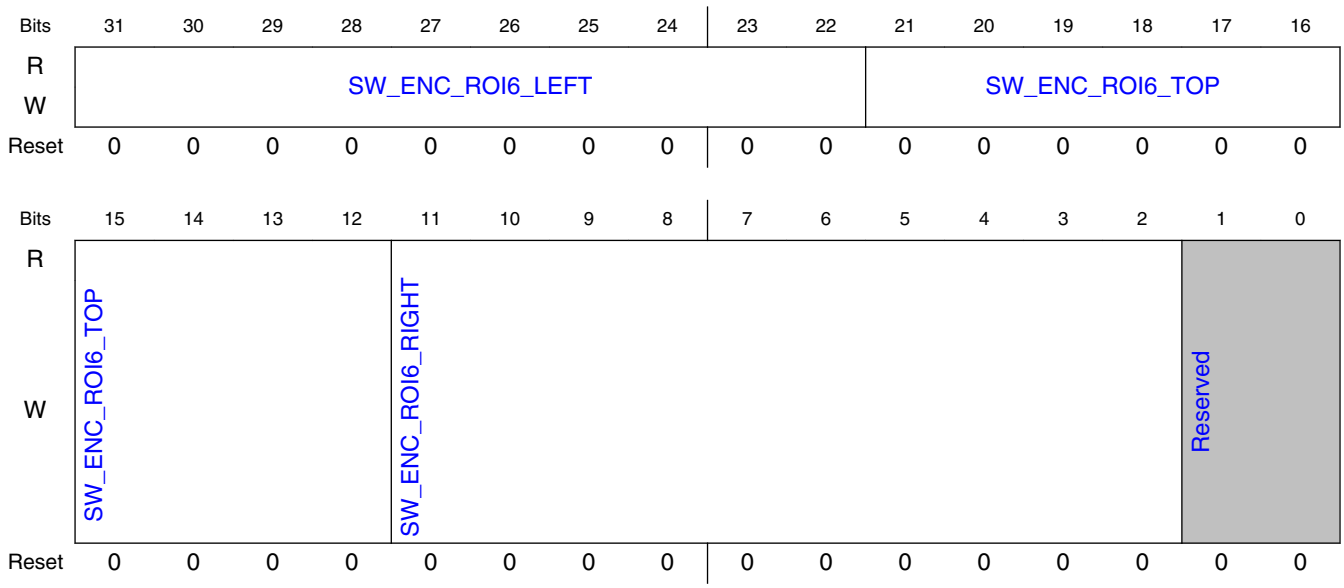
16.3.5.2.205.1 Offset

Register	Offset
swreg256	400h

16.3.5.2.205.2 Function

ROI6 Area

16.3.5.2.205.3 Diagram



16.3.5.2.205.4 Fields

Field	Function
31-22 SW_ENC_ROI6_LEFT	ROI6 Area Left
21-12 SW_ENC_ROI6_TOP	ROI6 Area Top
11-2 SW_ENC_ROI6_RIGHT	ROI6 Area Right
1-0 —	Reserved

16.3.5.2.206 ROI6&7 Area (swreg257)

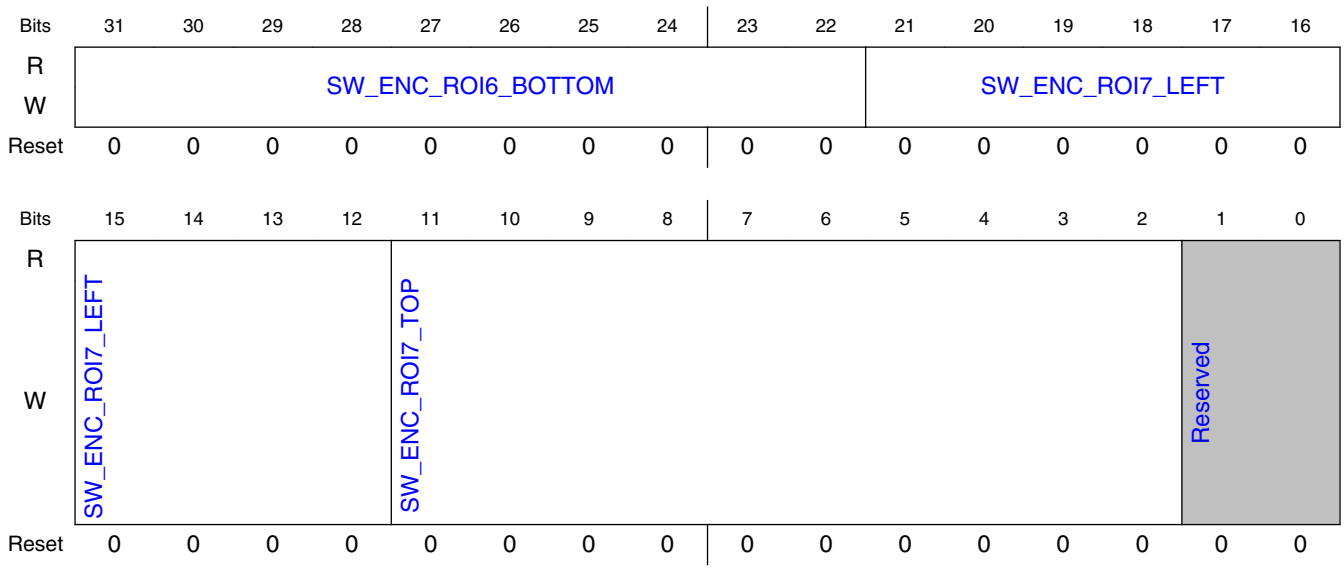
16.3.5.2.206.1 Offset

Register	Offset
swreg257	404h

16.3.5.2.206.2 Function

ROI6&7 Area

16.3.5.2.206.3 Diagram



16.3.5.2.206.4 Fields

Field	Function
31-22 SW_ENC_ROI6_BOTTOM	ROI6 Area Bottom
21-12 SW_ENC_ROI7_LEFT	ROI7 Area Left
11-2 SW_ENC_ROI7_TOP	ROI7 Area Top
1-0 —	Reserved

16.3.5.2.207 ROI7&8 Area (swreg258)

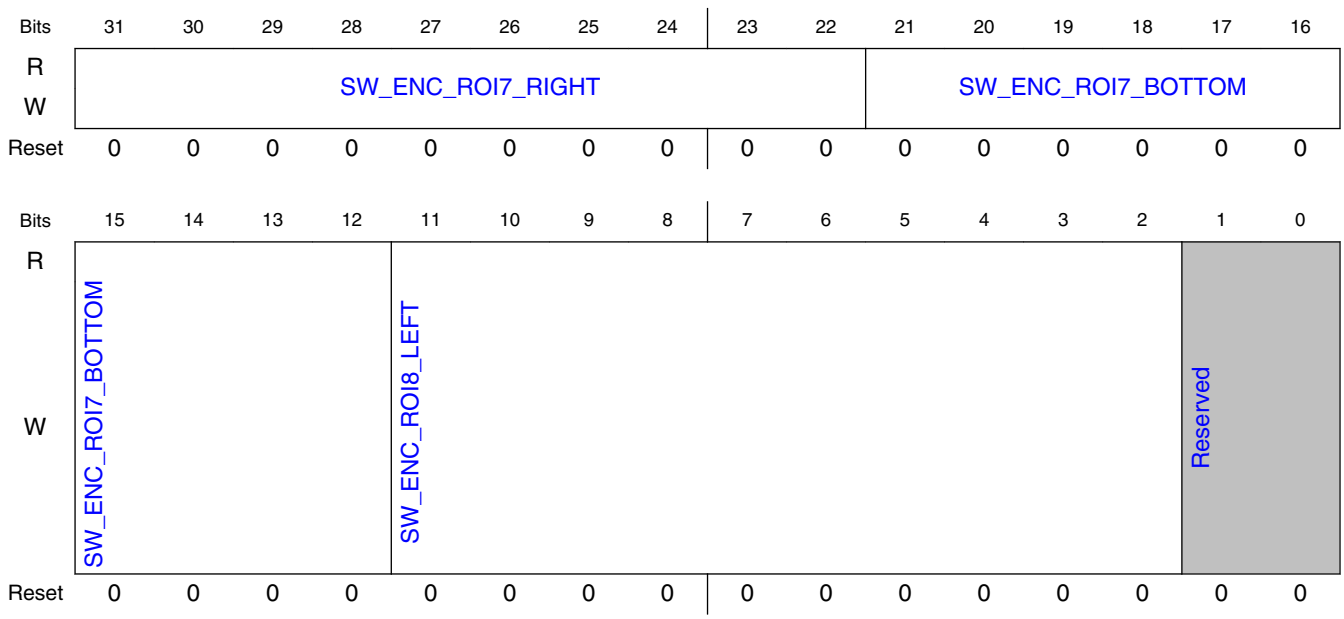
16.3.5.2.207.1 Offset

Register	Offset
swreg258	408h

16.3.5.2.207.2 Function

ROI7&8 Area

16.3.5.2.207.3 Diagram



16.3.5.2.207.4 Fields

Field	Function
31-22 SW_ENC_ROI7_RIGHT	ROI7 Area Right
21-12 SW_ENC_ROI7_BOTTOM	ROI7 Area Bottom
11-2 SW_ENC_ROI8_LEFT	ROI8 Area Left
1-0 —	Reserved

16.3.5.2.208 ROI8 Area (swreg259)

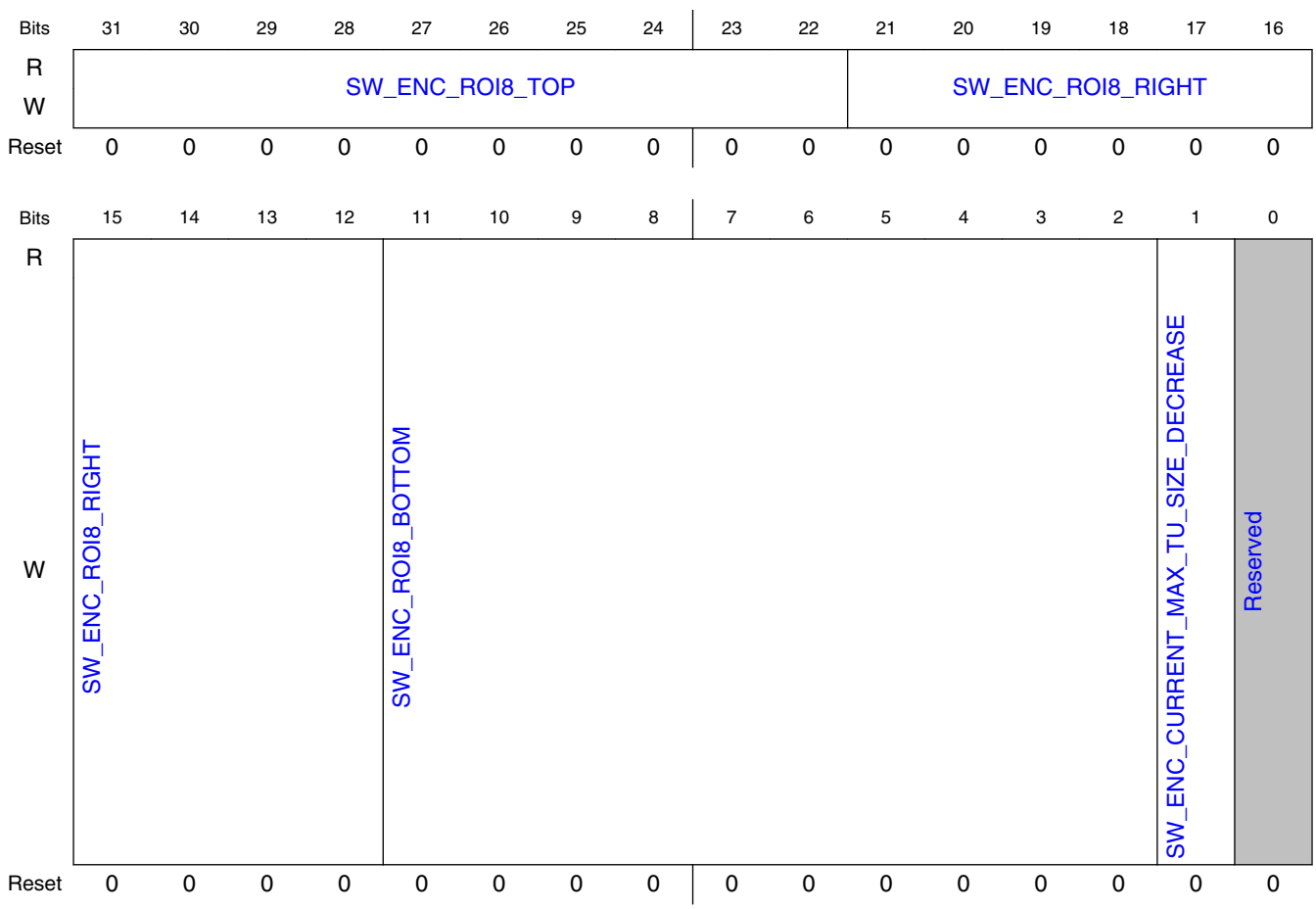
16.3.5.2.208.1 Offset

Register	Offset
swreg259	40Ch

16.3.5.2.208.2 Function

ROI8 Area

16.3.5.2.208.3 Diagram



16.3.5.2.208.4 Fields

Field	Function
31-22 SW_ENC_ROI8_TOP	ROI8 Area Top
21-12 SW_ENC_ROI8_RIGHT	ROI8 Area Right
11-2 SW_ENC_ROI8_BOTTOM	ROI8 Area Bottom
1 SW_ENC_CURRENT_MAX_TU_SIZE_DECREASE	Decrease max tu size in current frame.[0] 0b - max tu size 32. 1b - max tu size 16.
0 —	Reserved

16.3.5.2.209 ROI qp (swreg260)

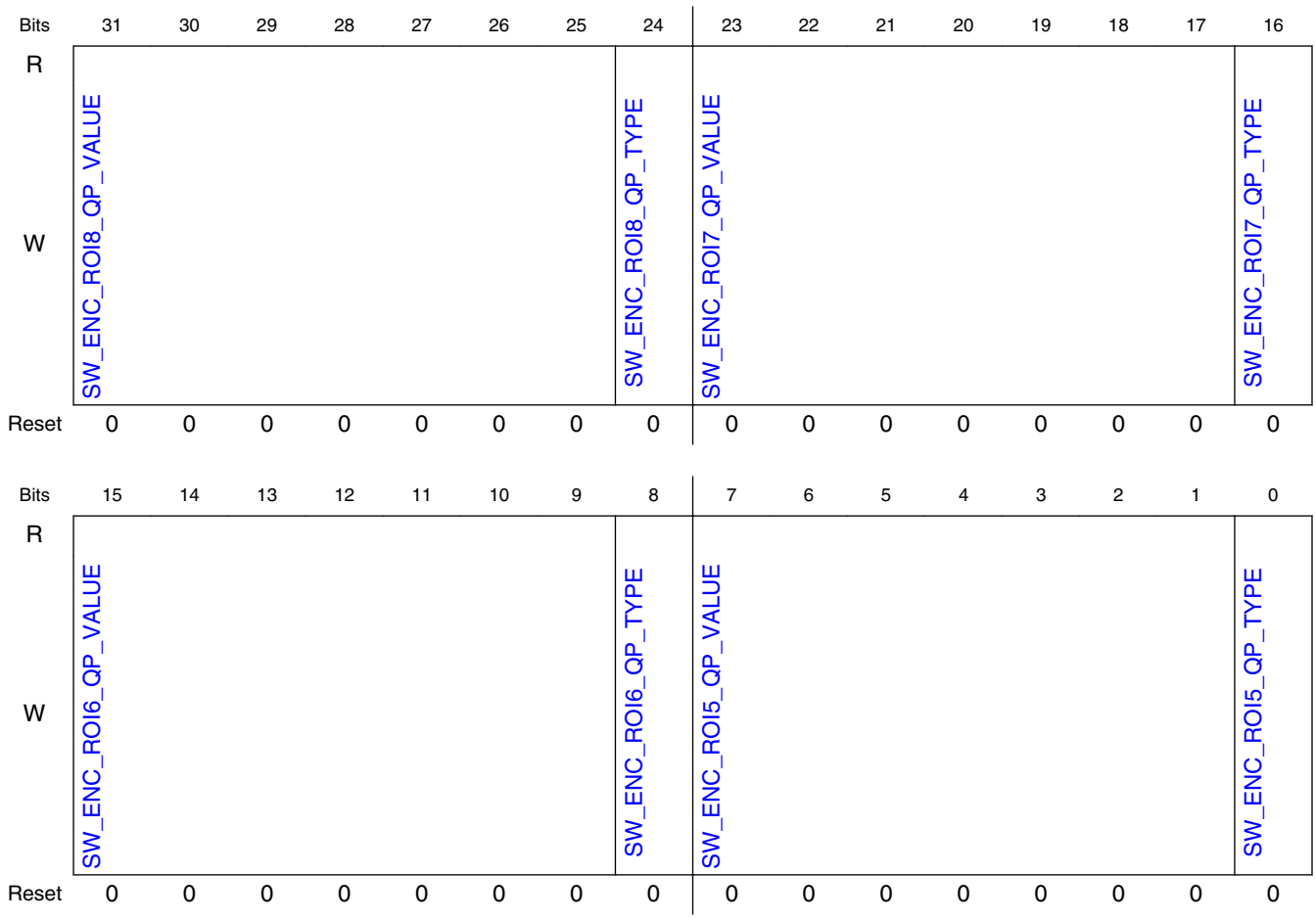
16.3.5.2.209.1 Offset

Register	Offset
swreg260	410h

16.3.5.2.209.2 Function

ROI qp

16.3.5.2.209.3 Diagram



16.3.5.2.209.4 Fields

Field	Function
31-25 SW_ENC_ROI8_QP_VALUE	ROI8 qp value. delta or absolute value
24 SW_ENC_ROI8_QP_TYPE	ROI8 qp type. 0b - delta 1b - Absolute value
23-17 SW_ENC_ROI7_QP_VALUE	ROI7 qp value. delta or absolute value
16 SW_ENC_ROI7_QP_TYPE	ROI7 qp type. 0b - delta 1b - Absolute value
15-9	ROI6 qp value. delta or absolute value

Table continues on the next page...

Field	Function
SW_ENC_ROI6 _QP_VALUE	
8 SW_ENC_ROI6 _QP_TYPE	ROI6 qp type. 0b - delta 1b - Absolute value
7-1 SW_ENC_ROI5 _QP_VALUE	ROI5 qp value. delta or absolute value
0 SW_ENC_ROI5 _QP_TYPE	ROI5 qp type. 0b - delta 1b - Absolute value

16.3.5.2.210 Stride Control (swreg261)

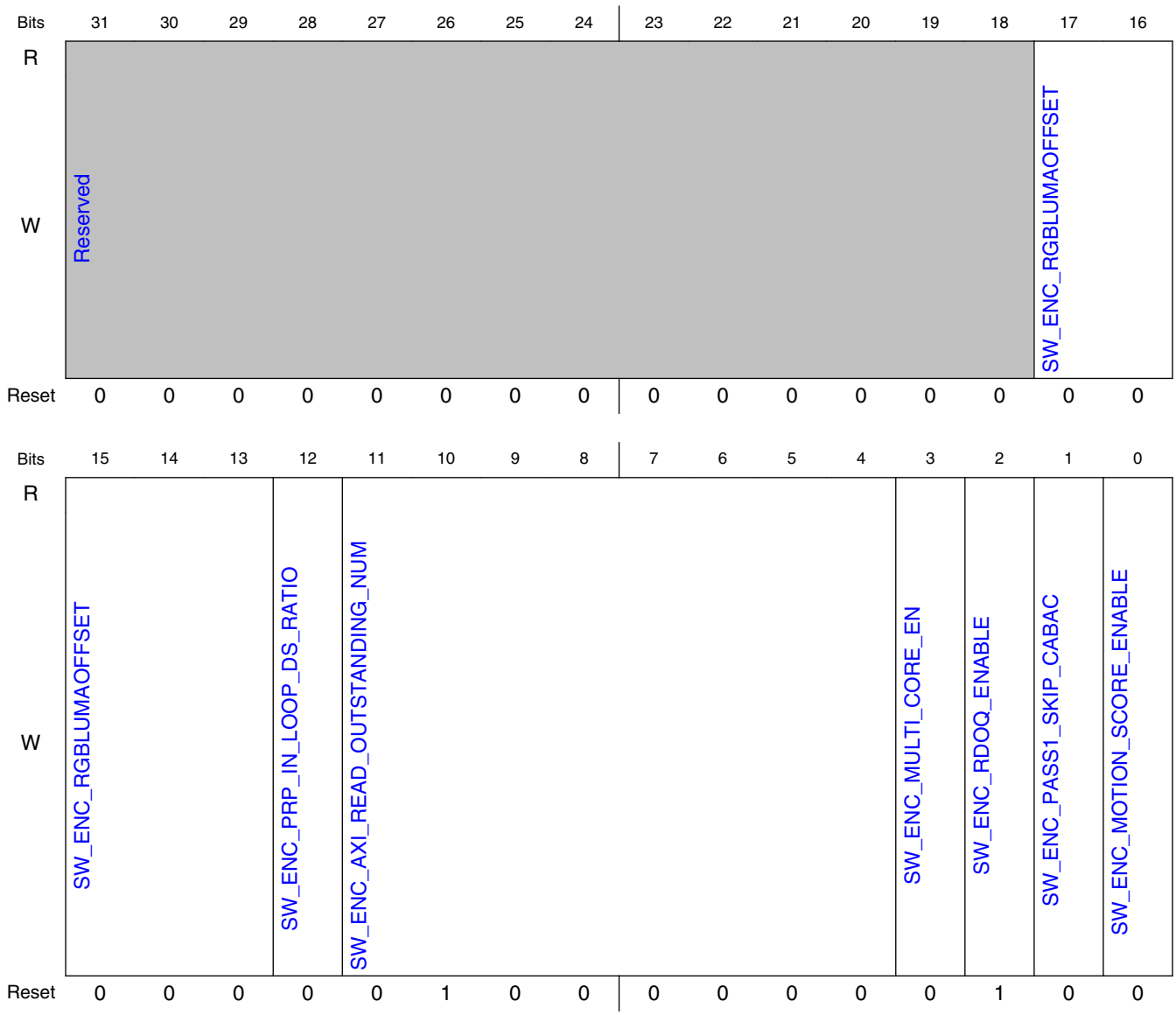
16.3.5.2.210.1 Offset

Register	Offset
swreg261	414h

16.3.5.2.210.2 Function

Stride Control

16.3.5.2.210.3 Diagram



16.3.5.2.210.4 Fields

Field	Function
31-18 —	Reserved
17-13 SW_ENC_RGBLUMAOFFSET	RGB to YUV conversion luma offset
12	down-scaling ratio of In-loop downscaler in prp

Table continues on the next page...

Field	Function
SW_ENC_PRP_IN_LOOP_DS_RATIO	
11-4 SW_ENC_AXI_READ_OUTSTANDING_NUM	AXI outstanding number of read operation
3 SW_ENC_MULTI_CORE_EN	Enable multi-core encoding mode
2 SW_ENC_RDOQ_ENABLE	Enable RDOQ
1 SW_ENC_PASS1_SKIP_CABAC	CABAC SKIP for pass one.
0 SW_ENC_MOTION_SCORE_ENABLE	Enable motion score computing for 2-pass Agop

16.3.5.2.211 Multicore sync ctrl (swreg265)

16.3.5.2.211.1 Offset

Register	Offset
swreg265	424h

16.3.5.2.211.2 Function

Multicore sync ctrl

16.3.5.2.211.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_REF_READY_THRESHOLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_DDR_POLLING_INTERVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.211.4 Fields

Field	Function
31-16 SW_ENC_REF_READY_THRESHOLD	EXTRA_LINES_NUM bigger vertical search window ready
15-0 SW_ENC_DDR_POLLING_INTERVAL	Interval cycles for HW reading the DDR RECON_PIXELLINES_READY when value not bigger than search window.

16.3.5.2.212 Multicore sync address L0 LSB (swreg266)

16.3.5.2.212.1 Offset

Register	Offset
swreg266	428h

16.3.5.2.212.2 Function

Multicore sync address L0 LSB

16.3.5.2.212.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MULTICORE_SYNC_L0_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MULTICORE_SYNC_L0_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.212.4 Fields

Field	Function
31-0 SW_ENC_MULTICORE_SYNC_L0_ADDR	For multicore sync. Address LSB of RECON_PIXELLINES_READY for ref buffer L0

16.3.5.2.213 Multicore sync address L0 MSB (swreg267)

16.3.5.2.213.1 Offset

Register	Offset
swreg267	42Ch

16.3.5.2.213.2 Function

Multicore sync address L0 MSB

16.3.5.2.213.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MULTICORE_SYNC_L0_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MULTICORE_SYNC_L0_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.213.4 Fields

Field	Function
31-0 SW_ENC_MULTICORE_SYNC_L0_ADDR_MSB	For multicore sync. Address MSB of RECON_PIXELLINES_READY for ref buffer L0

16.3.5.2.214 Multicore sync address L1 LSB (swreg268)

16.3.5.2.214.1 Offset

Register	Offset
swreg268	430h

16.3.5.2.214.2 Function

Multicore sync address L1 LSB

16.3.5.2.214.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MULTICORE_SYNC_L1_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MULTICORE_SYNC_L1_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.214.4 Fields

Field	Function
31-0 SW_ENC_MULTICORE_SYNC_L1_ADDR	For multicore sync. Address LSB of RECON_PIXELLINES_READY for ref buffer L1

16.3.5.2.215 Multicore sync address L1 MSB (swreg269)

16.3.5.2.215.1 Offset

Register	Offset
swreg269	434h

16.3.5.2.215.2 Function

Multicore sync address L1 MSB

16.3.5.2.215.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MULTICORE_SYNC_L1_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MULTICORE_SYNC_L1_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.215.4 Fields

Field	Function
31-0 SW_ENC_MULTICORE_SYNC_L1_ADDR_MSB	For multicore sync. Address MSB of RECON_PIXELLINES_READY for ref buffer L1

16.3.5.2.216 Multicore sync address recon LSB (swreg270)

16.3.5.2.216.1 Offset

Register	Offset
swreg270	438h

16.3.5.2.216.2 Function

Multicore sync address recon LSB

16.3.5.2.216.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MULTICORE_SYNC_REC_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MULTICORE_SYNC_REC_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.216.4 Fields

Field	Function
31-0 SW_ENC_MULTICORE_SYNC_REC_ADDR	For multicore sync. Address LSB of RECON_PIXELLINES_READY for recon buffer

16.3.5.2.217 Multicore sync address recon MSB (swreg271)

16.3.5.2.217.1 Offset

Register	Offset
swreg271	43Ch

16.3.5.2.217.2 Function

Multicore sync address recon MSB

16.3.5.2.217.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_MULTICORE_SYNC_REC_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_MULTICORE_SYNC_REC_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.217.4 Fields

Field	Function
31-0 SW_ENC_MULTICORE_SYNC_REC_ADDR_MSB	For multicore sync. Address MSB of RECON_PIXELLINES_READY for recon buffer

16.3.5.2.218 Programmable AXI urgent sideband signals (swreg272)

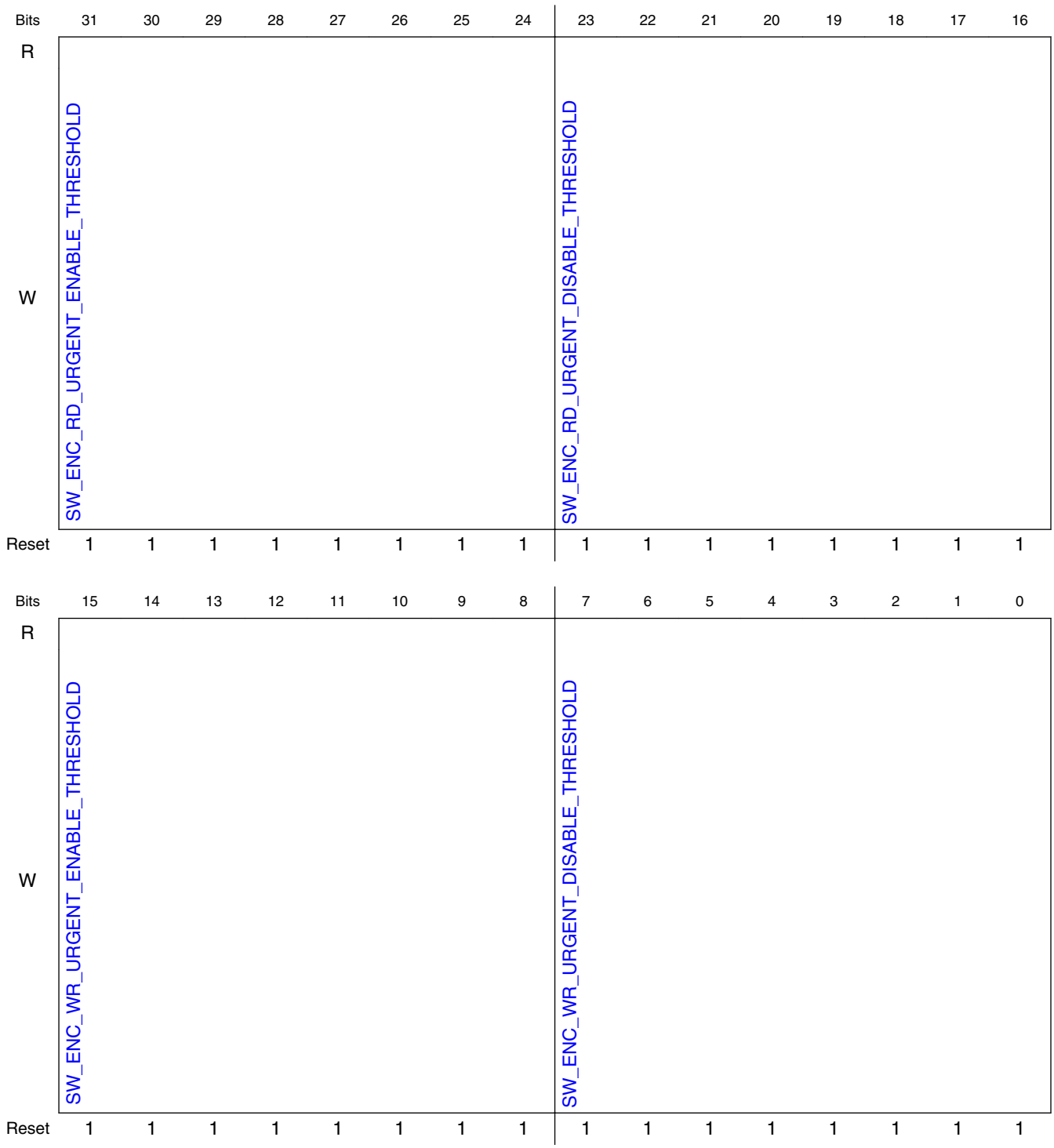
16.3.5.2.218.1 Offset

Register	Offset
swreg272	440h

16.3.5.2.218.2 Function

Programmable AXI urgent sideband signals

16.3.5.2.218.3 Diagram



16.3.5.2.218.4 Fields

Field	Function
31-24 SW_ENC_RD_URGENT_ENABLE_THRESHOLD	axi_rd_urgent=1 when total number of rd cmds is bigger or equal to threshold.255=disable
23-16 SW_ENC_RD_URGENT_DISABLE_THRESHOLD	axi_rd_urgent=0 when total number of rd cmds is smaller than threshold.255=disable
15-8 SW_ENC_WR_URGENT_ENABLE_THRESHOLD	axi_wr_urgent=1 when total number of wr cmds is bigger or equal to threshold.255=disable
7-0 SW_ENC_WR_URGENT_DISABLE_THRESHOLD	axi_wr_urgent=0 when total number of wr cmds is smaller than threshold.255=disable

16.3.5.2.219 roimap cu ctrl index address LSB (swreg273)

16.3.5.2.219.1 Offset

Register	Offset
swreg273	444h

16.3.5.2.219.2 Function

roimap cu ctrl index address LSB

16.3.5.2.219.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROIMAP_CUCTRL_INDEX_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROIMAP_CUCTRL_INDEX_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.219.4 Fields

Field	Function
31-0 SW_ENC_ROIMAP_CUCTRL_INDEX_ADDR	Address LSB of roi map cu ctrlindex buffer

16.3.5.2.220 roimap cu ctrl index address MSB (swreg274)

16.3.5.2.220.1 Offset

Register	Offset
swreg274	448h

16.3.5.2.220.2 Function

roimap cu ctrl index address MSB

16.3.5.2.220.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROIMAP_CUCTRL_INDEX_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROIMAP_CUCTRL_INDEX_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.220.4 Fields

Field	Function
31-0 SW_ENC_ROIMAP_CUCTRL_INDEX_ADDR_MSB	Address MSB of roi map cu ctrl index buffer

16.3.5.2.221 roimap cu ctrl address LSB (swreg275)

16.3.5.2.221.1 Offset

Register	Offset
swreg275	44Ch

16.3.5.2.221.2 Function

roimap cu ctrl address LSB

16.3.5.2.221.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROIMAP_CUCTRL_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROIMAP_CUCTRL_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.221.4 Fields

Field	Function
31-0 SW_ENC_ROIMAP_CUCTRL_ADDR	Address LSB of roi map cu ctrl buffer

16.3.5.2.222 roimap cu ctrl address MSB (swreg276)

16.3.5.2.222.1 Offset

Register	Offset
swreg276	450h

16.3.5.2.222.2 Function

roimap cu ctrl address MSB

16.3.5.2.222.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_ROIMAP_CUCTRL_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_ROIMAP_CUCTRL_ADDR_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.222.4 Fields

Field	Function
31-0 SW_ENC_ROIMAP_CUCTRL_ADDR_MSB	Address MSB of roi map cu ctrl buffer

16.3.5.2.223 poc type/bits setting (swreg277)

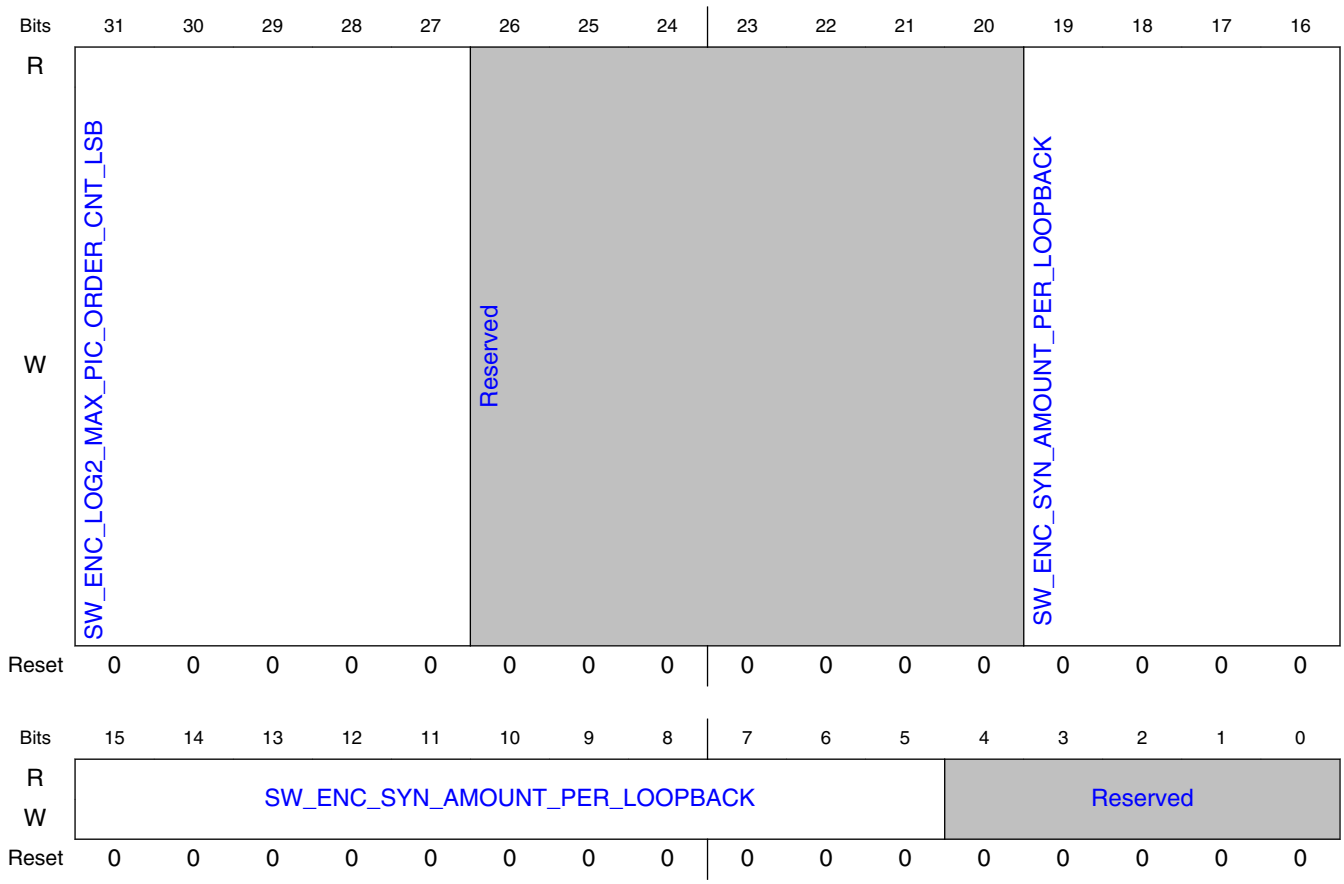
16.3.5.2.223.1 Offset

Register	Offset
swreg277	454h

16.3.5.2.223.2 Function

poc type/bits setting

16.3.5.2.223.3 Diagram



16.3.5.2.223.4 Fields

Field	Function
31-27 SW_ENC_LOG2_MAX_PIC_ORDER_CNT_LSB	number of bits in pic_order_cnt_lsb
26-20 —	Reserved
19-5 SW_ENC_SYN_AMOUNT_PER_LOOPBACK	Handshake sync amount for every loopback
4-0 —	Reserved

16.3.5.2.224 stream output buffer1 address (swreg278)

16.3.5.2.224.1 Offset

Register	Offset
swreg278	458h

16.3.5.2.224.2 Function

stream output buffer1 address

16.3.5.2.224.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_OUTPUT_STRM_BUF1_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_OUTPUT_STRM_BUF1_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.224.4 Fields

Field	Function
31-0 SW_ENC_OUTPUT_STRM_BUF1_BASE	stream output buffer1 base address. Only valid if sw_enc_HWStreamBufChain=1.

16.3.5.2.225 stream output buffer1 limit size (swreg280)

16.3.5.2.225.1 Offset

Register	Offset
swreg280	460h

16.3.5.2.225.2 Function

stream output buffer1 limit size

16.3.5.2.225.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_OUTPUT_STRM_BUFFER1_LIMIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_OUTPUT_STRM_BUFFER1_LIMIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.225.4 Fields

Field	Function
31-0 SW_ENC_OUTPUT_STRM_BUFFER1_LIMIT	Stream buffer1 limit size (bytes). Only valid if sw_enc_HWStreamBufChain=1. If buffer1 limit is reached buffer_full_IRQ will be generated.

16.3.5.2.226 poc type/bits setting (swreg281)

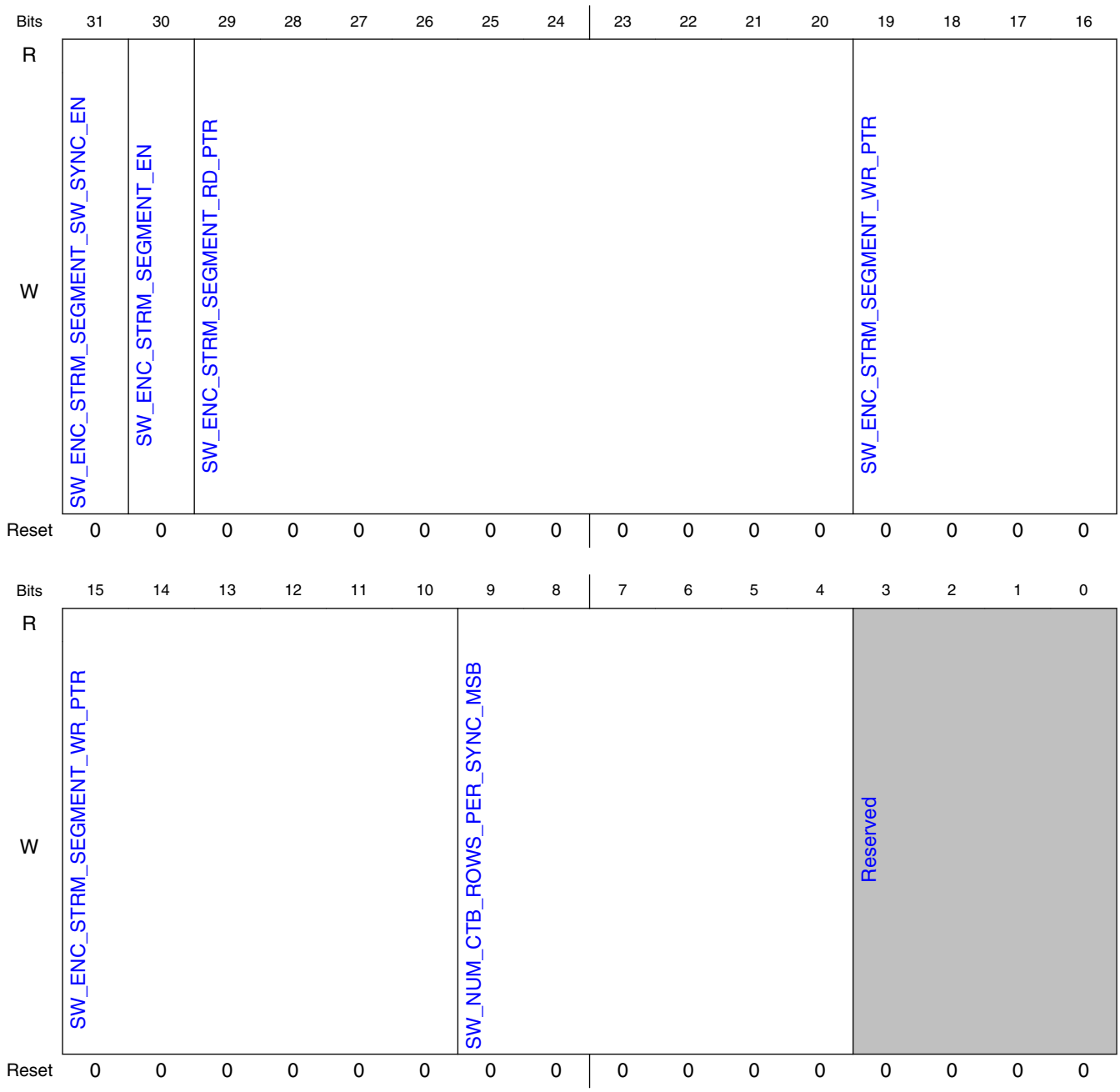
16.3.5.2.226.1 Offset

Register	Offset
swreg281	464h

16.3.5.2.226.2 Function

poc type/bits setting

16.3.5.2.226.3 Diagram



16.3.5.2.226.4 Fields

Field	Function
31 SW_ENC_STRM_SEGMENT_SW_SYNC_EN	Stream segment software handshake enable. Set 0 will disable this handshake

Table continues on the next page...

Field	Function
30 SW_ENC_STRM_SEGMENT_EN	Stream segment function enable
29-20 SW_ENC_STRM_SEGMENT_RD_PTR	the number of segments that have been read out from output buffer
19-10 SW_ENC_STRM_SEGMENT_WR_PTR	the number of segments that have been filled into the output buffer
9-4 SW_NUM_CTB_ROWS_PER_SYNC_MSB	Number MSB of CTB rows for every HW sync. Bit[14:9]. Unit is CTB
3-0 —	Reserved

16.3.5.2.227 HW synthesis config register 4, read-only (swreg287)

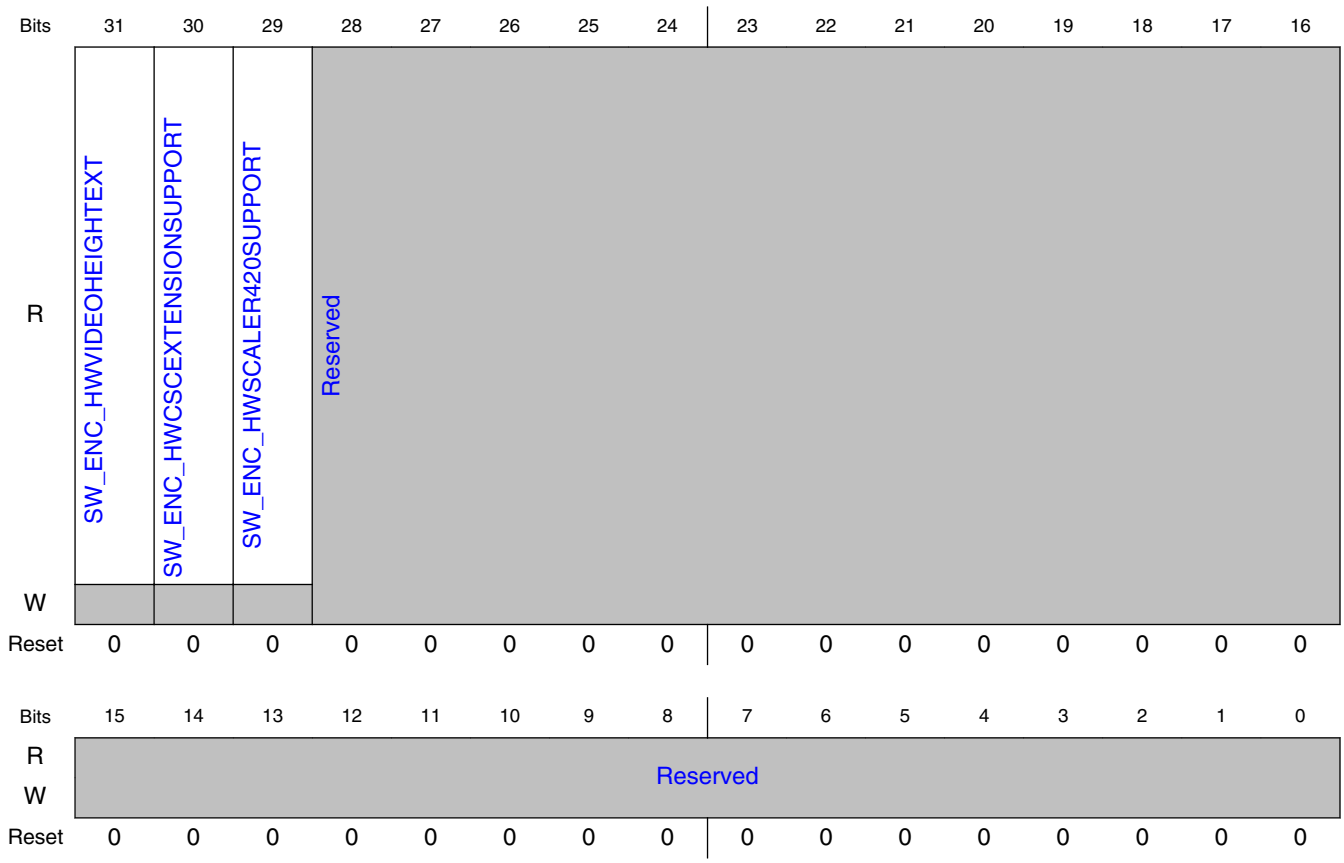
16.3.5.2.227.1 Offset

Register	Offset
swreg287	47Ch

16.3.5.2.227.2 Function

HW synthesis config register 4, read-only

16.3.5.2.227.3 Diagram



16.3.5.2.227.4 Fields

Field	Function
31 SW_ENC_HWVIDEOHEIGHTEXT	Maximum allowed video height extended from 8192 to 8640. 0b - Not. 1b - Yes.
30 SW_ENC_HWCSCEXTENSIONSUPPORT	RGB to YUV conversion extension. 0b - not supported. 1b - supported
29 SW_ENC_HWSCALER420SUPPORT	out-loop scaler output YUV420SP. 0b - not supported. 1b - supported
28-0 —	Reserved

16.3.5.2.228 Pre-processor color conversion parameters1 (swreg289)

16.3.5.2.228.1 Offset

Register	Offset
swreg289	484h

16.3.5.2.228.2 Function

Pre-processor color conversion parameters1

16.3.5.2.228.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ENC_RGBCOEFFG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ENC_RGBCOEFFH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.3.5.2.228.4 Fields

Field	Function
31-16 SW_ENC_RGBCOEFFG	RGB to YUV conversion coefficient G
15-0 SW_ENC_RGBCOEFFH	RGB to YUV conversion coefficient H

16.4 VPU BLK_CTRL

16.4.1 Overview

The Block Control (BLK_CTRL) module contains general purpose registers (GPRs), which control varied features of the associated peripherals.

16.4.2 Memory Map and register definition

This section includes the VPU_BLK_CTRL module memory map and descriptions of all registers.

16.4.2.1 VPU_BLK_CTL register descriptions

16.4.2.1.1 VPU_BLK_CTL memory map

VPU_BLK_CTRL base address: 3833_0000h

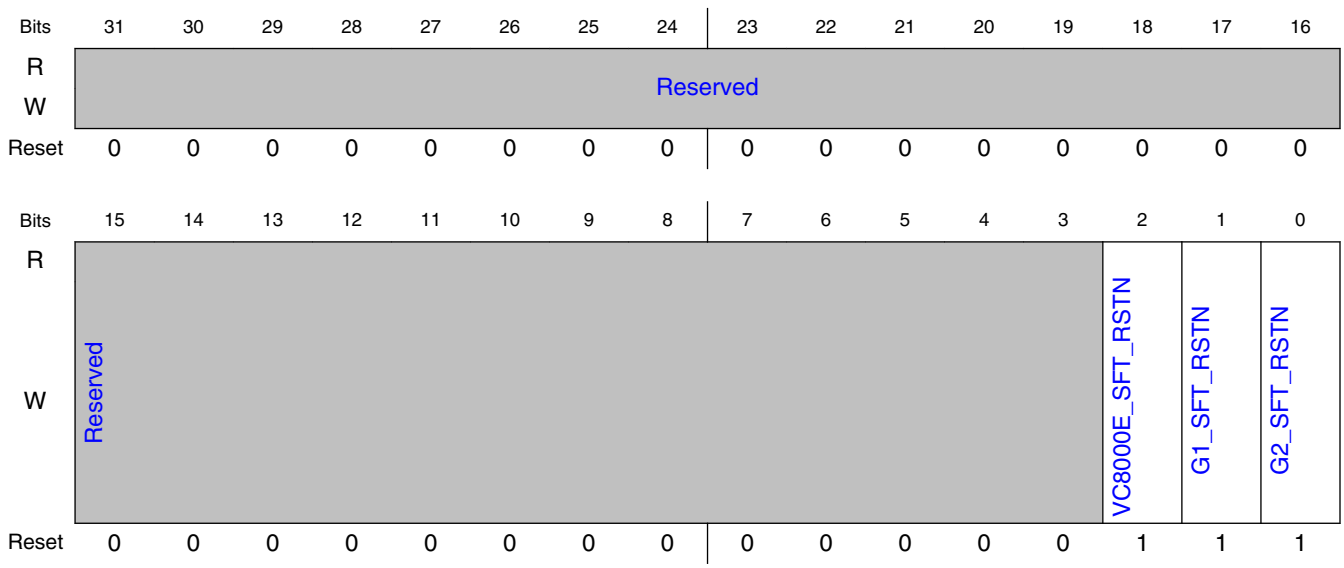
Offset	Register	Width (In bits)	Access	Reset value
0h	VPUMIX block soft reset control (BLK_SFT_RSTN_CSR)	32	RW	0000_0007h
4h	VPUMIX block clock enable control (BLK_CLK_EN_CSR)	32	RW	0000_000Fh
8h	VPUMIX G1 fuse_dec control (G1_FUSE_DEC_CSR)	32	RW	0000_0000h
10h	VPUMIX G2 fuse_dec control (G2_FUSE_DEC_CSR)	32	RW	0000_0000h
14h	VPUMIX VC8000E fuse_enc control (VC8000E_FUSE_ENC_CSR)	32	RW	0000_0000h
18h	VPUMIX block cache enable control (VPU_CACHE_EN_CSR)	32	RW	0000_0000h
1Ch	VPUMIX block pending transaction status (VPU_NO_PENDING_CSR)	32	RO	0000_0000h
20h	VPUMIX G1 outstanding read beat limit control (G1_OTR_BEAT_LIMIT_CSR)	32	RW	0000_0000h
24h	VPUMIX G2 outstanding read beat limit control (G2_OTR_BEAT_LIMIT_CSR)	32	RW	0000_0000h
28h	VPUMIX VC8000E outstanding read beat limit control (VC8000E_OTR_BEAT_LIMIT_CSR)	32	RW	0000_0000h

16.4.2.1.2 VPUMIX block soft reset control (BLK_SFT_RSTN_CSR)

16.4.2.1.2.1 Offset

Register	Offset
BLK_SFT_RSTN_CSR	0h

16.4.2.1.2.2 Diagram



16.4.2.1.2.3 Fields

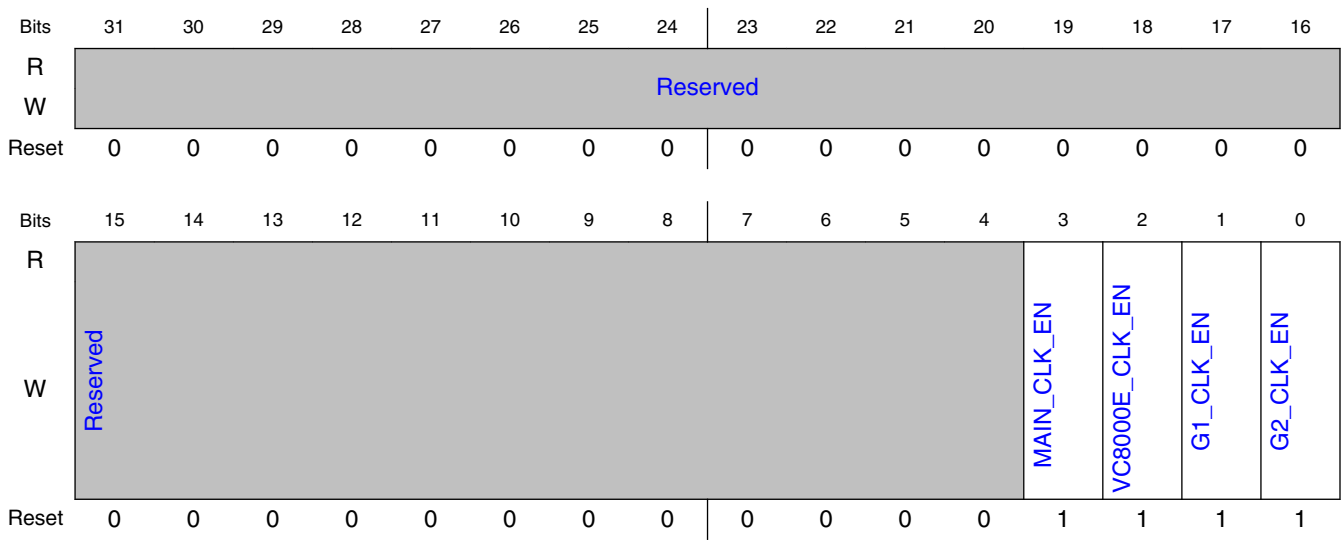
Field	Function
31-3 —	Reserved
2 VC8000E_SFT_RSTN	Soft reset for VC8000E 0b - Reset 1b - Normal
1 G1_SFT_RSTN	Soft reset for G1 0b - Reset 1b - Normal
0 G2_SFT_RSTN	Soft reset for G2 0b - Reset 1b - Normal

16.4.2.1.3 VPUMIX block clock enable control (BLK_CLK_EN_CSR)

16.4.2.1.3.1 Offset

Register	Offset
BLK_CLK_EN_CSR	4h

16.4.2.1.3.2 Diagram



16.4.2.1.3.3 Fields

Field	Function
31-4 —	Reserved
3 MAIN_CLK_EN	Clock enable for noc main bus, active high 0b - Disable 1b - Enable
2 VC8000E_CLK_EN	Clock enable for VC8000E 0b - Disable 1b - Enable
1 G1_CLK_EN	Clock enable for G1 0b - Disable 1b - Enable
0 G2_CLK_EN	Clock enable for G2 0b - Disable 1b - Enable

16.4.2.1.4 VPUMIX G1 fuse_dec control (G1_FUSE_DEC_CSR)

16.4.2.1.4.1 Offset

Register	Offset
G1_FUSE_DEC_CSR	8h

16.4.2.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	G1_FUSE_DEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	G1_FUSE_DEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.4.2.1.4.3 Fields

Field	Function
31-0 G1_FUSE_DEC	G1 fuse decoder enable for each bit 00000000000000000000000000000000b - Disable 00000000000000000000000000000001b - Enable

16.4.2.1.5 VPUMIX G2 fuse_dec control (G2_FUSE_DEC_CSR)

16.4.2.1.5.1 Offset

Register	Offset
G2_FUSE_DEC_CSR	10h

16.4.2.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	G2_FUSE_DEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	G2_FUSE_DEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.4.2.1.5.3 Fields

Field	Function
31-0 G2_FUSE_DEC	G2 fuse decoder enable, active high for each bit 00000000000000000000000000000000b - Disable 11111111111111111111111111111111b - Enable

16.4.2.1.6 VPUMIX VC8000E fuse_enc control (VC8000E_FUSE_ENC_CSR)

16.4.2.1.6.1 Offset

Register	Offset
VC8000E_FUSE_ENC_CSR	14h

16.4.2.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VC8000E_FUSE_ENC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VC8000E_FUSE_ENC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.4.2.1.6.3 Fields

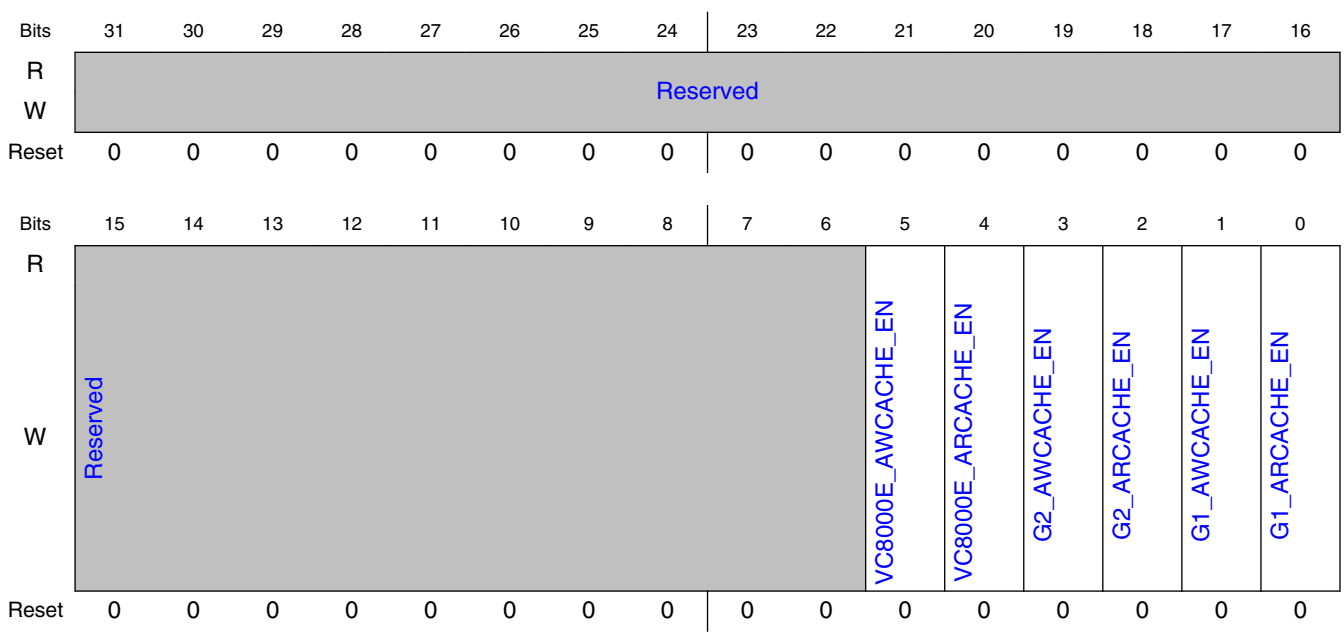
Field	Function
31-0 VC8000E_FUSE_ENC	VC8000E fuse encoder enable, active high for each bit 00000000000000000000000000000000b - Disable 11111111111111111111111111111111b - Enable

16.4.2.1.7 VPUMIX block cache enable control (VPU_CACHE_EN_CSR)

16.4.2.1.7.1 Offset

Register	Offset
VPU_CACHE_EN_CSR	18h

16.4.2.1.7.2 Diagram



16.4.2.1.7.3 Fields

Field	Function
31-6 —	Reserved

Table continues on the next page...

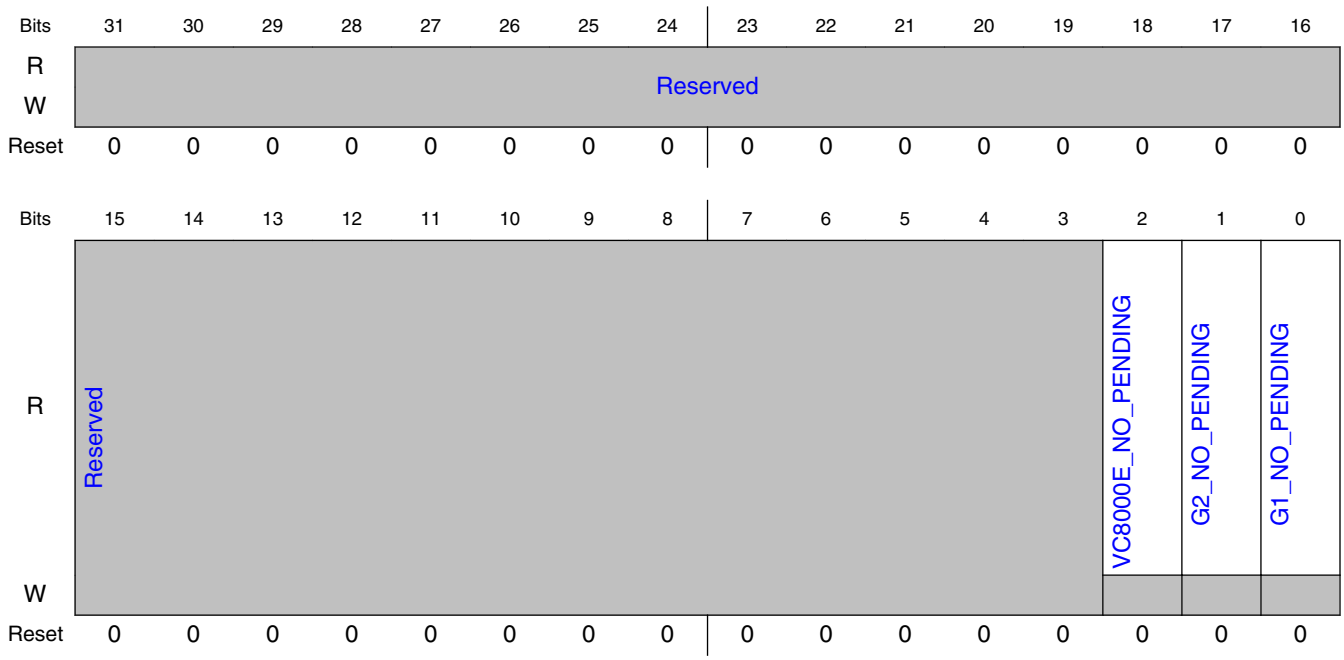
Field	Function
5 VC8000E_AWCACHE_EN	VC8000E AXI AW cacheable enable, active high 0b - Disable 1b - Enable
4 VC8000E_ARCACHE_EN	VC8000E AXI AR cacheable enable, active high 0b - Disable 1b - Enable
3 G2_AWCACHE_EN	G2 AXI AW cacheable enable 0b - Disable 1b - Enable
2 G2_ARCACHE_EN	G2 AXI AR cacheable enable 0b - Disable 1b - Enable
1 G1_AWCACHE_EN	G1 AXI AW cacheable enable 0b - Disable 1b - Enable
0 G1_ARCACHE_EN	G1 AXI AR cacheable enable 0b - Disable 1b - Enable

16.4.2.1.8 VPUMIX block pending transaction status (VPU_NO_PENDING_CSR)

16.4.2.1.8.1 Offset

Register	Offset
VPU_NO_PENDING_CSR	1Ch

16.4.2.1.8.2 Diagram



16.4.2.1.8.3 Fields

Field	Function
31-3 —	Reserved
2 VC8000E_NO_PENDING	Pending transaction status of NOC AXI port connected with VC8000E 0b - Pending transaction 1b - No pending transaction
1 G2_NO_PENDING	Pending transaction status of NOC AXI port connected with G2 0b - Pending transaction 1b - No pending transaction
0 G1_NO_PENDING	Pending transaction status of NOC AXI port connected with G1 0b - Pending transaction 1b - No pending transaction

16.4.2.1.9 VPUMIX G1 outstanding read beat limit control (G1_OTR_BEAT_LIMIT_CSR)

16.4.2.1.9.1 Offset

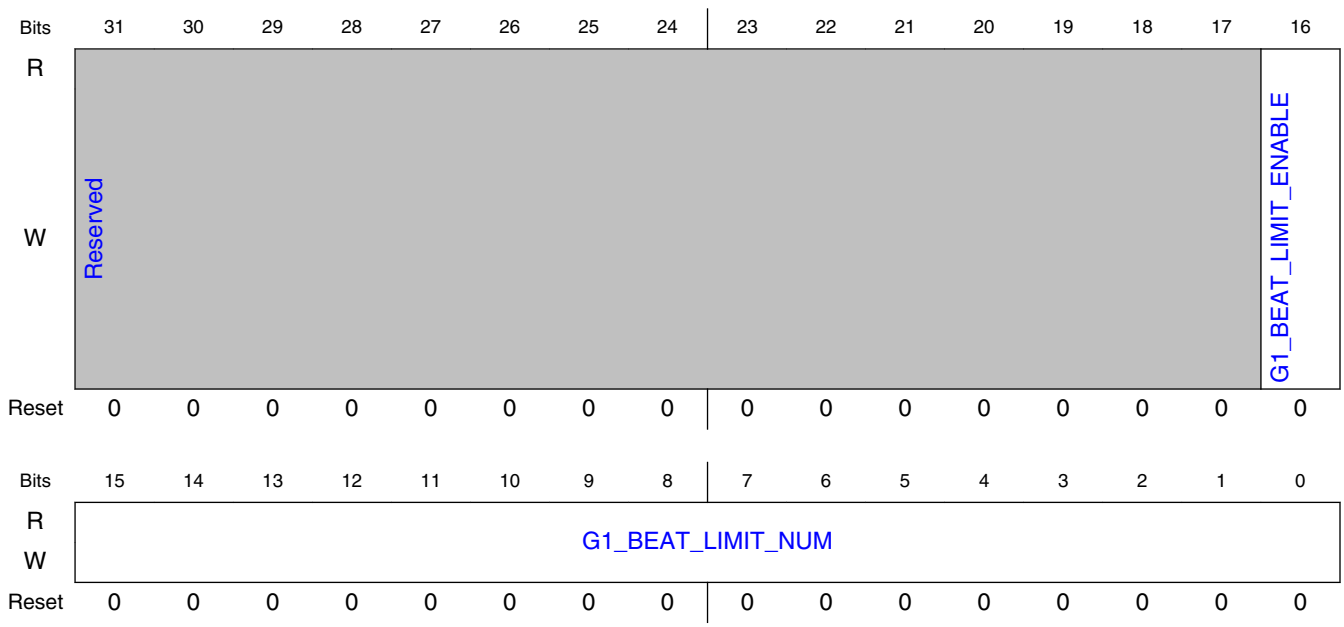
Register	Offset
G1_OTR_BEAT_LIMIT_CSR	20h

16.4.2.1.9.2 Function

NOTE

In normal cases, there is no need to configure and enable this. Needs to be configured/enabled in performance verification. This is used to guarantee DRAM AXI bus will not be blocked with a slow master's read data sink rate.

16.4.2.1.9.3 Diagram



16.4.2.1.9.4 Fields

Field	Function
31-17 —	Reserved
16 G1_BEAT_LIMIT_ENABLE	G1 AXI outstanding read beat limit enable 0b - Disable 1b - Enable

Table continues on the next page...

Field	Function
15-0 G1_BEAT_LIMIT_NUM	G1 AXI outstanding read beat limit number

16.4.2.1.10 VPUMIX G2 outstanding read beat limit control (G2_OTR_BEAT_LIMIT_CSR)

16.4.2.1.10.1 Offset

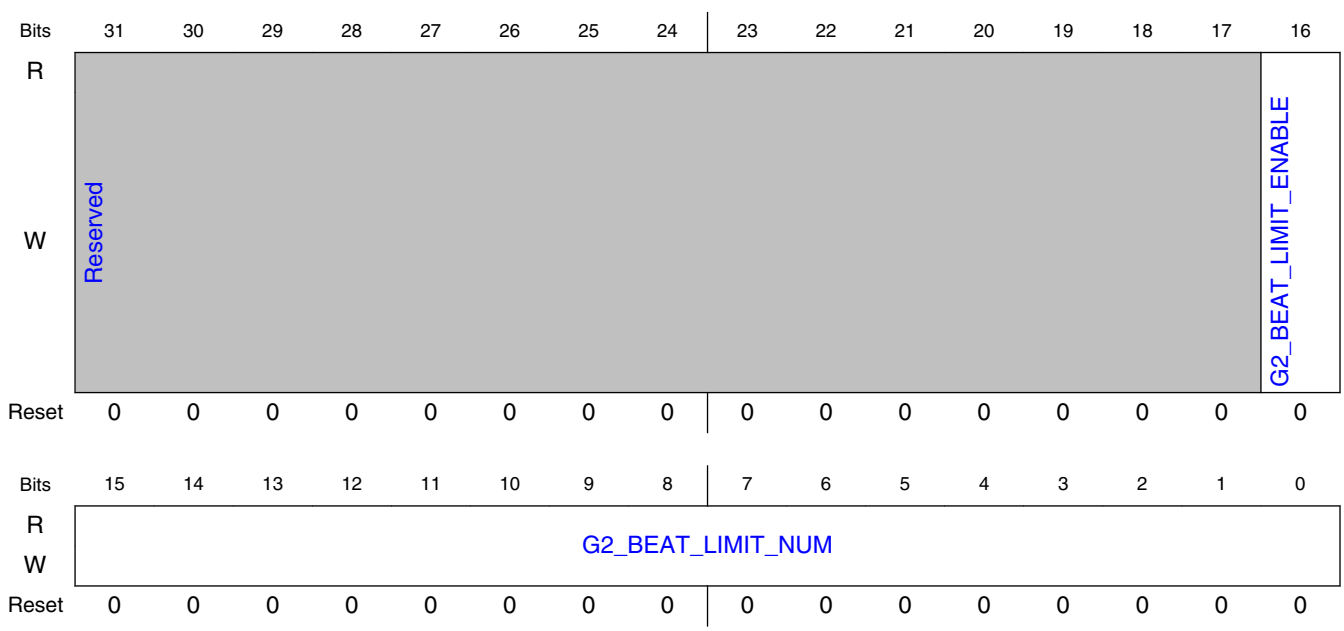
Register	Offset
G2_OTR_BEAT_LIMIT_CSR	24h

16.4.2.1.10.2 Function

NOTE

In normal cases, there is no need to configure and enable this. Needs to be configured/enabled in performance verification. This is used to guarantee DRAM AXI bus will not be blocked with a slow master's read data sink rate.

16.4.2.1.10.3 Diagram



16.4.2.1.10.4 Fields

Field	Function
31-17 —	Reserved
16 G2_BEAT_LIMIT_ENABLE	G2 AXI outstanding read beat limit enable 0b - Disable 1b - Enable
15-0 G2_BEAT_LIMIT_NUM	G2 AXI outstanding read beat limit number

16.4.2.1.11 VPUMIX VC8000E outstanding read beat limit control (VC8000E_OTR_BEAT_LIMIT_CSR)

16.4.2.1.11.1 Offset

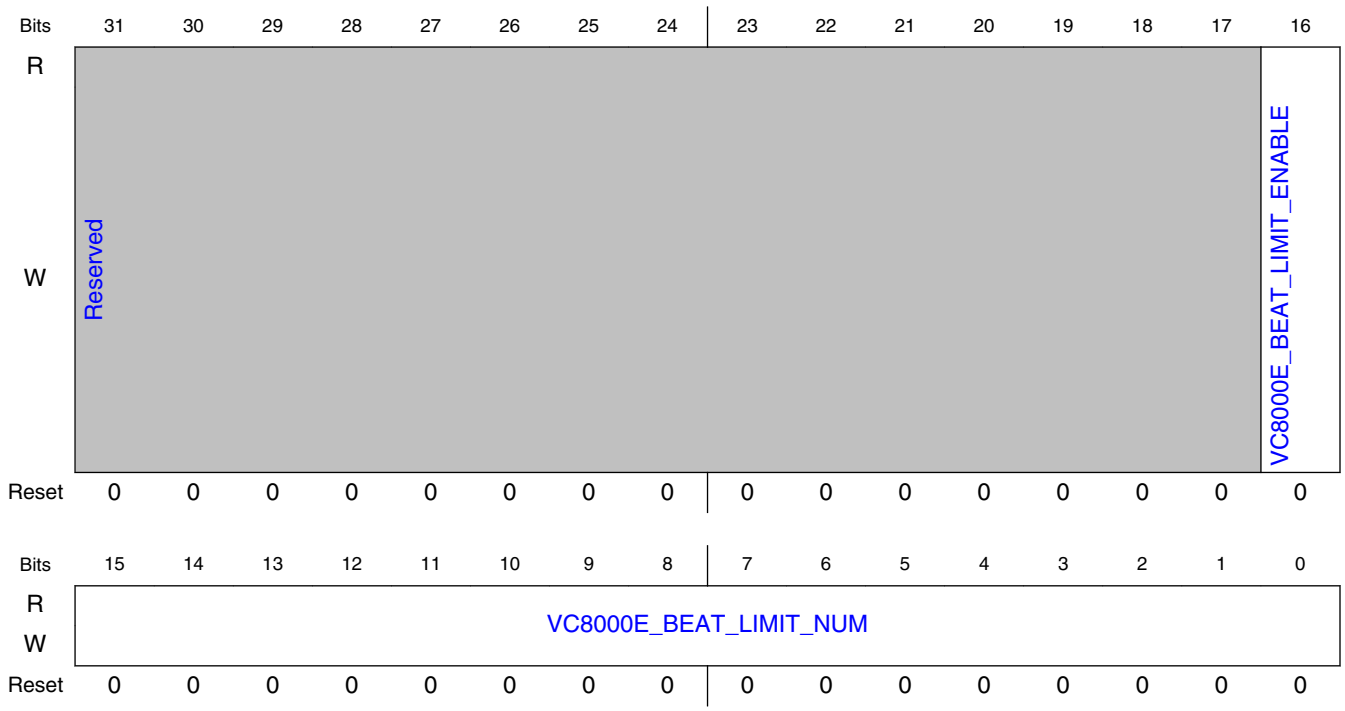
Register	Offset
VC8000E_OTR_BEAT_LIMIT_CSR	28h

16.4.2.1.11.2 Function

NOTE

In normal cases, there is no need to configure and enable this. Needs to be configured/enabled in performance verification. This is used to guarantee DRAM AXI bus will not be blocked with a slow master's read data sink rate.

16.4.2.1.11.3 Diagram



16.4.2.1.11.4 Fields

Field	Function
31-17 —	Reserved
16 VC8000E_BEAT_LIMIT_ENABLE	VC8000E AXI outstanding read beat limit enable 0b - Disable 1b - Enable
15-0 VC8000E_BEAT_LIMIT_NUM	VC8000E AXI outstanding read beat limit number

Chapter 17
Low Speed Communication and Interconnects

17.1 I2C Controller (I2C)

17.1.1 Overview

This chapter describes block-level operation and programming of I2C. The chapter is intended for a block-driver software developer. To understand how the block is integrated at the SoC level, a system software developer should see discussions of the block in the appropriate SoC-level chapter(s).

References: This document assumes an understanding of the following document:

- *The I2C Bus Specification*, Version 2.1, by Philips Semiconductor

The Inter IC (I2C) provides functionality of a standard I2C slave and master. The I2C is designed to be compatible with the standard NXP I2C bus protocol.

NOTE

Four independent I2C channels are available.

I2C is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange, minimizing the interconnection between devices. This bus is suitable for applications requiring occasional communications over a short distance between many devices. The flexible I2C standard allows additional devices to be connected to the bus for expansion and system development. See the connection diagram in the figure below.

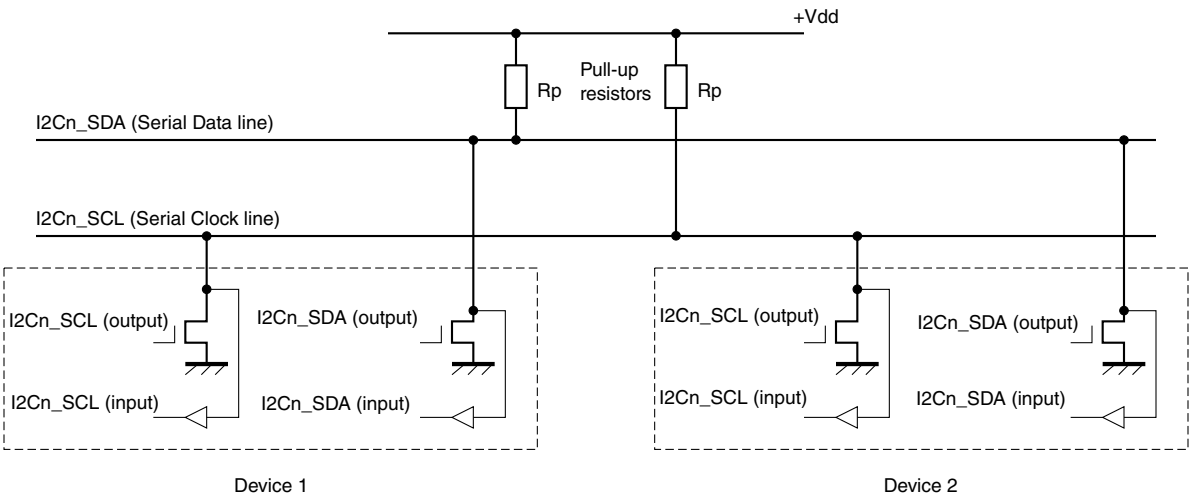


Figure 17-1. Connection of devices to I2C bus

The I2C interface speed is dependent on the I2C bus loading and timing characteristics. For pin requirement details, see *The I2C Bus Specification*.

17.1.1.1 Block Diagram

The I2C system is a true multimaster bus including arbitration and collision detection that prevents data corruption if multiple devices attempt to control the bus simultaneously. This feature supports complex applications with multiprocessor control and can be used for rapid testing and alignment of end products through external connections to an assembly-line computer. The figure below shows the block diagram of I2C.

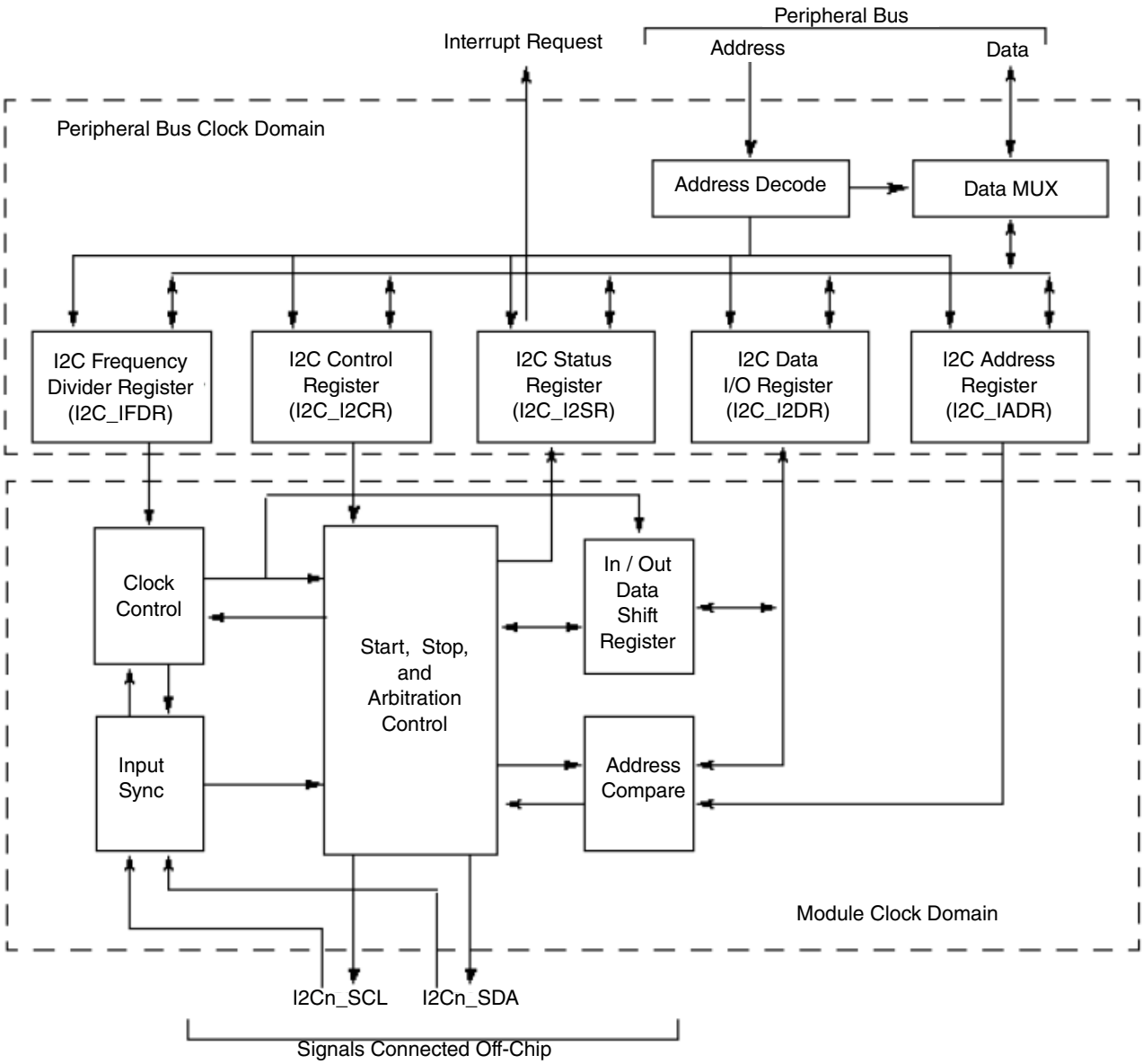


Figure 17-2. I2C block diagram

17.1.1.2 Features

The I2C has the following key features:

- Compatibility with I2C bus standard
- Multimaster operation
- Software programmability for one of 64 different serial clock frequencies
- Software-selectable acknowledge bit
- Interrupt-driven, byte-by-byte data transfer
- Arbitration-lost interrupt with automatic mode switching from master to slave

- Calling address identification interrupt
- Start and stop signal generation/detection
- Repeated Start signal generation
- Acknowledge bit generation/detection
- Bus-busy detection

17.1.2 Functional description

This section provides a complete functional description of the block.

17.1.2.1 Modes and operations

The I2C operates primarily in two functional modes: Standard mode and Fast mode.

- In Standard mode, I2C supports the data transfer rates up to 100 kbits/s.
- In Fast mode, data transfer rates up to 400 kbits/s can be achieved. Per block operation, there is no special configuration required for Fast or Standard mode. It is the data transfer rate that distinguishes Standard and Fast mode.

17.1.2.2 I2C system configuration

After a reset, the I2C defaults to Slave Receive operations. Thus, when not operating as a master or responding to a slave transmit address, the I2C defaults to the Slave Receive state.

For exceptions, see Initialization sequence.

NOTE

The I2C is designed to be compatible with the Philips™ I2C bus protocol. For information on system configuration, protocol, and restrictions, see the *I2C Bus Specification*, version 2.1, by Philips Semiconductors. The I2C supports Standard and Fast modes only.

17.1.2.3 Arbitration procedure

If multiple devices simultaneously request the bus, the bus clock is determined by a synchronization procedure in which the low period equals the longest clock-low period among the devices, and the high period equals the shortest. A data arbitration procedure determines the relative priority of competing devices.

A device loses arbitration if it sends logic high while another sends logic low; it immediately switches to Slave Receive mode and stops driving I2Cn_SDA. In this case, the transition from master to Slave mode does not generate a Stop condition. Meanwhile, hardware sets the arbitration lost bit in the I2C Status register (I2C_I2SR[IAL] to indicate loss of arbitration).

17.1.2.4 Clock synchronization

Because wire-AND logic is used, a high-to-low transition on SCL affects devices connected to the bus. Devices start counting their low period when the master drives SCL low. When a device clock goes low, it holds SCL low until the Clock High state is reached. However, the low-to-high change in this device clock may not change the state of SCL if another device clock is still in its low period. Therefore, the device with the longest low period holds the synchronized clock SCL low.

Devices with shorter low periods enter a High Wait state during this time (see Figure 17-3). When all devices involved have counted off their low periods, the synchronized clock SCL is released and pulled high. There is then no difference between device clocks and the state of SCL, so all of the devices start counting their high periods. The first device to complete its high period pulls SCL low again.

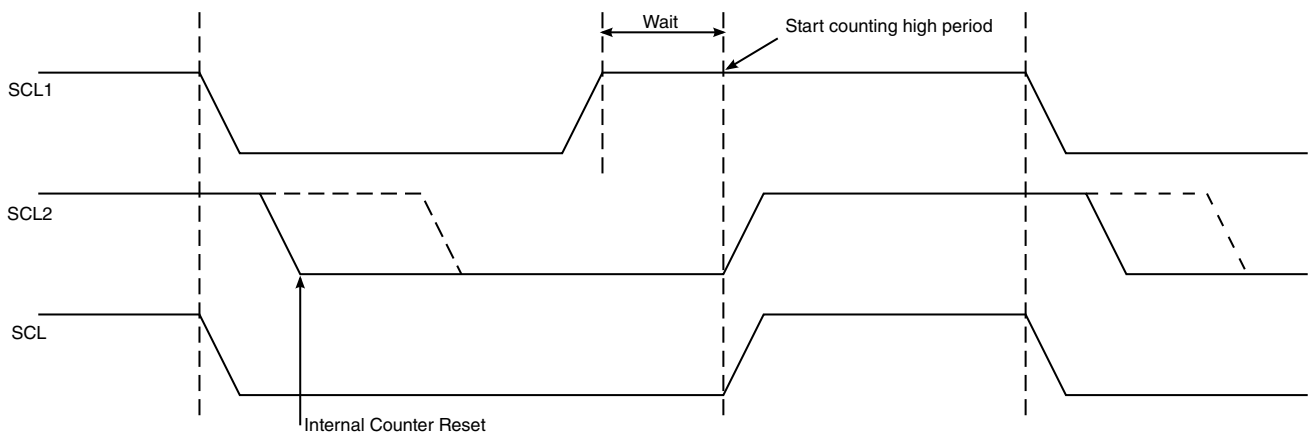


Figure 17-3. Synchronized clock SCL

17.1.2.5 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfers. Slave devices can hold SCL low after completing one byte transfer (9 bits). In such a case, the clock mechanism halts the bus clock and forces the master clock into a Wait state until the slave releases SCL.

17.1.2.6 Clock stretching

Slaves can use the clock synchronization mechanism to slow down the transfer bit rate. After the master has driven SCL low, the slave can drive SCL low for the required period and then release it. If the slave SCL low period is longer than the master SCL low period, the resulting SCL bus signal low period is stretched.

17.1.2.7 Peripheral bus accesses

I2C is a 16-bit block. Only half-word accesses should be performed to the block.

17.1.2.8 Generation of transfer error on IP bus

If an address is received on the peripheral slave bus interface but it is not implemented, an access error is generated.

17.1.2.9 Reset

The I2C can be reset in the following ways:

- Global reset: A hard asynchronous reset of the whole I2C
- Software reset: An internal reset for the whole I2C (except for I2C_IADR and I2C_IFDR registers) initiated by deasserting the I2C_I2CR[IEN] bit

17.1.2.10 Interrupts

There is only one interrupt from the block, which is enabled by setting the I2C_I2CR[IIEN] bit.

The interrupt is generated in any one of the following conditions:

- One byte transfer is completed (the interrupt is set at the falling edge of the ninth clock).
- An address is received that matches its own specific address in Slave Receive mode.
- Arbitration is lost.

17.1.2.11 Byte order

The block only supports the Little-Endian mode.

17.1.3 External Signals

This section discusses I2C signals that connect off-chip.

For I2C compliance, all devices connected to the I2Cn_SCL and I2Cn_SDA signals must have open-drain or open-collector outputs. The logic AND function is implemented on both lines with external pull-up resistors.

Inputs of I2Cn_SCL and I2Cn_SDA also need to be manually enabled by setting the SION bit in the IOMUX after the corresponding PADs are selected as I2C function.

The table below describes all I2C signals that connect off-chip.

17.1.4 Initialization

NOTE

Ensure the input select pins for IOMUXC are configured correctly for I2C.

17.1.4.1 Initialization sequence

Before the interface can transfer serial data, registers must be initialized, as listed here.

1. Set the data sampling rate (I2C_IFDR[IC] to obtain SCL frequency from the system bus clock.
2. Update the address in the (I2C_IADR) to define its slave address (address can range from 0 to 0x7f).
3. Set the I2C enable bit (I2C_I2CR[IEN]) to enable the I2C bus interface system.
4. Modify the bits in the I2C_I2CR to select Master/Slave mode, Transmit/Receive mode, and Interrupt-Enable or not.

17.1.4.2 Generation of Start

After completion of the initialization procedure, serial data can be transmitted by selecting the Master Transmit mode. On a multimaster bus system, the busy bus (I2C_I2SR[IBB]) must be tested to determine whether the serial bus is free. If the bus is

free (IBB = 0), the Start signal and the first byte (the slave address) can be sent. The data written to the data register comprises the address of the desired slave and the LSB indicates the transfer direction.

The free time between a Stop and the next Start condition is built into the hardware that generates the Start cycle. Depending on the relative frequencies of the system clock and the SCL period, it may be necessary to wait until the I2C is not busy after writing the calling address to the data register (I2C_I2DR), before proceeding to load data into the data register (I2C_I2DR).

17.1.4.3 Post-transfer software response

Sending or receiving a byte sets the data transferring bit (I2C_I2SR[ICF]), which indicates one byte of communication is finished. Upon completion, the interrupt status (I2C_I2SR[IIF]) is also set. An external interrupt is generated if the interrupt enable (I2C_I2CR[IIEN]) is set. The software must first clear the interrupt status (I2C_I2SR[IIF]) in the interrupt routine.

See the flow chart in Figure 17-5.

The data transferring bit (I2C_I2SR[ICF]) is cleared either by reading from I2C_I2DR in Receive mode or by writing to this register in Transmit mode.

The software can service the I2C I/O in the main program by monitoring the interrupt status (I2C_I2SR[IIF]) if the interrupt enable is deasserted. In this case, the interrupt status should be polled in the data transferring bit (I2C_I2SR[ICF]) because the operation is different when arbitration is lost.

When an interrupt occurs at the end of the address cycle, the master is always in Transmit mode; that is, the address is sent. If Master Receive mode is required, then I2C_I2CR[MTX] should be toggled and a dummy read of the I2C_I2DR register must be executed to trigger receive data.

During Slave-mode address cycles (I2C_I2SR[IAAS] = 1), the slave read/write bit I2C_I2SR[SRW] is read to determine the direction of the next transfer. The transmit/receive bit (I2C_I2CR[MTX]) should also be programmed accordingly. For Slave-mode data cycles (IAAS = 0), SRW is invalid. MTX should be read to determine the current transfer direction.

17.1.4.4 Generation of Stop

A data transfer ends when the master signals a Stop, which can occur after all data is sent.

For a master receiver to terminate a data transfer, it must inform the slave transmitter by not acknowledging the last data byte. This is done by setting the transmit acknowledge bit (I2C_I2CR[TXAK]) before reading the next-to-last byte. Before the last byte is read, a Stop signal must be generated.

17.1.4.5 Generation of Repeated Start

After the data transfer, if the master still requires the bus, it can signal another Start followed by another slave address without signaling a Stop.

17.1.4.6 Slave mode

In the slave interrupt service routine (see Figure 17-5), the block addressed as slave bit (IAAS) should be tested to check if a calling of its own address has just been received. If IAAS is set, software should set the Transmit/Receive mode select bit (I2C_I2CR[MTX]) according to the I2C_I2SR[SRW]. Writing to the I2C_I2CR clears the IAAS automatically. The only time IAAS is read as set is from the interrupt at the end of the address cycle where an address match occurred; interrupts resulting from subsequent data transfers will have IAAS cleared. A data transfer can now be initiated by writing information to I2C_I2DR for slave transmits, or read from I2C_I2DR in Slave Receive mode. A dummy read of I2C_I2DR in Slave Receive mode releases SCL, allowing the master to send data.

In the slave transmitter routine, the receive acknowledge bit (I2C_I2SR[RXAK]) must be tested before sending the next byte of data. Setting RXAK means an end-of-data signal from the master receiver, after which the software must switch it from Transmit to Receiver mode. Reading the data register (I2C_I2DR) then releases SCL so the master can generate a Stop signal.

17.1.4.7 Arbitration lost

If several devices try to engage the bus at the same time, one becomes master. Hardware immediately switches devices that lose arbitration to Slave Receive mode. Data output to 12Cn_SDA stops, but 12Cn_SCL is still generated until the end of the byte during which arbitration is lost. An interrupt occurs at the falling edge of the ninth clock of this transfer if the arbitration is lost (I2C_I2SR[IAL] = 1), and the Slave mode is selected (I2C_I2CR[MSTA] = 0).

See the flow chart in Figure 17-5.

If a device that is not a master tries to transmit or do a Start, hardware inhibits the transmission, clears MSTA without signaling a Stop, generates an interrupt to the Arm platform, and sets I2C_I2SR[IAL] to indicate a failed attempt to engage the bus. When considering these cases, the slave service routine should first test I2C_I2SR[IAL], and the software should clear it if it is set.

For Multimaster mode, when an I2C is enabled when the bus is busy and asserts Start, the I2C_I2SR[IAL] bit gets set only for 12Cn_SDA=0, 12Cn_SCL=0/1, 12Cn_SDA=1, and 12Cn_SCL=0; but not for 12Cn_SDA=1 and I2Cn_SCL=1, which is the equivalent of Bus Idle state.

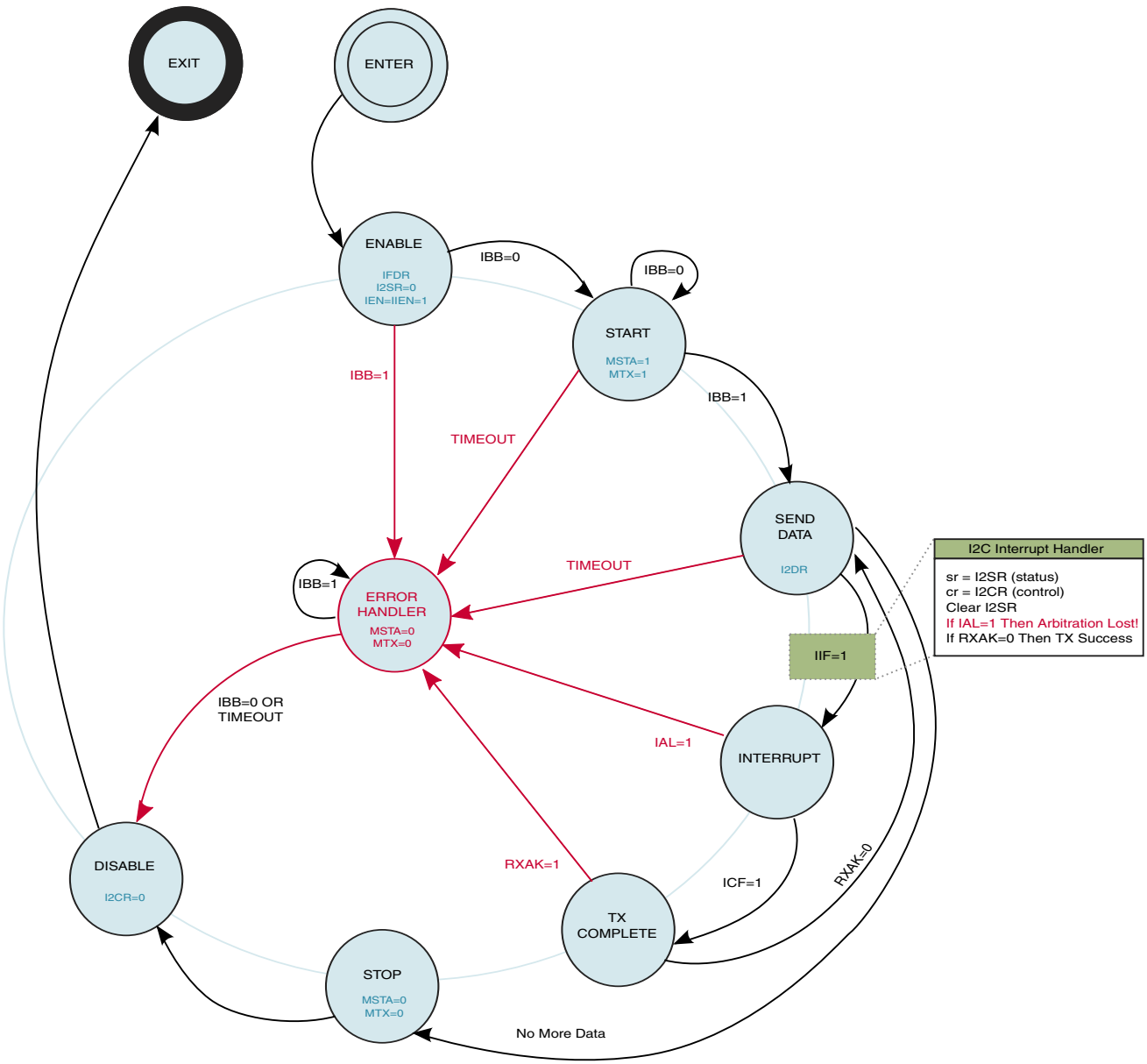


Figure 17-4. I2C Programming state diagram

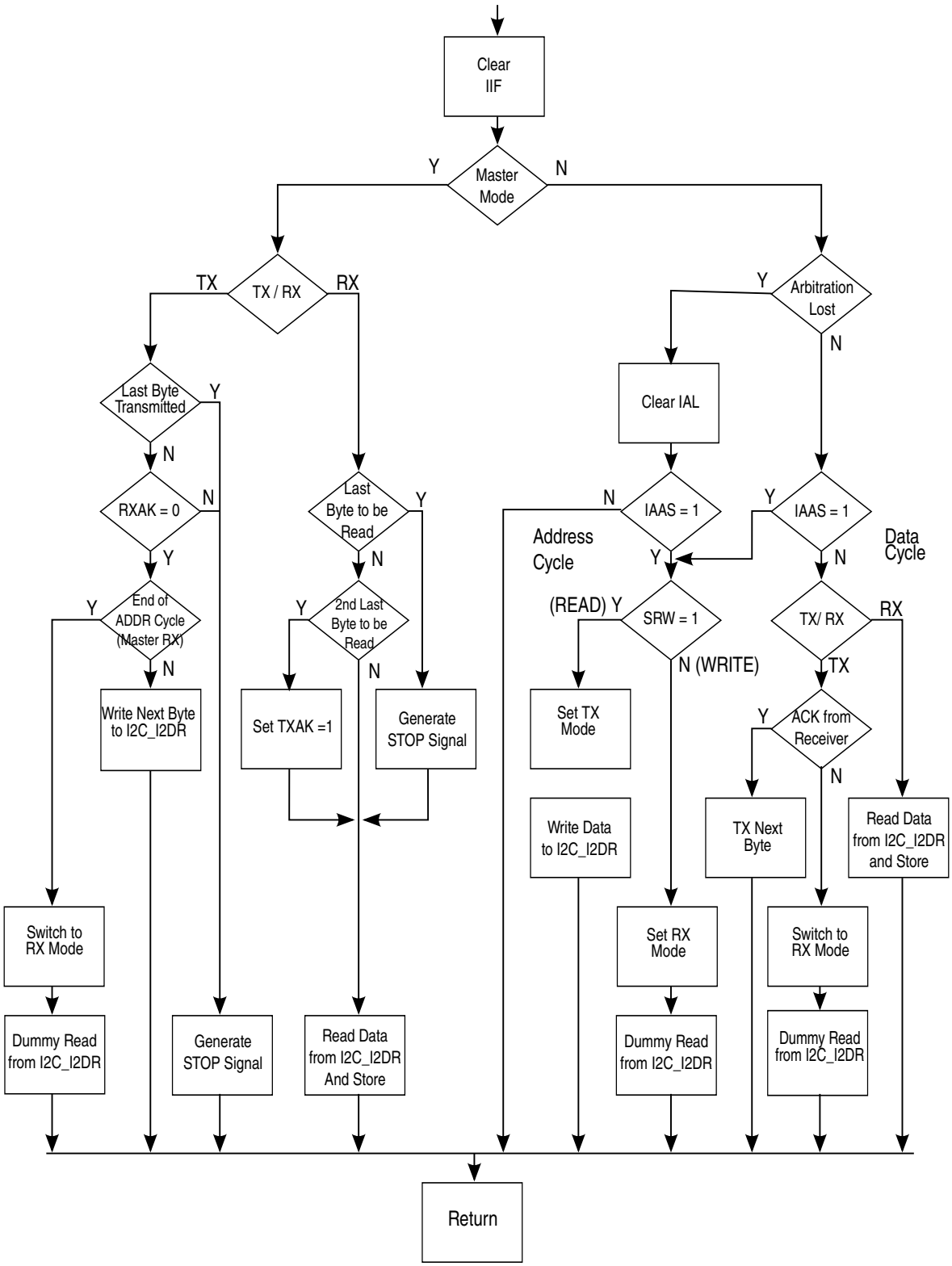


Figure 17-5. Flowchart of typical I2C interrupt routine

NOTE

For a Repeated Start only, the Stop-generation stage does not occur in Master mode. A loop repeats itself without stopping for the next start.

For Master Receive mode, I2C is programmed as Master Transmit during Address mode and after slave address transfer; the MTX bit should be cleared and a dummy read on the I2C_I2DR register should be performed so I2C can read the next receive data.

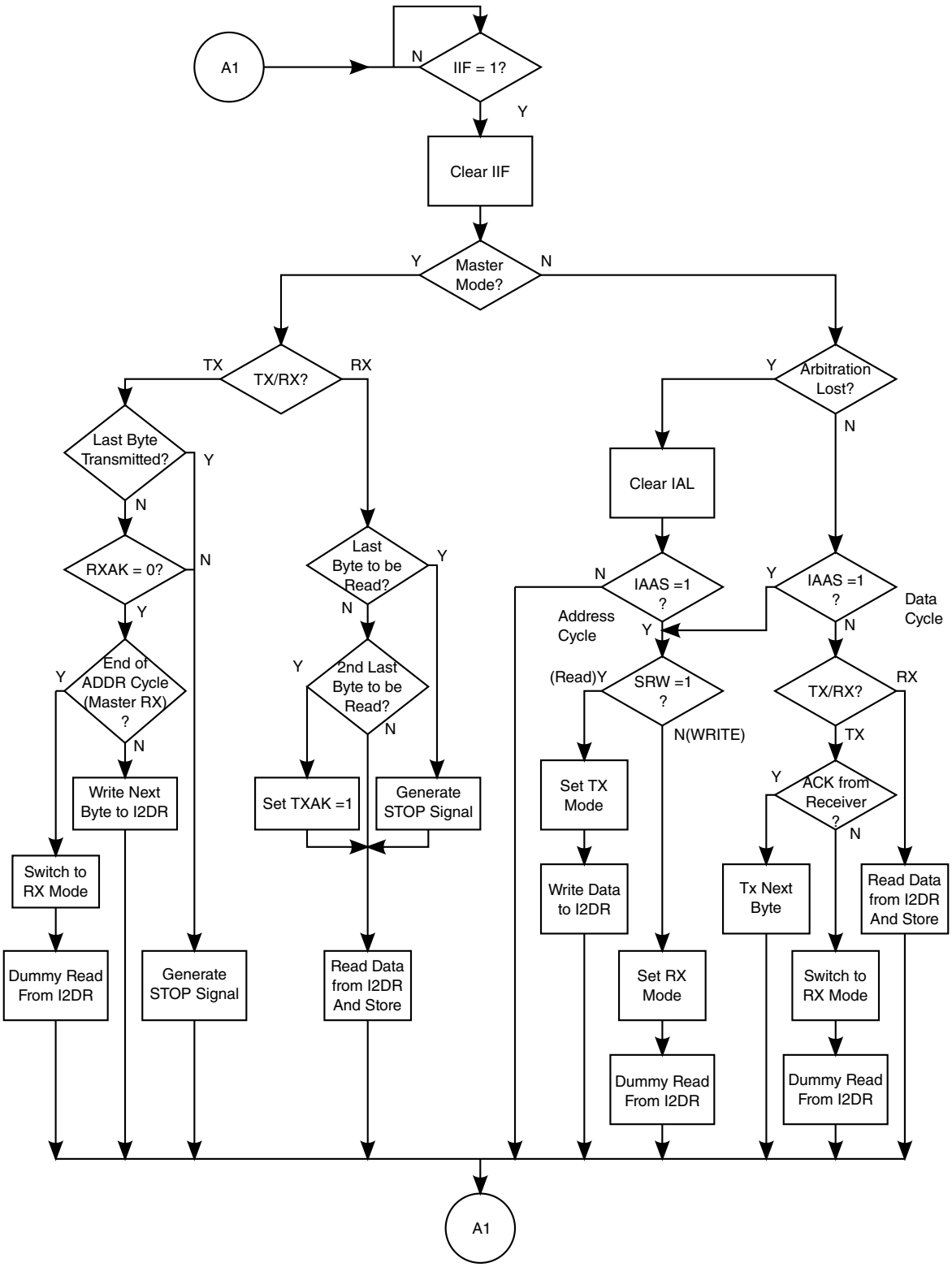


Figure 17-6. Flowchart for typical I2C polling routine

NOTE

The timeout value depends on the bus frequency at which I2C is operating. The minimum timeout for polling the IIF bit at a maximum I2C bus frequency of 400 kHz is T_{min} = 25 µs (=2.5 x 10 µs). This value can be calculated for any bus frequency. The formula is $T_{min} = 10/F_{SCL}$, where F_{SCL} is the frequency of the I2C clock (SCL).

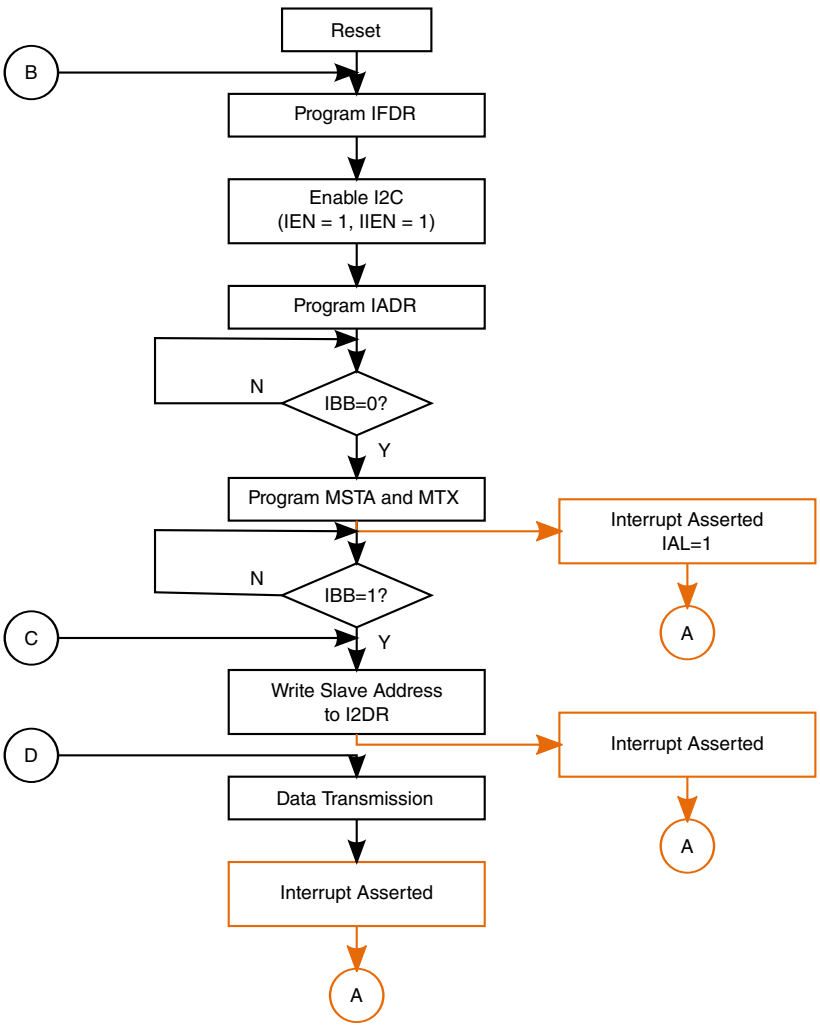


Figure 17-7. Detailed flowchart of a typical I2C Master Transmit mode, part 1

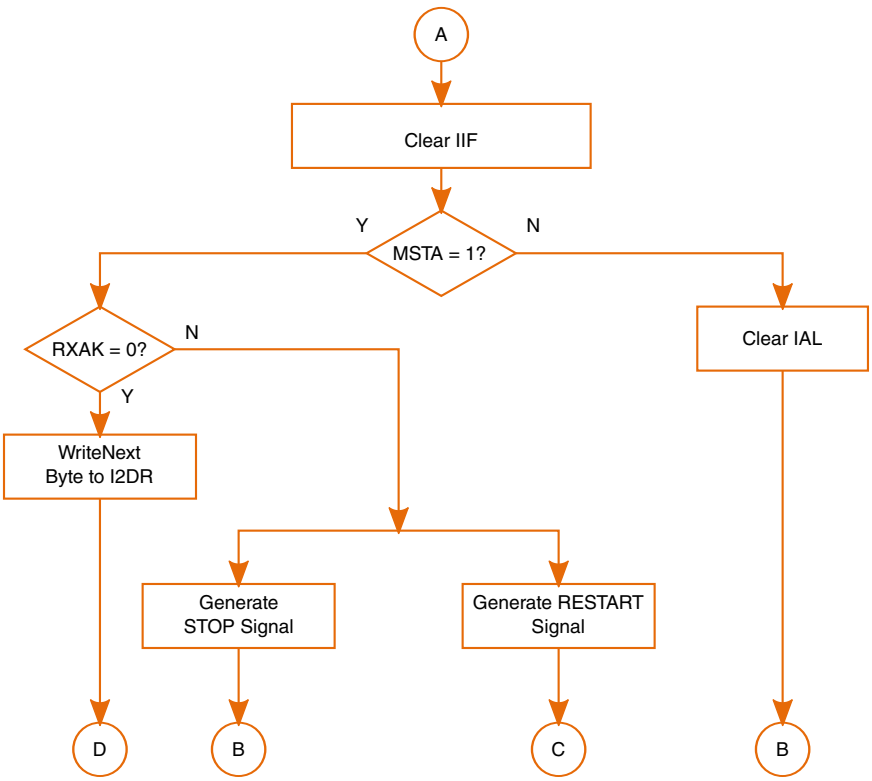


Figure 17-8. Detailed flowchart of a typical I2C Master Transmit mode, part 2

Figure 17-7 and Figure 17-8 show the Master Transmit mode operation with interrupt subroutine. If an interrupt is generated and the MSTA bit is 0, then bus arbitration is lost and IAL is set. Software can clear the IAL bit and reprogram I2C. If the MSTA bit is 1, then it is a transfer-generated interrupt. In this case, software can check the RXAK bit for a data receive acknowledgement by the slave and, accordingly, decide to do one of the following:

- Generate a STOP
- Generate a REPEATED START by writing to the I2C_I2CR register
- Perform the next data transfer by writing to the I2C_I2DR register

NOTE

The IBB bit is asserted by a Start condition on the bus, and it is deasserted by a Stop condition on the bus. Therefore, if arbitration is lost due to an unexpected Stop condition during transfer, then IBB is cleared. If arbitration is lost due to a data mismatch, then it is not cleared. Software should always clear the IEN bit and then set it if arbitration is lost.

17.1.5 Software restriction

Software should ensure that there is a delay of at least two module clock cycles after it sets the I2C_I2CR[RSTA] bit and before writing to the I2C_I2DR register. The maximum possible clock period of the module clock is 78 ns.

17.1.6 I2C Memory Map/Register Definition

The I2C contains five 16-bit registers.

NOTE

Registers at offsets 0x0002, 0x0006, 0x000A, and 0x000E are reserved for future additions.

I2C memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30A2_0000	I2C Address Register (I2C1_IADR)	16	R/W	0000h	17.1.6.1/ 7361
30A2_0004	I2C Frequency Divider Register (I2C1_IFDR)	16	R/W	0000h	17.1.6.2/ 7362
30A2_0008	I2C Control Register (I2C1_I2CR)	16	R/W	0000h	17.1.6.3/ 7363
30A2_000C	I2C Status Register (I2C1_I2SR)	16	R/W	0081h	17.1.6.4/ 7365
30A2_0010	I2C Data I/O Register (I2C1_I2DR)	16	R/W	0000h	17.1.6.5/ 7366
30A3_0000	I2C Address Register (I2C2_IADR)	16	R/W	0000h	17.1.6.1/ 7361
30A3_0004	I2C Frequency Divider Register (I2C2_IFDR)	16	R/W	0000h	17.1.6.2/ 7362
30A3_0008	I2C Control Register (I2C2_I2CR)	16	R/W	0000h	17.1.6.3/ 7363
30A3_000C	I2C Status Register (I2C2_I2SR)	16	R/W	0081h	17.1.6.4/ 7365
30A3_0010	I2C Data I/O Register (I2C2_I2DR)	16	R/W	0000h	17.1.6.5/ 7366
30A4_0000	I2C Address Register (I2C3_IADR)	16	R/W	0000h	17.1.6.1/ 7361
30A4_0004	I2C Frequency Divider Register (I2C3_IFDR)	16	R/W	0000h	17.1.6.2/ 7362

Table continues on the next page...

I2C memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30A4_0008	I2C Control Register (I2C3_I2CR)	16	R/W	0000h	17.1.6.3/ 7363
30A4_000C	I2C Status Register (I2C3_I2SR)	16	R/W	0081h	17.1.6.4/ 7365
30A4_0010	I2C Data I/O Register (I2C3_I2DR)	16	R/W	0000h	17.1.6.5/ 7366
30A5_0000	I2C Address Register (I2C4_IADR)	16	R/W	0000h	17.1.6.1/ 7361
30A5_0004	I2C Frequency Divider Register (I2C4_IFDR)	16	R/W	0000h	17.1.6.2/ 7362
30A5_0008	I2C Control Register (I2C4_I2CR)	16	R/W	0000h	17.1.6.3/ 7363
30A5_000C	I2C Status Register (I2C4_I2SR)	16	R/W	0081h	17.1.6.4/ 7365
30A5_0010	I2C Data I/O Register (I2C4_I2DR)	16	R/W	0000h	17.1.6.5/ 7366
30AD_0000	I2C Address Register (I2C5_IADR)	16	R/W	0000h	17.1.6.1/ 7361
30AD_0004	I2C Frequency Divider Register (I2C5_IFDR)	16	R/W	0000h	17.1.6.2/ 7362
30AD_0008	I2C Control Register (I2C5_I2CR)	16	R/W	0000h	17.1.6.3/ 7363
30AD_000C	I2C Status Register (I2C5_I2SR)	16	R/W	0081h	17.1.6.4/ 7365
30AD_0010	I2C Data I/O Register (I2C5_I2DR)	16	R/W	0000h	17.1.6.5/ 7366
30AE_0000	I2C Address Register (I2C6_IADR)	16	R/W	0000h	17.1.6.1/ 7361
30AE_0004	I2C Frequency Divider Register (I2C6_IFDR)	16	R/W	0000h	17.1.6.2/ 7362
30AE_0008	I2C Control Register (I2C6_I2CR)	16	R/W	0000h	17.1.6.3/ 7363
30AE_000C	I2C Status Register (I2C6_I2SR)	16	R/W	0081h	17.1.6.4/ 7365
30AE_0010	I2C Data I/O Register (I2C6_I2DR)	16	R/W	0000h	17.1.6.5/ 7366

17.1.6.1 I2C Address Register (I2Cx_IADR)

Address: Base address + 0h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								ADR							0
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Cx_IADR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value 0.
7–1 ADR	Slave address. Contains the specific slave address to be used by the I2C. Slave mode is the default I2C mode for an address match on the bus. **NOTE:** The I2C_IADR holds the address to which the I2C responds when addressed as a slave. The slave address is not the address sent on the bus during the address transfer. The register is not reset by a software reset.
0 Reserved	This read-only field is reserved and always has the value 0.

17.1.6.2 I2C Frequency Divider Register (I2Cx_IFDR)

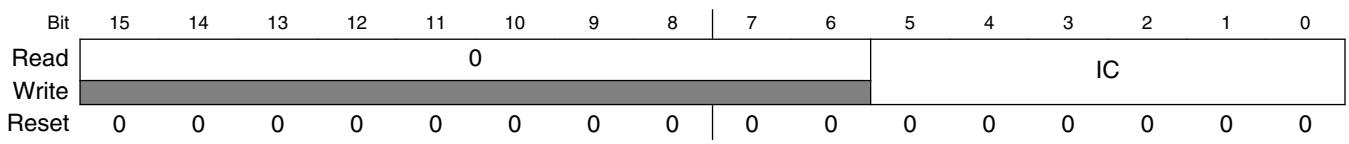
The I2C_IFDR provides a programmable prescaler to configure the clock for bit-rate selection. The register does not get reset by a software reset.

The following table describes the divider and register values for the register field "IC."

Table 17-1. I2C_IFDR Register Field Values

IC	Divider		IC	Divider		IC	Divider		IC	Divider
0x00	30		0x10	288		0x20	22		0x30	160
0x01	32		0x11	320		0x21	24		0x31	192
0x02	36		0x12	384		0x22	26		0x32	224
0x03	42		0x13	480		0x23	28		0x33	256
0x04	48		0x14	576		0x24	32		0x34	320
0x05	52		0x15	640		0x25	36		0x35	384
0x06	60		0x16	768		0x26	40		0x36	448
0x07	72		0x17	960		0x27	44		0x37	512
0x08	80		0x18	1152		0x28	48		0x38	640
0x09	88		0x19	1280		0x29	56		0x39	768
0x0A	104		0x1A	1536		0x2A	64		0x3A	896
0x0B	128		0x1B	1920		0x2B	72		0x3B	1024
0x0C	144		0x1C	2304		0x2C	80		0x3C	1280
0x0D	160		0x1D	2560		0x2D	96		0x3D	1536
0x0E	192		0x1E	3072		0x2E	112		0x3E	1792
0x0F	240		0x1F	3840		0x2F	128		0x3F	2048

Address: Base address + 4h offset



I2Cx_IFDR field descriptions

Field	Description
15–6 Reserved	This read-only field is reserved and always has the value 0.
IC	I2C clock rate. Prescales the clock for bit-rate selection. Due to potentially slow I2Cn_SCL and I2Cn_SDA rise and fall times, bus signals are sampled at the prescaler frequency. The serial bit clock frequency may be lower than IPG_CLK_ROOT divided by the divider shown in the I2C Data I/O Register. **NOTE:** The IC value should not be changed during the data transfer, however, it can be changed before a Repeat Start or Start programming sequence in I2C. The I2C protocol supports bit rates of up to 400 kbps. The IC bits need to be programmed in accordance with this constraint.

17.1.6.3 I2C Control Register (I2Cx_I2CR)

The I2C_I2CR is used to enable the I2C and the I2C interrupt. It also contains bits that govern operation as a slave or a master.

Address: Base address + 8h offset

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	IEN	IIEN	MSTA	MTX	TXAK	0	0	
Write						RSTA		
Reset	0	0	0	0	0	0	0	0

I2Cx_I2CR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value 0.
7 IEN	I2C enable. Also controls the software reset of the entire I2C. Resetting the bit generates an internal reset to the block. If the block is enabled in the middle of a byte transfer, Slave mode ignores the current bus transfer and starts operating when the next Start condition is detected. Master mode is not aware that the bus is busy, so initiating a start cycle may corrupt the current bus cycle, ultimately causing either the current master or the I2C to lose arbitration. Subsequently, bus operation returns to normal. 0 The block is disabled, but registers can still be accessed. 1 The I2C is enabled. This bit must be set before any other I2C_I2CR bits have an effect.
6 IIEN	I2C interrupt enable. **NOTE:** If data is written during the Start condition, that is, just after setting the I2C_I2CR[MSTA] and I2C_I2CR[MTX] bits, then the ICF bit is cleared at the falling edge of SCLK after Start. If data is

Table continues on the next page...

I2Cx_I2CR field descriptions (continued)

Field	Description
	written after the Start condition and falling edge of SCLK, then the ICF bit is cleared as soon as data is written. 0 I2C interrupts are disabled, but the status flag I2C_I2SR[IIF] continues to be set when an Interrupt condition occurs. 1 I2C interrupts are enabled. An I2C interrupt occurs if I2C_I2SR[IIF] is also set.
5 MSTA	Master/Slave mode select bit. If the master loses arbitration, MSTA is cleared without generating a Stop signal. **NOTE:** The module clock should be on for writing to the MSTA bit. **NOTE:** The MSTA bit is cleared by software to generate a Stop condition; it can also be cleared by hardware when the I2C loses the bus arbitration. 0 Slave mode. Changing MSTA from 1 to 0 generates a Stop and selects Slave mode. 1 Master mode. Changing MSTA from 0 to 1 signals a Start on the bus and selects Master mode.
4 MTX	Transmit/Receive mode select bit. Selects the direction of master and slave transfers. 0 Receive. When a slave is addressed, the software should set MTX according to the slave read/write bit in the I2C status register (I2C_I2SR[SRW]). 1 Transmit. In Master mode, MTX should be set according to the type of transfer required. Therefore, for address cycles, MTX is always 1.
3 TXAK	Transmit acknowledge enable. Specifies the value driven onto I2Cn_SDA during acknowledge cycles for both master and slave receivers. **NOTE:** Writing TXAK applies only when the I2C bus is a receiver. 0 An acknowledge signal is sent to the bus at the ninth clock bit after receiving one byte of data. 1 No acknowledge signal response is sent (that is, the acknowledge bit = 1).
2 RSTA	Repeat start. Always reads as 0. Attempting a repeat start without bus mastership causes loss of arbitration. 0 No repeat start 1 Generates a Repeated Start condition
Reserved	This read-only field is reserved and always has the value 0.

17.1.6.4 I2C Status Register (I2Cx_I2SR)

The I2C_I2SR contains bits that indicate transaction direction and status.

Address: Base address + Ch offset

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	ICF	IAAS	IBB	IAL	0	SRW	IIF	RXAK
Write								
Reset	1	0	0	0	0	0	0	1

I2Cx_I2SR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value 0.
7 ICF	Data transferring bit. While one byte of data is transferred, ICF is cleared. 0 Transfer is in progress. 1 Transfer is complete. This bit is set by the falling edge of the ninth clock of the last byte transfer.
6 IAAS	I2C addressed as a slave bit. The Arm platform is interrupted if the interrupt enable (I2C_I2CR[IIEN]) is set. The Arm platform must check the slave read/write bit (SRW) and set its Transfer/Receive mode accordingly. Writing to I2C_I2CR clears this bit. 0 Not addressed 1 Addressed as a slave. Set when its own address (I2C_IADR) matches the calling address.
5 IBB	I2C bus busy bit. Indicates the status of the bus. **NOTE:** When I2C is enabled (I2C_I2CR[IEN] = 1), it continuously polls the bus data (SDA) and clock (SCL) signals to determine a Start or Stop condition. 0 Bus is idle. If a Stop signal is detected, IBB is cleared. 1 Bus is busy. When Start is detected, IBB is set.
4 IAL	Arbitration lost. Set by hardware in the following circumstances (IAL must be cleared by software by writing a "0" to it at the start of the interrupt service routine): • I2Cn_SDA input samples low when the master drives high during an address or data-transmit cycle. • I2Cn_SDA input samples low when the master drives high during the acknowledge bit of a data-receive cycle. For the above two cases, the bit is set at the falling edge of the ninth I2Cn_SCL clock during the ACK cycle. • A Start cycle is attempted when the bus is busy.

Table continues on the next page...

I2Cx_I2SR field descriptions (continued)

Field	Description
	• A Repeated Start cycle is requested in Slave mode. • A Stop condition is detected when the master did not request it. **NOTE:** Software cannot set the bit. 0 No arbitration lost. 1 Arbitration is lost.
3 Reserved	This read-only field is reserved and always has the value 0.
2 SRW	Slave read/write. When the I2C is addressed as a slave, IAAS is set, and the slave read/write bit (SRW) indicates the value of the R/W command bit of the calling address sent from the master. SRW is valid only when a complete transfer has occurred, no other transfers have been initiated, and the I2C is a slave and has an address match. 0 Slave receive, master writing to slave 1 Slave transmit, master reading from slave
1 IIF	I2C interrupt. Must be cleared by the software by writing a "0" to it in the interrupt routine. **NOTE:** The software cannot set the bit. 0 No I2C interrupt pending. 1 An interrupt is pending. This causes a processor interrupt request (if the interrupt enable is asserted [IIEN = 1]). The interrupt is set when one of the following occurs: • One byte transfer is completed (the interrupt is set at the falling edge of the ninth clock). • An address is received that matches its own specific address in Slave Receive mode. • Arbitration is lost.
0 RXAK	Received acknowledge. This is the value received from the I2Cn_SDA input for the acknowledge bit during a bus cycle. 0 An "acknowledge" signal was received after the completion of an 8-bit data transmission on the bus. 1 A "No acknowledge" signal was detected at the ninth clock.

17.1.6.5 I2C Data I/O Register (I2Cx_I2DR)

In Master Receive mode, reading the data register allows a read to occur and initiates the next byte to be received. In Slave mode, the same function is available after it is addressed.

Address: Base address + 10h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								DATA							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Cx_I2DR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value 0.
DATA	Data Byte. Holds the last data byte received or the next data byte to be transferred. Software writes the next data byte to be transmitted or reads the data byte received. **NOTE:** The core-written value in I2C_I2DR cannot be read back by the core. Only data written by the I2C bus side can be read.

17.2 Universal Asynchronous Receiver/Transmitter (UART)

17.2.1 Overview

The Universal Asynchronous Receiver/Transmitter (UART) provides serial communication capability with external devices through a level converter and an RS-232 cable or through use of external circuitry that converts infrared signals to electrical signals (for reception) or transforms electrical signals to signals that drive an infrared LED (for transmission) to provide low speed IrDA compatibility.

UART supports NRZ encoding format , RS485 compatible 9 bit data format and IrDA-compatible infrared slow data rate (SIR) format.

17.2.1.1 Block Diagram

The following figure shows the UART block diagram.

The "Module Clock" is the UART_CLK and the "Peripheral Clock" is the IPG_CLK. Both come from CCM.

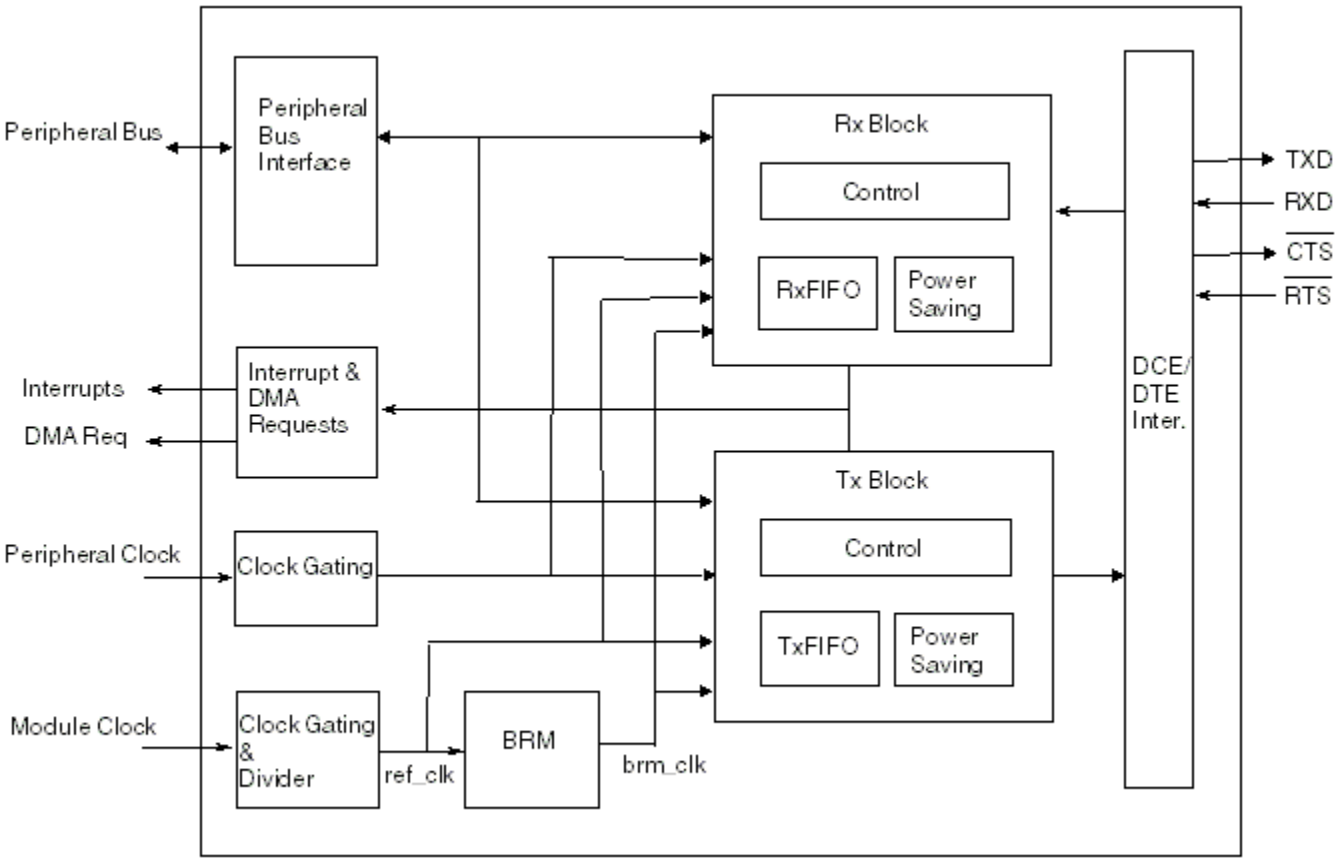


Figure 17-9. UART Block Diagram

17.2.1.2 Features

The UART includes the following features:

- High-speed TIA/EIA-232-F compatible, up to Mbit/s
- Serial IR interface low-speed, IrDA-compatible (up to 115.2 Kbit/s)
- 9-bit or Multidrop mode (RS-485) support (automatic slave address detection)
- 7 or 8 data bits for RS-232 characters, or 9 bit RS-485 format
- 1 or 2 stop bits
- Programmable parity (even, odd, and no parity)
- Hardware flow control support for request to send (RTS_B) and clear to send (CTS_B) signals
- RS-485 driver direction control via CTS_B signal
- Edge-selectable RTS_B and edge-detect interrupts
- Status flags for various flow control and FIFO states
- Voting logic for improved noise immunity (16x oversampling)
- Transmitter FIFO empty interrupt suppression

- UART internal clocks enable/disable
- Auto baud rate detection (up to 115.2 Kbit/s)
- Receiver and transmitter enable/disable for power saving
- RX_DATA input and TX_DATA output can be inverted respectively in RS-232/ RS-485 mode
- DCE/DTE capability
- RTS_B, IrDA asynchronous wake (AIRINT), receive asynchronous wake (AWAKE) interrupts wake the processor from STOP mode
- Maskable interrupts
- Two DMA Requests (TxFIFO DMA Request and RxFIFO DMA Request)
- Escape character sequence detection
- Software reset (SRST_B)
- Two independent, 32-entry FIFOs for transmit and receive
- The peripheral clock can be totally asynchronous with the module clock. The module clock determines baud rate. This allows frequency scaling on peripheral clock (such as during DVFS mode) while remaining the module clock frequency and baud rate.

17.2.1.3 Modes of operation

- Serial RS-232NRZ mode
- 9-bit RS-485 mode
- IrDA mode

To set UART in different modes, see the table below.

Table 17-2. UART mode definition

MDEN (UMCR[0])	IREN (UCR1[7])	UART Mode	Descriptioin
0	0	RS-232	RXD/TXD data is serial RS-232 NRZ format
0	1	IrDA (Interface)	RXD/TXD data is IrDA-compatible infrared slow data rate (SIR) format
1	0	RS-485	RXD/TXD data is RS485 compatible 9 bit data format
1	1	Undefined	Undefined

17.2.2 Functional Description

This section provides a complete functional description of the block.

17.2.2.1 Interrupts and DMA Requests

See the following table for the lists of all interrupt and DMA signals and associated interrupt and DMA sources of the UART. See register description section for explanation of interrupt/DMA enable and status.

Table 17-3. Interrupts and DMA

Interrupt/DMA Output	Interrupt/DMA Enable	Enable Register Location	Interrupt/DMA Flag	Flag Register Location
interrupt_uart	RRDYEN	UCR1 (bit 9)	RRDY	USR1 (bit 9)
	IDEN	UCR1 (bit 12)	IDLE	USR2 (bit 12)
	DREN	UCR4 (bit 0)	RDR	USR2 (bit 0)
	RXDSEN	UCR3 (bit 6)	RXDS	USR1 (bit 6)
	ATEN	UCR2 (bit 3)	AGTIM	USR1 (bit 8)
interrupt_uart	TXMPTYEN	UCR1 (bit 6)	TXFE	USR2 (bit 14)
	TRDYEN	UCR1 (bit 13)	TRDY	USR1 (bit 13)
	TCEN	UCR4 (bit 3)	TXDC	USR2 (bit 3)
interrupt_uart	OREN	UCR4 (bit 1)	ORE	USR2 (bit 1)
	BKEN	UCR4 (bit 2)	BRCD	USR2 (bit 2)
	WKEN	UCR4 (bit 7)	WAKE	USR2 (bit 7)
	ADEN	UCR1 (bit 15)	ADET	USR2 (bit 15)
	ACIEN	UCR3 (bit 0)	ACST	USR2 (bit 11)
	ESCI	UCR2 (bit 15)	ESCF	USR1 (bit 11)
	ENIRI	UCR4 (bit 8)	IRINT	USR2 (bit 8)
	AIRINTEN	UCR3 (bit 5)	AIRINT	USR1 (bit 5)
	AWAKEN	UCR3 (bit 4)	AWAKE	USR1 (bit 4)
	FRAERREN	UCR3 (bit 11)	FRAERR	USR1 (bit 10)
	PARERREN	UCR3 (bit 12)	PARITYERR	USR1 (bit 15)
	RTSDEN	UCR1 (bit 5)	RTSD	USR1 (bit 12)
	RTSEN	UCR2 (bit 4)	RTSF	USR2 (bit 4)
	DTREN (DCE)	UCR3 (bit 13)	DTRF	USR2 (bit 13)
	RI (DTE)	UCR3 (bit 8)	RIDELT	USR2 (bit 10)
	DCD (DTE)	UCR3 (bit 9)	DCDDELT	USR2 (bit 6)
	DTRDEN	UCR3 (bit 3)	DTRD	USR1 (bit 7)
	SADEN	UMCR (bit 3)	SAD	USR1 (bit 3)
dma_req_rx	RXDMAEN	UCR1 (bit 8)	RRDY	USR1 (bit 9)
	ATDMAEN	UCR1 (bit 2)	AGTIM	USR1 (bit 8)
	IDDMAEN	UCR4 (bit 6)	IDLE	USR2 (bit 12)
dma_req_tx	TXDMAEN	UCR1 (bit 3)	TRDY	USR1 (bit 13)

17.2.2.2 Clocks

This section describes clocks and special clocking requirements of the UART.

17.2.2.2.1 Clock requirements

UART module receives 2 clocks, *peripheral_clock* and *module_clock*. The *peripheral_clock* is used as write clock of the TxFIFO, read clock of the RxFIFO and synchronization of the modem control input pins. It must always be running when UART is enabled. There is an exception in stop mode (see Clocking in Low-Power Modes).

The *module_clock* is for all the state machines, writing RxFIFO, reading TxFIFO, etc. It must always be running when UART is sending or receiving characters.This clock is used in order to allow frequency scaling on *peripheral_clock* without changing configuration of baud rate (*module_clock* staying at a fixed frequency).

The constraints on *peripheral_clock* and *module_clock* are as follows:

- *peripheral_clock* and *module_clock* can totally be asynchronous. They can also be synchronous.
- Due to the 16x oversampling of the incoming characters, *module_clock* frequency must always be greater or equal to 16x the maximum baud rate. For example, if max baud rate is 4 Mbit/s, *module_clock* must be greater or equal to 4 M x 16 = 64 MHz.

NOTE

The restriction that *peripheral_clock* frequency must be higher or equal to 16x baud rate has been removed. There is no limitation on *peripheral_clock* frequency to baud rate.

17.2.2.2.2 Maximum Baud Rate

The max baud rate the UART can support is determined by the max frequency of the module_clock.

For example, if the SoC can provide the fastest module_clock 66.5 MHz, the UART can transmit and receive serial data with the maximum baud rate 66.5M/16 = 4.15 Mbit/s.

The UART supports serial IR interface low speed. In the low speed IrDA mode, the max baud rate is 115.2 Kbit/s. To support the 115.2 Kbit/s, module_clock frequency must be higher or equal to 1.8432 MHz.

17.2.2.2.3 Clocking in Low-Power Modes

The UART supports 2 low-power modes: DOZE and STOP.

In STOP mode (input pin *stop_req* is at '1'), the UART doesn't need any clock. In this mode the UART can wake-up the Arm platform with the asynchronous interrupts (see Low Power Modes).

- If before entering in STOP mode the software has enabled RTSDEN interrupt, when RTS will change state (put at '0' by external device started to send), the asynchronous interrupt will wake-up the system, *peripheral_clock* and *module_clock* will be provided to the UART before first start bit, so that no data will be lost.
- If RTS doesn't change state (already at '0' before entering in STOP mode), then wake-up interrupt (AWAKE) will be sent at the arrival of first Start bit (on falling edge). In this case, the UART must receive the *peripheral_clock* and *module_clock* during the first half of start bit to correctly receive this character (for example, at 115.2 Kbit/s, UART must receive *peripheral_clock* and *module_clock* at maximum 4.3 microseconds after falling edge of Start bit). If the UART receives *peripheral_clock* and *module_clock* too late, first character will be lost, and so should be dropped. Also, if autobaud detection is enabled, the first character won't be correctly received and another autobaud detection will need to be initiated.

In Doze mode, UART behavior is programmable through DOZE bit (UCR1[1]). If DOZE bit is set to '1', then UART is disabled in Doze mode, and in consequence, UART clocks can be switched-off (after being sure UART is not transmitting nor receiving). On the contrary, if DOZE bit is set to '0', UART is enabled and it must receive *peripheral_clock* and *module_clock* .

17.2.2.3 General UART Definitions

Definitions of terms that occurs the following discussions are given in this section.

- Bit Time-The period of time required to serially transmit or receive 1 bit of data (1 cycle of the baud rate frequency).
- Start bit-The bit time of a logic 0 that indicates the beginning of a data frame. A start bit begins with a 1-to-0 transition, and is preceded by at least 1 bit time of logic 1.
- Stop bit-1 bit time of logic 1 that indicates the end of a data frame.
- BREAK-A frame in which all of the data bits, including the stop bit, are logic 0. This type of frame is usually sent to signal the end of a message or the beginning of a new message.
- Mark - When no data is being sent, the serial port's transmit pin's voltage is 1 and is said to be in a MARK state.

- Space - The serial port can also be forced to keep the transmit pin at a 0 and is said to be the SPACE or BREAK state.
- Frame-A start bit followed by a specified number of data or information bits and terminated by a stop bit. The number of data or information bits depends on the format specified and must be the same for the transmitting device and the receiving device. The most common frame format is 1 start bit followed by 8 data bits (least significant bit first) and terminated by 1 stop bit. An additional stop bit and a parity bit also can be included.
- Framing Error-An error condition that occurs when the stop bit of a received frame is missing, usually when the frame boundaries in the received bit stream are not synchronized with the receiver bit counter. Framing errors can go undetected if a data bit in the expected stop bit time happens to be a logic 1. A framing error is always present on the receiver side when the transmitter is sending BREAKs. However, when the UART is programmed to expect 2 stop bits and only the first stop bit is received, this is not a framing error by definition.
- Parity Error-An error condition that occurs when the calculated parity of the received data bits in a frame does not match the parity bit received on the RX_DATA input. Parity error is calculated only after an entire frame is received.
- Idle-One in NRZ encoding format and selectable polarity in IrDA mode.
- Overrun Error-An error condition that occurs when the latest character received is ignored to prevent overwriting a character already present in the UART receive buffer (RxFIFO). An overrun error indicates that the software reading the buffer (RxFIFO) is not keeping up with the actual reception of characters on the RX_DATA input.

17.2.2.3.1 RTS_B - UART Request To Send

The UART Request To Send input controls the transmitter. The modem or other terminal equipment signals the UART when it is ready to receive by setting '0' on the RTS_B pin.

Normally, the transmitter waits until this signal is active (low) before transmitting a character, however when the Ignore RTS (IRTS) bit is set, the transmitter sends a character as soon as it is ready to transmit. An interrupt (RTSD) can be posted on any transition of this pin and can wake the Arm platform from STOP mode on its assertion. When RTS_B is set to '1' during a transmission, the UART transmitter finishes transmitting the current character and shuts off. The contents of the TxFIFO (characters to be transmitted) remain undisturbed. The operation of this input is the same regardless of whether the UART is in DTE or DCE mode.

17.2.2.3.2 RTS Edge Triggered Interrupt

The input to the RTS_B pin can be programmed to generate an interrupt on a selectable edge.

See the table below for summary of the operation of the RTS edge triggered interrupt (RTSF).

To enable the RTS_B pin to generate an interrupt, set the request to send interrupt enable (RTSEN) bit (UCR2[4]) to 1. Writing 1 to the RTS_B edge triggered interrupt flag (RTSF) bit (USR2[4]) clears the interrupt flag. The interrupt can occur on the rising edge, falling edge, or either edge of the RTS_B input. The request to send edge control (RTEC) field (UCR2[10:9]) programs the edge that generates the interrupt. When RTEC is set to 0x00 and RTSEN = 1, the interrupt occurs on the rising edge (default). When RTEC is set to 0x01 and RTSEN = 1, the interrupt occurs on the falling edge. When RTEC is set to 0x1X and RTSEN = 1, the interrupt occurs on either edge. This is a synchronous interrupt. The RTSF bit is cleared by writing 1 to it. Writing 0 to RTSF has no effect.

Table 17-4. RTS_B Edge Triggered Interrupt Truth Table

RTS_B	RTSEN	RTEC [1]	RTEC [0]	RTSF	Interrupt Occurs On…	interrupt_uart
X	0	X	X	0	Interrupt disabled	1
1->0	1	0	0	0	Rising edge	1
0->1	1	0	0	1	Rising edge	0
1->0	1	0	1	1	Falling edge	0
0->1	1	0	1	0	Falling edge	1
1->0	1	1	X	1	Either edge	0
0->1	1	1	X	1	Either edge	0

There is another RTS_B interrupt that is not programmable. The status bit RTSD asserts the *interrupt_uart* interrupt when the RTS_B delta interrupt enable = 1. This is an asynchronous interrupt. The RTSD bit is cleared by writing 1 to it. Writing 0 to the RTSD bit has no effect.

17.2.2.3.3 CTS_B - Clear To Send

This output pin serves two purposes. Normally, the receiver indicates that it is ready to receive data by asserting this pin (low). When the CTS_B trigger level is programmed to trigger at 32 characters received and the receiver detects the valid start bit of the 33 character, it de-asserts this pin. The operation of this output is the same regardless of whether the UART is in DTE or DCE mode.

17.2.2.3.4 Programmable CTS_B Deassertion

The CTS_B output can also be programmed to deassert when the RxFIFO reaches a certain level. Setting the CTS trigger level (UCR4[15:10]) at any value less than 32 deasserts the CTS_B pin on detection of the valid start bit of the N + 1 character (where N is the trigger level setting). However, the receiver continues to receive characters until the RxFIFO is full.

17.2.2.3.5 TX_DATA - UART Transmit

This is the transmitter serial output. When operating in RS-232/RS-485 mode, NRZ encoded data is transmitted, and the data can be inverted (controlled by INVT (UCR3[1])) before transmitted. When operating in infrared mode, a 3/16 bit-period pulse is output for each 0 bit transmitted, and no pulse is output for each 1 bit transmitted.

For RS-232/RS-485 applications, this pin must be connected to an RS-232/RS-485 transmitter. The operation of this output is the same regardless of whether the UART is in DTE or DCE mode. See Figure 17-10.

17.2.2.3.6 RX_DATA - UART Receive

This is the receiver serial input. When operating in RS-232/RS-485 mode, NRZ encoded data is expected, and the data can be inverted (controlled by INVR (UCR4[9])) before sampled. When operating in infrared mode, a narrow pulse is expected for each 0 bit received and no pulse is expected for each 1 bit received.

External circuitry must convert the IR signal to an electrical signal. RS-232/RS-485 applications require an external RS-232/RS-485 receiver to convert voltage levels. The operation of this input is the same regardless of whether the UART is in DTE or DCE mode. See the figure below.

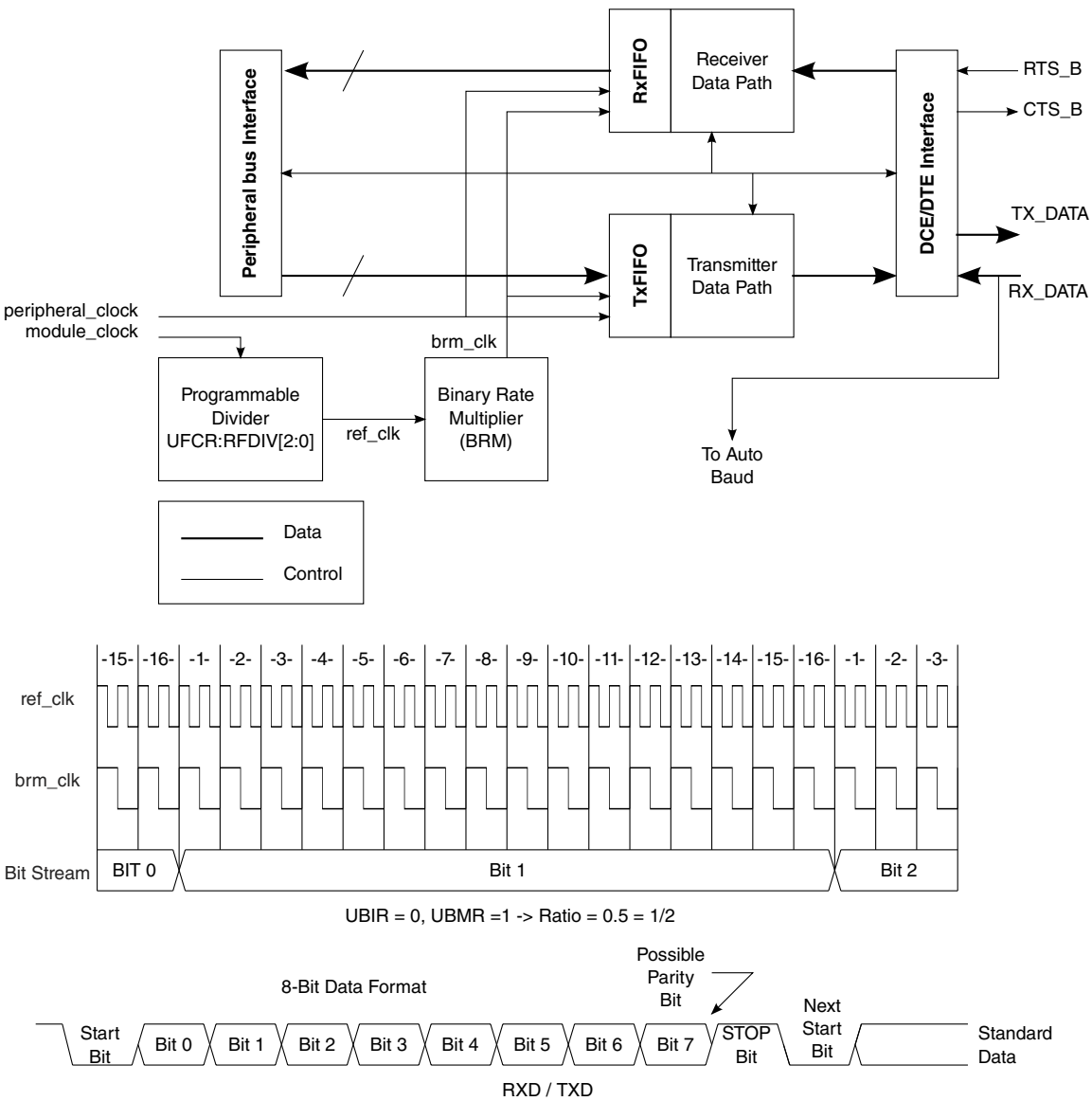


Figure 17-10. UART Simplified Block and Clock Generation Diagrams

17.2.2.4 Transmitter

The transmitter accepts a parallel character from the Arm platform and transmits it serially. The start, stop, and parity (when enabled) bits are added to the character.

When the ignore RTS bit (IRTS) is set, the transmitter sends a character as soon as it is ready to transmit. RTS_B can be used to provide flow-control of the serial data. When RTS_B is set to '1', the transmitter finishes sending the character in progress (if any), stops, and waits for RTS_B to be set to '0' again. Generation of BREAK characters and parity errors (for debugging purposes) is supported. The transmitter operates from the clock provided by the Binary Rate Multiplier(BRM). Normal NRZ encoded data is transmitted when the IR interface is disabled.

The transmitter FIFO (TxFIFO) contains 32 bytes. The data is written to TxFIFO by writing to the UTXD register with the byte data to the [7:0] bits. The data is written consecutively if the TxFIFO is not full. It is read (internally) consecutively if the TxFIFO is not empty. TXFULL bit (UTS[4]) can be used to control whether TXFIFO is full or not. The TxFIFO can be written regardless of the transmitter is disabled or enabled. If the UART is disabled, user can still write data into the TxFIFO correctly. But in this case the write access will yield to a transfer error.

17.2.2.4.1 Transmitter FIFO Empty Interrupt Suppression

The transmitter FIFO empty interrupt suppression logic suppresses the TXFE interrupt between writes to the TxFIFO.

When TxFIFO is empty, the software can either send one or several characters. If the software sends one character, it would write the character into the UTXD register, then that character is immediately transferred to the transmitter shift register, assuming the transmitter is already enabled. Without interrupt suppression logic, the TXFE interrupt flag would be set immediately. But, with this logic, the interrupt flag is set when the last bit of the character has been transmitted, for example, before the transmission of the parity bit (if exists) and the stop bit(s).

So, the suppression logic doesn't immediately send the TXFE interrupt flag. It allows the software to write another character to the TxFIFO before the interrupt flag is asserted.

When the transmitter shift register empties before another character is written to the TxFIFO, the interrupt flag is asserted. Writing data to the TxFIFO would release the interrupt flag. The interrupt flag is asserted on the following conditions:

- System Reset
- UART software reset

- When a single character has been written to Transmitter FIFO and then the Transmitter FIFO and the Transmitter Shift Register become empty until another character is written to the Transmitter FIFO
- The last character in the TxFIFO is transferred to the shift register, when TxFIFO contains two or more characters. See the figure below.

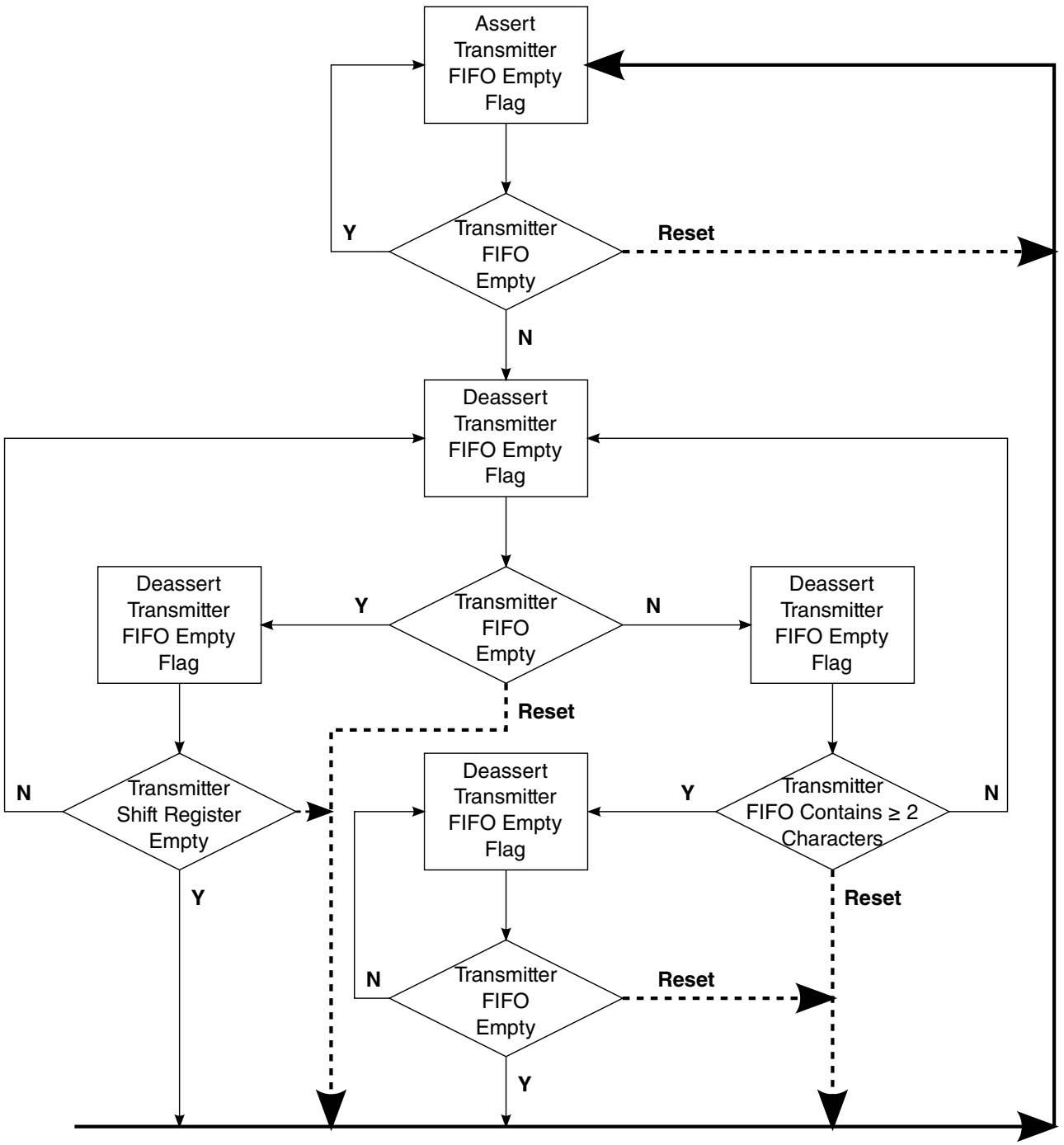


Figure 17-11. Transmitter FIFO Empty Interrupt Suppression Flow Chart

17.2.2.4.2 Transmitting a Break Condition

Asserting SNDBRK bit of the UCR1 Register forces the transmitter to send a break character (continuous zeros). The transmitter will finish sending the character in progress (if any) before sending break until this bit is reset.

The user is responsible to ensure that this bit is high for long enough to generate a valid BREAK. The transmitter samples SNDBRK after every bit is transmitted. Following completion of the BREAK transmission, the UART will transmit two mark bits. The user can continue to fill the FIFO and any character remaining will be transmitted when the break is terminated.

17.2.2.5 Receiver

See the figure below for the receiver flow chart.

The receiver accepts a serial data stream and converts it into parallel characters. When enabled, it searches for a start bit, qualifies it, and samples the following data bits at the bit-center.

Jitter tolerance and noise immunity are provided by sampling at a 16x rate and using voting techniques to clean up the samples. Once the start bit is found, the data bits, parity bit (if enabled), and stop bits (either 1 or 2 depending on user selection) are shifted in. Parity is checked and its status reported in the URXD register when parity is enabled. Frame errors and BREAKs are also checked and reported. When a new character is ready to be read by the Arm platform from the RxFIFO, the receive data ready (RDR = USR2[0]) bit is asserted and an interrupt is posted (if DREN = UCR4[0] = 1). If the receiver trigger level is set to 2 (RXTL[5:0] = UFCR[5:0] = 2), and 2 chars have been received into RxFIFO, the receiver ready interrupt flag (RRDY = USR1[9]) is asserted and an interrupt is posted if the receiver ready interrupt enable bit is set (RRDYEN = UCR1[9] = 1). If the UART Receiver Register (URXD) is read once, and in consequence there is only 1 character in the RxFIFO, the interrupt generated by the RDR bit is automatically cleared. The RRDY bit is cleared when the data in the RxFIFO falls below the programmed trigger level.

Normal NRZ encoded data is expected when the IR interface is disabled. The RxFIFO contains 32 half-word entries. Characters received are written consecutively into this FIFO. If the FIFO is full and a 33rd characters is received, this character will be ignored and the USR2[ORE] bit will be set.

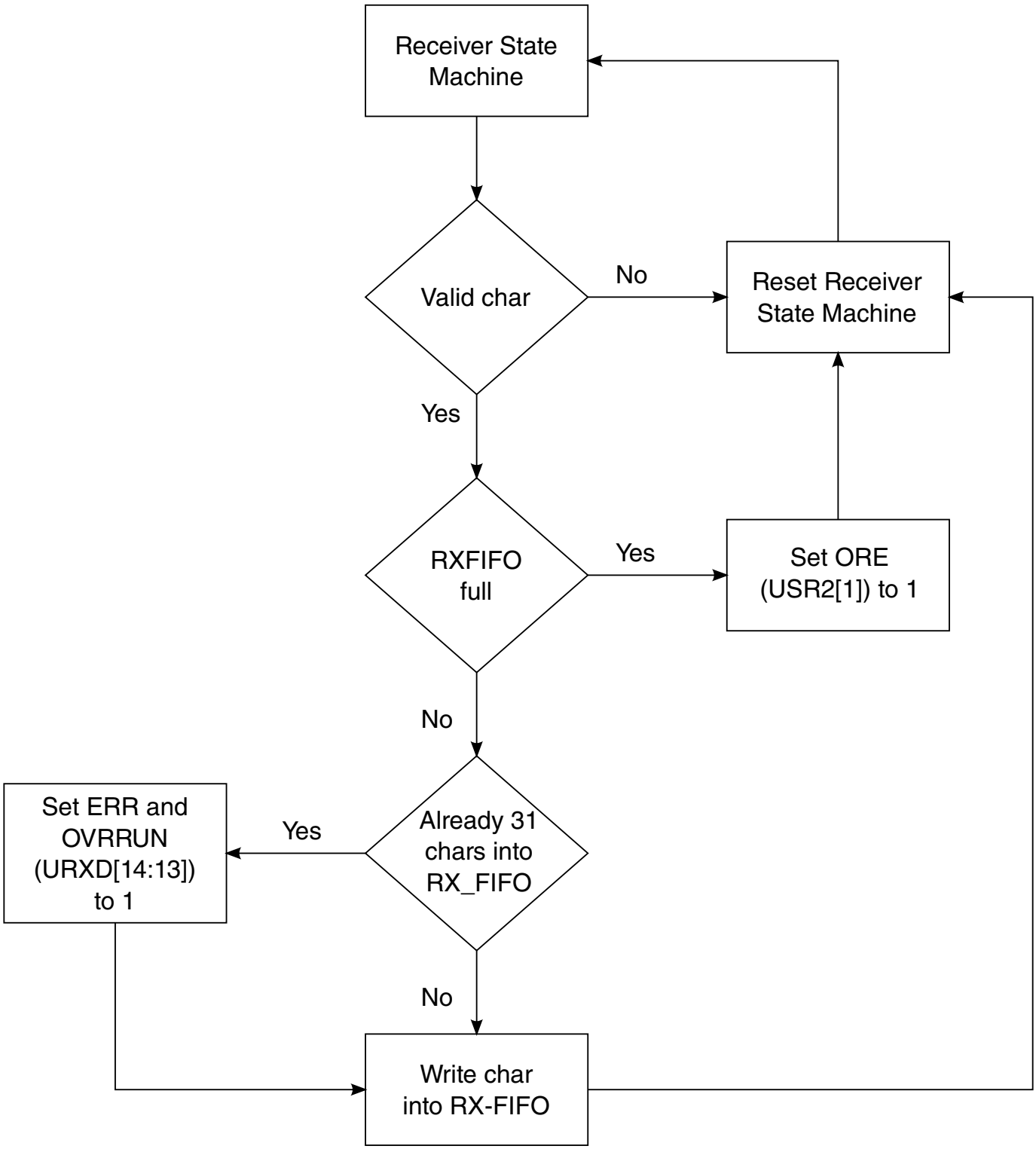


Figure 17-12. Receiver Flow Chart

17.2.2.5.1 Idle Line Detect

The receiver logic block includes the ability to detect an idle line. Idle lines indicate the end or the beginning of a message.

For an idle condition to occur:

- RxFIFO must be empty and
- RX_DATA pin must be idle for more than a configured number of frames (ICD[1:0] = UCR1[11:10]).

When the idle condition detected interrupt enable (IDEN = UCR1[12]) is set and the line is idle for 4 (default), 8, 16, or 32 (maximum) frames, the detection of an idle condition flags an interrupt (see the table below). When an idle condition is detected, the IDLE (USR2[12]) bit is set. Clear the IDLE bit by writing 1 to it. Writing 0 to the IDLE bit has no effect.

Table 17-5. Detection Truth Table

IDEN	ICD [1]	ICD [0]	IDLE	*interrupt_uart*
0	X	X	0	1
1	0	0	asserted after 4 idle frames	asserted after 4 idle frames
1	0	1	asserted after 8 idle frames	asserted after 8 idle frames
1	1	0	asserted after 16 idle frames	asserted after 16 idle frames
1	1	1	asserted after 32 idle frames	asserted after 32 idle frames
NOTE: This table assumes that no other interrupt is set at the same time this interrupt is set for the *interrupt_uart* signal. This table shows how this interrupt affects the *interrupt_uart* signal.				

During a normal message there is no idle time between frames. When all of the information bits in a frame are logic 1s, the start bit ensures that at least one logic 0 bit time occurs for each frame so that the IDLE bit is not asserted.

17.2.2.5.2 Aging Character Detect

The receiver block also includes the possibility to detect when at least one character has been sitting into the RxFIFO for a time corresponding to 8 characters. This aging character capability allows the UART to inform the Arm platform that there is less character into the RxFIFO than the Rx trigger and, no new character has been detected on the RXD line.

The aging capability is a timer which starts to count as soon as the RxFIFO is not empty and its trigger level is not reached (RRDY=0). This counter is reset when either a RxFIFO read is performed or another character starts to present on the RXD line. If none of those two events occurs, the bit AGTIM (USR1[8]) is set when the counter has

measured a time corresponding to 8 characters. AGTIM is cleared by writing a 1 to it. AGTIM can flag an interrupt to Arm platform on *interrupt_uart* if ATEN (UCR2[3]) has been set.

To summarize, AGTIM is set when:

- There is at least one character into RxFIFO.
- No read has occurred on RxFIFO and RXD line has stayed high, for a time corresponding to 8 characters.
- The RxFIFO trigger is not reached (RRDY=0)

17.2.2.5.3 Receiver Wake

The WAKE bit (USR2[7]) is set when the receiver detects a qualified Start bit. For this, two conditions must be fulfilled, firstly a falling edge on RX_DATA line must be detected and secondly the RX_DATA line must stays at low level for more than a half-bit duration.

When the wake interrupt enable WKEN (UCR4[7]) bit is enabled, the receiver flags an interrupt (*interrupt_uart*) if the WAKE status bit is set. The WAKE bit is cleared by writing 1 to it. Writing 0 to the WAKE bit has no effect. The WAKE status bit can be asserted in either serial RS-232 mode or IR mode. The generation of the WAKE interrupt needs the clock *module_clock* .

When the asynchronous wake interrupt (AWAKE) is enabled (AWAKEN = UCR3[4] = 1), and the Arm platform is in STOP mode, and UART clocks have been shut-off, then a falling edge detected on the receive pin (RX_DATA) asserts the AWAKE bit (USR1[4]) and the *interrupt_uart* interrupt to wake the Arm platform from STOP mode. Re-enable UART clocks and clear the AWAKE bit by writing 1 to it. Writing 0 to the AWAKE bit has no effect. When IR interface is enabled (UCR1[7]=1), the AWAKE bit is always not asserted. The generation of the asynchronous AWAKE interrupt does not need any clocks.

In IR mode, if the asynchronous IR WAKE interrupt is enabled (AIRINTEN = UCR3[5] = 1), and if the Arm platform is in STOP mode (UART clocks are off when Arm platform in STOP mode), then the detection of a falling edge on the receive pin (RXD_IR), asserts the AIRINT bit (USR1[5]), and the *interrupt_uart* interrupt. This interrupt wakes the Arm platform from STOP mode. Software re-enables UART clocks and clear the AIRINT bit by writing 1 to it. Writing 0 to the AIRINT bit has no effect. When IR interface is disabled (UCR1[7]=0), the AIRINT bit is always not asserted. The generation of the asynchronous AIRINT interrupt does not need any clocks.

Recommended procedure for programming the asynchronous interrupts is to first clear them by writing 1 to the appropriate bit in the UART Status Register 1 (USR1). Poll or enable the interrupt for the Receiver IDLE Interrupt Flag (RXDS) in the USR1. When asserted, the RXDS bit indicates to the software that the receiver state machine is in the idle state, the next state is idle, and the RX_DATA pin is idle (high). After following this procedure, enable the asynchronous interrupt and enter STOP mode.

17.2.2.5.4 Receiving a BREAK Condition

A BREAK condition is received when the receiver detects all 0s (including a 0 during the bit time of the stop bit) in a frame. The BREAK condition asserts the BRCD bit (USR2[2]) and writes only the first BREAK character to the RxFIFO. Clear the BRCD bit by writing 1 to it. Writing 0 to the BRCD bit has no effect.

Asserting BRCD would generate an interrupt on *interrupt_uart*. The interrupt generation can be masked using the control bit BKEN (UCR4[2]). Receiving a break condition will also effect the following bits in the receiver register URXD:

URXD(11) = BRK. While high this bit indicates that the current char was detected as a break.

URXD(12) = FRMERR. The frame error bit will always be set when BRK is set.

URXD(10) = PRERR. If odd parity was selected the parity error bit will also be set when BRK is set.

URXD(14) = ERR. The error detect bit indicates that the character present in the rx data field has an error status. This can be asserted by a break.

17.2.2.5.5 Vote Logic

The vote logic block provides jitter tolerance and noise immunity by sampling with respect to a 16x clock (*brm_clk*) and using voting techniques to clean up the samples. The voting is implemented by sampling the incoming signal constantly on the rising edge of the *brm_clk*.

See Figure 17-13. The receiver is provided with the majority vote value, which is 2 out of the 3 samples. For examples of the majority vote results of the vote logic, see the following table.

Table 17-6. Majority Vote Results

Samples	Vote
000	0

Table continues on the next page...

Table 17-6. Majority Vote Results (continued)

Samples	Vote
101	1
001	0
111	1

The vote logic captures a sample on every rising edge of *brm_clk*, however the receiver uses 16x oversampling to take its value in the middle of the sample character.

The receiver starts to count when the Start bit is set however it does not capture the contents of the RxFIFO at the time the Start bit is set. The start bit is validated when 0s are received for 7 consecutive 1/16 of bit times following the 1-to-0 transition. Once the counter reaches 0xF, it starts counting on the next bit and captures it in the middle of the sampling frame (see Table 17-6). All data bits are captured in the same manner. Once the stop bit is detected, the receiver shift register (SIPO_OUT) data is parallel shifted to the RxFIFO.

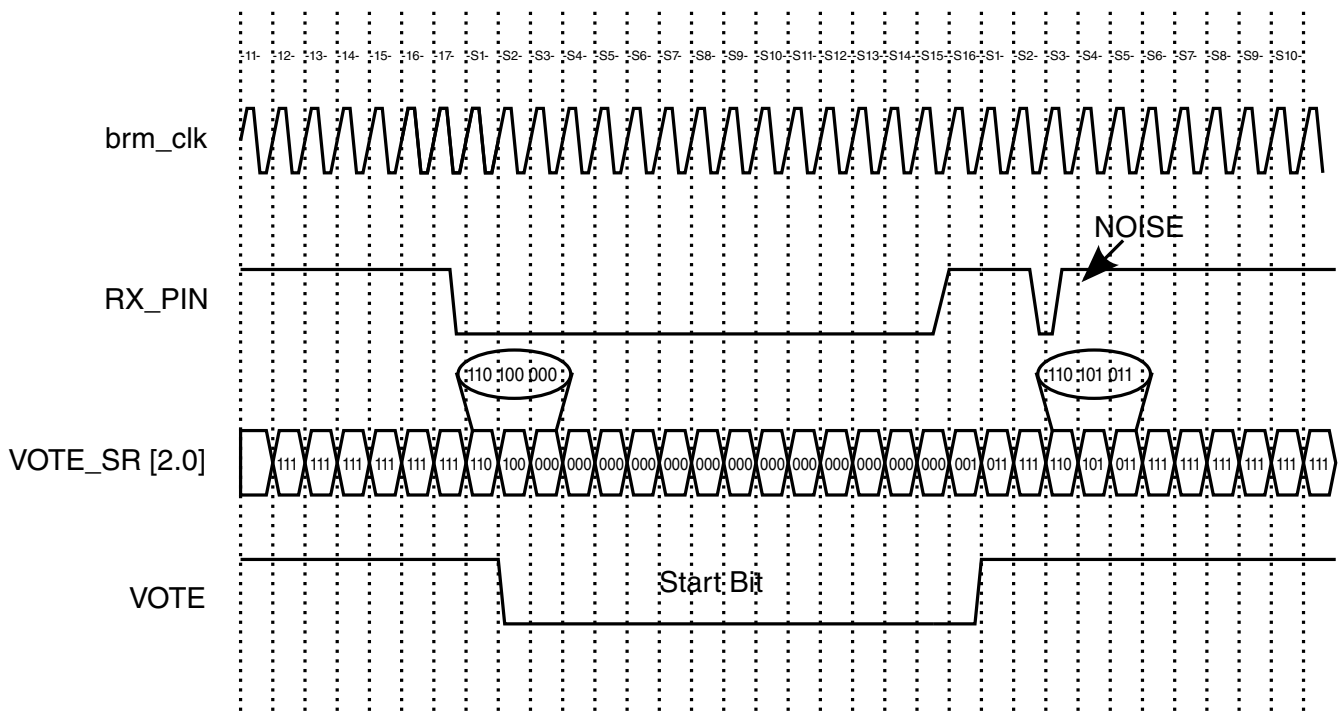


Figure 17-13. Majority Vote Results

A new feature has been recently implemented, it allows to re-synchronize the counter on each edge of RX_DATA line. This is automatic and allows to improve the immunity of UART against signal distortion.

There is a special case when the *brm_clk* frequency is too low and is unable to capture a 0 pulse in IrDA. In this case, the software must set the IRSC (UCR4[5]) bit so that the reference clock (after internal divider) is used for the voting logic. The pulse is validated by counting the length of the pulse.

Refer to Infrared Interface for more details.

17.2.2.5.6 Baud Rate Automatic Detection Logic

When the baud rate automatic detection logic is enabled, the UART locks onto the incoming baud rate. To enable this feature, set the automatic detection of baud rate bit (ADBR = UCR1[14] = 1) and write 1 to the ADET bit (USR2[15]) to clear it.

When ADET=0 and ADBR =1, the detection starts. Then, once the beginning of start bit (transition from 1-to-0 of RX_DATA) has been detected, UART starts a counter (UBRC) working at reference frequency. Once the end of start bit is detected (transition from 0-to-1 of RX_DATA), the value of UBRC - 1 is directly copied into UBMR register. UBIR register is filled with 0x000F.

So, at the end of start bit, registers gets following values:

```
UBRC = number of reference clock periods (after divider) during Start bit.
UBIR = 0x000F
UBMR = UBRC - 1
```

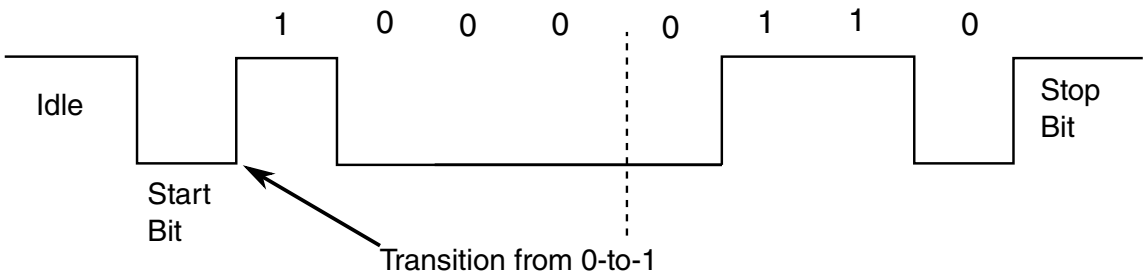
The updated values of the 3 registers can be read.

See Table 17-7 for list of parameters for baud rate detection and Figure 17-14 for baud rate detection protocol diagram.

If any of the UART BRM registers are simultaneously written by the baud rate automatic detection logic and by the peripheral data bus, the peripheral data bus would have lower priority.

Table 17-7. Baud Rate Automatic Detection

ADBR	ADET	Baud Rate Detection	*interrupt_uart*
0	X	Manual Configuration	1
1	0	Auto Detection Started	1
1	1	Auto Detection Complete	0
NOTE: This table assumes that no other interrupt is set at the same time this interrupt is set for the *interrupt_uart* signal.			



Note: LSB Transmitted first.

Figure 17-14. Baud Rate Detection Protocol Diagram

17.2.2.5.6.1 Baud Rate Automatic Detection Protocol

The receiver must receive an ASCII character "A" or "a" to verify proper detection of the incoming baud rate. When an ASCII character "A" (0x41) or "a" (0x61) is received and no error occurs, the Automatic Detect baud rate bit is set (ADET=1) and if the interrupt is enabled (ADEN=UCR1[15]=1), an interrupt *interrupt_uart* is generated.

When an ASCII character "A" or "a" is not received (because of a bit error or the reception of another character), the auto detection sequence restarts and waits for another 1-to-0 transition.

As long as ADET = 0 and ADBR = 1, the UART continues to try to lock onto the incoming baud rate. Once the ASCII character "A" or "a" is detected and the ADET bit is set, the receiver ignores the ADBR bit and continues normal operation with the calculated baud rate.

The UART interrupt is active (*interrupt_uart* = 0) as long as ADET = 1 and ADBR = 1. This can be disabled by clearing the automatic baud rate detection interrupt enable bit (ADEN = 0). Before starting an automatic baud rate detection sequence, set ADET = 0 and ADBR = 1.

The RxFIFO must contain the ASCII character "A" or "a" following the automatic baud rate detection interrupt.

The 16-bit UART Baud Rate Count Register (UBRC) is reset to 4 and stays at 0xFFFF when an overflow occurs. The UBRC register counts (measures) the duration of start bit. When the start bit is detected and counted, the UART Baud Rate Count Register retains its value until the next automatic baud rate detection sequence is initiated.

The Baud Rate Count Register counts only when auto detection is enabled.

17.2.2.5.6.2 New Baud Rate Determination

In order to fight against the problems caused by the distortion and the noise on the RX_DATA line, the duration of the baud rate measurement has been extended.

Previously, as described above, this determination was based on the measurement of the START bit duration. Now, this measurement is based on the duration of START bit + bit0. Bit0 is the first bit following the START bit. In fact, the counter which is started at the falling edge of START bit is no longer stopped at next rising edge (end of START bit), but it is stopped at the next falling edge (end of bit0). As the character sent is always a "A" (41h) or a "a" (61h), this second falling edge will always be present and it will indicate the end of bit0. Once this counter is stopped, the result is divided by 2 and used by the BRM to determine the incoming baud rate.

NOTE

UBRC register contains the result of this division by two, in consequence it reflects the measurement of the duration of one bit.

17.2.2.5.6.2.1 New Autobaud Counter Stopped bit and Interrupt

A new bit has been added in USR2 register: ACST (USR2[11]). This bit is set immediately after the determination of the baud rate.

So,

- if ADNIMP is not set (default), ACST is set to 1 after the end of bit0,
- If ADNIMP is set to 1, ACST is set to 1 at the end of START bit.

If ACIEN (UCR3[0]) is set to 1, ACST will flag an interrupt on *interrupt_uart* signal. This interrupt informs the Arm platform that the BRM has just been set with the result of the bit length measurement. If needed, the Arm platform can perform a read of UBMR (or UBRC) register and determine by itself the baud rate measured. Then the Arm platform has the possibility to correct the BRM registers with the nearest standardized baud rate.

NOTE

ACST is set only if ADBR is set to 1, for example, the UART is autobauding.

Clear the ACST bit by writing 1 to it. Writing 0 to the ACST bit has no effect.

17.2.2.6 Escape Sequence Detection

An escape sequence typically consists of 3 characters entered in rapid succession (such as +++). Because these are valid characters by themselves, the time between characters determines if it is a valid escape sequence.

Too much time between two of the "+" characters is interpreted as two "+" characters, and not part of an escape sequence.

The software chooses the escape character and writes its value to the UART Escape Character Register (UESC). The software must also enable escape detection feature by setting ESCEN (UCR2[11]) to 1. The hardware compares this value to incoming characters in the RxFIFO. When an escape character is detected, the internal escape timer starts to count. The software specifies a time-out value for the maximum allowable time between 2 successive escape characters (see the table below). The escape timer is programmable in intervals of 2 ms to a maximum interval of 8.192 seconds.

Table 17-8. Escape Timer Scaling

UTIM Register	Maximum Time Between Specified Escape Characters
0x000	2 ms
0x001	4 ms
0x002	6 ms
0x003	8 ms
0x004	10 ms
...	...
0F8	498 ms
0F9	500 ms
...	...
9C3	5 s
...	...
FFD	8.188 s
FFE	8.190 s
FFF	8.192 s
NOTE: To calculate the time interval: (UTIM_Value + 1) x 0.002 = Time_Interval Example: (09C3 + 1) x 0.002 = 5 s.	

The escape sequence detection feature is available for all the reference frequencies. Before using Escape Sequence Detection, the user must fill the ONEMS register. This 24-bit register must contain the value of the UART internal frequency divided by 1000. The internal frequency is obtained after the UART internal divider which is applied on *module_clock* clock.

Example I:

- If the input clock *module_clock* frequency is 66.5 MHz.
- And if the input clock *module_clock* is divided by 2 with the internal divider: UFCR[9:7] = 3'b100

$$ONEMS = \frac{66.5 \times 10^6}{2 \times 1000} = 33250 = 81E2h$$

Figure 17-15. Calculation of Frequency for ONEMS Register

Example II:

- If the input clock *module_clock* frequency is 66.5 MHz.
- And if the input clock *module_clock* is divided by 1with the internal divider: UFCR[9:7] = 3'b101

$$ONEMS = \frac{66.5 \times 10^6}{1000} = 66500 = 103C4h$$

Figure 17-16. Calculation of Frequency for ONEMS Register

The escape sequence detection interrupt is asserted when the escape sequence interrupt enable (ESCI) bit is set and an escape sequence is detected (ESCF set). Clear the ESCF bit by writing 1 to it. Writing 0 to the ESCF bit has no effect.

17.2.3 Binary Rate Multiplier (BRM)

The BRM sub-block receives *ref_clk* (*module_clock* clock after divider). From this clock, and with integer and non-integer division, BRM generates a 16x baud rate clock .

The UART transmitte will shift data out based on this 16x baud rate clock. The UART receiver will sample the serial data line based on this 16x baud rate clock. The input and output frequency ratio is programmed in the UART BRM Incremental Register (UBIR) and UART BRM MOD Register (UBMR). The output frequency is divided by the input frequency to produce this ratio. For integer division, set the UBIR = 0x000F and write the divisor to the UBMR register. All values written to these registers must be one less than the actual value to eliminate division by 0 (undefined), and to increase the maximum range of the registers.

Updating the BRM registers requires writing to both registers. The UBIR register must be written before writing to the UBMR register. If only one register is written to by the software, the BRM continues to use the previous values.

The following examples show how to determine what values are to be programmed into UBIR and UBMR for a given reference frequency and desired baud rate. The following equation can be used to help determine these values:

$$\text{BaudRate} = \frac{\text{Ref Freq}}{\left(16 \times \frac{\text{UBMR} + 1}{\text{UBIR} + 1} \right)}$$

Figure 17-17. Frequency and Baud Rate for UBIR and UBMR

With:

Reference Frequency (Hz): UART Reference Frequency (*module_clock* after RFDIV divider)

Baud Rate (bit/s): Desired baud rate.

Integer Division ÷ 21

Reference Frequency = 19.44 MHz

UBIR = 0x000F

UBMR = 0x0014

Baud Rate = 925.7 kbit/s

NOTE

Observe that each value written to the registers is one less than the actual value.

Non-Integer Division

```
Reference Frequency = 16 MHz
Desired Baud Rate = 920 Kbits/s
```

$$\frac{UBMR + 1}{UBIR + 1} = \frac{RefFreq}{16 \times BaudRate} = \frac{16 \times 10^6}{16 \times 920 \times 10^3} = 1.087$$

```
Ratio = 1.087 = 1087 / 1000
UBIR = 999 (decimal)= 0x3E7
UBMR = 1086 (decimal)= 0x43E
Non-Integer Division
Reference Frequency = 25 MHz
Desired Baud Rate = 920 kbit/s
Ratio = 1.69837 = 625 / 368
UBIR = 367 (decimal)= 0x16F
UBMR = 624 (decimal)= 0x270
```

Non-Integer Division

```
Reference Frequency: 30 MHz
Desired Baud Rate = 115.2 kbit/s
Ratio = 16.276043 = 65153 / 4003
UBIR = 4002 (decimal) = 0x0FA2
UBMR = 65152 (decimal) = 0xFE80
```

17.2.4 Infrared Interface

17.2.4.1 Generalities-Infrared

The Infrared interface is selected when IREN (UCR1[7]) is set to 1.

The Infrared Interface is compatible with IrDA Serial Infrared Physical Layer Specification. In this specification, a "zero" is represented by a positive pulse, and a "one" is represented by no pulse (line remains low).

In the UART:

In TX: For each "zero" to be transmitted, a narrow positive pulse which is 3/16 of a bit time is generated. For each "one" to be transmitted no pulse is generated (output is low). External circuitry has to be provided to drive an Infrared LED.

In RX: When receiving, a narrow negative pulse is expected for each "zero" transmitted while no pulse is expected for each "one" transmitted (input is high).

NOTE

Rx part of IR block expects to receive an inverted signal compared to IrDA specification. Circuitry external to the IC transforms the Infrared signal to an electrical signal.

The IR interface has an edge triggered interrupt (IRINT). This interrupt validates a zero bit being received. This interrupt is enabled by writing a "one" to ENIRI bit.

The behavior of Infrared Interface is determined by 3 bits INVT (UCR3[1]), INVR (UCR4[9]) and IRSC (UCR4[5]).

17.2.4.2 Inverted Transmission and Reception bits (INVT & INVR)

The values of INVT and INVR depend of the IrDA transceiver connected on the TXD_IR and RXD_IR pins of the UART. If this transceiver is not inverting on both paths Tx and Rx, a Zero is represented by a positive pulse and a One is represented by no pulse (line remains low). In this case, the bit INVT must be set to 0 and the bit INVR must be set to 1 (because Rx IR block expects an inverted signal).

On the contrary user must set INVT=1 and INVR=0 if both paths of the transceiver are inverting, that is, a Zero is represented as a negative pulse and a One is represented by no pulse (line remains high). The transceiver can also be inverting on only one path (Tx or Rx), in this case INVT and INVR must be together equal to 1 or to 0, depending on which path is inverted.

17.2.4.3 InfraRed Special Case (IRSC) Bit

The value to apply to IRSC bit is based on 2 parameters: the baud rate and the Minimum Pulse Duration (MPD) of the transceiver.

According to IrDA Standard Specification, for SIR (Serial IR) baud rates from 2.4 Kbit/s to 115.2 Kbit/s this nominal pulse duration is equal to 3/16 of a bit duration (at the selected baud rate). But, for all the baud rates a Minimum Pulse Duration is also specified. According to IrDA Standard, a Zero is represented by a light pulse, so the IrDA transceiver can't emit a light pulse shorter than the MPD. For SIR, the MPD is constant and equal to 1.41 µs.

But user must take into account the electrical MPD associated with the transceiver on the receiver path. Typically this value is 2.0 µs, but for some manufacturers MPD can go down to 1.0 µs.

In order to understand the meaning of IRSC bit, one must understand how the RX path works in IrDA mode.

When the UART is in IrDA mode, a Zero is not only detected by the state of the RXD_IR line, but also with the duration of the pulse. This pulse duration can be measured with 2 different clocks. In this case, clock is selected with the IRSC bit.

- If IRSC = 0, the clock used is the BRM clock.
- If IRSC = 1, the clock used is the UART internal clock (UART clock after the divider (RFDIV)).

In normal operation, IRSC=0. This means that at any time, the user must ensure that the frequency of BRM_clock is high enough to measure the pulse. The pulse must last at least 2 BRM clock cycles. If this condition is not fulfilled, IRSC must be set to 1.

Let's examine two examples, for a Minimum Pulse Duration equal to the MPD from the IrDA SIR specification (i.e., 1.41 µs).

1: Calculation of BRM Clock Period (Clock Period < 1.41 µs)

The user wants to receive IrDA data at 115.2 Kbit/s. The UBIR and UBMR registers are set in order to create the BRM_clock with a frequency of 16*baud rate = 16 * 115.2K = 1.843 MHz. But at the same time, in order to correctly detect the pulse, the user must be sure that 2* BRM_clock period is lower than 1.41 µs. Lets check:

BRM_clock period = 1/1843000 = 542 ns

So 2*BRM_clock period = 1.09 µs < 1.41 µs. It is fine.

2: Calculation of BRM Clock Period (Clock Period > 1.41 µs)

This time the user wants to receive at 19.2 Kbit/s. So, the BRM_clock is set to 16*19200 = 307.2 kHz. Let's check if 2* BRM_clock period < 1.41 µs:

1. BRM_clock period =1/307200 = 3.25 µs

   So 2*BRM_clock period = 6.50 µs >> 1.41 µs. It doesn't work.

   So, in this case, the BRM clock can't be used to measure the pulse duration and the user must select the UART internal clock by setting IRSC =1.

NOTE

Like for Escape character detection, when IR Special Case is enabled (IRSC=1), the UART must measure a duration. In order to do that, the user must fill the ONEMS register. Refer to Escape Sequence Detection.

17.2.4.4 IrDA interrupt

Serial infrared mode (SIR) uses an edge triggered interrupt flag IRINT (USR2[8]). When INVR =0, detection of a falling edge on the RXD pin asserts the IRINT bit. When INVR=1, detection of a rising edge on the RXD pin asserts the IRINT bit. When IRINT and ENIRI bits are both asserted, the *interrupt_uart* interrupt is asserted. Clear the IRINT bit by writing 1 to it. Writing 0 to the IRINT bit has no effect.

17.2.4.5 Conclusion about IrDA

Before using the UART in IrDA, the baud rate limit must be calculated. This baud rate limit will inform the user if IRSC bit has to be set or not.

Let's determine this limit:

As already described, if IRSC = 0, the following condition must always be fulfilled

$$2 \times BRMClockPeriod < MinPulseDuration$$

Figure 17-18. Calculation of Baud Rate

So,

$$BRMClockFrequency > \frac{2}{MPD}$$

So, knowing BRM_clock frequency = 16 * Baud Rate, we get:

$$BaudRate > \frac{1}{8 \times MinPulseDuration}$$

So, the user needs to set IRSC = 0 when:

- If Minimum Pulse Duration = 2.5 us and Baud Rate > 50 Kbit/s.
- If Minimum Pulse Duration = 2.0 us and Baud Rate > 62.5 Kbit/s.
- If Minimum Pulse Duration = 1.41 us and Baud Rate > 88.6 Kbit/s.

NOTE

For baud rates lower than the limit, IRSC must be set to1.

17.2.4.6 Programming IrDA Interface

17.2.4.6.1 High Speed

As an example, the following sequence can be used to program the IrDA interface in order to send and receive characters at 115.2 Kbit/s.

Assumptions:

- Input UART clock = 90 MHz
- Internal clock divider = 3 (divide Input UART clock by 3)
- Baud rate = 115.2 Kbit/s
- IrDA transceiver is not inverting on both channels: for Tx and Rx, a Zero is represented by a positive pulse, and a One is represented by no pulse (line stays low).
- Interrupt: Sent to Arm platform when 1 char is received into the Rx FIFO (RDR)

Registers values and Programming orders:

```
UCR1 = 0x0085
UCR1[7] = IREN = 1: Enable IR interface
UCR1[0] = UARTEN = 1: Enable UART
UTS = 0x0000
UFCR = 0x0981
TXTL[5:0] = 0x02: Default value
RFDIV[2:0] = 0x3: Divide Input UART clock by 3 (resulting internal clock  is 30 MHz)
RXTL[5:0] = 0x01: Default value
UBIR = 0x0202
UBMR = 0x20BE Baud rate = 115.2 kbit/s with internal clock = 30 MHz
UCR2 = 0x4027
UCR2[14] = IRTS = 1: Ignore level of RTS input signal
UCR2[5] = WS = 1: Characters are 8-bit length
UCR2[2] = TXEN = 1: Enable Rx path
UCR2 [1] = RXEN = 1: Enable Tx path
UCR2[0] = SRST_B = 1: No software reset
UCR3 = 0x0000

         UCR4 = 0x8201
CTSTL[5:0] = 0x20: Default value
UCR4[9] = INVR = 1: Inverted Infrared Reception (because IrDA transceiver is not inverting)
UCR4[1] = DREN = 1: To enable RDR interrupt (sent when one char is received)
```

The UART is ready to send a character as soon as there is a write into UTXD register. And an interrupt will be sent to Arm platform when a character is received.

17.2.4.6.2 Low Speed

This time, we keep the same assumptions but the speed is now 9.6 Kbit/s. So, this baud rate is below the limit (even with a Min. Pulse Duration of 2.5 us) and thus IRSC must be set to 1.

Assumptions:

- Input UART clock = 90 MHz
- Internal clock divider = 3 (divide Input UART clock by 3)
- Baud rate = 9.6 Kbit/s
- IrDA transceiver is not inverting on both channels: for Tx and Rx, a Zero is represented by a positive pulse, and a One is represented by no pulse (line stays low).
- Interrupt: Sent to Arm platform when 1 char is received into the Rx FIFO (RDR).

Registers values and Programming orders:

```
UCR1 = 0x0085
UCR1[7] = IREN = 1: Enable IR interface
UCR1[0] = UARTEN = 1: Enable UART
UFCR = 0x0981
UFCR[15:10] = TXTL[5:0] = 0x02: Default value
RFDIV[2:0] = 0x3: Divide Input UART clock by 3 (resulting internal clock  is 30 MHz)
UFCR[5:0] = RXTL[5:0] = 0x01: Default value
UBIR = 0x00FF
UBMR = 0xC354 Baud rate = 9.6 kbit/s with internal clock = 30 MHz
UCR2 = 0x4027
UCR2[14] = IRTS = 1: Ignore level of RTS input signal
UCR2[5] = WS = 1: Characters are 8-bit length
UCR2[2] = TXEN = 1: Enable Rx path
UCR2 [1] = RXEN = 1: Enable Tx path
UCR2[0] = SRST_B = 1: No software reset
UCR3 = 0x0000
UCR3[1] = INVT = 0: Positive pulse represents 0.
UCR4 = 0x8221
UCR4[15:10] = CTSTL[5:0] = 0x20: Default value
UCR4[9] = INVR = 1: Inverted Infrared Reception (because IrDA transceiver is not inverting)
UCR4[5] = IRSC = 1: Because data rate is below the limit and thus the UART internal clock is
used to measure the pulse duration.
UCR4[1] = DREN = 1: To enable RDR interrupt (sent when one char is received)
```

The UART is now ready to send a character as soon as there is a write into UTXD register. An interrupt will be sent to Arm platform when a character is received.

17.2.5 9-bit RS-485 Mode

17.2.5.1 Generalities

The UART provides a 9-bit mode to facilitate multidrop (RS-485) network communication. To enable this mode, set MDEN bit in the UMCR register to 1. When 9-bit RS-485 mode is enabled, UART transmitter can transmit the ninth bit (9^{th} bit) set by TXB8, and UART receiver can differentiate between data frames (9^{th} bit = 0) and address frames (9^{th} bit = 1).

The CTS_B pin can be used to control RS-485 output driver outside the chip.

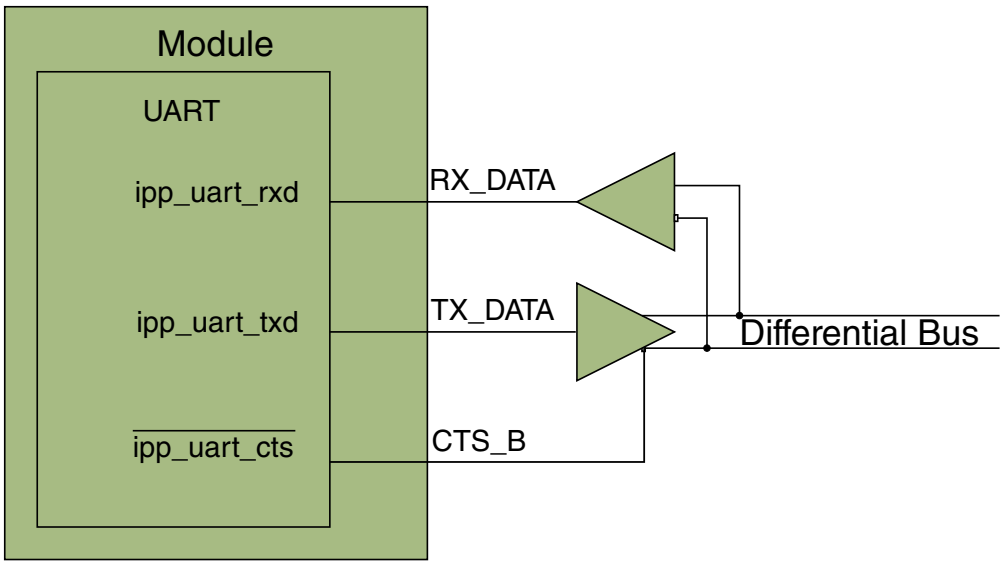


Figure 17-19. RS-485 driver connection (UART in DCE mode)

17.2.5.2 Transmit 9-bit RS-485 frames

To transmit 9-bit RS-485 frames, user need to enable parity (PREN=1) to enable trasmitting the ninth data bit, set 8-bit data word size (WS=1), and write TXB8 (UMCR[2]) as the 9th bit (bit [8]) to be transmitted (write '0' to TXB8 to transmit a data frame, write '1' to transmit a address frame). The other data bit [7:0] is written to TxFIFO by writing to the UTXD same as normal RS-232 operation.

17.2.5.3 Receive 9-bit RS-485 frames

To receive 9-bit RS-485 frames, user need to enable parity (PREN=1) to enable receiving the ninth data bit, set 8-bit data word size (WS=1). The receiver will save the 9-bit data to RxFIFO, and user should read the 9th databit (bit [8]) by reading the PRERR (URXD[10]) bit, and read data bit [7:0] by reading the RX_DATA (URXD[7:0]).

There are two slave address detect modes, normal detect mode and automatic detect mode, and can be selected by SLAM (UMCR[1]).

17.2.5.3.1 RS-485 Slave Address Normal Detect Mode

To enable Normal Detect mode, clear SLAM (UMCR[1] to 0). The receiver ignores all data frames (9th bit = 0) until an address frame is received (9th bit = 1). At that time, the slave address detected (SAD = USR1[3]) bit is asserted and the *interrupt_uart* interrupt is

generated (if SADEN = UMCR[3] = 1). The address byte and sebsequent bytes are all put into RxFIFO along with their 9th bit. The UART will also generate DMA request *dma_req_rx* when the RxFIFO reaches the selected threshold (controlled by RXTL) if receive ready DMA (RXDMAEN = UCR1[8]) request is enabled.

User should read the 9th databit (bit [8]) by reading the PRERR (URXD[10]) bit, and read data bit [7:0] by reading the RX_DATA (URXD[7:0]).

In this mode, once the UART has detected a 9th bit is equal to '1', it will always save the subsequent frames to RxFIFO. So the software must decide whether the address and data in RxFIFO are needed or not.

17.2.5.3.2 RS-485 Slave Address Automatic Detect Mode

To enable Automatic Detect Mode, set SLAM (UMCR[1]) to 1. The receiver tries to detect an address byte (frame 9th bit = 1) that matches the programmed SLADDR (UMCR[15:8]) character. If the received byte is a data or an address byte that does not match the programmed SLADDR character, the receiver will discard these data.

Once the UART receives a matching address byte, it will assert the slave address detected (SAD = USR1[3]) bit and the *interrupt_uart* interrupt will be generated (if SADEN = UMCR[3] = 1). The address byte and sebsequent bytes are all put into RxFIFO along with their 9th bit. If receive ready DMA(RXDMAEN = UCR1[8]) request is enabled, the UART will also generate DMA request *dma_req_rx* when the RxFIFO reaches the selected threshold (controlled by RXTL).

If another address byte is received and this address byte does not match SLADDR character, the receiver will discard the address byte and subsequent data byte. If the address byte again matches SLADDR character, the receiver will put this address byte and subsequent data byte in the RxFIFO along with their 9th bit.

User should read the 9th databit (bit [8]) by reading the PRERR (URXD[10]) bit, and read data bit [7:0] by reading the RX_DATA (URXD[7:0]).

See Initialization for 9-bit RS-485 programming guide.

17.2.6 Low Power Modes

These modes are controlled by the signals *doze_req* and *stop_req*. The control/status/data registers won't change when getting in/out of low power modes.

Table 17-9. UART Low Power State Operation

	Normal State (*doze_req* = 1'b0 & *stop_req* = 1'b0)	Doze State (*doze_req* = 1'b1)		Stop State (*stop_req* = 1'b1)
		DOZE bit = 0	DOZE bit = 1	
UART-Clock	ON	ON	ON	OFF
UART Serial / IrDA	ON	ON	OFF	OFF

17.2.6.1 UART Operation in System Doze Mode

While in Doze State (when *doze_req* input pin is set to 1'b1), the UART behavior depends on the DOZE (UCR1[1]) control bit.

While the DOZE bit is negated, the UART serial interface is enabled. While the system is in the Doze State, and the DOZE bit is asserted, the UART is disabled. If the Doze State is entered with the DOZE bit asserted while the UART serial interface was receiving or transmitting data, it will complete the receive/transmit of the current character and signal to the far-end transmitter/receiver to stop sending/receiving.

17.2.6.2 UART Operation in System Stop Mode

The internal baud rate clocks of the transmitter and receiver are gated off if the *stop_req* signal to UART is asserted. Even though the clocks at the input of the UART continue to run during system Stop mode, the UART will not do any transmission or reception.

The following UART interrupts wake the Arm platform processor from STOP mode:

- RTS (RTSD)
- IrDA Asynchronous WAKE (AIRINT)
- Asynchronous WAKE (AWAKE)
- RI (RIDELT in DTE mode only)
- DCD (DCDDELT in DTE mode only)
- DTR (DTRD in DCE mode only)
- DSR (DTRD in DTE mode only)

When an asynchronous WAKE (awake) interrupt exits the Arm platform from STOP mode, make sure that a dummy character is sent first because the first character may not be received correctly.

17.2.6.3 Power Saving Method in UART

The RXEN (UCR2[1]), TXEN (UCR2[2]) and UARTEN (UCR1[0]) bits are set by the user and provide software control of low-power modes.

Setting the UARTEN (UCR1[0]) bit to 0 shuts off the receiver and transmitter logic and the associated clocks.

If the UART is used only in transmit mode, UARTEN and TXEN must be set to 1. If the UART is used only in receive mode, UARTEN and RXEN must be set to 1. Setting TXEN or RXEN to 0 allows to save a lot of power.

17.2.7 UART Operation in System Debug State

The bit UTS [11] controls whether the UART will respond to the input signal *debug_req*, or whether it will continue to run as normal.

If the UART is programmed to respond to *debug_req*:

1. The UART will halt all operations upon detecting the *debug_req* input.
2. A transfer in progress, either to/from a core (using the IP Bus interface) or to/from an external device, will be completed before halting. This means a single byte/word transfer, not an entire FIFO. Reception of any further data from an external device will be disabled.
3. Internal registers will continue to be writable and readable using the IP Bus interface. A read will leave the contents unaffected.
4. The RX FIFO is affected in debug mode in the following way:
   - All writes into the RX FIFO are prevented.
   - The bit RXDBG (UTS[9]) is used to select the readability of the RX FIFO during debug mode:

     RXDBG = 0: hold the read pointer at the location it had upon entering debug mode, and URXD register returns only the data value at that location, no matter how many reads attempted.

RXDBG = 1, selectable at any time: Allow to read the characters received in Rx FIFO. It will not be possible to re-read previously read locations, nor will it be possible to readjust the read pointer to the value it had prior to entering debug mode.

17.2.8 Reset

This section describes how to reset the block and explains special requirements related to reset.

17.2.8.1 Hardware reset

All of registers, FIFOs, state machines and sequential elements can be reset to their initial values by hardware reset or power on reset.

17.2.8.2 Software reset

The status registers USR1 and USR2, BRM registers UBIR and UBMR, TxFIFO and RxFIFO, and transmitter and receiver state machines can be reset by software reset. Internal logic will keep the software reset asserted for about 4 *module_clock* cycles.

Programmer can follow the following software reset sequence:

1. Clear the SRST_B bit (UCR2[0])
2. Wait for software reset complete: poll SOFTRST bit (UTS[0]) until it is 0.
3. Re-program baud rate registers: Re-write UBIR and UBMR.

17.2.9 Transfer Error

The UART can generate a transfer error on the peripheral bus in the following cases:

- Core is writing into a read-only register.
- Core is accessing (read or write) an unused location within the assigned address space reserved to UART.
- Core is writing into UTXD register with transmit interface disabled (TXEN=0 or UARTEN=0)
- Core is reading URXD register with receive interface disabled (RXEN=0 or UARTEN=0)

17.2.10 Functional Timing

This section includes timing diagrams for functional signaling.

17.2.10.1 IrDA Mode

According to IrDA specification, the low speed (115.2Kbit/s and below) IR frame format is compatible with UART frame.

In this figure, an example data 0x65 is used.

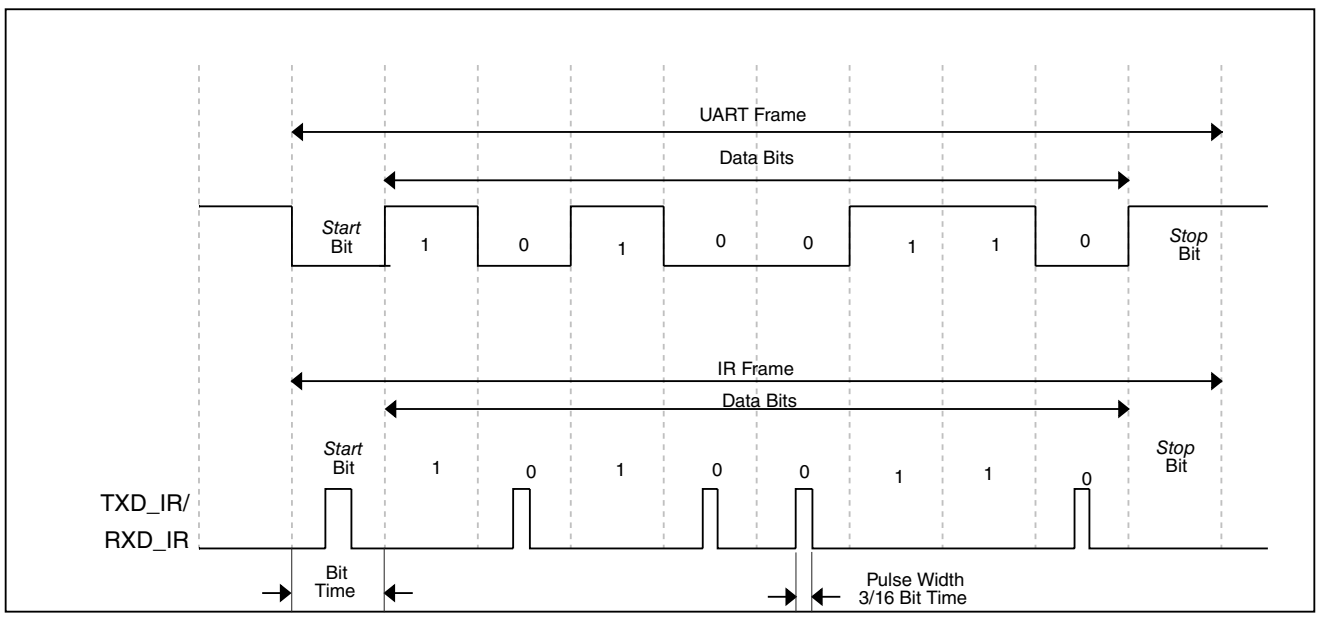


Figure 17-20. Timing diagram of Low Speed IR (<=115.2 Kbit/s) Data Line

17.2.11 External Signals

Tables that lists the conventions for representing signals and describing all UART signals that connect off-chip can be found here.

The chip-level IOMUX modifies the direction and routing of the UART signals based on whether the UART is operating in DCE mode (UARTn_UFCR[DCEDTE]=0) or DTE mode (UARTn_UFCR[DCEDTE]=1). The routing of the external signals to the UART module is shown in the figure below.

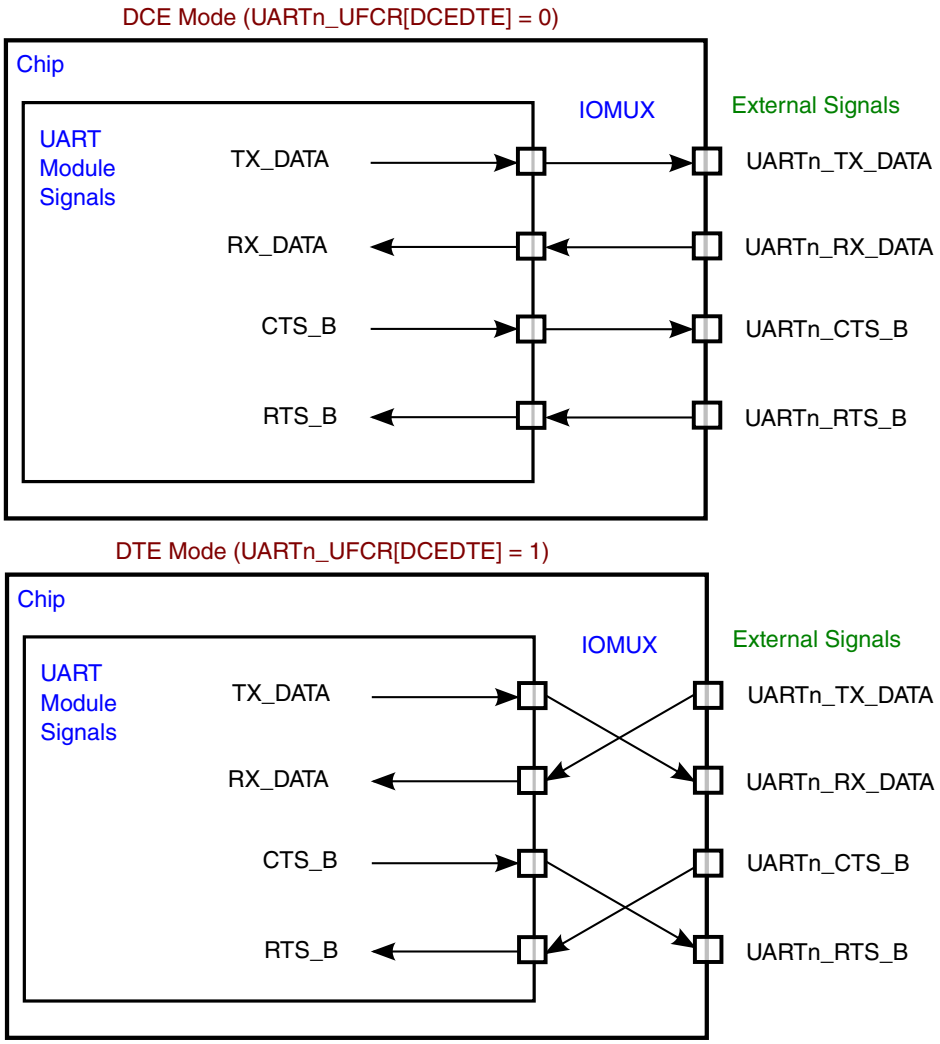


Figure 17-21. UART external signals to module signals routing with respect to DCE/DTE mode

The following tables describes the external signals of UART:

The user must configure the input path to the UART by properly configuring the DAISY bits in the IOMUXC_UARTn_RX_DATA_INPUT and the IOMUXC_UARTn_UART_RTS_B_SELECT_INPUT registers.

For IOMUXC_UARTn_UART_RTS_B_SELECT_INPUT[DAISY]:

- Configurations that select UARTn_RTS_B for the pad are only valid when UARTn_UFCR[DCEDTE]=0 (DCE mode)
- Configurations that select UARTn_CTS_B for the pad are only valid when UARTn_UFCR[DCEDTE]=1 (DTE mode)

For IOMUXC_UARTn_UART_RX_DATA_B_SELECT_INPUT[DAISY]:

- Configurations that select UARTn_RX_DATA for the pad are only valid when UARTn_UFCR[DCEDTE]=0 (DCE mode)
- Configurations that select UARTn_TX_DATA for the pad are only valid when UARTn_UFCR[DCEDTE]=1 (DTE mode)

17.2.11.1 Detailed Signal Descriptions

17.2.11.1.1 Interrupt Signals

17.2.11.1.1.1 *interrupt_uart* - UART Interrupt

Output interrupt request.

17.2.11.1.2 DMA Request Signals

17.2.11.1.2.1 *dma_req_rx* - Receiver DMA Request

Output DMA Request signal for receiver interface.

17.2.11.1.2.2 *dma_req_tx* - Transmitter DMA Request

Output DMA Request signal for transmitter interface. Set at 0 when TXDMAEN (UCR1[3]) is at 1 and TRDY (USR1[13]) is also at 1.

17.2.11.1.3 Special Signals

17.2.11.1.3.1 *stop_req* - Stop Mode

Input stop mode. Indicates to UART that Arm platform is going to enter in Stop Mode and clocks are going to stop running.

See Low Power Modes for more information about Stop Mode.

17.2.11.1.3.2 *doze_req* - Doze Mode

Input doze mode. Arm platform requests UART to switch in doze mode (power saving mode).

See Low Power Modes for more information about Doze Mode.

17.2.11.1.3.3 *debug_req* - Debug Mode

Input debug mode. Indicates UART it has to enter in debug mode.

See UART Operation in System Debug State, for more information about Debug Mode.

17.2.12 Initialization

17.2.12.1 Programming the UART in RS-232 mode

As an example, the following sequence can be used to program the UART in order to send and receive characters in RS-232 mode.

Assumptions:

- Input uart clock = 100 MHz
- Baud rate = 921.6 Kbps
- Data bits = 8 bits
- Parity = Even
- Stop bits = 1 bit
- Flow control = Hardware

Main program:

1. UCR1 = 0x0001

   Enable the UART.

2. UCR2 = 0x2127

   Set hardware flow control, data format and enable transmitter and receiver.

3. UCR3 = 0x0704

   Set UCR3[RXDMUXSEL] = 1.

4. UCR4 = 0x7C00

   Set CTS trigger level to 31,

5. UFCR = 0x089E

   Set internal clock divider = 5 (divide input uart clock by 5). So the reference clock is 100 MHz/5 = 20 MHz.

Set TXTL = 2 and RXTL = 30.

6. UBIR = 0x08FF
7. UBMR = 0x0C34

   In the above two steps, set baud rate to 921.6Kbps based on the 20MHz reference clock.

8. UCR1 = 0x2201

   Enable the TRDY and RRDY interrupts.

9. UMCR = 0x0000

   UMCR stay at default value 0x0000

Interrupt service routine for the transmitter:

- Write characters into UTXD

The TRDY interrupt will be automatically de-asserted when the data level of the TxFIFO exceeds the TXTL=2. Note: For the first time the interrupt may be de-asserted after 4 characters are written into the TxFIFO because of the shift register.

Interrupt service routine for the receiver:

- Read characters from URXD

The RRDY interrupt will be automatically de-asserted when the data level of the RxFIFO is below the RXTL=30.

17.2.12.2 Programming the UART in 9-bit RS-485 mode

As an example, the following sequence can be used to program the UART in order to send and receive frames in RS-485 mode.

Assumptions:

- Input uart clock = 100 MHz
- Baud rate = 5 Mbps

Main program:

1. UCR1 = 0x0001

   Enable the UART.

2. UCR2 = 0x4127

Set software flow control ($\overline{CTS}$ pin is controlled by UCR2[12]), enable parity(enable 9th bit rxd/txd), 8-bit word size , and enable transmitter and receiver.

3. UCR4 = 0x7C00

   Set CTS trigger level to 31,

4. UFCR = 0x0A9E

   Set RFDIV = 5 (divide input uart clock by 1), so the reference clock is 100 MHz. Set UART in DCE mode (RS-485 driver connection outside the chip is the same as Figure 17-19)

   Set TXTL = 2 and RXTL = 30.

5. UBIR = 0x0003
6. UBMR = 0x0004

   In the above two steps, set baud rate to 5 Mbps based on the 100 MHz reference clock.

7. UCR1 = 0x2001 when UART as a master ,

   or UCR1 = 0x0201 (or 0x0101) when UART as a slave.

   Enable TRDY interrupt when UART as a master, enable RRDY interrupt or DMA request when UART as a slave.

8. UMCR = 0xA50B

   Enable 9-bit RS-485 mode, enable SAD interrupt, set automatic slave address detect mode, set slave address is 0xA5.

Interrupt service routine for the transmitter:

- Transmit data: write its ninth bit (bit[8]) to UMCR[2], write its bit [7:0] into UTXD[7:0]

The TRDY interrupt will be automatically de-asserted when the data level of the TxFIFO exceeds the TXTL=2.

Note: For the first time the interrupt may be de-asserted after 4 characters are written into the TxFIFO because of the shift register.

Interrupt service routine for the receiver:

- Receive data: read its ninth bit (bit[8]) from URXD[10] , read its bit [7:0] from URXD[7:0].

Note: in RS-485 mode, URXD[10] bit is not the parity error, instead it holds the ninth bit (bit[8]) of the received data.

The SAD interrupt can not de-assert automatically, it needs MCU write 1 to USR1[3] to clear it . The RRDY interrupt or DMA request will be automatically de-asserted when the data level of the RxFIFO is below the RXTL=30.

17.2.13 References

- EIA/TIA-232-F Interface Standard

  http://www.eia.org, http://www.tiaonline.org/standards

- IrDA Standard

  http://www.irda.org

17.2.14 UART Memory Map/Register Definition

UART supports 8-bit, 16-bit and 32-bit accesses to 32-bit memory-mapped addresses. Any access to unmapped memory location will yield a transfer error.

All registers except the ONEMS described in this section are 16-bit registers. The ONEMS register is a 24-bit register.

- For 32-bit write accesses, the upper two bytes will not be taken into account.
- For 32-bit read accesses the upper two bytes will return 0.

The ONEMS register is expanded from 16 bits to 24 bits in order to support the high frequency of the BRM internal clock *ref_clk* (*module_clock* after divider). The ONEMS register can be accessed as 8 bits, 16 bits or 32 bits.

- For 32-bit write accesses, the most significant byte of the ONEMS will be discarded.
- For 32-bit read accesses, the most significant byte of the ONEMS will be read as 0.

UART memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3086_0000	UART Receiver Register (UART1_URXD)	32	R	0000_0000h	17.2.14.1/ 7412

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3086_0040	UART Transmitter Register (UART1_UTXD)	32	W	0000_0000h	17.2.14.2/ 7414
3086_0080	UART Control Register 1 (UART1_UCR1)	32	R/W	0000_0000h	17.2.14.3/ 7415
3086_0084	UART Control Register 2 (UART1_UCR2)	32	R/W	0000_0001h	17.2.14.4/ 7417
3086_0088	UART Control Register 3 (UART1_UCR3)	32	R/W	0000_0700h	17.2.14.5/ 7420
3086_008C	UART Control Register 4 (UART1_UCR4)	32	R/W	0000_8000h	17.2.14.6/ 7422
3086_0090	UART FIFO Control Register (UART1_UFCR)	32	R/W	0000_0801h	17.2.14.7/ 7424
3086_0094	UART Status Register 1 (UART1_USR1)	32	R/W	0000_2040h	17.2.14.8/ 7426
3086_0098	UART Status Register 2 (UART1_USR2)	32	R/W	0000_4028h	17.2.14.9/ 7429
3086_009C	UART Escape Character Register (UART1_UESC)	32	R/W	0000_002Bh	17.2.14.10/ 7431
3086_00A0	UART Escape Timer Register (UART1_UTIM)	32	R/W	0000_0000h	17.2.14.11/ 7431
3086_00A4	UART BRM Incremental Register (UART1_UBIR)	32	R/W	0000_0000h	17.2.14.12/ 7432
3086_00A8	UART BRM Modulator Register (UART1_UBMR)	32	R/W	0000_0000h	17.2.14.13/ 7432
3086_00AC	UART Baud Rate Count Register (UART1_UBRC)	32	R	0000_0004h	17.2.14.14/ 7433
3086_00B0	UART One Millisecond Register (UART1_ONEMS)	32	R/W	0000_0000h	17.2.14.15/ 7434
3086_00B4	UART Test Register (UART1_UTS)	32	R/W	0000_0060h	17.2.14.16/ 7435
3086_00B8	UART RS-485 Mode Control Register (UART1_UMCR)	32	R/W	0000_0000h	17.2.14.17/ 7436
3088_0000	UART Receiver Register (UART3_URXD)	32	R	0000_0000h	17.2.14.1/ 7412
3088_0040	UART Transmitter Register (UART3_UTXD)	32	W	0000_0000h	17.2.14.2/ 7414
3088_0080	UART Control Register 1 (UART3_UCR1)	32	R/W	0000_0000h	17.2.14.3/ 7415
3088_0084	UART Control Register 2 (UART3_UCR2)	32	R/W	0000_0001h	17.2.14.4/ 7417
3088_0088	UART Control Register 3 (UART3_UCR3)	32	R/W	0000_0700h	17.2.14.5/ 7420
3088_008C	UART Control Register 4 (UART3_UCR4)	32	R/W	0000_8000h	17.2.14.6/ 7422

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3088_0090	UART FIFO Control Register (UART3_UFCR)	32	R/W	0000_0801h	17.2.14.7/ 7424
3088_0094	UART Status Register 1 (UART3_USR1)	32	R/W	0000_2040h	17.2.14.8/ 7426
3088_0098	UART Status Register 2 (UART3_USR2)	32	R/W	0000_4028h	17.2.14.9/ 7429
3088_009C	UART Escape Character Register (UART3_UESC)	32	R/W	0000_002Bh	17.2.14.10/ 7431
3088_00A0	UART Escape Timer Register (UART3_UTIM)	32	R/W	0000_0000h	17.2.14.11/ 7431
3088_00A4	UART BRM Incremental Register (UART3_UBIR)	32	R/W	0000_0000h	17.2.14.12/ 7432
3088_00A8	UART BRM Modulator Register (UART3_UBMR)	32	R/W	0000_0000h	17.2.14.13/ 7432
3088_00AC	UART Baud Rate Count Register (UART3_UBRC)	32	R	0000_0004h	17.2.14.14/ 7433
3088_00B0	UART One Millisecond Register (UART3_ONEMS)	32	R/W	0000_0000h	17.2.14.15/ 7434
3088_00B4	UART Test Register (UART3_UTS)	32	R/W	0000_0060h	17.2.14.16/ 7435
3088_00B8	UART RS-485 Mode Control Register (UART3_UMCR)	32	R/W	0000_0000h	17.2.14.17/ 7436
3089_0000	UART Receiver Register (UART2_URXD)	32	R	0000_0000h	17.2.14.1/ 7412
3089_0040	UART Transmitter Register (UART2_UTXD)	32	W	0000_0000h	17.2.14.2/ 7414
3089_0080	UART Control Register 1 (UART2_UCR1)	32	R/W	0000_0000h	17.2.14.3/ 7415
3089_0084	UART Control Register 2 (UART2_UCR2)	32	R/W	0000_0001h	17.2.14.4/ 7417
3089_0088	UART Control Register 3 (UART2_UCR3)	32	R/W	0000_0700h	17.2.14.5/ 7420
3089_008C	UART Control Register 4 (UART2_UCR4)	32	R/W	0000_8000h	17.2.14.6/ 7422
3089_0090	UART FIFO Control Register (UART2_UFCR)	32	R/W	0000_0801h	17.2.14.7/ 7424
3089_0094	UART Status Register 1 (UART2_USR1)	32	R/W	0000_2040h	17.2.14.8/ 7426
3089_0098	UART Status Register 2 (UART2_USR2)	32	R/W	0000_4028h	17.2.14.9/ 7429
3089_009C	UART Escape Character Register (UART2_UESC)	32	R/W	0000_002Bh	17.2.14.10/ 7431
3089_00A0	UART Escape Timer Register (UART2_UTIM)	32	R/W	0000_0000h	17.2.14.11/ 7431

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3089_00A4	UART BRM Incremental Register (UART2_UBIR)	32	R/W	0000_0000h	17.2.14.12/ 7432
3089_00A8	UART BRM Modulator Register (UART2_UBMR)	32	R/W	0000_0000h	17.2.14.13/ 7432
3089_00AC	UART Baud Rate Count Register (UART2_UBRC)	32	R	0000_0004h	17.2.14.14/ 7433
3089_00B0	UART One Millisecond Register (UART2_ONEMS)	32	R/W	0000_0000h	17.2.14.15/ 7434
3089_00B4	UART Test Register (UART2_UTS)	32	R/W	0000_0060h	17.2.14.16/ 7435
3089_00B8	UART RS-485 Mode Control Register (UART2_UMCR)	32	R/W	0000_0000h	17.2.14.17/ 7436
30A6_0000	UART Receiver Register (UART4_URXD)	32	R	0000_0000h	17.2.14.1/ 7412
30A6_0040	UART Transmitter Register (UART4_UTXD)	32	W	0000_0000h	17.2.14.2/ 7414
30A6_0080	UART Control Register 1 (UART4_UCR1)	32	R/W	0000_0000h	17.2.14.3/ 7415
30A6_0084	UART Control Register 2 (UART4_UCR2)	32	R/W	0000_0001h	17.2.14.4/ 7417
30A6_0088	UART Control Register 3 (UART4_UCR3)	32	R/W	0000_0700h	17.2.14.5/ 7420
30A6_008C	UART Control Register 4 (UART4_UCR4)	32	R/W	0000_8000h	17.2.14.6/ 7422
30A6_0090	UART FIFO Control Register (UART4_UFCR)	32	R/W	0000_0801h	17.2.14.7/ 7424
30A6_0094	UART Status Register 1 (UART4_USR1)	32	R/W	0000_2040h	17.2.14.8/ 7426
30A6_0098	UART Status Register 2 (UART4_USR2)	32	R/W	0000_4028h	17.2.14.9/ 7429
30A6_009C	UART Escape Character Register (UART4_UESC)	32	R/W	0000_002Bh	17.2.14.10/ 7431
30A6_00A0	UART Escape Timer Register (UART4_UTIM)	32	R/W	0000_0000h	17.2.14.11/ 7431
30A6_00A4	UART BRM Incremental Register (UART4_UBIR)	32	R/W	0000_0000h	17.2.14.12/ 7432
30A6_00A8	UART BRM Modulator Register (UART4_UBMR)	32	R/W	0000_0000h	17.2.14.13/ 7432
30A6_00AC	UART Baud Rate Count Register (UART4_UBRC)	32	R	0000_0004h	17.2.14.14/ 7433
30A6_00B0	UART One Millisecond Register (UART4_ONEMS)	32	R/W	0000_0000h	17.2.14.15/ 7434
30A6_00B4	UART Test Register (UART4_UTS)	32	R/W	0000_0060h	17.2.14.16/ 7435

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30A6_00B8	UART RS-485 Mode Control Register (UART4_UMCR)	32	R/W	0000_0000h	17.2.14.17/ 7436

17.2.14.1 UART Receiver Register (UARTx_URXD)

NOTE

The UART will yield a transfer error on the peripheral bus when core is reading URXD register with receive interface disabled (RXEN=0 or UARTEN=0).

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHARRDY	ERR	OVRRUN	FRMERR	BRK	PRERR	Reserved		RX_DATA							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_URXD field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 CHARRDY	Character Ready. This read-only bit indicates an invalid read when the FIFO becomes empty and software tries to read the same old data. This bit should not be used for polling for data written to the RX FIFO. 0 Character in RX_DATA field and associated flags are invalid. 1 Character in RX_DATA field and associated flags valid and ready for reading.
14 ERR	**Error Detect.** Indicates whether the character present in the RX_DATA field has an error (OVRRUN, FRMERR, BRK or PRERR) status. The ERR bit is updated and valid for each received character. 0 No error status was detected 1 An error status was detected
13 OVRRUN	**Receiver Overrun.** This read-only bit, when HIGH, indicates that the corresponding character was stored in the last position (32nd) of the Rx FIFO. Even if a 33rd character has not been detected, this bit will be set to '1' for the 32nd character. 0 No RxFIFO overrun was detected 1 A RxFIFO overrun was detected
12 FRMERR	**Frame Error.** Indicates whether the current character had a framing error (a missing stop bit) and is possibly corrupted. FRMERR is updated for each character read from the RxFIFO. 0 The current character has no framing error 1 The current character has a framing error
11 BRK	**BREAK Detect.** Indicates whether the current character was detected as a BREAK character. The data bits and the stop bit are all 0. The FRMERR bit is set when BRK is set. When odd parity is selected, PRERR is also set when BRK is set. BRK is valid for each character read from the RxFIFO. 0 The current character is not a BREAK character 1 The current character is a BREAK character
10 PRERR	**In RS-485 mode, it holds the ninth data bit (bit [8]) of received 9-bit RS-485 data** **In RS232/IrDA mode, it is the Parity Error flag**. Indicates whether the current character was detected with a parity error and is possibly corrupted. PRERR is updated for each character read from the RxFIFO. When parity is disabled, PRERR always reads as 0. 0 = No parity error was detected for data in the RX_DATA field 1 = A parity error was detected for data in the RX_DATA field
9–8 -	This field is reserved. **Reserved**
RX_DATA	**Received Data**. Holds the received character. In 7-bit mode, the most significant bit (MSB) is forced to 0. In 8-bit mode, all bits are active.

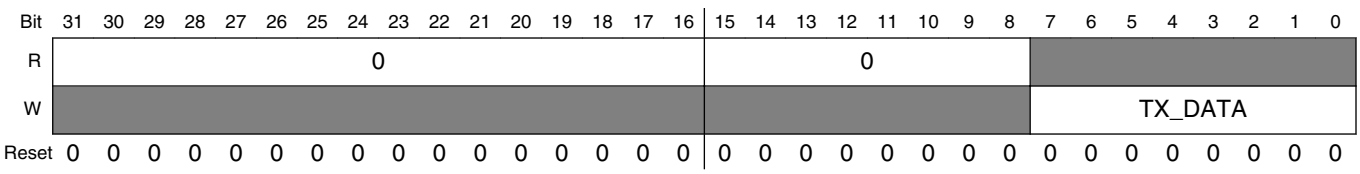
17.2.14.2 UART Transmitter Register (UARTx_UTXD)

NOTE

The UART will yield a transfer error on the peripheral bus when core is writing into UART_URXD register with transmit interface disabled (TXEN=0 or UARTEN=0).

Memory space between UART_URXD and UART_UTXD registers is reserved. Any read or write access to this space will be considered as an invalid access and yield a transfer error.

Address: Base address + 40h offset

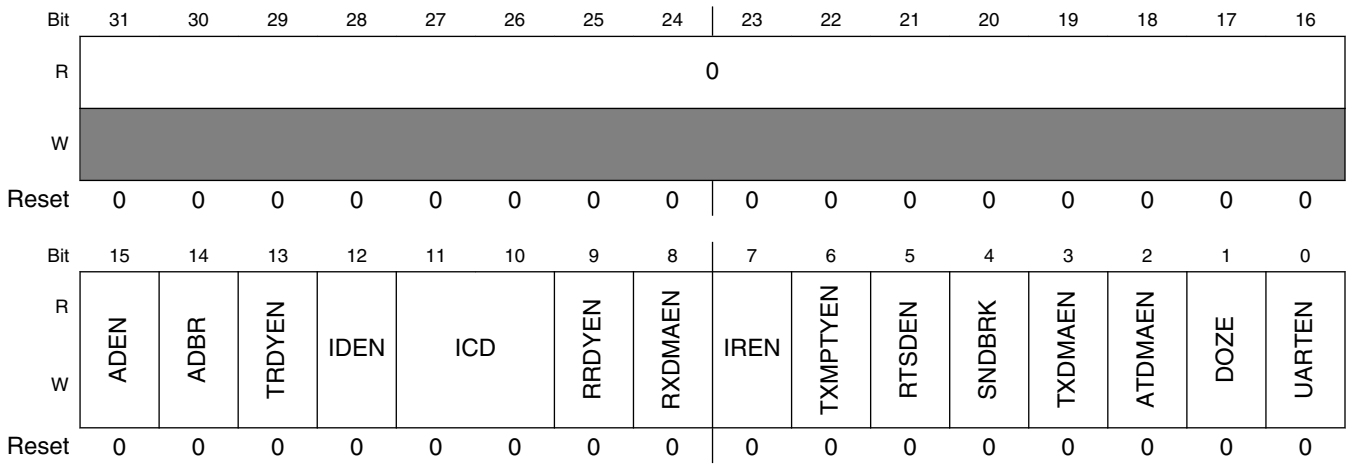


UARTx_UTXD field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 Reserved	This read-only field is reserved and always has the value 0.
TX_DATA	**Transmit Data**. Holds the parallel transmit data inputs. In 7-bit mode, D7 is ignored. In 8-bit mode, all bits are used. Data is transmitted least significant bit (LSB) first. A new character is transmitted when the TX_DATA field is written. The TX_DATA field must be written only when the TRDY bit is high to ensure that corrupted data is not sent.

17.2.14.3 UART Control Register 1 (UARTx_UCR1)

Address: Base address + 80h offset



UARTx_UCR1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 ADEN	**Automatic Baud Rate Detection Interrupt Enable**. Enables/Disables the automatic baud rate detect complete (ADET) bit to generate an interrupt (*interrupt_uart* = 0). 0 Disable the automatic baud rate detection interrupt 1 Enable the automatic baud rate detection interrupt
14 ADBR	**Automatic Detection of Baud Rate**. Enables/Disables automatic baud rate detection. When the ADBR bit is set and the ADET bit is cleared, the receiver detects the incoming baud rate automatically. The ADET flag is set when the receiver verifies that the incoming baud rate is detected properly by detecting an ASCII character "A" or "a" (0x41 or 0x61). 0 Disable automatic detection of baud rate 1 Enable automatic detection of baud rate
13 TRDYEN	**Transmitter Ready Interrupt Enable**. Enables/Disables the transmitter Ready Interrupt (TRDY) when the transmitter has one or more slots available in the TxFIFO. The fill level in the TXFIFO at which an interrupt is generated is controlled by TxTL bits. When TRDYEN is negated, the transmitter ready interrupt is disabled. **NOTE:** An interrupt will be issued as long as TRDYEN and TRDY are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TRDY interrupt. 0 Disable the transmitter ready interrupt 1 Enable the transmitter ready interrupt
12 IDEN	**Idle Condition Detected Interrupt Enable**. Enables/Disables the IDLE bit to generate an interrupt (*interrupt_uart* = 0). 0 Disable the IDLE interrupt 1 Enable the IDLE interrupt

Table continues on the next page...

UARTx_UCR1 field descriptions (continued)

Field	Description
11–10 ICD	**Idle Condition Detect**. Controls the number of frames RXD is allowed to be idle before an idle condition is reported. 00 Idle for more than 4 frames 01 Idle for more than 8 frames 10 Idle for more than 16 frames 11 Idle for more than 32 frames
9 RRDYEN	**Receiver Ready Interrupt Enable**. Enables/Disables the RRDY interrupt when the RxFIFO contains data. The fill level in the RxFIFO at which an interrupt is generated is controlled by the RXTL bits. When RRDYEN is negated, the receiver ready interrupt is disabled. 0 Disables the RRDY interrupt 1 Enables the RRDY interrupt
8 RXDMAEN	**Receive Ready DMA Enable**. Enables/Disables the receive DMA request *dma_req_rx* when the receiver has data in the RxFIFO. The fill level in the RxFIFO at which a DMA request is generated is controlled by the RXTL bits. When negated, the receive DMA request is disabled. 0 Disable DMA request 1 Enable DMA request
7 IREN	**Infrared Interface Enable**. Enables/Disables the IR interface. See the IR interface description in Infrared Interface, for more information. Note: MDEN(UMCR[0]) must be cleared to 0 when using IrDA interface. See Table 17-2 0 Disable the IR interface 1 Enable the IR interface
6 TXMPTYEN	**Transmitter Empty Interrupt Enable**. Enables/Disables the transmitter FIFO empty (TXFE) interrupt. *interrupt_uart*. When negated, the TXFE interrupt is disabled. **NOTE:** An interrupt will be issued as long as TXMPTYEN and TXFE are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TXFE interrupt. 0 Disable the transmitter FIFO empty interrupt 1 Enable the transmitter FIFO empty interrupt
5 RTSDEN	**RTS Delta Interrupt Enable**. Enables/Disables the RTSD interrupt. The current status of the RTS_B pin is read in the RTSS bit. 0 Disable RTSD interrupt 1 Enable RTSD interrupt
4 SNDBRK	**Send BREAK**. Forces the transmitter to send a BREAK character. The transmitter finishes sending the character in progress (if any) and sends BREAK characters until SNDBRK is reset. Because the transmitter samples SNDBRK after every bit is transmitted, it is important that SNDBRK is asserted high for a sufficient period of time to generate a valid BREAK. After the BREAK transmission completes, the UART transmits 2 mark bits. The user can continue to fill the TxFIFO and any characters remaining are transmitted when the BREAK is terminated. 0 Do not send a BREAK character 1 Send a BREAK character (continuous 0s)
3 TXDMAEN	**Transmitter Ready DMA Enable**. Enables/Disables the transmit DMA request *dma_req_tx* when the transmitter has one or more slots available in the TxFIFO. The fill level in the TxFIFO that generates the *dma_req_tx* is controlled by the TXTL bits.

Table continues on the next page...

UARTx_UCR1 field descriptions (continued)

Field	Description
	NOTE: A DMA request will be issued as long as TXDMAEN and TRDY are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the transmit DMA request. 0 Disable transmit DMA request 1 Enable transmit DMA request
2 ATDMAEN	**Aging DMA Timer Enable**. Enables/Disables the receive DMA request *dma_req_rx* for the aging timer interrupt (triggered with AGTIM flag in USR1[8]). 0 Disable AGTIM DMA request 1 Enable AGTIM DMA request
1 DOZE	**DOZE**. Determines the UART enable condition in the DOZE state. When *doze_req* input pin is at '1', (the Arm Platform executes a doze instruction and the system is placed in the Doze State), the DOZE bit affects operation of the UART. While in the Doze State, if this bit is asserted, the UART is disabled. See the description in Low Power Modes. 0 The UART is enabled when in DOZE state 1 The UART is disabled when in DOZE state
0 UARTEN	**UART Enable**. Enables/Disables the UART. If UARTEN is negated in the middle of a transmission, the transmitter stops and pulls the TXD line to a logic 1. UARTEN must be set to 1 before any access to UTXD and URXD registers, otherwise a transfer error is returned. This bit can be set to 1 along with other bits in this register. There is no restriction to the sequence of programing this bit and other control registers. 0 Disable the UART 1 Enable the UART

17.2.14.4 UART Control Register 2 (UARTx_UCR2)

Address: Base address + 84h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESCI	IRTS	CTSC	CTS	ESCEN	RTEC		PREN	PROE	STPB	WS	RTSEN	ATEN	TXEN	RXEN	SRST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

UARTx_UCR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 ESCI	**Escape Sequence Interrupt Enable**. Enables/Disables the ESCF bit to generate an interrupt. 0 Disable the escape sequence interrupt 1 Enable the escape sequence interrupt
14 IRTS	**Ignore RTS Pin**. Forces the RTS input signal presented to the transmitter to always be asserted (set to low), effectively ignoring the external pin. When in this mode, the RTS pin serves as a general purpose input. 0 Transmit only when the RTS pin is asserted 1 Ignore the RTS pin
13 CTSC	CTS **Pin Control**. Controls the operation of the CTS_B module output. When CTSC is asserted, the CTS_B module output is controlled by the receiver. When the RxFIFO is filled to the level of the programmed trigger level and the start bit of the overflowing character (TRIGGER LEVEL + 1) is validated, the CTS_B module output is negated to indicate to the far-end transmitter to stop transmitting. When the trigger level is programmed for less than 32, the receiver continues to receive data until the RxFIFO is full. When the CTSC bit is negated, the CTS_B module output is controlled by the CTS bit. On reset, because CTSC is cleared to 0, the CTS_B pin is controlled by the CTS bit, which again is cleared to 0 on reset. This means that on reset the CTS_B signal is negated. 0 The CTS_B pin is controlled by the CTS bit 1 The CTS_B pin is controlled by the receiver
12 CTS	**Clear to Send**. Controls the CTS_B pin when the CTSC bit is negated. CTS has no function when CTSC is asserted. 0 The CTS_B pin is high (inactive) 1 The CTS_B pin is low (active)
11 ESCEN	**Escape Enable**. Enables/Disables the escape sequence detection logic. 0 Disable escape sequence detection 1 Enable escape sequence detection
10–9 RTEC	**Request to Send Edge Control**. Selects the edge that triggers the RTS interrupt. This has no effect on the RTS delta interrupt. RTEC has an effect only when RTSEN = 1 (see Table 17-4). 00 Trigger interrupt on a rising edge 01 Trigger interrupt on a falling edge 1X Trigger interrupt on any edge
8 PREN	**Parity Enable**. Enables/Disables the parity generator in the transmitter and parity checker in the receiver. When PREN is asserted, the parity generator and checker are enabled, and disabled when PREN is negated. 0 Disable parity generator and checker 1 Enable parity generator and checker
7 PROE	**Parity Odd/Even**. Controls the sense of the parity generator and checker. When PROE is high, odd parity is generated and expected. When PROE is low, even parity is generated and expected. PROE has no function if PREN is low. 0 Even parity 1 Odd parity

Table continues on the next page...

UARTx_UCR2 field descriptions (continued)

Field	Description
6 STPB	**Stop**. Controls the number of stop bits after a character. When STPB is low, 1 stop bit is sent. When STPB is high, 2 stop bits are sent. STPB also affects the receiver. 0 The transmitter sends 1 stop bit. The receiver expects 1 or more stop bits. 1 The transmitter sends 2 stop bits. The receiver expects 2 or more stop bits.
5 WS	**Word Size**. Controls the character length. When WS is high, the transmitter and receiver are in 8-bit mode. When WS is low, they are in 7-bit mode. The transmitter ignores bit 7 and the receiver sets bit 7 to 0. WS can be changed in-between transmission (reception) of characters, however not when a transmission (reception) is in progress, in which case the length of the current character being transmitted (received) is unpredictable. 0 7-bit transmit and receive character length (not including START, STOP or PARITY bits) 1 8-bit transmit and receive character length (not including START, STOP or PARITY bits)
4 RTSEN	**Request to Send Interrupt Enable.** Controls the RTS edge sensitive interrupt. When RTSEN is asserted and the programmed edge is detected on the RTS_B pin (the RTSF bit is asserted), an interrupt will be generated on the *interrupt_uart* pin. (See Table 17-4.) 0 Disable request to send interrupt 1 Enable request to send interrupt
3 ATEN	**Aging Timer Enable**. This bit is used to enable the aging timer interrupt (triggered with AGTIM) 0 AGTIM interrupt disabled 1 AGTIM interrupt enabled
2 TXEN	**Transmitter Enable**. Enables/Disables the transmitter. When TXEN is negated the transmitter is disabled and idle. When the UARTEN and TXEN bits are set the transmitter is enabled. If TXEN is negated in the middle of a transmission, the UART disables the transmitter immediately, and starts marking 1s. The transmitter FIFO cannot be written when this bit is cleared. 0 Disable the transmitter 1 Enable the transmitter
1 RXEN	**Receiver Enable**. Enables/Disables the receiver. When the receiver is enabled, if the RXD input is already low, the receiver does not recognize BREAK characters, because it requires a valid 1-to-0 transition before it can accept any character. 0 Disable the receiver 1 Enable the receiver
0 SRST	**Software Reset**. Once the software writes 0 to SRST_B, the software reset remains active for 4 *module_clock* cycles before the hardware deasserts SRST_B. The software can only write 0 to SRST_B. Writing 1 to SRST_B is ignored. 0 Reset the transmit and receive state machines, all FIFOs and register USR1, USR2, UBIR, UBMR, UBRC , URXD, UTXD and UTS[6-3]. 1 No reset

17.2.14.5 UART Control Register 3 (UARTx_UCR3)

Address: Base address + 88h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DPEC		DTREN	PARERREN	FRAERREN	DSR	DCD	RI	ADNIMP	RXDSEN	AIRINTEN	AWAKEN	DTRDEN	RXDMUXSEL	INVT	ACIEN
W																
Reset	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

UARTx_UCR3 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–14 DPEC	This bit is not used in this chip.
13 DTREN	This bit is not used in this chip.
12 PARERREN	**Parity Error Interrupt Enable. Enables/Disables the interrupt. When asserted, PARERREN causes the PARITYERR bit to generate an interrupt.** 0 Disable the parity error interrupt 1 Enable the parity error interrupt
11 FRAERREN	**Frame Error Interrupt Enable. Enables/Disables the interrupt. When asserted, FRAERREN causes the FRAMERR bit to generate an interrupt.** 0 Disable the frame error interrupt 1 Enable the frame error interrupt
10 DSR	This bit is not used in this chip.
9 DCD	This bit is not used in this chip.
8 RI	This bit is not used in this chip.
7 ADNIMP	**Autobaud Detection Not Improved-. Disables new features of autobaud detection (See Baud Rate Automatic Detection Protocol, for more details).** 0 Autobaud detection new features selected 1 Keep old autobaud detection mechanism

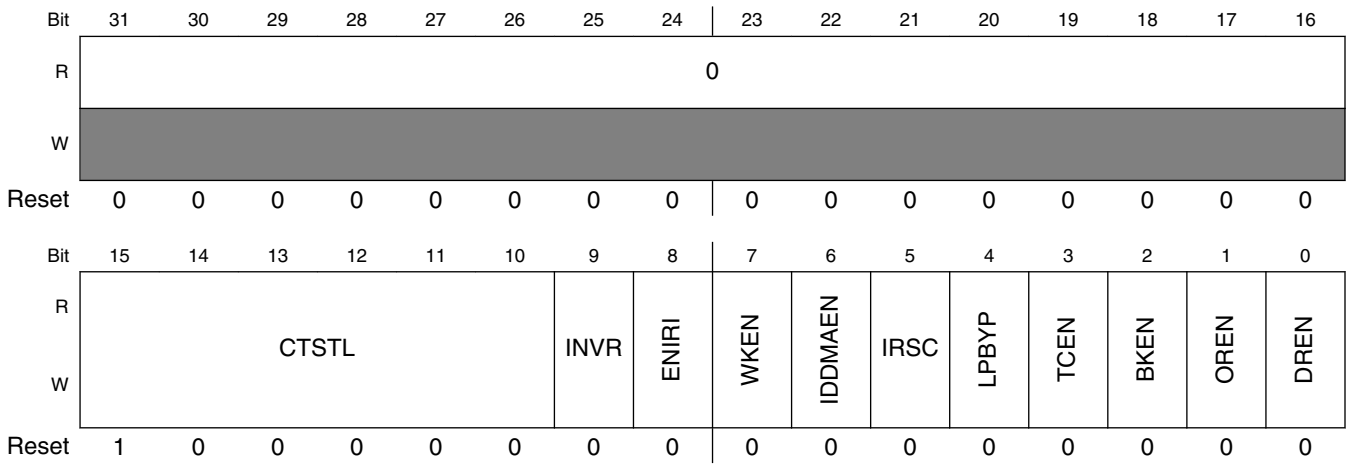
Table continues on the next page...

UARTx_UCR3 field descriptions (continued)

Field	Description
6 RXDSEN	Receive Status Interrupt Enable. Controls the receive status interrupt (*interrupt_uart*). When this bit is enabled and RXDS status bit is set, the interrupt *interrupt_uart* will be generated. 0 Disable the RXDS interrupt 1 Enable the RXDS interrupt
5 AIRINTEN	Asynchronous IR WAKE Interrupt Enable. Controls the asynchronous IR WAKE interrupt. An interrupt is generated when AIRINTEN is asserted and a pulse is detected on the RXD pin. 0 Disable the AIRINT interrupt 1 Enable the AIRINT interrupt
4 AWAKEN	Asynchronous WAKE Interrupt Enable. Controls the asynchronous WAKE interrupt. An interrupt is generated when AWAKEN is asserted and a falling edge is detected on the RXD pin. 0 Disable the AWAKE interrupt 1 Enable the AWAKE interrupt
3 DTRDEN	This bit is not used in this chip.
2 RXDMUXSEL	RXD Muxed Input Selected. Selects proper input pins for serial and Infrared input signal. **NOTE:** In this chip, UARTs are used in MUXED mode, so that this bit should always be set.
1 INVT	Invert TXD output in RS-232/RS-485 mode, set TXD active level in IrDA mode. In RS232/RS-485 mode(UMCR[0] = 1), if this bit is set to 1, the TXD output is inverted before transmitted. In **IrDA mode**, when INVT is cleared, the infrared logic block transmits a positive IR 3/16 pulse for all 0s and 0s are transmitted for 1s. When INVT is set (INVT = 1), the infrared logic block transmits an active low or negative infrared 3/16 pulse for all 0s and 1s are transmitted for 1s. 0 **TXD is not inverted** 1 **TXD is inverted** 0 **TXD** Active low transmission 1 **TXD** Active high transmission
0 ACIEN	**Autobaud Counter Interrupt Enable.** This bit is used to enable the autobaud counter stopped interrupt (triggered with ACST (USR2[11]). 0 ACST interrupt disabled 1 ACST interrupt enabled

17.2.14.6 UART Control Register 4 (UARTx_UCR4)

Address: Base address + 8Ch offset



UARTx_UCR4 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–10 CTSTL	**CTS Trigger Level**. Controls the threshold at which the CTS_B pin is deasserted by the RxFIFO. After the trigger level is reached and the CTS_B pin is deasserted, the RxFIFO continues to receive data until it is full. The CTSTL bits are encoded as shown in the Settings column. Settings 0 to 32 are in use. All other settings are Reserved. 000000 0 characters received 000001 1 characters in the RxFIFO ... — ... — 100000 32 characters in the RxFIFO (maximum)
9 INVR	**Invert RXD input in RS-232/RS-485 Mode, d**etermine **RXD input** logic level being sampled in **In IrDA mode.** **In RS232/RS-485 Mode(UMCR[0] = 1), if this bit is set to 1, the RXD input is inverted before sampled.** **In IrDA mode**,when cleared, the infrared logic block expects an active low or negative IR 3/16 pulse for 0s and 1s are expected for 1s. When INVR is set (INVR 1), the infrared logic block expects an active high or positive IR 3/16 pulse for 0s and 0s are expected for 1s. 0 **RXD input is not inverted** 1 **RXD input is inverted** 0 **RXD** active low detection 1 **RXD** active high detection
8 ENIRI	**Serial Infrared Interrupt Enable**. Enables/Disables the serial infrared interrupt. 0 Serial infrared Interrupt disabled 1 Serial infrared Interrupt enabled

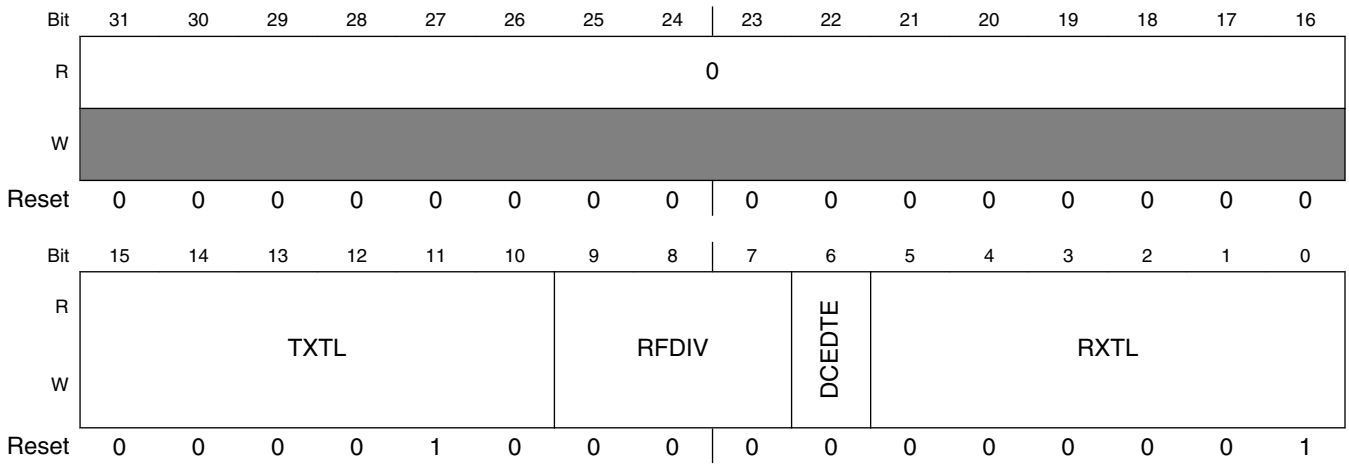
Table continues on the next page...

UARTx_UCR4 field descriptions (continued)

Field	Description
7 WKEN	**WAKE Interrupt Enable**. Enables/Disables the WAKE bit to generate an interrupt. The WAKE bit is set at the detection of a start bit by the receiver. 0 Disable the WAKE interrupt 1 Enable the WAKE interrupt
6 IDDMAEN	**DMA IDLE Condition Detected Interrupt Enable** Enables/Disables the receive DMA request *dma_req_rx* for the IDLE interrupt (triggered with IDLE flag in USR2[12]). 0 DMA IDLE interrupt disabled 1 DMA IDLE interrupt enabled
5 IRSC	**IR Special Case**. Selects the clock for the vote logic. When set, IRSC switches the vote logic clock from the sampling clock to the UART reference clock. The IR pulses are counted a predetermined amount of time depending on the reference frequency. See InfraRed Special Case (IRSC) Bit. 0 The vote logic uses the sampling clock (16x baud rate) for normal operation 1 The vote logic uses the UART reference clock
4 LPBYP	**Low Power Bypass**. Allows to bypass the low power new features in UART. To use during debug phase. 0 Low power features enabled 1 Low power features disabled
3 TCEN	**TransmitComplete Interrupt Enable**. Enables/Disables the TXDC bit to generate an interrupt (*interrupt_uart* = 0) **NOTE:** An interrupt will be issued as long as TCEN and TXDC are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TXDC interrupt. 0 Disable TXDC interrupt 1 Enable TXDC interrupt
2 BKEN	**BREAK Condition Detected Interrupt Enable**. Enables/Disables the BRCD bit to generate an interrupt. 0 Disable the BRCD interrupt 1 Enable the BRCD interrupt
1 OREN	**Receiver Overrun Interrupt Enable**. Enables/Disables the ORE bit to generate an interrupt. 0 Disable ORE interrupt 1 Enable ORE interrupt
0 DREN	**Receive Data Ready Interrupt Enable**. Enables/Disables the RDR bit to generate an interrupt. 0 Disable RDR interrupt 1 Enable RDR interrupt

17.2.14.7 UART FIFO Control Register (UARTx_UFCR)

Address: Base address + 90h offset



UARTx_UFCR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–10 TXTL	**Transmitter Trigger Level**. Controls the threshold at which a maskable interrupt is generated by the TxFIFO. A maskable interrupt is generated whenever the data level in the TxFIFO falls below the selected threshold. The bits are encoded as shown in the Settings column. Settings 0 to 32 are in use. All other settings are Reserved. 000000 Reserved 000001 Reserved 000010 TxFIFO has 2 or fewer characters ... — ... — 011111 TxFIFO has 31 or fewer characters 100000 TxFIFO has 32 characters (maximum)
9–7 RFDIV	Reference Frequency Divider. Controls the divide ratio for the reference clock. The input clock is *module_clock*. The output from the divider is *ref_clk* which is used by BRM to create the 16x baud rate oversampling clock (*brm_clk*). 000 Divide input clock by 6 001 Divide input clock by 5 010 Divide input clock by 4 011 Divide input clock by 3 100 Divide input clock by 2 101 Divide input clock by 1 110 Divide input clock by 7 111 Reserved
6 DCEDTE	**DCE/DTE mode select**. Select UART as data communication equipment (DCE mode) or as data terminal equipment (DTE mode).

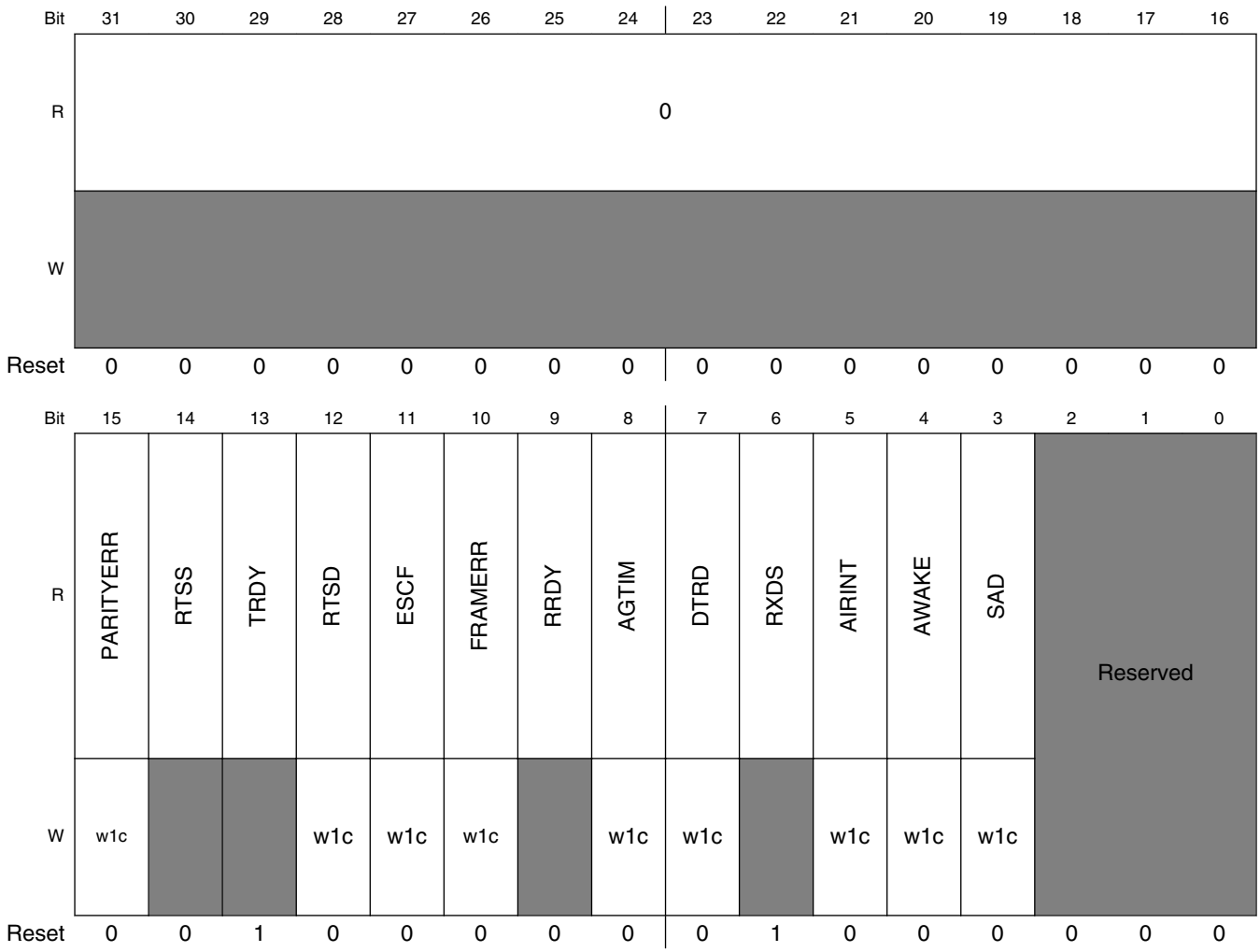
Table continues on the next page...

UARTx_UFCR field descriptions (continued)

Field	Description
	0 DCE mode selected 1 DTE mode selected
RXTL	**Receiver Trigger Level**. Controls the threshold at which a maskable interrupt is generated by the RxFIFO. A maskable interrupt is generated whenever the data level in the RxFIFO reaches the selected threshold. The RXTL bits are encoded as shown in the Settings column. Setting 0 to 32 are in use. All other settings are Reserved. 000000 0 characters received 000001 RxFIFO has 1 character ... — ... — 011111 RxFIFO has 31 characters 100000 RxFIFO has 32 characters (maximum)

17.2.14.8 UART Status Register 1 (UARTx_USR1)

Address: Base address + 94h offset



UARTx_USR1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 PARITYERR	**Parity Error Interrupt Flag**. Indicates a parity error is detected. PARITYERR is cleared by writing 1 to it. Writing 0 to PARITYERR has no effect. When parity is disabled, PARITYERR always reads 0. At reset, PARITYERR is set to 0. 0 No parity error detected 1 Parity error detected (write 1 to clear)
14 RTSS	RTS_B **Pin Status**. Indicates the current status of the RTS_B pin. A "snapshot" of RTS_B is taken immediately before RTSS is presented to the data bus. RTSS cannot be cleared because all writes to RTSS are ignored. At reset, RTSS is set to 0.

Table continues on the next page...

UARTx_USR1 field descriptions (continued)

Field	Description
	0 The RTS_B module input is high (inactive) 1 The RTS_B module input is low (active)
13 TRDY	**Transmitter Ready Interrupt / DMA Flag**. Indicates that the TxFIFO emptied below its target threshold and requires data. TRDY is automatically cleared when the data level in the TxFIFO exceeds the threshold set by TXTL bits. At reset, TRDY is set to 1. 0 The transmitter does not require data 1 The transmitter requires data (interrupt posted)
12 RTSD	**RTS Delta.** Indicates whether the RTS_B pin changed state. It (RTSD) generates a maskable interrupt. When in STOP mode, RTS assertion sets RTSD and can be used to wake the processor. The current state of the RTS_B pin is available on the RTSS bit. Clear RTSD by writing 1 to it. Writing 0 to RTSD has no effect. At reset, RTSD is set to 0. 0 RTS_B pin did not change state since last cleared 1 RTS_B pin changed state (write 1 to clear)
11 ESCF	**Escape Sequence Interrupt Flag**. Indicates if an escape sequence was detected. ESCF is asserted when the ESCEN bit is set and an escape sequence is detected in the RxFIFO. Clear ESCF by writing 1 to it. Writing 0 to ESCF has no effect. 0 No escape sequence detected 1 Escape sequence detected (write 1 to clear).
10 FRAMERR	**Frame Error Interrupt Flag**. Indicates that a frame error is detected. The *interrupt_uart* interrupt will be generated if a frame error is detected and the interrupt is enabled. Clear FRAMERR by writing 1 to it. Writing 0 to FRAMERR has no effect. 0 No frame error detected 1 Frame error detected (write 1 to clear)
9 RRDY	**Receiver Ready Interrupt / DMA Flag**. Indicates that the RxFIFO data level is above the threshold set by the RXTL bits. (See the RXTL bits description in UART FIFO Control Register (UART_UFCR) for setting the interrupt threshold.) When asserted, RRDY generates a maskable interrupt or DMA request. RRDY is automatically cleared when data level in the RxFIFO goes below the set threshold level. At reset, RRDY is set to 0. 0 No character ready 1 Character(s) ready (interrupt posted)
8 AGTIM	Ageing Timer Interrupt Flag. Indicates that data in the RxFIFO has been idle for a time of 8 character lengths (where a character length consists of 7 or 8 bits, depending on the setting of the WS bit in UCR2, with the bit time corresponding to the baud rate setting) and FIFO data level is less than RxFIFO threshold level (RXTL in the UFCR). Clear by writing a 1 to it. 0 AGTIM is not active 1 AGTIM is active (write 1 to clear)
7 DTRD	This bit is not used in this chip.
6 RXDS	Receiver IDLE Interrupt Flag. Indicates that the receiver state machine is in an IDLE state, the next state is IDLE, and the receive pin is high. RXDS is automatically cleared when a character is received. RXDS is active only when the receiver is enabled. 0 Receive in progress 1 Receiver is IDLE

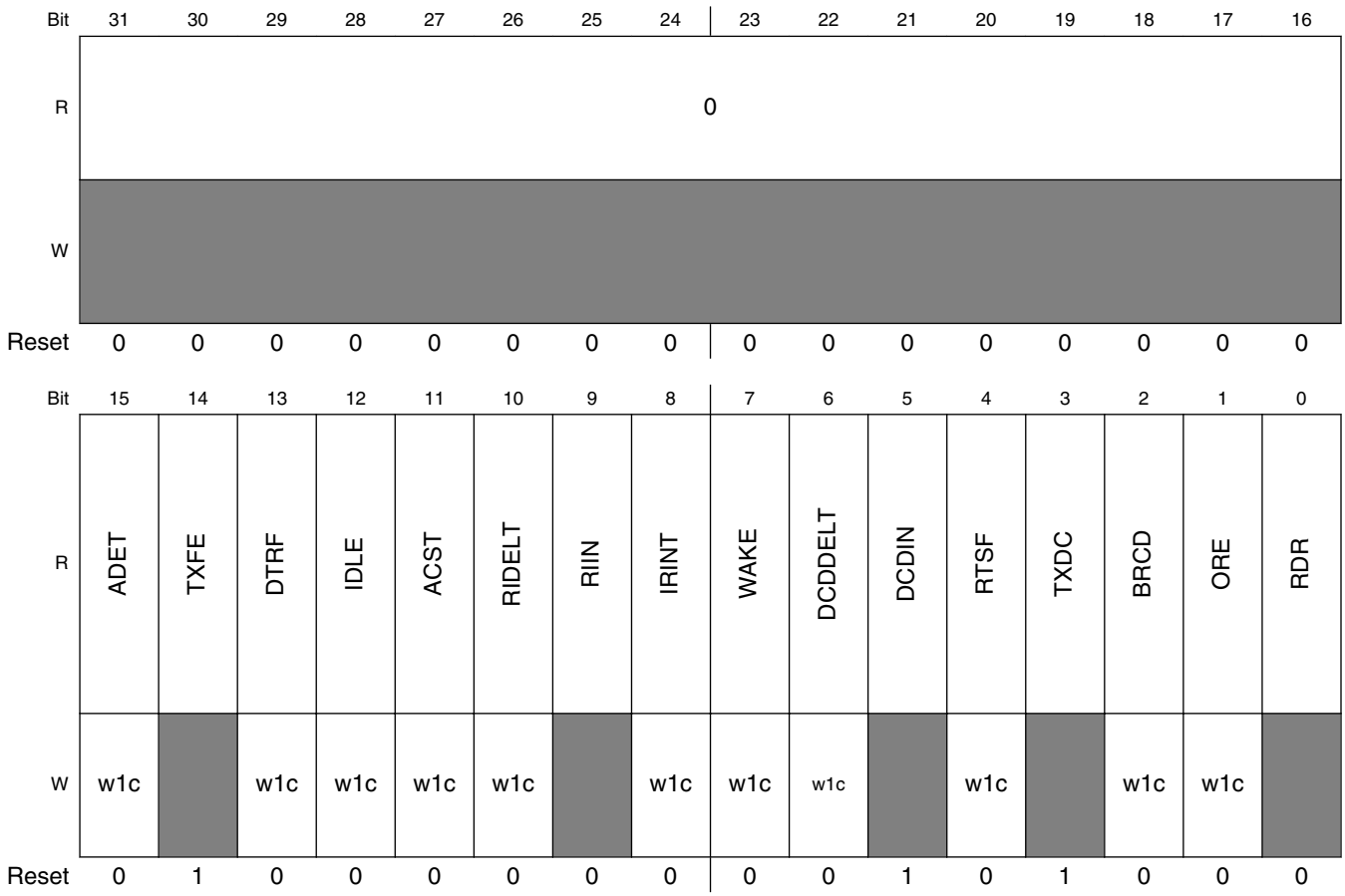
Table continues on the next page...

UARTx_USR1 field descriptions (continued)

Field	Description
5 AIRINT	Asynchronous IR WAKE Interrupt Flag. Indicates that the IR WAKE pulse was detected on the RXD pin. Clear AIRINT by writing 1 to it. Writing 0 to AIRINT has no effect. 0 No pulse was detected on the RXD IrDA pin 1 A pulse was detected on the RXD IrDA pin
4 AWAKE	Asynchronous WAKE Interrupt Flag. Indicates that a falling edge was detected on the RXD pin. Clear AWAKE by writing 1 to it. Writing 0 to AWAKE has no effect. 0 No falling edge was detected on the RXD Serial pin 1 A falling edge was detected on the RXD Serial pin
3 SAD	RS-485 Slave Address Detected Interrupt Flag. Indicates if RS-485 Slave Address was detected . SAD was asserted in RS-485 mode when the SADEN bit is set and Slave Address is detected in RxFIFO (in Nomal Address Detect Mode, the 9th data bit = 1; in Automatic Address Detect Mode, the received charater matches the programmed SLADDR). 0 No slave address detected 1 Slave address detected
-	This field is reserved. Reserved

17.2.14.9 UART Status Register 2 (UARTx_USR2)

Address: Base address + 98h offset



UARTx_USR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 ADET	**Automatic Baud Rate Detect Complete**. Indicates that an "A" or "a" was received and that the receiver detected and verified the incoming baud rate. Clear ADET by writing 1 to it. Writing 0 to ADET has no effect. 0 ASCII "A" or "a" was not received 1 ASCII "A" or "a" was received (write 1 to clear)
14 TXFE	**Transmit Buffer FIFO Empty**. Indicates that the transmit buffer (TxFIFO) is empty. TXFE is cleared automatically when data is written to the TxFIFO. Even though TXFE is high, the transmission might still be in progress. 0 The transmit buffer (TxFIFO) is not empty 1 The transmit buffer (TxFIFO) is empty
13 DTRF	This bit is not used in this chip.

Table continues on the next page...

UARTx_USR2 field descriptions (continued)

Field	Description
12 IDLE	**Idle Condition**. Indicates that an idle condition has existed for more than a programmed amount frame (see Idle Line Detect. An interrupt can be generated by this IDLE bit if IDEN (UCR1[12]) is enabled. IDLE is cleared by writing 1 to it. Writing 0 to IDLE has no effect. 0 No idle condition detected 1 Idle condition detected (write 1 to clear)
11 ACST	**Autobaud Counter Stopped**. In autobaud detection (ADBR=1), indicates the counter which determines the baud rate was running and is now stopped. This means either START bit is finished (if ADNIMP=1), or Bit 0 is finished (if ADNIMP=0). See New Autobaud Counter Stopped bit and Interrupt, for more details. An interrupt can be flagged on *interrupt_uart* if ACIEN=1. 0 Measurement of bit length not finished (in autobaud) 1 Measurement of bit length finished (in autobaud). (write 1 to clear)
10 RIDELT	This bit is not used in this chip.
9 RIIN	This bit is not used in this chip.
8 IRINT	**Serial Infrared Interrupt Flag.** When an edge is detected on the RXD pin during SIR Mode, this flag will be asserted. This flag can cause an interrupt which can be masked using the control bit ENIRI: UCR4 [8]. 0 no edge detected 1 valid edge detected (write 1 to clear)
7 WAKE	**Wake**. Indicates the start bit is detected. WAKE can generate an interrupt that can be masked using the WKEN bit. Clear WAKE by writing 1 to it. Writing 0 to WAKE has no effect. 0 start bit not detected 1 start bit detected (write 1 to clear)
6 DCDDELT	This bit is not used in this chip.
5 DCDIN	This bit is not used in this chip.
4 RTSF	**RTS Edge Triggered Interrupt Flag. Indicates if** a programmed edge is detected on the RTS_B pin. The RTEC bits select the edge that generates an interrupt (see Table 17-4). RTSF can generate an interrupt that can be masked using the RTSEN bit. Clear RTSF by writing 1 to it. Writing 0 to RTSF has no effect. 0 Programmed edge not detected on RTS_B 1 Programmed edge detected on RTS_B (write 1 to clear)
3 TXDC	**Transmitter Complete**. Indicates that the transmit buffer (TxFIFO) and Shift Register is empty; therefore the transmission is complete. TXDC is cleared automatically when data is written to the TxFIFO. 0 Transmit is incomplete 1 Transmit is complete
2 BRCD	**BREAK Condition Detected**. Indicates that a BREAK condition was detected by the receiver. Clear BRCD by writing 1 to it. Writing 0 to BRCD has no effect. 0 No BREAK condition was detected 1 A BREAK condition was detected (write 1 to clear)
1 ORE	**Overrun Error**. When set to 1, ORE indicates that the receive buffer (RxFIFO) was full (32 chars inside), and a 33rd character has been fully received. This 33rd character has been discarded. Clear ORE by writing 1 to it. Writing 0 to ORE has no effect.

Table continues on the next page...

UARTx_USR2 field descriptions (continued)

Field	Description
	0 No overrun error 1 Overrun error (write 1 to clear)
0 RDR	**Receive Data Ready**-Indicates that at least 1 character is received and written to the RxFIFO. If the URXD register is read and there is only 1 character in the RxFIFO, RDR is automatically cleared. 0 No receive data ready 1 Receive data ready

17.2.14.10 UART Escape Character Register (UARTx_UESC)

Address: Base address + 9Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								ESC_CHAR							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1

UARTx_UESC field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
ESC_CHAR	**UART Escape Character**. Holds the selected escape character that all received characters are compared against to detect an escape sequence.

17.2.14.11 UART Escape Timer Register (UARTx_UTIM)

Address: Base address + A0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																				TIM											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UTIM field descriptions

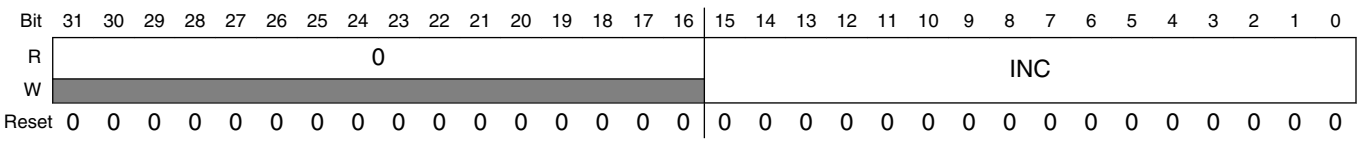
Field	Description
31–12 Reserved	This read-only field is reserved and always has the value 0.
TIM	UART Escape Timer. Holds the maximum time interval (in ms) allowed between escape characters. The escape timer register is programmable in intervals of 2 ms. See Escape Sequence Detection and Table 17-8 for more information on the UART escape sequence detection. Reset value 0x000 = 2 ms up to 0xFFF = 8.192 s.

17.2.14.12 UART BRM Incremental Register (UARTx_UBIR)

This register can be written by both software and hardware. When enabling the automatic baud rate detection feature hardware can write 0x000F value into the UBIR after finishing detecting baud rate. Hardware has higher priority when both software and hardware try to write it at the same cycle[3].

Please note software reset will reset the register to its reset value.

Address: Base address + A4h offset



UARTx_UBIR field descriptions

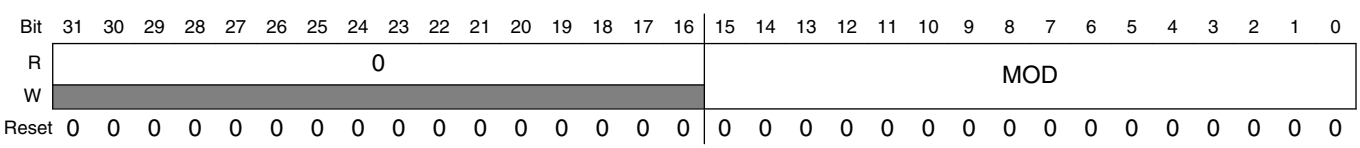
Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
INC	Incremental Numerator. Holds the numerator value minus one of the BRM ratio (see Binary Rate Multiplier (BRM)). The UBIR register MUST be updated before the UBMR register for the baud rate to be updated correctly. If only one register is written to by software, the BRM will ignore this data until the other register is written to by software. Updating this field using byte accesses is not recommended and is undefined.

17.2.14.13 UART BRM Modulator Register (UARTx_UBMR)

This register can be written by both software and hardware. When enabling the automatic baud rate detection feature hardware can write a proper value into the UBMR based on detected baud rate. Hardware has higher priority when both software and hardware try to write it at the same cycle[4].

Please note software reset will reset the register to its reset value.

Address: Base address + A8h offset



3. Note: The write priority in the new design is not same as the original UART. In the orginal design, software has higher priority than hardware when writing this register at the same time.
4. Note: The write priority in the new design is not same as the original UART. In the orginal design, software has higher priority than hardware when writing this register at the same time.

UARTx_UBMR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
MOD	**Modulator Denominator.** Holds the value of the denominator minus one of the BRM ratio (see Binary Rate Multiplier (BRM)). The UBIR register MUST be updated before the UBMR register for the baud rate to be updated correctly. If only one register is written to by software, the BRM will ignore this data until the other register is written to by software. Updating this register using byte accesses is not recommended and undefined.

17.2.14.14 UART Baud Rate Count Register (UARTx_UBRC)

Address: Base address + ACh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																BCNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

UARTx_UBRC field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
BCNT	**Baud Rate Count Register.** This read only register is used to count the start bit of the incoming baud rate (if ADNIMP=1), or start bit + bit0 (if ADNIMP=0). When the measurement is done, the Baud Rate Count Register contains the number of UART internal clock cycles (clock after divider) present in an incoming bit. BCNT retains its value until the next Automatic Baud Rate Detection sequence has been initiated. The 16 bit Baud Rate Count register is reset to 4 and stays at hex FFFF in the case of an overflow.

17.2.14.15 UART One Millisecond Register (UARTx_ONEMS)

NOTE

This register has been expanded from 16 bits to 24 bits. In previous versions, the 16-bit ONEMS can only support the maximum 65.535MHz (0xFFFFx1000) *ref_clk*. To support 4Mbps Bluetooth application with 66.5MHz *module_clock*, the value 0x103C4 (66.5M/1000) should be written into this register. In this case, the 16 bits are not enough to contain the 0x103C4. So this register was expanded to 24 bits to support high frequency of the *ref_clk*.

Address: Base address + B0h offset

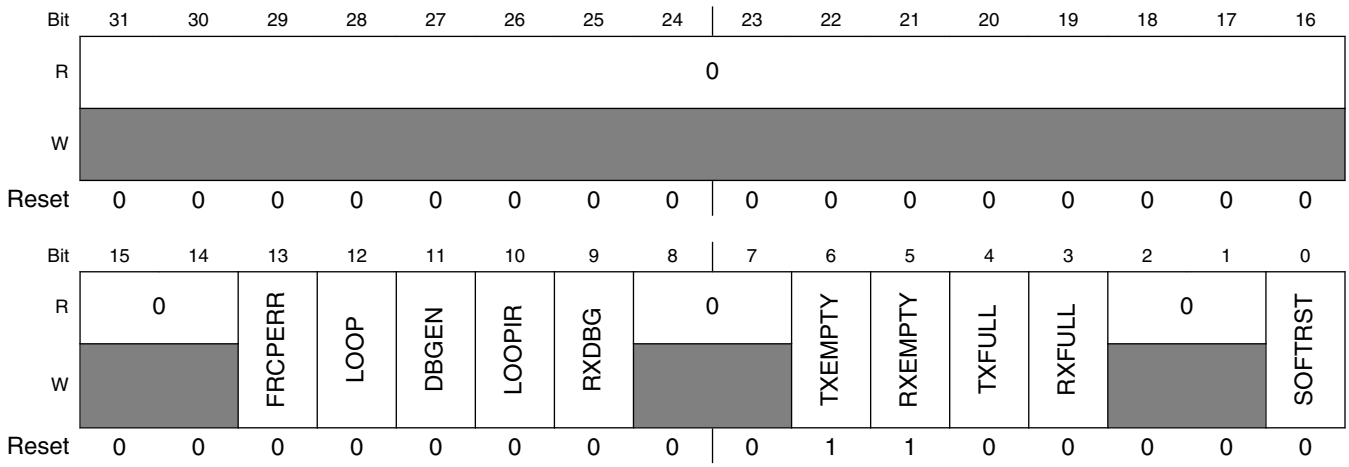
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ONEMS																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_ONEMS field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
ONEMS	**One Millisecond Register.** This 24-bit register must contain the value of the UART internal frequency (*ref_clk* in Block Diagram) divided by 1000. The internal frequency is obtained after the UART BRM internal divider (F (*ref_clk*) = F(*module_clock*) / RFDIV). In fact this register contains the value corresponding to the number of UART BRM internal clock cycles present in one millisecond. The ONEMS (and UTIM) registers value are used in the escape character detection feature (Escape Sequence Detection) to count the number of clock cycles left between two escape characters. The ONEMS register is also used in infrared special case mode (IRSC = UCR4[5] = 1'b1), see InfraRed Special Case (IRSC) Bit.

17.2.14.16 UART Test Register (UARTx_UTS)

Address: Base address + B4h offset



UARTx_UTS field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13 FRCPERR	Force Parity Error. Forces the transmitter to generate a parity error if parity is enabled. FRCPERR is provided for system debugging. 0 Generate normal parity 1 Generate inverted parity (error)
12 LOOP	Loop TX and RX for Test. Controls loopback for test purposes. When LOOP is high, the receiver input is internally connected to the transmitter and ignores the RXD pin. The transmitter is unaffected by LOOP. If RXDMUXSEL (UCR3[2]) is set to 1, the loopback is applied on serial and IrDA signals. If RXDMUXSEL is set to 0, the loopback is only applied on serial signals. 0 Normal receiver operation 1 Internally connect the transmitter output to the receiver input
11 DBGEN	This bit is not used in this chip. debug_enable_B**.** This bit controls whether to respond to the *debug_req* input signal. 0 UART will go into debug mode when *debug_req* is HIGH 1 UART will not go into debug mode even if *debug_req* is HIGH
10 LOOPIR	**Loop TX and RX for IR Test (LOOPIR)**. This bit controls loopback from transmitter to receiver in the InfraRed interface. 0 No IR loop 1 Connect IR transmitter to IR receiver
9 RXDBG	This bit is not used in this chip. **RX_fifo_debug_mode.** This bit controls the operation of the RX fifo read counter when in debug mode. 0 rx fifo read pointer does not increment 1 rx_fifo read pointer increments as normal

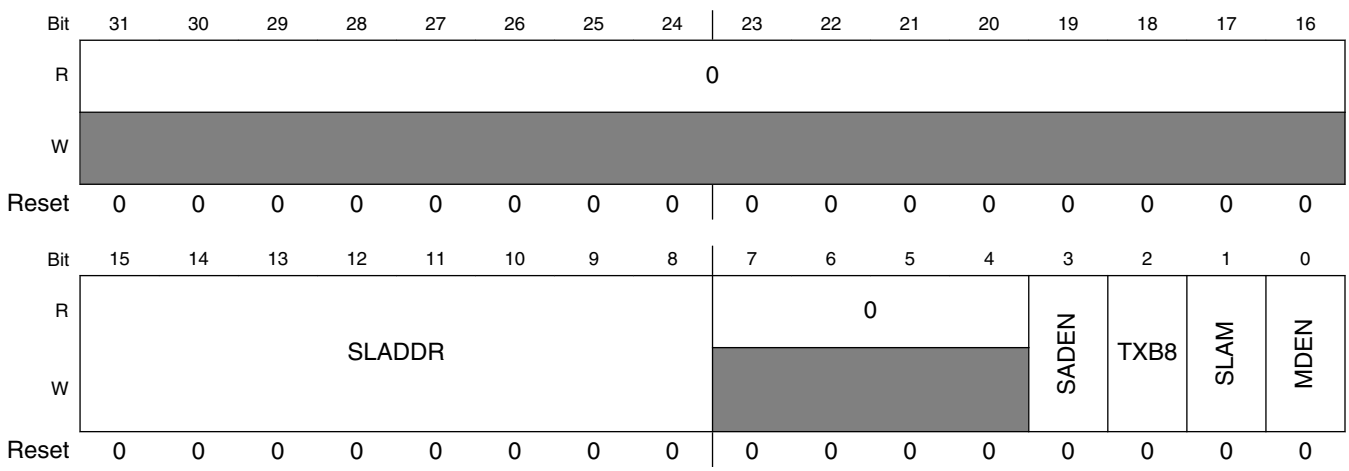
Table continues on the next page...

UARTx_UTS field descriptions (continued)

Field	Description
8–7 Reserved	This read-only field is reserved and always has the value 0.
6 TXEMPTY	TxFIFO Empty. Indicates that the TxFIFO is empty. 0 The TxFIFO is not empty 1 The TxFIFO is empty
5 RXEMPTY	RxFIFO Empty. Indicates the RxFIFO is empty. 0 The RxFIFO is not empty 1 The RxFIFO is empty
4 TXFULL	TxFIFO FULL. Indicates the TxFIFO is full. 0 The TxFIFO is not full 1 The TxFIFO is full
3 RXFULL	RxFIFO FULL. Indicates the RxFIFO is full. 0 The RxFIFO is not full 1 The RxFIFO is full
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SOFTRST	Software Reset. Indicates the status of the software reset (SRST_B bit of UCR2). 0 Software reset inactive 1 Software reset active

17.2.14.17 UART RS-485 Mode Control Register (UARTx_UMCR)

Address: Base address + B8h offset



UARTx_UMCR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 SLADDR	RS-485 Slave Address Character. Holds the selected slave adress character that the receiver wil try to detect.
7–4 Reserved	This read-only field is reserved and always has the value 0.
3 SADEN	RS-485 Slave Address Detected Interrupt Enable. 0 Disable RS-485 Slave Address Detected Interrupt 1 Enable RS-485 Slave Address Detected Interrupt
2 TXB8	Transmit RS-485 bit 8 (the ninth bit or 9th bit). In RS-485 mode, software writes TXB8 bit as the 9th data bit to be transmitted. 0 0 will be transmitted as the RS485 9th data bit 1 1 will be transmitted as the RS485 9th data bit
1 SLAM	RS-485 Slave Address Detect Mode Selection. 0 Select Normal Address Detect mode 1 Select Automatic Address Detect mode
0 MDEN	9-bit data or Multidrop Mode (RS-485) Enable. 0 Normal RS-232 or IrDA mode, see Table 17-2 for detail. 1 Enable RS-485 mode, see Table 17-2 for detail

Appendix A
SDMA Scripts

A.1 Introduction

This appendix provides descriptions of scripts that may be used to perform data transfers using the Smart DMA (SDMA) block of the SoC.

The SDMA block supports data transfer from core memory space to core memory, from core memory space to peripherals, and vice versa.

A.1.1 SDMA Scripts Overview

Table A-1 gives an overview of the SDMA scripts.

Table A-1. SDMA Scripts Overview

Data Transfer	Script to Use	Use Case	Parameters Through Context	Parameters Through Buffer Descriptor
Memory	ap_2_ap	Memory Copy	None	Word length (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, first address is memory source address, second address is memory destination address
Shared UART	uartsh_2_mcu	Uart Rx	UART Rx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01), Counter, Memory destination address, Second address not used
UART	uart_2_mcu	Uart Rx	Uart Rx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01), Counter, Memory source address, Second address not used
SPDIF	mcu_2_spdif	Spdif Playback	SPDIF Tx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b10 or 2'b00), Counter, Memory source address, Second address not used

Table continues on the next page...

Table A-1. SDMA Scripts Overview (continued)

Data Transfer	Script to Use	Use Case	Parameters Through Context	Parameters Through Buffer Descriptor
	spdif_2_mcu	Spdif Record	SPDIF Rx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b10 or 2'b00), Counter, Memory source address, Second address not used
Shared Peripheral	mcu_2_shp	Peripheral Transmit	Tx fifo address (r6) Event_mask (r1) Event2_mask(r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory source address, Second address not used
	shp_2_mcu	Peripheral Receive	RX fifo address (r6) Event_mask (r1) Event2_mask(r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory destination address, Second address not used
Peripheral	mcu_2_app	Peripheral Transmit	Tx fifo address (r6) Event_mask (r1) Event2_mask(r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory source address, Second address not used
	app_2_mcu	Peripheral Receive	RX fifo address (r6) Event_mask (r1) Event2_mask(r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory destination address, Second address not used
SSI	mcu_2_ssiapp	Audio Playback	Tx FIFO 0 address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory source address, Second address not used
	ssiapp_2_mcu	Audio Record	Rx FIFO 0 address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory destination address, Second address not used
Shared SSI	mcu_2_ssish	Audio Playback	Tx FIFO 0 address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory source address, Second address not used
	ssish_2_mcu	Audio Record	Rx FIFO 0 address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Mem destination address, Second address not used
Peripheral	p_2_p	Peripheral Transfer	Destination address (r6) Event_mask (r1) Event2_mask (r0) Source address (r2) Information (r7)	Counter, First address and Second address not used
HDMI	hdmi_dma	HDMI playback	Buffer chain length (r4) Pointer address (r6)	None
ECSPI	mcu_2_ecspi	ECSPI Tx	ECSPI Tx FIFO address (r6) Event_mask (r1) Event2_mask(r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory source address, Second address not used
SAI	mcu_2_sai	Audio playback with multi-FIFO script	Tx FIFO 0 address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory destination address, Second address not used
	sai_2_mcu	Audio record with multi-FIFO script	Rx FIFO address (r6) Event_mask (r1) Event2_mask(r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory destination address, Second address not used

Table continues on the next page...

Table A-1. SDMA Scripts Overview (continued)

Data Transfer	Script to Use	Use Case	Parameters Through Context	Parameters Through Buffer Descriptor
CAN FD	zcanfd_2_mcu	CAN FD Rx	CAN Rx FIFO address (r6) Event_mask (r1) Event2_mask(r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory destination address, Second address not used
QSPI	zqspi_2_mcu	QSPI Rx	QSPI Rx FIFO 0 address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory source address, Second address not used
	zzmcu_2_qspi	QSPI Tx	QSPI Tx FIFO 0 address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Mem destination address, Second address not used

In Table A-1, the data transfer column lists the possible types of DMA channels that involve the SDMA and a specific script is attached to each channel. It must be noted that some scripts cover several DMA channels. The scripts deal with the channels that have generic peripherals where only the watermark level is needed (no aging timer, no end_of_packet feature, and so on). The location refers to whether a script resides in ROM or in RAM and the size of the script is given in the instructions. The parameters columns are explained in the next section.

The following terms are used in Table A-1 which lists all the supported data transfers.

EMI or External Memory:

The external memories are connected to the External Memory Interface. Thus, a data transfer to EMI means a data transfer to any of the external memories (NAND Flash, NOR Flash, PSRAM, SDRAM, and so on).

Host mem or ARM internal memory:

The memory space accessible through the MAX of the ARM platform. It does not correspond to the peripherals, although they are accessible through the MAX as well. A data transfer from Host mem means data is read from the internal memory of the ARM or from the external memory. It is possible to point to an external memory through the MAX because it is also connected to EMI module. The next diagram replaces the SDMA and its DMA port in the hardware architecture. The Host mem is accessible through the Per DMA port of the SDMA, the EMI is accessible through the Burst DMA and the DSP mem through the DSP port.

Shared Peripheral:

A same peripheral can be connected to the shared peripheral bus (output of SPBA module) and to the ARM platform. This is the case for some UART, CSPI, and SSI. Therefore, "shared UART" indicates that the UART is connected to the shared peripheral, whereas "UART" means the UART is connected to the ARM platform.

Figure A-1 gives an overview of the SDMA in its hardware environment.

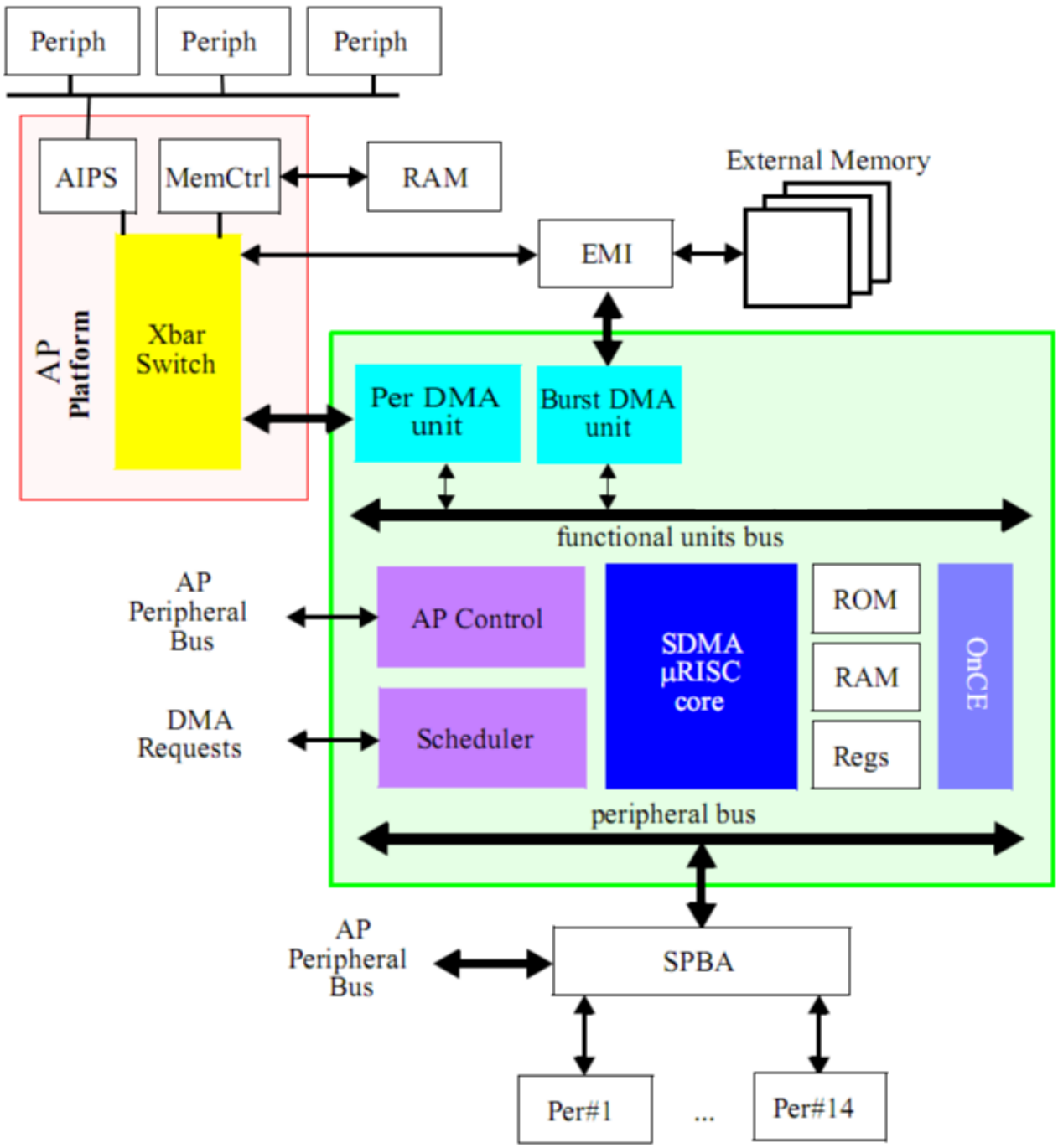


Figure A-1. SDMA Connections

A.2 Scripts Parameters: Definition

A.2.1 Parameters Definition

A.2.1.1 Parameters Required by Data Transfer Script Involving a Peripheral

The scripts that have a peripheral as a player in the transfer of data need to have the following input parameters:

A.2.1.1.1 Watermark Level (WML)

It is the upper or the lower threshold of the receive or transmit FIFO of the peripheral that triggers a DMA request to the SDMA.

The WML determines the data transfer loop size, meaning the number of bytes that will be read/write from/to the receive/transmit FIFO. This parameter must be a multiple of the peripheral FIFO data size. As an example, if UART1 is the data transfer destination, data must be written to the Tx FIFO of the peripheral. For instance, the watermark level can be set to 8 bytes (the UART is a 1 byte FIFO data size), meaning the UART_TX DMA request will be sent to the SDMA each time there are more than 8 free locations in the FIFO; therefore SDMA loop size will be set to 8 and 8 bytes will be written to the UART_TX FIFO. In fact, the loop size is the minimum between the WML and the count field of the buffer descriptor attached to the channel. This concept is explained later on in the chapter. Similarly, if the UART is the source of a data transfer and if the WML is set to 16, each time the DMA request is received by the SDMA, which triggers the associated channel, the data transfer loop size is set to 16. This means 16 bytes will be read from the UART_RX FIFO. The watermark level is a "long" (32-bit data) programmed by the ARM and is not supposed to evaluate a lot during application. The WML is always given in bytes.

NOTE

For the transmit FIFO, the WML = FIFO_SIZE - FIFOTX_THRESHOLD For the receive FIFO, the WML = FIFORX_THRESHOLD The FIFOTX_THRESHOLD/ FIFORX_THRESHOLD are the values of the peripheral transmit and receive FIFO level at which a DMA request is triggered. For instance, if TX FIFO is 32 bytes depth and if the DMA request is sent when the number of bytes present in the FIFO is below a threshold of 10, the WML will be 22. It means that when the SDMA will receive the DMA request from the

peripheral, it will be possible to store 22 bytes in the TX FIFO.
For the receive FIFO, the WML equals the threshold.

A.2.1.1.2 Event_mask and Event2_mask

As previously said, when the WML threshold is over or under passed, a DMA request is sent to the scheduler part of the SDMA.

The SDMA scheduler has 48 input pins, called "events", which are connected to the different DMA request.

The mapping between the DMA request name and the event number is SoC dependent and defined in the "Interrupts and SDMA Events" chapter. For instance, in SoC A, the UART_RX DMA request is connected to events pin 5; whereas for SoC B, it may be connected to events pin 15. The script reads the EVENTS(32 events requests) and EVENTS2(remaining 16 events requests) register to check if the received DMA request is compliant with the expected one and it guaranteed that a second DMA request sent during the data transfer is not lost.

For instance, while SDMA is reading the UART_RX FIFO (again, the number of data to be read is given by WML value), the peripheral may receive new data from its input ports. Therefore after the first data transfer, the number of data available in the UART received FIFO can be still higher than the WML threshold, so a second DMA request may be sent.

The event_mask value is generally a 1-bit high value, which means that if the script attached to the channel must be triggered by DMA request number I, event_mask[I] must be set to 1, if the script attached to the channel must be triggered by DMA request number 32+I, event2_mask[I] must be set to 1. Some scripts (like ata_2_mcu) may be triggered by several DMA requests; in that case several bits of the event_mask must be asserted. The event_mask and event2_mask are long (32-bit) data.

A.2.1.1.3 Peripheral Address

The scripts of the SDMA scripts library are SoC independent but as their goal is to address peripheral, they required the base address or the FIFO address of the peripheral.

Passing FIFO or peripheral base address depends on the scripts.

A.2.1.1.4 Data Length

The scripts whose name contains the key word "app" or "shp" are generic: they are used to transfer data from/to a 8, 16, 24 or 32-bit peripheral data size.

Some jumps to some routines of these scripts depend on this peripheral size. This parameter is required as input of these scripts. This parameter is passed through the command field of the buffer descriptor and is coded on bits 25 and 24 of the mode:

00: 32-bit data transfer.

01: 8-bit data transfer.

10: 16-bit datatransfer.

11: 24-bit data transfer.

A.3 Scripts

The following scripts are all based on buffer descriptor mechanism.

A.3.1 SDMA ROM Scripts

The ROM holds the generic memory to memory scripts (ap_2_ap, ap_2_bp, bp_2_ap, bp_2_bp) and the peripheral most useful scripts (app_2_mcu, mcu_2_app, uart_2_mcu, uartsh_2_mcu, mcu_2_shp, shp_2_mcu).

In these memory to memory scripts, the number of bytes to transfer from the source memory to the destination is divided by 4 to get the number of 32-bit words that can be transferred. Most of the time, the transfer will be a transfer of words. There will be a byte or half data transfer at the end of the transfer if the total number of bytes to transfer is not an multiple of 4.

A.3.1.1 ap_2_ap

This script is used to transfer data inside the application processor memory using BurstDMA.

Using this script there is no restriction on the number of bytes to transmit, the source address and destination address alignment, or on the source memory and destination memory type. But word alignment and source/destination memory type will impact performance. In case that the source and destination are in the same memory type, being word aligned allows the SDMA to use the copy capability of the DMA units. When it is not the case the SRAM of the SDMA is used as temporary buffer.

A.3.1.1.1 Parameters Transmitted Through the Context ap_2_ap

A.3.1.1.2 Parameters Transmitted Through the Buffer Descriptor ap_2_ap

The number of bytes to transmit is stored in the first Buffer descriptor word.

The source address is stored in the second Buffer descriptor word (address field).

The destination address is stored in the Extended Buffer address

A.3.1.1.3 Use of the General Register During the Execution ap_2_ap

- r0: nb of bytes to transfer, counter during loop
- r2: channel cb address, scratch
- r3: current buffer descriptor pointer, AP destination address, scratch
- r4: mode
- r5: source address loaded by getbd,
- r6: destination address loaded by getcb, scratch
- r7: pointer in the scratch buffer

A.3.1.1.4 Overview of Script Functionality ap_2_ap

The parameters of the transfer (number of bytes to transfer, source address and destination address) are retrieved from the buffer descriptor.

Depending on the address alignment and the DMA units involved, if possible, the transfer is performed using the copy capability of the DMA unit, or using the SDMA SRAM as a temporary buffer.

When the transfer is finished, this algorithm restarts (a new buffer descriptor is read).

The count in the BD mode word after the script is over is meaningful only if error was reported, else it has the same value as was fed as input to the script.

A.3.1.2 uartsh_2_mcu

This script is used to transfer data from the shared UART to the External memory. The UART is connected to the ARM platform.

The UART is an 8-bit peripheral, but when reading the FIFO, 16 bits are retrieved. In fact, the 8 MSB are the status bytes for the current data. At each access the value of this byte is checked to verify that a valid data was retrieved. If an error is detected the transfer is stopped and the information is sent back to the Host through the buffer descriptor with the byte count field updated.

A.3.1.2.1 Parameters Transmitted Through the Context uartsh_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Rx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.3.1.2.2 Parameters Transmitted Through the Buffer Descriptor uartsh_2_mcu

The number of bytes to transmit is stored in the first Buffer descriptor word.

The destination address is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.2.3 Use of the General Register During the Execution uartsh_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = AP M3 destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.2.4 Overview of Script Functionality uartsh_2_mcu

When the UART sends a DMA request, it can be for three different reasons.

When the RX threshold is over passed. It deals with the most frequent event.

When the AGEING timer has reached its final value, meaning there is less data than the RX threshold in the RX FIFO, and they have been present for too long. It is called the aging DMA request.

When IDLE timer reached its final value; meaning the RX FIFO has been empty for too long. It is called the iddle dma request. The IDLE timer is not used, so only normal and AGEING DMA requests are supported by the script.

The first time the script is executed, a buffer descriptor is opened, the buffer destination address and its size are retrieved from the BD.

Then the script checks if the RRDY bit of USR1 is set. If yes, it means that the Watermark level has been reached, there are "WML" bytes in RX FIFO. If the count field of the opened BD is higher than WML-1, the SDMA script will perform WML-1 read accesses to the RX FIFO and WML write access to the destination buffer (a read access then a write access is call a data transfer loop). If the count field is lesser than the WML-1 value, the count field will be the data transfer loop size. So the data transfer loop size equals min(WML-1,BD count)

So there will be always one data remaining in the UART RX FIFO after every data transfer loop and the ageing timer will always trigger a DMA request. It means that the channel on which the script is run will be always closed on an AGEING DMA request.

Now assume the script is triggered by a DMA request, but RRDY is not set. It means that it deals with an AGEING DMA request; there is at least one data in the RX FIFO. The data is read and written into the destination buffer, and then the RDR is checked again. If it is set and if the destination data buffer is not full (BD count is not 0), a new data is read and so on until RDR is 0. When RDR is 0, the ageing timer interrupt flag is disabled by setting the AGTIM bit, the count field of BD is updated and the current BD is closed. Then the script tries to open the next BD if the Continuous bit of the current BD was set, if there is no more BD to open, the channel is stopped.

When reading a data from the UART RX FIFO, the MSB are check to verify that a valid data has been retrieved. If an error is detected the transfer is stopped, the error bit is set in the BD, the count is updated, the BD is closed and an interrupt is sent to the ARM.

When the buffer is full, this algorithm restarts (a new buffer descriptor is read, if it exists).

The next diagram illustrates the script algorithm, it is quite complex but things become more obvious when we notice that there is a loop for the normal DMA request (in sky-blue) and on when the AGEING DMA request was detected (in yellow).

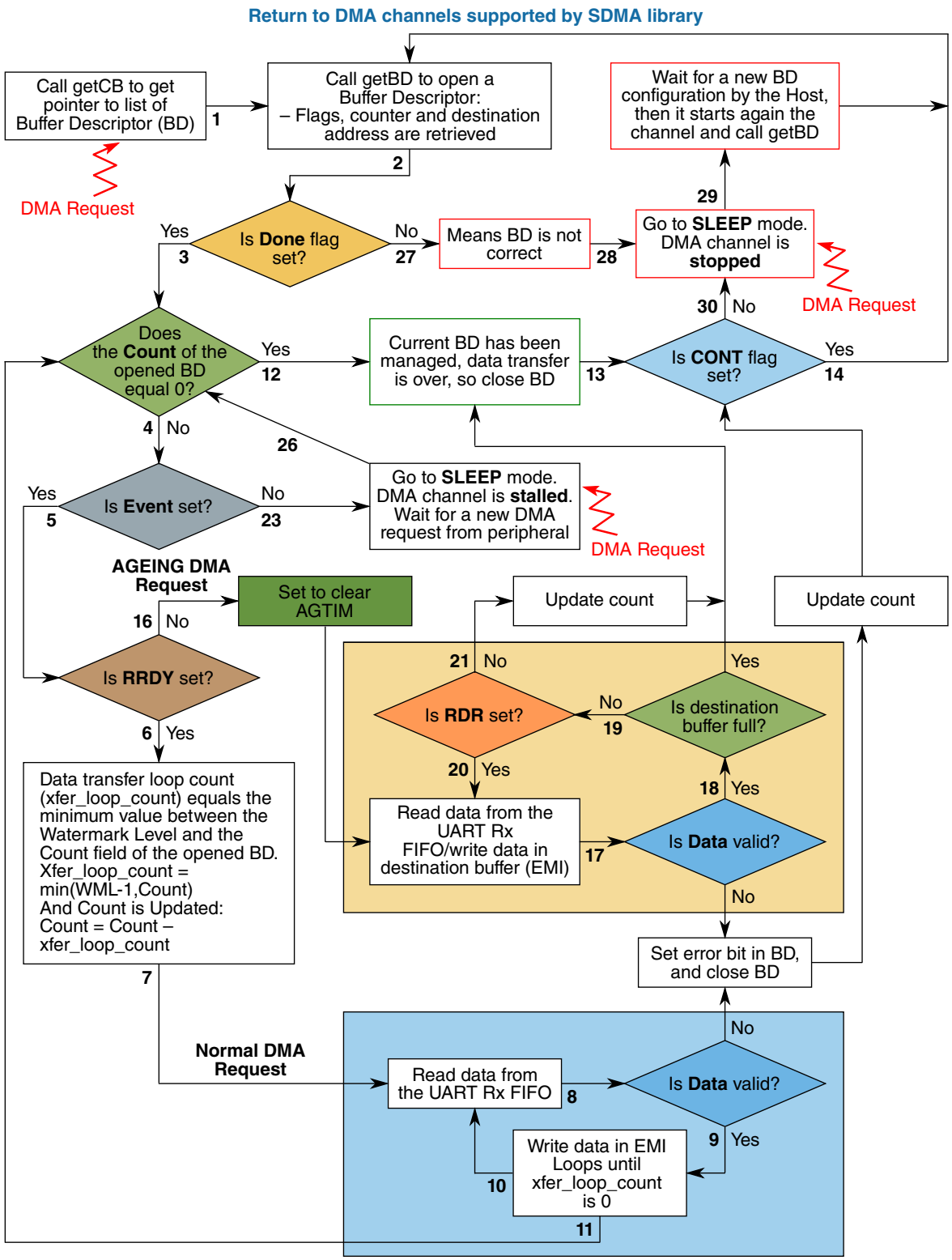


Figure A-2. Script Algorithm

A.3.1.3 shp_2_mcu

This generic script is used to transfer data from a 8, 16, 24 or 32-bit peripheral connected to the shared peripheral bus accessed through the SPBA to memories accessed by the BurstDMA (External memories).

It can be used for SSI (8, 16, 24 or 32-bit data size), for CSPI (32-bit data size) or for SDHC(MMC)/SIM (16-bit data size), these peripherals being connected to the shared peripheral bus.

A.3.1.3.1 Parameters Transmitted Through the Context shp_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Rx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.3.1.3.2 Parameters Transmitted Through the Buffer Descriptor shp_2_mcu

The peripheral size/data length is set in the command field of the first Buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first Buffer descriptor word (count field)

The destination address in the external memory is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.3.3 Use of the General Register During the Execution shp_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = BD destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.3.4 Overview of Script Functionality shp_2_mcu

The parameters of the transfer (peripheral size, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the peripheral sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" data in the FIFO.

During this data transfer loop, the number of bytes that are transferred from the RX FIFO of the peripheral to the EMI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory write access is fixed by the parameter peripheral size.

The necessary number of transfer loops are executed in order to fill the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is filled, this algorithm restarts (a new buffer descriptor is read).

A.3.1.4 mcu_2_shp

This generic script is used to transfer data from memories accessed by the BurstDMA (External memories) to a 8, 16, 24 or 32-bit peripheral connected to the shared peripheral bus accessed through the SPBA

. It can be used for SSI (8, 16, 24 or 32-bit data size), for CSPI (32-bit data size), for SDHC(MMC)/SIM (16-bit data size) or for UART3,4 (8-bit data size), these peripherals being connected to the shared peripheral bus.

A.3.1.4.1 Parameters Transmitted Through the Context mcu_2_shp

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check events - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Tx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be retrieved from the peripheral each time the channel is started.

A.3.1.4.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_shp

The peripheral size/data length is set in the command field of the first Buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first Buffer descriptor word (count field)

The source address in the external memory is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.4.3 Use of the General Register During the Execution mcu_2_shp

r0 = mask to check events2

r1 = mask to check events

r2 = EVENTS / EVENTS2

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.4.4 Overview of Script Functionality mcu_2_shp

The parameters of the transfer (peripheral size, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the peripheral sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" empty rooms in the FIFO.

During this data transfer loop, the number of bytes that are transferred from the EMI to the TX FIFO of the peripheral is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory read access is fixed by the parameter peripheral size.

The necessary number of transfer loops are executed in order to empty the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is empty, this algorithm restarts (a new buffer descriptor is read).

A.3.1.5 uart_2_mcu

This script is similar to uartsh_2_mcu script. The only difference between these scripts is the way SDMA access UART RX FIFO.

In case of uartsh_2_mcu, the UART RX FIFO can directly be accessed by SDMA as it is on Shared Peripheral Bus. In case of uart_2_mcu, SDMA script uses one of its DMA (Peripheral DMA) to access UART RX FIFO.

A.3.1.6 app_2_mcu

This generic script is used to transfer data from a 8, 16, 24 or 32-bit peripheral connected to the AIPS accessed through the Periphera DMA of SDMA, to memories accessed by the BurstDMA (External memories).

It can be used for SSI (8, 16, 24 or 32-bit data size), for CSPI (32-bit data size) or for SDHC(MMC)/SIM (16-bit data size).

A.3.1.6.1 Parameters Transmitted Through the Context app_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Rx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.3.1.6.2 Parameters Transmitted Through the Buffer Descriptor app_2_mcu

The Data Length peripheral size/data length is set in the command field of the first buffer descriptor word, specially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field).

The destination address in the external memory is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.6.3 Use of the General Register During the Execution app_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = AP M3 destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.6.4 Overview of Script Functionality app_2_mcu

The parameters of the transfer (peripheral size, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the peripheral sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" data in the FIFO.

During this data transfer loop, the number of bytes that are transferred from the RX FIFO of the peripheral to the EMI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory write access is fixed by the parameter peripheral size.

The necessary number of transfer loops are executed in order to fill the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is filled, this algorithm restarts (a new buffer descriptor is read).

A.3.1.7 mcu_2_app

This generic script is used to transfer data from memories accessed by the BurstDMA (External memories), to a 8, 16, 24 or 32-bit peripheral connected to the AIPS accessed through the Periphera DMA of SDMA.

It can be used for SSI (8, 16, 24 or 32-bit data size), for CSPI (32-bit data size) or for SDHC(MMC)/SIM (16-bit data size).

A.3.1.7.1 Parameters Transmitted Through the Context mcu_2_app

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Tx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.3.1.7.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_app

The peripheral size/data length is set in the command field of the first buffer descriptor word, specially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field).

The source address in the external memory is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.7.3 Use of the General Register During the Execution mcu_2_app

r0 = mask to check events2

r1 = mask to check events

r2 = TX FIFO address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.7.4 Overview of Script Functionality mcu_2_app

The parameters of the transfer (peripheral size, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the peripheral sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" empty room in the FIFO.

During this data transfer loop, the number of bytes that are transferred from the EMI to the TX FIFO of the peripheral is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory read access is fixed by the parameter peripheral size.

The necessary number of transfer loops are executed in order to fill the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is filled, this algorithm restarts (a new buffer descriptor is read).

A.3.1.8 mcu_2_spdif

This script performs a data move from AP buffer to 24 bits FIFO of the SPDIF.

This peripheral is on the peripheral bus, the SDMA accesses it through the SPBA. The MCU buffer is accessed through the burst DMA.

A.3.1.8.1 Parameters Transmitted Through the Context mcu_2_spdif

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check events - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: Address of SPDIF STL register

r7: Watermark level - Used to determine the maximum data that can be sent to the peripheral each time the channel is started.

A.3.1.8.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_spdif

The data length (Either 32-bit or 16-bit) is set in the command field of the first Buffer descriptor word, especially bits 24, 25.

The number of bytes to transmit is stored in the first Buffer descriptor word (count field)

The source address in the external memory is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.8.3 Use of the General Register During the Execution mcu_2_spdif

r0 = mask to check events2

r1 = mask to check events

r2 = EVENTS / EVENTS2

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.8.4 Overview of Script Functionality mcu_2_spdif

The parameters of the transfer (data length, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the spdif sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" of empty room in the FIFO (Sum of left and right FIFO's).

During this data transfer loop, the number of bytes that are transferred from the EMI to the TX FIFO of the spdif is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory read access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to empty the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is empty, this algorithm restarts (a new buffer descriptor is read).

A.3.1.9 spdif_2_mcu

This script performs a data move from 24 bits FIFO of the SPDIF to AP buffer.

This peripheral is on the peripheral bus, the SDMA accesses it through the SPBA. The MCU buffer is accessed through the burst DMA.

A.3.1.9.1 Parameters Transmitted Through the Context spdif_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check events - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: Address of SPDIF SRL register

r7: Watermark level - Used to determine the maximum data that can be sent to the peripheral each time the channel is started.

A.3.1.9.2 Parameters Transmitted Through the Buffer Descriptor spdif_2_mcu

The data length (Either 32 bit or 16 bit) is set in the command field of the first Buffer descriptor word, especially bits 24, 25.

The number of bytes to transmit is stored in the first Buffer descriptor word (count field)

The destination address in the external memory is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.1.9.3 Use of the General Register During the Execution spdif_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = EVENTS / EVENTS2

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.1.9.4 Overview of Script Functionality spdif_2_mcu

The parameters of the transfer (data length, number of bytes to transfer and destination address) are retrieved from the buffer descriptor.

When the spdif sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" date in the FIFO (Sum of left and right FIFO's).

During this data transfer loop, the number of bytes that are transferred from the RX FIFO of the spdif to the EMI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory write access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to empty the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is empty, this algorithm restarts (a new buffer descriptor is read).

A.3.2 SDMA RAM scripts

A.3.2.1 mcu_2_ssiapp

This script performs a data move from AP buffer to 8, 16, 24 or 32-bit FIFO of the SSI in the AP region with 2 FIFOs enabled.

The MCU buffer is accessed through the burst DMA, while SSI is accessed with Per DMA.

A.3.2.1.1 Parameters Transmitted Through the Context mcu_2_ssiapp

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check events - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: Address of SSI Transmit FIFO 0 register

r7: Watermark level - Used to determine the maximum data that can be sent to the peripheral each time the channel is started.

A.3.2.1.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_ssiapp

The data length (8, 16, 24 or 32-bit) is set in the command field of the first buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field).

The source address in the memory is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.2.1.3 Use of the General Register During the Execution mcu_2_ssiapp

r0 = mask to check events2

r1 = mask to check events

r2 = Tx FIFO address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.2.1.4 Overview of Script Functionality mcu_2_ssiapp

The parameters of the transfer (data length, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the SSI sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" of empty room in the FIFO (Sum of left and right FIFO's).

During this data transfer loop, the number of bytes that are transferred from the EMI to the TX FIFO of the SSI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory read access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to empty the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is empty, this algorithm restarts (a new buffer descriptor is read).

A.3.2.2 ssiapp_2_mcu

This script performs a data move from the 8, 16, 24 or 32-bit FIFO of the SSI to the AP buffer.

The SSI is in the AP region and 2 FIFO enabled. The MCU buffer is accessed through the burst DMA while the SSI is accessed through Per DMA.

A.3.2.2.1 Parameters Transmitted Through the Context ssiapp_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the SSI RX FIFO 0

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.3.2.2.2 Parameters Transmitted Through the Buffer Descriptor ssiapp_2_mcu

The data length (8, 16, 24, 32 bits) is set in the command field of the first buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field)

The destination address is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used

A.3.2.2.3 Use of the General Register During the Execution ssiapp_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = AP M3 destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.2.2.4 Overview of Script Functionality ssiapp_2_mcu

When the SSI sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" of data in the FIFO(sum of left and right FIFOs).

During this data transfer loop, the number of bytes that are transferred from the RX FIFO of the SSI to the EMI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory write access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to fill the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is full, this algorithm restarts (a new buffer descriptor is read).

A.3.2.3 mcu_2_ssish

This script is similar to mcu_2_ssiapp.

The only difference is the way SDMA core accesses the SSI TX FIFO. For script of mcu_2_ssiapp, SDMA core accesses the TX FIFO through Functional Unit Bus and regards it as the peripheral DMA, whereas for the mcu_2_ssish, the access is through the Shared Peripheral Bus and SSI can be directly accessed by SDMA.

A.3.2.4 ssish_2_mcu

This script is similar to ssiapp_2_mcu. The only difference is the way SDMA core accesses the SSI RX FIFO.

For script of ssiapp_2_mcu, SDMA core accesses the RX FIFO through Functional Unit Bus and regards it as the peripheral DMA, whereas for the ssish_2_mcu, the access is through the Shared Peripheral Bus and SSI can be directly accessed by SDMA.

A.3.2.5 p_2_p

This script performs a data move of 32 bits from the peripheral's FIFO connected either on SPBA or AIPS bus to another.

If destination FIFO belongs to ASRC then pad adding is performed after every N samples from the source device. If the source FIFO belongs to ASRC then after transmitting N samples a pad is swallowed. In case there is a transaction between ASRC(source) and SPDIF (destination) no pad adding or swallowing is required because SPDIF expects even number of samples.

This script presently does not deal with the transactions involving two devices connected to AIPS bus.

A.3.2.5.1 Parameters Transmitted Through the Context p_2_p

r0: LWML event mask

r1: HWML event mask

r2: source address

r6: destination address

r7: INFO. See details in the following table.

p_2_p (r7 INFO)

Bits	Name	Description
0-7	Lower WML	Watermark Level
8	PS	1 : Pad Swallowing 0 : No Pad swallowing
9	PA	1 : Pad Adding 0 : No Pad swallowing
10	SPDIF	If this bit is set both source and destination are on SPBA
11	Source Bit(SP)	1 : Source on SPBA 0 : Source on AIPS
12	Destination Bit (DP)	1 : Destination on SPBA 0 : Destination on AIPS
13-15	----------	MUST BE 0
16-23	Higher WML	HWML
24-27	N	Total number of samples after which Pad adding/Swallowing must be done. It must be odd.
28	Lower WML Event (LWE)	SDMA events reg to check for LWML event mask 0 : LWE in EVENTS register 1 : LWE in EVENTS2 register
29	Higher WML Event(HWE)	SDMA events reg to check for HWML event mask 0 : HWE in EVENTS register 1 : HWE in EVENTS2 register
30	----------	MUST BE 0
31	CONT	1 : Amount of samples to be transferred is unknown and script will keep on transferring samples as long as both events are detected and script must be manually stopped by the application 0 : The amount of samples to be is equal to the count field of mode word

A.3.2.5.2 Parameters Transmitted Through the Buffer Descriptor p_2_p

In the first buffer descriptor word, the mode and number of types to transfer is included.

The Buffer address and Extended Buffer address are not used.

A.3.2.5.3 Use of the General Register During the Execution p_2_p

The register usage for each case(spba_2_spba, asrc_2_spba…) are different, so don't present here.

A.3.2.5.4 Overview of Script Functionality p_2_p

The script get the number of data to transfer per event according to the bit 31(CONT) of INFO.

If CONT is 0, each transfer loop will transmit the number of data as in the mode count; Or else, it means the number of samples to be transferred is not known and the transfer must take place as long as both events are detected and the script must be manually stopped by the application in case the transfer needs to be stopped.

When PS or PA is set, then after each N words transfer, a pad swallowing or pad adding will be done. This is to handle the ASRC involved transfer.

The transfer is executed only when the event is set. After each transfer for the event, the script will try to obtain a new BD.

A.3.2.6 hdmi_dma

This script sets up the HDMI DMA start and end address from the buffer chain, and launches an HDMI DMA transfer on the HDMI DMA done event.

A.3.2.6.1 Parameters Transmitted Through the Context hdmi_dma

r4: Number of DMA buffers in loop

r6: Address of data structure setup by software. Detailed data structure below:

Table A-2. SDMA Scripts Overview

Name	Offset	Width (bytes)	Description
CONTROL	0	4	Address of HDMI DMA control register
STATUS	4	4	Address of HDMI DMA status register
START	8	4	Address of HDMI DMA start address register
START1	12	4	Start address of buffer 1 for HDMI DMA transfer
END1	16	4	End address of buffer 1 for HDMI DMA transfer
START2	20	4	Start address of buffer 1 for HDMI DMA transfer. Available only when value of r4>1
END2	24	4	End address of buffer 1 for HDMI DMA transfer. Available only when value of r4>1

Table continues on the next page...

Table A-2. SDMA Scripts Overview (continued)

Name	Offset	Width (bytes)	Description
STARTn	4+8*n	4	Start address of buffer 1 for HDMI DMA transfer. n is the value of r4.
ENDn	8+8*n	4	End address of buffer 1 for HDMI DMA transfer. n is ghe value in r4.

A.3.2.6.2 Parameters Transmitted Through the Buffer Descriptor hdmi_dma

None

A.3.2.6.3 Use of the General Register During the Execution hdmi_dma

r0 = N/A

r1 = control register address

r2 = status register address

r3 = DMA start address register address

r4 = offset of DMA buffer chain end

r5 = offset of current DMA buffer

r6 = pointer of data structure that described above

r7 = N/A

A.3.2.6.4 Overview of Script Functionality hdmi_dma

The parameters of the data structure used in the script are retrieved from the channel context. No parameter is retrieved through the buffer descriptor as no real SDMA transfer performed.

When the HDMI DMA done request is captured, the script will clear the DMA done bit in the status register and set up the DMA start and end address to the next buffer (rewind if it reaches the end of the buffer chain). Next it will restart the HDMI DMA transfer, then yield and wait for the next DMA done request.

When an error is captured, it will stop the channel.

A.3.2.7 mcu_2_ecspi

This script is similar to mcu_2_app, but it uses SPI Exchange (XCH) as PIO mode in the SDMA script instead of using Start Mode Control (SMC), for the i.MX6/7/8M Family of Applications Processors affected by the Errata (ERR009165) on ECSPI. Please note, in this case, do not program the ECSPIx_CONREG[XCH] bit since the script will set XCH every time after the FIFO is filled.

A.3.2.7.1 Parameters Transmitted Through the Context mcu_2_ecspi

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1.

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1.

r6: address of the peripheral TX FIFO.

r7: Watermark level - Used to determine the maximum data that can be sent to the peripheral each time the channel is started. Bit 31 is used by the script to determine if ERR009165 is fixed. If ERR009165 is fixed, set bit 31 to 1, so that the script uses SMC mode for higher performance.

A.3.2.7.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_ecspi

The peripheral size/data length is set in the command field of the first buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field).

The source address in the external memory is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.2.7.3 Use of the General Register During the Execution mcu_2_ecspi

r0 = mask to check events2

r1 = mask to check events

r2 = TX FIFO address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.2.7.4 Overview of Script Functionality mcu_2_ecspi

The functionality of this script is similar to mcu_2_app, with the exception of the following:

1. First, the script checks the value of the ECSPIx_TESTREG[RXCNT] bit field. No action is performed if there is a lot of data (3/4 of FIFO size) that cannot be fetched in time from the RX FIFO. Otherwise, filling the TX FIFO may cause the ECSPI RX FIFO to overflow since the same size of data will be shifted in during data shift out.
2. The script checks the value of bit 31 of r7 (Watermark). If bit 31 is set to 1, this step is ignored. If bit 31 is not set, the mode is set to XCH every time after the FIFO is filled, to trigger SPI transfer.

A.3.2.8 mcu_2_sai

This script is used for the audio multi-FIFO case, where the destination/source address is not fixed, but incremental and the address will rollback to the first FIFO after traversing all the FIFO's. This script is based on the mcu_2_shp script, with additional parameter defines added for r7 (Watermark).

A.3.2.8.1 Parameters Transmitted Through the Context mcu_2_sai

r0: mask to check events2

r1: mask to check events

r6: address of the peripheral TX FIFO

r7: Details are provided in the following table.

Bits	Name	Description
0-11	WML	Watermark level (wml*fifo_number)
12-15	FIFO number	Number of FIFOs to fill.
16-19	FIFO offset	Offset between two FIFOs. Note that the offset between all the FIFOs should be the same.

Table continues on the next page...

Bits	Name	Description
		0: No offset between the FIFOs, which means it is continuous. For example, FIFO1 address is 0x30010020 and FIFO2 address is 0x30010024. 1: 1 word offset between the FIFOs. For example, FIFO1 address is 0x30010020 and FIFO2 is 0x30010028 ... N: N words offset between the FIFOs.
23	SW done	SW done enabled/disabled. Some modules cannot clear events automatically once the DMA request is serviced, so SW done is used to notify the module that the DMA request is done.
24-27	SW done selector	SW done selector for the right IP/event. Please refer to the SDMAARMx_DONE0_CONFIG and SDMAARMx_DONE1_CONFIG registers for additional details.
28-31	Words per FIFO	Number of words per FIFO. 0: Only one word per FIFO filling, then jump to the next FIFO. 1: Two words per FIFO filling, then jump to the next FIFO. ... N: N words per FIFO filling, then jump to the next FIFO.

A.3.2.8.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_sai

The peripheral size/data length is set in the command field of the first buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field).

The source address in the external memory is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.2.8.3 Use of the General Register During the Execution mcu_2_sai

r0 = mask to check events2

r1 = mask to check events

r2 = EVENTS / EVENTS2

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.2.8.4 Overview of Script Functionality mcu_2_sai

This script is based on the mcu_2_shp script, with additional parameters for audio multi-FIFO cases. The additional parameters are defined in Table A-2.

For example, if there are 4 FIFOs, 1 word offset between the FIFOs, 2 words per FIFO, and SW done selector 0, the script runs as shown below:

1. Check r7 to get the value of the fields: FIFO number, FIFO offset, number of words per FIFO, SW done selector, and the other fields defined for r7.
2. Start to fill FIFO1 with 2 words continuously and then jump to FIFO2 (FIFO1 + 2 * 4).
3. After filling FIFO4 with 2 words continuously, rollback to FIFO1.
4. Repeat steps 3 and 4 until FIFO1-4 fill to the Watermark level.
5. Configure the SW done selector to trigger SW done and wait for the next DMA request event to jump to step 1 again.

A.3.2.9 sai_2_mcu

This script is similar to the mcu_2_sai script. The only difference is the direction of the data transfer. This script performs a data move from the FIFO of the SAI to AP buffer.

A.3.2.10 zcanfd_2_mcu

This script is based on the app_2_mcu script. This script adds a 'done 4' instruction that triggers every time the Watermark level is reached, since CAN FD cannot clear the DMA event automatically. The RX FIFO of zcanfd is incremental.

A.3.2.10.1 Parameters Transmitted Through the Context zcanfd_2_mcu

r0: mask to check events2

r1: mask to check event

r6: address of the peripheral RX FIFO.

r7: Watermark level - Used to determine the maximum data that can be sent to the peripheral each time the channel is started.

A.3.2.10.2 Parameters Transmitted Through the Buffer Descriptor zcanfd_2_mcu

The peripheral size/data length is set in the command field of the first buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field).

The destination address in the external memory is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used.

A.3.2.10.3 Use of the General Register During the Execution zcanfd_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = AP M3 destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.3.2.10.4 Overview of Script Functionality zcanfd_2_mcu

This script is based on app_2_mcu script. This script adds a 'done 4' instruction so that the SDMA can clear the DMA request event (EP), since CAN FD cannot clear the DMA request event automatically after all the data in RX FIFO is fetched.

A.3.2.11 zqspi_2_mcu

This script is similar to zcanfd_2_mcu script.

A.3.2.12 zzmcu_2_qspi

This script performs a data move from AP buffer to FIFO of the QSPI.

This script is similar to zqspi_2_mcu script, with an additional parameter (bit 31) defined for r7.

Bit31 is used to determine if the FIFO address is incremental (bit31 = 1) or fixed (bit31 = 0).

Appendix B
Revision History

B.1 Substantive changes from revision 2 to revision 3

Substantive changes from revision 2 to revision 3 are as follows:

B.1.1 System Debug Revision History

Topic Cross-Reference	Change Description
Debug System Features	• Updated the features list to remove the support of 4 JTAG security levels

B.1.2 Fusemap Revision History

Topic Cross-Reference	Change Description
Lock Fusemap	• Added GP5_LOCK fuse at address 0x410[9:8] • Added the note "The GP5_LOCK fuse locks..... .....Field Representative". at the end of the table. • Added the note "Definitions for fuse settings are as follows...."
Fusemap Descriptions Table	• Added GP5_LOCK fuse at address 0x410[9:8] • Reserved at 0x410[7:6] • Removed the note "Definitions for fuse settings are as follows...."

B.1.3 IOMUXC Revision History

Topic Cross-Reference	Change Description
IOMUXC_GPR22	Removed IOMUXC_GPR_GPR22[SJC_CHALLENGE_RESPONSE_AUTHENTICATION_FAIL] bitfield.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Arteris — Portions of this document contain content provided by Arteris, Inc. or its subsidiaries (collectively, "Arteris"). Arteris is the sole owner of such content.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

eIQ — is a trademark of NXP B.V.

I2C-bus — logo is a trademark of NXP B.V.

Microsoft, Azure, and ThreadX — are trademarks of the Microsoft group of companies.

Oracle and Java — are registered trademarks of Oracle and/or its affiliates.

Synopsys & Designware — are registered trademarks of Synopsys, Inc.

Synopsys — Portions Copyright © 2018-2022 Synopsys, Inc. Used with permission. All rights reserved.

TensorFlow, the TensorFlow logo and any related marks — are trademarks of Google Inc.

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© NXP B.V. 2024. All rights reserved.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: August 2024
Document identifier: IMX8MPRM